

MICROHOBBY

AÑO I N.º 6

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO I N.º 6

150 Ptas.

Canarios 160 ptas.

**TECNICAS
DE ZOOM**

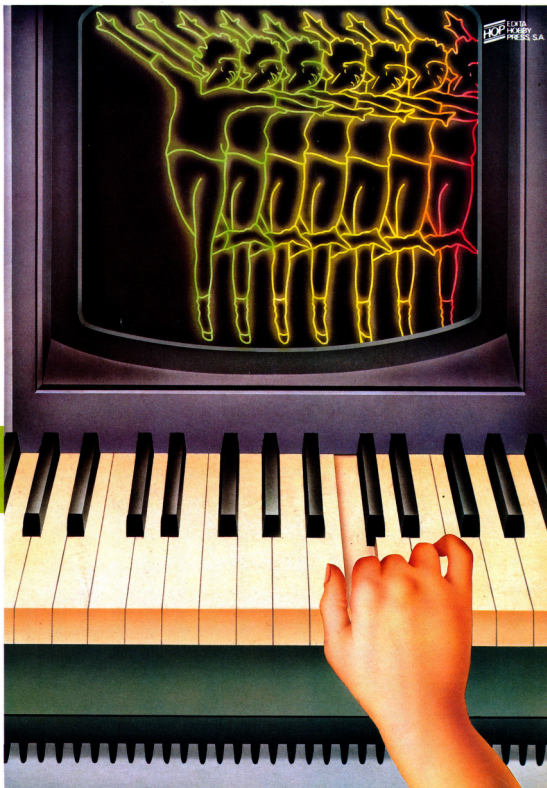
**MAS
SOBRE
LOS
COLORES**

**AMSTRAD
SINFONICO**

**PROGRAMACION
INTERACTIVA**

SOFTWARE

**3D VOICE CHESS:
AJEDREZ EN TRES
DIMENSIONES**



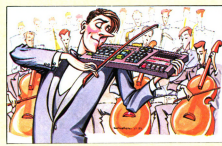
HOPI
LEITA
HOBBY
PRESS S.A.

MICROMANIA. Sólo para adictos

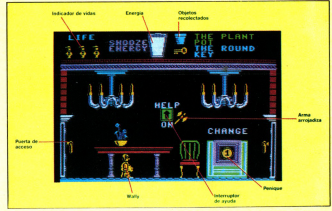


«COMO SE PROGRAMA UN JUEGO»

... para aprovechar a tope tus posibilidades como programador.



«AMSTRAD MUSICAL»... amplios artículos a tu medida.



«PATAS ARRIBA»... la sección que destripa los mejores juegos, POKE A POKE, para hacerte invencible.

Una revista con marcha para los que necesitan saber TODO sobre ordenadores.



... Y además, la posibilidad de ganar una POLAROID si encuentras al travieso BYTE enmascarado.

¡MUSICA MAESTRO!
TODA UNA ORQUESTA EN TU AMSTRAD

PATAS ARRIBA LOS "POKES" DE PYJAMARAMA Y AUTOMANIA



LO NUNCA VISTO: ¡MAPA EN TRES DIMENSIONES DEL EVERYONE'S A WALLY!

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

Año I • Número 6 • 8 al 14 de Octubre de 1985
150 pts. (sobretasa Canarias, 10 pts.)

Director Editorial
José I. Gómez-Centurió
Director Ejecutivo
Victor Prieto

Subdirector
José María Díaz
Redactora Jefe
María García

Diseño
José Flores
Colaboradores
Francisco Portalo
Pedro Sudón
Miguel Sepúlveda
Francisco Martín
Jesús Alonso
Pedro S. Pérez
Amalio Gómez
Juan J. Martínez

Secretaría Redacción
Carmen Santamaría

Fotografía
Carlos Candel
Javier Martínez
Portada
Manuel Barco

Ilustradores
J. Igual, J. Pons, F. I. Frontán,
J. Septien, Pejo, J. I. Mora,
Luigi Pérez

Edita
HOBBY PRESS S.A.

Presidente
Moria Andriano
Consejero Delegado
José I. Gómez-Centurió

Jefe de Publicidad
Concha Gutiérrez
Publicidad Barcelona
José Galán Cortes
Tel: (93) 303 10 22/313 71 62

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, s/n
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49 480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedit, S. A. Valencia, 245
Barcelona

Impreso
ROTEC, S. A. Cita, de Irún,
Km. 12,450 (MADRID)

Fotocomposición
Novacom, S.A.
Nicolás Morales, 38-40

Fotomecánica
GROF
Ezequiel Solano, 16

Depósito Legal:
M-28468-1985

Derechos exclusivos
de la revista
**COMPUTING with
the AMSTRAD**

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

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

Se solicitará control OJD

5 Primera plana

Suponiendo que a nuestros lectores les interesa, además de lo puramente concerniente al **Amstrad**, lo que suceda de importancia y actualidad en el terreno informático, continuamos con la política de hacernos eco en estas páginas de esa inquietud, mostrando una panorámica verdaderamente de Primera Plana de todo aquello que es noticia en el mundo del ordenador.

6 Primeros pasos

Seguimos adentrándonos en la comprensión y uso de las variables en programación. Además, comenzamos a estudiar la sentencia INPUT, de capital importancia para dotar de vida e interacción a nuestros programas.



10 ProgramAcción

El mundo de los colores en el **Amstrad** es lo suficientemente rico como para necesitar una revisión detallada y profunda.

En este número, continuando con lo expuesto anteriormente, examinamos las técnicas que nos permitirán dominar completamente su utilización.

14 Mr. Joystick

Tal vez los programas más inquietantes de los ordenadores sean los de ajedrez, en los que una máquina parece pensar y actuar como nosotros mismos lo haríamos. En esta línea, un programa de ajedrez en tres dimensiones que no sólo juega aceptablemente bien, sino que incluso no dice las jugadas que se van realizando durante la partida de viva voz, merece la pena conocerlo.

18 Amstravagancia

En esta sección, que aparecerá de vez en cuando en nuestras páginas, pretendemos mostrar todos los múltiples aspectos que constituyen el mundo de la informática y de formas muy distintas, algunas sorprendentes, otras atrevidas, pero siempre procurando mantener el interés y el buen gusto. Esperamos que la sorpresa de este número sea bienvenida.

20 Serie oro

El **Amstrad**, como casi todos los ordenadores de hoy día, destaca por sus posibilidades de sonido.

Music es un programa que permite, tanto al no iniciado como al conocedor, experimentar con todo lo que el sonido puede dar de sí en la creación de música. Tres instrumentos distintos, una buena presentación y un manejo sencillo y amigable, garantizan mucho tiempo de creatividad y diversión.

Después del relax de una buena música, a lo mejor no viene mal algo de acción. Tron, el segundo Serie Oro, la proporcionará con creces.



26 Código Máquina

Todos sabemos ya la importancia de los registros para el programador en código máquina. Pues resulta que el Z80 da más de sí, porque permite agruparlos por pares, incrementando significativamente la potencia y claridad de nuestras rutinas.

30 Análisis

Casi podríamos decir que sobran las palabras. Resulta difícil creer que el humilde Basic pueda realizar con tanta velocidad y sencillez uno de los efectos más difíciles: dar sensación de animación tridimensional. No dudamos que podréis incorporar rápidamente esta técnica a vuestros programas.

3-D VOICE CHESS

Ajedrez tridimensional con voz en castellano
Amstrad CPC 464, CPC 664 y CPC 6128

P.V.P.
2.300.- (cinta)
3.300.- (disco)



PUBLISHED:
DEEP THOUGHT
SOFTWARE

DISTRIBUTED:

cp software



Producido en exclusiva para España por:

ACE

Actividades Comerciales Electrónicas, S.A.
Tarragona 100 - Tel 325 10 58 - 08015 Barcelona Telex 93133 AC EÉ E

YA DISPONIBLE EN



Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

PASCAL EN DISCO

En la línea de dotar el **Amstrad** del mayor número de lenguajes posibles distintos de Basic, sobre todo lenguajes compilados que permitan incrementar considerablemente las prestaciones del ordenador mediante una mayor velocidad, y una menor ocupación del espacio de memoria disponible, Indescomp comercializa desde hace algún tiempo la versión de Hisoft Pascal para **Amstrad** en disco.

Los programas de utilidades de Hisoft son bien conocidos de los usuarios de otros ordenadores, como el Spectrum, y destacan por su facilidad de manejo y sus altas prestaciones, dentro de las lógicas limitaciones que un ordenador del rango del **Amstrad** impone.

Es de destacar la posibilidad de obtener ficheros en código máquina, compilados, directamente ejecutables por el ordenador sin necesidad de la presencia del compilador en memoria.

El aumento de velocidad con respecto a un mismo programa escrito en Basic es del orden de un 40 por 100 como mínimo.

EUROPA UNIDA

Con idea de desarrollar una nueva generación de ordenadores, no se sabe bien todavía hasta qué punto revolucionarios, las firmas Thompson, Olivetti y Acorn han llegado a un acuerdo de colaboración en el campo de los micros caseros, en cuanto a investigación, fabricación y venta de los mismos.

El fuego parece ir dirigido hacia Japón, tratando de frenar el avance del «standard» MSX, y hacia el otro lado del Atlántico.

Parece que a la Europa informática le preocupa también los monstruos americanos Apple, Commodore y Atari (*tienen motivos*). De la verdadera fiera, de IBM, nadie habla acerca del impacto que el acuerdo europeo pueda tener en su comercio, por la sencilla razón de que el «gigante azul» ni siquiera va a enterarse de ello comercialmente hablando.

Todo el mundo se apresura a establecer un «Standard Internacional», a ver si suena la flauta y sólo venden ellos al mayor número de hogares de hoy posibles.

Ya se sabe, Dios los cría...

TARJETAS INTELIGENTES

Hace poco tiempo, la empresa francesa Bull, fabricante de micro y miniordenadores, desarrolló de la mano del ingeniero Roland Moreno, una especie de tarjeta de crédito dotada de memoria y de un microprocesador, esto es, una tarjeta de crédito inteligente. La importancia a corto plazo de esto puede ser enorme, simplificando enormemente las transacciones bancarias, por ejemplo.

A la vista de ello, 5 de las más importantes empresas informáticas y electrónicas de Japón, pueden estar interesadas en introducir el invento en su país.

Al parecer, el acuerdo tomaría la forma de la creación de una filial japonesa de la empresa Innovatron, que detenta la patente de la tarjeta inteligente, a cambio de la cual los japoneses instalarían en Francia una factoría preparada para el uso de la tecnología inventada por Moreno.

Este ha anunciado también que se encuentra en negociaciones con Apple Computer (*creadora del Macintosh*), para que se incorpore un lector de tarjetas a la próxima generación de ordenadores personales de la firma americana. Posibles beneficios: acceso seguro a los bancos de datos electrónicos, y creciente éxito en la lucha contra la piratería, entre otros.

Primera plana

SINCLAIR SE RENUEVA

La compañía inglesa Sinclair Research, creadora del archifamoso Spectrum, presenta ahora un modelo de 128 Kbytes de memoria, siguiendo las actuales tendencias del mercado de los ordenadores caseros.

Lo verdaderamente interesante para nosotros, es que el nuevo Spectrum se presenta en España como primicia mundial, y se fabricará íntegramente en nuestro país, por la empresa Investrónica, distribuidora oficial de los productos Sinclair.

La máquina se presentó en el Sonimag de Barcelona, el día 23 de septiembre, y, al parecer, posee sustanciales mejoras respecto a su antecesor, el Spectrum Plus.

Puede funcionar en dos modos, uno de 64 K, para guardar compatibilidad con el software escrito para modelos anteriores, y, por supuesto, el modo 128 K.

En este último, de la gestión de pantalla se encarga otra ULA diferente del modo emulador Spectrum Plus.

El basic ha sido modificado y mejorado, de tal manera que los programas Basic en modo 128 K no correrán en el Spectrum «normal».

Su precio rondará las 50.000 ptas.



BASIC EN 2 DIMENSIONES

F. L. Frontán

A la hora de trabajar con algunos programas en los que utilizamos cadenas de caracteres, se nos presentan ocasiones en las que nos es más conveniente utilizar variables en lugar de dichas cadenas. Por ejemplo:

```
A$ = «AUSTRALIA»
```

significa que a partir de esta instrucción, podremos utilizar la variable A\$ en lugar de la cadena «AUSTRALIA». Es decir, que:

```
PRINT A$
```

Le imprimirá siempre AUSTRALIA. Para la asignación de las variables de cadena podemos utilizar una o varias letras, incluso palabras, pero siempre han de ir seguidas de el carácter \$, el cual le indica al ordenador que dicha asignación corresponde a una cadena, de tal forma que ésta pasará a ser lo que llamamos variable de cadena. La denominación de variable viene dada porque el contenido de la misma podrá cambiar a lo largo del programa.

El programa I nos muestra este punto.

Como verá si lo corre, el valor de A\$ varía según las distintas asignaciones que se realizan durante el programa. A\$ siempre toma el último valor asignado.

La utilidad de las variables

Puede que se esté preguntando para qué le sirve utilizar la misma variable para usos diferentes en lugar de emplear una distinta para cada uno de ellos, pero como veremos más adelante ésta es una circunstancia que en algunas ocasiones nos resultará bastante útil, pues conseguiremos reducir las variables de cadena a simples letras seguidas por el carácter.

El programa II, el más complicado de los que utilizaremos hoy, nos muestra un ejemplo del uso de estas variables.

Hasta ahora hemos estado trabajando sin darle importancia al hecho de que las letras fueran mayúsculas o minúsculas, pero como ya sabrá, en Basic las palabras clave deben ir siempre escritas en mayúsculas. Su Amstrad ya viene preparado para que esto ocurra así, pues aunque tecleemos:

```
10 print «Hola»
```

saldrá impreso como:

```
10 PRINT «Hola»
```

es decir, que la palabra clave será transformada a mayúsculas, pero como podrá ver, la palabra «Hola» no sufre ningún cambio, puesto que al venir entre comillas se la considera como una cadena y éstas son totalmente invariables.

