

AMSTRAD

Semana

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 51

160 Ptas.

Canarias 165 pts.

**JAUQUE MATE A
LOS ERRORES EN
LOS PROGRAMAS**

**¡SENSACIONAL
REGALO!
CARATULAS PARA
TUS DISCOS**

**PINCELES
HECHOS DE BITS.**

**Los secretos
últimos del Alien 8**

**CON FISION, CHERNOBYL
NO HUBIERA OCURRIDO**

**Inversión
de pantallas
gráficas**



SOMOS MAYORISTAS

MICRO-1

PRECIOS INCLUIDO IVA

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 275 96 16 - 274 75 02

(Metro O'Donnell o Goya)
Aparcamiento gratuito en Felipe II

Ofertas en software: 2 programas al precio de 1 y además regalo un reloj digital completamente gratis. ¡¡Asombroso!! ¿Verdad?

BAT MAN _____	2.300 ptas.	KUNG-FU MASTER _____	2.300 ptas.
ROCK'N LUCHA _____	2.300 ptas.	SABOTEUR _____	2.300 ptas.
YIER AR KUNG FU _____	2.300 ptas.	PING-PONG _____	2.300 ptas.
THE WAY OF THE TIGER _____	2.300 ptas.	MILLION II _____	2.500 ptas.
WEST BANK _____	2.100 ptas.	OLE TORO _____	2.300 ptas.
CAMELOT WARRIORS _____	2.300 ptas.	TURBO ESPRIT _____	2.100 ptas.
RAMBO _____	2.300 ptas.	SABRE WULF _____	1.650 ptas.
WORLD CUP (DISCO) _____	3.300 ptas.	BATALLA DE LOS PLANETAS _____	2.100 ptas.
MILLION II (DISCO) _____	3.300 ptas.	SABOTEUR-COMBAT LINX DISCO _____	3.300 ptas.

RAMBO-MATCH DAY (DISCO) **3.300** ptas.

SOFTWARE DE REGALO (OFERTA 2 x 1)
DECATHLON BEACH HEAD SOUTHERN BELLE
DRAGONTORC

LAPIZ OPTICO
3.295 PTAS.

CASSETTE ESPECIAL ORDENADOR
4.495 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
7.650 PTAS.

AMPLIACION DE MEMORIA ANTA 64 K.3
12.500 PTAS.

IMPRESORAS
20 % DE DESCUENTO SOBRE P.V.P.

TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLE SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69

CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLE SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3"	830

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

RATON DE PANTALLA CON SOFTWARE 8.900 PTAS.

OFERTAS EN JOYSTICKS

QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT V	1.695

PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO. TEL. (91) 275 96 16 - 274 75 02
O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.

Tiendas y distribuidores grandes descuentos.
Dirigirse a Dipromsa. C/ Galatea, 25. Tel. (91) 274 75 03

AMSTRAD

sumario

Año 11 • Número 51 • 2 al 8 de Septiembre
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea.
Ceuta y Melilla, 155 ptas.

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.ª Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

Fernando Chaumel

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sopena

Robert Chatwin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Alberto Suñer

Secretaría Redacción

Carmen Santamaria

Fotografía

Carlos Candel

Chema Sacristán

Portada

M. Barco

Ilustradores

J. Igual, J. Pons, F. L. Frontán,
J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

María Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

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

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400
(Fuencarral) 28049 Madrid
Teléfonos: Suscrip.: 734 65 00
Redacción: 734 70 12

Dto. Circulación

Paulino Blanco

Distribución

Coedis, 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

Déposito Legal:

M-28468-1985

Derechos exclusivos

de la revista

COMPUTING with the AMSTRAD

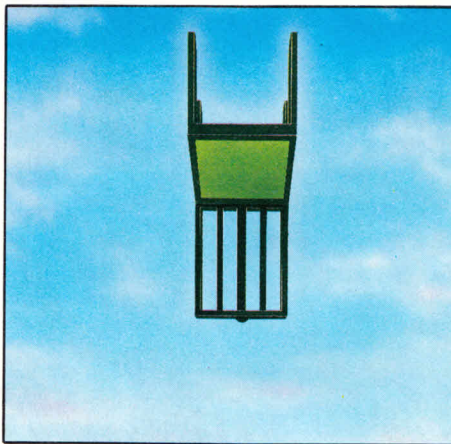
Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.
Americana de Ediciones, S.R.L. Sud
América 1.532. Tel.: 21 24 64. 1209
BUENOS AIRES (Argentina).

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



Primeros pasos

Una de las cosas que todo programador debe prever es controlar todos aquellos errores que, paradójicamente, escapan a su control a la hora de realizar el programa, por ejemplo, que el usuario indique un nombre de fichero que no esté en el disco, o que esté lleno, etc.



Patas arriba

En este número 51, que conmemora nuestro primer aniversario, nuestra revista hermana, MICROMANIA, nos ha hecho un regalo; nada menos que un completo «Patas arriba» de uno de los juegos más famosos para **Amstrad**; el «Alien 8», de Ultimate. Para que los amantes de los juegos sepan que poseen un lugar importantes en **AMSTRAD SEMANAL**.

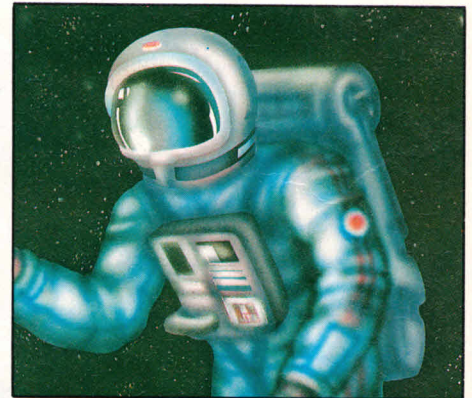
Código Máquina 18

Explicamos, punto por punto, como manejar la pantalla de alta resolución desde máquina, mostrando los equivalentes de los comandos Basic DRAW, PLOT y otros.



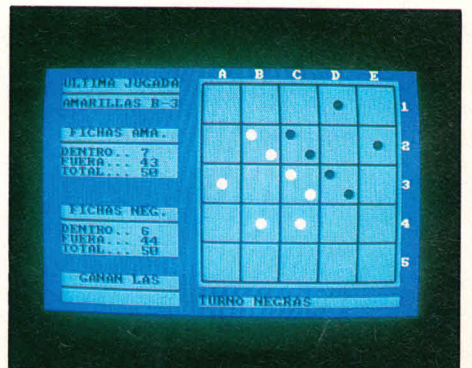
ProgramAcción

En lenguaje máquina se pueden hacer muchas más cosas desde Basic, o son imposibles, o muy costosas de realizar en términos de tiempo y esfuerzo. Una de ellas es la inversión vertical y horizontal de pantallas, a toda marcha.



Serie Oro

FISION es un juego, para dos personas, que imita lo que ocurre durante una reacción nuclear. Indudablemente, una descripción más prolija sólo serviría para anular el placer de la sorpresa que este programa va a producirnos, cuando juguéis con él.



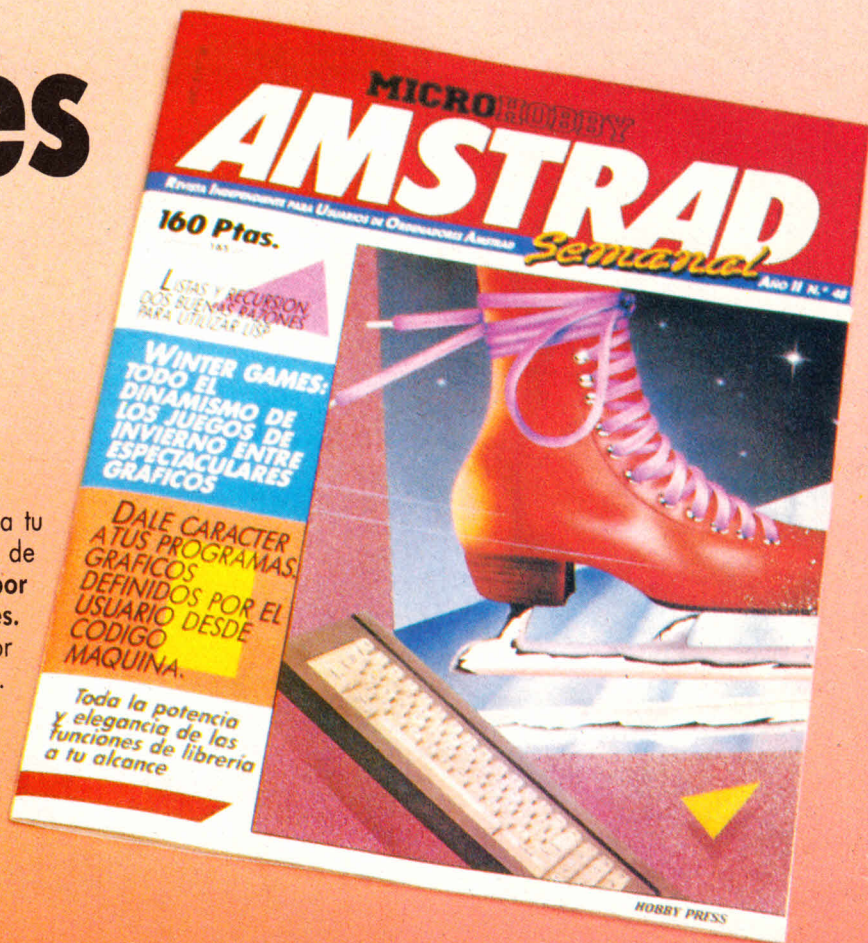
OFERTA ESPECIAL I ANIVERSARIO

6 meses Gratis de AMSTRAD CASSETTE

Suscríbete ahora a Microhobby Amstrad, o realiza tu renovación, y recibirás, totalmente gratis, un regalo de excepción: una suscripción a Amstrad Cassette por seis meses.

Cada cinta contiene los programas publicados por Microhobby Amstrad durante un mes.

Todos los programas de nuestras cintas se encuentran desprotegidos, con el objeto de facilitar su copia en disco y la revisión de los listados.



En cada cinta encontrarás:

- Apasionantes juegos llenos de acción y dinamismo.
- Utilidades con las que sacar mayor partido a tu ordenador.
- Rutinas en código máquina, para que las utilices en tus propios programas.
- Y pequeños trucos de programación, para que, poco a poco, te conviertas en un experto.

Recorta o copia el cupón que aparece cosido en las páginas de esta revista.
APROVECHA ESTA OFERTA ÚNICA,
válida sólo para España
hasta el 31 de
noviembre de
1986.

editorial

Hemos superado los **50 números**. El primer aniversario de **AMSTRAD SEMANAL** ya es una realidad.

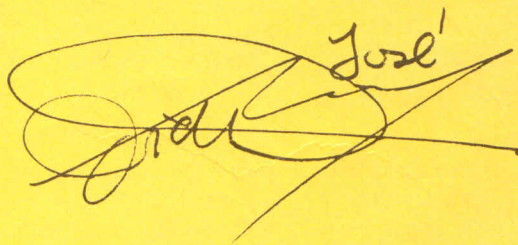
Resulta un tanto extraño, acostumbrados a hablar de informática, bytes y lenguajes, expresar a través del papel lo que esto significa para nosotros. Representa que, a lo largo de **50 semanas**, la primera y única revista semanal que se ocupa de los ordenadores **Amstrad** ha recibido un constante jaleo y apoyo en su labor por parte de miles de personas, de todos aquellos que, asidua u ocasionalmente, nos leen.

Sentimos de forma continua y profunda esa multitudinaria presencia, a la vez que la inmensa responsabilidad que ello deposita sobre nuestros hombros. Una dulce carga que nos impulsa a mantener una cada vez más clara línea ascendente en cuanto a veracidad, interés en la información y calidad de edición.

Las claves de nuestra revista siempre han sido una máxima actualidad y un tratar de mantener un constante y directo contacto con todos nuestros lectores, porque creemos, sabemos, que sólo de esta manera un semanario cobra sentido, manteniendo la agilidad y el interés que caracterizan a **AMSTRAD SEMANAL**.

Sentimos, pues, que se ha alcanzado un meta, que marca el fin de una etapa y el nacimiento de otra, de renovación y «aire fresco», sin perder por ello un ápice de lo que hasta ahora ha constituido las líneas maestras de nuestra revista. Toda la redacción, en pleno, ha contribuido en esta editorial, y a todos nos gustaría acabarla de la misma forma que la hemos empezado.

Hemos superado los 50 números. Gracias a todos.

A handwritten signature in black ink, appearing to be 'José', written over a large, stylized scribble.

DIRECTOR DE AMSTRAD SEMANAL

GESTION DE ERRORES

ASI SE GESTIONAN LOS ERRORES

En alguna ocasión, al escribir un programa se nos ha escapado el dedo y hemos pulsado la tecla que no debíamos: cometemos un error.



Quizá nos hemos dado cuenta al momento y lo corregimos antes de pulsar el **RETURN** que mete la línea en memoria. La cosa no tiene mayores consecuencias. Pero un gran número de veces no advertimos estos errores hasta que se está ejecutando el programa y el **Amstrad**, que no pasa ni una, saca en la pantalla un mensaje y nos advierte del fallo:

Syntax error in...

y a continuación nos presenta la línea equivocada para que la corriamos adecuadamente.

Pero este tipo de errores no son los únicos que podemos cometer, tal como nos los mostrará el Programa 1.

Programa uno

Si le «cargamos» en el ordenador y a continuación lo ejecutamos tecleando «**RUN**» y pulsando «**RETURN**», ¿qué ocurrirá?

El programa nos va a pedir un par de datos numéricos, que se utilizarán como «**dividendo**» y «**divisor**». Después calcula el cociente de la división y nos visualiza la pantalla. Hasta aquí no observamos nada extraño.

Ahora ejecutemos de nuevo el Programa, pero con unos datos un poco especiales. El dividendo será un número cualquiera mientras que como divisor vamos a elegir un cero. ¿Qué conseguimos con ello?

Siempre nos han enseñado en Matemáticas que al dividir un número por cero obtenemos un valor tan grande como nosotros queramos al que llamamos «**infinito**». Pero, ¿qué es «infinito»?

De momento sabemos que es una cifra enorme, pero no podemos determinar cuál es. Por eso la llamamos «infinito» y basta, hemos cumplido.

Pero al **Amstrad** no podemos decirle que una variable contiene un valor indeterminado («infinito», por ejemplo), sino que debemos concretar. Y más bien el que lo hace es el propio ordenador: toma el número más grande que puede manejar dependiendo de sus características internas.

Si quiere conocer cuál es, volvamos a ejecutar el Programa 1 con los datos «especiales», tal como dijimos anteriormente. Al pulsar **RETURN** nos aparece en la pantalla un mensaje:

Division by zero

¿Qué ha ocurrido?

Sencillamente lo que se acaba de visualizar en nuestro monitor: hemos intentado efectuar una división en la que el divisor es cero. El Basic se da cuenta de ello, ve que el resultado es un número que no puede tratar correctamente y nos informa de ello, ya que la situación es anómala o errónea.

Y el programa continúa ejecutándose. Prueba de ello es que se imprime el valor del cociente resultante de la división. Mejor dicho, el programa escribe ese valor máximo que el **Amstrad** toma como techo de cálculo.

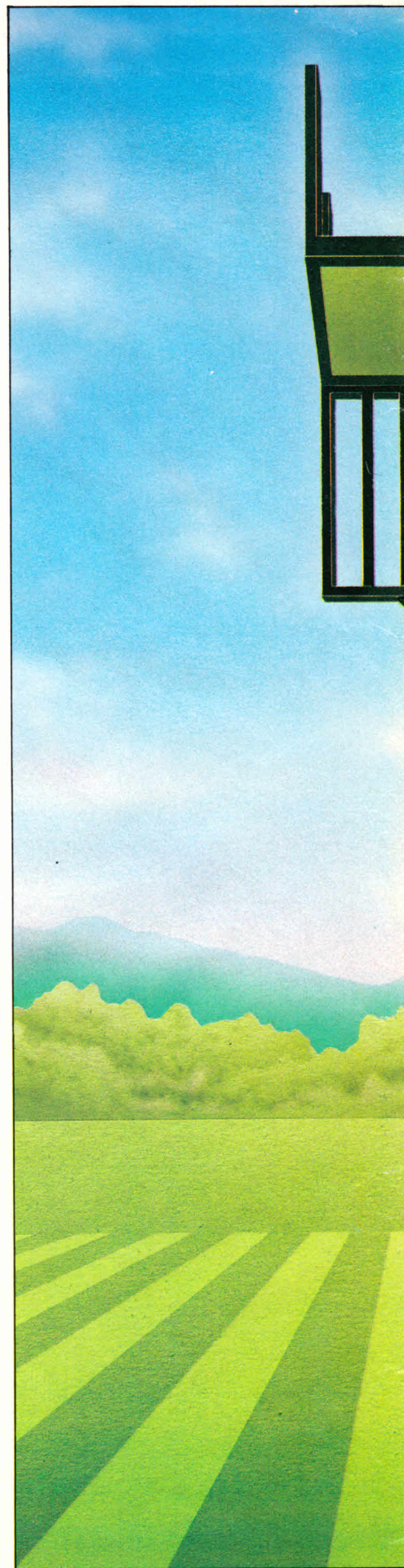
Ya conocemos cuál es:

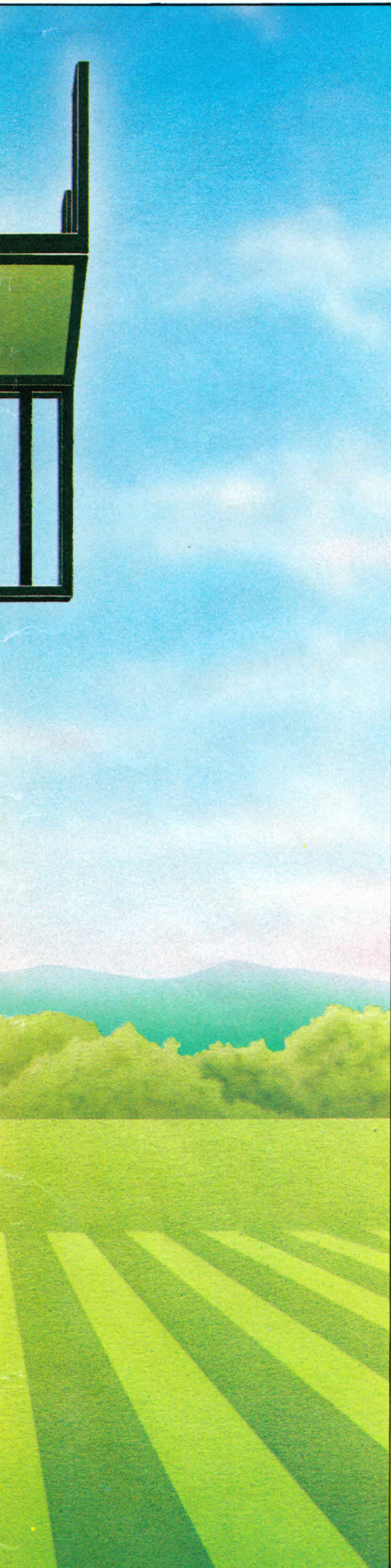
1.70141E+38

es decir, 1 segundo de 38 ceros, que resulta para todos nosotros una cantidad grandísima e inimaginable.

Viendo esto puede surgir una pregunta: ¿quién se ha encargado de interrumpir la ejecución del programa para visualizar el mensaje y después continuar por donde nos habíamos detenido?

