

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO II N.º 26

160 Ptas.

Canarias 165 pts.

**SUBROUTINAS
ESTRUCTURADAS
Y PROGRAMACION
MODULAR**

**EL TONO,
LA PIEZA CLAVE
EN EL MUNDO
DE LA MUSICA**

**Ajuste y
representación de rectas
por mínimos cuadrados**

**DIBUJAR
CON UN
SOLO DEDO Y
STAR MOUSE**

SOFTWARE

**NIGHTSHADE:
La ciudad encantada
de los espectros
demoniacos
de la noche**



Te ofrecemos algo "muy Especial"

En el mes de septiembre nació una **AMSTRELLA** que vino a demostrar que los **Amstrad** estaban ahí y había que contar con ellos. Hoy, miles de personas nos dan la razón.

Por este motivo, y después de los 6 primeros meses de andadura juntos con nuestros lectores, ha vuelto a suceder algo muy importante: ha nacido una **AMSTRELLA MUY ESPECIAL**.

¡Ya está a la venta en tu quiosco!
por sólo **350** ptas.

MICROHOBBY AMSTRAD
Especial Año I N.º 1

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

350 Ptas.

CONTROL DEL IVA POR ORDENADOR

HISTORIA DE LA INFORMÁTICA: EL ENIAC A LA QUINTA GENERACION

TODOS LOS COMANDOS RSX PARA TU AMSTRAD

¿QUE SE PUEDE HACER CON UN ORDENADOR?

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

Nombre _____ C. Postal _____
Localidad _____ Edad _____
Teléfono _____

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? Sí No

DESEO RECIBIR El Especial de **MICROHOBBY AMSTRAD** Profesional

FORMA DE PAGO Talón bancario adjunto o nombre de **HOBBY PRESS S. A.**
 Medianeira, tarjeta de Crédito, N.º _____ Fecha de caducidad _____
 Contra reembolso (importe 75 ptas. de gastos de envío).
Fecha y firma: _____

HOBBY PRESS
Para gente inquieta.

AMSTRAD

sumario

Año II • Número 26 • 25 de Febrero al 3 de Marzo 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-Centurió

Director Ejecutivo
Victor Prieto

Subdirector

José María Díaz

Redactora Jefe

Marta García

Diseño gráfico

José Flores

Colaboradores

Francisco Portalo, Pedro Sudán

Miguel Sepúlveda,

Francisco Martín,

Jesús Alonso, Pedro S. Pérez,

Amalia Gómez,

Juan J. Martínez,

David Sopena, Alberto Suárez,

Eduardo R. Velasco,

Javier Barcelo

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Javier Martínez

Portada

M. Barco

Ilustradores

Javier Igual, J. Pons, F. L.

Frontán, J. Septien, Pejo, J. J.

Mora, Luigi Pérez, J. Siemens

Edita

HOBBY PRESS S.A.

Presidente

Maria Andriano

Consejero Delegado

José I. Gómez-Centurió

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortes

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

Secretaría de Dirección

Marisa Cagorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

La Granja, 39

Poligono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDEC, S. A. Crta. de Irún,

Km. 12,450 (MADRID)

Fotocomposición

Novacom, S.A.

Nicolás Morales, 38-40

Fotomecánica

GICF

Ezequiel Solano, 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

América 1.532. Tel.: 71 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

Sistema de desarrollo para Amstrad. Norris deja Control Data.

6 Primeros pasos

Estructura y modularidad en los programas son las dos palabras mágicas que separan a un programador de un aficionado. En Basic, esta estrategia no puede llevarse al más alto nivel, como en otras lenguajes, pero algo se puede hacer, con las subrutinas.

Serie Oro 12

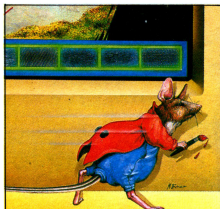
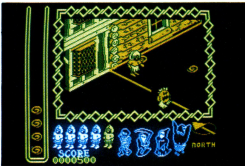
Ya se sabe, un ordenador es la mejor calculadora del mundo, el sueño de cualquier científico o estudiante de Ciencias, porque, ¿quién no ha tenido que realizar un ajuste por mínimos cuadrados a mano? Con el programa de este número eso no sucederá más.

16 Análisis

Con la excusa de mostraros cómo se puede programar un reloj en Basic, no hemos podido resistir la tentación de mostrar también el uso del comando EVERY para ver cómo más de un programa puede usarse simultáneamente.

Mr. Joystick 18

Ultimate nos maravilla de nuevo con su creación más sofisticada Nightshade y la ciudad encantada de la noche.



Banco de pruebas 20

Star Mouse es una utilidad al estilo del mítico MacIntosh, pensada para dotar al ordenador de un entorno de creación artística mucho más amigable que el que posee de fábrica. Lo estudiamos a fondo.

24 Programación

El tono es lo que básicamente distingue a un sonido de otro, lo que separa una flauta de un violín. Por eso, como es tan importante, analizamos cómo se maneja en el Amstrad esta posibilidad.



Código Máquina 28

Esta vez le toca el turno a las instrucciones de transferencia de bloques. Detrás de un nombre tan aparatoso, se esconden una serie de órdenes de las más útiles del Z80.

POWER

SOFTWARE, S.A.

NAPOLES, 98, 1° 3ª - Tels. 232 24 61
08013 BARCELONA (SPAIN)

¡ATENCIÓN! SOLO POWER SOFTWARE, S.A.
POSEE LOS DERECHOS DE FABRICACION Y
COMERCIALIZACION EN EXCLUSIVA PARA TÓDA
ESPAÑA DEL PROGRAMA...

CONTIENE UN MARAVILLOSO POSTER
A TODO COLOR DE LOS PICAPIEDRA Y
LAS INSTRUCCIONES DEL JUEGO EN
CASTELLANO.

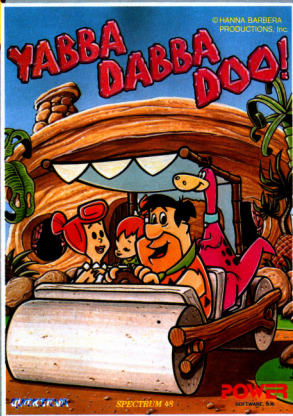
YABBA DABBA DOO!

1.800 Pts.

COMMODORE CASSETTE
COMMODORE DISKETTE
AMSTRAD CASSETTE
AMSTRAD DISKETTE
SPECTRUM



POWER SOFTWARE, S.A. PERSEGUIRÁ POR
TODOS LOS MEDIOS LEGALES Y/O
COMERCIALES A SU ALCANCE, CUALQUIER
TIPO DE PIRATERIA, COMERCIALIZACION,
IMPORTACIONES O DUPLICADOS ILEGALES
QUE SOBRE SUS PROGRAMAS SE PRACTIQUE.



Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.

Nombre

Dirección

Teléfono

Firma:

SISTEMA CANTIDAD

Contrarreembolso. Adjunto Talón. Giro Postal.

Deseo recibir información de sus programas en: MSX AMSTRAD

¡YABBA DABBA DOO!

Primera plana

NORRIS DEJA CONTROL DATA



A

través de su ya larga carrera como hombre de negocios, William Norris, fundador y máximo director de Control Data, de 74 años de edad, se ha ganado una controvertida reputación de innovador y visionario.

Para algunos, Norris es un genio que creó de la nada la primera compañía que fabricó supercomputadores y creció rápidamente hasta convertirse en una fuerza mayor en el mundo del proceso de datos.

Para otros, se trata de un «cabezadura» bien intencionado que ha comprometido a su compañía en proyectos bienintencionados pero de dudosa rentabilidad, y le consideran directamente responsable de la difícil situación que Control Data atraviesa en estos momentos.

Hace poco, William decidió que ya era hora de retirarse y así se lo comunicó a su gente en el cuartel general de la compañía en Minneapolis. Le sucederá Robert Price, de 55 años, que trabaja estrechamente con Norris desde hace tiempo.

En 1985, Control Data ha perdido del orden de 300 millones de dólares, ha tenido que despedir a 7.000 personas y atraviesa serias dificultades financieras.



COMO UNIX, PERO MAS BARATO

H

oy día, el Unix es sin duda el sistema operativo de moda en entornos multiusuario y multitarea, pero tiene algún que otro inconveniente: primero, es muy caro; segundo, el sistema operativo completo ocupa más o menos 15 Megabytes en un disco duro.

Un buen intento de hacer accesible algo parecido al UNIX para un PC es el sistema operativo UNETIX, que permite ejecutar concurrentemente en un PC hasta 10 tareas, una de las cuales puede ser un programa PC-DOS (DOS nativo del PC).

La casa creadora del UNETIX, Lantech Systems, asegura que su sistema es compatible con el UNIX de AT&T (el no va más del UNIX), pero mucho más barato (250\$), y, aunque se recomienda disco duro, puede correr en un PC con dos unidades de disco.

SISTEMA DE DESARROLLO PARA AMSTRAD

Gremlin Graphics, una conocida casa de software que ha creado la serie de Monty Mole, la versión de Amstrad de Rocky y Profanation, ha creado una división encargada de la comercialización de utilidades para toda la serie CPC de Amstrad, etiquetada bajo el nombre de Discovery.

El primer producto de la nueva división se denomina PYRADEV, y es nada menos que un sistema de desarrollo completo que permite crear al programador software para Amstrad, tanto bajo Amsdos como CP/M.

El paquete contiene 5 programas:

- Un editor de 80 columnas, de los llamados a pantalla completa.
- Un Macroensamblador-linker para Z80.

- Un programa Monitor para depuración de software junto con un desensamblador.

- Un programa llamado DISC-NURSE pensado para manipular el disco a nivel de sectores.

- Una utilidad de manejo de ficheros en disco llamada FILE MANAGEMENT.

El editor permite editar rápidamente ficheros de hasta 32 kbytes de extensión, y usa sus propias rutinas de lectura y escritura en disco.

El macroensamblador/linker permite incluir hasta 32 ficheros fuente en cualquier disco, con lo que podemos crear librerías de rutinas comunes a gran mayoría de programas, y usarlas una y otra vez sin reescribirlas.

El Monitor servirá para depurar un programa en lenguaje máquina de muchas formas: ejecutándolo paso a paso, desensamblándolo, etc.

En cuanto a los otros dos programas, con ellos podemos examinar un disco con lupa y editar cualquier sector del mismo, corregirlo y modificarlo, etc.

Lo más importante: los programas harán uso de toda la memoria disponible en el sistema, con lo que el 128 y los 464-664 ampliados presentarán importantes mejoras.



BIENVENIDAS SUBROUTINAS

Poco a poco nos vamos convirtiendo en unos «expertos» programadores y, por este motivo, ha llegado la hora de conocer unas herramientas que van a servir para ahorrar bastante trabajo, y a la vez vamos a conseguir con ellas que nuestros programas sean mucho más claros y legibles. Son las «subrutinas».

H

asta ahora estamos acostumbrados a construir un programa todo seguido, de principio a fin, de forma que nos hemos limitado a ir poniendo instrucción tras instrucción en el orden correcto para que se fueran realizando secuencialmente —una tras otra— todas y cada una de las acciones que habíamos previsto.

Pero imaginemos que hay una serie de instrucciones que realizan una misma función y que se repiten varias veces dentro de nuestro programa. Hemos tenido que teclear más de una vez las mismas o parecidas líneas de instrucciones.

Y, ¿no le ha dado la sensación de, al escribirlas repetidamente, estar realizando un trabajo innecesario o al menos ingrato? Sea sincero.

Este tipo de programa podría ser algo parecido al programa 1 que va a escribir una ficha con todos nuestros datos separándolos por una línea de asteriscos, por ejemplo.

Lo vamos a dividir en dos partes diferenciadas. La primera será la que nos pide todos nuestros datos y los va metiendo en las distintas variables —líneas 30 a 80— mediante tantas instrucciones INPUT como datos queramos dar al ordenador.

La segunda es la que nos saca en la pantalla todos y cada uno de los datos que le hemos ido dando. Vamos escribiendo la ficha por medio de las instrucciones PRINT de las líneas 100 a 190.

Observe una cosa: las líneas 100, 130, 160 y 190 son iguales. Todas ellas lo único que hacen es separar de una forma estética cada uno de los datos. Pero si todas ellas cumplen una misma misión, ¿para qué repetirlas?

La necesidad de las subrutinas

La respuesta a esta pregunta es muy sencilla. Aunque como hemos dicho al comenzar ya somos unos «diestros» programadores, la verdad es que escribimos varias veces una misma línea porque de momento no nos queda

otro remedio». No conocemos otro sistema de hacerlo.

Lo primero que tenemos que hacer es colocar al final del programa el grupo de instrucciones que se repiten —en este caso es sólo una.

A esta serie de líneas que realizan una función muy particular —escribir asteriscos— que vamos a tener que realizar varias veces a lo largo de nuestro programa es lo que llamamos «SUBROUTINA».

Bueno, ya hemos delimitado qué es lo que vamos a tener que repetir. Sólo nos queda decirle la forma en que debe indicar al ordenador que vaya a ejecutar esta subrutina cuando lo necesite y la manera de volver al programa del que provenimos.

Con:

GOSUB número

le ordenamos al **Amstrad** que se dirija a la subrutina cuya primera línea es la indicada por «número».

Una vez allí, continúa la ejecución siguiendo todas las líneas de este subprograma.

Ida o vuelta

La última instrucción deberá ser un comando que indique al ordenador que vuelva automáticamente al programa principal. No olvide que se trata de un viaje de ida y vuelta. Basta con teclear:

RETURN

y estaremos de regreso. Pero, ¿dónde saltaremos en este viaje de vuelta? Piénselo un poquito.

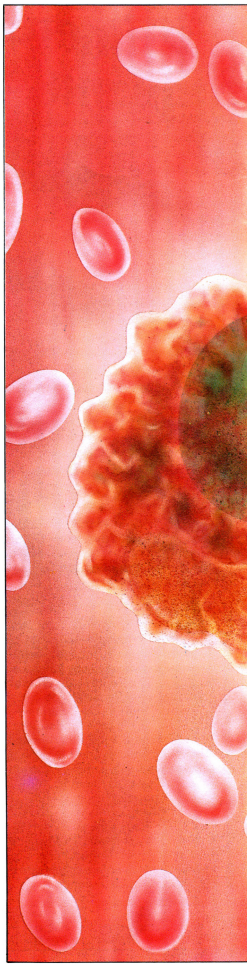
La última instrucción del programa principal que se ha ejecutado es la que nos indicó que debíamos ir hacia la subrutina, y esto ya lo hemos hecho. Luego tendremos que regresar a la línea siguiente y allí continuar con el programa.

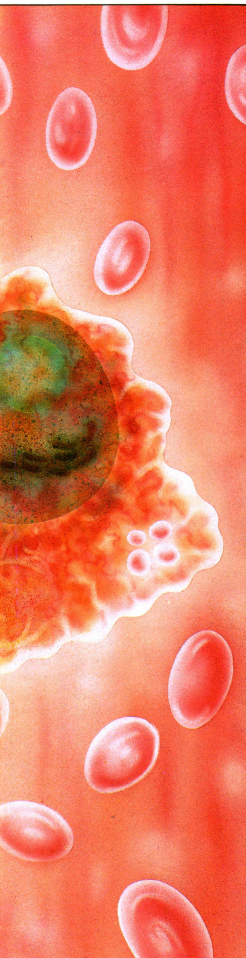
Por ejemplo, si tenemos una línea semejante a:

130 GOSUB 1000

una vez hayamos cumplido lo especificado en la rutina que comienza en la línea 1000, con RETURN saltaremos a la siguiente a la 130.

