



Juegos en GNU/Linux:
Wolfenstein 3D - Spear of Destiny

Eye del Cul

1. Introducción

Bien, ya lo dijimos ayer,¹ pero vosotros no estabais:

«En su momento fue un videojuego revolucionario; en una escena todavía dominada por los arcades de plataformas, *Wolfenstein 3D* nos introducía en pequeños laberintos por los que nos podíamos mover libremente. Y al contrario que sus predecesores, ofrecía una ambientación más visceral y cercana a los jugadores: quiero decir, en lugar de castillitos fantásticos poblados de trasgos, aquí encontramos nazis malvados, retratos de Hitler y cruces gamadas..., y además, ¡zombis! ¿Quién podría resistirse? Después de todo, el Canal Historia lleva años sobreviviendo gracias a la Segunda Guerra Mundial [...]»

En este artículo hablaré de algunas de las alternativas que tenemos para hacer funcionar en GNU/Linux este viejo clásico y su segunda parte, *Spear of Destiny*.

2. Emulación

Sí, porque siempre existe la posibilidad de ejecutar *Wolfenstein 3D* con un emulador. Hoy por hoy, cuando el que más o el que menos dispone de un Pentium III con un montón de megahertzios, es una opción muy asequible. *Wolfenstein 3D* y *Spear of Destiny* corren perfectamente en DOSBox o en Dosemu —yo mismo los he probado en Dosemu con un Pentium II a 350 MHz, y eran más o menos jugables—.

La emulación con DOSBox es más completa; el inconveniente está en que se necesita una máquina un poco más potente que con DOSemu. Pero bueno, insisto; esto sólo significa un problema con los ordenadores más viejos. Los jugones más atrevidos incluso pueden probar correr el *Wolfenstein 3D* con el puerto [NewWolf](#), emulándolo con Wine.²

De todas maneras no he escrito este manual para ponerme a hablar sobre emuladores. Es todo un mundillo, y quienes no sepan usarlos y quieran aprender deberían buscar en otro lado los tutoriales oportunos.

¹Concretamente en <http://www.arcades3d.net/descargas/wolf3d/wolf3d.var>.

²Nota “a posteriori”: Funciona... En el mencionado P2/350 MHz, 96 MB de RAM y con una Nvidia TNT2 de 32 MB se puede jugar con Wine + NewWolf sin mayores inconvenientes.

3. Wolf4SDL

Así que hablaremos ahora de Wolf4SDL. Es un proyecto relativamente reciente, cuya primera versión pública apareció en diciembre de 2007, y la última en 2011. Se distribuye en forma de código fuente, así que hay que compilarlo primero para conseguir un ejecutable que podamos usar; pero no temáis, es sencillo, y aquí explicaré cómo hacerlo. De todos modos, hemos preparado unos [binarios ya compilados](#)³ para los impacientes.

3.1. Requisitos

¿Y qué se necesita para compilar Wolf4SDL? La lista es corta y terminaremos enseguida:

- Librerías SDL y SDL_mixer (ojo, de la **rama 1.2**).
- Las herramientas usuales de compilación; en este caso, GNU Make más el compilador de C++ incluido en GCC. Yo he probado las versiones 4.2.4, 3.4.6 y 3.3 de GCC y todas han funcionado sin problemas..., parece un código bastante sólido.

Si por alguna casualidad no tenéis alguna de estas dos cosas —sería extraño, porque la mayoría de distribuciones de Linux las instalan por defecto—, lo más fácil sería acudir a vuestros respectivos repositorios de paquetes. No debería haber ninguna complicación aquí...

