



Destripando Tomb Raider 1 desde Linux

fanta <fanta@56k.es>



Sumario

Destripando Tomb Raider 1 desde Linux.....	1
Descargar el juego y disfrutarlo con dosbox-x.....	3
Los archivos PHD del juego Tomb Raider.....	4
Trabajando con el archivo LEVEL1.PHD.....	5
Comprobar el número de Tiles de texturas.....	6
Extraer los tiles de texturas.....	7
Script en bash para extraer las tiles de texturas de un PHD.....	8
Destripando los vídeos de transición.....	10
Abrir los objetos 3D del juego.....	14
Ver bien con dosbox Tomb Raider.....	16
Cambiar en el juego GAME OVER por un FUCK YOU.....	19

Es posible jugar a **Tomb Raider I** en GNU/Linux y Windows usando **OpenTomb**.

El problema de jugarlo utilizando opentomb es que el pixel gordo desaparece. Digamos que no estamos jugando al original pixelado a tope, las superficies dejan de tener ruido para pasar a ser lisas y muchas texturas han sido cambiadas para utilizar una con mayor resolución.

Aunque todo eso está bien sinceramente **a mi me gusta el Tomb Raider tal y como se jugaba en MSDOS, es decir, dejándote la vista.**



Descargar el juego y disfrutarlo con dosbox-x

El juego puede descargarse desde aquí:

http://fanta.56k.es/games/1990/1996-Tomb_Raider_I/Tomb_Raider_I.tar.gz

El tema es que con **dosbox-x** (no confundir con el dosbox tradicional) aparte de jugar a **Tomb Raider I** en GNU/Linux vamos a poder hacer más cositas chulas (el modo **debug de Dosbox** es buena cosa).

Para **descargarlo y jugarlo** entonces:

```
$ wget "http://fanta.56k.es/games/1990/1996-Tomb_Raider_I/Tomb_Raider_I.tar.gz"
$ tar xfvz Tomb_Raider_I.tar.gz
$ cd TOMB1
$ dosbox-x -c "mount c: ." -c "imgmount d tr.cue -t iso" -c "keyb sp" -c "cls" -
c "c:" -c "cd tombraid" -c "tomb"
```



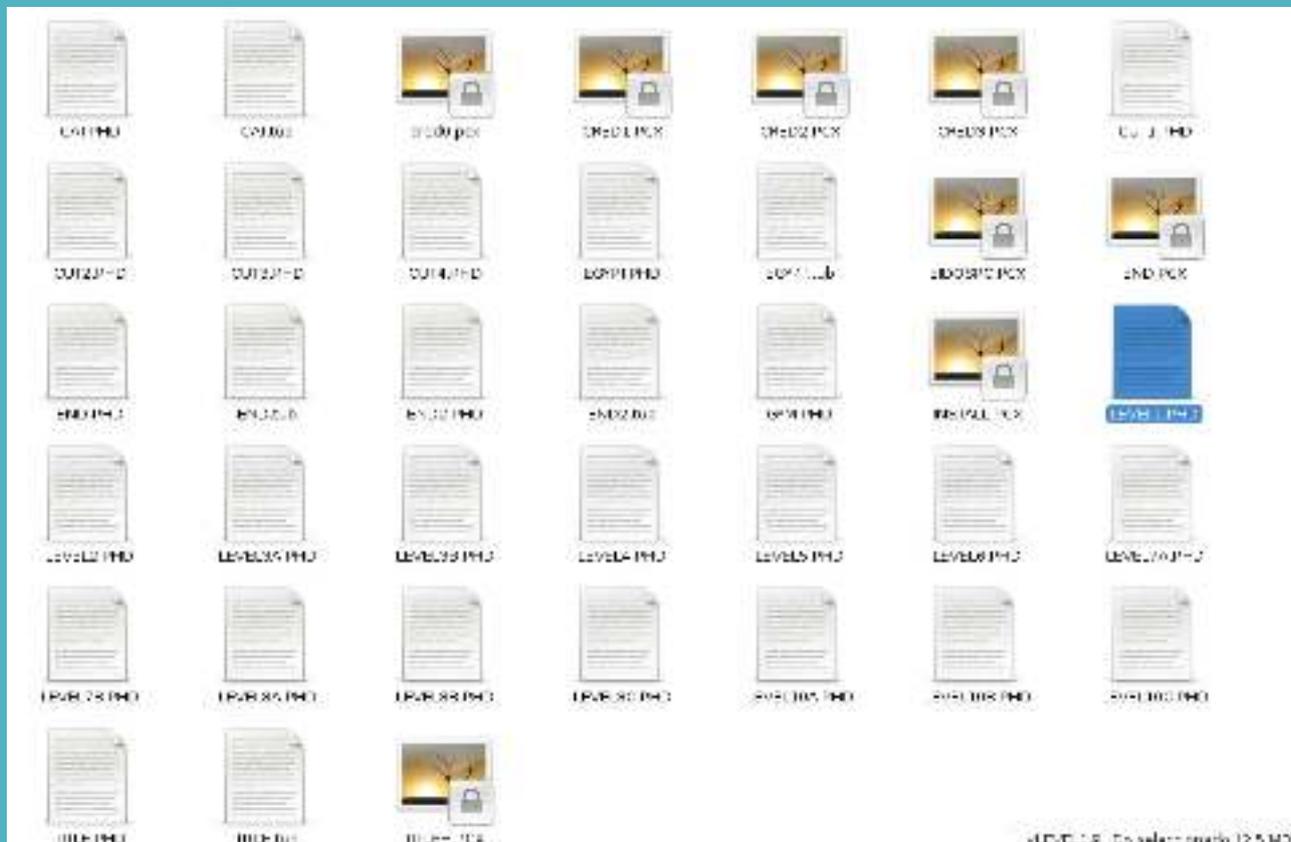
Y con eso ya podemos disfrutar de **Tomb Raider I en GNU/Linux** como si estuviéramos en e1996.

