

MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO I N.º 7

150 Ptas.

Canarios 160 ptas.

**GRAFICOS EN
MOVIMIENTO**

AMPLIA INFORMACION

**SONIMAG 85:
BARCELONA,
23/SEPTIEMBRE**

**COMO UTILIZAR
LOS COMANDOS
WHILE... WEND**

**RESOLUCION
GRAFICA DE
ECUACIONES**

SOFTWARE

**JUGGERNAUT:
CONDUCE UN
TRAILER DE 18
TONELADAS**



HOP EDITA HOBBY PRESS S.A.

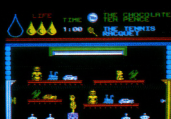
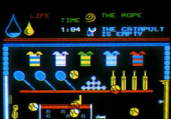
¡ YA ESTA AQUI !



SENSATIONAL SOFTWARE FROM

MIKRO-GEN

Herbert's Dummy Run



HERBERT'S DUMMY RUN
SPECTRUM/COMMODORE/AMSTRAD

LO QUE DICE LA CRITICA:
HERBERT'S ES UNA MEZCLA ENTRE EL "PYJAMARAMA" Y EL "EVERYONE'S A WALLY", PROGRAMAS DE LOS QUE HA ESCOGIDO LO MEJOR PERO AUMENTANDO LA CALIDAD DE GRAFICOS Y MOVIMIENTO.

RECOMENDADO PARA TODO TIPO DE USUARIOS, PARA LOS QUE PREFIERAN LOS JUEGOS SENCILLOS DE ARCADE Y PARA LOS QUE PREFIERAN LOS JUEGOS COMPLICADOS EN LOS QUE ADEMAS DE SER HABILIDOSO ES NECESARIO ACTUAR CON LOGICA Y CREAR UN PLAN DE ESTRATEGIA A SEGUIR. Y SOBRE TODO RECOMENDADO PARA LOS QUE DISFRUTARON CON PYJAMARAMA Y WALLY.

VALORACION: * * * * *

AMSTRAD

sumario

Director Editorial

José I. Gómez-Centurió

Director Ejecutivo

Victor Prieto

Subdirector

José María Díaz

Redactora Jefe

María García

Diseño

José Flores

Colaboradores

Francisco Partalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Juan J. Martínez

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Javier Martínez

Portada

Javier Iguat

Ilustradores

J. Iguat, J. Pons, F. L. Frantón,

J. Septon, Pejo, J. J. Mora,

Luigi Pérez

Edita

HOBBY PRESS S.A.

Presidente

María Andrión

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 Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

La Granja, s/n

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Feropadre

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDEC, S. A. Crta. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Depósito 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.: 21 21 64. 1.209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidario de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

Año 1 • Número 7 • 15 al 21 de Octubre de 1985
150 ptas. (sobretasa Canarias, 10 ptas.)

5 Primera plana

Últimas noticias del mundo del ordenador: los problemas de Apple Computer, nueva directiva en Sinclair Research y la presentación en España del PCW8256.

ProgramAcción 6

Para dar vida y animación a los gráficos de un programa, lo mejor que se puede hacer, dicen algunos, es recurrir al lenguaje máquina. No obstante, como pretendemos estudiar en unos cuantos artículos, el Basic Amstrad tiene mucho que decir en este sentido.

10 Sonimag, 85

Informamos de la Feria del Sonido y la Imagen que se celebró en Barcelona del 23 al 29 de septiembre. Muchas novedades han aparecido en nuestro país, sobre todo la presentación oficial del Amstrad CPC6128.

13 Análisis

Muchos ordenadores poseen en su juego de instrucciones un comando que permite representar elipses, definiendo a nuestro gusto el centro y ejes de la misma.

Análisis, en este número, estudia un pequeño programa que, consiguiendo el mismo efecto, te ayudará a comprender el proceso seguido en la representación de estas curvas.

14 Serie oro

Siempre ha constituido un verdadero problema para cualquier estudiante de BUP, tanto la resolución y representación gráfica de ecuaciones de segundo grado como el de sistemas de dos incógnitas.

Serie oro, con el Incognitor, te lo pone fácil: no más trabajos cálculos a mano.

También tienen cabida aquí los clásicos de la programación, los programas pioneros en el mundo de los juegos. Atari empezó, y Smaley cierra dignamente el capítulo de los «comecocos».

18 Mr. Joystick

Conducir un trailer de 24 ejes, acompañado de las tareas de carga, descarga, aparcamiento y tráfico por una ciudad de intrincada estructura, es una misión en extremo difícil tanto para el neófito como para un conductor de primera. Juggernaut te acerca al mundo de los camiones pesados.

Boulder Dash: un simpático personaje a la caza de los diamantes subterráneos.

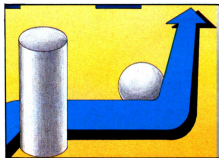


Primeros pasos 22

Ni un solo programa funciona sin incorporar en su interior tareas repetitivas, bucles. Una de las mejoras formas de hacerlo es usar los bucles estructurados, como el WHILE.. WEND.

26 Código Máquina

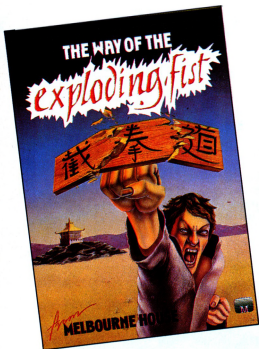
Seguimos estudiando los registros dobles del Z80, en una de sus funciones más esenciales: actuar de punteros para acceder a zonas predeterminadas de la memoria.



Siempre los primeros en tener lo mejor...

circulo de soft

MICROAMIGO S.A.



AJEDREZ (3 Dimensiones) 3.500 Ptas.
THE COVENANT 2.700 Ptas.
EXPLODING FIST 3.100 Ptas.

¡Un increíble regalo por cada programa!

Este magnífico reloj digital de cinco funciones puede ser tuyo si pides tus programas al **Círculo de Soft**. Si tu compra es de dos programas te obsequiamos con: **Un conmutador TV-Ordenador...** y ambos regalos si pides tres programas.



y además...

GREMLINS	2.300 Ptas.
BEACH HEAD	2.700 Ptas.
COMBAT LINX	2.100 Ptas.
POLE POSITION	1.900 Ptas.
DRAGONTORE (prox.)	2.100 Ptas.
DECATHLON	2.100 Ptas.
KNIGH LORE (Novedad)	1.950 Ptas.
ALIEN 8 (Novedad)	1.950 Ptas.
KONG STRIKES BACK	2.100 Ptas.
HUNCHBACK II	2.100 Ptas.
ONE ON ONE (Novedad)	3.100 Ptas.
ROCKY	2.100 Ptas.
RAID OVER MOSCOW	2.100 Ptas.
BOXING	1.950 Ptas.

Nuestra dirección:
 Castellana, 268, 3.º C
 28046 MADRID
 Tel.: 733 25 00

CUPON DE PEDIDO

Recorta o copia este cupón, o pide tus programas por teléfono. Deseo recibir a vuelta de correo el(los) siguiente(s) programa(s).

TÍTULO _____ P.V.P. _____

Contrarrembolso Giro Postal Talón adjunto a Microamigo. S. A.

Nombre _____ Apellidos _____ Fecha caducidad _____

Domicilio _____ Localidad _____ Provincia _____ C.P. _____

Teléfono _____ Ebad _____

ORDENADOR

UN ORDENADOR MUY PROFESIONAL

H

ace poco, en estas mismas páginas, cuando informábamos a nuestros lectores del PC Show de Londres, comentamos la existencia de un nuevo modelo de **Amstrad** dirigido sin ambages al mundo profesional. Se trata del **PCW8256**, siglas de ordenador personal y procesador de textos.

A guisa de recordatorio, esta máquina tiene 256 Kbytes de memoria RAM, una unidad de disco integrada e impresora, funcionando todo ello bajo el sistema operativo CP/M Plus, y lo más importante, un programa procesador de textos, el **Locoscript**, que se entrega gratis al comprar el paquete, además de un Basic y un Logo.

La presentación oficial del nuevo ordenador se realizará en el hotel Meliá de Madrid el día 15 del mes de octubre, y el precio de venta al público de todo el paquete: monitor fósforo verde, teclado, unidad de disco e impresora será, según fuentes de Indescomp, de 159.900 ptas.

Primera plana

REESTRUCTURACION EN SINCLAIR RESEARCH

Sinclair Research, la compañía británica fundada por **Clive Sinclair** y creadora del Spectrum, ha sufrido profundos cambios en la composición de su directiva, como probable resultado de las dificultades financieras que han sufrido.

Sir Clive se dedicará a investigar nuevas tecnologías e ideas, pero abandona el puesto de presidente ejecutivo, que ocupaba hasta hace poco.

Al parecer, casi todo el mundo, incluido tal vez el propio señor Sinclair, piensan que su fuerte no es precisamente la dirección financiera de un negocio, sino desarrollar la labor creativa que corresponde al inventor e investigador.



Los PROBLEMAS DE APPLE

En 1977, **Wozniak** y **Jobs** fundaron los ordenadores personales y Apple Computer en un garaje. Ambos socios caminaron juntos durante algunos años, en los que Apple crecía hasta convertirse en una de las compañías de informática más fuertes del mundo.

Los problemas de Apple comenzaron con el proyecto **Macintosh**.

Jobs creía firmemente en él, pero **Wozniak** pensaba que era una locura. Resultado: el **Macintosh** nació y **Wozniak** abandonó Apple para dedicarse a construir lavadoras (!).

Comenzó a sonar entonces, junto al del superviviente **Jobs**, el nombre de **John Sculley**, director general de Apple, el cual no tardó mucho en tener series «diferencias» con el entrañable **Jobs**.

El resultado no ha sido una gran sorpresa: **Jobs** dimitió de su puesto de presidente de Apple y se marchó con idea de fundar una nueva compañía de informática, pero tuvo el detalle de llevarse consigo a los mejores cerebros de Apple Computer, prácticamente todos ellos comprometidos en los proyectos de investigación y desarrollo más secretos que imaginarse pueda.

Obviamente, **Sculley** y sus amigos echaban chispas ante la despedida de **Jobs**, y han manifestado a los cuatro vientos su inequívoca voluntad de hacerle polvo, perdón, de mandarle.

Para mayor «inria», dicen las malas lenguas que la postura de **Jobs** ante la amenaza es tirarse al suelo de risa, aunque él, lo que es él, no ha dicho esta boca es mía. Veremos en qué acuerdo acaba toda esta fanfarria.



ANIMACION

Seguro que alguna vez habréis usado algún juego comercial para Amstrad y os habéis sentido asombrados de los gráficos que usan y de su movimiento. ¿A que sí?



Indudablemente, es cierto. Pero el Basic Amstrad, potente como es, permite introducirse en el mundo de los dibujos animados con facilidad, aunque sin llegar a las cotas marcadas por el código de máquina.

Así vamos a examinar juntos, durante unos cuantos artículos, las posibilidades de animar nuestros programas de manera sencilla, haciendo uso de las sentencias Basic LOCATE y PRINT, cuyo estudio se realizó ya en nuestra revista, cuando hablábamos de las ventanas.

Los ingredientes de la animación

La animación simple puede reducirse a una receta, cuyos ingredientes son tres:

- Posicionamiento en pantalla
- Impresión de la figura
- Dejar pasar un período de tiempo adecuado (*retardo*) entre las dos acciones anteriores para dar sensación de continuidad y suavidad en el movimiento.

Combinándolas en el orden correcto, esto es, **posición, impresión y retardo**, el milagro sucede.

El programa 1 demuestra este teorema por el simple procedimiento de imprimir un mensaje en pantalla y desplazarlo hacia arriba.

Para este suceso, cómo no, existe una palabra técnica: **SCROLL** (del inglés *SCroll* **ROLL**, «**enrollar la pantalla**»), que conviene irse acostumbrando a utilizar.

Las líneas 30 y 40 posicionan (LOCATE) e imprimen (PRINT) el mensaje en la posición 10,14 (*horizontal, vertical*).

Observar atentamente el comedido de la línea 50: colocar el cursor de texto en la última línea de la pantalla, para preparar el terreno a la línea 60, la cual imprime un CHR\$(10) y la pantalla se desplaza una línea hacia arriba, «**matando**» lo que haya en la primera línea de la pantalla. Notar también que la pantalla sólo hace scroll cuando el cursor de texto está en la última línea.

El bucle FOR...NEXT de las líneas 60-90 pone en pantalla 20 «**blancos**» CHR\$(10), y fuerza a que el mensaje se desplace hacia arriba.

¿Y la línea 80? Bien, recordar el tercer ingrediente de la animación: el retardo. Si se omite esta línea y se ejecuta el programa, se verá muy claro por qué es imprescindible.

El programa 2 realiza scroll hacia abajo de un mensaje. El truco, como antes, es colocar el cursor de texto adecuadamente, en este caso en la PRIMERA línea de la pantalla, y recurrir a CHR\$(11), un código de control que intenta colocar de nuevo el cursor una línea más arriba de donde se encuentre.

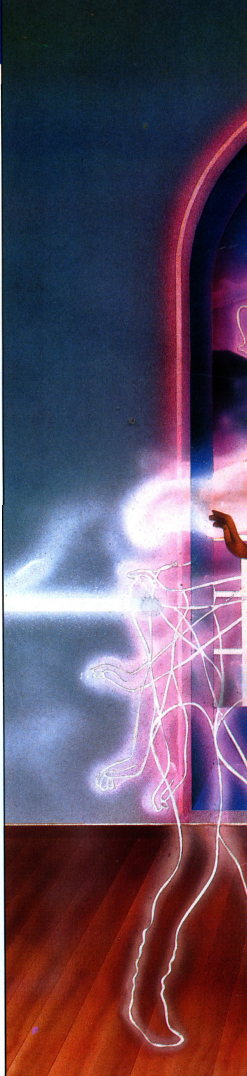
Sin embargo, como astutamente hemos colocado inicialmente el cursor en el tope superior de la pantalla, la única forma en la que el Amstrad puede cumplir lo que ordena el código de control, es desplazar la pantalla hacia abajo; así de simple.

