

# MOVIMIENTO Y ANIMACION

¿Quieres darle vida a tus juegos de programación? Empieza entonces con estos sencillos caracteres gráficos, que puedes generar a partir de la memoria gráfica de tu ordenador.

Jugar con los juegos que se venden para tu ordenador es divertido sólo hasta cierto punto. Llega un momento en que la mayor parte de la gente siente la necesidad de dar rienda suelta a su imaginación y crear programas de juegos propios.

La programación de juegos no es fácil; tienes que empezar con cosas muy sencillas e ir aumentando poco a poco en complejidad. Pero eso te ayudará a pensar con lógica y aumentará tu habilidad como programador. Y también te divertirás más.

Lo primero que tienes que aprender en la programación de juegos, aparte de los trucos del BASIC, es la técnica de la animación.

Para crear la ilusión de movimiento, el programador de ordenadores utiliza en gran parte la misma técnica que el caricaturista que anima una película de dibujos. Lo que hace es crear dos (o más) imágenes y alternarlas rápidamente (idealmente, unas 24 veces por segundo).

Pero existe una diferencia importante. En la animación de dibujos, el dibujante cuenta con el proyector de películas, que le permite olvidarse de una imagen cuando ya no la necesita. En la animación por medio de un ordenador no ocurre esto. Como no has algo para evitarlo, cualquier seg-

mento de imagen que «proyectes» sobre un área dada de la pantalla permanecerá allí indefinidamente.

Una forma de olvidarse de la imagen que ya no se necesita es, simplemente, imprimir algo sobre ella. El ordenador no puede situar dos imágenes al mismo tiempo en la misma posición de la pantalla.

Así si, por ejemplo, la línea 10 de un programa cualquiera dice al ordenador que imprima una A en una determinada posición, cualquier otra línea de programa que imprima (por ejemplo) una B en la misma posición se desembarazará de la A.

Los programas que siguen contienen varios ejemplos de esta clase de sustitución.

¿Pero qué ocurre si no tienes nada

■	LOS PRINCIPIOS DE LA ANIMACION
■	MOVIMIENTO DE GRAFICOS
■	COMO UTILIZAR LOS GRAFICOS INCORPORADOS





# PROGRAMACION DE JUEGOS

que imprimir encima del carácter no deseado? Acuérdate en ese caso de incluir en alguna línea posterior una instrucción que imprima un espacio en blanco en la correspondiente posición de la pantalla.

Si te olvidas de este detalle, tu pantalla pronto se verá abarrotada con trozos no deseados de brazos, piernas y cuerpos.

La manera de obtener los caracteres gráficos en la pantalla difiere mucho de un ordenador a otro. Hay diferencias entre los caracteres gráficos incorporados en ROM y lo mismo ocurre con la forma en que se imprimen (con PRINT) en pantalla. Por último también hay diferencias en el modo de hacer que se muevan.

Esta versión de un ciempiés para MSX utiliza exactamente los mismos signos de máquina de escribir que otras máquinas, pero el método de generarlos sobre la pantalla es diferente. Intenta introducir el siguiente programa:

```
)))  )))  )))  )))  )))
ooo <  ooo <  ooo <  ooo <
)))  )))  )))  )))  )))
```

```
10 CLS
20 PRINT"))))"
30 PRINT"ooo<"
40 PRINT"))))"
50 LOCATE 0,0
60 PRINT"(((("
70 PRINT"ooo<"
80 PRINT"(((("
90 GOTO 10
```

Cuando lo ejecutes (RUN) observarás una imagen rápidamente cambiando. Se debe a los conjuntos separados de símbolos de las sentencias PRINT que se superponen unos a otros. Al mismo tiempo, la sentencia GOTO de la última línea crea un bucle continuo: le dice al ordenador que vuelva a empezar.

Sin las líneas 10 y 50 el programa no podría funcionar adecuadamente. En ambas líneas se hace uso de instrucciones de movimiento de cursor y de borrado de pantalla, que combinadas con la instrucción PRINT permiten llevar a cabo la animación.

