

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 83

190 Ptas.

PERSONALIZAR LOS PROGRAMAS DE JUEGOS ES FÁCIL CON

ESPIA DE GRÁFICOS

Esta aplicación permite encontrar y modificar a nuestro gusto, permanentemente, los gráficos de todos los juegos comerciales cuya protección lo permita.



TRUCOS

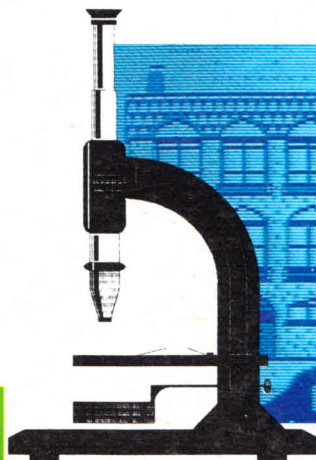
Hágase su propia versión del "Arkanoid"

Ahorrar memoria en el disco



LASERBASE
Batalla a muerte en el sistema solar

HEXAM
Probamos un ensamblador que no ocupa memoria



DISEÑO ASISTIDO EN EL AMSTRAD PC:

LOS SECRETOS DE GEM DRAW

CATÁLOGO DE SOFTWARE

Y LISTÍN TELEFÓNICO

PARA EL PCW

Una cinta gratis por la compra de números atrasados



Con la compra de siete números atrasados recibirás gratis una cinta de programas de **YOUR COMPUTER** (el mejor software inglés), totalmente gratis.

Aprovecha la oferta y consigue tu colección

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.^a Díaz

Redactor Jefe

Juan José Martínez

Diseño y maquetación

Rosa María Capitel,

Valeriano Cenalmor

Redacción

Eduardo Ruiz de Velasco

y Carmen Elías

Colaboradores

Javier Barceló, David Sopena,

Robert Chatwin, Antonio Cuadra,

Pedro Sudón, Miguel Sepúlveda,

Francisco Martín, Jesús Alonso,

Pedro S. Pérez, Amalio Gómez,

Alberto Suñer

Secretaría Redacción

Marisa Cogorro

Fotografía

Carlos Candel

Chema Sacristán

Miguel Lamana

Ilustradores

J. Igual, M. Barco, J. Siemens,

F. L. Frontán, Pejo,

Edita

HOBBY PRESS, S.A.

Presidente

María Andriano

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Jefe de Publicidad

Miguel Bendito

Jefe de Administración

Raquel Jiménez

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid

Pedidos y suscripciones:

734 65 00

Redacción: 734 70 12

Dto. Circulación

Paulino Blanco

Dto. Marketing

Emilio Juarez

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDIC, S. A. Crta. de

Irún. Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos de la revista

COMPUTING with the AMSTRAD

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

AMSTRAD Semanal no se hace necesariamente solidaria de las opiniones vertidas por sus colaboradores en los artículos firmados. Reservados todos los derechos.

EDITORIAL

Los videojuegos para ordenador, al menos en nuestro país, poseen un carácter marcadamente gráfico; es decir, el arma usada por los programadores para atraer al público es conjuntar la acción con la estética. **AMSTRAD Semanal** ha pensado muchas veces lo interesante que sería poder personalizar los juegos, esto es, inspeccionar el programa, encontrar los gráficos, darles la forma que más nos gustase y volver a grabar el juego así modificado en cinta/disco. La próxima vez que jugáramos a él, los cambios se reflejarían como por arte de magia. En este sentido, hemos escrito un programa, llamado «Espía de Gráficos», que permite hacer esto mismo, con el auxilio de otras herramientas ya publicadas en nuestra revista. Seguro que se van a divertir mucho jugando con este programa.

Resulta obvio decir que el «Espía...» está escrito en Código Máquina (no quedaba otra elección), y que para diseñarlo se ha tenido que usar un ensamblador. Estas herramientas son un tanto desconocidas en España, hay muy pocas comercializadas. Para los interesados en el tema, probamos a fondo en este número uno llamado «Hexam» que, aparte de ser muy potente, presenta la particularidad de estar en un cartucho de ROM, esto es, se conecta al bus de expansión del ordenador y no gasta un byte de memoria. Las ventajas de este planteamiento son innumerables: nosotros las decimos todas.

Por último, ocupa un lugar de honor esta semana la prueba de un gran programa de diseño para el Amstrad PC, llamado «Gem Draw». Podríamos decir mucho de su facilidad de uso, de los magníficos dibujos que crea y de lo perfectos que salen en la impresora, pero lo mejor es remitirlos al artículo en cuestión, que lo estudia detenidamente.

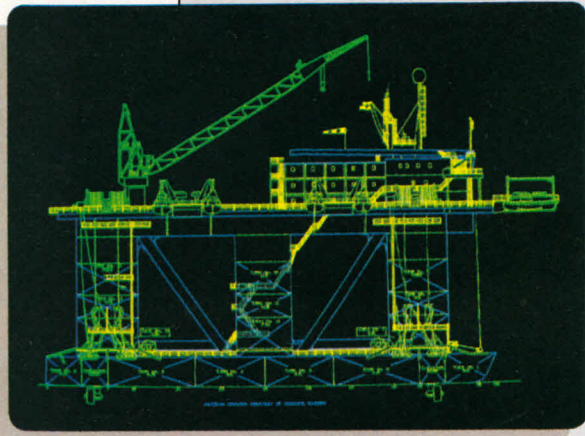
Los «jugones» encontrarán un original y adictivo programa para los CPCs, el «Laserbase», que les transportará a la velocidad de la luz al cruento escenario de la batalla espacial más decisiva del milenio.

SUMARIO	4 Hoy por Hoy.	35 Catálogo de Software.
	6 Lenguajes: Tipos de datos en Pascal.	38 Mundo del PCW: Listín Telefónico (y II).
	12 Banco de Pruebas: Hexam.	42 Sin Duda Alguna.
	16 Mercado Común.	44 Infobytes.
	20 Trucos.	46 Serie Oro: La aventura de Laserbas.
	24 Mundo del PC: Gem Draw.	

CAD-CAM. La historia continúa

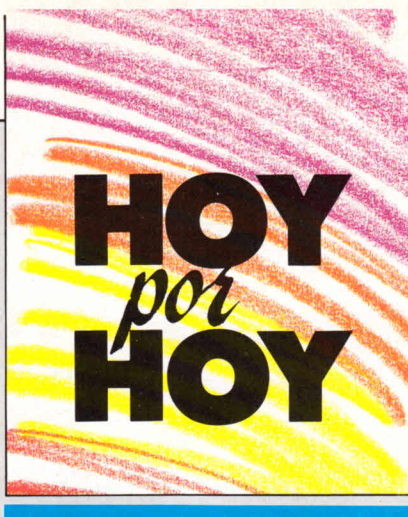
Pepper Pro 1280 es un nuevo sistema gráfico para AutoCAD que actúa como el eje principal de estaciones de trabajo CAD.

Está distribuido en España por Softronic S. A., la misma



empresa que importa para España AutoCAD. La nueva tarjeta, con un alto nivel de resolución y 32 bits, puede convertir a un PC en una estación de trabajo MIPS. Actúa de una forma instantánea, nítida y precisa. Para conseguir esta rapidez, el zoom, el encuadre y el scroll se hacen por medio del hardware.

Todo esto, unido a su gran resolución (1.280 x 1.024), compatibilidad (emula a los estándar CGA, MDA y PGC), colores simultáneos (16 ó 256) y la gran variedad de su paleta de colores hacen de él uno de los sistemas gráficos más compatibles que existen.



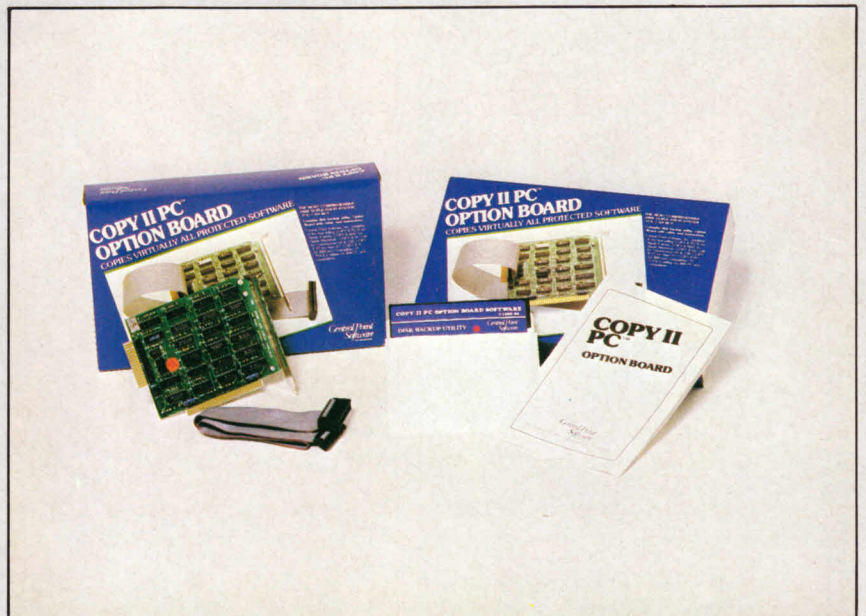
Copión para PC

First, S. A., ha comenzado a distribuir en España la tarjeta *Copy II PC Opcion Board*, de la casa estadounidense Central Point Software. Tiene un programa especial, que forma parte de la nueva generación de Transition Copier, que copia las

transiciones magnéticas de un diskette a otro sin mirar nada en particular. Es decir, el sistema de copia más poderoso que existe actualmente, ya que no trabaja con las limitaciones que siempre producen la circuitería estándar de los discos. Reproduce el 98 por 100, sin necesidad de especificar ningún parámetro, aunque también se puede optar por uno de ellos.

Su instalación y manejo es muy sencillo. Funciona a través de menús. Una de las opciones que incorpora es *Transition Editor* con la que se puede ver el formato físico del disco original, editar las pistas y visualizar los datos en Hexadecimal o ASCII.

Si todo esto es cierto, lo que parece claro es que este copión va a tener un gran éxito.



Backup para el disco duro de Amstrad

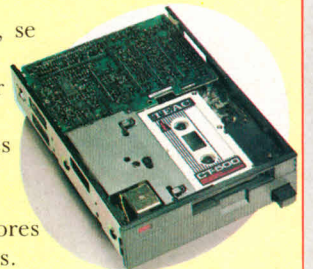
Tekdata ha sacado un nuevo periférico, llamado *Ex-Streamer*, con el que se pueden hacer copias del disco duro del Amstrad PC 1512 o del de cualquier otro compatible IBM.

Su capacidad es de 22 Mbytes y, por 499 libras, incluye una tarjeta de control que se introduce en la unidad central del ordenador con el software y cables necesarios para su funcionamiento.

Además, con una tarjeta

opcional, se pueden compartir sus utilidades entre varios ordenadores parecidos.

Parecidas características se pueden aplicar al Backup para disco duro de IBM y compatibles que acaba de sacar la casa Teac Corporation.



First se sube al carro de los compatibles

Dunross PC-XT 640 Turbo es el nombre nuevo compatible que va a distribuir en España First, S. A.

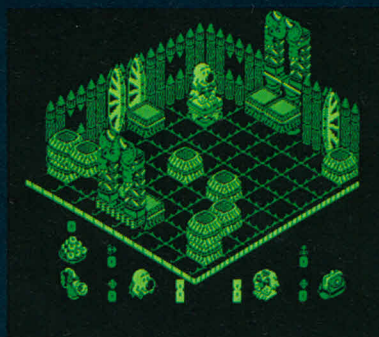
Lleva un microprocesador 8088-2 de Intel, 640 K de memoria RAM, que se pueden ampliar a 3 Mbytes, y un ventilador para evitar los problemas de sobrecalentamiento que pudiesen surgir. Se puede elegir entre versiones de una o dos unidades de disco de 5 1/4" y versiones de disco duro de 10, 20 y 33 Mbytes, todas ellas con teclado en castellano. Su capacidad es de 80/40 x 25 caracteres en modo texto, y 640 x 200 en modo gráfico con salida paralela (centronic) para impresora.

Según First, este compatible trabaja un 70 por 100 más rápido de lo normal en los PCs, debido al modo turbo de actuación. Su precio es de 115.000 pts., en configuración básica: 1 floppy.

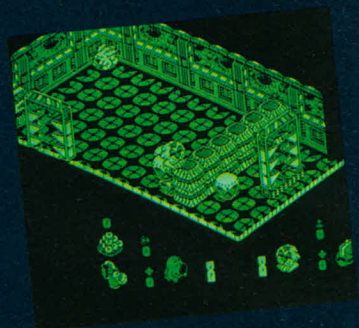


Head over Heels. Tras las huellas de Batman

Los PCW son los Amstrad más abandonados en cuanto a juegos. Por eso, siempre es de agradecer cualquier novedad en este sentido.

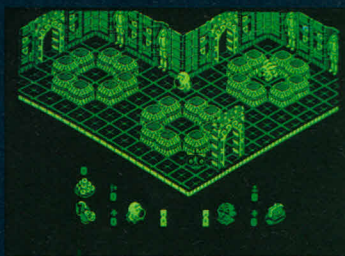


Y novedad es el juego del que vamos a hablar: *Head over Heels*. Este programa, distribuido en España por Erbe Software, es lo último que ha salido al mercado para el PCW, aunque también



existe una versión para los CPCs. Está en la misma línea que *Batman*, pero, según todos los indicios, le supera en casi todos los aspectos.

Es un juego de acción y estrategia con gráficos tridimensionales. Pero, de todas estas características, hablaremos con mayor detenimiento en nuestras páginas de juegos. Todavía no podemos adelantar su precio.



Nueva hoja de cálculo para Amstrad PC

Qcalc es la nueva hoja de cálculo para el Amstrad PC 1512 y todos los compatibles IBM que tengan 256 K, ya que, aunque también funciona en los de 128 K, al programa le queda poca memoria libre para poder trabajar bien. Lo acaba de sacar a la venta la empresa Proteus Compting, con un precio dos veces más barato que cualquiera de las existentes con

sus mismas características.

Este programa puede satisfacer las necesidades de todo tipo de usuarios, desde principiantes hasta profesionales, debido a su capacidad de 256 columnas por 9.999 líneas y un contexto sensible que ayuda a los noveles a introducirse en las diferentes pantallas.

Incorpora 25 funciones

aritméticas, tres lógicas (If, Then y Else) y cinco de tratamiento de listas numéricas (Sum, Maximum, Minimum, Average y Count).

Dentro de poco estará disponible una versión para usar en la red Novell.

Si desean más información, pueden escribir a Craig Thatcher, 2 Iffley Road, Hammersmith, London W6 OPA.

Tipos de datos en Pascal

Uno de los más fuertes rivales del Basic en el mundo de los lenguajes de programación es, sin duda, el Pascal. La criatura de Niklaus Wirth reivindica en su favor poseer una organización y una estructura impecables.

Cuando uno está diseñando un programa a medida, bien para un hipotético cliente o para uno mismo, observa lleno de admiración que le faltan datos, es decir, que si se trata de manejar apuntes contables, por ejemplo, se ve obligado a tratarlos de la misma forma que si estuviera manipulando frases, palabras o números.

Casualmente, al menos de cara al programador, todo lo que un ordenador tiene son eso, caracteres y números, al menos en Basic. Ahí está el quid de la cuestión: el asunto cambia completamente cuando se trata de otros lenguajes, como el Pascal.

Cualquier lenguaje de programación que pretenda hacer la vida del sufrido operador un poco más fácil, debe poseer dos cosas: estructura y posibilidad de definir tipos de datos más y más complejos partiendo de unos pocos que podríamos llamar *ladrillos elementales*.

La primera condición no es que el Pascal simplemente la cumpla, sino que es *el* lenguaje estructurado. Estas palabras nacieron con él.

La segunda condición es algo más esotérica, pero creemos que su necesidad quedará patente con el programa ejemplo de este artículo.

Imagine que usted desee mecanizar la gestión de una pequeña biblioteca informática personal. Por razones obvias, los datos que se necesitan manipular no son ni palabras ni números, sino programas.

Desgraciadamente, este tipo de dato no existe, y así, a *bote pronto*, su ordenador no sabrá de qué le habla.

Si el programa que mantiene nuestra biblioteca informática hubiera que escribirlo en Basic estándar (Locomotive Basic, Mallard Basic o IBM Basic), no habría forma de pasar de aquí. Usted tendría que usar lo que el ordenador tiene, y arreglárselas como Dios le dé a entender, probablemente definiendo montones de matrices para el nombre del programa, su precio, la versión, etc.

Si se quiere, este enfoque tiene la ventaja de la sencillez, pero el gran inconveniente de la poca claridad y el verse obligado a tratar algo de forma antinatural. Si manejamos información acerca de programas, lo lógico es tratarla como un todo, a pesar de que dicha información se

subdivida en categorías más pequeñas.

Afortunadamente, lenguajes más modernos que Basic, como Pascal y C, tienen resuelto este problema. Veamos cómo se hace en Pascal. Se



supone que el lector posee unos mínimos conocimientos del mismo.

Un programa en Pascal posee una estructura absolutamente rígida: primero se declara el nombre del programa; luego, las constantes, los tipos de datos, las variables, los procedimientos y funciones, y, por último, el cuerpo principal de la aplicación. Una vez anotado este detalle, vamos a plantearnos seriamente el problema de gestionar nuestra biblioteca de software.

La entidad que nos vamos a ver obligados a manipular vamos a llamarla *programa*. Y, para que signifique algo en concreto, analicemos de qué otras informaciones se va a componer el objeto *programa*.

En nuestra opinión, sobre él debemos conocer lo siguiente:

- título,
- tipo,

LENGUAJES

- versión,
- sistema operativo,
- precio.

Una vez definida nuestra información, el próximo paso es dar forma a cada uno de los campos, de tal modo que el ordenador sea capaz de entenderlos. Veamos. Evidentemente, el título tiene que ser una cadena de caracteres, una palabra. Por ello, y haciendo uso de un tipo de dato predefinido en Pascal, podemos declararlo como un array de caracteres de, digamos, 10 de longitud. La cosa sería, en principio, así:

```
titulo : array[1..10] of char;
```

La definición es correcta, pero se le pueden poner varias objeciones de estilo. En primer lugar, la declaración pertenece a una variable, es decir, la sintaxis correcta en Pascal es:

```
var titulo : array[1..10] of char;(1)
```

Cada vez que una función o procedimiento emplee una variable de ese tipo, habría que volver a declararla exactamente igual. Además del tedio que eso supone, no todos los compiladores lo aceptarían. Algunos sólo admiten como parámetros variables que pertenecen a un determinado tipo de datos. Y aquí está la clave. Si declaramos nuestro *titulo* de la siguiente manera:

```
type  
  titulo_c = array[1..10] of char;(2)
```

podremos escribir acto seguido:

```
var titulo : titulo_c;
```

y el programa ya sabe de qué le estamos hablando. Acabamos de inventarnos una estructura de datos nueva, correspondiente a una matriz de 10 caracteres de longitud, que será tratada exactamente igual que los tipos predefinidos carácter o número. Y, además, podemos declarar cuantas variables o estructuras queramos, por complejas que sean, pertenecientes al nuevo tipo.

Esta manera de ver las cosas resuelve una serie de problemas graves.

Imagínese que usted ha escrito cuarenta funciones que manejan a *titulo*, tal y como lo ha declarado en (1). Siga imaginando: de pronto descubre que la matriz de 1 a 10 es demasiado corta, tiene que ser hasta 20. Bien, pues tendría que cambiar una por una todas las declaraciones. Si se ha seguido la práctica normal en Pascal de compilar por trozos, tendría que corregir el código fuente y recompilar de nuevo.

Sin embargo, tal y como está declarado en (2), dentro de la sección *type* del programa bastaría con modificar eso, poniendo simplemente:

```
type  
  titulo_c = array[1..20] of char;(3)
```



De este modo, todo el trabajo que hizo con las funciones anteriormente quedaría intocado. Ésa es la gran ventaja de definir los propios tipos de datos.

Todavía se puede perfeccionar un poco más la declaración. En efecto, las funciones que manipulen a *título* probablemente tendrán que llevar la cuenta de los caracteres que hay metidos en cada variable. Por ejemplo, si se desea imprimirlo, hay que hacerlo carácter a carácter hasta que lleguemos al final. Si nos pasamos, podría ocurrir un desastre. Tendrá que hacer comprobaciones constantes del tipo:

if longitud > 20 then...

Bien, pero ¿qué ocurre si le interesa de nuevo cambiar la longitud de *título* a 30? Pues que otra vez tendrá que corregir y recompilar a lo largo de todo el programa. Para evitarlo, la solución es definir la longitud máxima como una constante, así:

```
const
  MAXIMO__C = 30;
```

y

```
type
  título__c = array[1..MAXIMO__C] of char;
```

En caso de cambios, sólo habrá que modificar

el valor de la constante, y el resto del código del programa permanecerá inalterado, es decir, siempre podremos escribir:

if longitud > MAXIMO__C then...

Por último, téngase presente que estamos hablando, al fin y al cabo, de una cadena alfanumérica, las variables “\$” del Basic, extremadamente común, por lo que algunos compiladores de Pascal tienen el tipo *string* (cadena) ya predefinido. ¿Algún problema? Para nosotros, ninguno. Bastaría con escribir:

```
type
  título__c = string[MAXIMO__C];
```

y nuestro programa seguiría funcionando *tan ancho*. De esta forma, gracias a las declaraciones de tipo, la portabilidad (y rentabilidad) de nuestro código fuente aumenta muchísimo.

Todo lo dicho anteriormente puede aplicarse a lo que comentemos a partir de ahora. Vamos a estudiar con algo de detenimiento otro campo interesante: el tipo de programa.

Pascal tiene, para estos casos, una facilidad única, que se conoce como *tipo enumerado*. Fíjese en la declaración:

```
type
  clase__programa = (GESTION, JUEGO, UTILIDAD);
```

```
program type_demo (input, output);
const
  MAXIMO_C = 20;{LONGITUD MAXIMA DE LAS CADENAS}
  MAXIMO_E = 100;{NUMERO MAXIMO DE PROGRAMAS EN LA LIBRERIA}
type
  clase_programa = (GESTION, JUEGO, UTILIDAD, OTRO);
  {TIPOS DE PROGRAMAS PREVISTOS.ES POSIBLE ANADIR OTROS}

  título_c = array[1..MAXIMO_C] of char;
  {TIPO DE DATOS PARA EL TITULO DE LOS PROGRAMAS}

  comandos = array[1..1] of char;
  {TIPO DE DATOS PARA LAS ORDENES QUE ACEPTARA EL PROGRAMA}

  programa = record
    título : título_c;
    tipo : clase_programa;
    version, dos : real;
    precio : integer;
    usado : boolean;
  end;
  {RECORD QUE ALMACENA LA INFORMACION RELEVANTE DE CADA PROGRAMA}

  biblio = array[1..MAXIMO_E] of programa;
  {MATRIZ CON ESPACIO PARA 100 PROGRAMAS}

  contador = 0..MAXIMO_E;
  {TIPO DE DATOS PARA LAS VARIABLES 'CONTADORAS'}

var
  libreria : biblio;
  orden : comandos;
  lleno : boolean;
  {FLAG QUE INDICA SI QUEDA ESPACIO PARA MAS PROGRAMAS}
```

LENGUAJES

Así de claro y fácil. Le estamos diciendo al ordenador que éste es un tipo de dato que puede tomar esos tres valores, con lo que la legibilidad de nuestro programa aumenta en gran medida. Si, una vez más, se nos ocurre que nos hemos dejado en el tintero la posibilidad de añadir una clase genérica de varios a nuestro tipo de programas, nada más fácil:

```
type
  clase_programa = (GESTION, JUEGO,
  UTILIDAD, OTRO);
```

y eso es todo. Las variables posteriormente declaradas como pertenecientes a este tipo de datos incorporarán la modificación. Sin más cambios.

El resto de los campos de nuestro tipo de programa son más de sentido común. Así, en versión y sistema operativo, habrá que teclear cosas como 2.5 y 3.0. La solución más evidente es declararlos de tipo *real*, uno de los predefinidos en Pascal. En cuanto al precio, no parece probable que sea 3.250,25 pesetas, por lo que declararlo como *integer* debiera bastar.

A estas alturas, ya tenemos clara toda la información relevante del tipo de datos programa, pero todavía no la hemos organizado en un todo coherente. Esto se hace mediante la sentencia Pascal *record*:

type

```
programa = record
  titulo : titulo_c;
  tipo : clase_programa;
  version, dos : real;
  precio : integer;
  usado : boolean;
end;
```

Ya está. Hemos creado una entidad llamada *programa*, con unas ciertas características, que vamos a poder manipular de una vez, como si fuera un número o un carácter. El lector observará que aparece un campo más, *usado*, de tipo *boolean*, es decir, cierto o falso. Su utilidad se verá más tarde, y su existencia se debe a la lógica elegida para gestionar nuestra biblioteca.

Bueno, hasta ahora sólo tenemos definido un programa. ¿Cómo nos las arreglaríamos para poder trabajar con 100, por ejemplo?

De nuevo recurrimos a otro tipo de datos:

type

```
biblio = array[1..MAXIMO_E] of
programa;
```

```
procedure lee_programa (i : contador);
var
  eleccion : 1..4;
begin {lee_programa}
  writeln('TITULO: ');
  readln(libreria[i].titulo);
  writeln('VERSION: ');
  readln(libreria[i].version);
  writeln('SISTEMA OPERATIVO: ');
  readln(libreria[i].dos);
  writeln('PRECIO: ');
  readln(libreria[i].precio);
  repeat
    writeln('TIPO DE PROGRAMA: ');
    writeln('GESTION(1) JUEGO(2) UTILIDAD(3) OTRO(4)');
    readln(eleccion);
  until (eleccion >= 1) and (eleccion <= 4);
  case eleccion of
    1 :
      libreria[i].tipo := GESTION;
    2 :
      libreria[i].tipo := JUEGO;
    3 :
      libreria[i].tipo := UTILIDAD;
    4 :
      libreria[i].tipo := OTRO;
  end; {case}
  libreria[i].usado := TRUE;
end; {lee_programa}
```

```
procedure nuevo;
var
  i : contador;
```

```
begin {nuevo}
  writeln('NUEVO');
  i := 1;
  while (libreria[i].usado) and (i < MAXIMO_E) do
    i := i + 1;
  if (i = MAXIMO_E) then
    lleno := TRUE
  else
    lee_programa(i);
  end; {nuevo}
```

```
procedure eliminar;
var
  i : contador;
begin {eliminar}
  writeln('ELIMINAR QUE:');
  readln(i);
  if (i < 1) or (i > MAXIMO_E) then
    writeln('INEXISTENTE')
  else
    libreria[i].usado := FALSE;
  end; {eliminar}
```

```
procedure lista_programa (i : contador);
begin {lista_programa}
  writeln(libreria[i].titulo);
  writeln(libreria[i].version);
  writeln(libreria[i].dos);
  writeln(libreria[i].precio);
  case libreria[i].tipo of
    GESTION :
      writeln('GESTION');
    JUEGO :
      writeln('JUEGO');
```

en donde `MAXIMO_E` es una constante que vale 100, puesta por las mismas razones de flexibilidad que comentamos anteriormente.

