



**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERÍA TÉCNICA EN INFORMÁTICA DE**  
**SISTEMAS**

**SHOOT 'EM UP DESARROLLADO CON OGRE 3D**

José Carlos Pavón Montañez

11 de febrero de 2010





## ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

SHOOT 'EM UP DESARROLLADO CON OGRE 3D

- Departamento: Lenguajes y sistemas informáticos
- Director del proyecto: Manuel Palomo Duarte
- Autor del proyecto: José Carlos Pavón Montañez

Cádiz, 11 de febrero de 2010

Fdo: José Carlos Pavón Montañez



Copyright © 2010 José Carlos Pavón Montañez. 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

<b>1. Introducción</b>	<b>5</b>
1.1. Objetivos . . . . .	5
1.2. Estructura del documento . . . . .	5
<b>2. Fundamentos</b>	<b>7</b>
2.1. Historia de los videojuegos . . . . .	7
2.2. Matamarcianos . . . . .	11
<b>3. Planificación</b>	<b>15</b>
3.1. Planificación temporal . . . . .	15
3.1.1. Fases del desarrollo . . . . .	15
3.1.2. Diagrama de Gantt . . . . .	18
<b>4. Diseño</b>	<b>19</b>
4.1. Historia . . . . .	19
4.2. Gráficos . . . . .	21
4.3. Sonido . . . . .	25
4.4. Interfaz . . . . .	25
4.4.1. Menú Principal . . . . .	25
4.4.2. Elementos de la interfaz . . . . .	30
4.5. Fase . . . . .	32
4.6. Elementos del juego . . . . .	35
4.7. Casos de uso . . . . .	41
4.8. Diagrama de clases conceptuales . . . . .	45
4.9. Contrato de las operaciones . . . . .	46
4.9.1. Clase Sonido . . . . .	46
4.9.2. Clase Fase1 . . . . .	48
4.9.3. Clase Fase2 . . . . .	48
4.9.4. Clase Fase3 . . . . .	49
4.9.5. Clase Fase4 . . . . .	49
4.9.6. Clase Opciones . . . . .	49

4.9.7. Clase Records . . . . .	51
4.9.8. Clase Menu . . . . .	51
4.9.9. Clase Input . . . . .	52
4.9.10. Clase Juego . . . . .	52
<b>5. Desarrollo</b>	<b>59</b>
5.1. Modelado 3D . . . . .	59
5.1.1. Creación de las texturas . . . . .	64
5.1.2. Exportación de los modelos . . . . .	69
5.1.3. Problemas en la exportación de modelos . . . . .	71
5.2. Sonido . . . . .	72
5.2.1. Música . . . . .	73
5.3. Codificación . . . . .	74
5.3.1. Motor Gráfico . . . . .	74
5.3.2. Entrada y Salida . . . . .	75
5.3.3. Sonido . . . . .	76
5.3.4. Física . . . . .	76
5.3.5. Interfaz . . . . .	77
5.3.6. Archivos de configuración . . . . .	77
5.4. Pruebas . . . . .	78
5.4.1. Pruebas de colisiones . . . . .	78
5.4.2. Pruebas de menús . . . . .	78
5.4.3. Pruebas de récords . . . . .	78
5.4.4. Pruebas de opciones . . . . .	79
5.5. Documentación . . . . .	79
<b>6. Conclusiones</b>	<b>81</b>
6.1. Trabajo realizado . . . . .	81
6.2. Ampliaciones . . . . .	81
<b>A. Definiciones y abreviaturas</b>	<b>83</b>
<b>B. Publicación</b>	<b>91</b>
<b>C. Instalación</b>	<b>93</b>
C.1. Instalación de las dependencias . . . . .	93
C.1.1. Distribuciones basadas en Debian . . . . .	93
C.1.2. Desde el código fuente . . . . .	93
C.2. Compilación . . . . .	94
C.3. Ejecución . . . . .	94
C.4. Posibles problemas . . . . .	94



C.4.1. Compilación . . . . .	94
C.4.2. Juego Ralentizado . . . . .	94
<b>D. Manual de usuario</b>	<b>95</b>
D.1. Controles . . . . .	95
D.1.1. Menú . . . . .	95
D.1.2. Juego . . . . .	96
D.2. Juego . . . . .	97
D.2.1. Disparos . . . . .	97
D.2.2. Potenciadores . . . . .	98
D.2.3. Interfaz . . . . .	99
D.2.4. Enemigos finales . . . . .	100
D.3. Ajustes . . . . .	101
<b>Bibliografía</b>	<b>103</b>
<b>GNU Free Documentation License</b>	<b>107</b>
1. APPLICABILITY AND DEFINITIONS . . . . .	107
2. VERBATIM COPYING . . . . .	109
3. COPYING IN QUANTITY . . . . .	109
4. MODIFICATIONS . . . . .	110
5. COMBINING DOCUMENTS . . . . .	112
6. COLLECTIONS OF DOCUMENTS . . . . .	113
7. AGGREGATION WITH INDEPENDENT WORKS . . . . .	113
8. TRANSLATION . . . . .	113
9. TERMINATION . . . . .	114
10. FUTURE REVISIONS OF THIS LICENSE . . . . .	114
11. RELICENSING . . . . .	115
ADDENDUM: How to use this License for your documents . . . . .	115



# Índice de figuras

2.1. Captura de Spacewar! Extraído de wikipedia.org . . . . .	12
2.2. Captura de Space Invaders. Extraído de wikipedia.org . . . . .	12
3.1. Diagrama de Gantt . . . . .	18
4.1. Texturas del juego . . . . .	24
4.2. Menú principal . . . . .	26
4.3. Menú de selección de misión . . . . .	27
4.4. Menú de opciones . . . . .	28
4.5. Menú de récords . . . . .	28
4.6. Nuevo récord . . . . .	29
4.7. Menú de créditos . . . . .	29
4.8. Diagrama de menús . . . . .	30
4.9. Puntuación actual . . . . .	30
4.10. Número de vidas . . . . .	31
4.11. Barra de vida . . . . .	31
4.12. Tiempo de escudo y munición . . . . .	31
4.13. Texto de localización . . . . .	32
4.14. Introducción de la misión 2 . . . . .	32
4.15. Misión 1 . . . . .	33
4.16. Misión 2 . . . . .	34
4.17. Misión 3 . . . . .	34
4.18. Misión 4 . . . . .	35
4.19. Nave jugador . . . . .	35
4.20. Nave enemiga . . . . .	36
4.21. Helicóptero . . . . .	36
4.22. Platillo volante . . . . .	36
4.23. Meteorito . . . . .	37
4.24. Tanque . . . . .	37
4.25. Enemigo final fase 1 . . . . .	38
4.26. Enemigo final fase 3 . . . . .	38
4.27. Enemigo final fase 4 . . . . .	38

4.28. Disparo especial . . . . .	39
4.29. Bomba . . . . .	39
4.30. Potenciador de vida . . . . .	40
4.31. Potenciador de escudo . . . . .	40
4.32. Potenciador de disparo . . . . .	40
4.33. Diagrama de casos de uso . . . . .	41
4.34. Diagrama de clases conceptuales . . . . .	46
5.1. Blender. Extraido de <a href="http://blogs.gamefilia.com">http://blogs.gamefilia.com</a> . . . . .	61
5.2. Interfaz de Blender . . . . .	62
5.3. Extrusión, escalado, rotación... . . . .	62
5.4. Extrusión, escalado, rotación... . . . .	63
5.5. Añadiendo los detalles . . . . .	63
5.6. Resultado . . . . .	64
5.7. Gimp. Extraido de <a href="https://commons.wikimedia.org">commons.wikimedia.org</a> . . . . .	65
5.8. Costuras en Blender . . . . .	65
5.9. Patrón de la textura . . . . .	66
5.10. Interfaz de GIMP . . . . .	66
5.11. Textura terminada . . . . .	67
5.12. Helicóptero con texturas . . . . .	67
5.13. Resultado final . . . . .	68
5.14. Tanque . . . . .	68
5.15. Enemigo Final 1 . . . . .	69
5.16. Bomba . . . . .	69
5.17. Ogre mesh exporter . . . . .	70
5.18. Preferencias de Ogre mesh exporter . . . . .	71
5.19. Audacity. Extraido de <a href="http://audacity.sourceforge.net">audacity.sourceforge.net</a> . . . . .	72
5.20. Editando audio con Audacity . . . . .	73
5.21. SoundKonverter . . . . .	74
5.22. Ogre 3D. Extraido de <a href="http://wikipedia.org">wikipedia.org</a> . . . . .	75
5.23. SDL. Extraido de <a href="http://libsdl.org">libsdl.org</a> . . . . .	76
5.24. Colisiones . . . . .	77
D.1. Teclas durante el menú . . . . .	96
D.2. Teclas durante el juego . . . . .	97
D.3. Bomba . . . . .	98
D.4. Puntuación actual . . . . .	99
D.5. Número de vidas . . . . .	99
D.6. Barra de vida . . . . .	99
D.7. Tiempo de escudo y munición . . . . .	100

## **Agradecimientos**

Me gustaría agradecer a mi familia y a mi novia el apoyo recibido y a Pablo y Fabian la ayuda e ideas que me han aportado.



## **Notación y formato**

Cuando nos refiramos a un programa en concreto, utilizaremos la notación:  
*Blender*.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:  
`apt-get`.





# Capítulo 1

## Introducción

### 1.1. Objetivos

El objetivo principal de este proyecto es crear un juego de matamarcianos desarrollado con software libre. El juego se hará para el sistema operativo GNU/Linux (aunque pueda ser fácilmente portado a otras plataformas), con la intención de ayudar a fomentar el uso de este sistema operativo, ya que una de las críticas que más recibe es la falta de juegos.

Con este juego no se pretende competir con otros juegos comerciales ya que no va a ser realizado por un equipo de personas, sino por una sola; así que lo que se pretende es aprender, en líneas generales, cada una de las tareas necesarias para crear un videojuego en tres dimensiones. Por lo tanto, el juego no será espectacular en el apartado gráfico, pero se espera que sea adictivo y jugable. Estará pensado para jugadores casuales, ya que no será necesarias muchas horas para adquirir la destreza necesaria para jugar al ser simple y fácil.

El juego ha sido liberado bajo licencia GPL v3 (GNU General Public License). Se incluyen los términos de la licencia en inglés en el archivo License.txt.

Para mas información visite:

<http://www.gnu.org/licenses>

### 1.2. Estructura del documento

En esta sección describiremos el contenido de cada uno de los capítulos y apéndices de la memoria.

**Capítulo 1** Trata sobre las licencias tanto del juego como de la memoria y expone los objetivos principales del proyecto.

**Capítulo 2** Se muestran los conceptos básicos necesarios sobre los videojuegos para una mejor comprensión de la memoria.

**Capítulo 3** Se expone la planificación temporal del proyecto.

**Capítulo 4** Se muestra el diseño del juego, tanto de la parte artística, como pueden ser los gráficos, el sonido o la interfaz; hasta los casos de uso y documentación de las funciones.

**Capítulo 5** Se explica la forma de llevar a cabo cada una de las tareas necesarias para el desarrollo del proyecto.

**Capítulo 6** Trata sobre las conclusiones que se han podido extraer del proyecto y de posibles mejoras o ampliaciones futuras.

**Apéndice A** Explica las principales definiciones y abreviaturas que nos serán de ayuda a la hora de comprender mejor algunos conceptos de la memoria.

**Apéndice B** Muestra dónde podemos encontrar tanto el código fuente del juego como la documentación y los pasos que debemos seguir para obtenerlos.

**Apéndice C** Aquí encontraremos las instrucciones necesarias para compilar e instalar el juego, así como soluciones a posibles errores que podamos encontrar.

**Apéndice D** En este apéndice encontraremos el manual de usuario, en el que podemos encontrar explicaciones sobre los controles del juego, elementos de la interfaz y algunos consejos para acabar con los enemigos finales.

# Capítulo 2

## Fundamentos

### 2.1. Historia de los videojuegos

Texto extraído de wikipedia.org

Un videojuego[[WikiEs](#)] (del inglés video game) o juego de vídeo es un programa de computación creado para el entretenimiento, basado en la interacción entre una o varias personas y un aparato electrónico (ya sea un ordenador, un sistema arcade, una videoconsola, un dispositivo handheld o actualmente un teléfono celular) que ejecuta dicho videojuego. En muchos casos, estos recrean entornos y situaciones virtuales en los que el jugador puede controlar a uno o varios personajes (o cualquier otro elemento de dicho entorno), para conseguir uno o varios objetivos por medio de unas reglas determinadas.

Se desarrollan en un monitor de computadora o en una televisión, y en los que el programa va grabado en cartuchos o discos ópticos, discos magnéticos o tarjetas de memoria especiales para juegos. Algunos son de bolsillo. La enorme popularidad alcanzada por estos juegos a finales de la década de 1970 ha dado origen a una importante industria.

Los jugadores emplean una serie de teclas o botones y una palanca, también llamada joystick. La partida se juega entre una persona y la máquina, o entre dos o más personas que compiten con la máquina o entre sí.

Los videojuegos se dividen en distintas categorías: educativos, de aventuras y deportivos. Los más populares emplean sonidos reales y colores, además de rápidos efectos visuales. Los juegos deportivos, como el fútbol, el baloncesto o el hockey sobre hielo, adquirieron especial popularidad a finales de la década de 1980, cuando determinados equipos profesionales prestaron su nombre a estas versiones en vídeo de su deporte.

Los establecimientos dedicados a la venta exclusiva de videojuegos son ya clásicos en la industria del ocio del mundo entero. Los sistemas de 16 bits, intro-

ducidos por las empresas japonesas a comienzos de la década de 1990, mejoraron notablemente la calidad de los gráficos de los videojuegos destinados al uso doméstico. El CD-ROM (un sistema de memoria con enorme capacidad de almacenamiento) ofrece hoy una amplia variedad de juegos a precios muy competitivos. En el futuro, el uso de la televisión de alta definición y de las líneas de telecomunicaciones para la transmisión de este tipo de juegos contribuirá a aumentar aún más su realismo.

Las críticas a los videojuegos parten de los efectos negativos que el hecho de pasar demasiado tiempo ante la pantalla e inhibirse por completo en un universo de fantasía puede tener en el desarrollo emocional de los niños, adolescentes y jóvenes. Ocasionalmente se ha comprobado que la rapidez con que se mueven los gráficos puede provocar ataques en las personas que padecen diversos tipos de epilepsia.

Los defensores de los videojuegos afirman que estos juegos enseñan a resolver problemas técnicos, estimulan la habilidad de los jugadores y familiarizan a los niños y a los adolescentes con el uso de los equipos informáticos. Incluso sostienen que mejoran la comunicación cuando se juega en familia. Los videojuegos se emplean también como entretenimiento en clínicas y hospitales, así como en ciertas terapias de rehabilitación.

La industria del videojuego surgió con fuerza en Estados Unidos, Europa y Australia con la llegada del tenis de mesa y otros juegos muy sencillos a finales de la década de 1970 y se extendió rápidamente por todo el mundo. Tras conocer un crecimiento espectacular a lo largo de la década de 1980, la industria japonesa (especialmente la Nintendo Company Ltd.) se lanzó de lleno a perfeccionar y desarrollar la tecnología del juego, introduciendo juegos tan populares como el Super Mario Bros. Las empresas afincadas en Japón, como Nintendo y SEGA(dreamcast), continúan dominando el mercado mundial.

Desde 1993 estas dos compañías están realizando esfuerzos para controlar y establecer el contenido de los juegos. La iniciativa responde a las críticas, especialmente de los padres, preocupados por la intensificación de la violencia y la introducción de temas para adultos en los juegos infantiles.

## **Orígenes**

La historia de los videojuegos comienza aproximadamente en 1947, cuando la idea de un videojuego fue concebida y patentada por Thomas T. Goldsmith Jr. y Estle Ray Mann, quienes inscribieron una aplicación para patentes en 25 de enero de 1947 y les fue otorgada en 14 de diciembre de 1948.

Tendrían que pasar años (hasta 1952) para que el primer videojuego real de la historia fuese creado. OXO, que así se llamaba, era en realidad un juego gráfico computarizado, fruto de la tesis doctoral de Alexander Sandy Douglas para

la Universidad de Cambridge en Inglaterra sobre la interactividad entre computadoras y seres humanos. Es una versión del conocido "Tres en línea" escrito para la computadora EDSAC (el primer calculador electrónico de la historia). OXO podía tomar sus decisiones en función de los movimientos del jugador, que transmitía las órdenes a través de un conexión dial integrado en el sistema. Pero aun así existen discrepancias a la hora de considerar a OXO como el primer videojuego de la historia, ya que no cuenta con vídeo animación.

En 1957, se crea William Higinbotham's Tennis for Two y en 1962, Spacewar! fue desarrollado, siendo considerado como el primer videojuego computacional. Más tarde, en 1974, Computer Space salió a la venta, convirtiéndose en el primer videojuego comercial, que funcionaba con monedas. Se reconoce a Nolan Bushnell y Ted Dabney como sus creadores. Es notable también la contribución a la historia del videojuego de la primera consola casera de videojuegos, la Magnavox Odyssey creada en 1972. También fue muy importante que la gente jugara y para ello empezaron a venderlos a precios accesibles. El primer juego de vídeo fue tenis para dos.

### **Principales géneros**

**Aventura** Los videojuegos de aventura son un género de videojuegos, caracterizados por la investigación, exploración, la solución de rompecabezas, la interacción con personajes del videojuego, y un enfoque en el relato en vez de desafíos basados en reflejos.

**Disparos** En estos videojuegos el protagonista ha de abrirse camino a base de disparos. Según su temática y desarrollo pueden clasificarse en diferentes subgéneros como disparos en primera persona o "FPS", disparos en tercera persona o "TPS", disparos en primera persona multijugador masivo en línea o "MMOFPS", acción táctica, videojuegos de disparos de desplazamiento lateral o Run-And-Gun", matamarcianos y videojuegos de pistola.

