

Tim Hartnell, já nosso conhecido de obras traduzidas nesta colecção, *Como Programar o seu ZX Spectrum* e a recente *ZX Spectrum na Educação*, continua a surpreender-nos com a qualidade dos seus trabalhos! Desta vez ele reúne 49 programas de jogos, da autoria dos melhores programadores do Reino Unido. Os programas, muitos dos quais utilizam subrotinas de código-máquina, incluem jogos verdadeiramente emocionantes entre os quais destacamos: «Protector», «Bombardeiro», «Pássaros», «Helicóptero», «Corridas», «Zombies», «Space Treck», «Lógicos», «Fantasmas», «Biorritmo» e muitos, muitos outros! Leitor, não perca mais tempo! Vá para casa fazer «explodir» o seu Spectrum!



EDITORIAL PRESENÇA

49 Jogos Explosivos para o ZX Spectrum

182

TIM HARTNELL

DAVID PERRY, GRAHAM CHARLTON, NEIL PELLINACCI e MALCOLM YOUNG

49

JOGOS EXPLOSIVOS

PARA O ZX SPECTRUM



TEMPOS
LIVRES

CULTURA E TEMPOS LIVRES

1. ABC do Xadrez, *Pejar Trifunovitch e Sava Vukovitch*
4. ABC do Bridge, *Pierre Jais e H. Lahana*
5. Guia Prático de Fotografia, *W. D. Emanuel*
6. ABC do Judo, *E. J. Harrison*
7. Como Fazer Cinema, *Paul Petzold*
8. Bridge Moderno, *Pierre Jais e H. Lahana*
9. Fotografia — Técnicas e Truques I, *Edwin Smith*
10. ABC dos Estilos, da Arquitectura ao Mobiliário, *A. Ausel*
11. Fotografia — Técnicas e Truques II, *Edwin Smith*
12. A Pesca Submarina, *António Ribera*
13. Teoria dos Finais de Partida, *Yuri Averbach*
14. Aprenda Rádio, *B. Fighiera*
15. Guia do Cão, *Louise Laliberté-Robert e Jean-Pierre, Robert*
16. ABC do Aquário, *Anthony Evans*
17. Iniciação à Electricidade e Electrónica, *Fernand Huré*
18. Os Transistores, *Fernand Huré*
19. Karaté I, *Albrecht Pflüger*
20. Iniciação ao Radiocomando dos Modelos Reduzidos, *C. Péricono*
21. Construa o seu Receptor, *B. Fighiera*
22. Montagens Electrónicas, *B. Fighiera*
23. O Berbequim Eléctrico, *Villy Dreier*
24. Cactos, *J. Nilaus Jensen*
25. Iniciação a Alta Fidelidade, *Peter Turner*
26. O Aquário de Água Doce, *Paulo de Oliveira*
27. ABC do Ténis, *Fonseca Vaz*
28. Karaté II, *Albrecht Pflüger*
29. ABC da Criação de Canários, *Curt Af Enehjelm*
30. Ginástica Feminina, *Sonja Helmer Jensen*
31. Cartomancia, *Rhea Koch*
32. Calculadoras Electrónicas de Bolso, *E. Dam Ravn*
33. O Pastor Alemão, *Gilles Legrand*
34. Xadrez — Teoria do Meio Jogo I, *Bondarevsky*
35. Manual do Super 8 — I, *Myron A. Matzkin*
36. ABC da Criação de Periquitos, *Cyril H. Rogers*
37. O Livro dos Gatos, *Barbel Gerber e Horst Bielfeld*
38. Manual do Super 8 — II, *Myron A. Matzkin*
39. ABC do Mergulho Desportivo, *Walter Mattes*
40. Circuitos Integrados/Aplicações Práticas, *F. Bergfoi*
41. A Apicultura, *H. R. C. Riches*
42. ABC do Cultivo das Plantas, *H. G. Witham Fogg*
43. ABC da Criação de Pombos, *Kai R. Dahl*
44. Construção de Caixas Acústicas de Alta Fidelidade e. R. Brauti
45. Raças de Canários, *Klaus Speicher*
46. Jogos de Cartas, *Graciano Dolma*
47. Cocker Spaniels, *H. S. Lloyd*
48. ABC da Caça, *Fabian Abril*
49. Aprenda Televisão, *Gordon J. King*
50. Iniciação à Pesca, *Juan Nadal*
51. Basquetebol, *Marius Norregard*
52. Cães de Casa, *Santiago Pons*
53. Aprenda Electrónica, *T. L. Squires e C. M. Deason*
54. A Avicultura, *Jim Worthington*
55. A Produção de Coelho, *P. Surdeau e R. Henaff*
56. ABC dos Computadores, *T. F. Fry*
57. Natação para Crianças, *John Idorn*
58. O Boxer, *Anni Mortensen*
59. Voleibol, *Ole Hansen e Per-Göran Persson*
60. Iniciação à Vela, *Donald Law*
61. ABC da Filatelia, *Jacqueline Caurat*
62. A Pesca à Beira-Mar, *J. M. Böelle e D. Doyen*
63. Enxerto de Árvores de Fruto, *Alejo Rigau*
64. A Cultura do Moranguero, *Luis Alstina Grau*
65. Emissores-Receptores (Walkies-Talkies), *P. Duranton*
66. Iniciação à Fotoelectrónica, *Heinz Richter*
67. Doces e Conservas de Fruta, *Robin Howe*
68. A Criação de Hamsters, *C. F. Snow*
69. A Criação de Porcos, *Roy Genders*
70. Calendário do Horticultor, *Luis Alstina Grau*
71. Jogos Electrónicos, *F. G. Rayer*
72. Cultivo de Cogumelos e Trufas, *Alejo Rigau*
73. Aprenda Televisão a Cores, *Gordon J. King*
74. Gravação em Fita Magnética, *Ian R. Sinclair*
75. Poda de Árvores e Arbustos, *Roy Genders*
76. Como Treinar o Seu Cão, *E. Fitch Daglish*
77. Instrumentos de Medida e Verificação, *Heinrich Stöckle*
78. A Criação de Caracóis, *Mattias Josa*
79. Rádio — Fundamentos e Técnicas, *Gordon J. King*
80. Como Fazer Gelados, *Sylvie Thiébaud*
81. Iniciação à Jardinagem, *Noel Clarasó*
82. A Congelação dos Alimentos, *Suzanne Lapointe*
83. Windsurf — Prancha à Vela, *Ernstfried Prade*
84. Raças de Cães, *O. Hasselfeldt*
85. Rummy e Canasta, *Claus D. Grupp*
86. A Encadernação, *Annie Persuy*
87. Aprenda Electricidade, *Heinz Richter*
88. Taxidermia, Embalsamento de Aves e Mamíferos, *Harry Hjortaa e*
89. Jogging — Correr para Manter a Forma, *Werner Sonntag*
90. ABC da Cozinha Chinesa, *Sonva Richmond*
91. Jogos T.V., *C. Tavernier*
92. Amplificadores de Som, *Richard Zierl*
93. O Livro do Poker, *Claus D. Grupp*
94. Aprenda a Desenhar, *Rose-Marie de Prémont e Nicole Philippi*
95. O Minitrampolim na Escola, *Sonja Helmer Jensen e Klaus Dano*
96. Jogos de Luzes e Efeitos Sonoros para Guitarras, *B. Fighiera*
97. O Cultivo do Tomate, *Louis N. Flawn*
98. Pilhas Solares, *F. Juster*
99. A Criação Doméstica de Coelho, *C. F. Snow*
100. Iniciação ao Futebol, *Wieland Mannle e Heinz Arnold*
101. Horóscopos Chineses, *Feorg Haddenbach*
102. Guia Prático de Marcenaria, *Charles H. Hayward*
103. Andebol I, *Fritz e Peter Hattig*
104. Dispositivos Anti-Roubo, *H. Schreiber*
105. Perus, Pintadas e Codornizes, *Jérôme Sauze*
106. Crepes — Doces e Salgados, *Florence Arzel*
107. Aperitivos e Entradas, *Myrette Tiano*
108. Ténis de Mesa, *Leslie Woollard*
109. Aprenda Surf, *R. Abbott e M. Baker*
110. Futebol — Técnica e Tática, *Kurt Lavall*
111. A Vaca Leiteira, *Collin T. Whittemore*
112. O Cubo Mágico, *Josef Trajber*
113. O Perdigueiro Português, *José M. Correia*
114. Pizzas e Massas à Italiana, *Marieanne Ránk*
115. O Cubo Para Quem Já o Faz, *Josef Trajber*
116. A Pirâmide Mágica. A Torre, o Barril do Diabo, *M. Mrowka-W. J. Weber*
117. Gansos e Patos, *Marie Mourthe*
118. Iniciação ao Kung-Fu, *A. P. Harrington*
119. Electrónica e Fotografia, *Hanns-Peter Siebert*
120. O Livro da Fortuna, *Douglas Hill*
121. Construção de um Alimentador de Corrente, *Waldemar Baitinger*
122. Hóquei em Patins, *Francisco Velasco*
123. Técnicas de Tiro, *Anton Kovacic*
124. Aprenda a Tricotar, *Uta Mix*
125. ABC da Patinagem, *Christa-Maria e Richard Kerler*
126. A Pesca e os seus Segredos, *Armand Deschamps*
127. O Osciloscópio, *R. Rateau*
128. Guia Prático da Banda do Cidadão, *T. M. Normand*
129. Sumos e Batidos, *Manfred Donderski*
130. Introdução à Programação de Microcomputadores, *Peter C. Sanderson*
131. Aprenda Crochê, *Uta Mix*
132. ABC do Microprocessador, *P. Mélusson*
133. Guia Prático de Basic, *Roger Hunt*
134. Introdução à Electrónica Digital, *Ian R. Sinclair*
135. ABC do Vídeo, *David K. Mathewson*
136. Fotografia em Movimento, *Don Morley*
137. Guia de Cobol, *Ray Welland*
138. Fotografia a Pequena Distância, *Sidney F. Ray*
139. Guia Moderno da Canaricultura, *Manuel Gonçalves*

140. Minieletrônica para Amadores, *Heinz Richter*
141. ABC da Programação de Computadores, *John Shelley*
142. Tarto — O Futuro Pelas Cartas, *Edwin J. Nigg*
143. ABC da Equitação, *Dorothy Johnson*
144. Como Programar o Seu ZX 81, *Patrick Gueulle*
145. 100 Avarias TV e a Maneira Prática de as Detectar, *P. Duranton*
146. ABC da Horticultura, *Louis Giordano*
147. Basic para Microcomputadores, *A. P. Stephenson*
148. Como Programar o seu ZX Spectrum, *Tim Hartnell e Dilwyn Jones*
149. Iniciação aos Motores Diesel, *Dávid S. Maclean*
150. 60 Jogos para o ZX Spectrum, *David Harwood*
151. As Linhas da Mão, *Rose Hubert*
152. Cozinha Italiana, *Rotraud Degner*
153. Manual do ZX Spectrum, *R. J. Simpson e T. J. Terrel*
154. Z80 Assembler para o ZX Spectrum — Iniciação ao Código de Máquina.
João Paulo Fragoso
155. Aeróbica, *H. Schulz*
156. ABC do Atletismo, *Denis Watts*
157. 26 Programas Basic para Microcomputadores, *Derrick Daines*
158. Aprenda Pascal no seu Microcomputador, *Jeremy Ruston*
159. Guia Moderno da Suinicultura, *Colin Whittemore*
160. O Bar em Sua Casa — 888 Cocktails, *Aladar von Wesendonk*
161. Código de Máquina para Principiantes, *James Walsh*
162. Código de Máquina para Programadores Avançados, *Paul Holmes*
163. ABC da Fruticultura, *Henri Gosselin*
164. ABC da Canoagem, *Alan Byde*
165. Guia de Fortran, *Philip Ridler*
166. Manual da Secretária, *Philippa Ramage*
167. ABC das Antenas, *Gordon J. King*
168. Programar Aventuras no Seu Computador, *Andrew Nelson*
169. Guia do Sinclair QL, *Boris Allan*
170. Novas Aventuras no seu ZX Spectrum, *Peter Shaw e James Mortleman*
171. O Computador no Escritório, *John Shelley*
172. Sobremesas, *Fred Timber*
173. Rádio — Do Circuito Oscilante ao Receptor de Ondas Curtas, *Richard Zierl*
174. ABC das Aberturas, *V. N. Panov*
175. A Construção de Pequenos Transformadores, *M. Dorian e F. Juster*
176. Gui de Pascal, *David Watt*
177. O ZX Spectrum na Educação, *Tim Hartnell, Christine Johnson e David Valentine*
178. Iniciação ao Snooker, *John Pulman*
179. Circuitos com Diacs, Tiristores e Triacs, *Fritz Bergtold*
180. O Jogo do Mahjongg, *Ursula Eschenbach*
181. Cultura Hidropónica. Culturas sem solo, *Laura Fronty*
182. 49 Jogos Explosivos para o ZX Spectrum, *Tim Hartnell*

TIM HARTNELL

David Perry, Graham Charlton
Nell Pellinacci e Malcolm Young

49 JOGOS EXPLOSIVOS PARA O ZX SPECTRUM

PREFÁCIO

Os livros tão compactos como este não saem do Spectrum sem um certo investimento de tempo e esforço.

E gastou-se de facto bastante tempo e esforço tentando apresentar uma colecção de jogos para o Spectrum tão boa quanto possível.

Partindo do trabalho de alguns dos melhores programadores jovens do Reino Unido, juntei uma série de programas para o Spectrum que merecem ser considerados «de qualidade». Os principais contribuidores foram Malcolm Young, David Perry, Graham Charlton e Neil Pellinacci, cujo trabalho preenche a maior parte do livro.

Outras pessoas a quem pedi que produzissem obras para esta colecção foram Scott Vincent, Raymond Blake, Neal Cavalier-Smith, Paul Toland, Andrew Sweetland, Martin Jones, Graham White, Tim Rogers, Damian Steele, Neville Predebon e Michael Briggs. Também eles merecem os nossos parabéns pela qualidade dos programas que nos apresentaram.

É o momento de fazer «explodir» o seu Spectrum...

Tim Hartnell

Londres, 1983

Titulo original:

49 Explosives Games for the ZX Spectrum

© Copyright by Hartnell, 1983

Tradução de Conceição Jardim e Miguel Nogueira

Capa de António Marques

Reservados todos os direitos para a língua portuguesa à

EDITORIAL PRESENÇA, LDA:

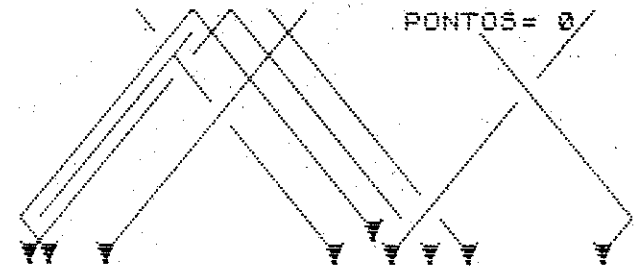
Rua Augusto Gil, 35-A — 1000 LISBOA

JOGOS DE PERÍCIA

PROTECTOR

O leitor já não precisa de gastar o seu dinheiro em jogos de perícia... «Protector» dá-lhe a possibilidade de salvar o mundo (ou pelo menos algumas cidades) sem sair de casa, como demonstram os exemplos de execução apresentados em seguida.

Os comandos do jogo são «M», «Q», «A» e «O». É o momento de realizar a sua missão e salvar o mundo...



```
10 POKE 23658,0: REM obter mai  
usculas  
20 CLS  
30 FOR F=1 TO 49  
40 BORDER F/7  
50 BEEP 0.05,F  
60 PRINT AT 5,10: INK 0: PAPER
```

```
4) FLASH 1;"PROTECTOR"
70 NEXT T
80 LET SC=0
90 LET HS=0
100 REMOTORA=0
110 LET N=0
120 PAPER 1: BORDER 1: INK 5: C
LS
130 LET Q#="███": REM Grafico
S ABC
140 LET A#=#
150 LET W#="+██████": REM Grafico
S DEFG
160 LET E#="███████": REM Graficos
SDEFG
170 FOR X=1 TO 12
180 READ A#
190 FOR Y=0 TO 7
200 READ A
210 POKE USA A#+Y,A
220 NEXT Y: NEXT X
230 DIM N$(15): DIM L(15,2):
DIM N(15): DIM X$(15)
240 DATA "P",00,00,00,00,00,00,00,
100,100,"B",10,10,10,10,10,10,10,10,
100,100,"C",10,10,10,10,10,10,10,10,
100,100,"D",10,10,10,10,10,10,10,10,
100,100,"E",10,10,10,10,10,10,10,10,
100,100,"F",10,10,10,10,10,10,10,10,
100,100,"G",10,10,10,10,10,10,10,10,
100,100,"H",10,10,10,10,10,10,10,10,
250 DATA "X",10,0,0,0,0,0,0,0,0,0,0,0,
0,100,"X",10,0,0,0,0,0,0,0,0,0,0,0,
0,100,"X",10,0,0,0,0,0,0,0,0,0,0,0,
0,100,"X",10,0,0,0,0,0,0,0,0,0,0,0,
100,100,"X",10,0,0,0,0,0,0,0,0,0,0,0,
260 FOR B=1 TO 5: LET N$(B,1)=0
#: LET N$(B,2)=W#: NEXT B
270 CLS
280 LET V=10: LET D=V: LET H=1:
LET U=5: GO SUB 970: GO SUB 980
290 FOR B=1 TO U: LET L(B,1)=INT
T (RND*H): LET L(B,2)=INT (RND*G
0): LET N(B)=INT (RND*D)+0-1: LET
T X$(B)=CHR$(196.5-N(B)/2): NEXT
T B: GO SUB 990
300 LET A1=17: LET A2=15
310 LET B=B+1: IF B>D THEN LET
B=1
320 PRINT AT 0,20; INK 5;"PONTO
0=";SC
```

```
330 PRINT AT L(B,1),L(B,2);X$(B
): LET L(B,1)=L(B,1)+1: LET L(B,
2)=L(B,2)+N(B): IF L(B,2)>31 THE
N LET L(B,2)=0
340 IF L(B,2)<0 THEN LET L(B,2)
=31
350 IF SCREEN$(L(B,1),L(B,2))<
">" THEN GO TO 710
360 PRINT AT L(B,1),L(B,2);"█":
REM Grafico J
370 LET L1=0: LET L2=0
380 IF INKEY#="" THEN GO TO 560
390 IF INKEY#="H" THEN GO TO 51
0
400 IF INKEY#="O" THEN LET L1=-
1
410 IF INKEY#="A" THEN LET L1=1
420 IF INKEY#="0" THEN LET L2=-
1
430 IF INKEY#="P" THEN LET L2=1
440 PRINT AT A1,A2:""
450 LET A1=A1+L1: LET A2=A2+L2
460 IF A1<1 THEN LET A1=1
470 IF A1>17 THEN LET A1=17
480 IF A2<1 THEN LET A2=39
490 IF A2>39 THEN LET A2=1
500 PRINT AT A1,A2;"X": GO TO 3
10: REM Grafico I
510 PRINT INK 2;AT A1-1,A2-1;"
█";AT A1,A2-1;"█";AT A1+1,A2-
1;"█"
520 FOR Y=1 TO D: IF ABS (L(Y,1
)-A1)>1 OR ABS (L(Y,2)-A2)>1 THE
N GO TO 540
530 GO TO 570
540 NEXT Y
550 PRINT AT A1-1,A2-1;"█";AT
A1,A2-1;"█";AT A1+1,A2-1;"█"
560 GO TO 310
570 LET L(Y,1)=L(D,1): LET L(Y,
2)=L(D,2): LET N(Y)=N(D): LET X$
(Y)=X$(D): LET D=D-1: LET SC=SC+
10
580 BEEP .15,15
590 IF D>0 THEN GO TO 640
600 LET H=H+1
610 PAUSE 500
620 CLS: PRINT AT 5,5; FLASH 1
█ PONTUACAO - ";SC: PAUSE 100
630 LET U=0: FOR X=1 TO 5: IF Z
$(X,2)<>E# THEN LET U=U+1
```

```

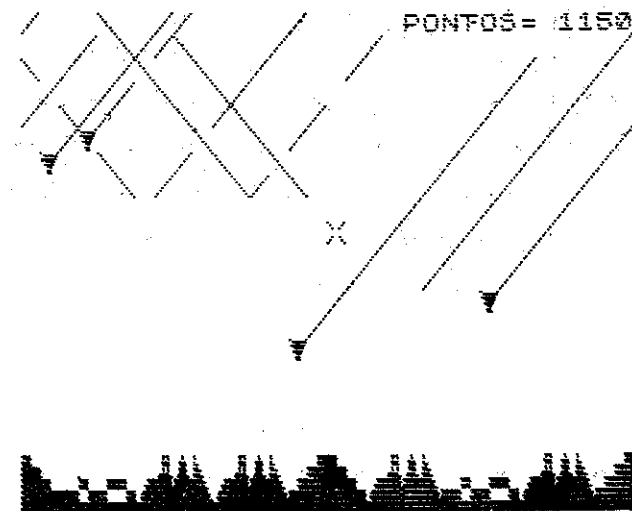
640 NEXT X
650 IF W=0 THEN GO TO 655
660 PRINT AT 10,0): FOR X=1 TO
U: PRINT AT 10,4*(X-1);R#);AT 11,
4*(X-1);W#; NEXT X
670 LET T=W*100: PRINT AT 11,(W
)+4);" = BONUS ";T: LET SC=SC+T
680 GO SUB 970
690 GO SUB 920: LET U=U+1: IF U
>15 THEN LET U=15
700 LET D=U: GO TO 695
710 IF L(B,1)<18 THEN GO TO 660
720 IF SCREEN# (L(B,1),L(B,2))<
>" THEN GO TO 680
730 LET Z=L(B,2)
740 IF Z>1 AND Z<6 THEN LET Z#(
1,1)=" ": LET Z#(1,2)=R#: GO
TO 840
750 IF Z>6 AND Z<10 THEN LET Z#(
2,1)=" ": LET Z#(2,2)=R#: GO
TO 840
760 IF Z>9 AND Z<14 THEN LET Z#(
3,1)=" ": LET Z#(3,2)=R#: GO
TO 840
770 IF Z>17 AND Z<22 THEN LET Z
#(4,1)=" ": LET Z#(4,2)=R#: G
O TO 840
780 IF Z>21 AND Z<25 THEN LET Z
#(5,1)=" ": LET Z#(5,2)=R#: G
O TO 840
790 IF Z>25 AND Z<30 THEN LET Z
#(6,1)=" ": LET Z#(6,2)=R#: G
O TO 840
800 LET L(D,1)=L(B,1): LET L(D,
2)=L(B,2): LET N(D)=N(B): LET X#(
D)=X#(B): LET D=D-1
810 IF D>0 THEN GO TO 660
820 IF W=0 THEN GO TO 650
830 GO TO 600
840 BEEP .15,2: GO SUB 920: GO
SUB 950: GO TO 800
850 CLS : FLASH 1: FOR X=1 TO 2
2: PRINT "=====
=====": NEXT X: PRINT AT 10
,9;"JOGO TERMINADO": FLASH 0
860 PRINT
870 FLASH 0: IF SC>=H5 THEN PRI
NT " OPTIMO!!
CONSEGUIU A MELHOR PONTUACA
O!! ": LET H5=SC: INPUT "INDIQUE
O NOME: ";N#
880 PRINT : PRINT PAPER 0; INK

```

```

7;"PONTUACAO MAXIMA:");N#
890 PRINT : PRINT FLASH 1;"CARA
EGUE EM B PARA JOGAR DE NOVO"
900 IF INKEY#<>"B" THEN BEEP .3
,1: GO TO 950
910 LET SC=0: GO TO 20
920 LET R#=" "+Z#(1,1)+Z#(2,1)
+Z#(3,1)+" "+Z#(4,1)+Z#(5,1)+
Z#(6,1)+"
930 LET R#=" "+Z#(1,2)+Z#(2
,2)+Z#(3,2)+" "+Z#(4,2)+Z#(5,
2)+Z#(6,2)+"
940 RETURN
950 PRINT AT 10,0): INK 4;0#(1 TO
2); INK 3;0#(3 TO 14); INK 4;0#(
15 TO 18); INK 3;0#(19 TO 30);
INK 4;0#(31 TO 34); INK 3;0#(35
TO 46); INK 4;0#(47 TO 58); INK
3;0#(59 TO 62); INK 4;0#(63 TO )
960 RETURN
970 PRINT AT 20,13;"NIVEL ";H
980 FOR F=1 TO 200: NEXT F: CLS
990 RETURN

```



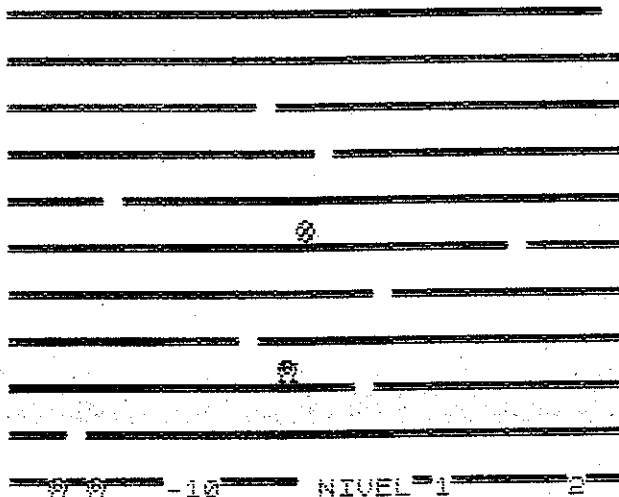
BEM ALTO

Paul Toland apresenta aqui um excitante jogo de perícia, que obrigará o leitor a saltar diversos andares através de orifícios em movimento. A medida que se atinge o topo passa-se ao «screen» seguinte. Existem cinco «screens» diferentes, e o leitor poderá observar nas imagens que se seguem os objectivos a cumprir em cada um deles:

HI-RIZE

NIVEL 1

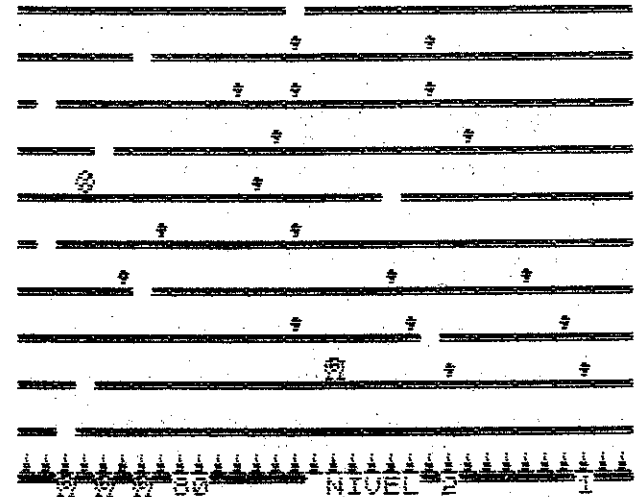
O seu objectivo consiste em atingir o topo a fim de passar ao nível seguinte. Sobee saltando pelos intervalos das linhas superiores e não caindo pelas da linha onde se encontra. Deve também evitar o garrafal que se move nas linhas, senão perde uma vida. Mova-se para a esquerda e a direita usando as teclas S e O, e salte com a tecla I.



HI-RIZE

NIVEL 2

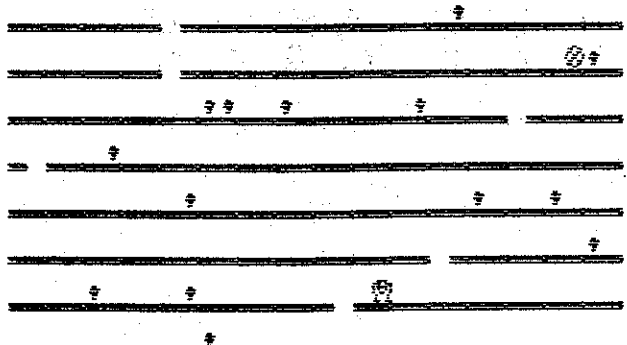
A partir deste nível encontrarão pontos vermelhos e ganhará pontos ao passar sobre eles. Neste nível não pode voltar para a linha na base.



HI-RIZE

NIVEL 3

Enquanto sobe, as linhas por baixo de si desaparecem, pelo que perderá uma vida se cair.



W W W 170

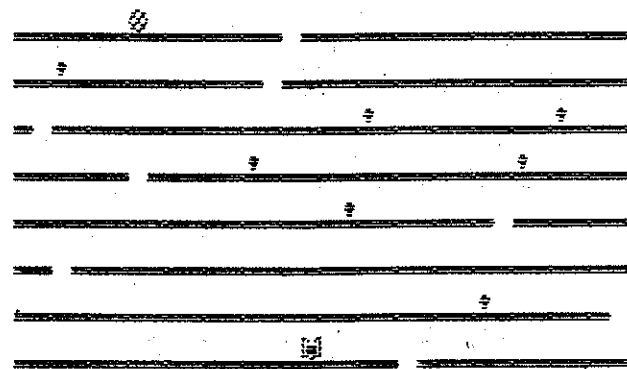
NIVEL 3

4

HI-RISE

NIVEL 4

Tudo vai ser mais rapido daqui em diante.



W W W 370

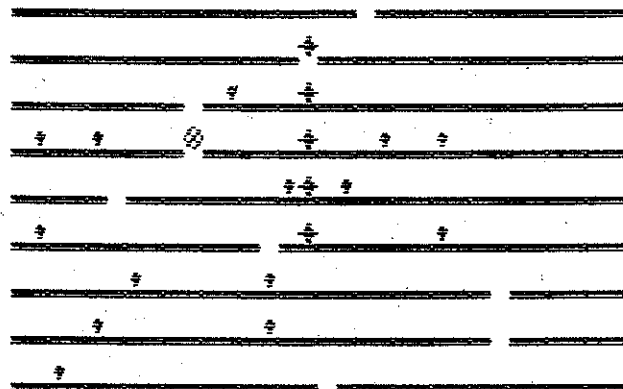
NIVEL 4

5

HI-RISE

NIVEL 5

O toque dos picos na parte superior e' letal...



W W 270

NIVEL 5

11

Apresentamos em seguida a listagem deste programa:

```

1 REM BEM ALTO © P. TOLAND
20 GO SUB 200: GO SUB 2500
30 LET TOT=TOT+1: LET UN=UN+2:
GO SUB 920
100 FOR I=H TO L: PRINT AT I*2-
1,H(I,1);CHR$(144): LET N=H(I,1)+
H(I,2): IF N<0 OR N>31 THEN LET
A=INT (RND+.5): LET H(I,2)=A-(A=
0): LET N=(H(I,2)=-1)*31
110 PRINT AT I*2-1,N;" ": LET H

```

```

(I,1)M: NEXT I
120 PRINT AT Y,X;" " IF SCREEN#
# (Y+1,X)="" THEN LET G=SC-20*
G: PRINT AT 21,0;SC;CHR# 144: BEEP
.1,-20: LET Y=Y+2: LET C=145:
LET T=0: IF L<10 THEN LET L=L+1
: IF G>2 THEN GO TO 300
130 LET T=T+1: IF T=UN THEN LET
C=145: BEEP .05,30
140 IF C=146 THEN GO TO 155
145 LET I#=INKEY#: IF I#>"1" T
HEN GO TO 160
146 IF SCREEN# (Y-1,X)="" THEN
BEEP .05,0: LET C=146: LET T=0:
GO TO 160
147 IF Y<20 THEN LET L=L-1: IF
L<10 AND G>2 THEN PRINT AT Y+1,0
;0#
148 LET SC=SC+10*G: PRINT AT 21
,0;SC;CHR# 144: BEEP .05,20: LET
Y=Y-2: LET X=X-H(Y/2+1,2)
150 LET X=X+(I#="0")-(I#="5"):
LET X=X+(X=-1)*32-(X=32)*32
154 IF SCREEN# (Y+1,X)="" THEN
GO TO 160
155 LET N=ATTR (Y,X): IF N=50 T
HEN LET SC=SC+10*G: BEEP .05,0:
PRINT AT 21,0;SC;CHR# 144
157 IF N=40 OR N=51 THEN GO TO
310
160 IF G>3 THEN LET H=Y/2+(Y/2=
0)
162 PRINT INK 0;AT Y,X;CHR# 0:
IF Y=0 THEN LET G=G+1: RESTORE 2
900: FOR I=1 TO 13: READ N: BEEP
.1,N: NEXT I: GO TO 90
170 PRINT AT BY,BX;" " : LET BX=
BX+D: LET BX=BX+(BX=-1)*32-(BX=0
0)*32: LET A#=SCREEN# (BY+1,BX):
IF SCREEN# (BY+1,BX-D)="" OR A
#="" THEN LET BY=BY+2: LET D=-H
(BY/2+1,2): IF BY>11 OR BY>Y THE
N LET BY=0: LET BX=16: LET D=-H(
1,2)
175 LET N=ATTR (BY,BX): IF N=40
THEN GO TO 310
177 IF N=51 THEN LET D=-D: LET
BX=BX+D
180 PRINT INK 0;AT BY,BX;CHR# 1
47
185 IF G=2 AND Y<20 THEN PRINT
INK 0;AT 20,0;N#

```

```

190 GO TO 195
200 FOR I=L*2 TO 20: PRINT AT I
X;CHR# 140;AT I-1,X)"" : BEEP .
1,20-1: NEXT I
310 BEEP 2,-10: LET M=M-1: IF M
=0 THEN INPUT "OUTRO JOGO?" ;A#:
GO TO 1+(A#="N")*9999+(A#="Y")*
9999
320 LET G=G-1+(G=1)
330 GO TO 900
350 RESTORE 950: LET T#="" : FOR
I=1 TO 63: READ N: LET T#=#+CH
R# (N+128): NEXT I
310 DATA 0,14,13,0,7,11,0,10,0,
11,0,0,7,11,0,0,0,11,0,11,0,0,10
,0,0,0,10,0,0,0,10,0,0,0,10,0,0,
0,0,0,11,0,0,10,0,0,10,14,0,0,0,
10,0,0,10,13,14,0,10,10,0,14,10,
000 LET T#="" "+T#(1 TO 20)+""
""+T#(20 TO 42)+""
""+T#(43 TO )
900 RESTORE 950: FOR I=0 TO 65:
READ N: POKE USR "0"+I,N: NEXT
I
905 LET N=0: DIM B#(32): RANDOM
IZE : LET TOT=0: LET UN=0: LET G
C=0: LET M=4: LET G=1: RETURN
910 DATA 205,205,205,0,0,0,0,0
911 DATA 60,231,100,155,150,00,
00,201
912 DATA 195,155,120,100,160,10
0,201,24
913 DATA 24,192,102,153,150,102
,100,24
914 DATA 0,0,24,00,00,24,24,0,0
0
915 DATA 16,16,16,00,00,00,204,
24
916 DATA 24,24,00,255,00,24,24,
24
920 LET N#="" : LET L#="" : FOR I
=1 TO 32: LET N#=N#+CHR# 140: LE
T L#=L#+CHR# 144: NEXT I
930 IF P#="0" THEN GO SUB 950
940 LET H=1: LET L=10: LET T=10
0: LET X=16: LET Y=26: LET C=145
950 INK 1: PAPER 0: BORDER 0: C
L3
960 LET BX=16: LET BY=0
970 IF G>1 THEN FOR I=1 TO 20:
PRINT INK 2;AT INT (RND*10)*2,RN
D*31;CHR# 148: NEXT I
980 IF G>4 THEN FOR I=2 TO 10 0

```

```

1000 PRINT AT 1,15: INK 3;CHR#
1100 NEXT I
1200 DIM H(10,2)
1300 FOR I=1 TO 21 STEP 2: PRINT
AT I,0;L#: NEXT I
1400 FOR I=1 TO 10: LET H(I,1)=I
NT (RND*31): LET A=INT (RND+.5):
LET H(I,2)=A-(A=0): NEXT I
1500 LET D=-H(1,2)
1600 PRINT AT 21,0;90;AT 21,15:"
NIVEI";CHR# 144;G;AT 21,20;TOT:
FOR I=1 TO H-1: PRINT AT 21,I*2
;CHR# 144; NEXT I
2000 RETURN
2100 DATA 2,2,2,14,14,14,11,11,1
1,7,11,7,2
2200 BORDER 1: INK 7: BORDER 1: C
L: INPUT ": PRINT AT 10,0:"Des
eja instrucoes? "
2300 LET I=0
2400 LET I#=INKEY#: PRINT AT 0,0
; INK 1;T#: IF I#="" THEN LET I=
I+1-(I=0)*3: GO TO 2510
2500 LET P#="N": IF I#="S" OR I#
="E" THEN LET P#="O"
2600 RETURN
2700 LET I=0: INK 7: PAPER 1: CL
S: BORDER 1: INPUT ": PRINT AT
4,10: INK 0:" NIVEL "16: GO SUB
28100+28500
2900 LET I#=INKEY#: PRINT AT 0,0
; INK 1;T#: IF I#="" THEN LET I=
I+1-(I=0)*3: GO TO 2950
3070 BORDER 15,10: RETURN
3080 PRINT AT 0,0:" O seu object
ivo consiste em atingir o top
o e fim de passar ao nivel segu
inte. Deve saltar
o pelos intervalos das linhas su
periores e nao caindo pelos
de linha onde se encontra.
Deve tambem
evitar o barril que se move das l
inhas, sendo perde uma vida."
3090 PRINT " Move-se para a esqu
erda e a direita usando as tecl
as S e O, e salta com a tecla I"
: RETURN
2700 PRINT AT 0,0:" A partir des
te nivel encontrara pontos vermel
hos e ganha pontuacao passando
sobre eles."

```

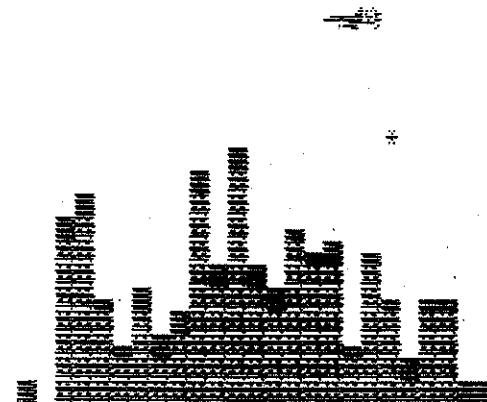
```

2710 PRINT " Neste nivel nao pod
e voltar pa-ra a linha da base."
: RETURN
2800 PRINT AT 0,0:" Enquanto sob
e, as linhas por baixo de si d
esaparecem, pelo que perdera u
ma vida se cair." : RETURN
2900 PRINT AT 10,0:" Tudo vai se
r mais rapido daqui em diante."
: RETURN
3000 PRINT AT 10,0:" O toque dos
picos na parte su-perior e le
tal...": RETURN
3000 RETURN
3000 INK 0: PAPER 7: BORDER 7

```

BOMBARDEIRO

Nenhuma colecção de jogos para computador poderia estar completa sem um do tipo «Bombardeiro». Esta versão é bastante melhor do que muitas outras, e constitui um verdadeiro desafio para o jogador. Vejamos um exemplo de execução:



O cenário é familiar. Você encontra-se numa nave espacial que desce sobre a superfície de um planeta. Não dispõe de meios para subir novamente, e deve portanto «fabricar» a bomba uma

Você controla a sua base de lasers usando as teclas «Z» e «Y», além da tecla SPACE para disparar. Conseguirá 50 pontos por cada pássaro que atingir na descida, e 25 se o pássaro se encontrar na linha inferior. No entanto, cada pássaro atingido não será morto, apenas atordoado, e depressa voltará ao combate.

Você começa com três bases de lasers e perderá uma cada vez que for atingida por um pássaro ou uma bomba. O número de bases é indicado na linha superior da imagem, juntamente com a sua pontuação (à esquerda) e a pontuação máxima actual. Ganhará uma base extra quando tiver destruído três ondas de atacantes.

Neste ponto, você ver-se-á ameaçado por alienígenas mutantes. Estes deslizam como os pássaros, mas podem subir, largando mais bombas.

Para você ter um objectivo, talvez lhe interesse saber a pontuação máxima obtida por Neil Pellinacci — e só depois de bastante prática: 12 000.

O programa está escrito inteiramente em Basic, exceptuando uma rotina em código-máquina localizada no endereço 32500. Apresentamos em seguida uma listagem das instruções em código-máquina:

```

10 LD HL,784
20 LD DE,22525
30 LD BC,32
40 PUSH HL
50 ADD HL,DE
60 DBC HL
70 LD A,(HL)
80 C 405
90 JR NZ,32539
100 POP HL
110 DBC HL
120 LD B,H
130 OR L
140 JR NZ,32569
150 LD HL,32563

160 LD B,32
170 LD A,(HL)
180 CP 125
190 JR NZ,32536
200 DBC HL
210 DJNZ 32523
220 LD C,0
230 RET
240 LD (HL),78
250 ADD HL,BC
260 LD (HL),125
270 LD 32517

```

Depois de terminar a introdução do programa, talvez lhe interesse alterar o endereço onde se encontra o código-máquina. Se assim acontecer não se esqueça de alterar as linhas 90, 317, 412 e 7020. A subrotina determina a posição de cada bomba e desloca-a para baixo uma posição de cada vez. Verifica ainda se o planeta se encontra já completamente coberto de bombas. O resultado deste teste é detectado ao voltar ao Basic na linha 317.

Deve introduzir o programa e em seguida gravá-lo em fita, antes de tentar executá-lo para o caso de ter cometido algum erro na escrita dos valores que constituem o código-máquina. É também conveniente gravar o programa de tal modo que entre automaticamente em execução ao ser carregado no computador.

```

10 RANDOMIZE : GO TO 90
20 PRINT AT 0,8-LEN STR# 0; IN
K 0;0: RETURN
30 LET BC=BC+(INKEY#="X" AND B
C<27)-(INKEY#="Z" AND BC>0): PR
INT AT 0,BC; PAPER 0; INK 0; "
40 IF INKEY#<>" " THEN RETURN
50 GO TO 520
60 FOR X=1 TO 3: FOR Y=1 TO 10
70 IF L(X,Y)<>-1 THEN LET L1=L
(X,Y): LET C1=C(X,Y): LET C2=Y:
LET L2=X: RETURN
80 NEXT Y: NEXT X: LET WF=1: R
ETURN
90 FOR X=20 TO 7+INT (RND*10)
STEP -1: PRINT AT X,BC+2; PAPER
7;" " : BEEP .001,40-X: PRINT AT
X,BC+2; PAPER 0;" " : NEXT X: PR
INT AT X,BC+2; INK 0; PAPER 7;" "
: RETURN
90 CLEAR 32499
95 BORDER 0: PAPER 0: CLS : BC
ORDER 0
100 INK 4: BRIGHT 1: PRINT AT 0
,25; INK 0;"000000"
105 GO SUB 7000: GO SUB 9000
107 LET H3=0
110 GO SUB 3000: LET BC=13: LET
S=0: LET LB=3: LET UU=0
120 LET FF=0
130 LET HT=0
140 LET BFF=0
150 LET WF=0
200 DIM L(3,10): DIM C(3,10)
205 FOR B=0 TO 2

```

```

210 FOR A=0 TO 9
220 LET L(B+1,A+1)=B*2+1: LET C
(B+1,A+1)=A*3+1
230 NEXT A
235 NEXT B
237 PRINT AT 0,0; INK 5;"000000
"; AT 0,17; LB
: REM GRAFICOS BFG
240 FOR B=1 TO 3
250 FOR A=1 TO 10: PRINT AT L(B
,A), C(B,A); "A": NEXT A: REM GRA
FICOS CD
255 NEXT B
260 LET BH=0
265 FOR K=1 TO INT (AND*15+1)
280 LET B=INT (AND*3+1): LET A=
INT (AND*10+1)
285 IF L(B,A)=-1 THEN GO TO 315
310 PRINT AT L(B,A), C(B,A); "X"
: BEEP .01, -10: PRINT AT L(B,A),
C(B,A); "A": BEEP .01, 5: REM GRA
FICOS AB e CD
312 IF AND>.6 THEN IF ATTR (22,
BC+2)=126 THEN GO SUB 30
315 IF INKEY$<>" " THEN GO SUB 3
0
317 IF NOT USA 32500 THEN GO TO
4000
318 IF ATTR (21,BC+1)=126 OR AT
TR (21,BC+2)=126 OR ATTR (21,BC+
3)=126 THEN GO TO 1000
320 NEXT K
330 IF BH>20 THEN GO SUB 50: IF
WF=0 THEN GO TO 405
340 IF WF=1 THEN GO TO 2000
400 LET L2=INT (AND*3+1): LET C
2=INT (AND*10+1)
402 LET L1=L(L2,C2): LET C1=C(L
2,C2)
404 IF L1=-1 THEN GO TO 400
405 PRINT AT L1,C1; " "
406 LET CD=INT (AND*3-1)
410 PRINT AT L1,C1; OVER 1;"A"
: BEEP .001,L1+20: PRINT OVER 1;
AT L1,C1;"A": REM GRAFICOS CD
411 IF AND>.8 THEN IF L1>6 THEN
PRINT AT L1+1,C1; INK 6; PAPER
7; " "
412 LET ZZ=USA 32500: IF INKEY$
<>" " THEN GO SUB 30: IF HF=1 THE
N LET HF=0: GO TO 290
413 IF ATTR (21,BC+1)=126 OR AT

```

```

TR (21,BC+2)=126 OR ATTR (21,BC+
3)=126 THEN GO TO 1000
415 LET C1=C1+CD: IF C1=32 THEN
LET C1=0
416 IF C1=-1 THEN LET C1=30
418 IF AND>.35 THEN LET CD=INT
(AND*3-1)
419 IF WF>=3 THEN IF AND>.6 THE
N IF L1>8 THEN LET L1=L1-2
420 LET L1=L1+1: IF L1<>21 THEN
GO TO 410
425 IF C1=BC+1 OR C1=BC+2 THEN
LET BFF=1: GO TO 1000
430 PRINT AT L(L2,C2), C(L2,C2);
"X": GO TO 200: REM GRAFICOS AB
520 PLOT BC*8+20,3: DRAW INK 3,
PAPER 0,0,128: PLOT OVER 1,BC*8
+20,3: DRAW OVER 1,0,128
530 IF BC+2=C1 OR BC+1=C1 THEN
GO TO 510
540 IF SCREEN$(5,BC+2)<>" " TH
EN PRINT AT 5,BC+1;" " : LET S=
S+25: GO SUB 20
550 RETURN
610 PRINT AT L1,C1; INK 7;"X"
: AT L1,C1; " " : LET C1=-1: LET L1
=-1: BEEP .05, -30: LET L(L2,C2)=
-1: LET S=S+50: GO SUB 20: REM G
RAFICOS KL
520 LET HF=1: LET BH=BH+1: RETU
RN
1000 REM BANG
1010 FOR Y=1 TO 4: FOR Z=1 TO 10
: BEEP .015,20-Z: PRINT AT 21,BC
: INK AND*7; "X": NEXT Z: REM
GRAFICOS HIJ
1020 FOR Z=10 TO 1 STEP -1: BEEP
.015,20-Z: NEXT Z: NEXT Y
1030 LET LB=LB-1: PRINT AT 0,17;
: INK 5; LB
1040 IF LB=0 THEN GO TO 1050
1045 IF BFF=1 THEN LET BFF=0: GO
TO 430
1050 GO TO 415
1055 PRINT AT 0,17; INK 1; PAPER
7; FLASH 1; LB
1070 FOR X=1 TO 20: PRINT AT X,0
: " " : NEXT X
1080 BEEP .2,50: BEEP .25,30
1090 FOR X=1 TO 7: PRINT AT X+7,
0; INK X; " " : JOGO TERMINAD

```

```

0
1100 BEEP .2,50: BEEP .25,30
1270 IF S=H5 THEN LET H5=0
1280 PRINT AT 0,32-LEN STR# H5;
INK 5;H5
1290 IF S=H5 THEN FOR Z=1 TO 5:
FOR X=20 TO -20 STEP -2: BEEP .0
1,X: NEXT X: NEXT Z
1300 FOR X=0 TO 20: BEEP .005,20
: NEXT X
1305 FOR X=0 TO 20: BEEP .005,0:
NEXT X
1310 FOR X=1 TO 300: NEXT X: BOR
DER 0: CLS: BORDER 0: PRINT AT
0,25: INK 5: "000000": AT 0,32-LEN
STR# H5: INK 5;H5: GO TO 110
1400 STOP
2000 REM FINAL DA ONDA
2010 FOR X=1 TO 20: PRINT AT X,0
:
: NEXT X
2020 FOR X=1 TO 7: PRINT AT 3+X*
2,2: INK X): " ONDA DE ATROQUE D
ESTRUIDA "
2030 NEXT X
2040 PAUSE 150: BEEP .2,50: FOR
X=-30 TO 35: BEEP .01,X: BEEP .0
15,20: NEXT X: BEEP .2,50
2050 FOR X=1 TO 20: PRINT AT X,0
:
: NEXT X
2060 LET UU=UU+1: IF UU=3 THEN 0
ORDER 0: PAPER 0: CLS: BORDER 0
: PRINT AT 0,25: INK 5: "000000":
AT 0,32-LEN STR# H5: INK 0;H5: 0
BEEP .2,10: BEEP .2,-10: LET LB=L
B+1
2070 GO TO 120
3000 REM PAGINA DE TITULO
3010 RESTORE 3900
3020 READ X: IF X=999 THEN GO TO
3100
3040 READ Y: PLOT X,Y
3050 READ X: IF X=-999 THEN GO T
O 3020
3060 READ Y: DRAW X,Y: GO TO 305
0
3100 PRINT AT 10,4: INK 5: "© NEI
L PELLINACCI 1983"
3110 PRINT AT 13,7: INK 2: PAPER
7: "C O M A N D O S"
3120 PRINT AT 15,2: INK 6: "Z...".

