

MICROHOBBY

AÑO I N.º 14

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO I N.º 14

150 Ptas.

Canarios 160 ptas.

FILER:
crea tus archivos
de datos en
cinta/disco

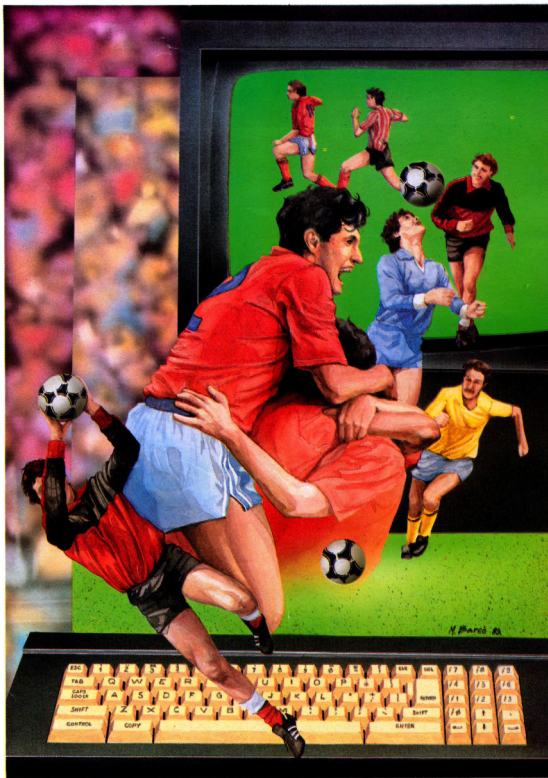
¡Quinielas!
Hazte millonario
con tu Amstrad:
con quinielón
podrás acertar
hasta 14
resultados

**OPERACIONES
MATEMATICAS
EN EL AMSTRAD**

**RUTINAS
DE SPRITES
E IMPRESION
DE CARACTERES
EN PANTALLA**

SOFTWARE

DYNAMITE DAN,
primer juego comercial
en España con pokes
de vidas infinitas



HOBBY PRESS, S.A.

UN "MILLON" EN JUEGO

Gráficos a color, rapidez de acción y sonido dan a este juego la sensación de una máquina de millón real.

PVP
1.900 (cinta)
2.900 (disco)



464
664
6128
AMSTRAD

ACE

ACTIVIDADES COMERCIALES ELECTRONICAS, S.A.
Tarragona, 110. Tel. 325 10 56* 08015 Barcelona. Telex 93133 ACEEE

YA DISPONIBLE EN



Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

AMSTRAD

sumario

Año I • Número 14 • 3 al 9 de Diciembre de 1985
150 ptas. (sobretasa Canarias, 10 ptas.)

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

Victor Prieto

Subdirector

José María Díaz

Redactora Jefe

Marta García

Diseño

José Flores

Colaboradores

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Juan J. Martínez

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Javier Martínez

Portada

Manuel Barco

Ilustradores

J. Igual, J. Pons, F. L. Frontán,

J. Sepián, Pejo, J. J. Mora,

Luigi Pérez

Edita

HOBBY PRESS S.A.

Presidente

Maria Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortes

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

Secretaría de Dirección

Marisa Cogorro

Subcripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

La Granja, s/n

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDIC, S. A. Crta. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Navocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GRUF

Ezequiel Solana, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos

de la revista

COMPUTING with the AMSTRAD

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud

América 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidario de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

8 Primeros pasos

Además de las cadenas alfanuméricas, de indubitable utilidad, es necesario conocer como se manejan los números y los cálculos relacionados con ellos en el **Amstrad**. Primeros pasos estudia el tema y propone el atento examen de un interesante programa de demostración.

Serie oro 12

El programa quinielas que presentamos en este número muy bien podría calificarse de excepcional. Nos permite hacer todo tipo de apuestas desarrollando las distintas combinaciones según los métodos considerados mejores por la «teoría» quinielística.

Las probabilidades de acertar 14 son pocas, pero tal vez el programa sirva de ayuda a alguien y le permita investigar la estrategia más conveniente.

Mr Joystick 18

¿Conseguirá el superagente Dan dinamita apoderarse de los planos del superdestructoro Rayo Devastador?



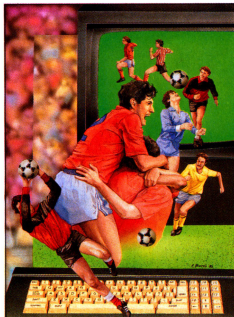
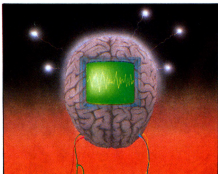
20 ProgramAcción

Nunca está de más mejorar la apariencia estética de nuestros programas, añadiéndoles belleza y originalidad.

En este sentido presentamos dos nuevos comandos del **Amstrad Basic**, creados haciendo uso del sistema **RSX**, que permiten girar caracteres a derecha e izquierda de 90 en 90 grados.

Código máquina 28

Damos un paso de gigante en el estudio de los gráficos y, por tanto, de la forma en la que el **Amstrad** organiza la memoria de pantalla. También, las primeras rutinas de impresión de gráficos en código máquina.



Análisis 33

Un pequeño programa muy bien construido nos muestra la técnica de los ficheros de texto en cinta/disco. Como escribir, leer en ellos, abrirlos y cerrarlos.

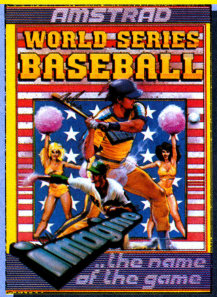
¡NUEVO!

SIEMPRE LOS PRIMEROS EN TENER LO ULTIMO

circulo de soft

MICROAMIGO S.A.

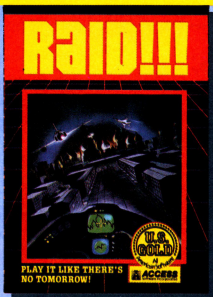
P.º de la Castellana, 268, 3.º C. 28046-MADRID.
Tel.: (91) 733 25 00



BASEBALL

Impresionante simulación en 3' dimensiones. Se puede competir contra el ordenador o contra otro jugador. No es necesario conocer el beisbol. Hay un modo de demostración. Pantallas gigantes para ver de cerca la acción.

P.V.P.: 2.100 ptas.
Precio C. de Soft: 1.890 ptas.



RAID

Defiéndete con tu escuadrilla de aviones del ataque nuclear que han lanzado sobre ti. Tu viaje estará lleno de peligros hasta que llegues a las bases de lanzamiento de misiles enemigos. Tendrás que destruirlos para salvar a tu país de una catástrofe nuclear. Gráficos y acción sensacionales.

P.V.P.: 2.300 ptas.
Precio C. de Soft: 2.070 ptas.



MATCH DAY

¡Ahora para Amstrad! No se trata de un juego de fútbol cualquiera. Fantástica acción en 3 dimensiones y animación total que dan vida al fútbol. Quedarás maravillado con el control del balón y desarrollarás tu destreza y técnica jugando contra otro jugador o contra el ordenador.

P.V.P.: 2.300 ptas.
Precio C. de Soft: 2.070 ptas.

¡¡¡...Y LOS TRES PROGRAMAS POR SOLO 5.400 PTAS!!!

¡HAZTE HOY MISMO SOCIO DEL CIRCULO DE SOFT! Además de poder adquirir tus programas al mejor precio, recibirás información de forma periódica y gratuita, del mejor software que aparezca en el mercado.

¿QUE HAY QUE HACER PARA SER SOCIO DEL CIRCULO DE SOFT? Así de fácil: envíanos por correo tu nombre, dirección y modelo de ordenador, o bien, pide por teléfono o por correo tu primer programa. ¡Y entrarás a formar parte del CIRCULO DE SOFT de forma inmediata!

Si, quiero ser SOCIO desde hoy mismo del CIRCULO DE SOFT y recibir periódicamente información de novedades de software, así como beneficiarme desde hoy mismo de los precios reducidos reservados a los SOCIOS y de sus Ofertas Especiales. El ser SOCIO no me obliga a compra alguna.

Si prefieres formalizar tu compra por teléfono puedes hacerlo llamando al (91) 733 25 00. ¡NI SE COBRAN LOS GASTOS DE ENVIO POR CORREO!!

TITULO _____	P.V.P. _____	ORDENADOR _____
_____	_____	_____
_____	_____	_____
<input type="checkbox"/> Contrarrembolso	<input type="checkbox"/> Giro Postal	<input type="checkbox"/> Telón adjunto a Microamigo, S.A.
<input type="checkbox"/> Tarjeta VISA n.º _____	Fecha caducidad _____	
Nombre _____	Apellidos _____	Edad _____
Domicilio _____		Teléfono _____
Localidad _____	C.P. _____	Provincia _____



UNIX PARA EL QL

Este sistema operativo es, por definición, multiusuario y multiárea, posee una estructura jerárquica de ficheros en disco, con subdirectorios, y un sistema de movimiento de datos a través de «pipelines», pudiendo redirigir la salida de un programa a otro o a cualquier dispositivo físico del ordenador, de una forma muy sencilla.

Los lectores se percatarán de que estamos describiendo un microUnix. Su versatilidad y potencia son enormes, siendo además compatible con el Unix Standard a nivel de código fuente del lenguaje C.

El QL ve incrementada su ROM a 144 Kbytes por la adición del OS9.

La casa británica inglesa Cumana, conocida entre otras cosas por las unidades de disco que fabrica, ha desarrollado para el Sinclair QL una versión de sistema operativo OS9, hasta ahora implementado solamente en los ordenadores Dragon dentro del campo de los micros domésticos.

Primera plana



ORDENADORES VIVOS

Emparado con los ordenadores de hace 10 años, el Amstrad que usted tiene en su casa les da «mil vueltas», porque es mucho más potente, pequeño y barato.

Estas mejoras se deben a muchos factores, naturalmente, resumidos todos ellos en el tópico «avance de la tecnología».

Sin embargo, dentro de todo este cúmulo, existe uno de importancia capital, responsable en gran parte de la presencia cada vez más amplia del ordenador en nuestra vida: la miniaturización.

Resulta que cuanto más pequeño es un chip, más barato resulta, y la máquina que lo incorpora deviene más potente por el simple hecho de que le caben más.

Estas razones tienen el suficiente peso económico para que se investigue sin cesar en cómo fabricar chips cada vez más pequeños, con una mayor escala de integración, que se dice.

Tanto es así, que en un futuro cercano se alcanzará el límite. Simplemente no podrán fabricarse chips más pequeños (aunque en ese momento los de un mega serán cosa corriente).

No es de extrañar que la gente esté empezando a pensar en materiales e incluso tecnologías alternativas a la electrónica. Las opciones son

muchas, pero tal vez la más atractiva y revolucionaria sería construir ordenadores basados en material biológico, en el carbono en lugar del silicio.

Si esto pudiera conseguirse, en el mismo espacio físico la capacidad de proceso y almacenamiento se vería multiplicada por un factor de miles de veces, ya que los «transmisores» de información, serían las propias moléculas orgánicas, muchísimo más pequeñas que cualquier chip.

Y todavía hay más: un importante sector de la comunidad científica que investiga el tema ve la llegada de los ordenadores biológicos como un paso más de la evolución, que hasta ahora culminaba en nosotros. Incluso se plantean la posibilidad de la unión directa, la simbiosis, hombre ordenador.

De momento, todas estas inquietantes y atractivas posibilidades entran dentro del campo de la ciencia ficción (¿a Dios gracias?), pero nombres como IBM, Hitachi y una gran cantidad de universidades con los mejores cerebros del mundo van a hacer todo lo posible para convertirlas en realidad en el mínimo tiempo posible.

¿Dejarán los ordenadores de ser simplemente una herramienta estúpida para llegar a ser parte de nosotros mismos? ¿Para mejorarnos?

¿Cuánto tiempo la palabra «humano» se ajustará a nuestros estándares de hoy?

NUEVO MODELO DE IBM

IBM ha lanzado una configuración superior de su último modelo, el IBM AT, el ordenador personal más potente del mundo (según IBM).

La nueva configuración posee 512 K de RAM, un Floppy de 1,2 megabytes y un disco duro de 30 millones de caracteres de almacenamiento.

Al parecer, una de las razones de la aparición del nuevo modelo se debe a la próxima aparición, valga la redundancia, de otra máquina basada en el 80386 de Intel, el no va más de los procesadores actuales.

Y la otra razón, dicha en temerosos susurros, es que el disco de 20 millones del AT tenía un porcentaje de fallos anormalmente alto. IBM nunca reconoció este hecho, pero en USA lanzó un programa de alcance nacional para identificar y sustituir las placas de los controladores de disco sospechosas.

ROBOTS ESCULTORES

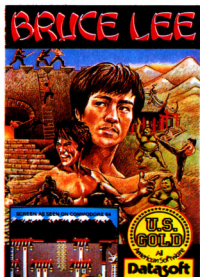
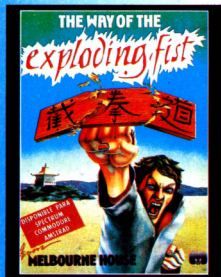
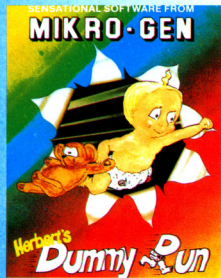
Renault está utilizando en Francia brazos robot controlados por ordenador para esculpir en plástico los modelos de los prototipos de sus coches. El tiempo que toma este proceso se ha reducido a 6 semanas, mediante la ayuda de este completo sistema CAD.

El modelo se esculpe aprovechando el output de otro programa que dibuja el coche en dos dimensiones, el cual se traslada a tres dimensiones automáticamente.



PREPA

Dis



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

RA Las MALETAS A

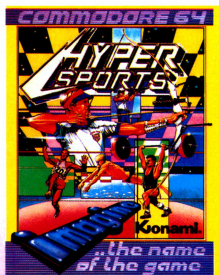
ney World!

ERBE

Software

TE INVITA

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



NUMEROS

Hemos comentado en esta sección más de una vez que los ordenadores manejan exclusivamente números, al menos internamente.

El Amstrad no es una excepción y, por ello, vamos a revisar un poco cómo se tratan los números y qué puede hacerse con ellos.

Javier Iguar



Lo más básico es comenzar con las cuatro operaciones matemáticas básicas: suma, resta, multiplicación y división.

