

# INPUT

Publicación práctica  
para usuarios de

# sinclair

Revista mensual 1986

Precio 350 Ptas

Año 1 Número 9



**COMO FUNCIONA  
UN MONITOR**

**CONTROL  
MULTITECLA**

**UN PROGRAMA  
QUE APRENDE**





# AHORA TE PUEDES PROGRAMAR UN VIAJE ALUCINANTE AL EPCOT CENTER CON ALEA. ES LOGICO.

Te presentamos, en estreno mundial, todo un reto a tu inteligencia: la colección de programas Logicolor.

Con los tres juegos de la colección Logicolor tu mente desafiará a la fría lógica del ordenador.

**AUTOS LOCOS:** Construye tu propia escudería y apuesta por tu bolido favorito. Un primer contacto con el uso de los símbolos. Para chicos entre 10 y 12 años. Incluye también un super-master mind contra el ordenador.

**MANZANAS Y GUSANOS:** Utilizando fórmulas puedes recoger las manzanas y dejar fuera los gusanos; proteger las ánforas de los golpes del martillo; defender los globos aerostáticos de las flechas enemigas; o evitar que los cañones destruyan las torres de tu fortaleza. Tu inteligencia lógica es la única arma que necesitas. Para chicos entre 12 y 14 años, y para quienes desean mantener su mente en forma.

**REHENES:** Tendrás que desarrollar una estrategia lógica si quieres eliminar a los conspiradores y salvar la corona. ¿Te gustaría descubrir la fórmula que abre el cofre de los diamantes? Intenta descubrir un procedimiento lógico para rescatar a los rehenes. Para chicos entre 14 y 16 años, y para quienes se las dan de genios.

Además, la compra de cada programa de la colección Logicolor te da derecho a participar en el fabuloso concurso EPCOT, y si consigues vencer al ordenador, tus posibilidades de conseguir un magnífico premio se duplican.

Si resultas ganador puedes elegir uno de estos SUPERPREMIOS:

- 1) UN FANTASTICO VIAJE PARA DOS PERSONAS DE 9 DIAS AL EPCOT CENTER; visitarás Marineland, el Museo Aeroespacial de la Nasa, Disneyworld, el Epcot Center y otros muchos lugares.
- 2) UN SUPERORDENADOR IBM-PC portátil.
- 3) UNA PAGA MENSUAL DE 30.000 Pts. durante un año para ti solo.

Encontrarás las Bases para participar en el concurso, junto con las fichas, en cada programa. Envíalas a ALEA antes del 21 de Julio de 1986.



¡Atención!, si envías tus fichas antes del 23 de Junio, tus posibilidades de ganar son aún mayores.

Alea también ha pensado en los más "peques"; tus hermanos de 4 a 9 años. Para ellos tenemos una serie de juegos que les ayudarán al aprendizaje de la escritura y la lectura. Comprando cualquiera de ellos, participarás automáticamente en el concurso LEXA, pudiendo llegar a conseguir una beca de estudios de hasta 500.000 Pts.

Puedes pedir tus programas llamando al teléfono de Madrid (91) 446 57 64 o bien enviándonos el cupón que hay al pie de esta página. También encontrarás los programas de la colección Logicolor en la microtienda de tu barrio, El Corte Inglés y Galerías Preciados.

Animo, por sólo 3.875 Pts. obtienes un magnífico programa y ¡hasta cuatro participaciones para el gran concurso EPCOT!

¡NO LO DEJES ESCAPAR! Prográmate ahora mismo un premio alucinante.

## álea

apartado de correos 10.048 de Madrid

GRUPO SOFT

COMMODORE 64/128 MSX/64K SPECTRUM 48/PLUS AMSTRAD 464/664/6128

Deseo adquirir los siguientes programas de su biblioteca, al precio de 3.875 Pts. cada uno, más 465 Pts. IVA.

Con la compra de los programas adquiero el derecho a participar en el concurso EPCOT y/o LEXA.

Forma de pago:

- ☐ Contrareembolso  
☐ Con cheque adjunto a nombre de Alea, S. A.  
☐ Con cargo a la tarjeta de crédito:  
☐ Visa  
☐ American Express  
☐ Diners Club

Nombre del titular: \_\_\_\_\_

Nº de la tarjeta: \_\_\_\_\_

Válida desde \_\_\_\_/\_\_\_\_/\_\_\_\_ hasta \_\_\_\_/\_\_\_\_/\_\_\_\_

Firma del titular: \_\_\_\_\_

Nombre: \_\_\_\_\_

Dirección: \_\_\_\_\_

Tel.: \_\_\_\_\_

COLECCION LEXA

EL DUENDE  
 EL TESORO  
 EL TORREON  
 EL OASIS

COLECCION LOGICOLOR

AUTOS LOCOS  
 MANZANAS Y GUSANOS  
 REHENES

INPUT SINCLAIR





**AÑO 1 NUMERO 9**

**DIRECTOR:**

Alejandro Diges

**COORDINADOR EDITORIAL:**

Francisco de Molina

**DISEÑO GRAFICO:**

Tomás López

**COLABORADORES:**

Antonio Taratíel, Luis R. Palencia,  
Francisco Tórtola, Benito Román,  
Esther de la Cal, Ernesto del Valle,  
Equipo Molisoft.

INPUT Sinclair es una publicación juvenil de  
EDICIONES FORUM

**GERENTE DIVISION DE REVISTAS:**

Angel Sabat

**PUBLICIDAD:** José Real-Grupo Jota

Madrid: c/ General Varela, 35

Teléf. 270 47 02/03

Barcelona: Avda. de Sarnia, 11-13, 1º

Teléf. 250 23 99

**FOTOMECANICA:** Ochoa, S. A.

**COMPOSICION:** EFCA, S. A.

**IMPRESION:** Edime, S. A.

Deposito legal: M. 27.884-1985

**SUSCRIPCIONES:** EDISA

López de Hoyos, 141. 28002 Madrid

Teléf. (91) 415 97 12

**REDACCION:**

Alberto Alcocer, 46, 4º

28016 Madrid. Teléf. 250 10 00

**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 el ira incluida la sobretasa aérea.

**Se ha solicitado el control OJD**

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 paginas.

Copyright ilustraciones del fondo gráfico de Marshall  
Cavendish, págs. 16, 17, 21, 22, 24, 25, 26, 27, 31, 32,  
33, 34, 35, 36, 37, 38, 40, 43, 46, 47.

# INPUT sinclair

## SUMARIO

EDITORIAL	4
ACTUALIDAD	6
APLICACIONES	
<b>MUSICA COMPRIMIDA</b>	<b>12</b>
PROGRAMACION	
<b>CONTROL MULTITECLA</b>	<b>16</b>
CODIGO MAQUINA	
<b>ENSAMBLADO A MANO (II)</b>	<b>22</b>
REVISTA DE HARDWARE	
<b>¿TELEVISOR O MONITOR?</b>	<b>40</b>
INTELIGENCIA ARTIFICIAL	
<b>UN PROGRAMA QUE APRENDE</b>	<b>52</b>
REVISTA DE SOFTWARE	
	<b>57</b>
EL ZOCO	
	<b>66</b>
PROGRAMACION DE JUEGOS (COLECCIONABLE)	
<b>PROGRAMA PARA JOYSTICKS</b>	<b>31</b>
<b>A LA CAZA DEL PATO</b>	
<b>BARAJA Y REPORTE</b>	



# UN MUNDO MUY MOVIDO

Aunque eran de sobra conocidas las dificultades financieras por las que venía pasando durante el último año **Sinclair Research**, a todos sorprendió la noticia de la compra de ésta firma por parte de su más directo rival: **Amstrad**. Se da la paradoja de que además la firma fue adquirida a precio de saldo, sin contar las deudas contraídas con los proveedores.

**Alan Sugar** es el artífice del *boom* de **Amstrad**, un hombre conocedor de los mercados internacionales de componentes, en los que se mueve ágilmente desde que lanzase sus primeros equipos de alta fidelidad. Ambos personajes tiene algo más en común que ser británicos, comparten la idea fija de poner los avances de la tecnología electrónica al alcance de cualquier consumidor.

Sin embargo, mientras que *sir Clive* tiene vocación investigadora, **Sugar** es un experto en el control de compras y *stocks*, factor este último que llevo a **Sinclair** a la bancarrota, al acumularse una gran cantidad de equipos invendidos en sus almacenes. Es curioso observar como la compañía que no ha mucho estuvo valorada en 136 millones de libras esterlinas, haya sido comprada por tan sólo 5.

Por otro lado ambas firmas parece que disponen

de modelos muy avanzados de máquinas dispuestas a enfrentarse en las tiendas con los **Amiga**, de **Commodore**, y **520 ST**, de **Atari** e incluso los atractivos **MSX II**. Este tipo de competencia tan directa promete ser estimulante para el mercado y la producción de nuevo *software*. Pero sin duda los más beneficiados deberán ser los usuarios que accedan a la nueva generación de microordenadores con elevadas prestaciones.

Algunos de los que se incorporan a nuestro inmejorable conjunto de lectores, escriben a la redacción para expresarnos su desconcierto por la paginación del artículo que ocupa las ocho páginas centrales de **INPUT**. Piensan que su ejemplar está mal encuadrado o que nos hemos confundido al hacer la revista. Nada más lejos de la verdad para su tranquilidad. Se trata de un curso de aprendizaje del BASIC —código máquina en el futuro— a través de un coleccionable, que explica el modo de desarrollar juegos propios. Dentro de algunos números lanzaremos unas tapas que permitirán que este coleccionable se transforme en un libro y, por tanto, cada capítulo no tiene porqué ocupar las ocho páginas, pudiendo ser más o menos.

## 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.

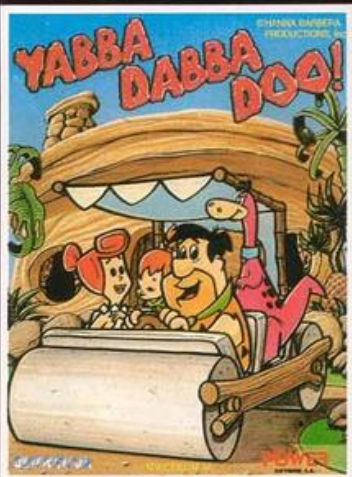
Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.º B. 28016 Madrid

### ELIGE TUS PROGRAMAS

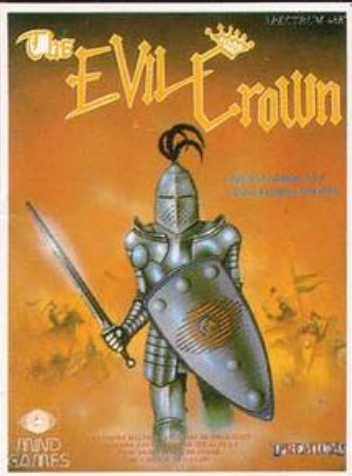
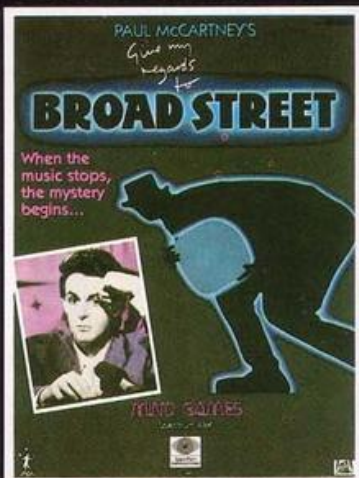
Primer título elegido	Segundo título elegido
Tercer título elegido	Programa que te gustaría conseguir
Qué ordenador tienes	Nombre
1.º Apellido	2.º Apellido
Fecha de nacimiento	Teléfono
Dirección	Localidad
Provincia	



## JUEGOS INTELIGENTES PARA MENTES INTELIGENTES



NOVEDAD

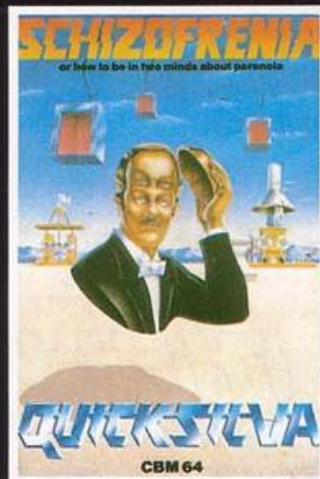


### COMMODORE

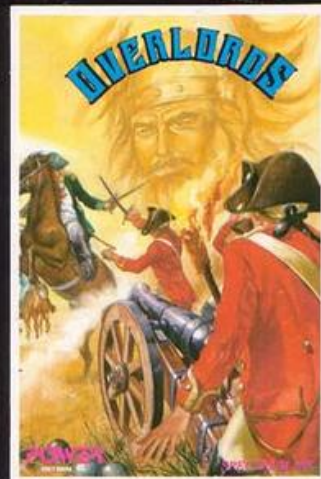
BROAD STREET	Cas. 2.600 - Disc. 3.600 P.V.P.
EVIL CROWN	Cas. 2.600 - Disc. 3.600 P.V.P.
TREASURE ISLAND	Cas. 2.600 - Disc. 3.600 P.V.P.
YABBA DABBA DOO	Cas. 1.800 - Disc. 3.100 P.V.P.
SCHIZOFRENIA	Cas. 2.600 - Disc. 3.600 P.V.P.
ANT ATTACK	Cas. 1.700 - Disc. 2.800 P.V.P.
FALL OF ROME	Cas. 1.800 - Disc. 2.800 P.V.P.
SEE SAW	Cas. 1.600 P.V.P.
TRASHMAN	Cas. 1.600 P.V.P.

### SPECTRUM

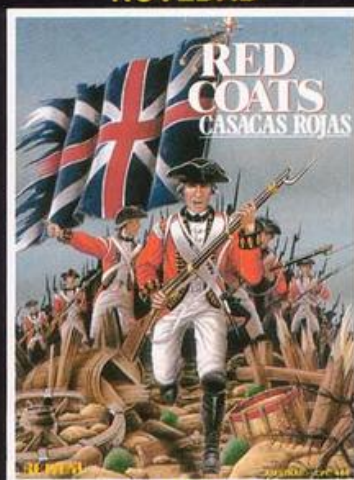
BROAD STREET	2.400 P.V.P.
EVIL CROWN	2.400 P.V.P.
TREASURE ISLAND	2.400 P.V.P.
YABBA DABBA DOO	1.800 P.V.P.
RED COATS	1.800 P.V.P.
SCHIZOFRENIA	2.400 P.V.P.
OVERLORD	1.800 P.V.P.
ANT-ATTACK	1.700 P.V.P.
FALL OF ROME	1.800 P.V.P.
GATECRASHER	1.400 P.V.P.



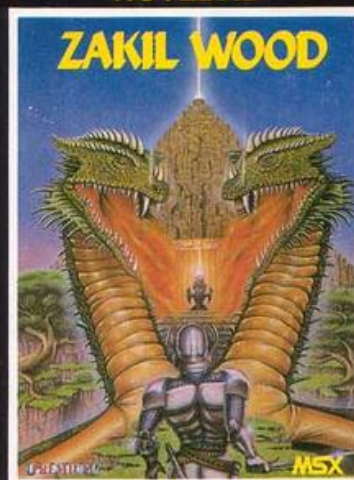
NOVEDAD



NOVEDAD



NOVEDAD



NOVEDAD

### AMSTRAD

YABBA DABBA DOO	Cas. 1.800 - Disc. 3.100 P.V.P.
RED COATS	Cas. 2.100 - Disc. 3.100 P.V.P.

### M.S.X.

ZAKIL WOOD	Cas. 2.800 P.V.P.
GAMES DESIGNER	Cas. 2.400 P.V.P.

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos

Nombre \_\_\_\_\_

Dirección \_\_\_\_\_

Teléfono \_\_\_\_\_ Ciudad \_\_\_\_\_

Firma: \_\_\_\_\_

Título \_\_\_\_\_ Sistema \_\_\_\_\_ Precio \_\_\_\_\_

Título \_\_\_\_\_ Sistema \_\_\_\_\_ Precio \_\_\_\_\_

Título \_\_\_\_\_ Sistema \_\_\_\_\_ Precio \_\_\_\_\_

Título \_\_\_\_\_ Sistema \_\_\_\_\_ Precio \_\_\_\_\_

☐ Contrarrembolso. ☐ Adjunto Talón. ☐ Giro Postal. **INPUT 5**

Deseo recibir información de sus programas en: MSX ☐ AMSTRAD ☐



## NUEVO DOMICILIO

**D**inamic Software, joven empresa nacional que desarrolla programas propios, se acaba de cambiar de casa. Al mismo tiempo han cambiado de nombre, pasando a llamarse Microdigital Soft, S.A.. Es por ello que Dinamic Software y Future Stars (muy sajón, verdad?) pasan a ser marcas registradas suyas. Para quienes interese, las nuevas señas son: Plaza de España, 18. Torre de Madrid, 29. Ofi. 11. 28013 MADRID

## LOS 10 HITS EN UN PAQUETE

**G**allup es un prestigioso y conocido instituto que, entre otras cosas, se dedica a esclarecer las cifras reales que en el mundo se barajan. La alianza Gallup/Microscope elabora desde hace algún tiempo una serie de tablas, que recogen el volumen de ventas y otras informaciones contrastadas con las obtenidas a partir de los profesionales- e incluso las propias revistas- de los programas existentes en el mercado británico. De esta

manera se pretende evitar las especulaciones. Dichos resultados son regularmente publicados por diversas revistas británicas también. Ahora, ABC Soft tiene intención de poner a la venta mensualmente la recopilación de los diez programas con mayor éxito de ventas, según las evoluciones semanales del mencionado ranking. La selección es realizada por la firma Beau-Jolly.

## SINCLAIR YA NO ES DE SINCLAIR

**B**uena es la que ha armado sir Clive. Un tranquilo día de principios de abril una escueta nota llegaba a los teletipos de las agencias informativas. Aunque pudiera parecerlo, no era 28 de diciembre y sin embargo la noticia era clara: En el informativo de las diez de la mañana, la BBC anunciaba la compra de Sinclair Research por parte de su mas directo rival, Amstrad. En las negociaciones de la compra, se le ofreció a Sinclair la posibilidad de quedarse con parte de la empresa que el mismo fundara, pero lo rechazó alegando mayor interés por investigar en nuevos campos, tales como los telefonos y las televisiones de bolsillo. Aunque lo que sucederá en el futuro es aún confuso para los ajenos la empresa, todo parece indicar que la marca Sinclair seguirá existiendo, manteniendo sus productos tradicionales. Alan Sugar, el artifice de Amstrad, es un hombre que conoce como nadie campos como el marketing y el control de los stocks de componentes y equipos fabricados. Aquí es donde tropezó Sinclair, mas intuitivo con los movimientos del mercado.

## AMSTRAD SIGUE IMPACTANDO.

**I**mparable, Amstrad sigue ganado puestos, hasta tal punto que se estima puedan haberse vendido casi cincuenta mil unidades del CPC 6128, previéndose un fuerte incremento en base al menor P.V.P., unas ochenta y cinco mil pts. en la versión con fósforo verde y cerca de ciento veinte mil si es de color, IVA aparte, por supuesto. Otra baza que parece estar jugando este fabricante es la del sistema operativo CP/M, de Digital Research, que aún no siendo precisamente de nuevo cuño, si dispone de una gran biblioteca de programas. Este mismo sistema está siendo utilizado por Commodore en el C-128, aunque ya existía en cartucho para el C-64, y se especula en torno a que Sinclair lo utilizará en breve (curiosa coincidencia con la reciente absorción de Sinclair). Una de las dificultades que surgen con el CP/M es la falta de compatibilidad con el formato del diskette. A este respecto, el CPC 6128 dispone ya de programas profesionales tan populares como el Multiplan (potente hoja de trabajo), el SuperCalc II (otra popular hoja), el dBase II (una de las bases de datos más extendidas), etc..

## MAS CONFORT EN LA PANTALLA

**E**ssilor es una importante firma francesa de la industria óptica. Durante algún tiempo sus técnicos han estado desarrollando lo que han dado en definir como "una lente antichoque, dos veces mas ligera que la lente mineral". Este producto comercial, Essilor Soft, ha sido desarrollado para proteger la vista de quienes trabajan habitualmente con monitores de video.

## CURSO DE BASIC

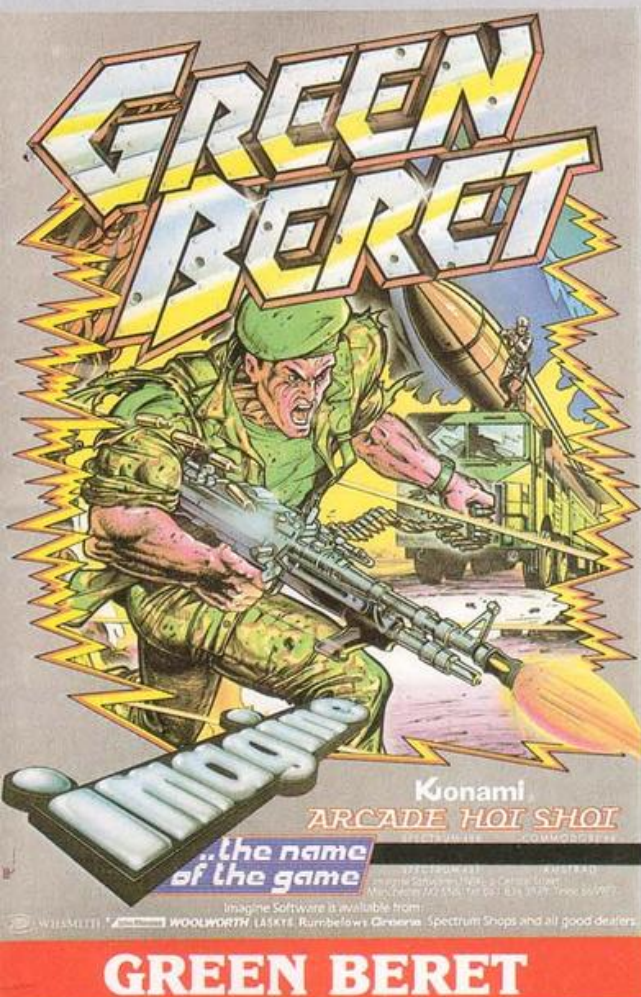
**C**eac es un centro de enseñanza multidisciplinaria a distancia. Uno de sus productos mas recientes es un curso de lenguaje BASIC con profusión de ejemplos, que comienza desde el nivel mas bajo. Por otro lado, los cursos han sido escritos en versiones personalizadas para Commodore y Spectrum. Cada lección incorpora un cuestionario de evaluación, que es corregido en la sede del centro.



# FREE dades

"V"

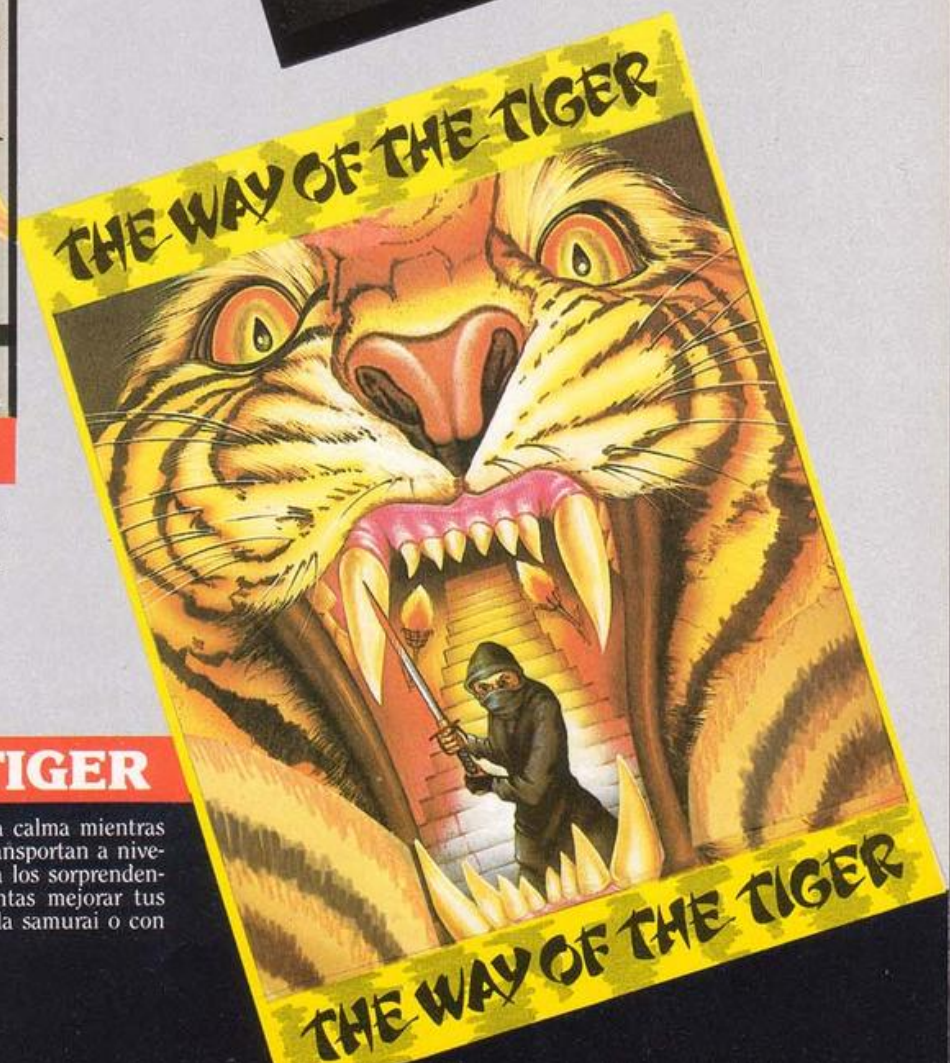
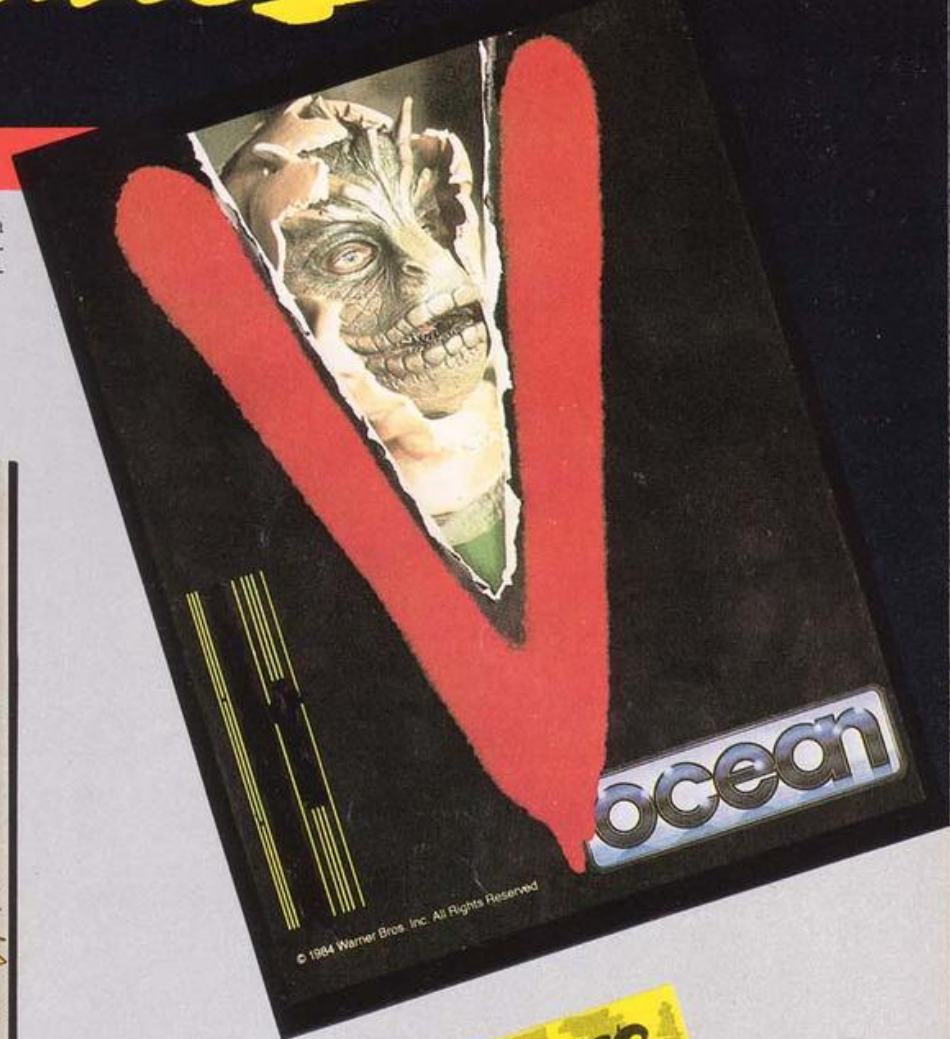
La serie de ciencia-ficción televisiva invade tu ordenador. La Tierra es visitada por alienígenas de aspecto humano. Si embargo, estos reptiles pronto revelarán su objetivo final... conquistar la Tierra y esclavizar a la raza humana.



Green Beret (boina verde), es el programa de mayor acción que hayas visto jamás en un ordenador. Decir Green Beret es decir un hombre perfectamente entrenado para la lucha. Prepárate...

## THE WAY OF THE TIGER

Entra en el mudo de los samurais. Mantén la calma mientras el movimiento y las rutinas de combate te transportan a niveles que nunca pensaste posibles. Experimenta los sorprendentes efectos del "Triple Scroll" mientras intentas mejorar tus técnicas de lucha cuerpo a cuerpo, con espada samurai o con mil posibilidades más.

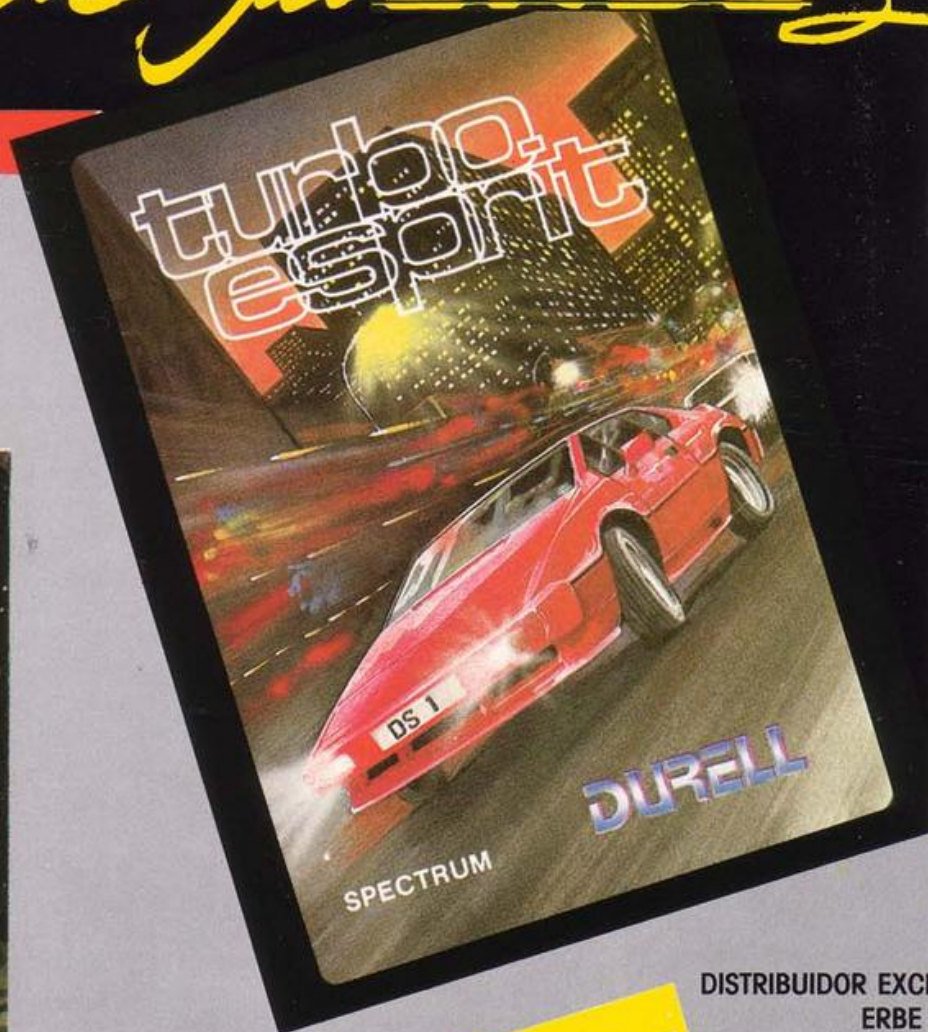




# Para que juegues ERBE

## TURBO ESPRIT

Tu misión: vigilar y cuidar el cumplimiento de la ley que se ve amenazada por una terrible banda de delincuentes que han hecho del tráfico de narcóticos su negocio más rentable. Tus medios: un Lotus Turbo Sprit dotado de uno de los máximos adelantos técnicos y con el que deberás patrullar por calles y avenidas.



Prepárate para una carrera sin límites a través del espacio con este nuevo juego de Ultimate. Los impresionantes gráficos y el inimitable estilo al que esta compañía nos tiene acostumbrados te transportarán a una auténtica aventura espacial a bordo del Cyberun.

## LA LEYENDA DE LAS AMAZONAS

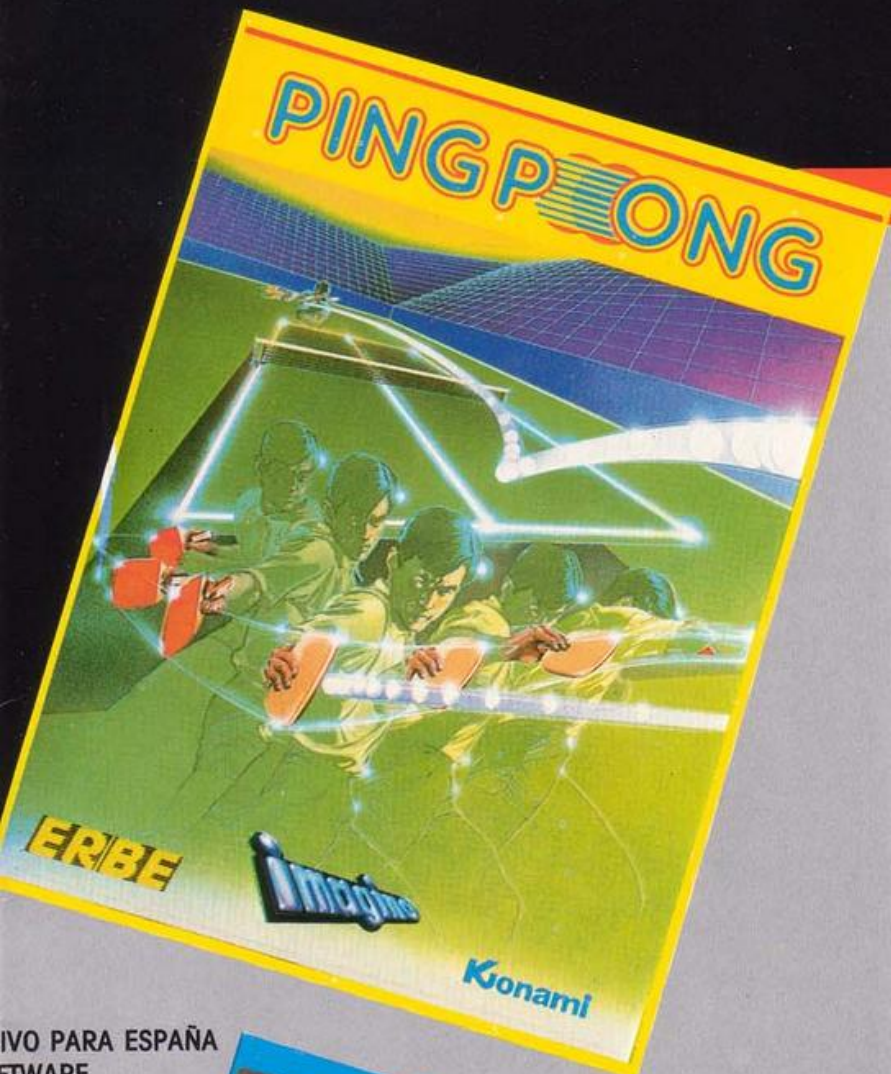
Un avión se estrella en la jungla. Sólo quedan 2 supervivientes: lady Wilde y su hija que es raptada por las Amazonas, una raza de mujeres míticas que quieren conservar su cultura a toda costa y que no admiten la presencia de extraños en su territorio.