```

```

3130 PRINT AT 17,2: INK 6: "X...".
3140 PRINT AT 19,2: INK 6: "Y...".
3150 PRINT AT 21,0: INK 2: PAPER
7: "CARREGUE NUMA TELA."
3160 INK 7: PLOT 0,0: DRAW 0,50:
DRAW 50,0: PLOT 170,70: DRAW 77
,0: DRAW 0,-50: DRAW -45,0: PLOT
0,0: DRAW 45,0
3165 INK 2: PLOT 0,00: DRAW 255,
0: DRAW 0,10: DRAW -255,0: DRAW
0,-10
3170 INK 4
3200 FOR X=30 TO -30 STEP -1: BE
EP .01,X: BEEP .01,-X: NEXT X: P
AUSE 0
3210 FOR X=1 TO 21: PRINT AT X,0
:
: NEXT X
3220 RETURN
3400 RETURN
3500 DATA 35,110,0,50,30,-14,-30
,-14,30,-14,-30,-14,-300
3510 DATA 35,110,0,50,-300
3520 DATA 100,110,0,30,30,-14,-3
0,-14,30,-20,-300
3530 DATA 101,110,0,50,30,-20,-3
0,-20,-300
3540 DATA 100,110,30,20,-30,0,30
,20,-300
3700 DATA 000
4000 REM ATERRA
4010 BEEP .2,0: BEEP .15,-1: BEE
P .25,0
4020 FOR Z=1 TO 20
4030 FOR X=1 TO 3: FOR Y=1 TO 10
4040 IF L(X,Y)<0 OR L(X,Y)>20 TH
EN GO TO 4055
4050 PRINT AT L(X,Y),C(X,Y): INK
3: " " : LET L(X,Y)=L(X,Y)+1: PR
INT AT L(X,Y),C(X,Y): INK 3: " "
: REM GRAFICOS CD
4060 NEXT Y: NEXT X
4070 NEXT Z
4080 FOR Z=20 TO -30 STEP -3: BE
EP .01,Z: BEEP .01,Z*1.5: BEEP
.01,-Z+10: NEXT Z
4090 BEEP .15,-5: BEEP .15,-10:
BEEP .15,1
4095 FOR B=1 TO 5

```

```

4100 FOR N=1 TO 10 STEP 2: PRINT
   AT N,0: INK AND#0+1:
5000 ATERRARRRM.
41000 NEXT N
41000 NEXT B
40000 BERT 2, -10
40000 GOTO 10000
40000 GOTO
70000 BERT M/O
70000 BERTORE 70000: FOR B=1 TO 40
   BERTORE DOKE 0,0,40,700,0: NEXT B
70000 DATA 0,0,0,0,0,0,17,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
70400 DATA 0,0,0,0,0,0,14,0,0,0,0,0,0
70800 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
12000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
70800 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0027
00000 RETURN
00000 FOR C=0 TO 00: READ B: DOKE
   DATA+0,0: NEXT C
00010 DATA 0,0,0,0,0,0,7,1,0,0,0,0
00020 DATA 0,0,0,0,0,0,4,0,0,0,0,0,0,0
00030 DATA 0,0,0,7,0,0,14,0,0,0
00040 DATA 0,0,0,0,0,0,4,0,0,0,0,0,0,0
00050 DATA 0,0,0,0,0,4,14,0,0,0
00060 DATA 0,0,0,0,0,0,0,0,0,7,0,0,0,2
00070 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
00080 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
00090 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
00100 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
00110 DATA 100,70,0,0,0,0,0,0,0,0,0,0,0,0
00120 DATA 100,0,0,0,0,0,0,0,0,0,0,0,0,0
00000 RETURN

```

Vejamos agora a função de cada parte do programa:

- 10 — elimina as subrotinas seguintes, que se encontram no principio a fim de permitir ao programa ser executado a uma velocidade máxima.
- 20 — Subrotina que imprime a pontuação.
- 30 — Desloca a base de lasers. Se se carrega na tecla SPACE, o comando é enviado para a linha 520.

- 50 — Início da subrotina que só é chamada quando restam apenas alguns pássaros. Localiza o primeiro vivo, e selecciona-o para atacar. Se não restarem pássaros, passa WF para 1 a fim de indicar ao ciclo principal o que se passou
- 80 — Uma rotina para dificultar o jogo.
- 90 — Define a área que será ocupada pelo código-máquina.
- 100 — Inicialização do jogo.
- 120/160 — Define os pássaros.
- 200/235 — Define os pássaros.
- 240/255 — Imprime os pássaros.
- 290/320 — Leva alguns dos pássaros a baterem as asas aleatoriamente.
- 317 — Verifica estado das bombas; move-as para baixo.
- 318 — Verifica se há colisões entre a base bomba.
- 330 — Selecciona o pássaro seguinte sem recorrer à rotina de selecção aleatória, ver a subrotina 50.
- 340 — Verifica se terminou a onda atacante.
- 400/404 — Selecciona um pássaro aleatoriamente, e verifica se existe.
- 410/420 — Ciclo principal.
- 411 — Larga uma bomba.
- 419 — Se o pássaro é um mutante, pode movê-lo para cima.
- 520 — Rotina de disparo.
- 610/620 — Explosão do pássaro.
- 1000/1050 — Rotina de base atingida.
- 1060/1310 — Rotina de jogo terminado.
- 2000/2070 — Rotina de finalização da onda atacante.
- 2060 — Início das ondas de mutantes.
- 3000/3210 — Rotina da página de título.
- 3500/3700 — Dados para a página de título.
- 4000/4300 — Esta rotina é chamada quando o planeta se encontra completamente coberto de bombas; leva os pássaros a mudarem de cor e a aterrarem lentamente.
- 7020 — Carrega o código-máquina.
- 7030/7060 — Dados para o código-máquina.
- 9000 — Define os UDG's.
- 9010/9020 — Dados para os caracteres gráficos.

Finalmente, vejamos uma lista das principais variáveis usadas no programa:

- HS — Pontuação máxima
- BC — Posição da base
- S — Pontuação
- LB — Bases restantes
- WW — Número da onda atacante
- HF — Flag de tiro (1 — atingido; 0 — não)
- WF — Flag de final de onda atacante (1 — final; 0 — não)
- L(3,10), C(3,10) — Posições dos 30 pássaros.
- BH — Número de pássaros atingidos
- LI, CI — Posição dos pássaros em ataque
- CD — Direcção horizontal dos pássaros

Tal como acontece com a maioria dos programas apresentados neste livro, existem muitas coisas que podem ser alteradas de acordo com os desejos do utilizador. Por exemplo, os números aleatórios podem ser alterados, modificando-se assim a probabilidade de ocorrência de diversos acontecimentos (é o que se pode fazer actuando, por exemplo, nos valores utilizados nas linhas 411 e 419). Note-se que é igualmente possível alterar as imagens apresentadas em várias fases do jogo.

HELICÓPTERO

Este jogo de Malcolm Young tem bastante acção. Você tripula um helicóptero de salvamento voando sobre o deserto.

Terá de salvar tantos soldados quantos puder antes de ficar sem combustível. Pode recolher um soldado quer aterrando sobre ele (o que lhe dará 50 pontos) ou, para tornar a operação mais fácil e rápida, descendo o gancho do helicóptero quando se encontra exactamente dois espaços acima do soldado. Neste caso obterá 25 pontos.

Pode igualmente usar o gancho para eliminar árvores. No entanto, ao fazê-lo você reduz a cobertura do helicóptero. Esta é importante porque lhe permite esconder-se dos inimigos que se deslocam em jeeps e disparam sobre si. Estes só podem disparar na vertical, e você pode destruir os jeeps largando bombas sobre eles. Você só os pode bombardear enquanto se encontra no céu, e fá-lo-á carregando também na tecla 1 (o programa sabe se você quer largar uma bomba ou o gancho, o que permite usar a mesma tecla para ambos os casos).

Depois de recolher um homem, a ambulância torna-se negra e você ganhará um bônus de 100 pontos se colocar o soldado nessa ambulância.

Para aumentar o número de coisas que você deve ter em conta ao jogar este programa, verificará que por vezes surge um camião-tanque na parte inferior da imagem. Se conseguir bombardeá-lo, ganhará vinte unidades de combustível extra, além de 10 pontos.

Este programa termina quando você fica sem combustível para o helicóptero ou quando tiver perdido todas as vidas de que dispõe inicialmente. No final do programa é apresentada uma tabela de pontuações, que você poderá ampliar facilmente até cinco posições.

```

100 SUB 500
110 DIM a=(5,10): DIM b(5)
120 LET t1=0: LET s1=0: LET d1=0
130 LET s2=0: LET pk=0: LET vidas=0
140 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: C15
150 LET y1=0: LET u=100: LET c
comb=200
160 BORDER 6: PAPER 6: INK 0: C
170 LET x#="": LET y#="": L
ET a#="": LET b#="": LET g#="": LET t#="
180 FOR g=11 TO 21: PRINT PAPER
0: AT 0,0: "": NEXT g
190 FOR t=0 TO 31: PAPER 0: LET
r=INT (RAND*4): PRINT INK 0: AT 1
0-r,t;"Y": INK 4: AT 15-0,t;"#":
NEXT t
200 LET u=INT (RAND*10)+1: LET x
=INT (RAND*30): LET a=1
210 PRINT PAPER 0: AT 11,a-1: "
: INK 2: AT 11,a: a#
220 LET comb=comb-1: PRINT PAPER
0: INK 7: BRIGHT 1: AT 0,0;"COM
b=": comb; " : AT 0,10;"PONT.=": sc
: AT 0,20;"VIDAS=": vidas
230 IF comb<0 THEN GO TO 1000
240 LET a=a+0.5 AND (a<30): IF
NOT a THEN LET pk=a
250 PRINT OVER 0: AT 4,x;"
260 IF u=11 AND x+1=a THEN LET
sc=sc+50: LET pk=0: BEEP .5,.5

```

```

0000 LET U=U+(INKEY#="0")-(INKEY
#="7"): LET X=X+(INKEY#="0")-(IN
KEY#="9")
0000 IF U<11 THEN LET X=X+(INKEY
#="0")-(INKEY#="9")
0010 LET U=U+(U<9)-(U>9): LET X
=X+(X<9)-(X>9)
0020 LET C=BITR(U,X): IF C<>49
AND C<>48 THEN GO TO 1000
0030 LET C=BITR(U,X+1): IF C<>4
8 AND C<>48 THEN GO TO 1000
0040 PRINT OVER S;AT U,X;C#;BEM
P 1,-1;S: PRINT OVER U;AT U,X;C#
0040 IF U<.3 AND NOT PK THEN L
ET U=U
0040 IF S THEN GO SUB 400+U
0050 IF U<.3 THEN LET J=1
0060 GO TO J*10+170
0070 LET J=10+INT(RND*4)+1
0080 LET U=100
0090 LET J4=RND: LET J4=(J4*.5)-
(.5)
0100 LET A#=# AND J4=1: LET A#
=A#+(U# AND J4=-1): IF RND<.3 GO T
HEN LET T=1
0110 IF T=1 THEN LET A#=(T#(0 T
0) AND J4=1)+(T#(0 TO 0) AND J4
=-1)
0120 LET JX=30 AND J4=-1
0130 PRINT INK 0;AT J4,0;"
0140 PRINT INK 1;AT J4,JX;A#
0150 LET JX=JX+J4
0160 IF JX=0 OR JX=31 THEN LET I
N=100: LET T=0: LET J=0: PRINT
OVER S;AT J4,0;"
0170 IF U<10 AND INKEY#="1" THEN
GO TO 010
0180 IF T=1 OR NOT (JX>INT X-1
AND JX<X+1) THEN GO TO 170
0170 LET S=U+1: FOR F=J4-1 TO S
STEP -1: LET S=BITR(F,JX+1): IF
S=48 OR S=49 THEN PRINT OVER S
; INK 0;AT F,JX+1;"": BEEP 1,1;
PRINT INK 0;AT F,JX+1;"": NEX
T F
0200 PRINT INK 0;AT F,JX+1;"
0210 IF (JX+1=X+1 OR JX+1=X) AND
T=0 THEN BEEP 1,-5: GO TO 1000
0220 GO TO 170
0230 REM transporter soldado
0240 LET S4X=INT(RND*20)+1: LET

```

```

S4X=10+INT(RND*5): LET S4=10:
LET T=INT(RND*20)+10
0300 PRINT INK 1;AT S4,S4X;"X":
BEEP 1,5;S
040 LET T=T-1: PRINT AT S4,S4X
;"": IF NOT T THEN LET S=T=0: LE
T S4=0: RETURN
050 IF U=0 AND X+1=S4X THEN L
ET S=C=0+00: BEEP 1,5: LET S=T=0:
LET S4=0: LET PK=1: RETURN
060 IF INKEY#<>"1" THEN RETURN
070 PRINT AT U+1,X+1;"1";AT U+0
,X+1;"1"
080 IF S4X=U+0 AND S4X=X+1 THEN
FOR S=U+0 TO S4X STEP -1: PRINT AT
T S,X+1;"X": BEEP 1,5: PRINT AT
S,X+1;"": NEXT S: LET PK=0: LE
T S=T=0: LET S4=0: LET S=C=0+05
090 PRINT AT U+1,X+1;"": AT U+0
,X+1;"":
0910 RETURN
0910 LET S=J4: FOR F=U+1 TO S: L
ET S=BITR(F,X+1): IF S=49 OR S=
48 THEN PRINT OVER S: INK 0;AT
F,X+1;"X": CHS# 0: BEEP 1,5;S-1:
PRINT INK 0;AT F,X+1;"": NEXT F
GO TO 090
0920 PRINT INK 0;AT F,X+1;"
0930 IF S=48 THEN BEEP 1,-5: LET
S=C=S+10: LET C=C+00+00+00*(17=
1): PRINT AT J4,JX;"": LET J
X=0: GO TO 340
0940 GO TO 170
0950 REM
0960 RESTORE
0970 DATA "e",1,1,1,255,255,255,
210,24
0980 DATA "f",15,124,0,63,63,195
,210,24
0990 DATA "g",128,128,128,255,25
5,210,24
1000 DATA "h",48,10,10,252,252,1
00,210,24
1010 DATA "i",5,15,124,254,255,1
00,210,24
1020 DATA "j",5,5,192,192,255,1,
5,5
1030 DATA "k",255,5,5,5,5,5,5
1040 DATA "l",7,5,5,5,5,5,5
1050 DATA "m",125,97,97,255,255,
100,210,24
1060 DATA "n",255,255,199,239,25

```

```

0100 DATA "K",16,16,16,16,16,16,
0200 DATA "L",108,73,110,108,114
0300 DATA "M",36,36,24,16,16,16,
0400 DATA "N",153,99,78,56,8,20,
0500 DATA "O",201,126,24,24,56,1
0600 DATA "P",126,255,255,255,25
0700 FOR a=1 TO 16: READ c#
0800 FOR b=8 TO 7: READ d: POKe
0900 c#+e,d: NEXT e
1000 NEXT c
1100 RETURN
1200 PRINT INK 0; FLASH 0; AT 4,x
1300: BEEP 1,1
1400 PRINT INK 0; FLASH 0; PAPER
0; AT 4,x; " ": LET c=INT (RND*1
0)+1: LET x=INT (RND*30)
1500 LET vidas=vidas-1
1600 IF c=1 THEN PRINT FLASH
1; TAB 3; "GASTOU TODO O COMBUSTIV
EL!": GO TO 1040
1700 IF vidas THEN GO TO 170
1800 IF sc>h(5) THEN GO TO 1000
1900 INPUT "Nome? ";n#
2000 FOR r=5 TO 1 STEP -1: BEEP
.1,r: IF sc>h(r) THEN NEXT r
2100 FOR g=5 TO r+2 STEP -1: BEE
.1,g: LET h(g)=h(g-1): LET s#(
g)=s#(g-1): NEXT g
2200 LET h(g)=sc: LET s#(g)=n#
2300 BORDER 1: PAPER 2: INK 7: B
RIGHT 1: CLS
2400 PRINT INK 0; AT 3,2; "CAMPEO
E":
2500 FOR a=1 TO 5: BEEP .5,a: PR
INT INK 0; AT 7+a,6; s#(a); " "; h(a
): NEXT a
2600 PRINT FLASH 1; PAPER 4; INK
0; AT 10,7; "DESEJA NOVO JOGO ?"
2700 IF INKEY#="Y" OR INKEY#="N"
THEN GO TO 1040
2800 IF INKEY#="" THEN GO TO 121
0
2900 GO TO 00
3000 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: STOP

```

SERPENTE

Escrito por David Perry, este programa é um jogo que exige bastante atenção visual e concentração.

Vejamos um exemplo de execução:

COBRA

INDIQUE NIVEL DE JOGO.

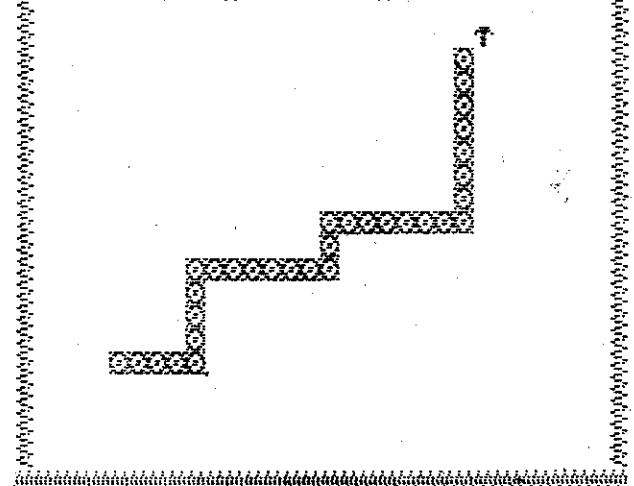
(1 -- 9)

Use as teclas O,Z,I & P para subir, descer, esquerda & direita. Sempre que comer a cobra crescerá.

Tenha cuidado em não tropeçar em si mesmo ou na sebe!

BOA SORTE!

<PONTOS: 22>M>COBRA<M<MAXIMO: 5



O objetivo consiste em conduzir a serpente num jardim — rodeado por uma sebe electricada — comendo simultaneamente os cogumelos que vão surgindo. Cada um destes leva a serpente a crescer, e você deve mover-se de tal modo que a serpente nunca toque em si mesma — senão o jogo termina.

E em seguida a listagem do programa:

```

20 GO SUB 210
30 REM ***Image# inicial***
30 LET I=0: INVERSE 0: BRIGHT
0: FLASH 0: BORDER 0: PAPER 0: I
NK 7: CLS
40 PRINT " "
50 PRINT " "
60 PRINT " "
70 PRINT " "
80 PRINT " "
90 PRINT " "
100 PRINT " "
110 PRINT AT 14,1:"Use as teclas
s 0,Z,I e P para subir, descer
e esquerda e direita. Sempre qu
e comer a cobra cresce."
PRINT " Tenha cuidado e# não fo
pacar e# si mesmo ou da sebe!"
130 PRINT: PRINT INK 0;TAB 14;
"000 0000":
140 PRINT AT 10,5: INK I:"INDIO
US NIVEL DE JOGO."
150 PRINT AT 10,10: INK I/2;("(1
-- 0)"
160 REM Definir nivel
160 LET A#=INKEY#: IF A#="" THE
N LET I=I+1: IF I>7 THEN LET I=0
170 IF A#<"1" OR A#>"0" THEN GO
TO 140
180 LET L=VAL A#
190 INK 0
200 GO TO 350
210 REM Graficos
220 FOR A#=1 TO 0
230 FOR B#=0 TO 0
240 FOR N=0 TO 7
250 FOR 0

```

```

0000 POKÉ USA B#+N,0
0001 NEXT N
0002 NEXT 0
0003 DATA "0",100,102,105,103,15
0004 DATA "0",100,100
0005 DATA "0",0,00,100,100,04,04
,04,04
0006 DATA "0",00,06,100,102,102,
100,00,00
0007 DATA "0",0,04,04,05,05,05,1
0008 DATA "0",00,06,00,0,06,06,0
0009
0010 LET I=0
0011 RETURN
0012 BORDER# 7
0013 PAPER# 7
0014 LET I=0
0015 GO TO 0
4000 CLS
4001 REM
410 PRINT AT 0,0:"
411 PRINT AT 0,0: INK 0;"<PONTO
0:0>": INK 1;AT 0,10;">": INK
;"0000": INK 1;"<#>": INK 4;AT 0
,0:;"<MAXIMO:";HS;AT 0,00;">": R
M GRAFICO 0
400 FOR N=1 TO 0
410 PRINT AT N,0: INK 1;"$
M: REM
GRAFICO E
440 NEXT N
450 PRINT AT 21,0: INK 1;"~~~~~
~~~~~": RE
M GRAFICO 0
460 LET X=10
470 LET Y=10
480 LET B#=""
490 FOR N=0 TO 10
500 LET B#="0"+B#+STR# N
510 PRINT BRIGHT 1: INK 2;AT 10
,2: "0": REM GRAFICO A
520 NEXT N
530 GO SUB 000
540 LET D#=INKEY#
545 REM ***ENTRADA#
550 IF D#="0" OR D#="A" OR D#="
" OR D#="N" THEN LET B# OR D#
560 LET X=X+(D#="N")-(D#="0")
570 LET Y=Y+(D#="0")-(D#="1")

```

```

560 IF X=21 OR X=0 THEN GO TO 8
570 IF Y=31 OR Y=0 THEN GO TO 8
580 IF ATTR (X,Y)=122 OR X=22 OR
Y=32 THEN GO TO 830
610 GO SUB 700
620 LET C$=STR$ X
630 IF X<10 THEN LET C$="0"+STR
$ X
640 LET D$=STR$ Y
650 IF Y<10 THEN LET D$="0"+STR
$ Y
660 LET X1=VAL A$(1 TO 2)
670 LET Y1=VAL A$(3 TO 4)
680 IF ATTR (X1,Y1)<>122 THEN G
O TO 700
690 PRINT AT X1,Y1;" "
700 REM **IMPRIMIR COBRAS
710 PRINT BRIGHT 1; INK 2; AT X,
Y;"0"; REM GRAFICO 0
720 LET A#=#+C#+D#
730 IF A#>9 THEN LET A#=#%10 TO
730 LET A=A-(A>9)
740 GO TO 540
750 LET M=ATTR (X,Y)
760 IF M>60 THEN RETURN
770 LET A=A+L
780 LET SC=SC+L
790 PRINT AT 0,0; INK 5;"<PONTO
0:";60;">" AND SC<100); INK 1; A
T 0;11;"M"; INK 2;"COBRAS"; INK
1;"M"; INK 4; AT 0,20;"<MAXIMO:"
JHS; AT 0,31;">"; REM GRAFICO 0
800 PRINT PAPER 7; INK 4; AT INT
(RND*10)+0,INT (RND*20)+0;"#";
REM GRAFICO 0
810 FOR N=50 TO -50 STEP -10: G
O TO 810;N: NEXT N
820 RETURN
830 LET X=X-(X=22)+(X=-1)
840 LET Y=Y-(Y=32)+(Y=-1)
850 FOR F=50 TO -50 STEP -5: G
O TO 850;F: NEXT F
860 REM ***FIM
870 IF SC>10 THEN LET HS=SC
870 PRINT AT 10,0;"PONTUACAO MA
XIMA: ";HS
880 PRINT AT 12,0;"Carreque num
0 tecla."
890 LET I=0
900 PRINT AT X,Y; INK 1;"0"; G

```

```

80 .01,I#2: REM GRAFICO 0
805 REM ***POSICAO CARRO
810 PRINT AT 0,0; INK 1; FLASH
1;">"; REM GRAFICO 1; REM "PONTOS:"
820 LET C$=INKEY$: IF C$=" " THEN
N LET I=I+1; IF I>7 THEN LET I=0
830 IF C$=" " THEN GO TO 860
840 CLS
850 GO TO 80

```

CORRIDAS

Este rápido jogo de acção foi escrito por Raymond Blake. Você conduz o seu automóvel por um percurso, passando sobre bandeiras onde ganha pontos. Ao mesmo tempo, tenta evitar o automóvel suicida comandado pelo computador, que se move em direcção oposta à sua, tentando estragar o seu divertimento...

Você pode mudar de percurso em quatro pontos, tantas pistas quantas quiser. O carro suicida também pode mudar de pista, mas apenas uma de cada vez. Apesar disto, o leitor verificará que se trata de um jogo bastante difícil.

No início da corrida, você ganha pontos passando sobre bandeiras vermelhas, que se tornam amarelas quando você passa sobre elas. Quando não tiver bandeiras vermelhas (ou quiser simplesmente mudar um pouco), pode ganhar pontos com as bandeiras amarelas desde que passe sobre a bandeira magenta. Mas isto dará ao seu opositor Kamikaze uma possibilidade de o apanhar.

Aparecerá então uma outra bandeira magenta noutra ponta do percurso, e passando sobre ela modificará novamente a cor das bandeiras que dão pontos.

```

20 FOR I=1 TO 5: READ Z#: FOR
J=0 TO 7: READ Z: POKE USA Z#+J,
Z: NEXT J: NEXT I
30 LET HS=0
40 LET C=0: LET CL=0: LET SC=0
45 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS
50 PRINT "#####"
#####"
60 PRINT "###"; INK 2;"#"; INK 4;"#"; INK 7;"#";
###"
REM Entre as linhas 65 e 100
user o grafico 0
70 PRINT "###"; INK 2;"#"; INK
7;"#####"

```



```

0330 IF @#="6" THEN LET @#="8":
LET @#="#": GO TO 480: REM 07071
CO C
0340 IF @#="7" THEN LET @#="5":
LET @#="#": GO TO 480: REM 07071
CO E
0350 IF @#="8" THEN LET @#="7":
LET @#="#": REM 070710000
0401 IF (X>8 AND X<13) OR (U>13
AND U<18) THEN PRINT AT X,U) INK
C1;"": GO SUB 0300: GO TO 410
0400 LET @=INT (X+(@#="6")-(@#="
7"))+(U+(@#="8")-(@#="5"))
0400 PRINT AT X,U) INK C1;"": RE
M 070710000
0405 IF @=100 THEN GO SUB 000
0407 IF @=84+@1 THEN LET @=@+1
0410 LET X=X+(@#="5")-(@#="7"):
LET U=U+(@#="8")-(@#="5"):
0420 PRINT AT X,U) INK @1##
0430 PRINT AT 10,17;@: @=@+10
1,000
0450 IF X=X1 AND U=U1 THEN GO TO
0500
0500 IF FLAG>0 THEN LET FLAG=FLAG
+1: REM 070710000
0510 IF @=0 THEN LET @=(X1+(@#="6")-(@#
="7"))+(U1+(@#="8")-(@#="5"))<<"#
" THEN GO TO 0300
0520 IF @#="5" THEN LET @#="7":
LET @#="#": GO TO 0300: REM 07071
CO B
0530 IF @#="6" THEN LET @#="8":
LET @#="#": GO TO 0300: REM 07071
CO E
0540 IF @#="7" THEN LET @#="5":
LET @#="#": GO TO 0300: REM 07071
CO C
0550 IF @#="8" THEN LET @#="6":
LET @#="#": GO TO 0300: REM 07071
CO D
0580 IF (X>8 AND X<13) OR (U>13
AND U<18) THEN PRINT AT X,U) "":
GO SUB 0300: GO TO 010
0601 IF (X1=8 AND @#="6") OR (U1
=14 AND @#="8") OR (X1=10 AND @#
="7") OR (U1=17 AND @#="5") THEN
PRINT AT X1,U1) INK C1;"": GO
SUB 700
0602 IF (X1>8 AND X1<13) OR (U1>
13 AND U1<18) THEN PRINT AT X1,U
1) INK C1;"": GO TO 010

```

```

0600 LET Z=INT (X1+(@#="6")-(@#
="7"))+(U1+(@#="8")-(@#="5"))
0600 PRINT AT X1,U1) INK ((Z1-84
)-(120 AND Z1-84>0)): FLAG=1: REM
0 Z1-84>0;"": REM 070710000
0600 LET Z1=Z
0610 LET X1=X1+(@#="6")-(@#="7")
LET U1=U1+(@#="8")-(@#="5")
0600 PRINT AT X1,U1) INK @1##
0600 IF X=X1 AND U=U1 THEN GO TO
0600
0600 GO TO 0300
0700 IF FLAG=0 THEN RETURN
0700 PRINT AT X1,U1) INK C1;"":
0700 LET @=0: REM 070710000: L
ET FLAG=FLAG+1/2
0700 LET X1=X1+1*(@#="8")-(@#="
5")
0700 LET U1=U1+1*(@#="7")-(@#="
5")
0700 RETURN
0800 FOR I=25 TO -10 STEP -1: BE
G I,0,1: NEXT I
0810 PRINT AT 14,1) PAPER 4; INK
@; FLAG=1;"Corregue nusa tecla
para jogar"
0820 IF @>0 THEN LET @=@-1
0800 IF INKEY#<>" " THEN GO TO 02
0
0830 IF INKEY#="" THEN GO TO 030
0840 GO TO 030
0850 LET @=0: LET C1=8-C1: LET
@=@-10: @=@+.01,00: @=@+.01
0800
0880 LET FLAG=10
0890 LET @=@+10
0900 LET @=INT (RND*20)+1: LET
@=INT (RND*30)+1: IF @=000000
REM @ (LIN,COL)<>0 THEN GO TO 070
0910 PRINT AT LIN,COL) INK @1,FL
AG=1;"":
0900 RETURN
0920 LET I#=INKEY#: IF U>13 AND
U<18 THEN GO TO 030
0930 IF U>20 THEN GO TO 030
0940 IF I#="5" AND U>2 THEN LET
U=U-1: LET FLAG=FLAG-1
0940 IF I#="8" AND U<0 THEN LET
U=U+1: LET FLAG=FLAG+1
0900 RETURN
0950 IF I#="5" AND U>23 THEN LET
U=U-2: LET FLAG=FLAG+1

```

```

0000 IF A#="0" AND X>20 THEN LET
0001 X=X+10: LET traj=traj-1
0002 RETURN
0003 IF X>10 THEN GO TO 0075
0004 IF A#="6" AND X>7 THEN LET
X=X+2: LET traj=traj+1
0005 IF A#="7" AND X>1 THEN LET
X=X-2: LET traj=traj-1
0006 RETURN
0007 IF A#="0" AND X>20 THEN LET
X=X+10: LET traj=traj-1
0008 IF A#="7" AND X>14 THEN LET
X=X-10: LET traj=traj+1
0009 RETURN
10000 DATA "a",10,50,104,50,10,4,
4,4
10010 DATA "b",24,50,106,50,24,15
30,205
10020 DATA "c",224,70,50,255,255,
50,70,204
10030 DATA "d",153,255,153,24,50,
100,90,204
10040 DATA "e",7,114,34,255,255,3
4,114,7
50000 IF flag>0 THEN LET flag=flag
0-1

```

ZOMBIES

Zombies, por David Perry, é um jogo em que você deve tentar atrair Zombies para os pântanos. Os Zombies são cegos, e movem-se sempre em direção a si.

O próprio programa contém as instruções necessárias. Use as teclas de cursos para se mover.

```

10 REM
11 REM
12 REM
13 REM
14 REM
15 REM
16 REM
17 REM
18 REM
19 REM
20 REM
21 REM
22 REM
23 REM
24 REM
25 REM
26 REM
27 REM
28 REM
29 REM
30 REM
31 REM
32 REM
33 REM
34 REM
35 REM
36 REM
37 REM
38 REM
39 REM
40 REM
41 REM
42 REM
43 REM
44 REM
45 REM
46 REM
47 REM
48 REM
49 REM
50 REM
51 REM
52 REM
53 REM
54 REM
55 REM
56 REM
57 REM
58 REM
59 REM
60 REM
61 REM
62 REM
63 REM
64 REM
65 REM
66 REM
67 REM
68 REM
69 REM
70 REM
71 REM
72 REM
73 REM
74 REM
75 REM
76 REM
77 REM
78 REM
79 REM
80 REM
81 REM
82 REM
83 REM
84 REM
85 REM
86 REM
87 REM
88 REM
89 REM
90 REM
91 REM
92 REM
93 REM
94 REM
95 REM
96 REM
97 REM
98 REM
99 REM
100 REM
101 REM
102 REM
103 REM
104 REM
105 REM
106 REM
107 REM
108 REM
109 REM
110 REM
111 REM
112 REM
113 REM
114 REM
115 REM
116 REM
117 REM
118 REM
119 REM
120 REM
121 REM
122 REM
123 REM
124 REM
125 REM
126 REM
127 REM
128 REM
129 REM
130 REM
131 REM
132 REM
133 REM
134 REM
135 REM
136 REM
137 REM
138 REM
139 REM
140 REM
141 REM
142 REM
143 REM
144 REM
145 REM
146 REM
147 REM
148 REM
149 REM
150 REM
151 REM
152 REM
153 REM
154 REM
155 REM
156 REM
157 REM
158 REM
159 REM
160 REM
161 REM
162 REM
163 REM
164 REM
165 REM
166 REM
167 REM
168 REM
169 REM
170 REM
171 REM
172 REM
173 REM
174 REM
175 REM
176 REM
177 REM
178 REM
179 REM
180 REM
181 REM
182 REM
183 REM
184 REM
185 REM
186 REM
187 REM
188 REM
189 REM
190 REM
191 REM
192 REM
193 REM
194 REM
195 REM
196 REM
197 REM
198 REM
199 REM
200 REM
201 REM
202 REM
203 REM
204 REM
205 REM
206 REM
207 REM
208 REM
209 REM
210 REM
211 REM
212 REM
213 REM
214 REM
215 REM
216 REM
217 REM
218 REM
219 REM
220 REM
221 REM
222 REM
223 REM
224 REM
225 REM
226 REM
227 REM
228 REM
229 REM
230 REM
231 REM
232 REM
233 REM
234 REM
235 REM
236 REM
237 REM
238 REM
239 REM
240 REM
241 REM
242 REM
243 REM
244 REM
245 REM
246 REM
247 REM
248 REM
249 REM
250 REM
251 REM
252 REM
253 REM
254 REM
255 REM
256 REM
257 REM
258 REM
259 REM
260 REM
261 REM
262 REM
263 REM
264 REM
265 REM
266 REM
267 REM
268 REM
269 REM
270 REM
271 REM
272 REM
273 REM
274 REM
275 REM
276 REM
277 REM
278 REM
279 REM
280 REM
281 REM
282 REM
283 REM
284 REM
285 REM
286 REM
287 REM
288 REM
289 REM
290 REM
291 REM
292 REM
293 REM
294 REM
295 REM
296 REM
297 REM
298 REM
299 REM
300 REM
301 REM
302 REM
303 REM
304 REM
305 REM
306 REM
307 REM
308 REM
309 REM
310 REM
311 REM
312 REM
313 REM
314 REM
315 REM
316 REM
317 REM
318 REM
319 REM
320 REM
321 REM
322 REM
323 REM
324 REM
325 REM
326 REM
327 REM
328 REM
329 REM
330 REM
331 REM
332 REM
333 REM
334 REM
335 REM
336 REM
337 REM
338 REM
339 REM
340 REM
341 REM
342 REM
343 REM
344 REM
345 REM
346 REM
347 REM
348 REM
349 REM
350 REM
351 REM
352 REM
353 REM
354 REM
355 REM
356 REM
357 REM
358 REM
359 REM
360 REM
361 REM
362 REM
363 REM
364 REM
365 REM
366 REM
367 REM
368 REM
369 REM
370 REM
371 REM
372 REM
373 REM
374 REM
375 REM
376 REM
377 REM
378 REM
379 REM
380 REM
381 REM
382 REM
383 REM
384 REM
385 REM
386 REM
387 REM
388 REM
389 REM
390 REM
391 REM
392 REM
393 REM
394 REM
395 REM
396 REM
397 REM
398 REM
399 REM
400 REM
401 REM
402 REM
403 REM
404 REM
405 REM
406 REM
407 REM
408 REM
409 REM
410 REM
411 REM
412 REM
413 REM
414 REM
415 REM
416 REM
417 REM
418 REM
419 REM
420 REM
421 REM
422 REM
423 REM
424 REM
425 REM
426 REM
427 REM
428 REM
429 REM
430 REM
431 REM
432 REM
433 REM
434 REM
435 REM
436 REM
437 REM
438 REM
439 REM
440 REM
441 REM
442 REM
443 REM
444 REM
445 REM
446 REM
447 REM
448 REM
449 REM
450 REM
451 REM
452 REM
453 REM
454 REM
455 REM
456 REM
457 REM
458 REM
459 REM
460 REM
461 REM
462 REM
463 REM
464 REM
465 REM
466 REM
467 REM
468 REM
469 REM
470 REM
471 REM
472 REM
473 REM
474 REM
475 REM
476 REM
477 REM
478 REM
479 REM
480 REM
481 REM
482 REM
483 REM
484 REM
485 REM
486 REM
487 REM
488 REM
489 REM
490 REM
491 REM
492 REM
493 REM
494 REM
495 REM
496 REM
497 REM
498 REM
499 REM
500 REM
501 REM
502 REM
503 REM
504 REM
505 REM
506 REM
507 REM
508 REM
509 REM
510 REM
511 REM
512 REM
513 REM
514 REM
515 REM
516 REM
517 REM
518 REM
519 REM
520 REM
521 REM
522 REM
523 REM
524 REM
525 REM
526 REM
527 REM
528 REM
529 REM
530 REM
531 REM
532 REM
533 REM
534 REM
535 REM
536 REM
537 REM
538 REM
539 REM
540 REM
541 REM
542 REM
543 REM
544 REM
545 REM
546 REM
547 REM
548 REM
549 REM
550 REM
551 REM
552 REM
553 REM
554 REM
555 REM
556 REM
557 REM
558 REM
559 REM
560 REM
561 REM
562 REM
563 REM
564 REM
565 REM
566 REM
567 REM
568 REM
569 REM
570 REM
571 REM
572 REM
573 REM
574 REM
575 REM
576 REM
577 REM
578 REM
579 REM
580 REM
581 REM
582 REM
583 REM
584 REM
585 REM
586 REM
587 REM
588 REM
589 REM
590 REM
591 REM
592 REM
593 REM
594 REM
595 REM
596 REM
597 REM
598 REM
599 REM
600 REM
601 REM
602 REM
603 REM
604 REM
605 REM
606 REM
607 REM
608 REM
609 REM
610 REM
611 REM
612 REM
613 REM
614 REM
615 REM
616 REM
617 REM
618 REM
619 REM
620 REM
621 REM
622 REM
623 REM
624 REM
625 REM
626 REM
627 REM
628 REM
629 REM
630 REM
631 REM
632 REM
633 REM
634 REM
635 REM
636 REM
637 REM
638 REM
639 REM
640 REM
641 REM
642 REM
643 REM
644 REM
645 REM
646 REM
647 REM
648 REM
649 REM
650 REM
651 REM
652 REM
653 REM
654 REM
655 REM
656 REM
657 REM
658 REM
659 REM
660 REM
661 REM
662 REM
663 REM
664 REM
665 REM
666 REM
667 REM
668 REM
669 REM
670 REM
671 REM
672 REM
673 REM
674 REM
675 REM
676 REM
677 REM
678 REM
679 REM
680 REM
681 REM
682 REM
683 REM
684 REM
685 REM
686 REM
687 REM
688 REM
689 REM
690 REM
691 REM
692 REM
693 REM
694 REM
695 REM
696 REM
697 REM
698 REM
699 REM
700 REM
701 REM
702 REM
703 REM
704 REM
705 REM
706 REM
707 REM
708 REM
709 REM
710 REM
711 REM
712 REM
713 REM
714 REM
715 REM
716 REM
717 REM
718 REM
719 REM
720 REM
721 REM
722 REM
723 REM
724 REM
725 REM
726 REM
727 REM
728 REM
729 REM
730 REM
731 REM
732 REM
733 REM
734 REM
735 REM
736 REM
737 REM
738 REM
739 REM
740 REM
741 REM
742 REM
743 REM
744 REM
745 REM
746 REM
747 REM
748 REM
749 REM
750 REM
751 REM
752 REM
753 REM
754 REM
755 REM
756 REM
757 REM
758 REM
759 REM
760 REM
761 REM
762 REM
763 REM
764 REM
765 REM
766 REM
767 REM
768 REM
769 REM
770 REM
771 REM
772 REM
773 REM
774 REM
775 REM
776 REM
777 REM
778 REM
779 REM
780 REM
781 REM
782 REM
783 REM
784 REM
785 REM
786 REM
787 REM
788 REM
789 REM
790 REM
791 REM
792 REM
793 REM
794 REM
795 REM
796 REM
797 REM
798 REM
799 REM
800 REM
801 REM
802 REM
803 REM
804 REM
805 REM
806 REM
807 REM
808 REM
809 REM
810 REM
811 REM
812 REM
813 REM
814 REM
815 REM
816 REM
817 REM
818 REM
819 REM
820 REM
821 REM
822 REM
823 REM
824 REM
825 REM
826 REM
827 REM
828 REM
829 REM
830 REM
831 REM
832 REM
833 REM
834 REM
835 REM
836 REM
837 REM
838 REM
839 REM
840 REM
841 REM
842 REM
843 REM
844 REM
845 REM
846 REM
847 REM
848 REM
849 REM
850 REM
851 REM
852 REM
853 REM
854 REM
855 REM
856 REM
857 REM
858 REM
859 REM
860 REM
861 REM
862 REM
863 REM
864 REM
865 REM
866 REM
867 REM
868 REM
869 REM
870 REM
871 REM
872 REM
873 REM
874 REM
875 REM
876 REM
877 REM
878 REM
879 REM
880 REM
881 REM
882 REM
883 REM
884 REM
885 REM
886 REM
887 REM
888 REM
889 REM
890 REM
891 REM
892 REM
893 REM
894 REM
895 REM
896 REM
897 REM
898 REM
899 REM
900 REM
901 REM
902 REM
903 REM
904 REM
905 REM
906 REM
907 REM
908 REM
909 REM
910 REM
911 REM
912 REM
913 REM
914 REM
915 REM
916 REM
917 REM
918 REM
919 REM
920 REM
921 REM
922 REM
923 REM
924 REM
925 REM
926 REM
927 REM
928 REM
929 REM
930 REM
931 REM
932 REM
933 REM
934 REM
935 REM
936 REM
937 REM
938 REM
939 REM
940 REM
941 REM
942 REM
943 REM
944 REM
945 REM
946 REM
947 REM
948 REM
949 REM
950 REM
951 REM
952 REM
953 REM
954 REM
955 REM
956 REM
957 REM
958 REM
959 REM
960 REM
961 REM
962 REM
963 REM
964 REM
965 REM
966 REM
967 REM
968 REM
969 REM
970 REM
971 REM
972 REM
973 REM
974 REM
975 REM
976 REM
977 REM
978 REM
979 REM
980 REM
981 REM
982 REM
983 REM
984 REM
985 REM
986 REM
987 REM
988 REM
989 REM
990 REM
991 REM
992 REM
993 REM
994 REM
995 REM
996 REM
997 REM
998 REM
999 REM
1000 REM

```

```

10 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: CLS
12 LET LEU=5: GO SUB 55
14 LET GO=0
15 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: CLS: DIM A(LEU): DIM B
(LEU): LET KILL=0
16 PRINT AT 0,3:
17 FOR N=1 TO 20: PRINT AT N,3
: " ", INK 4:
: " ", INK 0: " ": NEXT N
18 PRINT AT 21,3:
19
20 BRIGHT 0: BORDER 7
20 PRINT AT 0,20: "7": AT 3,27: "
5": INK 0: "X": INK 0: "0": AT 4,20
: "0": AT 0,00: "Hovei": AT 1,20: "11
11": AT 0,20: "Creve": AT 7,20: "11
11":
21 PRINT AT 0,26: "Voce=": INK
0: "4": AT 0,20: INK 0: "Zomb=": IN
X 0: "X": AT 10,20: INK 0: "Dest.":
INK 0: "B": REM graficos B, B e
0
22 PRINT AT 12,27: "SOR": AT 13,
26: "SORTE": AT 14,26: "
23 PRINT AT 16,20: "": LEU-KILL:
AT 17,27: "PAPA": AT 18,26: "MATAA":
24 FOR N=1 TO LEU
25 LET A(N)=INT (RAND*20): LET
B(N)=INT (RAND*20): IF ATTR (A(N)
+1,B(N)+4) (<>)124 THEN GO TO 25
26 PRINT AT A(N)+1,B(N)+4: INK
0: PAPER 4: "0": NEXT N: REM gra
fico B
27 FOR N=1 TO LEU*5
28 LET A=INT (RAND*20): LET B=I
NT (RAND*20): IF ATTR (A+1,B+4) (<>)
124 THEN GO TO 28
29 PRINT AT A+1,B+4: INK 0: PA
PER 4: "0": NEXT A: REM grafico
0
30 LET X=INT (RAND*20): LET Y=I
NT (RAND*20): IF ATTR (X+1,Y+4) (<>)
124 THEN GO TO 30
31 PRINT AT X+1,Y+4: INK 0: PA
PER 4: "1": FOR N=0 TO 50 STEP 5:
BEEP .02,N: NEXT N: REM grafico
0
32 PRINT AT X+1,Y+4: INK 0: PA
PER 4: "1": REM grafico B

```



```

33 FOR N=1 TO LEV: IF A(N)=100
AND N=LEV THEN GO TO 49
34 IF A(N)=100 THEN NEXT N
35 IF KILL=LEV THEN GO TO 51
36 IF N>LEV THEN LET N=LEV
37 PRINT AT A(N)+1,B(N)+4; BRT
HT 1; INK 4; " "
38 IF A(N)<X THEN LET A(N)=A(N
)+1
39 IF B(N)>Y THEN LET B(N)=B(N
)-1
40 IF A(N)>X THEN LET A(N)=A(N
)-1
41 IF B(N)<Y THEN LET B(N)=B(N
)+1
42 LET A=ATTR (A(N)+1,B(N)+4)
43 IF A=124 THEN PRINT AT A(N
)+1,B(N)+4; INK 0; PAPER 4; BRIGHT
1; "X"; BEEP .01,N; NEXT N; REM
grafico 0
44 IF A=00 THEN LET 50=50+10;
FOR I=7 TO 0 STEP -1: PRINT AT A
(N)+1,B(N)+4; BRIGHT 1; PAPER 4;
INK I; "0"; BEEP .02,I*7; NEXT I
: PRINT AT A(N)+1,B(N)+4; INK 0;
PAPER 4; BRIGHT 1; "0"; LET KILL
=KILL+1; PRINT AT 10,20; "LEV=X
ALL"; " "; LET A(N)=100; IF N<=LE
V THEN NEXT N; REM grafico 0
45 IF N>=LEV THEN LET N=LEV
46 IF N<=LEV AND A(N)=X AND B(
N)=Y THEN FOR I=0 TO 7: PRINT AT
A(N)+1,B(N)+4; BRIGHT 1; PAPER
4; INK I; FLASH 1; "I"; BEEP .1,I
*7; NEXT I; GO TO 07; REM grafico
0
47 IF KILL=LEV THEN GO TO 51
48 IF N<=LEV THEN NEXT N
49 IF LEV=KILL THEN PAUSE 30;
GO TO 51
50 FOR G=1 TO 2: PRINT AT X+1,
Y+4; INK 4; BRIGHT 1; " "
51 LET X=X+(INKEY#="0")-(INKEY
#="7"); LET Y=Y+(INKEY#="0")-(IN
KEY#="7")
52 LET A=ATTR (X+1,Y+4)
53 IF A=124 THEN PRINT AT X+1,
Y+4; INK 0; PAPER 4; BRIGHT 1; "4
"; BEEP .04,20; NEXT G; GO TO 32
: REM grafico 0
54 IF A=00 THEN GO TO 103
55 LET LEV=0; CLS : PRINT AT 1

```