3.2. Obtener el código fuente

El código fuente de Wolf4SDL está disponible para descargar en la [página oficial del proyecto](#),⁴ y además nosotros hemos alojado una [copia en Arcades 3D](#).⁵ En cualquiera de los dos casos hay que descomprimir el archivo descargado. Por ejemplo, abris una terminal y accedéis a la carpeta en la que lo hayáis dejado; en mi caso:

```
$ cd documentos/descargas
```

³<http://www.arcades3d.net/linux/archivos/Wolf4SDL-1.7-bin.tar.bz2>

⁴<http://homepage.o2mail.de/mkroll/>

⁵<http://www.arcades3d.net/linux/archivos/Wolf4SDL-1.7-src.tar.bz2>

Y ahora lo descomprimís... Los autores distribuyen un archivo en formato ZIP, nosotros usamos Bzip2 para ahorrarnos espacio. Así que:

```
$ unzip Wolf4SDL-1.6-src.zip  
(Descomprimir un fichero .zip)
```

```
$ tar -jxvf Wolf4SDL-1.6-src.tar.bz2  
(Abrir un paquete .tar.bz2, como los que tenemos en Arcades 3D)
```

De todas formas, lo que supongo que haréis la mayoría de vosotros será no complicaros la vida y usar una herramienta gráfica, como Ark o File-Roller.

3.3. Definir la versión del juego que usaremos

¿Y ahora? Pues nada, ya podemos ponernos a trabajar; bueno, “trabajar” suena como algo largo y tedioso, pero esto quedará despachado en menos de cinco minutos. Lo primero es acceder al directorio que habrá aparecido al desempaquetar el archivo con el código:

```
$ cd Wolf4SDL-1.7-src
```

Una vez ahí, hay que abrir con un editor de texto el fichero `version.h`, y especificar qué versión de *Wolfenstein 3D* o *Spear of Destiny* deseamos jugar. Diferentes variantes de este puerto para diferentes juegos se compilan por separado.

No temáis, es fácil; seguramente muchos lectores sabrán arreglárselas sin esta explicación. Pero, por si las moscas... El caso es que al comienzo del archivo veréis una serie de líneas como éstas:

```
//#define SPEAR  
//#define SPEARDEMO  
//#define UPLOAD  
#define GOODTIMES  
#define CARMACIZED  
//#define APOGEE_1_0  
//#define APOGEE_1_1  
//#define APOGEE_1_2
```

Esta mierda funciona del siguiente modo: una línea que comience por dos barras —“//”— está comentada y a efectos prácticos es como si no existiese, mientras que las líneas que empiezan por “#define” sí que tienen efecto. Cada variante de *Wolfenstein 3D* o *Spear of Destiny* que queramos compilar requiere una combinación diferente.

Así, la versión 1.4 de *Wolfenstein 3D* que distribuyó Activision —y que es la que más fácilmente encontraréis por Internet— necesita sólo estas dos líneas:

```
#define GOODTIMES
#define CARMACIZED
```

...Mientras que las demás deben ir comentadas. Y para jugar a la versión completa de *Spear of Destiny*:

```
#define SPEAR
#define CARMACIZED
```

Esto está explicado también en el propio archivo `version.h`, justo debajo del bloque de líneas que he citado. No tiene mucho misterio.

3.4. Compilación e instalación

Tras haber especificado de este modo qué versión del juego deseamos usar, sólo queda comenzar con la compilación en sí. Y para ello basta teclear en la terminal:

```
$ make
```

El proceso dura poco y, si cumplís los requisitos mencionados antes, no deberían surgir problemas. Al terminar, habrá aparecido en el directorio de trabajo —`Wolf4SDL-1.6-src`— un archivo ejecutable llamado `wolf3d`.

Pues bien, éste es el ejecutable que usaréis para jugar. Sólo os queda copiarlo a la carpeta en la que tengáis la instalación original de *Wolfenstein 3D* o *Spear of Destiny*, y ejecutarlo. Aunque debería mencionar que `Wolf4SDL` sólo funcionará si los **nombres de los archivos** de datos del juego **están en minúsculas**; es un requerimiento que comparte con `SDLWolf3D`, del que hablaremos más abajo. En concreto, vais a necesitar los siguientes ficheros:⁶

⁶Sustitúyase `.wl6` por `.sod`, en el caso de *Spear of Destiny*, o `.wl1` si se trata de la demo *shareware* de *Wolfenstein 3D*, etc.

