

MICROHOBBY

AÑO I N.º 5

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO I N.º 5

150 Ptas.

Canarios 160 ptas.

**ESCUCHA
LA VOZ
DE TU
AMSTRAD**

**¿MAS
MEMORIA
PARA EL
CPC464?**

**MONITOR
DE INGLES:
APRENDER
JUGANDO**

SOFTWARE

**ROCKY: LA GRAN
ESPERANZA BLANCA
¿CONSEGUIRA VENCER
A FRANK BRUNO?**



ERBE Software

¡¡NO HAS VISTO NADA IGUAL!!



**EL MEJOR PROGRAMA DE KARATE QUE PUEDES ENCONTRAR
EL JUEGO DEL AÑO ¡PIDELO YA!**

**DISTRIBUIDOR EXCLUSIVO DE ESPAÑA: ERBE SOFTWARE
STA. ENGRACIA, 17 - 28010 MADRID. TEL. 447 34 10**

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO I N.º 4

Director Editorial
José I. Gómez-Centurión

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
Amalia Gómez
Juan J. Martínez

Secretaría Redacción

Carmen Santamaría

Fotografía
Carlos Candel
Javier Martínez

Portada
Manuel Barco

Ilustradores
J. Igual, J. Pons, F. L. Frontán,
J. Sepián, Pejo, J. J. Mora,
Luigi Pérez

Edita

HOBBY PRESS S.A.

Presidente

María Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

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

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González
M.ª del Mar Calzado

Redacción, Administración y Publicidad

La Granja, s/n

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Año I • Número 5 • 1 al 7 de Octubre de 1985
150 ptas. (sobretasa Canarias, 10 ptas.)

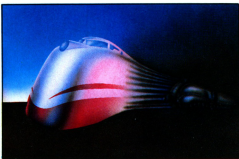
5 Primera Plana

Ante la subida de precios y las medidas de homologación, se especula con reformar el **Amstrad CPC464**.

Nuevas tecnologías aumentan un 40 por 100 la velocidad operacional de los chips.

Banco de 6 pruebas

Para hacer más amigable la interface hombre-máquina, nada más lógico que conseguir que el ordenador emplee la voz humana a la hora de comunicarse con nosotros. Examinamos un sintetizador de voz compatible CPC464 y CPC664.



14 Serie oro

Los programas educativos poseen el atractivo de que nos permiten aprender rápida y fácilmente, pues la paciencia y capacidad didáctica de nuestro interlocutor no se acaba nunca, hagamos lo que hagamos. Consientes de ello, presentamos una aplicación que permitirá practicar inglés a todo el mundo, aunque la metodología aplicada en el programa está pensada para los más jóvenes.

No todo ha de ser serio, a pesar de ser también útil y atractivo. Digger nos sumirá en la emoción de la aventura en un planeta extraño, donde debemos conseguir un precioso combustible esencial para nuestra supervivencia.

Primeros 10 pasos

Las variables son esenciales a la hora de realizar un programa, por simple que sea. Ahorran memoria e incrementan la rapidez y claridad de cualquier aplicación.

Vamos a aprender, paso a paso, cómo se manejan.

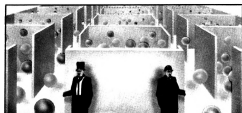


26 ProgramAcción

Está muy claro que los colores, adecuadamente combinados, aumentan significativamente la presencia de cualquier cosa que se encuentre en la pantalla del ordenador. **Amstrad** tiene nada menos que 27 colores a nuestras órdenes para conseguirlo. Veremos de forma simple cómo utilizarlos.

30 Código Máquina

Como ya habrán podido observar los lectores que sigan estas series, los registros son fundamentales en la programación en lenguaje máquina, debido al juego de instrucciones y concepción del Z80. Por tanto, seguimos aprendiendo nuevas formas de manejarlos, recalcando la importancia relativa y posibilidades de cada uno de ellos.



Se solicitará control OJD

Imprime
ROTEDEC, S.A. C/ta. de Trún.
Km. 12,450 (MADRID)

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

Fotomecánica
GROF
Ezequiel Solano, 16

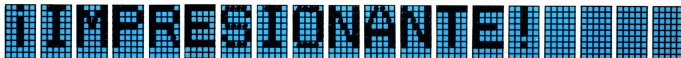
Depósito Legal:
M-28468-1985

Derechos exclusivos de la revista
COMPUTING with the AMSTRAD

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

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

ASTRON 1400



Impresionante en precio y prestaciones. Con todo lo que puede y debe exigirse a una impresora de la más avanzada tecnología: alta velocidad (140 caracteres por segundo), amplio juego de caracteres, alimentación con cualquier tipo de papel (continuo, hojas sueltas, en rollo), excelente calidad de impresión (matriz de 9 x 9), interface centronics (100% compatible con el IBM PC) y, opcionalmente, Serie RS-232 C, ...

Compacta y robusta hasta en el menor detalle de diseño y a un precio que hasta hoy parecía imposible. Impresionante ASTRON 1400.

ESPECIFICACIONES

- Sistema:** Matriz de puntos (9 x 9) con agujas reemplazables.
Velocidad: 140 c.p.s.
Fuentes: 96 caracteres ASCII y 32 semi-gráficos.

Columnas: 80 (caracteres normales), 132 (comprimidos), 40 (ensanchados), 66 (ensanchados-comprimidos).

Alimentación: Fricción y tracción.

Papel: Hojas sueltas de 102 a 254 mm. de anchura.

Papel continuo de 242 mm. (opcionalmente de 102 a 254 mm.).
Papel en rollo de 102 a 254 mm.

Interface: Paralelo CENTRONICS. Opcionalmente, Serie RS-232C.

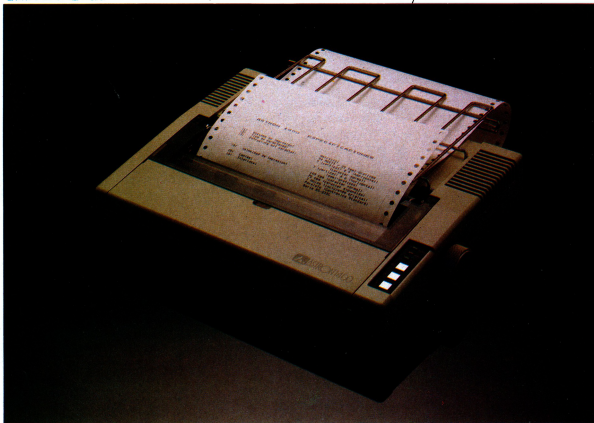
Buffer: 2 kbytes de memoria.

P.V.P.

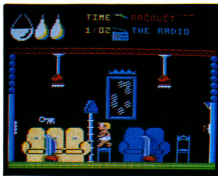
66.900 pts.



Pº de la Castellana, 179. 28046 MADRID. Teléfono: 442 54 33 / 44



Primera plana



HERBERT

Quizá el nombre de **HERBERT** no significa nada para la mayoría del público, en cambio si hablamos del hijo de **WALLY**, el famoso héroe de **MICRO GEN**, esto ya empieza a tener más sentido.

WALLY es el protagonista de tres aventuras: **AUTOMANIA**, **PYJAMARAMA** y **EVRYONE'S A WALLY**. Realizadas por la firma británica **MICRO GEN**.

Por supuesto, estamos hablando de juegos para **SPECTRUM**, que como siempre es el precursor en este mundo. En cambio para el **Amstrad**, solamente hemos podido ver el último de la trilogía **EVRYONE'S A WALLY**, aunque no se ha comercializado en España.

Por fin, tenemos la ocasión de contemplar un juego de **MICRO GEN** en nuestro país, y este camino comienza con **HERBERT**, que para facilitar su uso por los usuarios españoles, viene con los mensajes de pantalla traducidos al castellano, cosa que siempre representa una gran ventaja si se compara con los juegos que emiten los mensajes en inglés.

Su aparición en el mercado nacional se anuncia próxima.

MACROPROCESADORES

Mitsubishi, el gigante japonés de la electrónica y de casi todo, está decidida a obligarnos a cambiar el nombre de los microprocesadores.

Un nuevo procedimiento en la fabricación de chips y un nuevo material, el silicio de titanio, permitirán conseguir verdaderos monstruos que manejen hasta 4 megabytes de memoria, un 40 por 100 más rápido que los actuales.



CP/M, CADA VEZ MAS CERCA

Parece que la fiebre del disco en formato 5 1/4 pulgadas se ha desatado en el mundo de **AMSTRAD**.

La tentación de aumentar ventas, tanto de software como de hardware, al permitir a todos los usuarios de la gama **AMSTRAD** acceder a la vasta biblioteca de software **CP/M**, es demasiado fuerte.

Tanto es así, que lejos de esperar a que los fabricantes se animen a adaptar más programas a discos de formato 3 pulgadas, escogido oficialmente por la compañía inglesa, ella misma, junto con **TIMATIC SYSTEMS**, a desarrollado el nuevo drive.

Esperemos que este interesante producto tenga una buena acogida, sin que por ella disminuya la constante labor de adaptar más programas **CP/M** a los discos de 3 pulgadas.

NOVEDADES DE ICL

La principal empresa británica fabricante de ordenadores personales, **Internacional Computers (ICL)**, ha anunciado el lanzamiento de un nuevo ordenador de la gama alta de los personales, capaz de servir simultáneamente a cuatro usuarios, destinado por supuesto al mundo de la gestión, y específicamente a las pequeñas empresas.

Al parecer, **ICL** considera que el mercado de los ordenadores multiusuario crecerá mucho más rápidamente que el de los personales «a secas».

El precio de esta máquina, en Inglaterra, estará alrededor de los 660.000 ptas.

AMSTRAD CPC-464 REMODELADO

Amstrad España, afectada por las medidas arancelarias del decreto del 17-6-1985, intenta adecuar el modelo más bajo de la gama, el **CPC 464** para que éste no sufra el incremento de precios que supone la elevación del arancel.

A esto hay que añadir también la homologación a las que se deben someter todos los modelos de ordenadores que quieran ponerse a la venta en España a partir del próximo noviembre.

Ante este aluvión de novedades y modificaciones, los directivos de **INDESCOMP**, distribuidor oficial de **Amstrad** en España, viajaron a Londres con motivo de la **P.C. WORLD SHOW**, donde pudimos comprobar personalmente cómo se reunían en el stand de **Amstrad** con **ALAN SUGAR** (director de **Amstrad**) para tratar de encontrar una solución viable al tema.

El **CPC 464**, debe mantener su competitividad en el mercado español y una subida de precio le pondría en la cola del ranking de ventas. Además, no debemos olvidar los nuevos modelos de la casa, que le han relegado al puesto de ordenador casi exclusivamente de uso recreativo.

Con el objetivo de mantener la competitividad del modelo e incluso, hacerlo mucho más interesante frente a los ojos de los consumidores de software recreativo, se estudia la astuta medida de aumentar la memoria mediante una expansión interna; con lo cual ésta sería superior a los 64 K mínimos exigibles y pondría al **CPC 464** en un precio bastante favorable frente a los 64 K de los otros ordenadores que probablemente sufrirán la subida de aranceles.

Una vez comprobada la viabilidad técnica de la citada expansión, el modelo bajo de la creciente gama **Amstrad**, estará dotado de memoria adicional y adecuado a las normas de homologación, incluyendo la inefable ñ, lo cual le convertirá en uno de los modelos más competitivos en los ordenadores de juegos. Incluyendo en su precio el monitor y las altas prestaciones que sólo puede dar el **LOCOMOTIVE BASIC**, lenguaje residente en todos los modelos **Amstrad**.

AMSTRAD AL HABLA

Los usuarios de ordenadores de hoy día no son como los de antes. Este comentario, puede escucharse a menudo entre los fabricantes y vendedores de software y hardware acompañado de suspiros nostálgicos.

La razón es muy simple: la informática evoluciona rápidamente, pero la gente cambia aún con mayor rapidez. Los usuarios de ordenadores saben lo que quieren y cómo lo quieren, porque la gran mayoría de ellos lo usan para su trabajo, como herramienta, y si también pueden divertirse un poco con su máquina, bueno, ¿por qué no?

Oh tempora, oh Moris

Los días de los pioneros, en los que ser poseedor de una computadora implicaba un adevado programador dedicado en cuerpo y alma a su misión sagrada, explorando un terreno completamente virgen, se encuentran enterrados, cubiertos de polvo, entre los balances de beneficios de las grandes compañías que se reparten el pastel con fruición.

El romanticismo ha muerto en la informática, pero de ahí ha salido un enorme grupo de usuarios experimentados que no sólo quieren una herramienta de trabajo, sino un instrumento que sea sencillo de manejar, rápido y eficiente.

Ahí está el quid de la cuestión. Como no hay más remedio que interactuar con un ordenador para que nos entienda y nos sirva de algo, hay que buscar una forma de hacerle saber nuestros deseos.

Aunque hoy nos parezca increíble, primero se comenzó con los teletipos y las tarjetas perforadas. Por no haber, no había ni monitores monocromos.

Estos métodos demenciales de introducir datos a las máquinas pronto fueron abandonados, y se sustituyeron por los famosos comandos, es decir, uno se aprende un resumen crítico del lenguaje inglés, y, tras más o menos 100 horas de práctica puede plantearse COMENZAR a programar un poco.

Para los programadores, miel sobre hojuelas (a ellos les cuesta mucho menos), pero los sufridores que sólo deseaban utilizar su ordenador lo pasaban bastante mal.

Interfaces más amigables

Por eso, desde hace ya tiempo, se ha tratado de suprimir las dificultades de aprendizaje en el uso de ordenadores, para así hacer desaparecer las barreras que frenan a muchos a la hora de adquirir uno.

Ya puestos, no hay que conformarse con conquistas pequeñas, así que... **¿qué pasaría si los ordenadores hablan?** Y, sobre todo, **¿qué pasaría si un computador ENTENDIERA EL LENGUAJE HUMANO SIN MAS?** Yo hablo y tú obedeces.

La cantidad de dinero invertida en la consecución de ambos objetivos ha sido, y es, astronómica; no obstante, sólo se ha conseguido a medias.

El hacer que un ordenador entienda el sentido de nuestras frases, sin más, pertenece todavía al mundo de la ciencia ficción. Muchos investigadores se quedan calvos por año intentándolo (aunque *quién sabe lo que andará por los laboratorios*).

Sin embargo, el que una máquina pueda sintetizar la voz humana y pueda obedecer órdenes simples preprogramados por medio de la voz, ya es otro cantar. Es mucho más fácil, y si no, aquí tenemos el sintetizador de voz de **Amstrad** para probarlo.

Voz por ordenador: ¿ciencia o magia?

El aparato, o periférico, o cualquier otra palabra griega de dudoso significado con la que se nos ocurra nombrarlo, viene dividido en varias partes:

- Dos altavoces destinados a producir el sonido en stereo.
- El hardware propiamente dicho, que se conecta al bus de expansión del **Amstrad**.
- Una cinta de cassette conteniendo el software necesario para su funcionamiento.



No hemos podido averiguar si existe una versión en disco del invento, aunque sí sabemos que es perfectamente compatible tanto desde el punto de vista hardware como software, con el **Amstrad CPC464** y el CPC664. De hecho, este artículo se está escribiendo con un procesador de textos y el programa del sintetizador de voz coexistiendo pacíficamente en la memoria de un 664.

