



MARTIN WREN-HILTON

13 JOGOS PARA O SPECTRUM



16K
OU
48K



MARTIN WREN-HILTON

**13 JOGOS
PARA O SPECTRUM
16K OU 48K**


EDIÇÕES CETOP

EDIÇÕES CETOP

APARTADO 33 • 2726 MEM MARTINS CODEX
SEDE. ESTRADA LISBOA-SINTRA, km 14 • TELEF. 926 13 51 • TELEGS. EDICETOP

COLECÇÃO GO TO INFORMÁTICA

**13 JOGOS PARA O SPECTRUM
16 K OU 48 K**

Volumes publicados nesta colecção:

06 19 001 — *Introdução à Programação em Pascal — Exercícios*

06 19 002 — *13 Jogos para o Spectrum 16K ou 48K*

*Este livro foi traduzido da edição original com o título:
Games to play on your ZX Spectrum*

© *Martin Wren-Hilton, 1982*

*Direitos em língua portuguesa reservados por
CETOP — Centro de Ensino Técnico e Profissional à Distância, Lda.*

*Nenhuma parte desta publicação pode ser re-
produzida ou transmitida por qualquer forma
ou por qualquer processo, electrónico, mecânico
ou fotográfico, incluindo fotocópia, xerocópia
ou gravação, sem autorização prévia e escrita
do editor. Exceptua-se naturalmente a transcri-
ção de pequenos textos ou passagens para apre-
sentação ou crítica do livro. Esta excepção não
deve de modo nenhum ser interpretada como
sendo extensiva à transcrição de textos em re-
colhas antológicas ou similares donde resulte
prejuízo para o interesse pela obra. Os trans-
gressores são passíveis de procedimento judicial*

Edição n.º 06 19 002/059

Editor: Tito Lyon de Castro

Capa: estúdios P. E. A.

Tradução e Revisão Técnica: Dr. Norberto Fernando G. Fernandes

*Execução técnica:
Gráfica Europam, Lda.,
Mem Martins*

INTRODUÇÃO

O Zx Spectrum, quer seja o 16K ou 48K, é uma máquina impressionante. Quero com isto dizer que ele tem um grande potencial — potencial para fazer muito mais do que você pensa. Ao escrever este livro, compilei uma quantidade de jogos fáceis que vos divertirão ou espantar-vos-ão, desde o brilhante programa *BREAKOUT* (Demolidor) até ao confuso *MASTERMIND*. Além de ser divertido fazer correr estes programas, eles ensinarão também algumas técnicas de programação interessantes, bem como alguns comandos que não se encontram disponíveis no *Manual* do seu Spectrum!

Aprendendo como usar essa máquina e através dela, acredito que você também poderá modificar qualquer dos programas listados, adicionando linhas onde quiser. Não quer isto dizer que estes programas contenham erros que precisem de ser corrigidos, mas que uma pequena alteração aqui ou ali, poderá ajudar a «personalizar» o programa.

É necessário notar que todos estes programas correm em qualquer modelo do Spectrum.

Finalmente, espero que aprecie programar o seu Spectrum e que tenha muitas horas de prazer jogando.

MARTIN WREN-HILTON

NOTA DO EDITOR PORTUGUÊS

Todos os programas contidos neste livro são apresentados de duas formas.


Na primeira reproduziu-se rigorosamente a versão original, em letra tipográfica e conforme o autor a concebeu.

Na segunda, adaptaram-se para português nomes de ficheiros, comentários, etc., de modo a tornar os jogos mais compreensíveis para quem não esteja familiarizado com a língua inglesa. A impressão dos programas, neste caso, foi efectuada numa impressora T.S. 2040.


ÍNDICE

	Pág.
Introdução	7
1 Demolidor <i>Breakout</i>	9
2 Encordoamento artístico <i>String art</i>	13
3 Helicóptero <i>Helicopter</i>	17
4 Corrida de minhocas <i>Worm race</i>	23
5 Flor <i>Flower</i>	27
6 <i>Mastermind</i>	29
7 Monitor <i>Monitor</i>	35
8 Bombardeiro <i>Bomber</i>	39
9 Caleidoscópio <i>Kaleidoscope</i>	45
10 Cliente <i>Customer</i>	49
11 Espiral <i>Spiral</i>	59
12 Moto suicida <i>Stunt bike</i>	61
13 Damas <i>Draughts</i>	67

Tome nota das seguintes convenções usadas para representar caracteres gráficos nas listagens dos programas:

Um g pequeno significa «carácter gráfico»: é seguido pela letra ou número da tecla correspondente. Assim «g6» quer dizer  .

Os caracteres que necessitam de CAPS SHIFT têm um «c» adicionado. Assim «g6c» é  .

Um quadrado  é usado para representar um espaço em branco.

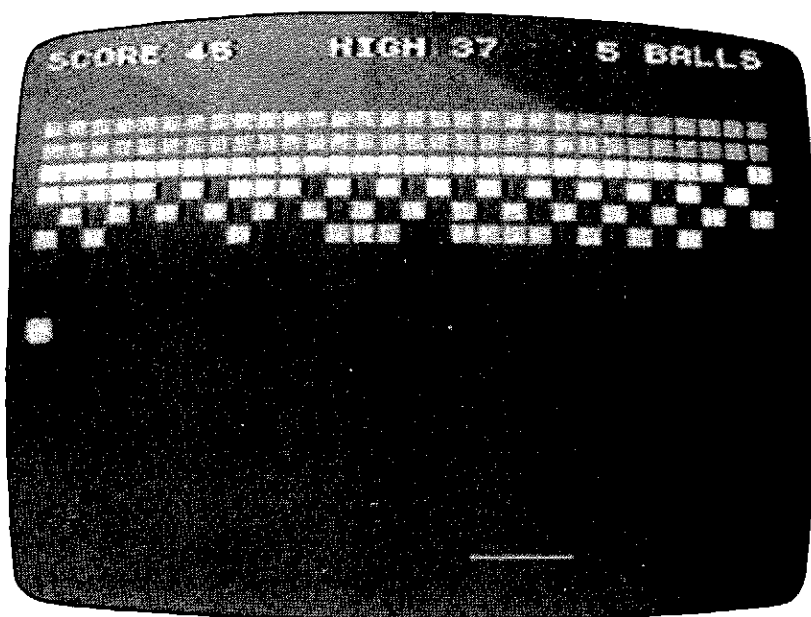
Quadrados à volta da caracteres, querem dizer inverso-video.

Um jogo que vai fazê-lo subir
pelas paredes...



DEMOLIDOR

breakout



Este jogo, baseado no jogo electrónico com o mesmo nome, parece ser extremamente popular, a julgar pela reacção dos meus amigos quando o jogam. Para aqueles desafortunados que nunca jogaram *BREAKOUT* devo dizer que se trata de dismantelar uma série de tijolos de uma parede, usando uma raquete (controlada por si). Parece simples à primeira vista, mas na prática, é muito mais que um desafio.

O programa deve ser introduzido como está listado, mas tendo em atenção que nas linhas 100 a 115 os caracteres que lá estão são sublinhados — quatro em cada linha do programa.

A tecla 5 move a raquete para a esquerda e a 8 para a direita. Veja se consegue bater o meu recorde de 243.

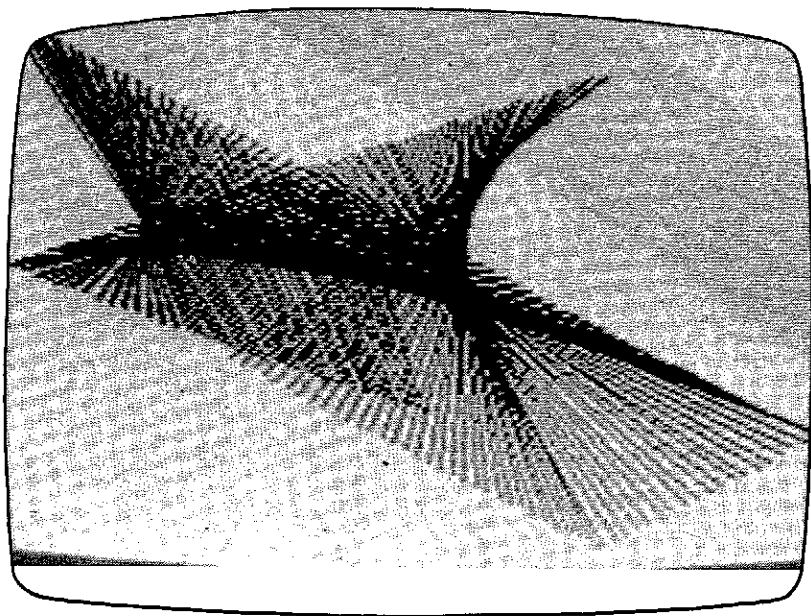
```

10 LET HS = 0: OVER 0
20 LET SC = 0: RESTORE: BORDER 0: PAPER 0: INK 1: BRIGHT 1:
   CLS: LET S = INT (RND * 5 + 16): LET X = 10: LET Y = 5:
   LET A1 = -1: LET B1 = 1: LET N = 9
30 FOR B = USR "A" TO USR "A" + 15: READ C: POKE B, C: NEXT B
50 PRINT INK 6; "□ SCORE □"; SC: TAB 13; "HIGH □"; HS;
   TAB 24; "9 BALLS" ' ' '
60 FOR A = 1 TO 6: READ B: INK B
70 PRINT "□ gB [30 times]"
80 NEXT A
85 INK 1
90 LET Z = ATTR (X, Y) < > 65 AND ATTR (X, Y) < > 67
95 PRINT AT X, Y; INK 5; "gA"
100 LET S = S + (INKEY$ = "8") - (INKEY$ = "5"): LET S =
   S + (S = -1) - (S = 26): PRINT AT 21, S; INK 3; "□ _ _ _ _ □"
110 LET A1 = A1 - 2 * A1 * Z - 2 * (X = 21) + 2 * (X = 2):
   LET B1 = B1 + 2 * (Y = 1) - 2 * (Y = 30): IF Z THEN LET SC = SC + 1:
   BEEP .01, 45 - 2 * x: PRINT AT 0, 7; INK 6; SC
115 LET S = S + (INKEY$ = "8") - (INKEY$ = "5"): LET S =
   S + (S = -1) - (S = 26): PRINT AT 21, S; INK 3; "□ _ _ _ _ □"
120 PRINT AT X, Y; "□": LET X = X + A1: LET Y = Y + B1:
   IF (Y < S + 1 OR Y > S + 4) AND X = 21 THEN GO TO 140
125 IF SC = 180 THEN PRINT AT 3, 0; : RESTORE 1020: GO TO 60
130 GO TO 90
140 PRINT AT 10, 11; INK 2; PAPER 7; "□ MISSED! □"; AT X, Y;
   INK 2; PAPER 0; "gA": BEEP 1, -10
150 LET N = N - 1
160 IF N = 0 THEN GO TO 200
170 PRINT AT 0, 24; INK 6; N
180 FOR Q = 1 TO 250: NEXT Q: PRINT AT 10, 11; "[10 spaces]";

```


ENCORDOAMENTO ARTÍSTICO

string art



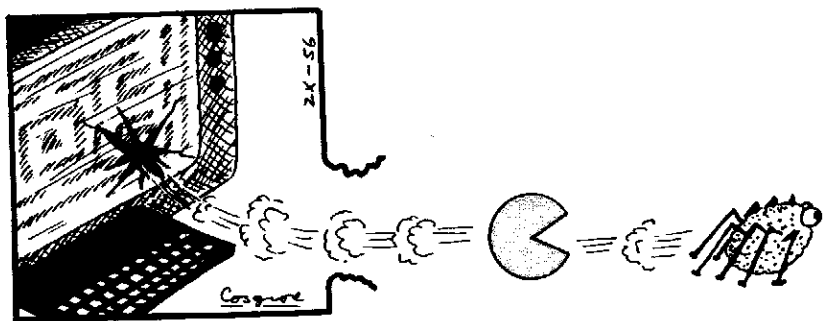
Este curto programa é mais uma divertida demonstração gráfica da habilidade do Spectrum para gerar «encordoamentos artísticos». Para aqueles que não conhecem, o encordoamento artístico envolve um painel revestido a tecido onde são martelados pregos, numa série de linhas, os quais serão posteriormente ligados por fios coloridos. Esta técnica muito simples produz resultados muito atractivos tendo em conta os materiais que são usados. Nesta simulação, o computador desenha uma série de linhas que crescem gradualmente, contraem-se, mudam de direcção e esbarram nos limites do écran. Para aqueles que já jogaram com o *Qix* esta espécie de movimento será familiar.

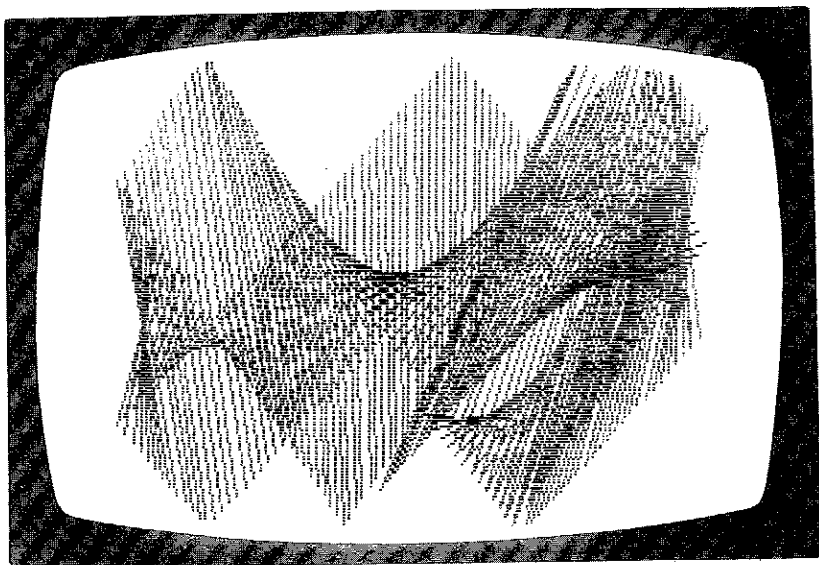
Não há nenhuma particularidades especiais neste programa, excepto numa das suas linhas que permite mudar aleatoriamente a cor do entrelaçado o que pode permitir melhorar o efeito geral.

```

10 BORDER 1: PAPER 7: INK 0: CLS
20 LET X = RND * 255: LET Y = RND * 175
30 LET L = RND * 255: LET M = RND * 175
40 LET U = 4 - RND * 8
50 LET V = 4 - RND * 8
60 LET P = 4 - RND * 8
70 LET Q = 4 - RND * 8
120 FOR K = 1 TO 150
130 BEEP .02, L/5: PLOT X, Y: DRAW L - X, M - Y
140 IF L + P > 250 OR L + P < 5 THEN LET P = -P
150 IF M + Q > 170 OR M + Q < 5 THEN LET Q = -Q
160 IF X + U > 250 OR X + U < 5 THEN LET U = -U
170 IF Y + V > 170 OR Y + V < 5 THEN LET V = -V
180 LET X = X + U
190 LET Y = Y + V
200 LET L = L + P
210 LET M = M + Q
230 NEXT K
240 IF INKEY$ = "" THEN GO TO 240
250 RUN

```





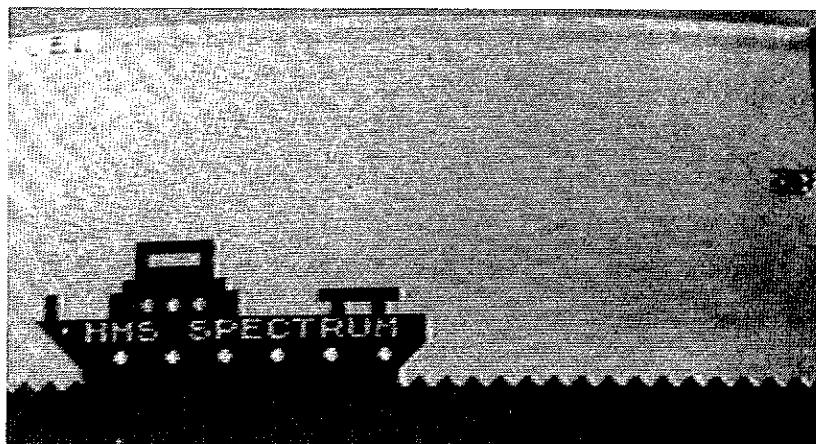
```
10 BORDER 2: PAPER 1: PRINT :
PAPER 1: INK 6: CLS
20 LET X=RND*255: LET Y=RND*17
S
30 LET L=RND*255: LET M=RND*17
S
40 LET U=4-RND*8
50 LET V=4-RND*8
60 LET P=4-RND*8
70 LET Q=4-RND*8
120 FOR K=1 TO 250
130 PLOT X,Y: DRAW L-X,M-Y
140 IF L+P>250 OR L+P<5 THEN LE
T P=-P
150 IF M+Q>170 OR M+Q<5 THEN LE
T Q=-Q
160 IF X+U>250 OR X+U<5 THEN LE
T U=-U
170 IF Y+V>170 OR Y+V<5 THEN LE
T V=-V
180 LET X=X+U
190 LET Y=Y+V
200 LET L=L+P
210 LET M=M+Q
230 NEXT K
240 IF INKEY#="" THEN GO TO 240
250 RUN
```


*Para cima um pouco,
um pouco para a esquerda...*

3

HELICÓPTERO

helicopter



Já pilotou alguma vez um helicóptero? Não? Nem eu, mas aqui está a sua oportunidade de fazê-lo. A parte principal do jogo consiste em aterrar com o helicóptero na plataforma de um barco, sem ficar sem combustível. Começa com 500 galões, o que é representado por uma linha verde. À medida que o combustível vai sendo consumido, esta linha vai ficando amarela até aos 200 galões e vermelha até aos 100. Se deixar os comandos do helicóptero, ele abranda e a sua distância do nível do mar diminui.

Tente, jogando por si, utilizando as teclas 0 e 1. Não se esqueça que o computador não reage se estiver a carregar nas duas teclas simultaneamente, portanto carregue uma de cada vez! No início, provavelmente, cairá algumas vezes mas, à medida que vai agarrando o jeito, tente gastar menos combustível. Um grande desafio!

