

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO I N.º 19

160 Ptas.

(incluido
I.V.A.)

**GENERACION
DE NUMEROS
ALEATORIOS**

**EDICION
DE
COMANDOS
CON UN
TOQUE
DE TECLA**

**Transforma los
programas BIN
en Datas Basic
para teclear**

**DIBUJO
DE SUPERFICIES
REGLADAS
SIMULACION
DE VOLUMENES**

SOFTWARE

**Bruce Luis
lucha contra los
grandes maestros**



¡FANTÁSTICO!



Si quieres un ordenador de «una pieza» piensa en el AMSTRAD CPC 464. Tendrás un ordenador de una vez por todas. Gracias a sus 64K RAM y 32K ROM y a sus casi ilimitadas posibilidades de crecimiento, tienes garantizado que el ordenador CPC 464 no se te quedará pequeño.

COMPLETO

Además, gracias a su monitor (color o fósforo verde) de alta resolución (hasta 640 x 200 píxeles direccionados individualmente) y a su unidad de cassette incorporada al teclado, podrás disfrutar de tu AMSTRAD de una manera independiente, prescindiendo de televisor y del radiocassette de tu casa (a veces tan solicitados).

¿Y QUE ME DICES DE LOS PROGRAMAS?

Actualmente ya hay cientos de ellos disponibles en España. Sin olvidar que son varias las revistas dedicadas sólo a AMSTRAD y que el número de libros y periféricos del CPC 464 crecen día a día, potenciando así la creatividad de tu ordenador personal.

CARACTERÍSTICAS TÉCNICAS

- Microprocesador Z80 (4MHz).
- Memoria de 64K RAM y 32K ROM.
- Gráficos de alta resolución de hasta 640 por 200 píxeles direccionables individualmente.
- Unidad de cassette incorporada en el teclado.
- Monitor color o fósforo verde incluido en el Sistema.
- Texto en pantalla de 20,40 y 80 columnas por 25 líneas.
- LOCOMOTIVE BASIC ampliado.
- Paleta de 27 colores y efectos de «flash».
- Teclado profesional tipo QWERTY con bloque numérico y teclas para cursor independientes.
- Salida Centronics paralela.
- Lector de discos de 3" (180K por cara) opcional (con CPM y Dr. LOGO incluidos junto a la unidad de disco).
- Manuales en castellano.

Al comprar tu ordenador CPC 464, AMSTRAD ESPAÑA te obsequia con 8 cassettes de programas y el libro «Guía de Referencia BASIC para el programador».

Exije la **GARANTIA AMSTRAD ESPAÑA ÚNICA VÁLIDA PARA ACCEDER AL SERVICIO TÉCNICO OFICIAL.**

PRECIO:

— **66.900 ptas.**

(monitor fósforo verde)

— **95.900 ptas.**

(monitor color)

¡¡ Increíble !!

ESPAÑA

AMSTRAD

GARANTIA INDESCOMP

Avd. del Mediterráneo, 9 - 28007 Madrid Tels. 433 45 48 - 433 48 76 - Telex 47660 FAX - 4332450

AMSTRAD

sumario

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

Victor Prieto

Subdirector

Jose Maria Diaz

Redactora Jefe

María García

Diseño

José Flores

Colaboradores

Francisco Portalo, Pedro Sudón

Miguel Sepúlveda,

Francisco Martín,

Jesus Alonso, Pedro S. Pérez

Amalio Gómez

Juan J. Martínez,

David Sopuerta, Alberto Suárez,

Eduardo R. Velasco

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Javier Martínez

Portada

M. Barco

Ilustradores

Javier Iguel, J. Pans, F. L.

Frontón, J. Septien, Pejo, J. J.

Mora, Luigi Pérez

Edita

HOBBY PRESS S.A.

Presidente

María Andrino

Consejo Delegado

José I. Gómez-Centurión

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortes

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

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración**y Publicidad**

La Granja, 39

Poligono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Impreme

ROTEDIC, S. A. Cita. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Navacom, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solano, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos

de la revista

COMPUTING with**the AMSTRAD**

Representante para Argentina, Chile,

Uruguay y Paraguay, Cis.

Americana de Ediciones, S.R.L. Sud

América 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de los

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

Año II • Número 19 • 7 al 13 de Enero de 1986

160 ptas. (Incluido el IVA)

5 Primera plana

El banco en casa por televisión. El proyecto sprit sin fondos.

Primeros 6 pasos

Nos introducimos con el artículo de esta semana en un campo relativamente poco explorado del **Amstrad**, esencial para simulaciones en experimentos científicos, en juegos y para la programación en general: los números aleatorios. **¿Puede un ordenador crear algo al azar?**



12 ProgramAcción

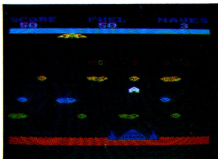
El teclado del **Amstrad**, como saben nuestros lectores e indica el manual, es definible por el usuario, es decir, una determinada tecla puede realizar más de una función. Detrás de esto se esconden muchos secretos y posibilidades de nuestro ordenador, que creemos están magníficamente abordadas y explicadas en el artículo de esta semana.

Análisis 16

El **Amstrad** es una potentísima calculadora, y posee unas excelentes capacidades gráficas. Usando estas dos características y se obtendrá el **Amstrad Análisis** que tenemos el placer de presentarlos en este número.

Decodificador Binario 20

De vez en cuando, a algún programador se le ocurre una idea loca, pero genial. Como por ahora no tenemos sección de Amstradmilagos, a la increíble rutina en código máquina que podréis leer y probar en el interior de la revista le hemos dado el prosaico nombre de **Decodificador Binario**. Por favor, estadud atentamente las inmensas posibilidades que ofrece.



24 Serie Oro

SPACE TRUCKER es un juego clásico por la emoción y el centelleo que ofrece siempre el mator marcianitos, pero no tan clásico por sus excelentes gráficos y sus técnicas de programación, realmente buenas. Probadlo y veréis.



Código máquina 28

Los ensambladores no estarían completos si no existieran unos programas llamados Herramientas de Depuración Dinámicas, a los cuales pertenece el **MONA3** de Hisoft, que se ofrece en el mismo paquete que el **GENA3**. **AMSTRAD Semanal** lo analiza con detalle en este número.

COMPUTIQUE

Te da más

GARANTIA **AMSTRAD**
ESPAÑA

64.900 Ptas.

Amstrad 464 f.
verde



VENTA A PLAZOS HASTA 36 MESES

Al comprar tu Amstrad te regalamos

• Estuche con ocho programas originales

- Fruit Machine
- Plaga Galáctica
- Procesador Texto
- Amstrew
- Almirante Graf
- Laberinto Sultan
- Oh Mummy
- Animal, Vegetal,
Mineral

• Joystick Gunshot I

• Un estupendo libro de Basic

• Los cuatro mejores programas:

- Deathlon
- Jef Set Willy
- Sabrewulf
- Beach-Head

• Guía de referencia del programador

- y además te obsequiamos con un curso
de introducción al Basic.



Nuevo Amstrad CPC6128: 109.500 ptas. (F. Verde)

COMPUTIQUE

Servimos a tiendas
Abrimos sábados por la tarde

Embajadores, 90 Tfno. 2270980
28012 Madrid



Nueva bajada de precios

IBM, número uno en el mundo de los ordenadores, ha vuelto a bajar sus precios en el mercado europeo.

El corte oscilará entre un 15 y un 40 por 100, según el modelo de que se trate.

La decisión de IBM tiene lugar un tiempo después de que su gran competidor, Apple Computer, recurriera a la misma medida.

En estos momentos, gran parte de las empresas fabricantes de ordenadores de todos los sectores informáticos se enfrentan a la dura necesidad de ajustar drásticamente sus precios a las nuevas circunstancias del mercado: una expansión menor y la aparición de grandes novedades tecnológicas en los campos del hardware y del software que afectan a los equipos antiguos, porque ahora las prestaciones de los nuevos son mucho mayores por el mismo dinero.

Este es el caso del IBM PC y del Macintosh de Apple.

Parece que se avecina una sangrienta guerra de precios.

Los abogados se informatizan

El colegio de abogados de Madrid ha llegado a un acuerdo con CCS para establecer unas normas de homologación a los programas destinados a dar soporte informático a los bufetes. Al parecer, los equipos responsables de la mecanización serán ordenadores personales pertenecientes a la firma Olivetti.

Los bancos en casa

E

l Banco de Santander ha adoptado para sus clientes el sistema del banco en casa por televisión, que permitirá a los usuarios beneficiarse de hasta 13 servicios distintos que serán posteriormente enriquecidos con 11 más.

El equipo teleinformático del banco en casa básicamente de una pantalla de televisión y un teclado alfanumérico.

La televisión puede ser una cualquiera doméstica convenientemente preparada, sin excluir su uso como televisión normal, y el teclado puede ser también el de cualquier ordenador personal o doméstico, el de un Amstrad, por ejemplo.

Opcionalmente, se puede ampliar el sistema incluyendo una impresora.

No obstante esta posibilidad, el Banco de Santander ha decidido adoptar como política de comercialización base el alquiler del terminal y el teclado al usuario, por un precio de 1.000 pesetas al mes.

Como para conectar con el ordenador central del banco hay que hacerlo mediante un teléfono, el coste de las llamadas y el tiempo de «conversación» corre a cargo del cliente.

Respecto al tema de la seguridad y secreto de las operaciones, tema clave para la viabilidad de todo esto, el sistema que se ha seguido consta de cuatro claves: tres facilitadas por el banco, que permiten acceder al menú de transacciones e inspeccionar la cuenta, pero no permiten alterar nada ni dar órdenes. La cuarta si lo permite, y, para mayor seguridad, está generada por el propio ordenador, esto es, el banco no la conoce.

Esto hace muy difícil el delito bancario a través de ordenador.

Las consultas también pueden realizarse por teléfono. El ordenador responde a ellas en tiempo real y con voz humanizada. Hasta ahora existen 9.000 usuarios de pantalla y 92.000 por teléfono exclusivamente.

Primera plana

No hay dinero

El proyecto Sprit es un intento europeo de dar respuesta a las investigaciones japonesas y norteamericanas de conseguir ordenadores de quinta generación, máquinas casi pensantes.

Pues bien, Sprit se ha quedado sin un duro. El presupuesto de 5 años, 900 millones de dólares, se ha evaporado en 8 meses.

Se espera conseguir nuevos fondos para este proyecto, en el que España también participa, porque es la única forma de acabar dependiendo más aún de Japón y USA en algo tan crítico como la informática del conocimiento.



Nuevos productos de Wang Computers

Wang Computers es una de las empresas con más solera en el campo del proceso de textos y en la automatización de oficinas, junto con el tema de la compatibilidad IBM.

Wang ha anunciado la disponibilidad del Wang Advanced PC y el Wang Office Assistant.

El primero, es un ordenador multipuesto y multitarea, dotado con 512 K de Ram en su configuración base y que corre bajo el sistema operativo XENIX (una versión de UNIX creada por Microsoft para el AT de IBM) y dirigido, por supuesto, a la gestión y al mundo profesional.

El segundo, Wang lo define como un producto de quinta generación, y como el «ordenador de la secreta-ria».

CASUALIDADES Y DEFINICIONES

Siempre hemos pensado que dentro del universo de los números «todo» ocurre dentro de un orden lógico establecido. Vamos a jugar a crear con nuestra imaginación un mundo en el que no calculamos números a través de complicadas fórmulas matemáticas, sino que van a surgir «por casualidad».

T

ambién vamos a tratar sobre la definición de funciones propias. El hecho de ser usuarios nos da libertad de creación. Vamos a convertirnos en algo así como dueñecillos de nuestro mundo microinformático. ¡Qué barbaridad!

En la vida real hay multitud de sucesos que ocurren «al azar», es decir: por casualidad. Por ejemplo, están tomando últimamente mucho auge los juegos de azar: Lotería Primitiva, Bingo, etc... Vamos a intentar aplicar la velocidad y obediencia con que trabaja nuestro **Amstrad** para simular estos casos en los que interviene el azar.

El azar también tiene sus reglas

Para ello, utilizamos los números aleatorios. Pero, **¿qué es un número aleatorio?** Cuando sacamos un billete de lotería y no elegimos el número, estamos ante uno de ellos: el vendedor de lotería nos dará al azar un número sin que éste haya sido calculado anteriormente. Nos ha dado un determinado décimo, pero podría habernos dado cualquier otro billete. Por lo tanto, un número aleatorio es, como el billete de lotería que hemos comprado, un número que hemos obtenido al azar.

Función RND

Bien. Pues con nuestro ordenador podremos meter un poco de **casualidad** en los programas. Con la función RND el **Amstrad** nos permite obtener números aleatorios.

Si tecleamos:

```
aleatorio=RND
```

seguido de

```
PRINT aleatorio
```

nos aparecerá en la pantalla un número decimal comprendido entre 0 y 1 (*mayores o iguales que 0 y menores que 1*). Es muy importante tener en cuenta que el número aleatorio así obtenido puede a veces valer 0, pero nunca alcanzará el valor 1.

En realidad el ordenador no producirá números al azar sino que mediante un algoritmo produce una secuencia cíclica de números que parecen aleatorios. Sería casi imposible crear una función que «fabricara» números que no siguieran ninguna ley. Pero no se asuste porque no podemos conseguir unos números totalmente casuales ya que tenemos 65536 números diferentes donde elegir.

Vamos a intentar generar varios números al azar con el programa 1.

Es simplemente un bucle que permite que aparezca en la pantalla la sucesión de los 20 primeros números aleatorios de la secuencia. Observe que todos son diferentes y están comprendidos entre 0 y 1.

Pero generalmente necesitaremos valores que estén comprendidos en un rango más amplio. ¿Qué ocurre si cambiamos la línea 40 por:

```
40 aleatorio=RND*10
```

en el programa 1? Cámbiela y vuelva a ejecutar el programa.

Ahora, los valores que nos salen en la pantalla han cambiado. Son números que, en vez de estar comprendidos entre 0 y 1, tienen un rango de 0 a 10. Multiplicando RND*10 hemos conseguido un margen de valores más amplio.

3455

6892

568

5734

Primeros pasos

De esta forma podemos conseguir números aleatorios comprendidos dentro de cualquier intervalo. No olvide que tanto en este caso con el número 10 como en general con el valor que pongamos, nunca lograremos que nos salga el valor máximo, ya que RND nunca nos devolverá un 1.

Pruebe poner en la línea 40 factores diferentes y verá que se cumple lo que le decimos.

Pero esto puede que no nos sea muy útil todavía. Así obtendremos largos números decimales, pero nuestro amigo el lotero nunca nos dará un décimo de este tipo: siempre nos dará un número entero. **¿Qué hacer para lograr esto?**

Es muy fácil. Cambiemos ahora la línea 40 por:

```
40 aleatorio=INT (RND*10)
```

y convertiremos los números anteriores en enteros comprendidos entre 0 y 9. Observe que no nos sale nunca el valor 10, por lo que comentábamos antes.

Con la función INT recortamos todos los valores obtenidos por el producto $RND*10$ y les dejamos solamente con la parte entera. De esta manera sí que podemos pedir un número al vendedor de lotería sin que se ría de nosotros a causa de los decimales.

1179

0134

Los números aleatorios permiten simular muchas situaciones de la vida real

Spongamos que queremos simular los números que saldrían al lanzar un dado. Los valores obtenidos al tirar el dado estarían comprendidos entre 1 y 6. Por tanto, el factor por el que tendríamos que multiplicar RND ha de ser 6.

Con el programa 2 obtendremos un simulador de las tiradas de un dado.

Lo único nuevo de este programa es que en la línea 30 hemos introducido una pregunta: **Número de tiradas**. Este valor metido en la variable **tiradas**, nos sirve para repetir el bucle que nos calcula el número aleatorio tantas veces como lanzamientos queramos hacer. Cada vez que se recorre el bucle es como si tirásemos el dado.

En la línea 60 obtenemos el valor del número al azar. Al multiplicar por 6 obtenemos con la función INT números enteros comprendidos en-

tre 0 y 5. Para alcanzar los valores del dado **¿Qué hacemos?** Sencillamente sumarle 1 en la línea 70.

Así que con este pequeño programa hemos conseguido un dado **«electrónicamente computerizado»**. Impresionado, **¿verdad?** Podemos jugar al parchís con ayuda de nuestro ordenador.

Vamos a hacer una prueba. Desenchufe y enchufe su ordenador. Cargue a continuación el programa 11 y ejecútelos. En la pantalla le aparecerán una serie de cifras después de haberle dicho el número de tiradas que desea hacer. Anote estas cifras. Vuelva a repetir la operación desde el principio (*desenchufar*). Observará que la serie que ahora le sale es exactamente igual que la de antes. **¿Qué ha ocurrido?**

¿Qué significa pseudorandom?

