

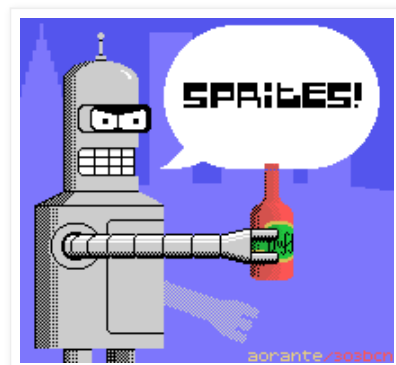
06 abril 2010

Técnicas para pintar imágenes en Screen 2 (2 de 2)

Hay un truco con el que podemos paliar la limitación de colores del screen2, consiguiendo un color más en determinadas zonas de la pantalla. Consiste en utilizar planos de figuras móviles (sprites). Esto es posible gracias a que toda la información de los sprites (gráfica y atributos), se encuentra en la memoria de vídeo (VRAM) y el formato SC2 consiste en un volcado de esta memoria.

Antes tenemos que tener en cuenta una serie de puntos:

1. Solo podemos utilizar 32 sprites de 8x8 o 16x16 pixels y solo podrán situarse 4 en la misma línea, ya que el quinto se borraría o se mostraría parcialmente.
2. Para cada uno de ellos, solo podemos utilizar uno de los 16 colores de la paleta de los MSX.
3. Tendremos que programar una aplicación para situar los sprites sobre la imagen, para luego guardar el fichero SC2.



En este tutorial, explicaré como desarrollar este efecto, utilizando un PC o MAC en combinación con un MSX (físico o emulado). También se puede hacer perfectamente utilizando aplicaciones nativas. Lo importante es conocer la idea y luego aplicarla con las aplicaciones que más nos gusten.

Dibujando los sprites

Empezaremos por dibujar lo que serán los sprites, sobre nuestra imagen. Para ello recomiendo utilizar un programa gráfico que disponga de capas, ya que nos ayudara a distinguirlos de la imagen principal. Utilizaremos una o más capas y en ellas dibujaremos la forma con el color que necesitemos.

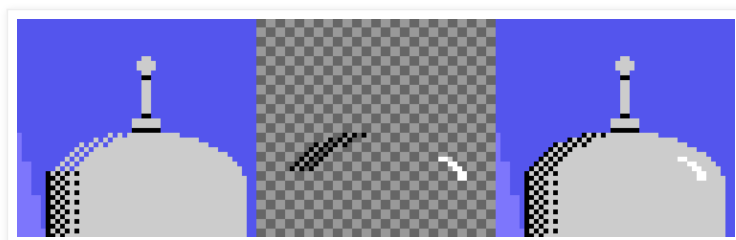


Imagen normal, capa de sprites y combinación de ambas.

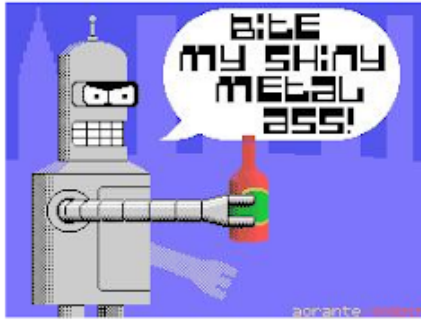
Una vez tengamos terminada la composición, necesitaremos tener por un lado la imagen de fondo en formato SC2 y los sprites en formato código Basic.

Obteniendo el fondo en SC2

Para obtener la imagen del fondo en SC2, utilizaremos el [Screen Conversor](#) de Jannone. Su manejo es sencillo:

1. Cargamos nuestra imagen.
2. Seleccionamos la paleta de los MSX1, inhabilitamos la opción de "Error Diffusion" y pulsamos el botón "Convert".
3. Guardamos la imagen SC2, utilizando la opción Save "all dump", y copiamos el código que muestra, a un editor de texto y lo guardamos con extensión ".BAS".