El montaje del sintetizador no reviste ningún problema: en la pieza que se conecta al bus de expansión hay dos slots, cada uno para un altavoz. Una vez enchufados, todo lo que hay que hacer es encender el **Amstrad** y cargar el programa de la cinta. El ordenador está listo para hablar.

Es imprescindible que ni el **Amstrad**, ni ningún periférico conectado a él, esté encendido si queremos conectar o desconectar el sintetizador. De lo contrario, pueden ocurrir daños irreparables.





teizador tal vez sea la cinta con el programa. Sus creadores han sabido aprovechar las facilidades sabiamente implementadas por **Amstrad** en el firmware. Me refiero a la facilidad con que se manejan las **RSX (Resident System eXtensión)**, lo que en definitiva significa que, para un programador en lenguaje máquina, resulta relativamente poco complejo añadir nuevos comandos al **Basic Amstrad**.

Los analizaremos dentro de un momento.

El software de la cinta consta de dos partes: un programa **Basic** que comprueba que el hardware del sintetizador está presente, nos indica la longitud del programa en máquina que viene inmediatamente después, y nos pregunta si queremos que se ubique en alguna dirección en concreto. Una vez respondidas estas cuestiones, la parte en máquina del programa sintetizador de voz se carga en la memoria y ajusta el **HIMEM** adecuadamente.

Nuevos comandos para el Basic

Los nuevos comandos del sintetizador toman control del **Amstrad** desde 4 niveles distintos, es decir, poseen una jerarquía. Cada uno de ellos cumple una determinada función, y no es muy aconsejable usar simultáneamente los

Banco de pruebas



Un diez para los manuales

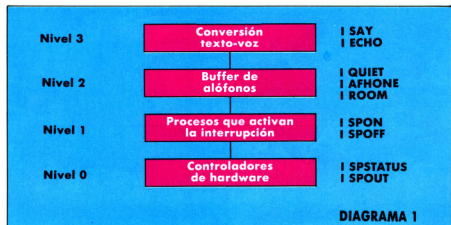
El manual del usuario, al menos la versión inglesa, única a la que hemos podido acceder, la verdad es que es excelente.

En sólo 20 páginas, de forma sencilla, se introduce a lector en el uso de software del sintetizador, desde los niveles más elementales, hasta los más complejos, indicando cómo se puede usar desde lenguaje máquina.

Incluso se dan, para los interesados en el hardware, un pequeño diagrama de bloques y una ligera noción de cómo manejar directamente el sonido del sintetizador, accediendo directamente al procesador de voz.

Hay que tener en cuenta que este aparato ha sido creado pensando en las reglas de sintaxis y pronunciación del idioma inglés. Ello no implica que en español la máquina produzca únicamente sonidos incomprensibles, sino tan sólo que se requiere un trabajo extra sobre el manual, y alguna experimentación para que el resultado corresponda a lo que esperamos oír. Por ejemplo, en una palabra al dársela al ordenador para que la «lea», tal vez hay que modificarla un poco, añadiendo más vocales o algo parecido. Este sintetizador no es la panacea universal en cuanto a pronunciación; tiene sus lagunas incluso con palabras inglesas. Sin embargo, cumple su función aceptablemente bien.

En esencia, la parte más interesante del sin-



comandos de los niveles 3, 2 y 0 (diagrama 1).

El nivel 3 es el más alto de todos, es decir, el más cercano al lenguaje humano y el más lejano al lenguaje máquina (normalmente, alto nivel y bajo nivel tienen en informática esos significados). Aquí se produce la conversión de textos en formato ASCII a los **ALOFONOS**, usando un «intérprete». Los alófonos así producidos son enviados a ocupar su lugar en una cola de espera en el segundo nivel. Cuando se llena, los comandos de nivel 3 suspenden la ejecución de cualquier programa **basic**, y del propio intérprete **Basic**, hasta que todo el texto ha sido reconvertido.

En el segundo nivel es donde se encuentra este buffer, capaz de almacenar hasta 64 alófonos.

La principal misión de este nivel es asegurar, mediante interrupciones, que el programa **Basic** de fondo se seguirá ejecutando normalmente y que los alófonos se «pronunciarán» en el momento oportuno.

El nivel 1 maneja también las interrupciones mediante sus comandos, y permite que un proceso sea inicializado y lea caracteres del buffer, para enviarlo al procesador de «palabra hablada» del sintetizador. Cuando el proceso se desactiva, se impide que sean leídos más caracteres del buffer, a la vez que se envía una señal al procesador de voz para que el último alófono no suene para siempre. El buffer queda listo para llenarse de nuevo con los datos de los niveles 2 y 3.

El nivel 0, por fin, maneja directamente

COMANDO I SAY

1. USO: Conversión del texto ASCII a palabra hablada.

2. SINTAXIS: I SAY, variable de cadena.

Por ejemplo:

I say, **¡hola!**

a\$=**hola**: I say, a\$

3. COMENTARIOS: Todos los datos del buffer de alófonos son enviados al procesador de voz antes de manipular la nueva frase, si es que existen.

Si dicho buffer se llena, el Basic queda en suspenso hasta que todo el texto sea procesado.

COMANDO I ECHO

1. USO: Activa la conversión texto-voz sobre texto escrito en la pantalla.

2. SINTAXIS: I ECHO, modo.

3. COMENTARIOS: «modo» es un número entero de 0 a 4 que indica el tipo de respuesta que se producirá al activar el comando.

Así:

— Modo 0: Desactiva cualquiera de los otros cuatro modos.

— Modo 1: Todo el texto incluido entre los delimitadores «» (SHIFT + «») es hablado y escrito en pantalla.

— Modo 2: Todo texto enviado a la pantalla, incluido listado, se escuchará por los altavoces.

— Modo 3: Lo mismo que en el modo 2, pero además se imprime en pantalla.

— Modo 4: Lo mismo que en el modo 1, pero el texto no aparece en pantalla.

el firmware del **Amstrad** y recibe los alófonos que llegan desde el nivel 2 a través de 1.

De aquí salen directamente al procesador de voz, en un formato que pueda entender.

Este es, un poco a grandes rasgos, el proceso que sigue un texto desde que se introduce como una cadena literal, hasta que nuestros oídos lo comprenden.

Como lo prometido es deuda, hemos distribuido en estas páginas un resumen de los 9 comandos que pueden leerse en el diagrama 1, para que cada uno pueda hacerse una idea de lo que el sintetizador es capaz de hacer.

En resumen:

1. El sintetizador está preparado para el inglés, no para el español.

2. Es posible obtener sonidos comprensibles en nuestra lengua, pero requiere un poco más de esfuerzo y experimentación y no siempre será posible.

3. El manejo del aparato y del programa es sencillo y cómodo.

4. Los manuales (*en inglés*) son muy buenos.

5. Se suministra información tanto para el usuario avanzado como para el novel. Sin embargo, la primera es sólo a nivel de inicio. Se ha pensado más en el no iniciado, lógicamente.

6. El sintetizador de voz puede mejorar muchísimo nuestros programas, por razones obvias.

COMANDO I APHONÉ

1. USO: Envía alófonos directamente a su buffer, obviando la conversión texto-voz.

2. SINTAXIS: I APHONÉ, lista de alófonos.

3. COMENTARIOS: Un alófono es un número entre 0 y 63. Los números mayores se introducen dentro de este rango. El número máximo de la lista está limitado por la longitud de una línea Basic.

COMANDO I ROOM

1. USO: Indica cuántos lugares libres hay en el buffer de alófonos.

2. SINTAXIS: I ROOM, variable entera.

3. COMENTARIOS: El resultado del comando se devuelve en esta variable. Los primeros 5 bytes indican el número de lugares libres en el buffer. El bit 6 marca el estado de las interrupciones. Cuando están desactivadas, pueden ser recibidos más alófonos.

El bit 7, cuando está a 1, significa que el sintetizador está «ocupado» manejando un alófono.

COMANDO I QUIET

1. USO: Limpia el buffer de alófonos y envía una pausa al sintetizador.

2. SINTAXIS: I QUIET

3. COMENTARIOS: Puede usarse en cualquier momento en que se desee impedir un sonido.

COMANDO I SPON

1. USO: Activa un proceso por interrupción, y permite que los alófonos del buffer sean oídos independientemente de lo que suceda con el Basic.

2. SINTAXIS: I SPON

3. COMENTARIOS: No se envían alófonos al procesador de voz hasta que un proceso dad sea activado con este comando.

COMANDO I SPOFF

1. USO: Efecto contrario a I SPON

2. SINTAXIS: I SPOFF

3. COMENTARIOS: Este comando no tendrá efecto a menos que se use después de I SPON. Cualquier alófono que quede en el buffer permanece intacto. Debe usarse cuando no sea necesaria la voz durante largos períodos de tiempo.

COMANDO I SPSTATUS

1. USO: Lee el estado del procesador de voz directamente del hardware.

2. SINTAXIS: I SPSTATUS, variable entera

3. COMENTARIOS: En la variable entera se devuelve un entero de 16 bytes que muestra el estado del procesador de voz.



Los primeros tienen lo último.

MICRO **M** **WORLD**
W

Te ofrece *Ya,* el nuevo

AMSTRAD CPC-6128

128 Kb.

Doble capacidad
al mismo precio.



**¡MIRA
QUE PRECIOS!**

AMSTRAD
CPC-6128
(f. Verde)

109.500 Ptas.

AMSTRAD
CPC-6128
(color)

134.500 Ptas.

VEN A VERLO A NUESTROS CENTROS

Fuencarral, 100
Tel. 221 23 62

Modesto Lafuente, 63
Tel. 254 88 36

Colombia, 39
Tel. 458 61 71

Padre Damián, 18
Tel. 259 86 13

José Ortega y Gasset, 21
Tel. 411 28 50

Pedidos por teléfono: 441 12 11
Solicita gratuitamente nuestro catálogo de productos
Servimos a tiendas. Tel. 91-441 12 11

LAS VARIABLES, CLAVE DE LA PROGRAMACIÓN

La semana pasada tuvimos ocasión de escribir y ejecutar nuestros propios programas, haciendo un uso masivo de la sentencia Basic más sencilla y obvia de todas: PRINT.

Sin embargo, el método de programación escogido fue muy primitivo y, por tanto, muy fácil de comprender. Vamos a intentar, paso a paso, añadir a nuestros programas un poco más de sofisticación, introduciendo el CONCEPTO FUNDAMENTAL DE VARIABLE.

Aunque la salida, el OUTPUT, de los programas que veremos no sea tal vez muy espectacular, no le queda duda de que se avanza derecho hacia la comprensión del Basic.

Los programas son secuenciales

Recordemos solamente un concepto esencial más: un programa Basic consiste en una secuencia de instrucciones agrupadas en líneas, cada una etiquetada con su correspondiente número.

Para introducir una en el programa, basta con escribir el número de línea que deseamos que tenga, seguido de las apropiadas PALABRAS-CLAVE (comandos) de Basic y pulsar (ENTER).

Para ver todas las instrucciones del programa:

LIST (ENTER)

Para ejecutarlo:

RUN (ENTER)

Y, por fin, para deshacernos de él y limpiar la memoria antes de introducir un nuevo programa:

NEW (ENTER)

Echemos una mirada al programa número 1, nuestro viejo conocido. Bien, el programa número 2 es otra forma de obtener la misma salida

por pantalla de manera mucho más elegante.

Como ya sabemos, las palabras entrecuilladas se denominan STRINGS, palabra inglesa que estimamos preferible a la traducción española «tiras», aunque hablar de tiras de caracteres recuerda cómo el Amstrad las interpreta, esto es, de forma serial. Para el ordenador, **CASA** es la serie de caracteres C, A, S, A. Nunca son tratados, aunque a nosotros nos parezca lo contrario al verlo impreso en la pantalla, como un todo.

Lógicamente, nos puede interesar escribir una misma frase entrecuillada muchas veces a lo largo de una aplicación.

Por ejemplo, en una carta de negocios hay frases casi standard, que se repiten a menudo en esa misma carta o en otras.

Las variables ahorran esfuerzo y espacio

El Amstrad resuelve el problema asignado a cada cadena una etiqueta, de forma que al «invocarla» en un programa, la cadena de caracteres correspondiente aparece en la pantalla.

Así, en la línea 10 del programa número 2, la etiqueta «A\$» se asigna a la cadena **PROGRAMAR**, de tal forma que cuando el ordenador se encuentra con la línea 40 del programa:

40 PRINT A\$

la palabra **PROGRAMAR** aparece en la pantalla.



La ventaja de esta técnica es la gran cantidad de espacio de memoria del ordenador y de esfuerzo nuestro que se ahorra al escribir los programas.

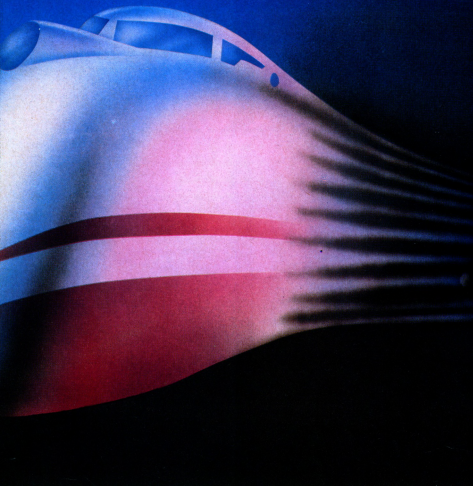
En efecto, basta pensar por un momento en que la palabra representada por A\$ tuviera que aparecer 100 veces en la salida del programa por pantalla.

Análogamente hacemos con las líneas 20 y 30 del programa 2; éstas causan la salida por pantalla deseada cuando se ejecutan las líneas 50 y 60.

En este momento, hay que hacer notar los siguientes puntos:

1. Hemos elegido las etiquetas de modo tal que todas comienzan por una letra y van seguidas del signo «\$».

No estamos obligados a usar una sola letra, podemos poner más, pero SI HAY QUE PONER EL SIGNO «\$», para avisar al ordenador de que queremos etiquetar una cadena de



caracteres (veremos posteriormente cómo etiquetar otro tipo de «cosas»).

2. El hecho de que en el programa 2 las etiquetas se encuentren en orden alfabético es completamente arbitrario. NO TIENEN POR QUE SEGUIR NINGUN TIPO DE ORDEN.

3. Aunque hemos usado un signo «=» para relacionar etiqueta con objeto, su significado no debe entenderse como «igual a...», sino más bien como «asigna a A\$ la cadena entrecomillada que viene después del signo =».

4. La etiqueta debe estar en la parte izquierda de la SENTENCIA DE ASIGNACION (que así se llama), y lo que se nombra en la parte derecha del signo igual.

Pruébese si no, con la sentencia:

10 «PROGRAMAR»=A\$

y ya se verá lo que ocurre. El pobre Amstrad ni se entera de lo que queremos decirle.

5. La etiqueta reemplaza a la palabra entrecomillada y a las propias comillas, ya que, al decir:

40 PRINT A\$

las comillas no aparecen.

6. El ordenador no distingue entre mayúsculas y minúsculas en el caso de nombres de etiquetas. En lo que a él respecta, los programas 2 y 3 son idénticos y producen el mismo resultado.

Esto, que puede parecer obvio, depende del ordenador del que se trate. Otras máquinas son muy rígidas en ello.