Sencillo. Al principio del artículo dijimos que el ordenador produce una secuencia cíclica de números con apariencia aleatoria. Es necesario añadir que al enchufar el ordenador dicha secuencia comienza por el primer número. Por tanto, si no inventamos algo para que esto no ocurra, siempre obtendremos las mismas series.

Pero nuestro **Amstrad** ya ha previsto esto. La palabra clave que utiliza para sacarnos de semejante apuro es **RANDOMIZE**.

Con esta orden le decimos al ordenador la posición de la secuencia desde la que queremos empezar. Teclée: **RANDOMIZE (1) y PRINT RND**

Nos aparecerá un valor en la pantalla. Repitamos el proceso pero tecléando:

```
RANDOMIZE (2000)
seguido otra vez de
PRINT RND
```

y el valor que nos sale es diferente del primero. Si volvemos a escribir:

```
RANDOMIZE (0) y
PRINT RND
```

nos volverá a aparecer el primer número aleatorio que obtuvimos.

En el caso que no pongamos la posición en la que queremos que comience la sucesión de números, el ordenador nos la preguntará haciendo aparecer en la pantalla el mensaje:

Random number seed?
algo así como: **¿Número aleatorio preseleccionado?**

La resolución al problema que nos plantea nuestro programa II sería añadirle la siguiente línea:

5 RANDOMIZE

y contestándole con distintas posiciones de comienzo obtendríamos tiradas muy diferentes en nuestra partida de parchís.

La función RND también admite parámetros. Si el parámetro es positivo, RND nos da el valor del siguiente número aleatorio en la secuencia cíclica.

Si el parámetro es 0, nos devuelve el número anterior.

Y si es negativo y comprendido entre -1 y -16, nos entrega el valor correspondiente y hace que la secuencia vuelva a comenzar desde el principio.

Los parámetros de la función RND

Vamos a intentar con el programa III poner sus ideas en orden.

En la línea 30 nos pregunta el ordenador el parámetro que llevará la función RND y, dependiendo del valor de la variable «parámetro», generará una cuestión u otra. Pruebe con todos los parámetros que se le ocurran —positivos o negativos— apunte los valores conseguidos y estamos seguros que sacará unas conclusiones acertadas.

No se piense que ya ha terminado sus peregrinar por los números aleatorios. Le proponemos que insista en probar al máximo estas dos palabras clave y sus parámetros para conseguir en cualquier momento la secuencia de números aleatorios deseada.

Y por azar hemos llegado a una de las pautas de las funciones numéricas en la que podemos desarrollar nuestro espíritu de creación: vamos a definir funciones. **¿Quién ha dicho que la técnica va en contra de la imaginación?**

En algún momento quizá haya necesitado utilizar una función que realice una serie de cálculos numéricos para obtener un determinado valor que necesitamos dar a una variable.

Si solamente los realizamos una vez no hay ningún problema, los hacemos y se acabó.

Pero vamos a suponer que dichos cálculos se tienen que repetir muchas veces a lo largo del programa y para diferentes parámetros. Sobre todo si son muy complicados y frecuentes **¡estamos apañados!**

Nuestro **Amstrad**, que es muy obediente y siempre está dispuesto a echarnos una mano, se ha provisto de una herramienta que nos va ser muy útil para ahorrar trabajo e instrucciones a nuestro programa, permitiéndonos definir nuestras propias funciones.

El arte de definir funciones

Para crear dichas funciones utilizamos la orden DEF FN seguida de un nombre en algún sitio del programa. Cada vez que necesitamos usarla la llamaremos con FN y el nombre anteriormente definido. Pero vamos a dejarnos de cosas teóricas y veamos un ejemplo. Vamos a definir una función que nos calcule la hipotenusa de un triángulo rectángulo conociendo los dos catetos en el programa IV.

En la línea 30 hemos definido la función con el nombre «**Pitágoras**». En las variables que están entre paréntesis pasaremos a la función los valores de los catetos.

Lo que va detrás del signo = es la definición real de la función. Podemos poner aquí cualquier expresión o conjunto de expresiones que utilizando los valores de los parámetros obtengan los que nosotros necesitamos.

Después de definida la función ya podemos invocarla desde cualquier parte del programa como si se trata de una de las internas del ordenador.

A continuación en las líneas 40 a 70 damos a las variables «**cateto 1**» y «**cateto 2**» los valores que vamos a pasarle a la función como parámetros para que ésta realice los cálculos con los que nosotros la hemos definido. También comprobamos que la longitud de los catetos sea mayor que 0.

Calculamos el valor de la hipotenusa en la línea 80. Aquí el número que nos devuelve la función «**Pitágoras**», a la que pasamos los contenidos de «**cateto 1**» y «**cateto 2**», lo metemos en la variable «**hipotenusa**».

Y por fin en la línea 100 imprimimos el resultado de nuestros cálculos. Con esta forma de programar, además de ganar en trabajo e instrucciones, la claridad de nuestros programas es mucho mayor ya que si conocemos exactamente qué es lo que hace la función, cada vez que nos encontremos con su nombre sabremos a qué cálculos corresponde el valor que nos devuelve y no será necesario seguirlos uno por uno.

Y para terminar con esta visión general del mundo de los números le vamos a proponer una «**fórmula genial**» para hacerse millonario: **¡Hemos encontrado la solución a la Lotería Primitiva!**

Mediante el programa V vamos a poder contar desde ahora con multitud de amigos «**ricos y famosos**».

Su estructura es muy sencilla. La línea 20 nos pone la pantalla con 20 caracteres por cada fila. En la siguiente —30— colocamos el origen de la secuencia de números aleatorios en una posición determinada por el tiempo (*TIME*) que lleva encendido el ordenador.

Definimos la función «**aleatorio**» en la línea 40. No tiene parámetros y nos devuelve un número entero al azar comprendido entre 1 y 49.

El siguiente paso es calcular cada uno de los seis números de nuestro pronóstico, mediante la función que hemos definido, y comprobar que no se repiten, líneas 50 a 250. Y para que aparezca nuestra puesta en la pantalla utilizamos las líneas 260 a 350. Fácil, ¿no?

Ojalá que estos artículos sobre los números hayan contribuido a hacer de su manejo una cosa agradable y atractiva. Es un campo donde uno se puede detener. Por eso le pedimos que analice, pruebe, investigue... Esperamos además que la próxima semana este último programa le haya servido para algo más que lo puramente informático.



```

10 REM PROGRAMA I
20 CLS
30 FOR contador=1 TO 20
40 aleatorio=RND
50 PRINT aleatorio
60 NEXT contador

```

```

10 REM PROGRAMA II
20 CLS
30 INPUT "NUMERO DE TIRADAS: ",tiradas
40 PRINT
50 FOR contador=1 TO tiradas
60 aleatorio=INT(RND*6)
70 dado=aleatorio+1
80 PRINT dado
90 NEXT contador

```

```

10 REM PROGRAMA III
20 CLS
30 INPUT "PARAMETRO DE RND: ",parametro
40 PRINT "RND(PARAMETRO)="RND(parametro)
50 PRINT"COMIENZO DE LA SERIE"
60 FOR contador=-10 TO -7
70 PRINT RND
80 NEXT contador
90 PRINT
100 GOTO 30

```

```

10 REM PROGRAMA IV
20 CLS
30 DEF FNPitagoras(x,y)=SOR(x*x+y*y)
40 INPUT "CATETO MAYOR: ",cateto1
50 IF cateto1<=0 THEN GOTO 40
60 INPUT "CATETO MENOR: ",cateto2
70 IF cateto2<=0 THEN GOTO 60
80 hipotenusa=FNPitagoras(cateto1,cateto2)
90 PRINT
100 PRINT "HIPOTENUSA = ";hipotenusa

```

```

10 REM PROGRAMA V
20 MODE 0
30 RANDOMIZE TIME
40 DEF FNaleatorio=INT(RND*49)+1
50 numero1=FNaleatorio
60 numero2=FNaleatorio
70 IF numero1=numero2 THEN GOTO 60
80 numero3=FNaleatorio
90 IF numero1=numero3 THEN GOTO 80
100 IF numero2=numero3 THEN GOTO 80

```

```

110 numero4=FNaleatorio
120 IF numero1=numero4 THEN GOTO 110
130 IF numero2=numero4 THEN GOTO 110
140 IF numero3=numero4 THEN GOTO 110
150 numero5=FNaleatorio
160 IF numero1=numero5 THEN GOTO 150
170 IF numero2=numero5 THEN GOTO 150
180 IF numero3=numero5 THEN GOTO 150
190 IF numero4=numero5 THEN GOTO 150
200 numero6=FNaleatorio
210 IF numero1=numero6 THEN GOTO 200
220 IF numero2=numero6 THEN GOTO 200
230 IF numero3=numero6 THEN GOTO 200
240 IF numero4=numero6 THEN GOTO 200
250 IF numero5=numero6 THEN GOTO 200
260 CLS
270 PRINT:PRINT:PRINT
280 PRINT" LOS NUMEROS QUE TE VAN A HACER MILLONARIO SON: "
290 PRINT:PRINT
300 PRINT TAB(9);numero1;PRINT
310 PRINT TAB(9);numero2;PRINT
320 PRINT TAB(9);numero3;PRINT
330 PRINT TAB(9);numero4;PRINT
340 PRINT TAB(9);numero5;PRINT
350 PRINT TAB(9);numero6;PRINT

```

Primeros pasos



Para que tus dedos no realicen el trabajo duro. M. H. AMS-TRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un consejo mensual, solicitados.

MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS PARA QUE VD. NO TENGA QUE TECLEARLOS. TOTALMENTE DESPROTEGIDOS CON EL OBJETO DE FACILITAR SU COPIA EN DISCO.

Todos los programadores y aficionados a la microinformática, sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y evitar que malgastes largas horas sobre el teclado de tu ordenador tratando de descifrar incomprensiblemente mensajes de error.

AMSTRAD SEMANAL te ofrece cada mes los programas publicados en los cuatro números correspondientes, en una cinta de cassette desprotegida, que te permitirá copiar los programas en disco y tener acceso a los listados para su estudio y posterior edición de rutinas.

Programas incluidos en la cinta número 1			
Título	Rev. n.	Título	Rev. n.
EASYDRAW	1	MAD ADDER	3
EGGRLITZ	2	HELEX	3
E CODIGO SECRETO	2	CHANGEN	4
VENTANAS BORRANTIS	3	PROGRAMACION	4

Programas incluidos en la cinta número 2			
Título	Rev. n.	Título	Rev. n.
GRAFICOS	8	INCOGNITION	7
MUSICA	6	MONITOR	5
TRON	6	ANAUSIS	5-8
ENSAMBLADOR	8	CEBRIC	8
HEBEL	8	ANIMACION1	7
TOCKIT	8	ANIMACION2	8
PRIMEROS PASOS	7	SMALLEY	7

Programas incluidos en la cinta número 3			
Título	Rev. n.	Título	Rev. n.
ANALISIS	9-12	SPRITE112	11
FRUITES	9	AMKARD	11
MENUSDISC	9	OTELLO	12
K5x101	10	EVENT121	12
K5x102	10	EVENT122	12
CUATRODATAS	10	FRUITES	12
SPRITE111	11		

Por sólo 675 pts.
(incluidos gastos de envío)



COMPATIBLES
CON LOS MODELOS
CPC-464, CPC-664
y CPC-6128

Recíbelos en tu casa
cómodamente enviándonos
con la menor demora
posible, el cupón que se
encuentra en la última
página de la revista

DISPONIBLE PARA ZX SPECTRUM
AMSTRAD

SOFTWARE

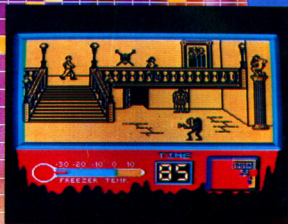
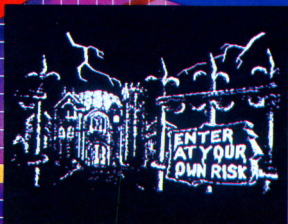
Sound-on-Sound
JUEGA CON EL FUTURO

Sound on Sound es una marca registrada
producida y distribuida por Iberón, s. a.
Tel. 971.22.00 / 04/09/12/16



¡¡¡NO LO SUENES!!! ¡JUEGALO!
SIENTE LA EMOCION DE LO DESCONOCIDO
CORRE TU PROPIO RIESGO
SALVA A TU COMPAÑERO... ¡¡¡ ATRAPADO/A
REUNE LOS FRAGMENTOS DEL CUADRO
SON TU AMULETO

¡¡¡POR FIN EN CASTELLANO!!!
LA PRIMERA COMEDIA MUSICAL EN VIDEO-JUEGO



Los secretos del teclado

Muchos son los ordenadores que disponen de una posibilidad, tan desconocida como útil: la definición de teclas por el usuario, para esto disponen de un segundo teclado, adosado al de tipo QWERTY convencional, son las teclas de usuario o función.



El número de éstas, aunque variable, suele oscilar entre las 10 y las 20, según marcas y modelos. En el caso de la familia CPC son doce las teclas de este tipo. Y naturalmente, cómo no, podemos disponer de ellas a nuestro antojo aprovechando la potencia del **Basic Locomotive**. Sin embargo, no es precisamente a mediocridades a lo que nos tiene acostumbrados este Basic. No contentos los fabricantes con esta posibilidad, extendieron ésta a todo el teclado y lo que es más importante, con una facilidad de uso sorprendente. Pero, como una imagen vale más que mil palabras, teclee el programa 1 y ejecútelo. Ahora, intente escribir cualquier cosa. ¿Sorprendido? Bien, ése era el objetivo. Pero no adelantemos acontecimientos y vayamos por partes.

Hemos hablado de dos opciones que, si bien muy parecidas, son totalmente distintas en lo que a su uso y utilidad respecta. En la primera de ellas nos referíamos a la definición del teclado de función, éste se encuentra ubicado en la parte derecha del teclado, bajo las teclas que controlan el movimiento del cursor en el caso de los dos modelos más viejos, y sobre éstas en el caso del 6128. Estas teclas se encuentran numeradas del 0 al 9 y en los dos modelos de disco se ha incorporado una f minúscula acompañando a estos dígitos, con el objetivo de recordar que la verda-

dera utilidad de estas teclas no estriba en su empleo numérico sino en la posibilidad de redefinirlas conforme a las necesidades de cada usuario.

El comando KEY

La definición de estas teclas es muy sencilla, basta con:

KEY N, cadena alfanumérica

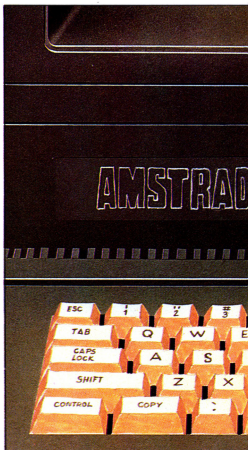
Donde N es un entero no negativo menor o igual que 12. Y cadena alfanumérica es una cadena encerrada entre comillas o el nombre de una variable, evidentemente, también alfanumérica. El número corresponde al número de la tecla de función 0 para f0, 1 para f1, etc... Los números 11 y 12 quedan reservados para las teclas marcadas con un punto (.) y ENTER, las más pequeñas de las dos. Veamos un ejemplo. Ponga manos al teclado y:

KEY 5, **esta tecla la he definido yo**

Pulse enter y observe lo que sucede cada vez que pulsa la tecla f1. ¿Sencillo verdad? Ahora ya es capaz de definir cualquiera de esas doce teclas. Intentemos ahora dar el salto al resto del teclado, no obstante detengámonos por un momento y pasemos a un tema que aunque distinto, su conocimiento nos será, más que útil, imprescindible.

Caracteres de control

Al leer el capítulo 7 del manual, llama la atención un apartado cuyo título reza, CODIGOS DE CONTROL. A primera vista, difícilmente podemos intuir cuál es la misión de

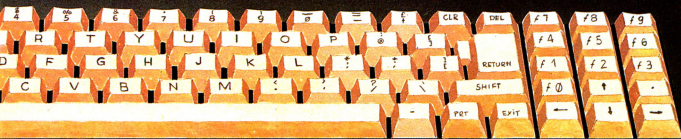


estos cabalísticos códigos, no es que sea complicada su utilización, sin embargo, sí puede resultar más que difícil el encontrar una aplicación lo suficientemente satisfactoria y/o elegante como para que su uso quede justificado. Es por esto, que no podíamos dejar pasar esta oportunidad, sin riesgo de ser acusados de necios.

