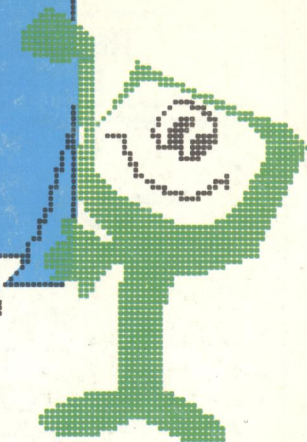


VIDEO BASIC

20 LECCIONES DE BASIC
PARA APRENDER CON EL C-64



INGELEK



JACKSON

La tableta gráfica

El lápiz óptico

CAD/CAM

Alta resolución

*Como trazar
un punto y una recta*

*Animación y movimiento
de un carácter*

Videojuego N.º 12

12

COMMODORE

C-64

RESULTADO DEL SORTEO DE PROMOCION ENTRE LOS LECTORES DE VIDEOBASIC

El día 31 de julio, tal como establecen las bases del concurso VIDEOBASIC ante D. Juan M. BOLAS ALFONSO, notario del Ilustre Colegio de Notarios de Madrid, se ha efectuado el sorteo de 40 IMPRESORAS SEIKOSHA, cuyo resultado ha sido el siguiente:

SEIKOSHA GP 500 y GP 50* (LECTORES)

- ANTONIO ESPAÑA CERRATO. MADRID.
- VICENTE RODRIGUEZ CASTRO. PONFERRADA (LEON)
- JUAN JOSE FERNANDEZ COLLADO. CADIZ.
- JOSE JAVIER GOMEZ FERNANDEZ. MADRID.
- JAIME MAYORAL FERRER. SAN FELIU DE LLOBREGAT (BARCELONA).
- JOSE MARIA FERNANDEZ BENEYTO. BARCELONA.
- JOSEP FARREGAS SALA. VALLIRAMA (BARCELONA).
- FRANCISCO JOSE TAMARGO GARCIA. SALINAS-CASTRILLON (ASTURIAS).
- FELIPE SOTO GONZALEZ. SAN SEBASTIAN (GUIPUZCOA).
- EDUARDO MORENO MACIAS. SEVILLA.
- SALVADOR BLANQUER AZWAR. VALENCIA.
- RICARDO IBAÑEZ PUYAL. HOSPITALET (BARCELONA).
- SILVESTRE MATEOS REDONDO. SALAMANCA.
- SOFIA MORALES GARRIDO. MADRID.
- MANUEL GUILLEM TECLA. VALENCIA.
- GEMA COTEREAU BADIOLA. MADRID.
- JAIME SANCHEZ VAZQUEZ. SEVILLA.
- MAN CHUNG LAM. FUENGIROLA (MALAGA).
- MANUEL MANZANO MARTINEZ. VALENCIA.
- CESAR RODA MARTINEZ. REQUENA (VALENCIA).

* Los agraciados deberán enviarnos fotocopia de la tarjeta de garantía de su ordenador para remitirles la impresora adecuada.

SEIKOSHA GP 500 (SUSCRIPTORES)

- JOSE A. CAMARGO MORALES. MADRID.
- ANGEL MARTIN AGUADERO. BEJAR (SALAMANCA).
- ISABEL MEDIAVILLA SANCHEZ. SANTANDER.
- MANUEL CARMONA LUQUE. UTRERA (SEVILLA).
- REGINA RAMOS GOMEZ. BARCELONA.
- MIGUEL ANGEL ALONSO JIMENEZ. MADRID.
- ROBERTO PECINO CANO. LA LINEA (CADIZ).
- FERNANDO ANGOSO SANCHEZ. SALAMANCA.
- MARIANO RUMBERO SANCHEZ. VALENCIA.
- ANTONIO CRESPO INSAUSTI. LAS ARENAS (VIZCAYA).

SEIKOSHA GP 50 (SUSCRIPTORES)

- TOMAS ROMERA SANZ. MADRID.
- FRANCISCO GUTIERREZ FERNANDEZ. MADRID.
- EDY SERI. MADRID.
- JUAN PABLO VERDU MIRA. SAN VICENTE RASPEIG (ALICANTE).
- ALBERTO SOLE BAQUES. SITGES (BARCELONA).
- FRANCISCO VAZQUEZ LAZARO. BILBAO (VIZCAYA).
- IGNACIO JAVIER PONTE MARTIN. AGUADULCE (ALMERIA).
- JORGE GRAU CRESPO. CASTELLO DE RUGAT (VALENCIA).
- JOSE MARIA NEBOT GOMEZ DE SALAZAR. MADRID.
- JOSE MANUEL AVILA FERNANDEZ. LEGANES (MADRID).

VIDEO BASIC

Una publicación de INGELEK JACKSON

Director editor por INGELEK:

Antonio M. Ferrer

Director editor por JACKSON HISPANIA:

Lorenzo Bertagnolio

Director de producción:

Vicente Robles

Autor: Softidea

Redacción software italiano:

Francesco Franceschini,

Stefano Cremonesi

Redacción software castellano:

Fernando López, Antonio Carvajal,

Alberto Caffarato, Pilar Manzanera

Diseño gráfico:

Studio Nuovaidea

Ilustraciones:

Cinzia Ferrari, Silvano Scolari,

Equipo Galata

Ediciones INGELEK, S. A.

Dirección, redacción y administración,

números atrasados y suscripciones:

Avda. Alfonso XIII, 141

28016 Madrid. Tel. 2505820

Fotocomposición: Espacio y Punto, S. A.

Imprime: Gráficas Reunidas, S. A.

Reservados todos los derechos de reproducción y

publicación de diseño, fotografía y textos.

© Grupo Editorial Jackson 1985.

© Ediciones Ingelek 1985.

ISBN del tomo 3: 84-85831-22-5

ISBN del fascículo: 84-85831-14-4

ISBN de la obra completa: 84-85831-13-6

Depósito Legal: M-15075-1985

Plan general de la obra:

20 fascículos y 20 casetes, de aparición quincenal,

coleccionables en 5 estuches.

Distribución en España:

COEDIS, S. A.

Valencia, 245. 08007 Barcelona.

INGELEK JACKSON garantiza la publicación de todos los fascículos y casetes que componen esta obra y el

suministro de cualquier número atrasado o estuche mientras dure la publicación y hasta un año después de terminada.

El editor se reserva el derecho de modificar el precio de venta del fascículo, en el transcurso de la obra, si las circunstancias del mercado así lo exigen.

Impreso en España.

INGELEK



JACKSON

SUMARIO

HARDWARE 2

La tableta gráfica. El lápiz óptico. CAD/CAM.

EL LENGUAJE 12

Alta resolución. Gráficos. Cómo entrar en alta resolución. El punto. La recta.

LA PROGRAMACION 26

Animación. Movimiento de un carácter.

VIDEOEJERCICIOS 32

Introducción

La elaboración de imágenes es uno de los aspectos más espectaculares y atractivos de un ordenador. Todos conocemos las imágenes que transmite la televisión de rótulos y dibujos realizadas con un ordenador gráfico, o los proyectos de automóviles que se obtienen con sistemas electrónicos específicos. Se trata de procesos que necesitan mucha memoria y una elevadísima velocidad de elaboración, y que en consecuencia prácticamente no existían hace bien pocos años. Hoy ya no es así. Lo demuestra el hecho de que la lucha comercial entre las casas constructoras se extienda también al campo de la resolución gráfica, con pixels direccionables por software que resulta más o menos capaz de producir, gestionar, animar y mover imágenes.

Nos introduciremos, pues, en este nuevo ambiente, examinando algunos periféricos específicos: el lápiz óptico y la tableta gráfica.

La tableta gráfica

Desde la Antigüedad el dibujo se convirtió en uno de los medios de expresión y comunicación más útiles, significativos e inmediatos.

Estaba bien claro que el ordenador no podía ignorar los aspectos gráficos.