El programa 2 nos dará una visión clara de cómo se produce este viaje de ida y vuelta.





La línea 50 sacará un mensaje indicándonos que en ese momento saltaremos a la subrutina de la línea 1000.

Cuando ya estemos en ella continuamos la ejecución normalmente. Este subprograma nos sacará un amable saludo —línea 1020— y cuando llegue al RETURN de la 1060 mandará al ordenador de vuelta al programa principal.

Y lo hacemos a la 60 tal como nos afirma el PRINT contenido en dicha línea.

Pero, de repente, nos encontramos con una cosa nueva:

```
100 END
```

¿Qué pinta aquí? Pruebe a quitarla y observe qué es lo que ocurre con la ejecución del programa.

Comprobará que no se detiene. Si no se utiliza END el ordenador continuará el proceso del programa siguiendo el orden de numeración de las líneas y ejecutará de nuevo la subrutina completamente hasta que llegue al final. —RETURN—.

Aquí nos obsequiará con el mensaje:

```
3 Unexpected RETURN in 1060
```

que significa que ha encontrado un RETURN sin haber saltado a una subrutina. Y así ha sido, ya que no ha habido salto sino a continuación por ella —falta la instrucción END—.

Para evitar esto es para lo que se ha colocado la línea

```
100 END
```

Y en ella termina el programa. **¿Ha quedado lo suficientemente claro?**

Pero volvamos a nuestro programa 1. Ahora ya conocemos la manera de evitarnos teclear todas las líneas que son iguales. Así que ¡manos a la obra!

Con y sin subrutinas

Eche un vistazo al programa 3 y compruebe las diferencias que existen entre ambos.

A partir de la línea 1000 hemos delimitado unas instrucciones de programa que van a realizar una función muy concreta —en este caso escribir una hilera de asteriscos—. Van a constituir una subrutina que simplemente tenga esa misión.

Observe también que hemos sustituido en el programa principal cada grupo de líneas que se repeta por una llamada a la subrutina. Y no se olvide del END.

¿No le parece que el programa ha quedado mucho más claro? Además, en este caso hay muy poca diferencia entre escribir una línea de asteriscos o una indicación al ordenador que salte a la subrutina, es muy sencilla.

Pero imagine que en vez de una línea, fuera un largo proceso de cálculo formado por miles de instrucciones. ¿Se atrevería a teclearlo repetidas veces y sin errores? No imaginamos su respuesta: ¡NO, prefiero utilizar el GOSUB!

Fijemos conceptos. Una subrutina es un pro-

Primeros pasos

grama dentro de otro general que sólo se ejecuta cuando se lo ordena el programa principal mediante una instrucción:

```
GOSUB
```

de llamada a la subrutina. Cuando ésta termina de ejecutarse, devuelve el control al principal mediante la orden:

```
RETURN
```

Continuemos. Dentro de un programa principal puede haber más de un grupo de instrucciones que se repitan. Cada uno de ellos realizaría una acción concreta que se ejecutaria varias veces. ¿Cómo lo haríamos en este caso?

Sencillamente formaríamos con cada uno de estos grupos una subrutina que necesariamente llevaría como última instrucción RETURN (vuelta al programa principal).

El programa 4 es un buen ejemplo de ello.

Observe que todas las subrutinas están situadas al final del programa principal, es decir, después de la instrucción END de la línea 140.

Por claridad a la hora de seguirlo, hemos numerado las líneas de las subrutinas comenzando por un número múltiplo de 1000, por ejemplo. Es más fácil dirigirse a la línea 8000 mediante:

```
GOSUB 8000
```

que hacerlo a la 3528:

```
GOSUB 3528
```

está de acuerdo, ¿verdad?

¿Nos admite otro consejo? Es muy conveniente que la primera instrucción de la subrutina sea un comentario de la función específica que realiza. No nos cuesta nada poner un oportuno:

```
REM ESTO ES LA INICIALIZACION
```

o algo semejante y seguro que a primera vista ya se va haciendo una idea de qué es lo que hace.

Quizá le estamos insistiendo demasiado en que es muy conveniente que su programa sea claro. Nuestra opinión, como habrá podido comprobar en más de una ocasión, es que además de que su programa funcione es muy importante que cualquier persona con unos conocimientos previos sea capaz de seguirle sin problemas. De ahí todas estas sugerencias que ayudan a que sea lo más legible posible.

En este caso concreto hemos desarrollado tres subrutinas que comienzan en las líneas 1000, 2000 y 3000. Su única misión va a ser

indicarnos mediante un mensaje que la ejecución del programa ha pasado por ahí. Compruébelo.

El programa principal maneja las subrutinas

Las instrucciones comprendidas entre las líneas 10 y 140 son las que forman lo que hemos llamado «programa principal». Es el que ordena y manda que se ejecute una subrutina u otra y determina el orden en que han de ejecutarse. «Es el jefe.»

Si ejecuta el programa 4 comprobará visualmente su funcionamiento así como el orden en el que se van sucediendo las instrucciones. Así está mejor, ¿cierto?

Intentemos ver cómo un pequeño cambio en el «programa principal» influye en el desarrollo de la ejecución. Cambie la línea 50 por:

```
50 GOSUB 2000
```

y la 80 por:

```
80 GOSUB 1000
```

¿Qué ocurrirá si hacemos que el programa corra de nuevo? Pruébelo.

El resultado es muy simple. Como hemos cambiado la colocación de las llamadas a las subrutinas, el ordenador saltará a ellas en el mismo orden que están las llamadas. No importa si una está colocada delante de otra al final del programa, lo que manda a la hora de la ejecución es la secuencia como están colocadas las llamadas.

Si le queda todavía alguna duda haga la siguiente comprobación. Convierta las líneas 80 y 110 en:

```
80 GOSUB 1000
```

y

```
110 GOSUB 1000
```

y ejecute el programa.

Seguimos teniendo ahora las tres subrutinas al final del programa, pero sin embargo, sólo lo hemos pasado por una. Y podría haber sido por cualquier otra ya que no es necesario que sea la primera.

Evidentemente, no sólo podemos crear subrutinas que se encarguen de escribir un mensaje o algo semejante. Cada una de ellas podemos aumentarla todo lo que queramos para que realicen una función concreta más complicada que una simple instrucción PRINT.

Si no fuese así estaríamos muy limitados y hemos comprado un **Amstrad** para algo más que para utilizarlo como máquina de escribir.

Cada subrutina puede ser por sí sola un programa largo y complejo que vamos a tener a nuestra disposición para manejarla desde el programa principal en el orden que sea más conveniente para que obtengamos los resultados deseados. Esto ya puede sernos de más utilidad.

Una pequeña muestra de las posibilidades de manejo de subrutinas es el programa 5.

Se trata simplemente de una sencilla calculadora en la que podemos operar dos números con cada una de las operaciones matemáticas básicas.

Hemos desarrollado cuatro subrutinas, una para cada operación, que nos hacen los cálculos necesarios y nos imprimen el resultado.

La de la suma, que es la primera, comienza en la línea 1000—después del programa principal—y las de la resta, multiplicación y división en la 2000, 3000 y 4000 respectivamente.

Sigámosle. El cuerpo principal del mismo abarca las líneas 10 a 100. Es la zona desde donde se manejan convenientemente las subrutinas. Aquí estarán todas las llamadas puestas en el orden necesario para que nuestra «herramienta» funcione como nosotros queremos.

Hay una recogida de datos en las líneas 30 y 40. En ellas, el ordenador nos va a ir pidiendo cada uno de los operandos por medio de instrucciones INPUT y almacenará los valores que vayamos dando en las variables correspondientes. Esta primera parte ya no le parecerá nueva puesto que es una forma de comenzar que hemos utilizado repetidas veces, ¿no es así?

Después viene una serie de llamadas a cuatro subrutinas. Son las que nos indican que hemos de saltar a las líneas 1000, 2000, 3000 y 4000 respectivamente y continuar allí realizando cada una de las cuatro operaciones matemáticas e imprimiendo los resultados con las adecuadas instrucciones PRINT, (buena se lo indican al **Amstrad**).

Al final de ellas nos encontramos con la instrucción RETURN—líneas 1050, 2050, 3050 y 4050—que, como seguramente ya ha supuesto, es la última sentencia que debe llevar cualquier subrutina y que nos manda de vuelta al programa principal. ¿Entendido?

Solamente nos queda analizar la pregunta de la línea 90. Con ella averiguamos si queremos comenzar a repetir el proceso con más cálculos o damos por finalizado ya nuestro ejercicio matemático. Elija.

Otra cosa. Las subrutinas, lo mismo que los bucles, pueden anidarse. Con esto queremos decir que podemos encontrar llamadas a una subrutina dentro de otra.

No le debe extrañar. Anteriormente le hemos comentado que cada una de ellas podía llegar a ser un programa por sí misma. Por tanto no es exclusivo del «cuerpo principal» el manejo de las subrutinas sino que podemos encontrarlas una llamada a las mismas en cualquier zona del programa.

Y por el momento ya hemos terminado. Le dejamos como tarea, por sí tiene tiempo y ganas, que analice el funcionamiento del programa 4 y vea cómo se manejan las subrutinas anidadas. Piense dónde devuelven el control cada una de las sentencias RETURN y ¡ánimo!

PROGRAMA 1

```
10 REM PROGRAMA I
20 CLS
30 INPUT "NOMBRE: "; nombre$
40 INPUT "APELLIDO: "; apellido$
50 INPUT "CALLE: "; calle$
60 INPUT "NUMERO: "; numero
70 INPUT "POBLACION: "; poblacion$
80 INPUT "PROVINCIA: "; provincia$
90 CLS
100 PRINT"*****"
110 PRINT nombre$; " "; apellido$
120 PRINT
130 PRINT"*****"
140 PRINT"CALLE ";calle$;" NUMERO";
numero
160 PRINT"*****"
170 PRINT poblacion$;" (";provincia$
";)"
180 PRINT
190 PRINT"*****"
*****"
```

PROGRAMA 2

```
10 REM PROGRAMA II
20 CLS
30 PRINT"COMENZAMOS EL PROGRAMA PRINCIPAL."
40 PRINT
50 PRINT"SALTAMOS A LA RUTINA DE LA LINEA 1000";GOSUB 1000
60 PRINT"YA ESTAMOS DE VUELTA"
70 PRINT
80 PRINT"FIN DEL PROGRAMA PRINCIPAL"
90 PRINT
100 END
1000 REM RUTINA NUMERO 1
1010 PRINT
1020 PRINT" ***BIENVENIDO A LA RUTINA NUMERO 1"
1030 PRINT
1040 PRINT" ***FIN DE LA RUTINA NUMERO 1"
1050 PRINT
1060 RETURN
```

PROGRAMA 3

```
10 REM PROGRAMA III
20 CLS
30 INPUT "NOMBRE: "; nombre$
40 INPUT "APELLIDO: "; apellido$
50 INPUT "CALLE: "; calle$
60 INPUT "NUMERO: "; numero
70 INPUT "POBLACION: "; poblacion$
80 INPUT "PROVINCIA: "; provincia$
90 CLS
100 GOSUB 1000
110 PRINT nombre$; " "; apellido$
120 GOSUB 1000
130 PRINT"CALLE ";calle$;" NUMERO";
numero
140 GOSUB 1000
```

```

150 PRINT poblacion%;"(;"provincia#
;"")
160 GOSUB 1000
170 END
1000 REM SUBROUTINA ASTERISCOS
1010 PRINT
1020 PRINT"*****
*****"
1030 RETURN

```

PROGRAMA 4

```

10 REM PROGRAMA IV
20 CLS
30 PRINT"COMENZAMOS EL PROGRAMA PRINCIPAL"
40 PRINT
50 PRINT"SALTAMOS A LA Rutina DE LA LINEA 1000";GOSUB 1000
60 PRINT"YA ESTAMOS DE VUELTA"
70 PRINT
80 PRINT"SALTAMOS A LA Rutina DE LA LINEA 2000";GOSUB 2000
90 PRINT"YA ESTAMOS DE VUELTA"
100 PRINT
110 PRINT"SALTAMOS A LA Rutina DE LA LINEA 3000";GOSUB 3000
120 PRINT"FIN DEL PROGRAMA PRINCIPAL"
130 PRINT
140 END
1000 REM Rutina NUMERO 1
1010 PRINT
1020 PRINT"***BIENVENIDO A LA Rutina NUMERO 1"
1030 PRINT"***FIN DE LA Rutina NUMERO 1"
1040 PRINT
1050 RETURN
2000 REM Rutina NUMERO 2
2010 PRINT
2020 PRINT"***BIENVENIDO A LA Rutina NUMERO 2"
2030 PRINT"***FIN DE LA Rutina NUMERO 2"
2040 PRINT
2050 RETURN
3000 REM Rutina NUMERO 3
3010 PRINT
3020 PRINT"***BIENVENIDO A LA Rutina NUMERO 3"
3030 PRINT"***FIN DE LA Rutina NUMERO 3"
3040 PRINT
3050 RETURN

```

PROGRAMA 5

```

10 REM PROGRAMA V
20 CLS
30 INPUT "PRIMER OPERANDO: ",primer
40 INPUT "SEGUNDO OPERANDO: ",segundo
50 GOSUB 1000
60 GOSUB 2000
70 GOSUB 3000
80 GOSUB 4000
90 INPUT "¿AS CALCULOS?";s#
100 IF s#="" THEN GOTO 20 ELSE END
1000 REM Rutina SUMA

```

```

1010 PRINT
1020 suma=primer+segundo
1030 PRINT primer;"+";segundo;"=";suma
1040 PRINT
1050 RETURN
1060 REM Rutina RESTA
2010 PRINT
2020 resta=primer-segundo
2030 PRINT primer;"-";segundo;"=";resta
2040 PRINT
2050 RETURN
3000 REM Rutina MULTIPLICACION
3010 PRINT
3020 multiplicacion=primer*segundo
3030 PRINT primer;"*";segundo;"=";multiplicacion
3040 PRINT
3050 RETURN
4000 REM Rutina DIVISION
4010 PRINT
4020 division=primer/segundo
4030 PRINT primer;" / ";segundo;"=";division
4040 PRINT
4050 RETURN

```

PROGRAMA 6

```

10 REM PROGRAMA VI
20 CLS
30 PRINT"COMENZAMOS EL PROGRAMA PRINCIPAL"
40 PRINT"SALTAMOS A LA Rutina DE LA LINEA 1000";GOSUB 1000
50 PRINT"YA ESTAMOS DE VUELTA"
60 PRINT"SALTAMOS A LA Rutina DE LA LINEA 2000";GOSUB 2000
70 PRINT"YA ESTAMOS DE VUELTA"
80 PRINT"FIN DEL PROGRAMA PRINCIPAL"
90 PRINT
100 END
1000 REM Rutina NUMERO 1
1010 PRINT
1020 PRINT"***BIENVENIDO A LA Rutina NUMERO 1"
1030 PRINT"***SALTAMOS A Rutina 3"
1040 PRINT"***VOLVEMOS A LA Rutina NUMERO 1"
1050 PRINT"***FIN DE LA Rutina NUMERO 1"
1060 PRINT
1070 RETURN
2000 REM Rutina NUMERO 2
2010 PRINT
2020 PRINT"***BIENVENIDO A LA Rutina NUMERO 2"
2030 PRINT"***SALTAMOS A Rutina 3"
2040 PRINT"***VOLVEMOS A LA Rutina NUMERO 2"
2050 PRINT"***FIN DE LA Rutina NUMERO 2"
2060 PRINT
2070 RETURN
3000 REM Rutina NUMERO 3
3010 PRINT"***BIENVENIDO A LA Rutina NUMERO 3"
3020 PRINT"***FIN DE LA Rutina NUMERO 3"
3030 RETURN

```

GANANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

Porque pretendemos que **AMSTRAD SEMANAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estas páginas, en las que podréis aportar ideas y programas interesantes para otros lectores.