Todas las sentencias siguientes:

```
10 pRinT «Hola»
```

```
10 PRinT «hola»
```

```
10 PRiNT «HOLA»
```

saldrán listadas como:

```
10 PRINT «Hola»
```

En el programa II vemos que todas las variables (nombre, hecho, tratamiento), vienen escritas en minúsculas. Esto ocurre porque yo lo he tecleado así, pero aquí hay una cuestión interesante.

En su Amstrad:

```
10 PRINT tratamiento$
```

y

```
10 PRINT TRATAMIENTOS$
```

aparecerá escrita de diferente manera, pero la variable será la misma, puesto que el Amstrad no diferencia entre tratamiento\$ y TRATAMIENTOS\$. En las variables las mayúsculas y las minúsculas son equivalentes. Así pues:

```
tratamiento$
```

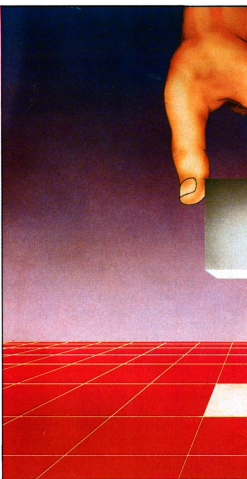
```
TRATAMIENTOS$
```

```
TRAtAMienTo$
```

```
trataMIENTOS$
```

serán consideradas como la misma variable.

El programa III nos muestra esta idea. Las líneas 30 y 40 asignan ca-



denas diferentes a la misma variable, ¡las apariencias engañan! El valor asignado en la línea 60 sustituye al de la 50, tal y como vimos que ocurría en el programa I.

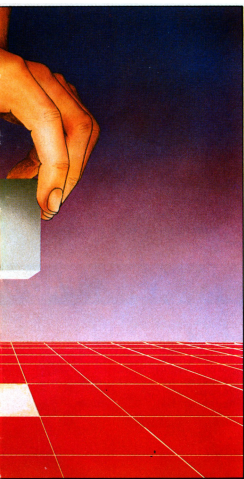
Medidas de seguridad

Mi recomendación es que siempre teclee las variables en minúsculas, de esta manera podrá distinguir fácilmente entre éstas y las palabras clave. Esto no simplifica mucho el hecho de teclear un programa, pero es una buena costumbre de programación, pues le permitirá saber a simple vista qué es cada cosa.

Otra buena costumbre es asignar a las variables nombres que estén relacionados con su contenido, logrando así que sus programas tengan más sentido.

Esto es lo que hemos hecho en el programa III, asignando «nombre\$» a «Sr. Gómez», «hecho\$» a «me debe dinero» y «tratamiento\$» a «págueme o si no...».

Esta práctica puede parecer un poco engorrosa, pero se dará cuenta de que le ayudará a hacer sus programas más legibles y, por tanto, más fáciles de descifrar.



Por ejemplo:

```
70 PRINT «Querido» nombre$  
explica mucho mejor lo que está re-  
alizando que:
```

```
70 PRINT «Querido» A$  
al igual que:
```

```
80 PRINT tratamiento$  
lo hace mejor que:  
80 PRINT BS
```

La moraleja que se desprende de todo esto es: **UTILICE PALABRAS PARA ASIGNAR A LAS VARIABLES SIEMPRE QUE LE SEA POSIBLE Y PREFERIBLEMENTE ESCRIBALAS EN MINUSCULAS.**

Reglas de nomenclatura

Actualmente podemos usar letras mayúsculas para las variables e intercalarlas con minúsculas y con números. Las reglas que rigen esto son:

— Todo nombre de una variable debe comenzar con una letra, aunque puede ir seguido por cualquier tipo de letra, número o carácter. Las letras mayúsculas y minúsculas son equivalentes.

— No está permitida intercalar espacios en blanco en el nombre de una variable.

— Las variables deben permanecer separadas de las palabras clave de Basic. Uno de los errores más comunes es colocar una variable junto a un comando.

Una de las mayores ventajas de utilizar variables en lugar de cadenas es que nos permite alterar fácilmente la salida del programa. En el caso de el programa II, si queremos que nuestra carta le llegue a otra víctima, tan sólo deberemos cambiar la línea 30. Por ejemplo:

```
30 nombre$ = «Sr. Pérez»
```

A partir de ese momento todos los usos de nombre\$ estarán referidos a el Sr. Pérez.

En este caso, al ser un programa corto no reviste mucha importancia, pero en otros más largos si ha utilizado la cadena «Sr. Gómez», en lugar de una variable, se verá obligado a realizar un gran número de modificaciones, una por cada vez que la haya utilizado.

El hecho de escribir las variables en minúsculas nos ayudará, sobre todo a los recién iniciados en la programación, a no confundirlas con las palabras clave.

Teclée el programa IV tal y como se muestra, asegurándose de que hay un espacio en blanco después de el print de la línea 40.

Cuando lo liste aparecerá:

```
Corra el programa y todo irá bien.
```

Sin embargo, suponga que hubiere olvidado el espacio entre print y palabra. Teclée así la línea 40:

```
40 printpalabra$
```

Ahora teclée LIST y RUN. Obtendrá el mensaje: «**Syntax error in 40**» (Error de sintaxis en la línea 40). Como podrá ver, ésta aparece ahora como:

```
40printpalabra$
```

y su micro no sabe qué hacer con esta variable.

Generalmente, las variables tienen a «absorber» a los comandos que están junto a ellas para crear otras más largas. De hecho, el comando «desaparece» de la línea, dejando al ordenador en serias dudas de qué hacer con esa nueva variable.

PARA EVITAR ESTAS CONFUSIONES, DEJE ESPACIOS ANTES Y DESPUES DE LAS PALABRAS CLAVE. Otra posibilidad de la que se dispone es de colocar ciertos caracteres como separadores o delimitadores.

Primeros pasos

Las comillas, por ejemplo, actúan como delimitadores:

```
PRINT«Hola»
```

Si cogemos la buena costumbre de escribir las variables en minúsculas, **todas las palabras clave estarán en mayúsculas y de este modo podrá ver fácilmente dónde ha cometido el error**, bien por haber omitido el separador o por haber olvidado el comando.

Bueno, una vez hechas estas advertencias, volvamos al programa II.

Formateando el texto: el comando TAB()

Este nos ayudará para estudiar una nueva idea, el uso de la función TAB(). Esta función nos permitirá especificar a qué distancia a lo largo de una línea queremos que nos imprima el texto de un comando PRINT.

En el Mode 1 la línea tiene 40 caracteres y, por tanto, la pantalla se puede considerar dividida en 40 columnas. TAB() decidirá en cuál de esas columnas empezará a escribir. La numeración de las mismas va desde 1 a 40.

Según en el Mode con el que estamos trabajando, así será el número de caracteres de la pantalla, es decir, el número de columnas. Por ejemplo, el Mode 2 sólo tiene 20 columnas.

Pruebe a cambiar la línea 20 de este programa por:

```
20 MODE 0
```

¿Ha visto lo que pasa?

Uno de los mayores errores de algunos supuestos buenos programas es su mala colocación en la pantalla. Un uso correcto de la función TAB() puede evitar estos problemas.

Para coger alguna práctica, veamos el programa IV. Este imprime un triángulo de asteriscos. ¿Podría usted inventar un programa que, con la ayuda de TAB(), dibujara una estrella de asteriscos?

Antes de que empiece le diré que lo podrá realizar más fácilmente si vuelve a Mode 1 de la siguiente manera:

MODE 1 (ENTER)

Y por el momento, esto es todo lo que tenemos que decir acerca de las variables de cadena. Sin embargo, existe otro tipo de variables, que son las llamadas variables numéricas.

Las variables numéricas: segunda clave

No existe prácticamente ninguna diferencia en la asignación de éstas y las de cadena, excepto que, lógicamente, en lugar de estar formadas por una serie de caracteres, lo estarán por números y, por tanto, podremos realizar con ellas una serie de preparaciones matemáticas.

Pruebe a correr el programa VI.

La línea 30 utiliza la variable numérica A para asignar al número 10. Dese cuenta de que para las variables numéricas no es necesario el uso del carácter \$, pues es suficiente con asignarle una letra del alfabeto. También habrá comprobado que tampoco se necesita poner el valor de estas variables entre comillas, como se puede ver claramente en la línea 30:

```
30 A=10
```

La línea 40 por supuesto que no imprimirá A, sino el valor que ésta tiene asignado, que en este caso es 10.

Lo más interesante ocurre en la línea 50. Aquí, multiplicamos el valor de A por 2 y, por tanto, se imprimirá 20.

Este es el punto más importante de las variables numéricas, el poder operar con ellas.

Algo importante de señalar es que el micro realiza los cálculos correspondientes, pero tan sólo imprime los resultados. Por ejemplo, en este caso no ha escrito 2×10 , sino tan sólo 20.

Una operación como $2 \times A$ ó $A \times B$ se conoce por el nombre de expresión numérica. Cuando el micro encuentra una, la realiza e imprime la solución en lugar de la expresión misma.

Corra el programa VI con las siguientes variaciones:

```
50 PRINT A+B
50 PRINT A/4
50 PRINT A x A
```

Todas las variables son intercambiables

Por lo dicho hasta el momento, puede estar pensando que las variables de cadena son para asignar palabras y que las variables numéricas son para los números. Las cosas no son nunca tan sencillas. Se puede hacer uso de las variables de cadena para asignar números, con la diferencia de que no podrá operar con ellos. Mire el programa VIII, el cual está basado en el programa VI, utilizando la cadena A\$ en lugar de la numérica A.

El mensaje «Type mismatch in 50» (Tipo incompatible en línea 50), que hemos recibido, muestra que estamos intentando realizar una operación con un tipo de variable errónea.

Como para las variables de cadena no tenemos (y no debemos) limitarnos a utilizar una sola letra para la asignación de las variables numéricas. **Podemos utilizar palabras de una manera exactamente igual que con las cadenas, pero omitiendo el signo \$ y por supuesto, tampoco podemos utilizar las comillas.**

También en este tipo de variable las mayúsculas son equivalentes a las minúsculas, y por tanto A será lo mismo que a.

Fíjese en el programa VIII. Está puesto para saludar amablemente a alguien que corra el programa. Sin embargo, tal y como está en este momento está ligeramente restringido, pues sólo un pequeño porcentaje de alumnos se llaman Miguel. Lo que realmente se necesita es la manera de que el ordenador descubra el nombre de la persona que lo va a utilizar, para que pueda realizar un mensaje a su medida.

El comando Input

El programa IX nos saca de apuros. El truco consiste en el uso de INPUT nombre\$ de la línea 40. La línea 30 de el programa VIII asigna el valor MIGUEL a la variable nombre\$. En el programa IX esta variable no tiene aún un valor determinado.

Cuando el programa llega a la línea:

```
40 INPUT nombre$
```

espera hasta que le introduzcamos (INPUT), el valor que queremos de nombre\$ para imprimir dicho valor.

En otras palabras: cuando el ordenador se encuentra con un INPUT seguido de una variable, le pregunta qué quiere que valga la variable, de hecho le muestra un signo de interrogación en la pantalla.

Ahora está en disposición de teclear la respuesta seguida de ENTER, que será quien se encargue de enviarlo al ordenador, para que éste siga con el desarrollo del programa.

Así, cuando corra este programa, la línea 30 le preguntará: «¿Cómo te llamas?», pero dese cuenta de que no son necesarias las interrogaciones, pues la línea 40 se encarga de esto.

PROGRAMAS

```
10 REM PROGRAMA I
20 MODE 1
30 A$="AUSTRALIA"
40 PRINT A$
50 A$="AMERICA"
60 PRINT A$
70 A$="AFRICA"
80 PRINT A$
```

```
10 REM PROGRAMA II
20 MODE 1
30 nombre$="Sr. Perez"
40 hecho$="Usted me debe dinero"
50 causa$="pagueme o si no ..."
60 PRINT
70 PRINT "Pobre "nombre$
80 PRINT TAB (4) hecho$causa$
90 PRINT TAB (15) "Atentamente"
100 PRINT TAB (17) "Miguel"
```

```
10 REM PROGRAMA III
20 MODE 1
30 letras$="Primer0"
40 LETRAS$="Segundo"
50 PRINT letras$
60 PRINT LETRAS$
```

```
10 REM PROGRAMA IV
20 MODE 1
30 palabra$="Hola"
40 PRINT palabra$
```

```
10 REM PROGRAMA V
20 MODE 1
30 PRINT
40 PRINT TAB (5) "*"
50 PRINT TAB (4) "*"
60 PRINT TAB (3) "*"
70 PRINT TAB (1) "*****"
```

Entonces el micro esperará nuestra respuesta y la enviará mediante ENTER. Sea lo que sea lo que hayamos escrito, pasará a ser el valor de nombre\$, aunque le hayamos mentido.

La línea 50 imprimirá el mensaje.

Lo más interesante de todo esto es que, al contrario de lo que ocurría en el programa VIII, en el programa IX el valor de nombre\$ no está fijado desde el principio, pero éste es tomado durante el programa gracias a INPUT.

Por todo esto, cada estudiante de la clase podrá ahora correr el pro-

grama y obtener un mensaje adecuado para cada uno.

A propósito, la línea 30 no es estrictamente necesaria, pero sirve para indicar a la gente qué clase de respuesta está esperando. Por otra parte, se encontrarán con un signo de interrogación seguido de un cursor.

El punto y coma del final de la línea 30 «introduce» la interrogación o «sugiere» el mensaje precedente. Si corre el programa omitiéndolo, lo verá más claro.

Recuerde también que cuando corra el programa IX y le puegno su nombre, debe teclear la respuesta y después ENTER, pues si omite este último, su Amstrad no la recibirá y continuará esperando, y podrá resultar bastante aburrido.

Si se equivoca antes de presionar ENTER puede borrar utilizando DELETE. Pero si ya lo había presionado, tendrá que cargar con lo que haya teclado.