Naturalmente que sabemos que todos dominan la aritmética sencilla, todos... menos el **Amstrad**, y ello puede crearnos algunos serios problemas.

Por ejemplo:

PRINT 5+3

obviamente imprimirá un hermoso 8 en la pantalla, pero:

PRINT 5+3*8

que resultará, 64 ó 29?

La ambigüedad de las expresiones

Como podéis observar, estamos en dificultades. Una persona ajena que leyera la expresión anterior dudaría sobre la intención del que la escribió, en el sentido de que se pretenda conseguir un resultado u otro.

Los ordenadores no pueden dudar, porque si lo hacen, se bloquean y dejan de funcionar. Para una máquina una cosa es 0 ó 1, cierta o falsa.

Esto implica la existencia de una serie de reglas incorporadas al Basic para poder manipular expresiones numéricas de una forma unívoca, sin lugar a ambigüedades.

Su nombre: Regla de la Precedencia de Operadores.

Esto quiere decir, sencillamente, que por acuerdo asignamos un orden de ejecución a los diferentes símbolos que operan sobre los números, por ejemplo, si la máquina se encuentra un signo «+» y un signo «*», podríamos decidir que primero se evalúen los miembros de la expresión que lo contengan; en el caso anterior de:

PRINT 5+3*8

el resultado sería 29. Si hubiéramos decidido que la suma tiene mayor prioridad, obtendríamos 64, un resultado tan diferente que la necesidad de unas reglas se hace patente.

La prioridad es la siguiente, de mayor a menor:

- Potenciación
- Multiplicación y división
- Suma y resta

Las operaciones que poseen la misma prioridad se evalúan de izquierda a derecha, por lo que el resultado «correcto» del ejemplo anterior es 29, no 64.

Bien, ¿y si yo quiero que la operación se realice justo de la forma «errónea»?

Todos los lenguajes de programación han previsto que en un momento dado interesa saltarse las reglas, y el método escogido para indicárselo al ordenador es casi siempre el mismo, derivado de la forma matemática normal de operar.

Gracias a los paréntesis

Si, es como te enseñaron en el colegio, mediante el uso de paréntesis.

Obtendremos 64 si escribimos:

PRINT (5+3)*8

Esta es la regla general para las 4 operaciones básicas, pero en el caso de la multiplicación y división el asunto está un poco menos claro debido a la propiedad conmutativa de la multiplicación.

En efecto, MATEMATICAMENTE es lo mismo calcular:

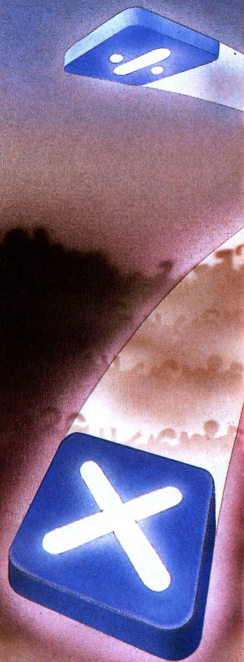
PRINT 5*8/5

que

PRINT (5*8)/5

o bien

PRINT 5*(8/5)



Primeros pasos

aunque aún en este caso, las reglas de precedencia se siguen al pie de la letra. Sin embargo, las propiedades matemáticas de los operadores le juegan una mala pasada a la claridad.

Un poco más complejo es el tema de «juguetear» con números fraccionarios. **¿Podría usted predecir los posibles resultados de:**

$$5/4 + 5 * 18/9 - 27$$

o algo parecido?

Espero que no, porque yo desde luego no puedo. Si es capaz de ello, mis más sinceras felicitaciones. Para aquellos que tenemos una mente menos rápida y un ojo menos agudo, la regla de oro es la siguiente: en la duda, utilice los paréntesis, que para eso están.

Es la única forma segura de no equivocarse y obtener resultados disparatados en su programa debido a errores que, como éstos, son difíciles de depurar.

Segunda regla de oro: deje que su ordenador haga la mayor cantidad de trabajo posible. Usted sólo siéntese y disfrute.

La exponenciación

A nivel de operaciones básicas, nos queda por hablar solamente de la exponenciación, esto es, la operación matemática que toma un número, llamado BASE y lo multiplica tantas veces por sí mismo como indique un segundo número, llamado exponente. Se representa así: A^B , en donde A es la base.

El estudio de esta operación implica interesantes posibilidades de aprender mucho más acerca del tratamiento de los números y de los límites de cálculo del **Amstrad** y de los ordenadores en general. Atención a lo que sigue.

Lo primero que uno podría plantearse es hasta dónde llega la tan careada capacidad de cálculo de los ordenadores, del **Amstrad**.

En efecto, por muy potente que sea un micro, el rango de números que puede calcular debe tener un límite, fundamentalmente porque los números, al representarse internamente en el ordenador, ocupan memoria.

Cada máquina tiene asignado un número fijo de bytes para representar un número y ahí se alcanza el techo de cálculo.

¿Cómo averiguarlo? Bien, el método de «la cuenta de la vieja» nunca falla, así que podríamos ordenarle al **Amstrad** que imprimiera 3 elevado a 1.000, por ejemplo.

Probablemente, el número astronómico que resulta sobrepasaría la capacidad de cálculo del ordenador, con lo que un oportuno mensaje de error nos forzaría a reiniciar con otro número más pequeño.

Esta manera de proceder no es precisamente muy astuta.

Tomarle el pelo al Amstrad

Nosotros, que se supone que pensamos, podemos aprovecharnos de la estupidez del micro, para que nos diga de una sola vez el rango de números que puede manejar.

Los matemáticos llevan diciendo hace mucho que un número dividido por cero es «infinito», es decir, primero, que si dividimos algo por nada, como resultado debemos obtener una cifra gordísima y segundo, al no saber exactamente cuál es, la llamamos «infinito» y listo.

Para una persona, este razonamiento resulta claro, pero para el **Amstrad** «infinito» quiere decir el máximo número que puede representar.

Por tanto, la orden:

```
PRINT 3/0
```

mostrará en pantalla el techo de cálculo del ordenador.

Este número es del orden de 10 elevado a 38, es decir, un 1 seguido de 38 ceros (*mil millones es un 1 seguido de 9 ceros*); una cifra inimaginable.

Bueno, el primer punto resuelto. El cálculo del techo inferior es igual de simple:

```
PRINT (3/0)↑-1
```

o sea, le decimos al **Amstrad** que divida un 1 entre (3/0) [«infinito»]. El resultado será «menos infinito», el número más pequeño que el ordenador puede manipular. En este caso,

```
10 REM PROGRAMA NUMERO 1
20 ON ERROR GOTO 280
30 :
40 CLS
50 overflow.error=6:ok=0:flag.error
   =0
60 :
70 WHILE NDT ok
80 :
90 GOSUB 200
100 :
110 PRINT"Calculando"
120 res=base^ex
130 :
140 GOSUB 240
150 :
160 PRINT base;"elevado a";ex;"vale
   ":"res
170 WEND
180 END
190 :
200 INPUT "Base: ",base
210 INPUT "Exponente: ",ex
220 RETURN
230 :
240 IF flag.error THEN PRINT"Ha hab
   ido un overflow":PRINT"El máximo ex
   ponente posible es:";ex
250 flag.error=0
260 RETURN
270 :
280 IF ERR<>overflow.error THEN RES
   UME NEXT
290 flag.error=-1
300 ex=ex-1
310 RESUME
320 :
```

aparece en la pantalla una cifra del orden de 10 elevado a -39.

¿Y qué pasa cuando nos salimos del rango permitido?

Fuera de rango

Entonces, el **Amstrad** nos obsequia con un lacónico mensaje de error: «overflow», que traducido al español significa «sobrepasamiento» y el programa se detiene.

Lejos de representar un inconveniente en los programas en los que usemos cálculos numéricos, podemos aprovechar este error de una forma muy conveniente, como por ejemplo en el programa número 1.

Este programa calcula el máximo exponente al cual puede elevarse un número para no salirse del rango de cálculo del **Amstrad**.

El procedimiento es muy simple, ya que sólo pueden suceder dos cosas:

- El resultado de elevar un número (*la base*) al otro (*el exponente*) cae dentro del rango de cálculo del micro. Estupendo, lo imprimimos y punto.
- El resultado cae fuera de rango: se produce un error overflow, lo detectamos mediante la sentencia 20 del programa, variamos el exponente y repetimos el intento.

Por tanto, el programa 1 consta de cuatro partes:

- bucle principal (sin fin): líneas 70-170;

2) subrutina de entrada de datos: líneas 200-220;

3) si ha ocurrido un error de overflow, se imprime un mensaje en pantalla, indicando el valor máximo posible del exponente: líneas 240-260;

4) subrutina de captura de errores: líneas 280-310.

Manejo de errores y cálculo numérico

La parte 4 del programa es el meollo del asunto; en efecto, sigamos los pasos del ordenador cuando se produce un error. La máquina salta, por efecto de la línea 20, a la rutina de la línea 280.

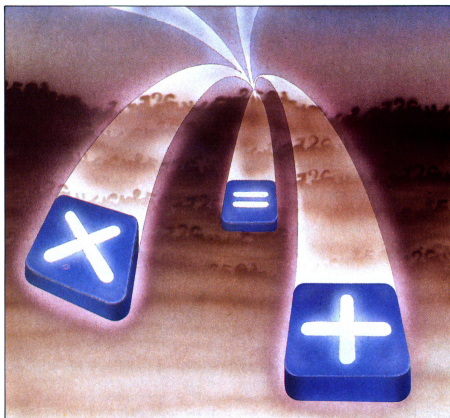
Esta línea comprueba que el valor numérico del error ocurrido sea igual al de la variable «**overflow.error**» (cada error posee un código numérico asociado; el 6 corresponde a overflow; ver capítulo 7 del manual). Si no lo es, nadie nos ha invitado a esa fiesta, por lo que optamos por ignorarlo mediante la orden:

RESUME NEXT

que fuerza al Basic a «**pasar**» de esa línea y ejecutar la siguiente. Aquí no ha pasado nada.

Si efectivamente el error detectado es el número 6, en la línea 290 la variable «**flag.error**» se pone a 1 para indicar a la rutina de la línea 240 que imprima un mensaje cuando le toque. Las líneas 300 y 310 son la auténtica clave del programa.

Cuando la rutina de error llega a esta línea, el valor de la variable «**ex**», que representa al exponente, se decreta en una unidad, y en la línea 310 la orden:



RESUME

obliga al Amstrad a continuar la ejecución EN LA MISMA LINEA DE PROGRAMA DONDE SE PRODUJO EL ERROR, es decir, en la línea 120.

Esto implica que el ordenador repetirá el cálculo de nuevo, pero con el exponente decrementado en una unidad, y lo hará una y otra vez hasta que en la línea 120 no se produzca error alguno, o sea, cuando el exponente sea el máximo al que podemos elevar la base sin salirnos del techo numérico del micro.

Interesante, ¿verdad?

Sin embargo, el programa no es perfecto. Como se observará al jugar con él un rato, con los números enteros o reales positivos todo marcha a pedir de boca, pero con los negativos al usar un exponente muy bajo el resultado final es cero.

El Amstrad se niega a considerar un error el overflow por «**debajo**». ¿Alguien podría explicar por qué?

Y, sobre todo, ¿alguien podría explicar por qué a nosotros?

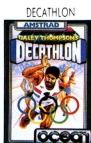
Le quedaríamos muy agradecidos.

NOVEDAD

Solamente lo mejor



2.300 ptas.



1.700 ptas.



1.900 ptas.



1.900 ptas.



1.900 ptas.



2.100 ptas.



2.100 ptas.

PIELOS POR CORREO.

COMPTIQUE

Embajadores N.º 90
Madrid 28012. Telf.: 227 09 80

QUINIELON

Quiniela siempre suena a dinero, a adilio con la suerte. Los ordenadores no pueden calcular rigurosamente todas las innumerables posibilidades de un boleto quinielístico, pero si pueden seleccionar una, en base de una serie de criterios (estrategia) elegida por el programador, y, además, hacer mucho más sencillo el tedioso protocolo del relleno de los boletos.

Así que, adelante: los 14 duermen en la RAM de tu Amstrad.

Realizado por el lector:
J. ARACIL XARRIE (Barcelona)

COMPATIBLE
CPC 464
CPC 664
CPC 6128

Como su propio nombre indica, este programa es una excelente aplicación para el desarrollo de nuestros boletos quinielísticos.

El autor señala en su extenso y detallado comentario del funcionamiento del programa que ha escogido un método, un tanto por «decreto», para clasificar lógicamente los partidos, de forma que la posibilidad de obtener un 1, X ó 2 en un determinado encuentro es distinta.

Lleno de modestia, Javier llama a su programa «cuasialeatorio»; ustedes decidirán sobre lo acertado o no de la denominación.

En el peor de los casos, lo que no se puede discutir es que el programa simplifica extraordinariamente el tedioso proceso de rellenar boletos.

DESCRIPCION

El programa permite realizar apuestas quinielísticas, de una sola columna o de ocho columnas, limitando a voluntad del usuario el número de dotes.

En el caso de una sola columna existen tres opciones. En la primera opción, una sola apuesta, se puede escoger el número de variantes. En la n.^o 2, el usuario puede fijar para cada partido si será triple o doble. En la n.^o 3, el usuario fija la cantidad de triples o dobles, asignando el ordenador los mismos, según un cálculo de probabilidades.

Para los boletos de ocho columnas, el usuario selecciona el número de fijos y el de variantes. Los fijos

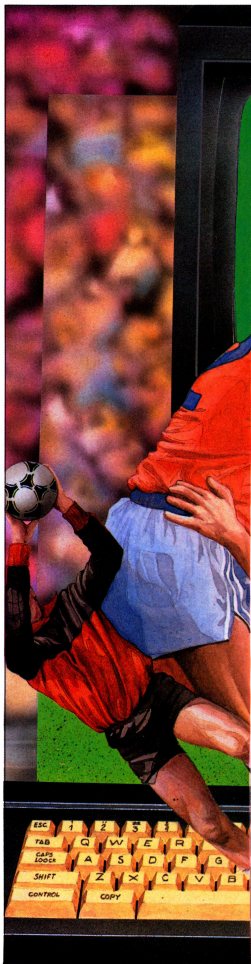
pueden ser asignados por el usuario o por el ordenador.

