

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO II N.º 45

160 Ptas.

Canarias 165 pts.

!NUEVA ETAPA DEL
CURSO DE MAQUINAI!
UN ARTICULO
DE IMPRESION

**DESVELAMOS
EL SECRETO
MANEJAR EL
DISCO EN
LENGUAJE
MAQUINA**

**DESPOJA A
TU AMSTRAD
HASTA DEL
ULTIMO CHIP
JUGANDO
CON EL AL
PLATO**

**Más allá de los
gráficos de tortuga**



ERBE *dades*

¿BUSCAS EMOCIONES FUERTES?



Green Beret (bomba verde), es el programa de mayor acción, que hayas visto jamás en un ordenador.
Decir Green Beret es decir un hombre perfectamente entrenado para la lucha.
Prepárate...



“V”

La serie de ciencia-ficción televisiva invade tu ordenador: La Tierra es visitada por alienígenas de aspecto humano. Sin embargo, estos reptiles pronto revelarán su objetivo final... conquistar la Tierra y esclavizar a la raza humana.



La crítica ha dicho: "Realizar un juego más completo debe ser poco menos que imposible".
En el papel de detective Jack Marlow, prepárate a enfrentarte al rey de la Mafía en el más apasionante juego que hayas visto.

AMSTRAD

sumario

Año II • Número 45 • 8 al 15 de Julio de 1986
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea.
Ceuta y Melilla, 155 ptas.

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.ª Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

Fernando Chaumel

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sapuerta

Robert Chatwin

Francisco Portalo

Pedro Sudán

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalia Gómez

Alberto Suñer

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Chema Sacristán

Portada

M. Barco

Ilustradores

J. Igual, J. Pons, F. L. Frontán,
J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

Maria Andino

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

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

Secretaría de Dirección

Marisa Cagorra

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid

Teléfonos: Suscrip.: 734 65 00

Redacción: 734 70 12

Dto. Circulación

Paulino Blanco

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDEC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novacom, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solano, 16

Depósito Legal

M-28468-1985

Derechos exclusivos

de la revista

COMPUTING with

the AMSTRAD

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

América de Ediciones, S.R.L. Sud

América 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

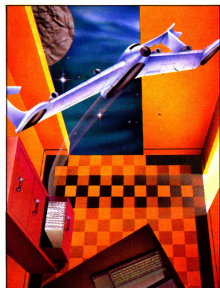
colaboradores en los artículos

firmados. Reservados todos los

derechos.

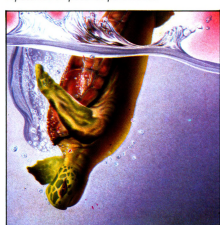
Program 6 Acción

Hasta ahora, manejar el disco desde código máquina estaba vetado a la gran mayoría de usuarios. Con este artículo, van a quedarle claras muchas cosas, pero sobre todo vas a descubrir cómo el LOCOMOTIVE de Amstrad puede aprovechar toda la potencia del CPM, de una forma tan sencilla como eficaz. Lee, lee y verás...



10 Análogo

Tras la lectura de este artículo, aprenderás a corregir y editar tus procedimientos en LOGO. Conoce el manejo de todas y cada una de las teclas de control. Y de paso, ¿por qué no?, repasemos un poco los parámetros.



Código 12 máquina

Convertir un programa hecho en BASIC a código máquina puede no ser tan difícil como pensamos. Comience desde ahora a utilizar llamadas al firmware y advierta el cambio de velocidad. ¡Viva la diferencia!

Para... PCW 16

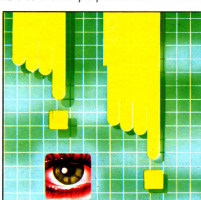
Si aún no ha hecho su declaración de la renta, MASTERSOFT se lo pone fácil. Haga ahora, con ayuda de su ordenador la declaración y descubra una de las múltiples razones que justifican la existencia de este tipo de aparatos.

22 MRS. Joystick

Después de haber sido capturado, juzgado y condenado injustamente, Monthy decide huir de la cárcel. Ayúdale en su presurosa huida.

Serie Oro 24

Jugar a las cartas siempre fue un peligro, al menos para el bolsillo. Si este juego, además, es nuestro viejo conocido, el plato y el oponente un CPC, la cosa es aún más grave. No desespere, y piensa que de una forma o de otra siempre puedes acabar reseteando.

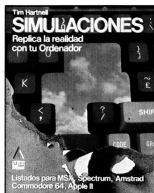


Primeros 30 pasos

Con este artículo damos por finalizado todo lo relativo a la lectura del teclado. Aprende ya en la recta final, las últimas utilidades de esta potente herramienta, que ofrece tu Amstrad a través de la utilización de KEY e INKEYS.

Ordena tus propias ideas

Le sacarás partido a tu ordenador



SIMULACIONES: REPLICA LA REALIDAD CON TU ORDENADOR

Tim Hartnell
1.643 ptas.



SISTEMAS EXPERTOS. INTRODUCCION AL DISEÑO Y APLICACIONES

Tim Hartnell
2.120 ptas.



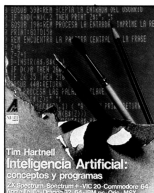
RUTINAS EN LENGUAJE MÁQUINA PARA AMSTRAD

Joe Pritchard
1.590 ptas.



PROGRAMACION AVANZADA DEL AMSTRAD

Don Thomasson
1.166 ptas.



INTELIGENCIA ARTIFICIAL. CONCEPTOS Y PROGRAMAS

Tim Hartnell
1.484 ptas.



EL LIBRO GIGANTE DE LOS JUEGOS PARA AMSTRAD

K. Bergin y A. Lacey
1.378 ptas.



PROGRAMACION DEL Z 80

Rodney Zaks
2.915 ptas.



CODIGOS Y CLAVES SECRETAS: CRIPTOGRAFIA EN BASIC

Listados para MSX, Spectrum, Amstrad, Commodore 64, Apple II
Gareth Greenwood
1.378 ptas.

☐ Les ruego me envíen el catálogo de su editorial.

☐ Les ruego me envíen los siguientes títulos:

TOTAL _____

☐ Adjunto talón bancario a
GRUPO DISTRIBUIDOR EDITORIAL, S. A.

☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).

☐ Giro postal.

Nombre _____

Profesión _____

Dirección _____

C. P. _____ Localidad _____

Provincia _____

MP2 _____

ANAYA
MULTIMEDIA

Adquiéralos en su librería habitual.

Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a:
Apdo. de Correos 14632, Ref. D. de C. 28080 MADRID

ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA

NUEVOS PROGRAMAS DE MASTERSOFT PARA LOS PCW Y 6128



MASTERSOFT ha sacado a la luz tres nuevos programas todos ellos para los PCW y 6128.

Estos son:

— **MASTERBLOCK.** Es una agenda en la que podremos trabajar con los teléfonos que nos interesen, creando su alta y su baja, hacer modificaciones, consultas por su nombre y teléfono, así como imprimir direcciones.

— **MASTERGEST.** Es un gestor de cuentas bancarias con el que se pueden crear cuentas, realizar movimientos, consultas de movimientos, borrado de movimientos, extractos de cuentas, elección de banco y grabación de datos.

— **MASTERQH.** Como se puede adivinar por su nombre, este programa va destinado a la confección de quinielas hípicas y podremos almacenar en él los datos referentes a cada caballo, tales como puesto en las carreras celebradas, fecha de la misma y, por supuesto, elaboración de un pronóstico.



JOYSTICK MOUSE

Ya disponemos en nuestro país del JOYSTICK MOUSE de la casa alemana **RUS WARE**. Este producto es distribuido en España por **ENFA IBERICA, S. A. S. A.**

Este ratón puede ser utilizado en

un AMSTRAD 464 (SCHNEIDER 464 en el mercado alemán).

Y el paquete donde va este compuesto por el programa de funcionamiento, el interface para el **ratón** y el **ratón** en sí.

Este, que está bien terminado, cuenta con la posibilidad de extraer su bolita de desplazamiento, para poderla limpiar o reemplazarla por otra nueva si fuera necesario.

Primera PLANA

TODO UN RETO EN TU PCW JUEGO DEL OTHELLO

por Zelig Software

El juego del Othello es similar al juego de las Damas. Consiste en conseguir el mayor número posible de fichas del mismo color. Las reglas del juego son sencillas y van explicadas en el manual que le acompaña.

El programa le permite jugar contra la máquina o contra otro jugador. Si se juega contra el ordenador puede escoger entre tres niveles de dificultad.

Equipo:

Amstrad PCW8256.
Amstrad PCW

Características:

Uno o dos jugadores.
Tres niveles de juego.

Nivel 0, Amateur.
Nivel 1, Medio.
Nivel 2, Experto.

Zelig Software:
Passeig de les Germanies 15
Tel. (96) 287 39 21
Gandia. Valencia.



EL DISCO EN MAQUINA

Quizá el sueño dorado de muchos programadores sea el poder tener un control total de su ordenador. Pues bien, hoy vamos a realizar uno de los sueños de cualquier usuario, conocer a fondo la unidad de disco; pero no desde Basic, no, ¡qué va!, nos vamos a meter en código máquina.



Quizá muchos de vosotros os preguntéis sobre la posible utilidad de esto, pues bien, os la vamos a decir, esto os puede dar pie a crearos vuestros propios programas para formatar, sin tener que recurrir al FORMAT o al DISKIT2/3, ver los ficheros de texto desde Basic, incluso, para los más osados hacer un IFILE-COPY o un IPIP.

Bien, pero antes que nada, debemos saber unas cuantas cosas:

Ya sabemos que el **Amstrad** puede direccionar más ROM de la que tiene, de hecho, la unidad de disco lleva su propia ROM, llamada **ROM del BIOS**. Si echamos un vistazo a la tabla 1, podremos ver el mapa de memoria de **CP/M 2.2**, que es similar al de nuestro **Amstrad** con una unidad conectada.

Ahora bien, dentro de de las llamadas a la ROM de disco, existen dos grupos diferentes, más uno adicional del sistema operativo.

Dentro de los grupos de la **ROM de BIOS**, están:

- La tabla de saltos extendida, que comienza a partir de la **#BE80** y contiene llamadas adicionales para escritura y lectura física del disco.

- La tabla standard de CP/M, que es aprovechada por **AMSDOS** cuando éste toma el control del disco. Al igual que la ROM alta de Basic, esta tabla es copiada en RAM al hacer un arranque de CP/M, y se sitúa detrás de **BDOS**, que es cargado del disco.

Si consideramos el grupo del sistema operativo, nos encontramos con rutinas de ayuda para la apertura y cierre de ficheros, escritura y lectura de los mismos, etc...

¡Pero ojo! A pesar de esto, es muy peligroso hacer llamadas al sistema operativo de disco sin saber exactamente qué hace la rutina, ya que operamos directamente con el controlador, y no con la CPU, por eso, las llamadas más usuales es aconsejable hacerlas al grupo de saltos del sistema operativo y no a la ROM de BIOS, a no ser que no le importe hacer caer polvo la unidad de disco.

En la tabla 2 están descritas las 9 rutinas básicas (hay más de 9) que componen el **BIOS** extendido. Ninguna de ellas es accesible desde el sistema operativo standard, de hecho no existen como tal. La única forma de acceder a ellas es mediante su código de control, y con una llamada a **KL FIND COMMAND**, dando la dirección donde está almacenado el código; con lo que se nos devuelve la dirección de selección de ROM (reg. C) y la dirección equivalente en HL.

Nosotros, para ir haciendo boca, crearemos algunos comandos **RSX** que nos sirvan para ir manejando poco a poco el disco desde máquina. Nuestro primer comando es un **ITYPE**.

En la tabla 3 están incluidas las rutinas que vamos a usar para crear el comando. De hecho, estas rutinas están incluidas en el sistema operativo básico, pero acceden a la ROM del BIOS para poder hacer operaciones más **«de bajo nivel»**, como posicionar la cabeza del disco, examinar la pista y el sector, el directorio, etc...

Antes que nada, vamos a examinar un poco el funcionamiento del **AMSDOS**:

Lo primero que podemos preguntarnos es acerca de las cabezeras. ¿Llevarán los programas cabezera o no? Pues son programas ASCII y se cargan a partir de la **#0100**; sin embargo, los ficheros Basic y BINarios sí llevan.



Byte	Función
0	Número USER #00... #0F
1...8	Nombre, rellenado con espacios si tiene menos de 8 car.
9...11	Tipo, también rellenado con espacios.
12...15	No usados, van en 00.
16...17	No se usan en disco, van en 0.
18	Contiene el tipo de fichero.
19...20	Longitud de datos.
23	Se pone a #FF si es el primer bloque.
24...25	Dirección de datos.
26...27	Dirección de entrada.



Este es el formato aproximado de una cabecera en disco, es equivalente al tipo definido en cinta. Contiene todos los datos sobre el fichero en particular, y su localización dentro del disco es siempre acompañando al fichero.

AMSDOS tiene acceso a las llamadas del cassette, pero adaptadas para el disco, esto le da al ordenador una facilidad absoluta para cambiar de cinta a disco a voluntad.

Aunque **AMSDOS** hace uso de estas rutinas, también tiene acceso a la ROM de disco, que facilita no sólo los

accesos a la propia unidad, sino que también incorpora en memoria un juego de instrucciones **RSX** para poder manejar los ficheros de disco a voluntad.

El mayor desconocido de estos comandos quizá sea el **IBASIC**, que inicializa el ordenador y vuelve a **AMSDOS**.

También **AMSDOS** dispone de su propio juego de errores añadidos al sistema, cada uno de ellos identificados con un código interno, estos códigos son retornados cuando una llamada no es satisfactoria, y son los siguientes:

#OE ... Fichero no abierto (File not open)
 #OF ... Final físico de fichero (EOF met)
 #1A ... Final de datos en fichero (EOF met)
 #20 ... Formato de datos erróneo (Bad command)
 #21 ... Fichero aún existe (File already exist)
 #22 ... Fichero no existente (not found)
 #23 ... Directorio lleno (Directory full)
 #24 ... Disco lleno (Disc full)
 #25 ... Disco equivocado (File not close correctly)
 #26 ... Fichero sólo lectura (File is read-only)

Los errores mayores de #40 son los provenientes del controlador de disco, y suelen ser errores de **HARD** normalmente.

El formato es el siguiente:

bits 7 y 6 siempre a 1 para indicar error «reservado».
 bit 5 ... data error.
 bit 4 overrun error.
 bit 3 drive not ready.
 bit 2 no data (en sector).
 bit 1 not writable (disco protegido contra escritura).
 bit 0 adress mask missing (falta marca de dirección).

Después de conocer un poco mejor **AMSDOS**, pasamos a describir qué hace nuestro nuevo comando **RSX**.

El programa se divide en tres partes: el comando en sí, el programa principal y la gestión de errores.

En las líneas 10 a la 100 creamos el comando.

El programa principal comprende de la línea 110 al final.

Las rutinas de error intercaladas están en las siguientes líneas:

PROGRAMACCION

— En la línea 110, donde se mira el número de parámetros existentes.

— En la línea 260, si el fichero no ha sido abierto correctamente, volvemos a Basic.

— En las líneas 290 a 300, donde comprobamos el final del fichero.

En las líneas 370 y 380, si pulsamos <ESC> el programa retorna a **BASIC**.

Por lo demás, la única complicación aparente del programa la aporta el manejo de indexados para cargar los parámetros del **RSX**.

Mapa de CP/M 2.2

— **TPA**: Transient Programa Area. Área de programas transitorios. Aquí se cargan los programas en CP/M.

— **BIOS**: Basic Input / Output System. Sistema básico de entradas/salidas. Es el sistema encargado de los ports de entrada y salida (unidad de disco, cassette, teclado, etc...). Reside en la ROM BIOS, identificada por el número 7.

— **CCP**: Console of Command Processor. Consola o monitor de proceso de comandos. Aquí se comprueban las entradas de teclado, la validez de los Comandos, etc...

— **BDOS**: Basic Disc Operative System. Sistema operativo básico de disco. Es donde residen las rutinas de control del disco. Estas rutinas son las principales, y su función determinada se ejecuta accediendo a las rutinas del BIOS, esto es, las rutinas del BIOS son el núcleo del sistema, mientras que el BDOS contiene «el programa principal».

Rutinas básicas del sistema

—CAS IN OPEN— # BC77

Abre un fichero de entrada de datos para el disco, equivale al «OPEN» de Basic. En código máquina necesita los siguientes parámetros:

B contiene la long. del nombre del fichero.

HL contiene la dirección donde está almacenado el nombre del fichero.

DE contiene la dirección de un área de memoria de 2k libres, que actúa como buffer del disco.

MAPA DE MEMORIA

Dirección	Grupo	Notas
#FFFF	ROM del BIOS	Se solapa con la memoria de pantalla y la ROM alta.
#C000	STACK del BIOS	Aquí reside la pila del BIOS, que es donde se guardan los datos y parámetros del sistema operativo para su utilización posterior.
#BEC0	SALTOS EXTENDIDOS del BIOS	Tabla de llamados extendidos del sistema operativo.
#BE80	VARIABLES FIRMWARE del BIOS	Los RSX van aquí, bajando CCP y BDOS.
#AD33	SALTOS STANDARD del BIOS	(Basic Disc Operative System). El sistema operativo básico de disco.
#9F06	CCP	Console Command Processor.
#9700	TPA	Transient Program Area. Área de programas transitorios. Aquí van los programas escritos en CP/M.
#0100	Página 0	
#0000		

LLAMADAS PRINCIPALES DEL BIOS

Llamada	Código	Función
SET MESSAGE	CTRL-A # 01	Habilita/inhabilita los mensajes de error.
SETUP DISC	CTRL-B # 02	Inicializa parámetros del disco.
SELECT FORMAT	CTRL-C # 03	Selecciona un formato standard.
READ SECTOR	CTRL-D # 04	Lee un sector del disco.
WRITE SECTOR	CTRL-E # 05	Escribe un sector en el disco.
FORMAT TRACK	CTRL-F # 06	Formatea una pista completa.
MOVE TRACK	CTRL-G # 07	Mueve la cabeza hacia una pista concreta.
GET DR STATUS	CTRL-H # 08	Comprueba el estado del disco.
SET RETRY COUNT	CTRL-I # 09	Selecciona el número de veces que ha de repetirse los mensajes de error.