¿Estamos locos?

No estamos locos. Sabemos lo que queremos. Y **lo que queremos es tener a mano Tomb Raider I ORIGINAL para ir destripando ese juego poco a poco.**

Los archivos PHD del juego Tomb Raider

Los archivos **PHD** en el juego Tomb Raider I son archivos que contienen los diferentes niveles así como información relativa a esos niveles.





Y eso está muy bien. Pero no nos dice mucho sobre este tipo de archivos salvo que contienen el mapa, texturas y lo necesario para jugar el nivel con un engine adecuado.

Trabajando con el archivo LEVEL1.PHD

Para ir analizando que contenido tienen estos archivos es bueno saber en que posición se guarda la información.

Para ello voy a usar el archivo LEVEL1.PHD del Tomb Raider I .

Comenzamos con ello si gustas.

Comprobar la versión (4 bytes)

Los primeros 4 bytes son para almacenar la versión.

Utilizando **el comando dd** vamos a poder visualizar desde el byte 0 hasta el 4. Así por ejemplo

```
$ dd skip=0 count=4 bs=1 status=none iflag=skip_bytes,count_bytes if=LEVEL1.PHD | hd | head -1
```

El resultado en Hexadecimal será: 20 00 00 00

Lo cual nos indica que **la versión es 32 en decimal**. El caso es que esto tampoco nos dice demasiado. Nos es de utilidad conocerlo para saber que **los primeros 4 bytes son para la versión** y poco más.

Comprobar el número de Tiles de texturas

Los siguiente 4 bytes son para almacenar el número de «TextTiles». El número de tiles de texturas que contiene el LEVEL1.PHD.

Tiles son agrupaciones de texturas. **Una traducción de tiles al castellano es «azulejos»**. Nos quedamos con que tenemos grupos de texturas.

Ejemplo del primer tile que vamos a extraer:



Ejemplo del segundo tile que vamos a extraer:



¿Cuántos tiles tiene el LEVEL1.PHD? 11.

Y se que tiene 11 ya que los siguientes 4 bytes son para indicar eso. Con el comando dd lo podemos ver así:

```
$ dd skip=4 count=4 bs=1 status=none iflag=skip_bytes,count_bytes if=LEVEL1.PHD | hd | head -1
```

```
fanta@debian:~/juegos/TONS1/DATA$ dd skip=4 count=4 bs=1 status=none iflag=skip_bytes,count_bytes if=LEVEL1.PHD | hd | head -1
XXXXXXXX 00 00 00 00
fanta@debian:~/juegos/TONS1/DATA$
```

El valor resultante es **0B** con lo que convertido de HEX a DEC tenemos 11.

Hemos de fijarnos que he puesto skip=4 y count=4. De esa forma saltamos los primeros 4 bytes del archivo (los referentes a versión) y obtenemos los 4 siguientes.

Extraer los tiles de texturas

Ciertamente esos grupos de texturas aunque están en el archivo no sabemos muy bien en que sitio ni posición. Bueno si lo sabemos.

Es lo que vamos a comentar ahora de modo que **relájate un poco** amigo que te noto en tensión.

Cada tile tendrá un máximo de 65536 bytes (son tiles de texturas con paletas de color de 8-bits).

Por tanto las posiciones si sabemos que existen 11 son las siguientes:

- 0-4 – La versión.
- 4-8 – Número de tiles.
- 8-65536 El primer tile

Sabemos que **el primer tile ocupará 65536 bytes** pero que comienza saltando 8 bytes iniciales.

El comando por tanto para **extraerlo con dd** es:

```
$ dd skip=8 count=65536 bs=1 status=none iflag=skip_bytes,count_bytes if=LEVEL1.PHD > /tmp/tile1-level1
```

Para sacar el contenido del **tile 2**:

```
$ dd skip=65544 count=65536 bs=1 status=none iflag=skip_bytes,count_bytes if=LEVEL1.PHD > /home/fanta/tile2-level1
```

Si nos fijamos count siempre será para extraer esto de 65536 pero skip (los bytes que saltamos para comenzar en esa posición) ya no es 8. Ahora skip es de (8+65536).

Para el tile3, tile4, ... tile11 iremos variando skip y sumando 65536 cada vez. **De esa forma deslizamos la ventana.**

Y como no vamos a estar todo el rato pensando **para eso vamos a crearnos un script** que en base a un PHD dado saque el número de tiles y las extraiga numeradas.

Un script en bash. Lets go !