Los códigos correspondientes a los caracteres comprendidos entre el 0 y el 32 están reservados a una utilización como códigos de control por el sistema. Así el envío de uno de estos caracteres a la pantalla, lejos de producir un garabato más o menos barroco, da como resultado extraños efectos, a menudo indeseados por desconocimiento de su utilización. No entraremos en detalles sobre la explicación detallada de cada uno de estos caracteres, no sería sino redundar en el manual y nada más lejos de nuestra intención. Lo único es soslayar la ausencia de explicación ante los códigos de control con parámetros, uno, dos o incluso nueve como es el caso del código EM (H19 o D25). Pero veamos primeramente y a modo de ejemplo, uno de estos caracteres de control sin parámetros. Teclee por ejemplo:

PRINT CHR\$(7)

CTM 644
colour Monitor



¿Piip? Correcto, lo que usted ha enviado a la pantalla es el carácter de control 7, cuyo nombre es BEL, no intente que su ordenador comprenda esta palabra, es sólo un memónico. El efecto que produce ya lo habrá observado (oído). Existe un grupo de estos códigos que precisan de la compañía de uno o más parámetros. Si queremos por ejemplo poner la pantalla en modo 2 (80 columnas), mediante el uso de estos caracteres, teclearemos:

```
PRINT CHR$(4)+CHR$(2)
```

El número 4 indica que se escriba el código de control 4, cuyo nombre es EOT y que el valor del parámetro sea 2.

Uno de los caracteres que más emplearemos será el CHR\$(13) (cuidado no confundir el hexadecimal con el decimal). Su nombre CR, equivale a la tecla ENTER o RETURN. Mediante este carácter podremos hacer que la acción expresada mediante nuestra tecla se autoejecute nada más pulsarla.

Definamos la tecla 1 del siguiente modo: KEY 1, «border»+rnd*27+chr\$(13)

Ahora basta la pulsación de la tecla 1 para asignar al borde de la pantalla un color aleatorio.

Podemos preguntarnos por qué razón no hemos utilizado el código GS (h1D o d29) para la definición del borde. Intentemos realizar esto con el carácter BEL, teclee:

```
KEY 2,CHR$(7)
```

Y ahora pulse la tecla f2.

¿Decepcionado? En vez del esperado piip, nos encontramos con un diminuto platillo volante. Volvemos a presionar f5, f5, f5, f5... No hay respuesta y cada vez la pantalla recuerda más a una invasión extraterrestre. No, no es que el manual mienta, lo que sucede es que estamos enviando estos códigos desde el editor, lo cual equivale a enviarlos a la pantalla gráfica. Desconozco la razón de por qué éstos no vienen descritos en el manual con el mismo detalle como han sido tratados aquellos que tenían un código superior al 32. Sin embargo, el programa 2 le muestra todos ellos, pudiendo utilizar todos éstos en sus programas, sin más que anteponer al CHR\$(N), CHR\$(1)+.

Podemos, sin embargo, eludir este problema mediante:

```
KEY 2, «PRINT  
CHR$(7)+CHR$(13)
```

¿Pita? Claro, ahora sí. Para evitar teclear PRINT, a partir de ahora ten-

dremos únicamente ?, que para nuestro CPC significa lo mismo. Apliquémoslo al ejemplo anterior:

```
KEY 2, «? CHR$(7)+CHR$(13)
```

Pero, y si unimos varias de estas sentencias lo que obtenemos a fin de cuentas es...? Efectivamente:

Un programa en una tecla

¿Y por qué no? Teclee por ejemplo:

```
KEY 2, «MODE 2: FOR N=1 TO  
1000: PRINT N,PEEK (N):  
NEXT»+CHR$(13)
```

Ahora la tecla f2 provocará la impresión del contenido de las 100 primeras posiciones de la RAM de su CPC.

Pero cuidado, no todo el monte es orégano, tenemos una limitación, 120 caracteres es el total que puede alcanzar el conjunto de «frases» creadas, a no ser a consta de un IMPROPER ARGUMENT, es por esto, que es una buena medida anular los tokens expandidos cuando ya no van a ser utilizados o simplemente hacer un reset, siempre y cuando esto sea posible.

Intente definir la tecla 3 para que le dé la memoria libre. FRE(0) es la función que le devuelve este valor.

Y ahora pruebe a definir la tecla 4 de forma que llene la pantalla con un caracter, por ejemplo «#». Al final del articulo encontrará una solución posible.

Y el teclado es suyo

Hasta aquí todo lo relativo al teclado de función, veamos ahora cómo definir el resto de las teclas. El mecanismo es algo más complejo, aunque no por ello complicado.

Cada tecla dispone de un código, el código de tecla, véase figura 1. Así la tecla asignada actualmente a la A tiene el código de tecla número 69. Existe otro concepto fundamental, código expansible. Realmente lo que hemos hecho hasta ahora es trabajar con estos códigos expansibles sólo que era nuestro CPC el que hacía el trabajo sucio. La razón es que las teclas f0, f1, f2... f9, (,) y (ENTER), tienen asignadas ya estos códigos. Veamos cómo se organiza internamente el teclado y comprenderemos esto. Cada tecla realmente no puede almacenar más que tres números, los cuales corresponden al código ASCII de los distintos caracteres que provocan según se la pulse sola o acompañada de SHIFT o CTRL. Bueno esto es cierto a medias, esto sólo sucede en el caso de que este número sea inferior a 128 o superior a 159. Todos los códigos comprendidos en este intervalo, lo que hacen es informarle de que el contenido de esa tecla, de la tecla en cuestión, se ha expandido mediante el código expansible con ese número. El contenido de las teclas de función no es, sino un código expansible para cada una de ellas del modo siguiente:

f0	128
f1	129
...	...
...	...
f9	138
(,)	139
ENTER	140

Ahora bien. ¿De qué forma asignamos una variable a uno de estos códigos? Muy sencillo, mediante la orden KEY arriba mencionada. Por ejemplo y volviendo al primer ejemplo, si queremos poner en el que tienen como código expansible el número 129 **ESTA ES MI TECLA**, escribiríamos:

KEY 129, **ESTA ES MI TECLA**

Ahora tendremos que asignar a la

PROGRAMAS

```
10 REM-----program 1-----
20 KEY 128," ESTO NO SIRVE "
30 FOR N=0 TO 70
40 KEY DEF N,1,128,128,128
50 NEXT N
```

```
10 REM-----PROG II-----
20 REM
30 MODE 1
40 FOR n=0 TO 32 STEP 3
50 PRINT N;" " ;CHR$(1)+CHR$(N);"
   ";N+1;" " ;CHR$(1)+CHR$(N+1);"
   ";N+2;" " ;CHR$(1)+CHR$(N+2)
60 PRINT
70 NEXT
```

```
10 REM-----prog 3-----
20 REM-----soluciones-----
30 REM
40 KEY 3,"? fre(0)" +CHR$(13)
50 KEY 4,"mode 1:for n=1 to 1000:pr
int chr$(35);next" +CHR$(13)
60 CLS:PRINT "DEFINIDAS TECLAS F3 Y
F4"
70 NEW
```

```
10 REM -----PROG IV-----
20 REM -----HELPROG-----
30 REM
40 DATA PRINT,INPUT,LOCATE,MODE,GOS
UB,NEXT,GOTO,FOR,BORDER,WINDOW,WHIL
E,CLS,USING,LINE INPUT,THEN,ASC,AFT
ER,EVERY,IF,RETURN,END,STEP,OPEN,CL
OSE,DIM
50 DATA p,27,o,34,k,37,m,38,h,44,n,
46,g,52,f,53,b,54,v,55,w,59,c,62,u,
42,l,36,t,51,a,69,z,71,x,63,i,35,r,
50,e,58,s,60,q,67,y,43,d,61
60 FOR N=1 TO 25
70 READ A#
80 KEY 127+N,A#
90 NEXT N
100 FOR N=1 TO 25
110 READ car#:cod=ASC(car#)
120 READ A
130 KEY DEF A,0,cod,127+N
140 NEXT
150 CLS:PRINT "PROGRAMA HELPROG EJE
CUTADO"
160 PRINT "TECLAS ALFABETICAS MAS S
HIFT PROVOCAN COMANDOS"
170 *NEW
```

tecla que deseamos expandir, este código. Esta labor la realizamos mediante una nueva sentencia KEY DEF.

La forma de utilizarla es muy sencilla:

KEY DEF N1, N2, N3, N4, N5

El primer parámetro expresa el número de tecla que queremos modificar, éstos están representados en la figura 1. El segundo parámetro admite dos valores 1 ó 0 dependiendo de que la tecla en cuestión deseemos que se repita al mantenerla pulsada o que para que esto suceda haya que volver a pulsarla.

El tercer, cuarto y quinto parámetros, son números en la gama de 0 a 255, en el caso de que éstos se encuentren entre 128 y 159 obtendremos la llamada al comando expandible con ese código. El primero de estos códigos permite la definición de la tecla cuando se la pulsa sola el segundo cuando se la pulsa con SHIFT y el tercero con CTRL. Así hagamos: KEY 130, **ESTA ES MI TECLA + SHIFT** KEY 131, **ESTA ES MI TECLA + CTRL**

Y ahora:

KEY DEF 69, 1, 129, 130, 131

Pulse ahora la tecla **A** sola, con SHIFT y con CTRL.

Haga ahora:

KEY DEF 67, 0, 129, 130, 131

Juegue ahora con las teclas **Q** y **A**, observe el diferente resultado que se obtiene al mantener pulsada la «Q» o la «A». La diferencia de definición está en el segundo parámetro 0 (no se repite) o 1 (repetición).

KEY 131, **ESTA ES MI TECLA + CTRL**

Y ahora:

KEY DEF 69, 1, 129, 130, 131

Pulse ahora la tecla **A** sola, con SHIFT y con CTRL.

Haga ahora:

KEY DEF 67, 0, 129, 130, 131

Juegue ahora con las teclas «Q» y «A», observe el diferente resultado que se obtiene al mantener pulsada la **Q** o la **A**. La diferencia de definición está en el segundo parámetro 0 (no se repite) o 1 (repetición).

Pulse **F0, F1...**, naturalmente que otra cosa esperaba, intente casar este hecho con lo explicado líneas atrás.

Bien hasta aquí prácticamente todo lo que al Basic respecta, en la segunda parte de este artículo intentaremos desvelar algunos misterios en torno al editor y cómo se almacena el teclado en la memoria de su CPC, así como la explicación de algunas utilidades que, sin duda, harán más simple y eficaz la conversación con nuestro **Amstrad**.

Aún queda, no obstante, un pequeño programa que intenta aprovechar estas utilidades (prog. 4). Su misión es el asignar a unas cuantas teclas una sentencia Basic entre las más utilizadas. Mediante el uso de éste antes de teclear cualquier programa, no sólo ahorraremos al escribir sino que evitaremos la tediosa tarea de corregir errores de sintaxis. La figura 2 muestra la ubicación que en el teclado tienen estas sentencias, para su obtención habrá que pulsarlas junto son SHIFT. Una vez ejecutado el programa se autodestruye, por lo que tenga la precaución de salvarlo en cinta o disco, antes de su ejecución.

Al igual que hemos hecho en este programa podemos redefinir nuestro teclado para que escriba en árabe, arameo o en letra cirilica, o cambiar la disposición de las teclas en atención a una mayor velocidad, pero esto, ya es otra historia.



Amstrad Ideas

AMSTRAD Semanal comunica a todos sus lectores la apertura de una nueva sección dedicada a recoger las mejores ideas que exploten al máximo las posibilidades del ordenador, materializadas en programas claros y cortos (máximo 25 líneas). Los mejores de entre todos ellos serán publicados con el nombre de su autor en la revista, recibiendo como premio, gratuitamente en su domicilio los cuatro primeros números de nuestra cinta mensual. Los programas enviados deberán incluir:

— Cinta de cassette con el programa o programas grabados.

— Explicación detallada del funcionamiento y propósito del programa, mecanografiado a 2 espacios o con letra clara.

Es imprescindible indicar en el sobre claramente: **AMSTRAD IDEAS**.

La dirección es:

Hobby Press, S. A.

La Granja, s/n.

Polígono Industrial de Alcobendas.

Madrid

SOFTWARE

Especialistas
en
SOFTWARE
para la
pequeña y
mediana
empresa

- AMSTRAD 6128
- AMSTRAD 8256
- IBM y compatibles

Programas de gestión
integrados

C/. Galileo, 25
Entrepantana A
Tels. 447 97 51 - 447 98 09
28015 Madrid

ANÁLISIS PARABOLOIDE

En la intrincada ciencia de la geometría, existen unas extrañas formas llamadas superficies regladas.

Estas son generadas por una serie de rectas que se apoyan en unas directrices, dando lugar a formas de inusitada belleza.

Análisis, esta vez trata sobre la creación de estas superficies, con un pequeño programa que te asombrará por sus resultados.

10-20 Título del programa.

30-70 Líneas que se encargan de dejar la pantalla, en condiciones idóneas para ejecutar el programa.

40 Elige como color del papel el negro, color del borde también el negro y color de la tinta el verde mar.

Obsérvese que al cambiar el color del papel y de la tinta con el comando INK, éste se realiza automáticamente sin tener que hacer un CLS.

50 INPUT, que nos solicita el número de líneas que van a formar nuestra superficie, su valor se almacena en la variable n.

60 Limpia la pantalla, para evitar que el mensaje anterior permanezca en la misma y estorbe a lo que va a aparecer posteriormente.

70 Seleccionamos el modo 2 de pantalla, porque en este la resolución es mayor y así los gráficos adquieren más definición.

80-110 Establecen las coordenadas de los vértices del primer paraboloide.

90 Coordenadas del vértice inferior izquierdo.

100 Coordenadas del vértice a su derecha.

110 Coordenadas del vértice superior.

120-140 Establecen los cálculos preliminares, que nos permitirán determinar la posición de cada recta, en las directrices.

130 modxa y modxb, son los valores resultantes de restar las coordenadas del punto 2, a las del punto uno y dividir el resultado entre el número de líneas elegida en la línea 50.

El objetivo de todo esto, es dividir la recta que une los puntos 1 y 2 en partes iguales.

140 La misma operación anterior, pero ahora con la recta que a los puntos 2 y 3.

150-220 Calcula las coordenadas de estos puntos y procede a dibujar las rectas que los unen.

160-170 Realizan un cálculo previo, con el objeto de dejar en condiciones de aplicación al ciclo FOR...NEXT de las líneas.

180-220. Al restar a las coordenadas x e y, los módulos obtenidos en las líneas 130, 140, entramos en el ciclo y con la primera rotación, al sumamos los módulos anteriores, obtenemos las rectas que unen los puntos x1, y1 y x2, y2.

180-220 Ciclo FOR...NEXT, que rota tantas veces como el valor de la variable n.

En cada rotación del ciclo, a las coordenadas anteriores p1,q1,p2,q2, se les suman los modxa, mody, modxb, modyb, obteniendo las coordenadas de los puntos que resultan de dividir cada recta en n partes.

Una vez calculadas éstas, la línea 210 se ocupa de dibujar en pantalla las rectas que unen dichos puntos.

230-280 Con todas las líneas anteriores hemos dibujado el paraboloide cuyas directrices son las líneas x1,y1-x2,y2 y x2,y2-x3,y3.

Con estas líneas alteramos los valores de las coordenadas, definiendo las nuevas directrices del paraboloide repitiendo el proceso de cálculo de puntos intermedios y dibujando la nueva superficie.

240 Asigna a x1,y1 los valores de x3,y3.

250 Da a x3, y3 los valores 635,0.

260-270 En primer lugar se incrementa en una unidad la variable control, la cual al no haber sido definida con anterioridad, tenía el valor cero.

La primera vez que pasa el programa por estas líneas, control toma el valor uno, la segunda dos.

La línea 270 comprueba si este valor es igual a 2, en cuyo caso finaliza el programa.

Cuando la variable control tiene el valor uno el programa, vuelve a la línea 130, procediendo a dibujar el segundo paraboloide.

280 Devuelve el control del programa a la línea 130.



200,270

0,0 320,0 635,0

Directrices y sus coordenadas

Es usted capaz de tomar el relevo del General Montgomery...

& Juegos ESTRATEGIA

le presenta en exclusiva
el WAR GAME, para Spectrum,
de mayor éxito en Inglaterra:

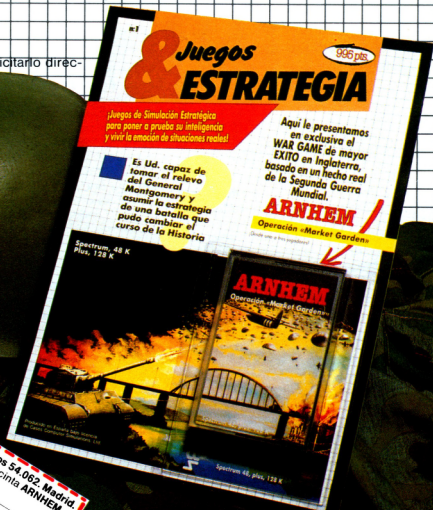
ARNHEM

(operación «Market Garden», basada
en un hecho real de la Segunda Guerra
Mundial)

Si no lo encontrara en su kiosco puede solicitarlo direc-
tamente a nuestra editorial sin
gastos de envío alguno por
su parte. No demore
su pedido, hay un
número limitado
de cassettes.