```

,12;"ZOMBIES!";AT 2,12;"=====
56 PRINT AT 3,0;"VOCE ESTA A
DE OLHAR LENTAMENTE"
57 FOR I=0 TO 7: PRINT AT I+6,
0; PAPER 7-I; INK I;"NO PANTANO!
"; NEXT I
58 PRINT AT 16,0;"MAS CONSEGUI
U MATAR-"; PRINT AT 19,10;50;" P
ONTOS!"
59 PRINT #1;TAB 2;"© Copyright
DAVID PERRY 1993"
60 LET I=0
61 PRINT AT 21,0; INK I;"CORRE
GUA NUMA TELA."
62 LET I=I+1; IF I>7 THEN LET
I=1
63 IF INKEY#="" THEN GO TO 51
64 LET LEV=0; PAUSE 0
65 GO TO 105
66 CLS : PRINT AT 0,12;"ZOMBIE
51";AT 1,12;"=====
67 PRINT : PRINT TAB 0;"VOCE A
COMO DE ATERRAR NUMA"
68 PRINT TAB 0;"ILHA POUQUA P
OR"
69 RESTORE 70; FOR N=0 TO 31:
READ A; PORE USA "A"+N,A; NEXT N
70 DATA 0,24,100,24,24,24,04,0
,100,100,100,100,100,24,00,24,12
0,100,00,00,00,100,100,100,00
,00,201,201,00,00,100
71 FOR N=7 TO 13: PRINT AT N,1
2; INK N-7;"ZOMBIES!"; BEEP .5,N
: NEXT N
72 PRINT AT 14,12;"ZOMBIES": 0
BEEP .5,00
73 PRINT AT 17,2;"© COPYRIGHT
DAVID PERRY 1993"
74 PAUSE 10; FOR N=0 TO 27: PR
INT AT 10,N; INK 2;"0"; INK 0;
"1"; BEEP .02,N; NEXT N; REM gra
ficos 0 e 0
75 PRINT AT 10,N;" "; PAUSE
10; BEEP .9,40; PAUSE 5; FOR N=
27 TO 0 STEP -1: PRINT AT 10,N;
INK 2;"0"; INK 0;"1"; BEEP .02
,N; NEXT N
76 CLS
77 PRINT AT 0,12;"ZOMBIES!";AT
1,12;"=====
78 PRINT AT 2,1;"Os Zombies se
o comedores de carne... Feliz

```

```

1000: 00 GO TO 10
1001: 00 VOICE VENTOSAS, PONTUAÇÃO DA
1002: 00 TENTE O PONTUAÇÃO DA TENTE
1003: 00 VOCE TENTE TENTE TENTE
1004: 00 COMO VOCE TENTE TENTE
1005: 70 PRINT AT 10,10;"ACABOTELE-SE"
1006: 00 PRINT AT 14,4;"Teclas de #0
1007: 00 VISUALIZ: 9-0"
1008: 01 PRINT AT 10,3;"VOCE="; INK
1009: 01 "+"; INK 0;" ZOMBIE="; INK 0;"
1010: 01 "PANTANO="; INK 0;"B"
1011: 01 INK 0;"REN GRAFICOS A, B e
1012: 0
1013: 02 LET I=0: PRINT #1,TAB 2;"©
1014: 02 Copyright DAVID PERRY 1983"
1015: 03 PRINT AT 10,9; INK I; PAPER
1016: 03 BRIGHT 1;" CARREGUE NUMA TEC
1017: 03 LA."
1018: 04 LET I=I+1: BEEP .01,I*7: IF
1019: 04 I>7 THEN LET I=0
1020: 05 IF INKEY#="" THEN GO TO 03
1021: 06 CLS : RETURN
1022: 07 CLS : PRINT AT 1,10;"ZOMBIE
1023: 07 ";AT 2,10;"=====
1024: 08 FOR N=0 TO 7: PRINT AT 4+N,
1025: 08 0; PAPER N; INK 7-N;"ELAS APANHAA
1026: 08 RAM-NO!!!"; NEXT N
1027: 09 GO TO 00
1028: 09 STOP
1029: 01 CLS : PRINT AT 1,12;"ZOMBIE
1030: 01 ";AT 2,10;"=====
1031: 02 FOR N=7 TO 0 STEP -1: PRINT
1032: 02 AT 4+N,7; PAPER N; INK 7-N;" P
1033: 02 A B E N 0 "; NEXT N
1034: 03 LET SC=0+(LEV*20)
1035: 04 PRINT AT 13,1;"VOCE PONTUOU
1036: 04 ";SC;" ATE "ACORDA!"
1037: 05 LET LEV=LEV+9: FOR N=1 TO 1
1038: 05 0 STEP 2: BEEP .02,N: NEXT N: PR
1039: 05 INT AT 16,4;"TENTE O NIVEL SEGUI
1040: 05 NTE"; PRINT : PRINT : GO
1041: 05 TO LEV;" ZOMBIES!"
1042: 06 PRINT AT 10,7;"BONUS = ";(L
1043: 06 EV-9)*20;" "
1044: 07 PRINT #1,TAB 2;"© Copyright
1045: 07 DAVID PERRY 1983"
1046: 08 LET I=0
1047: 09 PRINT AT 21,9; INK I;" CARR
1048: 09 EGUE NUMA TECLA."
1049: 100 LET I=I+1: BEEP .02,I*5: IF
1050: 100 I>7 THEN LET I=0
1051: 101 IF INKEY#="" THEN GO TO 00

```

```

102 GO TO 10
103 CLS : PRINT AT 1,12;"ZOMBIE
104 ";AT 2,10;"=====
105 104 FOR N=7 TO 0 STEP -1: PRINT
106 AT 4+N,0; PAPER N; INK 7-N;"UOC
107 E ACABOU DE SE SUICIDAR": NEXT N
108 GO TO 00
109 BORDER 0: PAPER 0: INK 7: 0
110 RIGHT 1: CLS
111 IF SC<=N(10) THEN GO TO 123
112 LET NUM=11: IF SC>=N(NUM) T
113 HEN INPUT "ESCREVA 0 INICIAIS ";
114 P#: IF LEN P#>3 THEN GO TO 100
115 PRINT AT 7,0; FLASH 1; BRIGHT
116 0; INK 7; PAPER 0;"ESPERE ALG
117 UNS SEGUNDOS!"
118 IF SC>=N(NUM) THEN LET N(NUM
119 )=SC: LET N#(NUM)=P#
120 FOR A=1 TO (NUM-1): LET B#
121 =N#(A): LET C#=N#(A+1): LET B=N(A
122 ): LET C=N(A+1): IF B<C THEN LET
123 N(A)=C: LET N(A+1)=B: LET N#(A)
124 =C#: LET N#(A+1)=B#
125 NEXT A: FOR N=1 TO NUM-1: I
126 F N(N)<N(N+1) THEN GO TO 111
127 NEXT N
128 CLS
129 PRINT AT 2,6;"TABELA DE PON
130 TUACOES"
131 PRINT AT 3,6;"=====
132 ====="
133 FOR N=1 TO NUM-1: PRINT AT
134 N+5,7; INK 0;"("; INK 2;N; INK 0
135 ;")";AT N+5,12; INK 7;N(N): PRIN
136 T AT N+5,17; INK 0;N#(N): NEXT N
137 PLOT 0,0: DRAW 255,0: DRAW
138 0,175: DRAW -255,0: DRAW 0,-175:
139 PRINT AT 17,5; PAPER 2; INK 7;"
140 CARREGUE NUMA TECLA." : LET I=0
141 LET I=I+1: IF I>7 THEN LET
142 I=0
143 PRINT AT 2,6; INK I;"TABELA
144 DE PONTUACOES"
145 BEEP .01,I*7: PAUSE 2: IF I
146 NKEY#="" THEN GO TO 110
147 BORDER 7: INK 0: BORDER 7: 0
148 RIGHT 0: CLS : GO TO 14
149 PRINT AT 10,4;"PONTUACAO MU
150 ITO BAIXA!!!"
151 PRINT AT 13,5;" CARREGUE NU
152 MA TECLA."
153 PAUSE 0: GO TO 114

```

AVENTURAS

O GIGANTE E A CIDADE

Estes dois programas de aventuras, escritos por Neil Pellinacci, são jogados de forma semelhante, se bem que o cenário e a solução sejam — evidentemente — diferentes. Vale a pena tentar convencer um amigo a introduzir os programas na máquina a fim de você não conhecer as soluções antes de jogar.

A CIDADE inclui instruções no programa que precede o jogo. Para terminar a aventura, você deve dizer a palavra de passe ao seu contacto. Você não sabe quem é o seu contacto, se bem que não haja muitas possibilidades.

Na primeira aventura, O GIGANTE, você deve tentar fugir do côvil do Gigante. Este é bastante mau, e só o deixará fugir se estiver morto. Você tem portanto de matá-lo para ganhar.

Em ambos os programas, tal como acontece de resto na maior parte dos jogos de aventuras, você dá as suas ordens em frases de duas palavras (verbos e substantivos): por exemplo LARGAR MACHADO. Depressa descobrirá quais as palavras que o programa reconhece.

Note que cada programa possui um carregador (que no caso de A CIDADE serve também como manual de instruções). Os possuidores de máquinas de 48 K podem desprezar a instrução CLEAR 32767, mas devem manter a CLS. Se você utilizar as instruções de SAVE incorporadas no programa, note que deve definir as cores e realizar os POKE's antes de gravar o programa.

Os carregadores foram usados para o espaço de memória livre para o programa principal. Para introduzir os programas, introduza primeiro, e grave, o carregador, e em seguida introduza e grave o programa principal, imediatamente a seguir. Ao gravar ambos os programas, ou ao executá-los, não se esqueça de que a linha de auto-execução é a 1 no caso dos carregadores e 900 no dos programas principais.

Vejamos agora o carregador de O GIGANTE:

```
10 REM AVENTURA © NEIL
    PELLINACCI 1983
20 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS
30 CLEAR 32767
40 POKE 23688,100: POKE 23688,
0: POKE 23684,71
50 LOAD ""
```

E o programa principal:

```
12 PRINT "ESTOU JUNTO A PORTA
A DO GIGANTE."
13 IF 0(9) THEN PRINT "NÃO TENHO
ARMAS E DEVO FUGIR DE PRESSA."
14 LET S=P: RETURN
18 PRINT "ESTOU NUMA PEQUENA
LOJA, QUE VENDE COMIDA PARA
CABALOS." : LET S=P: RETURN
33 PRINT "ESTOU NUMA SAÍDA EM
CURA. VEJO UMA PORTA PARA NORTE
E, E UMA CLAREIRA PARA ESTE."
34 LET N=P: LET S=P: LET E=P:
RETURN
35 PRINT "ESTOU NUMA CLAREIRA
A SAÍDA EN-CONTRA-SE A OESTE."
: LET S=P: RETURN
39 PRINT "ESTOU NUM CAMINHO
QUE SE DIRIGE PARA SUL. VEJO UMA
LOJA A NORTE, E UMA PONTE A ESTE."
40 LET N=P: LET S=P: LET E=P:
LET U=P: RETURN
42 PRINT "ESTOU JUNTO A UMA
PONTE, EXISTE UMA ESTRADA PARA
SUL." : LET S=P: LET U=P: RETURN
54 PRINT "ESTOU NUMA PASSAGEM
ESTREITA QUE SOBE LIGEIRAMENTE
PARA NORTE. VEJO LAZUL EM CIMA
DA LUZ DO SOL." : LET N=P: LET U=P:
RETURN
58 PRINT "ESTOU NUM CAMINHO
EM QUE SÓ SE VEJO UMA ESTRADA
PARA ESTE." : LET N=P: LET E=P
61 IF 0(4)=LOC THEN PRINT "VEJO
O UM TUMULO A MEUS PÉS."
62 RETURN
63 PRINT "EXISTE AQUI UMA
ESTRADA QUE CORRE NO SENTIDO
NORTE-SUL. A NORTE VEJO UMA PONTE." : L
```

```

81 N=P: LET S=P: RETURN
82 GO PRINT "VEJO UMA PASSEADA CL
PARALELA COM UMA GRANDE ARVORE."
: LET S=P: LET E=P: RETURN
72 PRINT "ESTOU NUMA FLORESTA
EXISTEM CLAREIRAS A ESTE E
A SUL."
73 LET U=P: LET E=P: LET S=P:
RETURN
75 PRINT "ESTOU AGORA NUMA CL
AREIRA, VEJO UM POCO PROFUNDO NO
SOLO." : LET U=P: LET D=P: RETUR
N
84 PRINT "A ESTRADA TERMINA A
SUL, VEJO UM RIO PARA SUL, COM M
ONTANHAS EM FUNDO." : LET N=P: R
ETURN
87 PRINT "UM ENORME LOBO SALT
OU DAS ARVORES E MATOU-ME!" : G
O TO 2200
90 PRINT "ESTOU NUMA CLAREIRA
OICO RUIDOSA DESTA." : LET N=P:
LET E=P: LET U=P: RETURN
92 PRINT "A FLORESTA TERMINA
A SUL, ENCONTRO UM RIO,
TALVEZ SEJA POSSIVEL ATRAVESSA-
LO." : LET N=P: LET U=P: RETURN
114 PRINT "ESTOU NA MARGEM DO
RIO, VEJO UM PORTAO ABERTO, GUAR
DADO POR UM CAO, E UMA MANSAO A
AIS ADVANTE."
116 LET S=200: IF NOT S THEN PR
INT "O CAO PODE TRAZER PROBLEMAS
..."
118 RETURN
117 PRINT "ESTOU PERTO DE UMA
MANSAO, VEJO UMA PORTA A MINHA
FRONTE." : LET N=P: RETURN
130 PRINT "ESTOU DENTRO DA MAN
SAO, NAO GOS TARIA DE VIVER AQUI
ESTA TUDO SOLO E CHEIRO MAL."
LET U=P: RETURN
800 PRINT "VELO..." : LET FL=0:
FOR N=P TO 0: IF O(Z)=LOC THEN
GO TO 805
810 NEXT Z: IF NOT FL THEN PRIN
T "NAO"
820 RETURN
830 IF Z=4 AND LOC=20 THEN GO T
O 810
840 PRINT O$(Z): LET FL=1: GO T
O 810

```

```

850 IF A$(P)=" " THEN LET A#=#
(O TO ): GO TO 850
860 IF A$(LEN A#)=" " THEN LET
A#=#( TO LEN A#-1): GO TO 850
870 LET B#="": FOR Z=P TO LEN A
#: IF A$(Z)=" " THEN LET B#=#(Z
+1 TO ): LET A#=#( TO Z-1): RET
URN
880 NEXT Z: RETURN
890 GO SUB 900: GLS
900 LET N=0: LET S=0: LET E=0:
LET U=0: LET D=0: LET O=0
910 GO SUB LOC#0: PRINT "ARIO
0: " ("NORTE" AND N): ("SUL" AN
D S): ("ESTE" AND E): ("OESTE" AN
D O): GO SUB 900
920 INPUT INK 0: "QUE DEVO FAZER
3" : LINE A#: IF A#="" THEN GO T
O C
930 GO SUB 930: PRINT INK 0: A#:
: B#: PRINT
940 IF LOC=4 AND AND>.00 THEN P
RINT "O GIGANTE SALTOU SOBRE MIM
E SE MATOU-ME O GRANDE, HORRI."
: GO TO 2200
950 IF LOC=30 AND NOT DOE AND R
ND>.9 THEN PRINT "O CAO ATAQUOU-M
E E MATOU-ME." : GO TO 2200
960 IF A#="N" AND N THEN LET LO
C=LOC-1: GO TO C+1-D
970 IF A#="S" AND S THEN LET LO
C=LOC+1: GO TO C+1-D
980 IF A#="E" AND E THEN LET LO
C=LOC+0: GO TO C+1-D
990 IF A#="O" AND O THEN LET LO
C=LOC-1: GO TO C+1-D
1000 IF A#="0" OR A#="E" OR A#="
N" OR A#="S" THEN PRINT "NAO POG
O IR PARA O CHARRADO." : GO TO C
1010 IF A#="I" THEN GO SUB 2200:
GO TO C
1020 IF A#="OLHA" THEN GLS : GO
TO C+1-P
1070 IF A#="DESCER" THEN GO TO 3
800
1080 IF A#="SUBIR" THEN GO TO 31
80
1090 IF A#="APANHAR" OR A#="AGRA
RAR" THEN GO TO 2100
1100 IF A#="LARGAR" THEN GO TO 2
200
1110 IF A#="MATAR" THEN GO TO 23
00

```

```

1120 IF A#="EMPURRAR" THEN GO TO
2400
1130 IF A#="ATRAVESSAR" THEN GO
TO 2500
1140 IF A#="ABRIR" THEN GO TO 26
00
1150 IF A#="ALIMENTAR" THEN GO T
O 2700
1160 IF A#="COMPRAR" THEN GO TO
2800
1170 IF A#="GRAVAR" THEN SAVE "A
VENTURA" LINE 999: PRINT "VERIFI
CAR...": VERIFY "AVENTURA": GO T
O C
1180 IF A#="DESISTIR" THEN GO TO
2970
1190 IF A#="SOCORRO" THEN PRINT
"O GIGANTE NAO ME DEIXARA" SEM
NENHUM QUE EU SEJA BEM EDUCADO.:
.: GO TO C
1200 IF A#="OUVR" THEN GO TO 29
00
1200 PRINT "NAO COMPREENDO.": GO
TO C
2000 PRINT "TENHO O SEGUINTE...
": LET PL=0: FOR Z=1 TO 9: IF O(
Z)=0 THEN PRINT O$(Z): LET PL=P
2010 NEXT Z: IF NOT PL THEN PRIN
T "NAO."
2020 RETURN
2100 IF B#="" THEN INPUT INK S: (
A#): "COMO ??": LINE B#: GO TO 2
100
2105 IF O#4 THEN PRINT "NAO POS
SO TRINSPORTAR MAIS.": GO TO C
2110 FOR Z=P TO 9: IF O$(Z) ( TO
LEN B#)=B# AND O(Z)=LOC THEN GO
TO 2130
2120 NEXT Z: PRINT "NAO POSSO TR
ANR 1580.": GO TO C
2130 IF B#="0550 DE CRO" AND LOC
=0 THEN PRINT "1580 SERIA ROUBO!
": GO TO C
2140 PRINT "OK...": LET O(Z)=O:
LET O#=#+P: GO TO C
2200 IF B#="" THEN INPUT INK S: "
QUE DEVO EU "(A#):"??": LINE B#
.: GO TO 2200
2210 FOR Z=P TO 9: IF O$(Z) ( TO
LEN B#)=B# AND O(Z)=0 THEN PRINT
"OK...": LET O(Z)=LOC: LET O#=#
+P: GO TO C

```

```

2220 NEXT Z: PRINT "NAO TENHO."
B#:"!": GO TO C
2300 IF B#="" THEN INPUT INK S: "
MATAR QUEM?": LINE B#: GO TO 230
0
2310 IF B#<>"GIGANTE" THEN PRINT
"NAO SOU UM ASSASSINO!": GO TO
C
2315 IF LOC<>4 THEN PRINT "NAO V
ELO NENHUM GIGANTE!": GO TO C
2320 IF O(9) THEN PRINT "NAO TEN
HO NENHUMA ARMA PARA MA- TAR O G
IGANTE.": GO TO C
2330 PRINT "OPTIMO. MATAMOS O
GIGANTE E CON-SERVIAMOS A LIBERDA
DE.": GO TO 2330
2400 IF B#="" THEN INPUT INK S: "
QUE DEVO EMPURRAR?": LINE B#: GO
TO 2400
2410 IF B#<>"PORTA" THEN PRINT "
NAO POSSO FAZER ISSO.": GO TO C
2420 IF LOC<>39 THEN PRINT "QUE
PORTA?": GO TO C
2430 IF NOT DO THEN PRINT "ESTA
TECHADA A CHAVE.": GO TO C
2440 PRINT "OK...": LET LOC=LOC+
P: GO TO C-P-D
2500 IF B#="" THEN INPUT INK S: "
QUE QUER ATRAVESSAR?": LINE B#:
GO TO 2500
2505 IF B#<>"RIO" THEN PRINT "NA
O SOU ASSIM TAO ESPERTO!": GO TO
C
2507 IF LOC<>31 AND LOC<>32 THEN
PRINT "SE ME MOSTRAR O RIO, EU
TENHO...": GO TO C
2510 IF O(2) THEN PRINT "VOU DAR
CISAR DE QUALQUER COISA PARA ME
AJUDAR.": GO TO C
2520 PRINT "OK...": LET LOC=LOC+
(LOC=31)-(LOC=32): GO TO C-P-D
2600 IF B#="" THEN INPUT INK S: "
POR FAVOR CONTINUE A ORDEM.": LI
NE B#: GO TO 2600
2605 IF B#<>"PORTA" THEN PRINT "
TALVEZ VA' LEVAR ALGUM TEMPO...
": GO TO C
2610 IF LOC<>39 THEN PRINT "NAO
ESTOU A VER PORTA NENHUMA!": GO
TO C
2620 IF DO THEN PRINT "TARRA JA'
ESTA' ABERTA!": GO TO C

```

```

0005 IF C(4) THEN PRINT "NAO TEN
0006 CHAVE...": GO TO C
0007 PRINT "OK...": LET C(P)=0: GO
TO C
0008 IF B<>"000" THEN PRINT "NA
0009 ME PARAR O QUE VALIDA A PERA...":
GO TO C
0010 IF LOC<>0 THEN PRINT "QUAL
0011 CHAVE?": GO TO C
0012 IF C(P) THEN PRINT "COM O Q
0013 UO O CHAVE?": GO TO C
0014 PRINT "O CHAVE DOORMECEU...":
LET C(P)=0: LET C(P)=LOC
: LET C(P)=0: GO TO C
0015 IF C(P) THEN PRINT "NAO TEN
0016 HO CHAVE...": GO TO C
0017 IF B<>"000" THEN INPUT INK 5:"
0018 ALGUMA COISA E ESPECIAL?" LINE
0019 # : GO TO C
0020 IF B<>"000" THEN PRINT "MAU
0021 LA ESTOU A VERDER A MINHA PRO
0022 MOVAÇÃO?": GO TO C
0023 IF B<>"000 DE OAO" THEN P
0024 RAR "QUE SE E GRATIAS?": G
TO C
0025 IF LOC<>6 THEN PRINT "NAO H
0026 AUM NENHUM OASO...": GO TO C
0027 IF C(P) THEN PRINT "NAO POS
0028 SO TRANSPORTAR MAIS NADA...": GO T
O C
0029 PRINT "OK...": LET C(P)=0:
LET C(P)=0: GO TO C
0030 IF LOC<>20 THEN PRINT "ALND
0031 NAO POSSO PAPER ISSO...": GO TO
C
0032 IF C(4)<>LOC THEN PRINT "NA
0033 O HA NAO DOU...": GO TO C
0034 PRINT "DESCOBRI UMA CHAVE...":
GO TO C
0035 IF LOC<>25 THEN PRINT "PARA
0036 O QUE?": GO TO C
0037 IF C(5) THEN PRINT "VOU PAR
0038 TIR O PAPER...": GO TO C
0039 PRINT "OK...": LET LOC=LOC-
7: GO TO C
0040 IF LOC<>18 THEN PRINT "ALND
0041 NAO POSSO PAPER ISSO...": GO TO
C
0042 IF C(6) THEN PRINT "COMO ME
0043 TROUXE PARA AQUI? PODE- ME EXP
0044 LICAR?": GO TO C
0045 PRINT "OK...": LET LOC=LOC+

```

```

7: GO TO C
0046 CLS : INPUT "TEM A CERTeza
0047 (S/N)?": LINE #
0048 IF B<>"S" OR B<>"N" THEN
0049 GO TO 0050
0050 INPUT "QUEX OUTRA AVENTURA
0051 (S/N)?": LINE #
0052 IF B<>"S" OR B<>"N" THEN
0053 GO TO 0054
0054 IF B<>"N" OR B<>"NAO" THEN
0055 GO TO 0056
0056 LET C=P/PI: LET P=PI/PI
0057 LET F=VAL "7": LET LOC=VAL
"04": LET C=0
0058 LET C=0: LET C=0: LET C=
VAL "1000": DIM C$(9,11): DIM C(
0)
0059 FOR Z=P TO INT (PI*PI): ARR
0060 C$(Z),Y# : LET C(Z)=VAL Y#
0061 NEXT Z
0062 DATA "0000 DE OAO","0","TAB
0063 LA","11","ANEL","14","CHAVE","00
0064 0","0000","20","PAPER","20","SOLO
0065 40","00","LIVRO","30","PISTOLA",
0066 RETURN

```

Apresentamos agora o carregador de A CIDADE:

```

10 DEM A Cidade
20 DEM Um Jogo de Aventura
30 DEM De Neil Pellipacci 1983
40 DEM Spectrum 16 K
50 DEM
60 DEM
70 BORDER 0: PAPER 0: INK 7: B
RIGHT 1
80 CLEAR 02797: CLS
90 POKE 23000,100: POKE 23000,
0: POKE 23004,71
100 GO SUB 200
110 LOAD "
120 DIM A$(32): PAPER 0: INK 0:
LET B#="": PAPER 0: GRAVADOR
: LET L=0: GO SUB 999
130 PAUSE 100: PAPER 0: CLS
140 LET A#=" A CIDADE
E": LET L=L+1: INK 0: GO SUB 999
150 LET A#="UMA AVENTURA PARA O
NX SPECTRUM...": LET L=L+1: GO SUB

```

```

000
240 LET A#="ESCRITA POR NEIL PE
LINADOOH 1983": INK 5: LET L=0:
GO SUB 900
000 LET A#="          CARREGUE NUMA
TACLA.": INK 7: LET L=15:
GO SUB 900
000 LET L=1: PAUSE 1: PAUSE 0:
CLS
000 FOR Z=100 TO 30 STEP 5: GOTO
.01,Z: GOTO .01,Z/10: GOTO .01
,Z-10: GOTO .01,Z+10: NEXT Z
070 RESTORE
000 READ A#: IF A#(1)="*" THEN
RETURN
000 IF A#(1)="+" THEN PRINT AT
0,0: INK 0:          CARREGUE NUMA
TACLA.": PAUSE 1: PAUSE 0: CLS
: GOTO .1,0: GOTO .1,10: GOTO .1
,0: GOTO .3,0: LET L=L+2: GO TO
000
000 GO SUB 900: LET L=L+2: GO TO
000
400 DATA " NESTA AVENTURA VOCE
DEVE"
410 DATA "RECUPERAR DINHEIRO AO
USADO. ESTE"
420 DATA "DINHEIRO ESTA' POREM
GUARDADO NO"
430 DATA "BANCO. O BANCO PERTEN
CE AO"
440 DATA "GANGSTER QUE ROUBOU O
SEU"
450 DATA "DINHEIRO."
460 DATA " VOCE DEVE PENETRAR N
O BANCO"
470 DATA "E RECUPERAR O DINHEIRO
O."
480 DATA "MAS NAO SERA' FACIL O
CARGAR"
490 DATA "PORQUE PASSARA POR LA
ORAO."
500 DATA "+"
510 DATA " PARA CONTRARIAR ISTO
VOCE"
520 DATA "AMEACOU A CIDADE COM
UMA BOMBA."
530 DATA "E PORTANTO A CIDADE E
STA' VANIA."
540 DATA "DEVE NO ENTANTO EVITA
R QUE A"
550 DATA "POLICIA O CONFUNDA, O
AO QUE"

```

```

560 DATA "ESTA NADA SABE SOBRE
A SUA"
570 DATA "MISSAO SECRETA."
580 DATA " QUANDO TIVER RECUPER
ADO O"
590 DATA "DINHEIRO, DEVE PROCUR
AR O SEU"
600 DATA "CONTACTO, QUE O AJUDA
A A TUGER."
610 DATA "+"
620 DATA " PARA JOGAR, ESCREVA
AS ORDENS"
630 DATA "EM PORTUGUES, USANDO
VERBOS E"
640 DATA "SUBSTANTIVOS. USE 'N'
PARA N"
650 DATA "PARA NORTE, 'S' PARA
SUL, ETC."
660 DATA " SE QUISER PARRAR, ESC
REVA"
670 DATA "'DESISTIR' OU 'PARRAR'
: 'GRAVAR'"
680 DATA "GRAVA O JOGO EM CASSE
TE, PARA"
690 DATA "PODER TERMINA-LO MAIS
TARDE."
700 DATA "+"
710 DATA "" "" "" "" "" PONHA O
GRAVADOR EM MOVIMENTO."
720 DATA "*"
800 PAPER 0: INK 7: STOP
900 FOR Z=02 TO 1 STEP -1: PRIN
T AT L,0:A#(Z TO 1): NEXT Z: RETU
RN

```

Depois de ter introduzido e gravado em fita este carregador, proceda do mesmo modo com o programa principal listado em seguida:

```

9 PRINT "ESTOU NUMA ESTACAO
DE METRO. ES-TA' UM COMBOIO PRES
TES A PARTIR. NO EXTERIOR ESPERA
UM TAXI.": RETURN
12 PRINT "ESTOU JUNTO A UM TR
LHO. VEJO UM PREDIO EM CONSTRUCA
O A LESTE.": LET E=P: LET S=P: R
ETURN
15 PRINT "CAIU SOBRE MIM UMA
ENORME CHAPA DE ACO. MORRI.": GO
TO 9999
21 PRINT "ESTOU NO EXTERIOR O
O BANCO. A PORTA ENCONTRA-SE

```

```

A MINHA FRENTE. A BOLA CONTINUA
PARA LESTE.": LET S=P: LET Q=10
RETURN
34 PRINT "ESTOU NUMA BUA DESE
TRA, ONDE SE VELO UM TAXI. A SUL
FOI O BUA DE CABO.": LET S=P: LE
T S=P: LET Q=0: RETURN
37 PRINT "ESTOU NO INTERIOR D
E UMA GRANDE LOJA. NAO VELO NING
UM. EXISTE UM CARA PARA LESTE.
": LET S=P: LET Q=P: LET
T S=P: RETURN
39 PRINT "ESTOU NUM PASSADOUO C
OM UM PORTO. LAI ABANON TUA
DESTE.": LET S=P: RETURN
40 PRINT "ESTOU NO INTERIOR D
O BANCO. NAO OUO QUALQUER RUIDO
": IF GO THEN PRINT "VELO UM
GUARDA DE COSTAS PARA MIM.":
37 PRINT : LET Z=P: LET S=NOT
GO: RETURN
41 PRINT "ESTOU NUM MERCADO D
E BUA. A ESTAVO UMA ESQUADRA,
EM UM LOCAL DECONSTRUCO A SUL.
40 LET N=P: LET S=P: LET E=P
RETURN
42 PRINT "CHEGUEI A ESQUADRA
. NAO ESTA DEBATA.":
43 IF NOT Q(7) THEN PRINT "UM
POLICIA VE O DINHEIRO E DE-MORO
M DE PARAR.": GO TO 9999
44 PRINT "UM POLICIA TIRA-ME O
QUE BRILHA COLOCA TUDO NOS SE
TOS ONDE ESTAVA.": LET N=P: L
ET Q=P: GO TO 788
45 PRINT "ESTOU NUMA PASSADOUA
ESTADO DE MOTO. VELO UM BUA
CO NA PARADA A SUL.": LET S=P: R
ETURN
48 PRINT "ESTOU JUNTO DA PORT
A DO COFRE, QUE ESTA FECHADA.
49 IF VC THEN PRINT "EXISTE UM
BUBO WELD.":
50 LET S=VC: RETURN
51 PRINT "VELO OUS PORTAS, U
NA D COSTA, OUTRA A SUL. A ENTRA
DO DO COFRE ESTA A NORTE.": LE
T Z=P: LET S=P: LET Q=P: RETURN
54 PRINT "ESTOU NUM LOCAL DE
CONSTRUCO, PORADO DE MUROS AL
TOS.": LET N=P: RETURN
57 PRINT "ESTOU NO HALL DE UM

```

```

GRANDE HOTEL VELO A MINHA FRE
NTE A RECEP-CAO.": LET S=P: RET
URN
60 PRINT "ESTOU NA CAVE DE UM
GRANDE BUI-FICIO.": LET N=P: L
ET Q=P: RETURN
63 PRINT "ESTOU NO INTERIOR D
O COFRE. PARATINGO O DINHEIRO,
DEVO INDICAR CODIGO. EXISTEM 9
COTAS, E TENHO DE CARRGAR N
O COFRETO.":
64 LET Z=P: RETURN
65 PRINT "SUBIU UMA PARADA A
TRAS DE MIM. FUI APANHADO.": GO
TO 9999
70 PRINT "ESTOU NO INTERIOR D
A PRINCIPAL SALA DO COFRE.": LE
T N=P: RETURN
699 IF A#(P)=" " THEN LET A#=#
(Q TO 1): GO TO 699
619 IF A#(LEN A#)=" " THEN LET
A#=#( TO LEN A#-P): GO TO 619
620 LET B#="": FOR Z=P TO LEN A
#: IF A#(Z)=" " THEN LET B#=#(Z
+P TO ): LET A#=#( TO Z-P): RET
URN
630 NEXT Z: RETURN
700 RESTORE 9999: FOR Z=P TO 7:
READ A#,B#: IF Q(Z)=0 THEN LET
Q(Z)=VAL B#
710 NEXT Z: LET OB=0: RETURN
630 PRINT "SAIDAS: "; ("NORTE "
AND N); ("SUL " AND S); ("ESTE "
AND E); ("OESTE " AND O)
619 PRINT "VELO.":
620 LET FL=0: FOR Z=P TO 7: IF
Q(Z)=LOC THEN PRINT Q#(Z): LET F
L=0
630 NEXT Z: IF NOT FL THEN PRIN
T "NAO DE INTERESSANTE.":
640 RETURN
999 GO SUB 9999: CLS
1000 LET N=0: LET S=0: LET E=0:
LET V=0: GO SUB LOC*3: GO SUB 99
99
1000 INPUT INK 9: "QUE DEVO FAZER
?" LINE A#: IF A#=" " THEN GO TO
C
1010 IF LOC=12 AND AND>.9 AND GO
THEN PRINT "O GUARDA VIROU-SE,
VU-ME E DIS-PAROU SOBRE MIM. MO
ARI.": GO TO 9999

```



```

1050 GO SUB 800: PRINT " INK 5:"
#:"
1060 IF A#="N" AND N THEN LET LO
C=LOC+7: GO TO C+1
1070 IF A#="S" AND S THEN LET LO
C=LOC+7: GO TO C+1
1080 IF A#="M" AND M THEN LET LO
C=LOC+7: GO TO C+1
1090 IF A#="O" AND O THEN LET LO
C=LOC+7: GO TO C+1
1100 IF A#="I" OR A#="S" OR A#="
M" OR A#="N" THEN PRINT "NAO POS
SO IR A TRAZER O BILHETE...": GO TO C
1110 IF A#="PASSAGEM" OR A#="BARR
ARRA" THEN GO TO 2000
1120 IF A#="LARGAR" THEN GO TO 2
000
1130 IF A#="I" THEN GO SUB 2000:
GO TO C
1140 IF A#="OLHAR" THEN CLS: GO
TO C+1
1150 IF A#="SOCCORR" THEN GO TO
2000
1160 IF A#="ABOCDORR" OR A#="PERA
RQUIX" THEN GO TO 2400
1170 IF A#="ABRIR" THEN GO TO 20
00
1180 IF A#="EMPURRAR" THEN GO TO
2000
1190 IF A#="DESCER" THEN GO TO 2
700
1200 IF A#="DESISTIR" OR A#="PAR
AR" THEN GO TO 2000
1210 IF A#="LER" OR A#="EXAMINAR
" THEN GO TO 2000
1220 IF A#="MATAR" THEN GO TO 20
00
1230 IF A#="CORTAR" THEN GO TO 3
000
1240 IF A#="CARREGAR" OR A#="PRE
MIR" THEN GO TO 3100
1250 IF A#="GRABAR" THEN SAVE "C
MODOE" LINE 800: PRINT "VERIFICAR
A...": VERIFY "CIDADE": GO TO C
1260 IF A#="DIZER" THEN GO TO 32
00
1300 PRINT "NAO COMPREENDO...": GO
TO C
2000 IF B#="" THEN INPUT INK 5:"
ABRIR O BILHETE ??": LINE 8#: GO T
O 2000
2002 IF LEN B#>13 THEN PRINT "NA

```

```

O POSSO": GO TO C
2005 IF OB=3 THEN PRINT "NAO POS
SO TRANSPORTAR MAIS NADA...": GO T
O C
2010 FOR Z=P TO 7: IF O$(Z) ( TO
LEN B#)=B# AND O(Z)=LOC THEN LET
O(Z)=0: LET OB=OB+1: PRINT "OK.
...": GO TO C
2020 NEXT Z: PRINT "NAO VEJO "B
#;"!": GO TO C
2100 IF B#="" THEN PRINT A#;" O
BILHETE?": GO TO C
2110 IF LEN B#>13 THEN PRINT "NA
O POSSO...": GO TO C
2120 FOR Z=P TO 7: IF O$(Z) ( TO
LEN B#)=B# THEN PRINT "OK...": L
ET O(Z)=LOC: LET OB=OB-1: GO TO
C
2130 NEXT Z: PRINT "NAO TENHO "B
#;"!": GO TO C
2200 PRINT "TENHO O SEGUINTE..."
: LET FL=0: FOR Z=P TO 7: IF O(Z
)=0 THEN PRINT O$(Z): LET FL=1
2210 NEXT Z: IF NOT FL THEN PRIN
T "NADA."
2220 GO TO C
2300 IF LOC=17 THEN PRINT "PERGU
NTE A POPULACAO...": GO TO C
2310 PRINT "SIGA A ESTRADA NACIO
NAL NO. 2...": GO TO C
2400 IF B#="" THEN PRINT A#;" O
BILHETE?": GO TO C
2410 IF B#<>"COMBOIO" AND B#<>"T
AXI" THEN PRINT "NAO POSSO...": GO
TO C
2420 IF B#="COMBOIO" AND LOC<>3
AND LOC<>15 THEN GO TO 2400
2430 IF B#="TAXI" AND LOC<>3 AND
LOC<>8 THEN GO TO 2400
2440 IF B#="COMBOIO" THEN GO TO
2470
2450 IF O(2) THEN PRINT "NAO TEN
HO DINHEIRO SUFICIENTE...": GO TO
C
2460 PRINT "O TAXI PARTE COMIGO
DENTRO...": LET LOC=LOC+1 AND LOC
=3 - (1 AND LOC=8): GO TO C+1
2470 IF O(3) THEN PRINT "PRECISO
DE BILHETE!": GO TO C
2480 PRINT "ENTRO E O COMBOIO PA
RTE...": LET LOC=(3 AND LOC=15)+(1

```

```

9 AND LOC=3): GO TO C-P
0499 PRINT "AINDA NAO POSSO FAZER
X ISSO.": GO TO C
0509 IF B#="" THEN PRINT "QUE DE
VO ABRI-LO?": GO TO C
0519 IF B#<>"PORTA" THEN PRINT "
NAO POSSO FAZER ISSO.": GO TO C
0529 IF LOC<>7 THEN PRINT "QUAL
PORTA?": GO TO C
0539 IF DO THEN PRINT "ERRA! JA
ESTA ABERTA!": GO TO C
0549 IF O(5) THEN PRINT "PRECISO
DE CHAVE.": GO TO C
0559 PRINT "OK...": LET DO=P: GO
TO C
0569 IF B#="" THEN PRINT "EMPURR
AR O QUE?": GO TO C
0579 IF B#<>"PORTA" THEN PRINT "
NAO POSSO...": GO TO C
0589 IF LOC<>7 THEN PRINT "NAO O
UZO NENHUMA PORTA...": GO TO C
0599 IF NOT DO THEN PRINT "ESTA
ABERTA A CHAVE.": GO TO C
0609 LET DO1=P: PRINT "OK...":
VELO PARA BAIXO PARA BAIXO...":
GO TO C
0619 IF LOC<>7 THEN GO TO 0499
0629 IF NOT DO1 THEN PRINT "NA
UMA PORTA PARA ATRAVESSAR!": GO
TO C
0639 PRINT "OK...": LET LOC=LOC+
1: GO TO C-P
0649 IF B#="" THEN INPUT INK S:(
B#): "O QUE?": LINE B#: GO TO 0
0659 IF B#<>"PISTA" THEN PRINT "
NAO POSSO FAZER ISSO!": GO TO C
0669 IF O(6) THEN PRINT "NAO TEN
HO NENHUMA PISTA.": GO TO C
0679 PRINT "A PISTA ESTA ESCRIT
A NUMA FOLHA DE PAPEL A4. QUANTO
TEM CONDUIZ NAO SEBE.": GO TO C
0689 IF B#="" THEN PRINT "MATER
QUANTO?": GO TO C
0699 IF B#<>"GUARDA" THEN PRINT
"PRECISO ME POR BANCOS MORRIS.":
GO TO C
0709 IF LOC<>12 THEN PRINT "NAO
VEJO O GUARDA.": GO TO C
0719 IF O(P) THEN PRINT "NAO TEN
HO ARMA.": GO TO C
0729 PRINT "OK...": LET GU=0: GO

```

```

TO C
0739 IF B#="" THEN INPUT INK S:"
QUE DEVO CORTAR?": LINE B#: GO T
O 0709
0749 IF B#<>"PORTA DO COFRE" THE
N PRINT "NAO POSSO FAZER ISSO.":
GO TO C
0759 IF LOC<>16 THEN PRINT "NAO
VEJO A PORTA DO COFRE.": GO TO C
0769 IF O(4) THEN PRINT "NAO TEN
HO NADA COM QUE CORTAR.": GO TO
C
0779 PRINT "OK...": LET UC=P: GO
TO C-P
0789 IF LOC<>21 THEN PRINT "NAO
VEJO NADA PARA A#": GO TO C
0799 IF LEN B#<>P THEN PRINT "PR
ECISO DE UM NUMERO ENTRE 1 E 9":
GO TO C
0809 IF B#<"1" OR B#>"9" THEN PR
INT "ISSO NAO E UM NUMERO!": GO
TO C
0819 LET Z=VAL B#: IF Z<>3 THEN
LET LOC=LOC+P: GO TO C-P
0829 LET LOC=LOC+P: PRINT "O POR
TAO DE FERRO DESLIZA LENTA-MENTE
...": GO TO C-P
0839 IF B#="" THEN PRINT "DIZER
O QUE?": GO TO C
0849 IF B#<>"PALAVRA DE PASSE" T
HEN PRINT "NAO POSSO DIZER ISSO"
GO TO C
0859 IF LOC=15 AND NOT O(7) THEN
PRINT "ENCONTREI O CONTACTO,
E TERMINARIA MISSAO. O CONTACTO E
RA O CON-DUTOR DO COMBOIO.": GO
TO 0000
0869 IF LOC=3 OR LOC=8 THEN PRIN
T "O CONDUTOR DO TAXI DIZ ALGO B
AS-TANTE DESAGRADAVEL.": GO TO C
0879 PRINT "NINGUEM PARECE INTE
RESSAR-SE.": GO TO C
0889 LET O=PI-PI: LET P=NOT O
0899 LET P=VAL "5": LET LOC=VAL
"13"
0909 LET OB=0: LET DO=0: LET GU=
P: LET UC=0: LET C=VAL "1000"
0919 DIM O(7): DIM B$(7,17)
0929 RESTORE 0000: FOR Z=P TO 7:
READ O$(Z),A$: LET O(Z)=VAL A$:
NEXT Z
0939 DATA "FACA","4","BOLSA","9"

```

```

"BIlhETE", "10", "LANTERNA", "10",
"CHAUV", "10", "PISTA ESCALTA", "20",
"MAIA COM DINHEIRO", "20",
0050 PRINT TAB 11; INK 0; "A CH
DADE" ESCALTA POR NELL PELLI
NACCI" CARREGUE EM QUALQ
UER TECLA"
0070 PAUSE P: PAUSE 0: FOR Z=50
TO -20 STEP -5: BEEP .01,Z: BEEP
.01,Z-3: BEEP .01,Z+5: NEXT Z
0080 RETURN
0090 INPUT "TEM A CETERZA (S/N)?
"; LINE A#: IF A#="" THEN GO TO
0090
0010 IF A#(P)="S" THEN GO TO 000
0
0020 GO TO C
0030 INPUT "QUER NOVA AVENTURA (
S/N)? "; LINE A#: IF A#="" THEN
GO TO 0000
0010 IF A#(P)="N" THEN STOP
0020 RUN 000

```

PORTAS DO DESTINO

Trata-se de uma obra-prima: escrita por Malcolm Young, que recompensará plenamente o tempo que você gastará a introduzir o programa na máquina.

Esta aventura gráfica baseia-se na rotina de labirinto que ocupa as linhas 1000 a 1400. A rotina que se inicia na linha 2000 trata os gráficos.

Quando Malcolm começou a escrever este programa, pensava simplesmente construir um programa que apresentasse um labirinto a três dimensões. No entanto, quando o terminou resolveu que poderia torná-lo muito mais interessante, e o resultado está aqui para ser visto. São incluídas todas as instruções necessárias. Um aviso: o programa produz labirintos bastante difíceis de resolver. Não use este programa se quiser um jogo que possa dominar numa hora ou pouco mais.

