



El Desastre Humano

(Projecte de desenvolupament)
CFGM Sistemes Microinformàtics i Xarxes

Jiayi Chen i Wenjing Chen
SMX2A
2023-2024



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 4.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/)

B) GNU Free Documentation License (GNU FDL)

Copyright © ANY Wenjing Chen / Jiayi Chen

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Wenjing C. / Jiayi C.)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

ÍNDICE

ÍNDICE	2
FICHA DE NUESTRO TRABAJO FINAL	5
1. Introducción	6
1.1 Contexto y justificación del nuestro trabajo.....	6
1.2 Objetivos del nuestro trabajo.....	6
1.2.1 Objetivos principales.....	6
1.2.2 Objetivos secundarios o alternativos.....	7
1.2.3 Cultura empresarial.....	8
1.3 Enfoque y método seguido.....	8
1.4 Planificación del trabajo.....	9
2. Herramientas de desarrollo	10
2.1 Godot Engine 4.2.....	10
2.2 Nuestro lenguaje programación: GD Script.....	11
2.2 Blender 4.0.....	11
2.3 ¿Por qué utilizamos el Godot y Blender?.....	12
2.4 Lista de otras herramientas que hemos utilizado.....	12
2.5 Recursos que utilizamos.....	16
3. Descripción del nuestro juego	17
3.1 Tipo de juego.....	17
3.2 Jugabilidad y objetivos.....	17
3.3 Público objetivo.....	17
3.4 Conceptualización de ideas y diseños.....	17
3.4.1 Contexto y trama del juego.....	17
3.4.2 Los protagonistas y sus habilidades.....	18
3.4.2.1 Introducción al protagonista unificado elegido por los jugadores.....	19
3.4.3 Los mapas.....	19
3.4.4 La trama principal y las misiones secundarias.....	19
3.4.5 Los enemigos y los NPCs.....	20
3.4.6 El combate.....	20
3.4.7 El inventario y las transacciones monetarias.....	20
3.4.8 El multijugador en LAN.....	21
3.5 Anàlisis de requisitos.....	21
3.5.1 Requisitos funcionales.....	21
3.5.2 Requisitos no funcionales.....	21
4. La producción artística	22
4.1 Creación de personajes.....	22
4.1.1 FishMan.....	22
4.1.2 Extraterrestre Normal.....	23
4.1.3 Robot.....	24
4.2 Esqueleto y animaciones.....	25
4.3 Integración y exportación de múltiples animaciones.....	27
4.4 Importación de archivos de modelo en Godot.....	29

4.5 Escena.....	30
4.5.1 Nave espacial.....	30
4.5.2 Interior de la nave espacial.....	31
4.6 Icono.....	33
5. La creación del nuestro proyecto.....	35
5.1 Control e interacción con los personajes.....	35
5.2 Comportamiento hostil e inteligencia artificial.....	36
5.3 Diálogo de personajes.....	38
5.4 Menú UI.....	39
5.4.1 Menú de inicio.....	39
5.4.2 Menú de pausa.....	43
5.5 Sonido y efectos de sonido.....	43
5.5.1 Efectos de sonido.....	43
5.5.2 Integración de audio.....	44
5.5.3 Voz de los personajes.....	45
6. Proceso de desarrollo e implementación.....	47
6.1 Proceso de implementación en Godot.....	47
6.1.1 Editor de escenas.....	47
6.1.2 Escritura de Scripts.....	48
6.1.3 Editor de animación.....	49
6.1.4 Editor de audio.....	52
6.1.5 Editor de menu.....	55
6.1.6 Editor de terrano 3D.....	58
6.1.7 Editor de diálogos.....	59
6.1.8 Multijugador.....	60
6.1.9 Interactivo.....	62
6.1.10 Fondo del cielo mundial.....	64
6.1.11 Partículas de GPU.....	64
6.1.12 Carga automática.....	66
6.2 Proceso de implementación en Blender.....	68
6.2.1 Modelado.....	68
6.2.2 Esculpido.....	69
6.2.3 Sombreado.....	70
6.2.4 Modificador.....	70
7. Pruebas y ejecución.....	72
8. Sitio web y servidores.....	84
8.1 Nuestro servidor web (InfinityFree).....	84
8.2 Nuestro servidor de correo externo.....	87
8.3 Nuestro sitio web de WordPress.....	88
8.3.1 Plugin WP Mail SMTP Lite.....	89
8.3.2 Plugin WP Forms.....	90
8.3.3 Plugin Newsletter.....	91
8.3.4 Plugin All in One SEO.....	91
8.3.5 Plugin MonsterInsights.....	91

8.3.6 Plugin GTranslate.....	92
8.4 Nuestro servidor correo en redes locales.....	94
8.5 Lanzamiento y Promoción.....	95
9. Riesgos, desafíos y errores.....	96
9.1 Riesgos que se pueden encontrar en el proyecto.....	96
9.2 Errores encontrados y soluciones.....	96
9.2.1 Errores.....	96
9.2.2 Soluciones:.....	96
10. Conclusión.....	100
11. Bibliografía y referencias.....	101
Blender:.....	101
SketchFab:.....	101
Web y server:.....	102
Godot:.....	102
12. Anexos.....	103
[1] Narrador del inicio.....	103
[2] PLAN DE EMPRESA.....	104
[3] Nuestro web.....	104
[4] Servidor de correo en red local.....	105
Servidor.....	105
Cliente.....	109
[5] Web HTML.....	109

FICHA DE NUESTRO TRABAJO FINAL

Título	El Desastre Humano (en inglés Human Disaster o Human Digrate)
Autores	Wenjing Chen, Jiayi Chen.
Idioma del trabajo	Castellano
Palabras claves	Godot 4.2, Blender 4.0, 3D, Videojuego, Software Libre, 1ra y 3ra persona, Multijugador, Juego FPS
Resumen del proyecto	
<p>El proyecto se basa en el desarrollo de un pequeño juego 3D de mundo abierto, que narrará la historia de un protagonista que, tras ser capturado por extraterrestres y transformado con características de animales mutantes, lucha para derrotar a los alienígenas y salvar el mundo.</p> <p>En el desarrollo de este proyecto, las herramientas principales que utilizaremos son <i>Godot</i>, <i>Blender</i> y <i>Mixamo</i>, y realizaremos investigaciones sobre cómo utilizarlas a medida que avanzamos en el desarrollo. Además de su historia principal, contará con un modo multijugador en línea, y al mismo tiempo, intentaremos añadir nuestros propios modelos y otros elementos para mantener un cierto nivel de originalidad en el proyecto.</p> <p>Asimismo, este proyecto también incluye información sobre la creación de sitios web, servidores de correo electrónico y otros contenidos, todos con el propósito de servir al tema de nuestro proyecto.</p>	
Project abstract	
<p>The project is based on the development of a small open-world 3D game, which will narrate the story of a protagonist who, after being captured by aliens and transformed with the mutant animal features, fights to defeat the aliens and save the world.</p> <p>In the development of this project, the main tools we will use are <i>Godot</i>, <i>Blender</i>, and <i>Mixamo</i>, and we will conduct research on how to use them as we progress in the development. In addition to its main story, it will feature an online multiplayer mode, and at the same time, we will try to add our own models and other elements to maintain a certain level of originality in the project.</p> <p>Furthermore, this project also includes information about the creation of websites, email servers, and other contents, all with the purpose of serving the theme of our project.</p>	

1. Introducción

1.1 Contexto y justificación del nuestro trabajo

Inicialmente, nuestro proyecto planificaba la creación de un sitio web que permitiera a los usuarios realizar compras en línea de manera sencilla o integrar diversas herramientas útiles en un solo lugar. Sin embargo, nos dimos cuenta de que esto era solo una pequeña parte de un vasto universo de posibilidades.

Inspirados por el humor absurdo y la libertad creativa de juegos como "**Goat Simulator**" y "**Deer Simulator**", y apasionados por el mundo de los videojuegos, decidimos enfocarnos en el desarrollo de un juego. Queremos crear un juego de tipo humorístico y absurdo, **no solo un juego 2D común, sino un desafío ambicioso en el terreno de los juegos 3D**. Este juego incluirá elementos de rol, un pequeño mundo abierto, disparos, narrativa y juego multijugador.

Dada la complejidad del proyecto, esto **será muy desafiante**, ya que **nunca hemos utilizado herramientas como Godot y Blender**. Esto significa que nuestro equipo tendrá que aprender desde cero cómo usarlas y dedicar mucho esfuerzo en el proceso.

Además de la creación del juego, también planeamos utilizar un servidor Hosting Web para crear un sitio web propio para nuestro juego. Este sitio web **podrá comunicarse con Internet** y albergar enlaces para que los jugadores y usuarios descarguen nuestro juego.

1.2 Objetivos del nuestro trabajo

1.2.1 Objetivos principales

En este proyecto, nuestros objetivos principales son:

- **Aprender el uso de las herramientas Godot y Blender:** Esto incluye desde la escritura de guiones hasta la creación de modelos de personajes, diseño de entornos, edición de texturas, y animación de personajes. Adicionalmente, vamos a investigar a fondo funcionalidades complejas como la locomoción, la cámara, la detección de colisiones, los sistemas de armamento, los diálogos, y la interactividad.
- **Desarrollar un juego multijugador en LAN:** Planeamos crear un juego que ofrezca un modo multijugador en red local, permitiendo a los usuarios colaborar en la defensa contra una invasión extraterrestre.
- **Crear un sitio web y correo electrónico con nuestro propio dominio:** Empleando los servicios de hosting web de InfinityFree y el servidor de correo de Zoho Mail, pretendemos configurar un sitio web oficial y un correo

electrónico personalizado (por ejemplo: support@desahuman.wuaze.com) para comunicarnos efectivamente y promocionar nuestro trabajo.

- **Crear un servidor correo en redes locales:** Configuraremos un servidor de correo electrónico y servidor en redes locales adecuado para nuestro uso interno.
- **Producir un juego multiplataforma:** Nuestro propósito es aprovechar la capacidad de Godot para llevar el juego a diversas plataformas, comenzando con Windows y Linux y, en el futuro, expandirnos a Android, iOS, Mac, y más.
- **Ofrecer una versión de beta completa pero sencilla:** A pesar de la limitación de tiempo y que el juego será básico, nos comprometemos a entregar un producto que incluya los elementos esenciales del juego y que sea intuitivo para el usuario.
- **Proporcionar un juego accesible para los usuarios:** Nuestro enfoque es diseñar una experiencia de juego que se alinee con los controles estándar de los videojuegos actuales, facilitando la inmersión de los jugadores en la mecánica del juego y garantizando su disfrute.
- **Realizar un proyecto apasionante y motivador:** Esperamos que el cumplimiento de estos objetivos convierta este proyecto en una experiencia gratificante y llena de desafíos, que nos permita aprender y crecer tanto personal como profesionalmente.

1.2.2 Objetivos secundarios o alternativos

Si hay suficiente tiempo, también hacemos los siguientes objetivos:

- **Perfeccionar nuestro juego:** Como es nuestra primera experiencia con estas herramientas, es probable que el resultado inicial sea rudimentario, pero con tiempo suficiente, planeamos optimizarlo.
- **Enriquecer la narrativa de nuestro juego:** Las limitaciones de tiempo pueden afectar el desarrollo de una trama completa, sin embargo, pretendemos enriquecer y optimizar constantemente la historia, permitiendo que evolucione según las elecciones del jugador y conduzca a múltiples finales posibles.
- **Aumentar la cantidad de personajes y enemigos en el juego:** Nos gustaría incluir una mayor variedad de personajes y enemigos en nuestro juego.
- **Expandir nuestro mapa:** Nos esforzaremos por desarrollar un mapa de juego más grande y rico, ofreciendo a los jugadores un espacio más amplio para explorar.
- **Desarrollar un plan de marketing y promoción:** Si es posible, crearemos un plan de marketing y promoción, incluyendo publicidad en redes sociales,

establecimiento de asociaciones, participación en exposiciones de juegos, entre otros, para aumentar la visibilidad del juego y atraer a jugadores potenciales.

1.2.3 Cultura empresarial

Para este proyecto también hemos elaborado una planificación empresarial con vistas a su futura realización. El nombre de empresa que definimos para este proyecto es DesHuman (D&H), que contiene nuestra cultura de empresa: misión, visión y valores.

Nuestros valores fundamentales se basan principalmente en la **innovación**, la **calidad**, la **colaboración**, la **integridad**, el **aprendizaje** y la **inclusión**.

Estos valores son la base de nuestra misión de crear un pequeño juego 3D intuitivo de mundo abierto que no sólo entretenga sino que también proporcione a los jugadores una experiencia de videojuego 3D significativa y agradable. Además, nuestro equipo valora profundamente el establecimiento de un ambiente en el que todos se sientan escuchados y puedan aprender unos de otros, y donde reconocer y comprender los errores sea considerado una parte importante del proceso de aprendizaje.

Nuestra visión es brindar a los usuarios una experiencia de juego en 3D memorable y lograr un lugar destacado en el mercado de videojuegos en 3D. Anhelamos que nuestro juego se distinga por su calidad, innovación y profundidad, para perfeccionar y enriquecer continuamente la experiencia que ofrecemos a los jugadores. Además, como estudiantes y desarrolladores emergentes, nuestra visión incluye también el crecimiento y avance personal. Vemos el desarrollo de este juego como una oportunidad para aprender más sobre programación, diseño, narrativa y muchas otras facetas del desarrollo de videojuegos. Cada paso que damos es una oportunidad para mejorar nuestras habilidades. [\[Anexos\]\[2\]](#)

1.3 Enfoque y método seguido

La realización de este proyecto se enfoca tanto en la práctica como en la teoría. Nuestro objetivo es utilizar Godot Engine y Blender, junto con una serie de otras herramientas, para desarrollar nuestro proyecto de juego 3D desde cero, alcanzando al menos una versión beta para las pruebas. Vamos a aprovechar al máximo los recursos y la ayuda que ofrece internet para asegurarnos de cumplir con las necesidades específicas del proyecto. Mientras seguimos este camino, también vamos a analizar en profundidad los problemas y errores que encontramos en el proyecto, buscando soluciones para estos. Esto servirá como base para nuestro continuo progreso, ayudándonos a entender y optimizar nuestro proceso de trabajo por completo.

Además, somos conscientes de que, en el futuro, el proyecto necesitará más optimizaciones y mejoras.

1.4 Planificación del trabajo

El Diagrama de Gantt es una herramienta clave en la gestión y planificación de nuestro proyecto. Este diagrama nos permite visualizar el cronograma del proyecto, incluyendo todas las fases de desarrollo, desde la concepción inicial hasta el lanzamiento final.

A continuación, se discuten los detalles presentados en gráfico incluidos en esta sección:



2. Herramientas de desarrollo



2.1 Godot Engine 4.2

Godot es un motor de juegos de código abierto y **multiplataforma, potente y altamente flexible**, que soporta el desarrollo de juegos en **2D y 3D** y puede ejecutarse en diversos sistemas operativos como **Windows, macOS y Linux**. Sus funcionalidades avanzadas y su facilidad de uso permiten a los desarrolladores crear juegos visuales y fluidos que pueden ser fácilmente desplegados en múltiples plataformas como **PC, Android, iOS y HTML5**, entre otros.

En lo que se refiere a los lenguajes de programación, Godot ofrece varias opciones como **GD Script, C++ o C#**, así como la capacidad de integrar otros lenguajes mediante **GD Native**, como **C o C++**. La comunidad también ofrece soporte para otros lenguajes, como **Rust, Nim, JavaScript, Haskell, Clojure, Swift y D**, ofreciendo a los desarrolladores una gran flexibilidad.

El **motor gráfico 2D y 3D** de Godot funciona de forma independiente y permite la combinación de contenido 2D y 3D mediante nodos de vista. Además ofrece un **lenguaje de shaders** personalizado similar a **GLSL** para la creación de materiales y el renderizado, y proporciona herramientas de renderizado visual para simplificar el proceso.

En cuanto a animación, Godot dispone de un potente sistema de animación que permite a los desarrolladores crear **animaciones de huesos, mezclas, árboles de animación, animaciones de transición** en tiempo real, entre otros. Prácticamente todas las variables de los objetos del juego pueden ser animadas, añadiendo dinamismo e interactividad al juego.

En cuanto al tratamiento de colisiones, Godot permite la creación de diferentes **estructuras de colisión** para distintos objetos, utilizando formas **primitivas o convexas** para representarlas, satisfaciendo así las necesidades de colisión de cada objeto.

Además de las funcionalidades mencionadas, Godot incluye la creación de terrenos, efectos de luz y sombra, sistemas de partículas, reproducción de audio, procesamiento de texturas, diseño de interfaces gráficas y soporte para diferentes periféricos, ofreciendo a los desarrolladores una solución integral para el desarrollo de juegos. Tanto los principiantes como los desarrolladores experimentados pueden encontrar en Godot las herramientas adecuadas para crear obras maestras de juegos.

Los plugins:

En Godot es posible añadir diferentes plugins para completar nuestro juego, entre los que hemos añadido **Terrain3D**, un sistema de terreno editable que podemos utilizar para generar terreno, y el plugin cliente **nakama**, que nos permite conectarnos a un servidor y luego jugar partidas multijugador a través de una serie de configuraciones.

2.2 Nuestro lenguaje programación: GD Script

En nuestro proyecto, para la escritura de scripts de juegos, hemos escogido el lenguaje de programación GDScript que viene integrado en el entorno de Godot. GDScript es un lenguaje de alto nivel, imperativo y orientado a objetos, cuya sintaxis es similar a la de Python, lo que lo hace fácil de aprender y usar. Esta lengua ha sido específicamente diseñada para integrarse con el entorno de Godot y ofrece una experiencia de desarrollo de juegos eficiente.

La razón principal para elegir GDScript es su alta integración con el entorno de Godot. Esta integración nos permite operar de forma fluida y optimizar el rendimiento, facilitándonos un desarrollo de juegos más intuitivo y eficiente. Como GDScript está hecho a medida para Godot, aprovecha al máximo las características y optimizaciones del entorno, aumentando la eficiencia en el desarrollo y rendimiento del juego.

2.2 Blender 4.0



En este proyecto, la herramienta de modelado 3D que usamos es Blender. Blender es un software de gráficos por computadora en 3D, libre y de código abierto, que tiene una amplia gama de funciones, desde modelado y animación hasta renderizado e incluso edición de video. Blender es completamente gratuito y es adecuado tanto para estudios profesionales como para principiantes que empiezan desde cero. A continuación, explicaremos detalladamente algunas utilidades de Blender:

- **En cuanto al modelado**, Blender nos ofrece una variedad de herramientas de edición de mallas, como extrusión, corte, subdivisión y fusión, que nos permiten crear modelos 3D complejos. Además del modelado de mallas tradicional, los usuarios también pueden modelar utilizando curvas y superficies NURBS. El modo de escultura de Blender es perfecto para crear detalles y texturas, y ofrece una amplia biblioteca de pinceles. Además, los modificadores del sistema en Blender pueden ayudarnos a automatizar tareas comunes de modelado, como subdivisión de superficies, espejo y array, entre otros.
- **En cuanto a la rigidez de los huesos**, Blender ofrece potentes herramientas para el rigging, que es el proceso de vincular un esqueleto a un modelo 3D para animarlo. Este proceso implica asignar los vértices del modelo a los huesos del esqueleto, lo que permite que el modelo se deforme de manera dinámica a medida que el esqueleto se mueve.
- **En el área de animación**, Blender permite controlar el movimiento de los objetos mediante fotogramas clave y realiza interpolaciones automáticamente. La funcionalidad de Animación No Lineal (NLA) permite repetir y combinar segmentos de animación. Dispone de potentes herramientas de rigging, incluyendo kinematics hacia adelante y hacia atrás.
- **Para el renderizado**, Cycles de Blender es un motor de renderizado de trazado de rayos imparcial, que proporciona efectos de iluminación realistas.

Eevee es un motor de renderizado en tiempo real, ideal para previsualizaciones rápidas y animaciones. Las capas y los pases de renderizado ofrecen un mejor control y facilitan la composición posterior.

- **En materiales y texturas**, la función de despliegue UV permite desplegar la superficie de un modelo 3D en un plano 2D para un mapeado preciso. Los nodos de sombreado se utilizan para crear materiales complejos que simulan diferentes efectos de superficie.
- **En términos de administración y extensibilidad**, Blender tiene buenas funciones de gestión de archivos, con soporte para enlazar y reutilizar bibliotecas. Su capacidad de extensión es muy alta, ya que las funciones pueden ampliarse con scripts en Python y hay una gran cantidad de complementos y recursos de la comunidad disponibles.
- **La interfaz de usuario es personalizable**, lo que permite a los usuarios ajustar la disposición de la interfaz y los atajos de teclado según su flujo de trabajo. Además, su potente API permite a los desarrolladores de terceros crear nuevas herramientas y complementos.

2.3 ¿Por qué utilizamos el Godot y Blender?

La razón por la que elegimos Godot y Blender como herramientas para nuestro proyecto es principalmente porque ambas son herramientas de código abierto y gratuitas. Esto significa que podemos usar, modificar y redistribuir los juegos y modelos que creamos con estas herramientas sin preocuparnos por problemas de derechos de autor.

Para nosotros, es muy importante tener la propiedad total y el control de nuestras creaciones, ya que podemos usarlas libremente en nuestros portafolios personales o en futuros proyectos. Además, tanto Godot como Blender son gratuitos, lo cual es una gran ventaja para nosotros, ya que tenemos un presupuesto limitado.

La fuerte comunidad de soporte y la abundancia de recursos de tutoriales en línea también nos han ayudado mucho a resolver problemas y aprender nuevas habilidades.

2.4 Lista de otras herramientas que hemos utilizado

1 - Mixamo



En nuestro proyecto elegimos utilizar Mixamo para manipular huesos y aplicar movimiento de personajes. Mixamo es una plataforma de servicios de animación basada en la nube que ofrece una serie de recursos de animación de alta calidad, incluyendo acciones de personajes y modelos de personajes. Mediante Mixamo, podemos añadir rápida y fácilmente animaciones a nuestros personajes.

.....

2 - Photopea



Es un editor gráfico cuyas funciones son básicamente las mismas que Photoshop, pero es gratuito y se puede editar online en la Web, lo que lo convierte en un excelente sustituto de Photoshop. Se utiliza para editar imágenes, realizar ilustraciones, diseñar páginas web o convertir entre diferentes formatos de imágenes.

3 - InfinityFree



InfinityFree es un servicio de alojamiento web gratuito que nos proporciona espacio online gratuito para alojar nuestro sitio web. Aunque es gratuito, ofrece muchas de las funciones que se encuentran comúnmente en los planes de alojamiento pagos, como almacenamiento ilimitado, ancho de banda ilimitado y soporte para múltiples dominios.

Las características principales incluidas en el plan gratuito son:

- Alojamiento ilimitado
 - Nombres de subdominios gratuitos
 - Ancho de banda ilimitado
 - Soporte para MySQL y PHP
 - Servicio gratuito de DNS (MX, CNAME, SPF)
 - Certificados SSL gratuitos
 - Instalador Softaculous
-

4 - Wordpress



WordPress es un sistema de gestión de contenidos gratuito y de código abierto que está basado en PHP y MySQL. Nos permite crear y mantener de manera fácil y eficiente nuestros sitios web de juegos. Además, tiene una versión .org que es de uso libre y sin coste.

En WordPress utilizamos una serie de plugins que nos ayudan a construir nuestro sitio web, la siguiente es la lista:

-Action Scheduler: Una biblioteca de programación sólida para usar en plugins de WordPress.

-Advanced Cron Manager: Ver, pausar, eliminar, editar y añadir eventos WP Cron.

-All in One SEO: Optimiza nuestro sitio web para motores de búsqueda de manera sencilla.

-Google Analytics para WordPress por Monster Insights: Integra Google Analytics en nuestro sitio web.

-GTranslate: Traduce nuestro web y hazla multilingüe.

-Newsletter y sus addons: Sirve para crear y enviar boletines de noticias a nuestros suscriptores.

-WP Mail SMTP: Mejora la entrega de correos electrónicos desde nuestro sitio web.

-WP Forms Lite: Sirve para crear formularios de contacto y encuestas.

5 - Zoho Mail, Brevo, Gmail



Zoho Mail, Brevo y Gmail son partes importantes de nuestro servidor de correo. Zoho Mail es nuestro servidor de correo principal, lo usamos porque nos permite comunicarnos con Internet, además nos permite tener correos electrónicos con nuestro propio dominio.

- **Zoho Mail:** Es un servicio de correo electrónico para negocios y profesionales. Zoho Mail es que nos deja usar nuestro propio dominio para el correo, lo que nos hace ver más profesionales.
- **Brevo:** Es una plataforma de marketing en línea todo en uno que nos ofrece un montón de herramientas como el marketing por correo electrónico, la automatización, anuncios por mensajes de texto, un CRM y funciones de chat. Lo que más hacemos con Brevo es usar su API para conectarlo con WP Mail en Wordpress.
- **Gmail:** Este es un servicio de correo electrónico gratuito que ofrece Google. Es súper popular y mucha gente lo usa en todo el mundo. Nosotros usamos una cuenta de Gmail para registrar nuestro Zoho Mail.

6 - Productos de Google



El servicio de **Google Drive** integrado en nuestro Gmail, proporcionado por el instituto, nos ayuda a guardar y hacer copias de seguridad de nuestros proyectos. Además, también tenemos un disco duro portátil para almacenar datos, siguiendo la regla 3-2-1.

Para nuestro sitio web, también usamos algunas herramientas gratuitas que nos da Google:

El **Google Search Console** nos ayuda a que la gente pueda encontrar nuestro sitio web cuando buscan cosas en Google.

El **Google Analytics** nos permite ver cuánta gente viene a nuestro sitio web y qué hacen allí. Nos da muchos datos interesantes sobre quién nos visita y cómo nos visita, lo que nos ayuda a entender mejor a nuestra audiencia y a mejorar nuestro sitio.

7 - VirtualBox



VirtualBox es una aplicación gratuita y de código abierto para crear máquinas virtuales que instalan sistemas operativos.

En este proyecto, usaremos Virtualbox para construir nuestros propios servidores, como nuestro propio servidor de correo interno con postfix, servidor de juegos nakama LAN, etc.

8 - Nakama



Nakama es un marco de servidor de juegos de código abierto que se puede utilizar para crear juegos multijugador en línea en Godot. Entonces podemos usarlo para construir un servidor en multijugador para nuestro juego.

9 - PostgreSQL, yaml



Para guardar la configuración de nuestro juego multijugador, utilizamos un archivo yaml para llamar a una base de datos PostgreSQL para guardar los datos de nuestro juego multijugador.

10 - Bind9, Dovecot, Postfix, MailUtils, Apache2, MySQL, Thunderbird, Roundcube

Bind9 es un servidor de DNS que se utiliza para traducir nombres de dominio en direcciones IP y viceversa.

Dovecot es un servidor de correo electrónico que nos permite gestionar y acceder a nuestros correos electrónicos de manera segura.

Postfix es un programa de transferencia de correo que se encarga de enviar y recibir correos electrónicos.

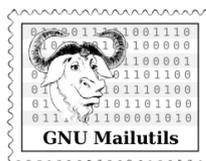
MailUtils es una colección de utilidades de correo electrónico que proporciona funciones adicionales para el manejo de correos electrónicos.

Apache2 es un servidor web que nos permite alojar y servir nuestro sitio web localmente.

MySQL es un sistema de gestión de bases de datos relacional que se utiliza para almacenar y recuperar datos de manera eficiente.

Thunderbird es un cliente de correo electrónico que utilizamos para acceder y gestionar nuestros correos electrónicos en el servidor.

Roundcube es una interfaz web de correo electrónico que ofrece una experiencia de usuario similar a la de un cliente de correo tradicional, pero a través de un navegador web.



11 - IsardVDI



IsardVDI es una solución de virtualización de escritorio que utilizamos para crear y gestionar máquinas virtuales en nuestro entorno. Podemos configurar y administrar las máquinas virtuales que utilizamos para nuestros servidores web y de correo electrónico con IsardVDI.

13 - Otras herramientas no tan utilizadas

- **Planner** - Hemos usado Planner para hacer nuestro Diagrama de Gantt en el proyecto. Es como un planificador que ayuda a organizar las tareas y los tiempos.
- **Textstudio** - Con Textstudio, conseguimos los estilos de texto para los íconos de nuestro juego, en formato PNG transparente, lo que nos permite usarlos en Photopea para diseñar íconos con títulos.
- **llovepdf** - llovepdf nos ha servido para cambiar el tamaño de las imágenes y comprimir el tamaño de los documentos, es muy útil y fácil de usar.
- **Logo.com** - En Logo.com, creamos de manera rápida y sencilla el logo de la empresa que planeamos en el plan de empresa.
- **Chatgpt** - A veces, también usamos Chatgpt para obtener inspiración y hacer preguntas sobre el proyecto. Nos da ideas y responde dudas que tenemos.
- **Vidnoz AI** - es una herramienta que utilizamos para convertir texto en voz. Nos permite crear archivos de audio en formato WAV con diferentes voces.

2.5 Recursos que utilizamos

Además de utilizar una serie de herramientas, también recurrimos a algunos recursos en Internet, o podríamos decir que son nuestros proveedores. Aquí está la lista:

- **Instituto Puig Castellar:** Agradecemos a Puig Castellar por proporcionarnos recursos de hardware, lo que nos permitió usar equipos en el instituto para trabajar en el proyecto.
- **Sketchfab, TurboSquid, CGtrader:** Estos son sitios web de modelos 3D. Algunos de los recursos de nuestro proyecto de juego provienen de aquí. Considerando los problemas de derechos de autor, descargamos solo modelos con licencia CC Attribution.
- **Pixabay:** Usamos Pixabay para buscar efectos de sonido y música para nuestro juego. El contenido es de uso gratuito, no requiere atribución al autor y se puede modificar o adaptar para crear nuevas obras.
- **YouTube:** En YouTube buscamos muchísimos tutoriales, porque no sabíamos usar Godot y Blender. Los enlaces específicos los hemos puesto en el punto XX.
- **BiliBili:** Además de ver tutoriales en YouTube, también buscamos tutoriales en BiliBili, una plataforma china.
- **Kenney:** Utilizamos los recursos gratuitos para crear pequeños escenarios de juego. Además tiene licencia de CC0.
- **Onlygfx, Pngall:** Hemos utilizado imágenes de Onlygfx y Pngall para crear íconos del nuestro juego.
- **Github:** Podemos encontrar los recursos, herramientas y plugins que necesitamos en GitHub.
- **FootageCrate:** Usamos FootageCrate para obtener algunos recursos de efectos para nuestros proyectos y está libre de regalías.

3. Descripción del nuestro juego

3.1 Tipo de juego

El Desastre Humano es un pequeño juego de mundo abierto tipo RPG con un estilo de humor absurdo. Los bugs que aparecen en el juego pueden considerarse como parte de la experiencia cómica que ofrecemos a los usuarios, haciéndolos reír y desesperarse al mismo tiempo. Aquí, los jugadores podrán explorar libremente varios escenarios, desde las calles de la ciudad hasta el interior de naves espaciales alienígenas. El entorno interactivo permite a los jugadores manipular mecanismos, resolver acertijos y hablar con personajes no jugables (NPC), y cada acción influye directamente en la experiencia del juego y el desarrollo de la historia.

Nuestro juego admitirá la ejecución en los sistemas operativos **Windows** y **Linux**.

3.2 Jugabilidad y objetivos

La jugabilidad inicial se centra en la exploración y la resolución de misiones. Los jugadores avanzarán en la narrativa completando tareas como encontrar objetos específicos, resolver acertijos y luchar contra enemigos. Planeamos implementar un sistema de evolución de personajes, donde los jugadores podrán subir de nivel con puntos de experiencia, desbloqueando habilidades para enfrentar desafíos más complejos en el futuro.

El objetivo principal del juego es completar la misión principal, derrotar a los extraterrestres y salvar el mundo, con la esperanza de que el protagonista recupere su forma humana. A medida que avanza el juego, los jugadores descubrirán más información sobre la invasión extraterrestre y desbloquearán nuevas áreas y funciones, aumentando la profundidad y complejidad del juego.

3.3 Público objetivo

Nuestro público objetivo son los jugadores que disfrutan de juegos de mundo abierto con humor. El juego está diseñado para atraer a jugadores de todas las edades, especialmente a aquellos que les gusta la exploración profunda y el rol. Además, la trama y el desarrollo de los personajes atraerán a quienes prefieren juegos con historias envolventes y acertijos desafiantes.

3.4 Conceptualización de ideas y diseños

3.4.1 Contexto y trama del juego

Nuestra idea básica para la trama es la siguiente: en 1993, los humanos desarrollaron la señal de onda mostaza e intentaron enviarla al espacio exterior. Esto fue un gran avance tecnológico, pero los humanos no se dieron cuenta de que al hacer esto, revelarían la ubicación de la Tierra. Como resultado, la Tierra comenzó a ser invadida por criaturas extraterrestres.

Nuestro protagonista es una persona común. Un día, un extraterrestre llega a la Tierra y secuestra a personas para investigarlas. El protagonista es capturado

accidentalmente y se convierte en una de las víctimas de un caso de desaparición masiva. (de miles personas)

Los extraterrestres realizan experimentos biológicos en mil personas y al final sólo sobreviven cinco. Estos cinco sobrevivientes son los protagonistas que podemos elegir, y cada uno representa un animal diferente: pez, gato, gallina, perro y conejo.

Cuando el protagonista despierta, descubre que se ha convertido en un animal. "Qué animal depende del personaje que elija el jugador como protagonista". Al elegir un protagonista, los otros cuatro se convierten en enemigos del protagonista principal, estando bajo control mental. El protagonista debe derrotarlos (como misiones secundarias) para liberarlos del control mental y reclutarlos para su equipo en la batalla final contra el jefe. Si el jugador no completa estas misiones secundarias, enfrentará al jefe final sin la ayuda de estos aliados, luchando solo.

3.4.2 Los protagonistas y sus habilidades

En realidad, nuestra idea inicial era tener dos protagonistas en la versión beta: uno sería FishMan, un hombre mutante con características de pez; y el otro sería CatLady, una mujer mutante con características de gato. Pero lamentablemente aún no hemos creado CatLady.

Nuestro juego incluye cinco protagonistas. Esto significa que los jugadores pueden elegir a uno de estos cinco personajes como su protagonista principal en el juego. Todos estos personajes fueron secuestrados por extraterrestres y sobrevivieron a los experimentos, resultando en mutantes con rasgos animales. Y los cinco protagonistas tienen diferentes líneas argumentales iniciales. En la actualidad, solo hemos escrito dos líneas argumentales iniciales diferentes sobre FishMan y CatLady. Para más detalles, consulte [\[Anexos\]\[1\]](#).

Dado que los cinco protagonistas tienen características de diferentes animales y son los sobrevivientes de los experimentos, cada uno tiene sus propios superpoderes, es decir, habilidades especiales. Por ejemplo, el protagonista FishMan puede respirar indefinidamente bajo el agua o caminar sobre ella. Además de sus habilidades especiales, cada protagonista también tiene debilidades. Por ejemplo, la protagonista CatLady tiene miedo al agua y no puede bucear sin obtener un objeto específico de una misión secundaria.

No solo esto, al principio, también pensamos en que estos personajes tuvieran etapas de evolución, es decir, niveles (aunque esto no se implementó en la versión beta debido a la falta de recursos, habilidades y tiempo).