**Educativos** Aunque antiguamente solo se ha usado para juegos infantiles, los juegos educativos son aquellos que enseñan mientras promueven diversión o entretenimiento. A diferencia de una enciclopedia, trata de entretener mientras se memoriza conceptos o información.

**Estrategia** Son aquellos juegos o entretenimientos en los que, el factor de la inteligencia, habilidades técnicas y planificación y despliegación, pueden hacer predominar o impulsar al jugador hacia la victoria del juego.

**Lucha** Recrean combates entre personajes controlados tanto por un jugador como por la computadora. El jugador ve a los combatientes desde una perspectiva lateral, como si se tratase de un espectador.

**Survival horror** Correspondientes al género de terror. El/los protagonistas viven aventuras dónde deben salir airosos de situaciones típicas de una película de terror (escapar de una casa llena de zombis, huir de un asesino, resolver misterios para aplacar a los fantasmas, etc.)

**Plataformas** Los videojuegos de plataformas o, simplemente, plataformas, son un género de videojuegos que se caracterizan por tener que recorrer, saltar o escalar una serie de plataformas y acantilados, con enemigos, mientras se recogen ítems para poder completar el videojuego. Suelen usar desplazamiento lateral hacia la izquierda o hacia la derecha.

**Rol** Se inspiran en los juegos de rol clásicos, donde el protagonista interpreta un papel y ha de mejorar sus habilidades mientras interactúa con el entorno y otros personajes.

**Musicales** Su desarrollo gira en torno a la música. Ejemplos: karaoke, baile, instrumentos...

**Party games** En este género los jugadores habrán de ir avanzando por turnos por un tablero virtual e ir superando diversas pruebas de tipos muy diversos en los que compiten entre sí por llegar lo antes posible a la meta, o conseguir la máxima cantidad posible de puntos.

**Simulación** Este género se caracteriza en marcar un aspecto de la vida real, llevada a un juego, donde tienes total control de lo que pasa. En muchos de ellos se enfocan tanto en inmiscuir al jugador hasta hacerlo creer que lo que está pasando es real, sobre todo con los géneros de simulación de combate o de pilotaje.

**Deportivo** Se basan en deportes, reales o ficticios, y pueden subdividirse en simuladores y "arcade"(menos realistas que los primeros).

**Carreras** Son videojuegos en los que se pilotan diferentes vehículos, ya sean reales o ficticios, para ganar en diferentes carreras. Dentro de este apartado se pueden distinguir dos variantes, arcades y simuladores.

**Sandbox** También conocidos como acción-aventura, se caracterizan por ser videojuegos en los que el jugador puede hacer lo que él quiera, como viajar libremente por el mapa del videojuego, e interactuar con casi todo lo que este a su disposición. Estos juegos son una mezcla de disparos, luchas y carreras. Son muy conocidos por su temática "antihéroe" que la mayoría tienen como protagonistas a criminales.

## **Videoconsolas**

Videoconsola[[WikiEs](#)] es un sistema electrónico de entretenimiento para el hogar que ejecuta juegos electrónicos (videojuegos) que están contenidos en cartuchos, discos ópticos, discos magnéticos o tarjetas de memoria.

Los primeros sistemas de videoconsolas fueron diseñados únicamente para jugar videojuegos pero a partir de la sexta generación de videoconsolas han sido incorporadas características importantes de multimedia, internet, tiendas virtuales, servicio en línea como: Xbox Live, PlayStation Network y Canales Wii.

## **2.2. Matamarcianos**

Texto extraído de wikipedia.org

Matamarcianos[[WikiEs](#)] (shoot 'em up en inglés) es un género de videojuegos en el que el personaje principal es un objeto volador (generalmente una nave espacial o un avión) o algún otro vehículo, que dispara contra hordas de enemigos que van apareciendo en pantalla. No se consideran matamarcianos en sí los juegos en 3D, la palabra hace alusión a juegos donde la acción se desarrolla en 2D (aunque los gráficos pueden estar dibujados con polígonos 3D).

### **Historia**

En los primeros videojuegos de este género el argumento solía ser siempre el mismo, unos extraterrestres venían a destruir la tierra y había que acabar con ellos. De ahí que durante mucho tiempo a los videojuegos se les llamara “Máquinas de marcianitos” y luego a este género en particular "Matamarcianos".

Es un género que no ha evolucionado demasiado desde su origen, primero eran pantallas estáticas, luego vinieron juegos con scroll vertical y horizontal, más tarde se incorporaron novedades como objetos que aparecían en pantalla y que daban ventajas.

A partir de ahí ha habido pequeñas variaciones como escudos, gran variedad de armas, o complejos sistemas de puntuación (en Ikaruga se premia destruir enemigos del mismo color de tres en tres, lo que le da mayor profundidad a la acción), pero la esencia de los juegos sigue siendo básicamente la misma que hace años. De ahí quizá la lenta decadencia de un género que fue estrella a lo largo de los años 80. Algunos de los matamarcianos más conocidos son Spacewar! y Space Invaders.

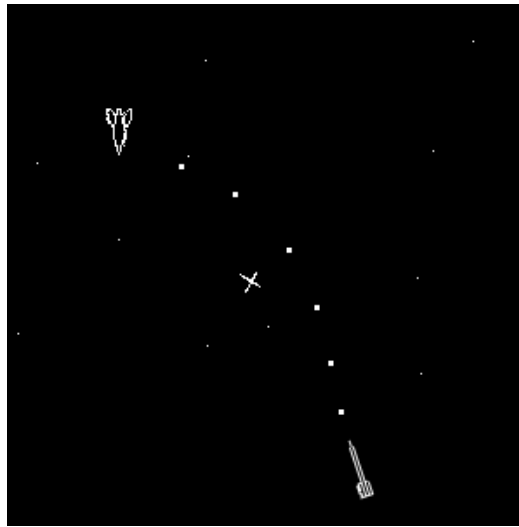


Figura 2.1: Captura de Spacewar! Extraído de wikipedia.org

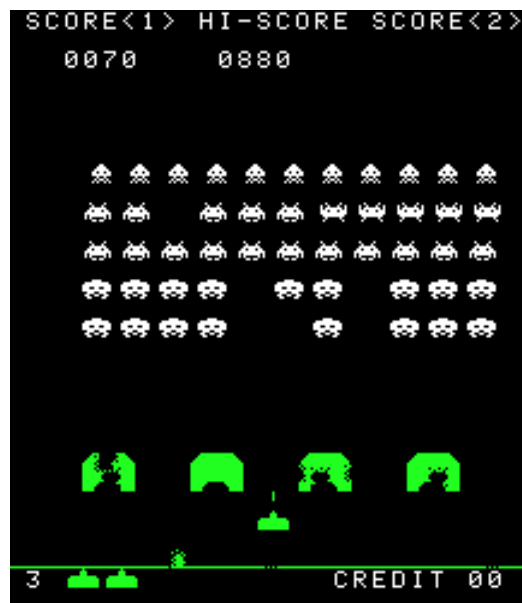


Figura 2.2: Captura de Space Invaders. Extraído de wikipedia.org



## Ejemplos

**Space Invaders** Spaces Invaders es uno de los primeros juegos shoot 'em up. Es uno de los videojuegos más importantes de la historia. Su objetivo es eliminar oleadas de alienígenas con un cañón láser y obtener la mayor cantidad de puntos posible.

**Galaxian** Galaxian expandió la fórmula creada por el Space Invaders. Al igual que ese juego, Galaxian presentaba una horda de alienígenas atacando que intercambiaban disparos con el jugador, pero, a diferencia del Space Invaders, Galaxian añadía un elemento dramático al tener a los marcianos atacando al estilo kamikaze a la nave controlada por el jugador.

**Gradius / Nemesis** Gradius se caracteriza por el uso de una barra de selección de armas, llamada "Power meter", basada en la recolección de cápsulas para comprar nuevas armas. Los jugadores controlan una nave interdimensional y deben combatir contra hordas enemigas a través de diversos escenarios.

**R-Type** En este juego el jugador controla a un caza espacial llamado R-9 que se caracteriza por un láser que se puede cargar para aumentar la fuerza de impacto contra los enemigos y a la vez, con una cápsula, llamada Force (fuerza) que se deja anclar a la nave y otorga un arma definida de acuerdo con el color o la letra que tiene el Power-Up.

**Galaga** El jugador controla una nave que debe enfrentarse contra un enjambre de marcianos que le atacarán disparando bombas y actuando como kamikazes. Cuando inicia el juego, los marcianos no aparecen inmediatamente. En su lugar, aparecen haciendo piruetas para después tomar un lugar en una formación. No disparan cuando están allí, sólo lo hacen cuando salen de la formación para tratar de acercarse lo más posible al jugador y así aniquilarlo.



# Capítulo 3

## Planificación

La planificación del proyecto que se estableció en un principio ha sido modificada en algunos aspectos, debido a la falta de tiempo y a la aparición de nuevas ideas y por lo tanto haber tenido que desarrollar nuevas clases y nuevos modelos y texturas.

A continuación se detallarán el desglose de tareas y el diagrama de Gantt del desarrollo del proyecto.

### 3.1. Planificación temporal

#### 3.1.1. Fases del desarrollo

##### Diseño

1. Arte: 6 días.

Interfaz: 1 día.

Sonido: 2 días.

Gráficos: 3 días.

2. Software: 10 días.

Casos de uso: 2 días.

Clases conceptuales: 2 días.

Contratos de operaciones: 6 días.

3. Documentación: 14 días.

Tiempo total empleado en el diseño: 30 días.

## Desarrollo

### 1. Arte: 40 días

Modelado 3D: 25 día.

Aprendizaje de Blender: 5 días.

Modelado: 20 días.

Sonido: 5 días.

Búsqueda de sonidos: 2 días.

Búsqueda de música: 2 días.

Edición del sonido con Audacity: 1 días.

Gráficos 2D: 10 días.

Aprendizaje de GIMP: 1 días.

Creación de texturas con GIMP: 7 días.

Creación de fondos con GIMP: 2 días.

### 2. Codificación: 79 días

Aprendizaje: 19 días.

Aprendizaje de OGRE: 10 días.

Aprendizaje de SDL: 2 días.

Aprendizaje de OIS: 2 días.

Clase Juego: 8 días.

Clase Menu: 3 días.

Clase Fase: 5 días.

Clase Records: 2 días.

Clase Opciones: 2 días.

Clase Sonido: 2 días.

Clase Input: 2 días.

Clase Nave: 2 días.

Clase NaveJugador: 5 días.

Clase NaveEnemiga: 5 días.

Clase Helicoptero: 2 días.

Clase Tanque: 2 días.

Clase Meteorito: 2 días.

Clase fragmentoMeteorito: 1 días.

Clase MeteoritoCabina: 2 días.

Clase Platillo: 2 días.

Clase Disparo: 2 días.

Clase DisparoTeledirigido: 1 días.

Clase Bomba: 1 días.

Clase Explosión: 1 días.

Clase Fuego: 1 días.

Clase Crater: 1 días.

Clase Cabina: 5 días.

Clase enemigoFinal1: 2 días.

Clase enemigoFinal2: 2 días.

Clase enemigoFinal3: 2 días.

3. Pruebas: 3 días

4. Manuales: 4 días

Usuario: 2 días.

Instalación: 2 días.

Tiempo total empleado en el desarrollo: 126 días.

### 3.1.2. Diagrama de Gantt

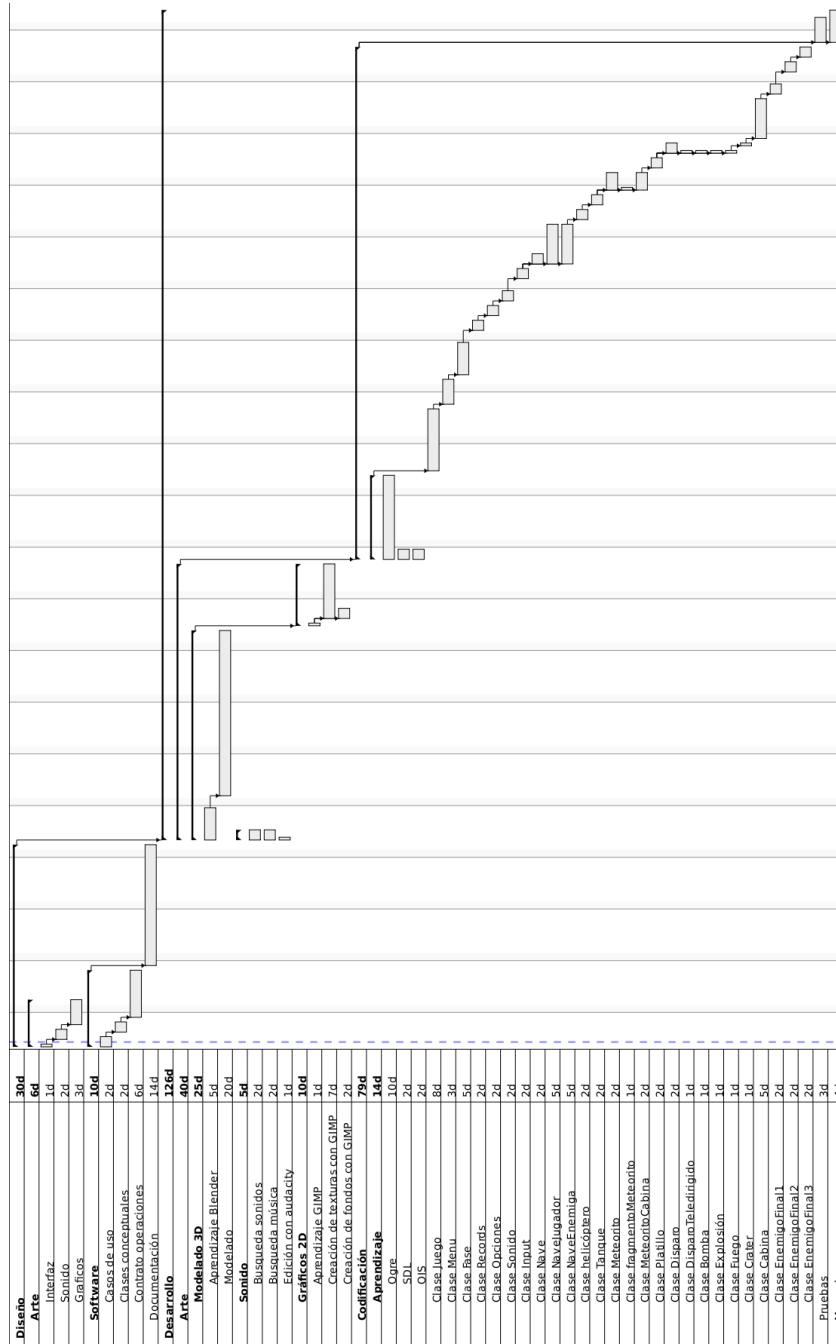


Figura 3.1: Diagrama de Gantt

# Capítulo 4

## Diseño

### 4.1. Historia

A continuación se adjunta el argumento del juego. Me gustaría agradecer a Pablo Pérez Luna su aportación al proyecto con ella.

#### Capitulo 1: Una Nueva Amenaza

Tierra año 2406, las diferentes naciones de la tierra decidieron formar una alianza secreta, conocida con el nombre de La Fundación. Durante los años siguientes La Fundación utilizó gran parte de los recursos mundiales en una investigación secreta que tenía por objetivo la creación del arma perfecta, conocida como proyecto Adan. El resultado fue un súper soldado creado para matar, sin sentimientos, capaz de sobrevivir a situaciones adversas y controlado por el alto mando de La Fundación. Nada salió como estaba previsto, Adan se volvía cada vez más rebelde y era capaz de pensar por sí solo, La Fundación veía como su más perfecta creación tomaba consciencia de sus posibilidades. Adan fue corrompiéndose y exigía cada vez un mayor poder dentro de La Fundación, hasta que, limitado por sus propios creadores, Adan decidió sublevarse y acabar con aquellos que osaban oponerse a su poder. Para ello reclutó un poderoso ejercito y estableció su centro de mando en el planeta Tenith a 567 parsecs de la tierra. Ahora, Adan tiene un único objetivo, reducir a cenizas aquello que se le ha negado y nada podrá detenerlo, solo existe una pequeña esperanza, solo una persona podría enfrentarse a Adan, el audaz y valeroso soldado de la flota aérea de La Fundación, Los Fantasmas de Acero...

## **Capítulo 2: Un Viaje Accidentado**

...tras la muerte de Galagorth, capitán de las tropas de Adan, la tierra está asegurada, al menos por ahora. Para acabar con todo esto de una vez, es necesario dar un mazazo directo en el centro de las tropas Adanesas. Nuestro valiente soldado, movido por el valor, monta en su nave y pone rumbo hacia Tenith...

...Algo le inquieta, un sentimiento le invade, es capaz de sentir la presencia de Adan, sabe que va en buen camino, escucha voces dentro de su cabeza, pero no entiende por qué, es una sensación extraña...

...un golpe sordo contra el cristal de la nave despierta a nuestro héroe de su trance... está llegando a Tenith, pero antes debe salir ileso de su cinturón de asteroides...

## **Capítulo 3: El Último Bastión**

...nuestro héroe a sobrepasado el cinturón de asteroides de Tenith, pero los problemas no han hecho más que empezar. El panel de mandos parpadea, una nueva comunicación entrante informa que infinidad de naves han salido a su encuentro. El comandante Gaunt, mano derecha de Adan, ha preparado una calurosa bienvenida y está dispuesto a cobrar cara su derrota...

## **Capítulo 4: Una Deuda de Sangre**

...antes de la explosión de su nave, el comandante Gaunt envió un mensaje, sus palabras eran claras, "No querrías continuar si supieras lo que yo se, La Fundación te está utilizando y te está ocultado cosas, que nunca deberías saber". Solo existe una forma de descubrir que encierran las misteriosas palabras del comandante, ir a Tenith y acabar con todo esto de una vez...