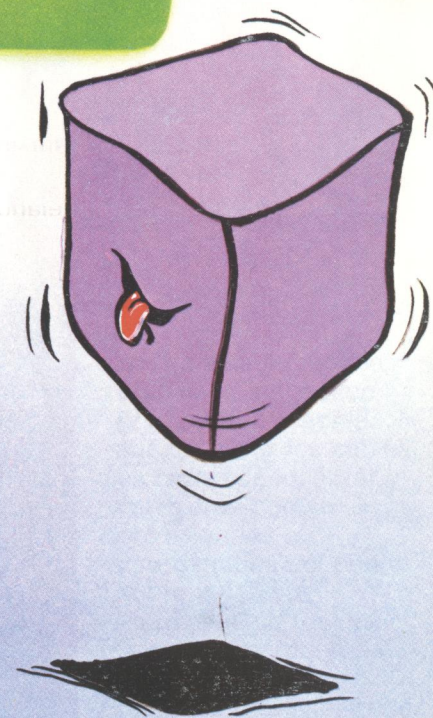
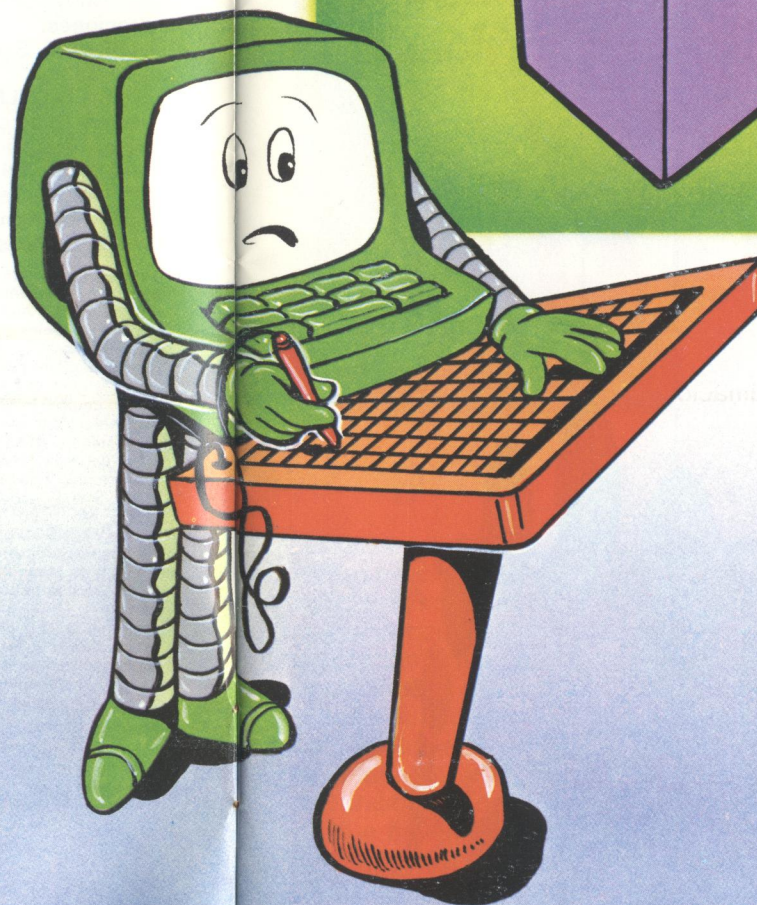
Después de unos primeros pasos, tímidos e inciertos, durante los años pasados, los posibles usos de los ordenadores dentro de este campo van haciéndose cada vez más numerosos e importantes, de tal forma que ya no es difícil pronosticar un futuro ya no demasiado lejano, con animaciones e imágenes completamente gestionadas y hasta realizadas con la ayuda de un ordenador.

Los primeros ejemplos podemos encontrarlos ya en los cines.

Y se habla además (y no es ciencia-ficción) de la posibilidad de hacer nuevas películas con intérpretes ya desaparecidos y simulados con un ordenador.

Como es natural, en estos casos se habla de ordenadores de millones de dólares;

pero de cualquier manera, también los ordenadores personales poseen notables capacidades gráficas que les hacen útiles —por no decir indispensables— en un amplio campo de



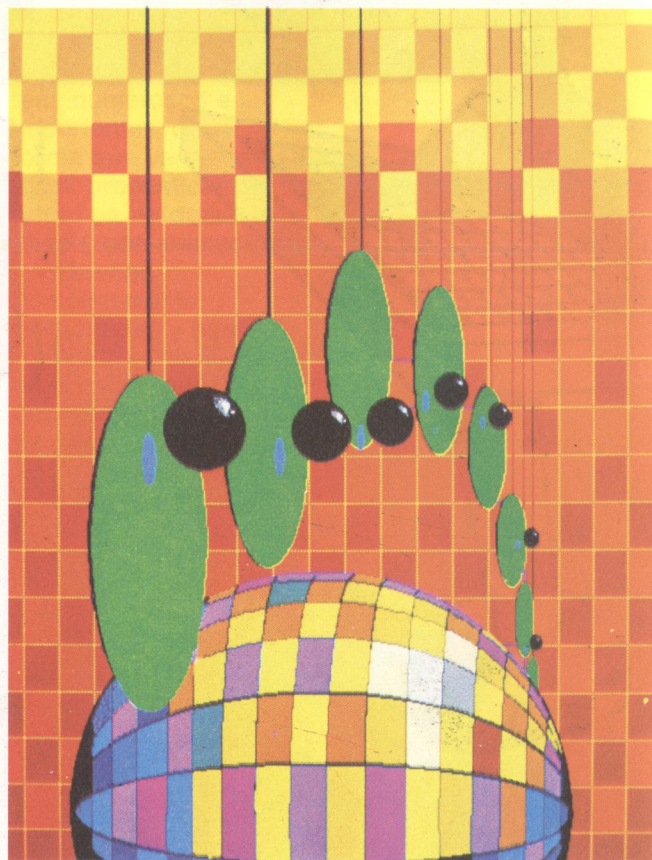
aplicaciones: proyectos en general, urbanismo, medicina, negocios, didáctica o juegos. Aún tratándose de una máquina esencialmente numérica, el ordenador es capaz de crear dibujos y gráficos. Para que las imágenes

puedan ser tratadas son necesarias dos cosas: una codificación de dichas imágenes bajo forma de números y unos periféricos capaces de comunicarle al ordenador las imágenes.

La tableta gráfica es un periférico que permite realizar dibujos de una manera sencilla e

inmediata. Se trata, fundamentalmente, de una pequeña tableta, (de tamaños muy variables de uno a otro modelo) y de una especie de pluma, que se puede colocar libremente (y con notable exactitud) en un punto cualquiera de la tabla.

Una vez seleccionado el



punto que se desea incluir en el dibujo, es suficiente con presionar la pluma sobre la tableta para que las coordenadas del punto le sean inmediatamente comunicadas al ordenador (naturalmente, después de que hayan sido convertidas a forma digital por el apropiado circuito de interface). El conjunto de coordenadas que se van indicando

sucesivamente con el movimiento de la pluma compondrán el dibujo, que va apareciendo en pantalla igual que si se pasara un lápiz sobre una hoja de papel. El principio de funcionamiento que permite realizar estas operaciones suele estar basado sobre una red muy tupida de cables cruzados, situada justo debajo de la superficie de trabajo. Cuando la pluma se presiona

contra esta superficie, un campo magnético producido en rápida sucesión por los cables, es capaz de localizar y establecer con precisión las coordenadas del punto sobre la tableta. La resolución (es decir, la capacidad de distinguir el más pequeño movimiento de la pluma) depende del número de hilos que formen la red; en los modelos para usos de

entretenimiento ésta es lógicamente mucho más limitada que en aquéllos de uso profesional, aunque llegue, hasta en los modelos más económicos, a las 10 líneas por milímetro (lo que significa que la tabla es capaz de notar variaciones de la posición de la pluma hasta valores de 1/10 de milímetro).

