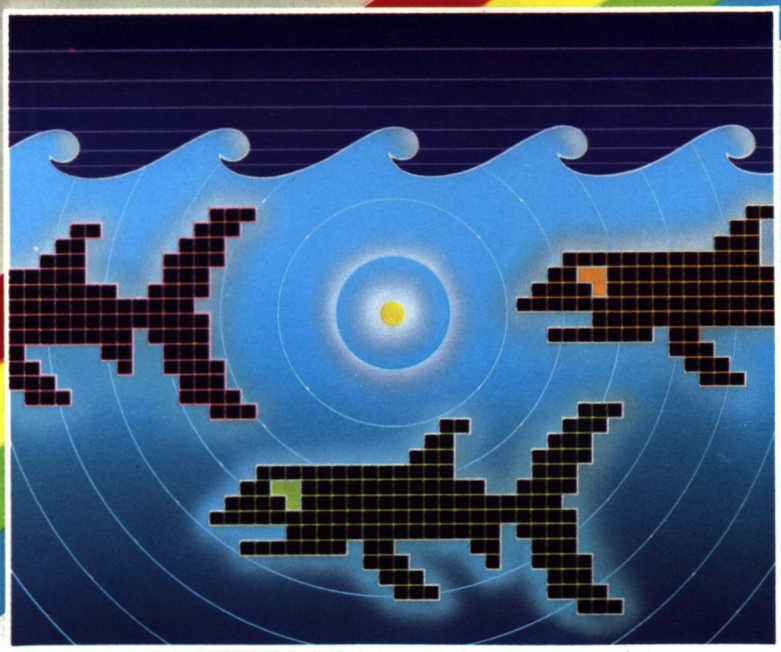
JUEGOS DINAMICOS PARA EL ZX SPECTRUM

T. HARTNELL



JUEGOS DINAMICOS PARA EL ZX SPECTRUM

Editorial Gustavo Gili, S. A.

08029 Barcelona Rosellón, 87-89. Tel. 322 81 61
28006 Madrid Alcántara, 21. Tel. 401 17 02
1064 Buenos Aires Cochabamba, 154-158. Tel. 361 99 98
03100 México, D.F. Amores, 2027. Tels. 524 03 81 y 524 01 35
Bogotá Diagonal 45 N.º 16 B-11. Tel. 245 67 60
Santiago de Chile Santa Victoria, 151. Tel. 222 45 67

JUEGOS DINAMICOS PARA EL ZX SPECTRUM

T. HARTNELL

GG

Título original
Dynamic Games for the ZX Spectrum
Publicada en 1983 por Sinclair Browne Ltd.
10 Archway Close, London N19 3TD.

Versión castellana de Jordi Creixell Sans, Ingeniero Industrial
Programación de Sun Tarrés
Escaneado por Miquel Coma Rivas

Printed in Spain
ISBN: 84-252-1205-7
Depósito legal: B. 5.807-85
FOTOCOMPOSICION: TECFA, S.A., Barcelona
IMPRESION: HUROPE, S.A. - Recaredo, 2 - Barcelona

Indice

Introducción

En muchos aspectos, un ordenador es el contrincante ideal para jugar.

Incapaz de vanagloriarse cuando gana, elegante perdedor, es incansable, paciente y siempre dispuesto. A pesar de que no puede dar este sentido humano, que puede ser una gran parte del placer de jugar con otras personas, el ordenador parece ser el compañero perfecto con el que pasar las horas.

Sin importar porqué «pensó» que compraba su Spectrum, nosotros le retamos a gastar al menos parte de su tiempo de computador jugando: aquí es donde viene este libro. Aquí encontrará una gran variedad de juegos, desde juegos de tablero como AJEDREZ y DAMAS, acción en programas como JOGGER y NIVEL CINCO, a nuestro gran juego de aventuras: LA VENGANZA DEL CASTILLO ENCANTANDO. Hemos propuesto interesantes programas de gran mérito, así como otros que le aportarán sugerencias e ideas para aplicar a sus propios trabajos.

Para ayudarle a obtener el máximo provecho de este libro, hemos escrito instrucciones detalladas de los juegos. En la mayoría de los casos la introducción presenta el programa línea por línea, explicando los trucos que los programadores han usado sugiriéndole cómo pueden ser aplicados a sus propios programas y juegos. Algunos programas también contienen indicaciones de cómo modificarlos según sus deseos, mejorándolos y desarrollándolos como quiera.

La mayoría de los programas de este libro fueron escritos por Tim Rogers, Paul Toland y Tim Hartnell. Rogers y Toland son estudiantes (Rogers en Londres y Toland en Belfast) y merecen ser felicitados por la calidad y originalidad de su trabajo, que se refleja en los programas que hemos seleccionado en este libro. En la página 169 puede encontrar una lista completa de los autores de los programas.

Es hora de empezar a entrar código a su Spectrum para que pueda disfrutar estos juegos que merecen realmente el adjetivo de «dinámicos».

Tim Hartnell
Londres, marzo de 1983

Juegos de acción

Nivel cinco

Tranquilícese, tómeselo con calma, porque aquí viene NIVEL CINCO, una rápida adaptación de un famoso juego de acción.

El área de juego está constituida por cinco niveles conectados por una serie de escaleras. Cuatro monstruos arrancan del nivel más bajo y se desplazan hacia la parte superior. Su trabajo es contenerlos en la prisión situada debajo del primer nivel. El problema principal es que, ya que estos monstruos se desplazan hacia arriba, la única manera de mantenerlos abajo es cavar agujeros en las plantas para que caigan a través de ellos. Puede cavar un agujero pulsando el «Ø», siempre y cuando no haya otro agujero en la pantalla. Cada agujero cavado se mantendrá un corto período de tiempo.

Usted controla sus movimientos usando las siguientes teclas: «5» (izquierda), «8» (derecha), «6» (arriba), «7» (abajo). El juego termina cuando se consigue encarcelar a todos los monstruos. En este punto será informado del tiempo que ha tardado en conseguirlo. El juego termina prematuramente si usted es saltado por un monstruo. Como es habitual en programas para Spectrum, la calidad de los gráficos sólo es sugerida por la impresión de muestra. Encontrará este programa muy rápido y capaz de crear dependencia. El autor del programa, Paul Toland, ha conseguido una puntuación de 6Ø jifis. Le desafiamos a mejorarla.

La línea 1 muestra qué gráfico hay en cada tecla, con los ladrillos en «A», secciones de escalera en «B», usted (el que corre), en «C» y los apuestos monstruos en «E». La línea 25 envía la acción a la subrutina de la línea 91ØØ, después de definir la tinta (INK) blanca y el papel (PAPER) y borde (BORDER) azules. En esta subrutina se imprimen las instrucciones en la pantalla. La línea 913Ø espera hasta que usted pulsa una tecla cualquiera y entonces borra la pantalla. El resto de las instrucciones aparecen junto con el mensaje «¡Buena suerte!». La instrucción RETURN de la línea 916Ø envía la acción a la línea 3Ø, donde se llama la subrutina de la línea 9ØØØ. 9ØØØ empieza restaurando (RESTORE) el puntero DATA a sí mismo (lo cual tiene el efecto de restaurarlo en el próximo elemento de DATA después del número que sigue a la palabra RESTORE). Esto asegura que cuando funcionan

los bucles I y J (líneas 9ØØ1 a 9Ø4Ø), los números leídos para ser colocados (POKE) en los gráficos son los de las líneas 9Ø5Ø, 9Ø6Ø, 9Ø7Ø, 9Ø8Ø y 9Ø9Ø, y no desde la línea 1Ø7Ø.

Una vez se han definido los gráficos, el programa va (mediante la segunda instrucción de la línea 3Ø) a la subrutina de la línea 7ØØ, que imprime los cinco pasadizos, uno por cada nivel. La línea 91Ø se restaura (RESTORE) a sí misma, de la misma forma que el RESTORE 9ØØØ anteriormente analizado, moviendo el puntero DATA al primer elemento de DATA que sigue al RESTORE. En este caso es la línea 1Ø3Ø la que contiene la información usada en los bucles I y J (1ØØØ a 1Ø5Ø) para poner las escaleras en su sitio. La línea 1Ø55 borra la última línea de la pantalla y el programa vuelve a la línea 4Ø, donde empieza la diversión.

Y y X, las coordenadas de la figura (Vd.), se igualan a 14 y 3 respectivamente en la línea 4Ø. M, que cuenta los monstruos, se iguala a 4 en la línea 5Ø, y se crea un vector en la segunda parte de esta línea para contener a los monstruos antes de que arranquen con el bucle de la línea 6Ø. El vector H (que contiene su agujero) se DIMensiona en la línea 7Ø, y la tinta (INK) y el papel (PAPER) se ponen a 8 en la línea 8Ø, justo antes de que usted sea impreso en la pantalla en la línea 9Ø.

La variable TIEMPO (recuerde que la mejor puntuación de Paul es 6Ø) se iguala a Ø en la línea 1ØØ y se va incrementando en la línea 395, donde, además, se imprime en la pantalla. El título del programa se coloca en la parte superior de la pantalla (11Ø) , dejando un espacio para que aparezca el TIEMPO. La bandera LIBRE se iguala a Ø en la línea 19Ø.

El bucle I comprueba la posición y controla la acción de los monstruos. Funciona de la línea 2ØØ a la 39Ø y contiene todas las acciones importantes del programa: sigue la pista de los monstruos (2Ø5), los reimprime (21Ø y 22Ø), controla sus inmediaciones (225 y 23Ø) y sus movimientos pasando a la línea 43Ø si le pasan o a la 28Ø si están subiendo. La coordenada horizontal se incrementa en a línea 24Ø, y la línea siguiente invierte la dirección del monstruo si alcanza los extremos de un pasadizo. ¡Monstruo fuera! puede gritar si se cuela por un agujero (línea 27Ø), antes de ser reimpreso en su nueva posición en la línea 28Ø.

La línea 295 rellena un agujero después de 4Ø ciclos de programa desde que ha sido creado. La línea 3ØØ lee el teclado, saltando a la siguiente rutina si no hay ninguna tecla pulsada en este momento. Si lee un «Ø», sabe que usted quiere cavar un agujero. Si H (3) es igual a cero, el ordenador sabe que no hay otros agujeros en la pantalla y define las coordenadas del agujero en la posición en la que usted se encuentra, inprimiendo un agujero en esta posición y saltando a la línea 39Ø.

El ordenador sabe que si usted está pulsando el cero en la línea 315, no pulsará «5», «6», «7» u «8» durante las próximas líneas, por tanto las salta para conseguir una máxima velocidad de ejecución. La línea 317 comprueba si la tecla pulsada está entre «4» y «9», si no está en este intervalo, salta a la línea 39Ø. La próxima línea le coloca en posición, y la línea 325 asigna un 15 a su dirección de ATTRibuto para hacerlo desaparecer de su actual situación, antes de reimprimirlo en la línea 39Ø (los ATTRibutos se almacenan después del campo de pantalla, desde la dirección 22528).

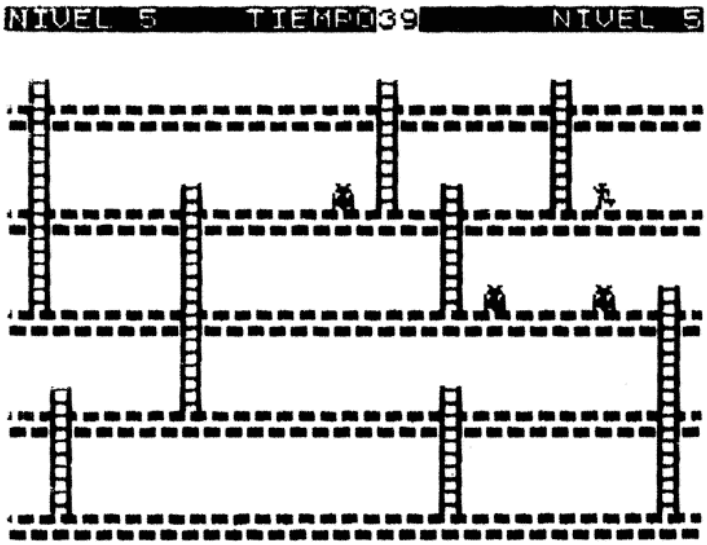
La línea 327 comprueba si usted está en un pasadizo y, si descubre que no, salta a la línea 35Ø para ver si desea moverse arriba o abajo por una escalera. N, el valor usado para almacenar el ATTRibuto de su posición, es igualado al ATTRibuto actual en 37Ø, antes de que usted sea impreso en 38Ø. Se alcanza 33Ø si usted está en un pasadizo, donde se puede mover según sus deseos mediante las teclas «5» a «8». La línea 34Ø le para si trata de rebasar los límites de la pantalla. Se reinicia el ciclo completo en 39Ø.

Después de ejecutar el bucle I cuatro veces, la variable LIBRE se examina para ver si han sido capturados todos los monstruos. Si no es así, vuelve a 19Ø, manteniendo en marcha el juego. La línea 41Ø indica que los ha cogido a todos y 43Ø que le han cogido ellos a usted. Se le ofrece una nueva partida en 5ØØ y el juego se reanuda si pulsa cualquier tecla que no sea «N» o «n».

```
    1 REM UDG ▘=A  ▙=B  ▚=C  ▟=E
    2 REM
   10 REM  NIVEL CINCO  P.TOLAND
   20 INK 7: PAPER 1: BORDER 1: C
LS
   25 GO SUB 9100
   30 GO SUB 9000: GO SUB 700
   40 LET X=14: LET Y=3: LET C=1
   50 LET M=4: DIM M(N,3)
   60 FOR I=1 TO M: LET M(I,1)=19
: LET M(I,2)=I*3: LET M(I,3)=1-I
NT (RND+.5)*2: PRINT AT 19,I*3;"
▟": NEXT I
   70 DIM H(3): LET H(1)=0: LET H
(2)=0: LET H(3)=0
   80 INK 8: PAPER 8
   90 PRINT AT Y,X; INK 5;"▚"
  100 LET N=15: LET TIEMPO=0
  110 PRINT AT 0,0; INVERSE 1;"NI
VEL 5     TIEMPO=        NIVEL 5"
  190 LET LIBRE=0
  200 FOR I=1 TO 4
  205 IF M(I,1)=21 THEN GO TO 295
  207 LET LIBRE=1
  210 LET D=M(I,1): LET A=M(I,2)
  220 PRINT AT D,A; OVER 1;"▟"
  225 IF ATTR (D-1,A)=13 THEN GO
TO 430
```

```
 230 IF ATTR (D-1,A)=14 THEN LET
 D=D-1: GO TO 280
 240 LET A=A+M(I,3)
 250 IF A=0 OR A=31 THEN LET M(I
;3)=-M(I,3)
 260 IF ATTR (D,A)=13 THEN GO TO
 430
 270 IF ATTR (D+1,A)=15 THEN LET
 D=D+4: BEEP .1,30: IF D>21 THEN
 LET D=21
 280 PRINT AT D,A; OVER 1;"■"
 290 LET M(I,1)=D: LET M(I,2)=A
 295 IF M(3)>0 THEN LET M(3)=M(3
)+1: IF M(3)=40 THEN PRINT INK 2
; PAPER 7;AT M(1),M(2);"■": LET
M(3)=0
 300 LET I$=INKEY$
 310 IF I$="" THEN GO TO 390
 315 IF I$="0" AND M(3)=0 AND AT
TR (Y+1,X)=58 THEN LET M(1)=Y+1:
 LET M(2)=X: LET M(3)=1: PRINT P
APER 1; INK 7;AT M(1),M(2);" ":
GO TO 390
 317 IF I$<"5" OR I$>"8" THEN GO
 TO 390
 320 PRINT AT Y,X; OVER 1;"■"
 325 POKE 22528+Y*32+X,N
 327 IF (Y+1)/4<>INT ((Y+1)/4) T
HEN GO TO 350
 330 LET X=X+(I$="8")-(I$="5")
 340 LET X=X+(X<0)-(X>31)
 350 IF I$="7" AND ATTR (Y-1,X)=
14 THEN LET Y=Y-1
 360 IF I$="6" AND ATTR (Y+1,X)=
14 THEN LET Y=Y+1
 370 LET N=ATTR (Y,X)
 380 PRINT AT Y,X; OVER 1; INK 5
;"■"
 390 NEXT I
 395 LET TIEMPO=TIEMPO+1: PRINT
AT 0,17;TIEMPO
 400 IF LIBRE THEN GO TO 190
 410 PRINT OVER 1;AT 9,1;"TODOS
LOS MONSTRUOS CAPTURADOS": BEEP
3,30
 420 GO TO 440
 430 PRINT AT Y,X;"■"; OVER 1;AT
 9,1;"LOS MONSTRUOS LE HAN ALCAN
ZADO": BEEP 3,-20
 440 PRINT AT 13,1;"DESPUES DE U
N TIEMPO DE ";TIEMPO
 500 INPUT "OTRO JUEGO ?";A$
 510 IF A$<>"N" AND A$<>"n" THEN
 RUN
 699 STOP
 700 FOR I=1 TO 5
 800 PRINT INK 2; PAPER 7;AT I*4
,0;"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■■■"
 900 NEXT I
```

```
 910 RESTORE 910
1000 FOR J=1 TO 8
1010 READ X,Y,L
1020 FOR I=0 TO L
1030 PRINT INK 6;AT Y+I,X;"H"
1040 NEXT I
1050 NEXT J
1055 PRINT AT 21,0; PAPER 0,,
1060 RETURN
1070 DATA 1,3,8,25,3,4,8,7,8,20,
7,4,30,11,8,2,15,4,17,3,4,20,15,
4
9999 STOP
9000 RESTORE 9000
9001 FOR I=0 TO 4
9010 FOR J=0 TO 7
9020 READ N: POKE USR "A"+I*8+J,
N
9030 NEXT J
9040 NEXT I
9045 RETURN
9050 DATA 207,207,207,0,0,126,12
6,126
9060 DATA 195,255,195,195,195,25
5,195,195
9070 DATA 56,144,124,20,48,47,34
,96
9080 DATA 28,9,62,40,12,116,68,6
9090 DATA 102,60,126,153,187,255
,255,153
9100 PRINT INVERSE 1;"   NIVEL 5
        NIVEL 5
9110 PRINT ,,"Hay cuatro monstru
os moviendose  por la pantalla,
intentando     llegar arriba.  H
ay 5 niveles   conectados por es
caleras y su   trabajo es encarc
elarlos en el  sotano situado de
bajo del       primer nivel."
9115 PRINT "Su unico medio para
conseguirlo  es cavar agujeros e
n las          plantas para que lo
s monstruos    caigan por ellos. L
os agujeros    permaneceran en pan
talla un       tiempo y luego desa
pareceran.     Solo puede haber un
 agujero a     la vez."
9120 PRINT "Vd. morira si es alc
anzado por     un monstruo. El ret
o es           encarcelar los cuat
ro monstruos   en el minimo tiempo
.";;;"         "; FLASH 1;"Pulse cual
quier tecla"
9130 PAUSE 0: CLS
9140 PRINT ;;;"Use las teclas 5,
 6, 7 y 8 para moverse y el 0 pa
ra cavar un    agujero.";;;"BUE
NA SUERTE! .... la necesitara"
9150 PAUSE 200: CLS
9160 RETURN
```



Circo

CIRCO es un juego de acción algo parecido a «Breakout», pero con un escenario más interesante y más difícil para el jugador. Usted debe hacer estallar unos globos situados en la parte superior de la pantalla mediante los saltos de dos payasos sobre un trampolín, situado en la parte inferior. Usted controla el trampolín con las teclas «5» y «8». Pulsando la tecla «1» el payaso del trampolín cambia de lado. El payaso que cae debe tocar su lado del trampolín para proyectar al otro hacia los globos. Lo ideal es aterrizar en el extremo del trampolín. Si el payaso aterriza cerca del centro del trampolín, el otro payaso no subirá suficientemente para tocar los globos.

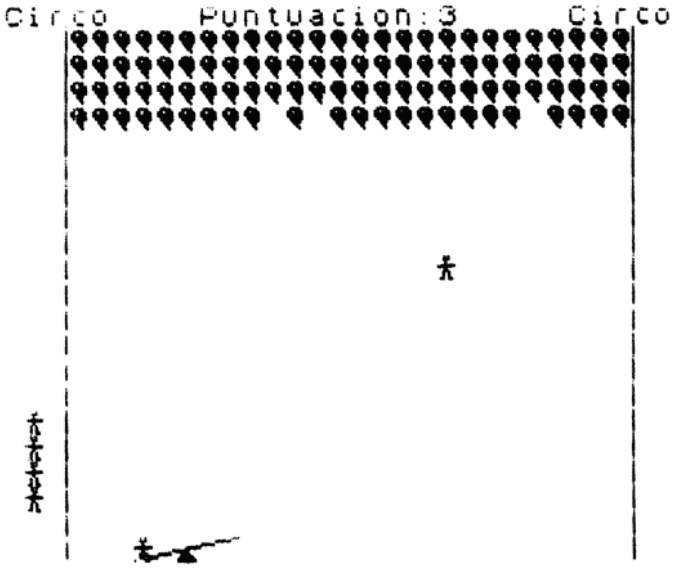
El juego es tan difícil que hemos debido incluir payasos adicionales para que el jugador pueda hacer estallar un número razonable de globos.

Cuando ejecute el programa, verá las instrucciones de la siguiente forma:

```
     CIRCO:    INSTRUCCIONES.
El circo ha venido a la ciudad
 y la atraccion estrella son los
 payasos. Dos payasos saltan
 desde un trampolin hasta unos
 globos en  el aire.
El payaso que cae no debe tocar
 el suelo ni el lado incorrecto
 del trampolin. Si cae cerca  de
```

```
la mitad del trampolin, el otro
payaso no subira suficiente
para tocar los globos.
El circo tiene solo seis payasos
por tanto tenga cuidado!
Vd. controla el trampolin con
las teclas 5 y 8. Puede cambiar
de lado al payaso del trampolin
pulsando '1'. Pulse 'N' para
parar y cualquier otra tecla
para empezar
```

Y, una vez iniciado el juego, la pantalla queda de esta forma:



El listado del programa muestra en la línea 5 los gráficos definidos por el usuario, con «A» para el pivote del trampolín, «B» para el payaso y «C» para el globo. Después de generar un número aleatorio en 2Ø, el programa va a la línea 8Ø donde se activa OVER, la tinta (INK) se hace negra, el papel (PAPER) blanco y el borde (BORDER) rojo, antes de borrar la pantalla para establecer el color del papel.

Desde aquí el programa salta a la subrutina de la línea 9ØØ, donde se llaman otras dos subrutinas (11ØØ y 1Ø2Ø). La primera imprime el título y las instrucciones en la pantalla y la segunda define los gráficos. Al volver de estas subrutinas, se dibuja el marco (líneas 9Ø2 a 9Ø5) y se colocan los globos en su lugar mediante el bucle I (líneas 91Ø a 93Ø). Este mismo bucle dibuja los payasos de reserva a la izquierda de la pantalla.

En las próximas dos líneas se definen una serie de variables. SC almacena la puntuación, o sea el número de globos que ha hecho estallar (18Ø). SS es la posición horizontal de la base del trampolín que se usa en la línea 5Ø para colocarlo y para dibujar el payaso. Observe que las líneas 4Ø y 6Ø son una subrutina que es solicitada repetidamente a lo largo del programa.

M cuenta los payasos en juego y se incrementa en la línea 3ØØ, a donde se llega desde la línea 23Ø. El vector B$ se utiliza para imprimir 26 espacios en blanco en la línea 33Ø cuando un payaso ha caído mal. OD y AN controlan la posición del payaso del trampolín. Cambiando estas variables por sus negativas (1Ø7) se invierte la posición del trampolín. OD y AN se usan en las líneas 5Ø, 26Ø y 28Ø.

La variable CA se usa para detectar cuándo debe cambiarse AN (véase líneas 95 y 27Ø) y HL se determina en 25Ø y se usa en 16Ø para detectar cuándo debe invertirse el sentido de movimiento del payaso en la coordenada Y (desplazamiento vertical) (véase líneas 16Ø y 14Ø). X es la coordenada inicial del payaso usada en la línea 13Ø e incrementada en la 14Ø por medio de A, que se cambia en 28Ø para invertir el movimiento horizontal del payaso. Y (determinado en 95Ø) es la coordenada vertical inicial. Este valor empieza en algún punto entre 5 y 8.

La línea 1ØØ lee el teclado detectando «5» y el «8» para cambiar la posición de la base del trampolín. Este valor se supervisa en la línea siguiente (1Ø5) para asegurarse que usted no intenta mover el trampolín más allá de los límites de la pantalla.

```
   5 REM U.D.G.  ▲=A 🤸=B 🎈=C
   7 REM
  10 REM CIRCO.  P.TOLAND.
  15 REM ----------------
  20 RANDOMIZE
  40 GO TO 80
  50 PRINT AT 21,ss;"▲";AT 21,ss
+od;"🤸"
  60 PLOT ss*8-16,7*(an=1): DRAW
 38,7*-an: RETURN
  80 OVER 1: INK 0: PAPER 7: BOR
DER 2: CLS
  85 GO SUB 900: GO SUB 60
  90 GO SUB 50
  95 IF ca THEN LET an=-an: LET
ca=0
 100 LET ss=ss+(INKEY$="8")-(INK
EY$="5")
 105 LET ss=ss+(ss=4)-(ss=27)
 107 IF INKEY$="1" THEN LET an=-
an: LET od=-od
 110 GO SUB 50
 130 PRINT AT y,x;"🤸"
 140 LET x=x+a: LET y=y+d
```

```
 150 IF x=3 OR x=28 THEN LET a=-
a: BEEP .01,1
 160 IF y<hl THEN LET d=-d
 165 LET no=ATTR (y,x)
 167 PRINT AT y,x;"▲"

 170 IF no=56 THEN GO TO 210
 180 LET sc=sc+61-no: BEEP .01,1
0
 185 PRINT AT y,x; OVER 0;"▲";AT
 0,20;sc
 190 LET d=-d
 195 IF sc=260 THEN GO TO 340
 200 GO TO 90
 210 IF y<21 THEN GO TO 90
 220 LET dis=x-ss: BEEP .01,5
 230 IF dis<-2 OR dis>2 OR SGN d
is<>-an THEN GO TO 310
 250 LET hl=11-ABS dis*5
 260 LET x=ss+od
 270 LET od=dis: LET ca=1
 280 LET a=SGN od: LET d=-1
 290 GO TO 90
 310 LET m=m+1
 320 BEEP .2,-10: BEEP .7,-20
 330 IF m<5 THEN PRINT AT 21,3;
OVER 0;b$: GO SUB 950: GO SUB 50
: GO TO 90
 335 BEEP 1,-40: RUN
 340 PRINT AT 9,4; FLASH 1;"LO H
ICISTE!!!PUNTUACION MAXIMA"
 345 BEEP 1,20: BEEP 2,30
 350 PAUSE 1: PAUSE 100
 360 RUN
 900 GO SUB 1100: GO SUB 1020
 902 PLOT 23,0: DRAW 0,167
 905 PLOT 232,0: DRAW 0,167
 910 FOR i=1 TO 4
 920 PRINT AT i,3; INK i;"♦♦♦♦♦♦
♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦";
 925 PRINT AT 15+i,1;"▲"
 930 NEXT i
 940 PRINT AT 0,0;"Circo     Punt
uacion:0      Circo"
 945 LET sc=0: LET ss=16: LET m=
0: DIM b$(25)
 950 LET od=-2: LET ca=0: LET hl
=0: LET x=3: LET y=5+INT (RND*4)
: LET a=1: LET d=1: LET an=-1
 955 PRINT AT 15+m,1; OVER 0;" "
 957 FOR i=15+m TO y STEP -1
 960 PRINT AT i,1;"▲"
 970 BEEP .05,21-i
 980 PRINT AT i,1;"▲"
 990 NEXT i
1000 PRINT AT y,x;"▲"
1010 RETURN
1020 RESTORE 1070
1030 FOR i=0 TO 23
```

```
1040 READ n: POKE USR "a"+i,n
1050 NEXT i
1060 RETURN
1070 DATA 0,0,0,0,16,56,124,254
1080 DATA 56,56,16,254,16,40,40,
108
1090 DATA 0,60,110,126,62,28,8,4
1100 PRINT "     CIRCO:    INSTRUC
CIONES."
1110 PRINT "El circo ha venido a
 la ciudad    y la atraccion estr
ella son los payasos. Dos payaso
s saltan     desde un trampolin
hasta unos   globos en  el aire.
"
1115 PRINT "El payaso que cae no
 debe tocar  el suelo ni el lado
 incorrecto  del trampolin. Si c
ae cerca  de la mitad del trampo
lin, el otro payaso no subira su
ficiente     para tocar los glob
os.          El circo tiene solo
seis payasos por tanto tenga cui
dado!"
1120 PRINT '"Vd. controla el tra
mpolin con     las teclas 5 y 8.
Puede cambiar de lado al payaso
del trampolin pulsando '1'. Puls
e 'N' para     parar y cualquier
otra tecla     para empezar"
1130 IF INKEY$="" THEN GO TO 113
0
1140 IF INKEY$="n" OR INKEY$="N"
 THEN STOP
1150 CLS
1160 RETURN
```

Jogger

Usted debe ayudar a cinco cansados practicantes del «jogging» a cruzar una concurrida autopista de seis carriles. Uno a uno los joggers ponen su vida en sus manos para que los guíe a través del tráfico usando las teclas «5» (izquierda), «8» (derecha), «6» (arriba) y «7» (abajo). Cualquier error es el fin del juego.

No sólo debe luchar contra el tráfico, el tiempo también es su enemigo. Cada vez que consiga pasar a los cinco joggers, el ordenador le informará del tiempo que ha necesitado para conseguirlo (medido en una conocida unidad: «latidos de corazón»). Después de un corto período de tiempo, para permitirle recuperar la respiración, otros cinco joggers se alinearán en la carretera, esperando su asistencia.

Como puede observar en las dos copias del juego, siempre hay cinco joggers en la pantalla, ya sea esperando cruzar, cruzando o ya

al otro lado de la autopista. Será informado de cuántos joggers quedan por ayudar (por si usted no supiera contarlos) y también de cuántos han hecho ya el trayecto. El tiempo, impreso en la segunda línea debajo de los joggers a salvo, se actualiza cada vez que usted consigue pasar un personaje.

Una vez se haya familiarizado con el juego, lo puede convertir en una segunda versión. Con esta modificación el jogger no para de moverse y se necesita una gran concentración para ayudarle a pasar la autopista. Le recomendamos familiarizarse totalmente con el juego antes de programar la segunda versión.

El programa comienza en la línea 2Ø con la llamada a la subrutina de la línea 41Ø. Aquí la tinta se hace azul, el papel blanco, el borde rojo y se borra la pantalla. El bucle A de las líneas 44Ø a 5ØØ define los seis gráficos. El primero (gráfico A o carácter 144) es el jogger, los demás son los coches. Observará en este bucle (45Ø) que E$ se iguala al carácter 143 más el valor de A. El bucle interno B lee (READ) los datos (DATA) de las líneas 56Ø a 61Ø y los almacena (POKE) en su posición en la línea 48Ø.

La línea 51Ø determina la variable TIEMPO (el transcurrido por serie de cinco joggers, medido en latidos de corazón) e iguala la variable MEJORTIEMPO a un número muy grande (9E7 hexadecimal). Su MEJORTIEMPO será siempre menor a éste, incluso la primera vez que intente pasar los joggers, por tanto la primera vez que consiga pasarlos igualará la variable MEJORTIEMPO a su tiempo.

A$, B$, C$ y D$, que contienen los coches, son inicializados en las líneas 52Ø y 55Ø. Puede colocar tantos coches como quiera en estos vectores, ya que son de 32 caracteres (contenido de la pantalla). Una vez tome confianza al juego puede complicarlo incrementando el número de coches. No obstante, no se pueden colocar de una forma totalmente aleatoria. Los vehículos han sido diseñados para moverse en una u otra dirección. Los tres carriles superiores circulan hacia la derecha y los inferiores hacia la izquierda. Puede usar el gráfico B para cualquier carril, manteniendo el «C» para A$ y B$ y «D», «E» y «F» para los C$ y D$.

Una vez se han inicializado las variables y definido los gráficos la instrucción RETURN de la línea 555 envía el programa al bucle J que empieza en la línea 3Ø. Como puede observar, este bucle controla los principales elementos del programa. Se ejecuta cinco veces, una por cada jogger. La línea 4Ø imprime los joggers que están a salvo (siempre uno menos que el valor actual de J) y el tiempo transcurrido. La línea 5Ø imprime los joggers que quedan por pasar, que es la diferencia entre J y cinco (por tanto, cuando J es 1, quedan por pasar cuatro joggers ya que uno está pasando).

El pequeño bucle de las líneas 6Ø y 9Ø imprime la figura de un jogger por cada uno que ha conseguido atravesar. La línea 1ØØ borra

los joggers que quedan por pasar y el bucle de 11Ø a 13Ø imprime un jogger por cada uno que ha pasado. La variable JA es la posición del jogger que está atravesando y JD es la posición en la parte inferior de la pantalla. Los joggers empiezan su maratón desde la posición 16, 19, que es aproximadamente el centro de la pantalla, al lado de la primera línea de automóviles. La línea 15Ø borra la posición donde ha estado el jogger el último instante, por tanto, cuando el jogger está en movimiento, esta línea borra sus posiciones previas. Las líneas 16Ø y 17Ø leen el teclado, moviendo el jogger según sus deseos antes de que se imprima en 18Ø.

La larga línea 19Ø imprime los carriles de automóviles. Lo hace por medio de una cadena de PRINT AT, ya que es la manera más rápida de ejecución. Las líneas 2ØØ y 23Ø comparan la posición del jogger con el contenido del vector en este punto, si el jogger y un automóvil coinciden, la acción pasa a la línea 35Ø, donde empieza la rutina de jogger atropellado. Las líneas 25Ø a 28Ø controlan el movimiento de los coches usando las instrucciones de que dispone el BASIC de Sinclair para la rotación de vectores alfanuméricos.

La variable TIEMPO se incrementa en la línea 29Ø. La línea 3ØØ comprueba el valor de JD y, si detecta que es dos, el ordenador sabe que otro jugador ha conseguido cruzar. Se oye una bonita serie de tonos crecientes antes de que NEXT J, al final de la línea 3ØØ, envíe la acción a 4Ø, donde un nuevo corredor desafiará los coches.

Si el jogger no ha conseguido atravesar (es decir, si JD no es igual a dos), el ordenador salta el resto de la línea 3ØØ y va a la 31Ø. Si J es menor que seis, el ordenador devuelve la acción a la línea 15Ø, donde se borra el jogger, antes de ser reimpreso en 18Ø. Si todos los joggers han conseguido atravesar, la línea 32Ø imprime «Este juego:» y el tiempo que han tardado; pone un 5 al final de la línea «Joggers a salvo:»; e imprime «LO CONSIGUIO!». La línea 33Ø le comunica el tiempo que ha tardado y la 34Ø envía el programa a la 62Ø.

La línea 62Ø compara el tiempo de este juego con la variable MEJORTIEMPO, alterando MEJORTIEMPO por TIEMPO, si TIEMPO es menor. Si MEJORTIEMPO es distinto de 1ØØØØ (lo cual ocurrirá la primera vez que intente jugar, si no consigue terminar la partida), el MEJORTIEMPO se imprime en la pantalla, por tanto usted puede ver lo bien que lo ha hecho. Un par de trinos rematan el programa a partir de la línea 62Ø, la variable TIEMPO se iguala a cero y el programa va a 3Ø para empezar una nueva partida.

Si no ha conseguido ayudar a cruzar a todos los joggers, la rutina de la línea 35Ø entra en acción (líneas 2ØØ a 23Ø con instrucciones IF/THEN). El bucle de 35Ø a 37Ø imprime el jogger intermitente en el punto donde ha sido atropellado, se oye un cierto sonido y es informado de que «LE HAN ATROPELLADO» y del número de joggers que han conseguido pasar (39Ø). La variable TIEMPO se iguala a 1ØØØØ,

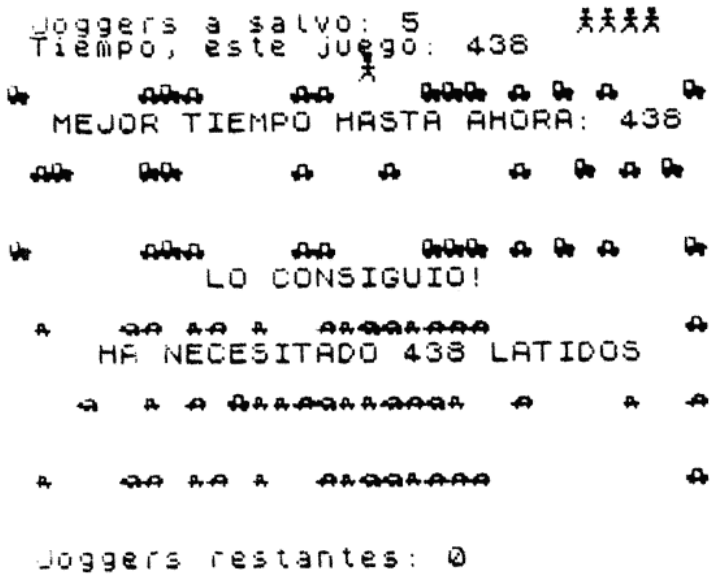
por lo tanto no será impresa ni influirá en su MEJORTIEMPO. Si el tiempo no se incrementara artificialmente en este punto, usted podría conseguir un brillante MEJORTIEMPO dejando que le atropellaran en el primer carril.

Una vez domine este juego, intente añadir más coches o invente sus propios vehículos. Unos pocos caballos, una o dos bicicletas o cualquier cosa que se mueva pueden mantenerle interesado por la pantalla. Cuando, a pesar de sus nuevas versiones, decrezca su interés por JOGGER, canvie el programa según el segundo listado de este juego. En esta versión los joggers se mueven siempre. La encontrará muy difícil y sugerimos reducir el número de vehículos o añadir la instrucción:

```
195  BEEP .Ø5,RND*5Ø
```

para facilitar el juego disminuyendo su velocidad. ¡Buena suerte jogger!, debe cruzar sin tener ningún accidente.

Cuando el programa está en marcha se ve así:



Y éste es el listado del programa:

```
  10 REM JOGGER
  20 GO SUB 410
  30 FOR J=1 TO 5
  40 PRINT AT 0,1;"Joggers a sal
vo: ";J-1;AT 1,1,"Tiempo transcu
rrido: ";TIEMPO;"
;AT 2,12;"                "
  50 PRINT AT 21,1;"Joggers rest
antes: ";5-J
  60 PRINT AT 0,26;
  70 FOR Z=1 TO J-1
  80 PRINT INK 2;"☃": REM GRAPH
IC A
  90 NEXT Z
 100 PRINT AT 21,26;"      ";AT 2
1,26;
 110 FOR Z=1 TO 5-J
 120 PRINT INK 2;"☃": REM GRAPH
IC A
 130 NEXT Z
 140 LET JA=16: LET JD=19
 150 PRINT AT JD,JA;" "
 160 LET JA=JA+(INKEY$="8")-(INK
EY$="5")
 170 LET JD=JD+(INKEY$="6")-(INK
EY$="7")
 180 PRINT AT JD,JA; INK 2;"☃"
 190 PRINT AT 3,0; INK 2;A$;AT 9
,0; INK 0;A$;AT 6,0; INK 4;B$;AT
 12,0; INK 3;C$;AT 18,0; INK 1;C
$;AT 15,0; INK 2;D$
```

```
 200 IF (JD=3 OR JD=9) AND A$(JA
+1)<>" " THEN GO TO 350
 210 IF JD=6 AND B$(JA+1)<>" " T
HEN GO TO 350
 220 IF (JD=12 OR JD=18) AND C$(
JA+1)<>" " THEN GO TO 350
 230 IF JD=15 AND D$(JA+1)<>" "
THEN GO TO 350
 250 LET A$=A$(31)+A$( TO 31)
 260 LET B$=B$(31)+B$( TO 31)
 270 LET C$=C$(2 TO )+C$(1)
 280 LET D$=D$(2 TO )+D$(1)
 290 LET TIEMPO=TIEMPO+1
 300 IF JD=2 THEN FOR Z=1 TO 20:
 BEEP .01,Z: NEXT Z: NEXT J
 310 IF J<6 THEN GO TO 150
 320 PRINT INK 2;AT 1,1;"Tiempo,
 este juego: ";TIEMPO;"      ";AT
 0,18;S;AT 10,8; FLASH 1; PAPER
6; BRIGHT 1;" LO CONSIGUIO!"
 330 PRINT ''TAB 4;"HA NECESITAD
O ";TIEMPO;" LATIDOS"
 340 GO TO 620
 350 FOR Z=1 TO 20
 360 PRINT AT JD,JA; FLASH 1; BR
IGHT 1; INK 2;" ": BEEP .008,Z
 370 NEXT Z
 380 PRINT AT 10,8;"LE HAN ATROP
ELLADO"
 390 PRINT AT 13,3;J-1;" JOGGERS
 LO HAN CONSEGUIDO"
 400 LET TIEMPO=10000: GO TO 620
 410 INK 1: PAPER 7: BORDER 2: C
LS
 440 FOR A=1 TO 6
 450 LET E$=CHR$ (143+A)
 460 FOR B=0 TO 7
 470 READ C
 480 POKE USR E$+B,C
 490 NEXT B
 500 NEXT A
 510 LET TIEMPO=0: LET MEJORTIEM
PO=9E7
 520 LET A$="
          "
 530 LET B$="
          "
 540 LET C$="
          "
 550 LET D$="
          "
 555 RETURN
 560 DATA 28,28,8,62,8,28,34,66
 570 DATA 0,60,36,231,255,102,0,
0
 580 DATA 240,144,158,255,254,10
2,0,0
 590 DATA 0,0,30,99,255,27,0,0
 600 DATA 0,0,56,40,126,36,0,0
 610 DATA 0,0,30,51,255,34,0,0
```

```
 520 IF TIEMPO<MEJORTIEMPO AND T
IEMPO<>10000 THEN LET MEJORTIEMP
O=TIEMPO
 530 IF MEJORTIEMPO<>9E7 THEN PR
INT AT 4,2; FLASH 1;"MEJOR TIEMP
O HASTA AHORA: ";MEJORTIEMPO
 540 FOR Z=50 TO 1 STEP -1: BEEP
 .05,Z: BEEP .05,-Z: NEXT Z
 550 CLS
 560 LET TIEMPO=0
 570 GO TO 30
```

```
  10 REM JOGGER
  15 REM El jogger siempre se mu
eve
  20 GO SUB 410
  30 FOR J=1 TO 5
  40 PRINT AT 0,1;"Joggers a sal
vo: ";J-1;AT 1,1;"Tiempo transcu
rrido: ";TIEMPO;"                "
;AT 2,0;"
                "
  50 PRINT AT 21,1;"Joggers rest
antes: ";5-J
  60 PRINT AT 0,26;
  70 FOR Z=1 TO J-1
  80 PRINT INK 2;"X";: REM GRAPH
IC A
  90 NEXT Z
 100 PRINT AT 21,26;"      ";AT 2
1,26;
 110 FOR Z=1 TO 5-J
 120 PRINT INK 2;"X";: REM GRAPH
IC A
 130 NEXT Z
 140 LET JA=16: LET JD=19
 150 PRINT AT JD,JA;" "
 155 LET Z$=INKEY$: IF Z$<"5" OR
 Z$>"8" THEN LET Z$=X$
 160 LET JA=JA+(Z$="8" AND JA<28
)-(Z$="5" AND JA>3)
 170 LET JD=JD+(Z$="6" AND JD>1)
-(Z$="7" AND JD<21)
 175 LET X$=Z$
 180 PRINT AT JD,JA; INK 2;"X"
 190 PRINT AT 3,0; INK 2;A$;AT 9
,0; INK 0;A$;AT 6,0; INK 4;B$;AT
 12,0; INK 3;C$;AT 18,0; INK 1;C
$;AT 15,0; INK 2;D$
 200 IF (JD=3 OR JD=9) AND A$(JA
+1)<>" " THEN GO TO 350
 210 IF JD=6 AND B$(JA+1)<>" " T
HEN GO TO 350
 220 IF (JD=12 OR JD=18) AND C$(
JA+1)<>" " THEN GO TO 350
```

```
 230 IF JD=15 AND D$(JA+1)<>" "
THEN GO TO 350
 250 LET A$=A$(31)+A$( TO 31)
 260 LET B$=B$(31)+B$( TO 31)
 270 LET C$=C$(2 TO )+C$(1)
 280 LET D$=D$(2 TO )+D$(1)
 290 LET TIEMPO=TIEMPO+1
 300 IF JD=2 THEN FOR Z=1 TO 20:
 BEEP .01,Z: NEXT Z: NEXT J
 310 IF J<6 THEN GO TO 150
 320 PRINT INK 2;AT 1,1;"Tiempo,
 este juego: ";TIEMPO;"      ";AT
 0,18;S;AT 10,8; FLASH 1; PAPER
6; BRIGHT 1;" LO CONSIGUIO!"
 330 PRINT '''TAB 4;"HA NECESITAD
O ";TIEMPO;" LATIDOS"
 340 GO TO 620
 350 FOR Z=1 TO 20
 360 PRINT AT JD,JA; FLASH 1; BR
IGHT 1; INK 2;"Ä": BEEP .008,Z
 370 NEXT Z
 380 PRINT AT 10,8;"LE HAN ATROP
ELLADO"
 390 PRINT AT 13,3;J-1;" JOGGERS
 LO HAN CONSEGUIDO"
 400 LET TIEMPO=10000: GO TO 620
 410 INK 1: PAPER 7: BORDER 2: C
LS
 420 LET X$="8"
 440 FOR A=1 TO 6
 450 LET E$=CHR$ (143+A)
 460 FOR B=0 TO 7
 470 READ C
 480 POKE USR E$+B,C
 490 NEXT B
 500 NEXT A
 510 LET TIEMPO=0: LET MEJORTIEM
PO=9E7
 520 LET A$="   ÄÄÄÄ     ÄÄ     ÄÄÄÄ
 Ä  Ä  Ä    Ä   Ä"
 530 LET B$="   Ä Ä     Ä   Ä
 Ä   Ä  Ä  Ä  ÄÄÄÄ"
 540 LET C$="Ä   ÄÄÄÄÄÄÄÄ
 Ä  Ä    ÄÄ  ÄÄ  "
 550 LET D$="ÄÄÄÄÄÄÄÄÄÄÄÄ   Ä     Ä
 Ä    Ä   Ä  Ä  Ä"
 555 RETURN
 560 DATA 28,28,8,62,8,28,34,65
 570 DATA 0,60,36,231,255,102,0,
0
 580 DATA 240,144,158,255,254,10
2,0,0
 590 DATA 0,0,30,99,255,27,0,0
 600 DATA 0,0,56,40,126,36,0,0
 610 DATA 0,0,30,51,255,34,0,0
 620 IF TIEMPO<MEJORTIEMPO AND T
IEMPO<>10000 THEN LET MEJORTIEMP
O=TIEMPO
 630 IF MEJORTIEMPO<>9E7 THEN PR
```

```
INT AT 4,2; FLASH 1;"MEJOR TIEMP
O HASTA AHORA: ";MEJORTIEMPO
 640 FOR Z=50 TO 1 STEP -1: BEEP
 .05,Z: BEEP .05,-Z: NEXT Z
 650 CLS
 660 LET TIEMPO=0: LET Z$="8"
 670 GO TO 30
```

Desactive esta mina

Este es un juego de muy difícil dominio ya que requiere controlar *dos* objetos de la pantalla al mismo tiempo. Su trabajo es limpiar la pantalla de minas y tiene una dificultad doble porque es una operación en dos etapas. Primero las minas deben ser desactivadas usando el *inversor.* Se consigue pasándolo por encima de una mina activada, pero atención: si lo pasa otra vez la mina se reactiva. El inversor se controla con las teclas «5» a «8».

