

MK1 - 32X

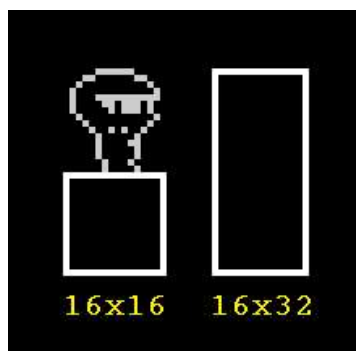
Tutorial para usar sprites de 16x32 píxeles.

1. INTRODUCCIÓN

Esta es una versión de la ya clásica Churrera de los Mojon Twins pero con un añadido para poder usar sprites más grandes de los que, por defecto, soporta el motor.

Hemos añadido unas funcionalidades extras en el archivo "config.h" muy fáciles de usar dejando el resto del motor tal y como lo conocéis, de manera que no te resultará difícil trabajar con él si ya lo has usado anteriormente o si ya te has leído el manual del MK1 Churrera oficial.

El uso de estos sprites te permitirá tener personajes más detallados a nivel gráfico:



Ahora te caben 2 Ninjajares de alto

Así que ve pensando como de shulo va a ser tu prota y sus enemigos, porque si en 16 píxeles se pueden hacer maravillas, en 32x no te digo trigo.

¿Todo esto es gratis?

Si hablamos de lereles, si, es gratis como el propio MK1, claro. Pero mover estos spritacos no es lo mismo que mover la mitad de tamaño, por eso, si buscas un juego de movimientos rápidos, mejor usa la Churrera normal que te dará más fluidez.

¿Tan lento va?

No hombre, no, pero se nota. Por eso te lo advierto, según que proyecto tengas pensado te puede merecer la pena ganar por un lado y sacrificar en otro.

También meter sprites el doble de grandes ocupa, por consiguiente, el doble de memoria, por lo que igual tienes que crear un mapa con menos pantallas de las que podrías en condiciones normales. Cuestión de valorar prioridades. Pero mola tener la opción, ¿no?

2. MODO DE EMPLEO

Verás que no es nada complicado. Solo hay que tocar en un par de archivos para disfrutar de sprites de 32x de alto:

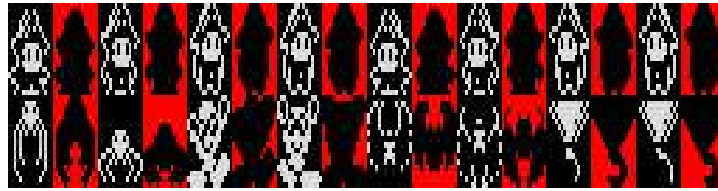
- compile.bat

En este archivo vamos a encontrar un par de líneas donde se convierten los gráficos desde nuestro PNG al formato que requiera la Churrera:

```
27
28 echo Importando GFX
29 ..\utils\ts2bin.exe ..\gfx\font.png ..\gfx\work.png tileset.bin 7
30
31 REM 16x16 Sprites
32 REM ..\utils\sprcnv.exe ..\gfx\sprites.png assets\sprites.h > nul
33 REM ..\utils\sprcnvbin.exe ..\gfx\sprites_extra.png sprites_extra.bin 1 > nul
34
35 REM or NEW 16x32 Sprites
36 ..\utils\sprcnv_32x.exe ..\gfx\sprites32.png assets\sprites.h > nul
37 ..\utils\sprcnvbin_32x.exe ..\gfx\sprites_extra_32.png sprites_extra.bin 1 > nul
38
39 ..\utils\sprcnvbin8.exe ..\gfx\sprites_bullet.png sprites_bullet.bin 1 > nul
40
```

Como ves, están comentadas con *REM* las dos líneas para usar gráficos de 16x16 píxeles, y a continuación van las nuevas. Solo deben estar activas las de 16x32, obviamente. En la carpeta "gfx" encontrarás las plantillas para crear los sprites, que es un archivo PNG justamente el

doble de alto que el típico (256x64), llamado "*sprites32.png*" y que luce tal que así:



Ya ves que no me he complicado la vida y son los mismos sprites del ejemplo que MK1 trae por defecto (Lala Beta) pero estirados hasta ocupar los 32 píxeles de alto. Ahora te toca a tí dibujar un sprite en serio, usando todo el tamaño disponible.

De igual forma, hay un PNG llamado "*sprites_extra_32.png*" que contiene el sprite especial que aparece cuando golpeamos a un enemigo pero el doble de alto, lo mismo.

El motor está listo para compilar y que genere un .tap con estos sprites. Tiene un mapa creado de 10x2 pantallas, así que tendrás que cambiarlo al tamaño de tu mapa una vez lo crees, como siempre.

- config.h

En este archivo he incluido los #defines que necesitas para usar el poder del "32x", que son estos:

```
14
15 // MODE BIG SPRITES (32x16)
16 #define MODE_32X // Use 32x16 Sprites instead of classic 16x16
17 // Background collision
18 #define TOP_BG_COLLISION_NORMAL 8 // Top point for Bounding box, reduce (in pixels) the height of
the collision box for background. eg: 8 means a 24x16 box instead of 32x16
19 // Enemy collision
20 #define BOX_WIDTH 13 // Bounding box width
21 #define BOX_HEIGHT 28 // Bounding box height
22 // Bounding box extras
23 #define BB_VARIABLE_TOP
24 #define TOP_COLLISION_NORMAL 0 // Top points for bounding box, reduce (in pixels) the height
of the collision box for enemies
25 #define TOP_COLLISION_JUMPING 8 // Top points for bounding box when jumping (if you use a
shorter sprite for jumping, adjust the box here)
26
```

Los explico uno a uno, que no son muchos:

- #define MODE_32X: Con éste activamos el modo 32x, o sea que es imprescindible, bueno en realidad todos los son para que funcione bien.
- #define TOP_BG_COLLISION_NORMAL: Este se usa para reducir la altura de la caja de colisiones con el escenario, si por ejemplo, nuestro

personaje no llega a ser 32 píxeles de alto sino un poco menos.

- `#define BOX_WIDTH / #define BOX_HEIGHT`: aquí definimos las dimensiones (Ancho y alto) de la caja de colisiones con los enemigos.

- `#define BOX_WIDTH` valdrá como máximo 16
- `#define BOX_HEIGHT` valdrá como máximo 32

- `#define BB_VARIABLE_TOP`: ésta, junto con las dos siguientes, sirven para tener una altura de la caja de colisiones variable en caso de que al saltar, nuestro personaje encoja y ocupe menos altura que en su posición normal en pié. Podemos dejar la "Normal" en 0 y variar la altura cuando salte ("`TOP_COLLISION_JUMPING`"). En juegos con vista "genital" no tiene sentido esto, porque no saltamos, así que déjalas todas con valor cero.

Y esto es todo, sólo comentar que este tutorial es aplicable para juegos de **48k o 128k** pero sin multinivel, es decir, un solo nivel. Para poder usarlo en modo multinivel, tendrás que seguir el tutorial del propio MK1 Churrera pero con la excepción de convertir los sprites con la utilidad llamada "*sprcnvbin_32x.exe*" en lugar de "*sprcnvbin.exe*", como hemos hecho en la segunda línea nueva que añadimos al *compile.bat*.

Esperamos desde el Pat Morita Team que este "mod" sea de tu agrado y lo aproveches para tus próximas creaciones para el ZX Spectrum.

Pat Morita