- `audiohed.wl6`
- `audiot.wl6`
- `config.wl6`
- `gamemaps.wl6`
- `maphead.wl6`
- `vgadict.wl6`
- `vgagraph.wl6`
- `vgahead.wl6`
- `vswap.wl6`

Una última nota: si deseáis compilar Wolf4SDL varias veces —porque queráis usarlo tanto con *Wolfenstein 3D* como con su secuela—, es importante que, justo antes de cada compilación, limpiéis el directorio de trabajo con la siguiente orden:

```
$ make clean
```

Esto elimina todo el trabajo anterior —borra el archivo ejecutable `wolf3d` y los ficheros temporales que crea el compilador como paso intermedio— y deja el código listo para ser compilado de nuevo.

4. ECWolf

También se trata de un puerto reciente, y mientras escribo esto —en febrero de 2014— cuenta además con un desarrollo bastante activo. Derivado de Wolf4SDL, incorpora mucho código y características avanzadas de Zdoom.⁷ Muy recomendable para jugar, puesto que permite usar muchas resoluciones gráficas distintas, o controlar al personaje con teclado y ratón. Tenéis más información, claro, en su página oficial.⁸

En realidad no resulta imprescindible compilar ECWolf. Sus autores distribuyen una copia ya compilada para Ubuntu, que quizás pueda usarse en otras distros que sigan también el sistema de Debian de instalación de paquetes. Y aquí en *Arcades 3D* hemos preparado otros binarios que deberían

⁷De Zdoom hablamos en <http://www.arcades3d.net/linux/zdoom-linux.var>.

⁸<http://maniacsvault.net/ecwolf/>.

funcionar en cualquier Linux que cuente con las librerías SDL, y no se remonte más atrás de 2007.⁹

Sin embargo, como el saber no ocupa lugar, os enseñaremos a compilar el código fuente y crear así vuestros propios binarios:

4.1. Requisitos

Para esa tarea necesitaréis lo siguiente:

- Por supuesto, el compilador GCC, preferiblemente de las versiones 4.4.x en adelante —las anteriores pueden funcionar también, pero no está garantizado—.
- CMake 2.6 o superior.
- Librerías SDL (1.2.x) y SDL_mixer (mínimo 1.2.11).

Existen otras dependencias (entre ellas, las librerías GTK2, o Libjpeg), pero ya no resultan imprescindibles: si las tenéis instaladas ECWolf las utilizará, de lo contrario se compilará sin ellas.

En fin, descargad lo que necesitéis de vuestro gestor de paquetes, incluyendo los llamados “dev” o “devel” si corresponde (por aclarar esto, en Debian y sus derivadas tendríais, por un lado, `libsdl1.2`, y por otro `libsdl1.2-dev`; os harán falta ambos).

4.2. Compilación

Lo primero es descargarse el código fuente de la página oficial.¹⁰ Los autores lo distribuyen en un fichero comprimido con 7-Zip, que en Linux puede abrirse mediante P7Zip:

```
$ 7za x ecwolf-1.2.3-src.7z
```

Con esto habrá aparecido una carpeta llamada `ecwolf-1.2.3-src`, dentro de la cual encontraréis otras más, que guardan los diferentes componentes de ECWolf. Tendréis que entrar en ella, y crear ahí un directorio de trabajo:

⁹Fueron compilados con GCC 4.4.2 y Glibc 2.10.1 en Gentoo Linux. Ya de paso, al prepararlos desactivé el uso de GTK2 y de Libjpeg, para reducir dependencias.

Disponibles en <http://www.arcades3d.net/linux/archivos/ecwolf-1.3.1-bin-x86.tar.bz2>.

¹⁰Mirad en <http://maniacsvault.net/ecwolf/download.php>.

```
$ cd ecwolf-1.2.3-src
$ mkdir compilacion
$ cd compilacion
```

Oh, le he puesto `compilacion` de nombre para que no existan dudas sobre su cometido, aunque eso es lo de menos. Desde esta nueva carpeta, que ahora estará vacía, invocáis CMake tal que así:

```
$ cmake -DCMAKE_BUILD_TYPE=Release ..
```

Los dos puntos del final del comando son importantes: llevan a CMake un nivel atrás en el árbol de directorios, a `ecwolf-1.2.3-src`, donde se halla el fichero `CMakeLists.txt` principal, que dirigirá este primer paso del proceso. No los olvidéis 🙄.