Además del inversor, verá en la pantalla el *transporte.* Pasándolo por encima de una mina desactivada la recoge, desapareciendo de la pantalla. Debe controlar el transporte de una manera muy cuidadosa, ya que si choca con una mina activada la operación habrá terminado. El transporte se controla mediante las teclas «5» a «8» con la CAPS SHIFT pulsada a la vez.

Hay 4Ø minas y puede escoger cinco velocidades de juego, por lo tanto tardará en encontrar aburrido este programa. Como puede observar en la copia de muestra, le va a costar mantenerlo todo bajo control.

Cuando ejecute el programa, la línea 3Ø envía la acción a la subrutina de la línea 6ØØ, donde se definen los gráficos «A», «B» y «C». El «Z» de la línea 68Ø desencadena el RETURN de la subrutina de la línea 6ØØ. Después del RETURN, la línea 3Ø llama a otra subrutina en la línea 52Ø, donde se imprimen las instrucciones:

```
◆◆◆◆◆ DESACTIVE ESTA MINA ◆◆◆◆◆◆

SU TRABAJO ES LIBRAR LA PANTALLA
 DE MINAS.
ES UNA OPERACION DE DOS ETAPAS,
 PRIMERO LA MINA DEBE SER
 DESACTIVADA PASANDO EL INVERSOR
 (✱)  POR ENCIMA DE ELLA.
EL CONTROL DE ESTA OPERACION
 SE HACE CON LAS TECLAS 5 A 8.
LUEGO LA MINA DESACTIVADA DEBE
 SER CARGADA POR EL TRANSPORTE
 (■), CONTROLADO CON [CAPS SHIFT]
 + 5 A 8.
EL TRANSPORTE NO DEBE CHOCAR
 CON UNA MINA ACTIVA.
```

```
ENTRE NIVEL DE DIFICULTAD 1-5
(5=FACIL)

PULSE 6 PARA PARAR.
```

La rutina INKEY$ de las líneas 54Ø, 55Ø y 56Ø acepta solamente entradas válidas, rechazando las que estén fuera de los márgenes «1» a «6» y terminando el programa si se selecciona la «6» (56Ø). La línea 57Ø iguala la variable SP a un valor relacionado con la tecla pulsada. SP controla la dificultad del juego. La línea 58Ø devuelve la acción a la línea 4Ø.

La línea 4Ø ajusta los parámetros de pantalla. La tinta y el papel se convierten a 5 (azul claro), se borra la pantalla para establecer los colores y el borde toma el mismo color. Se cambia el color de la tinta a negro y la variable T$ se iguala a un espacio. Las líneas 5Ø y 6Ø inicializan variables que controlan las posiciones del inversor y del transporte y el número de minas desactivadas. La rutina de las líneas 7Ø, 8Ø y 9Ø dibuja el marco en el que tendrá lugar la acción y la rutina de las líneas 1ØØ a la 15Ø coloca las minas, asegurándose de que no se colocan unas sobre otras (13Ø). Si esto ocurriera, habría menos de 4Ø minas en la pantalla y el detector de «fin de juego» (actuando al final de la línea 37Ø si se han cargado todas las minas) no actuaría nunca.

Una vez las minas dispuestas, impresas en rojo (véase el cambio de color de la tinta en la línea 9Ø), la tinta vuelve a cambiar de color en la línea 155. La variable I$ se usa para interpretar sus deseos de movimiento. La primera sección de control (de 18Ø a 21Ø) interpreta la información de las teclas «5» a «8» sin haber pulsado CAPS SHIFT para mover el inversor. La sección siguiente (de 22Ø a 26Ø) se encarga de las mismas teclas habiendo pulsado CAPS SHIFT. [Obsérvese que I$ se toma en la línea 22Ø para ser traducido a su código (CODE) a fin de interpretar las teclas afectadas con CAPS SHIFT. Esta es una técnica muy útil para leer las teclas en este estado en cualquier punto de la pantalla.] Los resultados de estas lecturas se asignan a las variables A, D, CA y CD, que se añaden a las posiciones actuales del inversor y del transporte (A y D para el inversor, y CA y CD para el transporte), *inmediatamente* las posiciones antiguas se borran con espacios (27Ø).

La variable N almacena un valor igual a los ATTRibutos de la nueva posición del inversor y la CN los del transporte. La línea 32Ø reimprime los dos móviles y la 325 hace sonar un corto BEEP. Como puede observar, la longitud de este BEEP está relacionada con el factor de dificultad que se haya escogido en un principio. Cuanto mayor es este factor, más largo es el BEEP producido en 325. Como usted

sabe, mientras suena el BEEP, cesa toda acción en el Spectrum. Así pues, usando una variable como ésta, puede regularse la velocidad del juego. Si encuentra el programa demasiado difícil, incluso con nivel 5, puede cambiar el nueve de la línea 57Ø a un valor menor. Si N se iguala a 4Ø o CN se iguala a 4Ø (33Ø), el ordenador sabe que se ha estrellado contra la pared y el programa va a la línea 4ØØ, donde le informa que «SE HA ESTRELLADO CONTRA LA BARRERA ELECTRIFICADA». Las líneas 34Ø y 35Ø hacen los cambios necesarios en las minas (si su inversor se encuentra sobre ellas), desactivando minas o activándolas. Si el valor de CN es 42, el Spectrum sabe que ha tocado una mina activa con el transporte y va a la línea 44Ø donde se le informa: «SU TRANSPORTE HA GOLPEADO UNA MINA ACTIVA»; con letras rojas intermitentes.

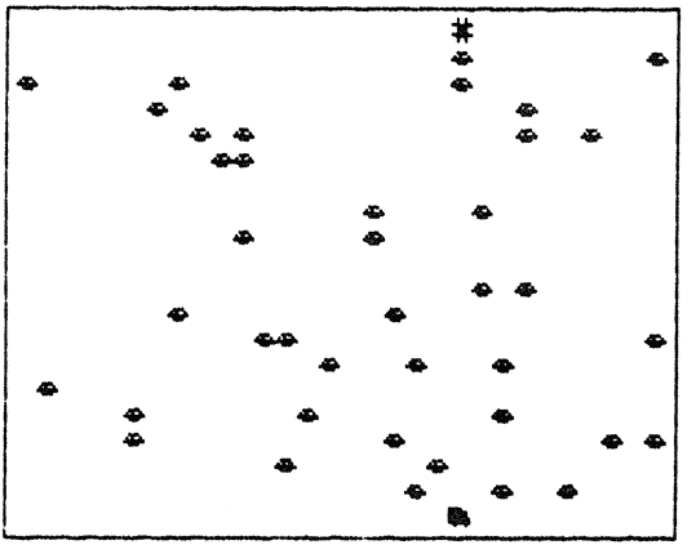
Si ha cargado una mina en el transporte (ATTRibuto CN es 41, línea 37Ø), la puntuación (S) de minas cargadas se incrementa en una unidad. Si S es igual a 4Ø (37Ø), el ordenador sabe que ya ha cargado todas las minas y la línea 47Ø le felicita: «LO LOGRO!!». Si no ocurre ninguna de las circunstancias descritas, la línea 38Ø hace volver el programa a la línea 16Ø para continuar el proceso.

El resto del programa, de 4ØØ a 51Ø, contiene los mensajes de final de programa y unas sinfonías de Spectrum. Una vez domine perfectamente el juego, puede eliminar la línea 325 y ver que tal lo hace. Es la prueba definitiva.

```
  30 GO SUB 600: GO TO 520
  40 INK 5: PAPER 5: CLS : BORDE
R 5: INK 0: LET T$="  "
  50 LET S=0: LET X=16: LET Y=1
  60 LET CX=16: LET CY=20: LET C
D=0
  65 LET A=1: LET D=0: LET CA=1
  70 PLOT 4,4: DRAW 247,0
  80 DRAW 0,167: DRAW -247,0
  90 DRAW 0,-167: INK 2
 100 FOR I=1 TO 40
 110 LET TX=INT (RND*30)+1
 120 LET TY=INT (RND*18)+2
 130 IF SCREEN$ (TY,TX)="" THEN
GO TO 110
 140 PRINT AT TY,TX;"◆"
 150 NEXT I
 155 INK 5
 160 LET I$=INKEY$
 170 IF I$="" THEN GO TO 270
 180 IF I$="5" THEN LET A=-1: LE
T D=0: GO TO 270
 190 IF I$="8" THEN LET A=1: LET
 D=0: GO TO 270
 200 IF I$="6" THEN LET A=0: LET
 D=1: GO TO 270
```

```
 210 IF I$="7" THEN LET A=0: LET
 D=-1: GO TO 270
 220 LET I=CODE I$
 230 IF I=8 THEN LET CA=-1: LET
CD=0: GO TO 270
 240 IF I=9 THEN LET CA=1: LET C
D=0: GO TO 270
 250 IF I=10 THEN LET CA=0: LET
CD=1: GO TO 270
 260 IF I=11 THEN LET CA=0: LET
CD=-1
 270 PRINT AT CY,CX;" ";AT Y,X;T
$: LET T$=" "
 280 LET X=X+A: LET Y=Y+D
 290 LET CX=CX+CA: LET CY=CY+CD
 300 LET N=ATTR (Y,X)
 310 LET CN=ATTR (CY,CX)
 320 PRINT INK 7;AT Y,X;"*";AT C
Y,CX;"■"
 325 BEEP .5,0
 330 IF N=40 OR CN=40 THEN GO TO
 400
 340 IF N=42 THEN LET T$=CHR$ 16
+CHR$ 1+"◆"
 350 IF N=41 THEN LET T$=CHR$ 16
+CHR$ 2+"◆"
 360 IF CN=42 THEN GO TO 440
 370 IF CN=41 THEN LET S=S+1: IF
 S=40 THEN GO TO 470
 380 GO TO 160
 400 CLS : PRINT FLASH 1; INK 0;
AT 0,0;"SE HA ESTRELLADO CONTRA
LA BARRERA ELECTRIFICADA";
 410 FOR I=1 TO 10: BEEP .02,-0
 420 BEEP .02,5: NEXT I
 430 GO TO 500
 440 CLS : PRINT FLASH 1; INK 2;
AT 0,2;"SU TRANSPORTE HA GOLPEAD
O UNA MINA ACTIVA"
 450 BEEP .2,-20: BEEP .5,-5
 460 GO TO 500
 470 PRINT AT 1,3; INK 0;"LO LOG
RO !!"
 480 BEEP 1,9
 500 INK 0
 510 PRINT '''"HA ELIMINADO ";S;"
 MINAS"
 515 PAUSE 200: CLS
 520 PRINT "◆◆◆◆◆ DESACTIVE ESTA
 MINA ◆◆◆◆◆◆";'''"SU TRABAJO ES L
IBRAR LA PANTALLA DE MINAS.";'"E
S UNA OPERACION DE DOS ETAPAS,
PRIMERO LA MINA DEBE SER
DESACTIVADA PASANDO EL INVERSOR
(*)  POR ENCIMA DE ELLA."
 525 PRINT "EL CONTROL DE ESTA O
PERACION     SE HACE CON LAS TEC
LAS 5 A 8.";'"LUEGO LA MINA DESA
CTIVADA DEBE    SER CARGADA POR E
L TRANSPORTE    (■), CONTROLADO C
```

```
ON[CAPS SHIFT] + S A S.";''"EL TR
ANSPORTE NO DEBE CHOCAR     CON
UNA MINA ACTIVA."
 530 PRINT ''"ENTRE NIVEL DE DIF
ICULTAD 1-5    (5=FACIL)";''"PULS
E 6 PARA PARAR."
 540 LET I$=INKEY$
 550 IF I$<"1" OR I$>"6" THEN GO
 TO 540
 560 IF I$="6" THEN STOP
 570 LET SP=VAL I$/9
 580 GO TO 40
 600 READ A$: IF A$="Z" THEN RET
URN
 610 FOR I=0 TO 7
 620 READ N: POKE USR A$+I,N
 630 NEXT I
 640 GO TO 600
 650 DATA "A",0,0,0,60,82,255,60
,0
 660 DATA "B",36,36,255,60,60,25
5,36,36
 670 DATA "C",0,0,248,254,254,25
5,255,34
 680 DATA "Z"
```



GRAFICOS: ◆=A ✱=B ■=C

Helicóptero

Este es un programa para probar sus aptitudes de vuelo. Dispone de cuatro helicópteros y su tarea es borrar un grupo de líneas amarillas.

Las líneas se borran aterrizando sobre ellas. Primero debe aterrizar en la más corta que vea, luego deberá seguir aterrizando en la más corta hasta borrarlas todas. Si hace una falsa maniobra perderá su helicóptero. Tiene cuatro helicópteros para completar la misión y cuantos menos utilice, mejor será su puntuación.

Para pilotar estas pequeñas máquinas debe usar el «Ø» para elevarse, el «5» y el «8» para moverse a derecha e izquierda. La gravedad tiende a hacer que descienda el aparato. Su puntuación final está relacionada con la cantidad de combustible que ha consumido y los helicópteros que ha perdido. Si se estrellan todos los helicópteros antes de borrar todas las líneas, aparecerá el mensaje «No hay más helicópteros».

La línea 1Ø pone el borde negro, la variable h se iguala a cero en 15 (para la máxima puntuación, vease líneas 532, 535), antes de igualar la tinta a amarillo y el papel a negro (3Ø). Después de esto, el programa va a la subrutina que empieza en la línea 61Ø, donde se toman los datos de la línea 9ØØ para construir el gráfico «a»: el pequeño helicóptero que se puede ver en la línea 45.

Después de definir el gráfico la pantalla se borra, estableciéndose el color negro. El bucle de las líneas 5Ø a 7Ø imprime un sólido bloque amarillo. Se dimensionan dos vectores para contener y ordenar las líneas sobre las que deben aterrizar los helicópteros. El bucle «a» se ejecuta de la línea 1ØØ a la 18Ø, usando el número aleatorio generado en la línea 11Ø y las instrucciones PLOT (15Ø) y DRAW (16Ø) para producir una base de aterrizaje. La rutina de las líneas 12Ø y 13Ø asegura que el vector «d» contiene diez líneas de distintas longitudes.

La variable r se iguala a 1 en la línea 185. Esta significa la línea en la que debe aterrizar el helicóptero (cuando r es 1, está apuntando a la línea representada por d [r]). Su variable de combustible, f, se iguala a 25ØØØ en la línea 19Ø. Además de controlar el combustible, f fija el tono del generador de sonido de la línea 24Ø (dividido por 1ØØØ). Como observará, esto produce un sonido que se identifica con la acción del helicóptero. El combustible se decrementa por 21 en las líneas 25Ø y, si se está elevando, en la 28Ø. La variable p, su coordenada horizontal, se iguala a 16 en la línea 21Ø antes de que su pequeño helicóptero empiece a funcionar en la línea 22Ø.

La línea 245 imprime los aparatos en reserva (usando I$ definido en la línea 45 como tres) en la esquina inferior derecha de la pantalla, y —después de decrementarse el combustible en la línea 25Ø e imprimir su volumen en la parte inferior izquierda de la pantalla— su heli-

cóptero se borra, antes de reaparecer. La variable u (inicialmente igualada a 1 en 2ØØ) controla su desplazamiento vertical (cuanto mayor es u, más bajo está) y las líneas 275 a 29Ø leen sus intenciones en el teclado («Ø»: elevarse, «5»: moverse a la izquierda, «8»: moverse a la derecha), antes de que el programa vuelva a 22Ø para reimprimir el helicóptero en su nueva posición.

En la línea 23Ø se detecta si hay un aterrizaje y si lo hay el programa se dirige a la línea 4ØØ dónde se comprueba donde ha sido. Si no ha aterrizado en la línea más corta— o sea, p distinto de d(r)— se oye el ruido producido por el accidente (BEEP .Ø1, —1Ø) y se le restan 1ØØØ unidades de combustible. El helicóptero se imprime en modo inverso (412) y se oye otro ruido (413) cuando el helicóptero desaparece (414). Se observa el contenido del vector I$, si está vacío, el ordenador sabe que se le han acabado los helicópteros. Recuerde que I$ fue igualado a tres helicópteros en la línea 45. La línea 416 elimina un helicóptero del final de este vector después de cada error de aterrizaje, por lo tanto cuando I$ queda vacío se sabe que se han usado los cuatro helicópteros (el cuarto no forma parte del vector cuando éste es inicializado). La línea 418 devuelve la acción a 2ØØ, donde arranca un nuevo helicóptero para cumplir su misión.

La línea 4ØØ comprueba si el aterrizaje ha tenido éxito. Si es así, se saltan todas las malas noticias de las líneas 4Ø5 a 418 y se va directamente a 42Ø, donde se borra el helicóptero y se borra la parte alta de las líneas amarillas (425). Un poco de sonido le permite recuperar el aliento y la variable r (que cuenta el número de líneas) se incrementa. Si r es menor de 11, todavía quedan líneas para aterrizar. Si no, se borran los helicópteros no usados (49Ø) y el combustible se incrementa en 1ØØØ unidades por cada helicóptero no usado, con ello se tiene en cuenta su precisión en la puntuación final. La nueva cifra de contenido de combustible se va imprimiendo a medida que se incrementa (52Ø).

La puntuación máxima se actualiza, si es preciso, en la línea 532 y se imprime en la próxima antes de que el programa vaya a 31Ø para anunciar que se ha terminado la partida. Si no ha conseguido borrar todas las líneas (es decir, si r es menor de 11), aparece el mensaje «No hay más helicópteros». Las líneas 33Ø y 34Ø esperan hasta que usted retira su dedo del teclado cuando termina la partida y vuelve a pulsar, con ello se sabe que usted está listo para emprender otra aventura. Si es así, el programa va a la línea 4Ø, borrando la pantalla (pero sin redefinir los gráficos ni la variable h de puntuación máxima) y dando paso a una nueva misión con sus cuatro nuevos helicópteros. ¡Buen vuelo!

```
  5 REM Helicoptero
 10 BORDER 0
 15 LET h=0
 20 INK 6
 30 PAPER 0
 35 GO SUB 610
 40 CLS
 45 LET l$="▝▝▝▝▝▝▝▝"
 50 FOR a=11 TO 21
 60 PRINT AT a,0; PAPER 6;
 70 NEXT a
 80 DIM d(10)
 90 DIM e(32)
100 FOR a=1 TO 10
110 LET b=INT (RND*32)
120 IF e(b+1)=1 THEN GO TO 110
130 LET e(b+1)=1
140 LET d(a)=b
150 PLOT b*8+4,87
160 DRAW INK 5;0,-a*8+1
180 NEXT a
185 LET r=1
190 LET f=25000
200 LET u=0
210 LET p=16
220 PRINT INK 4;AT u,p;"▝▘"
230 IF u=9+r THEN GO TO 400
240 BEEP .01,f/1000
245 PRINT INVERSE 1;AT 21,27;l$
;" "
250 LET f=f-21
255 PRINT AT 21,0; INVERSE 1;"G
as: ";f;" "
260 PRINT AT u,p;" "
270 LET u=u+1
275 LET i=IN 61438
280 IF INT (i/2)=i/2 THEN LET u
=u-2*(u>1): LET f=f-21
290 LET p=p+(i=251 OR i=250)*(p
<31)-(IN 63486<255)*(p>0)
300 GO TO 220
310 PRINT AT 7,8; INK 5;"PARTID
A ACABADA"
320 IF r<11 THEN PRINT AT 21,0;
PAPER 1; FLASH 1;"No hay mas he
licopteros!"
330 IF INKEY$<>"" THEN GO TO 33
0
340 IF INKEY$="" THEN GO TO 340
350 GO TO 40
400 IF p=d(r) THEN GO TO 420
405 BEEP .01,-10
410 LET f=f-1000
412 PRINT INVERSE 1;AT u,p;"▝▘"
413 BEEP .1,-20
414 PRINT AT u,p;" "
415 IF l$="" THEN GO TO 310
416 LET l$=l$(2 TO )
418 GO TO 200
```

```
420 PRINT AT u,p;" "
425 PRINT AT 10+r,0;"
                     "
430 FOR a=0 TO 20
440 BEEP .01,a
450 NEXT a
460 LET r=r+1
470 IF r<11 THEN GO TO 200
480 FOR a=1 TO LEN l$
490 PRINT AT 21,26+a;"■"
500 BEEP .5,a*10
510 LET f=f+a*1000
520 PRINT AT 21,0;"Gas: ";f
530 NEXT a
532 IF h<f THEN LET h=f
535 PRINT INVERSE 1;AT 21,14;"G
as maximo: ";h
540 GO TO 310
610 FOR b=0 TO 7
620 READ c
630 POKE USR "a"+b,c
640 NEXT b
650 RETURN
900 DATA 0,85,8,28,62,62,20,34
```

Mazurca

En MAZURCA se le desafía a la cacería de su vida. El programa está inspirado en los juegos Duck Shoot, donde se debe disparar a patos y otros objetos que pasan delante de usted.

La versión computerizada del juego tiene nuevos riesgos. En MAZURCA se mueven frente a usted varias filas de objetos, parcialmente tapados por cuatro barreras que se mueven en dirección contraria. Su deber es tocar tantos objetos como pueda sin afectar las barreras. Debe mover su pistola (gráfico «H») usando las teclas «5» y «8», obteniendo desplazamientos en la dirección indicada en las citadas teclas. Su gatillo es la tecla «F». En la pantalla debe haber sólo una bala a la vez. Dispone de un número limitado de balas y las que quedan están indicadas por la longitud de la línea de la parte inferior de la pantalla.

Como puede observarse en la muestra impresa, el juego dispone de una gran cantidad de objetos móviles que crean una excitante imagen. No obstante, esto no es todo. De tanto en tanto un pato atraviesa volando la pantalla, algo parecido al intruso galáctico que aparece en ciertos juegos de acción. Puede obtener un premio de seis puntos si lo abate, pero si no lo consigue se le restarán quince balas. Es bastante fácil tocarlo, ya que el pato vuela por debajo de las barreras, como puede ver en la muestra.

La subrutina de la línea 6ØØ se llama desde la 2Ø. Esta rutina define los gráficos: «A», «B», «C», «D», «E», «H», «I» y «J». Cuando la línea 6ØØ (READ C$) lee «Z» en la línea 75Ø, se activa el RETURN. Una vez de vuelta a la línea 3Ø, el papel y el borde se hacen amarillos y se borra la pantalla para establecer este color. La línea 35 hace la tinta negra y el programa salta a la línea 5ØØ, donde se imprimen las instrucciones:

```
LLLLLLLLLLL MAZURCA LLLLLLLLLLLL
DEBE DISPARAR A LOS OBJECTOS QUE
 SE MUEVEN, EVITANDO LAS LINEAS
 NEGRAS. LAS LINEAS MAS ALTAS
 DAN UNA PUNTUACION MAS ELEVADA.
MUEVA LA PISTOLA CON LAS TECLAS
 5 Y 8. DISPARE CON LA TECLA F
SOLO PUEDE VERSE UNA BALA A LA
 VEZ Y EL NUMERO DE BALAS
 RESTANTES ESTA EXPRESADO POR
 LA LONGITUD DE LA LINEA
 INFERIOR. CADA VEZ QUE UN PATO
 LLEGA AL EXTREMO DE LA PANTALLA
 SE LE DEDUCEN 15 DISPAROS.

PULSE [Y] PARA EMPEZAR Y
  [N] PARA PARAR
```

Después de esto se le pide que entre «Y» para empezar el juego. Esta instrucción se repite al final de cada juego, por tanto usted puede abandonar el juego en este punto respondiendo «N» o «n». Cualquier otra tecla reiniciará el programa. Al repetir el juego no se vuelven a inicializar los gráficos de la línea 6ØØ.

Al volver a la línea 4Ø se inicializan las variables y se dimensiona el vector que contendrá los objetivos móviles. Los bucles desde la línea 6Ø a la 1ØØ asignan los objetivos a este vector de una forma aleatoria. Se imprimen con la línea 95 y la línea 97 crea un tono creciente mientras se inicia el juego. La variable B del final de la línea 1ØØ controla la longitud de la línea situada debajo de usted. Línea que se corresponde con el número de disparos de los que dispone. B se decrementa en 21Ø y la línea de pantalla se despunta con el PLOT OVER. A$ (5) contiene las barreras móviles y de A$ (1) y A$ (4) los objetos sobre los que debe disparar. La línea 1Ø7 suena para informarle de que todo empieza a funcionar.

El bucle I de las líneas 11Ø a 35Ø es la parte principal del programa. Controla el movimiento de los objetos (igualando T$ al primer elemento del vector en la línea 12Ø y recolocándolo al final del vector en 13Ø), imprimiéndolo (14Ø) y escogiendo diferentes tintas para cada línea (la primera azul, la segunda roja, la tercera magenta y la cuarta

verde). Después de esto, la línea 17Ø borra la posición donde estaba la pistola y la 18Ø cambia el valor de la posición horizontal de acuerdo con la tecla pulsada (19Ø ajusta la posición si se intenta pasar los límites de la pantalla).

Se imprime la pistola en la línea 2ØØ. Si B es cero, se han usado todas las balas y la acción va de 2Ø5 a 43Ø, donde se le comunica que se ha quedado sin munición. La línea 21Ø es la que comprueba si puede disparar (si F es igual a uno, no hay otra bala moviéndose, por lo tanto puede disparar). La línea 21Ø inicializa la coordenada horizontal de la bala (FX) igual a la posición de la pistola (G). F es la coordenada vertical de la bala, por lo tanto (F, FX) es la posición donde debe imprimirse y borrar la bala. La línea 24Ø decrementa la coordenada F (lo que tiene el efecto de mover la bala una línea hacia arriba).

Los puntos por donde pasa la bala son observados en la línea 25Ø. Si SCREEN$ (F, FX) está vacío, la bala va a través del aire y no toca ningún objeto. Si se encuentra algún contenido en SCREEN$ (F, FX), se examina para discernir si se trata de la barrera (las líneas 26Ø y 265, si P es cinco, es la barrera). Si ha tocado la barrera, la partida se termina y el ordenador va a la línea 4ØØ para comunicarle la noticia. Si no ha tocado la barrera, ha tocado algo y este algo no puede ser más que uno de los objetos a los que apunta. La línea 27Ø le da una puntuación relacionada con la altura del objeto abatido (cuanto menor es el valor de P, más alto está el objeto en la pantalla). Si ha tocado algo, la bala debe parar y para ello se iguala F a uno (véase el final de la línea 27Ø). Si ha tocado el pato, P será igual a seis y usted recibe un premio de seis puntos y la variable D se iguala a 0. El programa salta a 3ØØ si ha tocado un pato. Si no, se imprime la nueva puntuación intermitentemente mediante la línea 275 y el elemento del vector tocado se cambia a un espacio, por lo tanto desaparece el objeto.

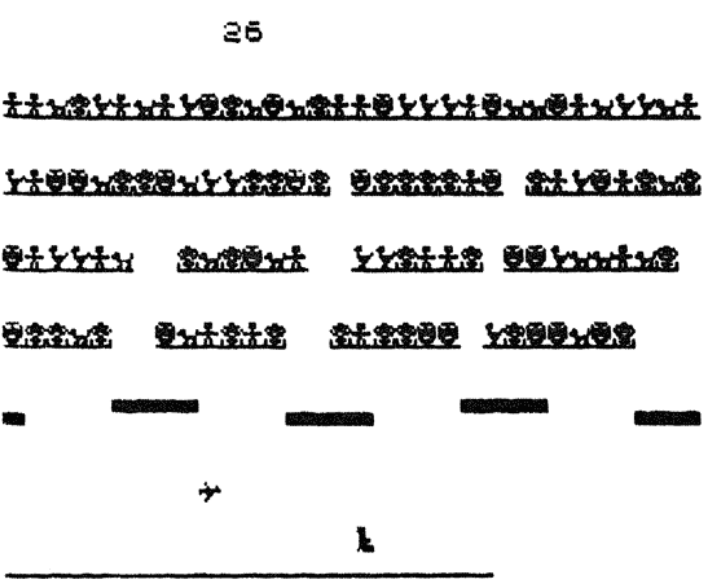
Si no ha tocado nada (por lo tanto no se llega al GO TO 3ØØ del final de la línea 28Ø), la línea 29Ø imprime la bala. Aproximadamente cada 5Ø pasos de programa la línea 3ØØ genera un nuevo pato, si no hay otro en la pantalla. La línea 32Ø mueve el pato y la 33Ø comprueba si ha llegado a la parte derecha de la pantalla. Si es así, el pato se borra, D se iguala a cero (la condición de «no pato») y se restan quince balas de su munición, borrándose una parte de la línea situada debajo de usted. La última parte de la línea 33Ø (GO TO 35Ø) se salta la impresión del pato en 34Ø. La línea 35Ø finaliza este bucle principal.

Recuérdese que A$ (5) contiene las barreras móviles. En la línea 36Ø T$ se iguala al elemento final de A$ (5), que se mueve al principio del vector en 37Ø. Esto es lo contrario de lo que se hace con el resto de los vectores A$ y, por lo tanto, las barreras se mueven en dirección opuesta y a la misma velocidad que los objetivos.

La línea 38Ø imprime la barrera y la 39Ø devuelve la acción a la línea 11Ø, donde vuelve a empezar el bucle principal. Si ha tocado

la barrera se activan, las líneas 4ØØ y 42Ø (P es igual a cinco, desde una parte anterior del programa). Después de informarle de la situación, el programa va a 45Ø para imprimir su puntuación. Si se le acaba la munición (es decir, si B llega a cero) se activan las líneas 43Ø y 44Ø, apareciendo en la pantalla el mensaje «NO LE QUEDAN MAS BALAS». Después de los mensajes de final de partida, vuelve a aparecer la página del título y si entra cualquier tecla que no sea «N» o «n», el juego se reanuda.

Aquí tiene una muestra del programa en acción:



Y éste es el listado:

```
   20 GO SUB 600
   30 PAPER 6: BORDER 6: CLS
   35 INK 0: GO TO 500
   40 LET G=16: LET F=1: LET S=0
   50 LET FX=0: DIM A$(5,32): CLS

   60 FOR I=1 TO 4
   70 FOR J=1 TO 32
   80 LET A$(I,J)=CHR$ (INT (RND*
5)+144)
   90 NEXT J
   95 PRINT AT I*3,0;A$(I)
   97 BEEP .4,I*2
  100 NEXT I: LET B=256
  102 LET D=0: DRAW 255,0
  105 LET a$(5)="
  107 BEEP 1,9
```

```
 110 FOR i=1 TO 4
 120 LET T$=A$(I,1)
 130 LET A$(I)=A$(I,2 TO )+T$
 140 PRINT INK I;AT I*3,0;A$(I)
 170 PRINT AT 20,G;" "
 180 LET G=G+(INKEY$="8")-(INKEY
$="5")
 190 LET G=G+(G=-1)-(G=32)
 200 PRINT AT 20,G;"▙"
 205 IF B=0 THEN GO TO 430
 210 IF F=1 AND INKEY$="F" THEN
PRINT AT 1,FX;" ": LET F=20: LET
 FX=G: BEEP .02,-15: LET B=B-1:
PLOT OVER 1;B,0
 220 PRINT AT F,FX;" "
 230 IF F=1 THEN GO TO 300
 240 LET F=F-1
 250 IF SCREEN$ (F,FX)=" " THEN
GO TO 290
 260 LET P=F/3: BEEP .02,5
 265 IF P=5 THEN GO TO 400
 270 LET S=S+5-P: LET F=1
 272 IF P=6 THEN LET S=S+6: PRIN
T AT 18,FX;" ": LET D=0: GO TO 3
00
 275 PRINT FLASH 1;AT 0,10;S
 280 LET A$(P,FX+1)=" ": GO TO 3
00
 290 PRINT AT F,FX;"▲"
 300 IF D=0 AND RND>.98 THEN LET
 D=1
 310 IF D=0 THEN GO TO 350
 320 LET D=D+1
 330 IF D=31 THEN PRINT AT 18,31
;" ": LET D=0: LET B=B-15: LET B
=B*(B>=0): DRAW OVER 1;-PEEK 236
77+5,0: BEEP .02,9: GO TO 350
 340 PRINT AT 18,D;" ✈"
 350 NEXT I
 360 LET T$=A$(5,32)
 370 LET A$(5)=T$+A$(5, TO 31)
 380 PRINT AT 15,0;A$(5)
 390 GO TO 110
 400 BEEP 2,-9
 410 PRINT FLASH 1;AT 0,0;"HA AL
CANZADO LA BARRERA "
 420 GO TO 450
 430 BEEP 1,1: BEEP 1,9
 440 PRINT FLASH 1;AT 0,0;"NO LE
 QUEDAN MAS BALAS"
 450 PRINT  FLASH 1; INVERSE 1;"
SU PUNTUACION ES ";S
 455 PAUSE 200
 460 PAUSE 200: CLS
 500 PRINT "▙▙▙▙▙▙▙▙▙▙ MAZURCA
▙▙▙▙▙▙▙▙▙▙"''"DEBE DISPARAR A
LOS OBJECTOS QUE SE MUEVEN, EVIT
ANDO LAS LINEAS  NEGRAS. LAS LIN
EAS MAS ALTAS    DAN UNA PUNTUAC
```

```
ION MAS ELEVADA."'"MUEVA LA PIST
OLA CON LAS TECLAS  5 Y 8. DISPA
RE CON LA TECLA F"'"SOLO PUEDE V
ERSE UNA BALA A LA    VEZ Y EL NU
MERO DE BALAS         RESTANTES E
STA EXPRESADO POR     LA LONGITUD
 DE LA LINEA          INFERIOR. C
ADA VEZ QUE UN PATO   LLEGA AL EX
TREMO DE LA PANTALLA SE LE DEDUC
EN 15 DISPAROS."
 510 PRINT '"PULSE [Y] PARA EMP
EZAR Y            [N] PARA PARAR"
 520 IF INKEY$="" THEN GO TO 520
 530 IF INKEY$="N" OR INKEY$="N"
 THEN STOP
 540 GO TO 40
 599 STOP
 600 READ C$: IF C$="Z" THEN RET
URN
 610 FOR I=0 TO 7
 620 READ N: POKE USR C$+I,N
 630 NEXT I
 640 GO TO 600
 650 DATA "A",BIN 00110000,BIN 0
1110000,BIN 00100001,BIN 0011111
0,BIN 00011100,BIN 00001000,BIN
00011000,255
 660 DATA "B",0,0,BIN 01100001,B
IN 11100001,BIN 00111110,BIN 000
10010,BIN 00110110,255
 670 DATA "C",BIN 00011000,BIN 0
1111110,BIN 10111101,BIN 0110011
0,BIN 00011000,BIN 10111101,BIN
10011001,255
 680 DATA "D",BIN 01111100,BIN 1
0010010,254,254,BIN 11000110,BIN
 01111100,BIN 00111000,255
 685 DATA "E",BIN 00111000,BIN 0
0111000,16,254,16,40,40,255
 690 DATA "H",BIN 00111000,BIN 0
0111000,BIN 00011000,BIN 0001100
0,BIN 00011110,BIN 00011000,BIN
00011111,BIN 00101111
 700 DATA "I",0,0,8,28,28,28,62,
0
 710 DATA "J",0,0,32,18,255,BIN
10011000,16,32
 750 DATA "Z"
```

GRAFICOS: =H =I =J

Alfabatalla

En este inusual juego usted tiene el control de un helicóptero verde mientras que el Spectrum se encarga de un caza azul. El objeto de la batalla es la posesión del alfabeto. Sus letras están a la izquierda de la pantalla y las del ordenador a la derecha.

Su helicóptero se mueve horizontalmente mediante las teclas «5» y «8» y verticalmente con «w» y «z». Debe posicionarse en frente de una letra no intermitente del enemigo. Entonces, usted pulsa «o» y la letra correspondiente de su lado empezará a parpadear. La letra del enemigo desaparecerá, su puntuación se incrementará y su aparato será transportado a la derecha de la pantalla para iniciar otro ataque. Puede pulsar más de una tecla a la vez (ya que se usa IN en lugar de INKEY$ para leer el teclado, líneas 41Ø y 42Ø), por lo tanto puede obtenerse un movimiento en diagonal. Mientras usted hace todo esto, el ordenador hace acciones parecidas para robar sus letras.

Volar en frente de las letras y pulsar «b» es un método para obtener letras. Hay otro. Si el caza del ordenador está situado en frente de una de sus letras no intermitentes, usted puede situarse en la misma línea y disparar con la tecla «o». El ordenador tratará de hacer lo mismo, por lo tanto la acción se mantiene muy interesante.

El juego termina cuando han sido capturadas todas las letras de un contendiente. Una vez terminado, el ordenador muestra la puntuación (tenga presente que el ordenador le llama «humanoide»).

Las líneas 1ØØ a 15Ø definen los gráficos (se usan las letras «a», «b», «c» y «d») leyendo de los datos (DATA) de la línea 9ØØ. Se dimensionan dos matrices, p y t; se distribuyen en cada una 22 letras del alfabeto posicionadas aleatoriamente y se imprimen en la pantalla.

La posición de su helicóptero se define por los valores de u y p. La coordenada vertical (p) se inicializa con el valor 29 (véase línea 29Ø) y la horizontal una línea más arriba de donde se ha impreso la última letra. La posición del caza del computador se establece de forma similar respecto a su última letra impresa, usando como coordenadas a y t (que empieza en 1 en la línea 31Ø). Las líneas 41Ø y 42Ø leen el teclado, usando IN como ya se ha mencionado, y su aparato se imprime en 43Ø. Si ha decidido capturar una letra o disparar al caza del computador, la línea 44Ø pasa la acción a la 58Ø, la cual a su vez dirige el programa a la subrutina de la línea 1Ø5Ø que dispara su arma-laser. Si a es igual a u en la línea 44Ø, el programa va a 7ØØ para capturar la letra e incrementar su puntuación. Si su aparato se encuentra totalmente a la izquierda de la pantalla, p será igual a 1 y el ordenador irá igualmente a 7ØØ a hacer el mismo trabajo citado.

La línea 46Ø borra la posición actual del caza del Spectrum y la siguiente rutina se utiliza para determinar la nueva posición. La variable que controla la posición vertical del caza (a) se cambia en la línea

48Ø y t se cambia en la siguiente para mantener su posición cercana a la de su helicóptero. Las líneas 61Ø y 62Ø añaden cierto componente aleatorio al movimiento del caza.

La línea 65Ø reimprime el caza en su nueva posición y al principio de 66Ø se determina si el caza le disparará o no, enviando el programa a 1ØØØ si decide hacerlo. Si t es 29, el ordenador sabe que está completamente a su lado de la pantalla, por lo tanto envía la acción a 11ØØ, donde se captura la letra amenazada. Se borra la letra de la misma altura, se ejecuta la subrutina de la línea 8ØØ para añadir sonido al juego y se coloca la letra intermitente correspondiente. Si la puntuación conseguida por usted (s) y la conseguida por el Spectrum (x) suman 22, la partida acaba y la línea 735 redirige el programa a 12ØØ, donde se borra la pantalla antes de mostrar el resultado de ALFABATALLA. La línea 745 borra el caza e iguala a Ø las variables p y u, antes de volver a 4ØØ para empezar de nuevo.

```
  10 REM Alfabatalla
 100 FOR a=1 TO 4
 110 FOR b=0 TO 7
 120 READ c
 130 POKE USR CHR$ (143+a)+b,c
 140 NEXT b
 150 NEXT a
 200 DIM p(2,22)
 205 DIM t(2,22)
 210 FOR p=1 TO 22
 215 FOR f=1 TO 2
 220 LET e=INT (RND*22)
 230 IF p(f,e+1)>0 THEN GO TO 22
0
 240 LET p(f,e+1)=p
 250 PRINT INK 6;AT e,(f=2)*31;C
HR$ (64+p)
 255 LET t(f,p)=e+1
 260 NEXT f
 270 NEXT p
 280 LET u=t(2,22)-1
 290 LET p=29
 300 LET a=t(1,22)-1
 310 LET t=1
 320 DIM h(2,22)
 330 LET s=0
 340 LET x=0
 400 PRINT AT u,p;"  "
 410 LET p=p+(IN 61438<>255)*(p<
29)-(IN 63486<>255)
 420 LET u=u+(IN 65278<>255)*(u<
21)-(IN 64510<>255)*(u>0)
 430 PRINT AT u,p; INK 4;"▟▙"
 440 IF IN 57342<>255 AND p>t TH
EN GO SUB 580: IF a=u THEN GO TO
 700
 450 IF p=1 THEN GO TO 700
 460 PRINT AT a,t;"  "
```

```
 470 IF p(1,u+1)<11 THEN GO TO 5
50
 480 LET a=a+(a<u)-(a>u)
 490 LET t=t+(ABS (a-u)<ABS (t-p
))-(ABS (a-u)>ABS (t-p))*(t>1)
 500 GO TO 650
 550 IF h(1,p(2,a+1))=1 OR h(2,p
(2,a+1))=1 THEN GO TO 600
 560 LET t=t+1
 570 GO TO 650
 580 GO SUB 1050
 590 BEEP .005,20: GO SUB 1050
 595 RETURN
 600 LET r=RND
 610 IF RND<.5 OR u=21 THEN LET
a=a-(a>0)
 620 IF RND>.5 OR a=0 THEN LET a
=a+(a<20)
 650 PRINT AT a,t; INK 5;"►"
 660 IF ABS (a-u)<3 AND p>t AND
h(1,p(1,u+1))=0 AND h(2,p(1,u+1)
)=0 THEN GO SUB 1000: BEEP .005,
0: GO SUB 1000: IF a=u THEN GO T
O 850
 670 IF t=29 THEN GO TO 1100
 680 GO TO 400
 700 IF h(1,p(1,u+1))=1 OR h(2,p
(1,u+1))=1 THEN GO TO 745
 705 PRINT AT u,0;" "
 710 GO SUB 800
 720 PRINT FLASH 1;AT t(2,p(1,u+
1))-1,31;CHR$ (64+p(1,u+1))
 730 LET s=s+((h(2,p(1,u+1)))=0)
 735 IF s+x=22 THEN GO TO 1200
 740 LET h(2,p(1,u+1))=1
 745 PRINT AT u,p;" "
 750 LET p=29
 760 IF u=a THEN LET u=(a=0)
 765 PRINT #1;AT 1,0;"Puntuacion
";(x),"Puntuacion ";(s)
 770 GO TO 400
 800 BEEP .005,10
 810 BEEP .01,20
 820 BEEP .005,10
 830 RETURN
 850 PRINT AT a,t;"  "
 860 LET a=u
 870 GO TO 1100
 900 DATA 0,64,48,62,31,31,31,32
 905 DATA 0,0,0,0,192,252,0,0
 910 DATA 0,3,7,15,31,8,127,0
 920 DATA 0,230,230,246,254,16,2
54,0
1000 PLOT OVER 1;(t+2)*8,(21-a)*
8+3
1010 DRAW OVER 1; INK 3;(p-t)*8,
0
1030 RETURN
1050 PLOT OVER 1;p*8,(21-u)*8
```

```
1060 DRAW OVER 1; INK 5;(t-p)*8,
0
1070 RETURN
1100 IF h(2,p(2,a+1))=1 OR h(1,p
(2,a+1))=1 THEN GO TO 1160
1105 PRINT AT a,31;" "
1110 GO SUB 800
1120 PRINT FLASH 1;AT t(1,p(2,a+
1))-1,0;CHR$ (64+p(2,a+1))
1130 LET x=x+((h(1,p(2,a+1)))=0)
1140 IF x+s=22 THEN GO TO 1200
1150 LET h(1,p(2,a+1))=1
1160 PRINT AT a,t;" "
1170 LET t=1
1180 IF u=a THEN LET a=(u=0)
1190 GO TO 765
1200 CLS
1205 PRINT AT 11,8;"PARTIDA ACAB
ADA"
1210 PRINT AT 19,0;
1220 PRINT "spectrum  ";
1230 FOR a=1 TO x
1240 PRINT "/";
1250 NEXT a
1260 PRINT TAB 29;x;"humanoide "
;
1270 FOR a=1 TO s
1280 PRINT "\";
1290 NEXT a
1300 PRINT TAB 29;s
1310 IF INKEY$<>"" THEN GO TO 13
10
1320 IF INKEY$="" THEN GO TO 132
0
1330 RUN
```

Pared

Esta es una versión llena de colorido del clásico «Breakout», donde se trata de atravesar una pared de ladrillo usando una pelota de diamante de cantos muy vivos. Como verá al ejecutar este juego, el gráfico «A» ha sido redefinido para parecerse a unos ladrillos. De esta forma, la pared que debe derribar parece muy real. Dispone de 12 pelotas y el movimiento de su raqueta se controla con las teclas «5» y «8» para ir a la izquierda y a la derecha respectivamente. La puntuación se muestra continuamente y se actualiza cuando es necesario.