El programa comienza con la actualización de la clasificación de los equipos de 1.^o y 2.^o división. Dicha clasificación sirve de base para el cálculo de probabilidades de cada partido de la quiniela. Como base del cálculo se ha establecido que con una diferencia de potencial «0», entre la clasificación del equipo local y la del visitante, existe un 50 por 100 de probabilidades de que el resultado sea «1», un 30 por 100 de «X» y un 20 por 100 de «2». Estos porcentajes varían en relación a las distintas diferencias de potencial de cada partido.

El programa puedes ser considerado cuasialeatorio, debido al cálculo de probabilidades que efectúa para cada partido, teniendo en cuenta la «lógica» de la clasificación de los equipos.



Nota: el parpadeo de la pantalla indica que el programa está esperando a que se pulse una tecla.





JORNADA No. 1	No. DE EQUIPOS 1	BOLETO No. 1
No. DE TABLEROS 1		1 2 3 4 5 6 7 8
1 BARCELONA	1 BARCELONA	
2 REAL MADRID	2 REAL MADRID	
3 SEVILLA	3 SEVILLA	
4 VALLECANO	4 VALLECANO	
5 BILBAO	5 BILBAO	
6 VALLADOLID	6 VALLADOLID	
7 REAL SOCIEDAD	7 REAL SOCIEDAD	
8 CAJASTUR	8 CAJASTUR	
9 ALAVÉS	9 ALAVÉS	
10 BARCELONA	10 BARCELONA	
11 REAL MADRID	11 REAL MADRID	
12 SEVILLA	12 SEVILLA	
13 VALLECANO	13 VALLECANO	
14 BILBAO	14 BILBAO	
15 VALLADOLID	15 VALLADOLID	
16 REAL SOCIEDAD	16 REAL SOCIEDAD	
17 CAJASTUR	17 CAJASTUR	
18 ALAVÉS	18 ALAVÉS	

Serie Oro

ESTRUCTURA

- Líneas**
- 10-90 Inicio.
 - 120-220 Menú inicial.
 - 230-500 Actualización clasificación equipos de 1.ª y 2.ª
 - 510-940 Confección de la quiniela.
 - 950-1170 Menú de apuestas.
 - 1180-1480 Resguardo quiniela de una columna. Las cifras entre paréntesis indican la clasificación de cada equipo.
 - 1690-1770 Quiniela de una sola apuesta (1 columna).
 - 1780-2000 Quiniela de una sola columna eligiendo el usuario los triples y los dobles. Calcula la diferencia de potencial de ambos equipos, la cual sirve de base para acotar el campo de posibilidades de cada signo. Un número aleatorio «sort» determina, según las probabilidades de cada signo, el resultado del partido.
 - 2010-2350 Quiniela igual que la anterior, pero eligiendo el ordenador los triples y los dobles, según las distintas diferencias de potencial de los equipos.
 - 2360-2720 Requisitos boletos de 8 columnas.
 - 2730-2930 Resguardo quiniela boleto de 8 columnas.

PRINCIPALES VARIABLES

- eq&(x) Equipo antes de clasificar.
- eqc1&(x,y) Equipo después de clasificar.
- c1(x) Variable usada para clasificar.
- e1q&(x) Equipo local quiniela.
- evq&(x) Equipo visitante quiniela.
- comb&(x) Triple o doble.
- ce1q(x) Clasificación equipo local quiniela.
- ceva(x) Clasificación equipo visitante quiniela.
- uxd(x) Indica el signo del resultado.
- part(x) Indica si es triple, doble o sencillo.
- nt(x) Número tabla diferencial de potencial.
- nvar Número de variantes.
- mdos Número máximo de dosis.
- dp Diferencia de potencial.
- p1 Probabilidades del signo «1».
- pX Probabilidades del signo «X».
- sort Número aleatorio para el cálculo de probabilidades.
- nbo1 Número de boletos.
- nfi1 Número de hijos.

```

10 ***** QUIN
11 ELA *****
20 *Factor: Josep Oracil i Xarrie
30 *Setiembre 1985
40 CLS:MODE 2:PAPER 0:PEN I:CLAS1=0
  r:clas=0
50 DIM eqs(20),eqcl$(2,20),cl(20),e
  lq$(14),evq$(14),comb$(14),
  celq(20),cevc(20),uxd(
14),part(14),nt(38)
60 SOUND 1,100,20,7
70 MODE 0:LOCATE 5,12:PRINT "1 - X
  - 2"
80 GOSUB 100
90 MODE 2:GOTO 120
100 c$=INKEY$:IF c$="" THEN BORDER
  1:BORDER 5:BORDER 1:BORDER 5:BORDER
  1:
  1: BORDER 7:GOTO 100
110 c$=UPPER$(c$):RETURN

```



```

120 ***** MENU INICIAL *****
****
130 CLS:SOUND 1,100,20,7
140 LOCATE 24,8:PRINT "MENU INICIAL"
150 LOCATE 24,9:PRINT "-----"
160 LOCATE 16,12:PRINT "1.- CONFECC
  IONAR LA QUINIELA"
170 LOCATE 16,14:PRINT "2.- CARGAR
  LA QUINIELA GRABADA"
180 LOCATE 7,18:PRINT "ELECCION ? :
  "
190 GOSUB 100
200 e=VAL(c$):IF e<1 OR e>2 THEN 190
210 IF e=1 THEN GOTO 230
220 IF e=2 THEN GOSUB 3740:GOTO 950
230 ***** ACTUALIZACION CLASIFIC
  ACIONES EQUIPOS *****
240 SOUND 1,100,20,7:div=1
250 IF div=1 THEN equips=18 ELSE eq
  uips=20
260 CLS:PRINT "ACTUALIZACION CLASIF
  ICACION : " :INPUT clas
270 PRINT "-----"
  " :PRINT
280 FOR i=1 TO equips:cl(i)=0:NEXT i
290 GOSUB 3470
300 FOR n=1 TO equips
310 LOCATE 1,n+3:PRINT eq$(n):LOCAT
  E 20,n+3:PRINT "LUGAR EN LA CLASIFI
  CACION : " :INPUT clas
320 IF clas=equips THEN 350
330 FOR i=1 TO n:IF cl(i)=clas THEN
  350
340 NEXT i:GOTO 360
350 PRINT "ERROR EN EL NUMERO DEL E
  QUIPO !!! :LOCATE 30,n+3:PRINT cla
  s:
  GOSUB 100:PRINT "
  " :LOCA
TE 30,n+3:
  " :PRINT "
  " :GOTO 310
360 cl(n)=clas:LOCATE 14,n+3:PRINT
  clas

```

```

370 clas=eq$(n):eqcl$(div,clas)=cl
  as$
380 NEXT n:PRINT
390 CLS:PRINT "ACTUALIZACION CLASIF
  ICACION : " :INPUT clas
400 PRINT "-----"
  " :PRINT
410 FOR n=1 TO equips
420 PRINT n:TAB(5)eqcl$(div,n)
430 NEXT n
440 PRINT TAB(40) " ESTA BIEN LA CLA
  SIFICACION (S/N) ? "
450 GOSUB 100
460 IF c$<"S" AND c$<"N" THEN 450
470 IF c$="N" THEN 250
480 IF c$="S" THEN IF div=1 THEN cl
  as=1:div=2:GOTO 250
490 IF c$="S" THEN IF div=2 THEN cl
  as=2:GOTO 510
500 GOTO 450
510 ***** CONFECCION DE LA QUINI
  ELA *****
520 CLS:SOUND 1,100,20,7
530 IF clas=1 THEN PRINT "FALTA AC
  TUALIZAR LA CLASIFICACION DE 1 DIVI
  SION":
  div=1:GOSUB 100:GOTO
  250
540 IF clas=2 THEN PRINT "FALTA AC
  TUALIZAR LA CLASIFICACION DE 2 DIVI
  SION":
  div=2:GOSUB 100:GOTO
  250
550 INPUT "JORNADA Nr. : " :jorn
560 div=1
570 LOCATE 1,4
580 FOR i=1 TO 18
590 PRINT i:TAB(5)eqcl$(div,i)
600 NEXT i
610 div=2
620 WINDOW #1,20,39,4,24
630 FOR j=1 TO 20
640 PRINT #1,j+20:TAB(5)eqcl$(div,j
  )
650 NEXT j
660 WINDOW #2,40,80,1,25
670 PAPER #2,0:PEN #2,1
680 LOCATE #2,1,1:PRINT #2,"CONFECC
  ION QUINIELA 'C' para corregir"
690 LOCATE #2,1,2:PRINT #2,"-----"
  "
700 FOR par=1 TO 14
710 LOCATE #2,19,4:PRINT #2,"
  "
720 LOCATE #2,19,6:PRINT #2,"
  "
730 LOCATE #2,1,8:PRINT #2,"
  "
740 LOCATE #2,1,4:INPUT #2,"EQUIPO
  LOCAL : " :i=x$
750 IF x$="c" OR x$="C" THEN par=pa
  r-1:LOCATE #2,1,par+9:PRINT #2,"
  "
  :GOTO 700
760 LOCATE #2,1,6:INPUT #2,"EQUIPO
  VISITANTE : " :j=y
770 x=VAL(x$)
780 IF x<19 AND y>19 THEN GOSUB 940
160:GOTO 710
790 IF x>19 AND y<19 THEN GOSUB 940
  160:GOTO 710

```

SUBROUTINAS

Lineas	Cometido
100-110	Subrutina «inkey&».
1490-1680	Cálculo quiniela de 1 columna.
2940-3070	Cálculo boleto de 8 columnas.
3080-3240	Elección de fichas por el ordenador en relación a las diferencias de potencial.
3250-3350	Dibujado del boleto de 8 columnas.
3360-3460	Columna boleto de 8 columnas.
3470-3540	Clasificación inicial de los equipos de 1. ^o y 2. ^o
3550-3700	Grabación de la quiniela.
3740-3900	Lectura de la quiniela del cassette.


```

2450 LOCATE 20,17:INPUT "Nr. MAXIMO
DE 2. 1. ";mdos:PRINT
2460 LOCATE 20,20:PRINT "ESTAS DE A
CUERDO (S/N) ?"
2470 GOSUB 100:PRINT
2480 IF c<<"S" AND c<>"N" THEN 24
70
2490 IF c$="N" THEN 2360
2500 CLS:LOCATE 30,10:PRINT " F1JOS
= 1 "
2510 LOCATE 30,11:PRINT " -----"
2520 LOCATE 20,13:PRINT " LOS PONE
EL ORDENADOR (S/N) ? "
2530 GOSUB 100
2540 IF c<>"S" AND c<>"N" THEN 25
30
2550 IF c$="S" THEN 2730
2560 !=1:CLS:LOCATE 10,2:PRINT "QUI
NIELA"
2570 LOCATE 10,3:PRINT "-----"
2580 FOR qui=1 TO 14
2590 PRINT qui;TAB(7);elq$(qui);TAB(
20)";";TAB(23);evq$(qui)
2600 NEXT qui
2610 LOCATE 40,2:PRINT "Nr. DE FIJO
S : ";nfi:j
2620 LOCATE 40,3:PRINT "-----"
2630 FOR i=1 TO nfi:j
2640 LOCATE 52,9:PRINT " "
2650 LOCATE 40,7:PRINT "FIJO Nr. :
";i
2660 LOCATE 40,9:INPUT "PARTIT Nr.
: ";qui:part(qui)=1
2670 LOCATE 40,12:PRINT elq$(qui)
;TAB(54)";";TAB(57);evq$(qui)
2680 NEXT i
2690 LOCATE 40,22:PRINT "ESTAS DE A
CUERDO (S/N) ?"
2700 GOSUB 100
2710 IF c<<"S" AND c<>"N" THEN 27
00
2720 IF c$="N" THEN 2500
2730 ***** RESGUARDO BOLETO DE B
COLUMNAS *****
2740 CLS:GOSUB 1,100,20,7
2750 LOCATE 22,12:PRINT "CALCULANDO
. . . .";IF #=0 THEN GOSUB 3080
2760 FOR bol=1 TO nbol
2770 CLS:SOUND 1,100,20,7:PRINT
2780 PRINT " JORNADA Nr. : ";jorn;
2790 PRINT TAB(20)"Nr. DE FIJOS : "
;nfi;j
2800 PRINT TAB(60)"BOLETO Nr. : ";b
ol
2810 PRINT #PRINT " Nr. DE VARIANT
ES : ";i4:-nvar
2820 GOSUB 3250
2830 FOR colu=1 TO 40 STEP 5
2840 GOSUB 2940:IF vanvar AND dos<
=mdos THEN 2850 ELSE 2840
2850 GOSUB 3360
2860 NEXT colu
2870 GOSUB 100
2880 NEXT bol
2890 CLS:LOCATE 12,12:PRINT "QUIERE
S HACER OTRA (S/N) O VOLVER AL MENU
(M)?"
2900 GOSUB 100
2910 IF c<<"S" AND c<>"N" AND c<>

