

INPUT

Publicación práctica
para usuarios de

sinclair

Revista mensual 1987

Precio 375 Ptas

Año 2 Número 16

**COMO
DIBUJAR
EN 3-D**

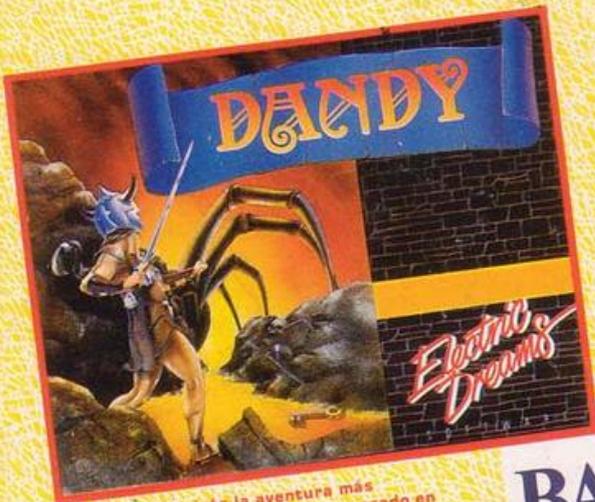
**MAS SOBRE
INTELIGENCIA
ARTIFICIAL**

**PINTA CON TU
SPECTRUM**

**Participa en el concurso de
aplicaciones**



...Te seguimos presentando el mejor software del año



Con DANDY vivirás la aventura más complicada que jamás te hayas pensado en una mazmorra. No te será fácil encontrar el tesoro. DANDY es la mazmorra definitiva.

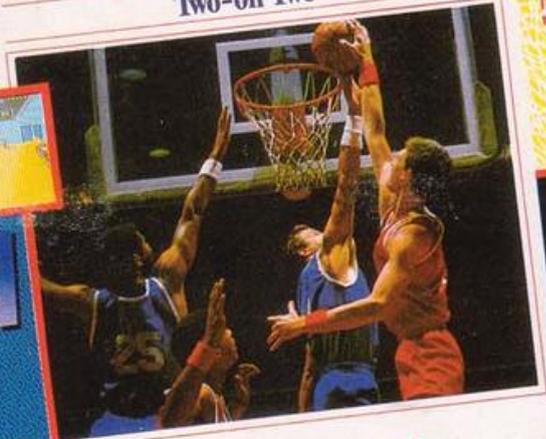
CSA



El universo, objetos tridimensionales, criaturas extrañas y la oscuridad del espacio, llenan de emoción y tensión este juego, donde tu supervivencia depende de tus reflejos.

CSA

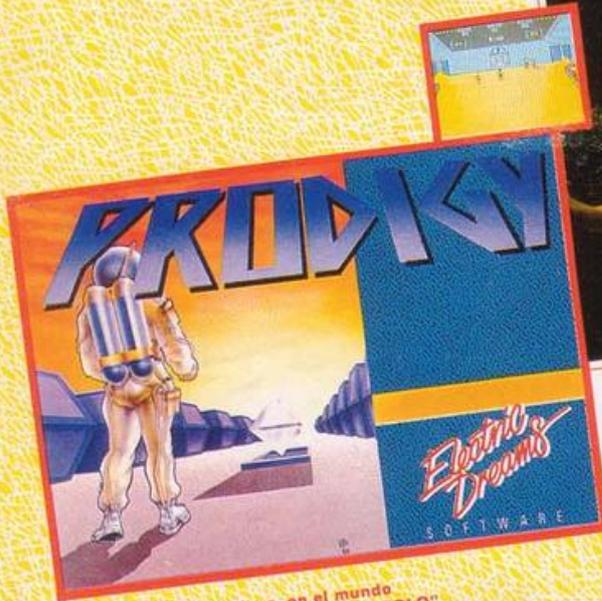
CHAMPIONSHIP BASKETBALL™ Two-on-Two™



GAMESTAR★

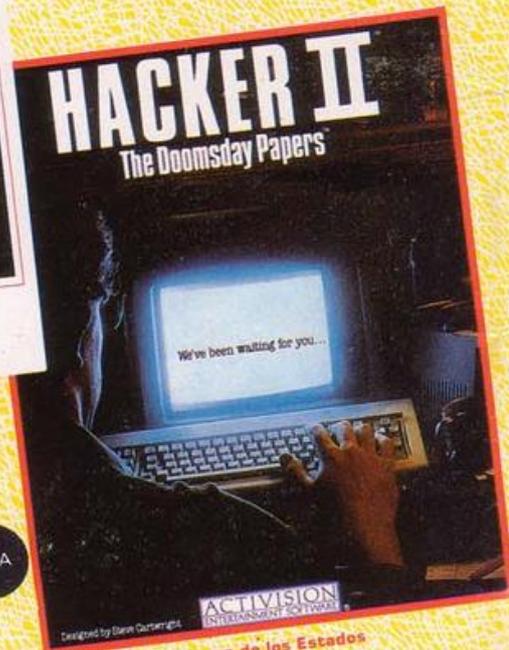
Este BASKET empieza donde otros acaban, porque se basa en el juego de equipo. Con los mejores gráficos de un juego de BASKET y la variedad de posibilidades de juego (Dos jugadores y la computadora, prácticas, liga de 23 jugadores etc.) nunca te cansarás de jugar hasta llegar a ser un campeón.

CSA



PRODIGY nos introduce, en el mundo "MEC" donde debemos conducir a "SOLO" el hombre sintético que cuida de "NEJO" y librarlo de los peligros más adversos, sin olvidarnos de WARDLOCK, el ser mecánico que quiere destruir toda vida orgánica. Sus efectos sonoros y en tres dimensiones lo hacen inmejorable y diferente.

CSA



Saludos del gobierno de los Estados Unidos... La CIA cuenta contigo para proteger a los países de Occidente. Los Rusos tienen en su poder el libro llamado "El día del juicio final". Con él pueden tener al mundo en sus pies. Y aquí entras tú, de lo demás, sólo podemos decirte: BUENA SUERTE. Falta te hace.

CSA

Disponibles para:

COMMODORE
SPECTRUM
AMSTRAD CASS/DISK

CSA

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES, O DIRECTAMENTE POR CORREO O TELEFONO A: **PROEIN, S.A.**

Distribuido en Cataluña por: DISCOVERY INFORMATICA C/ Arco Ins. 75 - BARCELONA Tels. 256 49 08 / 09

Velázquez, 10 - 28001 Madrid - Tels. (91) 276 22 08 / 09



AÑO 2 NUMERO 16

DIRECTOR: Manuel Pérez

COORDINADOR EDITORIAL: Francisco de Molina

DIRECTOR DE ARTE: Luis F. Balaguer

REALIZACION GRAFICA: Diego Tudela

COLABORADORES: Antonio Taratíel, Ernesto del Valle, Equipo Molisoft, Ramón Rabasó, Jaime Mardones

FOTOGRAFIA: Joan Boada

INPUT Sinclair es una publicación de PLANETA-DE AGOSTINI, S.A.

GERENTE DIVISION DE REVISTAS: Sebastián Martínez

PUBLICIDAD: José Real-Grupo Jota
Madrid: c./ General Varela, 35
Teléf. 270 47 02/03

Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: TECFA, S.A.

IMPRESION: Sirven Gráfico
C/ Gran Vía, 754-756, 08013 Barcelona
Depósito legal: B. 38.115-1986

SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Aribau, 185, 1.º
08021 Barcelona

DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S.A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

INPUT Sinclair es una publicación controlada por



INPUT Sinclair es independiente y no está vinculada a Sinclair Research o sus distribuidores

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.

© 1987 by Planeta-De Agostini, S.A.

Copyright ilustraciones del fondo gráfico de Marshall Cavendish

INPUT sinclair

SUMARIO

EDITORIAL	4
APLICACIONES	
MASTERFILE:	
EL MANEJO EFICAZ DE FICHEROS	5
EL COLOR EN TU SPECTRUM (I)	14
MECANOGRAFIA POR ORDENADOR	48
CODIGO MAQUINA	
LA POTENTE INSTRUCCION LDIR (II)	10
PROGRAMACION	
APRENDE A DIBUJAR EN 3.D (I)	17
PARTICIPA	
MUSICA Y SIMULACION DE PARTITURAS	27
INTELIGENCIA ARTIFICIAL	
LISP (I)	40
REVISTA DE SOFTWARE	56
EL ZOCO DE INPUT	66
PROGRAMACION DE JUEGOS (COLECCIONABLE)	31
ADIVINA MIS PALABRAS	
ULTIMANDO EL JUEGO DE LAS PALABRAS	

SACALE PARTIDO A TU ORDENADOR

Después de la compra del nuevo ordenador o de haber completado tu equipo con nuevos periféricos, ha llegado el momento de que te pongas a trabajar en serio con ellos y deshechar la tentación fácil de aparcarlos en un rincón. Como siempre, INPUT te ofrece la posibilidad de combinar el aprendizaje de las técnicas de programación y la información de las novedades de hardware, con el juego, la enseñanza de trucos útiles y el seguimiento de los más interesantes programas que se publican.

Conscientes de que el mundo de la informática vive un movimiento permanente, del que los medios de comunicación no debemos prescindir, hemos confeccionado una encuesta, con cuyas respuestas esperamos poder sintonizar mejor con vosotros. A fin de

que os resulte más sugestiva, se efectuará un sorteo entre todas las contestaciones recibidas.

Por último, hemos de señalar que con la publicación de este número de INPUT inauguramos un nuevo *Concurso de Aplicaciones y Utilidades*, destinado a intentar ampliar la gama de servicios que os prestan vuestros ordenadores.

A nuestro juicio, esta nueva competición estimulará un importante intercambio de experiencias y permitirá introducir nuevos temas en vuestra revista.

Sin duda, vuestras respuestas serán tan entusiastas como en anteriores ocasiones, lo que elevará la calidad y el interés de la revista.

Así pues a DISFRUTAR de vuestro INPUT y... a PARTICIPAR.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un *ranking* que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en INPUT.

El resultado de las votaciones será publicado en cada número de INPUT.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Aribau, 185. Planta 1. 08021 Barcelona

ELIGE TUS PROGRAMAS

Primer título elegido	_____	Segundo título elegido	_____
Tercer título elegido	_____	Programa que te gustaría conseguir	_____
Qué ordenador tienes	_____	Nombre	_____
1.º Apellido	_____	2.º Apellido	_____
Fecha de nacimiento	_____	Teléfono	_____
Dirección	_____	Localidad	_____
Provincia	_____		

INPUT SINCLAIR N.º 16

MASTERFILE: EL MANEJO EFICAZ DE FICHEROS

■	CARACTERISTICAS
■	OPCIONES DEL MENU
■	EJEMPLO DE MANEJO DE FICHEROS
■	REALIZACIÓN DE INFORMES

Uno de los programas más populares y a la vez más potentes para el manejo de ficheros con el ZX-Spectrum es el de la casa inglesa Campbell System.

Trataremos de hacer una descripción en profundidad del Masterfile. Aunque existe la versión 09, nos basaremos en la 06, que por un lado es un poco menos complicada de manejar, conservando lo fundamental, y sobre todo porque se trata de una versión castellanizada y comercializada en España.

La versión descrita es para *cassette*, pero resulta muy fácil prepararla para *Microdrive*. En nuestro caso particular —y con gran eficacia— estamos manejando el Masterfile con la unidad *Opus-Discovery*. Quedan libres para usuario 32 Kbytes de memoria. Sólo puede implementarse en el *ZX-Spectrum* de 48K.

Todos los datos del Masterfile pueden almacenarse en *cassette* o disco. Existe una opción para almacenar el programa BASIC (junto con toda la información general referente a Campos e Informes y los datos de todos los registros) más el programa en código máquina y otra para almacenar sólo los datos de los registros. Esta última opción permite una mayor rapidez en el proceso de LOAD/SAVE cuando se requieren varios ficheros para el mismo programa. Conviene por tanto conservar como patrón una copia del programa Masterfile sin ningún otro tipo de definición, con un fichero de datos totalmente vacío para, a partir de ella, efectuar las modificaciones que convenga en cada caso.

Para aprovechar al máximo la memoria disponible, la asignación de espacio de los diversos registros es «dinámica», es decir, se ocupa sólo el espacio requerido en cada caso, si bien para ello el programa deber ser un

poco más complejo a nivel interno. Habitualmente los programas de manejo de ficheros consideran de una forma rígida el espacio asignado a cada registro y a los campos que lo componen. Por ejemplo, si al campo «nombre» le asignamos 10 caracteres e intentamos poner en él la palabra Fuenteovejuna sólo nos cabrá parte de ella. Si por el contrario el nombre a introducir es Pepe desaprovecharemos bastante espacio. Con el Masterfile no se desperdicia espacio en memoria.

Antes de cargar el Masterfile debemos bajar el puntero RAMTOP reservando espacio suficiente para alojar el programa en código máquina (CLEAR 57343).

CARACTERISTICAS MAS IMPORTANTES

Cada registro (ficha) puede estar compuesto de hasta 26 partes o campos, cada uno de estos campos está identificado por una letra (A-Z). Si fuese necesario, pueden utilizarse otros diez campos más, identificados por un número (0-9), pero para ello previamente hay que hacer POKE 64284,48 : POKE 64306,48. Cada uno de los campos, además de la Ref. (A-Z) puede recibir un nombre, fácilmente asociable a los datos, para una identificación más cómoda (ej. apellido, fecha, valor, ciudad, etc.).

Los campos pueden tener hasta 128 caracteres cada uno.

Los datos de un fichero pueden visualizarse en hasta 36 formas (0-9 y A-Z) para adaptarse a nuestras necesidades. A cada una de estas formas de presentación (Informes) también se le pueden asignar un nombre.

Todo el programa está regido por menús, que aunque lógicos y fáciles de utilizar cuando se tiene cierta experiencia, resultan un tanto confusos al

comienzo. Más adelante iremos comentando cada una de las opciones basándonos fundamentalmente en ejemplos. En este caso «un ejemplo vale más que una explicación».

El programa permite obtener copias en papel de los informes pero sólo está preparado para utilizar la *ZX-Printer* o compatibles. La presentación en pantalla es la estándar de 32 caracteres por línea y 22 líneas en total.

MEJORAS DE LA VERSION 09

Las mejoras de esta versión sobre la 0.6 no afectan al manejo de los datos del fichero y pueden resumirse básicamente en:

- Manejo de hasta 51 caracteres por línea.
- Uso de otra impresora de tipo matricial, además de la *ZX-Printer*.
- Uso de *cassette* y *Microdrive*/disco.

En resumen se trata de un programa que puede dar mucho juego a nivel doméstico e incluso profesional, sobre todo si se utiliza con disco.

MANEJO DE LAS OPCIONES

En la primera pantalla aparece básicamente el Menú Principal (se muestra más abajo), con el nombre del fichero que estemos manejando, así con información de los registros utilizados, los seleccionados y la memoria que nos queda disponible:

MASTERFILE VER 06

Introducir regs	A
Elegir informe	C
Visual/Imprimir	D
Edit.def.informe	E
LOAD un fichero	L
Ver nombre datos	H
Consultar ficha	S
Invertir selección	I
RESET selección	R

Borrar regs. sel P
 Total / Promedio T
 SAVE prog/fich V
 BASIC usuario U
 Regs = 00000
 SEL = 00000
 Spa = 32300

Iremos explicando las posibilidades del programa agrupándolas por temas, en lugar de seguir el orden del menú. Los detalles puede consultarlos el lector en el manual.

Para clarificar mejor los conceptos nos apoyaremos en el siguiente supuesto: un Colegio que desea disponer, por cada curso, de los datos de sus alumnos (nombre, apellidos, dirección, teléfono, notas de las asignaturas, etc.) para consultarlos (o imprimirlos) conforme a diversas necesidades (listas de apellidos y teléfonos, nombres y notas de asignaturas, nombre y dirección para envío de correspondencia, etc.).

Resulta altamente recomendable no acometer directamente «en el teclado» la confección de una base de datos, aunque sea sencilla. Todo debe planificarse tranquilamente con ayuda de papel y lápiz antes de introducir los datos al ordenador. De lo contrario corremos el riesgo de entrar en bucles extraños o callejones sin salida o con ella difícil.

PLANIFICACION DEL TRABAJO

A.- Fichero de datos

En el caso del Colegio es deseable disponer de un «fichero» con «registros» conteniendo los siguientes datos, (a cada dato o «campo» del registro le asignaremos una referencia A-Z y un nombre). (Ver cuadro 1).

A1.- Asignación y manejo de nombres de campos

Situados en el Menú Principal elijamos la opción M. Aparecerá una pantalla con un rótulo en fondo azul con letras en blanco: «Ver nombre datos». A la izquierda y un poco más abajo, con letras blancas sobre fondo rojo: «< Ref. dato >»: «< nombre dato >». En el centro sobre fondo amarillo con letras negras surge el siguiente menú.

Adicionar item A
 Reemplazar item R
 Borrar item E
 Otro item M
 Visual/Imprimir D
 Reportar item G
 Apuntar item P
 Menú Principal M
 A este menú le denominaremos Menú 1

Pulemos A. Se nos solicitará: «Dar ref. de datos A-Z». Tan pronto tecleemos una referencia nos solicitará: «Introducir texto 1-128 c», siempre y cuando la ref. de dato esté libre. En caso contrario aparecerá: «este item (elemento) está ya registrado» y deberemos comenzar de nuevo.

Así iremos introduciendo en nuestro caso desde «A :1er Apellido» hasta «L : Grupo de curso».

En cualquier momento podemos conocer la asignación de campos. Para ello debemos pulsar sencillamente N, e irán apareciendo los campos en el orden en que se introdujeron (no necesariamente A-Z en forma consecutiva). Podemos situarnos directamente en uno particular mediante G. Una vez situados en una ref. de dato (campo) podemos proceder a eliminarlo (E) o modificarlo (R).

A2.- Introducción y modificación de datos en los registros

En el Menú Principal pulsamos A y tendremos una presentación en pantalla de aspecto idéntico al del apartado anterior, pero en este caso estamos en

condiciones de introducir los datos en cada uno de los campos que tenemos asignados. Para comenzar un nuevo registro debemos regresar al menú principal (M) y repetir la operación. Los registros quedan archivados uno a continuación de otro en el orden cronológico en que se introducen.

Dentro de un registro podemos situarnos dentro de cada campo con M ó G y proceder a introducir un valor (A), borrar (E) o modificar (R) uno ya existente. En el caso de la adición de datos, lo más cómodo es pulsar P y el ordenador nos irá solicitando automáticamente los datos de todos los campos (al final reaparece automáticamente el menú 1). Esta opción (P) tiene la desventaja de que pasaremos obligatoriamente por todos los campos y por tanto no es adecuado si solamente desemos operar en algún campo en particular.

En nuestro caso, irán introduciéndose los diversos registros: «Pérez, Ruíz, Antonio, Pez 27, 5, 3.5, 2C», etc.

Recordemos que antes de salir al menú principal (M) podemos verificar o modificar cualquier campo, pero una vez que hubieramos salido ya no, en esta opción (más adelante veremos el procedimiento).

En «Regs» al pié de página veremos los registros generados y la memoria que nos va quedando (Spa).

B.- Informes

No siempre es necesario manejar a la vez todos los campos de los registros. Lo normal es que se utilicen sólo algunos a la vez. También es lógico que podamos presentar estos datos distribuidos físicamente en la pantalla (o impresora) de forma conveniente, complementados con otro tipo de informaciones fijas, como pueden ser rótulos, líneas, cajetines, colores, *flash*, etc.

El **Materfile** nos permite tener pre-seleccionadas hasta 36 pantallas o informes diferentes.

En el caso de nuestro colegio podría interesarnos disponer, por ejemplo de: a) lista de teléfonos, b) fichas de alumno, c) resumen de notas, etc.

CUADRO 1

Ref.dato	NOMBRE
-----	-----
A	1er Apellido
B	2do Apellido
C	nombre
D	dirección
E	ciudad
F	teléfono
G	nomb. del padre
H	notas matemát.
I	notas historia
J	notas literatura
K	notas física
L	grupo de curso

Veamos el aspecto que pudiera tener cada uno de estos informes.

- a) FICHA DEL ALUMNO
- b) RESUMEN DE NOTAS
- c) LISTA DE TELEFONOS ALUMNOS COU

(A modo de ejemplo pueden ver los cuadros 2, 3 y 4 en esta misma página).

B1.- Definición de Informes

Antes de poder utilizar un informe hay que definirlo. Para ello pulsar E en el Menú Principal y nos aparecerá el menú correspondiente. Con esta opción podemos modificar también los datos de presentación de un informe ya existente.

Lo aconsejable es tomar papel y lápiz y diseñar con toda exactitud el as-

CUADRO 2

c) LISTA DE TELEFONOS ALUMNOS COU

Apellidos	Nombre	N. Padre	Teléfono
.....
.....
.....

provechar una parte sustancial de él. Con X eliminamos de golpe un informe.

En cada informe o pantalla existe una parte general que afecta a todo y otra variable que sólo afecta a cada parte o elemento.

El uso de las Opciones resulta bastante evidente y no insistiremos más que en lo básico.

d) Profundidad (indica el número de líneas hasta que se repitan los datos del siguiente registro). Define el número de registros por página.

B.1.2 Añadido de elementos de Informe

A cada parte del informe (o pantalla) se le denomina, tal como ya se ha dicho «elemento de informe». Estos elementos pueden ser:

a) Datos (procedentes de la base de datos formada por los sucesivos registros)

b) Literales (texto fijo). Las coordenadas x,y de su aparición en pantalla se dan en caracteres a partir de la esquina superior izquierda.

c) Cajetín (rectángulo para alojar cualquier información). Las coordenadas x,y se miden a partir de la esquina inferior izquierda. La «anchura» es la altura de la caja y la «long» el espacio horizontal. Todo en *pixels*.

d) Línea vert. (líneas para separar información)

e) Línea horiz. (líneas para separar información)

Si deseamos añadir un dato, deberemos fijar su referencia (Ref. del campo) así como su situación en la pantalla. Para ello deberán contestarse las siguientes preguntas:

a) ref. del dato (ref. del campo donde está el dato)

b) línea (ref. donde debe imprimirse el dato)

CUADRO 3

a) FICHA DE ALUMNO

Apellidos:.....	Nombre:
Domicilio:.....	Ciudad:
Nombre del padre:.....	Teléfono:

pecto y contenido que queremos dar a cada informe (nuestra hoja debe reflejar cómo quedará la pantalla).

Pulsamos A y aparecerá primeramente en pantalla la solicitud de Ref. de Informe. Una vez elegido uno cualquiera (0-9, A-Z) éste aparecerá en la esquina superior derecha y podremos ver el siguiente menú:

- Adc. elemt. nuevo A
- Reempl. elemento R
- Borrar elemento E
- Otro elemento N
- Visual/Imprimir D
- COPY def. inform C
- Borrar informe X
- Menú anterior M

Las opciones A-N del menú se refieren a «elementos» o «partes» del informe (rayas, datos, cajetines, rótulos, etc.). Con la opción D veremos el aspecto que tiene el informe. Con C podemos aprovechar el contenido de otro informe, si es que merece la pena

Con (A) añadiremos un elemento, con (R) podemos modificar su contenido y con (E) lo eliminamos.

B.1.1 Datos Generales

El programa tiene asignados datos generales por defecto pero no obstante podemos variarlos (opción R). Estos datos son:

- a) Fondo (color de fondo 0-7)
- b) Contorno (color de borde 0-7)
- c) Secuencia (secuencia de visualización a partir de un campo determinado. Por defecto el orden en que se crearon)

CUADRO 4

b) RESUMEN NOTAS

Alumno	Física	Litera.	Matemát.	Hist.	Grupo
---	---	---	---	---	---
...

- c) columna (igual que el caso anterior)
- d) longitud (espacio en caracter que puede ocupar el dato)
- e) profundidad (número de líneas reservadas para el dato)
- f) color, brillo, etc.
- g) relleno (determina si el color de fondo aparece en todo el campo o sólo hasta donde hay información)
- h) texto nulo (texto que se desea que aparezca cuando no hay otro dato en el campo).

En el caso de querer colocar un texto en una cierta parte, el **Masterfile** nos solicita el contenido del texto, su ubicación (fila, columna) y los atributos (color, flash, etc.)

El **Masterfile** utiliza como «nombre de informe» el del primer texto dado a un «elemento de informe» (ver opción C del Menú Principal).

Cuando el «elemento de Informe» que deseamos añadir es una línea horizontal o cajetín, deberemos facilitar los parámetros correspondientes.

Mediante la Opción D podemos visualizar como nos va quedando el informe.

B.1.3 Modificaciones en los elementos de Informe

El procedimiento para seleccionar, visualizar, modificar o eliminar el contenido asociado a un «elemento de Informe», opciones G, M, R, E, es análogo a lo ya visto por el caso de campos de un registro.

MANEJO DE INFORMES

Una vez ya definidos los Informes que deseamos utilizar, e introducida una colección de datos en sus respectivos registros/campos, podemos empezar a utilizarlos. Para ello pulsamos la Opción C en el Menú Principal. Aparece el catálogo de los informes que tenemos definidos con su referencia (0-9 y A-Z) y el nombre asignado. Una vez elegido uno nos aparece el siguiente menú (le llamamos Menú 2):

- Otra página M
- Avance 1-9 rgts #
- Ir al 1er rgt B
- Imprimir P

- Act. reg. sup. U
 - Borrar reg. sup. E
 - Omitir reg. sup. O
 - Buscar en fichero S
 - Total / Promedio T
 - Solicitar Informe # R
 - Commutar R
 - Menú Principal M
- Las opciones «m», «#» y «B» permiten posicionar los registros para su visualización en los informes.

Si deseamos completar, modificar o incluso cancelar un registro, lo primero que hacer es posicionarlo en primer lugar en la pantalla y luego pulsar «U» o «E». Si se encuentra en otro lugar (por ej. el quinto de una lista) podemos avanzarlo de uno en uno mediante «O» hasta situarlo el primero.

Podemos pasar a ver otros informes mediante R.

Existen dos opciones muy interesantes y en las que vamos a detenernos.

BUSQUEDA

En ficheros con muchos datos quizá nos cueste bastante tiempo localizar la información que deseamos. Por ejemplo, saber a que alumno corresponde un teléfono dado, cuántos viven en una determinada ciudad, quienes se llaman Luis, etc. Con ayuda de la opción de búsqueda (S) la cosa puede resultar muy fácil.

Oprimiendo «S» aparecerá el menú:

- Todo el fichero A
- El seleccionado L
- Visualizar D
- Menú Principal M

Mediante «A» o «L» podemos elegir que la búsqueda se realice en todos los registros o sólo en aquellos que previamente se han seleccionado.

Para selección de ciertos registros, en el menú principal comencemos por cancelar cualquier selección previa (R). Invertiremos en I, de forma que todos queden seleccionados. Pasamos luego al informe que nos interesa y aquel registro que no deseemos utilizar en la búsqueda lo marcamos con «O» (veremos que el contador SEL decrece en uno). Tal como se ha dicho, el registro debe ser único por página o estar a la cabeza de ésta.

Una vez efectuada la selección ya podemos operar con ella. Incluso podemos eliminar de golpe todos estos registros (o los complementarios si utilizan I) con la operación «P» del menú. Para la selección deberemos elegir el campo (A-Z) por donde efectuaremos la búsqueda. El programa nos indicará en la parte inferior izquierda el nombre que hemos dado a ese campo (ENTER si no estamos conformes). A continuación aparece otro menú:

- Caracter C
 - Númérico N
 - Menú ant. M
- Pulsaremos «C» si el campo es un literal y «N» si aloja valores numéricos.

Masterfile nos presenta un nuevo menú:

- Igual a E
- Diferente de U
- Menor que L
- Mayor que G
- Busca cadena S
- Menú Principal M

Esta opción la podemos utilizar para buscar algo igual, mayor, menor o simplemente diferente al parámetro que elijamos. Con «S» incluso podemos elegir una parte de un nombre (ej. los apellidos que tengan las letras «LEZ» en su constitución).

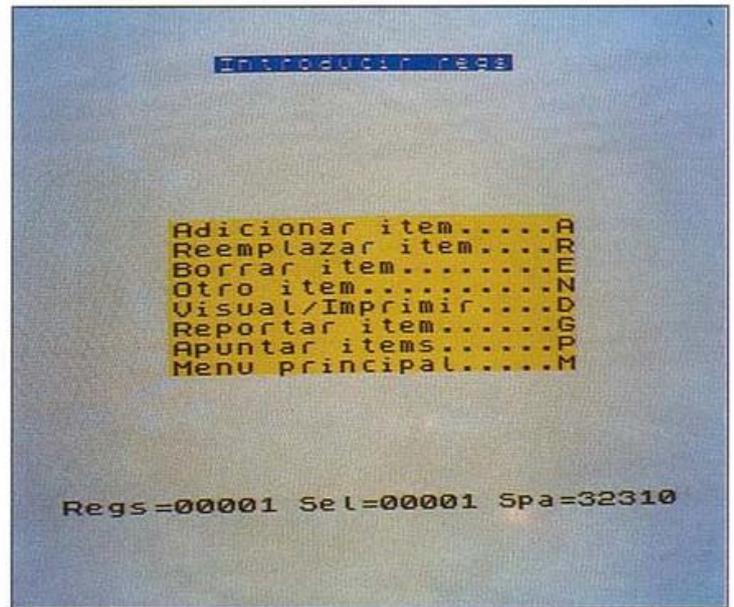
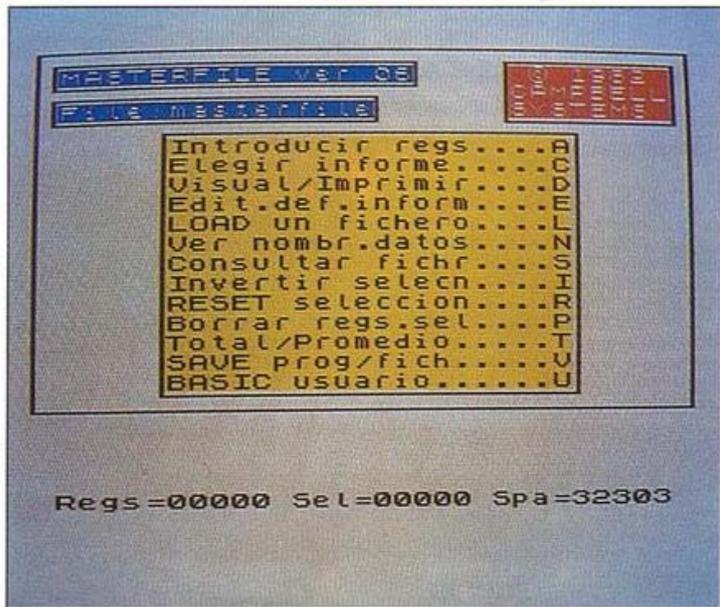
El indicador 'SEL' nos indicará cuántos registros cumplen nuestra condición y los podremos visualizar con «D».

La opción de búsqueda la podemos ejercer también directamente desde el menú principal.

TOTAL PROMEDIO

Cuando operamos con campos numéricos como puede ser nuestro caso a la hora de controlar las notas de los alumnos, puede resultar muy conveniente conocer para un campo determinado cuántos registros hay y cuál es su medida.

Una vez finalizada la visualización del informe requerido, pulsando «T» y eligiendo un campo (por ej. H, donde tenemos las notas de matemáticas) aparece al pie del informe el total de todas las notas (en este caso quizá no



sea de excesiva utilidad) y el promedio (si falta un valor lo toma por 0 y si aparece un dato no numérico lo avisa para que lo corriamos o lo saltemos).

Desde el menú principal también se puede ejecutar «T», si bien en este caso sólo aparecen los resultados sin visualizar los campos correspondientes.

No olvidemos que tendremos un resultado nulo, si no existen registros seleccionados (SEL=0).

BASIC DE USUARIO

El **Masterfile** permite adaptar el sistema a las necesidades de cada uno dentro de amplios márgenes. Cada usuario puede completar las operaciones estándar con ayuda de programación en BASIC. Para ello se conecta con el BASIC en tres ocasiones fundamentales:

- Al comienzo de cada registro se dirige a la línea 5000 (si es que la queremos utilizar). Este es el momento adecuado para preparar ciertas operaciones, como pudiera ser el *reset* o inicialización de variables. Al final de la operación se retorna a la situación en que se dejó el programa mediante **GO TO USR R**.

- Al final del manejo de los datos de cada campo pasa el control a la línea 6000 (si deseamos utilizarla). De

esta manera se puede tomar información, con vistas a una manipulación posterior en la línea 7000 del contenido de cada campo a través de la variable C\$. El primer carácter (C\$(1)) nos informa de la Ref. y el resto (C\$(2 to)) del contenido. El retorno al programa se realiza igualmente mediante **GO TO USR R**.

- Finalizada la manipulación de cada registro cede nuevamente el control a la línea 7000 del BASIC y es precisamente en este momento cuando se pueden efectuar las operaciones que nos interesen, como puede ser la generación de un nuevo campo con datos deducidos de otros campos, la eliminación del registro si no se cumplen ciertas condiciones, la modificación de campos, etc.

Si deseamos efectuar ninguna operación (tal como viene de origen) deberemos asignar a C\$ el valor nulo (7000 LET C\$=«»:GOTO USR R).

Podemos borrar el contenido de una campo haciendo que la variable C\$ tome el valor de su Ref. (x = A-Z) (7000 LET C\$=«x»: GOTO USR R).

Finalmente podemos sustituir todo el contenido de una campo haciendo que C\$ tome el valor de la ref. del campo en su primer carácter y el texto deseado en los siguientes. (7000 LET C\$(1)=«x»: LET C\$(2 TO)=« texto deseado »: GOTO USR R).

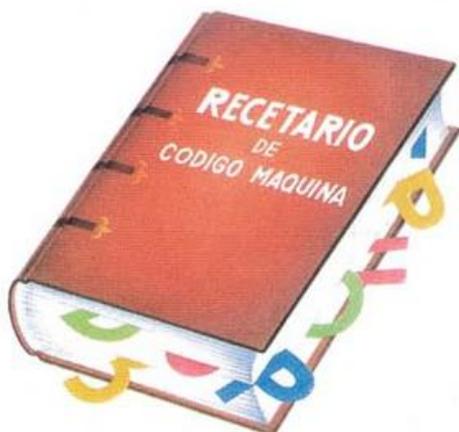
Siguiendo con nuestro ejemplo del Colegio, por ejemplo, puede calcular-

se la media de las notas de cada alumno y, en base a una tabla generada en BASIC, preparemos una información resumida de la situación del alumno (flojo en física, recuperar sábados, etc.). Esta información la podemos archivar en un campo, por ejemplo el R (Resultados) y dar pie a un nuevo informe que podemos tener a mano para una consulta rápida el día de la visita de los padres.

Incluso puede generarse un programa más complejo, de acuerdo a nuestras necesidades y habilidades (no olvidemos también que todos los datos almacenados están incluidos en la variable F\$).

RESUMEN

En líneas generales se trata de un programa de manejo de ficheros bastante potente y rápido (muy adecuado para **Microdrive** o *diskette* actualizando las líneas 4020, 4030, 4037 y 4050) suficiente para cubrir un gran abanico de posibilidades (control de *stocks*, inventarios, cuentas de una comunidad, etc.). El número de ficheros a manejar con el mismo programa resulta ilimitado. La impresión en pantalla de los diferentes informes admite una gran variedad de formatos, colores y atributos a la que sólo nuestro gusto y sentido de la organización puede poner límite.



En esta segunda parte sobre el tema del grupo de instrucciones LDIR, hoy vamos a presentar dos ejemplos de aplicación práctica e inmediata, no solamente teórica.

UN PROGRAMA PUBLICITARIO EN CODIGO MAQUINA

Seguramente alguna vez te habrás fijado en cualquiera de los muchos programas publicitarios que aparecen en los escaparates de las tiendas de informática, presentando algún equipo o producto. Son programas que suelen ir mostrando alternativamente en la pantalla de un monitor una serie de gráficos vistosos que van rotándose cada cierto tiempo. Si te has preguntado cómo poder conseguirlo de una manera fácil en nuestro Spectrum, la respuesta la tiene la instrucción LDIR.

Utilizando esta instrucción repetidas veces para transportar nuestras pantallas de una a otra zona de la memoria, podemos conseguir dicho efecto con la rapidez que nos ofrece el código máquina. Piensa entonces que la única limitación que tenemos es la memoria libre de que disponemos para almacenar nuestras pantallas, ya que para cada una de ellas necesitamos reservar un espacio de seis kilobytes (debido a la longitud del archivo de pantalla) más 768 bytes de la memoria de atributos.

En nuestro ejemplo trabajamos con cinco pantallas, por lo que deberemos escribir 10 cortas rutinas utilizando el LDIR (2 para cada gráfico almace-

LA POTENTE INSTRUCCION LDIR (II)

nado). En lo único que varían estas diez rutinas es en las direcciones fuente y destino de los bytes a manejar (direcciones almacenadas en los registros HL y DE). Una vez cargada la pantalla desde disco o cassette, ésta se encontrará situada en su demarcación normal (el espacio de la memoria reservado para la pantalla, que comienza en la posición 16384), por lo tanto, si previamente hemos bajado lo suficiente el RAMTOP, necesitaremos una rutina de traslado de los bytes que constituyen esa pantalla a su dirección de almacenaje en otra zona de la memoria por encima de nuestro RAMTOP.