Este método es sorprendentemente efectivo.

¿Alguien se atreve a escribir un programa que imprima un mensaje en el centro de la pantalla, lo desplace hasta la parte superior, luego a la inferior y vuelva al centro?

Si se consigue, es prueba concluyente de que lo anteriormente expuesto se ha captado a la perfección.

Hay casos, dentro de la animación, en que nos veremos obligados a usar otros métodos distintos; por ejemplo, si se quiere mover algo sobre un fondo fijo, el scroll no servirá para nada, por razones obvias.





Program Acción

En efecto, la pantalla, al desplazarse, destruye el fondo línea a línea.

Otras técnicas

Afortunadamente, existen formas de evitar esto. Ya que la mayor parte de las técnicas de animación implican mover varios gráficos de diferentes formas, vamos a comenzar con uno solo, como en los programas 3 y 4. Se distinguen en el sentido del movimiento del mismo gráfico.

En ambos, la línea 30 controla el movimiento en sentido horizontal; si la variable «bucle» se incrementa, el gráfico se desplaza hacia la derecha, si se decrementa, hacia la izquierda. Las líneas 40 y 50 posicionan el cursor y mueven el gráfico, mientras que el FOR...NEXT determina la coordenada X de impresión. La Y es la misma para ambos programas, de forma que el movimiento transcurra en la misma fila.

PRINT CHR\$(n) imprime el carácter cuyo código ASCII es n. Probar con otros números, de 32 a 255.

Todavía no tenemos lograda la verdadera animación, pues la figura deja un desagradable rastro al desplazarse. Estas dos líneas, añadidas al programa 3 ó 4, lo remediarán:

```
64 LOCATE bucle, 12
66 PRINT CHR$(32)
```

Sucede que, después de pintar el gráfico, la línea 66 pinta un espacio en blanco en la misma posición, borrándolo, y sugiriendo animación de verdad.

El método es: **pintar el gráfico, borrarlo y volverlo a pintar en la posición siguiente.**

Esta rutina de «ahora lo ves, ahora no lo ves» funcionará sólo si la diferencia entre sucesivas posiciones de impresión no es demasiado grande, y la variación de dicha posición es lo suficientemente rápida, aunque no MUY RAPIDA. Por eso la línea 66, la que borra, se ejecuta después del retardo de la línea 60. Probar ambos programas sin ella, demostrará una vez más la importancia del «tercer ingrediente».

PROGRAMAS

```
10 REM Programa 1
20 MODE 1
30 LOCATE 10,14
40 PRINT "Ejemplo de movimiento"
50 LOCATE 1,25
60 FOR bucle%=1 TO 20
70 PRINT CHR$(18);
80 FOR retardo%=1 TO 50:NEXT retardo
90 NEXT bucle%
```

```
10 REM Programa 2
20 MODE 1
30 LOCATE 10,14
40 PRINT "Ejemplo de movimiento"
50 LOCATE 1,1
60 FOR bucle%=1 TO 20
70 PRINT CHR$(11);
80 FOR retardo%=1 TO 50:NEXT retardo
90 NEXT bucle%
```

```
10 REM Programa 3
20 MODE 1
30 FOR bucle%=1 TO 40
40 LOCATE bucle%,12
50 PRINT CHR$(250);
60 FOR retardo%=1 TO 40:NEXT retardo
70 NEXT bucle%
```

```
10 REM Programa 4
20 MODE 1
30 FOR bucle%=40 TO 1 STEP -1
40 LOCATE bucle%,12
50 PRINT CHR$(251);
60 FOR retardo%=1 TO 40:NEXT retardo
70 NEXT bucle%
```

```
10 REM Programa 9
20 MODE 1
30 x=1;y=1
40 FOR derecha=1 TO 39
50 x=x+1
60 GOSUB 210
70 NEXT derecha
80 FOR abajo=1 TO 23
90 y=y+1
100 GOSUB 210
110 NEXT abajo
120 FOR izquierda=1 TO 39
130 x=x-1
140 GOSUB 210
150 NEXT izquierda
160 FOR arriba=1 TO 23
170 y=y-1
180 GOSUB 210
190 NEXT arriba
200 END
210 REM imprime grafico, espera y 1
o borra
220 LOCATE x,y
230 PRINT CHR$(248)
240 FOR retardo=1 TO 40:NEXT retardo
250 LOCATE x,y
260 PRINT CHR$(32)
270 RETURN
```



Para que sus datos no realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un casette mensual, solicitálos.

```
10 REM Programa 5
20 MODE 1
30 FOR bucle%=1 TO 39
40 LOCATE bucle%,12
50 PRINT CHR$(32);CHR$(250)
60 FOR retardo%=1 TO 40:NEXT retardo
70 NEXT bucle%
```

```
10 REM Programa 6
20 MODE 1
30 FOR bucle%=39 TO 1 STEP -1
40 LOCATE bucle%,12
50 PRINT CHR$(251);CHR$(32)
60 FOR retardo%=1 TO 40:NEXT retardo
70 NEXT bucle%
```

```
10 REM Programa 7
20 MODE 1
30 FOR vertical%=1 TO 24
40 LOCATE 19,vertical%
50 PRINT CHR$(249)
60 FOR retardo%=1 TO 60:NEXT retardo
65 LOCATE 19,vertical%
66 PRINT CHR$(32)
70 NEXT vertical%
```

```
10 REM Programa 8
20 MODE 1
30 FOR vertical%=24 TO 1 STEP -1
40 LOCATE 19,vertical%
50 PRINT CHR$(249)
60 FOR retardo%=1 TO 60:NEXT retardo
65 LOCATE 19,vertical%
66 PRINT CHR$(32)
70 NEXT vertical%
```

```
10 REM Programa 10
20 MODE 1
30 x=10;y=10
40 FOR arribaderecha=1 TO 5
50 x=x+1;y=y-1
60 GOSUB 210
70 NEXT arribaderecha
80 FOR abajoderecha=1 TO 4
90 x=x+1;y=y+1
100 GOSUB 210
110 NEXT abajoderecha
120 FOR izquierdaabajo=1 TO 4
130 x=x-1;y=y+1
140 GOSUB 210
150 NEXT izquierdaabajo
160 FOR izquierdaarriba=1 TO 4
170 x=x-1;y=y-1
180 GOSUB 210
190 NEXT izquierdaarriba
200 END
210 REM imprime grafico, espera y 1
o borra
220 LOCATE x,y
230 PRINT CHR$(248)
240 FOR retardo=1 TO 40:NEXT retardo
250 LOCATE x,y
260 PRINT CHR$(32)
270 RETURN
```

```
10 REM Programa 11
20 MODE 0
30 WHILE 1=1
40 LOCATE 10,12
50 PRINT CHR$(INT(4*RNDRND(1))+248)
60 FOR retardo%=1 TO 40:NEXT retardo
70 WEND
```

Sin ella, la rutina pasaría a ser del tipo «ahora no lo ves, ahora lo ves con un poco de suerte»; un desastre.

Los programas 5 y 6 también hacen el mismo trabajo, pero con una gran diferencia: la línea 50. ¿Se comprende por qué funciona?

Espero que sí, pero por si acaso, lo que pasa es que, a cada ronda del bucle, no sólo se pinta el gráfico, sino un espacio en blanco inmediatamente después de él. Al cambiar la posición de impresión, el espacio borra el gráfico. Así nos ahorramos las líneas 64 y 66 de los programas 3 y 4.

Nótese que el espacio viene antes del gráfico en el programa 5, y después en el 6, lógicamente.

¿Lógicamente? ¿Alguien puede decir por qué?

Movimiento en todas direcciones

Una vez comprendido todo esto, el movimiento arriba y abajo es un juego de niños. En lugar de modificar la coordenada X, cambiaremos la Y. Incrementándola, hacia abajo, decrementándola, hacia arriba.

De dos maneras los programas 7 y 8 lo hacen.

Se diferencia de los números 5 y 6 en que aquí tenemos que usar dos sentencias PRINT, líneas 50 y 65. Si no, no funcionarían (*compruébese*).

El programa 9 usa movimiento vertical y horizontal —no al mismo tiempo, claro— para desplazar un carácter alrededor de un cuadrado. La subrutina de las líneas 210-270 pinta, espera y lo borra. Es llamada cuatro veces, líneas 60, 100, 140 y 180, para cubrir las trayectorias derecha, abajo, izquierda y arriba, respectivamente.

Hasta ahora, el movimiento siempre ha sido en línea recta. Los desplazamientos en diagonal son muy fáciles de conseguir: simplemente hay que alterar a la vez las coordenadas horizontal y vertical (X e Y) de impresión (*programa 10*).

Esto es casi todo por ahora. Como se ha visto, la animación no es tan difícil como parece. Pronto, los colores llenarán de vida ese movimiento.

Para finalizar de veras, un poco de diversión en forma del programa 11. ¿Podéis predecir lo que hace?

COMPUTER CENTER

COMANDANTE ZORITA, 13
28003 MADRID

TELS.: (91) 233 07 35
(91) 233 07 81

AMSTRAD 464 Verde _____	57.900 ptas.
AMSTRAD 6128 _____	99.900 ptas.
DISKETTE 3" _____	895 ptas.
INTERFACE DISCO 5.25" _____	5.900 ptas.
CINTA C-15 ESPECIAL ORDEN _____	85 ptas.

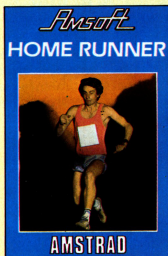
SOFTWARE ENTRETENIMIENTO (CASSETTE)

COMBAT LYNX _____	1.925 ptas.
DEAT PIT _____	1.925 ptas.
ALIEN-8 _____	1.875 ptas.
KNIGHT LORE _____	1.875 ptas.
HARRIER ATTACK _____	995 ptas.
JUMP JET _____	2.695 ptas.
SIMULAD. VUELO 737 _____	1.795 ptas.
FRUIT MACHINE _____	995 ptas.

LIBROS

CURSO AUTODIDACTICO BASIC AMSTRAD (Contiene manual y dos cassettes) _____	2.695 ptas.
HACIA LA INTELIGENCIA ARTIF. _____	1.300 ptas.
40 JUEGOS EDUCATIVOS _____	1.800 ptas.
MUSICA Y SONIDO PARA AMSTRAD _____	1.300 ptas.
PROGRAMANDO CON AMSTRAD _____	1.900 ptas.

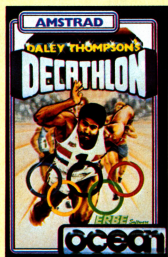
Tu pedido lo puedes recibir contra-reembolso (libre de gastos), llamando a los teléfonos (91) 233 07 35 y (91) 233 07 81.



1.495 Ptas



1.975 Ptas



1.975 Ptas



1.495 Ptas



1.795 Ptas.

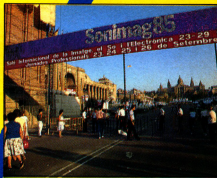


2.395 Ptas.

Sonimag 85

23 Salón Internacional de la Imagen, el Sonido y la Electrónica

La Feria del Sonido y de la Imagen siempre ha entrado en la categoría de «magnos certámenes» con toda la razón. En efecto, resulta un gran acontecimiento para el mundo del sonido y la imagen que los mayores fabricantes del sector presenten anualmente sus novedades y, al menos, parte de sus ideas y proyectos.



Esta máquina puede funcionar en modo 64 y 128 K, y se garantiza la absoluta compatibilidad del software Spectrum para el nuevo invento. Se ha remodelado el Basic y lleva un editor de pantalla completo, además de una puerta de comunicación serie y una interface MIDI, capaz de conectar con cualquier instrumento musical compatible con ese protocolo.

Hay más novedades de Investrónica, casi tan excitantes como el nuevo Spectrum 128 K.

La impresora BX 1000, por ejemplo, es una de ellas.

De matriz de puntos, puede alcanzar una velocidad de 100 caracteres por segundo y se conecta con el mundo exterior mediante una interface serie RS232, pensando en los Spectrum y en el QL. Como todas las impresoras, puede usar una gran cantidad de tipos de letra distintos, espaciado proporcional, etc.

QL: presente

Hablando del QL, este excelente ordenador ha sufrido hasta hace poco un trato desfavorable, de forma totalmente inmerecida.

En este mundillo, el ordenador ocupa un lugar un poco desplazado; la informática nunca ha sido la prima donna del Sonimag, y en la edición de este año, representaba aproximadamente un 10 por 100 de la extensión de la feria. No obstante, esta penuria de espacio, las marcas más importantes o al menos las que más sueñan, estaban presentes rivalizando por atraer la atención del público hacia sus stands.

La descripción detallada de toda la feria requería casi un libro en lugar de un artículo; vamos a intentar, no obstante, pasar revista a lo que sucedió en el Sonimag desde el punto de vista de la informática.

Lo nuevo de Sinclair

Sinclair Research, o lo que es lo mismo en España, Investrónica, presentó el flamante Spectrum 128 K.

Sinclair ha reaccionado rápidamente ante los augures que vaticinaban su inminente caída y, haciéndose eco de las exigencias actuales del mercado, se ha unido a la carrera de los 128 K con una excelente máquina, comparada con el Spectrum Plus, su antecesor.



En efecto, esta máquina, tecnológicamente hablando, es la más avanzada que existe de su categoría; tal vez los microdrives y la insistencia de presentarlo al público como un ordenador profesional hayan sido los responsables.

En este sentido, Investrónica aporta una unidad de disco para el QL, en formato de 3,5 pulgadas, capaz de formatear 720 Kbytes y un sustancioso aumento en los programas disponibles hasta ahora para el ordenador, consistentes casi exclusivamente en cosas como lenguajes, sistemas operativos y ensambladores. Tendremos disponible una contabilidad, un programa de historiales clínicos, etc.