## **Capítulo 5: Más allá del Deber**

...con el cuerpo de Adan entre sus manos, el valeroso soldado de los Fantasmas de Acero descubre que Adan es su propio hermano, ambos fueron creados en un laboratorio por La Fundación, pero al no mostrar progresos uno de los soldados fue descartado. Adan, en su último aliento, explica como La Fundación está intentando controlar la mente de todos los ciudadanos del mundo y él solo quería poner fin a aquello. Ahora lo comprendía todo, había sido engañado, manipulado y utilizado por La Fundación para conseguir los beneficios de esta, pero ahora que nuestro héroe conoce la verdad solo le queda una alternativa, aniquilar a La Fundación...



## 4.2. Gráficos

Para el videojuego se han desarrollado los siguientes diseños:

### Modelos 3D

- Modelo de la nave. El modelo de la nave será el mismo para la nave del jugador y las naves enemigas.
- Modelo del helicóptero. Modelo del helicóptero sin tener en cuenta la hélice, que se modelará a parte.
- Modelo de la hélice. El modelo de la hélice se crea por separado al del helicóptero para no tener que animarlo usando Blender, sino Ogre.
- Modelo del tanque. El modelo es sencillo ya que los detalles se le pondrán al aplicarle la textura.
- Modelo del platillo volante. Modelo simple con forma de esfera rodeada por un anillo.
- Modelo del meteorito. Es un modelo simple y con forma de esfera pero un poco deformado para simular la piedra.
- Modelo del potenciador de escudo. Modelo simple que simboliza un escudo.
- Modelo del potenciador de vida. Modelo simple de un corazón que simboliza la vida.
- Modelo del potenciador de disparo especial. Modelo simple de una bala que simboliza el disparo especial.
- Modelo del disparo. Es un modelo muy sencillo ya que el disparo es un pequeño rectángulo.
- Modelo de la bomba. Modelo sencillo ya que la bomba es pequeña.
- Modelo del título del juego. El título del juego con relieve y efecto metalizado.
- Modelo de las montañas. El modelo de las montañas deberá ser bastante extenso para que no se note que se repite al pasar el scroll.
- Modelo del enemigo final 1. El modelo del enemigo final es un poco más elaborado ya que es más grande e importante.

- Modelo del enemigo final 2. El enemigo final 2 está modelado en dos partes, por una el anillo del platillo y por otra la esfera central.
- Modelo del enemigo final 3. El modelo del enemigo final es un poco más elaborado ya que es más grande e importante.
- Modelo de la cabina de la nave. La cabina de la nave no es necesario modelarla por completo, solo la parte de delante y por dentro, que es la que captará la cámara.
- Modelo de los cañones de la cabina de la nave. Los cañones de la cabina de la nave deben crearse por separado para que tengan retroceso al disparar.
- Modelo del cráter. Un modelo simple para que se “funda” con el suelo.

### **Sistemas de partículas**

- Partículas del fuego de las naves. Las partículas deben tener un tiempo de vida corto para que parezca que siempre están en el mismo sitio, aunque la nave se mueva.
- Partículas de la lluvia. Debe tener una gran área para que cubra toda la pantalla.
- Partículas del fuego de los cañones de la cabina. Deben ser similares al fuego de la nave.
- Partículas del humo de los enemigos finales. Las partículas de humo tienen que hacerse mas grandes durante el tiempo que dure su vida, que debe ser largo, para que parezca que se desvanece.

### **Texturas**

- Textura de la nave del jugador. La textura de la nave del jugador será similar a la de las naves enemigas, ya que los modelos 3D son los mismos, pero con otro color, para poder diferenciarlas.
- Textura de las naves enemigas. Similar a la nave del jugador pero de color más claro.
- Textura del helicóptero. Textura con el dibujo de las ventanas y el camuflaje.
- Textura del tanque. Textura con el dibujo de las ruedas y el camuflaje.
- Textura del platillo volante. Texturas simulando piezas de metal.

- Textura del meteorito. Textura con aspecto de piedra o tierra.
- Textura de la bomba. Textura muy simple ya que la bomba será muy pequeña y no necesita detalles.
- Textura de las montañas. La textura de las montañas deberá ser igual a la textura del suelo para que se solapen y que no se note demasiado la diferencia.
- Textura de del suelo. Textura similar a la de las montañas y los cráteres.
- Texturas de los enemigos finales. Texturas simulando piezas de metal.
- Texturas de la cabina de la nave. La cabina de la nave tendrá varias texturas. En la primera, la nave aparecerá intacta y a medida que vayan impactando meteoritos contra la nave, habrá que cambiar la textura por otra en la que la cabina esté más deteriorada.
- Textura del cráter. La textura del cráter deberá ser similar a la textura del suelo y de las montañas, para que parezca que está incrustado en el suelo.
- Textura del SkyBox del espacio. El SkyBox estará formado por seis imágenes cuyos límites coincidan unos con otros para que al formar el cubo no se noten las uniones. Deberán ser de un tamaño lo suficientemente grande como para que no se vea borroso.
- Textura del sol. Textura transparente para que se funda con el cielo.
- Textura de las estrellas. Texturas transparentes para que se fundan con el negro del espacio.
- Textura de las nubes. Textura transparente para que se funda con el cielo.
- Textura de los rayos. Texturas transparentes para que se fundan con el cielo y las nubes.
- Textura de la interfaz durante el juego. Imagen que simula el interior de una nave.
- Textura del indicador de tiempo de escudo. Es un pequeño rectángulo azul que simboliza el tiempo restante de escudo.
- Textura del indicador de munición especial. Es una imagen de una bala que simboliza la munición restante.

- Textura de las vidas restantes. Es una imagen de una pequeña nave que simboliza una vida.
- Textura de la mira. Imagen que sirve para poder apuntar en la segunda fase.
- Textura del radar. Deberá tener algo de transparencia para que se pueda ver que hay detrás.
- Textura de la vida de la cabina. Deberá tener algo de transparencia para que se pueda ver que hay detrás.

En la siguiente imagen podemos ver una compilación de texturas del juego:

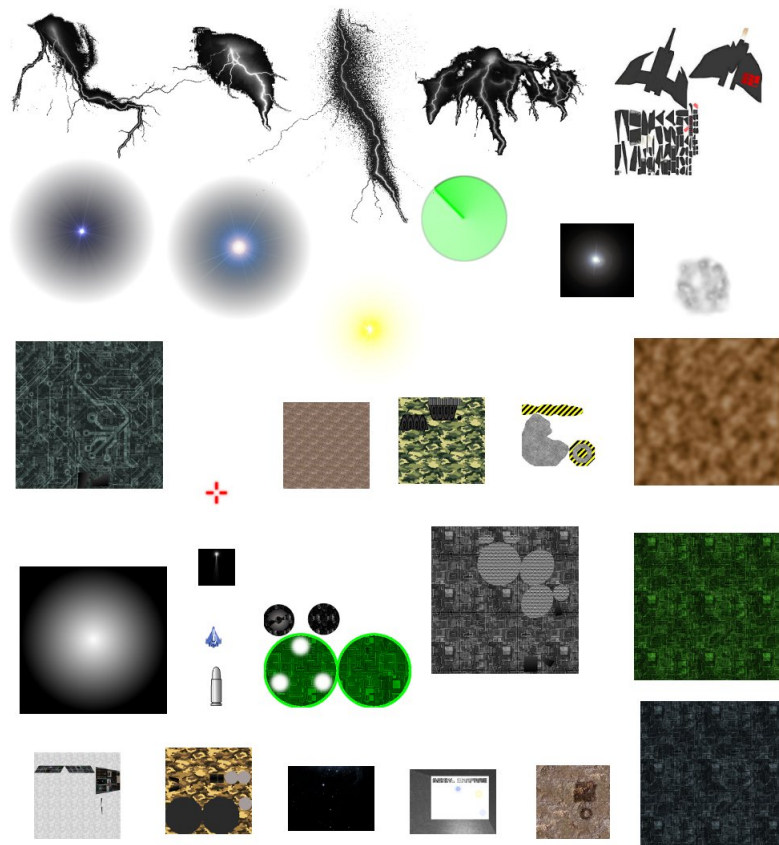


Figura 4.1: Texturas del juego

## 4.3. Sonido

### Música

- Música del menú. Deberá ser una música tranquila.
- Música de la introducción.
- Música de las fases. Será una música bastante animada para darle acción al juego.

### Efectos

- Sonido de disparo.
- Sonido de explosión. Deberá ser un sonido de corta duración, sin eco, de lo contrario al producirse varias explosiones seguidas se solaparía el sonido de unas con otras.
- Sonido de vida.
- Sonido de escudo.
- Sonido de derribado.
- Sonido de las letras de la introducción. Será un sonido simulando al de una máquina de escribir.
- Sonido de munición de arma especial.

## 4.4. Interfaz

### 4.4.1. Menú Principal

Desde el menú principal podemos realizar las siguientes acciones:

- Nueva partida o Reanudar: si no hemos empezado a jugar todavía, podremos hacerlo pulsando sobre “nueva partida”. en el caso de que ya estemos jugando, pero hayamos pausado el juego con la tecla escape, el botón de “nueva partida” se habrá convertido en “reanudar”, y pulsándolo volveremos al juego.
- Seleccionar misión: Pulsando sobre esta opción accederemos a un menú donde podremos seleccionar que fase queremos jugar.

- Opciones: Pulsando aquí entraremos en un menú donde podremos cambiar las opciones que veremos más adelante.
- Records: Pulsando aquí entraremos en el menú de Records.
- Créditos: Pulsando aquí entraremos en el menú de Créditos.
- Salir: Pulsando sobre salir se cierra el juego.



Figura 4.2: Menú principal

### **Seleccionar misión**

En el menú de selección de misión podremos elegir entre una de las cuatro misiones disponibles. Si estamos jugando ya la misión elegida, no empezará de nuevo, sino que reanudaremos la partida por donde lo dejamos. También podremos pulsar sobre volver para regresar al menú principal.



Figura 4.3: Menú de selección de misión

## Opciones

En el menú de opciones podremos crear nuevos usuarios, borrarlos, movernos por los usuarios que ya hayan sido creados, cambiar el volumen de la música o los efectos de sonido (entre 0 % y 100 %), la dificultad del juego (fácil, normal y difícil) o regresar al menú principal pulsando sobre “volver”.

Cuando creamos un nuevo usuario, tendremos que darle un nombre usando las letras del teclado de la A a la Z (podremos borrarlas pulsando la tecla retroceso). El nuevo usuario, por defecto, tendrá la dificultad en fácil, el volumen de la música al 50 % y el volumen de los efectos de sonido al 100 %. Todos los cambios que hagamos en las opciones y de todos los perfiles serán guardados automáticamente y cuando abramos el juego de nuevo, estarán tal y como las dejamos. En el caso de que borremos todos los perfiles, se creará uno automáticamente con el nombre “DEFECTO”, el cual podremos borrar más adelante.



Figura 4.4: Menú de opciones

### Récords

En este menú veremos que jugadores tienen las mayores puntuaciones. Pulsando “volver” regresamos al menú principal. Los récords se actualizarán automáticamente cada vez que completemos el juego, si estamos entre las cinco mejores puntuaciones. En tal caso, veremos que en el menú principal ha aparecido la palabra “NUEVO!” escrita en azul, como podemos observar en la imagen 4.6.



Figura 4.5: Menú de récords





Figura 4.6: Nuevo récord

## Créditos

Aquí veremos quienes han trabajado en el desarrollo del juego. Pulsando “volver” regresamos al menú principal.



Figura 4.7: Menú de créditos

## Diagrama

En la siguiente imagen podemos ver un diagrama del funcionamiento de los menús:

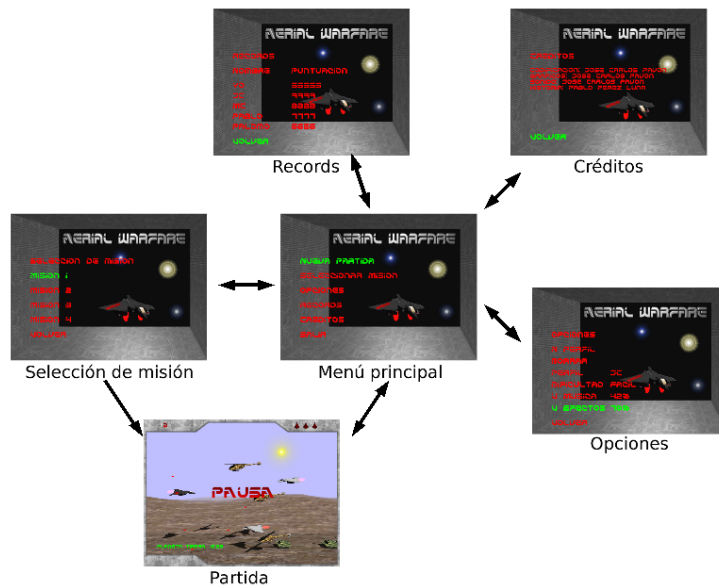


Figura 4.8: Diagrama de menús

### 4.4.2. Elementos de la interfaz

- Puntuación: En la esquina superior izquierda podemos ver la puntuación que tenemos acumulada.



Figura 4.9: Puntuación actual

- Vidas: En la esquina superior derecha podemos ver el número de vidas del que disponemos. Empezamos cada fase con tres vidas y podemos acumular hasta un máximo de cinco. En el caso de encontrarnos en la segunda fase, la vida se nos mostrará en la barra de la imagen 4.11.



Figura 4.10: Número de vidas



Figura 4.11: Barra de vida

- Escudo: Cuando consigamos un potenciador de escudo, en la esquina inferior izquierda podremos ver una barra azul que nos indicara el tiempo de escudo restante.
- Munición: Cuando consigamos un potenciador de disparo, en la esquina inferior izquierda, junto al indicador de escudo, podremos ver la munición que nos queda de disparo especial.



Figura 4.12: Tiempo de escudo y munición

- Texto de localización: Al inicio de cada fase se mostrará, por un breve periodo de tiempo, un texto con la localización de la fase y la hora.



Figura 4.13: Texto de localización

## 4.5. Fase

El juego consta de cuatro fases o misiones, cada una de ellas con sus propios enemigos, escenario, tipos de disparo, o incluso un modo distinto de manejar la nave, como es el caso de la segunda fase, que podemos apreciar en la imagen 4.16. Cada una de las fases empieza con una introducción, que nos irá mostrando la historia del juego. Al terminar la introducción el jugador tendrá que hacerse cargo de la nave. La última fase, además de la introducción inicial, muestra el final de la historia al terminar. En la imagen 4.14 podemos ver una captura de una de las introducciones.



Figura 4.14: Introducción de la misión 2

Una vez terminada la introducción, empieza la acción de la misión. El jugador toma el control de la nave, cuyo funcionamiento detallaremos más adelante,

y tendrá que eliminar a los enemigos hasta que se llegue a un determinado tiempo (este tiempo no será visible para el jugador), entonces aparecerá el enemigo final (esto no ocurrirá en la fase número dos) y deberá acabar con él. Una vez hecho esto, aparecerán las estadísticas de la misión (número de enemigos derribados, porcentaje de acierto, etc). Cuando pulsemos aceptar, avanzaremos hasta la siguiente misión y dará comienzo la introducción de esta.

Si el jugador pierde todas sus vidas antes de terminar la misión se le dará la opción de reiniciar la partida y seguir por donde iba pero, por contra, perderá toda la puntuación acumulada hasta entonces y continuará con cero puntos. A continuación podemos ver una captura de cada una de las fases:



Figura 4.15: Misión 1



Figura 4.16: Misión 2



Figura 4.17: Misión 3



Figura 4.18: Misión 4

## 4.6. Elementos del juego

### Nave del jugador

La nave del jugador aparecerá inicialmente a la izquierda de la pantalla. Se podrá mover por toda la pantalla sin salirse de los límites. Tendrá tres tipos de disparos, disparo principal, secundario y especial.



Figura 4.19: Nave jugador

### Nave enemiga

Las naves enemigas estarán disponibles en las fases uno, tres y cuatro. Aparecerán por la derecha de la pantalla y avanzarán hacia la izquierda, mientras disparan. Si salen de la pantalla por la izquierda o son derribadas, reaparecerán de nuevo.



Figura 4.20: Nave enemiga

### Helicóptero

Los helicópteros, al igual que las naves enemigas, aparecerán por la izquierda de la pantalla y avanzaran a la vez que disparan. Cuando lleguen al borde izquierdo de la pantalla darán media vuelta y volverán a la derecha, cuando lleguen a la derecha darán otra media vuelta e irán de nuevo a la izquierda, pero esta vez continuarán hasta desaparecer de la pantalla. Si salen de la pantalla por la izquierda o son derribadas, reaparecerán de nuevo.



Figura 4.21: Helicóptero

### Platillo Volante

El comportamiento del platillo volante es similar al de las naves enemigas, pero en vez de tener un movimiento rectilíneo, este será senoidal.



Figura 4.22: Platillo volante

### Meteorito

Los meteoritos aparecerán por arriba y la derecha de la pantalla, y se dirigirán hacia la izquierda y abajo. Cuando sean destruidos, se fragmentarán en dos o tres partes, que también habrá que destruir.



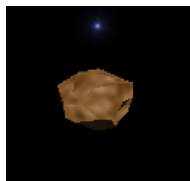


Figura 4.23: Meteorito

## Tanque

Los tanques aparecerán por la derecha de la pantalla y desaparecerán por la izquierda. dispararan hacia arriba y la izquierda. La única forma de destruirlos será mediante el uso de bombas. Si salen de la pantalla por la izquierda o son derribadas, reaparecerán de nuevo.



Figura 4.24: Tanque

## Enemigos finales

Habrán tres enemigos finales (en las fases uno, tres y cuatro). Estos enemigos aparecerán al acabarse el tiempo de cada fase y habrá que eliminarlos para poder continuar. Si ya ha aparecido el enemigo final, los únicos enemigos que seguirán apareciendo serán los tanques y los meteoritos. Además, los helicópteros no darán la vuelta hacia la derecha, sino que seguirán hacia adelante.



Figura 4.25: Enemigo final fase 1



Figura 4.26: Enemigo final fase 3



Figura 4.27: Enemigo final fase 4

## Disparo

En el juego habrá cuatro tipos de disparo: el disparo normal, del que dispondrán tanto la nave del jugador como los enemigos; el disparo especial, que será exclusivo de nuestra nave y consistirá en un disparo triple; el disparo secundario

de la fase tres, y por último un disparo teledirigido (el disparo del enemigo final de la fase cuatro).

