



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

**ESCUELA TÉCNICA SUPERIOR DE INGENIERIA
INFORMÁTICA**

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Far Exile: Creación de un videojuego

Juan Rovira Lluch



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departamento de Sistemas Informáticos y Computación

Director del proyecto : Ramón Mollá

Autor del proyecto : Juan Rovira Lluch

Fecha :21 de diciembre de 2012

Fdo : Juan Rovira Lluch

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2012 Juan Rovira Lluch

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".

Índice general

1 -Introducción.....	8
1.1 -Objetivos.....	8
1.2 -Estructura de la memoria.....	9
1.3 -Herramientas.....	9
Gimp 2.....	10
Blender.....	11
OpenOffice.....	13
Audacity.....	15
2 -Estado del arte.....	16
3 -Justificación del proyecto.....	19
3.1 -Análisis de riesgos.....	19
3.2 -Licencias.....	19
Este proyecto esta bajo licencia GNU (Ver apéndices).....	19
3.3 -Estudio de costes.....	19
4 -Especificación y adaptación del motor.....	21
4.1 -Definición.....	21
4.2 -Análisis de requisitos.....	21
4.2.1 -Requisitos funcionales.....	21
4.2.2 -Requisitos de rendimiento.....	21
4.2.3 -Requisitos de diseño.....	22
4.3 -Adaptación.....	22
Torque.....	22
Fly3D.....	23
Crystal Space.....	23
Ogre3D.....	24
OSG.....	25
Delta3D.....	26
Blender game engine.....	27
5 -Análisis.....	29
5.1 -Selección de objetivos.....	29
5.1.1 -Publico Objetivo.....	29
5.1.2 -Objetivos globales.....	29
5.1.3 -Objetivos locales.....	30
5.2 -Género.....	30
5.3 -Guión.....	35
5.4 -Especificación conceptual.....	35
5.4.1 -Entrada y salida.....	35
5.4.2 -Mapa de controles.....	36
5.4.3 -Principios de Norman.....	36
5.4.4 -Método de aprendizaje : Condicionamiento Operante.....	37
5.5 -Tópicos.....	37
5.6 -Pre-programación.....	37
6 -Diseño.....	39
6.1 -Bucle principal.....	39
6.1.1 -Diagrama.....	39
6.1.2 -Módulos.....	40
6.2 -Inteligencia Artificial.....	41

6.2.1 -Malo 1.....	41
6.2.2 -Lanzacohetes.....	42
6.2.3 -Jefe.....	42
7 -Implementación.....	43
7.1 -Códigos.....	43
7.1.1 -Primera fase.....	43
7.1.2 -Introducción.....	49
7.1.3 -Inicio.....	49
7.1.4 -Game Over.....	50
7.1.5 -Créditos.....	50
7.1.6 -Controles.....	50
7.1.7 -HUD.....	50
7.2 - Problemas concretos.....	53
7.2.1 -Colisiones.....	53
7.2.2 -Limites de pantalla.....	54
7.2.3 -Reproducción de sonidos.....	55
8 -Conclusiones.....	56
8.1 -Desarrollo del proyecto.....	56
8.2 -Posibles ampliaciones.....	56
9 -Agradecimientos.....	58
10 -Apéndices.....	59

Indice de apéndices

A.Ficha del videojuego.....	59
1.Título y versión.....	59
2.Referencias.....	59
3.Referencias.....	59
4.Tópicos.....	61
5.Historia.....	62
6.Plataformas.....	64
7.Mecánica y física del juego.....	64
8.Personajes.....	64
8.1.Principales.....	64
8.2.Enemigos.....	68
9.Escenarios.....	68
10.Objetos 3D.....	69
10.1.Personajes (naves).....	69
10.2.Enemigos.....	70
10.3.Jefes.....	71
10.4.Objetos.....	71
10.5.Escenografía.....	72
10.6.Pantallas.....	73
11.GUI.....	76
12.Mapa de Botones.....	76
13.Sonidos y Muscia.....	77
13.1.Acciones.....	77
13.2.Eventos.....	77

13.3.Pantallas.....	77
14. ArtBook.....	79
14.1.Personajes.....	79
14.2.Escenario.....	92
14.3.Pantallas.....	93
14.4.Naves y objetos.....	95
B.Tutorial de crear un ejecutable con blender.....	100
C.Instalación del juego.....	101
D.Tutorial para la herramienta de lógica del blender.....	102
E.Python y Jython.....	103
F.Indice de Imagenes.....	104
G.Bibliografía y otros enlaces.....	106
1.Bibliografía recomendada.....	106
2.Estructura del directorio del proyecto.....	106
3.Otros enlaces.....	107
H.Licencias y contratos.....	109

1 - Introducción

Este proyecto se redacta con carácter de Trabajo para el Proyecto de Final de Carrera, para la obtención por parte de quien lo suscribe del título de Ingeniero Informática de Sistemas.

La selección de este proyecto es una iniciativa propia, por gusto o por trato desde hace años con este tipo de material, he decidido dirigir todo mi conocimiento sobre el campo hacia la creación de un producto multimedia, en este caso un Videojuego.

Con el termino Videojuego se da a entender a la creación de un producto de ocio y diversión para un público. Un videojuego esta compuesto por elementos visuales y auditivos combinados con una interacción del usuario por medio del teclado, ratón o un joystick.

Mi intención, es obtener un videojuego que ofrezca diversión y un material que haga retomar los viejos clásicos de la época. El videojuego estará desarrollado y redactado por completo: fases, personajes, objetos, herramientas, etc...; con un final, el de conseguir como mínimo una demo jugable y dejar el producto preparado para una posible ampliación. Una demo o versión de demostración es simplemente una versión de videojuego incompleta pero jugable, se suelen crear para que el público pueda tener una primera impresión de los factores más importantes del producto (gráficos, jugabilidad, historia y ambiente (sonidos y música)).

1.1 - Objetivos

El objetivo principal de este proyecto es adquirir toda la experiencia posible en el campo de la creación de un software. Desarrollar este hasta conseguir un producto final que en este caso será una DEMO.

Los objetivos que pretendo alcanzar en la realización de este proyecto son los siguientes:

- Establecer mediante la elaboración de un guión de una pequeña historia un caso de estudio del videojuego para realizar un diseño de objetos y personajes y establecer relaciones entre ellos.
- Elegir uno entre todos los motores de 3D para el desarrollo de videojuegos que ofrece la red de Internet gratuitamente.
- La utilización de diagramas para establecer el orden de pantallas y de módulos de programación.
- Realizar una implementación en forma de Demo utilizando todos los puntos anteriores.

1.2 - Estructura de la memoria

La estructura de la memoria está dividida de la siguiente manera :

Lo primero será la introducción, donde se explica el contexto donde se va a desarrollar la aplicación, la necesidad de crearla y los objetivos y contenidos de la aplicación.

En el capítulo de el estado del arte se expone la realidad de los videojuegos actualmente en nuestra sociedad. Tanto en aspectos económicos, como en alternativas para el tiempo libre, podremos ver que la creación de estas aplicaciones es una inversión casi segura de éxito.

Luego la definición de la adaptación del motor de videojuegos que necesitará, el criterio para elegir uno y descartar los demás según las necesidades y objetivos del proyecto. Incluidas las herramientas para el desarrollo del mismo.

En la descripción del análisis se especifica el tipo de usuario para el proyecto (publico objetivo), el lenguaje de programación, la descripción de cada uno de los elementos del videojuego (tanto personajes como objetos, pantallas, etc...) y las relaciones entre ellos, controles e interfaz.

En el capítulo de diseño está la programación de cada uno de los objetos del juego.

Además, en las conclusiones, hay un esquema de la duración de las etapas, explicación de los problemas encontrados y una conclusión final después de acabar la demo del videojuego y puesto en uso.

En los apéndices están las referencias y nombres de los archivos y documentos utilizados en este proyecto. También estará una ficha completa del juego con capturas de pantalla , descripciones de los personajes y un apartado dedicado a bocetos y diseños. Por último, la bibliografía con todos los tutoriales, ejemplos y líneas de ayuda necesarios para el desarrollo del juego.

1.3 - Herramientas

Todas las herramientas utilizadas para el desarrollo de este videojuego son de distribución libre. Esto quiere decir que cualquier usuario puede descargarlas de Internet sin pagar licencias y disfrutar gratuitamente.



Ilustración 1: Logo Gimp 2.0

Gimp 2

Es un programa de edición de imágenes de mapa de bits, tanto dibujos como fotografías. Forma parte del proyecto GNU y está disponible bajo la Licencia pública general de GNU.

Es el programa de manipulación de gráficos disponible en más sistemas operativos (Unix, GNU/Linux, FreeBSD, Solaris, Microsoft Windows y Mac OS X, entre otros).

GIMP permite el tratado de imágenes en capas, para poder modificar cada objeto de la imagen en forma totalmente independiente a las demás capas en la imagen, también pueden subirse o bajarse de nivel las capas para facilitar el trabajo en la imagen, la imagen final puede guardarse en el formato xcf de GIMP que soporta capas, o en un formato plano sin capas, que puede ser png, bmp, gif, jpg, etc.

Con GIMP es posible producir imágenes de manera totalmente no interactiva (por ejemplo, generar al vuelo imágenes para una página web usando guiones CGI) y realizar un procesamiento por lotes que cambien el color o conviertan imágenes. Para tareas automatizables más simples, probablemente sea más rápido utilizar un paquete como ImageMagick.

El nombre de GIMP en español se forma con las iniciales de Programa de Manipulación de Imágenes de GNU leídas de atrás hacia adelante.

Extraído de : <http://es.wikipedia.org/wiki/GIMP>



Ilustración 2: Logo Blender 2.46

Blender

Es un programa informático multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales.

El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX.

Tiene una muy peculiar interfaz gráfica de usuario, que se critica como poco intuitiva, pues no se basa en el sistema clásico de ventanas; pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús y vistas de cámara.

Originalmente, el programa fue desarrollado como una aplicación propia por el estudio de animación holandés NeoGeo; el principal autor, Ton Roosendaal, fundó la empresa "Not a Number Technologies" (NaN) en junio de 1998 para desarrollar y distribuir el programa.

La compañía cayó en bancarrota en 2002, entonces los acreedores acordaron ofrecer Blender como un producto de código abierto y gratuito bajo los términos de la GNU GPL a cambio de €100.000. El 18 de julio de 2003, Roosendaal creó sin ánimo de lucro la Fundación Blender para recoger donaciones; el 7 de septiembre se anuncia la recaudación como exitosa (participaron también ex empleados de NaN) y el código fuente se hizo público el 13 de octubre.

Características :

- Multiplataforma, libre, gratuito y con un tamaño de origen realmente pequeño comparado con otros paquetes de 3D, dependiendo del sistema operativo en el que se ejecuta.

- Capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, NURBS, metaballs.
- Junto a las herramientas de animación se incluyen cinemática inversa, deformaciones por armadura o cuadrícula, vértices de carga y partículas estáticas y dinámicas.
- Edición de audio y sincronización de video.
- Características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógica.
- Posibilidades de renderizado interno versátil e integración externa con potentes trazadores de rayos o "raytracer" libres como kerkythea, YafRay o Yafrid.
- Lenguaje Python para automatizar o controlar varias tareas.
- Blender acepta formatos gráficos como TGA, JPG, Iris, SGI, o TIFF. También puede leer ficheros Inventor.
- Motor de juegos 3D integrado, con un sistema de ladrillos lógicos. Para más control se usa programación en lenguaje Python.
- Simulaciones dinámicas para softbodies, partículas y fluidos.
- Modificadores apilables, para la aplicación de transformación no destructiva sobre mallas.
- Sistema de partículas estáticas para simular cabellos y pelajes, al que se han agregado nuevas propiedades entre las opciones de shaders para lograr texturas realistas.

Extraído de : <http://es.wikipedia.org/wiki/Blender>



Ilustración 3: Logo OpenOffice.org

OpenOffice

OpenOffice.org (frecuentemente escrito **OOo** para abreviar) es una *suite* ofimática libre (código abierto y distribución gratuita) que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos. Está disponible para varias plataformas, tales como Microsoft Windows, GNU/Linux, BSD, Solaris y Mac OS X. Soporta numerosos formatos de archivo, incluyendo como predeterminado el formato estándar ISO/IEC OpenDocument (ODF), entre otros formatos comunes, así como también soporta más de 110 idiomas, desde febrero del año 2010.

OpenOffice.org tiene como base inicial a StarOffice, una suite ofimática desarrollada por StarDivision y adquirida por Sun Microsystems en agosto de 1999. El desarrollo de la suite está liderado por Oracle Corporation (inicialmente por Sun Microsystems), en colaboración con otras compañías como Novell, RedHat, RedFlag CH2000, IBM, Google, entre otras hasta octubre de 2010. El código fuente de la aplicación está disponible bajo la Licencia pública general limitada de GNU (LGPL) versión 3.

El proyecto y el programa son denominados «OpenOffice» de forma informal, aunque «OpenOffice.org» es el nombre oficial completo, ya que la denominación *openoffice* es una marca registrada en posesión de otra empresa.

Aplicaciones incluidas :



Writer: OpenOffice.org Writer es el procesador de textos de la suite ofimática. Writer permite exportar archivos de texto a los formatos PDF y HTML sin software adicional, lo que permite que pueda ser utilizado como un editor WYSIWYG para crear y editar páginas web.



Calc : OpenOffice.org Calc es una hoja de cálculo similar a Microsoft Excel o Lotus 1-2-3. Calc ofrece una serie de características no presentes en Excel, incluyendo un sistema que, automáticamente define serie de gráficas, sobre la base de la disposición de la información del usuario. Calc también puede exportar hojas de cálculo para el formato PDF.



Impress : OpenOffice.org Impress es un programa de presentación similar a Microsoft PowerPoint o Keynote de Apple. Puede exportar presentaciones al formato SWF, permitiendo que se reproduzca en cualquier computadora con un reproductor de Flash instalado. También incluye la capacidad de crear archivos PDF, y la capacidad de leer archivos de Microsoft PowerPoint (ppt). Su formato original es ODP.



Draw : OpenOffice.org Draw es un editor de gráficos vectoriales y herramienta de diagramación, similar a Microsoft Visio y comparable en características a las primeras versiones de CorelDRAW. Ofrece «conectores» entre las formas, que están disponibles en una gama de estilos de línea y facilitan la construcción de los dibujos, como diagramas de flujo. Tiene características similares a la edición de software de escritorio como Scribus y Microsoft Publisher. Draw también puede exportar sus creaciones al formato PDF y al estándar SVG.



Base : OpenOffice.org Base es un programa de base de datos similar a Microsoft Access. OpenOffice.org Base permite la creación y manejo de bases de datos, elaboración de formularios e informes que proporcionan a los usuarios finales un acceso fácil a los datos. Al igual que Microsoft Access, es capaz de trabajar como un *front-end* para diversos sistemas de bases de datos tales como el de Access (JET), fuente de datos ODBC y MySQL/PostgreSQL. Base está disponible desde la versión 2.0.



Math : OpenOffice.org Math es una aplicación diseñada para la creación y edición de fórmulas matemáticas. La aplicación utiliza un lenguaje de marcado para crear las fórmulas. Dichas fórmulas pueden ser incorporadas dentro de otros documentos de OpenOffice.org, tales como los creados por Writer; permitiendo al usuario la posibilidad de incrustarlas al documento como objetos OLE. Math soporta múltiples fuentes y puede exportar fórmulas a los formatos de archivo ODF, PDF o MathML.

Extraído de : http://es.wikipedia.org/wiki/Open_Office



Ilustración 4: Logo Audacity

Audacity

Audacity es un editor de audio libre (licencia gratuita), fácil de usar y multilingüe para Windows, Mac OS X, GNU/Linux y otros sistemas operativos. Puede usar Audacity para: Grabar audio en vivo, convertir cintas y grabaciones a sonido digital o CD, editar archivos Ogg Vorbis, MP3, WAV y AIFF, cortar, copiar, unir y mezclar sonidos, Cambiar la velocidad o el tono de una grabación y mucho más.

Extraído de : <http://es.wikipedia.org/wiki/Audacity>

2 - Estado del arte

Hace ya muchos años, nació una de las primeras máquinas con un entorno interactivo, el Pong(1974), montado sobre una placa electrónica y un osciloscopio. Tan solo trataba de dos barras que se desplazaba en dos sentidos y en una única dirección, y un punto que iba de lado a lado. Esta máquina se controlaba simplemente con dos potenciómetros, componentes eléctricos que hoy en día los utilizamos principalmente para controlar el volumen de la radio, o de la minicadena, etc... En ella, dos personas podían jugar, interactuar y competir. Así surgió la primera máquina de uso público. Y a partir de ahí, el negocio de los videojuegos no ha parado de despegar.



Ilustración 5: Versión del Pong para la venta en los comercios.

Definido desde hace años como una alternativa de tiempo libre, actualmente esta por encima de las grandes industrias del cine y la música. Una industria de más de diez millones de dolares que influye en la vida tanto socialmente y culturalmente como económicamente. Se pueden encontrar desde películas basadas en antiguos títulos de consolas (*Tron*, Steven Lisberger, 1982) hasta videojuegos basados en películas, de hecho, una de las mejores maneras que hay hoy en día de promocionar una película es por su juego.

Dentro del ámbito musical, los jugadores pueden disfrutar de impresionantes bandas sonoras originales creadas exclusivamente para videojuegos (Jason Hayes - World Of Warcraft, Jesper Kid – Assassin's Creed) .



Ilustración 6: Cartel promocional del Assassins Creed

En el ámbito artístico, algunos diseñadores de videojuegos ven una oportunidad de hacer verdaderas obras de arte. Videojuegos en los que existen momentos que transmiten emociones, historias que sobrecogen, música que te pone los pelos de punta, ... ; todo ello forma parte de una poesía, todo debe de tener una armonía y una rima. En este momento es en el que un ingeniero que se dedica a programar y crear, se transforma en el camino que te guía hacia un mundo que él ha creado. El **videojuego** será el gran **arte popular** del **siglo XXI** .

La industria del videojuego está más fuerte que nunca en España, o al menos así lo parece si nos atenemos a las cifras presentadas por el sector. La asociación de editores y distribuidores de 'software' de entretenimientos, asegura que el consumo de videojuegos alcanzó los 863 millones de euros en el pasado año, por delante de otros sectores como el cine, el vídeo o la música grabada. Actualmente, más de 50 obras “de firma española” tiene una difusión internacional.

Es innegable esta industria ha cambiado tanto en estos últimos 25 años que, simplemente, los títulos que se jugaban en la infancia de muchas personas amantes de este genero no son comparables con los que ahora se disfruta. Este cambio se refiere principalmente al hecho de que los propios géneros en los que los juegos se catalogaban han desaparecido, generándose otro nuevo tipo de géneros que poco tienen en común con sus antepasados. El Beat ´em up clásico ha desaparecido como tal dando lugar al Hack´n slash, las videoaventuras tienen como abuelo lejano al popularísimo sandbox y los matamarcianos y las aventuras gráficas

prácticamente no existen, dejando como único e irreconocible superviviente al género deportivo. Por mucho que algunos se empeñen no se puede comparar Golden Axe con God of War, ya que sencillamente no son el mismo tipo de juego aunque mantengan ciertas similitudes. Simplemente no sería justo ni ético, y nos llevaría a una discusión sin sentido que podría durar años.

Pero si hay un género que claramente ha desaparecido de la faz de la tierra es el género arcade. Entiéndase “género arcade” como un conjunto de subgéneros que se caracterizaban por ofrecer al jugador una acción frenética e inmediata donde una partida no solía durar más de 10 minutos. Durante una década entera, millones de locales se llenaron de estas maquinas recreativas, que ofrecían a los clientes unos minutos de diversión. Pero con la entrada de las nuevas tecnologías (los ordenadores) estas maquinas se quedaron obsoletas. Otro de los némesis de las maquinas recreativas fue el boom de las consolas. Millones y millones de consolas se vendieron por todo el mundo en menos de 5 años, y compañías como Atari, Sega y Nintendo se hicieron con el mercado de los videojuegos.

Aún queda un culto a esta época de los que aman este tipo de acción que solo se encuentran en los juegos arcade y ellos son los únicos que mantienen el espíritu de una época que para muchos tiene muy buenos recuerdos.

3 - Justificación del proyecto.

El tema elegido para el videojuego está un poco abandonado ya que existen muchos del mismo genero, es el de "mata marcianos". Con un pequeño estudio y preguntando a la gente de mi alrededor, entre varias opciones del tema del videojuego (aventura gráfica, shooter, rpg, ...) seleccioné la de los mata marcianos por que le veía un tema mucho mas productivo y menos ambicioso que los demás. Existe un gran público que desea recuperar viejas sensaciones y les motiva volver a experimentar lo que llegaron a sentir con este genero (emoción, reflejos, tensión...).

La motivación de ser el creador de un videojuego debería de ser suficiente para los amantes de este mundo del arcade y que tengan conocimientos mínimos.

3.1 - Análisis de riesgos

Este proyecto no tiene ningún tipo de riesgos. Económicamente es imposible obtener perdidas ya que no es la creación de un producto comercial. No hay animo de lucro ni ningún tipo de inversión. Temporalmente tampoco existen riesgo a tener en cuenta, lo único es encontrarse con conceptos nuevos que requieran ser asimilados y probados hasta controlarlos y poder usarlos adecuadamente. Pero no existen fechas limites que cumplir ni por parte de ningún cliente ni de ningún jefe de producción.

En resumen, no hay riesgos destacables ya que se trata de un proyecto didáctico y cada uno tiene su tiempo de aprendizaje.

3.2 - Licencias

Todos los sonidos, texturas e imágenes han salido de un banco de recursos multimedia gratuito como <http://recursostic.educacion.es/bancoimagenes/web/>.

Este proyecto esta bajo licencia GNU (Ver apéndices)

3.3 - Estudio de costes

Presupuesto : Para un proyecto de tal magnitud suelen salir presupuestos bastante elevados de dinero, pero en esta caso ya contamos con el material necesario y suficiente para desarrollar el videojuego (PC u ordenador y material de papelería). Además, la colaboración de

ciertas personas sera completamente voluntaria y gratuita. Por lo tanto, en el aspecto económico no gastaremos dinero.

Estimación temporal : Cada parte del desarrollo de este proyecto tiene una estimación de coste temporal según la complejidad de la misma. A continuación se muestra un gráfico de barras mostrando la duración de cada una de las tareas en el mismo orden en el que se realizaron :



Ilustración 7: Gráfica coste temporal

Estos datos están basados en un grupo de trabajo muy reducido, en el que los equipos de diseño solo se ocupan de su trabajo y el resto de tareas son obligación del autor de este proyecto por motivos de carácter de trabajo de final de carrera. A continuación se describe las tres tareas de las que se ocuparan dichos equipos de diseño dirigidos por el jefe de proyecto, que en este caso es el autor del mismo :

La tarea de “Escribir el guión del juego” se ocupa Toni Cuevas (Guionista).