Por último, Investrónica presentó también su aula informática, en la que un ordenador maestro controla hasta 64 Spectrum o QLs, pudiendo intercambiar información entre ellos de todo tipo.

La versatilidad digitalizada

Commodore estuvo presente naturalmente en Sonimag, con su nueva estrella, el Commodore 128 K.

Una vez más, se trata de una gran máquina, soberbiamente diseñada, conjuntando perfectamente funcionalidad y belleza, que tiene nada menos que tres procesadores distintos y es capaz de funcionar bajo el sistema operativo CP/M.

Posee las tradicionales buenas cualidades de los ordenadores Commodore en sonido y gráficos, junto con un Basic potente, sencillo de manejar (*para variar*), y cuenta con el aval de una de las marcas punteras del mundo de la informática casera y con el de poseer tal vez los mejores juegos o al menos los más cuidados y espectaculares que se han hecho nunca.

También tuvimos ocasión de admirar los periféricos que pueden conectarse a la máquina, entre otros una unidad de disco mucho más rápida que la malhadada 1501.

Atari: el no va más

Pasemos ahora a lo más revolucionario en tecnología que hemos visto en mucho tiempo, su nombre: Atari.

Atari, con la llegada de Tramiel (*fundador de Commodore*) **«al poder»**, se ha reformado completa-

mente y ha emprendido, esperemos que con éxito, la tarea de remodelar completamente su parque de ordenadores, bajo el lema: «Poder sin precio.»

Presentó dos modelos: uno de 128 Kbytes de RAM, el 130, excelentemente dotado para el manejo de gráficos y sonido, completamente compatible con todo el software Atari anterior y una completa gama de periféricos, entre unidades de discos, impresoras, etc.

Pero sin duda, la vedette de Atari es el modelo 520ST.

Esta máquina está inspirada en el concepto del Apple Macintosh y su concepción es absolutamente revolucionaria: un microprocesador de 32 bytes, 512 K de memoria RAM, un sistema operativo basado en iconos y ventanas, que hace innecesario el estudio de manuales para su manejo, una resolución gráfica y velocidad de proceso impresionantes, un diseño precioso, en fin, nos dejó verdaderamente atónitos. Es lástima que el 520ST cojee momentáneamente en el campo del software disponible para él.

Esperemos que muy pronto esta

carencia se remedie y las casas de software se animen a escribir aplicaciones usando el GEM, sistema operativo del Atari, serán dignos de verse.

Ni que decir tiene que este ordenador iba cumplidamente acompañado de los periféricos necesarios, a saber, impresoras y una unidad de disco de un Megabyte. Una jota.



Dragón: tecnología española

Dragón también estuvo presente: el Dragón 200 y el Dragón MSX constituyen unas interesantes novedades que deseamos tengan el éxito que merecen y potencien la investigación, desarrollo y fabricación de ordenadores en nuestro país.

Amstrad 128: la estrella (con perdón)

En cuanto a los ordenadores, nuestros lectores se preguntarán a qué esperamos para hablar de **Amstrad**. Bueno, el stand no era fácil de descubrir debido a la enorme cantidad de gente que lo cubría (*al menos, el día en el que nosotros estuvimos allí*), observando «**como locos**» toda la gama **Amstrad**, pero especialmente el 128 K, cómo no. De él no decimos nada más porque se le describió con profundidad en **AMSTRAD Semanal** número 3. Tuvimos oportunidad de observar un incremento significativo de programas para estas máquinas, especialmente software en disco.

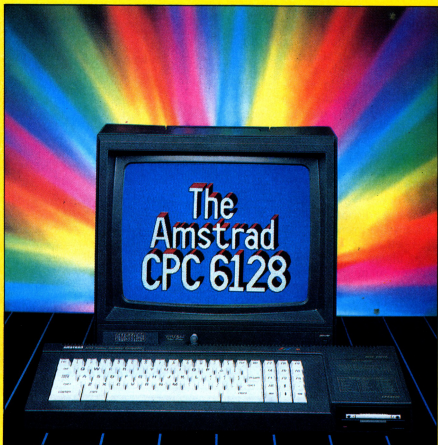
No todo iba a ser ordenadores en el Sonimag. Las casas de software tuvieron un activo protagonismo.

Imágenes por ordenador

Entre las muchas novedades que hubo, debemos destacar un gran programa creado para el Spectrum (*versión Amstrad en preparación*) por Pin S. A.: un digitalizador de imágenes.

Es decir, nosotros introducimos una fotografía en un aparato semejante a un scanner y dos minutos después aparece en la pantalla del Spectrum lista para ser manipulada casi, casi como queramos: ampliación vertical y horizontal, control de contrastes, etc.

El software responsable de esto puede funcionar en un entorno de cassette, microdrive o unidad de disco Discovery 1. Una gran idea a un costo razonable.



Tus copias no pastarán en mi rancho

La piratería también estuvo presente en el Sonimag, pero de forma un tanto insólita; una nota de prensa comunicaba que una serie de empresas de software se constituían en «**asociación empresarial**», con idea de cumplir varios fines etiquetados de muchas formas y resumibles en 4 palabras: muerte a los piratas.

En este controvertido tema, una iniciativa interesante. Veremos hasta qué punto es efectiva.

Recurriendo a un tópico, diríamos que «**Nada nuevo bajo el sol**».

No quisiéramos que esto se entendiera como un epitafio al Sonimag, ni como un comentario despectivo, pero la verdad es que ha habido pocas cosas realmente desconocidas. Lo que era nuevo en nuestro país, ha sido analizado y desmenuzado por las revistas especializadas, y en el Sonimag nos hemos podido dar el gusto de verlo con nuestros propios ojos.

No queda sino agradecer la existencia de las ferias informáticas y desear que se multipliquen rápidamente para beneficio del consumidor.



Sonimag 85

ELIPSES

En esta ocasión Análisis entra en el campo de la representación de funciones; más concretamente de funciones del tipo elipse.

Con este programa podemos situar el centro de la elipse en cualquier punto de la pantalla, definiendo además los ejes de la cónica. Dándole el mismo valor al eje horizontal que al vertical obtendremos una circunferencia.

10 REM, que contiene el título del programa.

20 Cambia el color de la tinta de amarillo a blanco.

30 Limpia la pantalla, para proceder a la representación de nuevas elipses.

40 Coloca el ordenador en el modo 2, de 80 caracteres por línea.

50 Hace que el ordenador calcule en grados el valor de los ángulos utilizados en nuestras operaciones.

60 Mensaje de información en pantalla.

70-80 Dos INPUTs que permiten introducir el valor de las coordenadas X e Y del centro de la elipse.

90 Otra línea de información en pantalla.

100-110 Los INPUTs que definen los valores de los ejes de la elipse.

120 Limpia de la pantalla toda la información anterior, dejándola lista para el trazado de la curva.

130-160 Un ciclo FOR...NEXT que varía

entre 0 y 360, con un intervalo de 0.6, es decir, que rota $360/0.6=600$ veces.

140 Inicializa el cursor gráfico en un punto de la pantalla, en caso de omitir esta instrucción, no obtendríamos ninguna representación gráfica.

150 Esta es la instrucción que verdaderamente dibuja la elipse, todo lo anterior han sido datos para su construcción e inicialización de pantalla.

Cada vez que oscila el ciclo FOR... NEXT, el ángulo a toma un valor creciente, que se incrementa en cada vuelta en 0.6 unidades, con el cual se calculan las coordenadas X e Y de cada punto de la elipse.

Al variar el ángulo desde 0 a 360 grados, cubrimos toda extensión de la cónica, cerrando el contorno de ésta.

170 Define la ventana de texto en la cual van a aparecer los datos representativos de la curva.

180 Especifica el color del papel de la ventana y el color de la tinta en la misma, el CLS final los coloca en la ventana.

180-200 Representan dentro de la ventana los datos más significativos de la elipse; posición de su centro, eje horizontal y eje vertical.

220-250 Dibujan los ejes de la cónica.

220 Inicializa el cursor gráfico, en la parte izquierda de la curva.

230 Dibuja el eje horizontal.

240 Coloca el cursor gráfico en la zona inferior de la cónica.

250 Dibuja el eje vertical.

260 Ciclo WHILE... WEND que mantiene en pantalla la elipse y sus ejes hasta que pulse una tecla, sin él los ejes de la cónica se borrarían inmediatamente.

270 Una instrucción de salto que hace que el programa vuelva a la línea 30, borrando la pantalla y permitiéndonos proceder a dibujar otra curva.



INCOGNITON

Programa realizado por el lector:
Leonardo Gil
(Murcia)

E

ste es un programa sin grandes complicaciones y con él se podrán resolver y revisar operaciones algebraicas, tipo ecuación.

Se puede abordar la resolución de ecuaciones de primer y segundo grado y una representación gráfica de dicho resultado casi perfecta.

Su manejo es muy sencillo: simplemente hay que poner a cada carácter algebraico el número que lo multiplica.

VARIABLES PRESENTACION

- d\$** = Lee datos de la línea 180.
- g\$** = Lee datos de la línea 340.
- sa** = Valor de un coseno.
- dr** = Valor de un seno.
- (Ubican la posición del cursor formando un círculo.)
- a** = Elige opción. Hay dos:
 - Ecuaciones de primer grado con dos incógnitas.
 - Ecuaciones de segundo grado.

VARIABLES PRIMERA OPCION

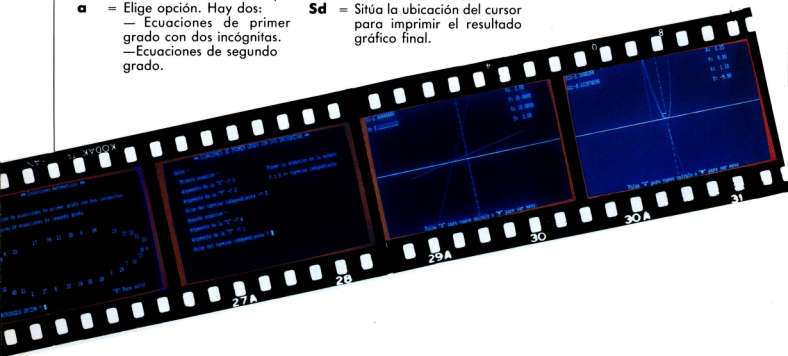
Ecuaciones de primer grado

- J\$** = Lee datos de la línea 630 a la 690.
- X** = Da valor al número que multiplica a la X de una de las ecuaciones del sistema.
- Y** = Da valor al número que multiplica a la Y de una de las ecuaciones del sistema.
- Z** = Da valor al término independiente de una de las ecuaciones del sistema.
- a** = Da valor al número que multiplica a la X de la segunda ecuación.
- b** = Da valor al número que multiplica a la r de la segunda ecuación.
- c** = Da valor al término independiente de la segunda ecuación.
- ZX** = Es el resultado final de la X.
- XZ** = Es el resultado final de la r.
- bu** = Son las coordenadas «Y» variantes de las dos rectas.
- mt**
- I** = Es la coordenada «X» variante de las dos rectas.
- Sd** = Sitúa la ubicación del cursor para imprimir el resultado gráfico final.

VARIABLES SEGUNDA OPCION

Ecuaciones de segundo grado.

- g\$** = Lee datos de la línea 1.270 a la 1.290.
- X** = Da valor al número que multiplica a la X al cuadrado.
- Xp** = Da valor al número que multiplica a la X.
- t** = Da valor al término independiente.
- bu** = Coordenada variante de la r de la parábola.
- cr** = Coordenada variante de la r de la recta.
- d** = Coordenada variante de la x de la recta y de la parábola.
- KI** = Coordenada r de la recta (dibujo).
- IQ** = Coordenada r del dibujo de la parábola.
- F** = Coordenada X del dibujo de la recta y de la parábola.
- Sd** = Ubica la posición del cursor para imprimir el resultado gráfico.