¿Qué hemos hecho realmente? Pues se ha creado una matriz de `MAXIMO_E` elementos, pero cada uno de ellos es del tipo programa; es decir, tenemos espacio reservado para `MAXIMO_E` programas, cada uno con la estructura definida para él. Nuestra biblioteca informática personal ya está implementada.

Quedan, antes de ver los procedimientos que la gestionan, sólo dos tipos de datos más, dependientes de la lógica elegida. Nos vamos a mover tecleando comandos de una sola letra; por ello, definimos:

```
type
  comandos = array[1..1] of char;
```

y, como previsiblemente habrá que usar variables contadoras:

```
type
  contador = 0..MAXIMO_E;
```

ya que nunca vamos a contar más allá del número de elementos del array de programas.

Ya podemos declarar las variables globales:

```
var
  libreria: biblio;
  orden: comandos;
  lleno: boolean;
```

Es todo lo que necesitamos para poner en marcha el programa. Vamos a echar un ligero vistazo al mismo, empezando por la parte principal, el *manager*.

```
begin {PROGRAMA PRINCIPAL}
  lleno := FALSE;
  write('QUE HAGO AHORA?:');
  readln(orden);
  while(orden [1] <> 's') and (not lleno) do
  begin
    procesa(orden);
    write('QUE HAGO AHORA?:');
    readln(orden);
  end;
  if lleno then
    writeln('NO TENGO SITIO PARA MAS PROGRAMAS');
  writeln('FIN DEL PROGRAMA');
end. {PROGRAMA PRINCIPAL}
```

La lógica no puede ser más sencilla: la variable *lleno* nos va a decir cuándo no hay sitio para guardar más programas, por lo que la inicializamos a `FALSE`. A continuación, se imprime un mensaje y se lee la orden. Si es "s", o si no queda sitio para más programas, el *manager* termina imprimiendo el mensaje adecuado. De lo contrario, se procesa la orden con el procedimiento:

```
UTILIDAD :
  writeln('UTILIDAD');
OTRO :
  writeln('OTRO');
end; {case}
writeln;
end; {lista_programa}
```

```
procedure listar;
var
  i : contador;
begin {listar}
  for i := 1 to MAXIMO_E do
  begin
    if libreria[i].usado then
      lista_programa(i);
    end;
  end; {listar}
```

```
procedure buscar;
var
  i : contador;
begin {buscar}
  writeln('BUSCAR QUE:');
  readln(i);
  if (i < 1) or (i > MAXIMO_E) then
    writeln('FUERA DE RANGO')
  else if libreria[i].usado then
    lista_programa(i)
  else
    writeln('INEXISTENTE');
  end; {buscar}
```

```
procedure procesa (orden : comandos);
begin {procesa}
  case orden[1] of
    'b' :
      buscar;
    'n' :
      nuevo;
    'e' :
      eliminar;
    'l' :
      listar;
    else
      writeln('ERROR.COMANDO DESCONOCIDO');
  end; {case}
end; {procesa}
```

```
begin {PROGRAMA PRINCIPAL}
  showtext;
  lleno := FALSE;
  write('QUE HAGO AHORA?: ');
  readln(orden);
  while (orden[1] <> 's') and (not lleno) do
  begin
    procesa(orden);
    write('QUE HAGO AHORA?: ');
    readln(orden);
  end;
  if lleno then
    writeln('NO TENGO SITIO PARA MAS PROGRAMAS');
  writeln('FIN DEL PROGRAMA');
end. {PROGRAMA PRINCIPAL}
```



procedure procesa (orden : comandos);

```
begin{procesa
  case orden[1] of
    'b':
      buscar;
    'n':
      nuevo;
    'e':
      eliminar;
    'l':
      listar;
  else
    writeln('ERROR: COMANDO
    DESCONOCIDO');
  end; {case}
end; {procesa}
```

Este procedimiento también está claro. Bifurca a otro que se encarga de cada comando o da un mensaje de error si éste es desconocido.

Veamos cómo funciona el procedimiento buscar:

```
procedure buscar;
  var
    i: contador;
begin {buscar}
  writeln('BUSCAR QUE:');
  readln(i);
  if (i < 1) or (i > MAXIMO__E) then
    writeln ('FUERA DE RANGO')
  else if librería[i].usado then
    lista__programa(i)
  else
    writeln('INEXISTENTE');
end; {buscar}
```

Aquí aparece por primera vez el misterioso campo *librería[i].usado*, que puede ser TRUE (verdad) o FALSE (¿mentira?). Este apartado lo inicializa el procedimiento *nuevo*, y si es verdad, indica que en ese elemento del array *librería* existe un programa archivado. El método no es muy sutil, pero es muy rápido para buscar y listar, por ejemplo:

```
procedure listar;
  var
    i: contador;
begin {listar}
  for i: = 1 to MAXIMO__E do
    begin
      if librería[i].usado then
        lista__programa(i);
    end;
end; {listar}
```

Como se puede observar, sólo se listan los elementos del array que tienen un programa archivado. La idea deviene en particularmente útil en el caso de eliminar un programa del archivo:

```
procedure eliminar;
  var
    i: contador;
```

```
begin {eliminar}
  writeln('ELIMINAR QUÉ:');
  readln(i);
  if (i < 1) or (i > MAXIMO__E) then
    writeln ('INEXISTENTE')
  else
    librería[i].usado = FALSE;
end; {eliminar}
```

Recurrimos al truco de poner en ese campo un FALSE, por lo que el procedimiento *nuevo* lo encuentra disponible, sin perder tiempo en borrar campo a campo el programa desechado:

```
procedure nuevo;
  var
    i: contador;
begin {nuevo}
  writeln('NUEVO');
  i: = 1;
  while (librería[i].usado) and
  (i < MAXIMO__E) do
    i: = i + 1;
  if (i = MAXIMO__E) then
    lleno: = TRUE
  else
    lee__programa(i);
end; {nuevo}
```

Obsérvese cómo el bucle *while* recorre el array hasta que encuentra un elemento libre, o hasta que llega al final del mismo. En este último caso, la variable global *lleno* se hace cierta, con lo que el programa principal se detendrá.

Aclaraciones finales

El propósito de este artículo gira alrededor de un primer contacto con las estructuras de datos en Pascal; por ello no se ha puesto mayor énfasis en la elegancia y la eficiencia de los procedimientos que gestionan la biblioteca. Aun así, funcionan bien, y para una cantidad de programas entre 100 y 1.000, aproximadamente, van muy rápidos.

Por otro lado, el lector se preguntará por qué no se han incluido un par de *procedures* para salvar y cargar los datos de los programas. La razón es simple: portabilidad. Existen diferencias entre los compiladores a la hora de tratar este tema. Por ello se deja como ejercicio al lector. En cualquier caso, el presente programa se escribió en Turbo Pascal, por lo que funcionará en el Amstrad PC y en los CPC y PCW, ya que existe una versión de este compilador para CP/M. Los que tengan el *Hisoft Pascal* también podrán teclear y ejecutar el programa, usuarios del CPC 464 incluidos.

Hexam: un ensamblador que no gasta memoria

Por: Víctor Prieto

Todos los compiladores tienen el mismo problema: son rápidos, eficaces, etc., pero consumen memoria; ni más ni menos. Algún aguerrido lector se preguntará «¿pero es un fallo consumir memoria?, si precisamente los ordenadores están para eso». En ordenadores con 512 K de RAM se puede introducir un compilador sin que se note la falta de ésta, pero si hablamos de un ordenador con 64 K de las cuales 22 están ocupadas por la copia del firmware en la RAM, el problema es preocupante.

De todas formas el usuario normal no tiene que afrontar estos problemas de memoria. Cualquier persona que quiera trabajar con un determinado lenguaje en el **Amstrad**, tiene de sobra con las veinte y pico K que le deja libre el compilador.

Los que hayan programado en Basic sabrán lo difícil que es realizar un «programón» de más de 15 K, que implica trabajar con cientos de líneas y tratar con miles de errores por cruces de valores entre las distintas variables.

Pero el objeto de todo esto no reside en la utilización de lenguajes varios, sino que se centra en un lenguaje muy concreto, el más árido y difícil de todos, aquel que solamente unos pocos elegidos dominan y que tiene tal riqueza en sus matices y técnicas de programación que cada nueva rutina es como poner una pica en Flandes. Nos referimos sin lugar a dudas al Ensamblador, la forma más primitiva y complicada de programar.

Pues bien, los adictos a este lenguaje están de enhorabuena. Por fin van a poder superar las míticas veinte y pico K de código de una sola pasada. Se acabó aquello de tener el compilador ocupando un espacio precioso en la memoria.

Para los auténticos programadores de este apasionante lenguaje, la casa ACE introduce una herramienta verdaderamente valiosa. La solución (que no nos explicamos cómo no se le había ocurrido antes a alguien) es bien sencilla un Ensamblador en ROM (en este caso Ensamblador y Desensamblador juntos) para que los viciosos del Código Máquina se den una «hartura de bytes».

Hexam

Con la sencilla operación de insertar un pequeño cartucho (10 x 6 x 2 cm) en el slot de expansión de nuestra máquina, el Ensamblador está listo para su uso.

Una vez encendido el ordenador, el sistema se inicializa listo para trabajar en Basic. La única diferencia es un mensaje que aparece en pantalla indicando que el *Hexam* está presente en el sistema preparado para entrar en funcionamiento cuando le llamemos.

Como se indicó arriba, el cartucho contiene dos programas que completan el set de herramientas de un avezado programador. Uno de ellos es el *Editor* con el que generaremos el código fuente que posteriormente ensamblaremos y otro el *Monitor* con el cual podremos analizar detenidamente y desensamblar el código generado por otros compiladores, o por el propio *Hexam*.

Situados en el Basic, la entrada en funcionamiento no puede ser más sencilla: tecleando | EDIT, entra en operación el editor de forma instantánea. Por otro lado, tecleando | MON tenemos el monitor listo para diferir las K de código que le metamos.



Tal vez para los usuarios no experimentados en el Código Máquina el *Hexam* no represente grandes diferencias con los ensambladores normales contenidos en cinta o disco. Es para los programadores de más entidad para los que está diseñada esta herramienta, con un completo sistema de edición y tratamiento de código.

En palabras de ACE «Quien guste de programar en Lenguaje Máquina, apreciará rápidamente la eficacia de este programa, así como el sistema de Soft-keys de que le hemos dotado para la selección de los diferentes comandos».

El editor

Situados en el editor, la pantalla se ha vuelto completamente negra, el área de trabajo está situado por encima de una línea horizontal, debajo de la cual se encuentra la línea de comandos.

Las Soft-keys responden a un sistema por medio del cual cualquier comando puede ser obtenido directamente en pantalla sin tener que escribir una sola letra, la única pulsación de la tecla de función correspondiente hace que éste aparezca por debajo del área de trabajo listo para ser completado con los parámetros necesarios o ejecutado directamente.

Este sistema permite un rápido acceso a los mismos, pudiendo recorrer su repertorio de forma rápida, e indicándonos una vez elegida nuestra opción los parámetros con que hemos de completar la sentencia. Así, no hemos de consultar manuales ni instrucciones, sólo pulsar y seguir; un sistema directo, rápido y seguro.

El total de los comandos que integran el editor está contenido en cuatro líneas, cada una compuesta por seis distintos asociados a las teclas de función comprendidas entre los números F4 y F9, con la tecla F6 para realizar el cambio de línea de comandos.

Cualquier operación relacionada con listados o modificaciones del texto es llevada a cabo gracias al cursor, el cual siempre marcará la posición inicial para cualquier modificación de los bloques de texto.

Éste se sitúa en la posición indicada por medio de las teclas de cursor, siempre que tratemos con pequeñas cantidades de texto. Para grandes listados o efectuar operaciones de inserción y borrado hemos de introducir unos parámetros determinados en los comandos correspondientes.

Para posicionar el cursor en una línea, basta con escribir el número de la línea en la sección de pantalla reservada a comandos y pulsar ENTER.

Inserción y otras opciones

Pulsando la tecla F7 aparecerá en la pantalla de texto ocupando la posición de cursor la

Banco de PRUEBAS



palabra NEW, que nos indica que estamos listos para escribir el texto deseado, el modo de edición se abandona con una nueva pulsación de F7.

La opción de revisión permite efectuar modificaciones dentro de un texto escrito anteriormente. En ella siguen operativas todas las posibilidades anteriores de movimiento del cursor y la salida de la misma se efectúa con una nueva pulsación de F8.

La definición de márgenes de tabulación puede realizarse a gusto del usuario, o utilizando los previamente definidos en el programa, los cuales solamente ocupan una extensión de 40 caracteres, quedando el resto de la pantalla sin tabular.

El borrado de líneas lo efectúa la roden DELETE. Con ella podemos eliminar líneas y bloques de texto utilizando el siguiente modo de operación basado en las teclas de comandos:

Una vez seleccionada la opción, aparecerá en pantalla una línea de comandos complementaria

en la cual tenemos los siguientes parámetros: + > o THRU.

Utilizando el + conseguiremos borrar la porción de texto comprendida entre la posición actual del cursor y el número de líneas especificado tras el signo más, de esta forma indicar + 25 borraría las siguientes 25 líneas de texto.

El > sigue la misma tónica con la particularidad de que ahora la porción de texto acotada corresponde a la comprendida entre la línea del cursor y el número especificado de líneas, pero contadas hacia arriba. En este caso > 20 borraría las 20 líneas anteriores a la línea del cursor.

El parámetro THRU abre las puertas a tres nuevas posibilidades:

THRU START, con lo que conseguiremos borrar el texto comprendido entre la línea de cursor y el principio del texto.

THRU END que efectúa el borrado entre la línea de cursor y el final del texto.

THRU LINE acota para el borrado el bloque de líneas comprendido entre la línea en la que se haya el cursor y el número introducido por teclado.

Con el conjunto de parámetros especificado es fácil delimitar el bloque que se ajuste exactamente a nuestras necesidades, siempre de una forma fácil y sin líos en el orden de los mismos.

Operaciones con ficheros

El tratamiento de ficheros en disco se efectúa con los comandos propios del Basic: CAT, LOAD, SAVE, MERGE.

CAT realiza un directorio del disco, sacando en la pantalla de texto el resultado del mismo.

LOAD efectúa la carga de un fichero desde el disco. Éste ha de estar en código fuente y debe haber sido creado por el editor, en otro caso el programa no reconocerá el fichero especificado, *Hexam* graba sus ficheros sin extensión alguna y con un nombre cuya longitud no ha de exceder de ocho caracteres, no siendo necesario situarlos entre comillas.

SAVE graba un fichero en disco, pudiendo ser el nombre del mismo especificado por defecto, es decir, cuando el fichero estaba anteriormente en disco y hemos procedido a su carga para añadir o modificar líneas, o introduciéndole con un nuevo nombre.

MERGE posibilita una aplicación muy útil y totalmente distinta a la que se le da en Basic.

Mediante este comando podemos añadir el texto contenido en el editor una determinada porción de otro fichero contenido en el disco.

MERGE PROGRAMA FROM 100 THRU 200

unirá al fichero contenido en memoria las líneas comprendidas entre la 100 y 200 (ambas incluidas) del fichero con el nombre PROGRAMA contenido en el disco.

Tratamiento de líneas

La copia de bloques de texto se realiza de forma similar al borrado de los mismos especificado con anterioridad en DEL. La forma de acotar los bloques es la misma para todos los comandos que trabajan con ellos.

En el caso de la copia se utiliza el comando COPY, en el cual la máxima cabida de texto para efectuar la copia es de 2 K. Por encima de esta cifra revasaremos la cantidad de memoria reservada a este fin.

EXTRACT permite extraer texto del fichero con el que estamos trabajando. Éste pasa a una zona de memoria transitoria para ser procesado posteriormente por el comando RETRIEVE.

RETRIEVE recupera el texto acotado por las órdenes COPY o EXTRACT y lo reproduce en la zona deseada a partir de la posición de cursor.

Para copiar más de una vez el texto archivado en el bufer de copia, es posible especificar el número de veces que deseamos reproducirlo detrás de la orden RETRIEVE.

RENUM dota de número a las líneas que todavía no tienen, actualizando de paso la numeración de las demás. Este comando se ejecuta desde el campo de trabajo por introducción del mismo, o de forma automática cuando pretendemos grabar un fichero o ensamblar un texto sin renumerar.

Búsqueda de cadenas

FIND busca la cadena especificada entre comillas a partir de la línea siguiente a la del cursor y a lo largo de todo el texto señalándonos con el cursor la posición de ésta cuando ha sido localizada. Una pulsación de ENTER hará que continúe la búsqueda por el resto del texto hasta dar de nuevo con la cadena requerida.

REPLACE realiza la sustitución de la cadena especificada entre comillas a lo largo de la porción del texto marcada por el comando RANGE. Con ella podemos cambiar el nombre de variables, etiquetas, códigos de instrucción y todo lo que se nos ocurra.

RANGE sirve para acotar el campo en el que posteriormente ha de actuar REPLACE. La anchura del campo delimitado puede variar desde el total de la pantalla hasta una columna de un solo carácter.

PRINT hace un volcado por impresora del texto que tenemos en el área de trabajo. La porción de texto que ha de ser impresa se especifica por medio de los parámetros de marca de bloques, pudiendo volcar desde una sola línea

hasta la totalidad del mismo, pasando por cualquier porción de éste.

AUTO TAB sitúa automáticamente el cursor en la columna correspondiente al código de operación, evitando de esta manera la tediosa y repetitiva operación de pulsar TAB al principio de cada línea.

Assembler

Es el comando que permite ensamblar el código fuente que tengamos en memoria en código directamente ejecutable por el ordenador.

Sus normas de sintaxis permiten que las etiquetas puedan escribirse en mayúsculas o minúsculas, debiendo obligatoriamente escribirse los códigos de instrucción en minúsculas.

Las especiales características de este ensamblador permiten incluir comentarios al final de los códigos de instrucción sin necesidad de ser separados por «;» que sólo serán utilizados en el caso de querer dedicar una línea completa como comentario o separador.

Dentro de la operación de ensamblado podemos requerir la ejecución de las siguientes operaciones:

LIST da un listado en pantalla del resultado del proceso, sacando de izquierda a derecha la posición adjudicada a la instrucción, código generado, código fuente y comentarios.

PRINT vuelca por impresora el listado anterior.

LIST/WE lista los mismos parámetros, pero deteniendo la ejecución en el comando donde se detecta un error. La pulsación de una tecla cualquiera pondrá en marcha el proceso de nuevo.

PRINT/E lista los errores detectados por impresora, sacando únicamente el tipo de error y el número de línea donde se ha producido.

La ejecución del comando ASSEMBLER puede ser detenida con ABORT.

BASIC permite el paso al Basic del ordenador desde nuestro ensamblador.

En los modelos 664 y 464 en este trasiego perderemos el código contenido en memoria, pero en el modelo 6128, éste permanecerá intacto y listo para ser recuperado al volver de nuevo al editor.

El monitor

En otro programa aparte del editor tenemos el monitor de Código Máquina, con el cual podremos desensamblar cualquier rutina producida por el *Hexam* o cualquier ensamblador del mercado.

La puesta en marcha del mismo se hace desde el Basic con el simple tecleo de I MON, instantes después aparecerá la pantalla del desensamblador listo para recibir el código requerido.

Banco de PRUEBAS



Las opciones del mismo se obtienen por el simple tecleo de la inicial correspondiente, las cuales permanecen en la parte inferior de la pantalla constantemente junto con todos los comandos del mismo.

En definitiva, un amplio abanico de posibilidades que permiten desensamblar cualquier porción de código, reubicarla en distintas porciones de memoria, pasar ficheros al disco, listar el desensamble por impresora y todas las operaciones necesarias para que cualquier Hacker haga maravillas con los programas comerciales.

Conclusiones

Tanto programadores expertos como iniciados con curiosidad en introducirse en el Código Máquina encontrarán en este programa en ROM una herramienta insustituible a la hora de realizar o analizar programas.

La forma fácil de utilización tanto del editor como del monitor, así como la amplia variedad de comandos incluidos, hacen que los ensambladores en disco o en cinta queden fuera de órbita tras la estela del *Hexam*, que con su completo editor proporciona la herramienta más potente de programación conocida hasta la fecha en los modelos **Amstrad**.

Esta sección está dedicada a todas las compras, ventas, clubs de usuarios de **Amstrad**, programadores y, en general, cualquier clase de anuncio que pueda servir de utilidad a los lectores. Todo aquel que lo desee puede enviarnos su anuncio, mecanografiado, a: **HOBBY PRESS, S.A. AMSTRAD Semanal**. Apartado de Correos 232 Alcobendas. Madrid.
¡ABSTENERSE PIRATAS!

Mercado
COMÚN

.....
Vendo o cambio programas para el 6128, 464, 664, a 200 ptas. cada uno. Tengo algunos como: *1942, Breakthru, Galván, Army Moves, Future Knight, Infiltrator* y muchos más. Interesados escribir o llamar a *Clemente Martín*. San Martín, 4, 2.º C. 47400 Medina del Campo (*Valladolid*). Tel. (983) 80 12 89 (de 9 en adelante). Enviar lista.

.....
Intercambio todo tipo de programas, juegos y aplicaciones, tengo más de 300. *Arturo Latierro*. Enrique Eguren,

.....
3- 4 A. Vitoria. Tel. 22 05 27, a partir de las 6 de la tarde.

.....
Vendo Amstrad CPC 6128, impresora Compute Mate 100, unidad de disco DDI-1, Pascal MT Plus y dBase II. Tel. (96) 373 94 54 (mañanas).

.....
Cambio programas para **Amstrad** CPC's. Enviar lista a la siguiente dirección: *Amador Abelleira*. Valencia, 23, 2.º D. Gijón (*Asturias*). Prometo contestar.

.....
NECESITAMOS personas

que conozcan, profundamente, lenguaje Ensamblador del 8086/8088 y dominen los ordenadores compatibles IBM PC, desde el punto de vista del lenguaje Máquina. Se valorarán en gran medida conocimientos acerca del Dos.

Si usted es uno de ellos, póngase en contacto con nosotros rápidamente, en la dirección que se indica a continuación, poniendo en lugar muy visible del sobre:

Referencia: Dos AMSTRAD Semanal.

Nuestra dirección es:

AMSTRAD Semanal

Ctra. de Irún. Km 12,400. Fuencarral (28049 Madrid).

De chip a chip
"Sábado Chip", de 17 a 19 h.

.....
Vendo videojuegos Philips G 7000 con dos videopacs (uno de ellos con expansión de memoria). Precio a convenir. Tel. 430 88 52.

.....
Cambio programas con usuarios del **Amstrad** 464, 664, 6128. Poseo más de 350 programas, entre ellos las últimas novedades. Contestaré a todos. Escribir a *José Enrique Sánchez Hurtado*. Torres Quevedo, 3, 3º C. 02003 Albacete. O bien llamar al tel. (967) 23 49 88.

.....
Desearía contactar con usuarios del **Amstrad** CPC 6128, preferiblemente de Elche. Interesados escribir a *Manuel González Tobar*. Concepción Arenal, 57, 6.º dcha. Elche (*Alicante*). Tel. 44 08 06. Al escribir poner vuestro número

de teléfono y lista de programas de vuestra propiedad, yo haré lo mismo en cuanto reciba vuestras cartas, incluso podríamos formar un club de usuarios en Elche.

.....
Cambio Compacto Panasonic (plato, sintonizador y cassette) por ordenador **Amstrad** o Commodore. *Luis Miguel Prego Otero*. Rúa do Muelle, 17. Cambados (*Pontevedra*). Tel. (986) 54 20 14.

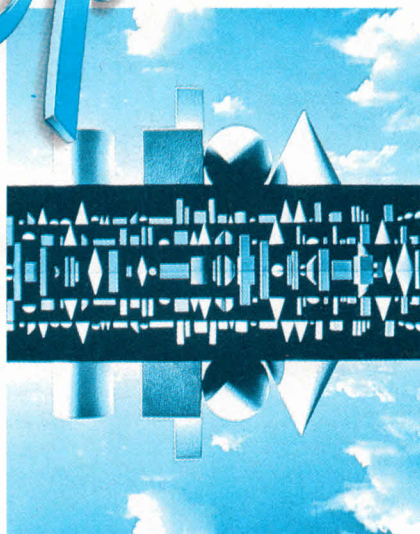
.....
Vendo Amstrad CPC 128, monitor en color, impresora *Seikosha SP 1000CPC*, con los siguientes programas: Tasword 128 con mailmerge, Wordstar con mailmerge, Multiplán, dBase II, cBasic, Dr. Graph, Micropen, Microscript, copiones y juegos múltiples, una verdadera ganga, sólo los programas en el mercado ya

valen más de 150.000 ptas. Todo por sólo 190.000 ptas., incluida de regalo la colección completa de **AMSTRAD Semanal**, 6 cintas nuevas para la impresora y muchos discos. Todo en perfecto estado y bajo garantía. Interesados llamar al (96) 352 51 94 ó 361 37 38 Valencia.