De ahí que dentro de ese grupo de diez rutinas que vamos a utilizar, cinco serán las encargadas de pasar cada pantalla a su dirección particular, y por lo tanto todas ellas tendrán como dirección fuente la 16384 (registro HL) y como destino una determinada dirección de memoria (registro DE), estando la de cada pantalla separada al menos la mencionada distancia de 6 Kbytes + 768 bytes, para no solapar información.

Las otras cinco rutinas tendrán la misión opuesta a la mencionada anteriormente; su cometido será, una vez que ya han sido cargadas y almacenadas en su posición, el ir sacando cada pantalla de dicha dirección cuando son llamadas y trasladarlas al archivo de presentación visual para que podamos verlas.

Así queda configurado el modo de actuación del programa, que consiste en cargar cada una de las cinco rutinas, y una vez terminado el ciclo de recopilación de datos ir llamando alternativamente a cada pantalla mediante su rutina de extracción o presentación. Pasado un cierto tiempo llamamos a la rutina de presentación de la siguiente pantalla, que será la que ahora veremos.

La información de cada una de las pantallas irá superponiéndose sobre la de la anterior, y por tanto ésta se perderá en la zona de presentación visual, pero no en su lugar de almacenamiento, ya que como recordaremos la instrucción LDIR genera por así decirlo una copia en la dirección de destino, y guarda los datos originales en la posición fuente. Debido a este mecanismo de actuación del LDIR nuestro tiempo de acceso a cada pantalla se reduce aún más al no tener que guardar de nuevo cada gráfico.

Para no complicar más nuestro programa y debido a que ya es suficientemente rápido, la trama de ejecución del mismo se hará desde el BASIC mediante un sencillo bucle, y el tiempo que tarda en presentar cada pantalla se seleccionará mediante una sentencia PAUSE.

Como datos técnicos del programa podemos indicar que las rutinas de "almacenamiento" de pantallas son del tipo:

```

10 ORG 30850 / 30862 /
    30874 / 30886 /
    30898
20 ENT 30850 / 30862 /
    30874 / 30886 /
    30898
30 LD HL,16384
40 LD DE,30971 / 37884 /
    44797 / 51710 /
    59623
50 LD BC,6911
60 LDIR
70 RET
    
```

donde varía la dirección de destino (línea 40) para las cinco pantallas, y la ubicación de las rutinas en memoria (líneas 10 y 20). La línea 50 indica el número de datos a trasladar.

En las rutinas de presentación varía la dirección fuente, que coincidirá con la de destino en las anteriores:

```

10 ORG 30910 / 30922 /
      30934 / 30946 /
      30958
20 ENT 30910 / 30922 /
      30934 / 30946 /
      30958
30 LD HL,30971 / 37884 /
      44797 / 51710 /
      59623
40 LD DE,16384
50 LD BC,6911
60 LDIR
70 RET
    
```

La situación de las rutinas en memoria será distinta (líneas 10 y 20) ya que cada una es diferente. Necesitaremos 120 bytes de memoria para almacenar las rutinas de manejo de pantallas ya que cada una de ellas ocupa 12 bytes.

Si ponemos nuestro punto de vista en el listado BASIC del programa, las líneas 10 a 130, ambas inclusive, constituyen el cargador BASIC de nuestra rutina en código máquina.

La línea 220 puede ser cambiada por 220 LOAD a\$SCREEN\$ si se va a utilizar el cassette para realizar la carga original de pantallas.

En cuanto al modo de funcionamiento es de destacar que debido a la estructura del programa, las dos últimas líneas de nuestras pantallas no se pierden, siempre que hayamos tenido la precaución de grabarlas previamente (es decir, antes de que aparezca ningún mensaje del sistema operativo).

```

1 REM *****
2 REM * MOLISOFT *
3 REM *****
10 CLEAR 30849: BORDER 0:
  CLS
20 FOR n= 30850 TO 30969
30 READ a: POKE n,a: NEXT n
34 REM
35 REM Rutinas de almacenaje
    
```

```

de las 5 pantallas
36 REM
40 DATA 33,0,64,17,251,120,
  1,0,27,237,176,201
50 DATA 33,0,64,17,252,147,
  1,0,27,237,176,201
60 DATA 33,0,64,17,253,174,
  1,0,27,237,176,201
70 DATA 33,0,64,17,254,201,
  1,0,27,237,176,201
80 DATA 33,0,64,17,255,228,
  1,0,27,237,176,201
84 REM
85 REM Rutinas de
  presentación de las 5
  pantallas
86 REM
90 DATA 33,251,120,17,0,64,
  1,0,27,237,176,201
100 DATA 33,252,147,17,0,64,
  1,0,27,237,176,201
110 DATA 33,253,174,17,0,64,
  1,0,27,237,176,201
120 DATA 33,254,201,17,0,64,
  1,0,27,237,176,201
130 DATA 33,255,228,17,0,64,
  1,0,27,237,176,201
160 REM
170 REM Fase de
  almacenamiento
180 REM
190 LET direccion=30850
200 FOR n=1 TO 5
210 INPUT PAPER 7; INK 1;
  "NOMBRE DE LA PANTALLA
  "; LINE a$
220 LOAD "*"m";1; a$SCREEN$
230 RANDOMIZE USR direccion
240 LET direccion=direccion+12
250 NEXT n
252 PAUSE 100
254 REM
255 REM Fase de presentacion
  alternativa
256 REM
260 LET rutina=30910
270 FOR n=1 TO 5
280 RANDOMIZE USR rutina:
  PAUSE 200
290 LET rutina=rutina+12
300 NEXT n
310 GO TO 260
    
```

MANEJO DE PROGRAMAS EN BLOQUE

Como segunda utilidad dentro de este capítulo práctico del LDIR, presentamos el "Manejo de programas en bloque". Pensamos que aunque es de aplicación menos espectacular, puede servir para ahondar más en nuestros conocimientos sobre la estructura y manejo de la memoria.

El nombre dado de "manejo en bloque", sirve para expresar la idea que se pretende desarrollar: la de tratamiento de nuestros programas BASIC como una sola unidad, transportable de manera similar a como haríamos con una serie de datos (que en el fondo viene a ser lo mismo).

La instrucción LDIR, como hemos venido recalando en este artículo, nos brinda la oportunidad de manejar bloques de bytes de una manera rápida y sencilla. ¿Por qué no intentarlo con nuestros programas BASIC?...

Según lo visto, lo único que nos falta son unos ciertos conocimientos para poder manejar nuestro programa "sin riesgos". La demostración más patente de dicha "manipulación" puede ser el almacenamiento que podemos hacer de nuestras rutinas BASIC como si fueran bytes de código máquina, pudiéndolas proteger incluso de la acción de NEW por nuestro RAMTOP. Una vez ejecutado dicho comando, comprobaremos que aunque no esté presente nuestro programa, podemos recuperarlo fácilmente haciendo uso de nuestra rutina "recuperadora" LDIR.

Tenemos que tener presente que para poder almacenar nuestro programa como una serie de datos normales, y si tenemos la intención de recuperarlo para que funcione, no sólo debemos guardar las líneas BASIC en sí, sino otra serie de zonas de memoria interconexionadas con éstas y que son vitales para una ejecución correcta. Así que cada vez que queramos conservar un programa en su forma de bytes (incluso podremos guardarlo en cinta o disco con este formato), deberemos guardar un bloque de memoria compuesto por:

— Las variables del sistema: Comien-

zan en la posición 23552 y se alargan hasta la 23734. Consisten en una especie de "tabla de chequeo" del sistema operativo, conteniendo toda clase de punteros y referencias necesarios para el funcionamiento del S.O. Guardan la información sobre el comienzo y longitud de cada una de las zonas variables (PROG, VARS, espacio libre, etc.), estado actual del teclado, pilas o stacks, etc... Si omitimos o incluimos las variables del sistema de otro programa o momento, corremos el grave riesgo de quedarnos "colgados".

— El programa BASIC: Espacio físico ocupado por las líneas del BASIC. Su importancia no requiere comentarios. Es una zona variable que depende del tamaño del programa, y cuya longitud viene indicada por las variables del sistema PROG (comienzo) y VARS (final).

— La zona de variables: Donde se almacena el nombre, tipo y contenido de todas las variables declaradas hasta ese momento en nuestro programa BASIC. Es una zona tan importante como el programa en sí. Se extiende hasta la dirección apuntada por la variable del sistema E_LINE, y acaba con una marca o delimitador (80H u ochenta en hexadecimal).

En teoría guardando este bloque de datos, y algunas zonas intermedias como los Mapas del Microdrive si está enchufado, y la información de los distintos canales, conseguiríamos el efecto deseado. Pero en la práctica siempre necesitamos una parte del espacio que se encuentra a continuación o espacio de trabajo. En esta zona es donde gestionamos todas las operaciones relacionadas con el manejo temporal de datos (INPUT de datos, EDIT de líneas, etc...).

NOTA: Si encuentras alguna dificultad de "situación" en el tema, consulta la parte referente a la memoria que se encuentra en tu manual, sobre todo el magnífico diagrama que incluye sobre la disposición de ésta.

De todo lo expuesto hasta ahora, sacamos en conclusión que debemos

guardar un bloque de memoria que se extiende desde la posición 23552 hasta la indicada por la variable del sistema WORKSP o STKBOT. Sin embargo te recomendamos que lo alargues hasta STKEND para evitar problemas (principio del espacio libre). Si el programa a almacenar es de longitud razonable, no notaremos la relativa pérdida de esos pocos bytes.

La parte principal de nuestro programa la componen las dos rutinas LDIR que sirven para "almacenar en" y para "recuperar de". Ambas están unidas y ocupan 31 bytes que se han colocado a partir de la posición 60000, aunque se podría haber apurado y colocarlas en la (65535-31), siempre que no se prevea algún tipo de incompatibilidad con interfaces, etc...

Una vez hecho lo anterior, y tras hacer NEW para borrar la zona por debajo del RAMTOP, podemos comprobar con agrado que al ejecutar RANDOMIZE USR 60014 recuperamos nuestro programa.

Este es un ejemplo de prueba, pero si deseásemos salvar cualquier otro programa, recordaremos que sólo es absolutamente necesario la ejecución de nuestra rutina en ensamblador, debiendo introducir "a pelo" en 130 DIREC... y 140 LONG... los valores de la dirección elegida, y longitud del bloque (es decir el valor '23552-(PEEK 23653+256*PEEK 23654)'). También deberemos calcular correctamente el espacio que necesitamos y, a continuación, reservarlo mediante CLEAR.

10	ORG	60000	; DIREC y LONG, que están a 0,
20	ENT	60000	; contendrán la dirección de
30	LD	HL,23552	; almacenado y la longitud del
40	LD	DE,(DIREC)	; bloque que ocupa nuestro
50	LD	BC,(LONG)	; programa. Dichos valores se
60	LDIR		; calculan desde el BASIC y son
70	RET		; "pokeados" en esas dirs. Hasta
80	LD	HL,(DIREC)	; la línea 70, forma parte de la
90	LD	DE,23552	; rutina de "traslado a". De la
100	LD	BC,(LONG)	; 80 a 120 la de "recupe-
110	LDIR		; ración". Ambas utilizan los
120	RET		; mismos valores "fuente" y
130	DIREC	DEFW 0	; destino" pero intercambiándolos
140	LONG	DEFW 0	; entre DE y HL.

Nuestro ejemplo consistirá en guardar o "autoguardar" el mismo programa que sirve para generar las direcciones y calcular si el programa cabe en el espacio deseado.

Con RANDOMIZE USR 60000 lo guardamos, encargándose el programa de bajar el RAMTOP el número elegido de posiciones. Si existiera algún problema nos lo hace saber, debiendo elegir una "dirección" más acorde con la talla del programa. Podemos tener dificultades eligiendo valores próximos a 25000 por debajo, o a 60000 por arriba. Uno de entre los valores correctos de "DIRECCION DE ALMACENADO" podría ser 30000.

```

1 REM *****
2 REM * MOLISOFT *
3 REM *****
5 REM Cargador
  BASIC
  de la rutina en código
  máquina
10 CLEAR 59999: FOR
  n=60000 TO 60030: READ
  a: POKE n,a: NEXT n
20 DATA 33,0,92,237,91,123,
  234,237,75,125,234,237,
  176,201
21 DATA 42,123,234,17,0,92,
  237,75,125,234,237,176,
  201,0,0,0
22 REM
  
```

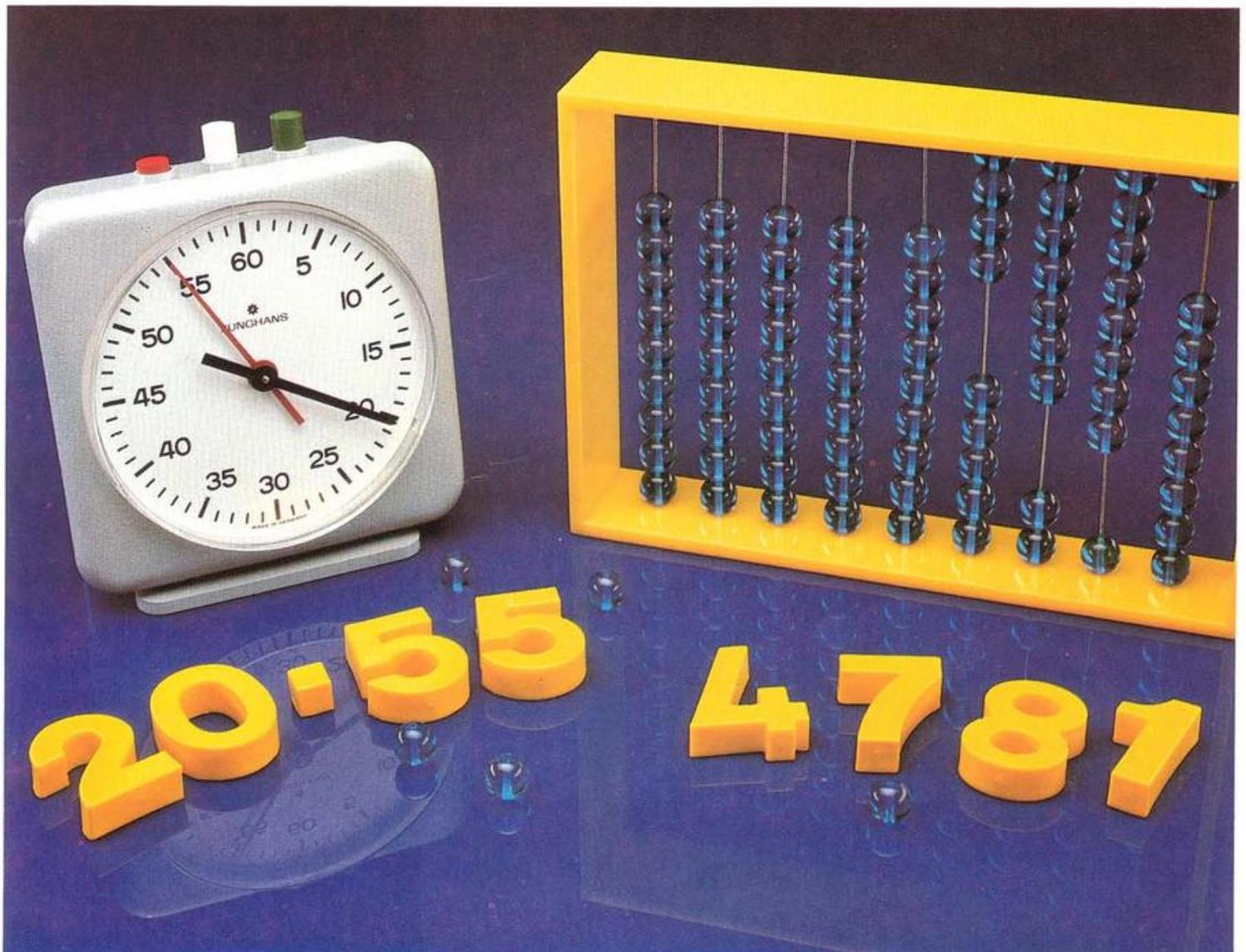
```

23 REM Calcula el final de la
    zona de trabajo
24 REM (comienzo del espacio
    libre)
25 LET stkend=(PEEK 23653
    +256*PEEK 23654)-1
30 INPUT PAPER 2; INK 7;
    BRIGHT 1; "DIRECCION DE
    ALMACENADO?"; direccion
34 REM
35 REM Calcula la longitud de
    los datos a guardar
36 REM
40 LET longitud=stkend-23552
44 REM
45 REM Direcciones del
    comienzo del espacio libre y
    final de los datos almac.
50 LET x=direccion+longitud

60 IF x>59999 OR
    direccion<(stkend+1) THEN
    PRINT #0; PAPER 1; INK 7;
    BRIGHT 1; "DATOS
    ERRONEOS. PULSA UNA
    TECLA": PAUSE 0: GO TO 30
64 REM
65 REM Calcula y 'pokea' los
    datos en las direcciones
66 REM 'LONG' y 'DIREC' de la
    rutina en C/M
67 REM
70 LET d1=INT (direccion/256):
    LET d2=direccion-256*d1
80 LET l1=INT (longitud/256):
    LET l2=longitud-256*l1
90 POKE 60027,d2: POKE
    60028,d1: POKE 60029, l2:
    POKE 60030, l1

94 REM
95 REM Mueve el RAMTOP lo
    necesario para almacenar los
    datos
96 REM
100 CLEAR (direccion-1):
    RANDOMIZE USR 60000
105 REM
110 REM Una vez rodado el
    programa y ejecutado un
    NEW,
115 REM con RANDOMIZE USR
    60014 lo recuperamos de
    nuevo
120 REM Si fuera necesario,
    habras de recolocar
125 REM el RAMTOP
    para dejar espacio de
    trabajo.

```





Con este número iniciamos, a un nivel muy elemental, una nueva serie de artículos, en la que trataremos de desvelarte todos los secretos de uno de los temas que con menor profundidad hemos tratado hasta ahora: *el color*.

Nuestro propósito es ofrecerte la información necesaria para obtener un óptimo aprovechamiento de las posibilidades gráficas del Spectrum, especialmente en aquellos aspectos relacionados con el color, aportando ideas, *trucos* y rutinas que podrás aplicar en el desarrollo de tus propios programas.

En la medida de lo posible, procuraremos seguir un orden ascendente en la explicación, a partir de los conceptos más simples, de forma que no tengas ningún problema para comprender su contenido, aun en el caso de que tu nivel de programación en BASIC sea muy bajo.

Así pues, para estar en condiciones de seguir adelante, basta con que te interese el tema.

ALTA Y BAJA RESOLUCION

Desde el punto de vista gráfico, el Spectrum presenta una serie de limitaciones que, según dicen, le colocan en clara desventaja con relación a otros micros de similares características. Uno de los inconvenientes señalados con más frecuencia, es la ausencia de color en «alta resolución». Como probablemente ya sabes, en el Spectrum sólo pueden coexistir dos colores dentro de un mismo carácter, uno para la tinta (INK), y otro para el

EL COLOR EN TU SPECTRUM (I)

papel (PAPER), lo cual implica la imposibilidad de dar un color específico a cada pixel de la pantalla. Este hecho no tiene por qué traducirse necesariamente en una pobre resolución gráfica, pero limita considerablemente la libertad de acción a la hora de realizar diseños medianamente complejos.

Para que tu micro tuviera «color en alta resolución», tendría que ser capaz de dar color «pixel a pixel», en vez de «carácter a carácter», pero desgraciadamente no puede hacerlo. Por ello, decimos que el ZX-Spectrum (o el Spectrum Plus) es un micro-ordenador de color en *baja resolución*.

Para comprobarlo, veamos qué pasa si trazamos dos líneas muy próximas, atravesando la misma columna de caracteres, con colores diferentes. Te clea:

```
1Ø DRAW INK Ø; Ø,1ØØ
2Ø PAUSE 4Ø
3Ø PLOT 4,Ø: DRAW INK 4; Ø,
  1ØØ
```

Como habrás observado, al trazar la segunda línea se altera el color de la primera. Ello se debe a que el ordenador sólo puede «memorizar» los atributos referidos a cada uno de los 768 caracteres de la pantalla, ignorando el color que pretendamos dar a cada pixel, de forma que dentro de un mismo carácter no pueda haber dos trazos con tinta de diferente color.

Veremos todo esto con más detalle cuando hablemos del *Archivo de Atributos*, en próximos capítulos. Por el momento, basta que comprendas las diferencias básicas entre alta y baja resolución, antes de pasar al epígrafe siguiente. Si tienes alguna duda, puedes recurrir al manual de uso de tu micro, donde encontrarás una descripción detallada del concepto de «carácter» y de los comandos empleados en el anterior programa.

El Spectrum está preparado para generar ocho colores básicos, ordenados según su intensidad, con un código que va del cero al siete. El siguiente programa lo muestra:

```
1Ø FOR N = Ø TO 7
2Ø PRINT INK N; AT 12,13+N;
  CHR$ 143; AT 13,13+N;N;
3Ø NEXT N
```

Utilizando los comandos INK y PAPER, seguidos del código de color correspondiente, se puede obtener cualquier combinación de colores. Un *truco* muy sencillo que se emplea frecuentemente para proteger listados de programa en BASIC, consiste en dar el mismo color a la tinta que al papel, de forma que en la pantalla no aparezca más que un plano monocromo sin nada escrito (al menos aparentemente). Te clea ahora lo siguiente:

```
1Ø REM PROGRAMA DE PRUEBA
```

Si a continuación introduces

```
INK 7: PAPER 7
```

y pulsas ENTER, la línea 10 se tornará invisible, pero aunque no puedas verla, en realidad seguirá estando ahí.

Como muestra de las diferentes combinaciones de color posibles, puedes teclear el siguiente programa:

```
1Ø FOR N= 1 TO 7Ø4
2Ø LET C1= INT (RND*7): LET
  C2= INT (RND*7):
  PRINT INK C1; PAPER C2;
  "a";
3Ø NEXT N
```

Si lo ejecutas varias veces, observarás que los resultados difieren. Esto se debe a que el programa utiliza dos variables aleatorias para generar los

códigos de color. Alterándolas, podrás manipular a tu gusto los resultados.

Teóricamente, existen $8 \times 8 = 64$ combinaciones posibles de colores, dentro de un mismo carácter (azul y rojo, azul y magenta, verde y amarillo, etc.). Sin embargo, en ocasiones se nos quedan cortos los ocho colores básicos a la hora de poner en práctica nuestros ambiciosos proyectos, y echamos de menos la posibilidad de mezclar colores para obtener otros nuevos, como si estuviéramos pintando al óleo. A continuación, te ofrecemos un interesante sistema que te permitirá lograrlo. Presta atención:

MEZCLANDO COLORES

¿Se pueden mezclar los colores?

Si consultas cualquier manual, la respuesta a tu pregunta será siempre negativa. Sin embargo, nosotros afirmamos que, con un poco de imaginación y por medios poco ortodoxos, sí es posible.

Nuestro método se basa en una técnica que podríamos llamar «puntillista», por su parecido con la empleada por algunos pintores impresionistas del siglo XIX. Consiste en alterar un punto de tinta de un color, con un punto de papel de otro color diferente, según muestra la figura número dos. Observando la pantalla a distancia normal, se produce un efecto visual mediante el cual nos parece ver un nuevo color, obtenido por la mez-

cla de los otros dos. Para comprobarlo, teclea el siguiente programa:

```
10 FOR N= 0 TO 7
20 READ A
30 POKE USR "A"+N,A
40 NEXT N
50 DATA 170,85,170,85,170,
85,170,85
60 PRINT AT 0,0; "COLORES
BASICOS"
70 FOR N= 0 TO 7
80 PRINT INK N; AT (N+1)*2,3;
CHR$ 143
90 NEXT N
95 PRINT AT 0,18; "MEZCLAS"
100 FOR N= 0 TO 7
110 FOR F= 0 TO 7
120 PRINT INK N; PAPER F; AT
(N+1)*2,18+F; CHR$ 144;
130 NEXT F
150 NEXT N
```

Al ejecutarlo, aparecerán 64 recuadros con las 64 combinaciones posibles que antes mencionábamos.

Si prefieres ensayar por tu cuenta las combinaciones que más te interesen, o comprobar esta técnica de una forma más participativa, entonces borra el anterior programa y ejecuta este otro:

```
10 FOR N= 0 TO 7
20 READ A
30 POKE USR "A"+N,A
40 NEXT N
50 DATA 170,85,170,85,170,
85,170,85
100 INPUT "TINTA?";T
110 INPUT "PAPEL?";P
120 FOR N= 1 TO 10
130 FOR F= 1 TO 10
140 PRINT INK T; PAPER P; AT
N,F; CHR$ 144
150 NEXT F
160 NEXT N
170 PRINT AT 1,12; "TINTA ";T
180 PRINT AT 3,12; "PAPEL ";P
```

Con algunas modificaciones, podrás emplear esta rutina en tus propios programas. Ten en cuenta que las primeras líneas sólo sirven para definir y

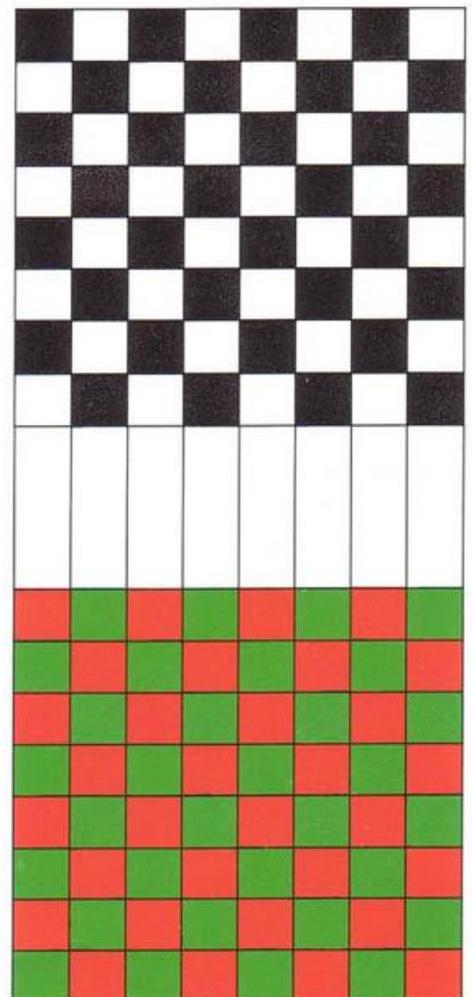
almacenar en la UDG la retícula ajedrezada que imprimimos posteriormente con

```
CHR$ 144
```

El programa propiamente dicho, comienza a partir de la línea 100.

BRIGHT

Para terminar, nos referiremos a un comando generalmente poco utilizado, que a pesar de su escaso uso resulta imprescindible en el diseño de gráficos. Aparentemente, puede parecer poco útil el dar un mayor brillo a un carácter determinado. No obstante, el comando BRIGHT puede aportar una mayor riqueza cromática al espectro gráfico, si sabe utilizarse. Esto se puede demostrar mejor en un monitor de blanco y negro (o un mo-



0 NEGRO	BLACK	■
1 AZUL	BLUE	■
2 ROJO	RED	■
3 MAGENTA	MAGENTA	■
4 VERDE	GREEN	■
5 CYAN	CYAN	■
6 AMARILLO	YELLOW	■
7 BLANCO	WHITE	□

nitro de color en modo b/n), debido a que las diferencias de brillo y luminosidad son más fácilmente apreciables. Veámoslo con un ejemplo:

```
5 LET T= 11
10 FOR N= 0 TO 7
20 PRINT INK N; AT T, 13+N;
   CHR$ 143;
30 NEXT N
40 BRIGHT 1
50 LET T=13: GOTO 10
```

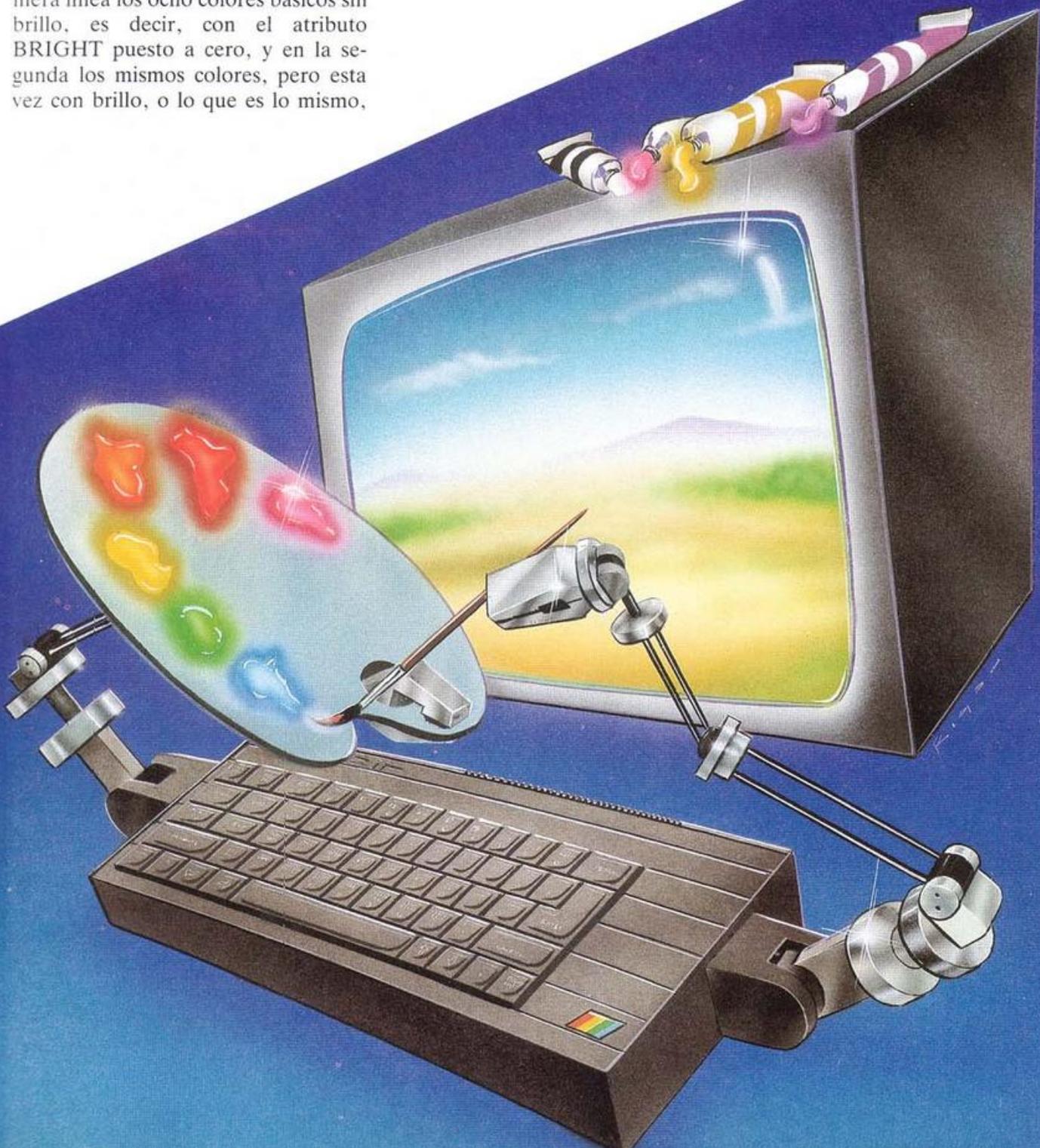
Este programa imprime en la primera línea los ocho colores básicos sin brillo, es decir, con el atributo BRIGHT puesto a cero, y en la segunda los mismos colores, pero esta vez con brillo, o lo que es lo mismo,

con el atributo BRIGHT puesto a uno. Con el monitor en blanco y negro, se aprecia cómo las diferencias son notables.

Ahora podríamos proponerte que intentarás aplicar el brillo a la técnica de mezclas de colores anteriormente descrita. Seguro que no te decepcionará el resultado.

En el próximo número volveremos a tratar el tema del color en el Spectrum, a un nivel ligeramente más

avanzado. Hablaremos de un método para alterar directamente el color y los demás atributos de una sola línea de programa, y de un espectacular truco que te permitirá «lucirte» en la presentación de tus pantallas, usando el comando BORDER.



APRENDE A DIBUJAR EN 3-D

- QUE SON DIBUJOS DE ESTRUCTURAS DE ALAMBRE
- DIBUJANDO UNA REJILLA
- DIBUJANDO UN CIRCULO

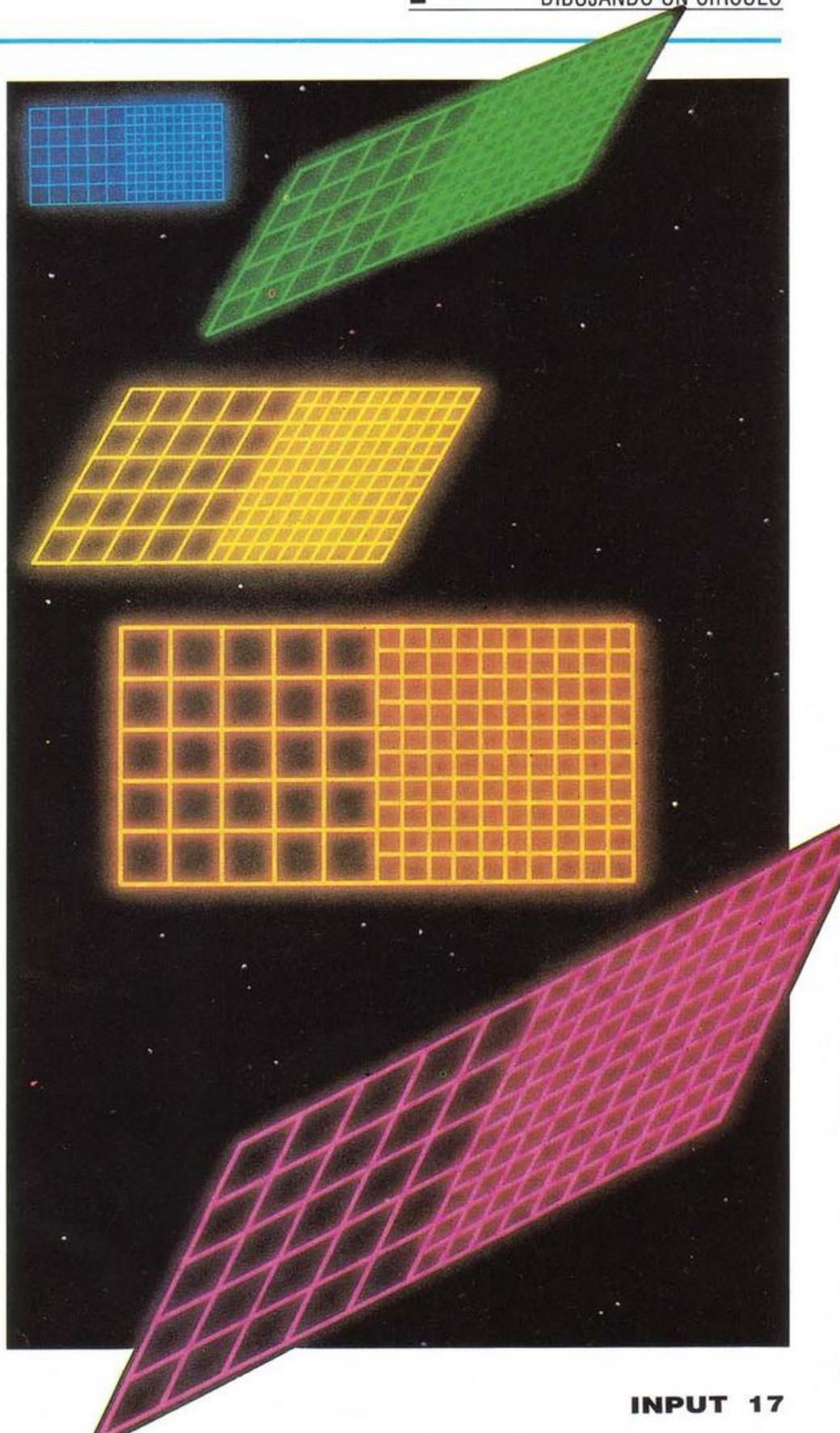
En este número os ofrecemos el primero de una serie de artículos sobre dibujos de estructuras de alambre, mediante rejillas y círculos.

En posteriores números de INPUT podrás ver cómo se convierten en bloques construidos en tres dimensiones.

El diseño y concepción de dibujos de «estructuras de alambre», utilizando los gráficos generados en un ordenador se ha impuesto en muchos campos, especialmente en el publicitario, gracias a que permite trabajar a la imaginación sin las limitaciones que conlleva el método clásico del papel y el lápiz. Si tú, como usuario de un ordenador doméstico, has deseado generar imágenes animadas de este tipo en la pantalla de tu televisor debes haberte quedado decepcionado al creer que tu micro no puede igualar el esplendor de las imágenes publicitarias producidas conjuntamente por programadores, fotógrafos y artistas. Sin embargo, no todo está perdido, porque en esta serie de artículos puedes aprender a dibujar y manipular imágenes en tres dimensiones de una forma que muchos ingenieros de diseño podrían envidiar.

La capacidad de tu ordenador para poder direccionar puntos individuales de pixels (elementos de imagen) en la pantalla proporciona un medio potente para poder realizar dibujos, como habrás podido apreciar en varios artículos sobre gráficos ya publicados en INPUT. El Spectrum ofrece 265 pixels horizontalmente y 176 pixels verticalmente. Por lo general puedes trazar líneas rectas o puntos. Combinando las posibilidades de color que tienen, pueden ser utilizados para proporcionar presentaciones de alta resolución.