Si el fichero se abre correctamente hay acarreo.

EJEMPLO:

Supongamos que queremos abrir un fichero llamado AMSTRAD.TXT.

Tomamos como buffer la # 4000, el nombre lo guardamos en la # 3000.

```
LD HL,          ; dirección donde está el nombre.
# 3000
LD DE,          ; dirección del buffer.
# 4000
LD B, 11        ; longitud del nombre.
CALL #BC77     ; abrimos el fichero.
JP C, INTRO    ; si hay aCarreo saltamos a la entrada de datos.
                ; (Jump if Carry).
NOP            ; aquí estaría la rutina del mensaje de error.
INTRO: NOP     ; aquí está la rutina de entrada de datos.
```



—CAS IN CLOSE— # BC7A

Cierra el fichero de entrada de datos, equivale al CLOSEOUT de Basic.

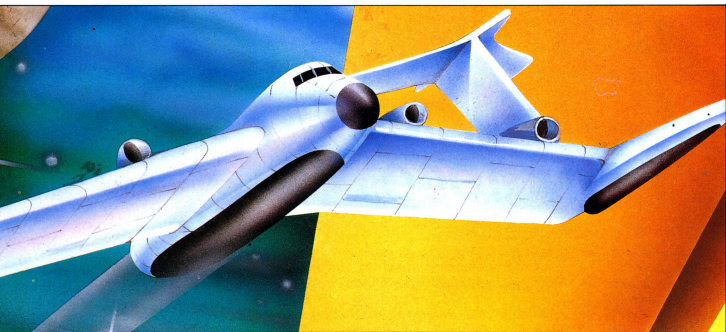
—CAS IN CHAR— # BC80

Lee un carácter del fichero abierto y lo deposita en el registro A.

Si se ejecuta la acción, A contiene el carácter (el código ASCII del carácter) y hay aCarreo.

Si se encuentra el final del fichero, A queda alterado, y hay aCarreo.

```
1 SYMBOL AFTER %2;CLEAR
10 DATA 1, 14, 128, 13, 18, 128, 1
95, 209, 188, 84, 89, 80, 197, 0, 9
128, 24, 4, 0, 0, 0, 254, 1, 1
92, 221, 102, 1, 221, 110, 0, 70, 1
97
20 DATA 35, 78, 35, 70, 96, 105, 1
93, 17, 87, 128, 205, 119, 188, 48,
30, 42, 12, 205, 90, 187, 205, 137
188, 48, 20, 205, 128, 188, 205,
90, 187, 175
30 DATA 205, 77, 187, 254, 32, 40,
10, 254, 27, 40, 2, 24, 251, 205,
122, 188, 201, 205, 24, 187, 24, 22
2, 0
40 FOR n=8000 TO 8057
50 READ a
60 POKE(n),a
70 NEXT
80 CALL MB000
```

```

10      ORG #B000      Inicio del programa
20      ENT #B000      Entrada del programa
30      LD BC,COMM      Posicion de tabla de comandos
40      LD HL,BYTES      Posicion de los bytes libres
50      JP #BCD1         Definimos el comando
60 NAME: DEFB "TYP"     Nombre del comando menos "E"
70      DEFB #CS,#00     Ultima letra «#B0 y final tabla
80 COMM: DEFW NAME      Definimos el nombre
90      JR ENTRY         Ejecutamos el programa
100 BYTES: DEFB #00,#00 Bytes de uso del comando
101      DEFB #00,#00     Testeamos el numero de parametros
110 ENTRY: CP 1          Si error, cerramos y RETorno BASIC
120      PET N?           Volvemos si hay mas de 1
130      LD H,(IX+1)
140      LD L,(IX+0)
150      LD B,(HL)
160      PUSH BC          Guardamos la longitud del nombre
170      INC HL
180      LD C,(HL)
190      INC HL
200      LD B,(HL)
210      LD H,B
220      LD L,C           Guardamos la posicion del nombre
230      POP BC
240      LD DE,BUFF       Definimos el buffer de entrada
250      CALL #BC77        Abrimos el fichero
260      JR NC,CLOSE      Si error, cerramos y RETorno BASIC
270      LD A,12
280      CALL #BB5A        Borramos la pantalla
290 READ: CALL #BCB9      Si EOF fichero, cerramos y BASIC
300      JR NC,CLOSE      Seguimos entrando datos
310      CALL #BCB0        Pasamos al buffer caracter entrada
320      CALL #BB5A        Visualizamos el caracter
330      XOR A            Borramos reg. A
340      CALL #BB1B        Buscamos tecla pulsada
350      CP 32
360      JR Z,WAIT         Si es «ESPACIO», esperamos
370      CP 27
380      JR Z,CLOSE       Si es «ESC», volvemos a BASIC
390      JR READ          Seguimos entrando datos
400 CLOSE: CALL #BC7A     Cerramos fichero de entrada
410      RET              Vuelta a BASIC
420 WAIT: CALL #BB1B      Esperamos pulsacion de tecla
430      JR READ          Retornamos a la entrada de datos
440 BUFF: NOP            Inicio Buffer de entrada

```

—CAS TEST EOF—

BC89

Esta rutina sirve para compensar la «falta de eficacia» de la anterior al detectar el final del fichero, equivale al EOF de Basic.

Si se ha llegado al final del fichero, hay aCarreo.

Estas son las rutinas básicas que vamos a utilizar para hacer un ITYPE, nombre.

Cómo utilizar el programa

Con ensamblador, cargar y ensamblar, grabar el objeto, y para cargarlo desde Basic:

```

10 SYMBOL AFTER 32: CLEAR
20 MEMORY &7FFF
30 LOAD "TYPE.BIN", &8000
40 CALL &8000

```

En Basic, simplemente es hacer funcionar el programa objeto incluido.

FUNCIONES:

Teclar ITYPE, "nombre.ext" para 664 y 6128

Para 464, a\$="nombre.ext" ITYPE, %a\$.

Pulsando la barra espaciadora se define el listado, para seguir pulsar cualquier tecla.

Si pulsamos ESC, el programa retorna a Basic, pero esto sólo funciona si ejecutamos un:

KEY DEF 66,0,27 antes de cargar el programa.

LAS TECLAS DE EDICION EN «LOGO»

Quando estamos creando un programa en cualquiera de los lenguajes de programación que nuestro Amstrad nos permite, necesitamos hacer continua y repetidamente modificaciones de toda índole y alcance. Para ello hemos de conocer algo que en el argot informático llamamos «EDITOR». Cada ordenador, y cada lenguaje dentro de un mismo ordenador, presentan peculiaridades sin cuyo conocimiento resulta prácticamente imposible centrarnos en la creación de un programa.

E

l problema se va a presentar, una vez cargado el intérprete «**LOGO**». La pantalla aparece en modo texto, con el inductor «» situado en el extremo superior izquierdo.

Escribimos nuestras primeras instrucciones, puede que hasta nuestro primer procedimiento. Lo repasamos y vemos que hemos permutado dos letras, por ejemplo. Un simple fallo mecanográfico. Intentamos subsanarlo. Comprobamos que no resulta fácil. Supongamos que hemos escrito todo un procedimiento sin errores tipográficos. Lo ejecutamos y no sale lo que deseábamos. A modificar... Pero, ¿cómo?

Hemos leído el exiguo texto que el manual de instrucciones dedica a «**LOGO**». Pasamos a lo que llamamos «**Modo editor**». Empezamos a escribir. Al poco tiempo, hacemos uso de las teclas de cursor, de las teclas «**ESC**», «**DEL**», «**CLR**», etc. No obedecen a nuestros deseos. No realizan la función esperada. No es lo que dice el manual.

A los lectores que hayan pasado por estas experiencias, dedicamos las siguientes líneas. De otra manera no resulta demasiado grato experimentar con el «**LOGO**» de Amstrad. Y la verdad es que merece la pena, a pesar de todo.

A continuación presentamos una tabla en la que se muestran las teclas y combinaciones de teclas que necesitaremos usar para editar nuestros procedimientos y/o programas, indicando a continuación el efecto de cada una de ellas.

TECLA	EFEECTO
CONTROL-A	El cursor se sitúa al comienzo de la línea en la que se encuentra.
CONTROL-B	El cursor se sitúa un lugar hacia la izquierda respecto a su posición anterior.
CONTROL-C	Se sale del modo de edición para entrar en el de ejecución.
CONTROL-D	Borra el carácter o elimina el espacio sobre el que está situado el cursor. Este no se mueve. El cursor se sitúa al final de la línea en la que está.
CONTROL-E	El cursor se sitúa un lugar a la derecha respecto a su posición anterior.
CONTROL-G	Se sale del modo de edición para entrar en el de ejecución.
CONTROL-K	Borra todos los caracteres situados a la derecha del cursor. Incluso el que quedaba debajo de él.
CONTROL-N	Mueve el cursor una línea hacia abajo.
CONTROL-P	Mueve el cursor una línea hacia arriba.



Existen, para esta función de edición, tres teclas más.

Pero a estas alturas del texto, vamos a pasar a una recomendación. Consideramos interesante se recorte la tabla anterior, o bien se fotocopie; así cuando trabajemos con «**LO-**



de estas funciones, lleva consigo otras, que podrían considerarse como una consecuencia o complemento de las indicadas. Mejor que explicarlas es pasar a la acción según veremos más adelante. Resulta sencillo y cómodo.

Leyendo con detenimiento la repetida tabla, es preciso establecer una diferencia muy importante entre las combinaciones de teclas «CONTROL-C» y «CONTROL-G».

Con la primera combinación, salimos del modo de edición entrando en el de ejecución. Los procedimientos o programas se ejecutarán tal como estaban. Si nuestro proyecto no era correcto, la ejecución resultará tal cual, y a la inversa.

Si nos proponemos entrar de nuevo en el modo de edición para hacer una o varias correcciones, y una vez efectuadas éstas, salimos de edición con la combinación «CONTROL-G», comprobaremos al ejecutar de nuevo el programa, que nuestras modificaciones han sido ignoradas. Esta es la importante diferencia.

Saliendo de edición con «CONTROL-C», las modificaciones serán observadas escrupulosamente. La ejecución responderá a nuestros cambios.

Aun a riesgo de desviarnos un tanto del tema creemos conveniente poner de manifiesto que la combinación «CONTROL-G» pulsada en modo **EJECUTION**, interrumpe el programa que se estaba ejecutando. El interés de esta observación, podría quedar justificado si alguna vez nos metemos sin querer en un bucle sin salida, o en una recursión que no responde a nuestros deseos.

Y ya seguimos con nuestras teclas.

La «**BARRA ESPACIADORA**». Inserta espacios en blanco a medida que el cursor se desplaza a la derecha. Los caracteres situados a su derecha, se desplazan igualmente.

La tecla «**DEL**» borra los caracteres situados a la izquierda del cursor. Los situados a su derecha se desplazan hacia la izquierda, acompañando a éste.

La tecla «**RETURN**» tiene un comportamiento que depende de la posición dentro de la línea en la cual se encuentra el cursor. Con éste al final de línea, sobre espacio en blanco, al pulsar la mencionada tecla, el cursor se sitúa al comienzo de la siguiente. Para no alargarnos, proponemos que el lector experimente con esta tecla, habiendo situado el cursor previamente en distintas posiciones dentro de una línea.



GO» podremos tenerla a la vista y efectuar consultas. Con el tiempo, y sin quererlo, es posible que lleguemos a memorizarla.

Las diferentes combinaciones de teclas incluidas en la tabla, realizan las funciones ya expresadas. Es preciso hacer notar que cada una

ANALOGO

Después de la primera lectura hasta este punto, sugerimos la puesta en marcha del **Amstrad** y la carga de «**LOGO**».

Enlazando con nuestro último artículo de «**ANALOGO**» (n.º 40 de nuestra revista), presentamos un procedimiento con ingredientes similares, que nos permitirá la aplicación de recursos ya conocidos, al trazado de «**GRAFICOS DE LA TORTUGA**».

Podemos valernos de nuestros recientes conocimientos adquiridos sobre el editor, para escribir nuestro procedimiento. Incluso para practicar. Hasta podríamos «**equivocarnos**» un poco.

Carguemos «**LOGO**». A la derecha del inductor «**E**» escribamos *ed*. Así nos situamos en modo de edición. Escribamos el siguiente procedimiento:

```
to espiral :lado :ángulo :incl :inca
if :lado > 150 [stop]
fd :lado rt :ángulo
make "lado :lado + :incl
espiral :lado :ángulo :incl :inca
end
```

Corrijamos, y a continuación pulsemos la combinación de teclas «CONTROL-C». A la derecha del inductor, escribamos:

```
cs ht espiral 6 90 2 0
```

Recordemos: *cs* clear screen (borra pantalla).

ht hyde turtle (oculta tortuga).

Los siguientes cuatro números corresponden: El primero, a la longitud del primer trazo de la espiral.

El segundo, al ángulo del primer quiebro de la espiral.

El tercero, al incremento del trazo.

El cuarto, al incremento del ángulo.

Si queremos seguir «jugando», una vez terminada la ejecución del procedimiento anterior, tecleemos a la derecha del inductor, otra vez:

```
cs espiral xx xx xx xx
```

sustituyendo las *xx* por cuatro números a nuestra elección. Con un poco de imaginación, nos pueden salir cosas bonitas y curiosas. Intentemos comprender el por qué de los dibujos que obtenemos.

IMPRESION EN PANTALLA

Nuestro curso de código máquina entra hoy en una nueva y decisiva etapa. Hemos revisado exhaustivamente todo el grupo de instrucciones máquina del Z80, desde la primera hasta la última, explicando su funcionamiento, su razón de ser y cómo extraer de ellas el máximo provecho en su aplicación a un Amstrad.

Por: **Alberto Suñer**



Hemos tratado también, y creo que lo hemos conseguido, enfocar una parte de la programación tradicionalmente considerada como árida y de «élite», de una forma amena, llena de ejemplos útiles, y sobre todo, interactiva, de forma que el lector participe activamente su ordenador desde la primera línea de cada artículo.

Con este mismo espíritu vamos a abordar la segunda y más interesante parte de nuestro curso, que se puede resumir en muy pocas palabras: vamos a enseñar a todo el mundo a programar en lenguaje máquina, siguiendo una metodología sistemática y profundamente estudiada.

¿Qué cómo vamos a hacerlo?

Bien, en mi opinión, de la forma más lógica: planteándonos una tarea, escribiéndola en Basic y transcribiéndola a lenguaje máquina línea a línea de programa.

Así, resultará tremendamente sencillo comprender cómo se programa en máquina, y adaptar nuestra mente a la forma de pensar requerida por este lenguaje, la cual no tiene nada que ver con la del Basic exceptuando que la lógica de un programa, en la mayoría de los casos, es la misma.

Antes de ceder la palabra a Alberto Suñer, nuestro experto en lenguaje máquina y director del curso, sólo queda dar las gracias al constante apoyo que esta sección ha recibido en todo momento de nuestros lectores a través, tanto de críticas constructivas, como de cartas de felicitación y aliento; esperamos que la segunda parte reciba una acogida igualmente positiva y benévola.

José María Díaz

una vez estudiadas una a una todas las instrucciones que puede utilizar el microprocesador Z80 que lleva incorporado nuestro Amstrad, empezaremos ahora con una serie de capítulos en los cuales intentaremos convertir diversos programas escritos en Basic, a lenguaje máquina.

Para ello, escribiremos primero el programa en el lenguaje que la mayoría de nosotros conocemos mejor, o sea el Basic, y a partir de ahí, intentaremos transcribirlo a código máquina, explicando paso a paso dicha conversión.

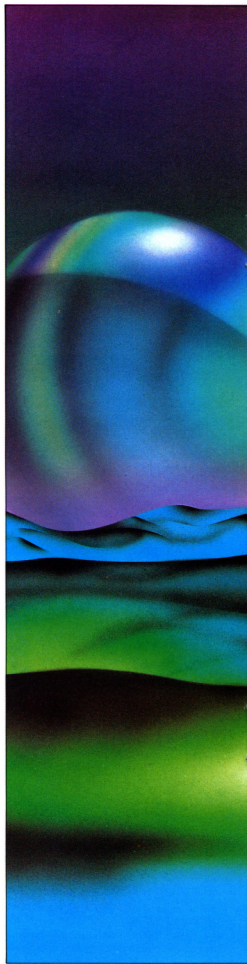
Asimismo, cuando sea necesario utilizar cualquier llamada al firmware, ésta se investigará con detenimiento, indicando los parámetros que son necesarios definir antes de la llamada a dicha rutina.

Como inicio a esta serie, he preparado un pequeño programa que creo fundamental para iniciarse en el estudio del lenguaje máquina. Se trata de la impresión de mensajes en pantalla.