À parte as palavras que aparecem no écran, todas as outras são símbolos gráficos. Se introduzir as linhas 1000 até 1060 primeiro e depois as 10 a 30 e premir RUN, os símbolos gráficos farão mais sentido.

```

10 LET S = 500: RESTORE: PAPER 5: BORDER 3: CLS
15 LET F$ = "g3c [28 times]"
20 FOR A = 1 TO 7: READ A$
30 FOR B = 0 TO 7: READ C: POKE USR A$ + B, C: NEXT B
40 NEXT A
50 INK 0
60 PRINT AT 0, 0; INK 2; PAPER 7; "FUEL" ' ' ' ' ' ' ' ' ' ' ' '
70 PRINT "□□□□□□□ g4c g3 g7 [28 spaces] g2c g8c g8c g8c g1c
[24 spaces] g5 □□□ g8c gD gD gD g8c □□□ g7 g3 g4c [18 spaces]
gB g8c HMS g8c SPECTRUM g8c [18 spaces] gB g8c gD g8c gD
g8c gD g8c gD g8c gD g8c gD gA [19 spaces] gB g8c [10 times] gA"
80 PRINT AT 14, 0; INK 1; "gC gC gC gC"; AT 14, 16; "gC [16 times]";
PAPER 1; INK 0; AT 14, 4; "gB"; AT 14, 15; "gA"
90 FOR A = 1 TO 7
100 PRINT PAPER 1; "[32 spaces]"
110 NEXT A
120 PRINT AT 13, 5; PAPER 6; "gD g8c gD g8c gD g8c gD g8c gD g8c gD"
130 LET X = 5: LET Y = 30
135 IF SCREEN$(X, Y) <> "□" THEN GO TO 400
140 PRINT AT X, Y; "gE gF"
145 IF INT (X + .5) = 10 AND (INT Y = 14 OR INT Y = 13 OR INT Y = 12)
THEN GO TO 500
150 PAUSE 10
155 PRINT AT 14, 0; INK 1; "gG gG gG gG"; AT 14, 16; "gG [16 times]"
160 PRINT AT 0, 5; INK 4; (CHR$ 16 AND S < 200); (CHR$ 6 AND
(S < 200 AND S >= 100)); (CHR$ 2 AND S < 100); F$ (TO S/18.6);
"[29 spaces]"; IF S = 0 THEN GO TO 600
165 LET S = S - 4
170 PRINT AT X, Y; "□□"

```

```

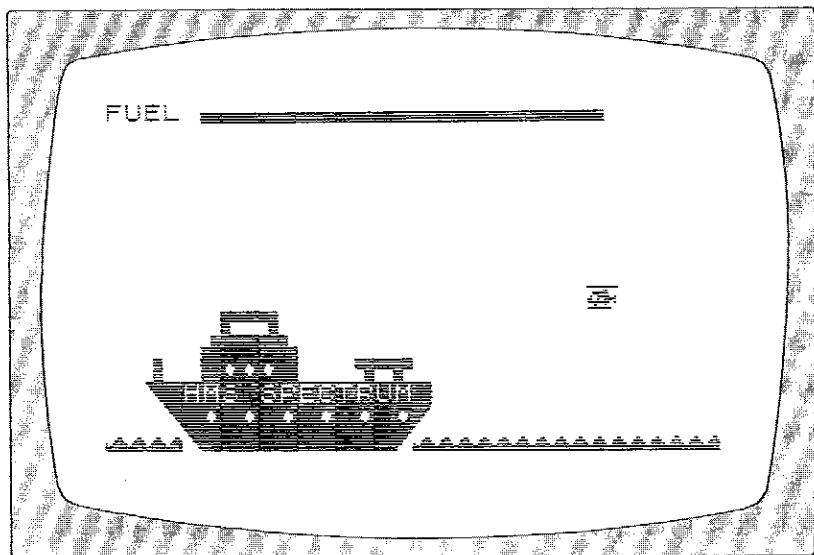
175 LET X = X - (INKEY$ = "1") + (INT X = 2): LET Y = Y -
    (INKEY$ = "0")
180 LET Y = Y + .25 - (INT Y = 30): LET X = X + .5
190 PRINT AT 14, 0; INK 1; "gC gC gC gC"; AT 14, 16; "gC [16 times]"
200 GO TO 135
300 PRINT AT X, Y; INK 1; "gC gC"
310 FOR A = 1 TO 7
320 PRINT AT 14 + A, Y; PAPER 1; INK 7; "gE gF"
325 IF A > 1 THEN PRINT AT 13 + A, Y; PAPER 1; "□ □"
330 BEEP .25, 1 - A: PAUSE 10
340 NEXT A
350 PRINT AT 2, 0; "Oh dear, you've just sunk to the bottom of the ocean."
360 INPUT "HIT ENTER TO TRY AGAIN □"; Q$
370 RUN
400 IF INT X = 13 THEN GO TO 300
410 FOR A = 1 TO 10
420 PAUSE 5: PRINT AT X, Y; INK 2; "gE gF"
430 PAUSE 5: PRINT AT X, Y; "□ □"
440 NEXT A
450 PRINT AT X, Y - 1; INK 2; PAPER 7; BRIGHT 1; "BOOM!"
460 PRINT AT 2, 0; "That's no way to treat a heli- □ □ copter!"
470 GO TO 360
500 PRINT AT 2, 0; "Well done! You have successfully landed your
    helicopter, with □ □ □ □"; S; "□ gallons of fuel left"
510 FOR A = 1 TO 3
520 BEEP .2, 6: PAUSE 10
530 NEXT A
540 GO TO 360
600 PRINT AT 2, 0; "Sorry, you're out of fuel."

```

```

610 GOTO 360
1000 DATA "A", 255, 254, 252, 248, 240, 224, 192, 128
1010 DATA "B", 127, 63, 31, 15, 7, 3, 1, 0
1020 DATA "C", 0, 0, 0, 96, 240, 249, 255, 255
1030 DATA "D", 255, 255, 231, 195, 195, 231, 255, 255
1040 DATA "E", 127, 1, 31, 39, 100, 127, 8, 127
1050 DATA "F", 252, 128, 192, 200, 248, 200, 128, 224
1060 DATA "G", 0, 0, 0, 6, 15, 159, 255, 255

```



```

10 LET s=500: RESTORE : PAPER
5: BORDER 3: CLS
15 LET f#=""
20 FOR a=1 TO 7: READ a#
30 FOR b=0 TO 7: READ : POKE
USR a#+b,c: NEXT b
40 NEXT a
50 INK 0
60 PRINT AT 0,0: INK 2, PAPER
7) "FUEL"
70 PRINT "

```




```

430 PAUSE 5: PRINT AT x,y); " "
440 NEXT a
450 PRINT AT x,y-1; INK 2; PAPER 7; BRIGHT 1;"BUUUUM!"
460 PRINT AT 2,0;"Que maneira d
e tratar um heli- coptero!"
470 GO TO 380
500 PRINT AT 2,0;"BOA! Consegui
u pousar o helicop-tero, com
;5;" litros de fuel."
540 GO TO 360
600 PRINT AT 2,0;"Que pena... a
cabou o fuel."
610 GO TO 360
1000 DATA "a",255,254,250,240,24
0,224,192,128
1010 DATA "b",127,63,31,15,7,3,1
,0
1020 DATA "c",0,0,0,96,240,240,2
55,255
1030 DATA "d",255,255,231,195,19
5,231,255,200
1040 DATA "e",127,1,31,36,100,12
7,0,127
1050 DATA "f",252,128,192,200,24
0,200,128,224
1060 DATA "g",0,0,0,6,15,150,255
,200

```

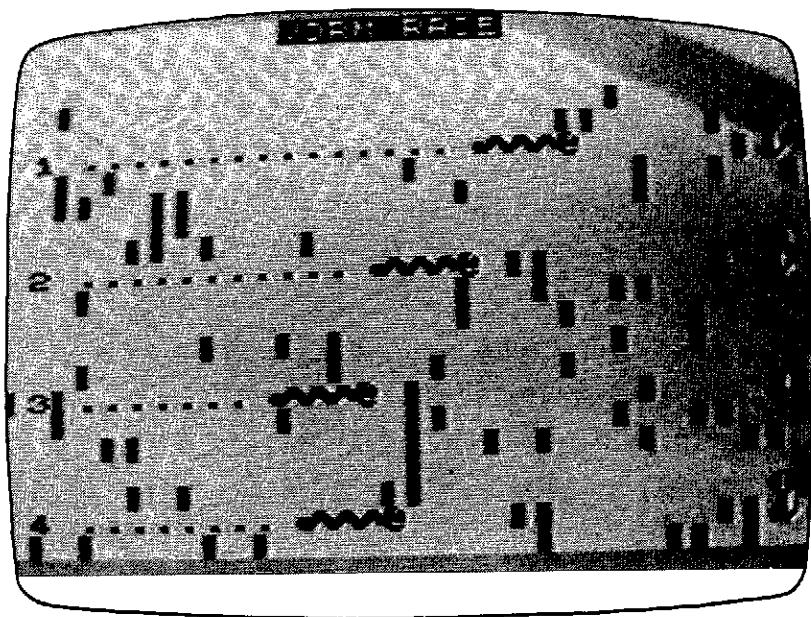


*Já ouviu com certeza falar de minhocas.
E de competição de minhocas?*



CORRIDA DE MINHOCAS

WORM RACE



Este foi o primeiro programa que escrevi num computador (um ZX 81). Gosto dele porque é muito louco! O principal — não sei se o será realmente — é apoiar uma das quatro minhocas enquanto ela corre na relva, tentando atingir a sua flor. Tente dar nomes de conhecidos, amigos, professores, patrões, etc., e observe a sua caminhada pela relva. Os efeitos de som são particularmente bons apesar de muito simples e dão ao jogo um novo significado. Uma noite nas corridas de minhocas é altamente recomendável.

Veja se consegue descobrir como os sons são produzidos.


```

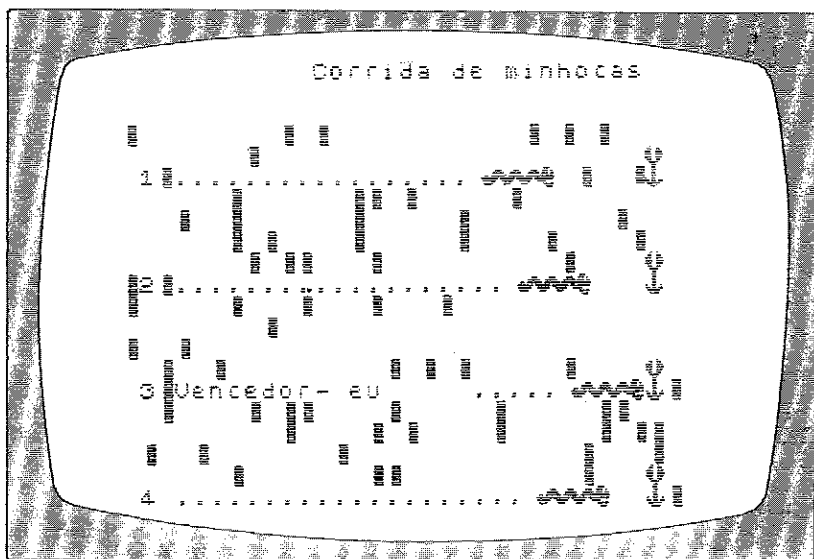
10 DIM L(4): DIM N$(4, 7): RESTORE: INK1: PAPER 7: BORDER 6: CLS
20 PRINT AT 0, 11; INK1; FLASH 1; "WORM RACE"'''
30 FOR A = 1 TO 5
40 READ A$
50 FOR N = 0 TO 7: READ B: POKE USR A$ + N, B: NEXT N
60 NEXT A
70 FOR A = 1 TO 19
80 PRINT "[33 spaces]"
90 NEXT A
100 FOR A = 1 TO 100
110 PRINT AT 3 + INT (RND * 19), INT (RND * 32); INK 4; "g5"
120 NEXT A
130 FOR A = 1 TO 4
140 PRINT AT A * 5, 3; INK 0; "gB gA gA gA gC"; AT A * 5, 1; A;
    AT A * 5, 30; "gD"; INK 2; AT A * 5 - 1, 30; "gE"
150 INPUT "Enter name (7 letters)"; N$(A)
160 LET L(A) = 2
170 NEXT A
180 LET M = 1 + INT (RND * 4)
190 LET L(M) = L(M) + 1
200 IF L(M) = 25 THEN GO TO 240
210 PRINT AT M * 5, L(M); INK 0; PAPER 7; ". gB gA gA gA gC"
220 BEEP .03, L(M)
230 GO TO 180
240 PRINT AT M * 5, 3; PAPER 7; "WINNER — □"; N$(M)
250 FOR A = 1 TO 4
260 PRINT AT M * 5 - 1, 30; INK 5; "gE"
270 BEEP .5, 10
280 PRINT AT M * 5 - 1, 30; INK 2; "gE"

```

```

290 PAUSE 25
300 NEXT A
310 INPUT "HIT ENTER TO RACE AGAIN!"; Q$
320 RUN
1000 DATA "A", 0, 14, 31, 191, 251, 241, 224, 0
1010 DATA "B", 0, 0, 0, 1, 3, 3, 1, 0
1020 DATA "C", 56, 116, 118, 254, 254, 224, 62, 28
1030 DATA "D", 24, 24, 24, 24, 153, 219, 126, 60
1040 DATA "E", 36, 102, 231, 231, 231, 102, 60, 24

```



```

10 DIM L(4): DIM N$(4,7): REST
ORE : INK 1: PAPER 7: BORDER 6:
CLS
20 PRINT AT 0,11: INK 1: FLASH
1:"Corrida de minhocas"
30 FOR a=1 TO 3
40 READ a$
50 FOR n=0 TO 7: READ b: POKE
USR a$+n,b: NEXT n
60 NEXT a
70 FOR a=1 TO 19
80 NEXT a

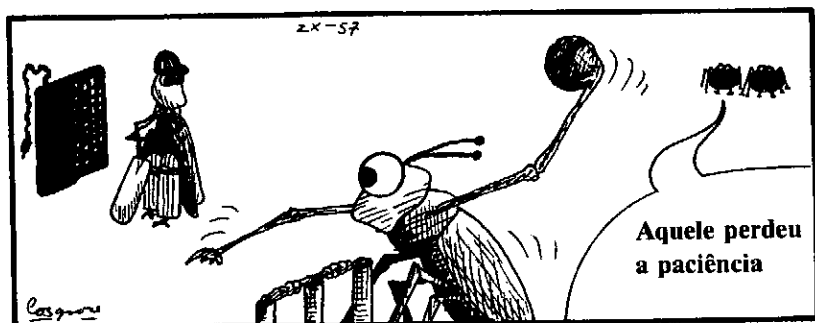
```



```

100 FOR a=1 TO 100
110 PRINT AT 3+INT (RND*19),INT
(RND*32); INK 4;"█"
120 NEXT a
130 FOR a=1 TO 4
140 PRINT AT a*5,3; INK 0;"~~~~~"
";AT a*5,1;a;AT a*5,30;"↓"; INK
2;AT a*5-1,30;"⊙"
150 INPUT "Escreva o seu nome (
max. 7 letras)";n$(a)
160 LET l(a)=2
170 NEXT a
180 LET m=1+INT (RND*4)
190 LET l(m)=l(m)+1
200 IF l(m)=25 THEN GO TO 240
210 PRINT AT m*5,l(m); INK 0; P
APER 7;"·~~~~~"
220 BEEP .03,l(m)
230 GO TO 180
240 PRINT AT m*5,3; PAPER 7;"Ve
ncedor- ";n$(m)
250 FOR a=1 TO 4
260 PRINT AT m*5-1,30; INK 5;"⊙"
"
"
270 BEEP .5,10
280 PRINT AT m*5-1,30; INK 2;"⊙"
"
"
290 PAUSE 25
300 NEXT a
310 INPUT "ENTER para outra cor
rida";q#
320 RUN
1000 DATA "a",0,14,31,191,251,24
1,224,0
1010 DATA "b",0,0,0,1,3,3,1,0
1020 DATA "c",56,116,116,254,254
,224,62,28
1030 DATA "d",24,24,24,24,156,21
5,126,60
1040 DATA "e",36,102,231,231,231
,102,60,24

```



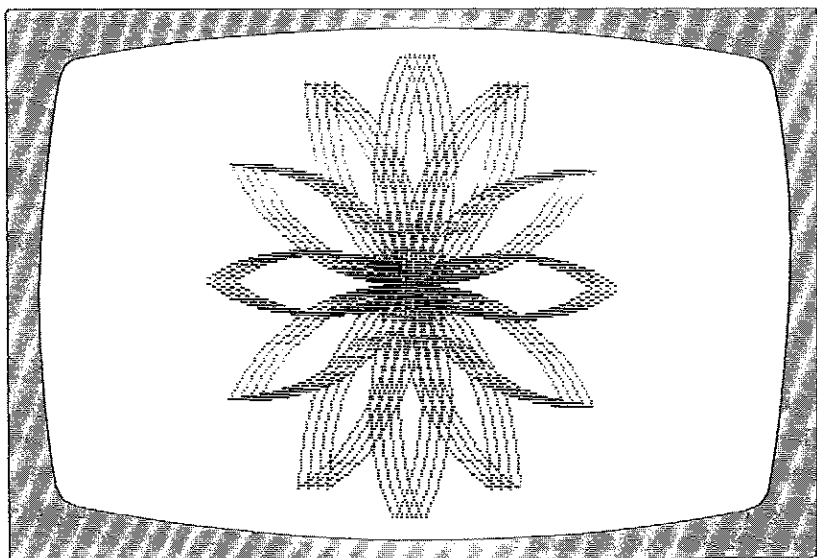


FLOR

flower

A habilidade do Spectrum para rapidamente DESENHAR arcos é muito raramente usada em programas. Este programa desenha uma flor no écran usando esta habilidade. Mudando linhas no programa pode variar-se o número, o tamanho e a forma das pétalas. Eu não lhe digo que variáveis deve modificar, porque pode conseguir resultados interessantes por acidente!