La tarea de “Producción de sonidos y música”, exactamente la producción y edición de la música se ocupa Jorge Macías (Editor de música).

De la tarea de “Creación de dibujos de personajes” se ocupa Lorena (Ilustradora).

Realmente, el tiempo estimado para la finalización de este proyecto esta calculado tomando en cuenta el volumen de trabajo y la cantidad de tareas distintas que tiene que realizar una única persona durante todo el desarrollo.

4 - Especificación y adaptación del motor

4.1 - Definición

Definición de motor gráfico o motor de videojuego : es un término que hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego. Del mismo modo existen motores de juegos que operan tanto en consolas de videojuegos como en sistemas operativos. La funcionalidad básica es proveer al videojuego de un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, streaming, administración de memoria y un escenario gráfico. El proceso de desarrollo de un videojuego puede variar notablemente por reusar o adaptar un mismo motor de videojuego para crear diferentes juegos.

4.2 - Análisis de requisitos

Para la creación de un producto software, es determinante establecer las condiciones y necesidades que ha de satisfacer el sistema.

El fundamento básico de esta tarea es la recopilación de información. A lo largo de la historia, se han empleado diversas técnicas como las entrevistas o los talleres y más recientemente el prototipado.

Se seguirá un esquema que permitirá describir los requisitos del sistema de una forma metódica y racional.

4.2.1 - Requisitos funcionales

La mayor parte del procesamiento recae sobre la cantidad de objetos 3D y los cálculos que realiza la inteligencia artificial. Será necesario optimizar dichos procesos para evitar ralentizaciones en el juego, asegurando que, como mínimo, el juego muestre 60 imágenes por segundo.

4.2.2 - Requisitos de rendimiento

De la misma forma, será muy importante optimizar el rendimiento de los procesos que dibujan las imágenes en pantalla, dado que se hará un uso intensivo de ellos. El sistema deberá responder en un tiempo lógico a los eventos generados por el usuario.

4.2.3 - Requisitos de diseño

La búsqueda del menor tiempo de respuesta determinará el diseño final.

4.3 - Adaptación

A continuación hay una pequeña lista de algunos de los motores más importantes que he investigado :



Ilustración 8: Logo Torque Game Engine

– Torque

Torque 3D fue creado por **GarageGames** para un desarrollo fácil de juegos, más rápido y más asequible. Se trata de un kit de desarrollo de software profesional ("SDK") que le ahorrará el esfuerzo necesario para construir un sistema de representación, la creación de redes de alta velocidad de varios jugadores, los editores en tiempo real, un sistema de scripting, y mucho más.

Como parte de 3D Torque, se tiene acceso completo al 100% del código fuente del motor. Esto significa que se puede añadir, modificar u optimizar todos los componentes del motor hasta el nivel más bajo de C++ de representación. Dicho esto, no hace falta ser un experimentado programador de C++ para utilizar 3D Torque. De hecho, **no es necesario saber C++ en absoluto**. Con TorqueScript y la colección de herramientas que se incluyen con Torque 3D, se puede construir juegos completos (de los muchos géneros diferentes) sin tocar una sola línea de código C++.

<http://www.garagegames.com/products/torque-3d/documentation/user/doc34>

– Fly3D

Fly3D 2.0 fue lanzado en 2002 y está incluido en el CD-Rom que acompaña a Watt y 3D de Policarpo Juegos, vol. 2 libro. Cuenta con avanzada tecnología de BSP, PVS, permite la detección de colisiones, nuevos efectos especiales como el humo y las explosiones, pre-cálculo de iluminación y de radiosidad, optimización de los mapas de iluminación, iluminación dinámica, sombras stencil, modelos de vértices y huesos de animación, y mucho más. Los Fly3D 2.0 descargables vienen con varios niveles de demostración y juegos, y están disponibles más de una docena de contribuciones de los usuarios. La versión actual es 2.10.

<http://devmaster.net/devdb/engines/fly3d-2-0#general-overview>



Ilustración 9: Logo Crystal Space

– Crystal Space

Crystal Space es un framework para el desarrollo de aplicaciones 3D escrito en C++ por Jorrit Tyberghein. Fue fundado el 26 de agosto del 1997. Crystal Space se usa típicamente como motor de juego pero el framework es más general y puede ser usado para cualquier tipo de visualización 3D. Crystal Space es muy portable y se ejecuta en Microsoft Windows, Linux, UNIX, y Mac OS X. Crystal Space es software de código abierto, licenciado bajo LGPL.

Puede usar opcionalmente OpenGL (en todas las plataformas), SDL (en todas las plataformas), X11 (Unix o GNU/Linux) y SVGALib (GNU/Linux). También puede usar rutinas de ensamblador usando NASM y MMX.

http://es.wikipedia.org/wiki/Crystal_Space

http://www.crystalspace3d.org/main/Main_Page



Ilustración 10: Logo Ogre3D

– **Ogre3D**

OGRE (Object-Oriented Graphics Rendering Engine, “motor de renderizado de gráficos orientado a objetos”) esta orientado a la escena, escrito en C++, es un motor 3D flexible diseñado para ser más intuitivo y fácil para los desarrolladores. La librería de clases abstrae la complejidad inherente al usar directamente librerías como Direct3D u OpenGL y proporciona una interfaz basada en objetos de mundo y otras clases intuitivas.

¿Es Ogre un motor de Juegos? No. OGRE puede ser (y de hecho es) usado para hacer juegos, pero OGRE está deliberadamente diseñado para proporcionar sólo una solución a la parte gráfica, mediante una clase global, otras características como el sonido, la conexión mediante red, inteligencia artificial (AI), colisiones, física, etc. Necesitará ser integrada mediante otras librerías, hay varios frameworks que hacen esto, tenemos una librería de integración para colisiones y física de muestra como ejemplo para nuestra distribución.

En resumen, aquí están los beneficios que la programación orientada objetos trae a OGRE:

- Abstracción : Interfaces comunes ocultan las diferentes implementaciones del API 3D y al Sistema Operativo.

- Encapsulación : Hay mucho manejo de estados y acciones específicas de contexto que son realizadas por la encapsulación del motor gráfico que me permite poner el código y los datos lo más cerca posibles de donde serán usados, lo cual hace el código más limpio y fácil de comprender y más modificable pues se evita la duplicación.

- Polimorfismo : Los métodos cambian según el tipo de objetos que uses, incluso si sólo aprendes una interfaz, ej: una clase especializada para manejar niveles internos completamente diferentes del control de la escena, pero parece idéntico a otras clases en el sistema y se llaman a los mismos métodos.

<http://www.ogre3d.org/>

<https://sites.google.com/site/ogreesp/caracteristicas>



Ilustración 11: Logo OpenSceneGraph

– OSG

El OpenSceneGraph es un código abierto de alto rendimiento de gráficos 3D juego de herramientas, utilizadas por los desarrolladores de aplicaciones en campos como la simulación visual, juegos, realidad virtual, visualización científica y el modelado. Escrito completamente en estándar de C++ y OpenGL se ejecuta en todas las plataformas Windows, OSX, GNU / Linux, IRIX, Solaris, HP-UX, AIX y sistemas operativos FreeBSD. El OpenSceneGraph se ha consolidado como el líder mundial en tecnología de escenario gráfico, muy utilizado en la vis-SIM, el espacio, científicos, gas-oil, los juegos y las industrias de realidad virtual.

El OpenSceneGraph es un Open Source y multiplataforma conjunto de herramientas gráficas para el desarrollo de aplicaciones de alto rendimiento de gráficos, como los simuladores de vuelo, juegos, realidad virtual y visualización científica. Se basa en el concepto de un Scenegrph, proporcionando un marco orientado a objetos en la parte superior de OpenGL. Esto libera al programador de la aplicación y optimización de llamadas de bajo nivel de gráficos y proporciona muchas utilidades adicionales para el desarrollo rápido de aplicaciones gráficas.

<http://www.openscenegraph.org/projects/osg/wiki/About/Introduction>



Ilustración 12: Logo Delta 3D

– **Delta3D**

Es un software de código abierto para juegos y simulaciones. Principalmente dirigido y apoyado por el modelador con entornos virtuales y de simulación. Alion Ciencia ha sido un importante contribuyente a las mejoras y características de este motor.

Delta3D se distribuye bajo la GNU Licencia Pública General Menor (LGPL). Los módulos externos tienen su propia licencia. Algunos módulos, como Qt, requieren que el usuario descargue e instale por separado para su uso con Delta3D.

Como multi-plataforma está adaptado para diferentes sistemas operativos y todos entornos de desarrollo son compatibles.

Delta3D se puede utilizar para crear un juego / simulación - que no es un "producto acabado" listo para el lanzamiento. Las pruebas, ejemplos, y otras aplicaciones de soporte se proporcionan. El usuario tiene disponibles todos estos recursos y referencias técnicas para crear su aplicación personalizada.

Con el fin de completar el uso de Delta3D se necesita un entorno de compilador / desarrollo. Además, se necesita un control de código fuente de aplicaciones para acceder al código fuente para las últimas versiones de Delta3D y las bibliotecas y módulos relacionados. Esto es necesario para hacer uso de los cambios en las bibliotecas de apoyo, así como las actualizaciones en curso a Delta3D que pueden no estar disponibles en el "paquete de lanzamiento", o en la documentación actual del producto.

<http://en.wikipedia.org/wiki/Delta3D>

<http://www.delta3d.org/index.php?topic=about>

Y este será el motor que voy a utilizar :



Ilustración 13: Logo Blender

– **Blender game engine**

El motor del juego Blender usa un sistema de gráficos de "ladrillos de la lógica" (una combinación de "sensores", los "controladores" y "actuadores") para controlar el movimiento y la visualización de los objetos en el motor. El motor del juego también se puede extender a través de un conjunto de Python enlaces. Versión 2.41 de Blender mostró una versión que se dedica casi enteramente al motor del juego. El audio es compatible con esta versión. Un nuevo sistema para la integración de los shaders de GLSL y la física de cuerpo blando se añadió en el comunicado de 2,48 para ayudar a la parte trasera del motor del juego en línea con los motores de juego modernos. Al igual que Blender, utiliza OpenGL , una capa de la cruz-plataforma de gráficos, para comunicarse con el hardware de gráficos.

El núcleo de la estructura del BGE son los bloques lógicos (Logic Bricks). El objetivo de los bloques lógicos es ofrecer un entorno visual fácil para diseñar aplicaciones interactivas sin conocer ningún lenguaje de programación. Hay tres tipos de bloques lógicos, Sensores, Controladores, Actuadores.

Para escribir juegos usando Python, el motor de juegos también tiene su propio Python API, separado del resto de Blender, el cual se puede usar para escribir scripts para controlar su juego. Esto está echo para crear un Python Controller y enlazarlo a un script de python.

http://en.wikipedia.org/wiki/Game_Blender

Se resume el concepto en un conjunto de funciones estructuradas para el manejo de objetos en 3D en un escenario que interactúan entre si. Estas funciones cambian según el motor que se elija, pero la mayoría de estos buscan abarcar todas las principales o básicas .

Iluminación, Sombras, Texturas, Shaders, Mallas, Escena de Gestión, Animación, Terreno, Física, Redes, Inteligencia, Artificial, Sonido, Herramientas y editores.

Según las necesidades de un vieojuego del genero arcade tipo matamarcianos, como es el de este proyecto, no son imprescindibles todas estas funciones, pero dado que ya he trabajado con el blender en una clase en la universidad y es una herramienta bastante sencilla y completa, se decidió utilizar el motor que lleva integrado.

En la siguiente tabla viene una comparativa de los motores gráficos nombrados anteriormente con estas funciones marcando con una “ ” si no es compatible con esta, con una “ok” si es compatible con esta y un “?” si no especifica compatibilidad :

Características	Motores						
	Torque	Fly3D	Crystal Space	Ogre3D	OSG	Delta3D	Blender
Iluminación	ok	ok	ok	ok	ok	ok	ok
Sombras	ok	ok	ok	ok	ok	ok	ok
Texturas	ok	ok	ok	ok	ok	ok	ok
Shaders	ok	ok	ok	ok	ok	ok	ok
Mallas	ok	ok	ok	ok	ok	ok	ok
Escenas de Gestión	ok	ok	ok	ok	ok	ok	ok
Animación		ok	?	ok	ok	ok	ok
Terreno	ok	ok	ok	?	ok	ok	ok
Física	?	ok	ok	?	ok	ok	ok
Redes		ok	?	ok		ok	?
Inteligencia Artificial	ok	ok	ok	?	?	ok	ok
Sonido	ok	ok	ok	ok	ok	ok	ok
Herramientas y editores	ok	ok				ok	ok

5 - Análisis

Aquí se recogen todos los datos y características que han sido obtenidos como resultado de reuniones con el equipo personal de guionistas y diseñadores que permitirán la realización de este ejercicio.

A la hora de realizar un videojuego no hay que seguir unas pautas comunes con la mayoría de los productos multimedia. La especificación de un videojuego puede variar, según el presupuesto, complejidad, tipo.... Este proyecto ha sido organizado de la siguiente manera:

5.1 - Selección de objetivos

5.1.1 - Publico Objetivo

Joven de entre 20 y 30 años. Aficionado a los juegos pero no con mucho tiempo para jugar. No hace falta que tenga ninguna habilidad especial. Estudiante o trabajador. Propietario de un ordenador o acostumbrado a usarlo.

5.1.2 - Objetivos globales

Generales :

- Entretener al público.
- Ser el primero de muchos juegos.
- Adquirir experiencia en la producción de videojuegos.

Aspectos educacionales :

- Aumentar la agilidad visual del jugador.
- Desarrollar los reflejos del jugador.

Fantasía que desarrolla el juego:

- Conquistar todos los territorios al vencer al jefe al final de cada zona en disputa.

Tipo de emoción que desarrolla el jugador:

- Recordar viejos juegos del mismo estilo (matamarcianos).

5.1.3 - Objetivos locales

- Destruir ciertas naves especiales que nos dejen un premio o más vida, o un escudo que dure cierto tiempo o un incremento del arma de nuestra aeronave.
- Conseguir la puntuación máxima registrada en la máquina.
- Llegar hasta el final de la historia.
- No perder todas las vidas.

5.2 - Género

Los videojuegos se pueden clasificar como un género u otro dependiendo de su representación gráfica, el tipo de interacción entre el jugador y la máquina, la ambientación y su sistema de juego, siendo este último el criterio más habitual a tener en cuenta.

Si se quiere hacer un videojuego, es bueno pensar de qué género va a ser el juego. Haciendo un pequeño repaso a los catálogos y cientos de listas en Internet, he podido hacer una lista de géneros bastante común :

- Acción. A su vez, este puede dividirse en :
 - Beat'em up: Juegos de lucha a progresión. Mientras luchas con gran cantidad de enemigos, avanzas sobre unos niveles. (Street of rage, Double Dragon, Viewtiful Joe...)



Ilustración 14: Street of rage

- Lucha: Clásico género de peleas. (Street Fighter, Fatal Fury, Tekken..)



Ilustración 15: Street fighter



Ilustración 16: Tekken

- Disparos: Las acciones básicas son mover al personaje y usar un arma. Son juegos con calidad de guión bastante bajo pero con un alto contenido de acción. Existen juegos de disparos en primera persona (Doom, Halo, Half Life) y en tercera (Rayman, Tomb Raider), una tercera subclase sería los shoot'em up (Metal Slug, Contra, r-Type).

- Plataformas: Basta con nombrar un par de clásicos (Mario bros y Sonic)



Ilustración 18: Mario Bros 3



Ilustración 17: Sonic (Mega Drive)

- Arcade: Simplicidad de acción rápida y de alta jugabilidad. Juegos repetitivos. La gran mayoría de estos arcades estaban pensados para su edición en maquinas recreativas.

- Conducción. Solo por la cantidad de juegos que existen de este tipo se merece un género propio. Carrera de coches, de naves, modificar tu vehículo, lanzar objetos en mitad de carrera. (Need for Speed, Mario Kart, Out Run...)



Ilustración 19: Need for speed Underground 2



Ilustración 20: Our Run

- Deportivo: Simulan juegos de deporte real como el golf, fútbol, hockey, tenis, etc...



Ilustración 21: Pro Evolvion Soccer

- Puzzle: Tienes que pensar y agilizar el pensamiento. Tras realizar ejercicio la dificultad aumenta para desarrollar nuestra habilidad mental. Últimamente se añade a este genero las aventuras gráficas (point and click).



Ilustración 22: Monkey Island



Ilustración 23: Run Away

- Rol: Se caracteriza por la interacción prácticamente total con el personaje, una historia profunda y una evolución del personaje a medida que la historia avanza. En este genero se encuentra los tipos RPG (juegos con batallas por turnos aunque actualmente sean a tiempo real) . La mayoría de estos están basados en novelas de mundos fantásticos; y también encontraremos los MMORPG (Massive Multiplayer Online RPG), RPG jugados por Internet donde se interactua con otros usuarios o jugadores de cualquier parte del mundo.
- Simulador: Se caracteriza en marcar un aspecto de la vida real. Consiste en inmiscuir al jugador hasta hacerlo creer que lo que esta pasando es real. Este género se divide en tipos de simulación :
 - Musical: desde el clásico Karaoke hasta la ultimas versiones del rock band donde puedes jugar a tener tu grupo de música.



Ilustración 24: Rock Band

Construcción: Debes de controlar la gestión de recursos y de tiempo para conseguir el objetivo del juego que puede ser construir un parque de atracciones, ciudades, hospitales, etc..

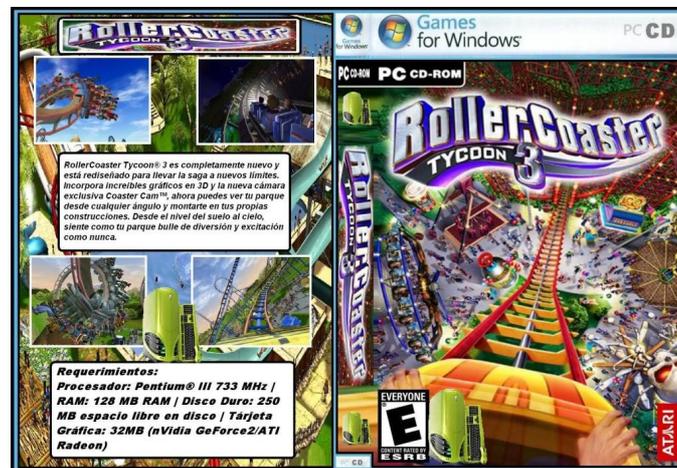


Ilustración 25: RollerCoaster Tycoon

- Vida: Controlar al personaje con capacidades y emociones humanas y controlar todos los aspectos de su vida.



Ilustración 26: The Sims

Cabe decir que algunos juegos incluyen elementos de distintos géneros. Por ejemplo, podríamos hacer un juego de fútbol pero que no se jugara, solo sería un juego de gestión deportiva (gastos, fichajes, entrenamientos), así este juego contendría elementos deportivos(fútbol) , algo de estrategia (gestión de recursos) y simulador (los resultados de tus decisiones se reflejan en el juego como sería en la realidad).

Este juego será un “Arcade”, ya que solo quiero que sea un juego eventual, sin la necesidad de guardar partidas ni invertir muchas horas en finalizar el juego.



Ilustración 27: Recreativa Tron

5.3 - Guión

Al principio, cuando todavía no había palabras para describir el largo paso de los años, en el origen de la tierra, hubo una batalla entre dos seres de poder cósmico. El origen fue una disputa por el sentido y la utilidad que querían darle a una de sus creaciones: la Humanidad. Para combatir se repartieron a los humanos y con ellos formaron dos ejércitos que se enfrentaron entre sí. Los ganadores se quedaron la tierra, los perdedores fueron exiliados al confín del espacio.

Pero ya han pasado miles de años. Cada civilización ha evolucionado por su lado. Los Humanos olvidaron con facilidad a sus Dioses pero los Creadores no se olvidan entre sí y en las sombras controlan su creación.

5.4 - Especificación conceptual.

5.4.1 - Entrada y salida.

- Entradas : teclado.

- Salidas : Gráfica y audio.
- Línea temporal de acontecimientos de cada zona : No soportar bifurcaciones, pero si la posibilidad de elegir la zona (fase).

5.4.2 - Mapa de controles

Diagrama que muestra las entradas de control, acción asociada y contexto en el que se produce la acción

Acción	Control	Contexto
Izquierda	←	Todo
Derecha	→	Todo
Arriba	↑	Todo
Abajo	↓	Todo
Disparar	Espacio	Juego
Bomba	Alt	Juego
Iniciar	Enter	Todo
Salir	Esc	Todo

5.4.3 - Principios de Norman

Los principios del diseño ayudan a los diseñadores a explicar y mejorar su trabajo. Tienen su origen en la teoría, la experiencia y el sentido común. Cuando se ponen en práctica para evaluar productos o prototipos se les denomina principios de usabilidad o heurísticos.

- Visibilidad : Partes relevantes deben percibirse
- Mapeados : Relaciones comprensibles entre controles y acciones.
- Posibilidades : Usos percibidos de un objeto.
- Restricciones : Bloqueo de las paredes para que no se puedan atravesar.
- Realimentación : Informar de acontecimientos u objetivos alcanzados

Diseñado con comportamientos generales, los niveles están preconfigurados y la resolución del juego se consigue mediante la comprensión del sistema.

5.4.4 - Método de aprendizaje : Condicionamiento Operante

El condicionamiento operante, también llamado condicionamiento instrumental, es una forma de aprendizaje mediante la que un sujeto tiene más probabilidades de repetir las formas de conducta que conllevan consecuencias positivas y, por el contrario, menos probabilidades de repetir las que conllevan consecuencias negativas. El condicionamiento operante es un tipo de aprendizaje asociativo que tiene que ver con el desarrollo de nuevas conductas en función de sus consecuencias, y no con la asociación entre estímulos y conductas como ocurre en el condicionamiento clásico.

5.5 - Tópicos

Las acciones, sonidos, situaciones, estilos de controles, tipos de pantalla, desarrollo de la historia, tipos de armas... Cada videojuego tiene una serie de características que identifican a ese juego como único o perteneciente a un género. Para este videojuego se han especificado las siguientes :

- Combate espacial en vertical.
- Invasión y defensa del planeta tierra.
- Gran cantidad de naves y disparos contra el jugador.
- Jefe al final de cada territorio.
- Selección de varios aviones para manejarlos.
- El armamento de los aviones se pueden cambiar y mejorar.
- Música rápida (d'n'b, heavy, etc..).

En el apéndice A (ficha del videojuego) se describe con detalle cada uno de los tópicos listados arriba.

