

150 Ptas.

Canarios 160 ptas.

25 años

S.I.M.O.  
LA ERA  
DEL GEMAMSTRAD  
multitarea  
con un reloj  
permanente  
en pantallaRutinas  
de movimiento  
de Sprites en  
código máquina.  
Diseñador  
de Sprites

SOFTWARE

Rocky Horror Show,  
la pesadilla de Brad  
y Janet

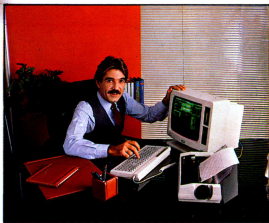
# POR EL PRECIO DE UNA MAQUINA DE ESCRIBIR..

**169.900 PTAS.**



**EL SISTEMA INFORMATICO PCW 8256 INCLUYE EN SU SUMINISTRO:**

- Teclado profesional en castellano.
- Unidad Central, 256K RAM
- Pantalla de alta resolución.
- Unidad de disco.
- Impresora.
- Programas de disco:
  - Procesador de Textos LocoScript en castellano.
  - Mollard BASIC con sistema JETSAM para ficheros indexados.
  - Sistema Operativo CP/M Plus.
  - Lenguaje Dr. LOGO.
  - Diversas utilidades.
- Completa documentación y Manuales en castellano.



**AMSTRAD** ESPAÑA

# AMSTRAD

## sumario

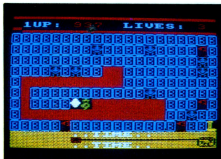
Año 1 • Número 15 • 10 al 16 de Diciembre de 1985  
150 ptas. (sobretasa Canarias, 10 ptas.)

### 5 Primera plana

Primer simposio sobre Inteligencia Artificial: el sistema Fermi. El revolucionario proyecto Enigma de Sinclair Research.

### SIMO 6

Amstrad Semanal no podía por menos que estar presente e informar a nuestros lectores de las incidencias y novedades del SIMO 1985. Como en todas las ferias, ha habido software. El sistema operativo GEM se impone entre los ordenadores y las máquinas que lo corren ya están en España: Atari, IBM y Apricot, entre otras.



### Serie Oro 22

Esta vez os traemos un juego espléndido de acción y estrategia que se explica a sí mismo mucho mejor en las fotos de pantalla que en 100.000 palabras.

### 14 ProgramAcción

Para resumir e integrar los precedentes artículos acerca del sistema RSX de ampliación del Amstrad Basic y del tratamiento de interrupciones, nada mejor que un ejemplo que crea un nuevo comando que a su vez activa un reloj de tiempo real por interrupción.



### Mr. Joystick 18

ROCKY HORROR SHOW. ¿Conseguirá la intrépida Janet, reunir todas las piezas de la DESMEDUSA y liberar de esta forma a Bard, de su estado de petrificación?



### 26 Código máquina

Seguimos analizando el uso y disfrute de sprites en código máquina, más un programa de diseño de sprites en Basic, y más, mucho más.

### Análisis 31

Los menús de opciones: cómo son y cómo se hacen. Un pequeño y útil programa explica cómo implementar esta útil técnica en nuestros programas.

#### Director Editorial

José I. Gómez-Centurión

#### Director Ejecutivo

Victor Prieto

#### Subdirector

José María Díaz

#### Redactora Jefe

Marta García

#### Diseño

José Flores

#### Colaboradores

Francisco Portalo, Pedro Sudán

Miguel Sepúlveda,

Francisco Martín,

Jesús Alonso, Pedro S. Pérez

Amalio Gómez

Juan J. Martínez,

David Sopuerta, Alberto Suárez,

Eduardo R. Velasco

#### Secretaría Redacción

Carmen Santamaría

#### Fotografía

Carlos Candel

Javier Martínez

#### Portada

M. Barco

#### Ilustradores

Javier Igual, J. Pons, F. L.

Frontán, J. Septien, Pejo, J. J.

Moro, Luigi Pérez

#### Edita

HOBBY PRESS S.A.

#### Presidente

María Andriano

#### Consejero Delegado

José I. Gómez-Centurión

#### Jefe de Publicidad

Concha Gutiérrez

#### Publicidad Barcelona

José Galán Cortes

Tel.: (903) 303 10 22/313 71 62

#### Secretaría de Dirección

Marisa Cogorro

#### Suscripciones

M.º Rosa González

M.º del Mar Calzada

#### Redacción, Administración y Publicidad

La Granja, 39

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

#### Dto. Circulación

Carlos Peropadre

#### Distribución

Coedit, S. A. Valencia, 245

Barcelona

#### Imprime

ROTEDEC, S. A. Crta. de Irún.

Km. 12,450 (MADRID)

#### Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

#### Fotomecánica

GROF

Ezequiel Solana, 16

#### Depósito Legal:

M-28468-1985

#### Derechos exclusivos

de la revista

#### COMPUTING with

#### the AMSTRAD

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud

América 1.532. Tel.: 21 24 24. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

# COMPUTIQUE

*Te da más*

GARANTIA **AMSTRAD**  
ESPAÑA

64.900Ptas.

Amstrad 464 f.  
verde



## Al comprar tu Amstrad te regalamos

- Estuche con ocho programas originales
  - Fruit Machine
  - Procesador-Texto
  - Almirante Graf
  - Oh Mummy
  - Plaga Galáctica
  - Amstraw
  - Laberinto-Sultán
  - Animal, Vegetal, Mineral
- Joystick Gunshot I
- Un estupendo libro de Basic
- Los cuatro mejores programas:
  - Deathlon
  - Sabrewulf
  - Jet Set Willy
  - Beach-Head
- Guía de referencia del programador
- y además te obsequiamos con un curso de introducción al Basic.

VENTA A PLAZOS HASTA 36 MESES



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

COMPUTIQUE

Servimos a tiendas  
Abrimos sábados por la tarde

Embajadores, 90 Tfno. 2270980  
28012 Madrid

## EL GOBIERNO PROHIBE LA VENTA DE ORDENADORES NO HOMOLOGADOS

El Gobierno prohibió la venta, importación y fabricación de ordenadores que no cumplan las normas de homologación UNE en todo el territorio nacional a partir del día 27 de noviembre de 1985, haciendo aplicación de los Reales Decretos 1250 y 1251 publicados en el BOE del 27 de julio de 1985.

La medida afecta decisivamente a la casi totalidad de lo que constituye el hardware de un ordenador, esto es, unidad central y periféricos, incluidos pantallas, impresoras y teclados.

Hasta este momento, sólo se han presentado dos solicitudes de homologación con fecha del 27 de noviembre, UNA DE ELLAS PERTENECE A INDESCOMP, DISTRIBUIDOR OFICIAL DE LOS PRODUCTOS AMSTRAD EN ESPAÑA, POR LO QUE ESTA MEDIDA GUBERNAMENTAL NO LE AFECTARÁ EN ABSOLUTO, ni a ellos, ni a los actuales o futuros usuarios de **Amstrad**.

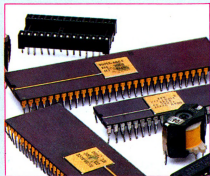
La Asociación Española de Importadores de Productos Electrónicos (ASIMELEC), asegura que la nueva normativa afecta a un volumen de importaciones que supera el MEDIO BILLON de pesetas, mientras que fuentes del Ministerio de Industria la rebajan a 2.000 millones de pesetas.

ASIMELEC denuncia al Ministerio por incumplir la promesa de prorrogar el plazo de homologación, ya que, según ellos, no ha habido tiempo material para homologar ni se han expuesto con suficiente claridad las condiciones materiales y documentación técnica necesaria para conseguirla. Todos los grandes nombres de la informática se verán afectados: IBM, Xerox, NCR, Hewlett Packard y, por supuesto, las compañías que dedican su atención al ordenador doméstico más que al personal, como Commodore y los MSX.

Sin embargo, en el Boletín Oficial del Estado del 28 de noviembre de 1985 se notifica UNA PRORROGA POR PARTE DEL GOBIERNO DE 6 MESES ANTES DE LA PLENA ENTRADA EN VIGOR DEL DECRETO LEY. LAS EMPRESAS IMPORTADORAS DISPONDRAN DE ESTOS 6 MESES PARA ADAPTAR SUS MAQUINAS A LAS NORMAS DE HOMOLOGACION.

Van a tener que darse mucha prisa, pero al menos la prórroga impide que España quede en una posición absolutamente tercermundista... durante los próximos 6 meses.

Después de todo el alboroto, la sangre no ha llegado al río, pero el Gobierno ha dado un serio toque de atención a los importadores en el sentido de que, al parecer, está decidido a llevar esta normativa hasta sus últimas consecuencias.



## MÁS Y MÁS CHIPS

En Japón, la industria del diseño y producción de chips está más boyante que nunca. Sony, por ejemplo, acaba de anunciar que incrementará su producción en un 50 por 100 para el ejercicio del 85/86, para poder hacer frente al enorme incremento de la demanda. Sony invertirá en este asunto 162 millones de dólares.



## COMPRAR ORDENADORES

La vista de la creciente importancia de los ordenadores personales en nuestras vidas a todos los niveles, entre ellos los universitarios, la Caja de Ahorros ha abierto una línea especial de créditos para la adquisición de ordenadores personales hasta un monto de **500.000 ptas.**, al parecer amortizable en 3 años y con condiciones de intereses especiales.

## Primera plana

### EL ENIGMA QUE VIENE

Sinclair Research ha decidido sumarse a las corrientes dominantes en el mercado de los ordenadores personales, los que serán la próxima generación de **«home computers»**.

Después del fracaso del QL y de la oportuna aparición del Spectrum 128K, ha sido dado a conocer al público el último proyecto de Sinclair, el proyecto ENIGMA.

Detrás de este críptico nombre se esconde una maravilla de ordenador, si se cumplen las prestaciones anunciadas por Sinclair Research, que son las siguientes:

— Microprocesador Motorola 68000 de 16/32 bits. Se ha abandonado definitivamente el 60008, el del QL, que pecaba de ser una extraña mezcla entre 8, 16 y 32 bits, lo cual disminuía bastante las prestaciones de uno de los microprocesadores más avanzados del mundo.

— Dos unidades de disco de 3,5 pulgadas: Sinclair se **«apunta»** al formato Sony, que se está convirtiendo rápidamente en algo parecido a un estándar. Por fin, la desgraciada filosofía de los microdrives se deja de lado. Ya era hora.

— Nada más y nada menos que un megabyte de memoria RAM, 1024 Kbytes. Sólo el Atari 520 ST + tiene hoy en día esa memoria a un precio razonable.

— Sistema operativo GEM, de Digital Research, un clónico del sistema operativo del Apple Macintosh que sigue la filosofía WIMP (del inglés *Windows, Icons, Mouse y Pull-down-Menus, o sea, ventanas, iconos, ratón y menús de opciones en lugar de comandos*). Este S.O. está especialmente diseñado para aprender a usarlo a pleno gas en pocos minutos.

— Lo más importante, el precio. Se espera vender el Enigma entre 500 y 1.000 libras esterlinas, y estará disponible para mayo del 86.

# 25 años S.I.M.O.

**La feria internacional del equipo de oficina e informática SIMO, cumple sus bodas de plata. Año tras año, la informática ha ido haciéndose con una representación más nutrida e importante, hasta el punto de que actualmente este sector ha desplazado totalmente al destinado al material de oficina, ocupando un 70 por 100 de la superficie dedicada expositores.**



**A**nte este magno acontecimiento, **AMSTRAD** semanal no podía mantenerse al margen y envió a su equipo de reporteros al mismo centro de la noticia.

La exposición, tradicionalmente tiene como marco el recinto ferial de IFEMA de la Casa de Campo de Madrid.

En ella se dan cita, tanto las grandes multinacionales de la informática, como las nuevas marcas, que empezando con el micro doméstico se introducen a marchas forzadas en el campo del ordenador personal de altas prestaciones, con un precio revolucionario.

Entre las primeras; IBM, ICL, OLIVETTI, APPLE, PHILIPS, etc.

Entre las de la nueva ola: ATARI, SINCLAIR, APRICOT y **AMSTRAD**, pueden citarse como destacados ejemplos.

Para realizar nuestro recorrido por los stands, qué menos que visitar en primer lugar el de la empresa número uno en el mundo informático; IBM, líder mundial en ventas.

En su departamento de ordenadores personales, tres pura sangre: IBM-PC, IBM-Portátil, IBM-AT.

Destacando entre todos la última creación de la firma, el todopoderoso AT.

El IBM-AT, incorpora la tecnología más avanzada incluyendo nuevos microprocesadores, coprocesador y diskettes de alta capacidad.

**Sus características técnicas:**  
Microprocesador Intel-80286 avanzado, de alto rendimiento.

Intérprete de direccionamiento compatible con el 8086.

256 KB de memoria RAM dinámica, 516 KB para el modelo ampliado.





Unidad de diskettes de alta capacidad de 1,2 MB.

Unidad de disco fijo de 20 MB para el modelo ampliado.

Ocho conectores de expansión E/S.

Zócalo para co-procesador 80287.

Direcciones de 24 bits, Bus de datos de 16 bits.

Ni que decir tiene que este modelo es compatible con casi todos los productos de Hardware y Software del IBM-PC, con la gran ventaja de que el rendimiento del modelo AT, puede duplicar e incluso triplicar la velocidad del IBM-PC.

La memoria se puede ampliar desde los 256 KB iniciales hasta un máximo de 3 MB.

La unidad central admite dos unidades de diskette y una unidad de disco fijo, o una unidad de diskette y dos de disco fijo, configuración con la cual se puede conseguir una capacidad de almacenamiento de 41,2 MB.

El teclado incorpora sustancialmente mejoras en la distribución de las mismas con un conjunto de 84 teclas e indicadores de modalidad.

PHILIPS, con su ordenador personal P3100, el cual puede ser integrado en una red de comunicaciones.

El P3100, dispone de una memoria RAM de 128 KB, ampliable a un máximo de 640 KB.

Puede utilizar una o dos unidades de disco flexible más una de disco duro de 10 MB.

Utiliza el sistema operativo MS DOS de Microsoft, compatible con IBM.



Ideal para ser conectado a otros ordenadores y a redes públicas de datos mediante los paquetes de comunicaciones; BSD, SDLC, SNA y X25.

También puede funcionar como terminal de videotex o de teletext. OLIVETTI, además de sus afamadas máquinas computerizadas de escribir y ordenadores de tratamiento de textos, tiene el M24. Un ordenador de 16 bits, con microprocesador Intel 8086. RAM de 128 KB, ROM de 16 KB y la posibilidad de dos discos flexibles de 360/720 KB o una unidad de disco flexible y una de disco duro de 10 MB.

Como opciones complementarias: ampliación de la memoria a 384 KB y la posibilidad de un disco duro de gran capacidad 30 MB. Teclados de 83 elementos en el modelo standard o 102 en el modelo plus.

Con los sistemas operativos: MS DOS, concurrent CP/M-86, PCOS, sistema UCSD-P.

CANON, con su ordenador personal A-200, compatible IBM. Incluyendo microprocesador 8086 de 16 bits, memoria de 256 KB ampliable a 512 KB y todos los demás aditamentos que caracterizan a estos ordenadores.

Disco flexible de 360 KB, cinco slots de expansión, etc.

ITT, y sus ordenadores personales XTRA de microprocesador 8088 de 16 bits.

Memoria de 128, expansible a 520 KB y demás características similares; discos flexibles de 360 KB y duros de 10 MB y 20 MB. Hasta aquí hemos visto la gama PC todos compatibles con IBM y con unas prestaciones muy similares, debida a su compatibilidad. Con la única excepción del nuevo modelo de IBM: el AT que mejora infinitamente las características de los PC compatibles.

Pero además de los PC, existe una gama de modelos en expansión que representan la mayor innovación dentro del campo de la informática.

Estos son los modelos de las nuevas marcas como ATARI y APRICOT, que sin llegar a tener la compatibilidad IBM, incorporan unas prestaciones inigualables a un precio reducido.

Estos ordenadores tienen como base el sistema operativo GEM y se manejan por medio de un ratón, lo que nos permite efectuar complicadísimas operaciones con un solo dedo.



La precursora de este sistema ha sido la casa APPLE con su famoso MACINTOSH, el ordenador más fácil de manejar del mundo gracias a su potente sistema operativo.

El MAC está dotado de un microprocesador 68000 de 16 bits. Incluye en un sistema compacto el monitor y la unidad de disco, formando un sistema poco voluminoso y fácil de manejar.

En el stand de APPLE también estaban los modelos anteriores de la casa: el Apple IIe y el IIc.