Y ahora también para
AMSTRAD
Compatible todos los modelos

Ya está a la venta!



Recorte o copie este cupón y envíelo a Hobby Press, S. A. Apdo. de Correos 54.062, Madrid.

Deseo recibir en mi domicilio, sin gastos de envío alguno por mi parte, la cinta ARNHEM.

Nombre _____ Edad _____

Dirección _____

Localidad _____

Código _____

La forma de pago elegida es la que señalo con una cruz

Giro Postal n.º _____

Tarjeta Visa n.º _____

Fecha de caducidad de la tarjeta _____

Fecha y Firma: _____

Provincia _____

Teléfono _____

Talón nominativo a Hobby

HOBBY PRESS, S.A. Editamos para gente inquieta.

YIE AR KUNG-FU

Las artes marciales siguen siendo un caudal inagotable, para la creación de juegos de ordenador.

En esta ocasión, se trata de una adaptación de un popular juego de las máquinas de los bares, el Yie Ar Kung-Fu.

B

ruce Luis era aún muy joven cuando su anciano maestro, comenzó a instruirle en el mundo del Kung-Fu.

Mientras transcurrían los largos años de adiestramiento, Luis se hacía cada vez más hábil en el dominio de su cuerpo.

También su mente se iba adecuando a la arcaica filosofía de las artes de combate.

Con el transcurso del tiempo, su cuerpo se iba esculpiendo a la medida de su nueva forma de vida, cobrando una agilidad felina, acompañada de una gran precisión en sus técnicas y un dominio absoluto de cada músculo del mismo.

Un día, al final del entrenamiento diario su maestro le dijo: «**mira a este dedo**» dejándole meditar unos momentos, le preguntó después: **¿qué ves?** Luis miraba fijamente en la dirección indicada, sin emitir palabra alguna.



El siguiente es Nuncha, el luchador diestro en el manejo de los Nunchacus, un arma que utilizada con ritmo y velocidad, no permite ninguna posibilidad al adversario, reduciéndole a la nada con pocos golpes.

A continuación le toca el turno de combate a Pole, el hombre venido de las fértiles regiones arrozíferas, cuyo arte de lucha se basa en la caña de bambú que manejada con destreza por su longitud permite mantener a raya cualquier contrario.

Club, el próximo en la lista, no tiene nada que ver con los anteriores, su técnica de lucha no se basa en la velocidad, su fuerte es la pegada.

Sus armas son de lo más efectivas, utiliza un escudo hasta los pies y una porra pesada, cuyos golpes son demolidores.

Sword, el maestro de la espada curva, con sus movimientos de defensa, en los que encontrar algún hueco es prácticamente imposible y que ejecuta los sablazos con una maestría impecable.

El maestro, por fin continuó: «**si al mirar hacia mi mano, te fijas solamente en el edo, perderás el universo maravilloso que te encuentras tras él, no verás los pájaros, ni los árboles, ni las flores, ni el sol ni las estrellas.**»

Bruce Luis miraba con la boca abierta hacia el anciano.

Tras esta magistral lección, le siguió diciendo: «ahora ya no puedo enseñarte más, has adquirido todo el saber y la fuerza de un verdadero luchador, el resto está en tus manos». «**Vete ahora y nunca emplees tus conocimientos en hacer el mal. Y recuerda que solamente podrás llegar a ser un verdadero hijo del dragón, cuando hayas vencido a todos los luchadores de las otras artes.**»

Tras esto, nuestro amigo abandonó a su chief y se dispuso a superar la prueba de fuego, el combate.

El primer adversario es el terrible Buchu, un peso pesado oriundo de las regiones del Himalaya, con el extraño poder de volar, unido a una desmesurada fuerza.

Superado este escoyo, nos enfrentamos a Star, bellísima mujer con el don y la gracia de las regiones orientales, dominadora del arte de lanzar las estrellas mortíferas, con una precisión implacable.



Compatible: CPC/464, CPC/664 y CPC/6128.





Mr. Joystick



Tonfun, el tailandés capaz de eliminar a cualquier diestro luchador, con sus efectivas Tonfas, arma defensiva y de ataque que disuade a cualquier valiente de entrar en combate.

Si Bruce Luis consigue superar a esta serie de expertos adversarios, aún le queda la prueba de fuego: Blues.

Éste fue el primer alumno de el maestro de Luis, pero no siguió sus acertados consejos y se ha convertido en el más malévolo de todos los luchadores, su destreza está por encima de los límites humanos y sólo un superdotado le logrará vencer.

Un juego de combate, en el cual nuestro amigo ejecuta 10 técnicas distintas, con un repertorio de movimientos más que aceptable.

Los gráficos son de tipo semicari-caturescos, con personajes con identidad propia y un decorado oriental con pagodas y cascadas, que ambienta los sucesivos combates.

El método de lucha, es por eliminación de forma que no podemos enfrentarnos al próximo rival, hasta no haber derrivado al que tenemos delante.

Lo que provoca el pique con la máquina para vencer a cada adversario, con el único empeño de ver cómo es el siguiente.

Los aficionados a las máquinas de los billares, saben lo que queremos decir.

DECODIFICADOR BINARIO

Ya que tengo que escribir una pequeña introducción a este programa creado por Alberto Suárez, comenzaré confesando que no sé muy bien si encuadrarla en la sección de Programación de la revista o en la sección de Amstradmilagros (por desgracia, esta interesante sección no es fija todavía). Todo empezó una tarde de esas en las que uno está harto de programar en general y de transpasar a mano código máquina a DATAS, en particular.

Acariciaba yo el sueño imposible de que esa tarea la realizara el ordenador, y tuve la suerte de que en ese momento pasara Alberto por allí y me viera la cara, entre languida y verde. Inmediatamente se acercó a enjugarme las lágrimas, y cuando le conté mis cuitas, me miró fijamente y dijo: «HUM... no sería imposible del todo», y desapareció, dejándose con la miel en los labios. Podía escucharse el ruido que hacía su cabeza según se alejaba. Dos días después, la rutina estaba lista y funcionando. Aún no me lo puedo creer.

T

odos estamos habituados a programar en Basic, por lo que es muy común ponerse delante del ordenador y empezar a escribir unas líneas que son las que van a constituir nuestro programa, sin preocuparnos de otra cosa que no sean los posibles errores de sintaxis que hoyamos podido cometer al introducir el listado.

Hasta ahí todo está muy claro, pero seguuro que la inmensa mayoría de nosotros nos hemos planteado en alguna ocasión la siguiente pregunta: **¿Cómo es posible que una serie de caracteres introducidos en un orden prefijado, den lugar a determinadas reacciones de nuestro ordenador?**

Pues bien, la respuesta está en su propio sistema operativo, que es el encargado de decodificar las líneas que nosotros escribimos desde Basic.

Una vez expuesto el programa deberemos intentar abrir caminos para el estudio de las interioridades de nuestro Amstrad, y averiguar en qué forma introduce un programa realizado en Basic en su memoria.

La primera pregunta que se nos plantea es la siguiente: **¿Dónde introduce el ordenador los datos que le damos desde Basic?** Pues bien, la solución es muy sencilla, la dirección de memoria #170 es la elegida como principio de los programas realizados en Basic.

Una vez dicho esto, podemos probar de ver qué datos se encuentran en dicha dirección si nosotros introducimos la siguiente línea de programa:

10 REM HOLA

Si desensamblamos a partir de la dirección #170 nos encontramos con lo siguiente:

```
#170 12 Longitud
#171 0 de la línea.
#172 10 Número
#173 0 de la línea.
#174 32 Código ASCII del carácter « »
#175 rem Código del TOKEN «REM».
#176 32 Código ASCII del carácter «.».
#177 h Código ASCII del carácter «H».
#178 0 Código ASCII del carácter «O».
#179 1 Código ASCII del carácter «L».
#17B a Código ASCII del carácter «A».
#17C 0 Código de fin de línea.
```

Como vemos, los dos primeros bytes indican la longitud de la línea y son almacenados de tal forma que el primer byte es el menos significativo y el segundo byte es el más significativo. De la misma forma almacena el número de línea en los dos bytes siguientes.

Observemos que la posición de memoria #17C contiene un cero y aun así la hemos incluido en el listado; esto es debido a que este byte le indica al sistema que ha acabado una línea Basic.

Como vemos el tema es apasionante, aunque en este artículo no pretendemos estudiar a fondo el problema, sino únicamente dar una introducción para aquellos usuarios que estuvieran interesados en su comprensión y análisis.

A partir de ahora nos centraremos en la explicación del programa que os presentamos, que es una rutina en código máquina que es capaz de generar un programa Basic.

La rutina en cuestión lo que hace es generar líneas «DATA» y luego introducir dentro de ella unos valores. Estos valores son extraídos de la posición de memoria que nosotros le indiquemos, luego son decodificados convenientemente para ser introducidos en otra posición de memoria en la cual el ordenador los trata como pertenecientes a un programa Basic.

Como vemos la utilidad del «generador de Datos» es inmensa, ya que es capaz de coger

un programa en código máquina que tengamos en memoria y poner todos los bytes que contiene el programa en líneas «DATA» Basic, despreocupándonos así de tener que teclear absolutamente nada.

El primero que hace la rutina es tomar varios datos que nosotros tenemos que darle desde Basic al hacer la llamada a la rutina. Para ello se utiliza el registro indexado IX, que apunta al último dato que nosotros damos en la llamada a la rutina.

Los datos necesarios para que la rutina funcione correctamente, deben ser los siguientes:

1. Número de línea en la cual queremos que empiecen a generarse las líneas «DATA».
 2. Dirección a partir de la cual tenemos nuestro programa en código máquina.
 3. Longitud de dicho programa.
- Estos datos debemos entrarlos de la forma que se indica a continuación:

CALL

&A000,(N.LINEA),(INICIO),(LONGITUD)

Una vez que la rutina coge estos datos iniciales, lo primero que hace es cargar en el registro HL, la dirección donde acaba el Basic, para empezar a colocar las líneas a partir de esa dirección.

Para obtener la dirección de fin de Basic, apuntamos a la variable #AE83 (si poseemos un Amstrad 464) o a la variable #AE66 (si tenemos una Amstrad 664 o un 128), esa dirección nos dirá dónde termina el programa Basic que tenemos en memoria.

El siguiente paso, será colocar a partir de esa dirección la primera línea «DATA», para ello colocamos en los dos primeros bytes la longitud de la línea, en los dos siguientes, el número de línea, luego colocamos el código del TOKEN «DATA» y por último colocamos los números correspondientes que son leídos de la dirección donde tenemos nuestro programa en código máquina.

Los bytes leídos a partir de la dirección indicada, son decodificados, y luego son tras-

18 FOR H\$A888 TO \$A133
 20 READ A\$:\$UM\$A
 30 POW N/A
 40 NEXT A\$
 50 IF \$UM\$(6,7,9) THEN PRINT "ERROR EN DATAS"
 70 DATA 54,161,221,119,2,221,182,134
 80 DATA 3,34,58,161,221,126,4,50
 90 DATA 50,51,161,221,126,5,50
 100 DATA 68,161,42,174,43,43
 110 DATA 221,42,58,161,42,8,58
 120 DATA 56,161,58,161,221,54,52
 130 DATA 161,285,93,168,42,93,161
 140 DATA 50,58,161,95,226,6,42
 150 DATA 8,58,58,161,225,84,52
 160 DATA 161,58,161,254,48,48
 170 DATA 220,35,35,34,182,174,34
 180 DATA 184,1174,24,186,174,34,188
 190 DATA 174,201,225,17,46,25
 200 DATA 6,7,225,282,168,54,44
 210 DATA 25,58,161,68,58,58,50
 220 DATA 161,221,35,177,237,75,54
 230 DATA 161,11,237,67,54,161,120
 240 DATA 177,325,62,1,58,54

AR5E 116R88 480 LD DE,6
 AR61 19 418 ADD HL,DE
 AR62 8687 426 LD B,7
 AR64 C0C4A8 438 OTR: CALL R8E
 AR67 352C 448 LD (HL),"
 AR69 29 456 INC HL
 AR6A 3630A1 460 LD A,(LON)A
 AR6D 3C 478 INC A
 AR6E 2235A1 480 LD (LON)A
 AR71 D023 498 INC B
 AR73 C5 508 PUSH BC
 AR74 E04E36A1 518 LD BC,(ADER)
 AR76 80 528 INC A
 AR77 E0433A1 538 LD (ADER),BC
 AR78 76 548 INC B
 AR7E 81 558 LD A,B
 AR7F 2895 568 JR NZ,SIQUE
 AR81 2891 578 LD (FLA)A
 AR83 3236A1 588 LD A,1
 AR85 C1 598 SIQUE POP BC
 AR87 1908 608 INC OTR: DJNZ OTR

AR8E 79 1118 PINTA: LD A,C
 AR8F C038 1128 ADD A,48
 AR97 73 1148 LD (HL),A
 AR98 3A22A1 1158 LD A,(LON)A
 AR99 3C 1168 INC A
 AR9A C22A1 1178 LD (LON)A
 AR9F C9 1188 RET
 1198 I
 AR95 3A39A1 1208 DIV1: LD A,(BITE)
 AR9F FE8A 1218 LD C,8
 AR9F FE8A 1228 DIV1: CP 16
 AR9F 3089 1238 INC A,SAL
 AR9F D0A8 1248 SUB 18
 AR9F D0A8 1258 INC C
 AR9F 3239A1 1268 LD (BITE),A
 AR9F 18F4 1278 JR DIV1
 AR9F C1 1288 RET
 AR97 79 1298 SAL 1
 AR9E FE80 1308 CP 8
 AR9A 28DC 1318 JR NZ,PINTA
 AR9E B2 1328 DEC HL
 AR9E 7E 1338 LD A,(HL)
 AR9E 23 1348 INC HL
 AR9F FE23 1358 CP 47
 AR9E 06 1368 RET C
 AR9E 12 1378 JR PINTA
 1408 I
 AR9A 3A39A1 1398 DOS1: LD A,(BITE)
 AR9F FECE 1408 CP 288
 AR9E 10 1418 RET C
 AR9F D6C2 1428 SUB 288
 AR9F 3239A1 1438 LD (BITE),A
 AR9F 048D 1448 LD C,2
 AR9F 18C5 1458 JR PINTA
 1468 I
 AR9A 3A39A1 1478 UNO: LD A,(BITE)
 AR9E FE64 1488 CP 160
 AR9F 3089 1498 INC A
 AR9E 0464 1508 SUB 180
 AR9E 3239A1 1518 LD (BITE),A
 AR9E 0E 1528 LD C,1

250 DATA 161,193,161,219,43,94,8
 260 DATA 225,50,58,161,198,6,58
 270 DATA 90,161,285,153,148,281,58
 280 DATA 50,161,119,35,54,8,35
 290 DATA 50,51,161,254,255,48,6
 300 DATA 68,58,68,161,24,12,62
 310 DATA 8,58,51,161,58,68,161
 320 DATA 68,58,68,161,58,51,161
 330 DATA 119,35,58,68,161,119,25,58
 340 DATA 54,148,35,54,32,201,221
 350 DATA 124,8,58,57,161,285,28
 360 DATA 245,161,285,93,161,285,245,168
 370 DATA 285,221,168,201,14,9,58
 380 DATA 57,161,185,48,3,13,24
 390 DATA 247,151,178,68,119,25,58
 400 DATA 50,161,48,58,58,161,201
 410 DATA 58,57,161,14,8,254,18
 420 DATA 9,4,9,216,12,2,58,57
 430 DATA 161,24,244,201,121,254,0
 440 DATA 32,328,43,126,35,254,47
 450 DATA 216,24,24,24,57,1,254
 460 DATA 208,216,214,208,158,57,161
 470 DATA 14,2,24,197,58,57,161
 480 DATA 254,168,216,214,188,58,57
 490 DATA 161,141,24,192,8,8



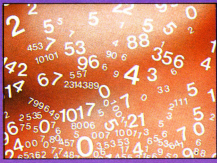
AR89 2B 618 DEC HL
 AR8A 3489 628 LD (HL),8
 AR8C E1 638 POP HL
 AR8D 3A32A1 648 LD A,(LON)A
 AR8E C88E 658 ADD A,6A
 AR8F 3236A1 668 LD (LON)A
 AR90 CDF9A8 678 CALL DATA
 AR91 658 RET
 AR99 3A32A1 698 DATA: LD A,(LON)A
 AR9C 77 708 INC HL
 AR9D 23 718 INC HL
 AR9E 3488 728 LD (HL),8
 AR9F 28C 738 INC HL
 AR9A 3630A1 748 LD A,(LINE)A
 AR9A FEFF 758 CP #FF
 AR9A 2586 768 JR B,LINE
 AR9A 3C 778 INC A
 AR9A 3239A1 788 LD (LINE),A
 AR9A 19E1 798 JR FN,LINE
 AR9A 8E90 808 ORN LINE: LD A,8
 AR9B 3239A1 818 LD (LINE),A
 AR9C 3A32A1 828 LD A,(LON)A
 AR9C 830 838 INC A
 AR9D 3236A1 848 LD (LINE),A
 AR9E 3A32A1 858 POP: LD (HL),A
 AR9D 77 868 LD (HL),8
 AR9E 3A32A1 878 LD A,(LINE)A
 AR9C 23 888 INC HL
 AR9C 348C 898 LD (HL),8BC
 AR9C 23 928 INC HL