Script en bash para extraer las tiles de texturas de un PHD

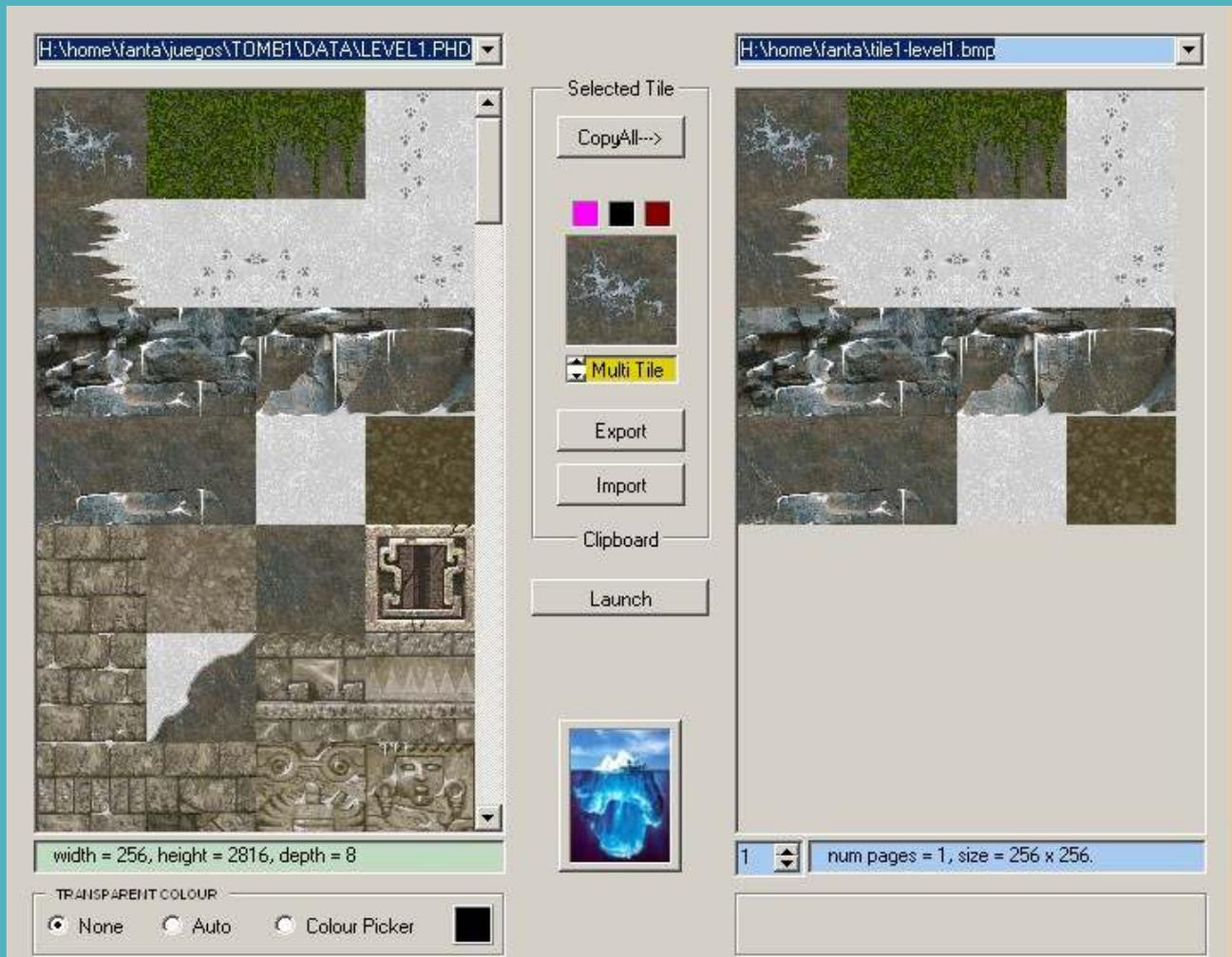
```
#!/bin/bash
phdFile=$1
nTexTilesHex=$(dd skip=4 count=4 bs=1 status=none iflag=skip_bytes,count_bytes if=$phdFile | xxd -ps -l 1)
nTexTilesDec=$(echo $(( 16#$nTexTilesHex )))
echo "$nTexTilesDec Tiles de texturas en $phdFile"
skipBytes=8
echo "---"
for i in `seq 1 $nTexTilesDec`;
do
    echo "Sacando Tile $i /tmp/$phdFile.tile$i"
    dd skip=$skipBytes count=65536 bs=1 status=none iflag=skip_bytes,count_bytes if=$phdFile > /tmp/$phdFile.tile$i
    skipBytes=$(expr $skipBytes + 65536)
    echo "$skipBytes"
done
```

```
fanta@debian:~/juegos/TOMB1/DATA$ bash extract_tiles.sh LEVEL1.PHD
11 Tiles de texturas en LEVEL1.PHD
---
Sacando Tile 1 /tmp/LEVEL1.PHD.tile1
65544
Sacando Tile 2 /tmp/LEVEL1.PHD.tile2
131080
Sacando Tile 3 /tmp/LEVEL1.PHD.tile3
196616
Sacando Tile 4 /tmp/LEVEL1.PHD.tile4
262152
Sacando Tile 5 /tmp/LEVEL1.PHD.tile5
327688
Sacando Tile 6 /tmp/LEVEL1.PHD.tile6
393224
Sacando Tile 7 /tmp/LEVEL1.PHD.tile7
458760
Sacando Tile 8 /tmp/LEVEL1.PHD.tile8
524296
Sacando Tile 9 /tmp/LEVEL1.PHD.tile9
589832
Sacando Tile 10 /tmp/LEVEL1.PHD.tile10
655368
Sacando Tile 11 /tmp/LEVEL1.PHD.tile11
720904
fanta@debian:~/juegos/TOMB1/DATA$
```

Y si quisiéramos hacer esto mismo pero con una **herramienta gráfica** pues esa es **TBuilder 2.3**.