DISTRIBUIDOR EXCLUSIVO  
ERBE  
C/. STA. E.  
28010  
TFNO. (9)

DELEGACION  
AVDA. MIS  
TFNO. (93)





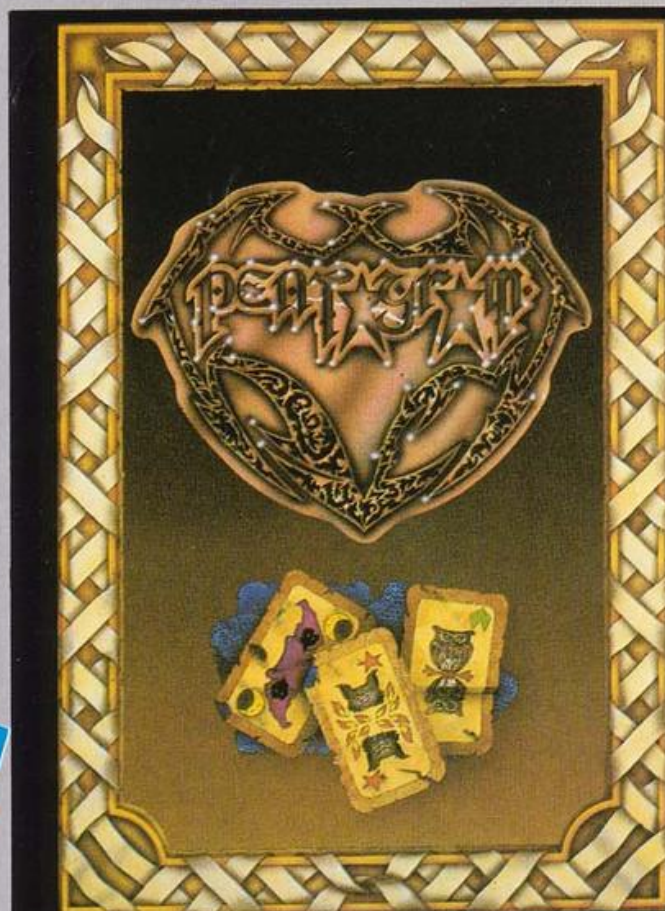
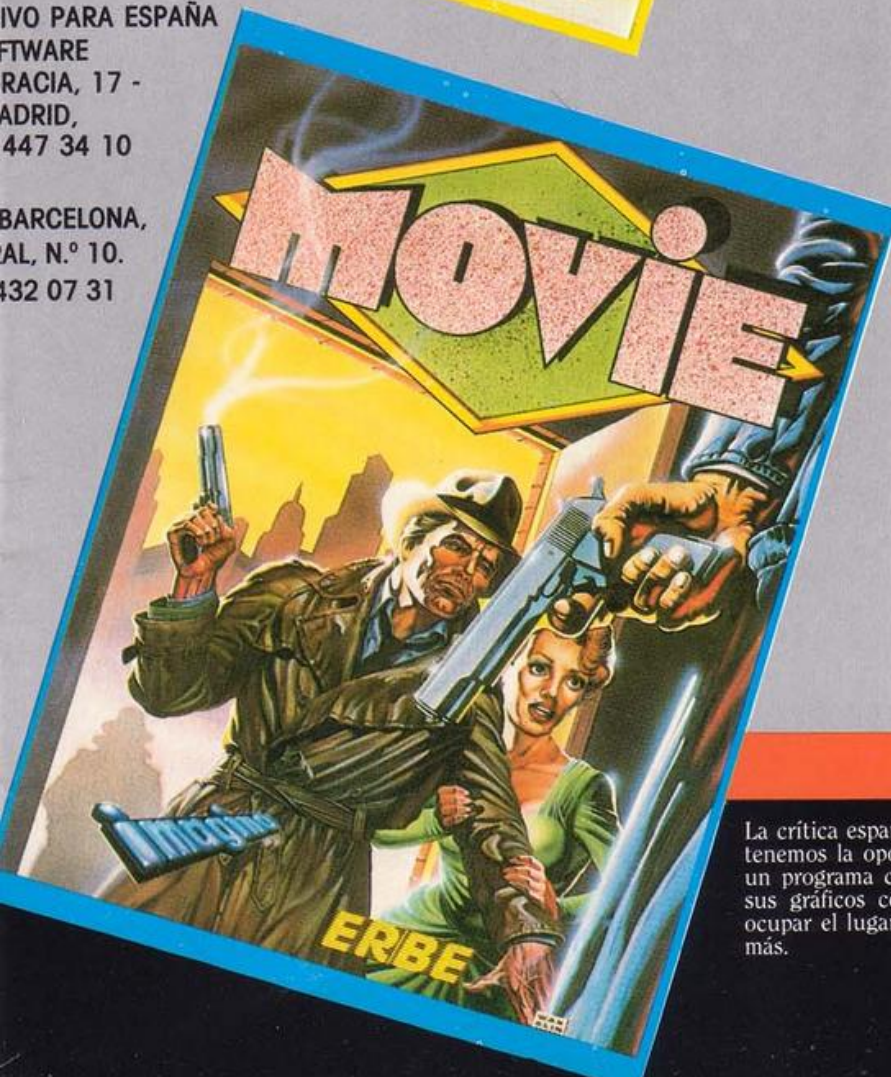
## PING-PONG

La gran sorpresa. Gráficos increíbles, movimiento super-rápido, podrás efectuar las mismas jugadas que si tuvieras la paleta en tu mano.

Botes, rebotes, efectos, dejadas, saques, cortadas, mates... todo es posible con esta maravilla llamada "Ping-Pong".

DISTRIBUIDOR PARA ESPAÑA  
SOFTWARE  
RACIA, 17 -  
MADRID,  
447 34 10

BARCELONA,  
AL, N.º 10.  
432 07 31



## PENTAGRAM

El juego sorpresa de Ultimate que debes descubrir tú mismo.  
... La continuación de "Saber Wulf".

## MOVIE

La crítica española ha dicho de este juego: "Todos los días no tenemos la oportunidad de disfrutar en nuestro ordenador de un programa como éste. "Movie" es un juego que, tanto por sus gráficos como por su gran originalidad, está llamado a ocupar el lugar de una auténtica estrella." No hace falta decir más.



## GANADOR DEL SORTEO

### ENCUESTA: "AYUDANOS A AYUDARTE"

Vuestra respuesta ha sido masiva y ya estamos procesando los datos para conocer con mayor precisión los gustos del lector medio de INPUT. Desde aquí os damos las gracias por vuestra inestimable colaboración.

Nos habría gustado daros un premio a todos. Sin embargo los invisibles dedos de la fortuna han sacado este nombre del montón de respuestas:

LUIS ABELARDO JUAREZ MARTINEZ  
de Madrid.

Quien recibirá el prometido  
SPECTRUM PLUS a lo largo de  
este mes. Enhorabuena.



# ORIENTATE

POR LOS EXITOS DEL MERCADO DEL REINO UNIDO

BUSCA, A PARTIR DEL 15 DE MAYO:



DE



SON LOS 10 PROGRAMAS PARA SPECTRUM, C64, C16 Y AMSTRAD MAS VENDIDOS CADA MES EN EL REINO UNIDO, SEGUN LAS ENCUESTAS DE **GALLUP**.

\* PIDENÓS LOS POSTERS MENSUALES CON LAS LISTAS A:

**abc analog, s. a.**

Santa Cruz de Marcenado, 31  
(3.º 13, 3.º 14 y 4.º 20)  
28015 MADRID

Tfnos: (91) 248 82 13-242 50 59  
Télex. 44561 BABC E

\* ENCUENTRALOS EN LA DIVISION **Online** DE GALERIAS

\* Y EN COMERCIOS ESPECIALIZADOS



Y a la venta en: EL CORTE INGLES -  
GALERIAS PRECIADOS - PRYCA -  
CONTINENTE - HYPERCORD y  
establecimientos especializados.

# MAX HEADROOM

Lo  
que  
yo  
quisiera  
saber  
es . . .

Disponible  
en - CBM 64  
SPECTRUM - 48/128 K  
y AMSTRAD.

Porqué  
alguien  
querrá  
jugar  
con  
algo  
diferente?

## QUICKSILVA

Los juegos más poderosos  
del universo

MIND GAMES ESPAÑA S.A  
MARIANO CUBI, 4 ENTLO.  
TEL. 218 34 00  
08006 BARCELONA

© Chrysalis Visual Programming Ltd.



Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.

Nombre \_\_\_\_\_  
Dirección \_\_\_\_\_  
Teléfono \_\_\_\_\_

Firma: \_\_\_\_\_

## MAX HEADROOM

SISTEMA \_\_\_\_\_ CANTIDAD \_\_\_\_\_

☐ Contrarreembolso. ☐ Adjunto Talón. ☐ Giro Postal.

Deseo recibir información de sus programas en: MSX ☐ AMSTRAD ☐



# MUSICA COMPRIMIDA EN EL SPECTRUM

■	MUSICA CON EL SPECTRUM
■	UN PROGRAMA COMPILADOR DE MUSICA
■	FUNCIONAMIENTO Y APLICACIONES

Quienes hayáis escuchado en alguna ocasión el *Commodore* o los *MSX*, comprenderéis enseguida por qué se dice que, en materia de sonido, nuestro *Spectrum* es la hermana fea de los microordenadores.

Aunque su diminuto zumbador y su «beep» de onda cuadrada no le den muchas posibilidades, eso no evita que una música bien realizada pueda dotar de una buena atmósfera a vuestros programas. En este artículo vamos a intentar facilitaros la tarea de hacer que quienes prueben vuestros juegos leviten quince centímetros sobre su silla.

Antes que nada, sería deseable disponer de un amplificador, o bien utilizar el equipo de alta fidelidad o el mismo *cassette* para este fin (a menos que querais aislar vuestra habitación y escuchar el exiguo sonido de la máquina con un buen estetoscopio).

Bromas aparte, el primer problema con que nos enfrentamos a la hora de introducir melodías en nuestros programas es la lentitud del intérprete BASIC, aparte de la gran cantidad de memoria que ocupan los datos. El programa que proponemos permite escribir melodías en una notación bastante más cómoda, y genera un hermoso bloque de código máquina (¿lo estabais esperando, eh?) que ocupa una tres veces menos memoria y cuya fluida ejecución os dará la posibilidad de conseguir efectos dignos de programas comerciales, pudiendo ser llamado desde cualquier programa vuestro.

Para componer música, no hace falta haber salido del Conservatorio; basta con una buena dosis de paciencia y, si tenéis, algún libro con nociones sencillas de solfeo. Aún sin conocimiento alguno, a base de experimentar con el programa podéis daros cuenta de lo sencillo que es meterle «marcha» al ordenador.

## MANEJO DEL PROGRAMA

Teclead el programa adjunto, poniendo el máximo cuidado al copiar los dos primeros DATAS, y salvadlo antes de lanzarlo por primera vez. (El código máquina es el peor enemigo de los sufridos tecladores; el que avisa no es traidor).

Una vez hecho esto, familiarizaos con el funcionamiento del programa utilizando la melodía que viene incluida. No tengáis miedo de experimentar; modificadla y observad los resultados.

A la hora de hacer vuestras propias composiciones, id escribiendo a partir de la línea que indica el programa. Las notas y sus duraciones se escriben en DATAS, separadas por comas: primero la duración y luego la nota. Los cuadros adjuntos os facilitarán el trabajo. Un consejo: utilizad líneas cortas para así facilitarlo aún más; corregir líneas largas es un calvario.

En la melodía podéis incluir la palabra «repe»: sirve para repetir un fragmento un número determinado de veces, de forma similar al «FOR NEXT» del BASIC. Se escribe de la forma:

Repe, N° veces, línea  
y se coloca al final del fragmento a repetir, colocando en el primer valor cuantas veces queréis que se ejecute, y en el segundo, el número de línea «DATA» donde comienza. El programa no es capaz de gestionar varios bucles «repe» simultáneamente, por lo que no se pueden anidar (poner uno dentro de otro). Con la ayuda de esta instrucción, podéis alternar varias notas muy rápidamente, consiguiendo una cierta sensación de polifonía. Así:

2000 DATA n/16,do,n/16,so,rep,8,2000  
hará el efecto de las notas «Do» y «Sol» mezcladas en una negra.

Al final de la melodía debéis teclear «fin,fin» para que el programa sepa que habéis terminado. (Es muy distraído y necesita que le repitan las cosas).

Una vez hecho esto, al lanzar el programa podréis escuchar vuestra creación en BASIC. Cuando hayáis quitado todos los errores, cambiad los parámetros a vuestro gusto; éstos son: velocidad en negras por minuto, y altura tonal de la melodía, en semitonos por encima (positivos) o por debajo (negativos) del «Do» central.

Cuando decidáis compilar en C.M., tened en cuenta que se ejecutará a una velocidad bastante mayor; cuando os pida una dirección para almacenarlo, dad un número alto (por ejemplo 50000) o la dirección que estiméis oportuna. Obtendréis la longitud de la tabla de notas, la longitud total del bloque generado, y la dirección a la que deberéis llamar mediante:

Let (una variable cualquiera) =  
USR (dirección de ejecución)

o bien:

RANDOMIZE USR (dirección de ejecución)

Al cargar el bloque desde cinta para utilizarlo de forma autónoma, protegédlo con:

CLEAR (dirección de comienzo 1)  
con el fin de que éste no sea invadido por el BASIC.

## FUNCIONAMIENTO DEL PROGRAMA

La misión prioritaria de este «compilador musical» es alojar en la dirección pedida por el usuario una rutina en código máquina de 55 bytes de longitud, seguida de una tabla de números que representan los valores de frecuencia y duración de cada nota. La rutina utiliza estos valores para llamar



a otra rutina perteneciente a la ROM: la subrutina BEEPER (dirección 03B5h), a la que se «alimenta» con la frecuencia en hertzios de la nota, y su duración en «T-estados». En el caso de tratarse de un silencio, se llama a la subrutina PAUSE (1F3Ah), residente también en ROM. Para más detalles, acudid al libro «THE COMPLETE SPECTRUM ROM DISASSEMBLY» de Melbourne House.

El bloque obtenido (código máquina y «ristra» de números) se compila formando un todo compacto, y está listo para ser ejecutado desde la dirección indicada. Los datos del bloque se almacenan en números de 2 bytes, contrariamente a la forma de «punto flotante» adoptada por el BASIC (5 bytes). (El desensamblaje de la rutina lo podéis ver en el cuadro adjunto).

Durante la ejecución de la melodía, la rutina se puede abortar pulsando **ENTER**. Por tanto, si la llamáis en comando directo, deberá ir precedida de PAUSE 0, iniciándose al pulsar cualquier otra tecla, con el fin de evitar que salgáis de ella nada más empezar.

Las aplicaciones de las rutinas generadas son infinitas: podéis generar varios fragmentos cortos y combinarlos con llamadas sucesivas desde BASIC para fabricaros un «bajo electrónico» con el Spectrum, o utilizar el programa como instrumento didáctico para aprender solfeo, o simplemente componer por placer, todo ello sin tener en cuenta lo que contribuirá un poco de música a hacer vuestros programas mucho más «calientes».

Que os las compongáis como podáis.

```

10 RESTORE : POKE 23609,25:
  BRIGHT 1: OVER 0: BORDER
  0: PAPER 0: INK 5: LET
  TI=60/100: LET A=0: CLS
15 PLOT 0,55: DRAW 255,0:
  FOR n=0 TO 1 STEP .1:
    PLOT 0,55-55^n: DRAW 255
    ,0: NEXT n
20 FOR n=0 TO 10: PLOT 47+16
  *n,54: DRAW (n-5)*8,-54:
  NEXT n
25 PLOT 31,54: DRAW -31,-32
  : PLOT 15,54: DRAW -15,-8

```



```

: PLOT 224,54: DRAW 31,-
32: PLOT 240,54: DRAW 15
,-8
30 INK 7: FOR n=0 TO 30: FOR
  m=1 TO 60-5*n: PLOT RND*
  255,56+n: NEXT m: NEXT n
35 GO SUB 910: LET X=0: LET
  Y=8: LET a$="* EL
  VIRTUOSO *": GO SUB 900:
  LET X=3: LET Y=3: LET a$=
  "1>ESCUCHAR MELODIA BASIC
  ": GO SUB 900: LET X=5:
  LET Y=3: LET a$="2>
  COMPILAR MELODIA EN C.M."
  : GO SUB 900: LET X=7:
  LET Y=3: LET a$="3>
  ESCRIBIR MELODIA": GO SUB
  900: LET X=9: LET Y=3:
  LET a$="4>MODIFICAR
  PARAMETROS": GO SUB 900
40 RESTORE : DIM z (12):
  FOR n=1 TO 12: READ v:
  LET z(n)=v: NEXT n
42 DATA 261.63,277.18,293.66
  ,311.13,329.63,349.23,369
  ,392.415,415.30,440,466.16
  ,493.88
45 GO SUB 905: PRINT AT 1+2+
  in,4: FLASH 1: OVER 1;" "
52 IF in=2 THEN GO SUB 910:
  LET x=0: LET y=2: LET a$=

```

```

"* COMPILAR MELODIA EN C.
M. *": GO SUB 900: GO SUB
95: LET X=9: LET Y=9: LET
a$="COMPILO? (S/N)": GO
SUB 900: PAUSE 0: IF
INKEY$="s" OR INKEY$="S"
THEN GO SUB 90
55 IF IN=1 THEN GO SUB 910:
LET X=0: LET Y=1: LET a$=
"* ESCUCHAR MELODIA EN
BASIC *": GO SUB 900: GO
SUB 95: LET X=9: LET Y=9:
LET a$="ESCUCHAR? (S/N)":
GO SUB 900: PAUSE 0: IF
INKEY$="s" OR INKEY$="S"
THEN LET comp=0: GO SUB
500
60 IF IN=3 THEN CLS : LIST
1000: STOP
65 IF IN=4 THEN GO SUB 910:
LET X=0: LET Y=4: LET a$=
"* MODIFICAR PARAMETROS *
": GO SUB 900: GO SUB 95:
LET X=10: LET Y=3: LET a$=
"INTRODUCE NUEVOS
PARAMETROS": GO SUB 900:
INPUT "NEGRAS POR MINUTO
? ";NEPM: LET TI=60/NEPM:
INPUT "SEMITONOS A
TRASPONER? ";A
70 GO TO 35

```





```

90 LET x=11: LET y=3: LET a$
  ="DIRECCION DE COMIENZO? "
  : GO SUB 900: INPUT
  "INTRODUCE VALOR: ";CDIR:
  LET COMP=1: GO SUB 910:
  LET X=0: LET Y=9: LET A$
  ="* COMPILANDO *": GO SUB
  900: LET X=3: LET Y=2:
  LET A$="DIRECCION DE
  COMIENZO: "+STR$ CDIR: GO
  SUB 900: LET X=5: LET Y=7
  : LET A$="ESPERA UN
  MOMENTO": GO SUB 900
91 RESTORE 200: FOR p=cdir+4
  TO cdir+54: READ da: POKE
  p,da: NEXT p: LET DIR=
  CDIR+6: LET POKE=CDIR+2:
  GO SUB 700: LET DIR=CDIR+
  9: LET POKE=CDIR: GO SUB
  700
92 LET DIR=CDIR+55: GO SUB
  500: LET LONG=DIR-CDIR-53
  : LET x=0: LET Y=3: GO
  SUB 910: LET a$="*
  COMPILACION TERMINADA *":
  GO SUB 900: LET x=3: LET
  a$="LONGITUD DE LA TABLA
  : "+STR$ LONG+" BYTES":
  GO SUB 900: LET X=5: LET
  A$="LONGITUD DEL BLOQUE:
  "+STR$ (DIR-CDIR)+" BYTES

```

```

": GO SUB 900: LET DIR=
  cdir+2: LET POKE=INT
  (LONG/4): GO SUB 700:
  LET DIR=CDIR: LET POKE=
  CDIR+55: GO SUB 700:
  RANDOMIZE USR (CDIR+4):
  LET X=7: LET A$="
  DIRECCION DE EJECUCION: "
  +STR$ (CDIR+4): LET Y=1:
  GO SUB 900: LET X=9: LET
  Y=8: LET A$="LO SALVO? (S
  /N)": GO SUB 900: PAUSE 0
  : IF INKEY$="S" OR INKEY$
  ="s" THEN SAVE "MUSICA"
  CODE CDIR, LONG+56: RETURN
94 RETURN
95 LET X=3: LET Y=3: LET A$=
  "PARAMETROS EN CURSO:":
  GO SUB 900: LET X=5: LET
  Y=3: LET A$="NEGRAS POR
  MINUTO: "+(STR$ (60/ti))
  : GO SUB 900: LET X=7:
  LET Y=3: LET A$="
  TRASPOSICION EN "+STR$ A
  +" SEMITONOS": GO SUB 900
  : RETURN
100 LET cu=8*ti: LET r=cu/2:
  LET b=r/2: LET n=b/2:
  LET c=n/2: LET sc=c/2:
  LET f=sc/2: LET sf=f/2:
  LET rp=r+b: LET bp=b+n:

```

```

LET np=n+c: LET cp=c+sc:
LET scp=sc+f: LET fp=f
+sf
102 LET REPE=988: LET do=a:
LET dob=a-1: LET dos=1+a
: LET reb=dos: LET re=2+
a: LET res=3+a: LET mib=
res: LET mi=4+a: LET mis
=5+a: LET fab=mi: LET
fas=6+a: LET fa=mis: LET
sob=fas: LET so=7+a: LET
sos=8+a: LET lab=sos:
LET la=9+a: LET las=10+a
: LET sib=las: LET si=11
+a: LET sis=12+a: LET o=
12: LET s=999: LET fin
=1000
103 RETURN
200 DATA 237,75,0,0,42,0,0,
197,62,191,219,254,230,
1,40,33,94,35,86,35,78,
35,70,35,229,197,225,
124,181,32,7,213,193,
205,61,31,24,3,205,181,
3,225,193,11,120,177,32,
215,201,193,201
500 GO SUB 100: RESTORE 1000
505 LET REP=0
510 READ D,T
515 IF T=S THEN GO TO 545+5
*COMP
520 IF D=FIN THEN RETURN
525 IF D=REPE THEN GO TO 555
530 GO TO 535+5*COMP
535 BEEP D,T: GO TO 510
540 GO SUB 600: GO TO 510
545 PAUSE D*50: GO TO 510
550 GO SUB 650: GO TO 510
555 READ T1: IF REP=0 THEN
LET REP=T
557 IF REP=1 THEN GO TO 505
560 RESTORE T1: LET REP=REP-
1: GO TO 510
600 BEEP d,t: IF t<0 THEN
LET ocen=0: LET ocde=INT
((11-t)/12): LET t=t+12*
ocde: GO TO 615
605 IF t>11 THEN LET ocde=0:
LET ocen=INT (t/12): LET
t=t-12*ocen: GO TO 615
610 IF f>=0 AND t<=11 THEN
LET ocde=0: LET ocen=0
615 LET freq=z(t+1)*(2^ocen)
/(2^ocde)
620 LET poke=INT (freq*d):

```



```

GO SUB 700: LET poke=INT (437500/freq-30.125): GO
SUB 700: RETURN
650 LET poke=d*50: GO SUB
700: POKE dir,o: POKE
dir,0: LET dir=dir+2:
RETURN
700 RANDOMIZE poke: POKE dir
,PEEK 23670: POKE dir+1,
PEEK 23671: LET dir=dir+
2: RETURN
900 FOR P=1 TO LEN a$: PRINT
AT x,y;"[CAPS SHIFT+9+
CAPS SHIFT+8+9]": FOR m=
1 TO 2: OUT 254,16: OUT
254,0: NEXT m: PRINT AT
x,y;a$(P): LET y=y+1:
NEXT P: LET y=0: BEEP .0
4,30: BEEP .02,30:
RETURN
905 PAUSE 0: LET i$=INKEY$:
BEEP .05,30: BEEP .025,
30: IF CODE i$>52 OR
CODE i$<49 THEN GO TO 905
906 LET in=VAL i$: RETURN
910 DIM b$(13*32): PRINT AT
0,0: PAPER 0:b$: RETURN
1000 REM INTRODUCE TUS DATAS
DESDE AQUI
1010 REM
1020 DATA n/16,la-o,n/16,la,
repe,8,1020
1030 DATA n/16,si-o,n/16,so,
repe,8,1030
1040 DATA n/16,do-o,n/16,dot
o,repe,8,1040
1050 DATA c/8,si-o,c/8,reto,
repe,4,1050
1060 DATA c/8,si-o,c/8,soto,
repe,4,1060
1070 DATA n/16,do,n/16,mito,
repe,8,1070
1080 DATA c/8,si-o,c/8,reto,
repe,4,1080
1090 DATA c/8,si-o,c/8,soto,
repe,4,1090
1100 DATA n/16,la-o,n/16,mit
o,repe,8,1100
1110 DATA n/16,la-o,n/16,dot
o,repe,8,1110
1120 DATA n/16,re-o,n/16,ret
o,repe,8,1120
1130 DATA n/16,fas-o,n/16,la
,repe,8,1130
1140 DATA n/16,so-o,n/16,so,
repe,16,1140
1150 DATA sc,doto,sc,so,sc,
mi,sc,so,repe,2,1150
1160 DATA sc,doto,sc,mito,sc
,reto,sc,doto
1170 DATA sc,reto,sc,so,sc,
fa,sc,so,repe,2,1170
1180 DATA sc,reto,sc,fato,sc
,mito,sc,reto,sc,mito,
sc,doto,sc,reto,sc,mito
,sc,fato,sc,soto,sc,la
to,sc,reto,c,soto,c,ret
o
1190 DATA sc/3,si,sc/3,doto,
sc/3,si,sc,si,scp,la,f,
so,n,so
1200 DATA fin,fin
9999 SAVE "ELVIRTUOSO" LINE
1: VERIFY ""

```

## NOMBRE

## NOTACION

Redonda	R
Blanca	B
Negra	N
Corchea	C
Semicorchea	SC
Fusa	F
Semifusa	SF
Puntillo *	P

\* El puntillo suma una figura a la mitad de su valor.

Ejemplo: SCP=Semicorchea + Fusa.

Cada figura dura la mitad de su precedente.

Se pueden sumar los valores con el signo "+". Ejemplo: B+SC=Blanca + Semicorchea.

Cuadro 1: Duración de las notas

## FRECUENCIA DE LAS NOTAS

Cada nota se escribe por su nombre, excepto SOL, que se escribe "S0". A las notas se les puede añadir:

# = Sostenido, escrito como S y sube un semitono.

b = Bemól, escrito como B y baja un semitono.

Añadir "+0" o "-0" suman o restan una octava respectivamente.

Ejemplo:

SIB-0-0 = SI Bemól bajado 2 octavas.

SOS+0 = SOL Sostenido subido 1 octava.

Cuadro 2: Frecuencia de las notas



# CONTROL MULTITECLA

■	COMO FUNCIONA TU TECLADO
■	DIFERENTES METODOS
■	IMPLEMENTACION DEL CONTROL DE JUEGOS
■	GENERACION DE MATRICES

Si quieres tener un control máximo en tus juegos, en vez de utilizar un joystick puedes programar el teclado para detectar más de una pulsación de tecla.

Una de las características más importantes de cualquier juego es el tipo de interacción existente entre el ordenador y el usuario. Para la mayoría de los fines el joystick ofrece un control más sofisticado, aunque probablemente es más versátil el teclado. Para saber qué tecla se ha pulsado, se utilizan normalmente la sentencia INKEY\$.

INKEY\$ es capaz de detectar casi todos los caracteres, pero su principal desventaja es que sólo puede detectar dos pulsaciones simultáneas cuando una de las teclas es **SHIFT** u otra tecla de control análoga, tal como **SYMBOL SHIFT**.

La detección de estas combinaciones es adecuada para casi todas las aplicaciones, pero surgen problemas cuando, por ejemplo, quieres controlar el movimiento vertical y horizontal, disparar el laser y arrojar bombas inteligentes sobre un avión. El conjunto de todas estas actuaciones requiere la pulsación simultánea de cuatro teclas. La mejor manera de resolver el problema, aunque de hecho puede que no siempre sea posible resolverlo, depende de cómo esté hecho el ordenador.

## EFFECTO DE LAS PULSACIONES DE TECLA

Para entender la manera de detectar más de una pulsación de tecla, es importante conocer cómo funciona el ordenador cuya tecla está siendo pulsada. Hay dos métodos muy extendidos entre los micros domésticos. En el primer método (*polling*) el ordenador explora secuencialmente el teclado,

recorriendo todas las teclas a intervalos regulares de tiempo para ver si se ha pulsado alguna.

En el segundo método la pulsación de tecla origina un mensaje llamado interrupción que se envía al procesador.

La interrupción hace que el ordenador deje de hacer por un tiempo lo que estaba haciendo para atender al teclado. Su tarea inmediata ahora es averiguar qué tecla ha sido pulsada. Este método es mucho más eficiente y versátil que el de la exploración secuencial, ya que si las teclas están inactivas no hay necesidad de prestarles atención, pero si se ha pulsado alguna tecla se le da la máxima prioridad a este hecho.

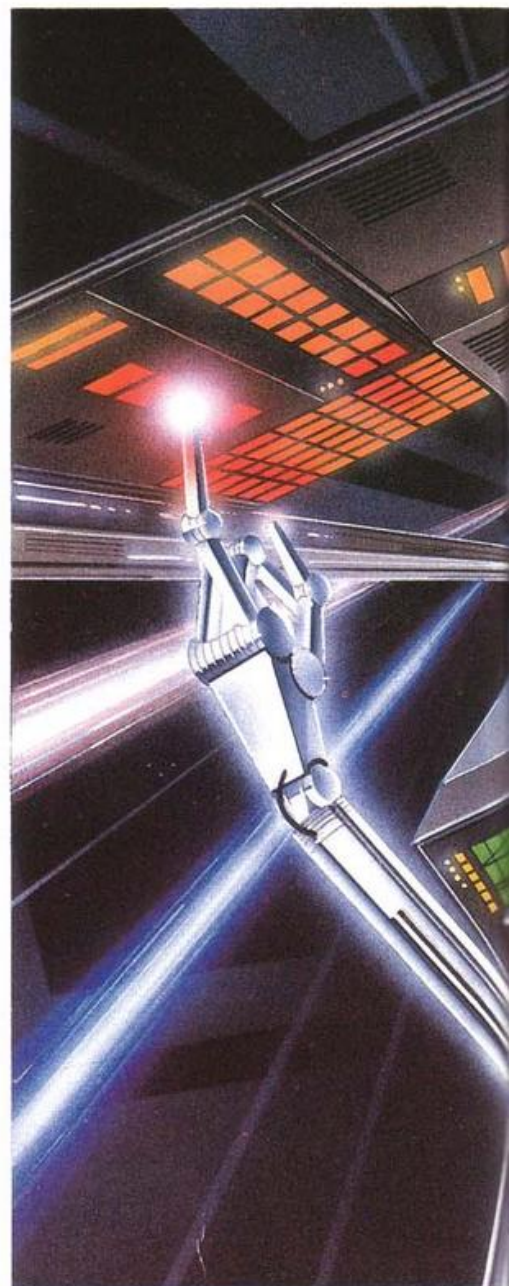
Sea cual sea el método para detectar las pulsaciones de tecla, el ordenador debe poder identificar cualquier tecla con rapidez y facilidad.

Casi todos los teclados utilizan un sistema llamado de generación matricial, en virtud del cual cada tecla genera un número que corresponde a su posición dentro de la matriz que forman las propias teclas. Lo que sucede cuando se genera este número depende en cada caso del diseño del ordenador.

El teclado del **Spectrum** está organizado en cuatro filas de diez teclas. Pero a efectos del ordenador puedes imaginar que cada fila está dividida en dos mitades, por lo que hay ocho grupos de cinco teclas. Cada grupo se comunica con una línea de entrada/salida o *port* de E/S. El teclado del **Spectrum** está especialmente adaptado para la detección de pulsaciones múltiples, ya que hay 65536 de estos *ports*, cada uno de ellos identificado por una dirección entre 0 y 65535.

En la tabla 1 puedes ver las direcciones de los *ports* correspondientes a cada uno de los ocho grupos en que se organiza el teclado.

GRUPO DE TECLAS	n	PORT
-----	-----	-----
CAPS/SHIFT ZXCV	0	65276
ASDFG	1	65022
QWERT	2	64510
12345	3	63486
67890	4	61438
YUIOP	5	57342
HJKL ENTER	6	49150
BNM SYMBOL/SHIFT		
SPACE	7	32766
-----	-----	-----





Para hallar las direcciones de cada grupo, utiliza la fórmula:  $254 + 256 * (255 - 2 \uparrow n)$ , siendo  $n$  el grupo de teclas listado en la tabla. Como alternativa puedes teclear y ejecutar este corto programa para calcular las direcciones:

```
10 INPUT "ESCRIBE EL NUMERO
DEL GRUPO DE TECLAS: ";n
20 PRINT "NUMERO DEL GRUPO
DE TECLAS ";n
30 PRINT "DIRECCION DEL PORT
";254+256*(255-2^n)
40 GO TO 10
```

Cada dirección de los *ports* de la tabla contiene un número que depende de la tecla que ha sido pulsada. Si tienes conectada una impresora, o cual-

quier otro tipo de *interface*, tienes que desconectarla para no tener resultados espúreos. Sustituye el primer programa por estas dos líneas, después ejecútalas y pulsa las teclas 1 a 5:

```
10 PRINT AT 0,0;IN 63486
20 GOTO 10
```

Observa cómo van cambiando los números de la pantalla cuando vas pulsando diferentes combinaciones de teclas.

El valor de la dirección del *port* está contenido en un byte cuyos bits octavo y sexto son siempre 1. El séptimo bit es normalmente un cero, pero se pone a 1 cuando hay una señal presente en el conector de auricular (EAR) o cuando el ordenador se ca-

lienta. Los bits 1 a 5 representan las teclas de cada grupo. Normalmente están a 1, pero cuando se pulsa una tecla, el bit correspondiente a dicha tecla se pone a 0. El quinto bit corresponde a la tecla más próxima al centro de la fila y el primer bit a la tecla del borde de la fila.