Step 2: Adjust conversion settings



Conversion:

Scale:

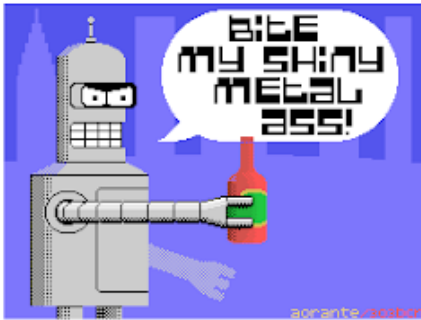
Palette:

Filters:

- ☒ Scale to compensate resolution
- ☐ Error diffusion
- ☐ Remove noise
- ☐ Find edges
- ☐ Make tiles (experimental, removes identical chars)

Note: final image will be cropped to the screen's resolution

Step 3: Download converted image; or go back and try different settings



Download VRAM dump:

This is how you can display it:

```
10 SCREEN 2
20 BLOAD"BENDERSX.SC2",S
30 GOTO 30
```

[Download preview image](#)

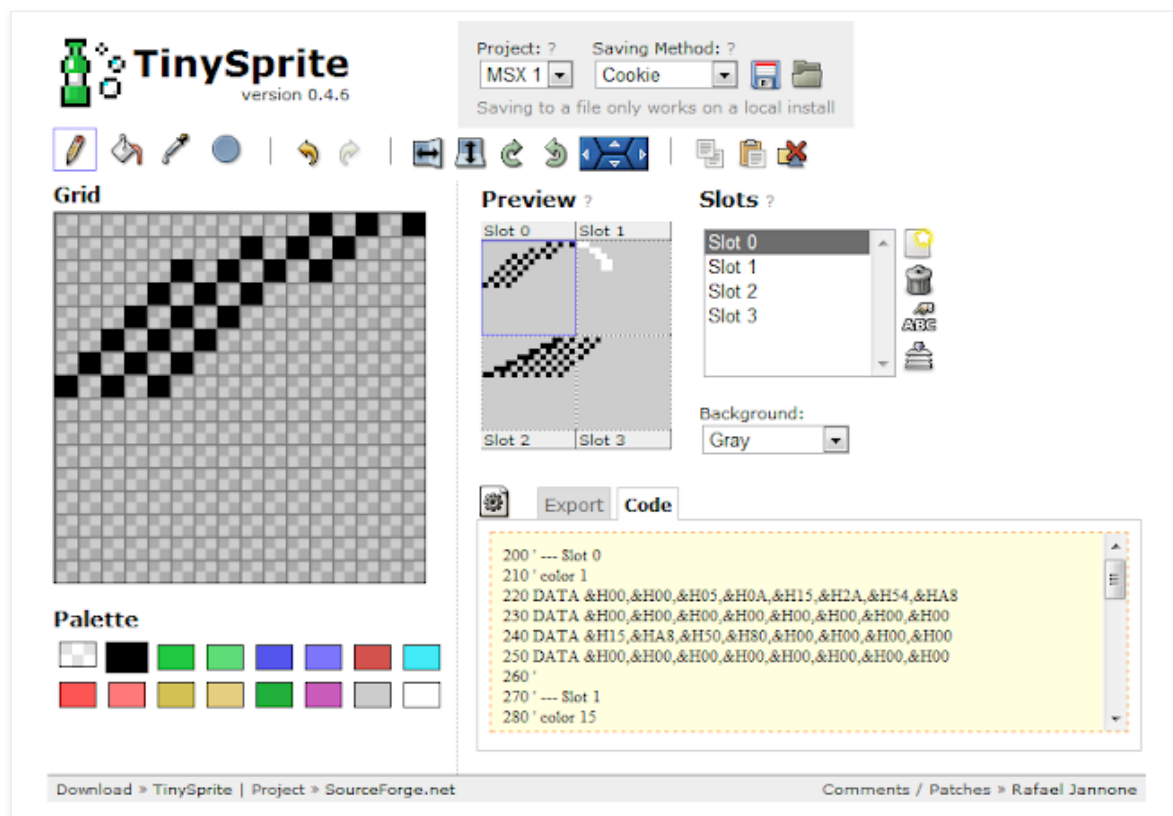
Give us some credit ;)

Powered by msx.jannone.org

El editor de sprites

Para obtener los datos de los sprites, utilizaremos el [TinySprite](#), también desarrollado por Jannone.