Descargar **Tbuilder** desde aquí: <https://www.aspidetr.com/en/tools/tbuilder/texture-builder-2-3-1/>

Con **wine** se puede ejecutar esta herramienta e indicarle un PHD para que extraiga los tiles de texturas. Puedes luego **convertirlo a otros formatos como por ejemplo BMP**.



Y aunque al final la gente termina por usar esas herramientas (terminamos haciendo eso todo cristo) es bueno saber como funcionan.

Sabiendo como funcionan podemos crear otras para línea de comandos, para otros sistemas operativos, etc...



Destripando los vídeos de transición

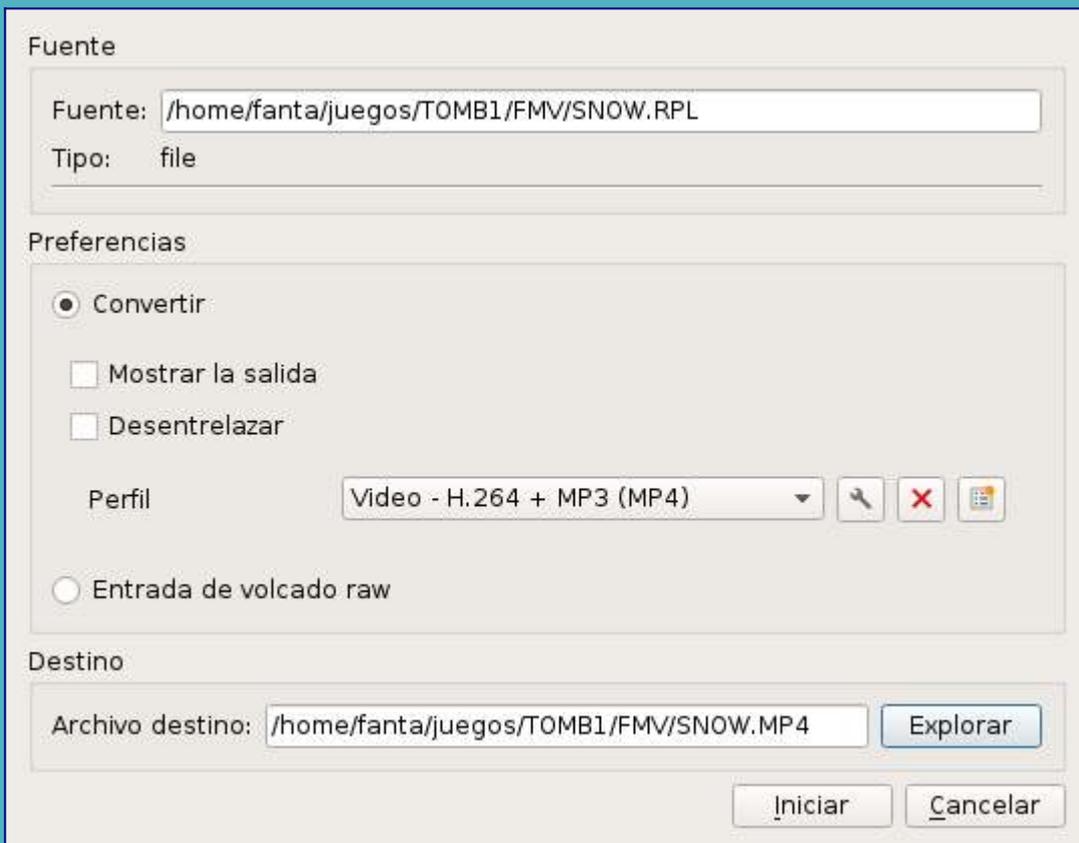
En el juego vamos a encontrar **en el directorio FMV** archivos como los que se ven en la siguiente captura:



Todos tienen una extensión RPL. Este formato es un formato que lleva el audio sin comprimir. Eso quiere decir que ya solamente el audio de esos vídeos de transición (la intro, el final, etc...) **es posible extraerlo usando Audacity.**



Y con VLC es posible reproducir esos vídeos (y por tanto convertirlos).





Y es bueno saber que es posible exportarlos a MP4 por ejemplo con otros codecs. De esa forma podríamos alterar los vídeos con un editor de vídeo.

Y eso está bien.

Podemos abrir los vídeos y verlos (ver END.RTL no es recomendable ya que es el final del juego). Usan **un codec privativo** desarrollado por **EIDOS**.

He mirado y en un foro he encontrado a un tipo que buscaba lo que yo: **poder convertir de un formato a RTL un vídeo.**