El método normal para dibujar líneas en sistemas gráficos controlados por ordenadores es similar a cómo uti-



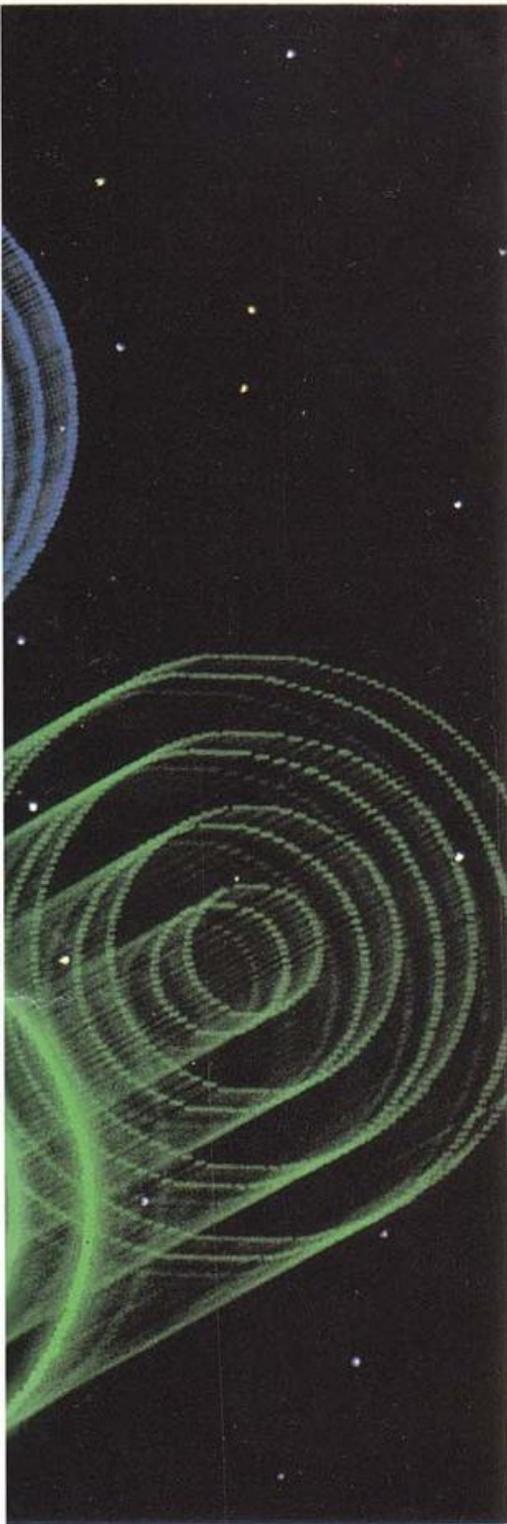


lizas un lápiz y un papel. Puedes mover el cursor sobre la pantalla sin marcarlo, o puedes desplazarlo por la pantalla dejando una señal. Cambiar de colores es tan simple como cambiar de lápiz. La principal diferencia entre el sistema de un ordenador y una persona que utiliza un lápiz es que el ordenador es más rápido y es mejor di-

bujando líneas rectas, una tarea que se hace difícil para la mayoría de la gente, si no emplea una regla.

Esta capacidad de dibujar líneas te permite dibujar también imágenes de perfiles o contornos e ilustraciones del tipo «estructuras de alambre» de objetos tridimensionales. Los gráficos de estructuras tramadas o de alambre,

son una representación imaginaria que consiste en una rejilla de líneas dispuesta sobre la superficie del objeto. Por lo general no se intenta ocultar las líneas detrás del objeto. Para poder hacer esto se requiere una considerable cantidad de materiales y de trabajo. El resultado es el de que el objeto aparece como si estuviera hecho



de una estructura de alambre unida para formar una rejilla.

Este tipo de visualización puede ser animado girando y moviendo el objeto de estructura de alambre, como se puede observar, por ejemplo, en las imágenes publicitarias de coches y juguetes. La animación rápida no puede hacerse sólo con un ordenador do-

méstico, ya que cada fotograma tiene que ser visualizado a un ritmo de, por lo menos, 25 fotogramas por segundo, para una animación continua, y un ordenador doméstico tarda mucho más de 1/25 de segundo para las visualizaciones más simples. En realidad, la mayor parte de los visualizadores comerciales animados por ordenador no están generados en «tiempo real» (como aparecen en la pantalla). En vez de ser así, cada fotograma se genera por separado, empleando segundos, minutos e incluso horas, para visualizaciones complicadas de alta resolución, y después son grabadas en magnetoscopios o filmadas para poder ser visualizadas, después, a la velocidad correcta.

Para las visualizaciones estáticas el tiempo no es lo importante, aunque puedes volverte un poco impaciente esperando a que la imagen sea completada. Sin embargo, con frecuencia es igualmente interesante, si no más, poder observar una imagen de estructura de alambre mientras es construida en la pantalla, al igual que lo es poder ver el objeto terminado, especialmente en los dibujos más complejos.

Para esta serie de artículos, las rutinas de dibujos van mucho más allá del campo del ZX81.

Es mejor empezar a dibujar formas simples —como un cubo o una esfera— y una vez que se haya ganado experiencia, se puede intentar dibujar formas más complejas e interesantes. En este artículo aprenderás cómo se pueden generar algunas formas bidimensionales y hacer que aparezcan en un espacio tridimensional.

VALORES INICIALES

Incluso para un ordenador de alta velocidad y de buena capacidad para dibujar líneas, se requiere una gran cantidad de trabajo para producir una imagen, bien elaborada, de «estructura de alambre». Se necesita, esencialmente, una orden para mover el cursor, sin que éste dibuje nada, y otra orden para dibujar desde la última posición del cursor a una nueva. Estas órdenes varían ligeramente en cada

ordenador. Y las órdenes o mandos, que por otra parte se han empleado con frecuencia en otros artículos sobre gráficos, son: PLOT, MOVE, DRAW y LINE.

Al principio de un programa de gráficos se dispone el ordenador para ser utilizado bajo el modo de gráficos y se limpia la pantalla. Es interesante estructurar el programa de tal forma que todas las órdenes para dibujar sean agrupadas en subrutinas. Además de ser una práctica de programación de sonido, este acercamiento metódico sirve para prevenir el tener que reescribir secciones del programa que obtienen los mismos resultados. Y en vista de que todas las órdenes o mandos que realizan ciertas tareas son recogidos en una sección del programa, es una forma simple de desarrollar el programa y entenderlo. La estructuración dilata el programa, pero esto no es importante para las presentaciones estáticas y la gran flexibilidad que ofrecen; particularmente este retardo ofrece grandes ventajas para una aplicación como la de dibujos en estructura de alambre.

Para componer la primera etapa de estas rutinas de dibujos teclea esta sección de programación, pero no hagas RUN todavía, porque es incompleta.

```

9000 REM INICIO
9010 BORDER 4: PAPER 7: INK
    Ø: CLS : LET N=Ø
9070 RETURN
9100 REM MOVIMIENTO
9110 PLOT X,Y
9120 RETURN
9200 REM DIBUJO
9210 DRAW X-PEEK 23677,Y-
    PEEK 23678
9220 RETURN
    
```

Ahora, cuando quieras dibujar no necesitarás decirle al ordenador cómo mover el cursor o cómo marcar la pantalla para configurar cada forma diferente. En lugar de esto es mucho más simple definir cada forma nueva en términos de estas rutinas básicas.

La forma más simple para dibujar es la línea, así que aquí tienes una rutina

para hacerlo (todavía no podrás hacer RUN con el programa).

```
9500 REM LINEA
9510 LET X=XS: LET Y=YS: GO
SUB 9100
9520 LET X=XE: LET Y=YE: GO
SUB 9200
9550 RETURN
```

Esta rutina especifica una posición de inicio, coordenadas XS, YS, y una posición de final, con las coordenadas XE, YE para la línea. Algunos ordenadores requieren las órdenes DRAW o LINE; para otros esto está especificado en las rutinas que van desde la línea 9100 a la línea 9220.

La rutina de dibujo de líneas es la base de la mayor parte de los programas de dibujo o diseño de estructuras de alambre.

Con estas líneas se establece uno de los «bloques de construcción» básicos en el proceso de diseño: la rejilla.

DIBUJANDO UNA REJILLA

Para representar una superficie y cualquier irregularidad que ésta pueda contener, como hendiduras, colinas o valles, es mejor visualizarla no como un área continua, encerrada en un rectángulo, sino como una rejilla de líneas horizontales y verticales. Cualquier característica irregular en la superficie puede ser mostrada como distorsión de estas líneas. Ahora pasemos a la siguiente sección del programa, pero sin hacer RUN, para definir una rutina que permita dibujar una rejilla.

```
5000 LET JA=LW/NX
5010 LET XS=XA
5020 FOR J=0 TO NX
5025 LET YS=YA: LET XE=XS:
LET YE=YA+LH
5030 GO SUB 9500
5040 LET XS=XS+JA
5050 NEXT J
5060 LET JA=LH/NY
5070 LET YS=YA+LH
5080 FOR J=0 TO NY
5090 LET XS=XA+LW: LET
XE=XA: LET YE=YS
```

```
5100 GO SUB 9500
5110 LET YS=YS-JA
5120 NEXT J
5130 RETURN
```

Las coordenadas (XA, YB) especifican la esquina izquierda del fondo de la rejilla. LW especifica el ancho y LH la altura. NX especifica el número de divisiones horizontales, y NY el número de divisiones verticales. La variable JA especifica la distancia entre las líneas verticales y el bucle FOR... NEXT dibuja líneas horizontales escalonadas en esta distancia. El segundo bucle FOR... NEXT dibuja líneas verticales en pasos calculados por la línea 5060.

La rutina entre las líneas 5000 y 5080, dibuja líneas horizontales en un paso desde la izquierda a la derecha, y líneas verticales en un paso desde el fondo hasta arriba, ya que una rejilla de este tipo sólo puede trazar superficies planas. Por ejemplo, no puede representar irregularidades junto con la superficie.

Para poder ver la rejilla en la pantalla teclea estas líneas, que llaman a la rutina, y ejecuta RUN.

```
100 GO SUB 9000
175 LET XA=0: LET YA=0:
LET LW=255: LET
LH=175: LET NX=16: LET
NY=12
180 GO SUB 5000
190 STOP
```

Ahora ves una rejilla que llena toda la pantalla. Para ver cuán versátil es el programa, realiza estos cambios:

```
175 LET XA=10: LET YA=10:
LET LW=240: LET
LH=144: LET NX=1:
LET NY=1
```

En esta ocasión se dibuja una caja rectangular porque se ha especificado una rejilla con una división horizontal y una vertical. Si se proporcionan valores apropiados a NX y NY, como arriba, se puede dibujar una rejilla en la que el número de divisiones horizontales sea diferente del número de divisiones verticales.

Realiza ahora nuevos cambios:

```
175 LET XA=0: LET YA=0: LET
LW=160: LET LH=144:
LET NX=15: LET NY=10
```

La rejilla ya no llena la pantalla y en su lugar encontramos un cuadrado en la parte inferior izquierda del área. La forma del cuadrado se obtiene proporcionando los valores apropiados a LW y LH, y el número de divisiones está especificado por NX y NY, igual que arriba.

DIBUJAR CIRCULOS

Las rejillas rectangulares no son los únicos «bloques de construcción» que pueden ser utilizados para dibujar estructuras de alambre; con frecuencia, es bastante útil para poder dibujar círculos. Algunos ordenadores poseen controles que te permiten dibujar círculos si se especifica el centro y el radio.

Sin embargo, estas órdenes directas no permiten el grado de control que se necesita para poder dibujar imágenes tridimensionales. Con la perspectiva, un círculo observado desde un punto podría parecer una elipse desde otro, o semejar una curva. Aunque la instrucción CIRCLE puede ser utilizada para dibujar elipses, no puede alcanzar ni competir con la tercera dimensión que se requiere para obtener formas naturales. Por lo tanto, es mejor poder definir una función general.

Una de las maneras de dibujar un círculo es la de emplear una serie de secciones pequeñas de líneas rectas. Partiendo de la base de que las líneas son cortas, la circunferencia del círculo aparecerá como una curva continua, pero cuanto más pequeñas sean las líneas, se necesitarán más y por lo tanto se prolongará el tiempo de dibujar. Aquí presentamos una rutina para dibujar un círculo de radio R, con el centro en (XS, YS); pero no ejecutes el programa todavía:

```
6000 IF N=0 THEN LET
N=20+INT (R/10)
6020 LET JA=2*PI/N
6050 LET XR=XS: LET YR=YS
```

```

6060 LET JB=0: LET XS=XS+R
6070 FOR J=2 TO N
6080 LET JB=JB+JA
6090 LET XE=XR+R*COS JB:
LET YE=YR+R*SIN JB: GO
SUB 9500
6100 LET XS=XE: LET YS=YE
6110 NEXT J
6120 LET XE=XR+R: LET
YE=YR: GO SUB 9500
6130 LET XS=XR: LET YS=YR
6160 RETURN
    
```

La variable N dispone el número de segmentos de línea que deben ser utilizados para la circunferencia del círculo. Si especificas $N=0$, la línea 6000 calcula cuántos segmentos se necesitan para el círculo más uniforme, teniendo en cuenta el tamaño de la visualización en pantalla.

La línea 6020 calcula el ángulo de cada segmento de línea en la circunferencia. La línea 6050 mueve el cursor a una posición de la circunferencia. El bucle FOR...NEXT dibuja cada segmento de línea, excepto el último, que es dibujado por la línea 6120 para asegurar que la última línea se una a la primera.

Para observar cómo funciona la rutina, borra la línea 180 y añade las siguientes líneas para llamarla:

```

150 FOR R=20 TO 70 STEP 10
155 LET XS=128: LET YS=
102: LET N=24
160 GO SUB 6000
170 NEXT R
    
```

La presentación en tu pantalla debe mostrar ahora círculos concéntricos en el centro de la misma. Al igual que con la rutina de rejilla, puedes variar los parámetros del programa que llaman a la rutina para cambiar la presentación. La línea 150 asigna el radio del primer círculo y la cantidad en que es incrementado para darle el radio de círculos sucesivos. La línea 155 especifica el centro de los círculos y el número de las secciones de línea en la circunferencia. A modo de prueba puedes modificar estos valores y observar el efecto en la presentación.

Si estás preguntándote qué es lo que ha pasado con la rutina de la rejilla, hay que decir que permanece en la memoria, pero desde que has escrito las líneas de código que llaman a la rutina el ordenador no puede mostrarlas. Rutinas como éstas pueden ser reunidas en una biblioteca de útiles rutinas gráficas de líneas, que pueden ser empleadas cuando sea necesario. Pueden ser utilizadas en cualquier tipo de programas, así como en dibujos de estructuras de alambre. También se pueden añadir nuevas rutinas en la medida en que se requiera. Una vez que estén en el ordenador hay que proceder a almacenarlas en cinta o

P y R

¿Puedo añadir color para destacar mis dibujos de estructura de alambre?

Por lo general los dibujos de estructuras de alambre se hacen sólo en dos colores, normalmente en blanco y negro. Una de las razones para esto es que muchos colores complicarían la imagen. Sin embargo, la cuestión más importante es que la adición de colores reduce la resolución de la imagen y en algunos ordenadores, como es el caso del Spectrum, puede causar problemas cuando un color se superpone sobre otro. Hacia el final de esta serie de artículos hablaremos sobre el empleo de los colores, pero no hay ninguna razón para que no puedas experimentar con colores ahora.

disco. Cuando desees utilizarlas, vuelve a cargarlas en la memoria, y si es necesario, toma varias rutinas.



En la próxima oportunidad podrás ver cómo se utilizan estas rutinas para crear dibujos de estructuras de alambre en tres dimensiones.

CONVENCIONES VISUALES

En teoría, por lo menos, no tiene sentido decir que uno puede crear un objeto tridimensional (por ejemplo una caja) en un medio de dos dimensiones, ya sea en un trozo de papel o en la pantalla del televisor. Pero lo que sí se puede hacer es emplear una convención visual de tal modo que nuestros ojos puedan percibir o «leer» una serie de líneas como representación de un objeto sólido.

La pintura de un paisaje, por ejemplo, aparenta tener una profundidad y una distancia porque el artista ha utilizado un truco visual: la perspectiva. La perspectiva depende del hecho de que nosotros comprendemos que los objetos se hacen pequeños cuando se mueven en la distancia, y que las líneas paralelas parecen converger cuando se encuentran más alejadas del observador.

La perspectiva no es la única con-

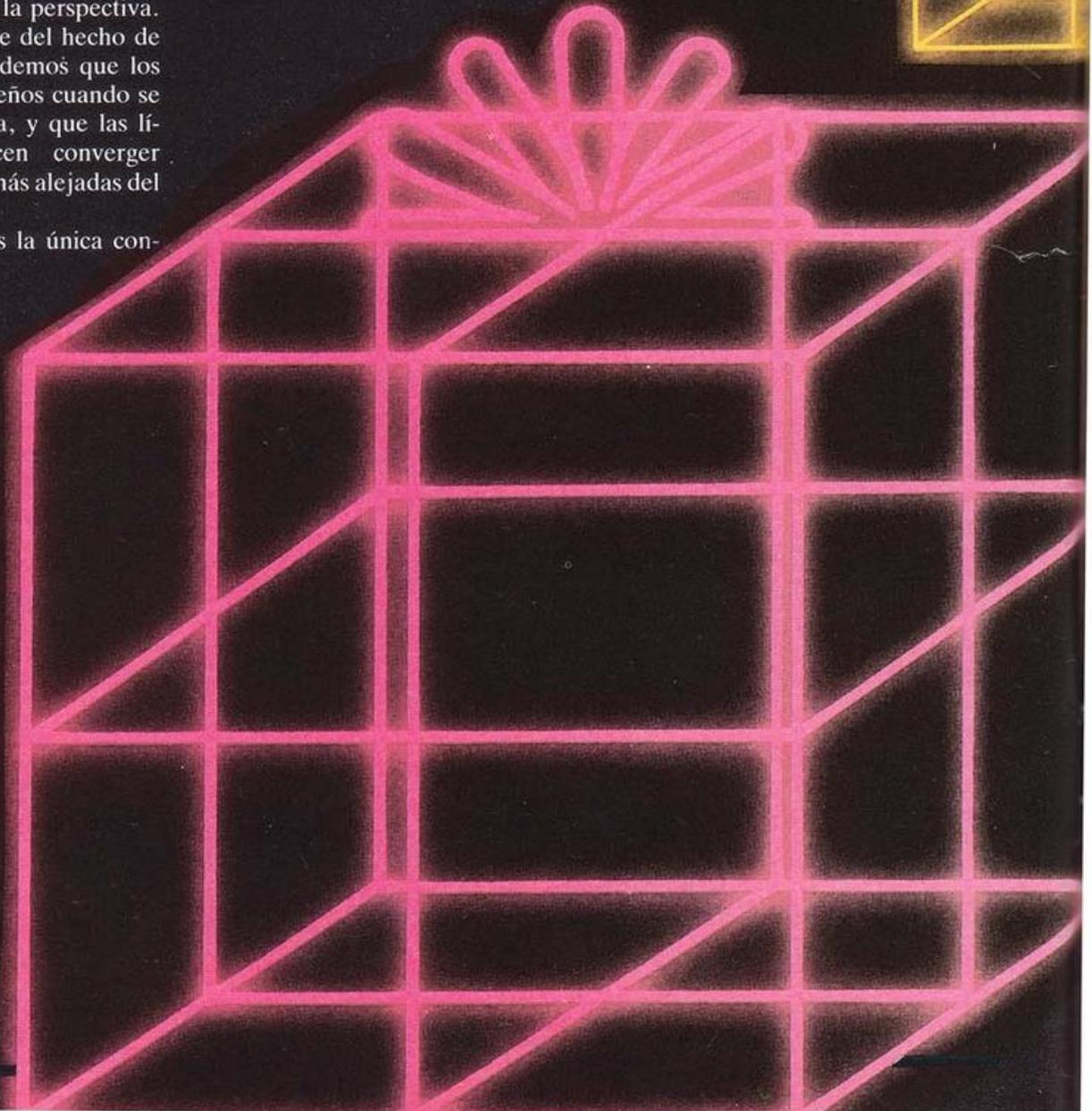
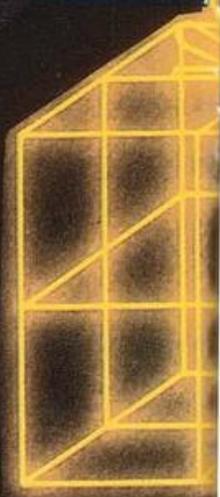
vención o regla que se utiliza en este tipo de representación. En los dibujos técnicos es habitual emplear una convención denominada proyección isométrica. En este caso, las líneas que se alejan desde la posición del observador son dibujadas en un ángulo, como en un dibujo en perspectiva.

Sin embargo, a diferencia de los dibujos en perspectiva estas líneas no convergen, ni los objetos se hacen pequeños a medida que se alejan. Esto tiene muchas ventajas en las aplicaciones técnicas, como pueden ser el tomar medidas directas de cualquier línea ya que su escala no está afectada por esta dirección.

Aunque se puede representar una perspectiva en una visualización por ordenador, como veremos más adelante, es más difícil de crear que una

proyección isométrica. Los dibujos isométricos se basan simplemente en componer un tercer eje en la pantalla.

Ya tienes un eje X (horizontal) y un eje Y (vertical). Todo lo que necesitas es un tercer eje, el Z, que está en un ángulo asignado a los dos primeros. Cualquier línea que sea dibujada en

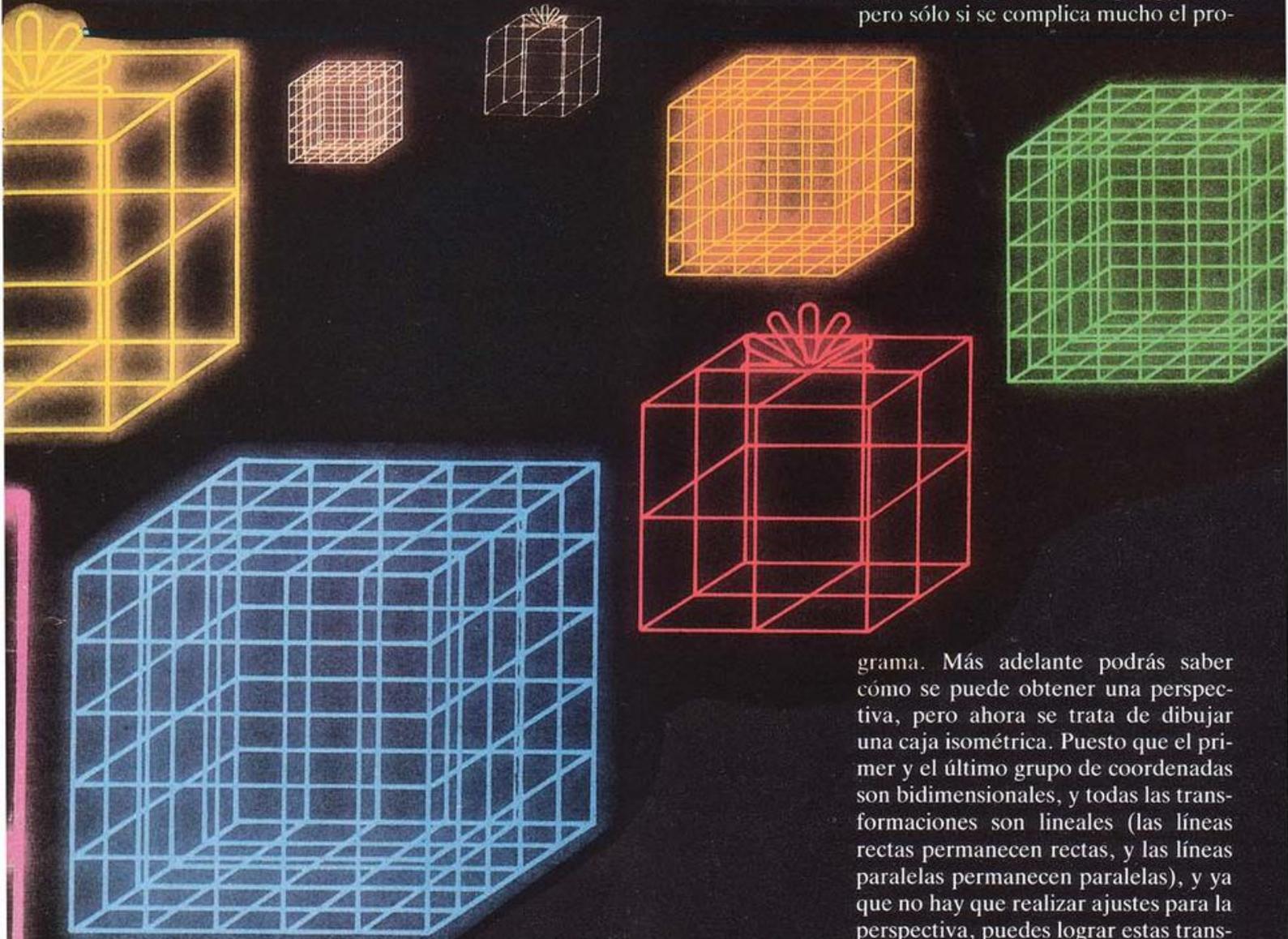


este ángulo se entiende que representa a una línea que va desde (o hacia) el observador. Esto es muy sencillo de entender si miras los diagramas de los ejes (fig. 1).

tarse al ángulo del eje Z. Esto requiere una serie de transformaciones de las coordenadas basadas en complicados principios matemáticos. Pero no te preocupes, tu ordenador puede hacerlo fácilmente.

nes (XP, YP) para que puedan ser visualizadas en pantalla. Es fácil seguir estos pasos si se observa el diagrama (fig. 2), que muestra una forma modificada en seis lados.

Durante esta transformación es posible tomar en cuenta una perspectiva, pero sólo si se complica mucho el pro-



TRANSFORMANDO EL PROGRAMA

Utilizando esta convención o regla, una de las formas de producir imágenes de estructuras de alambre en tres dimensiones es construirlas a partir de dos formas de dos dimensiones, empleando rutinas como las generadas para las rejillas y los círculos, tal y como hemos visto más arriba. Algunas de estas formas bidimensionales necesitarán ser distorsionadas para ajus-

Lo primero que debes hacer es transformar las coordenadas de dos dimensiones (X, Y), a coordenadas tridimensionales (X', Y', Z') en el plano deseado. Esto significa que tienes que especificar a qué distancia del eje Z debería estar el plano. Después necesitas transformar las coordenadas en 3-D (X', Y', Z') en un nuevo grupo de coordenadas 3-D (XE, YE, ZE) basadas en la posición y dirección desde la que estás observando al objeto. Finalmente, tienes que transformar estas coordenadas 3-D (XE, YE, ZE) en coordenadas de dos dimensio-

grama. Más adelante podrás saber cómo se puede obtener una perspectiva, pero ahora se trata de dibujar una caja isométrica. Puesto que el primer y el último grupo de coordenadas son bidimensionales, y todas las transformaciones son lineales (las líneas rectas permanecen rectas, y las líneas paralelas permanecen paralelas), y ya que no hay que realizar ajustes para la perspectiva, puedes lograr estas transformaciones cambiando las rutinas originales de Inicialización y de Dibujo.

Si has almacenado una copia del programa que explicábamos en la primera parte del artículo, cárgalo y después tecléa las líneas siguientes. Es una buena idea almacenar el listado completo porque necesitarás llamar algunas de las rutinas en la medida en que el programa se desarrolle en artículos sucesivos. Si no almacenaste el programa, tecléa las líneas 50000 a 51300. Estas líneas forman la rutina para dibujar una rejilla.

Ahora teclea las siguientes líneas.

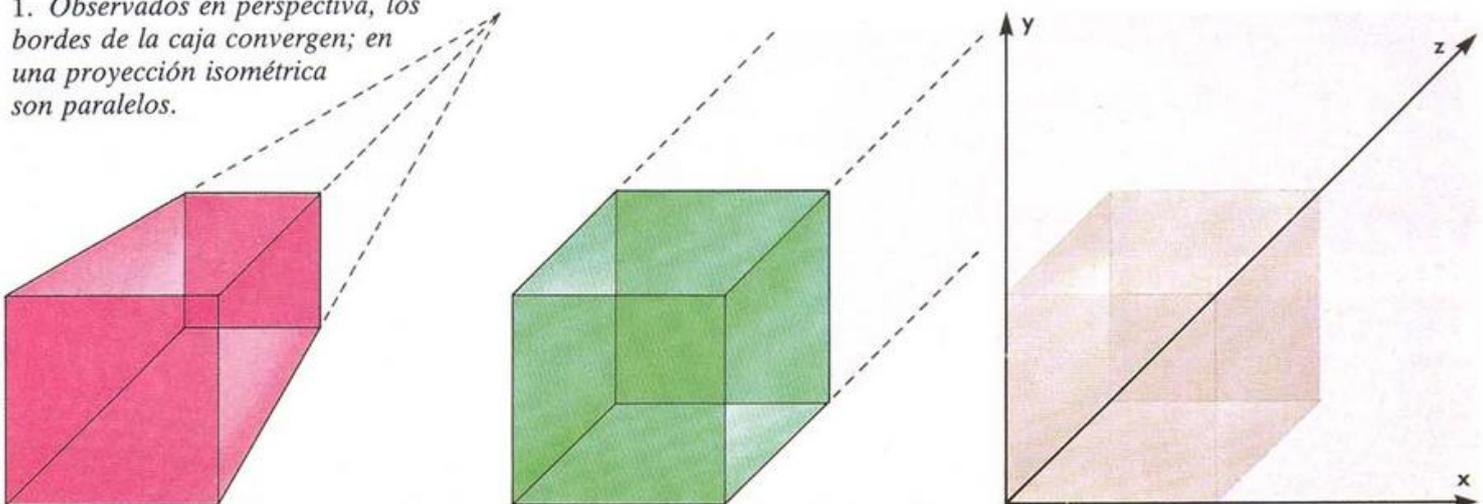
```

90000 LET XX=1: LET XY=0: LET
  YX=0: LET YY=1: LET
  XO=0: LET YO=0
90100 BORDER 4: PAPER 7: INK
  0: CLS
90700 RETURN
91000 REM MOVIMIENTO
91100 PLOT XS*XX+XY*YS+XO+
  127,
  YX*XS+YY*YS+YO+76
91200 RETURN
92000 REM DIBUJO
92100 >DRAW XE*XX+XY*YE+
  XO+127-PEEK 23677,
  YX*XE+YY*YE+YO+76-
PEEK 23678
92200 RETURN
95000 REM LINEA
95100 GO SUB 91000
95200 GO SUB 92000
95500 RETURN
  
```

La rutina de Inicio desde la línea 90000 hasta la línea 90700, pone en los valores iniciales la transformación de variables (XX,XY,YX,YY,XO,YO) a valores que hacen que la rutina de dibujo trabaje como antes.

La rutina mueve los ejes de tal forma que el origen esté en el centro de la pantalla. Se traslada el punto de origen porque es conveniente pensar en los ojos del observador como situados directamente sobre el origen en el centro de la pantalla. Este punto se verá con más claridad en el próximo

1. Observados en perspectiva, los bordes de la caja convergen; en una proyección isométrica son paralelos.



artículo, cuando puedas ver cómo se puede cambiar el punto de observación y cómo se puede añadir perspectiva.

Para poder ver el efecto de cada una de estas variables de transformación, y para tener una mejor percepción de lo que sucede, cambia las Líneas 1000 a 180, como se señala abajo, para llamar a la rutina de Rejilla:

```

1000 GO SUB 90000
1200 CLS
1300 LET XA=-40: LET
  YA=-20: LET LW=40:
  LET LH=40: LET NX=5:
  LET NY=5
135 GO SUB 50000
140 LET XA=0: LET YA=-20:
  LET LW=40: LET LH=40:
  LET NX=10: LET NY=10
145 GO SUB 50000
160 INPUT «ENTER XX,XY,YX,
  YY,XO,YO»,XX,XY,YX,YY,
  XO,YO
170 GO TO 1200
180 STOP
  
```

Ejecuta el programa para poder ver dos pequeñas rejillas cuadradas en el centro de la pantalla, con una línea que te solicita que introduzcas XX, XY, YX y así sucesivamente.

Éstas son las variables de transformación.

XX e YY colocan los factores de escala en las direcciones horizontal y vertical. Si se les da valores negativos se produce una reflexión cerca del eje.

XO e YO realizan el balance para situar las rejillas en diferentes posiciones en la pantalla. XY e YX permiten las rotaciones y distorsiones (divididas) de la imagen. Las variables de transformación tienen los siguientes valores iniciales: XX=1, XY=0, YX=0, YY=1, XO=0 y YO=0. Para transformar la imagen necesitas dar a estas variables valores diferentes, y ésta es la razón de la expresión INPUT en la línea 160. A continuación te presentamos un grupo de valores para que puedas hacer una prueba. Debes pulsar ENTER en cada uno de ellos:

-1,0,0,1,0,0 provoca una reflexión cerca del centro del eje vertical de la pantalla.

0,5,0,0,2,0,0 gradúa la imagen en dos, verticalmente, y a la mitad, horizontalmente.

1,0,0,1,50,40 mueve la imagen 100 unidades a la derecha y 50 unidades hacia abajo o arriba.

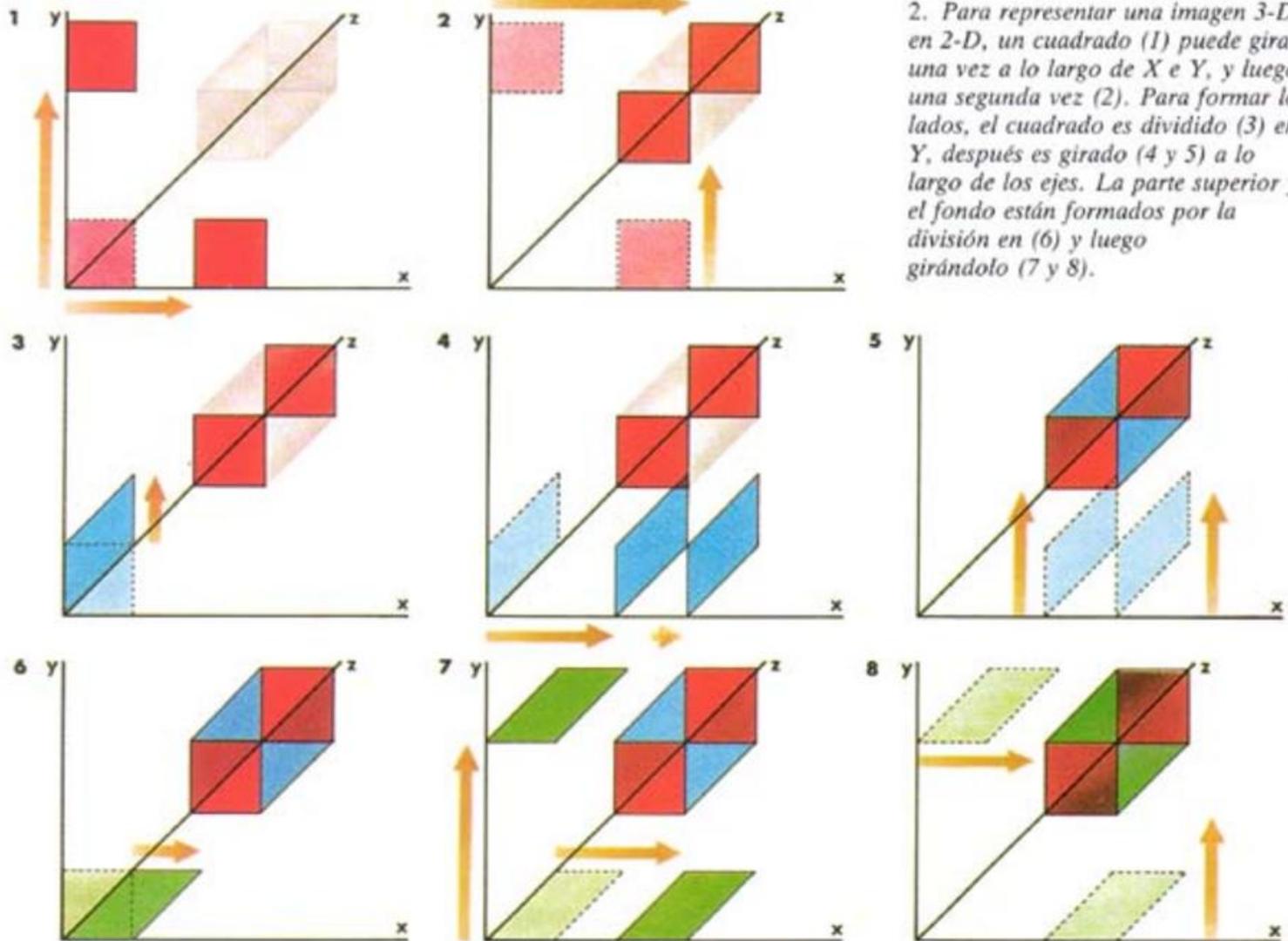
0, -1,1,0,0,0 hace rotar la imagen 90 grados en dirección contraria a las agujas del reloj cerca del centro de la pantalla.

1,0,5,0,1,0,0 divide la imagen horizontalmente.

1,0,5,0,5,1,0,0 divide las imágenes tanto horizontal como verticalmente.

Hay que pulsar SPACE después que se haya dibujado cada imagen.

