

MICROHOBBY

AMSTRAD

Semanal

AÑO II N.º 41

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

160 Ptas.

Canarias 165 pts.

COMO FUNCIONA
EL MACROENSAMBLADOR
DE MICROSOFT EN CP/M

**DISPANT:
MAGIA HECHA
REALIDAD EN
LA PANTALLA
GRAFICA DE TU
AMSTRAD**

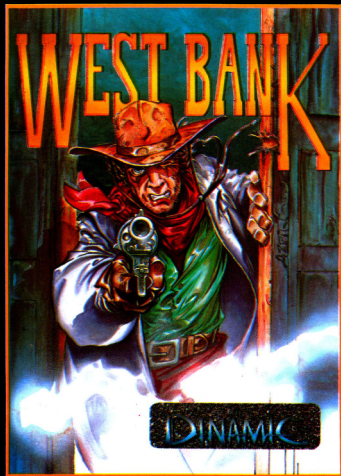
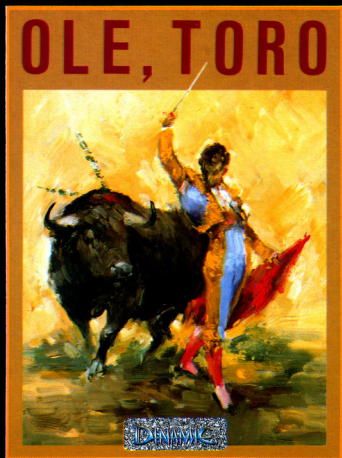
La **Lógica de los
Trucos y técnicas
de Dibujo es...**
binaria, claro.

**INFORMAT 86:
EL TODO INFORMATICO
SE REUNIO EN
BARCELONA**

**ON THE RUN:
POKER DE ASES**



AMSTRAD SIN LIMITES



CAMELOT WARRIORS

El misterio, la fantasía y la más sofisticada técnica en diseño gráfico han hecho posible este clásico de DINAMIC. Camelot Warriors. Imaginación sin límite.

OLE, TORO

Tres elementos se dan cita en el rito de la arena: la vida, la muerte y el arte. Ole, Toro. Originalidad sin límite.

WEST BANK

La rapidez de reflejos es la clave de tu supervivencia, y los nervios de acero un seguro para la victoria. West Bank. Adictividad sin límite.

AMSTRAD

sumario

Año 11 • Número 41 • 10 al 16 de Junio de 1986
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas.+10 ptas. sobretasa aérea
Ceuta y Melilla, 155 ptas.

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.º Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

Fernando Choumel

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sopena

Robert Chawin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Alberto Suárez

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Chema Sacristán

Portada

Javier Igual

Ilustradores

J. Igual, J. Fons, F. L. Frontón,

J. Sepián, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

Mario Andrino

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

Tel: (93) 303 10 22/313 71 62

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.º Rosa González

M.º del Mar Calzada

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid

Teléfonos: Suscrip.: 734 65 00

Redacción: 734 70 12

Dto. Circulación

Paulino Blanco

Distribución

Coediz, S. A. Valencia, 245

Barcelona

Imprime

ROTEAD, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novacomp, S.A.

Nicolas Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Déposito Legal:

M-28468-1985

Derechos exclusivos de la revista**COMPUTING with the AMSTRAD**

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud

Américo 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

5 Primera plana

La revolución del triple scroll en el «Way of the Tiger».

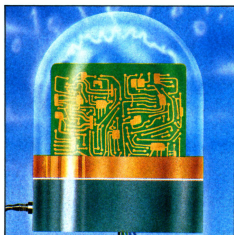
6 Primeros pasos

Dentro de nuestra serie relativa al manejo de los gráficos bit a bit en el **Amstrad**, nos toca analizar con profundidad la función XOR, que así, tan anodina como parece, resulta que tiene mucha «miga» gráfica.



12 CP/M Help

Una de las herramientas más potentes para el desarrollo de programas en lenguaje máquina bajo CP/M es el macroensamblador de Microsoft, el M80. Vamos a aprender a manejarlo rápidamente.



16 Serie Oro

Los gráficos ejercen sobre los «amstradviciosos» una atracción muy grande y, sobre todo, muy lógica. Por eso, para hacerlos la vida un poco más fácil, os presentamos a DISPANT, un programa de diseño de pantallas y gráficos en general bastante increíble.



20 Mr. Joystick

El boxeo fue, en su día, uno de los reyes de nuestros foros deportivos. Lo siga siendo o no, algo debe de tener, porque cada vez hay más juegos por ordenador que lo simulan, como el boxeo tridimensional de esta semana.

24 Informat 86

Hemos estado allí, y vosotros mismos juzgaréis, a través de nuestras palabras, si la feria ha merecido o no la pena.

¡No estamos para juegos!

LO NUESTRO ES HACER BUENAS GESTIONES



FACTURACION. Solo teclee un código y salen todos los datos del cliente. Numeración correlativa automática. Admite 30 productos distintos por factura. Automáticos, descuentos, cargos, IVA. Proporciona 5 totales por factura. (P.V.P. 15.300 incl. IVA.)

PRESUPUESTOS. Guarda en memoria los presupuestos y extiende las facturas. Conceptos de 200 caracteres cada uno (3 renglones de escritura). (P.V.P. 18.300 incl. IVA.)

CUENTAS, PROVEEDORES, BANCOS, CLIENTES. 3 ficheros separados. Resúmenes totales, unitarios o parciales. El mejor auxiliar de CONTABILIDAD al día. (P.V.P. 8.600 incl. IVA.)

CONTROL DE ALMACEN IVA. Código de 9 dígitos alfanuméricos. 25 dígitos denominación. Una sola pantalla entradas y salidas, con visión de asientos anteriores. Stocks máximo, mínimo y avisa para reaprovisionamiento. Totales entradas y salidas cada pantalla. (P.V.P. 15.300 incl. IVA.)

CLIENTES (con etiquetas). 11 campos distintos para localización. Etiquetas 4 modelos distintos en salida de dos. El más fiel auxiliar ahorrador de tiempo. (P.V.P. 8.600 incl. IVA.)

RECIBOS. Resuelve el problema interminable a asociaciones, comunidades, colegios. Fijos los campos del normalizado y 12 campos libres (4 numéricos con cálculos automáticos). Liquidaciones bancos. (P.V.P. 18.300 incl. IVA.) Con numeración automática (21.200 incl. IVA.)

RESTAURANTES. Tratamiento de minuta y facturas. Resúmenes por grupos. Mesas abiertas permanentemente, correcciones, cambios, etc., hasta emisión fra. final. (P.V.P. 35.000 incl. IVA.)

IVA POR ALMACEN. Rellena liquidaciones Hacienda. Introduce cuentas IVA gastos. (P.V.P. 18.900 incl. IVA.)

URBANIZACIONES. Lectura y tratamiento de contadores consumos (agua, gas, luz, etc.). Extensión recibos y totalizaciones bancos. Emisión etiquetas. (P.V.P. 40.000 incl. IVA.)

LIBROS DEL IVA. Controles de repercutido y soportado orden numérico. Resúmenes estudios comparativos. Rellena liquidación Hacienda. (P.V.P. 16.800 incl. IVA.)

FACTURACION Y ALMACEN. Gestión unida. Ficheros clientes, productos, descuentos y cargos. Todos los resúmenes. (P.V.P. 18.900 incl. IVA.)

COTIZACIONES. El mejor cuadro comparativo de precios. Le dice el mejor precio proveedor. (P.V.P. 26.300 incl. IVA.)

1 AÑO DE GARANTIA



**informática
GROTUR, S.A.**

C/ JAIME EL CONQUISTADOR, 27
28045 MADRID Tno. 474 55 00

474 55 32

Télex: IGSA 48452

WEST BANK, DE DINAMIC, PARA AMSTRAD

Nacida con la fiebre del oro que barrió el Oeste, tienen entre sus más prestigiosos establecimientos el **West Bank**, institución que provoca los tiroteos más concurridos al sur de Dakota.

Todos anhelan la riqueza que se esconden tras los muros del banco. Tu sangre fría es el único amuleto que puede impedirlo.

Los personajes

Green Jordan
Jack Vicious
Daisy
El mexicano
Bowie
Julius



El juego

Las oficinas del West Bank constan de 9 puertas, tu misión consiste en conseguir depósitos de dinero en cada una de ellas.

Perderás vidas cada vez que recibas un impacto de bala o dispares sobre alguien inocente.

El duelo

Entre fase y fase deberás superar un duelo contra tres asesinos a sueldo.

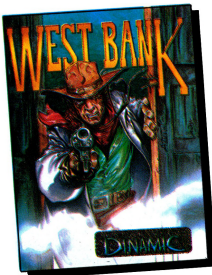
Si eres rápido podrás ganar muchas vidas y numerosos puntos, si eres lento probarás el sabor del polvo de Soft City.

El final

La defensa del West Bank es una empresa para los fuertes. Todos intentarán resistir dos días completos los incesantes tiroteos.

¿Cuál es el misterio que se oculta tras el segundo día? El interrogante queda en el aire.

Los secretos y los tesoros, fueron inventados sólo para los vencedores.



THE WAY OF THE TIGER

Siempre aparece un juego especial que aborda un tema conocido, pero que lo trata de una forma tan revolucionaria e ingeniosa que le dota de un aire completamente nuevo y de una vida y un interés mucho más largos.

Este es el caso del «Way of the tiger», el Camino, del tigre, que cuenta la historia de las aventuras y desventuras de un guerrero Ninja, experto en todo tipo de lucha cuerpo a cuerpo.

Hasta aquí todo normal, pero, mediante una nueva técnica de programación, conocida como «triple scroll», se ha conseguido un efecto

de tridimensionalidad y movimiento simplemente increíble. Merece la pena verlo y jugarlo.

El juego ha sido creado por Grem-lin Graphics y lo distribuye ERBE. Erbe software esta en: Santa Engracia, 17 28010 Madrid



Primera PLANA

NOTICIAS DEL CP/M USER GROUP

Los clubes de usuarios de CP/M son asociaciones de personas, usuarios, que están interesados en colaborar juntos para conseguir el máximo rendimiento de su ordenador, que, naturalmente, ejecuta el sistema operativo CP/M.

Para ello, cuentan con ofertas especiales de programas CP/M de todo tipo, además de lo que se conoce como software de dominio público, el cual todo el mundo puede usar y que se encuentra en las «bibliotecas» del club.

Estos programas, gratuitos o casi, abarcan también casi todo el espectro de aplicaciones informáticas.

Tenemos noticias de un club CP/M en Inglaterra, y vamos a dar aquí su dirección para todos aquellos que se encuentren interesados en adquirir mayor información del tema o, incluso, en formar parte del asunto.

El teléfono es el siguiente: (01) 247 06 91.

CPMSDOSUGUK

THE JOURNAL
OF THE
CP/M
USERS' GROUP
(U.K.)

Pascal Validation
Compiler Reports

by John Sauer

Reprinted by permission of CP/M User Group U.K. © 1985

vol.2

no.7

December 1985

LA FUNCION GRAFICA XOR

Ya hemos visto la manera de utilizar prácticamente la función lógica XOR para luego aplicarla con éxito a cualquiera de las rutinas gráficas que se pasean por nuestra mente. Con ella va a sernos posible dibujar y borrar formas y colores en la pantalla sin perturbar el resto de lo que allí aparezca.



gamos una especie de resumen de su forma de trabajo:

En el Programa I hemos coloreado en rojo un rectángulo. Cuando pulsemos una tecla una línea azul atravesará dicho rectángulo de lado a lado dividiéndolo en dos partes iguales.

Programa uno

Pulsando de nuevo cualquier otra tecla se borra la línea y volvemos a tener nuestro rectángulo original. Este proceso se repetirá una y otra vez mientras sigamos presionando una tecla.

```
10 REM PROGRAMA I
20 HOME 0
30 FDR franja=100 TO 500 STEP 4
40 MOVE franja,0
50 DRAW franja,300,3
60 NEXT
70 WHILE -1
80 PRINT CHR$(23);CHR$(1)
90 WHILE INKEY="" :WEND
100 MOVE 100,150
110 DRAW 500,150,3
120 WEND
```

Y todo esto lo conseguimos mediante la función lógica XOR que hace aparecer y desaparecer la recta. La línea 80 ha ordenado al micro que dibuje en modo XOR mientras que la 90 detiene el programa y espera hasta que pulsamos una tecla para continuar. A continuación traza de nuevo la línea recta.

Como recordará, cuando hacemos un XOR de algo consigo mismo un par de veces, volvemos a obtener ese algo en su estado original.

Supongamos que queremos dividir el rectángulo en otros dos más pequeños, pero no estamos muy seguros a la hora de elegir el sitio por donde hacerlo.

El Programa II es similar al anterior, pero en este caso nos va dar la oportunidad de cambiar la posición de la línea divisoria.

Programa dos

La línea de corte está colocada en un principio en el centro del rectángulo como antes, pero al pulsar cualquiera de las teclas de los cursores que indican arriba o abajo, la recta puede moverse a otra posición. Y de hecho se mueve, ¡yaya que sí!

El programa funciona de la siguiente manera: las líneas 10 a 60 se utilizan para dibujar el rectángulo mientras que la 70 fija el valor inicial de la coordenada *y* de la recta. La 80 llama a la subrutina que empleamos para dibujar la línea azul.

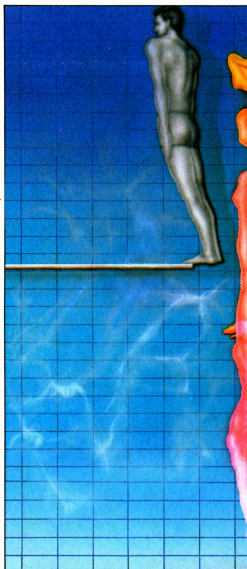
Cuando el ordenador detecta que hemos pulsado una tecla del cursor —líneas 100 a 110— el programa salta a la rutina de la 130 y traza de nuevo la recta en modo XOR. El resultado conseguido es que la línea se borra y desaparece de nuestra vista.

El valor de la variable *ya* se incrementa o decrementa, dependiendo de la tecla que hayamos pulsado, y volvemos a usar ahora la subrutina para dibujar de nuevo la recta, pero en esta ocasión, en una nueva posición.

El proceso se repite cada vez que presionemos una de las teclas del cursor. Las líneas 130 y 140 simplemente evitan que la línea se salga de los límites superior o inferior del rectángulo.

En este Programa II hemos utilizado el micro como ayuda para colocar la recta divisoria en la posición que nosotros queramos. Acabamos de dar el primer paso en el campo del diseño asistido por ordenador habitualmente conocido como CAD.

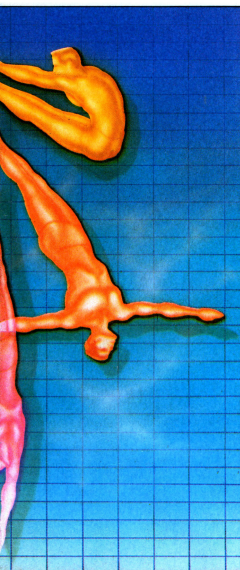
Pejo



Podemos abordar ahora una nueva etapa en nuestro camino. Imagine por un momento que es un arquitecto y desea resolver el problema que le representa colocar la puerta de la calle de una casa en su posición correcta.

```
10 REM PROGRAMA II
20 HOME 0
30 FDR franja=100 TO 500 STEP 4
40 MOVE franja,0
50 DRAW franja,300,3
60 NEXT
70 y=150
80 GOSUB 130
90 WHILE -1
100 IF INKEY(0)=0 THEN GOSUB 130:y=y+2:GOSUB 130
110 IF INKEY(2)=0 THEN GOSUB 130:y=y-2:GOSUB 130
120 WEND
130 IF y<0 THEN y=0:RETURN
140 IF y>300 THEN y=300:RETURN
150 PRINT CHR$(23);CHR$(1)
160 MOVE 100,y
170 DRAW 500,y,3
180 RETURN
```

El Programa III dibuja nuestra casa, representada por un rectángulo, y coloca una puerta verde a la izquierda de la pantalla. Usando las teclas del cursor que indican izquierda y derecha podemos desplazar la puerta de un



a la izquierda o la derecha. Con este sistema parece que estamos moviendo toda la puerta de un lado a otro de la pantalla.