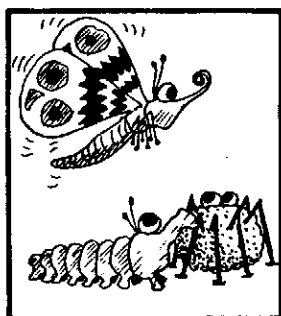
```
5 BORDER 0: PAPER 0: INK 1 + RND * 7: CLS: FOR t = 124 TO
  136 STEP 4
10 FOR a = 0 TO 2 * PI STEP 2 * PI / 12
15 LET x = t + SIN a * 80: LET y = 85 + COS a * 80
20 PLOT x, y
30 LET a1 = a - 2 * PI / 3
35 LET x1 = t + SIN a1 * 80: LET y1 = 85 + COS a1 * 80
40 BEEP .03, y1/8: DRAW x1 - x, y1 - y, - 2 * PI / 3
50 NEXT a
55 NEXT t
```



```

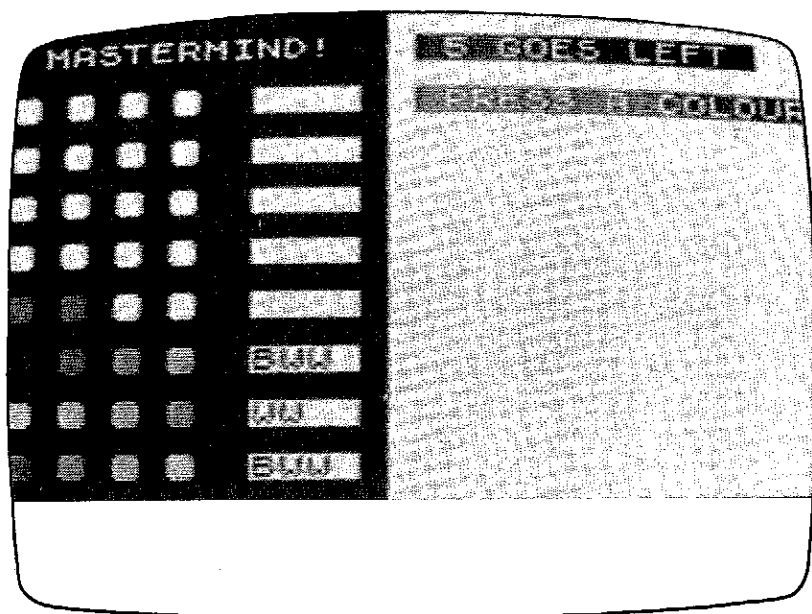
5 BORDER 0: PAPER 0: INK 1+RN
D#7: CLS : FOR t=124 TO 136 STEP
4
10 FOR a=0 TO 2*PI STEP 2*PI/1
2
15 LET x=t+5*SIN a#80. LET y=85+
COS a#80
20 PLOT x,y
30 LET a1=a-2*PI/3
35 LET x1=t+5*SIN a1#80. LET y1=
85+COS a1#80
40 BEEP .83,61/8: DRAW x1-x,y1
-y,-2*PI/3
50 NEXT a
55 NEXT t

```



MASTERMIND

mastermind



Aconselha-se a jogar este jogo numa televisão a cores porque vai ser necessário distinguir várias cores (muito necessário). Se nunca jogou o *Mastermind* então aqui está a sua oportunidade. O seu adversário — o Spectrum — inventa séries de quatro cores das seis possíveis que você vai tentar adivinhar (espero que use um pouco da sua dedução) tentando fazê-lo em oito jogadas ou menos. Por cada cavilha (as cavilhas são usadas no jogo original com o mesmo nome) colocada na coluna correcta com a cor correcta (evidentemente) recebe uma marca preta mas se a cor estiver correcta mas na coluna errada recebe uma marca branca. O computador não repete nenhuma cor no seu código, nem deixa nenhum espaço sem cavilha. Confuso? Se estiver, deve tentar jogar para obter todos os esclarecimentos. Se continuar a ter dificuldades é melhor perguntar a algum amigo se já alguma vez jogou o *MASTERMIND*.

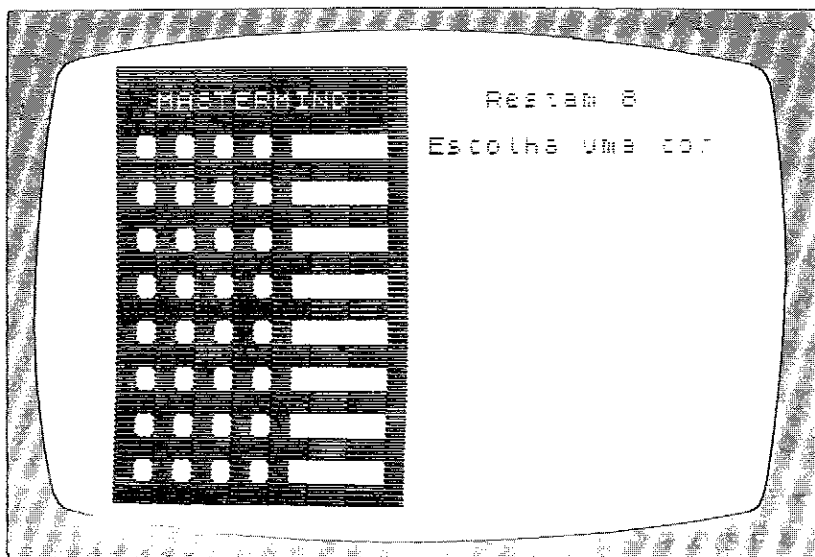
Somente o gráfico A é usado neste programa, para representar uma cavilha. Este gráfico (com o espera) aparece nos comandos PRINT, incluindo o gráfico inverso-video da linha 70 para representar os buracos onde entram as cavilhas. Para ajudá-lo a construir o programa, as linhas 235-245 arranjam aleatoriamente as quatro cores e as linhas 310-350 fazem a sua pontuação.

```
10 DIM G (4): BORDER 4: INK 0: PAPER 7: CLS: RESTORE: LET X = 8
20 FOR A = 0 TO 7: READ B: POKE USR "A" + A, B: NEXT A
30 PRINT "g8C [15 times]"
40 PRINT "g8c g8c MASTERMIND! g8c g8c"
50 PRINT "g8c [15 times]"
60 FOR F = 1 TO 8
70 PRINT "g8c gA g8c gA g8c gA g8c gA g8c g8c □ □ □ □ g8c"
80 PRINT "g8c [15 times]"
90 NEXT F
220 PRINT AT 1, 16: INK 6: PAPER 1; "□"; X; "□ GOES LEFT □"
230 PRINT AT 3, 16: INK 7: PAPER 2: FLASH 1; "□ PRESS A COLOUR □"
235 DIM A (4): FOR N = 1 TO 4
240 LET A (N) = INT (RND * 6) + 1: FOR M = 1 TO 4:
    IF A (M) = A (N) AND N < > M THEN GO TO 240
245 NEXT M: NEXT N
250 FOR N = 1 TO 4
260 LET IS$ = INKEY$: IF IS$ = " " THEN GO TO 260
270 IF IS$ < "1" OR IS$ > "6" THEN GO TO 260
280 PRINT AT X * 2 + 1, N * 2 - 1: PAPER VAL IS; " gA ":
    LET G (N) = VAL IS
290 IF INKEY$ < > " " THEN GO TO 290
```

```

300 NEXT N
310 LET B = 0: LET W = 0
320 FOR N = 1 TO 4: IF G (N) = A (N) THEN LET B = B + 1: GO TO 350
330 FOR M = 1 TO 4: IF N <> M AND G (N) = A (M)
    THEN LET W = W + 1: GO TO 350
340 NEXT M
350 NEXT N
360 PRINT AT X * 2 + 1, 10;
370 FOR A = 1 TO B: PRINT "B"; : NEXT A
380 FOR A = 1 TO W: PRINT "W"; : NEXT A
390 IF B = 4 THEN GO TO 420
400 LET X = X - 1
405 PRINT AT 1, 17; INK 6; PAPER 1; X: IF X = 0 THEN GO TO 500
410 GO TO 250
420 PRINT AT 3, 16; INK 3; PAPER 7; "You've got it! □ □"
430 INPUT "HIT ENTER TO PLAY AGAIN □"; QS
440 RUN
500 PRINT AT 20, 0; "THE ANSWER WAS □";
510 FOR A = 1 TO 4
520 PRINT INK A (A); "gA □";
530 NEXT A
540 GO TO 430
1000 DATA 126, 255, 255, 255, 255, 255, 126

```

```

10 DIM a(4): BORDER 4: INK 0:
PAPER 7: CLS: RESTORE: LET x=0
20 FOR a=0 TO 7: READ b: POKE
US0 "a"+a,b: NEXT a
30 PRINT " "
40 PRINT " "
50 PRINT " "
60 FOR f=1 TO 8
70 PRINT " "
80 PRINT " "
90 NEXT f
220 PRINT AT 1,19: INK 6: PAPER
1: "Restam "x
230 PRINT AT 3,15: INK 7: PAPER
2: FLASH 1: "Escolha uma cor "
235 DIM a(4): FOR n=1 TO 4
240 LET a(n)=INT (RND*6)+1: FOR
m=1 TO 4: IF a(m)=a(n) AND m<>n
THEN GO TO 240
245 NEXT m: NEXT n
250 FOR n=1 TO 4
260 LET i#=INKEY$: IF i#="" THE
N GO TO 260
270 IF i#<"1" OR i#>"5" THEN GO
TO 260
280 PRINT AT x*2+1,n*2-1: PAPER
VAL i#,"X": LET a(n)=VAL i#
290 IF INKEY#<>" " THEN GO TO 29
3
300 NEXT n
310 LET b=0: LET w=0

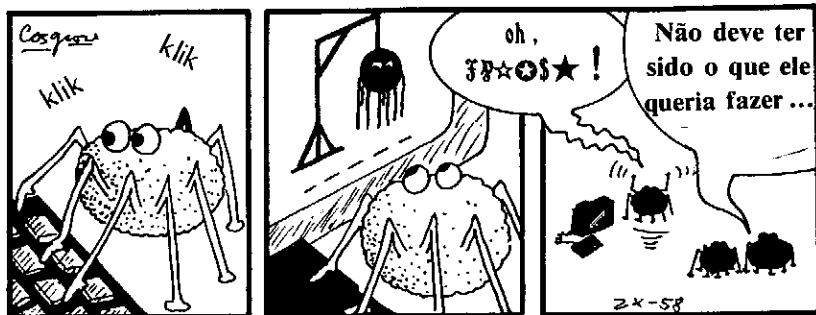
```



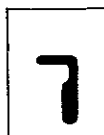
```

320 FOR n=1 TO 4: IF s(n)=a(n)
THEN LET b=b+1: GO TO 350
330 FOR m=1 TO 4: IF n<>m AND s
(n)=a(m) THEN LET w=w+1: GO TO 3
50
340 NEXT m
350 NEXT n
360 PRINT AT x+2+1,10;
370 FOR a=1 TO b: PRINT "C";: N
EXT a
380 FOR a=1 TO w: PRINT "E";: N
EXT a
390 IF b=4 THEN GO TO 420
400 LET x=x-1
405 PRINT AT 1,26; INK 6; PAPER
1;x: IF x=0 THEN GO TO 500
410 GO TO 250
420 PRINT AT 3,16; INK 3; PAPER
7;"DESCOBRIRU-O!!"
430 INPUT "Carregue em ENTER pa
ra jogar      ";q#
440 RUN
500 PRINT AT 20,0;"A resposta e
ra ";
510 FOR a=1 TO 4
520 PRINT INK a(a);"●";
530 NEXT a
540 GO TO 430
1000 DATA 126,255,255,255,255,25
5,255,126

```

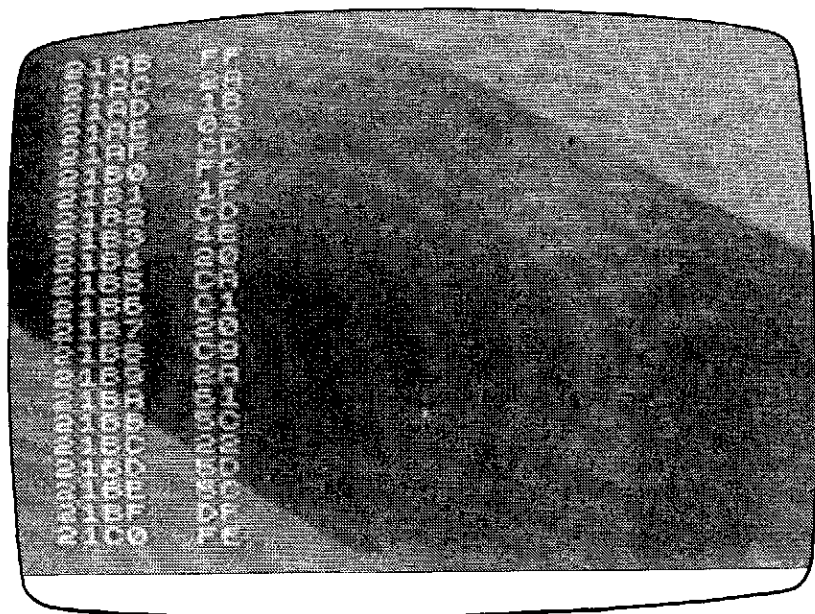


Um programa para ajudá-lo a escrever
os seus próprios programas...



MONITOR

monitor



É um programador utilizador do Código-Máquina ou HEXADECIMAL? Se for achará esta pequena rotina de monitor útil. Ele mostra um endereço em HEXADECIMAL, o conteúdo apenas com a introdução do novo código. Ou pode correr a memória rapidamente, procurando qualquer instrução particular. Se não estiver muito interessado em Código-Máquina, então este programa não é para si. Se gostar de aprender Código-Máquina sugiro que compre o livro *Programação Prática para o Spectrum em Linguagem Máquina*, de Steve Webes, editado por Publicações Europa-América.

Algumas linhas de programa são interessantes — a 90 faz scroll e a 160 e 170 fazem as conversões de HEX para decimal e de decimal para HEX. Não é mais necessário PEEK e POKE nem conversões trabalhosas para e de Hexadecimal.

```

10 REM RUN WITH CAPS LOCK ON
20 BORDER 1: PAPER 1: INK 7: CLS
30 LET H$ = " ": LET A = 0
40 PRINT AT 21, A; FLASH 1; "?"
50 PAUSE 0: LET I$ = INKEY$
60 PRINT AT 21, A; I$: LET H$ = H$ + I$
70 IF A < > 3 THEN LET A = A + 1: GO TO 40
80 GO SUB 160: LET D = X
90 POKE 23692, 255: PRINT AT 21, 31; "□"
100 LET V = D: GO SUB 170
110 PRINT A$; "□ □"; : LET V = PEEK D
120 GO SUB 170
125 PRINT A$ (3 TO); "□ □";
130 PAUSE 0: LET I$ = INKEY$: IF I$ = " " THEN GO TO 130
135 IF I$ = CHR$ 13 THEN GO TO 155
140 PRINT I$;
145 PAUSE 0: LET J$ = INKEY$: IF J$ = " " THEN GO TO 145
150 PRINT J$: LET H$ = I$ + J$: GO SUB 160: POKE D, X
155 LET D = D + 1: GO TO 90
160 LET X = CODE H$ - 48 - 7 * (CODE H$ > 57): FOR N = 2 TO LEN H$:
    LET X = 16 * X + CODE H$ (N) - 48 - 7 * (CODE H$ (N) > 58):
    NEXT N: RETURN
170 DIM A$ (4): FOR N = 1 TO 4: LET A$ (5 - N) =
    CHR$ (V - 16 * INT (V / 16) + 48 + 7 * (V - 16 * INT (V / 16) > 9)):
    LET V = INT (V / 16): NEXT N: RETURN

```

0000	00	00
0001	00	00
0002	00	00
0003	00	00
0004	00	00
0005	00	00
0006	00	00
0007	00	00
0008	00	00
0009	00	00
0010	00	00
0011	00	00
0012	00	00
0013	00	00
0014	00	00
0015	00	00
0016	00	00
0017	00	00
0018	00	00
0019	00	00
0020	00	00
0021	00	00
0022	00	00
0023	00	00
0024	00	00
0025	00	00
0026	00	00
0027	00	00
0028	00	00
0029	00	00
0030	00	00
0031	00	00
0032	00	00
0033	00	00
0034	00	00
0035	00	00
0036	00	00
0037	00	00
0038	00	00
0039	00	00
0040	00	00
0041	00	00
0042	00	00
0043	00	00
0044	00	00
0045	00	00
0046	00	00
0047	00	00
0048	00	00
0049	00	00
0050	00	00
0051	00	00
0052	00	00
0053	00	00
0054	00	00
0055	00	00
0056	00	00
0057	00	00
0058	00	00
0059	00	00
0060	00	00
0061	00	00
0062	00	00
0063	00	00
0064	00	00
0065	00	00
0066	00	00
0067	00	00
0068	00	00
0069	00	00
0070	00	00
0071	00	00
0072	00	00
0073	00	00
0074	00	00
0075	00	00
0076	00	00
0077	00	00
0078	00	00
0079	00	00
0080	00	00
0081	00	00
0082	00	00
0083	00	00
0084	00	00
0085	00	00
0086	00	00
0087	00	00
0088	00	00
0089	00	00
0090	00	00
0091	00	00
0092	00	00
0093	00	00
0094	00	00
0095	00	00
0096	00	00
0097	00	00
0098	00	00
0099	00	00



```

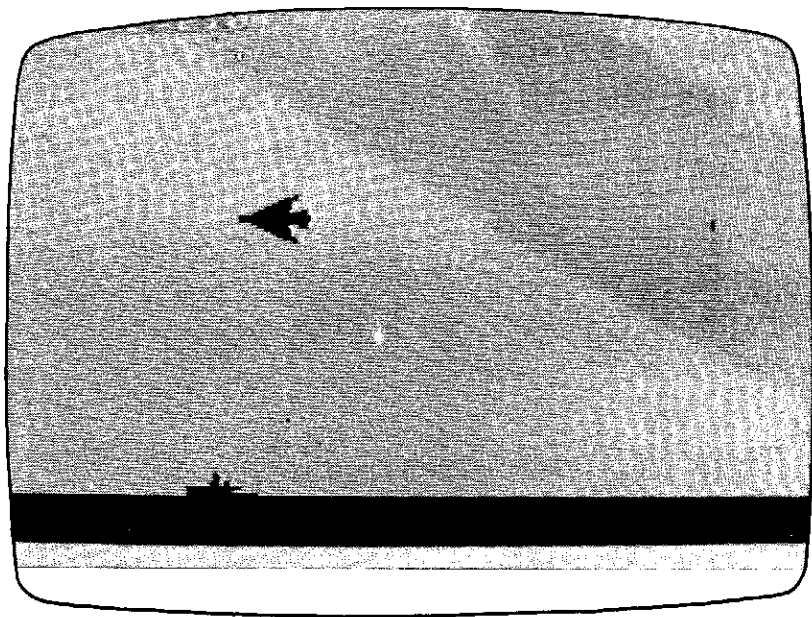
10 REM So funciona no modo de
CAPS LOCK
20 BORDER 1: PAPER 1: INK 7: C
L0
30 LET h#="": LET a=0
40 PRINT AT 21,a: FLASH 1;"?"
50 PAUSE 0: LET i#=INKEY#
60 PRINT AT 21,a;i#: LET h#=h#
+i#
70 IF a<>3 THEN LET a=a+1: GO
TO 40
80 GO SUB 160: LET d=x
90 POKE 23690,255
100 LET v=d: GO SUB 170
110 PRINT a#;" "; LET v=PEEK
d
120 GO SUB 170
130 PRINT a#(3 TO )" ";
140 PAUSE 0: LET j#=INKEY#: IF
j#="" THEN GO TO 130
150 IF j#=CHR# 10 THEN GO TO 16
+
160 PRINT j#
170 PAUSE 0: LET k#=j#
j#="" THEN GO TO 140
180 PRINT j#: LET r#=i#+j#: GO
SUB 160: POKE d,x

```




BOMBARDEIRO

bomber



Este jogo é quase o oposto do *HELICOPTER*. Aqui o objectivo é bombardear o máximo número possível de navios no tempo disponível. Um bónus adicional — quanto mais alto estiver, quando larga a bomba, maior será a pontuação se acertar. Usando as teclas 6, para baixo, 7, para cima e 0 para disparar, tem completo controlo sobre os movimentos e acções do avião.