El Basic del **Amstrad** posee un sistema que una vez se ha detectado un error, de la clase que sea, genera un mensaje que nos informa del tipo del que se trata. Es el «gestor» de errores.





```
10 REM PROGRAMA 1
20 CLS
30 INPUT "DIVIDENDO: ",dividendo
40 INPUT "DIVISOR: ",divisor
50 cociente=dividendo/divisor
60 PRINT
70 PRINT "EL COCIENTE ES";cociente
```

En el Manual del Usuario de su ordenador existe un capítulo donde aparece una lista completa con todos los errores que puede tratar, el número asociado a cada uno de ellos, así como el mensaje correspondiente y una breve explicación de la causa que lo ha producido. Echele una ojeada, verá que este error es el número 11 y anótelo para posteriores ocasiones.

Este «gestor» es el que también hace aparecer el aviso de «error sintáctico» cuando nos equivocamos al escribir una coma, un paréntesis o cualquier otro símbolo. Y lo mismo ocurre con el resto de los fallos que el **Ams-trad** puede detectar.

Volvamos al Programa 1. Recordemos que a pesar de que se había producido una circunstancia anómala, la ejecución del Programa continuó arrastrando un resultado que no era cierto. ¡Habrá que arreglar este problema! ¿no?

Será necesario que nosotros mismos tratemos el error que se ha producido haciendo que, en este caso, nos vuelva a pedir un nuevo divisor que sea distinto de cero. Y esto es lo que hacemos con el Programa 2. Escríbalo y observe detenidamente las modificaciones que se han introducido.

Programa dos

Sigamos el mismo proceso que con el anterior. La primera vez teclearemos el dividendo y el divisor con valores distintos de cero y el resultado obtenido será el mismo. Sin embargo, cuando asignamos al divisor el número cero, se producen unas consecuencias notablemente diferentes.

Con el Programa 1 se desviaba la ejecución del mismo en la línea 50 y aparecía en la pantalla el mensaje standard, propio del sistema, correspondiente a:

Division by zero

Ahora, con la solución que hemos propuesto, ya no vemos el mensaje anterior sino uno que podemos entender perfectamente en castellano. Y además no se corta la ejecución del programa, continúa permitiéndonos introducir un valor correcto a la variable «divisor» y proseguir así, indefinidamente. ¿Por qué esta diferencia de comportamiento?

Primeros PASOS

¿Se acuerda de todo lo que hemos dicho sobre el «gestor» de errores? Bueno, pues anúlelo para el Programa 11. El Basic del **Ams-trad** permite al programador gestionar a su gusto, o bien en función de sus necesidades, los errores que pudieran aparecer en la ejecución de un programa.

Para que esto sea así, y no como en los casos habituales, debemos decirle al ordenador que vamos a ser nosotros mismos quienes tratemos correctamente los fallos detectados y que por tanto, cuando encontramos uno, el «gestor» del Basic se inhiba y no actúe. Pero, ¿cómo conseguirlo?

Observe atentamente la línea 20 de este último programa:

```
20 ON ERROR GOTO 110
```

Con esta instrucción le estamos indicando al ordenador que en el caso que aparezca un error en la ejecución, bien sintáctico o de cualquier otro tipo, en lugar de ceder el control a la parte de la memoria ROM (la que dirige el micro), lo que debe hacer es dar un salto y continuar por la línea incada, que en nuestro caso es la 110.

A partir de allí tomaremos nosotros las acciones oportunas para tratar el error de acuerdo con nuestras necesidades dependiendo del tipo que sea el producido.

Una vez ejecutados las líneas 110 a 140, el programa se encamina de nuevo a la 50, debido al GOTO de la 150, y aparecerá otra vez en la pantalla un mensaje pidiéndonos un valor correcto para «divisor».

Si tecleamos ahora cualquier valor distinto de cero el ordenador actuará de la manera habitual, calculando el resultado de la operación e imprimirá el valor encontrado.

Así pues, parece que ya hemos descubierto la manera de conseguir nuestro objetivo. Se ha producido una división por cero, al dar a la variable «divisor» dicho valor, y el programa lo ha tratado convenientemente tal y como nosotros lo planeamos. ¿Con esto basta?

Nada más lejos de la realidad. Si no, volvamos a comenzar a trabajar con el Programa 2. Ejecútelo de nuevo y cuando el ordenador le pida los valores de «dividendo» y «divisor» dele dos números distintos de cero para que no se produzca un error detectable.

Después vuelva a ejecutarlo asignando, o tecleando, ahora un cero al divisor. Inmediatamente, cuando el programa llegue a la línea 60, se detecta el error y salta a la 110, donde comienza la rutina de tratamiento del

mismo. Y nos aparece en la pantalla el mensaje en castellano que nos informa de la situación que se ha producido volviendo a pedirnos que tecleemos un valor distinto de cero.

Pues obedezcamos. Se visualizará el resultado de la operación correctamente y volvemos a iniciar el ciclo debido a que el GOTO de la **línea 100** nos envía otra vez, con un salto atrás, a la 30.

Compliquemos las cosas. Dé ahora al dividiendo cualquier valor y al divisor de nuevo un cero. Y el **Amstrad** nos sorprende una vez más. En lugar de repetir el tratamiento de error tal como habíamos pensado, resulta que nos aparece en la pantalla el ya conocido mensaje asociado al error 11. ¿Recuerda?

Division by zero in 60

¿Por qué ha sucedido esto, si tenemos nuestro particular gestor de errores? La primera vez que dimos a «divisor» el valor 0 no ocurrió así, puesto que nos apareció el «aviso» en castellano y, sin embargo, ahora la respuesta que nos da el ordenador no es la misma. ¿Qué ha cambiado de una a otra?

La explicación es bastante sencilla. Dentro de las instrucciones que forman parte de la rutina de tratamiento de error también puede producirse alguno y en este caso el que lo trata no es el gestor que nosotros hemos creado a medida sino el general que va incorporado al Basic.

Parece que con la línea 150 se termina la gestión del error, ya que volvemos al programa principal. Pues bien, esto no es así. Dentro de esta rutina, sobre todo si es muy compleja, pueden existir sentencias GOTO que nos manden de un lado a otro dentro de la misma. Luego, cuando el ordenador se encuentra con una, no entiende que con ella se termine el tratamiento de error sino que interpreta que tiene que dar un salto dentro de las instrucciones que los gestionan.

Vemos con esto que se nos ha creado la necesidad de una sentencia que consiga que la ejecución del programa abandone la rutina de tratamiento de error y vuelva de verdad al programa principal. ¿Cómo superar este estado?

El Basic del **Amstrad** tiene una orden, semejante al GOTO, pero que a la vez le indica lo que nosotros andábamos buscando. Se trata de **RESUME**. Para ver su funcionamiento modifique el Programa 2 cambiando la siguiente línea:

150 RESUME 50

¿Ya lo he hecho? A continuación repita la misma secuencia de acciones que ya habíamos seguido con el programa antes de la modificación. Algo ha cambiado ya que ahora siempre somos nosotros, o nuestro gestor particular, los que tratamos el error convenientemente.

Cada vez que demos a «divisor» el valor cero, la ejecución saltará a la línea 110, que es,

como sabemos, donde comienza la rutina que acutará en caso de fallo. A partir de ahí se irán realizando instrucciones hasta que llegemos a la línea 150 que nos mandará de nuevo al programa principal, para que nos pida un valor correcto para «divisor».

Pero en esta ocasión si queda constancia de que se ha abandonado el gestor de errores como lo indica RESUME, de modo que si el divisor se vuelve a hacer cero, la ejecución del programa saltará otra vez a la rutina que trata esta situación no permitiendo que entre el propio del ordenador.

La instrucción:

RESUME «línea»

hace que el programa salte a la línea cuyo número está indicado por «línea» (tal como lo hace también GOTO), pero a la vez indica al Basic que estamos saliendo de una zona encargada del tratamiento particular y a medida de nuestras necesidades de un error que haya detectado el ordenador y se retorna al programa principal.

Pero no nos conformamos con esto. Vamos a analizar un poco más a fondo las posibilidades que nos ofrece la instrucción RESUME. Para ello modifique de nuevo la línea 150 del Programa 2 eliminando el número de línea a la que hay que volver. Quedaría de la siguiente forma:

150 RESUME

Probemos el programa que resulta después de este cambio. Si se da a la variable «divisor» valores distintos de cero, al no pasar por la rutina de gestión de errores, no se detecta nada anormal. Tecleemos un cero para esta variable a ver qué pasa.

La ejecución salta como en los casos anteriores, a la parte del programa que trata el error cuando éste es detectado. La línea 150 vuelve control al programa principal pero como en esta ocasión no le indicamos la línea a la que debe volver, lo hace a la misma en la que se detectó el error, es decir, a la 60.

```
10 REM PROGRAMA 11
20 ON ERROR GOTO 110
30 CLS
40 INPUT"DIVIDENDO: ";dividendo
50 INPUT"DIVISOR: ";divisor
60 cociente=dividendo/divisor
70 PRINT
80 PRINT "EL COCIENTE ES";cociente
90 FOR retardo=0 TO 3000:NEXT retardo
100 GOTO 30
110 REM TRATAMIENTO DE ERROR
120 PRINT
130 PRINT"EL DIVISOR ES CERO, Y ESO
NO ES VALIDO. TECLEALO DE NUEVO"
140 PRINT
150 GOTO 50
```



Primeros PASOS

Pero las condiciones que lo produjeron no han cambiado. La variable «divisor» sigue conteniendo el valor cero y por tanto el fallo persiste y volveremos a saltar de nuevo al gestor particular. Así nos estará sacando el mismo mensaje indefinidamente hasta que pulsemos dos veces la tecla ESC, claro.

Así pues, «**RESUME**» sin especificar número de línea devuelve el control de ejecución a la misma línea donde se produjo o detectó el error. ¿Comprendido?

Volvamos a hacer un cambio en la línea 150. En esta ocasión se convertirá en:

150 RESUME NEXT

Repitamos la prueba del programa con los valores que produzcan el error, o sea, asignando al divisor un cero. Si observamos detenidamente lo que aparece en la pantalla podremos deducir con un poco de paciencia lo que está sucediendo.

Después de introducir los valores de los operandos de la división mediante las instrucciones **INPUT**, el **Amstrad** detecta el error producido en la línea 60 y salta a la 110 para tratarlo oportunamente, con lo que se visualizará el mensaje adecuado incluido en la rutina de gestión de error.

A continuación llegamos a la línea donde se ha producido el cambio y en esta ocasión no ocurre como en el primer caso, con **RESUME 40**, en el que se nos volvía a pedir el valor del divisor, no como en el segundo en el que con sólo **RESUME**, sin número de línea, nos encontrábamos de nuevo con la del error.

Ahora la ejecución regresará al programa principal pero a la línea siguiente a la que causó el error. Por lo tanto el **Amstrad** hará la línea 70 sin ejecutar la del error y, como a la variable «cociente» no la hemos asignado ningún valor, aparecerá en la pantalla un cero.

Si continuamos dando números, cuando sean correctos, el resultado que se imprime también lo será. Sin embargo, si «divisor» contiene un cero, el valor del «cociente» que se visualice no es el bueno ya que la operación no llega a realizarse y por tanto la variable contiene la cantidad que tuviera almacenada de la anterior pasada con datos correctos.

Es evidente que en nuestro caso la versión del programa que utilizaríamos sería la que emplea **RESUME** con número de línea de retorno, o bien sería necesario reformar la rutina de tratamiento de error para poder emplear **RESUME** o **RESUME NEXT**. Si hemos visto otras posibles soluciones ha sido únicamente con carácter didáctico para poder ver en funcionamiento de un modo sencillo cada una de estas variaciones de la orden **RESUME**.

Es posible que todavía ande de vacaciones, o quizá acabe de regresar de ellas. ¡Bienvenido!, si así es. Quizá haya hecho un «**mon-tón**» de kilómetros en su propio coche y sus visitas a las gasolineras habrán sido bastantes frecuentes ¿no?

Bueno, pues le proponemos ahora que analice un programa con el que le será posible

comprobar si su coche está fuera de punto al detectar que su consumo ha sido muy elevado y se han ido más «perras» de la cuenta o si por el contrario si su automóvil consume menos que un «mechero» y ha sido una inversión muy rentable. Estamos, ya en serio, ante el Programa 3.

Programa tres

En él se nos van a pedir tres datos: la cantidad de dinero que nos hemos gastado en la gasolinera, el precio del litro de gasolina y el número de kilómetros que hemos recorrido en nuestro viaje turístico.

```
10 REM PROGRAMA III
20 ON ERROR GOTO 160
30 CLS
40 INPUT "IMPORTE TOTAL? ",importe
50 INPUT "PRECIO LITRO DE GASOLINA? ",precio
60 litros=importe/precio
70 PRINT
80 PRINT"EL NUMERO DE LITROS ES";litros
90 PRINT
100 ON ERROR GOTO 0
110 INPUT "KILOMETROS RECORRIDOS? ",kilometros
120 consumo=litros*100/kilometros
130 PRINT
140 PRINT"EL CONSUMO MEDIO HA SIDO DE";consumo;"LITROS"
150 END
160 REM RUTINA DE TRATAMIENTO DE ERROR
170 PRINT
180 PRINT"OJALA EL PRECIO DE LA GASOLINA FUERA ***** CERO PT S. *****"
190 PRINT
200 RESUME 50
```

La primera parte del programa se va a encargar de calcular la cantidad de litros que hemos echado en el depósito de nuestro coche y que depende, claro, del precio del litro de gasolina. La segunda tiene la misión de con-

firmarnos, o más bien de informarnos, si nuestras sospechas de que consumíamos un buen «chorro» de gasolina cada vez que pisábamos el acelerador, eran acertadas.

Veamos su funcionamiento sobre la marcha. Respondamos a las tres preguntas con los siguientes datos válidos: 2000, 90 y 240 respectivamente. El programa correrá sin detenerse y, al no detectarse ningún error, llegará a su final imprimiendo los resultados obtenidos partiendo de los valores que hemos introducido.

Pero este no es el caso que nos interesa, didácticamente hablando. Estamos viendo diversas formas de tratar de un error así que, ¡sigamos con ellas! Ejecute de nuevo el programa pero ahora dele al ordenador un dato que no sea del todo correcto, por ejemplo que la gasolina sea «gratis» (valga «cero» pesetas por litro) y sin embargo hayamos tenido que pagar al gasolinero una determinada cantidad.

En la línea 60 se va a producir un error ya que intentamos hacer una división por cero (recuerde que hemos quedado que la variable «precio» iba a contener un cero). Debido a la instrucción contenida en la línea 20, al encontrar el error la ejecución del programa saltará a la 160, lugar donde comienza el tratamiento del mismo.

20 ON ERROR GOTO 160

nos está indicando que cuando se produzca un fallo en alguna línea del programa, por la causa que sea, la gestión del error no lo haga el Basic, sino la rutina que comienza en la línea cuyo número figura al final de esta instrucción.

En la pantalla no aparecerá, entonces, un mensaje en castellano indicándonos que hay algo que no es correcto: el precio del litro de gasolina. Además el programa nos vuelve a pedir otro valor para el precio del litro ya que vamos otra vez al principal mediante el:

RESUME 50

de la línea 200 para continuar solicitándonos los kilómetros recorridos y, si es válida la cantidad tecleada, terminar con la exposición de los resultados obtenidos.

Así pues el programa está preparado para corregir el dato si nos hemos equivocado al teclear un 0 en lugar del que corresponda al precio de litro de gasolina. Pero pensemos otra cosa. ¿Qué ocurrirá si también por descuido le decimos al ordenador que no hemos recorrido ningún kilómetro? ¿Cuál sería el efecto conseguido al introducir un cero cuando el Amstrad nos pregunte por ellos en la línea 110? Pongámonos en esta situación.

El programa se ejecuta normalmente hasta la línea 120. Pero al llegar aquí nos encontramos con que se produce un error: estamos intentando hacer una división por cero, pero esta vez en dicha línea al querer calcular el consumo suponiendo que hemos recorrido «cero kilómetros».

¿Qué ocurrirá ahora? Quedamos que la rutina de tratamiento de errores comienza a partir de la línea 160, que está hecha a nuestra medida y a la que pensamos que llegará la ejecución cada vez que encuentre un fallo.

Pero esta vez no es así. El programa se detiene y aparece el mensaje standard correspondiente al error tipo 11.

Division by zero

¿Cuál es su defecto? Viendo los resultados obtenidos no tendrá muchos problemas en deducirlo usted mismo. Esta instrucción inhibe el posible tratamiento que hubiéramos preparado para las ocasiones en las que se produzca un error y activa de nuevo el mecanismo que el Amstrad ya tenía preparado para cumplir con tales fines.

En el Programa 3, si se produce un error antes de la línea 100, será nuestro propio gestor el que se ocupe de solucionarlo, pero una vez que la ejecución haya pasado por esta línea es el propio ordenador el que se encarga de ello dependiendo del tipo de error detectado.

Ahora bien, generalmente en un programa de cierta complejidad no siempre se sabe, a priori, el tipo de fallo que se va a producir ni la línea de programa donde ocurrirá.

No se preocupe, el Amstrad lo tiene todo controlado. Posee unas palabras clave que nos darán información en todo momento de estas cuestiones. Pero eso lo dejamos ya para una próxima ocasión. De momento nos quedamos saboreando, o recordando, las vacaciones en compañía del Programa 4 que nos servirá de introducción de estos nuevos conceptos. Intente ver su funcionamiento.

Programa cuatro

```

10 REM PROGRAMA IV
20 ON ERROR GOTO 120
30 CLS
40 INPUT"DIVIDENDO: ";dividendo
50 INPUT"DIVISOR: ";divisor
60 cociente=dividendo/divisor
70 PRINT
80 PRINT "EL COCIENTE ES";cociente
90 PRINT
100 GOTO 1000
110 END
120 REM RUTINA DE TRATAMIENTO DE ER
ROR
130 PRINT
140 PRINT"ERROR NUMERO:";ERR;"EN LA
LINEA";ERL
150 RESUME NEXT

```

Ultima
* Hora

EN SONIMAG NUEVOS PRODUCTOS AMSTRAD

Mil quinientos metros cuadrados ocupará el Stand de Sonimag en el que Amstrad España (Grupo Indescomp) presentará sus últimas novedades de cara a la campaña de ventas del último trimestre de 1986.

Dentro de la línea informática en la que Amstrad es líder indiscutible de ventas, dos nuevos equipos, cuya aparición se aguarda con auténtica expectación, vienen a completar la gama actualmente ofrecida por la firma británica. De un lado, el nuevo Sinclair Spectrum + 2, diseñado básicamente como ordenador de video-juegos e introducción a la informática personal. De otro, el Amstrad PC compatible en diversas configuraciones, que se perfila, por precio y prestaciones, como de auténtico impacto.

Además, Amstrad España presentará por vez primera en nuestro país los equipos de alta fidelidad y vídeo de esta marca. Se trata de modelos con y sin disco compacto, que siguen la filosofía de Amstrad en cuanto a la calidad y precio se refieren. Ejemplo de ello es el modelo más económico, dotado de plato giradiscos, sintonizador AM/FM, ecualizador, dos cassettes y altovoces, todo por un precio inferior a las 30.000 pesetas.

El Salón Internacional de la Imagen, el Sonido y la Electrónica, SONIMAG, se celebrará en Barcelona del 15 al 21 de septiembre.

ORDEMANIA SOFT

por fin..

GESPACK

Paquete integrado de gestión que le permite a Vd., de forma fácil y optimizando el tiempo, la gestión global de su empresa.

Este paquete incorpora los programas de CONTABILIDAD, FACTURACION y CONTROL DE STOCKS además de un programa de CONTROL DE PEDIDOS y todo de forma interactiva. Controle su empresa por sólo

29.900 pts.

Contabilidad

Contabilidad de fácil manejo y de gran potencia que permite trabajar con cuentas de hasta cuatro niveles, con capacidad según diskette de 500/1.000 cuentas y de 2.000/10.000 asientos.

Permite modificar o dar de baja apuntes ya integrados en el Mayor, programaciones de cierres, ficheros de Contabilidad y Cuenta de Explotación, ejecución de balances comparativos, reenlazándolos por meses, clave acceso restringido, etc...

19.900 ptas.