.....
Deseo contactar con usuarios (o un club de usuarios) de **Amstrad** PCW 8256 para intercambio de programas. Dispongo de Placon, dBase II, Facturación y almacén, Contabilidad + Vencimientos y los juegos *Batman* y *3D Clock Chess*. Me interesaría copiones, *Supercalc 2*, un programa de mediciones y presupuestos. Interesados escribir a *Salvador Montoya Gutiérrez*. Ocaña, 31, 12.º B. 28047 Madrid. Tel. 718 36 06. Preferentemente noche.

Radio Pestillo Cope

Todos los sábados, de 5 a 7 de la tarde, en "Sábado Chip". Dirigido por Antonio Rúa. Presentado por José Luis Arriaza, hecho una computadora. Dedicado en cuerpo y alma al ordenador, y a la informática. Haciendo radio chip... estilo Cope.



Cadena Cope
RADIO POPULAR



... de chip a chip

Mercado COMÚN

Vendo Amstrad CPC-6128, fósforo verde, con poco uso, manuales completos, etc. por 76.000 ptas. Se incluyen discos vírgenes, varios libros, muchas revistas, etc. Tel. (91) 850 23 49 *Eduardo C. Villalba (Madrid)*.

Desearía vender a buen precio: un modulador para Amstrad 464. Revistas **AMSTRAD Semanal** números del 1 al 42 inclusive. Preguntar por *M.ª Ángeles*. Tel. 23 31 55. Zaragoza.

Deseo contactar con usuarios de **Amstrad PCW-8256/512** para intercambio de programas. Interesados llamar al tel. (93) 399 25 30 de Badalona (*Barcelona*) o escribir al apartado de correos 60, de la misma ciudad a *Luis García*.

Si tienes un PCW y dispones de uno de estos programas: Gestión de Videoclubs, Declaración renta (desprotegido) o Gestión integrada de empresa, no lo pienses más y llama al tel. (968) 59 43 39. Tardes o escribe a *Adolfo Cerón*. General Aranda, 23. 30870 Mazarrón (*Murcia*). Pago bien.

Vendo ordenador Amstrad PCW-8256 (con impresora) ampliado a 512 Kb y segunda unidad de disco de 720 Kb, con los programas: Superficial, dBase II, cBasic Compiler y Pascal MT + . Todo por

160.000 ptas. Llamar al tel. (972) 23 93 71, de 21 a 23 horas. Preguntar por Luis.

Desearía contactar con usuarios del **Amstrad CPC-464/664/6128** para intercambio de programas de todo tipo, información, ideas, etc. Escribir a *Juan Antonio García Montoya*. Los Naranjos, 7, El Palmar (*Murcia*). Tel. (968) 84 19 56.

¡Atención! Se vende el siguiente paquete de programas de utilidades (en disco). Contabilidad doméstica y personal, Facturación, Genaload, Monaload, Tasword, Tascopy, Amsdraw II, Amscalc y Copión por el precio de 4.300 ptas (disco incluido). Para información escribir a *Josep Sánchez R.* Avda. San Narcís, 98. 17005 Gerona.

Vendo Amstrad PCW 8256 con monitor fósforo verde, con muy poco tiempo y ningún uso. Embalaje perfecto. Regalo discos, base de datos y revistas y varios programas, uno de contabilidad, etc. Todo en perfecto estado. Llamar a *Modesto*. Tel. (986) 25 37 92. O escribir a *Modesto Otero Durán*. Travesía de Vigo, 275, 2.º, puerta 32. Vigo.

¡Atención! Se vende el siguiente paquete de programas de juegos (en disco): *3DChess* (castellano), *Mercenario*, *Viernes 13*, *Flight Pilot*, *Sorcery*, *Bomb Jack* y *Fórmula 1* por el precio de 3.800 ptas. (disco incluido). Para información escribir a *Josep Sánchez R.* Avda. San Narcís, 99. 17005 Gerona.

Club de usuarios de Amstrad, en toda España, requiere socios. Tenemos más

de 500 socios. Al inscribirte recibirás completamente gratis cinco juegos (a tu elección). Tenemos gran variedad de juegos, pokes, ideas... Entre los juegos están: *Army Moves*, *Commando*, *The Way off the Tiger II*, *3DStar Trike I y II*, *1942*... Para más información dirigirse a *Carlos Martín Mesa*. Amatista, 63. Getafe (*Madrid*) Tel. (91) 682 80 69.

BOUTIQUE DEL ORDENADOR

AMSTRAD PCW ... 69.500
AMSTRAD 464 39.500

Procedentes de exposiciones o cambios. Garantizados, impecables.

Consulte para unidades de disco, impresoras, etc...

ENVÍOS A TODA ESPAÑA
Tel.: (91) 416 13 02 (tardes)

Vendo unidad de disco para CPC-664 y CPC-6128 casi a estrenar, precio muy interesante. Llamar o escribir a *José Luis Ruiz*. Cruz del sur, 3, 4.º D. 28007 Madrid. Tel. 274 84 77.

NECESITAMOS personas

que conozcan, profundamente, lenguaje Ensamblador del Z80 y dominen los ordenadores Amstrad CPC, desde el punto de vista del lenguaje Máquina.

Si usted es uno de ellos, póngase en contacto con nosotros rápidamente, en la dirección que se indica a continuación, poniendo en lugar muy visible del sobre:

Referencia: Código Máquina AMSTRAD Semanal.

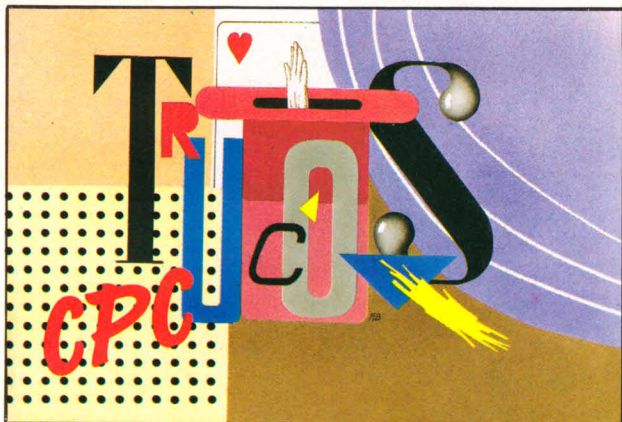
Nuestra dirección es:
AMSTRAD Semanal
Ctra. de Irún. Km 12,400.
Fuencarral (28049 Madrid).

GIGANTES *del Basket*

DA LA TALLA

Gigantes te espera cada martes en tu kiosco. Una revista imprescindible para todo aficionado. Gigantes del Basket te ofrece las últimas noticias, las mejores fotos, las opiniones que te interesan, las entrevistas que esperas, las anécdotas que te divierten y los los comentarios que te gusta conocer. Pídelas todas las semanas, porque Gigantes da la talla.





Carkanoid (CPC 664-6128)

La semana pasada estudiamos unas rutinas que nos permitían mover una raqueta y emular el movimiento de una bola con sus rebotes. Esta vez hemos realizado una mejora incluyendo también en la pantalla una serie de ladrillos que pondrán nuestra tarea más difícil.

Respondiendo simplemente con «return» a la pregunta «Nombre...?», obtendremos una pantalla de demostración con la disposición clásica de los ladrillos.

Para diseñar nuestras propias pantallas hemos creado el diseñador de ladrillos, que le permitirá de forma fácil y sencilla hacer sus propias composiciones. Su utilización se detalla en el siguiente truco.

La inclusión de la paleta con su movimiento no la hemos implementado, y dejamos al ávido lector esta tarea que estamos seguros conseguirá simplemente fundiéndolo con el número anterior y realizando alguna pequeña modificación.

```

10 REM -----MURO-----
20 REM SOLO CPC 664 CPC 6128
30 DEFINT a-z
40 INK 0,1:INK 2,22
50 BORDER 0
60 MODE 1
70 SYMBOL 254,0 ,254,254,254,
254,254,254,0
80 SYMBOL 255,0 ,127,127,127,
127,127,127,0
90 RANDOMIZE(TIME)
100 REM DIBUJA PANTALLA
110 INPUT "NOMBRE...";NOM$
120 CLS
130 IF nom$="" THEN 340
140 MODE 1
150 OPENIN NOM$
160 WHILE NOT EOF
170 INPUT #9,A$
180 PRINT:PRINT a$;
190 WEND
200 PAPER 0:PEN 1
210 REM PROGRAMA PRINCIPAL
220 pelota$=" o"
230 x=INT(RND*20)*2:ix=1:iy=-1
:y=25
240 IF x=0 THEN 230
250 LOCATE x,y:IF COPYCHR$(#0)
<>" " THEN iy=-iy:SOUND 5,401,
3,15
260 LOCATE x,y:FRAME:PRINT pel
ota$;
270 IF x=1 AND ix=-1 THEN ix=1
:SOUND 5,201,5,15
280 IF x=39 AND ix=1 THEN ix=-
1:SOUND 5,201,5,15
290 IF y =1 AND iy=-1 THEN iy=
1:SOUND 5,201,5,15
300 IF y =25 AND iy= 1 THEN iy
=-1:SOUND 5,201, 5,15
310 LOCATE x,y:PRINT " ";
320 x=x+ix:y=y+iy
330 GOTO 250
340 FOR N=1 TO 20:A$=A$+CHR$(2
55)+CHR$(254):NEXT
350 PRINT:PRINT a$;:PRINT:PRIN
T a$;:PRINT:PRINT a$;:PRINT:PR
INT a$;
360 GOTO 210

```

Editor de ladrillos

El objetivo de este programa es realizar las pantallas que después podrán ser utilizadas por el programa anterior. Una vez ejecutado el programa y que hayamos dado nombre a la pantalla, nos moveremos con las teclas del cursor. Para fijar un ladrillo pulsaremos el punto y para borrarlo, el «0».

Una vez estemos convencidos de la disposición de nuestros ladrillos pulsaremos «return» y automáticamente tendremos una copia de ésta en el disco.

Para utilizarlos ahora con el Carkanoid, tendremos que teclear el mismo nombre con el que fue grabado, cuando se nos solicite éste.

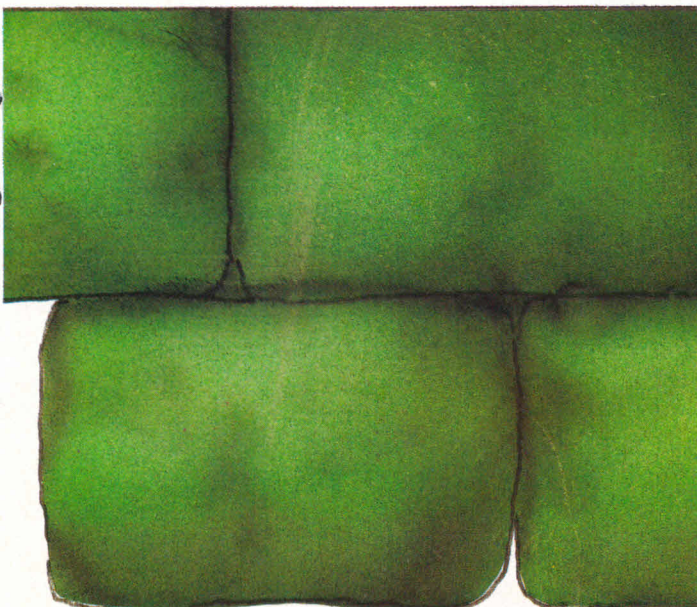
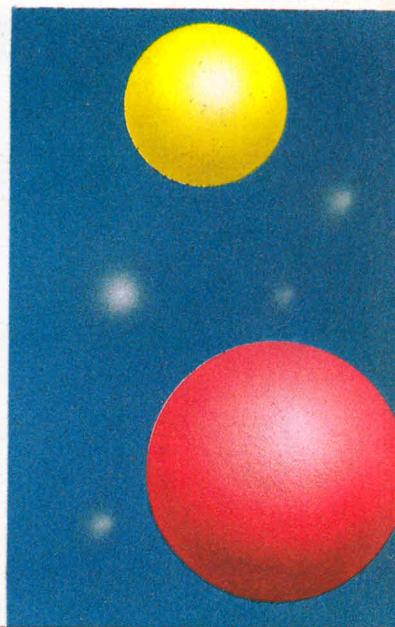
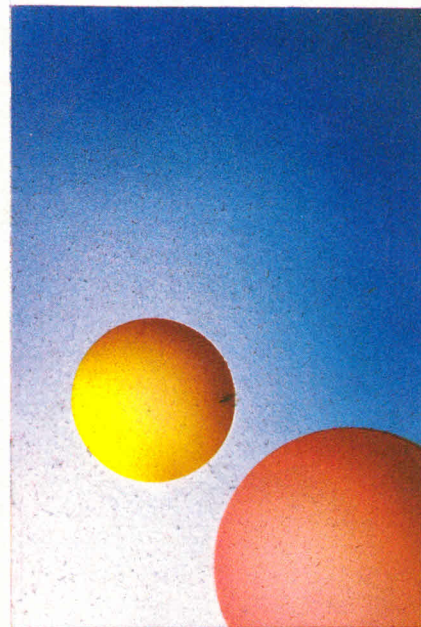
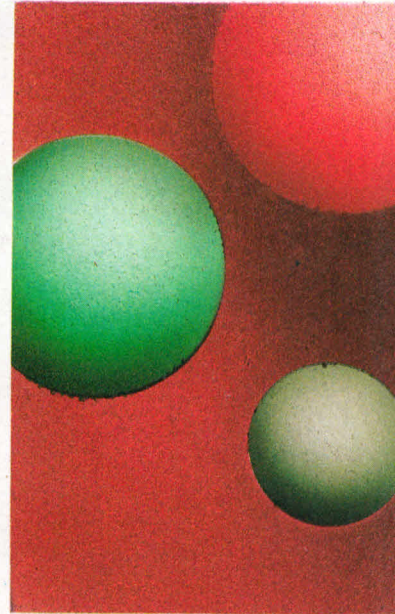
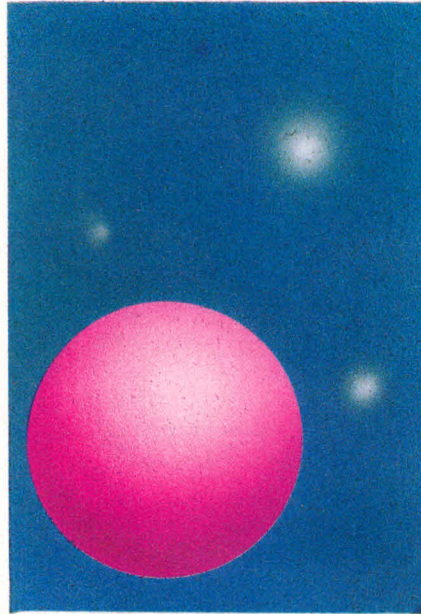
Imágenes fantásticas ▶

Como ya es habitual en esta sección no podían faltar los programas gráficos. En esta ocasión es un programa que nos permitirá observar toda una serie de bonitos y variados dibujos.

```

10 REM      ----      -----
20 REM SOLO CPC 664 CPC 6128
30 DEFINT a-z
40 INPUT "NOMBRE DE LA PANTALLA"; NOMBRES$
50 INK 0,1:INK 2,22
60 BORDER 0
70 MODE 1
80 SYMBOL 254,255,1,1,1,1,1,1,1,255
90 SYMBOL 255,255,128,128,128,128,128,128,255
100 REM PROGRAMA PRINCIPAL
110 Ladrillo$=CHR$(255)+CHR$(254)
120 x=1:y=1
130 CURSOR 1,1
140 WHILE tecla$<>CHR$(13)
150 LOCATE x,y
160 IF tecla$="." THEN PRINT Ladrillo$;:x=x+2
170 IF tecla$="0" THEN PRINT " ";
180 x=x+(tecla$=CHR$(242))*2-(tecla$=CHR$(243))*2
190 y=y+(tecla$=CHR$(240))*2-(tecla$=CHR$(241))*2
200 IF X>39 THEN X=1:Y=Y+2
210 IF Y>25 THEN Y=1
220 IF X<1 THEN X=38
230 IF Y<1 THEN Y=25
240 TECLA$=INKEY$
250 WEND
260 OPENOUT NOMBRES$
270 FOR y=1 TO 23 STEP 2
280 FOR x=1 TO 40:LOCATE x,y:A$=A$+COPYCHR$(#0)
290 NEXT:WRITE #9,A$:A$="":NEXT
T

```

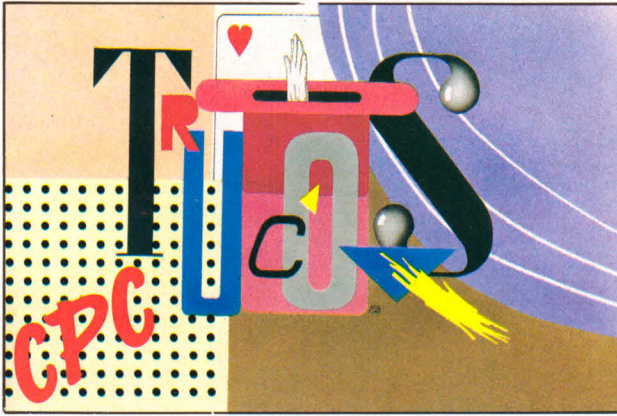


```

10 MODE 2:PLOT 100,200
20 FOR ang=0 TO 10
21 MODE 2:PLOT 100,200

30 FOR n=1 TO 200 STEP 0.4
40 DRAW 100+COS(N)*n+ang*n,20
0+SIN(n)*n
50 NEXT
60 NEXT

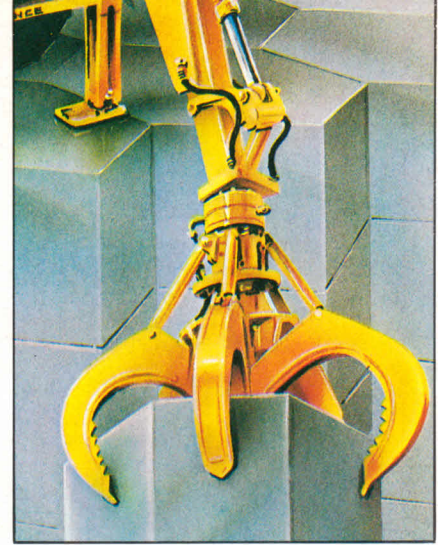
```



Caracteres multicolores

Por José María Arbex

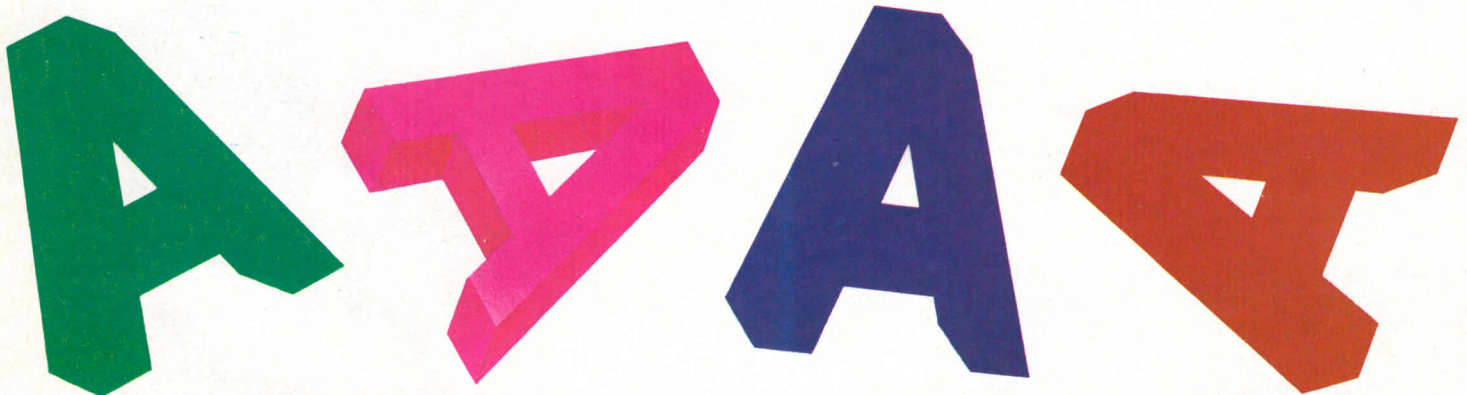
Este pequeño truco permite obtener caracteres de dos colores, empleando solamente el Basic. La parte de arriba del carácter aparecerá con el color de la pluma 1 y la parte de abajo con el de la pluma 3. En el caso de que no se redefinan los colores estos caracteres, serán amarillos y rojos. La idea del truco es muy simple. Se trata de redefinir cada carácter a ser impreso como su parte de arriba (4 bytes) y su parte de abajo (otros cuatro bytes), e imprimirlo con dos pasadas con distinto «pen», en modo transparente. El programa incluye un pequeño editor para probar su funcionamiento y está preparado para ser utilizado en modo 1 o modo 0, cambiando simplemente el valor de la variable MO de la línea 20.



Muévete

La semana pasada pudimos ver cómo podíamos hacer entender a nuestro ordenador algunas de las cosas que sucedían en su exterior a través de la toma de Joystick. Si usted dispone de la versión 664 ó 6128 también podrá realizar la operación inversa actuando a través de la conexión remote. Para conseguir que algo se mueva fuera necesitará poner en su ingenio una conexión del mismo tipo que la de los cassettes convencionales.

Para enviar una instrucción de «en marcha» desde el ordenador, sólo tendrá que hacer un Call a &BC6E motor ON, y si desea desactivarlo, Call &BC71.



```

20 mo=0:MODE mo:SYMBOL AFTER 2
56:SYMBOL AFTER 0
30 nhi=HIMEM+521:hi=1:PRINT CH
R$(22)+CHR$(1)
40 a$="Escribe lo que quieras:
":ret=1:FOR z=1 TO LEN(a$):F$
=MID$(A$,Z,1):x=x+1:GOSUB 60:N
EXT:hi=4:x=1:ret=0:MOVE 0,380:
DRAWR 356,0,2
41 f$=INKEY$:IF f$="" THEN 41
ELSE x=x+1:IF x=21+20*mo-20*(m
o=2) THEN hi=hi+1:x=1
50 IF f$=CHR$(127) AND x>1 THE
N PEN 0:PRINT CHR$(8)+CHR$(143

```

```

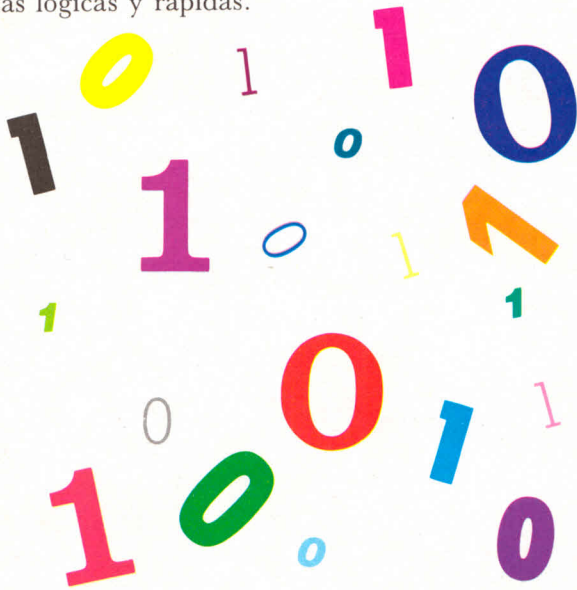
)+CHR$(8);:x=x-2:PEN 1:GOTO 41
60 y= 8*((ASC(f$)-64)-1)+nhi
70 FOR des=0 TO 2 STEP 2
80 IF des=0 THEN SYMBOL ASC(f$
),PEEK(y),PEEK(y+1),PEEK(y+2),
PEEK(y+3),0,0,0,0 ELSE SYMBOL
ASC(f$),0,0,0,0,PEEK(y+4),PEEK
(y+5),PEEK(y+6),PEEK(y+7)
90 LOCATE x,hi:PEN des+1:PRINT
f$;
100 SYMBOL AFTER 256:SYMBOL AF
TER 0:NEXT:SOUND 1,286,1,15,,
1:IF ret =1 THEN RETURN ELSE 4
1

```

Conversión a sistema binario

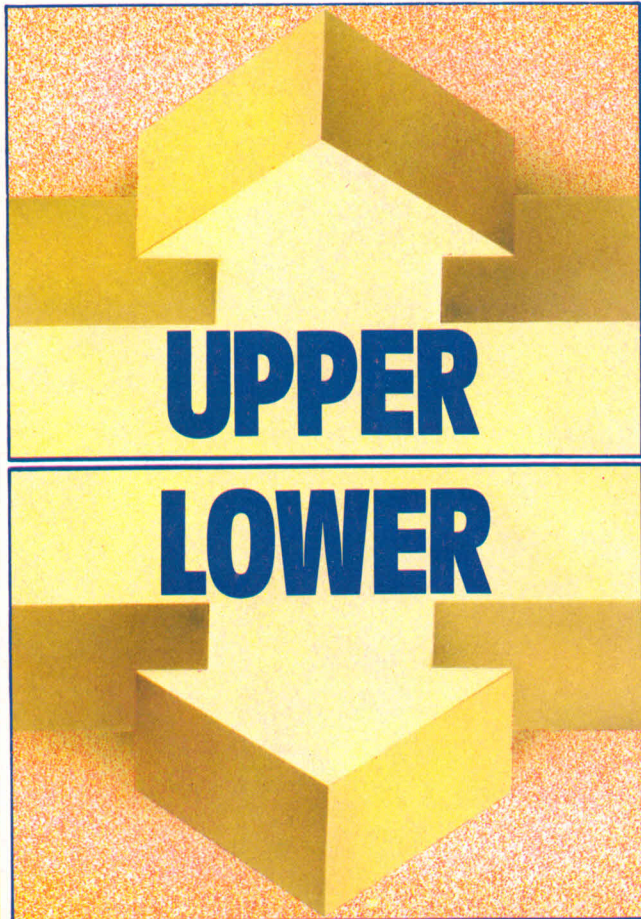
El sistema binario resulta totalmente imprescindible para realizar múltiples operaciones con el ordenador.