5.6 - Pre-programación

Al desarrollar Far Exile he optado por utilizar el modelo incremental dada la magnitud (para mí) del proyecto. He querido siempre tener una idea de cómo el proyecto ha ido avanzando y una vez obtenido un resultado claro, analizarlo e incrementarlo.

Se puede dividir el proyecto en los tres incrementos que se explican a continuación. Hay que tener en cuenta que se realizó el proyecto entre el primer y el segundo incremento durante algunos meses.

Incremento 1: Personaje

En este primer incremento se realizará un primer ejecutable donde se muestra el personaje. Primeras pruebas de movimiento de la nave y posibilidad de disparo.

Incremento 2: lógica y física del juego y escena básica

Una vez obtenidos los conocimientos básicos que necesitaba continué con el desarrollo del juego. Diseñé para el segundo incremento una pequeña fase con algunos obstáculos para probar las colisiones de las balas.

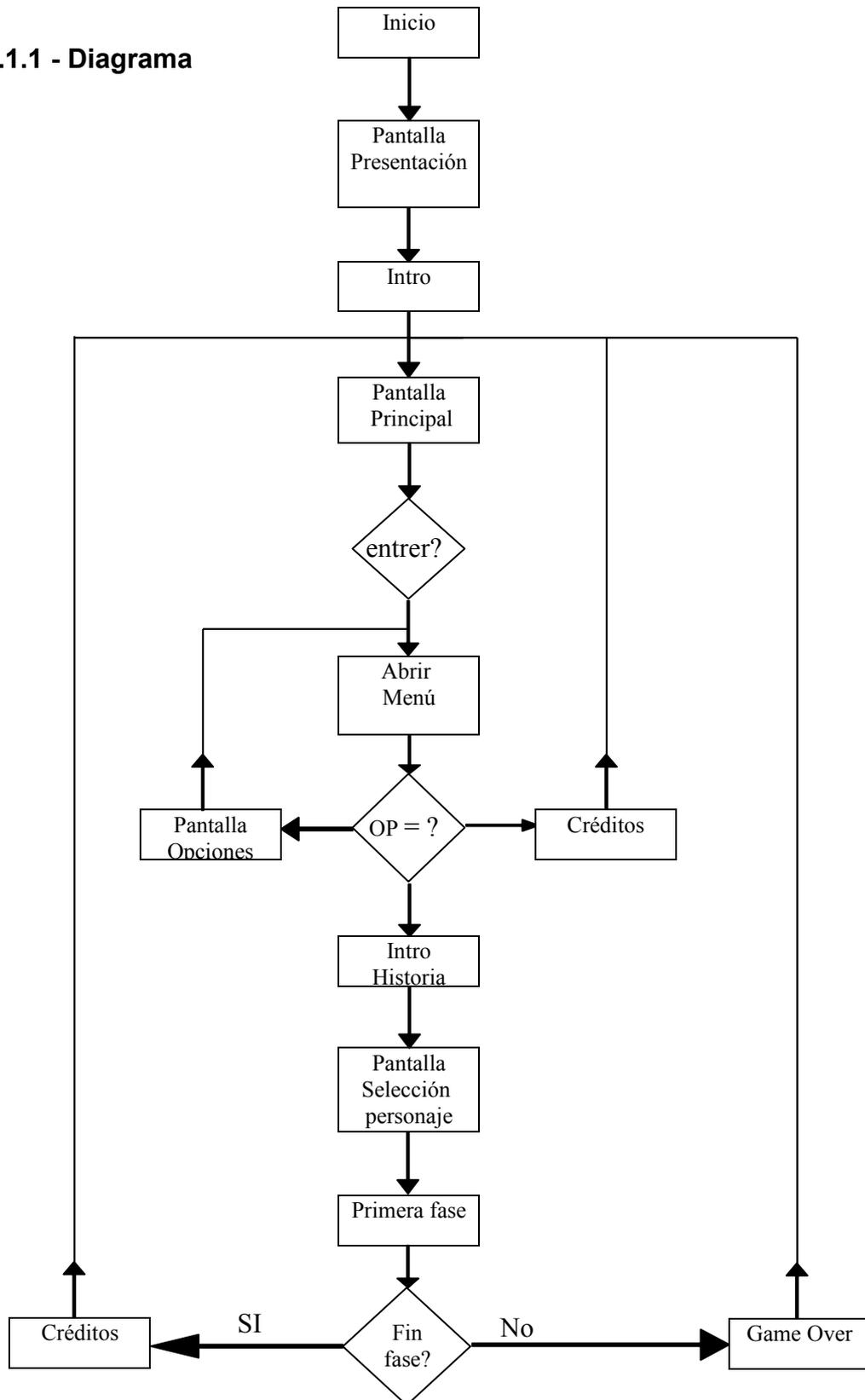
Incremento 3: Música y sonido

Después de observar que los dos incrementos anteriores funcionaban perfectamente, me dediqué a reproducir la música dentro de cada una de las escenas y a hacer que los objetos según las acciones o situaciones reproduzcan sonidos como de colisiones, explosiones, etc...

6 - Diseño

6.1 - Bucle principal

6.1.1 - Diagrama



Este es el diagrama de flujo de la demo, es una versión incompleta y reducida del producto final, por lo tanto habrán partes de este diagrama que no se realizarán.

6.1.2 - Módulos

- **Inicio**

Carga de todas las variables locales.

- **Pantalla presentación**

Se carga el escenario "logo". Al finalizar se carga el siguiente modulo.

- **Intro**

Se carga el escenario "intro". Al finalizar se carga el siguiente modulo.

- **Pantalla principal**

Se carga el escenario "menú".

- **Enter?**

Al pulsar la tecla enter o intro en la pantalla principal se pasa al siguiente modulo.

- **Abrir menú**

En el mismo escenario se añaden las opciones para comenzar a jugar.

- **Opción?**

Según la opción seleccionada en el modulo anterior se pasa a la pantalla de opciones de juego (volumen, dificultad, etc,,), o al siguiente modulo (intro historia).

- **Pantalla opciones**

Se muestra el escenario "opciones".

- **Créditos**

Se muestra el escenario de créditos. Al finalizar o apertar la tecla "enter" se pasa al siguiente modulo.

- **Intro historia**

Se muestra el escenario “historia”, una pequeña introducción a la historia.

- **Pantalla de selección de personajes**

Se muestra el escenario “selección de nave”, una lista de los personajes que se puede seleccionar para jugar.

- **Primera fase**

Se carga la primera fase con la nave seleccionada en el modulo anterior.

- **Fin fase?**

Si se acaban las vidas se pasa al modulo “game over”, si el jefe de la fase es derrotado se pasa al siguiente modulo.

- **Créditos**

Se muestra el escenario “créditos”. Al finalizar se pasa al siguiente modulo.

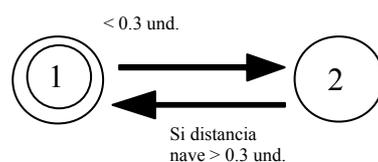
- **Fin**

En cualquier modulo si se pulsa la tecla “Esc” se cierra el juego.

6.2 - Inteligencia Artificial

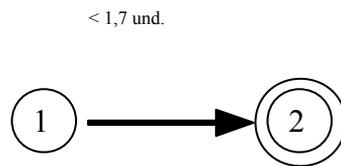
Aquí tenemos cada uno de los objetos que tiene algún sensor y tienen ejecutar alguna acción en respuesta cuando se active. Simplemente se muestra una diagrama de estados y a continuación se describe las acciones a ejecutar en cada uno de estos.

6.2.1 - Malo 1



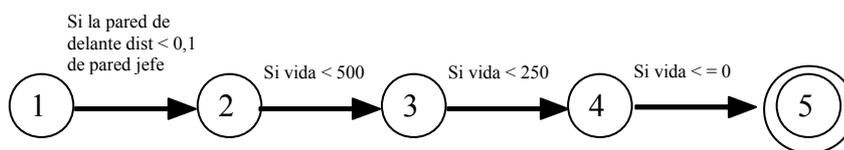
- Estado 1 : Ninguna acción. Eliminar objeto malo1 si bala o nave está a una distancia < 0.1 und.
- Estado 2 : Dispara bala3 y se mueve a la izq.

6.2.2 - Lanzacohetes



- Estado 1 : Ninguna acción. Viene con una animación IPO del modelo de la librería.
- Estado 2 : Dispara bala2 (misil).

6.2.3 - Jefe



- Estado 1 : Ninguna acción. Propiedad vida = 750.
- Estado 2 : Animación IPO (movimiento repetitivo de izquierda a derecha). Mensaje "Dispara1" (bala3).
- Estado 3 : Animación IPO (movimiento repetitivo de izquierda a derecha). Mensaje "Dispara2" (bala2).
- Estado 4 : Animación IPO (movimiento repetitivo de izquierda a derecha). Mensaje "Dispara3" (bala4).
- Estado 5 : Mensaje "finalizar_fase". Eliminar objeto.

7 - Implementación

7.1 - Códigos

Puesto que la herramienta de lógica de Blender Game Engine me permitía crear una pequeña programación sin escribir nada de código, explicaré esta sección con pseudocódigo. Utilizaré un lenguaje común para describir todas las condiciones y las acciones llevadas por los objetos en el juego, los nombres de los mensajes enviados, sonidos, y propiedades de los objetos son los mismos que los utilizados en el proyecto blender para su mejor comprensión.

El juego esta creado por escenarios y cada uno de ellos representa una pantalla del juego (Intro, primera fase, game over, etc...). Estos escenarios están compuestos por objetos los cuales tienen unos sensores. La programación en blender se realiza asignando distintos sensores a los objetos de manera que reaccionan ejecutando una acción ya sea reproducir un sonido, cambiar una variable de valor, eliminar objetos, reproducir animaciones, etc... (Este tema esta explicado en el apendice B).

7.1.1 - Primera fase

Nave1 :

– **Variables locales :**

- semáforo = true.
- vida = 2.

– **Eventos :**

- Si bala 2 o bala3 o bala4 o malo1 o piedra está a distancia < 0.1 → Mensaje “golpe” y “kitavidas” y sonido “golpe de chapa”.
- Si objeto1 está a distancia < 0.1 → Mensaje “mejora_arma” y sonido “mejora arma”.
- Siempre que el semáforo = true → movimiento en eje y +0,1.
- Si pulsar tecla “cursor arriba” y distancia a “pared arriba” > 0.1 → movimiento en eje y +0,1.

- Si pulsar tecla “cursor abajo” y distancia a “pared abajo” > 0.1 → movimiento en eje y -0,1.
- Si pulsar tecla “cursor derecha” y distancia a “pared derecha” > 0.1→ movimiento en eje x +0,1.
- Si pulsar tecla “cursor izquierda” y distancia a “pared izquierda” > 0.1→ movimiento en eje x -0,1.
- Si mensaje “kitavidas” → sonido “golpe de chapa” y vida-1.
- Si mensaje “comenzar_fase” → animacion IPO frames 1-300.
- Si mensaje “finalizar_fase” → animacion IPO frames 301-600.
- Si mensaje “semáforo rojo” → semáforo = false.
- Si vida = 0 → quitar la escena “hud” superpuesta y cargar escena “game_over”.
- **Objetos asociados :**
 - Empty 1 y 2 :
 - Variables local arma = 1.
 - Si mensaje “mejora_armas” → arma = 2.
 - Si mensaje “empeora_armas” → arma = 1.
 - Si pulsar tecla “Espacio ” → si mejora = 1 → crea bala (empty1 y empty2) y sonido “disparo”.
 - Si pulsar tecla “Espacio ” → si mejora = 2 → crea bala (empty1 y empty2), crea bala_izq (empty1), crea bala_der (empty2) y sonido “disparo”.

Cámara :

- **Variables locales :**
 - semáforo = true.

- **Eventos :**
 - siempre escena "hud" superpuesta.
 - una vez sonido "iniciar fase" y mensaje "comenzar fase".
 - Siempre que el semáforo = true → movimiento en eje y +0,1 y sonido "fase1".
 - Si mensaje "destruir piedra" → sonido "romper piedra".
 - Si mensaje "destruir malo" → sonido "explosión pekeña".
 - Si mensaje "golpe" → animación IPO.
 - Si mensaje "semáforo rojo" → semáforo = false y sonido "jefe".
- **Objetos asociados :**
 - Pared derecha, pared izquierda, pared detras y borrador.
 - Pared delante :
 - Si distancia "pared_fin" < 0.1 → mensaje "semáforo rojo".
 - Si distancia "peligro" < 0,1 → mensaje "peligro".

Piedras :

- **Eventos :**
 - Si distancia a bala < 0,1 → mensaje "destruir piedra" y eliminar objeto.

Bala (proyector del jugador) :

- **Eventos :**
 - Si distancia a piedra o a malo1 < 0,1 → eliminar objeto.
 - Si distancia a jefe < 0,1 → mensaje "impacto jefe" y eliminar objeto.
 - Siempre movimiento local en eje y +0,1.

Bala2 (misil perseguidor) :

- **Eventos :**
 - Si distancia a bala $< 0,1$ → eliminar objeto y mensaje “destruir malo”.
 - Si distancia a nave $< 0,1$ → mensaje “golpe” y eliminar objeto.
 - Siempre movimiento local en eje y $+0,01$ y seguir a “nave”.
 - Si mensaje “finalizar fase” → eliminar objeto.

Bala3 (proyectil enemigos) :

- **Eventos :**
 - Si distancia a nave $< 0,1$ → mensaje “golpe” y eliminar objeto.
 - Siempre movimiento local en eje y $+0,01$.
 - Si mensaje “finalizar fase” → eliminar objeto.

Bala4 (Cañon del jefe) :

- **Eventos :**
 - Si distancia a nave $< 0,1$ → mensaje “golpe”.
 - Siempre movimiento local en eje y $+0,03$.
 - Si mensaje “finalizar fase” → eliminar objeto.

Objeto1 :

- **Eventos :**
 - Si distancia a nave $< 0,1$ → mensaje “mejora arma” y eliminar objeto.

Jefe1 :

- **Variables locales :**
- Vida = 750.
- **Eventos :**
- Estado1:
 - Si mensaje “semaforo rojo” → estado = 2.
- Estado2:
 - Siempre → mensaje “dispara” y IPO de los frames 1 al 300.
 - Si vida = 700 → mensaje “kita_barra1”.
 - Si vida = 650 → mensaje “kita_barra2”.
 - Si vida = 600 → mensaje “kita_barra3”.
 - Si vida = 550 → mensaje “kita_barra4”.
 - Si vida = 500 → estado = 3 y mensaje “kita_barra5”.
 - Si mensaje “impacto_jefe” → Vida – 1.
- Estado3:
 - Siempre → mensaje “dispara2” y IPO de los frames 1 al 300.
 - Si vida = 450 → mensaje “kita_barra6”.
 - Si vida = 400 → mensaje “kita_barra7”.
 - Si vida = 350 → mensaje “kita_barra8”.
 - Si vida = 300 → mensaje “kita_barra9”.
 - Si vida = 250 → estado = 4 y mensaje “kita_barra10”.

- Si mensaje "impacto_jefe" → Vida - 1.
- Estado4:
 - Siempre → mensaje "dispara3" y IPO de los frames 1 al 300.
 - Si vida = 200 → mensaje "kita_barra11".
 - Si vida = 150 → mensaje "kita_barra12".
 - Si vida = 100 → mensaje "kita_barra13".
 - Si vida = 50 → mensaje "kita_barra14".
 - Si vida = 0 → estado = 5 y mensaje "kita_barra15".
 - Si mensaje "impacto_jefe" → Vida - 1.
- Estado5:
 - Una vez → sonido "explosión pequeña", sonido "objetivo_cumplido", mensaje "finalizar_fase" y eliminar objeto.
- **Objetos asociados :**
 - Empty1 :
 - Estado1 : Si mensaje "dispara" → estado = 2.
 - Estado2 : Cada 2 segundos crear objeto "bala3".
 - Empty2 :
 - Estado1 : Si mensaje "dispara" → estado = 2.
 - Estado2 : Cada 2 segundos crear objeto "bala3".
 - Empty3 :
 - Estado1 : Si mensaje "dispara" → estado = 2.
 - Estado2 : Cada 2 segundos crear objeto "bala3".
 - Empty4 :

- Estado1 : Si mensaje “dispara” → estado = 2.
- Estado2 : Cada 2 segundos crear objeto “bala3”.
- Empty5 :
 - Estado1 : Si mensaje “dispara2” → estado = 2.
 - Estado2 : Cada 2 segundos crear objeto “bala2”.
- Empty6 :
 - Estado1 : Si mensaje “dispara2” → estado = 2.
 - Estado2 : Cada 2 segundos crear objeto “bala2”.
- Empty7 :
 - Estado1 : Si mensaje “dispara3” → estado = 2.
 - Estado2 : Cada 2 segundos crear objeto “bala4”.
- Sphere.004:
 - Siempre apuntar a “Nave 1.1”.

7.1.2 - Introducción

Cámara :

- **Eventos :**
 - Siempre sonido “logo1”.
 - Si pulsar tecla “Enter” o pasan 12 segundos → cambia a escena “Inicio”.

7.1.3 - Inicio

Cámara :

- **Eventos :**
 - Siempre sonido “Inicio”.

- Si pulsar tecla "Enter" → cambia a escena "controles".

7.1.4 - Game Over

Cámara :

- **Eventos :**
 - Siempre sonido "game over".
 - Si pulsar tecla "Enter" → cambia a escena "Inicio".

7.1.5 - Créditos

Cámara :

- **Eventos :**
 - Siempre sonido "game over".
 - Si pulsar tecla "Enter" → cambia a escena "Inicio".

7.1.6 - Controles

Cámara :

- **Eventos :**
 - Si pulsar tecla "Enter" o pasan 9 segundos → cambiar escena "primera fase".

7.1.7 - HUD

Cámara :

- **Variables locales :**
 - vidas = 2.
 - vida_jefe = 750.

– **Eventos :**

- Si mensaje “golpe” y vidas = 2 → mensaje “tekedan2vidas”, mensaje “kitavida” y vidas – 1.
- Si mensaje “golpe” y vidas = 1 → mensaje “tekeda1vida”, mensaje “kitavida” y vidas – 1.
- Si mensaje “impacto_jefe” → vida_jefe -1.

Plane :

– **Eventos :**

- Comienza visible y a los 5 segundos eliminar objeto.

– **Objetos asociados :**

- Un plano y 3 textos sin eventos. Visibilidad heredada de “Plane”.

Barra de vida del jefe:

– **Eventos :**

- Siempre invisibles.
- Si mensaje “semaforo_rojo” → visible.
- Si mensaje “kita_barraX” → eliminar barraX.

– **Objetos asociados :**

- barra1, barra2, barra3, barra4, barra5, barra6, barra7, barra8, barra9, barra10, barra11, barra12, barra13, barra14, barra15.

Contador :

– **Variables locales :**

- int Text = 0.

- **Eventos :**
 - cada centésima de segundo → text + 1.
 - si mensaje “finalizar_fase” → stop contador.

Mensaje1:

- **Variables locales :**
 - String Text : “Misión Cumplida”.
- **Eventos :**
 - Siempre invisible.
 - Si mensaje “finalizar_fase” → visible.
- **Objetos asociados :**
 - Mensaje2. Visibilidad heredada de “mensaje1”.

Mensaje3 :

- **Variables locales :**
 - String Text : “PELIGRO”.
- **Eventos :**
 - Siempre invisible.
 - Si mensaje “peligro” → visible.

Marcador :

- **Variables locales :**
 - int marcador = 0.

- **Eventos :**
 - Si mensaje “destruir_piedra” → marcador + 20.
 - Si mensaje “destruir_malo” → marcado + 100.
 - Si mensaje “impacto_jefe” → marcador + 10.

Vida1 :

- **Eventos :**
 - Si mensaje “tekeda1vida” → eliminar objeto.

Vida2:

- **Eventos :**
 - Si mensaje “tekedan2vidas” → eliminar objeto.

7.2 - Problemas concretos

Durante la producción del videojuego, han surgido varios problemas, algunos se solucionaron en cuestión de minutos, y otros fueron mas difíciles ya que no se trataba de algún concepto no comprendido, si no que eran errores de diseño que han surgido debido a no conocer por completo las herramientas de edición o por realizar un mal planteamiento de la física del motor.

Entre los problemas encontrados, estos tres son los mas importantes y a tener en cuenta cuando se crea un videojuego de este estilo con un entorno 3D como es el Blender.

7.2.1 - Colisiones

El motor de videojuego del blender proporciona un sistema de física bastante realista. Y la utilización de esta física conlleva una serie de problemas que podríamos no tener en cuenta.

En un sistema físico, predominan las colisiones entre los objetos que son actores dinámicos (no estáticos) y las fuerzas como la de la gravedad. En la gravedad es simplemente un movimiento acelerado en el eje -z sobre cada objeto. Esto es un problema ya que las naves van

volando haciendo resistencia a esta gravedad. En este entorno 3D se colocó un suelo falso e invisible sobre el cual se posaban todos los objetos flotantes para que no se cayeran al suelo. Este problema realmente tiene otra solución, y es simplemente poner a 0 la aceleración (fuerza de gravedad).

En cuanto al sistema de colisiones, todo iba perfecto aparentemente. Si dejaba la gravedad en su valor inicial, las bala se veían atraídas hacia el suelo y por lo tanto no colisionaban con los objetos por que estaban a alturas distintas. Eliminando la gravedad este problema desapareció, pero surgió otro. Ahora cuando la nave colisionaba con cualquier objeto, al no haber gravedad, este se desplazaba hacia arriba o hacia abajo del objeto, quedándose estático en esa altura (eje z) y evitando todas las colisiones (cosa no deseada).

Al fin se obtuvo una solución que evitaba utilizar el sistema de colisiones y no tener problemas como los anteriores. Sustituir las colisiones por sensores de proximidad y transformando todos los objetos 3D del juego en estadios (no dinámicos). Resulta que si se utiliza sensores de proximidad con valores muy muy pequeños, se puede simular perfectamente la detección de una colisión sin que se llegue a dar esos desplazamientos tan indeseados anteriormente explicados.

Justifico esta solución con el argumento de que para este genero de videojuegos no es necesario un sistema de colisiones como lo sería en uno de simulación de conducción, o un juego de plataformas o en unos de los RPGs actuales en primera persona.

7.2.2 - Limites de pantalla

Los videojuegos de scroll tienen una peculiaridad, y es que el movimiento del personaje es independiente al de la cámara (en otros juegos la cámara sigue estos), entonces hay que aplicar algún método para no sobrepasar los limites de la pantalla, para evitar que el jugador pierda de vista su personaje.