AR9E 3A39A1 1398 DOS1: LD A,(BITE)
 AR9F FECE 1408 CP 288
 AR9E 10 1418 RET C
 AR9F D6C2 1428 SUB 288
 AR9F 3239A1 1438 LD (BITE),A
 AR9F 048D 1448 LD C,2
 AR9F 18C5 1458 JR PINTA
 1468 I
 AR9A 3A39A1 1478 UNO: LD A,(BITE)
 AR9E FE64 1488 CP 160
 AR9F 3089 1498 INC A
 AR9E 0464 1508 SUB 180
 AR9E 3239A1 1518 LD (BITE),A
 AR9E 0E 1528 LD C,1



LISTAD DESEMSABLADO

AR88
 AR88 D0688 18 LD \$80 \$888
 AR88 D06881 38 LD N,(X+1)
 AR88 2236A1 48 LD (ADER),BC
 AR89 D06882 58 LD B,(X+2)
 AR8C D06883 68 LD N,(X+3)
 AR8B 2236A1 78 LD (START),HL
 AR8D D06884 88 LD A,(X+4)
 AR95 3236A1 98 LD (LINE),A
 AR98 D06885 108 LD A,(LON)A
 AR98 3236A1 118 LD (LINE)A
 AR9E 2A6A8E 128 LD HL,(BAS1)
 AR91 2B 148 LD A,8
 AR92 2B 158 DEC HL
 AR92 D02A3A1 168 LD 1,(START)
 AR92 3E88 178 LD A,8
 AR92 3236A1 178 LD (FLA)A
 AR92 3236A1 188 LD (LON)A
 AR92 2236A1 198 LD (FORME),HL
 AR92 C0548 208 BUCLE: CALL INICIO
 AR95 2A3A1 218 LD HL,(POSHE1)
 AR98 3A32A1 228 LD A,(LON)A
 AR9E 3F 238 LD E,A
 AR9E 3E88 248 LD D,8
 AR9E 3E88 258 LD C,8
 AR9A 3236A1 268 LD (LON)A
 AR93 19 278 ADD HL,D
 AR94 3236A1 288 LD (FORME),HL
 AR9A 3A36A1 298 LD A,(FLA)A
 AR9A FE80 308 CP 8
 AR9C 38E4 318 LD (BUCLE),HL
 AR9E 23 328 INC HL
 AR9E 22A68 338 INC HL
 AR9E 22A68 348 LD (BAS2),HL
 AR95 22A68 358 LD (VARI),HL
 AR95 22A68 368 LD (VARI),HL
 AR9C 39 388 RET
 AR9C 85 398 JNC INICIO PUSH HL



AR9C 3428 938 LD (HL),"
 AR9C 39 948 RET
 958 I
 AR9A D0788 958 LD R8E: LD A,(X)
 AR9D C014A1 978 LD (BITE),A
 AR9D C014A1 988 CALL DOS
 AR9A C0F5A8 1008 CALL D10
 AR9E C0548 1018 CALL UNO
 AR9C 39 1028 RET
 1038 I
 AR9E 8E91 1048 UNIDAD: LD C,9
 AR9A 3A39A1 1058 BUC1: LD A,(BITE)
 AR9E B9 1068 CP C
 AR9E 2883 1078 JR 2,(PINTA)
 AR9D 8E90 1088 DEC D
 AR9E 18F7 1098 JR SUC
 1108 I

AR9E 18E4 1538 JR PINTA
 AR9E 1548 LONG: DEFS 1
 AR9E 1548 POSHE: DEFS 2
 AR9E 1578 ADOR: DEFS 2
 AR9E 1588 FLA0: DEFS 1
 AR9E 1598 BITE: DEFS 1
 AR9E 1608 START: DEFS 2
 AR9E 1618 UNO: DEFS 1
 AR9E 1628 BAS1: EDU #E6A
 AR9E 1638 BAS2: EDU #E6B
 AR9E 1648 VAR1: EDU #E6A
 AR9E 1658 VAR2: EDU #E6C

ETIQUETAS

AD0R A136 BAS1 AE6A BAS2 AE6B
 BUC A0CF BUCLE A032 BITE A139
 DATA A099 DIV1 A0F4 DIV1 A0F5
 DOS A014 FLA0 A138 INICIO A0E0
 LINE A0AE LINEA A133 UNO A13C
 LON0 A132 R8E A0CA OTR A064
 PINTA A06E POP A08A POPHE A134
 SAL A107 S18E A08A START A13A
 UNIDAD A08D UNO A123 VAR1 AE6A
 VAR2 AE6C



Para que sus datos no realicen el trabajo duro. M.H. AIMS TRAD lo hace por ti. Todos los trabajos que requieren este lenguaje se encuentran a tu disposición en un casette mensual selectivo.

¡SENSACIONAL!



Te presentamos un equipo sensacional: el **AMSTRAD CPC 6128**

Con un solo cable para enchufar a la red, el Sistema 6128 está listo para funcionar.

JUEGA Y APRENDE CON EL 6128

Para jugar, el 6128 es un ordenador muy serio; gracias a sus cientos de programas disponibles, tienes aseguradas horas de entretenimiento. Y en el mundo de la enseñanza no es menos.

Gracias a sus sensacionales capacidades gráficas (paleta de 27 colores y hasta 640 x 200 PIXELS) y sonoras (3 voces y 8 octavas, altavoz interior y salida estéreo) el 6128 es una herramienta inigualable. Además, dentro del paquete de programas que se entrega con el sistema, está incluido el lenguaje educativo por excelencia: el **Dr. LOGO** de Digital Research. Y para profundizar en el lenguaje de la informática recuerda que el 6128 es el ordenador idóneo, ya que posee uno de los más rápidos y potentes BASIC — el **LOCOMOTIVE BASIC** —, así como otros muchos lenguajes de programación: **FORTH, PASCAL**, etc.

TRABAJA CON EL 6128

Haz un sitio en tu negocio al 6128. Planifica presupuestos, lleva contabilidades, gestiona archivos, todo fácilmente gracias a su Sistema Operativo CP/M (en versiones 2.2 y Plus), que (como ya sabes) te permitirá acceder a la más extensa biblioteca de programas profesionales: bases de datos, procesadores de textos, hojas de cálculo, etc.

CARACTERÍSTICAS TÉCNICAS

- 128K RAM y 48K ROM (incluye Locomotive BASIC y Sistema Operativo).
- Monitor: Color de 14" y fósforo verde de 12".
- Unidad de Disco 3" incorporada (180K por cara).
- Teclado profesional.
- Sistema Operativo: AMS-DOS, CP/M 2.2 y CP/M Plus.
- Salida para segunda unidad de disco y cassette externa.

El CPC 6128 incluye en su suministro:

- Disco con Sistema Operativo CP/M 2.2 y lenguaje Dr. LOGO.
- Disco con Sistema Operativo CP/M Plus y utilidades.
- Disco con seis programas de obsequio.
- Manuales en castellano.
- **GARANTÍA AMSTRAD ESPAÑA ÚNICA VALIDA PARA ACCEDER AL SERVICIO TÉCNICO OFICIAL.**

TODO POR:

- **109.500 ptas.**
(monitor fósforo verde)
- **134.500 ptas.**
(monitor color)

¡¡Inmense!!

AMSTRAD

ESPAÑA

GARANTÍA INDESCOMP

Avd. del Mediterráneo, 9 - 28007 Madrid Tels. 433 45 48 - 433 48 76 - Telex 47660 FAX - 4332450

SPACE TRUCKER

Después de la última Gran Guerra, que tuvo lugar en 2095, en la cual el hemisferio Norte de la Tierra fue completamente volatilizado, Brasil tomó el control de todas las colonias en el espacio, formando el centro de Gobierno de toda la Federación en la colonia de Porto Quento.

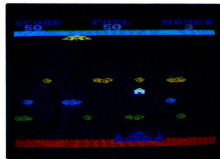
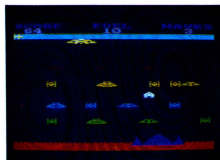


Tú eres un patrullero espacial recién llegado al planeta y como novato, no eres particularmente bueno en tu trabajo, por lo que la Federación te ha encomendado la patrulla de los sistemas estelares del borde externo de nuestra galaxia.

Usando un artefacto parecido a la actual lanzadera espacial, pero mucho más sofisticado, debes vigilar que el tráfico entre los planetas a tu cargo y las naves-madre no se vea interrumpido ni amenazado por alienígena o algo.

Sólo tienes 3 lanzaderas a tu disposición, en un lugar infestado de meteoritos y tráfico alienígena. Cualquier contacto con alguno de esos objetos te matará inmediatamente.

Atención: la fuerza de gravedad te empujará hacia el suelo del planeta, gastando combustible.



```

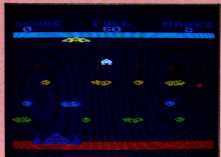
1 REM***** Space Trucker *****
2 REM
3 REM(c)Amstrad Semanal
4 REM
5 REM**** by Aramello Chapman ****
10 fr=FRE("):CLEAR:DIM ale$(16),na
$(8),hs(8)
20 FOR f=1 TO 8:hs(f)=(1000-(#f*100
+100)):NEXT :FOR f=1 TO 8 STEP 2:n
a$(f)="SPACE":na$(f+1)="Trucker":NE
XT
30 GOSUB 1220:REM U.D.G
40 GOSUB 1960:REM Machine Code
50 INK 14,24:INK 1,26:INK 0,0:INK 1
5,13:MODE 1:PEN 2:LOCATE 12,10:PRIN
T"Espera por favor!!"
60 GOSUB 930:REM Crea Trafico
70 GOSUB 1690:REM Instrucciones
80 GOSUB 1560:REM Variables
90 base:=INT(RND*15)+1
100 GOSUB 770:REM Dibuja Pantalla
110 LOCATE 1,2:PEN 13:PRINT score
120 REM*****
130 REM***** Buclé Principal ****
140 IF dir=1 AND shipx=base*2 AND
shipy=21 THEN GOTO 630
150 IF dir=-1 AND shipx=mothx AND s
hipy=5 THEN GOTO 690
160 IF INKEY(44)=0 THEN GOTO 180
170 GOTO 200
180 IF INKEY(60)<>0 THEN GOTO 180
190 GOTO 120
200 REM*****
210 REM***** Mueva nave *****
220 PEN 1:FOR f=1 TO 100-speed:NEXT
f
230 IF INKEY(47)=0 AND fuel>0 THEN
GOSUB 350:th=1:GOTO 260
240 newx=newx+(INKEY(71)=0 AND ship
x<2)-1:INKEY(63)=0 AND shipx<19:ne
w:=newx+dir AND shipy<4)
250 SOUND 1,20:newy,5,10,3,1
260 IF dir=1 AND (newy=10 OR newy=1
7 OR newy=13 OR newy=20) OR dir=-1
AND (newy=8 OR newy=15 OR newy=11 O
R newy=18) THEN 330
270 IF th=1 AND (shipy=9 OR shipy=1
2 OR shipy=16 OR shipy=19) OR shipy
=4 THEN GOTO 300
280 GOSUB 410:LOCATE shipx,shipy:PR
INT" :LOCATE newx,newy:CALL char:IF
RINT CHR$(232):shipx=newx:shipy=ne
w:y:cr=PEEK(&A0B):IF cr<>32 THEN GOT
0 500
290 GOTO 190
300 IF cr<>32 THEN GOTO 500
310 LOCATE shipx,shipy:PRINT" :GOS
UB 410:LOCATE newx,newy:CALL char:
r=PEEK(&A0B):IF cr<>32 THEN 500 EL
SE shipx=newx:shipy=newy:LOCATE shi
px,shipy:PRINT CHR$(232):
320 th=0:GOTO 190
330 LOCATE shipx,shipy:PRINT" !:shi
px=newx:shipy=newy:LOCATE shipx,shi
py:GOSUB 410:PRINT CHR$(232)
340 GOTO 190
350 FOR f=1 TO 10:SOUND 1,0,5,12,0,
0,5:NEXT
360 fuel=fuel-1
370 LOCATE 9,2:PEN 13:PRINT fuel:PE
N 1
380 RETURN
390 REM*****
400 REM***** Mueva Trafico *****
410 IF shipy<>9 OR shipy<>12 OR shi
py<>16 OR shipy<>19 OR shipy<>24 THE
N GOTO 430
420 LOCATE newx,newy:PRINT" !:GOSUB
440:LOCATE newx,newy:CALL char:cr=
PEEK(&A0B):IF cr<>32 THEN GOTO 500
ELSE LOCATE newx,newy:PRINT CHR$(2
32):GOTO 450
430 GOSUB 440:GOTO 450
440 FOR f=1 TO 4:CALL left,8:CALL r
ight,11:CALL left,15:CALL right,18:
CALL left,3:NEXT:RETURN
450 mothx=mothx-1
460 IF mothx<1 THEN mothx=20

```

RUTINAS

120	Bucle principal.
200	Mueve lanzadera y chequea para ver si ha habido colisiones.
390	Mueve alienígenas, rocas y la Nave Madre.
480	Pinta las explosiones y disminuye las vidas.
630	Incrementa el score y cambia la dirección de vuelo.
690	Idem línea 630.
770	Imprime las pantallas.
930	Inicializa las matrices con los modelos de caracteres pertinentes.
1220	Inicializa caracteres.
1560	Inicializa variables.
1690	Instrucciones.
1960	Pokea el código máquina en memoria.
2130	Pinta el high-score.

Serie Oro



```

470 RETURN
480 REM*****
490 REM***** Rutina de choque **
500 LET shipx=newx:LET shipy=newy:P
EN 3:LOCATE shipx,shipy:PRINT CHR$(
250)
510 SOUND 1,0,0,15,3,2,1:SOUND 2,0
,0,15,3,1,2
520 FOR f=#1 TO 100:NEXT
530 PEN 3:LOCATE shipx-1,shipy:PRIN
T CHR$(252):CHR$(250):CHR$(253)
540 SOUND 1,0,0,15,3,3,2:SOUND 2,0
,0,15,1,2,4
550 FOR f=#1 TO 200:NEXT
560 PEN 3:LOCATE shipx-1,shipy:PRIN
T " CHR$(251)"
570 SOUND 1,0,0,0,15,3,4,1
580 FOR f=#1 TO 2000:NEXT
590 lives=lives-1
600 IF lives=0 THEN FOR f=#1 TO 14:
LOCATE 6,f:PRINT STRING$(9," ");NEX
T:PEN 3:LOCATE 7,11:PRINT"5 A M E";
PEN 12:LOCATE 7,13:PRINT"0 V E R";
DR f=#1 TO 3000:NEXT:GOTO 2130
610 shipx=10:shipy=5:dir:=1:moth:=10
:scr:=2:new:=10:newy=5
620 GOTO 90
630 REM*****
640 REM***** Rutina de aterrizaje
****
650 score=score+fuel:SOUND 1,50,0,1
5,3,1,3:SOUND 1,300,0,15,3,1,4
660 DIR=#1
670 FOR f=#1 TO 1000:NEXT
680 GOTO 110
690 REM*****
700 REM***** Astropuerto *****
****
710 shipx=10:shipy=5:level=level+1:
newx=10:newy=5
720 LET dir:=1:moth=10
730 FOR f=#1 TO fuel
740 score=score+1:LOCATE 1,2:PEN 13
:PRINT score:SOUND 1,(fuel+10)-f,1,
15:NEXT f
750 IF level=5 THEN fuel=fuel+20:sp
ed=sped+30:IF sped>70 THEN sped=70
760 GOTO 90
770 REM*****
780 REM***** Dibuja Pantalla *****
790 MODE 0:PAPER 0:CLS:PEN 1:BORDER
0