También se puede utilizar INPUT con variables numéricas. El programa X nos lo demuestra. Córralo y cuando aparezca la interrogación introduzca una palabra en lugar de un número y verá lo que ocurre.

Una aplicación más seria de INPUT le permite calcular el producto de dos números, como nos demuestra el programa XI.

Fíjese detenidamente en la línea 70 y vea si puede descubrir qué es lo que hace.

Primeros pasos

En primer lugar, no está entre comillas la variable primero y, por tanto, el micro imprimirá el número que le llegue primero. La frase «Multiplicado por», aparecerá literalmente al estar entre comillas.

La variable numérica segundo no está entre comillas, aunque así lo parezca, pues las comillas de la izquierda están ya emparejadas con las situadas detrás de primero y, por tanto, no tienen ninguna relación con segundo.

«Es» se imprime literalmente al venir entre comillas. Primero x segundo no lo está, por tanto la operación se realiza y se escribe la solución.

Por último, cambie el programa XI para que realice sumas o restas de dos números.

Sólo nos queda desear que haya pasado un rato agradable con nuestros programas y sugerirle que si tiene algún problema vuelva a leer el artículo desde el principio y verá cómo todo le resulta más fácil. También debemos recordarle que éste es un curso manual y que si no realiza los programas expuestos le resultará mucho más difícil su comprensión.

RAMAS

```
0 REM PROGRAMA VI
0 MODE 1
0 A = 18
0 PRINT A
0 PRINT 2 * A
```

```
0 REM PROGRAMA VII
0 MODE 1
0 A$ = "18"
0 PRINT A$
0 PRINT 2 * A$
```

```
0 REM PROGRAMA VIII
0 MODE 1
0 nombre$ = "Miguel"
0 PRINT "Buenos dias, "nombre$
```

```
0 REM PROGRAMA IX
0 MODE 1
0 PRINT "Cual es tu nombre ";
0 INPUT nombre$
0 PRINT "Buenos dias, "nombre$
```

```
0 REM PROGRAMA X
0 MODE 1
0 PRINT "Que edad tienes ";
0 INPUT edad
0 PRINT "No creo que tengas ";edad
```

```
0 REM PROGRAMA XI
0 MODE 1
0 PRINT "PRIMER NUMERO ";
0 INPUT primero
0 PRINT "SEGUNDO NUMERO ";
0 INPUT segundo
0 PRINT primero " multiplicado por
el " segundo " es igual "
primero*segundo
```



OPERACION ARCO IRIS

J. Igual

Ya hemos investigado con cierto detalle la forma en que funcionan los colores en el Amstrad. Sabemos también que existen 3 modos de pantalla, cada uno de los cuales implica el uso de un número diferente de colores, y permite distinta cantidad de caracteres por línea.



R

Recordaréis también que el comando PEN permite cambiar el color del texto que escribamos en pantalla.

Por ejemplo, si acabamos de encender el ordenador, la orden:

PEN 2

cambiará el color a cian, PEN 3 a rojo, etc.

Consideramos que cada pluma (PEN) se llena con un tintero distinto (INK), de forma que PEN 2 NO ES CYAN, sino que la tinta que en este caso llena el tintero 2 SI ES CYAN.

Para cambiar el color del fondo, esto es, el papel (PAPER), usaríamos:

PAPER 3

que pondrá nuestro texto sobre fondo rojo.

Esta orden quiere decir «ponga el papel del mismo color que la tinta presente en el tintero número 3».

Por último, para poner toda la pantalla de un color de fondo de una vez, se usa:

PAPER 3: CLS

El borde de la pantalla puede adoptar los 27 colores de que dispone el Amstrad, mediante el comando BORDER seguido por un número de 0 a 26, independientemente del modo de pantalla en el que nos encontramos.

Si tecleamos:

BORDER 0

y luego:

PAPER 0

¿Veis la diferencia?

Border 0 quiere decir poner el borde con el color de tinta cuyo número es cero, es decir, negro (Tabla 1), mientras que Paper 0 significa, como ya dijimos, el color de tinta asignado a la pluma número 0, azul.

Colores por defecto

En modo 1, las 4 plumas disponibles tienen por defecto una serie de colores asignados:

PEN 0, tinta azul brillante.

PEN 1, tinta amarilla brillante.

PEN 2, cian brillante.

PEN 3, rojo brillante.

cuando el Amstrad se enciende o inicializa (tabla 2).

Sin embargo, podemos llenar cada una de estas plumas con los colores de tinta que deseemos, eso sí, limitadas a 4, como es propio del modo 1.

De hecho, incluso todas las plumas pueden producir texto con el mismo color.

De momento, vamos a hacer que PEN 2 escriba con verde brillante. El método es muy simple: como el número de la tinta verde brillante es 18, basta escribir:

INK 2,18

«llena el tintero 2 con verde brillante», en donde:

INK es la abreviatura de «llena un tintero con una nueva tinta».

2 es el número de tintero.

18 es el nuevo color en el que a partir de ahora se «mojará» la pluma número 2.



Hagamos un RESET del ordenador y tecleamos:

INK 2,18

Aparentemente no ha sucedido nada en absoluto. Pero al teclear:

PEN 2

el mensaje READY aparece de inmediato en verde brillante.

Así que realmente hemos cambiado el contenido del tintero número 2.

Cambio instantáneo de tinta

Para lo que va a seguir ahora, es conveniente escribir en la pantalla un poco de texto. Asumiendo que se ha ya hecho así, si tecleamos:

INK 2,20

en donde el 20 es el número de la tinta cyan, contenido primitivo del tintero número 2 (tabla 2), TODO EL TEXTO VERDE BRILLANTE CAMBIARA A CYAN INMEDIATAMENTE.

Resulta que INK es un comando retrospectivo, y actúa de esa forma. No hay que ser muy listo para predecir lo que ocurrirá al teclear:

INK 2,18

de nuevo.

El motivo de este sorprendente comportamiento se debe a la limitación de colores que impone cada modo de pantalla.

En efecto, si esto no funcionara así, podríamos tener los 27 a la vez, escribiendo sucesivamente:

INK 2,0 y poner texto...
INK 2,1 y poner más texto ..., etc.

La única forma que el **Amstrad** tiene de obedecer la regla de la limitación de colores es cambiar todo el texto de la pantalla al color especificado por INK, no importa cuándo hayamos escrito dicho texto.

Desde un punto de vista un poco más técnico, toda la pantalla del ordenador es refrescada cada cincuentaavo de segundo (*de no hacerlo así, el contenido de la memoria del display se perdería*), como los televisores.

Así, cada vez que el micro comienza a «redibujar» la pantalla, se pregunta a sí misma: «**¿qué tinta corresponde a la pluma 2?**», y dibuja cada punto con la información dispo-

nible en ESE MOMENTO, por lo que si hemos cambiado el color ..., hups!, todo cambia de golpe.

Si se ha ido siguiendo el artículo paso a paso, como se supone que debiera ser, a estas alturas tendremos texto verde brillante sobre fondo azul.

Cambiad a pen 3:

PEN 3

y, puesto que no hemos rellenado el tintero 3, el color de primer plano (finta) será rojo.

Tecleando:

PAPER 2: CLS

el papel TAMBIEN cambiará a verde brillante (*hemos cambiado la tinta mediante INK*).

En definitiva:

INK A,B

quiere decir «rellena la pluma A con el número de tinta B».

¿Puedes predecir el efecto de INK 3,7?

El «estado del juego» en este momento se muestra en la tabla 3, de la que se deduce que estamos escribiendo con la PLUMA 3 SOBRE EL PAPEL 2.

Obsérvese lo que ocurre si cambiamos de modo:

MODE 0

Todavía estamos escribiendo en Pen 3, que nos da letras púrpura, sobre fondo verde (papel 2), pero el resto de la pantalla es azul.

¿Qué sucede al cambiar de modo?

Cuando cambiamos de modo, la pantalla es puesta automáticamente en PAPER 0, y no hemos reasignado PEN 0, la cual es azul.

Si queremos cambiar esto, por ejemplo a rosa, basta con decir:

INK 0,16

porque en modo 0 disponemos de 16 colores.

En general, el Amstrad trata de respetar en lo posible el «status» en lo referente a los colores al cambiar de modo, aunque algunas veces no es posible. En el modo 2, como sólo hay 2 colores, las plumas 2 y 3 no trabajarían, sólo las plumas 1 y 2.

Si estas últimas hubieran sido modificadas en otro modo, se mantendrían invariables en el 2.

El programa 1 muestra el efecto de asignar sucesivamente todas las tintas posibles a PEN 1.

La línea 40 establece PEN 1. Entonces usamos un bucle FOR...NEXT para «recorrer» todo el espectro de colores (líneas 50-100).

La línea 60 realiza el trabajo de «llenado» de tinteros y la 90 espera a que pulsemos una tecla para continuar.

El programa 2 es un poco más complejo. Ahora usamos las plumas 1, 2 y 3.

Las líneas 50-80 forman un bucle FOR...NEXT que asigna números diferentes, pero consecutivos, de tinta a cada pluma (línea 60).

La línea 100 espera a la pulsación de una tecla, entonces la 110 incrementa la variable «color».

Este bucle está incluido en otro sin fin, de tipo WHILE...WEND.

PROGRAMAS

```
10 REM PROGRAMA 1
20 MODE 1
30 LOCATE 8,12
40 PEN 1
50 FOR color=0 TO 27
60 INK 1,color
70 LOCATE 10,12
80 PRINT "Esto es en PEN 1"
90 WHILE INKEY="" :WEND
100 NEXT color
```

```
10 REM programa 2
20 MODE 1
30 color=0
40 WHILE -1
50 FOR biro=1 TO 3
60 INK biro,(color+biro) MOD 27
70 PEN biro
80 PRINT "Esto es PEN ";biro
90 NEXT biro
100 WHILE INKEY="" :WEND
110 color=color+1
120 WEND
```

```
10 REM programa 3
20 MODE 0
30 FOR biro=1 TO 7
40 INK biro,1
50 PEN biro
60 LOCATE(3*biro-2),12
70 PRINT CHR$(249)
80 NEXT biro
90 WHILE -1
100 FOR hombre=1 TO 7
110 INK hombre,24
120 FOR retardo=1 TO 50: NEXT
130 INK hombre,1
140 NEXT
150 WEND
```

Por último, el programa 3 muestra cómo podemos usar INK para conseguir una animación de tipo simple en MODE 0.

Una vez más, el bucle FOR...NEXT de las líneas 30-80 simplemente imprime hombreritos en sentido horizontal.

Líneas 60 y 70 fijan la posición del hombre en pantalla y lo imprimen, usando CHR\$(249).

A cada hombre se le dibuja en el mismo color que el del fondo, por lo que no se le distingue. Más exactamente, se usa una pluma diferente para cada uno.

Para hacer esto, la línea 40 fija la tinta de la pluma en uso a la tinta de fondo (1), esto es, azul brillante.

La línea 50 se asegura de que escribimos a nuestro hombre con dicha pluma.

Para conseguir la animación, basta con cambiar el color de los hombreritos sucesivamente a amarillo por un breve tiempo, luego a azul otra vez, y así sucesivamente (FOR...NEXT de las líneas 100-140).

Las líneas 110 y 130, hacen que la tinta de la pluma sea amarilla y azul,

mientras que en la 120 esperamos un poco antes de cambiar de color.

Sería interesante experimentar cambiando los colores de todas las formas posibles, para ver qué sucede. Ejecúte todo lo que se le ocurra, y si por casualidad se topa con una combinación de colores especialmente desagradable y quiere volver a las condiciones iniciales, antes de jugar con ellos asigne a la tecla ENTER del bloque numérico la siguiente cadena de caracteres:

KEY 139, «CALL &BC0:PEN 1»+CHR\$(13)

Simplemente púlsela cuando desee obtener una pantalla legible.

Tabla 1

Colores de tinta

N.º de tinta	Color
0	Negro
1	Azul
2	Azul brillante
3	Rojo
4	Magenta
5	Malva
6	Rojo brillante
7	Púrpura
8	Magenta brillante
9	Verde
10	Cyan
11	Azul cielo
12	Amarillo
13	Blanco
14	Azul pastel
15	Naranja
16	Rosa
17	Magenta pastel
18	Verde brillante
19	Verde mar
20	Cyan brillante
21	Verde lima
22	Verde pastel
23	Cyan pastel
24	Amarillo brillante
25	Amarillo pastel
26	Blanco brillante

Tabla 2

Colores por defecto en modo 1

N.º de pluma	N.º de tinta	Color
0	1	Azul brillante
1	24	Amarillo brillante
2	20	Cyan brillante
3	6	Rojo brillante

Tabla 3

Colores por defecto

N.º de pluma	N.º de tinta	Color
0	1	Azul brillante
1	24	Amarillo brillante
2	18	Verde brillante
3	7	Púrpura

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

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

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

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

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

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

COMBAT LYNX simula una misión de defensa de unas bases atacadas por una división acorazada. Disponemos para enfrentarnos a ellos de un modernísimo helicóptero.

Este juego podría incluirse dentro del catálogo de los de estrategia, y su complejidad le dota de una gran dosis de adicción y belleza.

DALEY THOMPSON'S DECATHLON con este juego OCEAN enciende la llama olímpica y te reta a superar los récords de los campeonatos de todos los tiempos, el decathlon se desarrolla en dos días de competición y se compone de las siguientes pruebas:

PRIMER DIA: 100 m lisos, salto de longitud, lanzamiento de peso, salto de altura y 400 m lisos.

SEGUNDO DIA: 110 m vallas, lanzamiento de disco, salto con pértiga, lanzamiento de jabalina y los 1.500 m.

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



COMPATIBLE
CPC 464
CPC 664



COMPATIBLE
CPC 464
CPC 664



SOLAMENTE
CPC 464

verse por la ciudad e intentar amasar la mayor cantidad de dinero posible, con el cual podrá obtener informaciones, comprar adeptos, mapas de la ciudad e incluso sobornar al carcelero que custodia a LOEG.

