



Juegos en GNU/Linux:  
**Duke Nukem 3D**  
(usando el puerto de [Icculus.org](http://Icculus.org))

Karnizero

## 1. Introducción

De los diferentes *source-ports* de *Duke Nukem 3D* para GNU/Linux, el publicado en *Icculus.org*<sup>1</sup> es el más viejo, básico y experimental. Experimental no en el sentido de “estamos añadiendo montones de características innovadoras”, sino de “éste es nuestro trabajo, aún no lo hemos refinado, y seguramente tendrá muchos fallos que corregir”.

Con esto no quiero decir que no sea usable —lo es, y permite jugar a *Duke Nukem 3D* sin mayores inconvenientes—, sino que en algunas máquinas puede dar problemas y existen alternativas mejores. ¿Por ejemplo? JFDuke3D —sobre el que escribimos en *Arcades 3D* un [completo tutorial](#)—, o bien su sucesor, Eduke32.

Este puerto de Icculus tiene la ventaja, eso sí, de funcionar en un gran número de sistemas diferentes. La lista no se limita a Windows y Linux en PCs clónicos, sino que abarca Solaris, FreeBSD, Mac OS X y BeOS, y las máquinas basadas en los procesadores PowerPC.

En fin, si a pesar de sus inconvenientes decidís probarlo, seguid leyendo.

## 2. Instalando el juego en GNU/Linux

Aquí explicaré cómo hacer funcionar el *Duke Nukem 3D* en GNU/Linux. Y como siempre, comenzaremos con los requisitos, que en este caso son muy pocos:

### 2.1. Requisitos

- Necesitaremos los paquetes **SDL**, **SDL\_mixer**, **SDL\_net**, **SDL\_mixer\_dev** y **SDL\_net\_dev**. Los “dev” nos servirán para compilar el código fuente que nos descargaremos más adelante.<sup>2</sup>
- Tarjeta de sonido compatible con OSS/ALSA.
- Entorno gráfico X (Xfree86/Xorg).

---

<sup>1</sup>Véase <http://www.icculus.org/duke3d>.

<sup>2</sup>En muchas distribuciones los paquetes “dev” no existen porque su contenido ya se incluye en los paquetes SDL correspondientes. También podéis descargaros las librerías SDL de su sitio web oficial —<http://www.libsdl.org>— y compilarlas vosotros mismos en vuestro sistema.

- GCC 3.x (o sea, GCC 3.0 o superiores). También sirven las versiones más modernas de la serie 4.<sup>3</sup>
- Código fuente de *Duke Nukem 3D* y el motor Build (que podéis descargar de *Arcades 3D*)<sup>4</sup>

Y en principio, esto es todo.

Ahora, también necesitaremos algunos archivos del *Duke Nukem 3D* original, a saber:

- `game.con`
- `defs.con`
- `user.con`
- `duke3d.grp`
- `duke3d.cfg`
- `build.cfg`

Estos ficheros tienen que ser de la edición atómica. Con los de la versión normal de *Duke Nukem 3D* también lo he probado, pero no funciona.

## 2.2. Configuración e Instalación

Bien, abrimos una sesión de consola, y creamos una carpeta que se llame `juegos`.

```
$ mkdir juegos
```

Y dentro de ella, otra llamada `duke3d`.

Es recomendable, tanto si usáis Linux como Windows, que tengáis todos los juegos —y, en general, todos los programas de una misma categoría— almacenados en el mismo directorio.

---

<sup>3</sup>Por ejemplo, en enero de 2014, haciendo pruebas, he visto que el código puede compilarse con GCC 4.6.4.

<sup>4</sup>[http://www.arcades3d.net/linux/archivos/duke3d\\_icculus\\_01-2014.tar.bz2](http://www.arcades3d.net/linux/archivos/duke3d_icculus_01-2014.tar.bz2).

```
$ cd juegos
$ pwd
/home/alberto/juegos
$ mkdir duke3d
```

Y a continuación, copiaremos todos los archivos que he citado antes, en los requisitos, a este directorio (en este ejemplo sería `/home/alberto/juegos/duke3d`).

### 2.2.1. Obtener el código fuente

Una vez hecho esto, nos bajaremos el código fuente del juego desde Internet. Seguramente sabréis que 3D Realms ha liberado el código fuente de *Duke Nukem 3D* y *Shadow Warrior*, así que lo podremos descargar sin problemas. El único contratiempo es que dicho código está escrito para ser compilado con un compilador llamado Watcom C++, el cual ya **no** es gratuito. Watcom C++ es un lenguaje de programación basado en C++; resulta casi igual a éste, excepto por una serie de librerías y componentes, que fueron utilizados por 3D Realms para compilar el juego.

Así que si nos bajásemos el código fuente original no nos serviría de mucho. Y de esto mismo también se dieron cuenta en [Icculus.org](http://Icculus.org); así que revisaron todo el código para que pudiera ser compilado con un compilador normal de C++.

Por tanto, lo que tendremos que descargarnos es el código fuente de Icculus. Podéis conseguirlo aquí mismo, en [Arcades 3D](http://Arcades 3D), o bien en [Icculus.org](http://Icculus.org).

En el segundo caso, antes necesitábamos algún programa para acceder a directorios CVS, pero ahora los autores utilizan [Subversion](http://Subversion) para alojar este código fuente, por lo que tendremos que usar un cliente SVN. La mayoría de las distribuciones actuales de GNU/Linux cuentan con alguno, o al menos lo tendrán disponible para instalar desde sus respectivos repositorios de paquetes.