```

```

800 LOCATE 1,23:PEN 9:PRINT STRING$
(20,CHR$(234))
810 LOCATE 1,24:PRINT STRING$(20,CH
R$(143))
820 LOCATE 1,1:PEN 12:PRINT"SCORE
FUEL NAVES "
830 LOCATE 1,3:PAPER 15:PEN 6:PRINT
STRING$(20,CHR$(229))
840 LOCATE basex,22:PAPER 0:PEN 3:P
RINT CHR$(235):PEN 12:PRINT CHR$(2
44):CHR$(245):PRINT CHR$(246):PEN
3:PRINT CHR$(235):LOCATE basex,23
:PRINT CHR$(236):PEN 12:PRINT CHR$(
247):CHR$(248):CHR$(249):PEN 3:PR
INT CHR$(236)
850 IF level=1 THEN lev=1 ELSE IF 1
evel=2 THEN lev=5 ELSE IF level=3 T
HEN lev=9 ELSE IF level=4 THEN lev=
13 ELSE IF level=5 THEN level=1:GOT
O 850
860 LOCATE 1,9:PEN 11:PRINT ale$(1e
v+1)
870 LOCATE 1,12:PEN 8:PRINT ale$(1e
v+1)
880 LOCATE 1,16:PEN 7:PRINT ale$(1e
v+2)
890 LOCATE 1,19:PEN 10:PRINT ale$(
ev+3)
900 LOCATE 9,4:PEN 14:PRINT CHR$(22
5):CHR$(226):CHR$(227)
910 PEN 13:LOCATE 9,2:PRINT fuel:LO
CATE 17,2:PRINT lives
920 RETURN
930 REM*****
940 REM***** Crea Trafico *****
950 RESTORE 1050
960 FOR f=#1 TO 16
970 ale$(f)="
980 FOR g=#1 TO 20
990 READ n
1000 IF n=0 THEN n=32
1010 ale$(f)=ale$(f)+CHR$(n)
1020 fr=FR(" ")
1030 NEXT g,f
1040 RETURN
1050 DATA 242,243,0,0,0,241,0,0,0,0
,0,239,240,0,0,241,0,0,241,0
1060 DATA 241,0,0,0,0,237,238,0,0,2
41,0,0,242,243,0,0,0,0,0
1070 DATA 0,0,0,0,241,0,0,0,242,243
,0,0,0,0,0,0,0,241,0
1080 DATA 0,0,0,0,241,0,0,0,237,238
,0,0,0,241,0,0,0,242,243,0
1090 DATA 0,0,0,0,0,0,225,227,0,0,0
,228,0,0,0,225,227,0,0,0
1100 DATA 0,228,0,0,0,0,230,231,0,0
,0,228,0,228,225,227,0,0,0,0
1110 DATA 0,0,0,0,0,230,231,0,0,2
28,0,0,225,227,0,0,0,228,0
1120 DATA 0,228,0,0,0,230,231,0,0,0
,225,227,0,0,0,0,228,0,0,0
1130 DATA 0,242,243,0,0,0,241,0,0,0
,240,0,0,242,243,0,0,0,0
1140 DATA 0,0,0,0,237,238,0,241,0,0
,239,240,0,0,0,242,243,0,241,0
1150 DATA 0,241,0,241,0,0,242,243,0
,0,0,239,240,0,0,241,0,0,237,238
1160 DATA 0,0,0,0,241,242,243,0,239
,240,0,0,237,238,241,0,0,0,241
1170 DATA 225,227,0,0,228,0,0,230,2
31,0,0,228,0,0,230,231,0,0,0,228
1180 DATA 0,0,0,228,0,0,230,231,0,0
,225,227,0,0,230,231,0,0,228,0
1190 DATA 228,0,0,0,225,227,0,0,230
,231,0,0,0,228,0,0,0,225,227,0
1200 DATA 225,227,0,0,228,0,0,230
,231,0,0,0,225,227,0,0,0,228,0
1210 GOTO 1210
1220 REM*****
1230 REM***** U.D.Gs *****
1240 SYMBOL AFTER 225
1250 SYMBOL 225,0,7,15,57,121,255,3
0
1260 SYMBOL 226,60,255,195,255,255,
195,153,0
1270 SYMBOL 227,0,224,240,156,158,2
55,192,0
1280 SYMBOL 228,195,153,189,231,255

```

```

,189,153,195
1290 SYMBOL 229,34,119,34,0,0,34,11
9,34
1300 SYMBOL 230,7,6,31,62,119,248,2
55,15
1310 SYMBOL 231,224,96,248,124,238,
31,255,240
1320 SYMBOL 232,24,60,102,255,219,2
55,66,231
1330 SYMBOL 233,24,24,60,68,170,36,
146,168
1340 SYMBOL 234,0,0,0,34,119,255,25
5,255
1350 SYMBOL 235,0,0,16,16,16,16,24,
24
1360 SYMBOL 236,24,126,44,126,126,2
54,246,246
1370 SYMBOL 237,28,239,255,126,106,
63,6
1380 SYMBOL 238,24,126,231,183,254,
184,248,96
1390 SYMBOL 239,28,98,239,255,126,1
08,63,6
1400 SYMBOL 240,24,126,231,183,254,
184,248,96
1410 SYMBOL 241,60,126,219,175,110,
122,28,0
1420 SYMBOL 242,8,55,93,247,222,117
,63,29
1430 SYMBOL 243,216,116,202,119,191
,254,124,200
1440 SYMBOL 244,0,0,0,0,1,2,7,10
1450 SYMBOL 245,60,126,126,255,255,
0,255,170
1460 SYMBOL 246,0,0,0,0,128,64,224,
176
1470 SYMBOL 247,15,31,52,52,255,63,
255,255
1480 SYMBOL 248,255,255,146,146,255
,255,223,223
1490 SYMBOL 249,240,248,76,76,255,2
52,255,253
1500 SYMBOL 250,145,102,62,164,18,7
3,102,153
1510 SYMBOL 251,0,72,0,4,64,4,0,66
1520 SYMBOL 252,7,15,126,248,248,12
6,15,7
1530 SYMBOL 253,224,240,126,31,31,1
26,240,224
1540 SYMBOL 254,198,165,198,165,6,4
0,40,16
1550 RETURN
1560 REM*****
1570 REM***** Variables *****
1580 score=0
1590 level=1:sped=10
1600 moth=10
1610 shipx=10:newx=10
1620 shipy=5:newy=5
1630 fuel=50

```



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

TECLAS DE MANEJO DEL PROGRAMA

Z Izquierda.
X Derecha.
SPACE Averigüalo tú mismo!
H Freno.
R Avance

```

1640 lives=3
1650 dir=1:cr=32
1660 ENV 1,15,-2,10:ENV 2,15,-1,2:ENV
NV 3,15,-1,10:ENV 4,15,1,2:ENT 1,15
,5,2
1670 fr=FRE(")
1680 RETURN
1690 REM*****
1700 REM***** Instrucciones *****
1710 MODE 1:BORDER 0:GOSUB 1890
1720 LOCATE 1,7:PEN 2:PRINT " Estas
en el 2160, y como piloto de la
Space Trucker Company, tu trabajo
es transportar mercancías desde l
a nave nodriza en orbita hasta e
l Astropuerto en el planeta."
1730 PRINT " La S.T.C ha dicho,(gen
erosamente), que los pilotos seran
pagados por la cantidad de fuel q
ue consigas ahorrar en cada viaje (
Esto es para acabar con algunos te
rremotos derrochones)."
1740 PRINT " Como pronto veras,el p
laneta al que has sido enviado no e
s el mas indicado para disfrutar pi
lotando!"
1750 WHILE INKEY(47)<>:GOTO 1760
1760 LOCATE 1,1:PRINT STRING$(26,11
)
1770 GOSUB 1890
1780 LOCATE 1,6:PEN 2:PRINT " Hay r
iesgo de fallar en la mision.Vigila
a las naves alienigenas y los gran
des meteoritos.En la busqueda de
mayores beneficios la S.T.C no ha
dotado la nav e de laser,asi que d
eberas evitarlos."
1790 PRINT " Tu nave tiene cohetes
para disminuir el efecto de la gra
vedad
1800 PRINT " Aterriza en la parte
alta de la pista del Astropuerto y
solo en el medio."
1810 WHILE INKEY(47)<>:GOTO 1820
1820 LOCATE 1,1:PRINT STRING$(26,11
)
1830 GOSUB 1890
1840 LOCATE 17,7:PEN 3:PRINT"LAS TE
CLAS":LOCATE 17,8:PEN 1:PRINT STRIN
G$(10,CHR$(131))
1850 LOCATE 14,9:PEN 2:PRINT""Z"-IZ
QUIERDA":LOCATE 14,11:PRINT""X"-DER
ECHA":LOCATE 12,13:PRINT"ESPACIO"-
COHETES":LOCATE 14,15:PRINT"H"-PAR
A":LOCATE 14,17:PRINT""S"-HUEVE"
1860 WHILE INKEY(47)<>:GOTO 1870
1870 LOCATE 1,1:PRINT STRING$(26,11
)
1880 GOTO 2130
1890 LOCATE 1,1:PAPER 1:PEN 3:PRINT
STRING$(40,CHR$(229))
1900 FOR f=2 TO 4:LOCATE 1,f:PRINT CH
R$(229);CHR$(229):LOCATE 39,f:PRIN
T CHR$(229);CHR$(229):NEXT f
1910 LOCATE 1,5:PRINT STRING$(40,CH
R$(229))
1920 LOCATE 15,3:PEN 2:PAPER 0:PRIN
T"Space Trucker"
1930 LOCATE 1,20:PAPER 1:PEN 3:PRIN
T STRING$(40,CHR$(229)):FOR f=21 TO

```

```

(23:LOCATE 1,f:PRINT CHR$(229);CHR$(
229):LOCATE 39,f:PRINT CHR$(229);C
HR$(229):NEXT:LOCATE 1,24:PRINT STR
ING$(40,CHR$(229))
1940 LOCATE 1,22:PAPER 0:PRINT"Puls
a <ESPACIO> para seguir."
1950 RETURN
1960 REM*****
1970 REM*****Codigo Maquina****
1980 h1g=#FFF:address=h1g:1:high:h
igh:1:RESTORE 2050:left=high:right=h
igh:50:char=high+107
1990 READ n
2000 IF n=999 THEN READ n:IF n<>:C
THEN PRINT CHR$(7):PRINT STRING$(6,
CHR$(7)):GOTO 11,10:PEN 3:PRIN
T"ERROR EN CODIGO MAQUINA":END ELS
E IF n=16162 THEN sc=0:RETURN
2010 sc=sc+n
2020 POKE address,n
2030 address=address+1
2040 GOTO 1990
2050 REM***** SCROLL IZQU.***
****
2060 DATA $dd,$ee,$00,$26,$00,$cd,$
1a,$bc,$e5,$7e,$f5,$54,$5d,$eb,$23,
$cb,$f4,$cb,$fc,$0e,$f4,$7e,$12,$23
,$cb,$f4,$cb,$fc,$eb,$23,$cb,$f4,$c
b,$fc,$eb,$00
2070 DATA $20,$ef,$f1,$12,$ef,$7c,$
c6,$08,$67,$e6,$78,$20,$d7,$c9
2080 REM***** SCROLL DER.*****
**
2090 DATA $dd,$ee,$00,$26,$00,$cd,$
1a,$bc,$01,$f4,$00,$09,$cb,$f4,$cb,
$fc,$e5,$7e,$f5,$54,$5d,$2b,$cb,$f4
,$cb,$fc,$0e,$f4,$7e,$12,$2b,$cb,$f
4,$cb,$fc,$eb,$2b,$cb,$f4
2100 DATA $cb,$fc,$eb,$0d,$20,$ef,$
f1,$12,$e1,$7c,$c6,$08,$67,$e6,$78,
$20,$d8,$c9
2110 LOCATE $cd,$e6,$bb,$32,$8d,$A0,&
c
2120 DATA 999,16162
2130 REM***** Max. Punt.*****
2140 REM*****
**
2150 MODE 1
2160 LOCATE 4,5:PEN 3:PRINT CHR$(15
2160):STRING$(32,CHR$(154)):CHR$(156)
2170 LOCATE 16,3:PEN 2:PRINT"HIGH S
CORE":PEN 3
2180 FOR f=6 TO 15:LOCATE 4,f:PRINT
CHR$(149):LOCATE 37,f:PRINT CHR$(1
49):NEXT f
2190 LOCATE 4,16:PRINT CHR$(147):ST
RING$(32,CHR$(154)):CHR$(153)
2200 FOR f=1 TO 8
2210 IF score>hs(f) THEN GOSUB 2310
:f=10
2220 NEXT f
2230 FOR f=1 TO 8:PEN 1:LOCATE 8,f+
6:PRINT na$(f):LOCATE 18,f+6:PEN 3:
PRINT".....":hs(f):NEXT f
2240 LOCATE 1,17:PRINT STRING$(220,
" ")
2250 IF INKEY#<> THEN GOTO 2250
2260 PEN 2:LOCATE 1,20:PRINT STRIN
G$(40,CHR$(154)):LOCATE 1,22:PRINT S
TRING$(40,CHR$(154)):PEN 1
2270 LOCATE 1,23:PRINT STRING$(40,"
")
2280 LOCATE 9,21:PEN 3:PRINT"PULSA"
:1:PEN 1:PRINT"ESPACIO":1:PEN 3:PRIN
T"PARA SEGUIR."
2290 IF INKEY(47)<> THEN GOTO 2290
2300 LOCATE 1,1:PRINT STRING$(26,11
):GOTO 80
2310 a$="ABCDEFGHIJKLMNPOQRSTUVWXYZ
.2:$(1):(1)7*"+CHR$(254)
2320 c=19:LOCATE 1,20:PEN 1:PRINT a
$
2330 LOCATE 6,17:PEN 1:PRINT"USA TE
CLAS DEL CURSOR Y COPY":LOCATE 3,18

```

```

:PRINT"PARA SELECCIONAR LETRAS.(MAX
. DE 10.):PEN 2:LOCATE 1,19:PRINT
STRING$(40,CHR$(154)):LOCATE 1,22:P
RINT STRING$(40,CHR$(154))
2340 LOCATE 11,23:PEN 3:PRINT"PULSA
":PEN 1:PRINT"X" ):1:PEN 3:PRIN
T"PARA SALIR.":PEN 1
2350 x$=""
2360 FOR z=1 TO 10
2370 LOCATE c,21:PEN 2:PRINT" "
2380 IF INKEY(1)=0 AND c<40 THEN c=
c+1
2390 IF INKEY(8)=0 AND c1 THEN c=c
-1
2400 IF INKEY(9)=0 AND c=40 THEN LO
CATE 7,f+6:PRINT" " "z=10
16:GOTO 2450
2410 IF INKEY(63)=0 THEN z=11:GOTO
2450
2420 IF INKEY(9)<0 TO 50:LOCATE c,2
1:PRINT"":FOR a=1 TO 50:NEXT:GOTO
2370
2430 x$=x$+MID$(a$,c,1):LOCATE 7+z,
f+6:PEN 1:PRINT MID$(a$,c,1)
2440 FOR a=1 TO 200:NEXT
2450 NEXT
2460 IF z=101 THEN GOTO 2350
2470 hs(8)=score:na$(8)=x$
2480 f=0
2490 FOR z=1 TO 7
2500 IF hs(z)<hs(z+1) THEN t=hs(z+1
):hs(z+1)=hs(z):hs(z)=t:na$(z)=na$(z+1
):na$(z+1)=na$(z):na$(z)=t:na$(z)=f+1
2510 NEXT
2520 IF f=1 THEN GOTO 2480
2530 fr=FRE(")
2540 RETURN

```

VARIABLES

ale\$	Matriz que almacena el tráfico de las rocas.
na\$	Nombres de la tabla de high-score.
hs	Puntuaciones.
basex	Coordenadas del astropuerto.
score	Score.
level	Pantalla actualmente en pantalla.
sped	Velocidad del tráfico alienigena.
mothx	Coordenada X de la Nave Madre.
shpx	Coordenada X de la lanzadera.
shpy	Coordenada Y.
fuel	Combustible que te queda.
lives	Número de lanzaderas que quedan con vida.
dir	Dirección de movimiento de la lanzadera.
newx, newy	Nuevas coordenadas de la lanzadera.
left	Dirección del scroll por la izquierda.
right	Por la derecha.
char	Dirección de la rutina de lectura de caracteres en pantalla.
th	Flag.

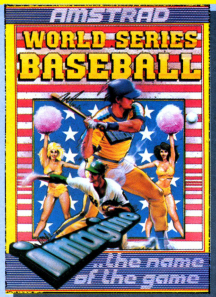
¡NUEVO!

SIEMPRE LOS PRIMEROS EN TENER LO ULTIMO

circulo de soft

MICROAMIGO S.A.

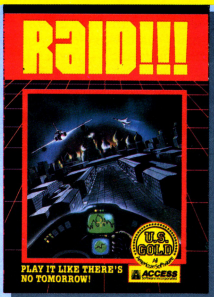
P.º de la Castellana, 268, 3.º C. 28046-MADRID.
Tel.: (91) 733 25 00



BASEBALL

Impresionante simulación en 3' dimensiones. Se puede competir contra el ordenador o contra otro jugador. No es necesario conocer el beisbol. Hay un modo de demostración. Pantallas gigantes para ver de cerca la acción.

P.V.P.: 2.100 ptas.
Precio C. de Soft: 1.890 ptas.