Facturación

Programa de gran agilidad y rapidez que incorpora el Control de Clientes, con gran capacidad de datos, Artículos, Albaranes, Facturas y recibos.

Generación automática y manual de documentos, valoración, a voluntad de los albaranes, todo tipo de listado, incluido el del IVA de las facturas emitidas para la declaración de Hacienda, etc...

15.500 ptas.

Control de stocks

Gran capacidad de datos, le permitirán a Vd. llevar con claridad y sencillez el control de su stock.

El programa le permite llevar un libro de entradas/salidas, reorganizarlo, hacer listados de stocks..., le avisará de los límites de stocks, mínimo y máximo por artículo, etc...

Todo para la llevanza de su almacén.

14.900 ptas.

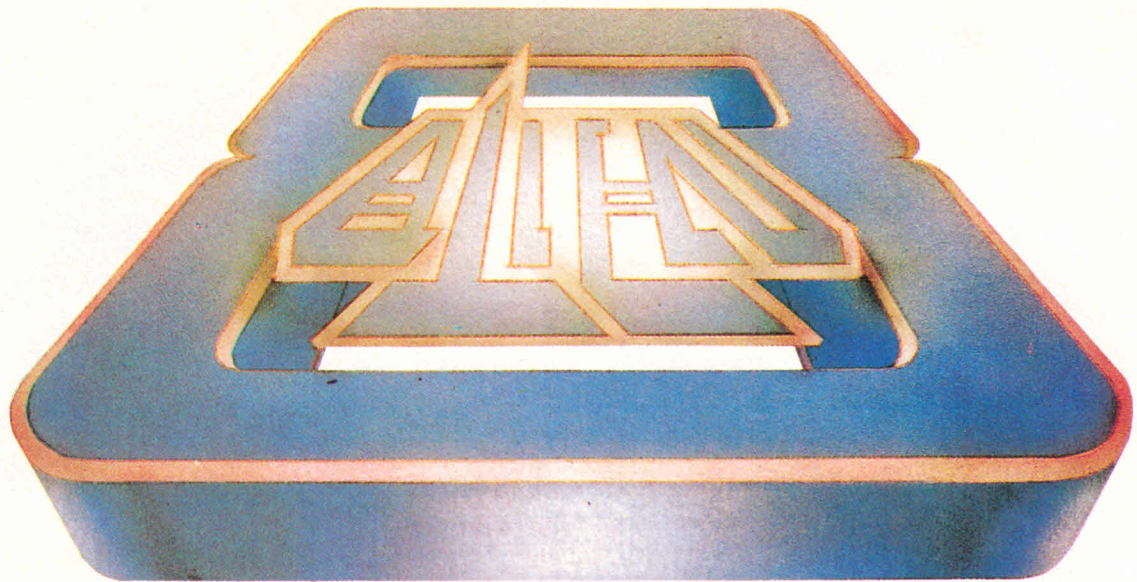
disponibles para: PCW 8256
PCW 8512

Asimismo, Contabilidad disponible para CPC 664/6128 **9.900 ptas.**

OFITES INFORMÁTICA
DISTRIBUIDOR EXCLUSIVO DE ORDEMANIA SOFT

● IVA NO INCLUIDO
Condiciones especiales para distribuidores
Si tiene alguna dificultad en obtenerlos dirijase a

Ofites
Informática
Avda. Isabel II, 16 - 8º
Tels. 4555344 - 4555333
Telex. 36698
20011 SAN SEBASTIAN



Queremos festejar este primer cumpleaños de MICROHOBBY AMSTRAD con un regalo un tanto peculiar para todos sus lectores. Y hemos creído que el mejor de los obsequios sería un especial «Patas Arriba», sobre uno de los juegos legendarios en la historia del ordenador personal: ALIEN-8.

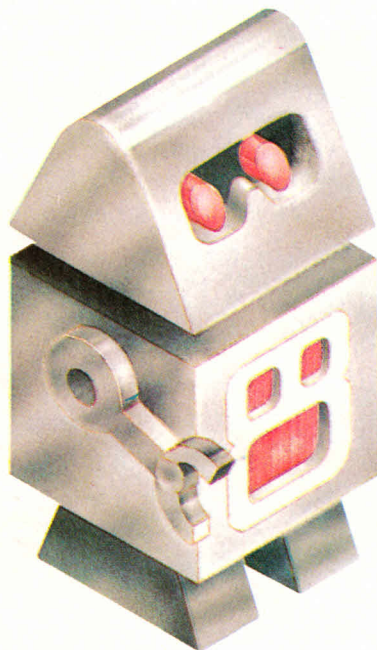
Aquí está por tanto nuestro regalo con pokes y demás elementos de ese género que tan popular se ha hecho en MICROMANIA. ¡Que ustedes lo pasen bien!

*Gabriel Nieto
Director de Micromanía*



En una galaxia distante, en un planeta a punto de extinguirse, el último de sus guardianes prepara la nave para el viaje final. Todos los libros, grabaciones y conocimientos diversos han sido almacenados a bordo de la cápsula, junto con los mejores individuos de su raza, conservados criogénicamente. El final del planeta se perfila próximo y las más vitales piezas del equipo han sido ya embarcadas y activadas. ALIEN 8, un cyboide que representa el último avance de la cibernética en inteligencia artificial, sube a bordo para dirigir la operación de conducir a estos seres hasta un lugar seguro.

Las escotillas de acceso se cierran y se produce el despegue hacia lo



desconocido. La inmensa negrura del espacio devora a nuestros héroes.

Transcurren eones y la prodigiosa máquina, hábilmente programada, funciona a la perfección. Acude con prontitud y silenciosamente alrededor de las selladas válvulas termóleas, cuidando de su perfecto funcionamiento.

La avanzada técnica en el desarrollo de la robótica existente en el ahora desaparecido planeta, ha permitido encomendar a ALIEN-8 la delicada tarea de mantener con vida a los astronautas criogenizados, activando continuamente los sistemas que les sirven de soporte.

El final del viaje

El largo viaje está a punto de finalizar y la computadora central alerta sobre la proximidad del sistema solar prefijado.

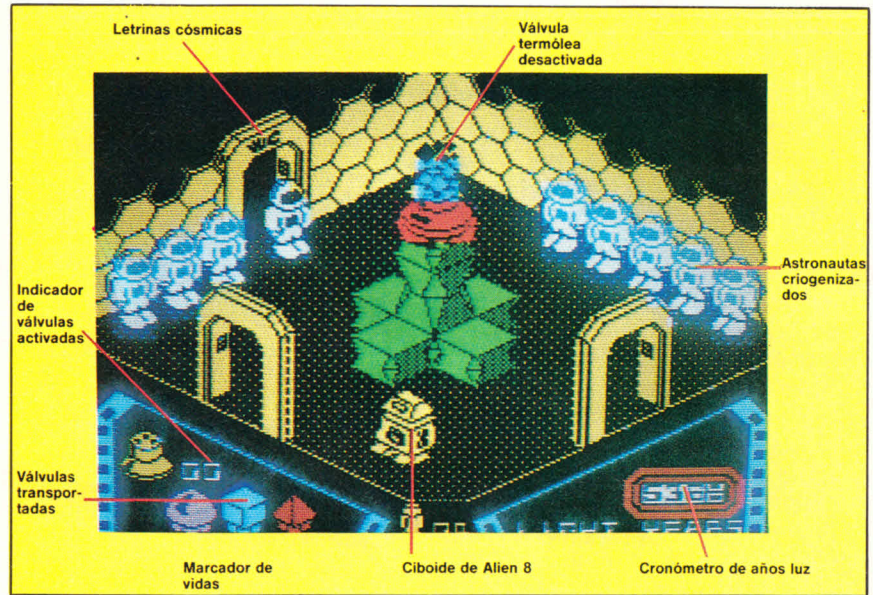
Polvo de eones envuelve el viejo banco de datos, núcleo de la computadora central, cansado tras centurias de cálculos repetitivos y por fin, activado para la secuencia de aterrizaje.

La habilidad de los programadores se pone a prueba en esta secuencia final, algunos cientos de años luz más tarde de su partida del extinguido planeta.

Casi inmediatamente, la nave disminuye su velocidad por debajo de la luz y todos los sistemas principales permanecen activados intentando contener la enorme masa.

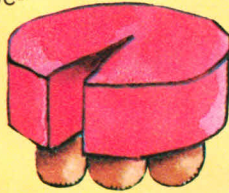
La computadora principal informa sobre la penetración de alienígenas en casi todas las áreas y soportes vitales, provocando su desactivación.

Todos los sistemas criogénicos deben ser reactivados antes de que los mecanismos de empuje de la nave estén ésta dentro de la órbita, a 0 años luz del planeta.

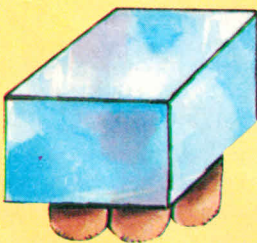


LAS VALVULAS TERMÓLEAS

Las válvulas termóleas son los únicos objetos que pueden ser recogidos y depositados tantas veces como queramos, con la única limitación de no transportar más de tres a la vez ni depositar más de dos en una misma habitación.



Hay cuatro tipos de válvulas termóleas: cuadrada, triangular, ovoide y cilíndrica. Cada una de ellas debe ser encajada sobre su apropiado zócalo, el cual se reconoce por la forma del resplandor que despide.



Además de activar los reactores, las válvulas termóleas tienen multitud de otras aplicaciones marginales.

Por ejemplo, podemos subirnos en ellas, para alcanzar determinados objetivos. En el momento de saltar, si pulsamos simultáneamente la tecla de «coger» objetos, la recuperaremos para futuros saltos en que pueda sernos de nuevo necesaria.

Otra utilidad se nos pone de

manifiesto en el momento de atravesar las habitaciones con ratones o minas. Si empujamos la válvula a través del suelo, ésta, que es invulnerable, nos irá abriendo camino.

Hay alguna habitación en la que los robodroides deben destruir las minas suficientes para pasar, pero al estar en bloques de dos pisos, sólo destruyen los del piso inferior, imposibilitando el paso de nuestro cyboide.

En estos casos, la solución es depositar una válvula sobre el robodroide y destruir simultáneamente los dos pisos de minas.

Pueden protegernos de las lomas que se desintegran (no lo harán si previamente depositamos sobre ellas una válvula) y de los objetos móviles (poniendo la válvula en medio de su trayectoria).

Por último, en caso de que seamos atrapados por el flujo de Energía Letal, la única solución es soltar un objeto encima de él, con lo que el movimiento se verá sensiblemente ralentizado posibilitando el que podamos dirigir al personaje.



Objetivo de la misión

En el interior de la nave hay 128 habitaciones. Veinticuatro de ellas son cámaras criogénicas en donde se conservan, a bajísima temperatura, 132 criogenautas.

Los sistemas que permiten mantenerlos con vida hasta el momento del aterrizaje de la nave en el planeta elegido, deben ser reactivados, acoplando a cada uno de los zócalos termóleas que están ubicados en las cámaras criogénicas, su correspondiente válvula. Estas están distribuidas por determinadas habitaciones y deben ser recogidas y transportadas hasta su cámara adecuada.

Táctica de juego

Para todos los adictos al KNIGHT LORE, el manejo del cyboide ALIEN-8 no debe presentar ninguna dificultad. Tanto el sistema de programación (FILMATION), como el manejo de ambos juegos es exactamente el mismo.

La diferencia sustancial entre ambos es que mientras en el KNIGHT LORE todos los objetos recogidos deben ser depositados en una misma habitación, en éste, cada objeto tiene su lugar adecuado, lo cual complica extraordinariamente el desarrollo de los acontecimientos.

El tiempo, que en este caso indica el número de años luz que faltan para llegar hasta el planeta elegido, transcurre de forma inexorable.

Aún dominando a la perfección todas las técnicas para optimizar los recorridos y conociendo los lugares estratégicos a los que hay que subir y adonde llevar los objetos, resulta casi imposible culminar la misión.

Puesto que el número de objetos máximo que puede transportarse de forma simultánea es de tres, mientras que las posibles necesidades de cada cámara son cuatro, es recomendable dejar algunos objetos (nunca más de dos en el mismo sitio) en zonas fácilmente accesibles para poder recuperarlos en el momento adecuado. Lo mejor es llevar al mismo tiempo tres objetos distintos, teniendo perfectamente localizado el sitio donde está el cuarto. Si en un momento dado nos encontramos con alguno repetido, lo cogemos para dejarlo cerca, en alguna habitación vacía. Es importante apuntar el sitio donde está cada objeto, ya que el elevado número de habitaciones (128) hace que rápidamente perdamos la noción del espacio. También es conveniente detener el juego momentáneamente (con la tecla SPACE), cada vez que accedemos a una nueva pantalla, para estudiar concienzudamente cómo abordar las dificultades que allí se nos presentan. Si no lo hacemos así perderemos un tiempo precioso recapacitando sobre cómo emprender determinada acción e incluso, repitiendo movimientos al percatarnos de que la táctica

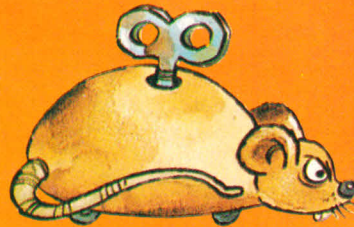
ENEMIGOS IMPLACABLES

Desafortunadamente, la nave espacial ha resultado invadida en algunas de sus más vitales dependencias, lo que obliga a mantener continuamente activadas las válvulas termóreas de las cámaras criogénicas.

Multitud de peligrosos seres nos esperan en cada una de las 128 zonas en que se divide la aeronave.

Vigilante mutante

Se mueve continuamente y con trayectoria rectilínea, por lo que sus desplazamientos resultan predecibles. Fácil de esquivar, aunque a veces su ayuda resulte imprescindible para salir de determinadas habitaciones.



Energía letal

Aunque en principio este tipo de fluido eléctrico no resulta mortal, empuja al protagonista hasta otros obstáculos que son sumamente peligrosos, provocando de este modo un fatal desenlace.

Ratón mecánico

Accionado por mecánicos resortes, sus erráticos desplazamientos resultan de todo punto imprevisibles. A veces, a su pesar, se convierte en aliado destruyendo las minas explosivas (a las que es inmune) en sus alocadas carreras. No obstante, es conveniente extremar los cuidados ante tan peligroso enemigo.

LOS ROBODROIDES TELEDIRIGIDOS

En la ardua tarea que supone sobrevivir, no podemos desdeñar ninguna de las ayudas que, de modo fortuito o intencionado, se nos presentan en el transcurso de la misión.

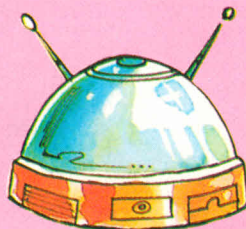
Hay colaboradores inesperados, como es el caso de los ratones mecánicos abriéndonos camino a través de las minas explosivas o los vigilantes mutantes que, en algunas ocasiones, transportan sobre sus cabezas pesados bloques sobre los que habremos de subirnos para conseguir escapar.

Los robodroides teledirigidos son, sin embargo, un caso claro de ayudantes dispuestos a colaborar sumamente.

Su cometido principal es despejar las zonas de minas explosivas y pueden activarse con sólo subir a la plataforma controladora que hay en determinadas habitaciones. Cuatro grandes flechas (ocultas en algún ca-

so) indican el sentido del desplazamiento. La velocidad del movimiento irá marcada por la proximidad al centro de la plataforma. Cuanto más cerca del borde nos situemos, más rápida será la acción del robodroide.

Cuando encontremos dos robodroides en una misma habitación, habrá que controlarlos alternativamente. Esto es, cada vez que bajemos de la plataforma y volvamos a subir, se pondrá en marcha el que estaba detenido y viceversa.



inicial no era la más adecuada. En este sentido la práctica ayuda bastante.

Algunos de los objetos se encuentran ocultos en lugares insospechados, por lo que es recomendable investigar incluso en los sitios más recónditos. Llegar hasta otros nos parecerá imposible, pero siempre hay alguna forma, aunque se tarde en descubrirla.

No existe ninguna habitación ni lugar adonde no pueda accederse o desde donde no pueda salirse: todo es cuestión de paciencia y habilidad.

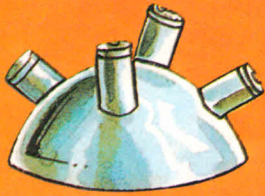
Un objeto puede recogerse estando en diagonal con él, por lo que a veces no será necesario saltar paredes u otros obstáculos para hacerse con él. Una última recomendación: no dejar nunca objetos en aquellas habitaciones donde haya elementos móviles ya que éstos, en sus desplazamientos, pueden destruirlos, sobre todo al cambiar de pantalla, y hay que recordar que el número de ellos es sólo ligeramente superior al necesario.

Huevos de alienígena

Están desprovistos de movimientos pero no por ello dejan de ser mortalmente peligrosos. A menudo se encuentran ocultos bajo bloques de piedra aparentemente sólidos que desaparecen bruscamente bajo nuestro peso, provocando la muerte instantánea.

Elipsoide magnético

Es, con diferencia, el más diabólico de los enemigos con el que podemos tropezarnos. Dotado de un detector de presencia magnética, persigue sin descanso a nuestro protagonista (no hay que olvidar que ALIEN-8 es metálico) y la única posibilidad de supervivencia es saltar por encima de él cuando le tengamos suficientemente próximo.



Minas explosivas

Sólo los ratones mecánicos y los robodroides teledirigidos están capacitados para sobrevivir al impacto contra uno de estos destructivos objetos.

AMSTRAD

Teclaea y salva en cinta el Programa CARGALIEN para Amstrad.

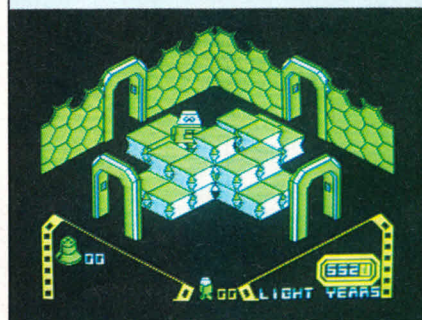
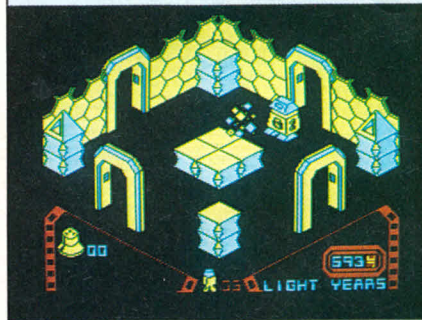


Una vez hecho esto tendrás unos magníficos programas cargadores que se utilizan con las CINTAS ORIGINALES del ALIEN 8 de Amstrad:

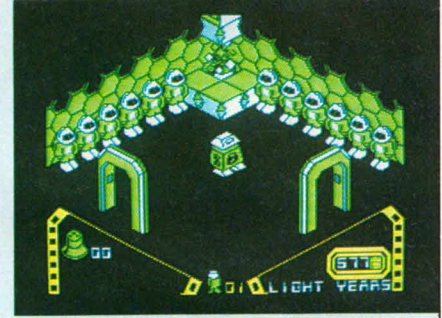
Sigue al pie de la letra las instrucciones de pantalla, indicando en cada caso «s» o «n» a las distintas opciones, poniendo por último el programa original. Hala, a disfrutarlo.