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.

- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:

- Descripción general del programa.

- Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.

- Instrucciones de manejo.

- Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.

- No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.

- En una sola cinta puede introducirse más de un programa.

- Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000** pesetas, en concepto de derechos de autor.

- Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

- **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.

- Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.

- Los programas sospechosos de plagio serán eliminados inmediatamente.

¡ENVIAMOS TU PROGRAMA!

Adjuntando las siguientes datos:
Nombre y apellidos,
dirección y teléfono.

Indicando claramente en el sobre:

AMSTRAD SEMANAL
a HOBBY PRESS, S. A. La Granja, 39
Pol. Ind. Alcobendas (Madrid)

LAS VARIABLES, CLAVE DE LA PROGRAMACION

la semana pasada tuvimos ocasión de escribir y ejecutar nuestros propios programas, haciendo uso masivo de la sentencia Basic más sencilla y obvia de todas: PRINT.



in embargo, el método de programación escogido fue muy primitivo y, por tanto, muy fácil de comprender. Vamos a intentar, paso a paso, añadir a nuestros programas un poco más de sofisticación, introduciendo el CONCEPTO FUNDAMENTAL DE VARIABLE.

Aunque la salida, el OUTPUT, de los programas que veremos no sea tal vez muy espectacular, no le quepa duda de que se avanza derecho hacia la comprensión del Basic.

Los programas son secuenciales

Recordemos solamente un concepto esencial más: un programa Basic consiste en una secuencia de instrucciones agrupadas en líneas, cada una etiquetada con su correspondiente número.

Para introducir una en el programa, basta con escribir el número de línea que deseemos que tenga, seguido de las apropiadas PALABRAS-CLAVE (comandos) de Basic y pulsar (ENTER).

Para ver todas las instrucciones del programa:

LIST (ENTER)

Para ejecutarlo:

RUN (ENTER)

Y, por fin, para deshacernos de él y limpiar la memoria antes de introducir un nuevo programa:

NEW (ENTER)

Echemos una mirada al programa número 1, nuestro viejo conocido. Bien, el programa número 2 es otra forma de obtener la misma salida por pantalla de manera mucho más elegante.

Como ya sabemos, las palabras entrecuilladas se denominan STRINGS, palabra inglesa que estimamos preferible a la traducción española «tiras», aunque hablar de tiras de caracteres recuerda cómo el Amstrad las interpreta, esto es, de forma serial. Para el ordenador, CASA es la serie de caracteres C, A, S, A. Nunca son tratados, aunque a nosotros nos parezca lo contrario al verlo impreso en la pantalla, como un todo.

Lógicamente, nos puede interesar escribir una misma frase entrecuillada muchas veces a lo largo de una aplicación.

Por ejemplo, en una carta de negocios hay frases casi standard, que se repiten a menudo en esa misma carta o en otras.

Las variables ahorran esfuerzo y espacio

El Amstrad resuelve el problema asignado a cada cadena una etiqueta, de forma que al «invocarlas» en un programa, la cadena de caracteres correspondiente aparece en la pantalla.

Así, en la línea 10 del programa número 2, la etiqueta «A\$», se asigna a la cadena PROGRAMAR, de tal forma que cuando el ordenador se encuentra con la línea 40 del programa:

40 PRINT A\$

la palabra PROGRAMAR aparece en la pantalla.

La ventaja de esta técnica es la gran cantidad de espacio de memoria del ordenador y de esfuerzo nuestro que se ahorra al escribir los programas.

En efecto, basta pensar por un momento en que la palabra representada por A\$ tuviera que aparecer 100 veces en la salida del programa por pantalla.

Análogamente hacemos con las líneas 20 y 30 del programa 2; éstas causan la salida por pantalla deseada cuando se ejecutan las líneas 50 y 60.

En este momento, hay que hacer notas los siguientes puntos:

Primeros repastos

1. Hemos elegido las etiquetas de modo tal que todas comienzan por una letra y van seguidas del signo «\$».

No estamos obligados a usar una sola letra, podemos poner más, pero SI HAY QUE PONER EL SIGNO «\$», para avisar al ordenador de que queremos etiquetar una cadena de caracteres (veremos posteriormente cómo etiquetar otro tipo de «cosas»).

2. El hecho de que en el programa 2 las etiquetas se encuentren en orden alfabético es completamente arbitrario. NO TIENE POR QUE SEGUIR NINGUN TIPO DE ORDEN.

3. Aunque hemos usado un signo «=» para relacionar etiqueta con objeto, su significado no debe entenderse como «igual a...», sino más bien como «asignada a A\$ la cadena entrecuillada que viene después del signo =».

4. La etiqueta debe estar en la parte izquierda de la SENTENCIA DE ASIGNACION (que así se llama), y lo que se nombra en la parte derecha del signo igual.

Pruébese si no, con la sentencia:

10 «PROGRAMAR»=A\$

y ya se verá lo que ocurre. El pobre Amstrad ni se enteraría de lo que queremos decirle.

5. La etiqueta reemplaza a la palabra entrecuillada y a las propias comillas, ya que, al decir:

40 PRINT A\$

las comillas no aparecen.

6. El ordenador no distingue entre mayúsculas y minúsculas en el caso de nombres de etiquetas. En lo que a él respecta, los programas 2 y 3 son idénticos y producen el mismo resultado.

Esto, que puede parecer obvio, depende del ordenador del que se trate. Otras máquinas son muy rígidas en ello.

PROGRAMAS

```
5 REM PROGRAMA II
10 A$="PROGRAMAR"
20 B$="ES"
30 C$="FACIL"
40 PRINT A$
50 PRINT B$
60 PRINT C$
```

```
5 REM PROGRAMA III
10 A$="PROGRAMAR"
20 B$="ES"
30 C$="FACIL"
40 PRINT A$
50 PRINT B$
60 PRINT C$
```

& Juegos ESTRATEGIA

¡Juegos de Simulación Estratégica para poner a prueba su inteligencia y vivir la emoción de situaciones reales!

Un WAR GAME de estrategia que reúne todos los condimentos necesarios para hacer de él un auténtico «plato fuerte», inteligencia, emoción, sorpresa y realismo.

Sólo 1.125 pts.
(I.V.A. incluido)

RATAS del DESIERTO

RATAS del DESIERTO

Operación Norte de Africa

(Desde uno o tres jugadores)

Disponible para

Spectrum

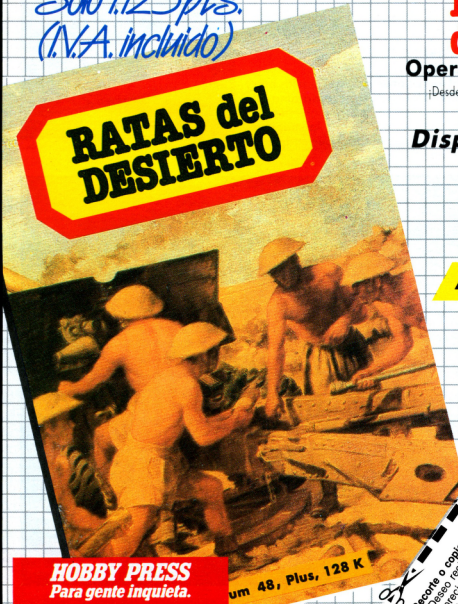
y

Amstrad

HOBBY PRESS
Para gente inquieta.

um 48, Plus, 128 K

Recorte o copie este cupón y envíelo a Hobby Press, S. A. Avdo. de Conchos 54, Log. Madrid.
 Deseo recibir en mi domicilio, sin gastos de envío alguno por mi parte, la cinta **RATAS DEL DESIERTO** al precio recibo de 1.125 pts. (I.V.A. incluido).
 Nombre: _____
 Dirección: _____
 Localidad: _____
 Código Postal: _____
 Provincia: _____
 Teléfono: _____
 Efectuar el pago en: Amstrad Spectrum Gira de pago Gira postal n.º _____
 Fecha de caducidad de la tarjeta: _____
 Fecha y firma: _____



AJUSTE POR MINIMOS CUADRADOS

Programa realizado por el lector:
José de la Torre Moreno

Todo el mundo sabe que un ordenador es la mejor calculadora del mundo, y que, además, permite hacer muchas más cosas «extrás», incluso en el caso de una tarea tan compleja y tediosa de calcular como el ajuste de datos experimentales por el método de mínimos cuadrados.



unque el título pueda sonar a «chino» a los profanos en el mundo de las Matemáticas avanzadas, no es necesario ni muchísimo menos conocer la teoría correspondiente para comprender intuitivamente la utilidad del programa (si bien está pensado para ser útil a estudiantes de un nivel de estudio medio-alto).

Muchos nos hemos encontrado más de una vez con una serie más o menos extensa de puntos (recordemos: parejas de valores $[x, y]$) que, dibujados en sus correspondientes ejes, parecían ajustarse en mayor o menor grado a una RECTA. Pues bien, el programa obtiene la ecuación de esa recta que se ajusta lo mejor posible a esos puntos.

En realidad, el programa es algo más que el simple cálculo de la ecuación, pues con ella, podemos realizar más cosas, a la completa elección del usuario, tales como obtener orígenes, obtener imágenes, representar los puntos y la recta..., y todo con la comodidad propia del «trabajo hecho por el ordenador»: sólo hay que apretar la tecla de opción elegida.

El programa funciona con dos ME-

NUS, en donde se especifican las opciones disponibles, lo cual hace las INSTRUCCIONES de manejo innecesarias.

Las opciones incluidas son las más «inmediatas»; pero el programa está estructurado de forma que se pueda ampliar todo lo que la imaginación y las necesidades que cada uno exijan.

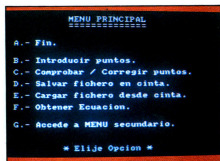


TABLA DE SUBRUTINAS

Líneas y letra del menú

Función

- 120-190 Constituye el programa principal que gobierna el Menú Principal. Dirige la acción a cada una de las subrutinas incluidas en el menú.
- 210 A Fin del programa. Restaura el ordenador a sus colores originales.
- 230-370 B Opción de introducir puntos. Se piden por parejas de valores separados por una coma. La primera vez que se accede a esta opción, es directa. Las demás se pregunta previamente por la naturaleza de los puntos que se van a introducir. (Ver Subrutina 2830)
- 390-610 C Permite corregir los puntos previamente introducidos. Se introduce el n.º y las coordenadas del punto correctas, separadas por una coma. El punto n.º «0» sale de la opción.
- 630-800 D Salva el fichero en cinta. El fichero se «etiqueta» con un número. La opción se puede cancelar pulsando «0» o simplemente pulsando (ENTER).
- 820-1000 E Carga un fichero desde cinta llamando al n.º con que se etiquetó. «0» o ENTER cancelan la operación de carga.
- 1040-1120 Constituye el programa secundario que gobierna el Menú Secundario.
- 1140-1520 F Es la parte central de todo el PROGRAMA. Se encarga de procesar todos los puntos introducidos y obtener la ecuación teórica que más se ajuste a la distribución de los mismos. Calcula también los errores de los coeficientes y el coeficiente de correlación que da idea de la bondad del ajuste.
- 1540-1650 I Calcula las imágenes teóricas de los puntos experimentales mediante la ecuación obtenida, y las compara con las introducidas, mostrando la diferencia de ambas.
- 1670-1790 J Obtener imágenes. Permite al usuario obtener todas las imágenes TEORICAS que necesite, usando la ecuación obtenida.
- 1810-1940 K Obtener orígenes. El usuario obtiene los orígenes TEORICOS de los valores introducidos.
- 1960-2070 L Muestra en pantalla los valores de todos los parámetros usados en el cálculo de la ecuación, tales como la Sumatoria de los X, Sumatoria de los Y, etc., así como el número de puntos introducidos y la Ecuación obtenida.
- 2090-2160 M Es la cartúla del Menú Principal.
- 2180-2240 G Es la cartúla del Menú Secundario.
- 2260-2310 Espera a que se pulse «M» para volver al menú correspondiente desde cualquiera de las operaciones.
- 2330-2550 Pregunta si los puntos que se van a introducir pertenecen a la misma recta (en cuyo caso dimensiona una nueva matriz y se introducen los nuevos puntos), o bien si pertenecen a un problema diferente, con lo que se borran todos los puntos introducidos previamente.
- 2570-2860 Inicio del programa. Gobierna la cartúla inicial.
- 2890-3130 H Es la representación Gráfica tanto de los puntos experimentales introducidos, como de la recta que corresponde a la ecuación teórica calculada previamente. Muestra los valores de los extremos de los ejes que corresponden a la zona donde se encuentran los puntos experimentales introducidos.

Serie Oro

```

1 * JOSE DE LA TORRE MORENO
2 * C/ CARRERA 38 CAUCINA (GRANADA)
3
4 DEF FNy(x)=a*x+b
5
6 * =====
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197 *
198 *
199 *
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *

```

```

600 NEXT
610 GOSUB 2090: RETURN
620 * == SALVAR FICHERO EN CINTA ==
==
450 MODE 1: BORDER 1: PEN 1
640 LOCATE 3,24: PAPER 3: PEN 2: PR
INT "Pulsa = 0 = para volver al M
ENÚ: PAPER 0: PEN 1
450 LOCATE 4,2: PRINT "E A L V A R
F I C H E R O *": PRINT: PRINT TA
B(6): STRING$(29,"*")
460 PRINT: PRINT: PRINT
460 INPUT * NÚMERO del fichero
"i n%
480 IF n#0 THEN FOR p=1 TO 4: PRI
NT: NEXT: PRINT TAB(11,1) "A H C E
L A D O": SOUND 1,1000,20,7: FOR p
=1 TO 2000: NEXT: GOSUB 2090: RETU
RN
490 LOCATE 1,24: PRINT SPACE$(39):
LOCATE 1,15
700 n%=STR$(n%)
710 OPENOUT n%
720 PRINT #9, n%
730 PRINT #9,n
740 FOR i=1 TO n
750 PRINT #9, p(i),p(i),2)
760 NEXT
770 CLOSEOUT
780 CLS: LOCATE 12,15: PRINT "Fiche
ro Salvado":FOR re=1 TO 1500: NEXT
790 GOSUB 2090
800 RETURN
810 * == CARGAR FICHERO DESDE CINT
A ==
620 MODE 1: BORDER 1: PEN 1
830 (LOCATE 3,24: PAPER 3: PEN 2: PR
INT "Pulsa = 0 = para volver al M
ENÚ: PAPER 0: PEN 1
840 LOCATE 4,2: PRINT "C A R G A R
F I C H E R O *": PRINT: PRINT TA
B(6): STRING$(29,"*")
850 PRINT: PRINT: PRINT
860 INPUT * NÚMERO del fichero
"i n%
870 IF n#0 THEN FOR p=1 TO 4: PRI
NT: NEXT: PRINT TAB(11,1) "A H C E
L A D O": SOUND 1,1000,20,7: FOR p
=1 TO 2000: NEXT: GOSUB 2090: RETU
RN
880 LOCATE 1,24: PRINT SPACE$(39):
LOCATE 1,15
890 n%=STR$(n%)
900 ERASE p: OPENIN n%
910 INPUT #9, n%
920 INPUT #9, n: DIM p(n,2)
930 FOR i=1 TO n
940 INPUT #9, p(i),p(i),2)
950 NEXT
960 CLOSEIN
970 CLS: LOCATE 12,15: PRINT "Fiche
ro Cargado":FOR re=1 TO 1500: NEXT
980 ==
990 GOSUB 2090
1000 RETURN
1010 * == MENÚ SECUNDARIO ==
1020 GOSUB 2180
1030 * == PROGRAMA SECUNDARIO ==
1040 bs="ahjklm"
1050 vs=INKEY$
==
1070 v% = LOWER$(v%)
1080 v=INSTR$(bs,v%)
1090 IF v#0 THEN 1050
1100 DN v: GOSUB 218,2890,1540,1670,
1810,1940,2090
1110 IF v#7 THEN GOTO 120
1120 GOTO 1050
1130 * == CALCULO DE LA ECUACION ==
==
1140 MODE 2: BORDER 0: PEN 1
1150 sx=0: sy=0: sz=0: sxy=0: sxm=
0: syem=0: scxm=0: scym=0: svxm=0
1160 * == Calculos previos ==
--
1170 FOR i=1 TO n
1180 sx=sx+p(i,1) : sy=sy+p(i,2) :
sz=sz+2*p(i,1)^2 : sxy=sxy+p(i,1)*p
(i,2)
1190 NEXT
1200 csx=sz^2 : csy=sz : scxm=s/n
: yem=sy/n
1210 FOR i=1 TO n
1220 scxm=scxm+p(i,1)*m : syem=syem
+p(i,2)*em : scym=scym+p(i,1)*p
(i,2)
1230 scym=scym+p(i,2)*yem^2 : scym
=scym+p(i,1)*m*em+p(i,2)*ym

