

Manual de Sprites Alive - Versión Basic  
Presentado por Tommy Pereira el 24 Septiembre 2003  
Escaneado y PDFado desinteresadamente por Steve R Sopp  
Traducido a castellano por MiguelSky  
"Gracias a Kevin Thacker por ponerlo online para todo  
el mundo, en el - Unofficial Amstrad WWW Resource"



**MENÚ PRINCIPAL**

**EL SUPERVISOR**

Comienzo del Supervisor .....	05
Supervisor Apéndice I Dimensiones de la Pantalla .....	45
Supervisor Apéndice II Mensajes de Error .....	46
Supervisor Apéndice III Notas para el Usuario Avanzado .....	48
Índice de Comandos del Supervisor .....	52

**EL DISEÑADOR**

Comienzo del Diseñador .....	55
Índice de Comandos del Diseñador .....	75

Quiero aprovechar estas líneas para dar gracias a Dawn, mi sufrida novia.  
Sin su paciencia este programa nunca se habría creado.



EL SUPERVISOR

GLEN COOK

**SUPERVISOR DE SPRITES**INTRODUCCION

La gama de ordenadores CPC de Amstrad está dotada de unas excelentes especificaciones de hardware. Sin embargo, el BASIC suministrado con el ordenador, a pesar de ser tan bueno, no muestra todo el potencial disponible en cuanto a gráficos y sonido.

Esta colección de programas solucionará esto ampliando el BASIC con una serie de comandos extra. Estos comandos extra permitirán al usuario manipular los gráficos más allá de lo que permite el BASIC suministrado. En el disco hay una serie de programas de demostración escritos completamente en BASIC para mostrarte el tipo de programas que podrás crear tu mismo. No necesitas tener ningún conocimiento de código máquina. Prueba la demostración por ti mismo.

RUN "DEMO1" y pulsa ENTER

Hay cinco programas de demostración en el disco. Se llaman DEMO1, DEMO2, DEMO3, DEMO4 y DEMO5. Después de leer este manual y usando los comandos extra serás capaz de escribir programas de una calidad similar a la de estos.

El programa tiene las siguientes características:

- 1) 64 Sprites definibles por el usuario
- 2) Secuencia de Animación
- 3) Movimiento fluido de píxeles
- 4) Detección de colisiones real
- 5) Utilización de Mode 0 o Mode 1
- 6) Control por Joystick o Teclado
- 7) Diseñador de Sprites comprensible
- 8) Comandos automatizados
- 9) Instrucciones de fácil entendimiento

El programa ha sido diseñado para eliminar grandes cantidades de BASIC, haciendo de este modo los programas más fáciles de escribir y entender. Esto se consigue haciendo que el Sprites tome la mayoría de las decisiones por ti. Es totalmente posible escribir un juego que tenga solo tres o cuatro líneas de BASIC para el bucle principal. El resto del programa podrían ser los sprites y los datos para la música.

Si le echas un vistazo a los listados de los programas de demostración podrás ver lo simples que son. Deberías ser capaz de escribir programas como esos en nada de tiempo.

**SUPERVISOR DE SPRITES**UNA INTRODUCCION A SPRITES Y GRAFICOS

Es muy importante que entiendas esta página antes de que empieces a utilizar el Sprites. Tómame tu tiempo para familiarizarte con los puntos expuestos en esta página. Si no comprendes el principio de funcionamiento de los sprites y el programa, encontrarás imposible utilizar el programa en toda su extensión.

Es importante familiarizarse con la diferencia entre gráfico y sprite. El gráfico es la forma real del carácter que será guardado en memoria. El sprite no es un gráfico real, pero sí puede asociarse a un gráfico. Esto puede parecer un poco confuso con un ejemplo explicaré qué quiero decir.

Por favor, ten siempre en mente que cada carácter individual en pantalla es un sprite independiente, sea una nave espacial, un hombre o un proyectil.

Tomemos como ejemplo el juego machaca-ladrillos, ese juego en el que una pelota va rebotando por la pantalla destruyendo los ladrillos de la parte superior. Tú controlas una raqueta en la parte inferior de la pantalla y tratas de mantener la pelota en juego golpeándola con tu raqueta. Para simplificar las cosas todos los ladrillos de la parte superior de la pantalla tendrán la misma forma, tamaño y color. Cada uno de los ladrillos de la parte superior de la pantalla es un sprite individual, la pelota es otro sprite y la raqueta es otro sprite.

Podríamos tener 40 ladrillos en la parte superior de la pantalla, digamos cuatro filas de diez. Cada uno de esos 40 ladrillos es un sprite. En total tenemos 42 sprites en la pantalla al principio del juego (ladrillos, raqueta y pelota). Sin embargo solo tendríamos que definir 3 gráficos, esto es el ladrillo, la raqueta y la pelota. Aunque tenemos cuarenta ladrillos, todos ellos son idénticos así que podemos asociar los 40 sprites al mismo gráfico.

Espero que esto empiece a tomar forma. No tiene sentido almacenar los 40 ladrillos en memoria porque es sólo desperdiciar memoria.

Llevemos la idea un poco más allá. Tenemos un hombre caminando por la pantalla en una secuencia animada. Ese hombre es un solo sprite. Sin embargo, para obtener el efecto de la animación tenemos que ir cambiando el gráfico del hombre cada vez que se mueve. En este caso es 1 sprite que tiene asociados una serie de gráficos.

Como otro ejemplo, considéralo como una imprenta. La imprenta está preparada para imprimir el dibujo de una chica. Desde ese gráfico obtendremos una serie de copias en papel. Tu gráfico almacenado en memoria es la imprenta y las copias en papel son tus sprites. Tan sólo necesitamos un gráfico para obtener cualquier número de sprites desde él.

Por favor recuerda la diferencia entre tus gráficos y tus auténticos sprites. Cualquier duda que te quede se aclarará cuando utilices el programa unas cuantas veces. ;¡La práctica te perfeccionará!!

**SUPERVISOR DE SPRITES**NOTAS IMPORTANTES SOBRE EL USO DE ESTE PROGRAMA

Puedes usar el Sprites en Modo 0 o Modo 1.

El PAPER de la pantalla debe establecerse siempre como cero. Puedes elegir el color que quieras mediante el comando INK 0,n. Si usas el comando PAPER con una variable distinta de cero los sprites no funcionarán correctamente.

Cuando utilices el Sprites NO DEBES hacer scroll con la pantalla. Si quieres escribir algo en la parte inferior de la pantalla debes usar punto y coma tras la información que quieras escribir. Si la pantalla hace scroll el sistema de coordenadas del Sprites no funcionará correctamente.

Cuando estés asignando los colores a las tintas con el INK i,c no puedes utilizar colores alternantes. Si intentas usar parpadeo se mostrará un mensaje de error.

El sistema de coordenadas utilizado en el Supervisor es muy similar al sistema de coordenadas de gráficos que utiliza el Amstrad; 0,0 es la esquina inferior izquierda de la pantalla. Las diferencias entre los dos sistemas son las siguientes.

El sistema del Amstrad tiene una serie de coordenadas en pantalla apuntando al mismo píxel.

p.e. Mode 0: 0,0 : 1,0 : 2,0 y 3,0 son los mismos puntos en la pantalla.

El sistema del Supervisor tiene un punto distinto en la pantalla para cada coordenada individual.

p.e. Mode 0: 1,0 es un punto distinto a 0,0.

Esto se aplica tanto a las coordenadas X como Y. Las coordenadas de pantalla máximas para cada modo se encuentran en el Apéndice 1.

Algunos de los comandos descritos en el manual devolverán valores al BASIC. Estos comandos tendrán el carácter '@' delante del nombre de la variable. Es MUY importante que incluyas este carácter cuando se muestre en un comando. Si no, se corromperá el programa y colgará el ordenador.

Es importante que grabes tu programa antes de ejecutarlo. Si introdujiste un error y pierdes tu programa es muy frustrante volver a teclearlo.

DEBES utilizar los comandos |ERASE o |RESET como primera instrucción de tu programa.

Si vas a utilizar un tamaño de pantalla distinto al estándar del Amstrad, el comando LOCATE no funcionará correctamente. Deberás usar el método de ensayo/error para encontrar el valor correcto para el comando LOCATE.

**SUPERVISOR DE SPRITES****PANTALLA Y COMANDOS GRAFICOS****|SCREEN,n**

Este comando permite al usuario cambiar la forma de la pantalla. Hay nueve tamaños distintos de pantalla para elegir, desde |SCREEN,0 a |SCREEN,7. Para volver a la pantalla estándar simplemente utiliza el comando |SCREEN sin ningún número tras él. Deberías experimentar con este comando con el BORDER de distinto color que el PAPER para que puedas ver la forma exacta que tomará la pantalla. Los tamaños de las diferentes pantallas se encuentran en el Apéndice 1.

NOTA: Por defecto se establece la pantalla estándar del Amstrad.

**|DGET,d,x,y**

Este comando coloca un gráfico en la memoria del ordenador. d es el número de gráfico, x e y son las dimensiones del gráfico. Para poner un gráfico en memoria, el gráfico debe estar en la esquina superior izquierda de la pantalla. Puedes hacer esto bien definiendo caracteres e imprimiéndolos en la esquina superior izquierda, o bien utilizando el comando PLOT para dibujar el carácter en la parte superior izquierda.

Debes entonces utilizar el comando |DGET para almacenar el gráfico. Los valores x e y serán el ancho y la altura del gráfico en píxeles. Para ahorrar el máximo de memoria posible, la parte superior del gráfico debe tocar la parte superior de la pantalla y el borde izquierdo del gráfico debe tocar el margen izquierdo de la pantalla. El resto de la pantalla debe estar completamente libre de cualquier gráfico. Es muy importante que no haya gráficos a la derecha del gráfico que quieres almacenar.

Si estas utilizando el programa Diseñador de Sprites, el cual recomendamos encarecidamente, no necesitarás usar este comando.

NOTA: Debe establecerse el modo correcto, p.e. MODE 0 o MODE 1. También la pantalla debe tener su tamaño por defecto, p.e. |SCREEN.

Puedes definir desde el gráfico 0 hasta el gráfico 63. El tamaño máximo de un gráfico es 32 x 32 píxeles.

Por favor recuerda que las dimensiones x e y deben ser especificadas en píxeles. NO utilices el sistema de coordenadas del Amstrad.

Antes de que introduzcas algún gráfico en memoria, debes asegurarte que has reservado suficiente memoria con el comando MEMORY. Si no reservas memoria suficiente puede corromperse tu programa.

Para calcular la cantidad de memoria que necesitas consulta la fórmula de la página siguiente.

**SUPERVISOR SPRITES**CALCULO DE TAMAÑO DE MEMORIA DE GRÁFICOS

Si vas a ubicar los gráficos en memoria mediante el comando |DGET necesitas saber cuanta memoria de gráficos vas a ocupar. Necesitas saberlo para calcular el valor correcto para el comando MEMORY.

Hay dos métodos del cálculo distintos, dependiendo del modo de pantalla que vayas a utilizar.

CALCULO PARA MODE 0

- 1) DIVIDIR LA DIMENSION X ENTRE 2 DESCARTANDO EL RESTO (INT)
- 2) SUMAR 1 AL RESULTADO
- 3) MULTIPLICAR EL RESULTADO POR LA DIMENSION Y
- 4) MULTIPLICAR EL RESULTADO DE 3) POR 2

$$\text{p.e. MEM} = [ \text{INT} ( \text{DIMENSION X} / 2 ) + 1 ] * Y * 2$$

Debes hacer esto para todos los gráficos a los que vayas a |DGET. Cuando tengas el total para todos los gráficos resta el resultado de 23389 para obtener el valor que debes utilizar con el comando MEMORY.

CALCULO PARA MODE 1

- 1) DIVIDIR LA DIMENSION X ENTRE 4 DESCARTANDO EL RESTO (INT)
- 2) SUMAR 1 AL RESULTADO
- 3) MULTIPLICAR EL RESULTADO POR LA DIMENSION Y
- 4) MULTIPLICAR EL RESULTADO DE 3) POR 4

$$\text{p.e MEM} = [ \text{INT} ( \text{DIMENSION X} / 4 ) + 1 ] * Y * 4$$

Debes hacer esto para todos los gráficos a los que vayas a |DGET.

Cuando tengas el total para todos los gráficos resta el resultado de 23389 para obtener el valor que debes utilizar con el comando MEMORY.

Puedes probar tus cálculos una vez que tengas DGETados tus gráficos en memoria. Una vez que todos los gráficos estén almacenados teclea la línea siguiente

```
A%=0:|SMEM,@A%:PRINT A%-1
```

El valor que se muestra en pantalla debe ser el mismo utilizado para el comando MEMORY.

**SUPERVISOR DE SPRITES****CARGANDO GRÁFICOS DESDE EL DISEÑADOR DE SPRITES**

Recomendamos encarecidamente al usuario que diseñe sus gráficos con el Diseñador de Sprites. Tiene una serie de ventajas sobre la captura manual de los datos desde la parte superior de la pantalla. Sin embargo si el usuario quiere usar Sprites en modo 1 tendrá que usar el método manual. Este método se describe en detalle en la página 9 bajo el encabezamiento |DGET.

Los usuarios del Diseñador de Sprites deben seguir detalladamente las instrucciones de debajo para evitar cualquier problema que pueda ocurrir.

1) Diseña totalmente tus gráficos y asegúrate que están listos para usarse con tu programa principal.

2) Antes de que selecciones la opción (8) CREATE SPRITE DATA en el Diseñador de Sprites, averigua cuánta memoria de gráficos necesitarás. Para hacer esto pulsa la opción (6) SPRITE DATA y pulsa Y. Al final de la lista de gráficos se mostrará la memoria total utilizada. Recuerda este número.

3) Ahora selecciona la opción (8) CREATE SPRITE DATA. Introduce el disco en el que quieres guardar los gráficos e introduce un nombre de fichero.

4) Ahora estás listo para cargar los gráficos en tu programa. Antes de que hagas esto debes insertar un comando MEMORY. Para calcular el comando MEMORY resta el total de memoria utilizada de 23389. El resultado de la operación es el que debe usarse con el comando MEMORY.

5) Deberías ahora utilizar el comando |DRAW. Esto cargará todos los gráficos en memoria. Si quieres cambiar los colores utilizados dentro del DISEÑADOR DE SPRITES utiliza el comando |COLOUR.

Ahora vamos a describir los comandos |DRAW y |COLOUR más detalladamente.

**|DRAW, "nombredefichero"**

Este comando puede utilizarse en modo directo o incluido dentro de tu programa. Este comando cargará la información del gráfico para el nombre de fichero que le indiques. Los datos del gráfico almacenados en el disco tiene el nombre xxxxxxxx.DRW. No incluyas el DRW cuando especifiques el nombre.

No debes utilizar este comando hasta que hayas utilizado comando MEMORY como se ha descrito más arriba.

Puedes repetir la prueba mostrada en la página anterior para asegurarte que has utilizado el valor correcto para el comando MEMORY.

**SUPERVISOR DE SPRITES****COMANDOS GENERALES DE PANTALLA**

Los colores que utilizas en tu programa están almacenados en memoria. Haciendo esto puedes conseguir resultados profesionales cuando muestres los sprites en la pantalla. Si pones todas las tintas a negro, colocas los sprites en la pantalla y pones entonces las tintas a su valor correcto parecerá que los sprites hayan sido colocados en pantalla al mismo tiempo.

El bloque de memoria utilizado para almacenar colores tiene 16 bytes de longitud.

El bloque de memoria es actualizado cuando cargas gráficos en el ordenador mediante el comando |DRAW. Si cargas cualquier gráfico en el ordenador, los colores que elegiste al usar el Diseñador se transferirán en este bloque de memoria. Puedes acceder a los colores utilizando el comando |COLOUR como se explica más abajo.

La única limitación usando este método es que no puedes utilizar parpadeo. Si intentas utilizar el comando Amstrad INK con más de dos variables (tinta y color) se mostrará un mensaje de error.

**| COLOUR**

Este comando pone todas las tintas de la pantalla en sus colores por defecto (los que se hayan definido con el comando Amstrad INK i,c). Este comando puede usarse en combinación con el comando |INKBLACK o el comando |DRAW.

Cuando se utiliza este comando en combinación con el comando |DRAW cambiará las tintas a los colores seleccionados por el Diseñador de Sprites.

**| INKBLACK**

Este comando cambia todos los colores de la pantalla, incluyendo el BORDER y PAPER, a negro. Los colores utilizados se recordarán, no necesitarás definir los colores otra vez, puedes simplemente utilizar el comando |COLOUR como se describe más arriba.

**| WP**

Este comando limpia la pantalla. La pantalla cambia a Mode 2, vuelve a su tamaño normal y los colores de PEN y PAPER cambian para facilitar la lectura.

**| CLS**

Si vas usar un tamaño de pantalla distinto al tamaño normal de la pantalla del Amstrad, cuando trates de usar el comando CLS normal una pequeña porción de la pantalla no se limpiará correctamente. Para limpiar la pantalla completamente debes utilizar este comando.

**SUPERVISOR DE SPRITES**SPRITES ESPECIALES Y COMO UTILIZARLOS

Me ha llevado algún tiempo decidir en qué parte del manual aparecería esta página. En una primera lectura puede parecer realmente confuso, sin embargo, si no la incluyo cerca del comienzo del manual podrías no entender por qué algunos sprites se comportan de manera distinta que otros. Te sugiero que leas esta página ahora y vuelvas a ella cuando entiendas el programa un poco más.

Los sprite 0 y sprite 1 se controlan con el joystick y teclado, el resto de los sprites (de 2 a 63) son controlados por tu programa.

Si no vas a utilizar un sprite controlado por joystick o teclado no debes usar los sprite 0 o sprite 1.

Las diferencias entre estos sprites especiales y los sprites de propósito general son las siguientes:

- . Sprite 0 y sprite 1 no rebotan en el borde de la pantalla.
- . Sprite 0 y sprite 1 no rebotan cuando colisionan con otro sprite.
- . Sprite 0 y sprite 1 se comportan de forma distinta cuando colisionan con otros sprites.
- . Sprite 0 y sprite 1 tienen comandos de proyectil distintos a los otros sprites.