La ciudad está repleta de gentes extrañas, los personajes con los que nos cruzaremos son:

SKAR: La bruja que ha raptado a LOEG tiene el poder de ser invisible.

TETH: El guardián del castillo, quien está cautivo por las ratas de la ciudad.

BREN: El artesano que confecciona mapas, exiliado de Aquitania.

DAIN: El místico.

MHOR: La acompañante de DAIN, experta en libros de leyendas.

RYDE: El piloto galaico que añora su lejana patria y a la que no puede regresar por la pérdida de las cartas de navegación.

PITA: La cortesana, ansiosa de joyas y riqueza, que tiene el defecto de ser una gran cotilla.

KARA: Una carterista de gran talento que junto con su hermana fue expulsada de Cantabria.

KELI: La hermana de KARA, otra digna representante de su profesión.

KAHN: Ladrón y carterista, conocido por su insaciable codicia.

Estos son los distintos tipos que pululan por las calles de DUN DARACH, utilizamos en nuestro beneficio y comprar sus favores con dinero es pieza clave en este juego.

El programa está ambientado en una ciudad celta, con personajes muy bien caracterizados; los hombres con largas melenas y botas, y las mujeres con faldas cortas y una belleza fuera de lo común en este tipo de juegos.

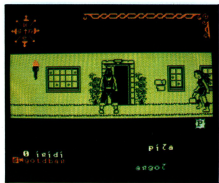
Cabe destacar el maravilloso movimiento con el que se ha dotado a los personajes, es fascinante observar cómo camina nuestro protagonista con sus melenas acompañando el ritmo de sus pasos en una conjunción muy realista y bella.

La ciudad está dividida en siete barrios, por los cuales debemos circular, existe hasta un barrio del placer, con casinos y salas de juego, un barrio de los banqueros, el del mercado, etc., completando el escenario de una auténtica ciudad celta.

CUCHULAIN puede robar, comerciar, cambiar objetos por información, vender mercancías y otras tareas que se descubren a lo largo del juego.

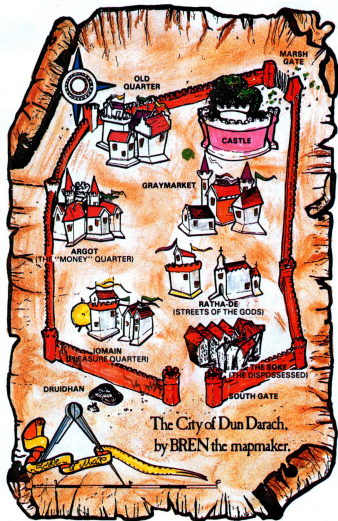
En la construcción del programa,

Mr. Joystick



se debe destacar el hecho de que disponemos de cuatro ángulos distintos de vista, para movernos por las calles y circular por las encrucijadas, de esta forma divisamos lo que tenemos en frente, detrás, a la derecha y a la izquierda.

GARGOYLE GAMES ha realizado una aventura para aquellos amantes de los juegos de larga duración, en los que es importante el uso de mapas confeccionados por los propios jugadores y con la innovación de que hemos de utilizar a fondo nuestro sentido comercial y especulativo.



3D VOICE CHESS

COMPATIBLE
CPC 464
CPC 664

Los programas de ajedrez siempre han despertado la curiosidad y la admiración de los aficionados a la informática y desde luego de los torafos de las 64 casillas, porque parece magia que una máquina, no sólo juegue, sino que encima tenga el descaro de ganarnos en buena lid.

El programa 3D Voice Chess, de DEEP THOUGHT (pensamiento profundo, nada menos) Software, nos lo va a poner difícil de veras, porque juega bien.

Los diseñadores del programa han tenido el detalle de presentar el tablero y las piezas en perspectiva tridimensional, y la originalidad de permitir que las jugadas sean cantadas de viva voz por el programa. Sí, sí, además de ganarte, habla.

Tal vez a los aficionados españoles no les sirva de mucho esta facilidad, porque como era de esperar, Voice Chess habla inglés sólo, al menos en la versión que ha caído en nuestras manos.

Su dicción en Times Square sería muy apreciada, sin duda, pero en Madrid difícilmente se le entiende una palabra.

De todas formas, esto es lo menos importante. Lo que interesa es su nivel de juego.

Hablando de niveles, tiene 7, que van desde el nivel «principiante» hasta el «correspondencia», pasando por el de «torneo».



Como suele suceder, se diferencian por la cantidad de tiempo medio por jugada que emplea el programa para calcular su respuesta, y que oscila desde 8 segundos en principiantes hasta 24 horas en correspondencia. Torneo es más aceptable: 4 minutos. En este último ya hay que andarse con cuidado, aunque sea un aficionado de mediana fuerza.

El programa permite también las opciones usuales:

— **Elegir color de piezas y nivel de juego.**

— **Jugada recomendada por la máquina.**

— **La máquina juega contra sí misma.**

— **Podemos cambiar la orientación del tablero.**

— **Elección de colores.**

— **Tecla para abandonar (nosotros, claro).**

— **Analizar una posición determinada.**

Y un largo etcétera que afortunadamente no hay que recordar porque para eso está el comando Help, el cual lista por pantalla absolutamente todos los comandos y opciones, explicando su función con detalle. Permisitos nos felicitar a los autores por esa buena idea.

En fin, en excelente programa por presentación, originalidad y nivel de juego. Nos parece increíble que los programadores hayan podido meter una pantalla de 16 Kbytes (el *table-ro*), un programa que juega bien al ajedrez y un sintetizador de voz. No cabe duda que saben lo que se hacen. Os vais a divertir.



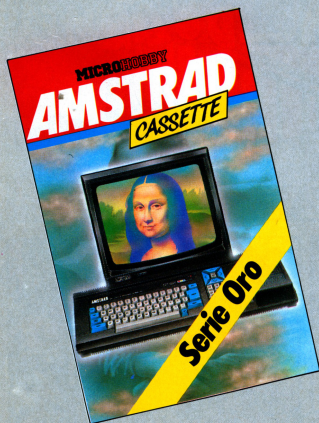
MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

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



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

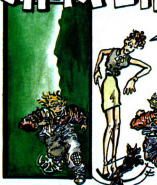
Programas incluidos en la cinta número 1

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

Envianos con la menor demora posible, el cupón correspondiente.



CHARLIE NUNCA DUERME!!

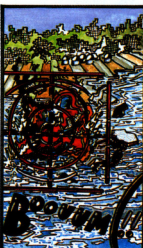
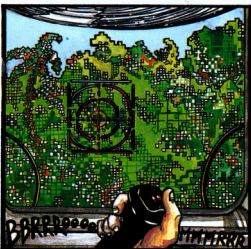




K.
4 CLEAR 70
3 INK 4
5 GO SUB 7000
BORDER 1 CLS:
BORDER 0
20 =***
¡¡AJÁ!
¡EMPEZAMOS!



ENEMIGO =
Mietiongs es
selya.
MISSION =
atal aniqui
"charlie" pel
en extremø
ATTACK =
guerrillas
evio avisø



J. J. Mora

AMSTRAD MUSICAL

Serie oro trae, en esta ocasión, a sus páginas un interesante programa para realizar música con el Amstrad, con él podemos simular el sonido de tres instrumentos: órgano, piano y oboe.

Asi podríamos decir que esta realización transforma el Amstrad en una orquesta sinfónica dirigida por nosotros mismos.

Además de tener la posibilidad de utilizar tres instrumentos, nuestro ordenador, como cualquier órgano serio, puede manejar cuatro octavas diferentes con lo que obtenemos una gran amplitud de notas musicales para nuestras composiciones.

Pero no solamente nuestro programa se limita a controlar la octava, la nota y el instrumento a utilizar, también podemos controlar el tiempo de repetición de las notas que ejecutamos.

Por si esto todavía parece poco para un órgano musical, tampoco hemos olvidado la posibilidad de ejecutar una melodía y luego volver a escucharla e incluso para aquéllos que deseen grabar su concierto en cinta o disco, este programa tiene su opción y por supuesto, otra que nos permite cargar la melodía almacenada en cinta o disco al programa.

Los amantes del sonido a gran volumen, pueden obtener una amplificación de la música generado por nuestro órgano, conectando a la salida STEREO, situada en la parte posterior de nuestro ordenador. Por medio del cual podemos sacar el sonido a través de una cadena de música, tocadiscos, cassette y cualquier otro artilugio creado para reproducir música.

INSTRUMENTO** Al seleccionar el instrumento, el programa fija automáticamente la duración del sonido idónea para dicho instrumento.

OCTAVA** La elección de la octava apropiada es muy importante a la hora de que nuestro órgano simule el sonido del instrumento en la realidad. Tonos muy agudos o demasiado graves, deforman el sonido ha-

ciendo que éste resulte un poco raro al oído.

DURACION** Fija el tiempo de repetición del sonido al pulsar una tecla. Utilizando tiempos de corta duración: 1, 3, 5, etc., conseguiremos interesantes efectos de vibrato fijando el instrumento en órgano u oboe.

MEMORIA RESET** Limpia de la memoria las notas almacenadas al ejecutar cualquier serie de notas. Todo lo que toquemos será almacenado en memoria, hasta una extensión de 300 notas; por encima de este número, la memoria de almacenamiento se desactiva automáticamente.

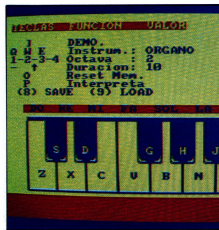
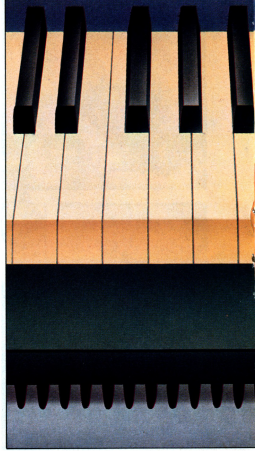
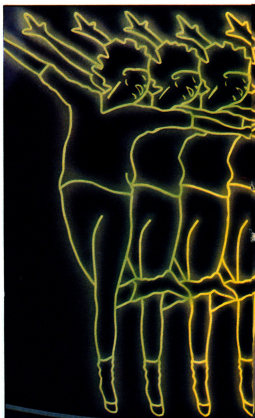
INTERPRETA** Ejecuta las notas almacenadas en la memoria, permitiéndonos escuchar lo que hemos tocado anteriormente; constituye un sistema inmejorable para mejorar nuestra técnica interpretativa. Antes de comenzar una nueva melodía es recomendable hacer un MEMORIA RESET.

SAVE** Permite grabar en cinta o disco la melodía almacenada en memoria.

LOAD** Carga de cinta o disco en nuestro programa, las notas grabadas en el soporte elegido.

COMPATIBLE
CPC 464
CPC 664

M. Barco





ESTRUCTURA DEL PROGRAMA

Con el listado a la vista podemos hacer la siguiente división en bloques:

BLOQUE 1

Líneas: 10-500.

Presentación del teclado en la pantalla (baja resolución).

Incluye en la línea 90 la definición de una ventana (#1) por la que se visualizarán los mensajes la hora de salvar en cinta o cargar una melodía.

BLOQUE 2

Líneas: 510-630.

Incluye, además de la definición de variables, la definición de 3 teclas de función para facilitar el tratamiento del programa interrumpido:

Tecla 0 = Borrado de pantalla y listado.

Tecla 1 = Ejecución automática (RUN).

Tecla 3 = Normalización de la velocidad de respuesta del teclado.

En la línea 590 podemos encontrar la definición del envolvente de PIANO y en la 620 se establece la velocidad inicial del teclado.

BLOQUE 3

Líneas: 640-680.

Presentación en pantalla del «Panel de control»: teclas, su función y valor actual.

BLOQUE 4

Líneas: 690-740.

Chequeo del teclado.

Determinación de la nota o salto a la opción pulsada.

BLOQUE 5

Líneas: 750-810.

Interpretación de la nota pulsada por los tres canales. Incluye:

Fórmula para hallar la frecuencia. (Apéndice VII del Manual.).

Comprobación de la variable MON para determinar si la nota pulsada debe ser memorizada para su posterior interpretación.

Hasta 300 notas pueden ser archivadas (DIM A(300)), pasadas éstas se desactivará MON.

BLOQUE 6

Líneas: 830-960.

Comprobación de la Función/Opción pulsada.

BLOQUE 7

Líneas: a partir de la 980.

Incluye las siguientes rutinas de Opciones:

1. Interpretación de la melodía memorizada.
2. Almacenamiento de la melodía en Cassette.
3. Carga de la melodía desde Cassette.
4. Rutina de Demostración. Las notas a interpretar se encuentran en líneas DATA's (1.400, 1.410 y 1.420).