Tomemos como ejemplo a nuestro protagonista FishMan: al principio, FishMan no tiene un cuerpo humano y es solo un pez normal, moviéndose en un tanque de agua. Los jugadores deben controlar el tanque de agua, asegurándose de que no se rompa, mientras resuelven acertijos y escapan. Al alcanzar los objetivos de la misión, FishMan comenzará a evolucionar, desarrollando gradualmente una forma humana.

Esto es lo que llamamos evolución de la primera a la segunda etapa. Cuanto más fuerte se vuelva el protagonista, más completa será su forma humana, hasta que ya

no pueda evolucionar más (como el aspecto de icono del juego). A medida que evoluciona, el protagonista podrá hacer más cosas, como usar armas de fuego. Pero debido a las dificultades y nuestras limitaciones, sólo pudimos crear una versión beta muy básica, que no incluye la evolución ni las habilidades especiales. Por ahora, el protagonista solo puede disparar armas.

3.4.2.1 Introducción al protagonista unificado elegido por los jugadores

El/La protagonista era originalmente un ser humano común, hasta que un día, por infortunio, fue secuestrado por extraterrestres. Durante los experimentos ajenos, le inyectaron un extraño elixir que provocó cambios sorprendentes en su cuerpo. Ahora, se ha convertido en un monstruo/mutante mitad humano, mitad animal, pero en su corazón todavía mantiene el coraje y el sentido de la justicia humanos.

Durante los días de cautiverio en los extraterrestres, el/la protagonista vio de primera mano el terrorífico poder de la tecnología alienígena y los experimentos despiadados que realizaban con los humanos. Esto despertó su ira hacia los extraterrestres y su deseo de salvar al mundo. Decidió levantarse en resistencia, enfrentándose en una lucha encarnizada contra los ajenos por defender la Tierra y todos los humanos cautivos.

3.4.3 Los mapas

Principalmente, hemos concebido tres escenarios principales.

El primero es la nave espacial donde despierta el protagonista (actualmente aún no está desarrollada por completo, solo tenemos un laboratorio y un pasillo simples). Esta es una nave espacial de una base de los extraterrestres.

El segundo escenario es el mundo original del protagonista, la Tierra.

El tercero es la nave espacial principal de los extraterrestres.

3.4.4 La trama principal y las misiones secundarias

La trama principal del juego es bastante fija y los jugadores no pueden desviarse demasiado de ella. A continuación, presentamos un borrador básico de nuestra trama principal:

1. **Trama inicial:** El protagonista es capturado y llevado a la nave espacial alienígena. Aquí, el jugador debe luchar, resolver acertijos, evolucionar y escapar. Las tramas iniciales de los cinco protagonistas aquí antes de ser capturados por extraterrestres son todas diferentes. Por ejemplo, FishMan será un trabajador y CatLady será una pianista. Los detalles del borrador se pueden encontrar en el [Anexo].
2. **Trama intermedia:** El protagonista regresa a la Tierra y comienza su camino hacia el fortalecimiento. En esta etapa, busca ayuda, resuelve una serie de problemas y lucha contra otros sujetos experimentales capturados y modificados, es decir, los otros cuatro protagonistas. Finalmente, el protagonista encuentra una oportunidad para llegar a la nave espacial principal de los extraterrestres.

3. **Trama final:** El protagonista encuentra la manera de llegar a la nave espacial principal y lucha contra el jefe final en una gran batalla. Al final, el protagonista gana, salva el mundo y obtiene pistas sobre cómo volver a ser un humano normal.

Misiones secundarias:

Además de la misión principal, nuestro juego cuenta con muchas misiones secundarias. Estas misiones secundarias son en tiempo real, lo que significa que si el jugador avanza en la misión principal sin completar las secundarias, las perderá para siempre.

Cada misión secundaria incluye una trama, que puede ser independiente o estar relacionada con otras. A diferencia de las misiones principales, los jugadores pueden decidir la dirección de las misiones secundarias. Esto significa que, dependiendo de las elecciones del jugador, la trama secundaria se desarrollará de manera diferente y afectará la trama principal en distintos grados.

Sin embargo, debido a problemas de tiempo y complejidad, nuestra versión beta no incluye la línea principal ni los ramales por el momento.

3.4.5 Los enemigos y los NPCs

Tenemos planeado incluir muchos tipos de enemigos, como extraterrestres, robots, soldados de fuerzas especiales, policías, y más. Estos enemigos aparecerán en la misión principal, misiones secundarias y en el mundo abierto.

Los NPC también serán variados y tendrán muchos roles diferentes.

3.4.6 El combate

Hemos establecido que, mientras el protagonista aún no haya evolucionado a una forma humana, no podrá combatir y solo podrá resolver acertijos para enfrentarse a los enemigos. Solo después de completar esa etapa podrá usar armas para atacar. Sin embargo, debido a la complejidad, en nuestra versión beta el protagonista puede atacar directamente con armas, similar a un juego de disparos en primera o tercera persona.

3.4.7 El inventario y las transacciones monetarias

Dado que hay misiones, nuestro juego naturalmente tendrá un sistema de inventario. El inventario se utiliza principalmente para almacenar objetos, armas y algunos objetos clave de las misiones, lo cual es útil durante la resolución de acertijos.

Además, el juego también tendrá transacciones monetarias, donde los jugadores podrán comerciar con NPCs para comprar cosas. Sin embargo, en nuestra versión beta, esta característica aún no está implementada.

3.4.8 El multijugador en LAN

Nuestro juego cuenta con un modo multijugador en red local, que por ahora no tiene ninguna trama y solo permite a los jugadores luchar juntos contra enemigos. Este modo todavía está en desarrollo. En nuestra versión beta solo habrá un esbozo básico del modo multijugador, sin mucho contenido adicional por el momento.

3.5 Análisis de requisitos

- Conocimientos de Godot
- Conocimientos de Blender
- Conocimientos sobre videojuegos
- Conocimientos de Photopea
- Conocimientos para crear servidores multijugador con Nakama
- Conocimientos para crear servidores de correo externos e internos
- Conocimientos para usar y configurar servicios de alojamiento web
- Conocimientos para crear y configurar sitios web con WordPress

3.5.1 Requisitos funcionales

- Al menos un área de mapa para que los jugadores exploren.
- Al menos un sistema de combate que permita a los jugadores enfrentarse a enemigos.
- Los personajes del juego deben poder moverse y tener controles básicos.
- Al menos un modo multijugador.
- Debe ser jugable en Windows y Linux.
- El juego debe tener una interfaz de usuario para el menú principal y la configuración.
- El juego debe ser jugable sin problemas.
- Debe haber enemigos en el juego.
- Debe haber diálogos en el juego.
- El juego debe tener opciones de primera y tercera persona.
- Movimiento agradable para el jugador.

3.5.2 Requisitos no funcionales

- Un sistema de nivelación de personajes.
- Un sistema de misiones y objetivos que guíe a los jugadores a través de la historia.
- Compatibilidad con todos los sistemas operativos.
- Optimización y mejora del juego.
- Añadir más misiones y narrativa.
- Un sistema de inventario y transacciones monetarias.
- Optimización de las perspectivas en primera y tercera persona.
- Mejora la sensación de control.
- Optimización de la tasa de fotogramas.
- Añadir más personajes y enemigos.
- Expansión de más escenarios.

4. La producción artística

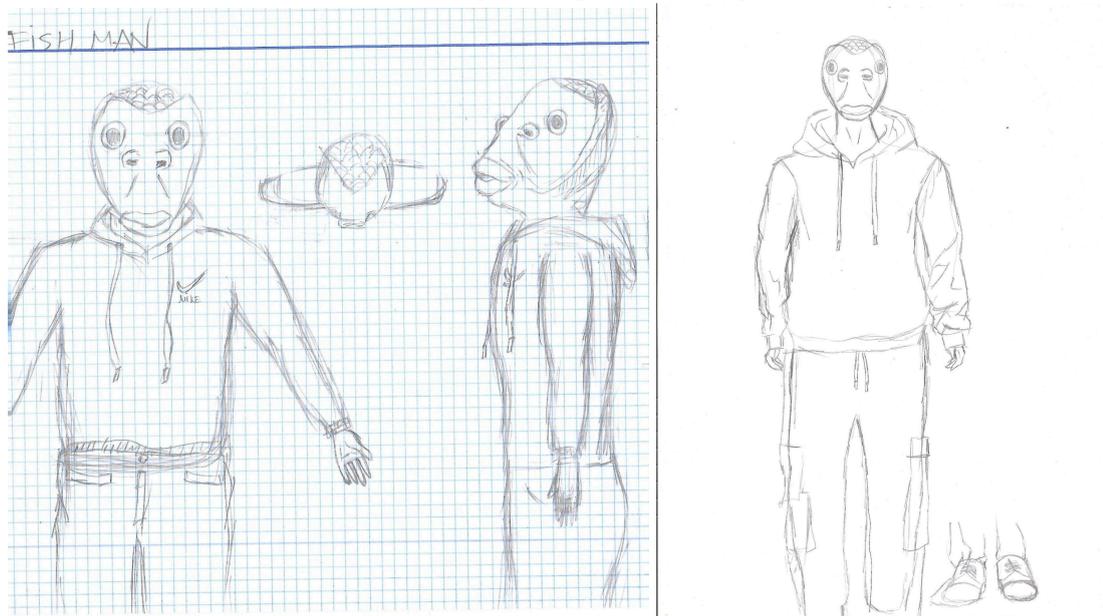
4.1 Creación de personajes

4.1.1 FishMan

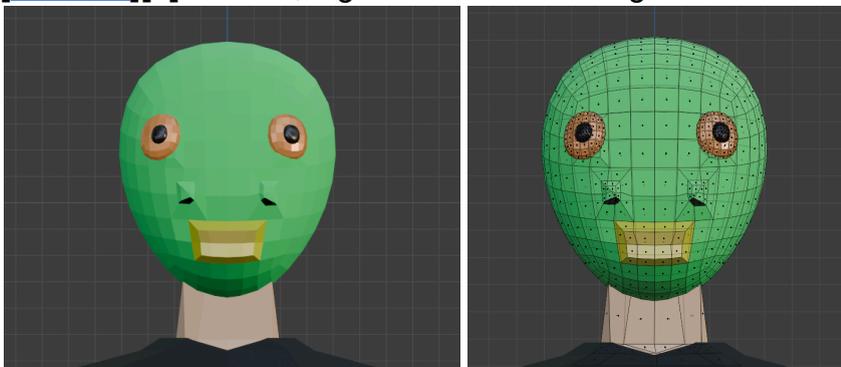
Al principio, no teníamos mucha inspiración para diseñar al protagonista, hasta que un día, mientras veíamos videos, encontramos una máscara de cabeza de pez. Era una máscara extraña y fea, pero pensamos que podría ser perfecta para uno de los protagonistas de nuestro juego, que tiene un estilo absurdo.



Así que empezamos a dibujar bocetos para uno de nuestros protagonistas, FishMan. Después de varios dibujos, llegamos al siguiente boceto simple:



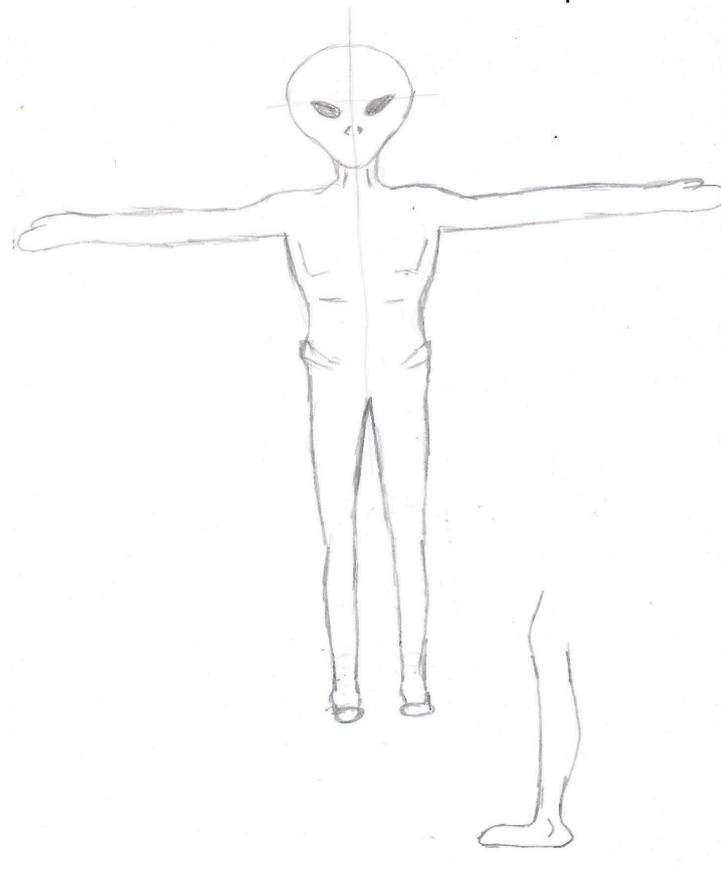
Con el boceto listo, pudimos usarlo como referencia para crear el modelo 3D. Como no teníamos experiencia previa, estuvimos viendo tutoriales mientras modelábamos [\[Blender\]](#)^[1]. Al final, logramos obtener el siguiente resultado simple en Blender:



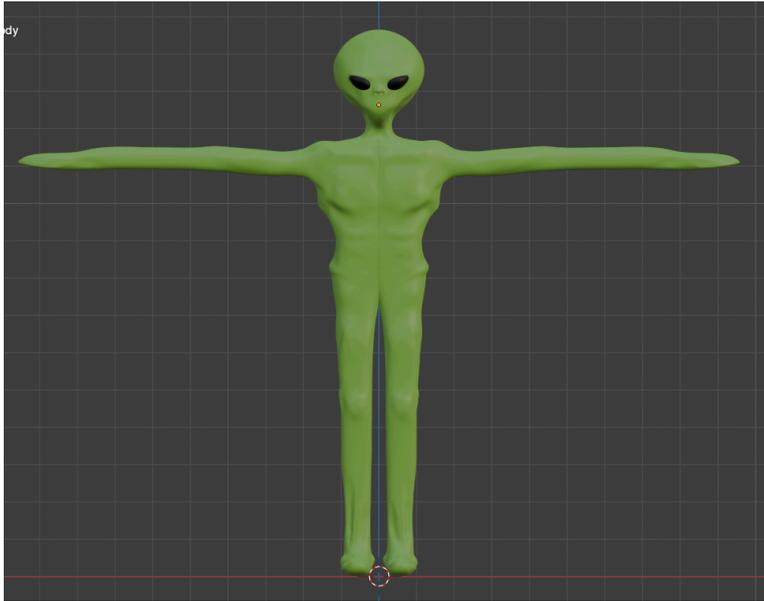


4.1.2 Extraterrestre Normal

Hoy en día, nuestra imagen de los extraterrestres suele ser la de criaturas con piel verde y grandes ojos negros. Decidimos seguir este diseño clásico para nuestro extraterrestre y para este modelo utilizamos el modo escultura para darle un aspecto más detallado. Nos basamos en un tutorial de Blender sobre cómo crear extraterrestres [\[Blender\]\[2\]](#) y también dibujamos un boceto muy simple. Este boceto nos sirvió como referencia para el modelado 3D.

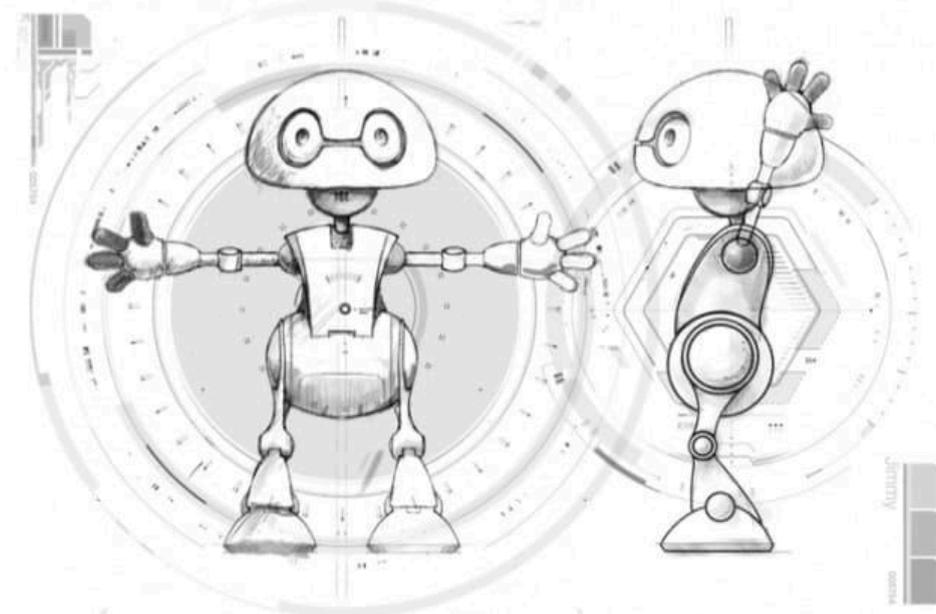


El resultado final del modelado en Blender es el siguiente:

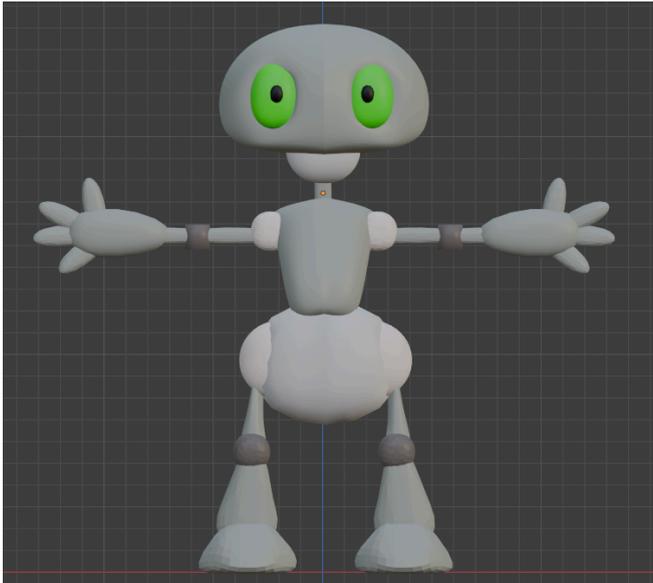


4.1.3 Robot

Para el Robot, no elegimos dibujar un boceto nosotros mismos, sino que optamos por obtener una imagen de referencia sobre robots de internet[Blender][3]. Y esto nos sirvió como nuestro primer experimento al usar Blender para modelar.



El resultado final del modelado en Blender es aproximadamente el siguiente:

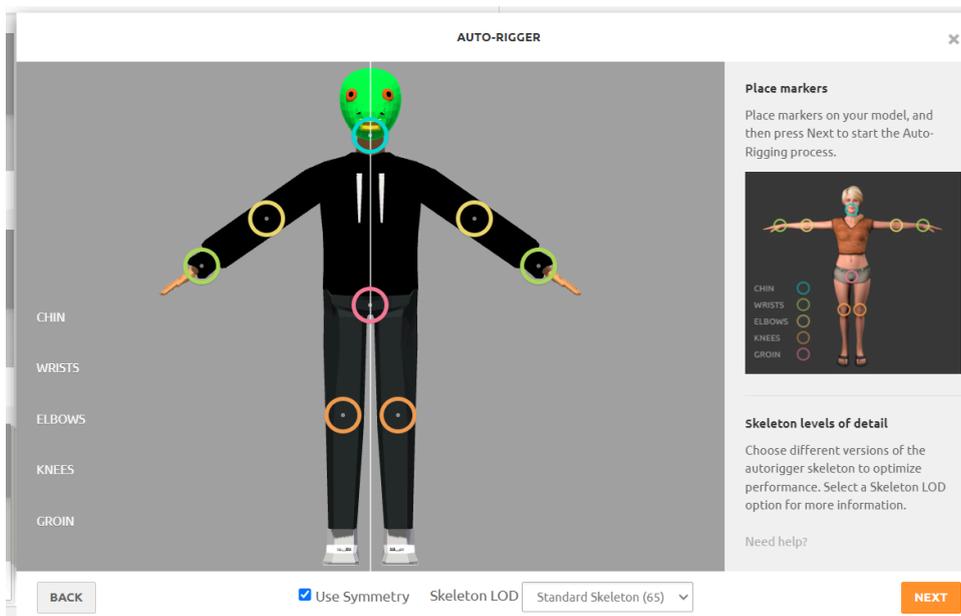


4.2 Esqueleto y animaciones

Cuando terminamos de modelar, es hora de usar la herramienta Mixamo, que nos ayuda a enlazar rápidamente los huesos e incluso aplicar animaciones de movimiento. En Mixamo, se pueden descargar animaciones en formato fbx, aunque solo de una en una.

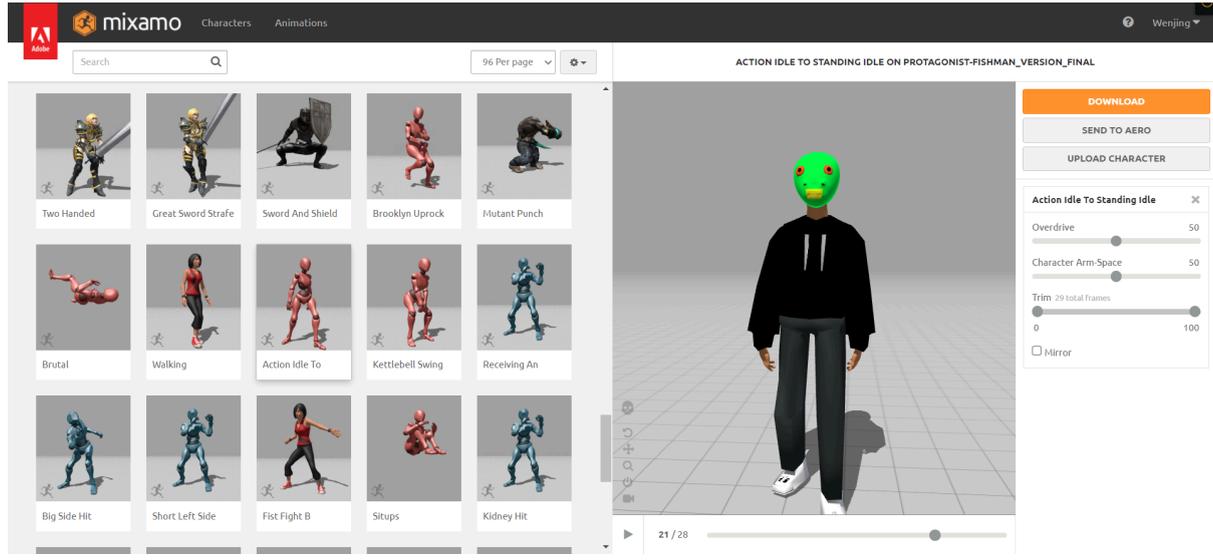
El uso de Mixamo es muy sencillo e intuitivo, no se necesita experiencia previa en la creación de animaciones. Solo necesitamos subir el modelo del personaje y seleccionar las animaciones adecuadas de la biblioteca de Mixamo para aplicarlas fácilmente a nuestro personaje. Esto nos permite ahorrar mucho tiempo y esfuerzo, permitiéndonos enfocarnos en otros aspectos del proyecto.

Durante el proceso de subir nuestro modelo, es necesario especificar el mentón, las muñecas, los codos, las rodillas y la ingle del modelo para que el esqueleto se enlace correctamente.



Una vez que el esqueleto está enlazado, podemos elegir cualquier animación de la biblioteca. Después de seleccionar la animación deseada, solo necesitamos hacer clic en "Descargar" en la esquina superior derecha, y obtendremos un archivo en formato fbx que incluye la animación y la textura del modelo.

Sin embargo, hay un pequeño inconveniente: las animaciones se descargan de una en una. Si necesitamos múltiples animaciones, debemos descargar cada una por separado y luego integrarlas en un solo archivo. También es posible descargar un paquete completo, donde las animaciones ya están integradas pero aún es necesario fusionarlo.

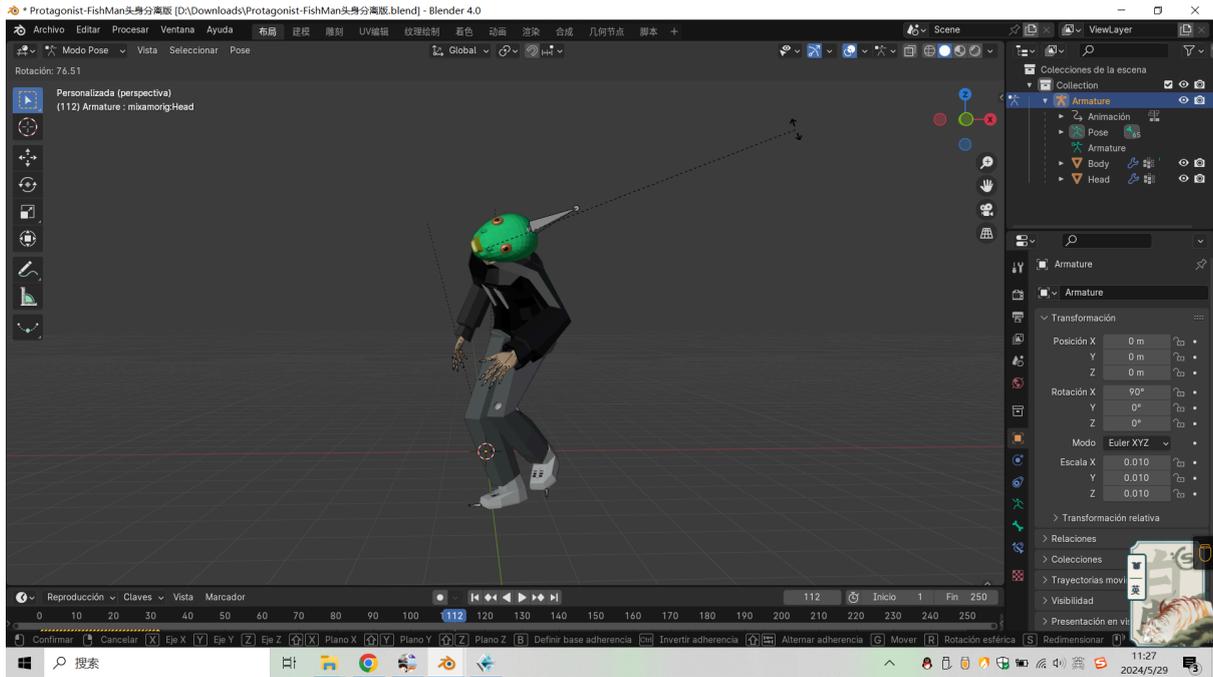


DOWNLOAD SETTINGS

Format	Skin
FBX Binary(.fbx) ▼	With Skin ▼
Frames per Second	Keyframe Reduction
30 ▼	none ▼

CANCEL **DOWNLOAD**

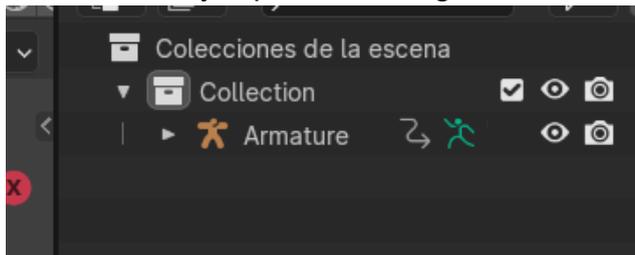
Una vez descargados los archivos, abrimos Blender e importamos el archivo fbx. El esqueleto estará automáticamente enlazado y el modelo tendrá las animaciones, tal como se muestra en la imagen.



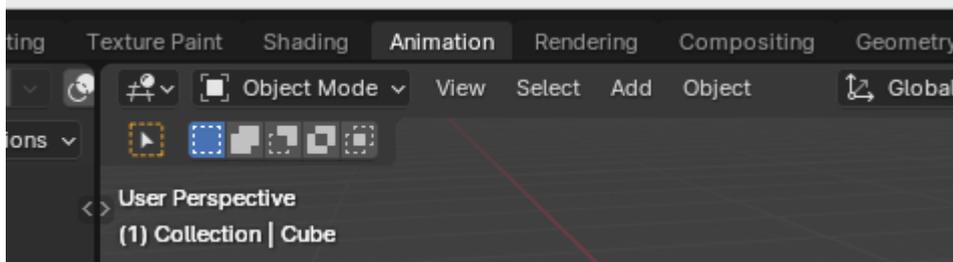
4.3 Integración y exportación de múltiples animaciones

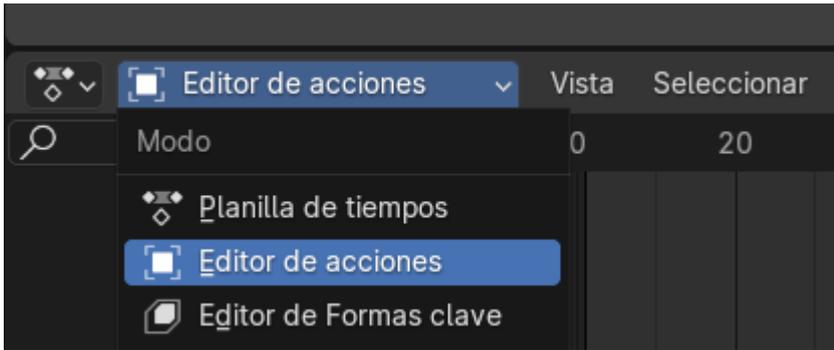
Mixamo tiene buena compatibilidad con otros programas y motores 3D, lo que nos permite integrar fácilmente las animaciones generadas por Mixamo en nuestro proyecto.

Si necesitamos integrar múltiples animaciones en un solo archivo, usaremos Blender para realizar esta operación. Primero, debemos importar todos los archivos fbx con las animaciones descargadas de Mixamo a Blender. Luego, eliminamos todos los esqueletos adicionales, dejando solo uno. No hay que preocuparse por las animaciones, ya que estas se guardan automáticamente.

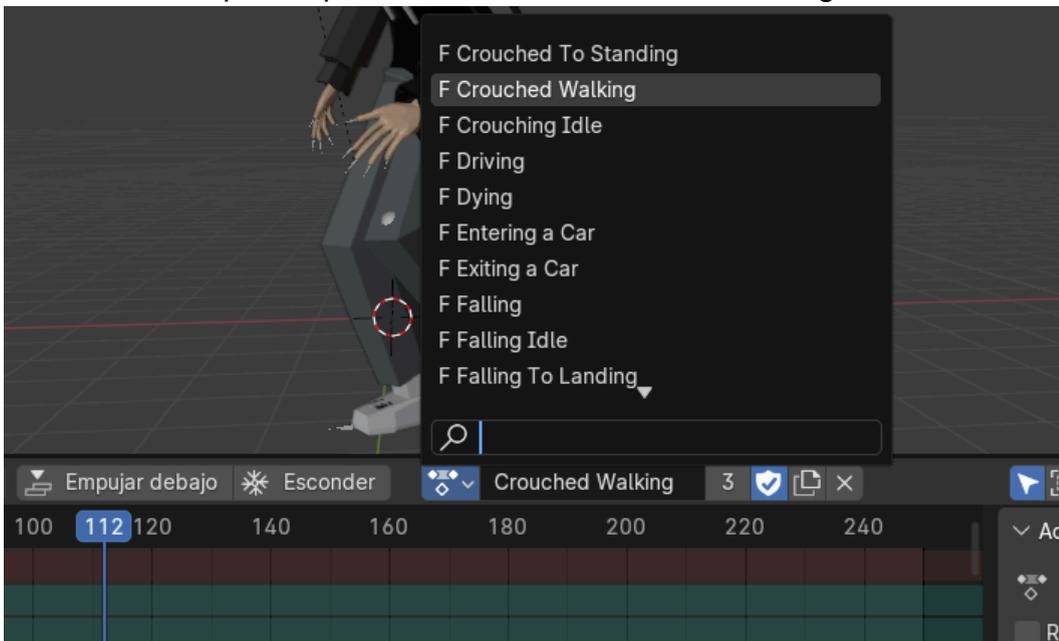


Después, en la pestaña de animación de Blender, en el modo "Editor de acciones".

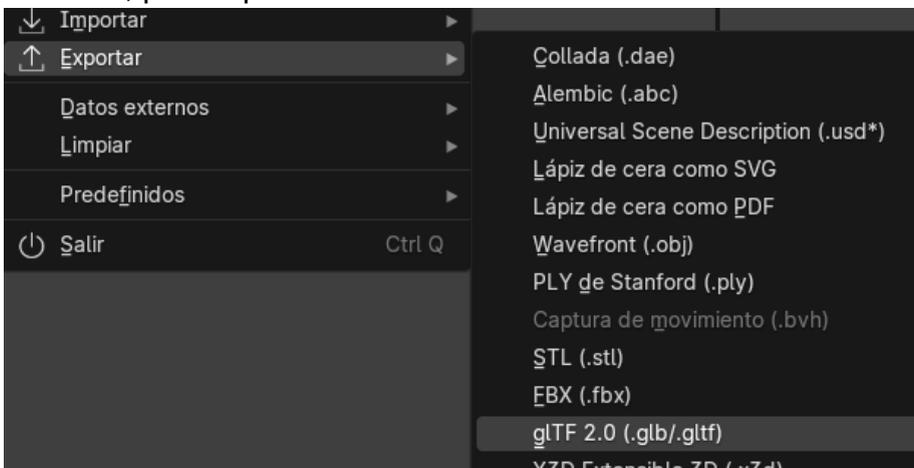




En el panel de la derecha, seleccionamos cada animación una por una y hacemos clic en "esconder" para guardarlas e integrarlas. Al principio, estas animaciones no tienen nombre, por lo que debemos nombrarlas antes de guardarlas.



Una vez completados estos pasos, guardamos los cambios y exportamos el archivo desde Blender en formato glb. Elegimos glb porque nuestro proyecto en Godot no está usando los plugins que sirven para soportar de importar con fbx y otros formatos, por lo que solo admite este formato.



De esta manera, obtenemos un archivo con múltiples animaciones que podemos importar a Godot.

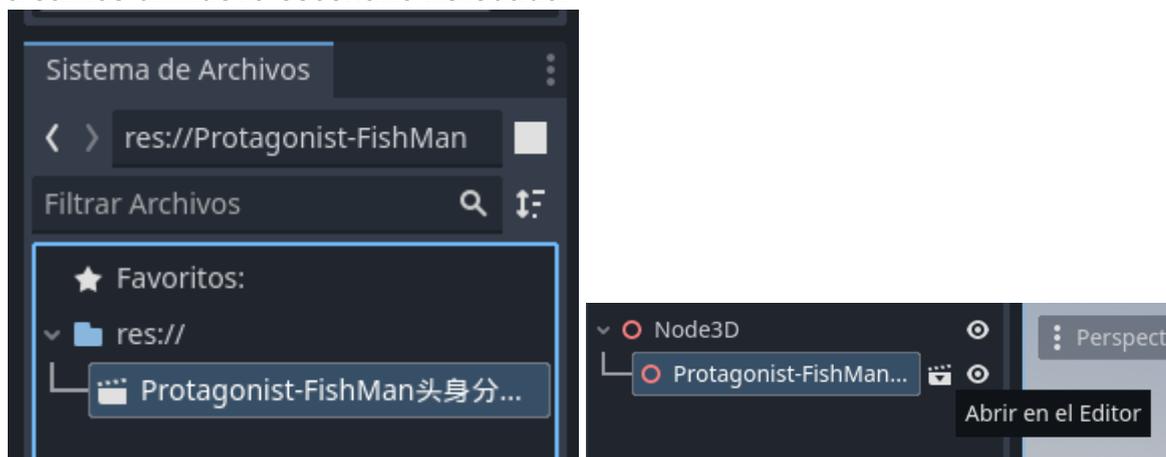
En cuanto a nuestros modelos, la cantidad de animaciones varía. Nuestros personajes principales suelen tener más de 40 animaciones integradas, incluyendo varias de espera, agacharse, caminar, correr, saltar, entre otras. En comparación, los enemigos y NPCs tienen menos animaciones.