Es muy fácil olvidar cuando estás escribiendo tus programas estas diferencias (yo lo he hecho varias veces). Si encuentras que uno de tus sprites se comporta distinto, comprueba tu programa para asegurarte de que no estás utilizando sprite 0 o sprite 1 para otro propósito que no sea el control por joystick/teclado.

El sprite 0 es el sprite del carácter principal. Se maneja controlado por el joystick.

El sprite 1 debería utilizarse solo en juegos de dos jugadores o en juegos que utilicen el teclado en lugar del joystick.

**SUPERVISOR DE SPRITES****SPRITES, GRÁFICOS Y COLOCARLOS EN PANTALLA****|SGET,s,d**

Este comando te permite asociar un sprite a un gráfico previamente definido. s es el número de sprite y d es el número del gráfico. De momento debes tener tus gráficos almacenados en memoria. Para utilizar tus gráficos debes asociarlos a un sprite.

s puede ser cualquier sprite no definido en el rango 0 a 63. d debe ser un gráfico definido en el rango 0 a 63.

Usemos el juego del machaca-ladrillos como ejemplo. Has definido 3 gráficos. El gráfico 0 es la raqueta, el gráfico 1 es la pelota y el gráfico 2 es el ladrillo. Haremos que el sprite 0 sea la raqueta, el sprite 2 la pelota y los sprites 3 a 42 los ladrillos. Para hacer esto podríamos usar los siguientes comandos.

```
|SGET,0,0:|SGET,2,1:FOR I%=3 TO 42:|SGET,I%,2:NEXT I%
```

Utilizamos el símbolo % porque estamos utilizando variables enteras, esto será explicado en detalle más adelante.

Utilizamos el sprite 0 para la raqueta ya que queremos controlarla con el joystick. Sin embargo, si queremos que la raqueta se controle con el teclado necesitamos utilizar el sprite 1.

La línea superior asocia los sprites a su gráfico correspondiente. Si queremos que la raqueta sea el sprite 1 y la pelota el sprite 2, tendremos que usar

```
|SGET,1,0:|SGET,2,1:FOR . . .
```

**IMPORTANTE:** Debes utilizar el |SCREEN pertinente antes de utilizar el comando |SGET. Si deseas utilizar un tamaño de pantalla distinto a la pantalla estándar del Amstrad debes cambiar el tamaño de la pantalla con el comando |SCREEN antes de introducir un comando |SGET, equivocarse en esto hará que los sprites se comporten incorrectamente.

**|SPUT,s,x,y**

Este comando se utiliza para colocar el sprite en pantalla. s es el número de número de sprite, x e y son las coordenadas para la posición del sprite que irá a la pantalla. s debe ser un sprite que hayas definido, x e y deben estar dentro de las coordenadas permitidas para esa pantalla. Las coordenadas se refieren a dónde irá la esquina superior izquierda del sprite.

Recuerda que si colocas un sprite en pantalla cerca de la parte inferior o del margen derecho debes tener en cuenta la altura y el ancho del sprite. p.e. si un sprite tiene una altura de diez píxeles entonces no sería buena idea colocarle en las coordenadas 100,8 simplemente porque el sprite se iría por la parte inferior de la pantalla.

**SUPERVISOR DE SPRITES****PONIENDO Y QUITANDO SPRITES DE LA PANTALLA****|SPUT,s**

Este comando se utiliza para quitar un sprite de la pantalla. s es el número de sprite. Si has quitado un sprite de la pantalla entonces utilizando otro comando |SPUT,s el sprite volverá a la pantalla en mismo sitio del que se quitó. En otras palabras el comando |SPUT,s encenderá y apagará el sprite en la pantalla.

Ejemplo.

```
|SPUT,0,100,100 - Coloca el sprite 0 en la posición 100,100
|SPUT,0         - Quita el sprite 0 de la pantalla
|SPUT,0         - Coloca el sprite 0 de nuevo en la pantalla
```

NOTA: Para poder utilizar el comando |SPUT,s el sprite debe haber sido puesto en pantalla inicialmente con un |SPUT,s,x,y

**|SPUT,s1,s2**

Este comando es idéntico al comando |SPUT,s excepto en que quita o pone varios sprites en pantalla. s1 es el primer sprite y s2 es el último sprite. No hay problema si no has definido todos los sprites entre s1 y s2 porque el programa ignorará aquellos sprites que no hayan sido definidos.

**|SPUTALL**

Este comando quita todos los sprites de pantalla. Esto es útil al final del juego cuando necesitas la pantalla limpia.

IMPORTANTE: Si has colocado un sprite en pantalla no debes intentar colocar el mismo sprite en una parte distinta de la pantalla sin primero quitar el sprite original.

```
p.e. |SPUT,0,100,100:|SPUT,0,120,30 - Esto es incorrecto
      |SPUT,0,100,100:|SPUT,0:|SPUT,0,120,30 - Esto es correcto
```

**|RESET**

Este comando reseteará todas las opciones del programa a sus valores por defecto. Los datos que asocian los sprites a los gráficos se borrarán. Los datos de los gráficos permanecerán intactos.

**|ERASE**

Este comando hace lo mismo que el comando |RESET, pero borrando los datos de los gráficos también. ESTE COMANDO DEBE SER EL PRIMER COMANDO UTILIZADO AL PRINCIPIO DE CUALQUIER PROGRAMA.

**SUPERVISOR DE SPRITES****ESTABLECIENDO LOS ATRIBUTOS DE LOS SPRITES****|XEDGE, s, n**

Este comando determina las propiedades del sprite cuando colisiona con los márgenes izquierdo y derecho de la pantalla. s es el número de sprite, n es el número que determina que hará el sprite cuando toque el margen de la pantalla. La tabla de debajo muestra como elegir el atributo correcto.

NUMERO	ATRIBUTO
1	Desaparecer
2	Parar
3	Rebotar
4	Trasladar

Desaparecer : El sprite se quitará de la pantalla cuando toque el borde de la pantalla.

Parar : El sprite se detendrá pero permanecerá en la pantalla.

Rebotar : El sprite rebotará en el borde de la pantalla como si hubiera un muro.

Trasladar : El sprite se quitará de la pantalla y será colocado de nuevo en el margen opuesto de la pantalla.

**|YEDGE, s, n**

Este comando es idéntico al comando |XEDGE excepto en que este determina lo que hará el sprite cuando toque los límites inferior o superior de la pantalla. Utiliza el mismo número de atributo que el comando |XEDGE.

Los comandos |XEDGE y |YEDGE pueden tener valores distintos. P.e. El |YEDGE se pone a rebotar mientras que el |XEDGE se ponga a trasladar.

NOTA: Por defecto está establecido que los sprites generales REBOTEN en todos los márgenes.

Las opciones por defecto para los sprite 0 y sprite 1 son que se PAREN en los márgenes.

No puedes alterar los atributos de margen para los sprites que definas como proyectiles. Los sprites proyectiles se describirán en detalle más adelante.

Los Sprite 0 y sprite 1 sólo pueden establecerse para que PAREN o se TRASLADEN. Si intentas ponerlos a rebotar o desaparecer se mostrará un mensaje de error.

**SUPERVISOR DE SPRITES****MOVIENDO LOS SPRITES****|SDIR,s,xs,ys**

Este comando determina la velocidad a la cual se moverá el sprite. s es el número de sprite, xs e ys son las velocidades del sprite. La variable velocidad determina cuantos píxeles a la vez se moverá el sprite. No recomendamos que este valor exceda de 8. Esto debería ser lo suficientemente rápido para la mayoría de las aplicaciones. Para conseguir que el sprite se mueva hacia la izquierda debes utilizar una velocidad negativa para xs. Para conseguir que el sprite se mueva hacia abajo en la pantalla tienes que utilizar una velocidad negativa para ys.

NOTA: En circunstancias normales la velocidad de los sprite 0 y sprite 1 se establecerá con los comandos |STIXSPEED y |KEBSPEED.

No puedes establecer la velocidad de los proyectiles con este comando. La velocidad de los proyectiles se establece utilizando el comando |MISSILE.

Tienes una serie de comandos que te permitirá controlar los sprites mediante el joystick, teclado o ambos. Estos van desde simplemente obtener la posición del joystick a mover realmente el sprite mediante el joystick o el teclado.

Los comandos de esta sección tratan de movimiento de los sprites en la dirección del joystick o el teclado.

Como ya sabes, el sprite 0 y el sprite 1 se controlan por medio de joystick y teclado. Los comandos de más abajo y de la página siguiente determinarán como se controlan los sprite 0 y sprite 1.

**|STIX,n**

El comando |STIX permite al sprite 0 moverse bajo el control del Joystick. El número tras el comando determina en qué dirección se permite moverse al sprite. Esto nos permite que el sprite se mueva solo izquierda-derecha; o izquierda, derecha, arriba, abajo pero no en diagonal; o izquierda, derecha, arriba, abajo y diagonal. Este número también determina si se activará o no el botón de fuego. El sprite se moverá bajo el control del joystick cada vez que el sprite 0 intente moverse. p.e. |MOVEALL o |MOVE,0,20 etc.

**|KEB,n**

Este comando es idéntico al comando |STIX excepto en que éste controla al sprite 1 y determina de qué manera se moverá el sprite bajo el control del teclado. El sprite se moverá bajo el control del teclado cada vez que el sprite 1 intente moverse. p.e |MOVEALL o |MOVE,0,20 etc.

**SUPERVISOR DE SPRITES**UTILIZANDO EL JOYSTICK Y TECLADO

Para poder informar al programa de las direcciones en que pueden moverse los sprite 0 y 1 debes usar la tabla siguiente.

NUMERO	DIRECCION	COMENTARIO
0	NINGUNA	JOYSTICK/TECLADO DESACTIVADO
1	ARRIBA	PERMITE AL SPRITE IR HACIA ARRIBA
2	ABAJO	PERMITE AL SPRITE IR HACIA ABAJO
4	IZQUIERDA	PERMITE AL SPRITE IR HACIA LA IZQUIERDA
8	DERECHA	PERMITE AL SPRITE IR HACIA LA DERECHA
16	DIAGONAL	PERMITE AL SPRITE MOVERSE EN DIAGONAL
32	FIRE	BOTON DE FUEGO ACTIVADO
64	CONTINUO	EL SPRITE NUNCA PARARA

Para utilizar esta tabla simplemente decide que características quieres activar y suma sus números para obtener un valor. Este es el número que tienes que poner tras los comandos |STIX o |KEB.

Ejemplo.

El Sprite 0 se moverá Izquierda-Derecha con el botón de fuego activado.

$4 + 8 + 32 = 44$  |STIX,44

El Sprite 1 se moverá Arriba, Abajo, Izquierda y Derecha sin movimientos diagonales ni botón de fuego.

$1 + 2 + 4 + 8 = 15$  |KEB,15

Si quieres el botón de fuego activado así como las direcciones sería

$1 + 2 + 4 + 8 + 32 = 47$  |KEB,47

Si quieres movimiento diagonal también

$1 + 2 + 4 + 8 + 16 + 32 = 63$  |KEB,63

El movimiento Continuo significa que el sprite no se detendrá, incluso si sueltas el joystick. El sprite solo se parará cuando toque un margen de pantalla.

Si no vas a utilizar el joystick no debes poner el sprite 0 en pantalla y no debes usar el comando |STIX.

Si no vas a utilizar el teclado no debes poner el sprite 1 en pantalla y no debes usar el comando |KEB.

NOTA: Los parámetros por defecto tanto para el joystick como para el teclado son ambos desactivados.

**SUPERVISOR DE SPRITES****UTILIZANDO EL JOYSTICK Y TECLADO****|STIXSPEED,u,d,l,r**

El comando |STIXSPEED informa al programa de la velocidad a la que el sprite 0 se moverá mientras esté bajo control del joystick. Todas las variables deben usarse, p.e. usa cuatro valores. Este es el caso aún incluso el joystick pueda solo ser programado para moverse en dos direcciones. Los valores deben ser todos números positivos.

**|KEBSPEED,u,d,l,r**

Este es idéntico al comando |STIXSPEED excepto en que controla la velocidad para el sprite 1.

Ejemplo.

|STIXSPEED,1,1,2,2      Esto moverá el sprite 0 a la velocidad de 1 píxel por vez cuando se mueva arriba y abajo y 2 píxeles cuando se mueva izquierda o derecha.

|KEBSPEED,4,4,2,2      Esto permite al sprite 1 moverse a la velocidad de 2 píxeles por vez cuando se mueva izquierda o derecha, y 4 píxeles por vez cuando se mueva arriba o abajo.

NOTA:      Los parámetros por defecto para los comandos |STIXSPEED y |KEBSPEED son todas las direcciones a velocidad 1.

**|KEBDEF,u,d,l,r,f**

Este comando permite al usuario definir las teclas que moverán el sprite 1. Este comando acepta números de tecla solamente. Esta información puede encontrarse en el Capítulo 7 página 23 del Manual de Usuario del Amstrad 6128. La información se encuentra también en la carcasa de la unidad de disco del 6128. Debes definir todos los números de tecla incluso si no usas todos ellos. P.e. Podrías no querer definir el botón de fuego, sin embargo has de usar un número de tecla para él.

Ejemplo.      |KEBDEF,58,63,60,61,18

El comando de la página anterior define el teclado de la siguiente manera

Arriba - E, Abajo - X Izquierda - S, Derecha - D, Fuego - ENTER

Por defecto están: teclas de flechas para dirección, espacio como fuego.

**SUPERVISOR DE SPRITES****UTILIZANDO EL JOYSTICK Y EL TECLADO**

Debido a la forma en que el ordenador comprueba ciertas teclas a veces se retornará un número de tecla erróneo. Esto puede afectar a la manera de que se mueven los sprites si estás utilizando ambos sprite 0 y sprite 1. Si encuentras que los sprites no se mueven en la dirección que has seleccionado o cambian de dirección por su cuenta, prueba a cambiar los números que elegiste con el comando |KEBDEF. Este problema también sucede con los comandos |READKEB y |READJOY.

Todos los comandos de arriba tratan de mover el sprite mediante el joystick o el teclado. Este tipo de control es bueno si quieres que el sprite se mueva en la dirección en la que mueves el joystick (space invaders, pacman). Sin embargo, hay juegos y situaciones en que necesitarás saber qué dirección de joystick/teclado se está pulsando sin mover realmente el sprite en esa dirección. Tomemos como ejemplo el juego Asteroids. Si mueves el joystick hacia la izquierda la nave rotará en el sentido contrario a las agujas del reloj, en vez de moverse hacia la izquierda.

El modo de abordar este problema es establecer la posición del joystick/teclado y entonces actuar sobre esta información. Los comandos de debajo comprobarán si cierta dirección está siendo pulsada.

**|READSTIX,k,@v%**

Este comando comprobará el joystick para ver si se está moviendo en una dirección en concreto. k debe contener un número entre 0 y 9. Este número indica la dirección a comprobar.

1			Si deseas comprobar si el joystick está siendo pulsado hacia delante el valor de k debería ser 1.
2	5		Para comprobar las diagonales izquierda y abajo k debería tener el valor 4, etc.
3	9	6	Número 9 comprueba el botón de fuego.
4	7		Número 0 comprueba NINGUNA dirección. Si se devuelve un valor de 0 en tu variable significa que el joystick está en la posición de centrado, si se retorna un valor de 1 significa que el joystick está siendo pulsado pero se desconoce la dirección.
8			

El comando devolverá un valor en v%. Esto puede ser 0 (sin pulsar) o 1 (pulsada).

**|READKEB,k,@v%**

Este comando actúa exactamente de la misma forma que el comando |READJOY excepto en que comprueba la dirección pulsada en el teclado. Las teclas de dirección están definidas por el comando |KEBDEF.

**SUPERVISOR DE SPRITES****MOVIENDO EL SPRITE****|MOVE, s**

Este comando mueve el sprite s. El sprite debe estar en la pantalla y controlado por joystick/teclado o habiéndole dado una dirección en que moverse con el comando |SDIR. Si el sprite no está en pantalla el comando será ignorado.

**|MOVE, s1, s2**

Este comando moverá todos los sprites que estén en pantalla que estén dentro de los límites s1 - s2. s1 es el sprite más bajo, s2 es el sprite más alto.

**|MOVEALL**

Este comando mueve todos los sprites que estén en la pantalla.

Ejemplo.

```

10 MODE 0 ; ESTABLECE MODO 0
20 ?"0":|DGET,0,7,7:CLS ; COLOCA GRÁFICO 0
30 |SGET,2,0:|SGET,3,0 ; LO ASOCIA A LOS SPRITES 2 Y 3
40 |SDIR,2,-1,-1:|SDIR,3,1,-1 ; ESTABLECE LAS DIRECCIONES
50 |SPUT,2,40,30:|SPUT,3,60,70 ; COLOCA LOS SPRITES EN PANTALLA
60 |MOVEALL:GOTO 60 ; MUEVE LOS SPRITES Y ENTRA EN BUCLE

```

Este programa no establece los comandos |XEDGE y |YEDGE ya que usaremos los valores por defecto de rebote en el margen de la pantalla. Experimenta con este programa alterando el modo de pantalla o velocidad de los sprites. Podrías probar incluso a alterar los atributos del margen poniendo una línea 45

p.e. 45 |XEDGE,2,4:|YEDGE,2,4

Mira si puedes hacer que el sprite 2 actúe controlado por el joystick en las cuatro direcciones. Como pista, necesitarás cambiar el número de sprite a 0.

Puedes notar que cuando los sprites colisionan salen rebotados. Esto se verá en la próxima página.

**|SWINDOW, s, x1, xh, y1, yh**

Este comando define una ventana en la que el sprite puede moverse. Cada sprite puede tener una ventana independiente. Cuando el comando |SGET se utiliza la ventana automáticamente toma todo el tamaño de la pantalla. Puedes alterar el tamaño de la ventana usando este comando. x1 es el margen izquierdo, xh es el margen derecho, y1 es el margen inferior e yh es el margen superior. Este comando debe usarse tras el comando |SGET.

Los cuatro valores deben expresarse en coordenadas del sprite.

**SUPERVISOR DE SPRITES****DETECCION DE COLISIONES****|COLLIDE, s, n**

Este comando determina qué hará el sprite si toca a cualquier cosa que no sea el margen de la pantalla. s es el número de sprite, n es el valor que determina lo que hará el sprite cuando colisione.