Cuando cada uno de los ocho bits de la fila está a 1, dicho bit contribuye al valor que hay en el *port*, pero no contribuye nada si es 0. El valor con el que contribuye cada bit es el siguiente:

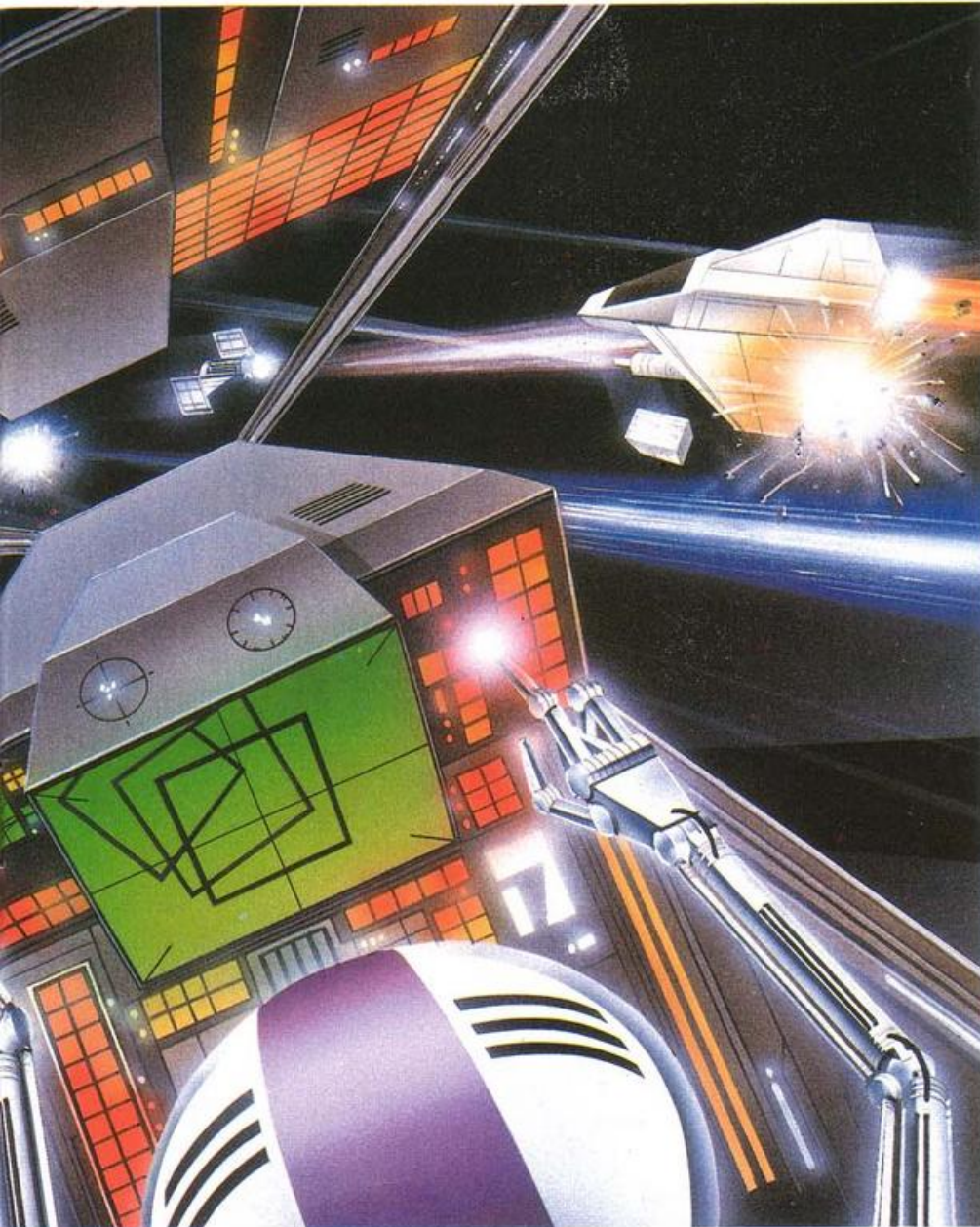
Bit	8	7	6	5	4	3	2	1
Val.	128	64	32	16	8	4	2	1

Así, si el séptimo bit es cero y se pulsa un cinco, la distribución de bits en el *port* 63486 es 10101111 y el correspondiente valor de este *port* será  $128 + 0 + 32 + 0 + 8 + 4 + 2 + 1 = 175$ . Con esto puedes detectar las pulsaciones simultáneas de hasta las cinco teclas del grupo. Esto se aplica también a los otros grupos de teclas; puedes añadirle líneas al programa anterior para experimentar con los valores de las direcciones de los *ports*.

La detección de, por ejemplo, seis pulsaciones de teclas simultáneas no es mucho más difícil que la detección de dos. Teclea ahora el siguiente programa para ver el grado de éxito que puedes conseguir:

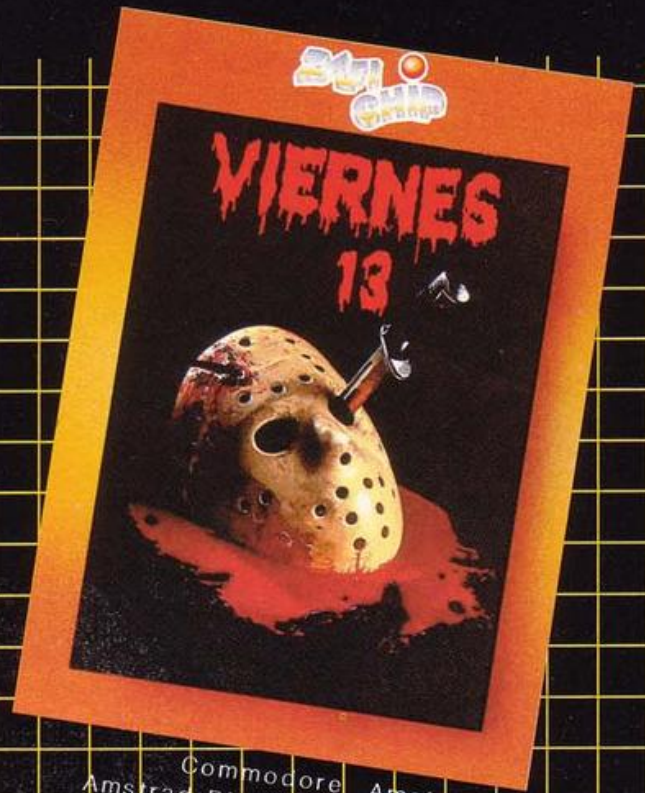
```
10 BORDER 0:PAPER 0:INK 7
20 CLS
30 LET y=11:LET X=15
40 LET p=63486
50 GO SUB 220
60 IF i=175 THEN GO SUB 290
70 LET p=61438
80 GO SUB 220
90 IF i=187 THEN GO SUB 380
100 IF i=175 THEN GO SUB 320
110 IF i=183 THEN GO SUB 350
120 IF i=189 THEN GO SUB 410
130 IF i=185 THEN GO SUB 380
:GO SUB 410
140 IF i=173 THEN GO SUB 320
:GO SUB 410
150 IF i=181 THEN GO SUB 350
:GO SUB 410
```

INPUT 17





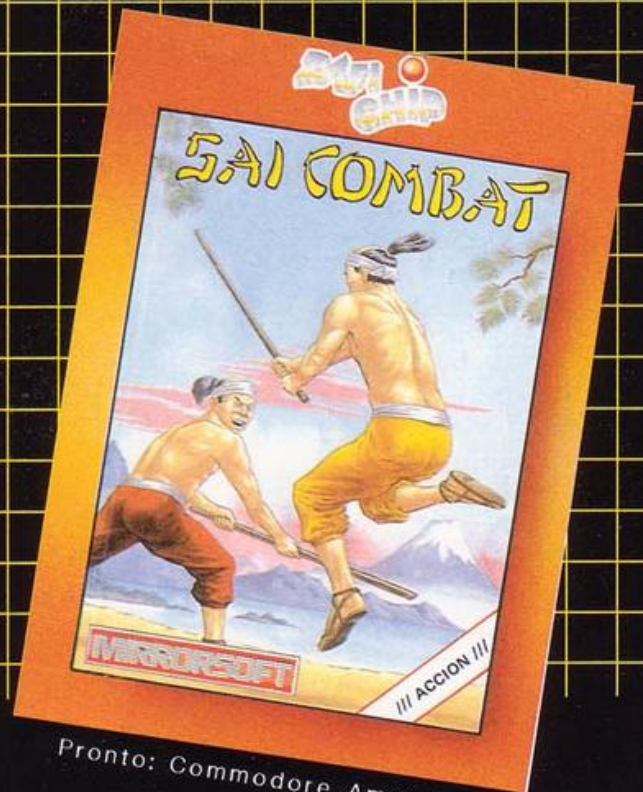
# Los Mejores Juegos



Commodore Amstrad  
Amstrad Disk



Spectrum Commodore



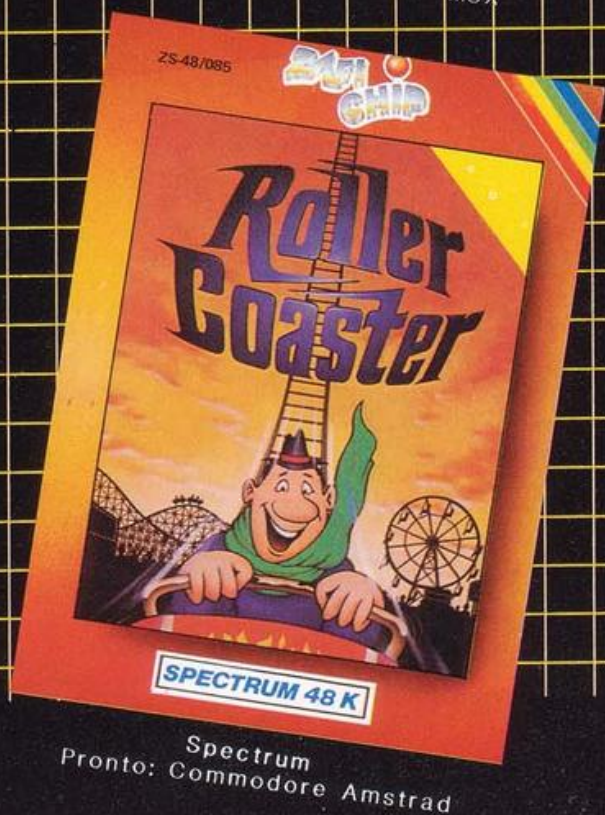
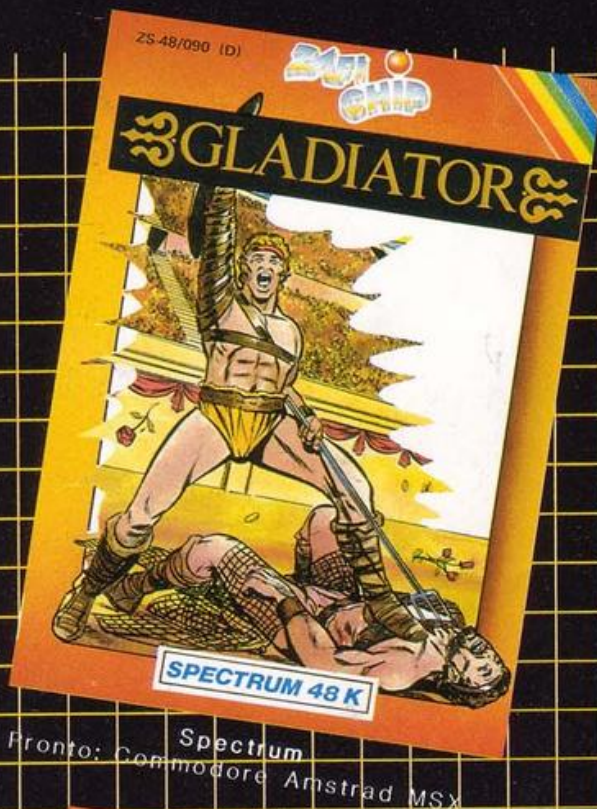
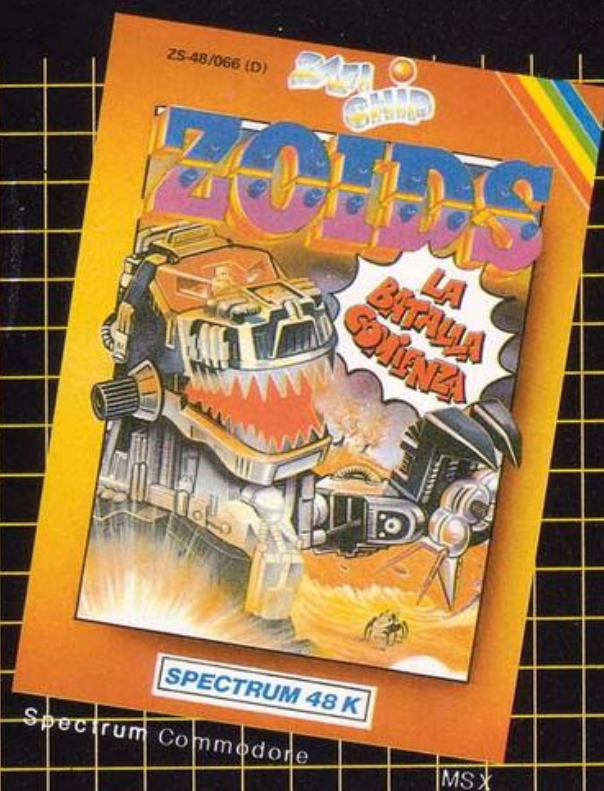
Pronto: Commodore Amstrad

Si deseas información y participar en los importantes sorteos que ZAFICHIP celebrará durante el año. . . ¡ESCRIBENOS!

Si están agotados en tu tienda habitual ¡¡LLAMANOS!!



# para esta Primavera



ZAFIRO SOFTWARE DIVISION  
 Paseo de la Castellana, 141. 28046 Madrid.  
 Tel. 459 30 04. Tel. Barna. 209 33 65. Telex: 22690 ZAFIR E  
 Programas editados, fabricados y distribuidos en España  
 con la garantía Zafiro. Todos los derechos reservados.





```

160 IF i=171 THEN GO SUB 380
    :GO SUB 320
170 IF i=179 THEN GO SUB 380
    :GO SUB 350
180 IF i=169 THEN GO SUB 380
    :GO SUB 320:GO SUB 410
190 IF i=177 THEN GO SUB 380
    :GO SUB 350:GO SUB 410
200 GO SUB 250
210 GO TO 40
220 LET i=IN p
230 IF i>191 THEN LET i=i-64
240 RETURN
250 PRINT AT y,x;" + "
260 PRINT AT y+1,x;
    "[3*ESPACIO]"
270 PRINT AT y-1,x;
    "[3*ESPACIO]"
280 RETURN
290 IF x<1 THEN RETURN
300 LET x=x-1
310 RETURN
320 IF y>19 THEN RETURN
330 LET y=y+1
340 RETURN
350 IF y<2 THEN RETURN
360 LET y=y-1
370 RETURN
380 IF x>28 THEN RETURN
390 LET x=x+1
400 RETURN
410 BEEP .004,20
420 BEEP .004,10
430 PRINT AT y,x;
    "[3*ESPACIO]"
440 RETURN

```

Pulsa las teclas para mover el blanco hacia arriba y hacia abajo, por ejemplo, y al mismo tiempo puedes disparar mientras estás moviendo el blanco.

En el siguiente programa se ha dado un paso más siguiendo este mismo

principio. Este programa te permite detectar seis pulsaciones de tecla para hacer posible que juegue un segundo jugador.

```

10 BORDER 0: PAPER 0 :INK 7
20 BRIGHT 0: OVER 0: CLS
30 PRINT AT 1,7; INK 6;
    FLASH 1;" P I S T O L E R
    0 "
40 PRINT:PRINT
50 PRINT INK 5;"CONSIGUE
    PUNTOS DISPARANDO A TU
    Oponente.CADA JUGADOR
    TIENE SEISBALAS."
60 PRINT:PRINT
70 PRINT TAB 7; INVERSE 1;
    "C O N T R O L E S: "
80 PRINT:PRINT"JUGADOR 1[14*
    ESPACIO]JUGADOR 2"
90 PRINT:PRINT"[5*ESPACIO]1
    [5*ESPACIO]---ARRIBA--
    [5*ESPACIO]0"
100 PRINT:PRINT"[5*ESPACIO]Q
    [5*ESPACIO]---ABAJO---
    [5*ESPACIO]P[4*ESPACIO]"
110 PRINT:PRINT"[5*ESPACIO]A
    [5*ESPACIO]---FUEGO---
    [5*ESPACIO]ENTER
    [4*ESPACIO]"
120 PRINT:PRINT:PRINT TAB 9;
    "QUE GANE EL MEJOR"
130 FOR n=USR"a" TO USR"i"+7
140 READ d
150 POKE n,d
160 NEXT n
170 DATA 1,3,6,15,31,63,15,
    15
180 DATA 192,192,0,192,192,
    192,0,135
190 DATA 15,15,15,15,15,12,
    12,14
200 DATA 248,248,128,128,
    128,192,192,224

```

```

210 DATA 0,0,0,0,0,0,126,0
220 DATA 3,3,0,3,3,3,0,
    113
230 DATA 128,192,96,240,248,
    252,240,240
240 DATA 15,15,1,1,1,3,3,7
250 DATA 240,240,240,240,
    240,48,48,112
260 LET y1=10: LET y2=10
270 LET b1=6: LET b2=6
280 LET s1=0: LET s2=0
290 RESTORE 330:FOR n=1
    TO 8
300 READ d,p
310 BEEP d,p
320 NEXT n
330 DATA .1,7,.09,12,.1,7,
    .09,12,.6,7,.45,2,.45,6,
    .5,0
340 PRINT#1;AT 0,0; FLASH 1;
    PULSA UNA TECLA PARA
    JUGAR"
345 PAUSE 0
350 LET p=254: GO SUB 570
360 IF i=191 THEN GO TO 350
370 INK 4: BRIGHT 1: CLS
380 PRINT INVERSE 1;"JUGADOR
    1[14*ESPACIO]JUGADOR 2"
390 PRINT"BALAS: 6
    [16*ESPACIO]6 :BALAS"
410 GO SUB 600
420 LET p=63486: GO SUB 570
430 IF i=190 THEN GO SUB
    810
440 LET p=61438: GO SUB 570
450 IF i=190 THEN GO SUB
    840
460 LET p=64510: GO SUB 570
470 IF i=190 THEN GO SUB
    870
480 LET p=57342: GO SUB 570
490 IF i=190 THEN GO SUB
    900
500 LET p=49150: GO SUB 570

```





```

510 IF i=190 THEN GOSUB 1020
520 LET p=65022: GO SUB 570
530 IF i=190 THEN GO SUB 930
540 GO SUB 600
550 IF b1=0 AND b2=0 THEN GO
    TO 1110
560 GO TO 420
570 LET i=IN p
580 IF i>191 THEN LET i=i-64
590 RETURN
600 PRINT AT y1,1;CHR$ 144;
    CHR$ 145
610 PRINT AT y1+1,1;CHR$ 146
    ;CHR$ 147
620 PRINT AT y2,29;CHR$ 149;
    CHR$ 150
630 PRINT AT y2+1,29;CHR$
    151;CHR$ 152
640 PRINT AT y1-1,1;
    "[2*ESPACIO]"
650 PRINT AT y1+2,1;
    "[2*ESPACIO]"
660 PRINT AT y2-1,29;
    "[2*ESPACIO]"
670 PRINT AT y2+2,29;
    "[2*ESPACIO]"
680 PRINT AT 0,11;PAPER 4;
    INK 9;s1
690 PRINT AT 0,20;PAPER 4;
    INK 9;s2
700 PRINT AT 1,9;b1
710 PRINT AT 1,22;b2
720 RETURN
730 PRINT AT 10,10;"AAGH!,
    ME HAS DADO!"
740 RESTORE 780: FOR n=1 TO
    11
750 READ d,p
760 BEEP d,p
770 NEXT n
780 DATA .5,2,.4,2,.2,2,.5,
    2,.3,2,.2,4,.4,4,.2,2,
    .4,2,.2,1,.5,2
790 PRINT AT 10,10;
    "[16*ESPACIO]"
800 RETURN
810 IF y1<4 THEN RETURN
820 LET y1=y1-1
830 RETURN
840 IF y2<4 THEN RETURN
850 LET y2=y2-1
860 RETURN
870 IF y1>18 THEN RETURN
880 LET y1=y1+1
890 RETURN

```



```

900 IF y2>18 THEN RETURN
910 LET y2=y2+1
920 RETURN
930 IF b1=0 THEN RETURN
940 BEEP .01,4:BEEP .01,0
950 FOR n=3 TO 27
960 PRINT AT y1,n;" ";CHR$
    148
970 NEXT n
980 PRINT AT y1,27;
    "[2*ESPACIO]"
990 IF y1=y2 OR y1=y2+1 THEN
    LET s1=s1+1: GO SUB 730
1000 LET b1=b1-1
1010 RETURN
1020 IF b2=0 THEN RETURN
1030 BEEP .01,0:BEEP .01,-10
1040 FOR n=27 TO 3 STEP-1
1050 PRINT AT y2,n;CHR$ 148;
    " "
1060 NEXT n
1070 PRINT AT y2,3;" "
1080 IF y2=y1 OR y2=y1+1
    THEN LET s2=s2+1:
    GO SUB 730
1090 LET b2=b2-1
1100 RETURN
1110 IF s1>s2 THEN PRINT AT
    10,8;FLASH 1;" JUGADOR
    1 GANA! "

```

```

1120 IF s2>s1 THEN PRINT AT
    10,8;FLASH 1;" JUGADOR
    2 GANA!"
1130 IF s1=s2 THEN PRINT AT
    10,8;FLASH 1;" EMPATE! "
1140 GOTO 260

```

Ejecuta el programa y podrás ver un título de presentación, seguido de un escenario dispuesto para un duelo entre dos seres vestidos con traje espacial. Este juego puede ser jugado por dos personas. Las teclas para controlar a los dos seres espaciales son:

1 para mover hacia arriba el de la izquierda, Q para moverlo hacia abajo y A para disparar el láser; las teclas O, P y E realizan las mismas funciones con el ser de la derecha. El título de presentación y el campo de batalla se establecen en las líneas 10 y 4000. A continuación viene el bucle principal, que detecta las pulsaciones de tecla y envía el programa a las rutinas que mueven las figuras y disparan los láseres. El bucle es la sección de programa contenida entre las líneas 420 y 560. El resto del programa contiene rutinas de movimiento, disparo y presentación de mensajes.



# ENSAMBLADO A MANO (II)

- PRACTICA CON TU ENSAMBLADOR
- RUTINA PARA EL DESPLAZAMIENTO LATERAL DE PANTALLA
- ENTIENDE COMO FUNCIONA EL PROGRAMA

Ahora que ya has aprendido a ensamblar a mano, el mundo del ordenador doméstico se convierte en un vivero para tí. Ya posees un buen conocimiento del BASIC; puedes manejar además el código máquina y la traducción de lenguaje ensamblador a código máquina no te presentará problemas.

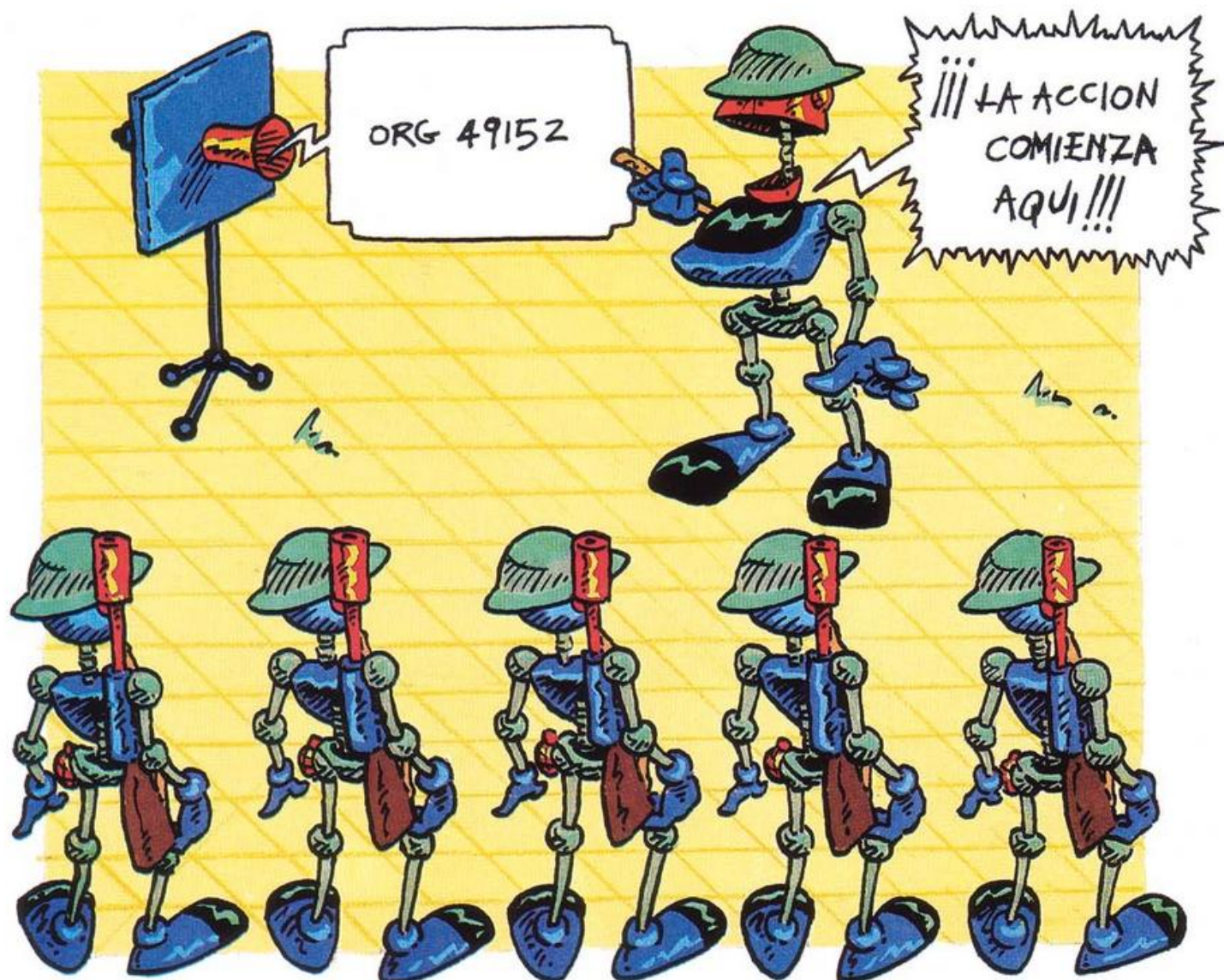
En la primera parte de este artículo has podido ver cómo se hace el ensamblado a mano y cómo se convierte el lenguaje ensamblador en código máquina, por lo que puedes introducirlo en el ordenador y ejecutarlo.

Cuando domines este tema, estarás en condiciones de emplearlo para algunas rutinas útiles.

## SCROLLING LATERAL

Los siguientes programas en lenguaje ensamblador sirven para desplazar lateralmente la pantalla hacia la derecha o hacia la izquierda el espacio de un carácter. Ensámblalos a mano e introduce el código máquina en tu ordenador, con ayuda de un programa monitor de código máquina. Si quieres almacenar estos programas con vistas a su posible utilización futura, no te olvides de ponerlos en secciones distintas de la memoria, para el caso de que quieras usar ambos al mismo tiempo.

Los dos programas funcionan independientemente de su situación en la memoria, por lo que los puedes poner





Defiende tu espacio aéreo

# EL Nº 1 ahora en SPECTRUM



Simulación de vuelo tridimensional de combate aéreo y ataque al suelo.

Cinco niveles, quince escenarios y capacidad de juego estratégico.

Estás en la cabina del caza que sería el sueño de cualquier piloto, pero desde luego eres un mal sueño para el pobre tipo que tienes delante, confiado en una misión sin problemas. Calientale la tobera con tus láser y apartate mientras estalla en una bola de fuego. Rápidamente ponte en picado para caer sobre los blindados

enemigos, como la peste entre los cerdos. SKYFOX es el juego que más rápidamente se está vendiendo en toda la historia de Electronic ARTS.

Posee la más asombrosa animación de alta velocidad que hayas visto en tu ordenador. Ahora puede ser tuyo totalmente traducido al castellano.

# SKYFOX™

## EN CASTELLANO



DRO SOFT

P.V.P. 2.500 ptas.



ELECTRONIC ARTS

**CARACTERISTICAS:** NACIONALIDAD: Federación galáctica. **FABRICANTE:** TOBEY ASTRONAUTICS **TIPO:** Caza interceptor multipropósito **PROPULSION AUXILIAR:** Un generador antigraavitatorio a 66 MKI **TRIPULACION:** Un humano. **ARMAMENTO:** Dos cañones láser de fuego continuo de 70 kilojulios 10 toneladas de empuje. 5 misiles rastreadores de calor tipo PHOENIX 5 misiles guiados por radar tipo TYPHOON **DEFENSA:** — 2 unidades deflectoras WCRC **AYUDAS ELECTRONICAS:** Radar SCANNER de largo y corto alcance conectable al piloto automático. **VELOCIDAD EN ATMOSFERA:** — 3.000 MPH (Mach IV a 35.000 pies).

Editado por DRO SOFT Fundadores, 3 - 28028 Madrid.

Tel.: 255 45 00/09



en cualquier posición que resulte conveniente para tu máquina.

Si quieres hacer un *scrolling* de pantalla hacia la derecha o hacia la izquierda, de más de un cuadrado de extensión, puedes llamar a esta rutina desde un bucle FOR ... NEXT. O si quieres hacerlo cada vez que se pulse una determinada tecla, puedes usar un programa en BASIC que utilice IN-KEY\$. La siguiente rutina sirve para desplazar lateralmente la pantalla hacia la izquierda. Aquí la hemos traducido a código máquina.

**Tecllea** para Spectrum

	Ld de,16384	11 00 40
	Ld hl,16385	21 01 40
	Ld b,192	06 C0
bucle	push bc	C5
	Ld a,(de)	1A
	Ld bc,31	01 1F 00
	Ldir	ED B0
	Ld (de),a	12
	inc de	13
	inc hl	23
	pop bc	C1
	djnz bucle	10 F3
	ret	C9

## FUNCIONAMIENTO

Las instrucciones **Ld de,16384** y **Ld hl,16385** cargan las direcciones de los dos primeros bytes del fichero de presentación de la pantalla del **Spectrum** en los registros DE y HL. La instrucción **Ld b, 192** carga el registro B con el número 192; en la pantalla del **Spectrum** hay 192 líneas y utilizaremos el registro B como contador de líneas.

El problema es que también hay que utilizar el registro B para otras cosas, por lo que de momento su contenido se almacenará temporalmente en el *stack*. Pero la instrucción relativa al *stack* no se quedará únicamente con el contenido del registro B, sino que también afecta al registro C. Por ello se utiliza **push bc**. No importa lo que haya en el registro C, ya que aquí no nos ocuparemos para nada de su contenido.

La instrucción **Ld a,(de)** carga el acumulador con el contenido de la direc-

ción de memoria almacenada en el registro DE; recuerda que es la 16384. Es un ejemplo de direccionamiento indirecto.

La instrucción **Ld bc,31** carga el registro BC con el número 31. En una línea hay 32 caracteres, pero el desplazamiento de la primera a la última posición se hace independientemente. Así, para esta operación sólo necesitas contar hasta 31.

La instrucción que resulta crucial aquí es **Ldir**, que significa cargar, incrementar y repetir. Lo que esta instrucción hace es cargar el contenido de la dirección de memoria que está almacenada en el registro HL —16385 en el primer paso— en la posición de memoria cuya dirección está almacenada en DE, que en el primer paso es 16384. Seguidamente se incrementan HL y DE, se decrementa BC y se com-

prueba si BC es cero; si no es cero se repite toda la instrucción.

El efecto global de esto es mover lo que había en la segunda posición de pantalla a la primera posición, a continuación lo que había en la tercera posición se traslada a la segunda posición, y así sucesivamente a lo largo de la primera línea. El proceso se detiene cuando el registro BC ha contado desde 31 hasta 0. La última operación desplaza la posición 32 de pantalla a la posición 31. Pero cuando se decrementa el registro BC, se alcanza el valor cero y el programa pasa a ejecutar la siguiente instrucción.

La instrucción **Ld (de),a** carga el contenido del acumulador en la dirección de memoria a la que está apuntando el registro DE. Si miras hacia atrás, verás que el acumulador contiene lo que contenía el registro DE antes de





que la instrucción **ldir** comenzara a incrementarlo; en otras palabras, el contenido de 16384, que era la primera posición de pantalla.

Sin embargo el registro DE ha sido repetidamente incrementado desde entonces. Ahora está apuntando a la última posición de pantalla, que es precisamente la posición en la que quieres poner el contenido de la primera posición de pantalla.

Las instrucciones **inc hl** e **inc de** incrementan los registros HL y DE de forma que quedan inicializados para el principio de la siguiente línea. Como habías cambiado a mano (o casi a mano) el contenido desde la primera posición de pantalla hacia la última, tienes que tomarte la molestia de incrementar tú estos registros en vez de dejar que la instrucción **ldir** lo haga por tí.

Seguidamente **pop bc** extrae el contenido de las dos direcciones de memoria que hay en lo alto del *stack*, cargándolo en los registros B y C. La instrucción **djnz** a continuación decrementa el registro B y salta si su contenido no es cero. Al principio de la primera pasada habías cargado el registro B con el valor 192. Este valor fue almacenado en el *stack* mientras que tú hacías otras cosas con los registros B y C. Con **djnz** se decrementa el registro B a 191 después del primer paso. Como el resultado no es cero la instrucción vuelve a saltar al sitio donde aparece por primera vez la etiqueta que se le dio: bucle.

Este salto se producirá una vez y otra mientras la instrucción prosiga su cuenta atrás desde 192 hasta cero, actuando cada vez sobre una línea distinta de la pantalla. Cuando el regis-

tro B llega a tener el valor cero, después de que ha sido decrementado mediante la instrucción **djnz**, el salto ya no se produce y el programa pasa a ejecutar la siguiente instrucción.

La instrucción **ret** significa return (volver). Ninguna rutina en código máquina funcionará sin una instrucción de este tipo, ya que si no la tiene no puede volver al BASIC. En tal caso el microprocesador empezaría a roturar el galimatías que se encontrara en la memoria a continuación de tu programa en lenguaje máquina. Lo más probable es que el programa se interrumpiera y la memoria resultara contaminada.