De cualquier manera, la tableta gráfica, como cualquier otro periférico, sirve únicamente para comunicarle informaciones al ordenador, o mejor dicho, al programa, que es quien decide cómo emplear estas informaciones que le son comunicadas. Por lo tanto, junto a la tableta gráfica debe usarse un programa que convierta las instrucciones impartidas mediante la pluma en instrucciones exactas, realizables por el ordenador. Para los ordenadores personales más comunes estos programas se proporcionan junto con la tableta gráfica, y suelen constituirse normalmente como parte integrante de todo el equipo necesario para dibujar con el ordenador.

El lápiz óptico

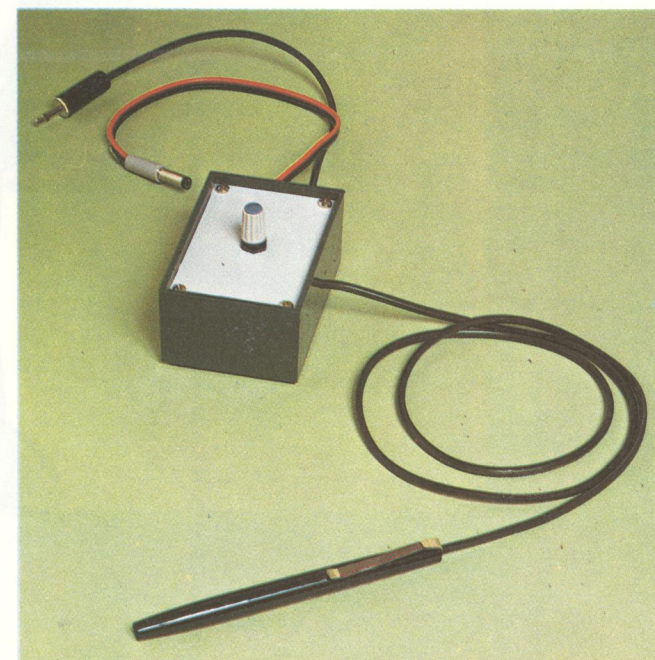
Otro dispositivo que permite dibujar de una forma más económica y con menor ocupación de espacio que con la tableta gráfica es el lápiz óptico (o lápiz

luminoso). Se trata de un sensor, de forma muy similar a la de un lápiz, capaz de interpretar las variaciones de luminosidad a las que se ve sometido. Su empleo es muy semejante al de la pluma de la tableta gráfica, con la única diferencia de que la superficie de trabajo está constituida —en lugar de por un soporte exterior— por la misma pantalla del ordenador. Es suficiente con apoyar el lápiz en cualquier punto de la pantalla y el ordenador recibirá (convertidas como de costumbre a forma digital) las coordenadas del punto seleccionado. Para entender la manera en que esto puede ocurrir, es necesario remontarse al principio de funcionamiento de las unidades de pantalla. ¿Lo recuerdas? Un haz de electrones ilumina continuamente una capa de substancias sensibles a la luz (compuesta de fósforos), situada sobre la cara interior de la pantalla, provocando una mayor o menor luminiscencia. El lápiz

luminoso es un dispositivo capaz de distinguir los puntos iluminados en pantalla. Dado que la posición del haz electrónico se puede conocer en cada instante, también la posición de la pluma se puede determinar con

facilidad por parte de los circuitos dedicados al control de la pantalla. De cualquier manera, la precisión del lápiz óptico es notablemente menor que la de la tableta gráfica, no pudiendo superar en ningún caso la definición propia de cada pantalla. Además obliga a trabajar con el brazo levantado y mirando fijamente a la pantalla. Sin embargo, no todos los lápices ópticos son iguales. Diferencias más o

menos pequeñas, en el tamaño y luminosidad que se requieran de la pantalla, hacen a algunos lápices mejores que otros. Es natural que los mejores sean aquéllos de un tamaño más pequeño, y que necesiten de una menor luminosidad en pantalla, dado que así cansarán menos tanto al brazo como a la vista. Una vez más son imprescindibles para el funcionamiento de este dispositivo los circuitos electrónicos de



interface y el programa de uso que se emplee, para hacerle comprensibles al ordenador las informaciones que va recibiendo.

Sin embargo, para trabajos de una cierta precisión y complejidad

el lápiz óptico no puede ser comparado con la tableta gráfica. Este no es más que un método cómodo y rápido para indicar los puntos deseados sobre la pantalla, dejándole a la tableta gráfica todos aquellos trabajos donde la complejidad y precisión necesarias sean claramente superiores.



CAD/CAM

Ya hemos visto cómo la introducción de los ordenadores, y en particular la de los personales, permite además de los usos tradicionales, también unas nuevas posibilidades de dibujo. Y esta posibilidad no queda limitada a las actividades cotidianas del trabajo de oficina o a las de la mera diversión; entre sus múltiples aplicaciones, poseen un particular relieve los recientes usos de los gráficos

computerizados en el campo de los proyectos industriales.

Estos usos, nacidos gracias a los estudios realizados en las grandes empresas del sector del automóvil, han conseguido que el ordenador colabore y se aproxime cada vez en mayor medida a los dibujantes y proyectistas, permitiendo superar algunos problemas que condicionaban de forma muy rígida desde hacía largo tiempo las fases de estudio y proyecto de cualquier producto industrial.

CAD y CAM (Computer Aided Design y Computer Aided Manufacturing, respectivamente, Diseño asistido por ordenador y Producción asistida por ordenador) son las abreviaturas conocidas universalmente que indican estos usos específicos del ordenador.

Las aplicaciones de dichas tecnologías (puesto que como veremos se trata de auténticas tecnologías) son ya numerosísimas y extremadamente difundidas: el proyecto y la elaboración ha significado, y sigue

significando, una notable reducción y optimización de esfuerzos y de costes en distintos productos industriales, permitiendo así la creación de mejores productos a precios más reducidos. El ejemplo más clásico y el que mejor ilustra las posibilidades actualmente alcanzables con el CAD/CAM es el de la industria de los circuitos integrados (o chips). Ya sabes que un circuito integrado no es más que un microscópico circuito electrónico situado sobre una placa de silicio. Parecería casi un asunto fácil poder reducir directamente el circuito proyectado (realizado necesariamente a una escala humana, es decir, con tamaños que permiten trabajar viendo con los ojos)

hasta el tamaño del circuito terminado, es decir, el de un microscópico chip. Pero esta es una idea muy equivocada. Desde la fase de proyecto es necesario tener en cuenta que es preciso optimizar y minimizar tanto los espacios disponibles entre los elementos electrónicos que constituyen el circuito, como los elementos mismos, para que al final quede ocupada la menor superficie posible. Por fuerza, este trabajo necesita de millares de cálculos, para poder

examinar todas las posibles disposiciones y combinaciones, cálculos que resultan completamente imposibles de realizar sin la ayuda de un ordenador adecuadamente preparado. En la pantalla del proyectista, el ordenador —gracias al programa CAD— va visualizando las posibles disposiciones de los componentes electrónicos, limitando, o hasta evitando la larga y aburrida fase del dibujo manual de varios circuitos.

Una vez resuelto el proyecto también la producción de los chips necesita la adopción de especiales cuidados y el respeto de determinadas tolerancias por parte de las máquinas empleadas en el trabajo, aspectos que únicamente un ordenador programado con sistemas CAM es capaz de coordinar con fiabilidad y seguridad. Al final del proceso productivo, la tecnología CAD/CAM es pues capaz de ofrecer circuitos integrados con calidades y prestaciones claramente

superiores a los conseguibles por otros procedimientos, y sobre todo con un precio claramente más bajo. Lo que hemos expuesto no es más que un ejemplo: en lugar de chips podríamos haber hablado de bastidores de automóviles o de componentes de motores a reacción, y el texto habría sido prácticamente el mismo. Es evidente que el uso de estas técnicas para el proyecto y la producción requiere ordenadores de gran tamaño y elevada potencia de cálculo (y en consecuencia, únicamente al alcance de las grandes empresas). Sin embargo, ya desde ahora están empezando a aparecer los primeros programas para ordenadores personales; en pocos años también las pequeñas industrias y los estudios profesionales deberían pues tener a su disposición estos útiles y potentes medios para proyectar y producir, permitiendo así ampliar aún más el ya extenso catálogo de productos fabricados actualmente con el CAD/CAM.