Devido ao grande número de caracteres gráficos usados neste jogo é aconselhável introduzi-los construindo as linhas 10 a 80 e 905 a 985 e fazendo RUN. Note-se o uso de ATTR neste programa para verificar se o míssil atingiu alguma coisa ou não.

Se estiver interessado em escrever os seus próprios «jogos de acção» é melhor lembrar que ATTR (a, b) pode ser usado para verificar não só a cor do quadrado (a, b) bem como dizer o que está nesse quadrado. Note que a linha 30 usa «valores reais» para mover o avião para cima e para baixo.

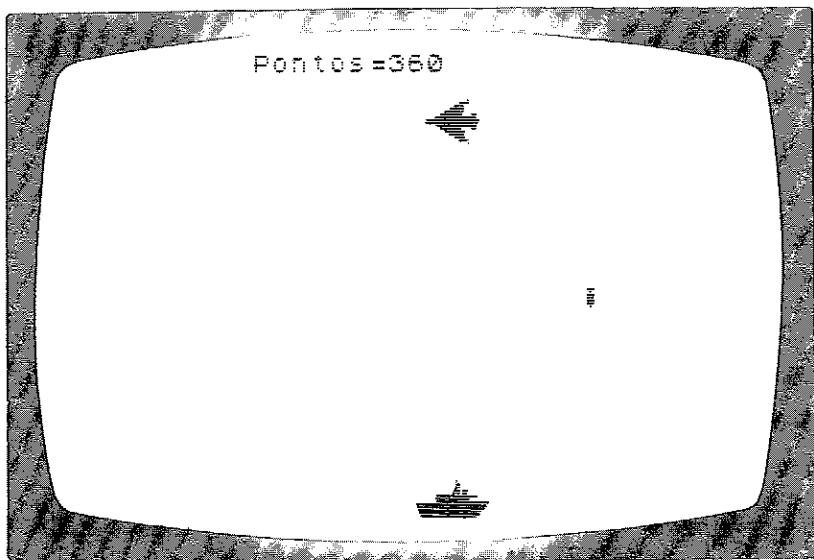
```
10 BORDER 6: INK 0: PAPER 5
20 CLS
30 RESTORE 905
40 FOR a = 0 TO 15: READ a$
50 FOR b = 0 TO 7: READ c
60 POKE b + USR a$, c
70 NEXT b
80 NEXT a
90 FOR a = 20 TO 21: PRINT AT a, 0; PAPER 1; "[32 spaces]"; NEXT a
100 LET p = 0: LET x = 19: LET y = 0: LET u = 2: LET v = 29:
    LET a = 0: LET b = 0: LET s = 0
115 LET p = p + 1
120 PRINT AT x, y; INK 5; "□"; INK 2; "□ gB gC gD"; AT x + 1, y;
    PAPER 1; INK 2; "□ gE gF gF gH"
130 LET u = u + (INKEY$ = "6" AND U < 10) - (INKEY$ = "7"
    AND u > 2): LET v = v - 1
140 IF v = 0 THEN GO SUB 200
150 LET y = y + 1: IF y = 27 THEN GO SUB 300
155 IF INKEY$ = "0" THEN GO SUB 400
157 IF b < > 0 THEN GO SUB 500
160 PRINT AT u - 1, v; "□ □ □ □"; AT u + 2, v; "□ □ □ □";
    AT u, v; "gI gJ gK □"; AT u + 1, v; "gL gM gN □"
165 PRINT AT 0, 0; FLASH 1; PAPER 4; INK 7; "SCORE ="; s
170 IF p < 600 THEN GO TO 115
180 GO TO 800
```

```

200 PRINT AT u - 1, v + 1; "□□□□"; AT u + 2, v + 1; "□□□□";
    AT u, v + 1; "□□□"; AT u + 1, v + 1; "□□□"
210 LET v = 29: LET u = 2: RETURN
300 PRINT AT 19, y - 1; PAPER 5; "□□□□□"; AT 20, y - 1;
    PAPER 1; "□□□□□"
310 LET y = 0: RETURN
400 IF b < > 0 THEN PRINT AT a, b; "□"
405 LET d = u
410 LET a = u: LET b = v + 1: RETURN
500 LET a = a + 1: IF a = 21 THEN LET b = 0: RETURN
503 PRINT AT a - 1, b; "□": IF a = 20 AND ATTR (a, b) < > 15
    THEN RETURN
505 IF ATTR (a, b) = 42 THEN GO TO 530
510 PRINT AT a, b; INK 7; "gP"
515 BEEP .05, -a/10
520 RETURN
530 PRINT AT a, b; ".": FOR z = 0 TO 20: NEXT z: PRINT AT a, b; "gR":
    FOR z = 0 TO 20: NEXT z: PRINT AT a, b; "gS": FOR z = 0 TO 20:
    NEXT z
535 RESTORE 900: FOR z = 0 TO 5: READ c: BEEP .5, c: NEXT z
537 PRINT AT x, y; PAPER 5; "□□□□□"; AT x + 1, y; PAPER 1;
    "□□□□□": LET y = 0
539 RESTORE
540 LET b = 0: LET s = s + (10 * (11 - d)): RETURN
800 FOR a = 0 TO 2: PAUSE 25: BEEP .5, -5: NEXT a: INPUT
    "HIT ENTER TO TRY AGAIN! □"; a$: CLS: GO TO 90
900 DATA 35, 20, 35, 20, 35, 20
905 DATA "b", 0, 0, 0, 0, 0, 63, 63, 63
915 DATA "c", 32, 48, 48, 118, 118, 255, 255, 255
920 DATA "d", 0, 0, 0, 0, 0, 224, 0, 255

```

925 DATA "e", 63, 63, 31, 31, 15, 0, 0, 0
930 DATA "f", 255, 255, 255, 255, 255, 0, 0, 0
935 DATA "h", 254, 252, 248, 240, 224, 0, 0, 0
940 DATA "i", 0, 0, 0, 0, 0, 3, 15, 255
945 DATA "j", 0, 3, 15, 63, 255, 254, 255, 255
950 DATA "k", 64, 192, 0, 0, 0, 24, 248, 252
955 DATA "l", 255, 15, 3, 0, 0, 0, 0, 0
960 DATA "m", 255, 255, 254, 255, 63, 15, 3, 0
965 DATA "n", 252, 248, 24, 0, 0, 0, 192, 64
970 DATA "p", 0, 56, 16, 56, 56, 56, 56, 16
975 DATA "q", 0, 0, 0, 24, 24, 0, 0, 0
980 DATA "r", 0, 0, 16, 56, 28, 8, 0, 0
985 DATA "s", 0, 144, 86, 60, 60, 88, 16, 0



```

10 BORDER 5: INK 0: PAPER 5
20 CLG
30 RESTORE 905
40 FOR a=0 TO 15: READ a#
50 FOR b=0 TO 7: READ c
60 POKE b+USR a#,c
70 NEXT b
80 NEXT a
90 FOR a=20 TO 21: PRINT AT a,
0: PAPER 1):
": NEXT a
100 LET p=0: LET x=19: LET y=0:
LET u=2: LET v=29: LET a=0: LET
b=0: LET s=0
115 LET p=p+1
120 PRINT AT x,y: INK 5: " " : IN
K 2: " " : AT x+1,y: PAPER 1: IN
K 2: " "
130 LET u=u+(INKEY$="8" AND u<1
0)-(INKEY$="7" AND u>2): LET v=v
-1
140 IF v=0 THEN GO SUB 200
150 LET y=y+1: IF y=27 THEN GO
SUB 300
155 IF INKEY$="0" THEN GO SUB 4
00
157 IF b<>0 THEN GO SUB 500
160 PRINT AT u-1,v: " " : AT u+
2,v: " " : AT u,v: " " : AT u+1,
v: " "
165 PRINT AT 0,0: FLASH 1: PAPE
R 4: INK 7: "Pontos=": s
170 IF p<600 THEN GO TO 115
180 GO TO 800
200 PRINT AT u-1,v+1: " " : AT
u+2,v+1: " " : AT u,v+1: " " : AT
u+1,v+1: " "
210 LET v=29: LET u=2: RETURN
300 PRINT AT 19,y-1: PAPER 5: "
" : AT 20,y-1: PAPER 1: " "
310 LET y=0: RETURN
400 IF b<>0 THEN PRINT AT a,b: "
"
405 LET d=0
410 LET a=0: LET b=v+1: RETURN
500 LET a=a+1: IF a=21 THEN LET
b=0: RETURN
505 PRINT AT a-1,b: " " : IF a=20
AND ATTR (a,b)<>15 THEN RETURN
505 IF ATTR (a,b)=42 THEN GO TO
530
510 PRINT AT a,b: INK 7: "5"
515 BEEP .05,-a/10
520 RETURN
530 PRINT AT a,b: ".": FOR z=0 T
O 20: NEXT z: PRINT AT a,b: "+":
FOR z=0 TO 20: NEXT z: PRINT AT
a,b: "*": FOR z=0 TO 20: NEXT z

```

```

535 RESTORE 900: FOR z=0 TO 5:
READ c: BEEP .5,c: NEXT z
537 PRINT AT X,Y: PAPER 5;"
"; AT X+1,Y: PAPER 1;" "; LE
T L=0
539 RESTORE
540 LET b=0: LET s=s+(10*(11-d)
): RETURN
500 FOR a=0 TO 2: PAUSE 25: BEE
P .5,-5: NEXT a: INPUT "ENTER pa
re Jogador de novo! ";a#: CLS : GO
TO 900
9000 DATA 35,20,35,20,35,20
9005 DATA "b",0,0,0,0,0,50,50,50
9010 DATA "c",30,40,40,110,110,2
50,255
9020 DATA "d",0,0,0,0,0,224,0,25
5
9025 DATA "e",50,50,31,31,15,0,0
,0
9030 DATA "f",255,255,255,255,25
5,0,0,0
9035 DATA "h",254,252,248,240,22
4,0,0,0
9040 DATA "i",0,0,0,0,0,3,15,255
9045 DATA "j",0,0,15,50,250,204,
255
9050 DATA "k",64,100,0,0,0,24,24
0,255
9055 DATA "l",255,15,0,0,0,0,0,0
9060 DATA "m",209,209,204,209,50
,250,0,0
9065 DATA "n",252,240,24,0,0,0,1
0,250
9070 DATA "o",0,50,10,50,50,50,5
0,5
9075 DATA "p",0,0,0,24,24,0,0,0
9080 DATA "q",0,0,10,50,20,0,0,0
9085 DATA "r",0,144,0,0,0,0,0,0
9090 DATA "s",0,144,0,0,0,0,0,0
9095 DATA "t",0,144,0,0,0,0,0,0
9100 DATA "u",0,144,0,0,0,0,0,0
9105 DATA "v",0,144,0,0,0,0,0,0
9110 DATA "w",0,144,0,0,0,0,0,0
9115 DATA "x",0,144,0,0,0,0,0,0
9120 DATA "y",0,144,0,0,0,0,0,0
9125 DATA "z",0,144,0,0,0,0,0,0
9130 DATA "0",0,144,0,0,0,0,0
9135 DATA "1",0,144,0,0,0,0,0
9140 DATA "2",0,144,0,0,0,0,0
9145 DATA "3",0,144,0,0,0,0,0
9150 DATA "4",0,144,0,0,0,0,0
9155 DATA "5",0,144,0,0,0,0,0
9160 DATA "6",0,144,0,0,0,0,0
9165 DATA "7",0,144,0,0,0,0,0
9170 DATA "8",0,144,0,0,0,0,0
9175 DATA "9",0,144,0,0,0,0,0
9180 DATA "A",0,144,0,0,0,0,0
9185 DATA "B",0,144,0,0,0,0,0
9190 DATA "C",0,144,0,0,0,0,0
9195 DATA "D",0,144,0,0,0,0,0
9200 DATA "E",0,144,0,0,0,0,0
9205 DATA "F",0,144,0,0,0,0,0
9210 DATA "G",0,144,0,0,0,0,0
9215 DATA "H",0,144,0,0,0,0,0
9220 DATA "I",0,144,0,0,0,0,0
9225 DATA "J",0,144,0,0,0,0,0
9230 DATA "K",0,144,0,0,0,0,0
9235 DATA "L",0,144,0,0,0,0,0
9240 DATA "M",0,144,0,0,0,0,0
9245 DATA "N",0,144,0,0,0,0,0
9250 DATA "O",0,144,0,0,0,0,0
9255 DATA "P",0,144,0,0,0,0,0
9260 DATA "Q",0,144,0,0,0,0,0
9265 DATA "R",0,144,0,0,0,0,0
9270 DATA "S",0,144,0,0,0,0,0
9275 DATA "T",0,144,0,0,0,0,0
9280 DATA "U",0,144,0,0,0,0,0
9285 DATA "V",0,144,0,0,0,0,0
9290 DATA "W",0,144,0,0,0,0,0
9295 DATA "X",0,144,0,0,0,0,0
9300 DATA "Y",0,144,0,0,0,0,0
9305 DATA "Z",0,144,0,0,0,0,0
9310 DATA "0",0,144,0,0,0,0,0
9315 DATA "1",0,144,0,0,0,0,0
9320 DATA "2",0,144,0,0,0,0,0
9325 DATA "3",0,144,0,0,0,0,0
9330 DATA "4",0,144,0,0,0,0,0
9335 DATA "5",0,144,0,0,0,0,0
9340 DATA "6",0,144,0,0,0,0,0
9345 DATA "7",0,144,0,0,0,0,0
9350 DATA "8",0,144,0,0,0,0,0
9355 DATA "9",0,144,0,0,0,0,0
9360 DATA "A",0,144,0,0,0,0,0
9365 DATA "B",0,144,0,0,0,0,0
9370 DATA "C",0,144,0,0,0,0,0
9375 DATA "D",0,144,0,0,0,0,0
9380 DATA "E",0,144,0,0,0,0,0
9385 DATA "F",0,144,0,0,0,0,0
9390 DATA "G",0,144,0,0,0,0,0
9395 DATA "H",0,144,0,0,0,0,0
9400 DATA "I",0,144,0,0,0,0,0
9405 DATA "J",0,144,0,0,0,0,0
9410 DATA "K",0,144,0,0,0,0,0
9415 DATA "L",0,144,0,0,0,0,0
9420 DATA "M",0,144,0,0,0,0,0
9425 DATA "N",0,144,0,0,0,0,0
9430 DATA "O",0,144,0,0,0,0,0
9435 DATA "P",0,144,0,0,0,0,0
9440 DATA "Q",0,144,0,0,0,0,0
9445 DATA "R",0,144,0,0,0,0,0
9450 DATA "S",0,144,0,0,0,0,0
9455 DATA "T",0,144,0,0,0,0,0
9460 DATA "U",0,144,0,0,0,0,0
9465 DATA "V",0,144,0,0,0,0,0
9470 DATA "W",0,144,0,0,0,0,0
9475 DATA "X",0,144,0,0,0,0,0
9480 DATA "Y",0,144,0,0,0,0,0
9485 DATA "Z",0,144,0,0,0,0,0
9490 DATA "0",0,144,0,0,0,0,0
9495 DATA "1",0,144,0,0,0,0,0
9500 DATA "2",0,144,0,0,0,0,0
9505 DATA "3",0,144,0,0,0,0,0
9510 DATA "4",0,144,0,0,0,0,0
9515 DATA "5",0,144,0,0,0,0,0
9520 DATA "6",0,144,0,0,0,0,0
9525 DATA "7",0,144,0,0,0,0,0
9530 DATA "8",0,144,0,0,0,0,0
9535 DATA "9",0,144,0,0,0,0,0
9540 DATA "A",0,144,0,0,0,0,0
9545 DATA "B",0,144,0,0,0,0,0
9550 DATA "C",0,144,0,0,0,0,0
9555 DATA "D",0,144,0,0,0,0,0
9560 DATA "E",0,144,0,0,0,0,0
9565 DATA "F",0,144,0,0,0,0,0
9570 DATA "G",0,144,0,0,0,0,0
9575 DATA "H",0,144,0,0,0,0,0
9580 DATA "I",0,144,0,0,0,0,0
9585 DATA "J",0,144,0,0,0,0,0
9590 DATA "K",0,144,0,0,0,0,0
9595 DATA "L",0,144,0,0,0,0,0
9600 DATA "M",0,144,0,0,0,0,0
9605 DATA "N",0,144,0,0,0,0,0
9610 DATA "O",0,144,0,0,0,0,0
9615 DATA "P",0,144,0,0,0,0,0
9620 DATA "Q",0,144,0,0,0,0,0
9625 DATA "R",0,144,0,0,0,0,0
9630 DATA "S",0,144,0,0,0,0,0
9635 DATA "T",0,144,0,0,0,0,0
9640 DATA "U",0,144,0,0,0,0,0
9645 DATA "V",0,144,0,0,0,0,0
9650 DATA "W",0,144,0,0,0,0,0
9655 DATA "X",0,144,0,0,0,0,0
9660 DATA "Y",0,144,0,0,0,0,0
9665 DATA "Z",0,144,0,0,0,0,0
9670 DATA "0",0,144,0,0,0,0,0
9675 DATA "1",0,144,0,0,0,0,0
9680 DATA "2",0,144,0,0,0,0,0
9685 DATA "3",0,144,0,0,0,0,0
9690 DATA "4",0,144,0,0,0,0,0
9695 DATA "5",0,144,0,0,0,0,0
9700 DATA "6",0,144,0,0,0,0,0
9705 DATA "7",0,144,0,0,0,0,0
9710 DATA "8",0,144,0,0,0,0,0
9715 DATA "9",0,144,0,0,0,0,0
9720 DATA "A",0,144,0,0,0,0,0
9725 DATA "B",0,144,0,0,0,0,0
9730 DATA "C",0,144,0,0,0,0,0
9735 DATA "D",0,144,0,0,0,0,0
9740 DATA "E",0,144,0,0,0,0,0
9745 DATA "F",0,144,0,0,0,0,0
9750 DATA "G",0,144,0,0,0,0,0
9755 DATA "H",0,144,0,0,0,0,0
9760 DATA "I",0,144,0,0,0,0,0
9765 DATA "J",0,144,0,0,0,0,0
9770 DATA "K",0,144,0,0,0,0,0
9775 DATA "L",0,144,0,0,0,0,0
9780 DATA "M",0,144,0,0,0,0,0
9785 DATA "N",0,144,0,0,0,0,0
9790 DATA "O",0,144,0,0,0,0,0
9795 DATA "P",0,144,0,0,0,0,0
9800 DATA "Q",0,144,0,0,0,0,0
9805 DATA "R",0,144,0,0,0,0,0
9810 DATA "S",0,144,0,0,0,0,0
9815 DATA "T",0,144,0,0,0,0,0
9820 DATA "U",0,144,0,0,0,0,0
9825 DATA "V",0,144,0,0,0,0,0
9830 DATA "W",0,144,0,0,0,0,0
9835 DATA "X",0,144,0,0,0,0,0
9840 DATA "Y",0,144,0,0,0,0,0
9845 DATA "Z",0,144,0,0,0,0,0
9850 DATA "0",0,144,0,0,0,0,0
9855 DATA "1",0,144,0,0,0,0,0
9860 DATA "2",0,144,0,0,0,0,0
9865 DATA "3",0,144,0,0,0,0,0
9870 DATA "4",0,144,0,0,0,0,0
9875 DATA "5",0,144,0,0,0,0,0
9880 DATA "6",0,144,0,0,0,0,0
9885 DATA "7",0,144,0,0,0,0,0
9890 DATA "8",0,144,0,0,0,0,0
9895 DATA "9",0,144,0,0,0,0,0
9900 DATA "A",0,144,0,0,0,0,0
9905 DATA "B",0,144,0,0,0,0,0
9910 DATA "C",0,144,0,0,0,0,0
9915 DATA "D",0,144,0,0,0,0,0
9920 DATA "E",0,144,0,0,0,0,0
9925 DATA "F",0,144,0,0,0,0,0
9930 DATA "G",0,144,0,0,0,0,0
9935 DATA "H",0,144,0,0,0,0,0
9940 DATA "I",0,144,0,0,0,0,0
9945 DATA "J",0,144,0,0,0,0,0
9950 DATA "K",0,144,0,0,0,0,0
9955 DATA "L",0,144,0,0,0,0,0
9960 DATA "M",0,144,0,0,0,0,0
9965 DATA "N",0,144,0,0,0,0,0
9970 DATA "O",0,144,0,0,0,0,0
9975 DATA "P",0,144,0,0,0,0,0
9980 DATA "Q",0,144,0,0,0,0,0
9985 DATA "R",0,144,0,0,0,0,0
9990 DATA "S",0,144,0,0,0,0,0
9995 DATA "T",0,144,0,0,0,0,0