Para poder predecir los resultados con precisión necesitas entender matrices de aritmética. Así que aunque puedas variar la forma de esta imagen, deberás saber cómo se pueden com-



2. Para representar una imagen 3-D en 2-D, un cuadrado (1) puede girar una vez a lo largo de X e Y, y luego una segunda vez (2). Para formar los lados, el cuadrado es dividido (3) en Y, después es girado (4 y 5) a lo largo de los ejes. La parte superior y el fondo están formados por la división en (6) y luego girándolo (7 y 8).

binar los valores y las rejillas para obtener una imagen 3-D completa. Y esto es precisamente lo que hace el siguiente programa de acuerdo a los procedimientos que se indican a continuación.

Borra las líneas 135, 145 y 175, y tecla:

```

110 GO SUB 9000
120 LET L=108: LET N=4
130 LET DX=.45*L: LET
    DY=.3*L
140 LET XN=-(L+DX)/2
150 LET YN=-(L+DY)/2
160 LET XO=XN
170 LET YO=YN
180 GO SUB 5000
190 LET XO=XN+DX
200 LET YO=YN+DY
210 GO SUB 5000
220 LET XY=DX/L: LET XO=XN

```

```

230 LET YY=DY/L: LET YO=YN
240 GO SUB 5000
250 LET YO=YN+L
260 GO SUB 5000
270 LET XX=DX/L: LET XY=0
280 LET YX=DY/L: LET YY=1:
    LET YO=YN
290 GO SUB 5000
300 LET XO=XN+L
310 GO SUB 5000
320 STOP
500 LET XA=0: LET YA=0: LET
    LW=L: LET LH=L: LET
    NX=N: LET NY=N
510 GO SUB 5000
520 RETURN

```

Cuando se pulsa RUN y se ejecuta el programa se debe ver un cubo en el centro de la pantalla. Todos los lados están dibujados por la misma rutina,

que es, simplemente, la que se utiliza para dibujar una rejilla con variables apropiadas para transformarla en la forma correspondiente. Pero se han de colocar otras dos variables antes de que el cubo pueda ser dibujado. Éstas son la variable L, que marca el tamaño del cubo, y la variable N, que dispone el número de cuadrados de rejilla. La variable DX especifica hasta qué punto de la izquierda o de la derecha de la parte posterior del cubo es desplazada desde el frente, y DY especifica cuál es la extensión de desplazamiento de arriba hacia abajo de los mismos lados. DX y DY aseguran, además, que el cubo aparezca proporcionado.

Otro par de variables —XN en la línea 140 e YN en la línea 150— sitúan balances para la posición de inicio de la parte frontal del cubo. Estas compensaciones son trasladadas a XO en

P y R

¿Puedo cambiar este programa para dibujar imágenes diferentes?

La rutina básica en este programa es la que dibuja la rejilla. Utiliza la sección que dibuja líneas, que podrías utilizar en un nuevo programa tuyo para, por ejemplo, dibujar triángulos. Esto no diferiría demasiado de la rutina de Rejilla, pero necesitarías más atención si hicieras el intento de llamarla varias veces para dibujar una forma como la de tetraedro —una figura de cuatro lados triangulares—.

Para dibujar la base los valores dados para la transformación de variables serían los mismos de la caja, pero tendrías que abandonar el dibujo del lado que no necesitas, es decir, el de la parte superior de la caja. Una tarea más complicada es cambiar las variables de transformación (líneas 270 a 300) para dar un ángulo a la rejilla triangular y situarla para poder dibujar los tres lados.

Sin embargo, conviene resaltar que es mucho más simple cambiar el programa para hacer que dibuje cajas de diferentes tamaños y formas no cerradas. Por ejemplo, cambia el valor de L en la línea 120 para variar el tamaño de la caja, y cambia al mismo tiempo el valor de N, en la misma línea.

A grandes valores de N, la caja es virtualmente sólida. Para abandonar el dibujo de los lados frontal y posterior, borra la línea 311 (que dibuja un arco en la caja) e inserta:

```
155 GOTO 220, después
    RUN.
```

la línea 160 y a YO en la línea 170, luego se puede dibujar la parte posterior del cubo (en la línea 180).

La parte frontal y la parte posterior del cubo son idénticas, así que para dibujar el frente del cubo el único requisito es incrementar la compensación en la dirección X (línea 190) y en la dirección Y (línea 200). Estas compensaciones alinean la posición de la parte posterior con los lados que has dibujado.

La línea 240 dibuja la parte superior, pero antes que nada hay que transformar la rejilla —XY e YY— y las compensaciones deben ser redefinidas en las líneas 220 y 230. La misma vista de la rejilla se emplea para la parte del fondo, que sólo necesita ser graduada y posicionada (línea 250) antes de ser dibujada. De igual forma, desde las líneas 270 a 310, transforman la rejilla para dibujar el lado derecho, y luego vuelven a su posición para dibujar el lado izquierdo.

Todas las rejillas cuadradas de cada lado son de la misma forma y tamaño y todas las líneas paralelas en cada rejilla permanecen paralelas.

La razón se encuentra en que no se ha hecho todavía ningún ajuste de perspectiva. Más aún, es difícil adaptar el programa para mostrar otras vistas del cubo.

Más adelante veremos cómo hacer dibujos de estructuras de alambre en perspectiva, que pueden ser observados desde cualquier posición. Mientras tanto, cambia las líneas 120 y 130, abajo, y añade estas pocas líneas para envolver el cubo como si fuera un regalo. No te olvides de pulsar SAVE y hacer una copia:

MICROTIPS

Dibujos fuera de pantalla

Si cambias la línea 120, de tal forma que L tenga valores más grandes, seguramente te encontrarás que el programa se bloquea y te da un mensaje de error. Esto sucede porque algunos ordenadores no pueden dibujar una línea si uno de los puntos especificados está fuera del alcance de la pantalla. En teoría, si la imagen está por entero en la pantalla, no deben presentarse problemas. El programa no está limitado por este problema, ya que en un próximo artículo se hace una revisión para asegurar que la máquina dibuje sólo las líneas que son válidas. Gracias a ello, el programa no es interrumpido con mensajes de error.

```
120 LET L=60: LET N=2
130 LET DX=.3*L: LET
    DY=.4*L
131 GO SUB 9920
9920 FOR M=0 TO PI STEP .01
9930 LET D=COS (6*M)
9940 LET S=D*COS M: LET
    T=D*SIN M
9950 LET S=S*20: LET T=ABS
    (T*20)
9960 PLOT S+127,T+106
9970 NEXT M
9980 RETURN
```

NO OLVIDES EL TELEFONO... ☎

Cuando, por cualquier motivo, nos escribas, no olvides indicar tu número de teléfono. Así nos será más fácil y rápido ponernos en contacto contigo. Gracias.

MUSICA Y SIMULACIÓN DE PARTITURAS

Beethoven es un programa que permite la composición de música en el Spectrum mediante la simulación de una partitura sobre la pantalla del ordenador.

La idea inicial era la de crear un programa educativo, no de utilidades. Las razones de esta decisión radican por un lado en que el Spectrum tiene una capacidad de generar música muy reducida. Por otra parte consideramos que el ordenador es una gran arma educativa para abordar y simplificar los temas difíciles e intuitivos, como la música.

De este modo creímos que crear un programa que incluyese las normas básicas del solfeo, asequibles, sencillas, con las que componer música o simplemente copiar de una partitura, sería de gran valor en manos de un profesor de instituto o incluso en las de cualquier chaval que, jugando con su ordenador, aprendería estos conceptos básicos, casi sin darse cuenta (el tiempo, las escalas, los tonos, los compases, etc.). Así nació la idea del programa Beethoven, que por cierto, en esos momentos todavía no tenía nombre.

De unos cuantos libros de solfeo básico tomamos los puntos y cualidades básicas que debía tener el programa y fuimos implementándolas a la idea original.

El programa está completamente escrito en BASIC, lo que redundaba en un grado de lentitud elevado, pero que, dada su labor educativa, creemos que debemos de aceptar para obtener los beneficios de una fácil modificación por parte de los usuarios y de una mayor comprensión del programa, lo que apoya esa labor pedagógica.

Como ya hemos dicho simula un pentagrama sobre el que se escriben directamente las notas musicales mediante el teclado.

El programa incluye las posibilidades de modificar la velocidad de la audición, el tono (octava) de la misma, la introducción de silencios, tipos de compás (tres tipos), alteraciones en armadura y en composición.

Además el programa incluye las siguientes utilidades: La posibilidad de escuchar música en cualquier momento, de corregir cualquier error en el modo editor, la de continuar con una composición ya iniciada y por último la de grabar en cassette la música ya compuesta y volverla a cargar en el programa en cualquier momento.

Todo el control del programa se lleva con facilidad por el teclado y trata de ser aclarado por mensajes que aparecen en pantalla en un lugar determinado para ello.

El programa, como se puede comprobar, no incluye todas las posibilidades del solfeo, pues su misión no lo precisa y el BASIC no lo permite. No obstante, al estar escrito en BASIC, está abierto a la introducción de nuevas funciones e ideas o a la mejora de las que ya posee por el usuario, en vistas a adaptarlo para sus propósitos.

Asimismo incluimos varias composiciones realizadas por nosotros (copiadas de partituras de cursos de solfeo), utilizando el programa Beethoven, como ejemplo de su funcionamiento.

INSTRUCCIONES DE UTILIZACIÓN

El programa, una vez en el ordenador, funciona simplemente con RUN. Tras realizarlo el ordenador, después de unos momentos que precisa para cargar los caracteres gráficos, presenta la pantalla del programa.

Tras teclear RUN la pantalla nos indicará que estamos en la rutina de introducción de armadura, ésta está constituida por los apuntes iniciales que se realizan sobre el pentagrama para indicar el compás y la escala utilizada. Pulsando "n" el ordenador nos dará una armadura estándar, con un compás de 4/4. La letra "n" también se puede pulsar cuando vayamos a escuchar o modificar una composición anterior. Si pulsamos cualquier otra tecla nos pedirá el compás primero, dándonos a elegir entre tres: 3/4, 2/4 y 4/4. Una vez elegida la opción nos pedirá que introduzcamos alteraciones o variaciones con el control de pentagrama, éste tiene las teclas "w" para subir y "s" para bajar, pulsando la tecla correspondiente a


```

1 REM ** MEGA SOFT GROUP **
2 REM BEETHOVEN
3 REM RAFAEL PEREZ
4 REM JAVIER CORDOBA
5 REM ***** 1986 *****
6 DATA "DO",0,"RE",2,"MI",4,"FA",5,"SOL",7,"LA",9,"SI",11
8 DATA 0,0,24,36,24,0,0,0,0,24,36,28,4,4,4,4,0,24,60,28,4,4,4,4
10 DATA 0,24,60,28,4,4,4,28,0,24,60,28,4,28,4,28,0,32,32,56,40,56,8,0
12 DATA 0,0,0,24,24,0,0,0,0,0,0,255,0,0,0,0,0,124,56,56,0,0,0
14 DATA 0,0,56,56,124,0,0,0,0,16,24,16,8,48,48,24,0,0,48,48,16,32,0,0,0,12,12,4,52,56,
16,16
16 DATA 8,8,8,8,8,8,8,8,0,16,16,16,84,56,16,0
59 REM inicializacion y variables
60 LET t=1
65 LET ctrlcp=0
70 LET n=0: LET lm=15
75 LET sl=0
80 DIM M(lm,5)
90 DIM N$(7,3): DIM F(7): DIM Z(7): DIM X$(3,3): DIM Y(7): DIM B(7)
100 PAPER 4: INK 0: BORDER 4: CLS
110 FOR a=1 TO 7: READ N$(a): READ F(a): LET Y(a)=F(a): NEXT a
120 FOR a=USR "a" TO USR "p"-1: READ b: POKE a,b: NEXT a
130 LET z$=" "
140 LET u=7: LET am=0: LET n2=7: LET tabsc=193: LET mod=0: LET gctr=0: LET
editmode=0: LET octava=2: LET fv=0: LET sl=0: LET gcd=0: LET rcd=0
150 LET X$(1)="4/4": LET X$(2)="3/4": LET X$(3)="2/4"
160 LET max=0
200 GO SUB 400: GO TO 700
399 REM DISENO DE PANTALLA
400 PRINT AT 1,22; PAPER 6; " TIEMPO ";AT 2,22;" <O P> ";AT 3,22;" ";AT 4,22;" ";
AT 5,22;" ";AT 5,24; INVERSE 1;"T ";AT 5, 29; INVERSE 0;" ";AT 6,22;"
410 PLOT 183,148: DRAW 65,0: FOR a=0 TO 8: PLOT 183+8*a,146: NEXT a
420 PRINT AT 8,22; PAPER 6;"OCT";AT 9,22;"L H"; BRIGHT 1;AT 9,23;"M"
430 PRINT AT 8,27; PAPER 7; "NOTA";AT 9,27;"SI"
440 PRINT AT 11,22; PAPER 7; INVERSE 1;"BEETHOVEN";AT 17, 22; "MEGA SOFT"; INVERSE
0; PAPER 4;AT 13,22;"JAVIER C.";AT 15,22;"RAFAEL P."
450 FOR a=1 TO 3: PRINT AT 14+a,1; PAPER 0;" ": NEXT a
460 PRINT AT 19,1;"a b c d e f g h i A B C D E";AT 20,1;" o p r s t u v w x y z | "
470 PRINT AT 1,1; PAPER 6; INVERSE 1;" ↑";AT 2,1;"w";AT 12,1;"s";AT 13,1;" ↓"
480 GO SUB 500
486 PRINT #1; "1PLAY 2 EDIT 3 PRINT 4 LOAD 5SAVE"
490 PRINT AT 5,26; PAPER 6; INVERSE 1;tabsc-183: GO SUB 4600
495 FOR a=1 TO 13: FOR b=2 TO 3: PRINT AT a,b; PAPER 5;" ": NEXT b: NEXT a; RETURN
499 REM BORRADO DE VENTANA
500 FOR a=1 TO 13: FOR b=4 TO 20: PRINT AT a,b; PAPER 5;" ": NEXT b: NEXT a
510 FOR a=3 TO 11 STEP 2: FOR b=4 TO 20: PRINT AT a,b; PAPE R 5; " ": NEXT b: NEXT a
520 RETURN
550 GO SUB 500: PRINT AT fv,cv; PAPER 5; CHR$( M(n,4)): LET cv=cv+2
560 RETURN
699 REM ARMADURA

```

```

700 LET a$="INTRODUCE LA ARMADURA, QUE INDICA EL COMPAS (PULSA)": GO SUB 5000:
    PAUSE 0
710 LET a$="Y LAS ALTERACIONES. PULSA N SI QUIERES ARMADURA STANDARD.": GO SUB
    5000: PAUSE 0
720 IF INKEY$="n" THEN LET compas=1: LET fv=7: LET gc=1: LET cv=5: GO SUB 4500: GO
    TO 850
730 LET a$="COMPAS: PULSA 1=(4/4) 2=(3/4) 3=(2/4)": GO SUB 5000: PAUSE 0
740 IF INKEY$="1" THEN LET compas=1: LET gc=1: GO TO 780
750 IF INKEY$="2" THEN LET compas=.75: LET gc=2: GO TO 780
760 IF INKEY$="3" THEN LET compas=.5: LET gc=3: GO TO 780
770 GO TO 730
780 LET a$="INTRODUCE ALTERACIONES CON CONTROL DE PENTAGRAMA": GO SUB 5000:
    PAUSE 0: LET a$="PULSA N PARA TERMINAR": GO SUB 5000
785 LET fv=7: LET cv=3: PRINT AT 7,3; PAPER 5; FLASH 1; OVER 1;" "
790 PAUSE 0: LET a$=INKEY$: GO SUB 2100: GO SUB 2000
800 IF INKEY$="h" THEN PRINT AT fv,cv; PAPER 5;"#": GO SUB 2100: LET F(n2)=F(n2)+1
810 IF INKEY$="i" THEN PRINT AT fv,cv; PAPER 5;"b": GO SUB 2100: LET F(n2)=F(n2)-1
815 GO SUB 4500
820 IF INKEY$="n" THEN PRINT AT fv,cv; PAPER 5; OVER 1; FLASH 0;" ": GO TO 850
830 GO TO 790
850 PRINT PAPER 5;AT 3,2;X$(gc)(1);AT 7,2;X$(gc)(2);AT 11,2;X$(gc)(3): LET a$="HA IDO
    TODO BIEN (S/N)": GO SUB 5000: PAUSE 0: IF INKEY$="n" THEN RUN
899 REM control de teclado
900 LET cv=5: LET a$="AHORA ESTAS EN EL CONTROL DE TECLADO": GO SUB 5000
905 PRINT PAPER 5; OVER 1; FLASH 1;AT fv,cv;" ": PAUSE 0: GO SUB 2100
910 LET a$=INKEY$
920 LET cd=CODE a$
930 IF cd>=97 AND cd<=101 THEN LET mod=0: LET sl=0: LET rcd=cd: GO SUB 2500: GO
    SUB 4000
940 IF cd<=65 AND cd<=69 THEN LET mod=0: LET sl=1: LET rcd=cd+32: GO SUB 2500: GO
    SUB 4000
945 IF a$="f" THEN PRINT AT fv,cv-1; PAPER 5;"k": GO SUB 2100: LET F(n2)=Y(n2): LET
    B(n2)=0
950 IF cd=105 THEN PRINT AT fv,cv-1; PAPER 5;"b": GO SUB 2100: LET F(n2)=F(n2)-1: LET
    B(n2)=1
955 IF cd=104 THEN PRINT AT fv,cv-1; PAPER 5;"#": GO SUB 2100: LET F(n2)=F(n2)+1: LET
    B(n2)=2
960 IF cd=103 THEN LET gcd=cd: PRINT AT fv,cv-1; PAPER 5;"."
965 IF a$="3" THEN GO SUB 1400
970 IF a$="w" OR a$="s" THEN GO SUB 2000
975 IF mod=1 THEN GO TO 905
980 IF editmode=1 THEN LET max=gmax: GO TO 1300
990 IF a$="2" AND max>=1 THEN LET n=1: GO SUB 500: LET gctr=ctrlcp: GO TO 1300
1010 IF a$="o" OR a$="p" THEN GO SUB 1900
1020 IF a$="1" THEN GO SUB 1200
1030 IF a$="M" OR a$="H" OR a$="L" THEN GO SUB 1700
1040 IF a$="5" THEN GO SUB 1500

```

NOTA: Por las obligadas limitaciones de espacio, este instructivo programa acabará en el próximo número.

ADIVINA MIS PALABRAS

■	UN JUEGO PARA DOS JUGADORES
■	PREPARACION DE LA PANTALLA
■	LAS REGLAS DEL JUEGO
■	VALORES DE LAS LETRAS
■	ESTRATEGIA

Introduce una buena palabra para los juegos de ordenador docentes. El juego de las palabras de INPUT es adecuado para todas las edades, puede hacerse tan difícil o tan fácil como desees, y es increíblemente adictivo.

Los juegos de ordenador no tienen por qué ser únicamente recreativos

como los juegos para salón o como algunos de simulación domésticos, sino que también pueden ser docentes.

«El Ahorcado» es el conocido juego que puede convertirse para que corra en un ordenador. Este juego puede ayudar a la gente en su expresión, en sus conocimientos generales, etc. Elige un tema como el de la ingeniería química y pronto dominarás una gran parte de su terminología.

El juego de palabras de INPUT es de este estilo, es para dos jugadores y se basa en adivinar palabras o frases. Es más interesante y divertido que «El Ahorcado», y es igual o más educativo.

Puedes jugarlo como «El Ahorcado», limitado a un determinado tema, puedes utilizar palabras con un determinado número de letras, emplear citas de Cervantes o lo que mejor te parezca.



EL JUEGO

Primero introduce los nombres de los dos jugadores. Después tienes la opción de elegir el número de palabras de las frases que introduce cada persona. Una interesante faceta de este juego es que cuanto más largas son las frases, algunas veces son más fáciles de adivinar, porque mayor es el número de pistas. Prueba y verás.

Una vez establecido el número de palabras, tendrás que elegir el número de vueltas que deberá efectuar el programa.

A continuación, el primer jugador deberá pensar en una frase e introducirla. No es necesario que el oponente esté encerrado y gritando en un armario mientras se efectúa la introducción, puesto que las letras no aparecen al pulsarlas. Pero si tienes un oponente honesto, tienes la opción de que las letras vayan apareciendo en la pantalla al pulsarlas. Si las letras pueden verse en la pantalla, se evita una escritura errónea, eliminando así las discusiones cuando aparece.

Entre cada palabra de una frase sólo debe haber un espacio. La longitud máxima de cualquier frase es de 64 caracteres en el Spectrum.

Una vez completada la frase, hay que pulsar la tecla de entrada y aparece la pantalla principal. En la parte superior de la misma hay la puntuación de cada jugador. Al principio del juego, cada jugador tiene 200 puntos, y el total puede aumentar o disminuir a medida que progresa el juego.

Debajo de las puntuaciones hay una tabla con los valores de las letras: las más corrientes tienen valores altos, y las menos corrientes, valores bajos. La frase misteriosa se presenta en forma de una fila de asteriscos.

En el fondo de la pantalla aparece un juego de instrucciones y un espacio para que introduzcas tus comandos y tus adivinanzas.

ESTRATEGIA

El jugador tiene tres opciones: comprar letras, adivinar una letra en una posición específica, o adivinar la frase.

En las primeras etapas del juego, una buena elección es comprar un espacio, aunque hay que asegurarse de que la frase contiene más de una palabra. La forma de proceder a partir de ahí es asunto tuyo. Las vocales son caras, pero tienen una probabilidad muy elevada de que sean válidas, mientras que las letras más baratas son arriesgadas debido a su rareza. A veces, las palabras son más fáciles de adivinar si se han encontrado algunas consonantes —un derroche de vocales no siempre es de ayuda.

A medida que la frase va tomando forma, probablemente verás que puedes adivinar una letra de una determinada posición. Por ejemplo, puedes tener una palabra como Q.E. Es evidente que la letra central es una U. De esta manera puedes sumar puntos. Una letra elegida correctamente sumará su valor a tu puntuación, mientras que una errónea sólo restará la mitad de su valor. Teclea XX para seleccionar la opción de adivinanza e introduce la tuya empleando un cursor una vez situado.

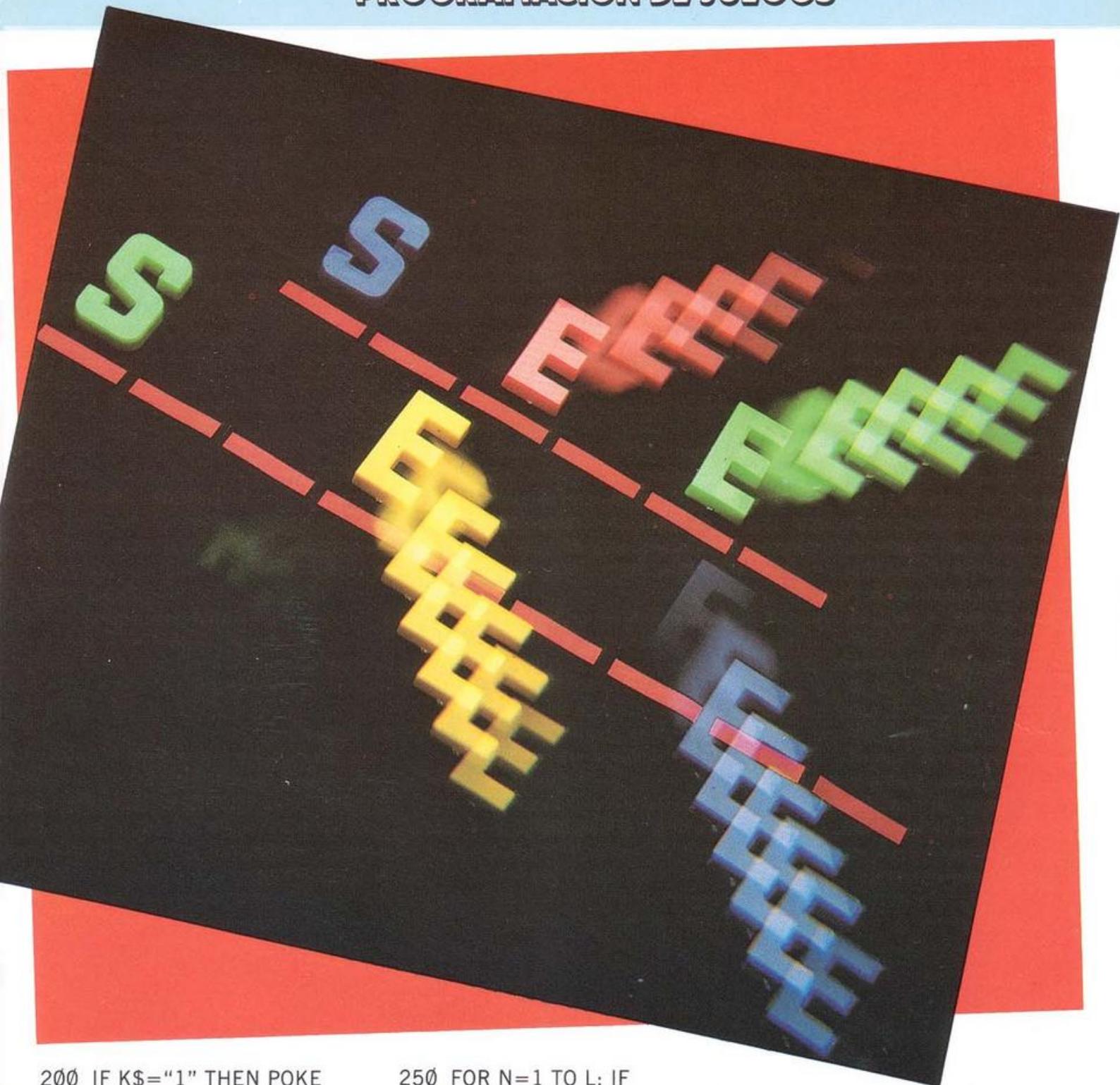
Cuando haya varias letras en su lugar, es posible que tengas un destello de inspiración y desees adivinar toda la frase. Para ello, teclea ZZ y la podrás introducir. Si la frase es la correcta, el valor de las letras que faltaban se sumará a la correspondiente puntuación. En cambio, si es errónea, se restarán 50 puntos. Un exceso de adivinanzas a tontas y a locas pronto dejará a cero tu puntuación.

Ahora introduce la primera sección del juego de las palabras. Estas líneas dejan la pantalla preparada para empezar el juego. Si haces correr el programa con RUN, verás que funciona, pero no podrás ir más allá, puesto que el resto del programa (que cubre las distintas elecciones) se termina en el siguiente capítulo.

No olvides salvar esta parte del programa con un buen SAVE.

```
10 LET R$="PALABRA": LET
W=14: LET D=0: LET F=1:
LET G$=" ": LET G=0: LET
K=0: LET Q$=" ": LET
TA=200: LET TB=200: LET
```

```
TC=0: LET B=0: POKE
23609,50: POKE 23658,8:
LET I$=" ": LET J$=" ": LET
Z$=" ": LET C$=" "
20 FOR N=0 TO 7: READ Y:
POKE USR "A"+N,Y: NEXT N
30 DATA 255,129,129,129,129,
129,129,255
40 INPUT "ENTRA NOMBRE DEL
PRIMER JUGADOR (HASTA 7
LETRAS)", LINE A$
50 INPUT "ENTRA NOMBRE DEL
SEGUNDO JUGADOR (HASTA
7 LETRAS)", LINE B$
60 IF LEN A$>7 OR LEN B$>7
THEN GO TO 40
70 CLS: INPUT "ENTRA
NUMERO DE PALABRAS DE
LA FRASE (1 A 9)", LINE C$
80 IF LEN C$<>1 THEN GO TO
70
90 IF CODE C$<49 OR CODE
C$>57 THEN GO TO 70
100 LET C=VAL C$
110 INPUT "ENTRA NUMERO
DE VUELTAS (1 A 9)", LINE
T$
120 IF LEN T$<>1 THEN GO
TO 110
130 IF CODE T$<49 OR CODE
T$>57 THEN GO TO 110
140 LET T=VAL T$
150 IF C>1 THEN LET
J$="S": LET I$="CON UN
SOLO ESPACIO ENTRE
CADA UNA. ": LET
R$="FRASE"
160 PRINT A$,"; ES TU
TURNO." "ENTRA TU
FRASE DE ";C;
"PALABRA";J$;" LAS
LETRAS SON INVISIBLES,
PERO SI QUIERES VER LAS
QUE ENTRES PRESIONA 0.
SI NO PULSA 1 PARA
CONTINUAR"
170 LET K$=INKEY$: IF K$=" "
THEN GO TO 170
190 IF K$="0" THEN POKE
23624,56: INPUT LINE S$:
CLS: POKE 23624,56: GO
TO 220
```



```

200 IF K$="1" THEN POKE
    23624,63: INPUT LINE S$:
    CLS: POKE 23624,56: GO
    TO 220
210 GO TO 170
220 LET L=LEN S$
230 IF L=0 THEN PRINT
    "ENTRADA ILEGAL": PAUSE
    100: CLS: GO TO 160
240 IF L>64 THEN PRINT
    "ENTRADA DEMASIADO
    LARGA": PAUSE 100: CLS:
    GO TO 160
    
```

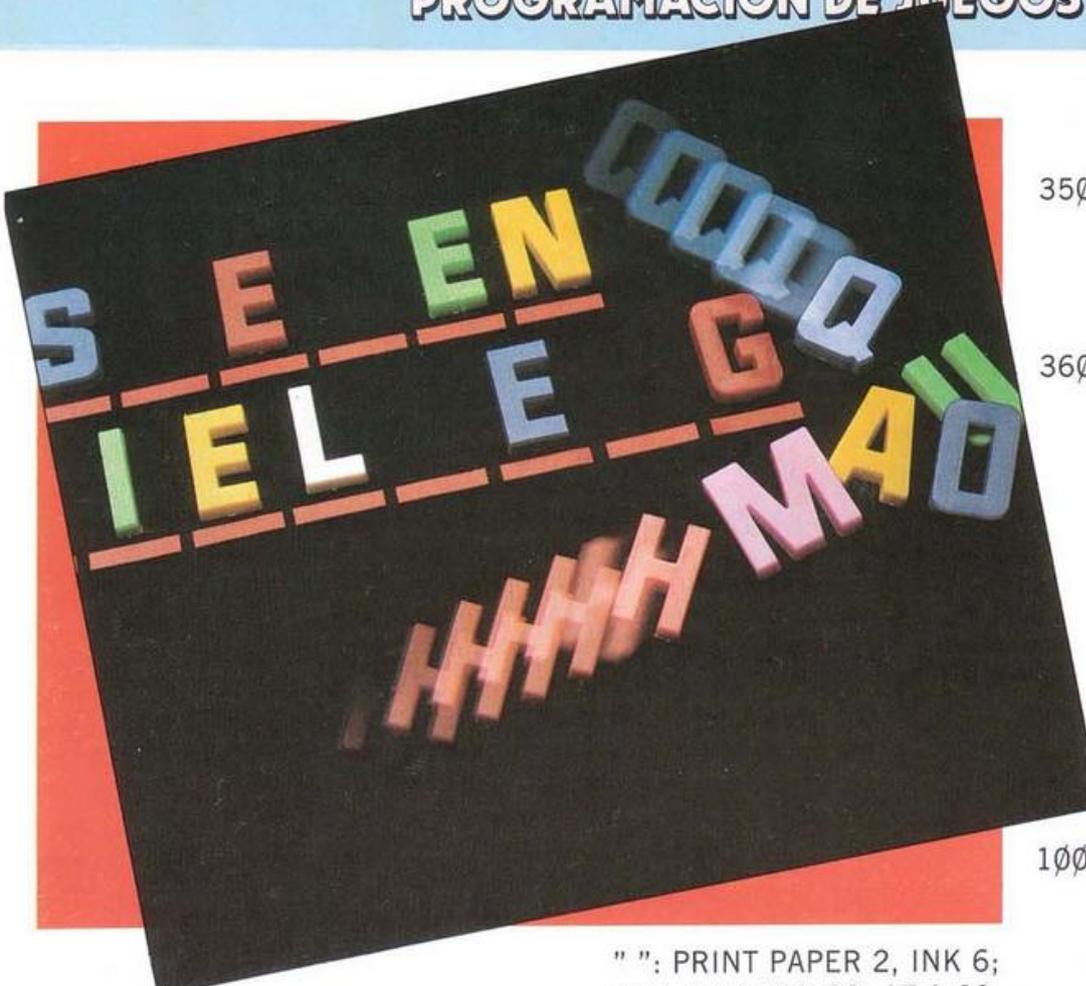
```

250 FOR N=1 TO L: IF
    S$(N)=CHR$ 32 THEN LET
    D=D+1: GO TO 270
260 IF CODE S$(N)<65 OR
    CODE S$(N)>90 THEN
    PRINT "HAS INTRODUCIDO
    UNA LETRA ILEGAL":
    PAUSE 100: CLS: LET D=
    0: GO TO 160
270 IF C=1 AND D=1 THEN
    PRINT "NO PUEDES TENER
    ESPACIOS EN UNA SOLA
    PALABRA": PAUSE 100:
    
```

```

    CLS: LET D=0: GO TO 160
280 NEXT N
290 IF D<>C-1 THEN PRINT
    "SE SUPONE QUE HAS
    INTRODUCIDO "; C;"
    PALABRAS ";I$;"CHEQUEA
    Y REINTRODUCE": PAUSE
    100: CLS: LET D=0: GO TO
    160
300 LET Z$=" "
    
```

PROGRAMACION DE JUEGOS



```

310 FOR N=1 TO L: LET
  Z$=Z$+"*": NEXT N
320 PRINT INK 1; AT 0,0;
  "PUNTOS ";A$;: PRINT INK
  1; AT 0,16;" PUNTOS ";B$;
  
```

```

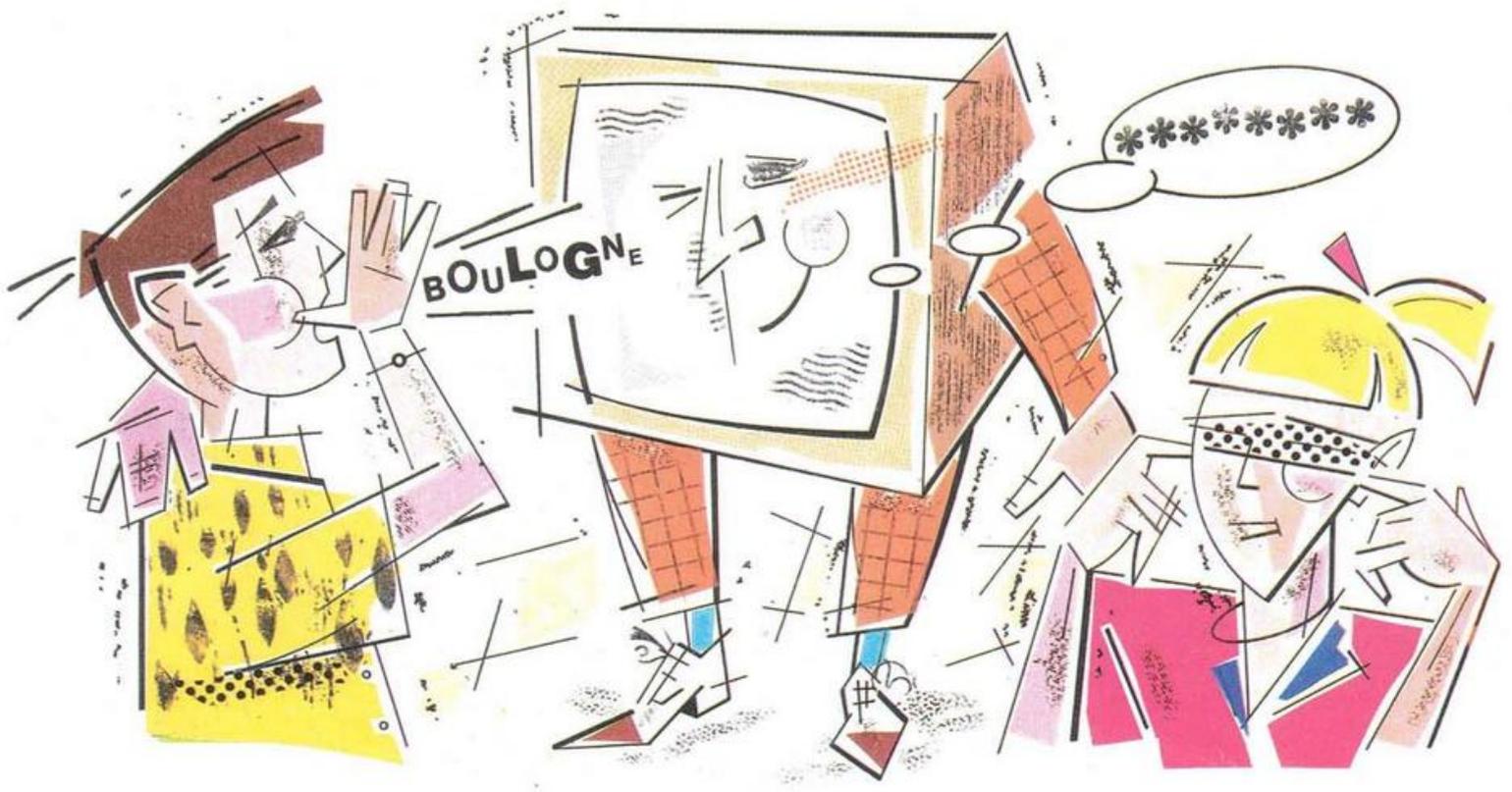
" ": PRINT PAPER 2, INK 6;
  AT 1,6;TA;TAB 28; AT 1,23;
  TB;TAB 31;" "
330 PRINT AT 3,3;"VALOR DE
  LOS CARACTERES"
340 FOR N=0 TO 26: READ G$:
  
```

```

LET Q$=Q$+G$: NEXT N:
  PRINT Q$: RESTORE 900
350 PRINT INK 1; AT 12,0;B$;"
  TIENES QUE ADIVINAR LA
  ";R$;" QUE CONTIENE ";L;
  " CARACTERES": PRINT
  PAPER 2; INK 6;Z$
360 INPUT "QUIERES COMPRAR
  UN CARACTER CON EL
  PRECIO DE LOS PUNTOS
  QUE SE MUESTRA?
  INTRODUCE EL CARACTER
  SELECCIONADO. SI NO
  PULSA XX PARA ADIVINAR
  UN CARACTER O ZZ PARA
  ADIVINAR TODA LA
  FRASE.", LINE D$
  
```

```

1000 DATA "A-20  ", "B-
  10  ", "C-10  ", "D-
  12  ", "E-20  ", "F-
  08  ", "G-12  ", "H-
  08  "
1010 DATA "I-20  ", "J-
  04  ", "K-06  ", "L-
  10  ", "M-10  ", "N-
  10  ", "O-20  ", "P-
  
```



PROGRAMACION DE JUEGOS

```

1Ø    ", "Q-Ø2    ", "R-
12    ", "S-12    "
1Ø2Ø DATA "T-12    ", "U-
2Ø    ", "V-Ø8    ", "W-
Ø8    ", "X-Ø4    ", "Y-
Ø8    ", "Z-Ø2    ",
"<GRAFICOS A>-2Ø "
    
```

El programa es muy sencillo, puesto que no hay gráficos que requieran el empleo de funciones especiales para visualizarlos.