La línea 15 pone el borde y el papel rojos y la tinta blanca. El marco, dentro del cual se juega, se imprime con la línea 2Ø (la parte superior) y el bucle de la 3Ø. Su color es verde. (No es en absoluto necesario mantener estos colores. Distintos aparatos de televisión reaccionan de forma diferente al Spectrum y combinaciones que dan una buena calidad de imagen en unos no la dan en otros. Le recomen-

damos, pues, que experimente con su aparato.) La variable S (puntuación) se iguala a cero en 35 y, A y B (que controlan el vuelo de la bola) se inicializan a uno en esta misma línea. Las líneas 4Ø y 42 son bucles que definen los gráficos de los ladrillos y de la bola, usando las instrucciones de datos de 21Ø y 215.

El siguiente bucle, de la línea 45 a la 55, imprime los ladrillos. La línea 55 consiste en 28 gráficos «A». La línea 5Ø escoge un número aleatorio para determinar el color de cada fila de ladrillos. La segunda parte de dicha línea comprueba que el color escogido no sea el rojo, ya que, si fuera así, se confundiría con el papel. Si se confirma que ha sido seleccionado el rojo, se salta otra vez al inicio de la línea para escoger otro color.

La línea siguiente (6Ø) determina la posición horizontal inicial de la bola. La bola se imprime en X, Y y las variables usadas para borrarla (DX y DY) se igualan a X, Y en esta línea. Si Y es mayor que 28 o menor que 3, la bola habrá golpeado un lateral y la línea 65 envía la acción a la 155 donde se invierte la dirección del movimiento (para simular el rebote de la bola en la pared).

Si X es menor que 2, la bola ha llegado a la parte superior del marco y la línea 7Ø la hace rebotar hacia abajo. La línea 75 examina la posición que ocupará inmediatamente la bola. Si encuentra que es un ladrillo, va a la subrutina de la línea 125, que hace el ruido del ladrillo al romperse y cambia el valor de B, que se usa para incrementar el valor de Y. Si X es mayor que 2Ø, la bola está en el fondo del marco. La línea 8Ø examina este extremo y comprueba si la raqueta está en una posición correcta para devolver la pelota. Si es así, se produce el sonido del rebote (BEEP .ØØ8,1Ø) y se multiplica por menos uno la variable A que controla el cambio de la posición vertical de la pelota.

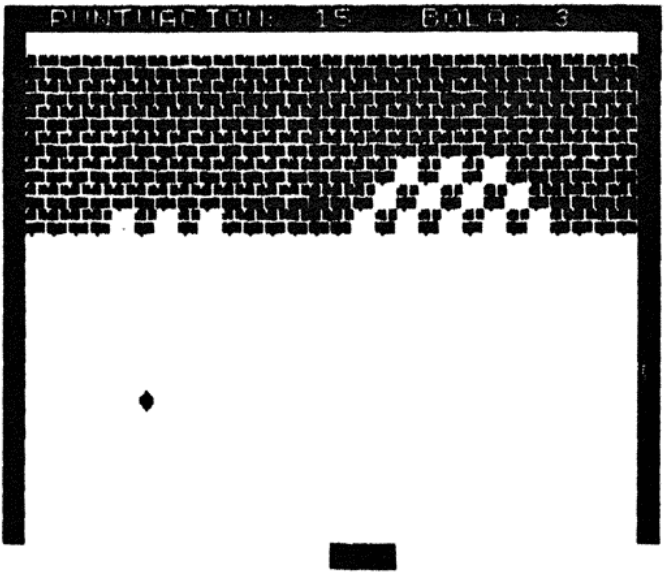
La primera parte de la línea 85 usa las variables DX y DY para borrar la anterior posición de la bola y ésta se imprime usando X y Y. La línea 9Ø imprime la raqueta. Véase el espacio en blanco dejado a ambos lados de la misma, que permite borrar automáticamente la posición anterior. La rutina de la línea 95 lee el teclado usando el sistema INKEY$ y cambiando el valor de N, que determina la posición de la raqueta.

La posición de la bola se actualiza en la línea 1Ø5. Si X es mayor que 21 (línea 11Ø), el ordenador sabe que la bola ha pasado la línea de la raqueta y ha caído al fondo. La subrutina de la línea 165 imprime el número de la bola, la variable del número de bola (Q) se incrementa en 1 y se genera un sonido de nueva bola en la línea 175. En 18Ø se comprueba el número de bolas usadas. Si es menor que 12, el programa cae en la próxima línea. Si es igual a 12 aparece el mensaje «PARTIDA ACABADA» intermitentemente y con una gran variedad de colores. A la vez suena una música de acompañamiento a la intermitencia. Después de este brillante final, empieza un nuevo juego.

Si Q no ha pasado de 12 se genera una nueva posición de inicio para la pelota. La última línea de la pantalla se borra totalmente para eliminar la última posición de la raqueta y la dirección de la bola (hacia arriba a la derecha o a la izquierda) se determina por el valor dado a M en la línea 185 e incorporado a la variable B en la línea 2ØØ. La línea 2Ø5 devuelve la acción a la 6Ø, donde se repite toda la acción descrita. Las dos líneas finales del programa —21Ø y 215— contienen los datos para los ladrillos y la bola respectivamente.

```
  10 REM Pared
  15 BORDER 2: PAPER 2: INK 7: C
LS
  20 PRINT INK 4;"                
                "
  25 LET Q=1
  30 FOR N=1 TO 20: PRINT AT N,1
; INK 4;"
        ": NEXT N
  35 LET S=0: LET A=1: LET B=1
  40 FOR N=USR "A" TO USR "A"+6:
 READ K: POKE N,K: NEXT N
  42 FOR N=USR "B" TO USR "B"+6:
 READ K: POKE N,K: NEXT N
  45 FOR N=2 TO 8
  50 LET Z=INT (RND*8): IF Z=2 T
HEN GO TO 50
  55 PRINT AT N,2; INK Z; BRIGHT
 1;"                              
": NEXT N
  60 LET N=15: LET X=20: LET Y=I
NT (RND*10)+5: LET DX=X: LET DY=
Y
  65 IF Y>28 OR Y<3 THEN GO SUB
150
  70 IF X<2 THEN LET A=-A
  75 IF SCREEN$ (X,Y)<>" " THEN
GO SUB 125
  80 IF X>20 AND (Y=N+2 OR Y=N+1
 OR Y=N+3) THEN LET A=-A: BEEP .
008,10: LET Y=Y+1
  85 PRINT AT DX,DY;" ";AT X,Y;"
♦": LET DX=X: LET DY=Y
  90 PRINT AT 21,N;"      "
  95 LET N=N+(INKEY$="8" AND N<=
26)-(INKEY$="5" AND N>0)
 105 LET X=X-A: LET Y=Y+B
 110 IF X>21 THEN GO TO 165
 115 PRINT AT 0,3; INVERSE 1;"PU
NTUACION: ";S: INVERSE 0
 120 GO TO 65
 125 BEEP .008,20
 130 PRINT AT DX,DY;" ";AT X,Y;"
♦"
 135 LET A=-1
 140 LET S=S+1
 145 RETURN
```

```
 150 BEEP .008,30
 155 LET B=1-2*(Y>28 OR Y<2)
 160 RETURN
 165 PRINT AT 0,20; INVERSE 1;"B
OLA: ";Q: INVERSE 0
 170 LET Q=Q+1
 175 BEEP 1,0: BEEP 0.20,20
 180 IF Q>12 THEN FOR Z=1 TO 10:
 PRINT AT 5,8; INK RND*6; FLASH
1;"PARTIDA TERMINADA": FOR D=0 T
O 30 STEP 3: BEEP .008,D: BEEP .
008,-(D): NEXT D: NEXT Z: RUN
 185 LET M=RND
 190 PRINT AT 21,0;"
                   "
 195 LET A=1
 200 LET B=-B*(M<.5)+B*(M>=.5)
 205 GO TO 60
 210 DATA BIN 1110111,BIN 111101
11,BIN 11110111,0,BIN 11111110,B
IN 11111110,BIN 11111110,0
 215 DATA BIN 1000,BIN 11100,BIN
 111110,BIN 111110,BIN 11100,BIN
 1000,0
```



Conducción 3-D

Este corto programa es muy efectivo y, una vez lo haya ejecutado algunas veces, lo encontrará muy interesante. Usted conduce por una carretera formada por dos líneas que convergen en el horizonte. Usando las teclas «5» y «8», intentará mantener su vehículo dentro de los límites de la carretera. También puede cambiar de velocidad usando la tecla «1».

El programa empieza definiendo ciertas variables (o, p, i$, q, d y r) y haciendo el papel y el borde negros y la tinta blanca. La línea 35 lee el teclado («5» y «8») multiplicándolo por la velocidad (i). En la línea 4Ø p se suma a d (inicializando previamente a cero en 25) y se usa para determinar las coordenadas de las líneas en la subrutina de la línea 1ØØ que representa la carretera. La variable o es su posición horizontal en la pantalla (véase la línea 7Ø) que se calcula en función de p en la línea 45. Su velocidad varía aleatoriamente en la línea 5Ø si p está entre 12Ø y 14Ø. La pantalla se borra en 55, antes de reimprimir la carretera en la subrutina de la línea 1ØØ. A pesar de que la pantalla se borra continuamente para ser impresa de nuevo, la instrucción DRAW se ejecuta tan rápidamente que el parpadeo es mínimo.

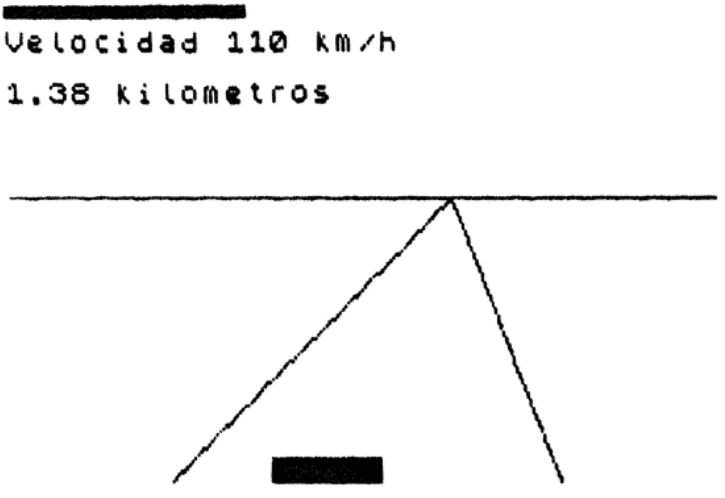
La línea 65 comprueba si ha tocado los extremos de la carretera y, si lo ha hecho, envía el programa a la subrutina de la línea 15Ø. Su vehículo se reimprime en la línea 7Ø y la distancia recorrida (s) se incrementa en un valor relativo a la velocidad, imprimiéndose a continuación en la pantalla.

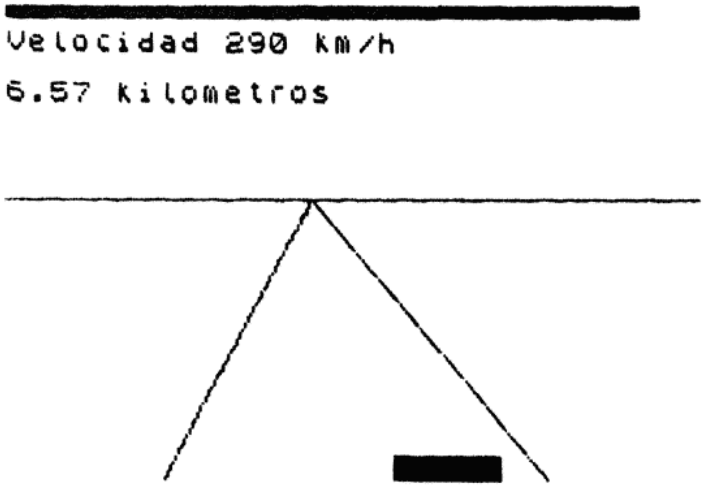
La línea 77 permite variar la velocidad (si d es igual a cero) y 8Ø imprime una línea en la parte superior de la pantalla que actúa como velocímetro. La línea 9Ø le reenvía a la 35 para seguir conduciendo. La subrutina de las líneas 1ØØ a 14Ø utiliza las instrucciones PLOT y DRAW para dibujar la carretera en la pantalla.

Si desea mejorar el programa, podría cambiar la línea 7Ø por un gráfico definido por usted, en el que se representara un parabrisas con un volante.

```
   1 REM Conduccion 3-D
   3 INK 7: BORDER 0: PAPER 0: C
LS
   5 LET o=16: LET s=0
  10 LET p=128
  12 LET i$="
                "
  15 LET i=4
  20 LET q=60
  25 LET d=0
  30 LET r=200
  35 LET p=p-(i*2)*((INKEY$="8")
-(INKEY$="5"))
  40 LET p=p+d
```

```
  45 LET o=(255-p)/8
  50 IF p>120 AND p<140 AND i<30
 THEN LET i=i+1: LET d=1*(INT (R
ND*2)-INT (RND*2))
  55 CLS
  60 GO SUB 100
  65 IF ATTR (21,0)=6 OR ATTR (2
1,o+4)=6 THEN GO TO 150
  70 PRINT AT 21,o;"          "
  75 LET s=s+i/100: PRINT AT 3,0
;s;" kilometros  "
  77 IF INKEY$="1" AND d=0 THEN
LET i=i-2*(i>2): BEEP .01,20
  80 PRINT AT 0,0; INK 2; BRIGHT
 1;i$( TO i); INK 6;AT 1,0;"Velo
cidad ";i*10;" km/h"
  90 GO TO 35
 100 PLOT 0,87: DRAW 255,0
 105 PLOT q,0
 110 DRAW INK 6;p-q,87
 120 PLOT r,0
 130 DRAW INK 6;p-r,87
 140 RETURN
 150 PRINT AT 11,1; FLASH 1; BRI
GHT 1; INK 2; PAPER 6;"Ha conduc
ido ";s;" kilometros"
 160 BORDER RND*7
 165 BEEP .007,30+RND*20
 180 IF RND<.96 THEN GO TO 160
 185 PAUSE 200
 190 RUN
```





Pánico en Noruega

Una vez haya conseguido óptimos resultados en CONDUCCION 3-D, puede medir sus fuerzas con otro programa de conducción, PANICO EN NORUEGA. En este programa, usted conduce en Noruega a través de una serie de paredes y túneles, algunos de ellos a través de montañas. Verá los muros avanzando hacia usted después de una curva obligándole a retrasar su camino. En este programa cuesta un poco aprender a «ver» los gráficos, pero una vez lo consiga lo encontrará un buen examen de conducción y de habilidad en el manejo de las teclas «5» y «8».

El vector a$, dimensionado en la línea 1Ø, contiene las nueve posiciones distintas que adoptan las paredes cuando se le aproximan. Los gráficos se apartan y se hacen mayores a medida que se acercan, debido al cambio de perspectiva. La variable t (línea 7Ø) determina su posición inicial antes de imprimir las paredes en el bucle b (líneas 9Ø a 13Ø). Su puntuación, s, se incrementa en la línea 1Ø7 y la 12Ø lee el teclado («5» y «8»). La lectura se modifica en función del elemento de a$ impreso y del valor de t (que no puede exceder de 21 para que quepa en la pantalla).

Si en la línea 14Ø se descubre que t tiene un valor situado entre 13 y 9, se habrá estrellado y el programa irá a la rutina de la línea 2ØØ donde se imprime su puntuación y se le ofrece un nuevo juego. Si después de una ejecución del bucle, no se ha estrellado, la variable i (que se inicializa a uno en la línea 5) se iguala al valor negativo de su propio contenido (es decir, de 1 cambia a −1 y de −1 cambia a 1) y se usa para determinar la nueva combinación de colores de PAPEL-

BORDE-TINTA de su próxima carretera. Las rutas con paredes amarillas y entorno negro representan túneles, las otras combinaciones corresponden a carreteras al aire libre.

Una vez familiarizado con el juego, puede añadir un registro de la puntuación máxima y quizás algún método para variar la velocidad de su vehículo. El mejor punto para iniciar esta modificación es el bucle b, incorporándole una instrución STEP.

```
   1 REM Norway panic
   5 LET i=1
   7 BORDER 4: INK 0: PAPER 7: C
LS
  10 DIM a$(9,10)
  20 FOR a=1 TO 9
  30 READ a$(a)
  40 NEXT a
  50 DATA "      ▪ ▪   ","     ▐ ▐
   "," ▟ ▙ ","   ","   ▟ ▙ ","
","█  █▛ ","  ","█ ▜▙ ","
█    █ ","█ ","█ ","█
     █"
  55 LET s=1
  60 LET a=9
  70 LET t=INT (RND*22)
  80 PRINT AT 21,14; INK 2;"####
#"
  90 FOR b=1 TO 12
 100 PRINT AT a+b,t;a$(INT ((b/1
.3)+.5))
 105 BEEP 1/s,b
 107 LET s=s+1: PRINT AT 0,0;s
 110 PRINT AT a+b,0;""
 120 LET t=t-((INKEY$="8")*(t>0)
-(INKEY$="5")*(t<21))*(INT (b/4)
)
 130 NEXT b
 140 IF t>14 OR t<9 THEN GO TO 2
00
 160 LET i=-i
 170 IF i=-1 THEN PAPER 0: BORDE
R 0: INK 6
 180 IF i=1 THEN PAPER INT (RND*
7)+1: BORDER 4: INK 0
 190 CLS
 195 GO TO 60
 200 FOR a=1 TO 20
 205 LET n=INT (RND*8): IF n=2 T
HEN GO TO 205
 210 PAPER n: CLS
 220 NEXT a
 225 PAPER 6: INK 0: CLS
 230 PRINT AT 11,8;"Puntuacion "
;s
 235 PRINT AT 18,7;"Pulse alguna
tecla"
 240 PAUSE 0
 250 RUN
```

Carrera mortal

¿Podrá eludir al temible reptante verde que aparece en este programa?

Se crea un laberinto rectangular en la pantalla. Usted se encuentra a la izquierda del mismo y el reptante sale de un punto indeterminado. Usando las teclas «5» y «8» para moverse a izquierda y derecha respectivamente y «w» y «z» arriba y abajo, debe ayudar al pequeño personaje a llegar a la parte derecha del laberinto.

La línea 5 activa el generador de números aleatorios y las líneas 1Ø a la 3Ø hacen el papel y el borde negros y la tinta amarilla. La línea 4Ø envía la acción a la rutina de inicialización de la línea 5ØØ, donde se toman los datos para los dos gráficos (el hombre y el reptante). Una vez definidos los gráficos, el programa vuelve a la línea 5Ø, donde aparece una función para obtener números aleatorios. Se borra la pantalla en la línea 55, activándose el color de papel definido en la línea 3Ø. Se inicializan dos variables: x (el número usado para hacer aberturas en el laberinto en las líneas 16Ø y 17Ø) y s (su puntuación inicial, que se decrementa en la línea 245 cada vez que se ejecuta el bucle principal).

La línea 1ØØ imprime una línea gruesa amarilla en la parte superior de la pantalla y el bucle de 11Ø a 13Ø imprime líneas verticales de color azul y una última de color amarillo (125). La línea 14Ø completa el laberinto con otra línea gruesa amarilla en la parte inferior de la pantalla.

En este punto se ejecuta la función a, imprimiendo espacios en blanco y puntos aleatorios del laberinto. Estos son los caminos que usará (y que también usará el reptante) en su carrera hacia la derecha de la pantalla. Las líneas 19Ø a 22Ø asignan valores a u (coordenada vertical inicial del hombre), t (la horizontal del reptante), a (la vertical del reptante) y p (la horizontal del hombre. Se le da el valor 1, ya que éste debe partir desde el lado izquierdo de la pantalla).

La línea 235 detecta si el reptante y el hombre ocupan la misma posición. Si así fuera, el reptante habría alcanzado al hombre y el programa va a la línea 4ØØ, donde después de un fragmento de «El Himno del Reptante» se le informa de que ha perdido la partida. El resto del bucle hasta 315 se encarga de leer el teclado.

En la línea 36Ø se imprime la noticia «HA GANADO LA CARRERA MORTAL». 365 comprueba si ha batido la puntuación máxima, igualando esta variable (h) a s (la puntuación lograda). Las líneas 37Ø, 375 imprimen la puntuación y la máxima puntuación respectivamente. Observe que la línea 37Ø utiliza la versión alfanumérica de la puntuación (LEN STR$) para posicionar la impresión. Las líneas 38Ø y 39Ø esperan a que quite las manos del teclado. Si lo hace y pulsa una nueva tecla (39Ø), se inicia una nueva carrera (desde la línea 55).

Cuando adquiera una gran habilidad en el juego, puede modificarlo añadiendo reptantes a la carrera.

```
  5 RANDOMIZE
  7 LET h=0
 10 BORDER 0
 20 INK 6
 30 PAPER 0
 40 GO SUB 500
 50 DEF FN a(x)=INT (RND*x)+1
 55 CLS
 60 LET x=20
 70 LET s=10000
 80 REM w para moverse adelante
 90 REM z para moverse atras
100 PRINT INVERSE 1;"
  "
110 FOR a=0 TO 20
120 PRINT INK 5;" █ █ █ █ █ █ █
█ █ █ █ █ █ █ █"
125 PRINT AT a,30; INVERSE 1;"
█ ";
130 NEXT a
140 PRINT AT 21,0; INVERSE 1;"
  "
150 FOR a=2 TO 28 STEP 2
160 PRINT AT FN a(x),a;" "
170 PRINT AT FN a(x),a;" "
180 NEXT a
190 LET u=FN a(x)
200 LET a=FN a(x)
210 LET p=1
220 LET t=FN a(15)*2-1
230 PRINT AT INT a,t; INK 4;"X"
235 IF ATTR (u,p)=4 THEN GO TO
400
240 PRINT AT u,p;"*"
245 LET s=s-16
250 BEEP .03,s/200
255 PRINT AT 0,1; INVERSE 1;s;"
"
260 PRINT AT u,p;" "
265 IF INKEY$="8" AND ATTR (u,p
+1)=6 THEN LET p=p+2: BEEP .1,p:
 BEEP .1,p-12
270 IF INKEY$="5" AND ATTR (u,p
-1)=6 THEN LET p=p-2: LET s=s-10
0: BEEP .1,0
275 IF p=29 THEN GO TO 320
280 LET u=u+(INKEY$="z")*(u<20)
-(INKEY$="w")*(u>1)
285 PRINT AT INT a,t;" "
290 LET a=a+RND*(INT a<u)-RND*(
INT a>u)
295 IF ATTR (INT a,t+1)=6 AND t
<p THEN LET t=t+2
300 IF ATTR (INT a,t-1)=6 AND t
>p THEN LET t=t-2
```

```
 315 GO TO 230
 320 FOR a=30 TO 60
 330 BEEP .01,a
 340 BEEP .01,a-12
 350 NEXT a
 360 PRINT AT u,1; OVER 1;"Gano
la Carrera de la Muerte"
 365 IF h<s THEN LET h=s
 370 PRINT AT u+1-2*(u=20),30-LE
N STR$ s;s
 375 PRINT AT 0,1; FLASH (h=s);
INVERSE 1;"PUNTUACION MAXIMA ";h
 380 IF INKEY$<>"" THEN GO TO 38
0
 385 IF INKEY$="" THEN GO TO 385
 390 GO TO 55
 400 FOR f=60 TO 30 STEP -1
 410 BEEP .01,f
 420 BEEP .01,f-12
 430 NEXT f
 445 PRINT AT u+1-2*(u=20),1; OV
ER 1;"Perdio la Carrera de la Mu
erte"
 450 PRINT AT INT a,t; FLASH 1;
INK 4;"*"
 460 GO TO 375
 500 DATA 24,24,48,94,16,104,76,
0
 510 DATA 0,34,84,12,12,84,34,0
 520 FOR a=0 TO 1
 530 FOR b=0 TO 7
 540 READ c
 550 POKE USR CHR$ (144+a)+b,c
 560 NEXT b
 570 NEXT a
 580 RETURN
```

Clono loco

Un solo zombi se ha reproducido 19 veces y 2Ø zombis están listos para cazarle. Usted está atrapado dentro de un área rectangular donde las principales características topográficas (a parte de los zombis y su desdichada persona) son un cierto número de peligrosos agujeros.

El zombi original puede ser fácilmente distinguido por su sutil método de persecución y por su odio a los humanos. Un odio que roza el de un maníaco. Como es bien conocido, el resto de zombis son idénticos a su antecesor y, por lo tanto, van todos detrás de su sangre. Estos, además de heredar las habilidades de su antecesor, han heredado también sus debilidades. La principal de ellas es la poca agudeza visual de estas criaturas. Esta será la clave de su supervivencia.

Recordará que se ha citado la existencia en la tierra de los zombis de una serie de peligrosos agujeros. La orientación de los zombis para perseguirle es muy buena, ya que perciben las emanaciones producidas por el *homo sapiens.* En cambio, esta capacidad es totalmente inútil para detectar los agujeros. Esto se puede conseguir procurando que entre usted y un zombi exista un agujero. Tenga en cuenta, sin embargo, que mientras trata de embaucar a un zombi, otro puede alcanzarle.

Puede moverse hacia arriba, abajo, derecha o izquierda, o, si se siente especialmente valiente, puede pararse de cuando en cuando. Los zombis pueden moverse horizontalmente, verticalmente y en diagonal. Cada partida da comienzo con un equipo completo de zombis y cuatro vidas para conseguir su objetivo. Ganará un punto por cada zombi que caiga por un agujero y, si consigue librar toda el área de zombis, se le premia con seis puntos más.

Cada vez que sobreviva, se redibuja la pantalla con una nueva colonia de zombis y una cantidad menor de agujeros. Cada vez que sea alcanzado por un zombi o que caiga por un agujero perderá una vida. Después de cuatro incidentes habrá muerto totalmente.

El programa dispone de un registro de la puntuación máxima para que pueda pasar los próximos cinco años intentando mejorarla. Puede moverse por la pantalla usando las teclas: «a» (arriba), «s» (abajo), «k» (izquierda) y «l» (derecha). Cualquier otra tecla parará su movimiento.

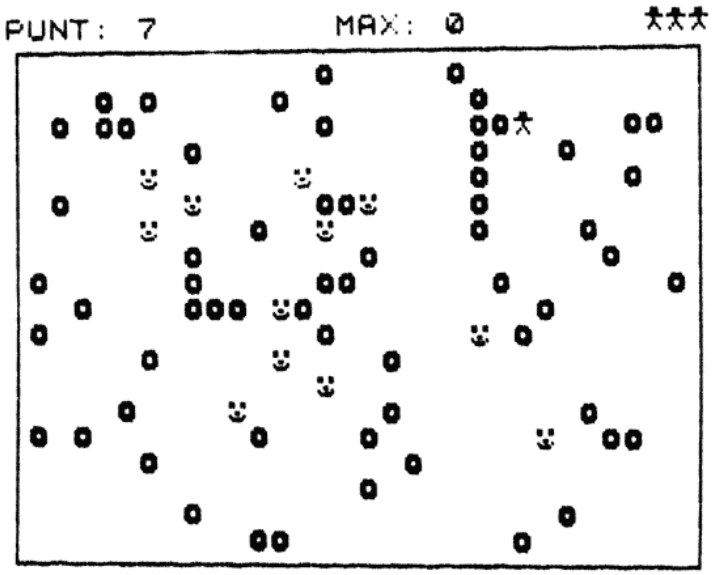
Después de activar el generador de números aleatorios mediante la línea 1, se DIMensiona una matriz para contener los miembros de la colonia de zombis. La variable rand se iguala a 0.1 (15) y la acción pasa a la subrutina de la línea 9ØØØ. Los gráficos de los agujeros, zombis y usted se determinan en esta subrutina. En la línea 91ØØ el borde se hace azul, el papel amarillo y la tinta negra. Se muestra una página de título, seguida de un recordatorio de las teclas que deben pulsarse. El mensaje: «Pulse una tecla para empezar» (92ØØ) precede a la instrucción PAUSE O, que para el programa hasta que usted pulsa una tecla.

La instrucción RETURN devuelve la acción a la línea 2Ø, donde se determina el número de vidas (la variable se llama men y se iguala a 3, ya que cuando sea igual a 0, usted todavía tiene una vida). La variable de la puntuación (score) se iguala a cero y en la línea 3Ø se inicializan otras variables necesarias en el programa.

En la dirección 2356Ø se almacena un cero (POKE). Esta dirección contiene el valor de la tecla pulsada y almacenando un cero en ella se consigue borrar la última tecla pulsada. La línea 1ØØ envía el programa a la línea 1ØØØ donde empieza una subrutina que dibuja la tierra de los zombis. Se dibuja el borde en la línea 1Ø1Ø y el bucle doble (i y j) de la línea 1Ø2Ø emplaza los agujeros. El número de aguje-

ros depende del valor de la variable rand (1Ø4Ø) y del número generado por la instrucción RND de esta misma línea.

Después de ejecutar la subrutina citada, la línea 11Ø dirige la acción a la de la línea 3ØØØ que coloca en la pantalla los zombis, contándolos (principio de la línea 3Ø5Ø) y detectando su posición para almacenarla en la matriz z (últimas dos partes de la línea 3Ø5Ø). La línea 12Ø llama a la subrutina de la línea 2ØØØ, que le coloca a usted en la pantalla. La serie de instrucciones «IF SCREEN$...» detectan si el área que rodea la posición inicial está libre. Si no lo estuviera, podría ser alcanzado por un zombi o caer por un agujero sin tener oportunidad de escapar o de corregir su trayectoria.



Si observa esta instantánea del juego, verá que el número de vidas que le quedan se indican en la parte superior derecha de la pantalla, por encima del área. Las líneas 13Ø a 15Ø imprime esta indicación, la línea 16Ø imprime la puntuación en la parte superior izquierda de la pantalla y la puntuación máxima en el centro de la pantalla (17Ø).

Recordará que en la línea 4Ø se había almacenado un cero en la dirección 2356Ø con el fin de borrarla. Ahora se espera que pulse una tecla («a», «s», «k» y «l»). Cuando lo haga, será almacenada en esta dirección por la línea 2ØØ. Antes de que el ordenador reaccione a sus instrucciones, las variables tx y ty se igualan a su posición actual. Estas variables se usarán para borrarlo, si es necesario, en la línea 45Ø. Si no ha pulsado ninguna de las cuatro teclas, la línea 25Ø salta a

la rutina que detecta su posición y lo reimprime. La línea 26Ø hace una rápida comprobación de que, en su alocada carrera, no se haya salido de los extremos del área. La 27Ø comprueba si está vacía la posición donde pretende ir. Si no lo está, la rutina de la línea 4ØØØ genera una indicación luminosa y acústica que pone de manifiesto su error. Si pasa el examen de la línea 27Ø, vuelve a ser impreso en 28Ø.

La rutina de 3ØØ a 49Ø se ejecuta cierto número de veces hasta que el valor de la variable num (incrementada en 4ØØ) se hace mayor que el valor de go (que se inicializó a 4 en la línea 3Ø y que se altere en la 485). Esta rutina contiene la malevolente inteligencia de los zombis. En la línea 41Ø se realiza un examen (SCREEN$) de la pantalla y, si encuentra un espacio en blanco, va a una rutina asociada en las líneas 5ØØ a 53Ø. Si el número de zombis que quedan (información contenida en la variable left) llega a cero (es decir, que ha destruido a todos los zombis, incluido el original), obtiene un premio de puntuación (8ØØØ a 8Ø5Ø) y se decrementa el valor de la variable rand (lo que implica que tendrá menos agujeros en la próxima partida, cosa que se determina en la línea 1Ø4Ø).

Cuando la rutina de los zombis ha terminado, la línea 3Ø5 devuelve la acción a la línea 2ØØ, donde el proceso se reanuda. El bucle de la línea 7ØØØ imprime los zombis con un fondo musical (7Ø2Ø y 7Ø4Ø). Una vez el número de vidas (men) ha llegado a cero (7Ø6Ø), el ordenador usa la rutina de la línea 7Ø7Ø para, si es necesario alterar, la variable que contiene la máxima puntuación y para ofrecer una nueva partida. Si entra «n» para indicar que no desea iniciar un nuevo juego, la línea 7Ø9Ø pone la tinta, el papel y el borde de forma que facilite el listado del programa o la entrada de otro. Si desea volver a jugar (y se detecta «y» en la línea 7Ø8Ø) la acción va a la línea 25, que asegura que no se borra su máxima puntuación (hi).

Puede modificar fácilmente las teclas a usar para moverse. Es suficiente alterar las líneas 21Ø a 24Ø, donde se usa el valor numérico de la última tecla entrada (dirección 2356Ø). Si lo hace, también deberá cambiar las instrucciones de la línea 915Ø a 918Ø. Para generar más o menos agujeros, cambie el valor inicial de la variable rand en las líneas 15 y 7Ø63. Para modificar la función de cambio de agujeros al final de cada partida, altere la línea 8ØØ6. Los gráficos definidos por el usuario son: «A» (Vd.), «B» (zombi) y «C» (agujero).

```
  1 RANDOMIZE
 10 DIM z(30,2)
 11 LET hi=0
 15 LET rand=0.1
 20 GO SUB 9000
 25 LET men=3: LET score=0
 30 LET left=0: LET max=20: LET
zombie=0: LET go=4
 40 POKE 23560,0
```

```
 100 GO SUB 1000
 110 GO SUB 3000
 120 GO SUB 2000
 130 IF men>2 THEN PRINT AT 0,29
;"★"
 140 IF men>1 THEN PRINT AT 0,30
;"★"
 150 IF men>0 THEN PRINT AT 0,31
;"★"
 160 PRINT AT 0,0;"PUNT: ";score
 170 PRINT AT 0,15;"MAX: ";hi
 200 LET a=PEEK 23560
 205 LET tx=px: LET ty=py
 210 IF a=97 THEN PRINT AT py,px
;" ": LET py=py-1: GO TO 260
 220 IF a=115 THEN PRINT AT py,p
x;" ": LET py=py+1: GO TO 260
 230 IF a=107 THEN PRINT AT py,p
x;" ": LET px=px-1: GO TO 260
 240 IF a=108 THEN PRINT AT py,p
x;" ": LET px=px+1: GO TO 260
 250 GO TO 300
 260 IF px<1 OR px>30 OR py<2 OR
 py>20 THEN GO TO 4000
 270 IF SCREEN$ (py,px)<>" " THE
N GO TO 5000
 280 PRINT AT py,px;"★"
 300 LET num=0
 305 IF num>go THEN GO TO 200
 310 LET zombie=zombie+1: IF zom
bie>max THEN LET zombie=1
 320 IF z(zombie,1)=0 THEN GO TO
 310
 330 LET zx=z(zombie,1): LET zy=
z(zombie,2): LET tx=zx: LET ty=z
y
 340 LET dx=0: LET dy=0
 350 IF zx<px THEN LET dx=1
 360 IF zx>px THEN LET dx=-1
 370 IF zy<py THEN LET dy=1
 380 IF zy>py THEN LET dy=-1
 390 LET zx=zx+dx: LET zy=zy+dy
 400 LET num=num+1
 410 IF SCREEN$ (zy,zx)=" " THEN
 GO TO 500
 420 IF SCREEN$ (zy,zx)="☺" THEN
 GO TO 305
 430 IF zx=px AND zy=py THEN GO
TO 7000
 440 LET z(zombie,1)=0
 450 PRINT AT ty,tx;" "
 460 LET score=score+1: PRINT AT
 0,0;"PUNT: ";score
 470 LET left=left-1: IF left=0
THEN GO TO 8000
 480 FOR n=1 TO 10: BEEP 0.01,n:
 NEXT n
 485 LET go=1+INT (left/5)
 490 GO TO 305
```

```
 500 PRINT AT ty,tx;" "
 510 PRINT AT zy,zx; FLASH 1; IN
K 2;"☻"
 520 LET z(zombie,1)=zx: LET z(z
ombie,2)=zy
 525 BEEP 0.02,30-left
 530 GO TO 305
1000 PAPER 6: CLS
1010 PLOT 4,4: DRAW 248,0: DRAW
0,159: DRAW -248,0: DRAW 0,-159
1020 FOR i=2 TO 20
1030 FOR j=1 TO 30
1040 IF RND>rand THEN GO TO 1060
1050 PRINT AT i,j; INK 1;"■"
1060 NEXT j: NEXT i
1100 RETURN
2000 LET px=INT (RND*13+5)
2010 LET py=INT (RND*17+5)
2020 IF SCREEN$ (py,px)<>" " THE
N GO TO 2000
2021 IF SCREEN$ (py-1,px-1)<>" "
 THEN GO TO 2000
2022 IF SCREEN$ (py,px-1)<>" " T
HEN GO TO 2000
2023 IF SCREEN$ (py+1,px-1)<>" "
 THEN GO TO 2000
2024 IF SCREEN$ (py-1,px)<>" " T
HEN GO TO 2000
2025 IF SCREEN$ (py+1,px)<>" " T
HEN GO TO 2000
2026 IF SCREEN$ (py-1,px+1)<>" "
 THEN GO TO 2000
2027 IF SCREEN$ (py,px+1)<>" " T
HEN GO TO 2000
2028 IF SCREEN$ (py+1,px+1)<>" "
 THEN GO TO 2000
2030 PRINT AT py,px;"☨"
2100 RETURN
3000 FOR j=2 TO 20
3020 LET i=INT (RND*30+1)
3030 IF SCREEN$ (j,i)<>" " THEN
GO TO 3020
3040 PRINT AT j,i; FLASH 1; INK
2;"☺"
3050 LET left=left+1: LET z(left
,1)=i: LET z(left,2)=j
3060 NEXT j
3100 RETURN
4000 FOR i=1 TO 20
4010 PRINT AT py,px;" ": PRINT A
T ty,tx;"☨"
4020 BEEP .02,i
4030 PRINT AT py,px;"☨": PRINT A
T ty,tx;" "
4040 BEEP .02,20-i
4050 NEXT i
4060 GO TO 7060
5000 FOR i=1 TO 20
```

```
5020 PRINT AT py,px;" ": PRINT A
T ty,tx;"A"
5030 BEEP .02,20+i
5040 PRINT AT py,px;"A": PRINT A
T ty,tx;" "
5050 BEEP .02,20-i
5060 NEXT i
5070 GO TO 7060
7000 FOR i=1 TO 20
7010 PRINT AT py,px; INK 3;"B":
PRINT AT ty,tx;" "
7020 BEEP .02,20+i
7030 PRINT AT py,px;"A": PRINT A
T ty,tx;"B"
7040 BEEP .02,20-i
7050 NEXT i
7060 LET men=men-1: IF men<0 THE
N GO TO 7070
7062 FOR i=1 TO 100: NEXT i
7063 LET rand=0.1
7065 GO TO 30
7070 IF score>hi THEN LET hi=sco
re: PRINT AT 0,15;"MAX: ";hi
7075 PRINT AT 1,6;"OTRO JUEGO(Y/
N)? "
7080 IF INKEY$="y" THEN GO TO 25
7090 IF INKEY$="n" THEN BORDER 7
: PAPER 7: INK 0: STOP
7100 GO TO 7080
8000 PRINT AT 0,10; FLASH 1;"***
 PREMIO ***"
8005 LET score=score+6
8006 LET rand=rand*0.6
8010 FOR i=1 TO 3
8020 FOR j=i TO 50
8030 BEEP 0.01,j
8040 NEXT j
8050 NEXT i
8060 GO TO 30
9000 POKE USR "A",BIN 00011100
9001 POKE USR "A"+1,BIN 00011100
9002 POKE USR "A"+2,BIN 01111111
9003 POKE USR "A"+3,BIN 00001000
9004 POKE USR "A"+4,BIN 00001000
9005 POKE USR "A"+5,BIN 00010100
9006 POKE USR "A"+6,BIN 00100010
9007 POKE USR "A"+7,BIN 00000000
9010 POKE USR "B",BIN 00000000
9011 POKE USR "B"+1,BIN 01100110
9012 POKE USR "B"+2,BIN 01100110
9013 POKE USR "B"+3,BIN 00000000
9014 POKE USR "B"+4,BIN 00011000
9015 POKE USR "B"+5,BIN 01000010
9016 POKE USR "B"+6,BIN 00111100
9017 POKE USR "B"+7,BIN 00000000
9020 POKE USR "C",0
9021 POKE USR "C"+1,BIN 00111100
9022 POKE USR "C"+2,BIN 01111110
9023 POKE USR "C"+3,BIN 01100110
```

```
9024 POKE USR "c"+4,BIN 01100110
9025 POKE USR "c"+5,BIN 01111110
9026 POKE USR "c"+6,BIN 00111100
9027 POKE USR "c"+7,0
9100 BORDER 1: PAPER 6: INK 0: C
LS
9110 PRINT AT 1,11;"Clono Loco"
9120 PRINT AT 1,11; OVER 1;"____
      "
9130 PRINT AT 4,4;"Mike O'Neill
Julio 1982"
9140 PRINT AT 8,4;"Teclas:"
9150 PRINT AT 10,4;"A = arriba"
9160 PRINT AT 11,4;"S = abajo"
9170 PRINT AT 12,4;"K = izquierd
a"
9180 PRINT AT 13,4;"L = derecha"
9190 PRINT AT 14,4;"Otra tecla =
 parar"
9200 PRINT AT 19,2;"Pulse una te
cla para empezar"
9210 PAUSE 0
9900 RETURN
```

Juegos de tablero

Pirandello

Basado en el juego inventado en 1880 llamado Reversi, Pirandello se le denomina también Othelo. Reversi juega en un tablero de ocho por ocho cuadros y los jugadores pueden colocar sus dos primeras fichas en cualquier cuadro libre de los cuatro centrales. En Pirandello, en cambio, las posiciones ocupadas en los primeros cuatro cuadros están fijadas, como verá al ejecutar el programa.

A partir del momento inicial, los movimientos siguen una regla muy simple: cada pieza nueva debe ser colocada al lado de una del oponente y otra pieza del propio jugador debe estar colocada al otro extremo de piezas del oponente. Si esto se consigue, todas las piezas del oponente situadas entre las dos propias se convierten en fichas propias. La línea de piezas puede seguir cualquier dirección: horizontal, vertical o diagonal. Si la pieza del jugador completa dos líneas, ambas líneas de piezas cambiarán de propietario. El objetivo del juego es convertir en propias tantas piezas del adversario como sea posible, de forma que cuando el tablero quede lleno o no se puedan hacer más movimientos legales, usted posea más fichas que el adversario. El ganador se decide simplemente por recuento del número de fichas que hay sobre el tablero.

Para explicar el cambio de propietario de las fichas se ilustra con un ejemplo: Imagine que juega con fichas X y el computador con 0. Si hubiera una línea XOOO y usted decide colocar su ficha así XOOOX, las fichas del oponente cambiarán así XXXXX.

El juego depende más del conocimiento de qué lugares son más fuertes que de convertir muchas fichas. Las posiciones más fuertes son las cuatro esquinas. En ellas una pieza controla toda una diagonal del tablero. En cambio, los cuadros siguientes a éstos en las diagonales son probablemente los más débiles. Otros cuadros fuertes son los que rodean los cuatro iniciales.

El Spectrum conoce muy bien el valor relativo de cada cuadro y tratará de ganar las mejores posiciones. A pesar de todo, no es la mejor estrategia depender sólo de las posiciones. Siempre aparecen varias opciones y escoger una u otra puede depender de las posiciones o de la cantidad de piezas que se puedan convertir en una jugada.

No obstante, el programa está optimizado solamente por el criterio de posiciones. Con ello se consigue una respuesta más rápida y, en la mayoría de los casos, suficientemente inteligente.

La línea 3Ø envía la acción a la línea 4ØØØ, una subrutina que imprime el nombre del programa y espera que usted pulse una tecla para empezar el juego. A continuación el borde se iguala a 5, el papel a 2 y la tinta a 9 (la instrucción INK 9 selecciona en cada caso la tinta que produce el máximo contraste). Se imprime la palabra PIRANDELLO repetida en todo lo alto de la pantalla con variados colores y con cierto acompañamiento musical. Aparece luego la instrucción: «Espere un momento», mientras se DIMensionan una serie de vectores y se inicializan unas variables.

La rutina de inicialización se inicia en 5ØØØ. Se DIMensionan tres vectores. El vector A (línea 5Ø1Ø) contiene las posiciones del tablero ordenadas por orden de mérito. El vector D se usará para contener las posiciones de las piezas que deban cambiar de propietario.

Los bucles de 5Ø3Ø a 5Ø7Ø llenan el vector A con caracteres 146 (que es el código de un gráfico definido por el usuario: un cuadro). Esto se usará al imprimir el tablero para indicar posiciones vacías. El bucle siguiente (5Ø8Ø a 511Ø) lee la información de las líneas de datos 532Ø y 533Ø y la almacena en el vector E. Si observa las dos líneas de datos (DATA), verá que contienen números del 12 al 89 en un orden aparentemente aleatorio. Sin embargo, no lo es. Representa la idea del programador respecto al valor relativo de las distintas posiciones del tablero. El cuadro 18, por ejemplo, es uno de los más valiosos; por lo tanto el ordenador verificará si puede tirar antes allí que en otra parte. Los cuadros menos valiosos son los que están los últimos en la instrucción DATA de la línea 533Ø: 28, 78 y 23.

El siguiente bucle (512Ø a 515Ø) llena el vector D con los datos de la línea 534Ø (1, 9, 10, 11, −1, −9, −10, −11). Representan los desplazamientos posibles desde una posición y se usa para saber si hay que cambiar piezas después de una tirada.

Las líneas 516Ø y 517Ø colocan las primeras piezas en posición. 518Ø especifica que el primer movimiento lo hará en el ordenador inicializando la variable: quien mueve.

Inmediatamente el Spectrum define los gráficos. Se usan tres: un cuadrado que indica una posición vacía, un diamante (impreso en rojo) que representa una pieza del ordenador y una línea diagonal (impresa en blanco) que es su pieza. Los datos para estos gráficos están contenidos en las líneas 535Ø a 537Ø y la «Z» al final de la línea 537Ø activa el cambio de color del borde, el borrado de la pantalla y el paso del programa a la línea 4ØØ.

Después de imprimir el tablero (4ØØ a 49Ø) se examina quién mueve. Si el contenido de la variable Y es 144, el ordenador sabe que debe mover él. Sino, el computador cambia Y a 144 (para que la próxi-

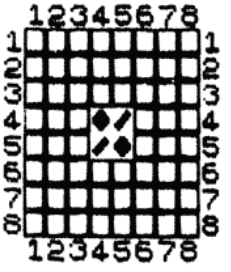
ma vez le toque mover a él) y la línea 53Ø acepta el movimiento del jugador en forma de dos números: las coordenadas vertical y horizontal. El movimiento se entra como un solo número de dos cifras (como: 25 u 81). Se rechazan los movimientos por debajo de 11 y por encima de 88.