```

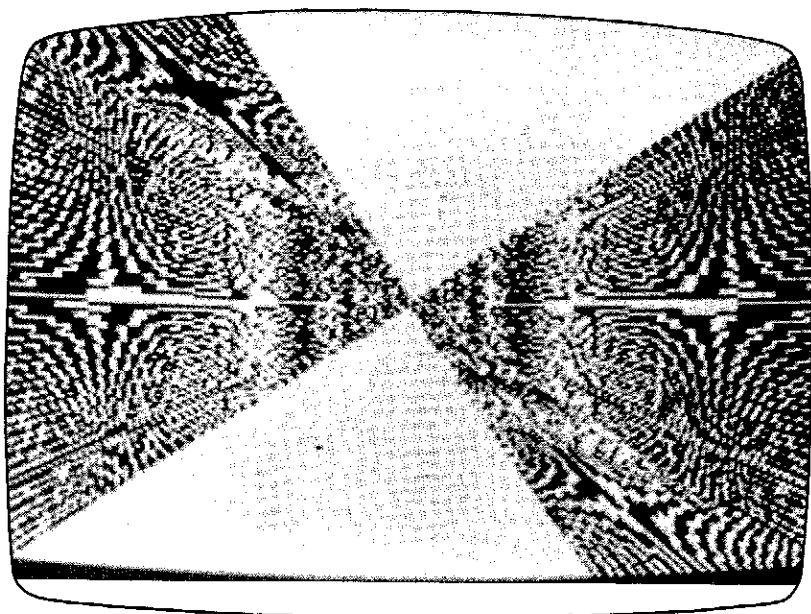


Hmmm... Spectrum... Spectrum...
deve existir um nome melhor



CALEIDOSCÓPIO

kaleidoscope



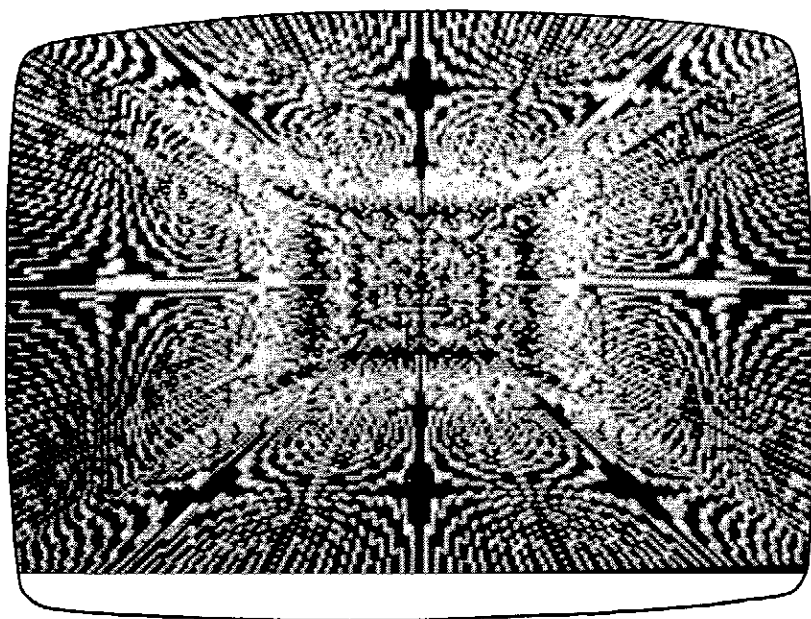
Este programa é uma demonstração fabulosa das capacidades gráficas do Spectrum. Os modelos gerados por *KALEIDOSCOPE* baseiam-se no facto de que o Spectrum *não pode* desenhar linhas rectas! É usado o comando *OVER 1*, de modo que quando qualquer ponto é desenhado sobre o outro, eles desaparecem. Estas duas características permitem *DESENHAR EM CIMA* outras linhas sobre as que o Spectrum já desenhou, conseguindo-se efeitos espantosos. Eu ia chamar a este programa de *OVERDRAWN*¹ mas lembrei-me da minha conta bancária!

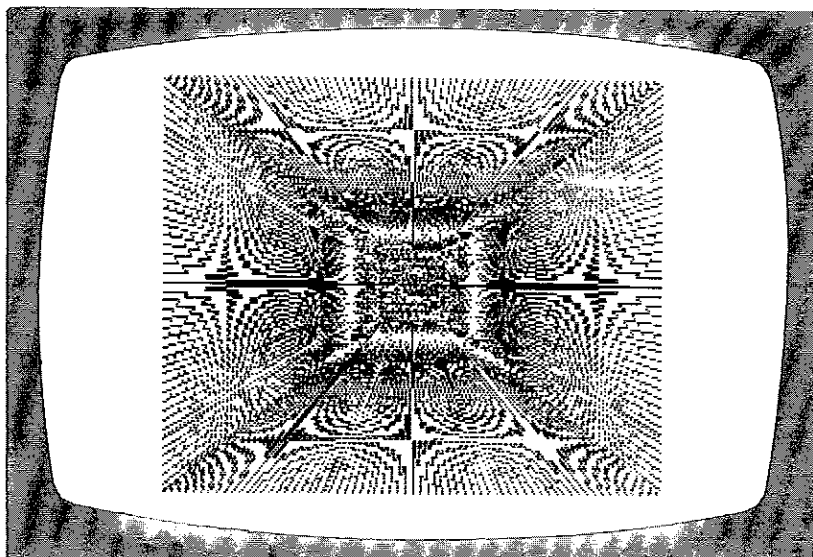
¹ Sacar a descoberto (*N. T.*)

```

10 INK 1 + RND * 7: PAPER 0: BORDER 0: OVER 1: CLS
15 LET R = 1 + .5 * RND: FOR A = 1 TO 22: PRINT "g8c [32 times]":
    NEXT A
20 FOR A = -87 TO 87 STEP R
30 BEEP .02, A/3: PLOT 128, 88: DRAW 127, -A: PLOT 128, 88:
    DRAW -128, A
40 NEXT A
50 FOR A = 127 TO -127 STEP -R
60 BEEP .02, A/3: PLOT 128, 88: DRAW A, -88: PLOT 128, 88:
    DRAW -A, 87
70 NEXT A

```



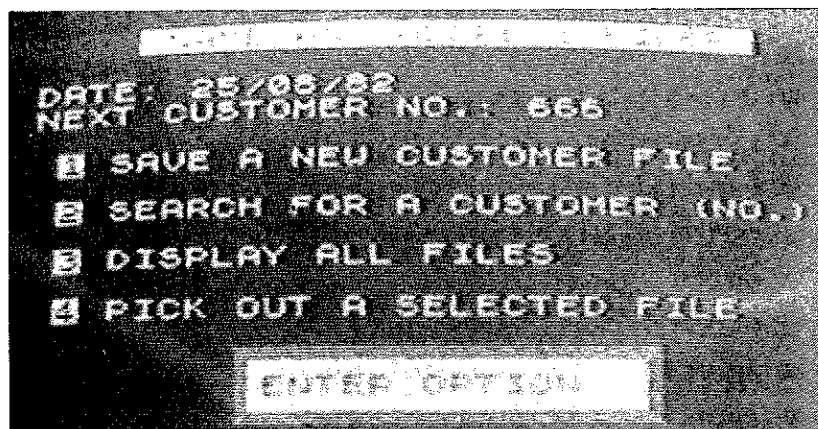


```
10 INK 2: PAPER 5: BORDER 0: 0
VER 1: CLS
15 LET r=1+.5*RND: FOR a=1 TO
22: PRINT "
": NEXT a
20 FOR a=-87 TO 87 STEP r
30 BEEP .02,a/3: PLOT 128,88:
DRAW 127,-a: PLOT 128,88: DRAW -
128,a
40 NEXT a
50 FOR a=127 TO -127 STEP -r
60 BEEP .02,a/3: PLOT 128,88:
DRAW a,-88: PLOT 128,88: DRAW -a
,87
70 NEXT a
```




CLIENTE

customer



Em rigor não se trata de um jogo fornecendo, apesar disso, uma interessante demonstração de como o computador pode ser usado para manipular «arquivos». Como poderei explicar o que é um arquivo? Um arquivo, como nós o entendemos, é uma caixa onde é guardada a informação. De modo semelhante o Spectrum pode guardar informação em *cassette*, usando arquivos de dados. Neste exemplo (que pode ser facilmente modificado para qualquer objectivo) um arquivo de 128 *bytes* é guardado em *cassette* com a seguinte informação: nome, morada, número de telefone, ordem de compra, notas e data da ordem. Apesar de ter sido desenhado para uma pequena empresa, pode-se usá-lo para guardar números de telefone e moradas de amigos e conhecidos.

NOTA: F\$ (128) será preparado no programa para guardar 8 «campos». Cada campo é referenciado a uma parte do arquivo, isto é, o campo 2 é a 1.^a linha de morada, por exemplo. A entrada do programa é directa.

Ao usar o programa, os dados são GUARDADOS no arquivo automaticamente e quando for pedido «NEXT CUSTOMER NO», (número do próximo cliente).: se apenas carregar em «ENTER» ficará a saber o número seguinte àquele que introduziu.

```
10 LET D = 0: INK 7: PAPER 1: BORDER 1
15 CLS: DIM F$ (128)
20 PRINT TAB 4; BRIGHT 1; "g8c NAME AND ADDRESS FILER g8c"
30 IF D = 0 THEN PRINT "DATE: □";
40 IF D = 0 THEN INPUT "Enter date (DD/MM/YY): □"; D$
45 LET F$ (121 TO ) = D$
50 PRINT F$ (121 TO ) ' "NEXT CUSTOMER NO.: □";
60 INPUT "Number: □"; C$: IF C$ = " " THEN LET C$ = STR$ CN
65 LET CN = VAL C$
70 PRINT CN ' ' TAB 1; PAPER 6; INK 0; "1";
80 PRINT "□ SAVE A NEW CUSTOMER FILE"
90 PRINT ' TAB 1; PAPER 6; INK 0; "2";
100 PRINT "□ SEARCH FOR A CUSTOMER (NO.)"
110 PRINT ' TAB 1; PAPER 6; INK 0; "3";
120 PRINT "□ DISPLAY ALL FILES"
130 PRINT ' TAB 1; PAPER 6; INK 0; "4";
140 PRINT "□ PICK OUT A SELECTED FILE"
150 PRINT INK 2; PAPER 6; AT 13, 8; "g4c g3 [14 times] g7";
    AT 14, 8; "g5c ENTER OPTION □ □ g5";
    AT 15, 8; "g1c g3c [14 times] g2c"
160 PRINT AT 14, 22; INK 4; PAPER 7; "?"
165 PAUSE 15
170 IF INKEY$ < > " " THEN GO TO 210
180 PRINT AT 14, 22; INK 4; PAPER 7; "□"
185 PAUSE 15
```

```

190 IF INKEY$ <> " " THEN GO TO 210
200 GO TO 160
210 LET IS = INKEY$
220 IF IS < "1" OR IS > "4" THEN GO TO 160
225 LET D = 1
230 CLS
240 GO TO 1000 * VAL IS
1000 RESTORE 9000: PRINT "CREATE A NEW FILE" ' "g3 [17 times]"
1005 LET A$ = "No □" + STR$ CN
1010 LET A = USR "A"
1020 FOR B = 0 TO 7
1030 READ C: POKE A + B, C
1040 NEXT B
1050 LET A = USR "B"
1060 FOR B = 0 TO 7
1070 READ C: POKE A + B, C
1080 NEXT B
1085 PRINT "CUSTOMER NO.: □"; CN
1090 PRINT ' "NAME:" ' TAB 9; "gA [20 times] gB" '
      "ADDRESS:" ' TAB 9; "gA [20 times] gB" ' ' TAB 9; "gA [20 times] gB"
      ' ' TAB 9; "gA [20 times] gB"
1100 PRINT "PHONE:" ' TAB 9; "gA [15 times] gB"
1110 PRINT "ORDER:" ' TAB 9; "gA gA gA gB" ' "NOTES:" ' TAB 9;
      "gA [22 times] gB"
1120 RESTORE 9020
1125 LET T = 0: LET L = 8
1130 FOR A = 0 TO 6
1135 LET T = T + L - 8
1140 READ L
1150 LET R = 5 + 2 * A: LET C = 9

```

```

1160 GO SUB 9500
1170 IF CODE I$ = 12 THEN PRINT AT R, C; "□": LET C = C - 1:
      GO TO 1160
1180 IF CODE I$ = 13 THEN GO TO 1230
1190 LET F$(T + C - 8) = I$
1200 IF C = L THEN GO TO 1230
1210 LET C = C + 1
1220 GO TO 1160
1230 NEXT A
1240 SAVE A$ DATA F$ ( )
1250 LET CN = CN + 1
1260 GO TO 15
2000 INPUT "ENTER CUSTOMER NUMBER: □"; N
2010 PRINT "□ SEARCHING FOR CUSTOMER □"; FLASH 1; N
2020 LOAD "No □" + STR$ N DATA F$ ( )
2040 GO SUB 9100
2050 IF INKEY$ = " " THEN GO TO 2050
2060 GO TO 15
3000 PRINT "□ SEARCHING"
3005 INK 1: PAPER 7
3010 LOAD " " DATA F$ ( )
3020 CLS: PRINT "NAME: □"; F$ (TO 20) ' "ADDRESS: □";
      F$ (21 TO 40) ' TAB 9; F$ (41 TO 60) ' TAB 9; F$ (61 TO 80) '
      "PHONE: □"; F$ (81 TO 95) ' "ORDER: □"; F$ (96 TO 98) '
      "NOTES: □"; F$ (99 TO 120) ' "DATE OF ORDER: □";
      F$ (121 TO 128)
3030 GO TO 3010
4000 PRINT "□ WHICH FIELD? (1 TO 8)"
4005 INPUT "FIELD: □"; F
4010 RESTORE 9020