Pero la mayor revolución, llevada de la mano de ATARI por vía de un precio increíble, es el mítico 520 ST.

Una máquina basada en el GEM y que por el módico precio de 200.000 ptas. incluye: monitor monocromo, teclado, disco flexible y ratón.

*Su microprocesador es el Motorola 68000.*

El distribuidor oficial en España de esta maravilla es Inestrónica, famosa por el Spectrum y demás productos Sinclair de los cuales es el concesionario oficial.

En el mundo de Sinclair, mencionamos el QL con sus nuevas unidades de disco y el nuevo Spectrum 128.



## 25 años S.I.M.O.

El stand de DSE presentaba los productos Apricot.

Modelos PC, F2, F10 y FP.

Los F2 y F10 basados en el GEM y con ratón por infrarrojos. La memoria RAM de ambos modelos es de 512 KB, ampliables a 720. El modelo F2 posee dos unidades de diskettes consiguiendo una capacidad total de 1,44 MB.

Mientras que el F10 incorpora un disco Winchester de 10 MB. La conexión del teclado a la unidad central se efectúa sin cable utilizándose también los rayos infrarrojos.

Modelos con un precio realmente interesante para las prestaciones que ofrecen.

Saliendo de los ordenadores GEM entramos en el mundo de **Amstrad**, que presenta modelos desde el doméstico por excelencia CPC 464, hasta el super-profesional PCW 8256, pasando por los CPC 6128. Comentados con una amplia selección de software, tanto recreativo como de gestión.

Tras lo cual queda la gama de los populares MSX, con casas como Sóni, Philips, Toshiba, Canon, Espectravideo y otras respaldando el standard.

En definitiva, una gran feria para los amantes de la informática, donde se pueden encontrar cualquier tipo de ordenador, desde el doméstico al usado por las grandes empresas.



PIDELO  
EN TU QUIOSCO

# YOUR COMPUTER, UN PLATO FUERTE PARA TU ORDENADOR

Ahora en tu quiosco, y como una auténtica primicia mundial, puedes conseguir el más sabroso y variado menú de Europa para tu ordenador.

Ya tienes editados en cassette los mejores juegos y utilidades publicadas por

la primera revista europea de ordenadores «Your Computer».

Disfruta cada mes de la gran calidad de sus programas. Además, a tu ordenador también le gustará.

Y todo por un precio inmejorable.

Your Computer. El más sabroso menú de Europa.

DISPONIBLE PARA:

SPECTRUM 48, PLUS, 128

COMMODORE 64

AMSTRAD

1

## YOUR COMPUTER

EL CORAZÓN DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES

AMSTRAD

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de Europa en ordenadores.  
Ahora reproducidos en cassette, en auténtica exclusiva mundial.

695:  
PTAS.

1  
YOUR  
COMPUTER  
EL CORAZÓN DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES  
MAYO 1988

- 1 Englebert
- 2 Jugler
- 3 Tropem
- 4 Space Eggs
- 5 Snake
- 6 Fruitie
- 7 Sprite 1, 2
- 8 Sprite

NOVEDAD MUNDIAL  
SINTAX

NOVEDAD MUNDIAL  
Total 10 programas de carga

EDITA

SINTAX, S.A.

P.º de la Castellana, 268  
28046 Madrid  
Tel. (91) 733 27 99

# LA ERA DEL GEM

**La edición del SIMO de 1985 se puede decir que ha transcurrido bajo el signo de la revolución dentro de lo clásico.**

e

ste enigmático aserto creemos que se justifica por el hecho de que pocas, muy pocas novedades importantes ha habido en lo relativo a hardware, a nuevos equipos que constituyan por sí mismos una revolución, pero el software es un sueño.

## ¿Qué se lleva en informática?

En la informática también hay modas, como en todas partes. Unas, causan mucho ruido y pocas nueces, pero otras son el producto de la madura reflexión de muchas mentes que tratan de integrar, en un solo producto de base, las exigencias de los consumidores y los últimos avances en la concepción tecnológica y filosófica de lo que debe ser la arquitectura de un ordenador.

En este sentido, el mercado y los fabricantes caminan codo con codo en exigir y crear ordenadores verdaderamente útiles, es decir, que sean auténticas herramientas de trabajo y/o diversión para cualquier persona, y que proporcionen las máximas prestaciones con la mínima dificultad de uso.



Esto, que parece obvio porque amplía enormemente el círculo de usuarios, sin perder por ello el enfoque profesional, de cábala de iniciados, que han tenido hasta hace muy poco tiempo, sólo ahora es factible y materializable en un producto concreto. La razón es muy simple: el avance tecnológico ha abaratado considerablemente el coste de los componentes y de las memorias, además de impulsar la aparición de microprocesadores mucho más potentes.

**Resulta que ya podemos humanizar los ordenadores, y enfocar su uso desde el punto de vista del hombre en lugar del de la máquina.** Apple lo ha hecho con el Macintosh, y Atari con el 520 ST.

## La saga del ratón

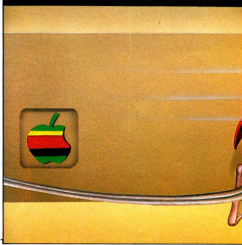
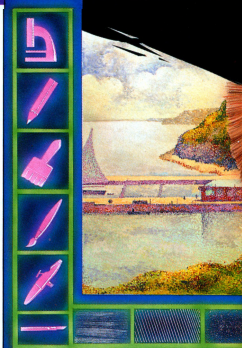
Sin embargo, ellos no fueron los primeros, aunque comercialmente se llevaran toda la gloria, sino que los pioneros en cómo hacer que las máquinas tuvieran una manera «humana» de funcionar fueron los señores del Laboratorio de Investigación de Palo Alto, en California, un centro de Rank Xerox.

Esta gente se dedicaba, fundamentalmente, a la investigación en el campo de la Inteligencia Artificial, y tenían un problema muy serio: carecían de lenguajes de programación altamente sofisticados, diseñados especialmente para la Inteligencia Artificial (LISP), pero carecían de un entorno de programación y desarrollo adecuado para el tratamiento de la información simbólica e imprecisa, es decir, de un sistema operativo «ad hoc».

En aquella época, la IA era una curiosidad de biblioteca más que una ciencia o una industria, y los ordenadores se empleaban casi exclusivamente para el cálculo numérico, por lo que los entornos de programación se ajustaban a esa tendencia.

El tándem de Xerox no se arrellanó demasiado por ello; si no tenían un entorno de desarrollo adecuado, ¡qué diablos!, harían uno.

El planteamiento fue extremadamente simple: si yo en IA manejo símbolos y objetos más que numerosos, el sistema operativo que necesito es uno orientado a objetos, de forma tal que lo que yo vea en la pantalla corresponda al objeto o símbolo que quiero someter a proceso.



Así nació el primer punto del nuevo sistema operativo: el concepto de ICONO, es decir, referenciar un objeto procesable por el ordenador mediante un dibujo. Por ejemplo, un documento se representa por una hoja de papel con líneas escritas, en lugar de algo como «CARTITA.DOC».

Segundo punto: si mis objetos ya están en pantalla, debo implementar un método de acceder a ellos que siga la misma filosofía «visual», es decir, en lugar de invocar a un documento por nombre, lo ideal sería señalarlo en pantalla con algo que permitiera saber al ordenador lo que quiero que haga.

El segundo punto se llama RATÓN, y consiste en un periférico que refleja su desplazamiento por la mesa de trabajo en pantalla mediante una pequeña flecha, de tal manera que todo lo que hay que hacer es mover el ratón por la mesa hasta que el cursor de pantalla apunte al documento, luego ¡click!, pulsamos el botón del ratón y la máquina sabe que deseamos editar un documento, por lo que ella solicita cargará en memoria, no sólo el documento, sino también el procesador de textos.



## Demasiada memoria

La gente de Xerox, además de ingenio y necesidad de un entorno de programación adecuado a símbolos, tenía dinero de sobra y unos ordenadores muy potentes, caracterizados por la famosa multitarea, es decir, la posibilidad de ejecutar varios programas independientes al mismo tiempo (*concurrentemente*).

Por esta razón, el tercer punto se llamó ventanas, y consistía en dividir la pantalla en tantas porciones como fuera necesaria, superpuestas o no, para poder mostrar al operador lo que estaba pasando con los distintos procesos que en ese momento se estaban ejecutando, todo ello, naturalmente, dirigido con el ratón, con un solo dedo.

Ya que tenemos, ratón e iconos, **¿por qué no permitir que las ventanas puedan ampliarse, disminuirse o moverse por la pantalla de acuerdo con sus necesidades?**

Ese fue el cuarto punto del nuevo entorno de programación, y, por darle algún nombre, podríamos decir que convirtió la pantalla de

un ordenador en algo infinitamente flexible, casi casi como una mesa de trabajo por la que pueden desplazarse papeles y objetos al gusto del **«usuario»**

**¡Fíjese bien: al gusto del usuario, para variar; no al de la máquina.**

El resultado final de este titánico esfuerzo de creatividad, ingenio y programación fue un híbrido lenguaje-de-programación-de-desarrollo, llamado SMALLTALK.

Y ahí se quedó la cosa, los Xerox felices con su nueva herramienta y los usuarios de ordenadores ignorantes de la existencia de algo que les evitara horas de manuales y de crisis nerviosas.

**¿Que por qué?** Por dinero, claro. No todo el mundo podía permitirse tener en su casa un ordenador como los de Palo Alto, y según la Ley de Máxima Fatalidad el SMALLTALK requería ingentes cantidades de memoria para funcionar.



**25 años**  
**S.I.M.O.**

## Llega Mac Intosh

Como se suele decir, transcurrieron buñolicos años en los que miles de usuarios noveles se vieron sometidos a tortura con el entorno de programación que se llamó MacIntosh, de Apple.

Antes que el Mac estuvo el Lisa, que incorporaba las ideas del SMALLTALK, pero pasó sin pena ni gloria debido a los precios (*10.000 dólares*) que ostentaba.

En fin, el Mac fue el primer ordenador personal que se manejaba con un entorno WIMP (*ventanas, iconos, ratón y menús*) a un precio razonable (*no muy razonable*).

Armó tal revuelo, que de pronto todo el mundo se puso a hablar de que los ordenadores debían ser «amigables», fáciles de usar, y que debía pensarse sobre todo en el grupo mayoritario que no sabía programación, ni quería saberla. Tan sólo pretendían usar una herramienta de trabajo con la máxima comodidad posible.

Los manuales de usuarios y los sistemas operativos basados en comandos comenzaron a ser vistos con malos ojos; inevitablemente se cayó en la exageración.

Como por arte de magia, justo en ese momento se hicieron de uso común máquinas con microprocesadores muy potentes y grandes cantidades de memoria RAM, y la idea de Apple (*y de Xerox*) corrió por los mentes de los diseñadores y las casas de software como un reguero de pólvora: **«los comandos han muerto, viva el ratón»**.

## Por favor, IBM, ¿qué te parece la idea?

Más de uno empezó a frotarse las manos pensando en el dinero que podría sacarse de la nueva moda, pero para que el negocio fuera significativamente jugoso habría que ver **¿qué pasa con IBM?**

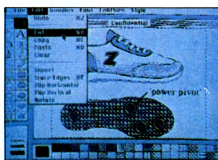
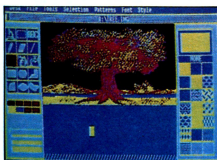
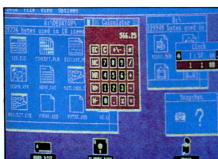
Los entornos WIMP desde luego que correrían en un PC si alguien se tomaba la molestia de escribirlo, pero, ¿se vendería?

Digital Research (*CP/M*) pensó que sí, y creó un sistema operativo basado en esta filosofía, preparado para correr bajo el sistema operativo de IBM, el «**Gestor Gráfico de entornos**», en inglés *GEM (Graphics Environment Manager)*.

Los comienzos fueron difíciles, pero al cabo de un tiempo los ejecutivos usuarios de IBM fueron capaces de manejar el ratón y observar por sí mismos las ventajas de un entorno GEM en términos de tiempo de aprendizaje y dinero ahorrados, por lo que la idea originada en Xerox y divulgada por Apple sentó plaza definitivamente en el elitista mundo del IBM PC.

Una vez recibido el espaldarazo, GEM se adaptó a otros ordenadores con microprocesadores diferentes del 8088 de IBM, y su radio de acción se extendió considerablemente. Comenzaron a aparecer paquetes de software basados en GEM, bien desarrollados por la propia Digital Research, bien por otras compañías (*GEM Write, GEM Paints GEM Draw, etc...*).

Esta es, a grandes rasgos, la historia de los sistemas operativos «amigables», basados en el sistema WIMP, y los usuarios no especializados en informática creo que nunca podrán agradecerse lo suficiente a los tres protagonistas del drama: Xerox, Apple y Digital Research.



Conscientes de todo ello, cruzamos mentalmente el umbral del SIMO para ver de cerca el «**state of the art**» de los entornos GEM en 1985.

### GEM en el SIMO

Las máquinas capaces de ejecutar GEM o su homólogo Microsoft Windows pueden dividirse en dos grandes grupos: IBM y compatibles y los demás.

Al IBM, que aparte de ser IBM y poseer una de las mayores bibliotecas de software del mundo es un ordenador notoriamente obsoleto por concepción y tecnología, el GEM le viene algo grande.

Se requiere un PS considerablemente amplio unas 512 K, configuración PC standard 128 K) para correrlo, y el microprocesador 8088 de Intel, corazón del PC, no es exactamente la última maravilla tecnológica. Pero, en fin,

funciona y es un IBM. La verdad es que las pantallas que tuvimos ocasión de admirar del GEM Paint en el stand de Digital Research eran bastante espectaculares, aunque costó lo suyo cargar el programa: se «colgaba» una y otra vez.

En cuanto a los compatibles, el panorama es muy confuso y variopinto.

GEM y MS Windows correrán en todas las máquinas que porten una determinada versión del sistema operativo MS DOS, pero en prestaciones y velocidad de ejecución depende del diseño de la máquina y del microprocesador que lleven (el 8086 es mucho más rápido que el 8088).

Lo que tanto IMB como sus compatibles parecen estar de acuerdo en imponer unos precios completamente disparados para las prestaciones de sus equipos, teniendo en cuenta el estado actual de la técnica y otras máquinas que, como luego veremos, poseen superiores «performances» a unos precios muchísimo más bajos. Si no me cree, vaya a un concesionario IMB y eche cuentas de lo que necesita para correr GEM en su PC.

Los joys del SIMO en cuanto a entornos WIMP, en mi opinión fueron sin duda los APRICOT y los ATARI.

Apricot presentó inicialmente un modelo, el FIE con un entorno WIMP de su propia cosecha, el «Activity», que ha resultado ser un auténtico desastre porque entelenece terriblemente la mecánica de trabajo con el ordenador.

Visto esto, Apricot optó por incorporar GEM a sus máquinas, y le ha quedado muy bien. GEM necesita un mínimo de 512 de RAM, por lo que cualquier modelo Apricot que los implemente debe doblar su configuración básica, fijada en 256 K.

Los Apricot de la serie F eran pequeños y bonitos, ligeros y manejables. Tuvieron ocasión de ver GEM en acción y era algo impresionante.

Además Apricot ha decidido rizar el rizo en cuanto a lo que se refiere al ratón y al teclado. Ambos periféricos están unidos al ordenador mediante rayos infrarrojos, lo cual implica un máximo de manejabilidad, pero les obliga a estar enfocados hacia la unidad central. Existe, no obstante, una conexión por fibra óptica a guisa de cable para los que no se fían demasiado de los infrarrojos, aunque por lo que vimos nosotros no falla en absoluto.

Por otra parte, estos ordenadores son sistemas MS DOS, para mantener compatibilidad con IBM, y GEM es simplemente una aplicación más, no el sistema operativo de base.

Al precio de 210.000 ptas., creo que 256 K RAM, un microprocesador 8086 de auténticos 16 bits, MS DOS y teclado con una unidad de disco de 3,5 pulgadas no está nada mal, aunque para correr GEM la broma se pone en cuatrocientos y muchas mil pesetas, contando con dos unidades de disco y sin incluir el monitor.

Apricot también presentó un gran ordena-

dor, llamado Apricot XEN, tipo IBM AT pero con un mega de memoria RAM y un sistema operativo tipo WIMP, Microsoft Windows, que al parecer es multitarea, cosa de la que el GEM no es capaz.

No pudimos observar en detalle cómo trabajaba MS Windows, ni otras máquinas aparte del Apricot que lo implementaran. El coste del equipo ascendía a cerca del millón de pesetas.

