



Juegos en GNU/Linux:
Duke Nukem 3D (usando JFDuke3D)

Karnizero - Eye del Cul

Seguramente, muchos de los que habéis compilado y jugado al *Duke Nukem 3D* en Linux habéis bajado el código fuente desde icculus.org; y ahora estaréis pensando, «vale, puedo jugar al *Duke*, pero yo quiero jugarlo con gráficos buenos, con texturas de alta resolución y todo eso». Pues bien, para eso tenemos el puerto **JFDuke3D** —también llamado Jonof—, que además de proporcionarnos aceleración gráfica soporta modelos en 3D y texturas de alta resolución —o sea, el «*High Resolution Pack*» o HRP—; y lo mejor de todo, ¡HAY VERSIÓN LINUXERA!

Sí, es una genial noticia, pero no la única; ya que en este caso... ¡No hay que compilar nada! De hecho sólo tendremos que bajarnos los binarios —es decir, los archivos ejecutables correspondientes— y listo.

1. Instalando el juego en GNU/Linux

1.1. Requisitos

Aunque no tengamos que compilar, sí necesitaremos bajarnos y configurar las librerías con las que ha sido compilado el ejecutable, y son las siguientes (probablemente, algunas o incluso la mayoría de ellas ya vendrán con tu distribución):

- SDL 1.2 o superior.
- libgtk-x11 2.0 o superior.
- libpango 1.0 o superior.
- libglib 2.0 o superior.
- SDL_mixer 1.2 o superior.
- libcairo.
- libfreetype.
- libvorbis.
- libogg.
- libsmpeg.
- libICE.
- ncurses.

Una vez que tengamos instaladas las librerías citadas ya podremos ejecutar el binario de JFDuke3D sin problema alguno.

Nota: A grandes rasgos, las librerías estrictamente necesarias son GTK2 —que arrastra consigo otras como Pango, Cairo y demás—, SDL y SDL_mixer. Las otras mencionadas en la lista quizás os hagan falta, quizás no. En todo caso, resultan todas bastante comunes, y una mayoría vendrán ya incluidas en vuestro sistema Linux. Y las que no, se obtienen fácilmente de cualquier gestor de paquetes.

1.2. Configuración e Instalación

Bueno, más sencillo imposible:

Creamos una carpeta en nuestro directorio personal, o donde queramos tener instalado el juego; por ejemplo, en `/home/pepito/juegos/jfduke`.

```
$ mkdir juegos
$ cd juegos
$ mkdir jfduke
```

A continuación nos descargamos los binarios del *JFBuild* y del *JFDuke3D*. Sin embargo, no hace falta abrir el Firefox para bajárselos. Simplemente hacemos un “`wget`” a la siguiente dirección:

- <http://www.arcades3d.net/linux/archivos/jfduke3d-20051009-bin.tar.bz2>

```
$ cd juegos
$ cd jfduke
$ wget http://www.arcades3d.net/linux/archivos/jfduke3d-20051009-bin.tar.bz2
```

Lo cual nos descargará el archivo `jfduke3d-20051009-bin.tar.bz2` que gentilmente hemos alojado en *Arcades 3D*. Se trata de la versión de octubre de 2005, la última publicada antes de que el autor de JFDuke3D dejara el proyecto. Fue compilada por un servidor en 2006, en Debian GNU/Linux y con GCC 3.3, pero debería seguir funcionando en sistemas más modernos, siempre que contengan las librerías requeridas

A continuación descomprimos el fichero:

```
$ tar -jxvf jfduke3d-20051009-bin.tar.bz2
$ ls
.
..
duke3d
build
$
```

Y ya tenemos los ejecutables.

Ahora necesitaremos los siguientes archivos de la versión atómica del *Duke Nukem 3D*:

- duke3d.grp
- USER.CON
- GAME.CON
- DEFS.CON
- lookup.dat

Tened en cuenta las mayúsculas y minúsculas.