En nuestra versión beta, los enemigos, como los extraterrestres, solo tienen animaciones básicas de ataque, correr, levantarse y morir.

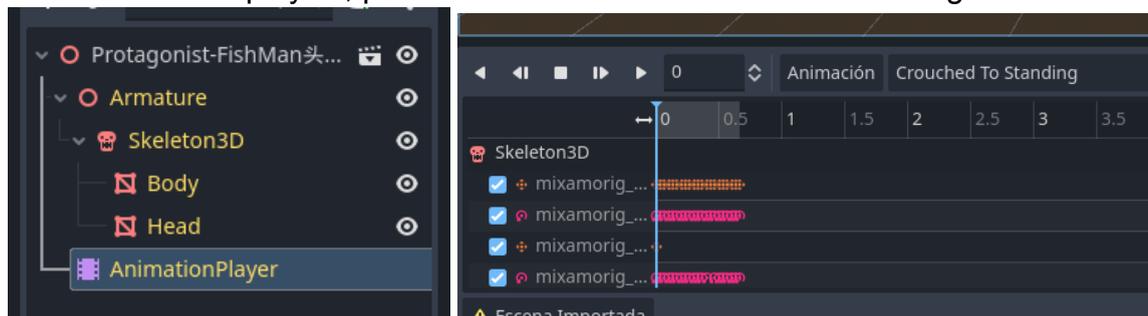
El NPC Robot tiene más animaciones, incluyendo una animación de espera.

4.4 Importación de archivos de modelo en Godot

En Godot, arrastramos nuestro archivo glb al sistema de archivos. Luego, en un escenario utilizando Node3D, arrastramos el archivo glb debajo de Node3D y podremos ver nuestro modelo en Godot. Abrimos el editor junto a nuestro modelo y creamos un nuevo escenario heredado.



En el "animation player", podremos ver todas las animaciones integradas.

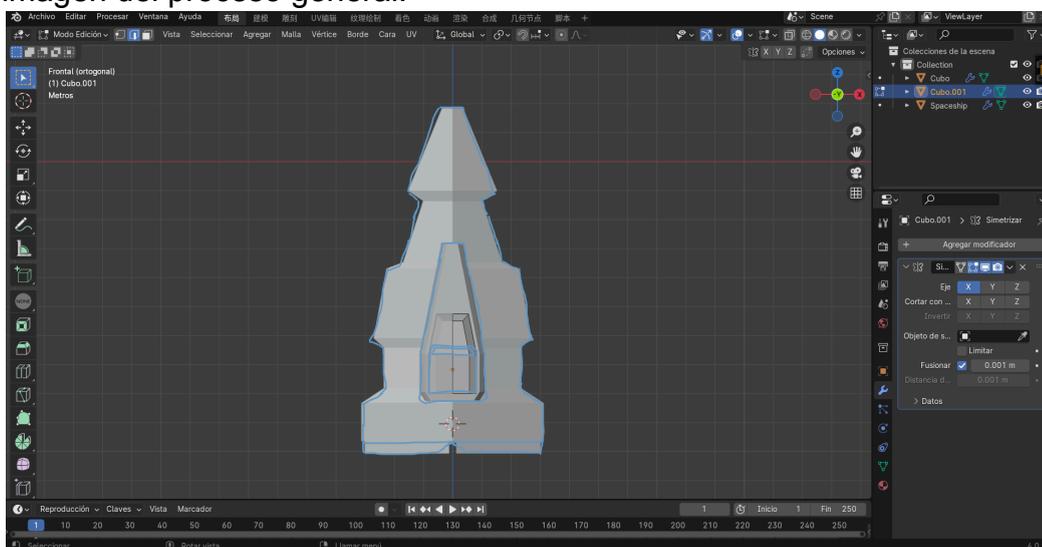




4.5 Escena

4.5.1 Nave espacial

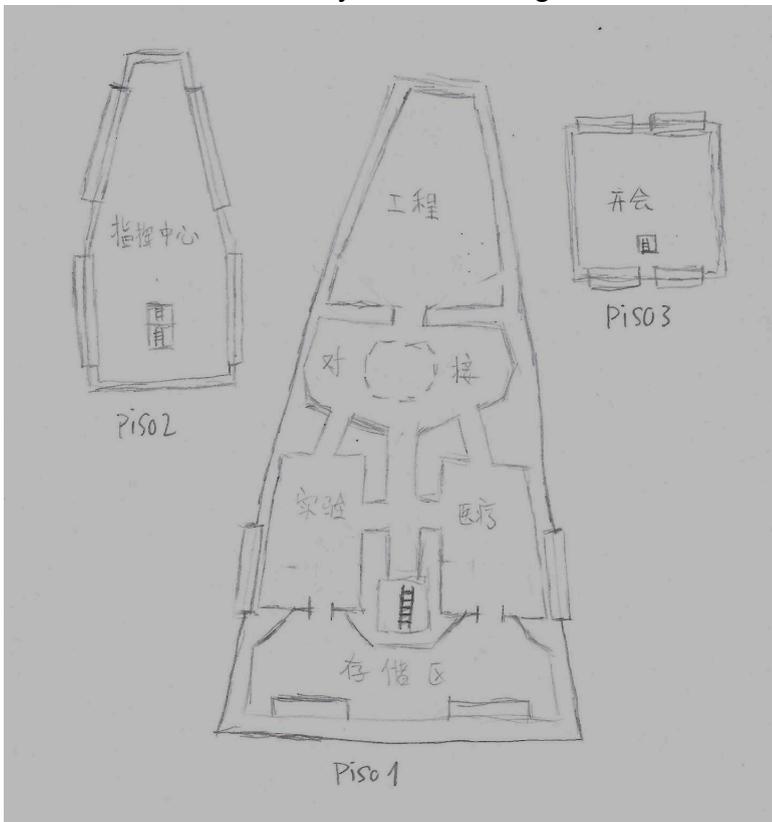
Como en el momento de crear la escena nuestro proyecto ya estaba muy ajustado de tiempo, buscamos un tutorial en video para crear rápidamente y de manera sencilla nuestra estación espacial [Blender][4]. A continuación, se muestra una imagen del proceso general:



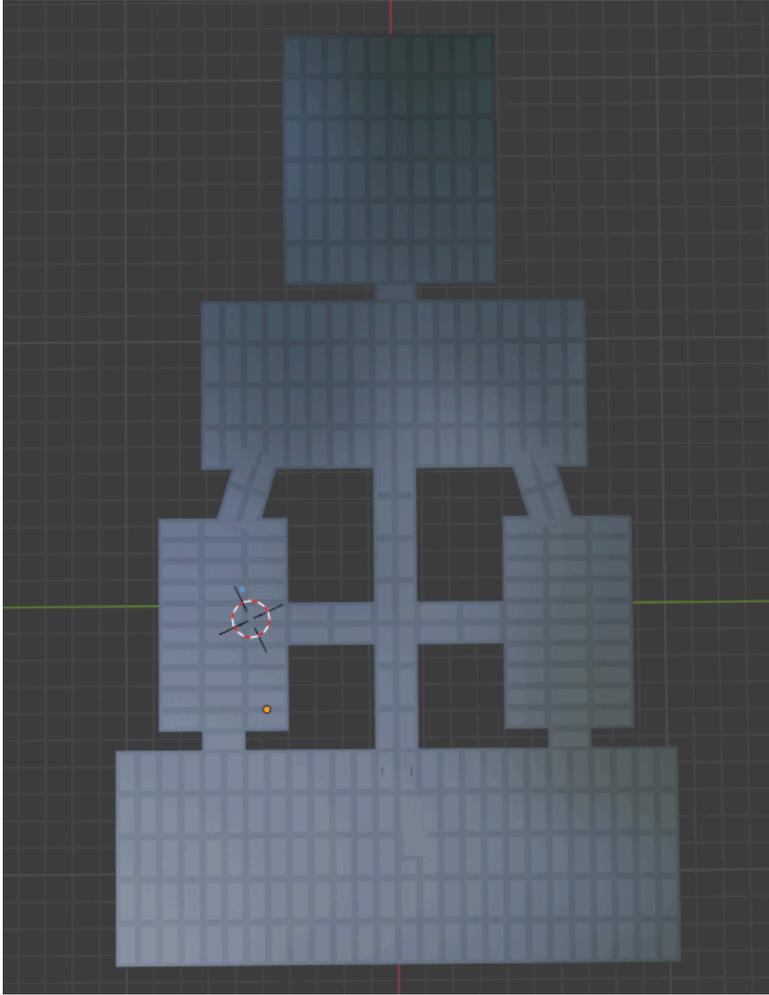


4.5.2 Interior de la nave espacial

Cuando modelamos rápidamente nuestra nave espacial siguiendo el tutorial, necesitamos considerar la estructura interna de la nave. Por eso, dibujamos un esquema simple del interior de la nave, que tiene tres niveles y cada sala tiene una función diferente, como el laboratorio, el centro de mando, la sala de reuniones, la enfermería, el almacén y la sala de ingeniería.



Una vez que tuvimos el boceto y asignamos estas áreas, comenzamos a modelar el interior de la nave espacial, tomando como referencia algunas construcciones de [\[Blender\]](#)^[5]. La mayoría de los muebles y objetos provienen de SketchFab [\[Sketchfab\]](#). Aquí está el resultado:





4.6 Icono

Como nuestro juego tiene un estilo humorístico y absurdo, pensamos que nuestro icono debería tener un poco de "personalidad". Así que dibujamos un boceto de uno de nuestros protagonistas, FishMan, haciendo una pose muy característica y acorde con su estilo, similar a las imágenes que se encuentran en internet.



Luego usamos Photopea para procesarlo, colorearlo y optimizarlo, obteniendo la primera versión de nuestro icono del juego:



Más tarde, consideramos que el icono era demasiado rudimentario, así que lo optimizamos nuevamente. Además, utilizamos TextStudio para proporcionar las fuentes en formato PNG transparente, y también imágenes de Onlygfx, Pngall y Pixabay. Finalmente, este es nuestro icono actual, que es la segunda versión:



Icono de versión sin título

5. La creación del nuestro proyecto

5.1 Control e interacción con los personajes

En nuestro juego podemos controlar el movimiento del personaje con el **teclado**, utilizar el **ratón** para mover la perspectiva del personaje, pulsar el **botón izquierdo** del ratón para disparar y pulsar el **botón derecho** del ratón para lanzar el objeto sujetado, entre otras acciones.

Actualmente, hemos asignado las siguientes acciones al personaje: la barra **espaciadora** hace que el personaje salte; la tecla **WASD** hace que el personaje se mueva en diferentes direcciones; [\[Godot\]\[1\]](#) **F5** cambia el punto de vista del personaje, actualmente tenemos una vista en primera persona y una vista en tercera persona; [\[Godot\]\[4\]](#) **CTRL** hace que el personaje se arrodille; **SHIFT** hace que el personaje corra; **ALT** gira el punto de vista del personaje 45 grados a la izquierda, a la derecha y hacia atrás; y los botones **número 1 y 2** para cambiar el arma del personaje; **R** para recargar el arma y **F** para interactuar con el escenario o los "NPC".

```
# Para tener el input de movimiento  
var input_dir = Input.get_vector("move_left", "move_right", "move_forward", "move_back")
```

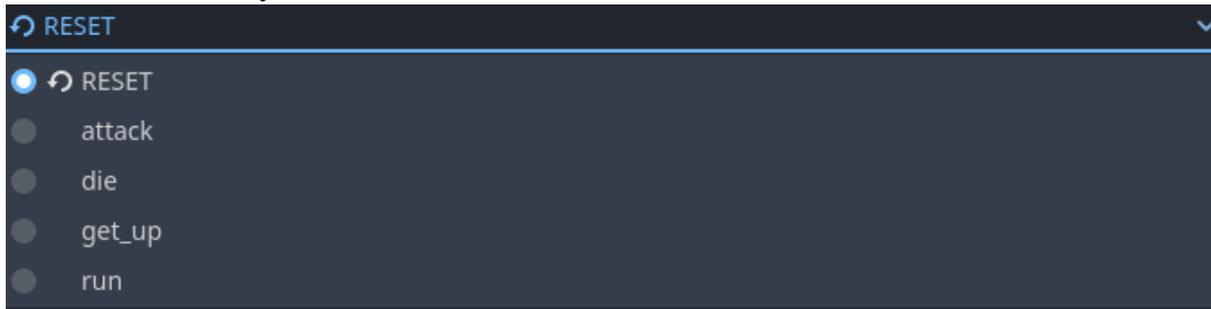
Estas asignaciones de teclado se establecen en la asignación de entrada a la configuración del proyecto de godot. Añada nombres y entrada de teclado y, a continuación, utilice el código para hacer referencia a estos nombres.



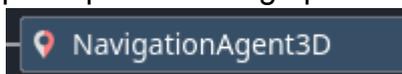
```
if Input.is_action_pressed("Run"):
    >| # && animation_tree.get("parameters/Transition/current_index") == 1
    >| SPEED_Normal = lerp(SPEED_Normal, speed_run, delta * speed_lerp)
    >| walking = false
    >| crouching = false
    >| sprinting = true
    >| jumping = false
elif not input_dir:
    >| walking = false
    >| sprinting = false
    >| crouching = false
else:
    >| SPEED_Normal = lerp(SPEED_Normal, speed_walk, delta * speed_lerp)
    >| walking = true
    >| crouching = false
    >| sprinting = false
    >| jumping = false
if Input.is_action_just_pressed("jump") and is_on_floor():
    >| velocity.y = JUMP_VELOCITY
    >| sliding = false
    >| jumping = true
```

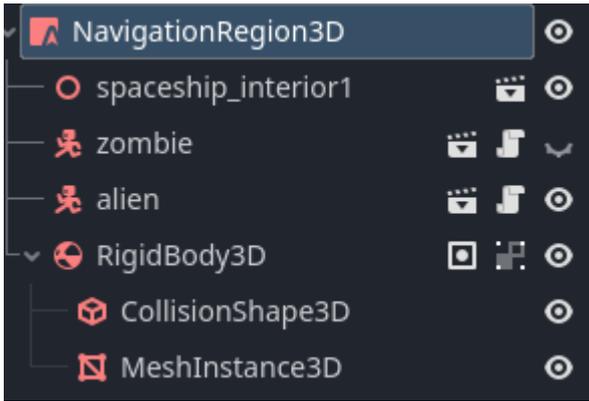
5.2 Comportamiento hostil e inteligencia artificial

Nuestro enemigo tiene una serie de animaciones de movimiento, como atacar, correr, levantarse y morir.



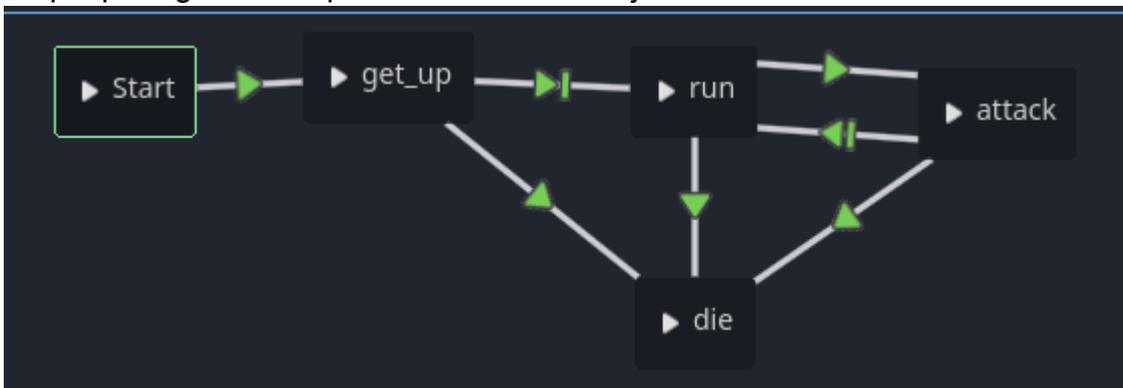
Para que este enemigo encuentre nuestra ubicación y se mueva en nuestra dirección, utilizamos las funciones de navegación y scripting de Godot, el enemigo podría hacer un script para obtener nuestra ubicación y luego moverse en nuestra dirección, seguido del uso de la navegación para hornear una malla para la escena para que el enemigo pueda sortear los obstáculos. [[Godot](#)][3]





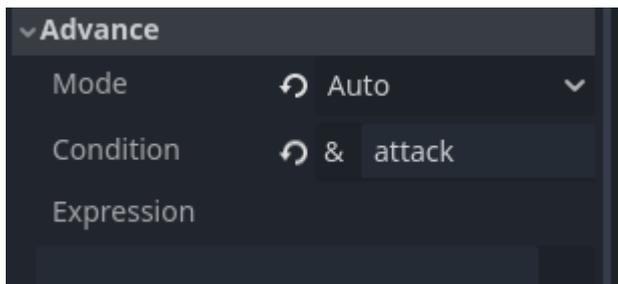
```
match state_machine.get_current_node():
  >| "run":
  >| >| # 导航
  >| >| # 将玩家的全局位置提供给导航
  >| >| nav_agent.set_target_position(player.global_transform.origin)
  >| >| # 获取僵尸要到下一个位置
  >| >| var next_nav_point = nav_agent.get_next_path_position()
  >| >| # 敌人找到我们的位置
  >| >| velocity = (next_nav_point - global_transform.origin).normalized() * speed
  >| >| rotation.y = lerp_angle(rotation.y, atan2(-velocity.x, -velocity.z), delta * 10.0)
  >| "attack":
  >| >| look_at(Vector3(player.global_position.x, global_position.y, player.global_position.z))
```

Una vez que estas funciones funcionan correctamente, creamos un árbol de animación para ese enemigo, planificamos la aplicación de la animación y utilizamos scripts para gestionar qué animación debe ejecutarse.



En nuestro árbol de animación expresamos lo que debe suceder cuando una acción se realiza o cuando una acción se inicia otra acción también se ejecutará. También habilitamos un nombre condicional para estos procesos de animación para que podamos llamarlo en el script.

```
8  ▾ func _target_in_range():
9  >| # 判断是否在攻击范围内
0  >| return global_position.distance_to(player.global_position) < attack_range
1
2  >|
3  ▾ func _hit_finished():
4  >| # 判断角色是否在攻击范围内并且是否获得被击中反馈 攻击范围+1
5  ▾ >| if global_position.distance_to(player.global_position) < attack_range + 1.0:
6  >| >| var dir = global_position.direction_to(player.global_position)
7  >| >| player.hit(dir)
8
9  ▾ func _on_area_3d_body_part_hit(dam):
0  >| health -= dam
1  >| emit_signal("zombie_hit")
2  ▾ >| if health <= 0:
3  >| >| anim_tree.set("parameters/conditions/die", true)
4  >| >| await get_tree().create_timer(4.0).timeout
5  >| >| queue_free()
6
```



5.3 Diálogo de personajes

Para la interacción del diálogo, creamos un NPC para que interactúe con nosotros, pero aún no es perfecto y se optimizará en el futuro.





5.4 Menú UI

5.4.1 Menú de inicio

El menú de inicio del nuestro juego

→ Jugar, Ajuste, Créditos, Ayuda, Salir

Cuando el ratón pasa por encima o hace clic en estos botones, se producen distintos efectos de sonido. Además, la pantalla tiene una música de fondo que se reproduce automáticamente, pero se puede controlar si se desea reproducir o pausar mediante un botón en la esquina superior derecha. En la parte inferior derecha se muestra el logo de nuestro juego. En la parte inferior izquierda se muestra el logotipo de nuestra empresa.

Las funciones de estos botones son las siguientes: hacer clic en "Iniciar el juego" cambiará la escena y le llevará al campo de batalla principal o al mapa inicial; hacer clic en "Configuración del juego" le permitirá ajustar diferentes parámetros del juego tales como los efectos de sonido o la resolución; hacer clic en "Sobre nosotros"

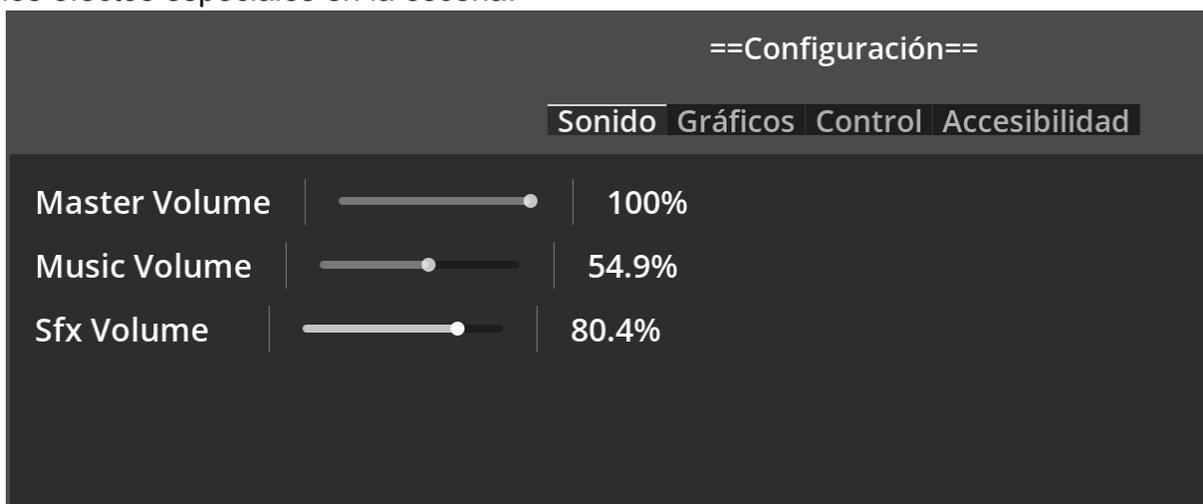
mostrará información sobre nosotros; hacer clic en "Ayuda del juego" proporcionará sugerencias sobre los controles del juego y tutoriales. Por último, hacer clic en "Salir" cerrará el juego. [\[Godot\]](#)[5]



Los configuraciones dentro del Ajustes:

- Ajustes del sonido:
 - Volumen maestro, volumen de música, volumen sfx (Efectos de sonido del juego)

Aquí podemos cambiar el volumen en nuestro juego, el volumen maestro es el volumen del juego como un todo incluyendo los otros volúmenes; el volumen de la música es sobre el volumen de la música; el volumen SFX es sobre el volumen de los efectos especiales en la escena.



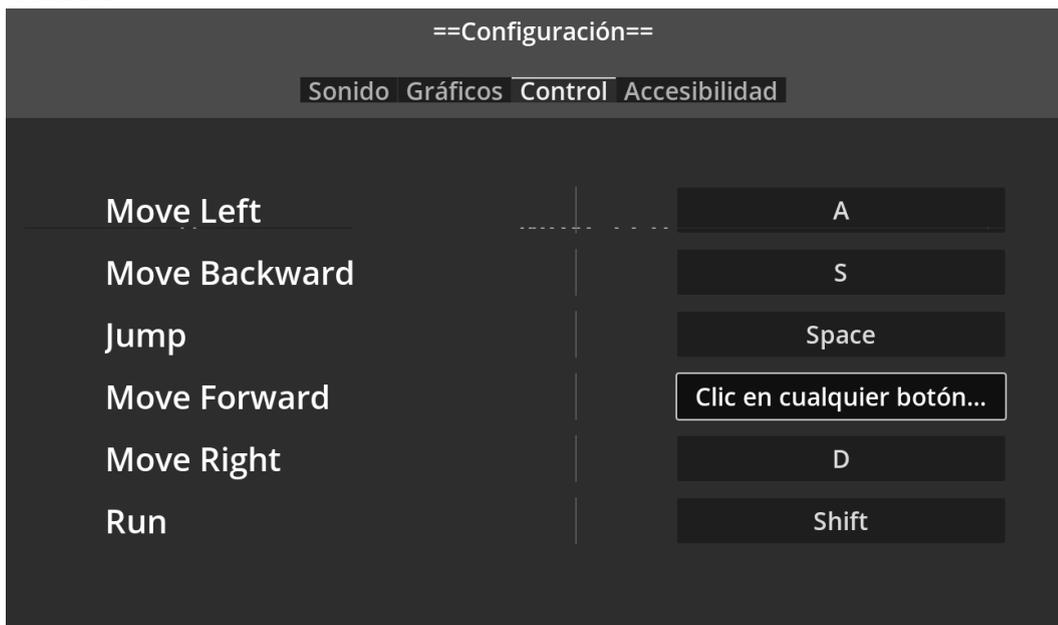
- Ajustes del gráficos:
 - Modo de pantalla, resolución

Contiene ajustes sobre el modo y la resolución de la ventana, y puede cambiar el tamaño de la ventana modificando los valores.

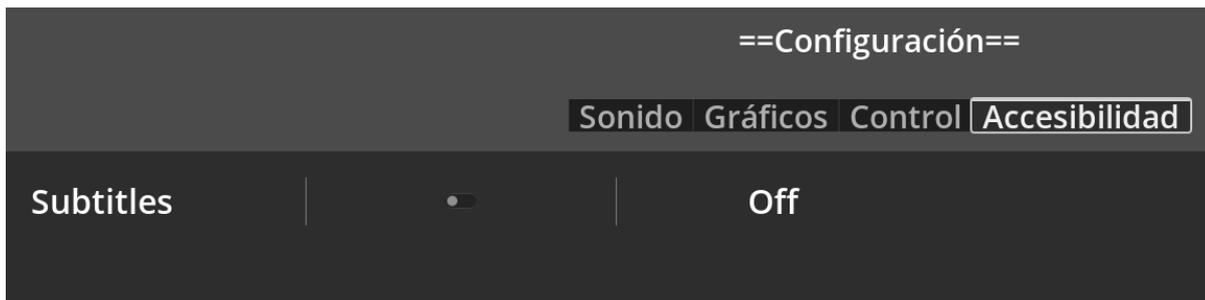


- Ajustes del control :

Contiene todas las asignaciones de entrada del teclado que se pueden cambiar. Podemos cambiar los botones de operación en el juego cambiando la entrada del teclado.

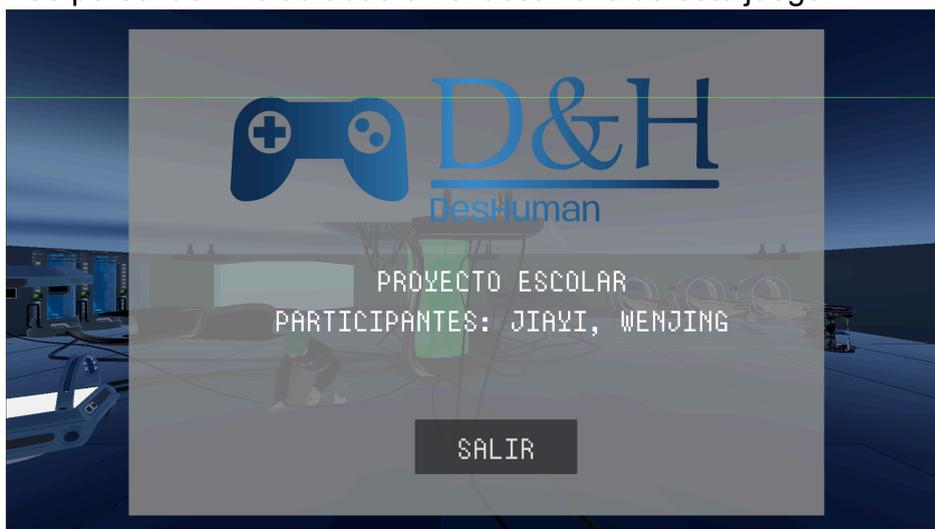


- Ajustes del accessibilitat:
Les funcions d'accessibilitat encara no estan desenvolupades.



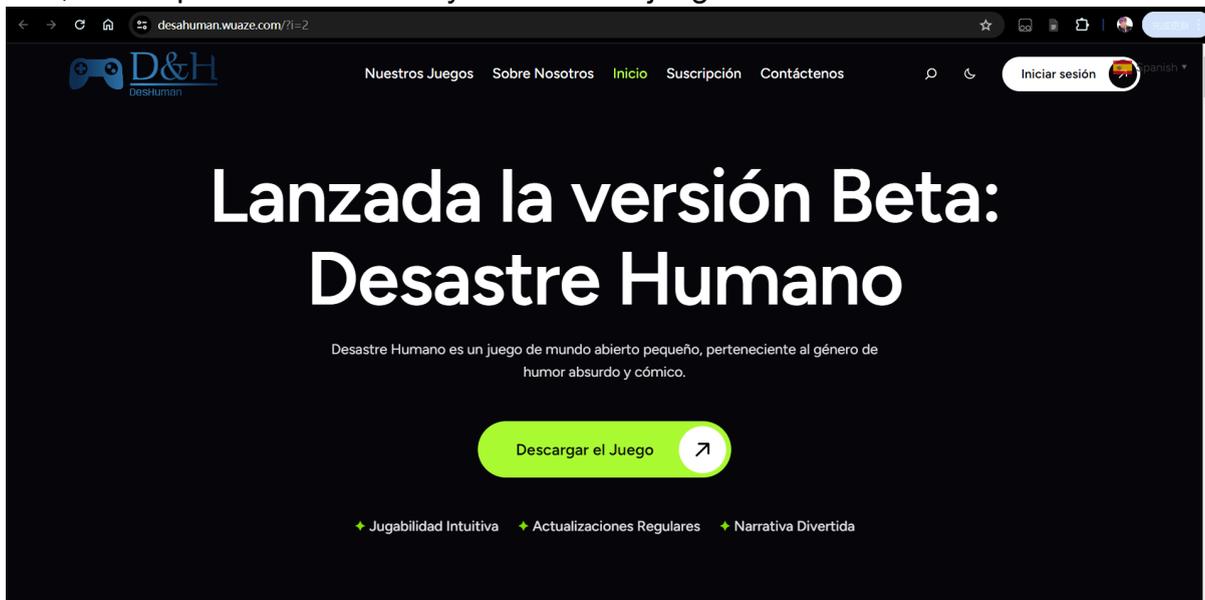
Contenidos dentro del créditos:

Los miembros sobre este proyecto se muestran aquí, y en el futuro puede haber más personas involucradas en el desarrollo de este juego.



Opción de ayuda:

Al hacer clic en el botón de ayuda, se te redirigirá automáticamente a nuestro sitio web, donde podrás encontrar ayuda sobre el juego.

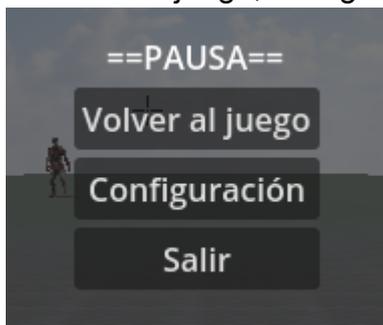


5.4.2 Menú de pausa

En el juego, hemos configurado una interfaz de pausa. Cuando hacemos clic en ESC en el teclado, aparecerá una pequeña interfaz y detendrá el juego. Cuando haga clic en botón "Configuración", irás a la interfaz de configuración mencionada anteriormente y, finalmente, para salir, cuando haga clic en este botón, volverá a la interfaz principal.

- **Cuando hacemos clic en el botón ESC**

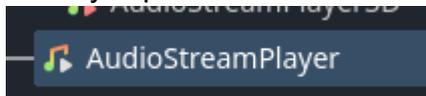
→ Volver al juego, configuración, salir



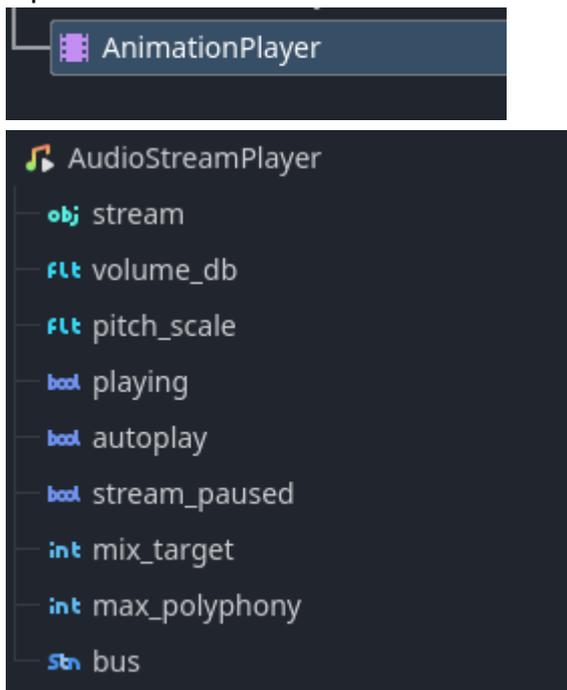
5.5 Sonido y efectos de sonido

5.5.1 Efectos de sonido

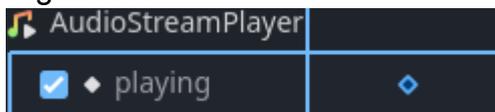
El control de reproducción de audio integrado en godot se puede utilizar para poner audio y reproducirlo en un sitio específico.



Podemos añadir pistas de atributo al audio especificado en la animación para reproducir el audio.



Por ejemplo, puede seleccionar aquí el atributo de reproducción para empezar a reproducir audio cuando la animación llegue a un número determinado de segundos.



Por supuesto, también podemos utilizar scripts para conseguirlo. Podemos añadir si se debe reproducir el audio en un juicio, como empezar a reproducir música de fondo cuando el personaje llega a una ubicación especificada.

```
@onready var so = $AudioStreamPlayer

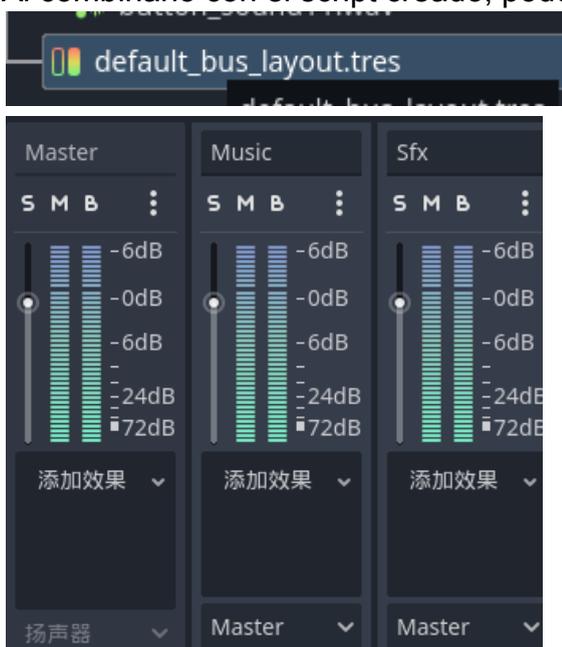
# Called when the node enters the scene tree for the first time
func _ready():
    >| pass # Replace with function body.

# Called every frame. 'delta' is the elapsed time since the previous frame.
func _process(delta):
    >| so._set_playing(true)
```

5.5.2 Integración de audio

Esta serie de audio puede dividirse en diferentes categorías, como efectos de sonido, música de fondo y efectos de sonido de diálogo, etc. Podemos utilizar un diseño de autobús de audio para gestionarlo. Por ejemplo, a continuación, actualmente tenemos un total de tres buses, es decir, el volumen principal, el volumen de la música de fondo y el volumen de los efectos de sonido.