La Figura 1 nos muestra la puerta dividida en un determinado número de franjas verticales. Para moverla hacia la derecha todo lo que necesitamos hacer es borrar la franja del lado izquierdo y dibujar otra a la derecha del límite de la puerta.

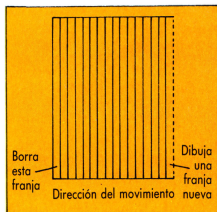


Figura 1. Cómo se mueve la puerta.

Observe que de nuevo utilizamos las instrucciones de las líneas 150 y 160 para detectar la tecla que hemos pulsado y producir un movimiento hacia la izquierda o hacia la derecha. ¿Ha visto que son diferentes?

Una de ellas llama primero a la subrutina y después cambia el valor de la coordenada **X** mientras que la otra modifica dicho valor y después hace el salto a la rutina. Con esto nos permite utilizar una misma subrutina para producir los desplazamientos izquierdo y derecho.

```

10 REM PROGRAMA III
20 MODE 0
30 INK 15,18
40 FOR franja=100 TO 500 STEP 4
50 MOVE franja,0
60 DRAW franja,300,3
70 NEXT
80 PRINT CHR$(23);CHR$(1);
90 FOR franja=0 TO 48 STEP 4
100 MOVE franja,0
110 DRAW franja,150,12
120 NEXT
130 X=0
140 WHILE -1
150 IF INKEY(1)=0 THEN GOSUB 180: X=X+4
160 IF INKEY(8)=0 THEN X=X-4:GOSUB 180
170 WEND
180 MOVE X,0
190 DRAW X,150,12
200 MOVE X+52,0
210 DRAW X+52,150,12
220 RETURN
  
```

Si bien todos estos programas son unos ejemplos muy sencillos, podemos utilizar estos mismos principios básicos para el diseño y desarrollo de programas más complejos.

El Programa IV nos va a dibujar una casa pero de un modo mucho más sofisticado.

Primeros PASOS

Programa cuatro

Hemos prescindido del uso del color y simplificado la utilización de las líneas básicas que componen el gráfico de la estructura de la casa por una mayor claridad a la hora de interpretar el programa.

Primero aparece dibujado en la pantalla el contorno de nuestro **«chalet de la sierra»**. Vamos a disponer, a continuación, de la posibilidad de añadirle puertas y ventanas en la posición que nosotros deseemos.

Después de trazar la casa —líneas 40 a 130— el programa está esperando a que pulsemos cualquiera de las teclas P o V. Si presionamos la P se dibuja una puerta y si es la V una ventana.

Cuando aparece cualquiera de estos dos elementos, podemos moverlo a lo largo y ancho de la pantalla empleando para ello las teclas de los cursores.

La técnica utilizada es similar a la de los programas anteriores. Cuando se detecta e identifica qué tecla del cursor hemos pulsado (líneas 240-300), se llama a la subrutina de la 330 para borrar el elemento elegido de la posición anterior y después calcular la nueva.

Volvemos a llamar a la subrutina, y volvemos a dibujar la puerta (o ventana) en la siguiente localización.

De esta forma somos capaces de mover objetos por la pantalla. Pero también podemos intentar cambiar su tamaño.

Pulsando las teclas **←** **→** **↑** **↓** reduciremos o agrandaremos el elemento elegido. Para ello cambiamos el valor de la variable **«S»**, de modo que cada vez se va a dibujar la **«forma»** con un tamaño distinto.

Una vez hayamos colocado nuestra puerta o ventana en su posición correcta podemos fijarla allí pulsando la tecla **«Copy»**.

Con ello damos el valor cero a la variable **«código»** y esto nos da como resultado que la figura vuelva a ser dibujada, pero esta vez empleando la lógica **«normal»**. Así nos aparece un dibujo que se queda fijo en la pantalla.

Después de haber pulsado la tecla **«Copy»**, el cursor regresa a la esquina superior izquierda y se prepara para recibir otro elemento según presionemos la tecla V o P.

Quizá le estén dando ganas de añadir formas nuevas de su cosecha particular al programa. Bastaría con añadirles a la subrutina a partir de la línea 540 y sería necesario re-

lado a otro de la casa hasta encontrar su colocación más conveniente.

Programa tres

Una recomendación. Sea muy cuidadoso cuando ejecute los programas de esta semana. La función lógica empleada para los gráficos que hayamos utilizado en los programas anteriores podrá afectar seriamente nuestro trabajo.

Podemos resolver este problema pulsando **«Ctrl»**, **«Shift»** y **«Espace»** para poner a cero e inicializar (o «resetear») el micro. Quizá resulte un poco pesado, pero funciona correctamente y nos evita complicaciones posteriores.

El Programa III es muy semejante al anterior, pero observe que ahora movemos un objeto sólido mientras que antes solamente desplazábamos una única línea.

Para ello no es necesario, sin embargo, dibujar la puerta completa repetidas veces. Sencillemente vamos haciendo un XOR con los límites de la misma pero cada vez un poco más

poner la instrucción RETURN al final de esta nueva rutina.

El objeto que acaba de añadir sería seleccionado si damos el valor 3 a la variable «for» cuando pulsemos la tecla apropiada.

```

10 REM PROGRAMA IV
20 MODE 0
30 REM DIBUJAR CASA
40 MOVE 0,50
50 DRAW 639,50
60 MOVE 200,50
70 DRAW 200,300
80 DRAW 600,300
90 DRAW 600,50
100 MOVE 200,300
110 DRAW 250,350
120 DRAW 550,350
130 DRAW 600,300
140 WHILE =1
150 WHILE for=0
160 IF INKEY(55)=0 THEN for:=1
170 IF INKEY(27)=0 THEN for:=2
180 MEND
190 s:=100
200 codi go=1
210 x=0;y:=399
220 GOSUB 330
230 WHILE codi go=1
240 IF INKEY(39)=0 THEN GOSUB 330:s
   s=-2:GOSUB 330
250 IF INKEY(31)=0 THEN GOSUB 330:s
   s=2:GOSUB 330
260 IF INKEY(0)=0 THEN GOSUB 330:y=
   y+2:GOSUB 330
270 IF INKEY(2)=0 THEN GOSUB 330:y=
   y-2:GOSUB 330
280 IF INKEY(8)=0 THEN GOSUB 330:x=
   x-4:GOSUB 330
290 IF INKEY(1)=0 THEN GOSUB 330:x=
   x+4:GOSUB 330
300 IF INKEY(9)=0 THEN codi go=0:GOS
   UB 330:for:=0
310 MEND
320 MEND
330 PRINT CHR$(23);CHR$(codi go);
340 MOVE x,y
350 IF for:=2 THEN GOTO 450
360 DRAWR s,0
370 DRAWR 0,s
380 DRAWR -s,0
390 DRAWR 0,s
400 DRAWR 0,5*s,0
410 DRAWR 0,s
420 DRAWR -0,5*s,0,5*s
430 DRAWR s,0
440 RETURN
450 DRAWR 0,75*s,0
460 DRAWR 0,-1,5*s
470 DRAWR -0,75*s,0
480 DRAWR 0,1,5*s
490 DRAWR 0,15*s,-0,2*s
500 DRAWR 0,45*s,0
510 DRAWR 0,-0,45*s,0
520 DRAWR -0,45*s,0
530 DRAWR 0,0,45*s
540 RETURN

```

Finalmente necesitaremos analizar dicho valor de un modo análogo a como se hace en la línea 350 para enviar al programa a la parte final (añadida) de la subrutina.

La función lógica XOR se usa habitualmente para hacer aparecer y desaparecer cualquier objeto y moverlo de un lado a otro de la pantalla. En realidad, lo que estamos haciendo, sencillamente, es borrarlo de su antigua posición y volver a dibujarlo en la nueva.

Esta técnica se usa con bastante frecuencia aplicándola a los juegos. El Programa V dibuja una especie de dial giratorio que se puede utilizar como radar o como panel de exploración de algún juego espacial o de simulación.

Programa cinco

Las líneas 40 a 60 dibujan un círculo. A continuación se traza una recta desde el centro hasta el perímetro del círculo, representando la aguja.

Este explorador va a girar en uno o en otro sentido según pulsemos las teclas de los cursores que señalan a la izquierda o a la derecha.

Usando el método que hemos estado viendo anteriormente, la subrutina de la línea 130 va haciendo XOR repetidas veces y hace aparecer y desaparecer la aguja. Las líneas 100 y 110 eligen el sentido de giro de la misma incrementando o decreciendo el valor de la variable «ángulo».

Aunque en este programa controlamos el movimiento con las teclas del cursor, podríamos hacer lo mismo incorporando al programa variables cuyo valor determinaría si hay

```

10 REM PROGRAMA V
20 MODE 1
30 DEG
40 FOR angulo=0 TO 360
50 PLOT 100*SIN(angulo)+320,100*COS
   (angulo)+200
60 NEXT
70 angulo=1
80 GOSUB 130
90 WHILE =1
100 IF INKEY(1)=0 THEN GOSUB 130:an
   gulo=angulo+1:GOSUB 130
110 IF INKEY(8)=0 THEN GOSUB 130:an
   gulo=angulo-1:GOSUB 130
120 MEND
130 PRINT CHR$(23);CHR$(1);
140 IF angulo=361 THEN angulo=1
150 IF angulo=0 THEN angulo=360
160 MOVE 320,200
170 DRAW 100*SIN(angulo)+320,100*CO
   S(angulo)+200
180 RETURN

```

movimiento o no, y de haberlo, en qué sentido se va a realizar.

Si alguna vez ha utilizado un programa de simulación de vuelo, probablemente no se le habrá pasado por alto que utiliza esta técnica para que nos aparezcan en la pantalla varios de los instrumentos que componen el tablero de control de un avión.

En el Programa VI hacemos uso de los temporizadores que tiene incorporado su Amstrad para realizar un reloj que funciona en tiempo real. El segundo se mueve cada segundo (como es de rigor) usando el temporizador número 1.

Programa seis

La línea 100 le dice al programa que salte a la subrutina de la línea 130 cada segundo y así mueva el segundo rojo.

Del mismo modo, la línea 110 ordena al programa que se desvíe a la subrutina de la línea 150 una vez por minuto para mover el minutero rojo. En esta ocasión se usa el temporizador número 2.

No hemos incluido la manilla horaria por razones evidentes. Sin embargo, podría hacerlo usted mismo, utilizando el temporizador número 3. El único problema radica en el hecho de estar pendiente una hora mirando

```

10 REM PROGRAMA VI
20 MODE 1
30 DEG
40 FOR angulo=0 TO 360
50 PLOT 100*SIN(angulo)+320,100*COS
   (angulo)+200,1
60 NEXT
70 m:=1*s:=1
80 GOSUB 170
90 GOSUB 210
100 EVERY 50,0 GOSUB 130
110 EVERY 3000,1 GOSUB 150
120 WHILE INKEY="" :MEND
130 GOSUB 170:s:=s+6:GOSUB 170
140 RETURN
150 GOSUB 210:m:=m+6:GOSUB 210
160 RETURN
170 PRINT CHR$(23);CHR$(1);
180 MOVE 320,200
190 DRAW 100*SIN(s)+320,100*COS(s)+
   200,2
200 RETURN
210 MOVE 320,200
220 DRAW 75*SIN(m)+320,75*COS(m)+20
   0,2
230 RETURN

```

atentamente la pantalla de su monitor para poder ver los resultados de su trabajo.

Nos acercaremos ahora a la técnica conocida como «goma elástica». Con ella es posible trazar una línea sobre la pantalla utilizando las teclas del cursor una especie de Etch-a-Sketch computerizada.

El Programa VII dibuja una línea con un extremo fijo en el centro de la pantalla. El otro está libre y se desplaza como un cursor cross-hair.

Programa siete

Podemos moverlo por la pantalla usando, no se sorprenda, las teclas del cursor de su Amstrad.

Observará cómo el extremo libre de la recta va a irse alejando o acercando siguiendo las órdenes que le damos con los cursores. Parece que la línea se estira o encoge como una pieza elástica, de ahí el nombre de «goma elástica».

Esta recta es móvil debido a que hemos hecho uso de nuevo de la función XOR que borra y vuelve a dibujar la línea en su nueva posición repetidas veces.

El programa usa dos pares de variables para contener los extremos de la recta «xx» e «yy» serán las coordenadas del fijo, mientras que «ax» y «ay» serán las del libre. La subrutina de la línea 170 simplemente traza una recta entre el punto «xx» «yy» y el «ax», «ay» utilizando la función lógica XOR.

Podemos fijar la recta en cualquier posición de la pantalla añadiendo las siguientes líneas al Programa VII:

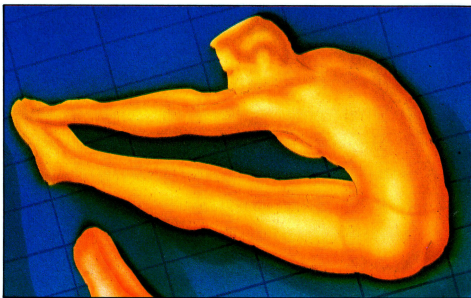
```
150 IF INKEY(9)=0 THEN GOSUB 170
```

: código=0: GOSUB 170: código=1: GOSUB 170

200 IF código=0 THEN RETURN

A continuación, cuando pulsemos la tecla «Copy», primero se borrará la línea y después vuelve a dibujarse con una lógica «normal» y así quedará permanentemente en la pantalla.

La línea 200 impide que el cursor quede también fijo. Finalmente volvemos a trazar la recta con el XOR lógico y de esta forma seguiremos obedeciendo a los mandatos que le demos con las teclas.



Así podemos trazar líneas continuas e incluso conseguiremos hacer algún bonito dibujo.

En un programa anterior dibujamos el contorno de una cama usando datos incorporados al mismo. En este momento ya somos capaces de repetirlo moviendo el cursor por la pantalla y trazando las líneas apropiadas. Y, ¿por qué no intentamos dibujar cualquier otra forma facilitat?

El inconveniente del Programa VII es que solamente podemos trazar con él una línea continua. No nos da la facilidad de levantar la pluma del papel y colocarla en otra posición. Tampoco podemos borrar una línea una vez que ya esté colocada en la pantalla.

El Programa VIII es similar al anterior, pero le hemos añadido las facilidades que el falabon. Utilizamos en él varias teclas cuyas funciones resumidas aparecen en la Tabla II.