La tabla de debajo muestra como asignar el atributo correcto para cuando colisione el sprite s.

NÚMERO	ATRIBUTO
1	Desaparecer
2	Parar
3	Rebotar

Desaparecer : El sprite se quitará de la pantalla cuando toque un objeto distinto del margen de la pantalla.

Parar : El sprite dejará de moverse, pero permanecerá en pantalla.

Rebotar : El sprite rebotará con el objeto que toque.

NOTA : La opción por defecto para los sprites de propósito general es rebotar cuando colisione.

La opción por defecto para el sprite 0 y el sprite 1 es PARAR cuando colisione.

No se puede alterar el atributo de colisión de los sprites definidos como proyectiles. Los sprites proyectil se describirán en detalle más adelante.

El sprite 0 y el sprite 1 pueden solamente PARAR cuando colisionan. Tú no puedes alterar el atributo de colisión en estos dos sprites. Si intentas alterar el atributo se mostrará un mensaje de error.

**|REPON, s**

Para que puedas escribir tu programa necesitarás ser capaz de decir si ciertos sprites están colisionando. Hay una serie de comandos para ayudarte a conseguir esto. No necesitas detalles de todas colisiones, y es por esto por lo que hemos incluido los comandos |REPON y |REPOFF.

El comando |REPON activa el informe de colisión para el sprite s. Este informe de colisión no dice que harán los sprites cuando colisionen. Si el informe está activado en un sprite en concreto y tiene sus atributos puestos a rebotar y el sprite colisiona con algo, el sprite rebotará y se guardará su número de sprite.

**SUPERVISOR DE SPRITES****DETECCION DE COLISIONES**

Si el mismo sprite tiene dicho informe desactivado y colisiona, sí rebotará pero el número de sprite no será guardado.

Es mejor si mantienes el número de sprites con el informe activado a un mínimo. Algunas de los comandos de detección de colisión son relativamente lentos, por tanto, cuantos menos sprites tenga que comprobar más rápido será tu programa.

NOTA : El estado por defecto para el informe de colisión para todos los sprites es DESACTIVADO. Para poder utilizar los comandos de detección de colisión el informe de colisión debe estar activado.

No puedes activar el informe de detección de colisión para sprites proyectil. Estos sprites tienen sus propias rutinas de detección. Trataremos con los comandos de proyectil más adelante en el manual.

**|REPOFF,s**

Este comando es el opuesto al comando |REPON. Este comando desactiva la detección de colisiones para el sprite s.

Los siguientes comandos comprueban el estado del informe de colisión. Si su estado es desactivado y un sprite está colisionando el número de sprite no será guardado.

**|COLLTEST,s,@v%**

Este comando dice al usuario si un sprite en concreto está colisionando. s es el número de sprite, v% es la variable en la que quieres que se guarde la información. Por favor recuerda que debes usar la @ delante de la variable. Esto es porque el programa Sprites está pasándole una variable al basic.

Tras utilizar este comando la variable que hayas utilizado contendrá un número. Este número será 0, 64, 128 ó 255.

0 - Sprite colisionando  
64 - Detección de Colisiones desactivada  
128 - Sprite no en pantalla  
255 - Sprite no colisionando

SUGERENCIA : Cuando estés colocando sprites en pantalla aleatoriamente puedes probar para asegurarte que no has puesto un sprite sobre otro sprite.

```
10 A%=0:FOR I%=2 TO 30:|SGET,I%,3: |REPON,I%
20 |SPUT,I%,RND(8)*140,RND(8)*160+30
30 |COLLTEST,I%,@A%:IF A%=0 THEN |SPUT,I%:GOTO 20
40 NEXT
```

Si un sprite colisiona, se quitará y el programa probará otra vez.

**SUPERVISOR DE SPRITES****DETECCION DE COLISIONES****|REPORT,@v%**

Este comando se usa para comprobar todos los sprites que están en pantalla para ver si están colisionando. Utilizando este comando, se comprueba si colisiona cada sprite en pantalla que tenga el informe de colisión activado. Todos los sprites que estén colisionando se almacenan en una tabla. La variable v% te dice la cantidad de sprites que estaban colisionando cuando el comando fue ejecutado. Por tanto, si v% devuelve el valor 0, entonces ninguno de los sprites con el informe de colisión activado colisionó.

Usando este comando junto con el comando |NEXTREP el usuario tendrá una lista de los sprites que estaban colisionando.

**|REPORT,s1,s2,@v%**

Este comando es idéntico al comando |REPORT,@v% excepto que este comprueba un rango de sprites en lugar de todos los sprites. s1 es el número de sprite inicial mientras que s2 es número del último sprite.

**|NEXTREP,@v%**

Este comando obtiene el siguiente valor de la tabla que se guardó con el comando |REPORT. Cuando se utiliza este comando la variable v% contiene el número de un sprite que está colisionando. Cuando está en la variable, ese número de sprite se quitará de la tabla. Haciendo otro |NEXTREP,@v% el siguiente número de sprite será tomado de la tabla. Si la variable v% contiene el valor 255 entonces eso significa que todos los número de sprite de la tabla han sido sacados.

Un punto importante a recordar es que solo los sprites que estaban colisionando en el momento del comando |REPORT se guardan en la tabla. Si mueves cualquier sprite después del comando |REPORT y algunos de los nuevos movimientos provocan una colisión esos no se guardarán.

Puedes insertar otro comando |REPORT antes de que saques todos los números de sprite de la tabla. Los anteriores números de sprite que estaban en la tabla se borrarán para hacer hueco a los nuevos números.

Un punto importante a recordar es que esta rutina es bastante lenta. No debes utilizarla sin necesidad.

**|CLEAREP**

Este comando limpia la tabla de informes. Si se lanza un comando |NEXTREP después de este comando la variable contendrá 255. p.e. No queda ningún sprite más colisionando.

**SUPERVISOR DE SPRITES****DETECCION DE COLISIONES****|HIT,s,@v%**

Este comando permite al usuario saber qué sprite ha tocado a otro sprite. P.e. Puedes saber que el sprite 4 está colisionando pero no hay manera de saber con qué sprite. Utilizando este comando serás capaz de averiguarlo.

s es el número del sprite que sabes que está colisionando, v% es la variable guarda el número del sprite que ha tocado el sprite a. Si v% contiene 255 el sprite debe haber tocado parte del decorado o un sprite que no tiene activado el informe de colisión.

**|MOVEHIT,s,@b%**

Este comando puede utilizarse tras mover un sprite. s es the número de sprite, b% es la variable que contiene el resultado de la comprobación de colisión. Si b%=255 el sprite no estaba colisionando, si b%=0 el sprite ha colisionado con algo.

La ventaja de usar este comando en vez del comando |COLLTEST es que este comando es mucho más rápido. Durante la rutina de movimiento de sprites el sprite es comprobado por colisión automáticamente. Esta rutina simplemente comprueba dentro de la rutina de movimiento si ha habido una colisión.

Si mueves el sprite otra vez se desactivará la señal de colisión. Es importante que comprendas totalmente cómo funciona este comando para poder aprovechar todas sus ventajas.

Ejemplo.

Mueves los sprites 0 a 10 con el comando |MOVE,0,10. Cuando el sprite 0 se mueve no toca nada dejando por tanto la señal de colisión desactivada. Cuando el sprite 1 se mueve toca al sprite 0 activando la señal de colisión para el sprite 1. Si introduces un comando |MOVEHIT,0,@b%, que comprobaría si el sprite 0 está colisionando, la rutina devolverá el valor 255 (sprite no colisionando). Esto es porque cuando el sprite 0 se movió no tocó nada. Si introduces un comando |MOVEHIT,1,@b%, para ver si el sprite 1 está colisionando, la rutina devolverá el valor 0 (sprite en colisión).

Si en vez de introducir el comando |MOVEHIT después del comando |MOVE metes un comando |COLLTEST,0,@b% la rutina devolverá el valor 0 (sprite colisionando). Esto es porque el comando |COLLTEST comprueba el sprite después de que todos se hayan movido.

Por favor, nota que: El informe de colisión debe estar activado para que funcione esta rutina.

**SUPERVISOR DE SPRITES****NOTAS IMPORTANTES**

En todos los comandos detección de colisión hemos usado la variable v%. Hay una '@' delante de este comando. Usamos la '@' porque dice al BASIC que le vamos a mandar un valor desde el programa Sprites a tu propio programa. Es extremadamente importante incluir este carácter antes de tu nombre de variable. No hacerlo puede destruir tu programa.

Hemos utilizado v% como nombre de variable en los comandos detección de colisión. Tú puedes usar cualquier nombre de variable.

Hemos utilizado el carácter '%' tras el nombre de la variable. Esto es para que el Sprites las considere variables enteras. Debes utilizar siempre variables enteras mientras pases o recibas variables desde el programa Sprites. Por tanto es importante que entiendas totalmente como usar variables enteras.

Un punto importante a recordar sobre pasar variables enteras al BASIC es que la variable debe ser conocida por el programa. Al inicio del programa las variables deben contener un número, y este puede ser cero. ie a%=0:v%=0:f%=10

Es buena idea grabar siempre tus programs antes probarlos. Aunque se han hecho grandes esfuerzos para asegurar que cualquier error que hayas cometido sea informado por la rutina de errores, hay algunos casos en los que un error podría no ser informado por el programa.

**MENSAJES DE ERROR**

Si el programa encuentra un error, se mostrará el comando en el que se encontró el error junto con el número de error. El programa se detendrá y el cursor aparecerá en la pantalla. Ahora puedes entrar en el programa pulsando la tecla ESC una vez. Si pulsas otra tecla que no sea ESC el programa continuará.

Si no presionaste la tecla ESC por error y el programa continua, puedes detenerlo pulsando la tecla ESC dos veces. Ahora debes clasificar el error.

Los números de error se encuentran en el Apéndice II.

Después de pulsar la tecla ESC dos veces el ordenador mostrará BREAK IN LINE xxxx. Ésta no tiene por qué ser la línea en la que sucedió el error. El error puede haberse producido en la línea anterior. Si un error ocurre en la última orden de una línea cuando se pulsa la tecla ESC el ordenador mostrará el número de línea siguiente al de la línea errónea.

p.e. Tu programa comienza en la línea 10 y acaba en la línea 250 con saltos de línea de 10. Si el ordenador muestra BREAK IN LINE 60 el error puede estar en la línea 60, o en el último comando de la línea previa, la 50.

**SUPERVISOR DE SPRITES****COMANDOS MISCELANEOS****|SXPOS, s, @x%**

Este comando proporciona al usuario la coordenada X de un sprite concreto. s es the número del sprite, @x% es la variable donde quieres que se guarde la coordenada X del sprite.

**|SYPOS, s, @y%**

Este comando proporciona al usuario la coordenada Y de un sprite concreto. s es the número del sprite, @y% es la variable donde quieres que se guarde la coordenada Y del sprite.

**|SDRAW, s, @d%**

Este comando proporciona al usuario el número de gráfico que usa en ese momento el sprite. s es the número del sprite, @d% es la variable donde quieres que se guarde el número de gráfico.

**|SMEM, @m%**

Este comando proporciona al usuario la cantidad de memoria que los gráficos ocupan en memoria. Este valor siempre debe ser mayor que el valor del HIMEM. @m% es la variable en la que quieres guardar la información de memoria.

**|XDIR, s, @x%**

Este comando proporciona al usuario la velocidad a la que se mueve el sprite en el plano horizontal. s es el número del sprite, @x% es la variable en la que quieres guardar la velocidad del sprite.

**|YDIR, s, @y%**

Este comando proporciona al usuario la velocidad a la que se mueve el sprite en el plano vertical.  
s es el número del sprite, @y% es la variable en la que quieres guardar la velocidad del sprite.

**|CSPRITE, s**

Este comando limpiará el sprite s de la memoria. Si el sprite está en la pantalla cuando se ejecute el comando, el sprite se quitará. Ahora podrías si lo deseas, usar el comando |SGET para asociar este sprite a otro gráfico.

**|WAIT, n**

Este comando pausará tu programa. n es la cantidad de tiempo en 1/20 de Segundo que esperará. Si n es 100 el programa esperará 5 segundos.

**SUPERVISOR DE SPRITES****ANIMANDO LOS GRÁFICOS**

El usuario puede definir 16 secuencias de animación. Las secuencias de animación constan de una lista de números de gráfico en secuencia para las ocho direcciones en las que se puede mover el sprite.

Los gráficos que se utilicen en la animación deben tener todos las mismas dimensiones x e y. También deben estar en secuencia uno tras otro. Puedes definir las secuencias gráficas del sprite para las ocho direcciones en las que el sprite puede moverse. Podrías no querer usar todas las direcciones, puedes querer animar el sprite en sólo una o dos direcciones incluso, pero debes definir el movimiento para las ocho direcciones.

El Sprites seleccionará automáticamente la secuencia gráfica correcta dependiendo de la dirección en que se mueva el sprite, una vez que la secuencia gráfica se ha definido.

Varios sprites pueden utilizar la misma secuencia de gráficos, por tanto puedes tener mucho más que 16 sprites animados en la pantalla a la vez.

1		La manera de definir tu secuencia es la siguiente
2	5	Estas son las ocho direcciones en que se puede mover tu sprite.
3	9 6	
4	7	Debes definir una secuencia gráfica para cada dirección.
8		

**|SEQUENCE, sn, 1l, 1h, 2l, 2h, 3l, 3h, 4l, 4h, 5l, 5h, 6l, 6h, 7l, 7h, 8l, 8h**

Este comando es bastante largo, necesitas hacerle una vez por secuencia y no cada vez que quieras animar un sprite. La l y la h en el comando ejemplo son para los números low (bajo) y high (alto) de una secuencia.

sn es el número de secuencia. Éste no debe confundirse con el número de sprite. Aprenderás a asociar una secuencia a un sprite rápidamente. El número de secuencia debe estar entre 0 y 15.

Cuando definas una secuencia de low a high, cada gráfico dentro de esa secuencia debe tener las mismas dimensiones x,y. Tampoco debe faltar ningún gráfico. Las secuencias pueden repetirse para varias direcciones.

Ejemplo: Has diseñado una serie de gráficos de un hombre, los gráficos de 0 a 5 serán el hombre caminando hacia la izquierda, los gráficos de 6 a 11 son el hombre caminando a la derecha. El hombre puede caminar a izquierda y derecha. El comando debe ser algo así

```
|sequence,0,0,5,0,5,0,5,0,5,6,11,6,11,6,11,6,11
```

Este comando establece la secuencia 0. Si el hombre empezara a moverse en una diagonal la figura aún se animaría.

**SUPERVISOR DE SPRITES****ANIMANDO LOS GRÁFICOS**

Una vez que has establecido tus secuencias de animación necesitas asociar estas secuencias a un sprite. Debes hacer esto después de introducir un comando |SGET. Cuando asocias un sprite a un gráfico, el número de gráfico que debes utilizar debe ser más bajo que el número de gráfico más alto definido en la secuencia.

P.e. Has asociado el gráfico 10 al número de sprite 0. La secuencia que vas a utilizar debe tener un número de gráfico que sea mayor que 10.

Una vez que el sprite se ha animado el número de gráfico se incrementa hasta que es igual al último número de la secuencia, entonces se restablece al número de secuencia inicial. Si introduces un comando |SGET con un número de gráfico más alto que el número de secuencia más alto, se producirá un error.

Si deseas utilizar solo un gráfico para cada una de las direcciones, debes usar el mismo número de gráfico tanto para los valores alto como bajo.

Ejemplo.

Has diseñado ocho gráficos de una nave espacial, un gráfico para cada dirección en que se puede mover la nave. El comando |SEQUENCE debe parecerse a éste.

```
|SEQUENCE,1,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7
```

**|ANIMATE,s,sn**

Este es un comando que asocia un sprite a una secuencia de animación. s es el número de sprite, sn es el número de secuencia. El sprite debe haberse asociado a un gráfico con el comando |SGET.

**|ANIMOFF,s**

Este comando desactiva la animación del sprite s. El gráfico que el sprite estaba utilizando en el momento en que el comando se ejecutó estará congelado. ie no animado. Si el sprite se mueve ahora el sprite se asociará al último gráfico.

**NOTAS SOBRE ANIMACIÓN**

Si un sprite toca a otro sprite mientras se anima es posible que se atasquen. Esto se debe a la naturaleza de la animación y detección de colisiones. En circunstancias normales, tan pronto un sprite colisione con otro en un píxel el sprite rebotará. Sin embargo si en el momento de la colisión el gráfico se solapa con el otro gráfico en más de un píxel debido al cambio de forma del sprite por la animación, pueden llegar a atascarse. Hay una serie de cosas que puedes hacer para evitar esto (si sucediera). Al final del manual encontrarás algunas notas que te ayudarán.

**SUPERVISOR DE SPRITES****UTILIZANDO PROYECTILES**

Se ha introducido una serie de comandos en el programa Supervisor para permitir a los sprites disparar proyectiles. Los comandos pueden parecer complicados al principio, pero con la práctica usarás los comandos con facilidad en poco tiempo.

Primero debes diseñar un gráfico del proyectil. Haz éste de la misma forma que defines cualquier gráfico, será un pequeño círculo o un proyectil de forma más rectangular.

El proyectil debe entonces asociarse a cada GRÁFICO que desees que dispare proyectiles. La razón de que asocies un proyectil a un gráfico en vez de a un sprite es sencilla.

Si has definido un sprite para que sea una nave espacial (como en el ejemplo de animación), el sprite puede moverse en ocho direcciones. Cada una de estas ocho direcciones tiene un gráfico distinto asociado a ella. Como hemos asociado el proyectil al gráfico en vez de al sprite sabremos en que dirección disparar el proyectil, y también con cuanta corrección colocaremos el proyectil en la pantalla.

Seguramente encuentres confusa esta descripción. Iremos ahora viendo los comandos asociados a los proyectiles y veremos algunos ejemplos después. Hay un programa de demostración en el disco que se llama DEMO1 que puedes ojear para hacerte mejor idea de los comandos.