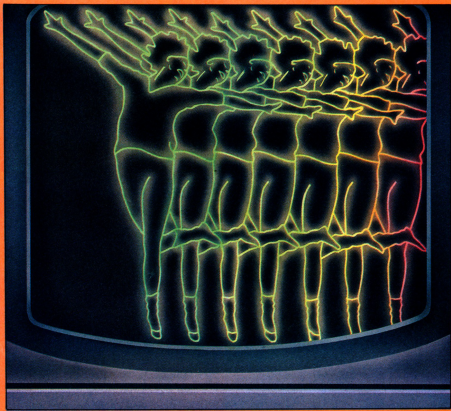
Observando el listado del programa «TECLADO 3» de una forma más o menos detenida podemos llegar a la conclusión de que podríamos haber ahorrado un buen número de líneas correspondientes a diversas rutinas que se repiten varias veces de forma prácticamente igual, especialmente las rutinas de creación del sonido (SOUND'S). Sin embargo, esta arquitectura del programa aparentemente repetitiva no es producto de la improvisación o del capricho sino que, por el contrario, se debe a limi-

taciones de Hardware. Efectivamente, el tiempo que la máquina emplea en realizar «saltos» o «llamadas» a subrutinas afectaría a la hora de producir los sonidos uniformemente como hemos podido comprobar en diferentes versiones previas a la definitiva. Es por ello que hemos suprimido llamadas a subrutinas únicas, viéndonos obligados, de este modo, a incrementar el número de líneas del listado en beneficio de un mejor funcionamiento del programa.

```

10 MODE 1
20 REM TECLADO-3
30 REM Juande Portalo (Microlingua)
40 REM *****
50 REM DIBUJAR T E C L A D O
60 PAPER 2:PEN 0
70 INK 0,1:INK 2,24
80 CLS
90 WINDOW #1,1,40,23,25
100 PAPER #1,3:PEN #1,2
110 CLS #1
120 REM DEFINICION DE BLOQUES Y SIM
    BOLOS 0:GRAFICOS
130 C147#=CHR$(147)
140 C153#=CHR$(153)
150 C155#=CHR$(155)
160 C158#=CHR$(158)
170 C149#=CHR$(149)
180 C154#=CHR$(154)
190 C1#=#C154#*C154#*C154#*C154#*C155#
200 C2#=#C154#*C154#*C154#*C153#
210 SYMBOL 254,&X10111111,&X1011111
    1,&X10111111,&X10011111,&X10011111,
    &X11000000,&X10111111,&X11111111
220 SYMBOL 255,&X11111101,&X11111101
    1,&X11111101,&X11111101,&X11110001,
    &X11,&X11111101,&X11111110
230 "
240 LOCATE 4,22:PRINT C147#;FOR N#
    1 TO 27 STEP 4:PRINT C1#;NEXT N
250 PRINT C2#
260 LOCATE 4,13
270 PRINT CHR$(136);:FOR N#1 TO 31:
    PRINT CHR$(148);:NEXT N:PRINT CHR$(
    132)
280 FOR M#14 TO 21
290 FOR N#4 TO 38 STEP 4:LOCATE N,M
    :PRINT C149#;NEXT N
300 NEXT M
310 REM TECLAS #
320 X=#7:GOSUB #30
330 X=#11:GOSUB #30
340 X=#23:GOSUB #30
350 X=#19:GOSUB #30
360 X=#27:GOSUB #30
370 GOTO 390
380 FOR N#14 TO 17:LOCATE X,N:PRINT
    CHR$(143);CHR$(143);CHR$(143);NEXT
    N:LOCATE X,18:PRINT CHR$(254);CHR$(
    143);CHR$(255):RETURN
390 PEN 2:PAPER 0:FOR N#1 TO 3:READ
    A#,A
400 LOCATE A,17:PRINT A#
410 NEXT N
420 PEN 0:PAPER 2:FOR N#6 TO 36 STE
    P 4
430 READ A#
440 LOCATE N,20:PRINT A#
450 NEXT N
460 PAPER 3:LOCATE 4,12:PRINT" DO
    RE MI FA SOL LA SI DO":PAPE
    R 2
470 DATA S,0,0,12,6,20,H,24,J,28
480 DATA Z,X,C,V,0,N,H,C
490 REM
500 REM
510 REM
520 REM
530 DIM A(380):Q=1
540 KEY 3:"SPEED KEY 28,3"+CHR$(13)

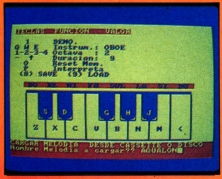
```



```

:KEY 0,"CLS:LIST "+CHR$(13):KEY 1,"
550 P(1,1)=1:P(1,2)=3:P(1,3)=6
560 P(2,1)=1:P(2,2)=3:P(2,3)=6
570 P(3,1)=0:P(3,2)=0:P(3,3)=0
580 EV(1)=15:EV(2)=1:EV(3)=15
590 ENV 1,1,0,1,1,0,10,14,-12 ENV
    OLVENTE DE VOLUMEN (PIANO)
600 LET DU=10:1=1
610 I$(1)="ORGANO " :I$(2)="PIANO
    " :I$(3)="OBOE "
620 SPEED KEY 1.1
630 T$="MUNHBBVCXXS212346M0P-89_
    "
640 LOCATE 2,2:PAPER 3:PEN 2:PRINT"
    TECLAS FUNCION VALOR":PAPER 2:PE
    N 0
650 LOCATE 4,4:PRINT" DEMO."
660 LOCATE 2,5:PRINT" O W E Instru
    m.":I$(1):LOCATE 2,6:PRINT"1-2-3
    4 Octava " :I$(2):LOCATE 2,7:PRINT"
    " :Duracion":I$(3)
670 LOCATE 4,8:PRINT" O Reset Me
    m.":LOCATE 4,9:PRINT" P Interpre
    ta":LOCATE 3,10:PRINT"(B) SAVE (9)
    LOAD"
680 "
690 A#=#INKEY$:IF A#="" THEN GOTO 690
700 A#=#UPPER$(A#)
710 A#=#INSTR$(A#,A#)
720 IF A#="" THEN GOTO 690
730 IF A#13 THEN GOTO 830
740 A#=#1
750 FR=#48#(2-(OCTAVA#+(10-A#)/12))
760 PER=#ROUND(125000/FR)
770 IF MON#1 THEN Q=#1:IF Q>300 TH
    EN MON#0 ELSE LET A#=#PE
780 "
790 SOUND 49,PE+(1/(OCTAVA#4))*P(1
    ,1),DU,7,EV(1)
800 SOUND 42,PE+(1/(OCTAVA#4))*P(1
    ,2),DU,7,EV(1)
810 SOUND 28,PE+(1/(OCTAVA#4))*P(1
    ,3),DU,7,EV(1)
820 GOTO 690
830 IF A#21 THEN MON#1:Q=1:LOCATE 2
    8,8:PRINT CHR$(233):PRINT CHR$(7):L
    OCATE 28,8:PRINT "
840 IF A#22 THEN GOTO 980
850 IF A#23 AND DU1 THEN DU=#DU-0.3
    :LOCATE 28,7:PRINT INT(DU):"
860 IF A#24 AND DU<100 THEN DU=#DU+0
    .3:LOCATE 28,7:PRINT INT(DU):"
870 IF A#14 THEN OCTAVA#=#1:LOCATE 2
    0,6:PRINT OCTAVA#2:"
880 IF A#15 THEN OCTAVA#=#1:LOCATE 20
    ,6:PRINT OCTAVA#2:"
890 IF A#16 THEN OCTAVA#=#1:LOCATE 20
    ,5:PRINT OCTAVA#2:"
900 IF A#17 THEN OCTAVA#=#2:LOCATE 20
    ,6:PRINT OCTAVA#2:"
910 IF A#18 THEN I=#1:DU=#1:LOCATE 2
    0,5:PRINT I$(1):LOCATE 28,7:PRINT D
    U:" :SPEED KEY 1.1
920 IF A#19 THEN I=#2:DU=#2:LOCATE 2
    0,5:PRINT I$(1):LOCATE 28,7:PRINT D
    U:" :SPEED KEY 255,3
930 IF A#20 THEN I=#3:DU=#3:LOCATE 20
    ,5:PRINT I$(1):LOCATE 28,7:PRINT D
    U:" :SPEED KEY 1.1
940 IF A#25 THEN GOTO 1060
950 IF A#26 THEN GOTO 1220
960 IF A#27 THEN GOTO 1370
970 GOTO 690
980 REM TODA LD MEMORIZADO
990 FOR N#1 TO 0
1000 SOUND 49,A(N)+1/(OCTAVA#4)*P
    (1,1),DU,7,EV(1)
1010 SOUND 42,A(N)+1/(OCTAVA#4)*P
    (1,2),DU,7,EV(1)
1020 SOUND 28,A(N)+1/(OCTAVA#4)*P
    (1,3),DU,7,EV(1)
1030 A#=#INKEY$:IF A#(0) THEN 1030
1040 NEXT N
1050 GOTO 690
1060 REM SALVAR FICHERO MELODIA
1070 SPEED KEY 20,3
1080 CLS #1:PRINT #1,"SALVAR MELODIA
    EN CASSETTE O DISCO"
1090 INPUT #1,"Nombre Melodia?":ME
    LOD$
1100 IF MELOD#="" OR MELOD#="8" THE
    N PRINT #1," A B O R T A D
    O":GOTO 1200
1110 LOCATE 1,23
1120 OPENOUT MELOD#.
1130 PRINT #5,MELOD#

```



```

1148 PRINT #9,Q
1150 FOR N=1 TO 0
1160 PRINT #9,A(N)
1170 NEXT N
1180 CLOSEOUT
1190 CLS #1:PRINT #1," O.K. S A L
V A D 0"
1200 FOR RE=1 TO 800:NEXT RE:CLS #1
:PRINT CHR$(7)
1210 A=18:GOTO 910
1220 'CARGA FICHERO DESDE CASSETTE
1230 SPEED KEY 20,3
1240 CLS #1:PRINT #1,"CARGAR MELODIA
DESDE CASSETTE O DISCO"
1250 INPUT #1,"Nombre Melodia a car
gar?":MELOD#
1260 LOCATE 1,23
1270 OPENIN MELOD#
1280 INPUT #9,MELOD#
1290 INPUT #9,Q
1300 FOR N=1 TO 0
1310 INPUT #9,A(N)
1320 NEXT N
1330 CLOSEIN
1340 CLS #1:PRINT #1," O.K. C A R
O A D 0 " :PRINT CHR$(7)
1350 FOR RE=1 TO 800:NEXT RE:CLS #1
1360 A=18:GOTO 910
1370 REM DEMO
1380 RESTORE 1400
1390 FOR L=1 TO 121:READ Z:LET A(L)
=Z:NEXT L
1400 DATA 0, 253, 239, 225, 213, 15
9, 253, 159, 253, 159, 190, 0, 15
9, 169, 159, 127, 159, 142, 127,
159, 142, 159, 0, 253, 239, 225,
213, 159, 253, 159, 253, 159, 190,
0, 159, 169, 179, 190, 169, 159,
142, 127, 159, 127, 142, 142, 142,
0, 253
1410 DATA 239, 225, 213, 159, 253,
159, 253, 159, 190, 0, 159, 169, 15
9, 127, 159, 142, 127, 159, 142,
159, 319, 0, 253, 159, 319, 284,
253, 179, 319, 284, 253, 190, 319,
284, 253, 281, 319, 284, 253, 213,
319, 284, 253, 0, 159, 142, 127,
89, 159
1420 DATA 142, 127, 95, 159, 142, 1
27, 180, 159, 142, 127, 186, 159,
142, 127, 159, 142, 127, 159, 142,
159, 159, 0, 0
1430 DU=21:FOR I=1 TO 3
1440 CLS #1:PRINT #1," DEMOSTRACION
N: ":PRINT #1," ":I*(I)
1450 IF I=3 THEN DU=15
1460 IF I=2 THEN DU=28
1470 FOR N=1 TO 121
1480 SOUND 49,A(N)*(1/(OCTAVA+4))*P
(1,1),DU,7,EV(I)
1490 SOUND 42,A(N)*(1/(OCTAVA+4))*P
(1,2),DU,7,EV(I)
1500 SOUND 28,A(N)*(1/(OCTAVA+4))*P
(1,3),DU,7,EV(I)
1510 NEXT N
1520 NEXT I
1530 AS=INKEY$:IF AS("<") THEN 1530
1540 CLS #1:A=18:GOTO 910

```

VARIABLES		
LISTA DE VARIABLES		
LINEA	NOMBRE	COMENTARIO
530	A(300)	Dimensión para almacenamiento de las notas que han de ser interpretadas con la opción de memorización.
530	Q	Servirá de índice a A(Q) durante el almacenamiento de la melodía.
550-570	P(1,1) P(1,2)... etc.	Indicador de la «distancia» entre las notas que han de sonar simultáneamente. (Efecto de armónicos en órgano y piano.)
580	EV(1) EV(2) EV(3)	Índice del envolvente.
600	DU	Duración de la nota.
600	I	Índice de instrumento.
610	IS(1),2,3	Nombre del instrumento.
630	TS	Cadena de teclas y opciones.
690	A\$	Tecla pulsada (INKEY\$).
710	A	Valor numérico de A\$ que indica la posición de A\$ en TS mediante INSTR y determina el número de opción elegida.
750	FR, PE	Forman parte de la Fórmula para hallar la frecuencia de la nota (Apéndice VII del Manual).
770	MON	Indicador de memorización activada.
870	OCTAVA	Número de la Octava (1=-1, 2=0, 3=1, 4=2).
1.090-1.250	MELODS	Variable correspondiente al nombre de la melodía a salvar en cinta o cargar: SAVE y LOAD.

PANEL DE CONTROL		
TECLAS	FUNCION	COMENTARIO
]	DEMOSTRACION	Interpreta una melodía almacenada originalmente en DATA's.
Q, W, E	INSTRUMENTO	Q=Órgano; W=Piano; E=Oboe.
1, 2, 3, 4	OCTAVA	Elección de la Octava.
↑	DURACION	'-'=Menos '↑'=Más.
O	MEMORIA RESET	Comienza a almacenar las notas destruyendo la melodía anterior si la hubiera.
P	INTERPRETA	Interpreta la melodía memorizada hasta el momento.
8	SAVE	Salva la melodía memorizada en cinta o disco.
9	LOAD	Carga una melodía desde cassette o disco.
TECLADO MUSICAL		
NOTAS		
Z=DO, =DO#, X=RE, D=RE#, C=MI, V=FA, G=FA#, B=SOL, H=SOL#, N=LA, J=LA#, M=SI, ' (coma)=DO (una octava más alta).		
TECLA ESPACIADORA=SILENCIO		



TRON

Como todos los días, amanece en la ciudad. Nada parece capaz de romper el silencio casi religioso de la mañana. Sin embargo, en el aire aletea una tensa emoción proveniente de la cueva de los Tronners. Hoy es el día de los días, el juego va a empezar.