No sólo existen los caracteres alfabéticos

Si tecleamos y ejecutamos el programa número 4, observaremos que el espacio en blanco, aunque para nosotros no es un carácter alfabéti-

co como los demás, PARA EL AMSTRAD SI, y su existencia se toma en cuenta si se lo asignamos a una etiqueta, modificando la apariencia de lo que aparece en la pantalla según esté o no.

También la puntuación de la línea 70 del programa 4 FORMATEA la escritura, consiguiendo que las palabras representadas por A\$, B\$, C\$ y D\$ se impriman una inmediatamente a continuación de otra.

Inténtese sustituir los puntos y comas de la línea 70 por comas para ver el efecto y, más tarde, manteniendo los puntos y comas, suprimir los espacios que preceden a las palabras en las líneas 40, 50 y 60.

En este programa (4), puede verse también una nueva palabra-clave o comando Basic, LA SENTENCIA REM (del inglés REMarks, comentarios).

Todo lo que siga a una sentencia REM dentro de la misma línea de programa es completamente ignora-

do por el **Amstrad**, por lo que podemos escribir lo que se nos antoje detrás de un REM sin que se produzca el temido «SYNTAX ERROR».

Sentencia REM: programas más legibles

Incluir sentencias REM en nuestros programas es una excelente técnica de programación, en el sentido de que permite recordar sin esfuerzo qué es lo que hace un programa y por qué lo hace.

Puedo asegurar que un programa medianamente largo y complejo sin sentencias REM, visto dos semanas después de terminarlo, se desconoce por completo POR QUE FUNCIONA. En caso de descubrir un error en ese momento, es una auténtica «tarea de chinos» localizarlo.

No obstante, las sentencias REM tienen un problema: consumen memoria. En programas MUY largos, tal vez merezca la pena plantearse suprimirlas, pero en caso contrario, ya que sobra memoria, usémosla en algo útil que ahorrará posteriormente muchos quebraderos de cabeza.

Nunca viene mal acostumbrarse desde el principio a la jerga empleada por los programadores, así que a partir de ahora nos referiremos a las etiquetas como VARIABLES, no necesariamente en el sentido matemático del término, sino, como comentábamos antes:

A\$ = «TEST»

debe leerse **el VALOR de la variable de cadena A\$ es TEST**. Y el hecho de dar a una variable un valor se conoce como ASIGNACION.

El programa número 5 ilustra cómo los variables de cadena (o ALFANUMERICAS), pueden mezclarse con cadenas literales sin ningún problema.

El espacio incluido en la cadena asignada a B\$ es muy importante. Para ver lo que quiero decir, lo mejor es ejecutar el programa primero con B\$ tal cual, y luego quitando el espacio (línea 40).

En las líneas 60 y 70 puede observarse el efecto de «pegamento» de la palabra-clave «;», y lo que ocurre si se quitan ..., bueno, probad y veréis.

Podemos cambiar el aspecto de la línea 50 tecleando:

```
50 PRINT A$B$
```

esto es, sin espacio entre las variables.

El programa funcionará. No obs-

PROGRAMAS

```
5 REM PROGRAMA I
10 PRINT "PROGRAMAR"
20 PRINT "ES"
30 PRINT "FACIL"
```

```
5 REM PROGRAMA II
10 A$="PROGRAMAR"
20 B$="ES"
30 C$="FACIL"
40 PRINT A$
50 PRINT B$
60 PRINT C$
```

```
5 REM PROGRAMA III
10 A$="PROGRAMAR"
20 B$="ES"
30 C$="FACIL"
40 PRINT a$
50 PRINT b$
60 PRINT c$
```

```
10 REM PROGRAMA IV
20 MODE 1
30 A$="TEST"
40 B$=" TEST"
50 C$=" TEST"
60 D$=" TEST"
70 PRINT A$; B$; C$; D$
80 PRINT "0123456789012345678901234
567890123456789"
```

```
10 REM PROGRAMA V
20 MODE 1
30 A$="MI NOMBRE ES"
40 B$="MIGUEL"
50 PRINT A$;B$
60 PRINT "MI NOMBRE ES ";B$
70 PRINT A$;"MIGUEL"
```

tante, la legibilidad se verá disminuida.

Para remediarlo, en parte, podemos escribir también:

```
50 PRINT A$B$
```

En las sentencias de asignación, opcionalmente, podemos colocar la palabra-clave LET (forma inglesa del imperativo):

```
40 LET B$="MIGUEL"
```

pero de hecho, no se usa y su inclusión se justifica para mantener la compatibilidad con la versión Basic

standard (si es que hay alguna).

Esta semana hemos aprendido a manejar un concepto absolutamente esencial para la programación: EL CONCEPTO DE VARIABLE.

Con ello y el estudio de la sentencia INPUT próximamente, estamos introduciéndonos en el mundo de la programación por la puerta grande, conocida entre los programadores como PROGRAMACION INTERACTIVA.

Hasta pronto.

COMPUTER CENTER

COMANDANTE ZORITA, 13
28003 MADRID

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

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

SOFTWARE ENTRETENIMIENTO (CASSETTE)

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

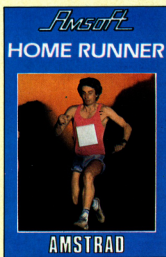
SOFTWARE GESTION (DISCO)

CONTAB. GRAL.	11.995 ptas.
CONTROL STOCK	6.995 ptas.
BASE DE DATOS	6.995 ptas.

LIBROS

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

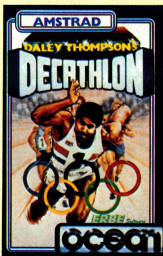
Tu pedido lo puedes recibir contra-reembolso (libre de gastos), llamando a los teléfonos (91) 233 07 35 y (91) 233 07 81.



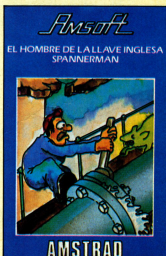
1.495 Ptas.



1.975 Ptas.



1.975 Ptas.



1.495 Ptas.



1.495 Ptas.



1.975 Ptas.

MONITOR DE INGLÉS

Este programa está pensado para refrescar los conocimientos de inglés, o para aprender, cómo se dicen en ese idioma, una serie de palabras y conceptos.

Se ha tratado de hacer su uso cómodo y agradable a la vista, pensando, sobre todo, en los más jóvenes, por aquello de aprender jugando. Sin embargo, no existen restricciones de edad para utilizar este programa.

COMPATIBLE
CPC 464
CPC 664

A qué que use el programa, verá en la parte inferior de la pantalla una palabra o concepto en español, y cuatro posibilidades de cómo se expresa eso en inglés, de las cuales sólo una es la correcta.

Moviendo una figura por la pantalla, se la sitúa enfrente de la que creemos válida, y se pulsa la barra espaciadora.

El ordenador nos dice si hemos acertado o no, y al cabo de 20 intentos, se nos da el tanto por ciento de aciertos, junto con una lista de las palabras que sería conveniente aprender para la próxima vez.

Al arrancar el programa disponemos de 3 opciones:

1. Usar el vocabulario que el propio programa posee incorporado en forma de sentencias DATA.

2. Cargar un fichero de palabras desde cinta/disco.

3. Salvar un fichero en cinta disco.

Una aclaración a esto último: los usuarios de equipos **Amstrad** con unidad de disco, bien sea el CPC464 u otros, deben tener en cuenta que la entrada/salida de datos se reali-

VARIABLES PRINCIPALES DEL PROGRAMA

NOMBRE	FUNCION
c\$, cb\$, a\$, b\$, c\$, p1%	Gráficos. Mensajes. Averigua si se está usando como vocabulario las líneas de DATAS o un fichero en cinta/disco. Almacena las 4 palabras alternativas. Número de la respuesta correcta. Definición del diccionario.
a\$ (x, y)	Número de preguntas.
a% (x)	Pregunta elegida al azar.
b\$ (x)	Número de respuestas erróneas.
p	Número de preguntas.
x	Entrada por teclado.
c% (x)	Movimiento.
b% (x)	Posición de la figura en la pantalla.
t\$, Z	Variables de uso general.
za	
s, y, d	

ESTRUCTURA DEL PROGRAMA

LINEAS	COMETIDO
40-50	Selección de colores y modo de pantalla.
60-210	Define ventanas.
220-270	Define gráficos para la figura.
280-480	Títulos.
490	Música
520	Dimensiona matices.
530	Fichero de cinta/disco o líneas DATA.
540-610	Lee las DATAS si no estamos usando un fichero de cinta/disco.
620-1230	DATAS.
1240	Inicializa puntuación.
1260-1680	Bucle principal.
1290	Selecciona palabra al azar.
1300	Mira si una palabra se ha elegido 2 veces. Si es así, vuelva a intentar con otra.
1320-1410	Impresión de las palabras.
1420-1550	Examina las teclas de movimiento de la figura y la desplaza en consecuencia.
1560	Chequea la respuesta.
1570	Informa si hemos acertado o no.
1660	Va al final del juego.
1690-1740	Fin del juego.
1840-1910	¿Otro juego?
1920-2010	Menú.
2020-2120	Carga el fichero de datos.
2130-2290	Crea el fichero de datos.
2300-2430	Lo salva.

za en el periférico que el sistema tenga en ese momento por defecto, es decir, si por ejemplo tenemos un CPC464 con unidad de disco conectada, las opciones 2 y 3 del programa se realizarán en él, a menos que ANTES DE EJECUTAR EL MONITOR DE INGLÉS, indiquemos al ordenador que deseamos usar el cassette mediante los comandos al efecto.

Si decidimos crear nuestro propio fichero de palabras, el **Amstrad** nos pedirá que introduzcamos 60 palabras. Para cada una de ellas debemos suministrar también las otras 4 alternativas, junto con la definición del diccionario y el número de la respuesta correcta (*del 1 al 4*).

Una vez salvado el fichero en cinta/disco, el ordenador ejecutará inmediatamente el resto del programa usando el nuevo vocabulario.

Es necesario destacar también que el programa siempre carga salva un fichero con el mismo nombre, y, por tanto, espera encontrar ese nombre en la cinta o en el disco cuando se le pide que lo cargue.

Si se desea modificar esto, obsérvense las líneas 2020-2430 del programa.

Serie Oro

```

10 REM *****
20 REM * Monitor de Ingles *
30 REM * By Steve W. Lucas *
40 MODE 1:INK 8,6:ST INK 1,24: INK 2,
20: INK 3,1
50 BORDER 22
60 REM ** definir ventanas **
70 WINDOW #1,15,1,5 :PAPER #1,1:1P
EN #1,3
80 WINDOW #2,15,6,10 :PAPER #2,2:
PEN #2,3
90 WINDOW #3,15,11,15:PAPER #3,1:
PEN #3,3
100 WINDOW #4,15,16,20:PAPER #4,2
:PEN #4,3
110 WINDOW #5,16,22,1,21:PAPER #5,0
:PEN #5,2
120 WINDOW #6,1,40,21,25:PAPER #6,3
:PEN #6,1
130 WINDOW #7,23,40,1,20:PAPER #7,1
:PEN #7,0
140 REM ** definir caracteres **
150 SYMBOL AFTER 240
160 SYMBOL 242,48,32,32,32,32,31,
15
170 SYMBOL 243,1,3,7,7,3,1,255,255
180 SYMBOL 244,0,120,192,240,176,24
0,240,136
190 SYMBOL 245,7,3,1,1,0,1,1,0
200 SYMBOL 246,255,255,253,129,120,
129,193,0
210 SYMBOL 247,134,120,128,128,128,1
20,192,0
220 REM ** definir caracter del gat
o **
230 RESTORE 2450
240 FOR y=1 TO 12:READ d
250 c:=CHR$(CHR$(d))
260 IF d<32 THEN CB:=CHR$(d)
270 NEXT x
280 REM ** titulos **
290 ASL * Monitor de Ingles *
300 CB="" by Steve W. Lucas
310 BB="" +CHR$(10)+CHR$(0)+"" +CHR
$(10)+CHR$(0)+CHR$(0)+""
320 FOR x=1 TO 37
330 PEN 3
340 LOCATE x,1:PRINT bb
350 LOCATE x+1,5:PRINT bb
360 LOCATE x+2,5:PRINT bb
370 LOCATE x+3,5
380 IF x<28 THEN PRINT cb$
390 LOCATE x,18
400 PEN 2
410 PRINT MID$(a$,x,1)
420 LOCATE x,17
430 PEN 1
440 PRINT MID$(c$,x,1)
450 NEXT x
460 PEN 3:FOR x=7 TO 29 STEP 4
470 LOCATE x,2 :PRINT cb$
480 NEXT x
490 PEN 1:GOSUB 1690:REM ** cancion
**
500 DATA 239,239,213,239,190,239,23
9,213,190,179,159,239,239,213,190,1
79,159,239,239,213,239,159,239,142,
239,127,239,127,142,159,179,190,213
,239
510 REM ** Para anadir preguntas ex
tras, incrementar el rango de la ma
triz en la proxima linea **
520 DIM a$(61,4),a$(61),b$(61),n$(5
0),c$(20),b$(60)
530 GOSUB 1930:REM ** utilizacion d
e las datas en el programa **
540 IF p1%>8 THEN 1240
550 REM ** uso de lista de datas en
el programa **
560 FOR x=1 TO 60
570 FOR y=1 TO 4
580 READ a$(x,y)
590 NEXT y
600 READ a$(x),b$(x)
610 NEXT x
620 REM ** Todas las lineas data co
ntienen cuatro respuestas alternati

```

```

vas, el numero de la respuesta corr
ecta y la definicion de la palabra
**
630 DATA control,kontroll,controll,c
ontrol,4,poder de dirigir y goberna
r
640 DATA cassette,cassette,casete,ca
ssete,2,aparato para grabar y repro
ducir cintas
650 DATA complicated,complicated,c
omplexated,complicated,1,complicado
660 DATA probable,probably,probabl
y,probably,3,muy probable
670 DATA altering,altering,alturing
,haltering,2,cambiando
680 DATA vibration,vibration,vibras
ion,vibration,2,movimiento continuo
690 DATA consider,concidur,considur
e,consider,4,tener en cuenta
700 DATA figure,figure,phigure,phi
gre,1,forma de un cuerpo
710 DATA wasteful,wastfull,wastefu
ll,wasteful,4,antieconomico
720 DATA amplifies,amplifies,ampl
ifies,amplifies,2,aumenta el volume
n
730 DATA flashing,flashing,flachs
ing,flashing,4,se enciende y se apa
ga constantemente
740 DATA manual,manual,mannuel,1,hecho a
mano
750 DATA encounter,encountur,encoun
ter,encounter,4,encontrar
760 DATA expreshun,expresion,expres
ion,hexpression,3,palabra o frase
con la se expresa algo
770 DATA content,contant,contente,k
ontenti,1,satisfecho o contenido
780 DATA accumulate,accumulate,acuma
late,accumer,late,2,amontonar
790 DATA assembl,assemble,asembel,
asembel,2,ensamblar o unir
800 DATA situation,situation,situa
sion,situachion,1,posicion en la qu
e se encuentra algo
810 DATA computer,computer,compute
r,computer,1,maquina para el proces
o de datos
820 DATA endless,hendless,endless,en
dless,3,sin final
830 DATA envelope,henvelope,envalop
e,envelope,1,cubierta de papel para
enviar cartas
840 DATA extraction,extrachcion,ext
raktion,extraccion,1,accion de extr
aer
850 DATA forcabley,forcibley,forsab
ley,forstabley,2,a la fuerza
860 DATA hallucination,hallucination
,hallucinatian,hallucination,2,accio
n de alucinar
870 DATA idolize,idolize,idalize,id
alize,2,adorar o idolatrar
880 DATA imposible,impossible,impos
sible,impossible,3,no posible o difi
cil