```

```

4015 LET T = 0
4020 FOR A = 1 TO F: READ L: LET T = T + L - 8: NEXT A
4030 DIM T$(L - 8)
4040 PRINT "ENTER FIELD "; F; "TEXT"
4050 INPUT T$(TO L - 8)
4060 PRINT "SEARCHING FOR FIELD "; FLASH 1; F; FLASH 0;
    ' ' TAB 1; FLASH 1; T$
4070 LOAD " " DATA F$ ( )
4080 IF F$ (T - L + 9 TO T) < > T$ THEN GO TO 4090
4085 CLS: PRINT "NAME: "; F$ (TO 20) ' "ADDRESS: ";
    F$ (21 TO 40) ' TAB 9; F$ (41 TO 60) ' TAB 9; F$ (61 TO 80) '
    "PHONE: "; F$ (81 TO 95) ' "ORDER: "; F$ (96 TO 98) '
    "NOTES: "; F$ (99 TO 120) ' "DATE OF ORDER: ";
    F$ (121 TO 128)
4090 GO TO 4070
8999 STOP
9000 DATA 128, 255, 0, 0; 0, 0, 0, 0
9010 DATA 128, 128, 0, 0, 0, 0, 0, 0
9020 DATA 28, 28, 28, 28, 23, 11, 35
9100 CLS: PRINT "CUSTOMER NUMBER "; N ' "NAME: ";
    F$ (TO 20) ' "ADDRESS: "; F$ (21 TO 40) ' TAB 9;
    F$ (41 TO 60) ' TAB 9; F$ (61 TO 80) ' "PHONE: "; F$ (81 TO 95) '
    "ORDER: "; F$ (96 TO 98) ' "NOTES: "; F$ (99 TO 120) '
    "DATE OF ORDER: "; F$ (121 TO 128)
9110 RETURN
9500 PRINT AT R, C; INK 6; "*"
9510 PAUSE 15
9520 LET I$ = INKEY$: IF I$ < > " " THEN GO TO 9560
9530 PRINT AT R, C; " "
9540 PAUSE 15

```

```

9550 LET IS = INKEY$: IF IS <> " " THEN GO TO 9560
9555 GO TO 9500
9560 IF CODE IS = 13 THEN PRINT AT R, C; "□"
9565 PRINT AT R, C; IS
9570 RETURN

```

CRIAR NOVO FICHEIRO

QUADRO No.: 1

NOME Basilio Vieira

ENDERECO: Rua A Lote 11/12

Linho

Sintra

TELEFONE: -----

ENCOMENDAS 5

NOTAS: Transporte Pago



```

10 LET d=0: INK 7: PAPER 1: BO
RDER 1
15 CLS : DIM f$(128)
20 PRINT TAB 4; BRIGHT 1;"FICH
EIRO NUMERO E ABERTO"
30 IF d=0 THEN PRINT "DATA: ";
40 IF d=0 THEN INPUT "Data(Dia
/Mes/Ano): ";d$
45 LET f$(121 TO )=d$
50 PRINT f$(121 TO )"Proximo
quadro No.:";
60 INPUT "Numero: ";c$: IF c#=
"" THEN LET c#=STR$ cn

```

```

65 LET cn=VAL c$
70 PRINT cn;"TAB 1; PAPER 6; I
NK 0;"1";
80 PRINT " SAVE FICHEIRO DO NO
VO QUADRO"
90 PRINT "TAB 1; PAPER 6; INK
0;"2";
100 PRINT " BUSCA DO QUADRO (No
.)"
110 PRINT "TAB 1; PAPER 6; INK
0;"3";
120 PRINT " APRESENTACAO DE TOD
OS OS FICHEIROS"
130 PRINT "TAB 1; PAPER 6; INK
0;"4";
140 PRINT " APRESENTAR O FICHEI
RO ESCOLHIDO"
150 PRINT INK 2; PAPER 6; AT 13,
8; "-----"; AT 14,8; "
Entre opcao "; AT 15,8; "-----"
160 PRINT AT 14,22; INK 4; PAPE
R 7;"?"
165 PAUSE 15
170 IF INKEY#<>" " THEN GO TO 21
0
180 PRINT AT 14,22; INK 4; PAPE
R 7;" "
185 PAUSE 15
190 IF INKEY#<>" " THEN GO TO 21
0
200 GO TO 160.
210 LET i#=INKEY#
220 IF i#<"1" OR i#>"4" THEN GO
TO 160
225 LET d=1
230 CLS
240 GO TO 1000*VAL i$
1000 RESTORE 9000: PRINT "CRIAR
NOVO FICHEIRO"
1005 LET a$="No. "+STR$ cn
1010 LET a=USR "a"
1020 FOR b=0 TO 7
1030 READ c: POKE a+b,c
1040 NEXT b
1050 LET a=USR "b"
1060 FOR b=0 TO 7
1070 READ c: POKE a+b,c
1080 NEXT b
1085 PRINT "QUADRO No. : ";cn
1090 PRINT "NOME:";TAB 9;"
-----"
1095 PRINT "ENDERECO:";TAB
9;"-----"
1100 PRINT "TELEFONE:";TAB 9;"-----"

```



```

1110 PRINT "ENCOMENDA:" TAB 9;"
1120 PRINT "NOTAS:" TAB 9;"
1120 RESTORE 9020
1125 LET t=0: LET l=8
1130 FOR a=0 TO 6
1135 LET t=t+l-8
1140 READ l
1150 LET r=5+2*a: LET c=9
1160 GO SUB 9500
1170 IF CODE i#=12 THEN PRINT AT
r,c;" ": LET c=c-1: GO TO 1150
1180 IF CODE i#=13 THEN GO TO 12
30
1190 LET f#(t+c-8)=i#
1200 IF c=l THEN GO TO 1230
1210 LET c=c+1
1220 GO TO 1150
1230 NEXT a
1240 SAVE a# DATA f#()
1250 LET cn=cn+1
1260 GO TO 15
2000 INPUT "No. DO QUADRO: ";n
2010 PRINT " PROCURANDO PELO QU
ADRO "; FLASH 1;n
2020 LOAD "No. "+STR$ n DATA f#(
)
2040 GO SUB 9100
2050 IF INKEY#="" THEN GO TO 205
0
2060 GO TO 15
3000 PRINT " BUSCA "
3005 INK 1: PAPER 7
3010 LOAD "" DATA f#()
3020 CLS : PRINT "NOME: ";f#(1
0 TO 20)"; "MORADA: ";f#(21 TO 40)";T
AB 9;f#(41 TO 60)";TAB 9;f#(61 TO
80)"; "TELEFONE: ";f#(81 TO 95)";
"ENCOMENDA: ";f#(96 TO 98)"; "NO
TAS: ";f#(99 TO 120)"; "DATA DA E
NCOMENDA: ";f#(121 TO 128)
3030 GO TO 3010
4000 PRINT " QUE CAMPO? (1 a 8)
"
4005 INPUT "CAMPO: ";f
4010 RESTORE 9020
4015 LET t=0
4020 FOR a=1 TO f: READ l: LET t
=t+l-8: NEXT a
4030 DIM t#(l-8)
4040 PRINT "ENTRE CAMPO: ";f;"
TEXTO"
4050 INPUT t#(1 TO l-8)
4060 PRINT "PROCURANDO CAMPO";
FLASH 1;f: FLASH 0;"TAB 1; FLAS
H 1;t#
4070 LOAD "" DATA f#()

```

```

4080 IF f#(t-l+9 TO t)<>t# THEN
GO TO 4090
4085 CLS : PRINT "NOME: "; f#(T
O 20)""MORADA: "; f#(21 TO 40)"T
AB 9; f#(41 TO 60)"TAB 9; f#(61 TO
80)""TELEFONE: "; f#(81 TO 95)"
"ENCOMENDA: "; f#(96 TO 98)""NO
TAB: "; f#(99 TO 120)""DATA DA E
NCOMENDA: "; f#(121 TO 128)
4090 GO TO 4070
8999 STOP
9000 DATA 128,255,0,0,0,0,0,0
9010 DATA 128,128,0,0,0,0,0,0
9020 DATA 20,20,20,20,20,20,11,05
9100 CLS : PRINT "NOME DO QUADR
O: "; r""NOME: "; f#( TO 20)""MO
RADA: "; f#(21 TO 40)"TAB 9; f#(41
TO 60)"TAB 9; f#(61 TO 80)""TEL
EFONE: "; f#(81 TO 95)""ENCOMEND
A: "; f#(96 TO 98)""NOTAS: "; f#(
99 TO 120)""DATA DA ENCOMENDA:
"; f#(121 TO 128)
9110 RETURN
9500 PRINT AT r,c; INK 6;"*"
9510 PAUSE 15
9520 LET i#=INKEY#: IF i#<>"" TH
EN GO TO 9550
9530 PRINT AT r,c;" "
9540 PAUSE 15
9550 LET i#=INKEY#: IF i#<>"" TH
EN GO TO 9550
9555 GO TO 9500
9600 IF CODE i#-13 THEN PRINT AT
r,c;" "
9665 PRINT AT r,c;i#
9670 RETURN

```

NOTAS



ESPIRAL

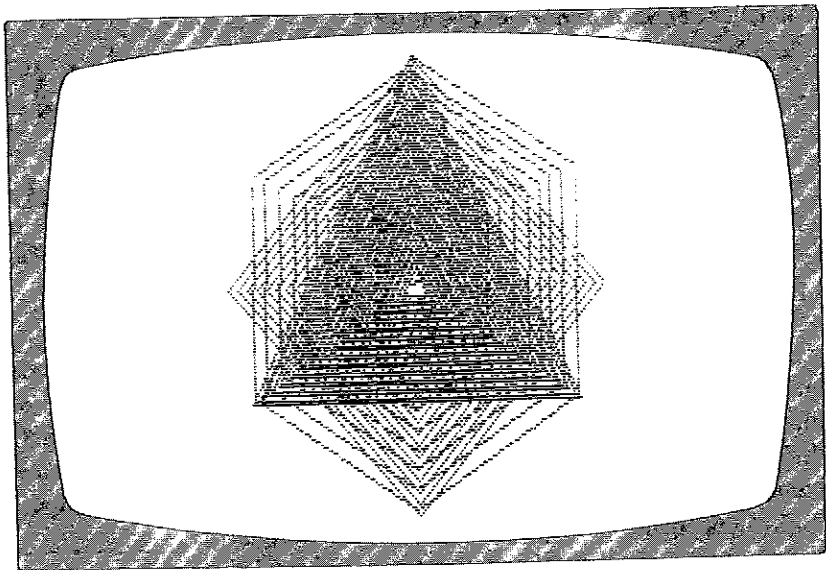
spiral

Este interessante programa reproduz muitas formas de desenhos magníficos. Não subestime este programa pelo seu tamanho. Introduza-o e prima RUN. Introduza os seus próprios valores; tente 3, 4, 6 e 8 para ver como funciona — prima V para limpar o écran e recomece, ou outra tecla para começar sem limpar o écran. Agora tente 3, 5, 6.66666666, 2.9, 4.95 ou outros valores quaisquer.

Pergunta: Qual a diferença entre 4.95 e 4.05?

Veja se consegue descobrir por que é que isso acontece.

```
10 BORDER 0: PAPER 0: INK 1 + RND * 7: CLS
15 LET r = 80
20 INPUT n
25 LET x = 128: LET y = r + 85
30 PLOT x, y
40 FOR b = 0 TO 1000
50 LET a = PI * 2 / n * b
55 LET x1 = 128 + SIN a * r: LET y1 = 85 + COS a * r
60 DRAW x1 - x, y1 - y: BEEP . 05, (80 - R) / 3
65 LET r = r - 1: IF r < 5 THEN GO TO 80
70 LET x = x1: LET y = y1
75 NEXT b
80 LET a$ = INKEY$: IF a$ = "" THEN GO TO 80
95 IF a$ = "v" THEN RUN
100 GO TO 15
```



```

10 BORDER 5: PAPER 7, INK 1+RN
D*7: OLS
15 LET r=88
20 INPUT n
35 LET x=128: LET y=r+85
40 PLOT X,Y
45 FOR b=0 TO 1000
50 LET a=PI*2/n*b
55 LET x1=128+31N a*r: LET y1=
85+COS a*r
60 DRAW x1-x,y1-y: BEEP .05,(3
0-(r)/3
65 LET r=r-1: IF r<5 THEN GO T
0 80
70 LET x=x1: LET y=y1
75 NEXT b
80 LET a#=INKEY$: IF a#="" THE
N GO TO 80
95 IF a#="V" THEN RUN
100 GO TO 15

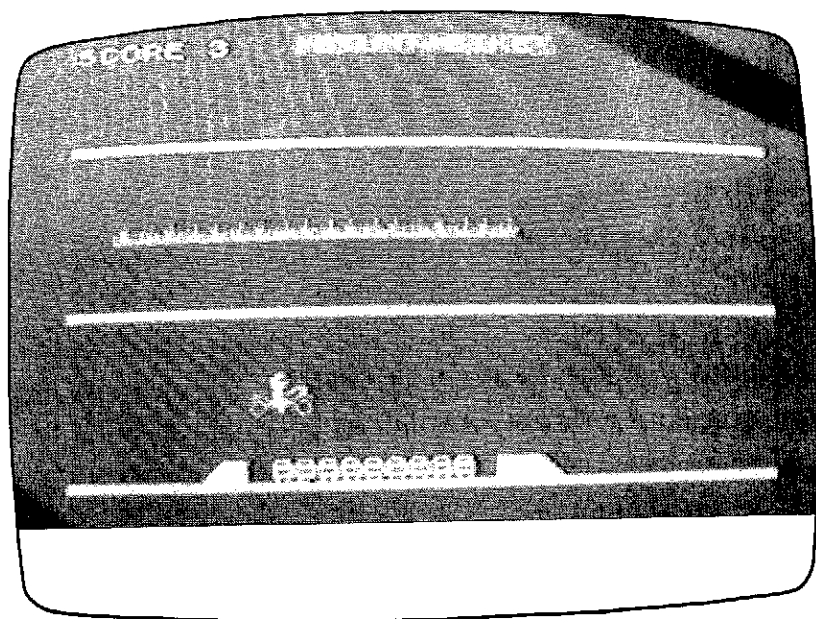
```

Conseguirá superar o salto de Evel Knievel?

12

MOTO SUICIDA

stunt bike



STUNT BIKE é um jogo de perícia onde você corre através de túneis de fogo, salta por cima de autocarros e rodopia enquanto acelera a fundo. Use a tecla 1 para travar e a \emptyset para acelerar. Há dois riscos específicos — não tente rodopiar no tunel de fogo ou cairá e, além disso, deve tirar a mão do acelerador imediatamente antes de deixar a rampa ou então perderá altura!

Como em relação a outros jogos introduza primeiro a parte gráfica entrando com os caracteres gráficos nas linhas 105, 110, 130, 150, 155, 160 e 240.

São usadas, neste programa, inúmeras técnicas que nunca deve ter visto. A instrução GOTO na lista 155 salta para o 170 se não tocar no tecto do túnel de fogo ou para a 300 se tocar. O comando LET na linha 170 usa valores lógicos de \emptyset e 1 para alterar a aceleração da moto

de acordo com as teclas que estiverem a ser premidas. Finalmente as linhas mais invulgares são a 272 e a 300. A instrução PRINT = 1 escreve nas linhas de baixo tal como PRINT = 0, PRINT = 2 escreve normalmente e PRINT = 3 envia os resultados para a impressora ZX Printer. Isto é o mecanismo «stream» do ZX Spectrum e o comando PRINT = foi criado para ser usado com a placa do «interface» RS 232.