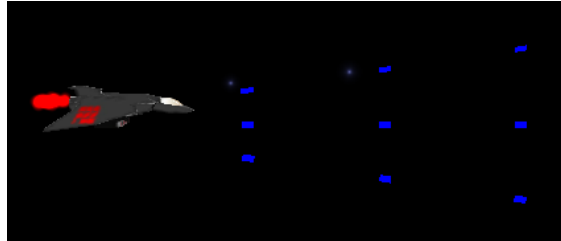


Figura 4.28: Disparo especial

### **Bomba**

La bomba únicamente la podrá utilizar la nave del jugador, y solo en las fases uno y cuatro. Es el único medio destruir los tanques, y ante enemigos finales tendrá un poder de destrucción mayor. El motivo por el que las bombas no están disponibles en las fases dos y tres, es que dichas fases se desarrollan en el espacio, donde no hay gravedad y las bombas no podrían caer.



Figura 4.29: Bomba

### **Montañas y suelo**

En las fases en las que sean necesarias las montañas y el suelo habrá dos objetos de cada clase, que se moverán hacia la izquierda para crear la ilusión de movimiento (scroll horizontal). Cuando uno de los objetos salga de la visión del jugador por la izquierda de la pantalla, se colocara a la derecha del otro objeto. De esta forma se irán turnando para que no se note el cambio.

## Potenciadores

De vez en cuando, cuando se destruya un enemigo (excepto tanques y fragmentos de meteorito), obtendremos un ítem que potenciará la nave del jugador. Para recogerlos habrá que pasar con la nave por encima de ellos. Los potenciadores son los siguientes:

- Vida: Cuando se consiga un potenciador de vida, una nueva vida será añadida al total disponible, siempre que no se supere el límite de cinco vidas.



Figura 4.30: Potenciador de vida

- Escudo: Cuando se consiga el escudo, una esfera protectora rodeará nuestra nave del jugador, haciéndola inmune a los ataques enemigos. El escudo durará hasta que algún enemigo o proyectil impacte contra la nave o hasta que se acabe el tiempo.



Figura 4.31: Potenciador de escudo

- Disparo especial: Con el potenciador de disparo, se obtendrá un arma especial, la cual solo se podrá utilizar 20 veces. Para reponer la munición habrá que conseguir otro potenciador de disparo.

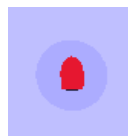


Figura 4.32: Potenciador de disparo

## 4.7. Casos de uso

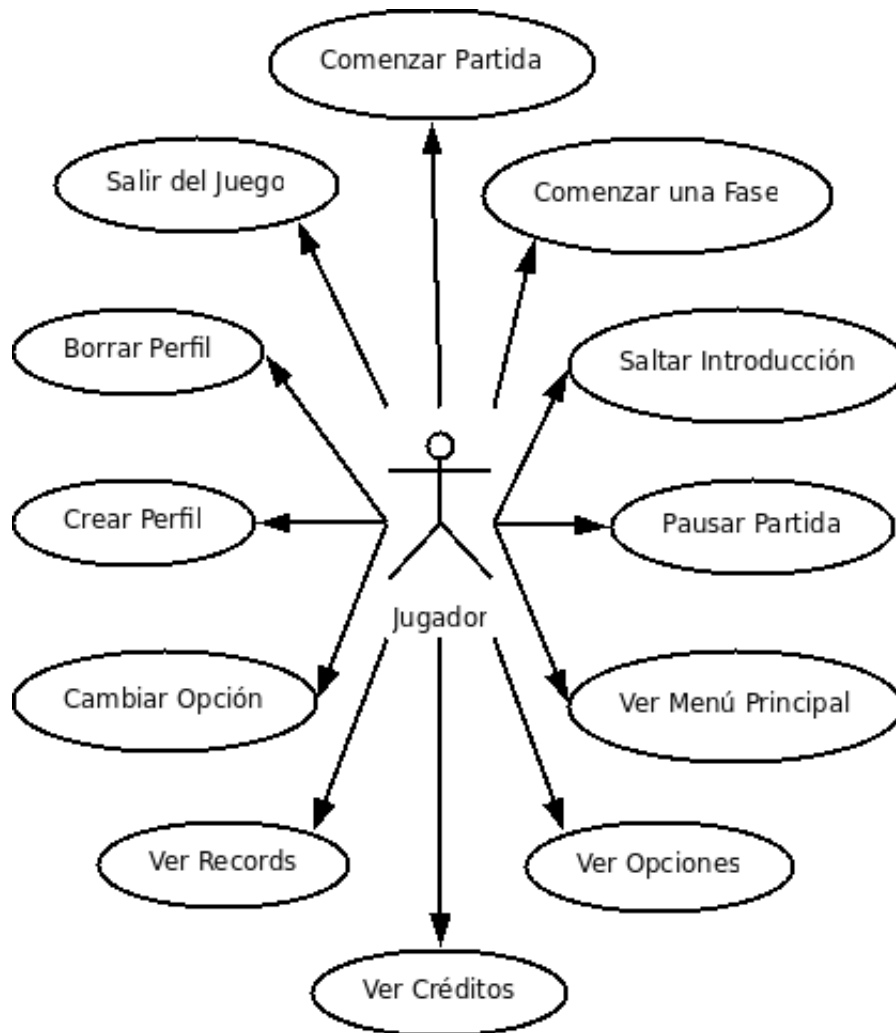


Figura 4.33: Diagrama de casos de uso

### ■ Comenzar partida

#### Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción "Nueva Partida".
3. Se inicia la partida.

- Comenzar una fase

  - Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Seleccionar Misión”.
3. Se muestra el menú de selección de misión.
4. El jugador selecciona la misión que desea jugar.
5. Se inicia la partida.

  - Escenario Alternativo

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Seleccionar Misión”.
3. Se muestra el menú de selección de misión.
4. El jugador selecciona la opción “Volver”.
5. Se muestra el menú principal.

- Saltar introducción

  - Escenario Principal

1. Se muestra la introducción de alguna de las misiones.
2. El jugador pulsa el botón de saltar introducción.
3. Se sale de la introducción y da comienzo la misión.

  - Escenario Alternativo

1. Se muestra la introducción de alguna de las misiones.
2. El jugador no pulsa el botón de saltar introducción.
3. Termina la introducción por si sola y da comienzo la misión.

- Pausar partida

  - Escenario Principal

1. El jugador pulsa el botón de pausa.
2. Se pausa el juego y se notifica al jugador.
3. El jugador vuelve a pulsar el botón de pausa cuando desee reanudar el juego.
4. Se reanuda el juego.

- Ver menú principal

  - Escenario Principal

1. Se muestra el desarrollo de una misión.
2. El jugador pulsa la tecla “Ir al menú”.
3. Se muestra el menú principal.

  - Escenario Alternativo

1. Se muestra cualquiera de los submenús.
2. El jugador selecciona la opción “Volver”.
3. Se muestra el menú principal.

- Ver opciones

  - Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Opciones”.
3. Se muestra el menú de opciones.
4. El jugador selecciona la opción “Volver”.
5. Se muestra el menú principal.

- Ver créditos

  - Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Créditos”.
3. Se muestra el menú de créditos.
4. El jugador selecciona la opción “Volver”.
5. Se muestra el menú principal.

- Ver récords

  - Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Records”.
3. Se muestra el menú de récords.

4. El jugador selecciona la opción “Volver”.
5. Se muestra el menú principal.

■ Cambiar Opción

Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Opciones”.
3. Se muestra el menú de opciones.
4. El jugador selecciona la opción que desea cambiar.
5. El jugador cambia la opción.
6. El jugador selecciona la opción “Volver”.
7. Se muestra el menú principal.

Escenario Alternativo

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Opciones”.
3. Se muestra el menú de opciones.
4. El jugador selecciona la opción “Volver”.
5. Se muestra el menú principal.

■ Crear Perfil

Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Opciones”.
3. Se muestra el menú de opciones.
4. El jugador selecciona la opción “N Perfil”.
5. El jugador escribe el nombre del nuevo perfil.
6. Se crea un perfil con dicho nombre.
7. Se selecciona el perfil recién creado.
8. El jugador selecciona la opción “Volver”.
9. Se muestra el menú principal.

Escenario Alternativo



1. Se muestra el menú principal.
2. El jugador selecciona la opción “Opciones”.
3. Se muestra el menú de opciones.
4. El jugador selecciona la opción “Volver”.
5. Se muestra el menú principal.

- **Borrar Perfil**

  - Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Opciones”.
3. Se muestra el menú de opciones.
4. El jugador selecciona la opción “Borrar”.
5. Se elimina el perfil seleccionado.
6. Se selecciona el primer perfil disponible.
7. El jugador selecciona la opción “Volver”.
8. Se muestra el menú principal.

- **Salir del juego**

  - Escenario Principal

1. Se muestra el menú principal.
2. El jugador selecciona la opción “Salir”.
3. Se sale del juego.

## **4.8. Diagrama de clases conceptuales**

A continuación se muestra un diagrama de clases simplificado, debido a la complejidad por el alto número de clases utilizadas en el juego:

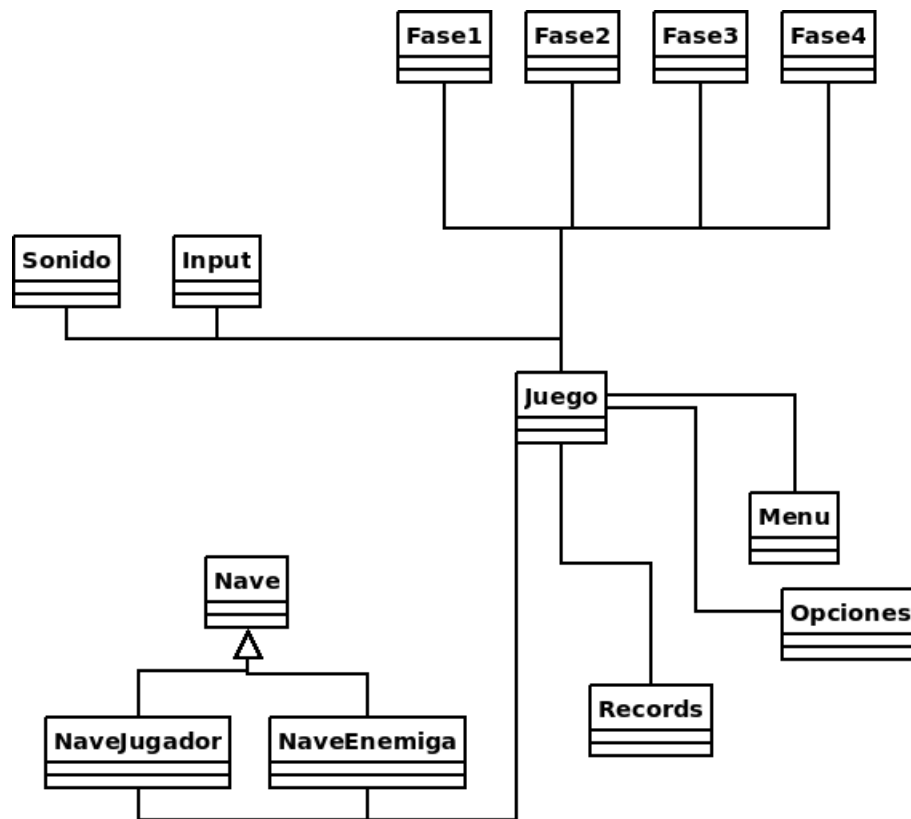


Figura 4.34: Diagrama de clases conceptuales

## 4.9. Contrato de las operaciones

### 4.9.1. Clase Sonido

La clase Sonido se utiliza para reproducir tanto la música como los efectos de sonido.

#### Documentación del constructor y destructor

##### Sonido::Sonido ()

Inicializa SDL, SDL\_mixer, selecciona el número de canales y carga los ficheros de los efectos de sonido.

**Sonido::~~Sonido ()**

Para la música, cierra el audio y libera la música y los efectos.

**Documentación de las funciones miembro****void Sonido::CambiarVolumenEfectos (int *n*)**

Cambia el volumen de los efectos a "*n*" (un número entre 0 y 128).

**void Sonido::CambiarVolumenMusica (int *n*)**

Cambia el volumen de la música a "*n*" (un número entre 0 y 128).

**void Sonido::PlayArma ()**

Reproduce un sonido cuando recibimos un arma especial.

**void Sonido::PlayClick ()**

Reproduce un sonido cuando se escriben las letras de la localización.

**void Sonido::PlayDerribado ()**

Reproduce un sonido cuando nos matan.

**void Sonido::PlayDisparo ()**

Reproduce el sonido del disparo.

**void Sonido::PlayEscudo ()**

Reproduce un sonido cuando recibimos un escudo.

**void Sonido::PlayExplosion ()**

Reproduce el sonido de las explosiones.

**void Sonido::PlayMusicaFase ()**

Carga el fichero de música de la fase y lo reproduce indefinidamente.

#### **void Sonido::PlayMusicaMenu ()**

Carga el fichero de música del menú y lo reproduce indefinidamente.

#### **void Sonido::PlayVida ()**

Reproduce un sonido cuando recibimos una nueva vida.

#### **void Sonido::StopMusica (int *n* = 500)**

Para la reproducción de la música con un "FadeOut" de 1 segundo.

### **4.9.2. Clase Fase1**

#### **Documentación del constructor y destructor**

##### **Fase1::Fase1 (Juego \* *juego*)**

Crea las luces, cámaras, etc necesarias y los enemigos de la fase. También crea la introducción.

##### **Fase1::~~Fase1 ()**

Borra las luces, camaras, etc y los enemigos de la fase. También borra el enemigo final, si existe.

### **4.9.3. Clase Fase2**

#### **Documentación del constructor y destructor**

##### **Fase2::Fase2 (Juego \* *juego*)**

Crea las luces, camaras, etc necesarias y los enemigos de la fase. También crea la introducción.

##### **Fase2::~~Fase2 ()**

Borra las luces, camaras, etc y los enemigos de la fase. También borra el enemigo final, si existe.

#### **4.9.4. Clase Fase3**

##### **Documentación del constructor y destructor**

###### **Fase3::Fase3 (Juego \* *juego*)**

Crea las luces, camaras, etc necesarias y los enemigos de la fase. También crea la introducción.

###### **Fase3::~~Fase3 ()**

Borra las luces, camaras, etc y los enemigos de la fase. También borra el enemigo final, si existe.

#### **4.9.5. Clase Fase4**

##### **Documentación del constructor y destructor**

###### **Fase4::Fase4 (Juego \* *juego*)**

Crea las luces, camaras, etc necesarias y los enemigos de la fase. También crea la introducción.

###### **Fase4::~~Fase4 ()**

Borra las luces, camaras, etc y los enemigos de la fase. También borra el enemigo final, si existe.

#### **4.9.6. Clase Opciones**

La clase Opciones se utiliza para cargar y guardar el fichero de opciones, para crear y borrar perfiles y para cambiar sus opciones.

##### **Documentación del constructor y destructor**

###### **Opciones::Opciones ()**

Abre el fichero opciones.xml en modo lectura. Si todo ha ido bien carga el árbol xml y lo cierra.

###### **Opciones::~~Opciones ()**

Abre el fichero opciones.xml en modo escritura. Si todo ha ido bien salva el árbol xml en dicho fichero y lo cierra. Al final libera los recursos del árbol.

## **Documentación de las funciones miembro**

### **void Opciones::Actualizar (string *dificultad*, string *vefectos*, string *vmusica*)**

Actualiza las opciones del perfil actual (dificultad, volumen de la música y volumen de los efectos).

### **void Opciones::BorrarPerfil (string *nombre*)**

Borra el perfil actual. Si era el único perfil disponible, se crea otro con nombre "defecto".

### **int Opciones::getDificultad ()**

Devuelve la dificultad del perfil actual (es un número del 0 al 2).

### **string Opciones::getPerfil ()**

Devuelve el nombre del perfil actual.

### **int Opciones::getVEfectos ()**

Devuelve el volumen de los efectos de sonido.

### **int Opciones::getVMusica ()**

Devuelve el volumen de la música.

### **bool Opciones::NuevoPerfil (string *nombre*)**

Crea un nuevo perfil de nombre "nombre". Crea un nuevo perfil de nombre "nombre" (si no existe ya) y le asigna las opciones por defecto.

### **void Opciones::PerfilAnterior ()**

Selecciona el perfil anterior si el perfil actual no es el primero.

### **void Opciones::PerfilSiguiente ()**

Selecciona el perfil siguiente si el perfil actual no es el último.

### **void Opciones::setNodo (string *nombre*)**

Selecciona el perfil con nombre "nombre" (si existe) como el actual.

### 4.9.7. Clase Records

La clase Records se utiliza para cargar y guardar el fichero de récords y para crear nuevos récords.

#### Documentación del constructor y destructor

##### **Records::Records ()**

Abre el fichero records.xml en modo lectura. Si todo ha ido bien carga el árbol xml y lo cierra.

##### **Records::~~Records ()**

Abre el fichero records.xml en modo escritura. Si todo ha ido bien salva el árbol xml en dicho fichero y lo cierra. Al final libera los recursos del árbol.

#### Documentación de las funciones miembro

##### **void Records::Actualizar ()**

Actualiza todos los récords del árbol en el overlay.

##### **bool Records::Nuevo (string *nombre*, string *puntuacion*)**

Recibe el nombre y la puntuación de un nuevo récord. Si está entre los cinco primeros se introduce en la posición adecuada.

### 4.9.8. Clase Menu

La clase Menu crea el menú y lo mantiene actualizado.

#### Documentación del constructor y destructor

##### **Menu::Menu (Juego \* *juego*)**

Crea las cámaras, luces, etc para visualizar el menú, lo configura y reproduce la música del menú.

##### **Menu::~~Menu ()**

Destruye el menú.

### Documentación de las funciones miembro

#### **void Menu::Actualizar (double *t*)**

Actualiza el menú dependiendo de dónde nos encontremos y las opciones que seleccionemos.