Las formas de pasar un número decimal a binario son múltiples y variadas y ésta que hoy proponemos puede considerarse como una de las más lógicas y rápidas.



```

10 REM CONVERSION DE BINARIO A
  DECIMAL
20 INPUT "NUMERO BINARIO:.....
.....",BINARIO$
30 FOR N=LEN(BINARIO$) TO 1 STEP -1
40 DECIMAL=DECIMAL+VAL(MID$(BINARIO$,N,1))*2^(LEN(BINARIO$)-N)
50 NEXT N
60 PRINT DECIMAL
70 RUN
  
```



Mejor que UPPER\$ y LOWER\$

Si es usuario de un 664 o de un 6128 y desea que su CPC olvide las mayúsculas, le bastará con hacer `Poke(&B68B),peek(&B68D)`. Si desea lo contrario tendrá que realizar la operación inversa: `Poke(&B68D),peek(&B68B)`.

En el caso de que disponga de un 464, las direcciones son ligeramente distintas.

Para anular las minúsculas: `Poke(&B541),peek(&b543)`

Para anular las mayúsculas: `Poke(&B543),peek(&b541)`



CUÁL ES SU TRUCO



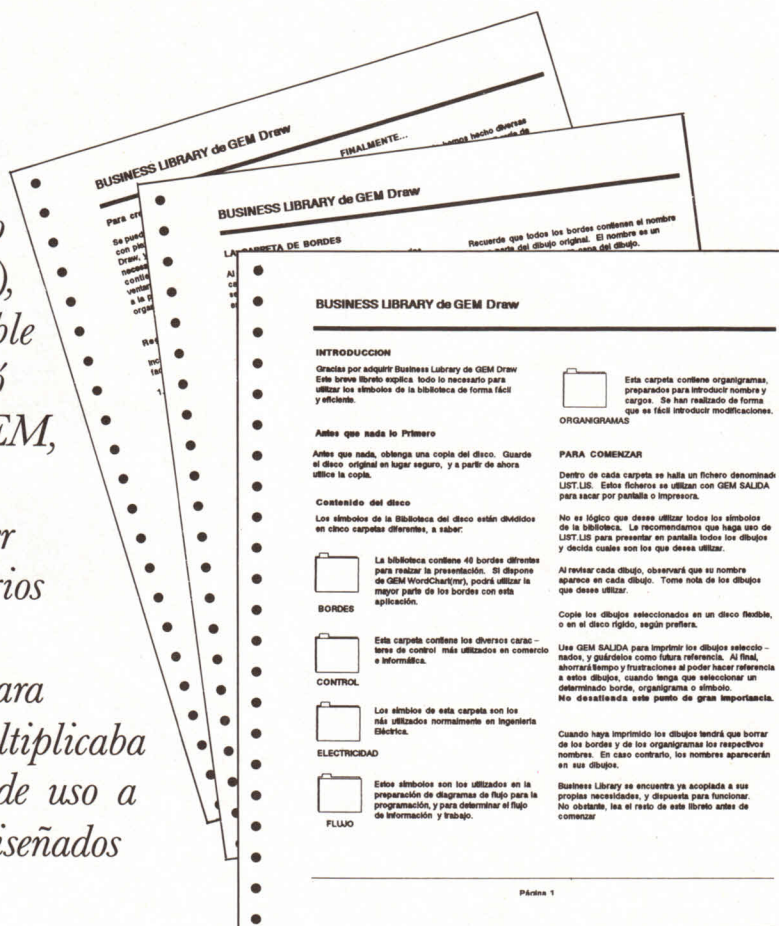
Desde esta columna queremos invitar a todos, sin excepción, a que participen en nuestra revista y se conviertan en parte aún más activa, si cabe, de la misma. Estamos convencidos que durante el uso diario de su ordenador han descubierto multitud de trucos y pícaras artimañas, maneras de hacer lo imposible, que han convertido horas de tedioso trabajo en momentos en los que la chispa de la inspiración le hacen a uno felicitarse por haber comprado un Amstrad. Si efectivamente nos los han descubierto, ¿por qué no nos los envían? Nosotros los miraremos todos, y los seleccionados recibirán como compensación por su esfuerzo cuatro cintas de cassette repletas de juegos, utilidades y programas publicados por **AMSTRAD Semanal**, de las que publicamos cada mes. Además, si el lector nos indica en la carta cuáles quiere, le enviaremos precisamente éstas, siempre que no estén agotadas.

Enviar las cartas a la siguiente dirección: **AMSTRAD Semanal**. Ctra. de Irún, Km. 12,400. 28049 Madrid.
Indicando claramente en el sobre: **AMSTRAD Semanal**. Referencia Trucos.

GEM Draw. El arte de dibujar con un PC

Por: Fco. Javier Barceló

Cuando Alan Sugar diseñó el Amstrad PC, decidió crear, por un precio realmente bajo (en principio), algo más que un compatible PC (en principio). Eligió incluir el entorno gráfico GEM, de Digital Research. Con ello, además de poner a disposición de los usuarios una inmensa biblioteca de programas diseñados para los PCs (en principio), multiplicaba la misma y su facilidad de uso a través de los programas diseñados en este entorno.



Por **AMSTRAD Semanal** pasa hoy la primera versión de un programa que pronto estará, en su versión definitiva, en la calle: *GEM Draw*. Su orientación está en el campo del dibujo lineal, y sus características responden a lo deseado.

El primer paso a dar con esta aplicación es instalarla. Gracias a Dios, esto se realiza automáticamente, mediante el programa *Instal* que viene en el disco maestro. Con este programa se obtiene una copia de los programas más importantes, bien en un disco flexible, o bien en el disco duro. El programa principal, tal y como vienen configuradas todas las aplicaciones de GEM, queda instalado en el subdirectorio (o carpeta) GEMAPPS, pero en cualquier caso es necesario tener el disco maestro introducido en el ordenador para poner en marcha el programa, debido a la protección *anti-piratería* que incorpora. La aplicación viene acompañada de dos carpetas con dibujos de ejemplos, que dan una idea de la potencia del programa.

Una vez en marcha, aparece la pantalla principal, en la que la primera fila está dedicada

FICHA TÉCNICA

Programa	: GEM DRAW
Sist. Operativo	: MS-DOS 3,20
Ordenador mínimo	: PC 1512 con 2 unidades de disco
Precio	: 22.300
Distribuidor	: Microbyte

a los índices de los distintos menús disponibles y la primera columna a los tipos de trazos que se pueden realizar.

El dibujo

Para empezar a dibujar, llevando el puntero a la columna izquierda, se elige el tipo de figura que se va a realizar. Podemos seleccionar rectángulos, elipses, círculos, polígonos, arcos, líneas y textos. Una vez escogida la figura, se coloca el cursor en el punto donde se quiere empezar a dibujar, y se aprieta el botón izquierdo del ratón. Ahora, se traslada el ratón

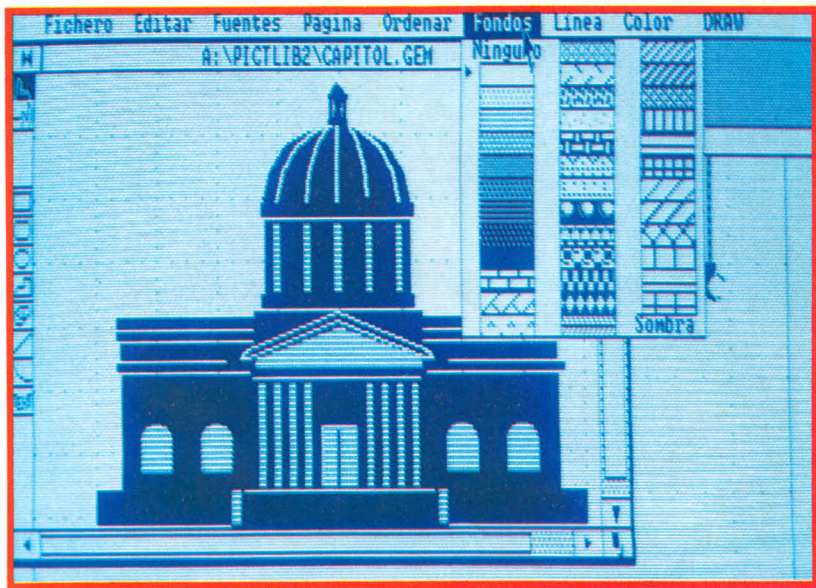
Mundo del PC

por la pantalla y, según la figura elegida, se va creando el dibujo. Una vez que se tiene la forma deseada, se suelta el botón del ratón y queda fijada. Ahora se puede elegir otra figura para continuar el dibujo, o mover la parte seleccionada por la pantalla para colocarla en el lugar deseado.

Además del dibujo propiamente dicho, se pueden duplicar las imágenes, moverlas por la pantalla, centrarlas o alinearlas automáticamente y modificarlas. Para ello, lo primero que hay que hacer es seleccionar la parte del dibujo en la que se va a operar. Si se desea funcionar con varias partes del dibujo a la vez, hay que agruparlas previamente, dado que cada vez que se realiza un tramo del dibujo, el programa lo considera una parte distinta.

La utilidad más importante es una mano dibujada en la columna izquierda. Al seleccionarla con el puntero, se puede elegir una parte del dibujo, o agrupar determinados elementos, y, en general, hacer uso de los menús de la primera fila, operando sólo con la parte seleccionada.

El texto se puede colocar en cualquier parte, en dos tipos de letras distintas y varios tamaños



diferentes, pudiéndose también mover y centrar automáticamente.

A la hora de realizar dibujos profesionales, en cuanto se familiariza uno con el programa, los resultados pueden ser asombrosos. La facilidad del zoom para ampliar permite dibujar con gran detalle cualquier cosa, y apoyándose en herramientas como la regla, de escala regulable (desgraciadamente sólo en pulgadas) o la cuadrícula, también de tamaño variable, evita la pérdida de referencias. No obstante, para alinear o centrar el texto es conveniente tener el zoom sin ampliar, dado que lo que se ve aumentado en la pantalla es parte del dibujo, y al imprimirlo todo, la figura o texto pueden no estar centradas.

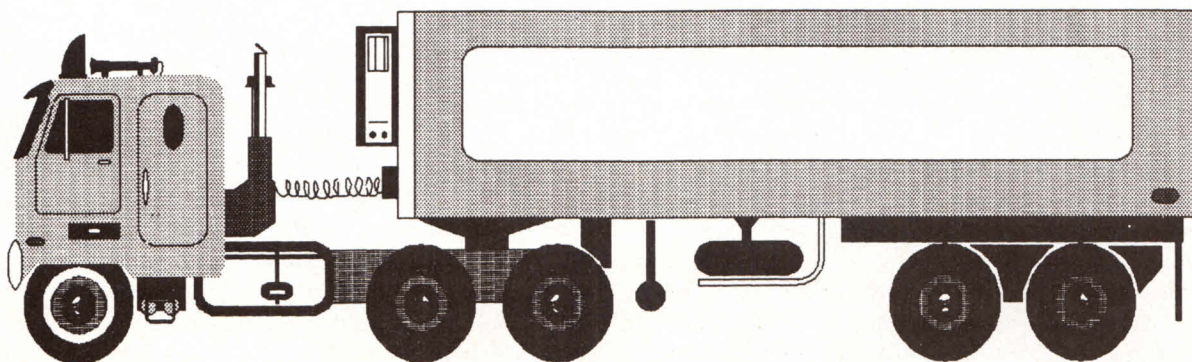
Dispone de una serie de menús y explicaciones que ayudan a su fácil manejo.

Menús

Ficheros. Incorpora funciones para cargar un dibujo del disco, salvarlo, cambiarlo de nombre, y conectar con la aplicación "OUTPUT" que permite imprimir los dibujos y salvarlos en disco

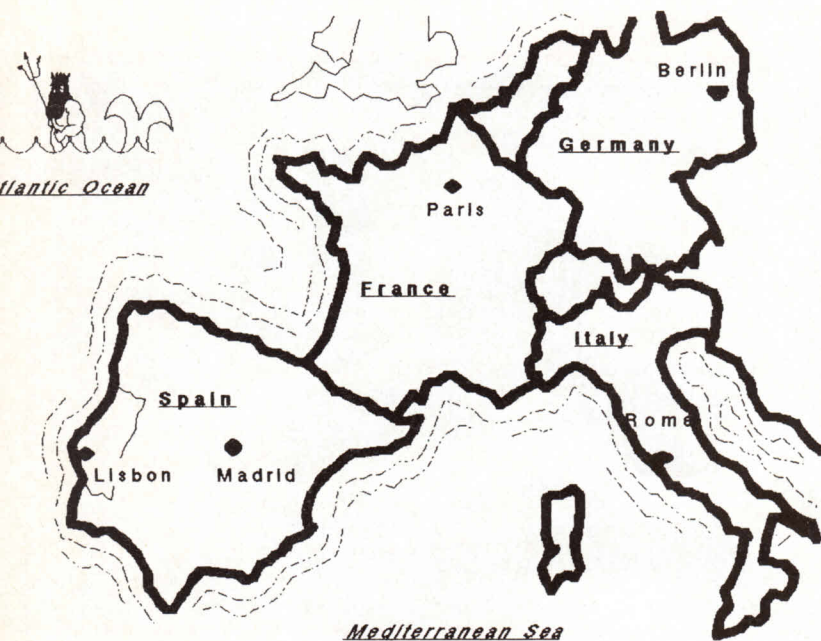


A pesar de estar dirigido al dibujo lineal, puede usarse con cierto éxito en el artístico.



El detalle con el que permite trabajar es una de las mayores ventajas del GEM Draw.

Europe



Los mapas pueden ser una de las aplicaciones más útiles del programa.

en modo de "IMAGEN". (Por cierto, si logran un dibujo bonito, y quieren que aparezca al arrancarse el ordenador, basta con salvarlo en este modo e incluir en el fichero AUTOEXEC.BAT que está en el disco de arranque, línea TYPE DIBUJO.EXT.)

Editar. Permite seleccionar una parte del dibujo, recuperar una parte borrada, duplicar una parte seleccionada previamente y agrupar todo el dibujo en una sola «selección».

Fuentes. Incorpora dos tipos de letras para los textos, eligiéndose dentro de éstas seis tamaños diferentes, así como el tipo de impresión, entre «negrita», «itálica», «subrayada» y normal.

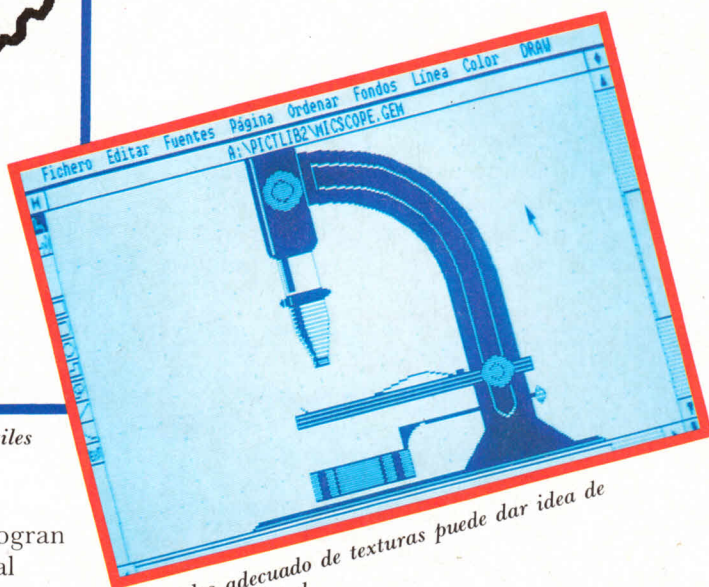
Página. Permite optar entre imagen normal, ampliada o reducida, y si se desea disponer de regla numerada, su escala, y si se desea dibujar sobre una trama o cuadrícula, el tamaño de ésta. También permite elegir si el dibujo se desea poner sobre el papel en forma horizontal o vertical, así como el tamaño de la hoja. Pero la parte fundamental es dar la medida de la hoja en la que se va a imprimir, dado que se puede

especificar para cada dibujo, pero no en la opción de impresión OUTPUT.

Ordenar. En caso de varias figuras, a través de este menú se pueden centrar en el papel, alinearlas a izquierda o derecha. Agrupar varias en una, desagruparlas, y repartirlas por el papel uniformemente. Para operar con cualquier comando, tienen que estar seleccionadas las partes de la imagen a ordenar.

Fondos. Treinta y nueve fondos de papel, ni más ni menos, se pueden elegir para dibujar encima de ellos. Se puede seleccionar un área para un determinado fondo, y cambiar de área para otro fondo, de manera ilimitada, como se ve, casi nada...

Líneas. Se puede elegir línea continua o tres tipos de línea discontinua, cuatro grosores distintos de cada tipo, y si se desea que las líneas señalen los vértices con cabeza en forma de flecha. De esta manera, se pueden resaltar las partes del dibujo que se deseen. Aunque es más aconsejable realizar primero todo el dibujo con



El empleo adecuado de texturas puede dar idea de tridimensionalidad.

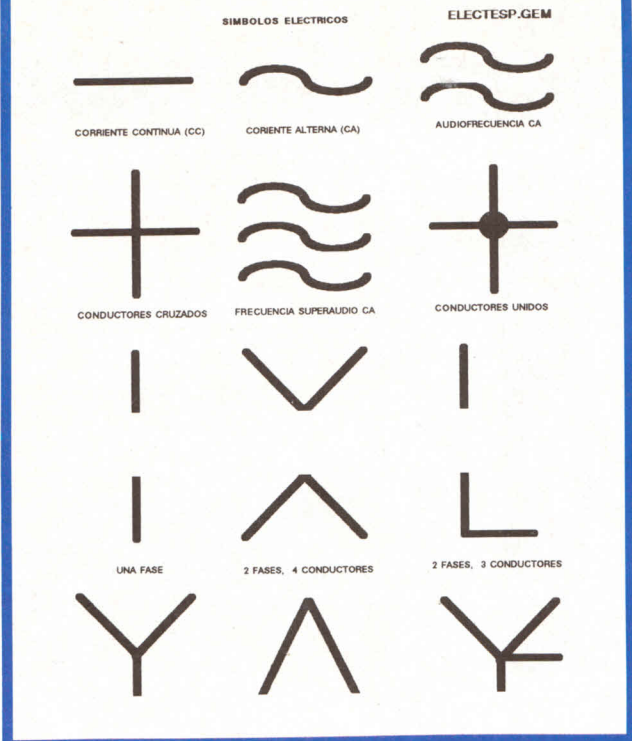
líneas finas, y posteriormente cambiar las líneas que se deseen.

Color. Hasta 16 colores se pueden combinar en pantalla para aumentar la vistosidad del dibujo, aunque naturalmente a la hora de imprimir esto resulte indiferente.

Complementos

GEM Draw proporciona (no se sabe todavía si incluido en el mismo paquete, o por separado), además, un disco de librería, donde ya vienen realizados multitud de diseños para utilizar en los dibujos. El *GEM Draw library* incluye 40 bordes diferentes para adornar los dibujos, así como un sinfín de símbolos, utilizados en electricidad,

Mundo del PC



Símbolos gráficos que forman una de las bibliotecas de GEM DRAW.

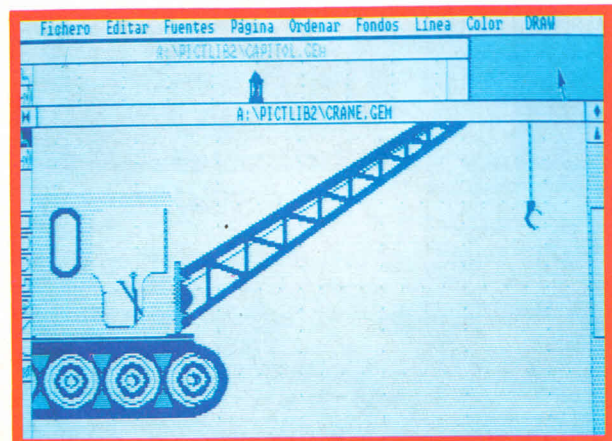
informática, empresa, etc.

Todos estos signos se pueden utilizar posteriormente en dibujos, mezclándolos, moviéndolos y cambiando su tamaño.

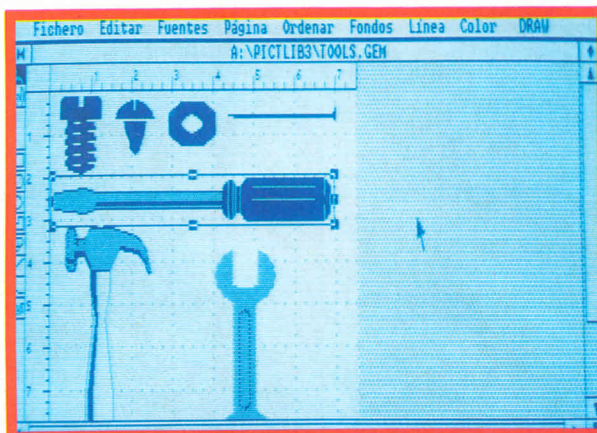
Este disquete viene, además, con instrucciones grabadas en disco, de manera que se pueden consultar por pantalla o imprimir a través del programa OUTPUT, quedando un pequeño manual de consulta francamente curioso y estéticamente perfecto.

Conclusión

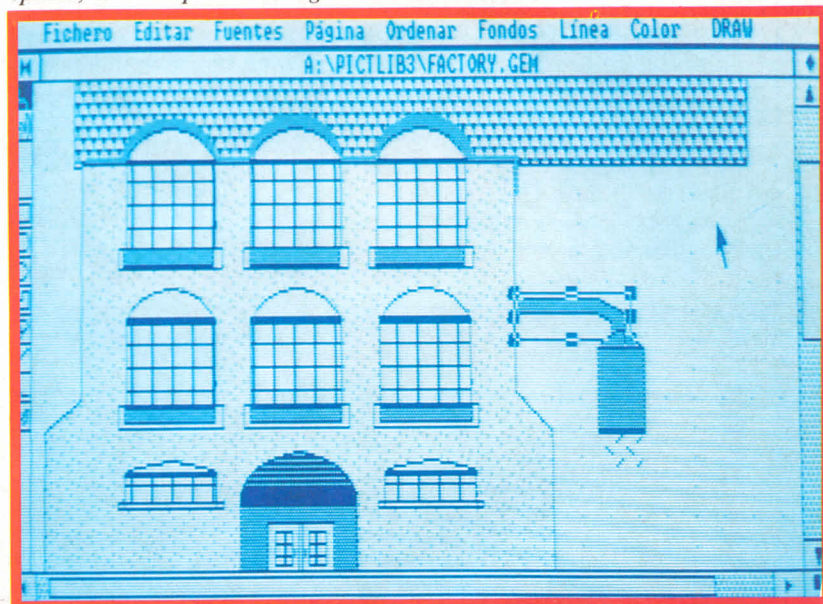
GEM Draw está claramente enfocado al dibujo lineal, y cumple con su cometido perfectamente. Probablemente, al principio resulte un poco difícil acostumbrarse al manejo del ratón, por lo que es



Poder manejar varias pantallas simultáneamente es uno de los mayores atractivos del GEM Draw.



En diseño industrial, a pesar de no ser la herramienta óptima, también pueden conseguirse buenos resultados.



Seleccionando una parte del dibujo, uno de sus elementos, podremos realizar en él sofisticadas modificaciones.

aconsejable realizar varias pruebas cambiando la sensibilidad del mismo a través del programa de instalación del GEM. Una vez superada esta dificultad, el manejo no ofrece complicaciones.

Desde luego, este programa hará que más de un estudiante de ingeniería o arquitectura realice las láminas de clase con esta herramienta, olvidando el tablero y la goma de borrar. Total, luego se le pasa el rotring...

Aunque quizá un poco «espartano» en algunas opciones, el nivel general del programa es muy bueno, y el límite de complicación de los dibujos que se pueden hacer se antoja muy alto. (Con Plotter ya que puede ser la *pera*...)

No podemos hablar del manual, dado que al ser ésta una versión preliminar, no nos ha sido facilitado todavía, pero tampoco es algo fundamental. Su manejo no presenta ningún problema, como todos los programas de GEM.

CON **ERBE** Software LOS MAYORES EXITOS D



Dos mundos, imagen paralela
al uno del otro, se comunican
en el espacio.

Uno es positivo, bueno, familiar; el otro
negativo, diabólico y desconocido.

Su unión, una ventana en el tiempo,
por lo cual los objetos y los seres
pueden pasar.

Tu misión, pasar de un mundo a otro
hasta recuperar las cosas que han
sido robadas y transportadas
al mundo oscuro.

DOUBLE DONBRE TAKE

ERBE
Software

ocean

DOUBLE TAKE

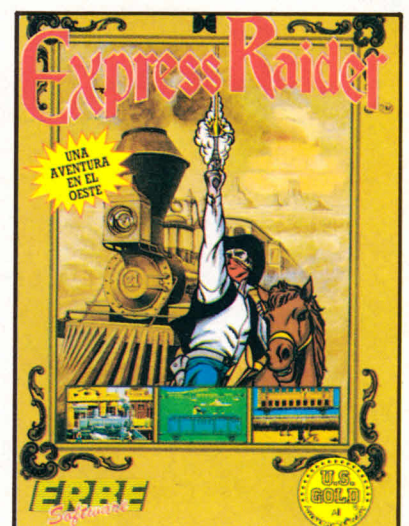
Dos mundos. Uno positivo y familiar; el otro, negativo, diabólico y desconocido. Su unión, un agujero en el tiempo y en el espacio a través del cual pasan los objetos. Tu misión es pasar de un mundo a otro mundo hasta recuperar las cosas que han sido robadas y transportadas al mundo oscuro... Pero date prisa, al pasar el tiempo todo será más difícil.



ERBE Software **martech**

UCHI-MATA

Basado en el judo, "Uchi-Mata", es probablemente el más avanzado y completo de los simuladores de artes marciales existentes. El juego ha sido diseñado bajo el asesoramiento de Brian Jacks, cinturón negro, 7.º Dan, cuatro veces campeón de Europa y Medalla Olímpica, uno de los mejores luchadores de judo del mundo.



ERBE
Software

WIG
GOLD
14

EXPRESS RAIDER

Como en las clásicas películas del Lejano Oeste, estarás en el centro de la acción desde el principio. Asaltos al tren, lucha sobre los vagones, cabalga sobre tu rápido caballo... Express Raider lo tiene todo.