Hay tres tipos de proyectiles: 0, 1 y 2. Los proyectiles de Tipo 0 se disparan desde el sprite 0, los proyectiles de Tipo 1 se disparan desde el sprite 1 y los proyectiles de Tipo 2 se disparan desde los sprites 2 - 63.

**|MISSTYPE,ty,d**

Debes haber definido tu gráfico(s) para tu proyectil(es). Puedes tener un gráfico distinto para cada tipo de proyectil o todos los tipos pueden usar el mismo gráfico. Este comando informa al Supervisor de qué gráfico usar para el proyectil de cada tipo de proyectil.

```
ie |MISSTYPE,0,2:|MISSTYPE,2,2
```

Esto asocia el proyectil tipo 0 al gráfico 2 y proyectil tipo 2 al gráfico 2. Recuerda que los proyectiles de tipo 0 son disparados desde el sprite 0 y los proyectiles tipo 2 se disparan desde sprites 2 - 63. Ambos utilizan el gráfico 2 como gráfico del proyectil.

```
|MISSTYPE,0,3:|MISSTYPE,1,4
```

Esto asocia el proyectil tipo 0 al gráfico 3 y el tipo 1 al gráfico 4.

Los gráficos que diseñes como gráficos de proyectil deben estar por encima del gráfico 1.

**SUPERVISOR DE SPRITES****UTILIZANDO PROYECTILES****|MISSILE,Dn,ty,Xo,Yo,Xs,Ys**

Este es el comando que asocia un tipo de proyectil a un número de gráfico. Dn es el gráfico que dispara el proyectil. ty es el tipo de proyectil (0,1,2).

Si necesitas que el sprite 0 dispare proyectiles y el sprite 0 es animado con gráficos de 0 a 7, necesitaremos utilizar 8 comandos |MISSILE, comenzando con el |MISSILE,0,0,xo,yo,xs,ys y finalizando con |MISSILE,7,0,xo,yo,xs,ys. Recuerda que la segunda variable es el tipo de proyectil y no el número de gráfico del proyectil.

La ventaja de asociar un proyectil a un gráfico en vez de a un sprite para utilizarlo en distintos tipos de juego es obvio. El juego Asteroids tiene una nave que puede rotar a través de ocho direcciones. Cuando disparas un proyectil, lo disparas en la dirección que apunta y se mueve la nave. Sin embargo el juego Space Invaders tiene una nave moviéndose a izquierda y derecha pero los proyectiles siempre se disparan hacia arriba.

El comando |MISSTYPE debe introducirse antes de que empieces a usar este comando.

Xo y Yo son las correcciones X e Y del proyectil. Para calcular las correcciones debes usar un papel cuadriculado o una calculadora.

**CALCULANDO LAS CORRECCIONES DE LOS PROYECTILES**

Hay una utilidad en el disco para ayudarte a calcular las correcciones. Esta utilidad se llama "OFFSETS". Se incluyen instrucciones completas dentro del programa.

Para simplificar las cosas llamaremos al gráfico que disparará el proyectil gráfico 'A', y al gráfico mismo del proyectil 'B'.

Como sabrás, todos los sprites se dibujan en pantalla con el origen en su esquina superior izquierda.

Usando papel cuadriculado o calculándolo puedes hacer un dibujo del gráfico A. Debes dibujar ahora donde quieres que esté el gráfico B. Ten en cuenta que el gráfico A no debe estar tocando el gráfico B. Calcula entonces la distancia entre la esquina superior izquierda del gráfico A y la esquina superior izquierda del gráfico B. Ésta es la corrección, debes calcularla tanto para X como para Y.

Las correcciones pueden ser positivas o negativas.

Xs e Ys son las velocidades a las que se desplazará el proyectil. No intentes establecer las velocidades de un proyectil con el comando |SDIR.

**SUPERVISOR DE SPRITES****UTILIZANDO PROYECTILES**

Las velocidades x e y del proyectil tienen que ser números con signo. Si quieres que el proyectil se desplace a la izquierda a lo largo del eje x tienes que usar un número negativo para la velocidad x. Si quieres que el proyectil se desplace hacia abajo en la pantalla tienes que usar un número negativo para la velocidad y.

De momento has definido los gráficos a usar como proyectiles y también qué gráficos dispararán proyectiles.

Ahora tienes que decidir a qué distancia y con cuánta frecuencia un sprite puede disparar proyectiles. Los siguientes dos comandos determinan estos factores.

**|MISSDELAY, ty, de**

Este comando limita lo rápido que se disparan los proyectiles. ty es el tipo de proyectil, como se describió en la página anterior (1,2). de es el intervalo de tiempo entre disparos. de contendrá un número, este número determina cuantos movimientos debe hacer un proyectil antes de que el siguiente proyectil pueda ponerse en pantalla.

p.e. Si le das a de el valor 4, el proyectil recién disparado tiene que moverse 4 veces antes de que el siguiente proyectil pueda ponerse en pantalla.

Este comando tiene algunas ventajas. Permite al proyectil disparado previamente alejarse del siguiente proyectil disparado, evitando así que los proyectiles colisionen unos con otros. Otra ventaja es que puedes alterar la frecuencia con que un sprite puede disparar proyectiles a lo largo del juego.

El valor mínimo para de es 2 y el máximo valor es 63.

**|MISSDIST, di**

Este comando determina la distancia a la que se desplazan los proyectiles antes de desaparecer de la pantalla. di debe contener el número de movimientos que puede hacer un proyectil antes de ser retirado. ie. Si el valor de di es 5, después de la quinta orden |MOVE para ese sprite, el sprite desaparecerá. Si no quieres que el sprite sea retirado, simplemente dale un valor de 127.

El valor mínimo para di es 5 y el máximo valor es 127.

Los comandos que afectan a proyectiles que han aparecido tan avanzado este manual están en el mismo orden en que deben aparecer en tu programa. Si no los pones en este orden los proyectiles pueden no funcionar correctamente. El orden en que deben aparecer en tu programa es

|MISSTYPE, |MISSILE, |MISSDELAY, |MISSDIST, |BULLET

**SUPERVISOR DE SPRITES****UTILIZANDO PROYECTILES**

Ahora estamos listos para definir unos cuantos sprites para ser utilizados como proyectiles.

**|BULLET, ty, sp1, sp2**

Este comando define cuántos sprites serán usados como proyectiles. ty es el tipo de proyectil. Hay tres bancos distintos de proyectiles que puedes utilizar, 0,1 y 2.

El sprite 0 tiene su propio banco (0). El sprite 1 tiene su propio banco (1). Los sprites 2-63 tienen su propio banco (2).

sp1 es el sprite más bajo que puede ser utilizado como proyectil, sp2 es el sprite más alto que puede utilizarse como proyectil.

Un sprite puede tener más de un proyectil en la pantalla al mismo tiempo.

**EJEMPLO:**

Deseas que el sprite 0 tenga hasta 10 proyectiles en pantalla al mismo tiempo. Simplemente tienes que usar el comando

```
|BULLET,0,20,29
```

Este comando define los sprites de 20 a 29 para que sean sprites proyectil.

Conseguir que el sprite dispare un proyectil es sencillo. En los sprites 0 y 1 simplemente activa el botón de fuego utilizando los comandos |STIX o |KEB. Cada vez que se pulse el botón se disparará un proyectil.

La otra manera de disparar un proyectil, que funciona con los sprites de 0 a 63, es usar comando |SHOOT que se describe más adelante.

NOTA: Cuando defines los números de sprites más bajo y más alto que se usarán como proyectiles no debes usar estos sprites para otro propósito. También el número de sprite más bajo no debe ser menor de 2.

Los tres bancos de proyectiles no deben solaparse.

Cuando arrancas tu programa y comienzas a disparar los proyectiles puedes obtener un error 'MISSILE TYPE MISMATCH'. Esto sucede cuando intentas disparar un proyectil del tipo equivocado.

**Ejemplo.**

Has definido los gráficos de 0 a 4 para disparar proyectiles de tipo 0 con el comando |MISSILE. Si introduces un comando |SHOOT,2 y el sprite 2 es el gráfico 1 ocurrirá un error porque |SHOOT,2 intenta disparar un proyectil de tipo 2 y al gráfico 1 sólo se le permite disparar proyectiles de tipo 0. TEN CUIDADO !!

**SUPERVISOR DE SPRITES****UTILIZANDO PROYECTILES****|SHOOT, sp**

Este comando permite a un sprite disparar un proyectil donde se introduzca este comando, siendo sp el sprite que disparará el proyectil.

Si has definido sólo un pequeño número de sprites para utilizarse como proyectiles y has usado un intervalo pequeño encontrarás que el sprite disparará los proyectiles en ráfagas. Esto se debe a que sólo ese número de proyectiles que has definido puede estar en pantalla al mismo tiempo. Si continuas presionando el botón de fuego o usando el comando |SHOOT la acción será ignorada hasta que un proyectil se quite de la pantalla liberando así ese proyectil para usarse otra vez.

Cuando los proyectiles colisionan con el margen de la pantalla desaparecerán y podrán usarse otra vez. Si un proyectil colisiona con parte del decorado, o un sprite que no tiene el informe de colisión activado, el proyectil también desaparecerá y podrá usarse otra vez.

Si el proyectil toca un sprite con el informe de colisión activado (|REPON,s) el proyectil dejará de moverse y permanecerá en pantalla. El sprite con el que colisionó también dejará de moverse (si se estaba moviendo).

Tenemos ahora dos comandos que tratarán los proyectils cuando han colisionado con un sprite, mientras tengan el informe de colisión activado.

**|MISSHIT, @v%**

Este comando almacena en la variable v% el número de proyectils que están actualmente colisionando y están aún en la pantalla. Esto no tiene en cuenta el tipo de proyectil. Si v% contiene 0, ninguno de los proyectiles que están actualmente en pantalla está colisionando. Si v% contiene el valor 3 entonces 3 de los proyectiles en pantalla están colisionando con un sprite.

**|MISSHIT, ty, @v%**

ty es el tipo de proyectil (0,1,2). Cuando este comando se ejecuta, si un proyectil del tipo que especifiques en ty está colisionando, el proyectil se quitará de la pantalla y v% contendrá el número del sprite con el que el proyectil ha colisionado. El sprite que ha tocado el proyectil ahora podrá moverse otra vez. Eso es hasta que tu decidas que hacer con sprite que alcanzó el proyectil. Tienes el número de sprite en v%, por tanto, si deseas quitar ese sprite puedes utilizar el comando |SPUT,v%.

**SUPERVISOR DE SPRITES****UTILIZANDO PROYECTILES**

Es importante que compruebes regularmente si algún proyectil ha colisionado con algún sprite. Si no lo haces tu juego se acabará lleno de sprites que no se mueven. También si sólo has especificado para usar un pequeño número de proyectiles con el comando |BULLET no podrás usar el sprite del proyectil otra vez hasta que se quite de la pantalla.

**NOTAS PARA EL USO DE PROYECTILES**

Si te encuentras que cuando disparas tus proyectiles no funcionan correctamente, hay varias razones por las que podría ser.

- 1) La corrección que has elegido está demasiado próxima al gráfico principal. Cuando el proyectil es colocado en la pantalla colisiona con el sprite que lo dispara. Debes incrementar la corrección entre el sprite y el proyectil.
- 2) No estás moviendo el proyectil lo suficientemente rápido. Si el proyectil no se ha alejado lo suficiente del sprite que lo disparó y vuelves a disparar, el siguiente proyectil colisionará con el anterior. Debes incrementar el intervalo entre disparos con el comando |MISSDEL, o incluir más rutinas |MOVE para mover el proyectil más rápido, o incrementar la velocidad de movimiento del proyectil desde dentro del comando |MISSILE.
- 3) No compruebas las colisiones de los proyectiles con la suficiente frecuencia. En cuanto encuentres un proyectil colisionando debes actuar sobre él. Debes comprobar las colisiones de proyectiles REGULARMENTE. Recuerda que los proyectiles no se quitan solos al tocar algo, hasta que TU lo quites con el comando |MISSHIT.
- 4) Si encuentras con que el proyectil no se dispara correctamente cuando mueves el sprite en la dirección hacia la que estás disparando, debes incrementar la distancia de corrección entre el gráfico principal que dispara y el proyectil.

**|MERGE, s, d**

Este comando debe usarse con el sprite en la pantalla. Cambia el gráfico que está actualmente asociado al sprite con otro gráfico. s es el sprite que debe estar en pantalla, d es el nuevo número de gráfico a utilizar. Los gráficos antiguo y nuevo deben tener las mismas dimensiones X e Y.

Este comando puede utilizarse si no deseas usar los movimientos que permite el comando |STIX. El comando |STIX es el apropiado cuando quieres que el programa sea un juego del tipo Space Invaders, si mueves el joystick a la izquierda el sprite se moverá a la izquierda. Si deseas que tu programa sea un juego del tipo Asteroids, donde al mover el joystick a la izquierda rota la nave sin realmente moverla, entonces este comando es el que debes utilizar en conjunción con el comando |READSTIX. Debes usar el sprite 0 en este caso, aunque el sprite no sea controlado directamente por el joystick.

**SUPERVISOR DE SPRITES****MÁS COMANDOS MISCELANEOS****| SCENERY, dn, x, y**

Este comando permite al usuario poner gráficos en pantalla para actuar como decorado. dn es el número de gráfico, los números x e y son las coordenadas donde quieres ubicar el decorado. Puedes utilizar este comando para poner marcos alrededor de los juegos o poner decorado en la pantalla. La posición en que se coloca un gráfico en la pantalla no es recordada así que para quitarlo tienes que limpiar la pantalla. Puedes usar un gráfico cualquier número de veces. Puedes usar un gráfico que esté asociado a un sprite, no hay problema. Este comando no tiene nada que ver con los auténticos sprites.

El sistema de coordenadas utilizado para el es el mismo sistema que usamos para los sprites. RECUERDA, no puedes ir más allá del margen de la pantalla.

**| EXPLODE, spr, spot, inc, lim, del**

Este comando hace que un sprite explote en la pantalla. Mientras la explosión está teniendo lugar todos los objetos en pantalla dejarán de moverse hasta que la explosión acabe.

spr es el sprite que tiene que explotar, debe estar en pantalla. spot es el número de píxeles que se colocan en la pantalla antes de que la explosión se expanda. inc es el tamaño del incremento de la explosión en píxeles. lim es el máximo tamaño de la explosión en píxeles. del es un retraso para ralentizar la desaparición de la explosión.

La única manera de conseguir utilizar este comando es probar con diferentes valores para ver el efecto que tiene en pantalla.

```
spots: rango 0 - 32767 , media 100 - 400
inc:    rango 1 - 32   , media  2 -- 10
lim:    rango.10 - 128 , media  20 - 100
del:    rango 1 - 255  , media  10 - 100
```

Un programa para probar el comando |EXPLODE podría ser como este

```
10 MODE 0:LOCATE 1,1:PRINT "0":|DGET,0,7,7
20 |SGET,2,0:A%=0:B%=0:C%=0:D%=0
30 CLS
40 INPUT "SPOTS ",A%
50 INPUT "INC ",B%
60 INPUT "LIM ",C%
70 INPUT "DEL ",D%
80 WHILE INKEY$=""
90 |SPUT,2,75,95:|EXPLODE,2,A%,B%,C%,D%
100 WEND:|SPUTALL:GOTO 30
```

**SUPERVISOR DE SPRITES****MÁS COMANDOS MISCELANEOS****|FRAME**

Este comando espera a la interrupción de refresco de pantalla antes de permitir al programa continuar. Es útil para ralentizar los gráficos si se mueven demasiado rápido. También hace que los graphics parezcan moverse más suavemente en algunas circunstancias, aunque creo que no hay demasiada diferencia.

**|STUCK,s,@v%**

Este comando dice al usuario si un sprite concreto se ha atascado con otro sprite o parte del decorado. Esto puede suceder durante la animación, y se describe en la sección de animación. Si un sprite se ha atascado debes |SPUT para quitarlo de la pantalla, cambiar las coordenadas y |SPUT otra vez en pantalla. s es el número de sprite a comprobar. v% contendrá un número 0 (no atascado) o 1 (atascado).

NOTA: Este rutina no es infalible. Sólo devolverá el valor de atascado si no puede moverse en cierta dirección. Si estás utilizando un joystick para controlar un sprite e intentas mover el sprite dentro del decorado el aviso de atasco puede activarse para ese sprite aunque aún pueda moverse en otras direcciones.

Un sprite que no es controlado por joystick/teclado intentará liberarse el solo moviéndose en todas direcciones. Sólo cuando no puede moverse en ninguna dirección se activará el aviso de atasco. Un sprite te puede parecer atascado, sin embargo puede que este rebotando muy pocos píxeles y por tanto el aviso de atasco no se activa (el sprite se está moviendo).

**|SPEEK,x,y,@v%**

Este comando prueba un píxel en pantalla. x e y son las coordenadas de pantalla a probar. v% contiene el número de tinta de ese píxel concreto. Este comando sólo funciona en Modo 0 o Modo 1.

**|SPOKE,x,y,i**

Este comando dibuja un píxel en la pantalla. x e y son las coordenadas de pantalla, i es la tinta. Debes utilizar el sistema de coordenadas del Sprites (No el del Amstrad).

**SUPERVISOR DE SPRITES****UTILIZANDO SONIDOS AUTOMATICOS**

El Supervisor permite a los sprites generar sonidos automáticamente de acuerdo a unas condiciones. Cuando una de esas condiciones se cumple el Supervisor genera un sonido. Hay dos comandos que manejan la generación automática de sonidos.

**|SOUND,nn,ch,ae,te,tp,np,ia,du**

Este comando es idéntico al comando sound estándar del Amstrad, con dos excepciones. La primera variable (sn) es el número de sonido. Puedes definir hasta 16 efectos de sonido diferentes (0 - 15). La otra diferencia entre este comando y la versión del Amstrad, es que tienes que utilizar todas las variables.

sn - Número del Sonido (0 - 15)  
 ch - Canales a utilizar y requerimientos de encuentro (0 - 255)  
 ae - Envolvente de Amplitud a utilizar (0 - 15)  
 te - Envolvente de Tono a utilizar (0 - 15)  
 tp - Período de Tono (1 - 4095)  
 np - Período de Ruido (0 - 31)  
 ia - Amplitud Inicial (0 - 15)  
 du - Duración del sonido (-32768 - +32767)

Por favor busca en el manual de usuario de tu Amstrad para una descripción completa de este comando.