Serie Oro

```
10 GOTO 40
20 IF ERR=6 THEN CLS: CLEAR: RESTORE
630: LOCATE 29,12: PRINT "** VALORES
NO APTOS **": FOR GU=1 TO 3000: NEXT:
GOTO 440
30 IF ERR=11 THEN CLS: CLEAR: RESTORE
630: LOCATE 28,12: PRINT "** DIVISIO
N POR CERO **": FOR GU=1 TO 3000: NEX
T: GOTO 440
40 MODE 2
50 BORDER 6
60 INK 0,0
70 INK 1,26
80 LOCATE 30,1
90 FOR th=1 TO 103
100 IF th=23 THEN LOCATE 10,3
110 IF th=28 THEN LOCATE 10,5
120 IF th=72 THEN LOCATE 10,7
130 FOR kj=1 TO 12: NEXT
140 READ d$
150 PRINT d$;
160 SOUND 1,200,1,5
170 NEXT
180 DATA *,* E,c,u,a,c,i,o,n,e,s,m,
a,t,e,m,a,t,i,c,a,s*,*M,e,n,u,-,l
-,R,e,s,o,l,u,c,i,o,n d,e,e,c,u,a,
c,i,o,n,e,s,d,e,p,r,i,m,e,r,g,r,a,d
,o,c,o,n,d,o,s,i,n,c,o,g,n,i,t,a,s,
2-,R,e,s,o,l,u,c,i,o,n d,e,e,c,u,a
,c,i,o,n,e,s,d,e,s,e,g,u,n,d,o,g,r,
a,d,o
190 RESTORE 340
200 FOR a=1 TO 40
210 sa=40+(35*COS(a))
220 dr=16+(5*SIN(a))
230 IF dr<10 THEN 210
240 LOCATE sa,dr: PRINT a
250 SOUND 1,450,1,7
260 NEXT
270 LOCATE 60,24: PRINT "~0~ Para sa
lir"
280 LOCATE 10,24: FOR bu=1 TO 15
290 SOUND 1,300,1,4
300 FOR sl=1 TO 10: NEXT
310 READ g$
320 PRINT g$;
330 NEXT
340 DATA I,N,T,R,O,D,U,Z,C,A O,P,C,
I,O,N :
350 INPUT a
360 CLS
370 IF a=1 THEN RESTORE 630: GOTO 44
0
380 IF a=2 THEN RESTORE 1270: GOTO 1
50
390 IF a=0 THEN CLS: CLEAR: BORDER 1:
INK 0,1: INK 1,24: NEW
400 RESTORE 420
410 CLS: LOCATE 32,12: FOR th=1 TO 10
: READ g$: PRINT g$: SOUND 1,300,1,4,
0,0: FOR sl=1 TO 10: NEXT: NEXT: FOR s=
1 TO 1500: NEXT: RESTORE 180: GOTO 40
420 DATA N,o,c,o,m,p,r,e,n,d,o
```

```
430 GOTO 90
440 ORIGIN 0,0: CLS: LOCATE 15,1: ON E
RROR GOTO 20
450 FOR th=1 TO 212
460 READ j$
470 PRINT j$;
480 SOUND 1,300,1,4: FOR sl=1 TO 2: N
EXT
490 IF th=35 THEN LOCATE 45,6
500 IF th=55 THEN LOCATE 45,8
510 IF th=75 THEN LOCATE 5,3
520 IF th=80 THEN LOCATE 7,5
530 IF th=94 THEN LOCATE 7,7
540 IF th=108 THEN INPUT x: LOCATE 7
,9
550 IF th=121 THEN INPUT y: LOCATE 7
,11
560 IF th=146 THEN INPUT z: LOCATE 7
,13
570 IF th=161 THEN LOCATE 7,15
580 IF th=174 THEN INPUT a: LOCATE 7
,17
590 IF th=187 THEN INPUT b: LOCATE 7
,19
600 NEXT
610 INPUT c
620 RESTORE 630
630 DATA *,* E,C,U,A,C,I,O,N,E,S,D,
E,P,R,I,M,E,R,G,R,A,D,O,C,O,N,D,O,S
I,N,R,C,O,G,N,I,T,A,S*,*P,o,n,e,r
l,a,e,c,u,a,c,i,o,n e,n,l,a,m,a,n,e
,r,a,Y = X + t,e,r
640 RESTORE 650
650 DATA ,m,i,n,o,i,n,d,e,p,e,d,i,e
,n,t,e,D,a,t,o,s -,P,r,i,m,e,r,a,e,
c,u,a,c,i,o,n -
660 RESTORE 670
670 DATA ,A,r,g,u,m,e,n,t,o,d,e,l,a
~,X~, -,A,r,g,u,m,e,n,t,o,d,e,l,a
~,Y~, -,U,a,l,o,r,d,e,l t,e,r,m,i,n
,o,i,n,d,e,p,e,n,d,i,e,n,t,e -
680 RESTORE 690
690 DATA ,S,e,g,u,n,d,a,e,c,u,a,c,i
,o,n -,A,r,g,u,m,e,n,t,o,d,e,l,a~,
X~, -,A,r,g,u,m,e,n,t,o,d,e,l,a~,Y
~, -,U,a,l,o,r,d,e
700 RESTORE 710
710 DATA l t,e,r,m,i,n,o,i,n,d,e,p,
e,n,d,i,e,n,t,e :
720 n=(x)*(b): v=(b): m=(a)*(y): s
=(c)*(y)
730 h=- (v) + (s): j=- (n) + (m)
740 zx=- (h) / (j)
750 LOCATE 5,21: PRINT : "El resultad
o es X=": zx
760 xz=((x)*(zx))+(z)/(y)
```

```

770 LOCATE 21,22:PRINT "Y=";xz
780 FOR q=1 TO 3500:NEXT
790 CLS:LOCATE 30,12:PRINT "Represe
ntacion Grafica":FOR t=1 TO 2500:NE
XT:CLS
800 CLS
810 PLOT 320,0:DRAW 320,400:PLOT 0,
200:DRAW 640,200:ORIGIN 320,200
820 FOR xe=-200 TO 200 STEP 20
830 PLOT 1,xe:PLOT -1,xe
840 PLOT xe,2:PLOT xe,-2
850 NEXT
860 LOCATE 1,25:PRINT "Espere por f
avor"
870 sd=1
880 FOR w=-1000 TO 1000 STEP 10
890 f=1*20
900 kl=bn*20
910 hx=mt*20
920 l=w/100
930 bn=((x*1)+z)/y
940 mt=((a*1)+c)/b
950 IF bn>10 THEN 1030
960 IF bn<-10 THEN 1030
970 po=INT(zx*100)
980 FOR yt=1 TO 10
990 IF w=po+yt THEN GOTO 1080
1000 NEXT
1010 LOCATE 63,1:PRINT;"X=", USING
"###.##";l:LOCATE 63,3:PRINT;"Y=", U
SING "###.##";bn
1020 PLOT f,kl
1030 IF mt>10 THEN 1100
1040 IF mt<-10 THEN 1100
1050 PLOT f,hx
1060 LOCATE 63,5:PRINT;"X=", USING
"###.##";l:LOCATE 63,7:PRINT;"Y=",US
ING"###.##";mt
1070 GOTO 1090
1080 PLOT f,kl:DRAW f,0:PLOT f,kl:D
RAW 0,kl: LOCATE 1,sd:PRINT;"X=";zx
:LOCATE 1,sd+2:PRINT;"Y=";xz:sd=sd+
4:sOUND 1,20,20,15,0,0,1
1090 REM
1100 NEXT
1110 b$=INKEY$:IF b$="a" OR b$="A"
THEN RESTORE 630:GOTO 440
1120 IF b$="m" OR b$="M" THEN RESTO
RE :GOTO 40
1130 LOCATE 1,25:PRINT"
Pulsa ~A~ para nuevo calculo o
~M~ para ver menu."
1140 GOTO 1110
1150 CLS :LOCATE 26,1:ON ERROR GOTO
1410
1160 FOR th=0 TO 121
1170 READ g$:PRINT g$;
1180 IF th=22 THEN LOCATE 45,4
1190 IF th=42 THEN LOCATE 45,6
1200 IF th=63 THEN LOCATE 5,3
1210 IF th=68 THEN LOCATE 7,5
1220 IF th=83 THEN INPUT x:LOCATE 7
,7
1230 IF th=96 THEN INPUT xp:LOCATE
7,9
1240 SOUND 1,300,1,5:FOR sl=1 TO 3:
NEXT
1250 NEXT:INPUT t
1260 RESTORE 1270
1270 DATA *,* ,E,C,U,A,C,I,O,N,E,S D
,E S,E,G,U,N,D,O G,R,A,D,O *,*,P,o,
n,e,r,l,a,e,c,u,a,c,i,o,n,e,n,l,a,m
,a,n,e,r,a,X,^,2,+,- X +,- T,,i,n,
d,e,p,e,n,d,i,e,n,t,e
1280 RESTORE 1290
1290 DATA ,= 0,D,A,T,O,S -,A,F,g,u,
m,e,n,t,o,d,e,l,a~,X,~,2,~, -,A,r,g
,u,m,e,n,t,o,d,e,l,a~,X,~, -,U,a,l,
o,r,d,e,l,t,e,r,m,i,n,o,i,n,d,e,p,e
,n,d,i,e,n,t,e -
1300 zx=(-xp+SQR((xp^2)-(4*x*t)))/(
2*x)
1310 xz=(-xp-SQR(xp^2-(4*x*t)))/(2*
x)
1320 LOCATE 7,12:PRINT;"Los resulta
dos de la ~x~ son "zx"y ";xz
1330 FOR sd=1 TO 3500
1340 NEXT
1350 CLS:LOCATE 30,12:PRINT "Repres
entacion Grafica":FOR me=1 TO 2500:
NEXT:CLS
1360 CLS
1370 ORIGIN 0,0:PLOT 320,0:DRAW 320
,400:PLOT 0,200:DRAW 640,200:ORIGIN
320,200
1380 FOR xe=-200 TO 200 STEP 20
1390 PLOT 1,xe:PLOT -1,xe:PLOT xe,2
:PLOT xe,-2
1400 GOTO 1440
1410 IF ERR=5 THEN MODE 2:LOCATE 30
,13:PRINT "** DIVISION POR CERO **
":FOR H=1 TO 3000:NEXT:CLEAR:RESTOR
E 1270:GOTO 1150
1420 IF ERR=11 THEN MODE 2:LOCATE 2
9,13:PRINT "** DIVISION POR CERO **
":FOR H=1 TO 3000:NEXT:CLEAR:RESTOR
E 1270:GOTO 1150
1430 IF ERR=6 THEN MODE 2:LOCATE 30
,13:PRINT "** VALORES NO APTOS **":
FOR H=1 TO 3000:NEXT:CLEAR:RESTORE
1270:GOTO 1150
1440 NEXT
1450 LOCATE 1,25:PRINT "Espere por
favor"

```



Para que los dedos, no realicen el trabajo duro, M.H. AMS TRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette manual, solicitálos.

```

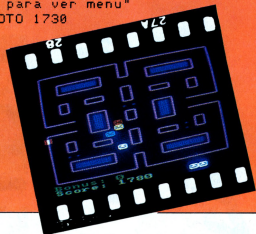
1460 sd=1
1470 IF x>0 THEN sm=zx:po=zx
1480 IF x<0 THEN sm=xz:po=zx
1490 FOR w=-800 TO 800
1500 l=w/100
1510 bn=x*(1^2)
1520 cr=-xp*(1)-t
1530 kl=bn*20
1540 lq=cr*20
1550 lu=INT(zx*100)
1560 hl=INT(xz*100)
1570 f=l*20
1580 IF w=lu THEN 1690
1590 IF w=hl THEN 1690
1600 IF bn>10 THEN 1640
1610 IF bn<-10 THEN 1640
1620 LOCATE 63,1:PRINT"X=", USING "
##.##";l:LOCATE 63,3:PRINT"Y=", USI
NG "##.##";bn
1630 PLOT f,k1
1640 IF cr>10 THEN 1720
1650 IF cr<-10 THEN 1720
1660 LOCATE 63,5:PRINT"X=", USING
"##.##";l:LOCATE 63,7:PRINT"Y=", U
SING "##.##";cr
1670 PLOT f,lq
1680 GOTO 1720
1690 IF sd=1 THEN PLOT f,k1:DRAW f,
0:PLOT f,k1:DRAW 0,k1:LOCATE 1,sd:P

```

```

RINT;"X1=";po:SOUND 1,20,20,15,0,0,
1
1700 IF sd=3 THEN PLOT f,k1:DRAW f,
0:PLOT f,k1:DRAW 0,k1:LOCATE 1,sd:P
RINT;"X2=";sm:SOUND 1,20,20,15,0,0,
1
1710 sd=sd+2
1720 NEXT
1730 b$=INKEY$:IF b$="a" OR b$="A"
THEN RESTORE 1270:GOTO 1150
1740 IF b$="m" OR b$="M" THEN RESTO
RE 180:GOTO 40
1750 LOCATE 1,25:PRINT;"
Pulsa "A" para nuevo calculo
o "M" para ver menu"
1760 GOTO 1730

```



MICROHOBBY AMSTRAD SEMANAL

**LE OFRECE AHORA SUS PROGRAMAS
YA GRABADOS, PARA QUE VD.
NO TENGA QUE TECLEARLOS**

Todos los programadores y aficionados a la microinformática sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta de cassette, sólo por **675 ptas. (sin más gastos por envío)**.

Envíanos con la menor demora posible, el cupón correspondiente.



Para que tus dedos no realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todas las listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicitánselo.

JUGGERNAUT

Conducir un trailer de 18 toneladas a través de las calles de una gran ciudad, no es tarea fácil para conductores inexpertos.

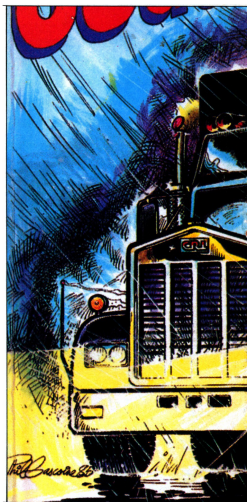
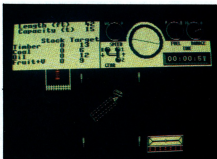
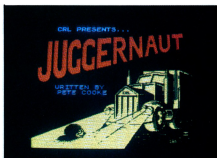
COMPATIBLE:
CPC 464
CPC 664

E

n este caso formamos parte de una gran compañía de transportes, como tal debemos recoger mercancías en distintos puntos de la ciudad y llevarlos a su destino, conduciendo nuestro supercamión.

Una vez pasada la moda de los controladores aéreos, las casas de software en su empeño por descubrir nuevos horizontes, hacen controladores de todo aquello que se puede conducir; coches de carreras, submarinos, helicópteros, máquinas de tren, motos, etc. Pero hasta ahora nadie lo había hecho con un camión de transporte.

En este caso, la acción de conducir un trailer pesado con las complicaciones que acarrea, controlando



velocidad y giros, es totalmente nuevo, el gran peso que arrastramos tras nosotros no permite frenadas ni giros bruscos.

La simulación está reproducida hasta tal punto, que controlamos volante, acelerador, freno y cambio de marchas.

Dentro de las distintas opciones que podemos seleccionar hay varias misiones claramente diferenciadas, Elegimos el tipo de camión y su longitud y el tonelaje máximo que puede transportar.

Podemos conducir o situarnos en el curso de prácticas.

Para los expertos existen opciones de alta puntuación, con dificultades extremas.

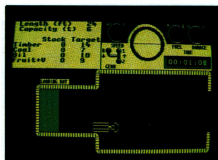
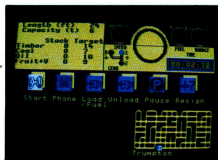
Por supuesto, también podemos definir las teclas a nuestro gusto con la comodidad que ello implica a la hora de manejar nuestro trailer.