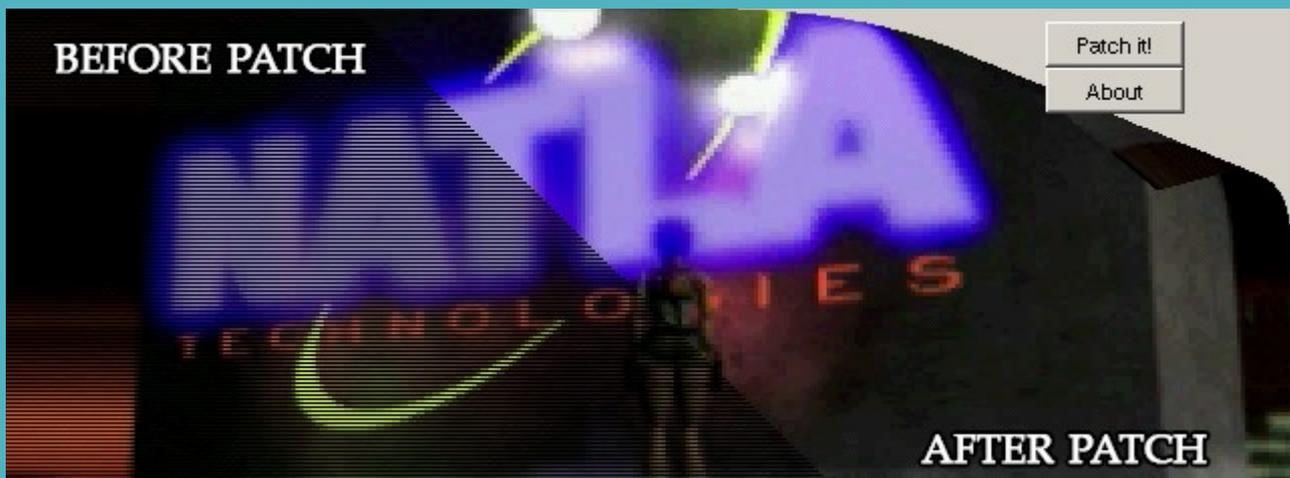
Y acompañaba una imagen de un software para windows que al parecer lo hace. Lo interesante es que te suelta esa captura el nombre del codec: **ESCAPE 2.0 Eidos video v130**



Y EIDOS al parecer no han publicado nunca detalles sobre este codec y yo no he encontrado ese software :(.

Lo más que he encontrado son algunos detalles aquí: wiki.multimedia.cx/index.php?title=Escape_124

Sea como sea si que existe un parche para mejorar el desentrelazado de los vídeos: www.core-design.com/community_ultimatetr1patch.html



Aunque **seamos sinceros**. Lo normal es darle a Escape y pasar esas intros e interludios ya que lo mismo ya los hemos visto antes o mil veces.

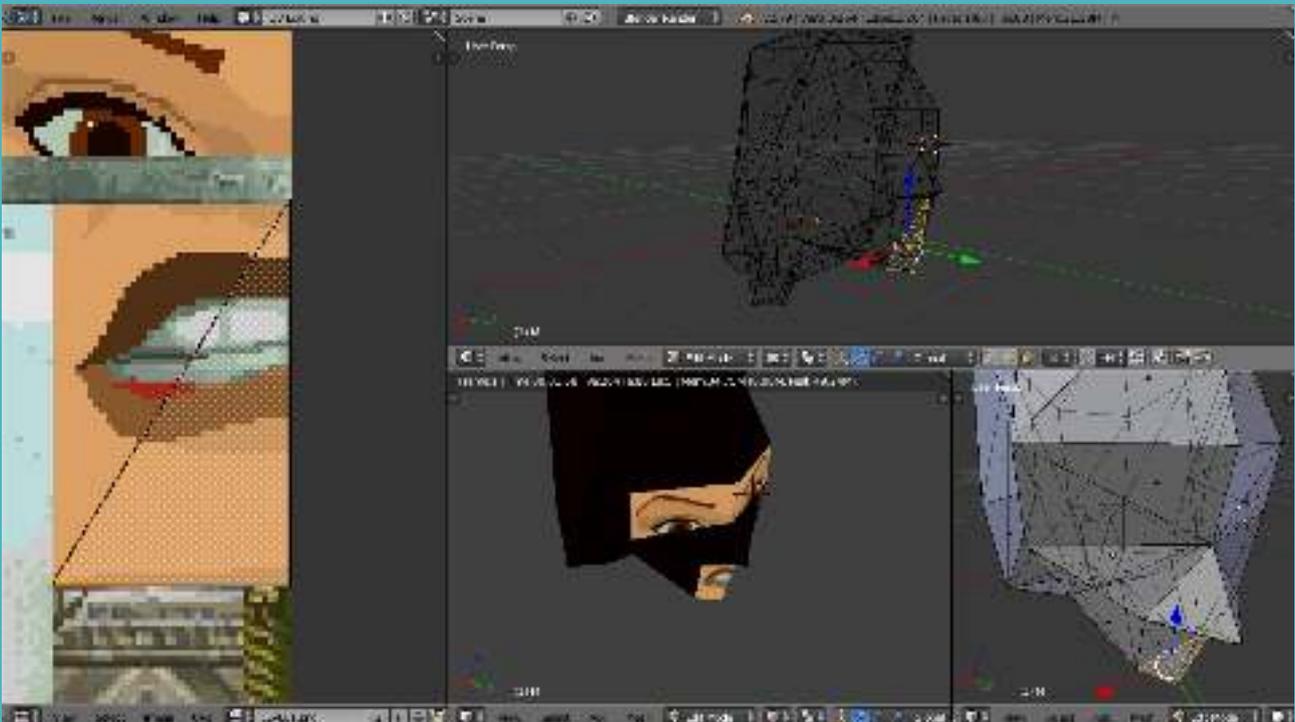
Abrir los objetos 3D del juego

Una vez tenemos destripadas las texturas **es bueno saber como destripar los objetos 3D.**

Esos objetos en Tomb Raider van desde puertas, palancas, murciélagos, lobos, osos, pistolas, puentes, ratas, ... a muchísimos otros cientos de objetos como gafas, un walkman, unos cascos, ...