Queda lo, a nuestro juicio, auténtica vedette del SIMO 85, el ATARI 520 ST, el cual cumple perfectamente el lema de Atari «potencia sin precio».

Vimos 2 configuraciones, el 520 ST, con 512 K de RAM y el 520 ST+, que lleva nada menos que una mega de memoria central.

Ambas versiones están dotadas con el 68000 de Motorola como CPU, de 16/32 bits, y poseen dos sistemas operativos: el GEM y el TOS.

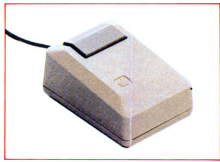
Este último es una versión del CO/M 68 K, adaptación hecha por Digital Research a los sistemas basados en el 68000 del CO/M «normal», que no tuvimos ocasión de ver funcionando, por lo que desconocemos sus características.

De GEM poco más hay que hablar, aunque verlo en acción en el 520 era un auténtico espectáculo, por la rapidez con que se realizaban las tareas, la belleza de los iconos y ventanas y la complejidad de las operaciones de acceso al tándem disco/ordenador que podían llevarse a cabo con un solo dedo, ratón en ristre.

En el stand de Electrónica estaban manejando el GEM Paint, y podían oírse murmullos de asombro de los espectadores con toda la razón del mundo. Es difícil explicar con palabras una situación y un programa concebido para manejarse a través de símbolos e imágenes, pero los dibujos y lo que se podía hacer con ellos en el Atari es simplemente increíble.

Inestrónica nos aseguró que el 520 ST se

25 años  
S.I.M.O.



venderá en unos 15 días (contando a partir del 19 de noviembre), a un precio de una 195.000 ptas. incluyendo el ordenador, monitor monocromo, unidad de disco, ratón y algo de software, como GEM, GEM Paint, GEM Write, DR Basic y DR Logo.

El 520 ST+ se venderá también en España, pero posteriormente a su hermano pequeño, a un precio de una 50.000 ptas. más.

Respecto al software, pudimos ver los catálogos de Atari en el stand de Atari, y vimos de todo: juegos, gestión y lenguaje de programación de todos los tipos.

En fin, una maravilla de ordenador a un precio ridículo por las prestaciones que ofrece.

No cabe duda de que el SIMO 85 muestra cómo será la próxima generación de ordenadores personales que habrá en nuestras casas: estarán basados en microprocesadores poderosos, tendrán gran cantidad de memoria RAM y un sistema operativo tipo WIMP, bien el GEM, o MS Windows, o lo que sea que venga detrás.

De momento, el Macintosh, los Apricot, el IBM con GEM y el Atari son ya una realidad en nuestras tiendas.



# RSX E INTERRUPCIONES

ALBERTO SUÑER

*En los últimos artículos de Programación hemos analizado exhaustivamente cómo pueden crearse nuevos comandos en el Basic Amstrad, mediante el sistema RSX, y también cómo se manipulan las interrupciones, que son, y la utilidad que tienen.*



Para remarcar el clavo completamente sobre este asunto, hemos pensado en incluir un ejemplo práctico que reúna RSX e interrupciones en una sola utilidad, concretamente se ha creado un nuevo comando Basic, «Reloj», que permite activar un reloj de tiempo real por interrupción.

En honor de aquellas personas que no han seguido los artículos anteriores, vamos a efectuar una breve recapitulación sobre los temas de los que estamos hablando. También servirá para refrescar la memoria un poco.

## Breve descripción de una RSX

El Amstrad posee una RAM y una ROM, como todo el mundo sabe. Sin embargo, a diferencia de otros ordenadores, los que lo diseñaron previeron la posibilidad de que dicha ROM pudiera ser potencia con otros programas, que denominaron RSX, extensiones residentes del sistema.

Estas RSX se identifican por su nombre precedido del signo «». El sistema operativo del ordenador se encarga, él solito, de buscarlos y ejecutarlos, siempre que se le informe debidamente de su existencia.

La inmensa ventaja de las RSX es que el sistema también las busca en RAM si no hay otras ROMs conectadas, y las ejecuta. Esto nos permite ampliar el Basic del Amstrad muy fácilmente, aunque es preciso usar lenguaje máquina.

Eso en cuanto a las RSX. Respecto a las interrupciones, recordemos que, desde que se enciende el ordenador, el microprocesador no cesa en ningún momento de ejecutar un programa en lenguaje máquina que llamamos sistema operativo o firmware.

## Más de un programa simultáneamente

Este programa es extremadamente complejo, y la tarea se complica con la necesidad de atender a otros procesos que normalmente transcurren de forma independiente al «programa principal», como por ejemplo, el teclado. En efecto, nadie nos impide pulsar una tecla mientras un programa, Basic o no, se esté ejecutando.

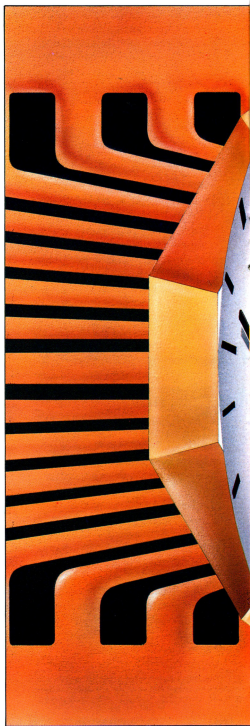
Frente a esta situación, se pueden usar dos estrategias:

— El Z80 está continuamente examinando los periféricos o lo que sea para ver si se requiere su atención para otras tareas urgentes.

— El Z80 ignora cualquier tarea secundaria hasta que recibe una señal de interrupción, abandona temporalmente la tarea principal, atiende y resuelve la secundaria y continúa con la primera desde donde la dejó.

Normalmente se adopta la segunda aproximación, porque ahorra tiempo, y por tanto dinero, de proceso.

Los creadores de Amstrad han previsto un complejo sistema de manejo de interrupciones para un ordenador de su categoría, permitiendo, entre otras cosas, asignar una prioridad a una serie de sucesos por interrupción.



Pero lo más importante para nosotros es que existe una forma estándar de notificar al sistema la presencia de un proceso interruptor, como ya vimos, de modo tal que sólo tenemos que escribir la rutina necesaria y el firmware se encarga de lo demás.

Así es posible, de una manera casi casi elemental, realizar varios procesos a la vez, esto es, la famosa multitarea.

Teniendo esto en mente, vamos a analizar lo que hemos hecho en este artículo.



Javier Igual

## El comando ireloj

Una de las tareas que parecen estar creadas para funcionar por interrupciones son los relojes de tiempo real, que casi todos los grandes ordenadores (IBM) tienen. En efecto, para que el reloj sea efectivamente de tiempo real, debe actualizarse cada segundo, pero sin interferir con cualquier otro programa que el sistema esté ejecutando.

Por otra parte, para manejar nuestro reloj lo más lógico sería des-

poner de un comando Basic al efecto, que actuara como todos los demás, en modo directo o en modo programa.

El nuevo comando se llama «ireloj», debe llevar delante el signo «@» y su sintaxis y función son las siguientes:

### 1) Orden: @RELOJ.

Esta orden, sin ningún parámetro, permite poner en hora el reloj. Para ello, se abre una ventana en la parte superior de la pantalla donde se nos pide horas, minutos y segundos. Una vez introducidos, se abre una

nueva ventana en la esquina superior izquierda de la pantalla y el reloj comienza a funcionar independientemente de cualquier otro programa.

### 2) Orden: @RELOJ, 1.

La presencia del parámetro 1 después de la orden indica al ordenador que deseamos que el reloj sea invisible, pero que continúe funcionando.

Por tanto, la ventana se borra y disponemos de toda ella para nuestro programa.

### 3) Orden: IRELOJ, 2.

Este parámetro implica que la ventana del reloj, con la hora actualizada, aparezca de nuevo en la pantalla.

## Las desventajas de las interrupciones

Obsérvese que las interrupciones presentan una desventaja bastante patente, y es que entlenen la velocidad con la que se ejecuta el programa principal, en este caso el Basic. Por tanto, debemos implementar una manera de desactivar nuestro reloj completamente.

Hemos pensado en una que esparamos resulte clara y sencilla; la orden es: **IRELOJ**.

Es decir, usamos exactamente la misma sintaxis para matar al reloj que para inicializarlo. El programa toma en cuenta si el comando se ejecuta por primera o segunda vez, y, si es la segunda, desactiva la interrupción completamente. Para que pueda volver a funcionar, es necesario usar IRELOJ de nuevo.

Resumiendo:

**IRELOJ.** Inicializa el reloj.

**IRELOJ, 1.** Borra el reloj de pantalla.

**IRELOJ, 2.** Lo muestra de nuevo.

**IRELOJ.** Al usarlo por segunda vez, lo **«mata»**.

El programa en lenguaje ensamblador está localizado en la dirección A000. Las partes de las que consta están separadas por líneas en blanco, y el método de habilitar interrupciones e inicializar una RSX ya es conocido de nuestros lectores, así como el paso de parámetros desde Basic a una rutina en máquina, por lo que no lo repetimos aquí.

Otro tanto ocurre con las llamadas al firmware, ya que la mayoría han sido analizadas en anteriores artículos en esta misma sección.

Aquellos que no dispongan de un ensamblador o que prefieran simplemente usar el programa tienen a su disposición un cargador Basic (programa número 1), que introduce el código máquina en las posiciones de memoria adecuadas, activa el nuevo comando y acto seguido se borra de la memoria, por lo que es imprescindible salvarlo en cinta/disco antes de ejecutarlo.

## PROGRAMA 1

10 FOR I=8A000 TO MA227	A851 328CA1	398	LD A,(MOD0),A
20 READ A	A854 3681	400	LD A,#0
30 FOR I,VAL("5*+*+)	A854 C3EEC	418	CALL ETHD
40 NEXT I	A859 06CBB	420	CALL CLEO
50 PRINT "CODIGO MAQUINA CARGADO..."	A85C 21B103	430	LD HL,#8B30
	A85F 0C758B	440	CALL LOCATE
	A862 21D8A0	450	LD HL,#0
	A865 08B6	460	LD B,#6
	A868 08B6	470	CALL BUCI
	A86A 00211FA2	480	LD IX,#0R2
	A874 CDEB40	490	CALL INPUT
	A871 21D6A8	500	CALL BUCI
	A874 68B4	510	LD B,#10
	A876 CDB8A0	520	CALL BUCI
	A879 00212A2C	530	LD IX,#1N2
	A870 CDEB40	540	CALL INPUT
	A883 21E8A0	550	LD HL,#SEC
	A883 08B8	560	LD B,#11
	A885 CDB8A0	570	CALL BUCI
	A88C 002125A0	580	LD IX,#SEC2
	A88C CDEB40	590	CALL INPUT
	A88F 21E3B0	600	LD HL,#M83
	A892 0C758B	610	CALL LOCATE
	A895 21C0A0	620	LD HL,#CORREC
	A898 0610	630	LD B,#6
	A89A 7E	640	BUC2: LD A,(LOC)
	A89C C05AB6	650	CALL PRINT
	A89E 23	660	INC HL
	A89F 67	670	LD A,(LOC)
	A8A1 C06B68	680	CALL WAIT
	A8A4 FE53	690	CP "0"
	A8A6 28AC	700	LD A,(IN0C)
	A8A8 38A1A1	710	LD A,#11
	A8AB CDEEC6	720	CALL SETMOD
	A8AE CDBCB7	730	CALL CLS
	A8AF 18F9	740	CALL JINTER
	A8B4 CDB1A1	750	CALL ACTIU
	A8B7 C9	760	RET
	A8B9 7E	770	
	A8BB C05AB7	780	BUC1: LD A,(HL)
	A8BE 23	790	INC HL
	A8BF 18F9	800	LD A,(LOC)
	A8C1 C9	810	LD A,#2
	A8C3 82	820	RET
	A8C4 43AF5252	830	CORREC: DEFM "CORRECTO (5/N
	A8C6 484F52A1	840	NOVA: DEFM "0RA":
	A8C8 284D4547	850	MIN: DEFM "MINUTOS":
	A8C9 284D4547	860	SEC: DEFM "SEGUNDOS":
	A8EB 06A2	898	INPUT: LD B,2
	A8ED CDEBB6	900	NECLA: CALL WAIT
	A8EF FE38	910	CP "0"
	A8F2 38F4	920	JR C,TECLA
	A8F4 38F4	930	CP "0"
	A8F7 38F5	940	JR C,TECLA
	A8FB 0D7780	950	LD (IX),A
	A8FD 0D23	960	INC IX
	A8FE C05AB7	970	CALL PRINT
	A188 98B8	980	DNEZ TECLA
	A18F 0D23	990	INC IX
	A18A C9	1000	
	A181	1010	SIND: DEF S
	A182	1020	
	A184	1030	BUFER: DEF S
	A185	1040	MOD0: DEF S
	A18D 3058	1050	PENI: DEF S
	A18E 18A8	1060	PAPI: DEF S
	A18F 1878	1070	RSXCOM: EQU #BCD1
	A190 0878	1080	ETHD: EQU #BCD1
	A191 1078	1090	SETMOD: EQU #BC8E
	A192 1180	1100	CL8: EQU #BBAC
	A193 1180	1110	LOCATE: EQU #B875
	A194 1120	1120	WAIT: EQU #B86A
	A195 1130	1130	PRINT: EQU #B85A
	A196 1140	1140	CUR_S1: EQU #B846
	A197 1150	1150	SETPEN: EQU #B839
	A198 1160	1160	SETAP: EQU #B836
	A199 1170	1170	GETPEN: EQU #B829
	A19A 1180	1180	GETAP: EQU #B827
	A19B 1190	1190	CL8: EQU #B84E
	A19C 1200	1200	SETCUR: EQU #B87E
	A19D 1200	1200	CUR_S1: EQU #B87E
	A19E 1210	1210	CAPALL: EQU #B86C
	A19F 1220	1220	INTS1: EQU #B849
	A1A0 1250	1250	INTNO: EQU #B8E4
	A1A1 1260	1260	ACTINT: EQU #B8E4
	A1A2 1270	1270	FLAG: DEF S
	A1A3	1280	
	A1A4	1290	
	A1A5	1300	INFER: LD HL,#L29
	A1A6	1310	LD B,#129
	A1A7	1320	LD HL,#EVEN
	A1A8	1330	JP ACTINT
	A1A9	1340	
	A1AA	1350	LD HL,#L29
	A1AB	1360	LD B,#129
	A1AC	1370	LD HL,#EVEN
	A1AD	1380	JP INTS1
	A1AE	1390	
	A1AF	1390	DESAC: LD HL,#L29
	A1B0	1400	JP INTNO
	A1B1	1410	
	A1B2	1420	CLOCK: DEF S