Como dijimos más arriba debes usar las 8 variables. Si no necesitas una cierta función, debes utilizar el valor cero.

Para definir las envolventes de amplitud y tono, debes utilizar los comandos estándar Amstrad ENV y ENT.

Una vez que hayas definido los sonidos con los comandos de arriba necesitas asociarlos a los sprites y sus acciones.

**|SATTR,sp,ac,sn**

Puedes conseguir que los sprites emitan un sonido cuando les suceda alguna de las siguientes situaciones

- 1 Disparar un proyectil
- 2 Explotar
- 3 Rebotar con el borde de la ventana
- 4 Rebotar contra algo

sp es el número de un sprite ya definido, ac es la acción (1-4 de arriba) en la que el sprite emitirá el sonido. sn es el número de un sonido que has creado con el comando |SOUND.

p.e |SATTR,0,1,2: Emite el sonido 2 cuando el sprite 0 dispara un proyectil.

**SUPERVISOR DE SPRITES**UTILIZANDO NODOS

En ciertos juegos (Pacman etc) los sprites sólo pueden moverse dentro de los confines de un laberinto. Hay una serie de comandos integrados en el Supervisor que permiten al usuario simular este tipo de juego.

Para crear un laberinto para los sprites, tenemos que definir un número de nodos. Un nodo es el punto de un laberinto en que se permite a los sprites cambiar de dirección.

Para mostrar esto, imagina un cuadrado grande en la pantalla. El cuadrado tiene cuatro esquinas, por tanto necesitará cuatro nodos. (superior izquierda, superior derecha, inferior izquierda e inferior derecha).

Para informar al Supervisor de la forma de tu laberinto necesitamos especificar tres variables. Las dos primeras variables son las coordenadas x e y de la posiciones en que los sprites pueden cambiar de dirección. La tercera variable es un valor que informa a los sprites de las direcciones que pueden tomar en ese nodo en particular.

Para facilitarte las cosas hemos escrito un programa que te ayudará a definir tus nodos. El programa tiene sus propias instrucciones. Encontrarás el programa en la cara uno de tu disco de sistema. Para sacar todo el partido a este programa intentaré explicarte los nodos más detalladamente.

Un nodo es un punto invisible en la pantalla. No puedes ver un nodo, sin embargo puedes ver sus efectos. El programa de demostración DEMO5, te muestra los nodos en acción. En esta demostración los sprites se mueven por la pantalla y parecen cambiar de dirección en puntos fijos, pero invisibles. Estos puntos son nodos. Para hacer el juego jugable necesitas mostrar los caminos que los sprites pueden tomar. Si ahora pruebas la DEMO6 en ella hemos incluido estos caminos. Los caminos se colocan para hacer el juego jugable. Imagínate intentando jugar al Pacman sin poder ver dónde puedes cambiar de dirección. Hasta que hagas estos caminos visibles es un trabajo tedioso, pero los resultados merecen la pena.

Como dijimos más arriba, necesitas especificar las coordenadas x e y para cada nodo. También necesitas especificar las direcciones por las que el sprite puede salir del nodo. Aunque el programa diseñador de nodos te sera de gran ayuda, te sugerimos que tomes un papel cuadriculado y MUY cuidadosamente diseñes tu laberinto. Tienes que tener en cuenta el ancho y el alto de los sprites que se usarán dentro del laberinto.

p.e. Si tienes un sprite con un ancho de 12 píxeles y estás diseñando un laberinto en modo 0 con el tamaño de pantalla estándar del Amstrad y los sprites tienen establecido que reboten en el margen de la pantalla, no puedes poner un nodo en el punto 154,40 simplemente porque el sprite nunca alcanzará ese punto. Rebotará cuando alcance 147,40.

**SUPERVISOR DE SPRITES****UTILIZANDO NODOS**

También tienes que tener en cuenta el alto y ancho de los sprites cuando estés diseñando los caminos que pueden tomar los sprites. Si el camino es demasiado estrecho o bajo el sprite colisionará con el camino y se atascará.

Los cambios de dirección se harán siempre que the esquina superior izquierda del sprite tenga las mismas coordenadas que el nodo. Si has puesto un nodo en 120,40 y has puesto el sprite en pantalla en 111,40 moviéndose hacia la derecha con una velocidad de dos el sprite se saltará el nodo moviéndose de 119,40 a 121,40. Tienes que procurar que la velocidad del sprite no le permita saltarse el nodo. Puedes beneficiarte de esto, permitiendo a ciertos sprites ignorar nodos concretos mientras que a otros sprites no.

Cuando estás utilizando el programa diseñador de nodos y vas a ver tus nodos, los sprites sólo cambiarán de dirección cuando toquen un nodo. Cuando los caminos que pueden tomar los sprites están dibujados en la pantalla y no hay un nodo en el punto en que se cruzan, los sprites no cambiarán de dirección. Cuando todos los caminos están dibujados, las direcciones por las que los sprites pueden salir de los nodos se muestran con un línea de color claro saliendo del nodo.

Los nodos (cuando se hayan diseñado) ocupan cierto espacio en memoria. El cantidad de memoria que utilizan se muestra después de que hayas guardado los nodos (para usarlos con el Supervisor) en disco. Tienes que tomar nota de este número. Cuando estes calculando el valor para tu comando MEMORY, una vez que lo tengas tienes que restarle la cantidad de memoria que usarán los nodos. Tienes que usar este resultado para el commando MEMORY.

Veamos ahora los comandos del Supervisor relacionados con los nodos.

**|NODE, "nombredefichero"**

Este comando carga los nodos que hayas definido con el programa diseñador de nodos en memoria. Tienes que haber tenido en cuenta la memoria extra que ocuparán los nodos y aplicada en el comando MEMORY.

Sólo puedes cargar los nodos en memoria DESPUES DE QUE TODOS los gráficos hayan sido almacenados en memoria. Si después de ejecutar este comando intentas cargar gráficos o DGETear gráficos en memoria se mostrará un mensaje de error.

El comando |RESET no limpia los nodos de la memoria. El comando |ERASE limpia los nodos así como los gráficos.

Cuando especificas el nombre de fichero no tienes que usar extensión. P.e. JUEGO1.nde. La extension se añadirá automáticamente. Este comando carga ficheros con la extension .NDE .

**SUPERVISOR DE SPRITES****UTILIZANDO NODOS****|NODESPEED, sp, xs, ys**

Este comando establece la velocidad a la que los sprites se moverán a través del laberinto, sp es el número de un sprite ya definido. Xs e ys son las velocidades del sprite. La velocidad especificada debe ser en ambas con números positivos entre 1 y 10.

Las velocidades de nodo no se usarán hasta que un sprite toque su primer nodo. Por tanto, si inicialmente no vas a SPUT un sprite directamente en un nodo, tendrás que definir también la velocidad del sprite con el comando |SDIR. Una vez que un sprite ha tocado un nodo los valores utilizados en el comando |SDIR se sobrescribirán.

No necesitas especificar las velocidades para sprite 0 y sprite 1. Las velocidades para estos dos sprites se toman de los comandos |STIXSPEED y |KEBSPEED.

Por favor, recuerda lo que comentamos más arriba sobre las velocidades de los sprites, asegúrate de que las velocidades que definas no permitan a los sprites saltarse los nodos (si no quieres específicamente que lo hagan).

**|NODEATTR, sp, at**

Este comando determina como se moverán por el laberinto los sprites que has definido. El sprite 0 y el sprite 1 se mueven por el laberinto controlados por joystick/teclado. Sin embargo, el resto de los sprites (2 - 63) necesitan que les digamos como moverse. Este comando hace exactamente eso. sp es el número de un sprite ya definido (2 - 63), at es el número de atributo que determina como se moverá el sprite.

Los sprites normales (2 - 63) pueden moverse aleatoriamente por el laberinto, perseguir al sprite 0/1, o huir del sprite 0/1.

Para determinar cual de estas acciones tomará el sprite, por favour utiliza la tabla siguiente.

- 1 - Movimiento aleatorio
- 2 - Perseguir al sprite 0/1
- 3 - Huir del sprite 0/1

Debes utilizar un de estos números para la variable.

NOTA: Si vas a utilizar la opción 2 o la opción 3 los nodos pueden perseguir/huir del sprite 0 del sprite 1, ello no pueden perseguir/huir de los dos sprites. El sprite que tengan que perseguir o huir está definido con el comando de la página siguiente. Si el sprite que tienen que perseguir/huir no está en la pantalla, los sprites asumiran la opción 1 hasta que el sprite 0 o el sprite 1 vuelva a estar en la pantalla.

**SUPERVISOR DE SPRITES****UTILIZANDO NODOS****|NODESPRITE, sp**

Este comando determina qué sprite perseguirán/huirán todos los otros sprites. sp es el número de sprite 0 o 1. Si sp es 0 los sprites perseguirán/huirán del sprite 0, que es controlado por el joystick. Si sp es 1 los sprites perseguirán/huirán del sprite 1, que es controlado por teclado.

**|NODEON, sp**

Este comando activa las funciones de laberinto para un sprite concreto. No todos los sprites necesitan seguir las direcciones que dictan los nodos. Algunos sprites pueden ignorar los nodos totalmente. Los sprites que hayan sido definidos como proyectiles también ignorarán los nodos. Si intentas utilizar este comando en un sprite proyectil se mostrará un mensaje de error.

Antes de utilizar este comando tienes que definir las NODESPEED y NODEATTR para el sprite.

sp es el número del sprite que se moverá por el laberinto. No puedes utilizar este comando con los sprite 0 y sprite 1 ya que ellos se activan automáticamente cuando se ha definido un laberinto.

**|NODEOFF, sp**

Este comando desactiva las funciones de laberinto en el sprite sp. Sp puede ser cualquier número de sprite entre 2 y 63.

**|NODEALTER, x, y, nd**

Este comando cambia las direcciones por las que un sprite puede salir de un nodo. No puedes añadir nodos extra con este comando, sólo cambiar nodos existentes.

x e y son las coordenadas de un nodo ya definido. nd es la variable que guardará las nuevas direcciones por las que el sprite puede salir del nodo. Para calcular las nuevas direcciones utiliza la tabla siguiente.

- 1 - Arriba
- 2 - Abajo
- 4 - Izquierda
- 8 - Derecha

Decide por qué direcciones pueden dejar ahora el nodo los sprites y suma sus valores, que están a la izquierda de la dirección. El total será el valor para la variable nd. No puedes usar el valor 0.

Ej Nueva dirección izquierda, derecha, arriba: 1 + 4 + 8. Por tanto nd = 13

**SUPERVISOR DE SPRITES****UTILIZANDO NODOS****|MAZEON**

Este comando activará el laberinto en todos los sprites. Después de que hayas establecido los nodos y definido todos sus atributos y velocidades y activados con el comando |NODEON, no actuarán como laberinto hasta que se ejecute este comando.

**|MAZEOFF**

Este comando desactiva el laberinto. Si deseas activar el laberinto otra vez no necesitas activar todos los sprites individualmente, simplemente utiliza este comando. Todos los sprites que estaban activados cuando se ejecutó este comando se recuerdan.

Estos dos comandos son los interruptores para la utilización de ficheros de laberinto.

**MÁS NOTAS SOBRE SPRITES Y NODOS**

Los sprites no pueden moverse en diagonal por el laberinto. Si un sprite se ha definido para utilizar el laberinto pero no ha tocado su primer nodo puede moverse en diagonal. Incluso si un sprite se ha activado con el comando |NODOON no seguirá las reglas de laberinto hasta que toque su primer nodo.

Para que los sprites utilicen el laberinto correctamente, necesitan ser SPUTados en un nodo o SPUTados en la pantalla con una dirección establecida con el comando SDIR para que puedan tocar un nodo.

Si el sprite 0 o el sprite 1 son SPUT en pantalla y no están en un nodo pueden moverse libremente por la pantalla hasta que toquen un nodo.

Cuando un sprite entra en un nodo, el sprite no saldrá por la dirección en que ha entrado si no hay una salida por ahí en el nodo. P.e. los sprites no invertirán su dirección.

Puedes crear sistema de un solo sentido dentro del maze. Por favor, mira el ejemplo de abajo.

Considera un cruce en T, desde el nodo el camino va hacia arriba y hacia abajo de la pantalla y otro camino se conecta al nodo desde la izquierda. Si el nodo en el punto de unión sólo permite al sprite salir por arriba o por abajo, has creado un sistema de un solo sentido. Un sprite que vaya hacia arriba y llegue al nodo sólo podrá continuar hacia arriba. Otro sprite viniendo desde la izquierda alcanza el nodo y sólo podrá ir arriba o abajo. Construyendo varios cruces, puedes conseguir un laberinto muy complejo.

**SUPERVISOR DE SPRITES****VELOCIDAD DE TU PROGRAMA**

No se han escatimado esfuerzos para hacer el código maquina del Supervisor tan rápido como sea posible. El problema que tiene este tipo de programa es que cuanto más genéricas haces las rutinas más lentas son.

Hemos tratado de acelerar la ejecución del programa todo lo posible automatizando muchos de los comandos. Hay un punto en el que tienes que poner el límite.

El problema escribiendo juegos en BASIC es la velocidad. Necesitas utilizar el BASIC para utilizar los comandos del Supervisor. Esto ralentiza los programas. Esto te obliga a escribir tus juegos lo más eficientemente posible.

Intenta que el Supervisor haga el trabajo por tí. Incorpora tantos comandos automatizados como te sea posible (MISSILE, SOUND EFFECTS, MOVEHIT etc).

Algunos de los comandos utilizados dentro del Supervisor son relativamente lentos comparados con otros. Por ejemplo, el comando REPORT tiene que comprobar cada sprite de la pantalla. Planea tu juego de manera que solo cierto grupo de sprites pueda estar colisionando y limita el comando REPORT a este grupo de sprites.

Intenta mantener los sprites tan pequeños como sea posible.

Cuando defines sprites mantenlos en un bloque. Mantén los números de sprite tan bajo como sea posible. Si defines el sprite 0 y el sprite 63 e introduces un comando que opera sobre todos los sprites ie MOVEALL todos los sprites necesitarán se comprobados (incluso aquello que no se hayan definido).

Si después de todo esto tu programa va a paso de tortuga hay varias alternativas.

- 1) Vendemos un compilador para convertir tu programa en BASIC en código máquina. Esto mejorará drásticamente la velocidad de tu programa.
- 2) Si tienes conocimientos de código maquina, vendemos una version reducida del Supervisor con documentación de todos los puntos de entrada e información de variables.

Por favor, contáctanos para disponibilidad de las opciones de arriba descritas.

Si tienes problemas con el Supervisor, por favor, escríbenos e intentaremos resolver el problema lo más rápidamente posible.

Si tienes algún comentario o idea útil para mejorar el programa, por favor, contáctanos. Si nos gusta la idea recibirás una copia revisada y actualizada del programa gratis.

SUPERVISOR DE SPRITESAPENDICE IDIMENSIONES DE PANTALLAMODE 0

PANTALLA	CARACTERES	DIMENSIONES
N	20 x 25	160 x 200
0	19 x 16	152 x 208
1	18 x 28	144 x 224
2	17 x 30	136 x 240
3	16 x 31	128 x 248
4	21 x 24	168 x 192
5	22 x 23	176 x 184
6	23 x 22	184 x 176
7	24 x 21	192 x 168

MODE 1

PANTALLA	CARACTERES	DIMENSIONES
N	40 x 25	320 x 200
0	38 x 16	304 x 208
1	36 x 28	288 x 224
2	34 x 30	272 x 240
3	32 x 31	256 x 248
4	42 x 24	336 x 192
5	44 x 23	352 x 184
6	46 x 22	368 x 176
7	48 x 21	384 x 168

**SUPERVISOR DE SPRITES****APENDICE II****MENSAJES DE ERROR**

01 El número de variables utilizado en el comando es incorrecto.  
02 El número de gráfico es mayor de 63.  
03 El número de sprite es mayor de 63.  
04 El gráfico no ha sido definido.  
05 El sprite no ha sido definido.  
06 El número de inicio es mayor que el número de final.  
07 El sprite no está en pantalla.  
08 El sprite ya está en pantalla.  
09 El gráfico ya ha sido definido.  
10 El sprite ya ha sido definido.  
11 El sprite está fuera de las coordenadas de la ventana.  
12 La coordenada X es igual a cero.  
13 La coordenada Y es igual a cero.  
14 La coordenada X es mayor de 32.  
15 La coordenada Y es mayor de 32.  
16 La variable usada es mayor de 7.  
17 La velocidad X es mayor de 10.  
18 La velocidad Y es mayor de 10.  
19 El modo de pantalla es mayor de 1.  
20 La variable del margen es igual a cero.  
21 La variable del margen es mayor de 4.  
22 El atributo colisión es igual a cero.  
23 El atributo colisión es mayor de 3.  
24 El estado de colisión es mayor de 1.  
25 La variable utilizada es mayor de 63.  
26 La velocidad izquierda es mayor de 10.  
27 La velocidad derecha es mayor de 10.  
28 La velocidad arriba es mayor de 10.  
29 La velocidad abajo es mayor de 10.  
30 La ventana es demasiado ancha.  
31 La ventana es demasiado estrecha (X).  
32 La ventana es demasiado baja.  
33 La ventana es demasiado estrecha (Y).  
34 La ventana es demasiado alta.  
35 Los colores no han sido definidos.  
36 La secuencia es mayor de 15.  
37 La secuencia de animación es mayor de 10 gráficos.  
38 El gráfico dentro de la secuencia no ha sido definida.  
39 La dimensión X dentro de la secuencia es distinta.  
40 La dimensión Y dentro de la secuencia es distinta.  
41 La secuencia no ha sido definida.  
42 El número del sprite gráfico actual es mayor que el número más alto de la secuencia.  
43 La coordenada X actual es distinta a la coordenada X dentro de la secuencia.  
44 La coordenada Y actual es distinta a la coordenada Y dentro de la secuencia.  
45 El tipo de proyectil no ha sido definido.  
46 El tipo de proyectil es mayor de 2.  
47 El tipo de proyectil ya está definido.  
48 El gráfico del proyectil no ha sido definido.  
49 La corrección X del proyectil es mayor de 40.  
50 La corrección Y del proyectil es mayor de 40.