### 4.9.9. Clase Input

La clase Input nos proporciona control sobre el ratón y el teclado.

### Documentación del constructor y destructor

#### **Input::Input (Ogre::RenderWindow \* *window*)**

Crea el sistema de entrada y el teclado.

#### **Input::~Input ()**

Destruye el teclado, si existe, y el sistema de entrada.

### Documentación de las funciones miembro

#### **void Input::capture ()**

Captura los cambios en el teclado si está inicializado.

#### **OIS::Keyboard\* Input::getKeyboard () [inline]**

Obtiene el teclado.

### 4.9.10. Clase Juego

La clase Juego se encarga de gestionar el estado del juego (pausar, reanudar, reiniciar, iniciar), crea las naves, meteoritos, disparos, luces... La clase Juego también se encarga de actualizar todos los elementos (nodos, camaras, overlays...)

### Documentación del constructor y destructor

#### **Juego::Juego ()**

Inicia Ogre, crea el sonido, las opciones, los récords, el menú...



**Juego::~~Juego ()**

Cierra Ogre, borra el sonido, las opciones, los récords, el mánager de entrada...

**Documentación de las funciones miembro****bool Juego::actualizarNodoHistoria (double *tiempo*)**

Actualiza la introducción desplazando el nodo de la historia.

**void Juego::BorrarBombas ()**

Borra las bombas.

**void Juego::BorrarCrateres ()**

Borra los cráteres.

**void Juego::BorrarDisparos ()**

Borra los disparos.

**void Juego::BorrarEstrellas ()**

Borra las estrellas.

**void Juego::BorrarFragmentosMeteorito ()**

Borra los fragmentos de meteorito.

**void Juego::BorrarHelicopteros ()**

Borra los helicópteros.

**void Juego::BorrarMeteoritos ()**

Borra los meteoritos.

**void Juego::BorrarMeteoritosCabina ()**

Borra los meteoritos de la fase dos.

**void Juego::BorrarNaves ()**

Borra las naves y reinicia el número.

**void Juego::BorrarPlatillos ()**

Borra los platillos.

**void Juego::BorrarPotenciadoresArma ()**

Borra los potenciadores de arma.

**void Juego::BorrarPotenciadoresEscudo ()**

Borra los potenciadores de escudo.

**void Juego::BorrarPotenciadoresVida ()**

Borra los potenciadores de vida.

**void Juego::BorrarTanques ()**

Borra los tanques.

**void Juego::crearGalaxia ()**

Crea la galaxia de las introducciones.

**void Juego::crearHelicopteros (int n)**

Crea "n" helicópteros.

**void Juego::crearIntroduccionFase1 ()**

Crea la introducción de la fase 1.

**void Juego::crearIntroduccionFase2 ()**

Crea la introducción de la fase 2.

**void Juego::crearIntroduccionFase3 ()**

Crea la introducción de la fase 3.

**void Juego::crearIntroduccionFase4 ()**

Crea la introducción de la fase 4.

**void Juego::crearIntroduccionFase5 ()**

Crea la introducción del final del juego.

**void Juego::crearMeteoritos (int *n*)**

Crea el número de meteoritos indicado por *n*.

**void Juego::crearMeteoritosCabina (int *n*)**

Crea el número de meteoritos indicado por *n*.

**void Juego::crearNaves (int *n*)**

Crea las naves. La primera nave es la nave del jugador y las demás son las naves enemigas.

**void Juego::crearPlatillos (int *n*)**

Crea "*n*" platillos.

**void Juego::crearTanques (int *n*)**

Crea el número de tanques indicado por *n*.

**void Juego::incrementarContadorDisparos ()**

Incrementa el contador de disparos.

**void Juego::iniciar ()**

Inicia el juego.

**void Juego::Jugar ()**

Reanuda la fase que ya se esté jugando. Si no se está jugando ninguna fase empieza la primera.

**void Juego::JugarFase1 ()**

Juega la primera fase, desde el principio, si no existe, o la reanuda si ya existe.

**void Juego::JugarFase2 ()**

Juega la segunda fase, desde el principio, si no existe, o la reanuda si ya existe.

**void Juego::JugarFase3 ()**

Juega la tercera fase, desde el principio, si no existe, o la reanuda si ya existe.

**void Juego::JugarFase4 ()**

Juega la cuarta fase, desde el principio, si no existe, o la reanuda si ya existe.

**void Juego::nuevaBomba (Vector3 *posicion*, Vector3 *velocidad*, string *nombreNave*)**

Crea una nueva bomba.

**void Juego::nuevoDisparo (Vector3 *posicion*, Vector3 *velocidad*, string *nombreNave*, double *vida*, Juego \**juego*)**

Crea un nuevo disparo. Si el disparo es del enemigo final de la fase cuatro, este será teledirigido.

**void Juego::pausar ()**

Pausa el juego y muestra el texto "PAUSA" en la pantalla.

**void Juego::reanudar ()**

Reanuda el juego y deja de mostrar el texto "PAUSA" en la pantalla.

**void Juego::reiniciar ()**

Reinicia la puntuación, los enemigos y las vidas.

**void Juego::reiniciarEnemigos ()**

Reinicia los enemigos a la posición inicial, elimina otros objetos como disparos, fragmentos de meteorito, bombas, potenciadores... También reproduce el sonido de derribado.

**void Juego::reiniciarEstadisticas ()**

Reinicia las estadísticas a cero.

**void Juego::salir ()**

Sale del juego.



# Capítulo 5

## Desarrollo

### 5.1. Modelado 3D

A la hora de hacer los modelos 3D, tenemos varias alternativas<sup>1</sup>:

- **3D Studio Max:** Autodesk 3D Studio Max es un programa de creación de gráficos y animación 3D desarrollado por Autodesk Media & Entertainment.  
3d Studio Max es uno de los programas de animación 3D más utilizados. Dispone de una sólida capacidad de edición, una omnipresente arquitectura de plugins y una larga tradición en plataformas Microsoft Windows. 3ds Max es utilizado en mayor medida por los desarrolladores de videojuegos, aunque también en el desarrollo de proyectos de animación como películas o anuncios de televisión, efectos especiales y en arquitectura.
- **Maya:** Maya es un programa informático dedicado al desarrollo de gráficos en 3D, efectos especiales y animación. Maya se caracteriza por su potencia y las posibilidades de expansión y personalización de su interfaz y herramientas'. MEL (Maya Embedded Language) es el código que forma el núcleo de Maya, y gracias al cual se pueden crear scripts y personalizar el paquete. Maya posee numerosas herramientas para modelado, animación, render, simulación de ropa y cabello, dinámicas (simulación de fluidos), etc.
- **Wings 3D:** Wings 3D es un programa de modelado 3D libre.  
Está disponible para la mayoría de plataformas, incluyendo Windows, Linux y Mac  
Está diseñado para modelar y texturizar elementos formado con un número de polígonos menor. Cuando se compara con otros programas de 3D (como

---

<sup>1</sup>Definiciones extraídas de wikipedia.org

puede ser Blender) se observan las diferencias existentes sobre todo en la interfaz gráfica de usuario, la cual aporta una mayor flexibilidad a cambio de ciertas limitaciones en algunas áreas. Por ejemplo, Wings 3D no puede manejar animaciones, sólo trae un renderizador de OpenGL, y muchas opciones pueden llegar a desactivarse si el elemento poligonal es demasiado complejo.

Sin embargo, Wings 3D es de muy fácil manejo y un sistema de iconos muy intuitivo. Aun careciendo de un potente renderizador, Wings 3D puede combinarse con otros programas como POV-Ray o YafRay para realizar imágenes de alta calidad.

- Blender: programa 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.

Finalmente me decanté por utilizar Blender. Para tomar esta decisión primero descarté utilizar 3D Studio Max y Maya, por ser programas propietarios y de pago. Aunque Wings 3D no tenga soporte para animaciones, esto no sería un impedimento, ya que no serán necesarias para el desarrollo del juego, pero aun así, me decidí por utilizar Blender porque considere que existe una mayor y mejor documentación para este y que el programa es más potente (aunque el aprendizaje fuera más complejo). Algunas de las características de Blender son:

- 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 vídeo.



- 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 o YafRay.
- 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.



Figura 5.1: Blender. Extraído de <http://blogs.gamefilia.com>

Vamos a mostrar el funcionamiento básico, en términos generales, de blender, [Gonz06] para ello nos serviremos del modelo del helicóptero. En la imagen 5.2 tenemos el programa blender. Podemos observar que la pantalla está dividida en varias secciones, abajo tenemos una barra con herramientas que nos serán útiles para el desarrollo del modelo y en la parte superior vemos las vistas en 3 dimensiones.

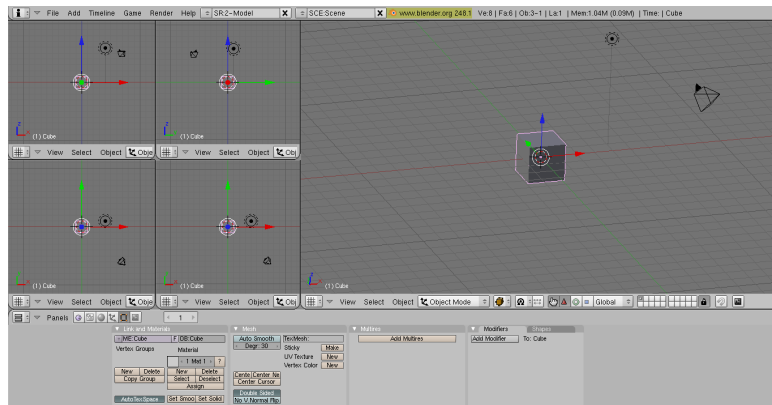


Figura 5.2: Interfaz de Blender

Empezaremos modelando el cubo que vimos en la imagen 5.2 con operaciones de escalado, rotación, extrusión y traslación sobre las caras, vértices y aristas del cubo hasta ir acercándonos cada vez mas al modelo deseado, como podemos ver en las imágenes 5.3 y 5.4

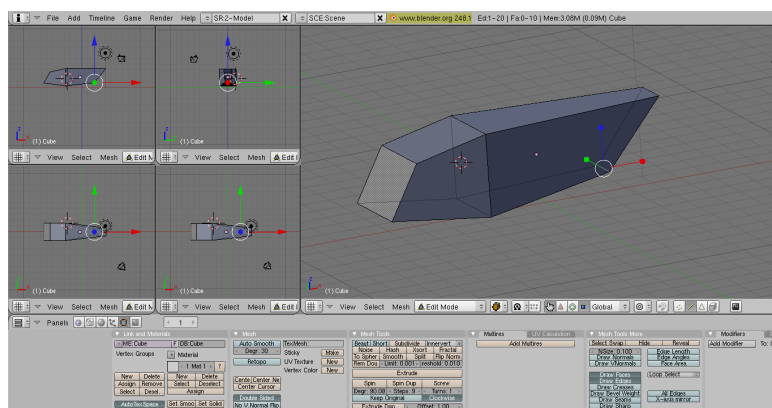


Figura 5.3: Extrusión, escalado, rotación...

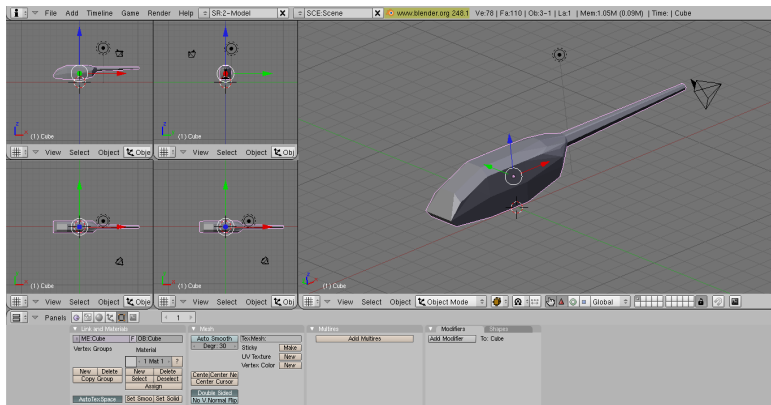


Figura 5.4: Extrusión, escalado, rotación...

Para terminar de modelar el helicóptero se añaden otros elementos, como cilindros, para completar los tubos de las patas o la hélice de la cola, como vemos en la imagen 5.5.

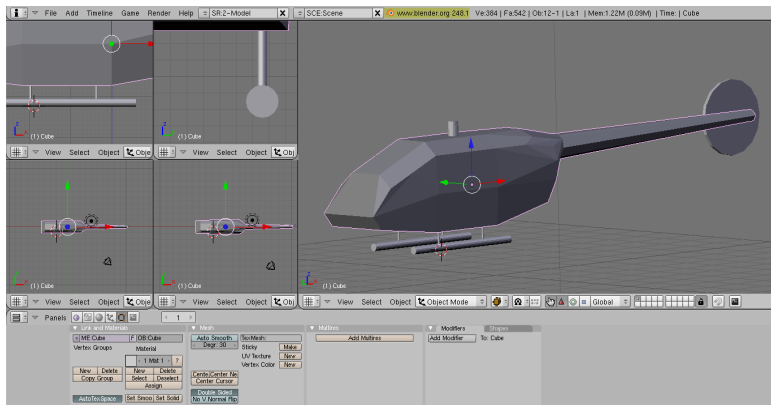


Figura 5.5: Añadiendo los detalles

En la imagen 5.6 tenemos el resultado final.

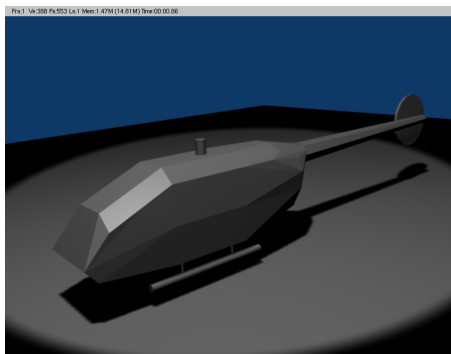


Figura 5.6: Resultado

### 5.1.1. Creación de las texturas

Para la creación de las texturas primero tenemos que elegir el programa con el que vamos a trabajar. Principalmente pensé en estos tres programas<sup>2</sup>:

- Adobe Photoshop: Adobe Photoshop es una aplicación en forma de taller de pintura y fotografía que trabaja sobre un lienzo y que está destinado para la edición, retoque fotográfico y pintura a base de imágenes de mapa de bits o conocidos en Photoshop como gráficos rasterizados, elaborado por la compañía de software Adobe Systems inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.
- Krita: Krita es un programa de edición de imágenes de mapa de bits. Es software libre y multiplataforma. Es parte de la suite KOffice del proyecto KDE.
- GIMP: GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Está englobado en el proyecto GNU y disponible bajo la Licencia pública general de GNU. La primera versión de GIMP se desarrolló inicialmente en sistemas Unix y fue pensada especialmente para GNU/Linux. Existen versiones totalmente funcionales para Windows, para Mac OS X, y se incluye en muchas distribuciones GNU/Linux. También se ha portado a otros sistemas operativos, haciéndolo el programa de manipulación de gráficos disponible en más sistemas operativos. Se le puede considerar como una alternativa firme, potente

---

<sup>2</sup>Definiciones extraídas de wikipedia.org

y rápida a Photoshop para muchos usos, aunque no se ha desarrollado como un clon de él y posee una interfaz bastante diferente.

Adobe Photoshop lo descarté por ser un programa privativo y de pago. Luego, tanto Gimp como Krita son dos buenos programas pero decidí elegir el primero por estar ya familiarizado con su funcionamiento.



Figura 5.7: Gimp. Extraído de commons.wikimedia.org

Una vez elegido el programa, pasemos a ver como se crean las texturas. Como vemos en la imagen 5.6, el helicóptero tiene un color sólido, para arreglar esto, debemos ponerle una textura. Para ello lo primero que tendremos que hacer será “recortar” el modelo y crear costuras, de forma que podamos luego dibujarla sobre una imagen plana. En la imagen 5.8 tenemos las costuras, que han sido realizadas para que puedan verse mejor.

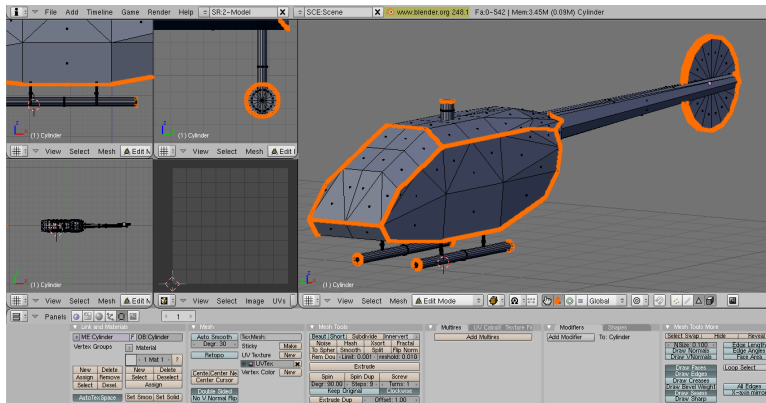


Figura 5.8: Costuras en Blender

Lo siguiente que tendremos que hacer será exportar la imagen de las costuras (imagen 5.9) para poder pintar la textura sobre ella de forma adecuada. Esto lo podremos hacer usando un script que nos proporciona Blender.