Dicho capítulo no es que sea fundamental en cuanto al aprendizaje de código máquina se refiere, sin embargo, es vital en cuanto a la programación de cualquier juego o utilidad.

Si por ejemplo nos proponemos hacer un programa de agenda o cualquier base de datos, lo primero que deberemos hacer es imprimir un menú en el cual se indiquen las opciones de que dispone dicho programa.

Si por el contrario nos proponemos hacer un juego, éste deberá tener diversos marcadores para informarnos de la situación en que nos encontramos.



Pues bien, éste será nuestro primer tema de estudio en esta serie de capítulos.

En este primer capítulo, veremos cuál es la técnica utilizada para imprimir un mensaje en pantalla con diferentes plumas. Así pues, el programa en Basic que hemos preparado, nos ofrece el siguiente resultado:

CURSO DE CODIGO MAQUINA

MICROHOBBY AMSTRAD

Cada uno de estos dos mensajes está escrito con un tipo de pluma distinto.

Como vemos, lo primero que hacemos en el programa en Basic, es colocar la pantalla en modo 1, para limpiar la pantalla y elegir dicho modo.

Para efectuar dicha operación desde código máquina, recurriremos a una llamada al firmware que nos proporciona el modo de pantalla elegido. Dicha llamada es la siguiente:

SELECCIONA MODO DE PANTALLA. #BCOE

Selecciona el modo de pantalla y se asegura de que dicho modo de pantalla sea correcto.

Condiciones de entrada.

El registro A debe contener el modo de pantalla que se requiere. Es pues el acumulador en este caso deberá contener valores comprendidos entre 0-2.

Condiciones de salida.

Se corrompen los registros AF, BC, DE y HL. Se preservan los demás registros.

Así pues en nuestro programa, antes de hacer la llamada a dicha rutina, deberemos cargar en el acumulador el valor 1, ya que éste es el modo de pantalla que deseamos obtener para ejecutar el programa.

A continuación se elegirá el número de **pluma** con el que se quiere imprimir el primer mensaje en pantalla. Esto lo haremos también utilizando otra rutina en el firmware de nuestro **Amstrad**:

SELECCIONA PLUMA DE TEXTO. #BB90

Selecciona una pluma de texto en la ventana seleccionada. Esta es la que se utilizará para escribir textos.

Condiciones de entrada.

El registro A debe contener el número de pluma a utilizar.

Condiciones de salida.

Se corrompen los registros AF y HL, los demás registros son preservados.

Código MAQUINA

Seguidamente deberemos elegir la posición de pantalla en la cual deseamos imprimir el texto, teniendo en cuenta que es a partir de esa posición donde va a empezar a imprimirse el mensaje.

Para ello, utilizaremos otra de las rutinas que el **firmware** nos proporciona y que es la siguiente:

SELECCIONA LA POSICION DEL CURSOR DE TEXTO. #BB75

Coloca la posición actual del cursor en las nuevas posiciones de fila y columna de la ventana seleccionada.

Condiciones de entrada.

El registro H deberá contener el número de columna en el cual se desea imprimir y el registro L contendrá el número de la fila.

Condiciones de salida.

Se corrompen los registros AF y HL y los demás son preservados.

En este momento tenemos ya seleccionados todos los parámetros del mensaje a imprimir por lo tanto estamos ya en condiciones de pintar dicho mensaje en pantalla.

Para provocar dicha impresión, cargaremos el registro par HL con la dirección del texto que se desea colocar en pantalla. Así pues en nuestra rutina deberemos cargar dicho registro con la dirección **«TXT1»** que es el primer texto a imprimir, y a continuación llamaremos a la rutina de impresión que hemos preparado y que hemos llamado **«PRINT»**.

Vamos a estudiar el funcionamiento de esta rutina paso a paso. En primer lugar cargamos en el acumulador el valor que contiene la primera dirección a la que apunta el registro doble HL:

PRINT: LD A,(HL)

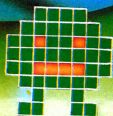
a continuación comparamos ese valor con 253 que en nuestro caso va a ser el valor que indicará el final de mensaje.

CP 255

si el valor contenido en el acumulador es igual a **255**, indicará que se ha finalizado la impresión del texto y se retornará al programa principal:

RET Z

En caso de que el valor contenido en A sea distinto a 255, la rutina de



impresión continuará su ejecución, provocando la impresión en pantalla del siguiente elemento de texto mediante la llamada a una rutina del **firmware**:

CALL #BB5A

una vez hecho esto, incrementamos el puntero HL y volvemos al principio de la rutina de impresión.

JR PRINT

IMPRESION DE UN CARACTER EN PANTALLA.

#BB5A

Imprime un carácter en la actual venta de pantalla y obedece los códigos de control.

Condiciones de entrada.

El registro A debe contener el código **ASCII** del carácter a imprimir.

Condiciones de salida.

Todos los registros y flags son preservados.

Una vez impreso el primer mensaje, deberemos seleccionar la pluma con la cual vamos a imprimir el segundo mensaje, esto lo haremos llamando a una rutina del firmware estudiada anteriormente:

LD A, 2

CALL #BB90

en este caso la pluma seleccionada será la número 2.

A continuación colocaremos el cursor en la posición de pantalla que se desea imprimir el segundo mensaje.

LD HL, #OAOF

CALL #BB75

Una vez seleccionados los parámetros para la impresión del segundo texto, cargamos en HL la dirección del segundo texto **«TXT2»** y llamamos a nuestra rutina de impresión.

Por último, sólo nos queda esperar a que se pulse una tecla, para ello utilizamos una rutina del firmware que provoca este efecto:

ESPERA A QUE SE PULSE UNA TECLA. #BB18

Intenta tomar el valor de una tecla desde el buffer de teclado. Esta rutina espera hasta que dicha **tecla** sea pulsada.

Condiciones de entrada.

No necesita ninguna condición.

Condiciones de salida.

El acumulador contiene el valor del carácter que se ha pulsado desde teclado. El Carry se pone a uno.

Se corrompen todos los demás flags. Se preservan todos los demás registros.

PROGRAMA 1

```
10 MODE 1
20 LOCATE 9,10
30 PEN 1
40 PRINT "CURSO DE CODIGO MAQUINA"
50 LOCATE 12,15
60 PEN 2
70 PRINT "MICROHOBBY AMSTRAD"
80 WHILE INKEY$="" :WEND
```

LISTADO DESENSAMBLADO

```
A000 10 ; ORG #A000
20 ;
30 ;MODE 1
40 ;
A000 3E01 50 LD A,1
A002 C0EBC 60 CALL #BC0E
70 ;
80 ;PEN 1
90 ;
A005 3E01 100 LD A,1
A007 C09BB 110 CALL #BB9B
120 ;
130 ;LOCATE 9,10
140 ;
A00A 218A0F 150 LD HL,#B9BA
A00D C075BB 160 CALL #BB75
170 ;
180 ;IMPRIMIMOS EL PRIMER MENSAJE
190 ;
A010 2135A0 200 LD HL,TXT1
A013 C02BA0 210 CALL PRINT
220 ;
230 ;PEN 2
240 ;
A016 3E02 250 LD A,2
A018 C09BB 260 CALL #BB9B
270 ;
280 ;LOCATE 10,15
290 ;
A01B 210FBA 300 LD HL,#BA0F
A01E C075BB 310 CALL #BB75
320 ;
330 ;IMPRIMIMOS EL SEGUNDO MENSAJE
340 ;
A021 214DA0 350 LD HL,TXT2
A024 C02BA0 360 CALL PRINT
370 ;
380 ;ESPERA A QUE SE PULSE UNA TECLA
390 ;
A027 C018BB 400 CALL #BB1B
A02A C9 410 RET
420 ;
430 ;ROUTINA GENERAL DE IMPRESION
440 ;
A02B 7E 450 PRINT: LD A,(HL)
A02C FEFF 460 CP 255
A02E CB 470 RET 2
A02F C05ABE 480 CALL #BB5A
A032 23 490 INC HL
A033 10F6 500 JR PRINT
510 ;
520 ;DATOS DE LOS MENSAJES A IMPRIMIR
530 ;
A035 43555253 540 TXT1: DEFM "CURSO DE CODIGO MAQUINA"
A04C FF 550 DEFB 255
A04D 40494352 560 TXT2: DEFM "MICROHOBBY AMSTRAD"
```

PROGRAMA CARGADOR

```
10 FOR N=&A000 TO &A060
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>262E THEN PRINT "ERROR
EN DATAS"
91 DATA 62,1,205,14,180,62,1
92 DATA 285,144,187,33,10,9,285
93 DATA 117,187,33,55,160,285,43
94 DATA 160,62,2,285,144,187,33
95 DATA 15,10,285,117,187,33,77
96 DATA 160,285,43,160,285,24,187
97 DATA 201,126,254,255,28,285,98
98 DATA 187,35,24,246,67,85,82
99 DATA 83,79,32,68,69,32,67
```

```
100 DATA 79,68,73,71,79,32,77
101 DATA 65,81,85,73,78,65,255
102 DATA 77,73,67,82,79,72,79
103 DATA 66,66,89,32,65,77,83
104 DATA 84,62,65,65,255,6,0
```



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



AMSTRAD PCW 8256

UN COMPLETO EQUIPO QUE INCLUYE:

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

PROGRAMAS PROFESIONALES

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

También disponible la versión PCW 8512, con 512 K RAM y 2º disco de 1 MBYTE incorporado.
P.V.P. 169.900,— Ptas. + IVA

SOLICITE DEMOSTRACION EN:

División informática de **El Corte Inglés**, División **Online** de GALERIAS.
Tiendas especializadas en informática y Equipos de oficina.

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

¡¡ Increíble !!

AMSTRAD

ESPAÑA

GRUPO INDESCOMP

¿HIZO YA SU DECLARACION?

Ya llega el verano. Y con el verano, los planes para ir de vacaciones, la playa, el mar, y un sinfín de planes optimistas para disfrutar el periodo de vacaciones que se aproxima. Y con esto, dos de los acontecimientos que todos los trabajadores esperan con más «impaciencia». LA DECLARACION DE LA RENTA y el dolor de cabeza.

El programa que el Banco de Pruebas analiza hoy, intenta simplificar un poco la realización de dicha declaración, realizando el impreso por usted.

La realización de la declaración de la renta resulta para la mayoría de la gente una tarea bastante complicada, y por lo menos Master Renta evita la posibilidad de cometer errores en los cálculos que ello lleva implícito.

En otras palabras, váyase al estanco, si todavía no lo ha hecho, cómprese el impreso, coja el librito de instrucciones que viene dentro y con los datos necesarios se puede sentar en su ordenador, y empezar a introducir los datos.

Renta para todos los Amstrad

En el disco en el que viene el programa, incluye las versiones para el CPC 6128 y para los dos modelos del PCW, si bien en el caso del CPC 6128 la impresora es imprescindible. Este disco contiene los programas necesarios del sistema operativo, por lo que para ponerlo en marcha habrá que inicializar el ordenador con este introducido, en el caso de los PCW, o teclear CPM en el caso del CPC, y el programa se pone en funcionamiento automáticamente.

El primer menú, permite optar por el tipo de declaración que se desea, bien simplificada o bien ordinaria. En el caso de la declaración simplificada, ésta se hace introduciendo los datos según el impreso de declaración D-101, mientras que en la declaración ordinaria se hace de acuerdo al impreso D-100. Independientemente del tipo de declaración que

se elija, el funcionamiento del programa resulta idéntico, variando únicamente entre una y otra declaración el contenido de las pantallas, que responde exactamente a los dos diferentes impresos.

Menús y más menús, claros y sencillos

Después de seleccionar el tipo de declaración se pasa al segundo menú, idéntico en ambas opciones, que

I. RENDIMIENTOS	INGRESOS	GASTOS	NETO (ING. - GAS.)
A.-Del trabajo	11	11	11
B.-De capital mobiliario	12	12	12
C.-De capital inmobiliario	13	13	13
D.-De activ. prof. o artist.	14	14	14
E.-De activ. empresariales	15	15	15
F.-De activ. agrarias	16	16	16
G.-En régimen de transparencia fiscal	17	17	17
H.-Rendimientos irregulares: Rendimiento medio anual	18	18	18
Resto rendimientos irregulares ... 19			
Minoración anual. por alimen. satisf. (2-X-85 a 31-XII-85) ... 20			
Rend. net. ord. (11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 - 20) ... 21			

Rendimientos implícitos sometidos a retención del 45 por 100	22	1750000
Base para el cálculo del tipo medio (22 + 23 - 24)	23	
Base 25	1750000	264320
Hasta	350000	83090
Resto		
SUMA	26	347410
TIPO MEDIO : (25 x 100) / 26 = 19.85 %		
Cuota al tipo medio (25 + 26 - 27 - 28 - 29) x tipo medio	30	347375
Cuota al tipo mínimo 30 x tipo mínimo	31	
TOTAL CUOTA INTEGRA (30 + 31)	32	347375
PULSE 1 PARA CONTINUAR O 2 PARA VOLVER AL MENU.		

A DEDUCIR:	
Retenciones por rendimientos del trabajo	33 50000
Retenciones por rendimientos del capital mobiliario	34 30000
Retenciones por rendimientos profesionales y artísticos	35
Retenciones imputables a socios de sociedades transparentes	36
Pagos fraccionados, actividades profesionales y artísticas	37
Pagos fraccionados, actividades empresariales y agrarias ..	38 80000
TOTAL retenciones y pagos fraccionados	39
CUOTA DIFERENCIAL (32 - 39) (positiva o negativa)	40 230375
PULSE 1 SI SE DESEA FRACCIONAR EL PAGO, SI NO PULSE 2	

IMPUESTO SOBRE LA RENTA DE LAS PERSONAS FÍSICAS

DECLARACIÓN SIMPLIFICADA - 1985

RENDIENTOS :	INGRESOS	GASTOS	NETO INGRESO-GASTOS	
A.- Del trabajo (01)	9000000	(02)	2300000 (03)	6770000
B.- Del capital mobiliario (04)	180000	(05)	0 (07)	180000
C.- Del capital inmobiliario (06)	210000	(08)	0 (10)	210000
D.- De act. agrarias (estimación objetiva singular simplificada) (09)				0
Anualidades por alimentos abonadas o recibidas (13)				0
Compensación base imponible negativa de años anteriores (14)				0
BASE IMPONIBLE (15)				9160000
Base (16)	9160000			
Hasta (17)	9000000	3540900		
Resto (18)	160000 al 63,30%	101400		
SUMA (19)		3540900		
TOTAL CUOTA INTEGRAL (20)			63	3540900
Rendimientos netos del segundo perceptor en importancia (21)				0
DEDUCCIONES :				
General (17.000 pts.) (22)			17000	
Por retiro (20.000 pts.) (23)			20000	
Por hijos (15.000 pts.) por cada uno (24)			0	
Por cada invidente, inválido físico o psíquico (38.000 pts.) (25)			0	
Por cada sueto pasivo o de 70 años o más (12.000 pts.) (26)			0	
Por ascendientes con menos de 500.000 pts. de ingresos (27)			0	
Gastos de enfermedad (15 por 100) (28)			4725	
Primas de seguros de vida (15 por 100 primas) (29)			9000	
Adquisición de vivienda (30)			0	
Suscripción de valores mobiliarios (31)			0	
Inversiones en bienes declarados de interés cultural (32)			0	
Inversión empresarial agraria (33)			0	
Por donación al estado o inst. benef. de bienes del P.N.E. (34)			0	
Por dividendos percibidos (35)			50000	
Por rendimientos netos del trabajo (36)			0	
Deducción variable (37)			0	
Por rendimientos de renta y plusvalía (38)			0	
Por trabajos realizados en el extranjero (39)			0	
Por compensación impuestos satisfechos en el extranjero (40)			0	
SUMA DE DEDUCCIONES (41)			100725	
DIFERENCIA (42)			3547583	
Incremento por pérdida beneficios deducción inversión años ant. (43)			0	
CUOTA LÍQUIDA (44)			3547583	
A DEDUCIR :				
Retenciones por rendimientos del trabajo (45)			3000000	
Retenciones por rendimientos del capital mobiliario (46)			25000	
Pagos fraccionados por actividades agrarias (47)			0	
TOTAL retenciones y pagos fraccionados (48)			3025000	
CUOTA DIFERENCIAL (49)			522583	
PAGO FRACCIONADO (60 por 100) (50)			313550	

presenta las cuatro posibilidades siguientes:

1. Realizar una nueva declaración, con lo cual se borrarían los datos anteriormente introducidos. Empieza a preguntar todos los datos para la declaración.

2. Modificación de los datos, que permite rectificar cualquier error cometido.

3. Listado de los datos. Permite listar por pantalla o impresora los datos introducidos anteriormente.

4. Rellenado de impresos. Permite rellenar los impresos de la declaración, una vez que se está conformando con los datos introducidos.

RENDIMIENTOS DEL TRABAJO

CONCEPTOS

Cotizaciones a la S.S., Montepíos y Mutualidades Oblig. 0

Otros Gastos 0

1 por 100 de Ingresos Integros 0

TOTAL GASTOS 0

CONFIRMA ? (S/N)

Para...
PCW

El programa no contempla la posibilidad de grabar en disco los datos introducidos, por lo que cuando se decida empezar a realizar la declaración, no se debe desconectar el aparato hasta haber terminado totalmente con ella. Cualquier interrupción haría que se perdieran todos los datos.