La primera solución que se aplicó fue la creación de paredes invisibles que evitaba atravesaras el personaje por que detectaba una colisión. Pero al eliminar estas colisiones transformando los objetos a estáticos, ya no existían estas colisiones, con lo que se tuvo que aplicar otro método. Este consistía en limitar el movimiento de la nave en cada uno de sus ejes siempre que no estuvieran a una distancia muy inferior para simular las colisiones eliminadas. Esto se consiguió añadiendo un sensor de proximidad al mismo sensor de teclado que iniciaba el movimiento con un operador "and" (si teclado "->" y si distancia < 0,1 de "pared_derecha").

Con esto además, al no haber colisiones, no existían ni rebotes ni desplazamientos indeseados, con lo que la nave se detiene en los bordes de la pantalla de una manera mas limpia.

7.2.3 - Reproducción de sonidos

En blender para reproducir un sonido, se debe de hacer desde una acción de un objeto. Esta asignación se realiza en el editor de lógica del blender.

En ocasiones surgían problemas con los objetos que desaparecían por una colisión. Resulta que si un objeto reproduce un sonido y a la vez desaparece, el sonido deja de escucharse debido a que el objeto hacia de "altavoz".

Para solucionar este problema se decidió que el objeto mismo no reproduzca el sonido, sino que mande un mensaje para que otro objeto que no vaya a desperecer sea el que reproduzca el sonido. Con este se solucionó el problema.

8 - Conclusiones

8.1 - Desarrollo del proyecto

Durante el desarrollo del proyecto he conseguido básicamente mi principal objetivo: obtener los conocimientos básicos para crear un videojuego y saber dónde y cómo encontrar la información necesaria para ello.

He aprendido cómo utilizar diversas aplicaciones de código abierto que también pueden ser necesarias en otro tipo de proyectos. Las diferentes tecnologías utilizadas me han parecido en general muy buenas y creo que se pueden obtener resultados muy buenos con ellas. A continuación describo mi opinión de estas tecnologías :

- Blender : conocía 3D-Studio Max desde hace tiempo y comencé a utilizar Blender hace 2 años en una clase de imagen sintética de la universidad. Aunque al principio parece muy complejo dado su interfaz de usuario tan diferente con el tiempo es posible trabajar con él de una forma rápida y con resultados profesionales.
- GIMP : me ha parecido un programa con muchas posibilidades. La mayor crítica que pueda hacerle es su compleja interfaz de usuario y la no existencia de un modo con una única ventana, aunque al parecer este último se arreglará en la próxima versión.
- Audacity : Audacity me ha permitido de una forma fácil modificar los archivos de música y sonido para el videojuego. Aunque no he trabajado muy a fondo con él me ha parecido una herramienta muy intuitiva y con muchas opciones.

8.2 - Posibles ampliaciones

- Nuevas fases: Sería posible añadir nuevos niveles con escenarios diferentes. Esta sería una de las ampliaciones más difíciles puesto que sería necesario crear el escenario 3D, buscar la música adecuada, añadir propiedades a los objetos 3D, crear la lógica. Pero usando el nivel ya creado a modo de ejemplo, facilitaría mucho la creación de nuevos niveles.
- Menú de selección de personaje : Consistiría en crear un escenario nuevo en el proyecto de blender donde colocar un panel de selección. En este panel se mostrarán 8 naves distintas, cada una de ellas seleccionables. Los diseños de estas naves están descritos y algunos ya ilustrados. Después, esta pantalla serviría para cargar una nave

u otra (según se haya elegido) en las fases del juego. Sería necesario programar con python para realizar esta ampliación.

- Menú de selección de pantalla : Al igual que en la pantalla de selección de personaje, en esta obtendríamos el mismo efecto (menú de selección) pero con la fase donde queramos jugar. Sería necesario programar con python para realizar esta ampliación.
- Opciones del juego : Nivel de dificultad, cantidad de vidas, cantidad de continuaciones, volumen de la música y de los efectos.

9 - Agradecimientos

A mis padres por todo el esfuerzo que han realizado para permitirme llegar hasta aquí.

A mi hermano, Alex Mateo, Hector Chiner, “Skippy” y “Teresuken” por ser fuentes de inspiración y de ideas frescas.

A “Padawan” y a otros amigos por su apoyo continuo y ánimos para realizar este proyecto.

A Toni por crear un guión idóneo para este proyecto.

A Jorge por crear la música del videojuego.

A Lorena por currarselo tanto con sus ilustraciones.

Mis agradecimientos también a la comunidad del software libre y de obras copyleft en general sin la cual este proyecto habría sido imposible de realizar para mí.

A Ramón Molla por las facilidades que me ha brindado desde el primer momento, así como su ayuda, atención e interés en dirigir este proyecto.

10 - Apéndices.

A. Ficha del videojuego.

1. Título y versión.

- FAR EXILE 0.9 - <http://farexile.jimdo.com/downloads/>

2. Referencias.

- Arcade : Juego ocasional, no tiene sistema de guardado de partidas. Corto, pero intenso. Sin historias complicadas.

3. Referencias.

- Aerofighters 2.

http://www.arcade-museum.com/game_detail.php?game_id=6818



Ilustración 28: Aero Fighters 2 (neo geo)



Ilustración 29: Aero fighters



Ilustración 30: Aero fighters selección de personaje)



Ilustración 31: Aero fighters

- Ikaruga.

http://www.arcade-museum.com/game_detail.php?game_id=10841



Ilustración 32: Ikaruga menú



Ilustración 33: Ikaruga

- 1942.

http://www.arcade-museum.com/game_detail.php?game_id=6766

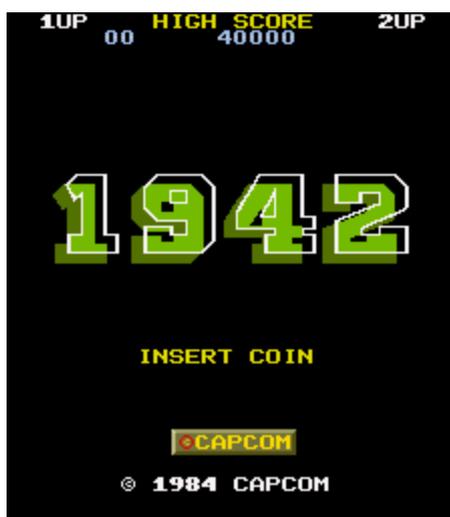


Ilustración 34: 1942 menú



Ilustración 35: 1942

4. Tópicos

Combate espacial en vertical.:

Combate de naves y aviones en un entorno que se desplaza verticalmente de arriba a abajo. Existirán tres niveles de movimiento:

- El inferior: Mostrará la zona terrestre. Aquí podremos ver construcciones, naturaleza, etc.... Habrán ciertos enemigos que podrán atacar desde esta posición(tanques, antiaéreos y algún jefe final).
- El intermedio: aquí se mostrarán zonas de transición dentro de la misma fase. El paso de una zona a otra podría mostrarse con una pasada de nubes por este nivel. Este nivel también tendrá uso para las animaciones que no tengan nada que ver con el combate (fuegos artificiales en un parque de atracciones, nubes, etc...)
- El superior : En este nivel se desarrollará toda la acción de combate. Los elementos principales son la aeronave del jugador, los enemigos, las balas y obstáculos.

Invasión y defensa del planeta tierra.

Uno de los fuertes de este juego será este tópico. El jugador va a tener la opción de jugar la historia desde el lado de los defensores o desde el lado de los invasores. Las zonas de conflicto son comunes, es decir, tanto para los defensores como para los invasores, tenemos las mismas zonas para conquistar. La diferencia entre jugar con unos u otros reside en el grupo de naves entre las que podremos elegir para jugar y en el punto de vista histórica de la guerra (la historia esta escrita solo por los vencedores).

Gran cantidad de naves y disparos contra el jugador.

Dado el genero del videojuego, es necesaria la intervención de una gran cantidad de enemigos y disparos. Tanta cantidad de objetos que hay que esquivar es lo que le da la gracia a este juego, con movimientos rápidos para esquivar y el ansia de destruir todo lo que venga a por ti. Digamos que en un principio podrían aparecer 200 enemigos y unas 500 balas por zona. Esta cantidad variará hasta conseguir un numero adecuado ya que lo que no queremos es que el jugador se agobie o se canse de morir.

Jefe al final de cada territorio.

No puede faltar en este tipo de clásicos un jefe final, un enemigo mucho más difícil que los demás de la zona y que debamos de vencerlo para finalizarla. Puesto que hay 9 zonas,

existirán 9 jefes más el reto final. Estos jefes tienen la característica principal de que tiene mucha vida y dispara muchas balas y algún disparo especial (único para cada jefe).

Se puede parar el desplazamiento de la capa inferior... para que el combate sea en estático o se puede mantener o aumentar la velocidad de este.

Selección de varios aviones para manejarlos.

Se podrá elegir entre 8 aeronaves, 4 por parte de la defensa y otros 4 por parte de los invasores. Se elegirá uno al principio de la partida y se jugará el desarrollo de la historia con el mismo siempre, con la posibilidad de cambiarlo en el momento en el que es derrotado.

Como es de esperar, cada aeronave tendrá unas características distintas entre los demás, como mas velocidad de desplazamiento, mayor rango de disparo o velocidad de disparo, disparo especial mas desbastador o incluso el uso de misiles teledirigidos.

Más adelante se especifican todas las aeronaves.

El armamento de los aviones se pueden mejorar.

Ejemplos claros de este tópico como los grandes éxitos de la saga de Aerofighters o el 1942. La potencia del disparo del jugador variará según vaya recogiendo objetos que aparecen al destruir ciertas naves. Estas naves pueden tener una aparición de 10 por zona. La manera de mejorar el disparo será por tamaño de la ráfaga de disparos y algún extra como misiles, láser o un escudo. Este armamento adicional sera fijo hasta que el jugador sea vencido o pase de zona al vencer al jefe.

Música rápida (d'n'b, heavy, etc..)

Opino que una de las mejores maneras de llevar el ritmo de este tipo de juegos es con la música. Se cambiará de música al cambiar de situación en una zona, por ejemplo en el paso a la lucha contra un jefe final de zona.

5. Historia.

En la tierra un único Imperio se alza, total y poderoso. Se divide en cuatro potencias:

- Los Selenitas, estos viven en la luna y son los sacerdotes, son los que están en comunicación con su dios. Desde la Luna, dirigen la evolución de la sociedad humana.
- Gran Imperio, cuya capital esté en Canadá. Se formó de una anexión de todos los estados de la tierra para acabar con las guerras civiles. Cuando la situación política en la tierra no

se sostenía, el Dios de los terráqueos se comunicó con los Selenitas, estos hicieron un comunicado a los terráqueos : “detened las guerras y unificaros o pereceréis en el nombre del Único”. En la tierra se rieron de ellos. Poco después, en la luna, una gran maquinaria tan antigua como la Humanidad cobró vida. Es uno de los grandes cañones de las antiguas guerras. Y de un disparo contra las capitales en guerra quisieron dejar un mensaje claro: “uníos o pereceréis”. Se forma el Gran Imperio de la Tierra.

- Solo dos potencias se escaparon a la globalización, Los Barones, que no serán más de un par de cientos, existen como república independiente. El Gran Imperio y los selenitas les permiten subsistir por que son los únicos capaces de extraer los materiales, que extraen del subsuelo oceánico cerca de las placas antárticas, necesarios para desarrollar la energía anti-gravitacional.
- Y los Technólogos, son una empresa asiática, que sobrevive políticamente independiente gracias a que posee la mitad de la riqueza de la tierra. Existen antes de la formación del Gran Imperio.

Y en los confines del cosmos una civilización evoluciona en un estado militar con un solo objetivo: recuperar su hogar. Se dividen en cuatro castas:

- La casta de los Technomagos: Dada las condiciones de vida en ese mundo, se creó una creencia hacia la fusión entre la máquina y la mente. Sus naves no tienen cañones, no disparan, básicamente son amplificadores de su poder mental, ellos atacan al piloto de la nave.
- La casta de los Technólogos : son todo lo contrario de los technomagos, por que cambian su cuerpo por la máquina. Es muy costoso y complicado formar un solo technólogo, pues necesitas un cerebro capaz de controlar una veintena de naves a la vez.
- La casta de los Guerreros : Son la pura raza, no tienen ninguna creencia más que en la guerra. Desde que nacen son entrenados en las peores condiciones, creándoles un odio hacia los terráqueos. Son la unidad militar básica de los exiliados.
- La casta de los Rechazados: Evolucionan a parte de las otras castas. Las deformaciones de nacimiento eran castigadas con la expulsión de las castas y el abandono en el espacio. Al primer rechazado le recogió su propio dios, encomendándole la misión de formar su propia casta con tal de crear el mayor ejército posible, todo hombre puede hacer la guerra. Además le entrega el secreto de la inmortalidad.

6. Plataformas.

El juego está desarrollado y pensado para jugarlo en un ordenador personal con cualquiera de los siguientes sistemas operativos:

- Windows.
- Linux.
- OS2.

7. Mecánica y física del juego.

El juego se desarrolla en 2 niveles distintos según la altura. El primer nivel será el suelo, tierra firme con sus edificios y varias construcciones. Aquí pueden aparecer algunos enemigos como los lanzamisiles de la primera fase. El segundo nivel será donde esté toda la acción del juego, aquí se sitúa la nave del jugador y los enemigos a batir. Habrá partes del nivel primero que invadan el segundo nivel, al estilo de edificios muy altos con los que te puedas chocar.

8. Personajes.

8.1. Principales.

Technologos (Terraqueos)

- Nombre : Ryu Kayatsuri
- Localización: Asia
- Origen: Industrias pesadas
- Físico: Delgado, de 1,70 m de altura, rasgos asiáticos, de 70 kg.
- Carácter: Joven, serio y inexpresivo, inmutable, y programado para lo que solo sabe hacer, pilotar.
- Nave y habilidades: Nave de armamento pesado, extra de ataque. Menos velocidad. Bomba destructiva de toda la pantalla.
- Historia: Es un humano creado artificialmente (clon sintético), especialmente dotado para el manejo de las naves de tecnologías pesadas, una nave que un humano normal no podría pilotarla.

Selenitas

- Nombre : Teothorus “El Santo”
- Localización: La Luna
- Origen: Templo de la Luna
- Físico: Un anciano, de unos 70 años, canoso y de pelo descuidado.
- Carácter: Afable, buena persona, es el buen pastor.
- Nave y habilidades: Bendición de batalla, Bomba : Cañonazo Lunar. Velocidad media.
- Historia : Dios se le aparece cuando es joven y le dice : “Teothorus, la gran guerra llegara en tu tiempo y tendrás un papel muy importante para ella, prepárate.” Entonces aprende a pilotar, aprende a manejar los cañones, etc... Al cabo del tiempo se vuelve el mejor sin dejar de ser puro de corazón..

Barones

- Nombre : Lord Agnar
- Localización: Antártida
- Origen: La casa de los Barones
- Físico: Es de gran estatura y corpulencia, pero de porte noble. De unos 1,90cm de altura, 100 kg de peso. Con aspecto élfico, como describen las antiguas novelas de la tierra.
- Carácter: Orgullosos.
- Nave y habilidades: Equilibrado en sus estadísticas. Extra de energía. Velocidad media. Bomba: se puede rellenar la vida 50%.
- Historia: Son una familia que se dedica a extraer y sintetizar la materia antigravitacional. No son guerreros, pero por herencia familiar, en cada generación se crea un guerrero, a la espera de que la guerra llegue. Este tiene una única misión, salvaguardar a toda la familia, por que si se deja de extraer material antigravitacional, se perderá la guerra. Es un piloto entrenado para defender a toda la humanidad.

Imperiales

- Nombre : Robert W. Cooper
- Localización: Canadá
- Origen: El Gran Imperio. Extra de daño a los monstruos de final de fase.
- Físico: Es un tipo de aspecto común, 30 años, 1.80 cm de altura, 73 Kg de peso.
- Carácter: Asustado, pero a la vez, perspicaz y observador. Un superviviente

Nave y habilidades: Bombas : Extra de daño a los monstruos de final de fase.
Velocidad de movimiento rápido. Tiro rápido.

- Historia: Tiene un trabajo típico, con sus vacaciones y su sueldo. Pero un fatídico día, le llaman a las filas para embarcarse en una batalla que no entendía y no compartía. Pero se adapta y comprende que si no pelea, mueres. Así son las invasiones.

Guerreros

- Nombre : Gor C.
- Localización: - , Nómadas del espacio.
- Origen: Casta de los guerreros
- Físico: Enorme y brutal. De unos 120 Kg, y mas de 2M de altura,
- Carácter: Nacidos para la guerra, no conocen otro estado que sea el de furia ante el fragor de la batalla.
- Nave y habilidades: Por el tamaño de los pilotos, suelen ser auténticos tanques espaciales, de potencia militar excesiva. Movimiento lento y gran daño extra. Bombas de doble uso, pequeño radio.
- Historia : Héroe de guerra, ha sobrevivido a miles de batallas, y siempre con el mayor numero de bajas. Nunca ha sido derribado, conquistador. Si le preguntas por su vida, no sabrá ni contestarte, ya que solo ha vivido y vive para la batalla.

Technologos (Exiliados)

- Nombre : Nº 41 (Padre de los Technologos)

- Localización: Nave Mundo
- Origen: Casta de los Technologos
- Físico: De sus cuerpos solo queda el cerebro, conservado y conectado a varias naves a la vez.
- Carácter: De seres humanos no les queda nada, son grandes organismos compuestos por naves espaciales.
- Nave y habilidades: Las legendarias naves de batalla del Padre de los Technologos, son dos, mas débiles, pero son dos. Bomba : 30 seg de un aliado.
- Historia: Es un científico del campo de los biotransplantes. Él es el creador de los 40 únicos technologos, finalmente se transforma él mismo en la maquina definitiva, la Nº 41.

Technomago

- Nombre : Mente Maestra.
- Localización: Asteroide colmena
- Origen: Casta de los Technomagos
- Físico: Consumido de no usar el cuerpo, su mente esta conectada a la nave. 45Kg, 1, 70 cm de altura. Apenas lleva ropa.
- Carácter: Sabio entre sabios. Están en comunión con el universo, es como un chaman espacial.
- Nave y habilidades: No tiene cañones, amplifica el poder mental para atacar al piloto adversario mediante ataques psíquicos. Doblegar voluntad y choque psíquico y. Bomba psíquica (vaciar la pantalla).
- Historia: Es el gran sabio que a través de un viaje astral, entro en armonía con el universo, mientras observaba la magnitud de La Creación. Iba enseñando ese conocimiento sobre lo que vio y sintió. Así nació la escuela de los technomagos de la que el, según sus palabras, es solo un pastor.

Rechazados

- Nombre : Un ojo el Inmortal

- Localización: Planeta ZX1000
- Origen: Los Rechazados
- Físico: Deforme, achaparrados por la gran gravedad del planeta en el que viven.
- Carácter: Como inmortal, no tiene miedo a la muerte, es un luchador eterno, la peor pesadilla en las guerras.
- Nave y habilidades: Equilibrada en sus estadísticas, tiene buena defensa y buen ataque, de velocidad media. Mala bomba. Es una nave antigua, hecha de pedazos que recogen de los desechos espaciales. Bomba : campo de defensa.
- Historia : Es el primer rechazado de los exiliados, acogido bajo el seno de su dios, que le ayudo a recoger y criar a todos los rechazados de las otras castas. Para ello, le dio el don de no morir jamás por causas naturales.

8.2. Enemigos.

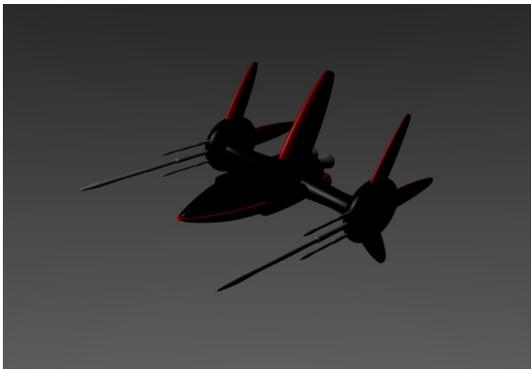
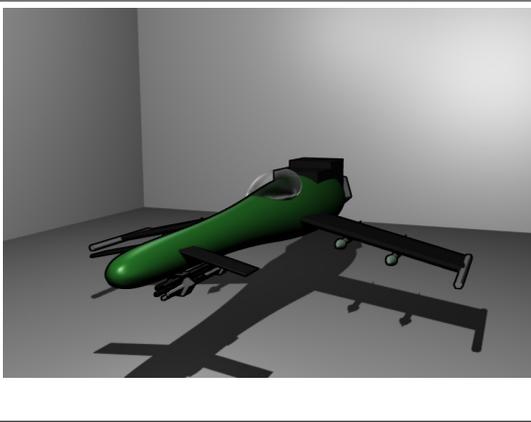
Según el bando que elijas, tus enemigos serán exactamente los personajes del bando opuesto.

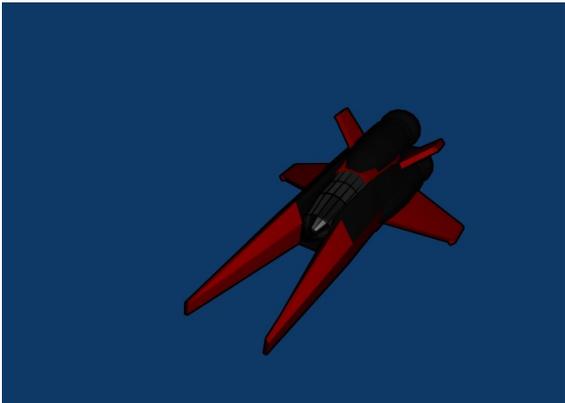
9. Escenarios.

- El desierto : Rocas gigantes, montañas de arena y áreas extensas de barro seco es lo que te espera en esta pantalla. Puede que algún oasis te salve.
- Autopista : La mejor manera de llevar a tu objetivo es seguir este gusano de cemento kilométrico. Cuidado con los puentes y las torres de alta tensión.
- Ciudad : Al final de la autopista llegas a tu primer obstáculo, una gran ciudad llena de rascacielos y helicópteros despistados. Al final nos encontraremos con la base militar.
- La Luna : Combate en zona sagrada, Cañones de plasma que defienden la zona y pilotos experimentados te harán la vida imposible.
- Espacio interestelar : La ultima frontera que te separa de tu enemigo final, y con el el final de esta guerra. Campos gravitatorios, lluvia de meteoritos, esto va a ser peor que el infierno.