```

```

1240 NEXT
1250 * == Calculo de las raices ==
==
1260 r1=(csx+m)/(n-2) : r2=(csy+2*intx2-
csx) : r3=(m/(n-2)*csy)
1270 IF r1<0 OR r2<0 OR r3<0 THEN G
OTO 1490
1280 * == Calculo de los Coefici
entes ==
1290 sqv=sQR(r1) : sqb=sqy+sDR(r2)
: sqm=sqy+sDR(r3)
1300 ax=(n*sqv-sqm)/(n*sq2-csx) :
bx=(2*sqv-sqy+sqm)/(n*sq2-csx)
1310 r4=scxm*scym : IF r4<0 GOTO 14
90
1320 re=sqv/sDR(r4)
1330 * == Resultados ==
1340 CLS
1350 LOCATE 28,1: PRINT "Número de
puntos: "n
1360 LOCATE 28,2: PRINT STRING$(b15
,"*")
1370 LOCATE 3,6: PRINT "Coeficiente
A = "r4
1380 LOCATE 45,6:PRINT "Coeficient
e B = "r3
1390 LOCATE 3,8:PRINT CHR$(185);"A
" : "sqg

```

```

1400 LOCATE 45,8:PRINT CHR$(185);"B
" : "sqb
1410 LOCATE 17,15: PRINT "Coeficien
te de corrección: "r1
1420 IF b#0 THEN sign=43 ELSE sig
n=45
1430 LOCATE 16,21: PRINT "La recta
es: y = "r4x" + "r3"CHR$(sign)AB
$(b)
1440 LOCATE 16,20: PRINT STRING$(14
,"*");PRINT " : "; PRINT STRING$(3
,"*")
1450 LOCATE 16,22: PRINT STRING$(14
,"*");PRINT " : "; PRINT STRING$(3
,"*")
1460 REM DEF FNy(x)=a+b
1470 GOTO 1510
1480 * == Posibles errores ==
--
1490 LOCATE 30,12: PRINT "HAY UNA R
AZA NEGATIVA"
1500 LOCATE 40,13: PRINT "o": LOCAT
E 70,14: PRINT "UNA DIVISION POR CE
RO"
1510 GOSUB 2260
1520 RETURN
1530 * == COMPARAR DATOS EXPERIMEN
TALES CON TEÓRICOS ==

```

TABLA DE VARIABLES USADAS

P	Matriz principal. En ella se almacenan los puntos que se introducen.
Pc	Matriz auxiliar. En ella se realizan las correcciones de los puntos.
q	Matriz de trabajo. Con ella se ordenan los puntos en la opción de representación gráfica, y hace de puente para poder ampliar la dimensión de P cuando se desean introducir nuevos puntos en la misma recta. Es el número de puntos que se desean agregar al problema en curso. Sirve para ampliar la dimensión de P.
ng	Número de la "Etiqueta" del fichero que se desea crear o cargar.
m%	Contador que gobierna la introducción de puntos, presentándonos cuando es necesario la pregunta de si los puntos pertenecen a la misma recta con la que estamos trabajando o bien se trata de un problema nuevo.
w	Sirve para seleccionar el programa Principal (caso de que accedamos desde sus opciones) o al programa Secundario (si accedemos desde una de sus opciones). Se encuentra en la subrutina de "Espera".
op	Número de puntos introducidos. Constituye la dimensión de P.
n	Contiene las opciones del menú principal. Opción seleccionada en el Menú Principal.
a\$, t\$, i\$, s	Contiene las opciones del Menú Secundario. Opción seleccionada en el Menú Secundario.
b\$, v\$, v	Valor literal de m%. Es la etiqueta de los ficheros. Origen introducido en la opción obtener imágenes. Se calcula su imagen mediante la función FNy(x).
n\$, n	Imagen introducida en la opción obtener orígenes. Se calcula su origen mediante la función FNx(y).
x\$, y\$, e\$, i\$, s	Usadas como INKEY\$, esperan a que se pulse una tecla.
1\$	Letra de los datos, usadas para la carátula de presentación.
i, j, k, g, re, pr	Subíndices de uso general.
exl, ex2	Extremos de la recta teórica dibujada.
xmax, ymax, xmin, ymin	Extremos de los ejes de la representación gráfica.
sx, sy, sx ² , sxy, sxm, sym, scxm, scym, scx, cxy, xm, ym	Cálculos para la obtención de la ecuación ajustada a los puntos experimentales.

El comando EVERY

ANÁLISIS

Todos los ordenadores contienen un microprocesador regulado por cuarzo, que controla el tiempo de modo real. AMSTRAD Análisis estudia hoy un comando que nos permite utilizar esta característica para realizar una tarea cada cierto tiempo exactamente: El comando EVERY.

Autor: Fco. Javier Barceló T.



El ejemplo presentado consiste en una subrutina que presenta un reloj en la pantalla. Se puede incluir en cualquier otro programa, haciendo que este otro empiece a partir de la línea 210. El único efecto negativo será que resulte un poco más lento.

Veamos el programa:

20-50: Comentarios.

60: Borra la pantalla, y establece el modo de pantalla.

70: Pregunta la hora.

80: Comprueba que la hora dada no sea mayor que 24, ni menor que 0.

90: Pregunta los minutos.

100: Comprueba que los minutos dados estén entre 0 y 59.

100-120: Realiza lo mismo para los segundos.

130: Detiene el programa hasta que se pulse ENTER, y cuando se pulsa empieza a correr el tiempo.

140: Borra la pantalla.

150-180: Dibujan el borde del reloj en la pantalla.

190: Comentarios.

200: A partir del momento que el programa ejecuta esta línea, cada segundo el programa irá desde donde esté a la línea 230, y luego volverá donde estaba anteriormente.

210-220: Bucle sin fin, para que el programa no se pare. Aquí empezaría el programa principal.

230: Comentarios.

240: Suma uno a la variable S (segundos).

250: Comprueba si los segundos suman sesenta, y si es así lo pone a cero, y suma uno a los minutos.

260: Comprueba si los minutos suman sesenta, y si es así pone a cero minutos y segundos, sumando uno a la hora.

270: Muestra la hora en la posición de pantalla especificada. Al poner el signo X se in-

vierten los colores de papel tinto en la pantalla. **280:** Retorna al programa principal, en el punto donde lo dejó.

Este comando va seguido de un número, que indica cada cuánto tiempo va a hacer el GOSUB a la línea indicada. En este caso será cada segundo ($50 \times 0.02 = 1$), cuando irá a imprimir la hora. Cuando transcurre el período de tiempo, el ordenador deja la tarea que estaba haciendo para ejecutar la subrutina, y cuando acaba ésta (RETURN) vuelve a la tarea anterior y en el mismo punto donde la dejó.



Para que tus dedos se realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los datos que se captan este lenguaje se encuentran a la disposición en un cassette memoria, calcitrante.

Amsoft Serie Oro

La mejor selección de juegos para AMSTRAD

CAMPEONES DEL MUNDO DE RALLIES



Participa en uno de los deportes más populares del mundo. Juega contra el ordenador (Jugar fuerte), o contra tus amigos, quizá, más fáciles de vencer.
CASSETTE Y DISCO.

FUTBOL



Disfruta la emoción de uno de los deportes más populares del mundo. Juega contra el ordenador (Jugar fuerte), o contra tus amigos, quizá, más fáciles de vencer.
CASSETTE Y DISCO.

3D BOXING



Participa en uno de los deportes más extenuantes gracias a esta magnífica simulación gráfica tridimensional. Enfrentate a los mejores pugiles: MAD JOE, QUASI y ROLAND.
CASSETTE Y DISCO.

RAID



¡Detén un ataque nuclear asaltando el Centro de Defensa Soviética! Un juego de acción de múltiples pantallas y diferentes niveles de destreza.
DISCO.

3D GRAND PRIX



Compíte en una carrera de Fórmula 1, en uno de los 8 circuitos internacionales. Guía tu prototipo, acelerando, frenando y cambiando de marchas, mientras tus competidores te pisan los talones.
CASSETTE Y DISCO.

SUPERTRIPPER



Tú, Supertripper, has de buscar los 28 disquetes desperdigados por el planeta Khun, y salir de allí. Con ayuda de los globos escapa de los aborígenes que te debilitan en los encuentros.
CASSETTE Y DISCO.

SORCERY PLUS



Lucha en busca de los Sorcerers. Sólo si liberas a todos podrás derrotar a los Necromancers. Encontrarás objetos que te servirán de ayuda o confusión. Descubre los pasadizos secretos, y disfruta de uno de los mejores juegos de aventuras de todos los tiempos.
DISCO.

AJEDREZ TRIDIMENSIONAL



Para jugadores de cualquier nivel. Proporciona numerosas posibilidades: repetición de movimientos, ver la partida desde el principio, análisis de posiciones, estudio del desarrollo completo de una partida, tablero tridimensional y convencional, etc.
CASSETTE Y DISCO.

P.V.P.

CASSETTE

2.300 pts.;

DISCO

3.000 pts.

AMSTRAD ESPAÑA

Avda. del Mediterráneo, 9. Tels. 433 45 48 - 433 48 76. 28007 MADRID
Delegación Cataluña: Tarragona, 110 - Tel. 325 10 58. 08015 BARCELONA

NIGHT SHADE

Las fuerzas del mal, en una terrible noche, cayeron sobre una pacífica villa; ninguno de los habitantes pudo resistir las plagas y todos fueron pronto absorbidos por los malignos influjos.

P

oco a poco, los moradores se transformaron en horrendas criaturas, dedicadas a apoderarse de cualquier humano que osase atravesar los límites de la ciudad.

Como la palabra corre más rápido que cualquier corcel, se extendió el rumor de la posesión demoniaca de la villa y ésta se convirtió en un reto para cualquier aventurero, ansioso de fama y riquezas.

Pronto comenzaron a correr cuentos y leyendas, sobre los intrépidos aventureros que se adentraban en las calles de la ciudad y jamás regresaban.



Un viejo anciano recorre el país de punta a punta, intentando atraer a nuevos aventureros que puedan desterrar a las fuerzas del mal de los límites de la villa.

La historia contada siempre es la misma y sus ojos se llenan de lágrimas, cuando relata cómo la ciudad fue poseída por las fuerzas diabólicas.

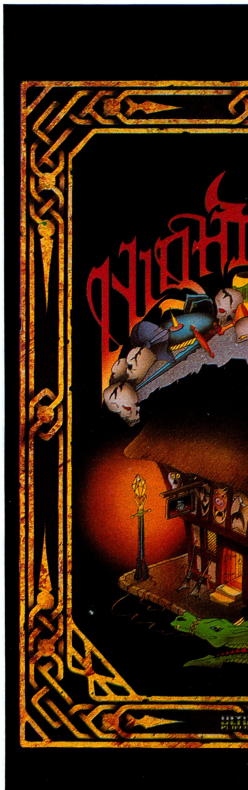
Los cuentos de fantasmas y de ermitaños embrujados por las fuerzas satánicas para que les obedezcan, y de las plagas y demonios que circulan por las calles de la villa, andan en labios de todos.

La misma muerte se encuentra airapada dentro del pueblo, por ser considerada como un visitante inoportuno para el mismo diablo.

Los esqueletos, acostumbrados a los ríos de sangre putrefacta de los muertos, acechan a las posibles víctimas que osen traspasar los límites de la ciudad.

Espantosos demonios y terribles hechizos, esperan para poder absorber a quién ose retar al poder satánico. Expulsale a las tinieblas, y el reino del poder de las sombras será tuyo.

Con este prólogo de horror y poderes sobrenaturales, el retador del reino del mal se acerca a las puertas de la ciudad.



Compatible: CPC1464, CPC1664 y CPC16128





Siguiendo la estela de Knight Lore y Alien 8, la última producción de Ultimate, continúa fiel al sello de la casa.

Utilizando el mismo método de representación tridimensional que en los anteriores. **Night Shade** se desarrolla en una ciudad amurallada, con calles, plazas, casas y una grandiosa catedral, admiración de todos sus moradores.

La diferencia fundamental con los anteriores programas de la casa radica en las técnicas de movimiento.

En Knight Lore y Alien 8, el protagonista se movía por la pantalla permaneciendo fijo el decorado de las habitaciones.

Ahora aquél permanece en el centro de la misma, siendo el fondo lo que se traslada en las distintas direcciones.

Con esta técnica siempre podemos observar con claridad lo que nos espera en nuestro recorrido por las calles y encrucijadas de la ciudad.

Como innovación principal, debemos reseñar que cuando entramos en cualquier casa o edificio, los muros se hacen transparentes, permitiéndonos ver el interior sin cambiar de pantalla.

Nuestro personaje es capaz de recolectar armas y hechizos distribuidos por las casas y calles, y de arrojarlos sobre las criaturas hostiles que nos amenazan.

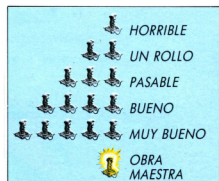
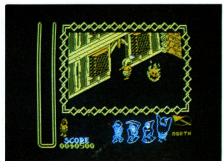
Los enemigos que debemos eliminar son: la muerte, el ermitaño, el fantasma y el esqueleto. Cada uno de ellos solamente puede ser aniquilado por un objeto que debemos localizar en la ciudad, la muerte de los cuatro poderes maléficos, nos hará culminar con éxito la aventura y haremos poseedores del inmenso poder de las sombras.

La utilización de un mapa para movernos por las calles y casas de la ciudad será de gran ayuda para cualquier jugador, dado el intrincado trazado de la misma.

El juego se realiza en una parte reducida de la pantalla, dejando los bordes izquierdo e inferior para información, personajes y marcador, siendo una verdadera pena que no se haya utilizado la pantalla al completo, constituyendo el único detalle en contra, puesto que el área de juego ocupa solamente el 60 por 100 de la pantalla y esto es desaprovechar demasiado.