La primera opción permite introducir los datos de la declaración. Está estructurada en pantallas, y cada una corresponde a una sección del impreso, estando además, realizadas de manera similar a la del impreso para facilitar la labor. Dentro de éstas, se introducen los **datos**, y todos los campos que suponen cálculos fijos de porcentajes, o cantidades fijas a deducir por familia, etc., son realizados automáticamente, así como las sumas y los diferentes subtotales y totales. Las preguntas que se realizan son similares a las del impreso, y no dejan lugar a confusión. Contempla, entre otras cosas, la posibilidad de realizar los correspondientes cálculos para el pago fraccionado de la liquidación.

La modificación de datos introducidos permite modificar sólo el apartado en que exista un error. Al seleccionar esta opción, van apareciendo todas las pantallas en el mismo orden y pulsando **RETURN** el cursor pasa de apartado en apartado sin modificar los datos existentes, hasta llegar al erróneo. En éste, se tecldea el valor correcto, y luego se sigue pulsando **RETURN** hasta acabar, o llegar a otro error.

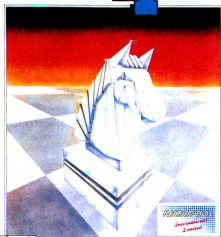
[illegible]

18 MICROHOBBY AMSTRAD

EXITOS AMSTRAD

¿SERAS CAPAZ DE PILOTAR UN "HARRIER"?

COLOSSUS CHES



CDS Software LTD

AMSTRAD DISK

HARRIER -

disco: 3.900 pts. IVA incluido
cassette: 2.500 pts. IVA incluido

COLOSSUS CHES -

disco: 4.400 pts. IVA incluido
cassette: 2.500 pts. IVA incluido

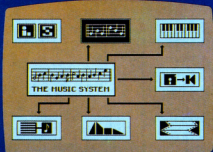


Full V.32B, High and Tumble emulation
AMSTRAD CPC-DISK

ATREVESE A ENFRENTAR AL COLOSSUS A OTRO JUEGO DE AJEDREZ

¿TE GUSTARIA IR GANANOLE "PRENDAS" A SAMANTHA?

COMMODORE
64/128K



ADVANCED THE
MUSIC SYSTEM



APPLICATION
SOFTWARE

SAMANTHA FOX -

Spectrum: 1.995 pts. IVA incluido
cas Amstrad: 1.995 pts. IVA incluido



SERMA

SISTEMA MUSICAL -

Commodore: 4.300 pts. IVA incluido
Amstrad cas: 4.300 pts. IVA incluido
Amstrad disco: 5.800 pts. IVA incluido

SAMANTHA FOX



Strip
Poker



PLUS

7 INTERNATIONAL
CARD STUD

AMSTRAD

CREA TU PROPIA MUSICA

RECORTA Y ENVIA ESTE CUPON A: ♦ SERMA C/. BRAVO MURILLO, 377 - 3.º A.
28020 MADRID. TELS. 733 73 11 - 733 74 64

HARRIER - ☐ disco ☐ cassette ☐ SAMANTHA FOX - Spectrum ☐ cas. Amstrad ☐

COLOSSUS - disco ☐ cassette ☐ S. MUSICAL - Commodore ☐ cas. Amstrad ☐ disc. Amstrad ☐

NOMBRE Y APELLIDOS: _____

DIRECCION: _____ CODIGO POSTAL: _____

POBLACION: _____ PROVINCIA: _____

FORMA DE PAGO: TALON BANCARIO ☐ CONTRA REEMBOLSO ☐



Todos los juegos a la venta
en la nueva tienda **KONAMI**
C/ Francisco Navacerrada 19.

En tu kiosk te espera algo muy inteligente

El AMSTRAD Especial número 2 incluye una cinta de casette adherida a la portada con un lenguaje Lisp completo que

te permitirá comprender y dominar las técnicas más complejas de inteligencia artificial. Por si fuera poco, en nuestra

cinta se incluyen también dos concursos: uno, de diseño gráfico de pantallas, para artistas, te permitirá ganar hasta 170.000 pesetas en premios. En el segundo regalamos un ordenador Amstrad CPC6128.

El número 2 de AMSTRAD Especial trata un amplio espectro de interesantes temas, como un comparativo de impresoras, que le ayudará a elegir la más adecuada a sus necesidades, como multitud de programas y rutinas de utilidad en lenguaje máquina como un paquete de soft integrado, con tres programas en uno y un largo etcétera que sería demasiado prolijo detallar.

Si no lo encuentras en tu kiosk, solicítalo directamente a nuestra Editorial.

MICROHOBBY
AMSTRAD
Especial

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD AÑO I N.º 2

475 ptas.

SOFTWARE INTEGRADO
3 PROGRAMAS EN UNO

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

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

NUESTRO PROGRAMA CREADOR
DE CRUCIGRAMAS DESAFIA A TU INGENIO

IMPRESORAS:
COMO HACER LA
MEJOR ELECCION

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

Re llena este cupón y envía a **HOBBY PRESS, S. A.** Ap. de Correos 232. Alcobendas, Madrid.

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

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? Si ☐ No ☐

¿Deseo recibir el Especial de **MICROHOBBY AMSTRAD** n.º 2 al precio de 475 ptas. (IVA incluido)

FORMA DE PAGO

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

A 01

COMPUTIQUE

Te da más

GARANTIA **AMSTRAD**
ESPAÑA

66.900 Ptas.

**Amstrad 464 f.
verde**

IVA incluido



VENTA A PLAZOS HASTA 36 MESES

Al comprar tu Amstrad te regalamos

- Estuche con ocho programas originales
 - Fruit Machine
 - Procesador texto
 - Almirante Graf
 - Oh Mummy
 - Plaga Galáctica
 - Amsdraw
 - Laberinto Sultan
 - Animal, Vegetal, Mineral
- Joystick Gunshot I
- Un estupendo libro de Basic
- Los cuatro mejores programas:
 - Decathlon
 - Jet Set Willy
 - Sabrewulf
 - Beach-Head
- Guía de referencia del programador
- y además te obsequiamos con un curso de introducción al Basic.



Nuevo Amstrad CPC6128: 94.900 ptas.

COMPUTIQUE

Servimos a tiendas
Abrimos sábados por la tarde

Embajadores, 90 Tfno. 2270980
28012 Madrid

MONTY ON THE RUN

Desde que Monty se fugó de la terrible prisión de Scudmore, no ha dejado de huir, aunque el camino que le separa de la libertad es largo, la aventura continúa.



Una nueva aventura de **Monty**? Pues sí, el héroe de Gremlins Graphics continúa en la brecha; su constante obsesión es escapar, y hasta que lo consiga todavía quedan unas cuantas aventuras más.

Monty on the run es la primera aventura del héroe que conocen los usuarios de **Amstrad**, pero en el Spectrum, nuestro amigo tiene una merecida fama.

Su historia comenzó con Monty is innocent, luego **Monty** on the run, y la serie no tiene fin.



Compatible: CPC144, CPC164 y CPC16128



Todas ellas basadas en los típicos juegos de laberintos, de los cuales fue precursor el famoso Manic Minner.

¡No me digas que en estos tiempos todavía no hacen programas tipo Manic Minner, si eso está más pasado que el ordenador de ocho bits!

Pues aunque así lo parezca, se siguen haciendo. Ultimamente hemos podido contemplar obras de este estilo, que de vez en cuando salen al mercado.

Aún recuerdo dos ejemplos recientes: Brian Blood Axe y Dynamite Dan, todos con la misma construcción y siguiendo de cerca la idea de su padre, el Manic Minner.

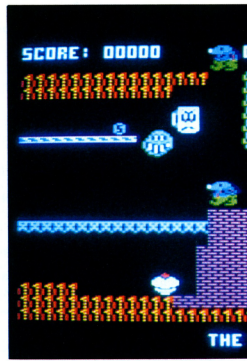
Pero los juegos de este tipo, ¿no son todos iguales? Hombre, pues sí, en ellos el reto está en pasar de una pantalla a otra, aumentando la dificultad según superamos mayor número de ellas.

Aparte de eso, ¿hay algún otro cliente? Bueno, también debemos encontrar unos objetos, que nos ayudan a aumentar nuestro marcador y a llegar al final.

¿Nada más? Que más quieres, son juegos de la prehistoria y en los albores de la programación, conseguir que algo se moviese en la pantalla ya era un logro.

¿De qué va la historia? **Monty** en su última aventura, se encontraba preso en la prisión de Scudmore, lugar del cual pocos hombres salen con vida.

Pero nuestro héroe, entre la intrincada maraña de corredores y pasos subterráneos, y gracias a la ayuda



MONTY RUN



inestimable de un amigo, ha conseguido abrirse paso hasta salir de los muros de la prisión.

Hasta aquí estábamos en su primera epopeya; Monty is innocent, en su segunda aventura, **Monty** tiene que escapar de la isla en la que está situada la prisión de alta seguridad.

Para ello sólo tiene que encontrar la barca que le conducirá hasta la ansiada libertad.

¡Pues eso debe de costar bastante trabajo? Mucho, en primer lugar tenemos que conocer las pantallas que componen el mapeado del juego, de forma que sepamos cómo pasar de una a otra y cuáles tienen salida o continen algo de interés.

¿Es difícil pasar de una pantalla a otra? Bastante, cada pantalla tiene sus trampas y elementos característicos, de forma que superarlos requiere perder un considerable número de vidas, hasta encontrar el camino correcto.

¿Cuáles son esos elementos? Tenemos de todo; prensas hidráulicas por debajo de las cuales hemos de pasar, nubes que nos elevan a las alturas, cuerdas para trepar, máquinas que nos aplastan, piscinas, monedas, cafeteras, murciélagos, etc.

¿Pues con tanto material circulando por ahí, debe ser entretenido? Lo es, porque además hay que superar una pantalla, después de lo cual te encuentras con una más difícil todavía, hecho que puede llevarte a la más absoluta demencia cibernética.

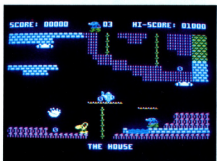
¿Este juego tiene puntos, o es de esos en que lo único es llegar al final? Los tiene, aunque lo interesante es encontrar el final; por tener, hasta tiene un ranking con las mejores diez puntuaciones, entre las cuales puedes poner la tuya.

¿A parte de eso algo interesante? En el juego hay 21 objetos diferentes, de los cuales solamente cinco componen la combinación que nos lleva hasta la barca. Un detalle interesante, es que esta combinación la podemos fijar nosotros al principio del juego, eligiendo los que más rabia nos den.

¿Y los gráficos qué tal? No están mal, usando el modo de 16 colores, los diseñadores han conseguido unas pantallas muy coloristas y de una amplia variación, contribuyendo de esta forma a amenizar el programa.

¿Qué me dices del movimiento? Bueno pues las cosas se mueven por ahí, se utilizan una gran cantidad de objetos y elementos móviles diferentes y las rutinas de movimiento cumplen su cometido.

Mister JOYSTICK



Monty da unos saltitos con doble mortal, que tienen su gracia pero la cosa no pasa de ahí, desde luego no estamos ante un Exploding fist o un Knight Lore, donde el tamaño de los personajes dificulta enormemente el movimiento.

Mientras en éstos las figuras miden ocho centímetros, **Monty** sólo mide tres.

¿Y en general qué te parece? Un juego basado en los programas de la primera generación, con gran cantidad de objetos, en el cual el mayor aliciente lo constituye el ir superando pantallas hasta conocer el total de las que componen el programa.



EL JUEGO DEL PLATO

El plato es un tradicional juego de cartas, muy sencillo, y que es más entretenido cuanto más rápido se juegue y más arriesgadas sean las apuestas.



Para empezar el programa te preguntará el número de personas que vais a jugar además de la máquina que siempre juega. Podréis jugar una, dos o tres personas.

El juego empieza repartiendo cuatro cartas a cada uno. La baraja está compuesta de cuatro palos, y en cada palo hay doce cartas. La que más valor tiene es la «A», seguida de la «K», «Q», la «J», «9» y así sucesivamente hasta llegar al «2».

La máquina realizará una primera apuesta y a continuación cada jugador hará la suya. Como hay dos rondas de apuestas, y entre las dos no se puede sobrepasar el dinero que hay en el plato, cada apuesta será, como máximo, inferior o igual a la mitad del dinero que el plato tenga en ese momento.

El siguiente paso es el descarte, primero el de la máquina, y después el de los jugadores. Las cartas están numeradas del uno al cuatro en el margen izquierdo del monitor. Para cambiar una de las cartas a la hora de descartarse, pulsa la tecla correspondiente a su número y se colocará boca abajo. Si quieres rectificar, no tienes más que volver a pulsar la misma tecla, y la carta se colocará de nuevo boca arriba. Una vez que estés boca abajo todas las cartas que desees cambiar, pulsa [ENTER], y le correrá el turno al siguiente jugador.

Una vez os hayáis descartado todos, se os repartirán cartas, y comenzará una segunda ronda de apuestas. Estas apuestas serán como las primeras; lo máximo que podréis apostar es la mitad de la cantidad que haya en ese momento en el pla-

to. Por otra parte ten en cuenta que quedarse eliminado en el momento que la cantidad que posees sea inferior a 27 ptas. Estas 27 ptas., son las necesarias para poder poner 25 ptas., en el plato cuando éste esté casi vacío, y poder hacer dos apuestas mínimas de 1 pta., cada una.

Ahora viene la parte decisiva. A cada jugador se le mostrará una carta que aparecerá en la parte superior de la pantalla. Si entre tus cuatro cartas tienes alguna del mismo palo que esa, y superior a ella, habrás ganado la suma de tus dos apuestas, y el dinero pasará del plato a tu fondo particular. Si pierdes, el dinero que hayas apostado pasará al plato.

Al principio las apuestas serán pequeñas, ya que el plato comienza con poco dinero, pero si arriesgáis un poco al principio, el plato aumentará, y las apuestas podrán ser mayores.

No pienses que la «máquina» va a ganar siempre, ya que ella sabe lo mismo que tú. Si te parece que ella gana más veces que tú es porque tiene mejor memoria; no se olvida de vuestros descartes y cuando calcula la probabilidad que tiene de ganar, tiene en cuentas todas las cartas que hay sobre el tapete.

Seguro que pronto le cojes el truco y ganas a la «máquina».



```
10 CLEAR:KEY 139,""
20 PAPER 0:BORDER 0:CLS
30 FOR C=0 TO 3:INK C,0:NEXT
40 ENT 1,15,-5,1:ENV 1,3,1,5
50 ENT 2,10,10,2:ENV 2,3,-1,7
60 ENT 3,48,-2,1:ENV 3,3,1,14
70 ENT 4,20,10,2:ENV 4,3,-1,14
80 ENT 5,30,-2,2:SYMBOL AFTER 222
90 SYMBOL 222,0,0,0,0,2,5,10,5
100 SYMBOL 223,0,0,0,0,128,64,160,8
110 SYMBOL 226,28,28,28,127,127,127,127,127,127
120 SYMBOL 227,0,28,62,127,62,28,0,8
130 SYMBOL 228,54,127,127,127,62,28,0,8
140 SYMBOL 229,0,28,62,127,127,62,0,8
150 SYMBOL 240,0,255,255
160 SYMBOL 249,0,0,0,0,255,255
170 SYMBOL 250,0,0,0,0,255,255
180 SYMBOL 251,96,96,96,96,96,96,96,96
190 SYMBOL 252,0,7,31,60,48,112,96,96
200 SYMBOL 253,0,224,248,60,12,14,6
210 SYMBOL 254,96,96,112,48,60,31,7
220 SYMBOL 255,0,14,12,60,248,224
230 MODE 1:CLEAR
240 WINDOW #0,1,40,1,19:PAPER 0:CLS
250 WINDOW #1,1,40,20,25:PAPER #1,2
260 CLS #1
270 PEN 2:LOCATE 7,12:PRINT "Cuantos
    vais a jugar (1-3) ?"
280 PEN 3:LOCATE 9,18:PRINT CHR$(24)
    " por Alejandro Gonzalez "+CHR$(2
    4)
290 INK 0,20:INK 1,20:INK 2,0:INK 3,
    6:BORDER 0
290 SOUND 7,200,20,15:PEN 1:LOCATE
    1,18:PRINT "PLATO"
300 LOCATE 1,19:PRINT STRING$(5,200)
    :t=124
310 x=120:y=375:DIM L(80)
320 FOR x=1 TO 80 STEP 2:L(x)=TES
    T(x,a):NEXT
330 FOR t=0 TO 7 STEP 2:MOVE x,y:GO
    TO 130,15,15,2,1
340 FOR x=1 TO 80 STEP 2
350 n=j=INKEY$(t+j):n=VAL(n):IF n>0
    THEN 390
360 IF L(x)=n THEN c=2 ELSE c=0
370 DRAW 10,0,c+INT(2*RN):NEXT x=y
    -3:NEXT t=t+1:IF t>10 THEN 320
380 n=j=INKEY$(t+j):n=VAL(n):IF n<1
    OR n>3 THEN 380
390 LOCATE 1,19:PRINT SPC(5):INK 1,
    24:SOUND 7,30,10,15
400 sm=INT((33-9+n)/2)+1
410 sl=sm*9+325:sl=9+325+249
420 p=25+25*n:j=975:j=975:j=975:
    j=975
430 CLS:PEN 3:LOCATE 15,14:PRINT "TR
    ANQUILO !!!"
440 PEN 2:LOCATE 8,8:PRINT "Espera a
    que baraja bien"
450 LOCATE 5,10:PRINT " (esta vez ta
    rdo 25 segundos )"
460 DIM A(47):DIM B(47):RANDOMIZE T
    IME
470 FOR t=0 TO 47
480 A(t)=INT(49*RN)
490 FOR tt=0 TO t-1
500 IF A(tt)=A(t) THEN 480
510 NEXT tt
520 FOR t=0 TO 47:B(t)=A(t):NEXT
530 CLS:MOVE 3,96:DRAW 0,304,2
540 MOVE 636,96:DRAW 0,304,2
550 LOCATE sm,6:PRINT "MAGUINA"
560 LOCATE sl+2,6:PRINT "J-1"
570 IF n=j THEN 610
580 LOCATE sl+2,6:PRINT "J-2"
590 IF n=j THEN 610
600 LOCATE sl+2,6:PRINT "J-3"
610 PEN #1,0:LOCATE #1,13,4:PRINT #
    1,"PLATO "+SPC(6)+"pts"
```