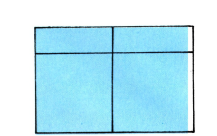
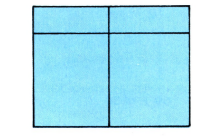
```

10 REM PROGRAMA VII
20 MODE 1
30 INK 3,24
40 código=1
50 x=320
60 y=200
70 a=400
80 b=300
90 GOSUB 170
100 WHILE -1
110 IF INKEY(0)=0 THEN GOSUB 170:
b=4:GOSUB 170
120 IF INKEY(2)=0 THEN GOSUB 170:
b=4:GOSUB 170
130 IF INKEY(1)=0 THEN GOSUB 170:
a=4:GOSUB 170
140 IF INKEY(8)=0 THEN GOSUB 170:
a=4:GOSUB 170
160 MEND
170 PRINT CHR$(23);CHR$(código);
180 MOVE x,y
190 DRAW a,b,código+1
210 MOVE a+10,b
220 DRAW a+10,b
230 MOVE a,b-10
240 DRAW a,b-10
250 RETURN
  
```

Por lo tanto, hemos descubierto la forma de hacer que permanezcan en la pantalla las líneas que nos interesen simplemente pulsando la tecla «Copy». La Tabla I nos muestra los valores utilizados en la selección de la «lógica» usada para dibujar un gráfico.

```

10 REM PROGRAMA VIII
20 REM INICIALIZACION DE VARIABLES
30 MODE 1
40 INK 3,24
50 código=1
60 nib=1
70 x=320
80 y=200
90 a=320
100 b=200
110 REM DIBUJAR CURSOR
120 GOSUB 290
130 REM DETECTAR TECLAS
140 WHILE -1
150 IF INKEY(0)=0 THEN GOSUB 370:b=
b=4:GOSUB 370
160 IF INKEY(2)=0 THEN GOSUB 370:b=
b=4:GOSUB 370
170 IF INKEY(1)=0 THEN GOSUB 370:a=
a=4:GOSUB 370
180 IF INKEY(8)=0 THEN GOSUB 370:a=
a=4:GOSUB 370
190 IF INKEY(9)=0 AND nib=1 THEN GO
SUB 440
200 IF INKEY(36)=0 AND nib=1 THEN G
OSUB 500
210 IF INKEY(69)=0 AND nib=0 THEN B
OSUB 500
220 IF INKEY(79)=0 THEN GOSUB 560
230 MEND
240 REM DIBUJAR CURSOR
250 PRINT CHR$(23);CHR$(1);
260 MOVE a,b
270 DRAWR 0,10,2
280 DRAWR 20,20
290 DRAWR 10,-10
300 DRAWR -20,-20
310 DRAWR -10,0
320 MOVERR 0,10
330 DRAWR 10,-10
340 MOVERR -5,5
350 DRAWR 20,20
360 RETURN
370 REM DIBUJAR LINEA
380 IF nib=0 THEN GOTO 420
390 PRINT CHR$(23);CHR$(1);
400 MOVE x,y
410 DRAW a,b
420 GOSUB 240
430 RETURN
440 REM FICRAR LINEA
450 PRINT CHR$(23);CHR$(0);
460 MOVE x,y
470 DRAW a,b,1
480 x=ai+yb
490 RETURN
500 REM ACERCAR O LEVANTAR LA PLUMA
510 GOSUB 370
520 IF nib=1 THEN nib=0 ELSE nib=1
530 x=ai+yb
540 GOSUB 370
550 RETURN
560 REM BORRAR LINEA
570 PRINT CHR$(23);CHR$(0);
580 MOVE a,b
590 DRAW x,y,0
600 x=ai+yb
:10 RETURN
  
```



En el Programa VII dibujamos todas las líneas desde un punto fijo en el centro de la pantalla. Sin embargo, podríamos intentar mover el extremo fijo de la línea de la misma manera que el libre.

Modifiquemos otra vez el programa cambiando la línea 150 por:

150 IF INKEY(9)=0 THEN código=0: GOSUB 170: x=a: y=b: código=1

De esta manera cuando fijamos una recta en la pantalla, la posición del extremo fijo se desplazará hasta la última posición del libre.

Programa ocho

Es un programa un poco más completo que el VII. Sería conveniente que dedicara un poco de tiempo a su estudio y entender totalmente su forma de funcionamiento.

Y esto es todo por el momento. Esperamos que haya disfrutado «un montón» a la vez que ha continuado su aprendizaje. Y siempre que quiera conseguir una obra «maestra», eche un vistazo a su «MICROHOBBY AMSTRAD».

MICRO-1

C/ Duque de Sesto, 50, 28009 Madrid

Tel.: (91) 275 96 16/274 53 80

(Metro O'Donnell o Goya)

Aparcamiento gratuito en Felipe II

el IVA lo paga
MICRO-1

SOFTWARE: ¡¡2 PROGRAMAS POR EL PRECIO DE 1!!
Y además, completamente gratis, un magnífico reloj de cuarzo. Increíble ¿verdad?

	Plas.
PING PONG	2.295
SABOTEUR	2.295
RAMBO	2.295
YIEAR KUNG FU	2.295
WORLD SERIES BASEBALL	2.095
MAPGAME	2.750
RAID	2.295
HYPERSPORTS	2.295
HIGHWAY ENCOUNTER	1.750
HIGHWAY ENCOUNTER DISCO	3.300
ALIEN B	1.750

	Plas.
DYNAMITE DAN	2.100
SABRE WULF	1.650
THEY SOLD A MILLION	2.500
FIGHTER PILOT	1.975
MASTER OF T. LAMP	1.950
NIGHT SHADE	1.950
HACKER	1.950
SUPER TEST	2.300
TORNADO LOW LEVEL DISCO	3.300
TORNADO LOW LEVEL	1.750
KNIGHT LORE	1.750

SOFTWARE DE REGALO: ¡¡OFERTA 2 x 1!!

Beach Head Decathlon Dummy Run Beach Head Southern Belle

Fabulosos
precios para tu Amstrad
CPC-464 CPC6128
PCW-8256 y
PCW-512

SOFTWARE DE GESTION PROFESIONAL

DBA II	17.800	DR. GRAPH	15.100
CBASIC	15.100	CONTABILIDAD	
DR DRAW	15.100	Y VTOS.	16.600

IMPRESORAS ¡¡20% DTO. SOBRE P.V.P.!!

COMPATIBLE IBM PC-XT 256 K
Y DOS DISKETTES DE 360 K
229.900 PTAS.

UNIDAD DE DISCO 5¼"
PARA AMSTRAD
34.900 PTAS.

LAPIZ OPTICO+INTERFACE
3.495 PTAS.

CINTA VIRGEN ESPECIAL ORDENADORES
69 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
15% DTO.
CASSETTE ESPECIAL ORDENADOR
5.295 PTAS.

JOSTICK QUICK SHOT II
1.995 PTAS.
JOYSTICK QUICK SHOT V
2.295 PTAS.
con la compra de un joystick
¡¡GRATIS 1 RELOJ DE CUARZO!!

DISKETTE 5¼"
295 PTAS.

DISKETTE 3"
990 PTAS.

MICRO-1

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 275 96 16/274 53 80
(Metro O'Donell o Goya)

el IVA lo paga
MICRO-1

1.395
ptas.

QUICK SHOT I+INTERFACE
2.695 PTAS.

1.695
ptas.

QUICK SHOT II+INTERFACE
2.995 PTAS.

1.695
ptas.

QUICK SHOT V+INTERFACE
2.995 PTAS.

NECESITAMOS DISTRIBUIDORES ¡¡GRANDES DESCUENTOS!!

DIPROINSA
DISTR. de PRODUCTOS
INFORMATICOS M., s.a.

C/ GALATEA, 25. 28042 MADRID
TF. 742 20 19 - 274 53 80

Recorta o copia este cupón y envíalo a:
MICRO-1, C/ Duque de Sesto, 50. 28009 MADRID. Tf.: 275 96 16.

NOMBRE _____
APRILIDOS _____
CALLE _____
C. POSTAL _____
CANTIDAD _____
DESCRIPCIÓN _____
PTAS. _____
PROVINCIA _____

¡SIN GASTOS
DE ENVÍO!

MACROENSAMBLADOR M80

Examinaremos en este artículo algunas de las características y posibilidades del macroensamblador M80 de Microsoft y mostraremos cómo podemos construir con esta herramienta un archivo '.BIN' con código máquina cargable desde el BASIC.

M

80 es un macroensamblador (es decir, un ensamblador que admite definición y uso de macros). Se ejecuta en CPM.

El macroensamblador M80 de Microsoft viene acompañado de un editor de enlaces y cargador (L80), y utilidades para obtener referencias cruzadas (CREF80) y gestionar librerías de programas objeto (LIB80).

El ensamblador ocupa 14 Kb de memoria y ensambla a una tasa de unas 1.000 líneas por minuto. Es un ensamblador de 2 pasadas. Admite mnemónicos del 8080 y del Z80 y contiene un conjunto amplio de directivas o pseudo operaciones, que satisfarán al programador profesional.

Podemos programar modularmente, construir rutinas y librerías de programas objeto a partir de fuentes independientes y enlazarlas con L80 para construir el módulo ejecutable.

Disponemos asimismo de un juego completo de directivas para definición de macros e inclusión de librerías de macros en el programa fuente.

Combinado con las posibilidades de L80 y el resto de utilidades, el macroensamblador M80 es una herramienta para la programación en ensamblador.

No daremos aquí una descripción completa de M80, para lo que remitimos al manual de Microsoft.

Construcción de un programa

Antes de pasar a examinar las posibilidades del ensamblador, veamos con un ejemplo elemental el proceso de construcción del programa ejecutable:

1) **Creamos el fuente con un editor de textos.** El nombre del archivo fuente deberá tener obligatoriamente la extensión '.MAC'. Por ejemplo:

```
COMIENZO DEL FUENTE
Prueba del M80
.Z80
Definición de macros:
VISUALIZA MACRO TEX
LD DE, TEX
LD C,9 ;; código de
función
CALL 5; BDO$
ENDM
MACRO
JP O
ENDM
FIN
Programa:
TEXT01: DB 'Prueba 1 del M80$'
COMIENZO:
VISUALIZA TEXTO1
FIN
END COMIENZO
FIN DEL FUENTE
```

Supongamos que llamamos al archivo fuente `PROGR1.MAC`.

2) **Ensamblamos el programa fuente:**

`A + M80 = PROGR1/R/L`

El ensamblado produce un fichero con la extensión '.REL' que contiene el objeto y otro con la extensión '.PRN' que contiene el listado del ensamblado.

3) **A continuación, si no ha habido errores en el ensamblado, creamos el objeto cargable con L80:**

`A + L80 PROGR1,PROGR1/N/E`

L80 crea un archivo '.CON' que podemos ejecutar:

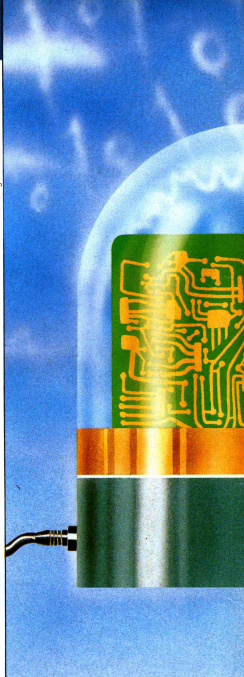
4) **Ejecución del programa**

`A + PROGR1`

Y nos aparecerá en pantalla el texto 'Prueba 1 del M80'.

Al principio del programa colocamos la directiva '.Z80' para indicar al ensamblador que vamos a usar los mnemónicos del Z80. La máquina asume por defecto '.8080'. A continuación definimos dos macros que vamos a usar en el programa: `VISUALIZA` con 1 parámetro y `FIN` sin parámetros.

James Iguad

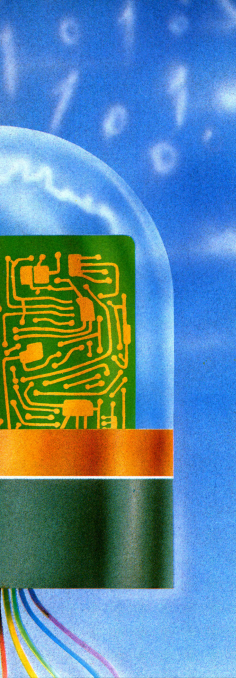


Directivas o Pseudo operaciones

El M80 soporta las directivas del macroensamblador reubicable de INTEL para el 8080. Además disponemos de otras directivas (.Z80, .PRINTX, .REQUEST, etc.).

En el apéndice damos una tabla con las directivas que admite el M80.

Disponemos de directivas para control del contador de posiciones y para indicar si un segmento es absoluto y reubicable (ASEG, CSEG, DSEG,



cipio y fin de comentario, en el estilo del Pascal o C. Y .PRINTX nos permite la visualización en pantalla de un comentario durante el ensamblado.

Las directivas de ensamblado condicional (IF, IFE, IFDEF, IFNDEF, IF1 e IF2) nos permiten preguntar si una expresión es diferente de 0 o es 0, si un símbolo está definido o no y si estamos en la pasada 1 a la 2 del ensamblador. Podemos escribir, por ejemplo:

```
.XLIST          no en listado
IF1            si pasada 1
.PRINTX       «Pasada 1»
ELSE
.PRINTX       «Pasada 2»
ENDIF
.LIST         prosigue con el listado
```

Macros

Si utilizamos el M80, iremos creando una librería de macros y configurando de esta forma un lenguaje de programación a nuestro gusto y medida.

Disponemos de un conjunto de directivas (REPT, IRP, IRPC, MACRO, EXITM, ENDM, LOCAL) que permiten declarar la repetición de bloques de instrucciones y definir macros:

REPT exp indica que se repita el bloque de instrucciones que hay a continuación hasta la directiva ENDM tantas veces como el valor de exp; IRP e IRPC indican que se repita el bloque a continuación dando en cada una de las repeticiones del bloque un valor diferente a un parámetro. El formato es IRP param (lista de valores) e IRPC param, caracteres. En cada una de las repeticiones param toma como valor sucesivos elementos de la lista de valores o caracteres de la cadena en el caso de IRPC.

Con la directiva MACRO definimos una macro-instrucción. Irá precedida del nombre de la macro y seguida de la lista de nombres de parámetros. La definición termina con END;. La directiva EXITM permite terminar la expansión de una macro antes del fin del bloque de definición. Podemos colocar etiquetas dentro de la definición de una macro y declararlas locales para cada expansión de la macro con la directiva LOCAL.

Podemos contener parámetros con texto en el interior de la Macro utilizando el operador '&' y colocar comentarios en la definición que no aparecen en las expansiones con ';;'.

Disponemos también de directivas para ensamblado condicional según esté presente algún parámetro o no en cada expansión de la macro: con IFNB (param) preguntamos si algún parámetro está presente, y con IFB (param) o IF NUL param preguntamos si está ausente.

Para controlar la impresión en el listado del ensamblado de la expansión de las macros dis-

Help CP/M

ponemos de .LALL para listar toda la expansión, .SALL para no listar la expansión y .XALL (asumido por defecto) para listar sólo las líneas de la expansión que produzcan código objeto.

El procesador de macros incorporado en M80 admite dentro de las macros llamadas a otras macros e incluso a sí misma (recursividad) sin límite práctico en cuanto a número de anidamientos, así como definiciones de otras macros e incluso de sí misma dentro de la definición de una macro. (Una macro se puede definir más de una vez, se toma siempre la última definición. Si una macro está definida dentro de otra, la definición sólo tendrá efecto cuando se expanda la macro que la contiene.)

En el programa ejemplo que pusimos al principio vemos la definición y uso de 2 macros sencillas. A continuación veremos ejemplos de concatenación de texto con parámetros, de definición recursiva de una macro, de la expansión condicional según la presencia o no de un parámetro y de la definición de macros dentro de macros.

Creemos una macro ROTC L/R,n que rote A+Carry a la izquierda (si L) o a la derecha (si R) tantos bits como el valor de n. ROTC L,3 deberá generar 3 instrucciones RLA y ROTC R,2 generará 2 instrucciones RRA:

```
ROTC MACRO DIR,N
REPT N          ;; repite n veces
R&DIR&A       ;; RLA o RRA
ENDM           ;; fin del bloque de
               ;; repetición
ENDM           ;; fin de la definición de
               ;; la macro
```

Los comentarios delimitados con ';;' no aparecen en la expansión de la macro. Vemos el uso de '&' para concatenar parámetros y texto. Podemos no utilizar REPT y usar en cambio la definición recursiva:

```
ROTC MACRO DIR, N
IF N
R&DIR&A
ROTC DIR,N-1
ENDIF
ENDM
```

Si queremos que el segundo parámetro sea opcional, y que si no está presente se asuma con el valor '1' escribiríamos:

```

ROTC MACRO      ROTC MACRO
DIR,N           IFB (N)
R&DIR&A        R&DIR&A
EXITM          ELSE
ENDIF          o
                bien: REPT N
REPT N         R&DIR&A
R&DIR&A       ENDM
ENDM          ENDFIF
ENDM          ENDM

```

Podemos crear una macro cuya expansión sea la definición de otra.
Por ejemplo en lugar de escribir:

```

ERROR1 MACRO
LD A,1
JMP RUTERROR
ENDM

ERROR2 MACRO
LD A,2
JMP RUTERROR
ENDM

```

etc., podríamos poner:

```

DEFERR MACRO ERR
ERROR&ERR MACRO ;; el cuerpo de la macro
es la definición de ERRORn
LD A,ERR
JMP RUTERROR
ENDM
ENDM
DEFERR 1 ; definición de la macro ERROR1
DEFERR 2 ; definición de la macro ERROR2
etc.