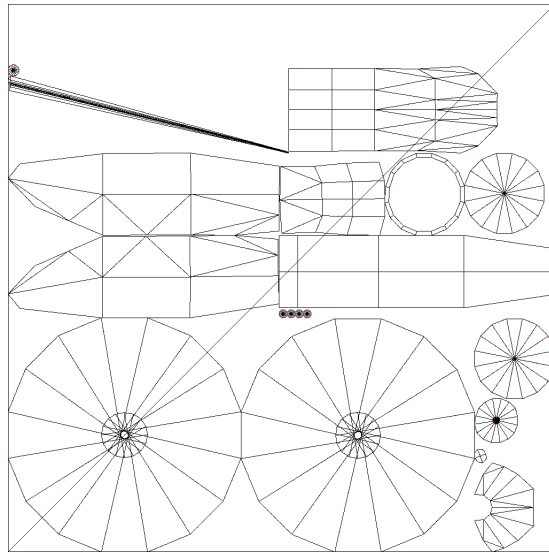


Figura 5.9: Patrón de la textura

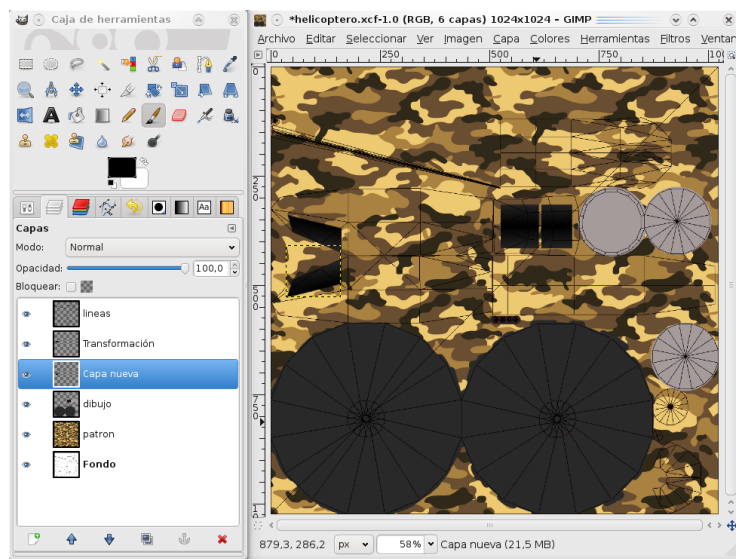


Figura 5.10: Interfaz de GIMP

En la figura 5.11 vemos como ya ha sido dibujada la textura sobre el “patrón” que obtuvimos con Blender.

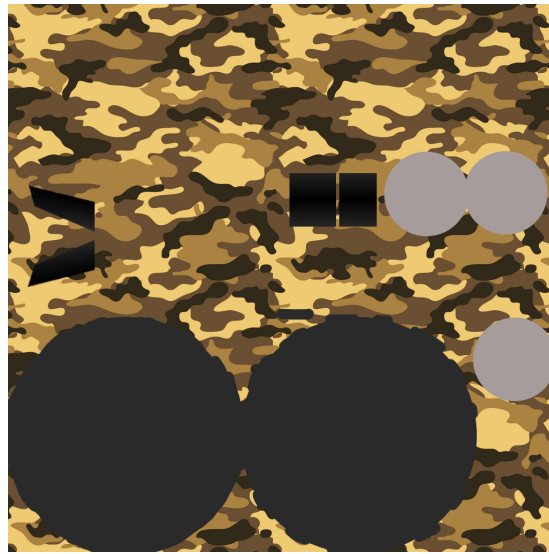


Figura 5.11: Textura terminada

Una vez creada la textura la cargamos en el Blender, el resultado es el que se ve en la imagen 5.12.

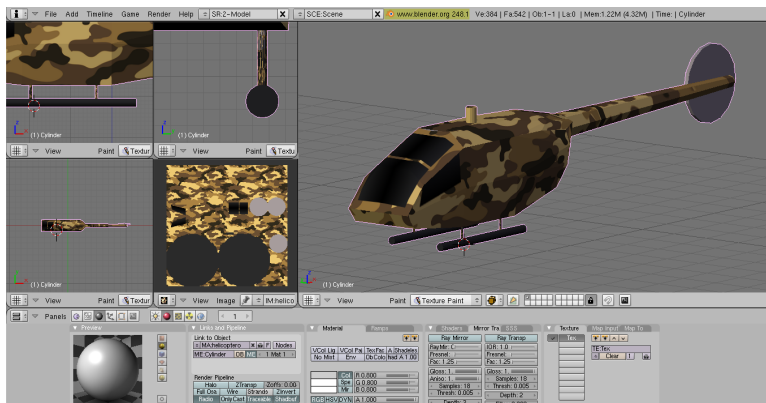


Figura 5.12: Helicóptero con texturas

Y en la imagen 5.13 podemos ver el resultado final de todo el proceso.

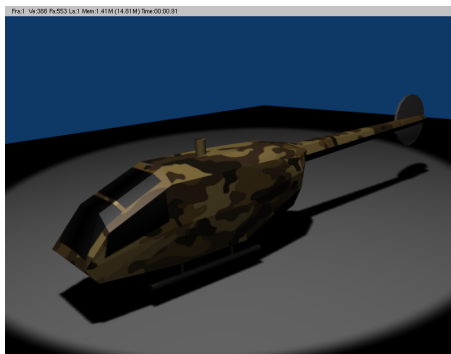


Figura 5.13: Resultado final

Como habrán observado, el helicóptero carece de hélice principal, esto se debe a que dicha hélice estará en movimiento durante el juego, por lo tanto, para añadirla al modelo 3D, tendríamos que crear una animación, que es una tarea más compleja, por lo tanto hemos preferido modelar la hélice como un objeto a parte y darle el movimiento a través de las opciones que nos brinda Ogre 3D.

En las siguientes capturas podemos ver otros modelos utilizados en el juego:

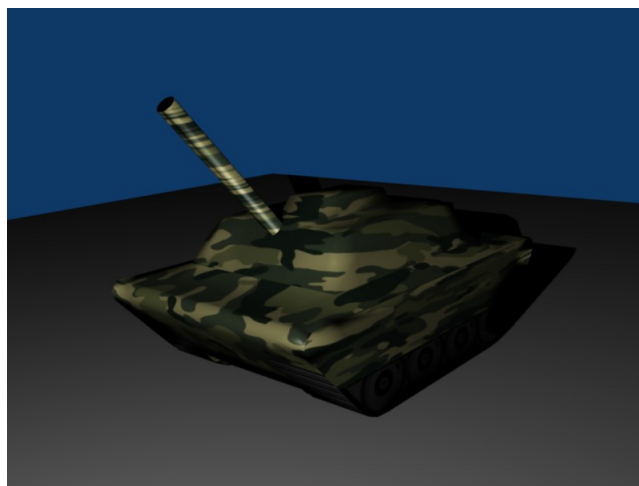


Figura 5.14: Tanque



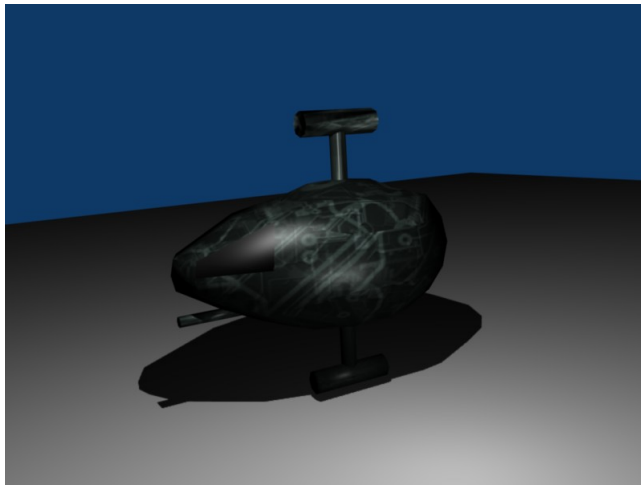


Figura 5.15: Enemigo Final 1

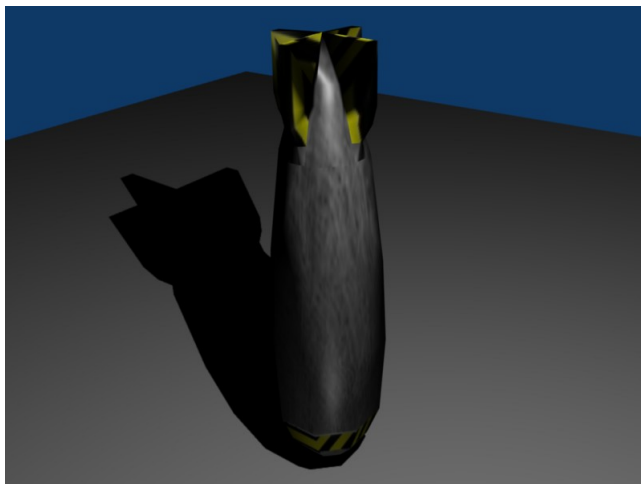


Figura 5.16: Bomba

### 5.1.2. Exportación de los modelos

Una vez que finalicemos todo el proceso anterior, tenemos que exportar el modelo de blender a un archivo que Ogre pueda utilizar. Para ello utilizaremos “Ogre mesh exporter”, un script para Blender proporcionado por Ogre [[OgreWi](#)]. Podemos verlo en la captura [5.17](#).

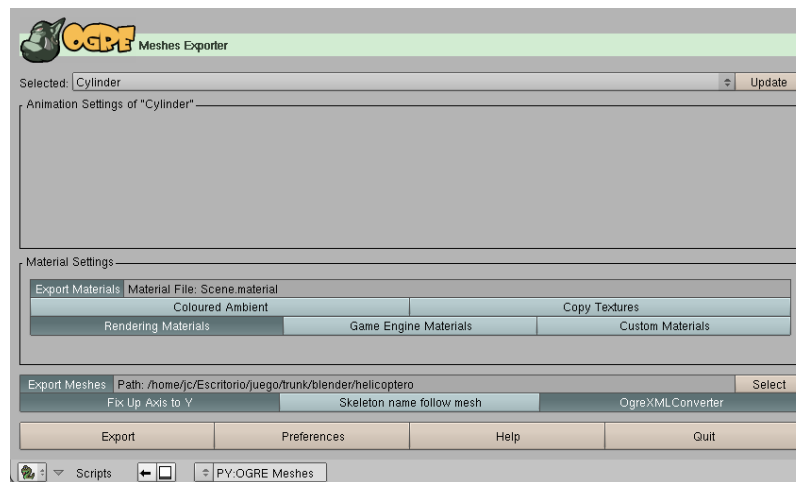


Figura 5.17: Ogre mesh exporter

Algunas de las opciones interesantes son “Fix Up Axis to Y” que nos será útil para cambiar los ejes de coordenadas, ya que Blender utiliza el eje Z como el vertical y el Y para la profundidad, al contrario que Ogre. Otra opción interesante es OgreXMLConverter, lo explicaremos a continuación.

Ogre mesh exporter no nos proporciona directamente un archivo `.mesh`, sino un archivo `.xml`, el cual, luego, tendremos que convertir en el `.mesh`. Esto lo haremos con el programa OgreXMLConverter. Para que Blender lo haga por nosotros, tendremos que marcar la opción mencionada anteriormente y en las preferencias (figura 5.18) indicar la ruta a dicho programa, en nuestro caso `/usr/bin/OgreXMLConverter`.

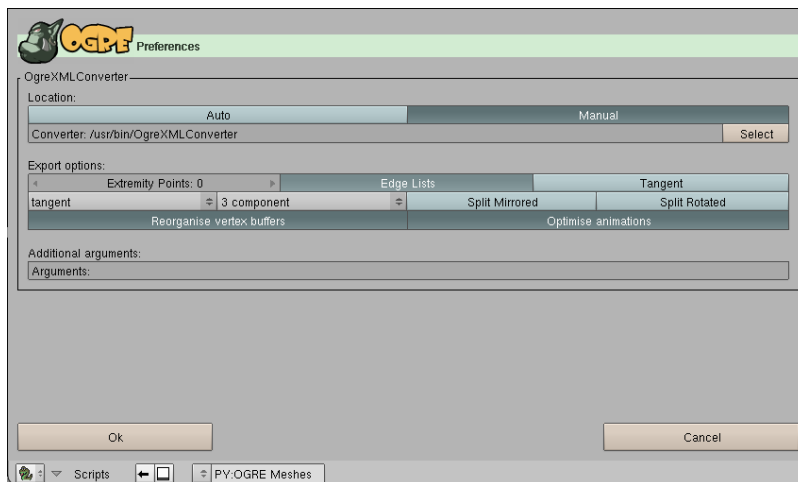


Figura 5.18: Preferencias de Ogre mesh exporter

### 5.1.3. Problemas en la exportación de modelos

Ogre mesh exporter nos proporciona, aparte del archivo `.xml`, un script de material de Ogre. Dicho script tiene la siguiente forma:

```
material Cubo
{
  receive_shadows on
  technique
  {
    pass
    {
      ambient 0.500000 0.500000 0.500000 1.000000
      diffuse 0.640000 0.640000 0.640000 1.000000
      specular 0.500000 0.500000 0.500000 1.000000 12.500000
      emissive 0.000000 0.000000 0.000000 1.000000
    }
  }
}
```

Vemos que el último número del parámetro `specular` es `12.500000`, pues bien, para que los modelos 3D, una vez en el escenario de Ogre, reciban las sombras adecuadamente, tendremos que cambiarlo a `0.0`.

## 5.2. Sonido

Algunos de los sonidos han sido extraídos de otros juegos libres para GNU/Linux.

Para editar los distintos sonidos del juego (amplificar, cortar, etc) nos hemos servido del programa Audacity.

Audacity es un programa multiplataforma de grabación y edición de sonidos fácil de usar, de libre uso y de código abierto distribuido bajo licencia GPL.

Es el editor de sonido más difundido en los sistemas Linux.

Algunas de sus características son:

- Grabación de audio en tiempo real.
- Edición archivos de audio tipo Ogg Vorbis, MP3, WAV, AIFF, AU y LOF.
- Conversión entre formatos de audio tipo.
- Importación de archivos de formato MIDI y RAW.
- Edición de pistas múltiples.



Figura 5.19: Audacity. Extraído de [audacity.sourceforge.net](http://audacity.sourceforge.net)

En la captura 5.19 podemos ver el programa.

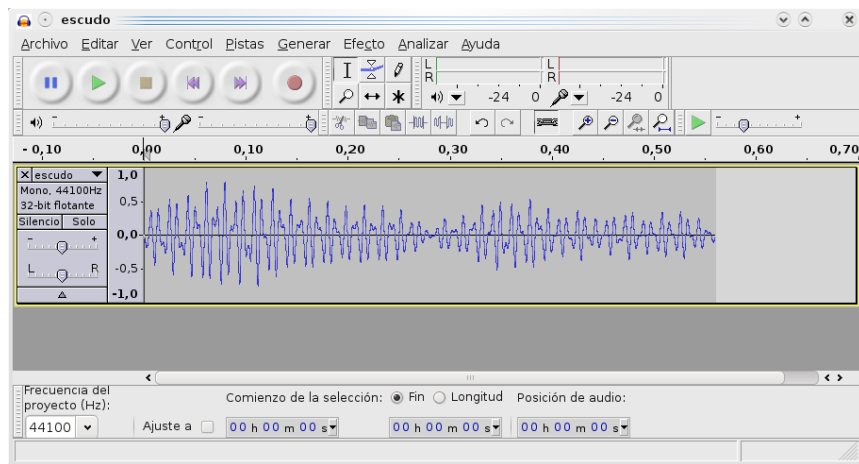


Figura 5.20: Editando audio con Audacity

### 5.2.1. Música

La música también se ha obtenido de juegos libres de GNU/Linux. Para convertir algunas de las canciones de *mp3* a *ogg* se ha utilizado el programa SoundKonverter (véase la imagen 5.21).

SoundKonverter es una interfaz gráfica para varios programas de codificación y decodificación de sonido. Es extensible a través de plugins y está diseñado para KDE, por lo que posee capacidades de integración con éste.

Sus aspectos más destacables son la extensa variedad de formatos que admite y su sencilla interfaz a través de la cual pueden realizarse multitud de operaciones a golpe de ratón.

Está disponible en inglés, alemán, francés, español, ruso, polaco, brasileño y portugués.

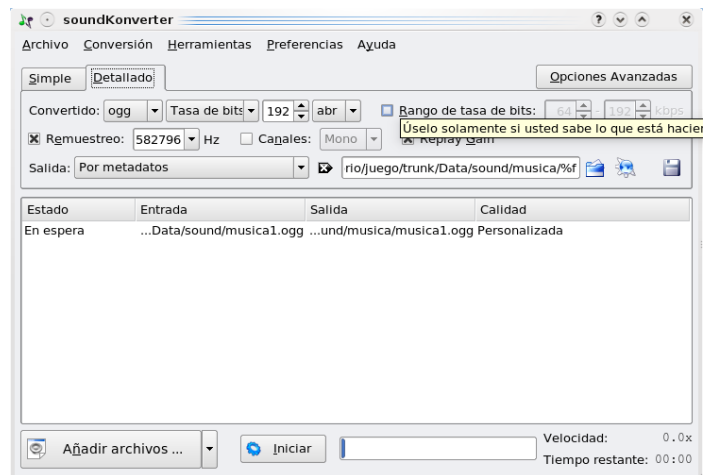


Figura 5.21: SoundKonverter

## 5.3. Codificación

### 5.3.1. Motor Gráfico

Como motor gráfico disponemos de<sup>3</sup>:

**OpenGL** OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compite con Direct3D en plataformas Microsoft Windows.

**OGRE 3D** OGRE 3D (acrónimo del inglés Object-Oriented Graphics Rendering Engine) es un motor de renderizado 3D multiplataforma orientado a escenas, escrito en el lenguaje de programación C++.

Sus bibliotecas evitan la dificultad de la utilización de capas inferiores de librerías gráficas como OpenGL y Direct3D, y además, proveen una interfaz basada en objetos del mundo y otras clases de alto nivel.

<sup>3</sup>Definiciones extraídas de wikipedia.org

El motor es software libre, licenciado bajo LGPL y con una comunidad muy activa. Incluso ha sido utilizado en algunos videojuegos comerciales, como por ejemplo Ankh y Earth Eternal.

Elegí Ogre 3D por ser un motor gráfico de más alto nivel que OpenGL, facilitando mucho la tarea de crear el videojuego.



Figura 5.22: Ogre 3D. Extraído de wikipedia.org

### 5.3.2. Entrada y Salida

Para la entrada y salida, principalmente tenía dos alternativas, OIS y SDL<sup>4</sup>:

**OIS** OIS (Object-Oriented Input System) es una librería para construir interfaces humano-ordenador con dispositivos de entrada como teclados, ratones o dispositivos para juegos como pueden ser joysticks o Wii remote. OIS está diseñado de forma que los desarrolladores puedan usar la entrada de estos dispositivos fácilmente desde sus aplicaciones.