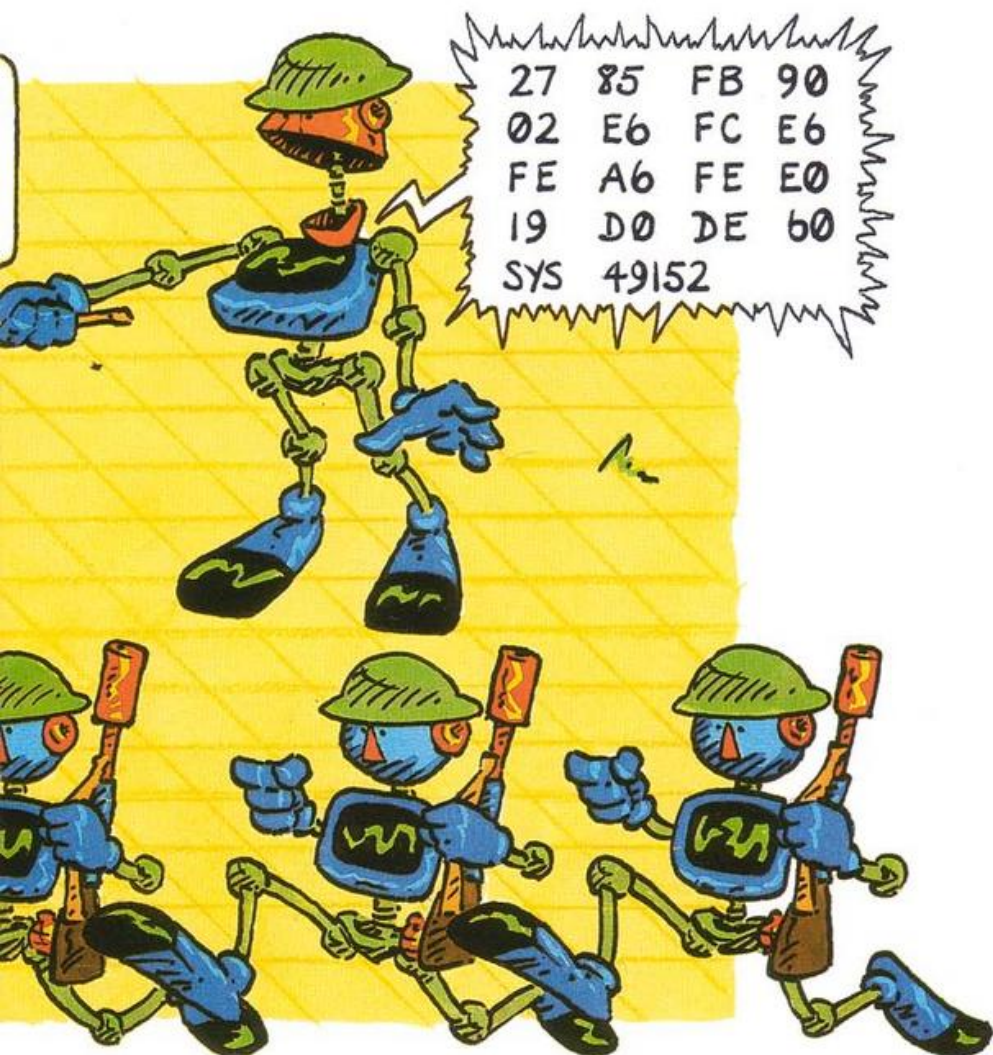
Aunque esto no ocurriera el microprocesador correría de un extremo a otro de la memoria una y otra vez. Llegaría a las rutinas de inicialización que en cualquier caso borrarían la memoria. Recuerda pues la importancia de **ret**.

Prueba ahora a ensamblar a mano la siguiente rutina para desplazar la pantalla lateralmente hacia la derecha.

**Teclea** para Spectrum

```
ld de,22527
ld hl,22526
ld b,192
bucle push bc
ld a,(de)
ld bc,31
lddr
ld (de),a
dec hl
dec de
pop bc
djnz bucle
ret
```

Como habrás observado, este programa empieza al final del fichero de presentación de pantalla, avanzando en el mismo de atrás hacia adelante. La otra única diferencia respecto al programa anterior es la instrucción **lddr**, que significa cargar, decrementar y repetir. Esta instrucción carga el contenido de la casilla de memoria cuya dirección está almacenada en HL en la casilla cuya dirección está en DE. Pero ahora lo que hace es decre-





mentar HL y DE. Decrementa BC, igual que en la instrucción **ldir**, repitiendo nuevamente el proceso si el contenido de los registros BC es distinto de cero.

El salto es de la misma longitud que antes. Por eso puedes comprobar si has colocado la etiqueta correctamente, mirando el anterior listado de código máquina, (acuérdate de contar los bytes a partir del final de la instrucción de salto, lo cual significa que has de incluir el byte de la propia instrucción sobre la que estás trabajando).

La siguiente rutina en lenguaje ensamblador hace el *scrolling* de pantalla hacia la izquierda en cualquier **ZX81** que tenga instalados 8K o 16K de RAM. Aquí lo tienes traducido a lenguaje máquina para tí.

Teclea para ZX81

LD DE,(16396)	ED 5B 0C	40
INC DE		13
LD H,D		62
LD L,E		6B
INC HL		23
LD B,24	06 18	
BUCPUSH BC		C5
LD A,(DE)		1A
LD BC,31	01 1F 00	
LDIR	ED B0	
LD (DE),A		12
INC HL		23
INC HL		23
INC DE		13
INC DE		13
POP BC		C1
DNJZ BUC	10 F1	
RET		C9

## PARA EL ZX81

El fichero de presentación del **ZX81** no está en un sitio fijo; por el contrario, el puntero de las direcciones de memoria 16396 y 16397 contiene la dirección de la primera posición de pantalla. La instrucción **LD DE, (16396)** carga los registros D y E con los contenidos de las direcciones 16396 y 16397; de acuerdo con la convención bajo-alto del **Z80**, el contenido de 16396 va al registro E y el contenido de 16397 va al registro D. Ahora el

programa ya sabe dónde tiene que empezar su rutina de *scrolling*.

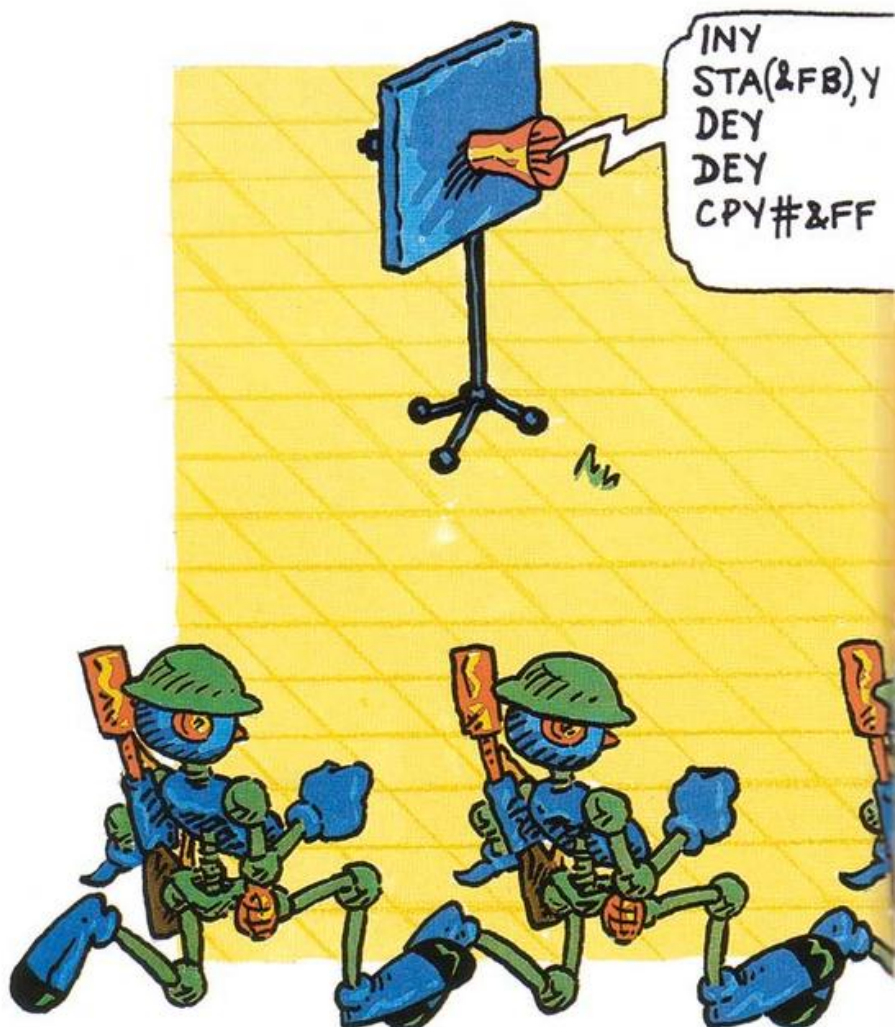
La presentación en pantalla del **ZX81** se produce además de una forma bastante diferente de la de otras máquinas. Hay un símbolo de nueva línea al principio de cada línea (que sin embargo no aparece en la pantalla). Se utiliza para que el fichero de presentación pueda encogerse, ocupando el menor espacio posible en la memoria no expandida de 1K del **ZX81**. Si no se está visualizando nada en la pantalla, el fichero de presentación puede contraerse hasta ocupar únicamente los 24 caracteres de nueva línea, marcando el punto donde tendría que comenzar cada línea.

En las versiones expandidas, estos caracteres de nueva línea se dejan al final de cada línea. El problema es que no deben interferir con ellos, ya que si lo haces el ordenador se parará.

El primer carácter de toda la pantalla, empezando por la parte superior izquierda, es un carácter de nueva línea. Para evitar interferir con él, la instrucción **INC DE** incrementa el par de registros DE, lo cual desplaza el programa a la segunda posición de pantalla.

Las dos instrucciones **LD H,D** y **LD L,E** copian el contenido del par de registros DE sobre el par HL. Por último **INC HL** incrementa el par de registros HL para dar la dirección de la tercera posición de pantalla.

**LD B,24** carga el registro B con el número 24. La presentación en pantalla del **ZX81** tiene 24 líneas y el registro B va a actuar como contador de las mismas. Pero el registro B tiene que ser utilizado también para otras cosas. Por eso su contenido se almacenará temporalmente en el *stack*, lo que se hace con la instrucción **PUSH BC**.





La instrucción **PUSH** realmente transfiere los contenidos de los registros B y C al *stack*. No existe una instrucción que permita transferir separadamente al *stack* el contenido del registro B solo. De todos modos, aquí no estamos interesados en lo que ocurre con el registro C, ya que ello no afecta para nada al registro B que es el que se va a usar como contador.

La instrucción **LD A,(DE)** carga el acumulador con el contenido de la célula de memoria cuya dirección está en el par de registros DE; en otras palabras, el contenido de la segunda posición de pantalla, recuérdalo. Seguidamente, **LD BC,31** carga los registros BC con el número 31. En una línea del **ZX81** hay 33 caracteres, pero como sabes el primero es un carácter de línea nueva. Y cuando los caracteres se desplazan una casilla para hacer el *scrolling*, sólo hay que hacer esta

operación 31 veces. El cuadrado que queda ha de ser desplazado de un extremo a otro de la línea, lo cual se hace mediante una operación separada.

La instrucción crucial aquí es la **LDIR**, que significa cargar, incrementar y repetir. Lo que hace esta instrucción es cargar el contenido de la dirección de memoria que está almacenada en HL —que en la primera pasada es la tercera posición de pantalla— en la posición de memoria cuya dirección está almacenada en DE, la segunda dirección de memoria durante la primera pasada. Seguidamente incrementa HL y DE, decrementa BC y mira si el contenido de BC es cero; si no es cero se repite toda la instrucción.

El efecto global de todo esto es trasladar lo que hay en la tercera posición de pantalla a la segunda posición, después lo que hay en la cuarta a la ter-

cera, y así sucesivamente a lo largo de toda la primera línea. El proceso se detiene cuando la cuenta del registro BC ha pasado desde 31 hasta 0. La última operación que realiza es mover el contenido de la posición 33 de pantalla a la posición 32. Después de que el contenido del registro BC se ha decrementado hasta cero el programa pasa a la ejecución de la siguiente instrucción.

La instrucción **LD (DE),A** carga el contenido del acumulador en la dirección contenida en los registros DE, siendo complementaria de la instrucción **LD A,(DE)** usada anteriormente. Dicha instrucción se utilizó para almacenar el contenido de la segunda posición de pantalla en el acumulador.

Ahora el contenido de la segunda posición de pantalla se coloca en la posición de pantalla contenida en los registros DE. Pero entretanto DE ha sido incrementado 31 veces por la instrucción **LDIR** por lo que ahora está apuntando hacia la última posición de pantalla de esa línea, que es precisamente donde queremos poner el contenido de la segunda posición de pantalla en el caso de que queramos hacer una rutina de *scrolling* cíclico.

Las instrucciones **INC HL** e **INC DE** incrementan los pares de registros HL y DE. Esto se hace una vez para que el programa se vaya a la siguiente línea de la pantalla y una segunda vez para evitar interferir con el carácter de línea nueva al principio de la línea.

**POP BC** saca los contenidos de las dos posiciones superiores de memoria del *stack* y los transfiere a los registros BC. En otras palabras, establece de nuevo el contenido inicial del registro contador BC.

**DJNZ** decrementa el registro B y salta si su contenido no es cero. Para empezar se cargó el registro B con 24. Este número se almacenaba en el *stack*, extrayéndose posteriormente. Después se decrementa a 23. Como este valor no es cero, la instrucción ejecuta el salto, por lo que el ordenador salta otra vez a la sección de programa donde se encuentra la etiqueta **BUCLE**.

Este salto se producirá una vez y otra. El ordenador se ve continuamen-

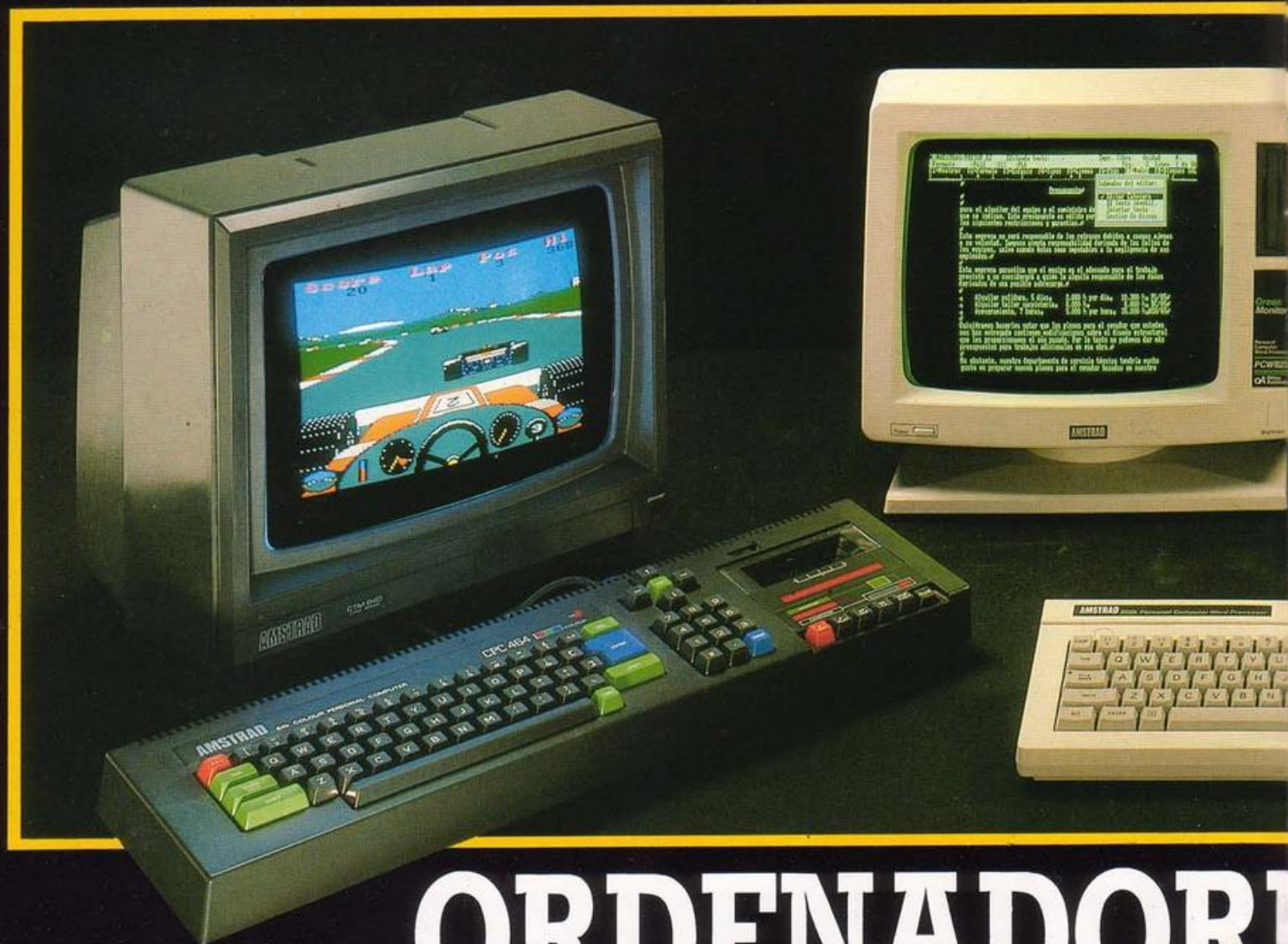
```
BNE LOOP
LDY #200
LDA &FD
STA(&FB),Y
LDA &FB
ADC #27
```





# AMSTRAD CPC - 464

# AMSTRAD



## ORDENADOR

### SERIE CPC

#### UNIDAD CENTRAL. MEMORIAS

- Microprocesador Z80A - 64K RAM ampliables - 32K ROM ampliables
- **TECLADO** • Teclado profesional con 74 teclas en 3 bloques - Hasta 32 teclas programables - Teclado redefinible
- **PANTALLA** • Monitor RGB verde (12") o color (14")

	Normal	Alta Res.	Multicolor
Col x líneas	40 x 25	80 x 25	20 x 25
Colores	4 de 27	2 de 27	16 de 27
Puntos	320 x 200	640 x 200	160 x 2

- Se pueden definir hasta 8 ventanas de texto y 1 de gráficos • **SONIDO**
- 3 canales de 8 octavas moduladas independientemente - Altavoz interno regulable - Salida estéreo • **BASIC**
- Locomotive BASIC ampliado en ROM - Incluye los comandos AFTER y EVERY para control de interrupciones

### AMSTRAD CPC 464

**CASSETTE** • Cassette incorporada con velocidad de grabación (1 ó 2 Kbaudios) controlada desde Basic • **CONECTORES**

- Bus PCB multiuso, Unidad de Disco exterior, paralelo Centronics, salida estéreo, joystick, lápiz óptico, etc.
- **SUMINISTRO** • Ordenador con monitor verde o color - 8 cassettes con programas - Libro "Guía de Referencia BASIC para el programador" - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

**TODO POR** 59.900 Pts. (monitor verde)  
90.900 Pts. (monitor color)

### AMSTRAD CPC 6128

**UNIDAD DE DISCO** • Unidad incorporada para disco de 3" con 180K por cara • **SISTEMAS OPERATIVOS**

- AMSDOS, CP/M 2.2, CP/M Plus (3.0)
- **CONECTORES** • Bus PCB multiuso, paralelo Centronics, cassette exterior, 2ª Unidad de Disco, salida estéreo, joysticks, lápiz óptico, etc.
- **SUMINISTRO** • Ordenador con monitor verde o color - Disco con CP/M 2.2 y lenguaje DR. LOGO - Disco con CP/M Plus y utilidades - Disco con 6 programas de obsequio - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA.

**TODO POR** 84.900 Pts. (monitor verde)  
119.900 Pts. (monitor color)



# PCW - 8256

# AMSTRAD CPC - 6128



# ES AMSTRAD

## AMSTRAD PCW 8256

### UNIDAD CENTRAL. MEMORIAS

- Microprocesador Z80A - 256K RAM de las que 112K se utilizan como disco RAM
- **TECLADO** • Teclado profesional en castellano (ñ, acento...) de 82 teclas
- **PANTALLA** • Monitor verde de alta resolución - 90 columnas × 32 líneas de texto
- **UNIDAD DE DISCO** • Disco de 3" y 173K por cara - Opcionalmente, 2.ª Unidad de Disco de 1 Mbyte integrable
- **SISTEMA OPERATIVO** • CP/M Plus de Digital Research • **IMPRESORA** • Alta calidad (NLQ) a 20 c.p.s. - Calidad estándar a 90 c.p.s. - Papel continuo u hojas sueltas - Alineación automática del papel - Caracteres normales, comprimidos, expandidos, control del paso de letra (normal, cursiva, negrita, subíndices, superíndices, subrayado, etc).
- **OPCIONES** • Kit de Ampliación a 512K RAM y 2.ª Unidad de Disco - Interface Serie RS 232C y paralelo

Centronics • **SUMINISTRO** • Ordenador completo con teclado, pantalla, Unidad de Disco e Impresora - Discos con el procesador de Texto LocoScript, CP/M Plus, Mallard, BASIC, DR. LOGO y diversas utilidades - Manuales en castellano - Garantía Oficial AMSTRAD ESPAÑA.

**TODO POR 129.900 Pts.**



Los más prestigiosos paquetes de Software Profesional, en formato AMSTRAD... a "precios AMSTRAD"

# ¡Increíble!!

Existe también la versión **PCW 8512** con **512K RAM** y la 2.ª Unidad de Disco de 1 Mbyte incorporada. **PVP. 174.900 Pts.**

\* El **PCW 8256** puede utilizarse como terminal y en comunicaciones.

El I.V.A. no está incluido en los precios.

**NOTA:** Es muy importante verificar la garantía del aparato ya que sólo **AMSTRAD ESPAÑA** puede garantizarle la ordenada reparación y sobre todo materiales de repuesto oficiales (Monitor, ordenador, cassette o unidades de discos).

# AMSTRAD

ESPAÑA

Avda. del Mediterráneo, 9. Tels. 433 45 48 - 433 48 76.  
28007 MADRID

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



te enviado a ejecutar el bucle mientras el contador se va decrementando desde 24 hasta 0, permitiendo que se realice la rutina de desplazamiento de carácter una vez cada línea. Cuando el contenido del registro B se hiciera 0, no se ejecuta el salto y el ordenador pasa a la siguiente instrucción.

**RET** significa RETURN (volver). Ninguna subrutina en código máquina funcionará bien sin una instrucción de este tipo, ya que sin ella el ordenador no podrá volver al BASIC. Lo que ocurriría en tal caso es que el microprocesador se pondría a rastrear toda la sección de memoria situada a continuación de tu programa en código máquina, lo que naturalmente será para él un galimatías sin sentido. Lo más probable es que el programa se detenga y además la memoria se verá contaminada.

Prueba ya a ensamblar a mano por tí mismo la rutina de *scrolling* de pantalla hacia la derecha. No te olvides de hacer la conversión de números y direcciones a hexadecimal.

Teclea ZX81

```
LD HL,(16396)
LD DE,790
ADD HL,DE
LD D,H
LD E,L
INC DE
LD B,24
BUCLE PUSH BC
LD A,(DE)
LD BC,31
LDDR
LD (DE),A
DEC HL
DEC HL
DEC DE
DEC DE
POP BC
DJNZ BUCLE
RET
```

Este programa empieza al final del fichero de presentación y avanza de atrás hacia adelante. Para hacer esto hay que cargar otra vez el contenido

del puntero de pantalla en DE, con la instrucción **LD DE,(16396)**, sumándole a continuación 790, que es el número de caracteres cuadrados menos dos que caben en una pantalla de **ZX81**. Como el puntero de pantalla está apuntando a la primera posición de pantalla, la última posición será ese número, más el número de posiciones de pantalla y menos uno. Pero el primer carácter cuadrado que se va a desplazar es el inmediatamente anterior, por lo que hay que restar uno más.

La instrucción encargada de hacer la suma es **ADD HL,DE**. Su efecto es sumar el contenido de HL y DE, poniendo el resultado en HL. El programa avanza exactamente igual que antes hasta que se llega a la instrucción **LDDR**. Esta instrucción es muy parecida a la **LDIR**; su significado es cargar, decrementar y repetir. En otras palabras, carga en la dirección de memoria contenida en DE, el contenido de la dirección de memoria que hay almacenada en HL, igual que hacía antes, pero ahora decrementa HL y DE, decrementa nuevamente BC y repite el proceso si el registro BC no contiene un cero.

La amplitud del salto es la misma que antes, con lo que puedes comprobar que la has calculado correctamente observando el anterior listado en código máquina. Finalmente, acuérdate de contar los bytes a partir del final de la instrucción de salto.

## 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.

## GANADORES DE LOS MEJORES DE INPUT SINCLAIR

En el sorteo correspondiente al número 7 realizado entre quienes escribisteis mandando vuestros votos a **LOS MEJORES DE INPUT** han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Carlos García Hormigos	Madrid	Ole Toro
Alfonso Jiménez Sánchez	Mérida (Badajoz)	Rambo
Francisco López Urtiaga	Pontevedra	Camelot Warrior
Santiago Mesa Ordóñez	Atienza (Guadalajara)	Winter Games
Delfín González Peralta	S. S. de los Reyes (Madrid)	Saboteur
José Manuel Pérez Merino	León	Shadow of Unicorn
Alfonso López Ruiz	Sama de Langreo (Asturias)	Superman
Juan Alfonso Gómez Ruiz	Badalona (Barcelona)	Misión Imposible
M.ª del Pilar González Franco	León	Panorama para matar
Esteban de la Sota Alonso	Las Arenas Gexto (Vizcaya)	Basket International



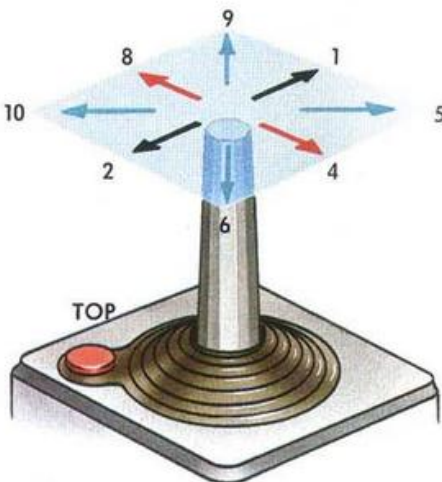
# LA PROGRAMACION PARA JOYSTICKS

- COMO LEER LOS JOYSTICKS DESDE EL BASIC
- HAZ TUS JUEGOS MAS PROFESIONALES
- EL PULSADOR DE FUEGO

Los *joysticks* son la clave para tener unos juegos más profesionales. Pero el hacer que tus juegos resulten más divertidos no significa que tengas que aprender código máquina; puedes empezar con el BASIC.

Una diferencia evidente entre los juegos comerciales y los producidos en casa es con frecuencia la existencia de una opción para *joystick*. Sin embargo no tienes que sumergirte en las profundidades intrincadas del código máquina para utilizar los *joysticks* en tus propios programas.

En esta sección del coleccionable, veremos la manera de utilizar los *joysticks* con programas escritos en BASIC, haciendo que tus programas de

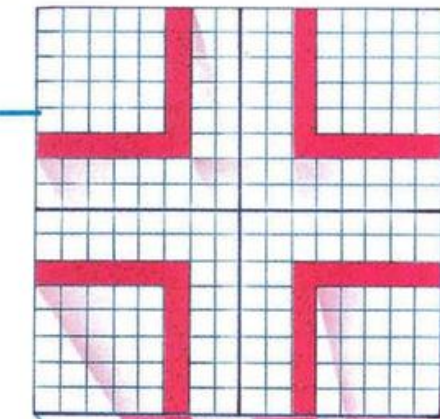


El joystick del **Spectrum** da uno de estos ocho valores dependiendo de la dirección en que se desvíe.

juegos resulten más profesionales y más divertidos de jugar.

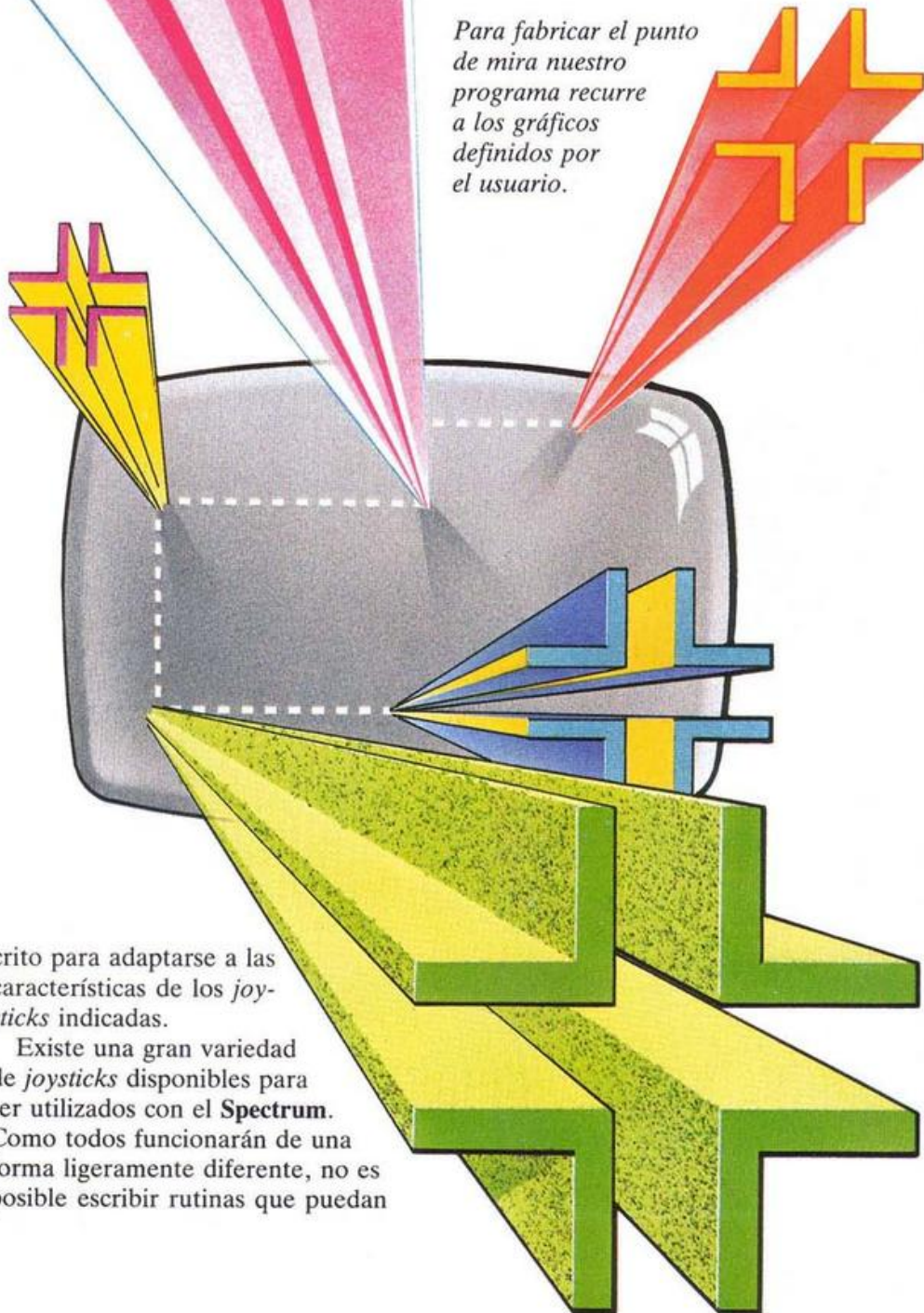
En el próximo capítulo utilizaremos la rutina de *joystick* en un juego, por lo que no debes de olvidarte de almacenar el programa.

Lo primero que necesitas es un *joystick* adecuado. Todo programa está es-



usarse con todos los modelos disponibles. En lugar de eso, lo que haremos es ver cómo se escribe una rutina que funcione con el tipo de *joystick* más ampliamente usado, el **Kempston**, así como con otros *joysticks* que se adaptan a la norma de **Kempston**.

Para fabricar el punto de mira nuestro programa recurre a los gráficos definidos por el usuario.



crito para adaptarse a las características de los *joysticks* indicadas.

Existe una gran variedad de *joysticks* disponibles para ser utilizados con el **Spectrum**. Como todos funcionarán de una forma ligeramente diferente, no es posible escribir rutinas que puedan



## UN PUNTO DE MIRA

Teclea la siguiente sección de programa y tendrás un punto de mira controlable con tu joystick:

```

100 BORDER 1:PAPER 1:INK 7:
    OVER 1:CLS:INK 8
110 FOR n=USR "a" TO USR "h"
    +7:READ a:POKE n,a:
    NEXT n
130 LET s=0:LET x=15:LET
    y=10
140 PRINT OVER 1;AT y,x;CHR$
    148;CHR$ 149;AT y+1,x;
    CHR$ 150;CHR$ 151
200 GOSUB 500
480 GOTO 200
500 IF IN 31=0 THEN RETURN
510 PRINT OVER 1;AT y,x;CHR$
    148;CHR$ 149;AT y+1,x;
    CHR$ 150;CHR$ 151
520 IF (IN 31=8 OR IN 31=9
    OR IN 31=10) AND y>1
    THEN LET y=y-1
530 IF(IN 31=4 OR IN 31=5 OR
    IN 31=6)AND y<20 THEN
    LET y=y+1
540 IF(IN 31=1 OR IN 31=5 OR
    IN 31=9)AND x<30 THEN
    LET x=x+1
550 IF(IN 31=2 OR IN 31=6 OR
    IN 31=10)AND x>0 THEN
    LET x=x-1
560 PRINT OVER 1;AT y,x;CHR$
    148;CHR$ 149;AT y+1,x;
    CHR$ 150;CHR$151
570 RETURN
1000 DATA 14,27,127,31,15,7,
    15,31
1010 DATA 0,0,0,0,0,192,112,
    188
1020 DATA 31,29,30,15,3,1,1,3
1030 DATA 206,30,124,248,224
    ,64,64,224
1040 DATA 12,12,12,12,252,
    252,0,0
1050 DATA 48,48,48,48,63,63,
    0,0
1060 DATA 0,0,252,252,12,12,
    12,12
1070 DATA 0,0,63,63,48,48,48
    ,48
    
```

El programa empieza inicializando los colores de la pantalla. A continuación se crean ocho GDU utilizando las sentencias DATA de las líneas 1000 a 1070. Sin embargo no se utilizan todavía, ya que no sólo estás definiendo un punto de mira sino que además estás creando la imagen de un pato que utilizarás cuando hayas tecleado el resto del programa (que veremos en el próximo capítulo).

La línea 140 imprime la mitad superior del punto de mira utilizando dos de los GDU, seguida de la mitad inferior que utiliza otros dos GDU. El punto de mira se controla llamando a la subrutina que empieza en la línea 500, llamada que se produce en la línea 200.



Una vez creados los GDU, la línea 130 establece la posición de partida del punto de mira. Dicha línea también pone a cero la puntuación, aunque tampoco ésta se utilizará hasta más ade-



La subrutina de *joystick* que comienza en la línea 500, examina el valor de IN 31 para detectar de qué forma está siendo accionado el *joystick*. Lo que hace la función IN es mirar a un determinado *port* dentro de la máquina para ver el valor que entra. Como el *joystick Kempston* direcciona el *port* número 31, el programa examina los valores de IN 31.

En la figura 1 se muestran los valores que puede tomar IN 31. Las cuatro direcciones principales arrojan los valores 1, 2, 4 y 8. Se pueden detectar además los movimientos diagonales del *joystick* observando el total de dos direcciones contiguas.

La primera línea de la subrutina (línea 500) detecta si el *joystick* se está posicionando cen-

tralmente. Cuando el *joystick* está suelto y los muelles le mantienen en la posición de equilibrio central, el valor de IN 31 será cero.