```

```

890 DATA mechanical,mecanica,mecan
ical,mehanicale,1,trabajo hecho a m
aquina
900 DATA necessary,necessary,necess
ary,necessary,3,que es objeto de ne
cesidad
910 DATA navigation,navigation,navig
ation,navigation,1,viaje realizado
por mar
920 DATA orchid,orkid,orchyd,orcid,
1,variedad de planta exotica
930 DATA oxygen,oxygen,oxygen,oxeju
n,3,gas esencial para la vida
940 DATA spaciou,spachiou,spasiou
,s,spaciou,4,amplio o de gran capac
idad
950 DATA allocation,allocation,alloc
ation,allocation,3,racion o cuota
960 DATA compulsory,compulsary,comp
ulsary,compulsory,1,obligatorio o
que debe ser hecho
970 DATA controller,controller,contr
oller,controlor,3,persona encargada
del control
980 DATA calendar,calender,callenda
r,calendar,1,registro de todos los
dias del ano
990 DATA appreciate,apreciate,aprec
iate,appreciate,1,valorar o aprecia
r
1000 DATA style,styal,stial,stile,1,
estilo
1010 DATA sterilize,steralize,steri
lize,sterilize,4,eliminar los micro
bios
1020 DATA syndicate,syndicate,synde
cate,sindicade,2,asosacion de traba
jadores
1030 DATA temprature,temperature,te
mperature,temperature,2,grado de ca
lor de un cuerpo
1040 DATA hidrogen,hydngen,hydrogen
,hidrogen,3,un componente de nuest
ra atmosfera
1050 DATA bugerigar,bugarigar,budge
rigar,budgerigar,3,perjuicio ave do
medica
1060 DATA administer,adminster,admi
nister,adminstar,1,administrar
1070 DATA pharmec,pharmacyc,pharme
ncy,pharmacy,4,farmacia
1080 DATA possession,posession,poss
ion,posession,1,aquello que poseem
os
1090 DATA retalaete,retaliante,retal

```



```

yate,retalate,2,vengarse o tomar re
presalia
1180 DATA stomach, stomache, stomake,
stomuch,1,viscera del cuerpo humano
1110 DATA rhither, rithem, rhythem, rh
ythm,4,orden acompasado de una seri
e de notas
1120 DATA successful, succesful, succe
ssful,5,realizado con exit
o
1130 DATA substansial, substanshul, s
ubstantiel, substansial,4,considerab
le
1140 DATA disappointing, dissappoint
ing, disappointing, dissappointing,1,de
cepcionante
1150 DATA capital, capitol, kapertal,
kapitol,1,de gran importancia
1160 DATA whippet, whippet, wippet, whi
ppet,2,determinada raza canina
1170 DATA transparent, transparent, t
ransparant, transparent,2,cuero atr
aves del cual se puede ver
1180 DATA necessity, necessity, necesse
ty, necessity,4, carencia que necesit
amos cubrir
1190 DATA asasination, asasination,
assasination, asasination,2, matar
con alevisia
1200 DATA bronkittis, bronkittis, bronc
hitis, bronchitus,3, inflamacion de l
os bronquios
1210 DATA centenary, centenary, sente
nary, sentenary,1, centesimo aniversa
rio
1220 DATA centrifugal, centrifugal, c
entrifugal, centrifugal,3, fuerza cen
trifuga
1230 DATA conventional, conventional,
conventional, conventional,1, que s
e ajusta a una serie de normas
1240 s:=0: REM ** pone a cero el ma
rcador **
1250 CLS:RANDOMIZE TIME
1260 REM ** bucle principal **
1270 WHILE c%<99
1280 FOR p=1 TO 20: REM ** ESCOGE L
AS 20 PREGUNTAS **
1290 v=INT(RND(1)*90+1)
1300 IF b%(x)=0 THEN TEND
1310 b%(x)=1
1320 IF t%(x) THEN TEND
1330 CLS #1:CLS #2: CLS #3:CLS #4:C
LS #5: CLS #6:CLS #7
1340 PRINT #7:PRINT #7:PRINT #7:PRI
NT #7," SUBIR = A"
1350 PRINT #7:PRINT #7:PRINT #7,"
BAJAR = Z"
1360 PRINT#7:PRINT#7:PRINT#7," (Pu
sa Espacio)"
1370 PRINT#7:PRINT#7," Para selecci
onar="
1380 LOCATE (40-LEN(b%(x)))/2,24:PE
N 1:PRINT b%(x)
1390 FOR y=1 TO 4
1400 PRINT #y:PRINT #y:PRINT#y," ";
a%(x,y)
1410 NEXT y

```

```

1420 Z=21z=2
1430 T$=INKEY$
1440 PRINT CHR$(30)
1450 t$=UPPER$(t$)
1460 IF T$="A" THEN zamziz=2:5:PRIN
T CHR$(7)
1470 IF T$="Z" THEN zamziz=2:5:PRIN
T CHR$(7)
1480 IF Z<2 THEN Z=2
1490 IF Z>17 THEN Z=17
1500 IF za0 THEN LOCATE #5,1,za:PR
INT#5," "
1510 IF za0 THEN LOCATE #5,1,za:1:
PRINT#5," "
1520 IF za0 THEN LOCATE #5,1,za:2:
PRINT#5," "
1530 LOCATE #5,2,z:PRINT#5," "
1540 LOCATE #5,1,1
1550 IF t$(">" THEN 1430
1560 IF z=2 THEN c%=1 ELSE IF z=7 T
HEN c%#2 ELSE IF z=12 THEN c%#3 EL
S c%#4
1570 CLS #7:PRINT CHR$(7):LOCATE #7
,4,3
1580 IF c%#a%(x) THEN PRINT #7,SPC(
2):"Correcto!"#s%*1 ELSE PRINT #7,
SPC(1):"Incorrecto!"LOCATE #7,6,6:
PRINT #7," La respuesta es:"#LOCAT
E #7,5,8:PRINT #7,a%(x,a%(x)):c%(p
)=s%
1590 LOCATE #7,3,12:PRINT #7,"Acier
tos!"#%
1600 LOCATE #7,7,15:PRINT #7,"Pulsa
"
1610 LOCATE #7,5,17:PRINT #7,"(Espa
cio)"
1620 LOCATE #7,1,19:PRINT #7,"Para
otra pregunta"
1630 gg$=INKEY$:IF gg$(">" THEN 16
30
1640 t%#
1650 NEXT p
1660 GOSUB 1770:REM ** FIN DEL JUEG
O **
1670 WEND
1680 END
1690 RESTORE 500
1700 FOR x=1 TO 34: READ d
1710 SOUND 5,d,20,15
1720 NEXT x:RETURN
1730 CLS:LOCATE 4,2:PRINT*Adios.
Hasta otra practica."
1740 PEN 2:LOCATE 10,10:PRINT*Acie
rtos!"#%
1750 GOSUB 1690
1760 END
1770 CLS:LOCATE 2,2:PRINT*Al final
del juego tus puntos son "#%
1780 GOSUB 1690:REM ** musica **
1790 PEN 2:LOCATE 3,5:PRINT*Pulsa l
a barra (Espacio) para practicas la
s palabras que necesitas."
1800 sp$=INKEY$:IF sp$(">" THEN 18
00
1810 CLS
1820 FOR p=1 TO 20:IF c%(p)>0 THEN
PRINT a%(c)(p),a%(c)(p))
1830 NEXT p
1840 FOR x=1 TO 20: c%(x)=0: NEXT x
:REM ** inicializar matriz de preg
untas **
1850 LOCATE 5,24:PRINT*Deseas pract
icar otra vez (S/N) ?"
1860 s%=0
1870 gg$=INKEY$:gg$=LOWER$(gg$):IF
gg$="s" THEN CLS:counter=counter+1
1880 IF counter=3 THEN RESTORE 630:
counter=0:FOR x=1 TO 60:b%(x)=0:NEX
T x
1890 IF gg$="n" THEN RETURN
1900 IF gg$="n" THEN CLS:PRINT*Adio
s. Hasta otra practica!"END
1910 GOTD 1870
1920 REM ** opciones **
1930 LOCATE 1,20:PRINT*Pulsa la opc
ion que desees!"#
1940 LOCATE 1,21:PEN 3:PRINT*1. Us

```

```

ar las datos del programa."
1950 PRINT*2. Cargar las datos de
la cinta."#
1960 PRINT*3. Crear nueva serie de
datos."#
1970 gg$=INKEY$:gg$=LOWER$(gg$):IF
gg$("<" OR gg$="3" THEN 1970
1980 IF gg$="1" THEN CLS:PRINT*Por
favor, espera un segundo!"RETURN
1990 IF gg$="2" THEN p%#1:GOSUB 2
030:REM ** Carga las datos de la ci
nta **
2000 IF gg$="3" THEN p%#1:GOSUB 21
40:REM ** Guarda en cinta tus datos
**
2010 RETURN
2020 REM ** Carga las datos de la c
inta **
2030 CLS:PEN 1:LOCATE 4,2:PRINT*Por
favor inserta la cinta"
2040 OPENIN"data"
2050 FOR x=1 TO 60
2060 FOR y=1 TO 4
2070 INPUT #9,a%(x,y)
2080 NEXT y
2090 INPUT #9,a%(x),b%(x)
2100 NEXT x
2110 CLOSEIN
2120 RETURN
2130 REM ** Escribe los datos para
guardarlos en cinta **
2140 FOR x=1 TO 60
2150 CLS:LOCATE 4,2:PRINT*Pregunta
numero "#x
2160 DOW# d%=8
2170 FOR y=1 TO 4
2190 LOCATE 2,y,2+4:PEN 2:PRINT*Res
puesta numero "#y:..:INPUT a%(
x,y)
2200 NEXT y
2210 LOCATE 1,18:PEN 1:PRINT*Qual e
la respuesta correcta (1-4)?:
INPUT a%(x)
2220 IF a%(x)=4 OR a%(x)=1 THEN PRI
NT CHR$(7):LOCATE 1,18:PRINT STRI
NG(70,CHR$(32)):GOTO 2210
2230 PEN 3:PRINT*Qual es la ayuda o
definicion de diccionario "":INPUT
b%(x)
2240 LOCATE 1,25:PRINT*Todo correct
o (S/N) ?"
2250 gg$=INKEY$:gg$=LOWER$(gg$):IF
gg$="s" OR gg$="n" THEN 2260 ELSE 2
250
2260 IF gg$="s" THEN d%#1
2270 CLS
2280 WEND
2290 NEXT x
2300 REM ** GUARDAR LOS DATOS EN CI
NTA **
2310 CLS:PEN 1:LOCATE 4,2:PRINT*Por
favor inserta la cinta para guarda
r los datos"
2320 OPENOUT"data"
2330 FOR x=1 TO 60
2340 FOR y=1 TO 4
2350 PRINT #9,a%(x,y)
2360 NEXT y
2370 INPUT #9,a%(x),b%(x)
2380 NEXT x
2390 CLOSEOUT
2400 CLS:PRINT*Pulsa la (Barra Espa
cio) para practica"
2410 gg$=INKEY$:IF gg$(">" THEN 24
10
2420 CLS:PEN 1:PRINT*Por favor, es
pera un segundo!"
2430 RETURN
2440 REM ** datos del grafico del g
ato **
2450 DATA 32,32,242,243,244,10,0,0
,0,245,246,247

```



P ara que tus de-

no realicen el trabajo duro, M.H. AEDS:
TRAD lo hace por ti. Todos los listados que incluyen
este logotipo se encuentran a tu disposición en un casete
estante mensual. Solicítalo.

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

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.

E

nviamos 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 DÍA: 100 m lisos, salto de longitud, lanzamiento de peso, salto de altura y 400 m lisos.

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

Nos vemos obligados a sustituir, dentro de los juegos que ofrecemos a nuestros suscriptores, el **POLE POSITION** por el **COMBAT LYNX**, debido a que la empresa 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 cual de los 3 juegos prefieren. Muchos gracias.

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



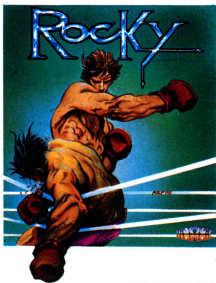
SOLAMENTE
CPC 464

FRANK BRUNO Vs. ROCKY

Siguiendo con la fiebre de juegos deportivos que invaden el mercado, en esta ocasión MR. JOYSTICK se adentra en el mundo del boxeo de la mano de dos programas sensacionales.

El noble arte de las doce cuerdas, conocido en todo el mundo y tachado de inhumano por asociaciones y medios de comunicación, ha dado durante su larga historia grandes campeones: JOHN L. SULLIVAN, MARVIN HART, JACK DEMPSEY, JOE LUIS, ROCKY MARCIANO, FLOYD PATTERSON, SONNY LISTON, CASSIUS CLAY, etc., son nombres que han forjado la historia de este noble deporte, consiguiendo sendos campeonatos del mundo, alcanzando la categoría de míticos para los aficionados de todos los tiempos.

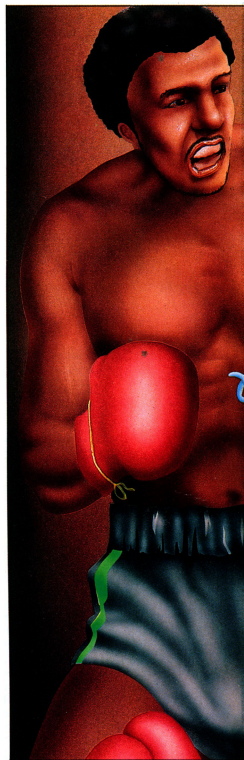
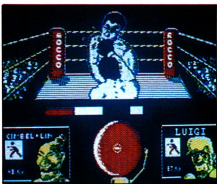
Dos casas de software: una inglesa ELITE, especialista como toda firma de software que se precie, en juegos para SPECTRUM; y otra la intrépida firma nacional DINAMIC (honrosa excepción en la producción española de software ya que es la única), han coincidido en la realización simultánea de un programa de boxeo. Utilizando ambos nombres de grandes boxeadores, que encabezan las cuidadas carátulas de sus juegos.



SOLAMENTE 464

Rocky, ha sido el nombre elegido para la firma hispana y **Frank Bruno**, el protagonista del juego de ELITE.

El boxeo es un deporte duro, de hombres de hierro y que a la larga produce consecuencias irreparables; es acostumbrado oír hablar de boxeadores sonados, ésta y otras causas han tenido como consecuencia el



que este deporte, con la elevación del nivel de vida y el advenimiento de la civilización del ocio, haya perdido un gran número de practicantes.

Ahora, gracias a nuestro ordenador y sentados ante el monitor, podemos vivir la tensión y emoción de los momentos cumbres de un combate de boxeo.

Comenzando con la producción nacional, **Rocky**, sigue la línea del grupo DINAMIC, los gráficos son



francamente extraordinarios, con una técnica de dibujo de cómic, se diría que cuando nuestro boxeador recibe un golpe, es como si nos lo propinasen en nuestra propia cara.