Alta resolución



En el interior de tu C64 existe un circuito integrado llamado VIC (Video Interface Controller, es decir, Controlador del Interface de Vídeo), que tiene entre sus cometidos el de acceder a determinadas informaciones contenidas en ciertas partes de la memoria y

usarlas para crear señales, que adecuadamente codificadas, puedan alcanzar la pantalla de TV o el monitor, generando las imágenes.

El VIC es un dispositivo programable, es decir, capaz de funcionar en distintos modos gráficos, según las

órdenes que lleguen a su entrada.

El primero de estos modos es el llamado "modo carácter", aquél que está a tu disposición en el momento del encendido y que es también el más sencillo de usar.

En este "modo", VIC está programado para manejar los datos de la memoria, obteniendo 25 líneas de 40 caracteres cada una. Naturalmente, estos caracteres pueden ser tanto los generados por la ROM (2 juegos diferentes), existentes en el teclado, como aquéllos definidos en la RAM por el programador.

Operar mediante caracteres involucra un área de pantalla de 64 puntos para cada uno de ellos.

Esto no permite componer dibujos o gráficos de una cierta dificultad para los que el detalle sea muy importante.

Para superar esta limitación se puede emplear entonces otro modo gráfico, llamado "gráfico en alta resolución", contrariamente al anterior llamado de "baja resolución".

La alta resolución

permite considerar que la pantalla está constituida por multitud de puntos luminosos, llamados "pixels", completamente independientes los unos de los otros. En alta resolución cada punto queda representado por un único Bit en la memoria, pudiendo estar únicamente encendido (bit 1) o apagado (bit 0). Puesto que la pantalla de tu C64 está compuesta por 25 líneas de 40 caracteres, y cada uno de ellos está formado por 8 * 8 puntos, la pantalla quedará dividida en 64000 puntos, dispuestos en 200 líneas de 320 puntos.

Desafortunadamente, Commodore no ha provisto a tu C64 de instrucciones adecuadas para poder dibujar en alta resolución. Para realizar dibujos, gráficos o diagramas, tendremos que recurrir a las instrucciones PEEK y POKE, yendo a trabajar directamente sobre las zonas de la memoria de pantalla que se deseen modificar.

Los gráficos

Dado que como acabas de ver no existen instrucciones gráficas en tu C64, nuestra lección sobre lenguaje resultará esta vez

forzosamente distinta a las anteriores. En lugar de atenernos como de costumbre a instrucciones específicas, insertadas



en fábrica en la memoria de tu ordenador, tendremos que intentar "crearlas" por nuestra cuenta, escribiendo programas que logren simular su funcionamiento. Los tres programas que tomaremos en consideración son:

- 1) Pasar a alta resolución, reservando una parte de la memoria para los gráficos.
- 2) Trazar un punto;

- 3) Trazar una línea recta entre dos puntos deseados. (Ten también en cuenta que además existen en el comercio numerosos libros y manuales que se ocupan exclusivamente de enseñar el uso de los gráficos en el C64. Algunas instrucciones de estos programas podrán resultarte poco claras, requiriendo su explicación el

conocimiento de nociones que quedan fuera de los límites de nuestro curso y en las que por lo tanto no podemos profundizar.) Antes de adentrarnos en estas operaciones será necesario retomar y estudiar a fondo durante un rato el funcionamiento de los operadores lógicos AND y OR, tema que ya conoces desde hace algún tiempo, pero que hoy se hace verdaderamente importante y fundamental. Veamos en primer lugar AND: opera sobre dos números binarios y devuelve el valor 1 en el caso de que ambos números sean 1 y el valor 0 si uno de los dos o ambos fueran 0:

1 AND 1 = 1	1 AND 0 = 0
0 AND 1 = 0	0 AND 0 = 0

Si los números estuvieran formados por varias cifras, la operación se realiza por parejas de números correspondientes:

$$1101 \text{ AND } 1011$$

se ejecuta así:

$$\begin{array}{r} 1101 \\ 1011 \\ \hline 1001 \end{array}$$

Cuando en un programa el operador AND recibe dos números decimales,

después de haberlos transformado en sus correspondientes valores binarios ejecuta la AND lógica y vuelve a transformar su resultado a número decimal. Por lo tanto, la instrucción:

```
PRINT 10 AND 7
```

```
10 decimal -----> 1010 binario
7 decimal -----> 0111 binario
```

```
10 AND 7 -----> 0010 binario -----> 2 decimal
```

provocará la impresión en pantalla del valor 2. Realizar el AND de dos números puede ser útil para conocer el valor de los bits de un byte o para poner a cero un

determinado bit. Así, si por ejemplo deseáramos conocer el contenido del quinto bit de una determinada localización de memoria, supongamos que la 3543, podríamos realizar el AND de ese número con el número binario 10000 (16 decimal), es decir PRINT PEEK (3543) AND 16, y el resultado sería 10000 (es decir, 16) únicamente si el quinto bit contuviera un 1, de lo contrario sería 0. También OR opera sobre dos números, transformándolos en binarios, y generando sobre cada pareja de números correspondientes el valor 0 si ambos números fueran 0 y el valor 1 en todos los demás casos.

```
0 OR 0 = 0      0 OR 1 = 1
1 OR 0 = 1      1 OR 1 = 1
```

por lo tanto,

```
12 OR 6
```

se realiza de la siguiente manera:

```
12 decimal -----> 1100 binario
6 decimal -----> 0110 binario
12 OR 6 -----> 1110 binario -----> 14 decimal
```

Los operadores AND y OR sirven, usando un POKE, para modificar un byte leído mediante un PEEK poniendo en 0 ó 1 determinados bits. Por ejemplo, dado el byte 10011101 (157 decimal), para poner a cero el tercer y cuarto bit se usará el número

11110011 (243 decimal) y el operador lógico AND: en efecto, 157 AND 243=145, es decir 10010001. En este punto, se insertará este valor en la localización de memoria leída anteriormente con PEEK y se habrá alcanzado el resultado deseado, es decir, la modificación de dos bits.

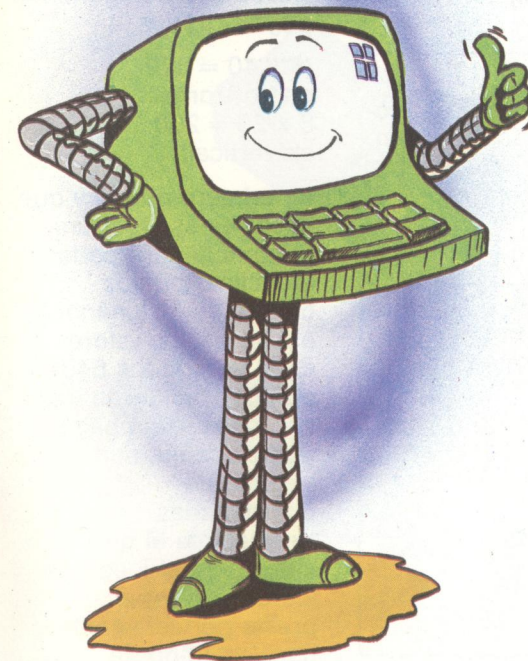
Cómo entrar en alta resolución

Para entrar con tu Commodore en el modo de alta resolución es necesario situar en 1 el bit de la posición 5 de la dirección 53265, reservando además en la memoria el espacio suficiente para la gestión de los gráficos. Esto puede realizarse con las instrucciones:

```
POKE 53265, PEEK (53265) OR 32
POKE 53272, PEEK (53272) OR 8
```

Ejecutando únicamente estas instrucciones, el resultado será sin embargo bastante pobre: la pantalla estará sucia por todos los datos existentes con anterioridad en la memoria y cuyos bits son los que corresponden ahora a los pixels de la pantalla. El único método que tenemos ahora a disposición para limpiar la página gráfica es el de poner a cero cada uno de los bits de dicha página. Las instrucciones a ejecutar —además de las que acabamos de ver— son:

```
FOR I=8192 TO 16191
POKE I, 0
NEXT I
```