Todo esto mediante un menú de opciones de tipo gráfico, con el uso de iconos.

Una vez definido todo esto, pasamos a la fase de acción. Tenemos



**FREE
POSTER
AND
STICKER
INSIDE**



cuatro modalidades: carretera, circulación por cruces, aparcamiento y manzanas.

En el juego nos constituimos en Lucas Traper, conductor para una pequeña firma de transportes, nuestro trabajo es sencillo, no requiere la utilización de la cabeza, solamente hemos de recoger varias mercancías en distintos puntos de la ciudad.

Carbón, aceite, madera, frutas y verduras constituyen nuestra carga.

Disponemos además de un mapa urbano, en el cual está indicada la posición de nuestra base de operaciones y la trama de calles de la ciudad.

Con todo esto cabría pensar que la cosa es bastante fácil, pero el problema radica en que debemos localizar a los distintos abastecedores.

¿Cómo hacerlo?

Usando el teléfono es la forma más sencilla y cómoda.

Para ello, tenemos que localizar una cabina y desde ella telefonar a nuestra oficina central.

El combustible también debe tener-

Mr. Joystick

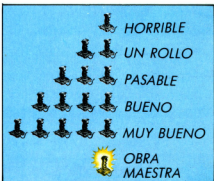
se en cuenta, un trailer de estas características consume grandes cantidades y debemos parar a repostar en las estaciones de servicio distribuidas al efecto. Las labores de carga, descarga, llenar el depósito de fuel, etc., también son controladas por iconos, facilitando gráficamente el desarrollo rápido de la acción y la comprensión de ésta.

Nuestro recorrido por las calles de la ciudad está jalonado de dificultades, glorietas, cruces de calles, carreteras estrechas, etc., pero además hemos de respetar el límite de velocidad de 30 MPH, en caso contrario la policía pondrá fin a nuestro recorrido.

Francamente un juego original, que maneja conceptos hasta ahora desconocidos, amenizado por la utilización de una buena dosis de estrategia, sin la cual servir a la pequeña empresa de transportes no es posible.

En todo momento hemos de tomar decisiones; el tipo de camión elegido al principio del juego es de gran importancia, grandes camiones son más difíciles de manejar pero transportan más mercancía.

Elegir la ruta adecuada, gastando la cantidad justa de combustible y procurando que el deterioro de nuestro trailer sea mínimo es la misión a cumplir.



BOULDER DASH

Nuestro pequeño amigo Rockford, es un hombre topo con una desmesurada inclinación hacia los objetos brillantes.

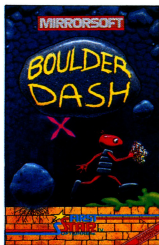
E

n esta ocasión, nuestro héroe se encuentra en las profundidades terrestres, vagando a través de misteriosas cavernas, en las cuales recoge los objetos a los que profesa devoción; todos los cuerpos brillantes y relucientes: diamantes.

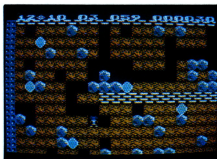
Pero extraer estas valiosas piedras de las profundidades no es tan fácil, la tierra se resiste a que le quiten sus frutos, el camino que atravesamos está repleto de grandes rocas, dispuestas a aplastarnos en cualquier momento, caminar debajo de ellas es muy arriesgado y ha de realizarse a toda velocidad, aprovechando el ligero temblor que se produce antes del desprendimiento.

Por si todo esto fuera poco, extrañas formas de energía subterránea también nos acosan, las profundidades subterráneas guardan terribles sorpresas para Rockford.

Lodos verdes que taponan las salidas y que se reproducen velozmente, paredes de sólida roca, mariposas voraces, etc., nos aguardan en los abismos subterráneos.



COMPATIBLE: CPC 464/CPC 664



Mr. Joystick



El programa contiene 16 cavernas diferentes de cada una de las cuales debemos encontrar la salida y recoger el máximo número de diamantes. El tiempo es un factor determinado a la hora de sumar bonus.

Las cavernas son por orden de aparición:

Intro, rooms, maze, butterflies, guards, firely dens, amoeva, enchanted wall, greed, tracks, crowd, walls, apocalypse, zigzag, funnel, enchanted box.

Cada una de las cuales posee sus propios moradores y peligros adicionales.

Los elementos con que nos encontramos en el juego:

Rockford, el hombre topo que posee el poder de escavar la tierra.

Rocas atrapadas por la tierra que caen cuando pasamos por debajo.

Diamantes, las piedras que atraen la atención de Rockford, sin ellas no podemos descubrir la salida de la caverna.

Núcleos de energía, atrapadas en las mismas cavidades que los diamantes, se introducen en los corredores y son mortíferos al contacto con nuestro hombre.

Mariposas, también altamente mortíferas, se convierten en diamantes al explotar.

Amebas, son manchas verdes que se extienden sin limitación por la caverna, a su contacto con los núcleos de energía y las mariposas se convierten en diamantes, Rockford puede controlar su crecimiento.

Muros mágicos, impenetrables para nuestro héroe, pero con la propiedad de que las piedras que caen sobre ellos se convierten en diamantes.

Muros de titanio, completamente indestructibles.

Zona de escape, se activa cuando el hombre topo ha recolectado el número preciso de diamantes, constituye el paso obligado hacia la próxima caverna.

Un típico juego sin complicaciones, con gráficos sencillos perteneciente a la primera generación de juegos para ordenador.



Suscríbete... y uno de estos tres sensacionales juegos será tuyo... ¡GRATIS!

Nos vemos obligados a sustituir, dentro de los juegos que ofrecemos a nuestros suscriptores, el **POLE POSITION** por el **COMBAT LYNX**, debido a que la empresa distribidora nos comunica la imposibilidad de obtenerlo en España a corto plazo.

Aquellas personas que hayan enviado la suscripción expresando su deseo de recibir el **POLE POSITION** recibirán, o habrán recibido ya, una carta en la que se facilita un cupón donde nos podrán hacer saber cual de los 3 juegos prefieren. Muchas gracias.

M.H. AMSTRAD te da a elegir entre tres de los mejores juegos existentes en el mercado para AMSTRAD; **COMBAT LYNX**, **DALEY THOMPSON'S DECATHLON** y **BEACH HEAD**, cualquiera de los cuales puede ser tuyo solamente con suscribirte a nuestra revista. **Aprovecha esta ocasión excepcional** y ahorra 2.100 pesetas (precio de venta del programa) más el importante descuento que se produce en el precio de cada número, por el hecho de ser suscriptor. Disfruta de las ventajas que supone recibir cómodamente tu revista a domicilio y de la seguridad de tener tu ejemplar aunque se haya agotado en los quioscos.

Enviamos tu boletín de suscripción y no le des más vueltas, el número de juegos para regalos de suscripción, aunque grande, es limitado, y estos se podrían agotar mientras lo estás pensando.

BEACH HEAD producido por U.S. GOLD es una misión de desembarco en una costa fuertemente defendida por las fuerzas aeronavales enemigas. Debes conducir tu flota hacia la bahía y repeler el ataque aéreo, si lo consigues tu siguiente obstáculo será una flotilla de destructores y acorazados, superada la cual desembarcarás tus anfibios en las arenas de la bahía, estos deben superar las defensas costeras y llegar a la fortaleza que es el objetivo final.

COMBAT LYNX simula una misión de defensa de unas bases atacadas por una división acorazada. Disponemos para enfrentarnos a ellos de un modernísimo helicóptero. Este juego podría incluirse dentro del catálogo de los de estrategia, y su complejidad le dota de una gran dosis de adicción y belleza.

DALEY THOMPSON'S DECATHLON con este juego OCEAN enciende la llama olímpica y te reta a superar los récords de los campeonatos de todos los tiempos, el decathlon se desarrolla en dos días de competición y se compone de las siguientes pruebas:
PRIMER DIA: 100 m lisos, salto de longitud, lanzamiento de peso, salto de altura y 400 m lisos.
SEGUNDO DIA: 110 m vallas, lanzamiento de disco, salto con pértiga, lanzamiento de jabalina y los 1.500 m.

COMPATIBLE
CPC 464
CPC 664



Utiliza el cupón
adjunto a la revista
o suscríbete por
teléfono
(91) 733 50 12
(91) 733 50 16

COMPATIBLE
CPC 464
CPC 664



SOLAMENTE
CPC 464



LOS BUCLES

Ni un solo programa funciona sin incorporar en su interior tareas repetitivas, bucles. Una de las mejores formas de hacerlo es usar los bucles estructurados, como el WHILE... WEND.

El comando INPUT permite al Amstrad inquirir información cuando un programa se está ejecutando.

Si el micro encuentra una sentencia del tipo:

```
10 INPUT nombre$
```

en medio de un programa, detiene su ejecución, imprime un signo «?» y espera hasta que le demos una respuesta.

En este caso, teclearíamos una serie de caracteres y, tras pulsar ENTER, la cadena sería asignada a la variable nombre\$, lista para ser procesada.

Resulta conveniente, al hacer una pregunta a la persona que se sienta frente al ordenador, imprimir un texto aclarativo de lo que estamos preguntando, así:

```
30 PRINT «Cuántos años tienes»;  
40 INPUT edad
```

Aún mejor, más económico desde el punto de vista de la memoria empleada, sería incluir dicho mensaje en la propia sentencia INPUT. Esto se hace de la siguiente manera:

```
10 INPUT «Cuántos años tienes; edad  
sencillo, ¿verdad?
```

Obsérvese, sin embargo, tres puntos importantes:

1. El mensaje se inserta entre el comando INPUT y la variable.
2. El mensaje va entre comillas, es decir, es un literal o cadena de caracteres.
3. Hay un punto y coma entre las comillas que dan fin al mensaje y la variable.

Todo esto, en forma de listado, está en el programa 1.

Análogamente, el programa «multiplicador» de la semana pasada, se convierte en el programa 2.

Y, por supuesto, este método también trabaja con variables de cadena (programa 3).

Antes de dejar de lado el comando, conviene comentar que puede utilizarse más de una variable en la misma sentencia, como muestra el programa 4.

El ordenador esperará hasta que le introduzcamos dos números.

Si tecleamos el primero y pulsamos ENTER, nos veremos sorprendidos por un mensaje de error: «?Redo From Start», porque no le hemos dado al micro un número suficiente de datos para llenar las dos variables que «guardan cola».

Debemos dar dos valores separados por una coma, del mismo modo que las variables se encuentran separadas en la línea 30 del programa 4.

Una vez hecho, se ve que el primer dato se almacena en la variable «primera» y el segundo en la variable «segunda».

Ni que decir tiene que otro tanto ocurre con las variables de cadena.

Colocar varios variables en una sentencia INPUT es cuestión de gustos: se ahorra memoria, pero el programa resulta menos claro.

Por último, el uso de las comas como separadores, explica por qué, en respuesta a una sentencia del tipo:

```
10 INPUT number  
no podemos responder 1,000 sino 1000. Tratar usando la coma y veréis que divertido.
```

While... Wend

Vamos a introducirnos ahora en algo completamente distinto. Sin duda se recordarán programas del tipo del programa 5, es decir, bucles sin fin o bucles incondicionales.

Hay por lo menos otra manera de obtener bucles incondicionales, la del programa 6.

Al ejecutarlo, la palabra «hola» será impresa repetidamente hasta que se pulse la tecla ESC.

Los comandos WHILE (en español mientras) y WEND (abreviatura de While END, «fin del mientras»), marcan el comienzo y el

final de un sector de programa que debe ser repetido.

A este sector le llamaremos el Cuerpo del bucle.

Naturalmente, no podemos poner en una sentencia un WHILE sin más. El Amstrad, ni corto ni perezoso, pregunta «mientras qué?» a su propia manera: enviando un mensaje de error y deteniendo la ejecución del programa.



Primeros pasos



Hay que colocar una expresión detrás del comando WHILE, que el **Amstrad** evaluará para saber si es cierta o falsa, esto es, si es 1 ó 0. La sección de programa marcada con WHILE... WEND se ejecutará mientras la expresión sea cierta. Cuando sea falsa, el flujo del programa pasa a la sentencia inmediatamente siguiente a WEND.

La línea 30 del programa 6 dice:

30 WHILE 1=1

Literalmente le estamos diciendo al **Amstrad**: «*hazlo mientras uno sea igual a uno*».

¿Lógicamente?

Resulta obvio que «*uno igual a uno*» es una verdad como un templo, y así lo interpreta el ordenador. No en vano hablábamos de bucles sin fin; éste se seguiría ejecutando hasta el próximo corte de luz.

El programa 7 es virtualmente idéntico al 6, excepto por la línea 30, que ahora dice:
30 WHILE -1

Al correrlo se obtendrá exactamente el mismo resultado, sorprendentemente.

Algo menos de sorpresa experimentaremos si tecleamos:

PRINT 1=1

Aparecerá -1.

La razón es que a las computadoras les en-

PROGRAMAS

cantan los números, y el **Amstrad** asigna al concepto verdad o cierto el valor -1. Por tanto, en ambos programas la línea 30 significa «mientras verdades».

Si, ya sé que eso no nos dice que es verdad o cierto, pero para el ordenador todo es cierto mientras no se demuestre lo contrario, y en el cuerpo del bucle se repite indefinidamente.

Por otro lado, al teclear:
PRINT 1=2

se verá un rotundo 0 en la pantalla.

Al cambiar la línea 30 por:
30 WHILE 0

queremos repetir un trozo de programa Básic mientras (lo que sea), sea falso. Por supuesto, el bucle no se ejecuta ni una vez.

En respuesta al ingenioso razonamiento de que es cierto que 0 es falso, y que, por tanto, el micro debiera ejecutar el bucle hasta el fin de los tiempos, debo decir que tan alambicados sutilezas el ordenador no es capaz de «captarlas» de una sola vez. La máquina no se para en barras: encuentra un cero, deduce que lo que sea es falso, no lo hace y basta.