La línea 1Ø inicializa todas las cadenas y las variables necesarias en el juego. Las líneas 2Ø y 3Ø establecen el espacio de gráficos definido por el usuario (UDG) empleado en la tabla de letras.

Las líneas 4Ø a 7Ø son las indicaciones del principio del juego. Las líneas 4Ø y 5Ø son los nombres de los jugadores, y la línea 6Ø comprueba que no sean demasiado largos para el espacio reservado en la pantalla. El número de palabras de la frase se introduce en la línea 7Ø.

La siguiente rutina, de las líneas 8Ø a la 1ØØ, es una serie de comprobaciones de validez, que aseguran que el número de palabras de la frase está dentro de los límites establecidos por el programa.

El número de palabras de la frase se llama C\$, y la línea 8Ø comprueba si la entrada sólo tiene un carácter de longitud. La línea 9Ø comprueba si la entrada es de 1 a 9, y entre las dos líneas se comprueba que la entrada esté entre el 1 y el 9 y sea un solo número. La línea 1ØØ convierte la cadena en una variable numérica.

Las líneas 11Ø a 14Ø corresponden al número de vueltas elegido. La línea 11Ø es la indicación y pide el número de vueltas, T\$. Las líneas 12Ø y 13Ø son comprobaciones de validez similares a las anteriores, mientras que la línea 14Ø convierte la cadena en una variable numérica.

La línea 15Ø comprueba que el número de palabras de la frase sea mayor que uno, e indica a los jugadores que debe haber un espacio entre las palabras. R\$ se hace igual a "FRASE", empleada más tarde en las indicaciones.

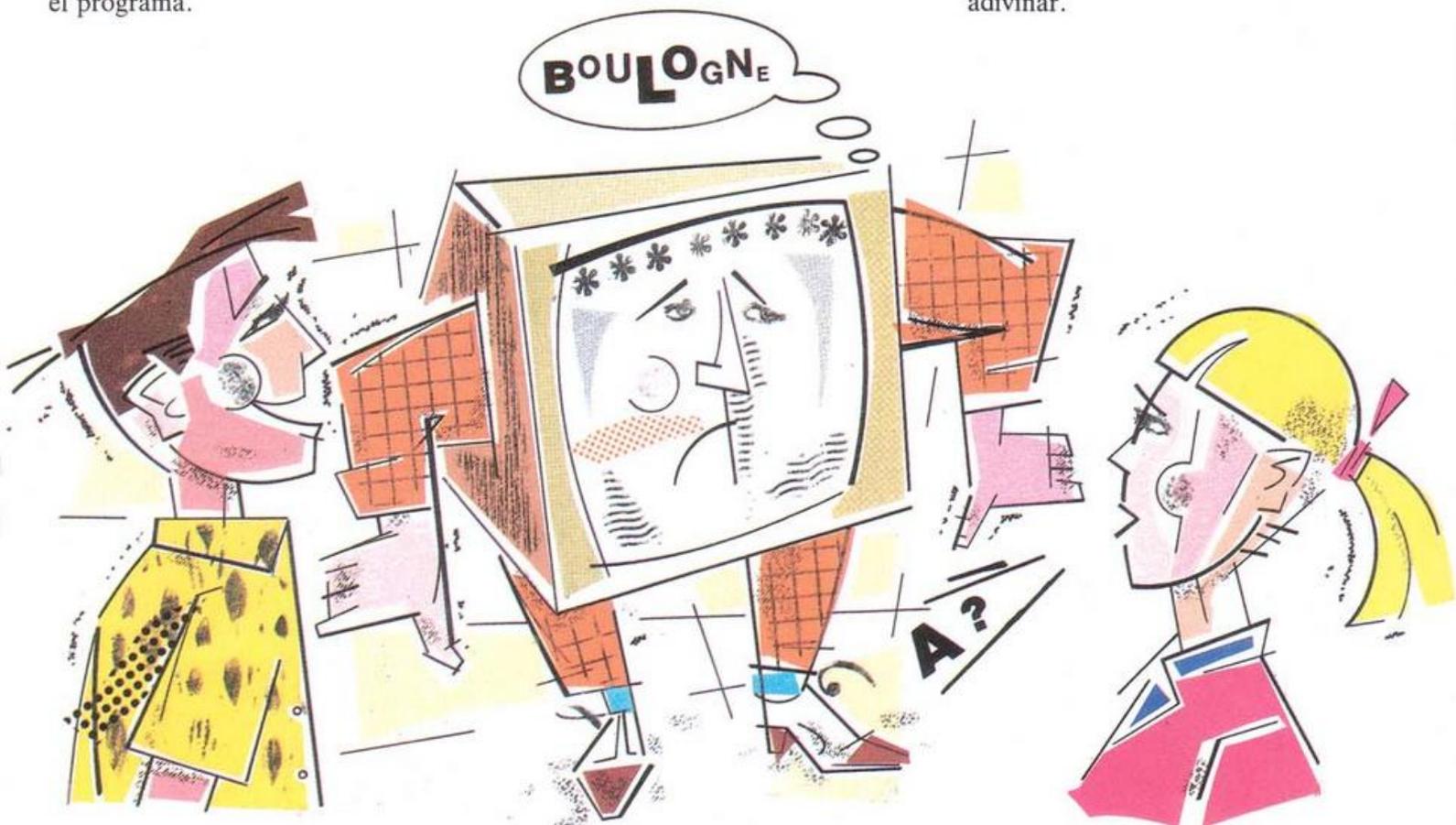
A continuación, el programa introduce la rutina de entrada, que se extiende de la línea 16Ø a la 22Ø, y da instrucciones al jugador que tiene el turno para teclear la frase misteriosa.

La selección de un Ø hará que la frase aparezca en la pantalla tal y como fue introducida; en caso contrario, ésta permanece invisible. La frase se llama S\$.

Las líneas 23Ø a 29Ø son comprobaciones de validez. Si la longitud de la frase es menor que un carácter, o sea si el jugador sólo ha pulsado ENTER, la línea 23Ø anuncia que se ha producido una entrada ilegal. La línea 24Ø, si la entrada tiene la longitud correcta, y la línea 25Ø comprueba el número de espacios (que debe ser inferior en uno al número de palabras dado por C). Las líneas 26Ø y 27Ø comprueban, junto con la 29Ø, si hay entradas ilegales.

Las líneas 30Ø y 31Ø establecen Z\$, una cadena ficticia que consiste en una cantidad de asteriscos igual a la longitud de S\$.

La rutina final, entre las líneas 32Ø a la 36Ø, establece el resto de la pantalla principal. Lee (READ) una tabla de valores de letras de la DATA del final del programa, imprimiendo las dos puntuaciones de los jugadores y presentando la cadena ficticia de los asteriscos. La línea 36Ø es una indicación para el jugador que trata de adivinar.



ULTIMANDO EL JUEGO DE LAS PALABRAS

Saca quinario, xileno y quincuagésimo de los polvorientos rincones de tu diccionario. Utiliza tácticas "sucias" para sacar de quicio a tus amigos con el juego de las palabras

En esta segunda parte final sobre el juego de las palabras de INPUT hay todo lo que necesitas para empezar a jugar. Teclea el resto de líneas para ver algunas interesantes aplicaciones del manejo de cadenas con tu máquina. Después trata de desconcertar a tus amigos con oscuras palabras y frases.

En el programa hay unas rutinas para cada una de las tres opciones: comprar letras, adivinar una letra específica de una posición específica y adivinar la frase completa. El programa también sigue las puntuaciones, el número de intentos y el número de vueltas.

Las líneas 370 a 410 manejan la entrada del jugador que adivina, bien la compra de letras, bien la adivinanza. D\$ es la decisión del jugador, y la rutina comprueba la entrada para ver si se trata de una letra o de un espacio

únicos, para indicar una compra, o si es XX o ZZ, para una adivinanza. La rutina también atrapa cualquier entrada no válida, enviando el programa a la línea de indicación de entrada, la 360.

COMPRA DE LETRAS

Si un jugador decide comprar una letra, lo primero que hace el ordenador es comprobar su valor. Para ello, el programa va a la subrutina que empieza en la línea 790.

Ésta comprueba el valor ASCII de la letra y después realiza un cálculo con este valor para determinar el punto de la tabla de letras en que está. Si el ordenador encuentra un blanco, significa que la letra ya ha sido vendida, por lo que el programa vuelve a la línea 360 para que el jugador vuelva a hacer una selección. En caso contrario, retira la parte de la cadena que contiene el valor, que fija utilizando VAL. Ésta se utilizará para actualizar la puntuación del jugador.

La rutina que viene después de la compra empieza en la línea 430. Las líneas 430 a 460 examinan la frase, buscando la existencia en ella de la palabra comprada. La cadena ficticia se actualiza, sustituyendo los asteriscos por la letra en cada posición que aparece en la frase, para visualizar a continuación la nueva cadena. Si la letra está en la frase, su valor se suma a la puntuación del jugador y, si no está, se resta la mitad de su valor. También se incrementa el total de adivinanzas, F. La línea 440 sustituye la letra elegida y su valor por blancos para indicar que esta opción ya no está disponible.

LETRAS ESPECIFICAS

Si el jugador que adivina desea probar una letra en una posición específica, deberá seleccionar XX. Esto hace que el programa salte a la línea 500 y, a continuación, el jugador deberá introducir la letra elegida. Hay varias trampas de error para asegurar que la entrada sea legal. Si se hace una



PROGRAMACION DE JUEGOS

■ JUEGO DE LAS PALABRAS
COMPLETO CON NUEVAS RUTINAS

■ COMPRA DE LETRAS

■ COMPROBACION DE LA ENTRADA
DEL JUGADOR

■ COLOCACION DE UNA LETRA
ESPECIFICA EN POSICION

■ ACERTAR LA FRASE
COMPLETA

■ ¡TRAMPA!

elección legal, el jugador recibe instrucciones sobre el posicionado del cursor para introducir la letra. Con el cursor situado en la posición adecuada, hay que pulsar la tecla Ø. El programa se asegura de que la elección está disponible y comprueba si ésta es correcta.

Si la letra es errónea, la línea 69Ø dice MALA SUERTE al jugador y resta la mitad del valor de la puntuación del jugador como multa. Si la letra y la posición son correctas, la línea 70Ø dice BUEN ACIERTO al jugador y suma el valor de la letra a su puntuación. Si la frase ha quedado completa, la línea 71Ø envía el programa a la línea 73Ø, que presenta ENHORABUENA al jugador.

LA FRASE ENTERA

Si el jugador que adivina es más ambicioso y ha elegido la opción de adivinar toda la frase (ZZ), la línea 385 envía el programa a la línea 90Ø. La rutina pide al jugador que introduzca la frase, la cual es comparada con la que se introdujo inicialmente. Si la frase es errónea, se restan 50 puntos

de la puntuación del jugador y se incrementa la cuenta de adivinanzas. Si es correcta, se calcula la puntuación con las letras no utilizadas, y el programa vuelve a la línea 73Ø, que presenta ENHORABUENA.

EL FINAL

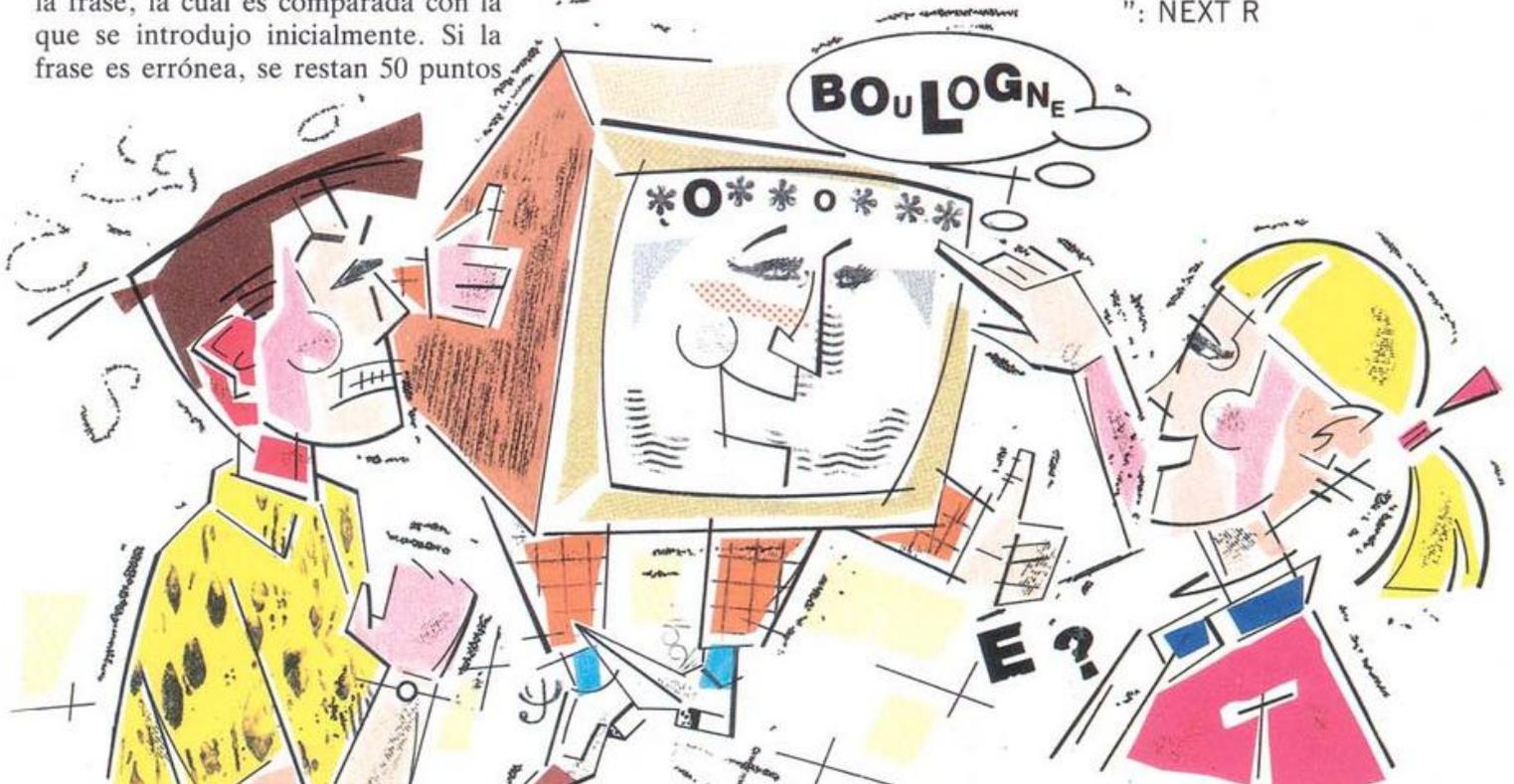
Una vez adivinada correctamente la frase, el programa comprueba si se ha utilizado el número de tiradas de cada jugador que se ha elegido al principio del programa. Si no ha sido así, el turno pasa al siguiente jugador, una vez puestas a cero todas las cadenas y las variables utilizadas en el juego.

Si el juego ha terminado, el programa salta a la línea 88Ø. En ella y en las dos líneas siguientes se comparan las puntuaciones, y se visualiza el resultado del juego.

El programa termina aquí, pero si lo deseas, puedes añadir una rutina "¿otro juego?" para hacer que el programa y el juego queden realmente completos.

```

37Ø IF D$<>"XX" AND
D$<>"ZZ" AND LEN D$>2
THEN GO TO 36Ø
38Ø IF D$=CHR$ 32 THEN GO
TO 41Ø
385 IF D$="ZZ" THEN LET
D$=" ": GO TO 90Ø
39Ø IF CODE D$<65 OR CODE
D$>9Ø THEN GO TO 36Ø
40Ø IF D$="XX" THEN LET
D$=" ": GO TO 50Ø
41Ø GO SUB 79Ø
42Ø LET E=Ø
43Ø LET E=E+1
44Ø IF E=L+1 THEN LET
TB=TB-G: LET Q$(M TO
M+7)=" ": PRINT AT 4,Ø;
Q$: LET D$=" ": GO TO
47Ø
45Ø IF S$(E)<>D$ THEN GO TO
43Ø
46Ø IF S$(E)=D$ THEN LET
Z$(E)=D$: GO TO 43Ø
47Ø PAUSE 1ØØ: PRINT AT 14,
Ø,: FOR R=1 TO 7: PRINT "
": NEXT R
    
```



PROGRAMACION DE JUEGOS

```

480 PRINT PAPER 2; INK 6; AT
1,22;TB;CHR$ 32: PRINT
PAPER 2; INK 6; AT 14,0;
Z$: PRINT "ADIVINA "; F:
LET F=F+1: IF S$=Z$
THEN GO TO 730
490 GO TO 360
500 INPUT "QUE CARACTER
QUIERES ADIVINAR?", LINE
D$
510 IF LEN D$>1 THEN GO TO
500
520 IF D$=CHR$ 32 THEN GO
SUB 790: GO TO 550
530 IF CODE D$<65 OR CODE
D$>90 THEN GO TO 500
540 GO SUB 790
550 PRINT PAPER 2; INK 6; AT
18,0;D$: PRINT AT 18,2;
"EN QUE POSICION? USA
LAS TECLAS DEL CURSOR Y
PRESIONA 0 PARA
ENTRAR"
560 PRINT PAPER 2; INK 6;
AT 14,0;Z$: PRINT PAPER
6; INK 2; AT 14,B;
Z$(B+1)
570 PAUSE 0: LET Y$=INKEY$:
IF Y$=" " THEN GO TO 570
590 IF Y$="8" AND B<L-1
THEN LET B=B+1
600 IF Y$="5" AND B>0 THEN
LET B=B-1
610 IF Y$="0" THEN GO TO
680
640 IF B>=32 THEN
LET W=15: LET
V=B-32
650 IF B<32 THEN LET W=14:
LET V=B
660 PRINT PAPER 2; INK 6; AT
14,0;Z$: PRINT PAPER 6;
INK 2; AT W,V;Z$(B+1)
670 GO TO 570
680 IF Z$(B+1)<>"*" THEN GO
TO 570
690 IF S$(B+1)<>D$ THEN
LET TB=TB-G/2: PRINT
FLASH 1; AT 17,0; "MALA
SUERTE": PAUSE 50: LET
B=0: GO TO 470
700 IF S$(B+1)=D$ THEN

```

```

PRINT FLASH 1; AT 17,0;
"BUEN ACIERTO": PAUSE
50: LET Z$(B+1)=D$: LET
TB=TB+G: LET B=0
710 IF S$=Z$ THEN GO TO
730
720 GO TO 470
730 PRINT INK 6; PAPER 2; AT
1,22;TB: PRINT AT 17,0;
"ENHORABUENA, ";B$;TAB
0;TAB 31;" ": PAUSE 100:
CLS
740 LET K=K+1: IF K=T*2
THEN GO TO 880
750 LET C$=A$: LET A$=B$:
LET B$=C$
760 LET TC=TA: LET TA=TB:
LET TB=TC
770 LET Q$=" ": LET D=0: LET
F=1
780 GO TO 160
790 LET M=(CODE D$-64)*
8-7
800 IF M=-263 THEN LET
M=209
810 IF Q$(M TO M+5)=" "
THEN GO TO 360
820 LET G=VAL Q$(M+2 TO
M+3)
830 RETURN

```

```

880 IF TA>TB THEN CLS: PRINT
A$;" HAS GANADO POR ";
TA;" PUNTOS A ";TB
890 IF TB>TA THEN CLS: PRINT
B$;" HAS GANADO POR ";
TB;" PUNTOS A ";TA
892 IF TA=TB THEN CLS: PRINT
"EL RESULTADO ES UN
EMPATE"
895 STOP
900 INPUT "ENTRA LA FRASE" ,
LINE H$
910 IF H$<>S$ THEN PRINT
FLASH 1; AT 17,0;
"ERRONEA!": PAUSE 50:
LET TB=TB-50: PRINT INK
6; PAPER 2; AT 1,22;TB:
PRINT AT 17,0;
"ACERTASTE ";F:
LET F=F+1: GO TO
360
920 FOR N=1 TO L: LET
D$=Z$(N): IF D$<>"*"
THEN GO TO 950
930 LET M=(CODE S$(N)-64)*8-
7: IF M=-263 THEN LET
M=209
940 LET TB=TB+G
950 NEXT N: GO TO
730

```



Puedes ganar 25.000 pts. o tu juego preferido

Encuesta para conocer tu perfil y preferencias

Hace ahora un año, la redacción de INPUT se inundó de encuestas enviadas por vosotros, los lectores. Gracias a esa entusiasta acogida fue posible conocer a fondo vuestras inquietudes y ajustar a ellas los contenidos de la revista.

Ahora, esta nueva encuesta pretende saber hasta qué punto hemos conseguido ese objetivo, y enterarnos de por dónde van ahora vuestras exigencias, siempre pensando en que INPUT siga siendo una herramienta indispensable para vuestros primeros escarceos por el mundo de la informática.

1. ¿Tienes ordenador? Sí No
2. En caso afirmativo. ¿Cuál?
Marca Modelo
3. ¿Cuánto tiempo hace que lo posees?
4. ¿Qué periféricos y accesorios tienes?
5. ¿Desde qué número lees INPUT?
6. ¿Con qué frecuencia la adquieres?
TODAS DE VEZ EN CUANDO OTROS
7. ¿Cuántas personas leen tu ejemplar de INPUT?
Sólo yo Yo y mis amigos
Mi hermano, mi padre y yo Mi hermano y yo
8. ¿Lees toda la revista? Sí No
9. ¿Tecleas los programas que publicamos?
Todo Casi todo
Alguna cosa Nada
10. Califica qué secciones te gustan más con mayor puntuación (10 a 0)
Programación
Código Máquina
Robótica
Coleccionable
Revista Software
Aplicaciones
Actualidad
Libros
Buzón
Educación
Inteligencia Artificial
11. ¿Qué secciones o temas echas de menos en INPUT?
12. ¿Crees que alguna sección debe ampliarse? ¿Cuál?
13. ¿Cuál debería reducir su extensión?
14. ¿Tienes en cuenta los comentarios sobre juegos (Revta. de Software) a la hora de adquirirlas en el mercado?
Al 100 % En gran medida Poco Nada
15. ¿De todos los artículos que has leído en INPUT, nos podrías destacar algunos?
N.º 1
N.º 2
N.º 3
N.º 4
N.º 5
N.º 6
16. SEXO:
M F
17. Edad:
Hasta 12 años De 21 a 25 años
De 12 a 14 años De 26 a 35 años
De 15 a 17 años De 36 a 45 años
De 18 a 20 años Más de 45 años
18. ¿Cuál es tu situación?
Estudio enseñanza básica
Estudio enseñanza media
Estudio enseñanza Universitaria
Estoy parado
Trabajo (en este último caso responde a la siguiente pregunta)
19. ¿Cuál es tu profesión?
20. ¿Con qué objetivo adquiriste tu ordenador?
Jugar Educativo Música
Utilidades Profesionales Dibujo
21. ¿Cuál es tu nivel de conocimientos de informática?
0 Bastante Aficionado con exp.
Profesional
22. Aparte de la informática, ¿qué otras aficiones tienes?
Deporte Cine Música
Video Pintura Teatro
Motociclismo Fotografía
23. ¿Estás satisfecho con tu ordenador?
24. En caso de que te tuvieses que comprar otro, ¿por cuál te decidirías?
25. ¿Lees otras revistas de informática? Sí No
26. ¿Cuáles?

Entre los que contestéis a esta encuesta, se sorteará un premio en metálico de 25.000 pts. y diez programas a elegir por los agraciados. Así, que escribe tus datos para participar. La fecha límite es el próximo 5 de marzo. No es preciso que contestes las preguntas para las que no tengas respuesta. Tampoco es necesario que las respuestas sean favorables a nuestra labor para que puedas participar en el sorteo. Critícanos si honestamente crees que debes hacerlo. Envíanos este cuestionario o su fotocopia a: ENCUESTA INPUT SINCLAIR, Aribau, 185, 1.º - 08021 Barcelona

Nombre Apellidos
Edad Dirección Dto. Postal
Teléfono de contacto Gracias y suerte con el premio.

LISP, EL LENGUAJE DE LAS LISTAS (I)

Aunque empezó con lentitud, el LISP ha ido ganando poco a poco velocidad y las «máquinas de LISP» pueden muy bien ser los prototipos de los ordenadores domésticos que se vendan en los próximos diez a veinte años.

El LISP fue desarrollado originalmente por John MacCarthy en el Instituto Tecnológico de Massachusetts a principios de la década de los 60, una época en que los lenguajes de ordenador más sofisticados eran las versiones primitivas del FORTRAN, el lenguaje parecido al BASIC que utilizan ingenieros y científicos. Su nombre se deriva de List Processing, procesado de listas, lo que nos puede dar una pista sobre la forma en que trabaja. Hoy día, veinticinco años más tarde, los programas en LISP continúan basándose en los mismos principios fundamentales, mientras que el FORTRAN ha debido ir incorporando nuevas capacidades a medida que la gente ha ido aprendiendo gradualmente a escribir mejor sus programas.

Incluso aunque tú no hayas oído hablar nunca antes del LISP, es muy probable que los programas que escribes se hayan visto influidos por él, debido a que muchos de sus conceptos han sido aplicados a otros lenguajes. No obstante, a pesar de su influencia sobre otros lenguajes, las ideas del LISP siguen siendo radicalmente diferentes de las de lenguajes como el BASIC o el PASCAL, siendo además el LISP un lenguaje considerablemente más potente que ellos.

LA IMPORTANCIA DE LA VELOCIDAD

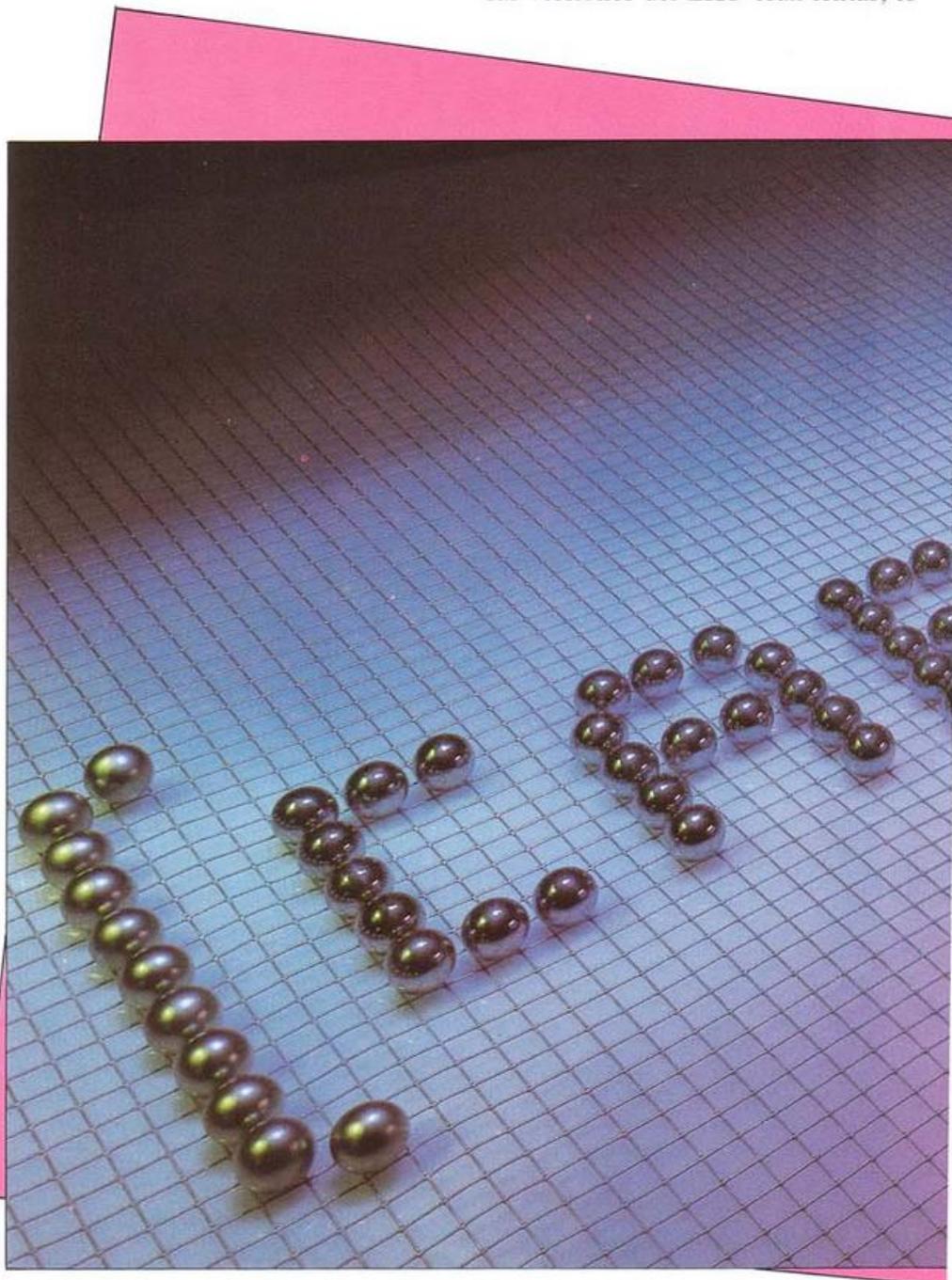
Es bien conocido que resulta más fácil escribir un programa en BASIC para realizar una determinada tarea, que hacerlo en código máquina. El BASIC es un lenguaje de más alto ni-

vel y sus recursos actúan sobre él como una especie de palanca intelectual haciendo que el programador pueda abordar problemas más difíciles.

En el futuro los ordenadores realizarán tareas mucho más complejas de

lo que lo hacen actualmente, lo que significa que los programadores tendrán que utilizar lenguajes aún más potentes, como el LISP, para escribir sus programas; pero los lenguajes más potentes tienden a ser más lentos.

Realmente es cierto que las primeras versiones del LISP eran lentas, lo



■ LA IMPORTANCIA DE LA VELOCIDAD
■ INTELIGENCIA ARTIFICIAL
■ OTRAS APLICACIONES
■ EL LISP Y EL ORDENADOR
■ LOS FUNDAMENTOS

■ CARGA DEL LISP
■ VALORES DE LOS ÁTOMOS
■ LOS NÚMEROS EN LISP
■ VERDADERO O FALSO
■ SEGUIMIENTO

cual le quitó popularidad. Sin embargo, las actuales versiones de LISP que corren sobre grandes ordenadores, son tan rápidas como cualquier otro lenguaje de alto nivel. Además el producto de la revolución tecnológica de los circuitos integrados es un enorme crecimiento de la velocidad y capacidad de procesamiento de un solo chip de silicio. Esto significa que ya no resulta vital apurar hasta el úl-

timo bit la velocidad del sistema utilizando el código máquina.

INTELIGENCIA ARTIFICIAL

El LISP es el principal lenguaje utilizado en Inteligencia Artificial. Su objetivo es aquí la creación de programas de ordenador que presenten un comportamiento similar en algún aspecto a la inteligencia humana. Un

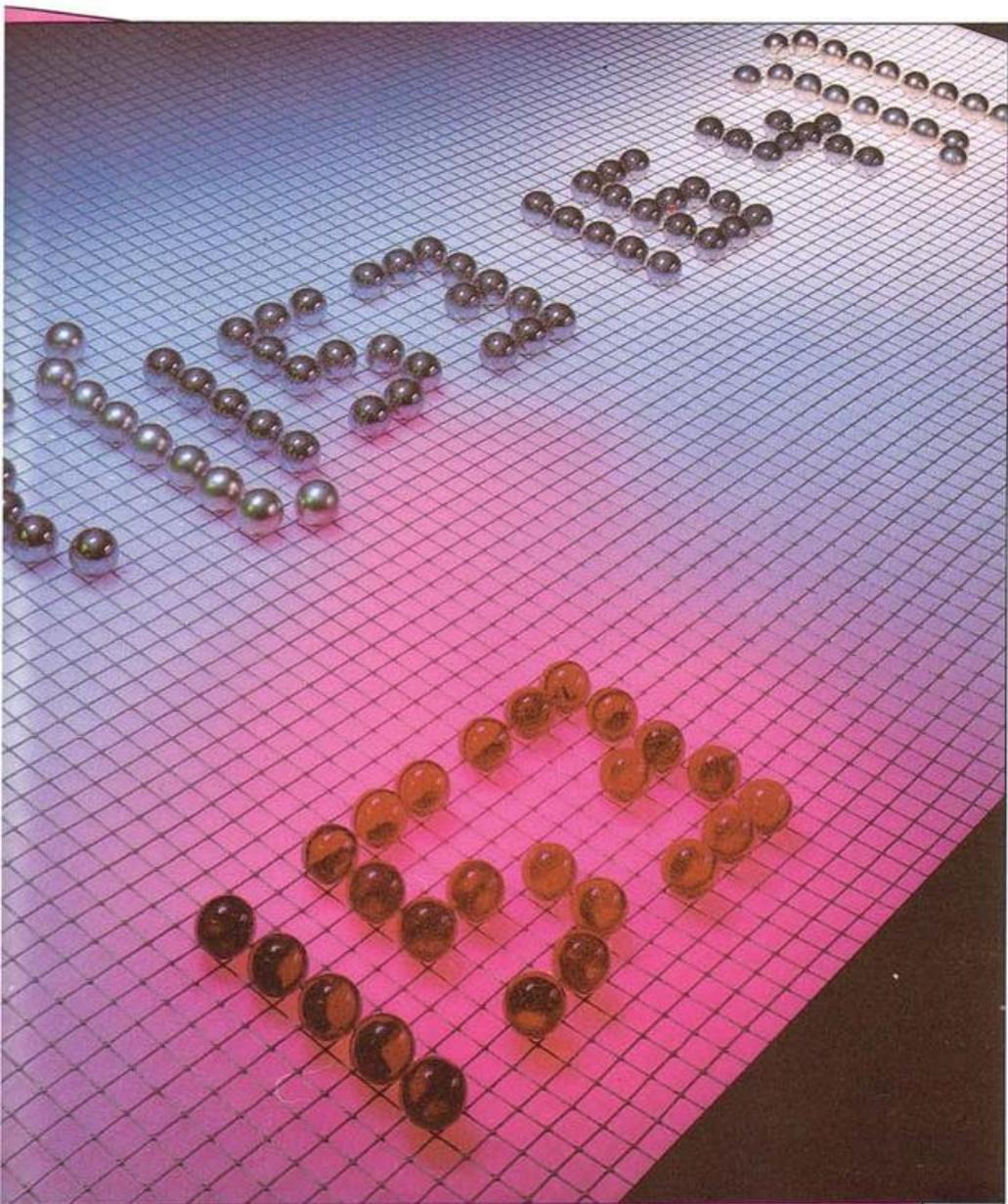
ejemplo típico es la máquina que juega al ajedrez. Otro ejemplo es uno de los más famosos en los que se utilizó el LISP. Se trata de un programa llamado ELIZA que desarrolló el papel de psicoanalista, comunicándose con sus pacientes a través de un terminal de ordenador. El programa simulaba la conversación que habría podido tener lugar, preguntando y respondiendo a las preguntas. Hubo personas que realmente se quedaron confundidas pensando que el ordenador entendía de verdad lo que ellas le decían. De hecho lo que hacía el ordenador era simplemente mirar a su entrada y replicar modificándola adecuadamente. El análisis necesario para hacer esto es mucho más fácil en LISP que en otros lenguajes.

Otra área de la inteligencia artificial son los Sistemas Expertos. En este caso se trata de programar en un ordenador todos los conocimientos de un experto humano en algún campo, tal como el mantenimiento de los motores de automóvil. Un usuario sin experiencia puede averiguar lo que le pasa a su coche tecleando los síntomas de los fallos. El ordenador le replica con su diagnóstico, haciendo nuevas preguntas que pueden ser necesarias para localizar con más precisión el problema.