Un juego típico de Ultimate, con buenos gráficos y larga duración.

Mr. Joystick



STAR MOUSE

Si deseamos crear una pantalla para su posterior utilización en un programa, con el paquete STAR MOUSE podemos realizarla con gran facilidad y rapidez.

El paquete en cuestión contiene un disco con el software necesario para la creación de pantallas, y un aparato que a partir de este momento denominaremos RATON (traducción castellana de la palabra MOUSE).



P

Para la utilización del programa y una vez cargado en el ordenador, apreciaremos la presencia de una ventana a la izquierda del monitor, que podremos desplazar de posición de izquierda a derecha a nuestra conveniencia, para así realizar completamente el dibujo de toda la pantalla sin ninguna limitación.

Una de las diferencias con otros programas de dibujo estriba sobre todo en la no aparición de mensajes a la hora de trabajar con él, normalmente en la aparición de mensajes se indica si se está desplazando o pintando, así como la posición en coordenadas del puntero guía.

Pero todo esto no es un inconveniente en ningún caso, pues para ello los programas que asisten a los distintos ratones existentes en el mercado incorporan una serie de iconos que nos representan cómo estamos trabajando.

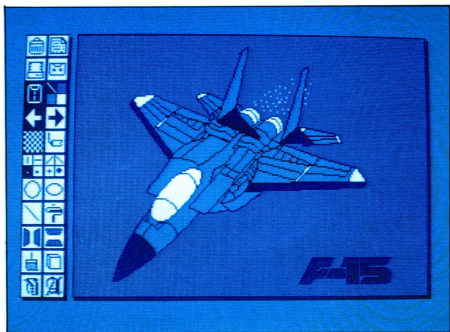
¿Qué son los iconos?

Los iconos son unos gráficos que representan herramientas de trabajo utilizadas normalmente por los dibujantes, y otros que siempre nos dan una idea de la función que realizan dentro del programa.

Tomando como referencia la ventana de iconos nada más terminar la carga, vamos a explicar qué representan los iconos, para qué sirven y cómo manejarlos de una manera más eficaz.

El primer icono representa un cubo de basura y su utilización nos permite la limpieza de la pantalla, o lo que es lo mismo, borra la pantalla y la deja lista para trabajar con ella sin ningún trazo dentro de ella. Para su utilización pondremos la flecha que guiamos con el ratón encima de dicho gráfico, y pulsaremos la tecla de ejecución que posee el ratón en su parte posterior. Si utilizamos este icono debemos tener especial cuidado de la ejecución estando la flecha encima del cubo, nos desaparecerá el dibujo inmediatamente, con los trastornos que esto nos puede causar.

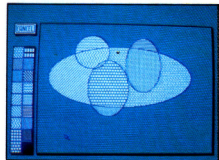
Al lado de este icono aparece una



especie de hoja doblada por una esquina: representa la opción de fichero y, al posicionar la flecha sobre ésta y pulsar el botón, el dibujo que tenemos en ese momento en pantalla se archiva momentáneamente en memoria y aparece un pequeño menú entre los que podremos elegir grabar o cargar una pantalla, saber el contenido de un disco a través de su directorio, borrar una pantalla del disco o retornar al dibujo tal como lo habíamos dejado.

El siguiente icono representa un ordenador y su pantalla. Al escoger esta opción desaparecerá la ventana de información, permitiendo la visualización por completo del dibujo que estamos realizando.

Para poder trabajar con cassette o disco poseemos dos iconos que representan una cinta y un disco, con ellos elegiremos el soporte de archivo a utilizar con el programa. Para señalar cuál es el que estamos utilizando, el icono aparece en fondo oscuro y tinta clara. Una de las ventanas de poder disponer de disco y cinta es la de poder trasladar pantallas de una a otro soporte de manera cómoda.



Hasta cuatro colores

El modo de trabajo del programa es en modo 1, con lo que podemos utilizar hasta cuatro colores, que, aunque son pocos, permiten la realización de pantallas con suficiente claridad y resolución. Para cambiar de color de trabajo, disponemos de un icono dividido en cuatro partes iguales y al lado del icono de disco; indica el color que se dispone en cada momento, y se encuentra con una raya atravesándolo de esquina a esquina.

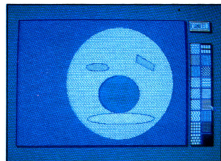
Debajo de estos iconos encontramos otros dos, representando unas flechas; éstas se orientan hacia la izquierda y la derecha; con ellas como anteriormente habíamos comentado desplazaremos la ventana de información de un lugar a otro de la pantalla según nuestra conveniencia, sin borrar el dibujo que se encuentre debajo y permitiéndonos dibujar en esa posición de la pantalla que antes se encontraba tapada.

A la hora de trabajar, la pantalla nos será de mucha utilidad poder cuadrícula con una rejilla de 8×8 , para facilitar el manejo de la misma y realizar un trabajo más rectilíneo y preciso. Para ello el icono en esta ocasión simula una hoja cuadrícula.

Si esforzamos un poco nuestra imaginación conseguiremos relacionar el icono que se encuentra a la derecha de la cuadrícula con un lápiz y una línea. Al elegir dicho icono, en el centro del monitor se observa la aparición de un rectángulo.

La utilización de este icono va aso-

Banco de pruebas



ciado a la utilización del icono de tirantes, y con él definiremos la frecuencia de trazo, o lo que es lo mismo, definiendo un trazo de línea de ocho pixel, y de éstos cuántos y en qué secuencia van a estar activados. Para definir el trazo utilizaremos el ratón desplazando éste arriba y abajo hasta lograr la serie deseada.

Podemos definir trazos y tramas...

Para definir el ancho del trazo a la hora de trabajar con la brocha o con las funciones de simetría, disponemos de dos iconos divididos a su vez en cuatro cada uno. Los trazos que disponemos son un punto del tamaño de un pixel, y otros tres más anchos, una barra vertical, otra horizontal y dos inclinados.

Con el programa podemos crear círculos del tamaño que deseemos, y para ello disponemos del icono de círculos. Al utilizar éste deberemos tener en cuenta que el programa necesita los dos extremos de la figura a realizar, obteniendo con ello un círculo del diámetro que hayamos señalado.

Además de poder crear círculos podemos realizar elipses, para ello debemos indicar con la flecha guía la parte inferior izquierda de la base de esta figura; notaremos la presencia al pulsar el botón de ejecución de un pequeño cuadro, desplazando el ratón hacia arriba y a la derecha delimitaremos la altura y el ancho de la misma.

Cuando tenemos que trazar una línea en lugar de realizarla a pulso,

debemos elegir la opción de tiralíneas que es el icono que representa una barra inclinada; para realizar una línea deberemos indicar el principio y el final de ésta. Anteriormente comentamos un icono con el que es posible definir lo que podríamos llamar el paso; por ejemplo, que en una línea se pinten dos pixel activados, dos no y que nos permita realizar líneas con distintas frecuencias de trazo.

...y rellenar cualquier figura con 18 tipos de entramados

Para rellenar una figura utilizaremos el icono destinado a esta función que es el que representa el rodillo; al elegir esta opción, desaparecerá la ventana de información del dibujo y en su lugar aparecerá otra en donde encontraremos los distintos tipos de fill que podemos utilizar; disponemos de 18 tipos distintos de entramados. Si deseamos introducir cualquiera de estos tipos en una figura debemos tener mucho cuidado de que la figura esté completamente cerrada, pues si no el fill continuará por toda la pantalla hasta encontrar un lugar en que se cierre; una vez tenido esto en cuenta, elijeremos la trama que mejor se acopla a nuestras necesidades, posicionando la flecha sobre ésta pulsaremos el botón del ratón y desplazaremos la flecha hacia el interior de la figura y volveremos a pulsar el botón.

Entre los iconos que disponemos encontraremos dos idénticas cualidades pero de realización distinta, y que representa una especie de ventana abierta. Con ayuda de éstos realizaremos figuras con forma simétricas en dos planos, vertical y horizontalmente. Una vez elegida la opción a realizar posicionaremos la flecha en el lugar elegido y desplazaremos el ratón por la pantalla, generando una figura como si utilizásemos un espejo para reflejarla.



STAR MOUSE

SOFTWARE LOADER

Programa creado para el diseño de pantallas en modo 1, permitiendo el uso hasta cuatro colores. Estas pantallas nos servirán para utilizarlas a modo de presentación, al igual que en los programas comerciales.

HARDWARE RATON

Este ratón no necesita nada más que conectarse al port del joystick, y a la corriente. El funcionamiento es mecánico y movido por una bola metálica.

APARATO	TIRA-LÍNEAS	PISTOLAS	BORRAR	BROCHA	FILL	DEF.FILL
STAR	SI	SI	SI	SI	SI	NO
CIRCULOS	ELIPSE	CUADRADO	TEXTO	JUEGOS CARACT.	SIMETRIA	
SI	SI	NO	SI	1	SI	
IMPRESORA	TRAZOS	RALLADO PANTALLA	COLOR	DEFINIR TRAZO	SENSIBILIDAD	
NO	8	SI	4	SI	1 SOLA	

Dibujar a pulso es una tarea bastante difícil con cualquier instrumento, para realizarla con ratón tenemos que escoger la opción de brocha. Con ella tendremos que elegir primero el trazo que se va a utilizar fino, ancho, redondo o cualquiera de los que disponemos. Después podremos realizar cualquier dibujo en pantalla como si utilizásemos un lápiz con distintos tipos de punta.

Este programa dispone de un icono destinado a 1 borrado de secciones o parte de la pantalla, éste representa una goma de borrar.

Cuando el dibujo está acabado y nos disponemos a colorearlo tendremos la oportunidad de comprobar que el icono de pistola, tiene un uso muy similar al de una de pintar, y que según repetimos el paso sobre una misma zona el pintado es mayor, donde la oportunidad de mezclar colores y cambiar los mismos.

También se puede incluir texto

Por último, para introducir texto en un dibujo podemos optar por el icono en forma de A mayúscula gótica. Al elegir esta opción en la pantalla aparece un cursor idéntico al que utiliza en Amstrad en el centro de la pantalla; debemos posicionar éste en el lugar donde se vaya a introducir el texto y después pulsar el botón del

ratón. Una vez hecha esta operación procederemos a introducir los caracteres con la ayuda del teclado del ordenador y al finalizar pulsaremos la tecla ENTER apareciendo la flecha guía sobre el icono de texto. Con esta opción no podemos introducir los caracteres que el ordenador dispone pero no están representados en teclado.

Cómo aprovechar Star Mouse en nuestros programas

Al trabajar con las pantallas creadas con ayuda del STAR MOUSE deberemos teclear, LOAD « < **NOMBRE** > », 49152, para la utilización en nuestros programas.

Un truco a tener en cuenta a la hora de empezar a crear una pantalla: puede ser de gran utilidad realizar con un color claro un fill que nos haga un encuadrado de pantalla, ya que la opción de cuadrícula no es fija y más vale borrar después de realizar el dibujo que hacerlo descentrado o torcido.

En el programa que la caso nos a suministrado y que según se nos a informado no es la versión definitiva que se introducirá en el paquete, hemos detectado algunos fallos que esperamos sean corregidos antes de su comercialización.

SI BUSCAS LO MEJOR

ERBE

Software

LO TIENE

SENSATIONAL SOFTWARE FROM
MIKRO·GEN



**Three Weeks
in
Paradise**

LA FAMILIA WALLY ATACA DE NUEVO EN LA MEJOR Y MAS DIVERTIDA AVENTURA QUE PUEDES IMAGINARTE. TODOS LOS PELIGROS Y COLORIDO DE LA SELVA EN TU ORDENADOR.

DURELL

SABOTEUR!



SPECTRUM disk
R.R.P.
£8.95

COMO EXPERIMENTADO MERCENARIO CUIDADOSAMENTE ENTRENADO EN ARTES MARCIALES, DEBES CUMPLIR LA MISION QUE TE HA SIDO ENCOMENDADA: ROBAR EL DISCO QUE CON LA LISTA DE LOS REBELDES TIENE EL GRAN DICTADOR.

DENTRO TONO... Y DE TONO A SUS MELODIAS

Ya hemos dado un paso en el conocimiento de las posibilidades sonoras del Amstrad. Sabemos usar el comando SOUND y dominamos la envolvente de volumen, el comando ENV.

Pero resulta que existe otra envolvente, la de tono, que vamos a estudiar profusamente en este artículo, el comando ENT.



Si ha seguido la serie hasta aquí, ya le resultará familiar nuestro viejo amigo:

SOUND 1,200,100,5

Esperamos que sea capaz de ver que esta orden dice al **Amstrad** que produzca un sonido por el canal A durante un segundo. El tono de la nota será 200 y su volumen 5.

Como recordará, el comando SOUND tiene la siguiente estructura:

SOUND canal, tono, duración, volumen y cambiando alguno de estos parámetros alteraremos el sonido resultante.

Sin embargo, la cosa no es tan simple. La semana pasada vimos que el volumen de la nota que se producía podía variarse mediante algo llamado envolvente de volumen.

Podemos tener hasta 15 de estas envolventes de volumen, definidas por el comando ENV y puestas en marcha añadiendo otro parámetro al final de nuestra instrucción básica SOUND.

De este modo, cambiando:

ENV 1,5,2,20

Y SOUND 1,200,100,5,1

conseguiremos una nota que dura un segundo incrementando su volumen mientras suena.

La estructura de un comando ENV es:

ENV N, P, Q, R

donde N indica el número de la envolvente, P da la cantidad de saltos o escalones que tiene, Q el cambio de volumen que se produce en cada escalón y R especificará la longitud de los mismos.

Pero todo esto no es tan sencillo como parece ya que vimos que el comando ENV podía tener hasta 16 parámetros de la siguiente forma:

ENV N, P1,Q1,R1,P2,Q2,R2,
P3,Q3,R3,P4,Q4,R4,
P5,Q5,R5

Esta ampliación en el número de parámetros se hacía para dotar de cinco etapas diferentes a la orden ENV. Por si no fuera ya suficiente, la envolvente de volumen no es la única que puede afectar a nuestro comando básico SOUND.

La envolvente de tono: el comando ENT

Hay otra, llamada envolvente de tono, que afecta al tono de la nota haciéndole alto o bajo-grave o agudo. Antes de ver cómo trabaja, vamos a «escucharlo» en acción.

Primero definimos una envolvente de tono con:

ENT 1,5,10,20

Después tecleamos:

SOUND 1,200,100,5,0,1

y pulsamos ENTER.

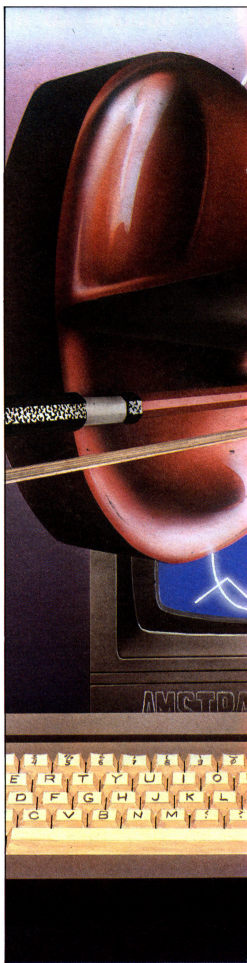
Si ha introducido las órdenes correctamente podrá escuchar un sonido, que dura un segundo, subiendo y bajando de tonalidad.