ESTRUCTURA DEL PROGRAMA

LINEAS	DESCRIPCION	LINEAS	DESCRIPCION
10-220	Definición de caracteres y envolventes.	2140-2320	manda a repartir cartas.
220-370	Definición de portada.	2320-2500	Comprobación del resultado de la máquina.
380-520	Definición de variables y confección de la baraja.	2500-2680	Comprobación del resultado del jugador 1.
530-760	Definición de la pantalla de juego.	2680-2820	Comprobación del resultado del jugador 2.
770-940	Reparto de cartas.	2820-2970	Comprobación del resultado del jugador 3.
950-1200	Apuesta de la máquina.	2970-3170	Confecciona otra baraja y reparte cartas.
1210-1320	Apuesta del jugador 1.	3170-3250	Subrutina de dibujar las cartas.
1330-1450	Apuesta del jugador 2.	3250-3340	Subrutina de dibujar las cartas boca abajo.
1460-1580	Apuesta del jugador 3.	3340-3420	Subrutina de colocar 25 pts., en el plato cuando hay menos de 10 pts., en él.
1590-1620	Comprueba si se ha hecho el descarte o no.	3420-3550	Subrutina de modificar el dinero del plato, máquina y jugadores.
1630-1640	En el margen izquierdo números de las filas de las cartas.	3550-3630	Subrutina de dibujar la carta que le toca a la máquina o jugador.
1650-1710	Descarte de la máquina.		Subrutina de final de partida.
1720-1830	Descarte del jugador 1.		
1840-1970	Descarte del jugador 2.		
1980-2100	Descarte del jugador 3.		
2110-2130	Borra número de margen izquierda y		

Serie
ORO

```

620 RESTORE 650:FOR t=1 TO 3:READ m
v,mv,dx,dy
630 MOVE mx,mv:DRAW dx,0,0:DRAW 0
,dy
640 DRAW -dx,0:DRAW 0,-dy:NEXT
650 DATA 172,28,296,24,142,26,356,2
8,182,24,436,32
660 LOCATE #1,sm,1:PRINT #1,CHR$(15
0):STRING$(5,155):CHR$(156)
670 LOCATE #1,sm,2:PRINT #1,CHR$(14
5):"":SPC(4):CHR$(145)
680 LOCATE #1,sm,1:PRINT #1,CHR$(15
0):STRING$(5,155):CHR$(156)
690 LOCATE #1,sm,1:PRINT #1,CHR$(14
5):"":SPC(4):CHR$(145)
700 IF nj=1 THEN 760
710 LOCATE #1,s2,1:PRINT #1,CHR$(15
0):STRING$(5,155):CHR$(156)
720 LOCATE #1,s2,2:PRINT #1,CHR$(14
5):"":SPC(4):CHR$(145)
730 IF nj=2 THEN 760
740 LOCATE #1,s3,1:PRINT #1,CHR$(15
0):STRING$(5,155):CHR$(156)
750 LOCATE #1,s3,2:PRINT #1,CHR$(14
5):"":SPC(4):CHR$(145)
760 GOSUB 3350
770 IF E(0)=1 THEN LOCATE sm+1,12:P

```

```

RINT"FUERA":GOTO 810
790 IF jj=0 THEN pc=pc+4 ELSE SOUND
1,110,60,12,3,5
790 C(0)=A(0):C(1)=A(1):C(2)=A(2):C
(3)=A(3)
800 x=ms:GOSUB 3010
810 IF E(1)=1 THEN LOCATE s1+1,12:P
RINT"FUERA":GOTO 850
820 IF jj=0 THEN pc=pc+4 ELSE SOUND
1,110,60,12,3,5
830 C(0)=A(4):C(1)=A(5):C(2)=A(6):C
(3)=A(7)
840 x=s1:GOSUB 3010
850 IF nj=1 THEN 900
860 IF E(2)=1 THEN LOCATE s2+1,12:P
RINT"FUERA":GOTO 900
870 IF jj=0 THEN pc=pc+4 ELSE SOUND
1,110,60,12,3,5
880 C(0)=A(8):C(1)=A(9):C(2)=A(10):
C(3)=A(11)
890 x=s2:GOSUB 3010
900 IF nj=2 THEN 900
910 IF E(3)=1 THEN LOCATE s3+1,12:P
RINT"FUERA":GOTO 900
920 IF jj=0 THEN pc=pc+4 ELSE SOUND
1,110,60,12,3,5
930 C(0)=A(12):C(1)=A(13):C(2)=A(14)

```

```

):C(3)=A(15)
940 x=s3:GOSUB 3010
950
960
970
980 a$=INKEY:a$="":IF E(0)=1 THEN
1210
990 SOUND 7,100,20,15
1000 LOCATE #1,16,6:PRINT #1,"Yo ap
uesto"
1010 FOR t=0 TO 3:FOR tt=0 TO 3
1020 IF t=tt THEN 1060
1030 IF INT(A(t)/12)<INT(A(tt)/12)
THEN 1060
1040 m=MIN(A(t),A(tt))
1050 IF A(t)=m THEN A(t)=1 ELSE A
(tt)=1
1060 NEXT:tt=1
1070 FOR tt=0 TO 3
1080 ptt=INT(A(tt)/12):intt=A(tt)-12
*ptt
1090 FOR t=ptt TO 47
1100 ptt=INT(A(t)/12):intt=A(t)-12*ptt
1110 IF nt=nt AND ptt=ptt AND A(t)<
>-1 THEN ptt=ptt+1
1120 NEXT:tt=ptt/(48-ptt)
1130 am=ROUND(pmt+p/5,-1)/2
1140 IF am/p/2 OR(p>0.9 AND m/p+
0.0) THEN am=INT(p/2)
1150 IF am1>0 OR am1=0 THEN am1=-1
1160 IF am1=0 THEN am1=1
1170 am=am+am1
1180 IF am1=0 THEN am=0
1190 SOUND 7,30,10,15
1200 LOCATE sm-1,2:PRINT am:CHR$(8)
:"pts"
1210 FOR t=1 TO 800:NEXT
1220 IF E(1)=1 THEN 1340
1230 SOUND 7,100,20,15
1240 LOCATE #1,5,6:INPUT #1,"Cual e
s la apuesta del J-1 ? ":aj1
1250 aj1=INT(ABS(aj1)):IF aj1>1
-1 THEN 1270
1260 IF (aj1/p/2 OR aj1=p/2) AND
aj1>10 THEN 1280
1270 LOCATE #1,5,6:PRINT #1,SPC(34)
:SOUND 7,50,10,15,,5:GOTO 1240
1280 aj1=aj1+aj1
1290 IF aj1>1 THEN aj1=aj1-aj1:GOTO
TO 1270
1300 LOCATE #1,5,6:PRINT #1,SPC(34)
1310 SOUND 7,30,15
1320 LOCATE s1-1,2:PRINT aj1:CHR$(
8):"pts"
1330 FOR t=1 TO 800:NEXT
1340 IF nj=1 THEN 1570
1350 IF E(2)=1 THEN 1470
1360 SOUND 7,100,20,15
1370 LOCATE #1,5,6:INPUT #1,"Cual e
s la apuesta del J-2 ? ":aj2
1380 aj2=INT(ABS(aj2)):IF aj2>1
-1 THEN 1400
1390 IF (aj2/p/2 OR aj2=p/2) AND
aj2>10 THEN 1420
1400 LOCATE #1,5,6:PRINT #1,SPC(34)
:SOUND 7,50,10,15,,5:GOTO 1370
1410 aj2=aj2+aj2
1420 IF aj2>1 THEN aj2=aj2-aj2:GOTO
TO 1400
1430 LOCATE #1,5,6:PRINT #1,SPC(34)
1440 SOUND 7,30,10,15
1450 LOCATE s2-1,2:PRINT aj2:CHR$(
8):"pts"
1460 FOR t=1 TO 800:NEXT
1470 IF nj=2 OR E(3)=1 THEN 1570
1480 SOUND 7,100,20,15

```

```

1490 LOCATE #1,5,6:INPUT #1,"Cual e
s la apuesta del J-3 ? ",a33
1500 a33=INT(ABS(a33)):IF a33>3
-1 THEN 1520
1510 IF (a33/p/2 OR a33=p/2) AND
a33>0 THEN 1530
1520 LOCATE #1,5,6:PRINT #1,SPC(34):
SOUND 7,50,10,15,,,5:GOTO 1490
1530 a3=a3+a33
1540 IF a3>3 THEN a3=a3-a33:GOTO
1520
1550 SOUND 7,30,10,15
1560 LOCATE 4,3,1,2:PRINT a3:CHR$(
8):"pts"
1570 LOCATE #1,5,6:PRINT #1,SPC(35)
1580 FOR t=1 TO 800:NEXT:pc=pc+4
1590 IF jj=1 THEN jj=0:GOTO 2170
1600 /
1610 / DESCARTE
1620 /
1630 FOR t=1 TO 4:LOCATE sm-2,6+3*t
1640 PRINT USING "M":15-t:NEXT
1650 IF E(0)=1 THEN 1720
1660 SOUND 7,50,10,15
1670 LOCATE #1,15,6:PRINT #1,"Mi des
carte"
1680 x=sm:FOR t=0 TO 3
1690 IF A(t)=1 OR A(t)=12*INT(A(t)
/12)+(5+INT(2*pt)) THEN 1710
1700 NEXT:t=0:GOTO 1720
1710 GOSUB 3180:A(t)=A(pc):pc=pc+1:
GOTO 1700
1720 IF E(1)=1 THEN 1840
1730 FOR t=1 TO 800:NEXT:t=SOUND 7,50
,20,15
1740 LOCATE #1,9,6:PRINT #1,"Descart
e del jugador 1 ?":x=sl
1750 t=INKEY$:t=VAL(t)
1760 IF INKEY$(19)<1 OR INKEY$(6)<
-1 THEN 1810
1770 IF t=1 OR t=4 THEN 1750
1780 t=t-1:IF A(t+4)=1 THEN A(t+4)
=B(t+4) ELSE 1820
1790 C(4)=A(t+4):GOSUB 3040:GOTO 17
50
1800 A(t+4)=1:GOSUB 3180:GOTO 1750

```



```

1810 FOR t=4 TO 7
1820 IF A(t)=1 THEN A(t)=A(pc):pc=
pc+1:GOTO 1820
1830 NEXT
1840 IF n=1 THEN 2110
1850 IF E(2)=1 THEN 1880
1860 LOCATE #1,9,6:PRINT #1,SPC(24)
1870 FOR t=1 TO 800:NEXT:t=SOUND 7,50
,20,15
1880 LOCATE #1,9,6:PRINT #1,"Descart
e del jugador 2 ?":x=s2
1890 t=INKEY$:t=VAL(t)
1900 IF INKEY$(18)<1 OR INKEY$(6)<
-1 THEN 1950
1910 IF t=1 OR t=4 THEN 1890
1920 t=t-1:IF A(t+8)=1 THEN A(t+8)
=B(t+8) ELSE 1940
1930 C(4)=A(t+8):GOSUB 3040:GOTO 18

```

VARIABLE DESCRIPCION

nj Número de jugadores.

sm Localización en pantalla de las cartas de la máquina.

sl Localización en pantalla de las cartas del jugador 1.

s2 Localización en pantalla de las cartas del jugador 2.

s3 Localización en pantalla de las cartas del jugador 3.

p Dinero que hay en el plato.

m Dinero que tiene la máquina.

j1 Dinero del jugador 1.

j2 Dinero del jugador 2.

j3 Dinero del jugador 3.

A Matriz que contiene la baraja.

B Matriz que contiene la baraja y se mantiene igual durante una partida.

C Matriz que contiene las cartas en juego.

E Matriz que contiene jugadores eliminados.

pc Primera carta de la baraja.

pr Probabilidad de que la máquina gane.

am1 Apuesta provisional de la máquina.

am Apuesta de la máquina.

aj11 Apuesta provisional del jugador 1.

aj1 Apuesta del jugador 1.

aj22 Apuesta provisional del jugador 2.

aj2 Apuesta del jugador 2.

aj33 Apuesta provisional del jugador 3.

aj3 Apuesta del jugador 3.

nt Número de carta del jugador o de la máquina.

ppc Palo de la carta que le toca a cada jugador.

npc Número de la carta que le toca a cada jugador.

pq Variable de dibujo de la carta con su palo.

q Variable de dibujar el número de la carta.

```

90
1940 A(t+8)=1:GOSUB 3180:GOTO 1890
1950 FOR t=6 TO 11
1960 IF A(t)=1 THEN A(t)=A(pc):pc=
pc+1:GOTO 1960
1970 NEXT
1980 IF n=2 OR E(3)=1 THEN 2110
1990 LOCATE #1,9,6:PRINT #1,SPC(24)
2000 FOR t=1 TO 800:NEXT:t=SOUND 7,50
,20,15
2010 LOCATE #1,9,6:PRINT #1,"Descart
e del jugador 3 ?":x=s3
2020 t=INKEY$:t=VAL(t)
2030 IF INKEY$(18)<1 OR INKEY$(6)<
-1 THEN 2080
2040 IF t=1 OR t=4 THEN 2020
2050 t=t-1:IF A(t+12)=1 THEN A(t+1
2)=B(t+12) ELSE 2070
2060 C(4)=A(t+12):GOSUB 3040:GOTO 2
020
2070 A(t+12)=1:GOSUB 3180:GOTO 202
0
2080 FOR t=12 TO 15
2090 IF A(t)=1 THEN A(t)=A(pc):pc=
pc+1:GOTO 2090
2100 NEXT
2110 LOCATE #1,9,6:PRINT #1,SPC(24)
2120 FOR t=9 TO 18 STEP 9:LOCATE sm
-2,t:PRINT " " :NEXT
2130 FOR t=1 TO 1000:NEXT:jj=1:GOTO
770
770
2140 /
2150 / RESULTADOS
2160 /
2170 IF E(0)=1 THEN 2340
2180 x=sm:GOSUB 3430
2190 IF am>0 THEN am=pc
2200 FOR t=0 TO 3
2210 pt=INT(A(t)/12):int=A(t)-12*pt
2220 IF nt>npc AND pt=pc THEN 2270
2230 NEXT
2240 SOUND 7,60,40,15,4,4
2250 LOCATE #1,9,6:PRINT #1,"La maqu
ina pierde":jam="pts"
2260 FOR t=1 TO 1000:NEXT:tp=am+im=
m-am:GOTO 2300
2270 SOUND 7,120,40,12,3,3
2280 LOCATE #1,10,6:PRINT #1,"La maq
uina gana":jam="pts"
2290 FOR t=1 TO 1000:NEXT:tp=am+im=
m+am
2300 IF m/27 THEN E(0)=1:m=0
2310 GOSUB 3350:FOR t=1 TO 1000:NEX
T:GOSUB 3240
2320 pc=pc+1
2330 LOCATE #1,9,6:PRINT #1,SPC(28)
2340 IF E(1)=1 THEN 2510
2350 x=sl:GOSUB 3430
2360 IF aj1>0 THEN aj1=p
2370 FOR t=0 TO 7
2380 pt=INT(A(t)/12):int=A(t)-12*pt
2390 IF nt>npc AND pt=pc THEN 2440
2400 NEXT
2410 SOUND 7,60,40,15,4,4
2420 LOCATE #1,7,6:PRINT #1,"El juga
dor 1 pierde":aj1="pts"
2430 FOR t=1 TO 1000:NEXT:tp=aj1+j1
1=aj1-aj1:GOTO 2470
2440 SOUND 7,120,40,12,3,3
2450 LOCATE #1,8,6:PRINT #1,"El juga
dor 1 gana":aj1="pts"
2460 FOR t=1 TO 1000:NEXT:tp=aj1+j1
1=aj1+aj1
2470 IF j1<27 THEN E(1)=1:j1=0
2480 GOSUB 3350:FOR t=1 TO 1000:NEX
T:GOSUB 3240
2490 pc=pc+1
2500 LOCATE #1,7,6:PRINT #1,SPC(38)
2510 IF E(1)=1 THEN 2830
2520 IF E(2)=1 THEN 2690
2530 x=s2:GOSUB 3430
2540 IF aj2>0 THEN aj2=p
2550 FOR t=0 TO 11
2560 pt=INT(A(t)/12):int=A(t)-12*pt
2570 IF nt>npc AND pt=pc THEN 2620