H

an transcurrido 100 años desde la Tercera Guerra Mundial. Los supervivientes de la catástrofe juraron que jamás sucedería en el planeta nada parecido y, como eran tan pocos, lograron ponerse de acuerdo: a partir de ahora, las diferencias entre naciones se dirimirán entre dos hombres especialmente entrenados para ello.

Imaginaron un juego, mortífero y espectacular, en el que cada campeón conduciría un campo de fuerza láser a través de un laberinto holográfico y cambiante. Aquél que obligara a su adversario a chocar con su estela o la acorralara, ganaría el juego y la «guerra», al precio de una sola vida.

Así nacieron los Tronners, dedicados desde su nacimiento a ser los últimos guerreros de un mundo agonizante, incapaz de sobrevivir sin la violencia, a pesar de la amarga lección de la historia.

VARIABLES PRINCIPALES

NOMBRE SIGNIFICADO

ti	Combustible que nos queda.
y1,y2	Coordenadas del jugador azul.
x1,x2	Coordenadas del jugador rojo.
y3,y4	Dirección del movimiento del jugador azul.
x3,x4	Dirección del movimiento del jugador rojo.
coly	Averigua si ha habido choque del azul.
colx	Averigua si el rojo ha chocado.

COMPATIBLE
CPC 464
CPC 664

Así vio la luz también el juego de los juegos: Tron.

El juego, en realidad, es muy simple. Los dos jugadores, cada uno dentro de su campo de fuerza, se mueven a través del área de juego, dejando a su paso una estela mortal. Si cualquiera de ellos choca con ella, deja de existir en el acto.

Ambos tienen a su disposición la misma cantidad de combustible y se mueven a la misma velocidad.

De los tres laberintos posibles, el más arriesgado y entretenido es, sin duda, el aleatorio: hace falta gran habilidad para sobrevivir.

Finalmente, una buena estrategia para ganar es no permanecer en la misma zona de la pantalla mucho tiempo, para no dar oportunidad al adversario de que nos rodee, destruyéndonos.

ESTRUCTURA DEL PROGRAMA

NUMERO DE LINEA	FUNCION
60	Rutina principal. Lee el teclado, chequea colisiones y reduce el combustible adecuadamente.
270	Muerte de ambos jugadores.
310	Muerte del jugador azul.
360	Muerte del jugador rojo.
410	Rutina de fin del juego. Espera hasta que se pulsa la barra espaciadora y retorna a la pantalla de presentación, donde se elige el tipo de laberinto en el que transcurre el juego.
490	Rutina de elección de pantalla.
630	Almacena variables muy importantes: combustible y posiciones iniciales de los jugadores al principio del juego.
710	Instrucciones.
940	Inicializa laberinto 1.
980	Inicializa laberinto 2.
1050	Laberinto 3.

```

18 REM*****
20 REM***** T R O N *****
30 REM*****By A.Chapman*****
40 LET scrd=0:LET scblue=0
50 GOSUB 710:REM instrucciones
60 REM ** Rutina Principal **
70 PLOT y2,y1,3:PL0T x2,x1,2
80 SOUND 1,1000,5,15,0,1:SOUND 2,
3400,5,15,0,0,1
90 IF INKEY(53)=0 THEN LET x3=1:LET
x4=0
100 IF INKEY(62)=0 THEN LET x3=-1:
ET x4=0
110 IF INKEY(63)=0 THEN LET x4=1:LE
T x3=0
120 IF INKEY(71)=0 THEN LET x4=-1:
ET x3=0
130 IF INKEY(4)=0 THEN LET y3=1:LET
y4=0
140 IF INKEY(5)=0 THEN LET y3=-1:LE
T y4=0
150 IF INKEY(14)=0 THEN LET y4=1:LE
T y3=0
160 IF INKEY(13)=0 THEN LET y4=-1:LE
ET y3=0
170 LET x1=x1+x3*2:LET y1=y1+y3*2:LE
ET x2=x2+x4*4:LET y2=y2+y4*4
180 LET ti=ti-1:IF ti=0 THEN GOTO 2
70
190 LOCATE 16,1:PEN 12:PRINT ti
200 LET colx=TEST(x2,x1)
210 LET coly=TEST(y2,y1)
220 IF colx=4 OR coly=4 THEN GOTO 2
40 ELSE IF colx<0 AND coly<0 THEN
GOTO 270
230 IF colx<0 THEN GOTO 310 ELSE I
F coly<0 THEN 360
240 GOTO 70
270 REM ** CHOQUE DE LOS DOS Y GAST
O DE COMBUSTIBLE **
280 REM*****
285 MODE 1:PEN 7
290 IF ti<1 THEN PRINT" El combu
stible se ha agotado." :GOTO 430
295 PRINT" Los dos conductores h
an chocado."
300 GOTO 430
310 REM ** Muerte del jugador Azul **
320 REM*****
330 MODE 1:PEN 3:PRINT" B
ien conductor Rojo."
335 LET scrd=scrd+1
340 PRINT" El jugador Azul ha
muerto"
350 GOTO 430
360 REM ** Muerte del jugador Rojo
**
370 REM*****
380 MODE 1:PEN 2:PRINT" B
ien Jugador Azul."
385 LET scblue=scblue+1
390 PRINT" El jugador Rojo ha
muerto"
400 GOTO 430
410 REM*****
420 REM ** Fin de Juego **
430 REM*****
434 LOCATE 19,8:PEN 1:PRINT"PLINTOS"
:LOCATE 18,9:PEN 3:PRINT"*****"
435 LOCATE 11,11:PEN 2:PRINT"AZUL...
....."scblue
436 LOCATE 11,13:PEN 3:PRINT"ROJO..
....."scrd

```



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 cassette mensual, solicitáloslo.

```

440 LOCATE 9,15:PEN 2:PRINT"Jugamos
otra vez (S/N) ?"
441 IF INKEY("<") THEN GOTO 441
442 IF INKEY="*" THEN GOTO 442
443 IF INKEY(48)=0 THEN GOTO 450
444 IF INKEY(46)=0 THEN INK 0,1:PAP
ÉR 0:PEN 1:CLS: BORDER 1:END
445 GOTO 441
450 LOCATE 8,17:PEN 1:PRINT"Cambiam
os de campo (S/N) ?"
451 IF INKEY("<") THEN GOTO 451
452 IF INKEY="*" THEN GOTO 452
453 IF INKEY(46)=0 THEN GOTO 460
454 IF INKEY(48)=0 THEN GOSUB 498:G
OTO 460
455 GOTO 451
460 GOSUB 630
465 ON maze GOSUB 948,980,1050
470 GOTO 66
498 REM** Eleccion de pantalla **
500 INK 1,24:CLS:PEN 2:LOCATE 12,4:
PRINT"-----"
510 LOCATE 12,5:PRINT"CAMPOS DE BATALLA"
518 LOCATE 12,6:PEN 1:PRINT"-----"
-----
520 LOCATE 3,9:PRINT"Existen 3 tipo
s de campo de batalla."
530 LOCATE 14,12:PEN 1:PRINT"1.CAMP
O ABIERTO"
540 LOCATE 14,14:PEN 3:PRINT"2.LABE
RINTO DE LA MUERTE"
550 LOCATE 14,16:PEN 2:PRINT"3.LABE
RINTO AL AZAR"
560 LOCATE 18,20:PRINT"Pulse el num
ero 1,2 o 3."
570 LOCATE 18,21:PEN 3:PRINT STRING
$(24," ")
580 LET k$=INKEY$
590 IF k$="1" THEN LET maze=1:GOSUB
630:GOSUB 948:RETURN
600 IF k$="2" THEN LET maze=2:GOSUB
630:GOSUB 980:RETURN
610 IF k$="3" THEN LET maze=3:GOSUB

```

```

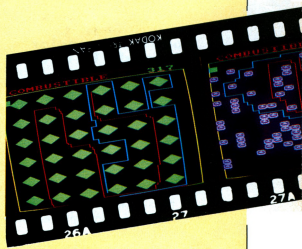
630:GOSUB 1050:RETURN
620 GOTO 580
630 REM*****VARIABLES*****
640 MODE 0
660 LET t:=999
670 LET x1=24:LET x2=319:LET y1=350
:LET y2=319
680 LET x3=1:LET x4=0:LET y3=-1:LET
y4=0
700 RETURN
710 REM ** INSTRUCCIONES **
720 MODE 1:INK 0,0:INK 1,24:INK 2,1
1:INK 3,6:CLS:INK 8,24:PEN 2:CLS:BO
RDER 3
730 PRINT:LOCATE 15,2:PEN 2:PRINT"-
-----"
740 PEN 1:LOCATE 15,3:PRINT" T R
O N "
750 LOCATE 15,4:PEN 3:PRINT"-----"
-----
760 LOCATE 12,5:PEN 3:PRINT"By Aram
ello Chapman"
765 PEN 1:PRINT:PRINT
770 PRINT" Estamos en el 3075.Es e
l centenario de los campeonatos de c
orredores TRON.Los dos mas grandes
corredores se enfrentan en busca d
el honor y la fama aunque uno de e
llos morira.!!!"
780 PRINT" Cada jugador intentara
no tropezar con la estela del otro
y a la vez cortar el paso al contra
rio."
785 PRINT" El campo de batalla esta
delimitado por un rectangulo amaril
lo electrificado."
790 PEN 1:PRINT TAB(15);"
-----"
800 LOCATE 1,22:PEN 3:PRINT STRING$(
40,"-"):LOCATE 3,23:PEN 2:PRINT"Pu
lsa (BARRA ESPACIO) para continuar"

```

```

:LOCATE 1,24:PEN 1:PRINT STRING$(40
,"-")
810 IF INKEY(47)<0 THEN GOTO 810
820 CLS:PRINT:PEN 1
840 LOCATE 16,5:PEN 3:PRINT"-----"
850 LOCATE 16,6:PEN 1:PRINT"CONTROL
ES"
845 LOCATE 1,8:PEN 3:PRINT STRING$(
40,CHR$(154))
850 LOCATE 16,7:PEN 2:PRINT"-----"
860 PEN 2:LOCATE 3,9:PRINT"JUGADOR
AZUL ";:PEN 3:PRINT TAB(26);"JUGA
DOR ROJO"
855 LOCATE 1,15:PEN 2:PRINT STRING$(
40,CHR$(154))
860 LOCATE 3,10:PEN 1:PRINT"-----"
870 PEN 2:PRINT:PRINT"1.ZODA-Z' DC
HA-"X":PRINT" ";:PEN 3:PRINT"1.ZODA
-1' DCHA-Z'"
880 PEN 2:PRINT:PRINT"SUBIR-F BA
JAR-C":PEN 3:PRINT" SUBIR-'6' B
AJAR-'3'"
895 LOCATE 1,17:PEN 1:PRINT STRING$(
40,CHR$(154)):LOCATE 1,19:PEN 3:PR
INT STRING$(40,CHR$(154))
900 LOCATE 6,18:PEN 2:PRINT"Pulsa u
na tecla para continuar"
910 IF INKEY("<") THEN GOTO 910
915 IF INKEY="*" THEN GOTO 915
920 GOSUB 498
930 RETURN
940 REM ** CAMPO ABIERTO **
950 CLS: LOCATE 2,1:PEN 3:PRINT"COM
BUSTIBLE -"
960 PEN 12:PLOT 0,382,1:DRAW 639,38
2,1:DRAW 639,0,1:DRAW 0,0,1:DRAW 0,
382,1
970 RETURN
980 REM ** LABERINTO DE LA MUERTE *
*
1000 GOSUB 940
1010 LET q$=CHR$(214)+CHR$(215)
1020 LET w$=CHR$(213)+CHR$(212)
1030 FOR r=3 TO 24 STEP 4: FOR q=2
TO 19 STEP 3:PEN 12:LOCATE q,r:PRIN
T q$:LOCATE q,r+1:PRINT w$:NEXT q:
NEXT r
1040 RETURN
1050 REM ** LABERINTO AL AZAR **
1070 GOSUB 940
1080 FOR w=1 TO 100:LET p=INT(RND*2
0)+3:LET q=INT(RND*20)+3:IF p=0 TH
EN LET p=2
1090 LOCATE p,q:PEN 7:PRINT CHR$(22
4):NEXT w
1100 RETURN

```



El programa DIGGER del n.º 5 de la revista es compatible sólo con el CPC 464, no con el CPC 664. Rogamos nos disculpen.

LOS REGISTROS DE 16 BITS

J. Igual

Vamos a introducirnos en una parte de la programación en lenguaje máquina más importante, si cabe, que las que hemos analizado hasta este momento.

Ya sabemos que el Z80 tiene una serie de registros, dijéramos de «propósito general» y, dentro de ellos, uno que además cumple unas ciertas funciones especiales ya vistas, llamado el acumulador. En efecto, analizamos operaciones de carga de registros, suma y sustracción, que sólo pueden realizarse con él.

D

ijimos también que todos los registros sólo pueden almacenar un byte a un tiempo y que, a pesar de esto, las direcciones de memoria ocupan 2 bytes, los llamados byte alto y byte bajo.

Por ejemplo, al ejecutar una instrucción de llamada a una subrutina en máquina, como CALL &3000, los códigos de operación eran CD 00 30, es decir, la dirección &3000 ocupa 2 bytes.

No estaría nada mal tener la posibilidad de manejar dos bytes a un tiempo; ello permitirá un ahorro considerable de instrucciones cuando manejáramos direcciones de memoria, por ejemplo.

Afortunadamente, los diseñadores del Z80 preveyeron esta situación y permitieron a los registros, aunque sólo sean de un byte, agruparse por parejas, con lo que resulta fácil manejar números de 16 bits de una «cada».

Podemos manejar números de 16 bits de una sola vez

Estas felices parejas son: BC, DE y HL. O sea, que podemos hacer cosas como:

```
LD HL, &AAFF
```

Después de esta instrucción de carga de 16 bits, el registro L contendrá el byte bajo de la dirección (&FF) y el H el alto (&AA). En inglés, el nombre de los registros H y L tiene bastante sentido, pues son las abreviaturas de High (alto) y Low (bajo), re-

DIRECCION	OPCODE	MNEMONICOS
3000	21 FF AA	LD HL, &AAFF
3003	7D	LD A, L
3004	32 F8 2F	LD (&2FF8), A
3007	7C	LD A, H
3008	32 F9 2F	LD (&2FF9), A
300B	C9	RET

Programa 1.

feridas a qué byte se almacena normalmente en cada uno.