OTRAS APLICACIONES

Otros programas más tradicionales, como compiladores y procesadores de textos, son mucho más sencillos cuando se escriben en LISP, habiéndose utilizado también este lenguaje para escribir los programas de algunos de los sistemas de diseño asistido por ordenador (CAD) utilizados en el diseño de circuitos integrados.

El LISP ha desempeñado un importante papel en los programas de ordenador que se ocupan del álgebra. Es



evidente que los ordenadores son muy buenos en aritmética. Sin embargo, con el LISP es posible hacer que lo sean también en álgebra. Esto significa que los ordenadores pueden hacer la clase de cálculos algebraicos que mantendría ocupado a un ejército de matemáticos y llenaría kilómetros de papel.

Algunos de los cálculos de la física teórica contienen un álgebra tan complicada que sólo puede hacerse mediante estos sistemas basados en el LISP. A un nivel más modesto, una aplicación típica del LISP que podrías hacer correr en tu ordenador doméstico podría ser el cálculo de derivadas, lo cual será muy útil si tienes que hacer docenas y docenas de ejemplos de cálculo en tus tareas de casa.

EL LISP Y EL ORDENADOR

Sería erróneo dar la impresión de que el LISP es el único lenguaje capaz de resolver problemas difíciles o de alcanzar relevancia en el futuro. Han aparecido varios lenguajes modernos que utilizan ampliamente sus conceptos, entre los que se incluyen el LOGO y el PROLOG. Con mucha frecuencia los compiladores e intérpretes de estos lenguajes fueron escritos en LISP. Por eso el LISP es en cierta medida el lenguaje ensamblador de la nueva generación de lenguajes de ordenador. Actualmente existen de hecho procesadores especiales cuyos lenguajes ensambladores están basados en el LISP, básicamente de la misma forma que los procesadores Z80, 6502 o 6809 de un ordenador doméstico son capaces de entender el código máquina adecuado.

Estos procesadores de LISP constituyen muchas veces el corazón de determinados ordenadores personales de alta potencia o de puestos de desarrollo de ingeniería cuyo principal lenguaje es el LISP. Estas «Máquinas de LISP» pueden muy bien ser los prototipos de los ordenadores domésticos que estén a la venta en la calle dentro de cinco o diez años.

En estos artículos sobre LISP, explicaremos los fundamentos del lenguaje y podrás seguir sus desarrollos

incluso aunque todavía no tengas un LISP. Como en el caso del BASIC, existen diferencias entre las versiones del LISP de ordenadores diferentes, aunque afortunadamente las variaciones de uno a otro suelen ser muy pequeñas; normalmente suelen consistir únicamente en diferencias ortográficas en los nombres de las funciones. Un inconveniente del LISP es que hace uso de un elevado número de paréntesis. Si eres una de esas personas que encuentran grandes dificultades para hacer cuadrar el número de paréntesis en las expresiones del BASIC, puede que el LISP no sea el lenguaje más adecuado para ti.

FUNDAMENTOS

Uno de los hechos que distingue al LISP del BASIC es que el LISP, además de números, es capaz de manipular símbolos. Es algo parecido a la forma en que el BASIC puede fraccionar y manejar cadenas de caracteres, si bien la capacidad de manejo de símbolos del LISP es mucho mayor.

Los objetos básicos que maneja el LISP se llaman átomos y listas. Un átomo es realmente como una variable en BASIC, empieza por una letra y puede continuar por cualquier longitud de letras y números (aunque en la práctica siempre hay un límite). Según esto, podrían ser átomos los siguientes: LUIS, átomo, NUM1, NUM2. Todos estamos familiarizados con muchos tipos de listas, por ejemplo una lista de cosas que comprar o la lista de los jugadores de algún equipo deportivo. Tu lista de compras podría ser: té, azúcar y leche. Una lista de LISP es algo muy parecido que incluye cierto número de átomos y probablemente otras listas. Para informar al LISP de que algo es una lista, se encierra ésta entre paréntesis. Así, tu lista de compras en LISP podría ser algo como:

(te azúcar leche)

Una lista de LISP puede contener tanto átomos como otras listas, por lo que otro ejemplo sería:

((te frio) (leche caliente) azúcar)

La anterior lista contiene dos listas y un átomo. Los paréntesis se utilizan de una manera muy parecida a las dobles comillas que usa el BASIC para mostrar dónde empiezan y terminan las cadenas de caracteres. La diferencia consiste en que las cadenas de caracteres no pueden contener otras cadenas de caracteres.

CARGA DEL LISP

Si quieres hacer tus propias pruebas con el LISP, no tienes necesidad de cambiar tu ordenador doméstico tan hábil con el BASIC por una máquina de LISP de seis millones de pesetas. De hecho la mayoría de los ordenadores domésticos se pueden convertir al LISP. La única razón por la que un ordenador entiende el BASIC es porque existe un gran programa en ROM que es un intérprete de BASIC. Es perfectamente posible sustituir este código máquina por otro programa que interprete o entienda los programas en LISP. En la práctica, el añadirle el LISP a algunos ordenadores consiste en enchufarles un chip o un pequeño módulo que contiene una ROM que lleva dentro el código máquina necesario. Se suele adoptar con más frecuencia la solución de mantener en la memoria principal del ordenador la sección de código máquina que contiene el intérprete LISP. En tales casos se carga el LISP a partir de una cinta o disco, igual que cualquier otro programa y una vez que el código está dentro de tu ordenador éste se comporta ya como un sistema LISP.

¿Qué es lo que se vería cuando esto sucede? En vez del acostumbrado ok del BASIC, te encontrarías con el mensaje Evaluate:. Esto significa que el LISP está esperando a que teclees alguna expresión válida en LISP, la cual evaluará a continuación e imprimirá su valor en la pantalla. Así es todo el LISP, tú tecleas expresiones y el ordenador te contesta con sus valores. En LISP se llama expresión a una expresión de tipo simbólico que contenga una lista o un átomo. Con frecuencia suele llamarse a esto una s-

expresión. No existe un comando equivalente al RUN que ponga realmente en marcha el programa. Como verás más adelante, este comando no hace falta.

Cuando te aparece el mensaje Evaluate: y el ordenador se queda esperando, tienes que decidir lo que vas a teclear. Lo más sencillo es un átomo, como LUIS. El LISP te responde entonces con un UNDEFINED. El diálogo sería algo parecido a lo que sigue:

```
Evaluate: LUIS
```

```
Value is: UNDEFINED
```

Esto significa que el LISP nunca ha visto anteriormente ese átomo y que no puede encontrar su valor. Al igual que las variables del BASIC, los átomos pueden tener valores. La situación se parece a hacer un PRINT de una variable en BASIC antes de asignarle un valor. Cuando el LISP se encuentra el átomo, intenta evaluarlo, es decir, encontrar su valor. Esta es una importante característica del LISP: todo se evalúa siempre a menos de que exista la intención expresa de no hacerlo.

Hay un método para eliminar las evaluaciones en LISP, la comilla sencilla '. Supongamos que tecleas 'LUIS. En estas condiciones, la respuesta del LISP será ahora LUIS. Con la comilla se detiene cualquier evaluación ulterior de lo que viene detrás. El valor de la expresión LUIS es precisamente el átomo que sigue a la coma. También puede usarse la coma con listas. Así:

```
'(te azucar leche)
```

se evaluará como (té azúcar leche).

Una aplicación mucho más interesante que la evaluación de átomos es la evaluación de listas. Nuevamente tenemos un ejemplo de esto en la comilla anterior: 'LUIS es realmente una abreviatura de (QUOTE LUIS). De hecho algunos sistemas sólo aceptarán la segunda forma de la expresión. QUOTE es aquí una función del LISP, mientras que el átomo LUIS es su argumento. Observa que QUOTE no halla el valor de su argumento.

Compara esta función con las funciones del BASIC, que podrían ser el tipo:

```
RND (20)
```

El BASIC encuentra para este valor un número aleatorio. Esto mismo en LISP tendría el siguiente aspecto:

```
(RND 20)
```

Observa la nueva situación de los paréntesis. Análogamente, si en BASIC existiera una función llamada QUOTE, se escribiría así:

```
QUOTE (LUIS)
```

Puede que te confunda al principio este nuevo uso de los paréntesis, pero una vez que se ha entendido es sencillo.

VALOR DE LOS ATOMOS

Al igual que las variables, los átomos también pueden tener un valor. Sin embargo surge la cuestión de cómo adquiere un átomo su valor. Por ejemplo, ¿cómo se le asigna a LUIS el valor 1? Se hace con la función SET de la siguiente forma:

```
(SET'LUIS 1)
```

Como de costumbre, el LISP lo evalúa todo. El primer argumento de SET se evalúa como LUIS, y el segundo se evalúa como 1; a continuación la función asigna el valor 1 a LUIS. Esto nos muestra dos cosas. En primer lugar, que en LISP todos los números son átomos que se evalúan para dar su valor numérico. Y en segundo lugar que, igual que ocurre en BASIC, las funciones pueden tener más de un argumento. Si ahora tecleas LUIS, el LISP te respondería con 1. La interacción con el LISP podría ser algo así:

```
Evaluate: (SET'LUIS 1)
```

```
Value is: 1
```

```
Evaluate: LUIS
```

```
Value is: 1
```

Como todas las demás funciones, ha

de resultar un valor para SET, normalmente el valor de su segundo argumento que en este caso es 1. Existe otra versión del SET que es SETQ, en la cual no se evalúa el primer argumento. Así, podrías teclear:

```
(SETQ LUIS 1)
```

sin colocar la comilla delante de LUIS. Para insistir aún más en el tema, supongamos que tecleas seguidamente:

```
(SETQ JORGE LUIS)
```

En este caso, el LISP evaluaría LUIS y encontraría que su valor es 1, y a continuación pondría el valor de JORGE a 1. JORGE no se evalúa y no tiene que ser protegido por medio de '.

La función SET se parece al comando LET que tienen algunas versiones del BASIC. Con frecuencia en BASIC las variables tienen que ser de un tipo declarado. En LISP no hay esta necesidad de especificar las variables: un átomo puede tener como valor un número, otro átomo o una lista. Por ejemplo:

```
(SETQ COMPRAS ' (te leche azucar))
```

asignará tu lista de compras al átomo COMPRAS.

Llegados aquí, puede que te hayas dado cuenta de una importante diferencia entre el LISP y el BASIC. Considera en primer lugar la lista:

```
(te leche azucar)
```

es decir, tu lista de compras. Sin embargo, si el té fuera una función (como SET) esta lista podría ser una expresión en LISP y podría ser evaluada. Más adelante veremos que es fácil definir átomos como té para que sean funciones. La cuestión es que en LISP los programas y los datos tienen exactamente el mismo aspecto. Este hecho tiene gran alcance; significa que un programa en LISP puede manejar otro programa en LISP con la misma facilidad con que maneja otros datos.

En LISP todo se hace por medio de

funciones y encontrando el valor de expresiones. Pero a veces esto no resulta muy conveniente. Por ejemplo, normalmente no estarás interesado en el valor real que resulte de la función SET, es mucho más importante que asigne un valor a un átomo. Por eso algunas funciones sólo se utilizan por sus efectos laterales y no por su valor.

DESMENUZANDO LAS PARTES

Para hacer algo con el LISP, evidentemente tienes que poder manipular listas; es algo análogo al manejo de cadenas de caracteres en el BASIC, que las tres funciones fundamentales que hacían esto son CAR, CDR y CONS. La reacción natural ante esta información es «So what?» (¿Qué?). Suponte, sin embargo, que te dijeran que estas funciones se llaman FIRST, REST y ADD to FRONT. Estas denominaciones te dan realmente mucha más información sobre lo que realmente hacen. De hecho CAR y CDR se han llamado así por ser las iniciales de «contenido del registro de dirección» (contents of address register) y «contenido del registro de decremento» (contents of decrement register), y proceden de los nombres de dos de los registros del ordenador IBM en el que se utilizó inicialmente el LISP. CONS significa construcción. Casi todo el mundo aprende a utilizar estos términos, pero podrían cambiarse fácilmente por ejemplo por (SETQ FIRST CAR).

Tal como sugiere el nombre propuesto como alternativa, CAR da como valor resultante el primer elemento de la lista. Por ejemplo:

```
(CAR '(te leche azucar))
```

da como valor resultante té. Observa que CAR evalúa sus argumentos, por lo que hay que utilizar una ' para evitar la evaluación de la lista. Para ver de otra forma cómo funciona esto, tecléa:

```
(SETQ COMPRAS '(te leche azucar))
```

y a continuación:

```
(CAR COMPRAS).
```

El LISP te respondería de nuevo con te como valor de esto, ya que se encuentra que el valor de COMPRAS es la lista:

```
(te leche azucar)
```

Con CDR resulta la lista que contiene todos los elementos de la lista dada, tomada como argumento, excepto el primero. Así el valor de:

```
(CDR '(te leche azucar))
```

es la lista:

```
(leche azucar)
```

Tal como sugiere su nombre, CONS sirve para construir listas. Hace esto de una manera muy específica, tiene dos argumentos, un átomo o una lista y una lista. CONS agrega su primer argumento al principio de la lista que figura como segundo argumento. Por ejemplo:

```
(CONS 'galletas '(te leche azucar))
```

se evalúa como

```
(galletas te leche azucar)
```

Para listas, el funcionamiento es el siguiente:

```
(CONS '(izquierdo derecho) '(mano pie))
```

que tiene el valor:

```
((izquierdo derecho) mano pie)
```

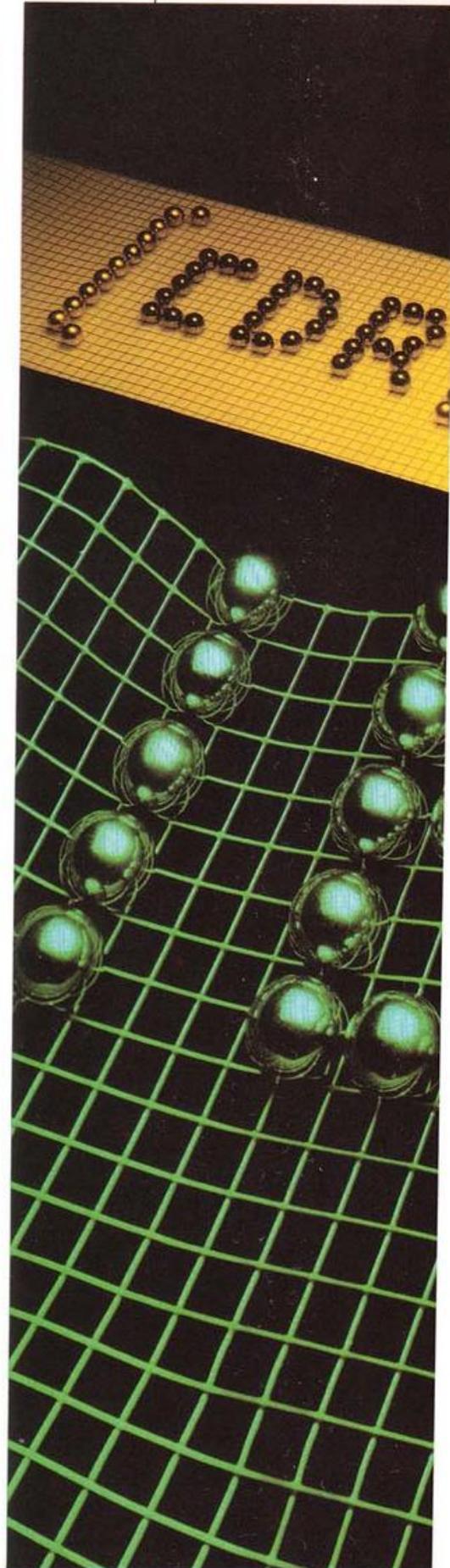
Con CAR y CDR se obtiene en este caso:

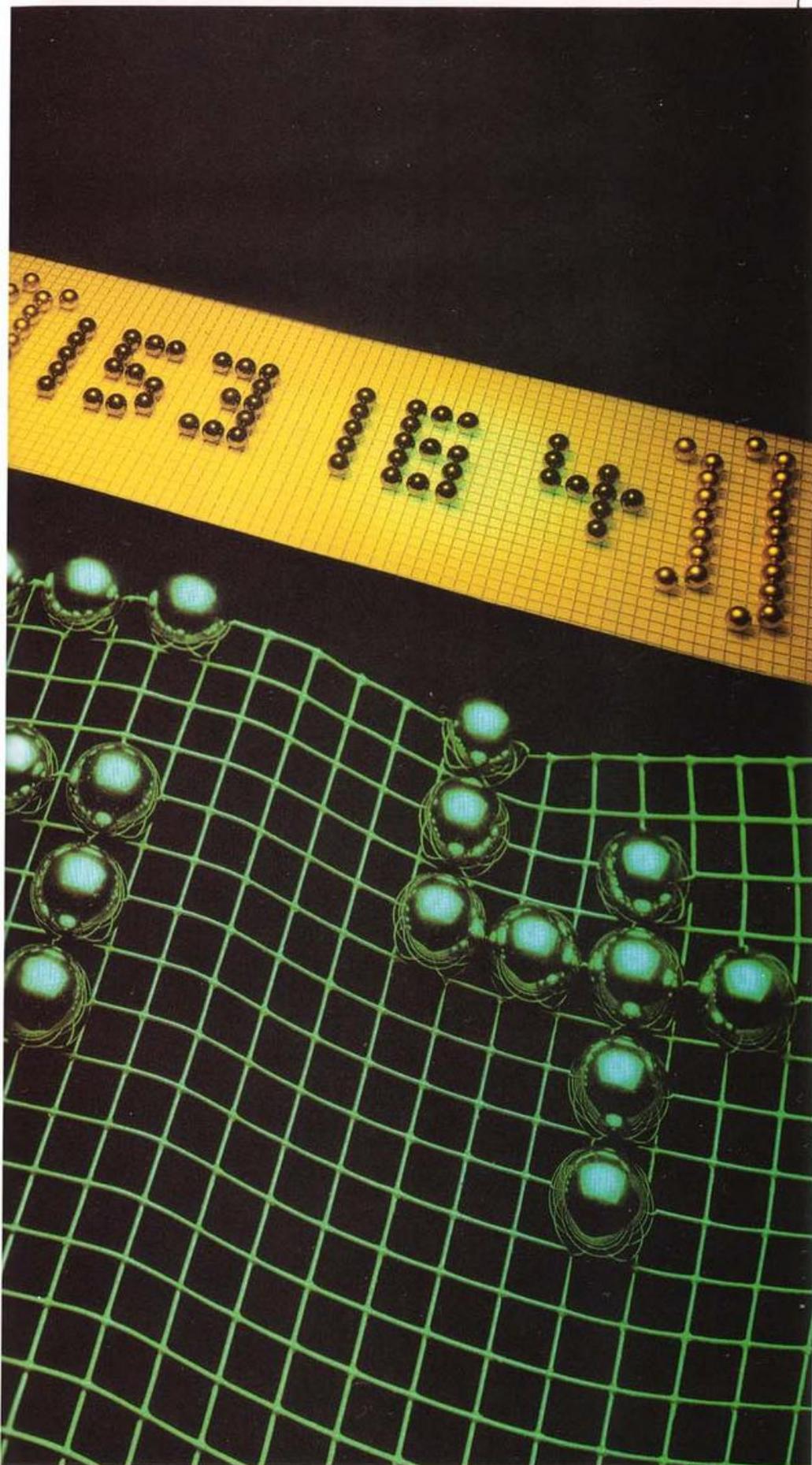
```
(izquierdo derecho)
```

y:

```
(mano pie).
```

Observa que en este caso con CAR se obtiene una lista y no un átomo. Supongamos que tomas el CDR de una lista con sólo un elemento, por ejemplo:





(CDR ' (LUIS))

Esto tiene todo el aspecto de una lista vacía (). De hecho en LISP existe un nombre especial para esto, el átomo NIL. Al tomar el CAR o el CDR de un átomo o de una lista vacía se produce un mensaje de error, tal como era de esperar. Existe una generalización de CAR y CDR. Imagínate que tratas de encontrar el átomo izquierdo de la lista:

((izquierdo derecho) mano pie)

Podrías encontrarlo como valor de la expresión

(CAR (CAR ' ((izquierdo derecho) mano pie)))

El LISP te permite hacerlo de una forma más corta con la función CAAR, pudiendo escribirse:

(CAAR ' ((izquierdo derecho) mano pie))

Análogamente, CADR significa tomar el CDR y a continuación el CAR. Los sistemas de LISP permiten también todas las demás combinaciones de CAR y CDR tomados en pares y a veces en ternas, cuaternas y así sucesivamente.

Si tuvieras dos átomos LUIS y JUAN y quisieras ponerlos en una lista, podrías sentirse inclinado a teclear:

(CONS 'LUIS 'JUAN)

Aunque se trata de una expresión válida, no es una lista. Su valor es el siguiente par separado por un punto:

(LUIS. JUAN),

pero de momento esto es una complicación innecesaria. Una manera de formar la lista deseada sería teclear lo siguiente:

(CONS 'LUIS (CONS ' JUAN NIL))

El segundo argumento del primer CONS se evalúa para dar la lista (SMITH) en cuyo principio se añade el argumento FRED. Sin embargo

LISP cuenta también con el método más breve siguiente:

```
(LIST 'LUIS 'JUAN)
```

Este es un ejemplo de otras de las características del LISP: las funciones con un número arbitrario de argumentos. Podrías teclear por ejemplo:

```
(LIST 'UNO 'DOS 'TRES)
```

para obtener la respuesta:

```
(UNO DOS TRES)
```

Esto es muy diferente de lo que ocurre en BASIC, donde las funciones siempre tienen que tener el mismo número de argumentos.

LOS NUMEROS EN LISP

Aunque una de las principales características del LISP es su capacidad de manejar símbolos, también es capaz de manejar números. Las implementaciones del LISP en ordenadores domésticos habitualmente utilizan solamente números enteros. Sin embargo algunos de los grandes sistemas de LISP pueden manejar números en punto flotante y hacer cálculos con ellos con tanta rapidez como cualquier otro lenguaje especial para el cálculo numérico, tal como el FORTRAN. Una peculiaridad que poseen con frecuencia estos sistemas es la aritmética de precisión arbitraria en la que se puede tomar cualquier número de cifras significativas, lo cual resulta perfecto para calcular el valor del número pi con unos cuantos millones de números decimales.

En comparación con el BASIC, el LISP maneja la aritmética de una forma un tanto extraña. Se parece más bien al LOGO, del que ya nos hemos ocupado anteriormente. Los equivalentes de los operadores habituales +, -, *, / y MOD (que da el resto de la división entera) son las funciones PLUS DIFFERENCE, TIMES, QUOTIENT y REMAINDER. La gran diferencia consiste en que el LISP utiliza notación de prefijo, en contraste con el BASIC, que utiliza notación interna. La notación con pre-

fijo se llama a veces notación polaca, en memoria del lógico polaco Lukasiewicz, que demostró que esta notación (así como la polaca inversa, consistente en poner al final el signo de la operación a realizar) son especialmente apropiadas para la aritmética.

Cuando en BASIC quieres sumar 2 y 2, escribes 2 + 2. En LISP esto se convierte en (PLUS 2 2). En la notación con prefijo los operadores aritméticos figuran en primer lugar en las expresiones. Esto permite al LISP tratar a los operadores aritméticos en las funciones exactamente igual que a los demás. Por eso puedes cambiar sus nombres si lo deseas. Algunas de las funciones aritméticas del LISP, como el PLUS, pueden tomar cualquier número de argumentos. Por ejemplo:

```
(PLUS 2 2 2 (TIMES 1 2))
```

se evalúa el resultado:

$$2+2+2+1 * 2 = 8$$

En LISP la función MINUS se utiliza para cambiar el signo de un número; por ejemplo el valor de (MINUS 2) es -2. Esto contrasta con el BASIC, donde el símbolo - se emplea tanto para el cambio de signo como para encontrar la diferencia entre dos números.

VERDADERO O FALSO

La siguiente cosa en que tenemos que fijarnos es la forma que tiene el LISP de manejar las expresiones lógicas y cómo hace los saltos condicionales dependiendo del valor de una expresión lógica. En BASIC los valores lógicos se tratan exactamente igual que la aritmética entera; típicamente se utiliza 0 para falso y -1 para verdadero, aunque son posibles otras elecciones. Una expresión lógica es algo del tipo:

A > B

Si A es mayor que B, entonces esta expresión en BASIC tiene valor verdadero, en caso contrario tiene valor falso. En LISP verdadero y falso se re-

presentan por dos átomos especiales T y NIL (el mismo NIL descrito anteriormente como la lista vacía).

Los valores verdadero y falso resultan de funciones especiales a las que con frecuencia se les llama predicados. Un ejemplo de ello es la función ATOM, que da como resultado el valor T si su argumento es un átomo y NIL en caso contrario. Tecleando:

```
(ATOM 'LUIS)
```

el LISP responderá con el resultado T, mientras que

```
(ATOM '(leche galletas))
```

dará como resultado NIL. Otros predicados de LISP son GREATERP, para ver cuál de dos átomos tiene mayor valor numérico y NULL que examina una lista para ver si está vacía. El predicado EQ ensaya la igualdad de dos átomos cualesquiera:

```
(EQ 'LUIS 'COME ) (EQ 'LUIS 'LUIS)
```

da como resultado T, mientras que

```
(EQ 'LUIS 'ENRIQUE)
```

tiene el resultado NIL. En realidad EQ ensaya si sus dos argumentos tienen el mismo valor, así con:

```
(SETQ F '(mermelada pasteles))
```

se tendrá que:

```
(SETQ G F)
```

seguido de:

```
(EQ GF)
```

Tendrá el valor T puesto que los valores resultantes de G y F son la misma lista. Además de éstos, hay muchos otros predicados en LISP.

Además de los operadores aritméticos, el LISP cuenta también con el usual complemento de los operadores lógicos. Estos también utilizan la notación de prefijo. Por ejemplo, (OR T T) tiene el valor T, (OR T NIL) tiene el valor T, (AND T NIL) tiene el valor NIL, etc.

En BASIC lo primero que suele ha-

cerse una vez que se está en condiciones de evaluar una expresión lógica es realizar saltos condicionales dependiendo de los valores de dichas expresiones, lo cual se hace con la sentencia IF. El equivalente de esto en LISP se hace por medio de la función COND. Esta función tiene un número arbitrario de argumentos (llamados cláusulas), cada uno de los cuales es una lista que contiene un cierto número de s-expresiones (listas de átomos). Cuando se evalúa la función COND se examinan todas las cláusulas por turno evaluando la primera s-expresión. Si su valor resulta ser T, se examinan todas las s-expresiones subsecuentes de esa cláusula. El valor de la función COND se toma entonces como el de la última s-expresión de la cláusula. En caso contrario, la evaluación continúa en la siguiente cláusula. Si ninguna de las cláusulas tiene una primera s-expresión que sea T, la función COND tiene entonces el valor NIL. Normalmente se organizan las cosas de modo que ningún valor NIL para una primera s-expresión dé el mismo resultado que un valor T.

Un caso típico de COND sería el siguiente:

```
(COND ((EQ LUIS 1)(PRINT'UNO))
      ((EQ LUIS 2)(PRINT'DOS))
      ((EQ LUIS 3)(PRINT'TRES))
```

En este ejemplo se ha introducido otra nueva función, PRINT; su efecto es imprimir el resultado después de evaluarlo. Aunque el LISP cuenta con un conjunto completo de estas funciones de impresión, suelen diferir de un sistema a otro.

Si ahora LUIS ha sido puesto a 2 con SET y se impone la anterior COND, imprimirá la palabra DOS en la pantalla. De hecho el valor de la función PRINT es usualmente el valor de su argumento, por lo que el LISP devolverá el valor DOS como valor de la COND. Está claro por qué sucede esto. En primer lugar el LISP compara el átomo LUIS con 1 y se encuentra con que no son iguales; esto significa que la primera s-expresión de la primera cláusula es NIL. Entonces el LISP se desplaza a la segunda cláusula

de la COND. Esta vez la primera s-expresión, tiene el valor T, por lo que se evalúa la segunda s-expresión (PRINT 'DOS) con el resultado de imprimir DOS en la pantalla y devolver el valor de la s-expresión (PRINT 'DOS), es decir, DOS.

Todo esto puede parecer bastante complicado y además hay que estar muy atento a los paréntesis en estas expresiones. Sin embargo hay una sentencia muy parecida en el BASIC que hará más claro el funcionamiento de COND.

Examina el siguiente fragmento de programa en BASIC:

```
IF LUIS = 1 then print «UNO»
ELSE IF LUIS = 2 THEN PRINT
«DOS»
ELSE IF LUIS = 3 THEN PRINT
«TRES»
```

Esta cadena de IF THEN ELSE es equivalente a la función COND. En BASIC estas cadenas terminan muchas veces con un simple ELSE. Esto significa que si no se satisface ninguno de los IFs el último ELSE dominará sobre todos los otros casos. Para tener este comportamiento en LISP, la función COND se termina con la cláusula:

(T (código en LISP para otros casos))

Si COND se encuentra con que no se satisface ninguno de sus otros argumentos, llega a esta cláusula y evalúa T que tiene el valor T. Así, las expresiones que siguen a T siempre son evaluadas.

SEGUIMIENTO

Puede que te resulte difícil recordar todo lo que sigue. El LISP te facilita las cosas por medio de una lista de objetos. Dicha lista contiene todos los átomos que el LISP sabe que tienen un valor. La función OBLIST o algo parecido tiene como valor la lista de objetos. Así, para ver todos los objetos conocidos por el LISP, teclea:

(OBLIST)

Fíjate que aunque esta función no tiene parámetros, también tiene que ir entre paréntesis. En la lista de objetos encontrarás todos los nombres de las funciones familiares como CAR y CDR. También, si haces algo como lo siguiente:

(SETQ LUIS 1)

te encontrarás el átomo LUIS añadido a la lista.

El hecho clave que permite escribir programas en LISP es, como en el LOGO, que el usuario puede definir sus propias funciones. Estas figurarán entonces en la lista de objetos y, en lo que al LISP respecta, son tan importantes como las que lleva incorporadas el propio LISP. En consecuencia puedes ampliar el LISP para que incluya cualquier cualidad peculiar que quisieras que tuviera; en la segunda parte de esta serie desvelaremos los misterios de cómo usar y definir funciones en LISP.

A NUESTROS NUEVOS LECTORES

En las páginas centrales de la revista encontrarás la sección «Programación de juegos» que se compone de una serie de artículos coleccionables que continúan mes tras mes.

Como la paginación de estos artículos es siempre correlativa con la del mes anterior apreciarás que no se corresponde con la del resto de la revista, pudiendo parecer a los más despistados que faltan páginas o que se trata de un error de encuadernación.

APRENDE MECANOGRAFIA POR ORDENADOR

Tanto si te dispones a sentarte para empezar a escribir tu primera novela, como si lo único que pretendes es tardar menos en teclear tus programas, el primer paso esencial es la capacidad de escribir a máquina con rapidez y precisión.

La quinta generación de ordenadores nos promete una comunicación mucho más directa entre máquina y usuario, considerando como corrientes cosas tales como el control de la máquina por la voz. Algunos de estos nuevos sistemas de control ya se han abierto camino en el campo de las máquinas industriales y comerciales de bajo costo.

Sin embargo, para los entusiastas del micro doméstico, parece que por desgracia aún tendrá que pasar algún tiempo hasta que estas novedades estén disponibles, por lo que de momento seguimos estando encadenados al teclado.

En realidad el teclado es un método razonablemente eficaz de comunicarle instrucciones e información al ordenador; su principal desventaja consiste en que su uso requiere mucho tiempo, en especial si no eres un mecanógrafo experto (y muy preciso). Y no hay cosa más frustrante que intentar copiar una lista impresa teniendo que estar constantemente cambiando la vista desde el teclado a la pantalla y de ésta al papel, para ir comprobando que lo que vas tecleando es correcto.

En consecuencia, ¿por qué no utilizar tu ordenador para que mejore tu capacidad de escribir a máquina? Se trata de una habilidad que no sólo resulta útil para escribir programas, sino que también resultará de gran valor en cuanto desees hacer uso de toda la potencia de tu ordenador como procesador de textos. El programa que si-

te ayudará a familiarizarte con el teclado y, si no haces trampas, a aumentar tu velocidad y tu precisión.

En realidad los poseedores de un Spectrum estáis en desventaja con respecto a otros ordenadores, debido a que el teclado del Spectrum es por su propia naturaleza menos adaptado para la mecanografía rápida. Pero naturalmente, para teclear programas sólo necesitas una pulsación de tecla para cada comando del BASIC.

EL PROGRAMA

Exactamente igual que se hace en los cursos de mecanografía que pueden llegar a costar una pequeña fortuna, el curso de mecanografía de INPUT está diseñado para ir aumentando metódi-



- ¿POR QUE APRENDER A ESCRIBIR A MAQUINA?
- APRENDIZAJE DE LAS TECLAS DE REFERENCIA
- COMO UTILIZAR EL PROGRAMA

- MEJORANDO LA VELOCIDAD Y LA PRECISION
- CAMBIO DE LOS DATOS
- PRACTICANDO CON TODO EL ALFABETO

APRENDIZAJE DE LAS TECLAS DE REFERENCIA

La habilidad en mecanografía consiste en pulsar en cada momento la tecla adecuada sin tener que estar constantemente mirando al teclado. Es más fácil de lo que parece, pero sólo se puede llegar a ello empezando de la forma correcta y conservando los buenos hábitos una vez que hayas llegado a adquirirlos.

Evidentemente, si tienes que llegar a la tecla correcta sólo por el tacto,

madas teclas de referencia (a las teclas a y ñ se les llama "teclas guía"). Son las teclas de la fila central, que empieza por ASDF... Por eso son éstas las primeras teclas que aprenderás. Una vez que domines las teclas de referencia, puedes llegar a las restantes partes del teclado moviendo tus dedos sólo una corta distancia cada vez.

Para obtener el máximo provecho del curso, tienes que intentar atenerte a él y no hacer trampas. Pero incluso si eres un mecanógrafo de "dos de-

camente tu destreza, de forma que vayas obteniendo resultados en cada etapa a medida que vas progresando. Pero no se trata de ningún proceso mágico y, al igual que en el aprendizaje de las demás cosas, requiere su tiempo.

Por ello hemos fraccionado el curso en una serie de etapas fáciles, añadiendo una parte de programa en cada una de las etapas. De esta forma, cuando estés listo para avanzar un paso más, puedes incorporar la parte de programa correspondiente al paso al siguiente nivel de eficiencia. Y a medida que vayas avanzando, irás actualizando constantemente tu velocidad y tu precisión.



NIVEL 3 . . .

tendrás que aprender dónde está cada tecla. Para ello lo más importante es que te asegures de que tus manos ocupan siempre la misma posición sobre el teclado, ya que en caso contrario no tendrías forma de saber dónde tienes cada dedo para presionar o no las distintas teclas. Los mecanógrafos profesionales consiguen esto, asegurándose de que sus dedos siempre están colocados sobre las mismas teclas, lla-

dos" podrás ver cómo mejora tu habilidad.

Cuando hayas tecleado el programa, tarea que te irá resultando más fácil a medida que vayas avanzando en el curso, y lo ejecutes, siéntate bien, con las manos correctamente colocadas sobre el teclado, listo para empezar, y sigue las indicaciones de pantalla que explicamos más adelante. El dedo meñique de tu mano izquierda debe estar situado sobre la tecla A, el siguiente dedo sobre la S, y así sucesivamente, con el dedo índice colocado sobre la F. El dedo índice de tu mano derecha debe estar sobre la tecla J, el dedo medio sobre la K, etcétera.

Esta disposición deja dos teclas sin cubrir en la parte central: la G y la H. Puedes llegar a la G con tu dedo índice izquierdo y a la H con tu dedo índice derecho. Si todo esto te parece complicado, no te preocupes. La primera parte del programa está diseñada para que te familiarices más con ello.

FUNCIONAMIENTO DEL PROGRAMA

Al ejecutar el programa, la pantalla te pedirá que selecciones una opción entre las cinco disponibles. Debes empezar por la lección 1 y dominarla antes de pasar a la lección 2, y lo mismo para las restantes lecciones.

molesto zumbido cuando te comas alguna letra o teclees la que no es. Al final del ejercicio, aparecerá en pantalla el tiempo que has tardado y el número de faltas que has cometido.