A133	1430	BLOCK:	DEFS 7	A1B6	2007	2020	JR	NZ, SEGU2
A13A	CD70BB	1440		A1B0	3E38	2030	LD	A, "0"
A13C	229CA1	1448	EVENT:	A1B6	3225A2	2040	LD	(SECC2), A
A140	CD9EA1	1470	CALL	A1B0	1095	2050	JR	MINU
A143	3HEFA1	1478	CALL	A1B0	1095	2050	JR	MINU
A144	FE82	1490	CALL	A1B0	1095	2050	JR	MINU
A148	C0	1500	RET	A1B0	1095	2050	JR	MINU
A149	CD93BB	1510	CALL	A1B0	1095	2050	JR	MINU
A14C	3290A1	1520	CALL	A1B0	1095	2050	JR	MINU
A14F	3E81	1530	LD	A1, 1			JR	NZ, MINU2
A151	CD04BB	1540	CALL	A1B0	1095	2050	JR	MINU
A154	2400	1550	LD	H, 0			LD	(MIN1), A
A15A	1087	1560	LD	D, 7			LD	(MIN1), A
A150	1E00	1570	LD	E, 0			JR	MINU3
A15A	2E01	1580	LD	L, 1			INC	A
A15C	CD64BB	1590	CALL	A1B0	1095	2050	LD	(MIN1), A
A15F	3E03	1600	LD	A, 3			RET	
A161	CD99BB	1610	CALL	A1B0	1095	2050	LD	(MIN1), A
A164	CD7E00	1620	CALL	A1B0	1095	2050	LD	(MIN1), A
A167	218101	1630	LD	HL, #0101			JR	NZ, MINU4
A16A	CD75BB	1640	CALL	A1B0	1095	2050	LD	(MIN2), A
A16D	211FA2	1650	LD	HL, HOR2			JR	HOR4
A170	0400	1660	LD	B, 0			JR	HOR4
A172	E5	1670	BUC:	PUSH	HL		INC	A
A173	C5	1680	PUSH	BC			LD	(HOR1), A
A174	7E	1690	LD	A, (HL)			JR	HOR4
A175	CD50BB	1700	CALL	A1B0	1095	2050	LD	(HOR1), A
A178	C1	1710	POP	BC			RET	
A179	E1	1720	POP	HL			LD	(HOR3), A
A17A	23	1730	INC	HL			CP	"2"
A17B	10F5	1740	DNZ	BUC			JR	NZ, HOR4
A17D	3E00	1750	LD	A, 0			LD	A, "0"
A17F	CD04BB	1760	CALL	A1B0	1095	2050	LD	(HOR2), A
A17E	2600	1770	LD	H, 0			LD	(HOR2), A
A184	1450	1780	LD	D, 00			LD	(HOR2), A
A18A	2E01	1790	CALL	A1B0	1095	2050	LD	(HOR2), A
A180	1E10	1800	LD	E, 24			LD	(HOR2), A
A186	CD64BB	1810	CALL	A1B0	1095	2050	LD	(HOR2), A
A18C	300A01	1820	LD	A, (F0H)			LD	(HOR2), A
A190	CD90BB	1830	CALL	A1B0	1095	2050	LD	(HOR2), A
A193	209CA1	1840	LD	HL, (POS)			LD	(HOR2), A
A196	CD70BB	1850	CALL	A1B0	1095	2050	LD	(HOR2), A
A199	C375BB	1860	JP	LOCATE			LD	(HOR2), A
A19C	1800	1870					LD	(HOR2), A
A19E	3A25A2	1880	POS:	DEFS 2			LD	(HOR2), A
A19E	3A25A2	1880	POS:	DEFS 2			LD	(HOR2), A
A1A1	FE39	1910	CP	A, (SECC1)			LD	(HOR2), A
A1A3	2007	1920	JR	NZ, SEGU01			LD	(HOR2), A
A1A5	3E30	1930	LD	A, "0"			LD	(HOR2), A
A1A7	3225A2	1940	LD	(SECC1), A			LD	(HOR2), A
A1AA	1095	1950	JR	SEGU01			LD	(HOR2), A
A1AC	3C	1960	SEGU01:	INC	A		LD	(HOR2), A
A1AD	3225A2	1970	LD	(SECC1), A			LD	(HOR2), A
A1B0	C9	1980	RET				LD	(HOR2), A
A1B1	3A25A2	2000	SEGU01:	INC	A		LD	(HOR2), A
A1B4	FE35	2010	CP	A, "5"			LD	(HOR2), A

# Program Acción

A1EA	3A1FA2	2300	HOR4:	LD	A, (HOR2)
A1EF	FE32	2310		CP	"2"
A1F1	3A20A2	2320	JR	NZ, HOR4	
A1F4	FE33	2340	CP	"3"	
A1FA	2018	2350	JR	NZ, HOR4	
A1FB	1007	2360	JR	CERO	
A1FC	3A20A2	2370	HOR4:	LD	A, (HOR1)
A1FD	FE39	2390	CP	"0"	
A1FF	2007	2390	JR	NZ, HOR2	
A201	3E30	2400	CERO:	LD	A, "0"
A203	3220A2	2410	LD	(HOR1), A	
A204	1005	2420	JR	HOR4	
A208	3C	2430	HOR2:	INC	A
A209	3220A2	2440	LD	(HOR1), A	
A20C	C9	2450	RET		
A20D	3A1FA2	2460	HOR4:	LD	A, (HOR2)
A210	FE32	2470	CP	"2"	
A212	2006	2480	JR	NZ, HOR4	
A214	3E30	2490	LD	A, "0"	
A21A	321FA2	2500	LD	(HOR2), A	
A219	C9	2510	RET		
A21A	3C	2520	HOR4:	INC	A
A21B	321FA2	2530	LD	(HOR2), A	
A21E	C9	2540	RET		
A21F	30	2550			
A220	30	2570	HOR1:	DEFB	"0"
A221	3A	2590	DEFB	"1"	
A222	30	2590	MIN2:	DEFB	"0"
A223	30	2600	MIN1:	DEFB	"0"
A224	3A	2610	DEFB	"1"	
A225	30	2620	SECC2:	DEFB	"0"
A226	30	2630	SECC1:	DEFB	"0"

## ETIQUETAS

ACTINT	BCE7	ACT10	A110	BLOCK	A133
BUC	A172	BUC1	ABBB	BUC2	A09A
BUFFER	A18	CANAL	BB04	CERO	A201
CLEAR	004E	CLOCK	A12D	CLS	BB6C
CORREC	A0C0	CUR_NO	BB7E	CUR_S1	BB70
DESHC	A127	ENP	A04E	ESPACE	AB14
EVENT	A13A	FLAG	A10F	GETCUR	BB70
GETHOD	BC11	GETPAP	BB99	GETPEN	BB93
HOR1	A220	HOR2	A21F	HOR4	A000
HOR4	A1EA	HOR4A	A1FA	HOR4B	A200
HOR43	A200	HOR44	A21A	INIC	A054
INPUT	A0E0	INTER	A110	INTNO	BCEC
INT S1	BCE5	LOCATE	BB75	MIN1	AB06
MIN1	A223	MIN2	A222	MINU	A1C4
MINU2	A1D2	MINU3	A1D7	MINU4	A1E5
MOD	A10C	NOME	A0BE	PAP	A10E
PAS4	A03D	PAS4A	A02F	PEN	A100
POS	A19C	PRINT	BB5A	RELOJ	A010
SE-COM	BC01	SHL	A040	SEC	A0E0
SECC1	A226	SECC2	A225	SEGU2	A10F
SEGU01	A1AC	SEGU02	A101	SETHOD	BCE0
SETPAP	BB96	SETPEN	BB98	SIN0	A105
TABLA	A009	TECLA	A00D	TIME	A17E
WAIT	BB06	WINDOQ	BB66		

## BABE LA INSTALCO

Especialistas  
en  
SOFTWARE  
para la  
pequeña y  
mediana  
empresa

- AMSTRAD 6128
- AMSTRAD 8256
- IBM y compatibles

Programas de gestión  
integrados

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

NL New Line

GABINETE DE INFORMÁTICA

- Clases de Informática sobre AMSTRAD EN GRUPOS O INDIVIDUALES
- Ordenadores AMSTRAD y periféricos Los mejores precios
- Software a la medida

ZURBANO, 4 ☎ 410 47 63  
28010 MADRID



Para que tus dedos no realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicitálos.

# ROCKY HORROR SHOW

**Una pareja de jóvenes enamorados, se ven envueltos sin comerlo ni beberlo en una angustiada pesadilla en la que sus vidas corren grave peligro de muerte.**

**T**odo empezó en los laboratorios del Dr. Frank situados en la planta de su grandiosa mansión.

En los cuales el diabólico Dr. ha desarrollado una terrible máquina capaz de petrificar al más pintado. El fruto de su siniestro trabajo del Dr. Frank se llama Medusa y su poder es tan descomunal, que solamente unos segundos de exposición al rayo malféfico son suficientes para la petrificación total.

Frank, como buen investigador también ha realizado la máquina que hace posible el proceso de despetrificación, tema que tendrá su oportuno tratamiento más adelante.

Un día cuando Janet y Brad, la pareja de jóvenes enamorados, paseaban románticamente por los alrededores de la mansión, ven el anuncio de entrada libre para visitantes, que con el pretexto de enseñar a quien lo desee el interior de la magnífica casa, ha colocado el Dr. para conseguir presas fáciles para sus experimentos.



Frank encantado por la constitución hercúlea del joven Brad, decide que es un bello ejemplar para ser convertido en decorativa estatua y se prepara para comprobar el funcionamiento de medusa con seres humanos.

Sin pensarlo dos veces apunta la máquina hacia Brad y el rayo le alcanza de lleno, comenzando el proceso de petrificación, que en 10 segundos se completa.

Janet que se encontraba fuera del radio de acción de Medusa, es afectada parcialmente quedando temporalmente paralizada y presencian-do cómo siniestros esbirros se llevan el cuerpo de su amado.

Cuando recupera su estado normal, un alma caritativa le revela que el único modo de devolver a su novio a la forma humana, es encontrar la máquina que realiza la despetrificación.

Pero la Desmedusa, que así se llama la máquina, está fraccionada en componentes los cuales se encuentran repartidos por toda la casa.

Janet solamente tiene que encontrarlos y recomponer Desmedusa, para salvar a Brad.

Misión difícil, porque solamente dispone de 100 minutos y por el hecho de que por la mansión circulan extraños personajes que pueden acabar con ella.

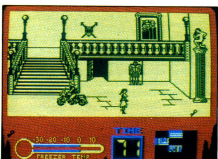
El Rocky Horror Show, es un juego emocionante, en el que el conocimiento de las distintas habitaciones y salas de la mansión es imprescindible para conseguir el éxito.

Descubrir la habitaciones secretas y utilizar el ascensor a nuestra conveniencia son armas claves.

También es importante saber los personajes que son mortales para nosotros y los que solamente rebajan nuestra energía.

El decorado de las distintas salas es muy acertado, con unos gráficos tipo cómic muy cuidados.

El movimiento del ascensor que en su ascenso y descenso, mediante un suave SCROLL, tiene un tratamiento idóneo y el efecto producido es muy bueno.



SOLOAMENTE  
CPC 664

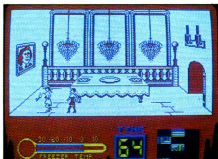
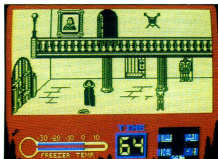
AMSTRAD



CRL SOFTWARE PRESENT  
A TIMEWARP PRODUCTION



**DON'T DREAM IT.**



## Mr. Joystick



Nuestro personaje, puede utilizar palancas y botones para abrir puertas, desactivar trampas, poner en marcha la calefacción y demás cosas que sólo se aprenden con un poco de práctica.

El tiempo es factor determinante, para SALVAR A BRAD disponemos de 100 unidades de éste, las cuales al agotarse provocan nuestra muerte.

Podemos elegir entre jugar con Janet o con Brad, que en determinados momentos pierde sus ropas quedándose completamente desnuda, un aditamento gracioso. Una aventura con buenos gráficos y de duración limitada, con lo cual, evitamos el tedio de ese tipo de aventuras interminables en las que nos dedicamos a vagar por parajes desconocidos en busca de no sé qué objetos y con un objetivo que tampoco conocemos, lo que después de varias horas de búsqueda nos induce a arrinconar el juego y dedicarnos a otra cosa.

# LOS DOS SON M LOS DOS SON

Spa



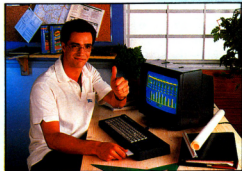
- 109.500 Ptas.  
(Monitor fósforo verde)
- 134.500 Ptas.  
(Monitor color).

#### EL AMSTRAD CPC 6128 INCLUYE EN SU SUMINISTRO:

- Teclado profesional de 74 teclas con unidad de disco 3" integrada.
- Monitor color o fósforo verde.
- Disco 3" con Sistema Operativo CP/M 2.2 y lenguaje Dr. LOGO.
- Disco 3" con Sistema Operativo CP/M Plus y utilidades.
- Disco 3" con seis programas de obsequio.
- Manuales en castellano.
- Garantía Oficial AMSTRAD ESPAÑA.

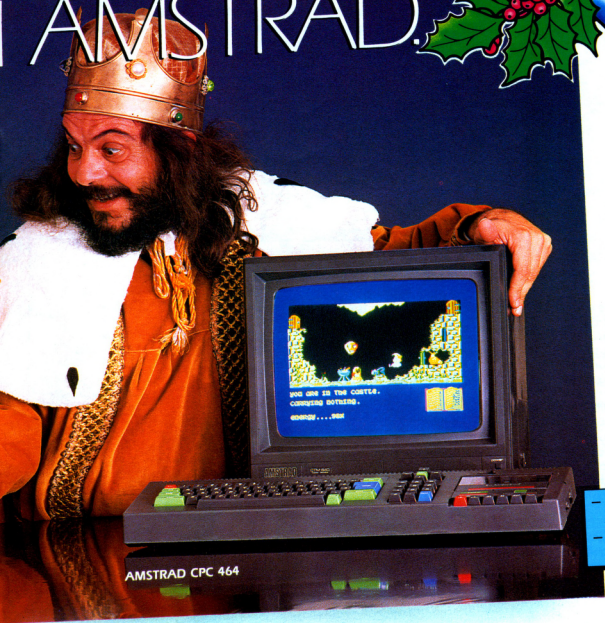
#### CARACTERÍSTICAS TÉCNICAS:

- Microprocesador Z80 (4MHz) con:
  - 64K RAM y 32K ROM (Mod. 464)
  - 128K RAM y 48K ROM (Mod. 6128)
- Gráfico de alta resolución de hasta 640 por 200 pixels.
- Unidad de cassette incorporada en el teclado (Mod. 464).
- Unidad de disco de 3" con 180K por cara integrada en el teclado (Mod. 6128) - [Opcional en el modelo 464].



# AMSTRAD ESPAÑA

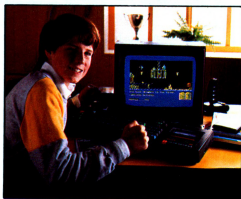
# MUY BUENOS AMSTRAD.



AMSTRAD CPC 464

- 66.900 Ptas.  
(Monitor fósforo verde)
- 95.900 Ptas.  
(Monitor color).

- Texto en pantalla de 20, 40 y 80 columnas.
- Sistemas Operativos AMSDOS, CP/M 2.2 y CP/M Plus (con la unidad de disco).
- Tres canales de sonido con 8 octavas moduladas independientemente. Altavoz interno y salida estéreo.
- Salidas: Centronics, cassette o unidad de disco externa (según modelo), PCB multiuso, joystick, etc.



#### EL AMSTRAD CPC 464 INCLUYE EN SU SUMINISTRO:

- Teclado profesional de 74 teclas con cassette integrado.
- Monitor color o fósforo verde.
- Ocho cassettes de programas.
- Libro "Guía de Referencia Basic para el Programador"
- Manuales en Castellano.
- Garantía Oficial AMSTRAD ESPAÑA.

Estas NAVIDADES HAY ORDENADORES MUY BUENOS... Y SON AMSTRAD

# LOS STOMPERS

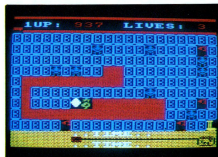
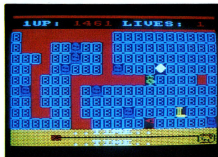
Los STOMPERS es un excelente y adictivo programa que incluye unas rutinas de scroll en máquina, aplicables a nuestros propios juegos, y una buena presentación que nos enseña a usar los caracteres gráficos.

# E

n la Tierra de Block, en el Norte de Blob, vivían felices un grupo de granjeros que cuidaban a los pájaros Wob-Wob para aprovechar sus huevos.

Sin embargo, un día, del Este de la Tierra de Grog, llegaron los diabólicos Stompers, completamente decididos a romperlos todos. Los granjeros, incomprensiblemente, no están muy de acuerdo, y han comisionado al intrépido Bod para que lo impida, armándole con misteriosos poderes que podrán ser usados en determinadas ocasiones.

Podríamos advertiros de los peligros a los que Bod debe enfrentarse, e incluso evitarlos, pero vamos a dejar que los descubráis vosotros mismos; total, unas vidas más o menos...



## TECLAS

Z	Rotación a la izquierda de Bod.
X	Rotación a la derecha.
/	Mueve a Bod.
	Desplaza bloques a la derecha.
\	Desplaza bloques a la derecha.

## VARIABLES

X	Coordenada x de Bod.
Y	Coordenada Y
Z	Coordenadas Y de Bod en el array inicial.
bootx	Coordenada X del enemigo.
booty	Coordenada Y del enemigo.
bootz	Coordenada Y del enemigo en el array inicial.
mh\$	Cabeza de los monstruos.
bh\$	Cabeza de Bod.
eh\$	Cabeza de los Stompers enemigos.
mb\$	Cuerpo del monstruo.
bb\$	Cuerpo de Bod.
eb\$	Cuerpo de los Stompers enemigos.
na\$	Nombre de los jugadores que han batido el récord.
hs	High scores.
sc	Scores.
level	Nivel de dificultad.
set (9,20)	Contenido de la pantalla.
eggs	Número de huevos recogidos.
ti	Tiempo que queda antes de...
bod	Dirección de Bod.
lives	Números de vidas que quedan.
pi	Cuál Stomper está presente.
monster	Cuál monstruo está presente.