PROGRAMA «CARGALIEN» Amstrad |

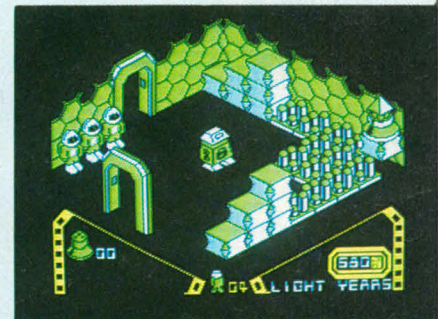
```
10 OPENOUT "A":MEMORY &109F:LOAD "ALIEN"
:CLS
20 POKE &1134,&95:POKE &113D,&15
30 POKE &1142,&C3:POKE &1143,193:POKE &1
144,160
40 FOR I=41000 TO 41165:READ A:POKE I,A:
NEXT I
100 INPUT "VIDAS INFINITAS S O N";V$
110 IF V$="N" THEN POKE 41096,&35
120 INPUT "NUMERO DE VIDAS DE 0 A 99";N
130 POKE 41105,N
140 INPUT "TIEMPO DE &0 A &9999";T
150 POKE 41111,INT(T/256):POKE 41110,T-I
NT(T/256)*256
160 INPUT "TIEMPO INFINITO S O N";T$
170 IF T$="N" THEN POKE 41116,0
```



```
180 INPUT "NUMERO DE HABITACIONES 1 A 24"
";N
190 POKE 41118,N
200 INPUT "ATRAVESAR OBJETOS S O N";A$
210 IF A$="N" THEN POKE 41123,2
220 INPUT "INMOVILIZAR RATAS S O N";I$
230 IF I$="N" THEN POKE 41128,0
240 INPUT "INMOVILIZAR ROBOT S O N";R$
250 IF R$="N" THEN POKE 41130,0
260 INPUT "INMOVILIZAR RATON S O N";I$
270 IF I$="N" THEN POKE 41132,0
280 INPUT "INMOVILIZAR FUERZA S O N";F$
```



```
290 IF I$="N" THEN POKE 41134,0
300 INPUT "INMOVILIZAR ANDROIDE S O N";I$
310 IF I$="N" THEN POKE 41136,0
320 INPUT "INMOVILIZAR BLOQUES S O N";I$
330 IF I$="N" THEN POKE 41138,0
340 INPUT "INMOVILIZAR ESTRELLAS S O N";I$
350 IF I$="N" THEN POKE 41140,0
360 INPUT "INMUNE A LOS OBJETOS S O N";N$
370 IF N$="N" THEN POKE 41142,0
380 CALL &1100
400 DATA &21,&DE,&08,&CD,&68,&A4,&18,&6D
,&21,&63,&12,&CD,&68,&A4,&21,&99,&11,&CD
,&68,&A4,&18,&6B,&21,&8F,&0B,&CD,&68,&A4
,&18,&65,&21,&FC,&0B,&CD,&68,&A4,&18,&5F
,&21,&C3,&12,&CD,&68,&A4,&18,&59,&21,&EA
410 DATA &0B,&CD,&68,&A4,&18,&53,&21,&7F
,&14,&CD,&68,&A4,&21,&93,&14,&CD,&68,&A4
,&21,&A7,&14,&CD,&68,&A4,&18,&41,&21,&81
,&0A,&CD,&68,&A4,&21,&9B,&0A,&CD,&68,&A4
,&18,&35,&AF,&77,&23,&77,&23,&77,&C9,&3E
420 DATA &00,&32,&8E,&21,&AF,&32,&4A,&00
,&3E,&05,&32,&22,&00,&21,&3C,&00,&22,&27
,&00,&18,&8B,&3E,&18,&32,&5E,&11,&3E,&00
,&32,&34,&15,&1B,&87,&18,&93,&18,&99,&18
,&9F,&18,&A5,&18,&AB,&18,&BD,&18,&03,&C3
,&00,&00,&3E,&C9,&32,&C5,&13,&18,&F6,&21
430 DATA &00,&15,&11,&00,&00,&01,&00,&80
,&ED,&80,&18,&B9
```

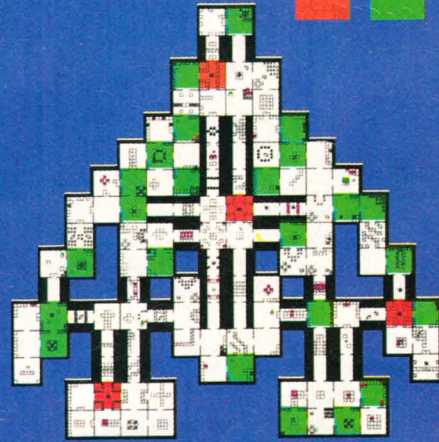
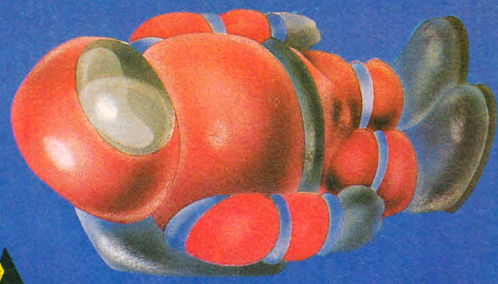
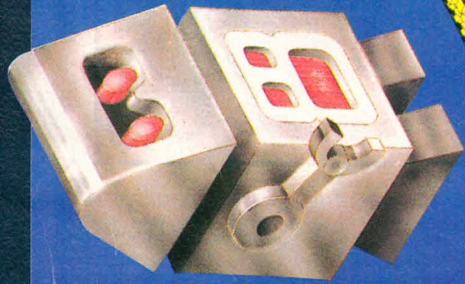


MAPA DE ALIEN 8

El cyboide ALIEN-8 tiene bajo su responsabilidad la importante misión de conservar con vida a los astronautas criogenizados hasta la llegada de la nave al planeta de destino.

Este cometido sólo puede cumplirse adecuadamente activando todas las válvulas termólicas presentes en cada una de las veinticuatro salas donde se mantienen los criogenautas.

Con la ayuda del MAPA podrás orientarte por entre los intrincados pasillos que conforman la estructura laberíntica de la aeronave.



■ Válvulas termólicas
■ Salidas

Ofites Informática

Presenta:

el lápiz al que gusta decir **SI**
mientras nuestros competidores dicen no

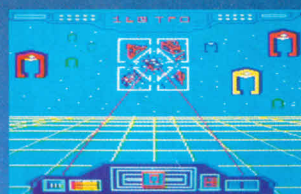
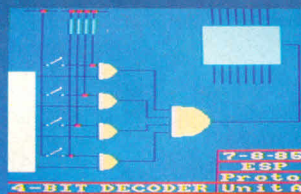
UNICO PARA AMSTRAD, CON PRECISION PIXEL

FUNCIONES	ESP	dk tronics	OTROS
UNICO MENU DE PANTALLA	SI	NO	
ARRASTRE OBJETOS PANTALLA	SI	NO	
TRASLADO OBJETOS PANTALLA	SI	NO	
TRASLADO DE CURSOR	SI	NO	
CAJAS ELASTICAS	SI	SI	
LINEA ELASTICA	SI	SI	
TRIANGULO ELASTICO	SI	NO	
ELIPSE ELASTICO	SI	NO	
DIAMANTE ELASTICO	SI	NO	
POLIGONO ELASTICO	SI	NO	
HEXAGONO ELASTICO	SI	NO	
OCTOGONO ELASTICO	SI	NO	
CUBO ELASTICO	SI	NO	
PIRAMIDE ELASTICA	SI	NO	
CIRCUNFERENCIAS	SI	SI	
CIRCULOS RELLENOS	SI	NO	
CAJAS RELLENAS	SI	NO	
ELIPSES RELLENAS	SI	NO	
CUÑAS	SI	NO	
SIMULADOR DE CORTES	SI	NO	
DISEÑO DE ZOOM	SI	SI	
IMAGEN ESPEJO E INVERTIDA	SI	NO	
FONDO DE REFERENCIA	SI	NO	
REJILLA DE FONDO	SI	NO	
OPCION DISPLAY X, Y	SI	NO	
RELLENADO CON COLOR	SI	SI	
LAVADO DE COLOR	SI	NO	
VOLCADO PANTALLA RESIDENTE	SI	NO	
DIBUJO DE BORDES EN 3 D	SI	NO	
TEXTO	SI	SI	
9 TAMAÑOS DE BROCHA	SI	NO	
18 TOBERAS MOSTRADORAS	SI	NO	
4 MEZCLAS BASICAS	SI	NO	
VARIADOR DE MEZCLAS	SI	NO	
SOMBREADO DE MEZCLAS XOR	SI	NO	
FICHERO ICONOS RESIDENTES	SI	NO	
FICHERO RELLENOS RESIDENTES	SI	NO	
26 COLORES DE PAPEL	SI	NO	
PALETA DE 15 TONOS DE COLOR	SI	NO	
POSICIONAMIENTO DE PUNTO	SI	SI	
RAYOS DESDE UN PUNTO FIJO	SI	NO	
DIBUJO REFLEJADO (ESPEJO)	SI	NO	
FUNCION HOME	SI	NO	
CONTROL DESDE TECLADO	SI	SI	
CONTROL CON JOYSTICK	SI	NO	
DISPONIBLES MODOS 1 Y 2	SI	?	

Compare con otros lápices



ESTOS SON
ALGUNOS EJEMPLOS
DE LOS GRAFICOS QUE VD.
PODRA REALIZAR CON NUESTRO
LAPIZ OPTICO



DE VENTA EN LOS MEJORES COMERCIOS
DE INFORMÁTICA

Si Vd. tiene alguna dificultad para obtener el lápiz óptico,
puede dirigirse a:

DISPONIBLE PARA:

- CPC 464 CASSETTE 4.900 Ptas.
- CPC 464-664 DISCO 6.900 Ptas.
- CPC 6128 DISCO 6.900 Ptas.

(IVA no incluido)

CONDICIONES ESPECIALES PARA DISTRIBUIDORES



Avda. Isabel II, 16 -8º
Tels. 455544 - 455533
Télex 36698
20011 SAN SEBASTIAN

UN MUNDO EN ALTA RESOLUCION

Veremos hoy la forma de trabajar en alta resolución desde código máquina. Estudiaremos cada uno de los comandos Basic y cómo simularlos desde código máquina.



El programa propuesto para esta semana realiza un cuadrado en pantalla con diferentes tintas proporcionando un marco para el texto a imprimir.

Como todos sabemos, los comandos de alta resolución de Amstrad, pueden ir acompañados hasta de cuatro parámetros, que son: coordenada X, coordenada Y, tinta y modo de tinta.

PLOT, X, Y, T, M

El modo de tinta se refiere a la forma en que se desea imprimir en pantalla. Existen cuatro modos de tinta utilizables, que son los siguientes:

NORMAL
XOR
AND
OR

Cuando se escribe en modo NORMAL, se imprime el estado actual borrando lo que existía anteriormente en ese punto.

En modo XOR, si nosotros deseamos imprimir un punto en una zona donde ya existe otro, ese punto se pone a cero, es decir, se borra de la pantalla. Si no existiese ningún punto en ese lugar, entonces se imprimiría el nuestro.

Al pintar en modo AND, únicamente se iluminará el pixel, si en el punto donde se imprime ya existe otro pixel iluminado.

Por último en modo OR, cuando se imprime un punto, éste se pinta sea cual sea el estado anterior de dicho pixel.

Cuando se trabaje en código máquina, no podremos seleccionar con una sola llamada los cuatro parámetros utilizables en los comandos de alta resolución Basic, sino que deberán ser elegidos realizando diferentes llamadas al firmware de nuestro Amstrad.

Veamos en primer lugar cuál es la rutina que selecciona el modo de tinta con el que se desea imprimir.

Modo de escritura para gráficos #BC59

Selecciona el modo de escritura para gráficos en alta resolución.

Condiciones de entrada

El registro A debe contener el modo de escritura elegido. Los valores correspondientes a cada modo, son los siguientes:

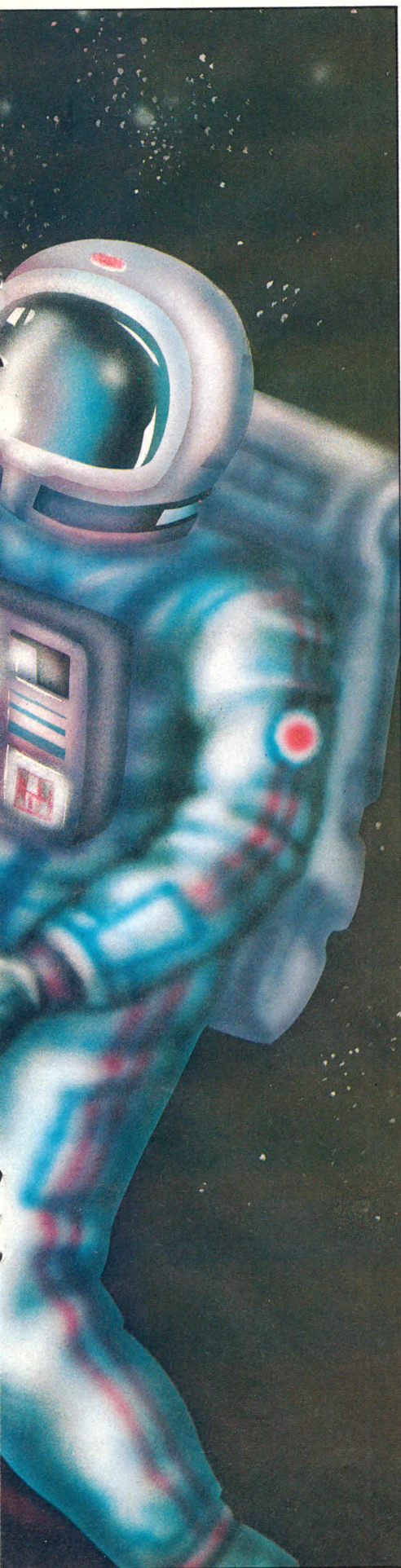
0...NORMAL
1...XOR
2...AND
3...OR

Condiciones de salida

Los registros AF, BC, DE y HL se corrompen, y se preservan los demás.

Otra de las llamadas que se deben hacer, es a la rutina encargada de seleccionar la tinta para los gráficos en alta resolución.





Selecciona la tinta para gráficos #BBDE

Selecciona la tinta para trabajar en alta resolución.

Condiciones de entrada

El acumulador debe contener el número de tinta deseado.

Condiciones de salida

Se corrompe el registro AF y se preservan los restantes.

Así pues, lo primero que hace nuestro programa en código máquina, es elegir el modo de pantalla de trabajo, y a continuación seleccionar el modo de escritura.

Una vez hecho esto, deberemos colocar el puntero de gráficos en el lugar donde deseamos realizar nuestro dibujo que en este caso será un cuadrado.

Para ello, deberemos llamar a una rutina del firmware que realice la función de PLOT (en este caso realizaremos un plot absoluto).

Coloca un punto en una posición absoluta #BBEA

La actual posición del cursor de gráficos, se coloca en la posición indicada. Si el punto que se desea imprimir se encuentra fuera de las coordenadas de la pantalla, entonces no se efectúa ninguna operación.

Condiciones de entrada

El registro doble DE, debe contener la coordenada X del punto a imprimir, y el registro HL debe contener la coordenada Y.

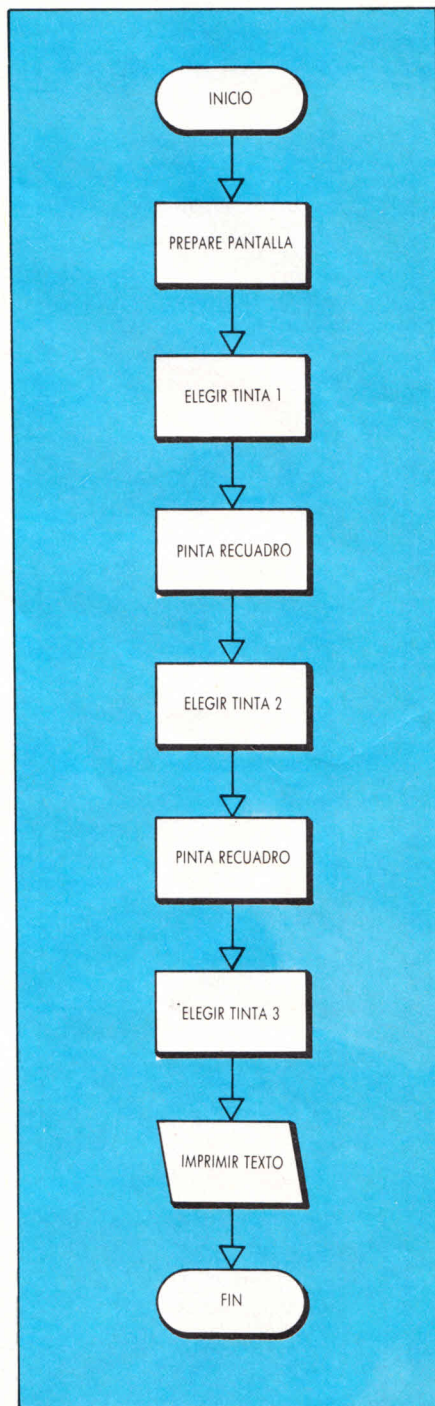
Condiciones de salida

Se corrompen los registros AF, BC, DE y HL, los demás son preservados.

Una vez colocado el cursor de gráficos en la posición adecuada, iniciaremos la impresión de nuestro cuadrado, mediante 'PLOT' (plots relativos), con los cuales iremos componiendo las líneas adecuadas.

Dado que las líneas que queremos imprimir tienen una longitud de 150 pixel las verticales y 250 las horizontales, realizaremos este trabajo mediante un bucle, dentro del cual se producirá la llamada a la rutina del firmware encargada de realizar los plots relativos.

Código MAQUINA



Efectúa un plot relativo #BBED

La posición actual del cursor gráfico, se coloca en la posición dada, tomando como coordenadas de referencia las de la anterior posición.