Estos archivos los tendrás que conseguir de algún lado —si tienes el juego original, mejor que mejor 😊—. Mira, por ejemplo, en la sección oportuna de *Arcades 3D*.¹ Una vez estén en tu poder, coloca estos cinco archivos en la carpeta en la que tienes los ejecutables; y desde un terminal, ejecuta el binario duke3d y... ¡A jugaaar!

```
$ cd juegos
$ cd jfduke
$ ./duke3d
```

Y para arrancar el Build —el editor de mapas—:

```
$ cd juegos
$ cd jfduke
$ ./build
```

Con esto ya podemos disfrutar del juego usando el puerto JFDuke3D...

¹<http://www.arcades3d.net/descargas/duke3d/duke3d.var>

1.3. Usando el High Resolution Pack

Ahora bien, si lo que queremos es ejecutar el *Duke* con el HRP («*High Resolution Pack*») para gozar de los modelos en 3D y demás pijadas gráficas, procederemos de la siguiente forma:

Primero hacemos un “`wget`” a la siguiente dirección:

- http://moreplease.free.fr/HRP/Parkar/dn3d_hrp-2005-11-01.zip

O sea:

```
$ cd juegos
$ cd jfduke
$ wget http://moreplease.free.fr/HRP/Parkar/dn3d_hrp-2005-11-01
.zip
```

En total son unos 100 MB —así que paciencia 😊—. Se trata de una versión antigua del «*High Resolution Pack*», porque las modernas están pensadas para *Eduke32* y probablemente no funcionarán bien con *JFDuke3D*. Cuando acabe la descarga, podemos descomprimir el fichero usando la herramienta de descompresión *File-Roller*, o cualquier otra de vuestro gusto:

```
$ file-roller dn3d_hrp-2005-11-01.zip
```

Tendremos que descomprimir los ficheros en la carpeta principal del *Duke Nukem 3D*, dónde hemos colocado antes los binarios. Como nosotros tenemos la versión atómica, extraeremos los ficheros `duke3d_hrp.zip` y `maphacks.zip`, situados dentro del subdirectorio comprimido `maphacks_atomic`. Fíjate que estos dos archivos **no** los tienes que descomprimir. O sea, seleccionas el primero de ellos y lo guardas en la carpeta del *Duke*, y después haces lo mismo con el segundo.

Ahora tendremos que ejecutar el *Duke*, pero deberemos indicarle que lea estos dos ficheros; y para ello, escribiremos una pequeña línea de comandos:

```
$ cd juegos
$ cd jfduke
$ ./duke3d /gduke3d_hrp.zip /ghacks.zip
```

Así el señor Duke tendrá un aspecto mucho mejor.

Te parecerá un poco pesado escribir ese comando cada vez que quieras jugar..., pero no te preocupes, es posible ahorrárselo. Podemos preparar un sencillo *script* que lo haga automáticamente.

Abre un editor de textos, por ejemplo el Gedit, el Kate o cualquier otro, y copia y pega el siguiente texto, tal y como aparece aquí (sustituyendo, evidentemente, «*/home/pepito*» por el directorio que uses tú):

```
#!/bin/bash
cd /home/pepito/juegos/jfduke
./duke3d /gduke3d_hrp.zip /ghacks.zip
```

Guarda el documento con el nombre `jfduke_hrp`, y dale permisos de ejecución:

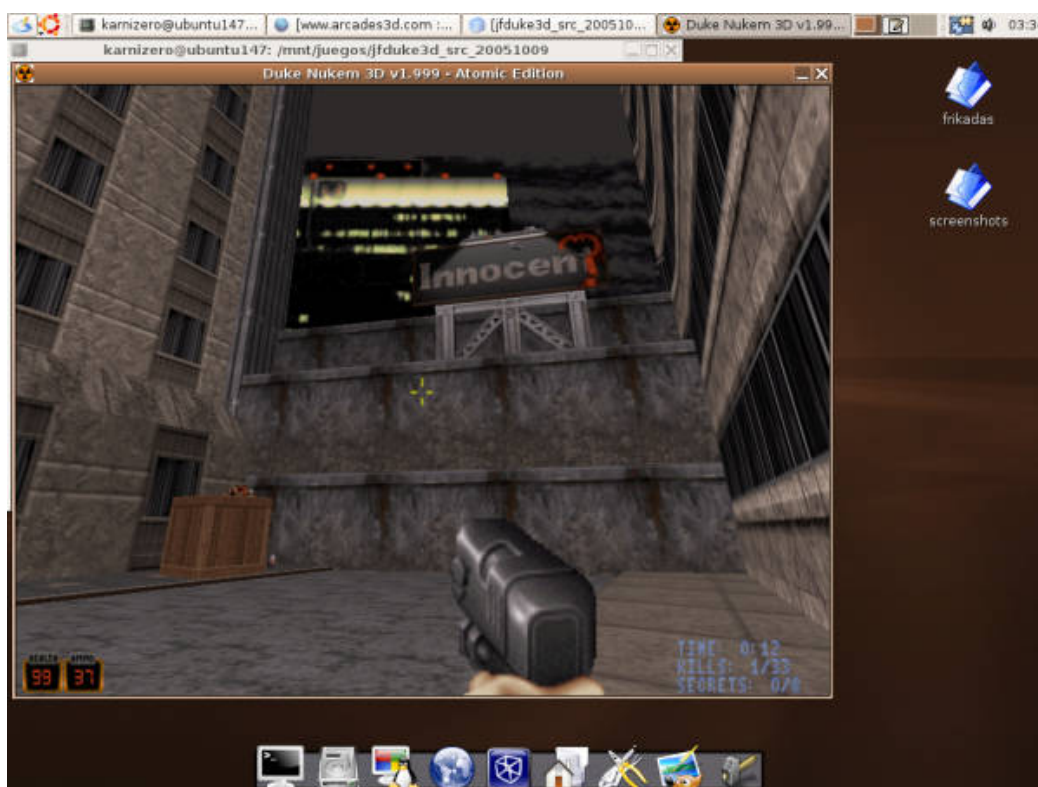
```
$ chmod u+x jfduke_hrp
```

A partir de ahora, siempre que ejecutemos el *script* el juego arrancará con el puerto JFDuke3D y usando el «*High Resolution Pack*». Puedes guardar el *script* en el escritorio, o en cualquier otro sitio que desees.

2. Capturas de Pantalla

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

Aquí podéis ver el juego ejecutándose con JFDuke3D, con el modo gráfico habitual (esto es, sin el «*High Resolution Pack*»). También puede ponerse a pantalla completa, para quien le guste más así:



Y aquí podéis ver el mismo rincón del primer nivel de *Duke Nukem 3D*... Pero esta vez, usando el «*High Resolution Pack*». La captura de pantalla no le hace justicia, pero creo que aún así la diferencia es bastante visible:



En todo caso, vale la pena recordar que para poder disfrutar del «*High Resolution Pack*» sin problemas —tirones, lentitud, fallos visuales, etc— hace falta un ordenador más o menos decente. No recuerdo los requerimientos exactos, pero sí es necesario disponer de aceleración 3D —OpenGL—, y probablemente una máquina de tipo Pentium III o superior con una cantidad aceptable de memoria RAM.

Una última captura de pantalla. Esta vez se trata del editor de mapas Build corriendo en una ventana del escritorio Fluxbox, en la versión 5.1 de Vector Linux. La versión de Build que incorpora el puerto JFDuke3D resulta bastante más potente que la original de *Duke Nukem 3D*, o que la que encontramos en el puerto de Icculus.



El nivel que aparece en el editor es «*The Return*», creado por Papamonos y comentado hace tiempo en la sección de mapas de *Arcades 3D*.

3. Compilando el Puerto JFDuke3D

Bueno, **Karnizero** nos explicó arriba cómo jugar empleando los binarios del puerto JFDuke3D. Pero el código fuente también está disponible para descargarlo, y los aficionados más maniáticos probablemente querrán compilarlo ellos mismos. Así que aquí comentaré cómo hacerlo —sí, yo soy uno de ellos y me divierte hacer un `make install` de vez en cuando—.