Si no puede hacer un movimiento legal, entre un Ø para indicarlo y el ordenador irá a la línea 1ØØ a hacer su movimiento. (Si el ordenador tampoco puede mover, véase la línea 35Ø, se comprueba si previamente usted no ha podido y, si es así, el programa va a la rutina de la línea 1ØØØ que cuenta las piezas y anuncia el ganador.) La línea 56Ø examina el número entrado para ver si está entre los límites (11 y 88) y si el cuadro al que se tira está vacío (es decir, si contiene un 146).

Una vez se ha aceptado el movimiento, el ordenador va a la sección de programa que empieza en la línea 16Ø, que usa los valores de desplazamiento contenidos en el vector E para saber las piezas que deben ser cambiadas (es decir, convertidas en piezas del oponente). Al final de esta rutina, que va de las líneas 16Ø a 32Ø, se verifica quién debe tirar. Si resulta igual a 144 o el testigo de impresión de tablero (J, inicializado en la línea 16Ø y cambiado siempre que sea necesario en 29Ø) es igual a uno, el ordenador va a la línea 4ØØ donde se imprime de nuevo el tablero. Si no se han examinado todos los cuadros (si en la línea 34Ø K no es igual a 6Ø) el programa vuelve a la línea 13Ø para seguir el examen. Si se llega a la línea 35Ø, se imprime el mensaje «Paso».

La línea 5ØØ examina el valor de Y, si su contenido es 144 el ordenador debe mover. Por lo tanto, el programa vuelve a la línea 1ØØ, donde se decide el movimiento (las líneas 1ØØ a 15Ø más 34Ø). Nótese que aquí se usa la información, contenida en las instrucciones de datos, de qué cuadros deben ser examinados prioritariamente.

Encontrará PIRANDELLO un juego muy interesante y el Spectrum un oponente de alto nivel. Una vez tenga una buena marca en el juego, puede mejorarlo cambiando los gráficos y variando la ordenación de los números de cuadros más interesantes, tratando de mejorar la estrategia del ordenador y haciéndolo un oponente más difícil.



```
  10 REM Pirandello
  30 GO TO 4000
 100 LET Y=145
 110 LET Z=144
 120 LET K=0
 130 LET K=K+1
 140 LET B=E(K)
 150 IF A(B)<>146 THEN GO TO 340
 160 LET J=0
 170 FOR X=1 TO 8
 180 LET N=D(X)
 190 LET E=0
 200 LET F=B
 210 IF A(F+N)<>Y THEN GO TO 250
 220 LET E=1
 230 LET F=F+N
 240 GO TO 210
 250 IF A(F+N)<>Z THEN GO TO 310
 260 IF E=0 THEN GO TO 310
 270 FOR A=B TO F STEP N
 280 LET A(A)=Z
 290 LET J=1
 300 NEXT A
 310 NEXT X
 320 IF Y=144 THEN GO TO 400
 330 IF J=1 THEN GO TO 400
 340 IF K<>60 THEN GO TO 130
 350 IF G<>0 THEN PRINT AT 0,0;
FLASH 1; INK 2; PAPER 7;"Paso."
: BEEP 1,1: BEEP 1,2: BEEP 1,1:
PRINT AT 0,0;"      "
 380 IF G=0 THEN GO TO 1000
 400 PRINT AT 6,11; BRIGHT 1;"12
345678"
 410 BEEP 0.05,32: BEEP 0.05,12:
 BEEP 0.05,32
 420 FOR A=1 TO 8
 430 PRINT TAB 10; BRIGHT 1; INK
 6;A;
 440 FOR B=2 TO 9
 445 LET W=A(A*10+B): BEEP 0.008
,(W=144)+50*(W=145)-60*(W=146)
 450 LET R=2*(W=144)+7*(W=145)+4
*(W=146)
 455 PRINT BRIGHT 1; INK R;CHR$
(W);
 460 NEXT B
 470 PRINT BRIGHT 1; INK 6;A
 480 NEXT A
 490 PRINT TAB 11; INK 6; BRIGHT
 1;"12345678"
 500 IF Y=144 THEN GO TO 100
 510 LET Y=144
 520 LET Z=145
 530 INPUT INK 2; PAPER 6; FLASH
 1; BRIGHT 1;" Entre su movimien
to  ";G
 540 IF G=0 THEN GO TO 100
 550 LET B=G+1
```

```
 550 IF B<12 OR B>89 OR A(B)<>14
6 THEN GO TO 530
 590 GO TO 160
1000 LET C=0
1010 LET H=0
1020 FOR A=1 TO 100
1030 IF A(A)=144 THEN LET C=C+1
1040 IF A(A)=145 THEN LET H=H+1
1050 NEXT A
1055 PRINT AT 18,0; FLASH 1; BRI
GHT 1; INK 2; PAPER 7;"  Puntuac
ion final:","Spectrum: ";C,,"Ust
ed:     ";H,
1060 IF C>H THEN PRINT AT 0,0; B
RIGHT 1; FLASH 1; INK 4;"Gane..g
racias por su buen juego": GO TO
 1060
1070 IF C<H THEN PRINT AT 0,0; B
RIGHT 1; FLASH 1; INK 4;"Gano..g
racias por su buen juego": GO TO
 1070
1080 IF C=H THEN PRINT AT 0,0; B
RIGHT 1; FLASH 1; INK 4;"Empate.
....gracias por el juego": GO TO
 1080
1090 STOP
4000 BORDER 5: PAPER 2: INK 9: C
LS
4010 FOR A=1 TO 22: PRINT TAB A;
 INK A/5+3;"Pirandello"
4020 BEEP 0.008,A: BEEP 0.008,A*
2: NEXT A
4030 PRINT AT 5,2; FLASH 1; INK
7; PAPER 0;" Por favor espere un
 momento "
5000 PAPER 0: INK 9
5010 DIM A(100): DIM E(60)
5020 DIM D(8)
5030 FOR A=1 TO 8
5040 FOR B=2 TO 9
5050 LET A(A*10+B)=145
5060 NEXT B
5070 NEXT A
5080 FOR A=1 TO 60
5090 READ B
5100 LET E(A)=B
5110 NEXT A
5120 FOR A=1 TO 8
5130 READ B
5140 LET D(A)=B
5150 NEXT A
5160 LET A(45)=144: LET A(46)=14
5
5170 LET A(56)=144: LET A(55)=14
5
5180 LET Y=144
5270 READ A$: IF A$="Z" THEN BOR
DER 0: CLS : GO TO 400
5280 FOR C=0 TO 7
```

```
5290 READ B: POKE USR A$+C,B
5300 NEXT C
5310 GO TO 5270
5320 DATA 19,81,82,12,62,17,32,8
7,69,14,39,84,64,67,37,34,49,15,
59,85,16,52,42,86,65,54,66,44,35
,57,36,47
5330 DATA 63,48,58,76,24,27,38,2
5,74,77,43,26,68,33,53,75,72,13,
29,18,88,22,83,79,73,28,78,23
5340 DATA 1,9,10,11,-1,-9,-10,-1
1
5350 DATA "A",0,24,60,126,126,60
,24,0
5360 DATA "B",0,4,14,28,56,112,3
2,0
5370 DATA "C",255,129,129,129,12
9,129,129,255,"Z"
```

Damas

Desafíe ahora a su Spectrum a jugar a DAMAS, uno de los juegos más populares después del ajedrez. Algunos historiadores de los juegos afirman que el juego de las DAMAS es un antecesor del ajedrez. En realidad, en el antiguo Egipto, en Grecia y en Roma se jugaban juegos parecidos. Después de una evolución de siglos, el juego alcanzó su forma actual a mediados del siglo XVII.

Existen muchas variaciones regionales de este juego, pero la versión propuesta es la que se podría considerar más representativa. Las DAMAS se juega entre dos oponentes, utilizando cuadrados alternados del tablero de ajedrez, por lo general los negros. Para hacerlo más fácilmente visible en la pantalla del televisor (y para poderlo copiar en impresora, si utiliza la función COPY para copiar el contenido de la pantalla) el juego utiliza los cuadrados blancos del tablero.

Cada bando empieza con 12 piezas. En esta versión usted tiene las pequeñas «x» y el ordenador las «o» (sin emargo, una vez conocido el juego, puede cambiar los gráficos). El ordenador tiene situadas sus fichas en la parte alta de la pantalla al comienzo de la partida, mientras que usted las tiene situadas en la parte inferior.

Unicamente se puede mover en diagonal, siguiendo los cuadros blancos. Si existe un enemigo en la diagonal e inmediatamente en frente de usted y un sitio libre a continuación, puede capturar esta pieza saltando por encima de ella (la pieza capturada desaparece del tablero) y aterrizando en el siguiente cuadro libre. Puede capturar más de una pieza a la vez si después de aterrizar en el cuadro libre es posible otra captura desde este punto.

En ciertas variantes del juego de las DAMAS existe la regla del «soplado», que establece que si puede hacerse una captura, debe hacerse. No existe tal regla, como tal, en este juego, pero el ordenador siempre capturará; si puede, por lo tanto, usted debe hacer lo propio si quiere ganar la partida.

El objetivo del juego es capturar todas las piezas del oponente o inmovilizarlas. En este programa, el Spectrum tenderá a jugar hasta el final, pero en ciertas circunstancias puede reconocer que está batido, entonces se rinde y le concede la partida. Usted puede rendirse en cualquier momento, entrando un 99 en su turno de movimiento. Si lo hace, Spectrum aceptará su derrota.

Si consigue llegar con una de sus piezas al extremo contrario del tablero, la pieza es coronada y se convierte en una DAMA. Esto se muestra en el programa con el cambio de gráfico de la pieza. Más adelante se explica brevemente cómo puede cambiar estos gráficos según sus gustos. Una DAMA tiene el doble de movilidad que una pieza normal. Puede moverse adelante y atrás, convirtiéndose en una pieza formidable. Nótese que, si aterriza en la última línea después de una captura, no puede en este mismo instante coronarla y seguir con sus capturas en dirección contraria. Su pieza no se convertirá en una DAMA hasta la siguiente jugada. El ordenador conoce esta regla y la aplica esperando un movimiento antes de coronar sus fichas.

Verá que el ordenador juega bastante rápido y con un grado de inteligencia y anticipación sorprendentes, lo que significa que deberá concentrarse para ganarle. Para añadir interés, el ordenador imprime en pantalla cada jugada, haciendo un sonido especial en las jugadas que considera más interesantes. El ordenador examina el tablero cuadro por cuadro desde la parte superior izquierda y toda la fila superior, pasando por cada línea de izquierda a derecha, hasta el final. Cuando encuentra una pieza propia, comprueba si puede realizar una captura. Si es así, la hace sin pensarlo más, reimprime el tablero y le permite entrar su respuesta. Si no puede realizar capturas, busca posibles movimientos, comprobando que no le sean peligrosos. También tiene cierta facilidad para elaborar movimientos que protejan piezas que se encuentren en peligro.

Después de haber recorrido el tablero y de haber almacenado los movimientos seguros en una matriz, examina si es posible obtener una DAMA con un solo movimiento. Si es posible tal movimiento, se hace, ya que el ordenador concede más importancia a obtener una DAMA que a hacer un movimiento seguro. Si no es posible, el ordenador escoge al azar su movimiento entre los almacenados. Al hacerlo aleatoriamente, el ordenador puede jugar una serie de partidas sin repetirse, aunque usted trate de forzarlo a un juego idéntico a uno que haya sido ya jugado.

Si el examen del tablero revela que no existen movimientos seguros, escoge una pieza al azar y la mueve. Tal movimiento ocurre raramente y sólo al principio o al final de las partidas. Si el ordenador dispone de muy pocas fichas en el tablero, puede considerar que su situación es desesperada y concederle la victoria.

Como se ha dicho, el programa tiene una elevada velocidad de respuesta. Ello es debido a que, en la mayoría de los movimientos, sólo se examina parte del tablero o, en el peor de los casos, todo el tablero, pero una sola vez. El tiempo de ejecución aumenta debido a las líneas que se imprimen de los movimientos que el ordenador considera. No obstante, esto añade interés al juego y se ha creído conveniente mantenerlas aunque lo hagan algo más lento. Si prefiere que el programa responda más rápidamente, puede borrar ciertas líneas del mismo, que le serán indicadas al analizarlo. También existen ciertos comentarios, tales como «¡Buena jugada!» o «¡Lo consiguió!»; que aparecen de cuando en cuando. Estas informaciones ralentizan un poco el programa y pueden ser eliminadas. Quizá, como ha pasado a muchos usuarios del programa, prefiera disponer de dos versiones. En este caso, grabe el programa entero y después borre las líneas superfluas, con lo que dispondrá de una copia que se ejecuta a la máxima velocidad. Cada vez que juegue podrá escoger entre ambas.

Después de que se ha activado el generador de números aleatorios en la línea 5Ø, la acción va a la subrutina de la línea 1Ø7Ø, donde se inicializan las variables. Si se analiza esta subrutina, se verá que empieza por determinar los colores del papel y del borde (azul) y el de la tinta (amarillo). Como en todos los programas contenidos en este libro, usted puede cambiar estos colores a su gusto y hallar la combinación que dé mejores resultados en su televisor. Aparece el mensaje «Espere un momento» en la parte superior de la pantalla, mientras se ejecuta el proceso de inicialización. (Puede considerar la inclusión de una línea tal como está en todo programa que tenga un proceso de inicialización, para informar al utilizador que ocurre algo cuando se ha pulsado RUN y ENTER.)

La línea 1Ø8Ø asigna valores a las piezas. Los nombres de las variables son: C, pieza del ordenador; H, pieza del usuario; CK, dama del ordenador; HK, dama del usuario; E, espacio en blanco; B, cuadro negro. La asignación de estas variables permite su utilización a lo largo del programa sin redefinición. Se ha hecho así para que pueda cambiar la representación gráfica del programa cambiando simplemente estas líneas. Si desea añadir gráficos definidos por el usuario, debe igualar, por ejemplo, H a 144 (código del gráfico uno, definible en la tecla A con el modo GRAPHICS), C a 145, etcétera.

La línea 1Ø8Ø inicializa la variable OF que representa «fuera del tablero». El ordenador se construye una fila de cuadros invisibles alrededor del tablero para evitar la captura de una ficha colocada en el

extremo. Si no se hiciera así, el ordenador podria suponer que hay un espacio libre detrás de una ficha situada en esta posición.

Una vez se ha definido la variable OF, se dimensiona la matriz Q (que contiene las piezas) y cada elemento de la matriz se llena con un espacio OF. El bucle siguiente (111Ø a 113Ø) usa los datos contenidos en las líneas 114Ø a 123Ø para llenar los cuadrados del tablero con los valores iniciales. Todos los elementos de la matriz no asignados mantienen el valor OF (−99), por lo tanto el ordenador sabe donde no debe perderse.

La línea 124Ø es un bucle que llena el vector N (ya dimensionado en la línea 1Ø9Ø) con números que representan los posibles movimientos desde cada cuadro sin capturar ninguna ficha. El ordenador sabe que doblando estos números, cuando sea necesario, obtiene el movimiento con captura. En la línea 126Ø se inicializan las variables que cuentan el número de fichas capturadas al oponente: CO, para el ordenador; HU para el usuario.

Al volver de la subrutina a la línea 9Ø aparece el comentario «Borre la siguiente línea para evitar que el Spectrum mueva primero». Si desea mover primero, debe borrar la línea 1ØØ. No es necesario nada más en el programa para conseguirlo. El Spectrum no es el mejor jugador de DAMAS del mundo, por lo tanto, lo mínimo que puede hacer por él es dejarle tirar primero. El programa transcurre a través de un bucle de subrutinas del 11Ø a 14Ø y la línea 15Ø lo reinicia una y otra vez. Las subrutinas que soportan el programa principal son:

68Ø imprime el tablero
85Ø acepta el movimiento del usuario
68Ø reimprime del tablero después del movimiento del usuario
16Ø inicia la investigación para el movimiento del computador.

Se empezará observando el proceso de investigación porque es la sección más interesante del programa. La línea 16Ø DIMensiona el vector S, que se usará para almacenar los movimientos seguros que descubra el ordenador en su búsqueda de capturas. La variable SC, el contador de movimientos seguros, se iguala a cero en la línea 17Ø, donde también se iguala la variable A (cuadro que se considera) a 89, uno más que el elemento 88, colocado en la parte superior izquierda del tablero. La línea 18Ø decrementa A, para indicar el cuadro de la parte superior izquierda, y la línea 19Ø lo examina para determinar si es una pieza ordinaria del ordenador (es decir, ve si este elemento de la matriz contiene el valor C) o una DAMA del ordenador (valor CK). La línea 2ØØ inicializa la variable B (que indica qué movimiento se considera desde este cuadro) a cero. Si A es menor que 29, el ordenador sabe que está en la última línea de su lado y que, por lo tanto, sólo

puede moverse una posición hacia adelante (los valores 3 y 4 indican estos movimientos), despreciando la consideración de los valores 1 y 2 de la variable B.

La línea 21Ø incrementa B (que se convierte en 1 o 3 si se considera una pieza situada en la última línea) y 22Ø determina M (que representa el cuadro *al* que apunta el ordenador, mientras que A es el cuadro *desde* el que se piensa mover el ordenador) igual a A *más* el elemento del vector N que indica el valor de B. Si M es mayor que 88 o menor que 11, el ordenador sabe que la ficha está fuera del tablero y no requiere ser considerada. La línea 24Ø es una de las que puede ser borrada si desea máxima velocidad: imprime el movimiento que está siendo considerado.

La próxima línea puede encontrar una captura. Si el cuadro al que se piensa ir está lleno (con una ficha del usuario «H» o una DAMA del usuario «HK») y el cuadro siguiente está vacío (E), el ordenador sabe que puede efectuar una captura. Si es así, la acción se dirige a la línea 32Ø, donde se muestra el mensaje. Lo conseguí!, intermitentemente. Como antes, si desea una ejecución a máxima velocidad, puede borrar las líneas 33Ø, 34Ø y 35Ø.

Siguiendo con el programa principal de movimiento del computador, la línea siguiente (26Ø) examina la peligrosidad del movimiento, viendo si la ficha movida al cuadro considerado puede ser objeto de una captura o si deja al descubierto otra ficha propia. Si la pieza pasa la larga serie de IF que configura el examen mencionado, el Spectrum va a la subrutina que empieza en la línea 47Ø. El ordenador puede almacenar hasta diez movimientos seguros (ya que el vector que los contiene no tiene más posiciones y que es el número que se considera más adecuado). El contador de movimientos seguros (SC), que ha sido igualado a cero en 17Ø, es incrementado, si su contenido es menor que diez. La línea 48Ø (que puede ser borrada) imprime «Movimiento seguro: de ... a ...» y suena un BEEP para subrayar la existencia de un movimiento seguro.

La línea 49Ø es muy interesante. Empaqueta las variables A y M en un único número que se almacena en el siguiente lugar libre del vector S. De esta forma, si es necesario, puede decodificarse. La línea 49Ø devuelve la acción a la parte principal del programa de búsqueda, pero se seguirá analizando la línea 5ØØ para ver cómo se realiza un movimiento seguro. Si no ha habido captura, el ordenador va a la línea 5ØØ. Si SC es igual a Ø, no existe ningún movimiento seguro y va a la línea 55Ø a seleccionar aleatoriamente un movimiento. Si, por el contrario, SC es mayor que cero, el programa sigue en la próxima línea y se selecciona uno de los números codificados en el vector S. Este número se almacena en XC y se decodifica, recapturando A en la línea 52Ø y M en la 53Ø. La línea 54Ø envía el programa a la línea 65Ø, donde se realiza el movimiento.

Volviendo a la primera sección del programa, se llega a la línea 27Ø, donde se incrementa B. El primero y segundo elementos del vector N contienen los movimientos que pueden ser realizados por cualquier pieza (excepto una que está situada en la línea inferior) y el tercero y cuarto elementos del citado vector contienen los movimientos a realizar por las DAMAS. Si B, el contador que selecciona los elementos del vector N que serán usados para efectuar los movimientos, es menor que su máximo (dos para una pieza ordinaria y cuatro para una DAMA), el programa vuelve a 21Ø, donde se incrementa el contador B y sigue la búsqueda. La línea 28Ø examina si se han analizado todos los cuadros del tablero y sino vuelve a la línea 18Ø, donde el contador de cuadros, A, se decrementa y la búsqueda sigue. No obstante, si se han utilizado todos los movimientos (es decir, si A es menor que doce en la línea 28Ø), el programa ejecuta las dos líneas siguientes, donde se examina la posibilidad de obtener una DAMA. El indicador FL se iguala a cero y el computador examina los elementos del vector que representan los cuadrados de la fila *anterior* a la que convierte las fichas ordinarias en DAMAS. Si encuentra una ficha propia en esta línea, el programa va a la subrutina de la línea 127Ø. Si esta subrutina encuentra un movimiento para poder hacer la DAMA, iguala FL a uno y envía la acción a la línea 65Ø para ejecutar realmente el movimiento.

Si tal movimiento no es posible, la línea 31Ø salta la sección donde se hace el movimiento de captura y va a la línea 5ØØ. Conviene recordar el orden de preferencia que sigue el ordenador:

- ¿Puedo capturar, y si es así, puedo volver a hacerlo?
- ¿Puedo hacer una DAMA?
- ¿Puedo hacer un movimiento que no exponga la pieza movida o a otra pieza?
- ¿Puedo hacer un movimiento legal?

Si la respuesta a una de las preguntas es «sí», se realiza el movimiento y se reimprime el tablero, quedando listo para recibir su movimiento. Si la respuesta es «no», se considera la siguiente pregunta. Si la respuesta a la última pregunta también es «no», el ordenador le concede la victoria. (Existe otra pregunta que se realiza cada vez que se reimprime el tablero: ¿cuántas piezas ha capturado cada jugador? Si resulta que algún jugador ha capturado 12 piezas, el juego entra en una subrutina que reconoce quién es el ganador y termina el juego.) Este tipo de ordenación de preguntas es típica de muchos juegos y es la que da más agilidad y habilidad al ordenador.

Si el ordenador llega a la línea 5ØØ, la respuesta a las dos primeras preguntas ha sido «no». Para formular la tercera (¿Puedo hacer un movimiento seguro?) el ordenador debe consultar el valor de SC (el

contador de movimientos seguros). Si es cero, no ha podido encontrar movimientos que se puedan considerar seguros, por lo tanto, la respuesta es «no». Se ha examinado ya el sistema para reconocer los movimientos seguros. Ahora el ordenador dispone sólo de un último recurso: escoger un movimiento legal aleatoriamente; lo hace a partir de la línea 55Ø.

Aquí la variable SC es otra vez útil para evitar la inicialización de una nueva variable. Después de todo, se inicializará a cero (línea 17Ø) antes de un nuevo movimiento, por ello no resulta peligrosa su reutilización aquí. La línea 55Ø incrementa SC y escoge un cuadro aleatoriamente entre 1 y 88. Si este cuadro no contiene una pieza del ordenador o del oponente (56Ø), la acción va a la línea 63Ø, donde, si se han escogido menos de 3ØØ números en esta sección, la acción se dirige a la línea 55Ø para intentarlo otra vez.

Por el contrario, si el cuadro escogido contiene una ficha ordinaria o una DAMA del ordenador, la rutina de las líneas 57Ø a 62Ø comprueba si se puede mover. Si es así, la línea 61Ø envía el programa a 65Ø, donde se realiza el movimiento. Si no encuentra un movimiento y han sido escogidos ya 3ØØ números aleatoriamente, el ordenador le concede la victoria en la línea 64Ø.

Antes de analizar la sección donde se permite la entrada de su movimiento, se examina la sección que comienza en la línea 37Ø. Se llega a esta línea después de conseguir una captura. La línea anterior ha incrementado la puntuación del ordenador (CO) antes de que 37Ø llame a la subrutina de impresión del tablero que muestra la pieza movida, la nueva puntuación del Spectrum y, por supuesto, un espacio en blanco donde estaba la ficha capturada. Como recordará, A es el número del cuadro desde el que se ha movido el ordenador y M el del cuadro al que se ha movido. Ahora A se iguala al cuadro al que se ha movido el ordenador y las líneas 39Ø a 46Ø examinan si es posible hacer una nueva captura. Si es así, la lleva a cabo, volviendo a comprobar lo mismo después de cada impresión del tablero. Cuando se ha terminado la secuencia de capturas, el ordenador pasa el turno a su oponente y le permite hacer su movimiento.

La sección de entrada de sus movimientos es simple. Usted mueve especificando el cuadro desde el que quiere moverse, indicando primero la fila y luego la columna como si se tratase de un solo número (tal como: 31). Inmediatamente pulse ENTER y ahora puede entrar dos números más que indiquen el cuadro a donde quiere aterrizar. Se ejecuta entonces el movimiento, se reimprime el tablero y si ha conseguido una captura, se le preguntará si puede saltar de nuevo. Si responde «no», el ordenador pensará su movimiento. Si responde «sí», deberá entrar el nuevo cuadro al que desea ir (por supuesto, el ordenador sabe ya qué ficha debe mover).

La sección que acepta sus demandas empieza en la línea 81Ø,

donde los comentarios REM le indican que puede terminar el juego entrando 99. Entrando un «1» ordenará al ordenador la copia de la pantalla en la impresora. Después de la copia, se le volverá a requerir su movimiento. Después de hacerlo en el vector que contiene las piezas (91Ø), dos bucles (92Ø a 95Ø) examinan si alguna pieza del jugador ha alcanzado el *estatus* de DAMA y, si es así, cambia el símbolo gráfico adecuadamente. Recuerde que, como ya se ha mencionado, una pieza no se comportará como DAMA hasta una jugada después de su coronación. En esta misma sección se coronan también las fichas del ordenador para evitar que sus fichas coronadas fueran movidas sin seguir la citada regla. No existe en el programa nada que le impida hacer trampas, ya que el ordenador ni sueña que esto pueda hacerse. A pesar de eso, una trampa puede parar el programa y obligarle a comenzar una nueva partida.

En la línea 96Ø el ordenador determina la diferencia entre los cuadros entrados. Si la diferencia es menor que 12, usted no ha realizado una captura, entonces el ordenador vuelve al bucle principal en el principio del programa para reimprimir el tablero antes de que el ordenador seleccione un nuevo movimiento. Por el contrario, si la diferencia no es menor que 12, se ha realizado una captura y el ordenador anunciará, dos veces de cada tres, esta captura con un comentario generado por las líneas 97Ø a 99Ø. Con comentario o sin él, se produce un tono creciente de sonido producido en la línea 995, se incrementa su puntuación (HU) en la línea 1ØØØ, el cuadro capturado se vacía (línea 1ØØØ), se reimprime el tablero (línea 1Ø3Ø) y se le da la oportunidad de efectuar otra captura. Si no lo hace, la línea 1Ø4Ø devuelve el programa al bucle principal, evitando pasar por la línea 1Ø6Ø que iguala el anterior cuadro de destino al nuevo cuadro de situación y envía el programa a la línea 86Ø para aceptar un nuevo cuadro de destino.

Se ha analizado ya el resto del programa, que controla las rutinas de inicialización y obtención de dama. La única parte del programa que no se ha descrito, y que se explica por sí misma, es la que está comprendida entre las líneas 68Ø y 78Ø, que imprime el tablero después de cada movimiento. Obsérvese que la línea 77Ø vuelve al programa principal si la puntuación del ordenador o la suya llegan a 12.

Esta es una muestra del tablero al principio de la partida y al cabo de unas cuantas jugadas.

```
                    57


      Spectrum: 0    Usted: 0

```


```
Movimiento seguro  77 hacia 68

      Spectrum: 2    Usted: 2

```


```
   10 REM Damas Spectrum
   50 RANDOMIZE
   80 GO SUB 1070
   90 REM Borre la proxima linea
si no quiere que el Spectrum emp
ieze primero
  100 GO TO 130
  110 GO SUB 680
  120 GO SUB 850
  130 GO SUB 680
  140 GO SUB 160
  150 GO TO 110
  160 DIM S(10)
  170 LET SC=0: LET A=69
  180 LET A=A-1
```

```
 190 IF Q(A)<>C AND Q(A)<>CK THE
N GO TO 280
 200 LET B=0: IF A<29 THEN LET B
=2
 210 LET B=B+1
 220 LET M=A+N(B)
 230 IF M>88 OR M<11 THEN GO TO
280
 240 IF 10*INT (M/10)<>M THEN PR
INT AT 0,0;A;" hacia ";M;"?"
 250 IF (Q(M)=H OR Q(M)=HK) AND
Q(M+N(B))=E THEN GO TO 320
 260 IF Q(M)=E THEN IF (Q(M-11)<
>H AND Q(M-11)<>HK) THEN IF (Q(M
-9)<>H AND Q(M-9)<>HK) AND Q(M+9
)<>HK THEN IF ((Q(M+22)<>HK OR Q
(M+18)<>HK) AND (Q(M+9)<>C OR Q(
M+9)<>CK OR Q(M+11)=C OR Q(M+11)
=CK)) AND Q(M+11)<>HK THEN GO SU
B 470
 270 IF B<2 OR (Q(A)=CK AND B<4)
 THEN GO TO 210
 280 IF A>11 THEN GO TO 180
 290 LET FL=0: IF Q(22)=C OR Q(2
4)=C OR Q(26)=C OR Q(28)=C THEN
GO SUB 1270
 300 IF FL=1 THEN GO TO 650
 310 GO TO 500
 320 LET Q(M+N(B))=Q(A): LET Q(M
)=E: LET Q(A)=E
 330 PRINT AT 0,0; FLASH 1; INK
3;"******************************
"
 340 FOR T=-10 TO 55 STEP 3: BEE
P .008,T: NEXT T
 350 PRINT AT 0,0; INK 2; PAPER
7; FLASH 1; BRIGHT 1;"          Le
cace.......          ": PAUSE 120:
 PRINT AT 0,0;"
                              "
 360 LET CO=CO+1
 370 GO SUB 680
 380 LET A=M+N(B)
 390 LET B=0
 400 LET B=B+1
 410 IF (A+2*N(B)<11 OR A+2*N(B)
>88) AND B<4 THEN GO TO 400
 420 LET M=A+N(B)
 430 IF Q(M)=C AND B>3 THEN RETU
RN
 440 IF (Q(M)=H OR Q(M)=HK) AND
Q(M+N(B))=E THEN GO TO 320
 450 IF B<2 OR (Q(A)=CK AND B<4)
 THEN GO TO 400
 460 RETURN
 470 IF SC<10 THEN LET SC=SC+1
 480 PRINT AT 0,0; INVERSE 1;"Mo
vimiento seguro: ";A;" hacia ";M
: BEEP 0.5,SC: PRINT AT 0,0;"
                              "
```

```
 490 LET S(SC)=100*A+B+20: RETUR
N
 500 IF SC=0 THEN GO TO 550
 510 LET XC=INT (RND*SC)+1
 520 LET A=INT (S(XC)/100)
 530 LET M=A+N(S(XC)-100*A-20)
 540 GO TO 650
 550 LET SC=SC+1: LET A=INT (RND
*88)+1
 560 IF Q(A)<>C AND Q(A)<>CK THE
N GO TO 630
 570 LET B=0
 580 LET B=B+1
 590 LET M=A+N(B)
 600 IF M>88 OR M<11 THEN GO TO
620
 610 IF Q(M)=E THEN GO TO 650
 620 IF B<2 OR Q(A)=CK AND B<4 T
HEN GO TO 580
 630 IF SC<300 THEN GO TO 550
 640 PRINT AT 0,0;"Le concedo la
 victoria": STOP
 650 LET Q(M)=Q(A): LET Q(A)=E
 660 PRINT AT 0,0;"Desde ";A;" h
acia ";M: FOR T=30 TO -30 STEP -
2: BEEP .008,T: NEXT T: PRINT AT
 0,0;"                "
 670 RETURN
 680 PRINT AT 4,2; BRIGHT 1; INK
 6;" Spectrum: ";CO;"    Usted: "
;HU;" ";;;
 710 PRINT  TAB 6; BRIGHT 1; INK
 6;"12345678"
 720 FOR F=80 TO 10 STEP -10
 730 PRINT TAB 5; BRIGHT 1;F/10;
 740 FOR G=1 TO 8: PRINT CHR$ Q(
F+G);: NEXT G
 750 PRINT F/10: NEXT F
 760 PRINT TAB 6; BRIGHT 1; INK
6;"12345678"
 770 IF CO=12 OR HU=12 THEN GO T
O 790
 780 RETURN
 790 IF HU=12 THEN PRINT AT 0,0;
"Usted ha ganado...Gracias por e
l",,"juego": STOP
 800 IF CO=12 THEN PRINT AT 0,0;
"He ganado...Gracias por el",,"j
uego": STOP
 810 REM 99 para CONCEDER
 820 REM 1 para COPIAR TABLERO
 850 INPUT FLASH 1;" Desde? ";A
 860 IF A=99 THEN PRINT AT 0,0;"
Gracias por el juego": STOP
 870 IF A=1 THEN COPY : GO TO 85
0
 900 INPUT FLASH 1;(A);" Hacia ?
 ";B
 910 LET Q(B)=Q(A): LET Q(A)=E
```

```
 920 FOR T=11 TO 17: IF Q(T)=C T
HEN LET Q(T)=CK
 930 NEXT T
 940 FOR T=82 TO 88: IF Q(T)=H T
HEN LET Q(T)=HK
 950 NEXT T
 960 IF ABS (A-B)<12 THEN RETURN

 970 LET TY=RND
 980 IF TY<0.3 THEN PRINT AT 0,0
;"Buen movimiento"
 990 IF TY>0.7 THEN PRINT AT 0,0
;"Me ha cazado!"
 995 FOR T=-20 TO -1 STEP 0.7: B
EEP .01,T: NEXT T: PRINT AT 0,0;
1000 LET HU=HU+1: LET Q((A+B)/2)
=E: GO SUB 680
1010 FOR T=82 TO 88: IF Q(T)=H T
HEN LET Q(T)=HK
1020 NEXT T
1030 INPUT FLASH 1;"Puede saltar
 otra vez? (Y/N) ";A$
1040 IF A$<>"Y" AND A$<>"y" THEN
 RETURN
1060 LET A=B: GO TO 860
1070 PAPER 1: BORDER 1: INK 7: C
LS
1075 PRINT AT 0,0;"Espere un mom
ento"
1080 LET H=120: LET HK=75: LET C
=111: LET CK=36: LET E=32: LET B
=143
1090 LET OF=-99: DIM Q(99): DIM
N(4)
1100 FOR M=1 TO 99: LET Q(M)=OF:
 NEXT M
1110 FOR M=1 TO 64
1120 READ D: READ G
1130 LET Q(D)=G: NEXT M
1140 DATA 81,B,82,C,83,B,84,C,85
,B,86,C,87,B
1150 DATA 88,C,71,C,72,B,73,C,74
,B,75,C,76,B
1160 DATA 77,C,78,B,61,B,62,C,63
,B,64,C
1170 DATA 65,B,66,C,67,B,68,C,51
,E,52,B
1180 DATA 53,E,54,B,55,E,56,B,57
,E,58,B
1190 DATA 41,B,42,E,43,B,44,E,45
,B,46,E
1200 DATA 47,B,48,E,31,H,32,B,33
,H,34,B,35,H
1210 DATA 36,B,37,H,38,B,21,B,22
,H,23,B,24,h
1220 DATA 25,B,26,H,27,B,28,H,11
,H,12,B,13,H
1230 DATA 14,B,15,H,16,B,17,H,18
,B
```

```
1240 FOR M=1 TO 4: READ N(M): NE
XT M
1250 DATA -11,-9,11,9
1260 LET CO=0: LET HU=0: RETURN
1270 IF Q(22)=C AND Q(11)=E THEN
 LET A=22: LET M=11: LET FL=1: R
ETURN
1280 IF Q(22)=C AND Q(13)=E THEN
 LET A=22: LET M=13: LET FL=1: R
ETURN
1290 IF Q(24)=C AND Q(13)=E THEN
 LET A=24: LET M=13: LET FL=1: R
ETURN
1300 IF Q(24)=C AND Q(15)=E THEN
 LET A=24: LET M=15: LET FL=1: R
ETURN
1310 IF Q(26)=C AND Q(15)=E THEN
 LET A=26: LET M=15: LET FL=1: R
ETURN
1320 IF Q(26)=C AND Q(17)=E THEN
 LET A=26: LET M=17: LET FL=1: R
ETURN
1340 RETURN
```

TIC-TAC-TOE

TIC-TAC-TOE (puntos y cruces) es un juego muy conocido en el que usted y su oponente colocan sus piezas (puntos y cruces) alternativamente en los sitios vacíos de un tablero de tres por tres posiciones, tratando de colocar tres en línea recta en cualquier dirección: vertical, horizontal o diagonal. El juego ha sido tan bien analizado, que al jugarlo se producen muy pocas sorpresas, al contrario de la mayoría de los programas de puntos y cruces, que son muy predecibles y donde usted a lo sumo, puede empatar; este programa ha sido construido con el objetivo de que sea poco predecible, pero igualmente brillante. A parte de tomar el cuadro del centro si es posible, los movimientos del Spectrum son aleatorios (excepto, por supuesto, la terminación de una posible línea de tres y el bloqueo de una línea que usted pudiera hacer). El programa permite tirar primero a uno cualquiera de los jugadores (esta decisión se toma en la línea 5Ø).

Si la función RND de la línea 5Ø es mayor de Ø,5 el ordenador decide tirar primero y, después de una pausa, va a la línea 9Ø, que envía la acción a la línea 7ØØ, donde una subrutina se encarga de imprimir el tablero. Volviendo de la citada rutina, el ordenador va a la de la línea 43Ø que es una rutina para examinar el posible ganador de la partida. Esta rutina continúa hasta la línea 55Ø.

Siguiendo en el bucle principal, el ordenador examina el cuadro central (número cinco). Si encuentra libre esta posición, coloca allí su

ficha y (tal como indica el final de la línea) va inmediatamente a la línea 6Ø, donde se llama a la rutina de impresión del tablero antes de aceptar su jugada.

Si el cuadro central no está libre, se imprime el mensaje «Espere un momento, por favor»; comprueba si existe un ganador y ejecuta la rutina de 14Ø a 28Ø, si no lo encuentra, examina si el contrario dispone de dos fichas en línea que puedan ser convertidas en tres. De no ser así, el ordenador escoge entre hasta 2Ø movimientos aleatorios, buscando su jugada. Si no encuentra ninguna jugada, examina cada cuadro secuencialmente (con la rutina de 35Ø a 38Ø). Si no encuentra ningún cuadro libre, sabe que la partida ha terminado en empate, ya que antes ha examinado si existía un ganador. La línea 4ØØ imprime el resultado. Después de una corta sinfonía (la rutina de 59Ø a 6ØØ), se reinicia el programa.

La rutina de movimiento del jugador de la línea 62Ø le permite entrar sus tiradas mediante la función INKEY$, de forma que no es necesario pulsar ENTER después de seleccionar un cuadro.

El juego está totalmente ejecutado mediante un ciclo sin fin de la línea 6Ø a la 1ØØ, imprimiendo el tablero, aceptando sus tiradas, imprimiendo el tablero, examinando la existencia de un posible ganador, haciendo una tirada, etc; hasta que termina el juego. La ventaja de hacerlo así es que usted puede modificar las distintas partes del programa, mejorándolas a su gusto sin necesidad de afectar al resto del programa.

Si desea mejorar el programa (aunque sea a costa de que el programa resulte más predecible), puede actuar en la sección de búsqueda aleatoria de las jugadas, predeterminando el orden para examinar los cuadros (desde la línea 29Ø).

```
  10 REM TIC TAC TOE
  20 DIM A(9)
  30 RANDOMIZE
  40 BORDER 1: INK 7: PAPER 1: C
LS
  50 IF RND>0.5 THEN PRINT AT 5,
5; FLASH 1;"Yo tirare primero..
.": FOR E=1 TO 50: BEEP 0.08,E: N
EXT E: CLS : GO TO 90
  60 GO SUB 700
  70 GO SUB 430
  80 GO SUB 620
  90 GO SUB 700
 100 GO SUB 430
 110 IF A(5)=0 THEN LET A(5)=1:
GO TO 60
 120 REM PARA COMPLETAR FILA/BLO
QUE
 130 PRINT AT 1,1; FLASH 1;"Espe
re un momento"
```

```
 140 LET D=1
 150 LET B=1
 160 IF B=1 THEN LET X=1: LET Y=
2: LET Z=3
 170 IF B=2 THEN LET X=1: LET Y=
4: LET Z=7
 180 IF B=3 THEN LET X=1: LET Y=
5: LET Z=9
 190 IF B=4 THEN LET X=3: LET Z=
7
 200 LET C=1
 210 IF A(X)=D AND A(Y)=D AND A(
Z)=0 THEN LET A(Z)=1: GO TO 60
 220 IF A(X)=D AND A(Y)=0 AND A(
Z)=D THEN LET A(Y)=1: GO TO 60
 230 IF A(X)=0 AND A(Y)=D AND A(
Z)=D THEN LET A(X)=1: GO TO 60
 240 IF B=1 THEN LET X=X+3: LET
Y=Y+3: LET Z=Z+3
 250 IF B=2 THEN LET X=X+1: LET
Y=Y+1: LET Z=Z+1
 260 IF C<3 THEN LET C=C+1: GO T
O 210
 270 IF B<4 THEN LET B=B+1: GO T
O 170
 280 IF D<2 THEN LET D=D+1: GO T
O 150
 290 REM MOVIMIENTO AL AZAR
 300 LET B=1
 310 LET C=INT (RND*9)+1
 320 IF A(C)=0 THEN LET A(C)=1:
GO TO 60
 330 LET B=B+1
 340 IF B<21 THEN GO TO 310
 350 LET B=0
 360 LET B=B+1
 370 IF A(B)=0 THEN LET A(B)=1:
GO TO 60
 380 IF B<9 THEN GO TO 360
 390 GO SUB 700
 400 PRINT '''' FLASH 1;TAB 5;"H
emos empatado!"
 410 GO TO 590
 430 REM MOVIMIENTO GANADOR
 440 FOR B=1 TO 4
 450 IF B=1 THEN LET X=1: LET Y=
2: LET Z=3
 460 IF B=2 THEN LET X=1: LET Y=
4: LET Z=7
 470 IF B=3 THEN LET X=1: LET Y=
5: LET Z=9
 480 IF B=4 THEN LET X=3: LET Z=
7
 490 FOR C=1 TO 3
 500 IF A(X)=A(Y) THEN IF A(Y)=A
(Z) THEN IF A(X)<>0 THEN GO TO 5
60
 510 IF B=1 THEN LET X=X+3: LET
Y=Y+3: LET Z=Z+3
```

```
 520 IF B=2 THEN LET X=X+1: LET
Y=Y+1: LET Z=Z+1
 530 NEXT C
 540 NEXT B
 550 RETURN
 560 BEEP 1,1: BEEP 2,2
 570 IF A(X)=1 THEN PRINT '''' F
LASH 1;TAB 5;"Le he ganado!
"
 580 IF A(X)=2 THEN PRINT ''' FL
ASH 1;TAB 5;"Me ha ganado!
"
 590 FOR X=1 TO 25: BEEP .05,X:
NEXT X
 600 FOR X=50 TO 25 STEP -0.5: B
EEP .05,X: BORDER RND*7: NEXT X
 610 RUN
 620 REM MOVIMIENTO DEL JUGADOR
 630 PRINT AT 1,1; FLASH 1;"Su m
ovimiento..."
 640 LET A$=INKEY$
 650 IF A$<"1" OR A$>"9" THEN GO
 TO 640
 660 LET B=VAL (A$)
 670 IF A(B)<>0 THEN GO TO 640
 680 LET A(B)=2
 690 RETURN
 700 REM IMPRESION
 710 PRINT AT 1,1;"
                  "
 720 PRINT AT 6,5;"1  2  3    ";
 725 LET BANDERA=0
 730 FOR B=1 TO 9
 735 IF A(B)=0 THEN LET FLAG=1
 740 IF A(B)=0 THEN PRINT " _ ";
: BEEP .05,10
 750 IF A(B)=1 THEN PRINT " O ";
: BEEP .09,50
 760 IF A(B)=2 THEN PRINT " X ";
: BEEP .05,-3
 770 IF B=3 THEN PRINT AT 8,5;"4
  5  6
                  ";
 780 IF B=6 THEN PRINT AT 10,5;"
7  8  9
                  ";
 790 NEXT B
 800 IF FLAG=0 THEN GO TO 400
 810 RETURN
```

Ajedrez

En el año 1978, cuando los programas de ordenador para jugar al ajedrez eran una rareza, los lectores del semanario rumano *Magazinul* participaron en una partida de ajedrez colectiva contra un ordenador llamado Felix C-256. El programa fue realizado por el matemático Vioreal Darie, del Instituto de Técnicas de Computación de Bucarest, y su desarrollo requirió dos años de trabajos. Felix hacía un movimiento y se invitaba a los lectores a escribir el movimiento de réplica. La mejor respuesta se entraba al ordenador y su respuesta aparecía en la publicación de la semana siguiente.

A pesar de las 6ØØ horas invertidas en su realización, Felix no jugó particularmente bien y finalmente fue derrotado por los lectores de *Magazinul.* Un experto en ajedrez comentó que algunos de los movimientos de Felix eran muy buenos, mientras que otros eran errores de principiante.

El programa propuesto puede ser objeto de las mismas críticas que Felix. No juega un ajedrez de buena calidad y lo encontrará fácil de ganar, pero ha sido incluido aquí por varias razones.

El ajedrez ha sido siempre una tentación para los programadores. Incluso en los inocentes días de los primeros microprocesadores fue un juego usado para demostrar la inteligencia de tales dispositivos. Para que pueda hacer frente al enorme número de permutaciones que ocurren en el ajedrez, el programa debe ser mucho más largo que el utilizado para jugar a damas o a puntos y cruces. Existen $1Ø^{120}$ juegos posibles de 4Ø movimientos, un número parecido al número de átomos del universo. Programar un ordenador para hacer frente a este reto es un trabajo formidable.

Han sido publicados muy pocos programas en BASIC para jugar al ajedrez. El hecho de que se pueda desarrollar uno que juegue a una velocidad razonable, es, por supuesto, interesante, aunque no le pueda enseñar a usted a jugar mejor al ajedrez. Este programa responde muy rápidamente en la mayor parte de las ocasiones. Localiza, almacena y clasifica las posiciones de sus piezas en menos de cuatro segundos y generalmente mueve al cabo de seis a quince segundos después. Cuando las posiciones en el tablero se hacen más complejas, la velocidad de respuesta baja, pero el Spectrum raramente tardará más de cuarenta segundos en dar su respuesta.