Sí CMake termina sin errores, podéis comenzar con la compilación en sí:

```
$ make
```

No suele tardar mucho, algo menos de cinco minutos en mi Pentium IV; en ordenadores más modernos supongo que tendréis que esperar incluso menos. Cuando acabe, si todo ha ido bien —cumpliendo los requisitos enumerados arriba no deberían surgir problemas—, descubriréis dos archivos más:

- `ecwolf` (Ejecutable).
- `ecwolf.pk3`.

Ya sólo os quedaría copiarlos a una carpeta que contenga los *Wolfenstein 3D* o *Spear of Destiny* originales. Incluso pueden estar los dos mezclados en un mismo directorio, en cuyo caso ECWolf os preguntará a cual deseáis jugar.

5. SDLWolf3D

Nota: Éste es ya material obsoleto. Los autores originales de este puerto del juego también lo consideran así, y han retirado todos los contenidos de su antigua página oficial. SDLWolf3D es viejo, y quizás disfuncional en un Linux moderno. Tenedlo en cuenta.

Y si aun así deseáis probarlo, seguid leyendo.

Lo primero que habrá que hacer es descargarse el código fuente. En *Icculus.org* lo han borrado, pero en *Arcades 3D* todavía lo guardamos junto con otras reliquias; se trata de una copia descargada del antiguo directorio SVN en noviembre de 2006.¹¹

También podéis obtenerlo de *Arcades 3D*;¹² se trata de una copia descargada del antiguo directorio SVN en noviembre de 2006.

Dejáis ese archivo en una carpeta temporal —en este texto vamos a llamarle, no sé, descargas—, y lo descomprimís:

```
$ cd ~/descargas
$ tar -jxvf sdlwolf3d_src_112006_correg.tar.bz2
```

Tras esto, habrá aparecido un directorio denominado `sdlwolf3d`, y en su interior os encontraréis con tres más: `docs`, `src` y `macsrc`. Siguiendo con el ejemplo anterior, en la consola de comandos tendríamos algo así:

```
$ cd sdlwolf3d
$ pwd
/home/usuario/descargas/sdlwolf3d
$ ls
docs
src
macsrc
```

Más que nada, para hacernos una idea de cómo está esto organizado. El siguiente paso, evidentemente, será compilar SDLWolf3D. Pero antes detengámonos en los requisitos mínimos necesarios.

5.1. Requisitos

Básicamente son los siguientes:

- Librerías SDL, versiones 1.2.x.
- Copia del *Wolfenstein 3D* o *Spear of Destiny* originales —aunque sólo nos interesan los archivos de datos del juego—. Algunas versiones del *Wolfenstein 3D* pueden no funcionar correctamente; lo mejor es que uséis la 1.4.

¹¹El enlace: http://www.arcades3d.net/linux/archivos/sdlwolf3d_src_112006_correg.tar.bz2.

¹²http://www.arcades3d.net/linux/wolf3d_src_112006.tar.bz2

Respecto a las librerías SDL, probablemente vendrán incluidas “de fábrica” en vuestras distribuciones de Linux. Y de lo contrario, ya sabéis; usad Apt-get u otros gestores de paquetes similares. O si lo preferís, acercaros por www.libsdl.org..., aunque, insisto, no creo que sean muchos los que tengan que preocuparse por instalar las SDL.

5.2. Compilación

Bueno, ya con las librerías SDL, podéis poner os manos a la obra. Aunque debéis tener en cuenta una cosa; este puerto para GNU/Linux de *Wolfenstein 3D* es compatible en realidad con varios títulos y versiones de ellos: el *Wolfenstein 3D* original, la edición *shareware*, y también *Spear of Destiny* y su respectiva demo. Y antes de compilar nada, tendréis que especificar cuál vais a usar.

Para ello, hay un archivo llamado `version.h`, dentro de la carpeta `src`. Habrá que editarlo y cambiar un pequeño valor. Veámoslo:

```
$ cd src
$ pwd
/home/usuario/descargas/sdlwolf3d/src
$ vi version.h
```

Bueno, quién dice Vi dice Beaver o cualquier otro editor de texto...