El movimiento es tan real y refleja la acción hasta tal punto, que cuando nos golpean, se observa el gesto de dolor en nuestra cara, vuelta completamente hacia atrás por el golpe encajado.

De igual forma, golpear a nuestro contrincante es un placer, un verda-

dero placer; éste encaja el golpe con verdadera resignación, su cara se vuelve a un lado y sus ojos se cierran en una dramática mueca de dolor, mientras gotas de sudor salen despedidas de su cabeza por los impactos recibidos en pleno rostro.

Izquierda, derecha, izquierda, derecha es el ritmo que llevará a la lona a nuestro adversario, que totalmente grogui no puede defenderse del aluvión de golpes que le cae encima.

Mr. Joystick

El camino que nos lleva hacia el campeonato mundial, está jalonado por adversarios cada vez más técnicos y poderosos, cada nuevo púgil al que nos enfrentamos para mejorar nuestros golpes y lanza sus puños con mayor precisión.

Buenos gráficos de DINAMIC, en su hiperrealista ROCKY.



SOLAMENTE 464

Frank Bruno's boxing, es realmente excitante. ELITE, casa caracterizada por la extrema dificultad de sus juegos; los cuales requieren varios cientos de horas delante del ordenador para ser completados, le ha dado al boxeo un toque especial de emoción y superadición.

El truco reside en que el programa utiliza una cara de la cinta para las rutinas de movimiento, golpes, marcadores, tiempo, etc., y la otra para cargar los distintos adversarios.



Con este sistema de multicarga, revolucionario en los juegos de cinta, se consiguen unos resultados francamente buenos, pudiendo al finalizar una fase de juego con el código obtenido, cargar la fase siguiente, que contiene gráficos distintos y aumenta la aventura en una longitud considerable.

Los resultados conseguidos con este sistema son asombrosos, mientras que en el **Rocky**, nuestro contrario siempre tiene la misma cara, en el **Frank Bruno's**, no solamente cambia la cara, sino que el púgil es totalmente distinto, es más, utiliza distintos golpes y movimientos no ejecutados por los demás, además de técnicas de lucha completamente distintas.

Los contrarios que nos separan de la corona mundial son: **CANADIAN CRUSHER**, **FLING LONG CHOP**, **ANDRA PUNCHEREDOV**, **TRIBAL TROUBLE**, **FRENCHIE**, **RAVIOLI**, **MAFIOSI**, **ANTIPODEAN ANDY** y **PETER PERFECT**.

CANADIAN CRUSHER, es un peso pesado, cuya actividad profesional antes de dedicarse al boxeo era la de leñador, como tal tiene la fuerza de un oso pardo, pero en cambio su lentitud es manifiesta.

Su país de origen es Canadá y como primer rival no ofrece grandes dificultades, **FRANK** es mucho más rápido y ágil; para el oso canadiense es un simple aperitivo, que sirve para desentumecer los músculos y ensayar series de golpes.

El próximo rival es mucho más serio, **FLING LONG CHOP**, es una estrella del boxeo tailandés, no solamente golpea con los puños, sino que también utiliza las piernas (*muy bien por cierto*).

Su golpe más efectivo, es una prodigiosa patada volante, que ejecuta a la perfección y que noquea a **FRANK**, cada vez que le alcanza.

FLING LONG PONG, es rápido y sus series de golpes son certeras y contundentes, es un hueso duro de roer y noquearle requiere una práctica que sólo se logra a base de horas.

Pero si el tío **F. L. CHOP** era duro, **ANDRA PUNCHEREDOV** es una máquina de reparar golpes, su gran esquila y sus prolongadas series de ganchos y directos al rostro, hacen la labor de **FRANK** muy difícil.

El ruso **PUNCHEREDOV**, tiene una técnica depuradísima y es un maestro en el arte de dar cabezados, golpe que ejecuta con una rapidez y contundencia asombrosas.

Un rival muy técnico de golpes duros y gran esgrima boxística.

TRIBAL TROUBLE, el zulu de una lejana tribu de Uganda, con una

guardia impenetrable y un ritmo de puños inigualable, todo un rival para un púgil que se precie.

Los demás rivales que vienen a continuación, poseen un nivel de boxeo inalcanzable y prometen ser rápidos y muy técnicos.

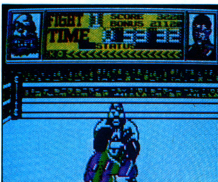
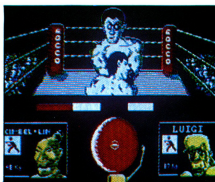
FRENCHIE, **RAVIOLI MAFIOSI**, **ANTIPODEAN ANDY** y **PETER PERFECT**, convierten el boxeo en un verdadero arte, vencerles no está al alcance de cualquiera.

FRANK BRUNO'S BOXING, es un juego ante todo super adictivo, vencer a un púgil para ver cómo es el siguiente y cómo pelea es una verdadera obsesión, los distintos contrincantes hacen que el programa adquiera verdadero interés.

La gama de golpes y movimientos de **FRANK** es muy amplia; guardia arriba, guardia abajo, opercut, directos de izquierda y derecha a la cara y estómago, esquiva a la derecha, esquiva a la izquierda y agacharse.

Todo esto aderezado con un movimiento veloz y unos buenos gráficos, un programa de horas y horas de entretenimiento.

FRANK BRUNO Vs. ROCKY



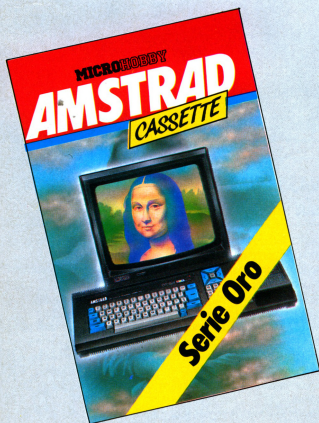
MICROHOBBY

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

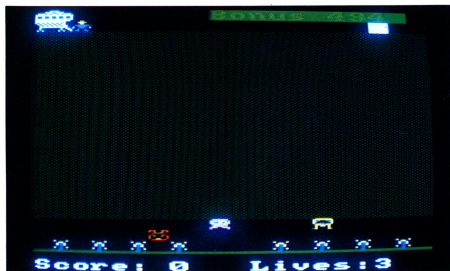
Título	Revista número
EASYDRAW	1
EGGBLITZ	2
CODIGO SECRETO	2
VENTANAS	2
BIORRITMOS	3
MAD ADDER	3
HEXER	3
CHARGEN	4
PROGRAMACCION	4

Enviamos con la menor demora posible, el cupón correspondiente.

DIGGER

Nos encontramos en el espacio profundo. En el camino de regreso a la base número 11 de Alpha Centauri, nuestros generadores de taquiones de pronto se quedan sin energía.

SOLAMENTE
CPC 664



N

os escapamos de permanecer inmóviles en el hiperespacio para siempre por un pelo, la energía que quedaba en los generadores Marsten apenas bastó para saltar al sistema estelar más cercano con un sol tipo G.

Como esperábamos, descubrimos un planeta que, debido a su excéntrica órbita alrededor del Sol doble de aquel sistema, poseía claros indicios de Dilithium, el combustible del que ahora dependían nuestras vidas, pues sin él el retorno a la Tierra a velocidad sublumínica tomaría aproximadamente 15.000 años.

El aterrizaje fue brusco, pero sobrevivimos y fuimos a por el Dilithium a las cavernas donde nuestros sensores lo detectaron.

A sus habitantes, porque tenía habitantes, no parecía importarles demasiado que cogiéramos algunos cristales, siempre y cuando no chocáramos con ellos o nos interpusiéramos en su camino; si lo hacíamos, era la muerte.

No sé si lo conseguiremos.

SUBROUTINAS PRINCIPALES DEL PROGRAMA

LÍNEAS	LO QUE HACEN
250- 420	Asigna las posiciones de comienzo, lee las DATAS y patea en la memoria de pantalla.
430- 620	Define los caracteres para la nave, dimensiona matrices e inicializa algunas variables.
630- 810	Dibuja la pantalla, inicializa la matriz para el lodo y pinta un mensaje si no es la primera pantalla.
810- 900	Mueve el hombre, lee el teclado o el joystick, y mira si hemos cogido un cristal o está siendo cogido.
910-1.010	Mueve los bichos de forma semialeatoria. Mira si el hombre ha sido capturado.
1.020-1.080	Introduce en &A800 una corta rutina en máquina para mover los caracteres.
1.090-1.190	Imprime el mensaje de cogido y borra los caracteres correspondientes.
1.200-1.430	Instrucciones.
1.440-1.610	Fin del juego.

VARIABLES PRINCIPALES DEL PROGRAMA

NOMBRE FUNCION

lives	Vidas.
x%, y%	Coordenadas del hombre.
gotit	Dónde coger un cristal.
ok	Dónde se ha cogido.
i%, i%	Variables contadoras usadas en los bucles.
k%, a\$, address	Variables de uso general. Dirección del carácter en la memoria de pantalla.
mod%(25,21)	Barro.
bug%(2,1)	Posición del bicho.
note%(10)	Música.
hiscore	Máxima puntuación.
chase	Oportunidad para que un bicho nos persiga.
burrow	Posibilidad de que un bicho haga una madriguera.
score%	Puntos.
bonus%	Bonos.
xx%, yy%	Coordenadas temporales del hombre o del bicho.

```

10 REM ***** Digger *****
20 REM *By R.A.Maddilove*
30 MEMORY &AFFF
40 MODE 1
50 GOSUB 1200 :REM instrucciones
60 GOSUB 1020 :REM codigo maquina
70 GOSUB 430 :REM inicializacion
80 MODE 0
90 WHILE aa="S"
100 WHILE lives
110 GOSUB 630 :REM pantalla
120 WHILE collected<(8 AND lives
130 GOSUB 250 :REM comienzo
140 WHILE ok AND collected<(8
150 GOSUB 820 :REM mueve hombre
160 GOSUB 910 :REM mueve bicho
170 WEND
180 IF NOT ok THEN GOSUB 1090 :REM
cogidos
190 WEND
200 WEND
210 GOSUB 1440 :REM fin del juego
220 WEND
230 MODE 1: PEN 1
240 END
250 REM *** comienzo ***
260 LOCATE 12,25:PRINT "Vidas:";USI
NG "":LIVES
270 RESTORE 200
280 DATA 0,20,40,0,0,60,60,0,0,20,4
0,0,84,252,252,168,168,252,252,84,4
0,84,168,20,0,168,84,0,84,168,84,16
8
290 DATA 3,2,1,3,2,2,1,1,0,3,3,0,1,
0,0,2,2,130,65,1,2,0,0,1,2,65,130,1
,1,3,3,2
300 DATA 5,15,15,10,79,143,79,143,7
9,143,15,15,15,15,79,143,10,
5,143,79,10,15,10,5,15,10,0,0,5
310 DATA 179,51,51,115,115,162,81,1
79,115,0,0,179,34,81,162,17,17,115,
179,34,17,17,34,34,17,0,0,34,179,0
,0,115
320 address=C000+50+8:GOSUB 400 :
REM hombre
330 address=C000+20*650+16:GOSUB 4
00 :REM bicho 1
340 address=C000+20*650+36:GOSUB 4
00 :REM bicho 2
350 address=C000+20*650+56:GOSUB 4
00 :REM bicho 3
360 bug<(0,0)=5:bug<(0,1)=21:bug<(1
,0)=18:bug<(1,1)=21:bug<(2,0)=15:bu
g<(2,1)=21
370 x%=3:y%=2:gotit=0:ok=-1
380 LOCATE 8,2:PRINT SPC(10)
390 RETURN
400 REM *** patea datos de caracter
***
410 FOR i%=0 TO 7:FOR j%=0 TO 3:REA
D k%:POKE address:j%+800*i%,k%:NEX
T:NEXT
420 RETURN
430 REM *** inicializacion ***
440 ENT -1,50,10,5
450 SYMBOL 248,3,14,117,255,146,146
,255,255

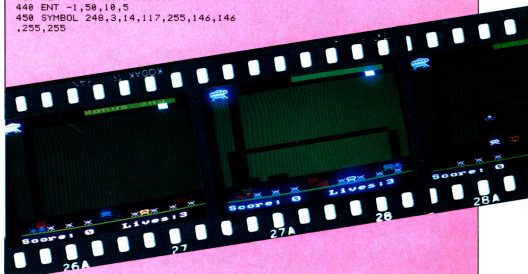
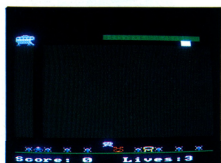
```

Serie Oro

```

460 SYMBOL 249,192,112,174,255,73,7
3,255,254
470 SYMBOL 250,255,170,85,42,63,32,
32,80
480 SYMBOL 251,248,168,88,172,252,2
2,19,41
490 SYMBOL 252,0,1,10,0,109,109,0,0
500 SYMBOL 253,0,128,80,0,182,182,0
,0
510 SYMBOL 254,0,85,42,21,0,0,0,0
520 SYMBOL 255,0,80,160,80,0,0,0,0
530 DATA 0,9,12,0,3,10,16,2,6,15,8,
20,24,13,23,26
540 RESTORE 530:FOR i%=0 TO 15:READ
j%:INK i%,j%:NEXT
550 INK 3,13,26: BORDER 0
560 DIM mud%(25,21),bug<(2,1),note%
(10)
570 DATA 379,319,239,213,190,478,15
9,142,379,119,319
580 FOR i%=0 TO 10:READ note%(i%):N
EXT
590 FOR j%=3 TO 20:mud<(2,j%)=32:NE
XT
600 hi.score=10:score%=0:lives=3:ch
ase=1:burrow=0
610 aa="S"
620 RETURN
630 REM *** pantalla ***
640 IF score<0 THEN LOCATE 4,10:PR
INT CHR$(22):CHR$(1):"FELICIDADES":
CHR$(22):CHR$(0):FOR i%=0 TO 3000:N
EXT i:score%=score+bonus%
650 CLS:RESTORE 660
660 DATA 248,249,8,8,10,250,251,0,8
,11,252,253,8,8,10,254,255
670 LOCATE 1,1: PEN 14:PRINT CHR$(22
):CHR$(1):
680 FOR i%=1 TO 17
690 IF i%=10 THEN PEN 2
700 READ j%:PRINT CHR$(j%):
710 NEXT
720 PEN 1
730 FOR i%=3 TO 20:LOCATE 1,i%:PRIN
T STRING$(20,CHR$(207)):FOR j%=1 TO
20:mod<(i%,j%)=207:NEXT:j%
740 FOR i%=21 TO 23:FOR j%=1 TO 20:
mud<(i%,j%)=32:NEXT:NEXT
750 FOR j%=2 TO 8 STEP 2:LOCATE j%,
23:PRINT CHR$(15):CHR$(3):CHR$(233)
:CHR$(8):CHR$(15):CHR$(4):CHR$(202)
:CHR$(8):CHR$(15):CHR$(5):CHR$(148)
:mod<(23,j%)=202:NEXT
760 FOR j%=13 TO 19 STEP 2:LOCATE j
%,23:PRINT CHR$(15):CHR$(3):CHR$(23