Condiciones de entrada

```

10          ORG #A000
20 ;
30 ;MODE 1
40 ;
50          LD A,1
60          CALL #BC0E
70 ;
80 ;MOVE 0,0
90 ;
100         LD DE,0
110        LD HL,0
120        CALL #BBF0
130 ;
140 ;DIBUJOS DE LAS ESQUINA
150 ;
160        LD B,15
170 BUC1:   PUSH BC
180        LD DE,5
190        LD HL,3
200        CALL #BBF3
210        CALL DRAWR
220        POP BC
230        DJNZ BUC1
240 ;
250 ;MOVE 0,299
260 ;
270        LD DE,0
280        LD HL,299
290        CALL #BBF0
300 ;
310 ;
320 ;
330        LD B,15
340 BUC2:   PUSH BC
350        LD DE,5
360        LD HL,-3
370        CALL #BBF3
380        CALL DRAWR
390        POP BC
400        DJNZ BUC2
410 ;
420 ;MOVE 539,0
430 ;
440        LD DE,539
450        LD HL,0
460        CALL #BBF0
470 ;
480 ;
490 ;
500        LD B,15
510 BUC3:   PUSH BC
520        LD DE,-5
530        LD HL,3
540        CALL #BBF3
550        CALL DRAWR
560        POP BC
570        DJNZ BUC3
580 ;
590 ;MOVE 539,299
600 ;
610        LD DE,539
620        LD HL,299
630        CALL #BBF0
640 ;
650 ;
660 ;
670        LD B,15
680 BUC4:   PUSH BC
690        LD DE,-5
700        LD HL,-3
710        CALL #BBF3
720        CALL DRAWR
730        POP BC
740        DJNZ BUC4
750 ;
760 ;PINTA LINEAS HORIZONTALES Y
770 ;VERTICALES
780 ;
790        LD DE,0
800        LD HL,150
810        CALL #BBF0
820        LD DE,639
830        LD HL,150
840        CALL #BBF6
850        LD DE,0
860        LD HL,248
870        CALL #BBF0

```

```

880        LD DE,639
890        LD HL,248
900        CALL #BBF6
910        LD DE,182
920        LD HL,399
930        CALL #BBF0
940        LD DE,182
950        LD HL,0
960        CALL #BBF6
970        LD DE,457
980        LD HL,399
990        CALL #BBF0
1000       LD DE,457
1010       LD HL,0
1020       CALL #BBF6
1030       CALL #BB18
1040       RET
1050 ;
1060 ;PINTA UN CUADRADO
1070 ;
1080 DRAWR: LD DE,0
1090       LD HL,100
1100       CALL #BBF9
1110       LD DE,100
1120       LD HL,0
1130       CALL #BBF9
1140       LD DE,0
1150       LD HL,-100
1160       CALL #BBF9
1170       LD DE,-100
1180       LD HL,0
1190       CALL #BBF9
1200       RET

```

```

10 REM *PROGRAMA CARGADOR*
20 FOR N=&A000 TO &A0E2
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>&6277 THEN PRINT "ERROR
  EN DATAS"
70 DATA 62,1,205,14,188,17,0
80 DATA 0,33,0,0,205,240,187
90 DATA 6,15,197,17,5,0,33
100 DATA 3,0,205,243,187,205,189
110 DATA 160,193,16,240,17,0,0
120 DATA 33,43,1,205,240,187,6
130 DATA 15,197,17,5,0,33,253
140 DATA 255,205,243,187,205,189,16
150 DATA 193,16,240,17,27,2,33
160 DATA 0,0,205,240,187,6,15
170 DATA 197,17,251,255,33,3,0
180 DATA 205,243,187,205,189,160,19
190 DATA 16,240,17,27,2,33,43
200 DATA 1,205,240,187,6,15,197
210 DATA 17,251,255,33,253,255,205
220 DATA 243,187,205,189,160,193,16
230 DATA 240,17,0,0,33,150,0
240 DATA 205,240,187,17,127,2,33
250 DATA 150,0,205,246,187,17,0
260 DATA 0,33,248,0,205,240,187
270 DATA 17,127,2,33,248,0,205
280 DATA 246,187,17,182,0,33,143
290 DATA 1,205,240,187,17,182,0
300 DATA 33,0,0,205,246,187,17
310 DATA 201,1,33,143,1,205,240
320 DATA 187,17,201,1,33,0,0
330 DATA 205,246,187,205,24,187,201
340 DATA 17,0,0,33,100,0,205
350 DATA 249,187,17,100,0,33,0
360 DATA 0,205,249,187,17,0,0
370 DATA 33,156,255,205,249,187,17
380 DATA 156,255,33,0,0,205,249
390 DATA 187,201,1,0,255,71,73

```

El registro doble DE contendrá el desplazamiento de la coordenada X, y en el registro HL se debe indicar el desplazamiento en la coordenada Y.

Condiciones de salida

Se preservan todos los registros excepto AF, BC, DE y HL.

De esta forma, si deseamos desplazar la posición del cursor de gráficos un pixel hacia arriba, deberemos colocar en los registros DE y HL los siguientes valores:

```
LD DE,0
LD HL,1
```

y a continuación realizar la siguiente llamada:

```
CALL #BBED
```

Así pues cargando dichos registros con diferentes valores, obtendremos las líneas verticales y horizontales que se desean para realizar el cuadrado.

Este está compuesto por tres pixel de espesor, cada uno de los cuales se pinta con diferente color, esto se consigue eligiendo cada uno de los colores, y a continuación efectuando la llamada a la rutina encargada de hacer el recuadro mediante plots relativos.

Por último, se imprime en medio de dicho recuadro el texto deseado, para ello se cargan los registros HL y DE con las coordenadas de impre-

sión y la dirección de nuestro texto respectivamente, y se efectúa la llamada a la rutina 'PRINT', estudiada en anteriores capítulos.

Hemos visto hasta ahora algunos de los comandos de alta resolución que se pueden ejecutar en nuestro ordenador, dejamos para el próximo capítulo el estudio de los restantes.

```

10 MODE 1
20 MOVE 0,0
30 FOR n=1 TO 15
40 MOVER 5,3
50 GOSUB 230
60 NEXT
70 MOVE 0,299
80 FOR n=1 TO 15
90 MOVER 5,-3
100 GOSUB 230
110 NEXT
120 MOVE 539,0
130 FOR n=1 TO 15
140 MOVER -5,3
150 GOSUB 230
160 NEXT
170 MOVE 539,299
180 FOR n=1 TO 15
190 MOVER -5,-3
200 GOSUB 230
210 NEXT
211 MOVE 0,150:DRAW 639,150
212 MOVE 0,248:DRAW 639,248
213 MOVE 182,399:DRAW 182,0
214 MOVE 457,399:DRAW 457,0
215 WHILE INKEY#="" :WEND
220 END
230 DRAWR 0,100:DRAWR 100,0:DRAWR 0
  -100:DRAWR -100,0
240 RETURN

```

Ya está a la venta el estuche n.º 2 de MICROHOBBY AMSTRAD

(Para contener los números 21 a 40 y las cintas 6 a 10)

¡Solicítalo hoy mismo y conserva ordenada tu colección!

Porque Microhobby Amstrad no es sólo una revista. Es una guía completa de tu ordenador Amstrad, en la que podrás consultar todas las dudas que se planteen. El estuche te permite tener todas tus revistas y cintas perfectamente ordenadas, y acceder a ellas

en cualquier momento sin necesidad de complicadas manipulaciones. Además, un índice facilita la localización del tema que te interese en breves segundos. Conserva tu colección de Microhobby Amstrad como se merece, y no dejes que se deteriore tu obra de consulta.



Recorta o copia este cupón y envíalo a Hobby Press, S. A. Apartado de Correos 232. Alcobendas (Madrid)

APELLIDOS _____ NOMBRE _____ FECHA DE NACIMIENTO _____
DOMICILIO _____ LOCALIDAD _____
PROVINCIA _____
C. POSTAL _____ TELÉFONO _____ PROFESIÓN _____ ¿ERES SUSCRITOR DE MICROHOBBY AMSTRAD? _____

Deseo recibir en mi domicilio el estuche número 2 (diseñado para contener las revistas 21 a 40 y las cintas 6 a 10) al precio de 670 ptas.

Forma de pago:

- Mediante talón bancario adjunto a nombre de Hobby Press, S. A.
 Mediante giro postal n.º _____
 Mediante tarjeta de crédito Visa Master Charge American Express

Número de la tarjeta _____

Fecha de caducidad de la tarjeta _____

Fecha y firma _____

INVERSION DE PANTALLAS



En esta misma sección ya os hemos presentado anteriormente varias rutinas que tratan del manejo de la pantalla, como la de magnificación o reducción. Ahora le toca el turno a la de inversión de pantalla, tanto vertical como horizontalmente.

El programa está ubicado a partir de la dirección de memoria &A000 hexadecimal, y tiene una longitud de 455 bytes. Además, utiliza unos buffers a partir de la dirección &2000 donde se realiza el trabajo de inversión.

Para poder utilizarla con una pantalla realizada anteriormente, deberemos escribir un programa Basic como el siguiente:

```
10 MODE N
29 MEMORY &1FFF
30 LOAD "INVER", &A000
40 LOAD "PANTALLA", &C000
50 CALL &A000
```

donde 'N' deberá ser el modo de

pantalla en el que se haya realizado la pantalla.

Tal como se ha escrito el programa anterior, se obtendría una inversión vertical de la pantalla. Si lo que se desea es una inversión horizontal, deberemos cambiar la línea 50 por la siguiente:

```
50 CALL &A000,1
```

Aunque para cada modo de pantalla se debe utilizar una rutina distinta, no debemos preocuparnos de ello ya que nuestro programa detecta el modo en el que se está trabajando, y llama a la rutina adecuada.

Para realizar la inversión de pantalla en forma vertical, se utiliza una sola rutina, ya que esta no depende del modo de pantalla en que se esté trabajando.

Esto es debido a que para realizar este tipo de inversión, se trabaja únicamente con bytes y no con bits.

Así pues, lo que hacemos es: tomar el primer byte del carácter situado en la esquina superior izquierda de la pantalla y lo colocamos en el lugar del último byte del carácter situado en la esquina superior izquierda de la pantalla, colocando el que ocupaba esta última posición en el lugar del primero, y así sucesivamente.

En lo que se refiere a la inversión horizontal, el método de trabajo es completamente distinto, ya que en esta ocasión, se deberá trabajar con cada uno de los bits que componen cada byte, por lo que la rutina deberá ser distinta para cada modo de pantalla.

Veamos cuál es el método a emplear en cada uno de los modos de pantalla para obtener este último tipo de inversión de pantalla.

Cuando trabajamos en modo 2, todos los bits que componen un byte contienen la información de si un pixel debe estar encendido o apagado, es decir, si un bit está a cero, dicho pixel estará apagado, y por el contrario si está uno, lo veremos iluminado.

Por lo tanto, para este modo de pantalla deberemos tomar cada uno de los bits de cada byte e invertirlos. Además deberemos colocar el primer byte de cada fila y colocarlo al final de ésta, y este último colocarlo al principio.

Una vez hecho este intercambio, deberemos proceder a la inversión de los bits que componen cada byte.

Esta operación deberá efectuarse con cada uno de los bytes que componen la pantalla.

Cuando se desee trabajar como el modo 1 de pantalla, en este caso, no todos los bits que componen cada byte llevan información de pixel iluminados o apagados, ya que en este caso únicamente se pueden obtener cuatro pixels por cada byte, por lo tanto, en este caso, cada dos bits llevan dicha información.

Elo es debido a que la información de color se guarda en esos dos bits, es por eso por lo que en este modo de pantalla se pueden obtener hasta cuatro colores, ya que son esos dos bits, es posible formar hasta cuatro combinaciones como se puede ver a continuación:

color 1	color 2	color 3	color 4
0001	10	11	

En este caso la rotación de los bits, se deberá efectuar por parejas. En este modo de pantalla, los bits que forman pareja son los siguientes:

- bits 0-4
- bits 1-5
- bits 2-6
- bits 3-7

La misma operación anterior deberá efectuarse con cada una de las parejas de bits.

Por último queda por ver cómo se deberá trabajar cuando nos encontremos en modo de baja resolución, es decir, en modo 0.

Como sabemos en este modo de pantalla, cada byte, tiene la posibilidad de encender o apagar dos pixel, es decir, cada cuatro bits llevan la información de cada uno de esos dos pixels.

Gracias a esta estructura es por la que se pueden obtener hasta 16 colores en este modo de pantalla, ya que ese valor es el número de variaciones que se pueden obtener con dicho número de bits.

Estas combinaciones se obtienen de forma idéntica a las explicadas anteriormente en modo 1, con la única diferencia de que en este caso se trabajan con cuatro elementos.

Los grupos de bits que dan información sobre cada uno de los pixel, son los que se indican a continuación:

bits 7-5-3-1
bits 6-4-2-0

De esta forma se obtiene cada una de las inversiones para cada uno de los modos de pantalla, ya que la rutina detecta automáticamente en qué modo se está trabajando.

Debemos tener en cuenta que las

operaciones anteriormente mencionadas, se deben efectuar sobre los 16384 bytes que componen la pantalla.

Si deseamos realizar diversos tipos de inversiones en una pantalla que hayamos preparado previamente, podemos realizar un programa en Basic que incluya hasta la línea 40 del programa que hemos visto al principio de este artículo, y además posea las siguientes líneas:

```
50 IF INKEY$="1" THEN CALL
&A000
60 IF INKEY$="2" THEN CALL
&A000, 1
70 IF INKEY$="S" THEN SAVE
"PANTA", B,&C000, &4000
80 GOTO 50
```

De este modo realizaremos las inversiones que deseemos y salvar la pantalla en disco cuando hayamos

PROGRAMACION

terminado, sin ensuciar la pantalla inicial.

Para salvar en disco la rutina de inversión de pantalla, deberéis copiar el listado ensamblador indicando como origen la dirección &A000.

Aquellos que no posean ensamblador, pueden copiar el cargador Basic listado a continuación, y una vez hecho, ejecutar el programa, en el caso de que no exista ningún error, podremos salvar la rutina de la forma siguiente:

```
SAVE "INVER",B,&A000,455
```

con lo que ya tendremos el programa listo para ejecutarlo cuando deseemos.

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```
A000          10      ORG  #A000
A000 FE00      20      CP  0
A002 CAFFA0   30      JP  Z,INVER
A005 CD11BC   40      CALL #BC11
A008 FE00      50      CP  0
A00A CA15A0   60      JP  Z,MOD00
A00D FE01      70      CP  1
A00F CA8AA0   80      JP  Z,MOD01
A012 C3D5A0   90      JP  MOD02
A015 CD2CA1  100     MOD00: CALL STORE
A018 210020  110     LD  HL,#2000
A01B 114F60  120     LD  DE,#6000+79
A01E 06C8    130     LD  B,200
A020 C5      140     NUD01: PUSH BC
A021 E5      150     PUSH HL
A022 D5      160     PUSH DE
A023 EB      170     EX  DE,HL
A024 0650    180     LD  B,00
A026 C5      190     NUD0K: PUSH BC
A027 1A      200     LD  A,(DE)
A028 3600    210     LD  (HL),0
A02A 0602    220     LD  B,2
A02C 1F      230     KK1:  RRA
A02D CB16    240     RL  (HL)
A02F 10FB    250     DJNZ KK1
A031 0E00    260     LD  C,0
A033 0602    270     LD  B,2
A035 1F      280     KK2:  RRA
A036 CB11    290     RL  C
A038 10FB    300     DJNZ KK2
A03A F5      310     PUSH AF
A03B CB11    320     RL  C
A03D CB11    330     RL  C
A03F 79      340     LD  A,C
A040 B6      350     OR  (HL)
A041 77      360     LD  (HL),A
A042 F1      370     POP  AF
A043 0E00    380     LD  C,0
A045 0602    390     LD  B,2
A047 1F      400     KK3:  RRA
A048 CB11    410     RL  C
A04A 10FB    420     DJNZ KK3
A04C F5      430     PUSH AF
A04D CB11    440     RL  C
```

```
A04F CB11      450     RL  C
A051 CB11      460     RL  C
A053 CB11      470     RL  C
A055 79        480     LD  A,C
A056 B6        490     OR  (HL)
A057 77        500     LD  (HL),A
A058 0E00      510     LD  C,0
A05A F1        520     POP  AF
A05B 0602      530     LD  B,2
A05D 1F        540     KK4:  RRA
A05E CB11      550     RL  C
A060 10FB      560     DJNZ KK4
```

Hisoft GENA3.1 Assembler. Page 2.

```
A062 CB11      570     RL  C
A064 CB11      580     RL  C
A066 CB11      590     RL  C
A068 CB11      600     RL  C
A06A CB11      610     RL  C
A06C CB11      620     RL  C
A06E 79        630     LD  A,C
A06F B6        640     OR  (HL)
A070 77        650     LD  (HL),A
A071 13        660     INC  DE
A072 2B        670     DEC  HL
A073 C1        680     POP  BC
A074 10B0      690     DJNZ NUDOK
A076 E1        700     POP  HL
A077 1600      710     LD  D,0
A079 1E50      720     LD  E,80
A07B 19        730     ADD  HL,DE
A07C EB        740     EX  DE,HL
A07D E1        750     POP  HL
A07E 0600      760     LD  B,0
A080 0E50      770     LD  C,80
A082 09        780     ADD  HL,BC
A083 C1        790     POP  BC
A084 109A      800     DJNZ NUD01
A086 CD36A1    810     CALL PINTA
A089 C9        820     RET
A08A CD2CA1    830     MOD01: CALL STORE
A08D 210020    840     LD  HL,#2000
A090 114F60    850     LD  DE,#6000+79
A093 06C8      860     LD  B,200
```

```

A095 C5      870  NUD03:  PUSH BC
A096 E5      880      PUSH HL
A097 D5      890      PUSH DE
A098 EB      900      EX  DE,HL
A099 0650    910      LD  B,80
A09B C5      920  NUD0X:  PUSH BC
A09C 1A      930      LD  A,(DE)
A09D 3600    940      LD  (HL),0
A09F 0E00    950      LD  C,0
A0A1 0604    960      LD  B,4
A0A3 1F      970  NUD0L:  RRA
A0A4 CB16    980      RL  (HL)
A0A6 10FB    990      DJNZ NUD0L
A0A8 0604   1000     LD  B,4
A0AA 1F     1010  NUD0M:  RRA
A0AB CB11   1020     RL  C
A0AD 10FB   1030     DJNZ NUD0M
A0AF CB11   1040     RL  C
A0B1 CB11   1050     RL  C
A0B3 CB11   1060     RL  C
A0B5 CB11   1070     RL  C
A0B7 79     1080     LD  A,C
A0B8 E6F0   1090     AND 240
A0BA B6     1100     OR  (HL)
A0BB 77     1110     LD  (HL),A
A0BC 13     1120     INC DE
A0BD 2B     1130     DEC HL
A0BE C1     1140     POP BC

```

Hisoft GENA3.1 Assembler. Page 3.

```

A0BF 10DA    1150     DJNZ NUD0X
A0C1 E1     1160     POP HL
A0C2 1600   1170     LD  D,0
A0C4 1E50   1180     LD  E,80
A0C6 19     1190     ADD HL,DE
A0C7 EB     1200     EX  DE,HL
A0C8 E1     1210     POP HL
A0C9 0600   1220     LD  B,0
A0CB 0E50   1230     LD  C,80
A0CD 09     1240     ADD HL,BC
A0CE C1     1250     POP BC
A0CF 10C4   1260     DJNZ NUD03
A0D1 CD36A1 1270     CALL PINTA
A0D4 C9     1280     RET
A0D5 CD2CA1 1290  MOD02:  CALL STORE
A0D8 210020 1300     LD  HL,#2000
A0DB 114F60 1310     LD  DE,#6000+79
A0DE 06C8   1320     LD  B,200
A0E0 C5     1330  NUD07:  PUSH BC
A0E1 E5     1340     PUSH HL
A0E2 D5     1350     PUSH DE
A0E3 0650   1360     LD  B,80
A0E5 7E     1370  NUD08:  LD  A,(HL)
A0E6 12     1380     LD  (DE),A
A0E7 1B     1390     DEC DE
A0E8 23     1400     INC HL
A0E9 10FA   1410     DJNZ NUD08
A0EB E1     1420     POP HL
A0EC 1600   1430     LD  D,0
A0EE 1E50   1440     LD  E,80
A0F0 19     1450     ADD HL,DE
A0F1 EB     1460     EX  DE,HL
A0F2 E1     1470     POP HL
A0F3 0600   1480     LD  B,0
A0F5 0E50   1490     LD  C,80
A0F7 09     1500     ADD HL,BC
A0F8 C1     1510     POP BC
A0F9 10E5   1520     DJNZ NUD07
A0FB CD36A1 1530     CALL PINTA
A0FE C9     1540     RET
A0FF CD2CA1 1550  INVER:  CALL STORE
A102 210020 1560     LD  HL,#2000
A105 11309E 1570     LD  DE,#9E30
A108 06C8   1580     LD  B,200
A10A C5     1590  NUD0A:  PUSH BC
A10B E5     1600     PUSH HL
A10C D5     1610     PUSH DE
A10D 0650   1620     LD  B,80
A10F 7E     1630  NUD09:  LD  A,(HL)
A110 12     1640     LD  (DE),A
A111 13     1650     INC DE

```

```

A112 23     1660     INC HL
A113 10FA   1670     DJNZ NUD09
A115 E1     1680     POP HL
A116 1600   1690     LD  D,0
A118 1E50   1700     LD  E,80
A11A 37     1710     SCF
A11B 3F     1720     CCF

```

Hisoft GENA3.1 Assembler. Page 4.