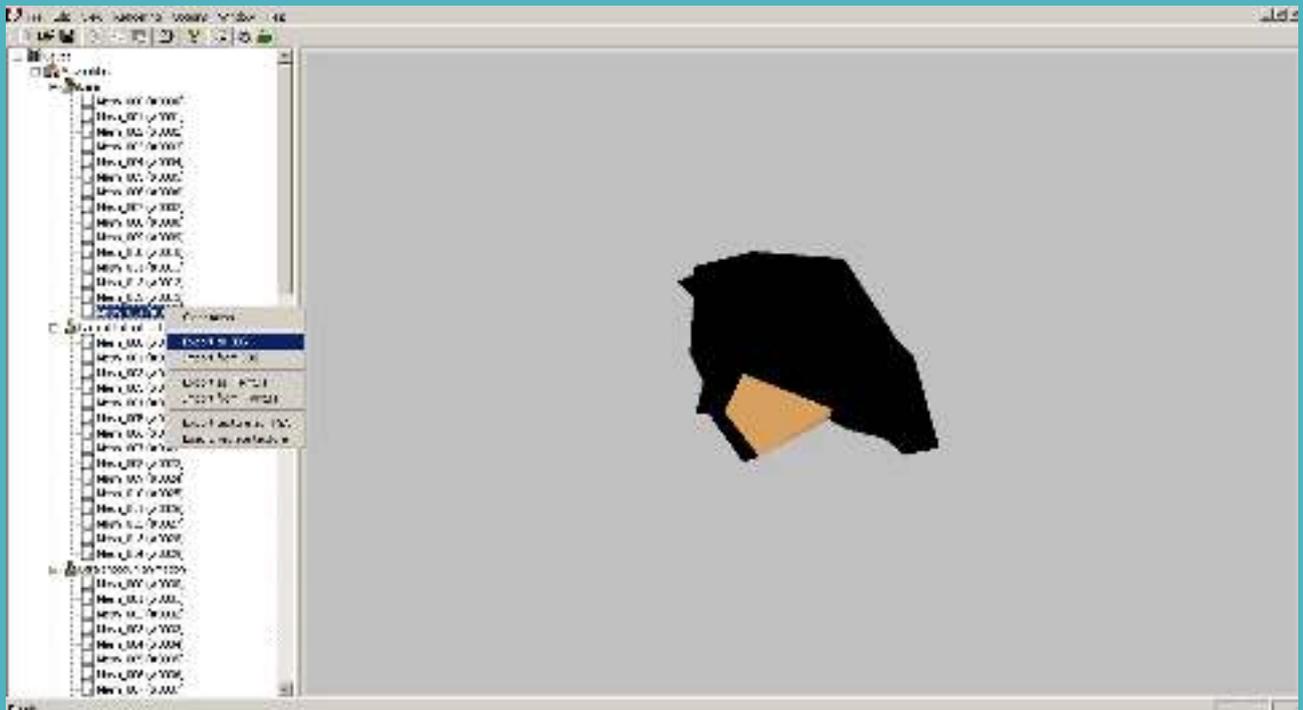
La finalidad de destripar las texturas es poder usarlas. La finalidad de extraer los objetos lo mismo. **¿Con que uniremos ambas cosas? Con Blender.**

Aquí **un ejemplo** esta mañana mapeando la cabeza de Lara:



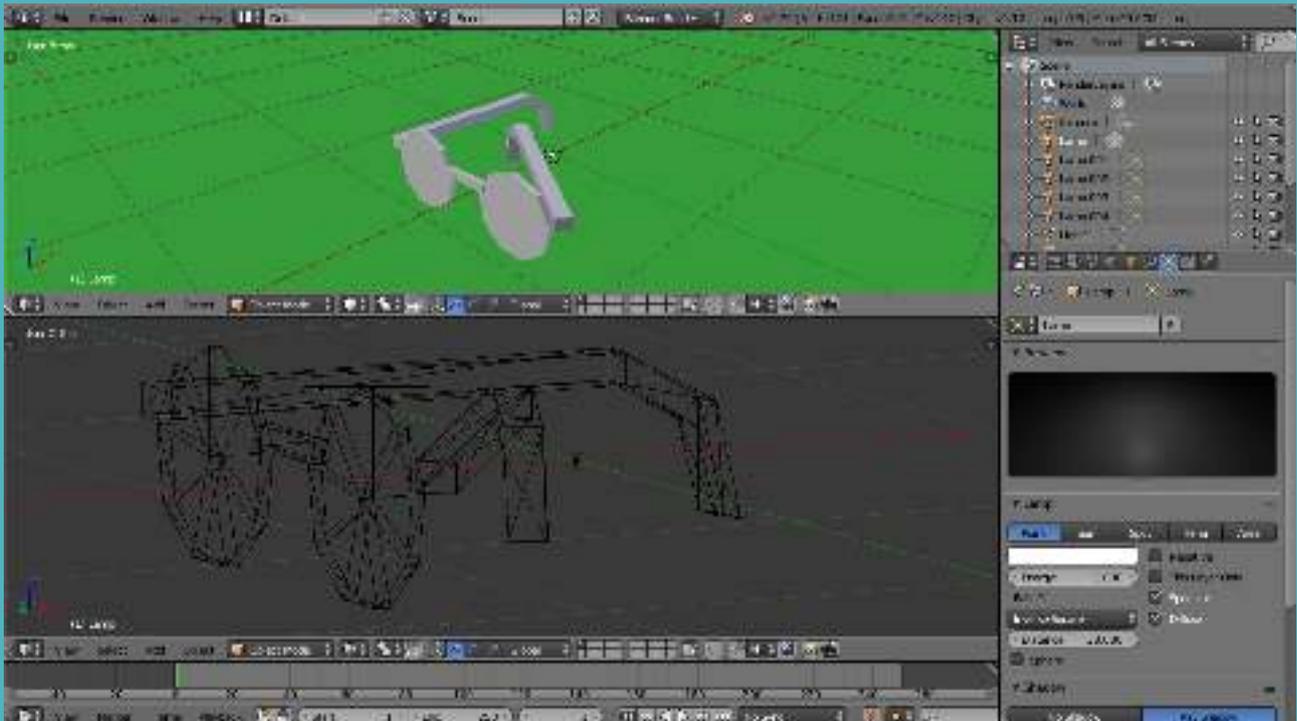
Exportar las mayas de un PHD (un archivo que contiene texturas, sonidos, objetos 3d, ... de un nivel en Tomb Raider I) se puede hacer de forma sencilla con el software: [Trview-x64.zip](#)