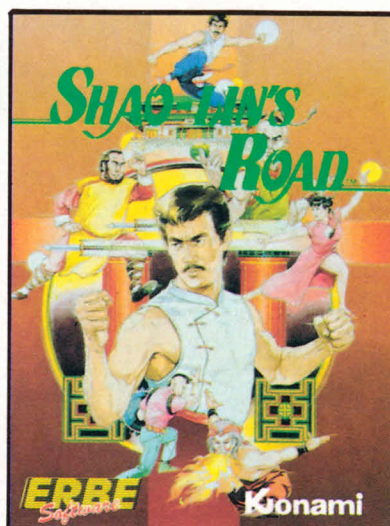


ERBE
Software

martech

PULSATOR

Un clásico juego de marcianos que te mantendrá pegado al ordenador durante horas. Cinco niveles con 49 pantallas cada uno... Una locura, ¿no?

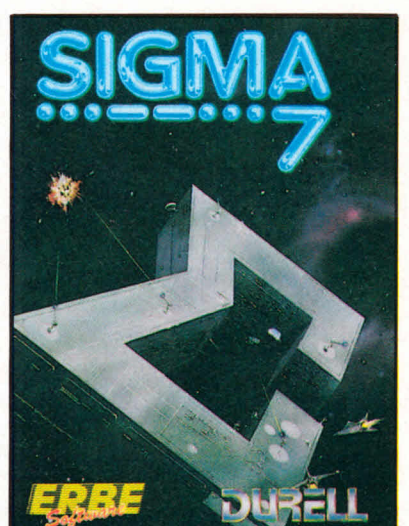


ERBE
Software

Konami

SHAO LINS ROAD

El juego de Konami de mayor éxito en las máquinas, en versión ahora para tu ordenador. Ayuda a Shao Lin a escapar de sus múltiples enemigos en su camino hacia la libertad. Salta, pega, agáchate y esquiva los puñales... Acción a tope.



ERBE
Software

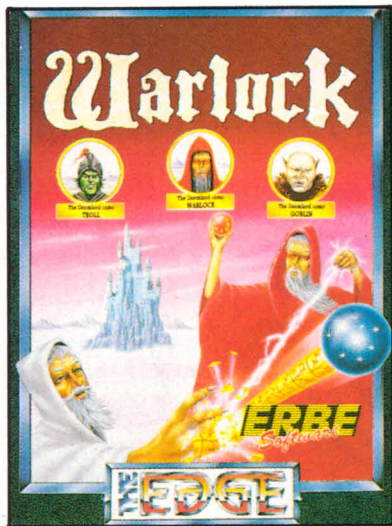
DURELL

SIGMA 7

Toda la acción de los juegos espaciales, en este fabuloso juego tridimensional. Un auténtico derroche de imaginación con siete niveles diferentes y totalmente distintos. Móntate en tu nave y abróchate el cinturón, el despegue es inminente.

EL MOMENTO POR SOLO

875 ptas.



WARLOCK

Encuentra el "Cristal del Poder" y combáte contra dragones, demonios y guerreros hasta encontrar al "Mago Blanco" para la lucha final. Quizá tengas en tu pantalla los más impresionantes gráficos en 3 dimensiones que hayas visto jamás.



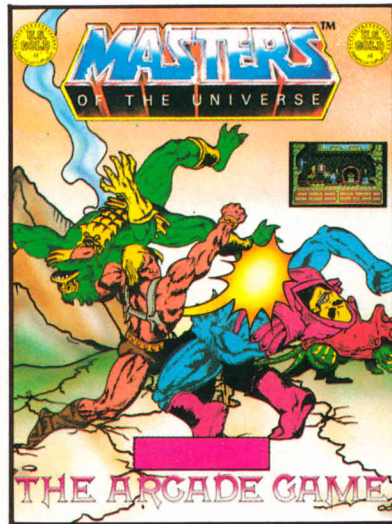
FIST II

La deseada continuación de "Exploding Fist" ya es una realidad; Fist II es mucho más que una segunda parte: más de 100 pantallas y 700 sprites diferentes, en las que los oponentes al Gran Maestro se esconden en bosques, acantilados, cavernas y lugares con paisajes que te asombrarán.



SHADOW SKIMMER

Un sorprendente súper-suave, súper-rápido juego de combate espacial que presenta increíbles pantallas llenas de colorido y acción como nunca antes habrás experimentado.



LOS MASTERS DEL UNIVERSO

Los héroes de los dibujos del T.V. con el célebre He-Man al frente van a luchar contra Skeletor, el Señor de las Fuerzas del Mal en una alucinante aventura en el castillo de Grayskull. Tú eres el protagonista.



CORTO CIRCUITO

Era una posibilidad entre un millón, pero el N.º 5, diseñado para ser un sistema de armamento dotado de inteligencia artificial, el más sofisticado robot del planeta, se escapó y se dio cuenta que estaba vivo. Ahora, el científico que le construyó quiere encontrarle y destruirle. Tú eres el N.º 5, no te dejes atrapar.



PALITRON

¡¡Es un juego insólito!! Por primera vez, un juego te permite no sólo guiar a su protagonista, un robot Servo D27, en su intento de localizar y destruir los cristales de un extraño mineral que está acabando con la vida en Palitron, si no que también te permite programar a otros robots para que te ayuden. Y todo con un color y unos gráficos extraordinarios.

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA:

ERBE SOFTWARE. C/. STA. ENGRACIA, 17 - 28010 MADRID. TELEF. (91) 447 34 10
DELEGACION BARCELONA. C/. VILADOMAT, 114 - TELEF. (93) 253 55 60.

Espía de gráficos

Por: Alberto Suñer

Hay determinados gráficos que merecen ser grabados. Con otros se podrían hacer unas cuantas modificaciones que los mejorarían. Incluso, ¿quién no ha soñado con introducir su personaje favorito en ese programa que tanto le divierte? El objetivo de este programa es, precisamente, ayudarle a buscar y «espíar» todos esos gráficos.

Existen muchos programas en el mercado, en los cuales aparecen unos gráficos verdaderamente bonitos, con movimientos tan perfectos que parecen otorgarles vida propia.

Estos gráficos han sido realizados por verdaderos artistas o bien han sido digitalizados de imágenes de vídeo o de dibujos extraídos de diversos cómics.

Ahora bien, nosotros únicamente podemos visualizarlos, cuando estamos ejecutando el programa, e incluso algunos de estos gráficos quizá no los podamos ver nunca, debido a que se encuentren en una fase del juego que sea de difícil acceso y a la cual haga falta mucho tiempo y habilidad para conseguir llegar.

En cualquier caso, aunque se puedan visualizar en alguna ocasión, siempre que deseemos observar dicho gráfico, o mostrárselo a alguien, deberemos cargar el juego, y llegar a la fase en que se encuentra el mismo.

Aparte de esta circunstancia, si queremos observar atentamente la secuencia de movimiento, nos será imposible, ya que no podremos aislar el gráfico a estudiar del entorno que lo rodea.

Por otra parte, quizá alguno de nosotros haya pensado alguna vez en almacenar los mejores gráficos que aparecen en los programas comerciales, por separado, para poderlos sacar por impresora, o bien únicamente para visualizarlos cada vez que lo deseemos.

Ahora bien, el problema que se nos plantea, es saber en qué dirección de memoria se encuentra cada uno de estos gráficos, para poderlos salvar en disco o cinta por separado.

Para evitar este problema, hoy os presentamos un programa que os permitirá revisar la memoria, mostrándonos en pantalla todo lo que en ella se encuentra, para de esta forma conseguir localizar todos y cada uno de los gráficos de que está compuesto cada juego.

Este programa consta de dos bloques, uno de ellos en Basic y otro en Código Máquina. El bloque en Basic nos permitirá cargar el bloque

que se desee visualizar, en memoria, y además nos permitirá salvar la ventana que se encuentre en pantalla en cada momento.

La rutina en Código Máquina, es la que se encarga de tomar el contenido de la memoria y pasarlo a pantalla, además de otras muchas opciones que se indicarán más adelante.

Utilización del programa

El funcionamiento es extremadamente sencillo. Además nos ofrece toda la información referente a la ventana que se encuentra en pantalla, en la parte inferior de la misma.

A continuación indicaremos cada una de las opciones posibles, y las teclas con las que se podrán ejecutar:

- Cursor arriba: incrementa la dirección de memoria tantos bytes como ancha sea la ventana.
- Cursor abajo: decrementa la dirección de memoria tantos bytes como ancha sea la ventana.
- Cursor derecha: incrementa la dirección de memoria.

PROGRAMA 1

```
1 REM ESPIA DE GRAFICOS
2 REM PROGRAMA 1
10 MODE 2: OPENOUT "D": MEMORY 999: CL
20 INPUT "DIRECCION DE LA MEMORIA A CARGAR EN MEMORIA (MAX. 39K):"; N$
30 LOAD N$, 1700
40 CALL 1000
50 CLEAR INPUT
60 MODE 2: INPUT "NOMBRE DEL GRAFICO A SALVAR:"; G$
70 POK=1643: GOSUB 100: DIR=DAT: POK=1
80 SAVE G$, B, DIR, LON
90 GOTO 40
100 DAT=PEEK(POK)+256*PEEK(POK+1): R
RETURN
```

ORDENADOR

PROGRAMA CARGADOR

```

1 REM ESPIA DE GRAFICOS
2 REM PROGRAMA CARGADOR
10 FOR N=&9000 TO &928A
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>58695 THEN PRINT "ERROR
  EN DATAS"
60 DATA 62,1,205,199,5,62,40
70 DATA 50,111,6,62,10,50,112
80 DATA 6,33,0,0,34,107,6
90 DATA 205,24,187,62,54,205,30
100 DATA 187,192,62,9,205,30,187
110 DATA 40,3,205,81,5,62,0
120 DATA 205,30,187,40,3,205,204
130 DATA 4,62,2,205,30,187,40
140 DATA 3,205,221,4,62,8,205
150 DATA 30,187,40,3,205,241,4
160 DATA 62,1,205,30,187,40,3
170 DATA 205,252,4,62,15,205,30
180 DATA 187,40,7,62,0,205,199
190 DATA 5,24,44,62,13,205,30
200 DATA 187,40,7,62,1,205,199
210 DATA 5,24,30,62,14,205,30
220 DATA 187,40,7,62,2,205,199
230 DATA 5,24,16,62,38,205,30
240 DATA 187,196,119,4,62,46,205
250 DATA 30,187,196,130,4,205,144
260 DATA 4,24,134,42,107,6,17
270 DATA 100,0,25,34,107,6,201
280 DATA 42,107,6,55,63,17,100
290 DATA 0,237,82,34,107,6,201
300 DATA 33,0,0,205,26,188,253
310 DATA 42,107,6,58,111,6,71
320 DATA 24,21,124,230,56,254,56
330 DATA 40,6,124,198,8,103,24
340 DATA 8,17,80,0,124,238,56
350 DATA 103,25,197,229,58,112,6
360 DATA 71,253,126,0,119,253,35
370 DATA 35,16,247,225,193,16,216
380 DATA 205,7,5,201,42,107,6
390 DATA 58,112,6,95,22,0,25
400 DATA 34,107,6,205,144,4,201
410 DATA 42,107,6,58,112,6,95
420 DATA 22,0,55,63,237,82,34
430 DATA 107,6,205,144,4,201,42
440 DATA 107,6,35,34,107,6,205
450 DATA 144,4,201,42,107,6,43
460 DATA 34,107,6,205,144,4,201
470 DATA 33,23,14,205,117,187,42
480 DATA 107,6,205,222,5,58,112
490 DATA 6,71,33,0,0,22,0
500 DATA 58,111,6,95,25,16,253

```

```

510 DATA 34,109,6,229,33,24,14
520 DATA 205,117,187,225,205,222,5
530 DATA 58,111,6,203,63,203,63
540 DATA 203,63,245,33,25,14,205
550 DATA 117,187,241,205,208,5,62
560 DATA 88,205,90,187,58,112,6
570 DATA 205,208,5,201,205,24,187
580 DATA 62,18,205,30,187,192,62
590 DATA 0,205,30,187,40,3,205
600 DATA 132,5,62,2,205,30,187
610 DATA 40,3,205,147,5,62,8
620 DATA 205,30,187,40,3,205,162
630 DATA 5,62,1,205,30,187,40
640 DATA 210,205,176,5,24,205,58
650 DATA 111,6,198,8,254,208,200
660 DATA 50,111,6,205,190,5,201
670 DATA 58,111,6,214,8,254,0
680 DATA 200,50,111,6,205,190,5
690 DATA 201,58,112,6,61,254,0
700 DATA 200,50,112,6,205,190,5
710 DATA 201,58,112,6,60,254,81
720 DATA 200,50,112,6,205,190,5
730 DATA 201,205,20,188,205,144,4
740 DATA 195,45,6,50,113,6,205
750 DATA 14,188,195,45,6,111,38
760 DATA 0,35,55,62,47,17,10
770 DATA 0,205,6,6,201,55,17
780 DATA 16,39,35,62,47,60,237
790 DATA 82,48,251,205,19,6,17
800 DATA 232,3,60,237,82,48,251
810 DATA 205,19,6,17,100,0,60
820 DATA 237,82,48,251,205,19,6
830 DATA 17,10,0,60,237,82,48
840 DATA 251,205,19,6,133,205,19
850 DATA 6,201,205,90,187,62,47
860 DATA 32,1,35,25,35,201,213
870 DATA 205,117,187,209,26,254,255
880 DATA 200,205,90,187,19,24,246
890 DATA 33,23,1,17,72,6,205
900 DATA 30,6,33,24,1,17,83
910 DATA 6,205,30,6,33,25,1
920 DATA 17,93,6,195,30,6,68
930 DATA 73,82,69,67,67,73,79
940 DATA 78,58,255,76,79,78,71
950 DATA 73,84,85,68,58,255,65
960 DATA 76,84,79,32,88,32,65
970 DATA 78,67,72,79,58,255,0
980 DATA 0,0,0,0,0,0,0

```



- Cursor izquierda: decrementa la dirección de memoria.
- M: incremento rápido de la dirección de memoria.
- N: decremento rápido de la dirección de memoria.
- B: retorna a Basic.
- F2: muestra la

ventana en modo 2.

- F1: muestra la ventana en modo 1.
- F0: muestra la ventana en modo 0.
- COPIA: modifica anchura y altura de la ventana.

Una vez dentro de la opción COPIA, tendremos las siguientes posibilidades:

Cursor arriba: aumenta la altura de la ventana.

Cursor abajo: decrementa la altura de la ventana.

Cursor derecha: aumenta la anchura de la ventana.

Cursor izquierda: decrementa la anchura de la ventana.

ENTER: abandono de esta opción.

Además, en la parte inferior de la pantalla, aparecerá información sobre la ventana que se está visualizando. Ésta nos indicará la dirección de memoria que se está visualizando, la anchura y altura de la ventana y la longitud que ocupa.

Dirección: 00000

Longitud: 00128

Alto x Ancho: 04 x 04.

Vamos a ver ahora cuál es el método a seguir para intentar localizar los gráficos que se encuentran en memoria.

En primer lugar, deberemos cargar el programa Basic, que se encargará de introducir en memoria la rutina en Código Máquina. Una vez ejecutado el programa, se nos pedirá el nombre del bloque que se desea visualizar en pantalla, y se cargará a partir de la dirección 1700.

Debemos tener en cuenta que la longitud del bloque a cargar en memoria, no debe sobrepasar los 39 kbytes, ya que de lo contrario no se admitirá la carga al sobrepasar la memoria disponible.

En este momento estamos en condiciones de buscar los gráficos que se encuentran en memoria y para ello nos ayudaremos de las teclas de cursor.

Se ha de tener en cuenta que, dado que los gráficos están almacenados en memoria en forma de bloques de dimensiones determinadas, nosotros deberemos ir modificando la altura y anchura de la ventana hasta que coincidan con la del gráfico en memoria, de lo contrario no podríamos distinguir ningún gráfico.

Debemos también fijarnos en el modo de

pantalla, ya que un gráfico realizado en un determinado modo, no es reconocible en un modo distinto.

Así pues, deberemos ir modificando tanto el modo de pantalla, como las dimensiones de la ventana, hasta localizar el gráfico deseado.

En un principio quizá nos parezca algo difícil conseguir localizar algún gráfico, pero una vez se haya utilizado varias veces este programa, nos resultará realmente muy fácil localizar cada uno de los gráficos que se encuentran en memoria.

Para hacer más sencilla la búsqueda de gráficos por primera vez, vamos a dar a continuación una serie de recomendaciones o consejos para habituarnos rápidamente a su utilización.

En primer lugar diremos que en cualquier programa, por lo general, los gráficos se encuentran ubicados en una misma zona. Cuando hayamos encontrado el primero, los restantes estarán a continuación de éste.

Si nos encontramos en esta situación, únicamente deberemos ir alterando las dimensiones de la ventana, para poder visualizar cada uno de los gráficos.

Cómo encontrar los gráficos

En caso de que nos resulte difícil encontrar el primer gráfico, deberemos dar una pasada rápida por toda la memoria utilizando la opción M o N, desechando todas aquellas zonas en las que se visualicen en la ventana formas sin ningún sentido.

Pasa por la zona donde se encuentra algún gráfico, aunque las dimensiones de la ventana no sean las mismas que las de éste, se visualizarán unos rasgos que nos indicarán si se trata de un gráfico, por lo que deberemos modificar las dimensiones de nuestra ventana.

En caso de que aún no se haya localizado ningún gráfico, deberemos ejecutar la misma operación, pero esta vez cambiando el modo de pantalla con cualquiera de las teclas correspondientes a dicha opción.

Una vez localizado el gráfico o gráficos deseados, podremos salvarlos en cinta o disco, pulsando la tecla B, con lo cual volveremos al Basic que nos pedirá el nombre que le queremos dar a dicha figura.

Debemos advertir, que aunque el programa se ejecute en pantalla en modo 1, los gráficos pueden estar almacenados en memoria en modo 2, por tanto deberemos revisar la memoria en cualquiera de dichos modos hasta encontrar el adecuado. Diremos también que por lo general si un programa se ejecuta en modo 0, los gráficos también se encontrarán en dicho modo.

Otra de las posibilidades que nos ofrece este programa, es averiguar las direcciones de memoria en que se encuentran cada uno de los gráficos.

Esto nos puede servir para incorporar nuestros propios gráficos a cualquier programa comercial, para

Gráficos por ORDENADOR

```

1 ;ESPIA DE GRAFICOS
10     ORG 1000
20     LD A,1
30     CALL CLS
40     LD A,40
50     LD (ALTO),A
60     LD A,10
70     LD (ANCHO),A
80     LD HL,0
90     LD (DIRMEM),HL
100  TECL: CALL #BB18
110     LD A,54
120     CALL #BB1E
130     RET NZ
140     LD A,9
150     CALL #BB1E
160     JR Z,NOTE
170     CALL GRADUA
180  NOTE: LD A,0
190     CALL #BB1E
200     JR Z,NOTE1
210     CALL UP
220  NOTE1: LD A,2
230     CALL #BB1E
240     JR Z,NOTE2
250     CALL DOWN
260  NOTE2: LD A,8
270     CALL #BB1E
280     JR Z,NOTES
290     CALL IZQ
300  NOTES: LD A,1
310     CALL #BB1E
320     JR Z,NOTE4
330     CALL DER
340  NOTE4: LD A,15
350     CALL #BB1E
360     JR Z,NOTES5
370     LD A,0
380     CALL CLS
390     JR PANTAL
400  NOTES5: LD A,13
410     CALL #BB1E
420     JR Z,NOTE6
430     LD A,1
440     CALL CLS
450     JR PANTAL
460  NOTE6: LD A,14
470     CALL #BB1E
480     JR Z,NOTE7
490     LD A,2
500     CALL CLS
510     JR PANTAL
520  NOTE7: LD A,38

```

```

530     CALL #BB1E
540     CALL NZ,ADMEM
550     LD A,46
560     CALL #BB1E
570     CALL NZ,SUBMEM
580  PANTAL: CALL PINTA
590     JR TECL
600  ADMEM: LD HL,(DIRMEM)
610     LD DE,100
620     ADD HL,DE
630     LD (DIRMEM),HL)
640     RET
650  SUBMEM: LD HL,(DIRMEM)
660     SCF
670     CCF
680     LD DE,100
690     SBC HL,DE
700     LD (DIRMEM),HL
710     RET
720  PINTA: LD HL,0
730     CALL #BC1A
740     LD IY,(DIRMEM)
750     LD A,(ALTO)
760     LD B,A
770     JR PINT
780  P_BUC: LD A,H
790     AND 56
800     CP 56
810     JR Z,P_PAS
820     LD A,H
830     ADD A,8
840     LD H,A
850     JR PINT
860  P_PAS: LD DE,0080
870     LD A,H
880     XOR 56
890     LD H,A
900     ADD HL,DE
910  PINT:  PUSH BC
920     PUSH HL
930     LD A,(ANCHO)
940     LD B,A
950  P_BUC1: LD A,(IY+0)
960     LD (HL),A

```

```

970     INC IY
980     INC HL
990     DJNZ P_BUC1
1000    POP HL
1010    POP BC
1020    DJNZ P_BUC
1030    CALL PINDIR
1040    RET
1050  UP:   LD HL,(DIRMEM)
1060     LD A,(ANCHO)
1070     LD E,A
1080     LD D,0
1090     ADD HL,DE
1100     LD (DIRMEM),HL
1110     CALL PINTA
1120     RET
1130  DOWN: LD HL,(DIRMEM)
1140     LD A,(ANCHO)
1150     LD E,A
1160     LD D,0
1170     SCF
1180     CCF
1190     SBC HL,DE
1200     LD (DIRMEM),HL
1210     CALL PINTA
1220     RET
1230  IZQ: LD HL,(DIRMEM)
1240     INC HL
1250     LD (DIRMEM),HL
1260     CALL PINTA
1270     RET
1280  DER: LD HL,(DIRMEM)
1290     DEC HL
1300     LD (DIRMEM),HL
1310     CALL PINTA
1320     RET
1330  PINDIR: LD HL,#OE17
1340     CALL #BB75
1350     LD HL,(DIRMEM)
1360     CALL PINUM
1370     LD A,(ANCHO)
1380     LD B,A
1390     LD HL,0
1400     LD D,0

```

poder, de esta forma, jugar a nuestra aventura preferida con un personaje creado por nosotros mismos.

Para poder hacer esto último, deberemos tener en cuenta que el nuevo gráfico sea de las mismas dimensiones que el que se desea sustituir.

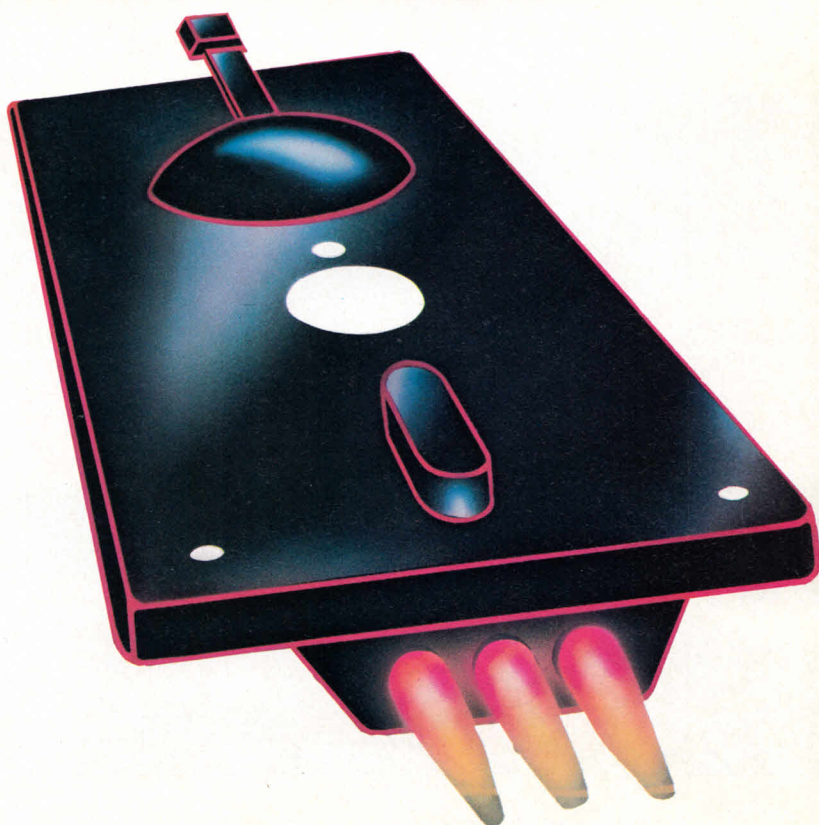
Otra posibilidad que existe es la de modificar las figuras de cualquier programa a nuestro antojo, y luego volverlas a cargar, con lo que el programa se ejecutará con los gráficos alterados.

Por último, sólo queda decir que para poder disfrutar con esta rutina, únicamente deberéis copiar el listado Basic, además del listado ensamblador, que aparece al final del artículo.

Aquellos que lo deseen pueden teclear el cargador Basic, y una vez ejecutado sin errores, salvar el código objeto de la forma que se indica:

```
SAVE" MIRBIN",B,1000,650
```

Para cargar el programa en memoria, únicamente deberemos ejecutar el programa Basic, el cual cargará el Código Máquina e inicializará el programa.