```

2 REM © 1983, Malcolm Young
10 DEF FN r(x)=INT (RND*x)+1
15 LET set=0
30 GO SUB 9000: REM instrucoe
5
40 BORDER 4: PAPER 9: INK 0: C

```

```

L3
0000 GO SUB 1000: REM construir
labirinto
0010 GO SUB 0000: REM inicializa
cao
0020 GO TO 000
0030 NEXT a
0040 REM rotina principal
1000 LET K#=INKEY#: IF K#="" THE
N GO TO 1000
1100 LET f=f+(K#="d")-(K#="e")+2
*(K#="7")
1200 LET f=f+4*(f<1)-4*(f>4)
1300 IF K#="#" THEN GO SUB 0000
1400 IF K#="a" THEN LET ##(04,04
)="": LET ((11))=((11))-1((10)): I
f ((11))<0 THEN GO TO 0000
1400 LET px=0x+((K#="b")*((f=2)-
(f=4)))
1500 LET py=0y+((K#="a")*((f=3)-
(f=1)))
1600 IF ##(04,04)="#" THEN LET s
=#1: LET px=0x-1(f=0)+1(f=4): LET
py=0y-1(f=0)+1(f=1)
1700 IF ##(04,04)=CHR# 128 THEN
GO SUB 0000
1700 IF ##(04,04)> "." THEN GO TO
0000
1750 IF 01=04 AND 02=04 THEN GO
TO 0000
1800 IF K#="q" THEN LET re=1-re:
LET K#="e": GO TO 1400
1900 IF K#="!" THEN GO SUB 7000
2000 IF ((7) AND K#="x" THEN GO
SUB 7000
2100 IF ((8) AND K#="t" THEN GO
SUB 5000
2200 IF K#="s" THEN INPUT "quer
steve? o joao? "; LINE a#: IF a#
="s" THEN GO SUB 7000: CLS
2300 CLS
2400 PRINT ##: INVERSE 1: "NORTE"
AND f=1, "ESTE" AND f=2, "SUL" AN
D f=3, "OESTE" AND f=4: INVERSE 0
): FLASH 1: "REPTER" AND re:
FLASH 0: TAB 10: "000000": ((11))
2500 BEEP .1,10: GO SUB 0000
2600 IF INKEY#="" AND (re#1) THE
N LET K#="e": GO TO 1000
2700 IF ## THEN PRINT AT 10,11:
FLASH 1: "MURO SOLIDO": BEEP 1,-1

```

```

0: LET a=NOT a
0000 GO TO 100
0000 REM fim do JOGO
0010 CLS
0020 PRINT AT 10,0:"Voce morreu
de fome."
0030 PRINT "Os seus ossos Junta
#1-se aos de muitos outros."
0040 GO TO 4000
0050 REM fim
0060 PAPER 0: INK 7: CLS
0070 GO SUB 7000: PRINT AT 4,0:
0080 FOR i=40 TO 40 STEP .5: GOTO
0090 PRINT "Optimo! Consegui
0100 NEXT i
0110 FLASH 1
0120 PRINT INK 6;"Executou a sua
missao com"
0130 INK 4
0140 IF ((10)>200) THEN PRINT "er
dade capacidade."
0150 IF ((10)<50) AND ((10)>100
THEN PRINT "bastante merito."
0160 IF ((10)<100) THEN PRINT "co
nsideavel porte."
0170 GO TO 4000
0180 REM construir labirinto
0190 PAPER 0: INK 5: CLS: PRINT
"O labirinto pode levar bastant
e tempo a ser definido, pelo que
convem guardar os maiores em
casette."
0200 PRINT "Por favor indique
as dimensoes do labirinto prate
ndido, com uma largura entre 20 e
100 e um comprimento entre 20
e 100"
0210 INPUT "Quer carregar um lab
irinto s/n? "; LINE a#
0220 IF a#"s" THEN GO SUB 7000:
RETURN
0230 INPUT "Indique largura: ";#
: LET #=INT (#/2): IF #<10 OR #>
50 THEN GO TO 0210
0240 INPUT "Indique comprimento:
";l: LET l=INT (l/2): IF l<10 O
R l>50 THEN GO TO 0210
0250 LET s=INT (#*(l/3)): LET s=IN
T (s+s/10): DIM a(s): DIM d(s)
0260 LET s=s+INT (RND*s/3): LET
t=INT (s/60): LET t1=(s-t*60)/10
0: LET t=t+t1

```

```

1030 POKE 23674,0: POKE 23673,0:
POKE 23672,0
1035 PRINT "Desorte aproximada
mente "t," minutos."
1040 LET #=#*2+1: LET n=l*2+1: G
O
1050 FOR i=1 TO #: LET ##(1,i)="
#": LET ##(n,i)="#" : NEXT i
1060 FOR i=2 TO n-1: LET ##(1,i)
="#" : LET ##(n,i)="#" : NEXT i
1070 LET nn=2*INT (RND*(l-2))+2
1080 LET ##=2*INT (RND*(#-2))+2
1090 LET ##(nn,##)="#" : LET #00
1100 IF ##(nn-2,##)="#" THEN GO
TO 1210
1110 IF ##(nn+2,##)="#" THEN GO
TO 1210
1120 IF ##(nn,##-2)="#" THEN GO
TO 1210
1130 IF ##(nn,##+2)="#" THEN GO
TO 1210
1140 IF k-1=0 THEN GO TO 1420
1150 IF d(k)=1 THEN LET nn=nn+l(k)
1160 IF d(k)=2 THEN LET nn=nn-l(k)
1170 IF d(k)=3 THEN LET ##=##+l(k)
1180 IF d(k)=4 THEN LET ##=##-l(k)
1190 LET k=k-1
1200 GO TO 1100
1210 LET l1=2*INT (RND*3)+2
1220 IF l1=0 THEN GO TO 1210
1230 LET d1=INT (RND*4)+1
1240 IF d1>2 THEN GO TO 1230
1250 LET s=-1+2*(d1=2)
1260 LET t=0
1270 GO TO 1300
1280 LET s=0
1290 LET t=-1+2*(d1=4)
1300 FOR i=2 TO (l1 STEP 2
1310 IF ##(nn+s*i,##+t*i)="#" TH
EN GO TO 1210
1320 NEXT i
1330 FOR i=1 TO l1
1340 LET ##(nn+s*i,##+t*i)="#"
1350 NEXT i
1360 LET nn=nn+s*l1
1370 LET ##=##+t*l1
1380 LET k=k+1
1390 LET l(k)=l1

```

```

1400 LET d(k)=41
1410 GO TO 1100
1420 REM no. de portas
1425 LET nd=INT ((4)*(5)/25):
LET st=nd/4
1430 LET l(4)=m: LET l(5)=n
1435 REM esconder objectos
1440 FOR o=1 TO 4
1450 FOR i=1 TO st
1460 LET l=FN r((l(5)-3)+1): LET c
=FN r((l(4)-3)+1): LET kn=FN r(9)*
10+o
1470 IF ##(l,c)<>"#" THEN GO TO
1480 LET ##(l,c)=CHR# (40+kn)
1490 NEXT i
1500 NEXT o
1510 REM esconder alimentos e vi
10es
1520 FOR o=5 TO 7 STEP 2
1530 FOR i=1 TO st*2
1540 LET l=FN r((l(5)-3)+1): LET c
=FN r((l(4)-3)+1): LET kn=FN r(9)*
10+o
1550 IF ##(l,c)<>"#" THEN GO TO
1560 LET ##(l,c)=CHR# (40+kn)
1570 NEXT i: NEXT o
1580 REM esconder dinheiro
1590 FOR i=1 TO st*4
1600 LET l=FN r((l(5)-3)+1): LET c
=FN r((l(4)-3)+1): LET kn=FN r(9)*
10+o
1610 IF ##(l,c)<>"#" THEN GO TO
1620 LET ##(l,c)=CHR# (40+kn)
1630 NEXT i
1640 LET ##((l(5)/2),(l(4)/2))=CHR#
(40+FN r(9)*10+3)
1650 REM esconder chaves
1660 FOR i=1 TO 4*st
1670 LET l=FN r((l(5)-4)+2): LET c
=FN r((l(4)-4)+2)
1680 IF ##(l,c)<>" " THEN GO TO
1690 LET ##(l,c)=CHR# 120
1700 NEXT i
1705 LET sat=1
1710 LET time=(PEEK 230672+255*PE
EK 230673+65536*PEEK 230674)
1720 LET time=time/55: BEEP 2,10
1730 INPUT "O labirinto descorou

```

```

";(time);" segundos","a construi
r); quer grava-lo?");as
1740 IF ##(1)=s THEN GO SUB 75
80
1750 CLS : RETURN
2000 REM image#
2010 BORDER 1: PAPER 0: BRIGHT 1
2020 LET x=24: LET y=15: REM ang
ulo
2030 FOR i=1 TO 6
2040 LET ly=oy+((f=3)-(f=1))*i
2050 LET lx=ox+((f=2)-(f=4))*i
2060 IF ##(ly,lx)<"#?" OR ##(ly,
x)=CHR# 120 THEN NEXT i
2065 LET dt=(##(ly,lx)=" ")
2070 LET vs=i-1-(i>5): REM maxim
a distancia de vista
2075 IF NOT vs AND ##(ly,lx)>"#
" THEN GO SUB 2500

```

```

2080 LET vx=vw*x: LET vy=vw*y
2090 FOR a=0 TO vw: REM divisao
de paisais
2100 LET b=255-d*x*2
2110 PLOT a*x,d*y: DRAW 0,175-(y
*d)*2: IF d=## THEN DRAW b,0
2120 PLOT 255-d*x,175-d*y: DRAW
0,-(175-y*d)*2: DRAW -b,0
2130 PLOT 254-d*x,175-d*y: DRAW
0,-(175-y*d)*2
2140 NEXT d
2140 FOR i=0 TO vw-1: REM desenh
ar muros
2150 LET ly=oy+((f=3)-(f=1))*(i+
1)
2160 LET lx=ox+((f=2)-(f=4))*(i+
1)
2170 LET d#=#(ly-(f=2)+(f=4),lx
-(f=1)+(f=3))
2175 LET wall=(d#="#"): LET rdr
=(d#="#")
2180 LET d#=#(ly+(f=2)-(f=4),lx
+(f=1)-(f=3))
2185 LET wall=(d#="#"): LET rdr
=(d#="#")
2190 IF (ldr+rdr) THEN GO SUB 23
00
2197 GO TO ldr*7+2000
2200 PLOT i*x,i*y+4*NOT wall: D
RAW x,c*(wall)
2205 PLOT i*x,175-(i*y+4*NOT wa

```

```

(1): DRAW X, -(G*(W/10))
0007 GO TO 747:RND*(200+200)
0010 PLOT 200-1*X,1*U+4*NOT W/10
(2): DRAW X, G*(W/10)
0020 PLOT 200-1*X,1*U-(1*U+4*NOT
W/10): DRAW X, -(G*(W/10))
0030 NEXT I
0040 INK 5: LET P=5
0050 LET V=200/P: LET S=(200-10**
X)/P
FOR I=1 TO P-1
0070 LET S=(IX+S*1)-V*1
0080 PLOT V*1,S: DRAW X,S*U
0090 NEXT I
0110 IF AT THEN PRINT FLASH 1: O
VER 1, AT 10,10,"AB": AT 11,10,"OO
": RM graphics A, B, C e D
0130 IF NOT AT THEN RETURN
0140 LET I=1
0150 WHILE VERIFICAR REPETICAO
0160 FOR J=0 TO 1
0170 LET T=U+(I=0)-(I=1)*1
0180 LET TX=PX+(I=0)-(I=1)*1
0190 LET T=U*(S*(U+(I=2)-(I=4))
TX+(I=3)-(I=1))="#":
0200 LET T=U*(S*(U+(I=2)+(I=4))
TX-(I=3)+(I=1))="#":
0210 NEXT I: RETURN
0230 RM desenhar porta
0240 LET K=1/(I+1)*5
0250 FOR J=1 TO 7
0260 IF LET THEN PLOT I*X+J*K,I*
U+J*K
0270 IF LET THEN DRAW INK 5;0,17
U-1*(U+J*K)*K
0280 IF LET THEN PLOT 200-(I*X+J
*K)*1,I*(U+J*K)*K
0290 IF LET THEN DRAW INK 5;0,17
U-1*(U+J*K)*K
0300 NEXT J
0310 RETURN
0320 PLOT 100,100+K: DRAW X,0,-P
H*1.75
0330 DRAW X/2,S,-K*1.5: DRAW -X/
2,K,S,-PI: DRAW X/2,S,K*1.5
0340 RETURN
0350 RM porta
0360 DRAW 0: INK 5: CLS
0370 PRINT AT 10,4;"B PORTA": BE
T 0,0,10
0380 LPRINT 0: INK 5
0390 RETURN


```

```

0400 RM
0410 IF (10) THEN GO TO 3000
0420 PRINT AT 2,4: FLASH 1,"US#
cheve, deo porta..."
0430 LET X=PX-(I=0)+(I=4): LET P
U*(S*(I=0)+(I=2)+(I=4))
0440 DRAW P 1,-P: GO TO 100
0450 RM
0470 IF PX>INT P(5)/2 AND PX<
VINT P(4)/2 THEN RETURN
0480 CLS: RESTORE 0000
0490 PRINT "B porta deve-se com
ce r(10)..."
0510 FOR I=-30 TO -20 STEP .5: S
T 0,0,10: NEXT I
0520 FOR I=1 TO FN r(8): READ T#
: NEXT I
0530 PRINT "E no centro de sala
você vê o que o trouxe aqui..."
0540 PRINT T#
0570 PRINT "Agora deve levar as
re objecto para uma saída situ
ada..."
0575 LET S*(P4,PX)=" "
0580 LET S*(I=FN r(10)): LET S2=P
U*(I=4)
0585 LET S*(S1,S2)=" "
0590 GO TO 0000
0610 LET S*(FN r(100)): LET I(11)=
I(11)+S
0620 PRINT "Encontrou tambem um
e vitamina" que aumenta a sua e
nergia de "
0640 DRAW 0: PAPER 0: INK 5
0650 DRAW 0
0660 GO TO 200
0680 DATA "Um escravo para toda
a vida", "Um diárete para Tombuot
U", "Um aspirador automatico", "A
bola de vlnel dos Campeões", "U
roreco", "a solucao do problema d
e caspo"
0690 DATA "Um NX 00 gratis...", "
Um abastecimento eterno de tinto
", "Um milhoes em barras de ouro"
0700 RM abrir porta
0710 LET J=0000 S*(P4,PX)-40: LE
T I=INT (J/10)
0720 PRINT AT 0,0: INK 5;"A port
a tem uma fechadura no." I
0730 IF (10) THEN PRINT "As suas
cheves abrem a porta!": PAUSE 1

```

```

3030 GO TO 3070
3030 IF (I(12)<>) THEN PRINT AT
21,0: INK 0;"A sua chave nao ser
ve...": GO TO 3030
3040 PRINT AT 0,0;"A chave serv
e! Quer saber a fe- chadura (a/n
)?": BEEP 1,10
3050 IF INKEY#="n" THEN GO TO 20
00
3060 IF INKEY#<>"e" THEN GO TO 3
050
3065 GO SUB 2700
3070 LET rate=(I-1)*10: BORDER 4:
PAGE 0: INK 1: CLS
3073 IF rate=0 THEN GO TO 2700
3075 PRINT "NO INTERIOR, esta'..
": BEEP .5,0: BEEP 1,10
3080 GO TO rate*200+3000
3090 REM vario
3100 PRINT AT 3,0;"NADA!";AT 5,0
;"Alguem esteve aqui antes."
3110 GO TO (AND>.5)*80+3120
3120 REM perder chave
3130 PRINT "A chave esta' presa
na fechadu- ra! Quer tentar lib
erta-la?"
3140 PAUSE 0: IF INKEY#<>"s" THE
N GO TO 3080
3150 IF (I(6) THEN PRINT "OK, nao
deve demorar muito usando o la
ser.": LET I(6)=I(6)-1: LET i=1:
GO TO 3100
3160 PRINT "OK, espero um moment
o enquanto voce tenta."
3170 PAUSE 70: FOR i=0 TO 10+FN
r(50): BEEP .01,AND*30-15: NEXT
i
3180 IF AND>.7 THEN LET I(13)=I(
13)-1: PRINT "Tera de a deixar a
qui.": GO TO 3100
3185 PRINT "Conseguiu!""Mas "
3190 PRINT "perdeu "i;" unidada
s de energia""na sua tentativa.
": LET I(11)=I(11)-1
3200 GO TO 4000
3200 REM laser
3200 LET U=FN r(6)+2: PRINT AT 3
,0;4*(1)"com pilhas suficientes
para"U,"utilizacoes!"
3310 IF (I(6) THEN PRINT AT 3,0)"
Outro "
3315 PRINT TAB 10;"TAB 1

```

```

3320 PRINT AT 9,0)"O laser tem U
# factor de consumo igual a 9. Qu
ere-o?"
3330 IF INKEY#="n" THEN GO TO 40
00
3335 IF INKEY#<>"s" THEN GO TO 3
300
3340 IF NOT (I(6) THEN LET I(13)=
I(13)+9: GO TO 3360
3350 IF (I(6) THEN PRINT "Como ja
tem um laser tera de le-var as
pilhas."
3350 LET I(6)=I(6)+U: BEEP .2,3:
BEEP .2,10: BEEP .2,0: BEEP .7,
10
3370 PRINT "Voce e' agora o ora
tilhoso pos- suidor de um laser
com ";I(6),"utilizacoes garantid
as!"
3380 GO TO 4000
3400 REM desmaterializador
3500 PRINT AT 3,0;4*(2);AT 3,0)"
Outro " AND (I(7)
3510 PRINT AT 5,11;"TAB 5)"
3520 PRINT "Isto permite-lhe de
sintegrar pa- redes solidas com d
ois disparos e esturrar qualquer
malandro que encontre."
3530 LET U=FN r(6)+2
3540 PRINT "Este tem ";U;" cara
s";4*(2);" tem um""factor de c
onsumo de 14"
3550 PRINT "Quer leva-lo consig
o?"
3560 IF INKEY#="n" THEN GO TO 40
00
3565 IF INKEY#<>"s" THEN GO TO 3
500
3570 CLS: FOR i=0 TO 10: BEEP .
05,i: NEXT i
3580 PRINT FLASH 1;AT 2,11;"PARA
BONS "
3590 PRINT "Voce nao precisou d
e pagar muito para ficar co
m esta formidavel arma, a melho
r de qualquer armaria."
3600 IF (I(7) THEN PRINT "Voce n
ao quer certamente andar com do
is $0" para ter as cargas gratis
...

```

```

3610 PRINT "Para usar esta arma
, aperte parao seu alvo e carreg
ue em 'X'"
3620 PRINT "Pode usa-la ";L(7)+
U;" vezes."
3630 PRINT "INVERSE 1;"DUISD GO
GOVERNO: ESTA ARMA PODERAZER-LH
E MAL A SAUDE."
3670 LET L(10)=L(10)+14: LET L(7
)=L(7)+U
3680 GO TO 4000
3690 REM correia transportadora
3700 PRINT AT 2,0;A#(3);AT 2,0;"
Outro " AND L(8)
3710 PRINT INK 4; PAPER 0; AT 4,1
9;" "AT 9,7; INVERSE 1;" += += +
:= "; FLASH 1;"T"; FLASH 0; INU
VERSE 1;" += += += "; AT 9,15;
3720 PRINT "Isto permite-lhe via
jar no hi- perespaco para difer
entes partes do labirinto, sem ne
o ter a qual-quer controle sobre
o destino, esse poder numa parede
sufocara imediatamente"
3730 LET U="N";L(3)+U
3740 PRINT "Este e' mais um pro
duto moderno, de deitar fora depo
is de usar, devendo ser lançado
ao fim de ";U;"utilizacoes."
3750 PRINT "A#(3); tem um facto
r de "; "consumo de U"
3760 PRINT AT 20,0;"Deseja usar
este produto?"
3770 IF INKEY#="n" THEN GO TO 40
00
3775 IF INKEY#<>"s" THEN GO TO 3
770
3790 CLS : LET L(8)=U: LET L(13)
=L(13)+9
3800 PRINT AT 1,0;"VENDIDO! A o
usada pessoa senta-se atraz do Z
X Spectrum."
3810 PRINT "Use esta correia ap
enas ";U;"vezes." "Para saltar
no hiperespaco car- regue em 't'"
3820 PRINT "INVERSE 1;"O FABRIC
ANTE NAO PODE SER RES- PONSABIL
IZADO PELO USO IMPROPRIO DESTE TR
ANSPORTADOR.
3830 FOR I=1 TO 5: BEEP .75,I: N
EXT I

```

```

3840 GO TO 4000
3890 REM chave
3900 PRINT AT 2,0;A#(4);
3910 PRINT AT 2,0;" "TDS 11;"
3920 PRINT AT 2,0;" "TDS 11;"
3930 PRINT "Esta chave de- lta
acesso a qual-quer porta de labi
rinto!"
3940 PRINT "Pode usa-la tantas
vezes quantas quiser." "Tem um f
actor de ";U;"
3950 PRINT FLASH 1;"Quer leva-la
consigo?"
3960 IF INKEY#="n" THEN GO TO 40
00
3970 IF INKEY#<>"s" THEN GO TO 3
960
3980 FOR I=1 TO 10: BEEP .1,5: P
AUSE 10: NEXT I
3990 CLS
4010 IF NOT L(9) THEN LET L(13)=
L(13)+3
4020 LET L(10)=1
4030 GO TO 4000
4040 REM alienígenos (energia)
4050 PRINT AT 2,10;"Comida!"
4060 PRINT "Voce corre para a m
esa e engole "; "toda a comida."
4070 LET am=FN r(20)+10
4080 PRINT AT 10,0; INVERSE 1;"O
brigue numa tecla." AT 7,0;"A c
omida dev- lta "; PAUSE 0
4090 IF am>25+FN r(6) THEN PRINT
"Indigestão! Não consegue mais
- quer pontos de energia!"; BEEP
1,-20: GO TO 4000
4100 PRINT am;" valiosos pontos
de energia."
4110 LET L(11)=L(11)+am
4120 GO TO 4000
4130 REM tesouro
4140 FOR I=1 TO FN r(5): READ t#
,am: NEXT I
4150 PRINT AT 2,0;A#(5);AT 2,17;
"i"; "Voce tem ";t#;
4160 LET am=FN r(am)+5: LET en=I
NT (am*5)
4170 PRINT "Tem o valor de ";am;
"pontos" "de tesouro e um factor
de con- sumo de ";en
4180 PRINT "O tesouro pode serv

```



```

in para passear nos monstros que eu
ardam e os seus passagens."
4360 PRINT "Queir leve-lo consigo
o?"
4370 IF INKEY#="n" THEN GO TO 43
4380 IF INKEY#>"s" THEN GO TO 4
370
4390 LET L(13)=L(13)+ten: LET L(1
9)=L(19)+am
4400 PRINT "Voce ganhou agora "
L(13), "pontos de tesouro."
4410 PRINT "O seu factor de con
vulsao e: Ja ""L(13)"; "Onde
movimento de ""L(13)"; "Unidad
es de emergencia."
4420 GO TO 4000
4430 DATA "Ouro!",60,"Um sacco de
prata!",30,"Uma bolsa cheia de
diamantes",40,"Um sacco de vici
tudo",200,"Um vaso com rubis",200
4435 PRINT
4440 PRINT
4510 DATA "Um irado", "Um louco",
"Um feroz", "Um horrivel", "Um ser
cruel", "Um violento", "Um selvagem",
"Um medonho"
4520 FOR i=1 TO FN r(8): READ t#
: NEXT i: PRINT
4530 DATA "gordito", "grande", "oio
ante", "redondo", "oblongo", "de
#001",0#
4540 FOR i=1 TO FN r(7): READ t#
: NEXT i
4550 PRINT INK 3; AT 2,5; a#; " "t
#
4560 PRINT INK 1; "Ele vai arranc
ar-lhe os bracos por te-lo acor
deado!"
4570 PRINT INK 0; "Que vai fazer
?"
4580 PRINT INK 1; "1/ sugar" "2/
pedar-lhe" "3/ correr" "4/ fazer
perra que ele desista"
4590 IF INKEY#="1" OR INKEY#>"4"
THEN GO TO 4000
4600 GOTO
4610 GO TO VAL INKEY#*50+4500
4610 PRINT "Voce tem "
4610 IF L(6) THEN PRINT "Um case
r."
4614 IF L(7) THEN PRINT "Um caio

```

```

o do tipo de materia."
4610 IF NOT (L(6)+L(7)) THEN PRI
NT "Nada."
4620 LET pr=(L(7)>0)+.8*L(6)>0
): IF NOT pr THEN LET pr=AND/2
4630 PRINT AT 2,1; "Carregue num
a tecla para lutar"; PAUSE 0
4640 FOR i=1 TO 5*(1/pr)
4650 LET ch=FN r(10)-1: BORDER of
: PRINT ch; INK 7-cl: LET m=USR
invert: BEEP .05,10-7*(100)
4660 IF L(6) THEN BORDER 2: BEEP
pr: BEEP .070,30: LET m=USR in
vert
4670 IF L(7) THEN BORDER 0: BEEP
7: BEEP .1,0: LET m=USR inver
t
4680 LET L(11)=L(11)-2
4690 NEXT i: BORDER 0: INK 4: PD
PRINT 0; BRIGHT 0: CLS
4694 IF AND pr THEN GO TO 4640
4698 PRINT AT 5,1; "PERDUI O "a
#(4 TO 1); " "t4"; "Atirou o seu c
orpo para um cantonete de Juroc
os e ficou outroque oscaram vi
r"
4700 FOR i=0 TO -30 STEP -1: BEE
P .5,1: NEXT i
4710 GOTO 4600
4720 BEEP 20000,50: PRINT AT 2,0
"Queir tentar de novo?"
4744 IF INKEY#="n" THEN CLS: GT
O
4746 IF INKEY#="s" THEN GO TO 50
4747 GO TO 4044
4748 PRINT AT 5,10; "Ganhou!"
4750 IF L(7) THEN PRINT "Mas com
o ralo antimateria nao podia t
er perdido!" : LET L(7)=L(7)-1: G
O TO 4000
4752 IF L(6) THEN PRINT "Trabalh
ou um caso o labor!" : LET L(6)=L
(6)-1: GO TO 4000
4754 PRINT "Que golpe! O "a#(4
TO 1); " "a# deve sentir-se hoje #a
(4 TO 0).
4755 GO TO 4000
4755 PRINT "Que vai dar so "a#(
4 TO 1); "t#"; "3"
4755 PRINT "Tem consigo.": FOR
a=1 TO 5: PRINT a#(a) AND (a+0)
: NEXT a

```

```

4670 DIM a$(18): INPUT a$
4673 FOR i=1 TO 6: BEEP .1,5: IF
a$=a$(i) THEN GO TO 4680
4676 NEXT i: PRINT AT 12,0;"Nao
percebi, tente de novo!": BEEP
.5,5: PAUSE 5: DIM a$(24): PRINT
AT 12,0;a$: GO TO 4673
4680 IF i=5 THEN GO TO 4695
4682 PRINT "O nao tao ";a$(4 TO
1)" ";t$;"agradece e aceita o
";a$(1,3 TO 1)"desaparecendo em 5
seguidas."
4684 LET l=(i+5)=6: LET r=(i-
3)=e(i)
4686 GO TO 4690
4690 INPUT "Quanto vai dar ao ";
(t$);"?";am
4693 IF am>0 THEN GO TO 4700
4694 PRINT "O ";t$;" ficou furio
so" "por voce tentar enganar-lo,
e ti-rou-lhe tudo excepto metade
do seu dinheiro."
4695 FOR i=0 TO -20 STEP -1: BEE
P .5,5,i: NEXT i
4696 LET t=l(11)/2: LET l=l(10):
LET m=l(4): LET n=l(9): DIM l(1
3)
4697 LET l(13)=1: LET l(4)=m: LE
T l(5)=n: LET l(10)=l: LET l(11)
=t
4698 GO TO 4690
4700 IF am>FN r(20)+10 THEN PRIN
T "O ";t$;" aceita a sua oferta"
"oferta e diz-lhe que sera" vind
e mais ";a$(4 TO 1)"da proxima ve
z.": BEEP 1,10: GO TO 4690
4702 PRINT "O ";t$;" "insultad
o pela sua oferta, fica mais ";a
$(4 TO 1)"e rouba-lhe o ";
4704 LET o=FN r(5): IF o=5 THEN
LET o=5
4706 LET l(13)=l(13)-e(o): LET l
(o+5+(o=5))=6
4708 PRINT a$(o,4 TO )
4709 GO TO 4690
4710 IF l(8) THEN PRINT "Voce ac
tivou o seu transportadore zzzzap
! Desapareceu!": BEEP 1,50: GO T
O 4690

```

```

4711 FOR i=0 TO FN r(40)+10 STEP
.5: BORDER FN r(7): BEEP .01,i:
LET l(11)=l(11)-.5: NEXT i: FOR
i=0 TO FN r(50): BEEP .01,-i: N
EXT i
4714 LET r=2-FN r(3)
4718 IF NOT r THEN GO TO 4730
4720 PRINT "Nao foi suficiente me
nte rapido! O ";t$;"corra muito.
";a$;"agora tem" de lutar."
4725 GO TO 4618
4730 PRINT "Conseguiu! Voce devi
e ir"as Olimpíadas fazer a #
aratonas!"
4740 PRINT "Depois de um rapido
sprint voce vem parar ao mesmo s
itio!"
4745 PRINT AT 20,0;"Descanse um
pouco, e depois cor-reque numa t
erça."
4750 BEEP .5,-50: BEEP .3,-40
4755 IF INKEY#="" THEN GO TO 475
0
4758 GO TO 4690
4760 FOR i=0 TO 5: PAPER FN r(7)
: BEEP .1,50: CLS : BEEP .5,-50:
NEXT i
4765 PAPER 6: INK 0: CLS
4770 PRINT "Nao resultou! O ";t$
;"ainda esta aqui!"
4773 LET m=FN r(10): LET l(11)=l
(11)-m
4775 PRINT "Gastou ";m;" unidade
s de energia" "a tentar."
4780 GO TO 4670
4800 PRINT "OK."
4805 GO SUB 4400: REM esperar
4810 BEEP 1,5
4820 BORDER 6: PAPER 0: INK 5: B
RIGHT 1: CLS
4840 LET ##(70,70)=" "
4850 GO TO 4830
4860 REM mapa
4865 IF l(10)<10 THEN PRINT INK
0;AT 5,0;"Nao tem dinheiro sufic
iente para pagar, sem dinheiro, n
ao ha" "ma-va.": BEEP .5,-30: GO
TO 100
4810 BORDER 5: BRIGHT 0: CLS : P
RINT AT 4,0; INK 6;"****NORTE***
*"
4820 IF INKEY#<>"" THEN GO TO 50
20

```

```

0030 LET ##=(PX-5)+(PX>5)+1: LET
0040 #4=(PX-10)+(PX>10)+1
0040 IF ##+10>(14) THEN LET #X=(
+4)-10
0050 IF #4+10>(15) THEN LET #4=(
+5)-10
0060 LET #=#(PY,PX): LET ##(PY
,PX)="*":
0070 FOR I=#4 TO #4+10: PRINT IN
K 0;TAB 0;"*";
0080 FOR C=#X TO #X+10
0090 IF CODE# ##(I,C)>48 AND CODE
##(I,C)<100 THEN PRINT "0"; GO
TO 0100: REM 0100 E; 0101#1
7 0010#
0100 IF ##(I,C)>"5" THEN PRINT #
0101 I; INK 0;"X"; GO TO 0100:
REM 0100# X
0110 PRINT ##(I,C)
0120 NEXT C: PRINT INK 0;"*";
0130 NEXT I: PRINT INK 0;TAB 0;"
*****SUL*****"
0140 LET L(13)=L(13)-2: LET L(10
)=L(10)-10: REM 0100 #10 0100 #
0100
0100 LET ##(PY,PX)=#4#
0100 PRINT INK 4; BRIGHT 1; AT 0,
0;"Energia restante=";L(14)"Por
tos de recursos=";L(10)"0 seu co
ntorno por movimento 0";L(10)
0170 PRINT INK 0; AT 17,0;"Voce t
raz comida";
0180 IF L(6) THEN PRINT INK 0;#4#
(1);L(6)"disparos."
0190 IF L(7) THEN PRINT #4#(2);L(
7)"disparos."
0200 IF L(8) THEN PRINT INK 0;#4#
(3)"Vida de ";L(8)
0210 IF L(9) THEN PRINT INK 4;#4#
(4)
0212 IF L(10) THEN PRINT INK 0;"
Use chave"
0220 IF NOT (L(10)+L(6)+L(7)+L(8
)+L(9)) THEN PRINT FLASH 1;"Mada
"
0225 IF INKEY#<>" " THEN GO TO 02
00#
0227 GO SUB 0000
0230 PAUSE 000: RETURN
0230 IF NOT 91 THEN RETURN
0230 PRINT "Energia este ";"N
OTR-" AND PY>91;"SUL-" AND PY<9
4;

```

```

0010 PRINT "OESTE-" AND PX>92;"E
STE-" AND PX<92;"
0020 RETURN
0030 REM chave
0010 PRINT AT 1,0;"Voce encontra
o chave!"
0020 BEEP .75,10
0030 IF NOT L(10) THEN PRINT AT
21,0;"Quer espanta-la?"
0040 IF L(10) THEN PRINT AT 21,0
;"Quer trocar de chaves?"
0050 LET ##(PY,PX)=" ": REM 000
00
0060 IF INKEY#="0" THEN RETURN
0070 IF INKEY#<>"s" THEN GO TO 0
000
0075 IF L(10)<1 OR NOT L(12) THE
N LET L(13)=L(13)+1
0080 BORDER 2: PAPER 4: INK 0: C
LS
0090 LET key=FN r(5): IF key=L(1
0) THEN GO TO 0095
0095 LET L(12)=key
0100 BEEP .1,0: BEEP .4,10: BEEP
.2,0: BEEP .5,10
0110 PRINT AT 19,2;"TEM 0000 UM
A CHAVE MARROVIM L(10) "Guarde-a
bem - a sua vida pode depender
dela."
0120 GO SUB 0400
0140 BORDER 0: PAPER 0: INK 0: C
LS
0150 RETURN
0000 REM transportador
0010 LET px=FN r(14): LET py=FN
r(15): LET r=FN r(14)
0020 FOR i=0 TO 10: LET cl=FN r(
0)-1: BORDER cl: PAPER 7-cl: BEE
P .01,10-FN r(20): CLS: NEXT I
0030 FOR i=1 TO 5
0040 LET cl=FN r(3)-1: BEEP .1,4
00
0050 BORDER cl: PAPER 7-cl: CLS
0060 PAUSE 20: POKE 20000,50: CL
S
0070 NEXT I: BORDER 1
0080 PRINT AT 0,0;"Voce foi tran
sportado para outra parte do labi
rinto."
0090 LET L(8)=L(8)-1: IF NOT L(8
) THEN LET L(13)=L(13)-5
0090 GO SUB 0400: RETURN

```

```

7000 REM largar objecto
7005 PAPER 0: INK 0: BORDER 4: C
LS
7010 INPUT "Que quer largar? "
n#: "?"
7020 IF n#="1" THEN RETURN
7030 FOR i=1 TO 5
7040 IF i=n# THEN GO TO 7070
7050 BEEP .1,10-FN r(20): NEXT i
7060 PRINT "Verdade, nao sei o qu
e e " n# "Tente de novo.": GO T
O 7010
7070 IF (i+5) THEN GO TO 7000
7080 PRINT "Mas voce nao tem " n#
" Tente de novo.": GO TO 7010
7090 IF i=5 THEN GO TO 7140
7100 PRINT "OK, diga adeus ao se
u " n# "
7110 LET l(13)=l(13)-e(i)
7120 LET l(1+5)=0
7130 BEEP 1,10: GO TO 7100
7140 INPUT "Quanto dinheiro quer
deixar? " tr
7150 IF tr>l(10) THEN PRINT "Uoc
e nao tem tanto, " n# : BEEP 1,
10: GO TO 7140
7160 LET l(13)=l(13)-tr/5: LET l
(10)=l(10)-tr
7170 PRINT "Diga adeus a " /tr "
pontos " de tesouro."
7180 IF l(13)<=0 THEN LET l(13)=
0
7190 BORDER 0: PAPER 0: INK 0: C
LS : RETURN
7200 REM desmaterializador
7210 LET dx=py+(r=0)-(r=1): LET
dx=dx+(r=2)-(r=4)
7220 IF dx>l(5)-1 OR dx<0 OR dx<
0 OR dx>l(4)-1 THEN PRINT "Nao p
ode desmaterializar a parte de ex
terior": PRUSRN 0: RETURN
7230 LET n#(dx,dx)=" "
7240 FOR i=40 TO 55: LET c1=0-FN
r(3)
7250 PAPER c1: BORDER 7-c1: CLS
7260 BEEP .05,i: NEXT i
7270 LET l(7)=l(7)-1
7280 BEEP .5,-20: RETURN
7290 REM gravar
7300 LET l(1)=py: LET l(2)=px: L
ET l(3)=r
7310 PRINT FLASH 1;AT 10,12;"Gra
vado"

```

```

7320 PRINT INVERSE 1;"Nome " "Lab
irinto" " " ;AT 20,0;"Carregue nume
ro de teclado limpar eimages."
7330 SAVE "Labirinto" DATA l():
CLS : SAVE "Labirinto" DATA n#()
7340 INPUT "Verificacao? " n#: IF n#
#>"0" THEN RETURN
7350 PRINT "Rebobine a cassette
e carregue em PLAY. " "Use a ver
ificacao de erro, se - creca: 0
0 TO 7000"
7360 VERIFY "Labirinto" DATA l()
: VERIFY "Labirinto" DATA n#()
7370 RETURN
7380 PRINT FLASH 1;AT 10,11;"Ler
dados."
7390 PRINT #0: FLASH 1;AT 1,0;"P
ode gravador em funcionamento."
7400 LOAD "Labirinto" DATA l():
LOAD "Labirinto" DATA n#()
7410 LET py=l(1): LET px=l(2): L
ET r=l(3)
7420 CLS
7430 RETURN
0000 REM variaveis
0010 IF NOT set THEN GO TO 0070
0020 LET px=2: LET py=px: LET r=
0: REM posicao do jogador
0030 DIM l(10): LET l(4)=0: LET
l(5)=0
0040 LET l(10)=200: REM tesour
o
0050 LET l(13)=1: REM consumo de
energia
0060 LET l(11)=INT (l(4)*l(5)/5)
0070 LET r=0: LET re=r: REM repe
tir
0080 LET g1=0: LET g=0
0090 REM definir graficos
0100 RESTORE 0050
0110 FOR i=1 TO 7: READ c#
0120 FOR e=0 TO 7: READ n: POKE
USR c#+e,n
0130 NEXT e: NEXT i
0140 FOR i=USR "s" TO USR "s"+10
0150 READ b: POKE i,b: NEXT i
0160 LET invert=USR "s"
0170 DIM a$(0,10): DIM e(5)
0180 FOR i=1 TO 6: READ g#,e: LE
T a$(i)=g#: LET e(i)=e: NEXT i
0190 BORDER 0: PAPER 0: INK 5: B
RIGHT 1: CLS

```



```

O tráfego por si seguido. Nos
tráfego também portas e chaves
proximas.
0000 PRINT "Sempre que pedir o
me ("M") perderei 10 pontos de
te ouro."
0010 GO SUB 0400: BORDO 0: CLS
0020 PRINT "Comando": PL
0030 ,107: DADO INK 4:70:0
0040 PRINT "M - Para cima D -
Direita M - Virar para tras.
.
0050 PRINT "M - Inverter"
0060 PRINT "D - Gravar jogo"
- repetir (se estiver num cor-
- redor com o lado)."
0070 PRINT INK 0: "Cada objecto r
ecolhido tem um factor de con-
umo indicado a energia que o
obriga a perderes cada passo.
0070 PRINT INK 7: "Voce pode esse
caricar as diques - se os labiri-
nto derreido, mas os maiores das
traco entre de minutos a ser
erados."
0080 PRINT INK 0: "pode gravar os
labirintos em cadaite para
o futuro."
0090 GO SUB 0400
0100 RETURN

```

JOGOS COM GRÁFICOS MÓVEIS

CORRIDA AO OURO

A sua tarefa, neste jogo escrito por Neil Pellinacci, consiste em transportar seis sacos cheios de ouro de uma caixa-forte de um banco para um cofre. Ambos estão separados por um rio. Este deve ser atravessado através de uma rede de pequenos diques. Infelizmente, o futuro destes não é garantido, porque troncos de árvore, descendo o rio, destroem ocasionalmente partes do sistema de diques. Você deve transferir todos os sacos para o cofre antes de o caminho ser cortado.

Você joga usando as teclas seguintes:

- O — Subir
- Z — Descer
- I — Esquerda
- P — Direita

A tarefa que você está a executar é indicada pela cor do seu homem. O branco indica que pegou num saco, e o amarelo que o está a transportar. Para pegar um saco, você desloca-se para o mesmo quadrado onde este se encontra. Coloca-o depois no cofre entrando neste. Em ambos os casos ouvirá um som, e o homem mudará de cor. Dispõe de três homens por jogo, e o número destes que resta em cada momento é indicado na parte superior da imagem.

Os troncos eliminam todas as secções de diques que encontram no caminho. No entanto, as secções acima e abaixo também são danificadas, e se você passar por uma destas e entrar numa zona do rio por onde tenha passado um tronco, perderá uma vida. Esta é a única maneira de você perder uma vida.

Depois de ter colocado os seis sacos no cofre, você passa ao nível seguinte. Neste existem duas formações de diques, que alternam entre si. A medida que o jogo progride, o número de troncos aumenta, até finalmente você descobrir que o jogo se torna praticamente impossível...

Verificará então que os diques quase não existem, e terá de recorrer a alguns truques. Você pode saltar para um tronco que se encontre imediatamente acima ou abaixo. De início será difícil fazê-lo, mas não impossível. Não se esqueça de que você não se desloca juntamente com o tronco; estará apenas a usá-lo como um pedaço de dique. Se tiver cuidado, pode até deslocar-se para a direita e para a esquerda usando os troncos.

O final do jogo é a parte mais difícil, sendo esta a razão principal para tentar obter a maior pontuação possível antes de o dique ser destruído.

O sistema de pontuação é simples. Obtêm-se 100 pontos por ir do cofre até à caixa-forte, e outros 100 por conseguir trazer um saco no sentido contrário. Se perder uma vida enquanto transporta o saco, não pontua. A maior pontuação de Neil é de cerca de 3300, o que lhe dá já uma ideia do que deve tentar atingir.

O programa está escrito essencialmente em Basic, mas utiliza uma rotina em código-máquina localizada no endereço 32530 que desloca os troncos da direita para a esquerda do visor, parando-os depois de colidirem com a parede de um dique. Os dados correspondentes a este código-máquina encontram-se nas duas últimas linhas do programa.

Vejamos agora a estrutura do programa:

- 100 — Subrotina que imprime a pontuação
- 1000/1030 — Preparação do jogo
- 1040/1045 — Inicialização de cada jogo
- 1050/1100 — Inicialização de cada nível do jogo
- 1510/1560 — Ciclo principal
- 1600/1630 — Detecta colisões
- 1700/1740 — Final de cada nível
- 1800/1820 — Rotina de perda de um homem
- 2000/2100 — Rotina de jogo terminado
- 6000/6650 — Desenha os diques
- 7000/7100 — Desenha o rio e as margens
- 8000/8100 — Página de título e instruções
- 9000/9120 — POKE caracteres gráficos
- 9200/9310 — Poke código-máquina

Vejamos agora a listagem:

```

10 CLEAR 32529: RANDOMIZE : GO
TO 1000
99 REM pontuacao
100 PRINT AT 9,11-LEN STR# 9: I
NK 7; PAPER 2; BRIGHT 1;S: RETUR
N

```

```

11000 REM
11010 GOTO 2: PAPER 4: CLS : GO
11020 GOTO 1000
11030 PRINT AT 9,9: INK 7; PAPER
9: BRIGHT 1; BRIGHT 1; "000000" OR
BRIGHT 9: VIBR8: BRIGHT 1: "0"
00000"
11040 LET H5=9
11050 GO SUB 8000: PRINT AT 9,9:
PAPER 2; INK 7; BRIGHT 1; "000000"
: AT 9,10: "0"
11060 LET H5=9: LET MEN=9: LET S=
9: LET RN=.999
11070 LET SK=9: LET L=2: LET C=23
: LET G=7
11100 GO SUB 7000: GO SUB 6000
11500 REM CICLO
11510 LET L1=L: LET C1=C
11520 LET D#=INKEY#: IF D#="" THE
N GO TO 11000
11530 LET L=L+(D#="N")-(D#="0")
11540 LET C=C+(D#="D")-(D#="H")
11550 LET AT=PTR (L,C): IF AT<>2
& THEN GO TO 11000
11560 PRINT AT 11,10: "000000"
11570 PRINT AT L1,C1: PAPER 9: IN
K 9: AT L,C: INK 9: PAPER 9:
0: REM GRAFICO 0
11580 IF AND>N THEN PRINT AT 7+1
AT (AND*9),9: INK 9: PAPER 9:
: LET RN=RN-.9999
11590 LET Z=USR 30000: GO TO 11510
11600 IF AT=9 THEN IF G=7 THEN L
ET G=9: BRIGHT 1,9: LET S=S+100:
GO SUB 100: LET SK=SK+1: GO TO
11000
11610 IF AT=01 THEN IF G=9 THEN B
RIGHT 1,49: LET G=7: LET S=S+100:
GO SUB 100: IF SK=9 THEN GO TO
11700
11620 IF AT=45 THEN GO TO 1800
11630 IF AT=6 THEN GO TO 1949
11640 LET L=L1: LET C=C1: GO TO 1
1000
11700 FOR D=-10 TO 99 STEP 5: BE
P 1,1: BRIGHT 1,1-3: NEXT D
11710 FOR D=15 TO -10 STEP -5: BE
EP 1,1: BRIGHT 1,1-3: NEXT D: BE
EP 1,15
11720 FOR A=1 TO 21: PRINT AT A,9
: PAPER 2/3:
: NEXT A

```

```

1730 BEEP .1,20: BEEP .1,20: FOR
A=1 TO 21: PRINT AT A,0: PAPER
0:": NEXT A
1740 GO TO 1000
1800 FOR A=10 TO -25 STEP -4: PR
INT AT L,0: PAPER 0: INK 0: "0": A
T L,0: PAPER 0: INK 0:": BEEP
.01,0: NEXT A
1806 PRINT AT L1,01: PAPER 0: IN
K 0:":
1810 LET MM=MM-1: PRINT AT 0,1
0: PAPER 2: INK 7: BRIGHT 1:MM
1815 IF MM=0 THEN GO TO 2000
1817 IF SK=0 THEN GO TO 1700
1820 LET L=0: LET C=20: LET S=7:
GO TO 1000
2000 REM FINAL
2010 FOR A=0 TO 7: FOR B=1 TO 21
: PRINT AT 0,0: PAPER A:": NEXT
B: NEXT A
2030 FOR A=1 TO 10: BEEP .1,0: B
EEP .1,0: BEEP .1,10: NEXT A
2035 IF 0<IS THEN LET IS=0: PRIN
T AT 0,0-LEN STR# IS: PAPER 0:
INK 7: BRIGHT 1:IS: BEEP .1,0: B
EEP .0,10
2038 FOR A=20 TO 2 STEP -1: PRIN
T AT A,0: BRIGHT 1: PAPER AND#7:
INK 9: JOCO TERMINADO JOCO TE
AMINADO": NEXT A
2040 FOR A=0 TO 7: BORDER A: BEE
P .2,1: NEXT A: FOR A=7 TO 0 S
TEP -1: BORDER A: BEEP .2,7-A-1:
NEXT A
2100 GO TO 1040
2200 STOP
2300 REM PONTES
2310 IF MM=0 THEN GO TO 6000
2320 LET MM=0
2350 PAPER 0: INK 0: PRINT AT 7,
0:": AT 7,10:":
2360 PRINT AT 0,0:": AT 0,20:":
": AT 0,10:":
2370 PRINT AT 0,0:": AT 0,10
":": AT 10,0:": AT 10,10:
":": AT 10,0:": AT 10,10:":
": AT 12,14:":": AT 12,20:":
2380 PRINT AT 10,0:": AT 10,14
":": AT 10,21:":

```

```

6100 PRINT AT 14,0:":
T 14,10:":": AT 10,0:":":
AT 10,10:":":
6500 RETURN
6600 PAPER 0: INK 0: PRINT AT 7,
10:":": AT 7,20:":": AT 0,10:":
6610 PRINT AT 0,10:":": AT 0,10:":
":": AT 0,10:":": AT 0,20:":":
T 10,10:":": AT 10,10:":": AT 10,1
0:":": AT 10,20:":":
6620 PRINT AT 11,10:":": AT
11,20:":": AT 10,10:":": AT 10,1
0:":":
6630 PRINT AT 10,0:":": AT 10,1
0:":": AT 10,20:":": AT 14,0:": 00
":": AT 14,20:":":
6640 PRINT AT 10,0:":": AT 10,1
7:":":
6650 LET MM=1: RETURN
7000 REM IMAGEN
7010 FOR A=1 TO 21: PRINT AT A,0
: PAPER 4:": NEXT A
7015 FOR A=0 TO 10: PRINT AT A,0
: PAPER 0: INK 0:": NEXT A
7020 INK 0: PAPER 0: FOR A=1 TO
0: PRINT AT 0,0:": NEXT A
7030 PRINT AT 0,0:":": AT 0,10:":": AT 4,0
":": AT 4,10:":":
7040 PRINT AT 0,10:":": AT 0,10:":
": AT 0,21:":": AT 0,21:":
7050 PRINT AT 10,10:":": AT 10,10
":":
7060 PRINT AT 17,0:":":
AT 10,10:":": AT 10,20:":":
T 20,14:":":
7070 FOR A=10 TO 21: PRINT AT A,
0:": NEXT A
7080 FOR A=10 TO 21: PRINT AT A,
0: INK 0: "0": NEXT A: PRINT AT
A,0: INK 7: "0": REM GRAFICOS A
7100 RETURN
8000 REM TITULO
8005 BORDER 0
8010 FOR A=1 TO 21: PRINT AT A,0
: PAPER 0:": NEXT A
8012 DIM I(0): FOR A=1 TO 0: LET

```



```

1000 I(0)=0-1: NEXT A
0010 FOR A=1 TO 0: LET II=INT (A
NO*0)+1: LET AA=H(A): LET I(A)=I
(II): LET IIII=AA: NEXT A
0020 GO TO 0030
0010 FOR A=1 TO 0: INK I(A): GO
COMP I(A): PRINT AT 4,7:"AAAAAAAA
AAAAAAAA": REM GRAFICO A
0020 PRINT AT 6,7:"AAAAAAAAAAAAAAAA
AAAAA": REM GRAFICO A
0020 PRINT AT 9,7:"A": AT 9,24:"A
": BEEP .4, I(A)+10: REM GRAFICO
A
0024 IF INKEY#(0)="" THEN BORDER 2
: RETURN
0025 NEXT A: GO TO 0010
0030 PRINT AT 10,7: PAPER 0: INK
0: BRIGHT 1: "CORRIDA DO OURO "
0040 INK 0: PAPER 0: PRINT AT 12
,12:"COMANDOS": PLOT 0,75: ORAU
0050 PLOT 0,75: ORAU -00,0
0050 INK 1: PRINT AT 14,0:"0....
0060 SUBIR": AT 16,0:"Z.....0000
0060 PRINT AT 16,0:"1.....00000000
0070 AT 17,0:"0.....00000000"
0070 PAPER 0: INK 0: PLOT 0,24:
0080 0000,0
0080 PRINT AT 19,0: PAPER 4: INK
7: BRIGHT 1: "CARREGUE NUMA TELA
D.
0090 PRINT AT 21,0: PAPER 2: INK
7: "Escreito por Neil Pellinacci"
1000 GO TO 0010
0000 REM COO
0000 REM COO
0010 REM COO 0100
0020 FOR A=0 TO 47: READ B: POKE
: READ C: POKE A, B: NEXT A
0100 DATA 0,120,00,24,00,100,200
,120
0110 DATA 200,120,100,100,107,10
0,200,100
0120 DATA 16,00,16,124,16,40,00,
00
0000 REM MOODS
0210 FOR A=0 TO 47: READ B: POKE
0220 A+B, B: NEXT A
0220 RETURN
0000 DATA 00,100,00,0,11,14,01,0
04,120,204,0,30,15,04,40,40,100,0
04,24

```

```

0310 DATA 32,4,04,45,24,2,04,0,3
05,10,02,202,30,10,227,30,100,00,
17,32,0,0,11,04,45,20,10,201,201

```

TARÂNTULA

No grande programa de Malcolm Young que se segue, Tarântula, você encontra-se num poço escuro com as dimensões aproximadas de um estádio de futebol, lutando contra temíveis aranhas.

As tarântulas deslizam pelos lados do poço procurando comida, e na opinião delas os seres humanos entram na categoria «comida»... Você está equipado com uma lanterna e um laser, e desloca-se pelo poço num tanque pouco seguro.

O programa contém todas as instruções necessárias, e você verificará que depressa descobre o modo de jogar depois de iniciar a execução do programa.