RAID

Defiéndete con tu escuadrilla de aviones del ataque nuclear que han lanzado sobre ti. Tu viaje estará lleno de peligros hasta que llegues a las bases de lanzamiento de misiles enemigas. Tendrás que destruirlos para salvar a tu país de una catástrofe nuclear. Gráficos y acción sensacionales.

P.V.P.: 2.300 ptas.
Precio C. de Soft: 2.070 ptas.



MATCH DAY

¡Ahora para Amstrad! No se trata de un juego de fútbol cualquiera. Fantástica acción en 3 dimensiones y animación total que dan vida al fútbol. Quedarás maravillado con el control del balón y desarrollarás tu destreza y técnica jugando contra otro jugador o contra el ordenador.

P.V.P.: 2.300 ptas.
Precio C. de Soft: 2.070 ptas.

¡¡¡...Y LOS TRES PROGRAMAS POR SOLO 5.400 PTAS!!!

¡HAZTE HOY MISMO SOCIO DEL CIRCULO DE SOFT! Además de poder adquirir tus programas al mejor precio, recibirás información de forma periódica y gratuita, del mejor software que aparezca en el mercado.

¿QUE HAY QUE HACER PARA SER SOCIO DEL CIRCULO DE SOFT? Así de fácil: envíanos por correo tu nombre, dirección y modelo de ordenador, o bien, pide por teléfono o por correo tu primer programa. ¡Y entrarás a formar parte del CIRCULO DE SOFT de forma inmediata!

Si quiero ser SOCIO desde hoy mismo del CIRCULO DE SOFT y recibir periódicamente información de novedades de software, así como beneficiarme desde hoy mismo de los precios reducidos reservados a los SOCIOS y de sus Ofertas Especiales. El ser SOCIO no me obliga a compra alguna.

Si prefieres formalizar tu compra por teléfono puedes hacerlo llamando al (91) 733 25 00. ¡NO SE COBRAN LOS GASTOS DE ENVIO POR CORREO!!

TITULO _____ P.V.P. _____ ORDENADOR _____

Contrarreembolso Giro Postal Talón adjunto a Microamigo, S.A. Tarjeta VISA n.º _____ Fecha caducidad _____

Nombre _____ Apellidos _____ Edad _____

Domicilio _____ Teléfono _____

Localidad _____ C.P. _____ Provincia _____

DESENSAMBLADOR - MONITOR DE HISOFT

La semana pasada tuvimos ocasión de analizar detalladamente qué es un ensamblador y para qué se utiliza, empleando como modelo y ejemplo el excelente programa de la casa inglesa de software Hisoft.



unque la necesidad de un programa de este tipo es incuestionable para cualquiera que desee programar en lenguaje máquina seriamente, dijéramos que un ensamblador sólo realiza para nosotros la mitad del trabajo: los verdaderos problemas suelen comenzar cuando se acaban de diseñar los programas y hay que probarlos.

El código máquina desconoce los mensajes de error; si nos equivocamos, el programa se cuelga o reacciona de forma impredecible y no hay más que hablar.

Fallos de diseño

Algunos errores pueden ser detectados por el propio ensamblador, como los fallos de sintaxis y otros, pero los fallos de lógica o los **despistes** sintácticamente correctos por ahora al menos quedan fuera del alcance de las máquinas.

Si nuestro programa depende críticamente de que el carry flag esté o no activado por una serie de sucesos a lo largo y ancho de una rutina crucial, por ejemplo, el ensamblador no puede averiguarlo.

Aquí es donde entra en escena el segundo programa del paquete de desarrollo de Hisoft: el MONA3.

MONA3

MONA3 es un potentísimo programa del tipo que los anglosajones llaman DDT (*Dinamic Debugging Tool*, Herramienta de Depuración Dinámica), y nos permite hacer con el código objeto, para chequear su buen funcionamiento, un montón de cosas.

En primer lugar, podemos desensamblarlo, es decir, hacer que aparezcan en pantalla los mnemónicos de lenguaje ensamblador de un programa en binario; por eso decimos que MONA3 es un desensamblador, porque realiza, entre otras cosas, la labor puesta a un ensamblador, al GENA3.

De acuerdo que esto no es muy espectacular de cara a depurar un programa que nosotros mismos hayamos creado usando GENA3, pero sí es importante para examinar rutinas hechas por otras personas, para ver cómo funcionan y aprender un poco.

GENA3 y MONA3, todo con todo

Además, Hisoft ha previsto un alto nivel de integración entre los dos programas, y es perfectamente posible tomar un fichero de código objeto, desensamblarlo a código fuente y **pasárselo** al ensamblador como si nosotros mismos lo hubiéramos escrito, con la enorme ventaja de poder gravarlo en cinta/disco, estudiarlo, modificarlo a nuestro gusto y, sobre todo, probarlo en las condiciones que queremos, por críticas y extremas que sean.

Este proceso, complejo a primera vista, resulta ser un juego de niños siguiendo las indicaciones del manual (*muy bueno por cierto, claro y lleno de ejemplos prácticos, para variar*), y sale a la primera sin ningún problema, prestando un mínimo imprescindible de atención, claro. Pruébese y se comprobará. Pero MONA3 nos permite hacer, como comenté antes, cosas más interesantes con el código objeto que queremos depurar: un poco más adelante vamos a verlo prácticamente comando por comando, pero podemos adelantarnos, un poco por encima, para tener una visión de conjunto del programa.

Panorámica de las posibilidades de MONA3

Por ejemplo, podemos ejecutar código máquina desde cualquier dirección, con el coman-

do «J»; esto es de gran utilidad, sobre todo si estructuramos el programa adecuadamente, en pequeños trozos o subrutinas cada uno de los cuales cumpla una tarea específica y que no interfiera con los demás.

Ahora bien, puede interesar, en un momento dado, precisamente lo contrario, es decir, saltarse un trozo de código máquina y continuar la ejecución en otra parte del programa, inmediatamente a continuación o en otra dirección diferente. La ventaja de esta posibilidad estriba en estudiar qué sucede con los flags y registros del Z80 si el programa bifurca a un sitio distinto del inicialmente previsto y hasta qué punto una subrutina independiente puede **encajar** en varios segmentos del programa original, o sustituirlos.

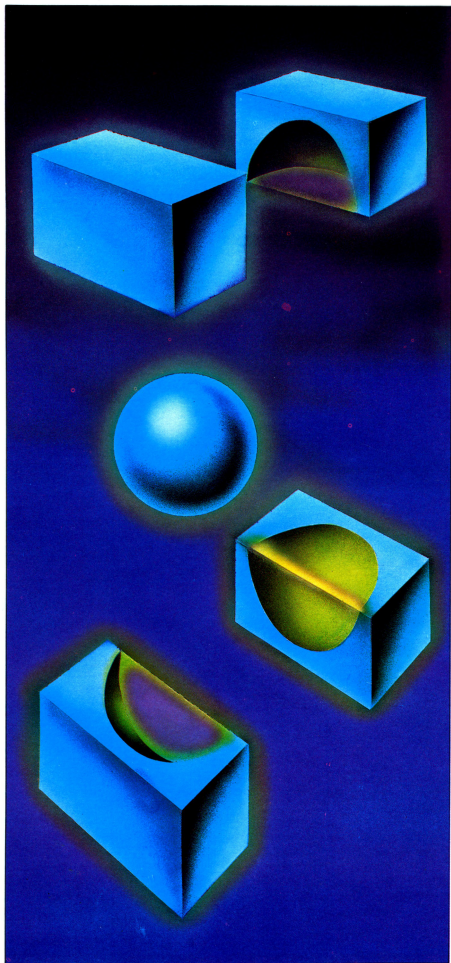
Aún hay más posibilidades. Imaginemos que, haciendo uso de lo anterior, hemos aislado una subrutina que falla, pero no sabemos dónde. Podemos obligar al MONA3 a que transfiera la ejecución a ese segmento del programa y, además, forzarle a que comience el proceso llamado **ejecución paso a paso**.

El nombrecito es autoexplicativo: MONA3 **agarr**a una instrucción de código máquina, la ejecuta y nos muestra en la pantalla el estado de los registros y los flags antes y después de hacerlo. De esta forma, es sencillísimo aislar la instrucción o instrucciones conflictivas y remediarlo de inmediato, porque también existen comandos para alterar el contenido de la memoria, byte a byte o por grupos. En fin, la ejecución paso a paso es particularmente útil para estudiar secuencialmente un trozo de código que no contenga saltos condicionales o llamadas a otras rutinas, aunque puede emplearse también en este caso teniendo un poco más de cuidado.

MONA3 permite también detener la ejecución de un programa en cualquier posición de la memoria contenida por él, tanto si nos encontramos en el modo paso a paso como en el de **ejecución ininterrumpida**.

El comando es «B» y coloca en la posición a donde señala el puntero de memoria de MONA3 (*que podemos mover, por supuesto*) lo que se conoce como el «**breakpoint**», un punto de ruptura.

Al ejecutar este comando, MONA3 coloca una instrucción especial en ese punto de la memoria y el código máquina real que existía allí



GANA
100.000 PESETAS
CON MICROHOBBY

AMSTRAD

SEMANAL

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

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.
- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:
 - *Descripción general del programa.*
 - *Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.*
 - *Instrucciones de manejo.*
- Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.
- En una sola cinta puede introducirse más de un programa.
- Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000** pesetas, en concepto de derechos de autor.
- Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.
- **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.
- Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.
- Los programas sospechosos de plagio serán eliminados inmediatamente.

**¡ENVIANOS TU
PROGRAMA!**

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

hasta entonces es salvaguardado en un buffer. Cuando la ejecución del programa en estudio alcanza el breakpoint, se muestra en pantalla el estado de registros y flags, el programa se detiene y se restituye el código objeto original.

MONA3 muestra en la pantalla, en casi cualquier momento, un display que nos permite inspeccionar registros y flags del microprocesador, junto con una zona de memoria de los alrededores de puntero interno de MONA3 con sus contenidos hexadecimales. Este display lo aprovechamos para cambiar directamente el valor de los registros del Z80 directamente, en respuesta a un número hexa seguido de un punto. El registro apuntado por la flecha tendrá a partir de ahora el valor indicado. Únicamente nos está prohibido alterar los registros SP (Stack Pointer, puntero del stack) e IR (Interrupt Register o vector de interrupción), porque si lo hacemos el resultado sería impredecible, es decir, cuél-

Comando a comando

Es el momento de analizar todas las facilidades ofrecidas por MONA3 detalladamente.

Comandos existentes

Los siguientes comandos son los que incluye MONA3, cuando utilizamos [ENTER] para terminar un número hexadecimal, dicho [ENTER] puede ser sustituido por otro carácter no hexadecimal «» significa espacio.

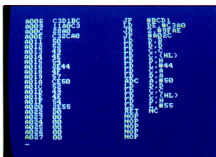
[CTRL]X Vuelve a Basic o al programa desde el que MONA3 fue ejecutado.

[CTRL]D Cambia la base en la que las direcciones de memoria estaban siendo impresas de 10 a 16 o al revés. Al principio, todas las direcciones se escriben en hexadecimal. Pulsando una vez el comando anterior, las direcciones se imprimirán en decimal y pulsando de nuevo, volvemos a la base 16. Este comando no afecta a los contenidos de los me-

SE BUSCA «ESPIA»

(de 10 a 15 años)

capaz de interferir red de ordenadores.
Ref.—HACKER



morias —los números de 8 bits se expresan siempre en hexadecimal, y los números introducidos desde el teclado deben encontrarse en dicha base.

[CTRL]A Imprime una página de código máquina. Empezando por la dirección del puntero. Es útil para ver qué instrucciones vienen detrás. Pulsando [CTRL]A de nuevo, volvemos al panel o pulsando cualquier otra tecla, obtendremos otra página de código máquina desensamblado.

«→» cursor derecha

incrementa el puntero en una unidad

«←» cursor izquierda

resta uno al valor del puntero

«↑» cursor arriba

resta ocho al valor del puntero. Es útil para retroceder rápidamente

«↓» cursor abajo

suma ocho al valor del puntero «avanzar»

«G» Recorre la memoria buscando una cadena determinada.

Aparecerá un símbolo «» a lo que deberás responder con el byte de la cadena que quieres buscar seguido de [ENTER], sigue introduciendo bytes (y [ENTER]) hasta que hayas definido toda la cadena. Pulsa [ENTER] otra vez y la memoria empezará a ser rastreada. Cuando el ordenador encuentra la cadena, el panel será «actualizado»; el puntero se colocará en el primer carácter de la cadena.

«H» Convierte un número decimal a su equivalente hexadecimal.

Para introducir el número, basta con escribir el número seguido de un carácter no decimal y obtendrás el número traducido a hexadecimal. Pulsa cualquier tecla para volver a MONA3.

«B» Copia inteligente. Se utiliza para copiar un bloque de memoria de una posición a otra. Es inteligente puesto que el bloque puede copiarse a posiciones que solapan la zona anterior. Es decir, podemos trasladar un bloque de, digamos 500 bytes desde su posición inicial a, simplemente, una posición más adelante (por ejemplo, de 1000 a 1001). «B» te pedirá la dirección de comienzo y final del bloque a copiar, y la posición en donde quieres copiarlo. Todos los números deben ser hexadecimales. Si la dirección de comienzo es mayor que la del final, el comando no tendrá efecto alguno.

«J» El número que se te pedirá después de los dos puntos «» es la dirección desde la que debe ejecutarse el código objeto en hexadecimal. Una vez hecho esto, el stack interno se reinicializa, se borra la pantalla y comienza la ejecución. Si quieres volver al panel en un punto determinado de la ejecución, deberás colocar un «breakpoint» en dicha posición (ver el comando «B»).

Pulsando [ESC] antes de terminar la dirección, el comando no será ejecutado.

[CTRL]C Continúa la ejecución desde la dirección actual del contador de programa (PC). Este comando se usa casi siempre después de «».

«B» Tabla o lista un bloque de memoria desde la dirección del puntero. Borra la pantalla e imprime la representación hexadecimal y los equivalentes. ASCII de los 160 bytes, empezando desde el valor del puntero. Las direcciones se escriben en decimal o hexadecimal dependiendo del estado del panel (ver [CTRL]D). La pantalla consiste en 20 filas de 8 bytes cada una, con el ASCII al final de ellas. A cualquier valor mayor de 127 se le resta 128 y entre 0-31 inclusive se representan como «». Al final de una página de listado, puedes volver al panel pulsando [ESC] o continuar con la siguiente página pulsando cualquier tecla.

«M» Coloca el puntero en una dirección determinada.

Se te pedirá una dirección hexadecimal. El puntero se colocará en dicha dirección y el panel mostrará la memoria de ahí en adelante. «M» es fundamental antes de introducir código máquina, tabular memoria, etc...

«N» Busca la siguiente aparición de la cadena hexadecimal especificada por el comando «G».

«O» va al destino de un salto relativo.

El comando toma el byte de la dirección del puntero, lo trata como un salto relativo y actualiza la memoria de acuerdo a este salto.

Recuerda que los saltos relativos mayores de #7F (127) se tratan como negativos. Ver comando «U».

«P» Llena la memoria entre dos límites de un valor dado.

«P» Pide la dirección de comienzo («First:»), la de finalización («Last:»), y el valor con el que hay que llenar el trozo de memoria elegido («With:»).

«R» Lee código objeto del cassette (o disco) creado por el comando «W» de MONA3 o bien los comandos «O» o «B» de GENA3. Se te pedirá el nombre del programa (pulsando [ENTER] si no lo sabes) y la dirección donde quieres cargarla. Esta dirección no puedes omitirla.

«>» Coloca un «breakpoint» después de la instrucción en la que está el puntero y continúa la ejecución.

«S» Actualiza el puntero de memoria, dándole el valor actual del stack (indicado por SP). Es útil para ver la dirección a donde una subrutina debe volver.

Mercado común

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

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

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A. AMSTRAD SEMANAL.**
Apartado de correos 54.026
28080 MADRID
¡ABSTENERSE PIRATAS!

El Club de Amstrad de Huesca desearía **contactar** con clubes o usuarios de **Amstrad**.
Nuestras señas son: José Juan Carranza Ferrer. C/ Ramón y Cajal, 25 - 4.º B. 22001 Huesca. Contestamos a todas las cartas.

«HACKER»
Persona que se introduce o interfiere ilegalmente en redes de ordenadores con ánimo de lucro.

Vendo Amstrad CPC-464 color, 36 juegos y utilidades (diseñador de pantallas, procesador de textos, contabilidad española, copiones...). Comprado en El Corte Inglés. Perfecto estado. Todo por 69.000 ptas. Roberto Agra. C/ General Ricardos, 122 - 3.º A. Madrid. Tel. (91) 471 58 30.

Vendo ordenador Amstrad CPC-664. Con unidad de disco, monitor de color y extras. Razón: Tel. (93) 780 85 78 de Barcelona.