Se desejar modificar a função de algum destes comandos pode usar OPEN # n, a\$, onde n é um número de 0 a 15 e a\$ é K, S ou P conforme o periférico que estiver ligado for um teclado, um écran ou uma impressora. Se não está muito certo de como tudo isto funciona nada melhor que experimentar para ver.

```
10 FOR A = 0 TO 95
20 READ B: POKE USR "A" + A, B
30 NEXT A
40 FOR B = 1 TO 9
50 LET V = 0: LET R = 0: LET X = 4: LET Y = 1
60 BORDER 0: PAPER 0: INK 7: BRIGHT 1
70 PRINT AT 0, 0; "SCORE "; B - 1; TAB 10; PAPER 1; INK 6;
   " STUNT BIKE "
80 FOR A = 0 TO 31
90 PRINT AT 5, A; INK 4; "g3"; AT 13, A; "g3"; AT 21, A; "g3"
100 NEXT A
105 FOR A = 0 TO B + 4: PRINT AT 9, A * 2 + 2; INK 2; "gK gL": NEXT A
110 PRINT AT 20, 6; INK 4; "gG g8c ";
120 FOR A = 1 TO B + 5
130 PRINT INK 5; "gF";
140 NEXT A
150 PRINT INK 4; " g8c g8c gH"
155 IF R > .7 THEN PRINT AT X - 2, Y; INK 6; " gC gE";
   AT X - 1, Y - 1; " gJ gI gA"; AT X, Y - 1; " gB "":
   GO TO 170 + 130 * (X = 12 AND Y < B * 2 + 10.5)
```

```

160 PRINT AT X - 1, Y - 1; INK 6; "□□gCgE"; AT X, Y - 1;
    "□gB gD gA"; AT X - 2, Y; "□□□"
170 LET R = R + .2 * (INKEY$ = "0") - .1 * (INKEY$ = "1" AND
    V > 0) - .12 * (V > 0); LET V = V + R; IF V > 10 THEN LET V = 10
175 IF V < 0 THEN LET V = 0
180 LET Y = Y + V / 10 ; IF INT Y = 28 THEN PRINT AT X - 2, Y;
    "□□□"; AT X - 1, Y - 1; "□□□□"; AT X, Y - 1; "□□□□":
    LET X = X + 8; LET Y = 1
190 BEEP .01, V * 3 - 20; IF INT (Y + .5) = 4 AND X = 20 THEN GO TO 220
200 IF X < 27 THEN GO TO 155
210 NEXT B: CLS: PRINT "CONGRATULATIONS! YOU HAVE JUST
    □□Cleared all 15 buses on your □□□□stunt bike,
    AND QUALIFIED FOR A PLACE IN THE GUINNESS BOOK OF
    □□□RECORDS!": STOP
220 PRINT AT 18, 3; "□□□" ' "□□□□□□□" ' "□□□□□□":
    IF R < .2 THEN LET R = .2
225 LET X = X - V / 7; IF X < 16 THEN LET X = 16
230 LET V = V - R / 2 - .42; LET Y = Y + 1; LET X = X + .3
235 IF X > = 20 THEN GO TO 270
240 PRINT AT X - 1, Y - 1; INK 6; "□□gCgE"; AT X, Y - 1;
    "□gB gD gA"; AT X - 2, Y; "□□□"
245 BEEP .03, 10 - 3 * (X - 15)
250 PRINT AT X - 1, Y - 1; "□□□□"; AT X, Y - 1; "□□□□";
    AT X - 2, Y; "□□□"
260 GO TO 225
270 IF Y > B + 17 THEN GO TO 210
272 PRINT #1; PAPER 2; FLASH 1; "You have crashed into the Buses Hit
    ENTER to play again □□□□□□□□"
280 IF INKEY$ = CHR$ 13 THEN RUN
290 GO TO 305

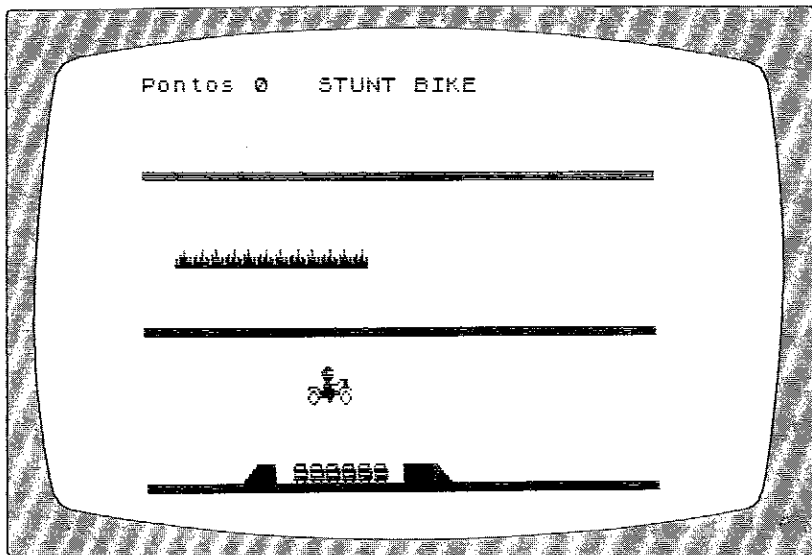
```



```

300 PRINT #1; PAPER 2; FLASH 1; "You have crashed in the Fire 
    Tunnel.  Hit ENTER to play again"
305 IF INKEY$ = CHR$ 13 THEN RUN
310 GO TO 305
1000 DATA 60, 66, 152, 164, 66, 66, 36, 24
1010 DATA 60, 66, 25, 37, 66, 66, 36, 24
1020 DATA 56, 124, 224, 252, 124, 120, 48, 127
1030 DATA 120, 124, 255, 255, 252, 24, 24, 0
1040 DATA 0, 0, 0, 0, 120, 152, 24
1050 DATA 62, 66, 127, 127, 65, 127, 127, 34
1060 DATA 1, 3, 7, 15, 31, 63, 127, 255
1070 DATA 128, 192, 224, 240, 248, 252, 254, 255
1080 DATA 63, 127, 254, 252, 248, 252, 236, 192
1090 DATA 0, 0, 0, 1, 3, 7, 15, 31
1100 DATA 8, 8, 40, 61, 189, 191, 255, 255
1110 DATA 4, 4, 36, 109, 109, 127, 255, 255

```



```

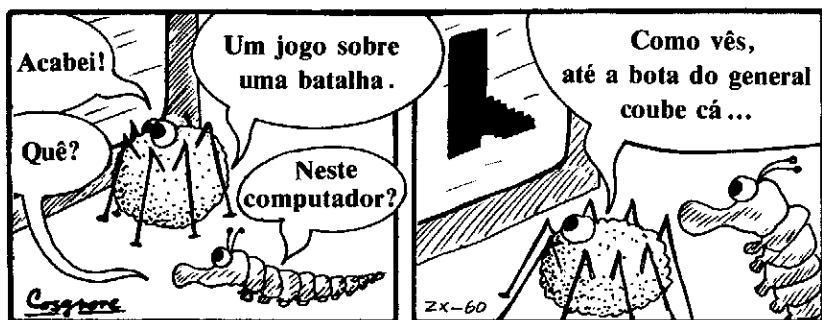
10 FOR a=0 TO 95
20 READ b: POKE USR "a"+a,b
30 NEXT a
40 FOR b=1 TO 9
50 LET v=0: LET r=0: LET x=4:
LET y=1
60 BORDER 0: PAPER 0: INK 7: B
RIGHT 1
70 PRINT AT 0,0:"Pontos ";b-1;
TAB 10: PAPER 1: INK 6:" STUNT B
IKE "
80 FOR a=0 TO 31
90 PRINT AT 5,a: INK 4;"=";AT
13,a;"=";AT 21,a;"="
100 NEXT a
105 FOR a=0 TO b+4: PRINT AT 9,
a*2+2: INK 2;"▀"; NEXT a
110 PRINT AT 20,b: INK 4;"▀ ";
120 FOR a=1 TO b+5
130 PRINT INK 5;"▀";
140 NEXT a
150 PRINT INK 4;"▀"
155 IF r>.7 THEN PRINT AT x-2,y
: INK 6;" ";AT x-1,y-1;" ";
:AT x,y-1;" "; GO TO 170+10
0*(x=12 AND y<b*2+10.5)
160 PRINT AT x-1,y-1: INK 6;"
";AT x,y-1;" ";AT x-2,y;"
"
170 LET r=r+.2*(INKEY$="0")-.1*
(INKEY$="1" AND v>0)-.12*(v>0):
LET v=v+r: IF v>10 THEN LET v=10
175 IF v<0 THEN LET v=0
180 LET y=y+v/10: IF INT y=28 T
HEN PRINT AT x-2,y;" ";AT x-1,
y;" ";AT x,y-1;" "; LET x=
x+6: LET y=1
190 BEEP .01,v*3-20: IF INT (y+
.5)=4 AND x=20 THEN GO TO 220
200 IF x<27 THEN GO TO 155
210 NEXT b: CLS : PRINT "
PARABENS!
Acabou
de saltar os 15 carros nasua ""
STUNT BIKE"" e qualificou-se par
a um lugar no Guinness Book!": ST
OP
220 PRINT AT 18,3:" ";
""": IF r<.2 THEN LET r=
.2
225 LET x=x-v/7: IF x<16 THEN L
ET x=16
230 LET v=v-r/2-.42: LET y=y+1:
LET x=x+.3
235 IF x>=20 THEN GO TO 270
240 PRINT AT x-1,y-1: INK 6;"
";AT x,y-1;" ";AT x-2,y;"
"

```

```

045 BEEP .03,10-3*(X-15)
050 PRINT AT X-1,Y-1;" ";AT
X,Y-1;" ";AT X-2,Y)" ";
060 GO TO 225
070 IF Y>b+17 THEN GO TO 210
072 PRINT #1; PAPER 2; FLASH 1;
"Estampou-se contra os carros.
Carregue em ENTER para jogar
outra vez"
080 IF INKEY#=CHR# 13 THEN RUN
090 GO TO 305
300 PRINT #1; PAPER 2; FLASH 1;
"Estampou-se no tunel de fogo.
Carregue em ENTER para jogar
outra vez"
305 IF INKEY#=CHR# 13 THEN RUN
310 GO TO 305
1000 DATA 60,66,152,164,66,66,36
,24
1010 DATA 60,66,26,37,66,66,36,2
4
1020 DATA 56,124,224,252,124,120
,48,127
1030 DATA 120,124,255,255,252,24
,24,0
1040 DATA 0,0,0,0,0,120,152,24
1050 DATA 62,66,127,127,66,127,1
27,34
1060 DATA 1,3,7,15,31,63,127,255
1070 DATA 128,192,224,240,248,25
0,254,255
1080 DATA 63,127,254,252,248,252
,236,192
1090 DATA 0,0,0,1,3,7,15,31
1100 DATA 8,8,40,61,109,191,255,
255
1110 DATA 4,4,36,109,109,127,255
,255

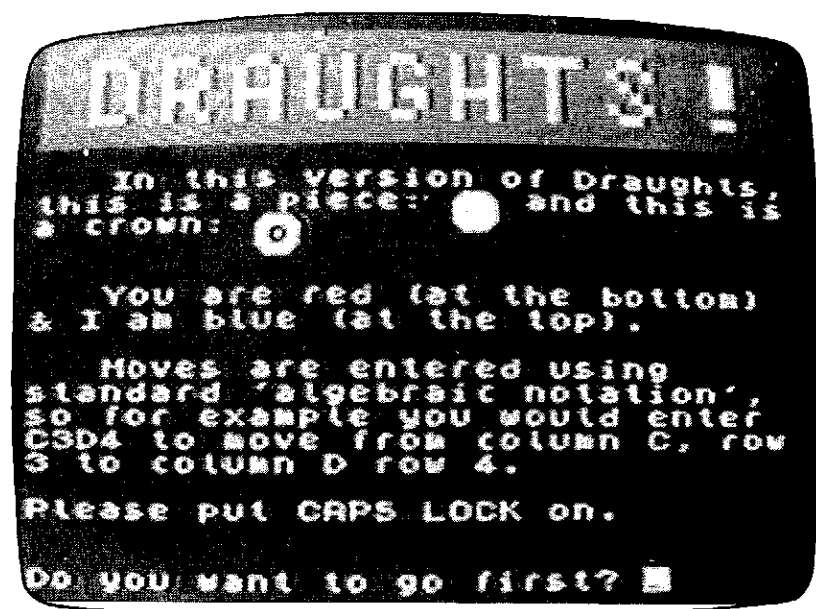
```



*À medida que o tempo frio aparece,
sente-se ao pé da lareira e jogue...*



DAMAS draughts



DRAUGHTS (damas) é um jogo para duas pessoas. Era jogado na Europa do Sul nos tempos medievais e parece que era uma consequência de jogos mais antigos do Médio Oriente. O objectivo do jogo é «comer» todas as peças do Spectrum ou fazer com que o Spectrum caia nas armadilhas. As instruções restantes encontram-se no próprio jogo.

Este programa ocupa quase todo o espaço num Spectrum 16 K e daí que não haja da parte do Spectrum uma verdadeira avaliação global do tabuleiro que um jogo inteligente como as damas merecia. Como resultado disso o padrão de jogo do Spectrum é bastante baixo — apesar de ser um adversário divertido. Se possui um 48 K, tente intro-

duzir alguns movimentos lógicos reais. Como o autor do *Jogo de Xadrez* publicado por Prism Computing, também recomendo que comece por incorporar uma rotina que avalie a situação do tabuleiro, isto é, que avalie qual a força da sua posição nesse momento e como é que determinados movimentos poderão pô-lo em perigo. Para conseguir isto, há necessidade de construir uma rotina para cada um destes casos.

Voltando ao jogo — a entrada é directa, sendo as peças desenhadas com gráficos. Note que uma dama é gerada automaticamente quando conseguir chegar ao lado contrário. Apesar de as regras do jogo original obrigarem sempre a comer, o Spectrum não avalia essa obrigatoriedade e portanto não o força a tomar essa atitude mas ele come sempre que pode.

O programa está subdividido em vários blocos tendo cada um, um título REM descrevendo para que serve. A única outra coisa que você realmente precisa saber é como o tabuleiro está numerado. Fica assim:

	72		71		70		69
66		65		64		63	
	59		58		57		56
53		52		51		50	
	46		45		44		43
40		39		38		37	
	33		32		31		30
27		26		25		24	

Note que os quadrados numerados são pretos. Usando este método invulgar, os quadrados opostos na diagonal têm valores correspondentes a ± 6 ou ± 7 consoante a direcção. É para isso que o «array» X é usado. O «array» A é usado para fazer o tabuleiro de 10 por 10 — 36 quadrados ficam de fora. Outras variáveis usados são:

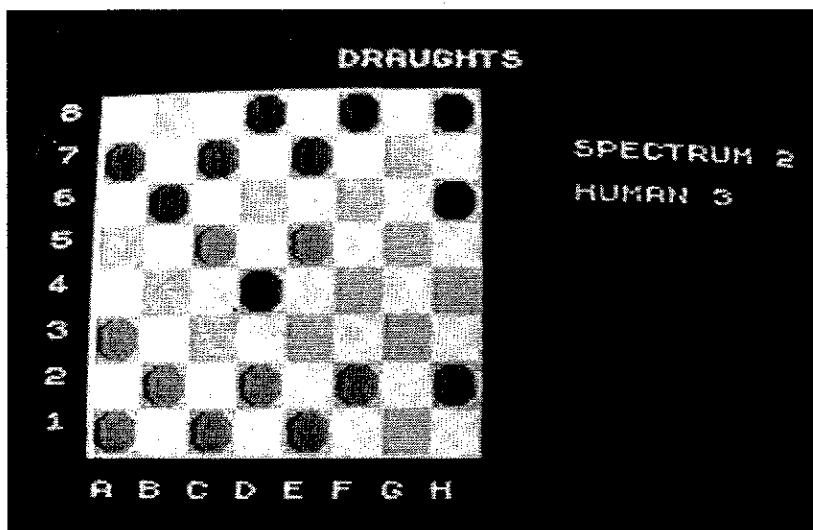
- H — peça do jogador
- C — peça do computador
- W — dama do jogador
- K — dama do computador

B— fora do tabuleiro
 LS— fim de jogo
 CS— pontuação do computador
 HS— pontuação do jogador
 MO— fim de movimento

Será fácil concluir, pois, o que cada parte do programa faz exactamente.

Pergunta: onde está a rotina do conjunto e a rotina que adivinha um movimento se não encontra uma boa jogada?

Se adicionar linhas extras para tornar o jogo mais competitivo eu gostaria de saber como fez!



10 BORDER 0: POKE 23624, 70: OVER 0: BRIGHT 1: INVERSE 0:

PAPER 0: INK 6: CLS

20 LET BD = 0

30 GO SUB 6000

40 GO SUB 4000

50 GO SUB 9000

60 LET MO = 0

70 GO SUB 5000

```

1000 REM ACCEPT PLAYERS MOVE
1010 LET MO = 0
1020 INPUT "ENTER MOVE: □"; LINE B$
1030 PRINT AT 21, 0; "Your move—"; B$ (TO 2); "□ to □"; B$ (3 TO);
      "□ □ □"
1040 LET Z (1) = CODE B$ (1) * 10 + CODE B$ (2)
1050 LET Z (2) = CODE B$ (3) * 10 + CODE B$ (4)
1060 FOR F = 1 TO 2
1070 RESTORE 9160: BEEP .02, 35
1080 READ X, Y
1090 IF X = 0 THEN PRINT AT 21, 0; INK 2; PAPER 7; "ILLEGAL MOVE!";
      PAPER 0; "[15 spaces]": BEEP .5, -10: GO TO 1020
1100 IF Z (F) = X THEN LET B (F) = Y: GO TO 1120
1110 GO TO 1080
1120 NEXT F
1160 LET MO = 0
1170 LET A (B (2)) = A (B (1))
1180 LET A (B (1)) = B
1190 IF ABS (B (1) - B (2)) > 7 THEN LET MO = 1:
      LET A ((B (1) + B (2)) / 2) = B
1200 GO SUB 5000
2000 REM THE 'INTELLIGENT' BIT
2020 FOR F = 24 TO 72: BEEP .015, F/2: BEEP .008, F/2 + 1
2050 IF A (F) <> C AND A (F) <> K THEN GO TO 2100
2060 FOR D = 1 TO 4
2070 IF A (F) <> K AND D > 2 THEN GO TO 2100
2080 IF (A (F + X (D)) = H OR A (F + X (D)) = W)
      AND A (F + 2 * X (D)) = B THEN LET Q = X (D): GO TO 2120
2090 NEXT D
2100 NEXT F

```

```

2110 GO TO 2240
2120 LET A (F + 2 * Q) = A (F)
2130 LET CS = CS + 1
2140 LET A (F + Q) = B
2150 LET P = F + 2 * Q: LET A (F) = B
2160 LET M = 0: FOR D = 1 TO 4
2170 IF (A (P + X (D)) = H OR A (P + X (D)) = W)
    AND A (P + 2 * X (D)) = B THEN LET M = X (D)
2180 IF A (P) < > K AND D > 2 THEN GO TO 2200
2190 IF M = 0 THEN NEXT D
2200 IF M = 0 THEN GO TO 60
2210 GO SUB 5000: LET A (P + M) = B: LET A (P + 2 * M) = A (P):
    LET A (P) = B: LET S = S + 1
2230 GO TO 60
2240 LET Y = 0
2250 BEEP .07, -2: LET F = 24 + INT (RND * 49): LET Y = Y + 1:
    IF Y > 400 THEN LET LS = 1: GO TO 5000
2260 IF A (F) < > C AND A (F) < > K THEN GO TO 2250
2270 FOR D = 1 TO 4
2280 IF A (F) = C AND D > 2 THEN GO TO 2250
2290 IF A (F + X (D)) = B THEN LET A (F + X (D)) = A (F):
    LET A (F) = B: GO TO 60
2300 NEXT D
2310 GO TO 2250
3000 REM CONVERT TO CROWNS
3010 FOR F = 24 TO 72
3020 IF A (F) = H AND F > 68 THEN LET A (F) = W
3030 IF A (F) = C AND F < 28 THEN LET A (F) = K
3040 NEXT F
3050 RETURN