```

Esto nos puede servir para también definir una macro cuya primera llamada expanda la llamada a una subrutina y el código de la subrutina y en las sucesivas llamadas solo se expanda la llamada a la subrutina:

```

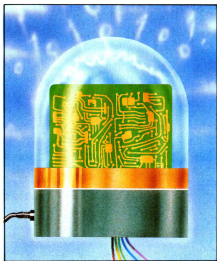
SUBRUT MACRO
SUBRUT MACRO
CALL SUBR
ENDM
CALL SUBR
LOCAL SSSS
JMP SSSS

SUBR
...
...
RET

SSSS
ENDM

```

Si declaramos la macro SUBRUT la primera vez que se la llame se volverá a definir SUBRUT con sólo la llamada a la subrutina y se expandirá el código que hay a continuación de esta redefinición hasta ENDM. Las siguientes veces que se llame a SUBRUT se utilizará la redefinición. De las 2 etiquetas en el cuerpo de la macro una es local (SSSS) y la otra global (SUBR).



Construcción de un '.BIN' con M80

Podemos usar el ensamblador para crear programas CPM o para crear archivos '.BIN' que contengan código máquina y RSX cargables desde el Basic.

Los archivos binarios cargables desde el Basic contienen una cabecera de 128 bytes donde figura la dirección de carga (para cuando no se especifique en la instrucción LOAD de BASIC), el número de bytes a cargar y un valor de control de los caracteres de la cabecera.

Mostraremos aquí dos macros que incorporadas al principio y al final de nuestro código crearán automáticamente la cabecera. Con estas macros escribiremos:

```

INCLUDE BINMAC.MAC
CABIN nnnnn ;establece cabecera
                del '.BIN' e indica
                dirección de
                carga
;
;
; A continuación escribimos el código que se
cargará con LOAD
COMIENZO ;Esta etiqueta tiene
                el valor nnnnn
.....
.....
;Después de terminar con el código que se
cargará, pondremos:
FINBIN ;calcula longitud
                del código máqui-
                na y coloca final
                del '.BIN'
END

```

Si creamos el fuente así o lo llamamos por ejemplo RSX1.MAC lo ensamblamos y creamos el '.BIN' con:

```

A → M80 =RSX1/R/L
A → L80 =RSX1,RSX1,.BIN/NE

```

A continuación podremos cargar desde el Basic el archivo. Con LOAD "RSX1.BIN" se

cargará automáticamente en la dirección que establecimos al principio de nuestro fuente (parámetro de CABIN). Si el contenido a cargar desde Basic es reubicable pondremos cero como dirección de carga.

El contenido de BINMAC.MAC será el siguiente:

```

BINMAC.MAC -Macro para la creación de un '.BIN' cargable desde Basic con M80
CABIN
N MACRO DCARGA
ASZG ;se posición 0 del archivo es la dirección 100h
CRG 12h ;el número del archivo es el número-
        de control de carga
LD 2,0 ;longitud de carga
KEY SET 2=HIGH DCARGA+LOW DCARGA ;calcula suma de control
DE ;
DW XHLNG ;longitud a cargar
CRG ;
DW XHLNG ;longitud a cargar
DB ;
KEY SET 85+2*LOW XHLNG+HIGH XHLNG ;calcula suma control
DB ;
DW XRS ;suma de control de la subrutina
CRG 15h ;suma de control de la subrutina
XCMD ;para calcular longitud del código
M80 ;para calcular longitud del código
        para que M80 calcule direcciones bin
        ENDM
FINBIN MACRO DEPHASE ;vuelve a modo normal
        para calcular longitud del código
        DEPHASE ;calcula longitud
        DEPHASE ;coloca en el archivo '.BIN' al final
        ENDM
FIN ;
FIN BINMAC.MAC

```

Apéndice - Tabla de directivas

A continuación enumeramos las pseudo-operaciones que admite el M80.	
- Control del módulo:	
NAME nombre	Da un nombre al módulo
END COMIENZO	Fin del ensamblado (Dirección de comienzo opcional)
- Control de 'locación cuater' o control de parámetros:	
ALOC	Declara segmento albinado los parámetros
CSEG	Declara segmento reubicable de datos
DCSEG	Declara segmento reubicable de control
CRG esp	Establece valor para el
PHASE esp	A efectos de cálculo de valor de los símbolos constantes que entran en la dirección esp, pero el código obtiene el valor a continuación del comentario
DEPHASE	Vuelve a ensamblado normal después de PHASE
- Definición de datos:	
DB ...	Define bytes
DC ...	Define cadenas de caracteres
DD ...	Define espacios de almacenamiento
DW ...	Define palabras
- Definición de símbolos:	
EQ ...	Define símbolo con valor fijo
IF ...	Define símbolo con valor variable
IF ...	Define con otro módulo
ENTER & PUBLIC	Define un símbolo
ORIGIN	Declara punto de entrada o entrada 'pública'
ORIGIN & EXT nombre	Declara el símbolo como externo
INCLUDE archivo	Incluye en el fuente el contenido del archivo indicado
REQUEST archivo	Al hacer las referencias no resuelve en el librería o programa objeto indicado
TITLE	Establece un título para los páginas del listado
SUBTITLE	Subtítulo para los páginas del listado
PAGE	Procede en la siguiente página
LIST	Esta línea y la que sigue se listan
JUST	Esta línea y la que sigue no se listan
CRF	Activa salida a referencias cruzadas
ICREF	Desactiva salida a referencias cruzadas. Estos dos directivas sólo son efectivos cuando se ensambla con el opción CR
- Comentarios:	
COMMENT	Para colocar comentarios de varias líneas
PRINT	Para visualizar un comentario en pantalla durante el ensamblado
- Formato del fuente:	
INDENT	Para indicar la línea en la que por defecto están los símbolos lo que sugiere referencias de EORE (se asume por defecto 20)
INDENT 20	Para indicar la línea en la que por defecto están los símbolos lo que sugiere sus referencias de EORE
- Ensamblado condicional:	
IF & SET esp	Si la expresión no es cero o ensamblamos las líneas que siguen
IF & EE esp	Se ensambla la expresión si es cero
IF 1	Si es la pasada 1
IF 2	Si es la pasada 2
IFDEF símbolo	Si está definido (tiene un valor) el símbolo
IFDEF símbolo	Si no está definido el símbolo
IFNDEF símbolo	Opcional para cada IF
ENDIF	Obligatorio al final del if
- MACROS	
- Control del listado de las expansiones de las macros:	
LALL	Lista toda la expansión
LALL	No lista la expansión
CALL	Lista sólo las líneas de la expansión que producen código
*Para la expansión condicional según la presencia o no de un parámetro:	
IF ... param	Si no existe el parámetro
IFN ... param	Si existe el parámetro
IF NOT param	Si no existe el parámetro
- Para la repetición de bloques de instrucciones:	
REP rep	Repite el bloque rep veces
REP rep,cond	Repite el bloque tantas veces como elementos en lista
IFPC param,cond	Repite el bloque tantas veces como caracteres en cadena
IFPC param,cond	Repetición de fe de bloque
IF (R) (R) en cada una de las expansiones de la suma valor a param cada una de los elementos de la lista o cadena.	
- Definición de Macros:	
MACRO	Define una macro
ENDM	Fin del bloque de definición a la macro
ENDM	Fin del bloque de definición
LOCAL var	Declara una etiqueta local en cada expansión de la macro

*Con estos tres programas
LO VERA TODO MAS CLARO*

Contabilidad

P.V.P. 19.900*

Facturación

P.V.P. 15.500*

Control de stock

P.V.P. 14.900*

PCW 8256

PCW 8512

ORDEMANIA
SOFT

te sorprenderá

* Estos precios no incluyen el IVA

Torres Quevedo, 34
Tel. (967) 22 79 44
Codigo Postal 02005
Albacete

DISPANT

De estructurado y bien ejecutado podríamos definir este diseñador gráfico de pantallas, que cuenta con un amplio número de posibilidades, gracias a las cuales haremos sin duda excelentes pantallas que podremos utilizar en otros programas.

Luis Javier Iáñez Pareja



Este diseñador gráfico que os envío no pretende en absoluto superar a los precedentes, aunque he de reconocer que está en gran parte basado en ellos.

Mi diseñador gráfico, muy guiado por «Screen Designer», pone en manos del usuario la posibilidad de **manejar simultáneamente y a placer dos pantallas gráficas**, (creo que el «Screen Designer» posee ambas pantallas, pero en cualquier caso no las pone a entera disposición del usuario). Una de ellas permanece en la memoria del ordenador sin ser exhibida en pantalla hasta que ello no sea requerido por el mismo usuario, y sin que ello suponga la pérdida de la que en ese momento estuviese en el monitor.

Es el uso de ambos tipos de pantallas (que yo personalmente he llamado gráfica y de memoria) por el mismo usuario, lo que diferencia a mi diseñador gráfico del resto de los diseñadores, o por lo menos de los que conozco.

Existen una serie de sencillas rutinas en máquina que hacen que el uso, trasiego y mezcla de imágenes y dibujos se realice por el usuario de una forma rápida y cómoda.

El programa incluye unas detalladas instrucciones que se ilustran nada más comenzar su ejecución. La inclusión de dichas instrucciones, que alargan notablemente la longitud del programa, se debe a que si bien el uso del programa no es en lo más mínimo complicado, sí que la memorización de todos los detalles del mismo puede resultararlo.

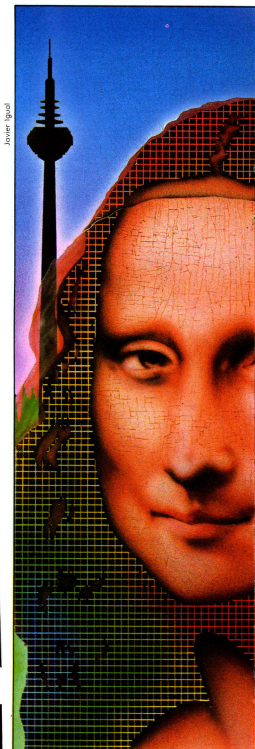
TABLA DE SUBRUTINAS EN CODIGO MAQUINA

MEZCLA	Superpone la pantalla gráfica a la de memoria.
DESDEMEM	Lleva la pantalla de la memoria a la pantalla gráfica.
AMEM	Lleva la pantalla gráfica a la memoria.
ARRIBA, ABAJO, DERECHA, IZQUIERDA	Mueven el contenido de la pantalla en la dirección indicada.
AUMENTO	Aumenta la mitad inferior de la pantalla.
AUMENTO 2	Aumenta la mitad izquierda de la pantalla a lo ancho.
CORRE, CORRE 1, CORRE 2	Realizan una serie de correcciones de los efectos secundarios de alguna de las anteriores subrutinas.
40049, 40119	Direcciones de llamada a las subrutinas de efectos especiales.

Nota: Desdemem, amem, arriba, abajo... son constantes que vienen definidas en las primeras líneas del programa. También pS (desde 1 hasta 3) son constantes que se utiliza en las instrucciones.

TABLA DE SUBRUTINAS EN BASIC

700-730	Carga la pantalla gráfica actual en disco.
740-770	Guarda la pantalla gráfica actual en disco.
780-870	Cambia el color a un lápiz.
880-910	Cambia el color de fondo.
920-960	Modo texto.
1000-1020	Inicializa los colores.
1060-1080	Carga o pokeado del código máquina. También comprueba si ha habido errores en la copia de los DATAS.
1120-1810	Instrucciones.



Javier Iguaz

Como habréis podido comprobar, el programa en sí no tiene mucha complicación; es un programa sencillo, sin que ello impida que posea una amplia gama de posibilidades que satisfarán al interesado en un diseñador gráfico capaz de alternar gráficos, tratándolos por separado y mezclándolos cuando crea necesario, y que también hará las delicias del que tan sólo busque un rato de diversión viendo cómo sus diseños cambian, se mueven, se entrecruzan, aumentan,...

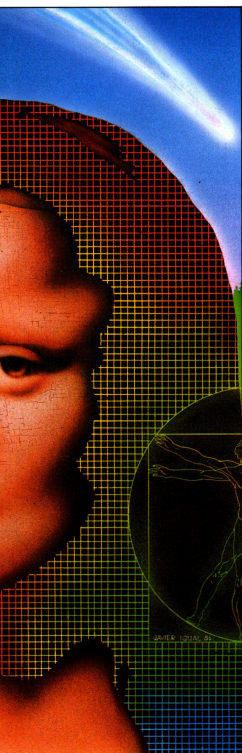


TABLA DE VARIABLES

X, Y	Abcisa y ordenada del cursor.
DX, DY	Incremento de la abcisa y de la ordenada del cursor
AX, AY	Anteriores abcisa y ordenada del cursor.
RX, RY	Abcisa y ordenada del origen de una recta o del centro de una circunferencia.
RA	Radio de circunferencia.
CIR	Angulo que se usa en la realización de una circunferencia.
A	Lápiz en uso.
C(0-15)	Colores de cada uno de los lápices.
LA, CO, SC	Respectivamente el lápiz, color y segundo color que se emplean como intermediarios en el cambio del color de un lápiz.
CF	Color de fondo.
PF	Color apropiado para el lápiz con el objeto de que se distinga del color del fondo cuando hay que ilustrar texto.
E	Variable de espera (pérdida de tiempo).
A	Nombre del fichero o pantalla que se quiere guardar o cargar. Interviene además en el modo texto.

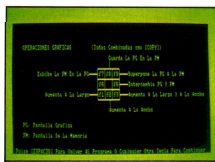
del **estado de variables** (qué color está en uso, el color del lápiz...) que el color del borde, que coincide con el del lápiz que está en uso, pero antes de desaprovechar el espacio de pantalla que los **Amstrads** nos brindan (error en el que incurre «**Screen Designer**») preferí hacer esto.

El programa es compatible con todos los BASICs de **Amstrad**, por lo que podrán hacer uso del mismo todos sus usuarios, excepto los que posean el PCW 256, pero los que no dispongan de la instrucción FILL deberán borrar la línea 460.

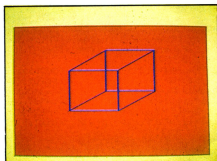
No he incluido el texto explicativo sobre el manejo del programa en este texto porque esta ya sobreabundantemente explicado en el mismo.

Quizás hayáis notado la ausencia del modo 2 y del modo 1, pero consideré contraproducente su inclusión, porque alargaría mucho el programa (sobre todo el código máquina) y porque nunca pensé en que mi diseñador gráfico se destinase a gráficas matemáticas, científicas, o de estadísticas, sino que fuera para aprovechar al cien por cien las posibilidades coloristas de los **Amstrad**.