```



```

3)CHR$(8);CHR$(15);CHR$(4);CHR$(20
2);CHR$(8);CHR$(15);CHR$(5);CHR$(14
8);smud$(23,JX)=20;NEXT
778 PEN 1:PRINT STRING$(20,CHR$(200
))CHR$(22);CHR$(8);
788 score%=score%/bonus%;PEN 15:PRI
NT " Ptsai:"score%
798 PEN #1,2:LOCATE #1,10,1:PRINT #
1,"Bonos 500"
808 collected%=:chase=chase-0.1:bu
rrow=burrow*0.1:bonus%=500
818 RETURN
828 REM *** mueve hombre ***
838 bonus%=bonus%/(bonus%):LOCATE
#1,15,1:PRINT #1,bonus%
848 xx%=:INKEY(c%)-1)-(INKEY(d%
))-1:yy%=:INKEY(a%)-1)-(INKEY(b%
))-1
858 char%=:smud$(yy%,xx%):IF char%#0
OR (char%#202 AND got.it) THEN RETU
RN
868 CALL &A800,xx%,yy%,xx%,yy%:xx%
-1:yy%=:smud$(yy%,xx%)-32
878 IF char%#202 THEN got.it=:INK
7,2,1:SOUND 2,38,40,15,0,1
888 IF xx%#3 AND yy%#2 AND got.it#0
THEN SOUND 2,280,280,15,0,1:score%=:sc
ore%+18:LOCATE 7,25:PRINT score%:smud
$(xx%,yy%)=:smud$(xx%,yy%):collected%=:col
lected%+1:INK 7,2
898 IF (xx%#bug$(8,0) AND yy%#bug$(8,
1)) OR (xx%#bug$(1,0) AND yy%#bug$(1,
1)) OR (xx%#bug$(2,0) AND yy%#bug$(2,
1)) THEN ok=0
908 RETURN
918 REM *** mueve bichos ***
928 SOUND 132,note%(INT(RND*11)),20
,14:SOUND 129,478,200,12
938 FOR i%#0 TO 2
948 IF RND%chase THEN xx%#bug$(i%,0
)+:(xx%#bug$(i%,0)-(xx%#bug$(i%,0)))*y
%:(xx%#bug$(i%,1)+:(xx%#bug$(i%,1)-(xx%#b
ug$(i%,1)) ELSE xx%#bug$(i%,0)+INT(
RND*3)-1:yy%#bug$(i%,1)+INT(RND*3)-
1
958 char%=:smud$(yy%,xx%)
968 IF char%#202 OR char%#0 OR char
%#255 OR (char%#207 AND RND%burrow)
THEN RETURN
978 CALL &A800,bug$(i%,0),bug$(i%,1
),xx%,yy%
988 smud$(bug$(i%,1),bug$(i%,0))=:32:
smud$(yy%,xx%)=:255:bug$(i%,0)=xx%:bu
g$(i%,1)=yy%
998 IF xx%#x% AND yy%#y% THEN ok=0
1808 NEXT
1818 RETURN
1828 REM *** codigo maquina ***
1838 RESTORE 1878
1848 FOR i%#0 TO 64
1858 READ a%:POKE &A800+i%,VAL("&"+
a%)
1868 NEXT
1878 DATA DD,46,0D,DD,5E,02,CD,2D,A
B,ES,DD,46,04,DD,5E,06,CD,2D,AB,D1,

```

```

01,00,00,C5,06,04,7E,36,00,12,23,13
,10,F8,01,FC,07,09,EB,09,EB,C1,10,E
B,C9,1D,21,00,C8,55,A7,CB,13,CB,13,
19,05,CE,11,50,00,19,10,FD,C9
1888 RETURN
1898 REM *** cogida ***
1908 SOUND 2,90,50,15,0,1,1
1918 LOCATE 8,2:PRINT "G U L P"
1928 lives=lives-1
1938 LOCATE xx%,yy%:PRINT " "
1948 IF got.it THEN collected%=:col
lected%+1:got.it=:INK 7,2
1958 FOR i%#0 TO 2
1968 LOCATE bug$(i%,0),bug$(i%,1):P
RINT " :smud$(bug$(i%,1),bug$(i%,0)
):#2
1978 NEXT
1988 FOR i%#0 TO 2000:NEXT
1998 RETURN
1208 REM *** instrucciones ***
1218 INK 0,0:INK 1,0:INK 2,15:INK 3
,11
1228 BORDER 0:PLOT -5,-5,3:PEN 1
1238 LOCATE 1,25:PRINT "DIGGER"
1248 FOR i%#0 TO 96 STEP 2
1258 FOR j%#0 TO 16 STEP 2
1268 IF TEST(i%,j%) THEN PLOT 200+i
%*2,365+j%*2:PLOT 200+i%*2,367+j%*2
:PLOT 202+i%*2,365+j%*2:PLOT 202+i%
*2,367+j%*2
1278 NEXT
1288 NEXT
1298 LOCATE 1,25:PRINT SPC(10):INK
1,6
1308 LOCATE 1,5:PEN 2
1318 PRINT "Nuestro Sistema Solar e
sta en peligro. La falta de materi
as energeticas puede provocar su e
xtincion."
1328 PRINT:PRINT "Trata de robar lo
s barriles radioactivos del planeta
Urón sin ser detectado por los Uró
bichos que rondan protegiendo los b
arriles de Uranio."
1338 PRINT:PRINT "Recuerda que debe
s recoger los barriles de uno en u
no y depositarlos en la nave."
1348 PEN 3:PRINT:PRINT:PRINT "Contr
oles joystick o teclado."
1358 PEN 2:PRINT:PRINT "A .... su
bir 2 .... bajar"PRINT " <
.... izquierda > .... derecha"
1368 PEN 1:LOCATE 1,25:PRINT "Puls
a ESPACIO o Fuego para comenzar"
1378 a%=""WHILE INKEY("<")>:WEND
1388 WHILE a%("<") AND JOY(0)<16
1398 a%=:INKEY$
1408 WEND
1418 CLS
1428 IF a%="" THEN a%=:69:bx%=:71:c%=:
39:d%=:31 ELSE a%=:72:bx%=:73:c%=:74:d%=:
75

```

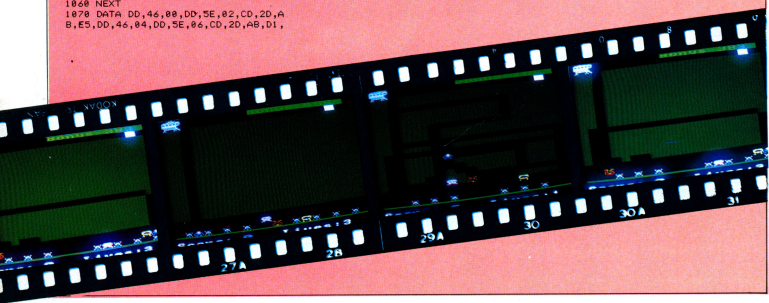
```

1438 RETURN
1448 REM *** game over ***
1458 LOCATE 12,25:PRINT "Vidas:"0"
1468 LOCATE 6,18:PRINT CHR$(22);CHR
$(1);"GAME OVER";CHR$(22);CHR$(0)
1478 FOR i%#0 TO 5000:NEXT:CLS
1488 PAPER 8:PRINT CHR$(30);STRING$
(100," ")
1498 LOCATE 1,25:PRINT STRING$(20,"
")
1508 LOCATE 3,3:PEN 12:PRINT "Puntu
acion Final:"LOCATE 8,4:PRINT:scor
e%
1518 LOCATE 3,25:PRINT "Otro juego
(S/N) ?"
1528 MOVE 0,0:DRAW 0,399,15:DRAW 63
7,399:DRAW 13,0:DRAW 0,0:MOVE 0,32
0:DRAW 639,320
1538 MOVE 0,16:DRAW 639,16
1548 PAPER 0:PEN 11
1558 IF score%>hi.score THEN hi.sc
ore=:score%:LOCATE 3,18:PRINT "Nuevo
record":LOCATE 7,15:PRINT "del dia
" ELSE LOCATE 6,18:PRINT "El recor
d esta en:"LOCATE 8,15:PRINT hi.sc
ore
1568 a%=""
1578 WHILE INSTR("SN",a%)<2
1588 a%=:UPPER(INKEY$)
1598 WEND
1608 score%=:0:lives=:3:chase=:1:burro
w=:0
1618 RETURN

```



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, solicitánselo.



RAMIFICACIONES

Análisis

Esta semana vamos a introducirnos en el mundo de los gráficos con un pequeño, pero sustancioso programa que nos permite dibujar, partiendo de un punto fijo, una serie creciente de ramificaciones, de forma que de cada punto surgen dos ramas que acaban en nuevos puntos, que a su vez constituyen los orígenes de las ramas del nivel superior, formando una retícula de rombos con estructura de árbol.

10-20 Líneas REM que identifican el programa.

30 La variable NIVELMAX, selecciona el número de niveles de que va a componerse nuestro árbol de ramificaciones.

Llamamos nivel, a cada nueva ramificación que parte de un punto. Un vistazo a la figura 1, nos dará una exacta idea de cómo se constituyen éstos y del número de puntos que genera cada nuevo nivel.

40 El comando DIM, es utilizado para inicializar una matriz de dos dimensiones, en la cual van a estar contenidas las posibles coordenadas X de los puntos de ramificación.

Este método supone un gran desplifarro, debido a que no todos los niveles tienen el máximo número de puntos.

Esta matriz, crea espacio para una cantidad elevada de puntos inexistentes.

¿Serías capaz de encontrar algún método para realizar la misma tarea, evitando el mencionado desplifarro de dimensionamiento?

50 Contiene los valores iniciales para los cambios que han de producirse en las coordenadas X e Y de los sucesivos niveles.

La variable ANCHURA, introduce un factor de varianza en la forma en que se dividen las líneas.

Es interesante, observar los efectos producidos al asignarle a ANCHURA, valores distintos de cero.

Otro efecto muy interesante, sería cambiar los valores de SEPARACION X, SEPARACION Y, sobre todo si este cambio se efectúa aleatoriamente.

60 Establece las coordenadas del punto que se encuentra en el nivel 0.

80-140 Constituye un ciclo FOR NEXT, que rota una vez por cada uno de los niveles. Cada vez que las coordenadas de todos los puntos de ramificación en ese nivel deben ser calculadas.

90-130 Conforman un segundo ciclo, anidado en el anterior, que realiza la labor de calcular las coordenadas de los puntos y dibujar las líneas que los unen.

La figura 1, nos muestre que el número de puntos que constituye cada nuevo nivel, es el cuadrado del número de estos contenidos en el inmediato inferior.

100 Se ocupa de asegurar que el cursor gráfico se encuentra en el punto adecuado, para realizar el trazado de las líneas. Obsérvese lo que ocurre al omitir esta línea, y comprenderemos su importancia.

110 Trazo la línea que se bifurca hacia la izquierda del punto.

En la figura 2, podemos observar que todos los puntos impares, están situados en la parte izquierda de la bifurcación; de esta manera, mediante el comando MOD calculamos si el punto es par o impar.

En caso de que éste sea impar, el resto de la línea procede a dibujar la recta que une el

punto de ramificación del nivel inferior con el punto del nivel en que nos encontramos.

120 Realiza los mismos cálculos y tanteos que la línea anterior, pero esta vez con las líneas que bifurcan hacia la derecha.

La figura 2, de nuevo nos muestra cómo se calculan las coordenadas de los puntos del nivel superior en función de las del punto situado en un nivel inferior; solamente es cuestión de sumar y restar los valores adecuados.

150 Un ciclo sin fin, que mantiene en pantalla el dibujo de nuestro árbol, sin esta línea la figura desaparecería de nuestra vista inmediata.

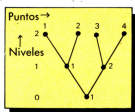


Figura I: Puntos y niveles.

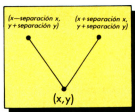


Figura II: Coordenadas de los puntos.



LOS COLORES EN EL AMSTRAD (I)

El Amstrad es uno de los ordenadores que posee posibilidades gráficas más amplias y potentes dentro de las máquinas de su rango.

27 colores y un variado juego de comandos gráficos están esperándonos para mejorar el aspecto de nuestros programas.

Si alguna vez hemos empleado software comercial, especialmente juegos, en el **Amstrad**, probablemente hemos visto con asombro los gráficos que estos programas utilizan, y seguro que ha aparecido el deseo de incorporar algo parecido en nuestros propios programas.

Sin embargo, la gran cantidad de comandos disponibles en el Basic Locomotive para este fin, hacen que la tarea no sea tan simple, porque no son fáciles de usar.

Por tanto, vamos a intentar subsanar esta dificultad introduciéndonos en el mundo de la programación a color de una manera más detallada y amigable que la que **Amstrad** usa en sus manuales.

Se supondrá que el lector posee un pequeño conocimiento del lenguaje Basic. Si no es así, no hay que preocuparse. El contenido del artículo podrá captarse perfectamente.

Incluso si no se tiene monitor en color, las técnicas descritas aquí pueden aprovecharse, en primer lugar, para aprenderlas, y en segundo lugar, cada color corresponderá a una intensidad de tono de verde distinta, por lo que es igualmente identificable.

En este caso, el peor de todos, sólo hace falta un poco de imaginación para asociar determinado tono de verde con algunos de los colores. Por

desgracia, no todos son fácilmente reconocibles en un monitor monocromo.

Cada modo implica un cierto número de colores

Así que vamos a comenzar por el principio. Como muchos ya conocen, el **Amstrad** posee tres modos de pantalla, cada uno de los cuales se diferencia de sus parientes en la dimensión horizontal que adquiere la pantalla de texto, esto es, el número de caracteres escritos que caben en una línea.

Si tecleamos:

MODE 0

y llenamos una línea de la pantalla con letras, observamos que caben exactamente 20.

Los comandos MODE 1 y MODE 2 aumentan la «extensión» de la línea a 40 y 80 caracteres respectivamente. Obsérvese también que el tamaño de los símbolos varía de mayor a menor y que el MODE 1 es el que el **Amstrad** posee por defecto al encenderlo.

La magnitud y número de caracteres no es la única diferencia entre los modos de pantalla. La fundamental, en lo que a este artículo respecta, es que TAMBIÉN SE DISTINGUE EN EL NÚMERO DE COLORES QUE PODEMOS UTILIZAR SIMULTANEAMENTE EN PANTALLA.

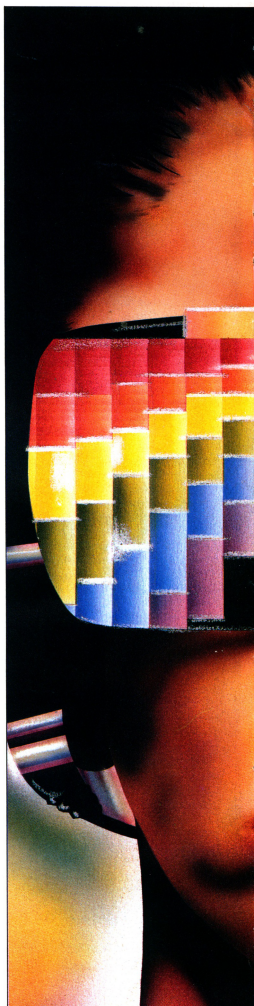
El modo 0 permite 16 colores, el 1, 4 colores y el 2 solamente 2.