Habría que añadir una determinada instrucción para que se percatara, por ahora fuera de nuestro estudio.

A mí, personalmente, los números no me gustan demasiado. Probablemente sea más legible emplear palabras más significativas, haciendo uso de variables, como el programa 8. Como condición de un bucle WHILE puede ir cualquier expresión evaluable, por complicada que sea, como por ejemplo ésta:

```
WHILE haya-página  
  READ página-siguiente  
  WEND
```

Esta rutina «leería» un libro hasta que no hubiera más páginas, admitiendo que READ página-siguiente fuera una subrutina que realizara las funciones necesarias, pero la lógica sería esa.

Bucles condicionales

El programa número 9 usa un bucle WHILE... WEND que evalúa una condición muy simple: las líneas 40 y 50 del programa serán repetidas hasta que la variable fin\$ sea igual a «si», o, lo que es lo mismo, hablando en términos del propio bucle, mientras la variable fin\$ sea distinta de «si», se repite el proceso.

Aunque esta variable sea mentada por primera vez en la línea 30, el **Amstrad**, la hace igual a cadena vacía, esto es, a nada, automáticamente. Por eso no necesitamos definirlo previamente, y, además, sabemos que en la primera iteración del bucle la condición especificada no se cumplirá; al menos se ejecutará una vez.

Exactamente lo mismo habría ocurrido con una variable numérica. Atención ahora al programa 10. En él, inicializamos la variable número a 0 (línea 30) e imprimimos su valor (línea 40).

PRIMEROS PASOS, NÚMERO 7

```
10 REM Programa 1  
20 HODE 1  
30 INPUT "Cuantos años tienes?edad"  
40 PRINT "No creo que tengas *edad"
```

```
10 REM Programa 2  
20 HODE 1  
30 INPUT "Primer número":primero  
40 INPUT "Segundo número":segundo  
50 PRINT primero * multiplicado por * segundo " da " primero*segundo
```

```
10 REM Programa 3  
20 HODE 1  
30 INPUT "Cual es tu nombre":nombre  
40 PRINT "Encantado de conocerte, " nombre$
```

```
10 REM Programa 4  
20 HODE 1  
30 INPUT "Dime 2 numeros":primero,segundo  
40 PRINT primero * multiplicado por * segundo " da " primero*segundo
```

```
10 REM Programa 5  
30 PRINT "hola"  
40 GOTO 30
```

```
10 REM Programa 6  
20 HODE 1  
30 INPUT "1=":  
40 PRINT "hola"  
50 WEND
```

```
10 REM Programa 7  
20 HODE 1  
30 WHILE -1  
40 PRINT "hola"  
50 WEND
```

```
10 REM Programa 8  
20 HODE 1  
30 verdares=1  
40 WHILE verdares  
50 PRINT "hola"  
60 WEND
```

```
10 REM Programa 9  
20 HODE 1  
30 WHILE fin$(*)="si"  
40 PRINT "Esto es un bucle WHILE...WEND condicional"  
50 INPUT "Quieres acabar?fin$"  
60 WEND  
70 PRINT "Ya me voy."
```

```
10 REM Programa 10  
20 HODE 1  
30 numero=0  
40 PRINT "Número es "numero  
50 numero=numero+1  
40 PRINT "El nuevo número es "nuevo numero
```

```
10 REM Programa 11  
20 HODE 1  
30 numero=0  
40 PRINT "Número es "numero  
50 numero=numero+1  
60 PRINT "El nuevo número es ahora " numero
```

```
10 REM Programa 12  
20 HODE 1  
30 numero=0  
40 WHILE -1  
50 numero=numero+1  
60 PRINT "El número es ahora " numero  
70 WEND
```

Entonces le sumamos 1 y atesoramos el resultado en la nueva variable «nuevonúmero», imprimiéndolo (líneas 50-60).

Con la línea 50 del 10 en la mente, examinemos el número 11. El trabajo que realiza es prácticamente el mismo que el programa anterior, y su lógica está basada en él. Hemos hecho, sin embargo, un cambio importante, también en la línea 50.

Vamos a analizarla un poco, para tratar de comprender, entre otras cosas, lo que significa una igualdad para un ordenador:

50 número= número+1

Desde un punto de vista estrictamente matemático, esta sentencia no tiene sentido. No se puede decir que algo es igual a sí mismo más uno, como si dijéramos, por ejemplo, 5=5+1.

Sin embargo, aquí se esconden dos importantes claves:

1. **El micro evalúa en primer lugar la parte derecha de la expresión**, esto es, (número+1). La primera vez que la máquina llega a la línea 50, mira el valor de número, que es cero, y le suma 1. Tenemos pues, que la parte derecha vale uno.

2. **Para el Amstrad, el signo «=», significa asignación en lugar de igualdad, normalmente.** Toma, por tanto, el valor de la parte derecha, 1, y lo asigna a la variable número. Como el valor anterior de número era cero, ahora será 1. Si aquí hubiera un bucle, se repetiría el cálculo, la segunda pasada asignaría a número el valor 2, resultado de tomar el valor actual (1) y sumarle 1.

En definitiva, la línea 50 aumenta el valor de la variable número en una unidad.

El ordenador no tiene ningún problema en tomar el valor de una etiqueta (variable), usarlo para un cálculo y cambiar el valor de esa misma etiqueta. **Por eso a las variables se las llama, precisamente, variables.**

El programa 12 incluye un bucle WHILE...WEND que imprime repetidamente el valor de número, una y otra vez.

Podrías alterarlo para que los números que aparecen en pantalla fueran de 4 en 4, y también para que empezara por el número 1000 y fueran descendiendo de 1 en 1?

Naturalmente, podemos controlar el trango de los números que se imprimen, alterando la línea 40, por ejemplo:

40 WHILE número[24]

Se verán los números del 1 al 24.

Si ahora hacemos que la línea 50 sea la 60, y la 60 la 50, intercambiándolas, la salida en el monitor es notablemente distinta. Podrías explicar por qué?

Una última cosa: la línea 30 de los programas 10, 11 y 12 es superflua; como ya sabemos, para el ordenador todas las variables son cero mientras no se le indique lo contrario. Se ha puesto simplemente por claridad.

Próximamente seguiremos analizando las características de los bucles.

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

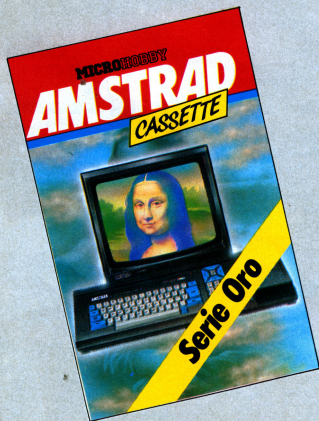
Semanal

LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS, PARA QUE VD. NO TENGA QUE TECLEARLOS

Todos los programadores y aficionados a la microinformática sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta cassette, sólo por **675 ptas.** (sin más gastos por envío).

Programas incluidos en la cinta número 1

Título	Revista número
EASYDRAW	1
EGGBLITZ	2
CODIGO SECRETO	2
VENTANAS	2
BIORRITMOS	3
MAD ADDER	3
HEXER	3
CHARGEN	4
PROGRAMACION	4



Enviamos con la menor demora posible, el cupón correspondiente.

USO DE REGISTROS COMO PUNTEROS

Los registros del Z80 pueden combinarse para formar registros dobles, BC, DE, HL, capaces de manipular números de 16 bytes.

Podemos cargarlos con valores constantes:
LD BC, &2FF8
que, en opcodes, se traduce en:
01 F8 2F

de la forma acostumbrada, es decir, byte bajo, byte alto.

También son muy comunes las instrucciones del tipo:

LD BC, (&2FF8)

en donde &2FF8 no indica aquí una constante, sino un puntero que señala a una determinada posición de memoria. Después de ejecutar esta instrucción, cuyos opcodes son:

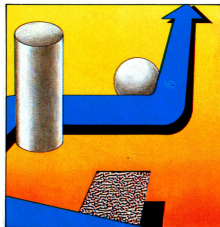
ED 4B F8 2F

el registro C contendrá el número almacenado en &2FF8 y B el que haya en &2FF9. Como se puede ver, esto es una suerte de peek de dos bytes a un tiempo. Resultará extremadamente útil.

La facultad de los registros dobles de manipular un número entre &0 y &FFFF, nos viene que ni pintada para apuntar a, o, más técnicamente, *indexar*, cualquier posición de memoria.

Por ejemplo, si hemos cargado HL con &2FF8 (LD HL, &2FF8), la instrucción:

LD A, (HL)



cargará el acumulador con el contenido de dicha posición de memoria, es decir, un peek de un byte *extraído de donde apunta HL*.

Lógicamente, la operación inversa:

LD (HL), A

existe, e incluso funciona.

DIRECCION	OPCODE	MNEMONICOS
3000	21 F8 2F	LD HL, &2FF8
3003	3E FF	LD A, &FF
3005	77	LD (HL), A
3006	C9	RET

Programa 1

El programa número 1 muestra el mecanismo de todo esto.

Tras ejecutarlo, Hexer debería mostrar que &2FF8 contiene &FF.

Línea por línea, ocurre lo siguiente:

- Línea 1: HL apunta (*indexa*) a &2FF8.
- Línea 2: el A contiene &FF.
- Línea 3: el contenido de la dirección apuntada por HL pasa a almacenar &FF.
- Línea 4: retorno al Hexer.

Indexación de memoria

Alguien, con toda la razón, se puede sorprender de por qué no se ha escogido la técnica que muestra el programa 2 para cumplir nuestro propósito.

DIRECCION	OPCODE	MNEMONICOS
3000	3E FF	LD A, &FF
3002	32 F8 2F	LD (&2FF8), A
3005	C9	RET

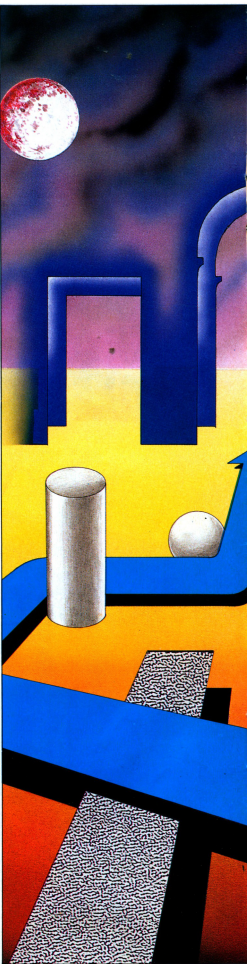
Programa 2

El motivo es que, con el método del programa 1, podemos beneficiarnos de las instrucciones:

INC HL
DEC HL

para movernos arriba y abajo de la posición inicial apuntada por HL, y

F. L. Frontón



Código máquina

DIRECCION	OPCODE	MNEMONICOS
3000	21 F8 2F	LD HL, &2FF8
3003	3E FF	LD A, &FF
3005	77	LD (HL), A
3006	23	INC HL
3007	77	LD (HL), A
3008	23	INC HL
3009	77	LD (HL), A
300A	C9	RET

Programa 3

alterar sus contenidos si así lo queremos, es decir, podemos *indexar* una determinada zona de la memoria y acceder a ella muy fácilmente, como se ve en el programa 3.

Acabamos de llenar tres posiciones consecutivas de memoria, de &2FF8 a &2FFA inclusive, con &FF.

Para averiguar hasta donde hemos comprendido este sencilla mecanismo, no estaría de más intentar escribir un programa gemelo del 3 que llenara 3 posiciones de memoria, partiendo de &2FFA, pero hacia atrás, hasta alcanzar &2FF8.

Otro ejercicio interesante: poner los valores 0, 1 y 2 en posiciones consecutivas partiendo de &2FF8 (acordaros de la instrucción *INC A*, opcode &3C).

LD B,n	06 n
LD C,n	0E n
LD D,n	16 n
LD E,n	1E n
LD H,n	26 n
LD L,n	2E n
LD (HL),n	36 n
LD A,n	3E n

Tabla 1: LD r,n opcodes

Resulta que, para el Z80, (HL) se convierte en una especie de «registro extra» de un byte, es decir, a menudo las instrucciones como *INCrementar* y *DECrementar* y otras, pueden aplicarse al contenido de la dirección de memoria a donde HL apunta, como en:

```
INC (HL)
DEC (HL)
ADD A,(HL) (Opcode &86)
SUB (HL) (Opcode &96)
```

Los códigos de operación de ellas llenan los huecos en las tablas que vimos anteriormente, por ejemplo la 1.

Los opcodes están relacionados

Recordaréis que los opcodes seguían una cierta regla, aumentaban

de 8 en 8 (*insistimos en que se examinen en binario*), salvo que de &2E se pasaba bruscamente a &3E. Falta, claro, el &36, opcode correspondiente a:

LD (HL),n

Análogamente sucede con la Tabla 2, que ahora ya está completa, salvo por el asterisco colocado en la que sería la instrucción:

LD (HL), (HL)

que no tiene ningún sentido.

Modos de direccionando

Entre la jerga de uso común en ordenadores, que casi nadie comprende, y casi nadie quiere comprender, programadores incluidos, estos malabarismos con contenidos de memoria apuntados por un registro se dice que es un *modo de direccionamiento*, nada menos, y le llaman *indirecto*, lo crean o no, porque no tenemos que decirle al micro directamente qué posición de memoria llama nuestra atención. Dejamos al Z80 que lo encuentre, que para eso le pagan.