Ya con un cliente SVN instalado en nuestro sistema, abriremos una sesión de consola, y teclearemos lo siguiente (o bien, cortar-pegar):

```
$ mkdir fuentes
$ cd fuentes
$ svn co svn://svn.icculus.org/duke3d/trunk duke3d
```

Con esto crearemos un directorio llamado `fuentes` y comenzaremos a descargarnos el código fuente desde el repositorio SVN —cosa que puede llevar su tiempo, sobre todo a la gente con conexiones lentas; en total son más de diez megabytes—.

En fin, si tenéis problemas con el SVN, podéis echar una ojeada a [la página de Duke Nukem 3D en Icculus](#), cuyas instrucciones pueden servir de ayuda.

### 2.2.2. Compilar el código

Lo hayamos descargado de aquí o de allá, comprobaremos que hay varias carpetas, entre las cuales encontraremos una denominada `source`, que contiene el código que debemos compilar. Si todo está bien, procederemos a la compilación.

Así que listamos el contenido de la carpeta `source`:

```
$ pwd
/home/alberto/juegos/duke3d/fuentes/duke3d/source
$ ls
.
..
...
Makefile ← Fijaos en la existencia de este archivo.
....
.....
.....
```

Siempre que aparezca un `makefile`, entenderemos que estamos ante un código que se puede compilar, a no ser que exista otro archivo llamado `configure`. En este último caso, ejecutaríamos primero el `configure`, y luego compilaríamos.

Como aquí sólo tenemos el `makefile`, procedemos a compilar directamente:

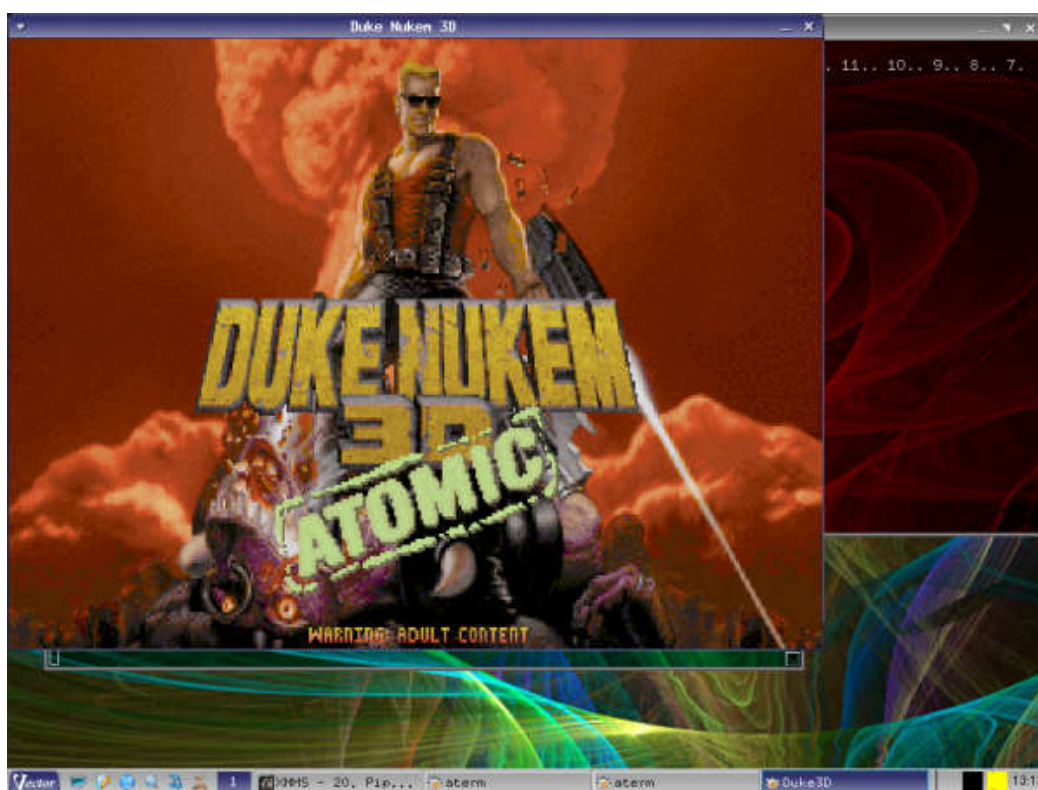
```
$ make
.
.
.
.
$
```

Y si no surgió ningún error de compilación, ya tenemos el *Duke Nukem 3D* compilado, con los ejecutables correspondientes. En la carpeta `source` encontraremos el ejecutable para arrancar el juego —`duke3d`—, y dentro de `buildengine` el editor de mapas `Build`. Para jugar, bastaría con colocar estos dos archivos en la carpeta donde hemos dejado los ficheros `.con`, `.grp` y `.cfg` del Duke Nukem original, mencionados al principio.

### 3. Capturas de Pantalla

Y como siempre, un par de capturas de pantalla para adornar este tutorial... Si a alguien le pica la curiosidad, están tomadas en Vector Linux corriendo en un Pentium II a 350 MHz. En esta máquina el juego funciona sin inconvenientes.

Aquí vemos el *Duke Nukem Atomic* cargando en una ventana del escritorio IceWM —por defecto el juego se ejecuta a pantalla completa, pero pulsando [ALT] + [INTRO] podéis convertirlo en una ventana del escritorio—:



Y con una partida recién comenzada; todo dukero que se precie debería reconocer el nivel:



## 4. Créditos

Este manual fue escrito por **Karnizero** en 2005. Las capturas de pantalla y la introducción —esta última añadida en marzo de 2009— son de **Eye del Cul**. Revisado en enero de 2014.

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<sup>A</sup>T<sub>E</sub>X.

¡Visita *Arcades 3D*!