**SUPERVISOR DE SPRITES**  
**MENSAJES DE ERROR (CONT)**

51 La distancia es menor de 2.  
52 El color es mayor de 26.  
53 La tinta es mayor de 15.  
54 El número de puntos es igual a cero.  
55 El incremento de la explosión es mayor de 32.  
56 El límite de la explosión es mayor de 128.  
57 La variable no puede ser cero.  
58 La variable es mayor de 9.  
59 La velocidad X del proyectil es mayor de 10.  
60 La velocidad Y del proyectil es mayor de 10.  
61 El gráfico del proyectil es mayor de 63.  
62 El gráfico de proyectil es igual a 0.  
63 El sprite 0 y el sprite 1 no pueden ser proyectiles.  
64 Hay más de 10 proyectiles definidos.  
65 El sprite utilizado para un proyectil ya está definido.  
66 El tipo del proyectil no concuerda.  
67 El intervalo del proyectil es mayor de 63.  
68 La distancia del proyectil es mayor de 127.  
69 El intervalo del proyectil es igual a 0.  
70 No puedes hacer eso a un sprite de proyectil.  
71 No puedes alterar eso en el sprite 0 o el sprite 1.  
72 Desbordamiento de Proyectil. Tipo 0.  
73 Desbordamiento de Proyectil. Tipo 1.  
74 Desbordamiento de Proyectil. Tipo 2.  
75 No puedes utilizar parpadeo de colores.  
76 La coordenada está fuera de la ventana (x).  
77 La coordenada está fuera de la ventana (y).  
78 Los gráficos deben ser definidos antes de los nodos.  
79 La velocidad no puede ser igual a cero.  
80 Los nodos no se han cargado en el Supervisor.  
81 Las velocidades de los sprites del nodo no se han definido.  
82 Los atributos de los sprites del nodo no se ha definido.  
83 El atributo del nodo no puede ser igual a cero.  
84 El atributo del nodo es mayor de 31.  
85 El número de sonido no puede ser mayor de 15.  
86 La envolvente de amplitud no puede ser mayor de 15.  
87 La envolvente de tono no puede ser mayor de 15.  
88 El período de tono no puede ser mayor de 4095.  
89 El período de tono no puede ser mayor de 31.  
90 La amplitud inicial no puede ser mayor de 15.  
91 La acción del sonido del sprite no puede ser igual a cero.  
92 La acción del sonido del sprite no puede ser mayor de 4.  
93 El sonido no se ha definido.  
94 El número de sprite es mayor de 1.  
95 La coordenada x no se ha definido.  
96 La dirección del nodo no es correcta.  
97 La coordenada y no se ha definido.  
98 La coordenada x es mayor de 383.  
99 La coordenada y es mayor de 247.

**SUPERVISOR DE SPRITES**  
**APENDICE III**  
**NOTAS USUARIO AVANZADO**

Debido a la técnica utilizada por el Supervisor para detectar las colisiones, las colisiones entre ciertas tintas serán ignoradas. Podrías beneficiarte de esto. La siguiente tabla muestra que tintas no registran una colisión cuando colisionan. Encuentra las dos tintas utilizadas en la parte superior y en el lateral izquierdo de la tabla, sigue las líneas hasta donde se cortan las dos tintas. Si hay una equis en la intersección el Supervisor no registrará una colisión. Hemos usado la notación Hexadecimal en los márgenes de la tabla, si no sabes hexadecimal sustituye A=10, B=11, C=12, D=13, E=14, F=15.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 :	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 :	X	0	X	0	X	0	X	0	X	0	X	0	X	0	X	0
2 :	X	X	0	0	X	X	0	0	X	X	0	0	X	X	0	0
3 :	X	0	0	0	X	0	0	0	X	0	0	0	X	0	0	0
4 :	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0
5 :	X	0	X	0	0	0	0	0	X	0	X	0	0	0	0	0
6 :	X	X	0	0	0	0	0	0	X	X	0	0	0	0	0	0
7 :	X	0	0	0	0	0	0	0	X	0	0	0	0	0	0	0
8 :	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0
9 :	X	0	X	0	X	0	X	0	0	0	0	0	0	0	0	0
A :	X	X	0	0	X	X	0	0	0	0	0	0	0	0	0	0
B :	X	0	0	0	X	0	0	0	0	0	0	0	0	0	0	0
C :	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
D :	X	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0
E :	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F :	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Cuando los sprites se solapan su color cambia. La siguiente tabla muestra a qué color cambiará el sprite. En la página siguiente te contamos las ventajas de utilizar estas tablas.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 :	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1 :	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2 :	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3 :	3	2	1	0	7	6	5	6	B	A	9	8	F	E	D	C
4 :	4	5	6	7	0	1	2	4	C	D	E	F	8	9	A	B
5 :	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6 :	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7 :	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8 :	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9 :	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A :	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B :	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C :	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D :	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E :	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F :	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

**SUPERVISOR DE SPRITES**  
**APENDICE III**  
**NOTAS USUARIO AVANZADO**

Hemos establecido que cuando un sprite colisiona con otro sprite ambos cambian de dirección. Esto no es estrictamente cierto, hay algunas maneras de evitar esto.

Considera un juego de plataformas. El carácter probablemente necesite subir una escalera. La escalera no puede tener la misma tinta que el fondo porque si no no se vería. Si miras en las tablas de la página anterior notarás que las colisiones entre ciertas combinaciones de tintas pueden no ser detectadas. Si el carácter principal tiene una de estas tintas y la escalera otra, el carácter podría moverse por la escalera sin que la rutina de colision se entere. Sin embargo aún hay un problema. Cuando las dos tintas se mezclan se forma una nueva tinta. Aquí es donde entra la segunda tabla. Si le damos a la nueva tinta el mismo color que el carácter principal nadie podrá distinguir cuando el carácter se está moviendo sobre la escalera. Hemos utilizado una escalera como ejemplo, puedes aplicar esta técnica a muchos juegos.

Ejemplo:

El sprite del carácter principal se define como tinta 1 y la escalera se define como tinta 10 (A). Vemos en la primera tabla que una colisión entre estas dos tintas no se detectará.

Mirando en la segunda tabla vemos que una colisión entre la tinta 1 y la tinta 10 (A) resulta en un cambio de tinta a la tinta 11 (B). Si le damos a la tinta 11 el mismo color que la tinta 1, no notarás el cambio de color.

Llevando este proceso un paso más allá, si estudias la primera tabla notarás que la tinta 8 no detectará colisiones con las tintas 0 a 7. Mirando a la segunda tabla cuando las tintas 0 a 7 colisionan con la tinta 8 producen una tinta mayor de 7. Por tanto tenemos dos colores para colores de fondo, sin embargo limitamos la cantidad de colores de primer plano a 7.

Si encuentras aceptable tener sólo siete colores de primer plano y quieres tener dos colores de fondo, sigue estas sencillas instrucciones.

Las dos tintas de fondo deben ser 0 y 8. Cuando hayas decidido qué colores necesitas para tus colores de primer plano, estos colores también deben copiarse en las tintas 9 a 15.

p.e.

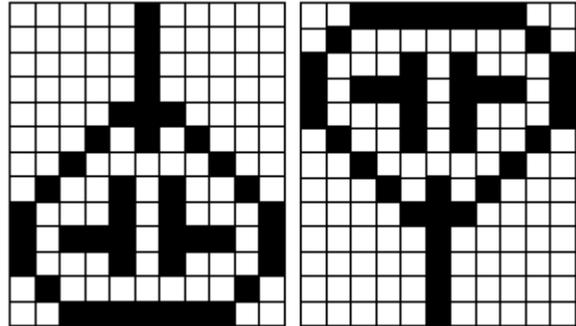
tinta 1 = tinta 9, tinta 2 = tinta 10, tinta 3 = tinta 11,  
tinta 4 = tinta 12, tinta 5 = tinta 13, tinta 6 = tinta 14,  
tinta 7 = tinta 15

**SUPERVISOR DE SPRITES**

Si te encuentras con que tus sprites animados se atascan con otros o con el decorado, las siguientes instrucciones pueden solventar el problema.

Bajo circunstancias normales tus gráficos deben parecerse a este, sin espacios alrededor del sprite.

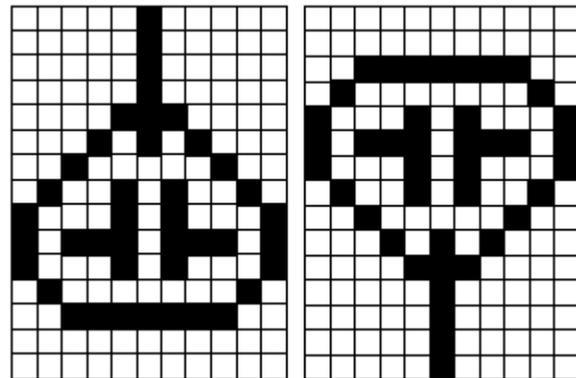
El problema del atasco sucede cuando el sprite cambia de dirección. Si el primer sprite sube en la pantalla y toca un objeto, podría darse la vuelta (como se ve en el gráfico B).



Tienes un tanque que está animado para moverse arriba y abajo en la pantalla dibujada arriba. Si te encuentras con que el tanque se atasca con otro objeto el mejor método de evitar esto es hacer el área alrededor del sprite más grande.

Cuando el tanque se da la vuelta el nuevo gráfico podría meterse más en el objeto que lo que se metía el anterior.

Para evitar este problema debes dejar un espacio detrás del gráfico que está expuesto a atascarse. Si el atasco sigue sucediendo prueba a aumentar el espacio a 3 ó 4 píxeles. Si esto no soluciona el problema necesitarás rediseñar tu gráfico.



**SUPERVISOR DE SPRITES****APENDICE III****NOTAS USUARIO AVANZADO**

La página anterior sugiere una manera de evitar colisiones haciendo el tamaño del gráfico más grande que el actual. Hay otro método para evitar que los gráficos animados se queden atascados.

Si dibujas una caja alrededor de tu gráfico en una tinta que siempre detecte colisiones p.e. tinta 7 y pones esta tinta al mismo color que el color de fondo, el usuario nunca verá la caja, si embargo el Supervisor detectará cuándo la caja está colisionando permitiendo al sprite cambiar de dirección llegar a atascarse.

Hemos utilizado este principio en las demostraciones de nodos (DEMO5 and DEMO6).

Si te estás preguntes cómo conseguimos que el decorado se parezca al que hicimos en la DEMO6, hemos usado el Advanced Art Studio.

No es un trabajo sencillo y no llevaría muchas páginas explicartelo paso a paso. Te explicaremos los principios básicos para que puedas hacerlo por tí mismo.

Si vas a cargar gráficos en el Supervisor con el comando |DRAW, tienes que anotar los colores que has seleccionado. Cuando entres en el Art Studio necesitarás alterar la paleta de colores para adecuarlos a los seleccionados en el Diseñador del Sprites.

Después diseña la disposición de la pantalla. Nosotros definimos un patrón y llenamos la pantalla con él, como si fuésemos a la ciudad con el bote de spray.

Lo siguiente que necesitas es alterar el patrón con el Editos de Patrón para que tenga el mismo tamaño que los sprites que vas a utilizar. Una vez tengas un patrón del mismo tamaño, debes limpiar el patrón a tinta 0 (el color de fondo). Selecciona la opción Brush y utilizando las teclas del cursor recorta las pistas de tu pantalla pintada.

Necesitas tomar nota de las coordenadas cada vez que cambies de dirección (crees un nodo). Cuando hayas terminado graba la pantalla sin activar la compresión.

Ahora tienes una lista con todas las coordenadas donde deben estar lo nodos. Las coordenadas utilizadas son las coordenadas del Amstrad y necesitan ser convertidas a las coordenadas del Sprites anter de introducirlas en el programade nodos.

El The Art studio te da las coordenadas para el centro de la brocha. Necesitas convertirlas a las de la esquina superior izquierda. Para hacerlo divide el tamaño de ka brocha entre 2tanto para x como para y. Resta el tamaño de x de la coordenada convertida del Art studio. Súmale el tamaño de y al la coordenada convertida del Art studio. El result de esto deben ser las coordenadaas que tienes que escribir en el programa de nodos.

Lo siento, nos quedamos sin espacio.

**SUPERVISOR DE SPRITES**INDICE DEL SUPERVISOR INDEX

COMANDO	PAGINA	DESCRIPCION
ANIMATE	29	Asocia un sprite a un número de secuencia.
ANIMOFF	29	Desactiva la animación de un sprite.
BULLET	33	Define número x de sprites que serán proyectiles.
CLEARREP	24	Limpia la tabla de Informes.
CLS	12	Limpia la pantalla.
COLLIDE	22	Define los atributos de sprite cuando colisiona con otro.
COLLTEST	23	Comprueba si un sprite concreto está colisionando.
COLOUR	12	Activa los colores de pantalla.
CSPRITE	27	Borra un sprite de la memoria.
DGET	9	Define un nuevo gráfico.
DRAW	11	Carga gráficos desde el Diseñador de Sprites.
ERASE	15	Limpia toda la información (311 sprites, incluyendo los gráficos.
EXPLODE	36	Provoca que explote sprite.
FRAME	31	Espera al retorno del refresco de pantalla.
HIT	25	Informa que sprite ha tocado un sprite concreto.
INKBLACK	12	Cambia todos los colores a negro.
KEB	17	Control por Teclado para sprite 1.
KEBDEF	19	Define las teclas para mover el sprite 1.
KEBSPEED	19	Define la velocidad del sprite 1.
MAZEOFF	43	Desactiva todos los nodos.
MAZEON	43	Activa todos los nodos.
MERGE	35	Inserta un nuevo gráfico en un sprite existente.
MISSDELAY	32	Define el intervalo entre proyectiles.
MISSDIST	32	Define el alcance de un proyectil.
MISSHIT	34	Informe colisión de proyectiles.
MISSILE	31	Define un proyectil para un gráfico.
MISSTYPE	30	Asocia un gráfico a un tipo de proyectil.

**SUPERVISOR DE SPRITES**

COMANDO	PAGINA	DESCRIPCION
MOVE	21	Mueve un sprite.
MOVEALL	21	Mueve todos los sprites.
MOVEHIT	25	Comprueba si colisiona el último sprite que se movió.
NEXTREP	24	Informa si el siguiente sprite colisionará.
NODO	40	Carga nodos en memoria.
NODOALTER	42	Cambia las direcciones de salida de un nodo.
NODOATTR	41	Establece los atributos para un nodo.
NODOOFF	42	Deshabilita un sprite para usar nodos.
NODOON	42	Habilita un sprite para usar nodos.
NODOSPEED	41	Establece la velocidad de los sprites en el laberinto.
NODOSPRITE	42	Determina si el sprite persigue/huye.
READKEB	20	Lee una dirección desde el teclado.
READSTIX	20	Lee una dirección desde el joystick.
REPOFF	23	Desactiva el informe de colisión.
REPON	22	Activa el informe de colisión.
REPORT	24	Comprueba por colisión todos los sprites y almacena el resultado en una lista.
RESEF	15	Limpia las variables de sprite.
SATTR	38	Establece los atributos de sonido de un sprite.
SCENERY	36	Coloca un gráfico en pantalla.
SCREEN	9	Cambia la forma de la pantalla.
SDIR	17	Define la velocidad de un sprite.
SDRAW	27	Informa del gráfico que está utilizando un sprite.
SECUENCE	28	Define una secuencia de animación.
SGET	14	Asocica un sprite a un gráfico.
SHOOT	34	Dispara un proyectil desde un sprite.
SMEM	27	Informa de la cantidad de memoria libre.
SOUND	38	Define un sonido.
SPEEK	37	Informa de la tinta en esas coordenadas.

SUPERVISOR DE SPRITES

COMANDO	PAGINA	DESCRIPCION
SPOKE	37	Establece una tinta en esas coordenadas.
SPUT	15	Coloca un sprite en pantalla.
SPUTALL	15	Quita todos los sprites de la pantalla.
STIX	17	Control por Joystick para el sprite. 0.
STIXSPEED	19	Define la velocidad para el sprite 0.
STUCK	37	Informa si un sprite no se puede mover.
SWINDOW	21	Define una ventanad parar un sprite.
SXPOS	27	Informa de la posición x del sprite.
SYPOS	27	Informa de la posición y del sprite.
WAIT	27	Pausa el programa.
WP	12	Cambia a Modo 2 y cambia los colores.
XDIR	27	Informa de la velocidad X del sprite.
XEDGE	16	Define los atributos para el sprite que
	toque el	margen de su ventana.
YDIR	27	Informa de la velocidad y del sprite.
YEDGE	16	Define los atributos para el sprite que
	toque el	margen de su ventana.

DISEÑADOR DE SPRITES  
GLEN COOK

**DISEÑADOR DE SPRITES**

El Diseñador de Sprites de tu disco de sistema te permite diseñar 64 sprites multicolores con sencillez. El programa sólo funciona en AMSTRAD 6128, ya que el programa utiliza los bancos de memoria extra.

Para utilizar el programa, simplemente inserta la cara correcta del disco en la unidad y escribe

RUN "DESIGN"

Se te pedirá entonces que introduzcas en la unidad un disco que contenga tus datos de gráficos. No te preocupes si no has grabado ningún dato todavía. Hemos incluido unos ficheros de ejemplo en el disco para tí. Presiona una tecla.

Después de poco tiempo se te presentará una lista de los ficheros de gráficos que estén almacenados en el disco.

- 1) DATA1
- 2) GAME
- 3) CREATE NEW FILE

Esta es la apariencia típica del menú. Este menú te dice que hay dos ficheros de gráficos en el disco que se llaman 'DATA1' y 'GAME'. La tercera opción te permite crear tu propio fichero de gráficos.

Selecciona la opción 1 presionando la tecla '1'

Tras una corta pausa, se mostrará el menú PRINCIPAL.