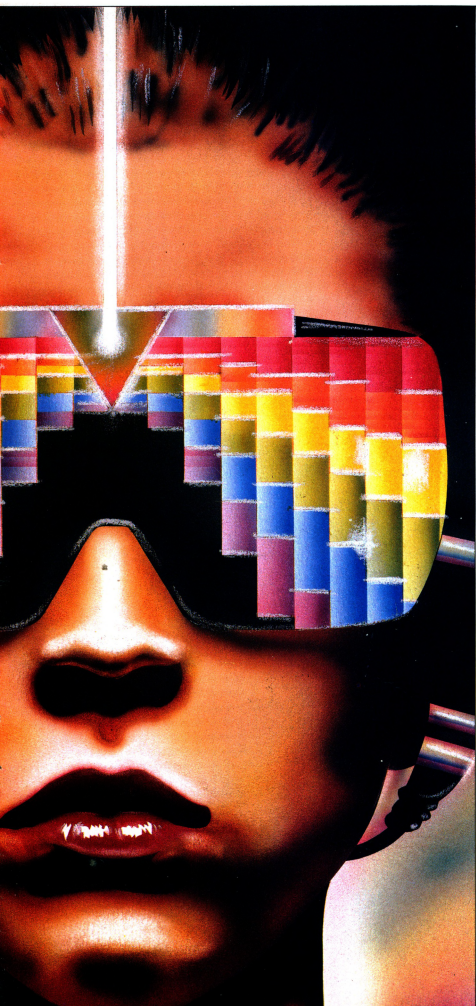
La relación resulta clara: a medida que aumentan los caracteres por línea, el número de colores disminuye.

Tabla 1

CARACTERÍSTICAS DE LOS DISTINTOS MODOS

Modo	N.º carac.	N.º colores
0	20	16
1	40	4
2	80	2





ProgramAcción

Pensando un poco, esta regla adquiere sentido inmediatamente. Basta recordar que, en los tres modos, LA CANTIDAD DE MEMORIA ASIGNADA A LA PANTALLA ES LA MISMA, 16 KBYTES.

Como el **Amstrad** codifica la imagen de la pantalla y los colores asociados en ella, el espacio para almacenar datos relativos a colores disminuirá si reclamamos más memoria para especificar caracteres. La Tabla número 1 resume todo esto.

En este momento, el ordenador debe estar en MODE 1. Si no es así, por favor ponedlo.

Dos colores como mínimo

Obviamente, el mínimo número de colores que debe existir en cualquier modo de pantalla es dos, porque de lo contrario, con un solo color, no distinguiríamos un ápice: siempre tendrá que haber un color de fondo y uno de primer plano, o lo que es lo mismo, UN COLOR DE PAPEL y UN COLOR DE TINTA.

Siguiendo con la analogía de una hoja de papel normal y corriente en lugar de un monitor, para escribir con una tinta determinada necesitamos un tintero y una pluma.

El **Amstrad Basic** tiene justamente estas palabras: papel, pluma y tinta, como palabras reservadas, con idea de hacer el manejo de colores lo más mnemotécnico posible, sólo que en inglés:

PAPER para papel
PEN para pluma
INK para tinta

En el momento de encender el **Amstrad** estamos escribiendo con una pluma mojada en tinta amarilla sobre papel azul.

Si tecleamos:

PEN 2

le estamos diciendo al micro que queremos mojar la pluma en otro tintero, y efectivamente los caracteres aparecen en cyan (*azul*), pero el papel, el color de fondo, no ha cambiado.

Existen más tinteros; por ejemplo:

PEN 3

escribirá en rojo, mientras que:

PEN 1

devolverá el color de tinta a amarillo, su color original.

Parece ser que al menos podemos manejar 3 tinteros distintos, ¿verdad?

Pues no, hay 4. Recuerdese que los ordenadores comienzan a contar con el 0, así que también podemos teclar.

PEN 0

Desgraciadamente, este tintero contiene azul, por lo que no veremos una sola palabra por mucho que escribamos. Teclee ENTER para conseguir una nueva línea y luego, cuidadosamente, PEN 1 para llamar a la tinta amarilla (si no funciona, no se pare en barras; resetee el Amstrad y listo).

¿Y qué sucede si no me creo que sólo existan 4 tinteros y tecleo PEN 4, ó 5 ó 6 ó 7?

Bueno, descubriremos que PEN 4 actúa exactamente igual que PEN 0, PEN 5 que PEN 1 y así sucesivamente hasta PEN 7 que corresponde a PEN 4.

Lo que sucede es que en MODE 1 sólo se permiten 4 colores o tintas a un tiempo en la pantalla, por lo que PEN es convertido por el Amstrad en PEN 0, al igual que PEN 8.

En MODE 1, dado un número de pluma, como sólo hay 4 colores disponibles, el número real que el Amstrad obedece es el resto de dividirlo por 4 (asumiendo que el número sea mayor que 3).

Por ejemplo, PEN 13 se convierte en PEN 1 (13/4 da de resto 1).

La tabla número 2 especifica los colores asignados a cada tintero en MODE 1. Para estar seguros de acertar siempre, basta con emplear números del 0 al 3.

El programa número 1 muestra todos los colores disponibles en este modo, excepto PEN 0 que sería invisible.

Ahora bien, si cambiamos la línea 20 del programa 1 por:

20 MODE 2

y lo ejecutamos, ¡menudo caos!

¿Qué sucede con PEN 2, y por qué PEN 3 es amarilla? Se supone que era roja.

Efectivamente, MODE 2 sólo usa dos colores. No cabe duda.

PEN 0 da azul brillante, y PEN 1 amarillo brillante.

Cualquier número superior revienta a los valores primitivos: PEN 3, por ejemplo, se comporta como el número 0. La Tabla número 3 muestra los colores y sus números asociados en este modo.

Al modificar de nuevo la línea 20 y escribir:

20 MODE 0

parece haber poca diferencia respecto al MODE 2. No olvidemos, sin embargo, que en MODE 0 disponemos de 16 colores, de 0 a 15. El programa número 2 los muestra, así como la Tabla 4 indica la relación entre color y número de pluma.

También podemos cambiar con facilidad el color de fondo de la pantalla, esto es, el PAPEL, mediante el comando PAPER.

Cambiando el color del papel y del borde

Con el Amstrad en MODE 1, si tecleamos:

PAPER 3

el papel se vuelve rojo, mientras que las letras permanecen en amarillo.

La sentencia PAPER 3 quiere decir:

«EL COLOR DE FONDO SERA EL MISMO QUE TIENE LA TINTA USANDO EL TINTERO NUMERO 3, ES DECIR, PEN 3».

Como puede verse en la Tabla 2, PEN 3 es rojo brillante, por lo que el papel también es rojo brillante.

De la misma forma y según esta ta-

bla, PAPER 2 colocará el fondo de color cyan brillante.

La escritura con esta combinación de color es bastante ilegible, así que vamos a cambiar la tinta a rojo (¿recordáis cómo?):

PEN 3

El color del papel es muy fácil de manejar: trabaja exactamente igual que PEN, con las mismas restricciones explicadas antes propias de cada modo.

Tabla 4

COLORES DE PLUMA POR DEFECTO EN MODO 0	
N.º de pluma	Color
0	Azul brillante
1	Amarillo brillante
2	Cyan brillante
3	Rojo brillante
4	Blanco brillante
5	Negro
6	Azul brillante
7	Magenta brillante
8	Cyan
9	Amarillo
10	Azul pastel
11	Rosa
12	Verde brillante
13	Verde pastel
14	Parpadeo entre azul/amarillo brillante
15	Parpadeo entre rosa/azul cielo

Tabla 2

COLORES POR DEFECTO EN MODO 1	
N.º de pluma (PEN)	Color
0	Azul brillante
1	Amarillo brillante
2	Cyan brillante
3	Rojo brillante

Tabla 3

COLORES DE PLUMA POR DEFECTO EN MODO 2	
N.º de pluma	Color
0	Azul brillante
1	Amarillo brillante

Insistimos que la regla de oro es: PAPER N pondrá de color de fondo el mismo color de tinta PEN N.

Cuando cambiemos el color del papel, sólo los caracteres que escribamos a partir de ese momento se ven afectados, el resto de la pantalla permanece en el color de fondo anterior.

Sin embargo, al llegar a la última línea, todo el papel cambia al nuevo color escogido.

Por supuesto que hay formas más rápidas de conseguir que toda la pantalla adopte un solo color, mediante la secuencia de comandos:

PAPER 1: CLS

y el color de fondo pasará a amarillo (PAPER 1), manteniéndose rojo en primer plano (todavía estamos en PEN 3).

La clave, como sin duda sospecharéis, es el comando CLS.

El programa número 3 ilustra cómo trabajan las distintas combinaciones de pluma y papel.

PROGRAMAS

```
10 REM PROGRAMA I
20 MODE 1
30 PEN 1
40 PRINT "Este es el PEN 1"
50 PEN 2
60 PRINT "Este es el PEN 2"
70 PEN 3
80 PRINT "Este es el PEN 3"
```

```
10 REM PROGRAMA II
20 MODE 0
30 FOR color = 0 TO 15
40 PEN color
50 PRINT "Este es el color ";color
60 NEXT color
```

```
10 REM PROGRAMA III
20 MODE 0
30 FOR fondo = 0 TO 15
40 PAPER fondo
50 CLS
60 PRINT"Este es el PAPER"; fondo
70 PRINT
80 FOR color = 0 TO 15
90 PEN color
100 PRINT "Este es el color";color
110 NEXT color
120 PRINT"Pulsa una tecla"
130 delay$ = INKEY$ : IF delay$=""
THEN GOTO 130
140 NEXT fondo
150 PAPER 0 : PEN 1
```

```
10 REM PROGRAMA IV
20 MODE 1
30 FOR color = 0 TO 26
40 BORDER color
50 LOCATE 16,12
60 PRINT "BORBER "; color
70 FOR pausa = 0 TO 500 :NEXT pausa
80 NEXT color
```

Program Acción

Aunque hasta ahora sólo hayamos examinado 16 colores de los 27 prometidos, el programa 4 muestra la totalidad de ellos en el borde de la pantalla, de donde se deduce que BORDER es el comando que altera el color de esa zona (*naturalmente*).

En la tabla número 5 pueden verse los números correspondientes a los 27 colores de **Amstrad**. Producirán el color especificado precedidos del comando BORDER. El ordenador ya no se comporta como en el caso de la pluma, porque el número que sigue al comando PEN etiqueta el número de tintero al que nos estemos refiriendo, no la tinta con la que dicho tintero se llenará.

Próximamente aprenderemos cómo asignar a cada pluma el color de tinta que deseemos, pero de momento creemos que el lector estará lo suficientemente ocupado experimentando con el material que le damos aquí.

Tabla 5

COLORES DE TINTA	
N.º de tinta	Color
0	Negro
1	Azul
3	Azul brillante
4	Rojo
5	Magenta
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

Los REGISTROS RESTANTES DEL Z80

Hasta ahora nos hemos concentrado en el estudio del registro acumulador, tal vez el más versátil e importante del Z80.

Sin embargo, existen otros registros dentro del microprocesador que sirven de ayuda y complemento al acumulador e incluso se utilizan en determinadas operaciones que aquél no sería capaz de llevar a cabo.



Los registros que nos interesan son los llamados: B, C, D, E, H y L. Existen otros, como el F y el **juego de registros alternativos**, cuyos nombres son análogos a los anteriores, pero se les añade un apóstrofe: H', B', etc. que de momento no nos conciernen.

De paso, aprovechamos para comentar que no hay más registros (*tranquilos*).

Todos ellos sólo pueden almacenar un byte a un tiempo, de la misma manera que el acumulador.

Por tanto, las instrucciones:

LD B,&2A
LD H,&0C

como cabría esperar, existen.

Vamos a emplear un pequeño convenio para explicar las instrucciones que atañen a estos registros, ahorrándonos tener que escribirlos uno por uno en pantalla continuamente.

La operación de carga puede representarse como:

LD r,n

en donde LD significa cargar (*en inglés LoaD*) como ya sabemos, r sustituye a cualquiera de los registros A, B, C, D, E, H, L y n significa un número entre 0 y 255, es decir, un byte.

La tabla número 1 muestra los códigos de operación para las operaciones de carga de todos los registros, incluido el ya conocido LD A,N.

Así, de dicha tabla se deduce que:

LD B,&2A

numéricamente sería:

06 2A (en hexadecimal)

mientras que:

LD H,&0C

se convierte en:

26 0C

Perfecto. Ya sabemos cómo cargar los diferentes registros. Y ahora, **¿para qué demonios nos sirven?**

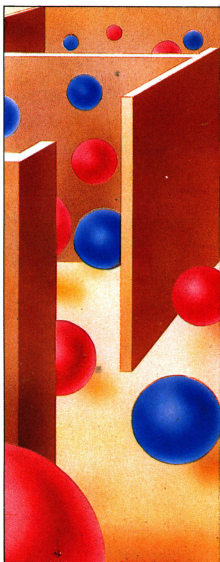
Los registros sustituyen a las variables Basic

Los que hayan usado el Basic para hacer programas recordarán la importancia que poseen las variables, por ejemplo. En código máquina no podemos permitirnos tales lujos, pero sí es relativamente sencillo usar los registros como variables un tanto rudimentarias.

Aunque cada registro sólo pueda almacenar un byte, se pueden obtener resultados muy satisfactorios. Recordad: incluso el **Amstrad Basic** está escrito en lenguaje máquina.

Como un ejemplo de lo antedicho, vamos a intentar colocar el cursor de texto en una fila y columna predefinidas e imprimir un asterisco allí.

Una vez más, tenemos la suerte de poder recurrir al firmware y usar una rutina que ya está escrita, capaz de



colocar el cursor de texto en cualquier lugar de la pantalla.

COMIENZA EN LA DIRECCION &BB75, y la llamaremos PostCur, posición del cursor de texto.

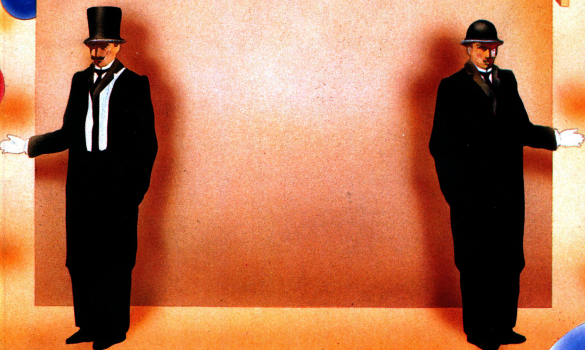
Para poder usarla, debemos CARGAR EL REGISTRO H con la columna que queremos y el REGISTRO L con la fila.

Una vez hecho esto, simplemente llamamos a PostCur.

Como recordareis, para pintar el asterisco todo lo que tenemos que hacer es cargar el acumulador con su código ASCII y llamar a otra rutina del firmware, nuestra vieja conocida PRINT (*dirección &BB5A*). Ya que hablamos de rutinas, no estaría de más borrar antes la pantalla para observar fácilmente si el programa completo funciona según lo previsto. Para ello, llamamos a la rutina ubicada en la dirección &BB6C, denominándola BorraText.

Por favor, introducir y ejecutar el siguiente programa mediante el Hexer:

Código máquina



Dirección	Opcode	Mnemónico
3006	CD 6C BB	CALL Borra Text
3003	26 14	LD H, &14
3005	2E 0C	LD L, &0C
3007	CD 75 BB	CALL PosTCur
300A	3E 2A	LD A, &2A
300C	CD SA BB	CALL Print
300F	C9	RET

Los registros H y L han sido usados clarísimamente como «variables» para almacenar los valores de columna y fila de la posición del cursor. Sería interesante alterarlos para observar el efecto y comprender cómo actúa el programa.