Pintaremos todos los sprites, añadiendo tantos "slots" como sprites necesitemos. Utilizaremos como referencia lo que hemos pintado desde nuestro software de dibujo. Cuando terminemos "Exportaremos" el resultado a Basic. Aconsejo cambiar el numero de linea inicial por uno más alto, para reservar unas líneas para añadir más código que necesitaremos. El resultado se nos mostrará cuando pulsemos el icono representado por una hoja con una rueda dentada. Lo seleccionaremos y lo copiaremos al programa Basic guardado en el punto anterior.



A Programar!

Sobre el programa que disponemos, tendremos que añadir unas líneas, para que se comporte como nosotros necesitamos. Incluyo los fuentes del que he utilizado para este tutorial, el cual os servirá para adaptarlo a vuestras creaciones. De todas formas os detallo algunos puntos para que entendáis su funcionamiento.

Primero añadiremos una línea que asigne el color negro al borde de la pantalla. Esto será necesario para que la imagen se visualice correctamente ya que el conversor suele aplicar el negro como el color 0 (transparente), de la paleta del MSX.

```
1 color , , 1
```

Luego indicaremos el modo de pantalla con el tipo de sprites que vamos a utilizar (0=8x8; 1=8x8 ampliado; 2=16x16; 3=16x16 ampliado).

```
10 SCREEN 2, 2
```

Quitaremos la línea "30 goto 30" y en su lugar añadiremos una que lance la función que añade los datos gráficos de los sprites en la memoria de vídeo.

```
30 GOSUB 10010
```

También será necesario añadir como última línea de programa, la función de retorno:

```
10030 RETURN
```

El siguiente paso será mostrar los sprites en la posición adecuada. Para ello utilizaremos la instrucción "PUT SPRITE". Antes necesitaremos saber la posición de cada sprite. Desde el programa de dibujo situaremos el cursor en la posición superior izquierda de cada bloque de 16x16 y apuntaremos las coordenadas (en Gimp se muestran a la izquierda de la barra de estado de la ventana principal). Debemos utilizar la sentencia PUT SPRITE tantas veces como sprites tengamos. Funciona de la siguiente manera:

PUT SPRITE número de plano, (coordenada X, coordenada y), color, número de sprite

Ejemplo:

```
100 PUT SPRITE 0, (33,23), 1, 0
```

Después de esto, guardamos la pantalla (sprites incluidos), con la siguiente sentencia (Nota: el nombre de fichero ha de ser diferente al original, si queremos conservarlo):

```
180 BSAVE"BENDER_S.SC2",0,&H3FFF,S
```

Y para finalizar he colocado una línea que espera que pulsemos una tecla para salir:

```
190 IF INKEY$="" THEN 190
192 END
```

A continuación podéis ver todo el programa (también se encuentra en el DSK que se puede descargar al final del tutorial):