```

```

>"M" THEN 2900
2920 IF c$="M" THEN 950
2930 IF c$="N" THEN END ELSE 2360
2940 ***** CALCULO BOLETO B COLUMNA
S *****
2950 van=0:mdos=0
2960 FOR i=1 TO 14:uxd(i)=0:NEXT i
2970 FOR qui=1 TO 14
2980 IF part(qui)=1 THEN uxd(qui)=1
;van+van:1:GOTO 3060
2990 dp=celq(qui)-cevg(qui)
3000 pi=50+dp#2
3010 px=30-dp
3020 sort=INT(100/RND+1)
3030 IF sort>10 AND sort<=pi TH
EN uxd(qui)=1+van+van:1:GOTO 3060
3040 IF sort>pi AND sort<=pi+px TH
EN uxd(qui)=2:GOTO 3060
3050 IF sort>pi+px AND sort<=100 TH
EN uxd(qui)=3:dos=dos+1:GOTO 3060
3060 NEXT qui
3070 RETURN 'uxd=1 / 2=
x / i
3080 ***** ELECCION DE FIJOS POR EL
ORDENADOR *****
3090 RESTORE 3130
3100 FOR i=1 TO 19
3110 READ nt(i)
3120 NEXT i
3130 DATA 19,18,17,16,15,14,13,12,1
1,10
3140 DATA 9,8,7,6,5,4,3,2,1
3150 fi=0:FOR i=1 TO 14:part(i)=0
:iNEXT
3160 FOR j=1 TO 19
3170 FOR qui=1 TO 14
3180 dp=celq(qui)-cevg(qui)
3190 IF fi>jomfi;j THEN 3240
3200 IF part(qui)=1 THEN 3220
3210 IF dp<nt(j) THEN part(qui)=1:fi
i=j:omfi=j+1
3220 NEXT qui
3230 NEXT j
3240 RETURN
3250 ***** DIBUJO BOLETO B COLUMNA
S *****
3260 PLOT 15,312:DRAW# 600,0
3270 PLOT 15,312:DRAW# 0,-288:PLOT
295,312:DRAW# 0,-288;
PLOT 615,312:DRAW# 0
,-288
3280 PLOT 15,232:DRAW# 600,0:PLOT 1
5,152:DRAW# 600,0
3290 PLOT 15,88:DRAW# 600,0:PLOT 15
,24:DRAW# 600,0
3300 LOCATE 40,5:PRINT "1 2 3
4 5 6 7 8"
3310 FOR n=1 TO 4:LOCATE 3,n+6:PRIN
T n;TAB(7);elq$(n);TAB(20)";";TAB(23
);evq$(n) : NEXT n
3320 FOR n=5 TO 8:LOCATE 3,n+7:PRIN
T n;TAB(7);elq$(n);TAB(20)";";TAB(23
);evq$(n) : NEXT n
3330 FOR n=9 TO 11:LOCATE 3,n+8:PRIN
T n;TAB(7);elq$(n);TAB(20)";";TAB(23
);evq$(n) : NEXT n
3340 FOR n=12 TO 14:LOCATE 3,n+9:PR
INT n;TAB(7);elq$(n);TAB(20)";";TAB(
23);evq$(n) : NEXT n
3350 RETURN
3360 ***** COLUMNA BOLETO DE B CO
LUMNAS *****
3370 FOR n=1 TO 4:GOSUB 3420:LOCATE
colu+n,6:PRINT bs:NEXT n
3380 FOR n=5 TO 8:GOSUB 3420:LOCATE
colu+n,7:PRINT bs:NEXT n
3390 FOR n=9 TO 11:GOSUB 3420:LOCAT
E colu+n,8:PRINT bs:NEXT n
3400 FOR n=12 TO 14:GOSUB 3420:LOCA
TE colu+n,9:PRINT bs:NEXT n
3410 RETURN
3420 col=uxd(n)
3430 ON col GOTD 3440,3450,3460
3440 bs="1":RETURN
3450 bs="X":RETURN
3460 bs="2":RETURN
3470 ***** SUBROUTINA CLASIFICACIO
N INICIAL *****
3480 IF div=1 THEN RESTORE 3520:GOT

```

Serie Oro

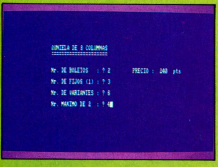
```

0 3500
3490 IF div=2 THEN RESTORE 3530
3500 FOR i=1 TO equip:READ eq$(i)
3510 NEXT i
3520 DATA BARCELONA,R.MADRID,BILBAO
,BIJON,AT.MADRID,BETIS,ZARAGOZA,R.S
OCIEDAD, SEVILLA,VALLADOLID,V
ALENCIA,CADIZ,CELTA,OSASUNA,ESPAÑOL
,SANTANDER, LAS PALMAS
,HERCULES
3530 DATA CARTAGENA,LOGROÑOS,HUELVA
,ELCHE,MURCIA,CORUNA,CASTELLON,ARAG
ON, CASTILLA,R.VALLCADEN
,SABADELL,SESTAO,ALBACETE,TENERIFE,
BARCELONA AT., MALAGA,MALLORCA,AT.MADRIEN
O, GVIWED,BIL
BAD AT.,MALAGA,MALLORCA,AT.MADRIEN O
3540 RETURN
3550 ***** GRABACION QUINIELA X
I *****
3560 CLS:SOUND 1,100,20,7
3570 LOCATE 29,10:PRINT "GRABACION
DE LA QUINIELA"
3580 LOCATE 29,11:PRINT "-----"
3590 LOCATE 26,13:PRINT "PARA DISCO
PULSAR CUALQUIER TECLA"
3600 LOCATE 12,15:PRINT "PARA CASSE
TTE PULSAR *REC Y PLAY* DESPUES O
UALQUIER TECLA"
3610 GOSUB 100
3620 SOUND 1,100,20,7
3630 CLS:LOCATE 27,12:PRINT "LA QUI
NIELA SE ESTA GRABANDO"
3640 OPENOUT "iquinela"
3650 PRINT #9,jorn
3660 FOR qui=1 TO 14
3670 PRINT #9,elq$(qui)
3680 PRINT #9,evq$(qui)
3690 PRINT #9,celq(qui)
3700 PRINT #9,cevg(qui)
3710 NEXT qui
3720 CLOSEOUT
3730 RETURN
3740 ***** LECTURA DEL CASSETTE
*****
3750 CLS:SOUND 1,100,20,7
3760 LOCATE 26,12:PRINT "PARA DISCO
PULSAR CUALQUIER TECLA"
3770 LOCATE 16,14:PRINT "PARA CASSE
TTE PULSAR *PLAY*DESPUES QUALQUIE
R TECLA"
3780 GOSUB 100
3790 SOUND 1,100,20,7
3800 CLS:LOCATE 20,12:PRINT "LA QUI
NIELA SE ESTA CARGANDO EN MEMORIA"
3810 OPENIN "iquinela"
3820 INPUT #9,jorn
3830 FOR qui=1 TO 14
3840 INPUT #9,elq$(qui)
3850 INPUT #9,evq$(qui)
3860 INPUT #9,celq(qui)
3870 INPUT #9,cevg(qui)
3880 NEXT qui
3890 CLOSEIN
3900 RETURN

```



P ara que tus dedos no realicen el trabajo duro, M.H. AMSTRAD Te hace por ti. Todas las listadas que incluyen este logotipo se encuentran a tu disposición en un cassette mensual. solución.



YOUR COMPUTER

La Revista de ordenadores de mayor venta en toda Europa

ISE PUBLICA DESDE AHORA EN ESPAÑA, EN FORMA DE CASSETTE!

Si, ya está confirmada la sensacional noticia. Muy pronto estará **en los quioscos** de toda España una selección de los mejores juegos y utilidades publicados por la prestigiosa Revista británica «YOUR COMPUTER», editados en cassette de alta calidad y con instrucciones en castellano.

El **prestigio** alcanzado por Your Computer, tanto en Inglaterra como en España y otros países, se debe, de una forma muy especial, a la **gran**

calidad de los programas que publica, la mayor parte de ellos en Código Máquina, y con la utilización de rutinas y técnicas de programación muy depuradas.

Ahora, a un precio inmejorable, podéis tener acceso a estos programas, **evitandoos** la difícil tarea de **teclearlos** en vuestro ordenador.

¡Y **cada mes** estará en la calle una nueva cinta!

Si no encontras la cassette de «Your Computer» en tu quiosco o tienda de informática, solicítala a nuestras oficinas:

SINTAX, S. A.

«YOUR COMPUTER»
Paseo de la Castellana, 268
28046 Madrid

Envía tus señas completas, teléfono y **marca de ordenador** e incluye **talón bancario**, o remite **Giro Postal** por el importe.

No te cobraremos gastos por el envío.

Si prefieres pagar **contra reembolso**, entonces incluye, junto a tu pedido, dos sellos de 50 ptas. cada uno para gastos de envío.

TAMBIEN DISPONIBLE
PARA

COMMODORE 64 y

SPECTRUM 48, PLUS, 128

1

YOUR COMPUTER

EL CORAZON DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES

AMSTRAD

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

695.-
PTAS.

DYNAMITE DAN

El superagente Dan Dinamita penetra en la peligrosa y custodiada mansión del Dr. Blitzen, con la misión de abortar el proyecto del Rayo Devastador, de ello depende la seguridad del mundo libre.

SOLAMENTE
CPC 464

E

n el día 10-4-1921, los servicios de espionaje de Estados Unidos, descubrieron un plan de desarrollo de un arma revolucionaria, con un poder energético equivalente a 21.100 kg de nitroglicerina.

El Rayo Devastador, nombre dado a la poderosa arma por su creador, está en plena fase de desarrollo y sólo existe en el papel.

Su diseñador, el Dr. Blitzen una vez concluidos los estudios teóricos y terminados los 230 metros cuadrados de planos y las 1.100 páginas de instrucciones de montaje, ha vendido su siniestro artefacto al bloque del Este.

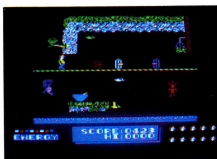
En su inespugnable mansión de los Alpes el Dr. y su equipo de seguridad, esperan la llegada del convoy que ha de transportar la documentación hasta sus compradores.

Transmitida la información a los servicios de inteligencia occidentales, éstos elaboran un plan para apoderarse de los documentos que pueden cambiar el destino del mundo.

En una audaz y arriesgada maniobra, los servicios de inteligencia consiguen comprar a la señora de la limpieza que presta sus servicios en la mansión.

Esta ha escondido cuidadosamente 8 cartuchos de dinamita, en diversos puntos de la casa, pero su afición por la bebida le ha hecho olvidar la situación de los mismos, cosa que no estaba prevista en los planos.

Una vez depositados los cartuchos entra en juego nuestro hombre; el superagente Dan dinamita.



Localizados éstos, el siguiente paso es llegar hasta la caja fuerte que guarda los planos del Rayo Devastador.

Encontrar el paredero de la caja y localizar los 8 cartuchos de dinamita, requiere eludir una larga serie de trampas y peligros, que sólo Dan puede conseguir, cualquier otro agente moriría en el empeño.

Después de todas estas peripecias volar la cerradura de la caja es una tarea de niños, con los planos en su poder, Dan puede decir que ha cubierto la mitad de su cometido.

Solamente le queda salir con vida de la mansión y entregar los planos al servicio de inteligencia.

Pero si penetrar en la mansión y descubrir el paradero de los planos era difícil, salir de ella cuesta mucho más, el cansancio, hace hasta en el agente secreto más famoso del mundo.



Depositado en el tejado de la misteriosa mansión por dirigible, comienza su arriesgada misión.

Pero la mansión no es una casa de campo normal y corriente, es una trampa mortal para cualquier extraño que quiera recorrerla. Pasadizos subterráneos, habitaciones letales, escaleras sin fin, pasillos subacuáticos, ascensores, trampas eléctricas y demás dispositivos extraños mantienen intacta la seguridad del malvado Blitzen.

Dan debe poner en constante riesgo su vida para salvar al mundo libre.

En primer lugar tiene que encontrar los ocho cartuchos de dinamita escondidos por la Sra. de la limpieza.





Mr. Joystick

Dan debe reponer fuerzas comiendo alimentos macrobióticos que están en determinados puntos de la casa: huevos, cócteles, frutas, queso, etc. ayudan a mantener la energía.

Otros objetos son imprescindibles para entrar en determinadas habitaciones y superar infernales corredores.

Dynamite Dan es el primer juego comercial en España que incorpora las siguientes opciones de juego:

Vidas infinitas.

Dan que aniquila bichos y objetos que le atacan.

Energía infinita.

Dan submarinista.

Opciones que se han añadido al juego normal, para facilitar el cumplimiento de la misión.

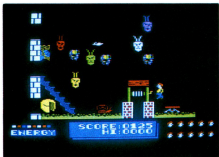
El juego en versión original era increíblemente difícil de resolver y en vista de esto se ha optado por añadir unas ayudas que permiten concluirlo con éxito.

Las distintas opciones de juego, pueden elegirse solas o combinadas, de forma que cada usuario pueda graduar a su gusto la dificultad de la misión.

Idea que permite resolver el juego utilizando todas las opciones primero, y luego una vez conocidos el emplazamiento de los objetos y la situación de los cartuchos de dinamita, ir poniendo el juego más difícil hasta conseguir completar la misión en la modalidad de juego normal.

Original idea en un juego comercial, que evita la desagradable experiencia de gastar dinero en un programa para entretenernos, y a las pocas horas tener que arrinconarlo porque nos resulta imposible pasar de un determinado punto, con el terrible pesar de haber tirado el dinero invertido.

Esperamos que surjan más programas adaptados a este concepto.



ROTACION DE CARACTERES

Los soberbios gráficos de alta resolución del Amstrad son ideales para mostrar en pantalla gráficos y diagramas de todo tipo.

Haciendo uso de estos poderosos comandos gráficos, los datos pueden ser representados de una forma mucho más intuitiva y digerible para las personas.



Supongamos que necesitamos crear un programa que dibuje en la pantalla una estadística en forma de diagrama de barras, para nuestro negocio.

Normalmente, los ejes, escala y etiquetas de los ejes de este tipo de dibujos están pensados para ser observados verticalmente, mientras que el modo de impresión normal de una cadena de caracteres está implementada en sentido horizontal, de izquierda a derecha.

Si queremos que nuestras barras y texto aparezcan verticales, una forma de lograrlo sería la que usa el programa número 1.

No obstante, ésta no es una escritura vertical real; gire la cabeza 90 grados a derecha e izquierda y trate de leer el texto. Se entenderá perfectamente lo que quiero decir.

Parece ser que hace falta una rutina que se capaz de girar los caracteres de una cadena alfanumérica a derecha o izquierda 90 grados, e imprimirla en la pantalla en la posición del cursor de texto.

De nuevo el lenguaje máquina y las RSX

Naturalmente, hay que recurrir al lenguaje máquina para lograrlo, pero vamos a rizar un poco el rizo y, para nuestra mayor comodidad, crearemos dos nuevos comandos Basic que basta con llamarlos y darles la cadena para que lo hagan.

Como nuestros lectores ya saben, esto está «chupado» haciendo uso de las facilidades que el sistema operativo del Amstrad proporciona bajo el nombre de RSX, Extensiones Residentes del Sistema.

Uno de los nuevos comandos, «I PRINT.UP», rota todos los caracteres de una cadena alfanumérica que se le proporciona como parámetro 90 grados en el sentido contrario a las agujas del reloj e imprime los caracteres verticalmente hacia arriba.

El otro, «I PRINT.DOWN», hace lo mismo que su compañero, pero la impresión es de arriba a abajo y la rotación 90 grados en el sentido de las agujas del reloj.