```

5 DIM P#(15): DIM W#(0,15): 0
IM A(0)
0 PRINT "Por favor descrete um
pouco.": GO SUB 0010: REM defini
caracteres
40 GO SUB 0000
00 BORDER 0: PAPER 0: INK 7: C
L0
00 PRINT AT y,x:t#
00 PRINT INK 0: PAPER 0: AT 21,
0:"PONT.=";score:AT 21,10:"ARRANH
00=";100;" VIDAS=";vitas
40 LET a=x: LET b=y
00 LET i#=INKEY#
00 GO TO (i#<"1" OR i#="0")*200
+70
70 BEEP .05,0: GO SUB VAL I#*1
000
00 PRINT AT y,a:" "; BRIGHT 1:
AT y,x: INK P#I#
00 GO SUB (RND<lev)*100+100
100 GO TO (d#(y+1,x+1)="0")*000
+110: REM grafico A
110 LET cn=cn*(cn<0)+1
120 GO TO (n>0)*05+40
120 IF NOT s(cn,1) THEN GO TO (
n>1)*70+40
130 LET sx=s(cn,1): LET sy=s(cn
,2)

```

```

140 LET d$(sy,ex)=" "
150 LET sy=ac+(sy<y+1)-(sy>y+1)
: LET s(cn,2)=sy
160 LET ex=ex+(ex<x+1)-(ex>x+1)
: LET s(cn,1)=ex
170 LET d$(sy,ex)="A": REM graf
180 A
190 GO TO 40
200 REM nova tarantula
210 IF n=15 THEN LET lev=.05: R
RETURN
220 LET n=n+1: LET nb=nb+1
230 LET sx=s(cn,1): LET sy=s(cn,2)
240 LET d$(sy,ex)="A": REM graf
250 A
250 LET c=INT (RAND*5)+2
260 PRINT BRIGHT 1: INK C: AT sy
-1, sx-1: "A": CHR# 0: BEEP .4, 1:
PRINT "A": REM graf: GO TO 270
270 PRINT INK 0: PAPER 5: AT 21,
20: "PONT.=": score: AT 21, 12: "ARINH
20: "nb: " VIDAS=": lives
280 RETURN
290 PRINT PAPER 7: BRIGHT 1: FL
PAPER 1: INK 2: AT 10, 5: "Argh!!! D
290 ARINH-NO.": LET lives=lives-1
310 FOR t=1 TO n: IF s(t,1)=x+1
AND s(t,2)=y+1 THEN GO TO 300
320 NEXT t: STOP
330 LET d$(s(t,2), s(t,1))=" "
340 LET s(t,1)=0: LET nb=nb-1
350 FOR t=1 TO 10: PRINT INK 0:
AT y, x: "A": CHR# 0: BEEP .25, 1:
PRINT "A": BEEP .25, 0: NEXT t
360 PAPER 0: INK 7: FLASH 0: CL
370
380 IF n=15 AND nb=0 THEN GO TO 60
390
400 PRINT INK 0: PAPER 5: AT 21,
20: "PONT.=": score: AT 21, 12: "ARINH
40: "nb: " VIDAS=": lives
410 LET rk=rk+(rk<7)
420 IF lives THEN RETURN
430 IF score<a(5) THEN GO TO 54
440
450 INPUT PAPER 5: INK 1: AT 0, 0
: "Voce ficou nos "; FLASH 1: "cin
co primeiros!"; FLASH 0: "Indica
o o nome: "; LINE p#
460 FOR r=5 TO 1 STEP -1: BEEP
.1, r: IF score>a(r) THEN NEXT r

```

```

520 FOR g=5 TO 7+2 STEP -1: BEE
p .1, g: LET a(g)=a(g-1): LET #$(
g)=#$(g-1): NEXT g
530 LET a(g)=score: LET #$(g)=#
#
540 BORDER 3: PAPER 5: INK 2: C
L 5
550 PRINT AT 3, 2: "Campeoes de 5
0 Lp:"
560 FOR r=1 TO 5: BEEP .5, r
570 PRINT AT 7+r, 4: FLASH (#$(r
)=p# AND score=a(r)): #$(r): a(r):
NEXT r
580 PRINT AT 10, 3: FLASH 1: INK
0: "Over Jogar de novo?": PAUSE
0
590 IF INKEY#="n" THEN STOP
600 GO TO 10
700 REM promover
710 BORDER 2: PAPER 6: INK 1: C
L 5
720 PRINT AT 3, 11: "PARABENS !!!
TAB 1: "Voce #stou "; score/10: "
tarantulas"
730 PRINT TAB 5: "Voce e: porta
rio"
740 LET rk=rk-1
750 PRINT TAB 3: "promovido ao
graf "; rk
760 RESTORE 3550: FOR #=1 TO 25
: READ n: BEEP .2, n: NEXT #
770 PRINT "Voce suda agora de
tenque deuido"
780 PRINT "a sua nova posicao"
790 PRINT "Comandara agora o
tenque"
800 FOR c=6 TO rk STEP -1: READ
c#: NEXT c
810 PRINT BRIGHT 1: PAPER rk: I
NK 0: c#
820 CHM d$(22, 32): LET s=7: LET
r=s/3: LET nb=0
830 GO SUB 9250: PRINT AT 20, 0:
"Carregue numa tecla.": PAUSE 50
840
850 BORDER 5: PAPER 0: INK rk:
CL 5
860 LET p=0: RETURN
1000 LET UD=(T#="E")-(T#="D"): R
EM graficos E, D
1010 LET HD=(T#="C")-(T#="F"): R
EM graficos C, F

```

```

1020 LET V=Y+1: LET H=X+1: LET W
1030 PAPER 5: INK 5
1040 GO TO 1050
1050 LET d=(R#50)
1060 LET d=#5+dt: LET d=d OR (d<1)
1070 FOR t=#5+vd TO d STEP vd
1080 LET u=#5-#5 OR V-#5<1
1090 LET l=((V+#5)-(V+#5>21))*
(V+#5-21)
1100 PRINT AT U-1,t-1,d#(U,t);
CHR# 8): IF U>1 THEN PRINT OVER
1:"U" AND vd=1:"I" AND vd=-1: REM
graficos U, I
1110 FOR #=ul+1 TO ll-1: PRINT F
[OSH d#(w,t)<>" "]:AT w-1,t-1,d
#(w,t): NEXT #
1120 PRINT AT ll-1,t-1,d#(ll,t);
CHR# 8): IF ll<21 THEN PRINT OVER
R1:"I" AND vd=1:"G" AND vd=-1:
REM graficos H, G
1130 LET #5=#5+1: NEXT t
1140 GO TO 1030
1150 LET dt=(R#vd)
1160 LET d=#v+dt: LET d=d OR (d<1)
1170 FOR t=#v+vd TO d STEP vd
1180 LET ll=(#5-#5) OR (#5-#5<1)
1190 LET u=#5+#5-(#5+#5>21)*
(#5+#5>21)
1200 PRINT AT t-1,(l-1)d#(t,(l
O U)
1210 IF ll>1 THEN PRINT OVER 1:R
T t-1,(l-1):"U" AND vd=1:"H" AND
vd=-1: REM graficos U, H
1220 IF U<20 THEN PRINT OVER 1:
AT t-1,(l-1):"H" AND vd=1:"G" AND
vd=-1: REM graficos H, G
1230 LET #5=#5+1
1240 NEXT t
1250 BEEP .2,5: PAPER 5: INK 7:
CLS: PRINT AT y,x)t#
1260 PRINT INK 5: PAPER 5: AT 21,
5:"PONT.=";score:AT 21,10:"ARRAN
R#=";R#;" VIDAS=";lives
1270 LET #=#-RND/3: RETURN
1280 LET UD=(T#="E")-(T#="D"): L
ET HD=(T#="O")-(T#="F"): REM gra
ficos E, D, O, F
1290 LET U=Y+1: LET H=X+1
1300 FOR T=1 TO R

```

```

2005 LET U=U+UD: LET H=H+HD
2010 LET U=U+(U<1)-(U>21): LET H
=H+(H<1)-(H>21)
2020 BEEP .5,10
2030 PRINT AT U-1,H-1:"K": REM g
raficos K
2040 GO TO (D#(U,H)<>" ")#20+255
5
2050 PRINT INK 5:AT U-1,H-1:"■":
NEXT T
2070 LET r=(r>2)/10: RETURN
2080 FOR T=5 TO 7: BEEP .1,t: P
INT INK t:AT U-1,H-1:"R":CHR# 8):
: OR BEEP .1,t+7: PRINT INK 7-t:"R"
:CHR# 8): BEEP .1,t-7: NEXT T: R
EM graficos R, B
2090 LET score=score+10: LET #=#
+RND/4: LET r=INT (5/1.5)+1
2100 FOR t=1 TO #: IF V#=(t,R) R
ND #=(t,1) THEN GO TO 2120
2110 NEXT T: STOP
2120 LET s(t,1)=0: LET #=#-1
2130 LET d#(V,H)="
2140 PRINT INK 5: PAPER 5: AT 21,
5:"PONT.=";score:AT 21,10:"ARRAN
R#=";R#;" VIDAS=";lives
2150 LET lev=lev+5.5*(lev<.2)
2160 LET #=#+1: IF #=15 THEN GO
TO 700
2170 IF #=15 AND #=# THEN PRINT
BRIGHT 1: PAPER 4: INK 5:AT 15,
5:"Por favor espere 2 segundos p
ara proxima onda.": GO SUB 9250
2180 RETURN
2190 LET T#="F": REM graficos F
2200 LET X=X-(X>0)
2210 RETURN
2220 LET T#="E": REM graficos E
2230 LET Y=Y+(Y<20)
2240 RETURN
2250 LET T#="D": REM graficos D
2260 LET Y=Y-(Y>0)
2270 RETURN
2280 LET T#="O": REM graficos O
2290 LET X=X+(X<31)
2300 RETURN
2310 REM DEFINIR CARACTERES
2320 DATA 55,95,125,55,125,155,5
5,55
2330 DATA 125,155,125,55,125,155
,155,125
2340 DATA 5,125,55,55,55,125,
5

```


A SELVA

Trata-se de um programa bastante difícil da autoria de Andrew Sweetland e Martin Jones. Na primeira fase, você deve guiar a sua canoa por um rio de floresta. Encontrará vários rochedos pelo caminho, e deve obviamente evitá-los. Se chocar com um deles, será ingerido por um peixe comedor de homens.

Na segunda fase, encontrar-se-á numa desagradável estrada como a indicada em seguida:



O seu carro não tem travões, e necessitará de toda a sua habilidade para conseguir evitar as árvores.

Use as seguintes teclas para controlar o barco e o automóvel:

- Q — Subir
- A — Descer
- O — Direita
- P — Esquerda

1 REM *** A SELVA ***

©Martin Jones & Andrew Sweetland

```

00 REM EPISODIO 1
01 GO SUB 100001: LET U=0
02 FOR V=1 TO 3: BORDER 0: BRT
03 FOR W=1 TO 3: BRT 7: CLS: BRT
04 FOR Y=14 TO 21: PRINT AT Y,
05 BORDER 1: NEXT Y
06 PRINT #1, AT 0,0: PAPER 1,,,
07 PRINT AT 0,0: BRIGHT 1: INK
08
09 FOR Y=10 TO 10: PRINT BRIGHT
10 AT Y,0: PAPER 0: NEXT Y
11 PRINT AT 1,00: BRIGHT 1: IN
12 PRINT AT 2,00: REM 97000
13
14 FOR Y=10 TO 10: LET G=6+(CIN
15 *100): PRINT AT Y,G: PAPER 1: IN
16 PRINT AT Y,G+10: NEXT Y: PAPER
17 PRINT AT 00,00: INK 0: PAPER
18 REM 97000
19 LET X=20: LET Y=0
20 IF X<20 THEN AT X,Y
21 OR AT X,Y+1)=14 OR AT X,
22 X,Y+2)=14 THEN GO SUB 100
23 PRINT AT X,Y: INK 7: PAPER
24 REM 97000
25 LET Y=Y+1
26 IF INKEY#="q" THEN IF (X)>10
27 AND Y<20 THEN IF AT X-1,Y)<
28 >14 AND AT X-1,Y+1)<>14 AND A
29 TTR (X-1,Y+2)<>14 THEN PRINT AT
30 X,Y: PAPER 1: INK 7: LET X
31 =X-1
32 IF INKEY#="o" THEN IF (X)<20
33 AND Y<20 THEN IF AT X+1,Y)<
34 >14 THEN IF AT X+1,Y+1)<>14 T
35 HEN IF AT X+1,Y+2)<>14 THEN P
36 RINT AT X,Y: PAPER 1: INK 7:
37 LET X=X+1
38 IF INKEY#="o" AND Y<20 THEN
39 LET Y=Y-1
40 OUT 254,0: OUT 254,10
41 FOR Z=1 TO 10: NEXT Z: IF Y
42 =30 THEN NEXT 1: GO TO 100
43 GO TO 55
44 REM FIM DO EPISODIO 1
45 PRINT AT X,Y: PAPER 1: INK
46 7: REM 97000: FOR I=50 TO 50: FOR G=0
47 TO 32: OUT 254,G+16: OUT 254,0:

```

```

NEXT 9: NEXT 7: REM graphics B8
110 GO TO 999
120 REM 00000001
130 PRINT AT X,4: PAPER 1: INK
7: " " FOR I=0 TO 99: OUT 254,1
9: OUT 254,9: NEXT 7: PRINT AT X
+1,4+1: PAPER 1: INK 4: " " AT X
-1,4+1: INK 7: " " : LET I=I+1:
IF I=9 THEN GO TO 9999: REM 9
graphics CK,CC,LM
140 FOR I=1 TO 500: NEXT 7: GO
TO 9
9999 REM 99999999 II
9999 CLR : PAPER 1: INK 4: CLR :
FOR I=0 TO 91: PRINT " " : NEXT I
9999 REM 99999999 M
910 FOR I=0 TO 99: GOTO .91,1
: NEXT 7
910 FOR I=1 TO 9: PRINT PAPER 4
: INK 1: AT I,9: "
: NEXT 7
920 FOR I=0 TO 9: PRINT AT I,10
: PAPER 4: INK 1: " AT I,10: "
: AT I,10: " : NEXT 7
930 FOR I=7 TO 10: PRINT AT I,2
9: PAPER 4: INK 4: " : AT I,10: "
: NEXT 7
930 FOR I=3 TO 4: PRINT AT I,10
: PAPER 4: INK 1: " : NEXT
7
930 FOR I=0 TO 10: PRINT AT I,0
: PAPER 4: INK 1: "
: NEXT 7
940 FOR I=10 TO 10: PRINT AT I,
1: PAPER 4: INK 1: " : NEXT 7
940 FOR I=15 TO 17: PRINT AT I,
11: PAPER 4: INK 1: " :
AT I,10: " : NEXT 7
950 FOR I=10 TO 20: PRINT AT I,
9: PAPER 4: INK 1: " : AT
I,10: " : NEXT 7
960 FOR I=13 TO 18: PRINT AT I,
10: PAPER 4: INK 1: " : AT I,17
: NEXT 7
960 PRINT PAPER 4: INK 1: AT 0,2
9: " AT 0,10: " : AT 0,10: "
: AT 0,10: " : AT 0,10: "
: AT 0,10: " : AT 0,10: "
: AT 11,10: " : AT 10,11: " : AT
10,10: " : AT 10,10: "

```

```

960 PRINT AT 10,10: PAPER 4: INK 1: AT 14,
10: " : AT 14,10: " : AT 14,10: "
: AT 16,10: " : AT 16,10: " : AT
16,10: " : AT 17,10: " : AT 17,10: "
: AT 18,10: " : AT 18,10: " : AT
18,10: " : AT 19,10: " : AT 19,10: "
: AT 19,10: " : AT 19,10: " : AT
19,10: " : AT 19,10: "
970 LET X#=" " : LET X=2: LET Y=0
: LET Y#="0"
980 IF X=(10 AND Y=01) OR (X=10
AND Y=01) OR (X=09 AND Y=01) TH
EN GO TO 7666
990 IF ATTR (X,Y)=12 THEN GO TO
9999
9999 PRINT AT X,4: PAPER 4: INK
9: "
9999 OUT 254,9: OUT 254,16
9999 LET I=INKEY#="0" THEN LET X#="
0" : LET Y#=" " : REM 99999999 0
9999 LET I=INKEY#="0" THEN LET X#="
0" : LET Y#=" " : REM 99999999 0
9999 LET I=INKEY#="0" THEN LET X#="
0" : LET Y#=" " : REM 99999999 0
9999 LET I=INKEY#="0" THEN LET X#="
0" : LET Y#=" " : REM 99999999 0
9999 PRINT AT X,4: PAPER 4: INK
9: " : REM 99999999 0
9999 IF X#="0" THEN LET Y=Y+1
9999 IF X#="0" THEN LET Y=Y+1
9999 IF X#="0" THEN LET Y=Y+1
9999 IF X#="0" THEN LET Y=Y+1
9999 GO TO 9999
9999 REM 0000000001
010 PRINT AT X,4: INK 2: FLASH
1: " : FLASH 0: AT X-1,4: PAPER 4
: INK 9: " " : FOR I=0 TO 99: OUT
254,10: OUT 254,0: NEXT 7: LET
I=I+1: IF I=9 THEN FOR I=0 TO
999: NEXT 7: GO TO 9999
9999 IF I=9 THEN GO TO 9999
7666 REM exito!
7610 PAPER 9: PAPER 9: INK 7: 0
10: PRINT AT 10,11: "PARABENS !"
: AT 12,2: "Voa consecutiv abria c
mainho" : AT 13,10: "oia SELVA !"
7600 FOR I=0 TO 99: GOTO .91,1:
GOTO .91,1: NEXT 7
7600 GO TO 9999
7640 STOP
9999 REM fra casso
9910 RESTORE : BORDER 9: PAPER 9

```

```

: INK 7: CLS
0000 PRINT AT 10,1: "Há 7 tabuletas
: Dependendo da posição das pedras
: as pedras mudam de lugar."
0005 FOR Y=1 TO 10: READ a,b: GO
: TO NEXT Y
0010 PRINT "1,0,1,0,0,1,0,0,0,0,0,0,0,
: 0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0
: 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
0015 PRINT AT 10,0: "Cada 3a coluna
: muda de lugar."
0020 IF INKEY#="R" THEN LET L1=3
0030 IF INKEY#="D" THEN PRINT US
0040 GO TO 0005
0050 STOP
0060 DATA
0070 DATA
0080 DATA
0090 DATA
0100 DATA
0110 DATA
0120 DATA
0130 DATA
0140 DATA
0150 DATA
0160 DATA
0170 DATA
0180 DATA
0190 DATA
0200 DATA
0210 DATA
0220 DATA
0230 DATA
0240 DATA
0250 DATA
0260 DATA
0270 DATA
0280 DATA
0290 DATA
0300 DATA
0310 DATA
0320 DATA
0330 DATA
0340 DATA
0350 DATA
0360 DATA
0370 DATA
0380 DATA
0390 DATA
0400 DATA
0410 DATA
0420 DATA
0430 DATA
0440 DATA
0450 DATA
0460 DATA
0470 DATA
0480 DATA
0490 DATA
0500 DATA
0510 DATA
0520 DATA
0530 DATA
0540 DATA
0550 DATA
0560 DATA
0570 DATA
0580 DATA
0590 DATA
0600 DATA
0610 DATA
0620 DATA
0630 DATA
0640 DATA
0650 DATA
0660 DATA
0670 DATA
0680 DATA
0690 DATA
0700 DATA
0710 DATA
0720 DATA
0730 DATA
0740 DATA
0750 DATA
0760 DATA
0770 DATA
0780 DATA
0790 DATA
0800 DATA
0810 DATA
0820 DATA
0830 DATA
0840 DATA
0850 DATA
0860 DATA
0870 DATA
0880 DATA
0890 DATA
0900 DATA
0910 DATA
0920 DATA
0930 DATA
0940 DATA
0950 DATA
0960 DATA
0970 DATA
0980 DATA
0990 DATA
: END

```

LÓGICA

Neste rápido jogo de lógica e habilidade, você encontra-se extremamente bem disfarçado sob a forma de um asterisco... Terá de evitar os Horríveis Pontos Negros que surgem aleatoriamente a sua volta.

Como verificará ao executar o programa, terá de indicar inicialmente o nível de dificuldade. Este pode variar entre 1 e 100, correspondendo os números menores aos jogos mais fáceis. O nível 100 é apenas para peritos...

Mantenha-se em movimento contínuo a fim de não ser apanhado pelos Pontos Negros. Use as teclas de cursor para se deslocar.

```

20 INPUT "Movimentos - teclas
de cursor, indique dificuldade
(1/100):" D
30 BORDER 4: LET X=10: LET Y=1
5: LET S=10: LET A=53
40 PRINT "  : POKER 20002,055
50 IF SCREEN$(X,Y)="#" THEN G
O TO 100
60 PRINT AT X,Y: " # ": LET S=S+1
70 PRINT AT X+INT (RND*3)-1,Y+
INT (RND*3)-1: INVERSE 1: " # "
80 PRINT P
90 IF CODE INKEY# < 53 OR CODE I
NKEY# > 53 THEN GO TO 40
100 LET S=CODE INKEY#
110 IF A=53 AND Y > 0 THEN LET Y=
Y-1
120 IF A=54 AND Y > 20 THEN LET X
=X+1
130 IF A=55 AND Y > 0 THEN LET X=
X-1
140 IF A=56 AND Y < 20 THEN LET Y
=Y+1
150 GO TO 40
160 PRINT " [X] Voce pontuou
: R: PAUSE 500: INPUT "Correque
6# ENTER : S#: CLS : RUN

```

NA HORTA

A mente desorganizada de Neal Cavalier-Smith concebeu este estranho jogo cujo objectivo é plantar couves. Você recebe

um ponto por cada couve plantada, e perde 10 se tentar plantar uma couve em cima de uma casa. Perderá ainda três pontos se as plantar perto de uma casa ou debaixo de uma árvore (não se esqueça, leitor, de que eu disse que era estranho...).

```

3 LET S=0
4 PAPER 0: INK 7: BORDER 0
5 PRINT "
6
7
8
9 PRINT "Você ganha 1 ponto p
10 de cada cou-ve que consiga plant
11 ar, mas per-de 10 se plantar uma
12 sobre um poste (não crescerá"
13 si), e 0 se se plantar junto
14 a uma casa ou sob uma árvore."
15
16 LET B#="X": LET S=0: LE
17 T S=0
18 DIM B$(12,20)
19 LET C#=""
20 FOR X=1 TO 12
21 LET C#="C#+"
22
23 LET B$(X)=C#+B#
24 NEXT X
25 LET Y=13
26 LET X=(AND*12+1)
27 PRINT AT 17,13: FLASH 1:"IN
28 COIO"
29 PRINT AT 21,0: PRINT INK S;
30 B$(X)
31 LET X=X+(AND*2-1)
32 LET S=S+1
33 IF X>12 THEN LET X=X-1
34 IF X<1 THEN LET X=X+1
35 IF INT X=6 AND Y=12 OR Y=10
36 THEN PRINT AT 20,4: INK 2:"
37 PRINT AT 21,4: INK 7:"
38 LET
39 T S=0: Y=12
40 IF INT X=6 AND Y=14 THEN PR
41 INT AT 20,20: INK 4:"
42 PRINT
43 AT 21,20: INK 4:"
44 LET S=S-3
45 IF INKEY#="5" THEN LET Y=Y-
46 1
47 IF INKEY#="8" THEN LET Y=Y+
48 1
49 IF SCREEN$(11,Y)(">") THEN
50 GO TO 220
51 PRINT AT 11,Y: INK 4:">"

```

```

100 POKE 23692,255
200 GO TO 110
220 BEEP 1,1: LET h=h+1: LET s=
s-10: IF h=3 THEN PRINT "Pontuac
20="; s: FOR X=1 TO 10: LET z=TAN
X: NEXT X: PAUSE 50: RUN
230 GO TO 110

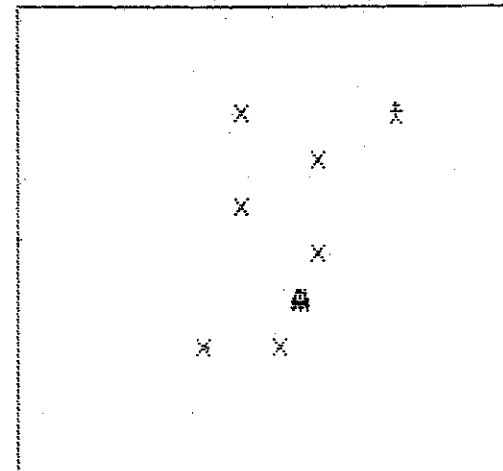
```

FANTASMAS

O fantasma persegue-o. A pequena zona de jogo apresenta algumas minas (X), e você deve tentar levar o fantasma a entrar em contacto com elas. Se tal acontecer, o fantasma será electrocutado...

As minas também constituem um perigo para si, pelo que deve evitá-las ao mesmo tempo que foge do fantasma.

Escrito por Neal Cavalier-Smith e Graham White, este programa mostra como é possível dar uma boa dose de inteligência a um fantasma.



Não conseguirá certamente fugir-lhe durante muito tempo... O programa é de facto muito rápido. Se for demasiado rápido para si das primeiras vezes que jogar, inclua alguns ciclos FOR/NEXT no programa a fim de o tornar mais lento.


```

1 GO SUB 600
2 GO SUB 600
3 LET a=0: LET b=0: LET p=0:
LET h=0
5 BORDER 0: PAPER 0: INK 7: C
L6
10 LET a=INT (RND*10)+2
15 PLOT 00,0: DRAW 200,0: DRAW
0,100: DRAW -200,0: DRAW 0,-100
20 LET b=INT (RND*10)+10
30 LET x=2*(INT (RND*0))+0
40 LET y=2*(INT (RND*10))+10
43 LET e=2*(INT (RND*0))+0: LE
T f=2*(INT (RND*0))+10
44 IF e=x OR e=b OR f=y OR f=6
THEN GO TO 43
45 PRINT AT e,f: PAPER 2: "X"
46 IF a=p THEN GO TO 49
47 LET q=p: FOR z=0 TO 7-p: GO
TO 43: NEXT z
48 IF y>20 THEN LET y=20
50 PRINT AT x,y: INK 5: CHR# 15
6: LET v=x: LET w=y
65 REM CHR# 150 e' US 049
66 PRINT AT a,b: INK 6: CHR# 14
4: LET c=a: LET d=b
68 REM CHR# 144 e' US 049
70 LET a=a-(x<a)+(x>a)
80 LET b=b-(y<b)+(y>b)
90 LET x=x+0*(INKEY#="5")-2*(I
NKEY#="7")
95 LET x=x+2*(x<1)-2*(x>20)
100 LET y=y+2*(INKEY#="0")-2*(I
NKEY#="9")
105 LET y=y+2*(y<5)-2*(y>20)
110 PRINT AT v,w: " " AT c,d: " "
115 IF a=x AND b=y THEN GO TO 4
60
120 IF SCREEN# (a,b)="X" THEN 0
0 TO 200
130 IF SCREEN# (x,y)="X" THEN 0
0 TO 300
140 GO TO 49
200 PRINT AT a,b: FLASH 1: CHR#
145
205 REM CHR# 145 e' US 049
210 FOR d=2 TO 40 STEP 4
211 BEEP d/600,-15
212 NEXT d
215 PRINT AT a,b: " "
216 LET s=s+10
217 PRINT AT 10,0: PAPER 4: FLA
SH 1: "pt."

```

```

218 PRINT AT 12,0: FLASH 1: PAP
ER 0: s
220 GO TO 10
300 PRINT AT x,y: FLASH 1: "X":
REM grafico B
310 FOR d=20 TO 1 STEP -1
311 BEEP d/600,25
312 NEXT d
315 PRINT AT x,y: " "
316 LET p=p-1: IF p=0 THEN GO T
O 500
320 GO TO 10
400 PRINT AT x,y: FLASH 1: "X":
REM grafico B
410 FOR d=1 TO 10
411 BEEP .01,INT (RND*60)-25
412 NEXT d
415 PRINT AT x,y: " ": PRINT AT
e,f: " "
416 LET p=p-1: IF p=0 THEN GO T
O 500
420 GO TO 10
500 PAPER 1: FLASH 1: CLS
510 PRINT AT 10,7: "S E M" U I
0 0 0
515 FOR n=40 TO -20 STEP -1
516 BEEP .01,n
517 NEXT n
520 FOR d=1 TO 200: NEXT d
530 FLASH 0: CLS
535 PAPER 0: CLS
540 IF h<0 THEN LET h=h
550 PRINT AT 5,10: PAPER 0: "MAX
IMO "h$ AT 10,0: PAPER 0: INK 0
: "0 e' regue nume tecla."
560 LET a#=INKEY#: IF a#="" THE
N GO TO 560
565 LET v=0: LET s=0
570 GO TO 5
580 FOR n=0 TO 7
590 READ d: POKE USR "a"+n,d
600 NEXT n
630 DATA 25,62,42,107,127,127,1
00,70
640 FOR n=0 TO 7
650 READ d: POKE USR "b"+n,d
660 NEXT n
670 DATA 145,82,0,195,0,74,137,
0
680 FOR n=0 TO 7
690 READ d: POKE USR "m"+n,d
700 NEXT n

```

```

710 DATA 0,20,0,62,0,0,20,04
720 RETURN
800 BORDER 0: PAPER 0: INK 7: C
L
805 PRINT TAB 5; INK 6; FLASH 1
;" P R I N T A S M I S M I "
810 PRINT " " "Voce deve mover o
seu bomeem ("; INK 5; CHR# 155); I
NK 7; " pelo visor ate obrigar o
ban- tasma ("; INK 6; CHR# 144);
INK 7; " que o perseguu, e cair
sobre as bombas ("; CHR# 15; "X");
PAPER 0; " "Mas tenha cuidado,
neo caia voce nelas! Tera 5
vidas. " "Boa sort
te....."
820 PRINT "Movimentos: teclas
830 PRINT TAB 5; FLASH 1;"Carr
eque numa tecla."
840 LET a#:=INKEY#: IF INKEY#=""
THEN GO TO 840
850 RETURN

```

SAPO

O Sapo é outro grande programa de Malcolm Young. O autor afirma que teve grande prazer na criação deste programa porque pôde explorar nele algumas técnicas de programação novas. «Por exemplo», escreveu, «tenho experimentado rotinas em código-máquina e produzi uma rotina de «scroll» para a esquerda completamente minha... O computador não pode normalmente reconhecer os gráficos definidos pelo utilizador usando SCREEN\$, e a maior parte das pessoas tem resolvido este problema recorrendo à função ATTR.

«Verifiquei que fazendo a variável de sistema CHAR\$ apontar para a memória de UDG, a função SCREEN\$ pode actuar do modo habitual, se bem que seja necessário usar as letras correspondentes aos caracteres gráficos. Usei esta técnica no programa entre as linhas 300 e 490. Esta técnica tem a vantagem de reconhecer quaisquer caracteres UDG, incluindo os negros sem FLASH ou BRIGHT.

«Construí o programa de tal modo que pudesse ser introduzido nas máquinas de 16 K sem quaisquer modificações.

O objectivo do programa de Malcolm consiste em apanhar tantos insectos quanto possível usando as teclas I e P para

esquerda e direita, e CAPS SHIFT para saltar. Deve-se evitar sair da imagem ou cair dos troncos ou chocar com a aranha, o que resultará na perda de uma vida. Cada jogo começa com 200 unidades de energia, diminuindo lentamente até zero. Quando atinge zero, o sapo morre.

Podem aparecer borboletas, e se forem apanhadas valem 10 pontos de energia e 50 pontos de bônus. O jogador possui apenas de uma probabilidade de 50% de apanhar cada borboleta, por muito rigoroso que seja o salto.

O leitor verificará que este jogo é bastante aditivo, e pode modificá-lo de modo a incluir uma opção de níveis de dificuldade, alterando as dimensões e o número dos troncos. Depois de introduzir o programa, grave-o com a instrução SAVE «sapo» LINE 5000.

```

10 REM SAPO
20 DEF FN r(x)=INT (RND*x)+1
30 CLEAR USR "a"-43
40 GO SUB 0000: REM uds
45 GO SUB 0110
50 GO SUB 0000: REM imagem
55 GO TO 560
70 POKE 23505,0: POKE 23507,50
80 PRINT AT 0,0: BRIGHT 1; INK
7; PAPER 2;"PONT.: "TAB 11;
"ENERGIA:";en;TAB 24;"SAPOS: " ;f
FOR
80 RETURN
100 REM ler teclado
110 LET k=PEEK 23555
120 LET x=x+(k=80)-(k=78)
130 IF x<1 OR x>31 THEN GO TO 3
30
140 LET k=IN 65278
145 GO TO (k=254 OR JP)*ds+150
150 PRINT AT y,x-1;" 0 " : REM g
grafico 0
160 IF NOT POINT (x+8,31) THEN
GO TO 450
165 RETURN
170 REM salto
180 LET jp=1: LET bp=10: LET ds
=40: PRINT AT y,x-1;" "
190 POKE 23505,lp: POKE 23507,h
h
200 FOR l=0 TO 1
210 LET y=y-jp
215 GO TO (y=17)*180+220
220 IF SCREEN# (y,x)>"f" THEN G
O TO 300

```

```

230 PRINT AT y-1,x;"a" AND JP<0
)AT y,x;"c";AT y+1,x;"d";AT y+2,
x;"a" AND (JP>0 AND y<16): GOTO 2
.0000,00
240 NEXT I
250 LET BP=BP+JP
270 LET JP=JP+1
280 POKE 20000,N*(y<3)+(y>16)
290 RETURN
300 LET a#=SCREEN$(y,x)
310 IF a#="i" OR a#="j" THEN LET
T e=(y-1)*20+x-2: LET i#(e TO e+
2)="": GO TO 350
320 IF a#="g" OR a#="h" THEN LET
T b=0: LET e=e+10: LET s=s+2
0: GO TO 350
330 FOR z=20 TO 30: BEEP .05,-z
: BEEP .1,-z-.5: NEXT z
340 LET frog=frog-1: LET s=8: B
EEP 1,-10: IF NOT frog THEN GO T
O 1000
345 LET en=200: GO TO 6000
350 LET s=s+10: LET JP=-1
360 BEEP .5,10
370 GO SUB 70
380 RETURN
400 IF SCREEN$(16,x)<>"e" THEN
GO TO 450
410 GO SUB 6000
420 PRINT AT y-1,x;"c";AT y,x;"
d"
430 PRINT AT 16,x-1;"eee";AT 17
,x;"b"
440 RETURN
450 BEEP .05,-5
460 PRINT AT 16,x;"":AT 17,x;"
c";AT 18,x;"d": REM graficos 0,
0
470 BEEP .05,-5
480 PRINT AT 17,x;"":AT 18,x;"
c"; BEEP .1,-5: PRINT AT 18,x;"
": REM graficos 0
490 GO TO 300
500 REM rotacao principal
510 IF AND>.05 THEN GO TO 560
520 LET st=30*(FN r(6)+9)-1
540 LET i#(st TO st+1)="L": RE
M grafico IL
550 PRINT AT 1,0: INK 1:i#
570 LET i#=#(3 TO )
580 LET x=x-NOT JP: LET l=USR m

```

```

0000 GOTO 0000,000,10
0010 GOTO 000,100
0020 GOTO TO (NOT 0)*40+000
0030 LET a=FN r(4): LET b=-1
0040 LET c=c+(a+1)-(a-2): LET
b=b*(a+1)-(a-4)
0050 LET c=c-(b+15)+(b+1): L
ET b=b*(b+(b+9)-(b+9))
0060 PRINT AT b,c,b: INK 2: OVER
LAY "G": REM graficos 0,1
0070 IF NOT 5 THEN LET b=(AND>.0
)*FN r(20)+9: IF 5 THEN LET b
=FN r(15)+1: LET b=b*(b)
0080 REM graficos
0090 LET s=s+(AND>.4)*((s*x)-(s>
x)1)
700 PRINT INK 0:AT 16,0;a#:AT 1
6,a;"LH": REM graficos L,H
710 HT W>10 AND (x=6 OR x=s+1).
THEN GO TO 800
720 LET en=en-1
730 PRINT AT 0,10: BRIGHT 1: IN
K 7: PRINT AT 0,0;"
740 HT W<6 THEN PRINT PAPER 2:
AT 0,7: FLASH 1;"Materia de L"
750 GO SUB 3000
760 GO TO 510
0010 PRINT LEX 0:AT 16,0;"":AT
17,0;"LH": REM graficos L,H
0020 FOR N=0 TO 1 STEP .1
0030 BEEP N,-N*10: NEXT N
0040 GO SUB 3400
0050 GO TO 500
40000 REM tabela de pontuacao
40100 GO SUB 0000: GO SUB 70
40200 IF s<=5 THEN GO TO 1000
40300 INPUT "Te# e #author pontuac
ao de hoje! Indique nome: "X:LIN
40400 LET s=s+c
40500 DIM a#(1)
40600 INPUT "De# ja novo jogo? "
a#
40700 IF a#="n" OR a#="N" THEN GO
TO 0000,00: PRINT AT 0,0;"OK.
date: outo ven.": POUND 00: GCR
0007: POUNDLINE USR 47000
40800 GO SUB 31000
40900 GO TO 0000
00000 REM titulo
00100 GOTO 0: PAPER 1: INK 4: C

```


TEMPOS LIVRES

PARES DE CARTAS

Este programa — escrito por Raymond Blake — baseia-se no jogo Pangolines, no qual é necessário escolher pares de cartas de um conjunto de 20 até não sobrar nenhuma. O objectivo consiste em fazê-lo no menor número de movimentos possível.

Quando lhe for pedido, indique as coordenadas da carta a virar, escrevendo primeiramente o número da coluna. A carta escolhida será então virada. Proceda novamente do mesmo modo, virando uma segunda carta. Se ambas forem iguais, são eliminadas. Se não, são viradas de novo ao contrário. O jogo termina quando não resta nenhuma carta, sendo indicada a pontuação do jogador. Quanto menor for a pontuação melhor. Raymond Blake desafia-o a bater o seu melhor: 17 lances.

```
30 BORDER 0: PAPER 0: INK 7: 0
L3
50 DIM a(4,5) RANDOMIZE
50 LET a#= "1110131415012223242
5010004004142434445"
50 FOR i=1 TO 10
50 FOR j=1 TO 5
100 LET (LEN a#): LET b=INT (RN
0*(1/2)*2+1: LET a#=a#(b TO b+1)
120 LET x=VAL a#(i): LET y=VAL
a#(j)
130 LET a#=(a#( TO b-1)+(a#(b+2
TO i) AND b<>(i-1)
150 LET a(x,y)=i: NEXT j: NEXT
i
200 LET sc=0
210 FOR i=1 TO 5: PRINT AT 0,4+
i-1: INK 0: INVERSE 1: BRIGHT 1:
1: NEXT i
220 FOR l=0 TO 3: PRINT AT j*5+
3,0: INK 0: INVERSE 1: BRIGHT 1:
j+1: FOR i=0 TO 4: FOR k=2 TO 5:
PRINT AT j*5+k,i*4+2: INK 6: █
```

```
█: NEXT k: BEEP .03,30: NEXT i:
NEXT l
330 PRINT AT 0,20: "Virar": AT 0,
25: " "
340 IF INKEY#<>" " THEN GO TO 34
5
350 LET a#=INKEY#: IF a#<"1" OR
a#>"4" THEN GO TO 350
360 PRINT AT 0,24: a#: BEEP .05,
20: LET a1=VAL a#
370 IF INKEY#<>" " THEN GO TO 37
5
380 LET a#=INKEY#: IF a#<"1" OR
a#>"5" THEN GO TO 380
390 PRINT AT 0,20: a#: BEEP .05,
20: LET a2=VAL a#
400 IF a(a1,a2)<>a THEN GO TO 4
50
410 PRINT AT 10,23: INVERSE 1: "
Reverte"
420 BEEP 1,-10: PRINT AT 0,24: "
": AT 10,23: "
430 GO TO 330
450 PRINT AT 0,20: " "
T 0,24: "
470 LET x1=(a1-1)*5+2: LET y1=(
a2-1)*4+2: LET x=x1: LET y=y1: L
ET c=a1: LET d=a2: GO SUB 000
480 LET a1=c: LET a2=d
500 PRINT AT 0,20: "Virar": AT 0,
25: " "
510 IF INKEY#<>" " THEN GO TO 51
5
520 LET a#=INKEY#: IF a#<"1" OR
a#>"4" THEN GO TO 520
530 PRINT AT 0,24: a#: BEEP .05,
20: LET a1=VAL a#
570 IF INKEY#<>" " THEN GO TO 57
5
580 LET a#=INKEY#: IF a#<"1" OR
a#>"5" THEN GO TO 580
590 PRINT AT 0,20: a#: BEEP .05,
20: LET a2=VAL a#
600 IF a(a1,a2)<>a THEN GO TO 6
50
610 PRINT AT 10,23: INVERSE 1: "
Reverte"
620 BEEP 1,-10: PRINT AT 0,24: "
": AT 10,23: "
630 GO TO 530
650 PRINT AT 0,20: " "
T 0,24: "
670 LET x2=(a1-1)*5+2: LET y2=(
```

```

p2-1)*4+2: LET x=x2: LET y=y2: L
RT c=p1: LET d=p2: GO SUB 800
800 LET p1=c: LET p2=d
805 LET sc=sc+1
780 IF a(q1,q2)<>b(p1,p2) THEN
GO TO 760
785 FOR i=0 TO 12 STEP 5: BEEP
.1,i: NEXT i
710 FOR i=0 TO 3: PRINT AT x1+i
,y1;"": AT x2+i,y2;"": NEXT
i
720 LET a(q1,q2)=0: LET a(p1,p2
)=0
730 LET c=0: FOR i=1 TO 4: FOR
j=1 TO 5: LET c=c+a(i,j): NEXT j
: NEXT i
740 IF c=8 THEN GO TO 900
750 GO TO 300
760 BEEP .5,0
765 FOR f=1 TO 200: NEXT f: FOR
i=0 TO 3: PRINT AT x1+i,y1: INK
8;"": AT x2+i,y2: INK 8;"":
: NEXT i
770 GO TO 300
800 RESTORE 1000: FOR z=1 TO a(
c,d): READ z#: NEXT z
820 PRINT AT x,y;z#(1 TO 3): AT x
+1,y;z#(4 TO 5)
830 PRINT AT x+2,y;z#(7 TO 9): A
T x+3,y;z#(10 TO 12)
850 RETURN
880 CLS
920 PRINT AT 8,2:"Voce limpou a
tabela apos": AT 10,11;c:" lanc
e s"
930 FOR i=1 TO 25: BEEP .01,i:
NEXT i
940 LET a#="Carregue numa tecla
": PRINT AT 10,2)
950 FOR i=1 TO LEN a#: PRINT IN
K 8;a#(i): IF a#(i)<>" " THEN B
EEP .50,20
955 IF a#(i)=" " THEN PAUSE 2
955 NEXT i
960 IF INKEY#="" THEN GO TO 960
970 RUN
1000 DATA "
1010 DATA "

```

SUBMARINO

Neste exigente jogo programado por Malcolm Young, é necessário perseguir e destruir um submarino no menor tempo possível, empregando a nossa esquadra de uma forma estrategicamente adequada.

Controla-se uma esquadra de navios de guerra formada por um cruzador, duas corvetas e três fragatas. A missão consiste em perseguir e destruir um submarino inimigo que tem frequentemente as rotas comerciais. O seu cruzador encontra-se equipado com um sistema de sonar, que produz um som cada vez mais agudo à medida que se aproxima do submarino.

Cada navio só pode transportar uma quantidade limitada de cargas de profundidade. Depois de estas terem sido lançadas, deve voltar ao porto. O programa contém todas as instruções necessárias.