La línea 510 sirve para borrar el punto de mira ya que ésta es la segunda vez que se imprime OVER 1; la primera vez lo hiciste en la línea 140. OVER 1 borra el gráfico cuando se utiliza por segunda vez, por lo cual el gráfico desaparecerá.

Ahora que ya ha sido borrada la anterior posición, se puede calcular la siguiente. La nueva posición depende de la forma en que el *joystick* haya sido accionado por el jugador. La línea 520 detecta las tres posiciones que indican un movimiento hacia abajo: ya sea vertical o según una de las diagonales. La línea 530 detecta las tres posibilidades análogas de movimiento hacia arriba y las líneas 540 y 550 detectan los movimientos hacia la izquierda y hacia la derecha respectivamente. Si te parece algo confuso, observa la figura 1.

La subrutina finaliza presentando el punto de mira en su nueva posición. Observa que al ser ésta la primera vez que dicho punto de mira aparece en esa posición, aparece con toda normalidad.

Para que puedas continuar desplazando el punto de mira, la línea 480 contiene la instrucción GOTO 200 para que la rutina del *joystick* pueda ser llamada repetidamente.

## P y R

¿Se le puede agregar una rutina de *joystick* a cualquiera de los juegos que aparecen en INPUT?

El control por *joystick* no es más que una alternativa conveniente al control por teclado por lo que puedes sustituir el núcleo de la mayoría de los programas que aparecen en nuestro coleccionable en los cuales se utilice INKEY\$ para leer el teclado.

Las líneas importantes van de la 520 a la 550.



# EL JUEGO DE LA CAZA DE PATOS

Para los que no puedan esperar a que se levante la veda o no quieran tiritar de frío en el campo, aquí presentamos una rutina de disparo contra los patos, que podréis utilizar junto con la rutina de *joystick*.

Si has seguido atentamente el capítulo anterior, tendrás guardado en cinta un programa para mover por la pantalla el punto de mira. Pero aunque resulte satisfactorio disponer de un programa de este tipo escrito en BASIC, tal como está no vale para mucho.

Por eso el siguiente paso es utilizar la nueva rutina dentro de un programa de juegos. Al añadir las siguientes líneas de programa especialmente escritas para tu máquina, tendrás un juego de caza de patos salvajes, aunque naturalmente puedes dibujar tus propios gráficos y disparar contra aviones, búfalos, dirigibles o cualquier otra cosa que te sugiera tu fantasía.

El objetivo del juego es abatir diez patos que aparecen en un corto intervalo de tiempo en posiciones aleatorias de la pantalla. El tanteo se basa en tu habilidad. Obtienes puntos por cada impacto, cuanto más rápido seas, más puntos obtendrás.

Además por cada tiro fallido se te quitarán puntos.

Carga en tu ordenador la rutina de *joystick* antes de teclear las nuevas líneas. A continuación añádele éstas y tendrás un programa de caza de patos con puntuación incorporada:

```

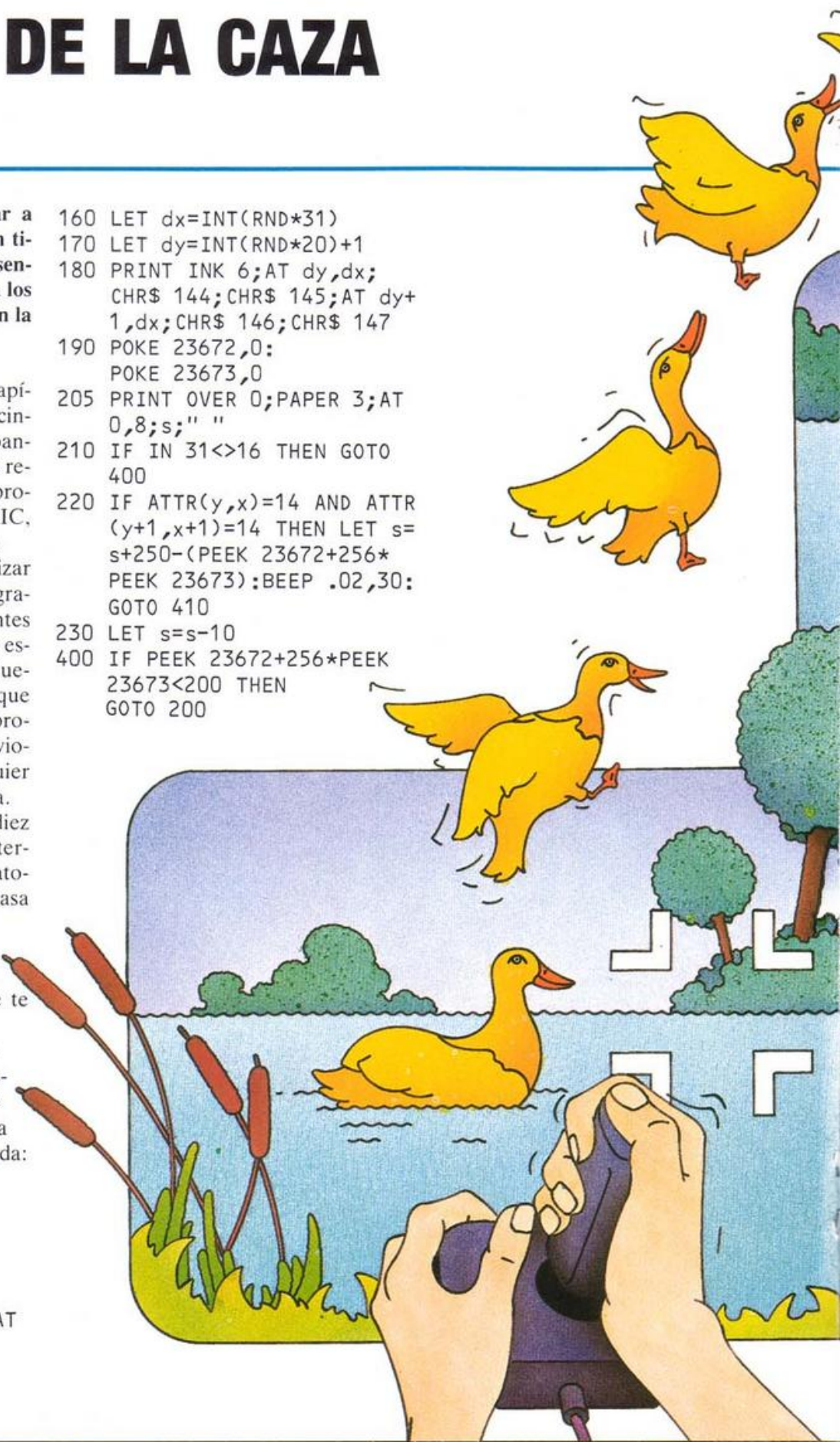
160 LET dx=INT(RND*31)
170 LET dy=INT(RND*20)+1
180 PRINT INK 6;AT dy,dx;
  CHR$ 144;CHR$ 145;AT dy+
  1,dx;CHR$ 146;CHR$ 147
190 POKE 23672,0;
  POKE 23673,0
205 PRINT OVER 0;PAPER 3;AT
  0,8;s;" "
210 IF IN 31<>16 THEN GOTO
  400
220 IF ATTR(y,x)=14 AND ATTR
  (y+1,x+1)=14 THEN LET s=
  s+250-(PEEK 23672+256*
  PEEK 23673):BEEP .02,30;
  GOTO 410
230 LET s=s-10
400 IF PEEK 23672+256*PEEK
  23673<200 THEN
  GOTO 200

```

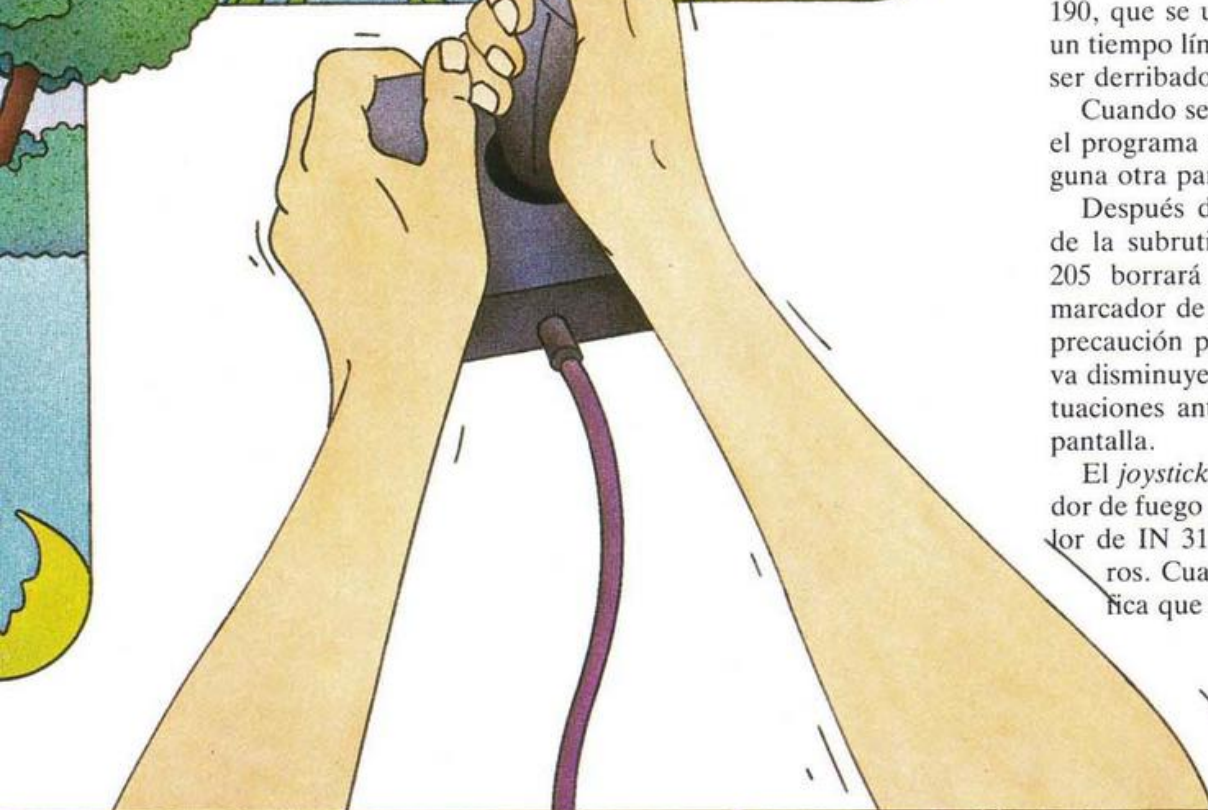
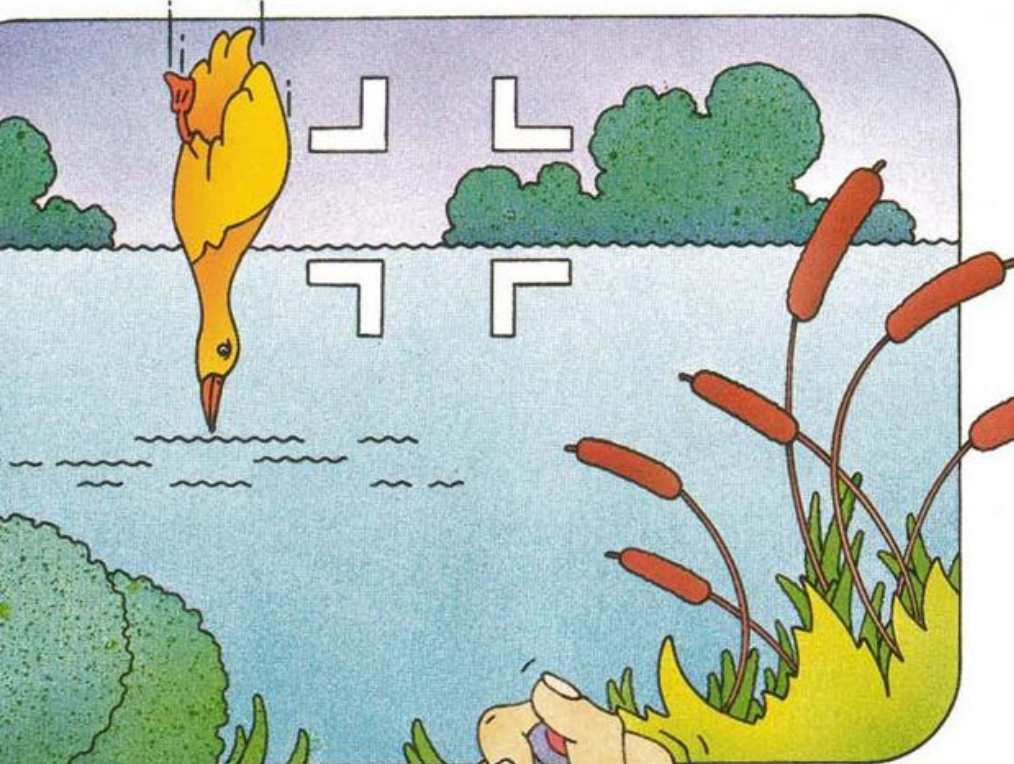
```

120 LET hs=0
125 PRINT PAPER 2;INK 7;
  " PUNTOS";TAB 14;
  "MAXPUNT.";TAB 31;" "
135 PRINT PAPER 3;OVER 0;AT
  0,23;hs
145 LET s=0
150 FOR n=1 TO 10

```







■	USO DE LA RUTINA DE JOYSTICKS
■	EL GRAFICO DEL PATO
■	RUTINA DE TIEMPOS
■	DETECCION DE LOS DISPAROS

```

410 PRINT INK 7;AT dy,dx;
    CHR$ 144;CHR$ 145;AT dy+
    1,dx;CHR$ 146;CHR$ 147
420 NEXT n
430 IF s>hs THEN LET hs=s
440 PRINT OVER 0;PAPER 3;AT
    0,8;s;AT 0,23;hs
450 PRINT OVER 0;PAPER 3;
    FLASH 1;AT 10,2;
    "PULSA UNA TECLA"
460 FOR n=1 TO 100:NEXT n
470 IF INKEY$="" THEN GOTO
    470
480 CLS:GO TO 125
    
```

Las líneas 120, 125 y 135 cuidan de las adiciones del tanteo. Se pone a cero la puntuación máxima y se presentan los valores de puntuación y puntuación máxima.

El bucle FOR ... NEXT que se extiende entre las líneas 150 y 420 presenta una serie de diez patos que el jugador debe intentar derribar. Cada vez que se recorre el bucle, en las líneas 160 y 170 se obtiene una nueva posición al azar del pato sobre la pantalla, y la línea 180 presenta los cuatro GDU que configuran el gráfico en esa posición.

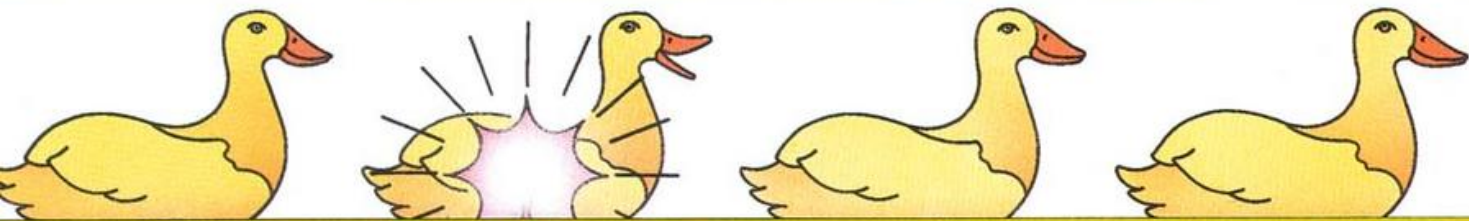
El tiempo se inicializa en la línea 190, que se utiliza además para tener un tiempo límite dentro del cual ha de ser derribado el pato.

Cuando se excede el tiempo límite, el programa presentará el pato en alguna otra parte.

Después de ejecutar el RETURN de la subrutina del joystick, la línea 205 borrará el primer carácter del marcador de puntuación. Esta es una precaución para que si la puntuación va disminuyendo, ninguna de las puntuaciones anteriores aparezca aún en pantalla.

El joystick va provisto de un pulsador de fuego y hay que examinar el valor de IN 31 para detectar los disparos. Cuando su valor es 16, significa que se ha apretado el gatillo.

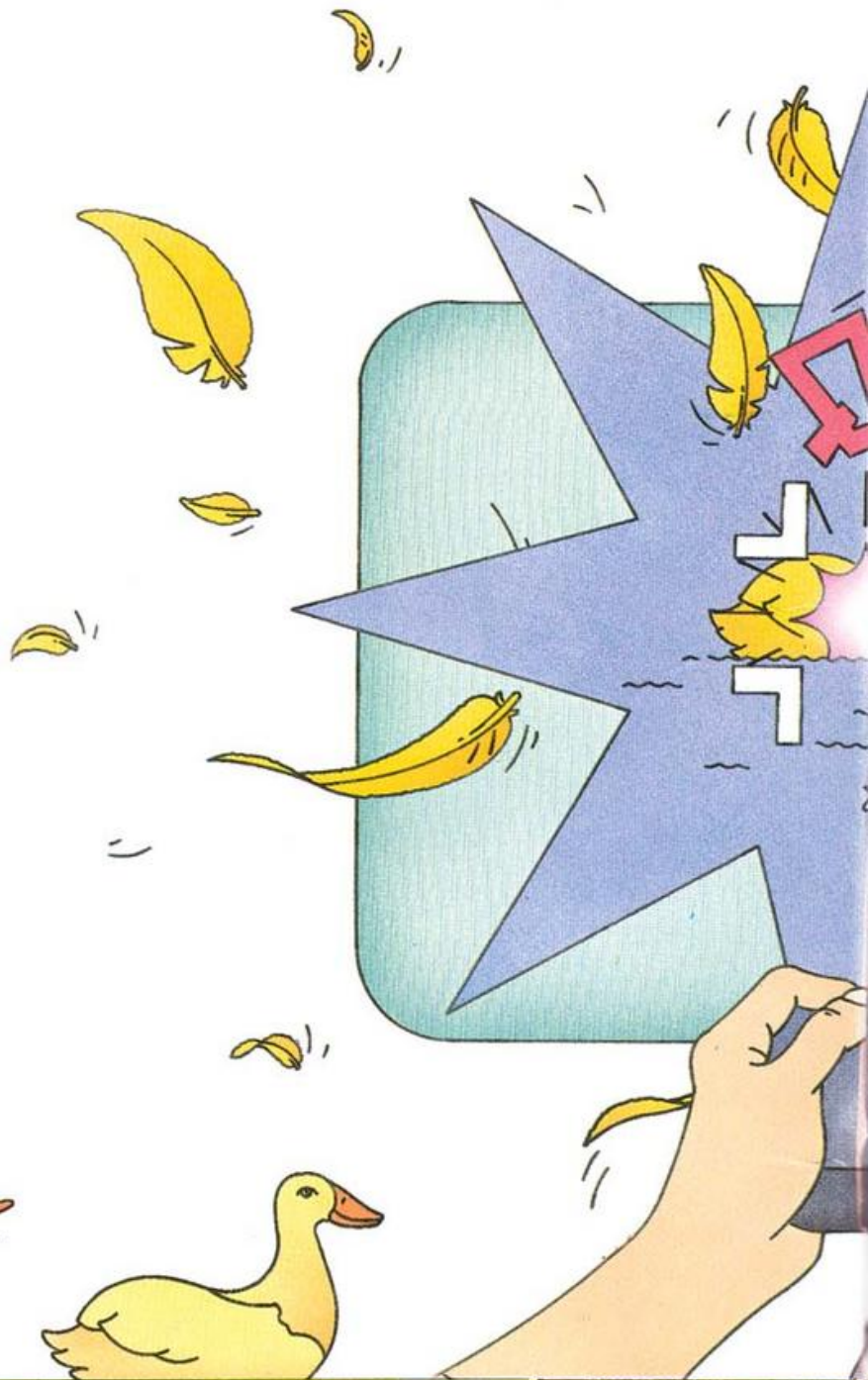




La línea 210 detecta los disparos, pero sólo cuando el punto de mira está quieto. Se podría comprobar si el jugador está disparando mientras el punto de mira se está moviendo, pero se necesitarían ocho comprobaciones, una para cada dirección de movimiento del *joystick*. Realmente esto no resulta factible en BASIC, ya que todas estas comprobaciones harían que el programa se volviera extraordinariamente lento.

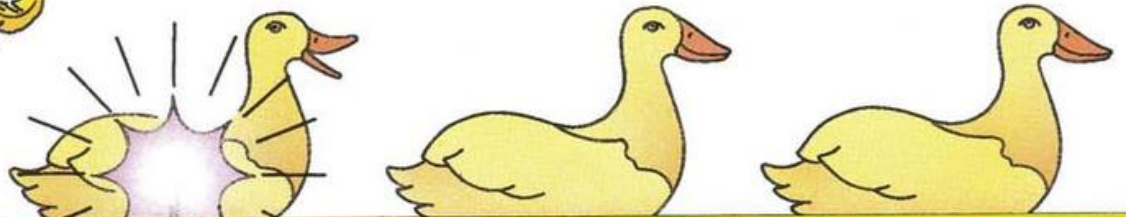
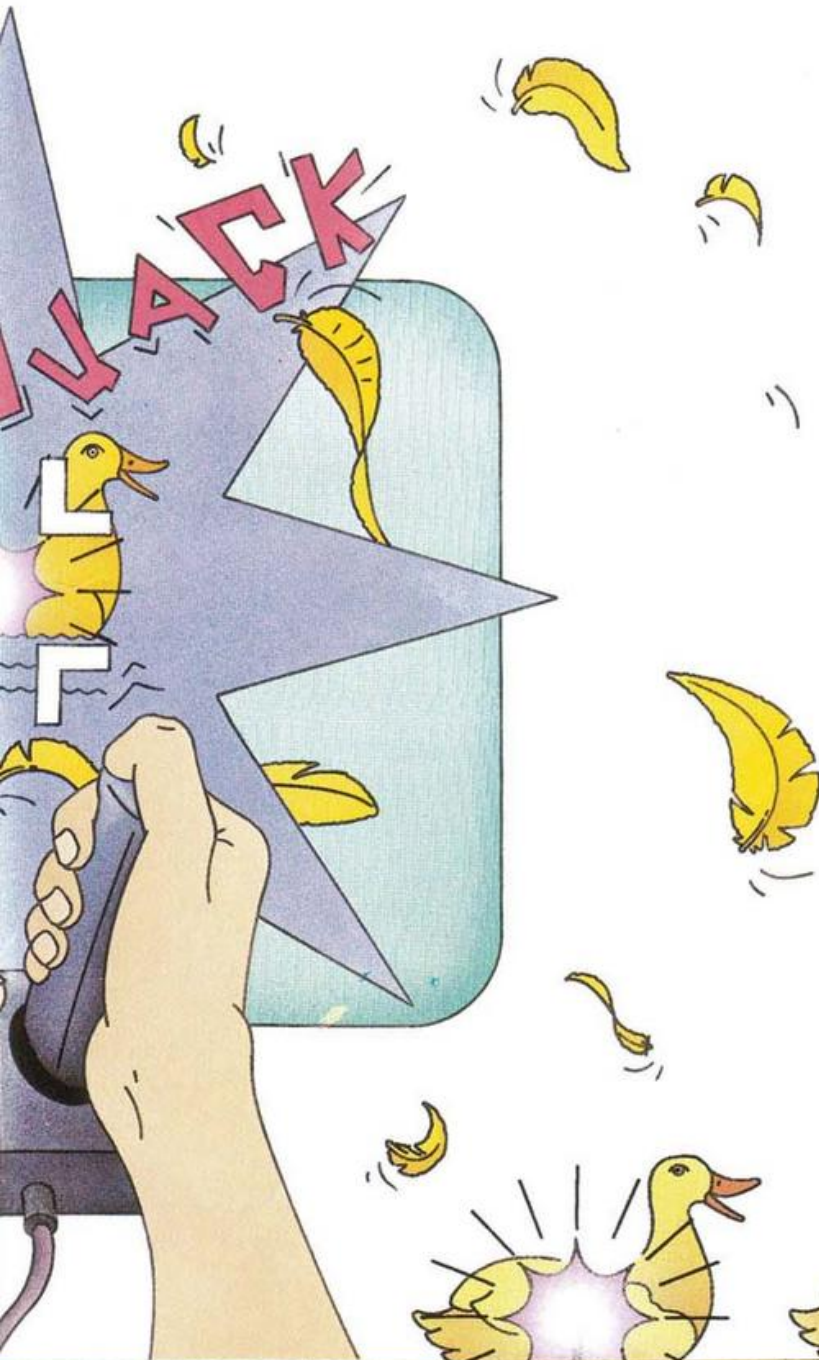
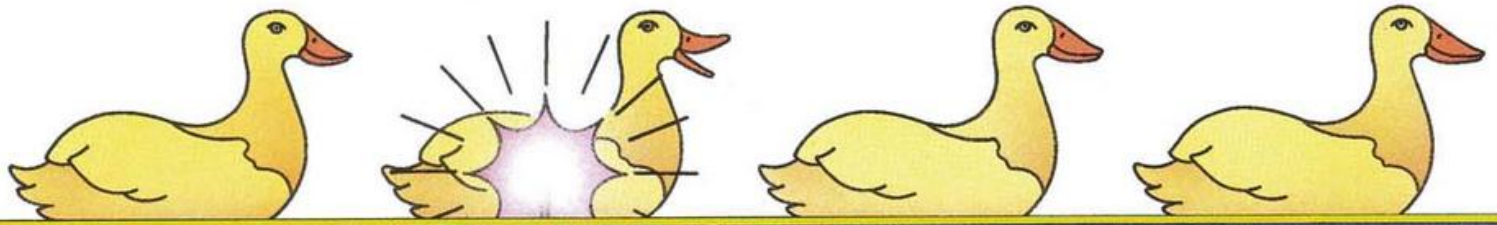
Si se ha pulsado la tecla de control, el programa pasa a comprobar si ha sido alcanzado el pato, para lo cual utiliza ATTR en la línea 220. Ya nos ocuparemos extensamente de ATTR más adelante; digamos de momento que sirve para ver el color de una determinada zona de la pantalla. Si el color es el mismo que el del pato, el programa lo cuenta como un impacto y se calcula el tanteo a partir del tiempo transcurrido para abatirlo. Además se produce un BEEP para indicar cada vez que se hace blanco.

Si el color obtenido mediante ATTR no corresponde con el color del pato, el programa continúa con la línea 230 en la que se produce una penalización de diez puntos que se restan del total. A continuación la línea 400 comprueba si ha transcurrido el tiempo límite, saltando a la línea 200





## PROGRAMACION DE JUEGOS



en caso de que no sea así. Si han transcurrido ya más de cuatro segundos, la línea 410 borra el pato de la pantalla.

Después de que hayan aparecido diez patos en la pantalla, el programa llega a la línea 430 en la que se compara la puntuación del jugador con la puntuación máxima. La puntuación máxima se modifica cuando el tanteo del jugador supera al récord existente. La puntuación máxima y la actual se presentan entonces en la línea 440.

El programa se completa por una rutina del tipo de ¿Quiere jugar otra vez? como suele ocurrir en este tipo de juegos.

### P y R

**¿Qué hay que hacer para cambiar el gráfico del pato por un blanco diferente?**

Básicamente se trata de alterar los DATA y de modificar las líneas que imprimen el GDU en la pantalla.

El programa utiliza un bloque de cuatro GDU definidos en las líneas 1.000 a 1.030. Si utilizas un GDU más grande, también tienes que modificar la línea 80 que los presenta en la pantalla.



# BARAJA Y REPARTE

■	DISPOSICION DE LAS CARTAS
■	DIBUJANDO LAS CARTAS
■	BARAJADO
■	REPARTO
■	USO DE LOS GDU

Los ordenadores pueden ser muy buenos jugadores de cartas si se programan correctamente; además nunca se aburren. Aquí tienes la forma de programar los gráficos de una baraja.

¿Te encuentras distanciado de tus amigos, parientes o colegas por haberles dejado sin un duro jugando a las cartas? ¿Eres tú el que estás sin blanca por haber jugado con expertos? Sea

como fuere, en los capítulos siguientes de nuestro coleccionable te presentamos la solución. Programando tu ordenador para que juegue contigo a las veintiuna, tendrás una víctima propiciatoria y una manera de jugar sin tener que vaciarle los bolsillos a nadie.

En esta primera parte nos ocuparemos de la manera de generar las rutinas gráficas con las que se construye la baraja. El resto del programa, que

es el juego propiamente dicho, se presentará en los dos capítulos siguientes. Pero no te olvides de almacenar cada una de las secciones en cinta a medida que vas construyendo el juego.

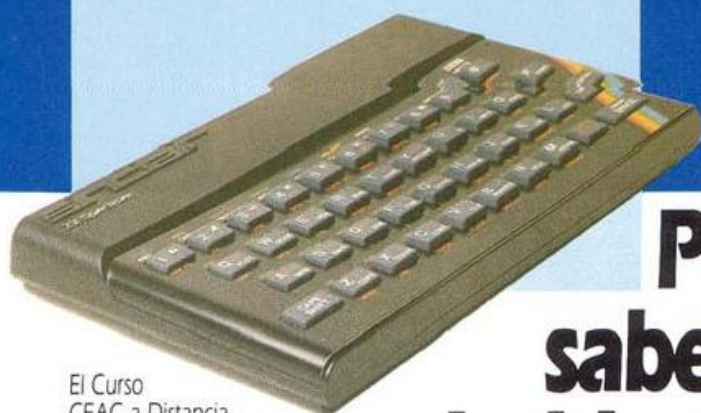
Si no eres un experto en el juego de las veintiuna, no te preocupes. En la última parte del programa presentaremos un conjunto completo de las reglas del juego. Pero antes debes ser capaz de programar un mazo de cartas.





# CURSO DE **BASIC** + MICROORDENADORES

prácticas con...



## Para saber cómo hablar con los ordenadores

El Curso  
CEAC a Distancia,  
BASIC + Microordenadores,  
le va a introducir paso  
a paso, con un cuidado  
método, en uno de los temas más  
apasionantes de nuestros días:  
**la programación de ordenadores.**

Al aprender PRACTICANDO desde un principio  
a programar BASIC, lenguaje diseñado  
especialmente para dar los primeros pasos  
en programación, estará sentando las bases  
para el estudio de cualquier otro  
lenguaje de alto nivel.

**Curso CEAC de BASIC + Microordenadores:**  
un diálogo permanente con el ordenador.

### Otros Cursos:

- Introducción a la Informática
- Electrónica (con experimentos)
- Contabilidad
- Fotografía
- Curso de Vídeo
- Decoración

## CEAC

CENTRO DE ENSEÑANZA A DISTANCIA  
AUTORIZADO POR EL MINISTERIO DE  
EDUCACIÓN Y CIENCIA N.º 8039185  
(BOLETÍN OFICIAL DEL ESTADO 3-6-83)  
Aragón, 472 (Dpto. M x L ) 08013 Barcelona  
Tel.: (93) 245 33 06

o llame...  
**(93) 245 33 06**  
de Barcelona



ESTAS ENSEÑANZAS SE AJUSTAN AL ART. 35  
DEL DECRETO 707/1976 Y A LA ORDEN MINISTERIAL DE 5/2/1979

Actúe ahora  
en su propio  
beneficio  
y pídasenos  
información.

**Sí,** deseo recibir a la mayor  
brevedad posible información  
sobre el Curso de: \_\_\_\_\_

**GRATUITAMENTE**

Nombre y apellidos \_\_\_\_\_ Edad \_\_\_\_\_

Domicilio \_\_\_\_\_

\_\_\_\_\_ N.º \_\_\_\_\_ Piso \_\_\_\_\_ Pta. \_\_\_\_\_ Tel. \_\_\_\_\_

C. Postal \_\_\_\_\_ Población \_\_\_\_\_

\_\_\_\_\_ Provincia \_\_\_\_\_

Profesión \_\_\_\_\_

**CEAC. Aragón, 472 (Dpto. M x L ) 08013 Barcelona**

889





# ¿MONITOR DE VIDEO O TELEVISOR?

■	COMO FUNCIONAN
■	TIPOS DE ENTRADAS DE SEÑAL
■	MONOCROMO Y COLOR
■	DEFINICION DE LA IMAGEN

Puede ser difícil realizar una elección entre un televisor de calidad y un monitor de video, pero en muchos casos no cabe más que una opción. En este artículo dedicamos nuestra atención a algunas de las diferencias generales entre los dos tipos.

La pantalla es un elemento vital de cualquier ordenador, aunque muchas veces se la considera como una especie de mula de carga en vez del sofisticado elemento electrónico que en realidad es. La pantalla es el principal

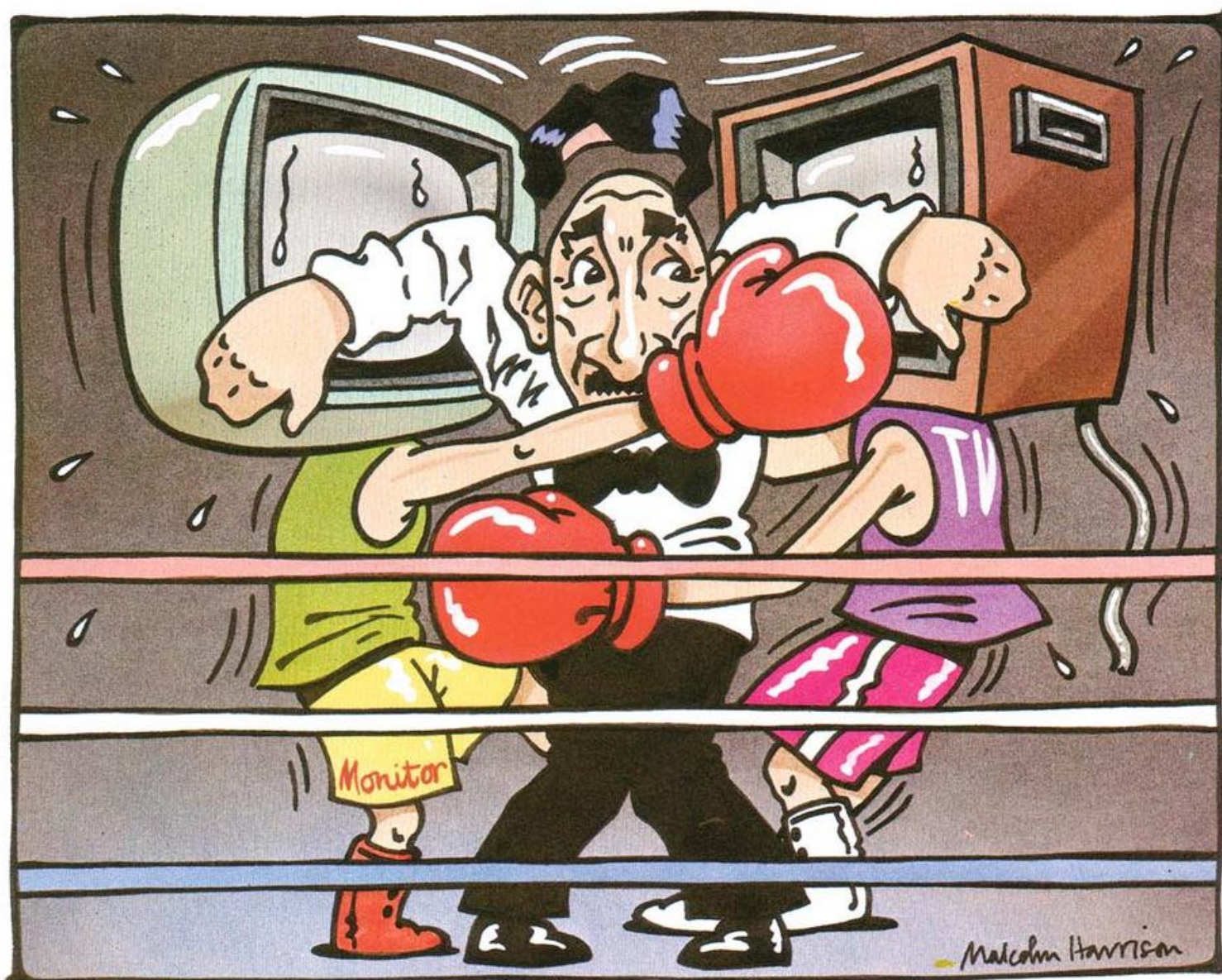
canal de comunicaciones entre el ordenador y nosotros. En el otro sentido la información fluye desde el usuario hacia el ordenador, normalmente a través del teclado, representando también la pantalla en este caso una réplica o eco de la información que se tecleó.