Otro posible detalle que puede echarse en falta es el hecho de que el usuario no tenga más información



Serie ORO



EXPLICACION DEL RESTO DEL PROGRAMA

- 10 Hace que el trazado de circunferencias se realice en grados. Además dimensiona la matriz de los colores de cada lápiz.
- 20-60 Define las constantes, inicializa las variables y llama a las rutinas desde la 1000 hasta la 1810, entre las que están las instrucciones.
- 90 Evita efectos secundarios del movimiento del cursor cuando lo que busca el usuario es la llamada a subrutinas en máquina.
- 130-310 Mueve el cursor, selecciona su velocidad y hace que el cursor deje o no rastro.
- 350-490 Cambia el lápiz, hace intermitente el cursor para su localización, fija el centro de circunferencia o el origen de una recta, traza circunferencia o recta, borra la pantalla, rellena recintos con la tinta del lápiz en uso y llama a las subrutinas en BASIC.
- 530-650 Llama a las subrutinas en código máquina.
- 660 Cierra el ciclo principal del programa.
- 1850 en adelante Datos correspondientes a los colores (las dieciséis primeras) y al código máquina (las siguientes).

```

10 DEG:DIH c(15)
20 p$(1)=" $
   f7$F8$F9
   f4$ f6$
   f1$F2$F3
   $ " "
30 p$(2)=" $
   f7$F8$F9
   f4$ f6$ f6$
   f1$F2$F3
   $ " "
40 p$(3)="Pulse [ESPACIO] Para Volt
   er Al Programa O Cualquiera Otra Tec
   la Para Continuar"!!linea=1860
50 GOSUB 1000
60 mezcla=40021:desdenem=40000:anem
   =40012:arriba=400B3:abajo=401071:izq
   uierda=40071:derecha=40095:aumento=
   40193:aumento2=40429:cambio=40360:c
   orre=40558:corre1=40569:corre2=4002
   4:ax=1*x=320:y=200:pf=24:rx=321:ry=2
   01
70 MODE 0:INK 0,0:CLS
80 BORDER c(a)
90 IF INKEY(9)>= THEN SOUND 530
100 '
110 ' ===== MOVIMIENTO DEL CUR
   SOR =====
120 '
130 IF INKEY(20)<>-1 THEN dx=-4
140 IF INKEY(10)<>-1 THEN dx=-4:dy=
   2
150 IF INKEY(3)<>-1 THEN dx=4:dy=2
160 IF INKEY(14)<>-1 THEN dy=2
170 IF INKEY(4)<>-1 THEN dx=4
180 IF INKEY(13)<>-1 THEN dx=-4:dy=
   -2
190 IF INKEY(5)<>-1 THEN dx=4:dy=-2
200 IF INKEY(11)<>-1 THEN dy=2
210 IF INKEY(12)>= THEN dx:=45:dy:=
   dy*5
220 x:=x+dx:y:=y+dy:dx:=0:dy:=0
230 IF x>=640 THEN x:=600:Y 350
240 IF x<0 THEN x:=640
250 IF y>=400 THEN y:=400:GOTO 320
260 IF y<0 THEN y:=400
270 IF INKEY(21)<>-1 THEN x:=(x+ax)/
   21:y:=(y+ay)/2
280 PLOT x,y,a
290 IF INKEY(23)=1 AND (x<>ax OR y
   <>ay) THEN PLOT ax,ay,0
300 ax:=x
310 ay:=y
320 '
330 ' ===== LAPIZ,PAPEL, BORDE,
   RECTAS, CURVAS, CARGAR, GUARDAR, ... ==
   =====
340 '
350 IF INKEY(36)=0 THEN ana:=1: SOUND
   1,20,10:IF a>15 THEN ana=1
360 IF INKEY(68)<>-1 THEN :PLOT x,y
   0:CALL anem:GOSUB 1120
370 IF INKEY(7)<>-1 THEN PLOT x,y,0
   1:PLOT x,y,a:GOTO 370
380 IF INKEY(59)>= THEN DRAW rx,ry,
   a
390 IF INKEY(34)=0 THEN SOUND 1,20:
   p$(a-rx)*2*(y-ry)*2*0.5:FOR cir=
   0 TO 360 STEP 110:ra:=PLOT rx+cos(cir
   )*ra,ry+sin(cir)*ra,a:NEXT
400 IF INKEY(67)>= THEN SOUND 1,20:
   rx:=x:ry:=y
410 IF INKEY(36)=0 AND c(a)>50 THEN
   BORDER INT(c(a)/100),c(a)-INT(c(a)
   /100)*100:FOR e=1 TO 100:NEXT
420 IF INKEY(36)=0 AND c(a)<50 THEN
   BORDER c(a):FOR e=1 TO 100:NEXT
430 IF INKEY(36)=128 THEN SOUND 1,2
   0:GOSUB 780
440 IF INKEY(62)=0 THEN SOUND 1,20:
   GOSUB 700
450 IF INKEY(51)=0 THEN SOUND 1,20:
   GOSUB 920
460 IF INKEY(50)=0 THEN SOUND 1,20:
   PLOT x,y,0:MOVE x,y,a:FILL a
470 IF INKEY(27)>= THEN SOUND 1,20:
   GOSUB 880
480 IF INKEY(52)=0 THEN SOUND 1,20:
   GOSUB 740

```

BB

```

490 IF INKEY(16)=0 THEN SOUND 1,20:
   CLS
500 '
510 ' ===== LLAMADAS A SUBRUTI
   NAS EN MAQUINA =====
520 '
530 IF INKEY(10)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:CALL desdenem
540 IF INKEY(11)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:CALL anem
550 IF INKEY(4)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:PLOT x,y,0:CALL cam
   bio
560 IF INKEY(13)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:PLOT x,y,0:CALL co
   rre2:CALL aumento:CALL cambio
570 IF INKEY(14)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:PLOT x,y,0:CALL au
   mento:CALL cambio
580 IF INKEY(5)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:PLOT x,y,0:CALL au
   mento:CALL cambio:CALL corre2:CALL
   aumento:CALL cambio
590 IF INKEY(3)<>-1 AND INKEY(9)=0
   THEN SOUND 1,20:PLOT x,y,0:CALL mez
   cla
600 IF INKEY(38)<>-1 THEN CALL 4004
   9
610 IF INKEY(46)<>-1 THEN CALL 4011
   9
620 IF INKEY(35)<>-1 THEN x:=B:CAL
   L izquierda
630 IF INKEY(69)<>-1 THEN y:=y+16:CA
   LL arriba:CALL corre1
640 IF INKEY(61)<>-1 THEN x:=B:CAL
   L derecha
650 IF INKEY(54)<>-1 THEN y:=y-16:CA
   LL abajo:CALL corre
660 GOTO 90
670 '
680 ' ===== SUBRUTINAS EN BASI
   C =====
690 '
700 INK 1,pf:WHILE INKEY<>"":WEND:
   MODE 1:CLS:LOCATE 5,13:PRINT "Due p
   antalla Oultere Cargar ?":LOCATE 5,1
   3:INPUT "",a$
710 MODE 0:LOAD a$,49152
720 IF c(1)<50 THEN INK 1,c(1) ELSE
   INK 1,INT(c(1)/100),c(1)-INT(c(1)/
   100)*100
730 RETURN
740 CALL anem:INK 1,pf:WHILE INKEY$
   <>"":WEND:MODE 1:LOCATE 5,13:PRINT
   "Due Pantalla Oultere Guardar ?":LOC
   ATE 5,15:INPUT "",a$
750 MODE 0:CALL desdenem
760 SAVE a$,b,49152,16384
770 RETURN
780 CALL anem:INK 1,pf:WHILE INKEY$
   <>"":WEND:MODE 1:CLS:LOCATE 5,5:INP
   UT "Due lapiz quiere cambiar ? " :
   a:IF la<1 OR la>15 THEN 780
790 LOCATE 5,7:PRINT "Parpadeo (s/n)
   " :a$=INKEY$:IF UPFR$(a$)<>"S" AND
   UPFR$(a$)<>"N" THEN 790
800 IF UPFR$(a$)="S" THEN 850
810 LOCATE 5,9:INPUT "Cual es el nu
   evo color ? ",co:IF co<0 OR co>26 T
   HEN 810
820 INK 1,a$,co:c(1a)=co
830 IF c(1)<50 THEN INK 1,c(1) ELSE
   INK 1,INT(c(1)/100),c(1)-INT(c(1)/
   100)*100
840 MODE 0:CALL desdenem:RETURN
850 LOCATE 5,9:INPUT "Que dos nuevo
   s colores ? ",co,co2:IF co=co OR co<
   0 OR co2<0 OR co>26 OR co2>26 THEN 85
   0
860 INK 1,a$,co,co2:c(1a)=co*100+co2
870 GOTO 840
880 CALL anem:INK 1,pf:WHILE INKEY$
   <>"":WEND:MODE 1:CLS:LOCATE 5,13:IN
   PUT "Due color pongo de fondo ? " :c
   1:IF c<0 OR c>26 THEN 880
890 INK 0,c:MODE 0:CALL desdenem
900 IF c<20 THEN pf=0 ELSE pf=24
910 RETURN
920 WHILE INKEY$<>"":WEND:TAG
930 MODE x,y
940 a$=INKEY$
950 IF INKEY(18)=0 THEN TAGOFF:RETR
   RN
960 PRINT a$:GOTO 940
970 '
980 ' ===== INICIALIZACION DE
   LOS COLORES =====
990 '
1000 FOR x=0 TO 15
1010 READ a$(x):a
1020 NEXT
1030 '
1040 ' ===== CARGA DE SUBRUTINA
   S EN MAQUINA =====
1050 '
1060 FOR x=40000 TO 40579

```

```

1070 READ a$=VAL("8"+a$):IF LEN(a$)=2 THEN POKE x,asuma+suma+a ELSE
IF suma=a THEN suma=0:PRINT"linea
"linea:" correcta":linea=linea+10:
a$=-1 ELSE PRINT"linea "linea":
en correcta":END
1080 NEXT
1090 *
1100 * ===== INSTRUCCIONES =====
=====
1110 *
1120 MODE 2:WHILE INKEY<>"":WEND:1
NK 1,24:INK 0,0
1130 WINDOW #1,32,47,8,17
1140 LOCATE 5,5:PRINT"MOVIMIENTO DE
L CURSOR"
1150 PRINT#,p$(1)
1160 LOCATE 40,7:PRINT"Arriba"
1170 LOCATE 48,10:PRINT"Arriba y De
recha"
1180 LOCATE 48,12:PRINT"Derrecha"
1190 LOCATE 48,14:PRINT"Abajo y Der
recha"
1200 LOCATE 40,17:PRINT"Abajo"
1210 LOCATE 10,10:PRINT" Arriba e
Izquierda"
1220 LOCATE 10,12:PRINT"
Arriba e Izquierda"
1230 LOCATE 10,14:PRINT" Abajo
e Izquierda"
1240 LOCATE 5,19:PRINT"Combinado co
n [SHIFT]: dos veces mas lento
1250 LOCATE 5,21:PRINT"Combinado co
n [F5]: cinco veces mas rapido
1260 LOCATE 5,23:PRINT"Combinado co
n [CONTROL]: deja rastro"
1270 LOCATE 2,25:PRINT p$(3)
1280 IF INKEY="" THEN 1290
1290 IF INKEY(47)=0 THEN 1290
1300 WHILE INKEY<>"":WEND:MODE 2
1310 LOCATE 5,5:PRINT"OPERACIONES B
RAFICAS (Todas Combinadas co
n [CDEFY])"
1320 WINDOW #1,32,47,8,17:PRINT#,p
$(2)
1330 LOCATE 40,7:PRINT"Guarda La PG
En La PM"
1340 LOCATE 48,10:PRINT"Superpone L
a PG A La PM"
1350 LOCATE 48,12:PRINT"Intercambia
PG Y PM"
1360 LOCATE 48,14:PRINT"Aumenta A L
o Largo Y A Lo Ancho"
1370 LOCATE 40,17:PRINT"Aumenta A L
o Ancho"
1380 LOCATE 10,10:PRINT" Exhibe La
PM En La PG"
1390 LOCATE 10,14:PRINT" Aumenta
o Largo"
1400 LOCATE 5,20:PRINT"PG: Pantalla
Grafica"
1410 LOCATE 5,22:PRINT"PM: Pantalla
De La Memoria"
1420 LOCATE 2,25:PRINT p$(3)
1430 IF INKEY="" THEN 1430
1440 IF INKEY(47)=0 THEN 1490
1450 WHILE INKEY<>"":WEND:MODE 2
1460 LOCATE 5,2:PRINT"OPERACIONES B
RAFICAS (Continuacion)"
1470 LOCATE 10,4:PRINT"[F3].....
.....Selecciona El
Color Del Papel"
1480 LOCATE 10,6:PRINT"[L]+[CONTROL
].....Selecciona El
Color Del Lapiz"
1490 LOCATE 10,8:PRINT"[L].....
.....Sele
cciona El Lapiz"
1500 LOCATE 10,10:PRINT"[A].....
.....Sube Toda La Im
agen Ocho Pixels"
1510 LOCATE 10,12:PRINT"[D].....
.....Baja Toda La Im
agen Ocho Pixels"
1520 LOCATE 10,14:PRINT"[D].....
.....Corre Toda La Imagen Dos Pix
eles A Derecha"
1530 LOCATE 10,16:PRINT"[I].....
.....Corre Toda La Imagen Dos Pixel
s A La Izquierda"
1540 LOCATE 10,18:PRINT"[R].....Re

```

```

llena Un Recinto Cerrado Por La Tin
ta Actual (FILL)
1550 LOCATE 10,20:PRINT"[C].....
.....Carga Una Pant
alla Desde Disco
1560 LOCATE 10,22:PRINT"[G].....
.....Guarda Una P
antalla En Disco
1570 LOCATE 2,25:PRINT p$(3)
1580 IF INKEY="" THEN 1580
1590 IF INKEY(47)=0 THEN RETURN
1600 WHILE INKEY<>"":WEND:MODE 2
1610 LOCATE 5,2:PRINT"OPERACIONES G
RAFICAS (Continuacion)"
1620 LOCATE 10,4:PRINT"[O].....Det
ermina El Centro De Circunferencia
O El Origen De Recta
1630 LOCATE 10,6:PRINT"[W].....
.....Traza Recta"
1640 LOCATE 10,8:PRINT"[O].....
.....Traza Circunferencia"
1650 LOCATE 10,10:PRINT"[M]+[N].....
.....Efectos Especiales"
1660 LOCATE 10,12:PRINT"[T].....
.....Intro
duce En El Modo Texto"
1670 LOCATE 10,14:PRINT"[RETURN].....
.....Vuelve Del Modo
Texto Al Modo Grafico"
1680 LOCATE 10,16:PRINT"[CLR].....
.....B
orra Pantalla Grafica"
1690 LOCATE 10,18:PRINT"[F].....
.....Hace Interactiva El Cursor
Para Su Localizacion"
1695 LOCATE 10,20:PRINT"[TAB].....
.....Recup
era Las Instrucciones"
1700 LOCATE 2,25:PRINT p$(3)
1710 IF INKEY(47)=0 THEN RETURN
1720 IF INKEY="" THEN 1700
1730 IF INKEY(47)=0 THEN RETURN
1740 WHILE INKEY<>"":WEND:MODE 2:L
OCATE 5,5:PRINT"RECOMENDACIONES"PR
1750 PRINT" Los efectos especiales
son de optimos resultados en la
aplicacion a texto y a rectas (en
general a trazos mas o menos finos)
pudiendo ser aumentado el res-
tado y repetido el proceso.
1755 PRINT" Manteniendo presionadas
varias teclas simultaneamente, las
operaciones corres- pondientes se
suceden alternativamente.(Pruebase
,por ejemplo, a mantener pulsa- do
simultaneamente [F5]+[F9]+[W])"
1760 PRINT"LOCATE 5,13:PRINT"ADVERT
ENCIAS"
1760 PRINT"El resultado d
el movimiento de imagenes no es
el esperado cuando estas rebasan
los limites de la pantalla.Recuerd
e que el aumento de imagenes a lo
largo y a lo ancho actua sobre l
a mitad derecha e inferior respecti
vamente.
1763 PRINT" Cuando se cambia el col
or a un lapiz n al papel o se carga
n o se guardan imagenes en disco,
el resultado se guarda automati
camente en la memoria y se pier
de lo que en esta hubiera.
1765 PRINT" Cuando el cursor rebasa
los limites de la pantalla aparece
por el lado opuesto.
1770 LOCATE 5,23:PRINT" No olvide q
ue las INSTRUCCIONES se recuperan co
n [TAB] !!!"
1780 IF INKEY="" THEN 1780
1790 IF c(1)<50 THEN INK 1,c(1) ELS
E INK 1,INT(c(1)/100),c(1)-INT(c(1)
/100)+1
1800 IF cf<50 THEN INK 0,cf ELSE IN
K 0,INT(cf/100),cf-INT(cf/100)+100
1810 MODE 0:CLS:CALL 40000:RETURN
1820 *
1830 * ===== COLORES Y COD160 M