La edición que tenemos que hacer es muy sencilla, y cualquier persona un poco despierta tendrá más que suficiente con los comentarios que incluye el propio archivo. Pero por si acaso lo comentaré rápidamente.

Hay que buscar tres líneas, en la mitad superior del fichero:

```
#ifndef WMODE
#define WMODE 3 ← O “2”, o “1”, etc.
#endif
```

El número que sigue a “`#define WMODE`” indica la versión de *Wolfenstein 3D* o *Spear of Destiny* para la que queremos compilar el puerto. Los valores pueden ser los siguientes:

- **0:** *Wolfenstein 3D shareware* (archivos de datos con extensión `.WL1`).
- **1:** *Wolfenstein 3D* completo (archivos de datos con extensión `.WL6`).

- **2:** Demo de *Spear of Destiny* (archivos de datos con extensión `.SDM`).
- **3:** *Spear of Destiny* (archivos de datos con extensión `.SOD`).

Basta con cambiar el valor que venga preestablecido por uno de estos cuatro números, dependiendo del juego en concreto que os interese. Y con esto ya podríais compilar tranquilamente el código:

```
$ pwd
/home/usuario/descargas/sdlwolf3d/src
$ make
.
. ← (Montones de letras y números).
.
$
```

Esto es bastante rápido. Cuando la compilación termine, tendremos en el directorio `src` un archivo ejecutable llamado `sdlwolf3d`. Como somos gente ordenada, lo movemos a una carpeta vacía a nuestra elección, que será la ubicación definitiva del juego:

```
$ mv sdlwolf3d /home/usuario/juegos/wolf3d
```

5.3. Sonido y música

Llegados a este punto, conviene aclarar que, por defecto, se compila una versión sin sonido del juego. Si queréis sonido, habréis de modificar también el archivo `src/Makefile`, aunque no temáis, es algo muy sencillo. Así, lo abrimos con cualquier editor de texto, y buscamos las siguientes líneas — forman el tercer o cuarto párrafo—:

```
# no sound
OBJS += sd_null.o
# sound using OSS
#OBJS += sd_oss.o fmopl.p
#CFLAGS += -D_REENTRANT
#LDLIBS += -lpthread
```

Sí, es justo lo que estáis pensando; comentamos la línea `“OBJS += sd_null.o”` y quitamos las almohadillas que preceden a las que hay bajo de `“# sound using OSS”`. De este modo, nos quedaría algo así:

```
# no sound
#OBJS += sd_null.o
# sound using OSS
OBJS += sd_oss.o fmopl.p
CFLAGS += -D_REENTRANT
LDLIBS += -lpthread
```

Y tras hacer estos cambios, guardamos el archivo y procedemos a compilar el código tal y como hemos visto antes, usando el comando `make`.

5.4. Instalación

Una vez compilado SDLWolf3D —con soporte para sonido, o sin él—, sólo nos quedaría copiar a la carpeta dónde dejamos el fichero `sdlwolf3d` los archivos de datos del juego original —que se supone que tenemos 🤪—. Éstos son los siguientes, aunque como ya he comentado antes, la extensión cambia según el juego que estemos instalando; WL6 para *Wolfenstein 3D*, SOD para *Spear of Destiny*, etc:

- `audiohed.wl6`
- `audiot.wl6`
- `config.wl6`
- `gamemaps.wl6`
- `maphead.wl6`
- `vgadict.wl6`
- `vgagraph.wl6`
- `vgahead.wl6`
- `vswap.wl6`

Cuidado, no olvidéis que estáis usando un sistema de tipo Unix, y que por lo tanto “`config.wl6`” **no** es lo mismo que “`CONFIG.WL6`”; los archivos anteriores deben estar en minúsculas, o el puerto no los encontrará. Lo comento para evitarle a más de uno algún dolor de cabeza estúpido...