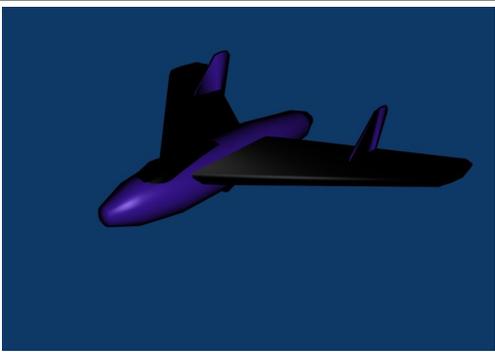
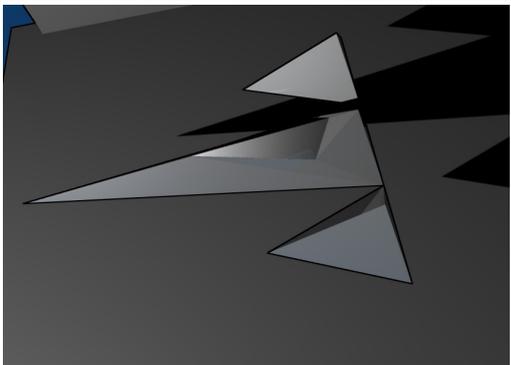
10. Objetos 3D.

10.1. Personajes (naves).

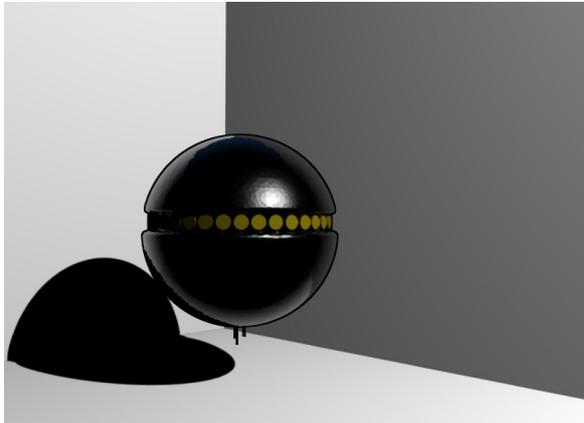
	<p>Nombre dibujo : Nave_1.jpg</p> <p>Nombre del modelo : Nave_1.blend</p> <p>Descripción : Modelo perteneciente al personaje Ryu Kayatsuri.</p>
	<p>Nombre dibujo : Nave_2.jpg</p> <p>Nombre del modelo : Nave_2.blend</p> <p>Descripción : Modelo perteneciente al personaje Teothorus "El Santo".</p>
	<p>Nombre dibujo : Nave_3.jpg</p> <p>Nombre del modelo : Nave_3.blend</p> <p>Descripción : Modelo perteneciente al personaje Robert W. Cooper .</p>
	<p>Nombre dibujo : Nave_4.jpg</p> <p>Nombre del modelo : Nave_4.blend</p> <p>Descripción : Modelo perteneciente al personaje Lord Agnar.</p>

	Nombre del dibujo : Nave_5.jpg
	Nombre del modelo : Nave_5.blend
	Descripción : Modelo perteneciente al personaje Gor C.

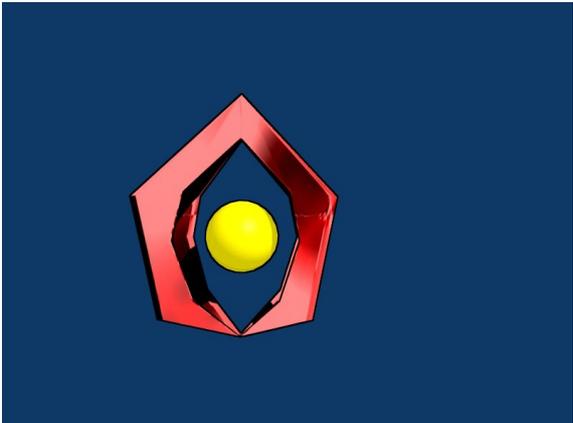
10.2. Enemigos.

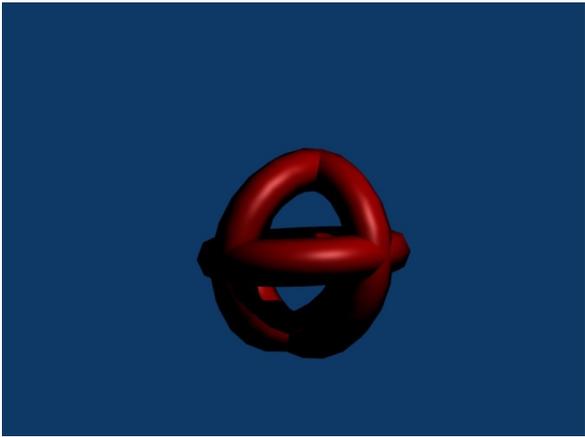
	Nombre dibujo : Enemigo_1.jpg
	Nombre del modelo : Enemigo_1.blend
	Descripción : Enemigo común del primer nivel
	Nombre dibujo : Enemigo_2.jpg
	Nombre del modelo : Enemigo_2.blend
	Descripción : Enemigo común del primer nivel
	Nombre dibujo : Enemigo_3.jpg
	Nombre del modelo : Enemigo_3.blend
	Descripción : Enemigo común del segundo nivel.

10.3. Jefes.

	<p>Nombre dibujo : Final_boss_1.jpg</p>
	<p>Nombre del modelo : Final_boss_1.blend</p>
	<p>Descripción : Enemigo final del primer nivel.</p>
	<p>Nombre dibujo : Final_boss_2.jpg</p>
	<p>Nombre del modelo : Final_boss_2.blend</p>
	<p>Descripción : Enemigo final del segundo nivel.</p>

10.4. Objetos

	<p>Nombre dibujo : objeto_1.jpg</p>
	<p>Nombre del modelo : objeto_1.blend</p>
	<p>Descripción : mejora de arma.</p>

	Nombre dibujo : objeto_1.jpg
	Nombre del modelo : objeto_1.blend
	Descripción : Recuperas una vida.

10.5. Escenografía.

	Nombre dibujo : palmera.jpg
	Nombre del modelo : palmera.blend
	Descripción : Palmera.

	Nombre dibujo : arbusto.jpg
	Nombre del modelo : arbusto.blend
	Descripción : Arbusto

	Nombre dibujo : lanzacohetes.jpg
	Nombre del modelo : lanzacohetes.blend



10.6. Pantallas.

- Logo



Ilustración 36: Far Exile - Logo (Elephant3)

- Título



Ilustración 37: Far Exile - Título

– **Primera fase**



Ilustración 38: Far Exile - 1ª Fase

– **Game over**



Ilustración 39: Far Exile - Game Over

– **Créditos**



Ilustración 40: Far Exile - Créditos

– **Controles**



Ilustración 41: Far Exile - Controles

11. GUI.



12. Mapa de Botones.

Acción	Control	Contexto
Izquierda	←	Todo
Derecha	→	Todo
Arriba	↑	Todo
Abajo	↓	Todo
Disparar	Espacio	Juego
Bomba	Alt	Juego
Iniciar	Enter	Todo
Salir	Esc	Todo

13. Sonidos y Muscia.

13.1. Acciones.

- Disparar : disparos.wav
- Colisionar : golpe_de_chapa.wav
- Coger objeto mejora arma : arma1.wav

13.2. Eventos.

- Jefe aproximandose : peligro.wav
- Combate con el jefe : jefe.wav
- Inicio pantalla : inicio_fase.wav
- Destruir Jefe : objetivo_cumplido.wav
- Destruir piedra : romper_piedra.wav
- Destruir malo : explosion_pkeña.wav
- Destruir misil : explosion_pekeña.wav

13.3. Pantallas.

- Logo : logo1.wav

- Título : efecto_entrada.wav y inicio.wav
- Primera Fase : 1_fase.wav
- Game over : game_over.wav
- Créditos : credits.wav

14. ArtBook.

14.1. Personajes.

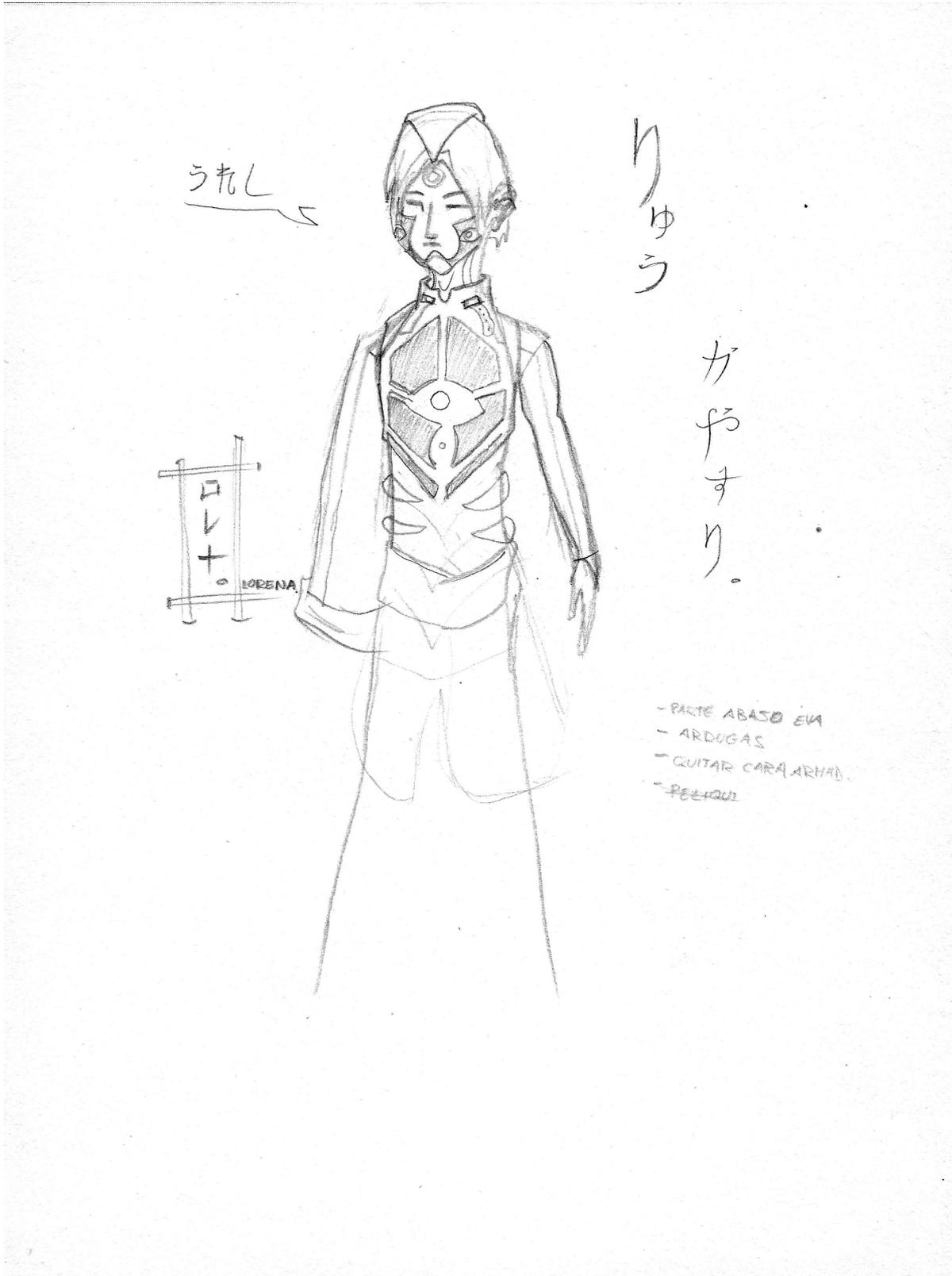


Ilustración 42: Ryu Kayatsuri



Ilustración 43: Lord Agnar



Ilustración 44: Gor C.



Ilustración 45: Robert W. Cooper



Ilustración 46: Teothorus "El Santo"

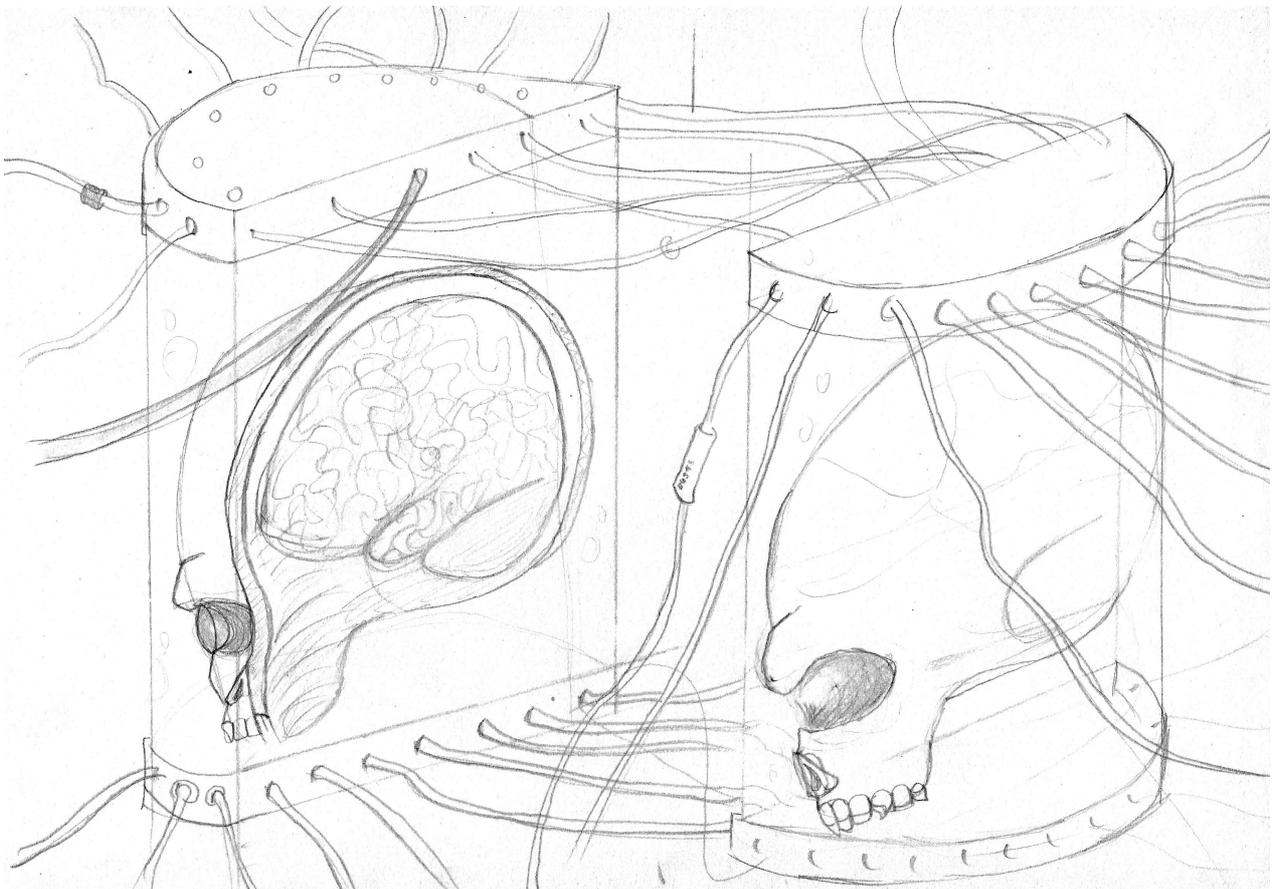


Ilustración 47: N° 41 (Padre de los Technologos)



Ilustración 48: Mente Maestra.