```



Serie ORO

```

2580 NEXT
2590 SOUND 7,60,40,15,4,4
2600 LOCATE #1,7,6:PRINT#1,"El Jugu
dor 2 pierde"aj2i:"pts"
2610 FOR t=1 TO 1000:NEXT:tp=p+aj2i
2620 SOUND 7,120,40,12,3,3
2630 LOCATE #1,8,6:PRINT#1,"El Jugu
dor 2 gana"aj2i:"pts"
2640 FOR t=1 TO 1000:NEXT:tp=p+aj2i
2650 IF J2(27 THEN E(2)=1:J2=0
2660 GOSUB 3350:FOR t=1 TO 1000:NEX
T:IGOSUB 3240
2670 pcp=+1
2680 LOCATE #1,7,6:PRINT #1,SPC(32)
2690 IF nJ=2 OR E(3)=1 THEN 2830
2700 s=3:GOSUB 3430
2710 IF aJ(3) THEN aJ=0
2720 FOR t=2 TO 15
2730 p=INT(a(t)/12):intw(a(t)-12)*t
2740 IF nt=ncp AND tp=pcp THEN 2780
2750 NEXT:t=SOUND 7,60,40,15,4,4
2760 LOCATE #1,7,6:PRINT#1,"El Jugu
dor 3 pierde"aj3i:"pts"
2770 FOR t=1 TO 1000:NEXT:tp=p+aj3i
2780 SOUND 7,120,40,12,3,3
2790 LOCATE #1,8,6:PRINT#1,"El Jugu
dor 3 gana"aj3i:"pts"
2800 FOR t=1 TO 1000:NEXT:tp=p+aj3i
2810 s=3*aj3
2820 IF J3(27 THEN E(3)=1:J3=0
2830 GOSUB 3350:FOR t=1 TO 1000:NEX
T:IGOSUB 3240
2840 nJ=0:FOR t=0 TO 3
2850 IF t=nJ THEN 2860
2860 NEXT:IF nJ=2 THEN 3560
2870 WINDOW #2,sm,38,8,19:PAPER #2,
0:CLS #2
2880 WINDOW #2,sm,38,1,3:PAPER #2,0
:CLS #2
2890 LOCATE #1,7,6:PRINT #1,SPC(32)
2900 LOCATE 10,18:PRINT"Espera a qu
e baje"
2910 FOR t=1 TO 120
2920 SOUND 1,100,0,5,13,,5:ti=INT(4
8*RAND)
2930 ti=INT(48*RAND):IF i=J THEN 2930
2940 B(1)=B(1)+B(1)+B(1)*x
2950 NEXT:FOR t=0 TO 47A(t)=B(t):IN

```

```

EXT
2960 am=0:aj1=0:aj2=0:aj3=0:pcp=0:J
=0
2970 LOCATE 10,18:PRINT SPC(20):GOT
O 770
2980 /
2990 /
3000 / SUBROUTINAS
3010 FOR t=0 TO 3
3020 p=INT(C(t)/12):q=C(t)-12*pq
3030 GOSUB 3050:NEXT:RETURN
3040 p=INT(C(4)/12):q=C(4)-12*pq
3050 IF J=0 THEN SOUND 7,115,15,12
,1,1
3060 PAPER 1:LOCATE x,19-3*t:PEN 3
3070 PRINT CHR$(251):SPC(5):CHR$(25
0)
3080 LOCATE x,18-3*t:PRINT CHR$(251
):PEN 2
3090 IF q=1 THEN PRINT"A":IGOTO 31
40
3100 IF q=10 THEN PRINT"K":IGOTO 31
40
3110 IF q=9 THEN PRINT"0":IGOTO 314
0
3120 IF q=8 THEN PRINT"J":IGOTO 314
0
3130 PRINT USING "M":2+q
3140 IF pq=1 OR pq=2 THEN PEN 3 ELS
E PEN 2
3150 PRINT CHR$(226+pq):PEN 3:PRIN
T SPC(3):CHR$(250)
3160 LOCATE x,17-3*t:PRINT CHR$(252
):STRING$(5,248):CHR$(253)
3170 PAPER 0:RETURN
3180 SOUND 7,50,20,15,2,2
3190 WINDOW #2,x,x+6,17-3*t,19-3*t
3200 LOCATE #2,2,2:PAPER #2,1:PRINT
#2,SPC(2)
3210 PRINT #2,CHR$(22):CHR$(1):
3220 PEN #2,3:LOCATE #2,1,1:PRINT #
2,CHR$(222):STRING$(5,218):CHR$(223
):
3230 LOCATE #2,1,2:PRINT #2,CHR$(21
7):STRING$(5,207):CHR$(219):
3240 LOCATE #2,1,3:PRINT #2,CHR$(21
7):STRING$(5,207):CHR$(219):
3250 PRINT #2,CHR$(22):CHR$(0):RET
URN
3260 IF p>9 THEN RETURN
3270 SOUND 7,25,10,15,,1
3280 IF E(0)=0 THEN p=p+25:inn=25
3290 IF E(1)=0 THEN p=p+25:1=1+25

```

```

3300 IF nJ=1 THEN 3340
3310 IF E(2)=0 THEN p=p+25:J2=J2-25
3320 IF nJ=2 THEN 3340
3330 IF E(3)=0 THEN p=p+25:J3=J3-25
3340 GOSUB 3350:RETURN
3350 PEN #1,1:LOCATE #1,21,4:PRINT
#1,USING "*****"10
3360 LOCATE #1,sm+2,2:PRINT #1,USIN
G "*****"1m
3370 LOCATE #1,s1+2,2:PRINT #1,USIN
G "*****"1j
3380 IF nJ=1 THEN RETURN
3390 LOCATE #1,s2+2,2:PRINT #1,USIN
G "*****"12
3400 IF nJ=2 THEN RETURN
3410 LOCATE #1,s3+2,2:PRINT #1,USIN
G "*****"13
3420 RETURN
3430 pcp=INT(A(p)/12):ncp=A(p)-12
*pcp
3440 PAPER 1:LOCATE x,1:PEN 3
3450 PRINT CHR$(251):SPC(5):CHR$(25
0)
3460 LOCATE x,2:PRINT CHR$(251):SPC
(3):PEN 2
3470 IF npc=1 THEN PRINT"A":IGOTO 3
520
3480 IF npc=10 THEN PRINT"K":IGOTO 3
520
3490 IF npc=9 THEN PRINT"0":IGOTO 3
520
3500 IF npc=8 THEN PRINT"J":IGOTO 3
520
3510 PRINT USING "M":12+ncp
3520 IF pcp=1 OR pcp=2 THEN PEN 3 E
LSE PEN 2
3530 PRINT CHR$(226+pcp):PEN 3:PRIN
T CHR$(250)
3540 LOCATE x,3:PRINT CHR$(254):STR
ING$(5,249):CHR$(255)
3550 PAPER 0:RETURN
3560 LOCATE #1,2,6:PRINT #1,SPC(36)
3570 FOR t=0 TO 3:IF E(t)=0 THEN 35
80 ELSE NEXT
3580 a=**"EL JUGADOR "+STR$(t):IF t=
0 THEN a="la MAQUINA"
3590 SOUND 7,100,20,15,,3:SOUND 7,6
0,20,15
3600 LOCATE #1,13,6:PRINT #1,"Gano
"ja8
3610 FOR t=1 TO 2000:NEXT:SOUND 7,1
50,20,15,2
3620 LOCATE #1,1,6:PRINT #1,"Para e
mpezar otra partida pulsa [ENTER]."
1
3630 IF INKEY(18)=1 THEN 3630 ELSE
SOUND 7,30,25,15:GOTO 230

```



P ara que los datos no se pierdan en el tiempo, el **AMSTRAD** le hace saber a todos los usuarios que existen este logotipo en su computadora y a disposición en un caso de emergencia, solicítelo.

SOMOS MAYORISTAS

MICRO-1

PRECIOS
INCLUIDO IVA

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 274 53 80

(Metro O'Donnell o Goya)
Aparcamiento gratuito en Felipe II

Ofertas en software: 2 programas al precio de 1 y además regalo fin de curso una calculadora completamente gratis. ¡¡Asombroso!!
¿Verdad?

BAT MAN	2.300	ptas.	KUNG-FU MASTER	2.300	ptas.
ROCK'N LUCHA	2.300	ptas.	SABOTEUR	2.300	ptas.
YIER AR KUNG FU	2.300	ptas.	PING PONG	2.300	ptas.
THE WAY OF THE TIGER	2.300	ptas.	MILLION II	0000	ptas.
WEST BANK	2.100	ptas.	OLE TORO	2.300	ptas.
CAMELOT WARRIORS	2.300	ptas.	TURBO ESPRIT	2.100	ptas.
RAMBO	2.300	ptas.	SABRE WOLF	1.650	ptas.
WORLD CUP (DISCO)	3.300	ptas.	BATALLA DE LOS PLANETAS	2.100	ptas.
MILLION II (DISCO)	3.300	ptas.	SABOTEUR-COMBAT LINX DISCO	3.300	ptas.

RAMBO-MATCH DAY (DISCO) 3.300 ptas.

SOFTWARE DE REGALO (OFERTA 2x1)
DECATHLON BEACH HEAD SOUTHERN BELLE
DRAGONTORC

LAPIZ OPTICO
3.295 PTAS.

CASSETTE ESPECIAL ORDENADOR
5.295 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
7.650 PTAS.

AMPLIACION DE MEMORIA ANTA 64 K.3
12.500 PTAS.

IMPRESORAS
20% DE DESCUENTO SOBRE P.V.P.

TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLE SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69

CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLE SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3"	990

**PRECIOS EXCEPCIONALES PARA TU AMSTRAD
CPC-464, CPC-6128, PCW-8256**

¡¡LLEGARON LAS REBAJAS DE VERANO A MICRO 1!!

OFERTAS EN JOYSTICKS

QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT V	1.695

**PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO TEL. (91) 274 53 80
O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.**

Tiendas y distribuidores grandes descuentos.
Dirigirse a Diproimsa. C/ Galatea, 25. Tel. (91) 742 20 19 ó 742 79 68

Bajo el original título de «LA LUPA» os presentamos un programa que, como su propio nombre indica, aumenta de tamaño los distintos caracteres que introduzcáis.

A

unque no se especifica en el texto del programa, el número máximo de caracteres aceptados es de treinta.

El desglose de las líneas es el siguiente:

10-40 Los clásicos REMs que nos indican el título. Sirven para recordarnos de qué va a tratar el programa.

50 REM con el que comienza la entrada de los datos y los caracteres.

60 Creación de la ventana #1, con WINDOW, dimensionada y posicionada según los parámetros indicados.

70 Inicio de un bucle WHILE... WEND cuyo cuerpo se va a estar repitiendo continuamente.

80 Se limpia la pantalla.

90 En el canal #1 se pregunta por la anchura de los caracteres, asignando el valor teclado a la variable «anchura». A continuación se limpia la ventana #1 definida anteriormente.

100 Del mismo modo se nos pide la altura de los mismos. Contestando a la instrucción INPUT de esta línea determinamos el valor que

va a contener «altura». Además dejamos en blanco la ventana #1 mediante CLS #1.

110 Por último nos dice que introduzcamos la palabra o los caracteres cuyo tamaño queremos aumentar. Introducimos en esta cadena literal la variable «palabra\$».

120 Bifurcación condicional. Especificamos que si la longitud de la variable literal (LEN(palabra\$)) es mayor que 5, la ejecución del programa volverá a la línea 110.

130 REM que nos informa que comienza la parte del programa cuya misión es sacar en la pantalla la ampliación de la palabra teleada.

140 Se escribe en la ventana #1 el carácter introducido en la línea 110. Es la «muestra» a copiar.

150 Bucle FOR... NEXT de variable «p0» y que rota desde 0 hasta la longitud del carácter menos 1. Se emplea para posicionar cada uno de los letras.

160 Un nuevo ciclo controlado por «p1». Nos sirve para recorrer en horizontal cada uno de los pixels que forman una letra o carácter.

170 Otro bucle igual que el anterior pero con «p2» como variable de control. En esta ocasión los recorre en vertical.

180 Con el comando TEST se traslada el cursor gráfico hasta cada uno de las distintas coordenadas «x» ($p1 + p0 \cdot 16$) e «y» ($p2$) especificadas mediante los valores que van tomando las variables de control en los bucles anteriores.

190 Si «tinta» es distinta de 0, o sea, si el pixel analizado tiene un color distinto del que tenemos como fondo, saltamos a la subrutina que se va a encargar de hacer el efecto de lupa: AMPLIARLOS.

200 Se cierran los distintos bucles FOR... NEXT.

210 Borra lo escrito en la ventana #1, que es precisamente el contenido de la variable de la cadena «palabra\$».

220 Línea que nos indica la manera de continuar.

230 El programa se detiene. No continúa hasta que pulsemos una tecla nuevamente.

240 En esta línea termina el bucle WHILE... WEND que comenzaba en la 70.

250 Comentario informático sobre la rutina posterior que va a intentar ampliar el tamaño de los pixels de la «muestra» que estén iluminados.

260 Bucle posterior de los dos que hay anidados para expandir cada uno de los pixels coloreados. Este primero le empleamos para ampliarlo horizontalmente. Para ello la variable de control «i» va desde 0 hasta el contenido de «anchura».

270 Aquí comienza el bucle interior. Con él vamos a poder ampliar la altura de los caracteres de la palabra elegida.

280 Dibujamos un punto en la pantalla. Sus coordenadas gráficas quedarán determinadas por la evaluación de las expresiones matemáticas que hay en los paréntesis.

290-300 Terminación de ambos bucles.

310 Instrucción que nos devuelve a la línea del programa principal siguiente a la que hacía la llamada a la subrutina.



¿ESTA PULSANDO UNA TECLA?

La última vez que nos reunimos nos fue presentada una nueva función: INKEY. Utilizándola podíamos mover un objeto a lo largo y ancho de la pantalla, o por lo menos esa fue la «tarea» que les dejamos.



Pero esta semana añadiremos más dimensiones y utilidades a todas las funciones que investigan si hemos pulsado o no una tecla. Además de desplazar o **pasar** nuestro móvil pueden servirnos para muchas más cosas. Procuraremos ver todas ellas.

¿Qué tal las **«deberes»**? Estamos seguros que por lo menos habrá intentado conseguir aplicar correctamente la función INKEY para lograr resolver el **«problema»** propuesto. ¿Recordamos por encima qué es lo que hace esta función?

INKEY simplemente investiga si se ha pulsado o no una tecla en particular. Cada una de ellas tiene un número asociado que la distingue de las demás: es el que vamos a utilizar para detectar si alguna en particular, la barra del espacio por ejemplo, está pulsada.

Queremos saber si la que está presionada es el espacio. Bueno, pues aplicamos esta función al número 47 que es el asignado a dicha tecla. Que nos devuelva un valor diferente de -1, la conclusión que podemos sacar es que si tenemos nuestro dedo apretando la tecla.

Por tanto, en nuestro programa tendremos que colocar una instrucción semejante a:

```
IF INKEY(47) < + -1 THEN...
```

Si en lugar de la barra espaciadora la tecla a investigar es la **«a»** lo primero que tendremos que hacer es averiguar el número que tiene asignado. Para ello lo mejor es echar mano del manual que nos entregaron con nuestro **Amstrad**, claro, y seguro que allí existe una tabla que nos responderá a todas nuestras preguntas.

En el caso que nos ocupa veremos que la **«a»** tiene asociado el número 69. Entonces bastará con sustituir la línea anterior por:

```
IF INKEY(69) < + -1 THEN
```

Y así para cualquier tecla. Pero, ¿qué ocurre si en lugar de la **«a»** queremos saber si la que se ha pulsado es la **«A»**? Si el análisis lo realizamos como **«letra»**, la diferencia es bien sencilla: una es mayúscula y la otra es minúscula. La función INKEY\$ lo tendrá claro. Las instrucciones necesarias serían:

```
IF INKEY$ = «A» THEN...
```

```
IF INKEY$ = «a» THEN...
```

Pero, ¿qué pasa con INKEY? ¿No nos servirá para conseguir los mismos fines?

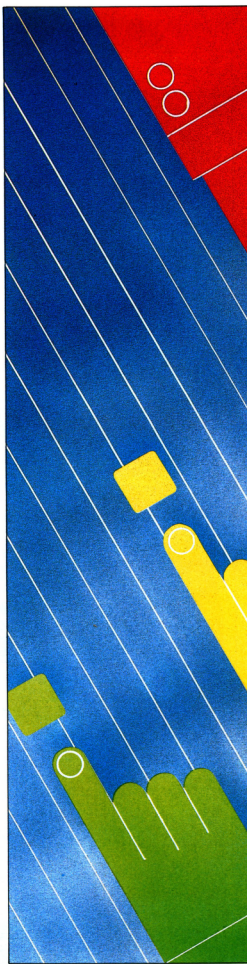
No se preocupe, como siempre nuestro ordenador tiene la solución. Para verla más gráficamente lo remitimos a una variante del Programa 5 de nuestro anterior artículo. La única diferencia es la tecla elegida y el mensaje que nos aparece en la pantalla.

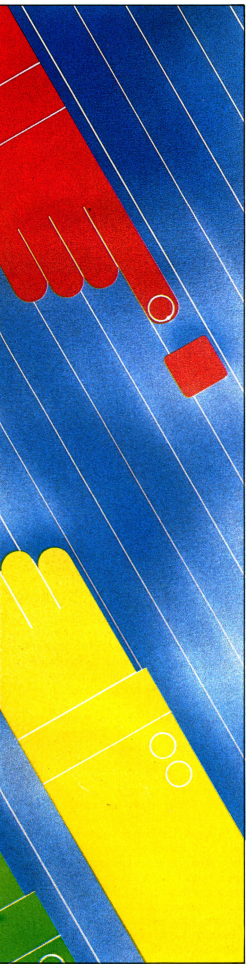
Programa uno

Con él conseguimos determinar si hemos pulsado la tecla **«a»** sola o acompañada de la de las mayúsculas, la tecla **CONTROL**, o si tenemos las tres a la vez.