3.1. Requisitos

Pues nada, manos a la obra. Lo primero de todo, necesitaremos lo siguiente:

- Código fuente del motor gráfico (JFBuild).²
- Código fuente del puerto JFDuke3D.³
- Librerías SDL instaladas. En este caso, SDL, SDL_mixer, SDL_ttf y SDL_image (con los respectivos paquetes “dev” en las distros que lo requieran, como Ubuntu).
- Librería Fmod.
- Ensamblador NASM.
- Y, obviamente, el compilador GCC.

Muchas distribuciones de GNU/Linux incluirán ya en su instalación por defecto las librerías citadas; y las que no, probablemente las tengan disponibles en sus respectivos repositorios de paquetes. En todo caso, no debería resultar problemático obtenerlas, bien mediante herramientas como Apt, Emerge y similares, bien descargándolas directamente de sus sitios web:

- SDL: <http://www.libsdl.org>.
- Fmod: la **versión 3.75**, necesaria para compilar JFDuke3D, es vieja y ya no se halla disponible en su página oficial, pero en *Arcades 3D* hemos subido una copia.⁴
- NASM: <http://sourceforge.net/projects/nasm/>.

²http://www.arcades3d.net/linux/archivos/jfbuild_src_20051009.tar.bz2.

³http://www.arcades3d.net/linux/archivos/jfduke3d_src_20051009.tar.bz2.

⁴<http://www.arcades3d.net/linux/archivos/fmodapi375linux.tar.bz2>.

3.2. Compilación e instalación

Solucionado el asunto de las librerías, ya podemos ponernos a instalar el JFDuke3D. Crearemos primero un directorio en el que desempaquetar los dos ficheros comprimidos que contienen su código fuente. En este ejemplo le he llamado `jonof`, y completada la extracción de archivos debería contener lo siguiente:

```
$ cd jonof
$ ls
jfbuild_src_20051009/
jfbuild_src_20051009.patch
jfduke3d_src_20051009/
jfduke3d_src_20051009.patch
```

Como veis, hay dos nuevas carpetas (`jfbuild_src_20051009` y `jfduke3d_src_20051009`), y dos ficheros con extensión `.patch`. Éstos son dos parches para aplicar al código de JFDuke3D. Básicamente, contienen la implementación del sonido en Linux. Nacieron como un apaño provisional, concebido por una tercera persona mientras Jonathon Fowler trabajaba en algo mejor. Sin embargo, poco después cesó el desarrollo de este puerto, y han quedado por tanto como única solución disponible: si queréis oír a los monstruos deciros cosas bonitas, hay que usarlos.

Tal que así:

```
$ patch -p0 < jfbuild_src_20051009.patch
$ patch -p0 < jfduke3d_src_20051009.patch
```

Para que la compilación vaya bien, necesitamos crear un enlace simbólico llamado `build` en esta misma carpeta, que apunte al directorio dónde hemos descomprimido el código fuente del JFBuild:

```
$ pwd
/home/usuario/jonof
$ ln -s jfbuild_src_20051009 build
```

Y ya falta menos...

Una vez hecho esto, entramos en el directorio del código de JFDuke3D. Tendremos que editar el archivo `Makefile`, pero no temáis, es sólo un pequeño cambio en una línea, nada serio. Esto podemos hacerlo con nuestro editor de texto favorito —Vi, Beaver..., cualquiera sirve—:

```
$ cd jfduke3d_src_20051009
$ vi Makefile
```

¿Y qué hay que hacer? Poca cosa. Buscamos la siguiente línea:

```
-I$(INC:=) -I$(EINC:=) -I$(SRC)jmact -I$(SRC)jaudiolib #-I../
jfaud/inc
```

Y la sustituimos por esta otra:

```
-I$(INC:=) -I$(EINC:=) -I$(SRC)jmact -I$(SRC)jaudiolib -I/usr/
local/include/fmod #-I../jfaud/inc
```

En realidad, la única diferencia que hay entre ellas es el texto `-I/usr/local/include/fmod`. Sí, en la carpeta `/usr/local/include` deberemos tener un enlace simbólico llamado `fmod`, que apunte al directorio en el que se guardan los “*headers*” de la librería `Fmod`; este directorio suele ser `/usr/include/fmod`, y los archivos que contiene, `fmod.h` y un par más de nombres similares.