El programa número 1 ilustra un poco este punto.

¿Podéis ver lo que hace?

Primero, cargamos HL con el número &AAFF, y se pretende depositarlo en las posiciones de memoria &2FF8 y &2FF9, el «espacio de trabajo» del Hexer; es decir, hablando en Basic por un momento, queremos hacer un par de POKES.

Recurrimos al acumulador, que hace de intermediario; primero, lo cargamos con L, depositamos el A en &2FF8 y repetimos el proceso con el registro H. Por último, el obligatorio RET.

Observa en el programa 1, que el código de operación de:

```
LD HL, &AAFF
```

es: 21 FF AA, es decir, primero se almacena en memoria el byte bajo y luego el alto.

Examinando el espacio de trabajo del Hexer después de ejecutar el programa, &2FF8 contendrá &FF y &2FF9 &AA.

Por tanto, podéis comprobar que es rigurosamente cierto que, al cargar un registro par, HL, BC o DE con un número de 16 bits, el byte bajo se



carga en L, C o E y el alto en H, B o C.

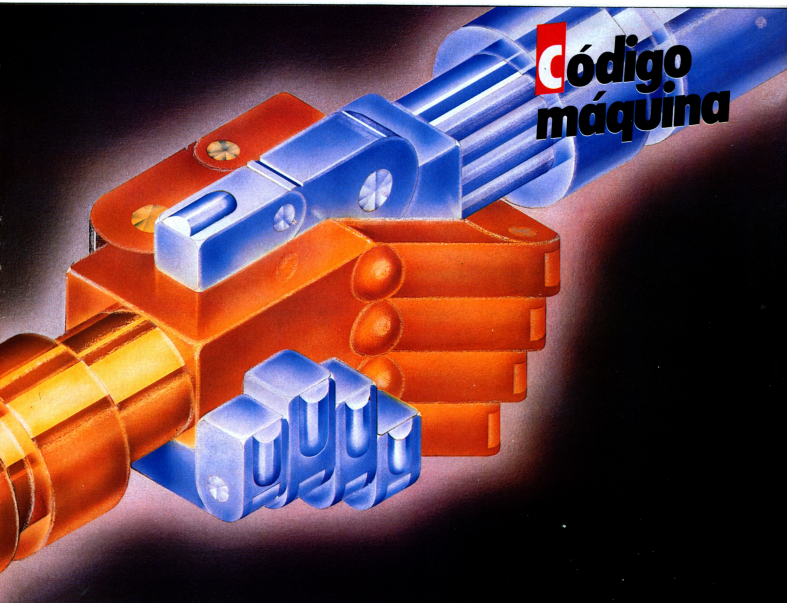
Con el firmware, casi todo es posible

El programa número 2 nos será bastante familiar: es el que pone un asterisco en una posición determinada de la pantalla, la cual debe encontrarse en los registros H y L. Este ejemplo, está hecho de forma artesanal, cargando laboriosamente primero el H y luego el L. Estas dos instrucciones ocupan 4 bytes.

DIRECCION	OPCODE	MNEMONICOS
3000	CD 6C BB	CALL CrlText
3003	26 14	LD H, &14
3005	2E OC	LD L, &OC
3007	CD 75 BB	CALL Posi_Cur
300A	3E 2A	LD A, &2A
300C	CD 5A BB	CALL PrintOut
300F	C9	RET

Programa 2.

Código máquina



DIRECCION	OPCODE	MNEMONICOS
3000	CD 6C BB	CALL CfrText
3003	21 0C 14	LD HL,&140C
3006	CD 75 BB	CALL PostCur
3009	3E 2A	LD A,&2A
300B	CD 5A BB	CALL PrintOut
300E	C9	RET

Programa 3.

El programa 3 hace lo mismo, pero de forma más eficiente, mediante la instrucción de carga de 16 bits; encima ahorramos un byte.

Ambos programas colocan el asterisco en la columna 20 (&14 en el registro H) y fila 12 (&0C en el L).

La Tabla 1 muestra los mnemónicos y opcodes de la instrucción que nos ocupa para el registro de los registros.

El programa número 4 las usa para cargar los registros DE y HL con los datos necesarios para dibujar una línea en la ventana de gráficos.

Empleamos dos rutinas de firmware como ayuda: una, que comienza en la dirección &BBF6, dibuja una línea en la pantalla gráfica basándose en las coordenadas almacenadas en el registro DE (coordenada X) y HL (coordenada Y). La segunda, ya es conocida: simplemente borra la pantalla y no necesita parámetros (dirección &BBDB).

DIRECCION	OPCODE	MNEMONICOS
3000	CD DB BB	CALL CfrGraf
3003	11 7F 02	LD DE,&27F
3006	21 8F 01	LD HL,&18F
3009	CD F6 BB	CALL GrafLine
300C	C9	RET

Programa 4.

MNEMONICOS	OPCODES
LD BC,mn	01 n m
LD DE,mn	11 n m
LD HL,mn	21 n m

Tabla 1.

Manejar gráficos desde código máquina

El punto de comienzo de la línea es, obviamente, el último «visitado» por el curso de gráficos, como en Basic. Y al haber borrado la ventana, dicho punto será (0,0).

Como se deduce del listado del programa 4, las coordenadas de destino la línea son 639 (&27F) para X y 399 (&18F) para Y, por lo que, al ejecutarlo, una línea cruzará la pantalla en diagonal de izquierda a derecha y de abajo a arriba.

Al volver al Hexer, o a Basic, si la llamada a la rutina se hizo con el comando CALL, la pantalla efectúa scroll hacia arriba de una línea. ¿Cómo te las arreglarías para evitarlo?, es decir, para que el programa 4 esperara a la pulsación de una tecla antes de retornar (la rutina del firm-

DIRECCION	OPCODE	MNEMONICOS
3000	CD DB BB	CALL CfrGraf
3003	16 02	LD D,&02
3005	1E 7F	LD E,&7F
3007	26 01	LD H,&01
3009	2E 8F	LD L,&8F
300B	CD F6 BB	CALL GrafLine
300E	C9	RET

Programa 5.

ware &BB18 puede ser una gran ayuda).

Bueno, pues ya nos hemos topado con la rutina GrafLine en el programa 4, vamos a ver si la aprovechamos para dibujar algo en la pantalla.

GrafLine corrompe los valores iniciales de los registros en los que se le suministran los parámetros que necesita. No obstante, el firmware o el sistema operativo se encargan de «guardar memoria» de donde estuvo por última vez el cursor gráfico.

Otra rutina del firmware, que comienza en la dirección &BBC0, equivale al comando gráfico del Basic MOVE. Requiere dos parámetros en los registros dobles DE (coordenada X) y HL (coordenada Y). La llamaremos GrafMove, y también corrompe los registros.

Comparando los programas 4 y 5 se puede observar dos importantes detalles:

1. El 4 ocupa 2 bytes menos de memoria debido al uso de las instrucciones de carga de 16 bits.

2. El 4 también resulta más claro e intuitivo en cuanto al concepto de pares de coordenadas que el 5.

En este caso, podríamos decir que se juntan el hambre con las ganas de comer: las instrucciones de carga de 16 bits conviene usarlas siempre que se pueda.

Claridad ante todo

De la comparación de estos 2 programas se deduce otra de las reglas de oro de la programación en máquina, la cual resulta tan obvia que todos los programadores, casi sin excepción, la olvidan a menudo, pagando más tarde amargos réditos por su despiste: HAGA LOS PROGRAMAS CLAROS. PIENSE QUE TENDRA QUE DEPURARLOS O MODIFICARLOS TARDE O TEMPRANO. Según la ley de máxima fatalidad, más bien temprano.

Echad una mirada al programa 6. Hay algunas cosas nuevas en él. Nos proponemos crear una ventana de

texto y ponerle, como color de fondo, rojo, es decir, PAPER 3 (asumiendo que estamos en MODE 1).

Para crear la ventana, recurrimos a la rutina TextWin (&BB66), la cual, ¡cómo no!, necesita recibir como datos la posición de la ventana en la pantalla. No creo que os sorprendáis mucho cuando os anuncie que los datos van en los registros DE y HL, pero esta vez, lo admito, de forma un tanto sádica.

En efecto, H y D (no H y L) deben contener las posiciones de columna de la ventana y L y E las filas. Los valores pueden verse en el listado del programa 6. ¡Ah!, otra cosa, como no hemos especificado lo contrario, nuestra ventana será la &0.

DIRECCION	OPCODES	MNEMONICOS
3000	CD 6C BB	CALL CfrText
3003	26 0A	LD H,&0A
3005	16 1E	LD D,&1E
3007	2E 0A	LD L,&0A
3009	1E 10	LD E,&10
300B	CD 66 BB	CALL TextWin
300E	3E 03	LD A,&03
3010	CD 96 BB	CALL SetPaper
3013	CD 6C BB	CALL CfrText
3016	C9	RET

Programa 6.

El asunto de escoger el color del papel es menos lio. Hay, por supuesto, otra rutina firmware, llamada SetPaper (&BB96), que se encarga del trabajo.

Sus pretensiones son mucho más humildes que las de sus compañeras, sólo pide un parámetro en el acumulador que debe contener un número válido de tinta en el modo de pantalla en el que nos encontremos. No es necesario decir que ambas rutinas corrompen todos los registros que utilizan.

Tal vez, el proceso resulte más claro si vemos cómo hacer lo mismo desde Basic:

```
10 CLS
20 WINDOW 10,30,10,16
30 PAPER 3
40 CLS
```

Más sobre los registros dobles

Hasta este momento, nos hemos concentrado en cargar los registros dobles con constantes, de la forma:

```
LD rr,mn
```

en donde rr es DE, BC o HL y mn es la constante, dividida en byte alto (m) y byte bajo (n).

MNEMONICOS	OPCODES
LD (pq),BC	ED 43 q p
LD (pq),DE	ED 53 q p
LD (pq),HL	ED 63 q p
LD (pq),HL	22 qp

Tabla 2.

Estos registros constituyen una forma ideal de realizar las funciones PEEK y POKE desde lenguaje máquina, aquellas que para 8 bits hacíamos con el acumulador:

```
LD A, (pq) PEEK
LD (pq), A POKE
```

La versión de 16 bits es:

```
LD (pq),rr POKE
LD rr,(pq) PEEK
```

en donde rr es cualquiera de los registros dobles y pq es una dirección de memoria de dos bytes, CUYO CONTENIDO es cargado en el par de registros, o bien el contenido de los registros es introducido en esa dirección. La Tabla 2 muestra los códigos de operación.

Lo interesante y útil de todo esto, es que manejamos 2 posiciones de memoria a un tiempo. Por ejemplo:

```
LD (&3000),HL
```

CARGARA &3000 CON L Y &3001 CON H, en la forma acostumbrada de byte bajo y luego, el alto.

De manera un tanto más «seria»:

```
LD (pq), rr
```

causa que:

```
(pq) <= = byte bajo
(pq+1) <= = byte alto
```

donde las flechas significan «es cargado con».

Pensadlo con atención y tratad de grabarlo en la mente: sólo se alteran los contenidos de las direcciones, no las direcciones mismas (en el ejemplo de más arriba, el número &3000 NO CAMBIA).

DIRECCION	OPCODES	MNEMONICOS
3000	21 34 12	LD HL,&1234
3003	22 F8 2F	LD (&2FF8),HL
3006	C9	RET

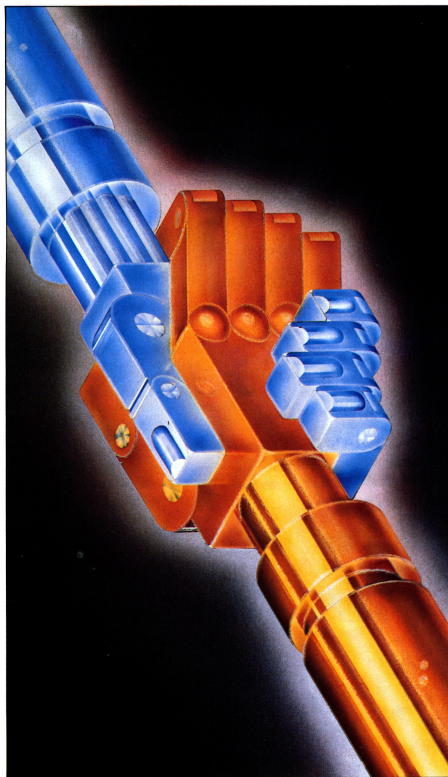
Programa 7

El programa 7 aclara este concepto importantísimo para la programación en máquina.

Después de ejecutarlo, Hexer mostrará en &2FF8 y &12 en &2FF9, según lo previsto.

Comprobar lo que sucede empleando en el programa 7 los otros dos pares de registros BC y DE. No hay diferencia.

Código máquina



DIRECCION	OPCODES	MNEMONICOS
3000	2A 00 30	LD HL,(&3000)
3003	22 F8 2F	LD (&2FF8),HL
3006	C9	RET

Programa 8.

Como comentamos antes, también podemos hacer PEEK de dos posiciones consecutivas de memoria mediante, por ejemplo:

LD HL, (&3000)

El contenido de la posición &3000 irá a parar al registro L y el de la &3001 descansará en paz en el registro H (en efecto, byte bajo, byte alto).

Observad ahora un punto importante, que quedará, por fin, completamente claro; fijaros en la diferencia entre estas dos órdenes:

LD HL, (&3000)
LD HL, &3000

Quizá alguno se sienta impaciente y se pregunte por qué volvemos a insistir entre la diferencia que hay entre direcciones y su contenido, pero la práctica en programar se lo explicará claramente con el tiempo.