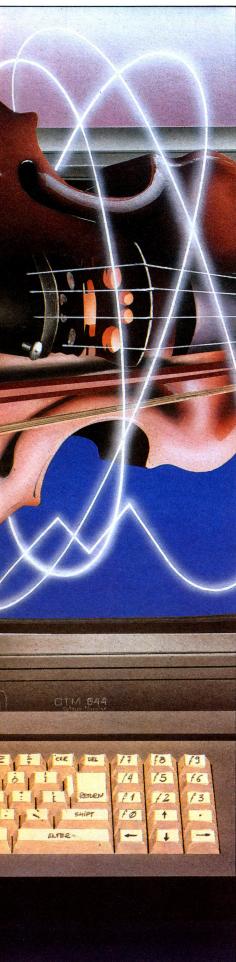
Lo que ocurre es que el 1 que está al final de comando SOUND ha llamado a la envolvente de tono señalada como la número 1. Esta envolvente, previamente definida, varía el tono por la instrucción SOUND de acuerdo con los parámetros que le hayamos dado.

Como podemos ver, la envolvente de tono tiene un aspecto muy similar al de la envolvente de volumen de la que nos ocupamos anteriormente.

Tiene la forma:

ENT S,T,V,W





No le habrá resultado muy difícil adivinar que S nos indica el número de la envolvente de tono. Podemos definir hasta 15, de modo que S tiene un rango de valores comprendidos entre 1 y 15. La envolvente 0 también existe pero equivale a hacer que la nota no cambie de tono.

Los parámetros de la envolvente de tono

Los parámetros T, V y W se asemejan, de nuevo, a los de la envolvente de volumen, pero en este caso influyen en la forma cómo debe variar la nota en «agudeza» o en «gravedad» del tono, no en cuanto al nivel de volumen.

El parámetro T nos dice el número de escalones en los que está dividida la nota asociada a la envolvente de tono. Puede tener valores comprendidos entre 0 y 239.

El parámetro V es el que nos indica cuánto debe variar el tono en cada escalón. Puede subir y bajar, al tomar valores entre -128 y 127.

Finalmente el parámetro W es la longitud —o duración— de cada escalón. La medimos en centésimas de segundo pudiendo valer desde 0 a 225. La Tabla II es una recopilación sobre los parámetros que necesita el comando ENT y de los valores que pueden tener dichos parámetros.

Ahora que ya conocemos qué es lo que significa cada uno de estos parámetros, veamos cómo trabajan sobre nuestro sonido preferido. La Figura 1 represento un diagrama de cómo varía el tono de la nota producida por:

SOUND 1,200,100,5

Como podemos observar, el tono se mantiene constante con un valor igual a 200 durante el segundo que dura la nota.

Vamos a definir ahora una envolvente de tono con la orden:

ENT 1,5,10,20

y la llamaremos con:

SOUND 1,200,100,5,0,1

Escucharemos cómo va descendiendo el tono del sonido cinco escalones durante el segundo que está sonando. La Figura II presenta los cinco escalones de la envolvente de tono de un modo gráfico.

Vamos a verlo detalladamente. El parámetro T es igual a 5, asegurando que habrá cinco escalones, mientras que el parámetro W, con un valor de 20, afirma que cada escalón durará el cuarto parte de un segundo.

Todavía con más detalle

El parámetro V es 10 y quiere decir que en cada escalón se suma 10 al valor de tono de la nota que está sonando.

En el caso de:

SOUND 1,200,100,5,0,1

Sonido

nos encontraremos que están sonando cinco notas con tonos de 210, 220, 230, 240 y 250. La envolvente coge el parámetro de tono del comando SOUND —200— y le va sumando 10 al llegar a cada escalón. Como el valor de dicho parámetro se va incrementando, la nota se hace cada vez más grave-baja.

Observe que el tono se incrementa desde el primer momento, el sonido comienza con un tono de 210 y no de 200 como cabría esperar. El efecto profusado por la envolvente de tono es inmediato. Dese cuenta también que un sencillo comando SOUND ha sido capaz de producir cinco notas gracias a la envolvente de tono que previamente hemos definido.

Anteriormente teníamos que usar cinco comandos SOUND para conseguir el mismo efecto, como en el Programa I.

Programa uno

Ahora, sin embargo, podemos obtener el mismo resultado definiendo una envolvente de tono con:

ENT 1,5,10,20

y llamarla usando:

SOUND 1,200,100,5,0,1

que es bastante más sencillo. Y la misma envolvente puede utilizarse para variar el tono de cualquier otra nota del mismo modo. Problemas:

SOUND 1,100,100,5,0,1

que llama a la misma envolvente pero comienza con un tono grave (110).

Hagamos una pequeña recopilación. Podemos definir una envolvente de tono usando el comando ENT. Cuando la llamamos, altera el tono del sonido producido por el comando SOUND.

Ambas envolventes pueden actuar simultáneamente

Por si todavía no está suficientemente asombrado le diremos que es posible encontrar ambas envolventes, de volumen y de tono, actuando al mismo tiempo. Pruébelo:

SOUND 1,200,100,5,1,1

y —a menos que hayamos borrado las envolventes de nuestro micro y sea necesario reponerlas— escucharemos cinco notas descendiendo tanto en volumen como en tono. Ambas envolventes pueden trabajar a la vez en perfecta armonía.

Como hemos dicho antes, podemos tener

hasta 15 envolventes de tono, de modo que vamos a definir otra con:

ENT 2,5-10,20

¿Podría adivinar su efecto antes de probarlo con el comando SOUND?

El parámetro T vale 5, de modo que tendrá cinco escalones. Ya que el parámetro W es 20, nos quiere decir que cada escalón durará 20 centésimas de segundo. El parámetro V es -10, por lo tanto el valor del parámetro de tono se decrementará en 10 por cada escalón de la envolvente.

Mientras que el valor del parámetro decrece, el tono de la nota, paradójicamente, aumenta. Así que habremos conseguido una nota que dura un segundo y que su tono se subirá cinco escalones. Llamaremos a la envolvente con:

SOUND 1,200,100,5,0,2

y lo escucharemos por nosotros mismos.

De nuevo, una sencilla envolvente ha producido cinco notas de diferentes tonos. Si no queremos hacer uso de la envolvente podemos recurrir a algo parecido al Programa II para alcanzar nuestros propósitos.

Programa dos

Como podemos ver:

SOUND 1,200,100,5,0,2

nos resulta mucho más cómodo.

Seguramente que le habrán llegado rumores de que la envolvente de tono espera que el sonido dure un cierto tiempo. Hasta ahora en nuestros ejemplos hemos usado comandos SOUND que han durado esa cantidad de tiempo. Supongamos que definimos una envolvente de tono con:

ENT 3,5,20,40

Como podemos ver por los parámetros T y W, la envolvente cuenta con que haya cinco escalones y que cada uno dure 40 centésimas de segundo.

Interacción de SOUND y ENT

Pero supongamos que el comando SOUND que invoca a la envolvente tiene un parámetro de duración de un segundo. En otras palabras, la duración del comando SOUND es menor que la asumida por la envolvente de tono. ¿Qué ocurre ahora?

Como la mayor parte de las cosas dentro de la Informática, la respuesta está en probarlo y verlo. Tecleando:

SOUND 1,200,100,5,0,3

encontraremos nuestra solución. El sonido sigue durando un segundo. La envolvente de tono sólo recorre dos escalones y medio antes de ser interrumpida.

SOUND 1,200,200,5,0,3

Tabla I: Rangos de parámetro del comando SOUND.

	Canal	Tono	Duración	Volumen		Envolvente de volumen	Envolvente de tono
				sin envolvente	con envolvente		
Rango	1 = A	0	1	0	0	0	0
	2 = B	A	A	A	A	A	A
	4 = C	4095	32767	7	15	15	15
Defecto	Ninguno	Ninguno	20	4	12	0	0

Tabla II: Rangos de parámetro del comando ENT.

Parámetro	Número S	Número de escalones en sección T	Cambio de tono por escalón V	Longitud de cada escalón W
Rango	0 A 15	0	-128	0
		A	A	A
		239	127	255

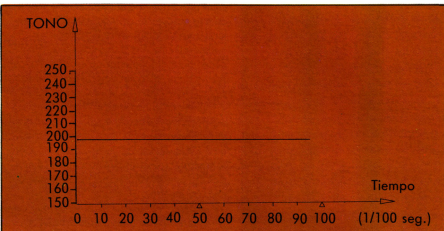


Figura I: SOUND 1,200,100,5.

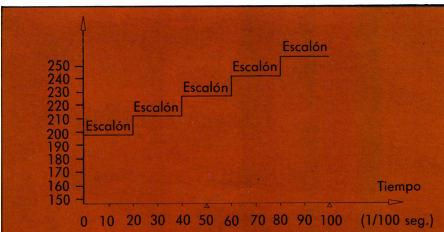


Figura II: SOUND 1,200,100,5,0,1.

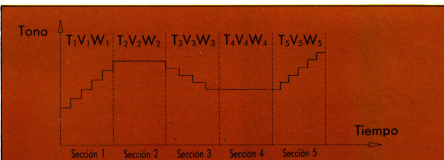


Figura III: Parámetros de las cinco secciones de una envolvente de tono.

que dura dos segundos, nos permitirá escuchar todos los efectos producidos por la envolvente.

Pero, **¿qué ocurre en el caso contrario, cuando la envolvente de tono dura menos tiempo que el comando SOUND?** Introduzcamos:

```
ENT 4,5,—10,10
```

que define una envolvente de tono que supone que va a durar medio segundo. Ahora llamamos a esta envolvente recién creada con:

```
SOUND 1,200,100,5,0,4
```

que deberá sonar durante un segundo.

Como podemos escuchar, el efecto de la envolvente de tono se aprecia durante medio segundo, en el que el tono de la nota aumenta. Durante el medio segundo restante la nota permanece con el tono final.

La envolvente tiene ese recorrido y el comando SOUND utiliza el tiempo restante sonando con el último tono.

Otro problema que se nos puede presentar es que el parámetro V de la envolvente de tono haga que éste intente tomar un valor que está fuera de rango.

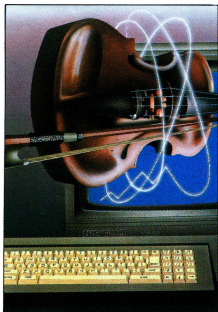
Como sabemos, el valor del parámetro de tono sólo puede tener un rango de valores comprendidos entre 0 y 4095. Pero, ¿qué ocurre cuando al incrementar o decrementar el tono en uno de los escalones de la envolvente intenta salirse fuera de rango?

Al encontrarnos con un problema similar en la envolvente de volumen vimos que el **Amstrad** recorre «el camino del reloj» con los valores dentro de rango. Este es también el caso de la envolvente de tono. Probemos:

```
ENT 5,5,—100,100  
SOUND 1,300,500,5,0,5
```

Y

```
ENT 6,5,100,100  
SOUND 1,300,500,5,0,6
```



y escucharemos lo que queremos decir. Hay una parte silenciosa que se da cuando el parámetro de tono es igual a cero.

Resumiendo...

Y esto es todo por esta semana, excepto avisar, como siempre, que la envolvente de tono no es tan sencilla como parece. Lo mismo que ocurre con la envolvente de volumen, puede tener cinco secciones en lugar de la única que hemos utilizado hasta ahora.

Esto significa que en lugar de:

```
ENT S,T,V,W
```

la definición actual de una envolvente de tono es:

```
ENT S, T1,V1,W1,T2,V2,W2,  
T3,V3,W3,T4,V4,W4,  
T5,V5,W5
```

De nuevo hemos conseguido llegar a un enorme engendro de 16 parámetros. Déjenos decirle otra vez que no es tan fiero como parece.

Aunque nos hemos encontrado con cinco secciones diferentes, cada una se comporta del mismo modo que la que vimos al principio. La diferencia es que en lugar de T, V y W la primera sección tiene los parámetros T1, V1, y W1, la segunda T2, V2 y W2 y así sucesivamente. La Figura III nos representa cómo se relacionan los parámetros con cada una de las secciones.

Aunque podemos tener cinco secciones en una envolvente de tono —*como es evidente según lo anterior*— no tenemos por qué utilizarlas todas a la vez. Para ilustrarlo vamos a fabricar una envolvente con tres secciones tal como la definida con:

```
ENT 1,5,10,20,5,—5,20,5,5,20
```

Esta envolvente de tono está numerada como la 1 y tiene tres secciones con una duración total de tres segundos. Tomando cada sección por orden podemos ser capaces de ver lo que ocurre. Cuando nos parezca que ya lo sabemos invocaremos a la envolvente con:

```
SOUND 1,200,300,5,0,1
```

y comprobaremos si nuestras suposiciones son ciertas.

No debemos preocuparnos por la cantidad de parámetros que tiene la envolvente de tono. Mientras no la demostramos que tenemos miedo todo irá bien.

Y para adquirir práctica le dejamos con el Programa III que le ayudará a crear sus propios envolventes de tono y escuchar si le gustan.

Programa tres

Y, ahora sí, esto es todo por esta semana, aunque no es el final de nuestro tratamiento de envolventes. Después de todo, hay algunas preguntas que necesitan una respuesta. Como por ejemplo, ¿por qué existen las envolventes?

PROGRAMA

```
10 REM PROGRAMA I  
20 SOUND 1,210,20,5  
30 SOUND 1,220,20,5  
40 SOUND 1,230,20,5  
50 SOUND 1,240,20,5  
60 SOUND 1,250,20,5
```

```
10 REM PROGRAMA II  
20 SOUND 1,190,20,5  
30 SOUND 1,180,20,5  
40 SOUND 1,170,20,5  
50 SOUND 1,160,20,5  
60 SOUND 1,150,20,5
```

```
10 REM PROGRAMA III  
20 REM ENVELOVENTE DE TONO  
30 DIM t(5),v(5),w(5)  
40 WHILE -1  
50 MODE 1  
60 INPUT "Número de secciones en la  
envolvente de tono? ",secciones  
70 IF secciones<1 OR secciones>5 TH  
EN CLS:GOTO 60  
80 CLS  
90 FOR bucle=1 TO secciones  
100 LOCATE 3,5:PRINT"Sección"bucle  
110 LOCATE 3,8:PRINT"Número de esca  
lonaje"  
120 LOCATE 30,8:INPUT "t,(bucle)  
130 IF t(bucle)<0 OR t(bucle)>255 T  
HEN LOCATE 30,8:PRINT SPACES(8):GOT  
O 120  
140 LOCATE 3,13:PRINT"Altura de cad  
a escalón"  
150 LOCATE 30,13:INPUT "v,(bucle)  
160 IF v(bucle)<=128 OR v(bucle)>12  
7 THEN LOCATE 30,13:PRINT SPACES(8)  
:GOTO 150  
170 LOCATE 3,18:PRINT"Tiempo de pau  
sa"  
180 LOCATE 30,18:INPUT "w,(bucle)  
190 IF w(bucle)<0 OR w(bucle)>255 T  
HEN LOCATE 30,18:PRINT SPACES(8):G  
O TO 180  
200 LOCATE 14,25:PRINT"PULSA ESPACI  
O"  
210 WHILE INKEY(47)=1:WEND:CLS  
220 WHILE INKEY<>"":WEND  
230 NEXT bucle  
240 ENT 1,t(1),v(1),w(1),t(2),v(2),  
w(2),t(3),v(3),w(3),t(4),v(4),w(4),  
t(5),v(5),w(5)  
250 duracion=(1)*w(1)+(2)*w(2)+(3)  
*w(3)+(4)*w(4)+(5)*w(5)  
260 SOUND 1,200,duracion,5,0,1  
270 CLS  
280 duracion=RIGHT$(STR$(duracion)  
,LEN$(STR$(duracion))-1)  
290 PRINT"SOUND 1,200,"duracion":  
"5,0,1"  
300 FOR bucle=1 TO secciones  
310 bucle=RIGHT$(STR$(bucle),1)  
320 PRINT "t("bucle";) "t(bucle)  
330 PRINT"v("bucle";) "v(bucle)  
340 PRINT"w("bucle";) "w(bucle)  
350 NEXT  
360 LOCATE 14,25:PRINT"PULSA ESPACI  
O"  
370 WHILE INKEY(47)=1:WEND:CLS  
380 WEND
```

INSTRUCCIONES DE CAMBIO TRANSFERENCIA Y BUSQUEDA (II)

Continuaremos hablando hoy de este bloque de instrucciones del cual hemos visto ya las de cambio: las instrucciones de transferencia de bytes.