Desgraciadamente el BASIC es muy lento para ejecutar una operación de este tipo (aproximadamente 20 segundos), a su término la página gráfica quedará limpia. Lo último que nos queda por hacer es seleccionar el color de la pantalla (es decir, el del fondo) y el color con el que deseamos trazar los distintos dibujos

```
INPUT "COLOR DE FONDO"; CF
INPUT "COLOR DE LA TINTA"; TINT
FOR I = 1024 TO 2023
POKE I,16*TINT+CF
NEXT I
```

Por lo tanto, y situando las instrucciones en un orden más adecuado, el programa será el siguiente:

```
5 PRINT CHR$(147)
10 INPUT "COLOR DEL FONDO"; CF
20 INPUT "COLOR DE LA TINTA"; TINT
25 PRINT CHR$(147)
30 POKE 53265, PEEK (53265) OR 32
40 POKE 53272, PEEK (53272) OR 8
50 FOR I = 8192 TO 16191
60 POKE I,0
70 NEXT I
80 FOR I = 1024 TO 2023
90 POKE I,16*TINT+CF
100 NEXT I
```

Resulta indispensable ejecutar las INPUT en primer lugar, puesto que después de la orden de pasar a alta resolución (líneas 30 y 40) los rótulos no aparecerían en pantalla.

El punto

Ahora que conocemos el camino para entrar en alta resolución y para limpiar la página gráfica, deseamos intentar encender y dibujar un punto determinado entre los 64000 que componen la página gráfica.

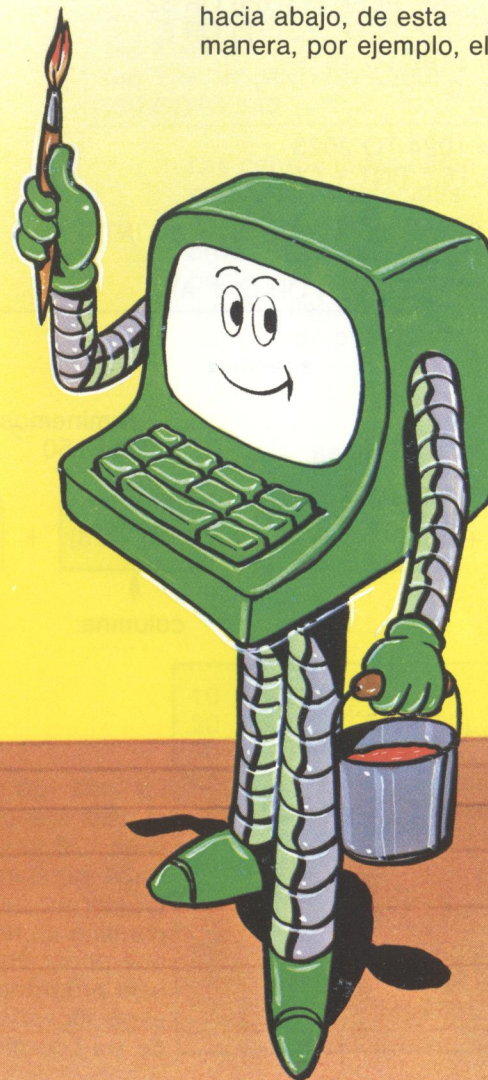
(8 x 40 = 320
en horizontal y
8 x 25 = 200
en vertical)

Lo primero que hay que hacer es escoger una referencia para poder identificar puntos concretos (asignándole así un par de valores a cada uno de los 64000 puntos): las líneas y las columnas de la página gráfica definen un sistema de coordenadas —semejante al que se emplea en el juego de los barcos— que se presta perfectamente para este tipo de

operación. Llamando a las coordenadas X e Y (X para las columnas e Y para las líneas), situamos el origen de nuestro sistema de

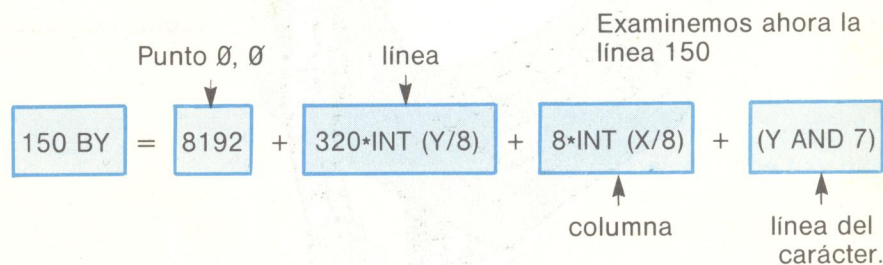
referencia en el ángulo superior izquierdo (convencionalmente se denomina "origen" al punto con coordenadas (0, 0). Orientamos pues el eje de las X de izquierda a derecha, y el de las Y de arriba hacia abajo, de esta manera, por ejemplo, el

ángulo inferior derecho será el punto (319, 199) mientras que el ángulo superior derecho será el punto (319, 0). Nuestro programa para dibujar un punto emplea este método para identificar cada punto de la pantalla.



```

10 PRINT CHR$(147)
20 INPUT "COLOR DEL FONDO"; CF
30 INPUT "COLOR DE LA TINTA"; TINT
40 INPUT "COORDENADA X"; X
50 INPUT "COORDENADA Y"; Y
60 IF X < 0 OR X < 319 OR Y < 0 OR Y < 199 THEN
    PRINT "VALORES INACEPTABLES. REPITE" : GOTO 40
65 PRINT CHR$(147)
70 POKE 53265, PEEK (53265) OR 32
80 POKE 53272, PEEK (53272) OR 8
90 FOR I = 8192 TO 16191
100 POKE I, 0
110 NEXT I
120 FOR I = 1024 TO 2023
130 POKE I, 16 * TINT + CF
140 NEXT I
150 BY = 8192 + 320 * INT (Y/8) + 8 * INT (X/8) + (Y AND 7)
160 A = 7 - (X AND 7)
170 POKE BY, PEEK (BY) OR (2 ↑ A)
    
```



La línea 160 selecciona el píxel a encender, la línea 170 lo enciende. Cada vez que quieras dibujar un punto en la pantalla no tendrás más que poner en marcha este programa especificando, —además de las coordenadas del

punto— únicamente los colores elegidos para tinta y fondo. Una posible (aunque bastante inútil) aplicación de nuestro programa podría ser la

de hacerle trazar puntos al azar a tu Commodore 64. Para poder lograr esto será suficiente con que borres las líneas 40, 50, 60 y añadas las siguientes instrucciones:

```

145 X = INT (RND (0) * 320)
146 Y = INT (RND (0) * 200)
180 GOTO 145
    
```

Las coordenadas serán entonces establecidas automáticamente por el ordenador, creando un bucle infinito que, poco a poco, llenará la pantalla de puntos luminosos.

La recta

Hemos llegado al último programa gráfico, aquél que traza una recta entre dos puntos, cualesquiera. Ahora ya no tenemos problemas especiales, nuestra recta será generada situando puntos, los unos a continuación de los otros, para que en pantalla aparezcan como si constituyeran una única línea continua. Dado que ya sabemos dibujar puntos, será suficiente con determinar los pixels a encender (mediante una sencilla expresión, que tenga en cuenta los extremos a unir) y todo quedará rápidamente solucionado. Sin entrar en detalles matemáticos aburridos, he aquí el listado del programa:

```

10 PRINT CHR$(147)
20 INPUT "COLOR DEL FONDO";CF
30 INPUT "COLOR DE LA TINTA";TINT
40 INPUT "X DEL PRIMER PUNTO";X1
50 INPUT "Y DEL PRIMER PUNTO";Y1
60 INPUT "X DEL SEGUNDO PUNTO";X2
70 INPUT "Y DEL SEGUNDO PUNTO";Y2
80 IF X1<0 OR X2<0 OR X1> 319 OR X2>319
    THEN PRINT "INACEPTABLE. REPITE":GOTO 40
90 IF Y1<0 OR Y2<0 OR Y1>199 OR Y2>199
    THEN PRINT "INACEPTABLE. REPITE":GOTO 40
94 PRINT CHR$(147)
    
```