Después del listado se ha incluido una modificación que, en caso de introducirse, permite al ordenador jugar contra sí mismo. Observando al ordenador tirando con las blancas y luego con las negras, puede detectar los puntos débiles del programa y, en un momento de especial inspiración, puede intentar mejorarlos. El programa se ha escrito deliberadamente de forma modular para facilitar el acceso a las

secciones que gestionan las distintas piezas, por tanto usted puede modificar el programa de forma gradual.

El hecho de que el programa Felix pudiera jugar ajedrez fue considerado una novedad en 1978. Esto puede considerarse sorprendente si se considera que el primer artículo serio sobre las posibilidades de un ordenador para jugar este juego fue «Programando un ordenador para jugar al ajedrez» y fue editado por Claude Shannon, en 1949 (un personaje importante en el desarrollo de los ordenadores). El primer programa no salió a la luz hasta después de una década, cuando Alex Bernstein y tres colegas más de IBM consiguieron hacer funcionar un programa para jugar al ajedrez en un IBM 7Ø4. A pesar de este retrasado comienzo, actualmente existen multitud de programas e incluso varias versiones del juego para Spectrum.

A pesar de todo, hacer un programa para jugar al ajedrez sigue siendo un gran reto para un programador. Una vez haya entrado este programa y lo haya trabajado un poco, se formará una mejor idea de cómo realizar un programa de este tipo. Desde este punto podrá apreciar mejor cómo funcionan las máquinas dedicadas al ajedrez y será capaz de intentar la construcción de un nuevo programa. Para construir este juego se han utilizado varias ideas básicas útiles para otros juegos de tablero.

La primera decisión que se ha tomado es apuntar a un objetivo realista. Se ha partido de una serie de esbozos de programas de reconocimiento de fichas y de toma de decisiones. Para unir estas secciones se consideró el uso de saltos dobles, ya que los saltos triples eran un lujo para un programa que no jugaría particularmente bien. Se intentó la utilización de una simple rutina que suministraba *n* saltos, pero no resultó un éxito. Claude Shannon, el personaje que escribió un artículo sobre el juego del ajedrez en ordenadores en 1949, anunció que existían $1Ø^{12Ø}$ secuencias diferentes de movimientos en una partida de 4Ø jugadas. Shannon añadió que un ordenador rápido tardaría $1Ø^{9Ø}$ años en examinar todos estos movimientos antes de mover el primer peón dos cuadros.

Se observó que $1Ø^{9Ø}$ años era un tiempo demasiado largo y que cuarenta veces $1Ø^{9Ø}$ años podría bastar para terminar con la paciencia de cualquier jugador. Por lo tanto, una investigación exhaustiva de las posibilidades de cada jugada es obviamente imposible, incluso si se escribiera el programa de forma que seleccionara las jugadas clave.

A pesar de crear un programa cuyos únicos recursos sean la previsión de las consecuencias de sus propios movimientos, el número de combinaciones a examinar y su realización con un interpretador BASIC lo deberían hacer muy lento. No obstante, una vez realizado el juego, la velocidad de respuesta fue sorprendente, incluso para sus realizadores. Debe tenerse presente que un programa que tenga subrutinas frecuentemente usadas situadas hacia el final del listado, se

ejecuta más lentamente que uno que las tenga al principio. Este fenómeno se produce porque cuando se llama a una subrutina el programa la busca desde el principio del listado línea a línea hasta que la encuentra. Si la subrutina está al principio, el ordenador debe analizar menos líneas.

El programa se diseñó cuidadosamente, escribiendo la primera versión sobre papel y depurándola antes de introducirla en el Spectrum. Esto permitió imaginar muchas veces la ejecución del programa, barajando los distintos componentes y optimizando la estructura del juego para hacerlo lo más lógico posible. Este procedimiento es el que ha permitido obtener una alta velocidad de respuesta.

Por lo tanto, las primeras dos ideas que han inspirado este programa han sido la necesidad de apuntar a unos objetivos moderados y la importancia de construir una estructura de programación que mantuviera el tiempo de respuesta tan corto como fuera posible.

La próxima idea importante es la disposición del tablero. El primer prototipo se realizó utilizando el sistema de numeración propuesto por A. L. Samuels en un artículo aparecido en el *Scientific American* (véase Strachey, C., «Systems Analysis and Programming», en *Readings from Scientific American,* W. H. Freeman and Co., San Francisco, 1971).

A pesar de que el sistema funcionaba correctamente, requería mucho tiempo para convertir los movimientos entrados por el jugador en números que pudiera usar el ordenador. Se desechó el método y se consideró el sugerido en 1981 por Graham Charlton (coautor de *The Turing Criterion-Machine Intelligent Programs for the 16K ZX81*, Interface, Londres, 1982), que, además, resulta ser el que se ha usado en el anterior juego de DAMAS. El tablero entero de copia en una matriz y los movimientos se realizan cambiando de posición los elementos de la matriz, reemplazando un elemento de la matriz por el carácter de un cuadro vacío cuando una pieza abandona una posición y colocando el código (CODE) de la pieza en el elemento de la matriz que representa la posición a la que se mueve. Para imprimir el tablero en la pantalla el ordenador de imprimir consecutivamente el carácter (CHR$) de cada elemento de la matriz. El sistema de numeración de Charlton facilita también la conversión entre los números entrados por el jugador para indicar los movimientos que desea efectuar y el número de posición de las fichas en la matriz.

Con el fin de que el jugador pueda utilizar la misma notación algebraica estandarizada en el argot ajedrecístico, se ha organizado el tablero refiriéndose a las columnas con letras y a las filas con números. El siguiente diagrama representa el tablero rodeado por las letras y números de la notación algebraica y, dentro del tablero, los elementos de la matriz asociada con su numeración:

	A	B	C	D	E	F	G	H
8	18	28	38	48	58	68	78	88
7	17	27	37	47	57	67	77	87
6	16	26	36	46	56	66	76	86
5	15	25	35	45	55	65	75	85
4	14	24	34	44	54	64	74	84
3	13	23	33	43	53	63	73	83
2	12	22	32	42	52	62	72	82
1	11	21	31	41	51	61	71	81

La gran ventaja del sistema de numeración Charlton sobre otros sistemas reside en el hecho de que cualquier movimiento puede ser descrito como suma o resta desde el cuadro inicial. Esto puede verse fácilmente mediante un ejemplo. Imagínese un caballo situado en el cuadro 45. Un caballo en ajedrez tiene un movimiento en forma de L, dos cuadros y luego uno o un cuadro y luego dos. Un movimiento posible para el caballo desde la posición en la que está (45) es ir al cuadro 66, esto es sumando 21 a la posición inicial. Ahora imagínese el caballo situado en el cuadro 11. Sumando 21 quedaría situado en 32, obsérvese que se trata igualmente de un movimiento legal. Del mismo modo desde el cuadro 52, sumando 21 se llega al 73, destino también legal. Los ocho movimientos de un caballo (suponiendo que ninguno de ellos lo coloque fuera del tablero) se obtienen sumando los siguientes números a la posición de inicio:

19, −19, 21, −21, 8, −8, 12, −12

De forma parecida, los movimientos de un peón (a parte del primer movimiento en que se le permite saltar dos cuadros) se determinan simplemente sumando una unidad a la posición actual (es decir, un peón puede moverse legalmente del cuadro 45 al 46 o del 23 al 24). Un peón consigue una captura mediante un movimiento diagonal, por tanto puede llegar a los cuadros +11 o −9. Sabiendo todo esto ya es posible programar el ordenador para investigar todos los movimientos legales desde un cuadro dado.

Esto es lo que se ha realizado en la rutina de inicialización del programa. La citada rutina se encuentra a partir de la línea 264Ø. Primeramente (en la línea 265Ø) se DIMensionan una serie de vectores que se usarán para distintos cometidos. Entre ellos se hallan los que contienen los movimientos potenciales de cada pieza. El vector N (los movimientos del caballo) se llena con los datos de la línea 287Ø. Si se observa esta línea, se verán los mismos ocho números que han aparecido anteriormente. Un modo de decidir cómo mover un caballo (es

el método usado por el programa, con algunas limitaciones, es generar un número aleatorio entre 1 y 8, y examinar el elemento correspondiente del vector N para ver si la posición que actualmente ocupa el caballo más el número contenido en el elemento es un cuadro vacío o una pieza del oponente. Si se determina que es una pieza del oponente, se efectúan unas cuantas pruebas para verificar que el caballo no puede ser capturado y, si es así, se efectúa el movimiento. Esta técnica constituye el corazón del programa AJEDREZ. Conviene tomarse cierto tiempo y releer lo explicado hasta aquí para entenderlo perfectamente. Con la información aportada usted podría escribir un programa para jugar al ajedrez desde cero, suponiendo que dispusiera del tiempo y de la voluntad para hacerlo.

Un programa que sólo tomara decisiones tirando los dados, como sugiere la explicación dada, caería presa de cualquier principiante rápidamente. Por lo tanto, el sistema de generación de movimientos debe ser modificado en una cierta extensión, tal como se ha realizado en este programa. Los vectores de las piezas están situados en el programa como sigue:

Caballo	de 286Ø a 287Ø
Torre	de 288Ø a 292Ø
Alfil	de 293Ø a 297Ø
Reina	299Ø llena el vector de la reina (Q) con los elementos de la torre y del alfil, porque los movimientos de la reina no son más que combinaciones de ambos.
Rey	de 3ØØØ a 3Ø1Ø

El próximo vector a rellenar, el vector S (línea 3Ø2Ø a 31ØØ), contiene el secreto de la velocidad del programa. Muchos juegos de tablero, como DAMAS, examinan los cuadros uno a uno desde el cuadro superior izquierdo al inferior derecho, buscando piezas, capturadas o movimientos. En este caso se ha tomado otra decisión para minimizar el tiempo de respuesta del ordenador. Los cuadros se examinan en un orden que estadísticamente mejora este tiempo de forma muy significativa. Este orden está contenido en las líneas de datos 3Ø3Ø. A continuación se muestran los números de posiciones a muestrear (pueden ser comparados con los del cuadro de la página 85):

```
46  56  36  66  47  57  45  55
37  67  35  65  28  78  27  77
44  54  26  76  38  68  17  87
18  68  34  64  25  75  16  86
48  24  74  15  85  14  84  43
53  33  63  23  73  52  42  62
32  83  13  72  82  12  82  41
51  31  61  21  71  11  81  58
```

El programa siempre examina primero el cuadro 46, después el 56, siguiendo la lista. Es interesante estudiarla unos momentos, ya que quizás usted mismo pueda variarla, intentando mejorar el programa.

Recapitulando lo expuesto hasta aquí, se podría resumir diciendo que a pesar de que AJEDREZ no es un programa que juegue al ajedrez especialmente bien, la constante fascinación que este tipo de programas produce a los programadores y el enorme aprendizaje que se obtiene de ellos justifican plenamente su inclusión en este libro.

El listado del programa es seguido por una variación de las primeras diez líneas, que permiten que el ordenador juegue contra sí mismo. Esta es una buena forma de depurar el programa, ya que se le puede dejar funcionando un día entero jugando contra sí mismo y viendo si el programa aborta en algún punto.

El sistema de numeración del teclado se ha ilustrado con algunos ejemplos y se ha explicado que permite expresar todos los movimientos como adiciones o substracciones desde los puntos de origen, contenidos en la matriz que representa el teclado. Los movimientos que puede hacer cada figura se almacenan en un vector. Tal como Q para los movimientos puede efectuar la reina y B para los que puede hacer el alfil. Se almacena también otro vector llamado S que contiene el orden en el que se examinan los cuadros de la pantalla.

A continuación se representa la organización del programa:

1Ø-3Ø	Información inicial. Asegura la colocación de CAP LOCK.
4Ø-6Ø	Estas tres líneas reimprimen el tablero después de un movimiento del ordenador, esperan el movimiento del jugador y reimprimen el tablero.
7Ø-235Ø	Este es el centro del programa y contiene toda su parte pensante. Se analizará con detalle posteriormente.
236Ø-246Ø	Reimprimen el tablero después de cada movimiento y (en las líneas 241Ø y 242Ø) promociona un peón a reina si ha llegado a la última fila.
247Ø-256Ø	Esta es la rutina del último recurso. Si el rey detecta que tiene problemas, intenta escapar, y si no se puede realizar ningún movimiento legal, la línea 256Ø dirige el programa a 2Ø6Ø, donde el ordenador le concede la victoria.
257Ø-263Ø	Esta sección acepta su movimiento, llama a una subrutina de comentarios y permite al jugador prescribir al ordenador una serie de cosas. El programa pide una entrada en forma de letra y número (tal como A5) entrados juntos, que le indican el cuadro desde el que pretende moverse, e inmediatamente una segunda entrada idéntica para ordenar el cuadro de destino. Se verifica la sintaxis de la entrada, pero no si es un movimiento

legal. Hecho esto se le presentan unas alternativas entre las que debe escoger:

C	indica al ordenador que lo pone en jaque.
P	ordena una copia sobre papel.
X	intercambio de fichas.
S	para terminar la partida.
ENTER	para seguir con el juego.

Una vez hecha esta entrada, las líneas 2625 y 263Ø construyen su movimiento.

Con esta sección se terminan las partes activas del programa, excepto las rutinas que comentan las capturas y la que se utiliza para intercambiar fichas. El resto del programa es la inicialización y se organiza como sigue:

264Ø Activa el generador de números aleatorios, iguala la variable MM a cero y lleva a cabo la inicialización de la variable A$ (que acepta la primera parte del movimiento del jugador y se utiliza para comprobarla) a «». MM es la variable más importante del programa. Cuando el programa encuentra un posible movimiento, MM cambia de cero a uno. A todo lo largo del programa la variable MM se verifica varias veces y, cuando cambia de estado, el ordenador cesa la búsqueda de posibles movimientos. Si después de repetir el proceso de búsqueda ocho veces no se obtiene ningún resultado, MM cambia también de cero a uno.

265Ø Aquí se dimensionan las matrices y vectores.

A	contiene el tablero. Es la matriz principal.
R	movimientos posibles para la torre.
B	movimientos posibles para el alfil.
N	movimientos posibles para el caballo.
Q	movimientos posibles para la reina.
K	movimientos posibles para el rey.
Z	se usa para intercambiar las fichas.
S	es la secuencia en la que se examinan los cuadros.
T	contiene la posición de cada pieza. Cambia a cada movimiento.

266Ø-267Ø Estas dos líneas contienen las variables usadas por las piezas. En el tablero representado en la pantalla, las figuras están representadas por letras y el ordenador dispone de las piezas impresas en mayúscula situadas en la parte superior de la pantalla. Usted dispone de las fichas representadas con letras minúsculas. A continuación puede observar una muestra del tablero al principio de la partida:

Como puede observar, el tablero está construido con puntos en lugar de cuadros y no se han incluido gráficos definidos por usuario. El objetivo del programa es jugar al ajedrez antes que resultar bonito, pero si desea hacerlo más vistoso, añada los gráficos usted mismo. Para indicar un cuadro debe entrarse una letra para indicar la columna, seguida de un número para indicar la fila. Por ejemplo, su reina empieza la partida en D1 y el caballo del ordenador lo hace en G8.

Estas son las variables de las piezas:

Usted - blancas: P-112; R-114; N-110; B-98; Q-113; K-107

Ordenador - negras: PB-80; RB-82; NB-78; BB-66; QB-81; KB-75

268Ø-31ØØ Los vectores para los movimientos posibles de las piezas. Ya se han descrito.

311Ø Hace el papel y el borde azules y la tinta blanca. Puede cambiar estos colores en cualquier partida.

312Ø-321Ø Esta rutina le permite intercambiar sus fichas con las del ordenador en cualquier momento después de jugar. El sistema de intercambio se realiza con la siguiente secuencia: el ordenador actúa como si se colocara un espejo en el centro del tablero y las fichas se reflejaran en él, es decir, una pieza situada en A2 será colocada en A7. Aunque se varían las posiciones de las piezas, el ordenador sigue jugando con las mayúsculas y el jugador con las minúsculas, resituadas todas en la posición anterior de sus contrarios. Cuando se selecciona la opción de «juego automático», se ejecuta el intercambio de fichas a cada jugada y se refresca la imagen de la pantalla cada dos jugadas. Así puede seguir el juego como si fuera uno de los participantes.

331Ø-338Ø Si su movimiento tiene su fin en un cuadro ocupado por una figura del ordenador, éste lo detecta en la línea 2625 y usa esta rutina para comentar su captura. El borde cambiará de color intermitentemente tres veces, se mostrará el mensaje «¡Me cazó!» y se oirán unas notas musicales.

Finalmente y antes de entrar en el listado del programa, se profundizará en el funcionamiento del programa. Con vistas a la mejora del programa usted debe conocer las distintas secciones para poder abordar la adecuada a cada modificación.

En la línea 7Ø la variable MM contiene el movimiento de la máquina. Se iguala a cero después de cada movimiento. Si quiere mover primero, puede borrar la instrucción GOTO 6Ø al final de la línea 3Ø. La variable UK cuenta el número de ciclos que el programa tarda en encontrar un movimiento. Se usa para asegurar que el rey no se moverá sin tener una buena razón para ello y para evitar que el programa le conceda la victoria fácilmente. Las próximas líneas (8Ø a 1ØØ) ejecutan los deseos del jugador expresados en su movimiento.

La línea 11Ø dimensiona el vector T. Se dimensiona a cada ciclo para inicializarlo e igualar a cero cada uno de sus componentes, quedando así listo para almacenar las posiciones de cada pieza. U es la variable que cuenta el número de piezas que el ordenador tiene en el tablero. «Espere un momento, por favor» aparece en la parte superior de la pantalla mientras el ordenador busca su jugada. La rutina de las líneas 12Ø y 13Ø apunta a cada uno de los cuadros del tablero en el orden dictado por los elementos del vector S, almacenando el contenido de cada cuadro en el vector T y contándolos. A la variable KM se la podría llamar el marcador del rey, ya que es la que se encarga de seguir permanentemente la pista a la figura más valiosa del tablero. KM se iguala a la posición del rey al final de la línea 12Ø. La línea 13Ø completa el bucle. Como se ha mencionado, U cuenta el número de piezas existentes en el tablero. Si U es menor que tres, el ordenador le concede la victoria.

La acción va ahora a la línea 58Ø donde la variable Z se iguala al marcador del rey (KM). La rutina siguiente, desde esta línea hasta la 81Ø, comprueba si el rey está en peligro. Esta comprobación se hace aunque usted no haya anunciado jaque al hacer su movimiento. Si se detecta algún peligro, el ordenador va a la línea 171Ø para intentar evitarlo.

Volviendo al principio del programa, a la línea 15Ø, el programa busca a su rey y lo coloca al final de sus piezas, de forma que éste no se mueva si existe otra alternativa. En este punto la máquina sabe si el rey está en peligro, ya que se ha examinado esta posibilidad en la línea 58Ø, si no es así, no conviene mover el rey y exponerlo a una posición menos segura.

La línea 17Ø escoge la pieza de la secuencia con la que se empieza la búsqueda, escogiéndose una de las tres primeras. La pieza escogida es necesariamente la primera si el jugador ha señalado jaque al final de la línea 17Ø. Si el número escogido es menor que el número de piezas del tablero, se incrementa el valor de Q para determinar qué elemento del vector T será el que empezará la búsqueda.

La línea 19Ø iguala la variable Z (que se usa para almacenar el cuadro de partida) al elemento del vector T escogido y el programa salta tres líneas a una subrutina que empieza en 23Ø. Se produce un BEEP en la línea 23Ø que informa que el ordenador está trabajando. Este BEEP suena de cuando en cuando para demostrar que el programa sigue en funcionamiento, aunque cada vez tiene más dificultades para escoger un movimiento.

Las cuatro líneas siguientes envían el programa a distintas subrutinas dependiendo de la pieza que se considere. De los destinos de estas líneas puede descubrir las secciones del programa que intentan capturar:

Reina — 82Ø
Torre — 13ØØ
Caballo — 154Ø
Peón — 2Ø7Ø

La línea 28Ø retorna a la 2ØØ, donde se examina MM. Si es igual a uno, el programa va a la línea 231Ø para hacer el movimiento y después vuelve a la línea 4Ø que llama a la subrutina de reimprimir el tablero antes de permitirle entrar su movimiento. Si MM es igual a cero, el programa sigue a la próxima línea y, si Q es menor que U, envía la acción a la línea 18Ø donde se incrementa Q y prosigue la búsqueda. Si Q no es menor que U, significa que se ha comprobado la posibilidad de captura de todas las piezas sin éxito.

Desde este punto el programa salta a la línea 218Ø (a pesar de intentar un flujo correcto del programa, el tiempo de respuesta ha obligado a estos saltos), donde Q se iguala a un número aleatorio entre cero y cuatro (mediante el doble uso de RND para intentar sesgar el resultado hacia números pequeños). Se comprueba que este número no exceda del de piezas disponibles. Usted puede cambiar el cinco de esta expresión por un número mayor para reducir la predicción del juego. Se puede colocar cualquier número hasta el nueve incluido. En la línea 219Ø se suma uno y se iguala a Z la pieza seleccionada. Los destinos de las subrutinas delatan las secciones donde se puede encontrar el control de las distintas piezas:

Peón — 214Ø
Caballo — 163Ø
Alfil — 142Ø
Torre — 118Ø
Reina — 94Ø
Rey — 171Ø (solamente si el rey no está en jaque y el número aleatorio es inferior a .Ø7, para evitar movimientos inútiles del rey).

Una vez que el ordenador ha ejecutado la subrutina escogida, comprueba si la variable MM aún es cero. Si es así, y no han sido examinadas todas las piezas, el programa va a 218Ø para escoger una nueva pieza. Si MM se ha convertido en uno, el ordenador va a la subrutina de la línea 231Ø para efectuar el movimiento y luego vuelve a la línea 4Ø para reimprimir el tablero. La próxima línea examina el valor de la variable UK y, si es mayor que ocho, resulta que el ordenador ha ejecutado la rutina de búsqueda más de ocho veces, por lo tanto, va a la línea 2Ø6Ø para concederle el juego.

La próxima secuencia efectúa el movimiento del ordenador. La línea 231Ø rechaza nueve veces de cada diez el movimiento del rey, si éste no está en jaque, igualando MM a cero y volviendo a 218Ø para examinar el posible movimiento de otra pieza. Se activa la línea 2312 en caso de movimiento de un peón. Comprueba si el programa trata de moverlo hacia atrás y, siendo así, bloquea el movimiento. Si el rey blanco está en el cuadro al cual se tiene la intención de mover, se imprime «¡Jaque!» en la pantalla y el ordenador retorna a la línea 19Ø para intentar otra captura.

Si el movimiento escogido logra pasar todas las pruebas (ésta es una zona del programa fácilmente modificable, si usted quiere evitar ciertos movimientos), la línea 2315 efectúa el movimiento y las líneas 232Ø a 233Ø le informan (usando una complicada línea de programa para convertir el número del vector a las coordenadas del tablero) del movimiento realizado. La línea 234Ø devuelve el programa al bucle en el punto de impresión del tablero.

A pesar de la rutina que determina si el rey está en jaque, el programa no siempre puede despejar esta situación. Puede considerar este defecto como un deseo del ordenador en concederle la victoria (o, si se siente generoso, permita al programa estar en jaque una jugada más). Igualmente, si el programa se pone a sí mismo en jaque (ocurre muy raramente), tómelo como una concesión de victoria.

```
  10 REM AJEDREZ
  20 PRINT "POR FAVOR COLOQUE  "
; FLASH 1;"MAYUSCULAS"; FLASH 0,
"DESPUES APRIETE ENTER": INPUT A
$: CLS : PRINT '''"Gracias, por f
avor espere un     momento"
  30 GO SUB 2540: GO TO 60
  40 GO SUB 2350
  50 GO SUB 2570
  60 GO SUB 2350
  70 LET MM=0: LET UK=0
  80 IF A$="S" THEN STOP
  90 IF A$="X" THEN PRINT AT 0,0
; FLASH 1;"Intercambio de fichas
": GO SUB 3120: PRINT AT 0,0;"
                ": LET A$=""
```

```
 100 IF A$="P" THEN COPY
 110 DIM T(16): LET U=0: PRINT A
T 0,0; INK RND*5+2; PAPER 9;" Po
r favor espere un momento "
 120 FOR Q=1 TO 64: IF A(S(Q))>=
BB AND A(S(Q))<=RB THEN LET U=U+
1: LET T(U)=S(Q): IF A(S(Q))=KB
THEN LET KM=S(Q)
 130 NEXT Q: IF U<3 THEN GO TO 2
060
 140 GO TO 580
 150 FOR Q=1 TO U: IF A(T(Q))=KB
 THEN LET T(Q)=T(U): LET T(U)=KM
 160 NEXT Q
 170 LET Q=INT (RND*3)*((A$="C")
+1)
 180 IF Q<U THEN LET Q=Q+1
 190 LET Z=T(Q): GO SUB 230
 200 IF MM=1 THEN GO SUB 2310: G
O TO 40
 210 IF Q<U THEN GO TO 180
 220 GO TO 2180
 230 BEEP .008,RND*30+20: IF A(Z
)=0B THEN GO SUB 820
 240 IF A(Z)=RB THEN GO SUB 1060
 250 IF A(Z)=BB THEN GO SUB 1300
 260 IF A(Z)=NB THEN GO SUB 1540
 270 IF A(Z)=PB THEN GO SUB 2070
 280 GO TO 200
 290 IF A(X)=107 THEN PRINT AT 0
,0; FLASH 1;"       Jaque!       ";
: LET Q=Q+1: LET MM=0: GO TO 190
 300 IF X+9>88 THEN GO TO 320
 310 IF A(X+9)<83 AND A(X+9)>65
AND RND<.96 THEN RETURN
 320 IF X-11<11 THEN GO TO 340
 330 IF A(X-11)<83 AND A(X-11)>6
5 AND RND<.96 THEN RETURN
 340 LET AD=0
 350 LET AY=1
 360 LET AX=X+Q(AY+AD)
 370 IF AX<11 OR AX>88 THEN GO T
O 400
 380 LET AP=A(AX)
 390 IF AP=Q OR AP=R AND RND>.8
OR AP=B AND RND>.5 THEN RETURN
 400 LET AY=AY+1
 410 IF AY<8 THEN GO TO 360
 420 LET AD=AD+7
 430 IF AD<56 THEN GO TO 350
 440 LET AY=1
 450 LET AX=X+N(AY)
 460 IF AX<11 OR AX>88 THEN GO T
O 480
 470 IF A(AX)=N THEN RETURN
 480 LET AY=AY+1
 490 IF AY<9 THEN GO TO 450
 500 LET AY=1
 510 LET AX=X+K(AY)
```

```
 520 IF AX<11 OR AX>88 THEN GO T
O 540
 530 IF (A(AX)=K OR A(AX)=P) AND
 RND>.1 THEN RETURN
 540 LET AY=AY+1
 550 IF AY<9 THEN GO TO 510
 560 LET MM=1
 570 RETURN
 580 LET Z=KM
 590 LET Y=0
 600 LET Y=Y+1
 610 LET X=Z+N(Y)
 620 IF X<11 OR X>88 THEN GO TO
640
 630 IF A(X)=N THEN GO TO 1710
 640 IF Y<8 THEN GO TO 600
 650 LET D=0
 660 LET Y=1
 670 LET X=Z+Q(Y+D)
 680 IF X<11 OR X>88 THEN GO TO
730
 690 IF A(X)=B OR A(X)=Q OR A(X)
=R THEN GO TO 1710
 700 IF A(X)<>E THEN GO TO 730
 710 LET Y=Y+1
 720 IF Y<8 THEN GO TO 670
 730 LET D=D+7
 740 IF D<49 THEN GO TO 670
 750 LET X=Z+11
 760 IF X>88 THEN GO TO 780
 770 IF A(X)=P THEN GO TO 1710
 780 LET X=Z-11
 790 IF X<11 THEN GO TO 150
 800 IF A(X)=P THEN GO TO 1710
 810 GO TO 150
 820 LET D=0
 830 LET Y=1
 840 LET X=Z+Q(Y+D)
 850 IF X<11 OR X>88 THEN GO TO
910
 860 IF A(X)=42 OR A(X)>=BB AND
A(X)<=RB THEN GO TO 910
 870 IF A(X)>=B AND A(X)<=R THEN
 GO SUB 290: IF MM<>1 THEN GO TO
 910
 880 IF MM=1 THEN RETURN
 890 LET Y=Y+1
 900 IF Y<8 THEN GO TO 840
 910 LET D=D+7
 920 IF D<49 THEN GO TO 830
 930 RETURN
 940 LET D=0
 950 LET Y=1
 960 LET X=Z+Q(Y+D)
 970 IF X<11 OR X>88 THEN GO TO
1030
 980 IF A(X)<>E THEN GO TO 1030
 990 IF RND>.5 THEN GO SUB 290:
IF MM=0 THEN GO TO 1030
```

```
1000 IF MM=1 THEN RETURN
1010 LET Y=Y+1
1020 IF Y<8 THEN GO TO 960
1030 LET D=D+7
1040 IF D<49 THEN GO TO 950
1050 RETURN
1060 LET D=0
1070 LET Y=1
1080 LET X=Z+R(Y+D)
1090 IF X<11 OR X>88 THEN GO TO
1150
1100 IF A(X)=42 OR A(X)>=BB AND
A(X)<=RB THEN GO TO 1150
1110 IF A(X)>=B AND A(X)<=R THEN
 GO SUB 290: IF MM=0 THEN GO TO
1150
1120 IF MM=1 THEN RETURN
1130 LET Y=Y+1
1140 IF Y<8 THEN GO TO 1080
1150 LET D=D+7
1160 IF D<28 THEN GO TO 1070
1170 RETURN
1180 LET D=0
1190 LET Y=1
1200 LET X=Z+R(Y+D)
1210 IF X<11 OR X>88 THEN GO TO
1270
1220 IF A(X)<>E THEN GO TO 1270
1230 IF RND<.1 THEN GO SUB 290
1240 IF MM=1 THEN RETURN
1250 LET Y=Y+1
1260 IF Y<8 THEN GO TO 1200
1270 LET D=D+7
1280 IF D<28 THEN GO TO 1190
1290 RETURN
1300 LET D=0
1310 LET Y=1
1320 LET X=Z+B(Y+D)
1330 IF X<11 OR X>88 THEN GO TO
1390
1340 IF A(X)=42 OR A(X)>=BB AND
A(X)<=RB THEN GO TO 1390
1350 IF A(X)>=B AND A(X)<=R THEN
 GO SUB 290: IF MM<>1 THEN GO TO
 1390
1360 IF MM=1 THEN RETURN
1370 LET Y=Y+1
1380 IF Y<8 THEN GO TO 1320
1390 LET D=D+7
1400 IF D<28 THEN GO TO 1310
1410 RETURN
1420 LET D=0
1430 LET Y=1
1440 LET X=Z+B(Y+D)
1450 IF X<11 OR X>88 THEN GO TO
1510
1460 IF A(X)<>E THEN GO TO 1510
1470 IF RND>.05 THEN GO SUB 290:
 IF MM<>1 THEN GO TO 1510
```

```
1480 IF MM=1 THEN RETURN
1490 LET Y=Y+1
1500 IF Y<8 THEN GO TO 1440
1510 LET D=D+7
1520 IF D<28 THEN GO TO 1430
1530 RETURN
1540 LET Y=1
1550 LET X=Z+N(Y)
1560 IF X<11 OR X>88 THEN GO TO
1600
1570 IF A(X)=42 THEN GO TO 1600
1580 IF A(X)>=B AND A(X)<=R THEN
 GO SUB 290
1590 IF MM=1 THEN RETURN
1600 LET Y=Y+1
1610 IF Y<9 THEN GO TO 1550
1620 RETURN
1630 LET Y=0
1640 LET X=Z+N(INT (RND*8+1))
1650 IF X<11 OR X>88 THEN GO TO
1640
1660 IF A(X)=42 THEN GO TO 1640
1670 LET Y=Y+1
1680 IF A(X)=E THEN GO SUB 290
1690 IF MM=1 OR Y>20 THEN RETURN

1700 GO TO 1640
1710 LET YK=1
1720 LET Z=KM
1730 LET X=Z+K(YK): LET X1=X
1740 IF X<11 OR X>88 THEN GO TO
2030
1750 IF A(X)=42 OR A(X)>65 AND A
(X)<83 THEN GO TO 2030
1760 IF A(X)>97 AND A(X)<115 THE
N GO TO 2030
1770 LET Z=X
1780 LET Y=0
1790 LET Y=Y+1
1800 LET X=Z+N(Y)
1810 IF X<11 OR X>88 THEN GO TO
1830
1820 IF A(X)=N THEN GO TO 2030
1830 IF Y<8 THEN GO TO 1790
1840 LET D=0
1850 LET Y=1
1860 LET X=Z+Q(Y+D)
1870 IF X<11 OR X>88 THEN GO TO
1920
1880 IF A(X)=B OR A(X)=Q OR A(X)
=R THEN GO TO 2030
1890 IF A(X)<>E THEN GO TO 1920
1900 LET Y=Y+1
1910 IF Y<8 THEN GO TO 1860
1920 LET D=D+7
1930 IF D<49 THEN GO TO 1860
1940 LET X=Z+11
1950 IF X>88 THEN GO TO 1970
1960 IF A(X)=P THEN GO TO 2030
```

```
1970 LET X=Z-11
1980 IF X<11 THEN GO TO 2000
1990 IF A(X)=P THEN GO TO 2030
2000 LET X=X1: LET Z=KM
2010 LET MM=1
2020 GO SUB 2310: GO TO 40
2030 LET YK=YK+1
2040 LET Z=KM
2050 IF YK<9 THEN GO TO 1720
2060 PRINT AT 0,0; FLASH 1;"  Le
 concedo la victoria!  ": STOP
2070 LET X=Z+9
2080 IF A(X)>=B AND A(X)<=R THEN
 LET MM=1: IF A(X)=P AND RND<.2
THEN LET MM=0
2090 IF MM=1 THEN RETURN
2100 IF Z=12 THEN RETURN
2110 LET X=z-11
2120 IF A(X)>=B AND A(X)<=R THEN
 LET MM=1: IF A(X)=P AND RND<.2
THEN LET MM=0
2130 RETURN
2140 IF Z-10*(INT (Z/10))=7 AND
A(Z-1)=E AND A(Z-2)=E AND (A(Z-1
3)=E OR A(Z-13)=42) AND (A(Z+7)=
E OR A(Z+7)=42) THEN LET X=Z-2:
LET MM=1: RETURN
2150 IF A(Z-1)=E AND A(Z-12)<98
AND A(Z+8)<98 THEN LET X=Z-1: LE
T MM=1: RETURN
2160 IF RND<.05 AND A(Z-1)=E THE
N LET X=Z-1: LET MM=1
2170 RETURN
2180 LET Q=INT (RND*RND*5): IF Q
>U THEN GO TO 2180
2190 IF Q<U THEN LET Q=Q+1
2200 LET Z=T(Q)
2210 IF A(Z)=PB THEN GO SUB 2140
2220 IF A(Z)=NB THEN GO SUB 1630
2230 IF A(Z)=BB THEN GO SUB 1420
2240 IF A(Z)=RB THEN GO SUB 1180
2250 IF A(Z)=QB THEN GO SUB 940
2260 IF A(Z)=KB AND A$<>"C" AND
RND<.07 THEN GO SUB 1710
2270 IF MM=0 AND Q<U THEN GO TO
2190
2280 IF MM=1 THEN GO SUB 2310: G
O TO 40
2300 LET UK=UK+1: IF UK>8 THEN G
O TO 2060
2305 GO TO 2130
2310 IF A(Z)=KB AND A$<>"C" AND
RND>.1 THEN BEEP .05,-3: LET MM=
0: GO TO 2180
2312 IF A(Z)=PB AND ((X-10*INT (
X/10)>Z-10*INT (Z/10) OR ABS (X-
Z)>11)) THEN LET MM=0: GO TO 218
0
```

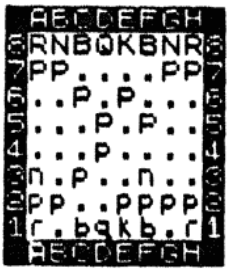
```
2314 IF A(X)=K THEN PRINT AT 0,0
;"       Jaque!              ": LET MM=0
: LET U=U+1: GO TO 190
2315 LET A(X)=A(Z): LET A(Z)=E
2320 PRINT AT 0,0;"     Me muevo
desde   ";
2330 LET FZ=INT (Z/10): PRINT CH
R$ (FZ+64);Z-10*FZ;"   a   ";: LET
 FX=INT (X/10): PRINT CHR$ (FX+6
4);X-10*FX
2340 RETURN
2350 IF MM=0 THEN GO TO 2370
2360 BEEP 1,RND*10: BEEP 1,10+RN
D*10: BEEP 1,-RND*10
2370 PRINT AT 0,0;"
                    ";AT 4,0;: GO SU
B 2450
2380 FOR X=8 TO 1 STEP -1
2390 PRINT TAB 10; INVERSE 1;X;
INVERSE 0;
2400 FOR Y=10 TO 80 STEP 10
2410 IF A(Y+1)=PB THEN LET A(Y+1
)=0B
2420 IF A(Y+8)=B THEN LET A(Y+8)
=0
2430 PRINT CHR$ A(X+Y);
2440 NEXT Y: PRINT INVERSE 1;X:
NEXT X: LET MM=0
2450 PRINT TAB 10; INVERSE 1;" A
BCDEFGH "
2460 RETURN
2470 LET Z=KM
2480 LET QK=0
2490 LET M=Z+K(QK)
2500 IF A(M)=42 OR A(M)>65 AND A
(M)<83 OR MM=0 THEN GO TO 2540
2510 LET X=M
2520 LET KM=X
2530 RETURN
2540 IF QK<8 THEN GO TO 2490
2550 IF A$<>"C" THEN RETURN
2560 GO TO 2060
2570 INPUT "DESDE (LETRA NUMERO)
 ? ";A$: BEEP .008,-3
2580 IF LEN A$<>2 THEN GO TO 257
0
2590 INPUT "DESDE ";(A$);" A ? "
;B$: BEEP .008,-10
2595 IF LEN B$<>2 THEN GO TO 259
0
2600 LET X=10*(CODE A$-64)+VAL A
$(2)
2610 LET Y=10*(CODE B$-64)+VAL B
$(2)
```

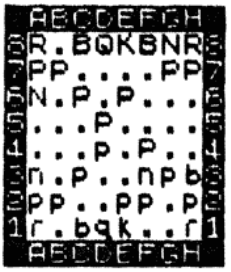
```
2620 INPUT "EntrE C - jaque
                P - imprimir t
ablero          X - cambiar fi
chas            S - parar jueg
o           ENTER - continuar
";A$
2625 IF A(Y)>=75 AND A(Y)<=82 TH
EN GO SUB 3310
2630 BEEP .008,-3: LET A(Y)=A(X)
: LET A(X)=46: RETURN
2640 RANDOMIZE : LET MM=0: LET A
$=""
2650 DIM A(99): DIM R(28): DIM B
(28): DIM N(8): DIM Q(56): DIM K
(8): DIM Z(88): DIM S(64): DIM T
(16)
2660 LET P=112: LET R=114: LET N
=110: LET B=98: LET Q=113: LET K
=107: LET E=46
2670 LET PB=80: LET RB=82: LET N
B=78: LET BB=66: LET QB=81: LET
KB=75
2680 FOR Z=1 TO 99: LET A(Z)=42:
 NEXT Z
2690 FOR Z=1 TO 64: READ X: READ
 Y: LET A(X)=Y: NEXT Z
2700 DATA 18,82,28,78,38,66,48,8
1
2710 DATA 58,75,68,66,78,78,88,8
2
2720 DATA 17,80,27,80,37,80,47,8
0
2730 DATA 57,80,67,80,77,80,87,8
0
2740 DATA 16,46,26,46,36,46,46,4
6
2750 DATA 56,46,66,46,76,46,86,4
6
2760 DATA 15,46,25,46,35,46,45,4
6
2770 DATA 55,46,65,46,75,46,85,4
6
2780 DATA 14,46,24,46,34,46,44,4
6
2790 DATA 54,46,64,46,74,46,84,4
6
2800 DATA 13,46,23,46,33,46,43,4
6
2810 DATA 53,46,63,46,73,46,83,4
6
2820 DATA 12,112,22,112,32,112,4
2,112
2830 DATA 52,112,62,112,72,112,8
2,112
2840 DATA 11,114,21,110,31,98,41
,113
2850 DATA 51,107,61,98,71,110,81
,114
```

```
2860 FOR Z=1 TO 8: READ N(Z): NE
XT Z
2870 DATA 19,-19,21,-21,-8,8,12,
-12
2880 FOR Z=1 TO 28: READ R(Z): N
EXT Z
2890 DATA 10,20,30,40,50,60,70
2900 DATA -1,-2,-3,-4,-5,-6,-7
2910 DATA -10,-20,-30,-40,-50,-6
0,-70
2920 DATA 1,2,3,4,5,6,7
2930 FOR Z=1 TO 28: READ B(Z): N
EXT Z
2940 DATA -11,-22,-33,-44,-55,-6
6,-77
2950 DATA 11,22,33,44,55,66,77
2960 DATA 9,18,27,36,45,54,63
2970 DATA -9,-18,-27,-36,-45,-54
,-63
2980 RESTORE 2890
2990 FOR Z=1 TO 56: READ Q(Z): N
EXT Z
3000 FOR Z=1 TO 8: READ K(Z): NE
XT Z
3010 DATA 1,11,9,10,-10,-9,-11,-
1
3020 FOR Z=1 TO 64: READ S(Z): N
EXT Z
3030 DATA 46,56,36,66,47,57,45,5
5
3040 DATA 37,67,35,65,28,78,27,7
7
3050 DATA 44,54,26,75,38,68,17,8
7
3060 DATA 18,88,34,64,25,75,16,8
6
3070 DATA 48,24,74,15,85,14,84,4
3
3080 DATA 53,33,63,23,73,52,42,6
2
3090 DATA 32,83,13,72,22,12,82,4
1
3100 DATA 51,31,61,21,71,11,81,5
8
3110 PAPER 1: BORDER 1: INK 7: C
LS : RETURN
3120 FOR Z=11 TO 88: LET Z(Z)=A(
Z): NEXT Z
3130 FOR Z=11 TO 88: LET X=Z-10*
INT (Z/10)
3140 IF X=0 OR X=9 THEN GO TO 31
60
3150 LET A(Z)=Z(Z+9-X*2)
3160 NEXT Z
3170 FOR Z=11 TO 88: LET M=A(Z)
3180 IF M>=B THEN LET A(Z)=A(Z)+
PB-P
3190 IF M<=RB AND M>=BB THEN LET
 A(Z)=A(Z)-PB+P
```

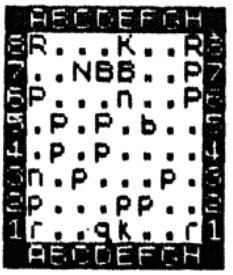
```
3200 NEXT Z
3210 RETURN
3310 LET COMMENT=INT (RND*4): BO
RDER RND*5+2: GO SUB 3330+10*COM
MENT
3320 BEEP 1,1: BEEP 1,2: BEEP 1,
-3
3330 BORDER 1: RETURN
3340 PRINT AT 0,0;"Bien hecho":
GO SUB 3370: RETURN
3350 PRINT AT 0,0;"Buen movimien
to": GO SUB 3370: RETURN
3360 PRINT AT 0,0;"Me cazo!"
3370 FOR I=1 TO 10: BEEP .05,I:
NEXT I
3380 PRINT AT 0,0;"              ":
 RETURN
```

A continuación se muestran dos pantallas copiadas al principio de una partida:





Y esta sería la situación un poco más avanzado el juego:

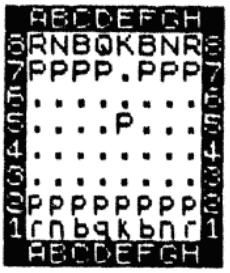


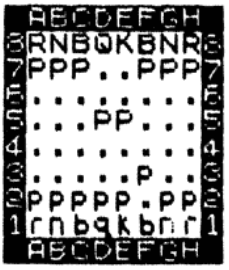
Es posible modificar el programa para que juegue contra sí mismo, obligándole a cambiar de bando a cada jugada. También puede obtenerse una copia permanente del juego copiando en papel el tablero cada dos jugadas. Para convertir el programa en «Auto-ajedrez», modifíquelo de la forma siguiente:

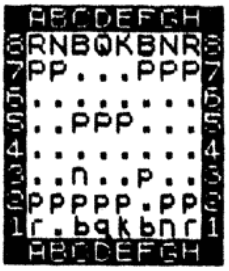
```
   10 REM AJEDREZ
   15 LET PR=1
   20 PRINT "POR FAVOR COLOQUE  "
; FLASH 1;"MAYUSCULAS"; FLASH 0,
"DESPUES APRIETE ENTER": INPUT A
$: CLS : PRINT ''"Gracias, por f
avor espere un     momento"
   30 GO SUB 2640: GO TO 60
   40 GO SUB 2350
   70 LET MM=0: LET UK=0
   80 IF A$="S" THEN STOP
   85 LET A$="X"
   90 IF A$="X" THEN PRINT AT 0,0
; FLASH 1;"Intercambio de fichas
": GO SUB 3120: PRINT AT 0,0;"
                                 ": LE
T A$=""
  100 IF PR=0 THEN COPY : LET PR=
1: GO TO 110
  105 LET PR=0
  110 DIM T(16): LET U=0: PRINT A
T 0,0; INK RND*5+2; PAPER 9;" Po
r favor espere un momento "
  120 FOR Q=1 TO 64: IF A(S(Q))>=
BB AND A(S(Q))<=RB THEN LET U=U+
1: LET T(U)=S(Q): IF A(S(Q))=KB
THEN LET KM=S(Q)
  130 NEXT Q: IF U<3 THEN GO TO 2
060
```

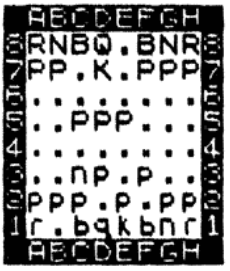
Si no desea una impresión sobre papel, borre las líneas 15, 1ØØ y 1Ø5.