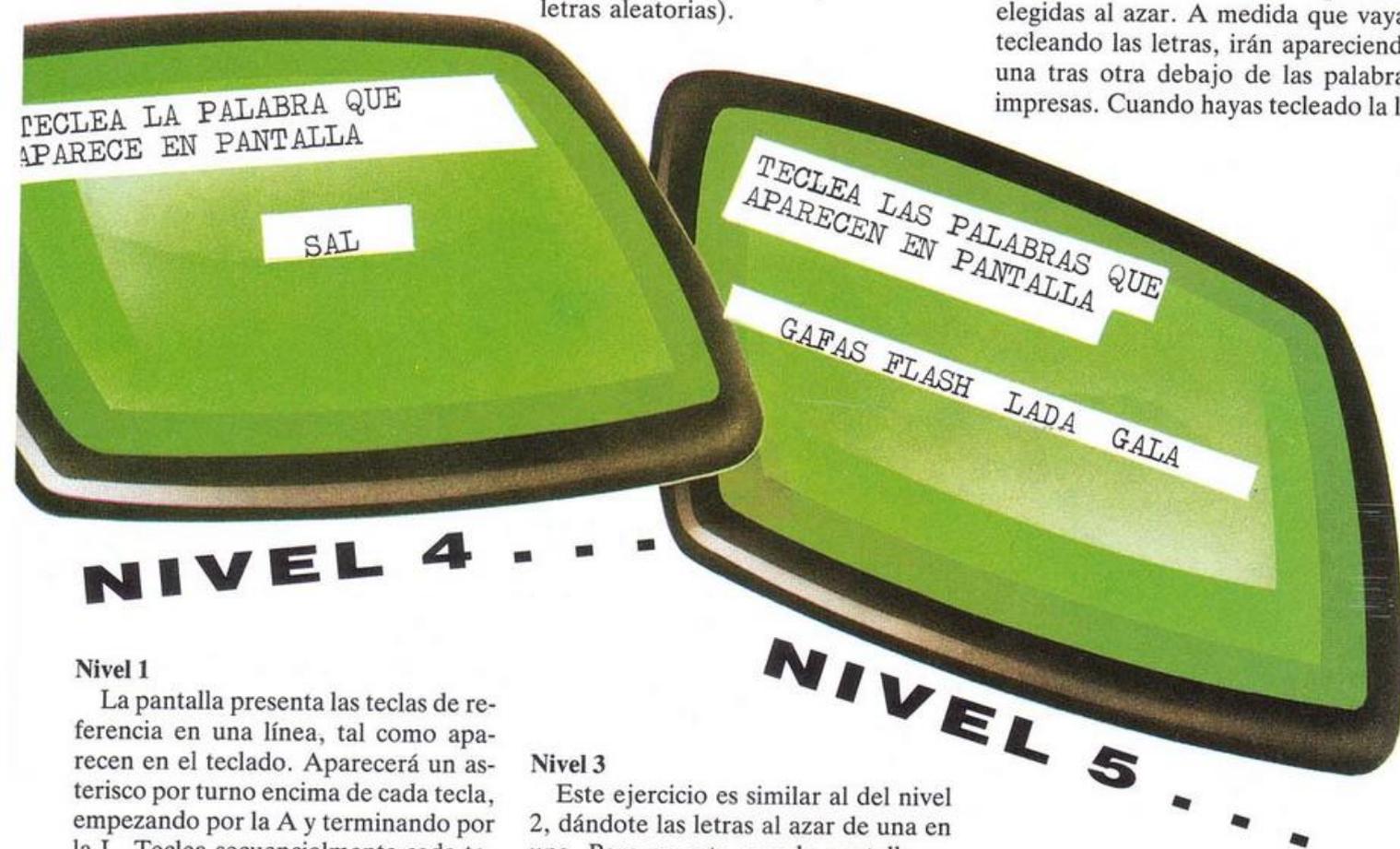
Nivel 2

Cuando ya estés familiarizado con la posición de las teclas, puedes pasar a este nivel, que es parecido al anterior; la única diferencia es que el asterisco se mueve ahora aleatoriamente de una tecla a otra. No intentes hacer trampas mirando al teclado. También aquí te aparecerá cuando termines el tiempo que has tardado y el número de errores cometidos (en cada realización del ejercicio te aparecerán 20 letras aleatorias).

de las teclas de referencia, por ejemplo SAL. Al igual que en el nivel anterior, aparecerán 20 palabras de forma aleatoria en el centro de la pantalla, una tras otra. A medida que vayas copiando cada palabra, un asterisco irá indicando cada letra. También aquí recibirás al final una puntuación por tu velocidad y por tu precisión. Procura no hacer trampas mirando al teclado o te encontrarás con un hándicap ante algunas de las pruebas más duras que veremos posteriormente en el curso.

Nivel 5

En esta ocasión te aparecerá en pantalla una «frase» de 6 o 7 palabras elegidas al azar. A medida que vayas tecleando las letras, irán apareciendo una tras otra debajo de las palabras impresas. Cuando hayas tecleado la lí-



Nivel 1

La pantalla presenta las teclas de referencia en una línea, tal como aparecen en el teclado. Aparecerá un asterisco por turno encima de cada tecla, empezando por la A y terminando por la L. Teclea secuencialmente cada tecla cuando el asterisco se va poniendo sobre ella, utilizando el dedo que le corresponde; recuerda que para teclear la G y la H únicamente debes mover los dedos índices, y utiliza el meñique derecho para «;» o «:». Si cometes una falta, el asterisco se quedará esperándote. Cuando la pulsación de tecla sea correcta, oirás un pequeño pitido, que se convertirá en un

Nivel 3

Este ejercicio es similar al del nivel 2, dándote las letras al azar de una en una. Pero en este caso la pantalla no te muestra su posición sobre el teclado, sino que las presenta aisladas en el centro de la pantalla. También aquí aparecerán 20 letras en cada realización del ejercicio.

Nivel 4

¡Por fin vas a poder teclear algunas palabras! Todas las palabras de este ejercicio están formadas por las letras

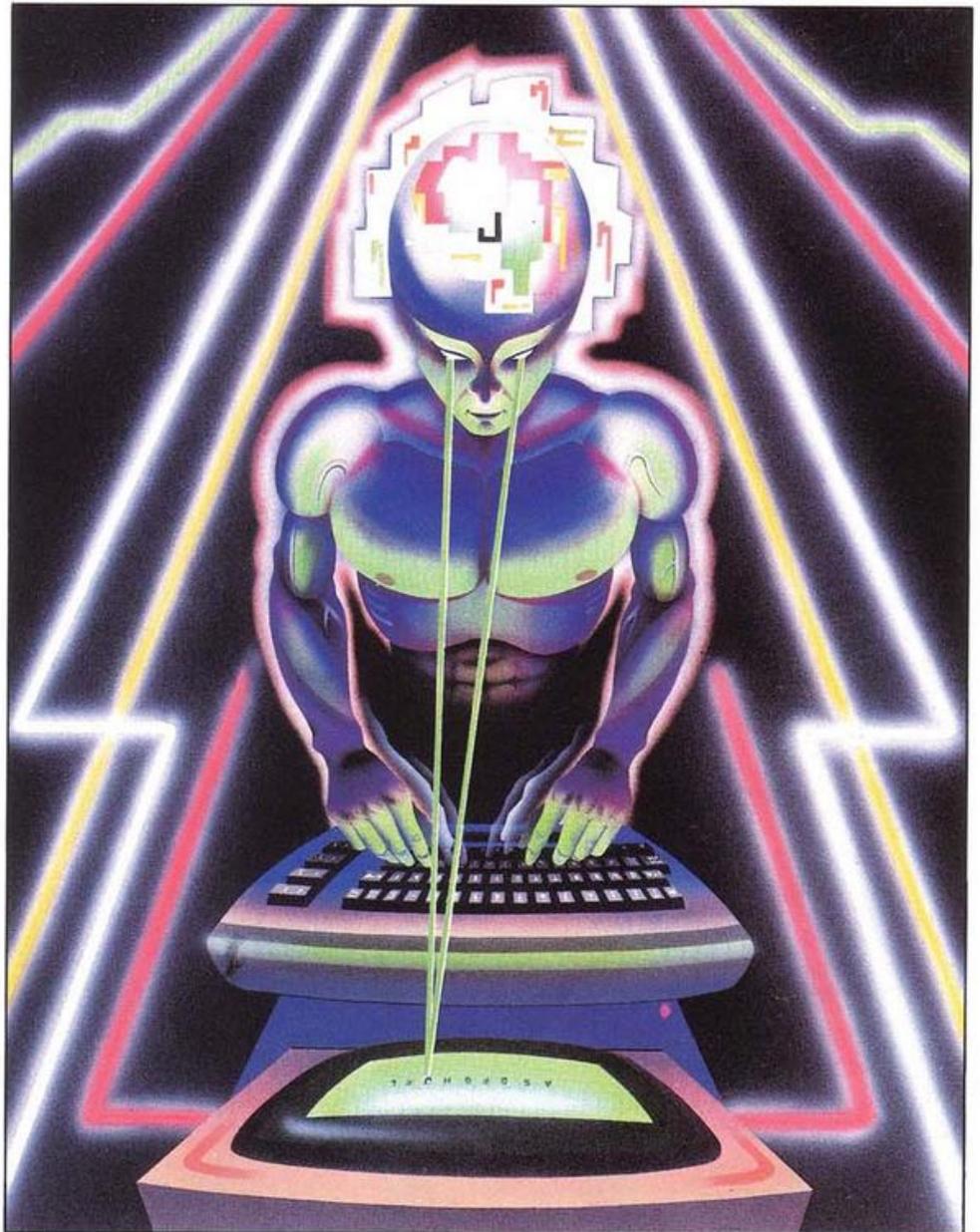
nea completa, se te dirá tu puntuación y la velocidad equivalente en palabras por minuto. Este valor está basado en una longitud media de palabra de 5 letras y un espacio entre cada dos palabras. Cuando teclees las palabras que van apareciendo en pantalla, no te olvides de los espacios. En el Spectrum tienes que pulsar para ello la tecla SPACE con el dedo meñique.

Practica cada uno de los niveles anteriores hasta que puedas hacer los correspondientes ejercicios con facilidad.

```

10 BORDER 1: PAPER 1: INK
7: CLS
20 POKE 23658,8: LET ER=0
30 LET S$="ASDFGHJKL"
100 PRINT INVERSE 1; AT 8,7;
" TUTOR PARA TECLEAR"
110 PRINT "TAB 6;
"SELECCIONA NIVEL (1 A
5)"
120 IF INKEY$=" " THEN GO TO
120
130 LET A$=INKEY$: IF
A$<"1" OR A$>"5" THEN
GO TO 120
140 BEEP .2,10
150 GO SUB VAL A$*100+100
160 GO TO 20
200 GO SUB 1000
210 FOR K=7 TO 23 STEP 2
220 PRINT AT 10,K;"*"
230 LET R$=S$((K-5)/2)
240 GO SUB 1100
250 IF C=0 THEN GO TO 240
260 PRINT AT 10,K;" "
270 NEXT K
280 CLS: GO TO 1300
300 GO SUB 1000
310 FOR K=1 TO 20
320 LET RN=INT (RND*9)*2+1
330 PRINT AT 10,RN+6;"*":
LET R$=S$((RN+1)/2)
340 GO SUB 1100: IF C=0
THEN GO TO 340
350 PRINT AT 10,RN+6;" "
360 NEXT K
370 CLS: GO TO 1300
400 CLS: PRINT; "TECLEA LA
LETRA QUE APARECE EN
PANTALLA"
410 FOR N=1 TO 100: NEXT N
420 POKE 23672,0: POKE
23673,0
430 FOR K=1 TO 20
440 LET RN=INT (RND*9)+1
450 PRINT AT 11,16;S$(RN)
460 LET R$=S$(RN)
470 GO SUB 1100: IF C=0
THEN GO TO 470

```



```

480 PRINT INVERSE 1; AT 11,
16;" ": NEXT K
490 CLS: GO TO 1300
500 CLS: PRINT "TECLEA LA
PALABRA QUE APARECE EN
PANTALLA": POKE 23672,
0: POKE 23673,0
510 LET TL=0: FOR N=1 TO
20: RESTORE: LET RN=INT
(RND*24)+1
520 FOR K=1 TO RN: READ T$:
NEXT K
530 PRINT AT 10,13;" ": PRINT
AT 10,13;T$
540 FOR M=1 TO LEN T$:
PRINT AT 9,11+M;" * "
550 LET R$=T$(M): GO SUB
1100
560 IF C=0 THEN GO TO 550
570 NEXT M
580 LET TL=TL+LEN T$: NEXT
N
590 CLS: PRINT 18,0;
"PALABRAS POR MINUTO="
"; INT (TL*500/(PEEK
23672+256*PEEK
23673)+.5): GO TO 1300
600 CLS: PRINT "TECLEA LAS
PALABRAS QUE APARECEN
EN PANTALLA": LET T$=" "
610 FOR N=1 TO 6: RESTORE:
LET RN=INT (RND*24)+1:
FOR K=1 TO RN: READ X$:
NEXT K

```

```

620 LET T$=T$+X$+" "
630 NEXT N: LET T$=T$ ( TO
    LEN T$-1)
640 POKE 23672,0: POKE
    23673,0: PRINT AT 10,0;
    T$
650 PRINT AT 12,0;: FOR M=1
    TO LEN T$
660 LET R$=T$(M): GO SUB
    1100
670 IF C=0 THEN GO TO 660
680 PRINT T$(M);: NEXT M
690 LET TL=LEN T$: GO TO
    590
1000 CLS: PRINT "PRESIONA
    LA TECLA MARCADA CON
    EL ASTERISCO"
1010 PRINT AT 12,7;"A S D F G
    H J K L"
1020 FOR K=1 TO 100: NEXT
    K: POKE 23672,0: POKE
    23673,0
1030 RETURN
1100 PAUSE 0
1110 LET A$=INKEY$: IF
    A$<>R$ THEN BEEP .2,-
    10: LET ER=ER+1: LET
    C=0: RETURN
1120 BEEP .05,20:LET C=1:
    RETURN
1300 PRINT AT 19,0;
    "TIEMPO=" ;(PEEK
    23672+256*PEEK
    23673)/50;" SEGUNDOS":
    PRINT "NUMERO DE
    ERRORES=" ;ER
1310 PAUSE 100: RETURN
2000 DATA "ALA", "ASA",
    "SALA", "SALAS",
    "SAJAS", "AJADA",
    "FAJADA", "FAJA", "ADA",
    "ALA", "LA", "FA"
2010 DATA "GAS", "GAFAS",
    "Jafa", "JALA", "SANA",
    "ANAGAZA", "ALADAS",
    "FLASH", "AS", "LADA",
    "GALA", "HADA"

```

Cuando ya hayas adquirido un dominio razonable de la fila central del teclado, por ejemplo, conseguir escribir unas 15 palabras por minuto sin co-

meter ni una sola falta, puedes pasar a la siguiente fase.

LAS TECLAS QWERTY

Si añades las siguientes líneas al programa anterior, puedes enriquecer tu repertorio con las teclas de la línea superior, la línea «QWERTY», de las que toma su nombre este tipo de teclado. Carga el programa con LOAD tal como está y teclea a continuación estas nuevas líneas. Algunas de ellas quedarán escritas por encima de otras que ya no hacen falta, otras son completamente nuevas.

```

30 LET S$
   ="QAWSEDRFTGYHUJIKOLP"
210 FOR K=6 TO 24
230 LET R$=S$(K-5)
320 LET RN=INT (RND*19)+1
330 PRINT AT 10,RN+5;"*":
    LET R$=S$(RN)
350 PRINT AT 10,RN+5;" "
440 LET RN=INT (RND*19)+1
530 PRINT AT 10,13;"
    ": PRINT AT 10,13;T$
540 FOR M=1 TO LEN T$:
    PRINT AT 9,11+M;" * "
610 FOR N=1 TO 4: RESTORE:
    LET RN=INT (RND*24)+1:
    FOR K=1 TO RN: READ X$:
    NEXT K
1010 PRINT AT 12,6;S$
2000 DATA "SEDA", "FRESA",
    "LOSA", "JURAS",
    "PIERDO", "TIPOS",
    "TORERO", "PITIDO",
    "RELIQUIA", "HALITO",
    "LITROS", "POLIPOS"
2010 DATA "TRILITA", "FERIA",
    "GRATIS", "PEPITA",
    "LUISILLO", "TAURO",
    "LEO", "QUERIDOS",
    "ARIES", "SAGITARIO",
    "PAYASO", "KILO"

```

Al ejecutar el programa, te aparecerá en pantalla el menú de cinco opciones que ya conoces. El nivel 1 presenta la siguiente secuencia de letras: QAWSEDRFTGYHUJIKOLP. Como invitación a teclear, te aparecerá el as-

terisco por encima de dichas teclas, cambiando de izquierda a derecha igual que antes.

Los niveles 2 y 3 funcionan exactamente de la misma manera que en la primera prueba, haciendo que teclees estos caracteres de forma aleatoria pero con un conjunto de letras más amplio.

Los niveles 4 y 5 son más difíciles que antes, ya que en ellos se te pide que teclees palabras más largas y más difíciles, cambiando de una fila de teclas a otra dentro de cada palabra.

Siéntate bien, con tus dedos correctamente colocados sobre las teclas de referencia e intenta pulsar las teclas de la fila de arriba, moviendo únicamente el dedo necesario para pulsar cada tecla y sin mover el resto de la mano.



Esta vez tendrás que utilizar el dedo meñique de tu mano izquierda para la Q, además de para la A, el siguiente dedo para la S y la W, etcétera, hasta llegar al dedo meñique de tu mano derecha que utilizarás para ; y para P. También en este caso tus dedos índices tendrán más trabajo que los demás; tienes que utilizar el índice de tu mano izquierda para F, G, R y T mientras que el índice derecho se encargará de pulsar H, J, Y y U. En cuanto hayas pulsado una tecla de la fila de arriba, el dedo correspondiente debe volver a su posición.

LAS TECLAS ZXCV

Cuando hayas adquirido una cierta práctica en el uso de las teclas de la fila

superior mezcladas con la fila central, habrá llegado el momento de pasar a la última fila.

La parte tercera del curso te enseña la manera de usar las líneas de fondo y central combinadas, progresando a través de los cinco niveles de habilidad, pero todavía sin mezclar las tres filas de teclas.

Aquí tienes las modificaciones que debes introducir en el programa:

```

30 LET S$
   ="AZSXDCFVGBHJMKL"
21Ø FOR K=6 TO 21
32Ø LET RN=INT (RND*16)+1
44Ø LET RN=INT (RND*16)+1
61Ø FOR N=1 TO 5: RESTORE:
   LET RN=INT (RND*24)+1:
   FOR K=1 TO RN: READ X$:

```

NEXT K

```

1Ø1Ø PRINT AT 12,6;S$
2ØØØ DATA "CASA", "CASACA",
  "CASADA", "MAZA",
  "MALAGA", "MANA",
  "MANANA", "BALA",
  "ANDAS", "CANGAS",
  "BLANDAS", "BLAS"
2Ø1Ø DATA "FACHA", "DACHA",
  "ALBA", "CLAMAN",
  "CLAVAN", "ZAFAS",
  "AVAL", "MALDAD",
  "MAL", "FAMA",
  "GANGA", "DAGA"

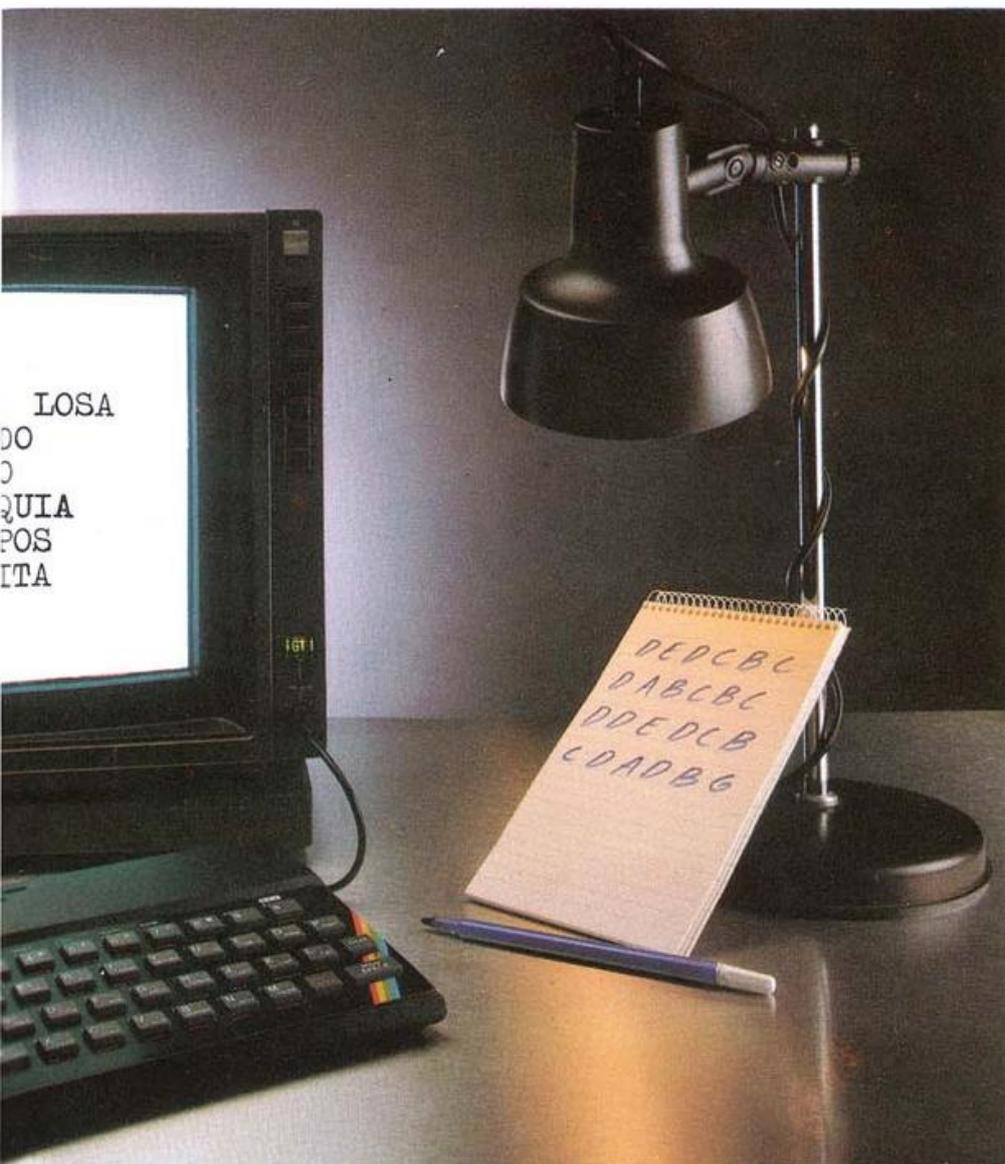
```

Al ejecutar ahora el programa, podrás ir progresando a lo largo de las cinco lecciones, pero con letras y palabras sacadas de las dos filas inferiores del teclado.

Una vez más, siéntate correctamente y coloca los dedos sobre las teclas de referencia. Esta vez deberás desplazar tus dedos hacia la fila de abajo del teclado volviendo a colocarlos inmediatamente sobre su tecla de referencia. El dedo meñique de tu mano izquierda actuará sobre la A y la Z, el anular sobre la S y la X, etcétera. El dedo índice de la mano izquierda se utilizará para las letras F, G, V y B mientras que el índice derecho se encargará para H, J, N y M. Los otros dedos tienen que actuar sobre las teclas de puntuación de la línea de fondo, que suelen variar de unos teclados a otros. En el caso del Spectrum no se pueden utilizar estas teclas si no se pulsa SYMBOL SHIFT; nos ocuparemos de esto más adelante. Observa que las teclas de puntuación no se incluyen en las palabras de prueba. En un artículo posterior tendrás la oportunidad de practicar los signos de puntuación con una nueva sección de programa escrita para este fin.

EL ALFABETO COMPLETO

Cuando tu habilidad en la utilización de las teclas de la fila inferior, sea comparable a la que tienes para la fila superior, puedes pasar a la siguiente etapa del curso. Esta vez puedes practicar utilizando todo el alfabeto.



P y R

¿Qué velocidad debo intentar alcanzar para empezar?

En esta primera fase del curso, tienes que preocuparte más de la exactitud que de la velocidad. Cuando ya consigas teclear todas las letras y todas las palabras sin errores, puedes intentar adquirir algo más de velocidad. Tienes que practicar con los niveles 1, 2 y 3 hasta que estés en condiciones de teclear las 20 letras en unos 12 o 13 segundos; para los niveles 4 y 5 debes intentar escribir unas 15 palabras o más por minuto.

Aquí tienes las alteraciones que tienes que introducir en tu programa anterior:

```

30 LET S$="QAZWSXEDCRFVT
   GBYHNUJMIKOLP"
21Ø FOR K=2 TO 27
23Ø LET R$=S$(K-1)
32Ø LET RN=INT (RND*26)+1
33Ø PRINT AT 1Ø,RN+1;"*":
   LET R$=S$(RN)
35Ø PRINT AT 1Ø,RN+1;" "
44Ø LET RN=INT (RND*26)+1
53Ø PRINT AT 1Ø,13;"
   ": PRINT AT 1Ø,13;T$
54Ø FOR M=1 TO LEN T$:
   PRINT AT 9,11+M;" * "
61Ø FOR N=1 TO 5: RESTORE:
   LET RN=INT (RND*24)+1:
   FOR K=1 TO RN: READ X$:
   NEXT K
    
```

```

1Ø1Ø PRINT AT 12,2;S$
2ØØØ DATA "VIENTO", "JUNTA",
"QUIEN", "VENTANA",
"MESA", "PROCESO",
"AYER", "BASTON",
"VISTA", "GASTO",
"KIMONO", "ROBERTO"
2Ø1Ø DATA "RADIO", "LOCAL",
"SOFA", "SILLA",
"EXTREMO", "ZAPATO",
"PUENTE", "FACIL",
"FUEGO", "EXACTO",
"LISTA", "REVISTA"
    
```

Como ocurre con todos los demás programas de este curso de mecanografía, cuando te familiarices con las palabras que figuran en las sentencias DATA, conviene que las cambies por otras nuevas. Sobre todo ten cuidado de no utilizar menos palabras que en el programa original, para no obtener un mensaje de error del tipo de OUT OF DATA. Naturalmente, puedes tener más entradas, pero no serán leídas por el ordenador a menos que cambies el programa.

Recorre progresivamente las cinco lecciones, igual que antes. Acuérdate de que tus dedos regresen a su posición correcta sobre las teclas de referencia después de pulsar una tecla en la fila de arriba o en la de abajo.

MEJOR TODAVIA

A medida que el curso va progresando, irás teniendo la oportunidad de mejorar tu precisión y tu velocidad sobre todas las teclas de letras del teclado. Veremos también la manera de incluir las teclas numéricas y los signos de puntuación, lo cual resulta crucial

para la introducción correcta de los listados de programas.

Añadiendo todas estas teclas y las teclas de control, tal como SHIFT, te encontrarás en seguida en condiciones de teclear con gran comodidad frases complejas y líneas de programa largas; veremos además una gran cantidad de ejercicios que te permitirán hacer prácticas para aumentar aún más tu destreza.

MEJORANDO EL TECLADO DEL SPECTRUM

Puedes introducir algunas mejoras en el teclado del Spectrum, de forma que resulte completamente más fácil aprender a teclear en él, al sustituir el teclado estándar de Sinclair por uno nuevo.

Existen en la actualidad varias casas que venden teclados «reales» para el Spectrum, si bien hay algunos problemas de adaptación que son muy fáciles de tener en cuenta.

La versión más fácil de utilizar es una unidad completamente separada que simplemente se enchufa en la parte trasera de tu ordenador. El teclado antiguo queda inoperativo y haces todas tus pruebas mecanográficas en el teclado nuevo, este último se asemeja mucho a un teclado totalmente profesional.

En otros tipos tienes que cambiar el teclado original, lo que te obliga a abrir tu ordenador, desconectar los dos cables planos del teclado antiguo y conectarlos al nuevo.

Este método tiene el inconveniente de que tu garantía deja de ser válida en el momento en que abres la carcasa del Spectrum.

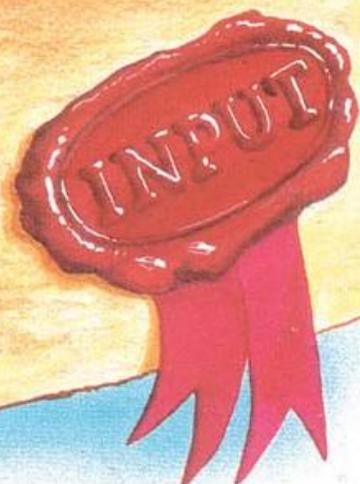


LOS MEJORES DE INPUT SINCLAIR

PUESTO	TITULO	PORCENTAJE
1.º	Comando	24,2 %
2.º	Rambo	16,1 %
3.º	Green Beret	13 %
4.º	Saboteur	13 %
5.º	Strike Force Harrier	8,1 %
6.º	Nightmare Rally	6,4 %
7.º	Batman	4,8 %
8.º	Winter Games	4,8 %
9.º	Street Hawk	4,8 %
10.º	Avenger	4,8 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Febrero de 1987



SOFTREPORTAJE

LO ULTIMO DE DINAMIC

Allá por 1980, aprendieron a programar con un ZX-81. Cuatro años después, y casi sin quererlo, sus programas alcanzaban los primeros puestos en las listas de éxitos. Más tarde firmaron importantes contratos con prestigiosas firmas británicas, y para cuando comenzaron a darse cuenta de lo que estaba ocurriendo, ya tenían su propia compañía productora-distribuidora-exportadora de software español.

Seguro que ya sabéis a quiénes nos estamos refiriendo. Estamos hablando, naturalmente, de **Dinamic**, un grupo que se ha convertido en todo un símbolo para los aficionados al deporte del joystick.

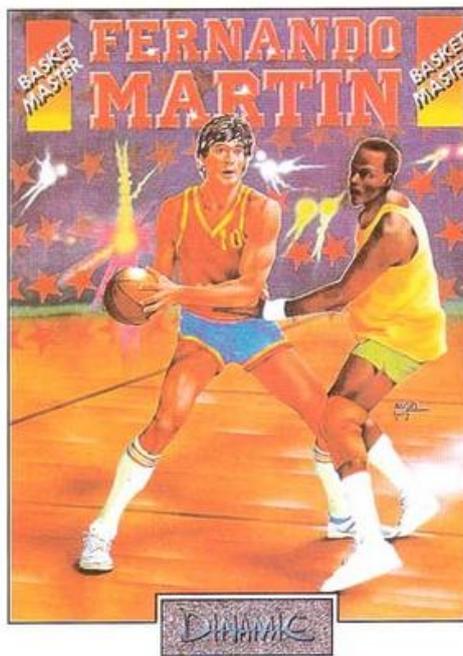
Después del éxito alcanzado por **Phantomas II**, programa que aún sigue subiendo como la espuma, **Dinamic** ha vuelto a ponerse de actualidad con el lanzamiento de una avalancha de títulos, que rompen con la tradición del grupo de presentar sus creaciones de dos en dos. Parece ser que tampoco ellos querían perderse los suculentos beneficios del despilfarro navideño, y buena prueba de ello es la gran campaña publicitaria que montaron en torno a sus últimos programas.

INPUT ha conseguido información sobre estos programas, y a continuación os la transmitimos tal cual ha llegado hasta nosotros.

★★★★★★★

FERNANDO MARTIN BASKET MASTER

Los creadores de este programa han conseguido reproducir el estilo, las jugadas, los errores y hasta el aspecto físico de ese gran baloncestista español que es

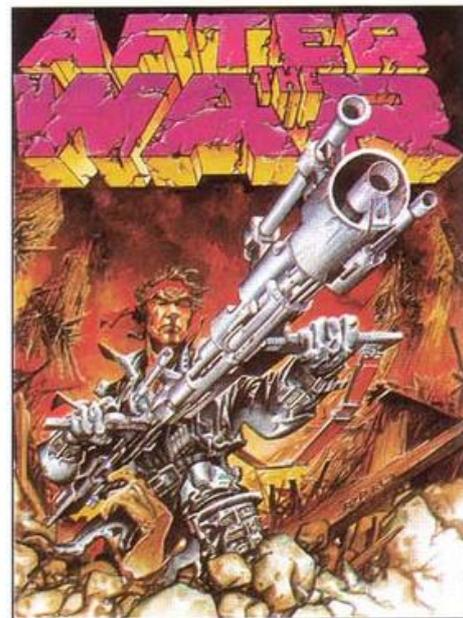


Fernando Martín, que ahora triunfa en el equipo norteamericano Portland, de Oregón. Según nos comentan quienes ya han tenido oportunidad de verlo, es casi tan real como jugar con el propio Fernando Martín en persona, con la diferencia de que en la vida real no llegaríamos ni a tocar la pelota...

La única forma de conseguir derrotar al «basket master» que se nos fue a los EEUU, es neutralizando sus espectaculares fintas, bloqueando sus lanzamientos de tres puntos, y eludiendo sus famosos taponés. Necesitarás mucho tiempo para conseguirlo, si es que alguna vez lo logras.



★★★★★★★



AFTER THE WAR

Después de una devastadora guerra nuclear, el planeta Tierra queda sumido bajo los efectos de la radiactividad. Los escasos supervivientes se agrupan en hordas que luchan a muerte por conseguir alimentos, entre las ruinas de las grandes ciudades arrasadas. Las bandas de salteadores y los conversores gamma ocultos en las esquinas, acaban pronto con los más débiles. Sólo un hombre dispuesto a todo y bien pertrechado, puede sobrevivir por sí solo entre los restos de un mundo condenado.

La calidad de este programa es ligeramente inferior a las producciones «estrella» de **Dinamic**, pero su precio será lo suficientemente competitivo como para asegurar, al menos, un éxito discreto.

★★★★★★★

ARMY MOVES

Army Moves no sólo es un programa hecho en España (un detalle que siempre debemos valorar, y que es común a todas las producciones de **Dinamic**), sino que además su tema es una versión «a la española» de las

hazañas de los cuerpos de élite extranjeros. El protagonista es un intrépido miembro del C.O.E. (Cuerpo de Operaciones Especiales), y su misión consiste en infiltrarse en las líneas enemigas para conseguir los planos de una mortífera bomba de partículas. A pie, en jeep o en helicóptero, el héroe hará todo lo posible por alcanzar sus objetivos.

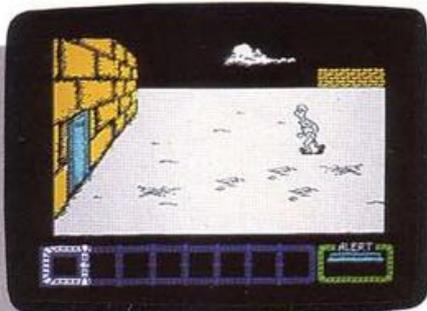


que lo habitan y acechan en todos los rincones de la mansión.

★★★★★★

DUSTIN

Muy poco es lo que sabemos de este programa. Parece ser que consiste en ayudar a un tal Dustin a escapar de la prisión donde se encuentra encerrado. La principal novedad que aporta, es la posibilidad de transportar a la vez nada menos que **ocho objetos**, usando un selector para utilizar cualquiera de ellos en el momento oportuno. Hay que destacar, además, que existen diversos sistemas de lucha, adecuados a cada tipo de arma.



★★★★★★

GAME OVER

Un programa con un repertorio gráfico realmente impresionante. Con toda sinceridad podemos decir que pocas veces hemos visto algo parecido. Sin duda, será un gran éxito.

El juego se ambienta en tres fases, cada una de las cuales se desarrolla en un planeta distinto, con diferentes peligros, escenarios y personajes. El objetivo consiste en llegar hasta el tercer planeta, y en él destruir el palacio imperial de la soberana del Universo.

Este programa, cuyo título se podría traducir como «Juego finalizado», dará que hablar.

★★★★★★

FREDDY HARDEST

Por último, un programa que puede ser un auténtico bombazo.

Un agente secreto de la Confederación Galáctica se estrella con su nave en un planeta hostil. Desesperado, se infiltra en la base enemiga y busca una nave con la que poder escapar.

La diversidad y perfección de los gráficos, la ambientación, el color y un alto grado de interés, pueden convertir a este programa en un número uno. El tiempo dirá si tenemos razón en nuestros augurios a «Freddy, el más duro».

★★★★★★

ARMY MOVES



★★★★★★

NONAMED

Se trata de otro «serie B» de Dynamic, que ya ha sido ampliamente comercializado. Un aspirante al honor de ser nombrado Caballero del Rey, debe sufrir una terrible prueba en la que casi todos sus antecesores sucumbieron: salir con vida del Castillo sin Nombre, eludiendo el roce mortal de los ogros



GUIA DEL COMPRADOR DE SOFT

LOS 50 MEJORES PROGRAMAS DEL MERCADO

Para la elaboración de esta lista, se han tenido en cuenta los rankings de publicaciones nacionales y extranjeras, las opiniones vertidas por nuestros lectores a través de

«Los Mejores de INPUT», el testimonio de vendedores y profesionales del ramo y, cómo no, el humilde criterio de quienes hacemos esta revista.