Al combinarlo con el script creado, podemos controlar el volumen del juego.



Este código es un script utilizado para controlar el volumen del bus de audio. Utilice variables de tipo de enumeración para seleccionar diferentes buses de audio ("Master", "Música", "Sfx"). Cuando el nodo esté listo, cuando cambia el valor del control deslizante, se actualiza el volumen del bus de audio actual y se actualiza la pantalla de la interfaz. Dependiendo del tipo de bus de audio actual, se envían diferentes señales de configuración al servidor de audio.

```
func get_bus():
  bus_index = AudioServer.get_bus_index(bus_name)

func set_slider_value():
  hslider.value = db_to_linear(AudioServer.get_bus_volume_db(bus_index))
  set_num_label()

func on_value_changed(value : float):
  AudioServer.set_bus_volume_db(bus_index, linear_to_db(value))
  set_num_label()

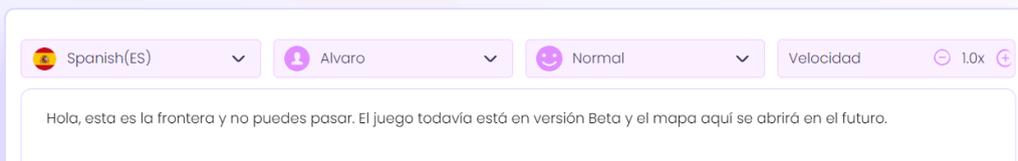
match bus_index:
  0:
    SettingSignal.emit_on_master_sound_set(value)
  1:
    SettingSignal.emit_on_music_sound_set(value)
  2:
    SettingSignal.emit_on_sfx_sound_set(value)
```

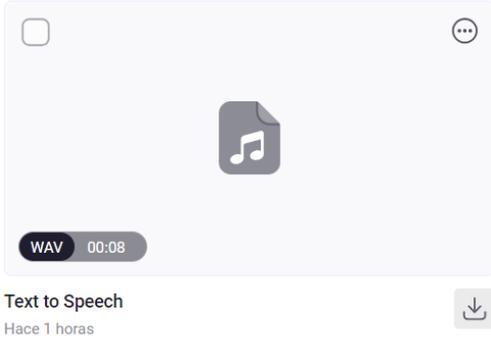
5.5.3 Voz de los personajes

En cuanto a la voz de los personajes, dado que la trama no está implementada en nuestra versión beta del juego, no hay una gran necesidad de textos. Sin embargo, utilizaremos Vidnoz AI para generar un archivo de audio simple en formato .wav para usar en el único diálogo presente en nuestra versión beta del juego.

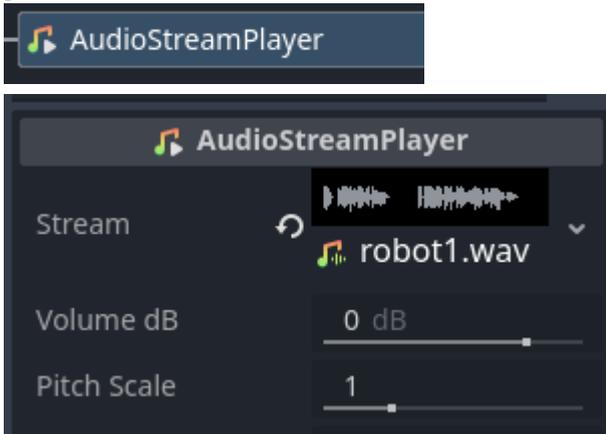
Texto a Voz Gratis – Convertir texto a voz online gratis en tiempo real

El mejor convertidor de texto a voz online gratis impulsado por inteligencia artificial, capaz de pasar texto a voces masculinas y femeninas en más de 140 idiomas y más de 470 voces. ¡100% GRATIS!





Aquí creamos un nodo de reproducción de audio al que añado el archivo de audio generado anteriormente.



A continuación, utilizamos un pequeño fragmento de código para determinar cuándo debe reproducirse el audio.

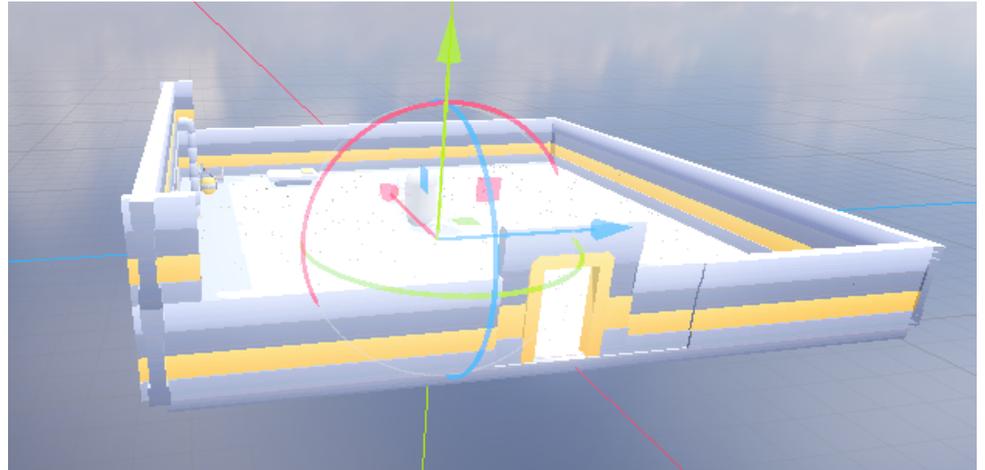
```
if event.is_action_pressed("interactable") and dialog.has_focus():
>| Input.set_mouse_mode(Input.MOUSE_MODE_VISIBLE)
>| dialog_robot.visible = true
>| option_dialog.visible = false
>| animation_text.play("TextDisplay")
>| audio.play()
```

6. Proceso de desarrollo e implementación

6.1 Proceso de implementación en Godot

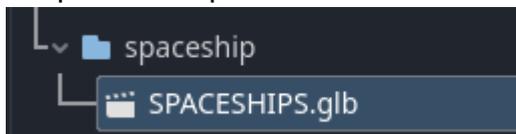
6.1.1 Editor de escenas

GridMap permite la colocación interactiva de una cuadrícula en una cuadrícula. Utilice una MeshLibrary que contiene una lista de fichas. Cada ficha es una malla con material y formas opcionales de colisión y navegación. [\[Godot\]](#)[2]



Además de usar ese nodo de bloque 3d para construir la escena, también podemos importar un modelo 3d externo para usarlo como escena, importar un archivo glb exportado desde blender o cualquier otro sitio a godot, y crear ese archivo como una escena heredada separada en godot.

Así podremos personalizar la escena según nuestras necesidades.





También existe una forma menos cómoda de crear escenas, utilizando formas de malla, cuerpos rígidos y sólidos geométricos, pero este método es más engorroso y requiere más CPU para escenas grandes.

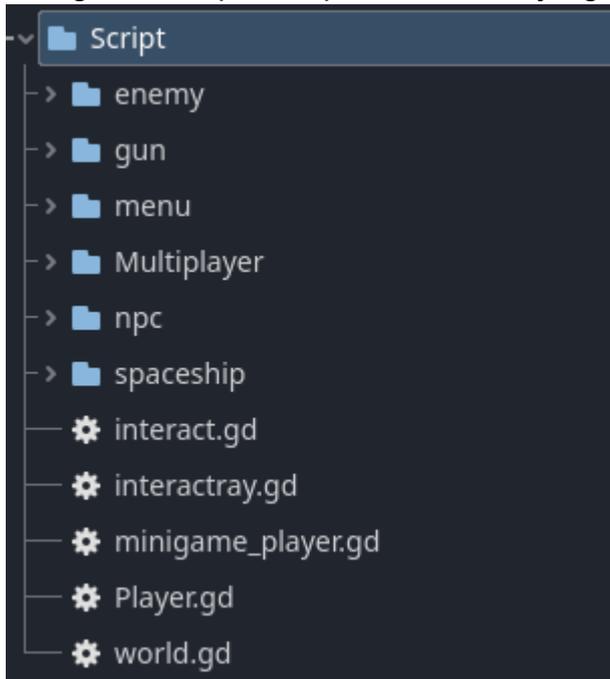


6.1.2 Escritura de Scripts

Para editar los scripts utilizamos el lenguaje de scripting GDscript que viene con godot, y el editor también está integrado en godot.

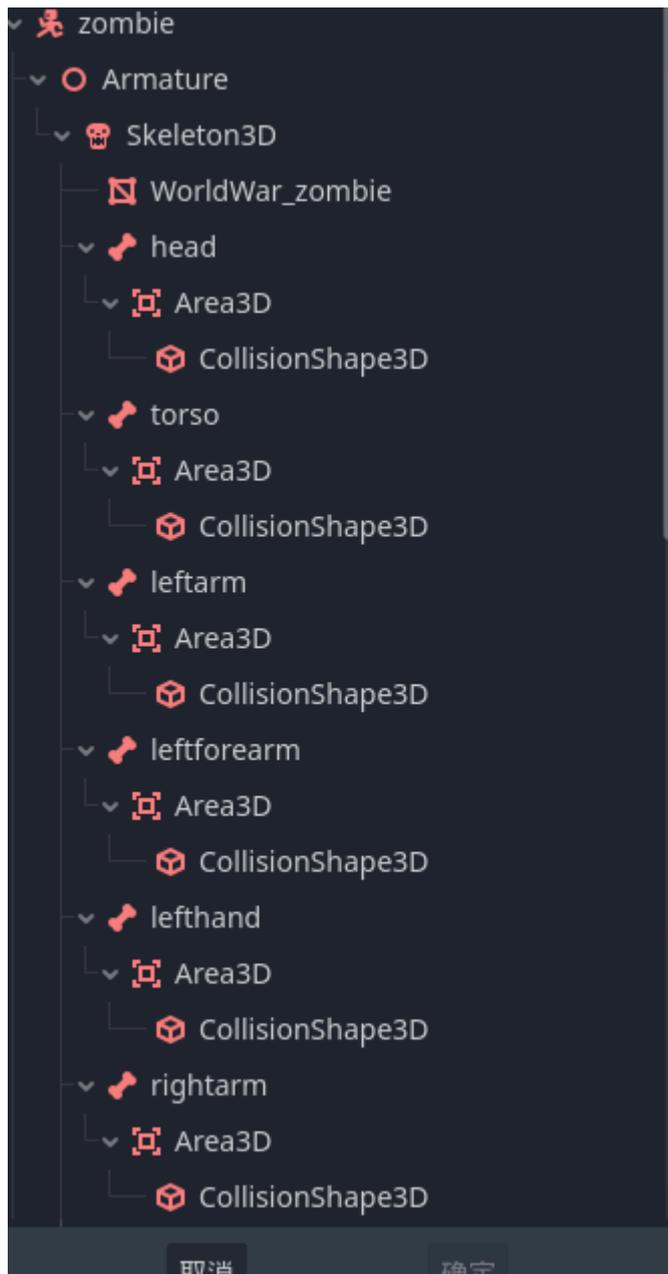
Hasta ahora hemos creado un número de scripts con diferentes efectos y funcionalidades, incluyendo pero no limitado a scripts para acciones de daño enemigo, movimiento, animaciones de acción y cambios de perspectiva, efectos de disparo, llamadas a menús, ajustes de jugabilidad, auto-carga, diálogos e interacciones, y más.

Aunque actualmente el juego funciona bien, hay un poco de sobrecarga de código en algunos scripts, lo que ralentiza el juego, por lo que se optimizará en el futuro.



6.1.3 Editor de animación

Todas las animaciones de personajes, nodos, objetos, estructuras de colisión, etc. Se pueden editar utilizando el nodo "Animación de personajes".

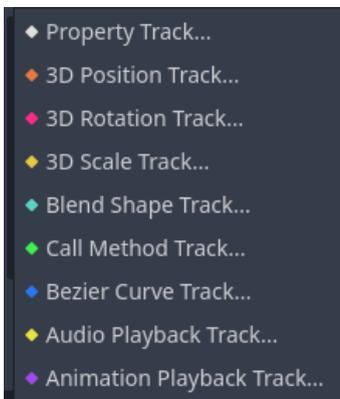


Podemos animar estos diferentes atributos de nodos, como la posición del nodo, el tamaño, la visibilidad y editar nuestras señales personalizadas.

Por ejemplo, cuando la animación se ejecute hasta un determinado fotograma, el daño del personaje se duplicará; cuando la animación se reproduzca hasta 3 segundos, el personaje será más grande; y crear los movimientos del personaje, como la posición, el tamaño y la rotación de los huesos.

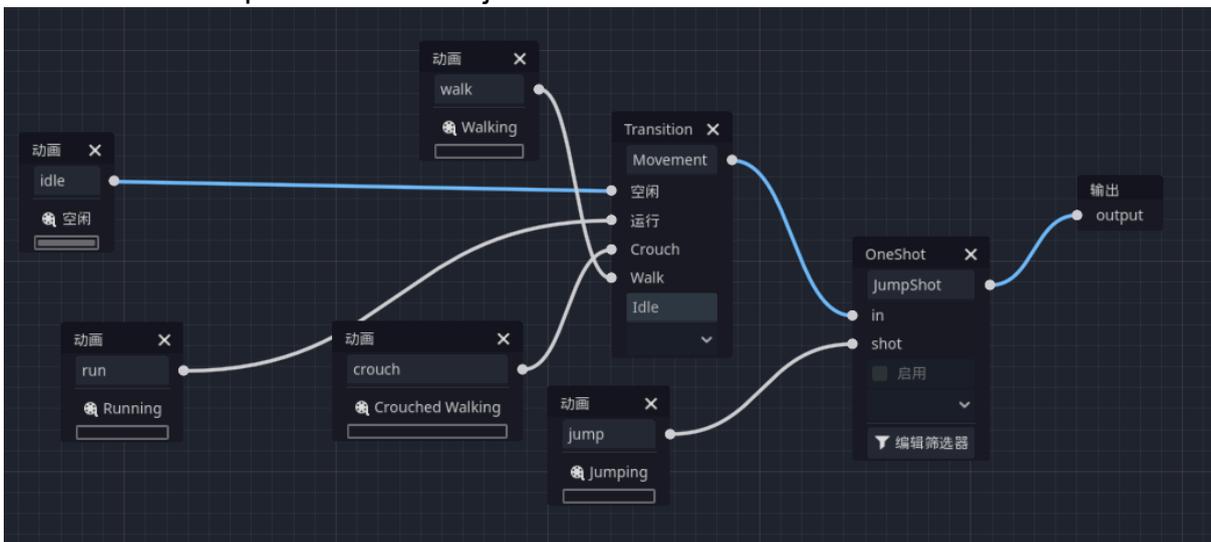


Estas animaciones se pueden lograr agregando pistas específicas. Diferentes pistas tienen diferentes funciones, pero las más utilizadas son cambios en los atributos de los nodos.



Una vez que tengamos animaciones, podemos usar árboles de animación para planificar y configurar llamadas fluidas a esas animaciones.

Establece diferentes animaciones en el árbol de animación y dáselas a diferentes nodos de animación para procesarlas, entonces en el script las animaciones en estos nodos serán condicionadas y si la condición es correcta, entonces la animación correspondiente será ejecutada.

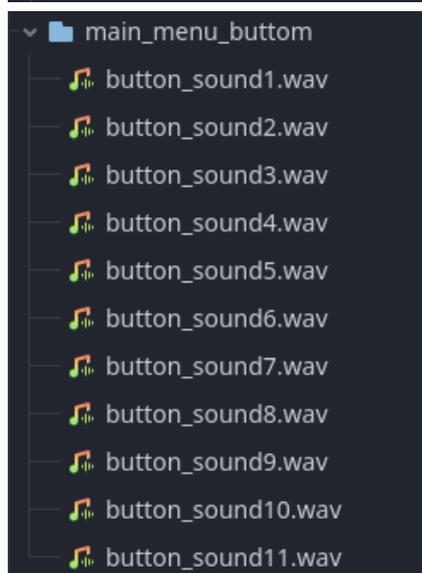
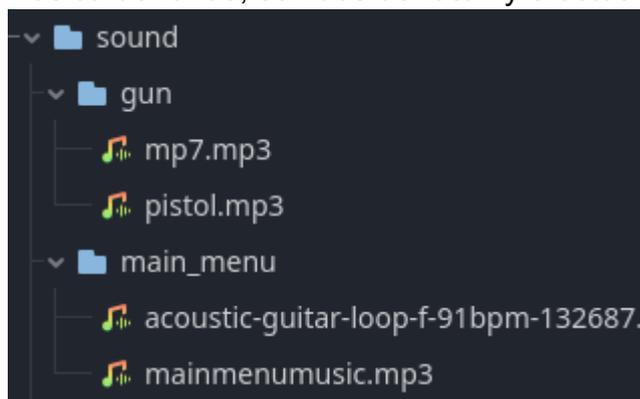


```
# Animation tree about movement
func handle_animation():
>| match CurrentAni:
>| >| IDLE:
>| >| >| animation_tree.set("parameters/Movement/transition_request", "Idle")
>| >| WALK:
>| >| >| animation_tree.set("parameters/Movement/transition_request", "Walk")
>| >| RUN:
>| >| >| animation_tree.set("parameters/Movement/transition_request", "Run")
>| >| CROUCH:
>| >| >| animation_tree.set("parameters/Movement/transition_request", "Crouch")
```

6.1.4 Editor de audio

El audio y los efectos de sonido utilizan un nodo de reproducción de sonido en Godot. Podemos personalizarlo poniendo nuestra música y efectos de sonido preparados en este nodo.

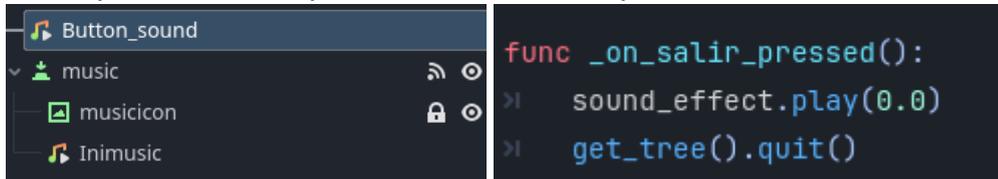
Las tres imágenes siguientes son algunos efectos de sonido, música y controladores de volumen que hemos preparado. Estos incluyen sonidos de prensa de botones, música de fondo, sonidos de ratón y efectos de sonido de ataque de armas.





De los dos nodos de reproducción de audio de la siguiente figura, el primero es una reproducción de audio en bucle aleatoria. Cuando nuestro ratón entra en el intervalo de botones, una de las listas de audio de este nodo será seleccionada aleatoriamente para la reproducción.

El segundo nodo de audio reproduce automáticamente música de fondo. Se añade un script a este nodo para controlar si se reproduce.



The screenshot shows the audio mixer for a sound named "Button_sound". The interface is in Chinese and displays several instances of the sound. Each instance has a "Stream" set to "butto" and a "Weight" of 1. Below the mixer, there are global settings for the sound, including Volume dB (0 dB), Pitch Scale (1), and various playback options like "Playing", "Autoplay", and "Stream Paused", all of which are currently disabled. The "Mix Target" is set to "Stereo" and "Max Polyphony" is set to 1. The "Bus" is set to "Sfx".

Button_sound

筛选属性

Weight 1

Stream butto

Weight 1

Stream butto

Weight 1

Stream butto

Weight 1

Stream butto

Weight 1

< 1 / 3 >

> Resource (1 处改动)

Volume dB 0 dB

Pitch Scale 1

Playing 启用

Autoplay 启用

Stream Paused 启用

Mix Target Stereo

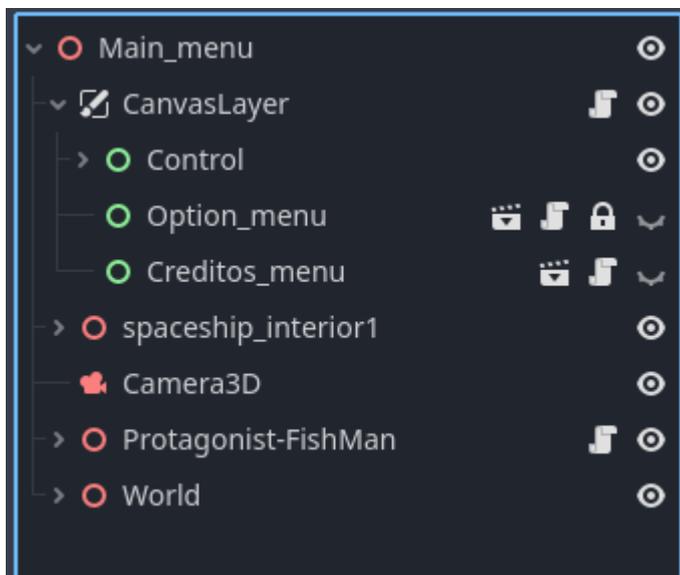
Max Polyphony 1

Bus Sfx

6.1.5 Editor de menu

La imatge següent és la nostra pàgina de menú principal temporal, que es compon de diferents nodes, inclosos text enriquit, botons, nodes de reproducció d'àudio i controls de textura.

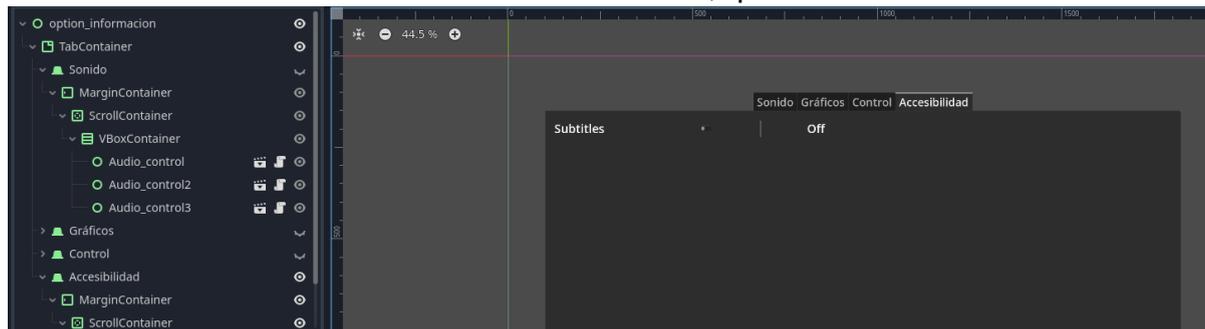
Els nodes de dalt es troben a un nivell inferior.



Els nodes utilitzats en les diferents interfícies de menú són similars, però la disposició i els scripts que hi ha darrere són diferents.

Els nodes del menú de les imatges següents són: "tabcontainer", el contenidor del node "tabbutton", que s'utilitza per mostrar el botó a sobre de la interfície i es pot personalitzar a través de la barra lateral del "margincontainer"; a "tabbutton" s'utilitza per controlar els marges de la interfície; "scrollcontainer" s'utilitza per crear un menú que permet lliscar, per tal d'acomodar text llarg o altres configuracions;

"vboxcontainer" s'utilitza per mostrar els nodes secundaris verticalment. L'últim node fill són els altres escenaris de la interfície d'usuari, que es descriuen a continuació.



Les imatges següents mostren nodes utilitzats per mostrar informació principal. Estan separades per millor distinció, i tots aquests interfícies de configuració requereixen scripts per ajustar les opcions del joc. Després de crear aquesta escena d'interfície d'usuari, es pot instanciar en altres escenes.

Per a diferents nodes, es poden personalitzar les seves opcions des de la barra lateral.

Altres nodes que no es mostren a les imatges són:

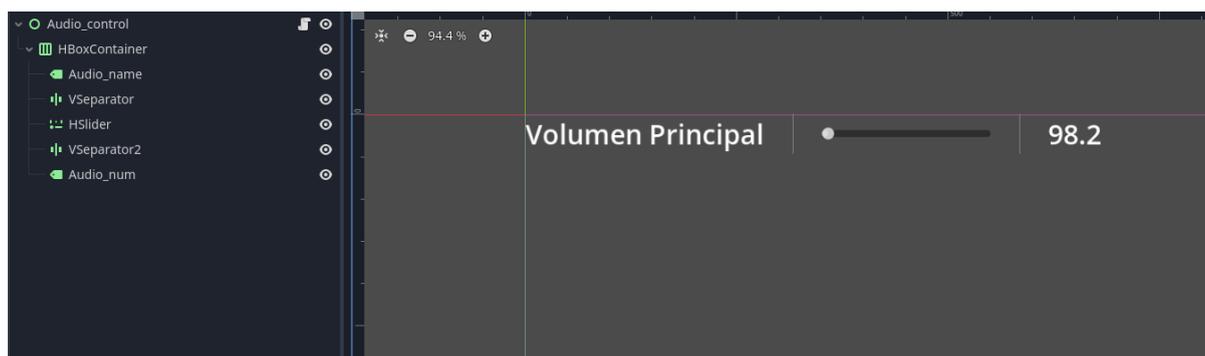
"HBoxContainer", similar al "vboxcontainer" però disposa els nodes horitzontalment.

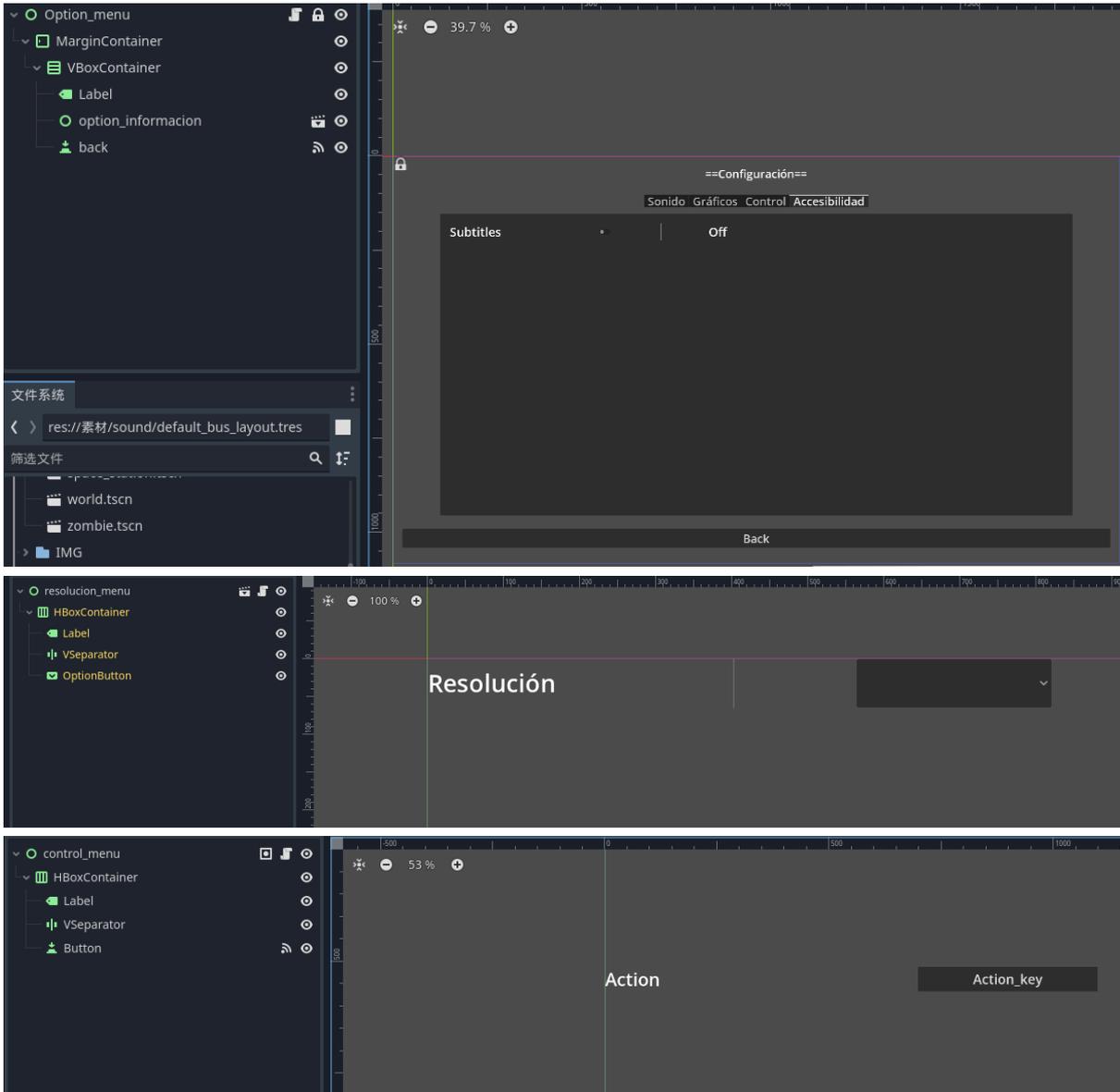
"Label", mostra text introduït.

"VSeparator", separa els nodes i, per defecte, mostra una línia, però es pot triar si es mostra o no des de la barra lateral.

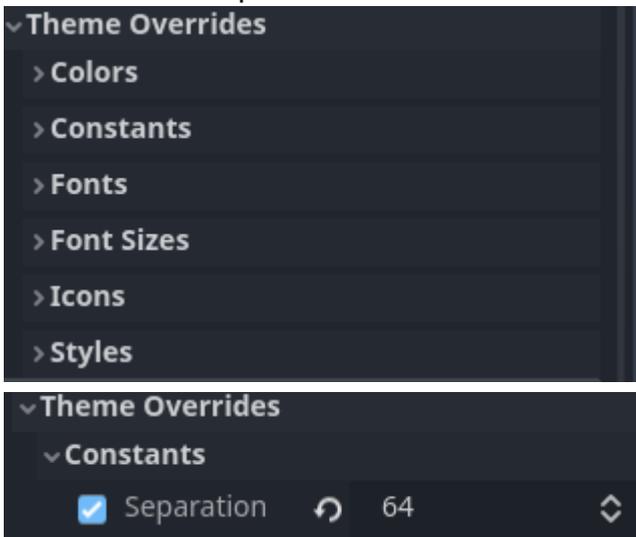
"HSlider", és un control deslizable que pot controlar nodes secundaris o ajustar configuracions del joc. Aquí s'utilitza per ajustar el volum del joc.

"OptionButton", és un botó binari que representa 0 i 1, es pot utilitzar per activar o desactivar opcions.





Les imatges següents són paràmetres que sovint s'han de canviar al menú d'edició. Sovint s'utilitzen per canviar la mida de la font, l'espaiat, la mida del contenidor, etc.



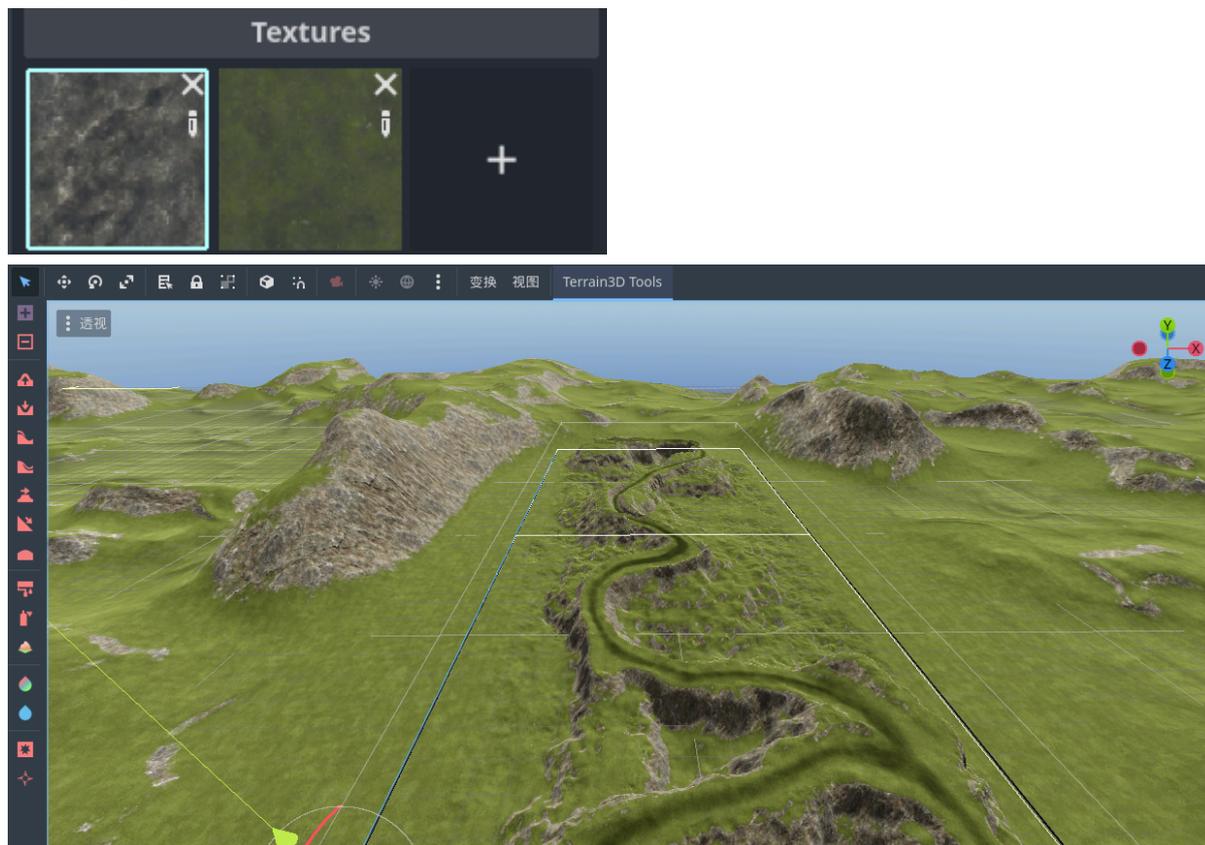


6.1.6 Editor de terrano 3D

Utilizamos un plugin de GitHub de código abierto para editar y generar un mapa en forma de montaña que se utilizó como escenario para nuestro juego. [[Godot](#)][9]

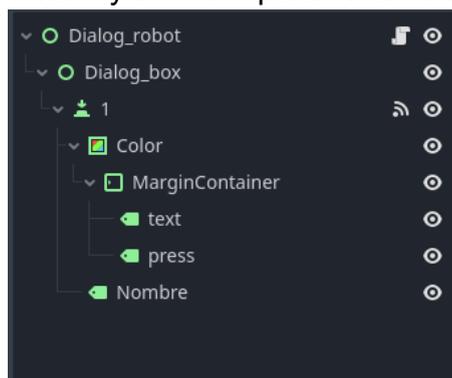


El plugin nos permite personalizar el mapa añadiendo texturas al terreno, y tras añadir las texturas podemos utilizar las herramientas propias del plugin para modificar el terreno.