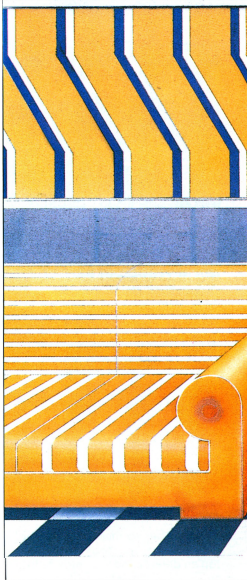
El programa número 2 es una corta rutina en Basic que coloca el código máquina a partir de &A000 y activa los nuevos comandos, mientras que el programa 3 es el listado en lenguaje ensamblador de la rutina en máquina.

Los afortunados poseedores de un ensamblador podrán colocarlo en la zona de memoria que más les acomode y variar los nombres de los nuevos comandos si los propuestos por nosotros no les gustan.

Obsérvese que ambos deben ir precedidos, al invocarlos desde Basic, del signo «!» para que el sistema sepa a qué atenerse:

! PRINT.DOWN
! PRINT.UP

y que el listado en ensamblador, aunque obtenido mediante el GEN5 de Hisoft, puede introducirse sin problemas usando el que publicamos en el número 8 de AMSTRAD Semanal.



Sintaxis

La sintaxis de los nuevos comandos no puede ser más sencilla, basta con invocarlos y darles después la cadena e imprimir, bien entrecomillada o bien como variable:

frase\$ = «Amstrad»
! PRINT.UP, frase\$
! PRINT.DOWN, «Amstrad»

Aquellas personas que tengan un CPC464, deben usar solamente el método siguiente:

frase\$ = «Amstrad»
! PRINT.UP, @ frase\$

La rutina en máquina, la que realmente hace el trabajo (el resto es la inicialización típica de un RSX), comienza en la dirección hexa &A031, inmediatamente debajo de la etiqueta «AQUI». Su funcionamiento es el siguiente:

a) LD L, (IX+0)
LD H, (IX+1)



coloca en el registro HL la dirección de memoria donde se encuentra el descriptor de cadena que el sistema operativo construye (*recordad que IX apuntaba a esa zona, denominada bloque de parámetros*).

b) LD B, (HL)

el primer byte del descriptor de cadena contiene su longitud, así que la cargamos en el registro B.

c) INC HL
LD E, (HL)
INC HL
LD D, (HL)
EX DE, HL

los dos bytes siguientes contienen la dirección verdadera de memoria donde los caracteres de la cadena se almacenan físicamente; la introducimos en el registro DE y luego en el HL, porque interesa que estén en él para el funcionamiento del resto de la rutina.

Alcanzamos ahora el bucle principal (*etiqueta «OTRO»*) que girará los caracteres y los imprimirá uno a uno mediante la instrucción DJNZ (de ahí que la longitud de la cadena esté en B).

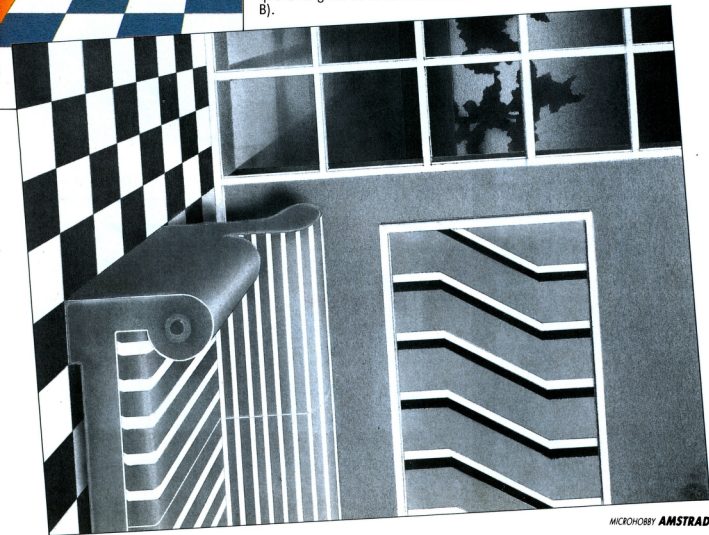
Programación

A lo largo de este bucle hay que hacer unas cuantas cosas.

Lo primero de todo, es hallar la dirección donde está la matriz de puntos que compone el carácter que vamos a girar; esto lo hacemos llamando a la rutina firmware MATRIZ (*CALL MATRIZ*), que devuelve la dirección en HL.

A continuación activamos la ROM baja (*CALL ROMSI*), ponemos la dirección de la matriz de puntos en DE y llamamos a ARRIBA o ABAJO que giran el carácter a derecha o izquierda.

Deshabilitamos la ROM (*CALL ROMNO*) y convertimos la zona de memoria de nuestro programa que comienza en CHR255 en una matriz de puntos con el mismo formato que las nativas del **Amstrad** llamado a SETMAT (*CALL SETMAT*), de SET MATRIZ.



```

10 REM PROGRAMA 1
20 CLS
30 A$="AMSTRAD"
40 LOCATE 10,10
50 FOR I=1 TO LEN(A$)
60 PRINT MID$(A$,I,1);CHR$(10);CHR$(
I);
70 NEXT

```

P Program Acción

A000		DIRORG:	LD A, #A000	A067	3E0B	LD A, #0B
A000		ORG	DIRORG	A068	CD5ABB	CALL PRINT
A000	0109A0	LD	BC, CMTAB	A06E	C1	POP BC
A003	2124A0	LD	HL, ESP_TR	A06F	E1	POP HL
A006	C3D1BC	JP	RSXCOM	A070	23	INC HL
A009	11A0			A071	10CA	DJNZ OTRD
A00B	C32BA0	COMTAB:	DEFW NDMTAB	A073	C9	RET
A00E	C32CA0	JP	P_DOWN	A074	2197A0	
A011	5052494E	JP	P_UP	A077	1A	ARRIBA: LD HL, CHR255
A01A	CE	NDMTAB:	DEFM "PRINT, DOWN"	A078	060B	LD B, #0B
A01B	5052494E	DEFM	"PRINT, U"	A07A	17	BUCLE1: RLA
A022	D0	DEFB	"*"+#B0	A07B	CB1E	RR (HL)
A023	00	DEFB	#00	A07D	23	INC HL
A024				A07E	10FA	DJNZ BUCLE1
A02B	3E0A	ESP_TR:	DEFS 4	A080	13	INC DE
A02A	1B02	P_DOWN:	LD A, #0A	A081	0D	DEC C
A02C	3E0B	JP	A001	A082	20F0	JR NZ, ARRIBA
A02E	3296A0	P_UP:	LD A, #0B	A084	C9	RET
A031	D0A600	A001:	LD L (UPDOWN), A	A085	219FA0	ABAJ0: LD HL, I_CHAR
A034	DD6601	LD	H, (IX+0)	A08B	1A	LD A, (DE)
A037	46	LD	B, (IX+1)	A089	0A0B	LD B, #0B
A038	23	INC	HL	A08B	2B	BUCLE2: DEC HL
A039	5E	LD	E, (HL)	A08C	17	RLA
A03A	23	INC	HL	A08D	CB16	RL (HL)
A03B	56	LD	D, (HL)	A08F	10FA	DJNZ BUCLE2
A03C	EB	EX	DE, HL	A091	13	INC DE
A03D	7E	OTRD:	LD A, (HL)	A092	0D	DEC C
A03E	E5	PUSH	HL	A093	20F0	JR NZ, ABAJ0
A03F	C5	PUSH	BC	A095	C9	RET
A040	CD45BB	CALL	MATRIZ	A096	00	UPDOWN: DEFB 0
A043	CD06B9	CALL	ROMSI	A097	00	CHR255: DEFB 0
A046	EB	EX	DE, HL	A098	00	DEFB 0
A047	0E0B	LD	C, #0B	A099	00	DEFB 0
A049	3A96A0	LD	A, (UPDOWN)	A09A	00	DEFB 0
A04C	F5	PUSH	AF	A09B	00	DEFB 0
A04D	FE0A	CP	#0A	A09C	00	DEFB 0
A04F	CC74A0	CALL	Z, ARRIBA	A09D	00	DEFB 0
A052	C4B5A0	CALL	NZ, ABAJ0	A09E	00	DEFB 0
A055	CD09B9	CALL	ROMNO	A09F	00	DEFB 0
A058	3EFF	LD	A, #FF	B906		N_CHAR: DEFB 0
A05A	2197A0	LD	HL, CHR255	B909		ROMSI: EQU #B906
A05D	CD4BBB	CALL	SETMAT	BB5A		ROMNO: EQU #B909
A060	3EFF	LD	A, #FF	BB5A		PRINT: EQU #BB5A
A062	CD5ABB	CALL	PRINT	BB5A		MATRIZ: EQU #BB5A
A065	F1	POP	AF	BBAB		SETMAT: EQU #BB5A
A066	CD5ABB	CALL	PRINT	BCD1		RSXCOM: EQU #BCD1
				AA0A		END

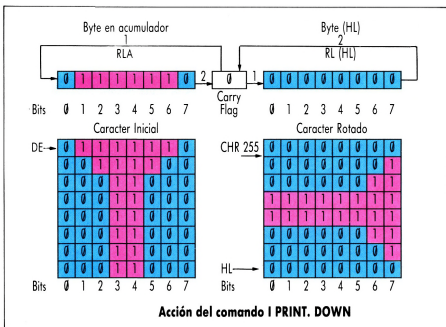
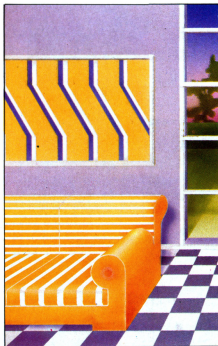


Figura 1

Por fin imprimimos el caracter y re-
petimos el ciclo.

Rutinas de giro

Vamos ahora con las rutinas ARRI-
BA y ABAJO.

Si se mira con detalle la rutina
ARRIBA, se puede ver que consta de
dos bucles.

El primero recorre los 8 bytes que
forman un caracter de uno en uno,
cargando el A con el contenido del
registro DE, que es donde la rutina
MATRIZ, tras el EX DE, HL, deja la
dirección requerida.

El segundo bucle, más interno, ro-
ta los caracteres bit a bit: se rota el
acumulador hacia la izquierda, de
forma que el bit 7 acaba en el Carry
flag (RLA).

Luego, este bit se incluye en el

apropiado byte de CHR255 con la
instrucción de rotación a la derecha
RR (HL).

El resultado de estas manipulacio-
nes es que la trama de bytes y bits
de CHR255 contiene al final una co-
pia exacta del caracter original que
estaba en el acumulador, pero gira-
do 90 grados.

Ya habréis podido observar que
todo esto resulta extremadamente
sencillo de manejar gracias al siste-
ma RSX y a las facilidades previstas
por el firmware del **Amstrad**. Los dos
nuevos comandos ayudarán a que
vuestros gráficos estén dotados de
una presentación mucho más agra-
dable y profesional.

Sólo para divertirse **¿alguien po-
dría adaptar la rutina para que el
texto resultante apareciera tal y como
se vería en un espejo?**

LLAMADAS AL FIRMWARE

ETIQUETA	DIRECCION	PROPOSITO
ROMSI	&B906	Activa la ROM baja donde se encuentra el firm- ware (<i>direcciones &0-&3FFF</i>).
ROMNO PRINT MATRIZ	&B909 &BB5A &BBA5	Desactiva la ROM baja. Imprime el caracter contenido en el acumulador. Encuentra la dirección de la matriz que consti- tuye el caracter dado como parámetro en el acu- mulador. Devuelve dicha dirección en el regis- tro HL.
SETMAT	&BBA8	Convierte una zona de memoria de 8 bytes al formato necesario para que represente un carac- ter alfanumérico. Requiere dos parámetros: el código ASCII del caracter en el acumulador y la dirección de la zona de memoria en HL. Le dice al firmware que existe una RSX. Necesi- ta dos parámetros: en BC, la dirección de la ta- bla de nombres de la RSX, y en HL la dirección de un espacio de trabajo (ESP_TR) de 4 bytes de longitud.
RSXCOM	&BCD1	

Programación

ETIQUETAS

```

ABAJO A0B5 ADU1 A02E ARRIBA A074
BUCLE1 A07A BUCLE2 A0BB CHR255 A097
CMTAB A009 DIRDIR A000 ESP_TR A024
MATRIZ BBA5 NDMTAB A011 N_CHAR A09F
OTRO A03D PRINT B85A F_DOWN A02B
P_UP A02C ROMNO F009 RMSI B906
RSXCOM BCD1 SETMAT BBA8 UPDOWN A096
    
```

```

10 REM Side Writer
20 REM By R.A.Waddilove
30 REM (c)Amstrad Semanal
40 MEMORY &9FFF
50 a=&A000
60 FOR i=1 TO 14
70 sum=0:READ a$,b$
80 FOR j=1 TO 21 STEP 2
90 t=VAL("%+MID$(a$,j,2)");POKE a,k
100 sum=sum+k:a=a+1
110 NEXT
120 IF sum<VAL("(M"+a$) THEN PRINT
130 NEXT
140 CALL &A000
150 PRINT "Side writer esta listo..
"
160 PRINT"Dispones de los comandos
'print.up' y 'print.down'"
170 DATA 0109A02124A0C5D1BC11A0,490
180 DATA C328A0C32CA0505249E5A,447
190 DATA 2E444F57CE505249E5A2E,3A1
200 DATA 55D000000000003E0A1B02,187
210 DATA 3E0B296A0DD6E00DD6601,440
220 DATA 46235E235A6B7EES3CDA5,525
230 DATA BBDC06B9E0E083A9A0F5,5AD
240 DATA FE0AC74A0C485A0CD09B9,660
250 DATA 3EFF2197A0CDBBB3EFFDC,6CF
260 DATA 5A0BF1CDBABE3E0CDB5BB,610
270 DATA C1E12310AC92197A01A06,4E0
280 DATA 0B17CB1E2310F4130D20F0,3A5
290 DATA C9219FA01A060B2B17CB16,374
300 DATA 10FA130D20FC900000000,303
    
```

Sigue la línea del futuro

no/har



GARANTIA
AMSTRAD
ESPAÑA

"Al comprar
tu AMSTRAD
te regalamos
3 programas
originales"

ORDENADOR AMSTRAD 464 F. Verde 66.900 Ptas.
ORDENADOR AMSTRAD 6.128 F. Verde 109.500 Ptas.

DISTRIBUIDOR INDESCOMP PARA ORENSE:



Almacenes Méndez

CAPITAN CORTES, 17. TELEF.- 228607. ORENSE

Servicios a tiendas y almacenes

Presenta este anuncio y obtendrás un OBSEQUIO en tu compra

ROBOT RON

Las pulgas mutantes atacan de nuevo, como cada primavera, año tras año.