```

Serie

```

AQUINA =====
1840 *
1850 DATA 0,24,20,6,26,1,2,8,10,12,
14,16,18,22,124,1126
1860 DATA 11,00,c0,21,20,4e,01,00,4
0,e,d,b0,cf,11,20,4e,21,00,c0,01,00,
40,e,d,b0,cf,21,20,4E,01,00,40,3e,00
,23,0b,3e,00,b0,20,f7,b9,c8,1b,f3,0
0,00,00,ddb
1870 DATA 00,00,00,01,00,40,21,00,c
0,3e,00,be,28,01,34,23,0b,b9,20,f5,
bb,c8,c3,77,9c,11,00,c0,21,01,c0,01
,00,40,ed,b0,c0,11,00,c0,21,50,c0,0
1,b0,3f,ed,b0,c9,117f
1880 DATA 11,00,00,21,ff,ff,01,ff,3
f,ed,bb,c9,11,00,00,21,b0,4f,01,b0,
3f,ed,bb,c9,01,00,40,21,00,c0,3e,00
,be,28,01,35,23,0b,b9,20,f5,bb,c8,c
3,bb,9c,1330
1890 DATA 01,50,00,ed,b0,52,6b,0
1,00,0b,eb,09,eb,01,51,00,ed,bb,e1,
c9,01,b0,07,09,eb,01,01,0b,09,eb,c
d,9c,c9,01,b0,07,09,cd,9c,9c,e
2,9c,ed,e2,9c,cd,e2,9c,c9,1973
1900 DATA 21,60,c4,11,20,4e,cd,f4,9
c,11,70,4e,cd,f4,9c,21,b0,c4,11,c0,
4e,cd,f4,9c,11,10,4f,cd,0,9c,21,00
,c5,11,60,4f,cd,f4,9c,11,b0,4f,cd,f
4,9c,21,50,c5,11,00,50,cd,f4,9c,11
,50,50,1b34
1910 DATA cd,f0,9c,21,a0,c5,11,a0,5
0,cd,f4,9c,11,f0,50,cd,f0,9c,21,f0,
c5,11,40,51,cd,f4,9c,11,90,51,cd,f0
,9c,21,40,cd,c11,e0,51,c0,cd,f4,9c,11
,3,0,52,cd,f0,9c,21,90,c6,11,80,52,cd,
f4,9c,1770
1920 DATA 11,60,52,cd,f0,9c,21,e0,c
6,11,20,53,cd,f4,9c,11,70,53,cd,f0,
9c,21,30,c7,11,c0,53,cd,f4,9c,11,10
,54,cd,f0,9c,21,80,c7,11,60,54,cd,f
4,9c,11,b0,54,cd,f0,9c,9,1b89
1930 DATA 00,21,00,c0,11,20,4E,01,0
0,40,7E,32,A7,9D,1A,77,3A,A7,9D,12,
23,13,0B,3E,00,B8,20,EE,B9,C8,1E,A
8,7E
1940 DATA 0E,19,06,2B,1A,EE,AA,77,0
f,b6,77,23,1A,EE,55,77,07,b6,77,23,
13,10,ED,0D,3E,00,B9,C7,EE,55,11,2B
00,19,EB,e1,1B,CD,21,20,4E,11,00,c
0,cd,c7,9d,21,20,56,11,00,c8,cd,c7,
9d,21,20,5e,11,00,1699
1950 DATA 0d,cd,c7,9d,21,20,66,11,0
0,d8,cd,c7,9d,21,20,6e,11,00,ed,c8,
cd,c7,9d,21,20,76,11,00,eb,cd,c7,9d,21
,20,7e,11,00,f0,cd,c7,9d,21,20,86,1
1,00,f8,cd,c7,9d,c9,172f
1960 DATA 1A,77,1B,2B,1A,EE,AA,b6,7
7,1B,21,1A,EE,55,b6,77,c3,62,9e,21
00,c1,11,20,4E,01,00,40,7E,00,2B
,df,ee,55,be,28,de,7E,ee,aa,be,28,d
f,23,13,0b,3e,00,b0,20,eb,b9,c8,1e
a8,7E
1970 DATA 0e,50,21,00,cf,36,00,23,10
0,FB,C9,06,50,21,80,ff,36,00,23,10,
FB,C9,787

```



Para que los datos no salgan al magullado, se recomienda utilizar el teclado. Todos los datos que ingresen en el teclado se encuentran a la disposición en un caso se manejan, satisfactorio.

CUATRO JUEGOS EN DISCO DE DESING-DESING

Una nueva casa en el mundo del software; Desing-Desing, lanza un disco con cuatro juegos: On the run; Dark star; Tank busters y Halls of the things, aprovechando de esta forma la capacidad del disco.

L

os programas están distribuidos de forma que en la cara B, se encuentra solamente Oh the run, estando los tres restantes situados en la cara A.

Para efectuar la carga de cualquier programa de la cara A, tenemos un menú de opciones, en el cual podemos elegir en primer lugar el idioma en que queremos las instrucciones, encontrándose afortunadamente el español entre ellos.

Una vez seleccionado el idioma, podemos elegir entre cargar directamente cualquiera de los juegos, o leer sus instrucciones.

ON THE RUN

Un intrépido astronauta se encuentra en apuros, el generador de su nave galáctica se ha quedado sin energía, y corre el riesgo de no poder escapar del planeta den el que ha aterrizado.

Para conseguir abastecer de energía a su generador, debe encontrar en las recónditas profundidades del planeta, las seis cápsulas energéticas.



nos deshacernos a golpe de láser.

Un programa de los típicos de búsqueda, de construcción laberíntica, en el cual el decorado por el que nos movemos es una selva con setas gigantes y extrañas flores.

Nuestro hombre deberá utilizar su propulsor de espalda para ascender en el aire y llegar a los escalones superiores del laberinto.

Los gráficos son de tamaño gigantesco, realizados en forma multicolor, consiguiendo un dibujo tanto infantil, pero efectivo.

Las instrucciones, aconsejan disparar sobre todo los que se mueva, cosa que debemos respetar fielmente, pues el contacto con cualquier elemento hostil nos resta energía.

Eliminar estos molestos bichos, por otra parte incrementa nuestro marcador, cosa que siempre es útil para superar nuestro récord.

La energía se recupera recogiendo los objetos que encontramos a nuestro paso.

Un juego de búsqueda con tiempo limitado y en el que conocer a la perfección las características del laberinto, es fundamental.

DARK STAR

El primer programa de la cara A, es una aventura galáctica. Evil Lord, el más malvado entre los tiranos de las estrellas, amenaza con hacerse con el poder en su universo.

En nuestras manos está detener su creciente poderío, o morir en el em-



punto de mira de nuestra torre de tiro preside la pantalla.

Nuestro enemigo es detectado por el radar, apareciendo un minúsculo punto verde que marca su posición, nos dirigiremos hacia él y tras centrar el punto verde en el trazo vertical de nuestra mira, observamos cómo aparece en el horizonte.

Según nos aproximamos hacia nuestro blanco, éste aumenta de tamaño, con impresionante realismo, si está en marcha observaremos cómo pasa delante de nosotros a gran velocidad, si está parado; cuidado pues girará sobre sí mismo hasta tenernos en su punto de mira y disparará.

Para aumentar el efecto tridimensional, elementos geométricos como prismas y pirámides de cristal aparecen en nuestro camino, amenizando la marcha.

Disponemos de misiles guiados y granadas antitanque.

Un programa de aniquilación total, con excelentes efectos tridimensionales, sin duda el mejor de los cuatro.

HALLS OF THE THINGS

Otro programa de laberintos, pero muy distinto al que ocupa la cara B, si en éste los gráficos eran de tamaño gigante, en Halls of the things, los gráficos son minúsculos, teniendo nuestro hombre una altura de un centímetro.

El objetivo en esta ocasión, es encontrar los siete anillos escondidos en los siete niveles más altos de la torre de la muerte.

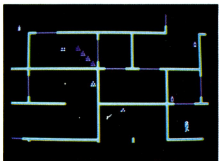
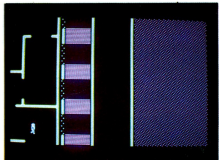
Una vez logrados debemos recoger la llave que se encuentra en las mazmorras subterráneas, y escapar con ella.

Distintos objetos y posiciones mágicas, nos ayudarán a recuperar las fuerzas perdidas en el combate con los peligrosos moradores de los corredores y salas, no olvides que todo lo que se mueve va a por ti.

Podemos ejecutar todas las acciones típicas de este tipo de juegos como: recoger objetos, soltarlos, abrir y cerrar puertas y corredores, disparar, lanzar hechizos, curar nuestras heridas, etc.

La sencillez de los gráficos, nos hace pensar en un programa de los pertenecientes a la primera generación de juegos de ordenador, anti-cuado pero entretenido.

Mister JOYSTICK



peño, antes de que destruya todo vestigio humano en la galaxia.

Pilotamos una nave intergaláctica, con la que debemos desplazarnos de planeta en planeta en busca de las bases del gran tirano.

El mapa táctico, una vez que llamamos penetrado en la atmósfera de cada planeta, nos revelará la posición de las bases y zonas de influencia que se hallan en su superficie.

Un programa de navegación espacial, con una construcción muy sencilla, en el cual el punto de mira de nuestro sistema de fuego láser, aparece constantemente en pantalla listo para aniquilar lo que encontremos a nuestro paso.

TANK BUSTERS

Un auténtico juego de tres dimensiones, basado en el mítico Battle Zone de ATARI, el juego que revolucionó el mundo de las máquinas de billares, dejando a los comeocos como una pieza de museo.

ATARI, la productora de los mejores juegos para megaordenadores, con este programa dio comienzo a la era de los simuladores de combate, consiguiendo una sensación de perspectiva y realismo tan grande, que versiones mejoradas de este programa son utilizadas para el entrenamiento de los artilleros de carros de combate.

Volviendo al juego de Desing-Desing, dirigimos nuestro tanque por un paisaje montañoso, en el cual el

En tu kiosco te espera algo muy inteligente

El especial Amstrad n.º 2. Incluye una cinta con un lenguaje LISP completo que te permitirá comprender y dominar las técnicas más complejas de la inteligencia artificial.

¡Dale inteligencia a tu Amstrad!



MICROHOBBY
AMSTRAD
Especial Año I N.º 2

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

475 ptas.

SOFTWARE INTEGRADO
3 PROGRAMAS EN UNO

TABLETA GRAFICA
GRAFPAD II:
EL ARTE POR ORDENADOR
A TU ALCANCE

TE OFRECEMOS UN
LENGUAJE DE
PROGRAMACION LISP
COMPLETO EN CINTA
DE CASSETTE

NUESTRO PROGRAMA CREADOR
DE CRUCIGRAMAS DESAFIA A TU INGENIO

IMPRESORAS:
COMO HACER LA
MEJOR ELECCION

ATENCION A NUESTRO
FABULOSO CONCURSO:
PUEDES GANAR UN
CPC-6128 CON
SOLO CARGAR LA CINTA

Rellena este cupón y envíalo a **HOBBY PRESS, S. A.** Ap. de Correos 232. Alcobendas. Madrid.

Nombre _____ Apellidos _____
Domicilio _____ C. Postal _____
Localidad _____ Provincia _____
Teléfono _____ Profesión _____ Fecha de nacimiento _____

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? Sí No
Deseo recibir el Especial de **MICROHOBBY AMSTRAD** n.º 2 al precio de 475 ptas. (IVA incluido)

FORMA DE PAGO
 Talón bancario adjunto a nombre de **HOBBY PRESS, S. A.**
 Mediante Tarjeta VISA, N.º _____ Fecha de caducidad _____
 Contra reembolso (supone 75 ptas. de gastos de envío).
Fecha y firma: _____

AD1

El único ordenador
concebido para sustituir
a la máquina de escribir.



AMSTRAD PCW 8256

UN COMPLETO EQUIPO QUE INCLUYE:

- Unidad Central (256 K RAM) • Teclado en castellano
- Unidad de disco (180 K por cara) • Pantalla de alta resolución • Impresora alta calidad (NLO)
- Programas: • Procesador de textos, sistema Operativo CP/M Plus, Mallard Basic con JET SAM para ficheros indexados, lenguaje DR LOGO.

PROGRAMAS PROFESIONALES

- Contabilidades • Almacenes • Facturación • HOJAS DE CALCULO **Multiplán**, Supercalc 2, Cracker, Plannercalc, BASES DE DATOS **DBase II**, Amfile, Flexfile, **Boriar**, LENGUAJES: Cobol, Fortran, Pascal MT +, Pilot, etc.

También disponible la versión PCW 8512, con 512 K RAM y 2º disco de 1 MBYTE incorporado.

P.V.P. 169.900,— Ptas. + IVA

SOLICITE DEMOSTRACION EN:

División informática de **GetCompu**s, División **OnLine** de GALERIAS.
Tiendas especializadas en informática y Equipos de oficina.

NOTA: El Amstrad también puede ser utilizado como "Terminal Inteligente" de grandes equipos informáticos.

¡¡ Increíble !!

AMSTRAD

ESPAÑA

GRUPO INDESCOMP

INFORMAT 86: TODA LA INFORMATICA EN BARCELONA

Día a día, la informática va cobrando más importancia en nuestras vidas. La iniciativa de las ferias como medio de acercar al usuario todo el producto del sector de una sola vez sólo puede calificarse como de buena. Si además, como en esta ocasión la organización de la feria corre a cargo de una entidad como es la Feria de Barcelona, se tiene una garantía. INFORMAT-86, con casi 150 expositores reunidos alrededor de más de ocho mil metros cuadrados, ha sido sin lugar a dudas el acontecimiento del mes para los usuarios y fanáticos de la informática.



Durante cinco días, del 13 al 17 de mayo, Barcelona se convirtió en el centro de atención del sector informático. Distribuidores, fabricantes y **vendedores** tuvieron la oportunidad de ofrecer sus productos en los pabellones de la prestigiosa Feria de Barcelona. **INFORMAT**, en la edición de este año ha venido a confirmarse como una feria de vocación verdaderamente profesional. Liberada de intrusos faltos de interés por un precio de 1.000 pesetas la entrada, precio quizá excesivamente exagerado, el desarrollo de la feria ha tenido para los que así lo han querido un carácter elitista cerrado y con gestos ocasionales de convención.



Como en ocasiones anteriores, **INFORMAT 86** ha venido a confirmar algo que ya todos sabíamos: una continua tendencia en el sector hacia una reducción considerable del precio final del producto, así como la constante aparición de equipos altamente competitivos con un precio en ocasiones inferior al de sistemas con un nivel inferior de prestaciones.



Al margen de los productos ya clásicos, compatibles **PC**, hojas de cálculo, procesadores de textos, etc., la atención de la feria, o al menos lo que verdaderamente podía ganarse, podríamos dividirlo en dos bloques básicos;

- Gráficos y ordenadores
- Inteligencia artificial
- Robótica



Ordenadores y vídeo

En diseño asistido por ordenador y procesamiento gráfico, el abarataamiento de la memoria, sigue proporcionando cada vez mayores resoluciones, con una gama de colores mucho más amplia. Sin embargo, el adelanto más significativo lo constituye la adaptación del vídeo como periférico del ordenador. Gracias a éste podemos utilizar la imagen del vídeo de dos formas totalmente distintas, conforme nuestras necesidades lo exijan. En primer lugar, podemos utilizar la **imagen analógica** como fondo, imprimiendo en primer plano mediante el ordenador; la otra posibilidad consiste en **digitalizar** una de las imágenes para después procesándola por medio del ordenador modificarla en cualquiera de sus



aspectos con un control sobre ella prácticamente absoluto.

En esta línea, la gama **MSX**, presentaba los primeros modelos de su segunda generación **MSX-2**.

Básicamente la diferencia de estos equipos con sus predecesores, está basada en una mayor resolución, así como la incorporación de un digitalizador, junto a la unidad mezcladora. El precio total del equipo rondará las 180.000 pesetas. Se ha sustituido el Z80-A por un Z80-C, que si bien constituye un considerable ahorro en el consumo de energía, no deja de ser, por desgracia, un Z80.

Excepcional fue nuestro primer contacto con el AMIGA, último mo-



delo de Commodore y que constituye una verdadera revolución por Hardware. Tres coprocesadores, liberan al 6800 de Motorola, microprocesador del Amiga, de las tareas lentas y tediosas de gestión de pantalla, incorporando incluso por Hardware movimiento de sprites, rotaciones y otras exquisiteces capaces de satisfacer al más sibarita. El direccionamiento de memoria directo alcanza la cifra de ocho megas y medio y la calidad de imagen, más que por su resolución, 640x400 pixels, asombra por su paleta de color,



Informat

EN VIVO

4096 colores. El precio de venta en España aún no se conocía, aunque en Gran Bretaña donde ha comenzado ya a distribuirse con notable éxito, el precio base ronda las 310.000 pesetas. Naturalmente podríamos extendernos páginas y páginas sobre el grafismo por ordenador, y las cosas que allí vimos, sin embargo el resto de los equipos constituían ya, por su precio, parte de lo puramente profesional y fuera del alcance del aficionado medio.