La primera de estas instrucciones se representa de la siguiente forma:
LDI

Tras su ejecución, un byte se transfiere desde la posición de memoria indicada por el contenido del registro par HL a la posición de memoria indicada por el contenido del registro doble DE. Una vez hecho esto, ambos registros pares se incrementan y el registro doble BC se decrementa.

Veremos a continuación un ejemplo de actuación para ver más claro el efecto que produce.

Supongamos que el valor y contenido de los registros dobles a los que afectan dicha instrucción son los siguientes:

	Posición de memoria	Contenido de dicha posición de memoria
HL	#7000	255
DE	#8000	50
BC	#1000	No afectado

Después de la ejecución de LDI obtendremos:

HL	#7001	—	—
DE	#8001	—	—
BC	#9FF	No afectado	

y el contenido de la posición de memoria #7000 y #8000 será:

#7000	50
#8000	255

Un ejemplo práctico de cómo trabaja la instrucción LDI, lo podemos ver en el programa

número 1. En primer lugar cargamos en el registro doble HL con la posición de memoria cuyo contenido queremos traspasar y luego cargamos en DE con la posición de memoria hacia la cual queremos traspasar el contenido de la anterior. Dado que en este programa nos proponemos traspasar el contenido de la posición #BB00 a la posición de pantalla #C000 lo que hacemos es cargar en HL esa primera dirección y en DE la siguiente.

Una vez hecho esto, colocamos la instrucción LDI, que es la que producirá el efecto deseado. Así pues una vez ejecutado dicho programa podremos observar cómo se ilumina un byte en pantalla.

Transferir un número fijo de bytes

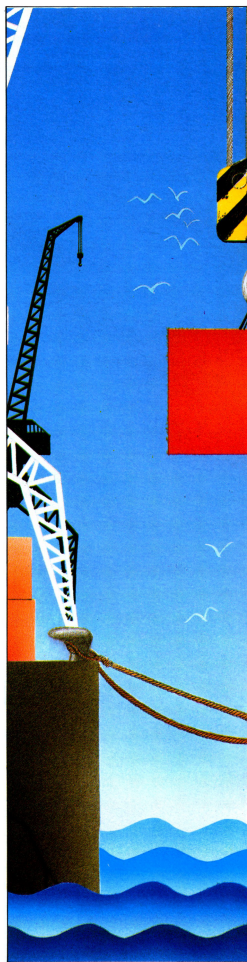
El programa número dos aprovecha además la facultad que tiene esta instrucción de decrementar el registro doble BC.

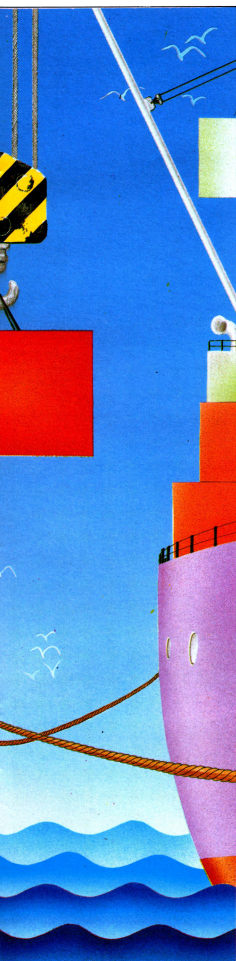
En este programa no vamos a traspasar un solo dato, sino que traspasaremos tantos bytes como indique el registro BC. Por lo tanto vamos a cargar HL y DE con los mismos valores que en el programa anterior, y además cargaremos BC con 1000, éstos serán los bytes totales a traspasar. El siguiente paso es ejecutar LDI y una vez hecho esto comprobaremos cuál es el valor actual del registro BC, si no es cero volveremos al principio y se ejecutará otra vez el programa. Si es cero retornaremos al Basic.

Precisamente el efecto que produce la instrucción que veremos a continuación es la realizada por el anterior programa. Esta instrucción se presenta de la siguiente forma:

LDIR

Su ejecución transfiere un byte de datos desde la posición de memoria indicada por el registro par DE. Ambos registros dobles se incrementan y el registro par BC se decrementa. Si tras este decremento el registro doble BC es cero, termina la ejecución de la instrucción. Si BC no es cero, BC se decrementa en dos y la instrucción se repite hasta que el contenido





de dicho registro doble sea cero. Debemos decir que detrás de cada traspaso de datos se comprueban las interrupciones.

El registro BC actúa de contador

Si antes de ejecutar la anterior instrucción BC con el valor cero, la instrucción se repetirá 65.535 veces o sea 64 Kbytes. Esto se debe a que LDIR decrementa el registro doble BC, y si no es cero se repite la instrucción. Así pues si BC vale cero, al decrementar BC éste se pone a #FFFF y dado que es cero, la instrucción se repite hasta que alcance ese valor.

El efecto que produce el programa número tres, es idéntico al producido por el programa anterior, si bien, como podemos comprobar, el número de instrucciones de este último es sensiblemente menor debido a que LDIR comprueba si el registro BC es cero y, si es así, deja de actuar mientras que la anterior instrucción no tenía esta capacidad.

Así pues este programa nos traspasará 1000 bytes a partir de la dirección #8B00 hacia la zona de pantalla.

El programa número cuatro realiza un traspaso de bytes desde la dirección #8B00 hacia la dirección #8B00. El número de bytes a traspasar es de #500, valor éste que hemos cargado en BC, que es el registro contador.

Bien, como podéis comprobar, lo que pretendemos con este programa, es precisamente traspasar el firmware del AMSTRAD a una nueva posición que será la #8B00.

Después de la instrucción LDIR, cargamos el acumulador con el carácter «A» y llamaremos a la rutina de firmware que imprime en pantalla, pero dado que tenemos dos bloques de firmware, en lugar de llamar al original llamaremos al que hemos creado que se encuentra a partir de la dirección #8B00. Por lo tanto la rutina de impresión estará situada en la dirección #8B5A, y obtendremos el carácter «A» en pantalla.

Instrucción LDD

La instrucción que veremos a continuación se representa de la siguiente forma:

LDD

Esta instrucción transfiere un byte desde la posición de memoria indicada por el contenido del registro par HL a la posición de memoria indicada por el registro doble DE. Una vez hecho esto, ambos registros se decrementan. También se decrementa el registro doble BC.

Como podemos observar esta instrucción produce el mismo efecto que LDI, pero en lugar de incrementar los registros HL y DE, en este caso se decrementan.

Vamos a ver en el siguiente ejemplo la forma en que actúa la introducción LDD.

Supongamos que los registros dobles HL, DE y BC contienen los siguiente valores:

Código máquina

Posición de memoria	Contenido de dicha posición
HL	#8000 250
DE	#7000 35
BC	#1000 No afectado

Después de la ejecución de la anterior instrucción obtendremos los siguiente valores:

HL	#7FFF	—
DE	#6FFF	—
BC	#FFF	No afectado

y en la posición de memoria #7000 obtendremos el valor 250 que anteriormente contenía la posición #8000.

El programa número cuatro presenta un ejemplo práctico de actuación de la instrucción anterior. Esta rutina se encarga de traspasar un byte desde la posición de memoria indicada por el registro doble HL hacia la posición indicada por el registro par DE.

El objetivo del siguiente programa es traspasar un bloque de bytes desde la posición indicada por HL hasta la posición indicada por DE.

El número de bytes a traspasar está indicado en el registro par BC. Dado que este registro se decrementa cada vez que actúa la instrucción LDD, comprobaremos si BC es cero. Si es así el programa termina y retorna al Basic, de lo contrario vuelve a ejecutarse dicha instrucción.

Este programa traspasa un bloque de #4000 bytes desde la posición #8000 hasta la posición #C000, correspondiente esta última a la zona de memoria de pantalla. Por lo tanto los registros pares BC, HL y DE deberán contener respectivamente los valores, #4000, #FFFF y #BFFF.

Instrucción LDDR

La última de las instrucciones de transferencia es la que se indica a continuación:

LDDR

Esta instrucción transfiere un byte desde la posición de memoria indicada por el contenido del registro par HL a la posición de memoria indicada por el registro doble DE. Una vez

hecho esto ambos registros se decrementan, el registro BC también es decrementado.

Si una vez decrementado el registro par BC éste ha llegado a cero, la instrucción termina en su momento. Si el registro BC no es cero, entonces se decrementa en dos y la instrucción se repite hasta que BC sea cero. Si BC vale cero antes de la entrada a la instrucción, entonces la instrucción se repetirá durante 65.535 veces. Después de cada transferencia de datos se reconocerán las interrupciones.

El programa número seis nos muestra cómo resolver la rutina que veíamos anteriormente, pero esta vez con un ahorro considerable de instrucciones. Esto es debido a que LDDR, ejecuta varias cosas a la vez, esto es, decrementa los registros dobles BC, HL y DE, además comprueba si BC vale cero, si es así la instrucción finaliza, de lo contrario la instrucción continúa hasta que dicho registro alcance el valor cero.

Por lo tanto este programa nos pasará un bloque de 16 K de datos desde la posición #8000 hasta la posición #C000 de pantalla.

El último de los programas que hemos preparado presenta una rutina que se encuentra en la posición de memoria #A000, y su misión es imprimir un mensaje en pantalla.

Ahora lo que haremos será traspasar ese programa desde la posición en que se encuentra hacia una nueva posición, por el ejemplo una posición de pantalla como la #C000, y una vez allí lo ejecutaremos.

Para poder traspararlo a esa nueva posición, debemos cargar en el registro doble HL, la posición donde se encuentra el programa, en el registro par DE deberemos poner la dirección de memoria donde va a ser trasparado, y en el registro doble BC pondremos el número de bytes que contiene el programa. Una vez hecha esta ejecución LDDR, y tendremos el programa en esa nueva posición.

Antes de trasparar un programa de una zona de memoria a otra, deberemos de tener en cuenta de que éste no tenga llamadas absolutas en forma de instrucciones CALL o JP, ya que si existieran, deberíamos corregir los saltos que se producen, porque de lo contrario el programa se perdería en la memoria y el ordenador se colgaría.



GRAFICO

```

10 :PROGRAMA-1
20 I
30 ORG #A000
40 LD HL,#B000
50 LD DE,#C000
60 LDI
70 RET
    
```

```

10 :PROGRAMA-2
20 I
30 ORG #A000
40 LD HL,#B000
50 LD DE,#C000
60 LD BC,#1000
70 BUC: LDI
80 LD A,B
90 OR C
100 JR NZ,BUC
110 RET
    
```

ETIQUETAS

BUC AB07

```

10 :PROGRAMA-3
20 I
30 ORG #A000
40 LD HL,#B000
50 LD DE,#C000
60 LD BC,#1000
70 LDIR
80 RET
    
```

```

10 :PROGRAMA-4
20 I
30 ORG #A000
40 LD HL,#B000
50 LD DE,#B000
60 LD SC,#500
70 LDIR
80 LD A,"*
90 CALL #B5A0
100 RET
    
```

```

10 :PROGRAMA-5
20 I
30 ORG #A000
40 LD HL,#C000
50 LD DE,B
60 LD BC,#4000
70 BUC: LDI
80 LD A,B
90 OR C
100 JR NZ,BUC
110 RET
    
```

ETIQUETAS

BUC AB09

```

10 :PROGRAMA-6
20 I
30 ORG #A000
40 LD HL,#C000
50 LD DE,B
60 LD SC,#4000
70 LDIR
80 RET
    
```

```

10 :PROGRAMA-7
20 I
30 ORG #A000
40 LD HL,#1000
50 LD DE,#C000
60 LD BC,15
70 LDDR
80 JR #C841
90 LD HL,#7000
100 LD (HL),48
110 LD A,(HL)
120 BUC: INC (HL)
130 CALL #B5A0
140 INC (HL)
150 DJNZ BUC
160 INC I
170 RET
    
```

ETIQUETAS

BUC AB15 INICI AB1C

PROGRAMAS

```

10 REM * PROGRAMA 1 *
20 REM
30 FOR N=#A000 TO &AB0
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&483 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,0,187,17,0,192,237
90 DATA 168,201,0,0,0,0,0
    
```

```

10 REM * PROGRAMA 2 *
20 REM
30 FOR N=#A000 TO &AB10
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&657 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,0,187,17,0,192,1
90 DATA 0,16,237,168,128,177,32
100 DATA 250,201,0,0,0,0,0
    
```

```

10 REM * PROGRAMA 3 *
20 REM
30 FOR N=#A000 TO &AB0C
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&424 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,0,187,17,0,192,1
90 DATA 0,16,237,176,201,0,0
    
```

```

10 REM * PROGRAMA 4 *
20 REM
30 FOR N=#A000 TO &AB11
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&615 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,0,187,17,0,139,1
90 DATA 0,5,237,176,62,65,205
100 DATA 90,139,201,0,0,0,0
    
```

```

10 REM * PROGRAMA 5 *
20 REM
30 FOR N=#A000 TO &AB10
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&504 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,0,192,17,0,0,1
90 DATA 0,64,237,168,128,177,32
100 DATA 250,201,0,0,0,0,0
    
```

```

10 REM * PROGRAMA 6 *
20 REM
30 FOR N=#A000 TO &AB0C
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&3A1 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,0,192,17,0,0,1
90 DATA 0,64,237,184,201,0,0
    
```

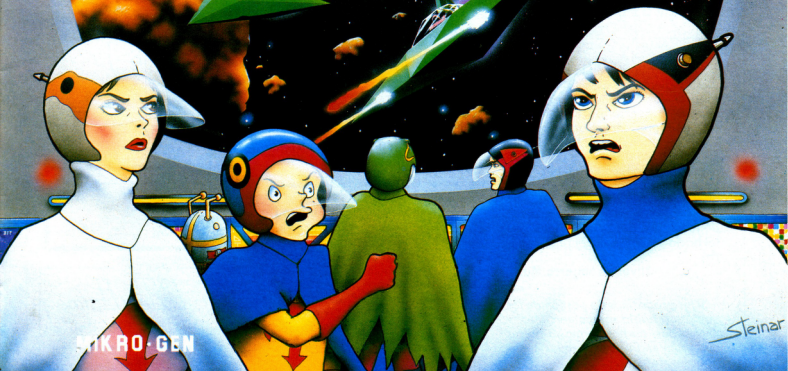
```

10 REM * PROGRAMA 7 *
20 REM
30 FOR N=#A000 TO &AB10
40 READ A:SUMA=SUMA+A
50 POKE N,A
60 NEXT
70 IF SUMA<>&A03 THEN PRINT "ERROR
EN DATAS"
80 DATA 33,20,168,17,79,192,1
90 DATA 15,0,237,184,195,65,192
100 DATA 33,0,112,54,48,6,18
110 DATA 126,285,98,187,52,14,249
120 DATA 201,0,0,0,0,0,0
    
```

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

BATTLE OF THE PLANETS

UNETE AL "COMANDO G" EN SU ULTIMA AVENTURA CONTRA ZOLTAR EN UN MARAVILLOSO JUEGO REALIZADO CON UNOS GRAFICOS TRIDIMENSIONALES QUE HAN DE SER VISTOS PARA SER CREIDOS.