continúa

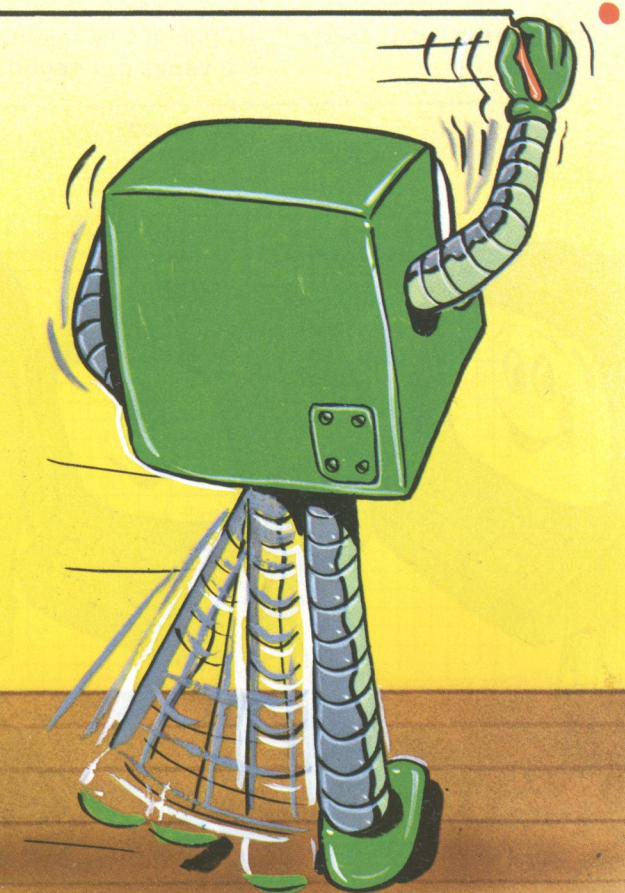
```
95 POKE 53265,PEEK(53265) OR 32
100 POKE 53272,PEEK(53272) OR 8
110 FOR I=8192 TO 16191
120 POKE I,0
130 NEXT I
140 FOR I=1024 TO 2023
150 POKE I,16*TINT+CF
160 NEXT I
170 IF X1>X2 THEN X3=X2:X2=X1:X1=X3:Y3=Y2:Y2=Y1:Y1=Y3
180 X3=X2-X1:Y3=Y2-Y1
190 IF X3=0 THEN 260
200 IF Y3<=X3 AND Y3>=-X3 THEN 320
210 PASO=SGN(Y3):CASO=1
220 FOR I=0 TO Y3 STEP PASO
230 X=I*X3/Y3+X1:GOTO 1000
240 NEXT I
250 GOTO 360
260 IF Y3=0 THEN 360
270 PASO=SGN(Y3):CASO=2
280 FOR I=Y1 TO Y2 STEP PASO
290 X=X1:Y=1:GOTO 1000
300 NEXT I
310 GOTO 360
320 CASO=3
330 FOR I=0 TO X3
340 Y=I*Y3/X3+Y1:X=I+X1:GOTO 1000
350 NEXT I
360 END
1000 BY=8192+320*INT(Y/8)+8*INT(X/8)+(Y AND 7)
1010 A=7-(X AND 7)
1020 POKE BY,PEEK(BY) OR (2↑A)
1030 ON CASO GOTO 240,300,350
```

La función SGN que puedes observar en las líneas 210 y 270 es la llamada función signo. Proporciona como resultado el valor 1 si el argumento es mayor que 0, el valor 0 si el argumento es 0 y el valor -1 si el argumento es negativo.

Como ejercicio (y sin buscarte muchas complicaciones), intenta introducir nuevas instrucciones que consigan (de una forma similar a lo que hemos hecho anteriormente con los puntos) dibujar rectas generadas al

azar por el ordenador. De cualquier manera, si en esta lecciones has encontrado cosas que no comprendes del todo o que te parecen poco claras, no te preocupes demasiado por ello. Como ya te he

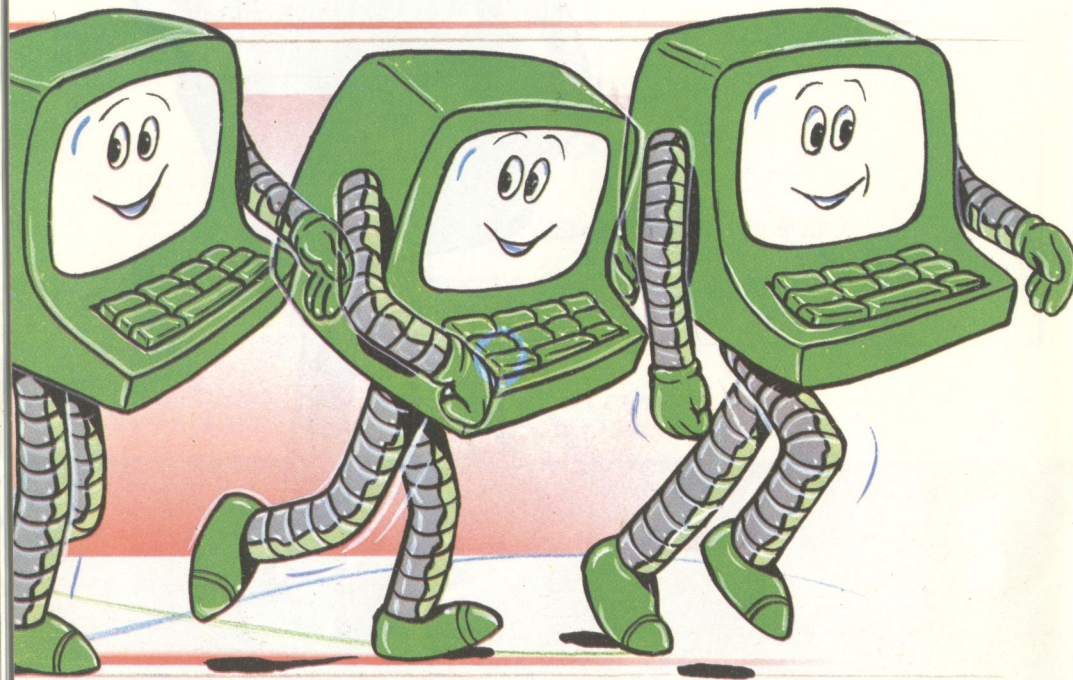
comentado, los gráficos están entre las aplicaciones más difíciles de usar en tu C64, siendo necesarios profundos estudios sobre la máquina, una gran paciencia y una habilidad notable para la programación.



Animación

Las capacidades gráficas de un ordenador se pueden emplear para crear animaciones y mover figuras en pantalla. Para obtener estos efectos se recurre a técnicas completamente semejantes a aquellas empleadas por los creadores de dibujos animados, puesto que es necesario visualizar en pantalla, al menos 4 ó 5 veces por segundo,