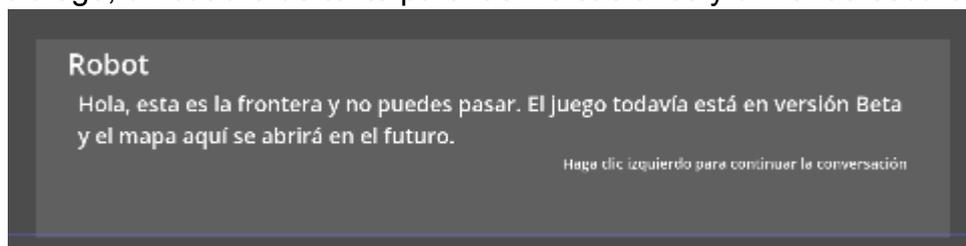


6.1.7 Editor de diálogos

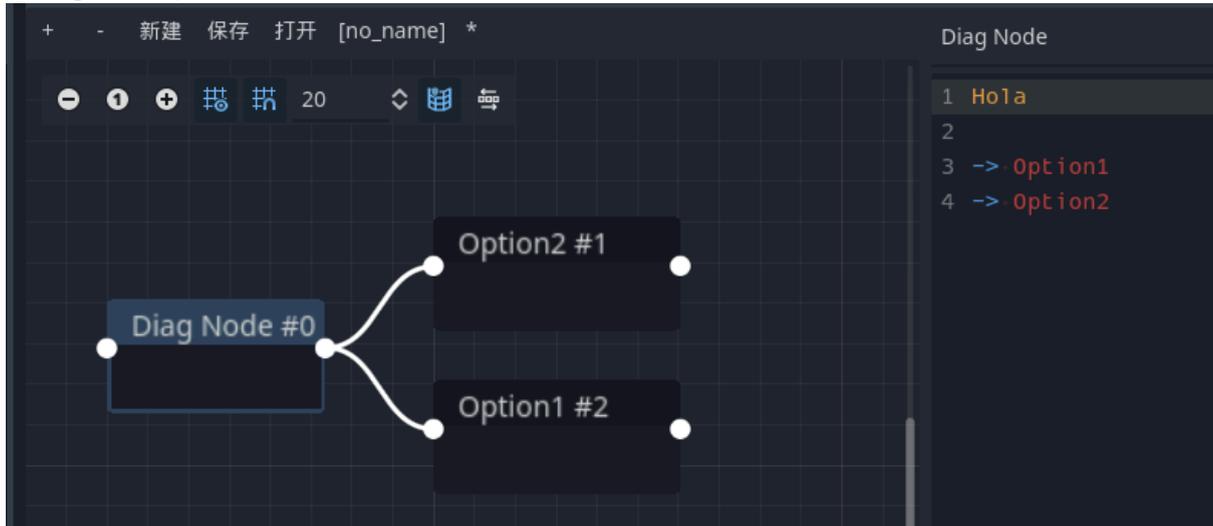
Para la configuración del sistema de diálogo, actualmente utilizamos nodos de control y botones para crear diálogos.



El cuadro de diálogo contendrá el nombre del personaje que habla, el contenido del diálogo, un cuadro de texto para las indicaciones y un fondo oscuro transparente.



A continuación, optamos por utilizar un sistema de diálogo relativamente sofisticado, y actualmente estamos planeando utilizar EZ Dialogue para crear un árbol de diálogo completo.

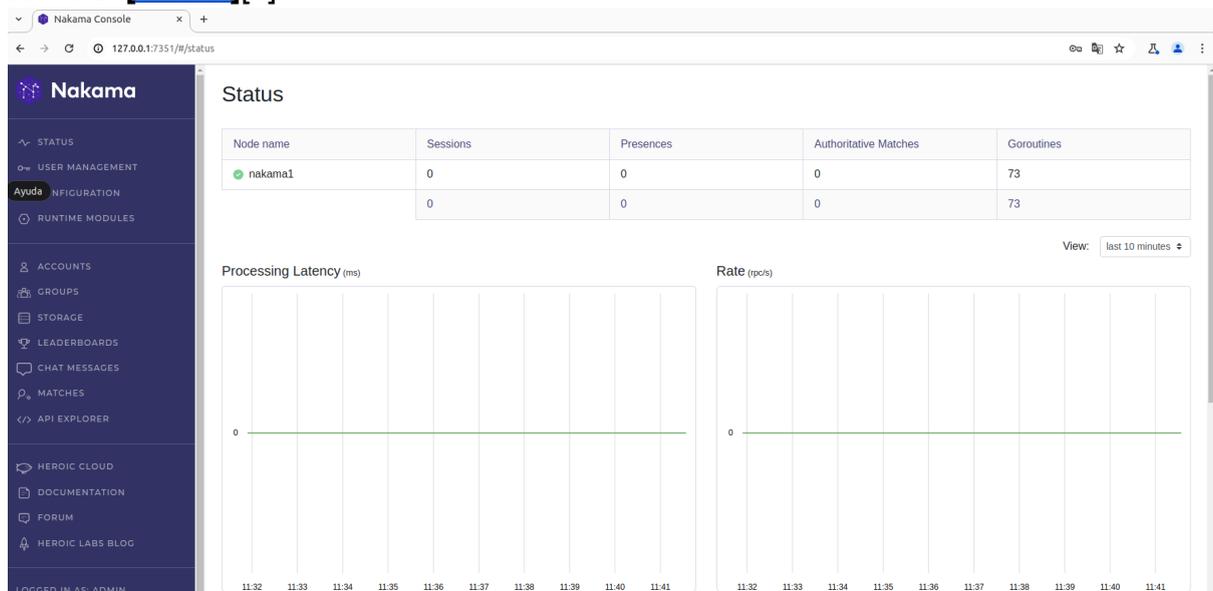


6.1.8 Multijugador

Utilizamos el servidor y el cliente nakama para configurar funciones multijugador como cuentas de usuario, perfiles sociales, chat en vivo, almacenamiento de datos y partidas multijugador.

Actualmente solo tenemos una función de red de área local incompleta que permite conectarse a jugadores en el mismo segmento de red.

Podemos ver los detalles de usuario del cliente a través de nuestro servidor. [\[Godot\]\[0\]](#)

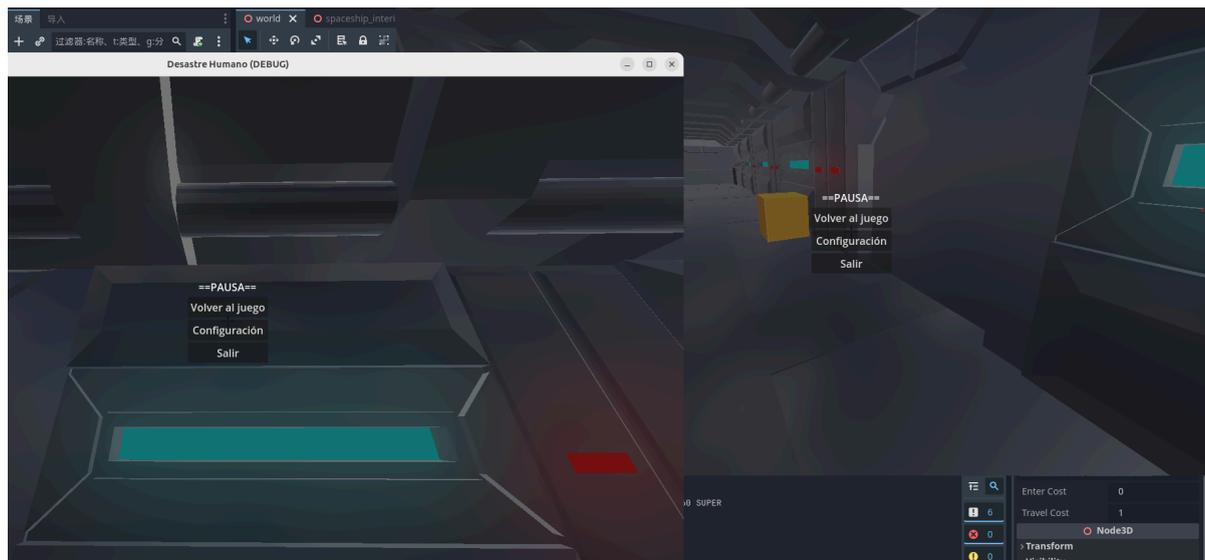


Luego use scripts para conectarse al servidor e iniciar señales punto a punto localmente para conectar a los jugadores.

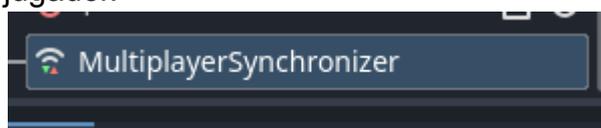
```
1  @tool
2  extends Node
3
4  # The default host address of the server.
5  const DEFAULT_HOST : String = "192.168.1.168"
6
7  # The default port number of the server.
8  const DEFAULT_PORT : int = 7350
9
10 # The default timeout for the connections.
11 const DEFAULT_TIMEOUT = 3
12
13 # The default protocol scheme for the client connection.
14 const DEFAULT_CLIENT_SCHEME : String = "http"
15
16 # The default protocol scheme for the socket connection.
17 const DEFAULT_SOCKET_SCHEME : String = "ws"
18
```

Utilice una interfaz para obtener el estado de conexión del jugador. En esta interfaz, el jugador puede elegir si desea iniciar una conexión o conectarse al mundo de otra persona, y puede ingresar nuestro nombre de usuario en el cuadro de texto.

A dark-themed user interface for registration and login. It features three input fields: 'Usuario' with the placeholder 'Escribe tu nombre de usuario', 'Email' with the value 'usuario2@elpulg.xeill.net', and 'Contraseña' with the value 'usuario2'. Below the fields are two buttons: 'Registra' and 'Acceso'.

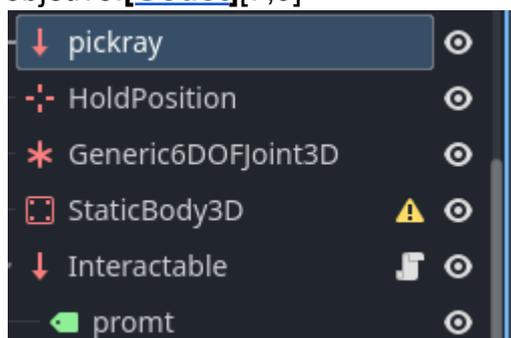


Para poder sincronizar lo que el jugador está haciendo en el mundo, tenemos que configurar todas las variables que se pueden generar para que cuando el jugador se mueva, ataque e interactúe en el mundo se sincronice con la perspectiva del otro jugador.



6.1.9 Interactivo

Establecemos varios rayos como nuestro método de juicio. Cuando los rayos colisionan con el objeto objetivo, podrán realizar interacciones específicas con el objetivo. [[Godot](#)][7,9]



Luego se utilizan diferentes guiones para mostrar diferentes interacciones con diferentes objetos.

Por ejemplo, cuando el rayo choque en un cuerpo rígido, podremos recogerlo con una tecla del ratón o del teclado y lanzarlo con otra tecla.

Cuando el objetivo con el que choca nuestro rayo esté en una clase y en el grupo objetivo, podremos interactuar con él en consecuencia. Por ejemplo, cuando el grupo objetivo es una puerta, hacemos clic en F y la puerta se abrirá.

```
func pickup_object():
    >| var collider = pickray.get_collider()
    >| if collider != null and collider is RigidBody3D:
    >| >| pick_object = collider
    >| >| joint.set_node_b(pick_object.get_path())
    # remove object
func remove_object():
    >| if pick_object != null:
    >| >| pick_object = null
    >| >| joint.set_node_b(joint.get_path())
```

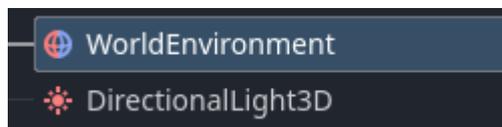
```
if Input.is_action_just_pressed("interactable"):
    >| if interactable_ray.get_collider() is Interactable:
    >| >| match weapon:
    >| >| >| weapons.MP7:
    >| >| >| >| Ammo_apply_point(weapons.MP7)
    >| >| >| weapons.PISTOL:
    >| >| >| >| Ammo_apply_point(weapons.PISTOL)
    >| >| >| >|
    >| >| # space ship open the door
    >| >| # 打算使用position的方式来实现批量开关门，不行就一个一个来。
    >| >| if interactable_ray.get_collider().is_in_group('door1'):
    >| >| >| spacheani.play("Opendoor1")
    >| >| >| await get_tree().create_timer(8.0).timeout
    >| >| >| spacheani.play_backwards("Opendoor1")
```

Cuando nuestro rayo choca con el objetivo, la información del objetivo se mostrará a través del script y se mostrarán las teclas utilizadas para interactuar.

```
func get_prompt():
    >| var key_name = ""
    >| for action in InputMap.action_get_events(prompt_action):
    >| >| if action is InputEventKey:
    >| >| >| key_name = OS.get_keycode_string(action.physical_keycode)
    >| return prompt_name + "\n\n[" + key_name + "]"
```

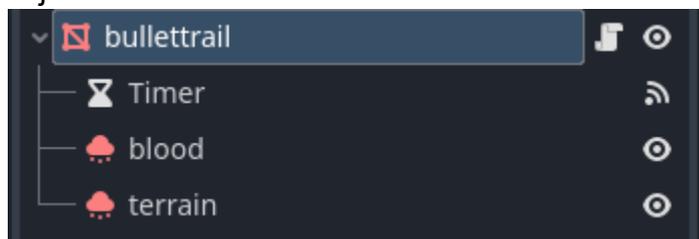
6.1.10 Fondo del cielo mundial

El fondo del cielo de nuestro mundo se forma utilizando una imagen HDR y diferentes imágenes pueden producir diferentes efectos.

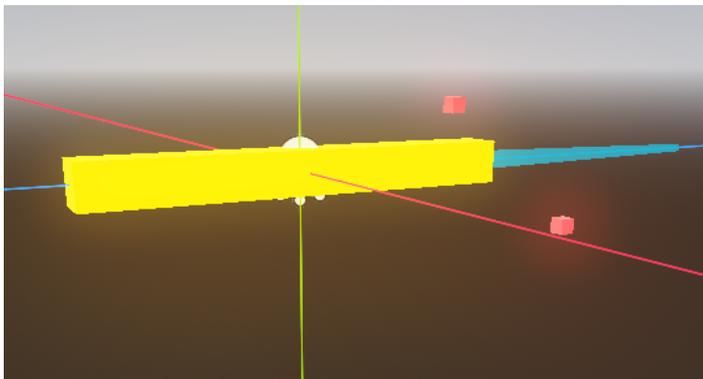


6.1.11 Partículas de GPU

Los efectos de partículas de nuestros ataques con armas utilizan partículas de GPU, que se pueden personalizar en la barra de propiedades para lograr el efecto objetivo.



Por ejemplo, en nuestro juego tenemos dos efectos diferentes de ataque con arma de fuego. La imagen de abajo es uno de ellos, cuando nuestra arma ataca, envía balas y crea un efecto de partículas.



También es posible utilizar scripts para instanciar los efectos de partículas que se han configurado. Por ejemplo, en el script de abajo, el efecto de partículas para la sangre se instanciará cuando nuestras balas golpeen al enemigo, de lo contrario el efecto de partículas se generará cuando las balas golpeen el suelo.

```
func trigger_particles(pos, gun_pos, on_enemy):  
  if on_enemy:  
    blood.position = pos  
    blood.look_at(gun_pos)  
    blood.emitting = true  
  else:  
    terrain.position = pos  
    terrain.look_at(gun_pos)  
    terrain.emitting = true
```


Aquí se utilizan dos recursos diferentes para que llamemos y cambiemos, creando diccionarios para los datos, almacenando los datos cambiados como diccionarios cuando son señalados por otros scripts y luego volviendo a enlazar los datos de la forma correcta.

```
extends Node

# 设置数值默认值
@onready var DEFAULT_SETTING : DefaultSettingResource = preload("res://resource/settings/DefaultSe
@onready var KEYBIND_RESOURCE : KeyBindResource = preload("res://resource/settings/KeyBindDefault.

var window_mode_index = 0
var resolution_mode_index = 0
var master_volume : float = 0.0
var music_volume : float = 0.0
var sfx_volume : float = 0.0
var subtitle_set = false

var loaded_data : Dictionary = {}

func _ready():
    > handled_signals()
    > create_storage_dictionary()
```

Define las señales para los datos aquí, y luego envía esos datos como señales a otros lugares para ser llamados.

```
extends Node

signal on_subtitles_toggled(value : bool)

signal on_window_mode_selected(index : int)
signal on_resolution_mode_selected(index : int)

signal on_master_sound_set(value : float)
signal on_music_sound_set(value : float)
signal on_sfx_sound_set(value : float)

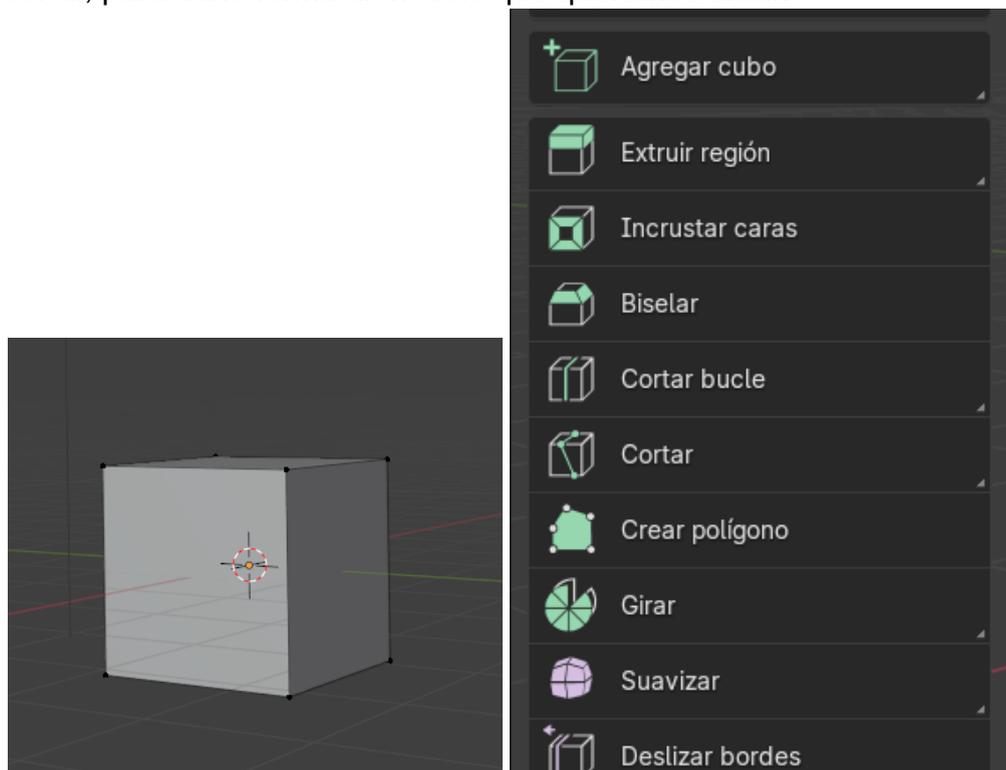
signal set_setting_dictionary(setting_dict : Dictionary)

signal load_setting_data(setting_dict : Dictionary)
```

6.2 Proceso de implementación en Blender

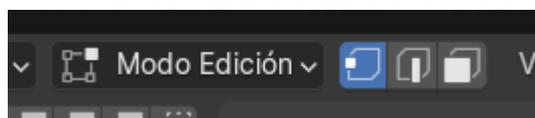
6.2.1 Modelado

Al comenzar con el modelado, normalmente empezamos con un cubo. Puedes añadir un cubo usando el atajo **Shift + A** y seleccionando "Malla" en el menú. Después, ingresamos al modo de edición presionando **Tab**. En el modo de edición, utilizamos las herramientas de la barra lateral izquierda, como extruir, mover y cortar, para darle forma al modelo que queremos crear.

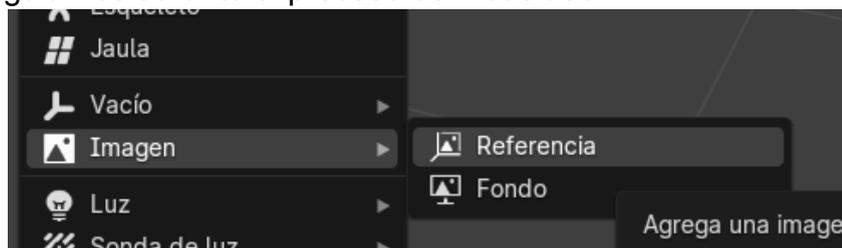


En el modo de edición, hay tres tipos de selección: modo vértice, modo arista y modo cara. Estos modos se pueden alternar dependiendo de cómo queramos manipular el modelo:

- Modo vértice: Permite seleccionar y mover puntos individuales.
- Modo arista: Permite seleccionar y mover bordes del modelo.
- Modo cara: Permite seleccionar y manipular caras enteras del modelo.



También podemos añadir una imagen de referencia usando el atajo Shift + A para guiarnos durante el proceso de modelado.



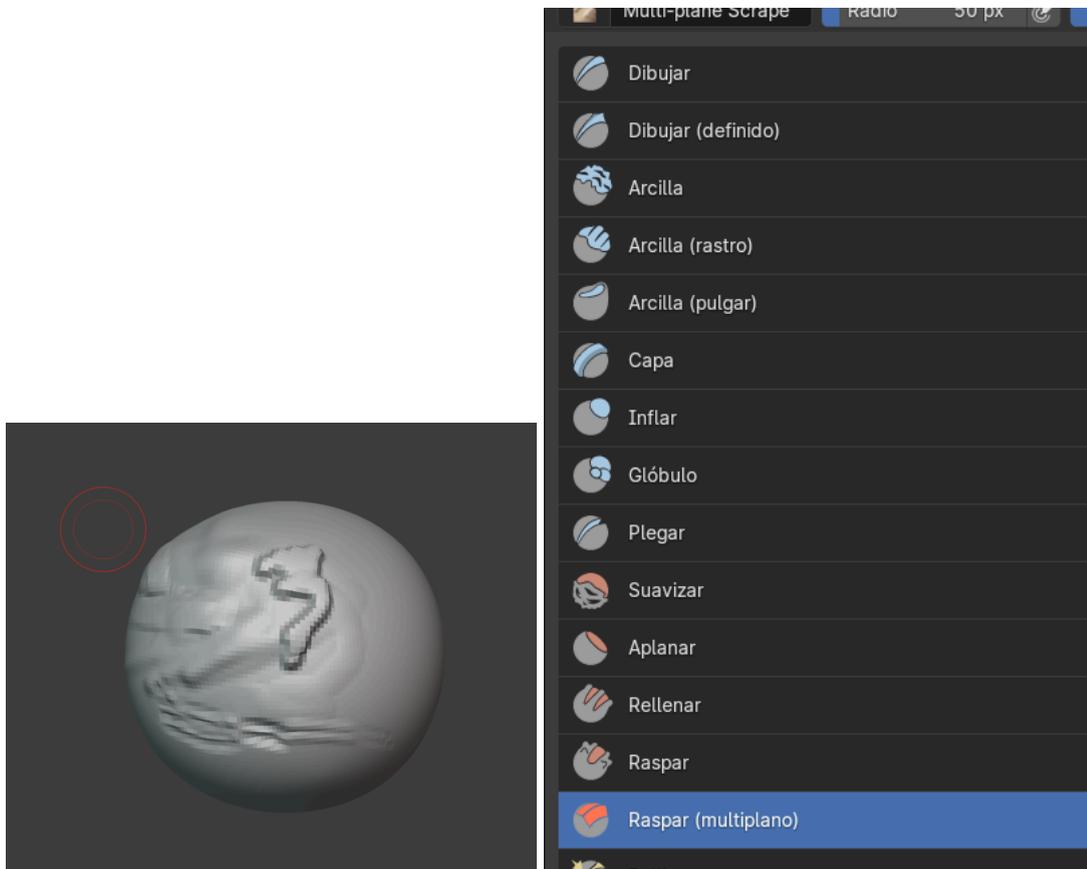
6.2.2 Esculpido

Blender tiene muchas herramientas de esculpido que nos permiten añadir detalles finos a nuestros modelos. Antes de usar estas herramientas, es crucial asegurarnos de que nuestro modelo tenga suficientes caras. Si el modelo es de baja resolución, las herramientas de esculpido no funcionarán bien.

Para aumentar la cantidad de caras, podemos aplicar un modificador de "**subdivisión de superficie**". Este modificador divide cada cara del modelo en partes más pequeñas, haciendo que el modelo sea más suave y detallado.

En el modo de esculpido, tenemos acceso a una variedad de pinceles que nos permiten realizar diferentes tipos de esculpido, como:

- **Dibujar:** Añade volumen al modelo.
- **Suavizar:** Suaviza la superficie del modelo.
- **Agarrar:** Permite agarrar y mover partes del modelo.

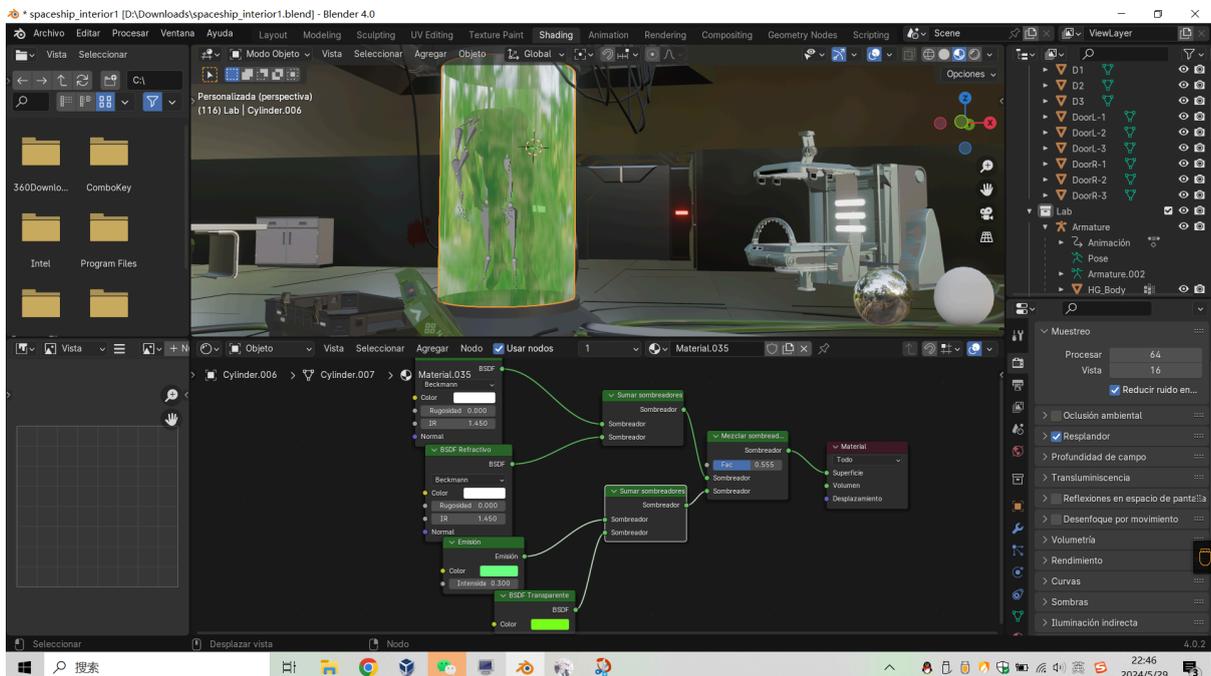


6.2.3 Sombreado

El sombreado en Blender implica la creación de materiales y texturas que se aplican a la superficie de los modelos 3D. En el Editor de Sombreado, podemos crear materiales complejos conectando diferentes nodos. Algunos de los nodos más utilizados incluyen:

- **Principled BSDF:** Un nodo versátil que combina varios tipos de sombreado en uno solo.
- **Image Texture:** Permite aplicar una imagen como textura en el modelo.
- **Mix Shader:** Combina dos shaders diferentes.

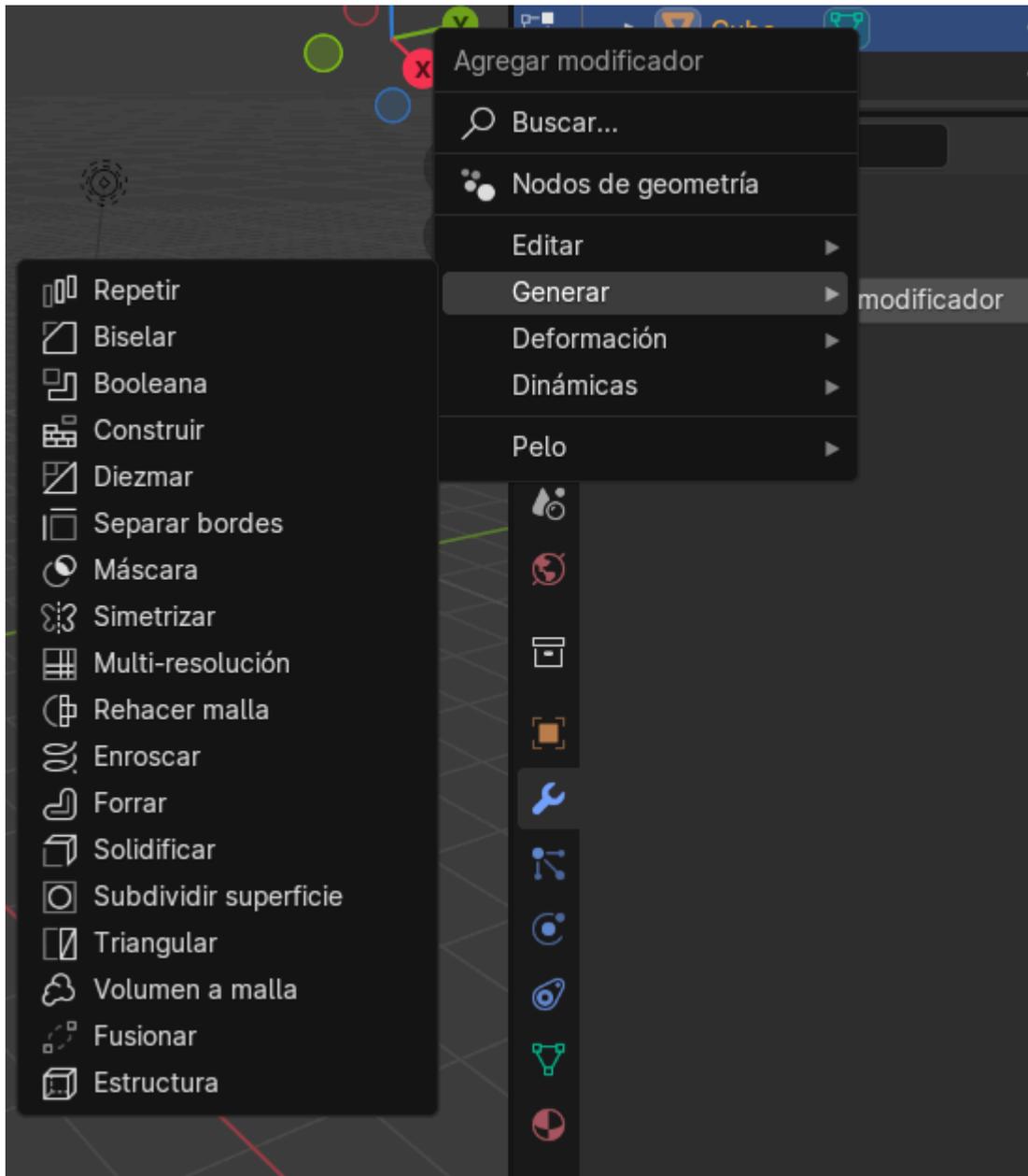
Mediante la conexión de estos nodos, podemos definir cómo se verá la superficie del modelo bajo diferentes condiciones de iluminación. Este proceso nos permite crear materiales realistas y detallados.



6.2.4 Modificador

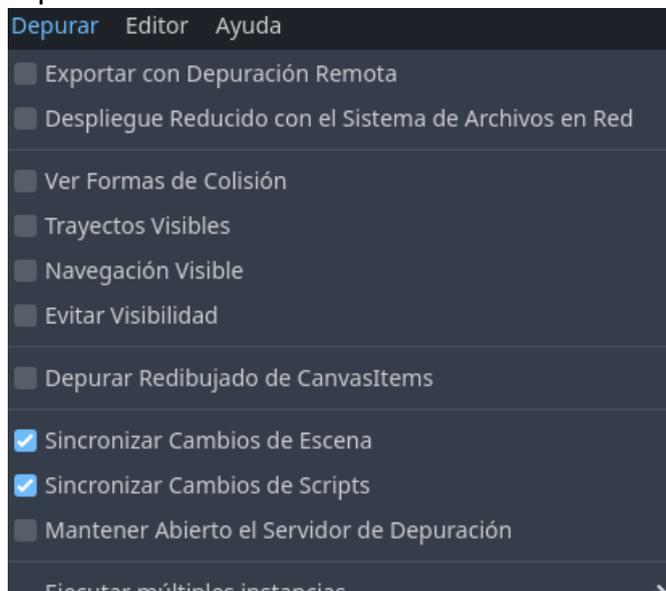
Los modificadores en Blender son herramientas que nos ayudan a modificar y optimizar nuestros modelos de manera no destructiva. Algunos de los modificadores más útiles incluyen:

- **Espejo:** Crea una copia simétrica del modelo a lo largo de un eje. Para modelar objetos simétricos, como personajes o vehículos, ya que solo necesitamos modelar una mitad y el modificador se encargará de la otra.
- **Subdivisión de superficie:** Añade más subdivisiones a la geometría del modelo, haciéndolo más suave y detallado.
- **Repetir:** Duplica el modelo en una matriz, sirve para crear objetos repetitivos como columnas o ruedas.

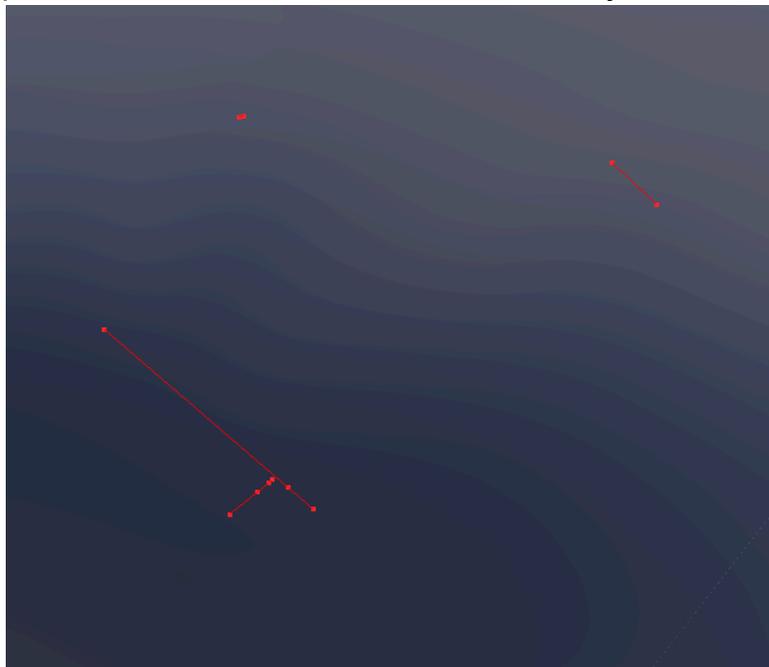


7. Pruebas y ejecución

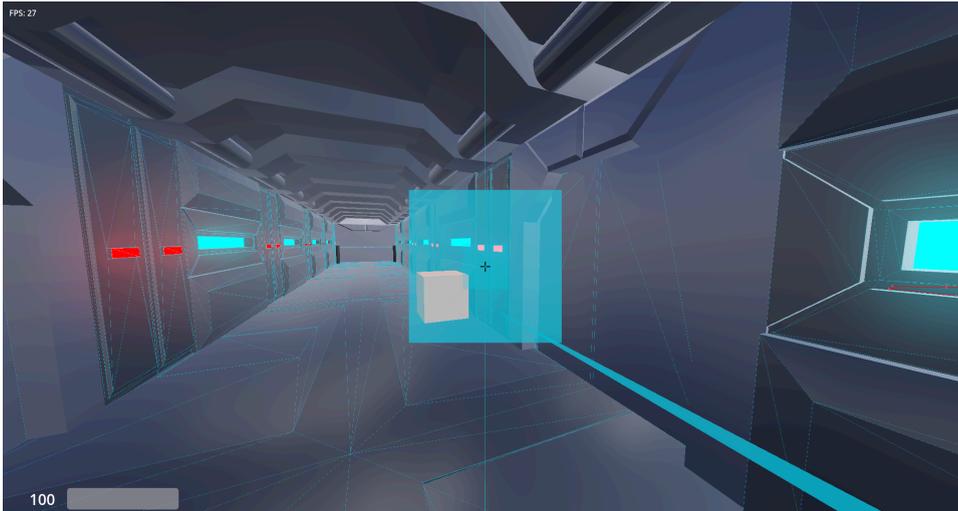
Al probar el juego, si nos encontramos con algunos problemas relacionados con las siguientes melodías, podemos utilizar el modo de ajuste para ver cuál es el error específico.