```

10 REM Submarino
20 REM © 1980 Malcolm Young
30 REM a=# b=# c=# d=# e=# f=#
9=1 b=# c=# d=# e=# f=#
40 DEF FN b()=500*(FN e(8),e(2
11))+FN e(2),e(1,2))
50 DEF FN s(a,b)=ABS(a-b)*ABS
(a-b)

```

```

00 DEF FN r(x)=INT (XND*x)+1
70 DEF FN t(j,k,l,m)=(ABS (j-k
) < 7) * (l==m)
00 GO SUB 9999: REM definir as
arices
00 GO SUB 9999: REM instrucciones
100 GO SUB 7999: REM construir
1000#
1100 REM vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
1200 REM #over 905
1300 LET m=FN r(9)
1400 LET ex=ex+(m#1)-(m#10)
1500 LET ex=ex+(m#2)-(m#10)
1600 LET ex=ex+(m#3)-(m#10)
1700 LET ex=ex+(m#4)-(m#10)
1800 LET ex=ex+(m#5)-(m#10)
1900 LET ex=ex+(m#6)-(m#10)
2000 LET ex=ex+(m#7)-(m#10)
2100 LET ex=ex+(m#8)-(m#10)
2200 IF FN t(ex,s(c,1),ex,s(c,2)
) THEN GO TO 9999
2300 IF FN t(ex,s(c,2),ex,s(c,1)
) THEN GO TO 9999
2400 NEXT c
2500 REM movimento do jogador
2600 PRINT #5:"Carregue para lan
car deacos."
2700 IF s(1,1)<=0 THEN PRINT AT
10,21:"Sobrar";AT 10,23:INK 3;F
LASH 0;INK
0
2800 IF s(1,1)>=8 THEN BEEP .1,1
0
2900 BEEP .1,1
3000 IF INKEY#="" THEN GO TO 330
3100 REM acao
3200 LET d=FN r(20)+10: LET m=FN
r(9)
3300 FOR i=1 TO 4
3400 LET m=m*(m<8)+1: BEEP .1,m
3500 PRINT AT 10,21:"Vozes tes
";
AT 10,21:CHR# (161+m);"sov."
3600 NEXT i
3700 PRINT BRIGHT 1;FLASH 1;IN
K #;PAPER 0;AT 10,21:CHR# (161+
m)
3700 BEEP 1,10: LET n=0
3800 LET n=n*(n<8)+1
3900 LET p=s(n,2): LET r=s(n,1)
4000 IF r<0 THEN GO TO 380
4100 PRINT #FLASH 1;INK n-1;PAP
ER 7;AT p,r;d#(n)
4200 PRINT PAPER n-1;INK 0;AT 0

```

```

0,0)=s(n);"(")s(n,1);",")s(n,2);
")"
430 PRINT AT 21,0:"Carregue de pr
ojetos para: ";s(n,3)
440 INPUT AT 0,0:"Indique movim
ento. ";LINE #;IF #="" THEN
PRINT INK n-1;PAPER 0+(n=8);AT
p,r;d#(n): GO TO 380
440 LET l=LEN #:IF l># THEN 0
BEEP 1,-10: GO TO 430
450 FOR c=1 TO l
460 IF #c<>"n" AND #c<>"s
" AND #c<>"o" AND #c<>"a"
AND #c<>"d" THEN INPUT AT 0,0
:"Indique ";LINE #:GO TO 440
470 IF #c="d" THEN GO TO 670
480 NEXT c
4900 REM #over javio
5000 FOR a=1 TO l
5100 PRINT AT p,r;PAPER 0;BRI0
HT 1;INK 0;
5200 BEEP .1,10
5300 LET p=p+(#a="e")-(#a)=
"n")+(<0)-(>8)
5400 IF #a="e" AND #a<8 THEN LET s(
n,0)=0: BEEP .1,10: PRINT AT 21,0
:"0: Carregue de projetos para a bor
da ";LINE #:GO TO 380
5400 LET p=p+(#a="e")-(#a)=
"o")+(<0)-(>8)
5500 IF ATTR (p,r)<64 THEN GO TO
380
5600 PRINT AT p,r;PAPER 0;INK
n-1(<6);OVER 1;d#(n)
5700 BEEP .1,10
5800 LET m=m-1
5900 PRINT BRIGHT 1;FLASH 1;IN
K #;PAPER 0;AT 10,21:CHR# (161+
m)
6000 NEXT a
6100 LET s(n,1)=r: LET s(n,2)=p:
IF NOT v THEN GO TO 380
6200 INPUT "Quer lancar uma cara
? ";LINE #
6300 IF #<>"e" AND #<>"n" THEN
INPUT "Lance uma cara? ";LINE
#;GO TO 380
6400 IF #="e" THEN GO TO 670
6500 IF #="n" THEN GO TO 380
6600 GO TO 100
6700 REM atrear carca
6800 IF s(n,0)<1 THEN PRINT FLAS

```

```

1010 AT 01,0;"Neo possui penhumb
carregal": GO TO 000
0000 PRINT AT 0,01; PAPER 5; INK
00;"PREFUNDO"
0000 FOR t=1 TO 7: BORDER 7: BRT
0,0,-1: BORDER 6: PAPER 6: NEXT
0
0000 PRINT FLASH 1; INK 7; PAPER
0,01,0;"Carregal de profundidade
de profundidade"
0000 INPUT "Indique profundidade
(1/10)";d
0000 IF d<1 OR d>10 THEN INPUT
(40);"Correção de entrada,");d
0000 TO 700
0000 LET d=INT d
0000 PRINT d,0,-10: BRT 0,0
0000 FOR t=10 TO -10 STEP -1
0000 BRT 0,0,0
0000 NEXT t
0000 BRT 0,0,-10: PAPER 00
0000 IF EX AND SC=0 THEN BRT 0
0,0,1: IF EX AND SC THEN GO TO 10000
0000 LET t=0: BRT 0,0,0
0000 PRINT FLASH 1; AT 01,0;"Falt
ou": IF EX AND SC THEN PRINT
T AT 01,0;"O submarino está por
baixo"
0000 PAPER 100
0000 PRINT AT 01,0;"
0040 GO TO 110
10000 BRT SC destruido
10000 BORDER 1: PAPER 1: BRIGHT 1
: INK 7: CLR
10000 PRINT AT 0,0;"PARABENS!"
10000 PRINT AT 0,0;"Voz consecui
u de sucesso"
10000 PRINT AT 0,0;"submarino em
baixo": test
10000 PRINT "trabalho: "PC/054
10000:"%
10000 INPUT "Cada novo jogo? ";
LINE #
10000 IF #="s" THEN RUN
10000 BORDER 7: PAPER 7: INK 0: 0
PRINT 0: CLR: STOP
00000 BRT COLLISAO
00010 BRT 1,-100
00020 PRINT AT 01,0;"COLISAO!";
00030 IF ATTR (0,0)=40 THEN PRINT
AT 00,0; FLASH 1; INK 0; PAPER

```

```

7);s#(0);"afundou-se!": LET s(0,
1)=-1
0040 IF ATTR (0,0)<0 THEN
GO TO 0100
0050 PRINT AT 00,0; FLASH 1; INK
0; PAPER 7);s#(0);"afundou-se!"
: LET s(0,1)=-1
0060 FOR s=1 TO 0 STEP -1: BRT
0,s#(10): NEXT s
0070 PAPER 100
0080 PRINT AT 00,0;TAB 31;" "TA
B 31;
0090 GO TO 110
0100 LET s=ATTR (0,0)-40
0110 BRT
0120 BRT ++++++
0130 BRT escape do sub
0140 BRT AND<.7 THEN GO TO 0200
0150 BORDER 0: LET u=1: PRINT AT
0,0,1; PAPER 2; INK 7;"VERMELHO"
0160 FOR t=1 TO 7: BORDER 4: BRT
0,0,-1: BORDER 0: PAPER 0: NEXT
0
0170 IF AND<.5 THEN GO TO 0500
0180 PRINT AT 01,0; INK 7; BRIGHT
T 1; s#(0);"voz escape!"
0190 PAPER 00
0200 LET t=0
0210 PRINT AT 01,0; PAPER 0; TLA
SH 1;"Correção em f e teste escape
teste."
0220 FOR t=1 TO 2
0230 LET t1=RN r(10): LET pf=(t1
*0+104)-200
0240 PRINT AT 0,00+0; PAPER 0; I
NK 1;"+"
0250 LET q=t*(0<10)+1
0260 PAPER 10
0270 PRINT AT 0,00+0; PAPER 0; I
NK 0-(0<0);4#(0)
0280 PAPER 40
0290 IF INKEY#<>"f" THEN GO TO 0
0300
0310 FOR i=7 TO 1 STEP -1
0320 PLOT 000,00: DRAW INVERSE 1
: INK 1;0,40
0330 BRT 0,0: NEXT i
0340 PLOT 000,00: DRAW INK 1;0,
40: PAPER 00
0350 IF t1=0 THEN GO TO 0300
0360 PRINT AT 01,0; PAPER 2; INK
7;"
00a accao evasiva!"

```



```

00010 PRUSE 100: PRINT AT 01,0;"O
00020 IF #0<100 THEN STOP: IF #0=100
00030 GO TO 100: SUGGESTIO: TOP # =
00040 PRUSE 100: PRINT AT 01,0;TAB
00050
00060 NEXT 1
00070 PRINT AT 00,0;"Tale e altre
00080
00090 PRINT AT 00,0;TAB
00100
00110 PRUSE 100: PRINT AT 01,0;"
00120 IF #0<100 THEN STOP: IF #0=100
00130 GO TO 100: SUGGESTIO: TOP # =
00140 PRUSE 100: PRINT AT 01,0;"
00150 IF #0<100 THEN STOP: IF #0=100
00160 GO TO 100: SUGGESTIO: TOP # =
00170 PRUSE 100: PRINT AT 01,0;"
00180 IF #0<100 THEN STOP: IF #0=100
00190 GO TO 100: SUGGESTIO: TOP # =
00200 PRUSE 100: PRINT AT 01,0;"
00210 IF #0<100 THEN STOP: IF #0=100
00220 GO TO 100: SUGGESTIO: TOP # =
00230 PRUSE 100: PRINT AT 01,0;"
00240 IF #0<100 THEN STOP: IF #0=100
00250 GO TO 100: SUGGESTIO: TOP # =
00260 PRUSE 100: PRINT AT 01,0;"
00270 IF #0<100 THEN STOP: IF #0=100
00280 GO TO 100: SUGGESTIO: TOP # =
00290 PRUSE 100: PRINT AT 01,0;"
00300 IF #0<100 THEN STOP: IF #0=100
00310 GO TO 100: SUGGESTIO: TOP # =
00320 PRUSE 100: PRINT AT 01,0;"
00330 IF #0<100 THEN STOP: IF #0=100
00340 GO TO 100: SUGGESTIO: TOP # =
00350 PRUSE 100: PRINT AT 01,0;"
00360 IF #0<100 THEN STOP: IF #0=100
00370 GO TO 100: SUGGESTIO: TOP # =
00380 PRUSE 100: PRINT AT 01,0;"
00390 IF #0<100 THEN STOP: IF #0=100
00400 GO TO 100: SUGGESTIO: TOP # =
00410 GO TO 0000
00420 REM Suscribirse suite proxim
00430
00440 PRINT AT 01,0; FLASH 1; BR
00450 I 1; INK 7;"Alerta!"
00460 PRUSE 100
00470 PRINT AT 01,0;"#(0)"; avist
00480 Suscribirse!"
00490 PRUSE 100
00500 GO TO 0000
00510 GO TO 0000
00520 REM #1000
00530 BORDER 4: PAPER 5: INK 1: B
00540 RIGHT 1: OLS
00550 FOR i=0 TO 10: PRINT BRIGHT
00560 1;"|"; PER 2
00570 S graphics 7
00580 NEXT 1
00590 PLOT 0,10: DRAW 100,0: DRAW
00600 0,100
00610 REM #1000
00620 LET #FN r (10)+10

```

```

7070 FOR i=1 TO #: LET mx=FN 711
01): LET #u=FN 7(10): IF ATTR (#0
,mx)>64 THEN PRINT AT #0,mx;"#
7075 NEXT 1
7080 REM device
7090 FOR a=1 TO 0: PRINT AT 010,
0),e(a,1); OVER 1; BRIGHT 1; PR
PR 7; INK a-1;d#(a); NEXT a
7100 PRINT AT 01,0; FLASH 1;"SUB
MARINO!"; FLASH 0; AT 1,01;"SUB
00"; AT 2,01; PAPER 4;"MARO"
7110 REM #10000
7120 BORDER 4
7130 FOR p=1 TO 0 STEP -1
7140 PLOT 000-p,100: DRAW p*2,0,
01
7150 NEXT p
7160 FOR i=0 TO 0: PRINT OVER 1;
AT 4+i,100;"+"; LET i=i+(i=0); N
EXT 1; REM graphics 0; 11
7170 PRINT OVER 1; AT 0,01;"+++++
+++++"
7180 RETURN
0000 REM instrucciones
0010 BORDER 1: PAPER 5: INK 1: B
RIGHT 1: OLS
0020 DIM a$(6,10); DIM s(6,0); 0
IM i(1): LET c=0: LET i=1: LET
d#="####"; REM graphics 0000
a
0030 FOR a=1 TO 6: READ .1,-a; 0
PRINT AT 1,0; INK 2; INVOCAR;"0
UMARINO!"; READ .05,a: LET i=i-
1; NEXT a
0040 INPUT BRIGHT 1;"Cada 1a inst
ruccion (s/d)?" #
0050 IF i#<>"s" THEN GO TO 0010
0060 FOR a=1 TO 6: READ t#: READ
a#
0070 PRINT AT 0,0;p#""a#
0080 DIM t$(LEN t#); DIM t$(LEN
a#)
0090 INPUT "Carreoue a# "; FLASH
1;"ENTER"; FLASH 0;" "; LINE p#
0100 PRINT AT 0,0;p#""t#; NEXT
a#
0110 IF t#="0" THEN GO TO 0210
0120 IF t#<>"s" THEN INPUT "over
det cu# some 'a sus prote (s/d)
"? ; #; GO TO 0110
0130 REM device
0140 POKE 20000,0: PROTORE 0000:

```

```

BORDER 4: PAPER 5: BRIGHT 6: CL
0100 FOR a=1 TO 5: READ c#
0110 PRINT AT 5,5:"Voz nome gost
0120 GOTO 0170
INPUT a#
0170 IF LEN a#>15 OR a#="" THEN
INPUT "Digite o nome do navio:":a#
0180 GOTO 0170
0190 LET s#(a)=a#
0200 PRINT "Nome do navio: ";a#;LNK a-1:AT
a#+5,5,5:GOTO 0210
0210 POKB a#(a):GOTO 0220
0220 POKB a#(a):GOTO 0230
0230 FOR a=1 TO 5:
READ a#
0240 FOR a=1 TO 5: READ d:
LET s(1,1)=a-1:LET s(1,2)=d:LET s(
1,3)=a:NEXT d
0250 LET ex=FN r(20)-1:LET ay=FN
r(20)-1:LET ax=FN r(10)
0260 LET y=FN b(1):LET ay=0:RE
TURN
0300 REM
0310 REM
0320 FOR c=1 TO 14: READ c#
0330 FOR b=0 TO 7: READ dt: POKB
c#+b,dt:NEXT b
0340 NEXT c:RETURN
0350 DATA "a",5,0,0,10,20,107,10
0360 DATA "b",10,10,50,100,200,2
0370 DATA "c",5,0,0,204,90,200,2
0380 DATA "d",107,74,44,31,240,5
0390 DATA "e",5,40,40,100,30,110
0400 DATA "f",20,100,100,100,10
0410 DATA "g",5,5,1,1,7,1,1,0:R
0420 DATA "h",0,0,0,0,224,0,0,0:
REM
0430 DATA "i",0,0,0,24,24,0,0,0:
REM dado #1
0440 DATA "j",0,0,0,0,0,0,0,0:
REM dado #2
0450 DATA "k",3,3,0,24,24,0,100,
0460 REM dado #3
0470 DATA "l",0,100,100,0,0,100,
100,0:REM dado #4

```

```

0170 DATA "#",100,100,0,24,24,0,
100,100:REM dado #5
0180 DATA "n",100,100,0,100,100,
0,100,100:REM dado #6
0190 DATA "Voz comando usa rot
e composta por 1 cruzador, 2 cor
vetes e 3 fragatas."
0200 DATA "A acata missao consiste
em des- truir um submarino in
#igo que frequenta as rotas co
merciais."
0210 DATA "O seu cruzador esta
equipado com um sonar, que pro
duz um som mais agudo ao aproxim
ar-se do inimigo."
0220 DATA "Cada navio pode trans
portar ate - nas uma quantidade li
mitada de cargas de profundidad
e; quando as cargas deve voltar
ao porto. Cruzador - 4 cargas
Fragata - 3 cargas
Corvete - 2 cargas."
0230 DATA "O submarino pode torp
edear qual- quer navio a uma dist
ancia de 5 Unidades, medida vert
ical ou horizontalmente."
0240 DATA "O submarino move-se u
na Unidade em cada lance, e pode
atingir a profundidade de 10 Un
idades. Se um dos seus navios se
encontrar directamente por cima
dele, soa- ra um alarme."
0250 DATA "O numero de movimento
s de que dispoe em cada lance
depende de um dado. Podera usa-
los para qual- quer navios ate
perfezer o total."
0260 DATA "O computador pergunta
o- lha- se se quer mover cada navio
e se nao quiser, carregue em R
NTER. As direccoes sao: N, S,
E, O. Pode disparar as cargas ca
rregando END."
0270 DATA "Se houver colisoes, s
era' arun- dado o navio mais peq
ueno, o mesmo acontecendo em
choques cominas."
0280 DATA " ", "Navios sob o seu
comando: Cte. Jose Belo (in
avio almirante) Antonio Enes (fra
gata 1) Joao Coutinho (fr

```

```

segata 2)      Pero Escobar (fra
gata 3)      Bobvista (corveta)
" 1)        Brava (corveta 2)

9300 DATA "Deixe 487 nomes a s
ua frota (E/N)?"
9300 DATA "o seu cruzador", "a 77
segata 1", "a fragata 2", "a fragat
a 3", "a corveta 1", "a corveta 2"
9310 DATA "CTE, JOAO BELO", "ANTO
NIO ENES", "JOAO COUTINHO", "PERO
ESCOBAR", "BOBVISTA", "BRAVA"
9320 DATA 4,3,3,3,2,2
9399 BORDER 7: PAPER 7: INK 0: B
RIGHT 0: INVERSE 0: CLS : STOP

```

O CARRASCO

Apresentamos aqui uma versão bastante boa (escrita por Raymond Blake) de um jogo muito popular. Foi pensado de tal modo que o leitor possa adaptar a lista de palavras, que aliás foi mantida deliberadamente pequena para o encorajar.

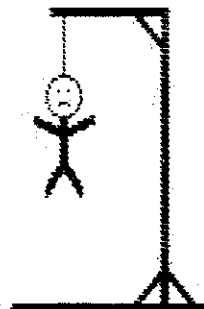
SPECTRUM HANGMAN

Da próxima vez que quiser ajuda,
não venha ter comigo!

-A--I-A

Letras erradas

BCDEFGHJKL



Para alterar as palavras, modifique simplesmente o número 30 da linha 70, substituindo-o pelo número total de palavras da sua lista, e escreva-as em declarações DATA no final do programa, tal como acontece nas linhas 1500 a 1520. O programa

dar-lhe-á maior prazer ainda se pedir a um amigo que altere a lista de palavras, de modo a você poder jogar sem as conhecer.

```

200 POKE 23550,3
205 LET T=0: RANDOMIZE
210 BORDER 0: PAPER 7: INK 0: C
LS : RESTORE
40 PRINT AT 0,7:"SPECTRUM HANG
MAN": INK 0: AT 1,7:"-----"
50 IF T=0 THEN PRINT AT 0,0:"S
e control: Estes #alucos vao ser
o crerime se voce nao conseguir a
divulgar a palavra!"
50 IF T=1 THEN PRINT AT 0,0:"O
h, nao! Agora querem enforcar o#
eu amigo! Voce vai ter de nos a
judar!"
50 LET W=0: LET W$="": LET C=0
70 FOR I=1 TO INT (RND*30+1):
READ A#: NEXT I
70 FOR I=20 TO 30: BEEP .02,I
: NEXT I
80 LET L=LEN A#: LET B$="-----"
-----(I TO L)
90 PRINT AT 12,0: INK 0;B$
100 PRINT AT 0,0: INK 1;"Indica
a letra"
120 INPUT C#: IF LEN C#<>1 AND
LEN C#<>L THEN GO TO 120
130 IF C#<"0" OR C#>"Z" THEN GO
TO 120
140 PRINT AT 0,0:"
150 IF LEN C#=1 THEN GO TO 180
160 IF C#=A# THEN LET B#=C#
170 IF C#<>A# THEN LET W=L
175 GO TO 200
180 LET W=0: FOR I=1 TO L: IF A
#(I)=C# THEN LET B#(I)=C#
190 IF C#<>A#(I) THEN LET W=W+1
195 NEXT I
200 PRINT AT 12,0: INK 2;B$: IF
W=L THEN LET C=C+1: GO SUB 730+
C*20: BEEP .0,5*(10-C)-25: LET W
#=C#+(C# AND LEN C#=1)
210 IF W<>L THEN FOR I=10 TO 30
: BEEP .02,I: NEXT I
220 PRINT AT 17,0:"Letras errad
as": PRINT AT 19,2: INK 4;W$
240 IF A#=B# THEN GO TO 500
250 IF C=10 THEN GO TO 500
260 GO TO 100
500 PRINT AT 0,0: INK 3;"Da pro

```

xima vez que quiser ajuda, nao ve
nha ter comigo!

```
520 FOR I=20 TO -30 STEP -1: BE  
EP .02, I: NEXT I: GO TO 550  
600 PRINT AT 3,0; INK 3; "Muito  
obrigado. Devo-lhe um co-po, am  
igo!"  
520 FOR I=-20 TO 30: BEEP .02, I  
: NEXT I  
650 PRINT AT 10,2; INK 1; "A pal  
avra era: "; AT 12,3; A#  
660 PAUSE 120  
590 LET F=1: GO TO 30  
600 FOR J=10 TO 12: PLOT 150, J:  
DRAW 30,0: NEXT J  
610 FOR J=212 TO 214: PLOT J, 13  
: DRAW 0, 100: NEXT J: RETURN  
620 FOR J=201 TO 203: PLOT J, 13  
: DRAW 10, 10: NEXT J  
630 FOR J=21 TO 23: PLOT 215, J:  
DRAW 10, -10: NEXT J: RETURN  
640 FOR J=111 TO 113: PLOT 211,  
J: DRAW -45,0: NEXT J  
650 FOR J=201 TO 203: PLOT J, 11  
0: DRAW 10, -10: NEXT J: RETURN  
660 PLOT 171, 110: DRAW 0, -20: R  
ETURN  
880 CIRCLE 171, 83, 7: PLOT 169, 8  
5: PLOT 173, 85  
890 PLOT 169, 30: DRAW 4, 0, .5*-P  
I: RETURN  
900 FOR J=170 TO 172: PLOT J, 75  
: DRAW 0, -14: NEXT J: RETURN  
920 FOR J=70 TO 72: PLOT 169, J:  
DRAW -10, 4: NEXT J: RETURN  
940 FOR J=70 TO 72: PLOT 173, J:  
DRAW 10, 4: NEXT J: RETURN  
960 FOR J=59 TO 62: PLOT 171, J:  
DRAW -7, -10: NEXT J: RETURN  
980 FOR J=59 TO 62: PLOT 171, J:  
DRAW 7, -10: NEXT J: RETURN  
1500 DATA "ISOTOPO", "POKER", "PLA  
NETA", "ESFERA", "PALACIA", "TRIUMF  
O", "CANGURU", "ATRIBUTO", "PRAZO",  
"CAMINHO"  
1510 DATA "HOTEL", "GALAXIA", "UNI  
VERSO", "MIRAGEM", "ESTREMECER", "C  
OLORADO", "SORRISO", "VISAO", "ESTR  
ADA", "SALTEADO"  
1520 DATA "ARDOSIA", "TEMPORAL", "  
COLEGA", "IMPRESSORA", "MAQUINA", "  
DESCOBRIR", "DIVISAO", "REVISTA", "  
PALMEIRA", "CORREDOR"
```

JOGOS NO ESPAÇO

FUGA ENTRE AS ESTRELAS

Este pequeno programa, escrito por Graham Charlton, produz um jogo surpreendentemente interessante. Deve usar as teclas «Z» e «M» para se deslocar para a esquerda e direita, evitando os asteriscos que se movem verticalmente.

```
10 BORDER 2: PAPER 2: CLS  
20 LET a=10: LET b=15: LET c=2  
0: LET t=0  
30 POKE 23692,0  
40 LET t=t+1  
50 PRINT AT 21, 31; " "  
60 FOR Z=1 TO 6  
70 INK 7: PRINT AT c, AND#31; "*"   
80 NEXT Z  
90 LET b=b+(INKEY$="M" AND b<3  
0)-(INKEY$="Z" AND b>1)  
100 IF SCREEN$(a,b)="#" THEN B  
EEP 2,3: PRINT "Portuugu "; t: FOR  
x=1 TO 200: NEXT x: RUN  
110 PRINT AT a,b; "0"  
120 GO TO 30
```

SPACE TREK

Apresentamos agora uma outra contribuição de David Perry para este livro. Neste jogo, você voa através das galáxias, evitando as estrelas, tentando disparar contra os cruzadores de batalha inimigos. Quando tiver terminado, deve abandonar a galáxia através do Buraco Negro, afastando-se para as profundezas do espaço...

O programa inclui todas as instruções necessárias. Os comandos são dados usando as teclas de cursor, e zero para disparar.

```

1 RAM
2 RAM
3 RAM SPACE TRAXI
4 RAM #####
5 RAM
6 RAM
7 RAM COPYRIGHT
8 RAM DAVID BERRY
9 RAM 1983
10 RAM
11 RAM
12 RAM
13 RAM
14 RAM
15 RAM
16 RAM
17 RAM
18 RAM
19 RAM
20 RAM
21 RAM
22 RAM
23 RAM
24 RAM
25 RAM
26 RAM
27 RAM
28 RAM
29 RAM
30 RAM
31 RAM
32 RAM
33 RAM
34 RAM
35 RAM
36 RAM
37 RAM
38 RAM
39 RAM
40 RAM
41 RAM
42 RAM
43 RAM
44 RAM
45 RAM
46 RAM
47 RAM
48 RAM
49 RAM
50 RAM
51 RAM
52 RAM
53 RAM
54 RAM
55 RAM
56 RAM
57 RAM
58 RAM
59 RAM
60 RAM
61 RAM
62 RAM
63 RAM
64 RAM
65 RAM
66 RAM
67 RAM
68 RAM
69 RAM
70 RAM
71 RAM
72 RAM
73 RAM
74 RAM
75 RAM
76 RAM
77 RAM
78 RAM
79 RAM
80 RAM
81 RAM
82 RAM
83 RAM
84 RAM
85 RAM
86 RAM
87 RAM
88 RAM
89 RAM
90 RAM
91 RAM
92 RAM
93 RAM
94 RAM
95 RAM
96 RAM
97 RAM
98 RAM
99 RAM
100 RAM
101 RAM
102 RAM
103 RAM
104 RAM
105 RAM
106 RAM
107 RAM
108 RAM
109 RAM
110 RAM
111 RAM
112 RAM
113 RAM
114 RAM
115 RAM
116 RAM
117 RAM
118 RAM
119 RAM
120 RAM
121 RAM
122 RAM
123 RAM
124 RAM
125 RAM
126 RAM
127 RAM
128 RAM
129 RAM
130 RAM
131 RAM
132 RAM
133 RAM
134 RAM
135 RAM
136 RAM
137 RAM
138 RAM
139 RAM
140 RAM
141 RAM
142 RAM
143 RAM
144 RAM
145 RAM
146 RAM
147 RAM
148 RAM
149 RAM
150 RAM
151 RAM
152 RAM
153 RAM
154 RAM
155 RAM
156 RAM
157 RAM
158 RAM
159 RAM
160 RAM
161 RAM
162 RAM
163 RAM
164 RAM
165 RAM
166 RAM
167 RAM
168 RAM
169 RAM
170 RAM
171 RAM
172 RAM
173 RAM
174 RAM
175 RAM
176 RAM
177 RAM
178 RAM
179 RAM
180 RAM
181 RAM
182 RAM
183 RAM
184 RAM
185 RAM
186 RAM
187 RAM
188 RAM
189 RAM
190 RAM
191 RAM
192 RAM
193 RAM
194 RAM
195 RAM
196 RAM
197 RAM
198 RAM
199 RAM
200 RAM
201 RAM
202 RAM
203 RAM
204 RAM
205 RAM
206 RAM
207 RAM
208 RAM
209 RAM
210 RAM
211 RAM
212 RAM
213 RAM
214 RAM
215 RAM
216 RAM
217 RAM
218 RAM
219 RAM
220 RAM
221 RAM
222 RAM
223 RAM
224 RAM
225 RAM
226 RAM
227 RAM
228 RAM
229 RAM
230 RAM
231 RAM
232 RAM
233 RAM
234 RAM
235 RAM
236 RAM
237 RAM
238 RAM
239 RAM
240 RAM
241 RAM
242 RAM
243 RAM
244 RAM
245 RAM
246 RAM
247 RAM
248 RAM
249 RAM
250 RAM
251 RAM
252 RAM
253 RAM
254 RAM
255 RAM
256 RAM
257 RAM
258 RAM
259 RAM
260 RAM
261 RAM
262 RAM
263 RAM
264 RAM
265 RAM
266 RAM
267 RAM
268 RAM
269 RAM
270 RAM
271 RAM
272 RAM
273 RAM
274 RAM
275 RAM
276 RAM
277 RAM
278 RAM
279 RAM
280 RAM
281 RAM
282 RAM
283 RAM
284 RAM
285 RAM
286 RAM
287 RAM
288 RAM
289 RAM
290 RAM
291 RAM
292 RAM
293 RAM
294 RAM
295 RAM
296 RAM
297 RAM
298 RAM
299 RAM
300 RAM
301 RAM
302 RAM
303 RAM
304 RAM
305 RAM
306 RAM
307 RAM
308 RAM
309 RAM
310 RAM
311 RAM
312 RAM
313 RAM
314 RAM
315 RAM
316 RAM
317 RAM
318 RAM
319 RAM
320 RAM
321 RAM
322 RAM
323 RAM
324 RAM
325 RAM
326 RAM
327 RAM
328 RAM
329 RAM
330 RAM
331 RAM
332 RAM
333 RAM
334 RAM
335 RAM
336 RAM
337 RAM
338 RAM
339 RAM
340 RAM
341 RAM
342 RAM
343 RAM
344 RAM
345 RAM
346 RAM
347 RAM
348 RAM
349 RAM
350 RAM
351 RAM
352 RAM
353 RAM
354 RAM
355 RAM
356 RAM
357 RAM
358 RAM
359 RAM
360 RAM
361 RAM
362 RAM
363 RAM
364 RAM
365 RAM
366 RAM
367 RAM
368 RAM
369 RAM
370 RAM
371 RAM
372 RAM
373 RAM
374 RAM
375 RAM
376 RAM
377 RAM
378 RAM
379 RAM
380 RAM
381 RAM
382 RAM
383 RAM
384 RAM
385 RAM
386 RAM
387 RAM
388 RAM
389 RAM
390 RAM
391 RAM
392 RAM
393 RAM
394 RAM
395 RAM
396 RAM
397 RAM
398 RAM
399 RAM
400 RAM
401 RAM
402 RAM
403 RAM
404 RAM
405 RAM
406 RAM
407 RAM
408 RAM
409 RAM
410 RAM
411 RAM
412 RAM
413 RAM
414 RAM
415 RAM
416 RAM
417 RAM
418 RAM
419 RAM
420 RAM
421 RAM
422 RAM
423 RAM
424 RAM
425 RAM
426 RAM
427 RAM
428 RAM
429 RAM
430 RAM
431 RAM
432 RAM
433 RAM
434 RAM
435 RAM
436 RAM
437 RAM
438 RAM
439 RAM
440 RAM
441 RAM
442 RAM
443 RAM
444 RAM
445 RAM
446 RAM
447 RAM
448 RAM
449 RAM
450 RAM
451 RAM
452 RAM
453 RAM
454 RAM
455 RAM
456 RAM
457 RAM
458 RAM
459 RAM
460 RAM
461 RAM
462 RAM
463 RAM
464 RAM
465 RAM
466 RAM
467 RAM
468 RAM
469 RAM
470 RAM
471 RAM
472 RAM
473 RAM
474 RAM
475 RAM
476 RAM
477 RAM
478 RAM
479 RAM
480 RAM
481 RAM
482 RAM
483 RAM
484 RAM
485 RAM
486 RAM
487 RAM
488 RAM
489 RAM
490 RAM
491 RAM
492 RAM
493 RAM
494 RAM
495 RAM
496 RAM
497 RAM
498 RAM
499 RAM
500 RAM
501 RAM
502 RAM
503 RAM
504 RAM
505 RAM
506 RAM
507 RAM
508 RAM
509 RAM
510 RAM
511 RAM
512 RAM
513 RAM
514 RAM
515 RAM
516 RAM
517 RAM
518 RAM
519 RAM
520 RAM
521 RAM
522 RAM
523 RAM
524 RAM
525 RAM
526 RAM
527 RAM
528 RAM
529 RAM
530 RAM
531 RAM
532 RAM
533 RAM
534 RAM
535 RAM
536 RAM
537 RAM
538 RAM
539 RAM
540 RAM
541 RAM
542 RAM
543 RAM
544 RAM
545 RAM
546 RAM
547 RAM
548 RAM
549 RAM
550 RAM
551 RAM
552 RAM
553 RAM
554 RAM
555 RAM
556 RAM
557 RAM
558 RAM
559 RAM
560 RAM
561 RAM
562 RAM
563 RAM
564 RAM
565 RAM
566 RAM
567 RAM
568 RAM
569 RAM
570 RAM
571 RAM
572 RAM
573 RAM
574 RAM
575 RAM
576 RAM
577 RAM
578 RAM
579 RAM
580 RAM
581 RAM
582 RAM
583 RAM
584 RAM
585 RAM
586 RAM
587 RAM
588 RAM
589 RAM
590 RAM
591 RAM
592 RAM
593 RAM
594 RAM
595 RAM
596 RAM
597 RAM
598 RAM
599 RAM
600 RAM
601 RAM
602 RAM
603 RAM
604 RAM
605 RAM
606 RAM
607 RAM
608 RAM
609 RAM
610 RAM
611 RAM
612 RAM
613 RAM
614 RAM
615 RAM
616 RAM
617 RAM
618 RAM
619 RAM
620 RAM
621 RAM
622 RAM
623 RAM
624 RAM
625 RAM
626 RAM
627 RAM
628 RAM
629 RAM
630 RAM
631 RAM
632 RAM
633 RAM
634 RAM
635 RAM
636 RAM
637 RAM
638 RAM
639 RAM
640 RAM
641 RAM
642 RAM
643 RAM
644 RAM
645 RAM
646 RAM
647 RAM
648 RAM
649 RAM
650 RAM
651 RAM
652 RAM
653 RAM
654 RAM
655 RAM
656 RAM
657 RAM
658 RAM
659 RAM
660 RAM
661 RAM
662 RAM
663 RAM
664 RAM
665 RAM
666 RAM
667 RAM
668 RAM
669 RAM
670 RAM
671 RAM
672 RAM
673 RAM
674 RAM
675 RAM
676 RAM
677 RAM
678 RAM
679 RAM
680 RAM
681 RAM
682 RAM
683 RAM
684 RAM
685 RAM
686 RAM
687 RAM
688 RAM
689 RAM
690 RAM
691 RAM
692 RAM
693 RAM
694 RAM
695 RAM
696 RAM
697 RAM
698 RAM
699 RAM
700 RAM
701 RAM
702 RAM
703 RAM
704 RAM
705 RAM
706 RAM
707 RAM
708 RAM
709 RAM
710 RAM
711 RAM
712 RAM
713 RAM
714 RAM
715 RAM
716 RAM
717 RAM
718 RAM
719 RAM
720 RAM
721 RAM
722 RAM
723 RAM
724 RAM
725 RAM
726 RAM
727 RAM
728 RAM
729 RAM
730 RAM
731 RAM
732 RAM
733 RAM
734 RAM
735 RAM
736 RAM
737 RAM
738 RAM
739 RAM
740 RAM
741 RAM
742 RAM
743 RAM
744 RAM
745 RAM
746 RAM
747 RAM
748 RAM
749 RAM
750 RAM
751 RAM
752 RAM
753 RAM
754 RAM
755 RAM
756 RAM
757 RAM
758 RAM
759 RAM
760 RAM
761 RAM
762 RAM
763 RAM
764 RAM
765 RAM
766 RAM
767 RAM
768 RAM
769 RAM
770 RAM
771 RAM
772 RAM
773 RAM
774 RAM
775 RAM
776 RAM
777 RAM
778 RAM
779 RAM
780 RAM
781 RAM
782 RAM
783 RAM
784 RAM
785 RAM
786 RAM
787 RAM
788 RAM
789 RAM
790 RAM
791 RAM
792 RAM
793 RAM
794 RAM
795 RAM
796 RAM
797 RAM
798 RAM
799 RAM
800 RAM
801 RAM
802 RAM
803 RAM
804 RAM
805 RAM
806 RAM
807 RAM
808 RAM
809 RAM
810 RAM
811 RAM
812 RAM
813 RAM
814 RAM
815 RAM
816 RAM
817 RAM
818 RAM
819 RAM
820 RAM
821 RAM
822 RAM
823 RAM
824 RAM
825 RAM
826 RAM
827 RAM
828 RAM
829 RAM
830 RAM
831 RAM
832 RAM
833 RAM
834 RAM
835 RAM
836 RAM
837 RAM
838 RAM
839 RAM
840 RAM
841 RAM
842 RAM
843 RAM
844 RAM
845 RAM
846 RAM
847 RAM
848 RAM
849 RAM
850 RAM
851 RAM
852 RAM
853 RAM
854 RAM
855 RAM
856 RAM
857 RAM
858 RAM
859 RAM
860 RAM
861 RAM
862 RAM
863 RAM
864 RAM
865 RAM
866 RAM
867 RAM
868 RAM
869 RAM
870 RAM
871 RAM
872 RAM
873 RAM
874 RAM
875 RAM
876 RAM
877 RAM
878 RAM
879 RAM
880 RAM
881 RAM
882 RAM
883 RAM
884 RAM
885 RAM
886 RAM
887 RAM
888 RAM
889 RAM
890 RAM
891 RAM
892 RAM
893 RAM
894 RAM
895 RAM
896 RAM
897 RAM
898 RAM
899 RAM
900 RAM
901 RAM
902 RAM
903 RAM
904 RAM
905 RAM
906 RAM
907 RAM
908 RAM
909 RAM
910 RAM
911 RAM
912 RAM
913 RAM
914 RAM
915 RAM
916 RAM
917 RAM
918 RAM
919 RAM
920 RAM
921 RAM
922 RAM
923 RAM
924 RAM
925 RAM
926 RAM
927 RAM
928 RAM
929 RAM
930 RAM
931 RAM
932 RAM
933 RAM
934 RAM
935 RAM
936 RAM
937 RAM
938 RAM
939 RAM
940 RAM
941 RAM
942 RAM
943 RAM
944 RAM
945 RAM
946 RAM
947 RAM
948 RAM
949 RAM
950 RAM
951 RAM
952 RAM
953 RAM
954 RAM
955 RAM
956 RAM
957 RAM
958 RAM
959 RAM
960 RAM
961 RAM
962 RAM
963 RAM
964 RAM
965 RAM
966 RAM
967 RAM
968 RAM
969 RAM
970 RAM
971 RAM
972 RAM
973 RAM
974 RAM
975 RAM
976 RAM
977 RAM
978 RAM
979 RAM
980 RAM
981 RAM
982 RAM
983 RAM
984 RAM
985 RAM
986 RAM
987 RAM
988 RAM
989 RAM
990 RAM
991 RAM
992 RAM
993 RAM
994 RAM
995 RAM
996 RAM
997 RAM
998 RAM
999 RAM
1000 RAM

```

```

100 PRINT AT 0,4; INK 2;"Galaxi
101 gal: PRINT AT 0,14; INK 5;"K
102 lev: "lev
103 LET a=2: LET b=5: LET c=1
104 PRINT AT a,b;"
105 IF a=1 THEN LET a=a-1
106 IF a=0 THEN LET b=b+1
107 IF a=3 THEN LET a=b+1
108 IF a=4 THEN LET b=b-1
109 IF b<1 THEN LET b=30
110 IF b>30 THEN LET b=1
111 IF a>5 THEN LET a=20
112 IF a>20 THEN LET a=5
113 IF lev=0 THEN BEEP .01,50:
114 IF SCREEN$(a,b)="0" THEN LET b=
115 b+2: LET gal=gal+1: PRINT AT
116 0,11;gal: GO SUB 300: LET lev=b
117 GO TO 40
118 IF SCREEN$(a,b)<>" " THEN
119 FOR i=7 TO 0 STEP -1: PRINT AT a
120 ,b: INK i;"X": BEEP .02,i*7: NEX
121 T i: GO SUB 300: PRINT AT 1,10:"
122 ": LET med=med-1: FOR n=1 TO
123 med: PRINT AT 1,10+n;"a": NEXT
124 n: GO TO 40
125 IF INKEY#="7" THEN LET a=1
126 IF INKEY#="8" THEN LET a=2
127 IF INKEY#="9" THEN LET a=3
128 IF INKEY#="0" THEN LET a=4
129 IF a=1 THEN PRINT AT a,b: I
130 NK 0;"a"
131 IF a=2 THEN PRINT AT a,b: I
132 NK 0;"b"
133 IF a=3 THEN PRINT AT a,b: I
134 NK 0;"c"
135 IF a=4 THEN PRINT AT a,b: I
136 NK 0;"d"
137 IF INKEY#="0" THEN GO SUB 3
138 00
139 BEEP .01,d*10: GO TO 120
140 IF a=1 THEN GO TO 400
141 IF a=2 THEN GO TO 300
142 IF a=3 THEN GO TO 600
143 IF a=4 THEN GO TO 700
144 STOP
145 IF b=5 THEN RETURN
146 FOR n=a-1 TO 0 STEP -1
147 LET s$=SCREEN$(n,b)
148 IF s$="K" THEN LET lev=lev-
149 1: PRINT AT 0,20;lev;" ": LET s
150 c=c+100: PRINT AT 1,25;c: FOR
151 i=7 TO 0 STEP -1: PRINT AT n,b;

```

```

INK 1;"K": BEEP .01,i*7: NEXT i:
PRINT AT n,b;" ": NEXT n
430 IF a#="*" OR a#="0" THEN BE
EP .03,10: LET n=0: RETURN
440 PRINT AT n,b; ("f" AND n>5):
BEEP .01,n
450 PRINT AT n,b; (" " AND n>5):
NEXT n: RETURN
500 IF b=30 THEN RETURN
505 FOR n=b+1 TO 30
510 LET a#=SCREEN# (a,n)
520 IF a#="K" THEN LET lev=lev-
1: PRINT AT 3,20;lev;" ": LET a
c=c+100: PRINT AT 1,20;cc: FOR
i=7 TO 0 STEP -1: PRINT AT a,n:
INK 1;"K": BEEP .01,i*7: NEXT i:
PRINT AT a,n;" ": NEXT n
530 IF a#="*" OR a#="0" THEN BE
EP .03,10: LET n=30: RETURN
540 PRINT AT a,n; ("e" AND n<31):
BEEP .01,n
550 PRINT AT a,n; (" " AND n<31):
NEXT n: RETURN
600 IF b=20 THEN RETURN
605 FOR n=b+1 TO 20
610 LET a#=SCREEN# (a,n)
620 IF a#="K" THEN LET lev=lev-
1: PRINT AT 3,20;lev;" ": LET a
c=c+100: PRINT AT 1,20;cc: FOR
i=7 TO 0 STEP -1: PRINT AT a,n:
INK 1;"K": BEEP .01,i*7: NEXT i:
PRINT AT a,n;" ": NEXT n
630 IF a#="*" OR a#="0" THEN BE
EP .03,10: LET n=20: RETURN
640 PRINT AT a,n; ("f" AND n<21):
BEEP .01,n
650 PRINT AT a,n; (" " AND n<21):
NEXT n: RETURN
700 IF b=1 THEN RETURN
705 FOR n=b+1 TO 1 STEP -1
710 LET a#=SCREEN# (a,n)
720 IF a#="K" THEN LET lev=lev-
1: PRINT AT 3,20;lev;" ": LET a
c=c+100: PRINT AT 1,20;cc: FOR
i=7 TO 0 STEP -1: PRINT AT a,n:
INK 1;"K": BEEP .01,i*7: NEXT i:
PRINT AT a,n;" ": NEXT n
730 IF a#="*" OR a#="0" THEN BE
EP .03,10: LET n=1: RETURN
740 PRINT AT a,n; ("e" AND n>0):
BEEP .01,n
750 PRINT AT a,n; (" " AND n>0):

```

```

NEXT n: RETURN
801 LET b=2: FOR n=20 TO 5 STEP
-1
810 BEEP .02,n*2: PRINT AT n,0:
INK 5;" "
820 NEXT n
830 PRINT AT 3,31: INK 5;" ": P
RINT AT 21,1: INK 5;" ": RETURN
1000 PRINT AT 10,4: PAPER 7: INK
0: BRIGHT 0;"Voce pontuou "sc)
1010 BEEP .1,24: BEEP .1,23: BEE
P .04,24: BEEP .5,21: BEEP .5,20:
BEEP .05,21
1500 IF cc<=n(10) THEN PAUSE 50:
GO TO 1040
1502 LET num=11: IF cc>=n(num) T
HEN INPUT "Indique o indicial:"
;p#: IF LEN p#>3 THEN GO TO 1505
1506 IF cc>=n(num) THEN LET n(nu
m)=cc: LET n#(num)=p#
1510 FOR a=1 TO (num-1): LET b#=
n#(a): LET c#=n#(a+1): LET b=n(a
): LET c=n(a+1): IF b<c THEN LET
n(a)=c: LET n(a+1)=b: LET n#(a)
=c#: LET n#(a+1)=b#
1520 NEXT a: FOR n=1 TO num-1: I
F n(n)<n(n+1) THEN GO TO 1510
1530 NEXT n
1540 CLS
1550 PRINT AT 2,7;"PONTUACoes MA
XIMAS"
1560 PRINT AT 3,7;"=====
====="
1565 FOR n=1 TO num-1: PRINT AT
n+5,7: INK 6; "(" : INK 2;n: INK 5
;" " : AT n+5,10: INK 7;n(n): PRIN
T AT n+5,17: INK 5;n#(n): NEXT n
1590 PLOT 0,0: DRAW 55,0: DRAW 0
,175: DRAW -255,0: DRAW 0,-175:
PRINT AT 17,0: PAPER 2: INK 7;"O
brigue num tecla.": LET i=0
1600 LET i=i+1: IF i>7 THEN LET
i=0
1610 PRINT AT 2,7: INK 1;"PONTUA
COES MAXIMAS"
1620 BEEP .01,i*7: PAUSE 2: IF I
NKEY#="" THEN GO TO 1600
1630 GO TO 10
2000 BORDER 1: PAPER 0: INK 7: B

```

SONDA ESPACIAL

```

DIGIT 1: CLS
0000 PRINT AT 0,0: "S P A C E T
0001 PRINT AT 0,0: "#####
0002 PRINT AT 0,1: "Comande a sua
0003 deve cobrir as fendas de cur
0004 sor e do para as fendas de cur
0005 ativo do para as fendas de cur
0006 e de do para as fendas de cur
0007 e de do para as fendas de cur
0008 e de do para as fendas de cur
0009 e de do para as fendas de cur
0010 e de do para as fendas de cur
0011 e de do para as fendas de cur
0012 e de do para as fendas de cur
0013 e de do para as fendas de cur
0014 e de do para as fendas de cur
0015 e de do para as fendas de cur
0016 e de do para as fendas de cur
0017 e de do para as fendas de cur
0018 e de do para as fendas de cur
0019 e de do para as fendas de cur
0020 e de do para as fendas de cur
0021 e de do para as fendas de cur
0022 e de do para as fendas de cur
0023 e de do para as fendas de cur
0024 e de do para as fendas de cur
0025 e de do para as fendas de cur
0026 e de do para as fendas de cur
0027 e de do para as fendas de cur
0028 e de do para as fendas de cur
0029 e de do para as fendas de cur
0030 e de do para as fendas de cur
0031 e de do para as fendas de cur
0032 e de do para as fendas de cur
0033 e de do para as fendas de cur
0034 e de do para as fendas de cur
0035 e de do para as fendas de cur
0036 e de do para as fendas de cur
0037 e de do para as fendas de cur
0038 e de do para as fendas de cur
0039 e de do para as fendas de cur
0040 e de do para as fendas de cur
0041 e de do para as fendas de cur
0042 e de do para as fendas de cur
0043 e de do para as fendas de cur
0044 e de do para as fendas de cur
0045 e de do para as fendas de cur
0046 e de do para as fendas de cur
0047 e de do para as fendas de cur
0048 e de do para as fendas de cur
0049 e de do para as fendas de cur
0050 e de do para as fendas de cur
0051 e de do para as fendas de cur
0052 e de do para as fendas de cur
0053 e de do para as fendas de cur
0054 e de do para as fendas de cur
0055 e de do para as fendas de cur
0056 e de do para as fendas de cur
0057 e de do para as fendas de cur
0058 e de do para as fendas de cur
0059 e de do para as fendas de cur
0060 e de do para as fendas de cur
0061 e de do para as fendas de cur
0062 e de do para as fendas de cur
0063 e de do para as fendas de cur
0064 e de do para as fendas de cur
0065 e de do para as fendas de cur
0066 e de do para as fendas de cur
0067 e de do para as fendas de cur
0068 e de do para as fendas de cur
0069 e de do para as fendas de cur
0070 e de do para as fendas de cur
0071 e de do para as fendas de cur
0072 e de do para as fendas de cur
0073 e de do para as fendas de cur
0074 e de do para as fendas de cur
0075 e de do para as fendas de cur
0076 e de do para as fendas de cur
0077 e de do para as fendas de cur
0078 e de do para as fendas de cur
0079 e de do para as fendas de cur
0080 e de do para as fendas de cur
0081 e de do para as fendas de cur
0082 e de do para as fendas de cur
0083 e de do para as fendas de cur
0084 e de do para as fendas de cur
0085 e de do para as fendas de cur
0086 e de do para as fendas de cur
0087 e de do para as fendas de cur
0088 e de do para as fendas de cur
0089 e de do para as fendas de cur
0090 e de do para as fendas de cur
0091 e de do para as fendas de cur
0092 e de do para as fendas de cur
0093 e de do para as fendas de cur
0094 e de do para as fendas de cur
0095 e de do para as fendas de cur
0096 e de do para as fendas de cur
0097 e de do para as fendas de cur
0098 e de do para as fendas de cur
0099 e de do para as fendas de cur

```

Trata-se de um jogo fascinante onde, uma vez mais, a nossa missão consiste em patrulhar um sector da galáxia. Como terá adivinhado, existem alienígenos que você deve encontrar e destruir.

O programa mostra um mapa do sector galáctico onde você se encontra, juntamente com as suas características mais relevantes em termos de jogo. Depressa aprenderá a ler este mapa, à medida que for executando o jogo.

Em qualquer momento pode mover-se, tentar detectar inimigos ou disparar sobre eles. Os seus detectores actuam de duas formas diferentes. O detector de curto raio de acção, que investiga os oito quadrados contíguos àquele onde a sua nave se encontra, consome pouca energia. Os detectores de longo alcance investigam dois quadrados numa única direcção, e gastam uma maior quantidade de energia.

O jogo termina se você aterrizar em cima de uma nave alienígena... Os inimigos não se deslocam dos mesmos sectores durante cada jogo. Você dispõe de reservas de energia limitadas, e deve tentar eliminar tantos alienígenos quanto possível antes de perder completamente a energia.

O programa inventa nomes diferentes para os alienígenos de cada vez que é executado. Cada um destes só pode disparar contra si depois de você próprio usar o laser sobre ele (revelando assim a posição em que se encontra), e apenas no caso de se encontrar a uma distância de um único quadrado. Os danos provocados na sua nave por um tiro inimigo são indicados em termos de energia (ou seja, cada tiro eficaz drena as baterias da sua nave). O jogo continua até você abater uma nave inimiga ou ficar sem energia. Tim Rogers converteu este jogo, a partir de um original meu.