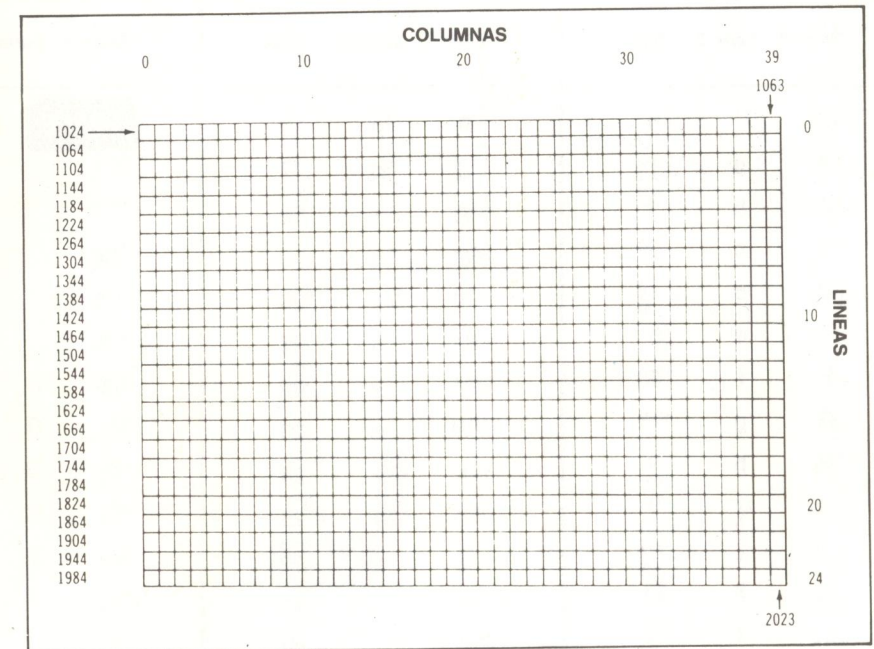
imágenes, cada una de las cuales resulta ligeramente modificada con respecto a la anterior. Esto crea la ilusión de la animación y, si existe además un desplazamiento sobre la pantalla, la del movimiento. Desde el punto de vista de la programación, animar significa modificar adecuadamente los



datos existentes en la memoria de pantalla. Como ya sabes, en el C64 la pantalla está compuesta por 25 líneas, dividida cada una de ellas en 40 caracteres. Existen por lo tanto

1000 posiciones en una pantalla, y cada una puede ser la representación de un carácter contenido en las correspondientes 1000 localizaciones de memoria de la RAM de vídeo.

Mapa de la memoria de imagen (pantalla)



Existen dos métodos para poder visualizar un carácter cualquiera. El primero se sirve de la

instrucción PRINT y eventualmente de las funciones conectadas con ella, TAB y SPC. El segundo sistema se sirve de la instrucción POKE para intervenir en la localización específica a

modificar en la memoria de pantalla.

PRINT "@"

POKE 1024, 0

son expresiones del todo equivalentes. Se trata de dos técnicas

que llevan al mismo resultado pero que actúan de dos formas distintas: PRINT opera con caracteres ASCII, mientras que POKE usa los códigos de pantalla.

PRINT CHR\$(72)
POKE 1024, 8

Códigos de pantalla

Serie 1	Serie 2	Poke-	Serie 1	Serie 2	Poke	Serie 1	Serie 2	Poke
@		0	P	p	16	SPACE		32
A	a	1	Q	q	17	!		33
B	b	2	R	r	18	"		34
C	c	3	S	s	19	#		35
D	d	4	T	t	20	\$		36
E	e	5	U	u	21	%		37
F	f	6	V	v	22	&		38
G	g	7	W	w	23	'		39
H	h	8	X	x	24	(40
I	i	9	Y	y	25)		41
J	j	10	Z	z	26	*		42
K	k	11	[27	+		43
L	l	12	£		28	,		44
M	m	13]		29	-		45
N	n	14	↑		30	.		46
O	o	15	←		31	/		47

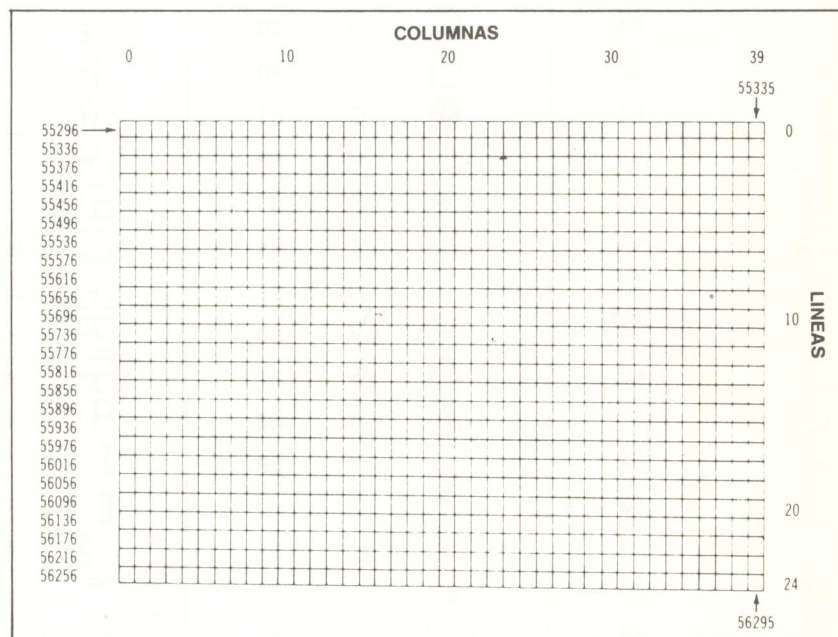
Serie 1	Serie 2	Poke	Serie 1	Serie 2	Poke	Serie 1	Serie 2	Poke
0		48		K	75			102
1		49		L	76			103
2		50		M	77			104
3		51		N	78			105
4		52		O	79			106
5		53		P	80			107
6		54		Q	81			108
7		55		R	82			109
8		56		S	83			110
9		57		T	84			111
:		58		U	85			112
;		59		V	86			113
<		60		W	87			114
=		61		X	88			115
>		62		Y	89			116
?		63		Z	90			117
		64			91			118
	A	65			92			119
	B	66			93			120
	C	67			94			121
	D	68			95			122
	E	69		SPACE	96			123
	F	70			97			124
	G	71			98			125
	H	72			99			126
	I	73			100			127
	J	74			101			127

Existe además otra zona de la RAM que es análoga a la zona de la memoria de vídeo, en la cual en cada una de las 1000 localizaciones que contiene, residen las informaciones

relativas, referentes al color a atribuirle al carácter correspondiente. Este área, denominada

MAPA DE COLOR, arranca en la localización 55296 y se extiende hasta la 56295, ambas inclusive.

Mapa de la memoria de color



Los valores a escribir en estas localizaciones con POKE, para determinar así el color del carácter son:

0 NEGRO	4 PURPURA	8 NARANJA	12 GRIS 2
1 BLANCO	5 VERDE	9 MARRON	13 VERDE CLARO
2 ROJO	6 AZUL	10 ROJO CLARO	14 AZUL
3 AZUL VERDOSO	7 AMARILLO	11 GRIS 1	15 GRIS 3

Si por ejemplo, deseas hacer aparecer una A de color amarillo en la 2.^a columna de la 1.^a línea de la pantalla, tendrás que teclear:

```
POKE 55297, 7
POKE 1025, 1
```

El mapa de color está dispuesto igual que el de pantalla, con la única diferencia de que las localizaciones están situadas 54272

posiciones más adelante. Esto permite obtener con toda facilidad el número de la posición del lugar del

mapa de color correspondiente al carácter a visualizar. Para hacer aparecer en Blanco una B en la 3.^a columna de la 2.^a línea puedes teclear:

```
POKE 1066, 2 : POKE 1066 + 54272, 1
```

POKE también permite visualizar un carácter gráfico definido. Examinemos el caso más sencillo, aquél que únicamente considera una sola posición del carácter; veamos ahora qué es lo que hay hacer para animar un personaje dibujado por nosotros.

```
10 POKE 56334, PEEK (56334) AND 254
20 POKE 1, PEEK, (1) AND 251
30 FOR I = 0 TO 1023
40 POKE 8192 + I, PEEK (53248 + I)
50 NEXT I
60 POKE 1, PEEK (1) OR 4
70 POKE 56334, PEEK (56334) OR 1
90 RESTORE
100 FOR I = 0 TO 15
110 READ A : POKE 8192 + I, A
120 NEXT I
130 DATA 24, 24, 60, 90, 153, 36, 36, 102
140 DATA 153, 90, 60, 24, 24, 36, 36, 102
190 PRINT "█"
200 POKE 53272, (PEEK (53272) AND 240) OR 8
300 POKE 55296, 1
310 FOR C = 1 TO 100 : POKE 1024, 0
320 FOR R = 1 TO 60 : NEXT R
330 POKE 1024, 1
340 FOR R = 1 TO 60 : NEXT R
350 NEXT C
```