No hay problema. ¡Enviad a buscar a Robot Ron!

Ron es un robot dirigido por control remoto especializado en destruir pulgas mutantes y armado de un super láser, por ejemplo. Pulga que pillas, pulga que mata.

Como era de esperar, nuestro robot no es inmune al ataque de las ominosas pulgas, en cuanto le tocan a él, lo destruyen.

Bueno, no os preocupéis. Podemos reconstruirlo, para eso está la tecnología, pero sólo tres veces. **Algún límite tiene que haber, ¿no?**

El programa consta de nueve pantallas distintas donde puedes probar tu habilidad, velocísimos reflejos, etc.

A medida que vamos avanzando por las pantallas, hay menos pulgas, pero se mueven con mayor agilidad y mala idea.

Hemos usado una rutina en máquina para pintar las pulgas, el robot y el disparo, la cual cumple su función unas 10 veces más rápido que el básico.

Ojo con las Datas, son muchas y es fácil equivocarse.



```

10 REM Robot Ron v The Evil Weevils
20 REM By R.A.Maddilove
30 REM(C) Amstrad Semanal
40 MEMORY $9FFF
50 GOSUB 420:REM inicializacion
60 GOSUB 1530:REM instrucciones
70 WHILE NOT Fed-up
80 GOSUB 1230:REM scores
90 WHILE lives
100 GOSUB 670:REM asigna variables
110 WHILE a(x,y)>0 AND m
120 GOSUB 200:GOSUB 260:IF p THEN G
GOSUB 290 ELSE GOSUB 360
130 WEND
140 IF m THEN GOSUB 930 ELSE screen
=screen+(screen-1):GOSUB 910
150 WEND
160 IF sc>highscore(5) THEN GOSUB 135
0
170 WEND
180 END
190 REM -----mueve hombre -----
200 i:=+(INKEY(g)>1)-(INKEY(h)>1)
i:=+(INKEY(e)>1)-(INKEY(f)>1):IF
a(i,j)>1 THEN RETURN
210 IF a(i,j)>1 THEN RETURN
220 CALL $A000,x,y,i,j,2:IF p=0 AND
i<>x OR j<>y THEN mx:=x:my:=y
SOUND 132,100,1,4
230 x:=i:y:=j
240 RETURN
250 REM ----- mueve robots -----
260 SOUND 130,m*20+500,10,4:IF r(n,0
):j=r(n,1):IF i=0 THEN 300
270 IF a(i,j)>1 THEN LOCATE i,j:PR
INT " r(n,0)=0:a(i,j)=0:m:=1:GOT
O 300
280 i:=-(i<>x)+(i>x):j:=+(j>y)-(j<y)
290 IF a(i,j)=0 THEN CALL $A000,r(n
,0),r(n,1),j,0:IF r(n,0),r(n,1)=0
:a(i,j)=1:r(n,0)=1:r(n,1)=j
300 i:=r(n+1,0):j:=r(n+1,1):IF i=0 TH
EN 340
310 IF a(i,j)>1 THEN LOCATE i,j:PR
INT " r(n+1,0)=0:a(i,j)=0:m:=1:G
OTO 340
320 i:=-(i<>x)+(i>x):j:=+(j>y)-(j<y)
330 IF a(i,j)=0 THEN CALL $A000,r(n
+1,0),r(n+1,1),j,1:IF r(n+1,0),r(n
+1,1)=0:a(i,j)=1:r(n+1,0)=1:r(n+1,
1)=j
340 n:=(n+2)MOD screen$2:IF r(n,0) 0
R r(n+1,0) OR m=0 THEN RETURN ELSE
340
350 REM ----- fuego -----
360 IF INKEY(d)<0 THEN RETURN
370 p:=m+1:IF r(n,0)=1 THEN p:=0:RE
TURN ELSE IF a(p,q)=1 THEN CA
LL $A000,p,q,p,q,4:IF p=0:a(p,q)=4:RET
URN
380 CALL $A000,p,q,p,q,3:IF a(p,q)=2:5
SOUND 129,10,300,15,1,1:RETURN
390 i:=m+1:j:=m+1:IF a(i,j)=1 THEN
a(p,q)=0:a(i,j)=4:CALL $A000,p,q,i,
j,4:IF p=0:SOUND 129,1500,100,15,2,0,1
5:sc:=sc+1:LOCATE 7,1:PRINT MID$(STR
$(sc),2):RETURN
400 a(p,q)=0:IF a(i,j)=3 THEN LOCAT
E p,q:PRINT " *p:=0:SOUND 129,0,0,0
ELSE CALL $A000,p,q,i,j,3:IF a(i,j)=2
:IF i=q
410 RETURN
420 REM ----- inicializar -----
430 DEFINT a-z
440 DIM a(20,25),r(20,1),name$(5),h
iscore(5)
450 FOR i=1 TO 5:name$(i)="E" grupo
A:hiscore(i)=0-1:10:NEXT
460 ENT 1,200,2,1:ENV 1,15,1,-1,25
470 ENV 2,15,-1,10
480 FOR i=0 TO 98
490 READ a$:POKE $A000,i,VAL("&"&a$
)
500 NEXT

```

TABLA DE VARIABLES

x,y	Coordenadas de Ron.
a(20,25)	Mapa de pantalla.
r(20,1)	Coordenadas de las pulgas.
name\$(5)	Nombres en la tabla de puntos.
hiscore(5)	Máxima puntuación.
p,q	Coordenadas del disparo.
sc	Puntos.
screen	Número de pantalla.
m	Desplazamiento a la izquierda de las pulgas.
n	Pulga que será movida.

Serie Oro

```

510 DATA DD,7E,00,67,87,87,87,3D
47,40,26,C0,7E,04,7D,87,87,6F,1
1,50,00,DD,46,02,05,19,10,FD,ED,26,
C0,DD,7E,08,3D,87,87,6F,11,50,00,DD
46,06,05,19,10,FD,11,FD,07,06,08,3
6,00,23,36,00,23,36,00,23,36,00
520 DATA 19,10,FD,EF,11,7F,81,05,0B
16,72,13,73,16,72,13,23,1A,72,13,2
3,16,72,13,78,01,FD,07,19,47,10,EF,
C9
530 FOR j=0 TO 159:READ j:FDIE 6A10
041,j:NEXT
540 REM Robot 1
550 DATA 20,60,60,60,105,131,195,40,
105,169,233,40,41,3,3,40,20,0,20,0,
0,40,40,0,0,160,160,0,80,160,240,0
560 REM Robot 2
570 DATA 80,240,240,160,160,1,2,80,
160,169,86,80,161,3,3,82,80,2,1,160
,0,240,240,0,20,0,0,40,40,0,0,20
580 REM Hombre
590 DATA 0,64,128,0,0,128,64,0,0,64
,128,0,4,12,12,8,8,76,140,4,32,12,1
,2,16,0,8,4,0,4,8,4,8
600 REM Laser
610 DATA 0,0,0,0,0,8,4,0,0,0,0,0,0,
84,168,0,0,84,168,0,0,0,0,0,8,4,0
,0,0,0,0
620 REM Explosión
630 DATA 8,84,168,4,4,136,68,8,168,
8,4,84,84,216,228,168,84,216,228,16
8,168,8,4,84,4,136,68,8,8,84,168,4
640 FOR i=1 TO 20:a(i,2)=3:a(i,24)=
3:NEXT
650 FOR i=2 TO 24:a(i,1)=3:a(20,i)=
3:NEXT
660 RETURN
670 REM -----assigna variables-----
680 CLS #1:PEN 1
690 FOR i=2 TO 19:FOR j=3 TO 23:a(i
,j)=0:NEXT:NEXT
700 j=1+INT(RND#5):DEG
710 FOR i=0 TO 250 STEP 10
720 x=10+6*ICOS(18j):y=12+6*SIN(18j):L
OCATE x,y:PRINT CHR$(207):a(x,y)=5
730 NEXT
740 x=10:y=12:p=0:a(x,y)=0
750 m=screen#2
760 FOR i=0 TO m-1
770 IF RND>0.5 THEN r(i,0)=2+INT(RN
D#5) ELSE r(i,0)=15+INT(RND#5)
780 IF RND>0.5 THEN r(i,1)=3+INT(RN
D#5) ELSE r(i,1)=15+INT(RND#5)
790 IF a(r(i,0),r(i,1)) THEN 770 EL
SE a(r(i,0),r(i,1))=1
800 NEXT
810 REM ----- crea pantalla -----

```

```

820 MOVE 0,18:DRAW 0,380,1:DRAW 638
,380:DRAW 638,18:DRAW 0,18
830 PEN 15:LOCATE 1,1:PRINT "Score:
" MID$(STR$(sc),2):PEN 14:PRINT TAB
(12)"High:" MID$(STR$(hi)score(1)),2)
840 PEN 13:LOCATE 1,25:PRINT "Lives
:" MID$(STR$(lives),2):PEN 12:PRINT
TAB(12)"Screen:" MID$(STR$(10-scre
e)),2)
850 CALL @A000,x,y,x,y,2
860 PEN 4
870 FOR i=0 TO screen#2-1
880 LOCATE r(i,0),r(i,1):PRINT CHR$(
231)
890 NEXT
900 PEN 15
910 FOR i=0 TO 2000:NEXT
920 RETURN
930 REM ----- muerte -----
940 SOUND 129,1500,100,15,2,0,15
950 lives=lives-1
960 CALL @A000,x,y,x,y,4
970 FOR i=0 TO 5000:NEXT
980 IF lives THEN RETURN
990 TAB
1000 PLOT -10,-10,2:MOVE 160,200:FR
INT "Game Over":
1010 TAGOFF:PRINT CHR$(23):CHR$(1):
i:TAB
1020 PLOT -10,-10,6:MOVE 164,202:FR
INT "Game Over":
1030 TAGOFF:PRINT CHR$(23):CHR$(0):
1040 FOR i=0 TO 5000:NEXT
1050 RETURN
1060 REM ----- sonido -----
1070 WHILE INKEY(47)<0 AND INKEY(76
)<0
1080 RESTORE 1190:READ pitch: SOUND
129,213,200,4: SOUND 130,253,200,4:
SOUND 1,127,2000,4
1090 WHILE pitch>0 AND INKEY(47)<0
AND INKEY(76)<0
1100 SOUND 4,pitch,10,6: SOUND 4,159
,10,6: READ pitch
1110 WEND
1120 READ pitch: SOUND 129,319,200,5
: SOUND 1,213,200,4: SOUND 1,159,200,
4
1130 WHILE pitch>0 AND INKEY(47)<0
AND INKEY(76)<0
1140 SOUND 4,pitch,12,6: READ pitch
1150 WEND
1160 WEND
1170 IF INKEY(47)>-1 THEN d=47:e=69
:f=71:g=39:h=31 ELSE d=76:e=72:f=73
:g=74:h=75

```



```

1180 RETURN
1190 DATA 142,127,119, 142,127,119,
127,119,106, 127,119,106, 119,106,
95, 119,106,95, 106,95,84, 106,95,8
4,0
1200 DATA 80,84,95, 84,95,106, 95,1
06,119, 106,119,127
1210 DATA 80,84,95, 84,95,106, 95,1
06,119, 106,119,127
1220 DATA 119,127,142,159,169,190,2
13,239,253,284, 319,253,284, 319,239
,253,284,213,239,253,190,213,239,169
,190,213,190,179,169,159,142,159,0
1230 REM ----- scores -----
1240 MODE 1: BORDER 1
1250 DRAW 0,398,1: DRAW 638,398: DRAW
638,0: DRAW 0,1
1260 PAPER 2: PEN 3: LOCATE 11,2: PRIN
T " High Scores ": PAPER 0
1270 FOR i=1 TO 5
1280 LOCATE 7,5+i*2: PEN 3: PRINT i:
PEN i: PRINT " : name#(i) : " : PEN 2
: WHILE POS(#0)<29: PRINT " : : i: WEND:
PEN 3: PRINT hi: score(i)
1290 NEXT
1300 PAPER 1: PEN 0: LOCATE 5,24: PRIN
T " Pulsa SPACE o FIRE " : PAPER 0
1310 WHILE INKEY("<")="": WEND: GOSUB 10
630

```

MICROHOBBY AMSTRAD SEMANAL LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS, PARA QUE VD. NO TENGA QUE TECLEARLOS

Todos los programadores y aficionados a la microinformática sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta de cassette, sólo por **675 ptas. (sin más gastos por envío)**.

Envíanos con la menor demora posible, el cupón correspondiente.