TRView puede abrirse con WINE desde GNU/Linux y una vez se abre se puede indicar un PHD (por ejemplo LEVEL1.PHD que es el nivel 1).



Una vez exportado a 3ds se **puede importar en Blender sin problemas.**

De esta forma podemos extraer los objetos 3D del juego para mapearlos con las texturas que queramos, adaptarlos, modificarlos, etc...



Resumen

Se podría decir que el material como texturas, objetos 3D, sonidos, vídeos, ... de Tomb Raider se puede extraer del juego original y en concreto de los archivos PHD del directorio DATA.

Se puede afirmar que puede hacerse desde GNU/Linux sin problemas o sin no muchos.

Ver bien con dosbox Tomb Raider

Para disfrutar Tomb 1 de la forma más cómoda es bueno toquetear un poco las opciones. Algo que va muy bien es antes de ejecutar dosbox-x **poner la resolución del escritorio a 800×600**.

Luego se ejecuta así por ejemplo:

```
dosbox-x -c "mount c: ." -c "imgmount d tr.cue -t iso" -c "keyb sp" -c "cls" -c "c:" -c "cd tombraid" -c "tomb"
```

Algo que suele funcionar bien para evitar parpadeos es activar el [«Double Buffering \(Fullscreen\)»](#).



Luego ya en el juego para conseguir un mejor aspecto se pulsa **F1 para cambiar a SVGA o a VGA**. Y si en SVGA nos va algo lento con F2 reducimos tamaño de pantalla o subimos con F3 si nos pasamos.

Puede verse la **diferencia de calidad** en estas 2 capturas siguientes:



Cambiar en el juego GAME OVER por un FUCK YOU

Una vez arrancamos el juego, como he dicho antes, podemos **a golpe de click de ratón darle a debugger** o usar **ALT+z** izquierdo + **pause**.

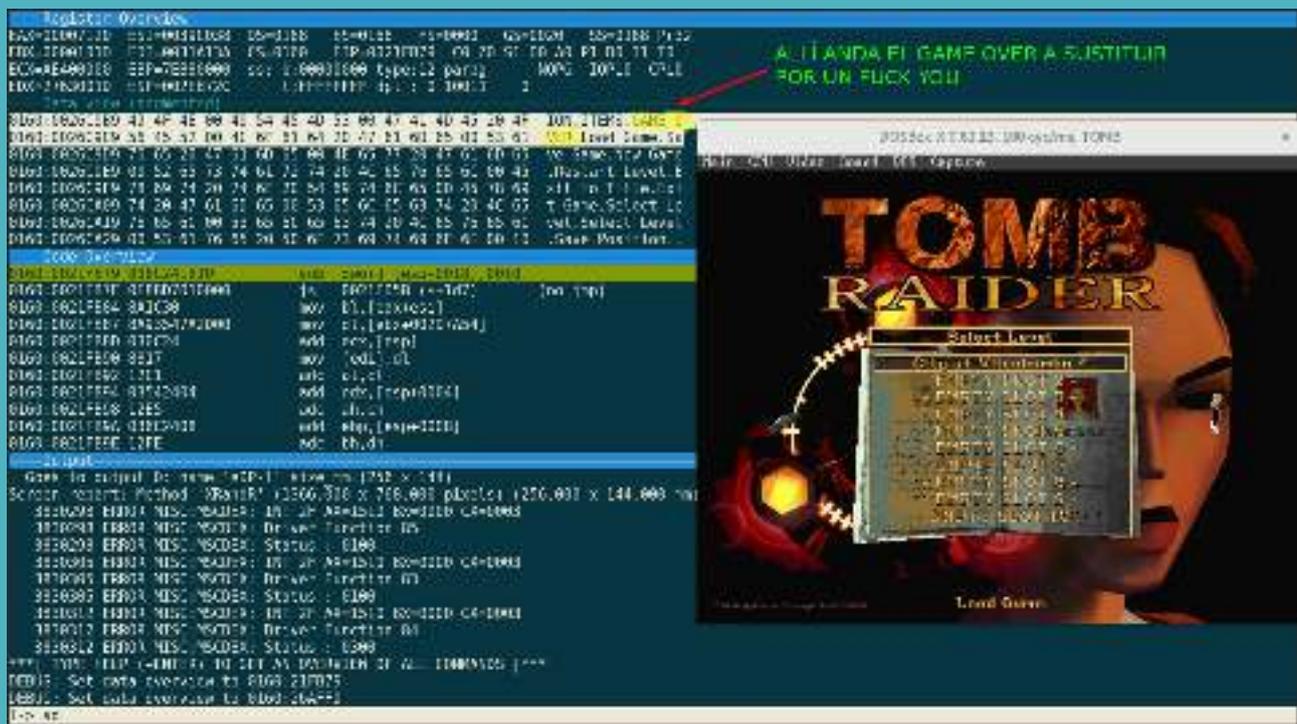
Para **ir directamente a la mandanga** en el debugger podemos escribir:

```
d 0160:0026AFF9
```