La segunda orden pone en L, &00 y en H, &30, mientras que la primera hace algo absolutamente distinto.

Si uno comienza a trabajar con ensamblador, este tipo de cosas es una fuente de errores casi segura al principio, pues el ensamblador no es capaz de detectar la diferencia de significado entre ambas órdenes, mnemónicos válidos del Z80.

En resumen, formalmente, el PEEK desde máquina puede representar-se como:

LD rr, (pq)

La ejecución de esta orden causa que:

rr byte bajo <= (pq)
rr byte alto <= (pq+1)

La Tabla 3 contiene los opcodes.

A estas alturas en el desarrollo del artículo resulta un poco tonto dar un ejemplo acerca de esta instrucción, pero creedme, es muy útil. Por si acaso, va el programa 8.

De todas maneras, tratad de prever qué contendrá el espacio de trabajo del Hexer (comenzando en &2FF8) al ejecutar el programa. Por favor, inténtese también reescribir el programa usando los otros registros dobles.

Una última puntualización: los registros no se pueden combinar de otra forma que la mostrada en este artículo. No existe el registro doble HB, por ejemplo.

Pronto veremos lo poderosos que pueden llegar a ser estos registros.

MNEMONICOS	OPCODES
LD BC, pq	ED 48 q p
LD DE, pq	ED 58 q p
LD HL, pq	ED 5B q p
LD HL, pq	2A q p

Tabla 3.

CUADRADOS EN MOVIMIENTO

Análisis

En esta ocasión análisis estudia las técnicas elementales de animación, con cuadrados que se alejan desde el primer plano de la pantalla hacia el fondo. Cambiando en la rutina de la línea 180 la forma de cuadrado por otra diseñada por nosotros mismos, podremos observar cómo ésta se aleja de la pantalla.

30 Pone nuestro **Amstrad** en el modo 0, en el cual pueden manejarse 16 colores.

40 Inicializa la variable CUADRADO, dándole el valor 1. Esto nos sirve para seguir la pista de los cuadrados y los colores utilizados para dibujarlos.

50-60 Asigna los valores iniciales a las variables XIZQUIERDA, YABAJO, YARRIBA.

Valores que corresponden a las esquinas del cuadrado exterior.

70 Efectúa una llamada a la rutina de dibujar cuadrados, para poner en pantalla el más grande de ellos.

80-120 El ciclo FOR... NEXT, rota dibujando cada vez un cuadrado más pequeño.

90-120 Calcula los nuevos valores de las ordenadas del cuadrado a dibujar.

Estos valores son calculados de forma que cada nuevo cuadrado quede inscrito dentro del anterior.

110 Llama a la rutina de dibujar cuadrados, usando los nuevos valores calculados en las líneas 90-100.

130 Un ciclo vacío que utilizamos para producir una pausa en el programa, manteniendo en imagen los cuadrados multicolores.

140 Hace una llamada a la rutina que cambia el color de los cuadrados al color del fondo, en este caso el azul.

150-170 Constituye un ciclo WHILE... WEND sin fin.

Cada vez que rota el ciclo, la rutina de cambio de color se llama, produciendo el efecto de animación al pintar sobre un cuadrado amarillo, otro azul, haciendo que el anterior se borre.

190-250 Esta es la rutina que contiene el proceso para dibujar los distintos cuadrados.

La figura 1 muestra la forma en la que se definen los cuatro variables que identifican los vértices del cuadrado.

El valor de la variable CUADRADO en la línea 210, elige el color en el cual se han de dibujar los lados del mismo.

260-300 Este ciclo FOR... NEXT rota 13 veces, cambiando el color de los lados, definida en la variable TINTA, al color del fondo (INK 1).

310-370 Estas son las líneas responsables del aparente movimiento de los cuadrados.

Cada vez que rota el ciclo FOR... NEXT, el color de la tinta es cambiado a amarillo (INK 24), para que una vez ejecutado el retardo de la línea 340, vuelva a ponerse del color del fondo.

Los cuadrados aparecen brevemente para después volver a desaparecer. De la misma forma que el FLASH varía entre 1 y 13, igualmente el tamaño de los cuadrados se hace más pequeño, produciendo de esta manera un efecto de animación muy conseguido.

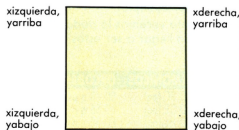


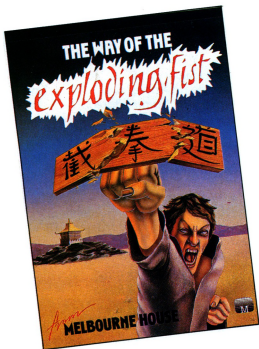
Figura 1: Coordenadas del cuadrado



Siempre los primeros en tener lo mejor...

circulo de soft

MICROAMIGO S.A.



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

¡Un increíble regalo por cada programa!

y además...

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

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



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

CUPON DE PEDIDO

P.V.P.

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

TÍTULO _____ ORDENADOR _____

Contraseña/bloqueo Giro Postal Talón adjunto a Microamigo, S.A.

Nombre _____ C.P. _____

Apellidos _____ Localidad _____

Domicilio _____ Provincia _____

Localidad _____ Teléfono _____

Fecha caducidad _____

Edad _____

Sin duda alguna

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

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

SIN DUDA ALGUNA está abierta a todos.

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

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

Me gustaría saber si se puede ampliar la memoria del **Amstrad CPC664** y hasta cuánto.

También quisiera averiguar si esta máquina puede trabajar con los lenguajes FORTRAN y COBOL.

Roberto Marfil Pérez

No está previsto ampliar la memoria RAM del 664, debido fundamentalmente a la aparición de su sucesor el CPC6128, con 128 kbytes de RAM.

Respecto a estos lenguajes, Fortran y Cobol, existen bajo el sistema operativo CP/M, pero aún no están disponibles.

Me gustaría saber si existe algún modem en España y en el extranjero para **Amstrad**.

También si al 664 se puede conectar un cassette estándar y no el de la marca.

José González

Respecto al modem, por ahora no existen en nuestro país para Amstrad, debido a la escasez de bases de datos accesibles al gran público. No cabe duda de que pronto aparecerán modems, aunque a corto plazo sus posibilidades de uso son pocas.

Al 664 se le puede conectar cualquier cassette normal y corriente. Sólo es necesario adquirir un cable especial que une el ordenador con dicho cassette.

Os escribo para que me aclaréis un concepto que exponéis en el artículo «Banco de pruebas» de número uno.

En el citado artículo decís que el **Amstrad CPC464** ha sido pensado para «pisar fuerte en el terreno de la diversión, de los juegos por ordenador». Yo acabo de adquirir dicho modelo y me gustaría emplearlo en un sentido más «serio», por ejemplo quiero que me ayude a comprender más la Informática. ¿El CPC464 puede hacer todo esto?

Esto es todo lo que me gustaría que me aclararais.

Jesús Moreno

El 464, en efecto, tira más hacia el terreno de los juegos, pero ello no es obstáculo para emplearlo en cosas serias como aprender programación o lenguajes.

Ahora bien, existen programas que sólo tienen sentido en disco, como bases de datos de regular tamaño, hojas de cálculo, etc.

— ¿Cómo puedo lograr en el 464 lo que en el 664 hace COPYCHR\$, es decir, que me lea el carácter de un determinado punto de la pantalla?

— ¿Cómo lograr borrar el buffer de teclado, como en el 664 lo hace CLEAR INPUT?

— ¿Y cómo lograr abrir un canal hacia la impresora, es decir, como un PR 1 de un Apple, o un OPEN 1 de Atari o de Commodore...?

Claudio Pérez-Olea (Madrid)

1. *LLamando a la rutina firmware &BB60. Esta rutina leerá un carácter de la ventana activa en ese momento en la posición indicada por el cursor (de texto).*

Debes emplear lenguaje máquina o ensamblador. El carácter leído se deposita en el acumulador.

La secuencia sería la siguiente:

```
CALL &BB60  
CALL &BB5A  
RET
```

La llamada a la segunda rutina provoca que el carácter leído se escriba en la pantalla.

Si quieres hacer esto desde Basic, debes introducir en la memoria los siguientes números en hexadecimal, mediante el comando POKE:

```
CD 60 BB CD 5A BB C9
```

precedidos cada uno del signo &, por supuesto, y efectuar luego un CALL a la dirección donde hayas cargado el primer número, ya desde Basic.

2. *Desde Basic, ejecuta la orden:*

```
CALL &BB03
```

3. *Añade a los comandos de pantalla (PRINT, LOCATE, etc.) #B. La impresora tiene asociado el canal #B.*

Por ejemplo:

```
LIST #B
```

listará el programa Basic que se encuentra en la memoria por la impresora.





... TAMBIEN PERIFERICOS PARA AMSTRAD

Mercado común

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

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

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

Apartado de correos 54.062
28080 MADRID

¡ABSTENERSE PIRATAS!

Desearía contactar con usuarios de **Amstrad** en toda España, para intercambiar impresiones, ideas, software, etc. Los interesados dirigirse a Juan Antonio Granado Andrade. Calle Los Molares, 8. Lebrija (Sevilla). Preferentemente zona de Barcelona.

Vendo ordenador Commodore Vic-20+C-11 DATASSETTE+Guía del Usuario VIC-20+ Introducción al Vic-20+cartucho de juego VIC AVENGER+10 revistas especializadas en este ordenador+libro Basic Básico+cassettes con programas interesantes. Todo ello por 30.000 pts., o cambio por impresora con enlace Centronics. También vendo por separado. Interesados escribir a: Daniel Coll Monné. Montseny, 77 - 2.º - 1. Sta. Eugenia de Ter. 17005 Girona. Tel. (972) 23 57 77. De 9 a 11 h.

Vendo las revistas de **MICROHOBBY** a partir del número 15 y hasta el 40, al precio de 100 pesetas cada una. Interesados llamar al teléfono 415 58 33. Preguntar por José M.º Samperio. Madrid. Horario de 12 a 1 y de 4,30 a 8.

Desearía contactar con usuarios del **Amstrad 464**, para intercambiar trucos, ideas, juegos, etc. Juan Ramón Puig Meya. Paseo Ramón Vall, 42 - 3.º - 1.ª. Navas (Barcelona).

Desearía contactar con usuarios OC Clubs de **Amstrad 464** para intercambio de información, ideas y programas. Contestaré a todas las cartas. Vicente Sánchez-Brunete Ingelmo. Camarena, 169 - 3.º Dcha. 28047 Madrid.

Vendo consola video-juegos, marca **INTELLIVISION**, nueva, con 10 cartuchos de juegos y adaptador para TV. 23.000 ptas. Manel Ciurana. Barcelona. Tel. 305 21 71.

Clases particulares de programación. En su casa con su micro o en la mía con mi **Amstrad CPC 664**. Nociones básicas de programación y lenguaje Basic. Horarios a convenir. Interesados llamar a: Albert Esplugas. Tel. (93) 239 62 93. Barcelona.

AMSTRAD CPC 664. Interesados en el intercambio de programas (*preferiblemente juegos*). A poder ser, lista de los mismos. Prometo respuesta. Escribid a: Arturo Martínez. Santiago, 20. P. Baja. San Vicente del Raspeig (Alicante).

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.

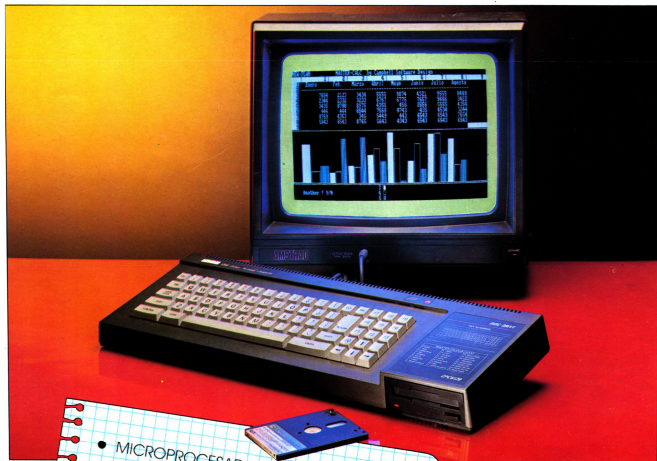
¡ENVÍANOS TU PROGRAMA!

Indicando claramente en el sobre:

AMSTRAD SEMANAL

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

AMSTRAD CPC-6128



- MICROPROCESADOR Z80A.
- 128 K DE MEMORIA RAM
- 48 K DE MEMORIA ROM QUE INCLUYEN EL LOCOMOTIVE BASIC Y EL SISTEMA OPERATIVO.

• 76 TECLAS, TECLADO NUMERICO Y DE CURSOR INDEPENDIENTE.

• TEXTO EN MONITOR DE 20, 40 U 80 COLUMNAS Y GRAFICOS CON DEFINICION DE HASTA 640 X 200 PUNTOS. 27 COLORES DISPONIBLES.

• HASTA 8 VENTANAS EN PANTALLA.

• GENERACION DE SONIDOS EN 3 VOCES Y 8 OCTAVAS.

• UNIDAD DE DISCO DE 3" (169 K BYTES)

• SISTEMAS OPERATIVOS AMS-DOS Y CPM/PLUS

• CONECTORES PARA IMPRESORA, JOYSTICKS, CASSETTE, SEGUNDA UNIDAD DE DISCO, ETC.

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

109.500 Pts.

SISTEMA COMPLETO IGUAL AL ANTERIOR PERO CON MONITOR EN COLOR.

134.500 Pts.

AMSTRADSM
ESPAÑA

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

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

AMSTRAD[™]

“Lo increíble”

Confirmado por la prensa especializada

tu Micro



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

Computer Schau



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

Computer persönlich



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

micro



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

POPULAR Computing WEEKLY



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

CT



¡Solución total a un precio fenomenal!

Personal Computer World



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

micro bit



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

COMPUTER CHOICE



El mejor Basic que he visto.

SCIENCE VIE MICRO

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

micros



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



AMSTRAD[™] “Lo increíble”