DIRECCION	OPCODE	MNEMONICO
3000	21 F8 2F	LD HL, &2FF8
3003	36 FF	LD (HL), &FF
3005	C9	RET

Programa 4

DIRECCION	OPCODE	MNEMONICOS
3000	21 F8 2F	LD HL, &2FF8
3003	36 FF	LD (HL), &FF
3005	23	INC HL
3006	C9	RET

Programa 5

DIRECCION	OPCODE	MNEMONICOS
3000	21 F8 2F	LD HL, &2FF8
3003	36 FF	LD (HL), &FF
3005	34	INC (HL)
3006	C9	RET

Programa 6

El programa 4 usa direccionamiento indirecto para cargar &2FF8 con &FF. Obsérvese la analogía y la diferencia con el programa número 1.

Atención a los programas 5 y 6, porque existe entre ellos una diferencia fundamental que se descubrirá por sí sola al ejecutarlos en el Hexer y examinar los resultados. No insistimos más en ello porque la diferencia entre posiciones de memoria y contenidos de memoria se discutió ampliamente en el artículo inmediatamente anterior a éste.

		r							
		B	C	D	E	H	L	(HL)	A
r	B	40	41	42	43	44	45	46	47
	C	48	49	4A	4B	4C	4D	4E	4F
	D	50	51	52	53	54	55	56	57
	E	58	59	5A	5B	5C	5D	5E	5F
	H	60	61	62	63	64	65	66	67
	L	68	69	6A	6B	6C	6D	6E	6F
	(HL)	70	71	72	73	74	75	*	77
	A	78	79	7A	7B	7C	7D	7E	7F

Tabla 2: opcodes para LD r,r'

Una vez examinados ambos programas, sin duda podréis predecir los resultados del programa 7. He-xer resultará de ayuda.

DIRECCION	OPCODE	MNEMONICOS
3000	21 F8 2F	LD HL, &2FF8
3003	34	INC (HL)
3004	23	INC HL
3005	34	INC (HL)
3006	23	INC HL
3007	34	INC (HL)
3008	C9	RET

Programa 7

El registro HL es el favorito de los dobles del Z80. No se pueden realizar las tareas vistas anteriormente con BC y DE; hay que usar el acumulador como intermediario, como muestra la Tabla 3. Como ejemplo del uso de estas instrucciones, mirad el programa 8. No hay que asustarse; en realidad, hace algo muy simple: copia los tres primeros bytes del propio programa 8 en el espacio de trabajo del Hexer, para lo cual se utilizan DE y HL como punteros.

HL apunta al «origen» de los datos, &3000, y DE al «destino». El acumulador se carga con el byte apuntado por HL, mediante:

LD A,(HL)

y el valor es enviado a la posición indexada por DE:

LD (DE),A

Ambos registros son INCrementados en uno para apuntar a los siguientes origen y destino.

El proceso se repite hasta que los

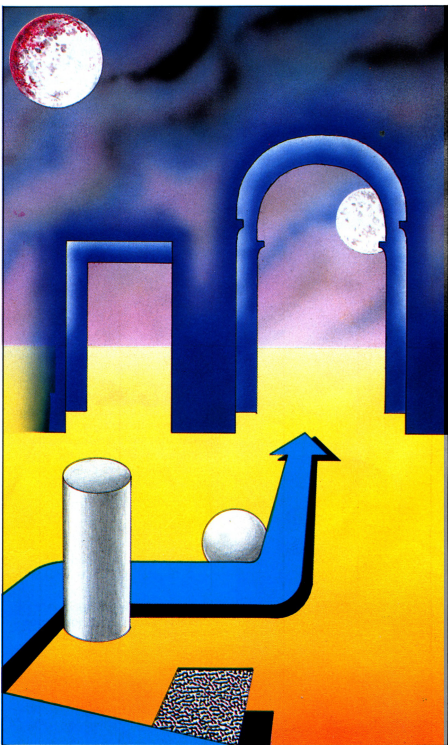
DIRECCION	OPCODE	MNEMONICOS
3000	21 00 30	LD HL, &3000
3003	11 F8 2F	LD DE, &2FF8
3006	7E	LD A, (HL)
3007	12	LD (DE),A
3008	13	INC DE
3009	23	INC HL
300A	7E	LD A, (HL)
300B	12	LD (DE), A
300C	13	INC DE
300D	23	INC HL
300E	7E	LD A, (HL)
300F	12	LD (DE), A
3010	C9	RET

Programa 8

MNEMONICOS	OPCODE
LD (BC),A	02
LD (DE),A	12
LD A, (BC)	0A
LD A, (DE)	1A

Tabla 3: direccionamiento indirecto con BC, DE

tres bytes han sido copiados. Estos ejemplos, aunque sencillos, están muy lejos de ser triviales, porque las técnicas descritas aquí tendrán que ser utilizadas sin descanso en cuanto se pretenda desarrollar un programa en lenguaje máquina.



SMALEY

Serie Oro

Nos encontramos con un programa clásico, responsable en gran medida de la existencia en el hogar de los ordenadores: un comecocos, uno de los primeros tipos de videojuegos.

S

e trata, como es sabido, de un laberinto colorista y compulsivo en el que tenemos que comer la máxima cantidad posible de «botonitos», mientras somos implacablemente perseguidos por el ominoso ejército de Grumpis, ávidos de sangre, de nuestra sangre concretamente.

Y como esto es el cuento de nunca acabar, cuando hemos completado un laberinto, orgullosos de nuestra habilidad, resulta que hay que comenzar otro; la sádica diferencia estriba en que los hambrientos Grumpis se mueven mucho más deprisa. Son como los bancos, están cada vez más cerca.

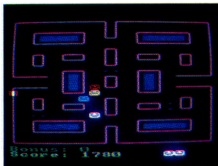
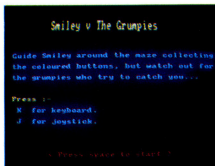
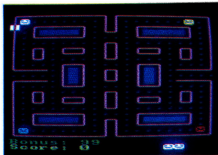
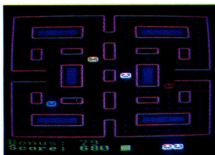
El programa, faltaría más, está lo más estructurado posible: los GOTO son tabú y no se ponen en la medida de lo posible.

El cuerpo principal (líneas 40-210) maneja una serie de subrutinas que imprimen las instrucciones, dibujan el laberinto y mueven los caracteres (nos referimos a los gráficos).

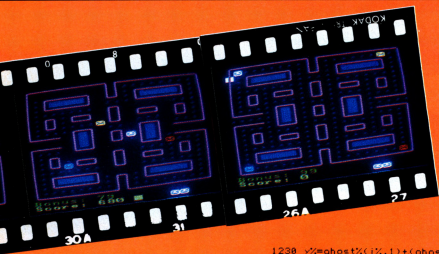
Cada subrutina está adecuadamente provista de una sentencia REM para poder identificarla con toda facilidad.

SUBROUTINAS

Interrupción 1	Usamos el «temporizador» 1 para generar una interrupción y ajustar el reloj.
Interrupción 2	El temporizador 0 es el contador de bonos. Alcanza el cero en 100 segundos. Los bonos se imprimen en la ventana número 1, por lo que no estorban a los otros caracteres.
Inicialización	Dimensiona las matrices, asigna colores y establece la máxima puntuación. K% es la tecla \uparrow y.
Dibuja laberinto	Lee las datas y añade 82 al código ASCII de cada carácter de las mismas antes de imprimirlos, ... de los Grumpis y de nuestro héroe, Smaley.
Posición inicial Asignación	Puntuación y bucles de retardo a valores iniciales. Imprime puntuación y tres gráficos.
Mueve hombre	Si es que hay que moverlo, si no, retorna. Encuentra las nuevas coordenadas de movimiento y pone a 1 el flag ok si es la misma que una de los Grumpis. Barra la figura anterior y pinta la nueva. También, si hemos comido algo, incrementa la puntuación.
Mueve Grumpis Fantasma	Pues eso, los mueve. Esta rutina fantasma mira dónde se encuentra Smaley y mueve los Grumpis hacia él si no hay una pared en el camino. Analiza si Smaley ha sido capturado y repone la comida cuando es necesario.
Cogido Instrucciones Siguiente pantalla	Hace algo de ruido y decrementa las vidas. Obviamente... Imprime la pantalla en modo transparente. Suma los bonos adquiridos a los puntos e incrementa la pantalla. Limpia la pantalla, imprime la puntuación y pregunta si deseamos jugar de nuevo.
Fin	



Serie Oro



```

990 PEN 14:LOCATE 14,25:PRINT CHR$(
224):CHR$(224):CHR$(224):
1000 gde lay%(0)=20:gdelay%(1)=40:gd
elay%(2)=80
1010 mdelay%iscore%0=0:bonus%0=isc
reen%1:ives=3
1020 RETURN
1030 :
1040 REM ** Mover hombre **
1050 IF mttime%< THEN RETURN
1060 mttime%=(mde lay%+x0manx%-(IN
KEY$(0))-1)+(INKEY$(0))-1:y0manx%
=(INKEY$(0))-1+(INKEY$(0))-1
1070 IF x%0 THEN x%=20
1080 IF x%21 THEN x%=1
1090 IF (x%ghost%(0,0) AND y%ghos
t%(0,1)) OR (x%ghost%(1,0) AND y%
ghost%(1,1)) OR (x%ghost%(2,0) AND
y%ghost%(2,1)) THEN ok0=0
1100 IF MID$(maze$(y0),x0,1)="" TH
EN RETURN
1110 PEN 14:LOCATE manx%,many%:PRIN
T "":LOCATE x%,y%:PRINT CHR$(224):m
anx%0:many%0%:IF MID$(maze$(y0),
x%,1)="" THEN RETURN
1120 SOUND 129,30,5,15:dots:lodots%
=1:score%=(score%+10):PEN 15:LOCATE 7,
25:PRINT score%:imaze$(y0)=LEFT$(ma
ze$(y0),x0-1)+" "+MID$(maze$(y0),x0
+1)
1130 RETURN
1140 :
1150 REM ** Mover fantasmas **
1160 FOR i%=0 TO 2
1170 IF gtime%(i%)(t% THEN GOSUB 12
20
1180 NEXT
1190 RETURN
1200 :
1210 REM ** fantasma **
1220 SOUND 130,370%+370,10,14:gti
me%(i%)(t%)+gdelay%(i%)

```

```

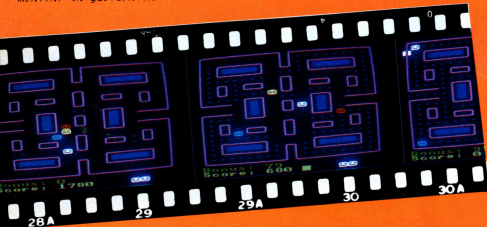
1230 y%ghost%(i%,1)+(ghost%(i%,1))
many%=(ghost%(i%,1)manx%):IF MID$(
maze$(y0),ghost%(i%,0),1)="" THEN
y%ghost%(i%,1)
1240 x%ghost%(i%,0)-(ghost%(i%,0)+
manx%)+(ghost%(i%,0)manx%):IF MID$(
maze$(y0),x0,1)="" THEN x%ghost%(
i%,0)
1250 IF MID$(maze$(ghost%(i%,1)),gh
ost%(i%,0),1)="" THEN temp%CHR$(1
40) ELSE temp% "
1260 PEN screen:LOCATE ghost%(i%,0),
ghost%(i%,1):PRINT temp%:PEN 11+x
:LOCATE x%,y%:PRINT CHR$(225):rghost
%(i%,0)=x%:rghost%(i%,f)=y%
1270 IF x%manx% AND y%many% THEN
ok0=0
1280 RETURN
1290 :
1300 REM ** Cogido! **
1310 FOR i%=255 TO 128 STEP -1
1320 PEN INT(RND*15):LOCATE manx%,m
anx%:PRINT CHR$(i%)
1330 CALL &BD19:SOUND 132,INT(RND*1
000)+1000,5,15
1340 NEXT
1350 iives=iives-1
1360 IF iives=0 THEN LOCATE 6,12:PE
N 14:PRINT CHR$(22):CHR$(1):%GAME
OVER%:CHR$(22):CHR$(0)
1370 RETURN
1380 :
1390 REM ** Instrucciones **
1400 ENV 1,10,1.100,10,-1,20
1410 BORDER 0:INK 0,0:INK 1,0:INK 2
,6:INK 3,24
1420 PAPER 0:CLS:PEN 1:LOCATE 1,25:
PRINT "Smiley vs. Grumpi!"
1430 SOUND 68,1100,4,1:SOUND 2,80
,1100,2,1:SOUND 4,95,1100,0,1

```

```

1440 FOR x%=0 TO 336 STEP 2
1450 FOR y%=0 TO 16 STEP 2
1460 IF TEST(x%,y%) THEN PLOT x%-12
0,366+y%*2,3:PLOT x%+120,366+y%*2,3
1470 NEXT
1480 NEXT
1490 LOCATE 1,25:PRINT SPACE$(25):
1500 INK 1,11:PEN 1:LOCATE 1,7
1510 PRINT "Dura a Smiley por el la
berinto cogiendo "PRINT:PRINT "las p
iezas coloreadas, pero atencion!"
1520 PRINT:PRINT "los grumpis inten
taran atraparte..."
1530 PEN 3:PRINT:PRINT:PRINT:
1540 PRINT "Pulsa "
1550 PEN 1:PRINT:PRINT " K para te
clado."
1560 PRINT:PRINT " J para joystick
."
1570 PEN 2:LOCATE 5,25:PRINT "( Pul
sa espacio para empezar )"
1580 a%=(9:lb%71:c%39:au%31
1590 WHILE INKEY(47)=1
1600 IF INKEY(37)=1 THEN LOCATE 1,
17:PRINT CHR$(243):PRINT:PRINT " "
PEN 3:LOCATE 1,22:PRINT "A. arriba
2..abajo 4..izq. 6..der. 8..PEN
2:ia%=(9:lb%71:c%39:au%31
1610 IF INKEY(45)=1 THEN LOCATE 1,
22:PRINT SPACE$(40):LOCATE 1,17:PR
INT " " :PRINT:PRINT CHR$(243):ia%=(2
:ib%73:c%74:d%75
1620 WEND
1630 PAPER 0
1640 RETURN
1650 :
1660 REM Siguiente pantalla
1670 PEN 14:LOCATE 2,12:PRINT CHR$(
22):CHR$(1):"Pantalla":screen%="comp
leta":CHR$(22):CHR$(10)
1680 screen=screen+1:IF screen=10 T
HEN screen=1
1690 score%=(score%+bonus):PEN 15:LOC
ATE 7,25:PRINT score%:
1700 RETURN
1710 :
1720 REM ** Fin del juego **
1730 FOR i%=0 TO 5000 NEXT
1740 FOR x%0 TO 320 STEP 4
1750 MOVE x%,0:DRAW x%,400,0
1760 MOVE 640-x%,0:DRAW 640-x%,400,
0
1770 NEXT
1780 IF score%>hi% THEN hi%=(score%
1790 PEN 14:LOCATE 1,3:PRINT STRING
$(26,"*")
1800 PEN 11:PRINT "Mejor puntuacion
"=hi%
1810 PRINT:PRINT:PEN 12:PRINT "Tus
puntos=":score%
1820 PRINT:PEN 14:PRINT STRING$(28
,"*")
1830 PEN 13:LOCATE 1,15:PRINT "Otro
juego...?":k%=-1
1840 WHILE k%=-1
1850 IF INKEY(43)=1 THEN k%=(43
1860 IF INKEY(46)=1 THEN k%=(46
1870 WEND
1880 RETURN

```



Para que los dedos, no realicen el trabajo duro. M.H. AAMS. TRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette manual, solicitándolo.