**SDL** Simple DirectMedia Layer (SDL) es un conjunto de bibliotecas desarrolladas con el lenguaje C que proporcionan funciones básicas para realizar operaciones de dibujo 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes.

Pese a estar programado en C, tiene wrappers a otros lenguajes de programación como C++, Ada, C#, Basic, Erlang, Lua, Java, Python, etc. También proporciona herramientas para el desarrollo de videojuegos y aplicaciones multimedia. Una de sus grandes virtudes es el tratarse de una biblioteca multiplataforma, soportando oficialmente los sistemas windows, linux, MacOS y QNX, además de otras arquitecturas/sistemas como Dreamcast, GP32, GP2X... De ahí le vienen las siglas Simple Directmedia Layer que más o menos alude a capa de abstracción multimedia. Desarrolladas inicialmente por Sam Lantinga, desarrollador de videojuegos para la plataforma linux. La biblioteca se distribuye bajo la licencia LGPL.

---

<sup>4</sup>Definiciones extraídas de wikipedia.org

Finalmente, aunque ya conocía la librería SDL, me decidí por OIS ya que está perfectamente integrada con Ogre y bastan unas pocas líneas de código bastan para hacerlo funcionar.

### 5.3.3. Sonido

Para el sonido he optado por utilizar SDL por estar ya habituado a su uso. Aunque no sea una librería muy potente, con las funcionalidades que nos brinda tenemos de sobra para nuestro propósito. Más concretamente se ha utilizado la librería SDL Mixer.

SDL Mixer extiende las capacidades de SDL para la gestión y uso de sonido y música en aplicaciones y juegos. Soporta formatos de sonido como Wave, MP3 y OGG, y formatos de música como MOD, S3M, IT, y XM.



Figura 5.23: SDL. Extraído de libsdl.org

### 5.3.4. Física

Para la física del juego no se ha necesitado ningún motor adicional ya que la mayoría de movimientos son muy simples, movimientos rectilíneos uniformemente acelerados. Lo único que había que hacer era asignarles una dirección y velocidad.

Para el cálculo de colisiones en las fases que se desarrollan en dos dimensiones se comprueba si las coordenadas X e Y de cada uno de los objetos que se quieren comprobar coinciden, teniendo en cuenta las dimensiones de los objetos. En el caso de las fases que se desarrollan en tres dimensiones el procedimiento es parecido al anterior, se comprueba la distancia entre ambos objetos, y si es menor al tamaño de ambos, han colisionado. Podemos observarlo con claridad en la siguiente imagen:



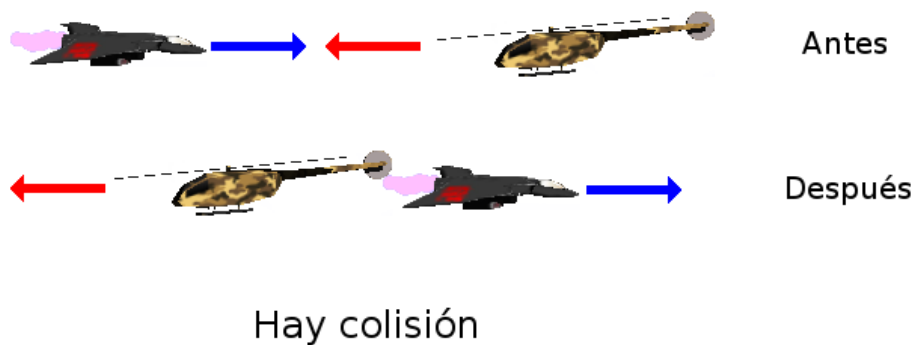


Figura 5.24: Colisiones

### 5.3.5. Interfaz

Para la interfaz se barajaron varias opciones, como CEGUI, MyGUI, QuickGUI o Navi [OgreWi]:

**CEGUI** CEGUI (Crazy Eddie's GUI) es una librería escrita en C++ para desarrollar interfaces gráficas de usuario. Está diseñada especialmente para usarse en videojuegos, pero la librería también se puede usar con cualquier otro propósito.

Pero finalmente decidí implementar la interfaz sin usar ninguno de las opciones citadas anteriormente, ya que el juego no necesita menús muy elaborados y se podían hacer fácilmente utilizando los `overlay` de Ogre.

### 5.3.6. Archivos de configuración

Para los archivos de configuración (opciones y récords) decidí usar la librería *mxml* por ser una librería simple, fácil de utilizar y completamente suficiente para el proyecto [MXMLRE]. Pueden obtener más información en su página web: <http://www.minixml.org/>

## **Algunos problemas con mxml**

Mxml tiene un problema a la hora de navegar entre los nodos del árbol: si entre nodo y nodo, en el archivo de texto, hay algún espacio o salto de línea, interpreta dicho espacio como otro nodo, produciéndose resultados no deseados. La solución es simple, escribir los nodos unos detrás de otros sin espacios.

## **5.4. Pruebas**

Para llevar a cabo las pruebas, como no se pueden probar todos los casos posibles, se han elegido los casos más representativos del resto. Ahora pasaremos a detallar cada una de las pruebas:

### **5.4.1. Pruebas de colisiones**

Las pruebas de colisiones han sido las más extensas, debido a la gran cantidad de objetos distintos que pueden colisionar entre sí. Por ejemplo, para comprobar las colisiones de la nave se han tenido en cuenta las siguientes situaciones: colisión de la nave con otra nave, con helicópteros, con platillos volantes, con meteoritos o sus fragmentos, con las distintas naves de los enemigos finales, con las bombas, disparos (normales o teledirigidos), potenciadores de vida, escudo y arma. Todas estas pruebas se han realizado en distintas condiciones, como pueden ser: con o sin escudo, en presencia o no del enemigo final, en distintas fases, etc. En todos los casos el juego ha funcionado sin problemas. Para el resto de Objetos del juego que pueden colisionar, se ha procedido a realizar pruebas de forma similar.

### **5.4.2. Pruebas de menús**

Para comprobar que los menús funcionan correctamente se ha navegado por cada uno de ellos de todas las formas posibles obteniendo un resultado satisfactorio. Tampoco se han apreciado errores cuando se cambia entre el menú (principal o de selección de misión) y la misión o viceversa.

### **5.4.3. Pruebas de récords**

Para probar los récords se ha comprobado que se lee el fichero “records.xml” correctamente, comprobando también casos como que no haya ningún récord. Cuando se genera una nueva puntuación, el sistema comprueba correctamente en que posición de la lista de récords debería ir, o si no ha entrado en la lista por ser

demasiado baja. Al finalizar el programa, el fichero se guarda en disco de forma correcta para poder usarlo en una ejecución posterior.

#### 5.4.4. Pruebas de opciones

Para probar las opciones se ha comprobado que se lee sin problemas el fichero “opciones.xml”. Todas las opciones asignadas en una ejecución anterior del juego vuelven a ser cargadas en memoria y guardadas de nuevo al finalizar. Se ha comprobado que al crear nuevos perfiles no se puedan repetir los nombres de estos, que las opciones de dificultad, volumen de la música y de los efectos de sonido funcionan correctamente.

Como vemos, el juego es estable y funciona como debería en la versión presentada. En futuras versiones del juego se intentará solucionar los errores que pudieran aparecer en un futuro y comprobar si el juego funciona bien en distintas arquitecturas y plataformas.

### 5.5. Documentación

Para crear la documentación he usado *doxygen*, un generador automático para lenguajes como C++, C o Java, entre otros. La elección de este programa la hice porque ya lo habíamos usado durante la carrera, por ser fácil de utilizar, tener muchas opciones de personalización y crear la documentación en varios formatos automáticamente como html o  $\LaTeX$ .

Para escribir la memoria he usado  $\LaTeX$ [Oeti07], que aunque al principio pueda parecer difícil, una vez aprendido simplifica mucho la edición de texto. Una de las ventajas es que genera automáticamente contenidos como el índice, el índice de figuras, de tablas, etc. También he aprovechado que doxygen genera código  $\TeX$  para incluirlo en la memoria, ahorrándome mucho trabajo.



# Capítulo 6

## Conclusiones

### 6.1. Trabajo realizado

El proyecto, en términos generales, ha cumplido las expectativas que se tenían en un principio, aunque ha habido algunas limitaciones a la hora de desarrollarlo, como pueden ser la falta de tiempo, ya que es un proyecto muy trabajoso o la falta de personas participando en el, ya que el proyecto abarca muchos campos como pueden ser la programación, la edición de audio, la creación de texturas o los modelos en 3D, siendo este último uno de los más problemáticos, debido a lo complejo y laborioso que resulta.

Aun así, la jugabilidad es buena y el juego es fluido, los efectos de sonido y la música son acordes al juego (aunque no haya sonido 3D, que en este tipo de juegos no es necesario) y el juego goza de un nivel gráfico, no de última generación, pero sí bastante aceptable.

Cabe destacar que todo el proyecto ha sido desarrollado usando software libre, tanto las librerías (Ogre, OIS, SDL) como los programas para crear los modelos (Blender), las texturas (Gimp), etc. Como era de esperar, todo el contenido del proyecto será liberado bajo una licencia libre, la GPL v3 y la memoria y documentación bajo la licencia, libre también, GFDL.

### 6.2. Ampliaciones

En futuras versiones del juego podrían incluirse varias mejoras que no se han realizado, fundamentalmente, por falta de tiempo. Algunas de ellas se muestran a continuación:

- Una de las primeras mejoras evidentes sería portar el juego a otras plataformas, como windows, para así aumentar mucho el número de posibles

usuarios. Esto no debería suponer mucha complicación, ya que todas las librerías utilizadas son multiplataforma. De todos modos, esto estaba fuera de los objetivos principales del proyecto.

- Otra de las mejoras necesarias para poder captar un público mayor sería la traducción del juego a otros idiomas, sobre todo el inglés. Aun así el juego no lo necesitaría tanto como pudiera ser un juego de rol o aventura gráfica, por ejemplo, ya que la única parte que el usuario tiene que leer es la historia.
- El proyecto dispone de página web, pero no es más que un repositorio donde poder descargarlo o echarle un vistazo a la documentación, por lo que no estaría mal una web donde se publicaran las puntuaciones de los usuarios de forma automática y así poder competir con jugadores de cualquier parte del mundo.
- Como habrán podido leer en la introducción de este documento, la historia del juego está pensada para que pueda continuar. Por lo tanto, el juego se podría mejorar ampliando el número de misiones, que podrían desarrollarse usando los recursos ya existentes (clases, modelos 3D, sonidos...) o creando algunos nuevos.

# Apéndice A

## Definiciones y abreviaturas

**Audacity** Programa multiplataforma de grabación y edición de sonidos.

**Blender** Programa multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales.

**C++** Lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

**CEGUI** CEGUI (Crazy Eddie's GUI) es una librería escrita en C++ para desarrollar interfaces gráficas de usuario. Está diseñada especialmente para usarse en videojuegos, pero la librería también se puede usar con cualquier otro propósito.

**Compilación** Es el proceso por el cual se traducen las instrucciones escritas en un determinado lenguaje de programación a lenguaje máquina, el cual es interpretado por la computadora.

**Direct3D** Direct3D es parte de DirectX, una API propiedad de Microsoft disponible tanto en los sistemas Windows de 32 y 64 bits, como para sus consolas Xbox y Xbox 360 para la programación de gráficos 3D. El objetivo de esta API es facilitar el manejo y trazado de entidades gráficas elementales, como líneas, polígonos y texturas, en cualquier aplicación que despliegue gráficos en 3D, así como efectuar de forma transparente transformaciones geométricas sobre dichas entidades. Direct3D provee también una interfaz transparente con el hardware de aceleración gráfica.

Se usa principalmente en aplicaciones donde el rendimiento es fundamental, como los videojuegos, aprovechando el hardware de aceleración gráfica disponible en la tarjeta gráfica.

**Distribución** Una distribución GNU/Linux (llamadas también distribuciones Linux) es una variante de ese sistema operativo que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de software libre o también incorporar aplicaciones o controladores privativos.

**Doxygen** Doxygen es un Generador de documentación para C++, C, Java, Objective-C, Python, IDL (versiones Corba y Microsoft) y en cierta medida para PHP, C# y D. Dado que es fácilmente adaptable, funciona en la mayoría de sistemas Unix así como en Windows y Mac OS X. La mayor parte del código de Doxygen está escrita por Dimitri van Heesch.

Doxygen es un acrónimo de dox(document) gen(generator), generador de documentación para código fuente.

Varios proyectos como KDE usan Doxygen para generar la documentación de su API. KDevelop incluye soporte para Doxygen.

**FadeIn** Efecto que hace que el sonido aumente gradualmente desde cero hasta su volumen normal.

**FadeOut** Efecto que hace que el sonido se desvanezca gradualmente.

**GIF** GIF (CompuServe GIF) es un formato gráfico utilizado ampliamente en internet, tanto para imágenes estáticas como para animaciones. Es un formato sin pérdida de calidad para imágenes con hasta 256 colores, limitados por una paleta restringida a este número de colores. Por ese motivo, con imágenes con más de 256 colores (profundidad de color superior a 8), la imagen debe adaptarse reduciendo sus colores, produciendo la consecuente pérdida de calidad.

**Gimp** GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes digitales en forma de mapa de bits.

**GNU** El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU. El 27 de septiembre de 1983 se anunció públicamente el proyecto por primera vez en el grupo de noticias net.unix-wizards. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman como el "Manifiesto GNU", que establecieron sus motivaciones para realizar el proyecto GNU, entre las que



destaca "volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras".

**GNU/Linux** sistema operativo similar a Unix que utiliza como base las herramientas de sistema de GNU y el núcleo Linux. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL de GNU (Licencia Pública General de GNU) y otras licencias libres.

**GPL** La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**Interfaz** Método para facilitar la interacción del usuario con el ordenador.

**JPG** JPEG además de ser un método de compresión, es a menudo considerado como un formato de archivo. JPEG/Exif es el formato de imagen más común utilizado por las cámaras fotográficas digitales y otros dispositivos de captura de imagen, junto con JPEG/JFIF, que también es otro formato para el almacenamiento y la transmisión de imágenes fotográficas en la World Wide Web. JPEG/JFIF es el formato más utilizado para almacenar y transmitir archivos de fotos en Internet. Estas variaciones de formatos a menudo no se distinguen, y se llaman JPEG.

**KDE** K Desktop Environment o Entorno de Escritorio K, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

**L<sup>A</sup>T<sub>E</sub>X** Lenguaje de marcado para documentos, y un sistema de preparación de documentos, formado por un gran conjunto de macros de T<sub>E</sub>X, escritas inicialmente por Leslie Lamport (LamportT<sub>E</sub>X) en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con L<sup>A</sup>T<sub>E</sub>X es comparable a la de una editorial científica de primera línea. L<sup>A</sup>T<sub>E</sub>X es software libre bajo licencia LPPL.

**Librería** Una biblioteca (o librería) es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que

proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Algunos programas ejecutables pueden ser a la vez programas independientes y bibliotecas, pero la mayoría de éstas no son ejecutables. Ejecutables y bibliotecas hacen referencias (llamadas enlaces o links) entre sí a través de un proceso conocido como enlace (o link), que por lo general es realizado por un software denominado enlazador o linker.

**Licencia** Una licencia de software es un contrato entre el licenciante (autor/titular de los derechos de explotación/distribuidor) y el licenciatarario del programa informático (usuario consumidor, usuario profesional o empresa), para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas.

**Linux** Linux es el núcleo o kernel del sistema operativo libre denominado GNU/Linux (coloquial pero erróneamente llamado Linux). Lanzado bajo la licencia pública general (GPL - General Public License) de GNU y desarrollado gracias a contribuciones provenientes de todo el mundo, Linux es uno de los mejores ejemplos de software libre cuyos desarrolladores originales siguieron la filosofía de ese movimiento. Fue creado por Linus Torvalds en 1991.

**Malla** Es la superficie creada mediante un método tridimensional generado por sistemas de vértices posicionados en un espacio virtual con datos de coordenadas propios.

**Matamarcianos** Matamarcianos (shoot 'em up en inglés) es un género de videojuegos en el que el personaje principal es un objeto volador, que dispara contra hordas de enemigos que van apareciendo en pantalla.

**MP3** MPEG-1 Audio Layer 3, más conocido como MP3, es un formato de audio digital comprimido con pérdida.

**Multiplataforma** Multiplataforma es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en Windows en un procesador x86, en GNU/Linux en un procesador x86, y en Mac OS X en uno x86 (solo para equipos Apple) o en un PowerPC.

**OGG** Ogg es un formato de archivo contenedor multimedia, desarrollado por la Fundación Xiph.org y es el formato nativo para los códecs multimedia que

también desarrolla Xiph.org.

El formato es libre de patentes y abierto al igual que toda la tecnología de Xiph.org, diseñado para dar un alto grado de eficiencia en el "streaming" la compresión de archivos.

**Ogre 3D** Motor de renderizado 3D multiplataforma orientado a escenas, escrito en el lenguaje de programación C++.

**OIS** OIS (Object-Oriented Input System) es una librería para construir interfaces humano-ordenador con dispositivos de entrada.

**OpenGL** OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compite con Direct3D en plataformas Microsoft Windows.

**PDF** PDF (acrónimo del inglés Portable Document Format, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto). Está especialmente ideado para documentos susceptibles de ser impresos, ya que especifica toda la información necesaria para la presentación final del documento, determinando todos los detalles de cómo va a quedar, no requiriéndose procesos anteriores de ajuste ni de maquetación. Cada vez se utiliza más también como especificación de visualización, gracias a la gran calidad de las fuentes utilizadas y a las facilidades que ofrece para el manejo del documento, como búsquedas, hipervínculos, etc.

**PNG** PNG (Portable Network Graphics) es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes. Este formato fue desarrollado en buena parte para solventar las deficiencias del formato GIF y permite almacenar imágenes con una mayor profundidad de contraste y otros importantes datos.