¿Cómo funcionan? La instrucción LOCATE 0,0 de la línea 50 coloca el cursor en la esquina superior izquierda de la pantalla (línea 0, columna 0).

Esto quiere decir que toda actividad de impresión posterior comienza en esta posición, de modo que todos los caracteres que se imprimen, desde la línea 60 en adelante, lo hacen encima de los que ya estaban allí.

Por su parte la instrucción CLS de la línea 10 hace algo más. Después de hacer volver al cursor a la parte superior izquierda de la pantalla, borra todo su contenido, dejándola preparada para que aparezca la siguiente imagen.

## MAS DESPACIO

El movimiento del insecto es hasta ahora más bien rápido y tal vez haya que retardarlo. La manera más fácil de hacer esto es utilizar un bucle FOR...NEXT. Introduce pues la línea:

```
45 FOR T=1 TO 50:NEXT
```

Cuando pulses RETURN y ejecutes (RUN) el programa, el movimiento resultará mucho más pausado. El bucle FOR...NEXT actúa como un contador, en este caso contando hasta 100, antes de que el programa vaya a la línea 50.

Puedes cambiar la duración de la





# PROGRAMACION DE JUEGOS

pausa sin más que sustituir «100» por cualquier otro número de tu elección: cuanto más grande el número, mayor será el retardo.

Intenta también cambiar la posición del bucle de retardo FOR...NEXT a la línea 15. Dependiendo del retardo que escojas, habrá una apreciable pausa —una pantalla limpia— cuando se ha borrado una imagen pero todavía no ha sido sustituida por la siguiente. Por esta razón es mejor utilizar LOCATE 0,0 que CLS, dentro de un programa de este tipo.

Aunque la imagen ya está ahí, todavía resulta algo espasmódica, debido a que el bucle de retardo sólo actúa sobre la primera imagen. Se puede obtener un movimiento mucho más adecuado introduciendo en el programa otro bucle FOR...NEXT que actúe sobre la segunda imagen. Inserta pues:

```
85 FOR I=1 TO 50:NEXT
```

Este retardo es más corto, a fin de crear un movimiento «de pierna» ligeramente irregular, pero naturalmente, puedes cambiarlo si quieres.

## CARACTERES EN MOVIMIENTO

El siguiente paso es alterar el programa de forma que el «cuerpo» del insecto parezca cruzar la pantalla. Para ello el BASIC MSX nos ofrece dos instrucciones que vamos a utilizar. La primera de ellas, que puede ser adecuada en aplicaciones sencillas, es la instrucción TAB. TAB siempre va seguido de un número entre paréntesis, por ejemplo TAB (15), que hace que el cursor se coloque en la columna 15 de la pantalla, o por una variable también entre paréntesis.

Además TAB siempre forma parte de la sentencia PRINT a la que se aplica. En este caso se hace uso de una va-

riable, la variable P de la línea 10, para hacer que la posición de TAB se mueva por la pantalla. Esta variable cambia su valor al ir incluida dentro de un bucle FOR...NEXT.

```
10 FOR P=0 TO 32
20 CLS
25 PRINT TAB(P)"))))"
30 PRINT TAB(P)"ooo<"
40 PRINT TAB(P)"))))"
50 FOR I=1 TO 45:NEXT
55 LOCATE 0,0
57 PRINT TAB(P)"(((("
60 PRINT TAB(P)"ooo<"
70 PRINT TAB(P)"(((("
80 FOR I=1 TO 50:NEXT
90 NEXT P
```

Lo que has hecho es suprimir la sentencia GOTO original de la línea 90. Ahora ha sido remplazada por un bucle FOR...NEXT, que incrementa en 1 la variable P cada vez que se repite el programa. Como P forma parte de la sentencia TAB, el insecto se mueve sobre la pantalla, a razón de un paso por cada ciclo del programa.

Al ejecutar (RUN) el programa, verás que el insecto se mueve sobre la pantalla y se para al llegar al lado de-