```

A11C ED52    1730     SBC HL,DE
A11E EB     1740     EX  DE,HL
A11F E1     1750     POP HL
A120 0600   1760     LD  B,0
A122 0E50   1770     LD  C,80
A124 09     1780     ADD HL,BC
A125 C1     1790     POP BC
A126 10E2   1800     DJNZ NUD0A
A128 CD36A1 1810     CALL PINTA
A12B C9     1820     RET
A12C 210101 1830  STORE:  LD  HL,#0101
A12F 110020 1840     LD  DE,#2000
A132 CD40A1 1850     CALL LEER
A135 C9     1860     RET
1870 ;
1880 ;
A136 210101 1890  PINTA:  LD  HL,#0101
A139 110060 1900     LD  DE,#6000
A13C CD83A1 1910     CALL IMPRE
A13F C9     1920     RET
1930 ;
1940 ;
1950 ;RUTINA-LEERSION
1960 ;H-POSICION-VERTICAL-INICIO-1
1970 ;L-POSICION-HORIZONTAL-INICIO-1
1980 ;DE-DIRECCION-GRAFICO
1990 ;
2000 LEER:  PUSH DE
A141 EB     2010     EX  DE,HL
A142 21B0BF 2020     LD  HL,#C000-80
A145 42     2030     LD  B,D
A146 1600   2040     LD  D,0
A148 1D     2050     DEC E
A149 19     2060     ADD HL,DE
A14A 115000 2070     LD  DE,80
A14D 19     2080  S_LBU:  ADD HL,DE
A14E 10FD   2090     DJNZ S_LBU
A150 D0E1   2100     POP IX
A152 3A81A1 2110     LD  A,(ALTO)
A155 47     2120     LD  B,A
A156 1815   2130     JR  COLO
A158 7C     2140  P_LBU:  LD  A,H
A159 E638   2150     AND 56
A15B FE38   2160     CP  56
A15D 2806   2170     JR  Z,P_PA
A15F 7C     2180     LD  A,H
A160 C608   2190     ADD A,8
A162 67     2200     LD  H,A
A163 1808   2210     JR  COLO
A165 115000 2220  P_PA:  LD  DE,0080
A168 7C     2230     LD  A,H
A169 EE38   2240     XOR 56
A16B 67     2250     LD  H,A
A16C 19     2260     ADD HL,DE
A16D C5     2270  COLO:  PUSH BC
A16E E5     2280     PUSH HL
A16F 3A82A1 2290     LD  A,(ANCHO)
A172 47     2300     LD  B,A

```

Hisoft GENA3.1 Assembler. Page 5.

```

A173 7E     2310  P_BU1:  LD  A,(HL)
A174 DD7700 2320     LD  (IX+0),A
A177 DD23   2330     INC IX
A179 23     2340     INC HL
A17A 10F7   2350     DJNZ P_BU1

```

PROGRAMACION

```

A17C E1      2360      POP HL
A17D C1      2370      POP BC
A17E 10D8    2380      DJNZ P_BU
A180 C9      2390      RET
A181 C8      2400      ALTO: DEFB 200
A182 50      2410      ANCHO: DEFB 80
                2420 ;
                2430 ;
                2440 ;
A183 D5      2450      IMPRE: PUSH DE
A184 EB      2460      EX DE,HL
A185 21B0BF  2470      LD HL,#C000-80
A188 42      2480      LD B,D
A189 1600    2490      LD D,0
A18B 1D      2500      DEC E
A18C 19      2510      ADD HL,DE
A18D 115000  2520      LD DE,80
A190 19      2530      S_BUC: ADD HL,DE
A191 10FD    2540      DJNZ S_BUC
A193 DDE1    2550      POP IX
A195 3A81A1  2560      LD A,(ALTO)
A198 47      2570      LD B,A
A199 1815    2580      JR COLOC
A19B 7C      2590      P_BUC: LD A,H
A19C E638    2600      AND 56
A19E FE38    2610      CP 56
A1A0 2806    2620      JR Z,P_PAS
A1A2 7C      2630      LD A,H
A1A3 C608    2640      ADD A,8
A1A5 67      2650      LD H,A
A1A6 1808    2660      JR COLOC
A1A8 115000  2670      P_PAS: LD DE,0080
A1AB 7C      2680      LD A,H
A1AC EE38    2690      XOR 56
A1AE 67      2700      LD H,A
A1AF 19      2710      ADD HL,DE
A1B0 C5      2720      COLOC: PUSH BC
A1B1 E5      2730      PUSH HL
A1B2 3A82A1  2740      LD A,(ANCHO)
A1B5 47      2750      LD B,A
A1B6 DD7E00  2760      P_BUC1: LD A,(IX+0)
A1B9 77      2770      LD (HL),A
A1BA DD23    2780      INC IX
A1BC 23      2790      INC HL
A1BD 10F7    2800      DJNZ P_BUC1
A1BF E1      2810      POP HL
A1C0 C1      2820      POP BC
A1C1 10D8    2830      DJNZ P_BUC
A1C3 C9      2840      RET

```

Pass 2 errors: 00

ALTO A181 ANCHO A182 COLO A16D

Hisoft GENA3.1 Assembler. Page 6.

```

COLOC A1B0 IMPRE A183 INVER A0FF
KK1 A02C KK2 A035 KK3 A047
KK4 A05D LEER A140 MOD00 A015
MOD01 A08A MOD02 A0D5 NUD01 A020
NUD03 A095 NUD07 A0E0 NUD08 A0E5
NUD09 A10F NUD0A A10A NUD0K A026
NUDOL A0A3 NUDOM A0AA NUDOX A09B
PINTA A136 P_BU A158 P_BU1 A173
P_BUC A19B P_BUC1 A1B6 P_PA A165
P_PAS A1A8 STORE A12C S_BU A14D
S_BUC A190

```

Table used: 408 from 1000

INVERPANT CA

```

10 REM *PROGRAMA CARGADOR*
20 FOR N=&A000 TO &A1C4
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>46637 THEN PRINT "ERROR
  EN DATAS"
70 DATA 254,0,202,255,160,205,17
80 DATA 188,254,0,202,21,160,254

```

```

90 DATA 1,202,138,160,195,213,160
100 DATA 205,44,161,33,0,32,17
110 DATA 79,96,6,200,197,229,213
120 DATA 235,6,80,197,26,54,0
130 DATA 6,2,31,203,22,16,251
140 DATA 14,0,6,2,31,203,17
150 DATA 16,251,245,203,17,203,17
160 DATA 121,182,119,241,14,0,6
170 DATA 2,31,203,17,16,251,245
180 DATA 203,17,203,17,203,17,203
190 DATA 17,121,182,119,14,0,241
200 DATA 6,2,31,203,17,16,251
210 DATA 203,17,203,17,203,17,203
220 DATA 17,203,17,203,17,121,182
230 DATA 119,19,43,193,16,176,225
240 DATA 22,0,30,80,25,235,225
250 DATA 6,0,14,80,9,193,16
260 DATA 154,205,54,161,201,205,44
270 DATA 161,33,0,32,17,79,96
280 DATA 6,200,197,229,213,235,6
290 DATA 80,197,26,54,0,14,0
300 DATA 6,4,31,203,22,16,251
310 DATA 6,4,31,203,17,16,251
320 DATA 203,17,203,17,203,17,203
330 DATA 17,121,230,240,182,119,19
340 DATA 43,193,16,218,225,22,0
350 DATA 30,80,25,235,225,6,0
360 DATA 14,80,9,193,16,196,205
370 DATA 54,161,201,205,44,161,33
380 DATA 0,32,17,79,96,6,200
390 DATA 197,229,213,6,80,126,18
400 DATA 27,35,16,250,225,22,0
410 DATA 30,80,25,235,225,6,0
420 DATA 14,80,9,193,16,229,205
430 DATA 54,161,201,205,44,161,33
440 DATA 0,32,17,48,158,6,200
450 DATA 197,229,213,6,80,126,18
460 DATA 19,35,16,250,225,22,0
470 DATA 30,80,55,63,237,82,235
480 DATA 225,6,0,14,80,9,193
490 DATA 16,226,205,54,161,201,33
500 DATA 1,1,17,0,32,205,64
510 DATA 161,201,33,1,1,17,0
520 DATA 96,205,131,161,201,213,235
530 DATA 33,176,191,66,22,0,29
540 DATA 25,17,80,0,25,16,253
550 DATA 221,225,58,129,161,71,24
560 DATA 21,124,230,56,254,56,40
570 DATA 6,124,198,8,103,24,8
580 DATA 17,80,0,124,238,56,103
590 DATA 25,197,229,58,130,161,71
600 DATA 126,221,119,0,221,35,35
610 DATA 16,247,225,193,16,216,201
620 DATA 200,80,213,235,33,176,191
630 DATA 66,22,0,29,25,17,80
640 DATA 0,25,16,253,221,225,58
650 DATA 129,161,71,24,21,124,230
660 DATA 56,254,56,40,6,124,198
670 DATA 8,103,24,8,17,80,0
680 DATA 124,238,56,103,25,197,229
690 DATA 58,130,161,71,221,126,0
700 DATA 119,221,35,35,16,247,225
710 DATA 193,16,216,201,255,82,83

```



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

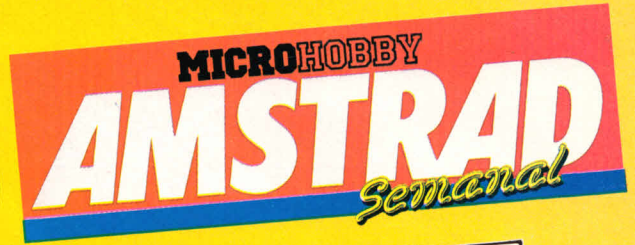
¡POR FIN, CARATULAS PARA TUS DISCOS!

Amstrad Semanal, en su primer Aniversario, ha pensado en haceros un regalo, tal vez modesto, pero sin duda útil, y realizado de todo corazón. Después de 50 números juntos, se han publicado multitud de cosas como juegos, utilidades, código máquina y otras muchas secciones que incluyen programas. Probablemente a todos os gustaría tenerlas ordenadas, y en disco, cosa que sin duda habéis hecho. Para decorarlos y distinguirlos un poco, estas carátulas creemos que os serán de gran ayuda y utilidad.

Utilidades	MICROHOBBY AMSTRAD <i>Semanal</i> Utilidades	CARA A _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ CARA B _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
	1. ^{er} ANIVERSARIO MICROHOBBY AMSTRAD SEMANAL 	

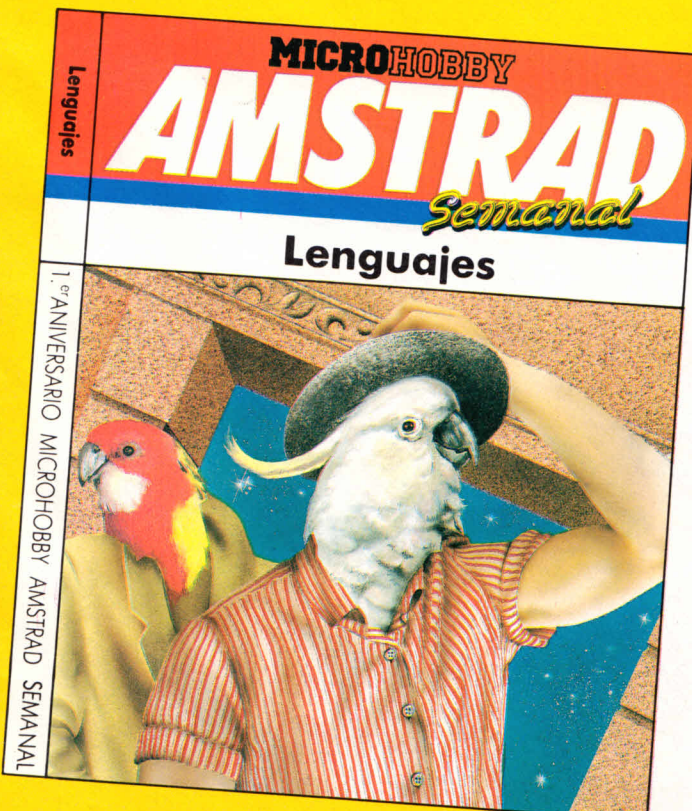
Juegos	MICROHOBBY AMSTRAD <i>Semanal</i> Juegos	CARA A _____ _____ _____ _____ _____ _____ _____ _____ _____ _____ CARA B _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
	1. ^{er} ANIVERSARIO MICROHOBBY AMSTRAD SEMANAL 	

1.º Aniversario



CARA A

CARA B



CARA A

CARA B

FISION

Fisión es un juego de estrategia bipersonal, basado en la idea de simular una reacción nuclear. A pesar de la sencillez de sus reglas, las complicadas estrategias que sobre él pueden desarrollarse son impredecibles.

Por Vicente Carriles Rodríguez



Sobre las instrucciones del juego poco hay que decir, pues vienen incorporadas en el propio programa. El juego se desarrolla sobre un tablero y existen dos tipos de fichas, uno para cada jugador. El objetivo consiste en dejar sin fichas al contrincante.

El programa lleva incorporado tres llamadas al firmware. Ca11&bb03, que limpia el buffer de teclado, y que además produce un efecto secundario que provoca que el programa no se pueda detener, por lo que no hay que olvidar grabarlo una vez teclado. Las otras dos llamadas &bb18 y 0 esperan a que se pulse una tecla y resetean el ordenador respectivamente.

A continuación damos la estructura del programa así como la tabla de variables por si alguien se anima a modificarlo incorporándole nuevas características y adaptándole a su propia personalidad.

ESTRUCTURA DEL PROGRAMA

10 - 100	Presentación del programa.
110 - 170	Se hacen las ventanas, dimensión de variables, creación de caracteres, define variables, elige colores y restaura el modo inicial.
180 - 360	Dibuja el tablero.
370 - 420	Imprime los contadores en cada una de las ventanas.
430 - 600	Empieza el juego, pide turnos y comprueba la jugada.
610 - 670	Actualiza contadores de fichas.
680 - 950	Hace las operaciones necesarias en caso de explosión.
960 - 990	Comprueba si hubo explosiones.
1000 - 1040	Imprime última jugada e imprime la situación de los contadores.
1050 - 1330	Imprime el actual estado del tablero.
1340 - 1470	Cuenta las fichas después de cada jugada y comprueba si ganó uno de los jugadores.
1480 - 1510	Borra fichas del tablero.
1520 - 1640	Imprime ganador y opción a seguir jugando o fin.
1650 - 1850	Instrucciones.

ULTIMA JUGADA		A B C D E					
NEGRAS C-4			●	●	●		1
FICHAS AMA.			●		●	●	2
DENTRO... 9		●		●	●	●	3
FUERA... 39			●		●		4
TOTAL... 48				●			5
FICHAS NEG.							
DENTRO... 13							
FUERA... 39							
TOTAL... 52							
GANAN LAS							
		TURNO AMARILLAS					

CA

```

10 REM ***** PRESENTACION ****
****
20 '
30 CALL &BB03:MODE 1:INK 1,24:INK 0
,1:INK 2,10:INK 3,0:PAPER 0:PEN 1:B
ORDER 1
40 PRINT TAB(12);"*** EXPLOSION ***
":LOCATE 7,4:PRINT CHR$(164);".JBS
& VCR 04/04/1986 (V1)":LOCATE 12,8:
PRINT"JUEGO DE TABLERO":LOCATE 13,1
1:PRINT"realizado por":LOCATE 8,13
:PRINT"Jose B. S. & Vicente C. R."
50 LOCATE 16,17:PRINT"para tu":LOC
ATE 10,19:PRINT"AMSTRAD CPC 464/612
8"
60 LOCATE 6,24:PRINT"QUIERES INSTRU
CCIONES (S/N)..?"
70 X%=INKEY$:X%=UPPER$(X%):IF X%="S
" THEN GOSUB 1650 ELSE IF X%="N"THE
N 110 ELSE 70
80 LOCATE 2,14:PRINT"QUIERE VOLVER
A VER LAS INSTRUCCIONES":PRINT" (S/
N)"
90 GOTO 70
100 '
110 REM ***** DEF VENTANAS,FICHAS *
****
120 '
130 MODE 1:PAPER#7,0:CLS#7:WINDOW#0
,1,40,1,25
140 DEFINT A-N:SYMBOL 250,60,126,25
5,255,255,255,126,60
150 BF=50:NF=50:DIM A$(5),B$(5),C$(
5),D$(5),E$(5),A(5),B(5),C(5),D(5),
E(5)
160 WINDOW#1,18,38,2,22:WINDOW#2,3,
15,2,4:WINDOW#3,3,15,7,11:WINDOW#4,
3,15,15,19:WINDOW#5,3,15,22,24:WIND
OW#6,18,38,24,24:PAPER#1,2:PEN#1,3:
CLS#1:BORDER 0
170 '
180 REM ***** DIBUJA EL TABLERO **
****
190 '
200 PRINT#1,CHR$(150);
210 FOR N=1 TO 5:FOR M=1 TO 3:PRINT
#1,CHR$(154);:NEXT M:PRINT#1,CHR$(1
58);:NEXT N
220 PRINT#1,CHR$(8)+CHR$(156);
230 FOR L=1 TO 4
240 GOSUB 310:PRINT#1,CHR$(151);
250 FOR N=1 TO 5:FOR M=1 TO 3:PRINT
#1,CHR$(154);:NEXT M:PRINT#1,CHR$(1
59);:NEXT N
260 PRINT#1,CHR$(8)+CHR$(157);
270 NEXT L

```

```

280 GOSUB 310:PRINT#1,CHR$(147);
290 FOR N=1 TO 5:FOR M=1 TO 3:PRINT
#1,CHR$(154);:NEXT M:PRINT#1,CHR$(1
55);:NEXT N
300 LOCATE 38,22:PAPER 2:PEN 3:PRIN
T CHR$(153):GOTO 320
310 FOR M=1 TO 3:PRINT#1,CHR$(149);
" ";CHR$(149);" ";CHR$(149);"
";CHR$(149);" ";CHR$(149);" ";
CHR$(149);:NEXT M:RETURN
320 INK 2,10:INK 3,0:PAPER 0:PEN 1:
);"B";TAB(28);"C";TAB(32);"D";TAB(3
6);"E"
330 FOR N=4 TO 20 STEP 4
340 F=F+1:F%=STR$(F):F%=MID$(F$,2,1
):LOCATE 39,N:PRINT F%
350 NEXT N
360 '
370 REM ***** NOMBRE DE CONTADORES
****
380 '

```



```

****
440 '
450 CALL &BB03:IF J%="" OR J%="NEGR
AS"THEN PEN#6,1:J%="AMARILLAS" ELSE
PEN#6,3:J%="NEGRAS"
460 PRINT#6,"TURNO ";J%;" ";
470 Z%=INKEY$:Z%=UPPER$(Z%):IF Z%(
HR$(69) OR Z%(CHR$(65) THEN 470 ELS
E PRINT#6,Z%;" ";
480 Y%=INKEY$:IF Y%<"1" OR Y%>"5" T
HEN 480 ELSE PRINT#6,Y%:G=VAL(Y%)
490 IF Z%="A" THEN 500 ELSE IF Z%="
B" THEN 520 ELSE IF Z%="C" THEN 540
ELSE IF Z%="D" THEN 560 ELSE IF Z%
="E" THEN 580

```

TABLA DE VARIABLES

J\$	Turno de jugada (amarillas, negras).
Z\$	Coordenadas de letras (A, B, C, D, E).
Y\$	Coordenadas de números (1, 2, 3, 4, 5).
A\$(N)	Guarda el color de cada una de las fichas que hay en las casillas.
B\$(N)	Guarda el color de cada una de las fichas que hay en las casillas.
C\$(N)	Guarda el color de cada una de las fichas que hay en las casillas.
D\$(N)	Guarda el color de cada una de las fichas que hay en las casillas.
E\$(N)	Guarda el color de cada una de las fichas que hay en las casillas.
A(N)	Guarda la cantidad de fichas que hay en cada una de las casillas.
B(N)	Guarda la cantidad de fichas que hay en cada una de las casillas.
C(N)	Guarda la cantidad de fichas que hay en cada una de las casillas.
D(N)	Guarda la cantidad de fichas que hay en cada una de las casillas.
E(N)	Guarda la cantidad de fichas que hay en cada una de las casillas.
BF	Contiene la cantidad de fichas amarillas que hay fuera del tablero.
BD	Contiene la cantidad de fichas amarillas que hay dentro del tablero.
NF	Contiene la cantidad de fichas negras que hay fuera del tablero.
ND	Contiene la cantidad de fichas negras que hay dentro del tablero.
BLA	Pertenece al total de fichas amarillas.
NEG	Pertenece al total de fichas negras.