TITULO CALIFICACION
(1 a 10)

TITULO CALIFICACION
(1 a 10)

THE GREAT ESCAPE	9
HEARTLAND	9
FIRELORD	9
THE SACRED ARMOR OF ANTIRIAD	9
STRIKE FORCE HARRIER	9
PYRACUSE	8
FROST BYTE	8
BATMAN	8
URIDIUM	8
COMMANDO	8
GREEN BERET	8
MOVIE	8
SKYFOX	8
ACE	8
SIR FRED	8
STAINLESS STEEL	7
JACK THE NIPPER	7
WINTER GAMES	7
PHANTOMAS II	7
GHOST'n GOBLINS	7
THANATOS	7
CAMELOT WARRIORS	7
DYNAMITE DAN II	7
CAULDRON II	7
INFILTRATOR	7

FUTURE KNIGHT	7
ROCKMAN	7
NUCLEAR BOWLS	7
MISTERIO DEL NILO	7
SABOTEUR	7
THE WAY OF THE TIGER	7
DRAGON'S LAIR	7
EQUINOX	7
PING-PONG	7
FAIRLIGHT II	7
AVENGER	7
COBRA	6
TENNIS	6
XAR-Q	6
BOBBY BEARING	6
NIGHTMARE RALLY	6
ASTERIX Y EL CALDERO MAGICO	6
MERMAID MADNESS	6
MIAMI VICE	6
STREET HAWK	6
BIGGLES	6
ZYTHUM	6
SHOW JUMPING	6
TRACER	6
PAPER BOY	6

NOVEDADES EN CANDIDATURA

TRAILBLAZER
TOP GUN
GOONIES
ARMY MOVES
ALIENS
EXPLODING FIST II
HARDBALL
THEY SOLD A MILLION
SUPER SOCCER
DEACTIVATORS
LIVINGSTONE SUPONGO
COSA NOSTRA
RAMON RODRIGUEZ
SILENT SERVICE
BREAK-THRU
CRYSTAL CASTLES
SHAO LIN'S ROAD
REVOLUTION
PRODIGY
BAZOOKA BILL
HIGHLANDER
SHORT CIRCUIT

FAIRLIGHT II: LA LEYENDA CONTINUA

Esencialmente, esta segunda parte de la leyenda de Fairlight no presenta grandes diferencias con respecto a la primera. Se han cambiado o añadido algunos pequeños detalles, ha variado el objetivo del juego y la forma de alcanzarlo, y poco más. No obstante, los autores han introducido un original sistema de carga en dos partes, en el que se ensaya por primera vez una curiosa forma de compatibilidad entre el modelo de 48 K y el extendido de 128 K. Si tu micro es un Spectrum 48 K (sea o no «plus») el programa se almacenará en memoria en dos partes, la primera al principio, y la segunda cuando completes la parte inicial de la aventura. Pero si tienes la suerte de poseer un 128 K, entonces todo el

programa se cargará de una vez. Entre los detalles añadidos al juego que antes comentábamos, destacaremos algunos especialmente interesantes, como la inercia que adquieren los personajes en sus desplazamientos, la incidencia del peso de los objetos transportados sobre el protagonista, y sobre todo, un cierto grado de inteligencia «independiente», que opera al margen de las órdenes dadas por el usuario. La verdad es que todos estos aspectos apenas son apreciables, aunque realmente están presentes.

Fairlight II es un programa recomendable tanto si tienes la primera parte como si no, pero en caso afirmativo, te advertimos que



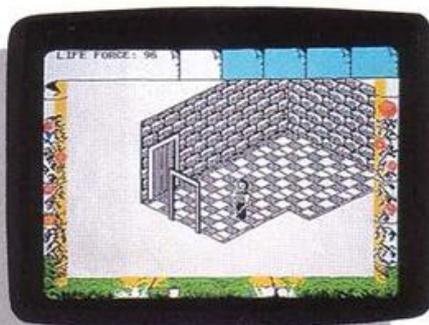
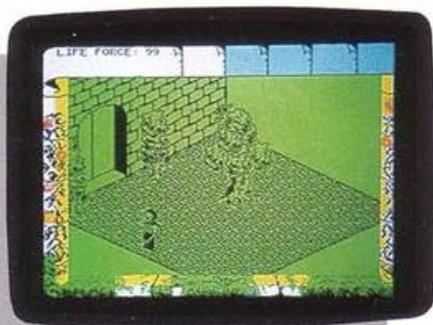
quizás encuentres el juego demasiado parecido a su versión precedente. En todo caso, la elección es tuya.

DATOS GENERALES

TITULO Fairlight II
FABRICANTE The Edge
ORDENADOR Spectrum
TEMA DEL PROGRAMA Aventura Fantástica

CALIFICACION (sobre 10 pts.)

ORIGINALIDAD	6
GRAFICOS	8
INTERES	7
COLOR	6
SONIDO	8
TOTAL	35



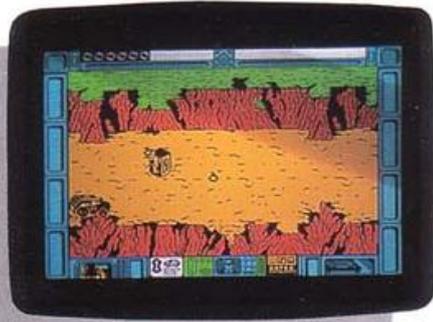
BREAK-THRU

Templa tus nervios, ármate de valor y sangre fría, y siéntate a los mandos del más sofisticado vehículo de combate del mundo. Prepárate para atravesar con él puentes, montañas, ciudades y aeropuertos, en una desenfadada lucha contra los tanques, jeeps, helicópteros, lanzallamas, campos de minas y demás artefactos y procedimientos bélicos en manos de los bien apertrechados enemigos. Suponemos que esta breve

descripción ha bastado para dejar bien claro cuál es el contenido del programa. En efecto, se trata de uno de esos trepidantes juegos de «acción continua», en los que valen más unos buenos reflejos para esquivar obstáculos, que todo el talento del mundo para resolver enigmas. Dicho en otras palabras, Break-Thru (que significa «Abrirse paso») es un arcade especialmente recomendado a los locos del joystick, que no son pocos, con todos los ingredientes



necesarios para garantizar un alto grado de adicción. El programa ha sido presentado recientemente como una de las



últimas novedades de US GOLD. Está basado en el popular juego de las máquinas tragaperras del mismo nombre, y su adaptación para Spectrum, realizada con gran acierto por Paul Houbart, respeta hasta en el último detalle la versión original. Los gráficos son prácticamente idénticos a los que hemos visto en las máquinas, y el *scroll* de pantalla, como el desplazamiento del vehículo, también discurre con la misma rapidez. Sólo debemos reseñar una pequeña nota negativa en el sonido, prácticamente ausente a lo largo de todo el desarrollo del programa. Sin embargo, **Break-Thru** puede considerarse, con toda justicia, como uno de los mejores juegos de la temporada a despecho del poco ruido que ha provocado hasta ahora en el mercado del software en el Reino Unido.

DATOS GENERALES

TITULO Break-Thru
FABRICANTE US Gold
ORDENADOR Spectrum
TEMA DEL PROGRAMA Arcade

CALIFICACION (sobre 10 ptos.)

ORIGINALIDAD	7
GRAFICOS	7
INTERES	8
COLOR	7
SONIDO	5
TOTAL	34

LA SENDA DE SHAO LIN

«Otra de artes marciales», habrá pensado más de uno. Pues sí, la cosa va de chinos, saltos, patadas y demás ingredientes clásicos, pero no se trata de otro programa cualquiera con el tema de siempre, sino de uno muy especial, una vídeo-aventura que ha obtenido un sensacional éxito en las máquinas tragaperras de todo el mundo: **Shao Lin's Road**. Con gran acierto, el equipo de **The Edge** se ha encargado de codificar el programa en su versión para

Spectrum, aunque sobre el papel es **Konami** la firma titular de los derechos. El planteamiento básico del tema es idéntico al original, y suponemos que muchos ya lo conocéis. Por si no es así, os haremos un pequeño resumen: **Lee**, el héroe de la aventura, ha completado su formación como maestro de artes marciales, y llevado por su espíritu inquieto, decide poner en práctica sus habilidades

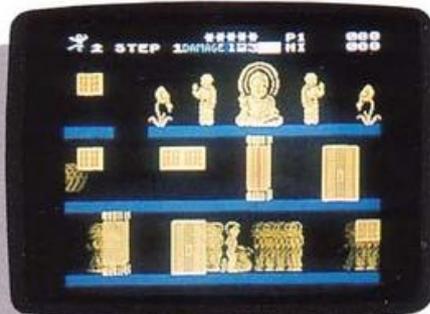
fuera del templo donde ha permanecido recluso durante su aprendizaje. Para su desgracia, los guardianes del tesoro de Triad, temerosos de que Lee revele su secreto, tratan de impedir por todos los medios su salida. La misión

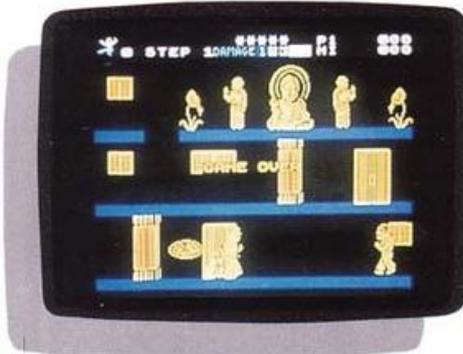
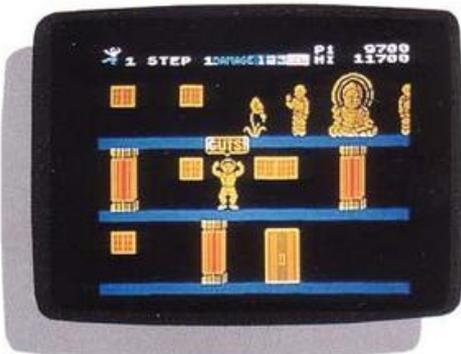
DATOS GENERALES

TITULO Shao Lin's Road
FABRICANTE Konami
ORDENADOR Spectrum
TEMA DEL PROGRAMA Artes marciales

CALIFICACION (sobre 10 ptos.)

ORIGINALIDAD	6
GRAFICOS	7
INTERES	7
COLOR	6
SONIDO	7
TOTAL	33





consiste en escapar, matando al mayor número posible de esbirros. Para ello, Lee dispone de una poderosa patada de kárate, un prodigioso salto (no podía faltar), y ciertos poderes mágicos que sólo puede utilizar en algunas ocasiones. El programa tiene todo lo necesario para gustar a pesar de su temática, muy conocida para los aficionados a estos juegos.

DEEP STRIKE

Deep Strike es un magnífico simulador de combate aéreo, ambientado en los felices tiempos del Barón Rojo, allá por 1917. Aunque cuando llegó a nuestra redacción era para nosotros un perfecto desconocido, debido a su ausencia en páginas de publicidad, pronto pudimos comprobar que se trataba de un programa con todos los elementos necesarios para entusiasmar a todo buen aficionado. El juego comienza sobre la pista de despegue. Después de consultar un rudimentario mapa para hacernos una idea de dónde estamos y adónde

debemos ir, emprendemos el vuelo. Inmediatamente, nos veremos envueltos en emocionantes combates, salpicados por las explosiones del

fuego dirigido desde emplazamientos terrestres, discurriendo entre grandes llanuras y escarpados relieves.

Deep Strike es un programa que no puede faltar en la colección de los aficionados a los simuladores de vuelo. Os lo recomendamos.

DATOS GENERALES

TITULO Deep Strike

FABRICANTE Erbe

ORDENADOR Spectrum

TEMA DEL PROGRAMA
Simulador de Vuelo

CALIFICACION (sobre 10 pts.)

ORIGINALIDAD 8

GRAFICOS 7

INTERES 8

COLOR 7

SONIDO 6

TOTAL 36



NUCLEAR BOWLS

Después de un largo silencio, **Made in Spain** vuelve a la carga con dos sensacionales programas: «El Misterio del Nilo» y «Nuclear Bowls». Con ellos inauguran dos nuevas series, una dedicada a dar a conocer

a programadores noveles ajenos al grupo (bajo el sello «Diabolic»), y otra reservada exclusivamente a las geniales creaciones de los «Made in Spain» propiamente dichos. De estos dos programas, el primero

en llegar a nuestra redacción ha sido **Nuclear Bowls**, una vídeo-aventura de tema poco original, pero con un alto grado de adicción e interés. Bowls es el ingeniero jefe de la central nuclear más potente de la Tierra. Un buen día, al llegar a su trabajo, descubre horrorizado que en su ausencia se ha producido una

terrible catástrofe: uno de los reactores se ha fundido, y la radiación ha transformado a los operarios en peligrosos mutantes. Sin pensárselo dos veces, Bowls se coloca su coraza anti-radiaciones, y se dispone a detener como sea el proceso de reacción en cadena. Para lograrlo, deberá reconstruir el reactor averiado con los objetos que vaya encontrando en su camino, alimentarlo de combustible, y ponerlo de nuevo en marcha desde la sala de control central.

Existe un límite de tiempo para alcanzar el objetivo final, ya que una prolongada exposición a las radiaciones sería fatal para Bowls. La única ayuda de que dispone nuestro esforzado protagonista, es un arma defensiva que funciona a base de «agua pesada», un preciado líquido con escasas reservas.

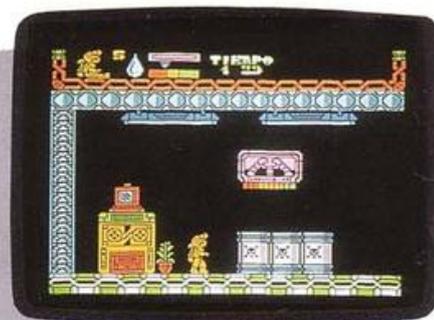
Para terminar, no podemos dejar de señalar un par de detalles verdaderamente geniales, como son

DATOS GENERALES

TITULO Nuclear Bowls
FABRICANTE Made in Spain (Zigurat)
ORDENADOR Spectrum
TEMA DEL PROGRAMA Central Nuclear

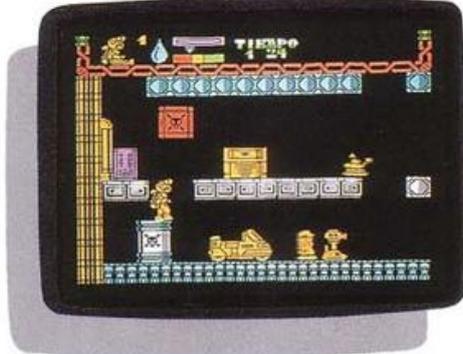
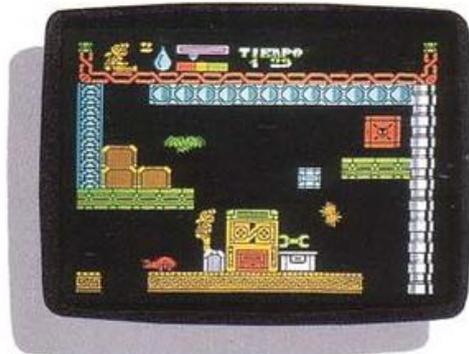
CALIFICACION (sobre 10 ptos.)

ORIGINALIDAD	6
GRAFICOS	8
INTERES	8
COLOR	8
SONIDO	6
TOTAL	36



el diseño gráfico y el color, dos aspectos en los que Manuel, Mario y

Amadeo han sabido lucirse a lo grande.

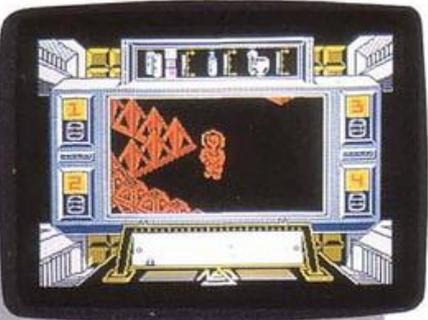


EL LABORATORIO GENETICO DE WARDLOCK

Hasta donde los bancos de memoria alcanzaban a recordar, en el mundo sólo habían existido las máquinas. Nadie creyó jamás en la existencia de vida orgánica, hasta que el supercomputador **Wardlock**, después de múltiples e infructuosos intentos, consiguió crear en su laboratorio dos

pruebas irrefutables y absolutamente sin precedentes: un bebé humano y un ser sintético. Ante tal acontecimiento, la reacción de las máquinas fue unánime: todos temían que las criaturas de Wardlock pusieran en peligro el orden establecido. Pero el genio creador de

la vida orgánica supo tranquilizarles, haciéndoles ver que sólo quería a sus nuevos seres para experimentar con ellos, como lo había hecho antes con las horribles criaturas que poblaban su laboratorio. Desgraciadamente para Wardlock, **Solo**, el ser sintético por él creado, descubre sus planes y decide huir al espacio exterior, llevándose consigo a **Nejo**, el bebé humano. La única forma de escapar del laboratorio de Wardlock, es atravesando cuatro zonas, una de hielo, otra vegetal, otra técnica, y por último una zona de fuego. En



cada una de ellas, no sólo acechan un sinnúmero de peligros, sino que además está el inconveniente adicional de tener que procurar al bebé todos los cuidados que necesita, alimentarle, lavarle y protegerle. En definitiva, una misión para la que se necesita una gran dosis de paciencia y habilidad.

Estamos ante un estupendo programa que puede presumir de tener, además de unos buenos gráficos, un originalísimo argumento. Los escenarios tridimensionales, el color y la ambientación, son tres de los aspectos que más nos han gustado del juego, aunque podemos afirmar sin reservas que también merece una opinión favorable en cuanto a todo lo demás. En resumen, un buen programa, sin ningún género de dudas.

DATOS GENERALES

TITULO Prodigy
FABRICANTE Electric Dreams
ORDENADOR Spectrum 48 K/128 K/ Plus
TEMA Arcade Tridimensional

CALIFICACION (sobre 10 ptos.)

ORIGINALIDAD	9
GRAFICOS	8
INTERES	8
COLOR	7
SONIDO	6
TOTAL	38

AVENTURAS DE UN RODAMIENTO

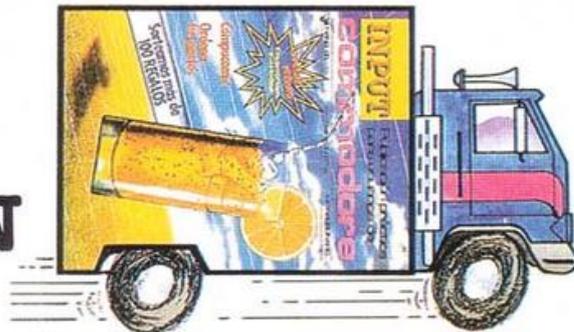
Desde estas páginas hemos tenido oportunidad de comentar las desventuras de todo tipo de curiosos personajes, pero nunca hasta ahora nos habíamos encontrado con uno tan peculiar como Bobby: nada

menos que **un rodamiento extraviado**. El objeto del juego consiste en devolverlo sano y salvo a su angustiada familia, hallando el camino de vuelta en un intrincado laberinto de metaplanos, habitado

por los temibles «Esferoides diabólicos». Según afirman sus autores, **Bobby Bearing** hace una original reinterpretación del movimiento en el espacio tridimensional, gracias a una nueva y supuestamente «revolucionaria» técnica bautizada con el pomposo nombre de «**Curvispace 3D**». No obstante, hemos de decir que el único aspecto

**LA
 REDACCION
 CAMBIA
 DE
 DIRECCION**

ESTAMOS



**Aribau
 n.º 185
 planta, 1
 08021
 Barcelona**

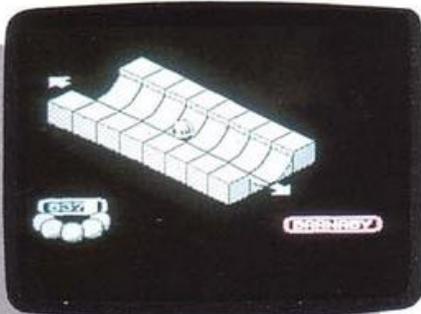
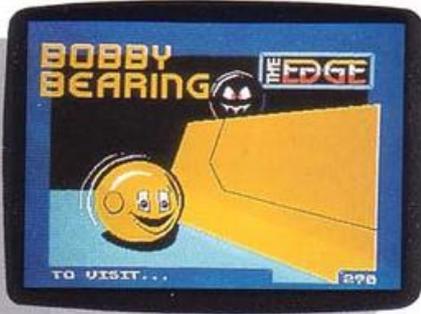
DATOS GENERALES

TITULO Bobby Bearing
FABRICANTE The Edge
ORDENADOR Spectrum 48 K
TEMA Arcade



CALIFICACION (sobre 10 ptos.)

ORIGINALIDAD	8
GRAFICOS	8
INTERES	7
COLOR	6
SONIDO	5
TOTAL	34



protagonista, que, como corresponde a un rodamiento, avanza girando sobre sí mismo. Por lo demás, se trata del mismo esquema 3D de siempre, sin ninguna innovación técnica apreciable.

Aunque la copia que hemos probado para hacer este comentario está en inglés, Dro Soft, firma encargada de la comercialización en España del programa, nos ha confirmado que ya está en el mercado la versión en castellano. Como recordaréis, esta misma firma adaptó recientemente a nuestro idioma un excelente

donde hemos encontrado algunos rasgos calificables como novedosos, es en el movimiento del

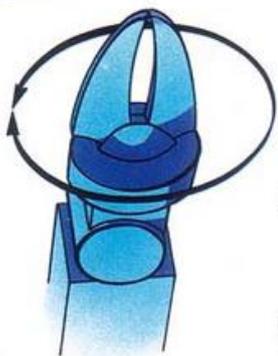
programa que ya hemos comentado en INPUT, Tujad, producido originalmente por Ariola Soft.

GANADORES DE LOS MEJORES DE INPUT SINCLAIR

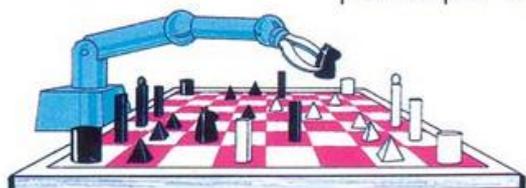
En el sorteo correspondiente al número 14 entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Gustavo Adolfo Pacheco Navas	Barcelona	Oletoro
José María Suárez Hernández	Madrid	Profanation
José Mario Graña Pérez	Santander (Cantabria)	Skyfox
F. Javier Gómez López	Sevilla	Basketball Inter
Juan Antonio Gómez Yélamos	Málaga	The way of the tiger
Héctor Solans Folguera	Barcelona	Scuba Dive
Miguel Angel Sal Cabo	Rudagüera (Cantabria)	Winter Games
Miguel Angel Farré Riera	Granollers (Barcelona)	Comando
Luis Remacha Monton	Badalona (Barcelona)	Thanatos
Juan José Saura Paredes	Elche (Alicante)	Interceptor Cobalt

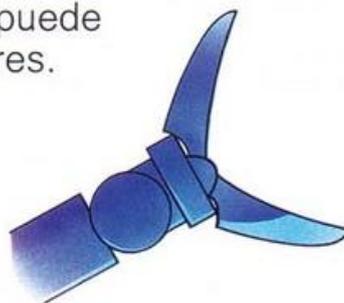
!PARTICIPA EN EL CONCURSO!



En INPUT estamos convencidos de que aún puedes hacer muchas más cosas con tu ordenador. Sin duda, muchos lectores estareis utilizando vuestro micro para funciones de lo más variadas, en unos casos; pintorescas, en otros; mientras que algunos listillos habrán podido utilizarlo para resolver tareas complejas. Es lógico, modificando programas y variando los periféricos nuestro ordenador puede prestar sus servicios en infinidad de facetas. INPUT quiere que esas aplicaciones y utilidades a las que has conseguido dedicar tu ordenador, sean conocidas por todos sus lectores y por eso ha organizado el «Concurso de Aplicaciones y Utilidades», en el que puede participar cualquiera de nuestros lectores.



BASES



UTILIDADES Y APLICACIONES: Si tu ordenador controla la calefacción de tu casa, gobierna un robot, dirige un pequeño negocio, organiza la maqueta de tu tren eléctrico, o cualquier cosa interesante u original; envíanos información gráfica y listados de tus programas, grabados en un cassette, diskette o microdrive.

Todo ello habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvideis indicar claramente para qué ordenador está preparado el material, así como vuestro

nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante los próximos tres meses **SORTEAREMOS:**

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
en material microinformático a elegir por los afortunados.

¡No os desanimeis!, por muy simples o complejas que puedan parecer vuestras ideas, todas están revisadas con el máximo interés.

INPUT SINCLAIR
Aribau, 185. Planta 1.^a
08021 BARCELONA

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

EL ZOCO

Vendo procesador de textos especial para GP-50-S o compatibles. Permite la impresión de textos en 64 columnas sin reducción de caracteres ni modificaciones en el hardware. Informa:

Manuel Cagiao
Apartado 2144
Tel. (981) 78 29 52 (20 h.)
15080 La Coruña

Desearía contactar con usuarios relacionados con Spectrum para intercambio de ideas, programas, etc. Interesados dirigirse a:

Carlos Lozano Barceló
Braulio Lausín, 10, 5ºF
Tel. (976) 41 38 84
50008 Zaragoza

Cambiaría el siguiente juego «Desert Rats» original (está aún sin abrir) por alguno de los siguientes war-games: Waterloo, Austerlitz, Pacific War o East Front, o bien lo vendería por 800 ptas. Interesados llamar a:

Eduardo
Tel. (985) 36 90 28

Vendo Spectrum 48K, impresora Seikosha GP-50S, Joystick Quick Shot II con interface Kempston, todo ello con sus correspondientes cables, transformadores y manuales. También incluye más de 50 revistas, cintas de programas que incluyen más de 80 juegos y utilidades, un curso de BASIC compuesto por 8 libros con ejercicios de autocomposición. Precio razonable y discutible.

Robert Gamigó
Pje. St. Ramón Nonat, 3, 1º, 2º
Tel. (93) 249 49 85
08028 Barcelona

Spectrum, Vendo ordenador Spectrum Plus, joystick, interface, cables, manuales y 70 juegos entre ellos: Saboteur, The Dambusters, Rambo, Exploding Fist, etc...; todo ello por sólo 32.000 ptas. Llamar:

Tel. 759 58 34
Madrid

Vendo ordenador Spectrum Plus, 48K con sus respectivos cables, y su transformador, en perfecto estado. También vendo interface tipo Kempston y joystick. Todo esto y 25 programas como Tapper, West Bank, Yabadabadu, etc. por el precio de 35.000 ptas. Interesados llamar a:

Manolo
Tel. (91) 650 00 61 (tardes)

Vendo Spectrum 48K. Nuevo, poco uso, con todos los cables y libros, interruptor en el transformador, regalo algunas revistas y más de cien juegos y

programas. 24.000 ptas. Interesados llamar o escribir a:

Luciano López Fernández
General Vives Camino, 10, 6ºD
19004 Guadalajara

Compro Spectrum 48K a su justo precio. Debe incluir al menos los manuales. Todos serán contestados. Razón:

Gregorio Riquelme Artés
Pº Alfonso XIII, 54, 5ºB
Cartagena (Murcia)

Intercambio, vendo y compro últimas novedades para el Spectrum en software. Hago pantallas de presentación y rutinas en c/m muy baratas y de gran utilidad. Dirigirse a:

Javier Piquer
Xaloc, 30
Palamós (Gerona)

Cambio, compro y vendo todo tipo de programas para el Spectrum 48K. Interesados dirigirse a:

Félix López Jordán
Pza. Fuensanta, 2, 13ºB
30008 Murcia

Vendo ordenador personal de bolsillo Casio PB-300 con libro de instrucciones en inglés. Nuevo. Sólo por 17.000 ptas. Preferiblemente en Madrid. Dirigirse a:

Andrés
Tel. 871 03 52
Madrid

Compro impresora de tamaño folio, (paralelo ó RS232), barata, matricial a poder ser. Interesados escribir a: (indicando cualidades y precio)

José Antonio Suárez Márquez
Vázquez Varela 19 3ºB
Vigo 4 (Pontevedra)

Cambio software para el Spectrum 48K y 128K de todo tipo. Interesados escribir a:

Fernando Rodríguez Pardo
Renfe, 20, Bajo Izquierda
04006 Almería

Vendo ZX Plus, con cables, etc. e instrucciones en castellano, Interface Kempston. Todo por 32.000 ptas.

Raúl Nicolás del Puerto
Salvador Martínez, 38
Tel 217 25 04
28041 Madrid

Vendo cintas LOAD'N'RUN nº 1 y 2, valor 1.700 por 1.300. Cintas SOFT Spectrum nº 1 y 2, Videospectrum nº 4, Microhobby y Cassete nº 1, Software Spectrum nº 1, valor 1.500 por 1.200. Todo el lote 2.300, valor 3.200.

Alvaro
Tel. (91) 251 39 91
De 14,30 a 15,00 h. Entre semana

Vendo Cámara Nikon F24 Photomic, Flash Nikon SB-7, Trípode, Teleobjetivo Nikkor 5mm F/1.4 y más accesorios o cambio por SP-800 con interface Masterfase 4. Llamar a:

Juan
Tel. (91) 638 73 75
De 4 en adelante

Vendo interface programable para joystick (Indescomp) por 4.500 ptas. o lo cambio por Interface Multijoystick. También vendo Robot programable Verbot comprado en Diciembre del 1985 por 12.000 ptas.

Fernando Ródenas
Pere Martell, 9, 5º, 1º
Tel. 21 24 37
43001 Tarragona

Necesito urgentemente un copión para programas en turbo. Cambiaría o compraría. Mi dirección es:

José Manuel Sánchez López
Avda. de la Raza, 29, 4ºA
Tel. (955) 24 66 23
21002 Huelva

Compro juegos del C 64 como Comando, Rambo, Spy us Spy, Fight Night, Impossible Mission, Karateka, Winter Games, La Ley del Oeste, The Gonnies, Sir Fred, Outlaws, Skool Daze, International Basketball, Fighting Warrior y Robin of the Wood. Llamar a:

Francisco
Tel. (958) 26 58 81

Desearía intercambiar libros y programas con usuarios del ZX-Spectrum. Preferentemente utilidades, aunque juegos también.

Jose Mº Martínez Salas
Federico García Lorca, bloque 21 B B
Algeciras (Cádiz)

Deseo intercambiar ideas, trucos, programas, etc, para Spectrum 48K y Plus. Los interesados llamar o escribir a:

Andrés Herrero
Jacinto Benavente, 11 bajo
Tel. (96) 158 15 80
Paterna (Valencia)

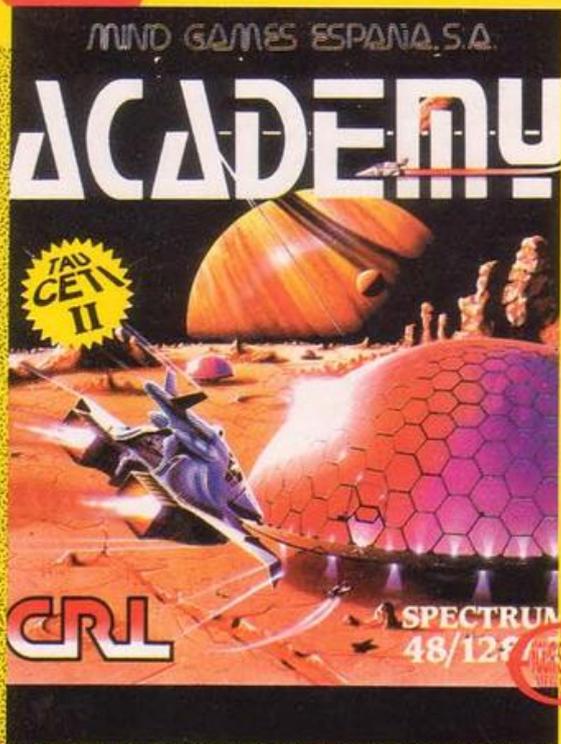
Vendo Spectrum Plus completo por 20.000 ptas.

Alberto
Tel. (93) 211 19 22 (noches)

Desearía intercambiar juegos para el Spectrum 48K, preferiblemente de Madrid. También desearía contactar con algún club del Spectrum de Madrid.

Mónica Monsó
Abtao, 25
Tel. 433 76 43
28007 Madrid

¡¡ LOS MEGA HITS!!

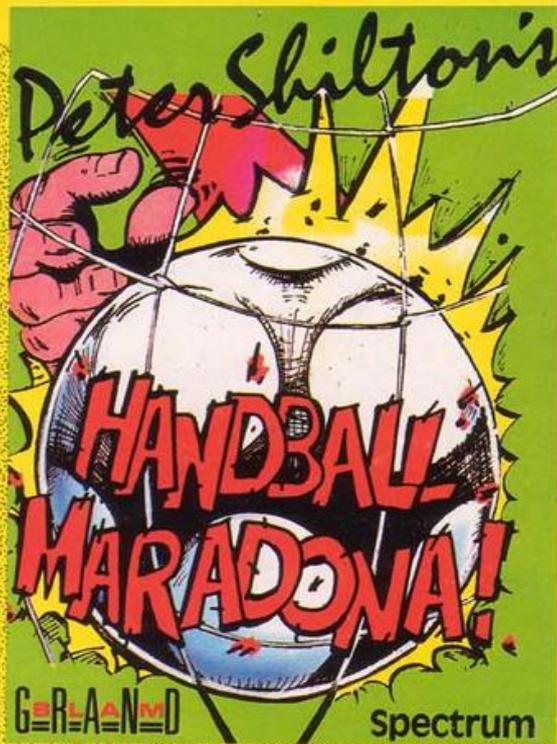


(S)

Si quieres ser piloto de Skimmer's tienes que ingresar en la «Academy» del Gal Corp con más de cien aspirantes anuales, sólo unos pocos cadetes consiguen la graduación. ¡Prepárate para matricularte en la «Academy»!
¡Tan Ceti te espera!

Disponibles en:

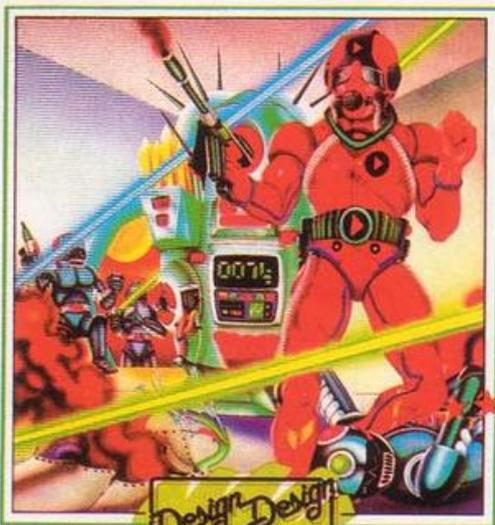
CASSETTE
Y
DISKETTE



(S)

¡Ayuda a Peter Shilton a cambiar el histórico resultado de México 86!!
Entrenamiento, partidos de Liga y desarrollo de la técnica futbolística.

MIND GAMES ESPAÑA S.A. (A) (S)
N·E·X·O·R

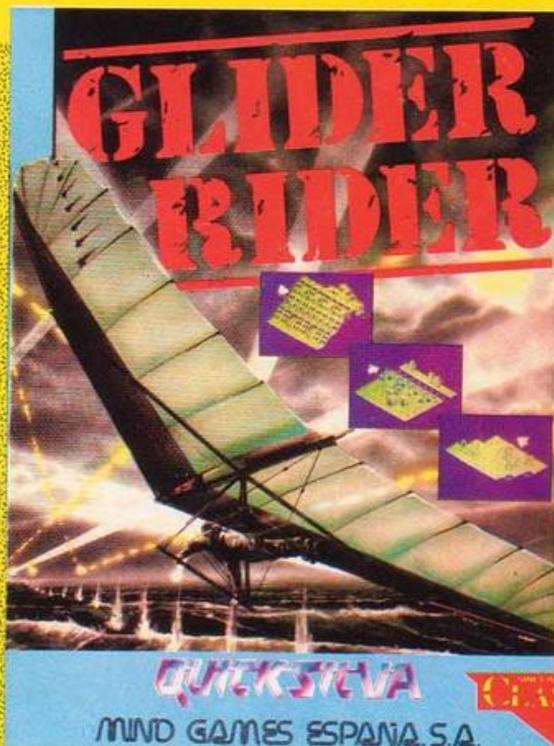


(A)

(S)

Sólo un hombre queda para proteger la última arma «Némesis» de las fuerzas de robots andromedanos. Tú tienes que escapar con el «Némesis» intacto o empezar la cuenta atrás hacia la autodestrucción. El tiempo acaba, cada segundo es importante.

VISITE LA DIVISION **Online** Galeries Precaladas
GALERIAS
Marcando estilo.



(A)

(C)

(S)

Esta es una real y verdadera misión diabólica. Como comandante Glenn White, has sido lanzado dentro de la isla artificial de Abraxas Corporación con sólo una moto, un ultraligero y nueve granadas. Tu misión, destruir los Reactores Nucleares de la isla... ¡suerte!

Editado y distribuido en España por:

MIND GAMES ESPAÑA S.A.
Mariano Cubi. 4 Entlo. Tel. 218 34 00 - 08006 Barcelona

AMSTRAD (A)
Comodore (C)
SPECTRUM (S)

A LA VENTA EN

EN TODOS LOS DISTRIBUIDORES DE NUESTROS PRODUCTOS

La Maquina alucinante



EL UNICO
ORDENADOR
CON MILES Y MILES
DE PROGRAMAS
DISPONIBLES.



Al comprar
tu nuevo Spectrum
pide el Pasaporte Fantástico.
Podrás conseguir
un reloj alucinante.

33.900 Ptas. + IVA



Microprocesador Z80A. 128 K RAM. 32 K ROM. Teclado de 58 teclas. 32 columnas x 24 filas de texto. Gráficos de alta resolución (256 x 192 pixels). 8 colores con dos niveles de brillo cada uno. Calculadora en pantalla. 3 canales de sonido programables e independientes. Cassette incorporada. Salida TV y monitor RGB.

Interface MIDI (Musical Instrument Digital Interface). Salida Serie RS 232 bidireccional. Dos conectores para joysticks. Conector plano compatible con todos los modelos Spectrum anteriores. Editor de pantalla y dos versiones BASIC en ROM. 48 K BASIC, compatible con Spectrum 16 K, 48 K y ZX+. 128 K BASIC, compatible con ZX Spectrum 128.

Nuevo **sinclair** ZX Spectrum +2

C/ Aravaca, 22. 28040 Madrid. Tel. 459 30 07. Telex 47660 INSC E. Fax 459 22 92. Delegacion en Cataluña: C/ Tarragona, 110. Tel. 325 10 58. 08015 Barcelona.