**Script** Guión, script o archivo de órdenes o archivo de procesamiento por lotes (en inglés script) es un programa usualmente simple, que generalmente se

almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los shells sean a la vez intérpretes de este tipo de programas.

**Scroll** En un videojuego, se denomina scroll, desplazamiento, rolo o voluta al movimiento en 2D de los gráficos que conforman el escenario. Hablaremos de juegos de scroll horizontal cuando la acción se desarrolla horizontalmente y juegos de scroll vertical cuando se desarrolla verticalmente. También podemos encontrarnos con juegos de scroll multidireccional como juegos de fútbol o hockey con vista aérea.

**SDL** Simple DirectMedia Layer (SDL) es un conjunto de bibliotecas que proporcionan funciones para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes.

**SoundKonverter** Frontispicio para varios programas de codificación y decodificación de sonido.

**Subversion** Software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

**T<sub>E</sub>X** Sistema de tipografía escrito por Donald E. Knuth, muy popular en el ambiente académico, especialmente entre las comunidades de matemáticos, físicos e informáticos. T<sub>E</sub>X es generalmente considerado como la mejor forma de componer complejas fórmulas matemáticas, pero, especialmente en la forma de L<sup>A</sup>T<sub>E</sub>X y otros paquetes de macros, se puede usar para otras tareas de composición.

**Textura** Una textura es una imagen del tipo bitmap utilizada para cubrir la superficie de un objeto virtual, ya sea tridimensional o bidimensional, con un programa de gráficos especial. Multitexturizado es el uso de más de una textura a la vez en un polígono.

**WAV** WAV (o WAVE), apócope de WAVEform audio format, es un formato de audio digital normalmente sin compresión de datos desarrollado y propiedad de Microsoft y de IBM que se utiliza para almacenar sonidos en el PC, admite archivos mono y estéreo a diversas resoluciones y velocidades de muestreo, su extensión es .wav.

**WYSIWYG** Acrónimo de What You See Is What You Get (en inglés, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

**WYSIWYM** Acrónimo que significa "lo que ves es lo que quieres decir"(en inglés: What You See Is What You Mean). Es un paradigma para la creación de documentos alternativo al modelo (más difundido) WYSIWYG.

En este paradigma, el usuario se encarga de introducir los contenidos de forma estructurada siguiendo su valor semántico, en lugar de indicar su formato de representación final. Por ejemplo, indicando si lo que está escribiendo es un título, una sección, un autor, etc. Para poder utilizar este tipo de editores es necesario conocer, a priori, la estructura del documento que se va a editar. Además, el procesador de texto debe ser capaz de generar el documento en su formato final, coherente con el texto introducido y siguiendo la estructura utilizada.

La principal ventaja de este sistema es que se produce una total separación entre contenido y presentación. Por lo que el usuario sólo debe preocuparse de estructurar y agregar los contenidos, dejando los aspectos visuales a cargo del sistema de exportación. Además la exportación podrá realizarse en distintos formatos a partir de los mismos contenidos.

**YafRay** YafRay (Yet Another Free Raytracer) es un raytracer libre, concebido bajo licencia LGPL, que usa un lenguaje de descripción de escenas en XML. Recientemente ha sido integrado al programa de modelado 3D Blender, y es usado frecuentemente para renderizar las escenas creadas con él.



# Apéndice B

## Publicación

La página web principal del proyecto es:

<http://pfc-josec87.googlecode.com>

En ella podemos encontrar el código fuente del juego, así como la documentación (generada con Doxygen) y la memoria, también con su código fuente.

Para descargarnos la última versión del repositorio bastará con introducir la siguiente orden en una terminal:

```
svn checkout http://pfc-josec87.googlecode.com/svn/trunk/  
pfc-josec87-read-only
```

Para ello deberá tener instalado el paquete subversion. Puede hacerlo, en distribuciones basadas en debian, con el comando `apt-get install subversion` con permisos de superusuario.





# Apéndice C

## Instalación

### C.1. Instalación de las dependencias

Para instalar las dependencias procederemos a instalar las siguientes librerías [OgreWi]:

- sdl
- sdl mixer
- ois
- ogre

#### C.1.1. Distribuciones basadas en Debian

Para ello utilizaremos el siguiente comando (con permisos de root) `apt-get install libsdl1.2debian-all libsdl1.2-dev libsdl-mixer1.2 libsdl-mixer1.2-dev libogre-main-1.4.9 libogre-dev libois1 libois-dev`

#### C.1.2. Desde el código fuente

Para instalar las distintas librerías desde el código fuente visitar la siguiente página para más información:  
[http://www.ogre3d.org/wiki/index.php/BuildingFromSource\\_Shoggoth](http://www.ogre3d.org/wiki/index.php/BuildingFromSource_Shoggoth)

## C.2. Compilación

Para compilar el juego escribimos en una terminal la orden `make` (en el directorio donde esté el código fuente) que se encargara de compilar el juego automáticamente. Para borrar los archivos que puedan crearse que no son necesarios ejecutaremos `make clean`.

## C.3. Ejecución

Después de compilar el juego obtendremos el ejecutable del juego. Para ejecutarlo basta con introducir `./juego`.

## C.4. Posibles problemas

### C.4.1. Compilación

Si durante la compilación obtenemos el siguiente error:

```
OGRE/GLX/OgreTimerImp.h:33:31: error: OgrePrerequisites.h:  
No such file or directory
```

Entonces abriremos el archivo `OgreTimerImp.h`, localizaremos la línea `#include "OgrePrerequisites.h"` y añadiremos `../`, de forma que quede así `#include "../OgrePrerequisites.h"`

### C.4.2. Juego Ralentizado

Uno de los posibles problemas que podemos tener es que el juego no funcione de forma fluida. Lo primero que deberemos comprobar será si tenemos habilitada la aceleración 3D. Para ello escribiremos en una terminal el comando `glxinfo | grep direct` y comprobamos que nos devuelva la siguiente línea `direct rendering: Yes`. Si obtenemos `direct rendering: No`, quiere decir que no tenemos habilitada la aceleración 3D y deberemos acudir a la documentación de nuestra distribución o nuestra tarjeta gráfica para solucionarlo.

# Apéndice D

## Manual de usuario

### D.1. Controles

Para el control del juego únicamente se utilizara el teclado. A continuación se detallara como debe usarse dependiendo de si nos encontramos en el menú o en el desarrollo de una fase.

#### D.1.1. Menú

En el menú, podemos movernos hacia arriba y abajo por las opciones usando las teclas arriba y abajo del cursor. Para ejecutar una acción, como empezar una nueva partida o navegar hacia otro submenú, utilizaremos la tecla “enter” y para salir de un submenú podremos utilizar la tecla “enter” sobre la opción volver. Para cambiar opciones pulsaremos las teclas izquierda y derecha del cursor. Podemos verlo con más detalle en la siguiente imagen:

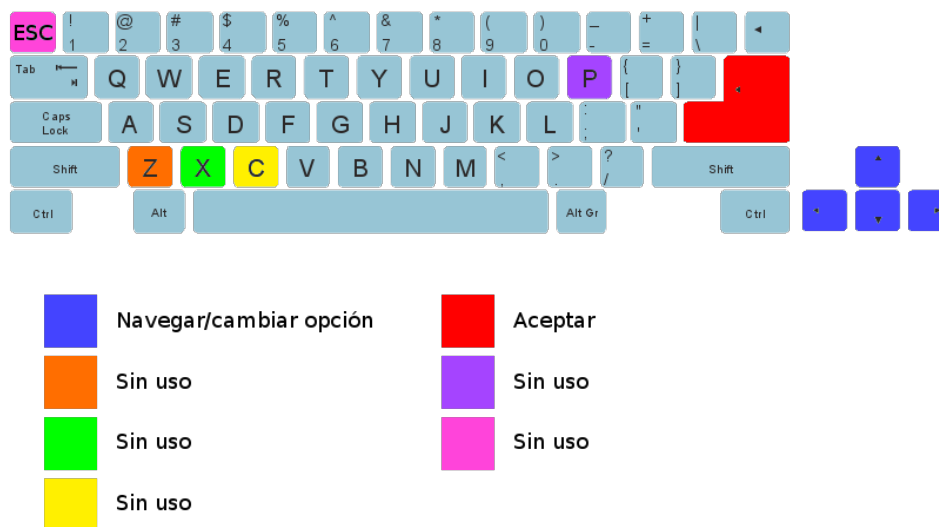


Figura D.1: Teclas durante el menú

Aunque no lo hemos visto en la imagen, para escribir los nombres de los perfiles podremos usar caracteres de la A a la Z y podremos borrarlo con la tecla retroceso.

## D.1.2. Juego

Durante el transcurso del juego las teclas a usar serán las siguientes:

- Cursores: para mover la nave arriba, abajo, delante o atrás, si estamos en una fase en dos dimensiones. Si estamos jugando en tres dimensiones, estos cursores sirven para rotar la nave hacia arriba, abajo, a la derecha o izquierda.
- Z: pulsando la tecla Z ejecutamos el disparo principal.
- X: la tecla X ejecuta el disparo secundario (en caso de estar en una fase en 2D).
- C: la tecla C ejecuta el disparo especial (en caso de estar en una fase en 2D y tener munición para dicho disparo).
- P: Pausamos el juego si no está pausado ya o lo reanudamos si sí lo está.
- Escape: Salimos momentáneamente de la partida y regresamos al menú principal.

En la siguiente imagen están marcadas las teclas:



Figura D.2: Teclas durante el juego

## D.2. Juego

Una vez que de comienzo la partida, nos encontraremos al mando de una nave, que deberá eliminar a las hordas de enemigos que vayan apareciendo. Para esto nos serviremos del disparo normal, del disparo secundario y del disparo especial, a la vez que nos movemos para esquivar sus ataques.

### D.2.1. Disparos

- Disparo normal: este será nuestro ataque principal. Consiste en un disparo que avanza hacia delante. Será el único disparo del que dispondremos en fases en tres dimensiones.
- Disparo secundario: este tipo de disparo es un poco mas complejo, vamos por partes: Si la fase se desarrolla en un planeta (hay tierra bajo nosotros) entonces este disparo consistirá en bombas que se sueltan y caen hacia el suelo. Con él podremos eliminar cualquier enemigo, incluso los tanques, pero no podremos eliminar otros disparos enemigos. En cambio, si la fase se desarrolla en el espacio, no habrá bombas, ya que no habrá gravedad que haga que la bomba caiga, en cambio, tendremos otro tipo de disparo similar

al disparo normal. Dicho disparo no irá hacia adelante, sino hacia arriba y abajo, pero tanto su alcance como velocidad serán menores. También tenemos que tener en cuenta que la cadencia del disparo secundario será mucho menor que la del disparo normal.



Figura D.3: Bomba

- Disparo especial: este disparo no estará siempre disponible, para habilitarlo tendremos que conseguirlo por medio del potenciador de disparo (los potenciadores se verán próximamente). Una vez obtenido dicho potenciador, podremos utilizar el disparo un máximo de 20 veces, hasta que volvamos a obtenerlo. El disparo consiste en tres disparos simultáneos, uno hacia delante y horizontal, otro con una ligera inclinación hacia arriba y otro con una ligera inclinación hacia abajo.

## D.2.2. Potenciadores

De vez en cuando, cuando destruyamos un enemigo (excepto tanques y fragmentos de meteorito), obtendremos un ítem que potenciará nuestra nave. Para recogerlos tendremos que pasar con la nave por encima de ellos. Los potenciadores son los siguientes:

- Vida: Cuando consigamos un potenciador de vida, una nueva vida será añadida al total disponible, siempre que no se supere el límite de cinco vidas.
- Escudo: Cuando consigamos el escudo, una esfera protectora rodeará nuestra nave, haciéndola inmune a los ataques enemigos. El escudo durará hasta que algún enemigo o proyectil impacte contra nuestra nave o hasta que se acabe el tiempo.

- Disparo especial: Con el potenciador de disparo, obtendremos un arma especial, la cual solo podremos utilizar 20 veces. Para reponer la munición tendremos que conseguir otro potenciador de disparo.

### D.2.3. Interfaz

- Puntuación: En la esquina superior izquierda podemos ver la puntuación que tenemos acumulada.



Figura D.4: Puntuación actual

- Vidas: En la esquina superior derecha podemos ver el número de vidas del que disponemos. Empezamos cada fase con tres vidas y podemos acumular hasta un máximo de cinco. En el caso de encontrarnos en la segunda fase, la vida se nos mostrará en la barra de la imagen 4.11.



Figura D.5: Número de vidas



Figura D.6: Barra de vida

- Escudo: Cuando consigamos un potenciador de escudo, en la esquina inferior izquierda podremos ver una barra azul que nos indicara el tiempo de escudo restante.
- Munición: Cuando consigamos un potenciador de disparo, en la esquina inferior izquierda, junto al indicador de escudo, podremos ver la munición que nos queda de disparo especial.



Figura D.7: Tiempo de escudo y munición

#### D.2.4. Enemigos finales

Una vez que hayamos acabado con todos los enemigos normales, pasaremos al enemigo final de la fase en la que nos encontremos. Pasaremos a detallar algunos consejos para superarlos.

- Enemigo final 1: Este enemigo final aparece por la derecha de la pantalla y se dirige hacia nosotros a la vez que nos dispara. La clave para superarlo está en esperarlo en un lado de la pantalla y hacer que venga hacia nosotros. Cuando este suficientemente cerca pasaremos por arriba o por abajo y lo esperaremos en el lado opuesto de la pantalla. Si es posible, es mejor pasar por arriba, ya que podremos dejarle caer alguna bomba, que le infringen más daño que los disparos normales.
- Enemigo final 2: Este enemigo final aparece la primera vez por la derecha y sigue hacia delante en un movimiento rectilíneo. Cuando desaparece de la pantalla por el lado izquierdo, vuelve a aparecer otra vez por otro lado, pero esta vez es aleatorio, no sabemos por donde va a aparecer, así que lo más recomendable es esperarlo en el centro de la pantalla.
- Enemigo final 3: Este enemigo aparece, como los anteriores, por el lado derecho y se dirige hacia el centro. Justo antes de llegar, empieza a rotar desde el centro de la pantalla a la vez que lanza sus proyectiles. Estos proyectiles tienen una diferencia con los demás, son teledirigidos. Una vez que el proyectil sea lanzado nos perseguirá por la pantalla. Para evitarlos tendremos dos opciones, esperar a que desvanezcan solos (tienen una duración



limitada) o dispararles. La estrategia a seguir es no quedarse en el centro de la pantalla, pues ahí es a donde van todos los proyectiles en un principio y tampoco arrinconarse en una esquina, pues nos acorralarán los disparos.

### **D.3. Ajustes**

Para cambiar los ajustes de los gráficos del juego iremos al directorio donde está instalado, entraremos en la carpeta `Conf` y abriremos con un editor de texto el fichero `config.cfg`. En dicho fichero podremos cambiar la resolución y decidir si queremos jugar en pantalla completa o no, cambiando estas líneas:

- `Full Screen=No`
- `Video Mode= 880 x 660`



# Bibliografía

## Referencias de Ogre3D

- [Ogre3D] The OGRE Team. *Ogre3D - Open source graphics engine*.  
<http://www.ogre3d.org>.
- [OgreWi] Ogre3D community. *Ogre Wiki*.  
<http://www.ogre3d.org/wiki/index.php>.
- [OgreMa] The OGRE Team. *Ogre Manual*.  
<http://www.ogre3d.org/docs/manual>.
- [OgreAP] The OGRE Team. *Ogre API Reference*.  
<http://www.ogre3d.org/docs/api/html>.
- [Junker] Gregory Junker. *Pro OGRE 3D Programming*. ISBN-13: 978-1590597101

## Referencias de programación de juegos

- [PLgam] John R. Hall. *Programming Linux Games*. ISBN-13: 978-1886411494

## Referencias de SDL

- [SDLTut] Alberto García Serrano. *Programación de videojuegos con SDL*.  
<http://www.agserrano.com> ISBN-13: 978-8495836083
- [SDLWik] The SDL community. *SDL documentation Wiki*.  
<http://www.libsdl.org/cgi/docwiki.cgi>.

- [AGAlba] Antonio G. Alba. *Tutorial de libSDL para la programación de videojuegos*. <http://softwarelibre.uca.es/wikijuegos>.
- [Pazera] Ernest Pazera. *Focus On SDL*. ISBN-13: 978-1592000302

## Referencias de OIS

- [OISPag] OISPag. *Object Oriented Input System project page*. <http://sourceforge.net/projects/wgois>.

## Referencias de Mxml

- [MXMLRE] Michael Sweet. *Mini-XML documentation*. <http://www.minixml.org/documentation.php>.

## Referencias de L<sup>A</sup>T<sub>E</sub>X

- [Oeti07] Oetiker, T.; Partl, H.; Hyna I. & Schlegl E. *La introducción no-tan-corta a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>* <http://www.ctan.org/tex-archive/info/lshort/spanish/lshort.pdf>
- [Mittel] F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, C. Rowley. *The L<sup>A</sup>T<sub>E</sub>X Companion* ISBN-13: 978-0201362992

## Referencias de Blender

- [Gonz06] Carlos González Morcillo. *Aprende en 24 horas Blender & Yafray*. <http://www.inf-cr.uclm.es/www/cglez>
- [Roland] Roland Hess. *The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender*. ISBN-13: 978-1593271664

## Referencias de C++

- [FundaC] F. Palomo Lozano, I. Medina Bulo, G. Aburrizaga García. *Fundamentos de C++*. ISBN-13: 978-8498280074

## Referencias de Make

- [Abur08] Gerardo Aburrizaga García. *Make. Un programa para controlar la recompilación.*

## Referencias de definiciones

- [WikiEs] Wikipedia Community. *Wikipedia, la enciclopedia libre.*  
<http://es.wikipedia.org>

- [WikiEn] Wikipedia Community. *Wikipedia, the free encyclopedia.*  
<http://en.wikipedia.org>



# GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The

“**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which



the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license

notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together

with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of

the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## **11. RELICENSING**

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

### **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. 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”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.