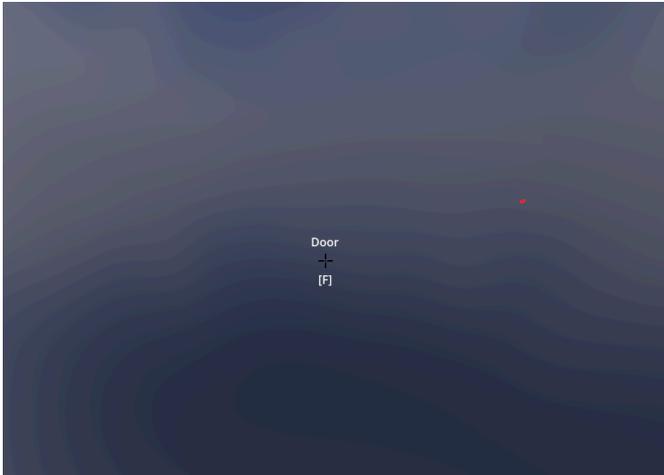
Por ejemplo, podemos comprobar si el enemigo navega correctamente. Los puntos rojos y las líneas en la imagen de abajo son los caminos del enemigo y nosotros, así que si el enemigo no está siguiendo la línea designada hacia nosotros entonces puede haber un error de horneado de la rejilla de navegación.



El rayo azul en la imagen de abajo es nuestro rayo colidable, a veces el rayo no funciona como esperábamos, puede ser porque no le dimos la longitud correcta o lo colocamos incorrectamente, podemos usar el modo debug para ver la razón exacta.



Comprueba si una puerta interactiva se abre cuando pulsamos el botón interactivo y se cierra tras un periodo de tiempo.



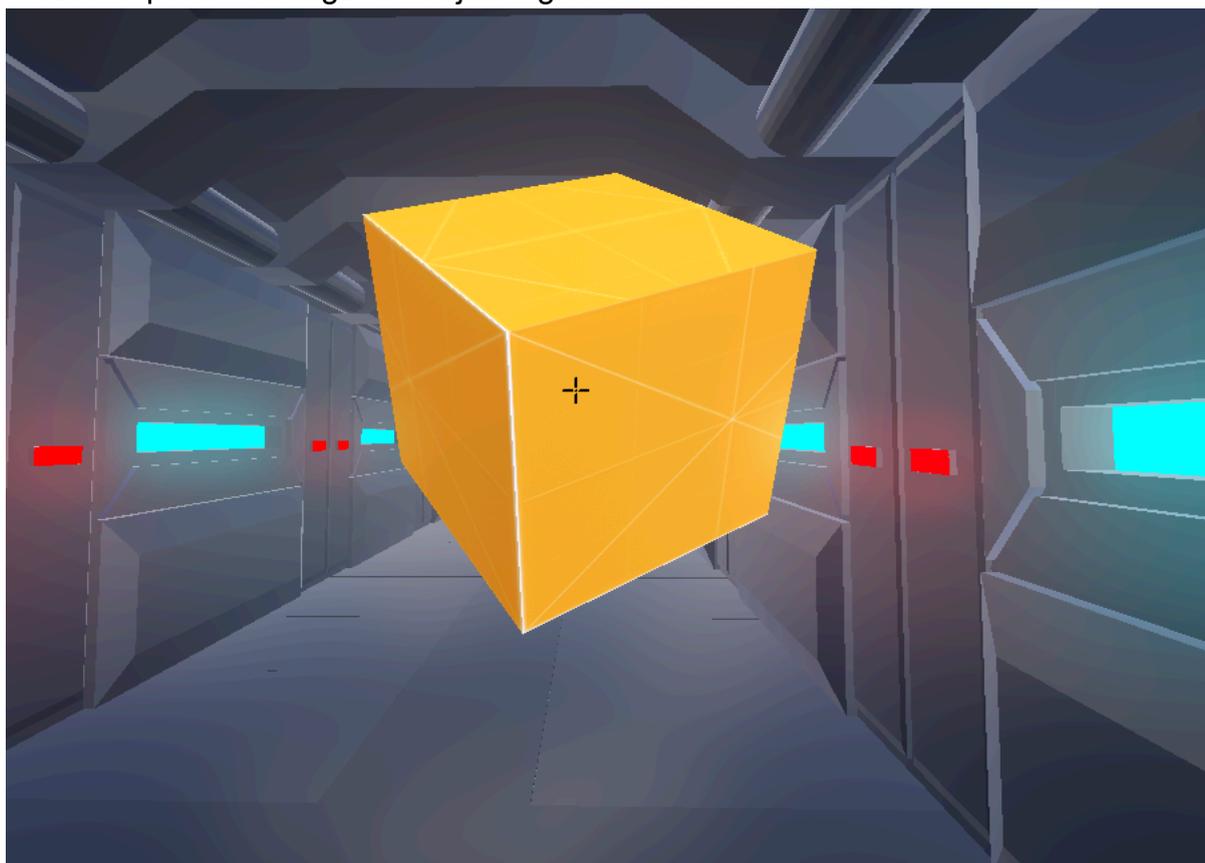
Probar si cuando nos quedamos sin munición para nuestras armas, al llegar a la estación de recarga de munición y pulsar el botón interactivo, si se recarga la munición.



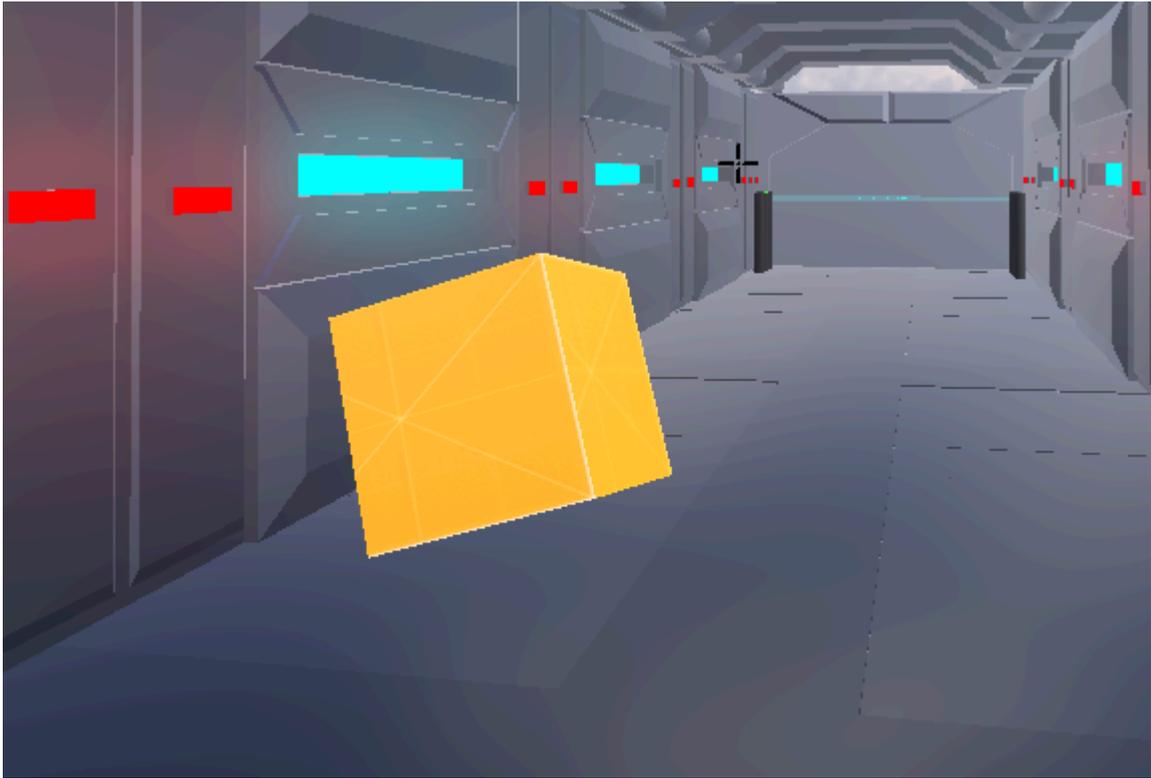
Comprueba si las balas infligen daño a los enemigos y si la animación de muerte se activa cuando el nivel de sangre del enemigo llega a cero.



Prueba si podemos coger un objeto rígido al hacer clic sobre él.



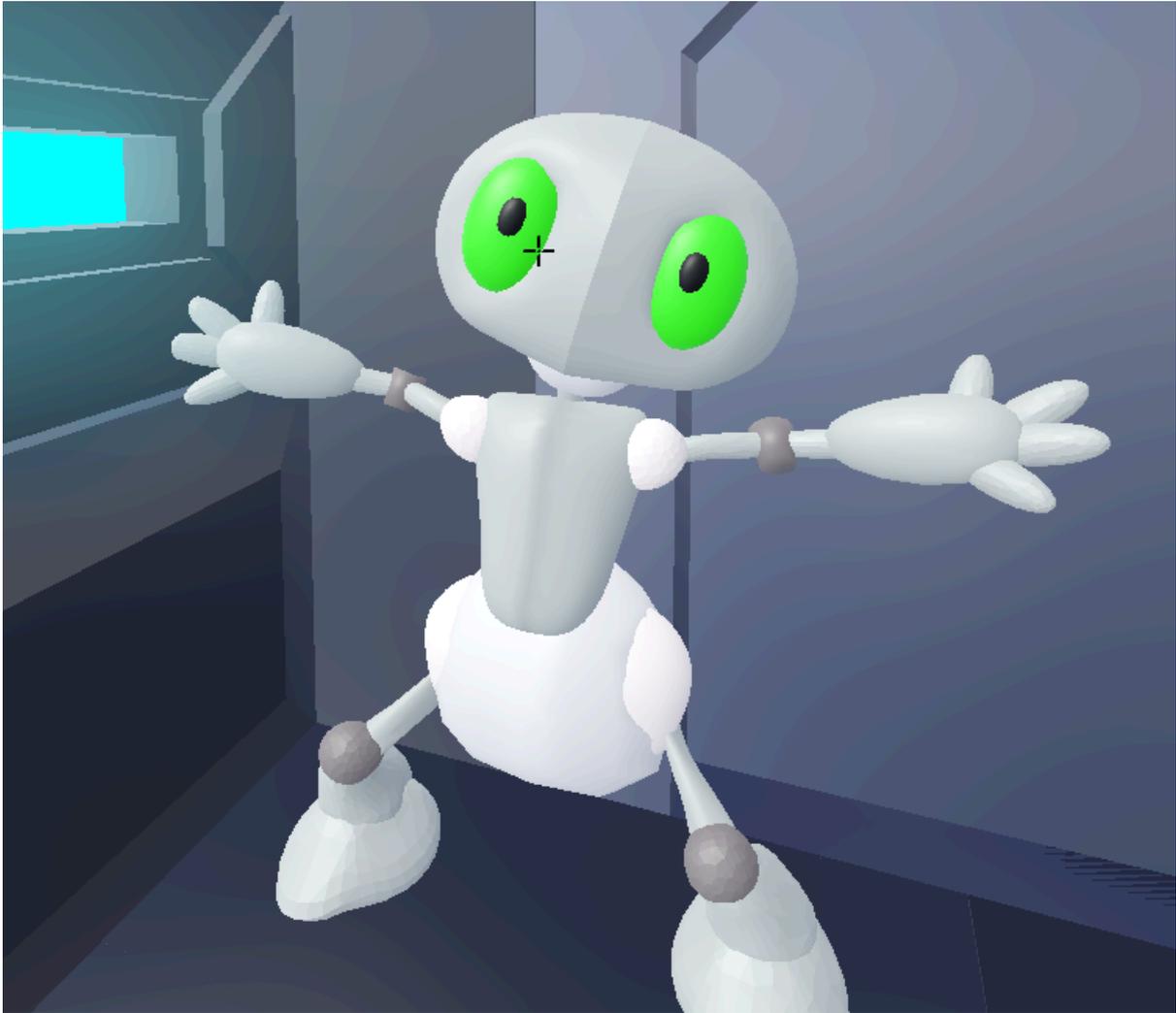
Prueba si podemos lanzar un objeto rígido cuando lo cogemos y luego hacemos clic con el botón derecho del ratón.



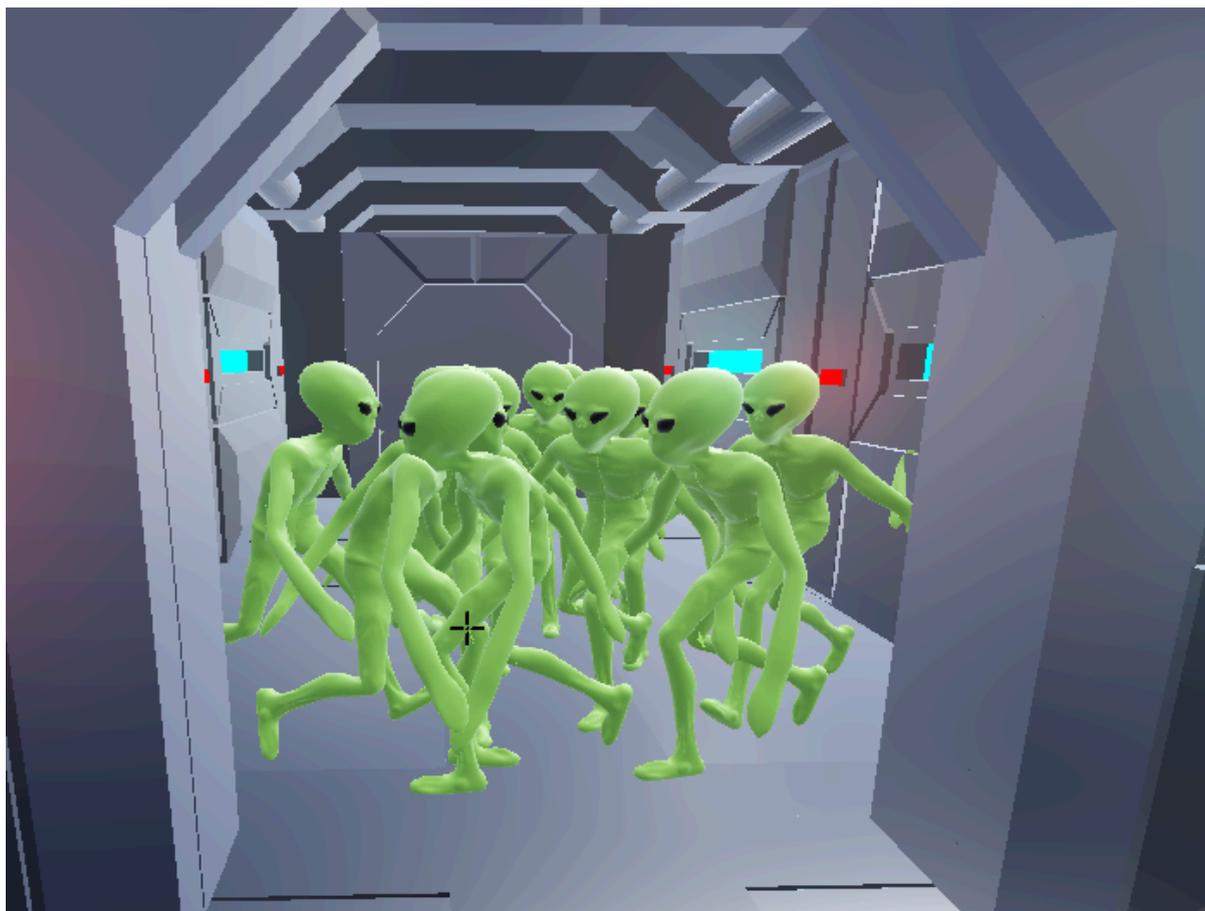
Comprueba si un robot NPC puede pulsar botones y entablar diálogos con la tecla de acceso directo "F".



Comprueba si el NPC robot realizará una animación de espera cada treinta segundos.



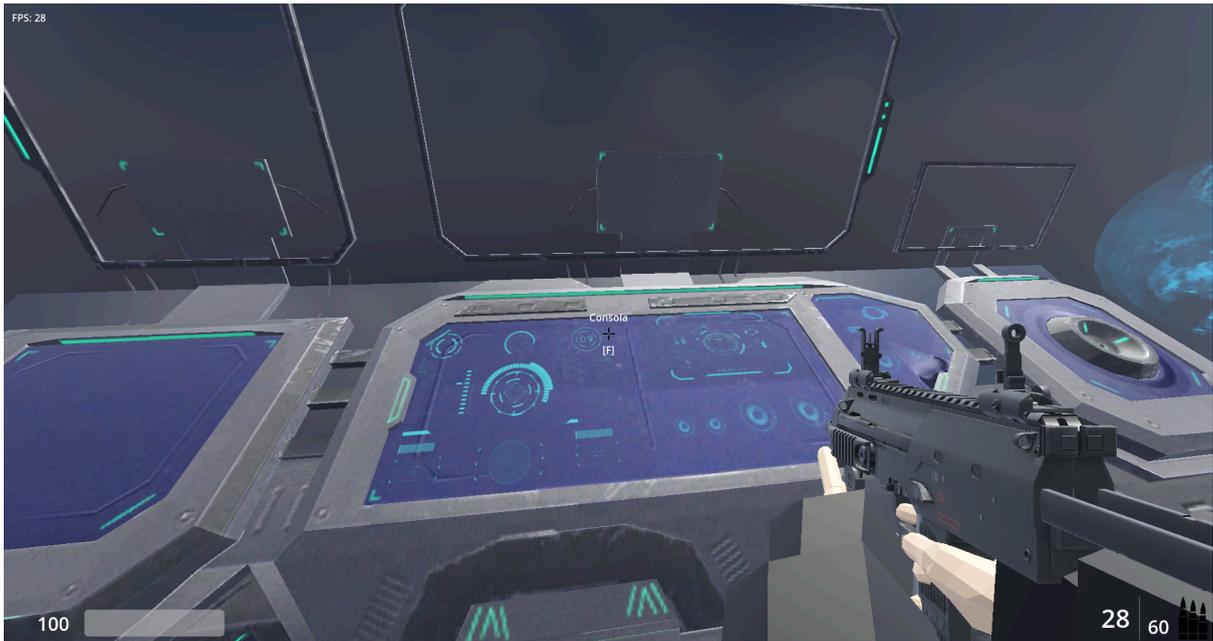
Prueba a ver si el enemigo no se genera automáticamente durante 60 segundos.



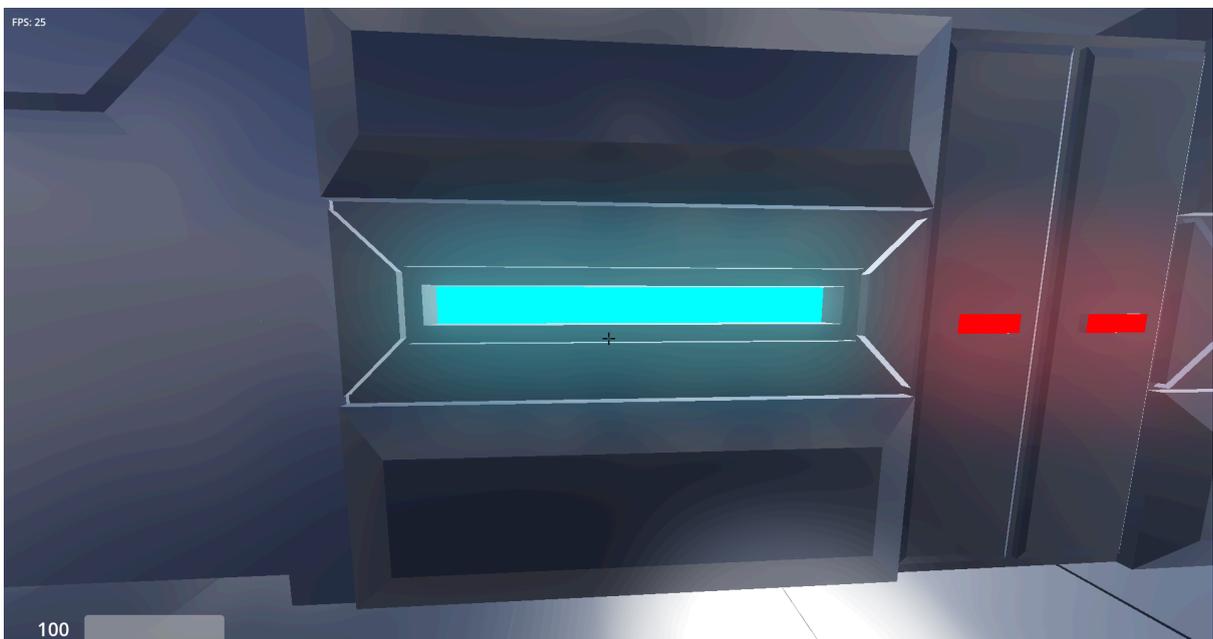
Comprueba si nuestro nivel de sangre disminuye cuando nos ataca un enemigo y si volvemos a la página principal cuando el nivel de sangre es cero.



Comprueba si somos transportados a otra escena después de interactuar con la consola.



Prueba a ver si puedes interactuar con los botones de la escena. El botón sirve para teletransportarse a la nave espacial.



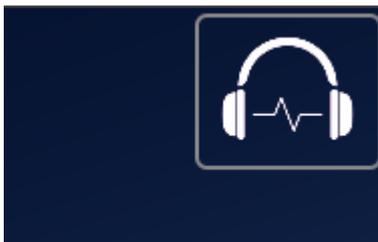
Comprueba si las balas del arma serán reemplazadas.



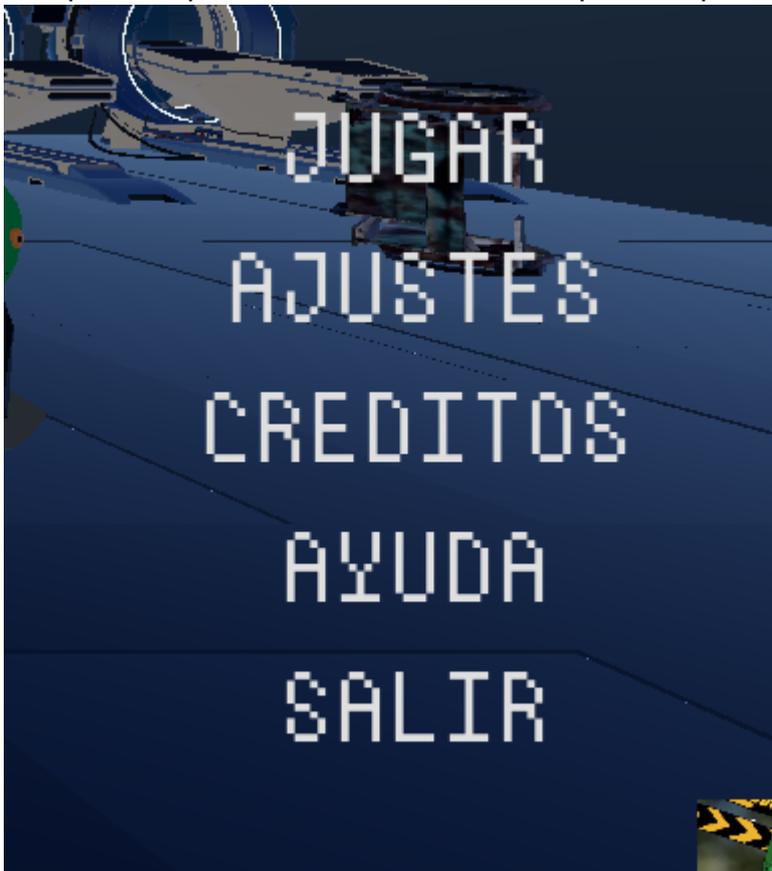
Comprueba si se cambia de arma cuando se pulsa el botón para cambiar de arma.



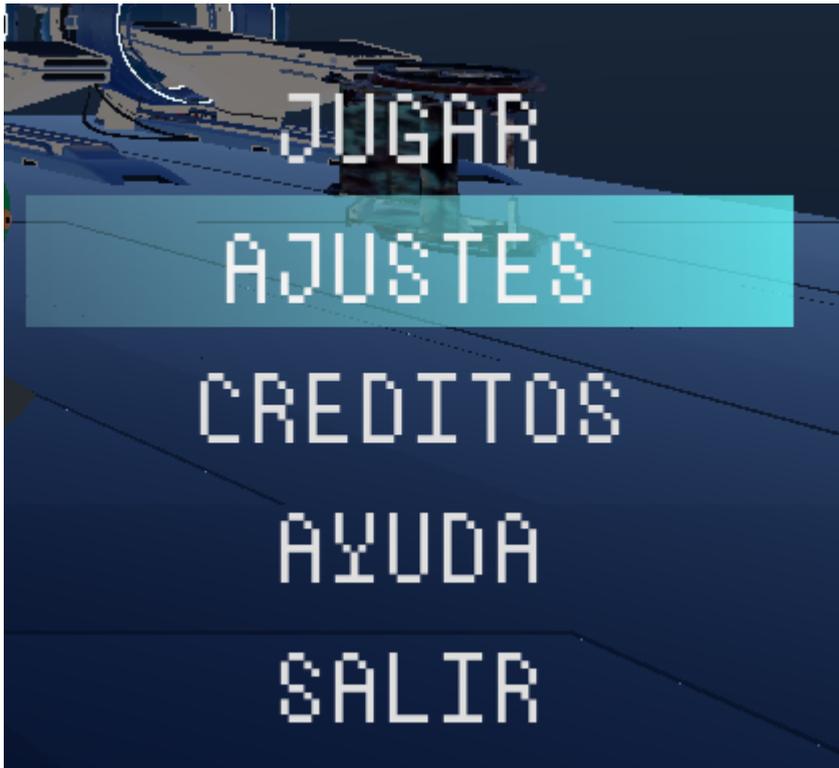
Comprueba que el interruptor de música de fondo de la interfaz principal funciona correctamente.



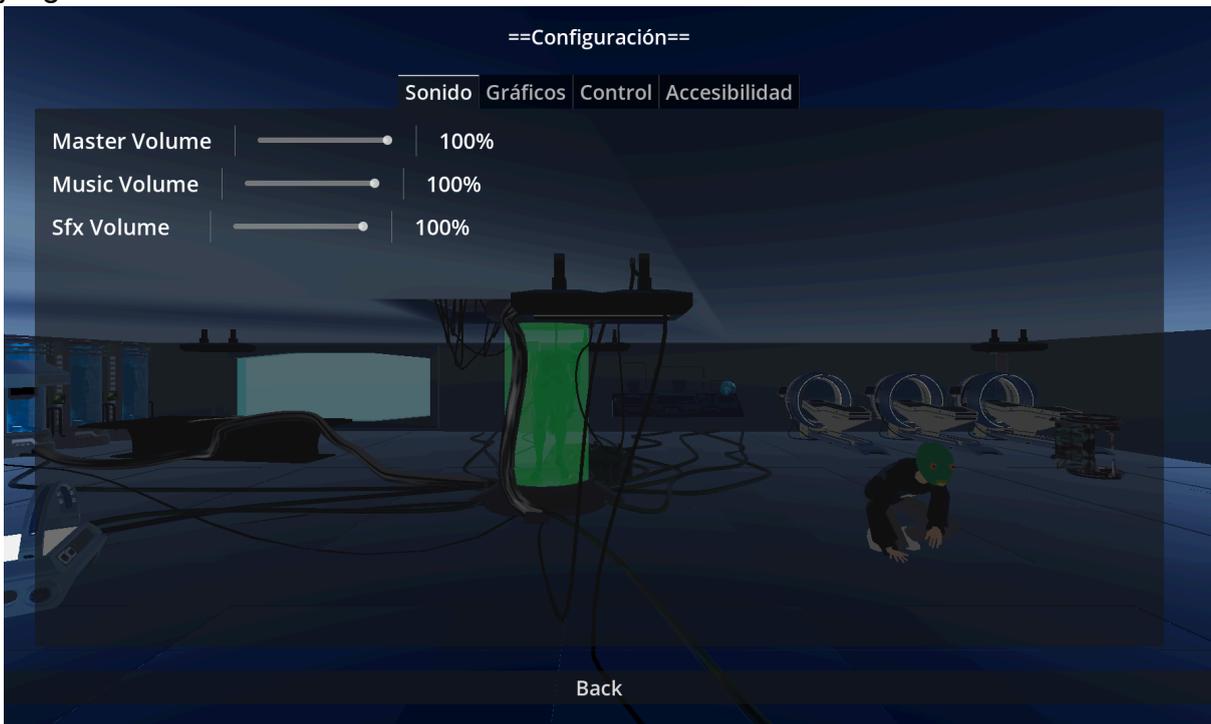
Compruebe que todos los botones de la pantalla principal funcionan correctamente.



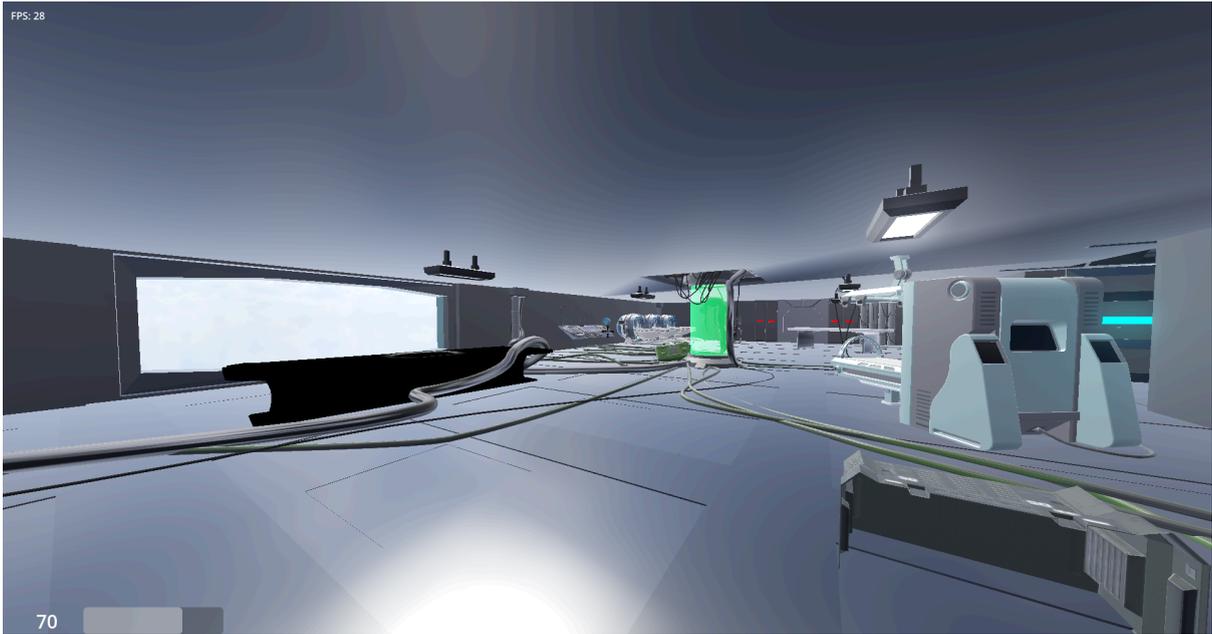
Comprueba si se activa un efecto de sonido cuando el ratón pasa por encima de un botón.



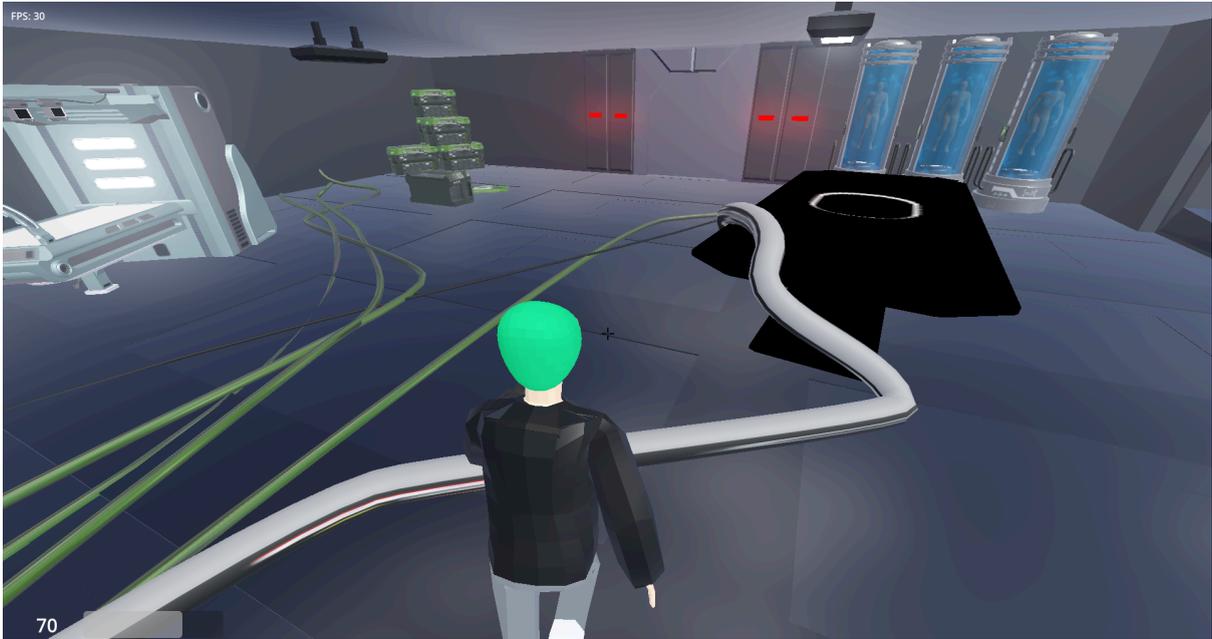
Prueba a ver si todos los ajustes son correctos cambiando la configuración del juego.



Prueba si se han aplicado correctamente las colisiones de la escena, así como las texturas.