Ilustración 49: Robert W. Cooper



Ilustración 50: Unojo el Inmortal



Ilustración 51: Unojo el Inmortal

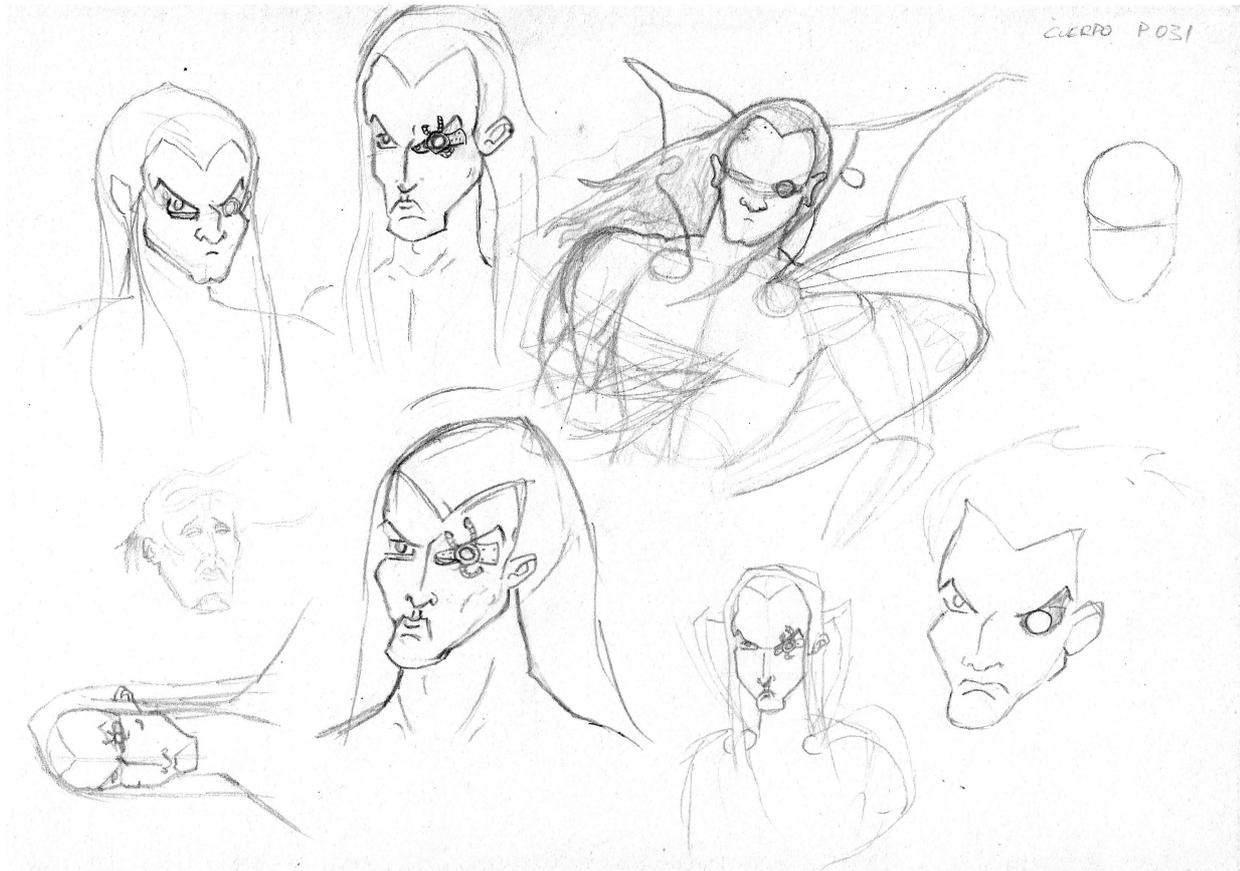


Ilustración 52: Lord Agnar

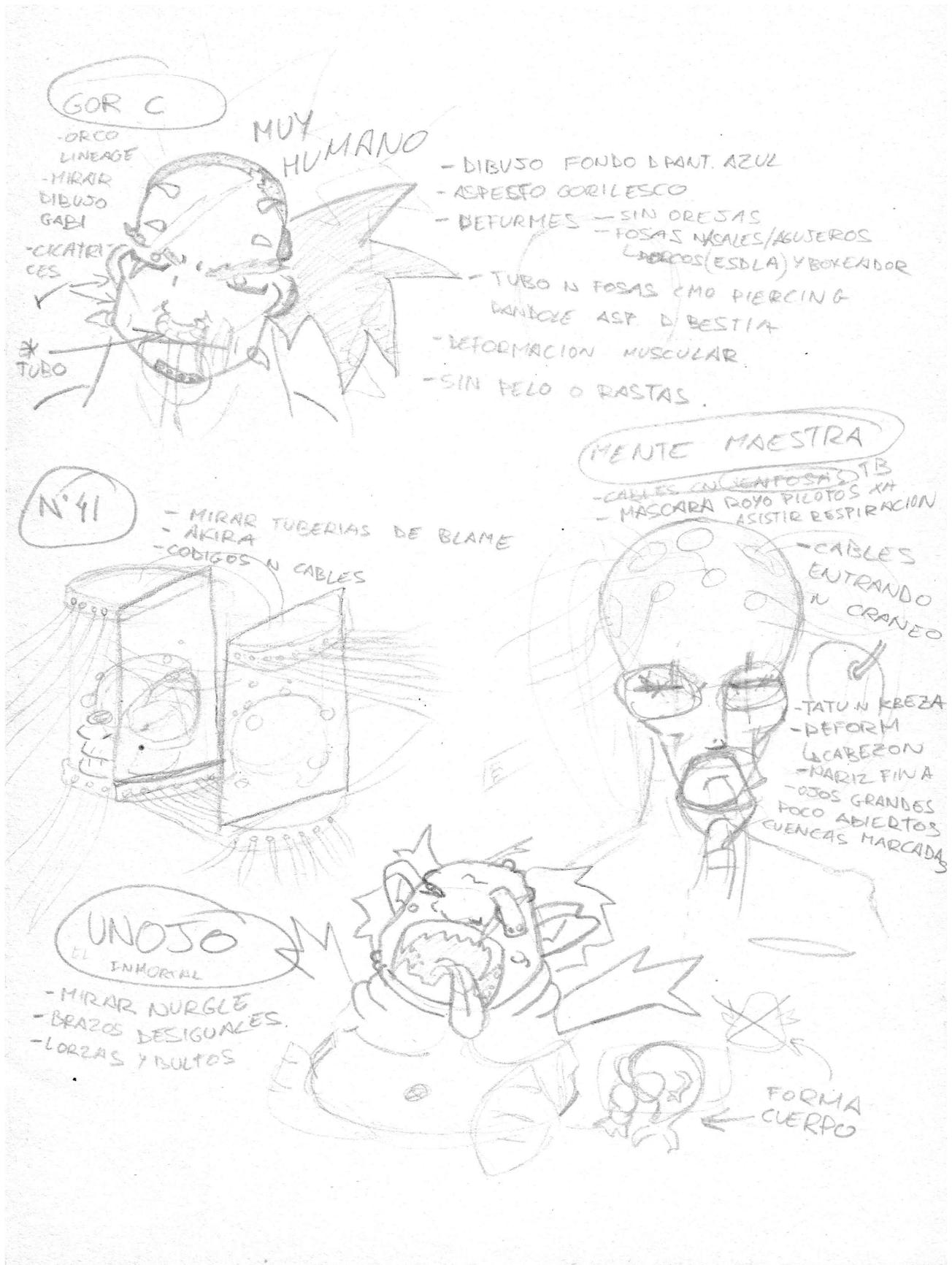


Ilustración 53: Bocetos 1

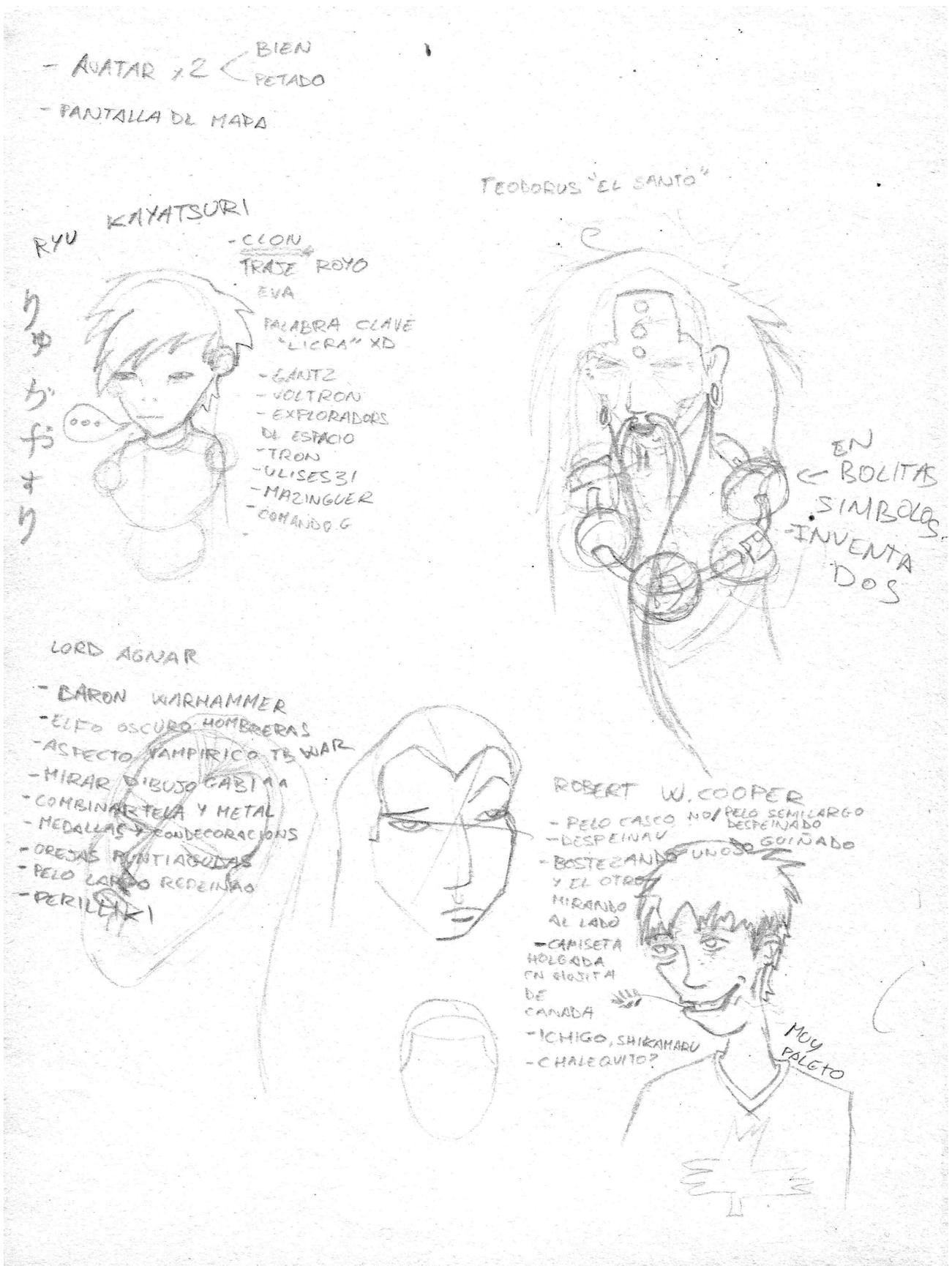


Ilustración 54: Bocetos2

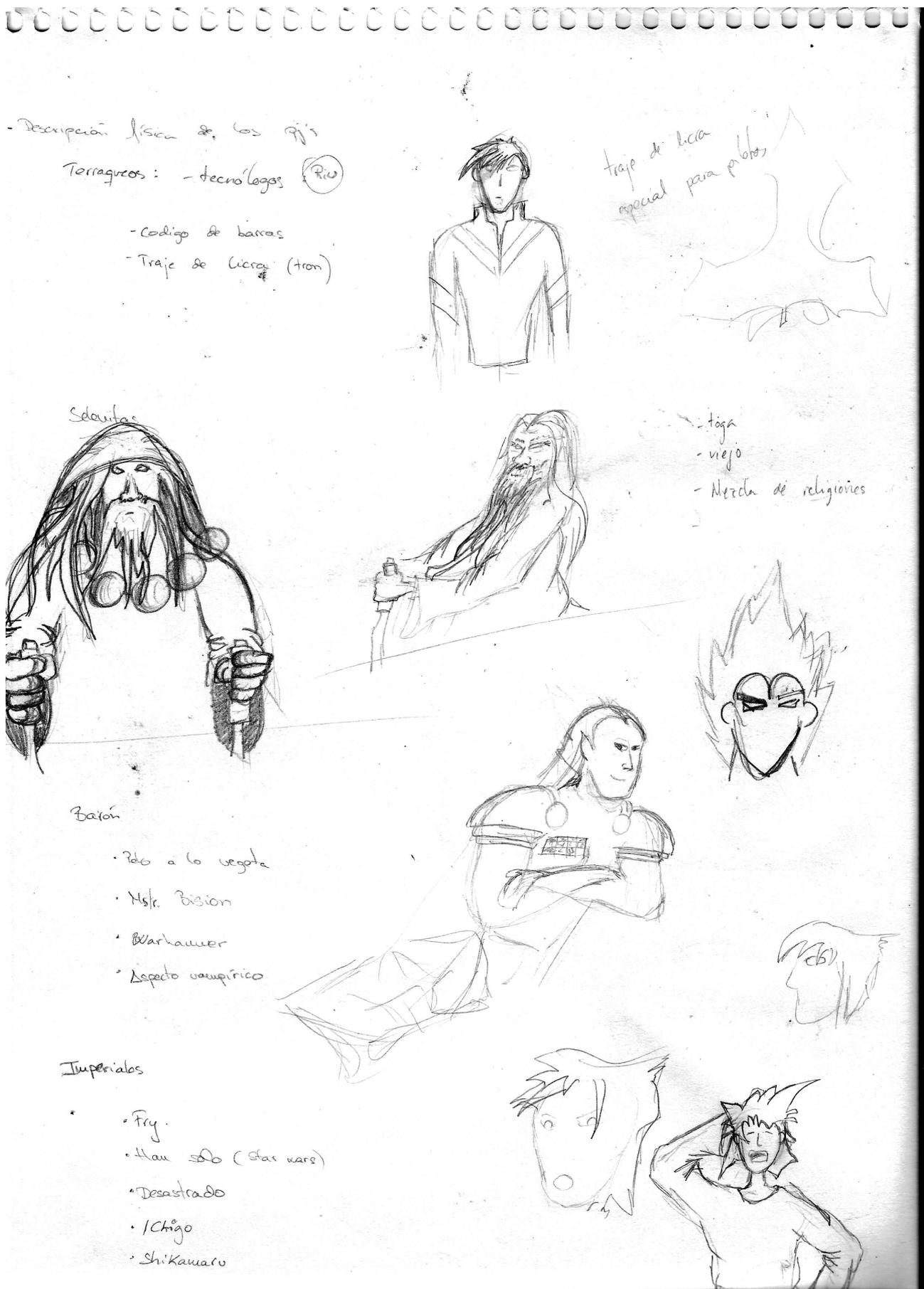


Ilustración 55: Bocetos3

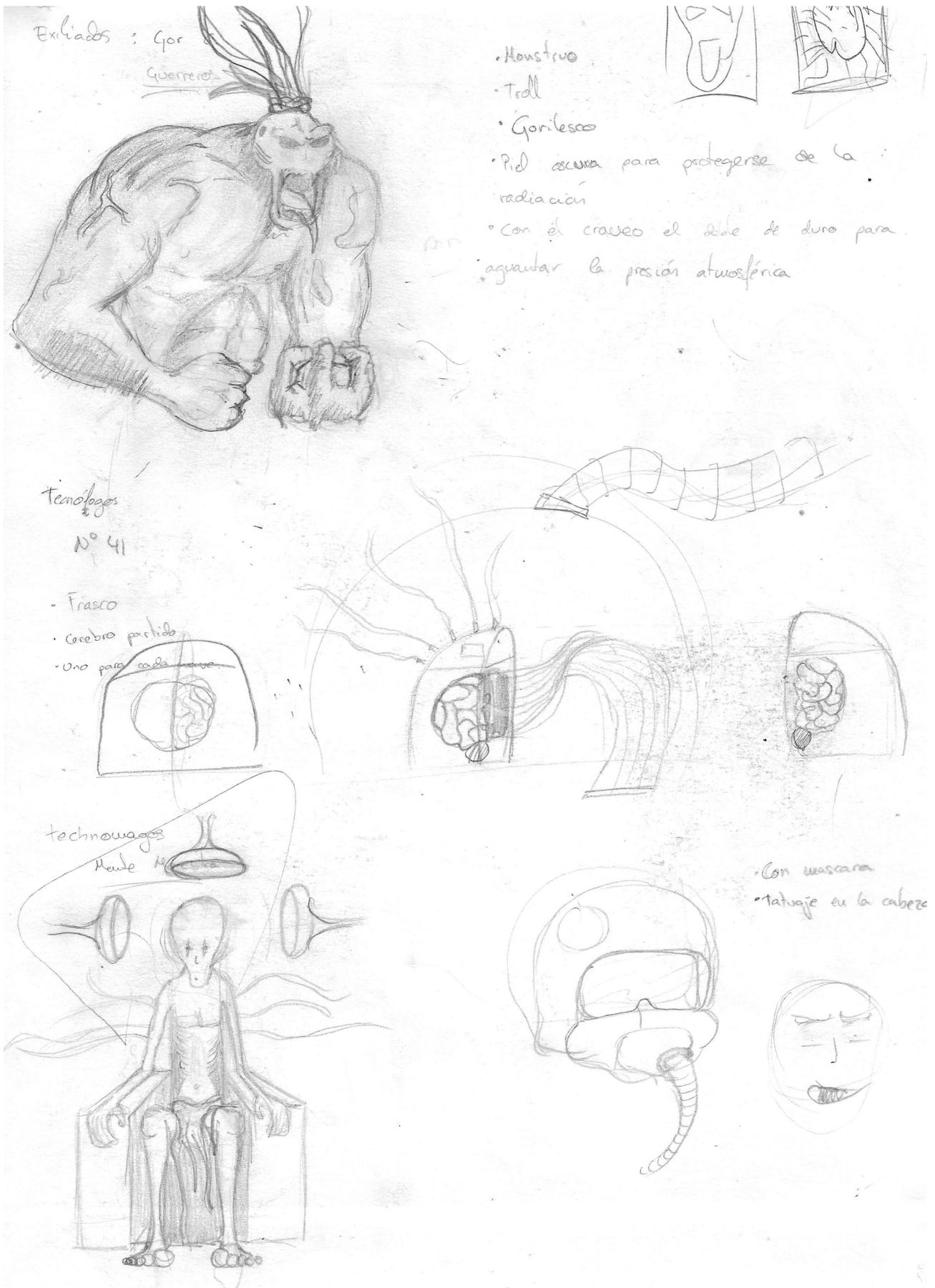


Ilustración 56: Bocetos4



Ilustración 57: Bocetos5

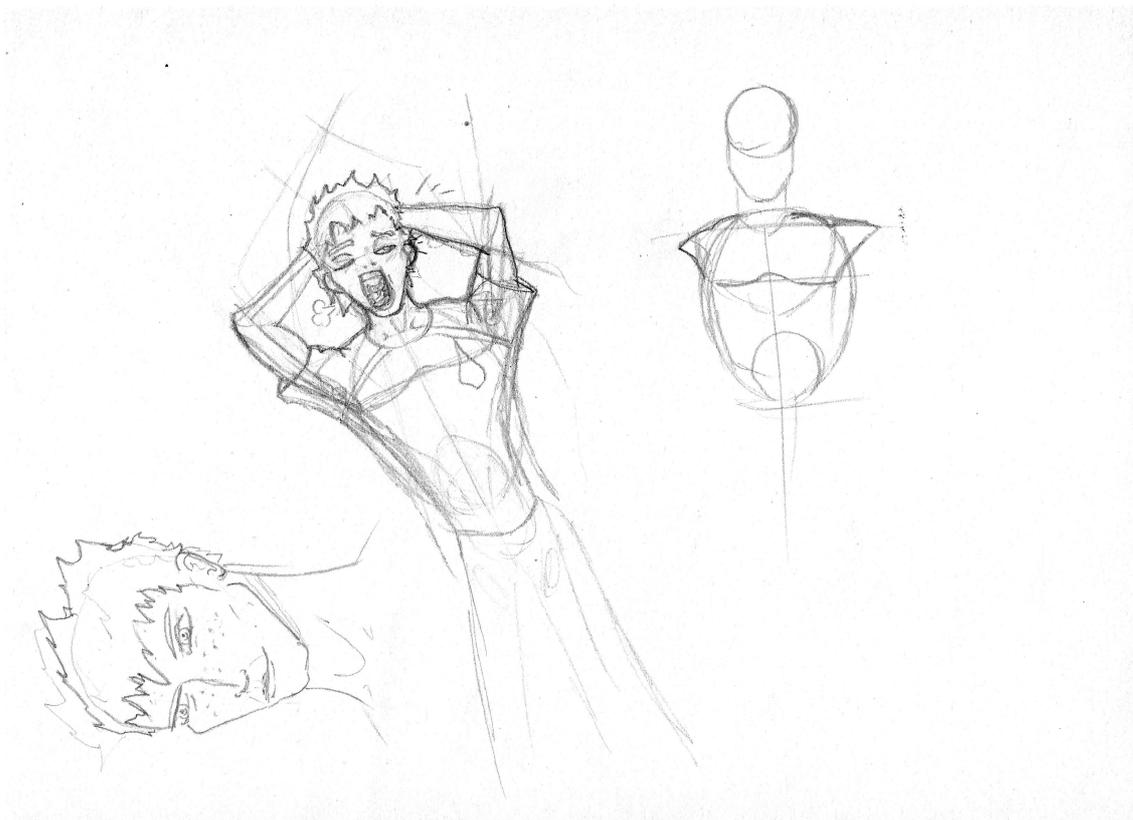


Ilustración 58: Bocetos6

14.2. Escenario.

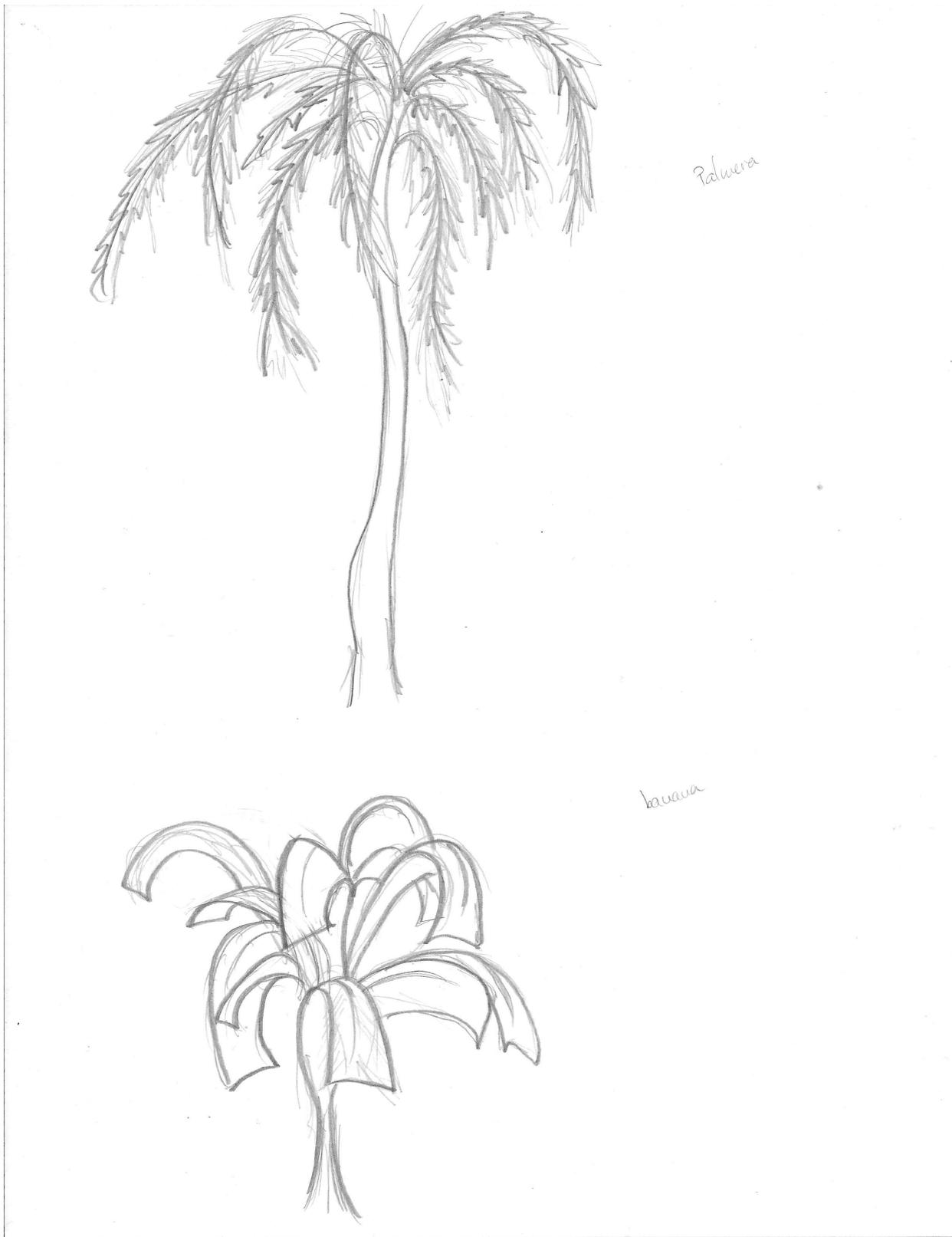


Ilustración 59: Palmera y arbusto

14.3. Pantallas.

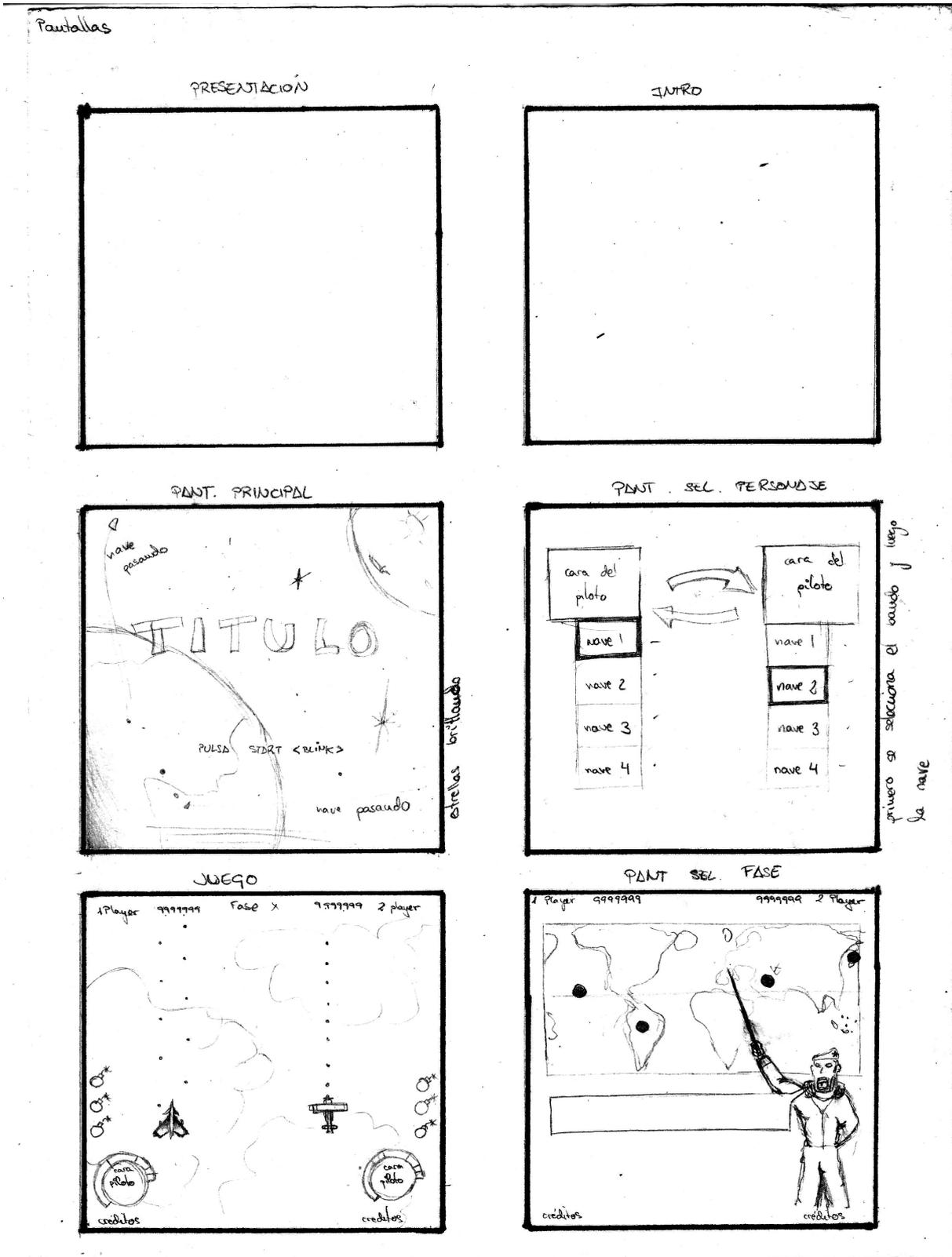


Ilustración 61: Pantallas1

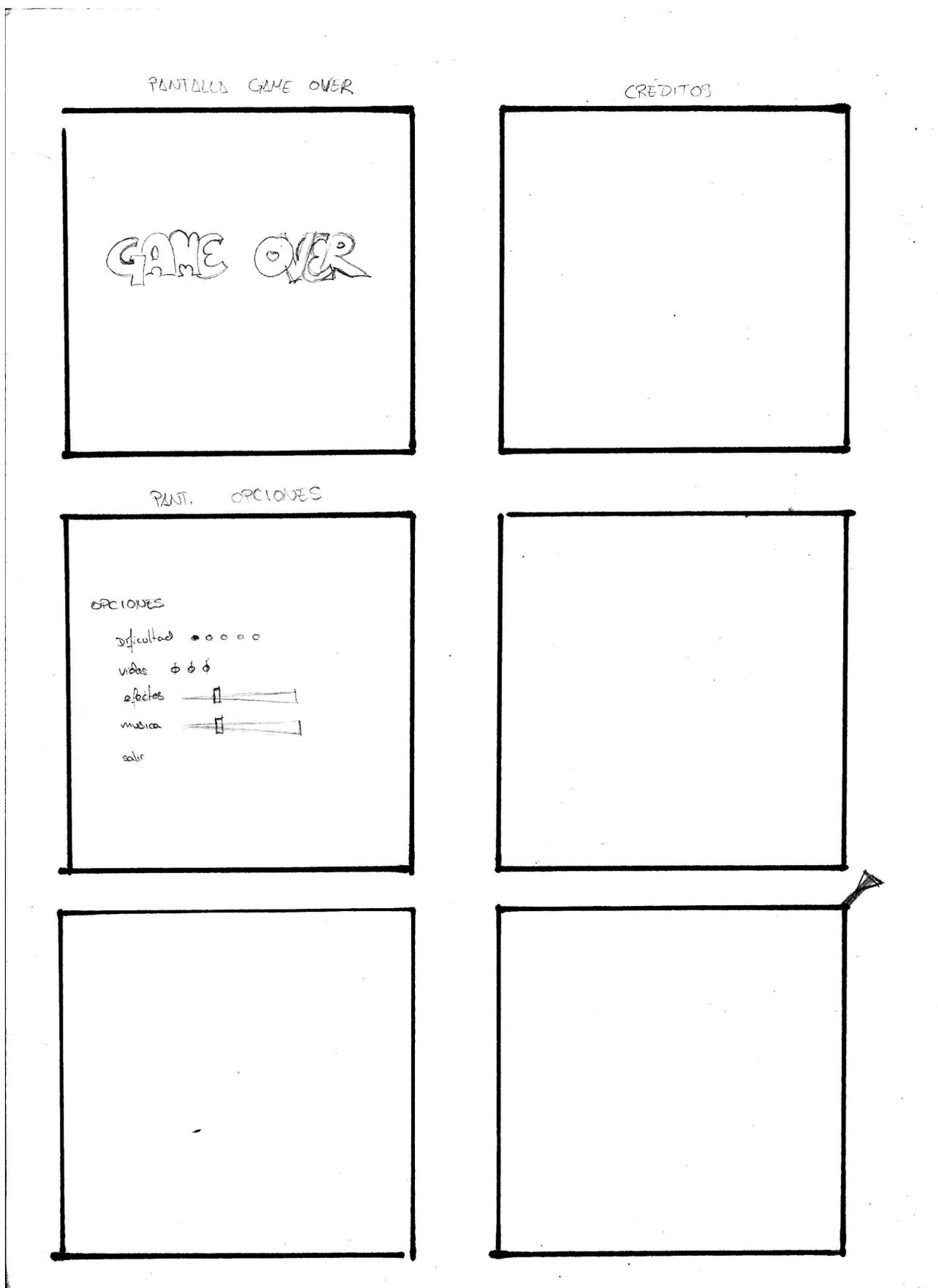


Ilustración 62: Pantallas2

14.4. Naves y objetos.