Serie Oro

```

1320 SOUND 129,0: SOUND 130,0: SOUND
132,0
1330 BORDER 0: MODE 0: WINDOW #1,2,19
,3,23: SC=0: SCREEN=9: LIVES=3
1340 RETURN
1350 REM ----- high score -----
1360 MODE 1
1370 DRAW 0,398,1: DRAW 638,398: DRAW
638,0: DRAW 0,0
1380 PEN 1: PAPER 0: LOCATE 7,5: PRINT
" Nueva puntuacion maxima"
1390 LOCATE 10,10: PEN 3: PRINT "Dise
no tu nombre"
1400 LOCATE 14,20: PAPER 2: PEN 3: PR
INT " 7....."
1410 LOCATE 16,20: IS="": NAME$(5)="
"
1420 WHILE INKEY$("<>"); WEND
1430 WHILE IS("<>CHR$(13))
1440 IF IS("<>CHR$(31) AND IS("<>2") AND
LEN(NAME$(5)) < 10 THEN NAME$(5)=NAME
$(5)+IS: PRINT IS
1450 IF IS("<>CHR$(127) AND LEN(NAME$(
5)) THEN NAME$(5)=LEFT$(NAME$(5),LE
N(NAME$(5))-1): PRINT CHR$(8): CHR$(1
6)
1460 IS=INKEY$
1470 WEND
1480 HISCORE(5)=SC
1490 FOR I=5 TO 2 STEP -1
1500 IF HISCORE(I): HISCORE(I-1) THE
N K$=NAME$(1): NAME$(1)=NAME$(I-1): N
AME$(I-1)=K$: SC=HISCORE(I): HISCORE
(I)=HISCORE(I-1): HISCORE(I-1)=SC
1510 NEXT I
1520 RETURN
1530 REM ----- instrucciones -----
1540 MODE 1: CALL &BB4: CALL &BC02
1550 INK 2,2: INK 0,11: INK 3,11: BORD
ER 11
1560 WINDOW #1,11,31,1,3: PAPER #1,3
:CLS #1
1570 WINDOW #1,10,30,2,4: PAPER #1,1
:CLS #1
1580 PEN 3: LOCATE 1,25: PRINT "Robot
Ron": PLOT -10,2
1590 FOR X=0 TO 158 STEP 2: FOR Y=2
TO 14 STEP 2
1600 IF TEST(X,Y) THEN PLOT X+230,Y
+Y*344: PLOT X+230,Y+Y*346
1610 NEXT Y: NEXT X: PEN #1,2

```

```

1620 WINDOW #1,2,40,5,24: PAPER #1,3
:CLS #1: INK 3,6
1630 WINDOW #1,1,39,6,25: PAPER #1,1
:CLS #1
1640 PRINT #1
1650 PRINT #1,"Invasion de pulgas g
igantes!"
1660 PRINT #1
1670 PRINT #1," Estas criaturas deb
en ser exterminadas"
1680 PRINT #1," a cualquier precio.
"
1690 PRINT #1," Envia a Robot Ron
para que pueda"
1700 PRINT #1," destruirlas con su
super laser."
1710 PRINT #1
1720 PRINT #1," Ron puede controlar
se usando las"
1730 PRINT #1," siguientes teclas..
"
1740 PRINT #1
1750 PRINT #1," A=aar Z=abaj
<=izq >=der"
1760 PRINT #1
1770 PRINT #1,SFC(11)"SPACE dispara
"
1780 PRINT #1
1790 PRINT #1,SFC(11)"D use Joystic
k"
1800 PEN #1,1: PAPER #1,2: LOCATE #1,
3,19: PRINT #1," Pulsa SPACE o FIRE
para comenzar ": PAPER #1,0
1810 WHILE INKEY$("<>"); WEND
1820 B0SUB 1060
1830 RESTORE 1840: FOR I=0 TO 15: REA
D J: INK 1, J: NEXT I: INK 7,26,13
1840 DATA 0,18,8,24,2,8,20,26,15,3,
7,9,25,16,22,23
1850 RETURN

```

```

A000 D07E00 LD A,(1X+0)
A003 87 ADD A,A
A004 87 ADD A,A
A005 87 ADD A,A
A006 87 ADD A,A

```

```

A007 87 ADD A,A
A008 3247A0 LD (LA047),A
A009 2650 LD H,#00
A00B D07E04 LD A,(1X+4)
A010 3D DEC A
A011 87 ADD A,A
A012 87 ADD A,A
A013 6F LD L,A
A014 115000 LD DE,#0050
A017 D04602 LD B,(1X+2)
A01A 05 DEC B
A01B 19 LA01B ADD HL,DE
A01C 10FD DJNZ LA01B
A01E E5 PUSH HL
A01F 26C0 LD H,#00
A021 D07E0B LD A,(1X+8)
A024 3D DEC A
A025 87 ADD A,A
A026 87 ADD A,A
A027 6F LD L,A
A02B 115000 LD DE,#0050
A02B D04606 LD B,(1X+6)
A02E 05 DEC B
A02F 19 LA02F ADD HL,DE
A030 10FD DJNZ LA02F
A032 11FD07 LD DE,#07FD
A035 0608 LD B,#08
A037 19 LA037 LD (HL),#00
A039 23 INC HL
A03D 3600 LD (HL),#00
A03F 23 INC HL
A040 3600 LD (HL),#00
A042 19 ADD HL,DE
A043 10F2 DJNZ LA037
A045 E1 POP HL
A046 1100A1 LD DE,#A100
A049 0608 LD B,#08
A04B 1A LA04B LD A,(DE)
A04C 77 LD (HL),A
A04D 13 INC DE
A04E 23 INC HL
A04F 1A LD A,(DE)
A050 77 LD (HL),A
A051 13 INC DE
A052 23 INC HL
A053 1A LD A,(DE)
A054 77 LD (HL),A
A055 13 INC DE
A056 23 INC HL
A057 1A LD A,(DE)
A05B 77 LD (HL),A
A059 13 INC DE
A05A 76 LD A,B
A05B 01FD07 LD BC,#07FD
A05E 09 ADD HL,BC
A05F 47 LD B,A
A060 10E9 DJNZ LA04B
A062 C9 RET

```



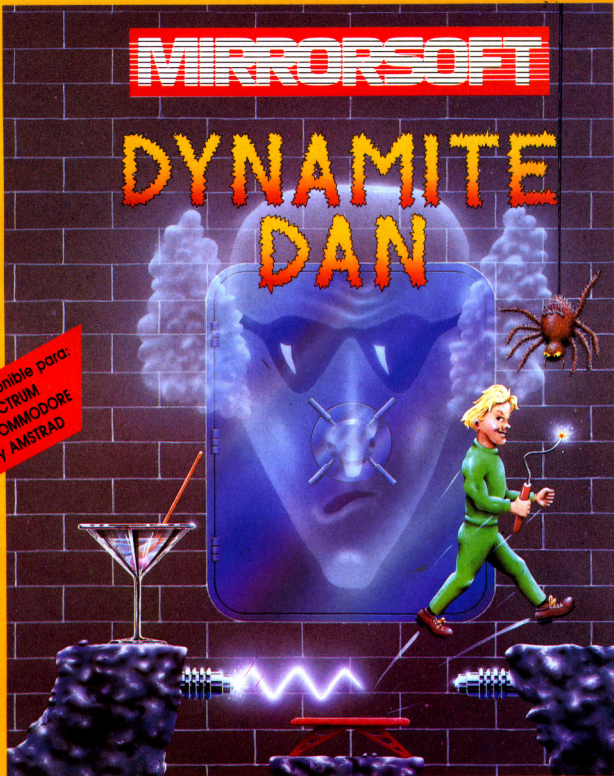
Pon **DINAMITA** a tu imaginación

¡¡POR PRIMERA VEZ EN EL MUNDO UN PROGRAMA DE JUEGOS QUE PUEDES HACER VARIAR A TU MEDIDA CUANTAS VECES QUIERAS!!.

MIRRORSOFT

**DYNAMITE
DAN**

2.100 Ptas.
Disponible para:
SPECTRUM
COMMODORE
Y AMSTRAD



¡No te lo pierdas!

DISTRIBUIDO EN ESPAÑA POR:

circulo de soft

MICROAMIGO S.A.

P.º de la Castellana, 268, 3.º C. 28046-MADRID.
Tel.: (91) 733 25 00

ALIENS ATERRIZANDO

La semana pasada vimos cómo está organizada la memoria de pantalla en Modo 0 con la ayuda de algunos programas sencillitos... En esta ocasión intentaremos dibujar en la pantalla un carácter multicolor por medio de varias rutinas cortas en código máquina.

A

ntes de nada hagamos un pequeño resumen de lo que conocemos.

La pantalla está organizada en filas de pixels y un carácter sencillo ocupa 32 bytes de memoria, ocho filas de cuatro bytes. El modelo binario de cada byte en la memoria de pantalla contiene información de dos pixels adyacentes horizontalmente.

Distribución de la memoria

Las filas de pixels están asociadas en grupos de ocho descendiendo por la pantalla. Hay 25 de estos grupos en ella —son las líneas que nosotros utilizaremos para situar el cursor del texto con LOCATE y para escribir con PRINT—. La dirección del comienzo de cada fila de pixels de uno de estos grupos está situada &800 bytes después de la correspondiente al comienzo de la fila anterior y cada grupo comienza &50 bytes después que el que tiene encima. La figura 1 nos muestra el Mapa de memoria del ángulo superior izquierdo de la pantalla.

Para dibujar un carácter de tamaño normal todo lo que se necesita es elaborar los datos requeridos y almacenarlos en ocho filas de 4 bytes en algún lugar de la RAM de pantalla. Para que nos sea más fácil, primero imprimiremos un carácter sencillo en una de estas 25 líneas en las que está dividida la pantalla.

Decodificación de un carácter

Para obtener los datos del carácter ejecutamos el programa 1 que

además los coloca a partir de la dirección &9002. Esta información será la que utilicen las rutinas en código máquina contenidas en este artículo.

El programa 2 es un listado en ensamblador de la rutina usada para dibujar el alien. Podemos hacer dos cosas para meter este programa en el **Amstrad**: utilizar un ensamblador para colocar los mnemónicos de las instrucciones o introducir los códigos hexadecimales, de uno en uno, usando el programa 3.

Para ver en acción la rutina de escritura del programa 2 tecleemos:

```
MODE 0: ??:?  
CALL &8000
```

El alien aparecerá en el ángulo superior izquierdo de la pantalla correspondiente a la dirección &C000.

Funcionamiento del programa 2

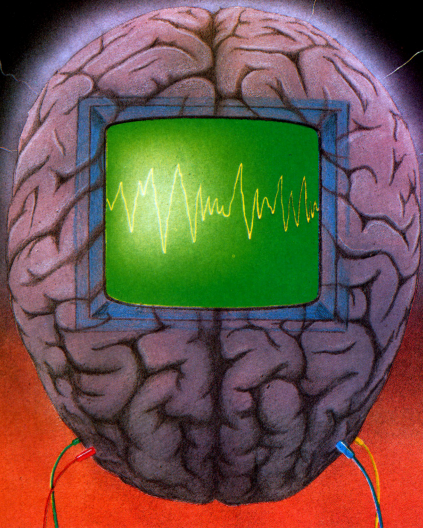
Primero se cargan el par de registros BC con el número de columnas y filas que estén almacenadas a partir de la dirección &9000, DE con la dirección de los datos, &9002 y HL con la dirección del sitio de la pantalla donde queremos dibujar, &C000. El registro B es el contador de bucle interno, el número de columnas y el C es el del bucle externo, el número de filas.

A continuación salvamos los valores de los contadores y de la dirección de la fila, después el bucle interno lo recorre recogiendo el dato contenido en la dirección a la que apunta DE y lo almacena en la dirección señalada por HL.

Incrementamos HL y DE para obtener el siguiente dato y la nueva dirección de la pantalla.

F. L. Frontón





¿Qué pasa con la altura? El alien tiene ocho pixels de alto y se comienza a dibujar en `&C0`. El bucle externo suma `&800` a la dirección contenida en HL cada vez, para obtener la del comienzo de la siguiente fila así que la dirección de la última es `&F800`. Supongamos que el carácter no esté exactamente en una línea de texto. Podríamos dibujarlo en el quinto pixel hacia abajo y así quedará en la mitad de una línea y a la mitad de la siguiente.

Aquí encontraremos problemas debido a que el carácter está repartido entre dos grupos de 8 filas. Cuando bajemos a la fila inferior del grupo, sumando `&800` a su dirección no obtendremos la de la fila superior del siguiente grupo de ocho.

Precauciones indispensables

En la figura 1 vemos que la dirección de la primera fila del próximo grupo es `&C050`, pero `&F800 + &800` nos da cero. Como una pareja de registros no puede contener números mayores que `&FFFF`, si superamos este valor se produce un desbordamiento, se activa el Carry flag y el par de registros comienza otra vez desde cero.

Lo que necesitamos hacer es controlar si se ha producido un desbordamiento y si hay que introducir después un factor de corrección `-&C050`. Si no se ha producido, todo irá correctamente.

El programa 4 es la misma rutina que antes, pero la hemos añadido que compruebe si se produce desbordamiento. Para ensayarlo escribiremos nuestro alien en el quinto pixel hacia abajo, en la dirección `&E000`.

Las cuatro primeras líneas siguen el tratamiento que hemos explicado en el programa anterior y comienzan en las direcciones `&E000`, `&E800`, `&F000` y `&F800`. Cuando sumemos `&800` a HL para calcular la dirección de comienzo de la quinta fila se producirá un desbordamiento. Esto activará el Carry flag así que se suma `&C050` para corregir el resultado. Observe que realizamos el control en cada fila y no solamente en la quinta.

Teclee de nuevo:

MODE 0: ?::?
CALL &8000

y veremos la mitad de nuestro alien dibujada en la primera línea y la otra mitad en la segunda.

Esta corta rutina imprimirá caracteres multicolores de cualquier tamaño y en el lugar de la pantalla que queramos. No importa si el alien está colocado exactamente en una de las 25 líneas de texto o está repartido entre dos o más, el código máquina lo controla y corrige cuando es necesario. ¡Inténtelo y lo verá! Cargue HL con cualquier valor a partir de &C000, ensamble la rutina de nuevo y llame a &8000.

Las ventajas del código máquina

Es difícil hacerse una idea de si la velocidad de la rutina en código máquina es muy superior a la del Basic cuando se imprime un solo carácter. El programa 5 llena totalmente la pantalla de aliens, y si consideramos que cada alien está formado por varios colores, su rapidez nos deja mudos de asombro.

El par de registros HL se usa para almacenar la dirección y el BC es el contador del bucle. Sus valores se guardan antes de dibujar el alien y se restauran después. La rutina de escritura propiamente dicha está separada del resto, la hemos puesto el LABEL (o etiqueta) PRINT por razones evidentes y se la llama como a una subrutina. El par de registros HL se usa para pasar la dirección donde vamos a escribir el carácter.

Creemos que todo lo anterior es suficiente para asimilarlo por esta semana. Pronto veremos cómo se consigue que las cosas se pongan en movimiento.

&C000	&C001	&C002	&C003	&C004	...
&C800	&C801	&C802	&C803	&C804	...
&D000	&D001	&D002	&D003	&D004	...
&D800	&D801	&D802	&D803	&D804	...
&E000	&E001	&E002	&E003	&E004	...
&E800	&E801	&E802	&E803	&E804	...
&F000	&F001	&F002	&F003	&F004	...
&F800	&F801	&F802	&F803	&F804	...
&C050	&C051	&C052	&C053	&C054	...
&C850	&C851	&C852	&C853	&C854	...
...