## RUTINAS

180 Bucle principal	Realiza el salto a la rutina principal de movimiento y chequea si Bod está muerto.
330 Tiempo	Decrementa el tiempo disponible.
410 Mueve a Bod	Incluye rotaciones.
600 Izquierda	Desplazamiento de los bloques a la izquierda.
700 Derecha	
800 Una vida menos	Mira a ver si quedan vidas.
960 Huevos a salvo	¿Quedan huevos?
1070 Enemigo	Si el nivel < 7 mueve y reemplaza el escenario.
	Si es > mueve al enemigo y borra el escenario.
	Imprime las diversas pantallas.
1240 Pantalla	
1530 Pone huevos en pantalla	
1610 UDG	Caracteres gráficos
2010 Variables	Inicializa variables
2180 Instrucciones	
2650 Código máquina	Scroll de pantalla a derecha e izquierda.
2810 High Score	Pide tu nombre si rebasas la máxima puntuación.

```

10 REM Ab Bod!
20 REM By Aranello Chapman
30 REM (c)AMSTRAD SENNAL
40 REM
50 REM
60 DIM set(9,20)
70 DIM hs(B):DIM na$(B)
80 RESTORE 100
90 FOR f=1 TO B:READ hs(f),na$(f):IN
EXT f
100 DATA 5500,Mr,5000,Del,4000,4000,
5500,Yes,11,5000,Stomper,2500,0,
Ah,Bod!,2000,Bib's,1500,Revenge
110 GOSUB 1610:REM U.D.G's
120 GOSUB 2650:REM codigo maquina
130 GOSUB 2180:REM Instrucciones
140 GOSUB 2010:REM Variables
150 GOSUB 1240:REM Dibuja Pantalas
160 LET edgs=0
170 LOCATE bootx,booty:PEN 1:PRINT
MID$(eh$,p1,1):LOCATE bootx,booty+1
:PRINT MID$(eh$,p1,1):LOCATE 18,2:PF
EN 3:PRINT lives
180 REM*****Bucle Principal***
****
210 LOCATE 6,2:PFEN 3:PRINT sc
220 IF INKEY(71)=0 OR INKEY(63)=0 D
R INKEY(30)=0 THEN GOSUB 410
230 IF INKEY(31)=0 THEN GOSUB 600
240 IF INKEY(22)=0 THEN GOSUB 700
250 IF set(z,*)=2 THEN LET sc=sc+10
155 IF set(z,*)=3
260 IF set(z,*)=3 THEN GOTO 960
270 IF level>2 THEN GOSUB 1070
280 IF booty=y AND bootx=x THEN GOT
O 800
290 IF set(z,*)=0 OR set(z,*)=4 THE
N GOTO 800
300 GOSUB 330:REM disminuye tiempo
310 IF t1<=0 THEN FOR f=1 TO 5:OUN
D 1,0,1,0,7,0,0,1:INK 0,3:BORDER 3:
OUND 1,0,25,5,0,0,10:INK 0,0:BORDER
0:NEXT f:FOR f=1 TO 100:NEXT f:GOT
O 800
315 LET fr=PRE("")
320 GOTO 180
330 REM*****TIEMPO*****
****
360 LET t1=t1-(level/50)
380 LOCATE 19,t1,23:PAPER 3:PFEN 1:PR
INT CHR$(234):PFEN 0:PRINT CHR$(23
5):PAPER 0
400 RETURN
410 REM*****Huevo Bod*****
****
440 IF INKEY(30)=0 THEN GOTO 520
450 IF INKEY(63)=0 THEN LET bod=bod
-1
460 IF INKEY(71)=0 THEN LET bod=bod
+1
470 IF bod=0 THEN LET bod=4
480 IF bod=5 THEN LET bod=1
490 LOCATE x,y:PFEN 12:PRINT MID$(b
$,bod,1)
500 LOCATE x,y+1:PRINT MID$(bb$,bod
,1)
510 RETURN
520 LOCATE x,y:PAPER 3:PRINT " :LOC
ATE x,y+1:PRINT " :PAPER 0
530 IF bod=1 AND x<20 THEN LET set
z,x)=0:LET x=x+1:SC=SC+1
540 IF bod=2 AND y<20 THEN LET set
z,x)=0:LET y=y+2:LET z=z+1:SC=SC+1
550 IF bod=3 AND x<11 THEN LET set
z,x)=0:LET x=x-1:SC=SC+1
560 IF bod=4 AND y<4 THEN LET set(z
,*)=0:y=y-2:LET z=z-1:SC=SC+1
570 LOCATE x,y:PFEN 12:PRINT MID$(b
$,bod,1):LOCATE x,y+1:PRINT MID$(b
$,bod,1)
580 SOUND 1,2900,20,5,0,0,1
590 RETURN
600 REM*****Gira a izquierda***
****
650 FOR f=1 TO 4:CALL high,y-1:CALL
high,y:NEXT f
640 LET p3=set(z,1)
650 FOR f=1 TO 19:LET set(z,f)=set(

```

```

z,f+1):NEXT f:LET set(z,20)=p3
660 LET x=x-1:IF x=0 THEN LET x=20
670 IF booty=y THEN LET bootx=bootx
-1
680 IF bootx=0 THEN LET bootx=20
690 RETURN
700 REM*****Gira a derecha***
****
730 FOR f=1 TO 4:CALL high+50,y-1:CALL
high+50,y:NEXT f
740 LET p3=set(z,20)
750 FOR f=20 TO 2 STEP-1:LET set
(f)=set(z,f-1):NEXT f:LET set(z,1)=p
3
760 LET x=x+1:IF x=21 THEN LET x=x-1
770 IF booty=y THEN LET bootx=bootx
+1
780 IF bootx=21 THEN LET bootx=1
790 RETURN
800 REM*****Pierdes vida***
****
830 LOCATE bootx,booty:PAPER 3:PRIN
T " :LOCATE bootx,booty+1:PRINT "
:PAPER 0:LOCATE x,y:PFEN 12:PRINT MID
$(bb$,bod,1):LOCATE x,y+1:PRINT MID
$(bb$,bod,1)
834 IF bootz<0 THEN LET bootz=1
835 IF set(bootz,bootx)<3 THEN LET
set(bootz,bootx)=0 ELSE IF set(boo
tz,bootx)=3 THEN PFEN 4:LOCATE bootx
,booty:PRINT CHR$(222):LOCATE bootx
,booty+1:PRINT CHR$(223)
836 LOCATE 1,23:PAPER 3:PFEN 1:PRIN
T STRING$(18,CHR$(234)):PAPER 0:PFEN
1
840 LET lives=lives-1:LET t1=t1-B
850 IF lives=0 THEN GOTO 910
860 LET set(z,*)=1
870 FOR f=1 TO 100:OUT &B000,0:OUT
&B000,1:SOUND 130,0,50,15,0,1:INEX
T f:OUT &B000,0:OUT &B000,0:LOCATE
18,23:PAPER 3:PRINT CHR$(234)
:PRINT CHR$(22)+CHR$(1):LOCATE 2,23
:PAPER 3:PFEN 5:PRINT STRING$(17,CHR
$(154)):PAPER 0:PRINT CHR$(22)+CHR$(
0)
880 IF booty<y AND bootx<x THEN G
OTO 170
890 IF set(bootz,*)=20 THEN LET booty
=4:LET bootz=1:GOTO 170
900 LET booty=20:LET bootx=20:GOTO
170
910 FOR f=1 TO 2:OUT &B000,0:OUT &B
000,1:RESTORE 940:FOR g=1 TO 11:REA
D d,n:SOUND 1,n,d,7:SOUND 4,n+2,d,7
:SOUND 5,0,3,0:NEXT g,f
920 OUT &B000,0:OUT &B000,0
930 LOCATE 1,1:SOUND 130,0,50,3,0,0
,1:PRINT STRING$(26,1)
940 DATA 50,1016,37,1016,12,1016,50
,1016,25,850,25,899,25,899,25,1016
,25,1016,25,1136,100,1016
950 GOTO 2820
960 REM*****Aterriza en huevo***
****
990 LET eggs=eggs+1
1000 SOUND 2,200,25,7,0,4:SOUND 2,2
00,25,5,0,1:LET sc=sc+100:LET set(z
,*)=1
1010 IF eggs=level+3 THEN GOTO 1040
1020 GOSUB 1530
1030 GOTO 180
1040 LET sc=sc+INT(t1)*10:LET level
=level+1
1050 LET t1=t1-B
1060 GOTO 180
1070 REM*****Huevo Enemigo***
****
1091 IF bootz<0 THEN LET bootz=1
1100 IF level>6 AND set(bootz,bootx
)=1 THEN LOCATE bootx,booty:PAPER 3
:PRINT " :LOCATE bootx,booty+1:PRIN
T " :LET set(bootz,bootx)=0:PAPER 0
:GOTO 1160
1110 IF set(bootz,bootx)=0 THEN LOC
ATE bootx,booty:PAPER 3:PRINT " :LO
CATE bootx,booty+1:PRINT " :PAPER 0
1120 IF set(bootz,bootx)=1 THEN LOC

```

# Serie Oro

```

ATE bootx,booty:PFEN 10:PRINT CHR$(2
00):LOCATE bootx,booty+1:PRINT CHR$(
201)
1130 IF set(bootz,bootx)=2 THEN PFEN
3:LOCATE bootx,booty:PRINT CHR$(21
2):LOCATE bootx,booty+1:PRINT CHR$(
213)
1140 IF set(bootz,bootx)=3 THEN PFEN
4:LOCATE bootx,booty:PRINT CHR$(22
4):LOCATE bootx,booty+1:PRINT CHR$(
223)
1150 IF set(bootz,bootx)=4 THEN PFEN
monster+5:LOCATE bootx,booty:PRINT
MID$(mh$,monster,1):LOCATE bootx,b
ooty+1:PRINT MID$(mh$,monster,1)
1160 LET MOV=INT(RND#4)+1
1170 IF MOV=1 AND bootx<20 THEN LET
bootx=bootx+1
1180 IF MOV=2 AND booty<20 THEN LET
booty=booty+2:LET bootz=bootz+1
1190 IF MOV=3 AND bootx>1 THEN LET
bootx=bootx-1
1200 IF MOV=4 AND booty>4 THEN LET
booty=booty-2:LET bootz=bootz-1
1210 LOCATE bootx,booty:PFEN 1:PRIN
T MID$(eh$,p1,1):LOCATE bootx,booty
+1:PRINT MID$(eh$,p1,1)
1220 SOUND 1,266,5,7:SOUND 1,399,5,
7
1230 RETURN
1240 REM*****Dibuja pantalaa***
****
1270 LET y=10:LET x=10:LET z=4:LET
bootx=20:LET booty=20:LET bootz=9
1280 PAPER 0:BORDER 0:CLS
1290 LET monster=INT(RND#4)+1:LET p
=INT(RND#4)+1
1300 FOR f=1 TO 9:FOR g=1 TO 20:LET
set(f,g)=1:NEXT g,f
1310 FOR f=1 TO 6+1:level
1320 LET x=INT(RND#20)+1:LET y=IN
T(RND#9)+1
1330 IF x=10 THEN GOTO 1320
1340 LET set(y,1)=4
1350 NEXT f
1360 FOR f=1 TO 3+1:level
1370 LET x=INT(RND#20)+1:LET y=IN
T(RND#9)+1
1380 IF set(y,1)=1 THEN LET set(y
,1)=2
1390 NEXT f
1400 LET count=1
1410 FOR f=4 TO 20 STEP 2:FOR g=1 T
O 20
1420 IF set(count,g)=1 THEN LOCATE
g,f:PFEN 10:PRINT CHR$(200):LOCATE
g,f+1:PRINT CHR$(201)
1430 IF set(count,g)=2 THEN LOCATE
g,f:PRINT CHR$(212):LOCATE g,f+1
:PRINT CHR$(213)
1440 IF set(count,g)=4 THEN LET mon
ster+5:LOCATE g,f:PRINT MID$(mh$,m
onster,1):LOCATE g,f+1:PRINT MID$(m
h$,monster,1)
1450 NEXT g:LET count=count+1:NEXT
f
1460 LOCATE x,y:PFEN 12:PRINT MID$(b
h$,bod,1):LOCATE x,y+1:PRINT MID$(b
h$,bod,1)
1470 LOCATE bootx,booty:PFEN 1:PRIN
T MID$(eh$,p1,1):LOCATE bootx,booty
+1:PRINT MID$(eh$,p1,1)
1480 LOCATE 1,2:PFEN 2:PRINT " LUP:
TAB(12):LIVES:
1490 LOCATE 1,1:PFEN 3:PRINT STRING$

```

```

(20,CHR*(131));LOCATE 1,3:PRINT STR
IND*(20,CHR*(140))
1500 LOCATE 1,22:PAPER 3:PEN 1:PRIN
T STRING(59,CHR*(234));LOCATE 20,2
4:PRINT CHR*(234):PAPER 0
1510 LOCATE 1,23:PRINT CHR*(22)+CHR
*(11):LOCATE 2,23:PEN 5:PRINT STRING
*(17,CHR*(154));LOCATE 19,23:PRINT
CHR*(230):CHR*(232):LOCATE 19,24:P
RINT CHR*(231):CHR*(234);LOCATE 7,2
4:PEN 4:PRINT ". TIME..":LOCATE 7,22
:PRINT ". TIME..".
1520 LOCATE 1,23:PRINT CHR*(22)+CHR
*(0)
1530 REM*****Pone huevo en pantal
a****
1540 LET y1=INT(RND*9)+1:LET x1=INT
(RND*20)+1:IF y1=20 AND x1=20 THEN
GOTO 1560
1570 IF set(y1,x1)=1 THEN SET set(y
1,x1)=3 ELSE GOTD 1560
1580 LET count:=1:FOR f#=2 TO 20 ST
EP 2:FOR g#1 TO 20
1590 IF set(count,g)=3 THEN PEN 4:
OCATE g,f:PRINT CHR*(222):LOCATE g,
f+1:PRINT CHR*(223) ELSE NEXT g:LET
count:=count+1:NEXT f
1600 RETURN
1610 REM*****BUL D.*****
****
1640 SYMBOL AFTER 99
1650 SYMBOL 200,126,255,129,185,165
,165,165,185
1660 SYMBOL 201,185,165,165,165,185
,129,255,126
1670 SYMBOL 202,0,248,248,80,112,80
,112,80
1680 SYMBOL 203,112,80,112,80,254,2
55,255,126
1690 SYMBOL 204,24,24,126,44,34,36,
24,60
1700 SYMBOL 205,110,111,103,119,62,
24,24,28
1710 SYMBOL 206,28,28,127,62,62,28,
8,28
1720 SYMBOL 207,62,127,127,127,62,2
8,20,54
1730 SYMBOL 208,28,28,127,54,42,34,
28,8
1740 SYMBOL 209,28,62,93,62,8,20,54
1750 SYMBOL 210,24,24,126,52,68,36,
24,60
1760 SYMBOL 211,118,246,236,328,124
,24,48
1770 SYMBOL 212,3,15,63,115,247,115
,55,15
1780 SYMBOL 213,3,1,1,1,1,1,3,7
1790 SYMBOL 214,24,60,60,126,94,74,
82,122
1800 SYMBOL 215,126,126,153,195,255
,255,255,155
1810 SYMBOL 216,0,195,66,66,90,60,9
0,153
1820 SYMBOL 217,219,126,60,24,24,36
,36,231
1830 SYMBOL 218,129,126,153,24,60,9
0,189,255
1840 SYMBOL 219,219,102,60,24,102,3
6,36,231
1850 SYMBOL 220,24,60,126,219,102,6
0,24,255
1860 SYMBOL 221,189,189,153,153,60,
126,255,255
1870 SYMBOL 222,24,60,60,126,126,25
5,255,255
1880 SYMBOL 223,126,126,60,60,24,0,
0,0
1890 SYMBOL 224,24,24,126,255,153,2
55,219,102
1900 SYMBOL 225,189,195,126,24,24,2
4,24,126
1910 SYMBOL 226,153,36,46,69,210,14
6,74,131
1920 SYMBOL 227,149,210,97,45,42,18
,156,73
1930 SYMBOL 228,28,62,107,255,247,9
1,99,254