MIKRO-GEN

BATTLE OF PLANETS ES EL PROGRAMA SELECCIONADO PARA EL
CAMPEONATO INTERNACIONAL DE JUEGOS DE ORDENADOR.
CONSIGUE LA MAXIMA PUNTUACION ESPAÑOLA Y PARTICIPA EN LA FINAL DE LONDRES.
(LAS BASES DEL CONCURSO ESTAN EXPLICADAS EN EL PROGRAMA)

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE C/. STA. ENGRACIA, 17. 28010 MADRID, TFNO.: (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TFNO.: (93) 432 07 31

Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

Me gustaría que en la sección correspondiente de su revista, me explicaran cómo puedo compaginar el programa que aparece en el libro que entregan con el ordenador al tratar el comando WEND (concretamente se trata de poner en funcionamiento un reloj), con cualquier programa de gestión a los cuales soy muy aficionado.

El problema radica en que cuando me desvío de la subrutina del reloj hacia otras tareas, el reloj se para.

Silvestre Segarra Masía

Si los programas de gestión de los que usted habla son comerciales, y protegidos, no hay forma, a no ser que estén en Basic y usted los consiga desproteger.

En cuanto a los programas hechos por usted, puede emplear el comando EVERY del Basic, que, mediante la técnica conocida como interrupciones, ejecuta una subrutina cada cierto periodo de tiempo.

La sintaxis del comando EVERY es la siguiente:

EVERY tiempo, temporizador, GOSUB número-de-línea en donde «tiempo», es una cantidad en cincuentavos de segundo, «temporizador» es justamente eso, uno de los 4 que posee el Amstrad y que sirven para indicar la prioridad (descendiendo de 3 a 0) y GOSUB número-de-línea es una llamada normal a la subrutina que usted quiere que se ejecute cada «tiempo» veces.

Amigos de MICROHOBBY: Estudié programación con IBM P/C en el cual, al crear un fichero tenía la sentencia APPEND para poder añadir datos al mismo fichero creado. Me compro el 664 y, ¿cuál es mi sorpresa?... me encuentro que cuando hago un programa para crear ficheros, cuando quiero añadirle más datos al fichero... ¡jime machaca los datos anteriores!

Es posible que con AMSTRAD no pueda acumular datos en los ficheros secuenciales y tenga que recurrir a poner los datos en READ-DATA o, ¿crear de nuevo todas las veces el fichero?

Manuel Goret Rochera

Efectivamente, el Amstrad sólo maneja ficheros secuenciales, pero la utilidad APPEND no se ha incluido.

La solución es, nos tememos, crear un nuevo fichero, trasvasar a él los datos del anterior y añadir los que hagan falta.

Otra posibilidad es olvidarse de los ficheros secuenciales y recurrir a los aleatorios. Si no recuerdo mal, en el disco que se entrega con el 664 viene un paquete de utilidad que crea nuevos comandos lrx que lo hacen.

Soy un poseedor del Amstrad CPC-464. Estoy en vistas de comprarme la unidad de discos que se puede acoplar a este ordenador.

1.º) Con esta unidad de discos, ¿a qué tipo de software podré llegar, o sea, ¿a qué tipo, CP/M, etc...?

2.º) ¿Podré cambiar programas grabados en discos de un ordenador CPC-664? ¿Se ejecutarán en mi ordenador metidos en esa unidad de discos?

3.º) ¿Existe algún tipo de programa, o copión, con el cual pueda pasar unos programas PROTEGIDOS, en cassette, a los discos?

Fernando Martínez

1) Software CP/M 2.2 y programa Basic o código máquina.

2) Si, siempre que esos programas no usen los nuevos comandos Basic del 664.

3) En Inglaterra se vende varios «copiones», pero no se puede garantizar que puedan con todos los tipos de protecciones que existen ahora o puedan existir.

—¿Hay alguna forma de conectar el monitor de f. verde a la salida de video del Spectrum, con algún circuito especial?

Antonio Sanz Arnaiz (Burgos)

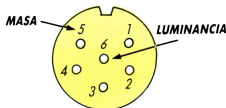
En principio, debiera poder hacerse perfectamente. Basta averiguar, en el zócalo del monitor del Amstrad, qué patilla corresponde a masa y cuál a la señal de video propiamente dicha, a la «luminancia».

Con esta información y conociendo lo mismo del Spectrum, bastaría conectarlo adecuadamente y listo.

En el manual de Amstrad viene un esquema del zócalo del monitor, en la página 7/39.

Para mayor comodidad, lo reproducimos aquí:

ZOCALO DEL MONITOR VISTO POR DETRAS



Patilla 1 ROJO
Patilla 2 VERDE
Patilla 3 AZUL

Patilla 4 SINC
Patilla 5 MASA
Patilla 6 LUM

PUBLICIDAD

New Line

GABINETE DE INFORMATICA

- **Clases de Informática sobre AMSTRAD**
Exclusivamente individuales.
- **Ordenadores AMSTRAD y periféricos**
Los mejores precios
- **Software a la medida**

ZURBANO, 4 ☎ 410 47 63
28010 MADRID

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

SOFTWARE: por cada programa GRATIS ¡¡1 BOLIGRAFO CON RELOJ DE CUARZO!!

HYPER SPORTS	2.300 ptas.
TORNADO LOW LEVEL	1.950 ptas.
EXPLODING FISTT	2.300 ptas.
JUMP JET	2.495 ptas.
ZORRO	2.600 ptas.
SABREWULF	1.650 ptas.
GHOSTBUSTERS	1.950 ptas.
GYROSCOPE	2.300 ptas.
HYGHWAY ENCOUNTER	1.750 ptas.
HIGHWAY ENCOUNTER DISCO	3.300 ptas.

DYNAMITE DAN	2.100 ptas.
RAID OVER MOSCOW	2.300 ptas.
THEY SOLD A MILLION	2.500 ptas.
FIGHTER PILOT	1.975 ptas.
MASTER OF T. LAMP	1.950 ptas.
NIGHTSHADE	1.950 ptas.
HACKER	1.950 ptas.
SUPER TEST	2.300 ptas.
MAPGAME	2.700 ptas.
TONADO LOW LEVEL DISCO	3.300 ptas.

JOYSTICK QUICK SHOTT II... 2.295 ptas.

JOYSTICK QUICK SHOT V ... 2.595 ptas.

PC-COMPATIBLE IBM 256 K
MONITOR FOSFORO VERDE
2 BOCAS DISKETTE 360 K
SOLO ¡¡243.900!!

TAPA METACRILATO PARA
TECLADO ¡¡1.900 ptas.!!

UNIDAD DISKETTE 5.25"
¡¡45.900 ptas.!!
(incluido controlador)

LAPIZ OPTICO
¡¡4.900 ptas.!!

IMPRESORA MARGARITA
¡¡49.900 ptas.!!

CASSETTE ESPECIAL
ORDENADOR 5.295 ptas.

PRECIOS SUPER-EXCEPCIONALES PARA
AMSTRAD CPC-472 Y CPC-6128
¡¡LLAMANOS, TE ASOMBRARAS!!

IMPRESORAS ¡¡20% DTO. SOBRE P.V.P.!!

SINTETIZADOR DE VOZ
Y AMPLIFICADOR:
7.900 ptas.

MODULADOR TV
8.400 ptas.

CINTA C-15 ESPECIAL
ORDENADOR 85 ptas.
DISKETTE 3" 990 ptas.

INTERFACE DISCO
5 1.4" 5.300 ptas.

UNIDAD DE DISCO 3" CON
CONTROLADOR: 49.900 ptas.

Libros:	
Curso autodidáctico Basic I	2.525 ptas.
Curso autodidáctico Basic II	2.525 ptas.
Programando con Amstrad	2.195 ptas.
Juegos sensacionales Amstrad	1.950 ptas.
Hacia la Inteligencia Artific.	1.295 ptas.
Música y sonidos con Amstrad	995 ptas.

Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A. AMSTRAD SEMANAL.**
Apartado de correos 54.062
28080 MADRID
¡ABSTENERSE PIRATAS!

Vendo Amstrad 464 verde 50.000. Disco con controlador 47.000. Todo por 95.000 pts. Garantía 4 meses. Con manuales, libros y 20 juegos y utilidades, los mejores. Tel. 888 58 41. José Luis. Cenas. Madrid.

Intercambio con usuarios **Amstrad** tanto programas como especialmente trucos y utilidades. Poseo un CPC 6128 y estoy también interesado programas en Disco. Contestaré a todos. Y a ti que empiezas. Luciano Sancha Rivero. C/ Avda. José Farinas, 54 - 5.º D. 21006 Huelva. Tel. 22 47 20 (9-10 noche).

Vendo o cambio juegos (últimos éxitos) para el ordenador **Amstrad** CPC 464 entre ellos tengo copiones, knight lore, decathlon, beach-head, etc... Borja Sancho. C/ Pza. San Francisco, 7. Borja (Zaragoza) Tel. (976) 86 77 28.

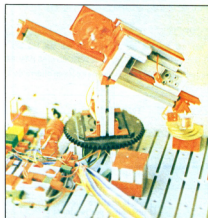
Cambio programas comerciales, últimos éxitos. Dragontorc, knighe lore, pijamarama, etc. Escribir a: C/ Mudela, 29. 4.º C. 28018 Madrid.

Desearía contactar con usuarios del Amstrad CPC 6128 para intercambio, de ideas, comentarios, etc. Tengo algunos programas interesantes. A ser posible de Madrid. Carlos. Tel. (91) 233 05 74 (comidas o cenas).

Vendo, cambio programas para **Amstrad** CPC 664. Interesados escribir a Juan Mucientes Rasilla. C/ Nicaragua, 14-5.º C. 15005 La Coruña.

Vendo lector de discos **Amstrad** nuevo, con garantía, por 40.000 pts. e impresora Seikoshia mod. SP 800 nueva, con garantía, por 54.000 pts. Interesados llamar al Tel. (972) 21 17 86.

MASTER COMPUTER



Si no lo encuentras en tu tienda habitual. llámanos y te lo enviaremos directamente contra reembolso.

Tenemos todos los modelos de AMSTRAD, periféricos, software y libros. Programas y juegos para el 8256.

AMSTRAD (programas e instrucciones en castellano)
Commodore
Apple-Accorn
Spectrum
Robot

Fischertechnik

precio incluyendo caja de construcción software, interface y adaptador 34.990 ptas.

Distribuidor para España
Centro Comercial, Local 15. Ciudad Sto. Domingo
Carretera de Burgos, Km 28 ALGETE - MADRID
Telf. 622 12 89

Ya tenemos el
ATARI 520 S.T.

ORDEMAMANIA SOFT

TE OFRECEMOS EL NUEVO
PLAN GENERAL CONTABLE
CON I.V.A.

- Contabilidad CPC 664 y CPC 6128
- Contabilidad PCW 8256

13.900 ptas.
37.500 ptas.

Disponemos de un equipo de software a tu servicio.
Hacemos programas a medida.

RECUERDA: —Damos solución a la pequeña y mediana empresa.

Torres Quevedo, 34

Tel. 967/22 79 44

02003 ALBACETE

...descubre el N.º 3

ya está en
tu quiosco

AMSTRAD

también disponible
para

SPECTRUM, PLUS, 128

COMMODORE 64

CARNIVAL

Si te gusta el tiro al blanco, con este programa podrás practicar sin necesidad de salir de casa. Si tienes buena puntería obtendrás disparos gratis.

BLOCKER

Demuestra tu habilidad esquivando las paredes y a tus enemigos. Cuantos más destruyas, más aparecerán ante ti.

SPACE

Al cargar este programa aparecerá ante ti un batallón de alienígenas. Tu misión es destruirlos, pero cuidado, su intención es eliminarte lo antes posible.

HAUNTED

En este caso debes coger todos los puntos que aparecen en el laberinto. ¡Atención!, los fantasmas están enfadados e intentarán deshacerse de ti a toda costa.

VAMPIRO

Es un programa en el cual pueden participar dos jugadores. La misión de cada uno será pintar las lápidas de un color distinto. La destrucción del enemigo significa la victoria.

SPLIT

Es una rutina en código máquina, que te permitirá siete colores en pantalla en MODO 1, en el cual normalmente sólo se pueden utilizar cuatro.

795 pts.
(Incluido IVA)

Si no lo encuentras en su quiosco, solicítelo directamente a nuestra editorial.

SINTAX, S.A.

Paseo de la Castellana, 268.
28046 Madrid. Tel. (91) 733 25 99

YOUR COMPUTER

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de ordenadores de Europa. Ahora reproducidos en cassette, en auténtica exclusiva mundial.

SINCLAIR STORE

EL CENTRO DEL HARDWARE

SPECTRUM 48 K
SPECTRUM PLUS
SPECTRUM 128
SINCLAIR QL
COMMODORE 64
COMMODORE 128
COMMODORE PC 20
AMSTRAD 472
AMSTRAD 6128
AMSTRAD 8256
Y
SPECTRAVIDEO
MSX



- EN SINCLAIR STORE USTED NO PAGA EL IVA
- IMPORTANTES DESCUENTOS Y/O REGALOS
- POR LA COMPRA DE UN ORDENADOR, CURSO GRATIS DE INFORMATICA
- SOFTWARE DESCUENTOS HASTA EL 20%
- MONITORES 20% DESCUENTO.
- EN TODAS LAS IMPRESORAS 20% DE DESCUENTO
- JOYSTICK QUICK SHOT II INTERFACE TIPO KEMPSTON 3.800 Pts.
- JOYSTICK ANATOMICO AMARILLO INTERFACE TIPO KEMPSTON 3.200 Pts.

- PC COMPATIBLE IBM P.V.P. 212.000 Pts.
- ¡ULTIMA NOVEDAD EN EL MERCADO! ATARI 520 ST YA DISPONIBLE. ¡VEN A PROBARLO!
- PRECIOS ESPECIALES PARA COLECTIVOS Y EMPRESAS
- DISTRIBUIDORES OFICIALES DE TODAS LAS MARCAS. CON AUTENTICO SERVICIO PROFESIONAL DE POST-VENTA
- VEN A VERNOS, NOSOTROS MANTENEMOS LAS REBAJAS, EN TODOS LOS ARTICULOS, HASTA EL 31 DE MARZO.
- NECESITAMOS DISTRIBUIDORES. SOMOS MAYORISTAS

sinclair store

SOMOS PROFESIONALES

BRAVO MURILLO, 2
(Glorieta de Quevedo)
Tel. 446 62 31 - 28015 MADRID
Aparcamiento **GRATUITO** Magallanes, 1

DIEGO DE LEON, 25
(Esq. Nuñez de Balboa)
Tel. 261 88 01 - 28006 MADRID
Aparcamiento **GRATUITO** Nuñez de Balboa, 114

FELIPE II, 12
(Metro Goya)
Tel. 431 32 33 - 28 009 MADRID
Aparcamiento **GRATUITO** Felipe II