```

10 BORDER 0: RANDOMIZE
20 INK 0
30 PAPER 0
40 CLS
50 GO SUB 2000: GO SUB 1000
70 GO SUB 3000
110 GO TO 70
1010 DIM a(10,10)
1020 DIM b(10,10)
1050 FOR a=1 TO 20
1060 LET g(FN a(x),FN a(x))=1
1100 NEXT a
1110 LET g(v,p)=2

```

```

1100 LET s(0,0)=0
1110 LET t0=TN a(0)
1120 LET t1=TN a(0)
1130 LET TN=TN a(0)
1140 PRINT LN
1150 LET v=TN a(00)
1160 PRINT v, s(1,0)
1170 PRINT v, s(1,0)
1180 LET LN
1190 LET TN a(x)=INT (RND*x)+1
1200 LET TN a(x)=x#(0), TO t(0),
1210 # (0), TO t(0), 0)
1220 LET x=10
1230 LET v=0
1240 LET s="": REM grafico
1250 DIM c#(0,0)
1260 FOR a=1 TO 5
1270 READ c#(a)
1280 NEXT a
1290 LET t=0
1300 DIM s(0): DIM c(0)
1310 FOR a=1 TO 5
1320 READ v(a): READ c(a)
1330 NEXT a
1340 LET s=10000+00000RND
1350 DIM x#(10,0)
1360 DIM c#(10,0)
1370 DIM t#(0,0)
1380 FOR a=1 TO 10
1390 READ c#: LET t(a,1)=LEN c#
1400 LET x#(a)=0#
1410 READ c#: LET t(a,2)=LEN c#
1420 LET c#(a)=0#
1430 NEXT a
1440 DIM s#(10,11)
1450 FOR a=1 TO 10
1460 LET t0=TN a(0)
1470 LET t1=TN a(0)
1480 LET v#(a)=TN v#(0)
1490 NEXT a
1500 FOR a=1 TO 5
1510 FOR b=0 TO 7
1520 READ c
1530 POKE USA CHR# (143+a)+b,c
1540 NEXT b: NEXT a
1550 RETURN
1560 OFS
1570 PRINT AT 0,10;"MAPA ";
1580 GO SUB 1000
1590 PRINT "MOTELAR"
1600 GO SUB 1000

```

```

00400 FOR a=1 TO 32
00100 PRINT "*"
00200 NEXT a
00300 SUB 1000
00400 PRINT AT 3,0;" 1034567000"
00500 FOR a=1 TO 10
00600 IF a<10 THEN PRINT a: GO T
00700 PRINT "0"
00800 PRINT s(1,0)
00900 LET v#=(s(a,0)+1)
01000 PRINT INCREASER(1);v#;
01100 NEXT a
01200 IF v#<>" " THEN GO SUB 1000
01300 IF a<10 THEN PRINT a: GO TO
01400 PRINT "0"
01500 PRINT a
01600 PRINT " 1034567000"
01700 PRINT AT 0,14; INK 0; PAPER
01800 FLASH 1;"Baterias:";INT a
01900 PRINT AT 0,14;
02000 FOR a=1 TO 5/200
02100 PRINT INK 5;"█");
02200 NEXT a
02300 SUB 1000
02400 IF a<1 THEN GO TO 4000
02500 IF a>0 THEN PRINT AT 0,14;
02600 "Have a nice day";AT 0,14;"destru
02700 t";AT 00 SUB 1000
02800 PRINT AT 10,14;"Have locali
02900 PRINT AT 11,14;"do subquest
03000 PRINT AT 12,14; FLASH 1;"1"
03100 IF "p";"4"
03200 GO SUB 1000
03300 PRINT AT 10,14;"do sector
03400 AT 14,14;c#(INT (c#*10+.5))
03500 PRINT AT 10,0;
03600 GO SUB 1000
03700 GO TO 4000
03800 GO SUB 1000
03900 PRINT FLASH 1;"Baterias com
04000 olete baterias"
04100 PRINT at;"animacoes mortos"
04200 PRINT ,c#(TN a(0));": desce
04300 e o que";"comandante de novo?"
04400 IF INKEY#<>" " THEN GO TO 41
04500 IF INKEY#="" THEN GO TO 410
0

```



```

4140 IF INKEY#<>"n" AND INKEY#<>
"z" THEN RUN
4160 PRINT "c#(FN a(5))": voce
401 BEBPE
4170 PRINT "UB pouco "
4180 LET a=FN a(0)
4190 LET a=FN a(0)
4200 PRINT FN a#(0)
4210 PRINT a#(0)
4220 GO SUB 9999
4230 PRINT AT 16,0;c#(FN a(5)):"
- Or de de, comendante?"
4240 BEBPE .1,AND#50
4250 PRINT AT 17,10;"1 Deteccao"
4260 GO SUB 1599
4270 PRINT TAB 12;"2 Movimento"
4280 GO SUB 1599
4290 PRINT TAB 12;"3 Disparar"
4300 GO SUB 1599
4310 PRINT AT 21,0;"Carregue 1,
2, 3 de sub comendante"
4320 LET i#=INKEY#
4330 IF CODE i#<40 OR CODE i#>51
THEN GO TO 4399
4340 GO SUB 1599
4350 GO SUB 1599
4360 IF i#="1" THEN GO TO 5000
4370 GO SUB 5999+1599*VAL i#
4380 FOR a=0 TO 20
4390 PRINT AT 21,a; "#####"
4400 BEBPE .01,a+10
4410 BEBPE .01,a
4420 PRINT AT 21,a;" "
4430 NEXT a
4440 PRINT AT 21,a;" "
4450 RETURN
4460 PRINT AT 16,0;
4470 FOR a=1 TO 0
4480 PRINT " "
5000 NEXT a
5040 PRINT AT 16,0;
5050 RETURN
5060 PRINT TAB 10; FLASH 1; INK
7;" DETECCAO "
5010 PRINT (c#(FN a(5)));"-"
5020 PRINT "(Carregue (2) ou pre
xima (1))?"
5030 LET i#=INKEY#
5040 BEBPE .01,0
5050 IF CODE i#<40 OR CODE i#>50
THEN GO TO 5030

```

```

5060 LET k=VAL i#
5070 LET a=a-10*k
5080 IF k=0 THEN GO TO 5500
5090 PRINT AT 16,0;" Deteccao.."
5100 LET f=0
5110 FOR a=- (0>1) TO (0<10)
5120 FOR b=- (0>1) TO (0<10)
5130 BEBPE .01,0
5140 IF a (0+a,0+b)=1 THEN LET f=
1
5150 NEXT b: NEXT a
5160 IF f=1 THEN PRINT "Nave ";Z
";" de vizinhanca!": GO TO 5160
5170 PRINT "Nave de vizinhanca."
5180 GO SUB 1599
5190 PRINT AT 21,0;"Desligar det
eccao?"
5200 BEBPE .1,AND#50
5210 IF INKEY#="" THEN GO TO 517
0
5220 IF INKEY#="n" OR INKEY#="N"
THEN GO SUB 5999: GO TO 5010
5230 GO SUB 1599
5240 GO TO 4999
5250 PRINT AT 16,0;c#(FN a(5)):"
: Deteccao?"
5260 PRINT "Norte:1 Este:2 Sul
:3 Oeste:4"
5270 IF INKEY#<>" " THEN GO TO 55
10
5280 LET i#=INKEY#
5290 BEBPE .01,0
5300 IF CODE i#<40 OR CODE i#>50
THEN GO TO 5520
5310 LET k=(VAL i#-1)*2+1;
5320 LET z=(a(0+a(k)*2,0+c(k)*2)
=1)
5330 IF z=1 THEN PRINT FLASH 1;"
positivo": GO TO 5500
5340 PRINT "negativo"
5350 GO SUB 1599
5360 IF INKEY#<>" " THEN GO TO 55
10
5370 GO TO 5160
5380 LET f1=0
5390 LET a=a-50
5400 PRINT "Rumo:"
5410 GO SUB 1599
5420 CIRCLE 40,24,10
5430 PLOT 40,24
5440 DRAW 0,10

```

```

7050 GO SUB 1500
7060 PRINT AT 10,10,"000"
7080 PLOT 40,04
7090 ORDU 10,09
7100 GO SUB 1500
7120 PRINT AT 10,10,"000"
7130 PLOT 40,04
7140 ORDU 0,10
7150 GO SUB 1500
7160 PRINT AT 10,10,"100"
7180 PLOT 40,04
7190 ORDU 10,09
7200 GO SUB 1500
7210 PRINT AT 10,14,"070"
7220 HNDUT "Grace?"
7240 IF >010 OR <0 THEN GO TO
7250 LET r=INT (r/45)+1
7260 IF r=1 THEN RETURN
7270 LET a(C,P)=0
7270 LET a(C,P)=0
7280 LET u=C+B(I)
7280 LET p=C+D(I)
7300 IF a(C,P)<>1 THEN GO TO 111
0
7310 PRINT AT AND*21,0; FLASH 1;
INK 0; PAPER 0;"Voice uttered by
#a bve "N#;"
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
7320 ORDU 1,0
7330 IF AND>.00 THEN GO TO 7310
7340 FOR a=1 TO AND*5
7340 PLOT 100,07
7350 ORDU 100-AND*200,07-AND*170
7360 NEXT a
7370 ORDU 0,10
7380 GO TO 7050
0000 LET r=1
0010 PRINT AT 14,14; INK 0; PAPER
0; FLASH 1;"ALPHA"
0020 GO SUB 1500; GO SUB 1500
0030 PRINT AT 10,0;C#(FN a(5));"
- Dirccao de tiro:"
0040 GO SUB 7010
0040 LET a=C+D(I)
0050 IF a(C+B(I),C+D(I))<>1 THEN
GO TO 0000
0070 ORDU 1,00
0080 PRINT AT 15,0; FOR a=1 TO
7: PRINT
": NEXT a: PRINT AT 15
,0;

```

```

0000 PRINT C#(FN a(5));" - Atine
F#000 0 "N#;"
00000 LET a(C+B(I),C+D(I))=4
00007 LET a(C+B(I),C+D(I))=4
00100 LET a=C+D(I)
00100 PRINT C#(FN a(5));" - Desli
0000 alerta?"
00100 IF INKEY#<>"" THEN GO TO 01
00
00100 IF INKEY#="" THEN GO TO 010
00
00140 GO TO 1110
00000 GO SUB 0000
00010 PRINT C#(FN a(5));" ";
00000 LET a(C+B(I),C+D(I))=0
00000 PRINT "Falhas#00."
00040 FOR a=0 TO 00
00000 ORDU 1,a
00000 NEXT a
00070 ORDU 1,00
00000 FOR a=1 TO 00
00000 ORDU a AND*7
00400 NEXT a
00410 ORDU a
00400 PRINT C#(FN a(5));" De "
N#;" estado "a responder ao tir
0110
00400 GO SUB 1500
00450 IF AND>.0 THEN GO TO 0700
00460 PRINT "ATINGIRIM-NOB!!!"
00470 LET a=-100*AND
00480 GO TO 0410
0700 PRINT C#(FN a(5));" Mas fe
lizante os"
0710 PRINT a N#;" falaram."
0720 GO TO 0040
0000 ORDU "Dr Gock", "Scottie", "T
. Logo", "Chackout", "Nulu"
00000 ORDU 1,0,1,1,0,1,1,1,0,
1,1,0,1,1,1
0040 ORDU "Golan", "Frax",
"florida", "Hob", "Jax", "Joc", "Ulen
", "Grip", "Lar", "Kre", "Joc", "Joc", "Joc"
00000 ORDU 200,100,100,100,10
00,100,0000
00000 ORDU 200,100,210,250,250,21
00,100,0000
00700 ORDU 240,100,100,250,47,57,
040,200

```

TEMPESTADE LUNAR

Este programa, escrito por David Perry, utiliza métodos sofisticados de tratamento de cadeias alfanuméricas para obter os meteoros e naves que se movem lateralmente no visor.

Por baixo encontra-se a superfície do planeta McClariana (que deve o seu nome ao intrépido explorador McClaren que o descobriu em 2068). Você deve tentar descer através destes obstáculos, e aterrizar numa das três pistas.

Para simplificar a sua tarefa, extremamente difícil, você dispõe de um botão que regula o impulso dos motores permitindo-lhe manter-se na mesma posição quando a confusão por baixo é demasiado grande. O jogo utiliza bastante o som e a cor, além de alguns interessantes gráficos definidos pelo autor.

TEMPESTADE
LUNAR

Indique nivel (1 a 3)

=====

0-Esquerda 0-Direita

0-Impulso

Tente aterrizar nas pistas:

1 0000
2 0000
3 0000
4 0000
5 0000
6 0000
7 0000

```

+-----+
| TEMPESTADE LUNAR |
| Copyright:       |
| DAVID PERRY     |
| 1988            |
+-----+
    
```

LS 0 BORDER 0: PAPER 0: INK 7: 0

0 LET sc=0: LET sc=100: LET i
=7: GO TO 45
10 CLS : RESTORE : GO SUB 20

```

11 LET X=0: LET Y=INT (RND*30)
+1
12 PRINT AT X,Y: " ": PRINT AT
3,0: INK 9:sc#:AT 7,0:sc#:AT 10,0:
sc#:AT 15,0:sc#:AT 18,0:sc#
13 PRINT AT 5,0: INK 9:sc#:AT 9
,0:sc#:AT 13,0:sc#:AT 17,0:sc#
14 IF INT X>10 THEN GO TO 70
15 IF SCREEN# (X,Y)<>" " THEN
GO TO 3: PRINT AT X,Y:sc#: REM 9
REM 10 0
16 BEEP .01,X*3
17 PRINT AT X,Y: " "
18 LET X=X+.5: IF IN 01400=204
THEN LET X=X-.5: LET SC=SC-1
19 LET Y=Y+(INKEY#="0")-(INKEY
#="0")
20 IF Y<0 THEN LET Y=0
21 IF Y>30 THEN LET Y=30
22 IF X>40 THEN GO TO 70
23 IF SCREEN# (X,Y)<>" " THEN
GO TO 30
24 LET SC=SC+1
25 PRINT AT X,Y: INK 7: BRIGHT
0: REM 0:OFFICE 0
26 LET sc#=#(2 TO 1)+sc#(1)
27 LET sc#=#(31)+sc#( TO 30)
28 GO TO 10
29 PRINT AT 20,0: INK 2:"=====
=====
=====
30 PRINT AT 10,0: INK 3:"=====
FLASH 1:"": FLASH 0: INK 7:""
: INK 3: FLASH 1:"": FLASH 0:""
FLASH 0:"": INK 3: FLASH 1:""
FLASH 0:"": INK 3: FLASH 1:""
FLASH 0: INK 7:"": INK 3: FLA
SH 1:"": FLASH 0:""
31 FOR n=1 TO 3: READ c#: FOR
n=0 TO 7: READ b: POKE USR c#+n,
b: NEXT n: NEXT n
32 DATA "c",24,36,00,165,210,2
40,100,100
33 DATA "a",0,60,126,171,210,1
20,00,0
34 DATA "b",105,100,00,126,126
00,100,100
35 RETURN
36 FOR n=7 TO 0 STEP -1
37 PRINT AT X,Y: INK n:"#": RE
M 0:OFFICE 0
38 BEEP .01,n*3: BEEP .01,50-n
+3
    
```

```

39 NEXT D
40 LET i = 7
41 PRINT AT 2,8; INK 5;"PONTUA
CAC: "13c
42 PRINT INK 1; BRIGHT 8; AT 6,
4;"Carregue em qualquer teclad!!!
43 LET i=i-1: IF i<0 THEN LET
i=7
44 IF INK#="" THEN GO TO 42
45 LET sc=8: CLS : LET d=8: BR
IGHT 1: INK 5
46 PRINT "
47 PRINT "
48 PRINT "
49 PRINT "
50 PRINT "
51 PRINT "
52 PRINT : INK 5: BRIGHT 8
53 PRINT "
54 PRINT "
55 PRINT "
56 PRINT "
57 PRINT "
58 PRINT "
59 PRINT AT 15,8; INK 1;"India
ue Nivel (1 e 3)
60 PRINT AT 15,8; INK 1/2;"===
=====
61 PRINT AT 19,3; INK 4; INK 7
;"5- Esquerda      0-Direita"
62 PRINT AT 19,11; INK 4;"0-Im
pulso"
63 PRINT AT 20,4; INK 6;"Tente
ateerar nas pistas"
64 PRINT #1; BRIGHT 8; INK 7;"
TEMPESTADE LUNAR © D. PERRY 1983
65 PRINT AT 15,8; INK 1;"India
ue Nivel (1 e 3)"

```

```

67 LET @#="INKEY#": IF @#<"1" OR
@#>"0" THEN GO TO 70
68 IF @#="" THEN GO TO 76
69 IF @#="1" THEN LET @#=""
70 IF d=1 THEN LET @#=""
71 IF @#="0" THEN LET @#=""
72 IF d=0 THEN LET @#=""
73 IF @#="3" THEN LET @#=""
74 IF d=3 THEN LET @#=""
75 GO TO 18
76 LET i=i-1: IF i<0 THEN LET
i=7
77 GO TO 66
78 STOP
79 PRINT AT x,y; INK 7; BRIGHT
8;" " : IF y=4 THEN LET sc=150:
GO TO 83: REM grafica C
80 IF y=16 THEN LET sc=150: GO
TO 83
81 IF y=20 THEN LET sc=150: GO
TO 83
82 GO TO 36
83 LET sc=sc+50: FOR n=1 TO 3:
FOR i=0 TO 7
84 PRINT AT 1,7; INK 1;"P A B
P 0 E 2 S 1"
85 PRINT AT 3,7;"Bonus: ";sc;"
86 PRINT AT 6,11;"Pontuacao: "
13c
87 NEXT i: NEXT D
88 FOR n = 80 TO 99 STEP 5: SEE
P 0 E 2 S 1: STDP ,001,50-ABS (5):
STDP ,005,n/2: NEXT D
89 CLS : GO TO 18

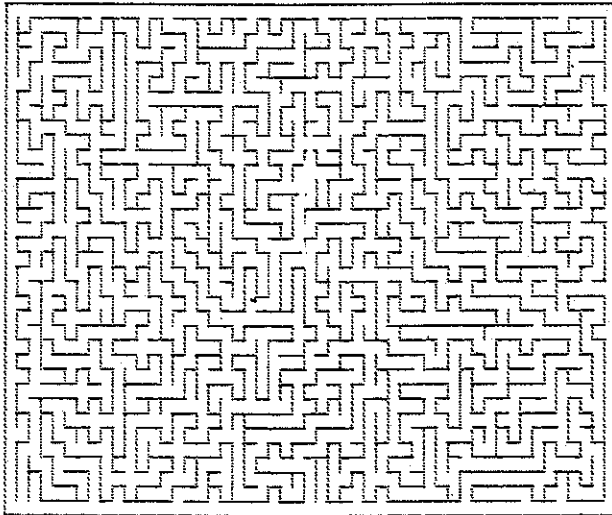
```

O PRAZER DO LABIRINTO

GERADOR DE LABIRINTOS

Vamos apresentar agora uma série de programas interessantes sobre labirintos, a maior parte dos quais escritos por Graham Charlton.

Começamos por um programa que desenha um labirinto de quaisquer dimensões, como o seguinte:



Quando o programa o pedir, indicamos um número qualquer entre 2 e 23 a fim de definir a complexidade do labirinto obtido. Notar que a produção do labirinto é demorada.

```

10 DIM B#(23)
20 DIM D#(23)
30 INPUT "NU#(2 a 23) ? " N#
40 LET T# = INT (255 / N#) # B
45 LET W# = INT (175 / N#) # B
50 PLOT 0,0
60 DRAW 1,0
70 DRAW 0,1
80 DRAW 1,1
90 DRAW 0,1 #
100 LET X# = 0
110 LET Y# = 0
120 LET N# = 1
130 PLOT X#,Y#
140 IF (POINT (X#+B,Y#)+POINT (X#,
Y#)+POINT (X#,Y#+B)+POINT (X#-B,Y#)
) << 4 THEN GO TO 200
200 LET Z# = Z# - 1
210 IF Z# = 0 THEN STOP
220 LET X# = INT B#(N#)
230 LET Y# = INT D#(N#)
240 GO TO 100
250 LET B#(N#) = INT B# * X#
260 LET D#(N#) = INT D# * Y#
270 LET N# = N# + 1
280 LET R# = INT (RND * 4)
290 LET C## = ((R# = 0) - (R# = 1))
300 LET D## = ((R# = 2) - (R# = 3))
310 IF POINT (X#+C#,Y#+D##) THEN GO
TO 200
320 DRAW C#,D#
330 LET X# = X# + C#
340 LET Y# = Y# + D#
350 GO TO 100

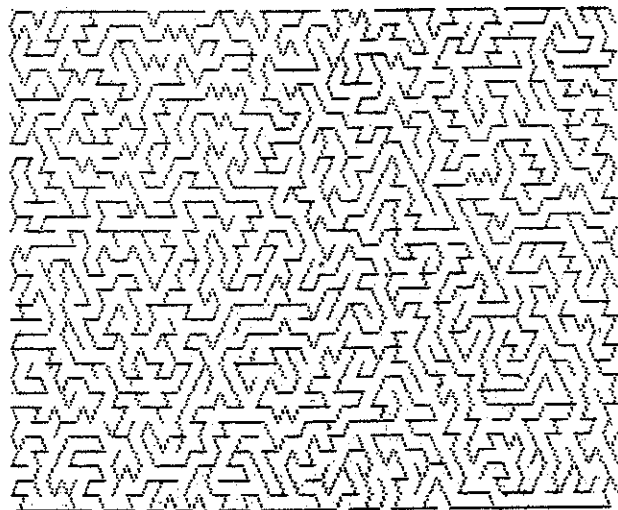
```



```

60  DD DDU t,0
70  DD DDU 0,w
80  DD DDU t,w
90  DD DDU 0,w
100  LET X=0
110  LET Y=0
120  LET N=1
130  PLOT X,Y
140  IF POINT (X+0,Y+0)+POINT (
X+0,Y-1)+POINT (X-1,Y+0)+POINT (
X-1,Y-1)<>4 THEN GO TO 200
150  LET N=N-1
160  IF N=0 THEN STOP
170  LET X=00000R R#(N)
180  LET Y=00000R R#(N)
190  GO TO 100
200  LET R#(N)=CHD# X
210  LET R#(N)=CHD# Y
220  LET N=N+1
230  LET I=INT (RND*4)
240  LET C=R#((I<2)-(I>1))
250  LET D=R#((I=0)+(I=3)-(I=1)-
(I=2))
260  IF POINT (X+C,Y+D) THEN GO
TO 200
270  DD DDU C,D
280  LET X=X+C
290  LET Y=Y+D
300  GO TO 100

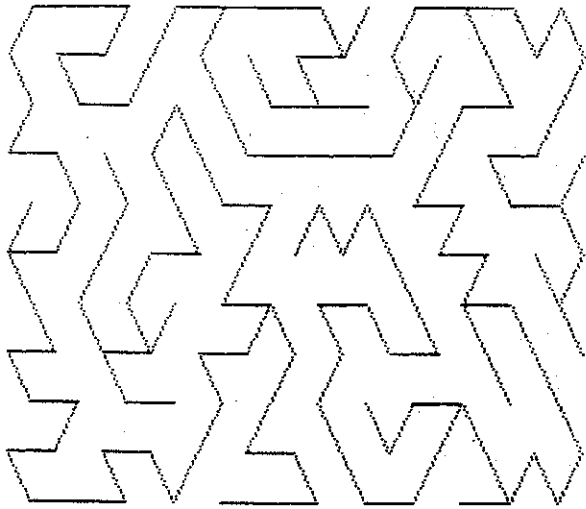
```



```

10  DD DDU R#(00000)
20  DD DDU R#(00000)
30  LET X=10000R#(0)
40  LET Y=00000R#(0)
50  LET N=1
60  INPUT "Numero (R = 20)? ";S
70  LET t=S#10
80  LET a=INT (00R (t+t-S*6))
90  PLOT X,Y
100  FOR c=-t TO t STEP t*2
110  IF X+C>255 OR X+C<0 THEN GO
TO 200
120  IF POINT (X+C,Y)=0 THEN GO
TO 300
130  NEXT C
140  FOR d=-S TO S STEP t
150  IF X+C>255 OR X+C<0 THEN GO
TO 310
160  FOR d=-4 TO 4 STEP 4*2
170  IF Y+D>175 OR Y+D<0 THEN GO
TO 300
180  IF POINT (X+C,Y+D)=0 THEN GO
TO 300
190  NEXT D
200  NEXT C
210  LET N=N-1
220  IF N=0 THEN STOP
230  LET X=00000R R#(N)
240  LET Y=00000R R#(N)
250  GO TO 100
260  LET R#(N)=CHD# X
270  LET R#(N)=CHD# Y
280  LET N=N+1
290  LET I=INT (RND*6)
300  LET C=R#((I=1)+(I=5)-(I=0)-
(I=4))+I*((I=0)-(I=2))
310  LET D=R#((I<2)-(I>3))
320  IF X+C>255 OR X+C<0 THEN GO
TO 300
330  IF Y+D>175 OR Y+D<0 THEN GO
TO 300
340  IF POINT (X+C,Y+D) THEN GO
TO 300
350  DD DDU C,D
360  LET X=X+C
370  LET Y=Y+D
380  GO TO 100

```



MUROS DO SUSPENSE

Neste programa é necessário competir com dois opositores, controlados pelo computador, na tentativa de resolver o labirinto. Os dois opositores são representados por um «O», e você por um*. Deve usar as teclas de cursor, e partirá do canto superior esquerdo tentando atingir o canto inferior direito.

Ganha quem primeiro resolver o labirinto.

```

5 CLS : PRINT TAB 7,"Aguarde
um momento"
10 DIM c$(23,23)
15 DIM b$(1000)
20 DIM v$(1000)
30 LET x=0
40 LET y=17
50 LET N=1
60 FOR #=N TO 20
70 LET c#(0)=" "
#####
75 NEXT #
100 LET c$(x,y)=" "
110 IF c$(x+1,y)="#" OR c$(x-1,

```

```

y)="#" OR c$(x,y+1)="#" OR c$(x,
y-1)="#" THEN GO TO 1000
120 LET N=N+1
130 IF N=5 THEN GO TO 2000
140 LET x=10000#b$(N)
150 LET y=10000#v$(N)
160 GO TO 100
1000 LET b$(N)=CHR$(x)
1010 LET v$(N)=CHR$(y)
1020 LET N=N+1
1030 LET r=INT (RND*4)
1040 LET c=(r=0)-(r=1)
1050 LET d=(r=2)-(r=3)
1060 IF c$(x+r*c,y+d*d)="#" THEN
GO TO 1000
1070 LET c$(x+c,y+d)="#"
1080 LET c$(x+r*c,y+d*d)="#"
1090 LET x=x+r*c
1100 LET y=y+d*d
1110 GO TO 100
2000 CLS : FOR #=2 TO 20
2010 PRINT INK 1;c#(0,0 TO )
2020 NEXT #
2030 PRINT AT 10,20,"X"
3000 LET x=1
3005 LET y=1
3010 LET u=1
3015 LET v=1
3020 LET s=0
3025 LET t=0
3100 PRINT AT x,y;"*"
3110 LET N#=INKEY#
3120 LET c=(N#="0")-(N#="7")
3130 LET d=(N#="8")-(N#="5")
3140 IF SCREEN$(x+c,y+d)="#" TH
EN BRAP .1,0: GO TO 3000
3170 PRINT AT x,y; INK 0;"■"
3180 LET x=x+c
3190 LET y=y+d
3200 PRINT AT x,y;"*"
3210 IF x=10 AND y=20 THEN INPUT
="De se sair outro jogo? ";v#: IF v#
="s" OR v#="0" THEN RUN
3211 IF x=10 AND y=20 THEN STOP
3260 LET s=s-1
3700 IF s=-1 THEN LET s=3
3900 PRINT AT v#,a; INK 0;"■"
4000 IF s=# THEN LET s=0
4010 LET c=(s=1)-(s=3)
4020 LET d=(s=5)-(s=2)

```

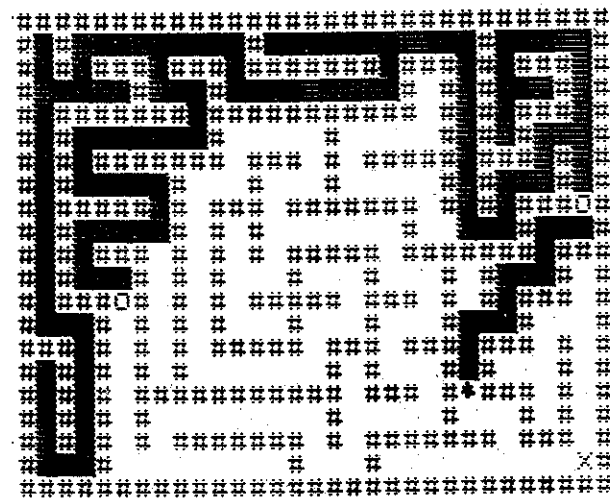

MURO DO SUSPENSE

Nesta versão do jogo, você enfrenta apenas um único oponente, igualmente controlado pelo computador. No entanto, este é bastante mais inteligente do que os do jogo anterior, e move-se ao dobro da velocidade a que você se pode deslocar.

```

4000 IF @CORREN# (P+C,A+4)="#" TH
EN LET S=S+10: GO TO 4050
4050 LET P=P+C
4060 LET A=A+4
4065 PRINT AT P,A;"0"
4070 IF P=10 AND A=20 THEN INPUT
="Desse lado outro jogo?" V# : IF V#
="#" OR A#="S" THEN RUN
4072 IF P=10 AND A=20 THEN STOP
0000 LET T=T-1
0070 IF T=1 THEN LET T=0
0080 PRINT AT U,V;INK 6;"#"
0090 IF T=4 THEN LET T=0
0010 LET C=(T=8)-(T=0)
0020 LET A=(T=1)-(T=3)
0030 IF @CORREN# (U+C,V+A)="#" TH
EN LET T=T+10: GO TO 0000
0050 LET U=U+C
0060 LET V=V+A
0065 PRINT AT U,V;"0"
0070 IF U=10 AND V=20 THEN INPUT
="Desse lado outro jogo?" V# : IF V#
="#" OR A#="S" THEN RUN
0072 IF U=10 AND V=20 THEN STOP
0080 GO TO 0100

```



```

5 CLS : PRINT TAB 7;"Aguarde
um momento"
10 DIM C#(20,20)
12 DIM B#(1000)
14 DIM G#(1000)
20 LET X=10
30 LET C=17
40 LET N=1
50 FOR #=0 TO 20
60 LET C#(#)= "#####"
#####
70 NEXT #
100 LET C#(X,C)="#"
110 IF C#(X+10,C)="#" OR C#(X-10,
C)="#" OR C#(X,C+10)="#" OR C#(X,
C-10)="#" THEN GO TO 1800
120 LET N=N+1
130 IF N=5 THEN GO TO 2000
140 LET X=CODEM B#(N)
150 LET C=CODEM G#(N)
160 GO TO 1800
1000 LET B#(N)=CHR# X
1010 LET G#(N)=CHR# C
1020 LET N=N+1
1030 LET T=INT (RND*4)
1040 LET C=(T=0)-(T=1)
1050 LET A=(T=2)-(T=3)
1060 IF @C#(X+10*C,C+10*A)="#" THEN
GO TO 1800
1070 LET C#(X+C,C+4)="#"
1080 LET C#(X+10*C,C+10*A)="#"
1090 LET X=X+10*C
1100 LET C=C+10*A
1110 GO TO 1800
2000 CLS : FOR #=0 TO 20
2010 PRINT INK 1;C#(#,0 TO
#####
2020 NEXT #
2030 PRINT AT 10,20;"X"
2040 LET X=1
2050 LET P=1
2060 LET U=1

```



```

0040 PRINT AT 1,0; PAPER 0; INK
0050 (1)
0060 PRINT AT 2,0; INK 1;S#;AT 2
1,0;S#
10000 GO SUB 0000
10005 INK 7
10100 FOR a=4 TO 20 STEP 2: PRINT
AT a,0; "*****"
*****: NEXT a: REM grafico
00
10300 FOR a=1 TO lev: FOR b=0 TO
20 STEP 2
10400 LET b1=INT (RAND*30)+1: PRIN
T AT b,b1; "●"
10500 NEXT b: NEXT a
10600 FOR a=1 TO lev: FOR b=4 TO
20 STEP 2
10700 LET b1=INT (RAND*30)+1: PRIN
T AT b,b1; INK 2; " "
10800 NEXT b: NEXT a
10900 FOR a=0 TO 20: PRINT AT a,0
) OVER 1; INK 2;S#; NEXT a
11000 INK 0: RETURN
10990 GO SUB 0000
10100 LET lives=0: LET t=0
20000 LET t1=21: LET t2=
40
20010 PRINT AT 1,0; PAPER 0; INK
0;T;AT t,c; INK 0; PAPER 7; " "
20020 LET a=USR 30300: IF a=0 THE
N GO TO 2100
20030 IF a=1 THEN IF t<2 THEN IF
SCREEN# ((-1,c)<>"") THEN LET t=t
-1
20040 IF a=2 THEN IF t<21 THEN IF
SCREEN# ((+1,c)<>"") THEN LET t=
t+1
20050 LET c=c+(a=4 AND c<31)-(a=0
AND c>0)
20060 IF SCREEN# (t,c)="" THEN LE
T a=USR 30300: GO TO 2100
20070 IF t<>fin THEN PRINT AT t,c
) INK 1; "X": LET t=t+1: GO TO 20
10: REM grafico 0
20080 IF t1=21 THEN LET t1=2: G
O TO 2010
20070 FOR a=1 TO 2: FOR b=40 TO
60 STEP 10: BEEP .004,a: BEEP .0
64,a-10: BEEP .004,30-a: BEEP .0
64,30-a-10: NEXT a: NEXT b
20090 GO TO 2000
21000 LET lives=lives-1: PRINT AT

```

```

1,0; PAPER 0; INK 0;lives
2110 IF lives<>0 THEN GO TO 2000
2120 PRINT AT 0,0; INK 1;S#;AT 1
,0;S#
2130 PRINT AT 0,0; INK 0;"JOGO T
ERMINADO";AT 10,0;"JOGO TERMINAD
O"
2140 FOR a=1 TO 256: LET b=USR 3
21500: NEXT a
2160 FOR a=1 TO 150: NEXT a: GO
TO 4000
2170 FOR a=1 TO 10: IF t<b(a) TH
EN GO TO 2100
2180 NEXT a: GO TO 4000
2190 FOR b=0 TO a STEP -1: LET h
(b+1)=b(b): LET b#(b+1)=b#(b): N
EXT b
2240 LET h(a)=t: INPUT "Escreva
o seu nome: " LINE 0#: LET h#(a)
=4#: LET t#(a, TO LFN 0#+1)=t#
2250 GO SUB 0000: GO TO 4070
4000 BORDER 0: PAPER 0: CLR: RE
M posicao de titulo
4010 GO SUB 0100: LET ink=0: GO
SUB 4000: REM titulo
4020 FOR c=1 TO 20: GO SUB 4000:
IF CHR# PEEK 23556="0" OR CHR#
PEEK 23556="K" THEN GO TO 4100
4030 NEXT c
4040 GO SUB 5000: REM pontoes
4050 FOR c=1 TO 20: GO SUB 4000:
IF CHR# PEEK 23556="0" OR CHR#
PEEK 23556="K" THEN GO TO 4100
4060 NEXT c
4070 GO SUB 5000: REM pontuacoes
#exiboes
4080 FOR a=1 TO 500: IF CHR# PEK
K 23556="0" OR CHR# PEEK 23556="
K" THEN GO TO 4100
4090 NEXT a: GO TO 4000
4100 LET a=USR 30300: IF CHR# PE
K 23556="0" THEN GO SUB 4000: G
O TO 1000
4110 GO SUB 4200: GO TO 4300
4200 FOR a=1 TO 34: LET b=USR 02
500+USR 30400: NEXT a: RETURN
4300 REM mudar teclas
4310 INK 0: BORDER 0: PAPER 0: C
LR: BORDER 0
4315 POKE 32557,1: POKE 32551,10
: POKE 32559,1: POKE 32559,10
4320 PRINT AT 1,0; INK 0;"DEFINI

```

```

40000 PRINT " INK 1; "
40001 PRINT " INK 0; TAB 11; PAPER
7; "
40002 GOTO 40000
40003 PRINT " INK 0; TAB 11; PAPER
7; "
40004 GOTO 40000
40005 PRINT " INK 0; TAB 11; PAPER
7; "
40006 GOTO 40000
40007 PRINT " INK 0; TAB 11; PAPER
7; "
40008 GOTO 40000
40009 PRINT " INK 0; TAB 11; PAPER
7; "
40010 GOTO 40000
40011 PRINT " INK 0; TAB 11; PAPER
7; "
40012 GOTO 40000
40013 PRINT " INK 0; TAB 11; PAPER
7; "
40014 GOTO 40000
40015 PRINT " INK 0; TAB 11; PAPER
7; "
40016 GOTO 40000
40017 PRINT " INK 0; TAB 11; PAPER
7; "
40018 GOTO 40000
40019 PRINT " INK 0; TAB 11; PAPER
7; "
40020 GOTO 40000
40021 PRINT " INK 0; TAB 11; PAPER
7; "
40022 GOTO 40000
40023 PRINT " INK 0; TAB 11; PAPER
7; "
40024 GOTO 40000
40025 PRINT " INK 0; TAB 11; PAPER
7; "
40026 GOTO 40000
40027 PRINT " INK 0; TAB 11; PAPER
7; "
40028 GOTO 40000
40029 PRINT " INK 0; TAB 11; PAPER
7; "
40030 GOTO 40000
40031 PRINT " INK 0; TAB 11; PAPER
7; "
40032 GOTO 40000
40033 PRINT " INK 0; TAB 11; PAPER
7; "
40034 GOTO 40000
40035 PRINT " INK 0; TAB 11; PAPER
7; "
40036 GOTO 40000
40037 PRINT " INK 0; TAB 11; PAPER
7; "
40038 GOTO 40000
40039 PRINT " INK 0; TAB 11; PAPER
7; "
40040 GOTO 40000
40041 PRINT " INK 0; TAB 11; PAPER
7; "
40042 GOTO 40000
40043 PRINT " INK 0; TAB 11; PAPER
7; "
40044 GOTO 40000
40045 PRINT " INK 0; TAB 11; PAPER
7; "
40046 GOTO 40000
40047 PRINT " INK 0; TAB 11; PAPER
7; "
40048 GOTO 40000
40049 PRINT " INK 0; TAB 11; PAPER
7; "
40050 GOTO 40000
40051 PRINT " INK 0; TAB 11; PAPER
7; "
40052 GOTO 40000
40053 PRINT " INK 0; TAB 11; PAPER
7; "
40054 GOTO 40000
40055 PRINT " INK 0; TAB 11; PAPER
7; "
40056 GOTO 40000
40057 PRINT " INK 0; TAB 11; PAPER
7; "
40058 GOTO 40000
40059 PRINT " INK 0; TAB 11; PAPER
7; "
40060 GOTO 40000
40061 PRINT " INK 0; TAB 11; PAPER
7; "
40062 GOTO 40000
40063 PRINT " INK 0; TAB 11; PAPER
7; "
40064 GOTO 40000
40065 PRINT " INK 0; TAB 11; PAPER
7; "
40066 GOTO 40000
40067 PRINT " INK 0; TAB 11; PAPER
7; "
40068 GOTO 40000
40069 PRINT " INK 0; TAB 11; PAPER
7; "
40070 GOTO 40000
40071 PRINT " INK 0; TAB 11; PAPER
7; "
40072 GOTO 40000
40073 PRINT " INK 0; TAB 11; PAPER
7; "
40074 GOTO 40000
40075 PRINT " INK 0; TAB 11; PAPER
7; "
40076 GOTO 40000
40077 PRINT " INK 0; TAB 11; PAPER
7; "
40078 GOTO 40000
40079 PRINT " INK 0; TAB 11; PAPER
7; "
40080 GOTO 40000
40081 PRINT " INK 0; TAB 11; PAPER
7; "
40082 GOTO 40000
40083 PRINT " INK 0; TAB 11; PAPER
7; "
40084 GOTO 40000
40085 PRINT " INK 0; TAB 11; PAPER
7; "
40086 GOTO 40000
40087 PRINT " INK 0; TAB 11; PAPER
7; "
40088 GOTO 40000
40089 PRINT " INK 0; TAB 11; PAPER
7; "
40090 GOTO 40000
40091 PRINT " INK 0; TAB 11; PAPER
7; "
40092 GOTO 40000
40093 PRINT " INK 0; TAB 11; PAPER
7; "
40094 GOTO 40000
40095 PRINT " INK 0; TAB 11; PAPER
7; "
40096 GOTO 40000
40097 PRINT " INK 0; TAB 11; PAPER
7; "
40098 GOTO 40000
40099 PRINT " INK 0; TAB 11; PAPER
7; "
40100 GOTO 40000

```

```

40049 LET INK=INK+1: IF INK=8 THE
N LET
40050 INK=0: RETURN
40051 PRINT
40052 PRINT
40053 PRINT
40054 PRINT
40055 PRINT
40056 PRINT
40057 PRINT
40058 PRINT
40059 PRINT
40060 PRINT
40061 PRINT
40062 PRINT
40063 PRINT
40064 PRINT
40065 PRINT
40066 PRINT
40067 PRINT
40068 PRINT
40069 PRINT
40070 PRINT
40071 PRINT
40072 PRINT
40073 PRINT
40074 PRINT
40075 PRINT
40076 PRINT
40077 PRINT
40078 PRINT
40079 PRINT
40080 PRINT
40081 PRINT
40082 PRINT
40083 PRINT
40084 PRINT
40085 PRINT
40086 PRINT
40087 PRINT
40088 PRINT
40089 PRINT
40090 PRINT
40091 PRINT
40092 PRINT
40093 PRINT
40094 PRINT
40095 PRINT
40096 PRINT
40097 PRINT
40098 PRINT
40099 PRINT
40100 PRINT

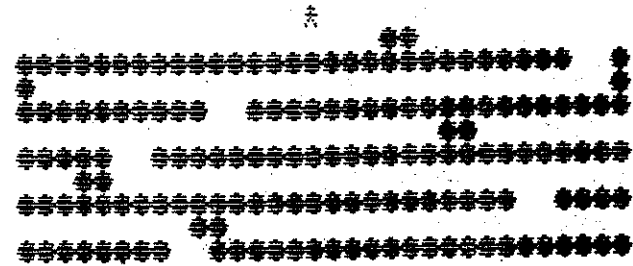
```



```

00010 RESTORE 0400
00020 READ a,b: IF a=99 THEN RETU
RN
00030 IF a=00 THEN PRUSE 5+5: GO
TO 0005
00040 PRINT a/3,b-3+lev: GO TO 0002
00050 DATA .0,4,.0,4,.0,10,.0,4,.
00060 .4,.0,10,.0,4,.0,4,.0,10,1
00070 DATA .00,10
00080 DATA .4,10,.4,10,.4,5,.4,5,
00090 .0,4,0
00100 DATA 00,00,.5,0,.5,4
00110 DATA 00,0
00120 PRINT C
00130 PRINT 0000: FOR a=USR "a"
TO USR "b"+7: READ b: POKE a,b:
NEXT a
00140 DATA 10,55,16,124,16,40,60,
00150 24,128,106,255,250,105,100,24
00160 LET a#=""
00170 DIM s(10): DIM s#(10,20)
00180 FOR a=1 TO 10: LET s#(a)="N
a" PRINT a: NEXT a
00190 FOR a=1 TO 10: READ b: LET
s(a)=b: NEXT a
00200 DATA 350,300,430,470,510,55
00210 600,600,700,1000
00220 LET a#=""
00230 GO TO 4000
00240 FOR N=0000 TO 32500: PRINT
N: " : PEAK Z: NEXT Z
00250 STOP
00260 SAVE "labirinto" LINE 0700:
SAVE "codigo" CODE 00000,500
00270 STOP
00280 CLEAR 31000: LOAD ""CODE :
GO TO 0000

```



4 — Escreva RUN 9000. Quando surgir a página de título, carregue em «S» para jogar. Se o programa estoirar, isso significará que você cometeu um erro em algum lado. Se o programa executar normalmente, tente redefinir as teclas. Verifique tudo.

5 — Quando tiver a certeza de que o programa está bom, faça a sua cópia final, usando RUN 9600. Depois verifique esta cópia usando as instruções VERIFY"" e VERIFY ""CODE.

Execute novamente o jogo, quer carregando-o, quer escrevendo RUN 9000.

O programa, como é óbvio, encontra-se escrito em código-máquina e em Basic. A parte em código-máquina lê o teclado, desloca o labirinto, trata os sons e executa certos outros efeitos especiais. Diversos POKE's existentes no programa principal controlam a execução destas rotinas. A linha 9500 pode ser usada a fim de verificar os valores que constituem o código-máquina, desde que o leitor defina os endereços apropriados. Existe suficiente espaço no mapa da memória usada para o código-máquina para lhe permitir incorporar as suas próprias rotinas de efeitos especiais, se o quiser fazer.

Se dispuser de um computador com uma memória de 48 K e quiser colocar o código-máquina num endereço mais elevado, deve recordar o seguinte:

- 1) É necessário alterar os endereços da função USR citados no programa principal.
- 2) Deve alterar alguns dos bytes do programa em código-máquina, dado que algumas das rotinas chamam outras. Para aqueles que sabem alguma coisa do assunto, será necessário alterar os dois bytes que se seguem às instruções CALL (205, em decimal).
- 3) Faça os ajustamentos necessários no programa carregador do código-máquina.

LABIRINTO A 3 DIMENSÕES

Este interessante programa foi escrito por Scott Vincent. Necessitará de um computador de 48 k para o executar.

Na memória do computador é guardado um labirinto de dez por dez, com uma única entrada/saída. O objectivo consiste em percorrer o labirinto, descobrir o tesouro e voltar depois ao ponto por onde se começou.

Se o leitor perde o sentido da direcção em que se move, pode carregar na tecla «R» para repetir todos os seus passos. O programa repetirá então todos os seus movimentos desde o início, mostrando-os rapidamente. Quando a repetição tiver terminado, você pode continuar o seu caminho.

Deverá usar as seguintes teclas para se deslocar no interior do labirinto:

- 1 — Mover para a esquerda
- 2 — Mover para a direita
- 0 — Avançar
- L — Virar-se

Note que deve carregar na tecla 4 para rodar o corpo; não pode usar para este efeito as teclas 1 ou 2.

Se e quando conseguir sair do labirinto trazendo o tesouro consigo, o computador imprimirá o número de movimentos de que necessitou para o efeito.

Note que o labirinto não é produzido aleatoriamente, ao contrário do que se passa nos programas do mesmo tipo já apresentados anteriormente. O jogador tem assim oportunidade de se habituar progressivamente a ele, dado que no caso contrário seria extremamente difícil resolver o labirinto. No entanto, o tesouro é colocado de cada vez num local diferente.