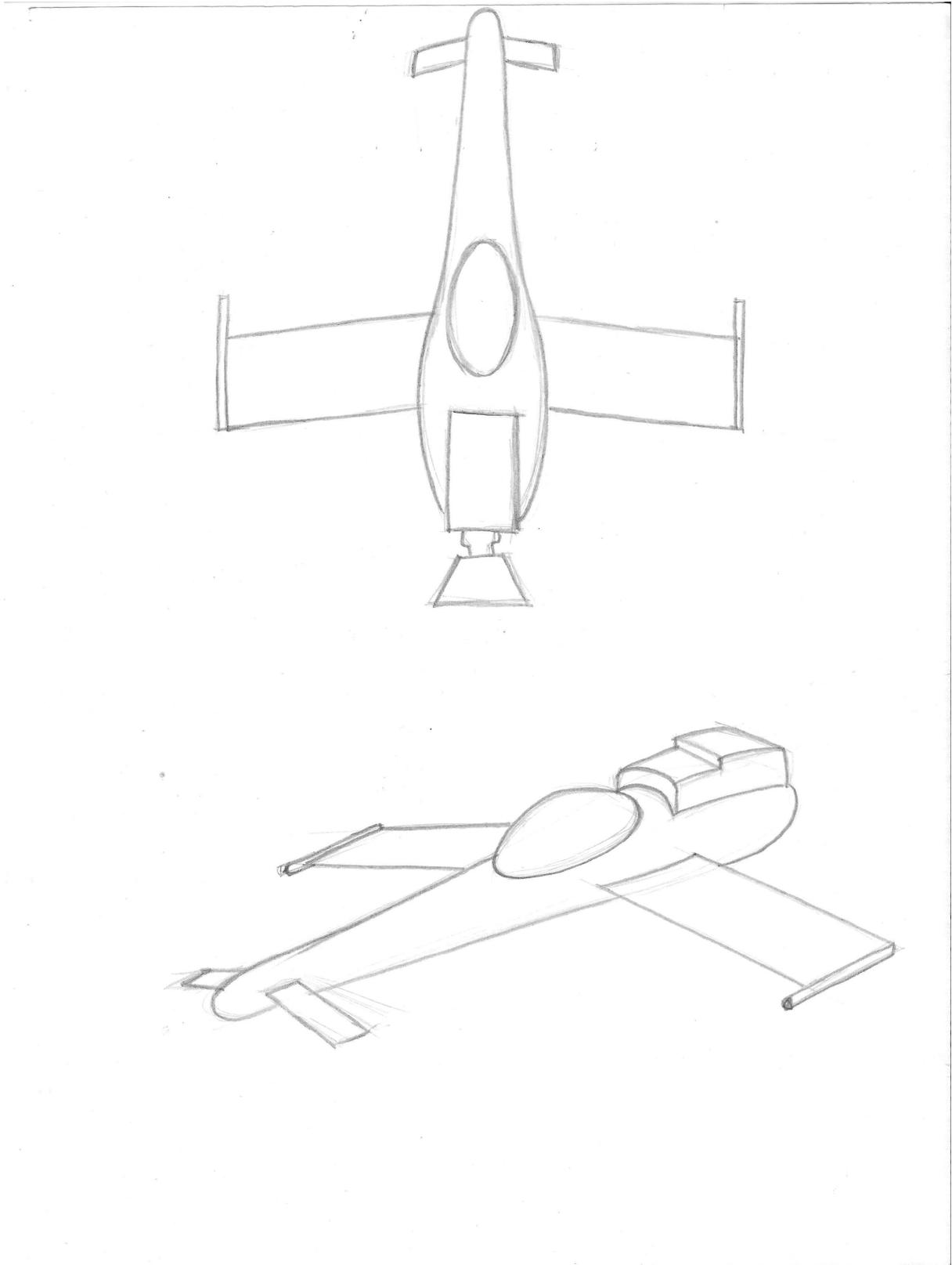


Ilustración 63: Nave de Teothorus “El Santo”

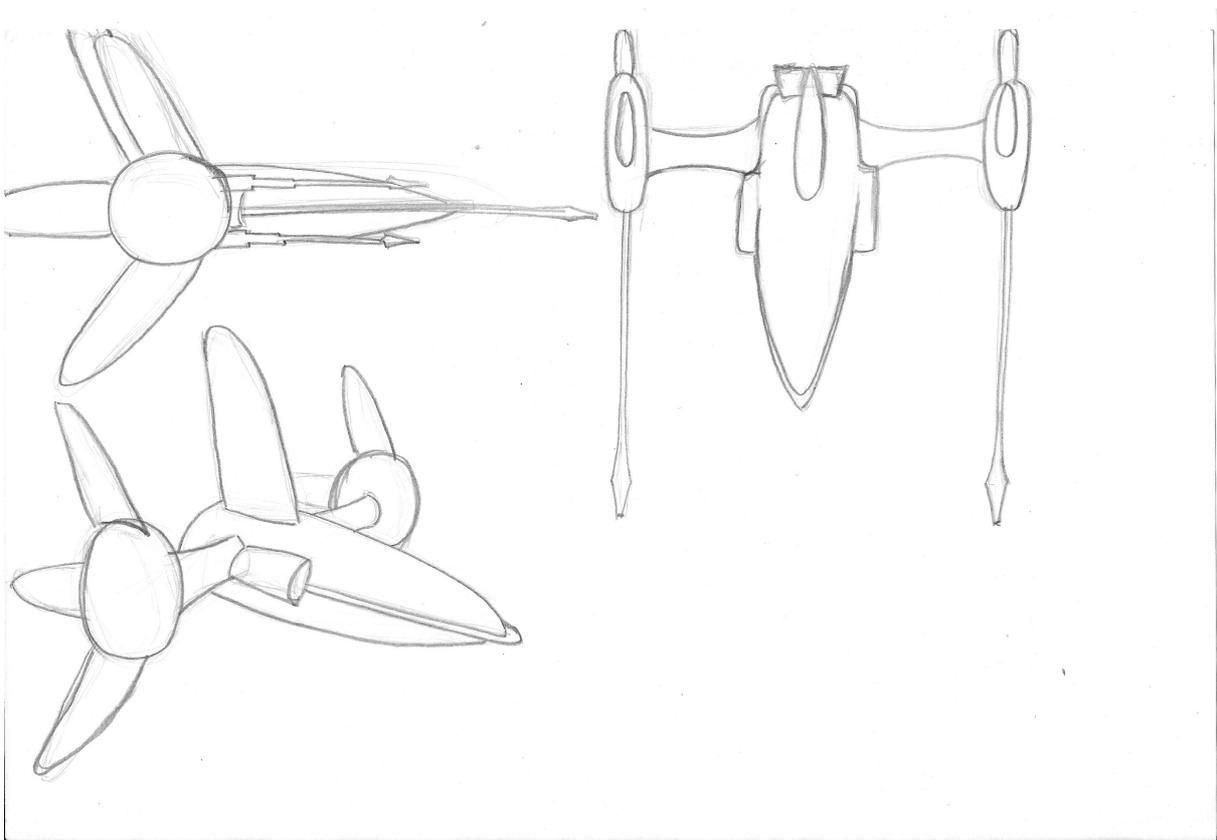


Ilustración 64: Nave de Ryu Kayatsuri

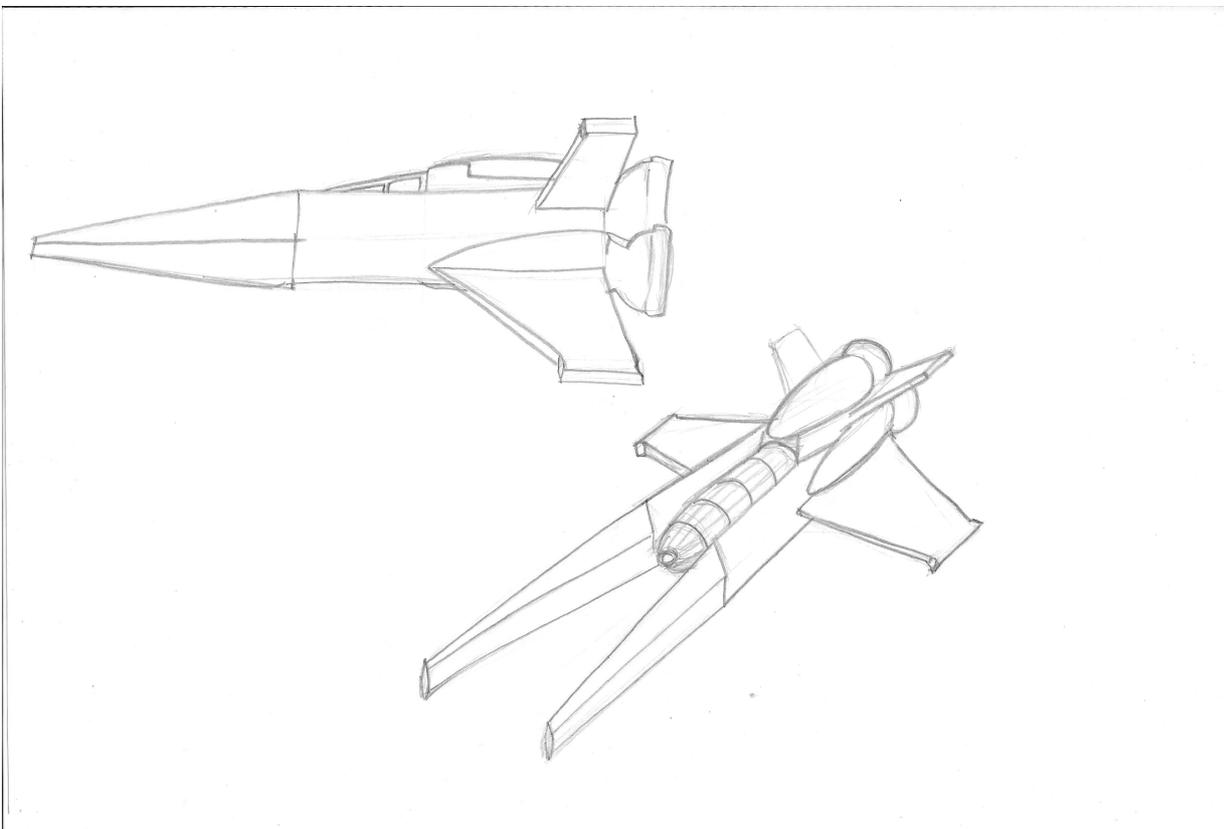


Ilustración 65: Nave de Gor C.

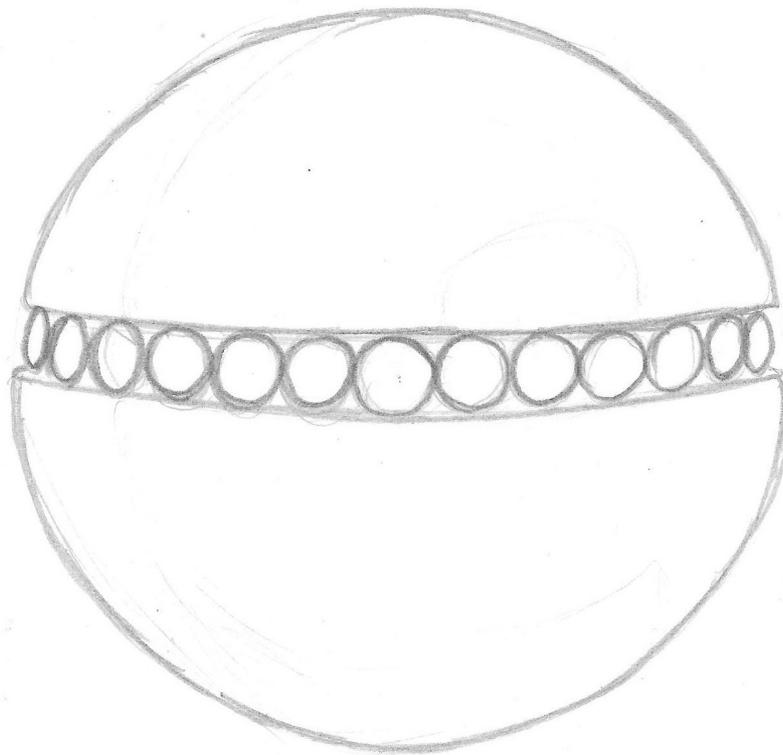


Ilustración 66: Enemigo final 2

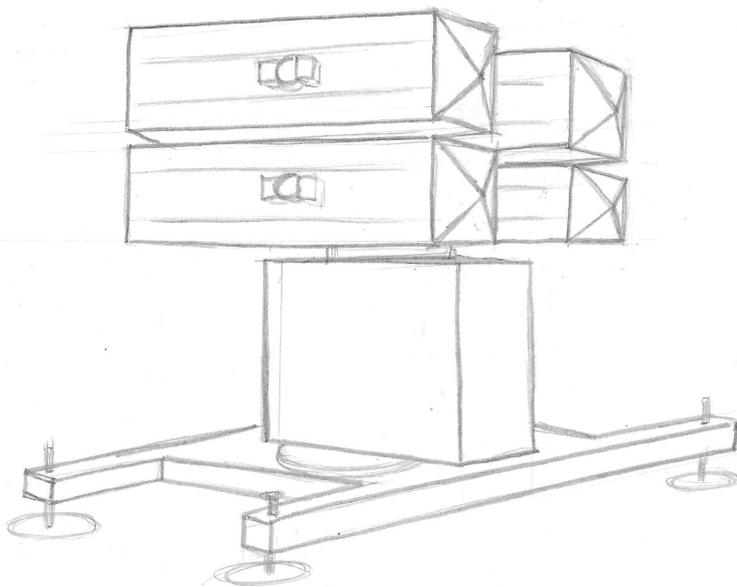


Ilustración 67: Lanazamisiles

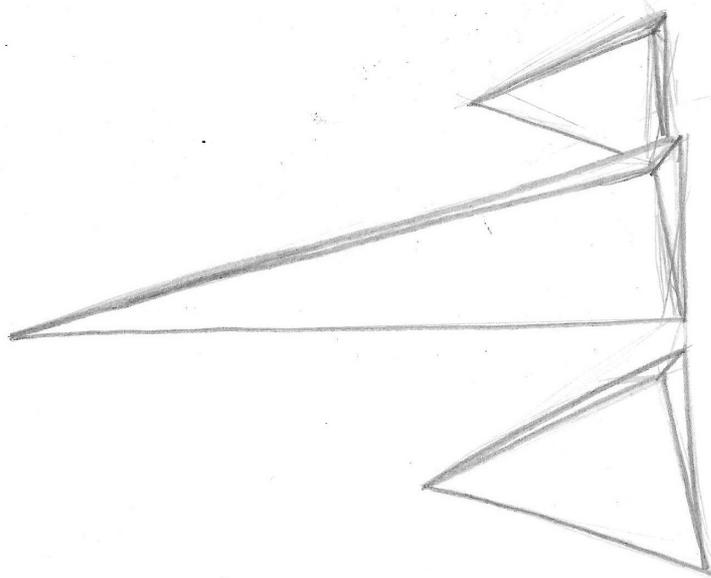


Ilustración 68: Enemigo3

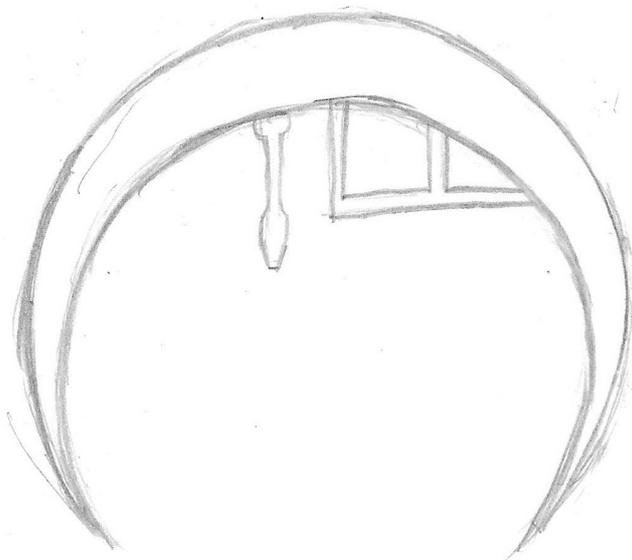


Ilustración 69: Enemigo2

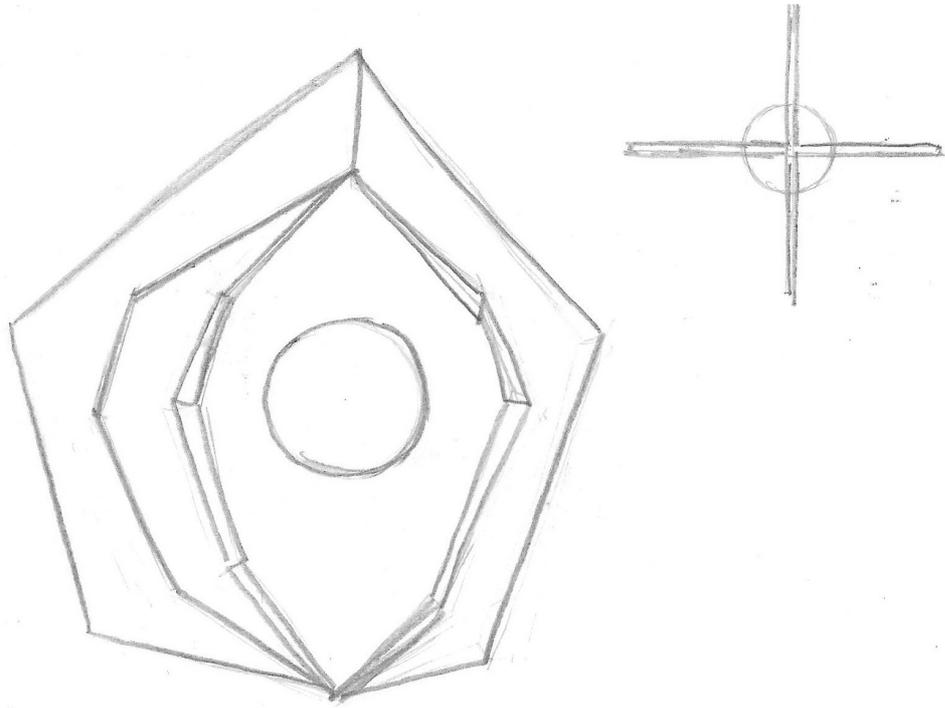


Ilustración 70: Objeto1

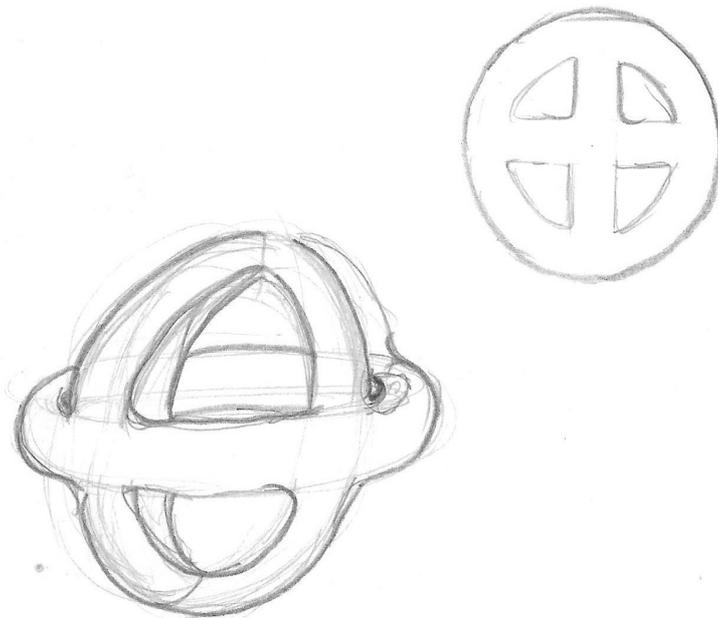


Ilustración 71: Objeto2

B. Tutorial de crear un ejecutable con blender.

Ademas de como crear un ejecutable, con esta página se puede publicar nuestro juego en una web.