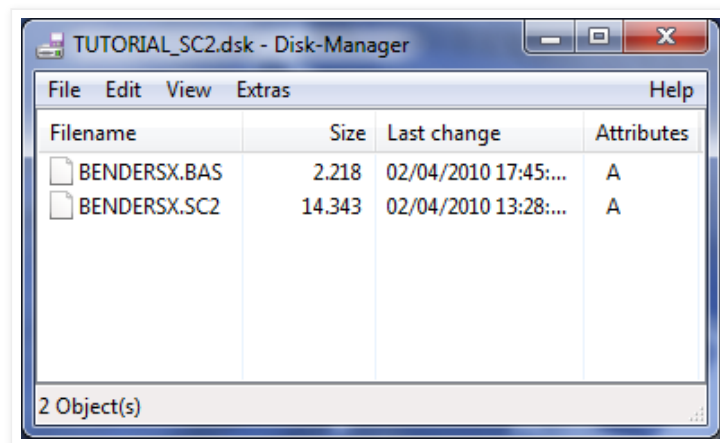
```
1 color ,,1
10 SCREEN 2,2
19 ' carga la imagen sin sprites
20 BLOAD"BENDERSX.SC2",S
30 GOSUB 10010
90 ' visualiza sprites
100 PUT SPRITE 0,(33,23),1,0
101 PUT SPRITE 1,(64,28),15,1
102 PUT SPRITE 2,(18,96),1,2
103 PUT SPRITE 3,(34,96),1,3
104 PUT SPRITE 4,(40,85),14,4
105 PUT SPRITE 5,(137,85),1,5
106 PUT SPRITE 6,(137,101),1,6
107 PUT SPRITE 7,(32,120),14,4
108 PUT SPRITE 8,(146,118),1,7
174 ' guarda la imagen con los sprites
180 BSAVE"BENDER_S.SC2",0,&H3FFF,S
184 ' espera una respuesta del teclado
190 IF INKEY$="" THEN 190
192 END
200 ' --- Slot 0
210 ' color 1
220 DATA 00,00,05,0A,15,2A,54,A8
230 DATA 00,00,00,00,00,00,00,00
240 DATA 15,A8,50,80,00,00,00,00
250 DATA 00,00,00,00,00,00,00,00
260 '
270 ' --- Slot 1
280 ' color 15
290 DATA C0,30,18,0C,0C,00,00,00
300 DATA 00,00,00,00,00,00,00,00
310 DATA 00,00,00,00,00,00,00,00
320 DATA 00,00,00,00,00,00,00,00
330 '
340 ' --- Slot 2
350 ' color 1
360 DATA 00,00,00,03,0D,3A,D5,00
370 DATA 00,00,00,00,00,00,00,00
380 DATA 0D,3A,D5,AA,55,AA,54,00
390 DATA 00,00,00,00,00,00,00,00
400 '
410 ' --- Slot 3
420 ' color 1
430 DATA 50,A0,40,80,00,00,00,00
440 DATA 00,00,00,00,00,00,00,00
450 DATA 00,00,00,00,00,00,00,00
460 DATA 00,00,00,00,00,00,00,00
470 '
480 ' --- Slot 4
```

```

490 ' color 14
500 DATA 80,00,00,00,00,00,00,00
510 DATA 00,00,00,00,00,00,00,00
520 DATA 00,00,00,00,00,00,00,00
530 DATA 00,00,00,00,00,00,00,00
540 '
550 ' --- Slot 5
560 ' color 1
570 DATA 01,00,01,00,01,00,01,00
580 DATA 01,00,01,00,01,00,01,02
590 DATA 00,00,00,00,00,00,00,00
600 DATA 00,00,00,00,00,00,00,00
610 '
620 ' --- Slot 6
630 ' color 1
640 DATA 00,04,28,50,A8,50,20,00
650 DATA 00,00,00,00,00,00,00,00
660 DATA 00,00,00,00,00,00,00,00
670 DATA 00,00,00,00,00,00,00,00
680 ' --- Slot 7
690 ' color 1
700 DATA 00,00,00,90,90,95,95,95
710 DATA 92,90,A0,C1,00,00,00,00
720 DATA 20,54,44,44,44,4C,54,64
730 DATA C4,44,54,48,80,00,00,00
740 DATA *
10000 REM -- LOAD SPRITES
10010 S=BASE(9)
10020 READ R$: IF R$="" THEN RETURN ELSE VPOKE S,VAL("&H"+R$):S=S+1:GOTO 10020
10030 RETURN

```

Ahora tendremos que copiar los ficheros SC2 y el BAS, en un disco de MSX o en un fichero de disco (.DSK). Para este último caso, podemos usar la utilidad [Disk-Manager](#) de Lex Lechz (Windows).



Y para finalizar lo ejecutamos y ya tenemos nuestra imagen!!!

Nota: Es posible que alguna posición no nos concuerde. En ese caso, tendremos que modificarlas hasta que consigamos que se muestre todo correctamente.

Para ver nuestra imagen final, podemos usar el siguiente listado:

```

10 color 15,1,1
20 SCREEN 2,2
30 BLOAD"BENDER_S.SC2",S
40 IF INKEY$="" THEN 40
50 END

```

NOTA: No sería muy complicado hacer un programa para crear un efecto de animación, cambiando el número de sprite de un plano dado. Solo es cuestión de animarse! ;-)

Más información:

- [Imagen final](#) del tutorial
- [Disco](#) con ejemplos
- [Screen Conversor](#) de Jannone
- [TinySprite](#) de Jannone
- [Disk-Manager](#) de Lex Lechz



Etiquetas: [MSX](#), [Pixel Art](#)

Enlaces a esta entrada

[Crear un enlace](#)

Entrada más reciente • • • • • • • • • • [Página principal](#) • • • • • • • • • • [Entrada antigua](#) |