La diferencia existente va a residir en el valor que nos va a devolver la función INKEY cuando estemos en una u otra situación.

Cuando no presionemos ninguna tecla, INKEY (69) nos va a estar dando como resultado el valor -1. Ya lo habíamos comentado. Por tanto si ejecutamos el programa es evidente que, como almacenamos este valor generado en la variable **«valor»**, no se cumplirán ninguna de las condiciones que hemos colocado en las líneas 60 6 90 y continuará hasta la 100 que será un «salto atrás» — a la línea 30 — para repetir de nuevo el bucle.





¿Cuándo terminará el programa? Pues en el momento que se cumpla una de dichas condiciones. Pruebe pulsar solamente la tecla «a». En la pantalla nos va a salir el valor generado por la función INKEY(69) y las conclusiones que el ordenador saca de ello.

Resultado que el valor generado, y almacenado en la variable «valor» es cero. Se cumplirá entonces la condición de la línea 60 y el mensaje aparecido nos ratificará nuestra acción. Además el programa termina debido a la instrucción END que hemos colocado al final de la línea.

```
10 REM PROGRAMA 1
20 CLS
30 LOCATE 1,1
40 valor=INKEY(69)
50 PRINT"VALOR GENERADO:";valor
60 IF valor=0 THEN PRINT"HAS PULSADO SEÑALMENTE LA TECLA A":END
70 IF valor=32 THEN PRINT"HAS PULSADO LA TECLA A Y LA DE MAYUSCULAS":END
80 IF valor=128 THEN PRINT"HAS PULSADO LA TECLA A Y LA DE CONTROL A LA VEZ":END
90 IF valor=160 THEN PRINT"HAS PULSADO LAS TRES TECLAS A LA VEZ":END
100 GOTO 30
```

Ejecutemos de nuevo el programa pero ahora variemos las condiciones. Pulsemos, por ejemplo, las teclas **CONTROL** y «a» a la vez. Ahora el **Amstrad** nos comunica que el valor generado es 128 y, por tanto, se cumplirá la condición de la línea 80.

Continuemos investigando qué pasa al pulsar la tecla de la «a» junto con la de «**SHIFT**» (o mayúsculas) o al presionar las tres a la vez. En la pantalla nos aparece lo que genera INKEY(69) junto con un mensaje explicativo. ¿Coincide con lo que usted dedujo al hacer distintas pruebas con el anterior programa 4? Esperamos que así haya sido.

Pero ahora vayamos con la tarea pendiente. Se trata de hacer que nuestro Don Sonrisitas (todavía no habíamos hablado hoy de él) se desplace o pasee, como guste, por la pantalla empleando para ello las teclas de los cursores. Nos parece que son las que nos muestran más gráficamente la dirección y el sentido hacia donde se va a producir el movimiento.

Cojamos el manual. ¿Qué número tienen asignados dichas teclas? No encontrará ninguna dificultad en ver que son los siguientes:

Izquierda: número de tecla 8

Derecha: número de tecla 1

Arriba: número de tecla 0

Abajo: número de tecla 2

Así que nuestro programa ha de tener unas instrucciones que nos detecten si alguna de ellas está pulsada y entonces variar la situación del objeto que queremos mover. Por ejemplo, si se cumple la condición:

$INKEY(0) < + - 1$

el ordenador interpreta, y con acierto, que hemos presionado la tecla de los cursores que tiene la flecha hacia arriba y que, por tanto,

Primeros PASOS

tiene que ejecutar las instrucciones que desplacen a Don Sonrisitas en esa dirección.

Sin embargo, si la que se cumple es:

$INKEY(8) < + - 1$

la conclusión que saca en esta ocasión es que el movimiento se hará hacia la izquierda y por tanto habrá que variar las coordenadas del punto donde vamos a colocar a nuestro amigo del modo conveniente para que así ocurra.

Y, ¿qué pasará si lo que se verifica es:

$INKEY(2) < + - 1$?

Saque usted mismo sus conclusiones, seguro que no se equivocará.

En esta ocasión no limitaremos el radio de acción de lugar donde se va a producir el **movimiento**. Cuando el objeto llegue a los límites de la pantalla haremos que aparezca de nuevo pero en el lugar opuesto al que tenía. Es decir, llega al borde inferior y en vez de salirse y desaparecer, volvemos a tenerle con nosotros pero ahora en el borde superior. ¿Comprendido?

Y hemos llegado al punto clave de este asunto. ¿Cómo hacerlo? Seguramente que el programa 2 le resultará de gran ayuda.

Programa dos

Para su realización hemos utilizado, como debía ser casi siempre, un diseño modular consistente en dividirlo en pequeñas rutinas o submódulos que realizan una acción muy concreta y que son llamadas desde un programa principal que controla el momento oportuno para ejecutar cada una de ellas. Dicho programa principal es el que está formado por las líneas comprendidas entre la 20 y la 60.

Es un simple bucle sin fin, ya que al final — línea 60 — le mandamos con un «**salto incondicional**» — o GOTO — de nuevo a la 30. Su misión es ejecutar primero la rutina de inicialización, que comienza en línea 260, y después repetir cíclicamente el resto de las mismas.

Las que más nos interesan son, la que genera las nuevas coordenadas, por ver trabajando a la función **INKEY**, y la que trata la nueva posición de **Don Sonrisitas** cuando llega a los bordes de la pantalla. Vayamos a su análisis.

La rutina que nos calcula las nuevas coordenadas es la que comienza en la línea 70 y llega hasta el RETURN de la 120. Su estructura es la misma que ya conocemos pero ahora en lugar de INKEY\$ hemos empleado la nueva INKEY.

Las condiciones que han de cumplirse para cambiar la posición del móvil es que al menos una de las teclas haya sido pulsada. O sea, que la función **INKEY** aplicada a los números asociados a las teclas elegidas nos devuelva un valor distinto de -1 por lo menos para alguno de ellos.

En el momento que esto ocurra, las líneas 80 a 110 se encargarán de hacer variar la coordenada correspondiente, y por tanto la nueva posición del objeto. ¿Está claro?

Pasemos ahora a ver qué pasa cuando nuestro amigo llega a los bordes de la pantalla. Si da unaojeada a la rutina de las líneas 130 a 180 comprenderá lo que le queremos decir.

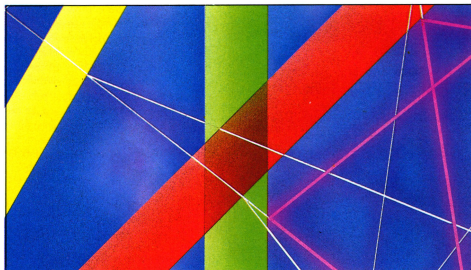
```
10 REM PROGRAMA 11
20 GOSUB 260
30 GOSUB 190
40 GOSUB 70
50 GOSUB 130
60 GOTO 70
70 REM NUEVAS COORDENADAS
80 IF INKEY(1) < -1 THEN x=x+1
90 IF INKEY(8) < -1 THEN x=x-1
100 IF INKEY(6) < -1 THEN y=y+1
110 IF INKEY(2) < -1 THEN y=y-1
120 RETURN
130 REM EXTREMOS
140 IF x=0 THEN x=40
150 IF x=41 THEN x=1
160 IF y=0 THEN y=25
170 IF y=26 THEN y=1
180 RETURN
190 REM IMPRIMIR Y BORRAR
200 LOCATE x,y
210 PRINT CHR$(224)
220 FOR i=1 TO 50:NEXT i
230 LOCATE x,y
240 PRINT " "
250 RETURN
260 REM INICIALIZACION
270 CLS
280 x=20
290 y=10
300 RETURN
```

Vemos que cuando el valor de una, o de las dos coordenadas se salen de los márgenes admitidos, el **Amstrad** mediante la ejecución de esta **pequeña subrutina** se encarga de dar a dicha coordenada el valor que le sitúa en la parte opuesta de la pantalla.

Pongamos un ejemplo. El nuevo valor obtenido para la coordenada «x» es cero. Esto quiere decir que nos hemos salido del margen superior de la pantalla y por tanto no es válido.

¿Y qué es lo que hacemos? Mire despacio la subrutina. La línea 160 nos dice que «**IF**» el valor de la variable «x» es igual a cero «**THEN**» almacenamos en dicha variable un 25.

Pero, ¿en qué lugar de la pantalla nos colocamos cuando la coordenada «y» de la pantalla de texto vale 25? La respuesta es bien sencilla: en la última línea. O sea, justamente en el lado opuesto al sitio donde nos encontramos. Es lo que pretendíamos, ¿no?



Las líneas 110, 150 y 170 hacen algo semejante pero para los otros bordes. Con ello Don Sonrisitas no encontrará obstáculos.

El resto del programa es, como ya hemos apuntado, semejante a lo ya visto. Por eso échele una ojeada e intente aclarar con él algún concepto que no le haya quedado suficientemente claro. Como siempre le proponemos que haga cuantas prácticas se le ocurran. No considere un fracaso que algo no funcione como usted desea. Analicelo, haga los cambios oportunos y continúe probando nuevas técnicas e instrucciones. Es la mejor manera de aprender.

Bueno, ahora nos surge una pregunta, ¿estas funciones que analizan si se ha pulsado o no una tecla sólo nos van a servir para hacer programas de movimiento?

Es evidente que no va a ser así. Son una herramienta muy poderosa a la que vamos a buscar múltiples aplicaciones.

Se imagine que una de sus aplicaciones pueda ser detener la ejecución de un programa hasta que pulsemos una tecla? Pues así es. El programa 3 es una muestra de lo que le estamos diciendo.

Programa tres

Consiste en un bucle WHILE... WEND simplemente. El único misterio está en la condición que le hemos puesto para que se ejecute o no.

Con:

WHILE INKEY\$ = ""

le estamos diciendo al ordenador que ejecute el cuerpo del bucle contenido entre WHILE y WEND mientras la función INKEY\$ nos esté devolviendo una cadena que no contenga ningún carácter, es decir, una cadena «vacía».

¿Recuerdan cuándo ocurría esto? Según la definición que dimos de esta función, dará como resultado "" cuando no pulsemos ningu-

na tecla. En el momento que lo hagamos, el valor generado será el carácter de la tecla pulsada.

Por eso nuestro bucle estará dando vueltas hasta que se nos ocurra presionar cualquier tecla. Con ello INKEY\$ creará una cadena distinta de la «vacía» y ya no se cumplirá la condición de entrada al bucle por lo que lo abandonaremos y continuaremos con el programa.

Dese cuenta de una cosa. En realidad la eje-

```
10 REM PROGRAMA III
20 PRINT "EL PROGRAMA SE DETENDRA EN UN BUCLE"
40 WHILE INKEY$=""
50 PRINT "PULSE CUALQUIER TECLA"
60 WEND
70 PRINT
80 PRINT "YA HA PULSADO UNA TECLA."
90 PRINT ""
```

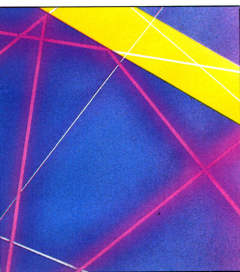
cución no se define sino que está girando dentro del bucle, gastando tiempo, «mientras» se cumpla la condición de entrada, que en este caso es no haber pulsado ninguna tecla.

También podríamos aplicar la función INKEY para realizar esta misión, pero ahora tendríamos que hacerlo con una tecla en concreto. Es lo mejor.

La forma de hacerlo es algo parecido a como lo hace el programa 4. Si lo teclea y ejecute verá cómo funciona.

Programa cuatro

En esencia funciona de una forma parecida a la del anterior, pero hay en realidad una gran diferencia: ahora sólo saldremos del bucle al pulsar la barra espaciadora. Con cualquier otra tecla no ocurrirá nada, solamente conseguiremos acabar la ejecución presionando la elegida. De ello se encarga la línea 40. (Recuerde que 47 es el número de tecla del «espacio», no el 55 como por error dijimos en el artículo anterior. Perdón por nuestro fallo.)



ma forma el programa pero dos veces consecutivos y en un espacio de tiempo corto.

Colocamos dos bucles WHILE... WEND con la misma condición que nos investiga si está pulsada o no la barra del espacio para salir de ellos. Se estaría ejecutando mientras:

```
INKEY(47)=-1
```

Parece que no habrá pegas.

Bueno. Escriba el programa 5 y ejecútelo a ver si ocurre así.

Programa cinco

Y resulta que se detiene en el primer bucle, pulsamos por primera vez el «espacio» e inesperadamente nos aparece un mensaje indicándonos que el programa ha terminado y el READY que nos lo confirma. ¡Arreal.

Aunque a priori no tendríamos problemas, parece que en la práctica si que los ha habido ya que el programa no funciona tal como nosotros pensamos.

¿A qué extraña circunstancia será debido este comportamiento?. Muy sencillo: el ordenador es mucho más rápido que nosotros y hace una exploración del teclado mientras todavía estamos pulsando una tecla.

Por eso, cuando llega el segundo bucle, la condición no se cumple y no entramos en él. La solución será retrasar la ejecución del programa justo en el momento que está entre ambos.

Y ¿qué mejor forma de hacerlo que mediante un bucle retardador, uno de esos que solamente gastan tiempo?. Añada al programa la línea:

```
60 FOR I=0 TO 500: NEXT I
```

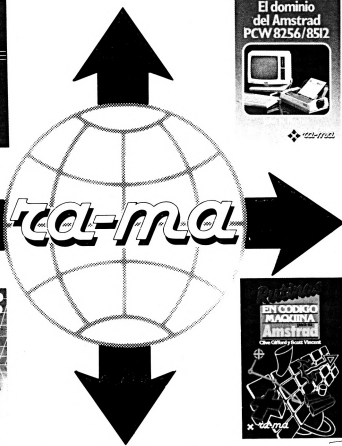
y verá cómo hemos solucionado el problema. Simplemente hemos retrasado la ejecución del segundo bucle. Es un método un poco rudimentario pero funciona.

```
10 REM PROGRAMA V
20 CLS
30 PRINT "PRIMERA PARADA"
40 WHILE INKEY(47)=-1
50 WEND
60 PRINT "SEGUNDA PARADA"
80 WHILE INKEY(47)=-1
90 WEND
100 PRINT "FIN DEL PROGRAMA"
```

Por el momento les dejamos. Intente pensar las múltiples aplicaciones que pueden tener estas funciones tanto para dar un movimiento más o menos perfecto a nuestro amigo Don Sonrisitas, como para cualquier otra cosa. Seguro que su imaginación puede descubrir aplicaciones insospechadas que le llenarán de asombro. «Hasta más ver.»

Sigamos con nuestros experimentos. Imagínemos que necesitamos detener de esta última

```
10 REM PROGRAMA IV
20 CLS
30 PRINT "EL PROGRAMA SE DETENDRA EN UN BUCLE"
40 WHILE INKEY(47)=-1
50 PRINT "PULSE ESPACIO"
60 WEND
70 PRINT
80 PRINT "YA HA PULSADO LA BARRA ES PACTADORA."
90 PRINT "BIEN"
```



Porque creemos en la Informática... Solo vendemos Informática.

TODO EN:

- Ordenadores Personales.
- Lenguajes de Programación.
- Técnicas de Programación.
- Telemática.
- Enseñanza asistida por Ordenador.
- Robótica, Sistemas Expertos, etc...

Suscribase a nuestro boletín bibliográfico RA-MA: microinformática (gratuito), editado mensualmente con las novedades aparecidas en el mercado nacional e internacional. Para recibirlo solo tiene que indicarlo en la casilla, cumplimentar y remitirnos el cupón que figura debajo.

NOMBRE
 DIRECCION
 CIUDAD
 CODIGO POSTAL
 PROVINCIA



Carretera de Cañillas, 144. 28043 MADRID
 Teléfono: 200 97 46 - 47

CREADOR DE VENTANAS

Miguel Angel Alonso



El fin de este programa es la creación de hasta siete ventanas mediante el comando WINDOW (la ventana 0 está reservada para enviar los mensajes del programa) sin necesidad de introducir las coordenadas que exige sino, simplemente colocando el cursor en el extremo superior derecho y en el inferior izquierdo y pulsar la tecla ENTER una vez fijada cada posición. Realizadas las operaciones anteriores se mostrará en pantalla la ventana creada y se preguntará a continuación si desea crear una nueva.

El programa se puede usar en los tres modos de pantalla para ello se solicita al principio del programa.

Para la comprensión de algunas partes del programa se recomienda la lectura de los programas GENPANT y BASCOM puesto que el movimiento del cursor se realiza de similar forma.

En la línea 10 se solicita el modo de pantalla, almacenándose en la variable t dicho valor.

En la línea 20 se fija el modo y se limpia la pantalla poniéndose a 1 el valor de i (usada para determinar

qué número de pantallas es) bifurcándose posteriormente a la línea 120.

En la línea 120 se hacen dos llamadas al firmware para habilitar y hacer visible el cursor, dimesionándose a continuación la variable que contendrá las posiciones del cursor.

En la línea 130 se ponen a 1 las variables de la fila y la columna. En la 140 se entra en un bucle para posicionar el cursor mientras no se pulse la tecla ENTER, bucle que termina en la línea 170.