```

```

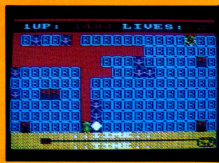
1940 SYMBOL 229,195,255,255,126,60,
28,12,6
1950 SYMBOL 230,0,31,32,223,192,64,
92,72
1960 SYMBOL 231,74,75,66,66,32,31,0
,0
1970 SYMBOL 232,0,248,4,250,2,58,18
,18
1980 SYMBOL 233,18,66,194,66,4,248,
0,0
1985 SYMBOL 234,146,255,73,255,146,
255,73,255
1990 SYMBOL 235,132,74,40,151,95,40
,68,146
1995 SYMBOL 236,198,165,198,165,6,4
0,90,16
2000 RETURN
2010 REM*****Variables*****
****
2040 ENT 1,5,1,1,10,-1,1,10,1,1,10,
1,5,1,1,5:ENV 3,3,2,2,3,-2:ENT 4,
5,-10,10,1,0,5
2050 LET ah=CHR*(214)+CHR*(216)+CH
R*(218)+CHR*(220)
2060 LET eb=CHR*(215)+CHR*(217)+CH
R*(219)+CHR*(221)
2070 LET bh=CHR*(204)+CHR*(208)+CH
R*(210)+CHR*(206)
2080 LET bb=CHR*(205)+CHR*(209)+CH
R*(211)+CHR*(207)
2090 LET bod=1
2100 LET eh=CHR*(202)+CHR*(224)+CH
R*(226)+CHR*(228)
2110 LET eb=CHR*(203)+CHR*(225)+CH
R*(227)+CHR*(229)
2120 LET liv=3
2130 LET sc=0
2140 LET level=1
2150 MODE 0
2160 LET ti=18
2170 RETURN
2180 REM*****Instrucciones*****
****
2200 BORDER 0:GOTO 2290
2220 MODE 1:PEN 3:PRINT STRING*(40,
CHR*(200)):LOCATE 1,2:PRINT STRING*
(40,CHR*(201))
2230 LOCATE 1,3:PRINT CHR*(200):TAB
(16):PEN 2:PRINT"¡AH BOD!" :TAB(40)
:PEN 3:PRINT CHR*(200)
2240 LOCATE 1,4:PRINT CHR*(201):STR
ING*(38,CHR*(200)):CHR*(201):LOCATE
2,5:PRINT STRING*(38,CHR*(201))
2245 IF f#=1 THEN RETURN
2250 LOCATE 1,20:PRINT STRING*(40,C
HR*(200)):LOCATE 1,21:PRINT STRING*
(40,CHR*(201))
2260 LOCATE 1,22:PRINT CHR*(200):TA
B(8):PEN 1:PRINT"Pulsa «ESPACIO»
para continuar.":TAB(40):PEN 3:PRIN
T CHR*(200)
2270 LOCATE 1,23:PRINT CHR*(201):ST
RING*(38,CHR*(200)):CHR*(201):LOCAT
E 2,24:PRINT STRING*(38,CHR*(201))
2280 RETURN
2290 INK 0,0:GOSUB 2220
2300 LOCATE 1,7:PEN 2
2310 PRINT" En la tierra de Block
al norte de Blob los malvados Stom
pers estan rompiendo los huevos de

```

```

los pajaros Mob-Mob que cuidan los
granjeros."
2320 PRINT" Unos dias despues de a
pazero los Stompers, los granjeros
estaban tristes y decidieron ir a c
asa a jugar con sus AMSTRAD (Habia
un antiguo modelo para granjeros)."
2330 PRINT" Como Bod no tenia AMST
RAD, para hacer algo interesante dec
idio salvar a los pajaros Mob-Mob
".
2340 PRINT"Ayudale en su tarea."
2350 IF INKEY<>"" THEN GOTD 2350
2360 WHILE INKEY<>"" :WEND
2370 LOCATE 1,1:PRINT STRING*(26,1)
)
2380 GOSUB 2220
2390 LOCATE 1,7:PEN 2
2400 PRINT" Bod debe recoger todos
los huevos de la pantalla.Los huev
os aparecen uno cada vez y tu debes
coger todos antes de que arda toda
la mecha de la INT."
2410 PRINT" En la pantalla hay ban
deras con las que consigues puntos
extras recogindolas. Hay tambien ri
esgos en la tierra de Block.Los res
tantes Stompers estan alrededor de
Bod.Despues de limpiar dos pantalla
s un Stomper mutante empieza a atac
ar."
2420 PRINT" Cada vez que Bod se m
ueve, el bloque donde estaba parado
desaparece dejando un agujero vacio
que debes evitar."
2430 IF INKEY<>"" THEN GOTD 2430
2440 WHILE INKEY<>"" :WEND
2450 LOCATE 1,1:PRINT STRING*(26,1)
)
2460 GOSUB 2220
2470 LOCATE 1,7:PEN 2
2480 PRINT" Como no es el dia de B
od,huye de la granja de huevos.Solo
tiempo raras poderas en la tierra d
e los bloques.Puede mover la fila d
e bloques en la que esta parado a l
a izquierda o a la derecha."
2490 PRINT" Bod no puede andar fue
ra de los bloques.Como puede girar
en las raras poderas en la tierra d
e aparece en otro."
2500 PRINT" Sumario-Recoge huevos
y banderas.Elude los monstruos movi
endote o parandote.No pierdas tiemp
o alrededor de ellos y no intentes
cruzar agujeros vacios."
2510 IF INKEY<>"" THEN GOTD 2510
2520 WHILE INKEY<>"" :WEND
2530 LOCATE 1,19:PRINT STRING*(26,1)
)
2540 GOSUB 2220
2550 LOCATE 16,7:PEN 2:PRINT"LAS TE
CLAS":LOCATE 16,8:PEN 1:PRINT STRIN
G*(10,CHR*(131))
2560 PEN 2:PRINT TAB(17):PRINT"Dira
r":PRINT"izquierda-" :TAB(26):"De
recha-" :X"
2570 PEN 3:PRINT TAB(15):"Moverse"
"/.
2580 PEN 1:PRINT:PRINT TAB(4):" Nu
eve Bloques a izquierda-" :X"
2590 PEN 2:PRINT:PRINT TAB(4):" Nu
eve Bloques a derecha-" :X"
2600 LOCATE 14,19:PEN 1:PRINT"¡BUEN
A-SUERTE!"
2610 IF INKEY<>"" THEN GOTD 2610
2620 WHILE INKEY<>"" :WEND
2630 LOCATE 1,1:PRINT STRING*(26,1)
)
2640 GOTD 2820
2650 REM*****Codigo Maquina*****
****
2680 LET hig=HIGHMEM:MEMORY hig:LET
a=dress+high+1500:LET high=high+1500:R
ESTORE 2740
2690 READ n
2700 LET sc=sc+n:IF n=999 THEN READ
n:IF n<>sc THEN PRINT CHR*(7):PRIN

```



```
T STRINGS(6,CHR*(7)):CLS:LOCATE 11,
10:PEN 3:PRINT"ERROR IN MACHINE COD
E="+END ELSE IF n=16121 THEN LET sc=
0:REM
2710 POKe address,n
2720 LET address=address+1
2730 GOTO 2690
2740 REM*****SCROLL IZQUIERDO*
```

```
****
2750 DATA $dd,$ee,$00,$26,$00,$cd,$
1a,$bc,$e5,$7e,$f5,$54,$5d,$eb,$23,$
fcb,$f4,$cb,$fc,$0e,$4f,$7e,$12,$23
,$cb,$f4,$cb,$fc,$eb,$23,$cb,$f4,$cb
,$fc,$eb,$0d
2760 DATA $20,$ef,$f1,$12,$e1,$7c,$
c6,$0b,$67,$ee,$3b,$20,$d7,$c9
2770 REM*****SCROLL DERECHO****
****
```

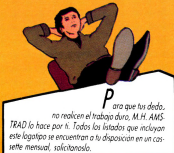
```
2780 DATA $dd,$ee,$00,$26,$00,$cd,$
1a,$bc,$01,$4f,$00,$09,$cb,$4f,$cb,$
fcb,$e5,$7e,$f5,$54,$5d,$2b,$cb,$f4
,$cb,$fc,$0e,$4f,$7e,$12,$2b,$cb,$f4
,$cb,$fc,$eb,$2b,$cb,$f4
2790 DATA $cb,$fc,$eb,$0d,$20,$ef,$
f1,$12,$e1,$7c,$c6,$0b,$67,$ee,$3b,$
$20,$d8,$c9
2800 DATA 999,16121
2810 REM*****MAX.PUNTUACION****
****
```

```
2820 MODE 1
2830 LET f1=1:GOSUB 2220:f1=0
2840 LOCATE 13,3:PRINT"MAXIMA
PUNTUACION":PEN 3
2850 FOR f=6 TO 14 STEP 2:LOCATE 4
,f:PRINT CHR*(200):LOCATE 4,f+1:PRI
NT CHR*(201):LOCATE 37,f:PRINT CHR*
(200):LOCATE 37,f+1:PRINT CHR*(201)
:NEXT f
2860 LOCATE 5,16:PRINT STRING$(32,C
HR*(201))
2870 FOR f=1 TO 8
2880 IF sc>hs(f) THEN GOSUB 2980:f=
10
```

```
2890 NEXT f
2900 FOR f=1 TO 8:PEN 1:LOCATE 8,f+
6:PRINT na$(f):LOCATE 18,f+6:PEN 3:
PRINT".....":hs(f):NEXT f
2910 LOCATE 1,17:PRINT STRING$(220,
" ")
2920 IF INKEY("<")="" THEN GOTO 2920
2930 PEN 2:LOCATE 1,20:PRINT STRING
$(40,CHR$(154)):LOCATE 1,22:PRINT S
TRINGS(40,CHR$(154)):PEN 1:
2940 LOCATE 1,23:PRINT STRING$(40,
" ")
2950 LOCATE 11,21:PEN 3:PRINT"PULSA
":PEN 1:PRINT"< ESPACIO >":PEN 3:
PRINT"PARA JUGAR.":PEN 1
2960 IF INKEY(47)<>0 THEN GOTO 2960
2970 LOCATE 1,1:PRINT STRING$(26,11
):GOTO 140
2980 LET a$="ABCDEFGHIJKLMNORSTUV
WXYZ.#%&()!()?'*+~CHR*(236)
2990 LET c=19:LOCATE 1,20:PEN 1:PRI
NT a$
3000 LOCATE 3,17:PEN 1:PRINT"USAR
CURSOR IZQUIERDO,DERECHO Y COPY":LO
CATE 4,18:PRINT"PARA SELECCIONAR LE
TRAS.(AB. DE 10.):PEN 2:LOCATE 1,
19:PRINT STRING$(40,CHR$(154)):LOCA
```

# Serie Oro

```
TE 1,22:PRINT STRING$(40,CHR$(154))
3010 LOCATE 12,23:PEN 3:PRINT"PULSE
":PEN 1:PRINT"< * >":PEN 3:PRINT
"PARA SALIR.":PEN 1
3020 LET x$=""
3030 FOR z=1 TO 10
3040 LOCATE c,21:PEN 2:PRINT" "
3050 IF INKEY(1)=0 AND c<40 THEN LE
T c=c+1
3060 IF INKEY(8)=0 AND c>1 THEN LET
c=c-1
3070 IF INKEY(9)=0 AND c=40 THEN LO
CATE 7,f+6:PRINT" "
3080 IF INKEY(63)=0 THEN LET z=11:G
OTO 3120
3090 IF INKEY(9)<>0 THEN LOCATE c,2
1:PRINT"*":FOR a=1 TO 50:INLOCATE a:GOT
0 3040
3100 LET x$=x$+MID$(a$,c,1):LOCATE
7+z,f+6:PEN 1:PRINT MID$(a$,c,1)
3110 FOR a=1 TO 200:NEXT a
3120 NEXT z
3130 IF z=101 THEN GOTO 3020
3140 LET hs(8)=sc:LET na$(8)=x$
3150 LET f=0
3160 FOR f=1 TO 7
3170 IF hs(2)<hs(z+1) THEN LET t=hs
(z+1):LET hs(z+1)=hs(z):LET hs(z)=t
:LET a$=na$(z+1):LET na$(z+1)=na$(z
):LET na$(z)=a$:LET f=f+1
3180 NEXT z
3190 IF f=1 THEN GOTO 3150
3200 LET fr=FRE("")
3210 RETURN
```



**P**ero que tu dedo,  
no realicen el trabajo duro. M.H. AMS  
TRAD lo hace por ti. Todos los listados que incluyen  
este logotipo se encuentran a tu disposición en un cas-  
sette mensual, solicitálo.

## MANTENGA SU AMSTRAD COMO NUEVO CON ESTA PRACTICA FUNDA.

POR SOLO: **2.250** ptas.

- También disponemos de fundas para:
- SEIKOSHA SP-800 ..... 1.100 ptas.
- SEIKOSHA SP-1000 ..... 1.100 ptas.
- SPECTRUM 16/48 ..... 330 ptas.
- SPECTRUM PLUS ..... 460 ptas.
- COMMODORE 64 y VIC 20 ..... 680 ptas.
- SAGA 1 EMPEROR ..... 550 ptas.



## INTERESANTES CONDICIONES PARA DISTRIBUIDORES

CUPON DE PEDIDO

Recorte o copie este cupón y envíelo hoy mismo a:  
**MICROSOFT-HARD**, Apartado 23.406, 08080-BARCELONA, o bien realice su pedido (firma)  
 por teléfono llamando al (93) 348 04 07. (Tardes de 5 a 9). Gastos de envío: 150 ptas.  
 Deseo recibir a vuelta de correo ..... funda(s).  
 Marca: ..... Modelo: 464-664-6128. Monitor: verde-color  
 —Contra reembolso: ..... —Talón adjunto a MICROSOFT-HARD, SL:  
 (MARQUE CON UNA «X» LAS OPCIONES DESEADAS)  
 —NOMBRE Y APELLIDOS .....  
 —DOMICILIO .....  
 —LOCALIDAD ..... COD. POSTAL .....  
 —PROVINCIA .....

# DISEÑANDO SUS SPRITES

*Hasta el momento hemos visto cómo se organiza la memoria de pantalla en Modo 0 y hacia el final del último artículo habíamos desarrollado por completo una poderosa rutina de impresión de sprites. Pero no hemos llegado al final de la historia, todavía nos queda por recorrer un largo camino.*

**E**sta semana daremos el listado de una rutina de movimiento más un editor de gráficos que nos permitirá diseñar nuestros propios sprites.

El editor está en Basic, pero para escribir el código máquina necesitamos un ensamblador. El publicado en el número 8 de M.H. **AMSTRAD** es muy adecuado para tratar los programitas que vamos a escribir.

Si ha tecleado los ejemplos del último artículo, tendrá una sencilla rutina de escritura que es capaz de imprimir un carácter multicolor de cualquier tamaño y en cualquier lugar de la pantalla.

Esta sencilla rutina almacena los datos del sprite directamente en la memoria de pantalla. Haciéndolo de esta forma se produce un efecto secundario inoportuno ya que se borra cualquier cosa que está en esa posición que no tenga nada que ver con el sprite. Necesitamos modificar un poco la rutina para poder superar este problema.

## Mejorar las rutinas de impresión

Sin embargo, antes de nada vamos a echar un vistazo a los programas de esta semana y veremos lo que hacen muy por encima. Será más fácil seguir la explicación una vez hayamos visto la ejecución del código máquina.

El programa 1 es un ejemplo de sprite para utilizarlo con el programa en código máquina. Necesitamos teclearlo y correrlo antes de pasar a

ejecutar el programa 2. Se trata de un hombre paseando a su perro por la playa en un día soleado y con las olas al fondo. Si no nos cree, ¡hágalo y verá!

Lo hemos creado utilizando el editor de sprites —programa 3— tiene 2 caracteres de ancho por 3 de alto y utiliza 10 plumas y tintas diferentes.

No es la clase de sprites que vamos a encontrar normalmente, sirve para mostrarnos la velocidad y potencia de la rutina. Cuando lo haya visto, imagínese intentando moverlos e imprimirlos desde el Basic. Nosotros ni siquiera lo probaríamos.