Este es el principio de un juego del programa modificado:




A B C D E F G H
8 R N B Q K B N R 8
7 P P P . . P P P 7
6 6
5 . . . P P . . . 5
4 4
3 . . . . . P . . 3
2 P P P P P . P P 2
1 r n b q k b n r 1
A B C D E F G H

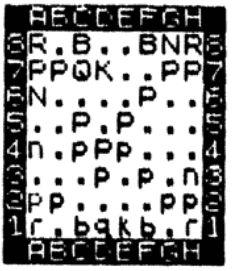

A B C D E F G H
8 R N B Q K B N R 8
7 P P . . . P P P 7
6 6
5 . . P P P . . . 5
4 4
3 . . n . . P . . 3
2 P P P P P . P P 2
1 r . b q k b n r 1
A B C D E F G H

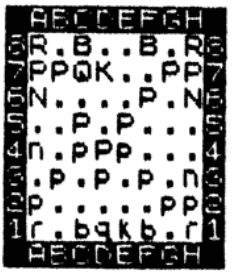

A B C D E F G H
8 R N B Q . B N R 8
7 P P . K . P P P 7
6 6
5 . . P P P . . . 5
4 4
3 . . n P . P . . 3
2 P P P . P . P P 2
1 r . b q k b n r 1
A B C D E F G H


A B C D E F G H
8 R N B Q . B N R 8
7 P P . K . P P P 7
6 6
5 . . P . P . . . 5
4 n . . P 4
3 . . . P . P . . 3
2 P P P . P . P P 2
1 r . b q k b n r 1
A B C D E F G H

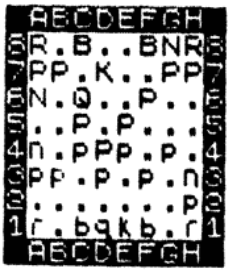

A B C D E F G H
8 R N B . . B N R 8
7 P P Q K . P P P 7
6 6
5 . . P . P . . . 5
4 n . . P P . . . 4
3 . . . P . P . . 3
2 P P P . . . P P 2
1 r . b q k b n r 1
A B C D E F G H


ABCDEFGH
8RNB..BNR8
7PPQK..PP7
6.....P..6
5..P.P...5
4n.PPp...4
3...P.p..3
2PP....PP2
1r.bqkbnr1
ABCDEFGH


ABCDEFGH
8R.B..BNR8
7PPQK..PP7
6N....P..6
5..P.P...5
4n.PPp...4
3...P.p.n3
2PP....PP2
1r.bqkb.r1
ABCDEFGH


ABCDEFGH
8R.B..B.R8
7PPQK..PP7
6N....P.N6
5..P.P...5
4n.PPp...4
3.P.P.P.n3
2P.....PP2
1r.bqkb.r1
ABCDEFGH


ABCDEFGH
8R.B..BNR8
7PPQK..PP7
6N....P..6
5..P.P...5
4n.PPp.P.4
3.P.P.P.n3
2P.....P.2
1r.bqkb.r1
ABCDEFGH

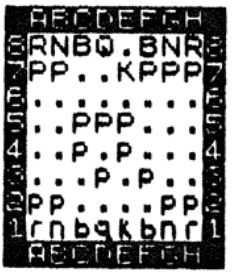

ABCDEFGH
8R.B..BNR8
7PP.K..PP7
6N.Q..P..6
5..P.P...5
4n.PPp.P.4
3PP.P.P.n3
2.....P.2
1r.bqkb.r1
ABCDEFGH

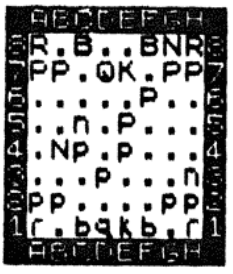
Dejar correr el programa modificado es una buena forma de comprobar el programa, ya que normalmente se examinan todos los GO TO y GO SUB. Si lo desea, elimine la impresión en papel y deje ejecutar el programa un día entero, jugando juego tras juego. Si se para en cualquier punto (otro que no sea concederse la victoria a sí mismo), podrá ver si existe algún error en el programa.

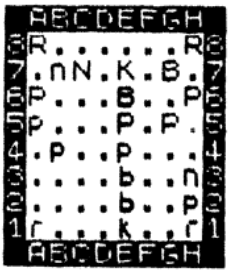
Puede hacerse una modificación muy simple para conseguir una impresión sobre papel a base de muestras aleatorias del juego, en lugar de imprimir el tablero cada dos movimientos. La modificación es como sigue:

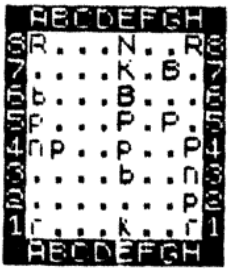
```
100 IF PR=0 AND RND<.3 THEN COP
Y
101 IF PR=0 THEN LET PR=1: GO T
O 110
105 LET PR=0
```

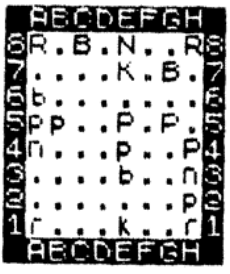
Aquí se muestran las «instantáneas» del juego obtenidas:

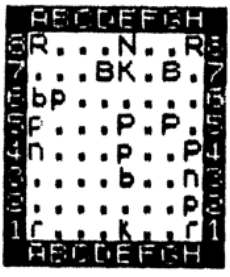


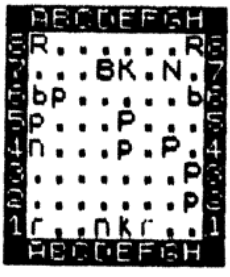


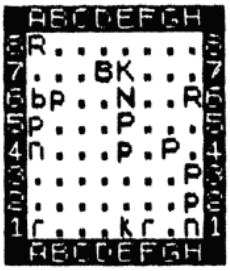


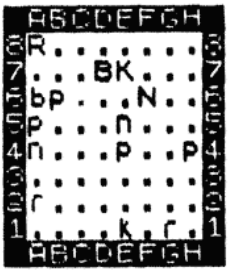












Si está interesado en profundizar en el juego del ajedrez y su mecanización, existe una gran cantidad de libros y artículos sobre el tema. A continuación se citan algunos de los más interesantes:

Intelligent Computer Games Levy, D., *Creative Computing,* noviembre de 1980 (págs. 158-163) y diciembre de 1980 (págs. 208-213)

Machine Intelligence: A Function of Human Ingenuity Georgiou, C.J., *Creative Computing,* junio de 1982 (págs. 124-135)

The Sargon Chronicle Ehara, T.H., *Creative Computing,* mayo de 1980 (págs. 42-44)

World Chess Championship Computer Ehara T.H., *Creative Computing,* enero de 1979 (págs. 134-136)

Attention, Chess Phreaks! Palenik, L., *Creative Computing,* enero de 1979 (pág. 78)

Tumult and the Toronto Tournament Dowhal, D., *Personal Computing,* mayo de 1978 (págs. 98-113)

A Romanian Rhapsody (source of the story of Felix at the start of this section), Friedberg, U., *Personal Computing,* octubre de 1978 (págs. 76-88)

Sargon vs. Microchess Martellaro. J., *on Computing,* invierno de 1979 (págs. 26-30)

Programming a Computer for Playing Chess Shannon, C.E., *Philosophical Magazine,* vol. 41 (7.ª serie) (págs. 256-275)

Chess Skill in Man and Machine Frey, Peter W. (Springer Verlag, 1977)

Technique in Chess Abrahams, G., (Dover Publications Inc. Estados Unidos, 1973)

The Complete Book of Chess Horowitz, I.A. y Rothenberg, e P.L. (Collier Macmillan, 1969)

Simulaciones de aventuras

La venganza del castillo encantado

Ninguna colección de programas de juegos se puede considerar completa sin incluir los juegos de aventuras. Se decidió incluir en este libro un gran programa (en términos de longitud) llamado LA VENGANZA DEL CASTILLO ENCANTADO. Se considera que el primer programa de aventuras que se ha producido se debe a Crowther and Woods, quienes escribieron un programa llamado Aventura en la Universidad de Stanford. El programa se ejecutó en un ordenador dotado de varios terminales servidos en tiempo compartido y la salida se imprimió en un rollo de papel mediante una impresora parecida a un Telex.

Aventura se convirtió en un programa enormemente popular y pronto aparecieron copias piratas en América e Inglaterra. Más y más aficionados a los ordenadores modificaron y mejoraron el programa, apareciendo muchas versiones, algunas de las cuales son radicalmente diferentes al programa original. No obstante, veinte años después Aventura todavía se vende y algunas compañías de software de Estados Unidos lo tienen en sus catálogos denominado «La Aventura Original».

Escribir una Aventura es un ejercicio fascinante. Es necesario construir un universo autocontenido y lógicamente consistente. Lógicamente consistente significa que el jugador puede cruzar un puente, ir por una carretera, volver por la misma carretera y volver a encontrar el puente o una razón muy convincente, por la cual el puente ya no está allí, tal como «Los extraterrestres de Epsilon IV se lo han llevado a un museo de su planeta mediante un transportador de material». Si una aventura no cumple estos requisitos, se la llama aventura aleatoria. Este tipo de aventuras son frustrantes y no consiguen retar al jugador.

En LA VENGANZA DEL CASTILLO ENCANTADO la acción tiene lugar en un castillo donde habitan cierto número de monstruos, incluidos «un hambriento guardián», «un temible soplafuegos» y el repugnante «guardián de la Laguna Negra». El jugador debe luchar (aunque puede intentar huir, con un cincuenta por ciento de posibilidades de

éxito) contra estos monstruos que vagan por el castillo. Además, aparecen hasta cuatro arcones que contienen sorpresas agradables o desagradables, una botella con una poción mágica que le puede beneficiar o perjudicar y otras cosas igualmente desconcertantes. Es poco probable que en cada juego encuentre todos los monstruos y sorpresas.

Usted no dispondrá de ningún mapa del castillo. Parte del interés de los juegos de aventuras consiste en tratar de deducir la relación espacial existente entre las distintas habitaciones que aparecen al vagar en el ambiente presentado por el ordenador. Para construir este programa se partió de un mapa del castillo escenario de la aventura. De esta forma las pistas que aparecen en las instrucciones PRINT reproducen con bastante realismo la relación entre las habitaciones. Se ha hecho así por dos motivos: para animarle a dibujarse su propio mapa al ir avanzando en el juego y para mostrarle con un ejemplo cómo se deben construir los ambientes de los juegos de aventuras. Una vez consiga dibujar un mapa que le parezca razonable, puede ir vagando por el castillo para comprobarlo.

El objetivo del juego es simple y está relacionado con una trampa que se ha introducido para hacer la construcción del mapa más difícil de lo que debiera ser sin ella. Todas las instrucciones que se pueden dar al programa son las direcciones de marcha (Norte, Sur, Este y Oeste) y todas las puertas están situadas también según estas direcciones. Para dibujar el mapa debe empezar colocando en un extremo del papel estas cuatro direcciones. La aventura empieza en la parte superior del papel y usted debe ir avanzando hacia la parte inferior (hacia el Sur). Las primeras indicaciones del programa son las siguientes:

```
Te encuentras en la entrada de
un antiguo y prohibido castillo
Estas en la fachada norte
mirando la desmoronada
estructura.
Te das cuenta de que el
portalon de entrada esta
abierto y sin guarda


Que quieres hacer ahora?
```

La última línea («¿Qué quieres hacer ahora?») sigue a cada descripción ambiental [aparte de unas pocas preguntas tales como «¿Quieres beber la poción?» que exigen una respuesta «S» (si) o «N» (no)] y puede ser respondida de siete formas:

Entre esto:	Y el ordenador entenderá esto:
«N»	Dirección Norte
«S»	Dirección Sur (respuesta a la primera pregunta)
«E»	Dirección Este
«O»	Dirección Oeste
«LU»	Luchar
«HU»	Huir
«Q»	Finalizar juego

Cualquier otra respuesta causará el borrado de la pantalla y su reimpresión.

Los juegos en los que los participantes representan un papel, y en los que generalmente existe un árbitro, siguen ciertas convenciones que determinan que la fuerza de carácter y la personalidad se repartan al principio de la partida mediante unos dados. A partir del número de los dados y mediante una tabla, a los jugadores se les asignan sus atributos (tales como la astucia). Cuando se produce un conflicto entre dos personajes, el resultado del conflicto se deduce de los atributos de los contendientes y de una tabla.

Para simplificar el sistema y para evitar cansarle tratando de entender complejas reglas antes de empezar a disfrutar el programa, se han tomado ciertas convenciones para asegurar que el Spectrum realice este trabajo por sí mismo, generando su personalidad y la de los monstruos. De esta forma se podrá deducir de una manera lógica e imparcial el resultado de cada combate que usted sostenga con los monstruos.

El argumento y el objetivo del juego son simples. Usted, como conocido explorador intrépido, ha llegado a las desmoronadas ruinas de un castillo. Siguiendo su imparable curiosidad, entra por el portalón del Norte, que está abierto. Una vez dentro nota que no puede salir por el mismo punto y se ve obligado a explorar todo el castillo buscando otra salida. Al encontrarse en la sala de la entrada «decorada con ricas telas», recuerda la vieja leyenda que pesa sobre el castillo. El último propietario fue un poderoso brujo que con su poder creó una legión de horrores para guardar su casa de los posibles intrusos. Esto había ocurrido trescientos años antes. A parte del portalón del Norte, que ha quedado mágicamente cerrado, la única vía para salir es la Laguna Negra, situada bajo el castillo, a la cual se llega mediante una puerta cerrada con llave. Se rumorea también que el mago, para proteger mejor su castillo, había creado una criatura de la que únicamente se conoce el nombre: El Guardián de la Laguna Negra.

Su personalidad se caracteriza por tres abributos: magia, fuerza y sabiduría. Cada criatura que se arrastra para toda la eternidad por las decadentes salas del castillo también tiene los mismos atributos en distintas proporciones. El estado de sus atributos aparece en la pantalla de la forma siguiente:

```
JORDI, tus atributos son:
     MAGIA:      15
     FUERZA:     16
     SABIDURIA: 14
```

Cuando se encuentra un monstruo y decide luchar con él (o intenta huir y no puede), sólo puede usar un atributo, pero lo puede escoger. Para hacer esta elección debe observar las diferencias existentes entre sus propios atributos y los de la criatura oponente, normalmente deberá escoger el atributo en el que tenga mayor ventaja para tener mayores posibilidades de ganar. Es decir, si su MAGIA fuera 2Ø y la del monstruo 4 y la diferencia entre los otros atributos fuera inferior, usted escogería la MAGIA para su lucha.

```
JORDI, tus atributos son:
     MAGIA:      15
     FUERZA:     16
     SABIDURIA: 14

Te encuentras en la sala de
entrada que esta decorada con
ricas telas. Las puertas estan
en direccion Este y Sur, y
se puede ver tambien un portal
abierto al oeste

Hay un Sopla-fuegos Temible. Su
sabiduria es 12, mientras que
su fuerza es 19 y su magia es 16


Que quieres hacer ahora?
```

```
JORDI, tus atributos son:
     MAGIA:      15
     FUERZA:     20
     SABIDURIA:  14

Esta es una sala respandeciente
del castillo: La Gran Sala.
Con un techo ricamente labrado.
Puede salir hacia el Norte y
hacia el Este por unas puertas
dobles de donde procede una
extrana musica. Desde las
ventanas del Oeste se puede ver
el Jardin y las ventanas de una
sala conmuchos cuadros colgados

Frente a ti hay un cofre
marcado con un gran numero 1

Lo abriras (S o N)?

Esta vez tienes una penaliza-
cion de 5 puntos de atributo

Y has perdido 5 puntos de sabidu
ria

Al final de esta crucial
batalla, tienes:
    MAGIA: 4
    FUERZA: 4
    SABIDURIA: 0
Todo ha terminado. Necesitarias
al menos 10 puntos en total
para escapar de las garras
del Guardian

Has luchado con valentia, pero
ahora seras comido por el
Guardian.....
```

Si este ejemplo le parece complicado, no se preocupe. El ordenador lo hace casi todo e ignora o rechaza sus entradas, si éstas no cumplen las reglas del juego.

Cada lucha le da la oportunidad de incrementar el atributo con el que batalla si gana. Pero si pierde, perderá puntos de atributo y, si los pierde todos, habrá perdido la partida. Le conviene obtener tantos puntos de atributo como pueda con vistas al Encuentro Final. Cuando

descubra la Laguna Negra (y, para frustrarle un poco, puede estar situada en dos sitios; por lo tanto sus mapas no serán 100 % transferibles de partida a partida) deberá superar tres batallas, una con cada atributo, contra El Guardián de la Laguna Negra.

Algún arcón de los que encontrará en el castillo puede contener «oro del dragón» y conviene guardarlo, porque se puede usar para comprar más puntos de atributo cuando encuentre al Guardián. Para sobrevivir debe haber superado todas las batallas y disponer de 10 puntos de atributo de reserva.

A continuación se describe el proceso de una lucha. Después de que haya seleccionado el atributo con el que quiere luchar, el ordenador mostrará en la pantalla la diferencia entre sus puntuaciones. Si, por ejemplo, usted decide luchar con SABIDURIA y su puntuación en este atributo es 15, mientras que la del monstruo es 2, el ordenador imprimirá (línea 524Ø) «La diferencia es 13», seguido de «y tu tienes ventaja», es decir, que su puntuación es más alta. A continuación y tal como se hace en los juegos de representación, una subrutina «tira» un dado de seis caras. El resultado es el «coste» del encuentro. Se genera un número aleatorio entre uno y la diferencia de puntuaciones. Este número se compara con su puntuación de atributo y con la del monstruo. El ganador es el que tiene la puntuación más parecida al número aleatorio. Usted puede cambiar este criterio por el contrario, si le parece más razonable. Si la diferencia entre puntuaciones de atributo es uno, el número aleatorio no puede ser otro que uno; por lo tanto ganará el participante con la puntuación más baja (con esto le sugerimos una posible trampa que le puede permitir ganar el juego). Dejamos para usted el trabajo de investigar qué ocurriría si la diferencia entre los atributos es cero.

En este juego ganar o perder una batalla se trata de forma muy distinta. Sus pérdidas o ganancias dependen del atributo que haya escogido y de la puntuación que tenga en el mismo.

Para seleccionar el atributo que usará para luchar debe entrar su inicial (M para la MAGIA, etc.) y el ordenador usa esta información en la rutina de la línea 534Ø para aplicar la penalización o la ganancia del encuentro. Si ha luchado con la MAGIA y ha perdido, se le deducen el número de puntos que ha salido en el dado, si los tiene. Es decir, si al empezar la batalla dispone de dos puntos de MAGIA y pierde, debiendo sufrir una penalización de tres puntos; no perderá ninguno. Como antes, esto resulta más simple en la práctica, ya que el ordenador determina si hay que aplicar o no la penalización, si pierde una batalla en la que ha seleccionado la fuerza la penalización como en el caso de la magia. Si es con la SABIDURIA con la que pierde, sólo perderá la mitad de la penalización que salga del dado, redondeada además por debajo (por lo tanto, si su penalización es uno y ha escogido la sabiduría para luchar, no le costará nada, ya que 0.5 se redondea a 0).

Resulta más divertido ganar. Usando la MAGIA puede ganar más que perder. Al ganar con MAGIA obtendrá un premio del doble de puntos de los expresados por el dado. Por el contrario, usando la FUERZA no conseguirá nada con una victoria (lo cual es mejor que perder, pero a pesar de todo no resulta conveniente luchar con los monstruos usando la fuerza). Ganando con la SABIDURIA obtiene la mitad del número del dado (recuerde que si pierde también pierde sólo la mitad).

El programa tiene una estructura preestudiada y, por lo tanto, es relativamente sencillo llegar a las secciones que requieren sus modificaciones. (Una vez haya resuelto el mapa codificado en el programa, usted puede desear cambiarlo, quizá de forma que con los mismos elementos se pueda formar un mapa más complicado.)

El programa está estructurado como sigue:

2Ø-17Ø	Información (en qué habitación está y dónde están las salidas) y principal ciclo del programa.
2ØØØ-2Ø5Ø	Descripciones de los monstruos.
4ØØØ-494Ø	Confrontación Final (La Laguna Negra, las tres últimas batallas y el final).
5ØØØ-562Ø	Acciones (entra su dirección de movimiento o su deseo de luchar o de huir; las luchas se dirigen en esta sección y se determinan las penalizaciones y premios a que hubiera lugar.)
7ØØØ-75ØØ	Contenidos (en esta sección podemos encontrar cosas tales como los cuatro arcones, la botella de poción mágica, etc.; estos elementos se colocan aleatoriamente en las salas).
8ØØØ-814Ø	Descripciones de las salas y llamada a la subrutina de los monstruos.
9ØØØ-967Ø	Inicialización.
99ØØ-999Ø	Rutina de sonido y retardo (se le llama repetidamente a lo largo del programa para añadirle color y darle la oportunidad de leer el contenido de la pantalla antes de borrarla).

La parte más importante de este programa y la clave de cualquier programa de aventuras es el mecanismo que mantiene la estructura del mundo generado en el juego. Este mecanismo determina la relación entre las distintas partes del sistema, comprobando los movimientos de forma que éstos resulten legales. Desde dos de las habitaciones del castillo se puede ver el Jardín, pero no sería un movimiento legal atravesar el Jardín para ir a la otra habitación, ya que no hay puertas que lo permitan. (El Jardín se ha incluido para hacer más interesante el paisaje descrito y para permitir a los que dibujan el plano del castillo disponer de un punto de referencia.)

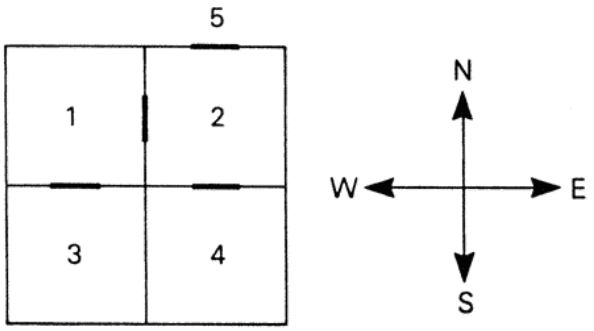
A continuación se describe el método usado para asegurar el seguimiento de los distintos componentes del sistema. No le recomendamos un estudio profundo de los próximos apartados antes de pro-

bar el juego, ya que, al hacerlo, le resultará menos divertida la partida. Veamos las líneas 957Ø a 968Ø. Estas instrucciones DATA contienen mucha información útil (también aparecen posiciones de reserva con vistas a la ampliación del programa que seguramente usted efectuará). Se analizará primero la cuarta línea del grupo (línea 96ØØ). Los primeros dos elementos son el número de la habitación situada al Norte. Hay un «OO», por lo tanto no existe ninguna salida hacia el Norte. Los dos elementos siguientes le informan que al Sur está la habitación siete, a continuación se informa que la habitación dos se encuentra al Este y por último que la cinco está al Sur. Los últimos dos ceros se usan a lo largo del programa para colocar el contenido de la habitación.

Mediante el sistema descrito, usted puede almacenar de una forma muy simple cualquier mapa. Si la acción estuviera situada en la habitación de la línea 96ØØ y el jugador intentara ir hacia el Norte, el ordenador mirará los primeros dos elementos de DATA. Si ambos son cero le responde «NO HAY SALIDA» y le vuelve a preguntar en qué dirección quiere ir. Si encuentra un número distinto a cero, la variable que almacena el número de habitación (que es Z en la mayor parte del programa) se iguala al valor (VAL) de estos primeros dos elementos.

El programa presentado también utiliza esta técnica de «seccionamiento de palabras» para almacenar los atributos de los monstruos en el vector M$ y los suyos en el J$. Haciéndolo así, resulta fácil modificarlos a lo largo del juego. Si usted derrota a un monstruo en una habitación y reduce su MAGIA a (por ejemplo) 9, si lo vuelve a encontrar en otra habitación, sus atributos deben haber quedado modificados. Si este encuentro se repitiera muchas veces, usted podría destruir al monstruo. Si se consigue que un elemento de la historia sea independiente del punto donde se encuentra, el programa resulta coherente.

Supóngase una Aventura con un universo muy simple. Sólo cuatro habitaciones relacionadas de la forma siguiente:



Las puertas se simulan con trazos más gruesos en las paredes. Las habitaciones y su situación podrían describirse de la siguiente forma:

Habitación 1	«ØØØ3Ø2ØØ» (nada al Norte, la habitación 3 al Sur, la 2 al Este y nada al Oeste)
Habitación 2	«Ø5Ø4ØØØ1» (nótese que el «exterior» de nuestro universo ha de quedar identificado)
Habitación 3	«Ø1ØØØØØØ» (pocas salidas en esta sala)
Habitación 4	«Ø2ØØØØØØ» (tampoco en ésta)

Suponga que estamos situados en la habitación dos. La instrucción PRINT que describe la habitación podría decir: «Es una habitación pequeña y cuadrada. Hay una puerta hacia el mundo exterior en la pared Norte, otra hacia el Sur y una más al Oeste. ¿Hacia dónde quiere ir?» Si el jugador contesta «E», el ordenador mira los «elementos del Este» del DATA que representa la habitación dos (son los elementos quinto y sexto) y ve que sólo hay dos ceros. Ahora el ordenador sabe que el jugador no puede moverse en esta dirección y le informa: «No puede atravesar las paredes», quedando a la espera de una nueva dirección. Si el jugador entra «S», indicando que desea ir en dirección Sur, el ordenador mira a los elementos tres y cuatro de la misma habitación y encuentra el número «04». Esto significa que el jugador se va a mover de la habitación dos a la cuatro. La variable que almacena la habitación en la que se encuentra el jugador se iguala al valor (VAL) de los elementos consultados. Nótese que en la instrucción PRINT se describe la habitación dos y sus salidas, pero no se informa del número de las habitaciones contiguas. Esta información está contenida en la instrucción DATA y no se muestra en la pantalla.

Le sugerimos entrar el programa y ejecutarlo hasta que empiece a entender su funcionamiento. Esto le dará una idea del plano del castillo. Le será fácil dibujarlo si hace una serie de cuadros en un papel y les pone nombres a medida que el programa los vaya descubriendo, pintando además corredores de conexión entre ellos de acuerdo con las informaciones que aparezcan en la pantalla. Este proceso lo puede repetir varias veces hasta que el mapa sea seguro. Después de ejecutar el programa varias veces, puede volver al libro, seguir el listado del programa y examinar los métodos de modificación que se describen.

```
  1 LET Q=0
 10 REM LA VENGANZA DEL CASTILL
O ENCANTADO
 20 GO SUB 9000
 30 CLS
 35 IF J$="000000" THEN GO TO 9
0
```

```
  40 PRINT INVERSE 1;N$;", tus a
tributos son:"
  50 IF VAL (J$(1 TO 2))>0 THEN
PRINT TAB 4; INVERSE 1;"MAGIA:
   ";J$(1 TO 2)
  60 IF VAL (J$(3 TO 4))>0 THEN
PRINT TAB 4; INVERSE 1;"FUERZA:
   ";J$(3 TO 4)
  70 IF VAL (J$(5 TO 6))>0 THEN
PRINT TAB 4; INVERSE 1;"SABIDURI
A:";J$(5 TO 6)
  80 IF MONEY>0 THEN PRINT TAB 4
; INVERSE 1;"RIQUEZA:  ";MONEY;"
Ptas"
  90 IF J$="000000" THEN PRINT "
La aventura ha terminado","Has a
gotado todas tus fuerzas.","Luch
aste bravamente y bien","pero no
 has podido aguantar la","batall
a....Adios...": STOP
 100 GO SUB PAUSE
 110 GO SUB ROOM
 120 LET M=0: IF Z>1 THEN IF RND
>.5 THEN GO SUB 7000: POKE 23692
,-1
 130 GO SUB PAUSE
 140 GO SUB ACTION
 150 GO SUB PAUSE
 170 GO TO 30
 180 REM ***************
2000 REM SUBRUTINA MONSTRUO
2010 IF Q=1 THEN PRINT "Hay un f
urioso Guardian en la","habitaci
on. Tiene una magia de ",M$(1,2
TO 3);", su fuerza es ";M$(1,4 T
O 5);" y su","sabiduria es ";M$(
1,6 TO 7)
2020 IF Q=2 THEN PRINT "Hay un S
opla-fuegos Temible. Su","sabidu
ria es ";M$(2,6 TO 7);", mientra
s que","su fuerza es ";M$(2,4 TO
 5);" y su magia es ";M$(2,2 TO
3)
2030 IF Q=3 THEN PRINT "Horror!
Has caido en el horri-","ble lug
ar del Alma-en-Pena.","Su fuerza
 es ";M$(3,4 TO 5);", su magia e
s ";M$(3,2 TO 3),"y su sabiduria
 es ";M$(3,6 TO 7)
2040 IF Q=4 THEN PRINT "Has trop
ezado con algo en la","oscuridad
.": GO SUB PAUSE: PRINT "Se desp
ierta y te encuentras","cara a c
ara con el Rompe-Rodi-","llas. S
u magia es ";M$(4,2 TO 3);", su
fuerza es ";M$(4,4 TO 5);" y su
sabiduria es ";M$(4,6 TO 7)
2050 IF Q=5 THEN PRINT "Ahora si
 que tienes problemas!!": GO SUB
```

```
 PAUSE: PRINT "Has encontrado al
 peor enemigo","del castillo: El
 Hombre Lobo.","Con una fuerza d
e ";M$(5,4 TO 5);", sabiduria de
 ";M$(5,6 TO 7);" y una magia de
 ";M$(5,2 TO 3);" , es un enemig
o dificil de vencer"
2060 GO SUB PAUSE
2070 RETURN
2080 REM ***************
4000 REM  FINAL DEL JUEGO
4010 PRINT "Has caido en el pant
ano de","fango que rodea La Lagu
na Negra","bajo el castillo. Par
a escapar","del castillo debias
haber","luchado con el Guardian
de la","Laguna Negra"
4020 PRINT '"En la lucha deberas
 usar todos","tus atributos... y
","vas a necesitar 10 puntos par
a","escapar"
4030 GO SUB PAUSE
4040 IF MONEY>0 THEN PRINT "Tien
es ";MONEY;" Ptas. en oro"
4050 GO SUB PAUSE
4060 PRINT "Los atributos del Gu
ardian:"
4070 PRINT TAB 3;"MAGIA: ";M$(6,
2 TO 3)
4080 PRINT TAB 3;"FUERZA: ";M$(6
,4 TO 5)
4090 PRINT TAB 3;"SABIDURIA: ";M
$(6,6 TO 7)
4100 PRINT "Sus atributos son:"
4110 PRINT TAB 3;"MAGIA: ";J$(1
TO 2)
4120 PRINT TAB 3;"FUERZA: ";J$(3
 TO 4)
4130 PRINT TAB 3;"SABIDURIA: ";J
$(5 TO 6)
4140 GO SUB PAUSE
4150 IF MONEY<100 THEN GO TO 427
0
4160 PRINT '"Puedes comprar punt
os de","atributo a 100 Ptas cada
 uno"
4170 PRINT "Si quieres comprarlo
s, entra la","inicial del atribu
to que","quieras seguido del num
ero"
4180 PRINT "de puntos. Entra 'N'
 si no","quieres comprar"
4190 INPUT FLASH 1;"Atributo? ";
E$
4200 IF E$="N" THEN GO TO 4270
4210 INPUT FLASH 1;"Cantidad? ";
AM
4220 IF MONEY-AM<1 OR AM<100 THE
N GO TO 4210
```

```
4230 IF E$="M" THEN LET J$(1 TO
2)=STR$ (VAL (J$(1 TO 2))+INT (A
M/100))
4240 IF E$="F" THEN LET J$(3 TO
4)=STR$ (VAL (J$(3 TO 4))+INT (A
M/100))
4250 IF E$="S" THEN LET J$(5 TO
6)=STR$ (VAL (J$(5 TO 6))+INT (A
M/100))
4255 PRINT "MAGIA: ";J$(1 TO 2);
" FUERZA: ";J$(3 TO 4),"SABIDURI
A: ";J$(5 TO 6);" ORO $";MONEY
4260 LET MONEY=MONEY-AM: IF MONE
Y>99 THEN GO TO 4190
4270 CLS : PRINT "Ahora el Ultim
o Examen...": GO SUB PAUSE
4275 INPUT FLASH 1; BRIGHT 1; PA
PER 2;"Aprieta ENTER cuando este
s listopara luchar ",A$: GO SUB
PAUSE: CLS
4280 PRINT '"Primero, Magia:"
4290 LET MH=VAL (J$(1 TO 2)): LE
T MG=VAL (M$(6,2 TO 3))
4300 PRINT TAB 3;"Tu: ";MH;"  Gu
ardian: ";MG
4310 LET DIFF=ABS (MH-MG)
4320 PRINT "La diferencia es ";D
IFF
4330 IF MH>MG THEN PRINT "a tu f
avor"
4340 IF MG>MH THEN PRINT "y el G
uardian ha ganado"
4350 GO SUB PAUSE: LET COST=INT
(RND*((MH+MG)/2))+1
4360 PRINT '"Esta vez tienes una
 penaliza-","cion de ";COST;" pu
ntos de atributo": GO SUB PAUSE
4370 LET RESULT=INT (RND*DIFF)+1
4380 LET RESHU=ABS (RESULT-MH)
4390 LET RESGA=ABS (RESULT-MG)
4400 IF RESHU>RESGA THEN PRINT "
Y has ganado ";COST;" puntos mag
icos": LET MH=MH+COST: LET J$(1
TO 2)=STR$ (MH): GO TO 4420
4410 PRINT "Y has perdido ";COST
;" puntos magicos": LET MH=MH-CO
ST: LET J$(1 TO 2)="00": IF MH>0
 THEN LET J$(1 TO 2)=STR$ (MH)
4480 INPUT FLASH 1; PAPER 7; INK
 2;"Pulsa ENTER cuando estes lis
t","para continuar este crucial"
,"reto",,A$: GO SUB PAUSE: CLS
4485 PRINT "Ahora Fuerza:"
4490 LET MH=VAL (J$(3 TO 4)): LE
T MG=VAL (M$(6,4 TO 5))
4500 PRINT TAB 3;"Tu: ";MH;"  Gu
ardian: ";MG
4510 LET DIFF=ABS (MH-MG)
```

```
4520 PRINT "La diferencia es ";D
IFF
4530 IF MH>MG THEN PRINT "y tu g
anas"
4540 IF MG>MH THEN PRINT "y el G
uardian gana"
4550 GO SUB PAUSE: LET COST=INT
(RND*((MH+MG)/2))+1
4560 PRINT '"Esta vez tienes una
 penaliza-","cion de ";COST;"  p
untos de atributo": GO SUB PAUSE
4570 LET RESULT=INT (RND*DIFF)+1
4580 LET RESHU=ABS (RESULT-MH)
4590 LET RESGA=ABS (RESULT-MG)
4600 IF RESHU>RESGA THEN PRINT "
Has ganado ";COST;" puntos de fu
erza": LET MH=MH+COST: LET J$(3
TO 4)=STR$ (MH): GO TO 4680
4610 PRINT "Has perdido ";COST;"
 puntos de fuerza": LET J$(3 TO
4)="00": IF MH>0 THEN LET MH=MH-
COST: LET J$(3 TO 4)=STR$ (MH)
4670 LET RESULT=INT (RND*DIFF)+1
4680 INPUT FLASH 1; BRIGHT 1; PA
PER 2; INK 7;"Ahora el reto fina
l.","Pulsa "; INVERSE 1;"ENTER";
 INVERSE 0;" cuando te atrevas",
A$: GO SUB PAUSE: CLS
4685 PRINT '"Ahora Sabiduria:"
4690 LET MH=VAL (J$(5 TO 6)): LE
T MG=VAL (M$(6,6 TO 7))
4700 PRINT TAB 3;"Tu: ";MH;"  Gu
ardian: ";MG
4710 LET DIFF=ABS (MH-MG)
4720 PRINT "La diferencia es ";D
IFF
4730 IF MH>MG THEN PRINT "y vas
ganando"
4740 IF MG>MH THEN PRINT "y el G
uardian gana"
4750 GO SUB PAUSE: LET COST=INT
(RND*(MH+MG))+1
4760 PRINT '"Esta vez tienes una
 penaliza-","cion de ";COST;" pu
ntos de atributo": GO SUB PAUSE
4770 LET RESULT=INT (RND*DIFF)+1
4780 LET RESHU=ABS (RESULT-MH)
4790 LET RESGA=ABS (RESULT-MG)
4800 IF RESHU>RESGA THEN PRINT "
Y has ganado ";COST;" puntos de
sabiduria": LET MH=MH+COST: LET
J$(5 TO 6)=STR$ (MH): GO TO 4820
4810 PRINT "Y has perdido ";COST
;" puntos de sabiduria": LET MH=
MH-COST: LET J$(5 TO 6)="00": IF
 MH>0 THEN LET J$(5 TO 6)=STR$ (
MH)
4850 LET A=VAL (J$(1 TO 2))
4860 LET B=VAL (J$(3 TO 4))
```

```
4870 LET C=VAL (J$(5 TO 6))
4880 GO SUB PAUSE
4890 PRINT '"Al final de esta cr
ucial","batalla, tienes:"
4900 PRINT TAB 3;"MAGIA: ";A
4910 PRINT TAB 3;"FUERZA: ";B
4920 PRINT TAB 3;"SABIDURIA: ";C
4930 IF A+B+C>9 THEN PRINT ' FLA
SH 1; BRIGHT 1; PAPER 7; INK 2;"
Lo has logrado, Oh heroe de","es
tos oscuros y peligrosos","tiemp
os. Yo te nombro "; INVERSE 1;"S
ir"; INVERSE 0;" ";N$: GO SUB PA
USE: STOP
4940 PRINT "Todo ha terminado. N
ecesitarias","al menos 10 puntos
 en total","para escapar de las
garras","del Guardian"
4950 PRINT '"Has luchado con val
entia, pero","ahora seras comido
 por el","Guardian....."
4960 GO SUB PAUSE: STOP
4970 REM **************
5000 REM SUBRUTINA DE ACCION
5005 LET D=4: IF B$(Z,9)="0" THE
N LET D=1
5010 PRINT '"Que quieres hacer a
hora?"
5015 INPUT Z$: IF Z$="Q" THEN ST
OP
5020 IF Z$="" THEN CLS : LET Z$=
"#"
5022 IF (D=4 AND Z$<>"LU") OR (D
=4 AND Z$<>"HU") THEN LET D=0: G
O TO 5160
5025 IF Z$="LU" OR Z$="HU" THEN
GO TO 5130
5030 IF Z$="N" AND B$(Z,1 TO 2)=
"00" THEN PRINT ,"NO SE PUEDE SA
LIR": GO TO 5010
5040 IF Z$="S" AND B$(Z,3 TO 4)=
"00" THEN PRINT "POR AQUI NO HAY
 PUERTA": GO TO 5010
5050 IF Z$="E" AND B$(Z,5 TO 6)=
"00" THEN PRINT "ESTO NO ES POSI
BLE": GO TO 5010
5060 IF Z$="O" AND B$(Z,7 TO 8)=
"00" THEN PRINT "NO PUEDES ATRA
VESAR LAS PAREDES": GO TO 5010
5070 IF Z$="N" THEN LET Z=VAL (B
$(Z,1 TO 2)): RETURN
5080 IF Z$="S" THEN LET Z=VAL (B
$(Z,3 TO 4)): RETURN
5090 IF Z$="E" THEN LET Z=VAL (B
$(Z,5 TO 6)): RETURN
5100 IF Z$="O" THEN LET Z=VAL (B
$(Z,7 TO 8)): RETURN
5110 IF Z$<>"LU" OR Z$<>"HU" THE
N RETURN
```

```
5130 IF B$(Z,9)="0" THEN PRINT "
No hay nadie para luchar contra"
,"el": GO TO 5010
5140 IF Z$="HU" THEN LET D=INT (
RND*2)
5150 IF D=1 THEN PRINT "En que d
ireccion?": GO TO 5015
5160 IF D=0 THEN PRINT "No!! Deb
es quedarte y luchar"
5170 PRINT '"Con que atributo,
prefieres","luchar?"
5180 LET Q=VAL (B$(Z,9))
5190 INPUT Z$: IF Z$<>"M" AND Z$
<>"F" AND Z$<>"S" THEN GO TO 519
0
5200 IF Z$="M" THEN LET HUM=VAL
(J$(1 TO 2)): LET MON=VAL (M$(Q,
2 TO 3))
5210 IF Z$="F" THEN LET HUM=VAL
(J$(3 TO 4)): LET MON=VAL (M$(Q,
4 TO 5))
5220 IF Z$="S" THEN LET HUM=VAL
(J$(5 TO 6)): LET MON=VAL (M$(Q,
6 TO 7))
5230 LET DIFF=ABS (HUM-MON)
5240 PRINT '' FLASH 1; BRIGHT 1;
 INK 2; PAPER 7;"La diferencia e
s ";DIFF
5250 IF HUM>MON THEN PRINT FLASH
 1; BRIGHT 1; PAPER 2; INK 7;"y
tienes ventaja"
5260 IF HUM<MON THEN PRINT FLASH
 1; BRIGHT 1; PAPER 2; INK 7;"y
tiene ventaja"
5270 LET COST=INT (RND*6)+1
5280 PRINT '' FLASH 1; BRIGHT 1;
 PAPER 3; INK 9;"Esta lucha tien
e una penaliza-","cion de "; INV
ERSE 1;COST; INVERSE 0;" puntos"
;
5290 LET RESULT=INT (RND*DIFF)+1
5300 GO SUB PAUSE
5310 LET RESHU=ABS (RESULT-HUM)
5320 LET RESMO=ABS (RESULT-MON)
5330 IF RESHU<RESMO THEN GO TO 5
390
5340 PRINT '' FLASH 1; PAPER 2;
INK 6;"Has sido derrotado!"
5350 IF Z$="M" THEN LET M$(Q,2 T
O 3)=STR$ (VAL (M$(Q,2 TO 3))+2*
COST): IF VAL (J$(1 TO 2))>=COST
 THEN LET J$(1 TO 2)=STR$ (VAL (
J$(1 TO 2))-COST)
5360 IF Z$="F" AND VAL (J$(3 TO
4))>=COST THEN LET J$(3 TO 4)=ST
R$ (VAL (J$(3 TO 4))-COST)
5370 IF Z$="S" THEN LET M$(Q,6 T
O )=STR$ (VAL (M$(Q,6 TO 7))+INT
  (COST/2)): IF VAL (J$(5 TO 6))>
```

```
=COST THEN LET J$(5 TO 6)=STR$ (
VAL (J$(5 TO 6))-COST)
5380 GO TO 5440
5390 REM VICTORIA HUMANA
5400 PRINT '' FLASH 1; BRIGHT 1;
 INK 6; PAPER 2;"Lo has derrotad
o!"
5410 IF Z$="M" THEN LET J$(1 TO
2)=STR$ (VAL (J$(1 TO 2))+2*COST
): IF VAL (M$(Q,2 TO 3))>=COST T
HEN LET M$(Q,2 TO 3)=STR$ (VAL (
M$(Q,2 TO 3))-COST)
5420 IF Z$="F" THEN LET M$(Q,4 T
O 5)=STR$ (VAL (M$(Q,4 TO 5))-CO
ST)
5430 IF Z$="S" THEN LET J$(5 TO
6)=STR$ (VAL (J$(5 TO 6))+INT (C
OST/2)): IF VAL (M$(Q,6 TO 7))>=
COST THEN LET M$(Q,6 TO 7)=STR$
(VAL (M$(Q,6 TO 7))-COST)
5440 GO SUB PAUSE
5450 PRINT "Despues de esta lucha tus","atributos son:"
5460 PRINT TAB 3;"MAGIA:";J$(1 T
O 2)
5470 PRINT TAB 3;"FUERZA:";J$(3
TO 4)
5480 PRINT TAB 3;"SABIDURIA:";J$
(5 TO 6)
5490 PRINT '"Y los del  ";
5500 IF Q=1 THEN PRINT "guardian
"
5510 IF Q=2 THEN PRINT "Temible"
5520 IF Q=3 THEN PRINT "Alma-en-
pena"
5530 IF Q=4 THEN PRINT "Rompe-ro
dillas"
5540 IF Q=5 THEN PRINT "Hombre-l
obo"
5550 PRINT "son:"
5560 PRINT TAB 3;"MAGIA:  ";M$(Q,
2 TO 3)
5570 PRINT TAB 3;"FUERZA:  ";M$(Q
,4 TO 5)
5580 PRINT TAB 3;"SABIDURIA:  ";M
$(Q,6 TO 7)
5590 GO SUB PAUSE
5600 PRINT "Pulsa ENTER para con
tinuar"
5610 INPUT T$: GO SUB PAUSE: CLS

5620 LET K=2+INT (RND*8)
5630 IF K=Z THEN GO TO 5620
5640 LET B$(K,9)=B$(Z,9)
5650 LET B$(Z,9)="0"
5660 IF RND>.5 THEN RETURN
7000 REM CONTENIDO
7005 IF B$(Z,9)<>"0" THEN LET M=
1: RETURN : REM NO TE CONCEDE 'C
```

```
ONTENIDO' SI HAY UN MONSTRUO PAR
A LUCHAR
7010 PRINT : GO SUB 7000+100*INT
 (RND*5+1)
7020 GO SUB PAUSE
7030 RETURN
7100 LET CHEST=CHEST+1: IF CHEST
=5 THEN RETURN
7110 PRINT "Frente a ti hay un c
ofre","marcado con un gran numer
o ";CHEST
7120 PRINT '"Lo abriras (S o N)?
"
7130 GO SUB 7600
7150 IF Z$="N" THEN RETURN
7160 LET J=INT (RND*3): GO SUB P
AUSE
7170 IF J=0 THEN LET CASH=100+IN
T (RND*300): PRINT "Contiene oro
 del Dragon por","valor de ";CAS
H;" Ptas!": LET MONEY=MONEY+CASH
: RETURN
7180 IF J=1 THEN PRINT "Ha salta
do fuera un duende ","apunaland
ote!": LET LOSS=INT (RND*6)+1: I
F VAL (J$(3 TO 4))-LOSS>=0 THEN
LET J$(3 TO 4)=STR$ (VAL (J$(3 T
O 4))-LOSS): RETURN
7190 IF J=2 THEN PRINT "Sale un
extrano humo, que te","produce s
ueno y te disminuye","tu poder m
agico": LET LOSS=INT (RND*6)+1:
IF VAL (J$(1 TO 2))-LOSS>=0 THEN
 LET J$(1 TO 2)=STR$ (VAL (J$(1
TO 2))-LOSS): RETURN
7200 IF POTION=1 THEN GO TO 7010
7210 LET POTION=1
7220 PRINT "Ves una botella pequ
ena grabada"
7230 PRINT "con curiosas letras
torcidas."
7240 PRINT "Beberas el brebaje q
ue hay den-","tro de ella (S o N
)?"
7250 GO SUB 7600
7260 IF Z$="N" THEN RETURN
7270 GO SUB PAUSE
7280 IF RND>.6 THEN PRINT "Conti
ene un brebaje para aumen-","tar
 tu sabiduria": LET J$(5 TO 6)=S
TR$ (VAL (J$(5 TO 6))=INT (RND*6
+1)): RETURN
7290 PRINT "Contiene un brebaje
que te de-","bilita, transmitien
dote sueno": GO SUB PAUSE: LET L
OSS=INT (RND*6)+1: IF VAL (J$(3
TO 4))<LOSS THEN RETURN
7295 LET J$(3 TO 4)=STR$ (VAL (J
$(3 TO 4))-LOSS): RETURN
```

```
7300 IF SCROLL=1 THEN RETURN
7310 LET SCROLL=1
7320 PRINT "Ves un rollo de pape
l.","Deseas leerlo (S o N)?"
7325 GO SUB 7600: IF Z$="N" THEN
 RETURN
7330 IF RND>.5 THEN PRINT "No pu
edes entender el idioma.": GO SU
B PAUSE: RETURN
7335 PRINT "Contiene un hechizo
magico.","Quieres leerlo (S o N)
?"
7340 GO SUB 7600: GO SUB PAUSE:
IF Z$="N" THEN RETURN
7350 IF RND>.5 THEN PRINT FLASH
1; BRIGHT 1; INK 7; PAPER 2;"Era
 un hechizo beneficioso": GO SUB
 PAUSE: LET J$(1 TO 2)=STR$ (VAL
 (J$(1 TO 2))+INT (RND*6+1)): RE
TURN
7360 PRINT PAPER 2; INK 7; FLASH
 1; BRIGHT 1;"  Era un hechizo p
erjudicial!!!",: GO SUB PAUSE: I
F VAL (J$(3 TO 4))>5 THEN LET J$
(3 TO 4)=STR$ (VAL (J$(3 TO 4))-
INT (RND*6+1)): RETURN
7400 IF SAFE=1 THEN GO TO 7010
7405 LET SAFE=1
7410 PRINT "En la pared hay una
caja dorada","pequena y en su pa
rte frontal","hay una llave"
7415 PRINT "Quieres abrir la caj
a? (S o N)"
7420 GO SUB 7600
7425 IF Z$="N" THEN RETURN
7430 IF RND>.3 THEN GO TO 7460
7435 GO SUB PAUSE: PRINT '"Un mu
rcielago sale chillando","y unde
 sus dientes en tu","garganta!!!
"
7440 GO SUB PAUSE
7445 PRINT "Luchas con el y...":
 GO SUB PAUSE: PRINT "...finalme
nte le retuerces el","cuello"
7450 IF VAL (J$(3 TO 4))>5 THEN
LET J$(3 TO 4)=STR$ (VAL (J$(3 T
O 4))-INT (RND*6+1))
7455 GO SUB PAUSE: GO TO 7620
7460 PRINT "Se oye un coro de vo
ces","angelicales": GO SUB PAUSE
: PRINT "Te sientes reconfortado
"
7480 LET J$(3 TO 4)=STR$ (VAL (J
$(3 TO 4))+INT (RND*6+1))
7490 RETURN
7500 GO TO 7100
7600 LET Z$=INKEY$
7610 IF Z$<>"N" AND Z$<>"S" THEN
 GO TO 7600
```

```
7615 PRINT
7620 BEEP .5,1: BEEP .5,2: BEEP
.5,-1: RETURN
7630 REM ****************
8000 REM CONTENIDO HABITACIONES
8010 IF Z=1 THEN PRINT "Te encue
ntras en la entrada de","un anti
guo y prohibido castillo","Estas
 en la fachada norte","mirando l
a desmoronada","estructura.",,"T
e das cuenta de que el","portalo
n de entrada esta","abierto y si
n guarda"
8020 IF Z=2 THEN PRINT "Te encue
ntras en la sala de","entrada qu
e esta decorada con","ricas tela
s. Las puertas estan","en direcc
ion Este y Sur, y","se puede ver
 tambien un portal","abierto al
oeste"
8030 IF Z=3 THEN PRINT "Esto es
solamente un almacen.","Solo tie
ne una salida, la que","has usad
o para entrar, al oeste"
8040 IF Z=4 THEN PRINT "Estas en
 una sala pequena con","una vell
a escultura a la Diosa","de la L
una en un pedestal en la","esqui
na Nordeste. Es la Camara","Real
. Las puertas estan al Sur,","Oe
ste y al Este"
8050 IF Z=5 THEN PRINT "La Sala
de las Conspiraciones,","una hab
itacion forrada en","madera, lle
na de rumores i","susurros, con
salidas al Este","y al Sur, de l
a que llega un","olor a azufre y
 un ambiente","sobrenatural"
8060 IF Z=6 THEN PRINT "Has entr
ado en la Guarida de","Brujos, c
on un cazo hirbiendo","en un fue
go de llamas verdes en","la esqu
ina Sudeste.","Esta sala con hed
or de azufre","y el eco de viejo
s sortilegios","la puedes abando
nar hacia el","Norte, Sur o Este
"
8070 IF Z=7 THEN PRINT "Te encue
ntras en un lugar que","parece t
ranquilo y apacible.","Es la Gal
eria de Pintura del","Castillo,
con un gran cuadro","del legenda
rio Guardian de La","Laguna Negr
a a la izquierda de","la ventana
 de la pared este.","Por la vent
ana puede ver las","ventanas de
Gran Sala, al otro","lado del Ja
rdin.","Las salidas son hacia el
 Norte","y hacia el Oeste"
```

```
8080 IF Z=8 THEN PRINT "Esta es
una sala respandeciente","del ca
stillo: La Gran Sala.","Con un t
echo ricamente labrado.","Puede
salir hacia el Norte y","hacia e
l Este por unas puertas","dobles
 de donde procede una","extrana
musica. Desde las","ventanas del
 Oeste se puede ver","el Jardin
y las ventanas de una","sala con
muchos cuadros colgados"
8090 IF Z=9 THEN PRINT "La music
a de un cuarteto de","cuerda lle
na la sala. Puede","abandonar la
 sala por unas","puertas situada
s al Oeste o por","una hacia el
Sur"
8100 IF Z=10 THEN PRINT "Estas a
hora en el Santuario del","Silen
cio. Una sala fria y","humeda. T
iene una salida hacia","el Norte
";: IF B$(10,3 TO 4)="12" THEN P
RINT " y una hacia el Sur"
8110 IF Z=11 THEN PRINT "Este de
be ser el Vestibulo de","las Apa
riciones, una oscura","sala, don
de la leyenda dice que","algunas
 noches se puede oir al","Guardi
an de la Laguna Negra.","Hay una
 puerta hacia el Norte",: IF B$(
11,3 TO 4)="12" THEN PRINT "y un
a al Sur"
8120 PRINT : IF Z=12 THEN GO TO
4000
8125 IF z<>1 THEN IF B$(Z,9)="0"
 AND RND>.55 THEN LET B$(Z,9)=ST
R$ (INT (RND*5)): REM BORRE PARA
 TENER MENOS MONSTRUOS
8130 IF B$(Z,9)<>"0" THEN LET Q=
VAL (B$(Z,9)): GO SUB 2000
8140 RETURN
8150 REM **************
9000 REM INICIALIZACION
9010 RANDOMIZE
9020 LET Z=1
9030 LET PAUSE=9900
9040 LET ROOM=8000
9050 LET ACTION=5000
9060 LET MONEY=0
9070 LET CHEST=0
9080 LET POTION=0
9090 LET SCROLL=0
9100 LET SAFE=0
9110 DIM B$(12,10)
9120 DIM M$(6,7): DIM J$(5)
9130 POKE 23692,-1: POKE 23658,8
9140 INPUT "Como te llamas? ";N$
9150 PAPER 1: INK 7: BORDER 1: C
LS
```

```
9160 PRINT "Hola, ";N$: PRINT "P
or favor espera un momento..."
9320 REM FILL ROOMS
9330 FOR T=1 TO 12
9340 READ T$
9350 LET B$(T)=T$
9360 NEXT T
9365 LET B$((10+INT (RND*2)),3 T
O 4)="12"
9370 REM DISTRIBUYE Y CREA MONST
RUOS
9380 LET M$(1,1)=STR$ (INT (RND*
9)+1)
9390 FOR A=2 TO 5
9400 LET M$(A,1)=STR$ (INT (RND*
5))
9440 LET B$(VAL (M$(A,1)),9)=M$(
A,1)
9450 NEXT A: LET B$(1,9)="0"
9460 REM CARACTERISTICAS
9470 FOR A=1 TO 6
9480 LET M$(A,2 TO 3)=STR$ (INT
(RND*11)+10)
9490 LET M$(A,4 TO 5)=STR$ (INT
(RND*11)+10)
9500 LET M$(A,6 TO 7)=STR$ (INT
(RND*11)+10)
9510 NEXT A
9520 REM CARACTERISTICAS HUMANAS
9530 LET J$(1 TO 2)=STR$ (INT (R
ND*11)+10)
9540 LET J$(3 TO 4)=STR$ (INT (R
ND*11)+10)
9550 LET J$(5 TO 6)=STR$ (INT (R
ND*11)+10)
9560 RETURN
9570 DATA "0002000000"
9580 DATA "0008030400"
9590 DATA "0000000200"
9600 DATA "0007020500"
9610 DATA "0006040000"
9620 DATA "0510070000"
9630 DATA "0400000600"
9640 DATA "0200090000"
9650 DATA "0011000800"
9660 DATA "0600000000"
9670 DATA "0900000000"
9680 DATA "0000000000"
9890 REM PAUSA RUTINA
9900 POKE 23692,-1: IF RND>.5 TH
EN GO TO 9950
9905 FOR I=RND*10 TO RND*30+10 S
TEP .5
9910 BEEP 0.03,I
9920 NEXT I
9930 PRINT
9940 RETURN
9950 FOR I=1 TO 50
9960 IF RND>.7 THEN BORDER RND*7
```

```
9970 NEXT I
9980 BORDER 1
9990 GO TO 9930
```