```

390 FOR N=2 TO 6:PAPER#N,2:PEN#N,3:
CLS#N:NEXT N
400 PRINT#2,"ULTIMA JUGADA";STRING$
(13,154):PRINT#3," FICHAS AMA.";STR
ING$(13,154);"DENTRO..","FUERA...","
TOTAL..."
410 PRINT#4," FICHAS NEG. ";STRING$
(13,154);"DENTRO..","FUERA...","TO
TAL...":PRINT#5," GANAN LAS ";STR
ING$(13,154):GOSUB 1020
420 '
430 REM ***** TURNOS DE JUGADA **

```

```

500 IF A$(G)="" THEN A$(G)=J$ ELSE
IF A$(G)<>J$ THEN 460
510 A(G)=A(G)+1:GOSUB 610:GOTO 430
520 IF B$(G)="" THEN B$(G)=J$ ELSE
IF B$(G)<>J$ THEN 460
530 B(G)=B(G)+1:GOSUB 610:GOTO 430
540 IF C$(G)="" THEN C$(G)=J$ ELSE
IF C$(G)<>J$ THEN 460
550 C(G)=C(G)+1:GOSUB 610:GOTO 430
560 IF D$(G)="" THEN D$(G)=J$ ELSE
IF D$(G)<>J$ THEN 460
570 D(G)=D(G)+1:GOSUB 610:GOTO 430

```

```

580 IF E$(G)=" THEN E$(G)=J$ ELSE
IF E$(G)<J$ THEN 460
590 E(G)=E(G)+1:GOSUB 610:GOTO 430
600 '
610 REM ***** OPERACIONES *****
****
620 '
630 IF J$="AMARILLAS" THEN BF=BF-1:
BD=BD+1 ELSE NF=NF-1:ND=ND+1
640 LOCATE#2,1,3:PRINT#2,STRING$(13
,32);:LOCATE#2,1,3:PRINT#2,J$;" ";Z
$;"-";Y$;
650 N3=G:N4=G:IF G=1 THEN N2=2 ELSE
IF G=2 THEN N2=6 ELSE IF G=3 THEN
N2=10 ELSE IF G=4 THEN N2=14 ELSE I
F G=5 THEN N2=18
660 GOSUB 1080
670 '
680 REM **** COMPROBAR EXPLOSIONES
****
690 '
700 IF A(1)>1 THEN A(2)=A(2)+1:A$(1
)=J$:B(1)=B(1)+1:B$(1)=J$:A(1)=A(1)
-2:GOSUB 1050:IF A(1)=0 THEN A$(1)="
"
710 IF B(1)>2 THEN A(1)=A(1)+1:A$(1
)=J$:B(2)=B(2)+1:B$(2)=J$:C(1)=C(1)
+1:C$(1)=J$:B(1)=B(1)-3:GOSUB 1050:
IF B(1)=0 THEN B$(1)="
"
720 IF C(1)>2 THEN B(1)=B(1)+1:B$(1
)=J$:C(2)=C(2)+1:C$(2)=J$:D(1)=D(1)
+1:D$(1)=J$:C(1)=C(1)-3:GOSUB 1050:
IF C(1)=0 THEN C$(1)="
"
730 IF D(1)>2 THEN C(1)=C(1)+1:C$(1
)=J$:D(2)=D(2)+1:D$(2)=J$:E(1)=E(1)
+1:E$(1)=J$:D(1)=D(1)-3:GOSUB 1050:
IF D(1)=0 THEN D$(1)="
"
740 IF E(1)>1 THEN D(1)=D(1)+1:D$(1
)=J$:E(2)=E(2)+1:E$(2)=J$:E(1)=E(1)
-2:GOSUB 1050:IF E(1)=0 THEN E$(1)="
"
750 IF A(2)>2 THEN A(1)=A(1)+1:A$(1
)=J$:A(3)=A(3)+1:A$(3)=J$:B(2)=B(2)
+1:B$(2)=J$:A(2)=A(2)-3:GOSUB 1050:
IF A(2)=0 THEN A$(2)="
"
760 IF B(2)>3 THEN A(2)=A(2)+1:A$(2
)=J$:B(1)=B(1)+1:B$(1)=J$:B(3)=B(3)
+1:B$(3)=J$:C(2)=C(2)+1:C$(2)=J$:B(
2)=B(2)-4:GOSUB 1050:IF B(2)=0 THEN
B$(2)="
"
770 IF C(2)>3 THEN B(2)=B(2)+1:B$(2
)=J$:C(1)=C(1)+1:C$(1)=J$:C(3)=C(3)
+1:C$(3)=J$:D(2)=D(2)+1:D$(2)=J$:C(
2)=C(2)-4:GOSUB 1050:IF C(2)=0 THEN
C$(2)="
"
780 IF D(2)>3 THEN C(2)=C(2)+1:C$(2

```

```

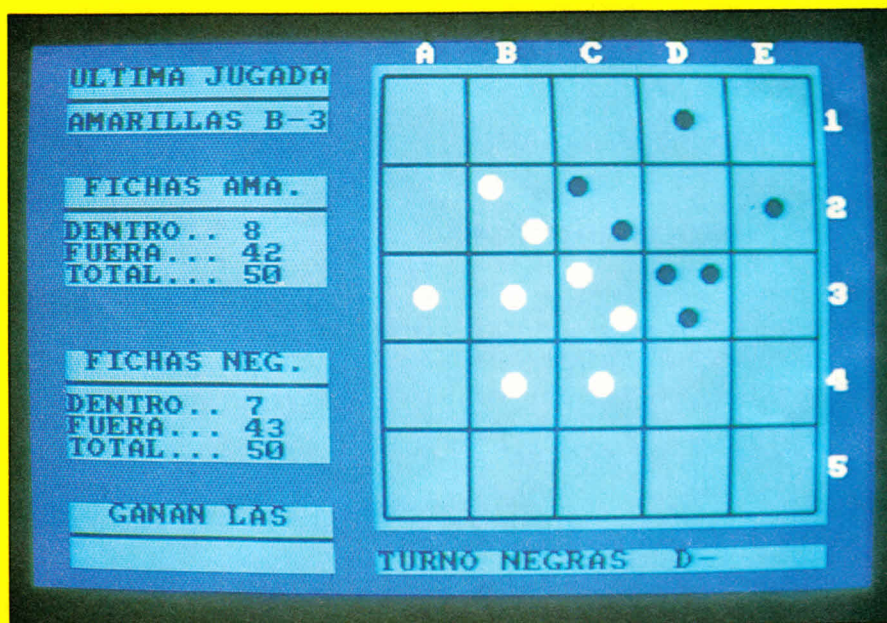
)=J$:D(1)=D(1)+1:D$(1)=J$:D(3)=D(3)
+1:D$(3)=J$:E(2)=E(2)+1:E$(2)=J$:D(
2)=D(2)-4:GOSUB 1050:IF D(2)=0 THEN
D$(2)="
"
790 IF E(2)>2 THEN D(2)=D(2)+1:D$(2
)=J$:E(1)=E(1)+1:E$(1)=J$:E(3)=E(3)
+1:E$(3)=J$:E(2)=E(2)-3:GOSUB 1050:
IF E(2)=0 THEN E$(2)="
"
800 IF A(3)>2 THEN A(2)=A(2)+1:A$(2
)=J$:A(4)=A(4)+1:A$(4)=J$:B(3)=B(3)
+1:B$(3)=J$:A(3)=A(3)-3:GOSUB 1050:
IF A(3)=0 THEN A$(3)="
"
810 IF B(3)>3 THEN A(3)=A(3)+1:A$(3
)=J$:B(2)=B(2)+1:B$(2)=J$:B(4)=B(4)
+1:B$(4)=J$:C(3)=C(3)+1:C$(3)=J$:B(
3)=B(3)-4:GOSUB 1050:IF B(3)=0 THEN
B$(3)="
"
820 IF C(3)>3 THEN B(3)=B(3)+1:B$(3
)=J$:C(2)=C(2)+1:C$(2)=J$:C(4)=C(4)
+1:C$(4)=J$:D(3)=D(3)+1:D$(3)=J$:C(

```

```

+1:B$(4)=J$:A(4)=A(4)-3:GOSUB 1050:
IF A(4)=0 THEN A$(4)="
"
860 IF B(4)>3 THEN A(4)=A(4)+1:A$(4
)=J$:B(3)=B(3)+1:B$(3)=J$:B(5)=B(5)
+1:B$(5)=J$:C(4)=C(4)+1:C$(4)=J$:B(
4)=B(4)-4:GOSUB 1050:IF B(4)=0 THEN
B$(4)="
"
870 IF C(4)>3 THEN B(4)=B(4)+1:B$(4
)=J$:C(3)=C(3)+1:C$(3)=J$:C(5)=C(5)
+1:C$(5)=J$:D(4)=D(4)+1:D$(4)=J$:C(
4)=C(4)-4:GOSUB 1050:IF C(4)=0 THEN
C$(4)="
"
880 IF D(4)>3 THEN C(4)=C(4)+1:C$(4
)=J$:D(3)=D(3)+1:D$(3)=J$:D(5)=D(5)
+1:D$(5)=J$:E(4)=E(4)+1:E$(4)=J$:D(
4)=D(4)-4:GOSUB 1050:IF D(4)=0 THEN
D$(4)="
"
890 IF E(4)>2 THEN D(4)=D(4)+1:D$(4
)=J$:E(3)=E(3)+1:E$(3)=J$:E(5)=E(5)
+1:E$(5)=J$:E(4)=E(4)-3:GOSUB 1050:

```



```

3)=C(3)-4:GOSUB 1050:IF C(3)=0 THEN
C$(3)="
"
830 IF D(3)>3 THEN C(3)=C(3)+1:C$(3
)=J$:D(2)=D(2)+1:D$(2)=J$:D(4)=D(4)
+1:D$(4)=J$:E(3)=E(3)+1:E$(3)=J$:D(
3)=D(3)-4:GOSUB 1050:IF D(3)=0 THEN
D$(3)="
"
840 IF E(3)>2 THEN D(3)=D(3)+1:D$(3
)=J$:E(2)=E(2)+1:E$(2)=J$:E(4)=E(4)
+1:E$(4)=J$:E(3)=E(3)-3:GOSUB 1050:
IF E(3)=0 THEN E$(3)="
"
850 IF A(4)>2 THEN A(3)=A(3)+1:A$(3
)=J$:A(5)=A(5)+1:A$(5)=J$:B(4)=B(4)

```

```

IF E(4)=0 THEN E$(4)="
"
900 IF A(5)>1 THEN A(4)=A(4)+1:A$(4
)=J$:B(5)=B(5)+1:B$(5)=J$:A(5)=A(5)
-2:GOSUB 1050:IF A(5)=0 THEN A$(5)="
"
910 IF B(5)>2 THEN A(5)=A(5)+1:A$(5
)=J$:B(4)=B(4)+1:B$(4)=J$:C(5)=C(5)
+1:C$(5)=J$:B(5)=B(5)-3:GOSUB 1050:
IF B(5)=0 THEN B$(5)="
"
920 IF C(5)>2 THEN B(5)=B(5)+1:B$(5
)=J$:C(4)=C(4)+1:C$(4)=J$:D(5)=D(5)
+1:D$(5)=J$:C(5)=C(5)-3:GOSUB 1050:
IF C(5)=0 THEN C$(5)="
"

```

```

930 IF D(5)>2 THEN C(5)=C(5)+1:C(5)
)=J$:D(4)=D(4)+1:D(4)=J$:E(5)=E(5)
+1:E(5)=J$:D(5)=D(5)-3:GOSUB 1050:
IF D(5)=0 THEN D(5)=""
940 IF E(5)>1 THEN D(5)=D(5)+1:D(5)
)=J$:E(4)=E(4)+1:E(4)=J$:E(5)=E(5)
-2:GOSUB 1050:IF E(5)=0 THEN E(5)=
""
950 /
960 REM ***** COMPARACION DE (BI)
****
970 /
980 IF BI=1 THEN BI=0:GOTO 700 ELSE
RETURN
990 /
1000 REM **** CONTADORES DE FICHAS
****
1010 /
1020 BLA=BF+BD:LOCATE#3,9,3:PRINT#3
,BD;" ":LOCATE#3,9,4:PRINT#3,BF;" "
:LOCATE#3,9,5:PRINT#3,BLA;" "
1030 NEG=NF+ND:LOCATE#4,9,3:PRINT#4
,ND;" ":LOCATE#4,9,4:PRINT#4,NF;" "
:LOCATE#4,9,5:PRINT#4,NEG;" ";;:RETU
RN
1040 /
1050 REM ***** IMPRIMIR FICHAS ***
***
1060 /
1070 N2=2:N3=1:N4=5:BI=1
1080 CY=N2:FOR N=N3 TO N4:CX=2:IF A
$(N)="AMARILLAS" THEN PEN#1,1 ELSE
PEN#1,3
1090 IF A$(N)<>"" THEN GOSUB 1400 E
LSE 1120
1100 IF A(N)=1 THEN LOCATE#1,CX+1,C
Y+1:PRINT#1,CHR$(250); ELSE IF A(N)
=2 THEN LOCATE#1,CX,CY:PRINT#1,CHR$
(250);:LOCATE#1,CX+2,CY+2:PRINT#1,C
HR$(250);ELSE IF A(N)=3 THEN LOCATE
#1,CX,CY:PRINT#1,CHR$(250);" ";CHR$
(250);:LOCATE#1,CX+1,CY+2:PRINT#1,C
HR$(250);
1110 IF A(N)=4 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX,CY+2:PRINT#1,CHR$(250);"
";CHR$(250);
1120 CX=CX+4:IF B$(N)<>"" THEN GOSU
B 1480 ELSE 1170
1130 IF B$(N)="AMARILLAS"THEN PEN#1
,1 ELSE PEN#1,3
1140 IF B(N)=1 THEN LOCATE#1,CX+1,C
Y+1:PRINT#1,CHR$(250);ELSE IF B(N)=
2 THEN LOCATE#1,CX,CY:PRINT#1,CHR$(
250);:LOCATE#1,CX+2,CY+2:PRINT#1,CH
R$(250); ELSE IF B(N)=3 THEN LOCATE

```

```

#1,CX,CY:PRINT#1,CHR$(250);" ";CHR$
(250);:LOCATE#1,CX+1,CY+2:PRINT#1,C
HR$(250);
1150 IF B(N)=4 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX,CY+2:PRINT#1,CHR$(250);"
";CHR$(250);
1160 IF B(N)=5 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX+1,CY+1:PRINT#1,CHR$(250);
:LOCATE#1,CX,CY+2:PRINT#1,CHR$(250)
;" ";CHR$(250)
1170 CX=CX+4:IF C$(N)<>"" THEN GOSU
B 1480 ELSE 1220
1180 IF C$(N)="AMARILLAS"THEN PEN#1
,1 ELSE PEN#1,3
1190 IF C(N)=1 THEN LOCATE#1,CX+1,C
Y+1:PRINT#1,CHR$(250); ELSE IF C(N)
=2 THEN LOCATE#1,CX,CY:PRINT#1,CHR$
(250);:LOCATE#1,CX+2,CY+2:PRINT#1,C
HR$(250)
1200 IF C(N)=3 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX+1,CY+2:PRINT#1,CHR$(250);
ELSE IF C(N)=4 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX,CY+2:PRINT#1,CHR$(250);"
";CHR$(250);
1210 IF C(N)=5 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250):LOC
ATE#1,CX+1,CY+1:PRINT#1,CHR$(250);:
LOCATE#1,CX,CY+2:PRINT#1,CHR$(250);
" ";CHR$(250);
1220 CX=CX+4:IF D$(N)<>"" THEN GOSU
B 1480 ELSE 1270
1230 IF D$(N)="AMARILLAS"THEN PEN#1
,1 ELSE PEN#1,3
1240 IF D(N)=1 THEN LOCATE#1,CX+1,C
Y+1:PRINT#1,CHR$(250); ELSE IF D(N)
=2 THEN LOCATE#1,CX,CY:PRINT#1,CHR$
(250);:LOCATE#1,CX+2,CY+2:PRINT#1,C
HR$(250);
1250 IF D(N)=3 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX+1,CY+2:PRINT#1,CHR$(250);
ELSE IF D(N)=4 THEN LOCATE#1,CX,CY
:PRINT#1,CHR$(250);" ";CHR$(250);:L
OCATE#1,CX,CY+2:PRINT#1,CHR$(250);"
";CHR$(250);
1260 IF D(N)=5 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX+1,CY+1:PRINT#1,CHR$(250);
:LOCATE#1,CX,CY+2:PRINT#1,CHR$(250)
;" ";CHR$(250);
1270 CX=CX+4:IF E$(N)<>"" THEN GOSU
B 1480 ELSE 1320

```

serie ORO

```

1280 IF E$(N)="AMARILLAS" THEN PEN#
1,1 ELSE PEN#1,3
1290 IF E(N)=1 THEN LOCATE#1,CX+1,C
Y+1:PRINT#1,CHR$(250); ELSE IF E(N)
=2 THEN LOCATE#1,CX,CY:PRINT#1,CHR$
(250);:LOCATE#1,CX+2,CY+2:PRINT#1,C
HR$(250);
1300 IF E(N)=3 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX+1,CY+2:PRINT#1,CHR$(250);
ELSE IF E(N)=4 THEN LOCATE#1,CX,CY
:PRINT#1,CHR$(250);" ";CHR$(250);:L
OCATE#1,CX,CY+2:PRINT#1,CHR$(250);"
";CHR$(250);
1310 IF E(N)=5 THEN LOCATE#1,CX,CY:
PRINT#1,CHR$(250);" ";CHR$(250);:LO
CATE#1,CX+1,CY+1:PRINT#1,CHR$(250);
:LOCATE#1,CX,CY+2:PRINT#1,CHR$(250)
;" ";CHR$(250);
1320 CY=CY+4:NEXT N
1330 /
1340 REM ***** CUENTA FICHAS DESPUE
S DE CADA JUGADA *****
1350 /
1360 BD=0:ND=0:FOR N=1 TO 5
1370 IF A$(N)<>"" THEN IF A$(N)="NE
GRAS" THEN ND=ND+A(N) ELSE BD=BD+A(
N)
1380 IF B$(N)<>"" THEN IF B$(N)="NE
GRAS" THEN ND=ND+B(N) ELSE BD=BD+B(
N)
1390 IF C$(N)<>"" THEN IF C$(N)="NE
GRAS" THEN ND=ND+C(N) ELSE BD=BD+C(
N)
1400 IF D$(N)<>"" THEN IF D$(N)="NE
GRAS" THEN ND=ND+D(N) ELSE BD=BD+D(
N)
1410 IF E$(N)<>"" THEN IF E$(N)="NE
GRAS" THEN ND=ND+E(N) ELSE BD=BD+E(
N)
1420 NEXT N
1430 GOSUB 1000
1440 BI=BI+1:IF BI>1 THEN IF ND=0 O
R BD=0 THEN LOCATE#5,L1,3:PRINT#5,J
$:FOR N=1 TO 2500:NEXT N:GOTO 1520
1450 IF J$="NEGRAS"THEN L1=4 ELSE L
1=3
1460 RETURN

```

```

1470 '
1480 REM ***** BORRAR FICHAS ***
****
1490 '
1500 FOR M=0 TO 2:LOCATE#1,CX,CY+M:
PRINT#1," ";NEXT M:RETURN
1510 '

```