Salve el programa 1 después de ejecutarlo y meta el programa 2. Para ejecutar la demostración del sprite teclee:

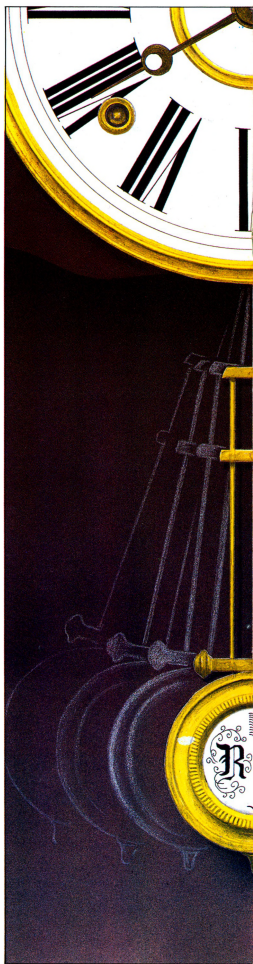
MODE 0: CALL &8000

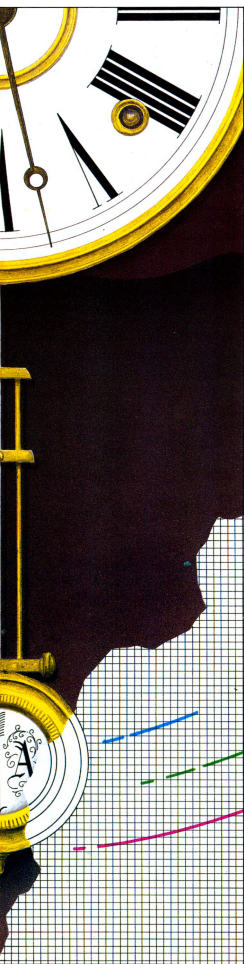
y utilice las teclas del cursor para mover el gráfico a lo largo y ancho de la pantalla. Podemos volver al Basic en cualquier momento pulsando la tecla ESC.

## La rutina de impresión

Echemos una mirada a la rutina de impresión y nos encontraremos con que es diferente a la desarrollada en nuestro último artículo. Comienza en la dirección &8087 dentro del programa 2 y está dividida en dos partes.

La primera borra el sprite antiguo y comienza en la dirección &8087. La segunda imprime el gráfico en la nueva posición y está en la dirección &80AD. La segunda parte es utilizada también por la rutina de inicialización para colocar el carácter en la pantalla.





F. L. Frontan

Esta última rutina de escritura utiliza la instrucción XOR para imprimir el sprite. Se recoge cada elemento de datos del sprite y se le hace un XOR con lo que haya en la memoria de pantalla «bucle 2» comprendido entre las direcciones &8098 y &809D en el programa 2 se nos demuestra como se hace.

La gran ventaja de utilizar la instrucción XOR es que después que se haya movido el sprite de un lugar, la pantalla queda de la forma que estaba inicialmente.

En el artículo «Animación Plus» del número 9 examinábamos la utilización de las funciones lógicas tales como XOR. Si no está muy seguro de lo que hace esta instrucción, vale la pena que lo revise.

Básicamente, cuando hacemos un XOR a dos bytes estamos haciendo una comparación bit a bit y el resultado está de acuerdo con las siguientes normas:

```
1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1
0 XOR 0 = 0
```

Si el byte de la pantalla es &00 y el dato del sprite es &CC haciendo XOR de los dos obtenemos &CC. Cuando el sprite se mueve hacemos otra vez XOR del dato con la memoria de pantalla que como ahora tiene el valor &CC, hallado anteriormente, al realizar esta función lógica de nuevo nos da el valor &00 (&CC XOR &CC=0) que es el valor que tenía antes.

### Imprimir sin borrar

Supongamos que el byte de la pantalla es &0C que el byte del dato es &CC. Cuando hagamos XOR de estos dos valores obtendremos &C0 que no es lo que queremos. Queremos decir con esto que cuando coloquemos un sprite sobre un color de fondo que no es 0 sus colores se alteran.

Al hacer otra vez &C0 XOR &CC obtenemos de nuevo &0C con lo que el color del fondo vuelve a ser el mismo que tenía inicialmente.

Para comprender mejor lo que le acabamos de decir, escriba varios caracteres en la pantalla antes de llamar a la rutina del sprite. A continuación mueva el sprite sobre ellos y lo verá más claramente.

## Código máquina

El problema del color se soluciona fácilmente redefiniendo las tintas para que produzcan los colores correctos. Para una mayor información ojea el sencillo artículo dedicado a los sprites en Basic publicado en el número 11 dentro de nuestra habitual sección PrograAcción.

Antes de llamar a «**sprint**», la nueva dirección del sprite se coloca en «**nueva**» y la dirección anterior en «**vieja**». Observemos que la instrucción de la posición &808B es LD HL, o tiene tres bytes, y que «**vieja**» se coloca en &808C, el segundo de los bytes, al comienzo del programa 2.

Esto quiere decir que los dos bytes que están a cero en la instrucción serán sustituidos por los distintos valores que vaya teniendo la antigua dirección del sprite.

Este método nos ahorra tiempo ya que LD HL,nn es más rápida que LD HL,(nn). Utilizamos la misma técnica para el tratamiento de alguna de las otras variables.

El tamaño de sprite en columnas y filas lo pasamos en los registros H y L. El diseñador de gráficos nos dará el tamaño.

Hay también un par de variables dignas de mención. Se trata de «**datnue**» y «**datvie**» que hacen posible la animación de caracteres asociando «**datvie**» al sprite 1 y «**datnue**» al sprite 2. Hacemos XOR al sprite de la antigua posición con «**datvie**» y colocamos «**datnue**» en «**nueva**».

Al principio vimos dos artículos en los que la pantalla estaba compuesta por filas de pixels repartidas en grupos de ocho —las líneas con las que trabajan las instrucciones LOCATE y PRINT—, y que la dirección de cada fila de pixels de un grupo está separado &800 bytes de la anterior.

La rutina de escritura lleva la dirección del sprite en la pareja de registros HL así que tenemos que sumarle &800 para obtener la fila siguiente. Es cosa de expertos pasar H al acumulador A, sumarle &8 y cargar el resultado en el registro H. Es más rápido que cargar la pareja BC con &800 y luego sumar HL y BC. Con el tiempo llegará a dominar trucos parecidos a éste.



guarda en el stock así que mediante la instrucción POP cargamos con su valor el registro HL. Como hay dos pixels horizontales por byte en Modo 0, la dirección se decrementa en &1000 (2\*800). El sprite se mueve hacia arriba dos filas de pixels del mismo modo que cuando se mueve horizontalmente.

Para restar &1000 de HL todo lo que se necesita es restar &10 de H cargando el contenido de H en el registro A. Una compensación de esto es que colocamos en A una copia del byte más significativo de la dirección.

La memoria de pantalla comienza en &C000 así que podemos restar &C0 de A, el byte más significativo de la dirección del sprite.

Si se activa el Carry flag es que la nueva dirección está por encima de la parte superior de la pantalla, por lo tanto hemos de sumar un factor de corrección al registro HL.

Esta situación ocurre entre grupos de ocho pixels a causa de la forma en que el **Amstrad** mapea la memoria de pantalla.

El programa vuelve otra vez al comienzo para dibujar el sprite en la nueva posición. Observemos que hay una llamada a &BD19 justo antes de la llamada a la rutina de impresión. Así evitamos el parpadeo producido por el retorno del haz en el monitor.

## Editor de sprites

El programa 3 es un editor de sprites. Contiene todas las instrucciones y su utilización es bastante sencilla. El diseñador puede contener dos sprites, de 2 caracteres de ancho por 3 de alto, a la vez para permitirle trabajar como si se trata de una secuencia de animación.

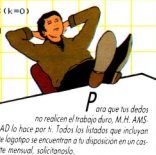
Hay comandos que nos dan en el reflejo del sprite horizontal y vertical, y también podemos girarle 90 grados.

Cuando tengamos completo el diseño, el programa escribe los datos del sprite como instrucciones DATA con su número de línea. Cada uno de los datos pueden salvarse directamente o finalizar el programa y si queremos teclear las instrucciones DATA que nos deja en la pantalla.

Y nada más por hoy. Diviértase diseñando sprites mientras nosotros nos dedicamos a conseguir algunas rutinas que nos sirvan para detectar choques entre ellos. Nos veremos pronto...

10 REM	0 TO 100:NEXT character#=UPPER#
20 REM Diseñador de Sprites	(INKEY#):GOSUB 990:PLOT 58:#+1
30 REM For R.A.W.	6,338-y#B,15
40 REM MICROHOBBY AMSTRAD	690 IF character#CHR#(6E) THEN
50 REM SEPPANAL	N GOSUB 1060:sdato(x,y)=color:
60 REM	open=color:GOSUB 990
70 MEMORY 18000	700 IF character#" THEN color
80 GOSUB 2130:REM instrucciones	R=(color+1) MOD 16:PEN color:
90 GOSUB 170:REM inicializaci	OCATE 6,24:PRINT MID\$(STR\$(col
on	or),2):":PEN 15
100 GOSUB 360:REM pantalla	710 IF character#"V THEN GOSU
110 GOSUB 610:REM diseño	B 1410
120 MODE 1	720 IF character#"H THEN GOSU
130 CALL MB4E:CALL MB02:REM	B 1570
puesta a cero	730 IF character#"N THEN GOSU
140 SPEED KEY 25,2	B 1130
150 END	740 IF character#"G THEN GOSU
160 REM -----	B 1740
-----	750 IF character#"S THEN GOSU
170 REM Inicializacion de vari	B 1930
ables	760 IF character#"I THEN GOSU
180 REM -----	B 830
190 REM -----	770 IF character#CHR#(13) THEN
-----	GOSUB 2370
200 SPEED KEY 250,250	780 IF character#CHR#(127) THE
210 DEFINIT a-z	N IF sdato(x,y) THEN open=dat
220 FOR i=1 TO 6	O(x,y):GOSUB 990:sdato(x,y)=0
230 READ a#	790 x#=(INKEY(1)-1) AND #0:Y#(15)
240 POKE 18000+i,VAL("%"+a#)	+(INKEY(8))-1 AND #0:Y#Y#-1
250 NEXT	AND #0:Y#Y#+1 AND #0:Y#Y#+1
260 paper=#8001	800 WEND
270 DATA 00,76,00,C3,E4,BB	810 RETURN
280 DIM sdato(15,23),sdato2(15	820 REM -----
,23),temp(15,23)	830 REM -----
290 FOR i=0 TO 15	840 REM -----
300 READ j:INK i,j	-----
310 NEXT	850 i:=1:j:=1
320 DATA 0,18,6,24,2,8,20,26,1	860 WHILE i=1
5,16,7,9,13,22,2,26	870 LOCATE 2,20:INPUT "Pen: "
330 color:=isprite#1	880 IF <0 OR >15 THEN i=-1
340 RETURN	890 WEND
350 REM -----	900 LOCATE 2,20:PRINT SPC(18)
-----	910 WHILE j=1
360 REM Pantalla principal	920 LOCATE 2,20:LINE INPUT "In
370 REM -----	k "?":a#
-----	930 IF LEN(a#) THEN j=VAL(a#):
380 MODE 0:BORDER 4	IF <0 OR >26 THEN j=-1
390 MOVE 0,0:DRAW 0,398,14	940 LOCATE 2,20:PRINT SPC(18)
400 DRAW 638,398:DRAW 638,0:DR	950 WEND
AW 0,0	960 IF INSTR(a#,"") THEN k=VA
410 CALL paper,14:PLOT -10,-1	L(MID\$(a#,INSTR(a#,"")+1)):IN
0,15	K i,j,k ELSE INK i,j
420 TAG:MOVE 0,365	970 RETURN
430 PRINT "DISEÑADOR DE SPRITE	980 REM -----
S: "	-----
440 TAGOFF	990 REM Dibujar un pixel
450 CALL paper,0:PAPER 0:PEN	1000 REM -----
13	-----
460 LOCATE 7,22:PRINT "Sprites "	1010 PLOT 480:#44,271-y#2,open
470 PRINT USING "#":sprite	1020 TAG:MOVE 51:#416,342-y#B
480 LOCATE 2,24:PRINT "Pen: "	1030 PRINT CHR#(129):TAGOFF
490 PEN color	1040 RETURN
500 PRINT MID\$(STR\$(color),2)	1050 REM -----
510 MOVE 32,350:DRAW 320,350,1	-----
3	1060 REM Chequear si el pixel
520 DRAW 320,140:DRAW 32,140	esta
530 DRAW 32,350:MOVE 470,280	1070 REM dibujado
540 DRAW 355,280:DRAW 355,215	1080 REM -----
550 DRAW 470,215:DRAW 470,280	-----
560 PLOT 36,342-16#B:PLOT 316,	1090 IF sdato(x,y)=0 THEN RETU
342-8#16	RN
570 PLOT 36,342-8#B:PLOT 316,3	1100 open=sdato(x,y):GOSUB 990
42-8#B	1110 RETURN
580 PLOT 51#8#16,348:PLOT 51#B	1120 REM -----
-16,142	-----
590 RETURN	1130 REM -----
600 REM -----	-----
-----	1140 REM -----
610 REM Dibujo del Sprite	-----
620 REM -----	-----
630 PEN 15	1150 LOCATE 2,20:PRINT "Nuevo"
640 PRINT CHR#(30):CHR#(23):CH	1160 FOR i=0 TO 15
R#(1)	1170 FOR j=0 TO 23
650 x:=0:y:=0:character#" "	1180 sdato(i,j)=0
660 WHILE INKEY#<0:":#WEND	1190 NEXT
670 WHILE character#<>"E"	1200 NEXT
680 open=sdato(x,y):PLOT 58+y#	1210 GOSUB 1250
16,338-y#B,15:GOSUB 990:FOR i=	1220 LOCATE 2,20:PRINT SPC(18)