Y ahora ya podréis jugar: ejecutad sin más el fichero `sdlwolf3d`:

```
$ ./sdlwolf3d
```

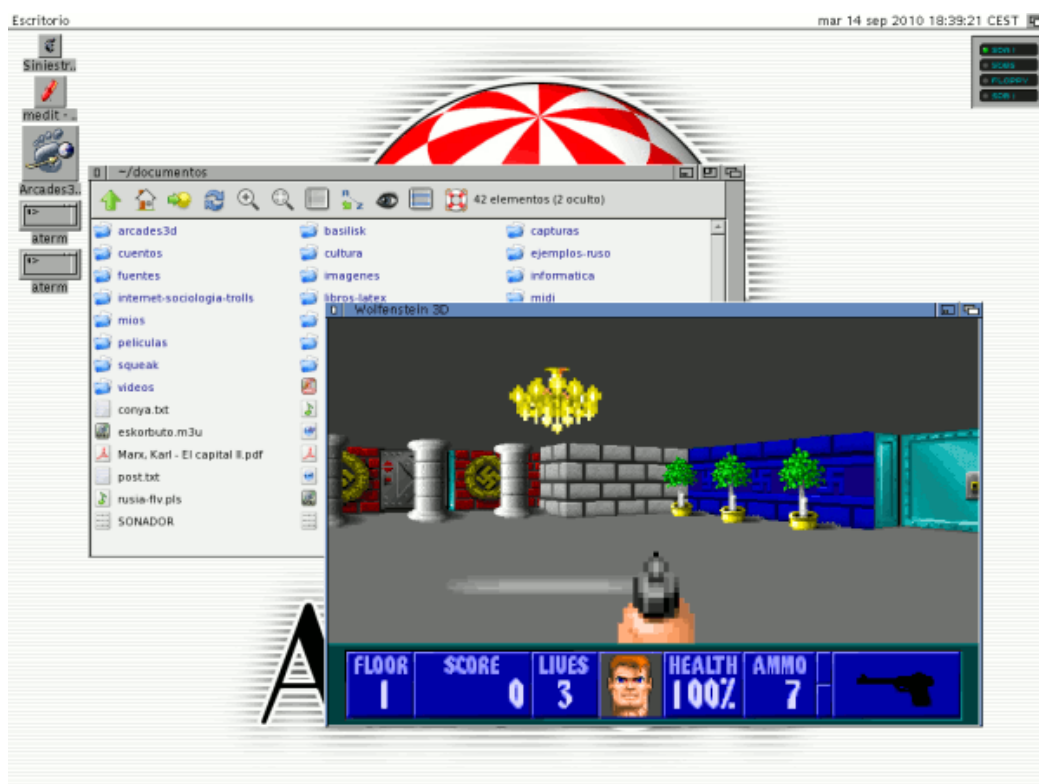
El juego arrancará en una pequeña ventana, pero debería ser posible ponerlo a pantalla completa pulsando al mismo tiempo [ALT] y [INTRO]. De lo contrario, es posible ejecutarlo del siguiente modo:

```
$ ./sdlwolf3d -fullscreen
```

6. Capturas de Pantalla

Y como siempre, las capturas de pantalla reglamentarias...