Ahora que ha sobrevivido (o no) en el CASTILLO ENCANTADO I, puede complicar un poco el plano e intentarlo de nuevo en el CASTILLO ENCANTADO II. Para desgracia de los ecologistas, el Jardín (que antes estaba situado en el centro del Castillo) ha sido colocado en el exterior. Al menos se puede ver a través de algunas ventanas, lo que le ayudará en sus averiguaciones.

Primero puede cambiar las instrucciones DATA de la forma siguiente:

```
9570 DATA "0000020000"
9580 DATA "0407000000"
9590 DATA "0000000700"
9600 DATA "0002050000"
9610 DATA "0006000400"
9620 DATA "0500100900"
9630 DATA "0200030000"
9640 DATA "0911000003"
9650 DATA "0008060800"
9660 DATA "0012000600"
9670 DATA "0800120000"
9680 DATA "0000000000"
```

También será necesario cambiar un poco las descripciones de las salas. Para que resulte más fácil la modificación, las palabras nuevas se han colocado en mayúsculas. Nótese que en la línea 8Ø3Ø no es necesario efectuar ningún cambio.

```
8000 REM CONTENIDO HABITACIONES
8010 IF Z=1 THEN PRINT "Te encue
ntras en la entrada de","un anti
guo y prohibido castillo","Estas
 en la fachada OESTE","mirando a
l ESTE la desmoronada","estructu
ra.",,"Te das cuenta de que el",
"portalon de entrada esta","abie
rto y sin guarda"
8020 IF Z=2 THEN PRINT "Te encue
ntras en la sala de","entrada qu
e esta decorada con","ricas tela
s. Las puertas estan","en direcc
ion NORTE y Sur": REM BORRE EL R
ESTO DE LA LINEA ORIGINAL
8030 IF Z=3 THEN PRINT "Esto es
solamente un almacen.","Solo tie
ne una salida, la que","has usad
o para entrar, al oeste"
8040 IF Z=4 THEN PRINT "Estas en
 una sala pequena con","una vell
a escultura a la Diosa","de la L
```

```
una en un pedestal en la","esqui
na Nordeste. Es la Camara","Real
. Las puertas estan al Sur","EST
E Y POR LA VENTANA SE VE EL","JA
RDIN"
8050 IF Z=5 THEN PRINT "La Sala
de las Conspiraciones,","una hab
itacion forrada en","madera, lle
na de rumores i","susurros, con
salidas al OESTE","y al Sur, de
la que llega un","olor a azufre
y un ambiente","sobrenatural. LA
 VENTANA DE LA","PARED NORTE PER
MITE VER EL","JARDIN"
8060 IF Z=6 THEN PRINT "Has entr
ado en la Guarida de","Brujos, c
on un cazo hirbiendo","en un fue
go de llamas verdes en","la esqu
ina Sudeste.","Esta sala con hed
or de azufre","y el eco de viejo
s sortilegios","la puedes abando
nar hacia el","Norte, OESTE o Es
te"
8070 IF Z=7 THEN PRINT "Te encue
ntras en un lugar que","parece t
ranquilo y apacible.","Es la Gal
eria de Pintura del","Castillo,
con un gran cuadro","del legenda
rio Guardian de La","Laguna Negr
a a la izquierda de","la PUERTA
de la pared NORTE.","Las salidas
 son hacia el Norte","y hacia el
 ESTE"
8080 IF Z=8 THEN PRINT "Esta es
una sala respandeciente","del ca
stillo: La Gran Sala.","Con un t
echo ricamente labrado.","Puede
salir hacia el SUR y","hacia el
NORTE por unas puertas","dobles
de donde procede una","extrana m
usica."
8090 IF Z=9 THEN PRINT "La music
a de un cuarteto de","cuerda lle
na la sala. Puede","abandonar la
 sala por unas","puertas situada
s al ESTE o por","una hacia el S
ur"
8095 REM NOTESE QUE SE SUPRIMEN
          LOS IF-THEN DE LAS
            SIGUIENTES DOS
              LINEAS
8100 IF Z=10 THEN PRINT "Estas a
hora en el Santuario del","Silen
cio. Una sala fria y","humeda. T
iene una salida hacia","el OESTE
 y una hacia el Sur"
8110 IF Z=11 THEN PRINT "Este de
be ser el Vestibulo de","las Apa
riciones, una oscura","sala, don
```

```
de la leyenda dice que","algunas
 noches se puede oir al","Guardi
an de la Laguna Negra.","Hay una
 puerta hacia el Norte","y una a
l ESTE"
8120 PRINT : IF Z=12 THEN GO TO
4000
8125 IF z<>1 THEN IF B$(Z,9)="0"
 AND RND>.55 THEN LET B$(Z,9)=ST
R$ (INT (RND*5)): REM BORRE PARA
 TENER MENOS MONSTRUOS
8130 IF B$(Z,9)<>"0" THEN LET Q=
VAL (B$(Z,9)): GO SUB 2000
8140 RETURN
8150 REM ************
```

No hay ninguna razón para no seguir trabajando sobre este programa, desarrollando tantas versiones del mismo como desee. Puede empezar con los mismos nombres de las salas y con el mismo número de ellas (12), rehaciendo el mapa mediante las instrucciones DATA y PRINT, que ya han sido modificadas en la última versión. Después puede tirar el mapa y tratar de olvidar sus detalles, creando por sí mismo un nuevo castillo. Puede incluso intercambiar «castillos» con sus amigos, a fin de conseguir un juego cada vez más sofisticado.

A pesar de que su programa debe ser representable en un mapa (sino no sería necesaria ninguna lógica, sería suficiente generar instrucciones PRINT de forma aleatoria cada vez que pulsara ENTER), puede hacer las interconexiones entre las habitaciones tan diabólicas como las pueda imaginar en dos dimensiones. También es fácil añadir más monstruos, si desea llenar más las estancias del castillo.

Una vez bien conocidas las posibilidades del diseño actual del castillo, puede doblar el número de habitaciones, incrementando el vector B$ y añadiendo las instrucciones DATA y PRINT, que harán coherente el conjunto. No recomendamos abordar esta modificación sin conocer perfectamente el programa original y haber realizado algunas variaciones sobre él con 12 habitaciones. Los monstruos están almacenados en el vector M$.

Quizá los cambios más fáciles a realizar sobre el programa original son los de los contenidos de los cofres, los efectos de beber la poción mágica y el resultado de leer, o de intentar leer, el «rollo de papiro». En su versión original el juego presenta una sola vez «la pequeña caja dorada», la poción y el «rollo de papiro», y cuatro veces los cofres numerados. Sin embargo, el contenido de estos elementos puede variar cada vez que aparecen y cierto contenido puede «saltar» de uno a otro cofre. (Cuando haya sido mordido en el cuello por cuatro vampiros, reduciéndose cada vez su fuerza, quizá deseará evitar que se sigan reproduciendo en todos los cofres del castillo.)

Seguramente pensará que es una pena disponer de un ordenador con la capacidad gráfica del Spectrum y hacer un juego con una salida en forma escrita, aunque subrayada mediante instrucciones del tipo FLASH, INVERSE y BRIGHT. Sin duda, la mayoría de los jugadores preferirán disponer de las escenas representadas en forma de dibujo. No obstante, el listado del programa resulta ya muy largo y, si lo fuera mucho más, sería muy pesada su introducción en el ordenador. Los programas que crean cuadros detallados son una larga serie de instrucciones DRAW y PLOT, siendo necesario editar un gran número de líneas para conseguir una figura interesante. Sobre el programa propuesto, usted puede ir sustituyendo las instrucciones PRINT de descripción de las distintas salas por series de PLOT y DRAW. Si se siente con ánimos de desarrollar una variante gráfica de este programa, hágalo y podría ser incluida en ediciones posteriores de este libro. Para ello háganos llegar sus programas.

Si no desea llegar tan lejos, puede dar a las descripciones escritas de las habitaciones distintos colores y diferenciar las descripciones de los monstruos con una melodía distintiva. En las líneas 2Ø1Ø a 2Ø5Ø puede insertar fácilmente informaciones de color y música mediante instrucciones del tipo «IF Q = ... THEN». Otra alternativa al dibujo de cada habitación y de cada monstruo es definir un bloque de gráficos de usuario para cada habitación y cada monstruo y construir con este bloque un borde para cada uno. También se puede llenar la pantalla con estos bloques, antes de que aparezcan las descripciones escritas.

Otros escenarios

Si no logra entender perfectamente el programa o prefiere iniciar los cambios antes de dibujar minuciosamente el mapa del CASTILLO ENCANTADO, puede modificar sustancialmente el programa sin variar su lógica. Si mantiene las salidas y las entradas en sus mismas posiciones, puede cambiar los nombres y las descripciones de las salas. Por ejemplo, podría describir toda la historia a bordo de una nave abandonada en el espacio en una órbita que desciende lentamente hacia un asteroide habitado por una extraña raza de extraterrestres. En lugar (o además de) los monstruos, puede disponer extraños y peligrosos artefactos, como androides que patrullan la nave, guardando una raza que ha partido muchos siglos antes. El objetivo de su búsqueda puede ser el motor de propulsión que resultará ser de una tecnología radicalmente avanzada. Una vez encuentre la sala de motores (que resultaría ser La Laguna Negra del Castillo Encantado), encontrará una fuerte defensa controlada por el Spectrum, que deberá vencer.

En lugar de hablar de MAGIA, FUERZA y SABIDURIA, sus atributos podrían ser oxígeno, raciones de comida y cartuchos para su pistola de rayos laser. No sólo puede canviar los atributos del programa original, dándoles distintos nombres, sino que puede añadir más (tales como habilidad o nervio). Al jugador se le puede permitir que distribuya sus atributos como prefiera, partiendo de un total de, por ejemplo, 9Ø puntos de atributo, en lugar de hacer esta distribución mediante un dado al principio del programa (o mediante un generador de números aleatorios, que es prácticamente lo mismo).

Otro método para distribuir los atributos al principio del juego es permitir al jugador escoger el «tipo» de persona que quiere ser (sacerdote, mago, ogro, luchador, ladrón o cualquier otra cosa). Con cada identidad queda asociado un grupo de atributos.

Se puede utilizar la estructura original del programa para contener una pirámide que usted puede explorar antes de que sea destruida para dejar espacio a una nueva presa en el Nilo. Puede reseguir los túneles, explorando las secretas cámaras de la pirámide, hasta llegar a la cámara mortuoria (La Laguna Negra del Castillo Encantado), donde debe superar una gran batalla para obtener su botín.

Con estas ideas puede ver que es fácil modificar la estructura de un programa de aventuras para obtener un nuevo juego que puede no tener ninguna relación con el original. Después de algunas experiencias modificando estos programas, seguramente se animará a elaborar uno propio. Puede tomar muchas ideas de programas comerciales para Spectrum o para otros ordenadores. Algunos de ellos, quizá sorprendentemente, no están basados en «realidades coherentes», sino que dependen de números aleatorios, sin presentar un universo que pueda concretarse en un mapa.

Es posible experimentar entretejiendo distintos programas, añadiendo un laberinto de tres dimensiones en un punto del recorrido o un examen de reflejos para evaluar su aptitud para el combate.

Finalmente recuerde que, al escribir un programa de aventuras, es esencial enunciar un objetivo claro (recolectar tesoros o matar dragones resulta aburrido, a menos que esto pase en su travesía hacia un gran problema: obtener algún final bien supremo o escapar vivo de la cámara de torturas). Además del objetivo, las distintas actividades de la aventura deben estar relacionadas con él.

Como ejemplo, en el Castillo Encantado intenta obtener muchos puntos de atributo o mucho dinero para comprarlos, para tener ventaja en la batalla final y disponer de diez puntos de reserva después de la misma. Por lo tanto, a cada encuentro (tal como abrir un cofre para intentar obtener dinero o luchar con un monstruo para intentar incrementar la puntuación de cierto atributo) se produce un suceso relacionado con el propósito principal del juego. Por supuesto, ya es bastante divertido intentar obtener el mapa del universo del programa y vagar

por el mismo observándolo, pero conviene además fijar «por qué» se está realizando tal exploración en un ambiente tan extraño.

Si está interesado en seguir aprendiendo sobre juegos de participación para implementar sus propias ideas, encontrará interesantes los siguientes tres libros (El Castillo Encantado está inspirado en ciertas ideas del primero de ellos):

Fantasy Role Playing Games Holmes, Eric J. (Arms y Armour Press, 1981)

Dicing With Dragons, an Introduction to Role-Playing Games Livingstone, Ian (Routledge & Kegan Paul, 1982)

*What is Dungeons and Dragons** — Butterfield, John, and Honigmann, D. (Penguin Books Ltd., 1982) (*Dungeons and Dragons is a registered trademark registered in the U.K. in the name of T.S.R. Hobbies, Inc., of Wisconsin, U.S.A., who also hold American rights to the name).

Juan Capitalista (Puesto de Limonadas)

Usted es Juan. Su misión: instalar un puesto de limonadas y hacer una fortuna (o, al menos, no llegar a la bancarrota). El programa le suministra informaciones para que pueda ir tomando las decisiones que en cada momento aconseje la situación del mercado. Se ha hecho un uso particularmente efectivo de los gráficos en color y alta resolución del Spectrum para enriquecer el programa. Incluso aparece un retrato suyo en el papel de Juan, detrás del mostrador del puesto de limonadas. No juzgue estos gráficos por las muestras impresas en el libro; lo que aparece en la pantalla es mucho más impresionante.

Una vez el programa en marcha, le indica que usted es el propietario de un puesto de limonadas. Cada día se le informa del boletín meteorológico. Un día cálido reportará muchas ventas; uno nublado no promete una gran recaudación. A partir de la predicción del tiempo y de su conocimiento del mercado, que irá adquiriendo a lo largo de sucesivos días, debe decidir cuántos vasos de limonada fabricará. También deberá fijar el precio de venta de los vasos en función de su coste y de lo que quiera ganar en cada venta. Por supuesto, cuanto mayores sean sus precios menores serán sus ventas. Al principio del juego, el alquiler de su puesto le cuesta 5Ø ptas. A medida que pasa el tiempo, y debido a la inflación, el alquiler aumenta. La inflación, además, erosiona sus beneficios, ya que hace aumentar el precio de las materias primas.

El ejemplo impreso muestra el tipo de información con la que trabajará cada día: el pronóstico del tiempo, el coste de las materias primas (Ø,5 Ptas. por cada vaso de limonada), el alquiler de 5Ø Ptas. y

el capital disponible de 2ØØØ Ptas. A partir de esta información, debe decidir cuántos vasos de limonada fabricará. El programa hace aparecer las copas fabricadas y dibuja el puesto de limonadas, donde aparece usted, las copas y el precio de venta.

El tiempo pasa rápidamente y las limonadas se van vendiendo (es decir, vaciando). Al final del día baja la persiana y el programa le informa de los resultados de la jornada: «Has vendido 2Ø copas; has ganado 2Ø Ptas; habías fabricado 20 y te habían costado 1Ø Ptas.; el alquiler te cuesta 5Ø Ptas.; tienes una pérdida de 4Ø Ptas».

Las instrucciones DATA de las líneas 15 y 2Ø representan los vasos vacíos (e) y los llenos (f) respectivamente. La próxima instrucción DATA contiene la previsión del tiempo y los números que gobiernan el efecto que la meteorología causará a sus ventas. La tinta se hace negra, la pantalla se pone en su brillo normal y se borra en la línea 45. Las líneas 5Ø y 6Ø definen los gráficos. La línea 5Ø lee los datos de las líneas 15 y 2Ø y la 6Ø los almacena. Después de un leve BEEP en la línea 7Ø, en la 8Ø se define una función que sirve para determinar cuántos vasos de limonada se venden hoy en la línea 84Ø.

Las líneas 1ØØ a la 16Ø imprimen las instrucciones y, después de una pausa (PAUSE 1ØØ en la línea 16Ø), aparece la frase «Buena suerte!». En 17Ø se produce la familiar instrucción «Pulsa una tecla para empezar» de forma intermitente. PAUSE Ø mantiene la pantalla fija hasta que se pulsa efectivamente una tecla (en el ZX81 la instrucción equivalente es PAUSE 4E4 o PAUSE 4ØØØØ) y se borra la pantalla.

En 18Ø se inicializan las variables. Su capital inicial (cash) se inicializa a 2ØØØ; el tiempo (weather) se iguala a 4; las materias primas (limonada) a Ø.5 y el alquiler (rent) a 5Ø. Las líneas 2ØØ a 22Ø escogen el pronóstico del tiempo y la 23Ø empieza a imprimir la información en la pantalla. La línea 25Ø recoloca (RESTORE) el puntero de los datos en la línea 25 que almacena el tiempo y sus variables de venta. Las variables (que se nombran from y to) se usan en la línea 84Ø para determinar cuántos vasos se han vendido.

Se le informa del estado de sus fianzas en la línea 29Ø y se le pregunta cuántos vasos de limonada quiere fabricar. Si sugiere un número negativo, la línea 3Ø2 rechaza su respuesta y la 3ØØ le pregunta de nuevo. La línea 3Ø5 le envía a la 9ØØØ, donde se comprueba si dispone de suficiente capital para afrontar la fabricación. Si es así, la subrutina le retorna instantáneamente. Se llega a la línea 9Ø1Ø, si usted ha intentado fabricar más copas de las que puede hacer con su capital disponible («No tienes suficiente dinero para hacer tantas»). Una vez consciente de sus limitados recursos económicos, se le vuelve a preguntar cuántas copas quiere fabricar. Se vuelve a comprobar su entrada en la línea 9ØØØ y, si es aceptable, puede volver a la rutina principal. Si no lo es, el programa vuelve a caer en la línea 9Ø1Ø y el

proceso se repite. (Ya que estamos analizando el final del programa, veamos la línea 9999. Esta línea no forma parte del programa en sí mismo, sino que se escribió durante su desarrollo como instrumento para el mismo. Sirve para listar el programa en cualquier momento. Devuelve la situación de la pantalla a su posición normal y la borra, listando a continuación el programa. La instrucción RETURN al final significa que esta línea puede ser llamada por el propio programa como una subrutina más. Cuando desarrolla un programa complejo, resulta interesante incluir una línea como ésta.)

A la vuelta de la subrutina de la línea 9ØØØ, la línea 3Ø7 determina si el número de copas a fabricar es menor que 1. Si es así, el número de copas vendidas será cero y el programa salta a la línea 95Ø, evitando la parte de venta porque no hay nada para vender.

Si ha pedido un número de copas coherente y se toma el negocio seriamente, el programa imprime las copas fabricadas mediante las instrucciones de las líneas 31Ø y 32Ø. La línea 33Ø le pregunta el precio de venta para hoy y la 333 rechaza cualquier precio mayor de 5 pesetas o menor de Ø.1 Ptas. Después de un corto BEEP (335), se imprime el precio de venta (34Ø) y se le pide que pulse cualquier tecla para empezar a vender (35Ø). Otro BEEP y el programa espera (PAUSE Ø) hasta que pulse una tecla. Su pulsación se contesta con otro BEEP (al final de la misma línea) y la línea 36Ø borra la pantalla, coloca la variable brillo y llena de un blanco brillante una parte de la pantalla con el bucle «i».

Ahora viene la sección más interesante desde un punto de vista visual. La rutina de las líneas 37Ø a 69Ø crea la figura de Juan y de su puesto con las estanterías para colocar las limonadas. La siguiente sección, usando la variable «count» (que se iguala a la variable «cups» en la línea 695), imprime los vasos llenos de limonada. Cada vez que se imprime una copa, la variable count se decrementa en 715 y se comprueba si ha llegado a cero. Cuando llega, el ordenador sabe que se han representado todas las copas. Una vez las copas en sus estanterías para atraer a los sedientos clientes, las líneas 73Ø a 83Ø completan el cuadro del puesto de limonadas.

La línea 84Ø activa la función definida en la línea 8Ø para determinar el número de vasos que ha conseguido vender hoy. Los bucles existentes después de la línea 86Ø cambian los vasos llenos por vacíos en las estanterías. Si se han vendido todos, se coloca el letrero de «TODO VENDIDO» (línea 9ØØ). Después de un par de segundos de pausa al principio de la línea 91Ø, baja la persiana del puesto de limonadas. Otra pausa al principio de la línea 94Ø y luego el ordenador imprime el «Informe de Ventas» del día: «Has vendido ... vasos. Has ganado ...».

La línea 1ØØØ calcula los beneficios o pérdidas (variable «prof»). Las líneas 1Ø1Ø a 1Ø3Ø utilizan la función SGN para determinar si la

jornada ha resultado un éxito o un fracaso. (La función SGN da −1, si el número es negativo; o + 1, si es positivo; o Ø si el número es cero.)

La inflación amenaza sus beneficios en la línea 1Ø5Ø, incrementando su alquiler el 10 % de los días. Igualmente en la línea 1Ø6Ø el generador de números aleatorios incrementa el coste de las materias primas el 1Ø % de los días. La línea 1Ø65 espera a que usted pulse una tecla para continuar. Si 1Ø67 descubre que no le queda dinero (es decir, si cash es menor que 1), el programa va a 15ØØ, anunciándole la implacable bancarrota. Si, por el contrario, usted todavía es un negociante solamente, el programa vuelve a la línea 2ØØ para permitirle otra jornada de negocio y habiendo actualizado su capital.

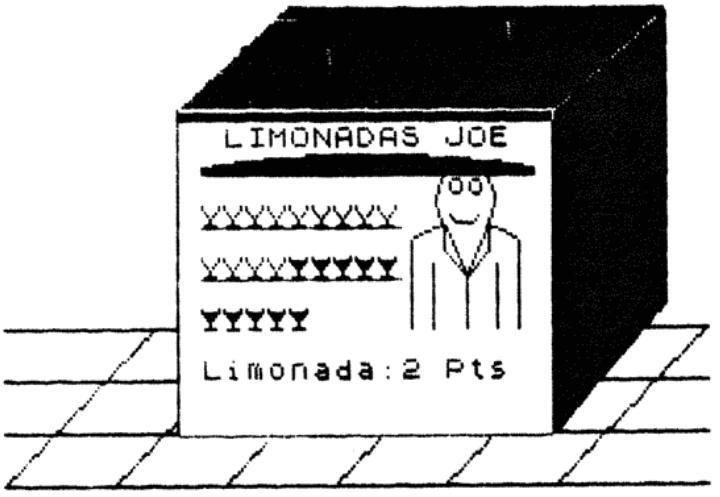
En el caso de caer en la línea 15ØØ, aparece el mensaje «CERRADO», Usted se encuentra en bancarrota y aparece el mensaje: «Estás en bancarrota y debes vender tu puesto para pagar las deudas». La línea 152Ø salta siempre sobre sí misma, terminando el programa sin que aparezca ninguna otra información. Para salir de esta situación puede pulsar BREAK.

```
PUESTO DE LIMONADAS

  Eres el propietario de un
puesto de bebidas donde se
venden limonadas.

  Cada dia se te comunica el
pronostico del tiempo, a partir
del cual, decidiras cuantas
copas haras. Tambien seras
informado de cuanto te cuesta
cada copa y podras decidir el
precio de venta.

El alquiler del puesto te cuesta
5Ø Ptas. al dia, pero aumentara
con la inflacion.
```



PUESTO DE LIMONADAS

Informe de Hoy

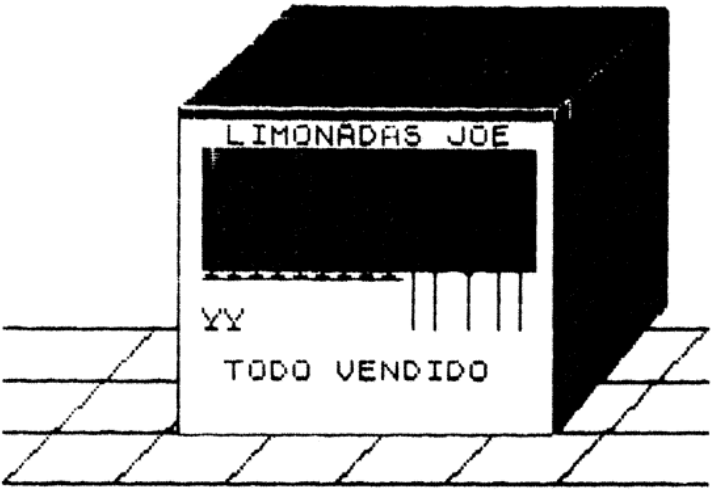
Tiempo : Templado

Cada copa te cuesta 0.5 pesetas

Alquiler: 50 Ptas.

Tienes 2000 Ptas. disponibles

Copas: YYYYYYYYYYYYYYYYYYYY



PUESTO DE LIMONADAS

Informe de Ventas

Has vendido 20 copas

Has ganado 40 Ptas.

Habias fabricado 20 y te habian costado 10 Ptas.

El alquiler te cuesta 50 Ptas.

Tienes una perdida de 20 Ptas.

Graficos usados:

Y ... Grafico 'f'

Y ... Grafico 'e'

```
  12 POKE 23609,128
  15 DATA "e",0,BIN 10000010,BIN
 10000010,BIN 01000100,BIN 00101
000,BIN 00010000,BIN 00010000,BI
N 01111100
  20 DATA "f",0,BIN 10000010,BIN
 11111110,BIN 01111100,BIN 00111
000,BIN 00010000,BIN 00010000,BI
N 01111100
  25 DATA "Tormentoso",10,30,"Hu
medo",25,40,"Nublado",35,50,"Tem
plado",45,70,"Caluroso",65,120,"
Ardiente",100,180,"Ola de calor"
,200,300
  45 INK 0: BRIGHT 0: CLS
  50 FOR J=1 TO 2: READ a$: FOR
i=0 TO 7
  60 READ a: POKE USR a$+i,a: NE
XT i: NEXT j
  70 BEEP .5,0
  80 DEF FN p(p,q)=q/(p/100)-100
 100 PRINT PAPER 6;"          PUEST
O DE LIMONADAS          "'"  Eres e
l propietario de un      "'"puest
o de bebidas donde se        "'"ve
nden limonadas.                "'
 110 PRINT "  Cada dia se te com
unica el","pronostico del tiempo
, a partir"
 120 PRINT "del cual, decidiras
cuantas"
 130 PRINT "copas haras. Tambien
 seras","informado de cuanto te
cuesta","cada copa y podras deci
dir el","precio de venta."
 140 PRINT '"El alquiler del pue
sto te cuesta"
 150 PRINT "50 Ptas. al dia, per
o aumentara","con la inflacion."
 160 PAUSE 100: PRINT INK 2;'''
        Buena Suerte !"
 170 PRINT FLASH 1;'"  Pulsa una
 tecla para empezar  ": PAUSE 0:
 CLS
 180 LET cash=2000: LET weather=
4: LET lemonade=.5: LET rent=50
 200 LET weather=weather+INT (RN
D*3-1)
 210 IF weather=0 THEN LET weath
er=1
 220 IF weather=8 THEN LET weath
er=7
 230 PRINT PAPER 6;"          PUEST
O DE LIMONADAS
 240 PRINT INK 7; PAPER 1;'"
     Informe de Hoy
 250 RESTORE 25: FOR i=1 TO weat
her: READ w$,from,to: NEXT i
```

```
 260 PRINT '"Tiempo : "; INK 1;w
$
 270 PRINT '"Cada copa te cuesta
 "; INK 1;lemonade;" pesetas"
 280 PRINT '"Alquiler: "; INK 1;
rent;" Ptas."
 290 PRINT '"Tienes "; INK 1;cas
h; INK 0;" Ptas. disponibles"
 300 BEEP .5,10: INPUT INK 2;"Cu
antas copas haras "; FLASH 1;"?
 ";cups
 302 IF cups<0 THEN GO TO 300
 305 GO SUB 9000: BEEP .5,5
 307 IF cups<1 THEN LET sold=0:
LET price=.1: LET cups=0: GO TO
945
 310 INK 6: PAPER 0: PRINT '"Cop
as: ";
 320 FOR i=1 TO cups: PRINT "Y";
: NEXT i: PAPER 7: PRINT '
 330 BEEP .5,10: INPUT INK 2;"Cu
al es su precio de venta para"'"
hoy "; FLASH 1;"?";price
 333 IF price>5 OR price<.1 THEN
 GO TO 330
 335 BEEP .1,5
 340 INK 0: PAPER 7: PRINT '"El
precio de cada copa es ",price;"
 Ptas."
 350 PRINT FLASH 1; INK 2;'"Puls
a una tecla para empezar"'"la ve
nta": BEEP .1,10: PAUSE 0: BEEP
.5,5
 360 CLS : BRIGHT 1: FOR i=18 TO
 21: PRINT AT i,0;"
                    ": NEXT i
 370 FOR i=0 TO 48 STEP 16
 380 PLOT 0,i: DRAW 255,0
 390 NEXT i
 400 FOR i=0 TO 200 STEP 40
 410 PLOT i,0: DRAW 55,48
 420 NEXT i
 430 PRINT PAPER 6;AT 8,8;"  LIM
ONADAS JOE  "; PAPER 6;AT 9,8;"
                "
 440 FOR i=92 TO 98
 450 PLOT INK 7;72,i: DRAW INK 7
;120,0,-PI/12
 460 NEXT i
 470 FOR i=1 TO 6: PRINT PAPER 6
;AT 9+i,8;" ";TAB 24; PAPER 6;"
 "
 480 NEXT i
 490 FOR i=1 TO 4: PRINT PAPER 6
;AT 15+i,8;"                    ":
NEXT i
 495 BRIGHT 0
 496 INK 6
 500 FOR i=112 TO 115
```

```
 510 PLOT 64,i: DRAW 136,0
 520 NEXT i
 530 FOR i=64 TO 199
 540 PLOT i,116: DRAW 40,31
 550 NEXT i
 560 FOR i=115 TO 24 STEP -1
 570 PLOT BRIGHT 0;200,i: DRAW B
RIGHT 0;40,31
 580 NEXT i
 590 FOR i=16 TO 23
 600 PLOT BRIGHT 0,200,i: DRAW B
RIGHT 0,40,40
 610 NEXT i
 615 INVERSE 1
 620 PLOT PAPER 0;64,116: DRAW P
APER 0;134,0
 630 PLOT PAPER 0;64,112: DRAW P
APER 0;134,0
 640 INK 0: INVERSE 0
 650 PLOT 63,16: DRAW 0,100
 660 PLOT 199,16: DRAW 0,96: DRA
W INVERSE 1; INK 6; PAPER 0,0,4
 670 PLOT 64,15: DRAW 134,0
 680 INK 2: PLOT 72,79: DRAW 72,
0
 690 PLOT 72,63: DRAW 72,0
 695 LET count=cups: GO SUB 700:
 GO TO 730
 700 IF count=0 THEN RETURN
 705 FOR i=11 TO 15 STEP 2: FOR
j=1 TO 9
 710 BRIGHT 8: PRINT AT i,8+j;"▼
": BEEP .1,20
 715 LET count=count-1: IF count
=0 THEN RETURN
 720 NEXT j: NEXT i: RETURN
 730 PLOT 160,80: DRAW 15,0,-PI*
3/2
 740 PLOT 148,48: DRAW 0,27: DRA
W 12,5,-PI/6
 750 PLOT 187,48: DRAW 0,27: DRA
W -12,5,PI/6
 760 PLOT 156,48: DRAW 0,20
 770 PLOT 179,48: DRAW 0,20
 780 PLOT 168,48: DRAW 0,16: DRA
W 8,8: DRAW 0,8: DRAW -8,-16
 790 DRAW -8,8: DRAW 0,8: DRAW 8
,-16
 800 PLOT 162,83: DRAW 10,0,PI/2
 810 CIRCLE 164,92,2: CIRCLE 171
,92,2
 830 PRINT AT 17,9; INK 0; PAPER
 6;"Limonada:";price;" Pts"
 840 LET sold=(INT (RND*(to-from
)+from)+(to-from))*80/FN p(lemon
ade,price)
 845 LET sold=INT sold: IF sold>
cups THEN LET sold=cups
 850 LET sellc=sold
```

```
 860 FOR i=11 TO 15 STEP 2: FOR
j=9 TO 17
 870 PRINT AT i,j;"Y": BEEP .1,1
0
 880 LET sellc=sellc-1: IF sellc
=0 THEN GO TO 900
 890 NEXT j: NEXT i: GO SUB 700:
 GO TO 860
 900 IF sold=cups THEN PRINT INK
 0; PAPER 4;AT 17,9;" TODO VENDI
DO "
 910 PAUSE 100: FOR i=103 TO 48
STEP -1
 920 BEEP .02,i-48: PLOT 72,i: D
RAW 120,0
 930 NEXT i
 940 PAUSE 100: BEEP .5,5
 950 CLS : PAPER 7: INK 0: PRINT
 PAPER 6;"          "; BRIGHT 1;"PUE
STO DE LIMONADAS"; BRIGHT 0;"
    "
 960 PRINT INK 7; PAPER 1;'"
   Informe de Ventas          "
 970 PRINT '"Has vendido "; INK
1;sold;" copas"
 980 PRINT '"Has ganado "; INK 1
;sold*price;" Ptas."
 990 PRINT '"Habias fabricado ";
cups;" y te","habian costado ";
INK 1;cups*lemonade;" Ptas."
 995 PRINT '"El alquiler te cues
ta ";rent;" Ptas."
1000 LET prof=sold*price-cups*le
monade-rent
1010 IF SGN prof=-1 THEN PRINT '
"Tienes una "; INK 2;"perdida";
INK 0;" de ";-prof;" Ptas."
1020 IF SGN prof=0 THEN PRINT '"
Te has arruinado"
1030 IF SGN prof=1 THEN PRINT '"
Tienes unas "; INK 4;"ganancias"
; INK 0;" de ";prof;" Ptas."
1040 LET cash=cash+prof
1050 IF RND>.9 THEN LET rent=ren
t+INT (RND*10+1)
1060 IF RND>.9 THEN LET lemonade
=lemonade+1
1065 BEEP .5,10: PAUSE 0: BEEP .
5,5
1067 IF cash<1 THEN GO TO 1500
1070 CLS : GO TO 200
1500 CLS : PRINT INK 1; FLASH 1;
"   CERRADO   "
1510 PRINT '"Estas en bancarrota
 y debes","vender tu puesto para
 pagar las","deudas"
1520 GO TO 1520
9999 STOP
```

```
9000 IF cups*lemonade<=cash THEN
 RETURN
9010 INPUT INK 1;"No tienes sufi
ciente dinero para";"hacer tanta
s"' INK 2;"Cuantas copas quieres
 hacer "; FLASH 1;"?";cups
9020 GO TO 9000
9990 STOP
9999 INVERSE 0: PAPER 7: BRIGHT
0: INK 0: BORDER 7: CLS : LIST :
 RETURN
```

Mejore su inteligencia

Sintaxis

Este es un juego «intelectual» para personas a las que les guste jugar con palabras. El objetivo del juego es obtener 1Ø frases que tengan sentido. Al ejecutar el programa el ordenador le pide unos momentos de espera y en este tiempo construye diez frases al azar, a partir de una serie de nombres, adjetivos, verbos y preposiciones. El resultado evidentemente son diez frases que no tienen ningún sentido. Usted debe ordenarlas.

Verá un cursor situado al lado derecho de la pantalla con el número de palabra a la que apunta. Puede correr el cursor arriba y abajo mediante las teclas «6» y «7». Pulsando cualquier tecla de «1» a «5» puede subrayar una palabra de la frase a la que el cursor apunta, es decir, si pulsa «3», queda subrayada la tercera palabra de la frase.

Dispone de un almacén de diez frases vacías al principio del juego. Para entrar palabras a este almacén debe subrayar una palabra de las frases aleatorias propuestas por el ordenador (mediante el método anteriormente expuesto) y pulsar ENTER. El ordenador le preguntará en qué frase debe añadir la nueva palabra e imprimirá las frases que usted está construyendo con la nueva palabra en su posición correcta. Si pretende entrar la palabra subrayada en la quinta frase de su almacén, pulse 5 cuando el computador le pregunte: «¿A qué frase?».