Gráficos por ORDENADOR

```

1410 LD A, (ALTO)
1420 LD E, A
1430 BUSUM: ADD HL, DE
1440 DJNZ BUSUM
1450 LD (LONG), HL
1460 PUSH HL
1470 LD HL, #0E18
1480 CALL #BB75
1490 POP HL
1500 CALL PINUM
1510 LD A, (ALTO)
1520 SRL A
1530 SRL A
1540 SRL A
1550 PUSH AF
1560 LD HL, #0E19
1570 CALL #BB75
1580 POP AF
1590 CALL PINU1
1600 LD A, "X"
1610 CALL #BB5A
1620 LD A, (ANCHO)
1630 CALL PINU1
1640 RET
1650 GRADUA: CALL #BB18
1660 LD A, 18
1670 CALL #BB1E
1680 RET NZ
1690 LD A, 0
1700 CALL #BB1E
1710 JR Z, NOTC
1720 CALL ARRI
1730 NOTC: LD A, 2
1740 CALL #BB1E
1750 JR Z, NOTC1
1760 CALL ABAJ
1770 NOTC1: LD A, 8
1780 CALL #BB1E
1790 JR Z, NOTC2
1800 CALL LEFT
1810 NOTC2: LD A, 1
1820 CALL #BB1E
1830 JR Z, GRADUA
1840 CALL RIGHT
1850 JR GRADUA
1860 ARRI: LD A, (ALTO)
1870 ADD A, 8
1880 CP 208

```

```

1890 RET Z
1900 LD (ALTO), A
1910 CALL REPON
1920 RET
1930 ABAJ: LD A, (ALTO)
1940 SUB 8
1950 CP 0
1960 RET Z
1970 LD (ALTO), A
1980 CALL REPON
1990 RET
2000 LEFT: LD A, (ANCHO)
2010 DEC A
2020 CP 0
2030 RET Z
2040 LD (ANCHO), A
2050 CALL REPON
2060 RET
2070 RIGHT: LD A, (ANCHO)
2080 INC A
2090 CP 81
2100 RET Z
2110 LD (ANCHO), A
2120 CALL REPON
2130 RET
2140 REPON: CALL #BC14
2150 CALL PINTA
2160 JP TEXTOS
2170
2180
2190 CLS: LD (MODO), A
2200 CALL #BC0E
2210 JP TEXTOS
2220 ;
2230 ;
2240 ; IMPRIME NUMEROS
DECIMALES
2250 ;
2260
2270 PINU1: LD L, A
2280 LD H, 0
2290 INC HL
2300 SCF
2310 LD A, 47
2320 LD DE, 10
2330 CALL DIEZ
2340 RET
2350
2360 ;
2370 PINUM: SCF
2380 LD DE, 10000
2390 INC HL
2400 LD A, 47
2410 DMIL: INC A

```

```

2420 SBC HL, DE
2430 JR NC, DMIL
2440 CALL PRINT
2450 LD DE, 1000
2460 MIL: INC A
2470 SBC HL, DE
2480 JR NC, MIL
2490 CALL PRINT
2500 LD DE, 100
2510 CIEN: INC A
2520 SBC HL, DE
2530 JR NC, CIEN
2540 CALL PRINT
2550 LD DE, 10
2560 DIEZ: INC A
2570 SBC HL, DE
2580 JR NC, DIEZ
2590 CALL PRINT
2600 ADD A, L
2610 CALL PRINT
2620 RET
2630 PRINT: CALL #BB5A
2640 LD A, 47
2650 JR NZ, PAS
2660 INC HL
2670 PAS: ADD HL, DE
2680 INC HL
2690 RET
2700
2710 PITEX: PUSH DE
2720 CALL #BB75
2730 POP DE
2740 PI001: LD A, (DE)
2750 CP 255
2760 RET Z
2770 CALL #BB5A
2780 INC DE
2790 JR PI001
2800
2810 TEXTOS: LD HL, #0117
2820 LD DE, TXT1
2830 CALL PITEX
2840 LD HL, #0118
2850 LD DE, TXT2
2860 CALL PITEX
2870 LD HL, #0119
2880 LD DE, TXT3
2890 JP PITEX
2900
2910
2920 TXT1: DEFM "DIRECCION:"
2930 DEFB 255
2940 TXT2: DEFM "LONGITUD:"
2950 DEFB 255
2960 TXT3: DEFM "ALTO X ANCHO:"
2970 DEFB 255
2980 *L+
2990 DIRMEM: DEFS 2
3000 LONG: DEFS 2
3010 ALTO: DEFS 1
3020 ANCHO: DEFS 1
3030 MODO: DEFS 1

```

FLANK BRUNO'S BOXING

de Soft Express
P.V.P.: 4.200 ptas.

El deporte del boxeo es probablemente el más medroso juego de contacto practicado hoy en día. Ejerce una influencia extraña, muchas veces de fascinación sádica, en bastantes de sus seguidores.

Combina la forma física, fuerza, coraje y entrenamiento para producir un deporte extremadamente excitante.

Ahora podemos experimentar el furioso deporte del boxeo sin poner en juego nuestra integridad física con Frank Bruno, uno de los más grandes boxeadores del mundo.

El objetivo del juego consiste en defenderse frente a ocho boxeadores de forma sucesiva, cada uno de ellos los más famosos del mundo.

Los estilos de lucha de cada uno de los boxeadores son diferentes. Cada púgil es un individuo más inteligente y peligroso que el anterior.

Los boxeadores con que sucesivamente nos enfrentaremos, son los siguientes:

1. El canadiense: posee la apariencia de un oso gris y la fuerza de un elefante.

2. Fling Long Chop: de nacionalidad japonesa, domina las artes marciales.

3. Andra Punchedov: es tan rápido, que se te sube a la cabeza antes que una copa de vodka.

4. El indígena africano: plantea más problemas que ningún otro. Tiene un temperamento que saca las mayores posibilidades que poseemos.

5. El francés Frenchi: puede parecer que es muy frío, suave y sofisticado, pero esta falsa fachada esconde una amenaza permanente,

Catálogo de SOFTWARE PCW

listo para hacernos ver las estrellas.

6. Raviolly Mafiosi: no es un hombre para meterse con él. Se sabe todos los trucos sucios y los utiliza sin ningún remordimiento.

7. El australiano Andy: se alimenta con una dieta de perro muerto de hambre. No siente ninguna lástima por su contrincante y no para de golpear.

8. Peter *el perfecto*: es de nacionalidad estadounidense, se trata del mundialmente famoso campeón del mundo, el más completo boxeador de la historia. Un contrincante de suma preparación con el que Bruno tendrá que utilizar todas sus habilidades para poder superarle.

Para defenderse del oponente, Bruno debe conseguir darle un golpe decisivo y derribarlo.

Trataremos de anular los golpes recibidos por el oponente y golpearle cuando su cuerpo o cabeza estén desprotegidos. Realizando fantásticos golpes repetidos, aumentarán los poderes pujilísticos de Bruno.

El indicador de K.O., se hace intermitente cuando el 100 por cien de los golpes asestados por Bruno recaen en su oponente. Esto le permite dar un gancho con la derecha y, finalmente, propinarle un fuerte golpe de abajo arriba.

La pantalla de juego se divide en dos partes. La mitad superior ofrece una

información completa de cómo se desarrolla el juego y la otra mitad se trata de una imagen en perspectiva del ring.

KNIFE

de Ofites Informática
P.V.P.: 8.848 ptas.

El *Knife* es un completo editor de sectores y modificador de ficheros, que le permitirá buscar y modificar información (bytes y caracteres) en cualquier lugar del disco.

La utilidad de la aplicación consiste en recuperar ficheros borrados, captar discos sector a sector, cambiar sectores de posición dentro del disco, comprender mejor la información presente en los directorios del disco, así como proteger nuestros discos contra la copia. Esta última posibilidad consiste en dañar las pistas de manera que el sistema no pueda leerlas.

De modo diferente a otros programas similares, viene en dos partes diferentes para dos necesidades distintas:

— *Knife*: se utiliza para modificar un fichero individual o aprender sobre la estructura del disco de CP/M y la forma en que se almacenan los ficheros en el disco.

— *Knife 2*: está dirigido para aquellos usuarios que precisen realizar modificaciones más radicales o deseen realizar cualquier operación rápidamente. Resumiendo, es una opción más potente pero menos «amistosa».

A. Comenzando con el Knife:

Tanto el *Knife* como el *Knife 2* están incluidos en el mismo disco, pero necesitan el CP/M 2.2 para arrancar. Una vez que hemos introducido el disco aparecerá en la porción superior de la pantalla un menú, con las siguientes opciones:

1. Almacén de ficheros en CP/M:

Es importante para que el programa sea realmente útil el completo conocimiento de la manera en que el CP/M almacena los ficheros en los discos para el **Amstrad**.

Los discos del **Amstrad** son formateados para tener nueve sectores de 512 bytes por pista, y como tiene cuarenta pistas, tenemos que cada cara de un disco puede almacenar 184.320 bytes, es decir, 180 K.

Las pistas se numeran del 0 al 39, mientras que los sectores van del 65 al 73. Sin embargo, el sistema CP/M utiliza sectores de 128 bytes más conocidos como *requisitos* o *sectores lógicos*. Éstos son los que saca el *Knife* en la pantalla.

Sin embargo, el *Knife 2* hace exactamente lo contrario, y se refiere siempre a sectores «físicos», es decir, agrupando consecutivamente cuatro sectores lógicos.

Los bloques son grupos de registros pertenecientes al mismo fichero. En el **Amstrad** hay ocho registros por bloque y cada bloque ocupa dos sectores físicos con la numeración continua. Los ficheros siempre ocupan un número entero de bloques.

El directorio del disco ocupa el bloque 0 y guarda un registro con todos los ficheros del disco y la dirección en la que pueden ser encontrados.

Cada entrada de directorio hace referencia sólo a dieciséis bloques (extensión). Si el fichero es demasiado largo para ser guardado en una extensión, el CP/M abre otra para él, creando una entrada de directorio idéntica. Estas extensiones no aparecerán necesariamente en relación con el directorio. Todo depende de dónde esté el espacio libre.

Se almacenan en bits no usados, datos como si el fichero estuviera colocado en los accesos de *leer/escibir* o en los de *sólo leer*, o si es un fichero directorio del sistema.

2. *Otros formatos de disco en el Amstrad:*

Catálogo de SOFTWARE PCW

Además del formato del sistema discutido anteriormente, está el formato *sólo para datos* que es igual, salvo que no hay pistas reservadas y que los sectores se numeran del 193 al 201.

El otro es el formato IBM, diseñado para ser compatible con los sistemas CP/M-86 utilizados en el IBM PC, con ocho sectores por pista, numerados del uno al ocho y una pista reservada.

3. *Examinar un disco:*

Esta opción permite avanzar o retroceder por las pistas y sectores del disco, copiar los registros por impresora, alterar el contenido de un sector, o buscar un patrón o cadena de texto en un disco.

4. *Mirar un fichero:*

El manejo de esta opción es similar a la anterior. Primero el programa nos da la oportunidad de seleccionar un área de usuario nueva, de forma que los registros del fichero se presenten en secuencia. Con ello, el usuario podrá utilizar las opciones descritas anteriormente para examinar su contenido.

5. *Recuperación de un fichero:*

Para restablecer un fichero borrado es preciso que en el disco no se escriban otros datos tras la utilización del comando ERA.

Es decir, que el CP/M no borra los sectores que ocupan los ficheros, sino que avisa en el directorio de que esos sectores están libres para ser utilizados. Por lo tanto, con un conocimiento suficiente del funcionamiento de dichos directorios y el *Knife* que

permite la lectura y modificación de sus bytes, pueden restaurarse ficheros borrados con anterioridad.

B. *Knife 2:*

Es un editor de discos alternativo al *KNIFE* que nos ofrece diferentes facilidades que nos acercan a la idea de una forma completamente diferente.

Los dos programas se complementan entre ellos, pero como ya hemos dicho anteriormente, la principal diferencia entre ellos estriba en que el *Knife 2* siempre se refiere a sectores físicos (es decir, de 512 bytes), en vez de a los sectores lógicos del CP/M.

1. *Examinar un disco:*

Cuando se ha seleccionado un disco satisfactoriamente, se visualizarán los 256 bytes del primer sector en la primera pista del disco.

En primer lugar se visualiza su numeración hexadecimal. A continuación la representación, también hexadecimal, del contenido de los dieciséis bytes, y después las ASCII (caracteres), que son representaciones de cada uno de los bytes. Dicha representación aparecerá en pantalla siempre que sean caracteres imprevisibles, es decir, con un código de valor mayor que 32. Si no, aparece en su lugar un punto.

En este momento podrá alterar los bytes visualizados, pasar a la segunda mitad del sector o acceder a los comandos.

2. *Alteración de bytes:*

Basta con escribir el nuevo valor cuando el cursor está encima de la representación hexadecimal del byte a alterar, o de su representación ASCII.

Las modificaciones no quedan escritas en el disco hasta usar la opción *sector escribir*, por lo que puede practicar un poco con el sistema hasta que se acostumbre a él, sin miedo a causar un daño irreparable.

3. *Movimiento por los sectores:*

El programa tiene opciones de avanzar y retroceder por

sectores y pistas. Se puede pasar directamente del primer sector o pista al último y viceversa.

Presionando COPIA se visualiza la otra mitad del sector sin afectar a nada más.

4. *Búsqueda de una cadena:*

Produce una petición en la parte inferior de la pantalla. En esta pregunta deben ser escritos hasta un número de ochenta caracteres. Entonces, el programa comienza a buscar la cadena desde el sector en el que se ha pedido, hasta que la encuentre.

Cuando encuentra la cadena, el sector y la pista en que ha ocurrido esto, se convierten en los actuales, apareciendo en la pantalla. Si la cadena no se encuentra aparecerá un mensaje de aviso.

5. *Búsqueda de una secuencia de bytes:*

Esta opción es similar a la anterior, pero en este caso hay que introducir hasta ochenta bytes, con los valores entre 0 y 255.

El programa supone que cada byte debe ser introducido en hexadecimal, precediéndolo con un carácter PT.

6. *Escribir sector para dar la posición:*

El interés de esta opción consiste en escribir el sector editado anteriormente en cualquier posición del disco.

El programa nos pide una pista y sector nuevos, donde escribe el sector (de 512 bytes) en esa posición del disco.

Los usuarios sin experiencia tendrán que tener mucho cuidado con esta orden, porque es muy fácil dañar el contenido del disco si el sector ha sido escrito en una posición no deseada.

Así mismo, otra aplicación de la presente opción consiste en copiar discos sector a sector, si bien, ello no es recomendable para «piratear» debido a la lentitud del proceso.

7. *Seleccionar un sector nuevo:*

Presionando esta opción, el ordenador nos pedirá un nuevo

número de sector para ser editado.

Solamente nos permite seleccionar sectores en la pista actual.

8. *Seleccionar una pista nueva:*

Es igual a la anterior, sólo que esta vez nos pide el número de pista.

Esta orden no altera el número del sector actual de la pista.

9. *Volver a escribir el sector:*

Escribe de nuevo los 512 bytes que han sido editados, y los coloca en la posición de origen del sector una vez que lo hemos modificado.

- Emitir extractos de una o varias cuentas por fechas, dando los totales de entradas, salidas y saldos.

- Balance de sumas y saldos.

3. *Listados del programa:*

- Listado del Plan Contable.
- Listado del Libro Diario.
- Listado de apuntes de la cuenta.
- Listado de apuntes por periodos.
- Listado de sumas y saldos.

CONTABILIDAD E.O.S.

de Leo Computer
P.V.P.: 16.250 ptas.

Leos Eos está pensado para llevar la contabilidad de pequeñas empresas, profesiones liberales o usuarios en general.

Es un programa realizado en CP/M 3.0 dirigido a aquellas personas que en cualquier momento deseen conocer su situación económica, así como una previsión de cobros y pagos.

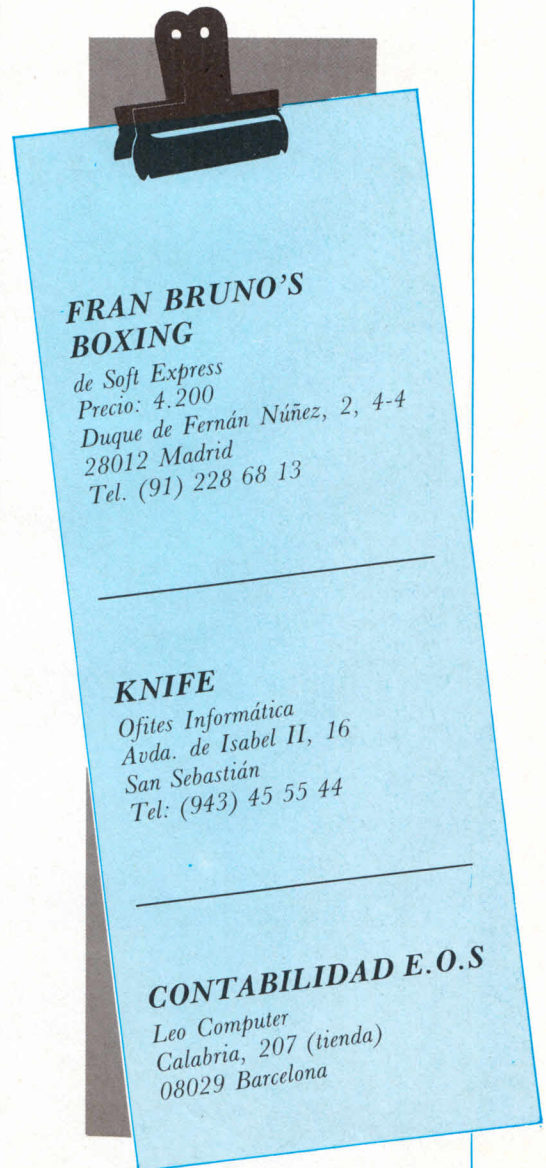
1. *Características:*

- Número máximo de cuentas: mil cuentas.
- Número mínimo de apuntes en PCW 8256: 1.900 apuntes.
- Número mínimo de apuntes en PCW 8512: 10.000 apuntes.

Permite en su proceso de instalación definir la posible utilización de una segunda impresora.

2. *Opciones:*

- Previsión de vencimientos de apuntes, bien para pago o cobro.
- Clasificar automáticamente por fechas los apuntes introducidos.



Listín telefónico (y II)

Como prometimos la semana pasada, aquí está la segunda y última parte del listado de este programa para el PCW, el cual nos hemos visto obligados a fraccionar dada su excesiva longitud, con idea de facilitar al máximo su tecleo a nuestros lectores. Nunca dejará de sorprendernos la increíble variedad de aplicaciones que los usuarios de la, según **Amstrad**, «máquina de escribir» son capaces de encontrar para su

ordenador. En cuanto al viejo argumento de que se tarda mucho menos en encontrar un número telefónico en una agenda que en un ordenador, bien, es cierto hasta un cierto número de teléfonos. ¿Se imaginan ustedes a Información de la propecta Telefónica diciendo: «Un momento. Voy a mirar la agenda. Mientras espera, le ofrecemos esta saludísima conferencia explicando el porqué de los monopolios»? ¿Eh?

```
360 letra$="CcNn":GOSUB 1080:IF respuesta=3 OR respuesta=4 THEN fichas=0:RETURN
370 GOSUB 1030:GOTO 330

380 REM          ***** leer fichas *****

390 PRINT FNat$(29,0)"Leyendo el listin telefonico. Por favor, espere.
      "
400 OPEN "I",1,"telefono.sec"
410 fichas=0
420 WHILE NOT(EOF(1))
430 fichas=fichas+1
440 INPUT R1,nombre$(fichas),telefono$(fichas),direccion$(fichas),pref$(fichas)
450 WEND
460 CLOSE 1
470 RETURN

480 REM          ***** buscar *****

490 sw=0
500 PRINT FNat$(2,68)" OPCION (B) BUSCAR "
510 IF fichas=0 THEN PRINT FNat$(29,0)";El listin telefonico esta vacio!";STRING
$(45," "):RETURN
520 PRINT;inv$;FNat$(29,0)"Para salir..... < RETURN >";TAB(80);norvi$;FNat$(3
,4)"Nombre..";FNat$(4,13);STRING$(30,"_");
530 PRINT FNat$(4,13);:INPUT">";busca$
540 busca$=UPPER$(busca$)
550 IF busca$="" THEN RETURN
560 cuenta = 1
570 a=INSTR(nombre$(cuenta),busca$)
580 WHILE cuenta <=fichas AND a=0:cuenta = cuenta +1:GOTO 570
590 WEND
600 IF sw=1 AND cuenta>fichas THEN dato=0: RETURN
610 IF cuenta> fichas THEN PRINT pit$;inv$;FNat$(4,25);"No encuentro ";busca$;r
orvi$:RETURN
620 dato =dato+1:IF dato>22 THEN sw=0:dato=0:GOSUB 1160:FOR j=3 TO 26:PRINT FNat
$(j,0);STRING$(90," "):NEXT ELSE PRINT FNat$(dato+5,
0);nombre$(cuenta);" C/ ";direccion$(cuenta):sw=1
630 PRINT FNat$(dato+5,78)>("pref$(cuenta);" " );telefono$(cuenta):cuenta=cuenta+
1:GOTO 570
640 dato=0:RETURN

650 REM * * * añadir ficha * * * *

660 PRINT inv$;FNat$(2,0)"Fichas"fichas;norvi$:PRINT FNat$(2,68)" OPCION (A) AÑA
DIR "
670 PRINT inv$;FNat$(29,0)"Solo 35 caracteres. Para salir < RETURN >";norvi$
680 PRINT FNat$(4,13) STRING$(35," ")
690 PRINT FNat$(6,14) STRING$(50," ")
700 PRINT FNat$(8,14) STRING$(10," ")
710 PRINT FNat$(10,14);STRING$(5," ")
720 PRINT FNat$(4,13) STRING$(35,"_")
730 PRINT FNat$(4,4)"Nombre..";:INPUT">";nombre$
740 nombre$= UPPER$(nombre$)
750 IF nombre$="" THEN RETURN
```

```

760 PRINT FNat$(6,4) STRING$(50,"_");FNat$(6,4);"Direccion.":;INPUT direccion$
770 direccion$= UPPER$(direccion$)
780 IF direccion$="" THEN RETURN
790 PRINT FNat$(8,4) STRING$(17,"_"); FNat$(8,4)"Telefono.":;INPUT;telefono$
800 IF telefono$="" THEN RETURN
810 PRINT FNat$(10,4)"Prefijo.____";FNat$(10,14);;INPUT pref$
820 IF pref$="" THEN RETURN
830 pref$(fichas)=pref$
840 fichas=fichas+1
850 nombre$(fichas)=nombre$
860 direccion$(fichas)=direccion$
870 telefono$(fichas)=telefono$
880 pref$(fichas)=pref$
890 fichero.cambiado=verdadero
900 PRINT inv$;FNat$(29,0)"Esta bien asi (S/N)...Terminar (F)in
    ";:letra$="SsNnFf":GOSUB 1080
910 ON respuesta GOTO 650,650,920,920,970,970
920 ficha = ficha-1:GOTO 650
930 GOTO 900
940 RETURN

950 REM          * * * escribir fichero * * *

960 PRINT FNat$(28,0)"Escribiendo el listin telefonico. Por favor, espere un mom
ento solamente.          "
970 OPEN"O",1,"telefono.sec"
980 FOR cuenta=1 TO fichas
990 WRITE R1,nombre$(cuenta),telefono$(cuenta),direccion$(cuenta),pref$(cuenta)
1000 NEXT
1010 CLOSE 1
1020 RETURN

1030 REM          * * * cambio de disco * * *

1040 RESET
1050 PRINT FNat$(29,0)"Ponga el disco donde tiene el listin telefonico y pulse R
ETURN",a$
1060 WHILE INKEY$="":WEND
1070 RETURN

1080 REM          * * leer tecla * *

1090 respuesta$=INKEY$
1100 WHILE respuesta$=""
1110 respuesta$=INKEY$
1120 WEND
1130 respuesta=INSTR(letra$,respuesta$)
1140 IF respuesta=0 THEN GOTO 1090
1150 RETURN

1160 REM          * * * esperar por tecla * * *

1170 PRINT inv$;FNat$(28,0);STRING$(90," ");FNat$(29,0);"Pulse cualquier tecla p
ara continuar";FNat$(27,0);norvis$
1180 WHILE INKEY$=""
1190 WEND
1200 RETURN

1210 REM          **** Pasar el listin a la impresora ****

1220 opcis$= "%%%%%%%%%..... O P C I O N (I) ..... L I S T A D O ...
..... %%%%%%%%%%"
1230 PRINT inv$;borra$:PRINT opcis$;norvis$
1240 PRINT FNat$(10,30)"(I).....Pantalla e IMPRESORA          ";FNat$(12,30
);"(P).....Solo en PANTALLA          "
1250 letra$="IiPp":GOSUB 1080:impr = respuesta
1260 PRINT FNat$(3,30)"S U B M E N U L I S T A D O"
1270 PRINT FNat$(10,30)"(N).....Ordenado por NOMBRES";FNat$(12,30)"(A).....C
rdenado por APELLIDOS"
1280 PRINT FNat$(16,30)"Elige...pulsa... (A / N)"
1290 letra$="NnAa":GOSUB 1080:salto=respuesta
1300 ON salto GOSUB 1730,1730,1790,1790
1310 PRINT borra$; FNat$(0,34)"LISTIN por ";nota$