Inteligencia Artificial

Verdaderamente sorprendente era ver cómo ejecutivos medios y gentes de despacho se sentían atraídos por este tema, y soportaban con un interés desmesurado, una charla sobre el tema que sólo era ofrecida en inglés o francés. Sí, curiosamente en toda la feria no existía una persona capaz de dominar y comprender las técnicas de la IA y que conociese el castellano. Bromas a parte, la facilidad de uso de estos programas que en ocasiones disponen de un interfaz de usuario muy próximo al lenguaje natural, junto a la potencia y elegancia de los sistemas expertos, cada vez más de moda, hacían de este tipo de programas algo verdaderamente impresionante.

Tuvimos ocasión de asistir a dos demostraciones, la primera en el Stand de AMAIA systems informatiques, con sede en Bayonne (Tel.: 33 59 55 10 01). Sobre un equipo





compatible PC, tuvimos ocasión de jugar con un programa que si bien no tiene la utilidad de una hoja de cálculo, sí supone un verdadero adelanto dentro del campo de la IA, un reconocido de letras. El programa estaba realizado en LISGLOG, una versión de Prolog a la que se le han incorporado algunas de las facilidades que soporta el LISP en el tratamiento de listas, algo muy parecido al conocido Microprolog. Aparte de este lenguaje, lleva incorporado también un módulo de programación visual mucho más rápidamente que lo que permitiría el PROLOG. El programa de demostración que ofrecían, el reconocedor de letras, permitía escribir en pantalla, mediante ratón, una letra con el tamaño y forma que nos apeteciese, tras varios



segundos de espera encontrábamos en pantalla las deducciones del programa, algo así como:

Es una **C** con un 77 por 100 de posibilidades.

Es una **O** con un 3 por 100 de posibilidades.

Resto de posibilidades despreciable.

Naturalmente nosotros habíamos escrito algo parecido a una **C**.

Si el garabato escrito en la pantalla por el usuario no era parecido a ningún carácter conocido, también lo advertía con un informe de carácter no reconocido. El equipo utilizado era un compatible **PC/AT** con un **80286** como procesador y una memoria del mega. La pantalla tenía dos posiciones, una horizontal y otra vertical, que facilitaban la visualización. El software de base que puede adquirirse con el equipo, consiste en:

- LISP
- PROLOG
- LISLOG
- MOTOR DE INFERENCIA
- PROCESO DE LENGUAJE NATURAL

BLAISE (lenguaje de programación de la unidad de visualización).

El otro stand donde se hacía gala de las facilidades de la IA, era el de SPERRY, en esta ocasión sobre un equipo con un sistema operativo propio y sobre el que corría un Prolog compilado verdaderamente rápido, en comparación con los hasta ahora conocidos. Desarrollado en este

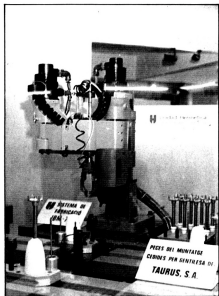


lenguaje tuvimos ocasión de comprobar la eficiencia de un generador de sistemas expertos, así como utilizar uno de estos sistemas ya generados. El sistema encontraba el avión ideal conforme a las preferencias de los pasajeros, al parecer este problema no es sencillo en los EE.UU., donde no existen tres púnicas clases de billete como es el caso de nuestro país. De este modo y tras un considerable número de preguntas que habían de ser respondidas por el operador, de forma muy cómoda, el programa en-

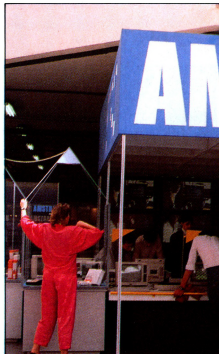
contraba el avión ideal conforme al número de billetes de que cada clase existían.

Por desgracia, este tipo de aplicaciones quedan aún lejos de nuestras posibilidades, aunque ¿por cuánto tiempo... ?

Robótica



La robótica es otra de las actividades dentro de la informática, que se presenta con alentador futuro, complicados tornos y brazos móviles movidos al unisono por un simple compatible PC. Pudimos observar también algunos, mucho menos pretenciosos, que con un carácter **educativo** y **lúdico** se ofrecían al usuario por



un precio módico. Tal era el caso del fabricado por **SVI**, sólo compatible con equipos spectravideo, y que consistía en un brazo realizado en plástico amarillo y sobre el que el usuario desde el teclado o joystick imponía un control bastante preciso y a un precio de 10.500 pesetas.

Pronto, muy pronto, cualquier ordenador doméstico será capaz de mover este tipo de artefactos y no es ciencia ficción pensar que puedan traernos las zapatillas o las gafas olvidadas como siempre sobre la mesilla.

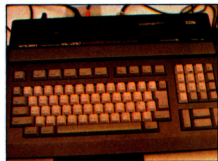
¿Y de los Home Computers qué?

Aunque la feria estaba destinada fundamentalmente al usuario profesional, habría sido una necesidad olvidar que el ordenador doméstico está siendo cada vez más utilizado para resolver pequeños problemas y permiten seguir trabajando en casa sin necesidad de tener una furgoneta para llevar el equipo de la empresa a casa y de casa a la empresa.

El líder de la feria en este apartado fue sin duda **MSX**, seguramente motivado por un momentáneo letargo de los productos **Amstrad** y que se encontraban poniéndose a punto y preparándose para la feria especializada y dedicada íntegramente a su persona y que se celebrará en Madrid durante los días 23, 24 y 25 de mayo. No obstante, tuvimos oca-

sión de utilizar unos productos nuevos de **PROTO E** especialmente diseñados para la familia **CPC** por la prestigiosa firma alemana **VORTEX**. Una ampliación de memoria de 512 K para el 464 nos hizo dudar de lo que veían nuestros ojos al ver un efecto zoom sobre un conjunto de Mandelbrot lleno de colorido y psicodelia. También presentaban una unidad de disco que convierte al **Amstrad** en un equipo verdaderamente profesional, con una capacidad de almacenamiento de un mega. La unidad se presentaba en dos versiones: **5¼** y **3½**; con idéntica capacidad para ambos formatos.

En el afortunado caso de que usted sea poseedor de un 464, (... ?) podrá conectar dos unidades, con lo que dispondrá de dos megas de memoria de almacenamiento más el medio mega de memoria central, si se decide también por la ampliación, verdaderamente impresionante.



Los productos **Sinclair** no se encontraban en la feria, lógico si consideramos la última sacudida recibida por la compra de **Amstrad** de los derechos sobre la compañía. Se rumoreaba la fabricación de un nuevo modelo, idéntico al 48 K, con teclado profesional y con la innovación de un cassette incorporado, algo muy acorde con la personalidad de los productos **Amstrad**.

Otros productos

En otro orden de cosas encontramos lo típico en estas ferias, impresoras, plotters y consumibles altamente sofisticados y con una presentación cada vez más impecable. Lejos están ya, aquellos artefactos cuadrados de esquinas puntiagudas y sin ningún esmero estético. Impresoras transparentes, discos de colores..., constituyan la afirmación de que en informática **la arruga también es bella**.



Informat

EN VIVO



Tuvimos ocasión de observar el funcionamiento de un revolucionario plotter, tanto por su precio como por su concepción. Como si de un pequeño robot se tratase y unido al ordenador por un largo cordón umbilical, **PENMAN** se desplaza sobre una hoja con increíble precisión dejando a su paso una estela de color. La compatibilidad con cualquier equipo está asegurada por su conexión **RS232**. También puede utilizarse como ratón de gran precisión o como tortuga en nuestras sesiones de **LOGO**. Es maravilloso verle buscar los bordes de la hoja cada vez que va a comenzar una nueva impresión impidiéndose a sí mismo, una vez tomadas las oportunas referencias salirse de la hoja y rayarnos la mesa. El precio, algo menos de 100.000 pesetas, aunque pudiera parecer algo elevado, no lo es si consideramos el precio medio de cualquier plotter.

Aunque oculta en un stand de difícil localización, la verdadera bomba de precios del **INFORMAT**, la constituyó un nuevo compatible **PC**, puesto en la calle por **SVI**, 256 K de memoria central, una unidad de disco de 360 K y lo que es más importante compatibilidad 100 por 100 asegurada, y todo ello, naturalmente monitor y teclado incluido, al precio de 157.000 pesetas. Una verdadera **ganga**.

En resumen **INFORMAT 86** constituye una feria bien organizada, seria e interesante para profesionales y aficionados, que de vez en cuando se animan a quitar la vista de sus monitores y a mirar con ilusión hacia adelante.



Esta semana os presentamos el conocido juego de los colores, aplicado a números.

Consiste en adivinar el código secreto formado por 4 cifras distintas. Tenéis 10 oportunidades para acertarlo.



En la pantalla aparecen sucesivamente, la cifra introducida, cuántos números de los que la componen que están incluidos en el código y los que están bien situados.

Reta a tu mente lógica y descubre el código secreto creado por tu Amstrad.

El programa realiza lo siguiente:

10-40 REMs que nos hablan del título.

50 Inicio de la definición de ventanas y funciones.

60 Se define la función, «NUMEROS(X)», como la cadena de un carácter, tomado por la derecha (RIGHTS) de la expresión literal obtenida al convertir, con el comando STR, el número «X» en variable alfanumérica.

70 Creación de la ventana #1 situada y dimensionada según los parámetros indicados.

80 REM que indica la preparación del código secreto.

90 Se asigna a la variable con subíndice

«A(1)», (elemento 1 de la matriz «A»), el valor entero del número aleatorio que nos devuelve RND multiplicado por 10. Nos dará un valor comprendido entre 0 y 9.

100 Igual pero para la variable «A(2)».

110 Se especifica que si las dos son iguales la ejecución del programa salte a la línea 100.

120 Se calcula «A(3)» igual que en casos precedentes.

130 Si esta última variable es igual a las dos anteriores la ejecución vuelve a la línea 120.

140 Se da valor a «A(4)» igual que antes.

150 Si es igual a alguna de las otras tres variables retorna a la línea 140.

160 Bucle FOR...NEXT para la creación del código.

170 Se asigna a la variable «A\$» (código secreto) el valor obtenido de esta misma variable más la cadena literal conseguida con la función «NUMEROS(A(I))».

180 Acaba el ciclo FOR...NEXT.

190 Borrarmos la pantalla.

200 Visualización de un texto orientativo.

210 REM que nos indica la entrada de datos.

220 Las variables «B» y «L» toman el valor 0. Son las que nos indican el número de cifras correcto y el de las que están colocadas en su lugar.

230 Limpiamos la ventana #1.

240 Se pregunta el número a introducir y se le asigna, como cadena literal, a «X\$».

250 Comprobamos que se han teclado cuatro cifras. En caso contrario el ordenador nos pide un nuevo dato (vuelve a la línea 230).

260 En cada paso la variable «N» —que contiene el número de veces que hemos intentado averiguar el código— se incrementa en una unidad.

270 REM. Nos dice que comienza el análisis.

280 Ciclo FOR...NEXT con «L» como variable de control.

290 Mediante el bucle y con el comando MID\$ se comparan los caracteres que ocupan la misma posición dentro del número introducido («X\$») y el código («A\$»). Si los caracteres de esas posiciones son iguales la variable «L» se incrementa en uno.

300 Analizamos si la cifra que ocupa la posición «L» dentro de la cadena «X\$» es igual a alguna de las cuatro del código. En caso afirmativo, «B» aumenta en uno.

310 Finaliza el bucle FOR...NEXT.

320 REM que informa de la exposición de los resultados.

330 Se escribe en la columna 11 el número introducido, dejando 6 espacios (SPC) la variable «B» y dejando un espacio la «L».

340 Si «L» es igual a 4, es decir, las 4 cifras están en su lugar, se salta a la línea 370.

350 Si «N» es igual a 10 hemos agotado todas las tentativas y la ejecución salta a línea 380.

360 Se retorna a la 220 para continuar averiguando el código.

370 Escritura en pantalla de un mensaje de felicitación por el acierto.

380 Aparece en pantalla el código secreto.

390 Cuando se pulse una tecla se vuelve a ejecutar el programa.

```

ANALISIS
10 REM *****
20 REM MICROHOBBY AMSTRAD
30 REM MASTERMIND
40 REM *****
50 REM DEFINICION DE FUNCIONES
Y VENTANAS
60 DEF FN NUMEROS(X)=RIGHT$(STR$(X),1)
70 WINDOW #1,15,30,20,20
80 REM PREPARACION DEL CODIGO SECRETO
90 A(1)=INT(RND*10)
100 A(2)=INT(RND*10)
110 IF A(1)=A(2) THEN GOTO 100
120 A(3)=INT(RND*10)
130 IF A(1)=A(3) OR A(2)=A(3) THEN GOTO 120
140 A(4)=INT(RND*10)
150 IF A(1)=A(4) OR A(2)=A(4) OR A(3)=A(4) THEN GOTO 140
160 FOR I=1 TO 4
170 A$=A$+FN NUMEROS(A(I))
180 NEXT I
190 CLS
200 LOCATE 10,4:PRINT "NUMERO ACIERTOS":PRINT
210 REM ENTRADA DE NUMEROS
220 L=0:B=0
230 CLS #1
240 INPUT #1, "NUMERO ",X$
250 IF LEN(X$)>4 THEN GOTO 230
260 N=N+1
270 REM COMPARACION DEL NUMERO CON EL CODIGO
280 FOR I=1 TO 4
290 IF MID$(X$,I,1)=MID$(A$,I,1) THEN L=L+1
300 IF MID$(X$,I,1)=MID$(A$,I,1) OR MID$(X$,I,1)=MID$(A$,2,I) OR MID$(X$,I,1)=MID$(A$,3,I) OR MID$(X$,I,1)=MID$(A$,4,I) THEN B=B+1
310 NEXT I
320 REM EXPOSICION DE RESULTADO
330 PRINT TAB(11);X$;SPC(6);B;
340 SPC(11);L;"
350 IF L=4 THEN GOTO 370
360 IF N=10 THEN GOTO 380
370 LOCATE 7,18:PRINT "ACERTO, NUMERO FELICIDADES"
380 PRINT #1,"ERA "A$
390 WHILE INKEY="" :NEND:RUN
  
```



Pro que los datos no realicen el trabajo duro, M.P. AMSTRAD le hace por ti. Todos los modelos que incluyen este logotipo se encuentran a tu disposición en cualquier manual, solicitados.

SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

OLVIDA TODO LO QUE HAS VISTO

No
en
U.S.A.

U.S. GOLD
SOFTWARE

DATA EAST

KUNG-FU MASTER

KUNG-FU MASTER

Commodore 64/128

LA VERSION OFICIAL DE LAS MAQUINAS

THE OFFICIAL COIN OP GAME

KUNG-FU MASTER
¡¡EL DEFINITIVO!!

Ofites Informática

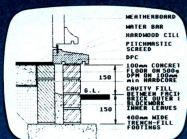
Presenta: la tableta gráfica GRAFPAD II-

LO ÚLTIMO EN DISPOSITIVOS DE ENTRADA DE GRAFICOS PARA AMSTRAD, COMMODORE Y BBC

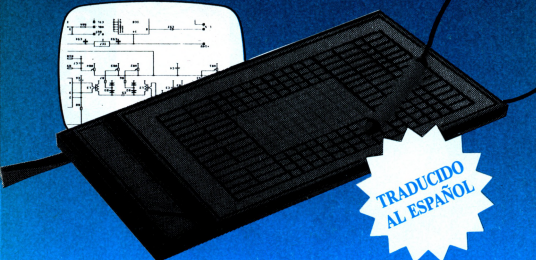
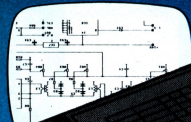
La primera tableta gráfica, de bajo costo, en ofrecer la duración y prestaciones requeridas por las aplicaciones de negocios, industria, hogar y educación. Es pequeña, exacta y segura. No necesita ajustes ni mantenimiento preventivo. GRAFPAD II es un producto único que pone la potencia de la tecnología moderna bajo el control del usuario.