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.

Seréis, semana a semana, los encargados de construir esta página con vuestras consultas. En más de una ocasión, aquello que os preocupaba ya ha sido contestado antes a otro lector o, por el contrario, puede suceder que determinada consulta aclare muchos quebraderos de cabeza de otros aficionados.

Las cartas «**sin duda alguna**», nos servirán de gran ayuda. Gracias a ellas podremos ir evaluando vuestras necesidades y, de este modo, modificando el contenido de MICROHOBBY AMSTRAD acorde con ello. **¡Os esperamos!**

Me gustaría que los programas que se publican en la revista fuesen para el 464 y para el 664, sin utilizar los nuevos comandos disponibles para éste, o al menos indicando los cambios que hay que realizar para que funcione en ambos.

Manuel García (Barcelona)

Comprendemos tu inquietud. Suponemos que tienes un 464 y te preocupa no poder disfrutar de los programas de la revista.

Bien, la mayoría de los programas que se publiquen serán compatibles para ambos ordenadores y, cuando no lo sean, trataremos de indicar las modificaciones necesarias.

Sin embargo, hay programas en los que esto no es posible, bien porque manejan código máquina en direcciones de memoria determinadas, bien porque si se hace uso intensivo de los nuevos comandos del 664. Simplemente hay efectos que en el 464 no se pueden conseguir, o habría que volver a escribir el programa en su totalidad.

Quando se tiene grabado un programa en cinta y, por ejemplo, contiene 8 bloques, se me plantea el problema de que si por el motivo que fuere, el bloque 1 no lo carga, el resto de los bloques, no se pueden listar.

¿Podrían explicar cómo ha de activarse cuando se quiere listar parte de los bloques que contiene un programa?

Damián Gallego (Vitoria)

El Amstrad salva en cinta por bloques de una longitud de 2 Kbytes cada uno, pero no salva por bloques de línea de programa necesariamente.

Es decir, un programa Basic, para el ordenador no es más que una serie de bytes que van de tal a tal dirección y, como tales, se vuelcan a la cinta. Si el primero u otro cualquiera de ellos no carga, el resto de los bytes que vienen no tienen ningún sentido para la máquina y, por tanto, no se da por enterada cuando le pedimos el listado.

F

elicidades en primer lugar, por la idea de hacer una publicación semanal para usuarios de ordenadores Amstrad. Me dirijo a vosotros para ofrecer una colaboración, una línea de programa que creo será útil para aquellos usuarios que hagan programas con texto. Y es que veréis lo que me pasó el otro día cuando recibí una carta por impresora de un amigo, decía: «... Manana por la mañana será mi cumpleaños y podré añadir un año más a mi lista de años, y siento añoranza.» Por otra parte, creo que él también se sorprendió cuando le pregunté si le gustó el «**barreno**» que le regalé. Nunca me había fijado en lo importante que son las eñes. Por ejemplo, no es lo mismo que mi hermana se quite el moño, que mi hermana se quite el mono. Pero sobre todo, lo que me decidió a hacer la línea de programa fue la imposibilidad de mandar a la gente a hacer puñetas, yo no me resigno a mandar a hacer punetas. Creo que ya os habréis dado cuenta de que mi línea de programa es para incluir la letra eñe en el teclado, en la tecla que está situada a la derecha de la interrogación, \ ; También saldrá en mayúsculas dando a esa tecla (\) y SHIFT.

He aquí la línea de programa:

```
1 SYMBOL AFTER 237:SYMBOL
237,60,0,216,102,102,102,102,0:
SYMBOL 238,216,230,246,222,206,
198,198,0:KEY DEF 17,1,174:KEY
DEF 22,0,237,238
```

Como véis es muy fácil y nada tedioso de programar, no vale la pena ir por el mundo escribiendo burocracias, con lo poco que cuesta poner las eñes en el teclado. Sin más se despidе,

Ramón Luis Montoro de Francisco (Madrid)

LIBROS EN CASTELLANO PARA TU AMSTRAD



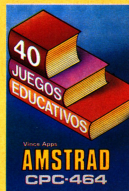
Manual de Referencia Basic para el Programador
La más autorizada y completa guía para programar en Locomotive Basic.
3.400.— Pts.



Juegos Sensacionales para AMSTRAD
Listados completos de 27 estupendos juegos de muy diversos estilos.
1.950.— Pts.



Programando con AMSTRAD
Fundamental para el usuario principiante.
Ameny y repleto de ejemplos.
2.400.— Pts.



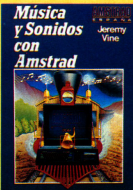
40 Juegos Educativos
Listados completos (matemáticas, geografía, música, etc.) para aprender divertidamente.
1.950.— Pts.



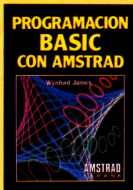
Código máquina para principiantes con AMSTRAD
Ideal para iniciarse en el código máquina del Z80 y en el sistema operativo del AMSTRAD.
2.100.— Pts.



Hacia la Inteligencia Artificial con AMSTRAD
Convierta su AMSTRAD en un compañero inteligente.
1.500.— Pts.



Música y Sonidos con AMSTRAD
Programa música y efectos sonoros y conviértalo en un sintetizador.
1.200.— Pts.



Programación de Basic con AMSTRAD.
Imprescindible para el principiante y eficaz herramienta para el programador avanzado.
2.100.— Pts.



Técnicas de Programación de Gráficos en el AMSTRAD
Este libro enseña a aprovechar las excelentes funciones gráficas del AMSTRAD, con múltiples ejemplos.
1.950.— Pts.



Curso Autodidáctico de Basic I y II
Un completo y estructurado Curso de Basic apoyado con numerosos ejemplos y acompañado de cassette.
2.900.— Pts. cada volumen



indescorp
PUBLICACIONES

Avd. del Mediterráneo, 9
Telfs.: 433 45 48 — 433 48 76
28007 MADRID

Delegación en Cataluña:
C/. Tarragona, 110 — Telf. 325 10 58
08015 BARCELONA

DE VENTA EN EL CORTE INGLES
Y TIENDAS ESPECIALIZADAS

Marca Registrada por el Grupo Indescorp.

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!

Desearía contactar con usuarios del **Amstrad** (en especial del CPC 664). Manuel Jesús Rodríguez Sedano. García Espín, 1.

Cambio programas Amstrad, sólo comerciales, más de 100 títulos, entre ellos últimos éxitos. The Way of Exploding Fist, Gremlins, copiones, utilidades, etc. E. Valderas Fdez. Apdo. 41. Puertollano (Ciudad Real).

Vendo:
Amstrad CPC-464. Color. Como nuevo.
100 juegos y utilidades (comerciales).
Libros y manuales.
Comprado en Corte Inglés.
Perfecto estado. Todo por 85.000 ptas. Rafa Urmeneta. Ctra. Zaragoza, s/n. Arguedas (Navarra). Tel. (948) 83 00 42.

Vendo ordenador Amstrad CPC-464 color e impresora original **Amstrad DMP-1**, todo como nuevo, la impresora está en garantía. Precio nuevo actual 160.000 ptas., lo vendo por 140.000 ptas. Tel. (96) 332 32 83). Valencia.

Estoy **interesado** en adquirir monitor fósforo verde para **Amstrad CPC664**. Precio a convenir. Tel. 351 64 70.

Vendo Amstrad CPC-664 con unidad de disco y monitor en color y, otro igual, en fósforo verde, ambos con tan sólo un mes de uso, están totalmente nuevos. Tienen garantía **Amstrad** España, manual en español, otro manual en inglés, los dos discos originales de regalo (c/pm, Logo y base de datos, procesador de textos, Random Files, Diseñador de gráficos, puzzle, animal, vegetal y mineral). Además regalo el Decathlon en cinta original y un disco virgen marca Amsoft. Con monitor en color, 120.000 ptas. Con monitor en fósforo verde, 95.000 ptas. Razón: Joaquín Domínguez Torrecilla. Sto. Entierro. Urbanización «Las Vigas». Bloque 1 - 2.º E. 11000 San Fernando (Cádiz). Tel. (956) 89 30 02.

Por cambio de ordenador **vendo** los siguientes programas a 600 ptas. cada uno: Rocky, Pole position, Decathlon, Gremlins, Knight Lore y algunos otros. Todos son originales. Mario García. Villa Begoña, 6 - 4.º Izq. Haro (La Rioja). (941) 31 14 45.

Vendo Amstrad 464 fósforo verde por 35.000 ptas. Está como nuevo (2 meses). También vendería 10 programas de juegos a 500 ptas. cada uno. Entre ellos tengo Rocky, Knight Lore y Combate Libre. Maite Uñarte Murua. Apdo. 1.582. Bilbao.

GANAR 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 incluyen:
 - Descripción general del programa.
 - Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
 - Instrucciones de manejo.

¡ENVÍANOS TU PROGRAMA!

Indicando claramente en el sobre:

AMSTRAD SEMANAL

a **HOBBY PRESS, S. A. La Granja, n.º 8. Pol. Ind. Alcobendas (Madrid)**

AMSTRAD[™]

“Lo increíble”

Confirmado por la prensa especializada

tu Micro



Micro, cassette y monitor en plena armonía. Su Basic es el más rápido de su categoría, superando al del Commodore, al del BBC e incluso al del Sinclair.

Computer Schau



Usuarios y técnicos lo confirman: se ofrece una relación precio/prestaciones que parece imposible.

Computer persönlich



Por un precio sorprendente se ofrece algo increíble. Un Basic superlativo.

micro



No hay en el mercado ningún ordenador en este nivel de precio que pueda enfrentarse a él.

C'T



¡Solución total a un precio fenomenal!

POPULAR Computing WEEKLY



Un ordenador personal extraordinario con unas enormes posibilidades como ordenador de gestión.

Personal Computer World



Su Basic es rápido, más rápido que casi todos los basics de 8 bits y que algunos de 16 bits.

micro bit



Su Basic se puede considerar impresionante... tiene unas características no usuales en microordenadores de su categoría.

COMPUTER CHOICE



El mejor Basic que he visto.

SCIENCE VIE MICRO

Se asedia a los distribuidores para conseguir un AMSTRAD. La demanda es desbordante.

micros



Calificado de “increíble”, las pruebas realizadas así lo han confirmado en casi todos los aspectos... es un equipo con posibilidades fuera de lo común...



AMSTRAD[™] “Lo increíble”

AMSTRAD CPC-6128



- MICROPROCESADOR Z80A.
- 128 K DE MEMORIA RAM
- 48 K DE MEMORIA ROM QUE INCLUYEN EL LOCOMOTIVE BASIC Y EL SISTEMA OPERATIVO.
- 76 TECLAS, TECLADO NUMERICO Y DE CURSOR INDEPENDIENTE.
- TEXTO EN MONITOR DE 20, 40 U 80 COLUMNAS Y GRAFICOS CON DEFINICIONES DE HASTA 640 X 200 PUNTOS. 27 COLORES DISPONIBLES.
- HASTA 8 VENTANAS EN PANTALLA.
- GENERACION DE SONIDOS EN 3 VOCES Y 8 OCTAVAS
- UNIDAD DE DISCO DE 3" (169 K BYTES)
- SISTEMAS OPERATIVOS AMS-DOS Y CPM/PLUS
- CONECTORES PARA IMPRESORA, JOYSTICKS, CASSETTE, SEGUNDA UNIDAD DE DISCO, ETC.

SISTEMA COMPLETO CON MONITOR EN FOSFORO VERDE, MANUAL EN CASTELLANO, GARANTIA OFICIAL AMSTRAD ESPAÑA, DISCO CON SISTEMA OPERATIVO CP/M 2.2 Y LENGUAJE DR. LOGO, DISCO CON SISTEMA OPERATIVO CP/M PLUS (CP/M 3.0) Y UTILIDADES, DISCO CON SIETE PROGRAMAS DE OBSEQUIO

109.500 Pts.

SISTEMA COMPLETO IGUAL AL ANTERIOR PERO CON MONITOR EN COLOR.

134.500 Pts.

AMSTRAD®
ESPAÑA

Avd. de Mediterráneo, 9, 28007 MADRID.
Tels. 433 45 48 - 433 48 76

Delegación Cataluña: C/. Tarragona, 110,
08015 BARCELONA - Tel. 325 10 58