```

```

1320 IF impr=3 OR impr=4 THEN 1340
1330 LPRINT:LPRINT FNat$(0,34)"LISTIN por ";nota$:LPRINT:LPRINT
1340 ff=1:FOR f=1 TO fichas
1350 punt=LEN (nombre$(f))
1360 PRINT FNat$(ff,0);STRING$(90,".");FNat$(ff+1,0)nombre$(f);FNat$(ff+1,32);di
reccion$(f);FNat$(ff+1,76)"(";pref$(f);)";
1370 PRINT FNat$(ff+1,82)telefono$(f)
1380 IF impr=3 OR impr=4 THEN 1400
1390 LPRINT CHR$(15);CHR$(27)"4"; nombre$(f) " ";direccion$(f) " ";pref$(f) " ";
telefono$(f)
1400 ff=ff+1
1410 IF ff>=28 THEN GOSUB 1160:ff=0:PRINT borras$
1420 NEXT
1430 GOSUB 1160:GOTO 80

1440 REM          ***** modificar fichas *****

1450 gesto=0:PRINT FNat$(2,68)" OPCION (M) MODIFI "
1460 PRINT inv$;FNat$(28,0)" <EXTRA>+<H>+<RETURN> para AYUDA - Dejar los
datos como estan < RETURN > ";norvi$
1465 PRINT FNat$(4,29);STRING$(60," ")
1470 PRINT FNat$(4,22)"Nombre.";
1480 PRINT FNat$(4,31);:INPUT">";busca$
1490 busca$=UPPER$(busca$)
1500 IF busca$="" THEN RETURN
1510 IF busca$=CHR$(253) THEN GOSUB 1960:IF nota =1 THEN 80
1520 IF gesto=1 THEN GOTO 1570
1530 FOR f=1 TO fichas
1540 IF LEFT$(busca$,15)=LEFT$(nombre$(f),15) THEN 1570
1550 NEXT
1560 IF f>fichas THEN PRINT FNat$(4,50)"NO HE ENCONTRADO "busca$:PRINT CHR$(7):F
OR f=1 TO 20:PRINT pit$: NEXT:GOTO 1440
1570 PRINT FNat$(6,10)"ficha antigua .-";FNat$(8,0)nombre$(f);FNat$(8,35)"C/ "di
reccion$(f);FNat$(8,77)"("pref$(f)") "
1580 PRINT FNat$(8,83)telefono$(f)
1590 PRINT FNat$(10,10)"Introduce la nueva ficha";FNat$(12,22)"Nombre.-";:INPUT
nomb$:nomb$=UPPER$(nomb$)
1600 IF nomb$="" THEN PRINT FNat$(12,32) nombre$(f):nomb$=nombre$(f)
1610 PRINT FNat$(14,22)"domicilio";:INPUT domi$:domi$=UPPER$(domi$)
1620 IF domi$="" THEN PRINT FNat$(14,33) direccion$(f): domi$=direccion$(f)
1630 PRINT FNat$(16,22)"prefijo";:INPUT prf$:prf$=UPPER$(prf$):IF prf$="" THEN P
RINT FNat$(16,32)pref$(f): prf$=pref$(f)
1640 PRINT FNat$(18,22)"Telefono";:INPUT telef$:telef$=UPPER$(telef$):IF telef$=
"" THEN PRINT FNat$(18,32) telefono$(f): telef$=telef
ono$(f)
1650 nombre$(f)=nomb$:telefono$(f)=telef$:direccion$(f)=domi$:pref$(f)=prf$
1660 PRINT FNat$(20,1)"Ficha modificada .-";FNat$(22,0)nombre$(f)" C/":PRINT FN
at$(22,35)direccion$(f);FNat$(22,77)"("";
1670 PRINT FNat$(22,78)pref$(f)".";FNat$(22,83)telefono$(f)
1680 PRINT inv$;FNat$(26,22)"Modificacion correcta (S/N)";norvi$;:letra$="SsNn":
GOSUB 1080
1690 ON respuesta GOTO 1710,1710,1440,1440
1700 IF INKEY$="N" OR INKEY$="n" THEN 1440
1710 IF nota=1 THEN 80
1720 GOSUB 970:GOTO 80

1730 REM          **** ordenar por nombres ****

1740 PRINT borras$"Ordenando por nombres.....espera un momento por favor
":PRINT:PRINT
1750 FOR f=1 TO fichas-1
1760 IF nombre$(f)>nombre$(f+1) THEN SWAP nombre$(f+1),nombre$(f):SWAP telefonos
(f+1),telefono$(f):SWAP direccion$(f+1),direccion$(f
):SWAP pref$(f+1),pref$(f):leo=1
1770 NEXT f:IF leo=1 THEN leo=0:GOTO 1750
1780 nota$="NOMBRES":RETURN
1790 REM * * * * * Ordenar por apellidos * * * * *

1800 PRINT borras$"Ordenando por apellidos.....espera un momento por favo
r":PRINT:PRINT
1810 FOR f=1 TO fichas-1
1820 long=LEN (nombre$(f)):long1=LEN (nombre$(f+1))

```

```

1830 sitio=INSTR(nombre$(f)," "):sitio1=INSTR(nombre$(f+1)," ")
1840 IF MID$(nombre$(f),sitio+1,long-siti+1)>MID$(nombre$(f+1),sitio1+1,long1-si
tuo1+1) THEN SWAP nombre$(f+1),nombre$(f):SWAP telef
ono$(f+1),telefono$(f):leo=1 ELSE GOTO 1860
1850 SWAP direccion$(f+1),direccion$(f):SWAP pref$(f+1),pref$(f)
1860 NEXT:IF leo=1 THEN leo=0:GOTO 1810

1870 FOR f=1 TO fichas-1
1880 long=LEN (nombre$(f)):long1=LEN (nombre$(f+1))
1890 sitio=INSTR(nombre$(f)," "):sitio1=INSTR(nombre$(f+1)," ")
1900 IF MID$(nombre$(f),sitio+1,long-siti+1)=MID$(nombre$(f+1),sitio1+1,long1-si
tuo1+1) AND LEFT$(nombre$(f),sitio)>LEFT$(nombre$(f+
1),sitio1) THEN SWAP nombre$(f+1),nombre$(f):SWAP telefono$(f+1),telefono$(f):le
o=1 ELSE GOTO 1920
1910 SWAP direccion$(f+1),direccion$(f):SWAP pref$(f+1),pref$(f)
1920 NEXT:IF leo=1 THEN leo=0:GOTO 1810
1930 nota$="APELLIDOS":RETURN
1940 PRINT MID$(nombre$(f),sitio+1,long-sitio+1),MID$(nombre$(f+1),sitio+1,long-
sitio+1)
1950 IF INKEY$="" THEN 1950

1960 REM          ***** subrutina de ayuda *****

1970 PRINT inv$ FNat$(28,0);"   Cuando encuentres la ficha que buscas puls
a ..... < C >          "

1980 nota =0:FOR f =1 TO fichas
1990 PRINT inv$;FNat$(4,33);f;" "; nombre$(f);norvis$
2000 ink$=INKEY$:WHILE ink$=""
2010 ink$=INKEY$: WEND
2020 IF ink$="c" OR ink$="C" THEN gesto=1: RETURN
2030 PRINT FNat$(4,33);STRING$(40," "):NEXT
2040 nota =1:RETURN
2050 fichero.cambiado=verdadero:RETURN

```

Correo..., más rápido...



Con el fin de acelerar lo más posible el **correo**, y poder resolver o contestar a todas las dudas y sugerencias que llegan a nuestra redacción, a partir de esta semana os rogamos, en beneficio de todos, consignar en el sobre, en lugar bien visible, una de las denominaciones siguientes:

- **Suscripciones AMSTRAD.** Para todos aquellos casos relacionados con petición de cintas, números atrasados, formalización de suscripciones, devoluciones, etc...
- **Mercado Común AMSTRAD.** Compras, ventas, intercambios, clubs...
- **Serie Oro AMSTRAD.** Para los programas que nos enviéis para su publicación.
- **Sugerencias AMSTRAD.** Para vuestras críticas, sugerencias o cualquier opinión que queráis vertir sobre la revista.

Es necesario revisar los listados con atención

Tengo un CPC 6128. Leyendo una de vuestras revistas, exactamente la número 74, me decidí a teclear el programa que se publicaba en la *Serie Oro*. Este programa es *Ice-Front*. Resulta que después de haberlo teclado cuesta mover a *Bob*. Además, cuando se acaban las vidas o pasa un tiempo determinado y la pantalla se borra, aparece inmediatamente o *Bob en el hielo* o *Bob en sandwich*, tras lo cual aparece:

Syntax error in 1540

1540 data 3e, 01, 21, 00, 8c, 11, a0, 00, 19, 3d, fe, 00, 20, fa, 11, 60, 9a, 01, a0, 00, ed, a0, e2, f8, 9b, c3, f0, 9b, c9.

Yo, la verdad, no sé dónde se encuentra el error y preguntándoles a algunos amigos, ellos tampoco lo saben. Así pues, os agradeceríamos que nos dijeseis dónde se encuentra el fallo, ya que no conseguimos verlo.

J. Enrique García Prieto

Lamentablemente, con los datos que nos da no es posible decir concretamente dónde está el error, ya que la línea de listado que nos envía está correcta. Creemos que puede haber pasado lo siguiente: el «despiste» se encuentra en otra parte completamente distinta del programa, pero en la zona de las datas del Código Máquina. Ocurre que el Amstrad, al intentar ejecutar una subrutina en máquina que contiene un error, «enloquece» y actúa de forma impredecible. Da la casualidad de que lo que sea que pase afecta al propio intérprete de Basic, y da un mensaje de error de sintaxis donde no existe ninguno. Nos vemos obligados a recomendarle que repase con esmerada atención el listado, especialmente la zona que se refiere a las datas, comprobando con especial cuidado que no ha puesto una «0» en lugar de un «0».

A vueltas con los pokes

1. Me gustaría saber cómo se introducen los pokes para el Amstrad.
2. He observado que en



Sin duda ALGUNAS

algunos programas comerciales en el Amstrad se consiguen apreciar más de cuatro colores, cosa que a mí me resulta imposible.

Álvaro Sanz Monasterio

Para introducir los pokes en los juegos comerciales, salvo que no estén protegidos, necesita un programa cargador que normalmente facilitamos en la revista cuando se habla de un juego con detalle.

En una consulta de estas mismas páginas encontrará una respuesta más detallada.

Efectivamente, en algunos juegos comerciales se ven más de cuatro colores en modo 1. Habrá usted observado que sólo sucede en determinadas zonas de la pantalla. El motivo es que los programadores engañan al Amstrad haciéndolo creer que una parte de la pantalla está en modo 1 y otra en un modo distinto.

Explicar concretamente el asunto está fuera del alcance de estas páginas, pero el método en sí recurre al uso de rutinas de interrupción; es decir, el ordenador conmuta entre dos rutinas de color y modo de pantalla tan rápidamente que parece que ambos modos están simultáneamente en el monitor.

File not found

Poseo un Amstrad CPC-464 con unidad de disco y he realizado algún que otro programa de acceso a ficheros ubicados en este dispositivo, encontrándome con el problema de que, al no existir dichos ficheros, el programa se interrumpía dando el mensaje: *File Not Found*.

De aquí mi pregunta: ¿no habría manera de controlar dentro de los programas los

errores de este tipo? (Tal cual lo hace el Amstrad 6128.)

Os he llamado por teléfono para preguntaros el tema que os explico en el punto anterior, pero me indicasteis que era más conveniente que os lo planteara por escrito, debido a la extensión del problema.

En el manual que se entrega con la unidad DD-1, en el capítulo 5.8 hay una relación de todos los errores, pero no logro entenderlos. Os quedaría agradecido que me explicara cómo podría acceder al registro A para poder sacar los mensajes por pantalla en español; por ejemplo, que no está en el fichero, o el disco no está en la unidad, o error de escritura, etc..., y así evitar la rotura del programa.

Otro tema que quiero comentaros es el siguiente:

Tengo un programa *Menú* (que os he enviado para su posible publicación), en donde una de las líneas del mismo enlaza con el procesador de textos que en su día publicasteis, pero que al terminar de utilizar este proceso y al intentar volver al menú, da un mensaje de MEMORY FULL.

La consulta que os quiero hacer es si hay algún modo de evitar que se llene la memoria al pasar de nuevo al *Menú* (sin desconectar el ordenador) y si es posible que me expliquéis el porqué de este mensaje.

Juan José Ferrer Pérez

Desgraciadamente, no hay solución para su primera pregunta. Amstrad omitió tratar los errores del disco en el 464, incluso aunque usted posea una unidad de disco. Por tanto, o se olvida del asunto, o adquiere un 6128. Lo lamentamos mucho.

Su segunda pregunta tiene una respuesta más clara y es, sobre todo, positiva. Sucede que nuestro procesador de texto y su programa Menú no caben simultáneamente en la memoria. Para conseguirlo, debe usar la orden CHAIN o CHAIN MERGE, cuya sintaxis y uso están explicadas en el manual.

Existe una precaución que debe tomar en cualquier caso. Cuando la rutina que usted llamó acaba su cometido, es esencial que retorne al programa Menú, por lo que hay que incluir otra orden CHAIN, que invoque al programa principal.

Actualidad, pokes, mapas, trucos,
los mejores juegos y programas para
SPECTRUM, AMSTRAD, COMMODORE y MSX



Todo el universo
del Software
mes a mes

**MICROMANÍA ya está a la venta
¡Pídela en tu Kiosco!**

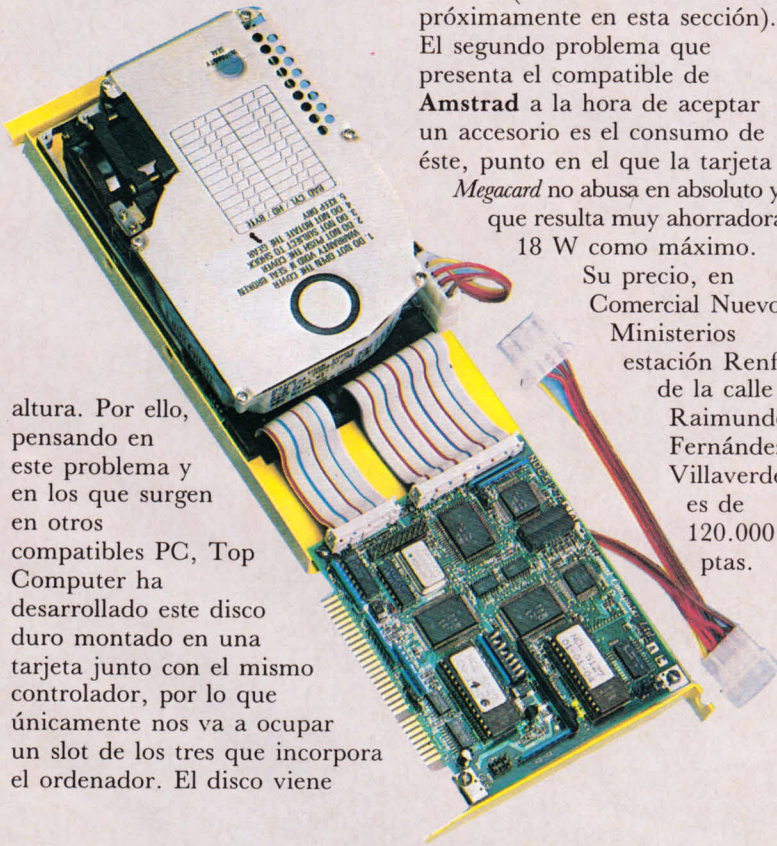
No hay dos sin tres

Los usuarios del **Amstrad PC 1512** con dos unidades de disco, quizá se encuentran con las alas cortadas a la hora de ampliar su equipo con un disco duro. El reducido espacio de la unidad central del ordenador no da opción a más de dos unidades de almacenamiento de media

firmado por NEC y cuenta con un tiempo medio de acceso de 85 mseg.; el controlador ocupa la tercera parte del área de la tarjeta y es capaz de manejar no sólo el disco duro que le acompaña, sino también otro, que en el caso del PC de **Amstrad** pudiera ser del tipo externo (de éstos hablaremos próximamente en esta sección). El segundo problema que presenta el compatible de **Amstrad** a la hora de aceptar un accesorio es el consumo de éste, punto en el que la tarjeta *Megacard* no abusa en absoluto ya que resulta muy ahorradora: 18 W como máximo.

Su precio, en Comercial Nuevos Ministerios estación Renfe de la calle Raimundo Fernández Villaverde, es de 120.000 ptas.

altura. Por ello, pensando en este problema y en los que surgen en otros compatibles PC, Top Computer ha desarrollado este disco duro montado en una tarjeta junto con el mismo controlador, por lo que únicamente nos va a ocupar un slot de los tres que incorpora el ordenador. El disco viene



A dos niveles

La mesa *Tableman* que os presentamos en la fotografía viene a solucionar el problema del emplazamiento del equipo informático en uso doméstico, como en oficina. Cuenta con dos bandejas. La superior, a una altura sobre el suelo de 79 cm, da cabida tanto al monitor como a la impresora; la inferior es extraíble, para que el conjunto ocupe el menor espacio cuando no se utiliza, y sostiene el teclado del ordenador dejando lugar para manejar textos o deslizar un ratón. Bajo las bandejas, se encuentra una rejilla en la que se pueden guardar manuales o libros de consulta y en un nivel inferior una cesta móvil de gran capacidad.

Las dimensiones de la mesa son de 102 cm de ancho por 50 cm de fondo. Su precio es de 22.300 ptas. en Peek & Poke, Génova 11. 28004 Madrid.

Hasta diez

Hasta diez discos de 5 1/4 pulgadas pueden entrar en este práctico archivador de plástico transparente y personalizado con el logotipo de Micro World, Zurbano, 76, Madrid.

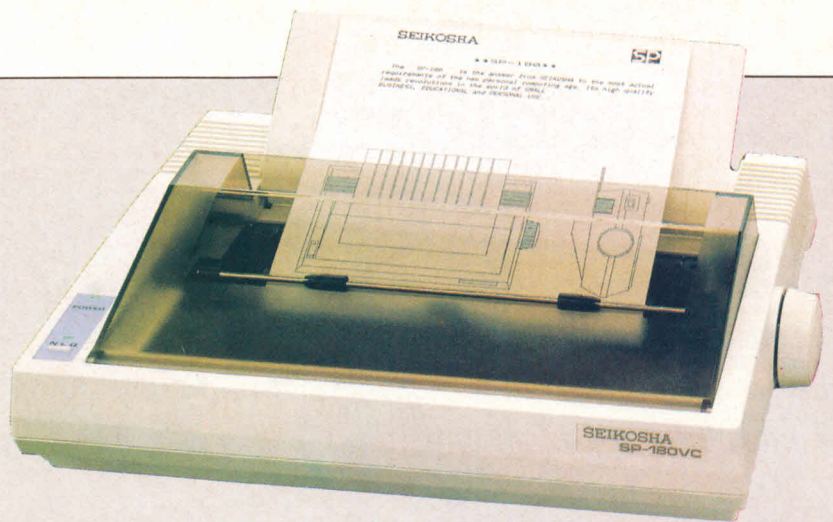
Su precio es de 560 ptas. y se mantiene la caja abierta y de pie abatiendo la tapa transparente.



Más por menos

Que una impresora para los Amstrad CPC cueste 35.000 ptas. no llama demasiado la atención, pero si pasamos a describir sus características y afirmamos que lleva la etiqueta de Seikosha, podríamos decir que nadie da más por menos.

Por descontado, la impresora Seikosha SP 180 A cuenta con todos los tipos de letra: pica, élite, condensada, expandida..., alcanzando una velocidad de impresión en modo Pica de 100 caracteres por segundo..., y, además, todos estos tipos se consiguen también en modo de alta calidad (NLQ) con una velocidad lógicamente más lenta, de 16 caracteres por



segundo, pero con la ventaja de que no sólo se obtiene mediante códigos de control desde el ordenador, sino también pulsando una tecla en la misma impresora.

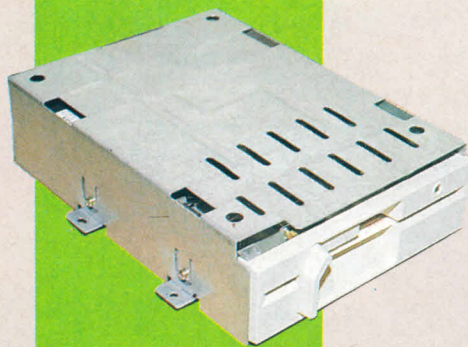
La admisión del papel se efectúa mediante fricción para

hojas sueltas y por tracción con alimentador para papel continuo.

La encontraréis en Informática Papelería, Plaza de Castilla de Madrid, junto al Asador.

La que faltaba

Todos aquellos que contaban con un Amstrad PC 1512 con la configuración mínima de un solo disco, ya pueden saltar a la categoría superior con el kit de la segunda unidad de disco que ofrece el mismo fabricante Amstrad.



Se encuentra en Micro World, Zurbano 76, de Madrid, al precio de 44.800 ptas. Su montaje no ofrece la menor dificultad, quedando alojada en el hueco de la izquierda de la unidad central y no necesita controlador, ya que la misma placa principal estaba preparada para aceptarlo. Tras su instalación, el disco A: pasa a ser el nuevo, mientras que el B: será el antiguo.

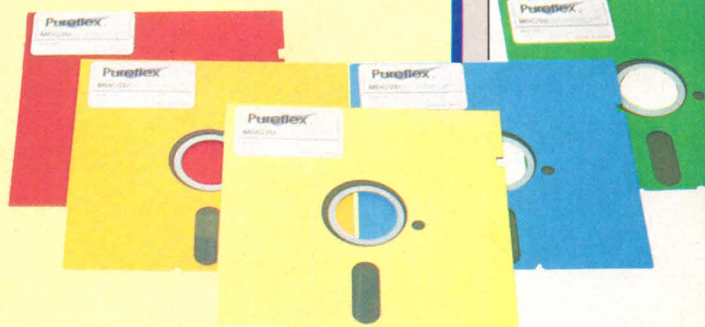
Escalera de color

Cuando se trabaja con prisas, localizar un disco de 5 1/4 pulgadas en un archivador de discos negros se parece a encontrar una aguja en un pajar. Para facilitar las cosas, aquí tenemos estos discos de colores de la firma Pureflex y que podéis encontrar en Informática Grotur, Jaime el Conquistador 27, Madrid. Tel. (91) 474 55 00, al precio de

4.000 ptas. la caja de diez unidades.

Cada caja contiene una gama de cinco colores con dos discos por color, aparte, claro está, de etiquetas adhesivas para nombrar su contenido.

De esta manera, con los discos de colores iremos a tiro hecho para localizar cualquier tema.



La aventura de Laserbase

Los juegos de «marcianitos» tienen un no se qué de entrañable que los hace revivir periódicamente en las páginas de todas las revistas que ofrecen programas a sus lectores. A nosotros nos ha vuelto a ocurrir: nos han enviado un buen programa relativo a una vieja idea. Aquí está.

El escenario de la batalla, porque de una guerra en el espacio se trata, nos sitúa en un lugar sorprendentemente cercano, cuya existencia y ubicación no pertenecen al nebuloso dominio de la ciencia-ficción: la Luna.

Una espantosa catástrofe ha sucedido en el Sistema Solar: una raza agresiva y desconocida ha ido atacando y destruyendo las colonias humanas de los planetas exteriores, metódicamente, sin misericordia. Todos los intentos de diálogo han sido infructuosos. Los

```
10 REM Laserbase
20 REM por
30 REM MK & PA Leybourne
40 REM(c) Amstrad Semanal
50 REM
60 CALL &BB4E:CALL &BC02
70 SYMBOL AFTER 256:MEMORY &9FFF:SYM
MBOL AFTER 32:GOTO 280
80 REM *****
90 REM Explosiones
100 REM *****
110 FOR a=1 TO 2:CALL &BD19:TAG:PRINT CHR$(245+a);:TAGOFF
120 MOVER -32,0:TAG:PRINT CHR$(245+a);:TAGOFF
130 MOVER -32,0:NEXT:RETURN
140 REM *****
150 REM Rutina de puntuacion
160 REM *****
170 si=PEEK(&A14B):c=3-si
180 IF n<243 THEN sc=sc+5*c:GOTO 200
190 sc=sc+2*c
200 PEN 2:PAPER 15:LOCATE 6,24:PRINT sc;
210 IF pow<0 THEN pow=0
220 LOCATE 16,24:PRINT CINT(pow);
230 BORDER 13:IF ((la<1) OR (pow<1)) THEN 2870
240 RETURN
250 REM *****
260 REM Inicializa
270 REM *****
280 DEFINT a-z:DEFREAL p
290 ENT 1,150,15,1:ENT 2,1,1,2,100,20,1:ENV 1,5,10,40
300 GOSUB 2330:REM Titulo
310 RESTORE 2160:GOSUB 1500:REM Define Aliens
320 GOSUB 2480:REM instrucciones
330 RESTORE 1520:GOSUB 1500:REM Define caracteres
340 MODE 0:ORIGIN 0,0,0,640,383,48
350 WINDOW #1,1,20,23,25:PAPER #1,1,5:CLS:CLS#1
360 INK 8,13,0:INK 9,25,13:INK 10,13:INK 11,26,13
370 INK 15,13:INK 0,0:INK 12,1:INK 13,26:INK 6,11
380 PAPER 0:BORDER 13:SPEED INK 150,150
390 d=-1:lx=96:hx=544:ly=156:hy=240
400 REM *****
410 REM define stella y torreta
420 REM *****
430 sc=0:pow=500:l=1:r=1:la=2:INK 3,6:INK 1,24
440 al=10:PAPER 0:PEN 1:FOR a=2 TO 21
450 b=INT(RND*18)+1
460 LOCATE b,a:c=INT(RND*4):d=d+1:IF d=4 THEN d=0
470 PEN 8+d:IF c=1 THEN PRINT CHR$(46); ELSE PRINT CHR$(248);
480 NEXT a
490 REM *****
500 REM define nave alien
510 REM *****
520 PRINT CHR$(23);CHR$(0)
530 CALL &A22B:REM dibuja nave
540 PLOT 288,287,0:DRAW 96,0:MOVER 0,-16:DRAW 0,-16:DRAW -96,0:DRAW 0,16:DRAW 96,0
550 MOVE 304,324:DRAW 64,0,4:MOVER 0,2:DRAW -64,0:MOVER 0,2:DRAW 64,0
560 PEN 2:PAPER 15:LOCATE 1,24:PRINT "Score";
570 LOCATE 11,24:PRINT "Ener.":GOSUB B 200:REM Score
580 PEN 3:PAPER 0
590 LOCATE 6,22:IF l THEN PRINT CHR$(239) ELSE PRINT CHR$(229)
600 LOCATE 15,22:IF r THEN PRINT CHR$(239) ELSE PRINT CHR$(229)
610 PEN 2:PAPER 15:sx=288:sy=176
620 nf=1
630 REM *****
640 REM dibuja alien
650 REM *****
660 lau=1:c=INT(RND*3)+10
670 IF c=11 THEN n=240 ELSE n=243
680 PRINT CHR$(23);CHR$(1);
690 xp=(c-1)*32:yp=271:CALL &A000,x
```

alienígenas han contestado que su objetivo final es la aniquilación de nuestro planeta madre, por quién sabe qué olvidados agravios.

La batalla no ha sido un camino de rosas para ellos. Han sufrido muchas bajas, y se están reorganizando para lanzar el ataque final. Sólo una cosa puede detenerlos: el complejo defensivo *Luna SDI*, constituido por las últimas reservas de material humano y militar del Sistema.