Si estás en una situación tan afortunada como para tener que escoger una nueva pantalla, o simplemente desees aprovechar lo mejor posible la que tienes, te resultará conveniente, si no esencial, tener algunas nociones sobre

la forma en que funciona una pantalla y la forma en que el ordenador controla la salida de video. En particular esto último es algo que afectará a tu elección.

## PRINCIPIOS BASICOS

Aunque hay varias formas diferentes de transferir la información desde el ordenador a la pantalla, una cosa es común a todos los tipos de pantallas utilizados en aplicaciones tanto co-





## PROTO® Joystick



P.V.P.  
RECOMENDADO  
(Ex I.V.A.)

**1.500** Pts.

Compatible con: SPECTRUM,  
ATARI, COMMODORE,  
SPECTRAVIDEO,  
AMSTRAD, CANON  
y OTROS

**AHORA  
CON TIRO DE RAFAGA**

## PROTO® CABLES

- **PROTO Cable Centronics AMSTRAD**  
Conecta al ordenador una impresora centronics  
3.200 Ptas.
- **PROTO Cable 2ª Unidad de Disco AMSTRAD**  
Conecta al ordenador la 2ª Und. de Disco  
2.000 Ptas.
- **PROTO Cable Audio M.S.X.**  
Conecta Magnetófono a ordenad. M.S.X.  
950 Ptas.
- **PROTO Cable Audio AMSTRAD**  
Conecta Magnetófono a ord. AMSTRAD  
950 Ptas.
- **PROTO Set de Cables Prolongadores AMSTRAD 8256**  
Prolongador impresora y prolongador alimentación  
3.750 Ptas.
- **PROTO Set de Cables Prolongadores AMSTRAD 464**  
Prolongador alimentación y prolongador monitor  
1.600 Ptas.
- **PROTO Set de Cables Prolongadores AMSTRAD 664-6128**  
Dos prolongadores alimentación y prolongador monitor  
2.300 Ptas.

Precios Ex IVA



**PROTOMECA**, S.A.

Avda. de la Constitución, 260 - Telf. 675 78 54 - TORREJON DE ARDOZ (Madrid)

## SERVICIO TECNICO DE REPARACION DE ORDENADORES

### REPARAMOS

AMSTRAD TODOS LOS MODELOS  
SPECTRUM  
COMMODORE  
TEXAS INSTRUMENTS  
SPECTRAVIDEO





merciales como industriales o domésticas: el tubo de rayos catódicos, también conocido por sus iniciales inglesas CRT. La base del funcionamiento de los tubos de rayos catódicos reside en el hecho de que ciertos compuestos de fósforo emiten luz cuando son excitados por un haz de electrones.

Si se dispara un chorro de electrones con un cañón electrónico sobre una lámina de vidrio cubierta con fósforo, cada vez que se produzca un disparo el fósforo brillará. Si se desplaza el cañón de forma que el chorro de electrones vaya recorriendo toda la pantalla, será toda la superficie de la misma lo que emitirá brillo al estar sometida al bombardeo de electrones. Si se hace que el haz de electrones se mueva, por ejemplo, describiendo la figura de un ocho, aparecerá en la pantalla la figura de un ocho.

Un tubo de rayos catódicos es un cono de vidrio cerrado en cuyo interior se ha hecho el vacío, provisto de un cañón electrónico en el extremo próximo al vértice y con su base cubierta de fósforo.

Imaginemos un trozo de cartulina en el que se ha recortado la forma de un hombre, interponiéndose entre el cañón de electrones y la pantalla. La pantalla sólo brillará en los puntos en que el chorro de electrones no haya sido bloqueado por la cartulina. En consecuencia, la zona de pantalla cuyo fósforo brilla tendrá una forma parecida a la de un hombre. Se podría conseguir el mismo efecto interrumpiendo el haz electrónico (apagando el cañón) en los instantes adecuados, es decir cuando pasa por los puntos deseados de la pantalla.

Esta es exactamente la forma en que funcionan los televisores y la mayoría de los monitores, a excepción de algunos de construcción especial. Un cañón electrónico barre todo el ancho de la pantalla 625 veces para crear 625 líneas. (En realidad barre primero 313 líneas y a continuación rellena los 312 huecos que había dejado entre dichas líneas.)

El sistema no tiene partes mecánicas móviles, el desplazamiento del haz de electrones se consigue por medio de campos magnéticos. Por eso está

recomendado no dejar materiales magnéticos —*diskettes* o cintas— cerca de un televisor. Para formar las imágenes, se conmuta el cañón de electrones encendiéndolo y apagándolo. La información que contiene la señal recibida por el televisor o el monitor, precisamente sirve para decirle al cañón de electrones cuándo tiene que emitirlos o no.

En lugar de recorrer toda la pantalla en lo que se llama un barrido por rastreo, algunos monitores producen la imagen guiando el haz de electrones por la pantalla como si fuera un lápiz, siguiendo las líneas de la imagen. Aunque se consiguen así resultados de excelente calidad, se trata de un procedimiento mucho más lento para usos generales. Aún suponiendo que el haz de electrones se desplazara por la pantalla a la nada despreciable velocidad de 40000 kilómetros por hora, una imagen de cierta complejidad aparecería ante el ojo humano como altamente insatisfactoria.

## PERSISTENCIA DE LA VISION

Así se llama una de las características del ojo humano que en otras circunstancias podría parecer un defecto, pero que en este caso permite la utilización del tubo de rayos catódicos como instrumento de comunicación visual.

El ojo humano tiene la propiedad de mantener cualquier imagen de las que se forman sobre su retina durante un tiempo de aproximadamente 1/25 de segundo. El cañón de electrones que hay en la parte trasera de los tubos de rayos catódicos de los televisores domésticos y de la mayoría de los monitores, barre la superficie de la pantalla 625 veces en 1/50 de segundo. Esto significa que se forma una nueva semi-imagen o «trama» sobre la pantalla cincuenta veces por segundo. De esta forma aparece sobre la pantalla una imagen nueva antes de que se haya desvanecido la antigua de nuestro ojo. Por eso podemos ver una imagen que se mueve con suavidad.

Si nuestro ojo fuera capaz de retener la imagen solamente durante una

centésima de segundo, la imagen de la pantalla nos parecería que parpadeaba de forma insoportable.

## COLOR

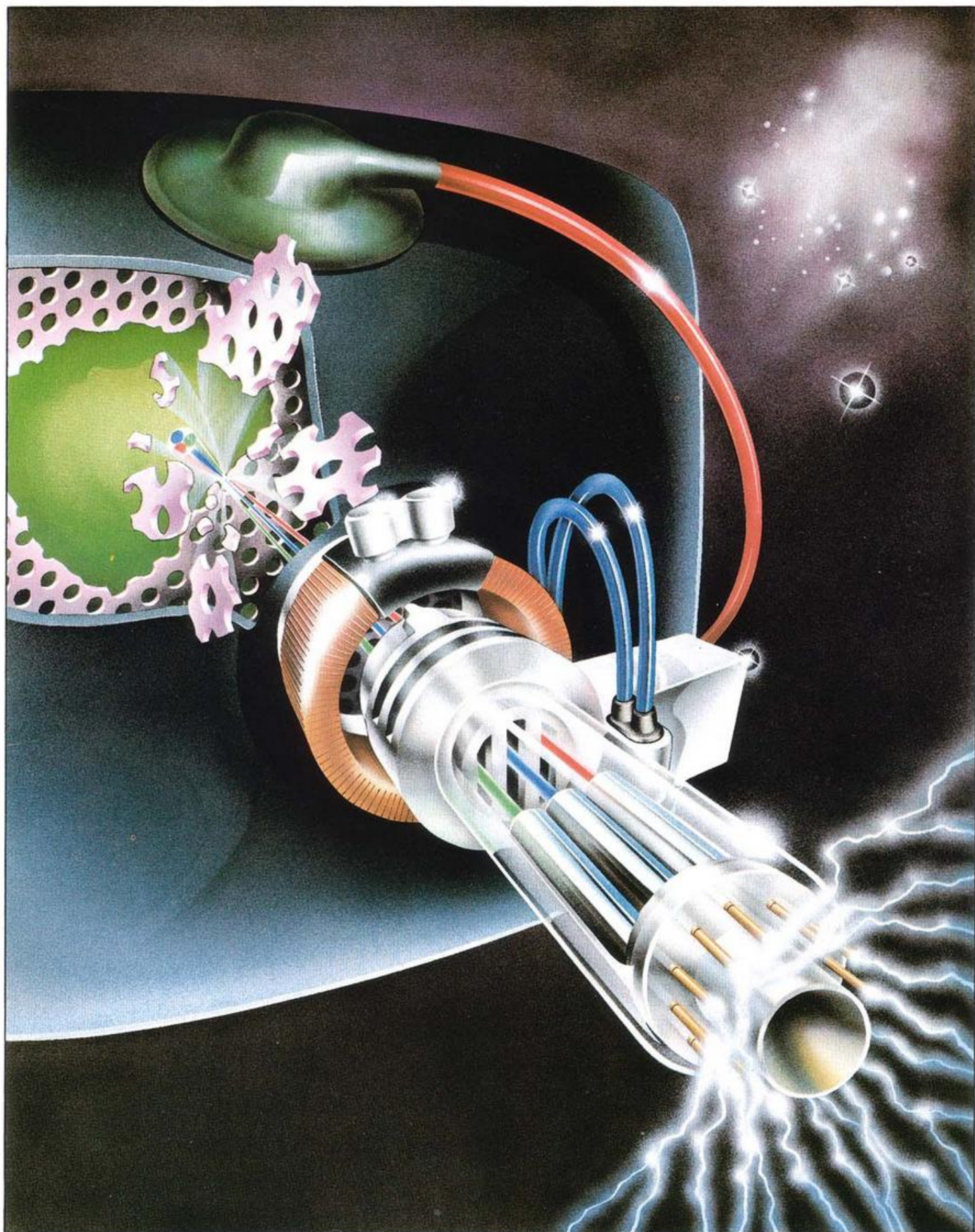
El principio básico empleado es el mismo para televisores en blanco negro y color. La única diferencia entre ambos es que sólo se requiere un cañón de electrones y un tipo de recubrimiento de fósforo en el caso del monocromo, mientras que en el caso del color hacen falta tres tipos de fósforos diferentes en el recubrimiento de la pantalla y normalmente tres cañones electrónicos (hay televisores que tienen un solo cañón, pero que emiten tres haces de electrones).

El color se debe a que uno de los fósforos del recubrimiento produce luz roja, otro luz verde y el otro luz azul. Uno de los chorros de electrones activa el fósforo rojo, otro el fósforo verde y el tercero el fósforo azul. Combinando el rojo, verde y azul con diferentes intensidades se pueden obtener todos los demás colores e intensidades.

## ENTRADA DE LA SEÑAL

Las pantallas, ya sean de un televisor o de un monitor, de color o de blanco y negro, pueden aceptar tres clases de señales de entrada. En primer lugar está la entrada normal de televisión enviada por las emisoras públicas, que es la señal que acepta el televisor de tu casa y produce en la pantalla las imágenes de televisión que estás acostumbrado a ver. Por otra parte está la llamada señal compuesta de videofrecuencia o vídeo compuesto. Esta es una de las señales que se emplea con más frecuencia en los monitores de vídeo y en la actualidad ya hay algunos televisores que tienen prevista una toma para aceptar dichas señales. La señal de vídeo compuesto contiene la información de la imagen y los impulsos de sincronismo. Esta última hace que el haz de electrones se sitúe en el lugar adecuado en cada momento.







# Quick

## Los Joysticks más

QUICKSHOT IV (3 en 1)  
Con mando de carreras

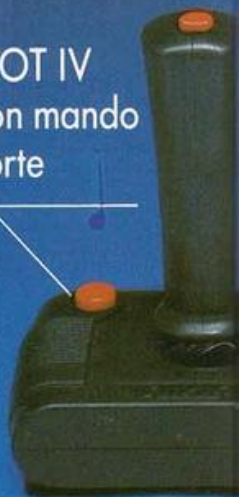


QUICKSHOT I



QUICKSHOT I MSX

QUICKSHOT IV  
(3 en 1) Con mando  
para deporte



QUICKSHOT VII - Portátil



QUICKSHOT IX  
Preciso y sensible

Los QUICKSHOT comercializados por SVI-España, S. A. son los únicos que tienen la GARANTIA OFICIAL SVI.



# Quickshot®

*vendidos del mundo.*



QUICKSHOT II MSX  
Con autodisparo



QUICKSHOT IV (3 en 1)  
Con mando para combate



QUICKSHOT II  
Con autodisparo



QUICKSHOT VII MSX  
Portátil

Importador exclusivo SVI-España.

**SVI**  
SPECTRAVIDEO



Por último, está la llamada entrada RGB. Las letras corresponden a rojo, verde y azul (Red, Green, Blue). Es la forma más directa y más precisa de especificar la información de color. La información relativa a cada color se introduce directamente y por separado desde el ordenador al monitor.

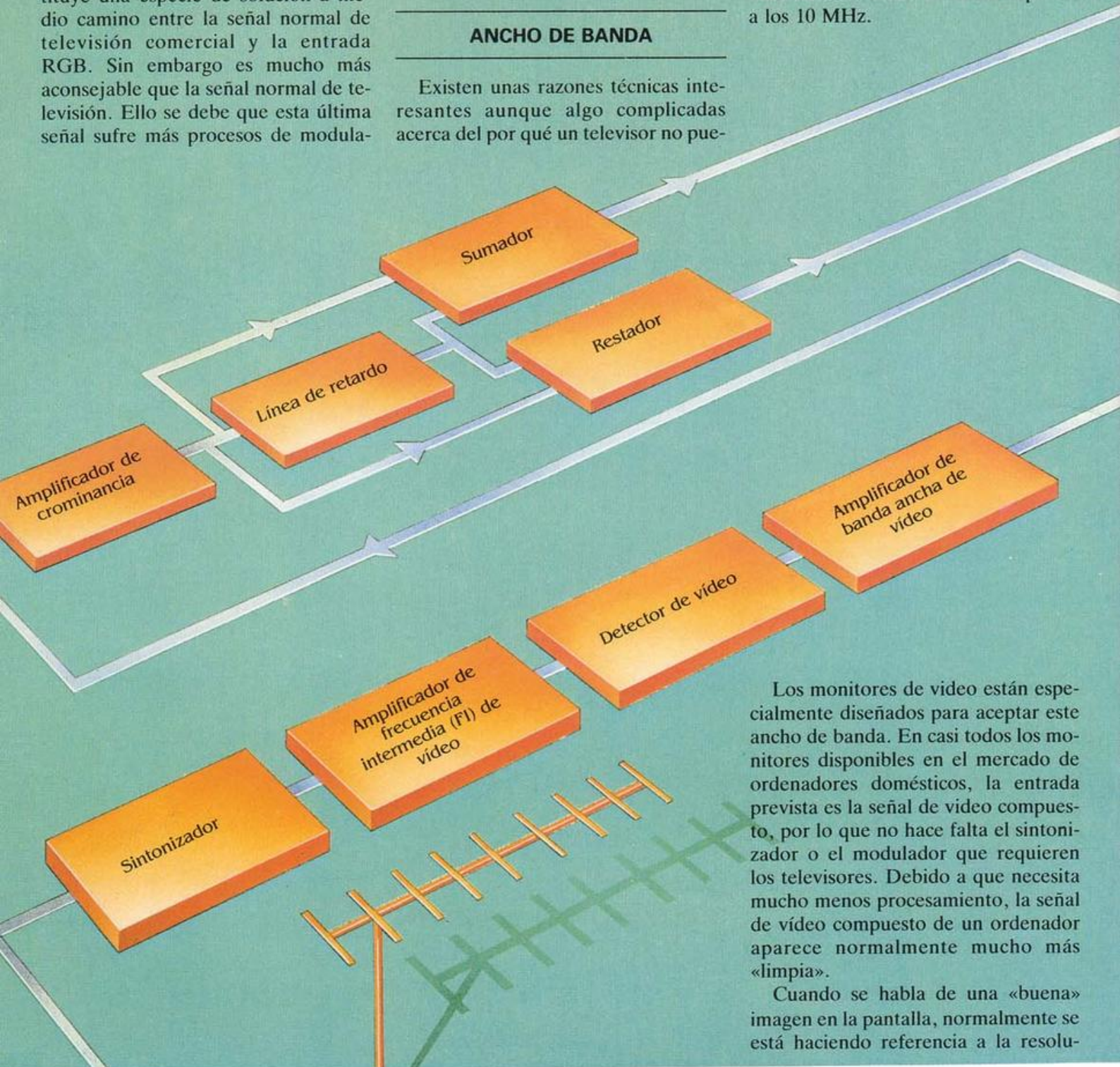
La señal compuesta de video constituye una especie de solución a medio camino entre la señal normal de televisión comercial y la entrada RGB. Sin embargo es mucho más aconsejable que la señal normal de televisión. Ello se debe que esta última señal sufre más procesos de modula-

ción, conversión de frecuencia y amplificación hasta que se convierte en una imagen y es evidente que cuantos más procesos sufre una señal, más probable es que se vea contaminada por el ruido y la distorsión. En el modo RGB, la señal de sincronismo va por separado (en otra patilla del conector).

## ANCHO DE BANDA

Existen unas razones técnicas interesantes aunque algo complicadas acerca del por qué un televisor no pue-

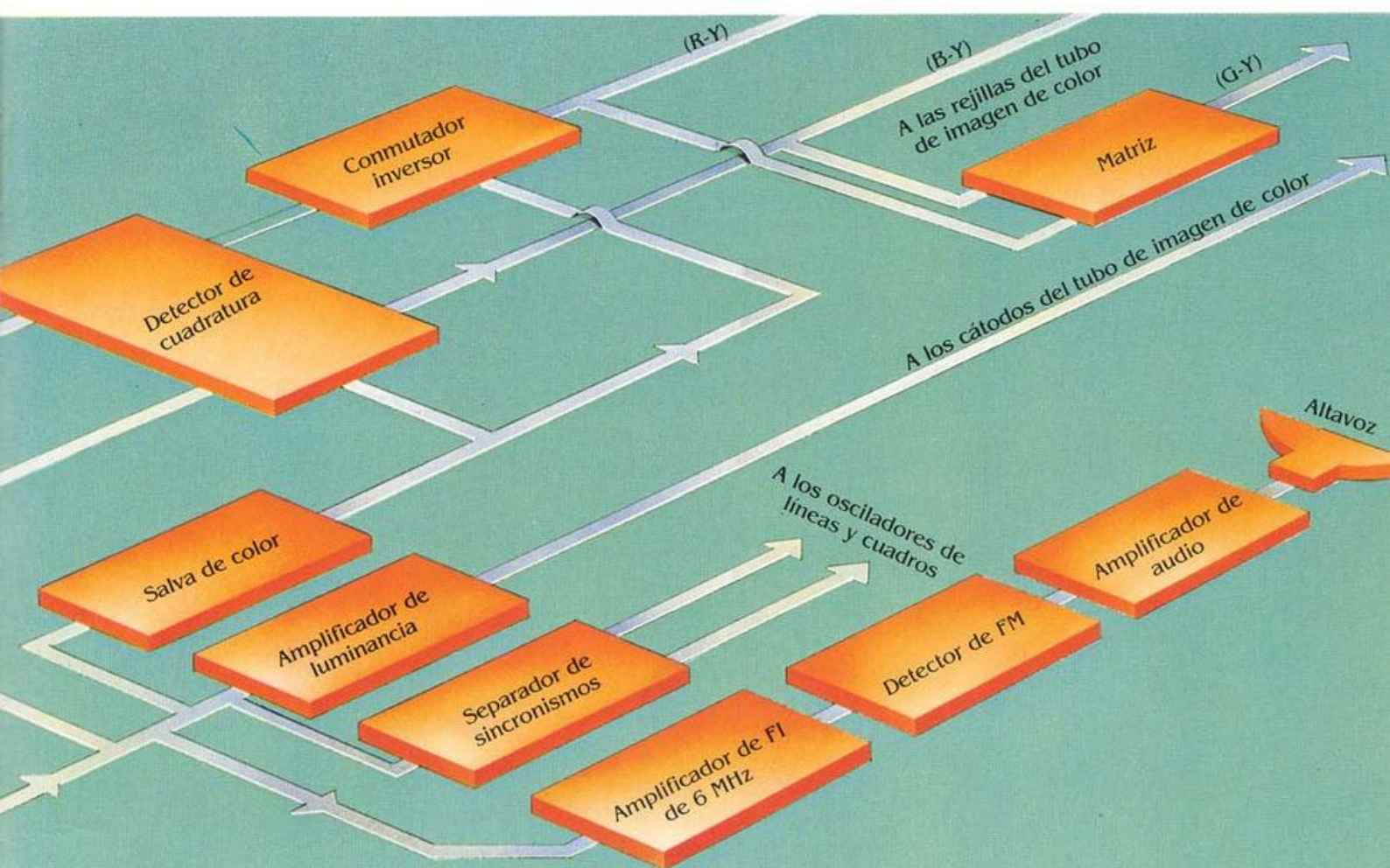
de dar tan buena resolución como un monitor. Por ejemplo, un televisor no puede manejar información que requiera un ancho de banda en frecuencia superior a los 5,5 MHz. Con este ancho de banda se puede conseguir una presentación en pantalla de 40 columnas, pero para una presentación de 80 columnas que sea satisfactoria hace falta un ancho de banda superior a los 10 MHz.



Los monitores de video están especialmente diseñados para aceptar este ancho de banda. En casi todos los monitores disponibles en el mercado de ordenadores domésticos, la entrada prevista es la señal de video compuesto, por lo que no hace falta el sintonizador o el modulador que requieren los televisores. Debido a que necesita mucho menos procesamiento, la señal de video compuesto de un ordenador aparece normalmente mucho más «limpia».

Cuando se habla de una «buena» imagen en la pantalla, normalmente se está haciendo referencia a la resolu-





ción y al color. Por desgracia ambas cosas no siempre son compatibles. Una pantalla de blanco y negro siempre tendrá una mejor definición de imagen que una pantalla en color. En una pantalla monocroma sólo se requiere un punto de fósforo para definir un punto de la imagen, mientras que se requieren tres puntos del mismo tamaño para definir ese mismo punto en una pantalla de color.

En una pantalla normal de cualquier clase hay unos 360.000 puntos de fósforo. En el caso de una pantalla de color se requerirían 120.000 de dichos puntos para el rojo, 120.000 para el verde y 120.000 para el azul. Esto significa que sólo hay disponibles 120.000 grupos de tres puntos para formar la imagen en una pantalla de color. Por el contrario una pantalla monocroma tiene 360.000 puntos disponibles para crear una imagen.

Todo el que haya trabajado con am-

bos tipos de pantalla, la monocroma y la de color, reconocerá inmediatamente la diferencia. El monitor en blanco y negro es mucho más nítido y claro que el de color, el cual presenta una imagen comparativamente más borrosa.

En muchas aplicaciones especializadas la elección natural será el monitor monocromo. El color normalmente elegido para estas aplicaciones —diseño asistido por ordenador o procesamiento de textos, por ejemplo— es el verde. Está considerado como el color más fácil para la vista y el menos cansado.

En cuanto a su peligrosidad para la salud, la inmensa mayoría de los informes que han sido publicados sobre el tema, declaran que el tubo de rayos catódicos es absolutamente inofensivo para los ojos. Normalmente se suelen oír quejas debidas a otras condiciones de trabajo, a colores de pantalla ina-

decuados, reverberaciones de la pantalla, distracción debida a otros focos luminosos y a veces efectos psicológicos. Sin embargo la pantalla de un televisor presenta un ligero centelleo y además suele estar hecha de un material reflectante que con frecuencia hace aparecer imágenes indeseadas y causantes de distracción. Esta es una de las razones por las que el uso de un televisor para un trabajo serio, que requiera una gran concentración durante largos períodos de tiempo, puede causar jaquecas.

Para producir una imagen con la misma resolución que las de la televisión o el vídeo ordinarios, nuestros ordenadores tendrían que controlar individualmente cada punto de fósforo de la pantalla. Y suponiendo que cada *pixel* necesite un byte de memoria, se necesitaría medio Megabyte para controlar toda la pantalla. Obviamente esto está fuera de lugar aquí, ya que



nos estamos ocupando de pequeños ordenadores domésticos y esa cantidad de memoria representa un tremendo gasto incluso en aplicaciones de ordenadores más grandes.

La mayoría de los micros domésticos disponen de aproximadamente 1K de memoria para dividir la pantalla entre 800 y 1000 cuadrados, típicamente 24 líneas por 40 columnas. Otros ordenadores te permiten elegir entre baja resolución y alta resolución, dependiendo de la forma de presentación seleccionada. Hay que observar sin embargo que esto no tiene nada que ver con la pantalla que utilices. Las limitaciones en la definición y la resolución de la pantalla vienen impuestas por el ordenador y no por el tubo de rayos catódicos.

Cada uno de estos aproximadamente 900 cuadrados es una casilla individual que puede contener un carácter. El ordenador contiene un conjunto de caracteres en ROM en el que corresponde un número del código ASCII para cada carácter. Cada carácter ocupa un cuadrado al ser presentado en la pantalla. De esta forma se puede almacenar un carácter en un solo byte, consiguiéndose un enorme ahorro de memoria.

## HACIENDO UNA ELECCION

Los dos factores más importantes a tener en cuenta a la hora de comprar una pantalla nueva son el tipo de ordenador y las futuras aplicaciones. A pesar de ser un elemento importante dentro del sistema, la pantalla siempre ha de estar supeditada al ordenador. Si las capacidades gráficas del ordenador son pobres, seguirán siéndolo independientemente de lo buena que sea la pantalla. Siempre es el ordenador el que controla la pantalla y es importante que ésta se compre con arreglo a las características del ordenador.

También es importante tener en cuenta el uso a que se va a destinar el ordenador. En las aplicaciones en las que nunca va a ser necesario el color, la elección obligada es un monitor de blanco y negro. Si es esencial una ima-

gen de alta calidad en colores, se requerirá un monitor con entrada RGB, si bien incluso para la mayoría de los usuarios profesionales de micros bastará con una entrada de vídeo compuesto. Para todos los poseedores de micros, con excepción de los de más alto precio, la elección entre un monitor de entrada RGB o uno de entrada de vídeo compuesto tendrá un interés puramente académico ya que se construyen muy pocos micros con salida RGB.

Para los juegos, la programación «recreativa» en general hecha en casa y otros usos domésticos de los ordenadores, bastará con un televisor ordinario. Aunque muchos juegos resultarían mucho mejor sobre pantallas de alta resolución, otros han sido escritos con conocimiento pleno de que sus gráficos sólo se iban a representar en un televisor ordinario. Ten presente además que debido a que los monitores se fabrican principalmente para aplicaciones comerciales, es frecuente que no incluyan características tales como el sonido.

Además, aunque su electrónica con frecuencia es más sencilla que la de un televisor, los monitores no son necesariamente más baratos que los televisores de un tamaño semejante. En lo que se refiere a pantallas, lo más grande no es necesariamente lo mejor. En muchos casos incluso podría ser mucho peor. Al aumentar o disminuir las dimensiones de la pantalla, lo que ocurre es que también aumenta o disminuye el tamaño de los *pixels* individuales. Si por ejemplo el ordenador produce 960 *pixels*, ése será el número de ellos que aparezcan en pantalla con independencia de lo grande o pequeña que ésta sea. Normalmente no se produce un aumento de la cantidad de información disponible en la pantalla al aumentar el tamaño de la misma, y en muchos casos lo que sucede es que una pantalla grande hace que ciertos contornos sean aún más destacados. Por cierto, recuerda que el tamaño de una pantalla se mide a lo largo de la diagonal de la misma, no según su anchura o su altura.

No se recomiendan los televisores viejos o los de muy bajo precio. Mu-

chas veces lo que no parecería en absoluto un problema serio en una imagen de un programa emitido por una emisora de televisión comercial, puede convertirse en algo desastroso cuando el televisor se utiliza con un ordenador.

Por ejemplo el sobrebarrido, que recorta los lados de la parte superior e inferior de la pantalla, puede no parecer demasiado serio en la presentación de imágenes televisivas, pero podría ocurrir que en el caso de utilizar un ordenador no apareciera en pantalla alguna información útil. Otro defecto frecuente de los televisores viejos es el que ocurre cuando un color se presenta con mucho más intensidad que los otros, produciendo un efecto de emborronamiento que puede hacer que las letras de la pantalla resulten muy difíciles de leer. Las reparaciones para eliminar este tipo de defectos suelen ser normalmente muy caras. Con frecuencia es más barato comprar un televisor mejor... o un monitor.

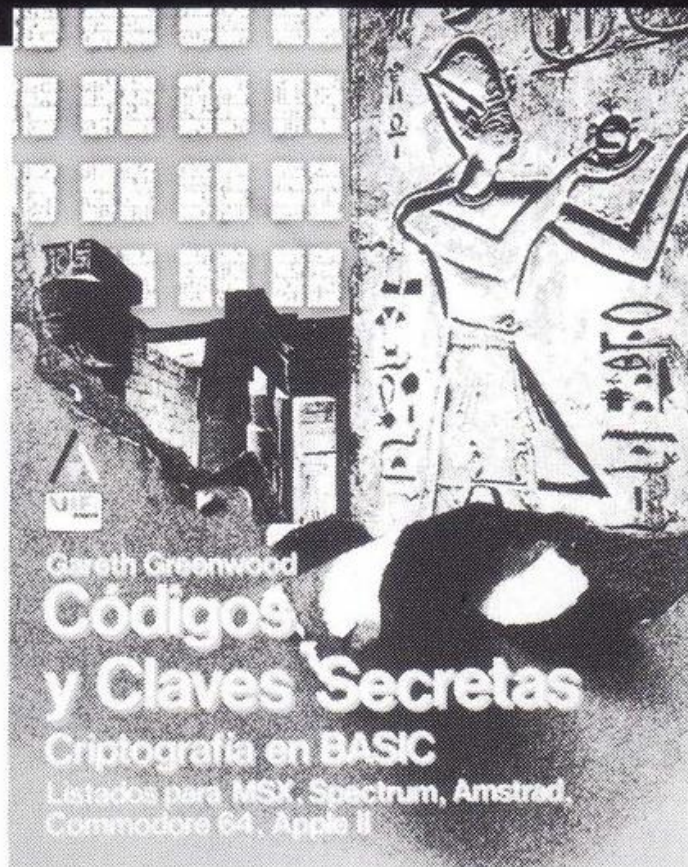
## CARACTERISTICAS DE LA VISUALIZACION

Tanto en el caso de un televisor como si tienes un monitor de video, puedes hacer varias cosas para mejorar las condiciones de visualización:

Puedes reducir la fatiga de la vista utilizando caracteres con colores claros contra un fondo negro. Para mejorar la definición utiliza los controles de brillo y contraste hasta lograr el ajuste óptimo. Los halos de color se pueden eliminar normalmente bajando la señal de color o eliminándola del todo; al fin y al cabo no necesitas el color para el procesamiento de textos.

Intenta situar tu mesa de trabajo, o el televisor o monitor, de forma que la pantalla no refleje la luz que entra por la ventana o la de alguna lámpara. En algunos casos puede ayudar el llevar ropas de tonos oscuros. Puedes minimizar aún más los efectos de las reflexiones teniendo la pantalla de forma que tengas que mirar ligeramente hacia abajo en vez de tenerla justamente enfrente de tus ojos o un poco más alta.





**Ordena  
tus propias ideas.**  
Le sacarás partido a tu ordenador

**ANAYA**  
MULTIMEDIA

Adquiéralos en su librería habitual. Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a: *Apdo. de Correos 14632*  
Ref. D. de C.  
28080 MADRID

- ☐ Les ruego me envíen el catálogo de su editorial.  
☐ Les ruego me envíen los siguientes títulos:

\_\_\_\_\_

TOTAL \_\_\_\_\_

- ☐ Adjunto talón bancario a  
GRUPO DISTRIBUIDOR EDITORIAL, S. A.  
☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).  
☐ Giro postal.

Nombre \_\_\_\_\_

Profesión \_\_\_\_\_

Dirección \_\_\_\_\_

C. P. \_\_\_\_\_ Localidad \_\_\_\_\_

Provincia \_\_\_\_\_

IS

### Códigos y Claves Secretas: Criptografía BASIC

Gareth Greenwood

\* 1.378 ptas.

La criptografía y el criptoanálisis ha jugado papeles muy importantes en muchos acontecimientos políticos y diplomáticos de la historia.

CODIGOS Y CLAVES SECRETAS: CRIPTOGRAFIA EN BASIC es una introducción a la criptología para usuarios de microordenadores. A lo largo del libro se describen y analizan las distintas técnicas de cifrado y descifrado de mensajes y se acompañan los programas en BASIC para que el ordenador realice el trabajo repetitivo a partir del ingenio, intuición y destreza del "rompeclaves".

Con tu ingenio y este libro aprenderás técnicas útiles, te divertirás, desarrollarás tu mente y, quién sabe, ¡tal vez descubras algo que antes no sabías!

El libro contiene versiones completas de todos los programas para SPECTRUM, AMSTRAD, MSX, COMMODORE y APPLE II.

### Simulaciones: Replica la realidad con tu ordenador

Tim Hartnell

\* 1.643 ptas.

Cómo desarrollar y disfrutar con las simulaciones por ordenador. Contiene 18 programas de simulación, listos para funcionar, que incluyen: un simulador de vuelo, una tortuga robot y la máquina de ajedrez de Torres Quevedo.

Incluye versiones para SPECTRUM, AMSTRAD, MSX, COMMODORE y APPLE II

### Sistemas Expertos: Introducción al diseño y aplicaciones

Tim Hartnell

\* 2.120 ptas.

Explora el fascinante mundo de los Sistemas Expertos en tu microordenador. Entre otros programas, este libro contiene dos interesantes intérpretes de PROLOG y LISP realizados en BASIC.

Incluye versiones para SPECTRUM, AMSTRAD, MSX, COMMODORE y APPLE II

Otros títulos:

### Tu primer libro del ZX SPECTRUM

J. Dewhirst y R. Tennison

\* 848 ptas.

### Diseño de Gráficos y Videojuegos.

(Tratamiento en tres dimensiones) (incluye cassette)

Ian O. Angell y Brian J. Jones

\* 3.392 ptas.