```



```

4010 PRINT PAPER 1; INK 4; “[34 spaces] g3c g7c □ g3c g7c □ g4 g7c
□ g7c g4 □ g4 g7c □ g7c g4 □ g3c g3c g7c □ g3c □ □ □ g3c □ □ □ □ □
g5c g5 □ g5c g5 □ g5c g5 □ g5c g5 □ g5c g1 □ g5c g5 □ □ g5c □
g1 g7c g2 □ □ g8c □ □ □ □ □ g5c g5 □ g4c g5c □ g4c g7 □ g5c g5
□ g5c g7 □ g4c g7 □ □ g5c □ g4 g1 g7c □ □ g3 □ □ □ □ □ g3 g2
□ g2 g1 □ g2 g1 □ g1 g2 □ g1 g2 □ g2 g1 □ □ g2 □ □ g3 □ □ □ g3 [36 spaces]”
4020 PRINT ‘ “□ □ □ In this version of Draughts, this is a piece: gA gB and this
is a crown: gE gF □ □ □ □ □ □ gC gD [22 spaces] gG gH □ ’ ’ ’
“□ □ □ You are red (at the bottom) □ □ & I am blue (at the top).”
’ ’ “□ □ □ Moves are entered using □ □ □ □ □ □ standard ‘algebraic
notation’, □ □ so for example you would enter □ □ C3D4 to move from
column C, row 3 to column D row 4.”
4030 PRINT ‘ “Please put CAPS LOCK on.”
4040 RETURN
5000 REM DISPLAY BOARD
5010 GO SUB 3000: BEEP .09, 20: BEEP .05, 15: IF MO = 1 THEN
LET HS = HS + 1
5020 PRINT AT 0, 12; INK 6; PAPER 1; “DRAUGHTS”; PAPER 0;
INK 6; AT 4, 22; “SPECTRUM □”; INK 7; CS; AT 6, 22; INK 6;
“HUMAN □”; INK 7; HS: IF BD = 1 THEN GO TO 5080
5030 FOR F = 3 TO 15 STEP 4
5040 PRINT INK 4; PAPER 7; AT F - 1, 2; “□ □ g8c g8c □ □ g8c g8c □ □
g8c g8c □ □ g8c g8c”; AT F, 2; “□ □ g8c g8c □ □ g8c g8c □ □ g8c g8c
□ □ g8c g8c”; AT F + 1, 2; “g8c g8c □ □ g8c g8c □ □ g8c g8c □ □ g8c g8c
□ □”; AT F + 2, 2; “g8c g8c □ □ g8c g8c □ □ g8c g8c □ □ g8c g8c □ □”
5050 PRINT AT F - 1, 0; CHR$( 57 - F/2); AT F + 1, 0; CHR$( 56 - F/2)
5060 NEXT F: LET BD = 1
5070 PRINT AT 19, 2: “A □ B □ C □ D □ E □ F □ G □ H”: INK 6:
PLOT 15, 160: DRAW 129, 0: DRAW 0, -129: DRAW -129, 0:
DRAW 0, 129: INK 6

```

```

5080 LET F = 0: FOR E = 0 TO 7: LET S = (E/2 = INT (E/2)):
      LET F = F + 7 * S + 6 * NOT S: LET D = 2 * S
5090 FOR J = 0 TO 3: LET V = A (79 - F - J)
5100 PRINT INK 1 + (V < C); PAPER 4; ATE * 2 + 2, 4 * J + 2 + D;
      ("□□" AND V = B); ("gA gB" AND (H = V OR C = V));
      ("gE gF" AND (K = V OR W = V)); ATE * 2 + 3, 4 * J + 2 + D;
      ("□□" AND B = V); ("gC gD" AND (H = V OR C = V));
      ("gG gH" AND (K = V OR W = V))
5150 NEXT J: NEXT E
5160 PRINT AT 21, 0; "[32 spaces]"
5170 GO SUB 7000: RETURN
6000 REM g8c INITIALISE U.D.G.s g8c
6010 RESTORE 9190
6020 FOR F = 1 TO 8: READ A$
6030 FOR G = 0 TO 7
6040 READ H: POKE USR A$ + G, H
6050 NEXT G
6060 NEXT F
6070 RETURN
7000 REM g8c VARIOUS CHECKS g8c
7010 IF LS = 1 THEN PRINT AT 8, 22; "I give up!": INPUT "Hit ENTER
      to play again □"; LINE A$: RUN
7020 IF CS = 12 THEN PRINT AT 6, 24; FLASH 1; "I WIN": PAUSE 0: RUN
7030 IF HS = 12 THEN PRINT AT 6, 24; FLASH 1; "YOU WIN":
      PAUSE 0: RUN
7040 LET Q$ = "□": IF MO = 1 THEN PRINT AT 6, 22; "again?":
      INPUT "Can you jump? □"; LINE Q$: PRINT AT 8, 22; "□□□□□□□":
      IF Q$ (1) = "Y" THEN GO TO 1000
7050 LET MO = 0
7060 RETURN

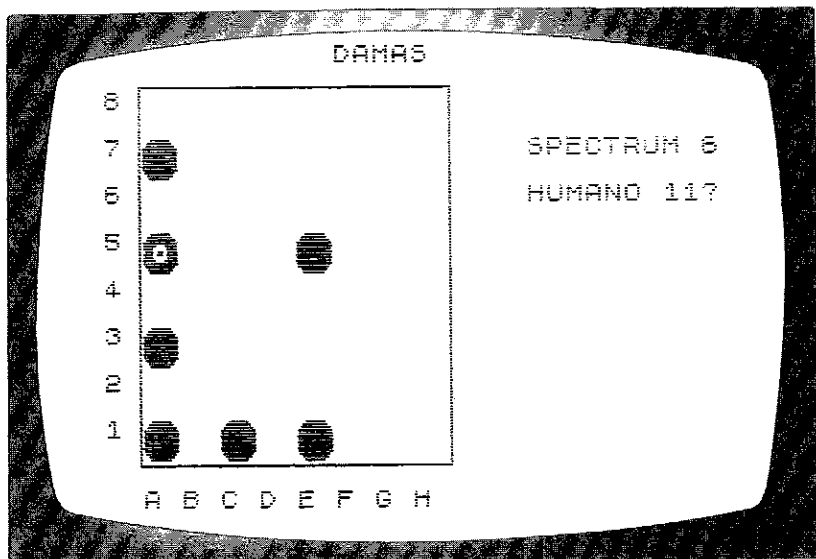
```

```

9000 REM
9010 RESTORE 9150: DIM A (100): LET Y = 0
9020 DIM X (5)
9030 FOR F = 1 TO 5: READ G: LET X (F) = G: NEXT F
9040 LET H = 0: LET C = 128: LET W = 1: LET K = 129: LET B = 255:
    LET Q = 0
9050 FOR F = 1 TO 100
9060 LET A (F) = 9: IF F < 73 AND F > 55 AND F <> 68 AND F <> 67
    AND F <> 62 AND F <> 61 AND F <> 60 THEN LET A (F) = C:
    GO TO 9090
9070 IF F < 54 AND F > 42 AND F <> 49 AND F <> 48 AND F <> 47
    THEN LET A (F) = B: GO TO 9090
9080 IF F < 41 AND F > 23 AND F <> 36 AND F <> 35 AND F <> 34
    AND F <> 29 AND F <> 28 THEN LET A (F) = H
9090 NEXT F
9100 LET LS = 0: LET CS = 0: LET HS = 0
9110 DIM B$(4): DIM B (2): DIM Z (2)
9120 LET MO = 0: LET B$(3) = "K"
9130 INPUT "Do you want to go first? "; LINE A$: CLS:
    IF A$(1) = "Y" THEN RETURN
9140 GO SUB 5000: LET Z = 57 + INT (RND * 3): LET Q =
    INT (RND * 2) - 7: LET A (Z + Q) = C: LET A (Z) = B: RETURN
9150 DATA -6, -7, 6, 7, 9
9160 DATA 699, 27, 701, 40, 703, 53, 705, 66, 710, 33, 712, 46, 714, 59,
    716, 72, 719, 26, 721, 39, 723, 52, 725, 65
9170 DATA 730, 32, 732, 45, 734, 58, 736, 71, 739, 25, 741, 38, 743, 51,
    745, 64, 750, 31, 752, 44, 754, 57, 756, 70
9180 DATA 759, 24, 761, 37, 763, 50, 765, 63, 770, 30, 772, 43, 774, 56,
    776, 69, 0, 0

```

9190 DATA "A", 0, 15, 31, 63, 127, 127, 127, 127
 9200 DATA "B", 0, 240, 248, 252, 254, 254, 254, 254
 9210 DATA "C", 127, 127, 127, 127, 63, 31, 15, 0
 9220 DATA "D", 254, 254, 254, 254, 252, 248, 240, 0
 9230 DATA "E", 0, 15, 31, 63, 127, 124, 120, 121
 9240 DATA "F", 0, 240, 248, 252, 254, 62, 30, 158
 9250 DATA "G", 121, 120, 124, 127, 63, 31, 15, 0
 9260 DATA "H", 158, 30, 62, 254, 252, 248, 240, 0



```

10 BORDER 0: POKE 23624,70: OV
ER 0: BRIGHT 1: INVERSE 0: PAPER
0: INK 6: CLS
20 LET bd=0
30 GO SUB 6000
40 GO SUB 4000
50 GO SUB 9000
60 LET mo=0
70 GO SUB 5000
1000 REM Entrada de jogadas
1010 LET mo=0
1020 INPUT "jogada: "; LINE b#
  
```



```

1030 PRINT AT 21,0;"sua jogada-"
; b$( TO 2);" para "; b$(3 TO );"
1040 LET z(1)=CODE b$(1)*10+CODE
b$(2)
1050 LET z(2)=CODE b$(3)*10+CODE
b$(4)
1060 FOR f=1 TO 2
1070 RESTORE 9160: BEEP .02,35
1080 READ x,y
1090 IF x=0 THEN PRINT AT 21,0;
INK 2; PAPER 7;"MOVIMENTO PROIBI
DO!"; PAPER 0;":
BEEP .5,-10: GO TO 1020
1100 IF z(f)=x THEN LET b(f)=y:
GO TO 1120
1110 GO TO 1080
1120 NEXT f
1160 LET mo=0
1170 LET a(b(2))=a(b(1))
1180 LET a(b(1))=b
1190 IF ABS(b(1)-b(2))>7 THEN L
ET mo=1: LET a((b(1)+b(2))/2)=b
1200 GO SUB 5000
2000 REM INTELEGENTE
2020 FOR f=24 TO 72: BEEP .015,f
/2: BEEP .008,f/2+1
2030 IF a(f)<>c AND a(f)<>k THEN
GO TO 2100
2060 FOR d=1 TO 4
2070 IF a(f)<>k AND d>2 THEN GO
TO 2100
2080 IF (a(f+x(d))=h OR a(f+x(d)
)=w) AND a(f+2*x(d))=b THEN LET
q=x(d): GO TO 2120
2090 NEXT d
2100 NEXT f
2110 GO TO 2240
2120 LET a(f+2*q)=a(f)
2130 LET cs=cs+1
2140 LET a(f+q)=b
2150 LET p=f+2*q: LET a(f)=b
2160 LET m=0: FOR d=1 TO 4
2170 IF (a(p+x(d))=h OR a(p+x(d)
)=w) AND a(p+2*x(d))=b THEN LET
m=x(d)
2180 IF a(p)<>k AND d>2 THEN GO
TO 2200
2190 IF m=0 THEN NEXT d
2200 IF m=0 THEN GO TO 60
2210 GO SUB 5000: LET a(p+m)=b:
LET a(p+2*m)=a(p): LET a(p)=b: L
ET s=s+1
2230 GO TO 60
2240 LET y=0
2250 BEEP .07,-2: LET f=24+INT(
RND*40): LET y=y+1: IF y>400 THE
N LET ls=1: GO TO 5000

```



```

5050 PRINT AT f-1,0;CHR# (57-f/2
);AT f+1,0;CHR# (56-f/2)
5060 NEXT f: LET bd=1
5070 PRINT AT 19,2;"A B C D E F
G H": INK 5: PLOT 15,150: DRAW 1
29,0: DRAW 0,-129: DRAW -129,0:
DRAW 0,129: INK 6
5080 LET f=0: FOR e=0 TO 7: LET
s=(e/2=INT (e/2)): LET f=f+7*s+5
*NOT s: LET d=2*s
5090 FOR j=0 TO 3: LET v=s(79-f-
j)
5100 PRINT INK 1+(v<c): PAPER 4;
AT e*2+2,4*j+2+d;(" " AND v=b/);
(" " AND (h=v OR c=v));(" " AN
D (k=v OR w=v));AT e*2+3,4*j+2+d
;(" " AND b=v);(" " AND (h=v O
R c=v));(" " AND (k=v OR w=v))
5150 NEXT j: NEXT e
5160 PRINT AT 21,0;"
..

6170 GO SUB 7000: RETURN
6000 REM DEFINIR USU
6010 RESTORE 9190
6020 FOR f=1 TO 8: READ a#
6030 FOR g=0 TO 7
6040 READ h: POKE USA a#+g,h
6050 NEXT g
6060 NEXT f
6070 RETURN
7000 REM PARA TER XERQUE
7010 IF ls=1 THEN PRINT AT 8,22;
"Desisto!": INPUT "Carregue em E
NTER para jogar outra vez "; LIN
E a#: RUN
7020 IF cs=12 THEN PRINT AT 6,24
; FLASH 1;"GANHEI!": PAUSE 0: RU
N
7030 IF hs=12 THEN PRINT AT 6,24
; FLASH 1;"GANHO!": PAUSE 0: RU
N
7040 LET q#=" ": IF mo=1 THEN PR
INT AT 6,22;"OUTRA VEZ?": PRINT
AT 21,0;"Pode saltar?(s/n) ": IN
PUT LINE q#: PRINT AT 8,22;"
": IF q#(1)="S" THEN GO TO 100
0
7050 LET mo=0
7060 RETURN
9000 REM
9010 RESTORE 9050: DIM a(100): L
ET y=0
9020 DIM x(5)
9030 FOR f=1 TO 5: READ g: LET x
(f)=g: NEXT f
9040 LET h=0: LET c=128: LET w=1
: LET k=129: LET b=255: LET q=0

```

```

9050 FOR f=1 TO 100
9060 LET a(f)=9: IF f<73 AND f>5
5 AND f<>55 AND f<>67 AND f<>62
AND f<>61 AND f<>60 THEN LET a(f
)=c: GO TO 9090
9070 IF f<54 AND f>42 AND f<>49
AND f<>48 AND f<>47 THEN LET a(f
)=b: GO TO 9090
9080 IF f<41 AND f>23 AND f<>36
AND f<>35 AND f<>34 AND f<>29 AN
D f<>28 THEN LET a(f)=h
9090 NEXT f
9100 LET ls=0: LET cs=0: LET hs=
0
9110 DIM b$(4): DIM b(2): DIM z(
2)
9120 LET mo=0: LET b$(3)="k"
9130 INPUT "Over ser o primeiro
a jogar?(s/n) ": LINE a#: CLS :
IF a$(1)="S" THEN RETURN
9140 GO SUB 5000: LET z=57+INT (
RND*3): LET q=INT (RND*2)-7: LET
a(z+q)=c: LET a(z)=b: RETURN
9150 DATA -8,-7,6,7,9
9160 DATA 699,27,701,40,703,53,7
05,58,710,33,712,46,714,59,716,7
2,719,26,721,39,723,52,725,65
9170 DATA 736,32,738,45,734,58,7
36,71,739,25,741,38,743,51,745,6
4,750,31,752,44,754,57,756,70
9180 DATA 759,24,761,37,763,50,7
65,63,770,30,772,43,774,56,776,6
9,0,0
9190 DATA "a",0,15,31,63,127,127
,127,127
9200 DATA "b",0,240,248,252,254,
254,254,254
9210 DATA "c",127,127,127,127,63
,31,15,0
9220 DATA "d",254,254,254,254,25
2,248,240,0
9230 DATA "e",0,15,31,63,127,124
,120,121
9240 DATA "f",0,240,248,252,254,
60,30,153
9250 DATA "g",121,120,124,127,63
,31,15,0
9260 DATA "h",158,30,62,254,252,
248,240,0

```

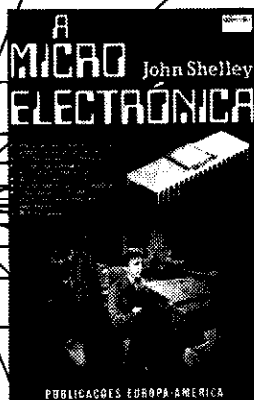

TUDO O QUE QUERIA SABER SOBRE COMPUTADORES... E TEVE MEDO DE PERGUNTAR

Para si que nada sabe ou receia perguntar, ou para si que já os conhece, dois livros que você deve ler.



Conheça os Computadores

Uma excelente introdução ao conhecimento dos computadores. Dos equipamentos aos programas, passando pelo processamento de dados, a codificação e os microcomputadores.



A Microelectrónica

O computador tradicional. O microprocessador e o microcomputador. As pastilhas («chips»). Como são feitas as pastilhas. Usando as pastilhas. Os microcomputadores nas escolas.

PUBLICAÇÕES EUROPA-AMÉRICA

Uma editora diferente que pensa... em si

Se deseja adquirir qualquer uma destas obras, poderá fazê-lo diretamente através dum simples postal para:

Edições CETOP – apartado 33 – 2726 MEM MARTINS CODEX.

Consegue saltar sobre nove autocarros com a sua moto? E lançar para o fundo do oceano um navio inimigo? Ou talvez a jardinagem e a corrida de minhocas esteja mais de acordo com o seu estilo. Estes jogos foram desenhados para testar ao máximo as suas faculdades e as potencialidades do Spectrum.

Dois destes jogos têm igualmente uma função utilitária, podendo usá-los em aplicações práticas da sua vida corrente.

Todos os programas correm no Spectrum 16K.