Comento esto último sólo por afán informativo. Si la librería `Fmod` está bien instalada, no tendremos que preocuparnos por el citado enlace simbólico ni por nada que tenga que ver con él. Si tenéis dudas sobre esto consultad la última parte del manual, a partir de la página [12](#).

¡Y por fin llegó el momento de teclear la palabra mágica! Para compilar el puerto `JFDuke3D` no nos queda más que un último paso:

```
$ make
```

Y si todo va bien, después de unos minutos y montones de letras y números en la consola de comandos, tendremos dos archivos ejecutables: `duke3d` —el fichero para arrancar el juego— y `build` —el editor de mapas—. Simplemente los copiamos a la carpeta en la que tengamos instalado el *Duke Nukem 3D* original, como ya vimos en la primera parte del manual... Y ya podemos jugar.

4. Anexo: Instalando Fmod y Timidity

4.1. Librería Fmod

Lo prometido es deuda, y aquí explicaré cómo instalar manualmente la librería de sonido Fmod. Ojo, me refiero aquí a la **versión 3.75**, que es la que necesitaréis para compilar JFDuke3D con sonido en Linux. Sí, se trata de una versión antigua y sin soporte, pero no hay otro remedio que recurrir a ella.

Pues nada, en primer lugar nos descargamos el archivo correspondiente, llamado `fmodapi375linux.tar.bz2`, y lo descomprimimos. Como en la página oficial de Fmod parece que ya no lo guardan, hemos subido una copia a *Arcades 3D*:

```
$ wget http://www.arcades3d.net/linux/archivos/fmodapi375linux.tar.bz2
$ tar -jxvf fmodapi375linux.tar.bz2
```

A continuación necesitamos crear varias carpetas dentro de `/usr/local`, y para hacerlo es probable que necesitéis acceder a la cuenta del administrador.

Así, por ejemplo:

```
$ su
Password: ← Escribimos la contraseña del administrador.
# mkdir -p /usr/local/fmod/include/fmod
# mkdir -p /usr/local/fmod/lib
```

Y accedemos a los contenidos que acabamos de descomprimir para comenzar a copiar los archivos necesarios. Lo único que nos interesa aquí es la carpeta llamada `api`:

```
# cd fmodapi375linux
# cd api
# cp -v libfmod-3.75.so /usr/local/fmod/lib
# cp -v inc/* /usr/local/fmod/include/fmod
```

Ahora necesitamos crear algunos enlaces simbólicos, para que los diversos programas puedan encontrar fácilmente la librería Fmod cuando lo requieran:

```
# ln -s /usr/local/fmod/lib/libfmod-3.75.so /usr/local/fmod/lib/libfmod.so
# ln -s /usr/local/fmod/lib/libfmod-3.75.so /usr/local/lib/libfmod.so
# ln -s /usr/local/fmod/include/fmod /usr/local/include/fmod
```

Y ya lo tenemos... Al final no era tan difícil, ¿verdad?

4.2. Timidity

Esto ya se explicó en otro tutorial, el de [Doom y derivados \(con Zdoom\)](#), y de hecho lo que sigue es más o menos una copia del texto que escribí en él. Sucede que descubrí por casualidad que con Timidity y cierto archivo de fuentes de sonido también es posible escuchar la música de *Duke Nukem 3D*. Así que si queréis deleitaros con las fantásticas melodías del juego, seguid leyendo...