<http://www.blendenzo.com/faqMakeExe.html>

C.Instalación del juego.

Descargar el juego de la siguiente página :

<http://farexile.jimdo.com/downloads/>

Descomprimir el archivo con el programa winrar que se facilita en la misma página y ejecutar el archivo "Far Exile (0.9.6).exe".

D. Tutorial para la herramienta de lógica del blender.

Basado en el juego de Yo Frankie de Spricot, esta página explica como utilizar los bloques lógicos para programar una escena de un videojuego.

También se incluye en la documentación del proyecto, el código completo del juego ofrecido por los autores del mismo.

http://wiki.blender.org/index.php/Org:Institute/Open_projects/Apricot/BGELogic

E. Python y Jython.

A través de estas páginas se puede aprender a utilizar el Python para crear scripts en blender :

- <http://wiki.blender.org/index.php/Doc:ES/2.4/Manual/Extensions/Python>
- http://www.fisicomolon.com/joomla/index.php?option=com_content&view=category&id=39:python-blender-24x&Itemid=58&layout=default

API : <http://www.blender.org/documentation/248PythonDoc/GE/index.html>

Además, se incluye en la documentación del proyecto, una copia de una pagina html con un tutorial bastante basico y sencillo "Pythonlopes".

F. Índice de Imágenes

Ilustración 1: Logo Gimp 2.0.....	10
Ilustración 2: Logo Blender 2.46.....	11
Ilustración 3:Logo OpenOffice.org.....	13
Ilustración 4: Logo Autacity.....	15
Ilustración 5: Versión del Pong para la venta en los comercios.....	16
Ilustración 6: Cartel promocional del Assassins Creed.....	17
Ilustración 7: Gráfica coste temporal.....	20
Ilustración 8:Logo Torque Game Engine.....	22
Ilustración 9: Logo Crystal Space.....	23
Ilustración 10:Logo Ogre3D.....	24
Ilustración 11:Logo OpenSceneGraph.....	25
Ilustración 12: Logo Delta 3D.....	26
Ilustración 13: Logo Blender.....	27
Ilustración 14: Street of rage.....	30
Ilustración 15: Street fighter.....	31
Ilustración 16: Tekken.....	31
Ilustración 17: Sonic (Mega Drive).....	31
Ilustración 18: Mario Bross 3.....	31
Ilustración 19: Need for speed Underground 2.....	32
Ilustración 20: Our Run.....	32
Ilustración 21: Pro Evolvion Soccer.....	32
Ilustración 22: Monkey Island	33
Ilustración 23: Run Away.....	33
Ilustración 24: Rock Band.....	33
Ilustración 25: RollerCoaster Tycoo.....	34
Ilustración 26: The Sims.....	34
Ilustración 27: Recreativa Tron.....	35
Ilustración 28: Aero Fighters 2 (neo geo).....	59
Ilustración 29: Aero fighters	59
Ilustración 30: Aero fighters selección de presonaje).....	59
Ilustración 31: Aero fighters.....	59
Ilustración 32: Ikaruga menú.....	60
Ilustración 33: Ikaruga.....	60
Ilustración 34: 1942 menú.....	60
Ilustración 35: 1942.....	60
Ilustración 36: Far Exile - Logo (Elephant3).....	73
Ilustración 37: Far Exile - Título.....	74
Ilustración 38: Far Exile - 1ª Fase.....	74
Ilustración 39: Far Exile - Game Over.....	75
Ilustración 40: Far Exile - Créditos.....	75
Ilustración 41: Far Exile - Controles.....	76
Ilustración 42: Ryu Kayatsuri.....	79
Ilustración 43: Lord Agnar.....	80
Ilustración 44: Gor C.....	81
Ilustración 45: Robert W. Cooper.....	81
Ilustración 46: Teothorus “El Santo”.....	82
Ilustración 47: N° 41 (Padre de los Technologos).....	82
Ilustración 48: Mente Maestra.....	83

Ilustración 49: Robert W. Cooper.....	84
Ilustración 50: Un ojo el Inmortal.....	85
Ilustración 51: Un ojo el Inmortal.....	86
Ilustración 52: Lord Agnar.....	86
Ilustración 53: Bocetos1.....	87
Ilustración 54: Bocetos2.....	88
Ilustración 55: Bocetos3.....	89
Ilustración 56: Bocetos4.....	90
Ilustración 57: Bocetos5.....	91
Ilustración 58: Bocetos6.....	91
Ilustración 59: Palmera y arbusto.....	92
Ilustración 60: Palmera y arbusto.....	92
Ilustración 61: Pantallas1.....	93
Ilustración 62: Pantallas2.....	94
Ilustración 63: Nave de Teothorus “El Santo”.....	95
Ilustración 64: Nave de Ryu Kayatsuri.....	96
Ilustración 65: Nave de Gor C.....	96
Ilustración 66: Enemigo final 2.....	97
Ilustración 67: Lanazamisiles.....	97
Ilustración 68: Enemigo3.....	98
Ilustración 69: Enemigo2.....	98
Ilustración 70: Objeto1.....	99
Ilustración 71: Objeto2.....	99

G. Bibliografía y otros enlaces.

1. Bibliografía recomendada.

- Blender Game Kit 2
- Blender. Curso de iniciación. Mercé Galan. Inforbook's. ISBN-10 84-96097-96-X
- The Essential Blender.
- Bounce, Tumble and Splash!

2. Estructura del directorio del proyecto.

A continuación un listado ordenado por niveles de las carpetas contenidas en el proyecto con una breve descripción del contenido :

- Proyecto
 - Documentación PFC
 - Betas : Cada una de las versiones de prueba obtenidas para probar el funcionamiento de algunas implementaciones.
 - Diseño
 - Diseños lápiz : Dibujos creados por Juan Rovira (Jefe de proyecto) y Lorena (Ilustradora).
 - Modelos 3D : archivos .blend creados con Blender de cada unos de los objetos del videojuego.
 - Planos : imágenes de planos reales de aviones.
 - Programa : Recopilación de programa utilizados para el desarrollo de este proyecto.
 - Proyecto blend : Proyecto completo del videojuego desarrollado en Blender. Incluye todos los archivos.

- Resultados : Imágenes obtenidas de capturas de pantalla mientras el juego estaba en marcha. También incluye renderizaciones en planos especiales de algunos diseños.

- Varios
 - Diseño y desarrollo de videojuegos en Blender (UPV)
 - Game Engine Python API : Documentación completa del código de la API de python para el motor de videojuegos de Blender.
 - Imágenes de referencia : Capturas de videojuegos clásicos y fotografías de maquinas recreativas referenciadas en este proyecto.
 - Pythonlopes : Archivos de una página html con un tutorial de python.
 - Yofrankie-windows : versión del videojuego mas famoso para windows hecho con blender.

- Far Exile (beta 0.9.6) : Contiene el videojuego creado para este proyecto.

3. Otros enlaces.

Documentos de diseño : <http://www.advertainmen.com/search/label/documento%20de%20diseño>

Comparativas de motores gráficos :

http://www.angelfire.com/indie/jetstream/crystalspace_review.htm

Foros :

<http://blenderartists.org/forum/forumdisplay.php?34-Game-Engine>

<http://www.gfxartist.com/>

Magazine dedicado a Blender :

<http://blenderart.org/>

<http://www.eichnews-online.com/blender.php>

Página dedicada a Blender con tutoriales de todo tipo :

http://www.fisicomolon.com/joomla/index.php?option=com_content&view=frontpage&Itemid=1&lang=es

Mejorar el Frame Rate :

<http://3dmedieval.blogspot.com.es/2011/05/increasing-frame-rate-in-blender-game.html>

H. Licencias y contratos.

Licencia de Documentación Libre GNU (traducción)

<http://gugs.sindominio.net/licencias/fdl-es.html>

Versión 1.1, Marzo de 2000

Esta es la GNU Free Document License (GFDL), versión 1.1 (de marzo de 2.000), que cubre manuales y documentación para el software de la Free Software Foundation, con posibilidades en otros campos. La traducción[1] no tiene ningún valor legal, ni ha sido comprobada de acuerdo a la legislación de ningún país en particular. Vea el [original](#)

Los autores de esta traducción son:

- Igor Támara <ikks@bigfoot.com>
- Pablo Reyes <reyes_pablo@hotmail.com>
- Revisión : Vladimir Támara P. <vtamara@gnu.org>

Copyright © 2000

Free Software Foundation, nc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Se permite la copia y distribución de copias literales de este documento de licencia, pero no se permiten cambios.

0. Preámbulo

El propósito de esta licencia es permitir que un manual, libro de texto, u otro documento escrito sea "libre" en el sentido de libertad: asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, de manera comercial o no. En segundo término, esta licencia preserva para el autor o para quien publica una manera de obtener reconocimiento por su trabajo, al tiempo que no se consideran responsables de las modificaciones realizadas por terceros.

Esta licencia es una especie de "copyleft" que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Esto complementa la Licencia Pública General GNU, que es una licencia de copyleft diseñada para el software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: Un programa libre debe venir con los manuales que ofrezcan la mismas libertades que da el software. Pero esta licencia no se limita a manuales de software; puede ser usada para cualquier trabajo textual, sin tener en cuenta su temática o si se publica como libro impreso. Recomendamos esta licencia principalmente para trabajos cuyo fin sea instructivo o de referencia.

1. Aplicabilidad y definiciones

Esta Licencia se aplica a cualquier manual u otro documento que contenga una nota del propietario de los derechos que indique que puede ser distribuido bajo los términos de la Licencia. El "Documento", en adelante, se refiere a cualquiera de dichos manuales o trabajos. Cualquier miembro del público es un licenciataria, y será denominado como "Usted".

Una "Versión Modificada" del Documento significa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma.

Una "Sección Secundaria" es un apéndice titulado o una sección preliminar al prólogo del Documento que tiene que ver exclusivamente con la relación de quien publica o, los autores del Documento o, el tema general del Documento(o asuntos relacionados) y cuyo contenido no entra directamente en este tema general. (Por ejemplo, si el Documento es en parte un texto de matemáticas, una Sección Secundaria puede no explicar matemáticas.) La relación puede ser un asunto de conexión histórica, o de posición legal, comercial, filosófica, ética o política con el tema o la materia del texto.

Las "Secciones Invariantes" son ciertas Secciones Secundarias cuyos títulos son denominados como Secciones Invariantes, en la nota que indica que el documento es liberado bajo esta licencia.

Los "Textos de Cubierta" son ciertos pasajes cortos de texto que se listan, como Textos de Portada o Textos de Contra Portada, en la nota que indica que el documento es liberado bajo esta Licencia.

Una copia "Transparente" del Documento, significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público general, cuyos contenidos pueden ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por pixeles) de programas genéricos de dibujo o (para dibujos) algún editor gráfico ampliamente disponible, y que sea adecuado para exportar a formateadores de texto o para traducción automática a una variedad de formatos adecuados para ingresar a formateadores

de texto. Una copia hecha en un formato de un archivo que no sea Transparente, cuyo formato ha sido diseñado para impedir o dificultar subsecuentes modificaciones posteriores por parte de los lectores no es Transparente. Una copia que no es "Transparente" es llamada "Opaca".

Como ejemplos de formatos adecuados para copias Transparentes están el ASCII plano sin formato, formato de Texinfo, formato de LaTeX, SGML o XML usando un DTD disponible ampliamente, y HTML simple que sigue los estándares, diseñado para modificaciones humanas. Los formatos Opacos incluyen PostScript, PDF, formatos propietarios que pueden ser leídos y editados únicamente en procesadores de palabras propietarios, SGML o XML para los cuáles los DTD y/o herramientas de procesamiento no están disponibles generalmente, y el HTML generado por máquinas producto de algún procesador de palabras solo para propósitos de salida.

La "Portada" en un libro impreso significa, la portada misma, más las páginas siguientes necesarias para mantener la legibilidad del material, que esta Licencia requiere que aparezca en la portada. Para trabajos en formatos que no tienen Portada como tal, "Portada" significa el texto cerca a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del trabajo.

2. Copia literal

Puede copiar y distribuir el Documento en cualquier medio, sea en forma comercial o no, siempre y cuando esta Licencia, las notas de derecho de autor, y la nota de licencia que indica que esta Licencia se aplica al Documento se reproduzca en todas las copias, y que usted no adicione ninguna otra condición a las expuestas en esta Licencia. No puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3.

También puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

3. Copiado en cantidades

Si publica copias impresas del Documento que sobrepasen las 100, y la nota de Licencia del Documento exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible, todos esos textos de Cubierta: Textos Frontales en la cubierta frontal, y Textos Posteriores de Cubierta en la Cubierta Posterior. Ambas cubiertas deben identificarlo a Usted clara y legiblemente como quien publica tales copias. La Cubierta Frontal debe mostrar el título completo con todas las palabras igualmente prominentes y visibles. Además puede adicionar otro

material en la cubierta. Las copias con cambios limitados en las cubiertas, siempre que preserven el título del Documento y satisfagan estas condiciones, puede considerarse como copia literal.

Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros (tantos como sea razonable colocar) en la cubierta real, y continuar el resto en páginas adyacentes.

Si publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente que pueda ser leída por una máquina con cada copia Opaca, o entregar en o con cada copia Opaca una dirección en red de computador públicamente-accesible conteniendo una copia completa Transparente del Documento, sin material adicional, a la cual el público en general de la red pueda acceder a bajar anónimamente sin cargo usando protocolos de standard público. Si usted hace uso de la última opción, deberá tomar medidas necesarias, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio por lo menos un año después de su última distribución de copias Opacas (directamente o a través de sus agentes o distribuidores) de esa edición al público.

Se solicita, aunque no es requisito, que contacte a los autores del Documento antes de redistribuir cualquier gran número de copias, para permitirle la oportunidad de que le provean una versión del Documento.

4. Modificaciones

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada haciendo el rol del Documento, por lo tanto licenciando la distribución y modificación de la Versión Modificada a quienquiera que posea una copia de este. En adición, debe hacer lo siguiente en la Versión Modificada:

- **A.** Uso en la Portada (y en las cubiertas, si hay alguna) de un título distinto al del Documento, y de versiones anteriores (que deberían, si hay alguna, estar listados en la sección de Historia del Documento). Puede usar el mismo título que versiones anteriores al original siempre que quién publicó la primera versión lo permita.
- **B.** Listar en la Portada, como autores, una o más personas o entidades responsables por la autoría o las modificaciones en la Versión Modificada, junto con por lo menos cinco de los autores principales del Documento (Todos sus autores principales, si hay menos de cinco).

- **C.** Estado en la Portada del nombre de quién publica la Versión Modificada, como quien publica.
- **D.** Preservar todas las notas de derechos de autor del Documento.
- **E.** Adicionar una nota de derecho de autor apropiada a sus modificaciones adyacentes a las otras notas de derecho de autor.
- **F.** Incluir, inmediatamente después de la nota de derecho de autor, una nota de licencia dando el permiso público para usar la Versión Modificada bajo los términos de esta Licencia, de la forma mostrada en la Adición (LEGAL)abajo.
- **G.** Preservar en esa nota de licencia el listado completo de Secciones Invariantes y en los Textos de las Cubiertas que sean requeridos como se especifique en la nota de Licencia del Documento
- **H.** Incluir una copia sin modificación de esta Licencia.
- **I.** Preservar la sección llamada "Historia", y su título, y adicionar a esta una sección estableciendo al menos el título, el año, los nuevos autores, y quién publicó la Versión Modificada como reza en la Portada. Si no hay una sección titulada "Historia" en el Documento, crear una estableciendo el título, el año, los autores y quien publicó el Documento como reza en la Portada, añadiendo además un artículo describiendo la Versión Modificada como se estableció en el punto anterior.
- **J.** Preservar la localización en red, si hay , dada en la Documentación para acceder públicamente a una copia Transparente del Documento, tanto como las otras direcciones de red dadas en el Documento para versiones anteriores en las cuáles estuviese basado. Estas pueden ubicarse en la sección "Historia". Se puede omitir la ubicación en red para un trabajo que sea publicado por lo menos 4 años antes que el mismo Documento, o si quien publica originalmente la versión da permiso explícitamente.
- **K.** En cualquier sección titulada "Agradecimientos" o "Dedicatorias", preservar el título de la sección, y preservar en la sección toda la sustancia y el tono de los agradecimientos y/o dedicatorias de cada contribuyente que estén incluidas.
- **L.** Preservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección. **M.** Borrar cualquier sección titulada "Aprobaciones". Tales secciones no pueden estar incluidas en las Versiones Modificadas.

- **M.** Borrar cualquier sección titulada "Aprobaciones". Tales secciones no pueden estar incluidas en las Versiones Modificadas.
- **N.** No retitular ninguna sección existente como "Aprobaciones" o conflictuar con título con alguna Sección Invariante.

Si la Versión Modificada incluye secciones o apéndices nuevos o preliminares al prólogo que califican como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, adicione sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección.

Puede adicionar una sección titulada "Aprobaciones", siempre que contenga únicamente aprobaciones de su Versión Modificada por varias fuentes--por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como un standard.

Puede adicionar un pasaje de hasta cinco palabras como un Texto de Cubierta Frontal, y un pasaje de hasta 25 palabras como un texto de Cubierta Posterior, al final de la lista de Textos de Cubierta en la Versión Modificada. Solamente un pasaje de Texto de Cubierta Frontal y un Texto de Cubierta Posterior puede ser adicionado por (o a manera de arreglos hechos por) una entidad. Si el Documento ya incluye un texto de cubierta para la misma cubierta, previamente adicionado por usted o por arreglo hecho por la misma entidad, a nombre de la cual está actuando, no puede adicionar otra; pero puede reemplazar la anterior, con permiso explícito de quien publicó anteriormente tal cubierta.

El(los) autor(es) y quien(es) publica(n) el Documento no dan con esta Licencia permiso para usar sus nombres para publicidad o para asegurar o implicar aprobación de cualquier Versión Modificada.

5. Combinando documentos

Puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, y listadas todas como Secciones Invariantes del trabajo combinado en su nota de licencia.

El trabajo combinado necesita contener solamente una copia de esta Licencia, y múltiples Secciones Invariantes Idénticas pueden ser reemplazadas por una sola copia. Si hay múltiples Secciones Invariantes con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único adicionándole al final de este, en paréntesis, el nombre del autor o

de quien publicó originalmente esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones Invariantes en la nota de licencia del trabajo combinado.

En la combinación, debe combinar cualquier sección titulada "Historia" de los varios documentos originales, formando una sección titulada "Historia"; de la misma forma combine cualquier sección titulada "Agradecimientos", y cualquier sección titulada "Dedicatorias". Debe borrar todas las secciones tituladas "Aprobaciones."

6. Colecciones de documentos

Puede hacer una colección consistente del Documento y otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en los varios documentos con una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para una copia literal de cada uno de los documentos en cualquiera de todos los aspectos.

Puede extraer un solo documento de una de tales colecciones, y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los otros aspectos concernientes a la copia literal de tal documento.

7. Agregación con trabajos independientes

Una recopilación del Documento o de sus derivados con otros documentos o trabajos separados o independientes, en cualquier tipo de distribución o medio de almacenamiento, no como un todo, cuenta como una Versión Modificada del Documento, teniendo en cuenta que ninguna compilación de derechos de autor sea clamada por la recopilación. Tal recopilación es llamada un "agregado", y esta Licencia no aplica a los otros trabajos auto-contenidos y por lo tanto compilados con el Documento, o a cuenta de haber sido compilados, si no son ellos mismos trabajos derivados del Documento.

Si el requerimiento de la sección 3 del Texto de la Cubierta es aplicable a estas copias del Documento, entonces si el Documento es menor que un cuarto del agregado entero, Los Textos de la Cubierta del Documento pueden ser colocados en cubiertas que enmarquen solamente el Documento entre el agregado. De otra forma deben aparecer en cubiertas enmarcando todo el agregado.

8. Traducción

La Traducción es considerada como una clase de modificación, Así que puede distribuir traducciones del Documento bajo los términos de la sección 4. Reemplazar las Secciones Invariantes con traducciones requiere permiso especial de los dueños de derecho de autor, pero puede incluir traducciones de algunas o todas las Secciones Invariantes adicionalmente a las versiones originales de las Secciones Invariantes. Puede incluir una traducción de esta Licencia siempre que incluya también la versión Inglesa de esta Licencia. En caso de un desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la versión original en Inglés prevalecerá.

9. Terminación

No se puede copiar, modificar, sublicenciar, o distribuir el Documento excepto por lo permitido expresamente bajo esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y serán automáticamente terminados sus derechos bajo esa licencia. De todas maneras, los terceros que hayan recibido copias, o derechos, de su parte bajo esta Licencia no tendrán por terminadas sus licencias siempre que tales personas o entidades se encuentren en total conformidad con la licencia original.

10. Futuras revisiones de esta licencia

La Free Software Foundation puede publicar nuevas, revisadas versiones de la Licencia de Documentación Libre GNU de tiempo en tiempo. Tales nuevas versiones serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar problemas o intereses. Vea <http://www.gnu.org/copyleft/>.

Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que una versión numerada particularmente de esta licencia o "cualquier versión posterior" se aplica a esta, tiene la opción de seguir los términos y condiciones de la versión especificada o cualquiera posterior que ha sido publicada(no como un borrador)por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada(no como un borrador) por la Free Software Foundation.

Addendum

Para usar esta licencia en un documento que usted haya escrito, incluya una copia de la Licencia en el documento y ponga el siguiente derecho de autor y nota de licencia justo después del título de la página:

Derecho de Autor © Año Su Nombre.

Permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; con las Secciones Invariantes siendo LISTE SUS TÍTULOS, con los siendo LISTELO el texto de la Cubierta Frontal, y siendo LISTELO el texto de la Cubierta Posterior. Una copia de la licencia es incluida en la sección titulada "Licencia de Documentación Libre GNU".

Si no tiene Secciones Invariantes, escriba "Sin Secciones Invariantes" en vez de decir cuáles son invariantes. Si no tiene Texto de Cubierta Frontal, escriba "Sin Texto de Cubierta Frontal" en vez de "siendo LISTELO el texto de la Cubierta Frontal"; Así como para la Cubierta Posterior.

Si su documento contiene ejemplos de código de programa no triviales, recomendamos liberar estos ejemplos en paralelo bajo su elección de licencia de software libre, tal como la Licencia de Público General GNU, para permitir su uso en software libre.