El porqué de la importancia del acumulador

Como vimos anteriormente, el siguiente programa funciona perfectamente con el acumulador:

```
LD A,n
LD (&2FF8), A
RET
```

DESGRACIADAMENTE, CON EL RESTO DE LOS REGISTROS ESTO NO FUNCIONA. Este programa:

```
LD B,n
LD (&2FF8), B
RET
```

falla porque la instrucción:

```
LD (&2FF8), B
```

no existe excepto en el caso del acumulador.

Análogamente:

```
LD B, (&2FF8)
```

también es ilegal.

Hay que adoptar una técnica de «rodeo» para poder escribir y leer de la memoria a través de registros diferentes al acumulador:

```
LD B,n
LD A,B
LD (&2FF8), A
RET
```

Donde podemos observar una instrucción nueva:

```
LD A,B
```

que permite cargar el acumulador con el contenido del registro B.

De hecho, esto no es más que uno de los códigos de operación que representan la operación de cargar un registro con el contenido de otro. Hay unos cuantos de ellos, como puede verse inspeccionando la tabla número 2. Según nuestro convenio, todos estos códigos de operación responden a la forma:

```
LD R,R,'
```

en donde ambos R y R', pueden ser cualquiera de los registros B, C, D, E, H y L.

Al usar la Tabla 2, recuérdese que R se obtiene de las filas y R' de las columnas; así, LD B,C tiene 41 hexadecimal como código de operación.

¡Aún no hemos terminado con esta tabla! Quedan por decir dos cosas:

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

Tabla 1: Opcodes del grupo de instrucciones LD r,n.

		r'						
		B	C	D	E	H	L	A
r	B	40	41	42	43	44	45	47
	C	48	49	4A	4B	4C	4D	4F
	D	50	51	52	53	54	55	57
	E	58	59	5A	5B	5C	5D	5F
	H	60	61	62	63	64	65	67
	L	68	69	6A	6B	6C	6D	6F
	A	78	79	7A	7B	7C	7D	7F

Tabla 2: Opcodes para LD r,r'.

1. Existe una relación entre estos códigos de operación, van seguidos.

2. Entre los registros L y A ha «desaparecido» una fila y una columna, que debería estar para seguir la misma relación que los anteriores códigos. Veremos lo que esto significa más tarde.

Esta relación existe entre muchos grupos de códigos de operación y ayudan al Z80 y al programador (antes de los ensambladores) a averiguar qué es lo que se pretende hacer (para los interesados en esta cuestión, recomendamos observar los opcodes de la Tabla 2 EN BINA-RIO).

Cargar un registro con otro, no altera a éste

Antes de que se nos quede en el tintero, es esencial comprender que las instrucciones tipo:

LD R,R'

HACEN UNA COPIA DEL CONTENIDO DE UN REGISTRO EN OTRO y no una transferencia, esto es, el contenido de R' se copia en R, pero R' mantiene su valor sin cambios.

Para demostrarlo, ejecútense el siguiente programa que posee la modesta pretensión de pintar un asterisco en pantalla:

Dirección	Opcode	Mnemónico
3000	26 2A	LD H,&2A
3002	6C	LD L,H
3003	5D	LD E,L
3004	53	LD D,E
3005	4A	LD C,D
3006	41	LD B,C
3007	78	LD A,B
3008	CD 5A BB	CALL PRINTOut
300B	C9	RET

Obsérvese el «largo viaje» que sigue el número &2A (ASCII DEL «»), desde el registro H hasta el acumulador, pasando por todos los demás. Y sin embargo, llega tarde tal vez, pero llega a su destino final: la pantalla, como debe ser.

Podemos hacer muchas más cosas con los registros, ya puestos.

Por ejemplo, nada más fácil que INCREMENTAR y/o DECREMENTAR sus contenidos, mediante los juegos de instrucciones:

INC R
DEC R

en donde R representa la habitual serie de registros.

Ver (¡sil otra tabla) si no la Tabla número 3.

Para ilustrar su uso, podemos incluirlos en los programas de este artículo, intentando que aparezcan dos asteriscos en la pantalla, uno encima del otro a poder ser.

La primera idea que nos viene a la mente es muy lógica: puesto que el registro L almacena la fila de la posición del cursor de texto, pintamos un asterisco en pantalla, INCREMENTAMOS L y volvemos a llamar a la rutina CurTPos y a PRINT, y ya está, como en este programa:

```
CALL Borrtxt
LD H,&12
LD L,&0C
CALL PosCur
LD A,&2A
CALL PRINTOut
INC L
CALL PosCur
CALL PRINTOut
RET
```

Este sombrío prelude probablemente haya hecho sospechar a más de uno que la cosa no es tan fácil (¡BINGO!).

En efecto, resulta que la rutina CurTPos «corrompe», altera el valor de los registros H y L. Cuando acaba con ellos, su contenido no se parece ni remotamente al que tenían antes de llamarla, por lo que el segundo asterisco vaya usted a saber dónde aparecerá. Así que ejecutar alegremente el programa anterior sería algo peligroso. Si algún espíritu aventurero lo ha hecho y ha obtenido como premio un magnífico «cuelgue» del Amstrad, bueno, gajes del oficio. La vida del programador es dura.

Si por el contrario, lo hizo y no pasó nada, compre lotería o algo en el quiosco más próximo. La suerte no llama dos veces a la puerta en el mismo siglo.

r	INC r	DEC r
B	04	05
C	0C	0D
D	14	15
E	1C	1D
H	24	25
L	2C	2D
A	3C	3D

Tabla 3: Opcodes de INC r y DEC r.

Bromas aparte, la rutina CurTPos «preserva» los registros B, C, D y E, pero corrompe el acumulador H y L.

El método a seguir, por tanto, es preservar nosotros el contenido de estos registros en otros para luego recuperarlos cuando necesitemos sus valores para fijar la posición del cursor.

Así, B preservará a H, C a L y D a A, obteniendo este resultado:

Dirección	Opcode	Mnemónico
3000	CD 6C BB	CALL Borrtxt
3003	26 12	LD H,&12
3005	44	LD B,H
3006	2E 0C	LD L,&0C
3008	4D	LD C,L
3009	CD 75 BB	CALL PosCur
300C	3E 2A	LD A,&2A
300E	57	LD D,A
300F	CD 5A BB	CALL PRINTOut
3012	60	LD H,B
3013	69	LD C,C
3014	2C	INC L
3015	CD 75 BB	CALL PosCur
3018	7A	LD A,D
3019	CD 5A BB	CALL PRINTOut
301C	C9	RET

El programa funcionará, aunque más tarde analizaremos formas más convenientes de preservar de todo daño los registros que deseemos. Como ejercicio, ¿por qué no intentar imprimir 3 asteriscos?

Instrucciones aritméticas

Ya vimos antes que podemos sumar y restar números del acumulador. Lamentablemente, las instrucciones:

ADD B,B
SUB, C,4

no existen.

Sin embargo, SI EXISTEN:

ADD A,R
SUB R

en donde R, una vez más, puede ser A, B, C, D, E, H, L.

Obsérvese que, en la resta, sólo necesitamos especificar el registro donde está el número que queremos restar. El otro registro se sobreentiende que es el A.

Los códigos de operación de estas instrucciones se encuentran en la Tabla 4, y harán falta para resolver los problemas 1 al 3 (junto con la Tabla

5), en donde se trata de predecir el resultado que se obtendrá en la posición &2FF8 al ejecutar los programas. Hexer permitirá inspeccionarla con facilidad.

r	ADD A,r	SUB r
B	80	90
C	81	91
D	82	92
E	83	93
H	84	94
L	85	95
A	87	97

Tabla 4: Opcodes de las instrucciones ADD A,r y SUB r.

Mnemonic	Opcode
CALL pq	CD ap
RET pq	C9
LD A, [pq]	3A q p
LD [pq], A	32 ap

Tabla 5: Más opcodes.

Por vez primera se da un listado exclusivamente en lenguaje ensamblador, sin los opcodes. Esta labor tendrá que realizarse a mano y, para ello, sugerimos los siguientes pasos:

1. Escribir los opcodes en papel antes de introducirlos con el Hexer.
2. Asegurarse de que hemos escrito números hexadecimales.
3. Cuidar de que el código termine con &C9, esto es, la instrucción RETorno al programa principal o, en este caso, al Hexer.
4. Ojo con la escritura. Resulta fácil confundir 8 con B, por ejemplo.
5. Las direcciones de memoria deben escribirse en el orden que el Z80 espera encontrar, primero el byte bajo y luego el alto. Por ejemplo, &2FF8 sería PRIMERO F8 y DESPUES 2F.
6. Vuelva a repetir todos estos pasos antes de ejecutar el código máquina. ¡Va en serio!
7. Elija como dirección de comienzo para ubicar el código máquina &3000. Si decide emplear otra, notifíquese al Hexer cuando lo pague.

Después de estudiar a fondo los ejemplos y ejercicios propuestos en este artículo, estamos preparados para manejar números de dos bytes usando para ello los registros por pares. La próxima semana veremos cómo hacerlo. Buena caza.

Problema I	Problema II	Problema III
LD B,801 LD A,801 ADD A,B LD (&2FF8),A RET	LD C,81 LD A,&F SUB C LD (&2FF8),A RET	LD D,&FF LD A,&2 ADD A,D LD (&2FF8),A RET

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!

¿Cómo puedo rellenar figuras de color distinto al del papel?
David Corredor / Alicante.

Suponemos que posees un **Amstrad CPC464**, porque el 664 tiene en su juego de comandos uno que permite rellenar cualquier área del color de tinta especificado: el comando FILL.

En el caso del 464 el asunto podría ser un poco más complejo, pero si observas el programa EASYDRAW del número uno de la revista, en la página 20, verás que el cuadro de subrutinas se indica una capaz de rellenar áreas con color, que comienza en la línea 2330 y termina en la línea 2520.

No te resultará muy difícil traspasar el algoritmo a lenguaje máquina siempre y cuando tengas en tus manos el manual de firmware de **Amstrad**. Por razones de espacio, no podemos exponer la rutina completa en lenguaje máquina.

Preguntas:

1. Para solicitar la cassette de la «Serie Oro» ¿publicarán cupón de pedido, o por carta?
2. El tipo de escritura en la impresora utilizada, ¿no podría cambiarse? Los símbolos se confunden y la lectura es muy pesada.
3. Sería interesante añadir comentarios sobre la acción de cada llamada «CALL», cada vez que aparecen en un listado.

Abelardo Sánchez.

Vamos a intentar responder a tus preguntas por turno:

1. El cupón que menciona se encuentra en todos los números de la revista, encartado en la última página.
- Lleva el siguiente encabezamiento: «SOLICITUD DE CINTAS DE PROGRAMAS Y NUMEROS ATRASADOS».
2. Suponemos que te refieres al listado del EASYDRAW del número 1 de la revista. En dicho programa nos vimos obligados a utilizar un tipo de letra algo menuda, por razones de espacio. Intentaremos, en la medida de lo posible, que los listados aparezcan al mayor tamaño que podamos.
3. Agradecemos tu sugerencia, y procuraremos incluir dicho comentario en aquellos programas en los que sea posible.

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!

GANAR 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

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

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.
- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:
 - Descripción general del programa.
 - Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
 - Instrucciones de manejo.

¡ENVÍANOS TU PROGRAMA!
a **HOBBY PRESS, S. A. La Granja, s/n. Pol. Ind. Alcobendas (Madrid)**

Desearía contactar con usuarios de **Amstrad** para formar un gran club, preferentemente de Huesca o provincia. Mis señas son: Alberto Carrera Martín. Tarbes, 1 - 4.º C. 22005 Huesca. Tel. (974) 24 24 49.

Se intercambian juegos y utilidades para el **Amstrad CPC 464**. Llamar interesado al teléfono (91) 734 25 89. Preguntar por Nabor.

Cambiaría monitor fósforo verde por uno a color, pagando una diferencia. (El monitor fósforo verde es el GT64.) Interesados llamar al teléfono (971) 41 54 48. Preguntar por Miguel o escribir a: Reyes Católicos, 172 - 4.º. Palma de Mallorca.

Vendo un **JOYSTICK QUICKSHOT II** en perfecto estado y cambio juegos de importación.

Escribir a Luis Miguel Calonge de Marco. Duquesa de Villahermosa, 31 - 6.º A. 50010 Zaragoza. O bien llamar al teléfono (976) 35 13 26.

Vendo ordenador **ATARI 600XL** + Data Cassette + 2 Manuales Basic + Mapa Memoria + Curso en Cassettes (*muy completo*). Todo por 60.000 ptas. José Víctor González. Gral. Franco, 47 - 7.º Izq. Logroño (*Rioja*). Tel. (941) 23 32 10.

Vendo sintetizador de voz **DKTRONICS** completo (*interface, altavoces, programa e instrucciones*) en perfecto estado. Tel. (968) 23 49 88. Preguntar por Alvaro.

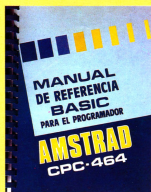
Desearía contactar con usuarios de **Amstrad** para intercambio de programas. Llamar de 10 a 11 h. de la noche o escribir a Julián Calero. Cataluña, 16 - 5.º Izq. Basauri (*Vizcaya*). Tel. (94) 440 46 88.

Desearía contactar con usuarios del 464/664 para intercambio de información, programas e ideas. Llamar al teléfono (91) 276 53 49. Madrid. De 2,30 a 5,30 h. y de 10,30 h. en adelante. Preguntar por Pedro.

Cambio o compro programas para **CPC 464**. Interesados escribir a David Corredor. Avda. Mediterráneo. Edificio Ilice. Benidorm (*Alicante*).

Me interesaría ponerme en contacto con usuarios **Amstrad**, con el fin de intercambiar información, comentar programas, etc. Escribir a José Antonio Correa. General Dávila, 38 - 9.º - 8. Gijón (*Asturias*).

LIBROS EN CASTELLANO PARA TU AMSTRAD



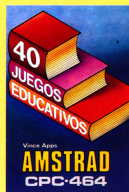
Manual de Referencia Basic para el Programador
La más autorizada y completa guía para programar en Locomotive Basic.
3.400.— Pts.



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



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



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



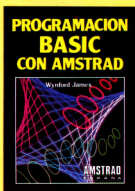
Código máquina para principiantes con AMSTRAD
Ideal para iniciarse en el código máquina del Z80 y en el sistema operativo del AMSTRAD.
2.100.— Pts.



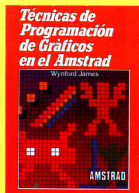
Hacia la Inteligencia Artificial con AMSTRAD
Convierta su AMSTRAD en un compañero inteligente.
1.500.— Pts.



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



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



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



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



indescamp
PUBLICACIONES

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

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

DE VENTA EN EL CORTE INGLES
Y TIENDAS ESPECIALIZADAS

Marca Registrada por el Grupo Indescamp.

AMSTRAD CPC-6128



- MICROPROCESADOR Z80A.
- 128 K DE MEMORIA RAM (41 K DE USUARIO EN BASIC Y 61 K EN CP/M PLUS)
- 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