

mvac7 blog

08 mayo 2013

sprites del V9938 (2 de 2). OR de colores.

Como explicaba en la [primera parte](#), el V9938, permite dibujar un color por cada línea del sprite, pero esto no soluciona mucho el trabajo de los grafistas. Por suerte el chip guarda un As en la manga: la superposición de sprites.

Dentro del valor de cada color, disponemos del bit 6 que indica si esa línea la fusionaremos con el color del plano anterior. Esto nos ofrece 3 colores o más por línea, ya que permite superponer varios sprites como si fueran [bitplanos](#).

Ahora ya podemos representar nuestras figuras móviles con más detalle, pero el sistema tiene dos problemas:

1. Gastamos 2 o más sprites. Si utilizamos esta medida para todas nuestras figuras, perdemos la ventaja de poder visualizar 8 sprites por línea.
2. El calculo del color resultante se realiza mediante OR. Esto nos obliga a jugar con colores muy determinados, por lo que limita el uso de los colores y complica la tarea del grafista.

El primer problema no es algo excesivamente importante. Tendremos que diseñar nuestros juegos adaptados a esta limitación. Si por ejemplo, nuestro juego es un [shoot'em up](#), para los disparos podemos usar solo un sprite. La superposición es un sistema totalmente combinable que podemos utilizar cuando creamos necesario.

El segundo problema, se puede solucionar modificando la paleta con el orden que cuadre con los cálculos de OR que nos interese.

Recomendación: A la hora de diseñar una fase de un juego, es mejor empezar por definir los colores de los sprites y luego los que necesitemos para los gráficos, donde también podremos aprovechar los de los sprites.

A continuación muestro un ejemplo de una de las tablas más útiles, ya que seguramente uno de los colores que más usaremos será el negro (1) . Con esta, dispondremos de acceso a varias combinaciones de tonos de colores:

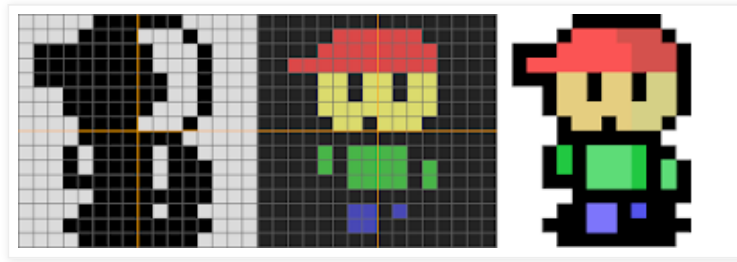
A	B	A or B
1 1	2 2	3 3
1 1	4 4	5 5
1 1	6 6	7 7
1 1	8 8	9 9
1 1	10 10	11 11
1 1	12 12	13 13
1 1	14 14	15 15

Nota: En la wiki de la MRC, hay un artículo que analiza muy bien el funcionamiento del color OR, donde se muestran varias tablas y casos de uso de juegos de Konami ([The OR Color](#)).

Como funciona?

Para el caso de 3 colores por línea, necesitaremos definir dos sprites que se situarán en planos consecutivos. En el sprite situado en el plano superior, en la información de los colores, el bit de OR, lo pondremos a 0 y en el de plano inferior lo pondremos a 1 (activo). A la hora de dibujar los sprites, los puntos que coincidan en los dos, se mostrarán

con el color resultante del OR de los dos colores asignados a la línea de cada sprite.



A la hora de visualizarlos, tendremos que procurar que siempre estén colocados en la misma posición, escribiendo las mismas coordenadas en los atributos de los dos sprites, excepto en Basic como ahora veremos.

Facilidades en Basic

La versión 2 del MSX Basic, proporciona algunas facilidades frente a otros lenguajes (C o ASM), gracias al comando **PUT SPRITE**, que nos simplificará algunas tareas con relación a la superposición de colores.

Lo primero que nos encontramos es su parámetro de color, que escribe el valor asignado a las 16 líneas de la tabla de colores de los sprites, situada en la VRAM (h1C00 en screen4). Es una ventaja si vamos a usar solo 3 colores para el diseño de nuestra figura, ya que nos ahorraremos de escribir esos datos para cada plano (16B por sprite).

```
60 PUT SPRITE 2, (100,130), &H1,2
70 PUT SPRITE 3, (100,130), &H42,3 -- Activamos el bit de OR (h40) y asignamos el color 2.
```

Obtendremos un sprite con los colores 1, 2 y como extra el 3 (donde se solapen los dos sprites).