### "Sprites" y gráficos en lenguaje máquina. (ZX Spectrum)

John Durst

\* 1.537 ptas.

### Astronomía: el universo en tu ordenador.

Maurice Gavin

\* 1.378 ptas.

### Inteligencia artificial: conceptos programas

Tim Hartnell

\* 1.484 ptas.

### El libro gigante los juegos para ZX Spectrum.

Tim Hartnell

\* 1.431 ptas.

### Juegos gráficos de aventura para ZX Spectrum.

Richard Hurley

\* 1.484 ptas. \* IVA incluido



ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA



# ¡DESCUBRE AL ASESINO!

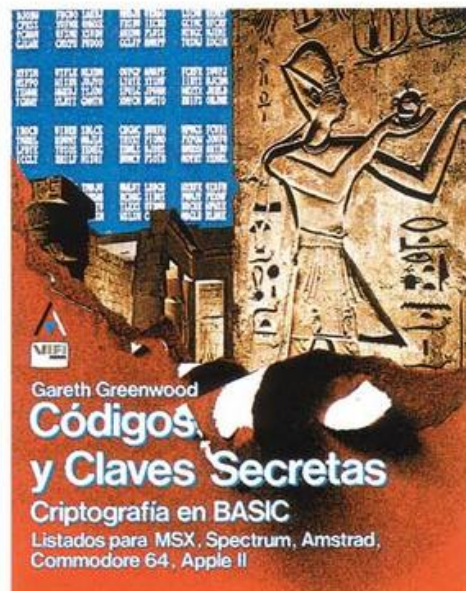
## BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

La solución del primer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de junio.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.



*Nota: El libro de reciente aparición «Códigos y claves secretas», de Anaya, contiene todos los programas útiles para esta labor. Aunque si has trabajado para los servicios de inteligencia de alguna potencia no te será de gran utilidad.*



## EL CASO DEL ANCIANO ASESINADO

La muerte es una forma de desorden; cae sobre la vida como el hacha sobre el tronco y quiebra siempre la sintaxis que después es preciso recomponer con dolor, a veces con recuerdos y, casi siempre, con notables dosis de olvido.

En otras ocasiones, la muerte es una estupidez o el resultado de varias estupideces hábilmente engarzadas por el azar o por la ambición. Investigando este tipo de muerte me gano la vida.

Aquella tarde, la primavera se había desplomado sobre las calles con una bola de fuego en sus entrañas. Me llamaron por teléfono y tuve que atravesar la ciudad en un momento especialmente difícil. Los conductores, todavía con sus ropas de invierno, se asaban de calor dentro de sus coches, lo que les ponía particularmente agresivos. Los semáforos, por el contrario, parecían haber sido tratados con tranquilizantes y pasaban del rojo al ámbar y del ámbar al verde con una parsimonia que desesperaba todavía más a los sufridos conductores.

Al fin llegué a una zona residencial y subí a un piso de cinco estrellas que acababan de abandonar los bomberos. Mi ayudante ya estaba allí, intentando preservar las escasas huellas que se habían resistido a la violencia de éstos. El incendio había sido de escasa magnitud, pero había bastado para asfixiar a un pobre viejo que permanecía atado a una de las sillas del salón. Una caja fuerte, hasta entonces oculta por un cuadro, aparecía abierta y desvalijada. Los bordes de la puerta de acero presentaban señales de violencia. Anoté en uno de mis códigos cifrados:

QE GZM HUAXQZOOM RUZSUPM BGQE EQ FDMFM P  
Q GZ FUBA PQ OMVM UZMOOQEUNXQ M OGMXCGUQ  
D FUBA PQ BMXMZOM

Al parecer, el incendio no había sido intencionado. Según los expertos, el origen fue una colilla que, tras resbalar del cenicero, había ido a caer al pie de una cortina originando un incendio de pequeñas dimensiones, pero con un humo lo suficientemente espeso para asfixiar a un anciano y hacer llegar a los bomberos.

Juan José Millas



# INPUT

sinclair

SERVICIO DE  
EJEMPLARES  
ATRASADOS

## ¡NO TE PIERDAS NI UN SOLO EJEMPLAR!

**INPUT SINCLAIR** quiere proporcionar a sus lectores este nuevo servicio de ejemplares atrasados para que no pierdan la oportunidad de tener en sus hogares todos los ejemplares de esta revista, líder en el mercado español.

Podréis solicitar cualquier número de

**INPUT SINCLAIR** que queráis, siempre al precio de cubierta (sin más gastos).

Utiliza el cupón adjunto, enviándolo a **EDISA** (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid, o bien llámanos por teléfono al (91) 415 97 12.



INPUT  
sinclair

siempre a  
tu servicio

### CUPON DE PEDIDO

SI, envíenme contrarreembolso ..... ejemplares de **INPUT SINCLAIR** de los números:

(marca con una (X) tu elección)

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

NOMBRE \_\_\_\_\_

APELLIDOS \_\_\_\_\_

DOMICILIO \_\_\_\_\_

NUM. \_\_\_\_\_ PISO \_\_\_\_\_ ESCALERA \_\_\_\_\_ COD. POSTAL \_\_\_\_\_

POBLACION \_\_\_\_\_ PROV. \_\_\_\_\_

TELEFONO \_\_\_\_\_ FIRMA \_\_\_\_\_



# UN PROGRAMA QUE APRENDE

En los tres capítulos de esta serie publicados hasta el momento hemos visto los principios generales de la Inteligencia Artificial en su parte teórica. Ahora le toca el turno a la parte práctica.

Incluso en un tema tan apasionante como el de la **Inteligencia Artificial**, la teoría puede llegar a resultar tediosa si no se administra en su justa medida. No obstante, aún a riesgo de aburrir a los más impacientes o a los que ya tuvieran algunos conocimientos sobre el tema, tuvimos que empezar por el principio: por la teoría. De otro modo hubiera sido muy poco didáctico.

En este capítulo vamos a dar por sentados unos principios muy básicos que, a pesar de su sencillez, es probable que algún lector no conozca. Para quien esté en ese caso nos remitimos a los números anteriores. En cuanto a aquellos que disfruten de un nivel más avanzado, les invitamos a olvidarse del texto de este artículo y a centrarse en el estudio del programa que lo acompaña el «programa inteligente» que hemos llamado **Sabelotodo**.

A pesar de estar escrito en BASIC, muestra lo que la **Inteligencia Artificial** puede llegar a hacer a través de un breve puñado de instrucciones. Otra ventaja, además de su corto listado, es que está pensado para que el lector lo modifique según sus necesidades o aficiones. Pero su verdadero valor reside en la experimentación de que puede ser objeto para ilustrar y comprender todo lo que hemos venido explicando en los últimos números. Como prometimos, «ni una palabra de teoría».

## SABELOTODO

Se trata de un curioso ejemplo de aplicación de la **Inteligencia Artificial** que burla las limitaciones del BASIC

e inaugura una serie de programas similares que irán apareciendo dentro de esta sección en los próximos números.

El **Sabelotodo** se basa en un sistema matricial de tablas de propiedades con **recursividad**, que permite memorizar datos y tratarlos deductivamente para «aprender» en la forma que describíamos en números anteriores. Es decir, un programa ¡que aprende!

Lo hemos hecho de forma que lo que aprenda sea Historia, pero es susceptible de ser modificado, con unos pocos retoques, para trabajar de la misma manera con otros temas, como Deportes, Records, etc.

A continuación pasamos a ver detenidamente qué es lo que hace, para describir después cómo lo hace:

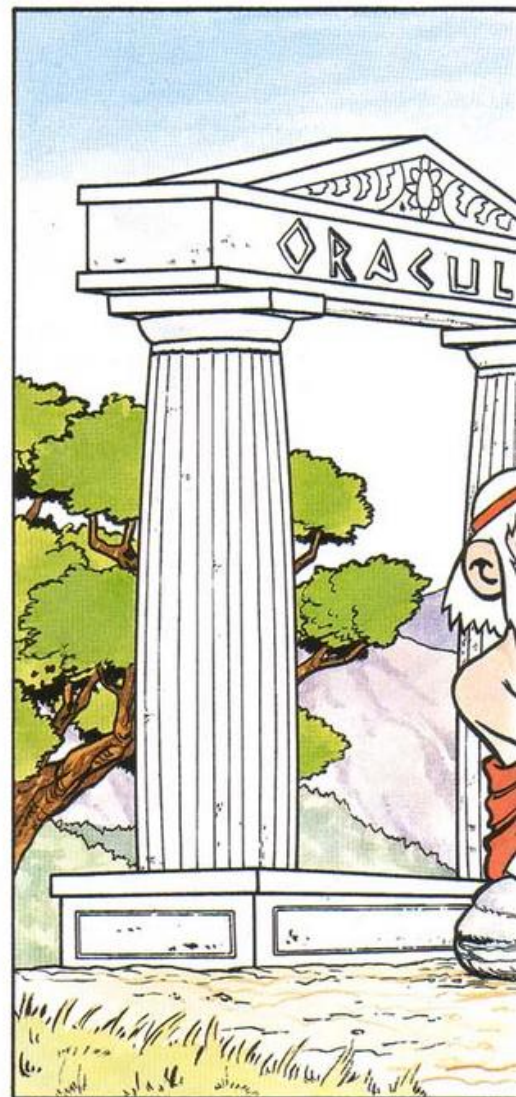
1º.- Pensamos un personaje histórico cualquiera, del que conozcamos algunas características, comenzamos a jugar.

2º.- El programa, que ha de adivinar el nombre del personaje, nos pregunta si posee una característica concreta (alguna que ya tenga en su memoria), y así sucesivamente hasta deducir, por eliminación, cuál es el personaje que habíamos pensado.

3º.- Si el programa no tiene almacenado previamente (si el programa no ha «aprendido» previamente) el nombre y las características del personaje, se da por vencido, nos pide su nombre, y nos pregunta más cosas sobre él para ampliar sus conocimientos.

4º.- De esta forma, partiendo de un solo personaje almacenado en la memoria, puede llegar a conocer (aprender jugando) un sinnúmero de personajes, hasta llenar totalmente la memoria.

El programa está estructurado con mucha claridad y acompañado de las oportunas sentencias REM, pues su misión es precisamente la de servir como material de trabajo para que



aprendáis a hacer vuestros propios programas de **Inteligencia Artificial**.

Ahora que sabemos qué es lo que hace, vamos a ver el cómo:

1º.- El programa toma la primera característica del primer personaje almacenado en la memoria y pregunta si nuestro personaje la posee.

2º.- Si la respuesta es negativa, coloca el valor «cero» a todos los personajes que posean esa característica y pasa al siguiente que tenga todavía el valor «uno».

3º.- Si es positiva, coloca el valor «cero» a todos los personajes que no la posean y pasa a la siguiente característica del primer personaje.

4º.- Sigue con el proceso anterior hasta que todos los personajes que conoce tengan el valor «cero», en cuyo caso se da por vencido, o hasta que encuentre uno que cumpla todas las condiciones, es decir, que hayamos respondido afirmativamente a todas sus características.

5º.- Si el personaje propuesto por el





ordenador no es el que habíamos pensado (a pesar de que hubieran coincidido todas las características), o si el ordenador se da por vencido, nos pide el nombre del personaje y almacena bajo él todas las características deducidas de las preguntas y respuestas anteriores, más las que introduzcamos directamente cuando nos pida más datos sobre él.

Por si esta descripción pudiera parecer demasiado enrevesada, vamos a verlo con un ejemplo práctico. Supongamos que el programa ya tiene almacenados en la memoria los nombres de cuatro insignes personajes históricos con sus correspondientes características, y que nosotros nos disponemos a jugar con él para que trate de adivinar un nuevo personaje que se nos ocurra. En este momento, las tablas de propiedades del programa muestran los siguientes datos:

**CERVANTES**  
fué literato

escribió El Quijote  
luchó en Lepanto  
era manco

**NAPOLEON**  
fué emperador  
conquistó Europa  
venció en Jena  
fué desterrado

**GOYA**  
fué pintor  
trabajó en la corte  
murió en 1828  
era aragonés

**KANT**  
fué filósofo  
era prusiano  
nació en 1724  
renovó la filosofía

Nosotros pensamos un personaje cualquiera, un pintor, por ejemplo: **Velázquez**.

El ordenador comienza por pregun-

tarnos: ¿Fué literato? (primera característica del primer personaje). Contestamos que no. A continuación el programa busca a todos los personajes que cumplan esa condición (de los cuatro, sólo la cumple **Cervantes**) y les coloca un «cero». El siguiente personaje que aún no ha sido eliminado es **Napoleón**. Por tanto, nos pregunta ¿Fué emperador? La respuesta es no, y el ordenador procede de la misma manera. El siguiente es **Goya**. El programa nos pregunta ¿Fué pintor? Contestamos que sí, con lo que se coloca un «cero» para **Kant**, que no cumple esa condición. Ya sólo nos queda **Goya**. La pregunta siguiente es ¿Trabajó en la corte? Como **Velázquez** también pintó en la corte, contestamos que sí. Después preguntará ¿Murió en 1828? A lo que responderemos negativamente. El programa coloca un «cero» a **Goya** y, como ya no quedan más personajes, se da por vencido y nos dice:

«No lo conozco. ¿Quién es?»

Introducimos su nombre (**VELAZQUEZ**), y a continuación nos pide que le digamos más cosas sobre él. Introducimos dos características más:

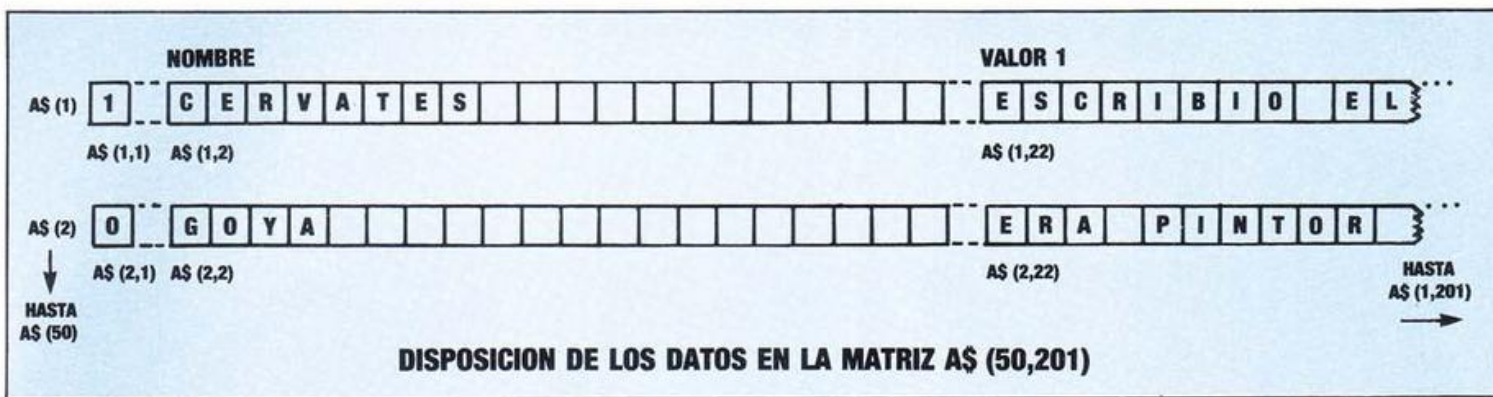
«Nació en Sevilla» y «Murió en Madrid»

Finalmente la tabla de propiedad de **Velázquez** queda así:

**VELAZQUEZ**  
fué pintor  
trabajó en la corte  
nació en Sevilla  
murió en Madrid

Las características a las que habíamos contestado afirmativamente («fué pintor» y «trabajó en la corte») se habían almacenado previamente en el lugar que iba a ocupar en la matriz el nuevo personaje. Antes de hacer cada pregunta, el programa consultaba esta lista de propiedades nueva para no repetir una pregunta ya formulada. Si al final hubiera resultado que el personaje sí era conocido por la máquina (por ejemplo, si hubiéramos pensado en **Goya**), esta nueva lista se hubiera borrado, pues no se trataría de un personaje nuevo.



**DESGLOSADO**

Cuando ejecutéis el programa, observaréis que hace lo mínimo necesario para cumplir con su cometido, y que no pierde el tiempo en adornos ni virtuosismos. Esto es así porque lo hemos hecho eliminando todos los elementos superfluos, y organizando el listado todo lo que ha sido posible, para que no haya ninguna dificultad en localizar, analizar y (éste es nuestro deseo) modificar las rutinas.

Hasta ahora hemos utilizado la palabra «característica» para referirnos a los elementos que caracterizan a los personajes. Desde el punto de vista de nuestra semántica, esto es correcto, pero en el *argot* de la **Inteligencia Artificial** no lo es. A partir de ahora nos referiremos a ese concepto llamándolo «valor».

Aquellos que no estén familiarizados con el tratamiento de matrices alfanuméricas encontrarán las explicaciones que vienen a continuación quizás demasiado complicadas. Si es así, recomendamos que se salten esa parte y practiquen un poco haciendo uso del manual. No es difícil de aprender.

La base de datos del programa se fundamenta en una matriz bidimensional de tipo alfanumérico (concretamente A\$(50,201)), de cincuenta por doscientos un caracteres, que hemos dividido imaginariamente de veinte en veinte caracteres por línea a fin de delimitar el espacio disponible para cada valor. Por esta razón, el programa no admite cadenas de más de veinte caracteres. Para hacerlo más gráfico, nos remitiremos a la figura 1. Cada línea de la matriz, A\$(1) por ejemplo,

consta de diez espacios imaginarios, más un espacio de un sólo carácter para indicar si el personaje en cuestión ha sido ya descartado o no (0/1). En el primer espacio se coloca el nombre, y en los nueve siguientes sus valores correspondientes, de forma que a cada personaje le corresponde una línea de la matriz.

Dicho esto, la primera sugerencia que se nos ocurre es ampliar la capacidad de la matriz para dar cabida a más personajes y/o más valores.

En cuanto al listado del programa, las sentencias REM ya indican claramente dónde empiezan y acaban las rutinas y cuál es su función. De todas formas, vamos a desglosarlas someramente.

La rutina 1000-1040 es la encargada de ordenar y repartir el trabajo a las demás. Procesa nuestras respuestas y envía el flujo del programa a la rutina 1100-1180 ó a la 1300-1360 según sea nuestra respuesta positiva o negativa respectivamente. Apréciense que ambas rutinas comparten la línea 1162 y siguientes (se ha hecho así para ahorrar memoria y acortar el listado).

El resto de las rutinas son las encargadas de introducir nuevos datos después de «darse por vencida» la máquina, salvar y cargar datos, pasar a un nuevo personaje, etc. Será misión del lector «desmenuzar» el programa línea por línea para llegar hasta el fondo de todas las rutinas e introducir las modificaciones que crea oportunas.

para ponerlo en marcha de nuevo no hay que hacer nunca RUN, pues eso borraría todas las variables, sino GO TO.

Al comenzar, el programa almacena en memoria un sólo personaje (**Cervantes**), con unos cuantos valores:

- Nació en el siglo XVI
- Murió en el siglo XVII
- Escribió **El Quijote**
- Estuvo en Lepanto
- Era manco
- Estuvo encarcelado

Esto debemos tenerlo en cuenta cuando juguemos por primera vez. A medida que vayamos usando el programa, éste aprenderá muchos más personajes y sus correspondientes características (valores). Si no queremos perder los datos almacenados al apagar el ordenador, cuando el programa nos pregunte si queremos pasar a otro personaje, podremos acceder a las rutinas de carga y grabado pulsando «N». Si no podemos esperar a que el programa nos haga esa pregunta (la hace cuando hemos terminado con un personaje), podemos interrumpir el programa pulsando **BREAK** y hacer:

SAVE «nombre» DATA A\$()

Cuando el programa nos haga una pregunta, basta con pulsar «S» para contestar afirmativamente, y «N» para hacerlo negativamente. Si el programa no identifica el personaje que le proponemos, preguntará «No lo conozco. ¿Quién es?». En este caso teclearemos su nombre (no más de veinte caracteres), y después de que nos diga «Dime más sobre él», teclearemos sus características (hasta nueve)

## FUNCIONAMIENTO

Como siempre, lo primero es advertir que, si interrumpimos el programa,



# ¡PARTICIPA EN INPUT!

Si quieres ver  tus programas,  
ideas,  o artículos,  publicados en  
tu revista,  examina  las  
bases y haznos llegar  el material.

Publicar tiene su recompensa.



## BASES

**PROGRAMAS:** Una vez desarrollado tu programa, que debe ser original y no haber sido enviado a ninguna otra publicación, puedes enviárnoslo aquí grabado en cassette, diskette o microdrive. Es preferible que vaya acompañado por un listado de impresora, pero no es imprescindible.

El programa 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.

**ARTICULOS E IDEAS:** Se aplica lo anteriormente dicho para los textos que acompañan a los programas; es decir, conviene detallar al máximo lo que desees que aparezca publicado en la revista, de la manera que te gustaría que otra persona hubiera explicado eso mismo. **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 olvidéis 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 cada mes **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 desaniméis!, por muy simples o complejas que puedan parecer vuestras ideas, todas serán revisadas con el máximo interés.

## INPUT SINCLAIR

Alberto Alcocer, 46, 4.º B  
28016 Madrid

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



una por una, seguidas de ENTER . Cuando hayamos terminado con ellas, teclearemos FIN.

No nos cansaremos de decir que se trata de un programa abierto a vuestras manipulaciones. Si podéis introducir alguna mejora (seguro que sí), no dejéis de hacerlo.

En el próximo número volveremos con otro interesante programa I.A., hasta entonces, esperamos que paséis muy buenos ratos con el **Sabelotodo**.

```

10 REM SABELOTODO
20 REM COPY ERNESTO DEL
  VALLE. 1986
30 INK 7: PAPER 1: BORDER 1:
  CLS
40 POKE 23609,50: POKE 23658
  ,8
50 PRINT AT 9,10;"SABELO
  TODO"
70 PAUSE 0
80 REM COMIENZO
90 CLS: BEEP .1,30: PRINT AT
  20,0;"ESTOY LISTO.[20*
  ESPACIO]COMENZAMOS ?"
95 PAUSE 0
100 IF INKEY$="N" THEN GO TO
  1750
110 DIM A$(50,201): GO SUB
  9000
120 LET R=0: LET LL=22: LET
  C=0: LET L=2: LET X=1:
  LET Y=22
1000 REM SELECTOR
1005 LET N$=A$(X,Y TO Y+19):
  IF N$="[20*ESPACIO]"
  AND R=1 THEN GO TO 1900
1006 IF N$="[20*ESPACIO]"
  AND R=0 THEN GO TO 1160
1007 FOR N=22 TO 182 STEP 20
1008 IF A$(L,N TO N+19)=N$
  THEN GO TO 1160
1009 NEXT N
1010 CLS : PRINT INVERSE 1;
  AT 1,0;"PERSONAJE ";
  L-1; INVERSE 0;AT 21,0;
  N$;" ?": BEEP .2,45
1020 PAUSE 0: IF INKEY$="S"
  THEN GO TO 1100
1030 IF INKEY$="N" THEN GO
  TO 1300
1040 GO TO 1020
1100 REM LO ES

```

```

1105 LET C=0: LET R=1
1110 FOR N=X TO L-1
1120 FOR F=22 TO 182 STEP 20
1130 IF A$(N,F TO F+19)=N$
  THEN LET C=1
1140 NEXT F: IF C=0 THEN LET
  A$(N,1)="0"
1150 LET C=0: NEXT N
1155 LET A$(L,LL TO LL+19)=
  N$: LET LL=LL+20
1160 LET Y=Y+20: IF Y<183
  THEN GO TO 1000
1161 IF Y>183 AND R=1 THEN
  GO TO 1900
1162 LET Y=22
1165 LET X=X+1: IF X>=L AND
  R=0 THEN GO TO 1800
1170 IF A$(X,1)="0" THEN GO
  TO 1165
1180 GO TO 1000
1300 REM NO LO ES
1305 LET R=0: LET C=1
1310 FOR N=X TO L-1
1320 FOR F=22 TO 182 STEP 20
1330 IF A$(N,F TO F+19)=N$
  THEN LET C=0
1340 NEXT F: IF C=0 THEN LET
  A$(N,1)="0": LET C=1
1350 NEXT N
1360 GO TO 1162
1400 REM LOAD
1410 LOAD "MATRIZ" DATA A$( )
1420 GO TO 1700
1500 REM OTRO
1530 LET T=0: CLS
1540 PRINT AT 21,0;"DIME MAS
  SOBRE EL.": BEEP .2,45
1550 INPUT LINE U$: IF LEN
  U$>20 OR LEN U$<1 THEN
  GO TO 1550
1560 IF U$="FIN" THEN GO TO
  1700
1570 LET A$(L,LL TO LL+LEN
  U$)=U$: PRINT AT T,0;U$:
  LET T=T+1
1580 LET LL=LL+20: IF LL>200
  THEN GO TO 1700
1590 GO TO 1540
1600 REM SAVE
1610 SAVE "MATRIZ" DATA A$( )
1620 BEEP .1,30: GO TO 1700
1700 REM FIN
1702 LET L=L+1: LET LL=22:
  LET C=0: LET X=1: LET
  Y=22

```

```

1703 IF L>50 THEN CLS :
  PRINT "NO HAY MAS
  MEMORIA": STOP
1705 IF L>50 THEN GO TO 9100
1710 FOR N=1 TO L: LET A$(N,
  1)="1": NEXT N
1720 CLS : BEEP .1,30: PRINT
  AT 20,0;"SEGUIMOS CON
  OTRO ?": PAUSE 0: IF
  INKEY$="N" THEN GO TO
  1750
1730 GO TO 1000
1750 PRINT AT 20,0;"PULSA 1
  PARA SALVAR.[12*
  ESPACIO]PULSA 2 PARA
  CARGAR.": PAUSE 0
1760 IF INKEY$="1" THEN GO
  TO 1600
1770 IF INKEY$="2" THEN GO
  TO 1400
1780 GO TO 1750
1800 REM NO LO SE
1810 CLS : PRINT AT 20,0;"NO
  LO CONOZCO.[18*ESPACIO]
  QUIEN ES ?": BEEP .2,
  45: INPUT LINE U$
1820 IF LEN U$>20 OR LEN U$<1
  THEN GO TO 1810
1830 LET A$(L,2 TO 2+LEN U$)
  =U$: GO TO 1500
1900 REM ES...
1910 CLS : BEEP .4,40: PRINT
  AT 20,0;"ES ";A$(X,2 TO
  2+19);"?": PAUSE 0
1920 IF INKEY$="N" THEN LET
  R=0: GO TO 1162
1930 IF INKEY$="S" THEN GO
  TO 1950
1940 GO TO 1900
1950 FOR N=2 TO 201: LET A$
  (L,N)=" ": NEXT N
1960 LET L=L-1: GO TO 1700
9000 REM DATAS
9010 FOR N=2 TO 122 STEP 20
9020 READ B$: LET A$(1,N TO
  N+LEN B$)=B$
9030 NEXT N
9040 DATA "MIGUEL DE
  CERVANTES","NACIO EN EL
  S.XVI","MURIO EN EL S.
  XVII","ESCRIBIO EL
  QUIJOTE","ESTUVO EN
  LEPANTO","ERA MANCO",
  "ESTUVO ENCARCELADO"
9050 RETURN

```



# LOS MEJORES DE INPUT SINCLAIR

PUESTO	TITULO	PORCENTAJE
1.º	Commando .....	22,8 %
2.º	Rambo .....	18,7 %
3.º	Saboteur .....	13,1 %
4.º	Sir Fred .....	10,1 %
5.º	Winter Games .....	10,1 %
6.º	The Dambusters .....	6,5 %
7.º	Profanation .....	6,1 %
8.º	Skyfox .....	4,4 %
9.º	Three weeeks in the Paradise	4,1 %
10.º	Basketball International .....	4,1 %
		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».

Mayo de 1986





# El mapa de N.O.M.A.D.

En alguna parte del espacio interestelar, se encuentra **Talos**, la fortaleza volante de **Cyrus T. Gross**, el más malvado criminal espacial conocido hasta el momento. La **Asociación de Mundos Libres** ha decidido declararle la guerra, y para ello ha construido a **N.O.M.A.D.**, el más sofisticado mecanismo autónomo de defensa y ataque, y que nosotros deberemos guiar desde el puerto espacial hasta la última estancia de este asteroide artificial, donde se encuentra nuestro odiado enemigo.

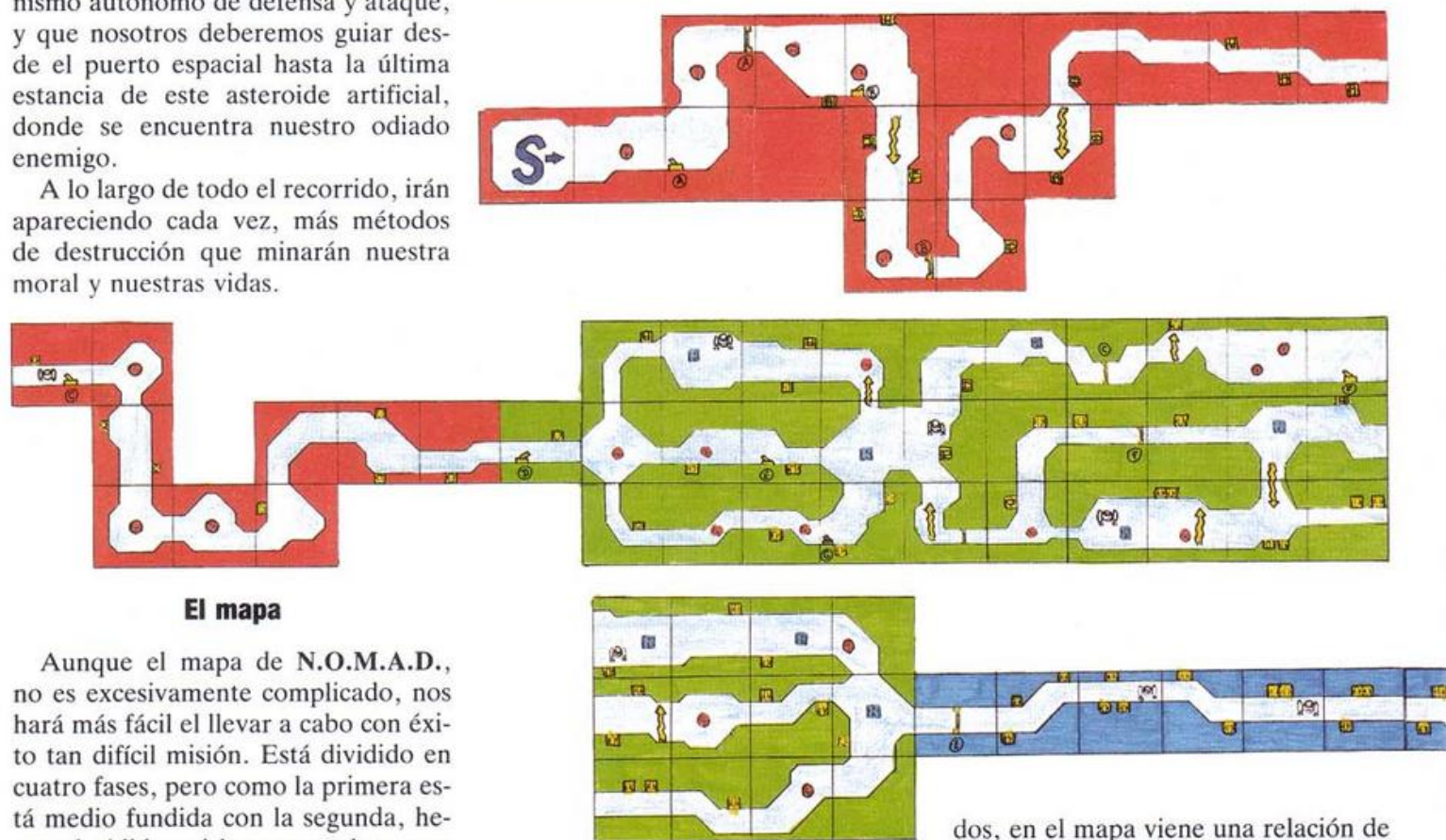
A lo largo de todo el recorrido, irán apareciendo cada vez, más métodos de destrucción que minarán nuestra moral y nuestras vidas.

cuando estemos en alguna de ellas y nos destruyan, nuestro lugar de comienzo, no será la salida, sino el principio de la fase.

La primera fase, coloreada de rojo, es la más fácil de todas, y la más difícil, la segunda (verde). Hay unos cuantos dibujos en el mapa que repre-

te destruir todos los que veamos, pues si nos matan, volveremos a pasar por ellos.

**Interruptores y barreras:** Estos son fácilmente utilizables a nuestro favor, pues basta con ponerlos hacia la derecha para saber que todas las barreras quedarán desactivadas. De todos mo-



El mapa

Aunque el mapa de **N.O.M.A.D.**, no es excesivamente complicado, nos hará más fácil el llevar a cabo con éxito tan difícil misión. Está dividido en cuatro fases, pero como la primera está medio fundida con la segunda, hemos decidido unir las y contarlas como una.

Estas fases se distinguen porque

sentan los diferentes impedimentos que **Cyrus** ha dispuesto para su defensa, y que son los siguientes:

**Discos de fuerza:** Nos atacan con balas iguales a las nuestras, son muy peligrosos, pues si estamos encima de ellos y disparan, nos destruirán aunque la bala no fuese dirigida hacia nosotros. Se encuentran en todas las fases.

**Cañones de plasma:** Su disparo es lineal y persistente. También se encuentran en todas las fases, desde la primera hasta la última. Es importan-

dos, en el mapa viene una relación de correspondencias entre interruptores y barreras.

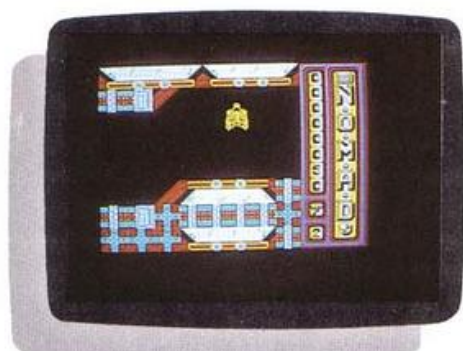
**Gravedad:** Está indicada en el mapa por una flecha, que señala hacia qué lugar nos veremos arrastrados. Normalmente, nos llevan hacia sitios en los que hay cañones o discos, y se pueden eludir si se juega con cuidado, pero es en la penúltima pantalla donde más peligro tiene.

**Misiles de cabeza rastreadora:** Son también bastante peligrosos, porque una vez disparados, seguirán incansablemente el calor de los motores de **N.O.M.A.D.** hasta ser destruidos por





# LA LEYENDA DE LAS AMAZONAS



una de nuestras balas o hasta que choquen con cualquier pared. Esto último ocurrirá si huimos de ellos, que es lo más juicioso.

**Robotrones:** Nunca salen de las salas donde están confinados. En un principio están ocultos, y saltarán sobre nosotros un momento después de que hayamos entrado en la pantalla donde se encuentran, por lo que debemos penetrar en éstas disparando, para evitar que nos cojan por sorpresa. En el mapa están puestos en el sitio exacto de su aparición.