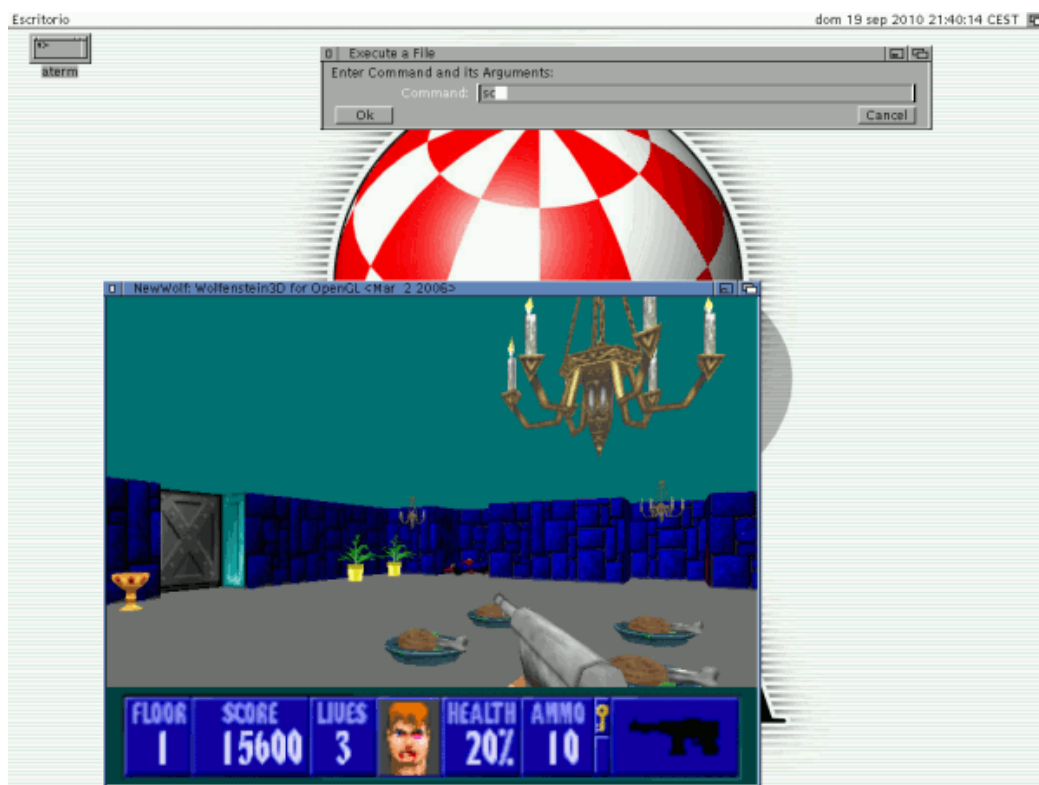
Éste es el aspecto de Wolf4SDL, del que hemos hablado al principio de este manual, usando una resolución de 640x400 píxeles;¹³ se trata de uno de los niveles originales de *Wolfenstein 3D*, no recuerdo cuál:



Los lectores más curiosos quizás se pregunten qué clase de escritorio es el que se ve en la imagen... Se trata de AmiWM, un gestor de ventanas nostálgico que trata de imitar la interfaz del Amiga. Tiene su gracia.

¹³Podéis especificar con qué resolución jugar, arrancando Wolf4SDL con el comando `./wolf3d --res ancho alto`, aunque los valores que uséis deben ser multiples de 320 y 200/240, respectivamente. Ya de paso, mencionaré que es posible especificar que el juego se ejecute en una ventana de escritorio añadiendo la opción `--windowed`.

Al principio de este manual dije que NewWolf puede funcionar en Linux, emulándolo con Wine. Pero claro, decirlo es fácil, hace falta alguna prueba... Y aquí la tenéis.¹⁴



La captura de pantalla muestra NewWolf corriendo emulado con la versión 0.9.28 de Wine. Fijaos en los modelos tridimensionales para las armas y objetos del escenario, algunos están bastante conseguidos.

¹⁴Imagen actualizada en septiembre de 2010; ésta es más bonita e ilustrativa que la anterior, aunque los detalles técnicos no cambian: sigo usando la misma versión de Wine —si algo funciona, ¿para qué cambiar?—.

Y aquí tenemos de nuevo a *Wolfenstein 3D*, ahora corriendo con el puerto de Icculus. Se carga en una pequeña ventana del escritorio, pero como ya hemos comentado se puede jugar también a pantalla completa:



7. Créditos

Manual escrito por **Eye del Cul** en enero de 2006. Revisado en noviembre de 2006, y de nuevo en septiembre de 2010. Vuelto a revisar en febrero de 2014, para borrar el contenido más obsoleto (WolfGL).

Podéis distribuir y modificar este documento a vuestro antojo, colgarlo en cualquier página web o publicarlo por partes... Eso sí, sería un detalle por vuestra parte que, de hacerlo, mencionárais la fuente ([Arcades 3D](#)) y a los autores originales; aunque bueno, siendo realistas, nadie os va a obligar. Todo queda en vuestra buena voluntad.

Versión en PDF creada con L^AT_EX.

¡Visita *Arcades 3D*!