Figura 1: Mapa de memoria de la esquina superior izquierda de la pantalla en Mode 0

PROGRAMA I

```
10 REM PROGRAMA I
20 POKe #9000,B:POKE #9001,4
30 FOR I=0 TO 31
40 READ J:POKE #9002+I,J
50 NEXT
60 REM ALIEN
70 REM Fill=#Columnas#
80 DATA 4,12,12,8,72,148,104,132,72
90 DATA 156,108,132,28,60,60,44,28
100 DATA 32,16,44,4,48,48,8,8,0,0,4
110 DATA 4,0,0,8
```

PROGRAMA II

```
8000 org #8000
PROGRAMA II
8000 ED4B0090 LD BC, (#9000)
8004 110290 LD DE, #9002
8007 2100C0 LD HL, #C000
8008 BUCLE1: PUSH BC
800A C5 FUSH HL
800B E5 FUSH HL
800C BUCLE2: LD A, (DE)
800D 77 LD (HL), A
800E 23 INC HL
800F 13 INC DE
8010 10FA DJNZ BUCLE2
8012 E1 POP HL
8013 01000B LD BC, #800
8016 09 ADD HL, BC
8017 C1 POP BC
8018 0D DEC C
8019 20EF JR NZ, BUCLE1
801B C9 RET
801C END
```

ETIQUETAS

BUCLE1 800A BUCLE2 800C

PROGRAMA III

```
10 REM PROGRAMA III
20 a=#8000
30 PRINT HEX$(a): " "
40 INPUT h$
50 POKe a, VAL ("0x" + h$)
60 a=a+1:GOTO 30
```

PROGRAMA IV

```
8000 org #8000
PROGRAMA IV
8000 ED4B0090 LD BC, (#9000)
8004 110290 LD DE, #9002
8007 2100C0 LD HL, #E000
8008 BUCLE1: PUSH BC
800A C5 FUSH HL
800B E5 FUSH HL
800C BUCLE2: LD A, (DE)
800D 77 LD (HL), A
800E 23 INC HL
800F 13 INC DE
8010 10FA DJNZ BUCLE2
8012 E1 POP HL
8013 01000B LD BC, #800
8016 09 ADD HL, BC
8017 2004 JR NC, DE
8019 0150C0 LD BC, #C050
801C 09 ADD HL, BC
801D OR: POP BC
801E 0D DEC C
801F 20E9 JR NZ, BUCLE1
8021 C9 RET
8022 END
```

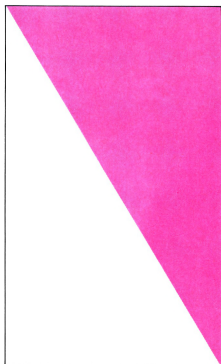
BUCLE1 800A BUCLE2 800C OR: 801D

PROGRAMA V

```
8000 ORR #8000
PROGRAMA V
LD HL, #C000
LD RC, 500
SIGLE: PUSH BC
PUSH BC
CALL PRINT
POP BC
POP HL
INC HL
INC HL
INC HL
INC HL
DEC BC
LD A, B
OR C
JR NZ, SIGLE
PRINT: LD RC, (#9000)
LD DE, #9002
BUCLE1: PUSH BC
PUSH HL
BUCLE2: LD A, (DE)
LD (HL), A
INC HL
INC DE
DJNZ BUCLE2
POP HL
LD RC, #800
ADD HL, BC
JR NC, DE
LD BC, #C050
ADD HL, BC
OR: POP BC
DEC C
JR NZ, BUCLE1
RET
END
```

ETIQUETAS

BUCLE1 801E BUCLE2 8020 OR: 801C
PRINT 8017 SIGLE 8006



COMPUTIQUE

Te da más

GARANTIA **AMSTRAD**
ESPAÑA

64.900 Ptas.

Amstrad 464 f.
verde



VENTA A PLAZOS HASTA 36 MESES

Al comprar tu Amstrad te regalamos

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



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

COMPUTIQUE

Servimos a tiendas
Abrimos sábados por la tarde

Embajadores, 90 Tfno. 2270980
28012 Madrid

Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

Muy señores míos:

Me voy a comprar un **Amstrad CPC 464** y la pregunta es: ¿Se le puede poner una unidad de disco al **Amstrad CPC 464** y ser compatibles los discos que salgan para el CPC 664?

Un saludo.

Francisco Fernández (Málaga)

Al CPC 464 puedes colocarle una unidad de disco. En principio, prácticamente la totalidad de software en disco, sobre todo bajo CP/M, correrá en ambos ordenadores, 464 y 664.

Me dirijo a ustedes para consultarles dos dudas. Una es referente a un juego publicado por esta revista en el número dos: «Egg Blitz». Este programa lo tecleé en mi ordenador (un CPC 464) y éste me indicaba «Memory full». Les ruego miren si esto es debido a alguna errata de publicación o a algún error de programación.

La otra duda es sobre la posibilidad de ampliación de memoria del **Amstrad CPC 464**. ¿Es posible? Si es así, ¿cómo?

¡Ah!, una terduda. ¿Podrían explicarme los esquemas de períodos de tono de los manuales del CPC 464? y ¿cómo se pueden generar las notas de la escala musical con «sound»?

Ahora ya sólo me resta despedirme. Gracias por escucharme.

Ramón J. Pachades

Respecto a tu primera cuestión, suponemos que intentaste cargar el programa en un 464 con unidad de disco (el mensaje MEMORY FULL así lo indica), porque de lo contrario funciona perfectamente.

Donde sí tendrás problemas con este programa es en un 664, por la razón anteriormente expuesta, el disco.

La posibilidad de ampliar la memoria del 464 es algo que todavía no está claro: teóricamente es posible, paginando memoria, pero que sepamos no hay nada comercial disponible.

Tengo en mente cambiar mi CPC-464 por un CPC 6128 y he oído rumores de que los programas en cinta comerciales e incluso los vuestros de la Serie Oro, no son compatibles con este ordenador.

¿Hay algo de cierto en esto?

Gracias.

Nota: Enhorabuena, os está saliendo muy bien la revista.

Mario Durán (Madrid)

Nuestros programas normalmente son compatibles con todos los **Amstrad** y, en caso contrario, se indica en la revista con toda claridad a qué ordenadores van dirigidos.

En el caso de soft comercial, existen programas que sólo funcionan en el 464.

Normalmente, los que corren en el 664, también lo hacen en el 6128.

Soy el casi arrepentido poseedor de un ordenador **Amstrad CPC 664**. Como no he podido obtener en casa comercial alguna, ninguno de los periféricos de este ordenador ni ninguno de los programas de gestión en disco que **Amstrad** insiste en anunciar, opté por entretener la ya larga espera aprovechando los juegos que tiene a su disposición el CPC 464. Utilicé para ello un cable de conexión al magnetófono, hecho por profesionales a propósito, ya que el cable de la casa tampoco existe. La decepción ha superado todas las previsiones.

La mayor parte de los juegos probados no se graban. En estos casos se exhibe con unanimidad el mensaje: «Memory full in ...» (En lugar de la línea de puntos aparece un número.) En el libro de instrucciones no se explica nada. ¿Saben ustedes qué significa este mensaje? ¿Es posible superar este problema? ¿Hasta qué punto son compatibles los juegos del CPC 464 en el CPC 664?

Esteban Padros (Barcelona)

Que nosotros sepamos, Indescomp vende un cable de cassette para permitir que el 664 pueda aprovecharse del software existente en cinta.

Por otra parte, algunos de los programas que existen para el 464 no son compatibles para el 664. Normalmente, esto sucede con los primeros programas que salieron para **Amstrad**, cuando el 664 aún no estaba disponible y se irá resolviendo con el tiempo.

Una casa de software respetuosa con sus clientes, lo menos que puede hacer es indicar en la cinta con qué máquina es compatible su programa. Como desgraciadamente no es así, la única solución es exigir que prueben el programa que piensas adquirir en un 664.

NL **New Line**

GABINETE DE INFORMÁTICA

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

ZURBANO, 4 ☎ 410 47 63
28010 MADRID

K-BITS

Ordenadores personales y de gestión
Aplicaciones para arquitectura

- Amstrad
- Sinclair
- Commodore
- Philips
- Canon
- Spectravideo
- Dragón
- Impresoras
- Monitores
- Periféricos
- Libros
- Revistas

Servimos a provincias

C/ Barquillo, 15.
Teléfono: (91) 232 57 37. Madrid.

¿Cómo conseguir que una serie de datos, sean utilizados por varios programas sin tener que teclearlos en cada uno de ellos? La solución, es crear un fichero de datos, independiente del programa, que se almacena en cinta o disco y del cual podemos recuperarlos cuando nos interesa.

En esta ocasión Análisis estudia como grabar datos en un archivo en cinta o disco y como leerlos después.

10,20 Los REM de costumbre.

30 Escribe un mensaje en pantalla, que nos indica que los datos se están grabando.

Su inclusión, se debe a que en los sistemas de disco esta grabación, se realiza sin mensaje alguno y se podría pensar que no está ocurriendo nada.

En los sistemas de cinta, es necesario por la aparición del mensaje Press REC and PLAY then any key.

50 Efectúa la apertura de una vía de transmisión de datos desde el programa a la cinta o al disco, permitiendo que éstos sean almacenados en un fichero, en este caso le hemos dado el nombre de archivo.

60-90 Ciclo FOR...NEXT, que se encarga de leer los datos y mandarlos al archivo.

Una vez comprendido el programa sería interesante, hacer que éste se repita dos o tres veces y observar los resultados.

70 Lee los datos almacenados en las variables; nombre\$ y número.

En nuestro caso la información viene de una línea DATA, pero esto podría realizarse también desde otros cauces, como por ejemplo el teclado.

80 WRITE #9 hace que la información almacenada en los anteriores variables, sea enviada a disco o cassette.

La primera vez que rota el ciclo les corresponden a Pepe y 1, la segunda; a Luis y 2, la tercera; a Carlos y 3.

90 El ciclo concluye después de tres vueltas, leyendo toda la información de la línea DATA.

100 Envía una cadena que será almacenada en el fichero de salida, ésta indica el final del archivo.

110 Cierra el canal de salida.

130-140 Provoca una pausa entre el almacenamiento y la lectura de datos.

En este momento, los datos están almacenados en un fichero llamado archivo.

160-230 Lee los datos del fichero y los copia en pantalla.

160 Crea una vía de comunicación entre el programa y el sistema de almacenamiento, para leer el fichero archivo. En el caso de utilizar cassette, rebobinar la cinta.

170-200 Conforman el ciclo FOR...NEXT, que efectúa la lectura de los datos.

180 INPUT #9 toma información de disco o cassette, almacenándola en las variables nombre\$, número.

190 Visualiza en pantalla los datos tomados del archivo.

210 Efectúa la lectura del final del fichero y la almacena en final\$.

220 Escribe el mensaje «Final del archivo».

230 Cierra la vía de comunicación establecida en la línea.

160

250 Contiene los datos que vamos a transferir al fichero.



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!

AMSTRAD Semanal comunica a todos sus lectores la apertura de una nueva sección dedicada a recoger las mejores ideas que exploten al máximo las posibilidades del ordenador, materializadas en programas claros y cortos (*máximo 25 líneas*). Los mejores de entre todos ellos serán publicados con el nombre de su autor en la revista, recibiendo como premio, gratuitamente en su domicilio los cuatro primeros números de nuestra cinta mensual. Los programas enviados deberán incluir:

- **Cinta de cassette con el programa o programas grabados.**
- **Explicación detallada del funcionamiento y propósito del programa, mecanografiada a 2 espacios o con letra clara.**

Es imprescindible indicar en el sobre claramente: **AMSTRAD IDEAS.**

La dirección es:
Editorial Hobby Press, S. A.
La Granja, s/n.
Polígono Industrial de Alcobendas.
Madrid.

Amstrad Ideas

Vendo Commodore 64. Incluyo Datassette C2N, joystick Quick Shot II, guía del usuario, Curso de Basic (2 libros), numerosas revistas especializadas, y un excelente surtido de juegos (*One on One, Decathlon, Beach Head, etc.*). Todo ello por 49.000 ptas.

Interesados llamar, de 14 a 16,30 al (972) 31 53 86. O bien, escribir a: Miquel Gasull Buenaventura, Alvaro de Bazán, 2. Palamós (*Girona*).

Vendo «Joystick» Quickshot I (sin estrenar), «Codename Mats», más «Gremlins» (en castellano), más «Fighter Pilot», más «Decathlon» o «World Cup Football», todo por 3.500 ptas. Gastos de envío por mi cuenta. Instrucciones completas y en castellano. Escribir a Manuel Angel Sánchez Costa. Amílcar Barca, 29 - 2.º D. 11009-Cádiz.

MICRO-1

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

Regalo de bienvenida al club de Software MICRO-1: 1 bolígrafo de acero con reloj de cuarzo incorporado.

match day	1.975	dragontorc	1.875	raid o moscow	1.975
southern belle	1.975	dummy run	1.975	jump jet	2.595
death pit	1.975	combat lynx	1.925	beach head	2.395
decathlon	1.975	exploding fist	2.095	basketball	1.950
gremlins	1.975	rocky	1.925	highway encout	1.975

¡Increíbles precios para tu AMSTRAD 464 y 128!
(¡llámanos, y te asombrarás)

Lápis Optico DKTRONICS 4.850

Tapa de metacrilato transparente para tu
AMSTRAD 2.450

Impresoras: ¡20% Dto. sobre P.V.P.!

Diskette 3": 1.050
Cinta-15: 85

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

3-D VOICE CHESS

Ajedrez tridimensional con voz en castellano
Amstrad CPC 464, CPC 664 y CPC 6128

P.V.P.
2.100.- (cinta)
3.300.- (disco)



PUBLISHED
DEEP THOUGHT
SOFTWARE

DISTRIBUTED

cp software



- Brillantes gráficos 3-D (relieff de color)
- 7 niveles de juego (de principiantes a expertos).
- Rápidas respuestas desde 5 segundos

Producido en exclusiva para España por:

ACE

Actividades Comerciales Electrónicas, S.A.
Tarragona. 100 - Tel. 325 10 58* 08015 Barcelona. Telex 93133 ACE E E

YA DISPONIBLE EN



Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

Rocky



Simulación tridimensional de un combate de boxeo. Lucha contra CIMBEL-LIN, TED MATA-RE, JANSEN SINO y FIGHTER BULL para alzarte con el título mundial.

SPECTRUM 48 K 1.800 ptas.
AMSTRAD CPC 464, 2.100 ptas.



Servimos a tiendas y almacenes telf. (91) 447 34 10 Telex 22542 JAGA E

Pedidos contrareembolso (Sin G/E y adhesivos regalo) a:

"MANSION" DINAMIC: C/ Tilos, 2,21 MONTEPRINCIPE, BOADILLA DEL MONTE, MADRID TELF. (91) 715 00 67