La segunda facilidad, es que podemos controlar la posición de los 2 sprites únicamente accediendo al que se encuentra en el plano superior.

```
100 PUT SPRITE 2, (100,Y(i)+20), &H1,2 -- movemos los planos 2 y 3 (OR) solo cambiando la
posición del sprite 0.
```

Ejemplo (bajar):

```
10 DEFINT A-Z
12 DIM Y(100)
14 COLOR 15,14,14:SCREEN 4,2
16 RESTORE 10540:GOSUB 1030:rem palette
20 RESTORE 10020:HL=BASE(24):GOSUB 1010:rem patrones sprites
30 RESTORE 11010:FOR I=0 to 99:READ A:Y(I)=A:NEXT
40 PUT SPRITE 0, (180,100),0,0
50 PUT SPRITE 1, (180,100),0,1:rem los colores se asignan a la vram (linea 80)
60 PUT SPRITE 2, (100,130), &H1,2
70 PUT SPRITE 3, (100,130), &H42,3:rem los colores se asignan desde putsprite
80 RESTORE 10350:HL=&H1C00:GOSUB 1010:rem colores sprites
90 FOR I=0 to 99
100 PUT SPRITE 2, (100,Y(i)+20), &H1,2
110 IF INKEY$="" THEN NEXT ELSE END
120 GOTO 90
1000 REM -- LOAD DATA based on the function from Rafael Jannone
1010 READ R$: IF R$="" THEN RETURN ELSE VPOKE HL,VAL("&H"+R$):HL=HL+1:GOTO 1010
1020 rem Set Palette
1030 FOR C=1 TO 15:READ R,G,B:COLOR=(C,R,G,B):NEXT
1040 RETURN
10000 REM colorOR
10010 REM SPRITE DATA
10020 DATA 07,0F,7F,7F,7F,1F,1F,0F,0F,17,17,1F,07,0F,1F,0F
10030 DATA E0,10,88,88,C8,C8,88,10,E0,B0,B8,B8,F8,B0,F8,F0
10040 DATA 00,07,0F,3F,0D,0D,0F,06,00,0B,0B,03,00,03,03,00
```

```

10050 DATA 00,E0,F0,F0,B0,B0,F0,E0,00,C0,D0,D0,00,40,00,00
10060 DATA 07,1F,3F,7F,7F,FF,FF,FF,FF,FF,FF,7F,7F,3F,1F,07
10070 DATA E0,18,C4,E2,E2,F1,F1,F1,F1,F1,E2,E2,C4,18,E0
10080 DATA 00,07,1F,3F,3F,7F,7F,7F,7F,7F,3F,3F,1F,07,00
10090 DATA 00,E0,F8,FC,FC,FE,FE,FE,FE,FE,FE,FC,FC,F8,E0,00
10330 DATA *
10340 REM COLOR MODE2 DATA
10350 DATA 01,01,01,01,01,01,01,01,01,01,01,01,01,01,01
10360 DATA 48,48,48,48,4A,4A,4A,4A,4A,42,42,42,42,44,44,44
10500 DATA *
10510 REM MSX2_Palette
10520 REM Red, Green, Blue
10540 DATA 0,0,0
10550 DATA 1,5,1
10560 DATA 3,6,3
10570 DATA 2,2,5
10580 DATA 3,3,7
10590 DATA 5,2,1
10600 DATA 2,6,7
10610 DATA 6,2,2
10620 DATA 7,0,0
10630 DATA 6,6,3
10640 DATA 7,7,0
10650 DATA 1,4,1
10660 DATA 7,3,0
10670 DATA 6,6,6
10680 DATA 7,7,7
11000 REM sinus 0 -140 (100)
11010 DATA 70,65,61,56,52,48,43,39,35,32,28,25,21,18,15,12
11020 DATA 10,8,6,4,3,1,1,0,0,0,0,0,1,2,3,5
11030 DATA 7,9,11,14,17,20,23,26,30,34,37,41,46,50,54,58
11040 DATA 63,67,72,76,81,85,89,93,98,102,105,109,113,116,119,122
11050 DATA 125,128,130,132,134,136,137,138,139,139,139,139,138,138,136
11060 DATA 135,133,131,129,127,124,121,118,114,111,107,104,100,96,91,87
11070 DATA 83,78,74,70

```

Referencias:

- [Portar](#)
- [Technical Data Book: Programmer's Guide](#)
- [MRC Wiki: The OR Color](#)



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

Enlaces a esta entrada

[Crear un enlace](#)

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