1250	REM	Borrar ventanas	2000	k=sdato(x,y)	2530	FOR i=0 TO columnas
1260	REM	-----	2010	sdato(x,y):vdat2o(x,y)	2540	POKE i,v:PEEK(x)
1270	ORIGIN	0,0,40,311,346,144	2020	sdato2(x,y):=i	2550	k=i+1:i=i+1
1280	ORIGIN	0,0,476,548,276,21	2030	NEXT	2560	NEXT
9:CLD			2040	NEXT	2570	NEXT
1290	ORIGIN	0,0,0,640,400,0	2050	IF sprite1 THEN sprite2	2580	longitud:=89000
1300	RETURN		2060	LOCATE 9,2	2590	WHILE INKEY="" :WEND:MOD
1310	REM	-----	2070	PRINT USING "#":sprite	2600	2: BORDER:OPEN 1
1320	REM	Imprimir el sprite	2080	GOSUB 1250:GOSUB 1320	2600	PRINT:INPUT "Nombre del S
1330	REM	-----	2090	s=i:y=j	2610	=":nombre\$
1340	FOR	x=0 TO 15	2100	LOCATE 2,20:PRINT SPC(18)	2610	IF nombre\$="" THEN nombre
1350	FOR	y=0 TO 23	2110	RETURN	=":"character"	
1360	IF	sdato(x,y) THEN gpenes	2120	REM	2620	1=9020:1=89002:as="9020 D
1370	NEXT	sdato(x,y):GOSUB 990	2130	REM	ATA "	
1380	NEXT		2140	REM	2630	PRINT:PRINT "9000-REM" :IN
1390	RETURN		2150	CALL 8BC02:CALL 8BB4E:REM	2640	PRINT "9010 REM Filas":IN
1400	REM	-----	2160	MODE 1:BORDER 10:INK 0,4	18*(STR\$(Filas+1),2)"/Columna	
1410	REM	Reflejo vertical	2170	PAPER 3:PEN 1:LOCATE 10,2	:=MID\$(STR\$(columnas+1),2)	
1420	REM	-----	1:PRINT " DISEÑADOR DE SPRITES "	2650	FOR y=0 TO filas	
1430	LOCATE	2,20:PRINT "Reflej	2180	PAPER 0:OPEN 2:PRINT:PRINT	2660	FOR j=0 TO columnas
o vertical"			"Tamaño máximo:16x24 pixels (	2670	as=as+MID\$(STR\$(PEEK(x)),	
1440	i=0:j=y		2k3 caracs.)"	2680	2):" :ink=i+1	
1450	FOR	x=0 TO 7	2190	PRINT:PRINT:PRINT " Use "	2680	IF LEN(as)>75 OR (n:column
1460	FOR	y=0 TO 23	"CHR\$(24):" las teclas del curs	as AND y=filas) THEN PRINT LE		
1470	k=sdato	(x,y)	or " :CHR\$(24):" para mover."	18*(as,LEN(as)-1):)=10:as=MI		
1480	sdato	(15-x,y)	2200	PRINT:PRINT " :CHR\$(24):	\$(STR\$(1),2) + " DATA "	
1490	sdato	(x,y):k	" COPY " :CHR\$(24):" para dibu	2690	NEXT	
1500	NEXT		ar un pixel."	2700	NEXT	
1510	NEXT		2210	PRINT:PRINT " :CHR\$(24):	2710	PRINT:PRINT "Salvar o Fin
1520	GOSUB	1250:GOSUB 1320	" DEL " :CHR\$(24):" para borrar	(48):" :>		
1530	x=i:y=j		" "	2720	as="":WHILE as<>"S" AND a	
1540	LOCATE	2,20:PRINT SPC(18)	" ENTER " :CHR\$(24):" salvar lo	s<>"":as=UPPER\$(INKEY):WEND		
1550	RETURN		s datos del sprite."	2730	IF as="F" THEN PRINT:FIN	
1560	REM	-----	2220	PRINT:PRINT " :CHR\$(24):	1:PEED KEY 25,2:END ELSE PRINT	
1570	REM	Reflejo horizontal	" :CHR\$(24):" para cambiar	CHR\$(13):SPC(20):CHR\$(13):		
1580	REM	-----	pens"	2740	nombre:=LEFT\$(nombre,8)+	
1590	LOCATE	2,20	2240	PRINT " :CHR\$(24):" :SP:CH	" :SPR"	
1600	PRINT	"Reflejo horizontal	R\$(24):" :...intercambiar sprite	2750	PRINT "Salvando " :CHR\$(34	
"			" "	1:nombre:CHR\$(34):" ,b,89000,&		
1610	i=0:j=y		2250	PRINT " :CHR\$(24):" :H:CH	1:HEX\$(longitud):PRINT	
1620	FOR	x=0 TO 15	R\$(24):" :...reflejo horizontal"	2760	SAVE nombre,b,89000,long	
1630	FOR	y=0 TO 11	2260	PRINT " :CHR\$(24):" :V:CH	itud	
1640	k=sdato	(x,23-y)	R\$(24):" :...reflejo vertical"	2770	PRINT:PRINT "Pulse una te	
1650	sdato	(x,23-y):=sdato(x,y)	2270	PRINT " :CHR\$(24):" :G:CH	cla..."	
1660	sdato	(x,y):k	R\$(24):" :...guardar los pixels"	2780	WHILE INKEY="" :WEND	
1670	NEXT		2280	PRINT " :CHR\$(24):" :I:CH	LE INKEY="" :WEND	
1680	NEXT		R\$(24):" :...redefinir ink"	2790	MODE 0:GOSUB 360:GOSUB 12	
1690	GOSUB	1250:GOSUB 1320	2290	PRINT " :CHR\$(24):" :F:CH	50:GOSUB 1320	
1700	x=i:y=j		R\$(24):" :...fin"	2800	as=0:y=0:PEN 15	
1710	LOCATE	2,20:PRINT SPC(18)	2300	PEN 1:LOCATE 14,24:PRINT	2810	RETURN
1720	RETURN		"Pulse espacio..."	2820	RETURN	
1730	REM	-----	2310	MOVE 0,0:DRAW 0,398,1:DRA	2830	REM Poner limites superio
1740	REM	Biso	W 638,398:DRAW 638,0:DRAW 0,0	2840	REM e inferior	
1750	REM	-----	2320	WHILE INKEY="" :WEND	2850	REM
1760	LOCATE	2,20:PRINT "Gr and	2330	WHILE INKEY="" :WEND	2860	J=23:(f-1):i=0
0..."			2340	CLS	2870	WHILE k=0
1770	i=0:j=y		2350	RETURN	2880	FOR i=0 TO 15
1780	FOR	y=0 TO 15	2360	REM	2890	k=k+sdato(i,j)
1790	FOR	x=0 TO 15	2370	REM	2900	NEXT
1800	temp	(15-y,x):=sdato(x,y)	2380	REM	2910	i=i+1:(k=0)
1810	NEXT		2390	k=0:FOR i=0 TO 15:FOR j=0	2920	WEND
1820	NEXT		TO 15:k=k+sdato(i,j):NEXT:INEX	2930	RETURN	
1830	FOR	x=0 TO 15	2400	IF i=0 THEN RETURN	2940	REM
1840	FOR	y=0 TO 23	2410	direccion:=C000+B4,50+15*	2950	REM Poner limites izquier
1850	sdato	(x,y):=temp(x,y)	4	2420	2960	REM y derecho
1860	NEXT		2420	LOCATE 2,20:PRINT "Salvan	2970	REM
1870	NEXT		do..."	2980	i=15:(f-1):k=0	
1880	GOSUB	1250:GOSUB 1320	2430	f=1:GOSUB 2830:y=j	2990	WHILE k=0
1890	x=i:y=j		2440	direccion:=direccion+800*	3000	FOR j=0 TO 23
1900	LOCATE	2,20:PRINT SPC(18)	(y MOD 8)+850:(y\8)	2450	3010	k=k+sdato(i,j)
1910	RETURN		2450	f=1:GOSUB 2830:filas:=j-y	3020	NEXT
1920	REM	-----	2460	f=1:GOSUB 2950:k=j-(i AND	3030	i=i+1:(k=0)
1930	REM	Intercambio	2470	direccion:=direccion+(x\2)	3040	WEND
1940	REM	-----	2480	f=1:GOSUB 2950:columnas=	3050	RETURN
1950	LOCATE	2,20	(i-x)\2	2490	k=89000	
1960	PRINT	"Cambio de sprites"	2500	POKE k,filas+1:POKE k+1,c		
1970	i=0:j=y		0:columnas+1:ink=2	2510	FOR j=0 TO filas	
1980	FOR	x=0 TO 15	2520	k:=direccion+8000:(i MOD 8		
1990	FOR	y=0 TO 23	)+850:(j\8):IF x#FFFF THEN x			
			:=8C050			



# GESTOR DE MENUS

# A análisis

La mayoría de los programas, en cuanto incorporen una mínima sofisticación, necesitan plantear al usuario una serie de opciones distintas, para que la persona pueda escoger la que más se acomode de la manera más sencilla y obvia posible.

El programa de esta semana de AMSTRAD Análisis está enfocado en este sentido; pretende mostrar una forma, de las muchas que hay, de implantar un menú de opciones dentro de un programa.

No se ha tenido especial cuidado en la presentación y estética del gestor de menús, pensando en que cada lector pueda crearla a su propio estilo y pensando, también, en resaltar con la máxima claridad la metodología elegida. Vamos pues, con el programa.

**10** Implementa un bucle WHILE...WEND sin fin aprovechando la «costumbre» del Amstrad de inicializar a 0 cualquier variable numérica que no se haya definido antes. Por eso la variable «fin» no aparece por ninguna parte, y el bucle se ejecuta siempre, porque es falso que «fin» sea cierta (distinta de 0).

**30-80** Imprime en pantalla las 5 opciones de que consta el menú (pueden ser más o menos de 5, por supuesto).

**90** Inicializa las variables «elección» y «num. opción»; esta última corresponde al número de operaciones que gestionará el programa menú.

**100-120** Prepara las variables que usarán para imprimir en pantalla un signo, el cual a su vez señala qué opción del menú estamos considerando escoger.

**130** Impresión del signo en el lugar adecuado.

**140** Bucle de espera para evitar que el «\*» parpadee.

**150-180** Ajusta las variables «elección» y «lugar» en función de que haya sido pulsado o no las teclas de movimiento vertical del cursor, aumentando o disminuyendo en 1 la posición de impresión del «\*».

**190** Se examina el teclado de nuevo si no se ha pulsado «ENTER», ya que esta tecla le indica al programa que deseamos escoger una opción del menú.

**200** El programa bifurca o la subrutina adecuada en función del valor de la variable «elección».

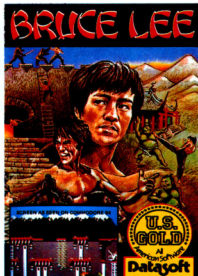
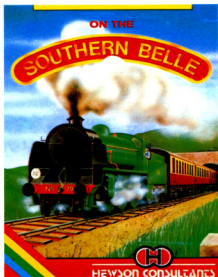
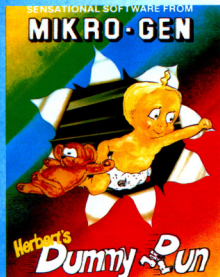
**230-300** Se imprime el número de opción escogida, se espera a que se pulse cualquier tecla (línea 290) y se vuelve al bucle sin fin del programa principal. En una aplicación real, antes del «goto 280» deben ir las líneas de Basic que implementen el trabajo de la opción en cuestión, lógicamente.





# PREPA

# Dis



SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

# PARA Las MALETAS A

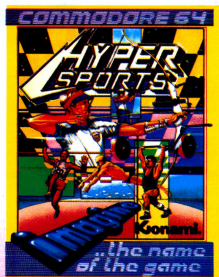
# Disney World!

## FREE

Software

### TE INVITA

COMO LO OYES. EN CADA UNO DE NUESTROS JUEGOS ENCONTRARAS UN CUPON. MANDANOSLO DEBIDAMENTE RELLENADO Y PARTICIPARAS EN EL SORTEO QUE ENTRE TODOS LOS CUPONES RECIBIDOS HAREMOS EL 15 DE ENERO. ¡IMAGINATE! NADA MENOS QUE VIAJE Y ESTANCIA PARA 2 PERSONAS DURANTE 1 SEMANA EN MIAMI, EN DISNEY WORLD.



# MICRO-1

MICROLID: Gregorio Fdez., 6  
Tel. (983) 35 26 27. Valladolid

Jorge Juan, 116. 28028 Madrid  
Tels. (91) 233 07 35-274 53 80

Hemos creado para ti el nuevo Club del Software MICRO-1. En él vas a encontrar los últimos títulos a unos precios increíbles. Para hacerte socio sólo es necesario que nos pidas uno de estos programas, teniendo como regalo de bienvenida un magnífico bolígrafo con reloj de cuarzo incorporado.

¡¡Primicia AMSTRAD: Ajedrez en tres dimensiones con sintetizador de voz y completamente en castellano: 2.395!!

	Ptas.		Ptas.		Ptas.
Match day	1.975	Dragontorc	1.875	Raid o moscow	1.975
Southern belle	1.975	Dummy run	1.975	Jump jet	2.595
Death pit	1.975	Combat lynx	1.925	Beach head	2.395
Decathlon	1.975	Exploding fist	2.095	Basketball	1.950
Gremlins	1.975	Rocky	1.925	Highway encout.	1.975
Popeye	1.875	Basketball	1.925	Hightsade	1.925

Joystick Quick Shot I  
1.795

Joystick Quick Shot II  
2.495

Joystick Quick Shot V  
¡¡El mejor!!  
2.995

Lápiz óptico DK'Tronics  
5.850

¡¡Increibles precios para  
tu AMSTRAD 464 y 6128!!

Cassette especial para  
664 y 6128: 5.295

Tapa de metacrilato  
transparente  
para tu AMSTRAD  
2.450

Impresoras  
20% Dto. sobre P.V.P.

Cinta C-15 85  
Diskette 3 995

## LIBROS

	Ptas.		Ptas.
Curso Autodidáctico Basic (I)	2.725	Curso Autodidáctico Basic (II)	2.750
Programando con AMSTRAD	2.250	Hacia la inteligencia artific.	1.375
Juegos sensoriales AMSTRAD	1.790	Código máquina para princip.	1.975
40 juegos educativos AMSTRAD	1.790	Programando Basic con AMSTRAD	1.975
Guía de referencia de Basic	1.900	Música y sonido con AMSTRAD	1.125

El pedido te lo enviamos URGENTEMENTE contra-reembolso SIN NINGUN GASTO DE ENVIO, LLAMANDO a los teléfonos: (91) 233 07 35-274 53 80 o escribiendo a MICRO-1. Jorge Juan, 116. 28028 Madrid.



PRESENTA...

# AMSTRAD

## NUEVOS PROGRAMAS EN CASSETTE Y DISCO

### ARGO NAVIS



El comandante de nave AMSTRAD-1 se encuentra atrapado en las profundidades de una central nuclear y debe salir con vida. Excelentes gráficos y sonido. P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

### JUMP JET



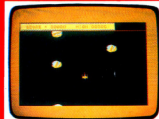
Te encuentras a los mandos de la nave "Air-craft". En una perfecta maniobra debes despegar del portaviones. (Excelente versión simulador vuelo-combat) P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

### ZEDIS II



Editor-desensamblador del Z-80 para el programador más avanzado. P.V.P.: CASSETTE 1.900 pts. DISCO 2.600 pts.

### ROCK RAID



Debes pilotar con agilidad la nave que a lo largo de su viaje galáctico sufrirá encuentros con meteoritos, residuos planetarios, etc. Gran movilidad y excelentes efectos. P.V.P.: CASSETTE 1.900 pts. DISCO 2.600 pts.

### MUSIC MAESTRO



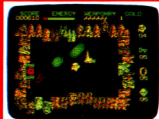
El más completo programa de música creado para el AMSTRAD. Permite crear sonidos, maticados y convertir tu ordenador en la mejor "caja de música". P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

### SYSTEM X



Ampliación del lenguaje Basic. Conjunto de 30 nuevas instrucciones (de ciclo, proteo) para ayudar en la programación. P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

### WIZARD'S LAIR



Te encuentras atrapado en las profundidades de una caverna, llena de obstáculos, adversidades, etc. ¿Serás capaz de salir con vida? P.V.P.: CASSETTE 1.900 pts. DISCO 2.600 pts.

### PAZAZ



Programa que permite de una manera sencilla la creación de pantallas con gráficos, dotados de movimiento, acompañados de música. P.V.P.: DISCO 2.900 pts.

### ODOJOB



La mejor utilidad para el mejor conocimiento del disco. (Copias de disco, Disk map, Disk track, sector, etc.) P.V.P.: DISCO 2.600 pts.

### MACADAM FLIPPER



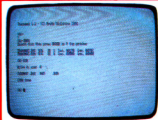
Un programa que nos trasladó al mundo de la máquina-flipper del mejor casino de Las Vegas. Posibilidad de creación del tablero, puntuaciones, etc. P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

### SYCLONE 2



Programa de utilidad que permite realizar copias de seguridad (back-ups) a distintas velocidades (tambours). P.V.P.: CASSETTE 1.800 pts. DISCO 2.500 pts.

### TRANSMAT

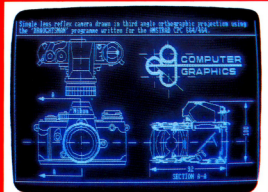


Passar los programas de un disco a otro no es problema. Con Transmat este proceso será fácil y sencillo. P.V.P.: DISCO 2.600 pts.

### OTROS PROGRAMAS EN STOCK

MIINI OFFICE	P.V.P. CASS. 3.200 pts.
	P.V.P. DIS. 3.900 pts.
WORLD CUP FOOTBALL	P.V.P. CASS. 1.800 pts.
BATTLE FOR MIDWAY	P.V.P. CASS. 1.800 pts.
FIGHTER PILOT	P.V.P. CASS. 2.200 pts.
SURVIVOR	P.V.P. CASS. 1.800 pts.
MOON BUGGY	P.V.P. CASS. 1.800 pts.
TECHNICIAN TED	P.V.P. CASS. 1.800 pts.
FRUITY FRANK	P.V.P. CASS. 1.800 pts.
DATABASE	P.V.P. CASS. 2.100 pts.
LOGO TURTLE GRAPHICS	P.V.P. CASS. 2.400 pts.
TASCOPY Y TASPINT	P.V.P. CASS. 2.600 pts.
FONT EDITOR	P.V.P. CASS. 1.900 pts.

### DRAUGHTSMAN



Sensado programa de dibujo que permite trazar la pantalla del AMSTRAD como un sencillo tablero de dibujo; sus resultados son espectaculars. P.V.P.: CASSETTE 4.500 pts. DISCO 5.200 pts.

ENVÍENOS A MICROBYTE **AS.**  
P.º Castellana, 179, 1.º - 28046 Madrid

Nombre				
Apellidos				
Dirección				
Población				
D.P.				
Telefono				
<b>ENVÍENOS GRATIS</b>				
JUEGO	C	D	Precio	TOTAL

**PRECIO TOTAL PESETAS**  
Incluye talón nominativo   
Contra-Reembolso

Pedidos por teléfono **91 - 442 54 33 / 44**

# WEST BANK



Defiende el banco de Soft City del ataque de los forajidos y consigue escribir tu nombre en la leyenda del «FAR WEST».



Fotos tomadas de un Spectrum 48 K

**SPECTRUM · 1950 · AMSTRAD**

(DISK · 3050)

PEDIDOS CONTRA  
REEMBOLSO:  
TEL.: (91) 715 00 67  
TIENDAS Y  
DISTRIBUIDORES:  
TEL.: (91) 447 34 10.

# DINAMIC

«MANSION DINAMIC»  
C/ TILOS, 2, 21,  
MONTEPRINCIPE  
BOADILLA DEL MONTE  
MADRID