Podría ocurrir que, una vez hubiera conseguido construir sus diez frases, dudara de si son coherentes. El ordenador no podrá ayudarle a resolver sus dudas. El único juez de sus frases es usted mismo.

La línea 2Ø activa el generador de números aleatorios y la 3Ø envía la acción a la 1ØØØ, donde se dimensionan las matrices y se asignan valores a las variables. Las líneas 1Ø4Ø y 1Ø5Ø buscan elementos vacíos en la matriz a$, volviendo a 1Ø4Ø para examinar el próximo si el actual no está vacío. La línea 1Ø6Ø lee la próxima palabra de las líneas de datos (DATA) situadas después de la línea 9ØØØ. Un elemento de la matriz *m* se iguala a la longitud de la palabra leída y la palabra misma (b$) se almacena en un elemento de la matriz a$ (línea 1Ø7Ø).

En 11ØØ la variable x$ se iguala a «a$» (no es que se igualen las variables. La variable x$ se iguala a los caracteres «a» y «$»). La matriz l se iguala a la *m* y la acción pasa a la subrutina de la línea 2ØØ, donde se imprimen las frases en la pantalla. (Se utiliza una matriz «puente» l para escribir las frases, ya que se utiliza la misma rutina para escribir las construidas por el ordenador *m* y las del jugador *n*.)

La instrucción RETURN de la línea 26Ø devuelve la acción a la línea 116Ø, donde se inicializan las variables *u, p* y *n*. Se usan en la localización de palabras en la pantalla. La línea 119Ø imprime el cursor de líneas. Se llama inmediatamente la subrutina de la línea 1ØØ, encargada de imprimir el cursor de palabras (el cuadrado negro del final de la línea 15Ø). La sección que empieza en la línea 1195 acepta sus demandas de movimientos y actúa en consecuencia. La línea 12Ø5 iguala la variable i$ a la tecla que está pulsando. La línea 125Ø detecta si usted ha pulsado ENTER y, si es así, le envía a la línea 126Ø para preguntarle a qué frase quiere dirigir la palabra apuntada en la pantalla. De acuerdo con su respuesta, x$ se iguala a «s$» (recuerde: no s$) y la línea 13ØØ llama a la subrutina de la línea 2ØØ para imprimir sus frases. Las líneas 134Ø y 135Ø esperan a que no esté pulsando una tecla y luego a que pulse otra, antes de devolver la acción a la línea 11ØØ, donde el proceso sigue.

```
    5 REM Sintaxis
   10 RESTORE
   20 RANDOMIZE
   30 GO TO 1000
  100 LET e=0
  110 LET v=u/2+1
  120 FOR b=3*(n>3)+1 TO n-1
  130 LET e=e+l(b,v)+1
  140 NEXT b
  150 PRINT AT u+(n>3),e+3*(n<4);
  OVER 1;"■"
  160 RETURN
  200 CLS
  205 FOR a=1 TO 11
  207 PRINT "Un ";
  210 FOR b=1 TO 5
  220 PRINT VAL$ (x$+"(b,a, TO l(
b,a))");" ";
  230 IF b=3 THEN PRINT
  240 NEXT b
  245 PRINT
  250 NEXT a
  260 RETURN
 1000 PRINT AT 11,0;"Por favor,es
pere un momento..."
 1005 DIM s$(5,11,10)
 1010 DIM a$(5,11,10)
 1015 DIM l(5,11)
 1017 DIM m(5,11)
 1018 DIM n(5,11)
```

```
1020 FOR a=1 TO 5
1030 FOR b=1 TO 11
1040 LET c=INT (RND*11)+1
1050 IF a$(a,c,1)<>" " THEN GO T
O 1040
1060 READ b$
1065 LET m(a,c)=LEN b$
1070 LET a$(a,c)=b$
1080 NEXT b
1090 NEXT a
1100 LET x$="a$"
1105 FOR e=1 TO 5
1106 FOR f=1 TO 11
1107 LET l(e,f)=m(e,f)
1108 NEXT f: NEXT e
1110 GO SUB 200
1160 LET u=0
1170 LET p=30
1180 LET n=1
1190 PRINT FLASH 1;AT u,p;"<";ST
R$ n
1192 GO SUB 100
1195 IF INKEY$<>"" THEN GO TO 11
95
1200 IF INKEY$="" THEN GO TO 120
0
1205 LET i$=INKEY$
1210 PRINT AT u,p;"  "
1215 GO SUB 100
1220 BEEP .01,20
1230 LET u=u+((i$="6")*(u<20)-(i
$="7")*(u>0))*2
1240 IF CODE i$>48 AND CODE i$<5
4 THEN LET n=VAL i$
1250 IF CODE i$<>13 THEN GO TO 1
190
1260 INPUT "A que frase?",q
1270 IF q<1 OR q>11 THEN GO TO 1
190
1280 LET s$(n,q)=a$(n,u/2+1)
1281 LET long=0: LET test=0
1282 FOR e=1 TO 10
1284 IF s$(n,q,e)=" " AND test=1
 THEN GO TO 1289
1285 IF test=1 THEN LET long=lon
g+1: LET test=0
1286 IF s$(n,q,e)=" " THEN LET t
est=1: GO TO 1288
1287 LET long=long+1
1288 NEXT e
1289 LET n(n,q)=long
1290 LET x$="s$"
1295 FOR e=1 TO 5
1296 FOR f=1 TO 11
1297 LET l(e,f)=n(e,f)
1298 NEXT f: NEXT e
1300 GO SUB 200
1340 IF INKEY$<>"" THEN GO TO 13
40
```

```
1350 IF INKEY$="" THEN GO TO 135
0
1360 GO TO 1100
1370 GO SUB 200
9005 REM
9010 DATA "abedul","carnicero","
perro","pez","guijarro","robot",
"espectador","escorpion","maestr
o","paraguas","camarero"
9015 REM
9017 DATA "negro","rubio","azul"
,"mudo","furioso","malo","sucio"
,"palido","aburrido","pequeno","
alto"
9019 REM
9020 DATA "ladra","rompe","corta
","gotea","sirve","gira","oscila
","nada","pica","anda","observa"
9025 REM
9030 DATA "un","por el","la","al
","su","en un","en el","sobre el
","en la","el","hasta el"
9035 REM
9040 DATA "playa","gato","piso",
"juego","comida","parque","polit
ico","estanque","florero","red",
"viento"
9999 PRINT AT 19,0;A: PAUSE 0: R
ETURN
```

```
Un pez aburrido gira
el juego
Un abedul palido corta
por el piso
Un paraguas malo oscila
la parque
Un robot azul anda              <3
en la comida
Un carnicero pequeno observa
en un florero
Un guijarro negro pica
al gato
Un perro furioso ladra
su estanque
Un espectador rubio nada
sobre el playa
Un maestro alto sirve
un viento
Un camarero sucio rompe
en el red
Un escorpion mudo gotea
hasta el politico
```

El pozo

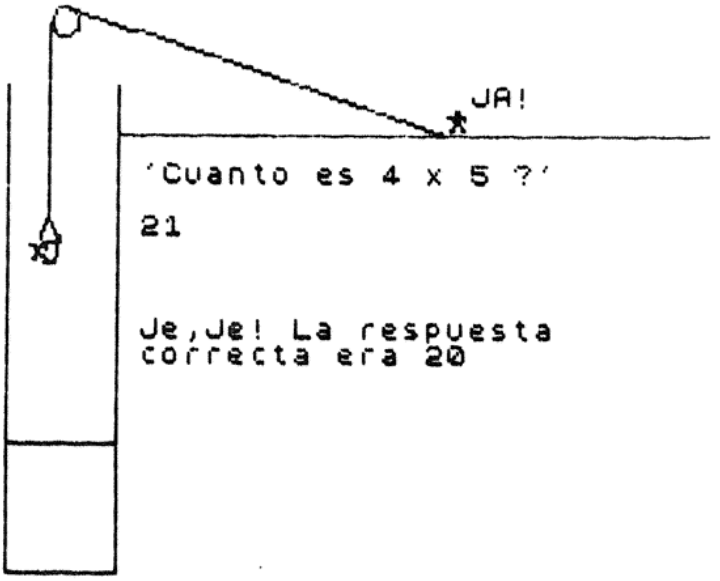
En este programa se muestra un cuidado escenario en el que usted se convierte en el desgraciado prisionero de un demente, quien decide que, debido a sus muchos pecados, debe ser encerrado en un cubo y colgado en un pozo. El loco le hará preguntas y si usted no logra demostrar su habilidad matemática, irá descendiendo por el pozo hasta ahogarse.

Así es como lo explica el propio programa:

```
En este juego usted es el
prisionero de un profe-
sor maniaco que le ha
colocado en un cubo
que desciende lentamente
hacia el agua.

El le hara preguntas.
Cuando se equivoque le
bajara.
Contestelas correctamen-
te y le izara.
```

Y ésta es la escena en la que se le hacen las preguntas:



Debido a su considerable grado de elaboración, el programa es muy directo y requiere un accionamiento mínimo por parte del jugador.

La rutina que empieza en la línea 1ØØØ dibuja la escena. Después de la introducción y de un primer dibujo, el programa le pregunta qué nivel tiene de matemáticas. El número es un tanto por ciento. Si entra

un valor inferior al 12 %, el ordenador no le cree y escoge otro tanto por ciento al azar. Le sugerimos que responda un 15 % la primera vez que ejecute el programa. La línea 71Ø iguala la variable de operación t$ a multiplicación (×) o adición (+). La 72Ø asigna a la variable «a» un número comprendido entre 1 y el número que ha entrado como nivel de conocimientos. Si t$ es un signo de multiplicación, en la línea 73Ø se asigna un número entre 1 y 1Ø a la variable «b». Si t$ contiene un signo de adición, la línea 735 se asigna un número entre 1 y 1ØØ veces el que ha entrado como nivel de conocimiento.

La línea 74Ø imprime la interrogación en la pantalla, sustituyendo el asterisco (*) por (×), si se trata de una operación de multiplicar (ya que el signo «×» es el que se usa normalmente como signo de multiplicar). El bucle g de la línea 777 a la 84Ø le da la oportunidad de entrar su respuesta. Resulta ineresante el análisis de esta rutina, ya que es muy útil en todo tipo de programas. En la línea 785 se aceptan los caracteres entrados por el jugador y en las líneas siguientes se almacenan formando un número. El profesor reacciona a su respuesta en las líneas 85Ø a 985 y se genera una nueva pregunta.

Observando la rutina situada hacia el final del programa que imprime la escena, puede ver la variable principal «l» que determina la profundidad del cubo. Si el profesor hace llegar el cubo al fondo, la partida termina trágicamente. Si va acertando las respuestas y el cubo llega a la polea del pozo, se produce un inesperado final, propio del loco profesor (de hecho, el desenlace que se produce es mecánicamente lógico).

```
   5 REM El Pozo
  10 LET l=100
  20 GO SUB 9000
 510 PRINT AT 6,6;"El pozo"
 520 PRINT ,,TAB 6;;"En este jue
go usted es el"
 530 PRINT TAB 6;"prisionero de
un profe-"
 540 PRINT TAB 6;"sor maniaco qu
e le ha"
 550 PRINT TAB 6;"colocado en un
 cubo"
 560 PRINT TAB 6;"que desciende
lentamente"
 570 PRINT TAB 6;"hacia el agua.
"
 580 PRINT ,,TAB 6;"El le hara p
reguntas."
 590 PRINT TAB 6;"Cuando se equi
voque le"
 600 PRINT TAB 6;"bajara."
 610 PRINT TAB 6;"Contestelas co
rrectamen-"
 620 PRINT TAB 6;"te y le izara.
"
```

```
 630 GO SUB 1005
 640 PRINT AT 21,6;"Pulse una te
cla"
 650 PAUSE 0
 660 GO SUB 1000
 670 PRINT AT 6,6;"El profesor d
ice:"
 680 INPUT "'Como va de matemati
cas? (%)'",d
 682 IF d<12 THEN LET d=INT (RND
*7)+4
 685 LET l=20+d
 690 IF d<0 OR d>100 THEN PRINT
AT 8,5;"'Entre un numero desde 0
-100": GO TO 680
 700 GO SUB 1000
 710 LET t$=CHR$ (42+INT (RND*2)
)
 720 LET a=INT (RND*d)+1
 730 IF t$="*" THEN LET b=INT (R
ND*10)+1: GO TO 740
 735 LET b=INT (RND*d*100)
 740 PRINT AT 6,6;"'Cuanto es ";
 750 LET u$=t$
 760 IF t$="*" THEN LET u$="x"
 765 LET c$=STR$ VAL (STR$ a+t$+
STR$ b)
 770 PRINT a;" ";u$;" ";b;" ?'"
 772 LET tt=0
 775 LET w$=""
 777 FOR g=1 TO LEN c$
 780 FOR t=1 TO l*100
 785 LET i$=INKEY$
 790 IF i$="" THEN NEXT t
 800 IF CODE i$<48 OR CODE i$>57
 THEN GO TO 910
 810 LET w$=w$+i$
 815 LET tt=tt+t
 820 IF w$(g)<>c$(g) THEN GO TO
910
 825 PRINT AT 8,6;w$: BEEP .1,20
 830 IF INKEY$<>"" THEN GO TO 83
0
 840 NEXT g
 850 PRINT AT 10,6;"Aah!! Has ac
ertado!"
 860 FOR a=1 TO tt/10
 870 BEEP .01,20
 880 NEXT a
 890 LET l=l-(l*10-tt/LEN c$)/10
0
 900 GO TO 700
 910 PRINT AT 8,6;w$
 915 FOR a=1 TO 20
 920 BEEP .1,0
 930 NEXT a
 940 PRINT AT 12,6;"Je,Je! La re
spuesta"
```

```
 950 PRINT AT 13,6;"correcta era
 ";c$
 952 FOR u=1 TO 3
 955 PAUSE 30
 960 PRINT AT 3,(200-l)/8+3;"JA!
"
 965 PAUSE 30
 970 PRINT AT 3,(200-l)/8+3;"
"
 975 NEXT u
 980 LET l=l+4
 985 GO TO 700
 990 STOP
1000 CLS
1005 PLOT 0,150
1010 DRAW 0,-150
1020 DRAW 40,0
1030 DRAW 0,150
1032 PLOT 40,135: DRAW 215,0
1035 PLOT 1,40: DRAW 39,0
1040 CIRCLE 20,170,5
1050 PLOT 20,175
1060 DRAW 200-l,-40
1065 PRINT AT 4,(200-l)/8+2; OVE
R 1;"*"
1070 PLOT 15,170
1080 DRAW 0,-l
1090 DRAW -4,-8
1100 DRAW 8,0
1110 DRAW -4,8
1120 PLOT 11,162-l
1130 DRAW 2,-6
1140 DRAW 4,0
1150 DRAW 2,8
1155 PRINT AT l/8+1,1; OVER 1;"*
"
1160 IF l=0 THEN BEEP 1,22: GO T
O 2000
1165 IF l=153 THEN BEEP 1,22: GO
 TO 3000
1170 IF l<20 THEN LET l=0: BEEP
1,20: GO TO 1000
1175 IF l>122 THEN LET l=153: BE
EP 1,20: GO TO 1000
1180 RETURN
2000 REM Caida
2010 FOR a=2 TO 16
2020 PRINT AT a-1,1;" "
2030 PRINT AT a,1;"*"
2035 BEEP .05,a
2040 NEXT a
2050 FOR a=17 TO 21
2055 PRINT AT a,1;"*"
2060 PRINT AT a-1,1;" "
2070 BEEP .2,21-a
2080 NEXT a
2090 PRINT AT 21,1; OVER 1;"_"
2100 PRINT AT 16,1;"_"
2110 STOP
```

```
3000 STOP
9000 DATA 24,24,254,58,24,36,102
,0
9010 FOR a=0 TO 7
9020 READ b
9030 POKE USR "a"+a,b
9040 NEXT a
9050 RETURN
```

Sugerencias de programación

Mejorando sus programas

«Los artistas han sido siempre los primeros en explorar nuevas tecnologías...» (David Thornburg, Computers and Society (revista Compute!*), marzo de 1982, p. 16)*

La llegada de los ordenadores personales ha dado a muchas personas una salida a su creatividad. El hecho de disponer de un acceso fácil a un ordenador ha hecho entrar en muchos hogares una actividad creativa, despertando habilidades que de otra forma hubieran permanecido dormidas.

Como en todas las formas del arte, los conocimientos y la técnica pueden hacer más fácil la implementación de sus ideas. Si dispone de una serie de trucos y técnicas, usted puede inventar y desarrollar sus programas libremente, sin la angustia que produce el no estar seguro del resultado de su esfuerzo. Por ejemplo, si se sabe cómo conseguir que las figuras se muevan en la pantalla, esto se convierte en la parte no significativa del programa, quedando energías y tiempo para concentrarse en otros aspectos que harán su programa mucho más sugestivo.

El atento examen de los programas de este libro u otros del mismo tipo constituye una vía segura para avanzar en el conocimiento de la programación. Solamente al entrar los programas y leer las notas que los acompañan podrá adquirir cierta cantidad de ideas que podrá incorporar a sus propias producciones. Si todavía es muy novel en el juego de la programación, puede empezar adaptando ligeramente los programas del libro, cambiando los gráficos definidos por el usuario, o las teclas que leen las rutinas INKEY$, y luego ir avanzando progresivamente hasta escribir programas enteros.

Cuando desarrolle un programa, mantenga claro un objetivo «humano». En muchos casos sus programas serán ejecutados por otras personas y, por ello deben ser lo más fáciles de utilizar y lo más divertidos que pueda. Por ejemplo, muchos programas incluidos en este libro y escritos para Spectrum utilizan las teclas «5», «6», «7» y «8» para controlar el movimiento de los objetos. Estas teclas tienen una flecha pintada y, por este motivo son, las más utilizadas en el control

de dirección. Sin embargo, si piensa sobre ello un momento observará que, en ciertos casos, no son las teclas más útiles. Están situadas muy juntas y en la parte superior del teclado. En juegos de acción rápida puede ser más conveniente el uso de teclas más separadas, tales como «Z» para moverse hacia la izquierda y «M» para ir a la derecha. Igualmente, pueden elegirse otras teclas para los movimientos hacia arriba y hacia abajo. Puede ver un ejemplo de lo expuesto en el programa CARRERA MORTAL.

Resulta interesante cuestionarse continuamente todas aquellas cosas que se hacen por hábito, tal como usar las teclas «5» a «8», para analizarlas y ver si son las que producen programas óptimos.

Escoja, pues, teclas que parezcan «naturales» en el juego en relación a su posición en el teclado o por ser iniciales de la palabra que designa la acción a realizar. Por ejemplo, puede usarse la «Z» (en la parte inferior izquierda del teclado) para ejecutar un movimiento a cuarenta y cinco grados en dirección Sudoeste, o la tecla «F» de fuego cuando deba usar su pistola laser para eliminar un extraterrestre, o la «R» de rodar cuando se trata de tirar un dado. Como ejemplo de esta última forma de comunicación, puede tomarse el de LA VENGANZA DEL CASTILLO ENCANTADO, donde se utiliza la tecla «E» para ir hacia el Este o la «HU» para intentar la huida. Una vez lee las instrucciones de un juego con este sistema de comunicación es difícil que las olvide.

Si es posible, el programa debe incluir preguntas explícitas, de tal forma que el jugador sepa en cualquier momento que es lo que debe hacer. Es mejor hacer esto que colocar al principio del programa una larga explicación, que requiere una gran cantidad de memoria y un cierto tiempo de mecanografiado. El Spectrum permite incluir en las instrucciones INPUT frases entre comillas («»); esta posibilidad facilita la inclusión de las citadas pistas en los programas. Seguramente sabe que es posible usar todos los controles de la instrucción PRINT, tales como: FLASH, BRIGHT e INVERSE; dentro de instrucciones INPUT, lo que permite realzar y hacer más comprensibles las indicaciones al jugador.

Una vez haya conseguido un programa que funcione correctamente y que se ajuste a sus deseos, conviene pasar cierto tiempo haciendo la presentación tan atractiva como sea posible. Aunque la salida del programa conste únicamente de instrucciones PRINT, no hay razón para no colocar las frases y números de forma atractiva. Cambiando el color en las instrucciones PRINT o el borde de pantalla a pantalla, puede mejorar la legibilidad del texto. Por ejemplo, en el programa de AJEDREZ el borde de la pantalla cambia de color aleatoriamente cuando una pieza del Spectrum es capturada por el jugador.

El sonido también puede mejorar un programa. Es sorprendente el efecto que pueden causar unos cuantos BEEP bien repartidos. No

es solamente aplicable a programas de acción, donde se desintegra un extraterrestre o se produce el rebote de una pelota, si no que puede realzar programas mucho más tranquilos como AJEDREZ. En este caso, el ordenador hace sonar un BEEP cuando va a la rutina de búsqueda de posibles movimientos. Se incluye esta función para que, en el caso de que el ordenador tarde mucho tiempo en tomar una decisión, el jugador esté seguro de que el programa no se ha perdido en un bucle infinito y mantenga la esperanza de obtener una respuesta.

Vemos, pues, que el uso del color y el sonido mejoran la calidad de cualquier programa. Además, estas características forman parte del software contenido en el Spectrum, siendo ilógico no usarlas, ya que no significan añadir complejidad a sus programas. El mismo argumento puede aplicarse a los gráficos definibles por el usuario. En el programa de AJEDREZ no se han incluido porque se ha perseguido la calidad del juego antes que su belleza. No obstante, no hay excusa (quizá, la pereza) para representar un extraterrestre con un asterisco, pudiendo hacerlo con un gráfico mucho más expresivo.

Sin perder de vista el valor de añadir colores con las instrucciones INK y PAPER o subrayar y realzar palabras con INVERSE, FLASH y BRIGHT, existen formas más sutiles de hacer más legible un mensaje, tales como: incluir algunas palabras en mayúsculas. Puede ver la utilización de este recurso en la segunda versión de LA VENGANZA DEL CASTILLO ENCANTADO. En este caso se han utilizado las mayúsculas para distinguir los textos añadidos a la primera versión. También puede ser importante la posición de la impresión de las frases y, en estas ocasiones, la instrucción PRINT AT es de gran utilidad.

Incremento de la velocidad del programa

En algunos casos la «calidad» de ejecución del programa es más importante que su velocidad de respuesta (como PIRANDELLO o DAMAS en este mismo libro). Pero incluso en programas como los citados, debe cuidarse la velocidad de ejecución para no aburrir al jugador. Por supuesto, en los juegos de acción la velocidad debe ser la preocupación número uno del programador. Un programa de acción lento no resulta nunca satisfactorio.

En las introducciones a algunos de los programas de este libro se ha llamado la atención del lector sobre los recursos usados para incrementar la velocidad del programa considerado. Le remitimos a estos comentarios a parte de los que figuran a continuación.

Cuando un programa llega a una instrucción GO TO o GO SUB, busca el número de línea correspondiente a lo largo de todo el programa, empezando por la primera línea. Por lo tanto, la primera regla a

observar es situar las rutinas de más frecuente uso al principio del programa para disminuir el tiempo de búsqueda en todos los cambios de secuencia. Si existe un bucle principal que el programa recorre repetidamente, conviene situarlo a partir de las primeras líneas. Verá muchos programas en este libro organizados de este modo.

Además, las partes del programa que se usen raramente o una sola vez deben situarse al final. Así, puede observar que las instrucciones que se dan al jugador al principio del juego están siempre situadas al final del listado.

Es evidente que cuantas menos líneas tiene un programa, más rápido se ejecuta. Dado un programa con cierta cantidad de instrucciones, puede minimizarse el número de líneas encadenando en cada línea varias instrucciones. Esto es especialmente cierto con instrucciones PRINT AT que deben ser encadenadas tanto como sea posible. El encadenamiento de instrucciones y la eliminación de instrucciones REM mejora la velocidad de ejecución pero empeora la claridad del programa y sus posibilidades de modificación. En muchos casos convendrá hacer dos versiones de un mismo programa. Una de ellas con muchos comentarios REM y las instrucciones sin encadenar para conseguir la máxima claridad durante el desarrollo del programa. La otra versión de alta velocidad se puede derivar de la primera sacando las líneas REM y compactando las instrucciones, cuando el programa ya funcione correctamente.

Si se programa de una forma cuidadosa y estructurada pueden suprimirse todas las instrucciones GO TO que no estén tipificadas mediante un IF/THEN. Algunos programadores puristas aseguran que la instrucción GO TO incondicional es *siempre* mala: sin embargo, no vemos razón para prescindir de esta instrucción, que algunas veces puede resultar útil. Las instrucciones GO SUB también pueden ser substituidas en muchas ocasiones. Si se utiliza una subrutina para una función no muy larga a la que se llama en dos o tres ocasiones, es mejor repetir esta rutina para cada vez que sea necesaria.

Una vez terminado el programa, es interesante sacar un listado completo del mismo para examinarlo críticamente y ver si los bloques que lo forman con autocontenidos. Descubrirá que, aunque no los haya realizado pensando en estructurarlos, la mayoría tienen una estructura modular espontánea. Sin embargo, puede ocurrir que los bloques no estén colocados de una forma óptima desde el punto de vista del tiempo de ejecución. Resulta conveniente en este caso reorganizar el programa y hacerlo más lógico.

No use paréntesis a menos que sea absolutamente necesario, ya que consumen memoria y alargan la ejecución. Verá que el BASIC del Spectrum requiere menos paréntesis que otros, permitiendo, por ejemplo, la instrucción PRINT CHR$ 134 en lugar de PRINT CHR$ (134).

Encontrará ciertas dificultades en la adecuación de programas escritos para el ZX81 de 16 K para introducirlos en el Spectrum. Los gráficos consumen más memoria en el Spectrum que en el ZX81. Cuando convierta un programa de uno a otro, elimine sistemáticamente todos los REM y condense todas las instrucciones PRINT. Examine luego el listado completo para intentar acortarlo de alguna forma, como, por ejemplo, añadiendo una subrutina para sustituir un bloque que se use más de una vez. Esto hará el programa un poco más lento, pero es el precio que cuesta hacerlo compatible con el Spectrum. Como sabe, el Spectrum es un procesador mucho más rápido que el ZX81, de forma que no es problema sacrificar la velocidad de proceso en este caso.

Apéndice

El generador gótico

Este programa no es un juego, pero puede resultar muy divertido y muy útil. El programa permite redefinir uno o todos los caracteres que imprime el Spectrum. Solamente tiene trece líneas, pero le permite obtener unos efectos extraordinarios.

Por ejemplo, el siguiente listado está realizado en caracteres góticos obtenidos a base de redefinir todo el juego de caracteres:

```
  5 GO TO 200
 10 CLEAR 31400
 15 PRINT "POR FAVOR ESPERE UN
MOMENTO..."
 20 LET I=15616
 30 LET A=31401
 40 FOR F=0 TO 1023
 50 POKE A+F,PEEK (P+F)
 60 NEXT F
 70 LET V=31145
 80 LET N=23606
 90 POKE N,V-256*INT (V/256)
100 POKE N+1,INT (V/256)
150 RUN 9999
200 INPUT "QUE LETRA DESEA CAMB
IAR? ";A$
210 LET A=31401+8*(CODE A$-32)
220 FOR F=0 TO 7
230 LET B$=""
240 INPUT C$: IF C$="" STOP " TH
EN STOP
245 IF LEN C$<>8 AND C$<>"" THE
N BEEP .5,20: GO TO 240
250 LET B=VAL (B$+C$)
255 POKE A+F,B
260 PRINT F+1;TAB 3;C$,B
270 NEXT F
280 PRINT
290 PRINT A$
300 PRINT
310 PRINT
320 PRINT
330 GO TO 200
```

Como puede ver, hace mucho efecto. Seguramente, en la mayoría de los casos no necesitará cambiar el vocabulario completo, sino solamente unos cuantos caracteres. Ejecute el programa y espere unos momentos. Aparecerá el mensaje «Qué letra desea cambiar?». Una vez haya respondido a esta pregunta, el computador le permite entrar una serie de números binarios (línea 24Ø) que redefinirán la letra elegida. Verá que si, por ejemplo, ha redefinido la letra «a», cada vez que aparezca lo hará en la forma de su propio carácter. Aquí tiene el programa listado con caracteres normales:

```
   5 GO TO 200
  10 CLEAR 31400
  15 PRINT "Por favor espere un
momento..."
  20 LET p=15616
  30 LET a=31401
  40 FOR f=0 TO 1023
  50 POKE a+f,PEEK (p+f)
  60 NEXT f
  70 LET v=31145
  80 LET n=23606
  90 POKE n,v-256*INT (v/256)
 100 POKE n+1,INT (v/256)
 150 RUN 9900
 200 INPUT "Que letra desea camb
iar? ";a$
 210 LET a=31401+8*(CODE a$-32)
 220 FOR f=0 TO 7
 230 LET b$="BIN "
 240 INPUT c$: IF c$=" STOP " TH
EN STOP
 245 IF LEN c$<>8 AND c$<>"" THE
N BEEP .5,20: GO TO 240
 250 LET b=VAL (b$+c$)
 255 POKE a+f,b
 260 PRINT f+1;TAB 3;c$,b
 270 NEXT f
 280 PRINT
 290 PRINT a$
 300 PRINT
 310 PRINT
 320 PRINT
 330 GO TO 200
```

Si se trata de reprogramar el alfabeto completo, es más sencillo utilizar números decimales en lugar de binarios. Si realmente necesita un alfabeto gótico, entre el siguiente pequeño programa:

```
1010 FOR a=31400 TO 32190
1020 INPUT (a);"?",p
1030 POKE a,p
1040 NEXT a
1099 STOP
```

Cuando lo ejecute, podrá entrar una larga lista de números. Aparece en la pantalla un número de posición de memoria y usted debe responder entrando los números de la lista adjunta. Una vez lo haya hecho, todo el alfabeto del Spectrum se habrá convertido en gótico. Este puede ser un recurso original para escribir las felicitaciones de Navidad del próximo año.

31400	62	31401	0
31402	0	31403	0
31404	0	31405	0
31406	0	31407	0
31408	0	31409	0
31410	16	31411	16
31412	16	31413	16
31414	0	31415	16
31416	0	31417	0
31418	36	31419	36
31420	0	31421	0
31422	0	31423	0
31424	0	31425	0
31426	36	31427	126
31428	36	31429	36
31430	126	31431	36
31432	0	31433	0
31434	8	31435	62
31436	40	31437	62
31438	16	31439	62
31440	8	31441	0
31442	32	31443	100
31444	8	31445	16
31446	38	31447	70
31448	38	31449	70
31450	16	31451	40
31452	16	31453	42
31454	68	31455	58
31456	0	31457	0
31458	8	31459	16
31460	0	31461	0
31462	0	31463	0
31464	0	31465	0
31466	4	31467	8
31468	8	31469	8
31470	8	31471	4
31472	0	31473	0
31474	32	31475	16
31476	16	31477	16
31478	16	31479	32
31480	0	31481	20
31432	0	31483	20
31434	8	31485	62
31436	8	31487	20
31438	0	31489	0
31490	0	31491	8
31492	8	31493	62
31494	8	31495	8
31496	0	31497	0

31498	0	31499	0
31500	0	31501	0
31502	8	31503	8
31504	16	31505	0
31506	0	31507	0
31508	0	31509	62
31510	0	31511	0
31512	0	31513	0
31514	0	31515	0
31516	0	31517	0
31518	24	31519	24
31520	0	31521	0
31522	0	31523	2
31524	4	31525	8
31526	16	31527	32
31528	16	31529	32
31530	28	31531	38
31532	66	31533	100
31534	56	31535	16
31536	0	31537	0
31538	24	31539	56
31540	24	31541	24
31542	24	31543	60
31544	0	31545	0
31546	56	31547	76
31548	36	31549	12
31550	16	31551	126
31552	0	31553	0
31554	126	31555	4
31556	8	31557	28
31558	70	31559	124
31560	0	31561	0
31562	12	31563	28
31564	44	31565	126
31566	12	31567	30
31568	0	31569	0
31570	126	31571	64
31572	124	31573	6
31574	66	31575	60
31576	0	31577	0
31578	24	31579	48
31580	112	31581	76
31582	100	31583	56
31584	0	31585	0
31586	126	31587	66
31588	4	31589	24
31590	16	31591	48
31592	0	31593	0
31594	28	31595	36
31596	24	31597	28
31598	38	31599	60
31600	0	31601	0
31602	28	31603	38
31604	50	31605	14
31606	12	31607	24
31608	0	31609	0
31610	0	31611	0
31612	16	31613	0
31614	0	31615	16

31616	0	31617	0
31618	0	31619	16
31620	0	31621	0
31622	16	31623	16
31624	32	31625	0
31626	0	31627	4
31628	8	31629	16
31630	8	31631	4
31632	0	31633	0
31634	0	31635	0
31636	62	31637	0
31638	62	31639	0
31640	0	31641	0
31642	0	31643	16
31644	8	31645	4
31646	8	31647	16
31648	0	31649	0
31650	60	31651	66
31652	4	31653	8
31654	0	31655	8
31656	0	31657	0
31658	60	31659	74
31660	36	31661	74
31662	64	31663	60
31664	0	31665	0
31666	56	31667	72
31668	24	31669	40
31670	104	31671	124
31672	0	31673	32
31674	96	31675	32
31676	56	31677	52
31678	52	31679	52
31680	24	31681	16
31682	56	31683	104
31684	96	31685	96
31686	96	31687	48
31688	24	31689	16
31690	32	31691	16
31692	24	31693	44
31694	44	31695	44
31696	16	31697	16
31698	56	31699	104
31700	120	31701	96
31702	96	31703	48
31704	24	31705	14
31706	26	31707	16
31708	60	31709	16
31710	16	31711	48
31712	24	31713	16
31714	40	31715	108
31716	44	31717	28
31718	44	31719	70
31720	60	31721	32
31722	32	31723	44
31724	54	31725	34
31726	54	31727	2
31728	4	31729	16
31730	32	31731	0
31732	48	31733	88

31734	24	31735	26
31736	28	31737	24
31738	0	31739	56
31740	24	31741	24
31742	24	31743	16
31744	96	31745	32
31746	96	31747	230
31748	107	31749	114
31750	108	31751	98
31752	19	31753	16
31754	48	31755	16
31756	16	31757	16
31758	16	31759	28
31760	24	31761	42
31762	126	31763	170
31764	42	31765	42
31766	42	31767	106
31768	0	31769	0
31770	40	31771	116
31772	52	31773	52
31774	52	31775	52
31776	0	31777	24
31778	44	31779	108
31780	108	31781	108
31782	108	31783	104
31784	48	31785	56
31786	52	31787	56
31788	52	31789	56
31790	48	31791	48
31792	64	31793	0
31794	48	31795	88
31796	88	31797	88
31798	56	31799	12
31800	8	31801	0
31802	44	31803	116
31804	32	31805	32
31806	32	31807	96
31808	48	31809	2
31810	28	31811	32
31812	108	31813	118
31814	6	31815	108
31816	48	31817	16
31818	48	31819	126
31820	48	31821	48
31822	48	31823	114
31824	60	31825	0
31826	40	31827	116
31828	52	31829	52
31830	52	31831	52
31832	10	31833	0
31834	34	31835	118
31836	50	31837	50
31838	54	31839	60
31840	16	31841	0
31842	84	31843	234
31844	107	31845	107
31846	107	31847	106
31848	20	31849	54
31850	84	31851	24

31852	60
31854	24
31856	68
31858	38
31860	38
31862	64
31864	12
31866	68
31868	28
31870	2
31872	48
31874	14
31876	8
31878	8
31880	0
31882	0
31884	32
31886	8
31888	0
31890	56
31892	24
31894	104
31896	0
31898	16
31900	84
31902	16
31904	0
31906	0
31908	0
31910	0
31912	255
31914	28
31916	120
31918	32
31920	0
31922	56
31924	24
31926	104
31928	0
31930	96
31932	56
31934	52
31936	24
31938	56
31940	96
31942	96
31944	24
31946	32
31948	24
31950	44
31952	16
31954	56
31956	120
31958	96
31960	24
31962	26
31964	60
31966	16
31968	24

31853	24
31855	42
31857	103
31859	38
31861	124
31863	48
31865	126
31867	8
31869	6
31871	12
31873	0
31875	8
31877	8
31879	14
31881	0
31883	64
31885	16
31887	4
31889	0
31891	72
31893	40
31895	124
31897	32
31899	56
31901	16
31903	16
31905	0
31907	0
31909	0
31911	0
31913	0
31915	34
31917	32
31919	126
31921	0
31923	72
31925	40
31927	124
31929	32
31931	32
31933	52
31935	52
31937	16
31939	104
31941	96
31943	48
31945	16
31947	16
31949	44
31951	44
31953	16
31955	104
31957	96
31959	48
31961	14
31963	16
31965	16
31967	48
31969	16

31970	40	31971	108
31972	44	31973	28
31974	44	31975	70
31976	60	31977	32
31978	32	31979	44
31980	54	31981	34
31982	34	31983	2
31984	4	31985	16
31986	32	31987	0
31988	48	31989	38
31990	24	31991	26
31992	28	31993	24
31994	0	31995	56
31996	24	31997	24
31998	24	31999	16
32000	96	32001	32
32002	96	32003	230
32004	107	32005	114
32006	108	32007	98
32008	19	32009	16
32010	48	32011	16
32012	16	32013	16
32014	16	32015	28
32016	24	32017	43
32018	126	32019	170
32020	42	32021	42
32022	42	32023	106
32024	0	32025	0
32026	40	32027	116
32028	52	32029	52
32030	52	32031	52
32032	0	32033	24
32034	44	32035	108
32036	108	32037	108
32038	108	32039	104
32040	48	32041	56
32042	116	32043	52
32044	52	32045	56
32046	48	32047	48
32048	64	32049	0
32050	48	32051	38
32052	88	32053	38
32054	56	32055	12
32056	8	32057	0
32058	44	32059	116
32060	32	32061	32
32062	32	32063	96
32064	48	32065	2
32066	28	32067	32
32068	108	32069	118
32070	6	32071	108
32072	48	32073	16
32074	48	32075	126
32076	48	32077	48
32078	48	32079	114
32080	60	32081	0
32082	4	32083	116
32084	52	32085	52
32086	52	32087	52

320888	10	320889	0
320890	34	320891	118
320892	50	320893	50
320894	54	320895	60
320896	16	320897	0
320898	84	320899	234
321000	107	321001	107
321002	107	321003	106
321004	20	321005	54
321006	84	321007	24
321008	60	321009	24
321010	24	321011	42
321012	68	321013	108
321014	38	321015	38
321016	38	321017	124
321018	64	321019	48
321020	12	321021	126
321022	68	321023	8
321024	28	321025	6
321026	2	321027	12
321028	18	321029	0
321030	14	321031	8
321032	48	321033	8
321034	8	321035	0
321036	0	321037	0
321038	8	321039	8
321040	8	321041	8
321042	8	321043	8
321044	0	321045	0
321046	0	321047	0
321048	0	321049	0
321050	0	321051	0
321052	0	321053	0
321054	20	321055	40
321056	0	321057	0
321058	0	321059	0
321060	0	321061	60
321062	66	321063	153
321064	161	321065	161
321066	153	321067	66
321068	60	321069	0
321070	0	321071	0
321072	0	321073	0
321074	0	321075	0
321076	0	321077	0
321078	0	321079	0
321080	0	321081	0
321082	0	321083	0
321084	0	321085	0
321086	0	321087	0
321088	0	321089	0
321090	0	321091	0

Autores de los programas

Agradecemos a los siguientes programadores la ayuda prestada en la elaboración de un libro lleno de interesantes y valiosos juegos:

NIVEL CINCO Paul Toland
CIRCO Paul Toland
JOGGER Tim Hartnell
DESACTIVA ESTA MINA Paul Toland
HELICOPTERO Tim Rogers
MAZURCA Paul Toland
ALFABATALLA Tim Rogers
PARED Ant Hurrion (modified bi Tim Hartnell)
CONDUCCION 3-D Tim Rogers
PANICO EN NORUEGA Tim Rogers
CARRERA MORTAL Tim Hartnell and Tim Rogers
CLONO LOCO Mike O'Neill
PIRANDELLO Graham Charlton (modified by Tim Hartnell)
DAMAS Tim Hartnell
TIC-TAC-TOE Tim Hartnell
AJEDREZ Tim Hartnell
LA VENGANZA DEL CASTILLO ENCANTADO Tim Hartnell
JUAN CAPITALISTA (Lemonade Stand) Paul Holmes
SINTAXIS Tim Rogers
EL POZO Tim Rogers
EL GENERADOR GOTICO Tim Rogers

Bibliografia

Hay muchos libros que pueden ayudarle con ideas para escribir juegos para el Spectrum. La lista siguiente incluye algunos de los mejores que hemos descubierto.

Libros de juegos para ordenador:

*40 Computer Games from kilobaud microcomputing** Gibbs E.A. y Perry, J. (eds.) (Wayne Green Inc., Estados Unidos, 1980)

*The Softside** *Sampler* Witham J. (ed.) (Hayden Book Co., Inc., Estados Unidos, 1982)

BASIC Computer Games Ahl D. (ed.) (Workman Publishing Co., Estados Unidos, 1980)

More BASIC Computer Games Ahl D. (ed.) (Workman Publishing Co., Estados Unidos, 1980)

Libros generales de juegos para obtener ideas:

The Way To Play the Diagram Group (eds.) (Corgi, 1977)

The Indoor Games Book Pennycook, A. (Faber & Faber Ltd., 1973)

Games for Two Wasley, J. (Proteus Publishing Ltd., 1981)

Everyman's Indoor Games Brandreth, G. (J.M. Dent & Sons Ltd., Everyman Paperback, 1982)

Dice Games, New and Old Tredd, W.E. (The Oleander Press, Cambridge, 1981)

Discovering Old Board Games Bell R.C. (Shire Publications Ltd., 1980)

*Registered Trade Marks

Los lectores interesados en aprender más sobre la programación del ZX Spectrum pueden encontrar material de interés en el libro **The ZX Spectrum Explored,* de Tim Hartnell, publicado por Sinclair Browne Ltd.

Otros libros sobre MICROINFORMATICA

C. Prigmore — **MICROSOFT BASIC**
Curso de autoenseñanza para principiantes

G. Ladevie — **La gestión con BASIC**
Comercio y pequeña empresa

G. Guérin — **Microinformática de gestión**
Alternativas y utilización

A.P. Mullan — **El ordenador en la Educación Básica**
Problemática y metodología

D. Daines — **Las bases de datos en la Educación Básica**
Utilización y ejemplos

G.W. Orwig/W.S. Hodges — **Programas educacionales para su ordenador personal**

P. Pellier — **Lenguaje máquina del ZX Spectrum**
Subrutinas y trucos

T. Hartnell — **Juegos dinámicos para el ZX Spectrum**

R.G. Hurley — **Los Micro Drives del ZX Spectrum**
Utilización y aplicaciones

I. Sinclair — **Introducción al Commodore 64**

I. Sinclair — **Lenguaje máquina del Commodore 64**

S. Money — **Gráficos y sonidos para el Commodore 64**

O. Bishop — **Juegos para el Commodore 64**

B. Lloyd — **Introducción al Dragon**

D. Lawrence — **Programas prácticos para el Dragon**

K.S. Brain — **Gráficos y sonidos para el Dragon**
Incluye subrutinas en código máquina

K.S. Brain — **Inteligencia artificial en el Dragon**

I. Sinclair — **Lenguaje máquina del Dragon**

M. James/S.M. Gee/K. Ewbank — **Juegos para el Dragon**

V. Apps — **40 juegos educacionales para el Dragon**

Colección «Su ordenador personal»

Así se empieza
Introducción a los ordenadores

Peter Lafferty

204 páginas, de 20 × 14 cm, con más de 100 ilustraciones a dos colores

Indice. Introducción. 1 ¿Qué es un ordenador doméstico? 2 Cómo utilizar su ordenador. 3 ¿Qué puede hacer usted con un ordenador? 4 Cómo escribir sus propios programas. 5 Cómo funcionan los ordenadores. 6 Ampliación del sistema. 7 Elección del ordenador. 8 Hacia el futuro. Apéndice 1. Apéndice 2. Glosario de terminología de ordenadores. Resumen de ordenadores. Bibliografía. Club de usuarios. Indice analítico.

Primeros pasos en BASIC

Susan Curran - Ray Curnow

208 páginas, de 20 × 14 cm, con más de 60 ilustraciones a dos colores

Indice. Introducción. 1 Cómo escribir en la pantalla. 2 Nuestros primeros programas. 3 Introducción de variables. 4 Bucles y ramificaciones. 5 Edición y corrección de errores. 6 Cómo manejar los datos. 7 Cómo escribir programas más largos. 8 Los siguientes pasos. Apéndice 1. Apéndice 2. Apéndice 3. Respuestas a las preguntas. Indice analítico.

El estudiante y el ordenador
Aplicaciones a la enseñanza

Susan Curran - Ray Curnow

168 páginas, de 20 × 14 cm, con más de 40 ilustraciones a dos colores

Indice. Introducción. 1 El ordenador como una ayuda para aprender. 2 Ordenadores para los niños. 3 Programas de recursos. 4 Cómo comprar software. 5 Hardware para la educación. 6 Algunos programas para que usted los pruebe. Apéndice 1. Apéndice 2. Bibliografía. Clubs de usuarios. Indice analítico.

Juegos, imágenes y sonidos

Susan Curran - Ray Curnow

168 páginas, de 20 × 14 cm, con más de 50 ilustraciones a dos colores

Indice. Introducción. 1 Resumen histórico de los juegos por ordenador. 2 Tipos de juegos para ordenador. 3 Gráficos por ordenador. 4 Generación de sonidos mediante su ordenador. 5 Hardware del ordenador. 6 Cómo escribir programas de juegos. 7 Algunos programas que usted puede probar. 8 Compra de programas. Glosario. Indice analítico.

Tim Hartnell brinda en este libro veinte juegos dinámicos para el ZX SPECTRUM. Aparecen desde juegos de tablero, como *Ajedrez* o *Pirandello*, a juegos de acción, como *Jogger* y *La carrera mortal*, e incluye un gran juego de aventuras: *La venganza del castillo encantado*.

Se introduce cada juego con detalle y, en la mayoría de los casos, se explica línea por línea. Se han subrayado los trucos y las técnicas especiales usadas por los programadores y Tim Hartnell le sugiere cómo los puede aplicar a sus propios programas. Al final del libro se ha incluido un capítulo con sugerencias para mejorar sus programas.

Editorial Gustavo Gili, S. A.
Rosellón, 87-89
08029 Barcelona