Antes que nada, **usuarios de Ubuntu** y similares: abrid Synaptic, buscad Timidity, marcadlo, clicad con el ratón en «*Aplicar*»... ¡Hecho! No hace falta que sigáis leyendo.

¿Pero qué pasa si vuestra distribución de Linux no tiene Timidity en sus repositorios, o si una lluvia de meteoritos destruye todos los servidores de Canonical y asociados, o si...? Lógicamente, el primer paso sería descargarnos alguna versión de Timidity. La más reciente a día de hoy creo que es la 2.13; podéis echar un vistazo a su página oficial⁵ para conseguirla.

4.2.1. Compilar Timidity

En principio, como ya he dicho, sería preferible descargar paquetes ya compilados para vuestras distribuciones. Pero si por alguna razón deseáis compilar Timidity vosotros mismos, el proceso es sencillo. Una vez descomprimido el código fuente, basta con teclear lo siguiente en una consola de comandos:

```
$ ./configure --enable-audio=oss,alsa --enable-server --enable-alsaseq
$ make
$ su
Password: ← Entramos en la cuenta del administrador.
# make install
```

⁵<http://timidity.sourceforge.net>.

Como podéis ver, la primera orden (`./configure`) admite diversas opciones; son importantes, porque con ellas indicaremos qué características del programa queremos activar. Así, en este ejemplo configuramos Timidity para que emplee OSS o ALSA para reproducir los midis (`--enable-audio=oss,alsa`), cosa que funcionará en la mayoría de sistemas Linux; las otras dos opciones no sabría explicar bien para qué sirven exactamente, pero creo que es recomendable usarlas... 😊

En fin, si deseáis ver con tranquilidad todas las opciones de configuración que pueden emplearse con Timidity, teclad una orden como ésta:

```
$ ./configure --help | less
```

Aunque de todas formas, Timidity, compilado tal y como muestra el ejemplo anterior, probablemente correrá sin problemas en casi todas las máquinas con Linux y una tarjeta de sonido bien configurada.

4.2.2. Instalando el archivo de sonidos

Bueno, ya tenemos Timidity compilado e instalado en nuestro ordenador... Lo siguiente que nos hará falta será un archivo de sonidos digitalizados, correspondientes a las notas e instrumentos que emulará Timidity al reproducir la música. Supongo que habrán bastantes circulando por Internet. El que uso actualmente, después de haber probado varios, y me ha dado mejores resultados, es [éste](#), enlazado al final de la [página oficial de la librería SDL_mixer](#).

No obstante, también puede obtenerse rápida y comodamente usando Wget:

```
$ wget http://www.libsdl.org/projects/SDL_mixer/timidity/timidity.tar.gz
```

Una vez tengamos el fichero de sonidos a buen recaudo en nuestro ordenador, lo copiamos a `/usr/local/lib`, y lo descomprimos e instalamos:

```
$ su
Password:
# cd /usr/local/lib
# mv /home/usuario/timidity.tar.gz .
# tar -zxvf timidity.tar.gz
```

Esto crea un directorio llamado `timidity` dentro de `/usr/local/lib`, con todo el contenido necesario para funcionar. Sí, incluido un archivo `timidity.cfg` hecho a medida —en versiones viejas de este tutorial, usando otros ficheros de sonidos, había que escribirlo desde cero o buscarse la vida—.

Por último, es posible que necesitéis una copia —o un enlace simbólico— de este archivo en el directorio `/usr/local/share/timidity`. En mi caso lo solucioné de esta manera:

```
# cd /usr/local/share
# ln -s /usr/local/lib/timidity timidity
```

Y ahora sí, hemos terminado... Reiniciad el ordenador para asegurarnos de que todo funciona correctamente, y a disfrutar.

5. Créditos

La primera parte del manual fue escrita por **Karnizero** en enero de 2006; las capturas de pantalla también son suyas. Unos meses más tarde yo, **Eye del Cul**, añadí la segunda parte (“Cómo compilar JFDuke3D”) y el anexo. Revisado en enero de 2014 para corregir enlaces rotos.

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*!