DIBUJO A MANO ALZADA
SOFTWARE DE ICONOS



DISEÑO DE ARQUITECTURA
CON SOFTWARE DDV



**TRADUCIDO
AL ESPAÑOL**

COMBINA EN UN ÚNICO DISPOSITIVO TODAS LAS PRESTACIONES DE LOS INTENTOS PREVIOS DE MECANISMOS DE ENTRADA DE GRAFICOS. LAS APLICACIONES SON MAS NUMEROSAS QUE EN LOS DEMAS DISPOSITIVOS COMUNES E INCLUYEN:

- selección de opciones
- entrada de modelos
- recogida de datos
- diseño lógico
- diseño de circuitos
- creación de imágenes
- almacenamiento de imágenes
- recuperación de imágenes
- diseño para construcción
- C.A.D. (diseño asistido por ordenador)
- ilustración de textos
- juegos
- diseño de muestras
- educación
- diseño PCB.

ESPECIFICACIONES

RESOLUCION:

1.280 x 1.024 pixels.

PRECISION:

1 pixel.

TASA DE SALIDA:

2.000 pares de coordenadas por segundo.

INTERFACE:

paralelo.

ORIGEN:

borde superior izquierdo o seleccionable.

DIMENSIONES:

350 x 260 x 12 mm.

DISPONIBLE AMSTRAD:

CASSETTE 23.900 ptas.

DISCO 25.900 ptas.

(IVA NO INCLUIDO)

- FACIL DE USAR.
- TRAZADO PCB.
- C.A.D.
- AREA DE DISEÑO DIN A4.
- COLOR EN ALTA RESOLUCION.
- USO EN HOGAR Y NEGOCIOS.
- VARIEDAD DE PROGRAMAS DISPONIBLES.
- DIBUJO A MANO ALZADA.
- DIAGRAMAS DE CIRCUITOS.

DE VENTA EN LOS MEJORES COMERCIOS DE INFORMÁTICA

Si Vd. tiene alguna dificultad para obtener la tableta gráfica, puede dirigirse a:



Avda. Isabel II, 16 - 8º

Tels. 455544 - 455533

Télex 36698

20011 SAN SEBASTIAN

CONDICIONES ESPECIALES PARA DISTRIBUIDORES

Ofites Informática

Presenta: **el lápiz al que gusta decir SI**
mientras nuestros competidores dicen no
UNICO PARA AMSTRAD, CON PRECISION PIXEL

FUNCIONES	ESP	de otros	OTROS
UNICO MENU DE PANTALLA	SI	NO	
ARRASTRE OBJETOS PANTALLA	SI	NO	
TRASLADO OBJETOS PANTALLA	SI	NO	
TRASLADO DE CURSOR	SI	NO	
CAJAS ELASTICAS	SI	SI	
LINEA ELASTICA	SI	SI	
TRIANGULO ELASTICO	SI	NO	
ELIPSE ELASTICO	SI	NO	
DIAMANTE ELASTICO	SI	NO	
POLIGONO ELASTICO	SI	NO	
HEXAGONO ELASTICO	SI	NO	
OCTOGONO ELASTICO	SI	NO	
CUBO ELASTICO	SI	NO	
PIRAMIDE ELASTICA	SI	NO	
CIRCUNFERENCIAS	SI	SI	
CIRCULOS RELLENOS	SI	NO	
CAJAS RELLENAS	SI	NO	
ELIPSES RELLENAS	SI	NO	
CUNAS	SI	NO	
SIMULADOR DE CORTES	SI	NO	
DISEÑO DE ZOOM	SI	SI	
IMAGEN ESPEJO E INVERTIDA	SI	NO	
FONDO DE REFERENCIA	SI	NO	
REJILLA DE FONDO	SI	NO	
OPCION DISPLAY X, Y	SI	NO	
RELLENADO CON COLOR	SI	SI	
LAVADO DE COLOR	SI	NO	
VOICADO PANTALLA RESIDENTE	SI	NO	
DIBUJO DE BORDES EN 3 D	SI	NO	
TEXTO	SI	SI	
9 TAMANOS DE BROCHA	SI	NO	
18 TOBERAS MOSTRADORAS	SI	NO	
4 MEZCLAS BASICAS	SI	NO	
VARIADOR DE MEZCLAS	SI	NO	
SOMBREADO DE MEZCLAS XOR	SI	NO	
FICHERO ICONOS RESIDENTES	SI	NO	
FICHERO RELLENOS RESIDENTES	SI	NO	
26 COLORES DE PAPEL	SI	NO	
PALETA DE 15 TONOS DE COLOR	SI	NO	
POSICIONAMIENTO DE PUNTO	SI	SI	
RAYOS DESDE UN PUNTO FLO	SI	NO	
DIBUJO REFLEJADO (ESPEJO)	SI	NO	
FUNCION HOME	SI	NO	
CONTROL DESDE TECLADO	SI	SI	
CONTROL CON JOYSTICK	SI	NO	
DISPONIBLES MODOS 1 Y 2	SI	?	



Compare con otros lápices



ESTOS SON ALGUNOS EJEMPLOS DE LOS GRAFICOS QUE VD. PODRA REALIZAR CON NUESTRO LAPIZ OPTICO



DE VENTA EN LOS MEJORES COMERCIOS DE INFORMATICA

Si Vd. tiene alguna dificultad para obtener el lápiz óptico, puede dirigirse a:



Avda. Isabel II, 16 -8º
 Tels. 455544 - 455533
 Télex 36698
 20111 SAN SEBASTIAN

DISPONIBLE PARA:

- CPC 464 CASSETTE 4.900 Ptas.
- CPC 464-664 DISCO 6.900 Ptas.
- CPC 6128 DISCO 6.900 Ptas.

(IVA no incluido)

CONDICIONES ESPECIALES PARA DISTRIBUIDORES

CADENAS

Las cadenas literales son las variables cuyo nombre termina con el signo del dólar(\$) y que pueden estar formadas por letras, números, signos de puntuación y espacios todo ello, englobado en un solo nombre.



Los caracteres que forman la cadena van entre comillas, puesto que son los delimitadores que señalan el principio y el final de la misma.

Cuando la variable esté formada por cifras no debe olvidar que se trata de literales y no se pueden realizar cálculos aritméticos con ellas o nos dará un error de incongruencia de tipos (type mismatch).

Sin embargo, podemos unir o «concatenar» cadenas utilizando el signo de sumar (+) y así formar una tercera diferente. Teclee:

```
primero$ = «primero»
segundo$ = «segundo»
unido$ = primero$ + segundo$
PRINT unido$
```

Vamos a ver gráficamente la diferencia existente entre sumar números o cadenas y compruebe lo que le decimos:

```
PRINT 2+2
```

da como resultado una cifra, mientras que:

```
PRINT «2»+«2»
```

nos devuelve una cadena concatenada. De modo que en el caso de las variables literales el signo + se lee «unir» no «sumar».

El programa 1 es un ejemplo de unión de cadenas, preste atención a la línea 70, es donde se realiza la concatenación.

Programa uno

Si ha ejecutado el ejemplo, habrá notado que las comillas no aparecen en pantalla. Como dijimos son limitadores, no forman parte de la cadena. De modo que si queremos que nos aparezcan debemos ingenierarnos de alguna manera para ha-

```
10 REM PROGRAMA I
20 CLS
30 asterisco$="*"
40 FOR fila=1 TO 10
50 LOCATE 11-fila,fila
60 PRINT asterisco$
70 asterisco$=asterisco$+"*"
80 NEXT fila
```

cerlo. Pero antes de nada vamos a acercarnos a lo que se llama código ASCII.

Como probablemente ya sabe su **Amstrad** trabaja a base de números. Absolutamente todos los caracteres tienen su código numérico correspondiente y todos ellos están recopilados en el Apéndice III del Manual del Usuario. Pero entre nosotros no nos comunicamos numéricamente, por lo tanto para entendernos con el ordenador necesitamos una sentencia que convierta el código numérico en un símbolo. Se trata de CHR\$. Véalo con:

```
PRINT
CHR$(71)CHR$(65)CHR$(84)CHR-
$(79)
```

y de paso comprenderá por qué se llaman cadenas.

De modo que gracias al código ASCII y la función CHR\$ podemos crear cualquier tipo de cadena for-

mada por toda clase de signos como decíamos en un principio.

Podemos librarnos de la necesidad de tener que mirar en una tabla el código ASCII de un carácter utilizando la función ASC que realiza justamente el camino contrario: coge un carácter y nos devuelve su código,

```
PRINT ASC(«G»)
PRINT ASC(«A»)
PRINT ASC(«T»)
PRINT ASC(«O»)
```

lo demuestra. ASC es capaz de trabajar con cadenas y de diferenciarlas de las cifras, pero sólo nos devuelve el primer carácter, ignorando el resto. Pruébalo con PRINT ASC(«GATO»).

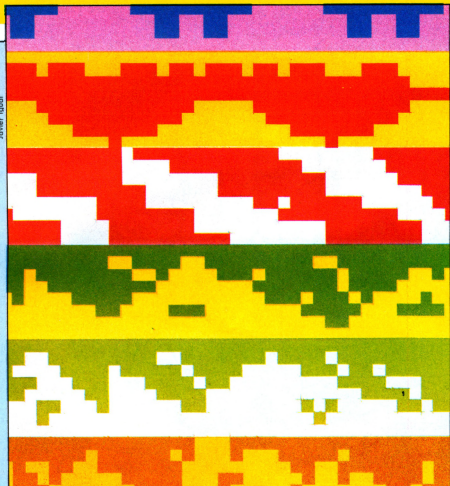
Pero ASC es algo más que un medio rápido para encontrar un código ASCII. Puede utilizarse para depurar datos, detectando los códigos errores del usuario como lo demuestra el Programa II.

Programa dos

Aquí se aceptan solamente las letras mayúsculas, de ello se encarga la línea 40 que revisa el valor ASCII de «entrada\$». Si ésta fuera del rango 65-90 (correspondiente a las mayúsculas) lo señala como un error, y sólo se ejecuta el programa hasta el final cuando el dato está en el margen señalado anteriormente, es decir, cuando introducimos una letra mayúscula.

```
10 REM PROGRAMA II
20 PRINT "Teclee una letra mayuscul
a "
30 INPUT entrada$
40 IF ASC(entrada$)<65 OR ASC(entra
da$)>90 THEN CLS:PRINT "Ha dicho un
a letra mayuscula":PRINT:GOTO 20
50 PRINT "Bien hecho!"
```

Sin embargo, un error como confundir una mayúscula con una minúscula es perfectamente humano y puede volvernos locos cuando el **Amstrad** lo rechace. El Programa III demuestra cómo ASC permite a nuestro ordenador corregir errores humanos. La línea 60 comprueba si se trata de una letra, ya sea mayúscula o minúscula.



Programa tres

Una vez superado positivamente el filtrado, la línea 80 comprueba si se trata de una minúscula, caso de ser así —su código es mayor que 90— resta 32 para pasarlo al grupo de las mayúsculas.

Pero ASC no es el único medio de

ciones afecta a los signos ni a los números.

Compare el Programa IV con el anterior y observará la economía que supone el uso de estas nuevas funciones.

Programa cuatro

Vamos a ver de cerca los códigos ASCII 0 a 31, ya que no imprimen caracteres cuando se usan con CHR\$, sino que se trata de códigos de control que influyen en el micro. Todos están en el capítulo IX del Manual, pruébelos. Al igual que todos los anteriores, pueden concatenarse a una cadena normal.

```
10 REM PROGRAMA IV
20 PRINT "Teclee una letra ";
30 INPUT entrada$
40 ascii=ASC(entrada$)
50 REM Comprueba si la entrada esta
dentro del rango de las letras
60 IF ascii<65 OR ascii>122 DR (asc
11)>90 AND ascii<97) GOTO 20
70 REM Si es minúscula, la transfor
ma en mayúscula
80 IF ascii>90 THEN ascii=ascii-32
90 entrada$=CHR$(ascii)
100 PRINT entrada$
```

solventar este problema, hay dos funciones que facilitan el trabajo. Si añadimos UPPER\$ a las cadenas, todas las letras minúsculas pasarán automáticamente a mayúsculas, mientras que la función LOWER\$ hará justamente lo contrario. Por supuesto que ninguna de las dos fun-

Primeros repasos

Antes de dejar CHR\$ intente esto:

```
cosa$ = CHR$(34) + «GATO» + CHR$(34)
```

y se resolverá aquella cuestión que se nos planteó al principio de que se nos visualizaran las comillas.

Para terminar vamos a dejar planteado un problema que a primera vista puede parecer tonto. En cualquier momento podemos necesitar conocer la longitud de una variable literal que puede haber estado cambiando todo el tiempo durante la ejecución del programa, de modo que nos será difícil determinar en un momento dado el número de caracteres que contiene.

La función que nos va a solucionar

```
10 REM PROGRAMA V
20 entrada$=""
30 WHILE LEN(entrada$)<4
40 PRINT "Teclee una palabra de cua
tro letras ";
50 INPUT entrada$
60 WEND
70 PRINT entrada$
```

el problema es LEN, su uso es muy sencillo: PRINT LEN («nombre de la variable») nos da la longitud de la misma. Pero tiene un par de casos muy interesantes:

Recuerde que los espacios vacíos cuentan como un carácter, compruébelo con:

```
PRINT LEN («»)
```

Y podemos definir una cadena nula, que no contenga ningún carácter —nulo\$=«». ¿Para qué queremos una cadena nula? Puede sernos muy útil como condición de control de un bucle WHILE... WEND, como lo demuestra el Programa V donde LEN funciona como un depurador que controla la correcta longitud de la palabra tecleada.

¡ JUEGA EL JUEGO DEL QUE TODOS HABLAN !

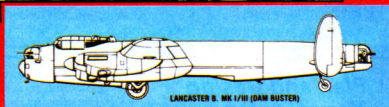
Distribuido en
España por
ERBE
Software

THE DAM BUSTERS

OFFICIALLY ENDORSED BY
R.A.F. No. 617 SQUADRON



Available for
SPECTRUM
48k £9.95



¡Apasionante!

Son las 21:15 horas del 16 de Mayo. Un bombardero Lancaster en vuelo especial, despegue de Inglaterra hacia Alemania. Después de meses de preparación, el escuadrón 617 vuela en una operación destinada a cambiar el curso de la II Guerra Mundial. Su objetivo es destruir las más importantes presas alemanas para paralizar los puntos vitales de sus fábricas de armamento.

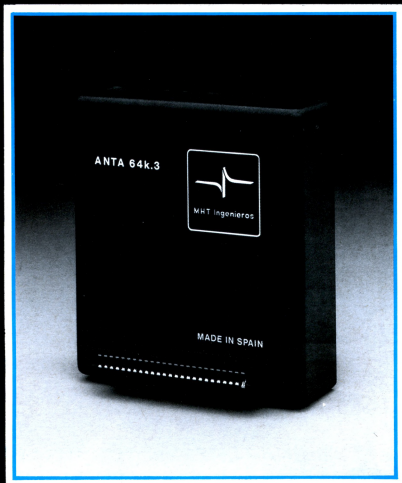
Este detallado y auténtico simulador te permite ocupar los puestos de **Piloto, Ingeniero de vuelo, Artillero delantero y trasero, Bombardero y Navegante**. Volarás a través del Canal de la Mancha y Europa intentando evitar a los temibles ME-110 alemanes, zeppelins, focos antiaéreos y todos los demás peligros a los que se enfrentó el comando Inglés.

PIDE ESTOS PROGRAMAS A ERBE, SANTA ENGRACIA, 17, 28010 MADRID. T.FN. (91) 447 34 10 - Y EN LAS MEJORES TIENDAS DE INFORMÁTICA, TIENDAS Y MAYORISTAS... CUMPLIMENTAMOS SUS PEDIDOS EN 24 HORAS

ANTA 64K.3

Los 64K de memoria que esperaba su Amstrad

*Ampliación de memoria, buffer de impresora y ram disk**



Si tiene un **AMSTRAD CPC 464, CPC 664 o CPC 6128** conéctele el **ANTA 64K.3** y seleccione la opción que necesite:

64K de Memoria

Para leer y escribir datos, cadenas y bloques de caracteres, así como copiar o trasladar pantallas.

64K de Buffer de Impresora

Permite seguir trabajando

con el ordenador mientras la impresora funciona.

64K de Ram Disk/Basic

La memoria simula el funcionamiento de un disco con mejor tiempo de acceso.

*Software de manejo contenido en ROM.

MHT Ingenieros