Comprueba que la animación del personaje funciona correctamente.



8. Sitio web y servidores

8.1 Nuestro servidor web (InfinityFree)

Nuestro servidor web [3], alojado en InfinityFree, ofrece un servicio de alojamiento gratuito con un excelente rendimiento para proyectos iniciales. Esta plataforma nos permite mantener una presencia online sin costes iniciales elevados, con la opción de hacer escalado hacia opciones de pago a medida que aumente la demanda de nuestro juego. Además, proporciona subdominios, y en nuestro caso, elegimos un subdominio de wuaze.com.

Básicamente, podemos supervisar el estado del servidor web, SSL, MySQL, registros DNS, y otras funcionalidades.

The screenshot shows the InfinityFree account control panel. At the top, it displays the account ID 'if0_36415836 (Proyecto de Desastre Humano)' and an 'Account Settings' button. Below this are three main action buttons: 'Control Panel' (green), 'File Manager' (orange), and 'Softaculous Installer' (purple). The main content area is divided into three columns:

- Account Options:** Upgrade to Premium, Statistics, Domains, FTP Details, MySQL Databases, Deactivation History, Account Settings.
- Domain and Website Options:** DNS Records, SSL/TLS, Redirects, Protected Directories, Error Pages, PHP Options, IP Blocks.
- Account Details:** USERNAME: if0_36415836; PASSWORD: [masked]; STATUS: Active; LABEL: Proyecto de Desastre Humano; WEBSITE IP: 185.27.134.176; HOSTING VOLUME: vol1_4; HOME DIRECTORY: /home/vol1_4/infinityfree.com/if0_36415836; CREATION DATE: 2024-04-23.

Nuestro proyecto ha aprovechado las ventajas de InfinityFree como plataforma de alojamiento web, estableciendo una infraestructura en línea eficiente y económica. En primer lugar, la gestión de DNS proporcionada por InfinityFree nos permite configurar y modificar los registros del sistema de nombres de dominio de forma sencilla. Aunque, como plano gratuito, solo podemos editar MX, SPF y CNAME, sin la capacidad de editar directamente A y AAAA, esto ha sido suficiente para asociar nuestro servidor con nuestro servidor de correo electrónico Zoho Mail para la verificación.

ACCOUNTS / IF0_36415836 / DNS RECORDS / DESAHUMAN.WUAZE.COM

DNS Records for desahuman.wuaze.com on if0_36415836

Account Settings

Manage if0_36415836

- Account Options
 - Home
 - Upgrade to Premium
 - Statistics
 - Domains**
 - FTP Details
 - MySQL Databases
 - Deactivation History
 - Account Settings
- Domain and Website Options

DNS Records

+ Add Record

DOMAIN	TYPE	TARGET	ACTIONS
desahuman.wuaze.com	A	185.27.134.176	
www.desahuman.wuaze.com	A	185.27.134.176	
._acme-challenge.desahuman.wuaze.com	CNAME	bce67728-c557-442a-a988-84c2f46ba5c9.acmedns.infinityfree.net	Delete
zb28586148.desahuman.desahuman.wuaze.com	CNAME	zmverify.zoho.eu	Delete
tisovfm7omnb.desahuman.wuaze.com	CNAME	gv-w7nyay4k7bz5lu.dv.googlehosted.com	Delete
5jon36xpbsrj.desahuman.wuaze.com	CNAME	gv-dtjou4glrgsxbv.dv.googlehosted.com	Delete
lzt0on4puh5e.desahuman.wuaze.com	CNAME	gv-mjxnmkoculqwy.dv.googlehosted.com	Delete
desahuman.wuaze.com	MX	10 mx.zoho.eu	Delete
desahuman.wuaze.com	MX	20 mx2.zoho.eu	Delete
desahuman.wuaze.com	MX	50 mx3.zoho.eu	Delete
desahuman.wuaze.com	TXT	v=spf1 include:zohomail.eu ~all	Delete

Aunque InfinityFree en sí mismo no ofrece certificados SSL, ofrece soporte a proveedores como Zero SSL y Let 's Encrypt para ayudarnos a implementar cifrado HTTPS gratuito, garantizando la seguridad de las interacciones en nuestro sitio web. Esto es crucial para proteger la información de nuestros usuarios y, en el entorno de Internet actual, los motores de búsqueda dan cada vez mayor preferencia a los sitios web cifrados.

Para ello, hemos activado un certificado SSL para nuestro sitio web a través de InfinityFree. Durante este proceso, tuvimos que añadir un registro CNAME mediante los registros DNS, y una vez el certificado estuvo activo, recibimos una clave pública y una clave privada, que debemos introducir y subir para activar el certificado. Estas pueden ser verificadas por herramientas externas como SSL Checker.

InfinityFree Go Premium wchen@elf

Home Profile Accounts **Free SSL Certificates** Site Builders Domain Checker Knowledge Base Community Forum

FREE SSL CERTIFICATES

SSL Certificates

All Pending Issued Expired Revoked Failed

DOMAIN	PROVIDER	STATUS	EXPIRES AT	ACTIONS
desahuman.wuaze.com	Google Trust	Issued	2024-07-23	Manage

+ New SSL Certificate Showing 1 to 1 of 1 results 1

Don't like manual installation?

iFastNet's Premium Hosting integrates Let's Encrypt directly into the cPanel control panel, so your free SSL certificates can be created, installed and renewed fully automatically!

Learn more about this and the other benefits of Premium Hosting from iFastNet!

Learn More

FREE SSL CERTIFICATES / DESAHUMAN.WUAZE.COM FROM GOOGLE TRUST

SSL certificate for desahuman.wuaze.com

雨爱 - 杨丞琳

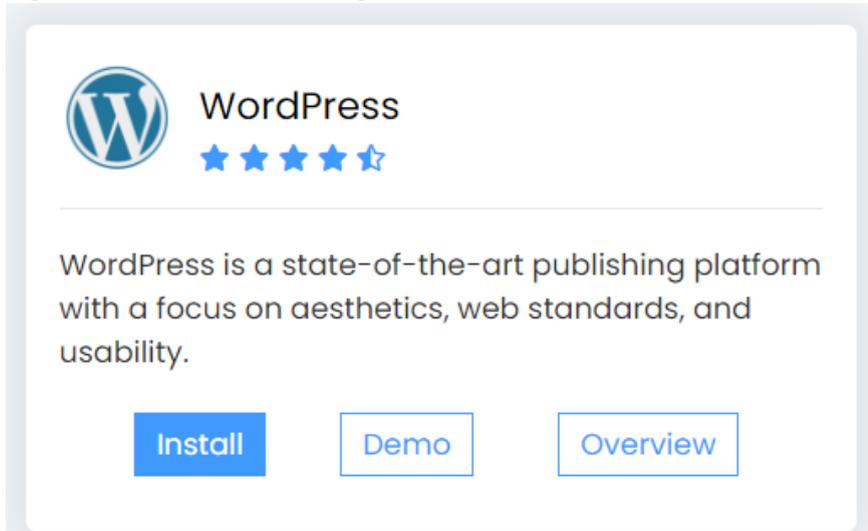
Step 4: Install SSL certificate on desahuman.wuaze.com

DOMAIN	CERTIFICATE PROVIDER	STATUS	CREATED AT	ISSUE DATE
desahuman.wuaze.com	Google Trust	Issued	2024-04-23 12:25 UTC	2024-04-24 UTC

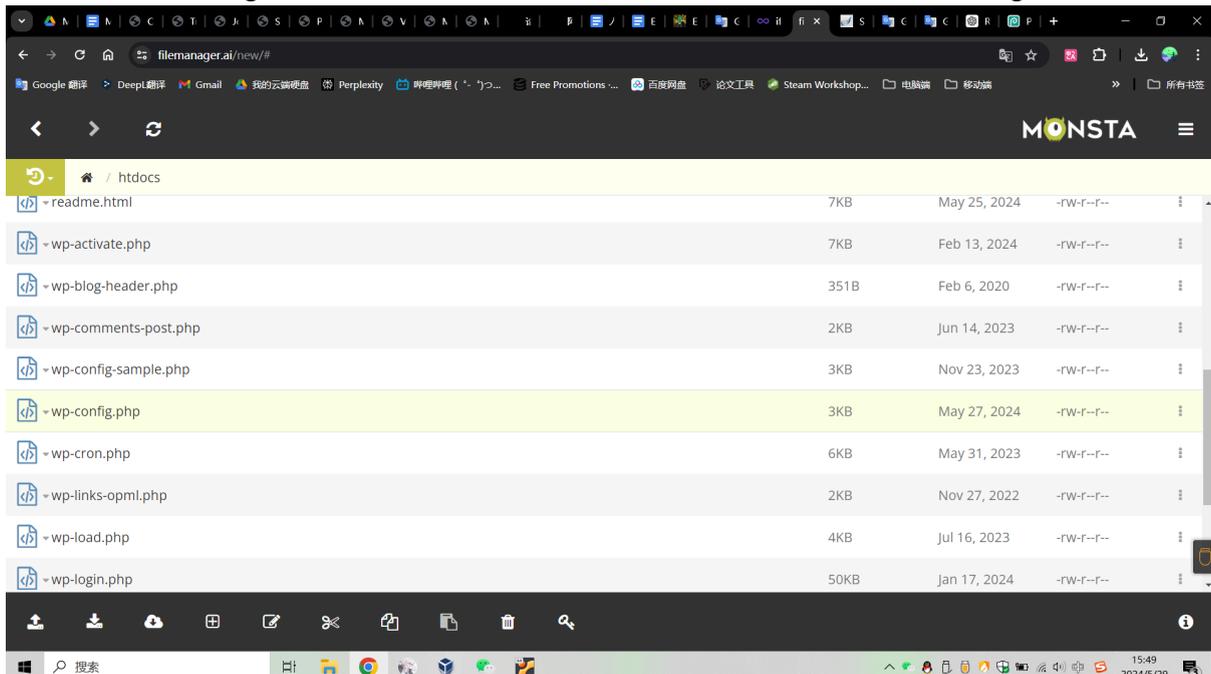
EXPIRATION DATE: 2024-07-23 UTC

SEND EXPIRATION REMINDERS:

Al usar este servidor de alojamiento web, necesitamos usar Softaculous Installer para instalar WordPress. Al instalar WordPress, no es necesario editar ninguna configuración de la base de datos, solo debemos ingresar los datos del administrador de WordPress. Una vez instalado, podemos acceder a WordPress ingresando el dominio registrado en el servidor en la barra de búsqueda.



Además, después de instalar WordPress, en el panel de InfinityFree, el administrador de archivos nos permite editar carpetas y archivos, incluyendo archivos de configuración de WordPress como el archivo PHP de configuración.



También ofrece el ControlPanel o cPanel, que tiene algunas opciones adicionales para la gestión. Aunque estas opciones también se pueden configurar directamente en InfinityFree, dado que estamos usando el plan gratuito, la cantidad de configuraciones disponibles es limitada.

STATISTICS	
Plan:	infinityfree
FTP accounts:	1 / 1
Sub-Domains:	0 / Unlimited
Add-on Domains:	1 / Unlimited
Parked Domains:	0 / Unlimited
MySQL Databases:	0 / Unlimited
Disk Quota:	5 GB
Disk Space Used:	233 MB
Disk Free:	4752 MB
Inodes Used:	19 % (11351 of 59400)
Bandwidth:	Unlimited
Bandwidth used:	503 MB
Bandwidth remaining:	Unlimited
Daily Hits Used:	1 % (339 of 50000)

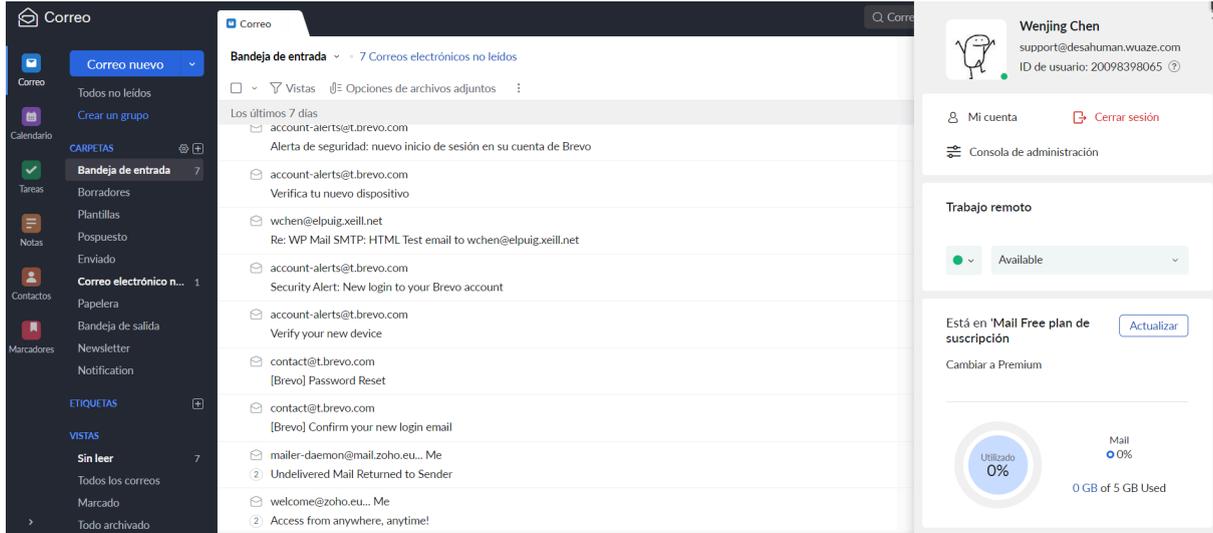
8.2 Nuestro servidor de correo externo

En cuanto a nuestro servidor de correo electrónico, dada la situación en la que los servidores locales no pueden comunicarse directamente con los servidores alojados en InfinityFree, hemos decidido optar por Zoho Mail. Zoho Mail es una aplicación dentro del ecosistema de Zoho que nos permite gestionar el correo electrónico de la empresa.

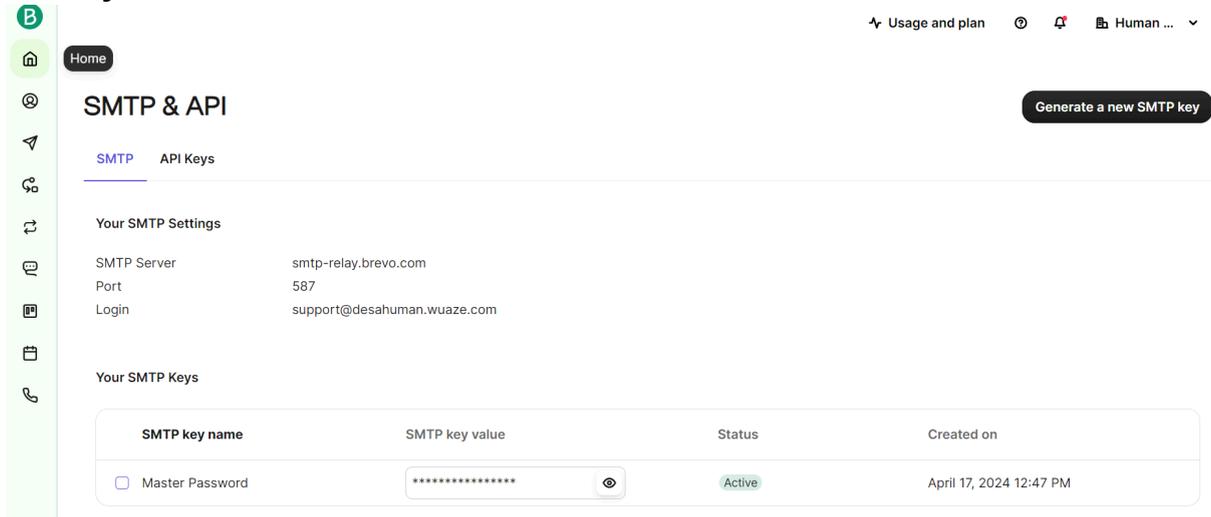
Hemos usado nuestra cuenta gmail del puig para crear una cuenta de Zoho mail y registramos un correo electrónico exclusivo con nuestro propio dominio aquí y lo hemos utilizado para registrarnos en Brevo, teniendo en cuenta sus API para prepararnos para el futuro desarrollo de nuestro sitio web de WordPress. A través de este servicio, podemos enviar mensajes masivos a todos los suscriptores para mantenerlos informados sobre actualizaciones, nuevos contenidos y eventos especiales.

Por lo tanto, nuestro servidor de comunicación externo estará en Zoho Mail y este correo electrónico se utilizará para comunicarnos con los clientes, enviar suscripciones y recibir información de fuentes externas.

Zoho Mail:

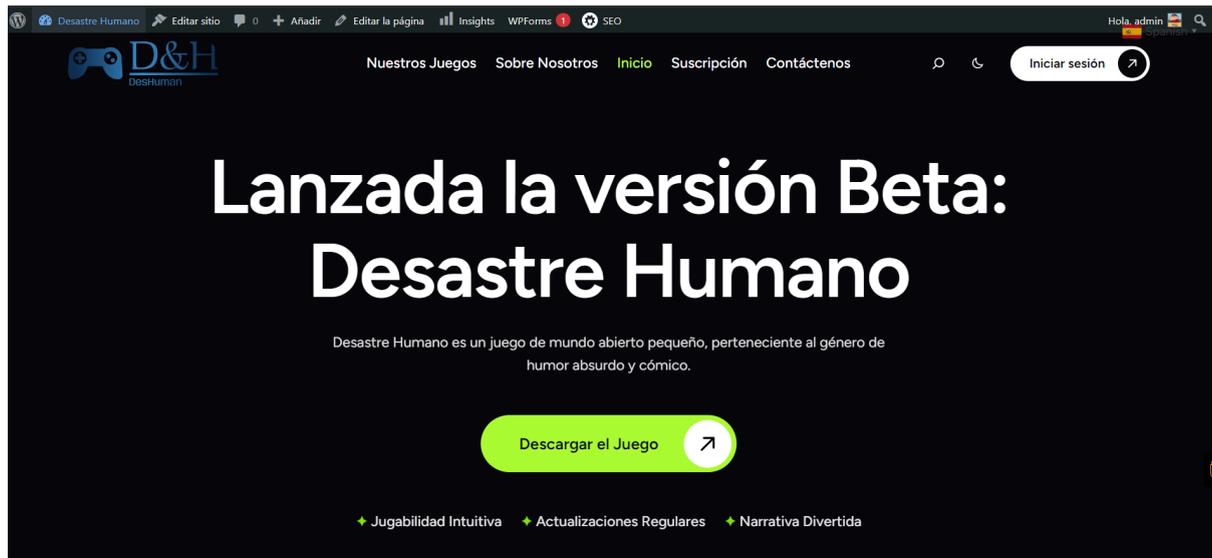


Brevo y API:



8.3 Nuestro sitio web de WordPress

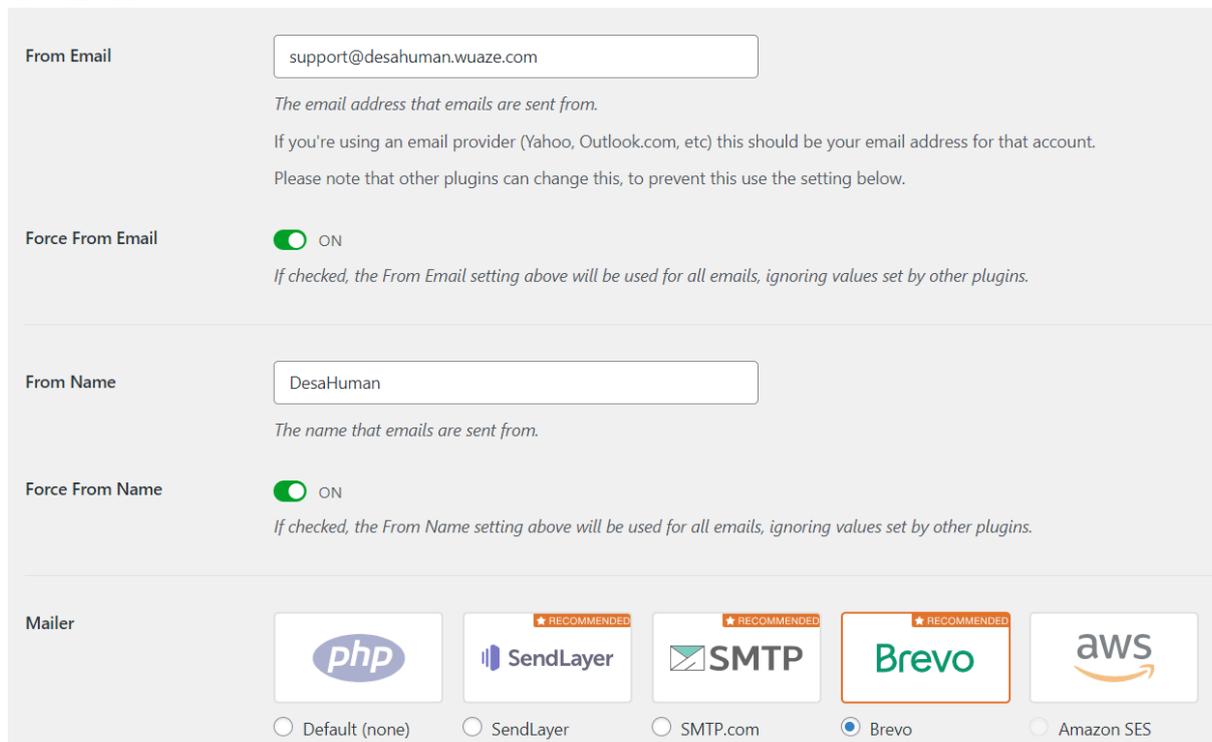
Nuestro sitio web es el corazón central de información para nuestros juegos. Aquí es donde los usuarios pueden encontrar una extensa gama de información detallada sobre la mecánica de juego, la trama, las últimas noticias y los recursos disponibles, además de descargar los juegos. Antes de esto, habíamos construido un prototipo muy simple de un sitio web usando código HTML; consulte [5] para obtener más detalles. Al construir este sitio, hemos añadido varios plugins para configurar nuestro sitio web. A continuación, describiremos brevemente los plugins más importantes que hemos utilizado en la web.



8.3.1 Plugin WP Mail SMTP Lite

Este es uno de los plugins más importantes para nuestro sitio web de WordPress. Nos ayuda a configurar el correo electrónico de envío y recepción. Elegimos Brevo como nuestro mailer porque usar su API es más conveniente. WP Mail SMTP nos ha ayudado a resolver problemas de envío de correo electrónico en el sitio web de WordPress. Y proporciona una opción SMTP, lo que nos permite utilizar un proveedor de servicios SMTP externo para enviar correos electrónicos.

Sin embargo, dado que estamos usando la versión Lite, nuestras capacidades están limitadas a la configuración del correo con Brevo y el correo electrónico del remitente.



Brevo

Brevo (formerly Sendinblue) is one of our recommended mailers. It's a transactional email provider with scalable price plans, so it's suitable for any size of business.

If you're just starting out, you can use Brevo's free plan to send up to 300 emails a day. You don't need to use a credit card to try it out. When you're ready, you can upgrade to a higher plan to increase your sending limits.

To get started, read our [Brevo documentation](#).

[Transparency and Disclosure](#)

API Key

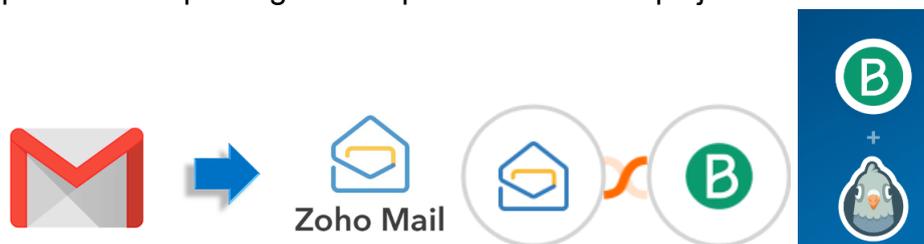
Follow this link to get an API Key: [Get v3 API Key](#).

Sending Domain

Please input the sending domain/subdomain you configured in your Brevo (formerly Sendinblue) dashboard. More information can be found in our [Brevo documentation](#).

Para configurar esto, primero usamos una cuenta de Gmail para registrar una cuenta en Zoho Mail. Luego, creamos un correo electrónico con nuestro propio dominio desde la cuenta de Zoho Mail y lo usamos para registrar una cuenta en Brevo. Después, obtenemos la clave API de Brevo y la introducimos en WP Mail SMTP. Esto nos permite utilizar el correo electrónico del dominio registrado en Zoho Mail como remitente de nuestro sitio web.

Aunque WP Mail SMTP admite directamente Zoho Mail, esto tiene un costo, por lo que tuvimos que seguir este proceso más complejo.



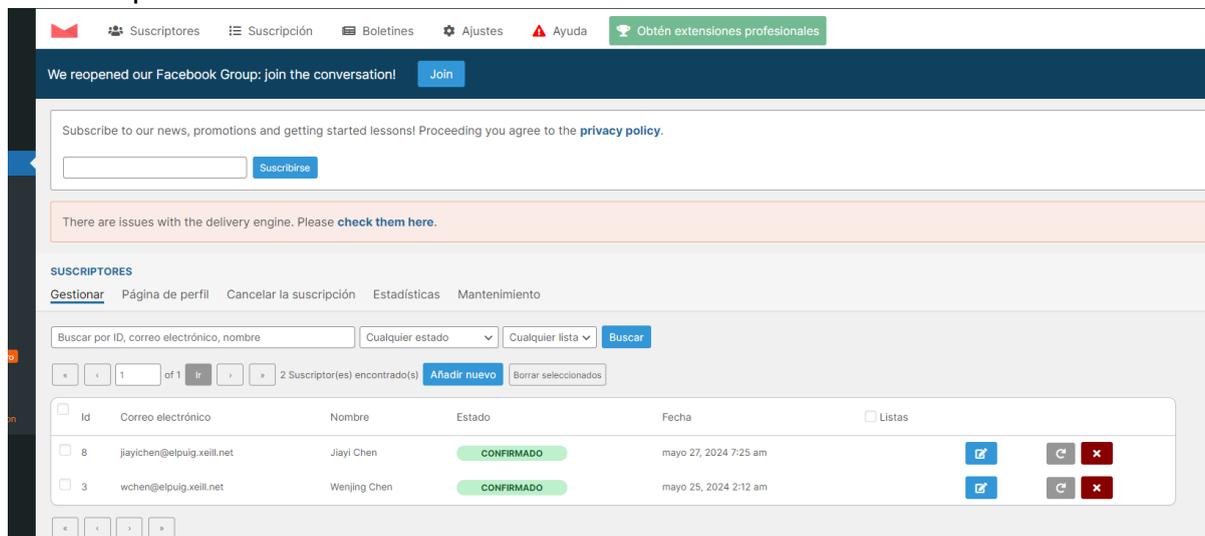
8.3.2 Plugin WP Forms

Utilizamos este plugin para crear formularios de contacto. Es muy fácil de usar, incluso podemos utilizar las plantillas prediseñadas sin necesidad de editar mucho.

The screenshot shows the WP Forms plugin interface. At the top, it says "Estás editando: Contacto con nosotros" and includes buttons for "Vista Previa", "Incrustar", and "Guardar". The main area is divided into a left sidebar for field configuration and a right preview area. The sidebar has sections for "Campos estándar" (Text, Paragraph, Accordion, Multiple choice, Checkboxes, Numbers, Name, Email, Number slider, reCAPTCHA) and "Campos elegantes" (Phone, Address, Date/Time, Web/URL). The preview area shows a form titled "Contacto con nosotros" with fields for "Nombre" (Name), "Apellido" (Last name), "Correo electrónico" (Email), and "Comentarios o mensajes" (Comments or messages), along with an "Enviar" (Send) button.

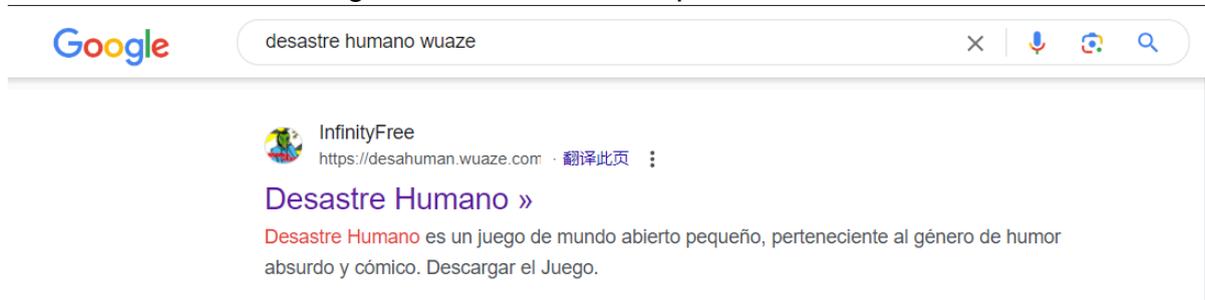
8.3.3 Plugin Newsletter

Usamos el plugin Newsletter para gestionar suscriptores y enviar boletines informativos, noticias y otros comunicados. Además, nos permite crear un formulario de suscripción de manera sencilla.



8.3.4 Plugin All in One SEO

Este plugin se utiliza para mejorar el SEO de nuestro sitio web y asegurarnos de que puede ser indexado por los motores de búsqueda. Lo hemos conectado y autorizado con Google Search Console, de modo que ahora, al buscar "desastre humano wuaze" en Google, nuestro sitio web aparece en los resultados.

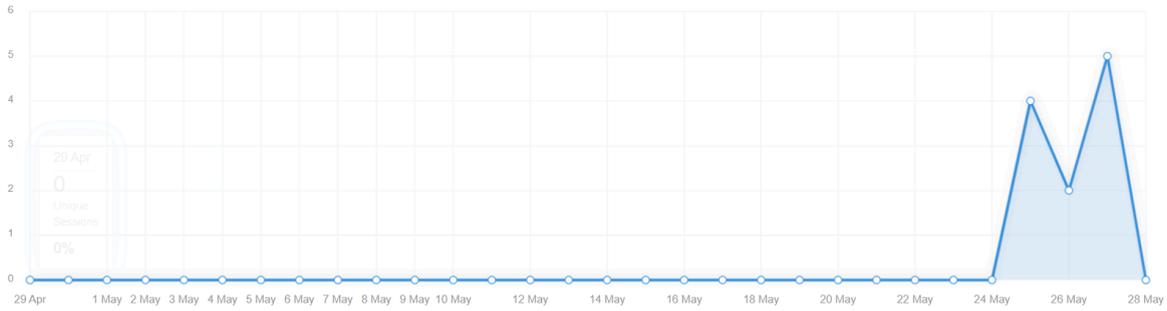


8.3.5 Plugin MonsterInsights

Este plugin se conecta con Google Analytics para rastrear y analizar los datos del sitio web, ayudándonos a comprender mejor el comportamiento de los usuarios y mejorar nuestro sitio web en consecuencia.

Sessions

Pageviews



Site Notes

Sessions 11 ↑ 100% vs. Previous 30 Days	Pageviews 27 ↑ 100% vs. Previous 30 Days	Avg. Session Duration 40s ↑ 100% vs. Previous 30 Days	Bounce Rate 27.3% ↑ 100% vs. Previous 30 Days
---------------------------------------------------------	----------------------------------------------------------	-----------------------------------------------------------------------	---------------------------------------------------------------

See Quick Links

8.3.6 Plugin GTranslate

Este plugin se encarga de añadir la funcionalidad multilingüe a nuestro sitio web. Con este plugin, no necesitamos crear páginas multilingües nosotros mismos, ya que automáticamente traduce nuestro sitio web a otros idiomas.

Opciones del widget

Apariencia del widget:

Traducir de:

* Estructura de URL de sub-directorio:

* Estructura de URL de subdominio:

Nombres nativos de idioma:

Cambiar automáticamente al idioma del navegador:

Activar CDN:

Mostrar en el menú:

Mostrar el selector de idioma flotante:

Selector CSS de la envoltura:

Tamaño de la bandera:

Estilo de la bandera:

Idiomas de la bandera: [Marcar todo](#) | [Desmarcar todo](#)

HINT: To reorder the languages simply drag and drop them in the list below.

<input checked="" type="checkbox"/> Africano	<input checked="" type="checkbox"/> Francés	<input checked="" type="checkbox"/> Kirguís	<input checked="" type="checkbox"/> Serbio
<input checked="" type="checkbox"/> Albano	<input checked="" type="checkbox"/> Frisón	<input checked="" type="checkbox"/> Lao	<input checked="" type="checkbox"/> Sesotho
<input checked="" type="checkbox"/> Amharca	<input checked="" type="checkbox"/> Gallego	<input checked="" type="checkbox"/> Latino	<input checked="" type="checkbox"/> Shona
<input checked="" type="checkbox"/> Árabe	<input checked="" type="checkbox"/> Georgiano	<input checked="" type="checkbox"/> Letón	<input checked="" type="checkbox"/> Sindhi
<input checked="" type="checkbox"/> Armenio	<input checked="" type="checkbox"/> Alemán	<input checked="" type="checkbox"/> Lituano	<input checked="" type="checkbox"/> Cingalés
<input checked="" type="checkbox"/> Azerbaiyano	<input checked="" type="checkbox"/> Griego	<input checked="" type="checkbox"/> Luxemburgués	<input checked="" type="checkbox"/> Eslovaco

Vista previa del widget

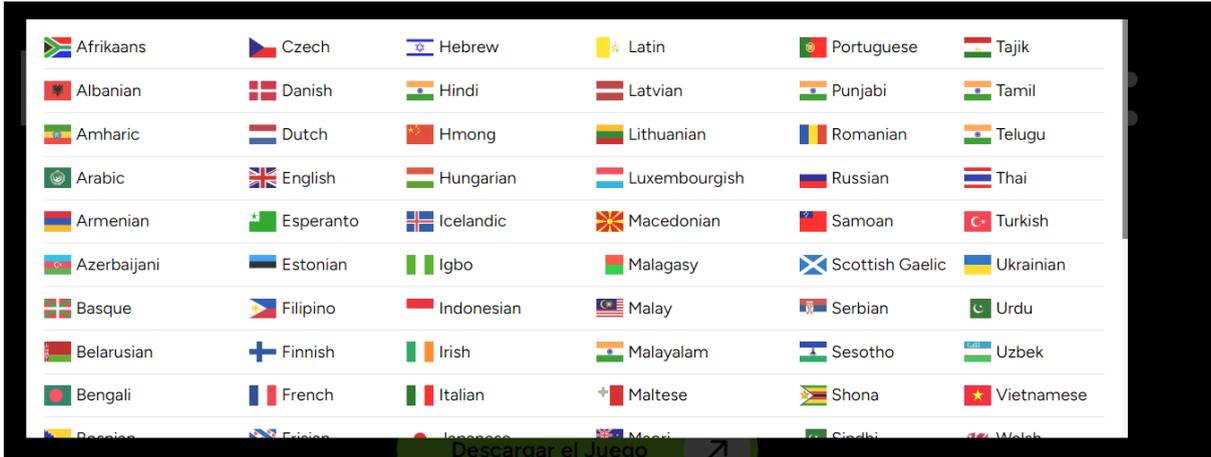
Spanish

Banderas alternativas

- Bandera de EEUU (Inglés)
- Bandera de Canadá (Inglés)
- Bandera de Brasil (Portugués)
- Bandera de México (Español)
- Bandera de Argentina (Español)
- Bandera de Colombia (Español)
- Bandera de Quebec (Francés)

Consejos de ubicación del selector de idioma

- La opción de mostrar el selector de idioma flotante es el modo más fácil y el que mejor funciona en la mayoría de webs.
- La opción de mostrar en el menú es mejor para Banderas, Banderas con el nombre del idioma, Banderas con el código del idioma, Nombres de idiomas, Códigos de idiomas.
- Puedes usar el widget de GTranslate en cualquier ubicación predefinida de widgets.
- El shortcode `[gtranslate]` se puede usar en cualquier parte de tu web.
- El shortcode `[gt-link lang="en" label="English" widget_look="flags_name"]` puede usarse para mostrar enlaces de idioma individuales.
- El selector envolvente CSS se puede utilizar para mostrar el selector de idioma dentro de los elementos coincidentes.