```

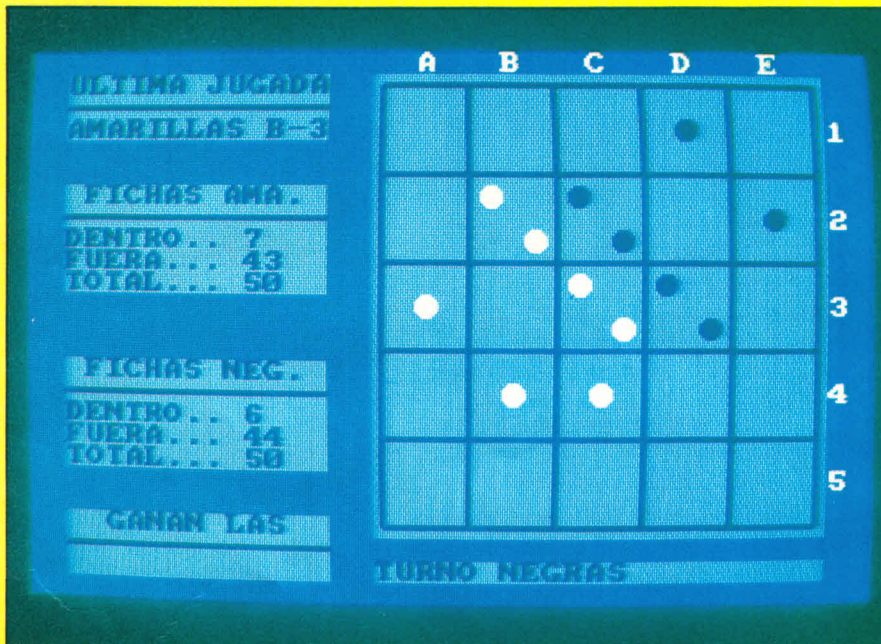
:PRINT TAB(5);STRING$(A,143);" ";
PEN 1:PRINT STRING$(A-1,143);" ";
PEN 2:PRINT STRING$(A-2,143);" ";
PEN 0:PRINT STRING$(A-3,143);" ";
PEN 1:PRINT STRING$(A-4,143);" ";
PEN 2:PRINT STRING$(A-5,143):NEXT:P
RINT:PRINT

```

```

1710 GOSUB 1840
1720 PRINT " EXPLICAREMOS DOS CO
NCEPTOS BASICOS:CAPACIDAD Y EXPLO
SION.TODAS LAS CASILLASTIENEN UNA CA
PACIDAD MAXIMA QUE ES IGUALAL NUMER
O DE CASILLAS ADYACENTES."
1730 PRINT:PRINT:PRINT " SI UN J
UGADOR COLOCA UNA FICHA EN UNA CA
SILLA Y ESTA ALCANZA SU CAPACIDAD M
AXIMA ENTONCES EXPLOTARA."
1740 GOSUB 1840

```



```

1750 PRINT " LA EXPLOSION SE EJEC
UTARA COLOCANDOCADA UNA DE LAS
FICHAS DE LA CASILLAEXPLOTADA EN
CADA UNA DE LAS CASILLASADYACENTE
S."

```

```

1760 PRINT:PRINT:PRINT " SI ALGU
NO DE LOS CUADROS ADYACENTES ESTA
OCUPADO POR FICHAS ENEMIGAS, ESTAS
SON INMEDIATAMENTE CAPTURADAS."
1770 GOSUB 1840

```

```

1780 PRINT " EL TURNO DEL JUGAD
OR TERMINA CUANDO NO HAY CASILLAS QU
E DEBAN SER EXPLOTADAS"

```

```

1790 PRINT:PRINT:PRINT " PUEDE OC
URRIR QUE COMO RESULTADO DEUNA REA
CCION EN CADENA, LA CASILLA PASEDE
CAPACIDAD; CUANDO LA EXPLOSION TE
NGALUGAR SE EJECUTARA EN LA FORMA H
ABITUAL,Y LA FICHA O FICHAS SOBRANT
ES SE DEJARANEN DICHA CASILLA."
1800 GOSUB 1840

```

```

1810 PRINT " EL OBJETIVO ES DEJAR
SIN FICHAS EN ELTABLERO AL ADVERSA
RIO. EL QUE LO LOGREHABRA GANADO L
A PARTIDA."

```

```

1820 PRINT:PRINT:PRINT " CONVIENE
SABER,POR ULTIMO, QUE CUANDO SE TIEN
EN VARIAS CASILLAS PARA EXPLOTARA
LA VEZ, NO IMPORTA EN QUE ORDEN
SEEFECTUEN LAS EXPLOSIONES, EL R
ESULTADOES EL MISMO."

```

```

1830 RETURN
1840 CALL &BB03:PRINT:PRINT:PRINT:P
RINT " PULSE UNA TECLA PARA CONTI
NUAR"

```

```

1850 CALL &BB18:CLS#0:RETURN ELSE 1
850

```

```

1520 REM ***** IMPRIME GANADOR FIN
****
1530 '
1540 INK 1,3:INK 2,9:INK 3,13:PAPER
3:CLS:BORDER 0
1550 A=6:PRINT:FOR N=1 TO 5
1560 PEN 2:PRINT TAB(5);STRING$(A,1
43);" ";PEN 1:PRINT STRING$(A-1,1
43);" ";PEN 0:PRINT STRING$(A-2,1
43);" ";PEN 2:PRINT STRING$(A-3,1
43);" ";PEN 1:PRINT STRING$(A-4,1
43);" ";PEN 0:PRINT STRING$(A-5,1
43)
1570 NEXT N
1580 PLOT 176,272:DRAW# 288,0:DRAW#
0,-144:DRAW# -288,0:DRAW# 0,144:DR
AW# 8,-8:DRAW# 272,0:DRAW# 0,-128:D
RAW# -272,0:DRAW# 0,128:PLOT 464,27
2:DRAW# -8,-8:PLOT 464,128:DRAW# -8
,0:PLOT 176,128:DRAW# 8,8
1590 LOCATE 16,11:PEN 0:PRINT"G A N
A N":LOCATE 18,13:PRINT"L A S":IF
J$="NEGRAS" THEN LOCATE 15,15:PRINT
"N E G R A S" ELSE LOCATE 13,15:PRI
NT"A M A R I L L A S"
1600 LOCATE 1,20:FOR N=1 TO 2:PEN 0

```

```

1610 PEN 0:PRINT TAB(7);"QUIERES JU
GAR DE NUEVO (S/N) ";
1620 X$=INKEY$:X$=UPPER$(X$):IF X$=
"S" THEN RUN ELSE IF X$<>"N" THEN 1
620
1630 CALL 0
1640 '
1650 REM ***** INSTRUCCIONES ***
****
1660 '
1670 MODE 1:WINDOW#0,1,40,10,25:WIN
DOW#7,11,31,2,7:PEN#7,3:PAPER#7,2:C
LS#7:LOCATE#7,4,2:PRINT#7,"** EXPLO
SION **"
1680 LOCATE#7,5,5:PRINT#7,"INSTRUCC
IONES"
1690 PRINT " TRATANDO DE ELABORA
R UN MODELO QUE,REPRODUJERA Y AY
UDARA A EXPLICAR ELPROCESO DE LA
FISION NUCLEAR, SE INVENTODE MANER
A CASUAL UN JUEGO: EXPLOSION
1700 PRINT:PRINT:PRINT " AL INIC
IO, EL TABLERO ESTA VACIO Y,CADA J
UGADOR EN SU TURNO COLOCA UNAF
ICHA EN UNA CASILLA QUE ESTE LIB
RE OQUE CONTENGA FICHAS PROPIAS."

```



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

Un número de película

A partir del día 1 estreno en toda España

Batman, Bomb Jack, Commando...

todos tus héroes preferidos juntos en este número,
para divertirse a tope.

MICRO

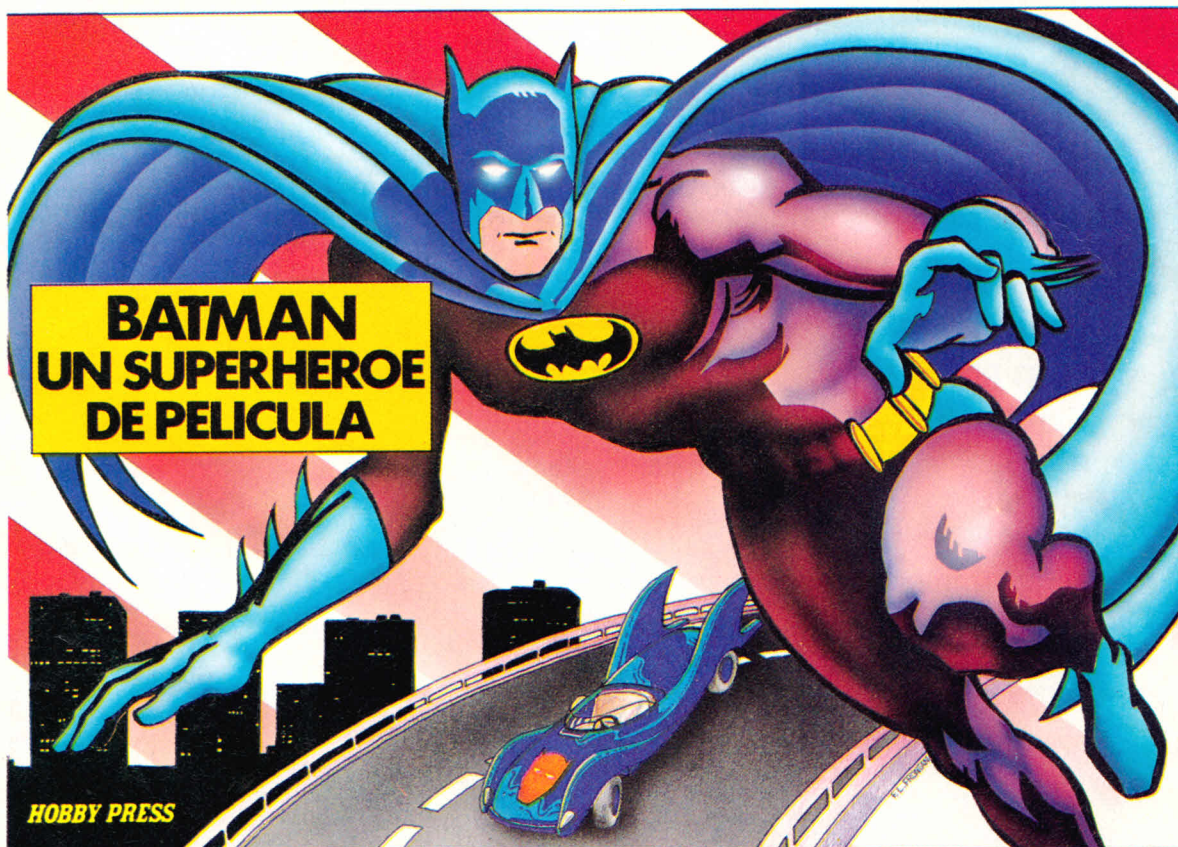
Manía

Año 11 - N° 15

Sólo para adictos

300 Ptas.

Canarias, Ceuta y Melilla 205 ptas.



AMSTRAD
SPECTRUM

ALIEN HIGHWAY
SPINDIZZY
FRANKIE GOES TO HOLLYWOOD
BOMB JACK
COMMANDO

MSX

NIGHT SHADE
y además
Gunfright
y
Starquake

SPECTRUM
MSX

AMSTRAD
COMMODORE

Utensilios y Cachivaches

UNA AUTENTICA
ORQUESTA SINFONICA

HISTORIA
DE SOFT

*Aventura
en el castillo
del Conde Drácula*

Pídelo en tu kiosco

Mercado común

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

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

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**
AMSTRAD SEMANAL.
Apartado de correos 54.062
28080 MADRID
¡ABSTENERSE PIRATAS!

MASTER HARD

Servicio Técnico
Para **AMSTRAD**
en Galicia,
León
y Asturias.
C/ Magdalena, 213
El Ferrol
Tel.: (981) 35 84 32

Desearía contactar con usuarios del **Amstrad** CPC 464, para el intercambio de ideas, información y juegos. Al ser posible en Valencia. Telf. 323 30 41. Preguntar por Mariano.

Vendo CPC 464 con monitor en color y unidad de disco DD1 todo en perfectas condiciones, además joystick quickshot IV y cable para impresora, disquetes con CPM2.2, Logo, Pascal 80, Fortran 80, M-Basic (intérprete y compilador), Micropen, Amsfile, Amsword, Randon y Randon-F, etc. (con sus manuales). Más de cien juegos (Copiones, Cyrus, Tascopy, Fist,

etc.). Todo por 100.000 pesetas. Interesados llamar al tel. 468 10 88 por la noche y preguntar por Manuel. Madrid.

Vendo dos programas originales en disco, (Sorcery y 3D Boxing) compatible **Amstrad** CPC 464/664/6128 por 2.200 pesetas cada uno ó 4.000 los dos. Interesados escribir a: Ernesto García Ortega. Chalet la Rosaleda n.º 20 Sigüenza (Guadalajara).

Vendo impresora IBM Graphics Printer, bidireccional de 80 c.p.c. Compatible Epson en gráficos. Utilizable con cualquier salida Centronics (**Amstrad**, MSX, IBM, etc.). De 40 a 132 columnas, con 48 tipos distintos de impresión. Está nueva, tiene sólo cuatro meses de uso. Garantía. Precio nueva 91.000. Vendo por 65.000 discutibles, y regalo 500 hojas de papel continuo. Interesados, llamar a Claudio o Diego, al (91) 742 28 37.

Vendo Amstrad CPC 464 con monitor de color, en perfecto estado, libros y manuales del mismo, incluyo algunos programas originales, todo por el precio de 72.000 pesetas. Interesados escribir a José Miguel Herrero, C/ C. Matas n.º 1, Fuentes de Ebro (Zaragoza), o llamar al tel. (976) 16 01 77.

Desearía contactar con propietarios de **Amstrad** CPC 464/472 que residen en Alicante (ciudad) para intercambio de juegos. Llamar al 24 11 26, preguntar por José Javier.

Vendo ordenador Atari 800 XL con datacorder, dos joysticks con mando a distancia y conexiones para periféricos. Todo a estrenar y con garantía. Además regalo curso de iniciación al Basic y dos

juegos Atari. Precio a convenir. Mi nombre es Antonio. C/ Basilio Sala 15-4.º en San Juan de Alicante.

Cambio monitor fósforo verde GT 64 **Amstrad** y alimentador modular MP-2 (para **Amstrad** 646 y 664), por monitor de color GTM 640 **Amstrad**. Monitor y modulador sólo tienen 4 meses. Interesados llamar al tel. 269 34 11. Preguntar por Javier Morera. Madrid.

Vendo programas a 500 y 800 pesetas cada uno. Entre ellos están: 3D Voice Chess, Raid!!!, Night Shade, 3D Grand Prix, Decathlon, Alien, Sorcery, Bead Head, Jet Set Willy, Knight Lore. Y muchos más, interesados llamar al tel. (958) 60 14 81.

Vendo juegos HOUSE OF USHER y DEFEND OR DIE, en cinta, completamente nuevos (sin estrenar) más una revista de programación para **Amstrad**, todo por 3.000 pts. Interesados llamar al Tel. 84 52 51 de Pontevedra de las 19,00 a las 22,00. Preguntar por Moisés.

MASTER COMPUTER

Centro Comercial Guadalupe
Ctra. Canillas, 136-1.ª planta
Tel. 200 80 65 **MADRID**

Centro Comercial El Bulevar
La Moraleja
Tel. 654 16 12 **MADRID**

También abierto domingos de 10 a 2
Centro Comercial Ciudad Sto. Domingo
Ctra. de Burgos, km 28.
Tel.: (91) 622 12 89
Algete (Madrid) **(Central)**

Desearía hacer intercambios compras—ventas de toda clase de programas; tanto en disco como en cinta, dispongo de muchas novedades; quien esté interesado escribir a Vicente Luengo, 19. Aligent (Valencia). O llamar al Tel. (96) 175 05 53, tendréis respuesta asegurada.

En
Septiembre
comienza
un
nuevo
hobby

Ya
está
a
la
venta
el
nº4.
Pídelo
en
tu
kiosco

hobby

PARA HACER Y CONOCER

250 Ptas.

AVIONES
TELEDIRIGIDOS,
LA OTRA FORMA
DE VOLAR

Robots domésticos,
el futuro cada vez
más cerca

Aprende a estudiar

Cómo hacer
fotografías en relieve

FABRICA TU PROPIO PAPEL

HOBBY PRESS

CONCURSO
GANA UN
EQUIPO DE MUSICA
¡ES MUY FACIL!

Canarias, Ceuta y Melilla, 240 ptas.

AMSTRAD DMP 2000

NO ENCONTRARA UNA IMPRESORA QUE LE HAGA TAN BUEN PAPEL.



Soportes abatibles que permiten colocar el papel bajo la impresora.

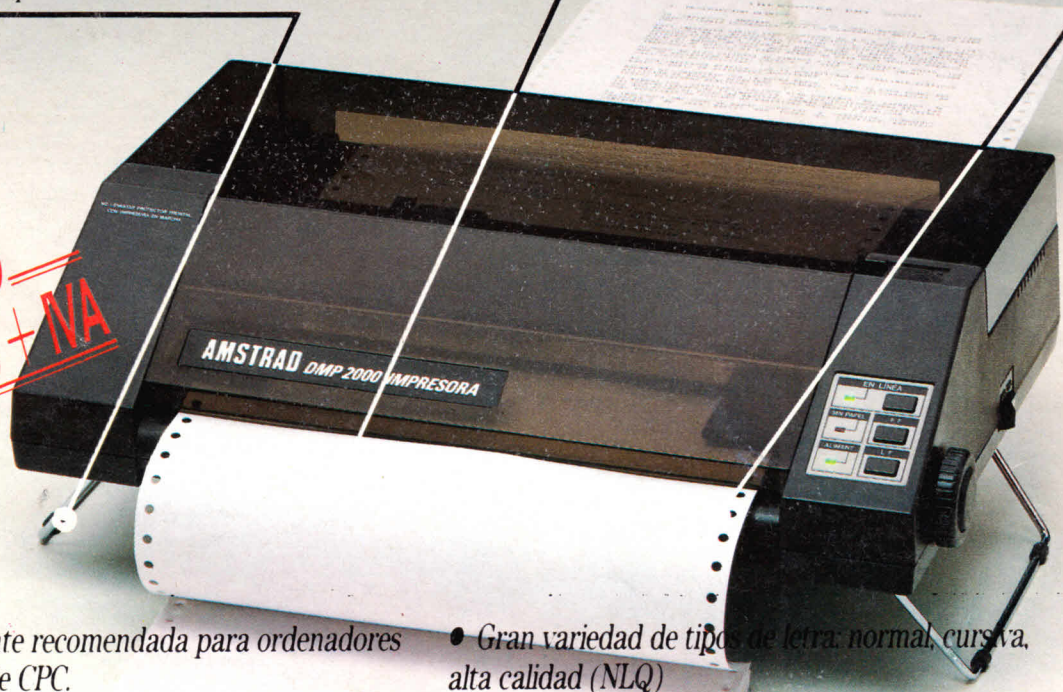


Cómodo sistema de carga frontal del papel.



Admite diferentes anchos de papel, tanto continuo (de 114 a 254 mm.) como hojas sueltas (102 a 241 mm.)

**POR SOLO
39.500 PTAS + IVA**



- Especialmente recomendada para ordenadores AMSTRAD serie CPC.
- Conectable a cualquier ordenador con interface centronics.
- Velocidad de impresión de 105 caracteres por segundo.

- Gran variedad de tipos de letra: normal, cursiva, alta calidad (NLQ)
- 40, 66, 80 y 132 caracteres por columna.
- Impresión de gráficos punto a punto en diferentes densidades.
- 96 caracteres ASCII y 8 sub juegos internacionales.

¡¡ Increíble !!

AMSTRAD ESPAÑA

GRUPO INDESCOMP