Vendo ZX-Spectrum 48 K por 18.000 ptas. e impresora ZX-Printer por 9.000 ptas. a los dos por 24.000 ptas. Comprado hace año y medio; las dos cosas en perfecto estado. Lo vendo por compra de un **Amstrad CPC-664**. Regalo programas, libros y revistas. Angel Campo Rey. Madrid. Tel. (91) 231 25 99.

Deseo **contactar** con usuarios de **Amstrad** preferentemente de la provincia de Avila. Se despide atentamente: Félix Enrique Ventero Muñoz.

Se vende Amstrad CPC-464 con monitor GT 65 **Amstrad** (fósforo verde) y joystick (Quick Shot II). Sólo tiene un mes y garantía **Amstrad** España para 6 meses. También con guía del usuario en castellano e inglés, manual de referencia Basic para el programador, y programas comerciales (Beach Head, Everyone's a Wally, Decathlon, Knight Lore, etc.) Todo ello por 55.000 ptas. Interesados llamar de 15 a 16 al Tel. 269 34 11 de Madrid.

Se **cambia** monitor GT 65, de fósforo verde por monitor a color pagando diferencia a convenir. Tel. (91) 402 84 30. Madrid.

Vendo Amstrad CPC-664 con unidad de disco y monitor de fósforo verde. Se incluye:

- Garantía oficial **Amstrad**.
 - Manual en castellano.
 - Disco sistema operativo CP/M y Leng. Logo.
 - Disco original obsequio con siete programas.
 - Programa «Hoja de cálculo» con manual en castellano.
 - Libro «Hacia la inteligencia artificial con **Amstrad**».
- Todo en perfecto estado y embalado: por 80.000 ptas. Razón: Joaquín Borrell. Tel. 347 72 08. Barcelona (a partir 10 noche).

Vendo Amstrad CPC-664 monitor verde, con discos y con los mejores juegos y utilidades del mercado y con joystick. Todo por sólo 74.000 ptas. Tel. 458 60 76 tardes.

Vendo Amstrad CPC-464 monitor color comprado en marzo 1985 como nuevo con el manual en castellano. Con el lote incluyen muchos juegos en cassette y el Amsword (tratamiento de textos). Todo por 70.000 ptas. Tel. (93) 256 96 47. Noches. Jaime. Barcelona.

Me interesaría contactar con usuarios **Amstrad** para intercambios de programa, en cassette o en disco. También cambio trucos, ideas, etc. Paso programas de cassette a disco. Me interesaría comprar un buen programa de ajedrez. Luis García Cantabrana. C/ Argentina, 14 3.º A. 01009 Vitoria. Tel. 14 69 60.

Vendo Amstrad CPC-464 monitor fósforo verde, joystick, unidad de disco DD1 con su interface (aún en garantía), manuales en castellano y multitud de programas. Todo ello al precio de 90.000 ptas. Ignacio González Rodríguez. C/ Sáhara, 32. Madrid. Tel. (91) 796 15 46. Llamar de 9-13 h.

Vendo programa de Contabilidad General para **Amstrad** con disco. Totalmente original. Prestaciones profesionales. Totalmente modificable. Adaptable a cualquier impresora paralelo. 450 cuentas auxiliares, 99 cuentas mayores, 6.900 apuntes por disco. Sin límite de apuntes. José Luis Muñoz Traperó. Las Torres. Plaza Cejilla, 2 - 2.º D. Jerez de la Frontera (Cádiz). Tel. (956) 32 68 95 - 31 24 58.

Programador y usuario de un **Amstrad**, desearía contactar con usuarios de cualquier modelo **Amstrad**, para intercambio de cualquier tipo de programas. Realizo programas a medida por encargo (Programas de gestión). Ponerse en contacto con: Miguel Angel Fanega López de la Franca. C/ Reyes Católicos, 1 - 1.º A. Daimiel (C. Real) Tel. (926) 85 09 08.

Se intercambian juegos para el **Amstrad** CPC-464 con usuarios de Barcelona o provincia. Llamar de 15,30 a 17 h o de 21 a 22 h al Tel. 212 42 84. Preguntar por Iván.

Cambio ordenador **Amstrad** CPC-464 monitor color en perfecto estado, más 30 programas comerciales, por **Amstrad** CPC-664 monitor color pagando diferencia justa. Llamar al Tel. (93) 212 42 84 de Barcelona. Preguntar por Iván. (Llamar de 21,30 a 22,00 h.).

Club **Amstrad** busca usuarios en toda España para intercambio de información, programas, juegos, revistas, ideas..., etc. Mandaremos una lista con todos nuestros programas y libros a los que nos escriban al: Apdo. de Correos 1.582 de Bilbao. Referencia: Club **Amstrad**. Muy agradecidos.

Desearía contactar con usuarios del **Amstrad** CPC-6128, en Cádiz y su provincia. Interesados escribir a: Silvia Rodríguez Fernández. Avda. Ramón de Carranza, 22 - P. 6.º D. 11006 Cádiz. Tel. 28 72 80.

Vendo impresora Centronics 730-2, matricial, tracción y fricción, 80 - 40 a doble anchura-, o 132 columnas -66 a doble anchura-, 80 caracteres por segundo. Regalo además 500 hojas de papel continuo y 3 cintas de recambio (originales). Con cable y conector para el **Amstrad**: 33.000 ptas. Con conector estándar: 30.000 ptas. Interesados llamar al Tel. (91) 742 28 37, preferiblemente de 10 a 14 h. Claudio Pérez-Olea. C/ Telémaco, 26. 28027 Madrid.

Desearía **contactar** con usuarios/as de ordenadores en general y de cualquier zona del mundo. Interesados/as escribid a: Ramón J. Puchades. C/ Fco. Climent, 6 - 1.º. 46007 Valencia.

Cambio juegos comerciales para **Amstrad** (en cinta) preferentemente con gente de Barcelona. Entre ellos están Exploding Fist, Combat Linx, Decathlon, copiones, etc. Alberto Adeva Antón. C/ Eduardo Todá, 36 int. 5.º-1.º 08031 Barcelona. Tel. 427 08 32.

Por **cambio** de ordenador **vendo o cambio** juegos para **Amstrad** CPC-464. Los interesados, preferiblemente de Zaragoza, llamar al Tel. (976) 37 22 78 (Miguel) o al (976) 31 36 69 (Alberto).

Vendo Amstrad 464 verde por 45.000 ptas. También tengo algunos juegos que vendería a 500 ptas. unidad. Urge por mili. Llamar al Tel. (94) 681 22 97. Germán García. C/ Ambrosio Meabe, 5 - 2.º. Durango (Vizcaya).

Vendo Amstrad CPC-464, color. En perfecto estado, con manuales y los programas siguientes:

- Ensamblador y desensamblador Devpac de Amsoft.
- Battle for midway.
- Hunter killer.
- Roland in the caves.
- Combat lynx.

Precio actual de todo, 111.000 ptas. apróx. y lo vendo por 80.000 ptas. Llamar a partir de las 18,00 h. y preguntar por Mario.

Desearía **contactar** con usuarios del **Amstrad** CPC-464. José María Gómez Vega. P.O.BOX-381. Pto. Santa María (Cádiz).

Vendo curso Autodidáctico de Basic Tamos I y II, con dos cassettes para prácticas cada uno. Los dos por 3.500 ptas. Juan Puig Arnaú. C/ P.º Ramón Vall, 44 - 3.º-2.º. Navas (Barcelona).

Desearía **adquirir Amstrad** de segunda mano. Interesados dirigirse a D. G. Leorza. C/ Don Vela, 24 - 3.º Izqda. 01009 Vitoria (Alava).

Intercambio toda clase de programas. Interesados dirigirse a Luis García Cantabrana. C/ Argentina, 14 - 3.º A. 01009 Vitoria (Alava).

Intercambio programas de **Amstrad** (en especial CPC-6128), juegos, copiones, utilidades. En disco o cassette. ¡Mandar lista! Interesados escribir a: F. Guilabert Mora. C/ Marqués de Asprillas, 3. Planta 3.º. Puerta 1. Elche (Alicante).

Usuarios **Amstrad**, **intercambio** programas, preferiblemente juegos, versiones comerciales; dirigirse a: Juan Martínez. P.º Fabra y Puig, 305 - 2.º-2.º. 08031 Barcelona.

Vendo Amstrad CPC-464; monitor color, manuales uno en castellano y otro en inglés, 12 programas originales, un joystick «Quick Shot II». Con garantía durante 6 meses. Todo nuevo con sólo una semana de utilización. Precio: 85.000 ptas. Miguel A. Campos Moya. C/ La Paz, 9 - Bajo B. Cartagena (Murcia).

✕ **Desearía contactar** con usuarios de **Amstrad**, para intercambio o compra de programas de utilidades y juegos. Interesados escribir a: Casimiro Hernández. Avda. Antonio Machado, 13 - 3.º Santa Marta (Salamanca).

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



GABINETE DE INFORMÁTICA

- **Clases de Informática sobre AMSTRAD**
En grupos o individuales
- **Ordenadores AMSTRAD y periféricos**
Los mejores precios
- **Software a la medida**

ZURBANO, 4 ☎ 410 47 63
28010 MADRID

«HACKER DE ACTIVISION»

El juego líder de ventas en Europa. Disponible para Sinclair, Amstrad y Commodore.
P.V.P.: **2.200 ptas.**

En tiendas especializadas y grandes almacenes o directamente por correo o teléfono a: **PROEIN, S.A.**
C/ Velázquez, 10
28001 MADRID. Telf.: (91) 276 22 08-09

Vendo ordenador Amstrad CPC 664 con unidad de disco incorporada y monitor en color. Está completo, con garantía Indescomp y absolutamente nuevo. Precio 105.000 ptas. Interesados ponerse en contacto con Gregorio. Apartado 25084. 08080 Barcelona, o al tel. (93) 353 68 53. A partir de las 19 horas.

Vendo Spectrum 48 K., microdrive, interface 1, (garantías en vigor y con fecha en blanco), manuales en español, más de 50 revistas, libro Basic de Microhobby, programas (más de 200). Todo por 59.000 ptas. Tel. 777 01 67. Madrid.

Desearía contactar con usuarios del CPC6128, para el intercambio de programas, ideas, impresiones... Josep M.ª Escubedo. Balmes, 66 - 2 - 3. 08007 Barcelona.

Vendo Commodore 64. Incluyo Datassette C2N, joystick Quick Shot II, guía del usuario, Curso de Basic (2 libros), numerosas revistas especializadas, y un excelente surtido de juegos (One on One, Decathlon, Beach Head, etc.). Todo ello por 49.000 ptas.

Interesados llamar, de 14 a 16,30 al (972) 31 53 86. O bien, escribir a: Miquel Gasull Buenaventura. Alvaro de Bazán, 2. Palamós (Girona).



¡Operación cambio!

Valoramos:
Tu AMSTRAD 464 en 50.000 ptas.
Un Spectrum+ en 30.000 ptas.
Amstrad CPC en 70.000 ptas.

En la compra del AMSTRAD CPC 6128 o PCW 8256.
Consulte para monitor color.
Precios especiales en impresoras y accesorios.

☎ 270 34 97.

Desearía contactar con usuarios del **Amstrad CPC464** para el intercambio de programas. Interesados escribir o llamar a Gustavo Yenes. Paseo de Heriz, 82. 20008-San Sebastián. Tel. (943) 21 58 38.

Quisiera establecer contacto con usuarios de **Amstrad** en Granada, para cambiar listados e ideas. Preguntar por Miguel llamando al Tel. 22 36 66 o bien escribiendo a Cuesta / Escoriaza, 9 - 1.º B. Granada.

De particular a particular **Amstrad**. Por reciente viaje a Inglaterra, **vendo** 12 de los mejores juegos del mercado inglés a precio español. Entre ellos el James Bond 007, A view to a kill, Brian Bloodaxe de los Monthy Piton, Danger Mouse, Defender (versión **Amstrad** del clásico BBC) Dragon's Gold y el famoso Rally II. Llamar al (96) 332 12 72 de Valencia y preguntar por José. José Fco. Sanfeliú Gimeno. Barón de Cárcer, 32, 8.º - 15.º 46001 Valencia.

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

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

AVISOS A NUESTROS LECTORES

Comunicamos a todos nuestros lectores que a partir del día 1 de enero de 1986 nuestro Departamento de Clientes será trasladado al Polígono Industrial de Alcobendas, C/ La Granja, 39. Alcobendas (Madrid). Para cualquier consulta que deseen realizar pueden llamar a los teléfonos **654 28 98** y **654 28 67** de 9.00 a 14.30 h. y de 16.00 a 18.30 h.

MICRO-1

Duque de Sexto, 50. 28012 Madrid
Tels. (91) 274 53 80-276 96 16



SOFTWARE: ¡¡GRATIS 1 BOLIGRAFO DE ACERO CON RELOJ INCORPORADO!!

	Ptas.		Ptas.
Bounty Bob	2.300	Bruce Lee	2.300
Fighting Warrior	2.100	Yier Kung Fu	2.300
Foupack	3.890	Exploding Fist	2.300
Combat Lynx	2.100	Southern Belle	2.300
Dummy Run	2.100	Dragonorc	2.300
Raid	2.300	Rocky	2.100
Match Day	2.300	World Series Baseball	2.100
Map Game	2.750	Dambusters	2.300
Hypersports	2.300	Ajedrez Tridimensional	1.975

IMPRESORAS: ¡¡20% DE DESCUENTO SOBRE P.V.P.!!

Lápis óptico DKTronics	4.850	Diskette 3"	1.050
Tapa metacrilato AMSTRAD	1.975	Cinta C-15 (especial)	85
Toshiba MSX 64 K	39.900	Cassette Especial	5.295

Joystick Quick Shot II
2.495 ptas.

Joystick Quick Shot I
1.995 ptas.

Joystick Quick Shot V
2.995 ptas.

Increíbles precios para tu AMSTRAD
464 y 6128 (verde y color).
(Llámanos y te asombrarás)

PC Compatible IBM 256 K
Monitor Fósforo Verde
2 Bocas Diskette 360 K
279.000 ptas.

Sabrewulf + Decathlon + Beach
Head + Jet Set Willy
2.500 ptas.

Commodore 64: 42.900 ptas.
Commodore 128: 74.900 ptas.

El pedido te lo enviamos URGENTEMENTE contra-reembolso SIN NINGUN GASTO DE ENVIO, LLAMANDO a los teléfonos:(91) 276 96 16-274 53 80 o escribiendo a MICRO-1. Duque de Sexto, 50. 28012 Madrid.

SACALE EL JUGO A TU ORDENADOR. DISEÑA TUS PROPIAS PANTALLAS Y DIVIERTETE JUGANDO CON...

Este mes:

YOUR COMPUTER

Te ofrece algo realmente sabroso:

MUSICA

Este magnífico programa escrito en código máquina te permitirá manejar el sonido y la música en tu Amstrad desde Basic, mediante un nuevo juego de comandos creados especialmente para ello.

CROSS

Tienes cuatro revólveres para destruir a tus enemigos en el mínimo tiempo posible. Necesitarás toda tu habilidad, rapidez de reflejos y suerte, mucha suerte.

JUMPER

Debes alcanzar la cima del Valle de las Cintas Deslizantes. Tienes que saltar por los huecos de las valladas, que se desplazarán a derecha e izquierda con una rapidez de vértigo.

MAGGOT

Te encuentras en la amable tierra de las setas gigantes. Tu misión es guardarla del ataque y la invasión de una peligrosísima serpiente polimórfica que las ataca sin piedad.

TIMEBOMB

Una organización terrorista de Oriente Medio ha colocado una bomba de tiempo en el laberinto de defensa del Laboratorio de investigación bacteriológica de Lexington.

RSX

Your Computer ha pensado en los usuarios del Amstrad CPC464 y ha creado un nuevo juego de comandos completo para tu ordenador, de forma que el Basic así ampliado no tenga nada que envidiar al de los otros modelos de la serie.

2

YOUR COMPUTER

EL CORAZON DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES

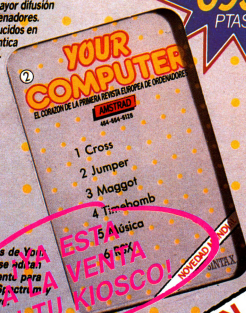
AMSTRAD

464-664-6128

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de Europa en ordenadores.

Ahora reproducidos en cassette. En auténtica exclusiva mundial.

695.
PTAS.



Las cintas de Your Computer se venden exclusivamente para Amstrad, Spectrum y Commodore.

ESTA REVISTA EN TU KIOSCO!

IGANA UN 128 K!
Total garantía de carga

SINTAX, S.A.

Si no lo encontrara en su kiosco, puede solicitarlo directamente a nuestra editorial:
Paseo de la Castellana, 268. Tel.: (91) 733 25 99. 28046 Madrid.