En la línea 180 se incrementa el valor de la variable z, la cual se usa para determinar si es el primero o el segundo valor de la definición de la ventana. Si se trata del primer valor se almacenan la fila y la columna en las variables fila y columna volvién-

dose a la línea 130 y en caso contrario a la línea 190.

En la línea 190 se borra en primer lugar la última posición del cursor y a continuación mediante WINDOW se crea la pantalla definida así como el color del papel. En el caso de que el valor de i sea mayor que siete se termina el programa, pues es el número máximo de ventanas que se pueden crear y en caso contrario se dirige a la línea 200.

En la línea 200 se pregunta si se desea crear otra ventana y en caso de responder n/N se termina el programa y en el caso contrario s/S se incrementa el valor de i y se retorna a la línea 130. La línea 230 implica que en el caso de que la respuesta no sea ninguna de las anteriores se solicita de nuevo la respuesta.

```
10 INPUT "MODO DE PANTALLA: ";m:modi=
  IF m:modi<>0 AND m:modi<>1 AND m:modi<>2 THEN 10
  ELSE IF m:modi=0 THEN t=0.5 ELSE t=m:modi
20 MODE m:modi+1:GOTO 120
30 IF t=CHR$(13) THEN RETURN
40 l=ASC(f,c):t=ASC(t):PRINT t:col=c+1
50 IF col=240 THEN GOSUB 70
60 RETURN
70 col=1:IF f<24 THEN f=f+1
80 RETURN
90 f=f+1:(ASC(t)=240 AND f>1)-(A
  SC(t)=241 AND f<24)
100 col=c+(ASC(t)=242 AND col)=1-
  (ASC(t)=243 AND col=(t=240)+1)
110 RETURN
120 CALL &B7B:CALL &B8B:DIM l=12
  4,(t=240)
130 f=i:col=1
```

```
140 WHILE t=CHR$(13)
150 LOCATE c,f:ite=t:WHILE t=
  "":t=INKEY$:WEND
160 IF ASC(ite)=239 AND ASC(t)=24
  4 THEN GOSUB 90:GOSUB 50 ELSE GOSUB
  30
170 WEND
180 z=z+1:IF z=1 THEN fila=f:column
  a=c:ite="":GOTO 130
190 t="":LOCATE c,f:PRINT t:
  z=0:WINDOW #1,column,c,f:PA
  PER #1,3:CLS #1:IF i>7 THEN EN
  D
200 LOCATE 1,24:PRINT "OTRA VENTANA
  S/N ";
210 a=INKEY$:IF a="" THEN 210 ELSE
  IF a="N" OR a="n" THEN END
  220 IF a="S" OR a="s" THEN i=i+1:
  GOTO 130
230 GOTO 200
```

MASTER COMPUTER

Centro Comercial Guadalupe
Ctra. Canillas, 136-1.ª planta
Tel. 200 80 65 MADRID

Centro Comercial El Bulevar
La Moraleja
Tel. 654 16 12 MADRID

También abierto domingos de 10 a 2
Centro Comercial Ciudad Sto. Domingo
Ctra. de Burgos, km 28.
Tel.: (91) 622 12 89
Algete (Madrid) (Central)

MASTER HARD

Servicio Técnico
Para AMSTRAD
en Galicia,
León
y Asturias.
C/ Magdalena, 213
El Ferrol
Tel.: (981) 35 84 32

MICRO DEALER AMSTRAD CENTER

MAYORISTAS DE INFORMATICA
AMSTRAD PCW 8512
AMSTRAD PCW 8256
AMSTRAD CPC 6128 Fósforo verde y color
AMSTRAD CPC 472 Fósforo verde y color
Impresoras, Interfaces, Joysticks
Diskettes de 3 pulgadas, cable, etc.
Sinclair, Commodore, New Print,
Spectravideo, Compatibles IBM
C/ Comandante Zorita, 13. 28020 Madrid
Telfs. 233 07 81 - 233 07 35

ORDEMANIA SOFT

DE VENTAS
ESTADISTICA

por fin..

GESPACK

Paquete integrado de gestión que le permite a Vd., de forma fácil y optimizando el tiempo, la gestión global de su empresa.

Este paquete incorpora los programas de CONTABILIDAD, FACTURACION y CONTROL DE STOCKS además de un programa de CONTROL DE PEDIDOS y todo de forma interactiva. Controle su empresa por sólo **29.900 pts.**

Contabilidad

Contabilidad de fácil manejo y de gran potencia que permite trabajar con cuentas de hasta cuatro niveles, con capacidad según diskette de 500/1.000 cuentas y de 2.000/10.000 asientos.

Permite modificar o dar de baja apuntes ya integrados en el Mayor, programaciones de cierres, ficheros de Contabilidad y Cuenta de Explotación, ejecución de balances comparativos, reenlázándolos por meses, clave acceso restringido, etc...

19.900 ptas.

Facturación

Programa de gran agilidad y rapidez que incorpora el Control de Clientes, con gran capacidad de datos, Artículos, Albaranes, Facturas y recibos.

Generación automática y manual de documentos, valoración, a voluntad de los albaranes, todo tipo de listado, incluido el del IVA de las facturas emitidas para la declaración de Hacienda, etc...

15.500 ptas.

Control de stocks

Gran capacidad de datos, le permitirán a Vd. llevar con claridad y sencillez el control de su stock.

El programa le permite llevar un libro de entradas/salidas, reorganizarlo, hacer listados de stocks..., le avisará de los límites de stocks, mínimo y máximo por artículo, etc...

Todo para la llevanza de su almacén.

14.900 ptas.

disponibles para: PCW 8256
PCW 8512

Asimismo, Contabilidad disponible para CPC 664/6128 **9.900 ptas.**

OFITES INFORMATICA

DISTRIBUIDOR EXCLUSIVO DE ORDEMANIA SOFT

● IVA NO INCLUIDO

Condiciones especiales para distribuidores
Si tiene alguna dificultad en obtenerlos diríjase a

Ofites
Informática
Avda. Sabal 11, 16 - 8º
Tel. 453.444
Telex 3668
20011 SAN SEBASTIAN

SOFTWARE de muchos rombos, para mayores

**TOTALMENTE
EN
ESPAÑOL**

C Compilador C

Versión completa del famoso C-Hisoft para CP/M. Capacidades de E/S, ficheros aleatorios y modos de acceso binario y ASCII. Incluye editor ED 80 compatible WORDSTAR.

**15.000
ptas.**

PASCAL 80 Compilador Pascal

Especial para Z-80. Deja el programa fuente en un programa directamente ejecutable. Incluye ED 80, editor compatible con WORDSTAR.

**15.000
ptas.**

KNIFE Editor sectores

Permite trabajo directo sobre disco, bien en hexadecimal o ASCII, recuperar ficheros perdidos o borrados, alterar y/o proteger directorios, todo bajo AMSDOS y CP/M.

**7.900
ptas.**

DEVPAC 80 Ensamblador/des

ED 80: Editor Configurable GEN 80: Macros, inclusión en disco, ensamblador condicional, manipulación bit a bit. MON 80: Monitor y debugger, puntos de ruptura y presentación de memoria.

**15.000
ptas.**

MODULA-2 Comp. Modula -2

Implementación total del lenguaje MODULA-2 para CP/M. Compilador en un único paso, listo para ser linkado.

**19.900
ptas.**

TORCH Tutor de CP/M

Diseñado específicamente para AMSTRAD. Incluye THE WAND, creador de menús de programas.

**7.900
ptas.**

POLYPRINT Multitipos

Transforme su impresora en una imprenta. Permite la impresión en 8 tipos distintos de letras; configurable para cualquier impresora.

POLY TYPEFACES Multitipos

Añade a la potencia del programa POLYPRINT 8 juegos adicionales de impresión a los ya existentes.

WRITE HAND MAN Sidekick en CP/M

Residente en memoria, sin interferir en su programa principal le ofrece: Calculadora (Hex-Dec), Block de notas y teléfonos, Calendario, Directorios, etc...

**11.900
ptas.**

POLYPLOT Impresora/Plotter

Permite realizar gráficos sofisticados en su impresora. Gráficos de pastel, histogramas comparativos, gráficos de líneas, imágenes de 960 PIXELS de densidad.

***11.900
ptas.**

POLYMAIL Mailing

Sencillo sistema de MAIL-MERGE. Idóneo para producir circulares. Incluye editor. Permite la realización de etiquetas autoadhesivas.

***9.900
ptas.**

CATALOG Clasificador

Asigna a cada disco un número de serie y además indexa y cataloga los ficheros en ese disco.

**8.900
ptas.**

MULTI-TEXT Módulo de textos

Módulo de textos, preparado para ser empleado con nuestro lápiz óptico ESP o con las teclas de cursor.

***10.900
ptas.**

FIRST STEPS Tutor de Newword

Explore las enormes capacidades del procesador de textos NEWWORD; guiado desde los fundamentos del proceso de textos.

**7.000
ptas.**

MASTER LOCOSCRIPT

Dos cintas audio con instrucciones claras para aprendizaje y apoyo al manual del tratamiento de textos LOSOSCRIPT.

**3.000
ptas.**

DRAUGHTS- MAN II

Nueva versión mejorada y compatible con nuestra tableta GRAFPAD II: Gran capacidad en gráficos.

**6.200
ptas.**

TYPING CRASH COURSE Inicia a teclear

Curso de iniciación a los teclados, recomendado para personas no acostumbradas a su uso.

**9.900
ptas.**

TWO FINGERS Curso mecanográfico

Conozca a fondo las posibilidades del teclado, escribiendo con sus diez dedos en lugar de sólo dos.

**9.900
ptas.**

*** los 4 juntos
23.800 ptas.**

**IVA
no
incluido**



DE VENTA EN LOS MEJORES COMERCIOS DE INFORMATICA
Si Ud. tiene alguna dificultad para obtener los programas, puede dirigirse a:

**Ofites
Informática**

CONDICIONES ESPECIALES PARA DISTRIBUIDORES
EDITOR Y DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA

Avda. Isabel II, 16 - 8º
Tels. 455544 - 455533
Télex 36698
20011 SAN SEBASTIAN

FUTBOL



MEXICO 70



ALEMANIA 74



ARGENTINA 78



ESPAÑA 82



MEXICO 86

TOTAL

¿Alguna vez penso en
organizar su propio cam-
peonato de fútbol?
Ha llegado ese momento.
Invente un uniforme para sus jugadores, elija entre sus
equipos favoritos y reclute a los mejores futbolistas,
creando así un equipo campeón, dotado
con los mejores jugadores
de todos los tiempos...
"SU EQUIPO
SERA EL CAMPEON".

**PREPARESE
PARA EL 90**



DISTRIBUIDO en
Cataluña y Baleares por:
DISCLUB, S. A.
Balmes, 58
BARCELONA
Teléf. (93) 302 39 08

Fabricado y
distribuido bajo
licencia por:



COMPULOGICAL S.A.

Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMÚN** 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 COMÚN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A. AMSTRAD SEMANAL.**

Apartado de correos 54.062
28080 MADRID

¡ABSTENERSE PIRATAS!

Vendo 1.—Amstrad CPC-664 con pantalla color. 2.—Impresora **PRINTER-80 INDESCOMP** (adaptada a **Amstrad**) matricial, 40-80-142 col; 80 cps; 640 puntos gráficos. 3.—Juego regalo: **COMBAT LINX** y otros. Precio total: 160.000 ptas. Tel. (93) 788 07 53 (21,30-22,30).

Me gustaría en primer lugar contactar con usuarios de **CPC 464** para intercambiar ideas, juego, etc. (a ser posible de Avila aunque no necesariamente). En segundo lugar me gustaría obtener el **Exploding Fist**. Bien por dinero o bien por otros programas. También estoy interesado en: **Beach Head**, **Hypers Sports**, **Decathlon**, **Fighter Pilot**. Con las mismas condiciones que **Exploding Fist**. Estoy abierto a otras ofertas. Los interesados llamar al (918) 22 74 99 desde las 6 a las 6,30 o bien de 8,30 en adelante. Fines de semana a cualquier hora.

Vendo Amstrad CPC-464, monitor color en perfecto estado, incluyendo lote de 50 programas a escoger entre más de 100: **DEVPAC/AMSWORD II/PASCAL/AMSBASE SCREEN DESIGNER/AMSCALC...**, últimos juegos, 7 libros. Todo por 85.000 ptas. Más información: **Eliseo González Real** c/ Pereio, 4. Viveda (Cantabria). Tel.: (942) 88 48 24.

Urge vender Spectrum Plus 48K. (7 meses de garantía). Fuente de alimentación, **Interface Kempston**, 4 libros de programación, más de 30 programas (**exploding Fist**, **Contabilidad**, etc.) 24.000 ptas. Interesados: **Alferez Monedero**. Academia G.^a Civil. Aranjuez. Tel. 891 21 45 (91); de 17,30 a 19 horas.

Vendo programa de contabilidad igual que el que viene en la revista. Precio razonable. Interesados llamar al Tel. (91) 207 81 89. Preguntar por Pedro.

Vendo consola de videojuegos Atari-2.600, un joystick, 5 cartuchos, **Defender**, **Berzeck**, **dog'em**, **Space Invaders**, y **Missile Comand**. Más 2 mangos de paleta, más instrucciones en castellano, cables, etc.; todo ello casi nuevo y en perfecto estado de funcionamiento. Además 5 revistas de **Microhobby (Spectrum)**. Todo ello por 19.500 ptas. (negociables). Interesados llamar al tel.: 23 51 14 de Murcia. Preguntar por **Javier** (llamar a cualquier hora de la tarde de los sábados).

Vendo Amstrad CPC 464 color. Programas utilidades (**Ensamblador**, **desensamblador**, **copión**), juegos muy buenos (**ajedrez**, **marcianos**, **codename mat**, etc.), manual del **firmaware**, **Pascal (Hisoft Pascal)**, tratamiento de texto (**Amsword**), programas de dibujo (**CAD/CAM**, **artísticos**, etc.)... 60.000 ptas. Noches. **Victor**, Tel. (93) 210 23 69.

Estoy interesado en comprar un monitor color, o cambiarlo por uno **GT/65** con garantía **Indescomp**, comprado hace 2 meses, pagando la diferencia justa. También cambio juegos, programas, utilidades, etc., tengo unos 100: mandaré lista. Interesados dirigirse a: Urb. Bahía de Algeciras Bloque 2C-2: 2.º C. Algeciras (Cádiz)... **Manolo Benitez**.

Deseo contactar con usuarios de **Amstrad**, de **Gandia** y alrededores, para ofrecerme como profesor de programación (200 ptas/h). O para intercambiar listados e ideas. Abstenerse los maniáticos de los juegos. **Santiago Torrego Calabuig**. c/ Abad Sola, 92, **Gandia** (Valencia).

Desearía contactar con usuarios 464 para cambio de utilidades y programas. Interesados escribir a: **Juan Carlos Martínez Lizán**. C/ **Pedro Díaz**, 12, 1.º A. **Cartagena** (Murcia).

Vendo o cambio programas de Amstrad. Usuarios de toda España interesados, escribir a **Carlos Ribó Vilaseca**. Urbanización **Mirasol**, 1, **Solana** (Lérida) o llamar al tel. (93) 811 28 35 para llamadas de fuera de la provincia de Barcelona. Llamar de 1,30 a 3,00 y de 8 a 9,30.

Vendo Amstrad CPC 464 con monitor en fósforo verde. Manual y 100 programas (juegos, aplicaciones...) incluidos. Interesados llamar al (91) 260 79 94. Preguntar por **Juan Carlos** de 3 a 5.

Cambio juegos excelentes de todo tipo (**knigh Lore**, **Alien 8**, **G. P. Rally II**, **Exploding Fist**, etc.) También utilidades. Escribir a **Jesús Alonso**, c/ **Ruiz 9** 4.º C, **Madrid 28004**. También llamar desde las 22,00 al 448 98 05.

CAULDRON • II



El Hechizo

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

SINCLAIR STORE

REGALO SEGURO



POR LA COMPRA DE TU ORDENADOR ESTE EQUIPO COMPLETO DE BASKET ES TUYO. TAMBIEN REGALO DE INTERFACE 2 Y UN JOYSTICK AL COMPRAR TU SPECTRUM 128 O PLUS

- Como siempre curso gratis de informática.
- Spectrum, Q.L., Commodore, Amstrad, Spectravideo y MSX.
- Teclado multifunción con sonido, **13.200 ptas.**
- Joystick + Interface + Kempston, **3.200 ptas.**
- Lápiz Óptico, **3.500 ptas.**
- Bibliografía 25% Dto.
- Tarjeta de Socio Club Sinclair Store.
- Servicio Técnico de Reparaciones, **3.700 ptas.**
- Necesitamos distribuidores, somos mayoristas.
- Ampliación de memoria Amstrad 464 ó 6128 256 K, **20.900 ptas.**
- Disco de Silicio, **19.900 ptas.**
- Por la compra de un spectravideo, 328, **25.000 ptas.** o spectravideo 728 MSX, **39.000 ptas.** Te regalamos un Joystick Quickshot I más 10 cintas y un cassette.

Amstrad 8512: 169.900 (IVA incluido). Amstrad 8256: 129.900 (IVA incluido). Amstrad 6128: 84.900 (IVA incluido).

Liquidación Software: West bank, Gremlins, Ole Toro y 200 títulos más, desde 990 ptas.

sinclair store

SOMOS PROFESIONALES

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

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

AV. FELIPE II, 12
(Metro Goya)
Tel. 431 32 33 - 28009 MADRID
Aparcamiento GRATUITO Av. Felipe II