- THE OPTIONS MENU
- 1) EDIT A DRAWING
  - 2) VIEW THE DRAWINGS
  - 3) ANIMATE THE DRAWINGS
  - 4) DISC OPERATIONS
  - 5) GRAPHIC EDITOR
  - 6) SPRITE INFORMATION
  - 7) ALTER THE COLOURS
  - 8) CREATE SPRITE DATA
  - 9) END THE PROGRAM

PLEASE MAKE YOUR SELECTION

Veremos en breve cada una de las opciones. Para seleccionar una opción, presiona el número relevante. Debes usar los números de la parte superior del teclado ya que las teclas de función de la derecha han sido deshabilitadas.

Selecciona la opción 2.

Te dice el número de gráficos definidos y el rango de gráficos. El programa te preguntará si quieres ver los gráficos. Pulsa 'Y'.

Te mostrará los gráficos que tienes almacenados en memoria ahora. Puedes usar estos gráficos para experimentar con las distintas opciones del programa.

56

### **DISEÑADOR DE SPRITES**

Si en cualquier momento mientras estás en el programa cometes un error, p.e. seleccionas la opción incorrecta, puedes pulsar la tecla 'ESC' para volver al menú PRINCIPAL.

Ahora veremos cada opción individual.

#### **OPCIÓN 1 : EDIT THE DRAWINGS**

Tras seleccionar esta opción se mostrará

PLEASE ENTER DRAWING NUMBER

Puedes escribir un número entre 0 y 63. Después de hayas escrito este número pulsa 'ENTER'.

Si el número del gráfico que has escrito es un gráfico que ya estaba definido se mostrará una cuadrícula con el gráfico dentro, sin embargo, si es un nuevo número de gráfico no definido te pedirá

PLEASE ENTER DRAWING COORDINATES

Introduce las coordenadas x e y del gráfico. Puedes usar cualquier coordenada entre 1 y 32. El tamaño máximo de gráfico que puedes usar es 32 x 32.

PLEASE ENTER DRAWING DESCRIPTION

Puedes introducir una descripción de hasta 10 caracteres. Debes introducir al menos un carácter para la descripción o no podrás continuar con el programa.

Después de introducir la descripción se te presentará la cuadrícula de diseño. Esta es la pantalla con la que diseñarás todos tus gráficos.

A lo largo de la parte superior de la pantalla están los 16 colores que puedes utilizar para diseñar tus gráficos. Bajo estos colores están los números 0 a 9 y las letras A a F. Para seleccionar un color simplemente presiona el número o letra relevante.

En medio de la pantalla está la cuadrícula principal, ésta consta de una serie de casillas, el número de recuadros que hay depende del tamaño de coordenadas que hayas seleccionado. Los recuadros también varían de tamaño dependiendo de las dimensiones del gráfico que hayas escogido. La cuadrícula siempre estará a escala. P.e. una versión más grande del gráfico real.

El área a la derecha de la cuadrícula de diseño es donde se colocará el gráfico. Este gráfico es a tamaño real.

En la parte inferior de la pantalla está el área de información. Este área informa al usuario de una serie de cosas.

GRAPHIC NO. : 05                                    DESCRIPTION : BIRD LEFT  
DIMENSIONS : 16 x 20                            SPRITE SIZE : 00360 X=01 Y=01  
SPRITE SPACE : 12324                            SPACE LEFT : 25776

57

### **DISEÑADOR DE SPRITES**

Las DRAWING NUMBER y DESCRIPTIONS son auto explicativas. Las DIMENSIONS son las dimensiones x e y de la cuadrícula en la pantalla.

El SPRITE SIZE es la cantidad de memoria ocupa en memoria el gráfico que estás diseñando en pantalla. Es importante usar la menor memoria posible. Esto se puede conseguir asegurándose que no hay espacios a la izquierda alrededor de tu gráfico. Si hay espacios a la izquierda después de que hayas acabado tu gráfico, puedes cambiar las dimensiones como se describe más avanzado el manual.

El SPRITE SPACE es la cantidad de memoria que ocupan todos los gráficos que llevas diseñados.

EL SPACE LEFT es la cantidad de memoria que te queda para diseñar tus gráficos. Tienes inicialmente 38100 Bytes. Por favor, recuerda que también tienes que escribir en memoria un programa para controlar los sprites. Otra vez, intenta utilizar la menor memoria posible.

Las X=01, Y=01 son las coordenadas de los cursores dentro de la cuadrícula.

Puedes mover los cursores alrededor de la cuadrícula pulsando las teclas de las flechas. Según muevas los cursores alrededor de la cuadrícula notarás que las coordenadas X e Y cambian.

Para diseñar un gráfico tienes que mover los cursores a la casilla en que quieres un color y presionar la tecla de color relevante. Si te equivocas puedes limpiar pulsando '0' (el color de fondo) en el recuadro en el que te equivocaste.

Nota que cuando seleccionas un color, éste aparece en la parte superior derecha de la pantalla.

Esto muestra al usuario el color seleccionado. Ahora puedes rellenas casillas de la cuadrícula con ese color pulsando la BARRA ESPACIADORA. Para cambiar el color seleccionado, simplemente presiona el número o letra de otro color.

Probaremos ahora el menú de opciones. Para hacer esto pulsa la tecla 'ESC' una vez. La parte inferior de la pantalla se limpiará y aparecerán las opciones del menú.

f1 RENAME	f4 DELETE	f7 MOVEMENT
f2 COPY	f5 CLEAR	f8 COLOUR SEARCH
f3 COLOUR LATCH ON	f6 DIMENSIONS	f9 MAIN MENÚ

Para seleccionar una opción pulsa la tecla de función apropiada.

Para volver a la cuadrícula de diseño pulsa la tecla 'ESC' otra vez.

Esta es la única pantalla en la que si pulsando 'ESC' no vuelves al menú PRINCIPAL. Para volver al menú PRINCIPAL cuando estés en la pantalla de diseño debes pulsar ESC, para llegar al menú de arriba y seleccionar la telcla de función 9.

58

### DISEÑADOR DE SPRITES

#### f1 RENAME

Pulsar f1 permite al usuario cambiar la descripción del gráfico actual.

Después de presionar f1 la descripción se mostrará en pantalla. El usuario pue introducir una nueva descripción ahora. Si no deseas hacer esto, pulsa 'ESC' para volver al menú.

Después de introducir una nueva descripción presiona 'ENTER' para volver a la cuadrícula de diseño.

#### f2 COPY

Pulsar f2 permite al ususario copiar el gráfico actual a otra área gráfica. Esto es importante para animación ya que el usuario puede entonces alterar sólo una sección del gráfico en vez de tener que hacer el gráfico completo desde el principio.

Tras pulsar f2 se le pedirá al usuario que introduzca

COPY THIS GRAPHIC TO DRAWING NUMBER :

Si no quieres copiar a otro gráfico pulsa 'ESC'.

Ahora debes escribir el número de gráfico al que quieres copiar este gráfico y pulsar 'ENTER'. Si el número de gráfico que introduzcas ya tenía un gráfico dentro, se te mostrará el gráfico que sobrescribirás en la parte inferior derecha de la pantalla y te preguntará

YOU WILL OVERWRITE THE DRAWING ON THE BOTTOM RIGHT,  
CONTINUE Y/N

Si seleccionas N volverá a la pantalla de diseño.

Si seleccionas Y el gráfico se sobrescribirá con el gráfico actual.

POR FAVOR NOTA: CUANDO COPIAS UN GRÁFICO Y VUELVES A LA CUADRÍCULA DE DISEÑO PRINCIPAL, EL GRÁFICO ACTUAL ES AHORA EL GRÁFICO QUE ACABA DE COPIARSE. MIRA EL NÚMERO DE GRÁFICO. CUIDADO !!

#### f3 COLOUR LATCH

Pulsan f3 activa/desactiva el bloqueo de color.

La opción de bloqueo de color es una función que permite que la barra espaciadora recuerde el último color que se escribió. Cuando el bloqueo de color está activado cada vez que pulses la barra espaciadopa en la pantalla de diseño el color que está en la parte superior derecha de la pantalla se colocará en la posición del curso X, Y. Si desactivas el

bloqueo de color, la barra espaciadora no recordará el último color pulsado, actuando entonces como borrador. Limpiará el color en posición del cursor X, Y con el color de fondo.

Pulsa 'ESC' para volver a la cuadrícula de diseño.

59

### **DISEÑADOR DE SPRITES**

#### **f4 DELETE**

Pulsar f4 permite al usuario borrar el gráfico actual de la memoria.

Tras pulsar f4 se preguntará al usuario

ARE YOU SURE YOU WANT TO DELETE THIS DRAWING Y/N

Si seleccionas 'N' volverá a la cuadrícula de diseño con tu gráfico aún intacto.

Si seleccionas 'Y' el gráfico se borrará y volverás al menú PRINCIPAL.

#### **f5 CLEAR**

Pulsar f5 permite al usuario limpiar la cuadrícula actual, pero guarda la cuadrícula en memoria.

Tras pulsar f5 se preguntará al usuario

ARE YOU SURE YOU WANT TO CLEAR THIS DRAWING Y/N

Si pulsas 'N' volverá a la cuadrícula de diseño con tu gráfico aún intacto.

Si seleccionas 'Y' el gráfico se borrará y volverás a la cuadrícula de diseño.

POR FAVOR NOTA : No debes limpiar la cuadrícula y volver entonces al menú PRINCIPAL. Aunque no haya gráfico almacenado en la cuadrícula cuenta como un gráfico. Si quieres volver al menú PRINCIPAL sin almacenar un gráfico, debes usar la opción f4 y borrar la cuadrícula de la memoria.

#### **f6 DIMENSIONS**

Pulsar f6 permite al usuario cambiar las dimensiones de la cuadrícula actual. Puedes incrementar o decrecer el tamaño de la cuadrícula.

Tras pulsar f6 se preguntará al usuario

OLD DIMENSIONS : 16 x 20  
NEW DIMENSIONS : x

Ahora debes introducir las dimensiones que te gustaría que tuviera la cuadrícula.

Si está agrandando la cuadrícula la cuadrícula de diseño se redibujará con las dimensiones más grandes.

Si está reduciendo la cuadrícula, el área que se recortará se sombreadá. Esto permite al usuario ver si pierde parte del gráfico.

60

### DISEÑADOR DE SPRITES

Después de que el área a recortar esté sombreada se preguntará al usuario

WITH DRAWING DIMENSIONS OF 10 x 10 YOU WILL LOSE THE DRAWING  
DATA IN THE SHADED AREA. DO YOU WISH TO CONTINUE Y/N

Si seleccionas 'N' se redibujará la cuadrícula completa y volverá a la cuadrícula de diseño.

Si seleccionas 'Y' la cuadrícula se redibujará con las nuevas coordenadas y volverá a la pantalla de diseño.

PISTA UTIL: Cuando vas a reducir el tamaño de la cuadrícula, si mueves los cursores x, y a las casillas a las que quieres reducir el gráfico, anotando las coordenadas x, y en la parte inferior de la pantalla te ahorrará contar las casillas a mano.

### f7 MOVEMENT

Pulsar f7 permite al usuario manipular los gráficos. Puedes hacer una imagen espejo del gráfico o invertirlo.

Tras pulsar f7 se preguntará al usuario

PLEASE MAKE YOUR SELECTION : M - MIRROR IMAGE  
: I - INVERT THE GRAPHIC

Si no deseas hacer ninguna de estas opciones, pulsar 'ESC' volverá a la cuadrícula de diseño.

Pulsar 'M' produce una imagen espejo del gráfico. Esto es útil si se utiliza junto con el comando copy. Sólo necesitas diseñar un gráfico para una dirección. Si copias entonces el gráfico a un área libre y pulsas f7 el gráfico se producirá y moverá el sprite en la dirección opuesta.

Pulsar 'I' invierte el gráfico actual.

### f8 COLOUR SEARCH

Pulsar f8 permite al usuario reemplazar cualquier color de la cuadrícula con otro color.

Tras pulsar f8 se preguntará al usuario

PLACE X,Y CURSORS OVER THE COLOUR YOU WISH TO CHANGE AND  
PRESS THE COPY KEY.

Ahora puedes mover los cursores alrededor de la cuadrícula hasta que intercepten el color que quieres cambiar. Pulsa la tecla COPY para seleccionar el color deseado. Te pedirá entonces que confirmes el color.

PLEASE CONFIRM YOU WANT TO CHANGE COLOUR : 5 Y/N

61

### DISEÑADOR DE SPRITES

Si pulsas 'N' volverá a la cuadrícula sin que se altere nada.

Si seleccionas 'Y' te preguntará

CHANGE COLOUR : 5 TO NEW COLOUR :

Debes ahora introducir el número o letra del color al que quieres cambiar.

Tras introducir este número/letra el mensaje que aparecerá

YOU CANNOT ALTER ANY MISTAKES AFTERWARDS. ARE YOU SURE Y/N

Si pulsas 'N' volverás a la pantalla de diseño sin alterar nada.

Si pulsas 'Y' se reemplazará el color antiguo con el nuevo color y volverás a la cuadrícula de diseño.

ATENCION : Si cambias un color a un color que ya se está usando en la cuadrícula de diseño, e intentas volver a cambiarle, cambiarás todos los de ese color, no sólo los que cambiaste originalmente. Cuidado !!

### f9 MAIN MENÚ

Pulsar f9 permite al usuario volver al menú PRINCIPAL. El gráfico que está actualmente en la cuadrícula de diseño se grabará en memoria.

NOTAS de la opción 1:

Si estás almacenando un gran número de gráficos en memoria mantén un ojo en el espacio libre. Si en lugar de mostrar un número muestra 'OUT OF MEMORY' no tienes suficiente memoria libre para use la opción 8 del menú PRINCIPAL.

### MOVER LOS GRÁFICOS

El usuario puede mover los gráficos alrededor de la cuadrícula de diseño presionando la tecla SHIFT y la tecla de flecha adecuada.

Si mueves parte del gráfico más allá del margen de la cuadrícula se borrará y no podrás recuperarlo.

### OPCIÓN 2 : VIEWING THE DRAWINGS

Tras seleccionar esta opción se mostrará

GRAPHICS DEFINED : 20  
GRAPHIC RANGE : 00 - 31

DO YOU WISH TO SEE ALL THE GRAPHICS Y/N

**DISEÑADOR DE SPRITES**

Esto da al usuario cierta información sobre los gráficos que lleva almacenados en la memoria.

DRAWINGS DEFINED : Informa al usuario del número de gráficos que lleva diseñados.  
 DRAWING RANGE : Informa al usuario del los gráficos más bajo y más alto que lleva diseñados.

Si pulsa "Y" a la pregunta la pantalla mostrará hasta 16 gráficos. Si hay diseñados más de 16 gráficos se puede ver el resto de los gráficos presionando una tecla.

Si pulsa 'ESC' mientras se están mostrando los gráficos volverá al menú superior. Y así, pulsando 'ESC' en este menú volverá al menú PRINCIPAL.

Si pulsa 'N' a la pregunta se mostrará

PLEASE ENTER DRAWING RANGE: x

Debe introducirse ahora el número de gráfico más bajo que se desea ver, seguido del número de gráfico más alto. Tras introducir estos dos valores se mostrarán los gráficos.

**OPCIÓN 3 : ANIMATE THE DRAWINGS**

Tras seleccionar esta opción se mostrará una cuadrícula de 64 números del 0 a 63. Estos números representan los gráficos almacenados en memoria. Bajo el gráfico 0 hay una flecha.

Si algún gráfico ya está definido los números de ese gráfico serán de diferente color. Si los colores no se han cambiado con la opción 7, los números de gráficos no definidos serán azules y los números de gráficos definidos serán rojos.

En la parte inferior de la pantalla está la descripción del gráfico al que está apuntando la flecha. Si la flecha está apuntando a un gráfico no definido, la parte inferior de la pantalla mostrará 'NOT DEFINED'.

Si el gráfico se describe las dimensiones x, y del gráfico se mostrarán en la parte inferior izquierda de la pantalla.

NOTA IMPORTANTE: Todos los gráficos que van a animarse deben tener las mismas coordenadas x e y.

Puedes mover la flecha alrededor de la pantalla utilizando las teclas de flechas. Para seleccionar un gráfico para animar pulsa BARRA ESPACIADORA. El número de gráficos que has seleccionado se listará en la parte inferior de la pantalla. Si te equivocas pulsa la tecla 'DEL' y el número de gráfico se quitará.

Puedes seleccionar hasta 8 gráficos para animarlos.

63

### DISEÑADOR DE SPRITES

Tras seleccionar todos los gráficos que quieres animar pulsa la tecla ENTER.

Si te equivocas (los gráficos tienen diferentes dimensiones x, y) se mostrará la pantalla de error.

La pantalla de error lista todos los gráficos que has seleccionado y resalta cualquier gráfico con coordenadas x, y distintas. Tras pulsar una tecla te devolverá al menú de selección para volver a intentarlo.

Una vez que has seleccionado los gráficos y no hay errores se mostrará la pantalla de opciones de animación.

La pantalla será como esta :

SPEED OF SPRITE :

SELECT DIRECCIÓN (U,D,L,R) :

DELAY BETWEEN MOVES :

La opción speed of sprite determina cuantos píxeles moverá el sprite cada vez. Para conseguir el movimiento más suave debes seleccionar '1' y pulsar ENTER. Puedes usar cualquier número hasta un máximo de 9. Si quieres que sprite se anime Pero no se mueva debes seleccionar '0'.

La opción select dirección permite al usuario determinar la dirección en la que el sprite se moverá, Arriba, Abajo, Izquierda o Derecha. Si has seleccionado que el sprite sea estático seleccionando '0' en la última opción puedes escribir cualquier dirección, el sprite permanecerá estático. No es necesario pulsar ENTER en esta opción.

La opción delay between moves permite al usuario seleccionar la velocidad entre movimientos del sprite. Puedes utilizar cualquier número entre 0 y 99. La media está entre 30 - 40.

Si seleccionas '0' el sprite se moverá cada vez que pulses una tecla. Esta característica es útil para comprobar el movimiento a cámara lenta de un sprite para asegurarse de que la animación es correcta.

Tras pulsar ENTER en esta opción la pantalla se limpiará y verás el sprite animado. Para volver al menú PRINCIPAL pulsa la tecla 'ESC'.