## El final

En la pantalla final, donde está **Cyrus**, aparecerá un robotrón cada vez que éste pulse un botón del trono. Hemos de tener cuidado y situarnos en una buena posición hasta que tengamos la oportuni-



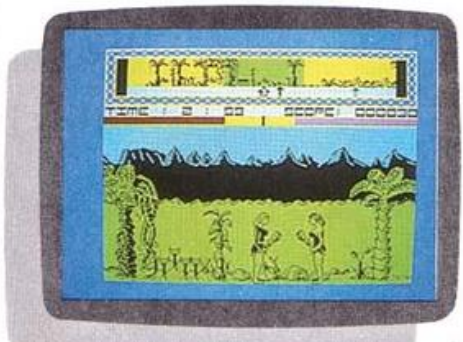
dad de acercarnos rápidamente y disparar a **Cyrus**, con lo que nuestra misión habrá terminado en teoría, pues hecho esto, se nos dará una vida extra y tendremos que volver a empezar...

## Conclusión

**N.O.M.A.D.** es un juego que al principio no dice gran cosa, pero según se avanza en él, aprendiendo a dominarlo, se le ve con otros ojos, y a quien le gusten los desafíos, seguro que terminará apasionándole.

Trás un grave accidente aéreo la hija de **Lady Wilde** ha desaparecido. Es necesario iniciar la búsqueda inmediatamente, antes de que sea

En las tres primeras zonas las amazonas portan únicamente garrotes, pero a medida que nos adentramos en la selva van mejor



## DATOS GENERALES

**TITULO** Amazon Women

**FABRICANTE** U.S. Gold

**ORDENADOR** Spectrum 48K

**TEMA DEL PROGRAMA**

Mujeres guerreras

## CALIFICACION (Sobre 10 ptos.)

<b>ORIGINALIDAD</b>	<b>8</b>
<b>INTERES</b>	<b>7</b>
<b>GRAFICOS</b>	<b>8</b>
<b>COLOR</b>	<b>6</b>
<b>SONIDO</b>	<b>8</b>
<b>TOTAL</b>	<b>37</b>

demasiado tarde, ya que probablemente ha sido raptada por las amazonas, una peligrosa tribu de mujeres que habitan en la zona. El camino que recorre la jungla pasa por diez zonas de dificultad creciente. En cada una de ellas hay varias amazonas dispuestas a cortar el paso a las siguientes zonas. **Lady Wilde** se puede defender atizándolas «garrotazos» en la cabeza, las piernas y el cuerpo, recibiendo diez puntos por cada golpe acertado.

En la parte superior de la pantalla figuran unos indicadores que informan sobre las energías que restan a cada una de las contendientes, que es posible recuperar a medida que transcurre el tiempo sin recibir ningún golpe.

pertrechadas y así en las cuatro siguientes portan espadas y las últimas unas afiladas hachas. Para estar en igualdad de condiciones, tras matar una amazona es posible apoderarse de su arma agachándose a recogerla.

La selva amazónica guarda muchas sorpresas, venenosos cardos cuyo contacto provoca disminución de energía, flechas errantes que aparecen en cualquier momento y dragones que si te alcanzan, terminarán con tu aventura. Estos dragones aparecen a partir de la zona siete, surgen de pequeños montículos y la mejor forma de hacerles frente es estando agachado ya que de este modo no llegan sus llamadas.

Se dispone de cuatro minutos para



atravesar cada zona y cuanto antes se consiga, mayor puntuación se obtiene.

El programa presenta un bonito

escenario con unos gráficos muy atractivos, a pesar de la ausencia de color.

U.S. Gold ha realizado un

entretenido juego que tiene una característica muy peculiar: es uno de los pocos del mercado que tiene como protagonistas a mujeres.

★★★★★★★★★★★★★★★★★★★★

## EL INSPECTOR MARLOWE

### DATOS GENERALES

**TITULO** Movie

**FABRICANTE** Imagine

**ORDENADOR** Spectrum 48K

**TEMA DEL PROGRAMA**

Policías y ladrones

### CALIFICACION (Sobre 10 ptos.)

**ORIGINALIDAD** 10

**INTERES** 10

**GRAFICOS** 9

**COLOR** 7

**SONIDO** 2

**TOTAL** 38



muestra los indicadores de tomar o dejar los objetos, la pistola, un puño que representa la posibilidad de golpear, la opción de hacer hablar al protagonista, una pisada que simboliza el inicio de la marcha y el lanzador de bombas.

Los objetos transportados pueden ser dejados en cualquier momento y permanecerán en el lugar en que sean depositados. Igualmente como los matones que andan sueltos por la casa que nunca salen de las habitaciones en las que están. El personaje puede moverse en las cuatro direcciones, aunque sus movimientos son algo lentos y por tanto el peligro se incrementa. Las teclas de control no son redefinibles, pero existen diversas opciones de distribución que permiten seleccionar la más adecuada. Como en otros juegos el control del personaje es total, pero tiene la peculiaridad de poder elegir un movimiento automático del detective alrededor de los objetos inanimados.

La gran novedad que **Movie** aporta es la capacidad de hacer hablar al personaje. De este modo, aunque sin ser un programa conversacional, admite una participación directa mediante la realización de preguntas a los habitantes de la casa. Cuando **Marlowe** se encuentra con un

confidente podrá obtener información de él mediante diferentes métodos, incluido el soborno.

Si el confidente no es un «vulgar soplón» rechazará el dinero airadamente e insultará gravemente al detective. A pesar de ello, hay que intentarlo ya que cualquier comentario puede significar una pista importante para localizar a la bella



### Tanya.

Los protagonistas del juego disponen de alrededor de 60 frases que permiten establecer un diálogo con el inspector, el cual puede realizar preguntas como ¿Cuál es la clave? ¿Por donde voy? ¿Qué hago ahora?, etc. Eso sí, todos los diálogos deben ser en inglés, por lo que aquellos que no están familiarizados con el idioma, deberán poner mucha atención al escribir o leer las frases que surjan y consultar las instrucciones donde figuran todas las sentencias.

Los secuaces de **Bugs** no siempre se muestran educados y suyas son frases como ¡Te mataré!, ¡Intente rezar!, ¡Elimínalo!, ¡Matadlos a todos! y cosas similares.

En situaciones de peligro hay que mantener la calma ya que la combinación de teclas para disparar, golpear o lanzar las bombas, resulta

Ataviado con un sombrero americano, una vieja gabardina, y las manos en los bolsillos, el detective **Marlowe** tiene que resolver un difícil caso. La hermosa **Tanya**, una rica y encantadora heredera ha sido secuestrada por una banda organizada de delincuentes a sueldo. Son los peligrosos años 30 en los que la mafia controlaba los bajos fondos de Nueva York, donde el inefable **Bugs Malloy** ha construido su imperio.

**Marlow** acaba de llegar a la mansión del jefe de la banda y es ahora cuando empiezan sus problemas. Los medios con que cuenta no son muchos: un revólver automático con seis balas, unos buenos puños y la facultad de recoger aquellos objetos que le ofrezcan alguna pista, como puede ser una comprometedor grabación.

La parte inferior de la pantalla



algo complicada; mientras, los secuestradores no dejarán de atacar. Por si fueran pocos los problemas, **Tanya** tiene una hermana gemela que trabaja para **Bugs** y no hay modo de distinguirlas. Cuando se encuentra con **Marlow** hay que tener cuidado ya que lleva al inspector por caminos equivocados hasta su muerte.

La ayuda de **Tanya** es imprescindible, ya que ella conoce la clave para abrir las puertas protegidas, por las que se consigue llegar hasta el despacho de **Bugs**, donde se encuentra la cinta grabada. Es recomendable el diseño de un mapa de la mansión ya que resulta muy útil para salir de ella o para el caso en que se tropieze con la hermana gemela de **Tanya**. Si la muchacha desea alguna cosa,

por ejemplo un *whisky*, es conveniente concedérselo, ya que en caso contrario puede molestarse y negarse a colaborar. Igualmente ocurre si alguno de los matones intenta golpearla, por lo que habrá que protegerla celosamente.

Por otro lado determinados objetos que parecen inalcanzables pueden conseguirse mediante la utilización de los que ya se poseen y un poco de ingenio.

La puntuación se obtiene por el número de habitaciones recorridas y las labores realizadas, apareciendo en pantalla al final de cada partida los puntos alcanzados.

Los gráficos son auténticos decorados, que junto con las conversaciones del protagonista con otros personajes hacen del juego casi una historia real.

El juego incluye los alicientes básicos de cualquier aventura policiaca, con hermosas chicas, alcohol, delincuentes sin escrúpulos, soplonos y peligrosas situaciones para la integridad del protagonista.

**Imagine** ha conseguido un programa



que auna buenos gráficos, un argumento interesante y sobre todo una participación más directa.

\*\*\*\*\*

## EL REINO DE CAMELOT

Un aguerrido caballero provisto de espada y armadura, ha de alcanzar el misterio que se oculta tras los muros del castillo de **Camelot**.

Para llegar allí, primeramente hay que atravesar un peligroso bosque con muchos enemigos como insectos, pájaros, extraños seres y veloces animales cuyo simple contacto provoca la muerte.

Además hay plantas carnívoras, grandes montañas y lagunas que añaden más dificultades a la misión. Una vez localizado **Aznaht**, el druida amo y señor del bosque, hay que conseguir que éste transforme al guerrero en un «viscoso batracio» con el fin de pasar a la segunda fase

que se desarrolla en un lago lleno de peligros con peces asesinos y mortales medusas, todos ellos bajo el dominio de **Kindo**, rey del lago.

A continuación se sumerge bajo tierra, en las grutas donde habita

**Azorniz**, dueño de estas cavernas, lugar que provoca en el protagonista la urgente necesidad de buscar una salida para alcanzar la superficie de la cueva y respirar aire puro.

Por último, llega al castillo de



### DATOS GENERALES

**TITULO** Camglot Warriors

**FABRICANTE** Dinamic

**ORDENADOR** Spectrum 48K

**TEMA DEL PROGRAMA**

Aventura medieval

### CALIFICACION (Sobre 10 pto.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	8
COLOR	8
SONIDO	6
<b>TOTAL</b>	<b>36</b>



**Camelot** que guarda la clave del misterio, protegido por **Arturo**, rey de estas tierras.

El guerrero puede moverse a izquierda o derecha, y está dotado de un potente salto que le permite acceder a difíciles lugares, además de una poderosa espada capaz de destruir a muchos de los animales

que aparecen en la aventura.

El juego incluye una pegadiza melodía que se escucha una vez cargado el programa y cada vez que se inicia la aventura.

Para evitar copias ilegales de los hábiles «piratas», el programa cuenta con un pequeño dispositivo exterior que se acopla en el *slot* de expansión

posterior, sin cuyo concurso el programa no funciona, y eso que lo hemos intentado.

La firma **Dinamic**, netamente española, ha realizado un buen trabajo con este programa que nos transporta a la Edad Media, aunque cuenta con algunos elementos que no pertenecen a dicha época.

★★★★★★★★★★★★★★★★★★★★

## LOS PICAPIEDRA

Continúa la aparición de juegos que tienen como protagonistas a conocidos personajes de famosas películas. En esta ocasión es el simpático **Pedro Picapiedra**, una inolvidable creación de los ya míticos **Hanna-Barbera**, quien surge en el televisor para llevar a cabo una

comienzo de la futura ciudad troglodítica. Para ello primeramente debe limpiar el terreno de piedras y grandes rocas, y una vez construido el hogar, buscar a **Wilma** para mostrarle su obra.

Sin embargo una serie de animales van a incordiarle constantemente

larga produce la pérdida de una de sus vidas. También existen unas peligrosas piedras rodantes que pueden ocasionar grandes daños. Una ventaja del juego es que se pueden seleccionar el número de vidas disponibles, hasta un máximo de ocho.

En pantalla aparece la puntuación, el número de vidas y el estado de las energías, así como el prehistórico

### DATOS GENERALES

**TITULO** Yabba Dabba Doo

**FABRICANTE** Quicksilva

**ORDENADOR** Spectrum 48K

**TEMA DEL PROGRAMA**

Aventura Prehistórica

### CALIFICACION (Sobre 10 pto.)

**ORIGINALIDAD** 6

**INTERES** 6

**GRAFICOS** 7

**COLOR** 5

**SONIDO** 6

**TOTAL** 30



escenario en el que se desenvuelve la aventura.

Una vez construida la casa debe buscar a **Wilma** que puede encontrarse en el *burger*, en la gasolinera, etc. Un detalle simpático: si se localiza el **troncomóvil**, se puede utilizar para transportar las rocas de una manera más fácil y menos pesada.

Se conceden puntos extras por la construcción de la primera casa y también puntúa cada piedra que se deposite en el gran cráter que está próximo al solar en que se ha iniciado la construcción.

Con una atractiva música, **Quicksilva** ha logrado crear un entretenido juego que nos recuerda aquellos célebres dibujos animados.

hogareña aventura.

Su objetivo es convencer a la bella **Wilma** que forme un feliz hogar con él. El único problema es que ella exige una confortable casa que demuestre el amor de **Pedro**.

En un desértico paisaje **Pedro** ha de edificar la primera casa que será el

impidiendo que realice tranquilamente su labor. Tortugas, dinosaurios y brontosaurios desde tierra, y pterodáctilos desde el cielo intentarán complicarle las cosas.

Cada vez que un animalito de estos le roce, provocará una disminución en las energías de **Pedro** que a la

★★★★★★★★★★★★★★★★★★★★



# EL GRAN CASINO

Los juegos de casino han arruinado a más de uno y aunque la banca casi siempre gana, para muchos es imposible evitar la tentación de conseguir hacerla saltar.

Jugar contra un ordenador tiene la ventaja de que aunque le desplumen a uno, basta pulsar un par de teclas para recuperar toda la fortuna perdida y continuar jugando.

El programa incluye cuatro juegos distintos. El primero de ellos es el conocido Bingo que permite apostar a un máximo de diez jugadores.

Es posible obtener los cartones a través de la impresora, seleccionando los que se deseen, pero en caso contrario es menos cómoda la selección.

Se puede seleccionar la velocidad de las extracciones, el precio del cartón y el número de cartones y jugadores. Cada extracción se anuncia con sonido y en la pantalla además del número aparece el extraído anteriormente, el número de extracciones y los ya aparecidos. Si se desea probar la rapidez de reflejos y la memoria, es posible jugar simultáneamente con nueve cartones, y realizar las extracciones con muy bajos intervalos de tiempos.

La ruleta es otro de los divertimentos que se ofrecen en este *cassette*. En pantalla se presenta un tapete verde compuesto de treinta y siete números y las casillas correspondientes a las distintas apuestas que se pueden realizar. Son cuatro los tipos de apuestas: tres con una probabilidad media de acierto y la cuarta apostar al número, algo muy difícil pero que en caso de

acierto representa importantes dividendos. Las apuestas se pueden realizar a pares o nones, rojos o negros y *manque* o *passe*, partiendo con cinco mil pesetas y tres participantes como máximo. Los dos juegos restantes están



basados en la baraja francesa, simulando en pantalla una auténtica mesa de juego. En el denominado Punto y Banca pueden participar hasta catorce jugadores, utilizando un total de seis barajas. Para aquellos que no conocen las reglas el programa facilita las instrucciones en pantalla.

Por último el Black-Jack, similar a las siete y media española, que con seis barajas permite apostar hasta siete participantes.

Al igual que en el anterior es posible disponer de jugadores automáticos que pueden hacer más real el desarrollo de la partida.

Resumiendo, un entretenido conjunto de juegos de QLS que pueden calmar los deseos de los apostantes más empedernidos, sin el riesgo de empeñar hasta la camiseta.

## DATOS GENERALES

TITULO Ruleta...

FABRICANTE QLS

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA

Juegos de Salón

## CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	5
COLOR	5
SONIDO	4
<b>TOTAL</b>	<b>28</b>

\*\*\*\*\*

# LA BATALLA DE BULGE

Cambiar los acontecimientos de la historia siempre ha sido un sueño difícilmente realizable. Con la llegada del ordenador los juegos de simulación adquieren una gran

relevancia y las escenificaciones bélicas son una de las aplicaciones más características.

Este programa de **Software Center** rememora los acontecimientos

acaecidos a partir del 16 de diciembre de 1944, cuando se inició la batalla para conquistar **Antwerp**, el principal puerto de suministro aliado en aquellos momentos de la



Segunda Guerra Mundial. El juego se desarrolla en tiempo real, lo que quiere decir que es necesario actuar con rapidez y

estrategia, consultar el mapa o descifrar los mensajes. Es conveniente examinar detenidamente el mapa e interpretar

La pantalla muestra una sección de la zona en que se desarrolla la batalla, pero es posible disponer de una imagen total del campo de

## DATOS GENERALES

**TITULO** The Bulge

**FABRICANTE** Software Center

**ORDENADOR** Spectrum 48K

**TEMA DEL PROGRAMA**

Simulación bélica

## CALIFICACION (Sobre 10 pto.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	6
COLOR	7
SONIDO	6
<b>TOTAL</b>	<b>33</b>



decisión ya que el enemigo continua su avance, incluso en los momentos de reflexión o durante la noche. No obstante existe la posibilidad de parar el juego y poder así elaborar la

correctamente todas las informaciones y mensajes que se reciben en pantalla ya que tener una información puntual y exacta es decisivo para conseguir la victoria.

batalla, visionando las posiciones de todas las fuerzas. En la parte inferior van sucediéndose las informaciones y mensajes que es preciso interpretar. En el margen derecho figura la hora y las unidades disponibles de los contendientes. La puntuación se obtiene a medida que se conquistan poblados y posiciones, valorando distintamente las unidades capturadas. Otra interesante posibilidad que ofrece el juego es la grabación de la situación en cualquier momento del juego que permite continuar la partida más adelante, con las ideas más claras.

★★★★★★★★★★★★★★★★★★★★

## LOS MEJORES

### DE INPUT

En la revista de Abril, y debido a un error de montaje, apareció publicada la lista correspondiente a INPUT-COMMODORE. A fin de que podáis seguir la marcha de vuestros favoritos reproducimos aquí la relación correcta.

PUESTO	TITULO	PORCENTAJE
1.º	Commando .....	18,4 %
2.º	Saboteur .....	18,1 %
3.º	Profanation .....	13,1 %
4.º	Sir Fred .....	9,9 %
5.º	The Dambusters .....	9,4 %
6.º	Camelot Warriors .....	7,6 %
7.º	Rambo .....	6,7 %
8.º	West Bank .....	6,7 %
9.º	Super Test .....	5,2 %
10.º	Three Weeks in Paradise .....	4,9 %
		<hr/> 100 %

Abril de 1986



**SOFTWARE: ¡¡2 PROGRAMAS POR EL PRECIO DE 1!!**  
**Y ADEMÁS, COMPLETAMENTE GRATIS, UN MAGNÍFICO RELOJ DE CUARZO. INCREÍBLE ¿VERDAD?**

PING PONG _____	2.295 ptas.	KNIGHT LORE _____	1.750 ptas.	MASTER OF T.	
SABOTEUR _____	2.295 ptas.	FAIRLIGHT _____	1.950 ptas.	LAMP _____	1.950 ptas.
RAMBO _____	2.295 ptas.	BATALLA DE LOS		NIGHTSHADE _____	1.750 ptas.
YEAR KUNG FU _____	2.295 ptas.	PLANETAS _____	2.100 ptas.	HACKER _____	1.950 ptas.
WORLD SERIES		ASTROCLONE _____	1.900 ptas.	SUPER TEST _____	2.300 ptas.
BASEBALL _____	2.095 ptas.	TURBO ESPRIT _____	2.100 ptas.	ALIEN 8 _____	1.750 ptas.
MAPGAME _____	2.750 ptas.	DYNAMITE		GUNFRIGHT _____	1.750 ptas.
RAID _____	2.295 ptas.	DAN _____	2.100 ptas.	TRES S. PARAISO	2.100 ptas.
MIKIE _____	2.100 ptas.	THE SOLD A		COSMIC	
HIGHWAY		MILLION _____	2.500 ptas.	WARTOAD _____	2.100 ptas.
ENCOUNTER _____	1.750 ptas.	OLE TORO _____	2.100 ptas.	N.O.M.A.D. _____	2.100 ptas.

**SOFTWARE DE REGALO:**

DUMMY RUN  
YENTZ

DRAGONTORC  
FIGHTING WARRIOR  
MAPSNACH

SHADOW FIRE  
SOUTHERN BELLE  
POLE POSITION

BOUNTY BOB  
SAIMAZOOM  
TAPPER

SPECTRUM PLUS + 6 JUEGOS  
¡¡27.800 PTAS.!!

GRATIS: 2 WALKIE TALKIES!!

¡¡NUEVOS PROGRAMAS EXPLOSIVOS FUTURE STARS!!

ALI BEBE KRIPTON RAIDERS TOMMY  
LOS TRES, 1.895 PTAS. Y ADEMÁS, GRATIS, 1 CALCULADORA

LAPIZ OPTICO DK'TRONICS  
¡¡2.890 PTAS.!!

INTERFACE-1 10.900 PTAS.  
MICRODRIVE 10.900 PTAS.

CASSETTE ESPECIAL PARA  
ORDENADOR 5.295 PTAS.

AMPLIACIONES DE MEMORIA  
¡¡3.995 PTAS.!!

SERVICIO TECNICO DE REPARACION  
TARIFA FIJA: 3.600 PTAS.

OFERTA IMPRESORAS: TODAS LAS MARCAS  
CON UN ¡¡20% DTO. SOBRE P.V.P.!!

PRECIOS EXCEPCIONALES PARA TU AMSTRAD  
CPC-464 CPC-6128 Y PCW-8256

**TECLADOS PROFESIONALES:**

— SAGA-1 9.295 PTAS. — INDESCOMP 13.195 PTAS.

WAFADRIE 16.995 PTAS. QUICK DISK 2.8" 23.995 PTAS.

OFERTAS JOYSTICK QUICK SHOT  
GRATIS A CALCULADORA

QUICK SHOT II + INTERFACE 3.695 PTAS.  
QUICK SHOT V + INTERFACE 3.995 PTAS.

CARTUCHOS MICRODRIVE  
DISKETTES 5 1/4"  
DISKETTES 3"  
CARTUCHERAS PARA MICRODRIVE

495 PTAS.  
295 PTAS.  
990 PTAS.  
150 PTAS.

CINTA C-15 ESPECIAL ORDENADOR  
INTERFACE CENTRONICS/RS-232  
AMPLIFICADOR DE SONIDO  
INTERFACE DOBLE KEMPSTON

69 PTAS.  
8.495 PTAS.  
2.390 PTAS.  
2.795 PTAS.

Pedidos contra-reembolso sin ningún gasto de envío. Teléf.: (91) 275 96 16 o escribiendo a MICRO-1  
C/ Duque de Sesto, 50. 28009-Madrid





# EL ZOCO

Pedro Manuel Garrido.  
Héroes del Alcazar, 16, 3º B.  
34004 Palencia.

**Vendo** Videojuego Videopac Philips G-7000, con palancas más 5 juegos (comecocos, golf y 3 más) precio a convenir; o lo cambio por impresora Sheikosa GP-50.

Curro.  
Avda. Sant Jordi, 33.  
Tel. (977) 303137.  
Reus (Tarragona).

**Vendo** ordenador Spectrum Plus comprado en Marzo de 1985, TV B/N 22 pulgadas, cassette especial ordenador, 57 revista Microhobby Semanal, y los 150 mejores juegos del mercado. Todo 55.000 ptas.

Javier Teijeira Lastres.  
José Cornide, 10, 6º izda.  
Tel. (981) 232637.  
La Coruña.

**Se vende** consola Atari más 7 juegos como Berzerk, Comecocos, Star-Wars, etc. El precio de venta es de 20.000 ptas.

Roberto.  
Tel. 719 43 64.  
Madrid.

**Vendo** Copion Turbo Lerm TC7 por 600 ptas. Cambio programas con gente de Guadalajara. Tengo programas buenos como el Night Lore, Profanation, Match Point, Scuba Dive, etc. Dirigirse a:  
Pedro Carpintero García.  
Carretera de Iriepal.  
Guadalajara.

**Ocasión.** Vendo ZX Spectrum, casi sin usar. (Accesorios, Libro, Revistas, Juegos, ...) Todo por sólo 24.000 ptas.  
Josep.  
Tel. (93) 7261483.  
Sabadell (Barcelona).

**Desearía** contactar con personas a las que las gustaría intercambiar información sobre el Spectrum.

Virginia Navas Martín.  
Zurbarambarri, 59 bis, 12º pta. 3.  
Tel. (94) 4462750.  
Begoña (Vizcaya).

**Vendo** cassette Computone. En garantía, nuevo. Precio a convenir.

Carlos.  
Tel. 233 58 71.  
Madrid.

**Cambio** o vendo compacto, televisión, cassette y radio (3 bandas), red, pilas y batería de coche, por una disquera para Spectrum o por un microdriver y su interface 1, que tengan instrucciones en castellano.

Elías Díez.  
Bº Lau Bide, 18, 1º izda.  
Tel. (943) 733639.  
Legazpia (Guipuzcoa).

**Estoy interesado** en intercambiar trucos e ideas para el ZX Spectrum. Los interesados pueden escribir a:

Carles Jordi Fernández i San José.  
Carretera a Bagá, 42, 2º.  
Guardiola de Berguedá (Barcelona).

**Vendo** joystick, precio 1250 ptas. (discutibles). Revistas Microhobby nº 1 al 29. Todas 1000 ptas cada una 33 ptas. Sueltas 50 ptas. cada una. Revistas Video Basic nº 1 al 8. Todas 2250 ptas, sale a 270 cada una (discutible). Sueltas 400 ptas. cada una. Todo es nuevo y sin usar. Dirigirse a:

Pedro Vicedo Molina.  
Perino, 1.  
Tel. (965) 810234.  
Biar (Alicante).

**Intercambio** todo tipo de Software para Spectrum 48K o Plus. Escribir a:

**Vendo** ZX 81 nuevecito por 5.000 ptas ó 6.000 ptas. con cables y manual. Re-

galo libro y listados. También vendo ampliación a 16K estropeada por 1.500.

José López Operador.  
Avda. de Madrid, 1.  
Tel. 850075.  
Benidorm (Alicante).

**Vendo** Spectrum 16K comprado en Enero 85. Doy fuente de alimentación... Daría cinta/s. Todo 23.000 ptas.

Miguel Angel Cebrián Martín.  
Daroca, 25, 2º A.  
Tel. (976) 317202.  
Zaragoza.

**Interesa** contactar con usuarios de Spectrum de toda España.

Manolo Tortajada Gírbés.  
Hernán Cortés, 19, 3º.  
46004 Valencia.

**Vendo** Spectrum Plus (Sept. 85), alimentador, cables, cinta de demostración, manuales en inglés y castellano, cinta original de THE DAMBUSTERS. Todo nuevo. Precio 27.000 ptas.

Ernesto González Serrano.  
Progreso, 41, 4º.  
Tel. 436643 (a partir de las 22 h.).  
Vigo 2 (Pontevedra).

**Se buscan** personas que deseen entrar en un club de usuarios de ZX Spectrum sin ánimo de lucro en programas. Los interesados pueden escribir a:

Diego Enrique Alarcón Aguirre.  
Carretera Estación, 10.  
Tel. (951) 431002.  
Albox (Almería).

**Cambio** seis juegos comerciales nuevos de Spectrum 48K por emisora de radioaficionado en buen estado, (no importa modelo). Escribir a:

Oscar José González Mazón.  
Urbanización Santa Bárbara  
bque 2, 1º D.  
Tel. (985) 430872 llamar de 2 a 3 ó de 6 a 7 de la tarde.  
Turón (Oviedo).

**Vendo** interface joystick por 3.500 ptas. y 40 revistas dedicadas al ZX Spectrum o cambio por programas. Escribir a:

Tomás Baño Nigorra.  
San Ignacio, 58, 5º-1º.  
07008 Palma de Mallorca (Balears).

**Vendo** números 1,2,4,5,6 de Micromanía por 1.100 ptas. No con ánimo de lucro. Interesados:

Manolo Tortajada Gírbés.  
Hernán Cortés, 19, 3º.  
46004 Valencia.

**Vendo** Spectrum Plus 64K. Sólo tiene dos meses por 25.000 ptas.

Alberto.  
Tel. (93) 347 17 80.

**Si se te hace difícil encontrar INPUT en tu kiosco habitual, resérvalo por adelantado, o háznoslo saber para que podamos remediarlo**



# ¿LO HUBIERA PODIDO COMPRAR MAS BARATO?

Los clientes de Regisa esta pregunta ya no se la hacen. Pero además cuando conozcan las **nuevas ofertas** de monitores, ordenadores, impresoras, unidades de disco, periféricos, software, etc. (**evidentemente todo con garantía**), que ha preparado Regisa, se van a llevar una agradable sorpresa.

ventas al mayor

## REGISA

Comercio, 11 - Tel. 319 93 08 - Barcelona

**lo mismo y más..., pero al mejor precio.**



**sinclair**   **AMSTRAD**   **SPECTRAVIDEO**   SEIKOSHA   **DK-TRONIC**  
**commodore**   **HIT BIT**   **:RITMAN:**   **FONTEC**  
                                  **SONY**

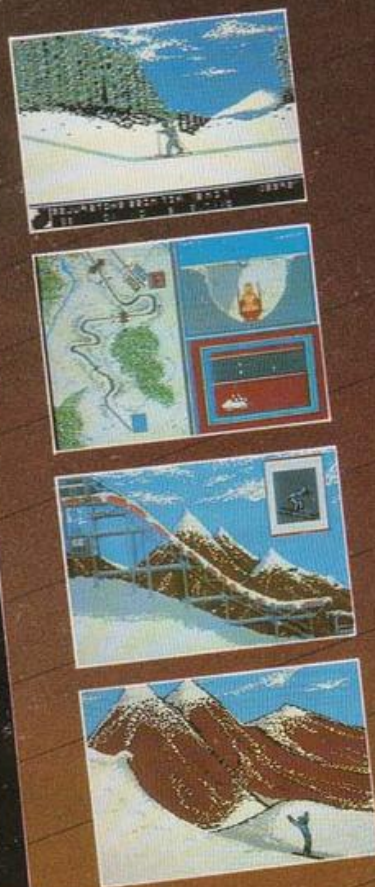
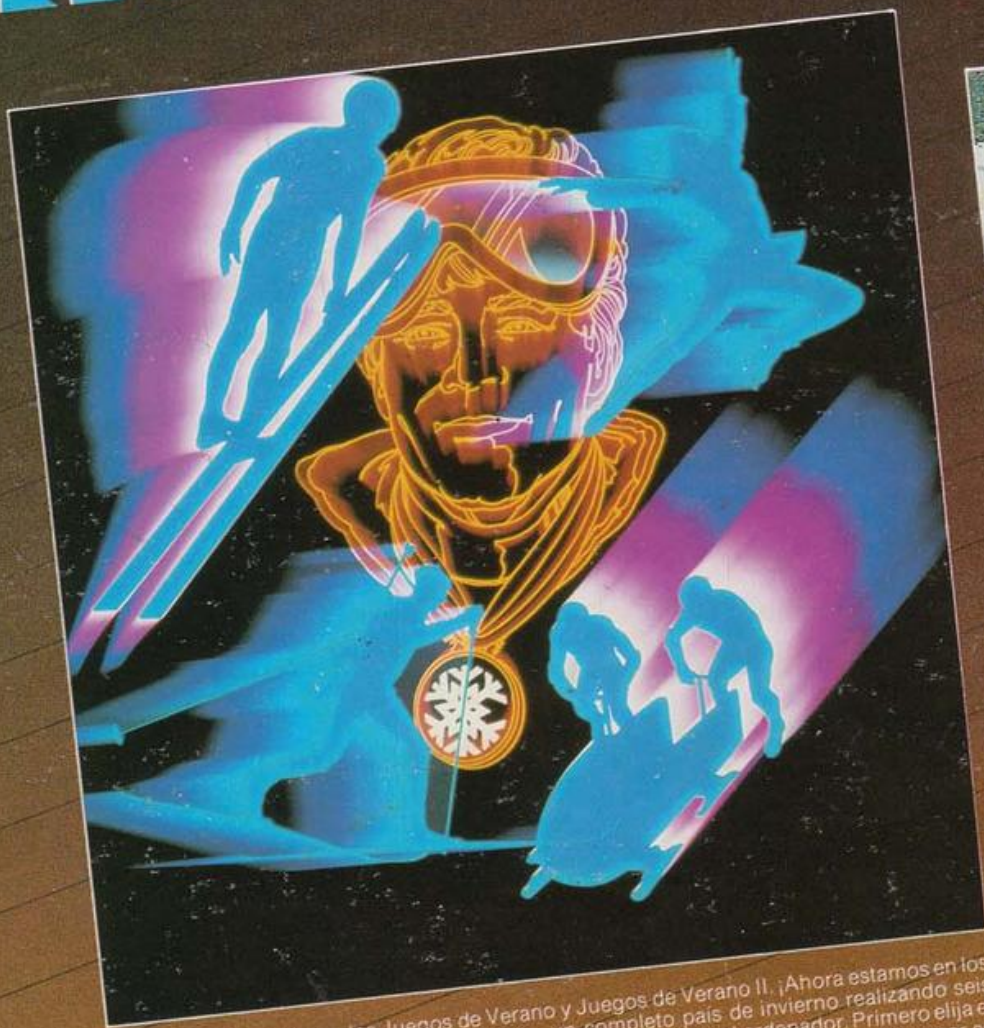
**Establecimientos recomendados:** • BAZAR DELHI. Reina Cristina, 11. Barcelona • INTERJOYA. Reina Cristina, 9. Barcelona • BAZAR TAIWAN. Plaza Palacio, 19 (Galerías). Barcelona • LOS GUERRILLEROS. I. Canarias, 128. Valencia • BAZAR KARDIS. I. Canarias, 130. Valencia • BAZAR DELHI. M. Ruano, 5. Lleida



Primero fué SUMMER GAMES  
despues SUMMER GAMES II

y ahora...

# WINTER GAMES



Ha conseguido el oro en los Juegos de Verano y Juegos de Verano II. ¡Ahora estamos en los Juegos de Invierno!, y qué increíble marco, un completo país de invierno realizando seis competiciones de acción. Puede competir contra sus amigos o el ordenador. Primero elija el país que quiera representar. Practíquelo, prepárese y aprenda una estrategia para ganar en cada competición. Ahora comience la ceremonia de apertura y la competición. ¿Será usted quien consiga el oro en la ceremonia de entrega de premios? La búsqueda del oro continúa... y está todo aquí: la estrategia, el reto, la competición, el arte y la pompa de los Juegos de Invierno.

- Seis competiciones de invierno: Bobsled, salto de ski, patinaje artístico, patinaje libre estilo, Hot Dog Aéreo y el ski de fondo.
- Ceremonias de apertura, cierre y entrega de premios con himnos nacionales.
- Compita contra el ordenador o contra sus amigos o familia.
- Control único por el joystick, necesita destreza y cronometraje.
- Uno a ocho jugadores.

**EPYX**  
COMPUTER SOFTWARE

Fabricado y distribuido en  
exclusiva por:

**COMPULOGICAL S.A.**

Santa Cruz de Marcenado, 31 - 28015 Madrid - Tel. 241 1063

Distribuido en Cataluña y Baleares por: **YA ESTA DISPONIBLE PARA EL SPECTRUM**  
DISCLU, S.A. - Balmes, 58 - BARCELONA - Tel. (93) 302 39 08 - P.V.P. 2.300 Ptas.