24	0	0	0	1	1	0	0	0	
24	0	0	0	1	1	0	0	0	
60	0	0	1	1	1	1	0	0	
90	0	1	0	1	1	0	1	0	
153	1	0	0	1	1	0	0	1	153
36	0	0	1	0	0	1	0	0	
36	0	0	1	0	0	1	0	0	
102	0	1	1	0	0	1	1	0	102

Desde la línea 10 a la línea 70 el programa se ocupa de copiar los primeros 128 caracteres (1024 bytes) desde la ROM de caracteres a la RAM, a partir de la localización 8192. En las líneas 100 a 140 se cambian los dos primeros caracteres del código de pantalla (@ y A) por el personaje en sus dos posiciones. Después de haber limpiado la pantalla

(190) la línea 200 apunta al nuevo mapa de pantalla y habilita el juego de caracteres ahora en RAM, incluidos los dos personajes. La parte comprendida entre la línea 300 y el final del programa se ocupa de la animación, obtenida escribiendo alternativamente con la instrucción POKE en el ángulo superior izquierdo de la pantalla las dos posiciones del personaje. Obseva la presencia de dos bucles de retardos en las líneas 320 y 340. Modificando el valor final del contador puedes obtener diferentes velocidades de movimiento. Un resultado idéntico se obtiene substituyendo las líneas 310 y 330 por las siguientes:

```
310 FOR C = 1 TO 100 : PRINT "☐@"
330 PRINT "☐A"
```

El carácter "☐" (HOME) situado antes del carácter gráfico es necesario para forzar la impresión en la misma posición anterior. La animación de las figuras, hasta de aquéllas muy complejas, está basada sobre estos elementos y técnicas, lo que cambia es el número de caracteres a controlar pero las reglas siguen siendo las mismas.

Movimiento de un carácter

El objeto de este programa es mover por la pantalla un carácter gráfico definido por nosotros. En lugar de la

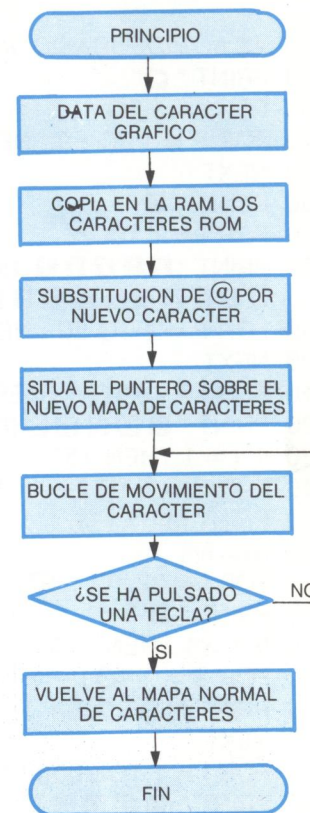
0	0	1	1	1	1	0	0	60
0	1	1	1	1	1	1	0	126
1	1	0	1	1	0	1	1	219
1	1	1	1	1	1	1	1	255
1	1	0	0	0	0	1	1	195
0	1	1	1	1	1	1	0	126
0	1	0	1	1	0	1	0	90
1	1	0	0	0	0	1	1	195

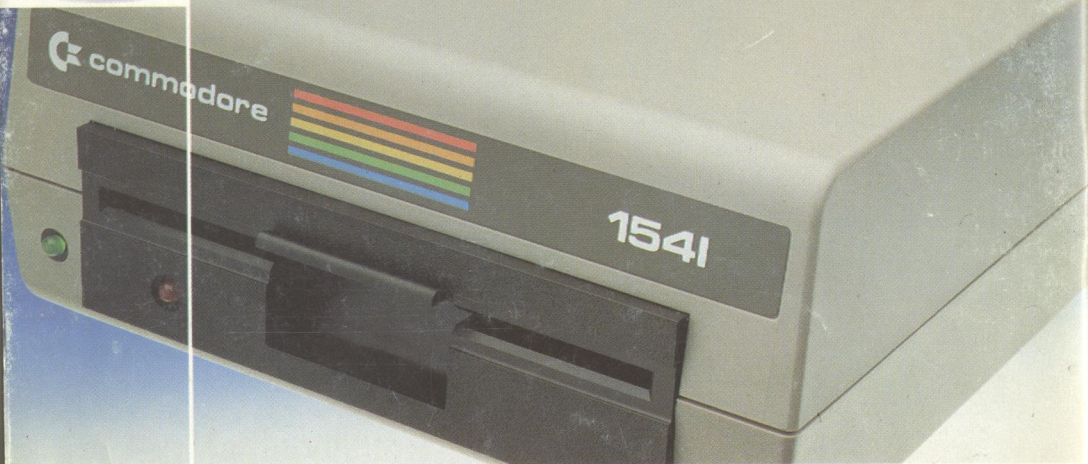
instrucción PRINT, haremos uso de la POKE en el interior de un bucle que imprime directamente el carácter en la memoria de pantalla para crear la ilusión del movimiento.

```
10 DATA 60, 126, 219, 255, 195, 126, 90, 195
15 POKE 56334, PEEK (56334) AND 254
20 POKE 1, PEEK (1) AND 251
25 FOR I = 0 TO 511
30 POKE I + 12288, PEEK (I + 53248) : NEXT
35 POKE 1, PEEK (1) OR 4
40 POKE 56334, PEEK (56334) OR 1
45 FOR C = 0 TO 7 : READ A : POKE 12288
+ C, A : NEXT
50 PRINT "☐" TAB (15) "MOVIMIENTO"
55 POKE 53272, (PEEK (53272) AND 240) + 12
60 LET M 1 = 1224 : LET M 2 = 55496
65 FOR I = 0 TO 22 : POKE M 1 + I, 0
70 POKE M 2 + I, 1
75 FOR D = 1 TO 90 : NEXT D
80 POKE M 1 + 1, 32 : NEXT
95 GET A$ : IF A$ = " " THEN GOTO 60
100 POKE 53272, 21 : END
```

La línea 10 contiene los datos decimales correspondientes a los bytes del carácter definido. Las instrucciones de las

líneas 15 a la 40 copian al nuevo mapa de memoria los primeros 64 caracteres, haciéndolo directamente desde la ROM (53248). El bucle en la línea 45 modifica los bytes que componen el carácter, substituyéndolo con el codificado en las líneas DATA. Después de haber situado —en la línea 55— el puntero para que el C64 busque la forma de los caracteres a imprimir en el nuevo mapa de caracteres, las instrucciones incluidas dentro del bucle FOR..NEXT entre las líneas 65 y 80 se ocupan del movimiento del carácter.





Su Commodore 64 tiene mucho que decirle. Unidad de Disco.

El Commodore 64 es el resultado de la experiencia internacional de Commodore como líder indiscutible en el mercado de los microordenadores.

El Commodore 64 es el ordenador más completo y potente de su categoría, ... pero todavía tiene mucho que decirle.

Por ejemplo su Unidad de Disco.

Sienta como aumenta notablemente la capacidad de memoria de su C-64, como agiliza la carga y descarga de programas y facilita la localización, casi instantánea, de cualquier dato.

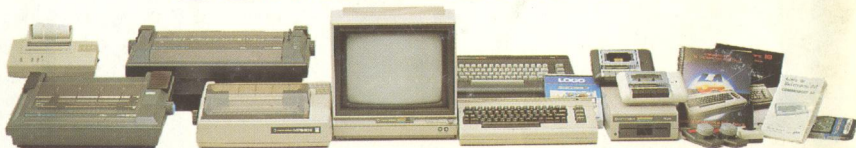
Amplie las posibilidades de su C-64, descubriendo su extensa gama de periféricos.

Ahora que ya sabe que su Commodore 64 tiene todavía mucho que decirle, prepárese a conocerle mejor.

PRINCIPALES CARACTERISTICAS

- 170 K de capacidad - Ficheros secuenciales y relativos y de acceso directo - Unidad inteligente, con sistema operativo incorporada.

commodore 64



Microelectrónica y Control

c/ Valencia, 49-53 08015 Barcelona - c/ Princesa, 47 3.º G 28008 Madrid
Unico representante de Commodore en España.

© 1985 Commodore