### OPCIÓN 4 : DISC OPERATIONS

Es importante que sepas como se almacena la información en el disco. Esto puede prevenir cualquier contratiempo y posible angustias.

En el disco hay un fichero que se llama 'DISKDATA.SPR'. Este fichero mantiene información sobre los ficheros de sprite que están almacenados en el disco. Si decides borrar cualquier fichero del disco, por favor utiliza la función de borrado dentro de esta opción.

64

### DISEÑADOR DE SPRITES

Si borras algún fichero de gráficos del disco son usar esta opción, el menú de carga aún mostrará ese fichero. Esto se debe a que los nombre de los ficheros gráficos están almacenados en DISKDATA.SPR. Solamente la función de borrado dentro de esta opción borrarán la información relevante de este fichero.

Si borras el DISKDATA.SPR del disco no podrás recuperar cualquier fichero de datos de gráficos que tuvieras en el disco. CUIDADO !

Si deseas copiar algún fichero de gráficos de un disco a otro, debes cargar el fichero de datos de gráficos que deseas copiar del disco, insertar el disco en el que lo quieres copiar y grabar el fichero.

Un disco formateado en blanco puede almacenar 4 juegos de información de gráficos. Cada fichero de información de gráficos ocupa 36k. Procura estar seguro de que cuando vayas a grabar datos de gráficos a disco tengas suficiente espacio libre en el disco... Pero no pierdas el sueño con esto, el programa comprueba si hay suficiente espacio. Esto te hará ahorrar un poco de tiempo !

El programa hace su propia limpieza p.e. automáticamente borra cualquier copia de seguridad que cree.

Los gráficos se almacenan en el formato

nombrefichero.da1 : primer banco de memoria (17k)  
nombrefichero.da2 : segundo banco de memoria (17k)  
nombrefichero.da3 : info de color del gráfico (2k)

Si quieres almacenar datos de gráfico en un disco nuevo, asegúrate de que está formateado. No hay problema si no está el fichero DISKDATA.SPR en el disco, el programa creará uno.

Si te equivocas y la unidad ya ha comenzado a girar, NO intentes sacar el disco. Es un mal hábito, puedes perder todos los datos del disco. El fichero DISKDATA.SPR se borra antes de actualizarse.

Tras seleccionar la opción 4 se mostrará el menú

- 1) LOAD DRAWINGS
- 2) SAVE DRAWINGS
- 3) ERASE DRAWINGS
- 4) RETURN TO MAIN MENU

### LOAD THE DRAWINGS

Tras seleccionar esta opción el mensaje de aviso se mostrará

YOU WILL DESTROY INFORMATION IN MEMORY CONTINUE Y/N

Si seleccionas 'N' volverás al menú de arriba.

65

### DISEÑADOR DE SPRITES

Si seleccionas 'Y' se mostrará el mensaje que se muestra cuando ejecutas el programa la primera vez.

PLEASE INSERT DISC WITH GRAPHIC DATA INFORMATION AND PRESS ANY KEY

Debes introducir el disco que te gustaría cargar y pulsar una tecla.

Si has introducido un disco que no tiene un fichero DISKDATA.SPR en él, se darán las siguientes opciones

DATA NOT FOUND

- 1) CREATE NEW DATA DISC
- 2) TRY ANOTHER DISC

Si seleccionas la opción 2 se te pedirá que introduzcas otro disco y lo vuelvas a intentar.

Seleccionar la opción 1 creará un fichero DISKDATA.SPR en el disco.

Si el disco que has introducido no tiene ficheros de gráfico almacenados o sólo has creado un fichero DISKDATA nuevo, se mostrará el mensaje

NO GRAPHIC FILES STORED AS YET  
PRESS ANY KEY TO CONTINUE

Pulsa una tecla mostrará el menú PRINCIPAL.

Si el disco que introdujiste tenía algún fichero en él, éstos se mostrarán con un número a la izquierda del nombre del fichero. Si escribes ese número se cargará el fichero pertinente.

Tras los nombres de los ficheros hay una opción CREATE NEW DATA. Esta opción permite al usuario comenzar un nuevo fichero que se puede grabar con la opción de grabar que se explica después.

Si ha habido algún contratiempo y el fichero que intentas cargar falta o está corrupto el programa comprobará el disco completo para ver si falta algún otro fichero. Si se encuentra que faltan más ficheros, sus nombres se mostrarán y el DISKDATA.SPR se actualizará. El usuario podrá ahora elegir un fichero del catálogo actualizado.

### SAVE THE DRAWINGS

Si el fichero que vas a grabar no es un fichero recién diseñado p.e. ha sido cargado del disco, se mostrará el siguiente mensaje.

CURRENT FILENAME : DATA1

DO YOU WISH TO SAVE TO THIS FILE Y/N

66

**DISEÑADOR DE SPRITES**

Si seleccionas 'N' se mostrará

ENTER NEW FILENAME :

Puedes introducir un nombre de fichero de hasta 8 letras.

El fichero se grabará con su antiguo nombre si seleccionaste 'Y' o saved con el nuevo nombre si seleccionaste 'N'.

Si no hay suficiente espacio en disco para grabar el fichero, te pedirá que introduzcas otro disco y lo intentes de nuevo.

**ERASING THE DRAWINGS**

Tras seleccionar esta opción se mostrará

PLEASE INSERT DISC WITH DRAWING DATA FILE YOU WOULD LIKE  
TO ERASE AND PRESS ANY KEY

Tras introducir el disco con el fichero(s) que quieres borrar y pulsar una tecla, se mostrará una lista de los ficheros que hay en el disco. Será similar a ésta

ERASING THE DRAWINGS

- 1) DATA1
- 2) TEST
- 3) TEST
- 4) RETURN TO MENU

PLEASE MAKE YOUR SELECTION

Puedes seleccionar uno de los ficheros escribiendo el número pertinente. Si seleccionas la opción RETURN TO MENÚ volverás al menú PRINCIPAL del programa.

Tras seleccionar el fichero a borrar, la pantalla se limpiará y se pedirá confirmación para borrar el fichero.

PLEASE CONFIRM YOU WANT TO ERASE

DATA 1 Yes or No

Si decides no borrar ese fichero debes pulsar la tecla 'N' key, esto te devolverá a la pantalla del directorio de ficheros.

Si seleccionas 'Y' el fichero se borrará del disco y volverá al menú PRINCIPAL.

**RETURN TO MAIN MENU**

Esta opción simplemente te devuelve al menú PRINCIPAL.

67

## DISEÑADOR DE SPRITES

### OPCIÓN 5 GRAPHIC EDITOR

Tras seleccionar esta opción se mostrará una cuadrícula de 64 números de 0 a 63. Estos números representan los gráficos almacenados en memoria. Bajo el gráfico 0 hay una flecha.

Si algún gráfico ha sido definido el número de ese gráfico será de diferente color. Si los colores no han sido cambiados con la opción 7, los números de gráficos no definidos serán azules y los de gráficos definidos serán rojos.

En la parte inferior de la pantalla, en el centro, está la descripción del gráfico al que apunta la flecha. Si la flecha está apuntando a un gráfico no definido la pantalla mostrará 'NOT DEFINED'.

Bajo la descripción del gráfico hay un mensaje

Delete , Rename , Copy , ESC to end

Pulsando la tecla adecuada (D,R,C,ESC) se habilitará esa función para actuar en el número de gráfico al que apunta la flecha.

Puede mover la flecha por la pantalla utilizando las teclas de las flechas de la parte derecha del teclado.

Si seleccionas cualquiera de las opciones de arriba (excepto ESC) cuando la flecha esté apuntando a un gráfico no definido, el ordenador emitirá un pitido y nada ocurrirá.

Veamos ahora las funciones en detalle.

#### DELETE

Pulsar la tecla 'D' cuando la flecha esté apuntando a un gráfico definido, mostrará el mensaje

ARE YOU SURE YOU WANT TO DELETE Y/N

Si seleccionas 'N' se borrará este mensaje y el gráfico quedará intacto.

Si seleccionas 'Y' el gráfico se borrará de la memoria, el color del número al que apunta la flecha cambiará y la descripción ahora mostrará NOT DEFINED.

#### RENAME

Pulsar la tecla 'R' cuando la flecha está apuntando a un gráfico definido mostrará el mensaje

OLD : old description NEW:

68

**DISEÑADOR DE SPRITES**

Se pedirá ahora que se introduzca una nueva descripción para ese gráfico. Si decides que quieres mantener la misma descripción del gráfico puedes pulsar la tecla ESC key.

COPY

Pulsar la tecla 'C' cuando la flecha está apuntando a un gráfico definido mostrará el siguiente mensaje

COPY DRAWING : 03 TO DRAWING :

Se pedirá ahora que se introduzca un número de gráfico. El programa hará una copia del primer gráfico que seleccionaste y lo copiará al número de gráfico que acaasde introducir.

Si has introducido un número de gráfico que ya estaba definido, se mostrará un mensaje de error.

YOU CANNOT OVERWRITE EXISTING DRAWINGS

Esto es auto explicativo. Si deseas sobrescribir un gráfico, debes utilizar el comando copy de EDIT A GRAPHIC.

Si introdujiste un número de gráfico no definido, el gráfico se copiará.

ESC

Pulsar la tecla ESC devolverá al usuario al menú PRINCIPAL.

OPCIÓN 6 SPRITE INFORMATION

Tras seleccionar esta opción se mostrará la siguiente pantalla

GRAPHICS DEFINED : 30  
GRAPHIC RANGE : 00 - 43

DO YOU WANT INFORMATION ON ALL GRAPHICS Y/N

Esta pantalla da al usuario información sobre los gráficos que está en memoria en ese momento. See opción 2 VIEW THE GRAPHICS para esta información.

Si quieres información de todos los gráficos pulsa 'Y'.

Si pulsas 'N' se mostrará

ENTER GRAPHIC RANGE : -

Debes introducir el número del gráfico más bajo y del gráfico más alto de los que quieres información.

69

**DISEÑADOR DE SPRITES**

Se mostrará entonces

OUTPUT TO Screen OR Printer

Puedes seleccionar uno pulsando 'S' o 'P'.

PRINTER SELECTED

Si seleccionaste impresora la siguiente pregunta será

DO YOU REQUIRE CONTROL CODES SENDING TO PRINTER Y/N

Si quieres enviar códigos de control a tu impresora, pulsa la tecla 'Y'. Estos códigos de control pueden cambiar la impresora a un fuente distinta, etc.

Si quieres enviar códigos de control ahora se mostrará

PLEASE TYPE EACH INDIVIDUAL CONTROL CODE AND PRESS RETURN

CONTROL CODE :

En este punto debes introducir cada código de control y pulsar ENTER. Cuando hayas acabado de introducir los códigos de control debes pulsar la tecla 'ESC'. Según introduzcas los códigos de control estos se irán mostrando en la parte inferior de la pantalla. Si te equivocas presiona la tecla 'DEL'.

NOTA: Puedes introducir un máximo de 6 códigos de control. No pueden ser mayores de 63.

Ahora se mostrará

DO YOU WANT THE LINE FEED CONTROL CODE SENDING TO PRINTER Y/N

Debes contestar 'Y' a esta pregunta. Si encuentras que tu impresora está imprimiendo un línea en blanco entre cada parte de información en el futuro debes contestar 'N'.

PLEASE ENSURE YOUR PRINTER IS ONLINE AND READY TO PRINT  
PRESS ANY KEY WHEN YOU ARE READY

Se mostrará. Tras comprobar tu impresora, pulsa una tecla. Tu impresora debe sacar toda la información de los gráficos que seleccionaste.

SCREEN SELECTED

La información que se mostrará será como la de abajo

DRAW	DIMENSION	DESCRIPTION	MEMORY
00	32 x 32	MAN	1088
01	32 x 32	MAN	1088
		TOTAL MEMORY	2176

70

### DISEÑADOR DE SPRITES

Esta información se necesitará como referencia cuando se esté utilizando el programa Sprites. El valor total de memoria se necesitará para calcular que valor utilizar para el comando basic MEMORY.

#### OPCIÓN 7 : ALTERING THE COLOURS

Tras seleccionar esta opción se mostrara  
DO YOU WISH TO ALTER

- 0) MODE 0 COLOURS
- 1) MODE 1 COLOURS

MODE 0 COLOURS: Estos son los colores con los que diseñará tus gráficos.

MODE 1 COLOURS: Estos son los colores para los menús y mensajes.

Para seleccionar el tipo de colores que quieres cambiar pulsa '0' o '1'.

#### MODE 0 COLOURS

Verás en la pantalla tres bandas de colores. Las dos bandas de colores de la parte superior son los 27 colores disponibles en este ordenador. Sólo contarás 26 colores, el color de fondo está también allí pero no puedes verle.

La tercera banda de colores son los colores que están seleccionados actualmente para el modo 0. Bajo esta banda de colores están los números 1 - 9 y las letras A - F. Sólo puedes utilizar 16 colores en modo 0.

En la parte superior de la pantalla bajo el primer color hay una flecha. Puedes mover esta flecha de un color a otro color utilizando las teclas de las flechas.

Para cambiar un color, mueve la flecha al color que quieras seleccionar y pulsa el número/letra del color al que quieres cambiar. El nuevo colour se mostrará ahora en la banda de la parte inferior.

También puedes cambiar el color de fondo moviendo la flecha al color que te gustaría para el fondo y presionando la BARRA ESPACIADORA.

Cuando estés utilizando pantalla de diseño notarás que la información en la parte inferior de la pantalla está en modo 2. El color del texto es inicialmente rosa, el color 16. También puedes cambiar este color. No tienes que utilizar unos de los dieciseis colores que seleccionaste para el modo 0. Para cambiar este color, mueve las teclas de las flechas al color que quieras y pulsa la tecla COPY. La información en la parte inferior de la pantalla de selección (también en modo 2) cambiará. Esto te permitirá decidir si este color es legible sobre ese color de fondo.

**DISEÑADOR DE SPRITES**

Cuando hayas hecho tu elección pulsa la tecla 'ESC'.

NOTA: Los colores del fondo y del texto en modo 2 no pueden ser la misma, si intentas hacerlo el comando será ignorado.

Los colores que seleccionaste se grabarán junto con los gráficos, por tanto, no necesitarás cambiar los colores cuando cargues los gráficos de nuevo en el programa.

Los colores también se cargarán con los datos de gráficos cuando cargues los gráficos en el programa Sprites. Para acceder a los colores en el Sprites utiliza el comando COLOUR.

**MODE 1 COLOURS**

Cambiar estos colores cambiar todos los menús y mensajes que se muestran en modo 1. Esta información se graba cuando grabas tus datos de gráficos al disco.

Cuando seleccionas esta opción se muestra

```
BACKGROUND 0 CURRENT : 00 CHANGE TO :  
FOREGROUND 1 CURRENT : 16 CHANGE TO :  
FOREGROUND 2 CURRENT : 06 CHANGE TO :  
FOREGROUND 3 CURRENT : 02 CHANGE TO :
```

Puedes escribir los números de los colores que quieras (0 - 27). Cuando hayas introducido los cuatro números los colores cambiarán y el mensaje

ARE THESE COLOURS OK Y/N

se mostrará. Si pulsas 'N' los colores volverán a los que estabas usando antes de ejecutar el comando. Entonces se te preguntará otra vez. Si no deseas cambiar los colores pulsa 'ESC'.

Si pulsas 'Y' como respuesta a la pregunta de los colores de arriba se cambiarán permanentemente y se mostrará el menú PRINCIPAL.

**OPCIÓN 8 CREATE SPRITE DATA**

Este es el comando que convierte tus gráficos de forma que el Sprites los puede entener.

Cuando seleccionas esta opción se muestra el siguiente mensaje .

```
WARNING  
YOU WILL NOT BE ABLE TO RETURN TO  
THE SPRITES DESIGNER PROGRAM AFTER  
YOU HAVE CREATED THE SPRITE DATA
```

**DISEÑADOR DE SPRITES**

YOUR DRAWINGS MUST BE FULLY READY  
TO BE USED BY THE MAIN SPRITES  
PROGRAM  
ARE YOU SURE YOU WANT TO CONTINUE Y/N

Para que el programa pueda convertir los gráficos de manera que el Sprites pueda entenderlos, el programa tienen que borrar gran parte de sí mismo. Si respondes 'Y' a esta pregunta no podrás volver al Diseñador sin resetear el ordenador y ejecutarlo otra vez.

**IMPORTANTE : GRABA TUS GRÁFICOS EN DISCO ANTES DE UTILIZAR ESTA OPCIÓN**

Si no quieres continuar creando los datos de sprites, pulsa 'N' y volverás al menú PRINCIPAL.

Si quieres continuar pulsa 'Y'. Se te pedirá un nombre de fichero.

SAVE AS FILENAME :

Puedes usar cualquier nombre de fichero de hasta ocho letras. Tas introducir el nombre del fichero pulsa ENTER. El programa te pedirá

PLEASE INSERT THE DISC YOU WANT THE  
DRAWINGS SAVING TO AND PRESS ANY KEY

Ahora debes introducir el disco en el que quieres almacenar los gráficos. No utilices el disco de sistema.

Tras pulsar una tecla la pantalla cambiará los colores unas cuantas veces. Esto es normal, el programa utiliza la pantalla como buffer para los gráficos. Finalmente la pantalla se limpiará y mostrará.

THE FILE HAS BEEN SAVED  
YOU MAY NOW RESET THE COMPUTER BY  
PRESSING SHIFT-CONTROL-ESC

El programa ha finalizado.

El fichero de gráficos que se ha creado tiene el formato

nombredeusuario.DKW

**OPCIÓN 9 : END THE PROGRAMA**

Esta opción es para terminar el programa y volver al BASIC.

Tras seleccionar esta opción se nmostrará el siguiente mensaje.

**DISEÑADOR DE SPRITES**

PROGRAM TERMINATION  
YOU WILL LOSE ALL THE INFORMATION IN  
MEMORY IF YOU END THE PROGRAM I HOPE  
YOU HAVE SAVED THE GRAPHICS TO DISK.  
END THE PROGRAM Y/N

Pulsar 'Y' resetea el ordenador.

Pulsar 'N' te devolverá al menú PRINCIPAL con los datos aún intactos.

**FELIZ DISEÑO**

(Esta página faltaba en el PDFs que envié originalmente)