우리 게임 회사 소개 시작 신청 문의하기

베타 버전 출시: Human Disaster

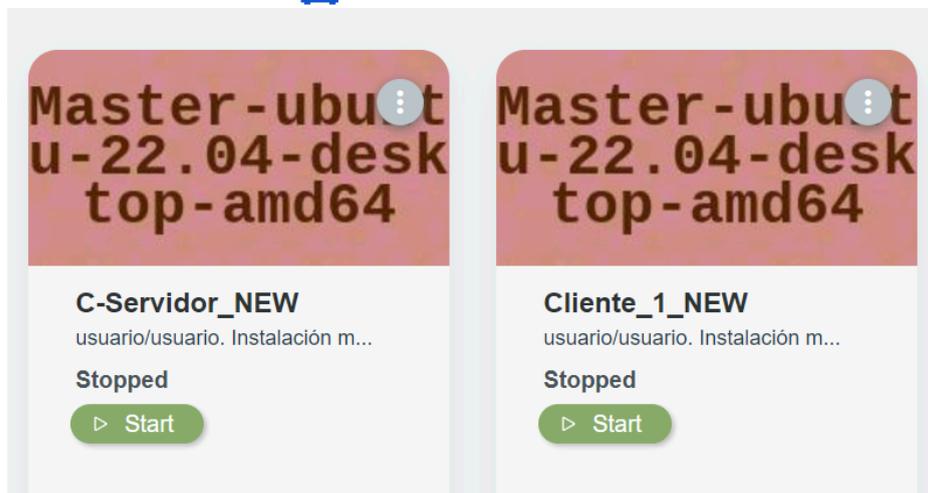
Human Disaster는 터무니없고 코믹한 유머 장르에 속하는 소규모 오픈 월드 게임입니다.

[게임 다운로드](#) ↗

◆ 직관적인 게임플레이 ◆ 정기 업데이트 ◆ 재미있는 이야기

8.4 Nuestro servidor correo en redes locales

Para crear nuestro servidor local de correo, utilizamos IsardVDI para crear dos máquinas virtuales. Aquí hay información de configuración detallada sobre nuestro servidor de correo electrónico [4].

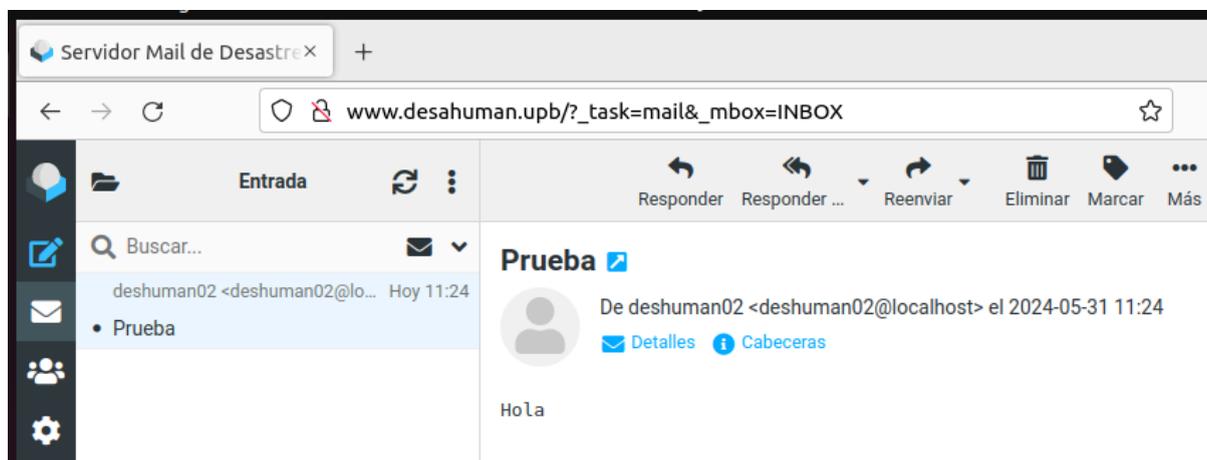
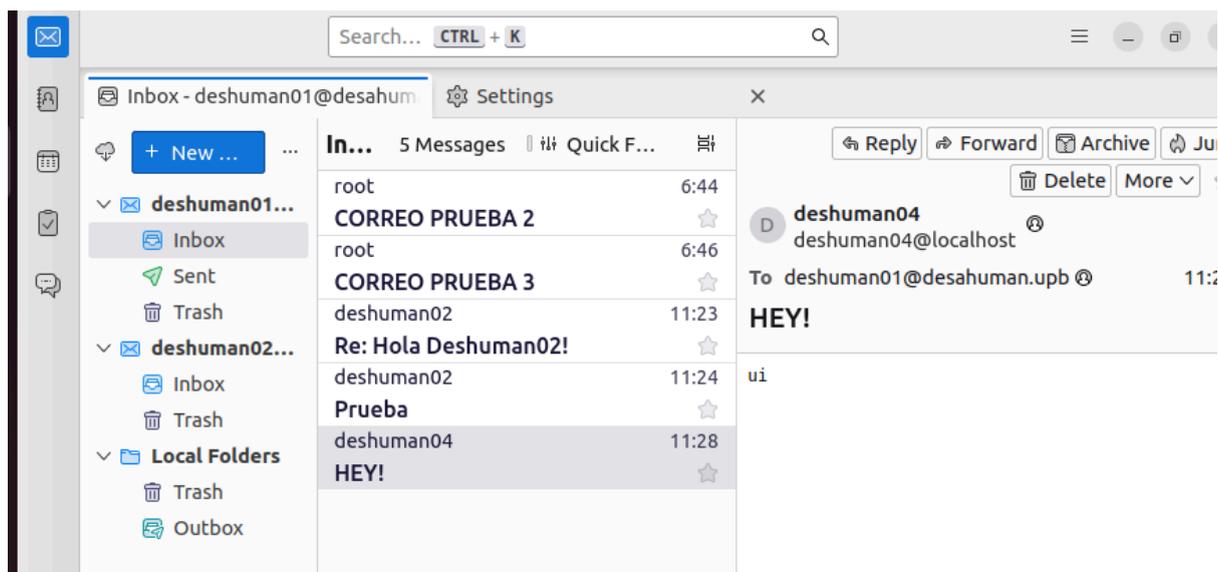


Hemos instalado Bind9, Postfix, Dovecot, Apache2, MailUtils, Thunderbird y Roundcube, y también cambiamos la configuración en Postfix para usar el formato Maildir para mostrar correos, cambiamos la configuración en mailutils y cambiamos el formato del buzón de Dovecot, etc.

Con estas configuraciones, nuestros servidores de correo pueden enviar correos electrónicos a direcciones externas. Aunque actualmente no podemos recibir respuestas desde fuera, los servidores internos pueden comunicarse libremente entre sí.

Esto lo logramos construyendo un servidor nosotros mismos. Siempre que se cree un usuario en el servidor, el cliente puede usarlo para iniciar sesión en Thunderbird o Roundcube y recibir y enviar correos electrónicos.





8.5 Lanzamiento y Promoción

En el futuro, planeamos lanzar nuestro juego en plataformas como Steam, Epic Store, Xbox Live y GOG.com. También consideramos lanzar el juego en tiendas físicas como Fnac y GAME.

Para promocionar nuestro proyecto de juego, utilizaremos las siguientes redes sociales:

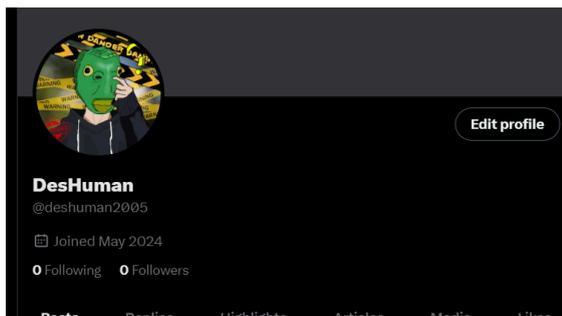
- **Tiktok**

<https://www.tiktok.com/@deshuman2005>



- **Twitter**

<https://x.com/deshuman2005>



Además, planeamos crear videos promocionales para aumentar la visibilidad y atraer a más jugadores. Estos videos se publicarán en nuestras redes sociales y plataformas de video para maximizar el alcance y el impacto de nuestra campaña de marketing. Pero como no tenemos tiempo para hacer un vídeo promocional en este momento, tendremos que dejarlo a un lado para el futuro.

9. Riesgos, desafíos y errores

9.1 Riesgos que se pueden encontrar en el proyecto

La siguiente tabla contiene una descripción general y una evaluación de los riesgos que podemos encontrar en nuestros proyectos.

Riesgos

9.2 Errores encontrados y soluciones

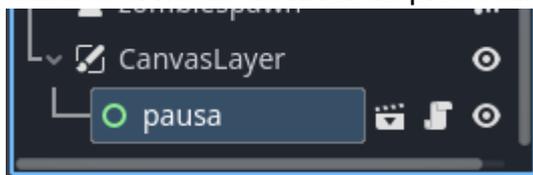
9.2.1 Errores

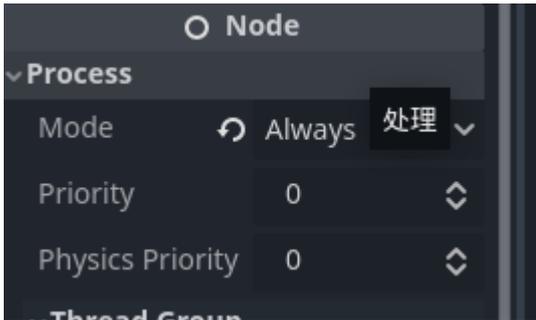
1. **Menú de pausa:** No hay suficientes capas para poder pulsar botones cuando aparece la interfaz.
2. **Animation player:** Errores en el juicio condicional de la animación de los personajes, no ejecutando bien cada acción.
3. **Animation player jump:** Los movimientos de salto del personaje no están bien presentados.
4. **Aimrayend:** La bala está en un lugar donde no debería estar.

9.2.2 Soluciones:

1. root/control/Node → PROCESS → ALWAYS

Cambie el modo de nivel de procesamiento a procesamiento frecuente.



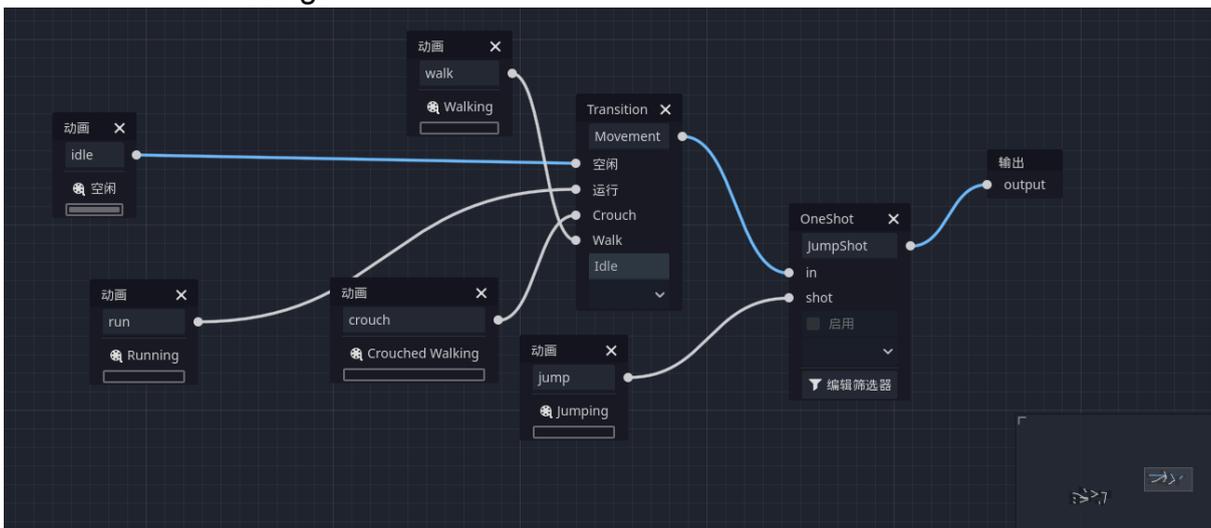


2. Añadir un no caminar si no se mueve a la condición de juicio, y añadir una reproducción de la animación juicio a continuación para caminar la animación si está caminando, de lo contrario está de pie.

```
elif not input_dir:  
    walking = false  
    sprinting = false  
    crouching = false
```

```
elif walking:  
    if animation_player.current_animation != "Walking":  
        animation_player.play("Walking")  
    else:  
        if animation_player.current_animation != "Idle":  
            animation_player.play("Idle")
```

3. Eliminar todo el código de juicio condicional de animación y sustituirlo por un juicio condicional utilizando el árbol de animación y una llamada de animación utilizando código.



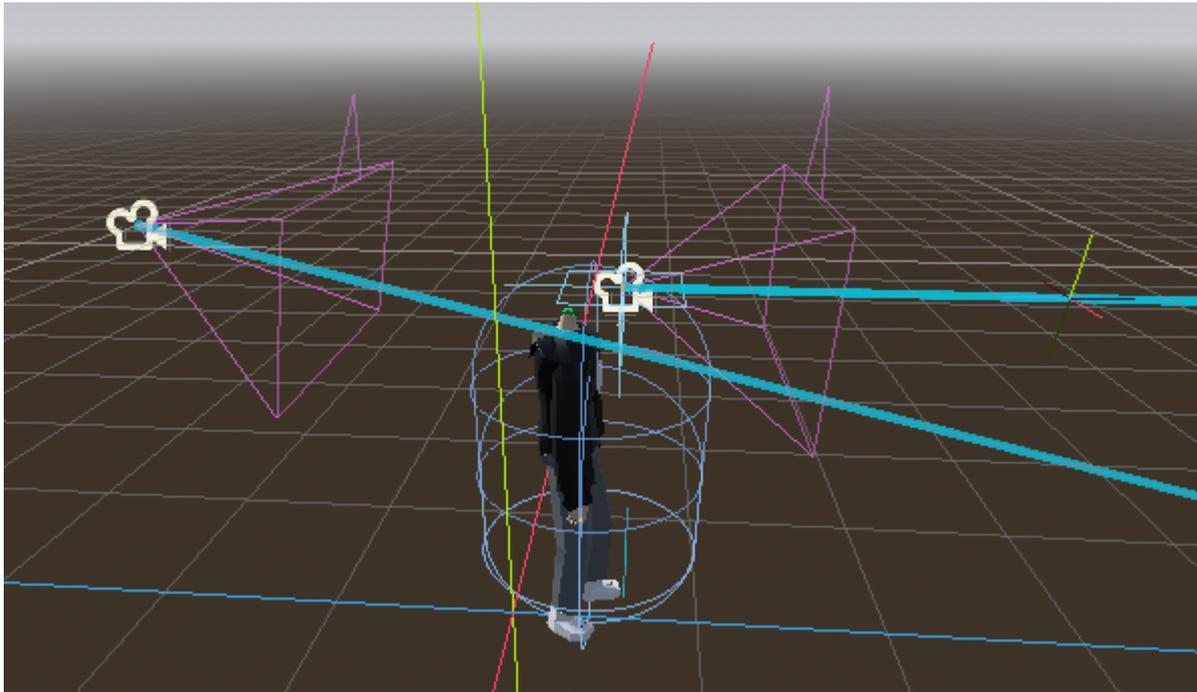
```
# Animation tree
▼ func handle_animation():
▼ | match CurrentAni:
▼ | | IDLE:
| | | animation_tree.set("parameters/Movement/transition_request", "Idle")
▼ | | WALK:
| | | animation_tree.set("parameters/Movement/transition_request", "Walk")
▼ | | RUN:
| | | animation_tree.set("parameters/Movement/transition_request", "Run")
▼ | | CROUCH:
| | | animation_tree.set("parameters/Movement/transition_request", "Crouch")

▼ func jump():
| animation_tree.set("parameters/JumpShot/request", AnimationNodeOneShot.ONE_SHOT_REQUEST_FIRE)
```

4. Las balas no están en el lugar correcto porque la alineación es incorrecta y los dos rayos están en la orientación equivocada.

crosshair

Camera3D
Aimray
Weapon_manager
aimrayend



10. Conclusión

Al principio, solo con nuestra pasión por los juegos en 3D, decidimos embarcarnos en la creación de un proyecto de juego en 3D. Teníamos grandes ambiciones, aunque en ese momento no nos dimos mucha cuenta de lo difícil que sería hacer un juego en 3D. Aún así, elegimos este tema sin dudarlo.

Nuestro equipo comenzó prácticamente desde cero, no sabemos utilizar las herramientas ni Godot ni Blender, por lo que nos enfrentamos a muchos desafíos durante el desarrollo del juego. Por ejemplo, experimentamos problemas con los modelos, fallos en el juego, cierres inesperados y una serie de situaciones que impedían que el juego funcionara. Además, la mayoría de los tutoriales y apoyo de la comunidad en línea estaban dirigidos a versiones antiguas de Godot y Blender, que diferían en varios aspectos de las versiones que estábamos utilizando, lo cual resultaba en errores y bugs. Con el tiempo, nuestras expectativas para el proyecto se redujeron.

Afortunadamente, perseveramos y logramos crear un juego muy básico, aunque tuvimos que eliminar muchas de las características y la trama que habíamos imaginado inicialmente. Aun así, esta experiencia nos permitió aprender mucho sobre el desarrollo de juegos.

En primer lugar, aprendimos a utilizar el motor de juegos Godot, que nos permitió desarrollar juegos y aplicaciones complejas y efectivas utilizando lenguajes de programación, gráficos, shaders y una variedad de nodos y scripts. También adquirimos conocimientos sobre modelado 3D utilizando Blender, desde formas geométricas básicas hasta diseños de personajes y entornos complejos. Además, aprendimos conceptos básicos de creación de sitios web, así como diseño de interacción de usuario, gestión de bases de datos y configuración de servidores, entre otros aspectos clave.

En cuanto a los resultados que hemos obtenido hasta ahora, estamos satisfechos, pero solo le daríamos una calificación de alrededor de seis sobre diez si tuviéramos que calificar. Creemos que aún falta mucho por mejorar en el juego, pero dadas las dificultades de crear un juego en 3D y las limitaciones de tiempo y recursos, solo pudimos lograr una versión beta básica y tuvimos que apresurarnos incluso para escribir nuestra documentación. En definitiva, la complejidad de este proyecto superó nuestro nivel actual de habilidades.

Para el futuro, es posible que sigamos mejorando este juego, agregando elementos que aún faltan y corrigiendo los errores que persisten. Somos conscientes de que nuestro proyecto de juego en 3D tiene mucho espacio para mejoras y perfeccionamiento.

Por último, a pesar de que los resultados no fueron los que esperábamos, disfrutamos mucho de este proyecto. Queremos agradecer a los creadores de tutoriales y recursos en línea, así como a los profesores que nos brindaron consejos. Su ayuda y apoyo han sido fundamentales para nuestro aprendizaje y crecimiento.

11. Bibliografía y referencias

Blender:

[1]

https://www.youtube.com/watch?v=vBlstqZKmEA&list=PLWnbm0Hd_z4QX9wc0fAgThJeXAuiDwY-7

https://www.youtube.com/watch?v=HM7TbsnZviM&list=PLWnbm0Hd_z4TNF18poVGNUVszKP74sdcJ

[2]

<https://www.youtube.com/watch?v=grvdGERUk6M>

[3]

<https://computerhoy.com/noticias/hardware/intel-presenta-jimmy-robot-hecho-impresion-3d-13631>

[4]

<https://www.youtube.com/watch?v=jo7FZBf4VkM>

https://www.youtube.com/watch?v=PcvnA56G_WA

<https://www.youtube.com/watch?v=hkGonXSd9ZQ>

[5]

<https://www.youtube.com/watch?v=H7Ck3w-F0aI>

https://www.youtube.com/watch?v=j_xDKMxcjYc

<https://www.bilibili.com/video/BV18U4y1P77q/>

SketchFab:

[Sci fi Monitor - Download Free 3D model by Siamak Tavakoli \(@payamtavakoli\) \[37a0fbb\]](#)

[Sci-Fi Lab Machine - Download Free 3D model by bossdeff \(@bossdeff\) \[675ae30\]](#)

[Sci-fi Pole - Download Free 3D model by Ahmed Ishaq Alizai \(@HardWireStudio\) \[c3e73be\]](#)

[Sci-Fi Panels - Download Free 3D model by Xavi Pujadas \(@XaviPujadas\) \[c1243c5\]](#)

[Low Poly Sci-Fi Tablet - Download Free 3D model by Snooze \(@Snooze\) \[ee1fde7\]](#)

[Sci-Fi Laboratory Op Table - Download Free 3D model by bossdeff \(@bossdeff\) \[ae314a5\]](#)

[Pool of Blood Low Poly Planes - Download Free 3D model by Shaz \(@shaz13\) \[7f26b96\]](#)

[wheelchair \(horror game hospital\) - Download Free 3D model by japan3d \(@japan3d\) \[5d1fb33\]](#)

[human - Download Free 3D model by kaytech429 \(@kaytech429\) \[437f3f1\]](#)

[Science lab cabinet - Download Free 3D model by tboiston \(@tboiston\) \[121829e\]](#)

[Electrical Control Panel sci-fi 3D Model](#)

[Sci-Fi Table - Download Free 3D model by nenkea \(@nenkea\) \[05de9f0\]](#)

Web y server:

<https://elpuig.xeill.net/Members/vcarceler/c1/didactica/apuntes/ud4/na8>
https://elpuig.xeill.net/Members/vcarceler/articulos/gnu-linux-como-router-reenvio-de-paquetes-y-nat/index_html
<https://elpuig.xeill.net/Members/vcarceler/articulos/introduccion-a-apache-http-server>
<https://elpuig.xeill.net/Members/vcarceler/articulos/introduccion-al-correo-electronico-con-postfix>
<https://elpuig.xeill.net/Members/vcarceler/articulos/cambio-de-mbox-a-maildir-en-postfix-y-mailutils>
<https://elpuig.xeill.net/Members/vcarceler/articulos/introduccion-a-dovecot>
<https://elpuig.xeill.net/Members/vcarceler/articulos/correo-electronico-con-postfix-dovecot-y-thunderbird-en-ubuntu-20.04>
<https://elpuig.xeill.net/Members/vcarceler/articulos/un-mua-web-roundcube>
https://www.youtube.com/watch?v=LG_CtvauPr8
https://www.youtube.com/watch?v=sW9gD_qLBcM

Godot:

<https://docs.godotengine.org/en/stable/>
[0]
<https://heroiclabs.com/docs/nakama/client-libraries/godot/>
[1]
<https://www.youtube.com/watch?v=sVsn9NqpVhg>
<https://www.youtube.com/watch?v=xIKErMgJ1Yk>
[2]
<https://www.youtube.com/watch?v=JxbnStn-BIY>
<https://kenney.nl/assets/space-station-kit>
[3]
<https://www.youtube.com/watch?v=iV710Vm5qm0>
[4]
<https://www.youtube.com/watch?v=EP5AYllgHy8>
[5]
https://www.youtube.com/watch?v=Z8jcy_jZyk&list=PLhBqFleCVBkXQiE8Nm4Co_1jJ4L7UIzr
[6]
<https://www.youtube.com/watch?v=a3GjUyBDwe4>
[7]
<https://www.youtube.com/watch?v=An3uHrAoHRw>
[8]
<https://www.youtube.com/watch?v=oV8c9aIXVwU>
[9]
https://www.youtube.com/watch?v=jLle1_xvOXU

12. Anexos

[1] Narrador del inicio

En 1993, los humanos desarrollaron una señal de onda llamada "señal de mostaza" e intentaron enviarla al espacio exterior. Fue un gran avance tecnológico, pero los humanos no se dieron cuenta de que esta acción revelaría la ubicación de la Tierra. Así, la Tierra comenzó a ser gradualmente cubierta por seres extraterrestres...

(Cambio de tono)

Oh, pero esto no es realmente de lo que quiero hablar. La historia que quiero contar es la legendaria historia de un animal.

(Cambio de escena)

Ahem, este es el protagonista de nuestra historia. Denle la bienvenida a (nombre del personaje).

- **Introducción del protagonista FishMan**

Este es nuestro protagonista, ¡(nombre del personaje)! Un pobre trabajador. ¡Miren, su aspecto cansado es como el de una tortuga que está a punto de morir de sed!

(Cambio de escena al cielo)

¿Hmm? ¿Espera? ¿Qué es eso en el cielo, un OVNI emitiendo una extraña luz?

¡Hey! ¡(nombre del personaje) , ¿lo has visto?! (El protagonista, desanimado, no se da cuenta del OVNI)

Hmm... Tengo un muy mal presentimiento. (El protagonista es abducido por el OVNI)

¡Dios mío! ¡(nombre del personaje) ha sido abducido por el OVNI! ¿Qué vamos a hacer??! (Tono fingiendo pánico)

(De repente, en tono profundo) Oh, no te preocupes, porque este es solo el comienzo de la leyenda de (nombre del personaje) San.

(Cambio de escena desde la pantalla de un televisor al protagonista desmayado en el espacio).

- **Introducción del protagonista CatLady**

Esta es nuestra protagonista, ¡(nombre del personaje)! Ella es una famosa pianista (comienza en un escenario: las luces la buscan hasta encontrarla), ¡Escuchen! ¡Qué melodía tan maravillosa!

(Cambio de escena al techo del teatro y luego al público)

¿Eh? ¿Qué está pasando? ¿Un terremoto? ¡Todos busquen refugio! En ese momento, una gran fuerza de succión vino del cielo, ¡todos fueron abducidos!

¡Dios mío! ¡(nombre del personaje) ha sido abducida por un OVNI! ¿Qué vamos a hacer??! (Tono fingiendo pánico)

(De repente, en tono profundo) Oh, no te preocupes, porque este es solo el comienzo de la leyenda de (nombre del personaje).

(Cambio de escena desde la pantalla de un televisor al protagonista desmayado en el espacio).

- **Escena de la Nave Espacial Extraterrestre**

(Cambio de escena a la vista en primera persona del protagonista, la pantalla parpadea irregularmente, oscureciéndose a intervalos).

(Delante del protagonista, dos extraterrestres están conversando).

Extraterrestre 1: "Comunicación encriptada (la sede nos ha ordenado modificar a estos seres inferiores, este es el sujeto de prueba número 995 y todavía no hemos tenido éxito)."

Extraterrestre 2: "Comunicación encriptada (¿Quién sabe cuál es el propósito del jefe con estos experimentos?)."

Extraterrestre 1: "Comunicación encriptada (Ah, las órdenes de los superiores son difíciles de entender)."

Extraterrestre 2: "Comunicación encriptada (Espera, uno de los sujetos está despertando)."

La cabeza del segundo extraterrestre gira 180 grados, fijando su mirada en el protagonista desmayado. El primer extraterrestre, al verlo, va a un armario cercano y saca un sedante, inyectando al protagonista, quien se desmaya de nuevo.

Aparece la interfaz de selección de animales, mostrando cada animal en negro (solo siluetas y contornos) para que el jugador elija.

[2] PLAN DE EMPRESA

📄 Análisis de Proyecto - Jiayi y Wenjing

[3] Nuestro web

<https://desahuman.wuaze.com> o <http://desahuman.wuaze.com>

[4] Servidor de correo en red local

Servidor

Nuestro servidor tiene dos redes, una que permite la comunicación con la red externa y otra que se utiliza para nuestra red interna.

La red externa es **enp1s0**, con una IP de **192.168.238.74**.

La red interna es **enp2s0**, con una IP de **192.168.207.2**.

A continuación se mostrará una captura de pantalla de nuestra configuración.

Este es la configuración de Netplan de nuestro servidor:

```
root@server:~# cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  ethernets:
    enp1s0:
      dhcp4: true
    enp2s0:
      addresses: [192.168.207.2/24]
      routes:
        - to: default
          via: 192.168.238.74
      nameservers:
        search: [desahuman.upb]
        addresses: [192.168.207.2]
  version: 2
  renderer: NetworkManager
```

Esta es la configuración de hosts (resolver nombres locales) y hostname de nuestro servidor:

```
root@server:~# cat /etc/hostname
server
root@server:~# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    server.desahuman.upb server

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Configuramos el archivo de configuración local de Bind9.

En el archivo de configuración local de bind9 (**named.conf.local**), añadimos dos zonas, una de directa y otra de inversa que apuntan a dos archivos de datos diferentes respectivamente.

```
zone "desahuman.upb" IN {
    type master;
    file "/var/cache/bind/file.desahuman.upb";
    allow-update { none; };
};

zone "207.168.192.in-addr.arpa" IN {
    type master;
    file "/var/cache/bind/inverso.207.168.192";
    allow-update { none; };
};
```

Estos son los archivos de datos de caché que creamos para Bind9, **zona directa** y **zona inversa**.

```
root@server:~# cat /var/cache/bind/file.desahuman.upb
$ORIGIN desahuman.upb.
$TTL 86400 ; 1 dia
@      IN      SOA      server.desahuman.upb. wchen.elpuig.xeill.net. (
                                2020070801 ; Serial
                                6H      ; Refresh
                                1H      ; Retry
                                2W      ; Expire
                                3H      ; Minimum TTL
                                )

@      IN      NS       server.desahuman.upb.
@      IN      MX 10    mail.desahuman.upb.
@      IN      A        192.168.207.2
server IN      A        192.168.207.2
mail   IN      A        192.168.207.2
www    IN      A        192.168.207.2
mail   IN      TXT      "v=spf1 a mx a: ip4:192.168.207.2 -all"
cliente IN     A        192.168.207.50
```

En configuración de la conversión inversa de nombres de dominio a direcciones IP establecimos la conversión inversa, donde los números representan la cuarta parte de la dirección IP y se combinan con el nombre de dominio inverso especificado.

```
root@server:~# cat /var/cache/bind/inverso.207.168.192
$ORIGIN 207.168.192.in-addr.arpa.
$TTL 86400 ; 1 dia
@      IN      SOA      server.desahuman.upb. wchen.elpuig.xeill.net. (
                                2020042501
                                6H
                                1H
                                2W
                                3H
                                )

@      IN      NS       server.desahuman.upb.
2      IN      PTR      server.desahuman.upb.
2      IN      PTR      mail.desahuman.upb.
2      IN      PTR      www.desahuman.upb.
50     IN      PTR      cliente.desahuman.upb.
```

- IP forward y NAT - /etc/sysctl.conf

```
root@server:~# sysctl -p
net.ipv4.ip_forward = 1
```

- Iptables y /etc/rc.local

```
iptables -t nat -A POSTROUTING -o enp0s3 -s192.168.207.0/24 -j MASQUERADE
```

```
GNU nano 6.2 /etc/rc.local
#!/bin/bash

iptables -t nat -A POSTROUTING -o enp1s0 -s 192.168.207.0/24 -j MASQUERADE

exit 0
```

- Maildir, hostname, destination, networks - /etc/postfix/main.cf

```
myhostname = mail.desahuman.upb
```

```
mydestination = $myhostname, desahuman.upb, localhost.$mydomain, localhost
```

```
mynetworks = 192.168.207.0/24 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

```
mailbox_size_limit = 0
```

```
inet_protocols = all
home_mailbox = Maildir/
```

- Maildir - /etc/mailutils.conf

```
GNU nano 6.2 /etc/mailutils.conf
mailbox{
    mailbox-pattern "maildir:///home/${user}/Maildir";
    mailbox-type maildir;
}
```

- Maildir - /etc/dovecot/conf.d/10-mail.conf

```
mail_location = maildir:~/Maildir
```

- SSL - /etc/dovecot/conf.d/10-ssl.conf

```
# SSL/TLS
ssl = no
```

- Plaintext auth - /etc/dovecot/conf.d/10-auth.conf

```
# See also ssl=required setting
disable_plaintext_auth = yes
```

- Apache PHP - /etc/php/8.1/apache2/php.ini

```
date.timezone = "Europe/Madrid"
```

```
upload_max_filesize = 50M
```

```
post_max_size = 18M
```

```
mbstring.func_overload = 0
```

- VirtualHost - /etc/apache2/sites-enabled/000-default.conf

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/roundcube

<Directory /var/www/roundcube>
    Options -Indexes
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

- ServerName - /etc/apache2/apache2.conf

```
# vim: syntax=apache ts=4 sw=4 st
ServerName www.desahuman.upb
```

- MySQL

```
mysql -u root -p
```

```
CREATE DATABASE roundcubemail /*!40101 CHARACTER SET utf8 COLLATE
utf8_general_ci */;
```

```
CREATE USER 'roundcube'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON roundcubemail.* to 'roundcube'@'localhost';
```

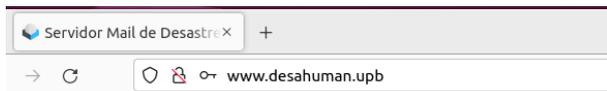
```
exit
```

```
mysql -u root -p roundcubemail < /var/www/roundcube/SQL/mysql.initial.sql
```

- Thunderbird

The screenshot shows the Thunderbird account configuration window. It includes fields for 'Your full name' (deshuman03), 'Email address' (deshuman03@desahuman.upb), and 'Password' (masked with dots). There is a 'Remember password' checkbox which is checked. A green message box states 'Configuration found by trying common server names.' Below this, the 'Available configuration' section shows 'IMAP' selected with sub-configuration: Incoming IMAP (NO ENCRYPTION) mail.desahuman.upb, Outgoing SMTP (STARTTLS) mail.desahuman.upb, and Username deshuman03. At the bottom, there are buttons for 'Configure manually', 'Cancel', and 'Done'.

- Roundcube



The login form contains a username field with 'deshuman04', a password field with masked characters, and a blue 'INICIAR SESIÓN' button.

Servidor Mail de Desastre Humano

Ciente

Nuestro cliente solo tiene una red interna, que se conecta a nuestro servidor. La red interna es **enp1s0**, con una IP de **192.168.207.50**. A continuación se muestra una captura de pantalla de nuestra configuración.

Este es la configuración de Netplan de nuestro cliente:

```
GNU nano 6.2 /etc/netplan/01-network-manager-all.yaml *
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp1s0:
      dhcp4: no
      addresses: [192.168.207.50/24]
      routes:
        - to: 0.0.0.0/0
          via: 192.168.207.2
      nameservers:
        search: [desahuman.upb]
        addresses: [192.168.207.2]
```

Esta es la configuración de hosts (resolver nombres locales) y hostname de nuestro cliente:

```
root@cliente:/home/usuario# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    cliente.desahuman.upb cliente

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
root@cliente:/home/usuario# cat /etc/hostname
cliente
```

[5] Web HTML

[Pagina web.zip](#)