Los extraños son grandes estrategas, y saben que para aniquilar a la Luna y atacar después a la Tierra, deben conseguir algo básico, a saber: acercarse a la Luna. Sólo se lo impiden las baterías de cañones láser de largo alcance que la defienden y han decidido intentar destruirlas enviando gigantescas naves madres repletas de velocísimos cazas. Así, aun a costa de grandes bajas, debilitarán lo suficiente el campo de fuerza que protege a la Luna y el ejército de reserva hará lo demás.

Ni que decir tiene que los avezados defensores de la Tierra son los de más rápidos reflejos del planeta, y sus puntuaciones en las máquinas de los bares así lo certifican.

No cabe duda tampoco de que tú, casualmente, eres su aclamado capitán, alias *Dedos de Pantera O'Sullivan* y que tu batería láser se encuentra justo enfrente de la nave madre enemiga, donde el ataque será más despiadado y virulento.

Si *Laserbase* fuera un juego en lugar de la más cruda realidad, diríamos que hacen falta unos ojos de lince y una mano de hierro para acertar a los esquivos y erráticos alienígenas, los cuales, sabiendo que se juegan su sucia vida, no están dispuestos a dejarse abatir por nada del mundo (del suyo, claro).

Si se tratara de un juego, añadiríamos que está muy bien hecho, y que tanto de su parte Basic como de la de Código Máquina se puede aprender mucho.

```
p, yp, lau, n
960 CALL &A068, sx, sy, nf, x: nf=0
970 FOR a=1 TO 1000: NEXT: lau=0
980 REM *****
990 REM rutina de movimiento
1000 REM *****
1010 b=INT(RND*4): c=INT(RND*4)
1020 IF b AND 1 THEN d1=-1 ELSE d1=
1
1030 IF c AND 1 THEN d2=-1 ELSE d2=
1
1040 b=b*16
1050 xp=xp+4*d1: yp=yp+2*d2
1060 IF xp>hx THEN xp=hx: d1=-1
1070 IF xp<lx THEN xp=lx: d1=1
1080 IF yp>hy THEN yp=hy: d2=-1
1090 IF yp<ly THEN yp=ly: d2=1
1100 CALL &A000, xp, yp, lau, n: REM mue
ve alien
1110 IF yp>160 THEN 1200
1120 REM *****
*
1130 REM rutina de disparo del alie
n
1140 REM *****
*
1150 SOUND 130, 40, 40, 15, 0, 2: CALL &A
0D2
1160 IF TESTR(0, -1)=3 THEN 1290: REM
rutina del laser
1170 SOUND 129, 150, 30, 15, 0, 1: SOUND
132, 600, 60, 4, 1, 0, 8
1180 MOVER -16, 0: GOSUB 110: pow=pow-
4: REM Explosion
1190 GOSUB 200: REM Actualiza puntos
1200 IF INKEY(ee)>-1 THEN SOUND 129
, 150, 30, 15, 0, 1: CALL &A104, sx, sy, l, r
: pow=pow-(la*0.25): GOSUB 210
1210 IF TESTR(0, 0)=7 THEN 1410
1220 sy=sy+2*((INKEY(cc)>-1)-(INKEY
(dd)>-1))
1230 sx=sx+4*((INKEY(bb)>-1)-(INKEY
(aa)>-1))
1240 CALL &A068, sx, sy, nf, x
1250 b=b-1: IF b=0 THEN 1010 ELSE 10
50
1260 REM *****
```



```
1270 REM mas rutinas del alien
1280 REM *****
1290 BORDER 6: MOVER -16, 0: FOR b=1 T
O 2
1300 PLOT 0, 0, b
1310 BORDER 9+b: FOR d=1 TO 2: GOSUB
110: REM Explosion
1320 NEXT d: NEXT b: PLOT 0, 0, b
1330 IF xp<300 THEN c=6 ELSE c=15
1340 IF c=6 THEN l=0 ELSE r=0
1350 LOCATE c, 22: PEN 3: PAPER 0: PRIN
T CHR$(229);
1360 pow=pow-50: la=la-1
1370 GOTO 1190
1380 REM *****
1390 REM muerte alien
1400 REM *****
1410 GOSUB 170: REM Score
1420 SOUND 132, 600, 60, 4, 1, 0, 12
1430 nf=1: CALL &A068, sx, sy, nf, x
```

```

1440 PLOT -1,0,7:MOVE xp,yp:TAG:PRI
NT CHR$(n+si);:TAGOFF
1450 PRINT CHR$(23);CHR$(1);:CALL &
A16A
1460 al=al-1:IF al<1 THEN GOTO 3010
ELSE 920
1470 REM *****
1480 REM Redefine caracteres
1490 REM *****
1500 READ zt:IF zt=999 THEN RETURN
ELSE READ b,c,d,e,f,g,h,i:SYMBOL zt
,b,c,d,e,f,g,h,i:GOTO 1500
1510 REM 0
1520 DATA 48,254,130,130,130,134,13
4,134,254
1530 REM 1
1540 DATA 49,8,8,8,24,24,24,24,24
1550 REM 2
1560 DATA 50,254,130,2,254,192,192,
192,254
1570 REM 3
1580 DATA 51,252,132,4,254,6,6,134,
254
1590 REM 4
1600 DATA 52,252,132,132,132,132,25
4,12,12
1610 REM 5
1620 DATA 53,254,128,128,254,6,6,13
4,254
1630 REM 6
1640 DATA 54,254,130,128,254,134,13
4,134,254
1650 REM 7
1660 DATA 55,254,2,2,6,6,6,6,6
1670 REM 8
1680 DATA 56,124,68,68,254,134,134,
134,254
1690 REM 9
1700 DATA 57,254,130,130,130,254,6,
6,6
1710 REM A
1720 DATA 65,124,68,68,254,194,194,
194,194
1730 REM F
1740 DATA 70,254,128,128,252,192,19
2,192,192
1750 REM G
1760 DATA 71,254,130,128,198,194,19
4,194,254
1770 REM H
1780 DATA 72,130,130,130,254,198,19
8,198,198
1790 REM N
1800 DATA 78,254,130,130,194,194,19
4,194,194
1810 REM O
1820 DATA 79,254,134,134,130,130,13
0,130,254
1830 REM P
1840 DATA 80,254,130,130,254,192,19
2,192,192
1850 REM S
1860 DATA 83,254,130,128,254,6,6,13
4,254
1870 REM a
1880 DATA 97,0,0,248,136,8,248,200,
248
1890 REM c
1900 DATA 99,0,0,248,136,192,192,20
0,248
1910 REM e
1920 DATA 101,0,0,248,136,248,192,2
00,248
1930 REM h
1940 DATA 104,0,0,128,248,136,200,2
00,200
1950 REM i
1960 DATA 105,0,32,0,32,32,48,48,48
1970 REM m
1980 DATA 109,0,0,254,146,218,218,2
18,218
1990 REM n
2000 DATA 110,0,0,248,136,200,200,2
00,200
2010 REM o
2020 DATA 111,0,0,248,136,200,200,2
00,248
2030 REM p
2040 DATA 112,0,0,248,136,200,248,1
92,192
2050 REM r
2060 DATA 114,0,0,248,128,192,192,1
92,192
2070 REM t
2080 DATA 116,0,32,112,32,48,48,48,
48
2090 REM v
2100 DATA 118,0,0,200,200,200,72,80
,112
2110 REM w
2120 DATA 119,0,0,146,146,218,218,2
18,254
2130 REM laser destruido
2140 DATA 229,0,0,0,44,60,228,126,2
03
2150 REM mini-alien 1
2160 DATA 240,0,0,20,42,62,42,20,0
2170 REM alien 1
2180 DATA 241,0,34,65,93,119,93,65,
34
2190 REM gran alien 1
2200 DATA 242,102,129,189,231,231,1
89,129,102
2210 REM mini-alien 2
2220 DATA 243,0,0,0,24,60,66,0,0
2230 REM alien 2
2240 DATA 244,0,0,36,24,60,90,0,0
2250 REM gran alien 2
2260 DATA 245,0,66,36,60,36,90,129,
129
2270 REM explosion
2280 DATA 246,0,42,126,63,126,63,22
9,0
2290 REM explosion
2300 DATA 247,124,126,130,65,130,65
,88,60
2310 REM stella
2320 DATA 248,0,0,0,16,0,0,0,0,999
2330 CLS:MODE 1:PEN 3
2340 PRINT" * *** ***
*** ***
2350 PRINT" * * * *
* * *
2360 PRINT" * *** ***
** ***
2370 PRINT" * * * *
* **
2380 PRINT" *** * * ***
*** * *
2390 PRINT:PRINT:PRINT
2400 PRINT" *** ** **
* ***

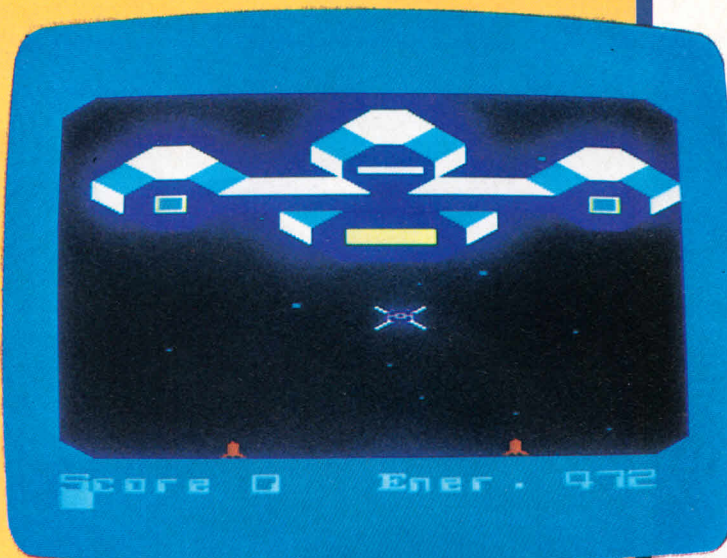
```

Serie
ORO

```

2410 PRINT"      * *      * *      *
      *
2420 PRINT"      ****      ***      **
      *      **
2430 PRINT"      * *      * *
      *      *
2440 PRINT"      ****      * *      **
      *      ***
2450 PEN 2:LOCATE 20,15:PRINT"BY":L
OCATE 12,17:PRINT "M.K & P.A Leybou
rne":LOCATE 19,19:PRINT"1987"
2460 PEN 1:LOCATE 8,24:PRINT"Prepar
ado para la batalla "
2470 GOSUB 3120:FOR a=1 TO 25:PRINT
:FOR b=1 TO 100:NEXT:NEXT:RETURN
2480 PEN 3:CLS:PRINT TAB(15)"LASER
BASE":PRINT TAB(15) STRING$(10,CHR$(
95)):PRINT
2490 PEN 1:LOCATE 1,3:PRINT CHR$(20
)
2500 PRINT "El final de la guerra g
alactica se esta ";
2510 PRINT "aproximando.Todos los p
lanetas mayores del Sistema estan
destruidos,excepto ";
2520 PRINT "la Tierra y su Luna."
2530 PRINT:PRINT "Tu controlas la t
orreta laser de la Luna, y debes de
struir cuantos";
2540 PRINT "alienigenas puedas.":PR
INT
2550 PRINT:PRINT "La puntuacion del
juego es como sigue:-"
2560 FOR b=243 TO 246 STEP 3
2570 PRINT:PRINT:PRINT TAB(8);:FOR
d=1 TO 3
2580 PEN 3:PRINT CHR$(b-d);:PEN 1:P
RINT " . . .";:PEN 2
2590 IF b > 243 THEN PRINT 2*d;:IF
d=2 THEN PRINT " ";:GOTO 2610 ELSE 2
610
2600 PRINT 5*d;
2610 NEXT d,b:GOSUB 2850:REM Espera
tecla
2620 LOCATE 1,3:PRINT CHR$(20):PEN
1
2630 PRINT "El ataque alienigena em
erge del transporte, fuera del alca
nde del laser."
2640 PRINT "Cada oleada causa 10 at
aques que hay que evitar antes de q
ue:-"
2650 PRINT
2660 PRINT " A). Los aliens destruy
an ambas torretas laser.
2670 PRINT:PRINT " B). Se te acabe
la energia.":PRINT
2680 PRINT:PRINT"Usa Joystick o Tec
lado. . . ."
2690 PEN 3:PRINT TAB(10)"> ";:PEN 1
:PRINT "Mueve ";:PEN 2:PRINT
"Dere."
2700 PEN 3:PRINT TAB(10)"< ";:PEN 1
:PRINT "Mueve ";:PEN 2:PRINT
"Izq."
2710 PEN 3:PRINT TAB(10)"A ";:PEN 1
:PRINT "Mueve ";:PEN 2:PRINT
"A."
2720 PEN 3:PRINT TAB(10)"Z ";:PEN 1
:PRINT "Mueve ";:PEN 2:PRINT
"Aba."

```



```

2730 PEN 3:PRINT TAB(6)"SPACE ";:PE
N 2:PRINT TAB(23)" fuego"
2740 PRINT:PRINT TAB(13)"Space o Fu
ego":en=0:s=1
2750 WHILE INKEY$("<")"":WEND
2760 FOR a=1 TO 35:a$=INKEY$:IF a$<
">"" THEN 2790
2770 NEXT:IF s=1 THEN 2760
2780 INK co,en MOD 27:en=en+1:GOTO
2760
2790 a$=UPPER$(a$)
2800 IF a$=" " THEN aa=31:bb=39:cc=
71:dd=69:ee=47:s=0:RETURN:REM Tecla
do
2810 IF a$="N" THEN CALL 0
2820 IF a$="Z" THEN ee=77:GOTO 2840
2830 IF a$="X" THEN ee=76:GOTO 2840
ELSE GOTO 2750
2840 aa=75:bb=74:cc=73:dd=72:s=0:RE
TURN
2850 PRINT:PRINT:PRINT TAB(10)"Puls
a para continuar"
2860 WHILE INKEY$("<")"":WEND:WHILE IN
KEY$="":WEND:RETURN
2870 REM *****
2880 REM fin del juego
2890 REM *****
2900 CLG:LOCATE 5,5:PEN 2:PAPER 0
2910 WHILE INKEY$ "<")"":WEND
2920 PRINT " Game Over":PRINT
2930 en=1:co=3:IF sc < hi THEN co=1
:GOTO 2950
2940 PEN 3:PRINT:PRINT " New Hi-Sc
ore ";sc:hi=sc:GOTO 2960
2950 PEN 1:PRINT:PRINT " Hi-Scor
e ";hi
2960 PEN 6:PRINT:PRINT:PRINT "
Otra vez? "
2970 PEN 2:PRINT:PRINT:PRINT " Spac
e o Fuego o N"
2980 GOSUB 2750:CLG:REM Espera tecl
a
2990 t=0:sc=0:LOCATE 7,24:PAPER 15:
PRINT STRING$(3," ")

```

Serie

ORO



```

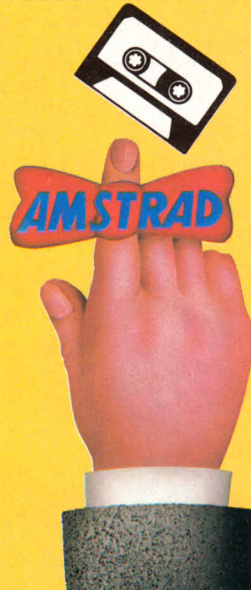
3000 GOTO 430
3010 REM *****
3020 REM redibuja aliens
3030 REM *****
3040 CLG:LOCATE 6,5:PEN 2:PAPER 0:P
RINT"Preparado! "
3050 LOCATE 8,10:PRINT "Otra  "
3060 LOCATE 9,15:PRINT "ola "
3070 FOR a=1 TO 2000:NEXT:CLG:t=t+1
:sc=sc+t*50
3080 GOSUB 200:a1=10:GOTO 440
3090 REM *****
3100 REM codigo maquina
3110 REM *****
3120 RESTORE 3180
3130 a=&A000:FOR b=3180 TO 3390 STE
P 10:f=0:READ a$,d
3140 FOR c=1 TO LEN(a$) STEP 2:e=VA
L("&" +MID$(a$,c,2)):f=f+e
3150 POKE a,e:a=a+1:NEXT c:IF f<>d
THEN 3170
3160 NEXT b:RETURN
3170 PRINT "Data error en linea";b:
PRINT:PRINT "Tecla GOTO 280 despue
s de corregir el ERROR":STOP
3180 DATA 3E07CDDEBDD7E02B7F5200DE
D5B4DA12A4FA1CDC0BBCD32A0CD3EA00E00
F1200D3A4FA1FEE63006,4915
3190 DATA 0CFEBE30010C79324BA13A4AA
1473A4BA180CDFCBBC9CD5BA0ED534DA122
4FA1F5CDC0BBF1B72801,5137
3200 DATA AF324BA1DD7E00324AA1C9DD6
E04DD6605DD5E06DD5607C93E04CDDEBDD
7E02B72803C386A0ED5B,4865
3210 DATA 46A12A48A1CDC0BBCD19BDCD8
9A0CDC4A0110C0021FAFFD5E5CDF9BB1118
0021F8FFCDC3BBE1D1D5,6033
3220 DATA CDF9BB11D0FF210000CDC3BB
1D5210600E5CDF9BB111800210800CDC3BB
E1D1CDF9BBC9CD5BA0ED,5710
3230 DATA 5346A12248A1CDC0BBC911F0F

```

```

FD5E1CDC3BB3E03CDDEBB11000021B0FFE5
D5CDF9BB010606CD38BC,5608
3240 DATA 010D0DCD38BCD1D5215000CDC
3BBD1E1CDF9BBC93E03CDDEBBBCD5BA0D511
F3FF192255A1D1211A00,5310
3250 DATA 192257A1CD23A1CD23A1C9214
00DD7E02B7280BE511AE00CDC0BBCD5CA1
E1DD7E00B7C811CE01CD,4836
3260 DATA C0BBCD5CA1C900000000000000
000000000000000000000000000000ED5B
57A12A55A1CD19BDCDF6,2772
3270 DATA BBC93E04CDDEBB2A4DA111100
019E52A4FA111F8FF19D1ED5351A12253A1
3E01325BA13E023259A1,4240
3280 DATA 3E10325AA101EBA1CDB9A13A5
AA13D325AA120F43A59A13D3259A120E33A
5BA13C325BA1FE2020D3,4403
3290 DATA C90A035F0A03572A53A13A5BA
1193D20FCE50A035F0A03572A51A13A5BA1
193D20FCEBE1C5CDEABB,4059
3300 DATA 21000011FFFFCDEDBBC1C9010
001000200030001000200FDFDFDFDFCF03
000100FFFF0200FEFF00,4143
3310 DATA 0001000000FEFFFEFF0100FEF
F020002000000300FEFFFFfFFFEFF0100FC
FFCFFFDFF,4587
3320 DATA 0116012140A27E23E50BC5CD5
ABBC1E179B020F2C9,2553
3330 DATA 1F0A020F0DD68FD71F0304D68
FD71F1104D68FD70F061F0205D60E0DD720
D60E00D71F11050E0DD7,3162
3340 DATA 20D60E00D71F09040E0DD50E0
CD420D50E0DD41F10050E00D61F0903D60E
0DD720D60E00D71F0905,2780
3350 DATA 0F0DD50F0C8F8F8F0F0DD41F0
2060E06D70F0CD68FD70F0DD60F0C0E0D20
2020D58FD42020200F06,2887
3360 DATA D50F0C0E06D68FD70F060E0DD
40F0D0E001F0207D50F0C8F208F8F8F8F8F
8F8F8F8F8F8F8F8F208F,3623
3370 DATA 0F0DD40F0C0E0D1F0A06D58FD
40E001F0308D58FD41F08080F060E0DD50E
0CD40F0C8F8F8F0F06D5,2769
3380 DATA 0E0DD40F0C0E001F1108D58FD
41F08090F0DD50F0C8F0F0D0E012020200F
0C0E008F0F0DD40F0C1F,1984
3390 DATA 090AD58F8F8FD40E000F0F1F0
10101D41F1401D51F1416D61F0116D71F04
070E01E91F1207E90000,2563

```



Para que tus dedos no realicen el trabajo duro, **AMSTRAD Semanal** lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicítanoslo.

2 FABULOSOS PROGRAMAS

GRATIS PARA TI

ARMY MOVES

Como miembro del Cuerpo de Operaciones Especiales, Dendhal ha sido adiestrado en varios sistemas de combate distintos, así como en el manejo de todas las armas, explosivos y técnicas de guerra en la selva. Ahora, tras largos años de entrenamiento, le ha llegado el momento de demostrar sus habilidades y atravesar, por tierra, mar y aire, las líneas enemigas. ¿Lo conseguirá?

GAME OVER

Los problemas de libertad existen hasta en las más lejanas galaxias. Si no, que se lo digan a los habitantes del planeta Porshaco, quienes están sufriendo en sus carnes viscosas la tiranía de la princesa Gremla. Afortunadamente, Arkos, el más hábil de todos los mega-terminators, ha decidido acabar con este juego; afortunadamente para nosotros, comienza Game Over.

Suscríbete hoy mismo a AMSTRAD y recibirás a vuelta de correo los dos mayores éxitos de Dinamic



¡No te pierdas esta oferta!

Envía hoy mismo tu cupón

Oferta válida sólo para España



Benefícate de las ventajas de la tarjeta de crédito. Un número más, gratis, en tu suscripción y la posibilidad de realizar el pago aplazado.

DRAGON'S LAIR[®]
OWNED BY MAGICOM, INC. AND USED BY PERMISSION

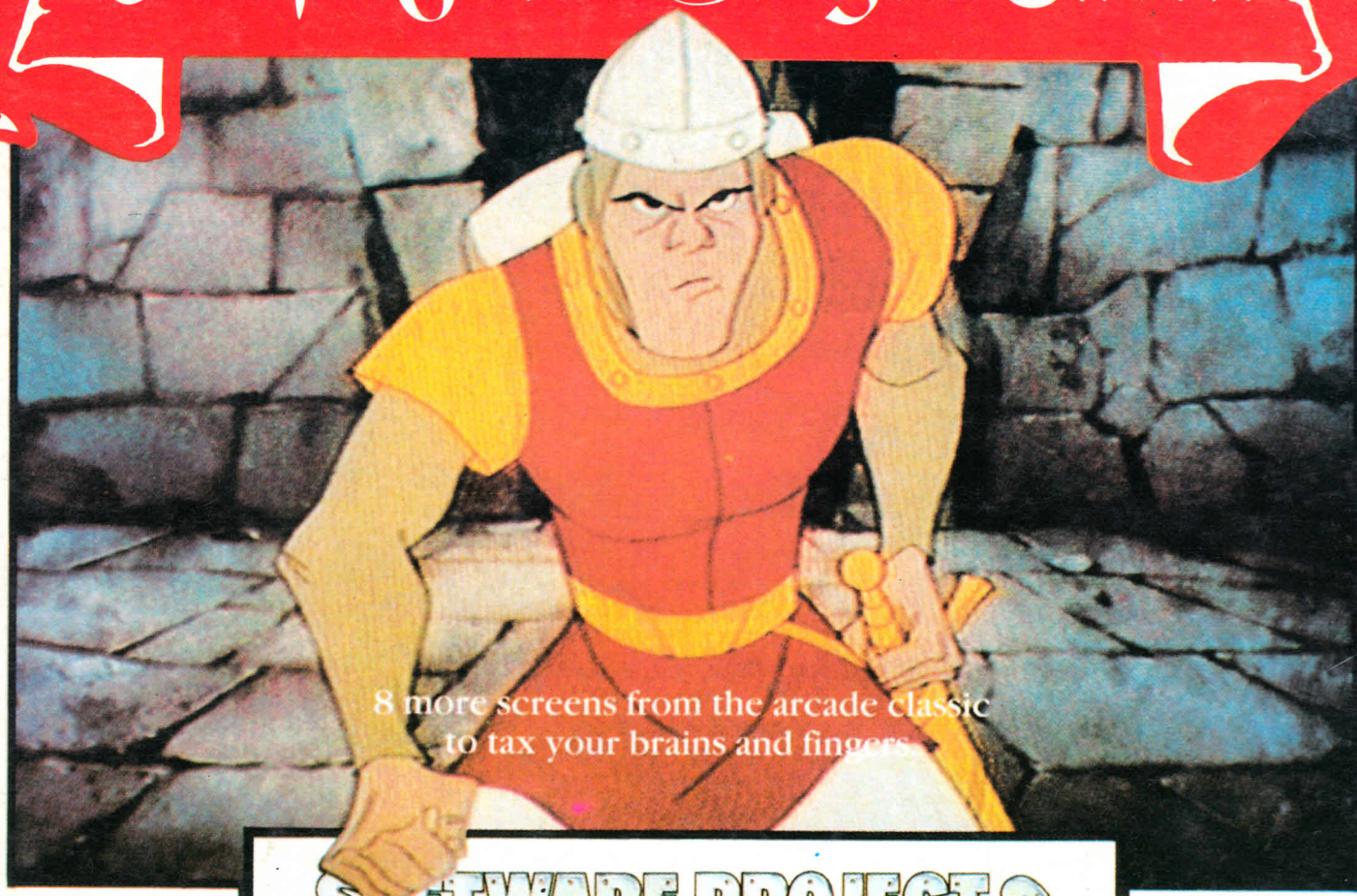
ERBE
Software

875 ptas.

* DISCO AMSTRAD 2.250 PTAS.

PART II

Escape from Singe's Castle



8 more screens from the arcade classic
to tax your brains and fingers.

SOFTWARE PROJECTS

ii LA 2.^a PARTE DE "DRAGON'S LAIR"!!

CONTINUACION QUE SUPERA Y MEJORA ENORMEMENTE A "DRAGON'S LAIR", EL JUEGO DE LAS MAQUINAS QUE SE CONVIRTIO EN LEYENDA. LA HUIDA DEL CASTILLO DE SINGE AÑADE AVENTURAS, ACCION Y EMOCION A LA QUE FUE PRIMERA PARTE...; AFILA TU ESPADA Y TU CEREBRO.

ERBE
Software

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA:

ERBE SOFTWARE. C/. STA. ENGRACIA, 17 - 28010 MADRID. TELEF. (91) 447 34 10
DELEGACION BARCELONA. C/. VILADOMAT, 114 - TELEF. (93) 253 55 60.