Per si no te hubiera dicho en que sitio está **lo normal es que indiques el registro «CS» y «EIP»** para posicionarte.

```
d CS:EIP
```

Pero no lo hagas ya que te irás bastante lejos y tendrás que ir dando para abajo bastante rato hasta llegar desde el 0021FB79 al 0026AFF9 en la memoria.



Alli nos encontramos las palabras **GAME OVER** en 2 líneas. Lo cual nos obligará a copiarlos esos 2 churros en HEXADECIMAL de ambas líneas:

En la Posición: **0160:0026C9B9**

```
49 4F 4E 00 49 54 45 4D 53 00 47 41 4D 45 20 4F
```

En la posición **0160:0026C9C9**

```
56 45 52 00 4C 6F 61 64 20 47 61 6D 65 00 53 61
```

Vamos a sustituir las 2 líneas, la zona que pone GAME OVER, por otras que incorporen FUCK YOU.

FUCK YOU convertido de ASCII a HEX es 46 55 43 4b 20 59 4f 55

El 20 es el espacio.

Aquí un conversor de ASCII a HEX y de HEX a ASCII www.rapidtables.com/convert/number/hex-to-ascii.html

Y para hacer eso usaremos SM (SET MEMORY) en el Debugger:

```
SM 0160:0026C9B9 49 4F 4E 00 49 54 45 4D 53 00 46 55 43 4b 20 59
SM 0160:0026C9C9 4F 55 20 00 4C 6F 61 64 20 47 61 6D 65 00 53 61
```

Y se tendría que ver como esto:

```
Register Overview
EAX=00007120  ESI=0039C038  DS=0168  ES=0168  FS=0000  GS=0020  SS=0168 Pr32
EBX=0000101D  EDI=0031A13A  CS=0160  EIP=0021FB79  C0 Z0 S1 00 A0 P1 D0 I1 T0
ECX=AE400000  EBP=7EB80000  ss: b:00000000 type:12 parbg  NOPG IOPL0 CPL0
EDX=2763001D  ESP=002EB72C  l:FFFFFFFF dpl : 0 10011  0

Data view (segmented)
0160:0026C9B9 49 4F 4E 00 49 54 45 4D 53 00 46 55 43 4b 20 59  ION.ITEMS.FUCK YOU
0160:0026C9C9 4F 55 20 00 4C 6F 61 64 20 47 61 6D 65 00 53 61  DU. Load Game.Save
0160:0026C9D9 76 65 20 47 61 6D 65 00 4E 65 77 20 47 61 6D 65  ve Game.New Game
0160:0026C9E9 00 52 65 73 74 61 72 74 20 4C 65 76 65 6C 00 45  .Restart Level.E
0160:0026C9F9 78 69 74 20 74 6F 20 54 69 74 6C 65 00 45 78 69  xit to Title.Exi
0160:0026CA09 74 20 47 61 6D 65 00 53 65 6C 65 63 74 20 4C 65  t Game.Select Le
0160:0026CA19 76 65 6C 00 53 65 6C 65 63 74 20 4C 65 76 65 6C  vel.Select Level
0160:0026CA29 00 53 61 76 65 20 50 6F 73 69 74 69 6F 6E 00 10  .Save Position..

Code Overview
0160:0021FB79 836C241810 sub dword [esp+0018],0010
0160:0021FB7E 0F88D7010000 js 0021FD5B ($+1d7) (no jmp)
0160:0021FB84 8A1C30 mov bl,[eax+esi]
0160:0021FB87 8A93547A2D00 mov dl,[ebx+002D7A54]
```

Luego con F5 en el debugger seguimos, jugamos una partida y nos dejamos morir. Por ejemplo ahogados en el agua.



Y al morir veremos el **FUCK YOU**



¿Mola? **Si, mola mucho.** Ya estás pensando en buscar en que parte está GAME OVER en los archivos para cambiarlo a ver si puedes distribuir un TOMB RAIDER con estos cambios ya permanentes :).

Y con esto terminamos este rollo sobre formas de destripar Tomb Raider I.

Espero que fuese de tu agrado.

Nota:

```
dosbox-x -set fullscreen=true -set fullresolution=1920x1080 -set  
windowresolution=1920x1080 -set cputype="Pentium" -set  
output=surface -set scaler="normal2x forced" -c "mount c: ." -c  
"imgmount d tr.cue -t iso" -c "keyb sp" -c "cls" -c "c:" -c "cd  
tombraid" -c "tomb"
```

Eso va bien para jugarlo a pantalla completa. Luego con F1 (en portatiles quizás fn + F1) se cambia a VGA.