```

5 BORDER 5: PAPER 5: INK 1: C
L5
10 RANDOMIZE : DIM n(9): DIM n
(9): DIM n$(201,9): DIM n$(200,1
2)
20 FOR x=0 TO 31: READ y: POKE
USR "a"+x,y: NEXT x
30 FOR x=1 TO 9: READ n(x): NE
XT x
40 FOR x=1 TO 9: READ n(x): NE
XT x
50 FOR x=1 TO 200: READ n$(x):
NEXT x
60 FOR x=1 TO 200: READ n$(x):
NEXT x
70 LET q$="001": LET p$="": LE
T l=1: LET tr=0: LET x=1+INT (RN
D*7)
80 LET la=201*(x=1 OR x=4 OR x
=6 OR x=7)+81*(x=2)+127*(x=3)+16
9*(x=5)
90 LET lb=201*(x=4 OR x=6)+100
*(x=1)+84*(x=2)+122*(x=3)+172*(x
=5)+195*(x=7)
100 LET lc=94*(x=1)+98*(x=2)+12

```

```

4*(x=3)+101*(x=4)+117*(x=5)+191*
(x=6)+199*(x=7)
110 CLS : FOR x=1 TO 9: IF n$(l
,x)="" THEN GO SUB n(x)
120 NEXT x
130 FOR x=1 TO 9: IF n$(l,x)=""
THEN GO SUB n(x)
140 NEXT x
150 IF l=2 OR l=1a OR l=1b OR l
=1c THEN GO SUB D00: IF l=2 AND
tr=1 THEN GO TO 70
160 LET a$=INKEY$: IF a$="" THE
N GO TO 160
170 LET p=VAL n$(l, TO 3)*(a$="
1")+VAL n$(l,4 TO 6)*(a$="0")+VA
L n$(l,7 TO 9)*(a$="2")+VAL n$(l
,10 TO )*(a$="L")
180 IF a$="L" THEN GO TO 300
190 IF p=0 THEN GO TO 160
200 BEEP .003,0: LET l=p: LET a
$=a$+"0" AND (L<100)+"0" AND L<
10)+STR$ l
210 LET p$=p$a$: GO TO 110
220 IF l=2 AND tr=0 THEN CLS :
PRINT AT 10,12: FLASH 1;"ENTRADA
": RETURN
230 IF l=2 THEN CLS : PRINT AT
0,4;"Optimo, necessitou de";AT 1
0,11;LEN p$;" movimentos.";AT 14
,6;"Carregue numa tecla.": PAUSE
404: RETURN
240 IF l=1a THEN PLOT 116,99: D
RAW OVER 1;22,0: PLOT 116,76: DR
AW OVER 1;22,0: RETURN
250 IF l=1b THEN PLOT 100,115:
DRAW OVER 1;54,0: PLOT 100,88: D
RAW OVER 1;54,0: RETURN
260 IF l=1c THEN PLOT 72,143: D
RAW OVER 1;110,0: PLOT 72,32: DR
AW OVER 1;110,0: RETURN
270 IF tr=1 THEN PRINT AT 9,11:
"Voce ja";AT 10,12;"tem o";AT 1
1,12;"tesouro.": RETURN
280 LET tr=1: PRINT AT 9,11;"Vo
ce acaba";AT 10,11;"de descobrir
";AT 11,11;"o tesouro."
290 RETURN
300 LET y=1
310 FOR n=1 TO LEN a$ STEP 3
320 IF n=1 THEN GO TO 300
330 IF p$(y)="1" THEN PRINT #1:
AT 0,0;"←"

```

```

340 IF P#(Y)="0" THEN PRINT #1;
AT 0,0;"↑"
350 IF P#(Y)="2" THEN PRINT #1;
AT 0,0;"↓"
360 IF P#(Y)="(" THEN PRINT #1;
AT 0,0;"←"
370 LET Y=Y+1
380 FOR X=1 TO 25 STEP 5: FOR S
=X TO X+4 STEP 1.5: BEEP .004,X:
BEEP .004,X+S: NEXT S: NEXT X
390 LET L=VAL A*(N TO N+2)
400 CLS : FOR X=1 TO 9: IF #(L
,X)="0" THEN GO SUB #(X)
410 NEXT X
420 FOR X=1 TO 9: IF #(L,X)="1
" THEN GO SUB #(X)
430 NEXT X
440 IF L=2 OR L=6 OR L=10 OR L
=14 THEN GO SUB 220
450 NEXT N: GO TO 150
5000 PLOT 49,175: DRAW 32,-32: P
LOT 49,0: DRAW 32,32: RETURN
5005 PLOT 24,175: DRAW 0,-175: 0
DRAW 0,32: DRAW 48,0: DRAW 0,111:
DRAW -48,0: RETURN
5010 PLOT 183,143: DRAW 32,32: P
LOT 183,32: DRAW 32,-32: RETURN
5015 PLOT 231,175: DRAW 0,-175: 0
DRAW 0,32: DRAW -48,0: DRAW 0,11
1: DRAW 48,0: RETURN
5020 PLOT 73,142: DRAW 27,-27: P
LOT 73,33: DRAW 27,27: RETURN
5025 PLOT 73,142: DRAW 11,-11: 0
DRAW 0,-37: DRAW -11,-11: PLOT 65
,65: DRAW 15,0: DRAW 0,55: DRAW
-15,0: RETURN
5030 PLOT 155,115: DRAW 27,27: P
LOT 152,33: DRAW -27,27: RETURN
5035 PLOT 182,33: DRAW -11,11: 0
DRAW 0,37: DRAW 11,11: PLOT 178,6
6: DRAW -15,0: DRAW 0,55: DRAW 1
5,0: RETURN
5040 PLOT 101,114: DRAW 19,-19:
PLOT 101,61: DRAW 19,19: RETURN
5045 PLOT 101,114: DRAW 0,-0: DR
AW 0,-37: DRAW -0,-0: PLOT 110,0
0: DRAW 0,0: DRAW 4,-4: DRAW -4,
4: DRAW 0,-23: DRAW -0,0: DRAW 6
,0: DRAW 4,4: RETURN
5050 PLOT 154,114: DRAW -19,-19:
PLOT 154,61: DRAW -19,19: RETUR
N

```

```

5055 PLOT 154,114: DRAW -0,-0: 0
DRAW 0,-37: DRAW 0,-0: PLOT 145,0
0: DRAW -0,0: DRAW -4,-4: DRAW 4
,4: DRAW 0,-23: DRAW 0,0: DRAW -
0,0: DRAW -4,4: RETURN
5060 PLOT 73,143: DRAW 110,0: DR
AW OVER 1;0,-110: DRAW 0,-1: DR
AW -111,0: DRAW OVER 1;0,110: RET
URN
5065 PLOT 101,115: DRAW 54,0: DR
AW OVER 1;0,-54: DRAW 0,-1: DRAW
-55,0: DRAW OVER 1;0,54: RETURN
5070 OVER 1: PLOT 117,77: DRAW 3
,3: PLOT 130,98: DRAW -3,-3: PLO
T 130,77: DRAW -3,3: PLOT 126,95
: DRAW -4,4: OVER 0: DRAW 23,0:
DRAW OVER 1;0,-22: DRAW 0,-1: DR
AW -23,0: DRAW OVER 1;0,23: RETU
RN
5080 DATA 16,48,96,255,255,96,48
,16,24,96,128,219,24,24,24,24
5090 DATA 0,10,0,255,255,0,10,0,
24,24,24,24,210,128,96,24
5100 DATA 5000,5010,5050,5000,500
0,200,5040,5000,200
5110 DATA 5005,5015,5050,5005,50
05,5005,5045,5005,5070
5120 DATA "111",,"000000010",,"
101",,"000010000",,"010101",,"01000
0011",,"000010101",,"000010111",,"0
00011",,"000000010",,"011",,"100000
000",,"111",,"000100000",,"101",,"10
1",,"001",,"010101",,"101",,"100001"
,"011",,"001",,"011",,"101"
5130 DATA "100000000",,"011",,"010
011",,"000110011",,"011",,"011",,"10
1",,"100011",,"101",,"000111",,"111"
,"011",,"111",,"000111",,"000011",,"
100000111",,"010000101",,"101",,"11
1",,"000101",,"010101",,"101",,"1000
10101",,"001",,"000100010"
5140 DATA "011",,"010111",,"111",,"
011",,"100010000",,"011",,"001",,"10
1",,"110011",,"111",,"011",,"000111"
,"001",,"001",,"110001",,"110000111"
,"001",,"101",,"110001",,"111",,"00
0010100",,"101",,"010100101",,"0101
01",,"111"
5150 DATA "100101",,"0000010101",,"
010000000",,"101",,"100000010",,"10
0000001",,"010100000",,"011",,"0000
01",,"111",,"111",,"000000111",,"111

```

"100111", "000111", "000100111",
"111", "000000100", "001", "001", "1
00001", "010011", "001", "000010011
", "010001"
6050 DATA "010110000", "100000010
", "110000000", "011", "000000101",
"111", "000101", "000111", "101", "1
10101", "000000111", "101", "011", "
000000110", "111", "000101", "001", "
"110000000", "100000101", "111", "0
11", "000001", "100011", "001", "000
100011"
6070 DATA "100011", "010000001", "
011", "101", "101", "001", "111", "01
1", "010101", "011", "101", "101", "1
01", "000000110", "011", "000110000
", "000011", "110000000", "00000011
1", "100011", "000000011", "0000001
10", "110000000", "000110111", "000
110000"
6080 DATA "110111", "110011", "000
011", "011", "011", "011", "000011",
"101", "101", "111", "101", "011", "0
00100000", "011", "101", "100000100
", "101", "011", "000010001", "011",
"101", "010001", "000111", "1101000
11", "111"
6090 DATA "000110100", "0000000110
", "000000100", "000010011", "10000
0000", "010011", "000100000", "0100
00010", "100000101", "000010000", "
101", "000101", "000000010", "111", "
011", "001", "01", "101", "101", "0
10000000", "000001", "000010000", "
011", "001", "000011"
6100 DATA "003000004002", "0000000
000001", "0000005000005", "00000000
0020", "0000007000000", "000000400020
03", "00000100000011", "0000000000005
", "00000020000053", "00000120000013",
"0000000000007", "0000000014015"
6110 DATA "0000110000010", "0100000
017016", "0000100000012", "01500000
0014", "0200000000021", "00000000000
10", "000017010010", "0200000000022
", "0160100000017", "00000000021000",
"0000000000024", "00000000020023"
6120 DATA "0200000000027", "0000000
000004", "0000000000025", "000000000
9031", "0000000000000", "00000000000
33", "0000000027020", "0000000000004
", "0200001000000", "0000000000000",
"00000000000037", "00000000000040"

6130 DATA "0000000034005", "0400000
043041", "0000000000000", "0000000700
0000", "0400000000000", "000004500440
40", "004000000000000", "07100000720070
", "000004700000000", "0000040041040",
"0400000000000", "04400000000045",
6140 DATA "00000000000001", "0000000
0000047", "000000000000000", "000000000
0030", "01100000100000", "0000000000070
00", "0410000000000", "000000000000004
", "000000000000000", "000000000000007",
"0000001000000", "001000000000000",
6150 DATA "00000000000000", "0000000
0000000", "000000000000000", "0000000
00000", "000000000000000", "00000000000
01", "000000000000000", "000000000000007
", "0010000000000", "040000000000004",
"000000700000004", "10100000000100",
6160 DATA "00000000000000", "0000000
0700071", "12100001000000", "070000000
0000", "000000740000000", "00000000000
00", "001000000000000", "0700000000000
", "000000000000000", "000000000000007",
6170 DATA "00000000000000", "0700000
0000070", "00000000000000", "004000000
0000", "000000000000000", "000000000000
01", "000000000000000", "000000000000000
", "000000000000000", "000000000000000",
"00000000000100", "000000000000004",
6180 DATA "000000000000004", "0000000
0000000", "00000000000000", "000000000
0000", "000000000000004", "07000000000
72", "145144147145", "0000000000002101
", "000000000000000", "104000000000000",
"000000000000000", "000000000000000",
6190 DATA "110000000000111", "1141110
115113", "000000000000000", "150000000
0152", "00000000111110", "000015000001
51", "1100000117110", "00001710000172
", "00000000000110", "1131141121115",
"1101100000117", "07700000000000",
6200 DATA "0000000100107", "0000104
0000120", "1201210000120", "000000000
0125", "0000120000124", "130120000001
20", "0000122120121", "000000000000104
", "1270000000120", "13100000000120",
"0000000000100", "120000001000100",
6210 DATA "00000001001031", "0000100
100120", "00000000000000", "130000000
0130", "13400000000100", "137000000001
00", "00001410000140", "00000001000100"

```

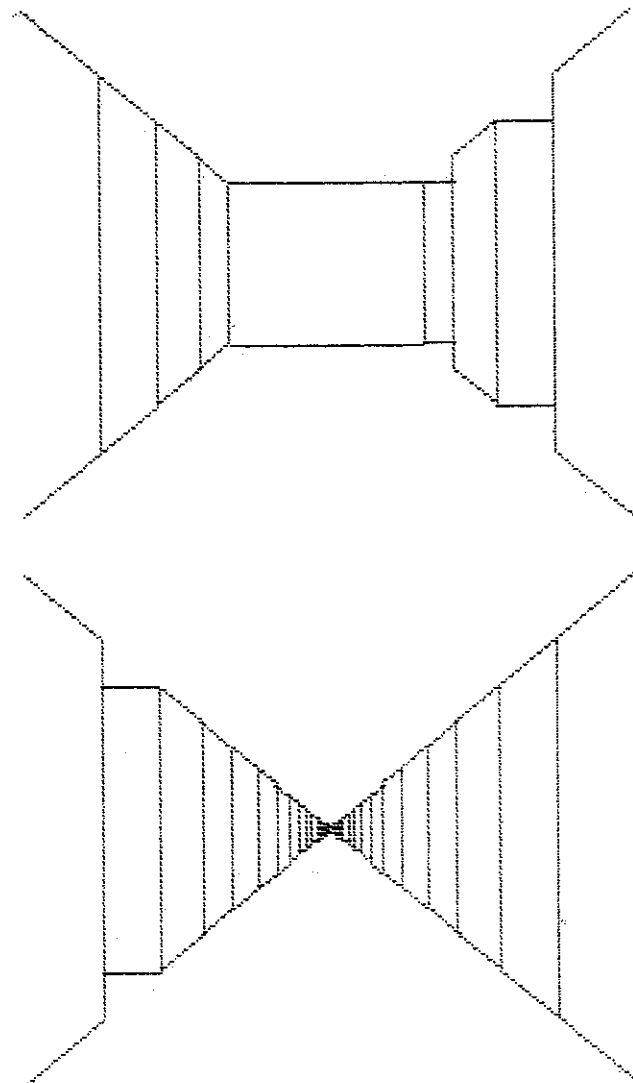
"00001400000140", "00001400000141",
"144147145140", "00001700000174",
0000 DATA "10001040000100", "00001400000140",
"00001400", "00001400000140", "14514514514",
4147", "00001000000100", "00001400000140",
40", "1100110011001100", "1100110011001100",
":", "00001000000100", "00001000000100",
"00000001000", "00000001000", "1000",
0000 DATA "00001000000100", "1000",
"00001000", "10000000000100", "10000000000100",
4100", "10000000000100", "10000000000100",
00", "00001000000100", "00001000000100",
", "10000000000100", "10000000000100",
"10000000000100", "0000000100170",
0040 DATA "00001700000171", "00000000",
107100", "17000000000100", "000011711",
0110", "00001700000170", "14714514514",
44", "1700000170177", "00001740000170",
", "00001700000170", "00001000000100",
"00001010000100", "1771700000170",
0000 DATA "0000104100100", "0000104",
000100", "00001700177170", "10000100700",
0100", "00001000000100", "10000000000100",
00", "00001040000100", "00001000000100",
", "1070000100100", "0000000100100",
"00000000000100", "0000000100100",
0000 DATA "10000000000100", "1000",
000107", "00001000000100", "00001000000",
0100", "00001000000100", "00001000000",
00", "00000000000", "00001000000100"

```

LABIRINTO A 3 DIMENSÕES — II

Quando tiver dominado o programa de Scott Vincent, pode progredir para o que apresentamos em seguida. Neste programa, escrito por Graham Charlton, você vê novamente o labirinto em três dimensões. No entanto, ao contrário do que acontece no anterior, o labirinto é agora diferente de cada vez que o programa é executado.

O labirinto é constituído por uma rede de 22 X 32, necessitando o computador de aproximadamente um minuto para o criar. Vejamos alguns exemplos de execução deste programa:




```

THEN PLOT 1400 1000,00+1: 0000 0000 0000
... TO PLOT 1400 1000,00+1: 0000 0000 0000
0001 TO PLOT 1400 1000,00+1
0002 TO PLOT 1400 1000,00+1
0003 TO PLOT 1400 1000,00+1
0004 TO PLOT 1400 1000,00+1
0005 TO PLOT 1400 1000,00+1
0006 TO PLOT 1400 1000,00+1
0007 TO PLOT 1400 1000,00+1
0008 TO PLOT 1400 1000,00+1
0009 TO PLOT 1400 1000,00+1
0010 TO PLOT 1400 1000,00+1
0011 TO PLOT 1400 1000,00+1
0012 TO PLOT 1400 1000,00+1
0013 TO PLOT 1400 1000,00+1
0014 TO PLOT 1400 1000,00+1
0015 TO PLOT 1400 1000,00+1
0016 TO PLOT 1400 1000,00+1
0017 TO PLOT 1400 1000,00+1
0018 TO PLOT 1400 1000,00+1
0019 TO PLOT 1400 1000,00+1
0020 TO PLOT 1400 1000,00+1
0021 TO PLOT 1400 1000,00+1
0022 TO PLOT 1400 1000,00+1
0023 TO PLOT 1400 1000,00+1
0024 TO PLOT 1400 1000,00+1
0025 TO PLOT 1400 1000,00+1
0026 TO PLOT 1400 1000,00+1
0027 TO PLOT 1400 1000,00+1
0028 TO PLOT 1400 1000,00+1
0029 TO PLOT 1400 1000,00+1
0030 TO PLOT 1400 1000,00+1
0031 TO PLOT 1400 1000,00+1
0032 TO PLOT 1400 1000,00+1
0033 TO PLOT 1400 1000,00+1
0034 TO PLOT 1400 1000,00+1
0035 TO PLOT 1400 1000,00+1
0036 TO PLOT 1400 1000,00+1
0037 TO PLOT 1400 1000,00+1
0038 TO PLOT 1400 1000,00+1
0039 TO PLOT 1400 1000,00+1
0040 TO PLOT 1400 1000,00+1
0041 TO PLOT 1400 1000,00+1
0042 TO PLOT 1400 1000,00+1
0043 TO PLOT 1400 1000,00+1
0044 TO PLOT 1400 1000,00+1
0045 TO PLOT 1400 1000,00+1
0046 TO PLOT 1400 1000,00+1
0047 TO PLOT 1400 1000,00+1
0048 TO PLOT 1400 1000,00+1
0049 TO PLOT 1400 1000,00+1
0050 TO PLOT 1400 1000,00+1
0051 TO PLOT 1400 1000,00+1
0052 TO PLOT 1400 1000,00+1
0053 TO PLOT 1400 1000,00+1
0054 TO PLOT 1400 1000,00+1
0055 TO PLOT 1400 1000,00+1
0056 TO PLOT 1400 1000,00+1
0057 TO PLOT 1400 1000,00+1
0058 TO PLOT 1400 1000,00+1
0059 TO PLOT 1400 1000,00+1
0060 TO PLOT 1400 1000,00+1
0061 TO PLOT 1400 1000,00+1
0062 TO PLOT 1400 1000,00+1
0063 TO PLOT 1400 1000,00+1
0064 TO PLOT 1400 1000,00+1
0065 TO PLOT 1400 1000,00+1
0066 TO PLOT 1400 1000,00+1
0067 TO PLOT 1400 1000,00+1
0068 TO PLOT 1400 1000,00+1
0069 TO PLOT 1400 1000,00+1
0070 TO PLOT 1400 1000,00+1
0071 TO PLOT 1400 1000,00+1
0072 TO PLOT 1400 1000,00+1
0073 TO PLOT 1400 1000,00+1
0074 TO PLOT 1400 1000,00+1
0075 TO PLOT 1400 1000,00+1
0076 TO PLOT 1400 1000,00+1
0077 TO PLOT 1400 1000,00+1
0078 TO PLOT 1400 1000,00+1
0079 TO PLOT 1400 1000,00+1
0080 TO PLOT 1400 1000,00+1
0081 TO PLOT 1400 1000,00+1
0082 TO PLOT 1400 1000,00+1
0083 TO PLOT 1400 1000,00+1
0084 TO PLOT 1400 1000,00+1
0085 TO PLOT 1400 1000,00+1
0086 TO PLOT 1400 1000,00+1
0087 TO PLOT 1400 1000,00+1
0088 TO PLOT 1400 1000,00+1
0089 TO PLOT 1400 1000,00+1
0090 TO PLOT 1400 1000,00+1
0091 TO PLOT 1400 1000,00+1
0092 TO PLOT 1400 1000,00+1
0093 TO PLOT 1400 1000,00+1
0094 TO PLOT 1400 1000,00+1
0095 TO PLOT 1400 1000,00+1
0096 TO PLOT 1400 1000,00+1
0097 TO PLOT 1400 1000,00+1
0098 TO PLOT 1400 1000,00+1
0099 TO PLOT 1400 1000,00+1
0100 TO PLOT 1400 1000,00+1

```

JOGOS PARA DOIS JOGADORES

QUADRADOS

Este programa, escrito por Neil Pallinacci, inclui versões para um ou dois jogadores. Em ambas as versões o jogo é jogado num mapa de dez por dez.

Você deve colocar marcas no mapa, tendo em conta que não pode colocar nenhuma delas junto às do computador.

O jogo para um jogador é chamado «AREA», e nele o leitor joga contra o computador. Perde quem primeiro ficar impossibilitado de colocar uma pedra no tabuleiro. Será rejeitado qualquer movimento que envolva a colocação de uma peça adjacente a uma já colocada pela máquina.

O computador coloca as suas peças aleatoriamente, a não ser que tenha problemas em descobrir um movimento, caso em que procurará metodicamente determinar o seu movimento. Se não conseguir executá-lo, admitirá a derrota.

O jogo para dois jogadores é chamado «LINHAS» e ganha quem primeiro conseguir colocar cinco peças em linha, horizontal ou verticalmente. Os movimentos serão rejeitados se o quadrado sugerido já estiver preenchido. No entanto, se um jogador colocar a sua peça adjacente às do adversário, a peça passa a ser deste. É possível usar várias técnicas para ganhar (tal como acontece na versão anterior). Mas basta um pequeno erro para perder.

É essencial que o leitor coloque o computador no modo «maiúsculas» (CAPS LOCK) imediatamente depois de ter indicado a opção de jogo. Quando o programa começa, imprime uma página de título e pede que seja escolhida uma opção (A a D).

As duas primeiras opções, para um ou dois jogadores, são óbvias. A terceira opção, alterar cores, permite-lhes definir as cores de ambos os jogadores carregando na tecla apropriada, 1 a 7, quando lhe for pedido. As duas cores devem obviamente ser diferentes.

Se pedir informação o computador descreverá rapidamente o objectivo do jogo.

Quando começar a jogar, o computador far-lhe-á a seguinte pergunta:

NOME DO JOGADOR 1

Escreva o seu nome, ou carregue em ENTER no caso de o nome já ter sido indicado anteriormente. Proceda do mesmo modo para o segundo jogador, quando necessário.

O visor apresenta o tabuleiro enquanto o jogo progride, ao mesmo tempo que indica a cor correspondente a cada jogador, a respectiva pontuação e a quantidade de quadrados dessa cor que se encontram actualmente no tabuleiro. Na parte inferior direita da imagem é indicado a quem pertence o movimento seguinte. Por baixo é apresentado o último movimento indicado ao computador, a branco.

Escreva o seu movimento indicando primeiro um número e depois uma letra, por exemplo 7A, 1D, sempre por esta ordem. Se escrever AA terminará o jogo, voltando à página inicial para definição de opções.

Este programa encontra-se escrito inteiramente em Basic, sendo formado por vários módulos diferentes ligados entre si pela rotina principal. Esta ocupa as linhas 540 a 1000, encontrando-se os diversos módulos a seguir.

```
5000 DIM L(15)
5010 DIM #*#*#*#
5020 GO SUB 6000
5030 GO SUB 6000: BORDER# 6: TYPE
5040 GO SUB 6000: INK# 7
5050 GO SUB 6000: PAPER 6: INK 7
5060 GO SUB 6000: GO SUB 6000: R
5070 Jogoador UM
5080 IF #7 OR #8 THEN GO TO 5000
5090 GO SUB 6000
5100 IF #7=1 THEN GO SUB 6000
5110 IF #7=2 THEN GO SUB 4000
5120 IF #7 OR #8 THEN GO TO 5000
5130 GO TO 700
5140 STOP
5150 Jogoador UM
5160 DIM P(20,15): PRINT INK 6: AT
16, 20: "JOGADOR UM", AT 17, 20: #
5170 PRINT AT 18, 20: INK 6: "Move
":
5180 GO SUB 2000: PRINT AT 19, 20
":
5190 #
5200 IF #7 THEN LET #7=1: RETURN
```

```
2000 IF B(X,Y) <> 0 THEN GO TO 200
2010 IF #7=1 THEN GO TO 2100
2020 LET L=(Y-1)*2+1: LET CC=(X
-1)*2+1
2030 LET P#=#7+1: LET B(X,Y)=1
2040 PRINT AT (L,CC): OVER 1: PAPER
": AT (L+1,CC): "
2050 IF B(X,Y+1)=0 OR B(X,Y-1)=0
OR B(X+1,Y)=0 OR B(X-1,Y)=0 THEN
2060 LET B(X,Y)=0: PRINT AT (L,CC):
OVER #7: PAPER #7: AT (L+1,CC): "
": LET P#=#7+1: LET #7=#7+1
2070
2080
2090 IF #7=1 THEN LET #7=0: RETU
RN
2100 LET #7=0: GO SUB 7000
2110 RETURN
2120 IF B(X,Y+1)=0 OR B(X,Y-1)=0
OR B(X+1,Y)=0 OR B(X-1,Y)=0 THE
N
2130 GO TO 2000
2140 GO TO 2000
2150 DIM #OVER#
2160 INPUT LINE ##: IF LEN # <> 0
THEN GO TO 2000
2170 IF #="AA" THEN LET #7=1: R
ETURN
2180 IF CODE #*(1) < 48 OR CODE #*
(1) > 57 THEN GO TO 2000
2190 IF CODE #*(2) < 48 OR CODE #*
(2) > 74 THEN GO TO 2000
2200 LET X=CODE #*(1)-48
2210 LET Y=CODE #*(2)-50
2220 RETURN
2230 DIM spectrum#
2240 PRINT AT 16, 20: INK 6: "JOG.
": AT 17, 20: #7: AT 18, 20: #7
2250 #7:
2260 FOR #=1 TO 15
2270 LET X=INT (RND*10)+2: LET Y
=INT (RND*10)+2
2280 IF B(X,Y) <> 0 THEN GO TO 200
0
2290 IF B(X+1,Y)=1 OR B(X-1,Y)=1
OR B(X,Y+1)=1 OR B(X,Y-1)=1 THE
N
2300 #: GO SUB 3000: IF #7 THE
N RETURN
2310 LET L=(Y-1)*2+1: LET CC=(X
-1)*2+1: PRINT AT (L,CC): OVER 1:
PAPER #7: AT (L+1,CC): "
":
2320 LET B(X,Y)=2: LET P#=#7+1
```



```

0000 RETURN
0001 DIM X#(10)
0002 FOR X# = 2 TO 11: FOR Y# = 2 TO 4
0003 IF b(x,y) <> 0 THEN GO TO 0040
0004 IF b(x+1,y) = 1 OR b(x-1,y) = 1
0005 OR b(x,y+1) = 1 OR b(x,y-1) = 1 THEN
0006 GO TO 0040
0007 RETURN
0008 NEXT Y#: NEXT X#: LET WF = 1: G
RETURN
0009 RETURN
0010 DIM JOGADOR, DOIS
0011 DIM P, Q, -10: PRINT INK 6; AT
10, 20; "JOG. DOIS"; AT 17, 20; A#
0012 PRINT AT 10, 22; INK 6; "MOVE
7:
0013 GO SUB 2000: PRINT AT 10, 20
0014 IF a# THEN LET a# = 2: RETURN
0015 IF b(x,y) <> 0 THEN GO TO 403
0016 LET L = (y-2)*2+1: LET CC = (X
-1)*2+1
0017 LET P = 0: Q = 0: LET b(x,y) = 0
0018 PRINT AT L, CC; OVER 1; PAPER
0019 AT L+1, CC
0020 IF b(x+1,y) = 1 OR b(x-1,y) = 1
0021 OR b(x,y+1) = 1 OR b(x,y-1) = 1 THEN
0022 LET b(x,y) = 1: PRINT AT L, CC
0023 OVER 1; PAPER 01; " "; AT L+1, CC
0024 LET P = P+1: LET Q = Q+1
0025 G-1
0026 DIM P, Q
0027 LET WF = 0: GO SUB 7000
0028 RETURN
0029 DIM JOGADOR terminado
0030 PRINT 1: PAPER 1: FOR a = 13
TO 24: PRINT AT a, 20;
": NEXT a
0031 PRINT AT 13, 23; "TERMINADO"
0032 IF a# THEN PRINT AT 16, 24; "
PARADO"; AT 16, 20; "DELO JOG. "; AT
17, 27; a#
0033 IF a# THEN PRINT AT 16, 23; "
0 JOG. "; AT 16, 20; "GAMADO"
0034 PRINT AT 16, 20; INK 6; "PRIM
0035 PRINT AT 16, 20; "PRIMO"
0036 PRINT 0: PAPER 0: GO SUB 0
0037 FOR K = 10 TO 40 STEP 5: DIM
P, Q, X: DIM P, Q, -K: NEXT K
0038 PAUSE 0: GO TO 0040

```

```

0000 DIM PAPER, DOIS
0010 PRINT AT 2, 20; INK 7; "JOGAD
OR UM"; AT 0, 20; A#; AT 4, 20; INK 7
: PAPER 01; " "; PAPER 0; INK 6; "
"; AT 17, 20; "PRIMO"
0020 PRINT AT 10, 20; INK 7; "JOG.
0016"; AT 10, 20; A#; AT 11, 20; INK
7; PAPER 02; " "; PAPER 0; INK 6
"; AT 17, 20; "PRIMO"
0030 RETURN
0040 RETURN
0050 DIM verificacao victoria
0060 FOR X# = 2 TO 11: FOR Y# = 2 TO 4
0070 IF b(x,y) <> 0 THEN LET a = b(x
,y): GO SUB 7000: IF WF = 1 THEN R
TURN
0080 NEXT Y#: NEXT X#: RETURN
0090 LET X1 = X: LET Y1 = Y
0100 FOR Z = 1 TO 4: LET X1 = X1+1:
IF b(X1,Y1) <> a THEN GO TO 7000
0110 NEXT Z: LET WF = a: RETURN
0120 LET X1 = X: LET Y1 = Y
0130 FOR Z = 1 TO 4: LET Y1 = Y1+1:
IF b(X1,Y1) <> a THEN GO TO 7000
0140 NEXT Z: LET WF = a: RETURN
0150 RETURN
0160 DIM variaveis
0170 DIM a#(10): DIM b#(10): LET
c1 = 4: LET c2 = 5
0180 DIM b(10,10)
0190 POKE 20000, 100
0200 RETURN
0210 DIM info
0220 BORDER 1: PAPER 1: INK 6: C
L0
0230 PRINT AT 1, 10; "AREA"; AT 2, 1
0; "----"
0240 PRINT AT TAB 4; "JOGO PARA UM
GO JOGADOR"; "VOCE GANHA
O JOGO COMPUTADOR. " "COLOCUE O JOG
ADOR DE MODO A QUE " "NAO FIQUE
EM ADJACENTES AOS DO " "COMPUTADO
R.
0250 PRINT " O COMPUTADOR JOGAR
A O JOGO COMO " "MODO E INDICAR O
QUEM GANHA. "
0260 PRINT " INK 7; " " CARRE
QUE NAHA TECLA. "
0270 PAUSE 0: BORDER 0: PAPER 0:
CLS: INK 0

```



```

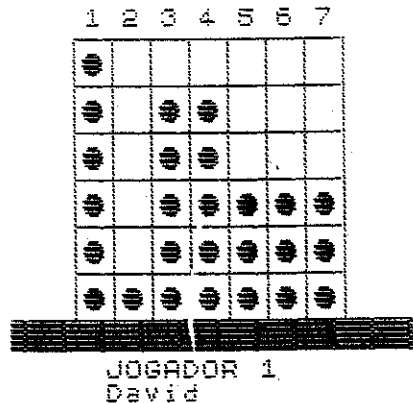
00010 IF @#="" THEN GO TO 0030
00020 LET @#=#
00030 INPUT "Nome do Jogador 1: "
LINE @#
00040 IF @#="" THEN GO TO 0050
00050 LET @#=#
00060 RETURN

```

QUATRO EM LINHA

Da fértil imaginação de David Perry saiu esta versão de Quatro em Linha. O programa contém todas as instruções necessárias para ser executado.

Apresentamos um exemplo de execução seguido da listagem do programa.



```

2 FOR n=0 TO 7: READ a: POKE
USR "a"+n,e: NEXT n: GO SUB 0000
: DIM r(7,7)
3 DATA 00,126,255,255,255,255
,100,00
10 BORDER 0: PAPER 0: INK 7: 0
RIGHT 1: CLS
20 PRINT AT 3,7;"1 2 3 4 5 6 7"

40 FOR n=52 TO 166 STEP 16
50 PLOT 0,40
60 DRAW 0,100
80 NEXT n

```

```

90 FOR n=44 TO 142 STEP 16
100 PLOT 0,0: DRAW 120,0
110 NEXT n
120 FOR n=34 TO 44: PLOT 25,n:
DRAW 100,0: NEXT n
130 LET p1@=1
140 BRIGHT 0
200 LET p1@=1: PRINT AT 10,0:
INK 7: PAPER 0: FLASH 1;"JOGADOR
1": PRINT AT 10,0;@#;"
210 INPUT "Mover Jogador 1? ";#
0
220 IF #>7 OR #<1 THEN GO TO
210
230 GO SUB 1000
240 LET p1@=2: PRINT AT 10,0:
INK 7: PAPER 0: FLASH 1;"JOGADOR
2": PRINT AT 10,0;@#;"
250 INPUT "Mover Jogador 2? ";#
0
270 IF #>7 OR #<1 THEN GO TO
260
280 GO SUB 1000
290 GO TO 0000
4000 FOR n=0 TO 1 STEP -1: IF r(
n,#0)="" THEN NEXT n: RETURN
4070 LET r(n,#0)=1
4080 IF p1@=1 THEN PRINT AT 3+(
n*2), (#0*2)+5: INK 0: BRIGHT 1;"
●": REM grafico B
4081 IF p1@=2 THEN PRINT AT 3+(
n*2), (#0*2)+5: INK 0: BRIGHT 1;"
●": REM grafico B
4090 RETURN
5000 BORDER 1: CLS: PLOT 0,0: D
RAW 0,175: DRAW 255,0: DRAW 0,-1
75: DRAW -200,0
5010 PRINT AT 2,0: BRIGHT 1;"OUR
TRO EM LINHA !!!"
5020 PRINT AT 3,0;"
5030 PRINT AT 4,0:" Neste jogo d
eve usar o teclado para mover as peças.
1 para VERTICAL, 2 para HORIZONTAL ou
DIAGONAL... Por exemplo:
5050 PLOT 0,0: DRAW 0,175
5060 PLOT 255,0: DRAW 0,175
5070 PRINT AT 10,4:"●●●●": PRINT
AT 10,11:"●":AT 11,11:"●":AT 12
,11:"●":AT 13,11:"●"

```

```

0010 PRINT AT 10,1; INK 7; PAPER
0011 FLASH 1; LOGADOR 1; " ": PAUSE
0012# INPUT "Nome do Jogador 1?"
0013# IF LEN #>10 THEN GO TO 0000
0014 PRINT AT 10,1; INK 7; PAPER
0015 FLASH 1; LOGADOR 1; " ": PAUSE
0016# PRINT AT 10,20; " = " ; INK 0
0017# PRINT AT 10,20; " = " ; INK 0
0018 PRINT AT 20,7; FLASH 1; INK
0019 "Carregue # ENTER. "; PAUSE 0
: RETURN

```

TANQUES

Trata-se de um jogo para duas pessoas, no qual é necessário disparar sobre a torre do tanque adversário.

Vejamos a lista dos comandos a usar pelo primeiro jogador:

Esquerda — Z
Direita — X
Subir — 2.^a linha/esquerda
Descer — 3.^a linha/esquerda
Disparar — Linha superior/esquerda

E agora os comandos do segundo jogador:

Esquerda — M
Direita — Symbol Shift
Subir — 2.^a linha/direita
Descer — 3.^a linha/direita
Disparar — Linha superior/direita

Convirá notar que a versão aqui apresentada utiliza para a leitura dos «ports» correspondentes ao teclado os valores ade-

quados à versão 3 do Spectrum. Estes valores são diferentes no caso das versões 1 e 2 deste computador. Se o leitor possuir uma destas últimas versões do computador, deve alterar, nas linhas onde existem REM's seguidas de número (por exemplo, REM 255) o valor que se encontra à frente da instrução IN, substituindo-o pelo número indicado na REM.

Note que, ao jogar, os tanques desaparecem durante um momento quando se dispara, mas que mesmo assim podem ser atingidos ou fugir da bala. Não se pode passar sobre as minas «largadas» pelas balas. Cada jogador dispõe de três vidas, e o primeiro a morrer, como é óbvio, perde. Este programa foi escrito por Neal Cavalier-Smith.

```

10 REM Tanques
20 REM Neal Cavalier-Smith
30 LET X1=INT (RND*20): LET Y1
=0: LET X2=INT (RND*20): LET Y2
=0
30 LET Y1B=1: LET X1B=0: LET Y
25=-1: LET X2B=0
40 LET CC=0: LET C1=0: LET C1F
=0: LET C1F=1-3
50 GO SUB 1000
60 GO SUB 1000
70 GO SUB 000
100 REM movimento
101 PRINT AT X1,Y1; " "
102 IF X1B=0 THEN PRINT AT X1,Y
1+Y1B; " "
103 IF Y1B=0 THEN PRINT AT X1+X
15,Y1; " "
110 IF IN 65022<101 AND X1<20 T
HEN LET X1=X1+1: LET X1B=1: LET
Y1B=0: REM 050
120 IF IN 64010<101 AND X1>0 TH
EN LET X1=X1-1: LET X1B=-1: LET
Y1B=0: REM 050
130 IF IN 65070=100 AND X1>0 TH
EN LET Y1=Y1-1: LET Y1B=-1: LET
X1B=0: REM 050
140 IF IN 65078=107 AND Y1<0 T
HEN LET C1=C1+1: LET Y1B=1: LET
X1B=0: REM 051
150 PRINT AT X2,Y2; " "
160 IF X2B=0 THEN PRINT AT X2,Y
0+Y2B; " "
170 IF Y2B=0 THEN PRINT AT X2+X
05,C0; " "
180 IF IN 40150<101 AND X2<20 T
HEN LET X2=X2+1: LET X2B=1: LET
Y2B=0: REM 050

```



```

000000 FLASH 1:"T F R M I N D O O
000001 PRINT TAB 7;"68098 0 000000"
000002 STOP
000003 GO TO 100
000004 PRINT AT X1,01; FLASH 1;"000000"
000005 FOR X=5 TO 100: NEXT X
000006 PRINT AT X1,01; FLASH 0,7
000007
000008 INT 10001=10001-1
000009 IF 10001=5 THEN PRINT INVER
000010 FLASH 1:"T F R M I N D O O
000011 PRINT TAB 7;"68098 0 000000"
000012 STOP
000013 GO TO 100

```

AUXILIARES EM CÓDIGO-MÁQUINA

MONITOR DE MEMÓRIA

O programa que se segue, escrito por David Perry, permite ao leitor examinar o conteúdo da ROM e da RAM sob a forma de endereço decimal e hexadecimal, valor nesse endereço em decimal e hexadecimal, e carácter correspondente a este valor. O programa admitirá também deslocar livremente códigos em memória, converter hexadecimal para decimal e vice-versa, além de executar qualquer rotina em código.

```

ESTE PROGRAMA REVELARÁ O CON-
TEÚDO DA ROM E RAM SOB A FORMA
DE ENDEREÇO DECIMAL E HEXADeci-
MAL, VALOR NESSE ENDEREÇO EM DE-
CIMAL E HEXADecimal, E CARACTÉR
CORRESPONDENTE A ESTE VALOR.
O PROGRAMA ADMITIRÁ TAMBÉM
DESLOCAR LIVREMENTE CÓDIGOS EM
MEMÓRIA, CONVERTER HEXADecimal
PARA DECIMAL E VICE-VERSA, ALÉM
DE EXECUTAR QUALQUER ROTINA EM
CÓDIGO.

```

Ende#	+hex:	dec:	hex:	chr#:
0	0000	000	0000	NEXT
1	0001	175	0001	000
2	0002	17	0002	?
3	0003	000	0003	COPY
4	0004	000	0004	COPY
5	0005	000	0005	NOT
6	0006	000	0006	THEN
7	0007	17	0007	?
8	0008	42	0008	*
9	0009	000	0009	4
10	000A	000	000A	/
11	000B	004	000B	:/
12	000C	000	000C	/
13	000D	000	000D	/


```

520 IF i#="0" OR i#="c" THEN CL
530 GO TO 300
540 IF i#="B" OR i#="H" THEN GO
TO 500
540 IF i#<"6" OR i#<"s" OR i#
<"L" OR i#<"(" OR i#<"H" OR i#
<"I" OR i#<"d" OR i#<"D" OR i#
<"e" OR i#<"m" OR i#<"c" OR
i#<"0" OR i#<"# " OR i#<"M" T
HEN GO TO 400
550 GO TO 410
560 REM executar rotina
570 INPUT "Endereco de execucao
? ";ex
580 RANDOMIZE USA ex: PAUSE 0:
XUN
590 REM deslocar memoria
600 INPUT "Deslocar de endereco
? ";#f
610 IF #f>65535 OR #f<0 THEN GO
TO 600
620 INPUT "Deslocar para endere
co? ";#t
630 IF #t>65535 OR #t<0 THEN GO
TO 600
640 INPUT "No. de bytes a deslo
car? ";n
650 IF #t+#f>65535 THEN PRINT #1
: "Deslocados bytes. Tente de nov
o "; PAUSE 0: GO TO 600
660 LET add=#t
670 FOR p=1 TO n
680 POKE #t,PEEK #f
690 LET #t=#t+1: LET #f=#f+1: N
EXT p
700 CLS
710 GO TO 300
715 REM instrucoes
720 DEF FN v(a#)=(CODE a#-48-7#
(a#>"0")-30*(a#>"Z"))
730 BORDER 7: PAPER 7: INK 0: C
LS: PRINT AT 1,7: "
740 PRINT AT 2,7: " M O N I T O
R.
750 PRINT AT 3,7: " © DAVID PE
RAY
760 PRINT AT 4,7: " 1983
770 PRINT AT 5,7: "
780 PRINT AT 8,0: " ESTE PROGRAM

```

```

8 REVELARA' O CON- TEUDO DA ROM
E AEM SOB A FORMA"
790 PRINT "DE ENDERECO DECIMAL
E HEXADECIMI- MAL, VALOR NESSE END
ERECO EM DE-
800 PRINT "CINAL E HEXADECIMAL,
E CARACTER"
810 PRINT "CORRESPONDENTE A EST
E VALOR."
820 PRINT " O PROGRAMA PERMITIR
A' TAMBEM DESLOCAR LIVREMENTE
CODIGOS EM MEMORIA, CONVERTER H
EXADECIMAL"
830 PRINT "PARA DECIMAL E VICE-
VERSA, ALEM DE EXECUTAR QUALQUER
ROTINA EM CODIGO."
840 REM Carregar programa
850 INPUT "Quer carregar um pro
grama (s/n)? ";t#
860 IF t#="0" OR t#="s" THEN GO
TO 800
870 CLS: RETURN
880 INPUT "Endereco de carga? "
:t
890 IF t>65535 OR t<0 THEN GO T
O 800
900 PRINT FLASH 1; PAPER 2; INK
7; AT 21,0: " Ponha o gravador em
movimento."
910 LOAD "CODE t: CLS: RETURN
920 REM gravar
930 INPUT "Endereco do codigo a
gravar? ";#f
940 IF #f>65535 OR #f<0 THEN GO
TO 900
950 INPUT "Numero de bytes? ";n
960 IF n>65535 OR n<1 THEN GO T
O 900
970 INPUT "Nome? ";s#
980 IF s#="" OR LEN s#>10 THEN
GO TO 970
990 SAVE s#CODE #f,n
1000 RETURN

```


Apresentamos agora uma versão do programa anterior escrita inteiramente em Basic. Esta versão foi igualmente concebida para o computador de 48 K, e escrita pelo mesmo autor — David Perry.

GUARDAR «SCREENS» EM CÓDIGO-MÁQUINA

Esta rotina, escrita por David Perry, foi concebida para o computador de 48 K. Permite ao utilizador guardar em memória e imprimir instantaneamente no visor qualquer imagem. Pode ser usada pelo leitor em qualquer programa de sua autoria.

```

10 BORDER 0: PAPER 0: INK 7: C
L5
45 RESTORE
46 FOR n=40000 TO 40028
50 READ a: POKE n,a
60 NEXT n
40 DATA 33,0,54,17,48,117,1,6,
27,237,176,201,33,48,117,17,8,54,
1,0,27,237,176,201,105,144,234,
0,0
50 PRINT AT 0,0:"Carregue imag
em do gravador e guarde-a im
ediatamente!": PRINT "Podere
a depois recupera-la quan-do quise
r!"
50 PRINT : PRINT "Para guardar
a imagem escreva:"
70 PRINT "      RANDOMIZE USR 40
800"
80 PRINT : PRINT "Para imprimi
r a imagem escreva:"
90 PRINT "      RANDOMIZE USR 40
910"
100 PRINT : PRINT : PRINT "Veja
um exemplo."
120 PRINT FLASH 1:" Ponha o gra
vador em movimento."
130 LOAD "SCREEN#": RANDOMIZE
USR 40000
140 PAUSE 0: CLS : PRINT AT 0,0
:" Carregue numa tecla para imor
tir a imagem do visor.": PAUSE
0: RANDOMIZE USR 40012: GO TO 1
40

```

```

1 CLS : PRINT "Este programa
guarda a imagem em memoria e em s
equida recupera-a.Como exemplo,
vou listar este programa, guar
da-lo e imprimi-lo de novo no vis
or."
2 PRINT : PRINT "O programa e
sta completamente escrito em B
asic e pode ser usa-do para muit
os fins."
3 PRINT : PRINT TAB 5:"Carreg
ue numa tecla.": PAUSE 0: CLS
10 REM
20 REM
30 REM
40 REM
45 CLS : LIST 50
50 LET a=30000: REM endereco o
nde se guarda a imagem
60 FOR x=0 TO 21
70 FOR y=0 TO 31
80 LET b=SCREEN# (x,y)
90 PRINT AT x,y:" "
95 POKE a,CODE a#
100 LET a=a+1
110 NEXT y
120 NEXT x
130 REM
140 REM ***imprimir imagem***
150 REM
160 LET a=30000: REM endereco o
nde esta a imagem
170 FOR x=0 TO 21
180 FOR y=0 TO 31
190 LET b=PEEK a
200 PRINT AT x,y;CHR# b
210 LET a=a+1
220 NEXT y
230 NEXT x

```




```

40 PAUSE @: CLS : PRINT AT 1,10;"COPIAR"
41 CLS : PRINT AT 1,10;"====="
42 PRINT AT 1,10;"===== "
43 PRINT AT 1,10;"===== "
44 PRINT AT 1,10;"===== "
45 PRINT AT 1,10;"===== "
46 PRINT AT 1,10;"===== "
47 PRINT AT 1,10;"===== "
48 PRINT AT 1,10;"===== "
49 PRINT AT 1,10;"===== "
50 PRINT AT 1,10;"===== "
51 PRINT AT 1,10;"===== "
52 PRINT AT 1,10;"===== "
53 PRINT AT 1,10;"===== "
54 PRINT AT 1,10;"===== "
55 PRINT AT 1,10;"===== "
56 PRINT AT 1,10;"===== "
57 PRINT AT 1,10;"===== "
58 PRINT AT 1,10;"===== "
59 PRINT AT 1,10;"===== "
60 PRINT AT 1,10;"===== "
61 PRINT AT 1,10;"===== "
62 PRINT AT 1,10;"===== "
63 PRINT AT 1,10;"===== "
64 PRINT AT 1,10;"===== "
65 PRINT AT 1,10;"===== "
66 PRINT AT 1,10;"===== "
67 PRINT AT 1,10;"===== "
68 PRINT AT 1,10;"===== "
69 PRINT AT 1,10;"===== "
70 PRINT AT 1,10;"===== "
71 PRINT AT 1,10;"===== "
72 PRINT AT 1,10;"===== "
73 PRINT AT 1,10;"===== "
74 PRINT AT 1,10;"===== "
75 PRINT AT 1,10;"===== "
76 PRINT AT 1,10;"===== "
77 PRINT AT 1,10;"===== "
78 PRINT AT 1,10;"===== "
79 PRINT AT 1,10;"===== "
80 PRINT AT 1,10;"===== "
81 PRINT AT 1,10;"===== "
82 PRINT AT 1,10;"===== "
83 PRINT AT 1,10;"===== "
84 PRINT AT 1,10;"===== "
85 PRINT AT 1,10;"===== "
86 PRINT AT 1,10;"===== "
87 PRINT AT 1,10;"===== "
88 PRINT AT 1,10;"===== "
89 PRINT AT 1,10;"===== "
90 PRINT AT 1,10;"===== "
91 PRINT AT 1,10;"===== "
92 PRINT AT 1,10;"===== "
93 PRINT AT 1,10;"===== "
94 PRINT AT 1,10;"===== "
95 PRINT AT 1,10;"===== "
96 PRINT AT 1,10;"===== "
97 PRINT AT 1,10;"===== "
98 PRINT AT 1,10;"===== "
99 PRINT AT 1,10;"===== "

```

```

72 PRINT : PRINT "Carregue num
a tecla para execu- tar novamente
e...": PAUSE @: RUN
73 CLS : PRINT "Acaba de carregar
o cabeçalho deum floppy alien
americano."
74 PRINT "Titulo:"
75 PRINT N#
76 PRINT : PRINT "Carregue num
a tecla para execu- tar novamente
e...": PAUSE @: RUN

```

COPIADOR
=====

Este programa copia programas em código-máquina automaticamente.

Se existir um carregador Basic, deve copia-lo primeiro escrevendo do MODOPE "" e depois salvar "nome" LINE "linha que carrega o código-máquina"

Este programa lê o cabeçalho, carrega o programa e em seguida indica o modo de o gravar.

- 1... Desbobine a cassete para o início de sessão em código
- 2... Verifique ligacoes
- 3... Carregue em qualquer tecla

UTILITÁRIOS E DEMONSTRAÇÕES

BIORRÍTMOS

Poderá agora representar graficamente os seus momentos altos e baixos utilizando este programa.

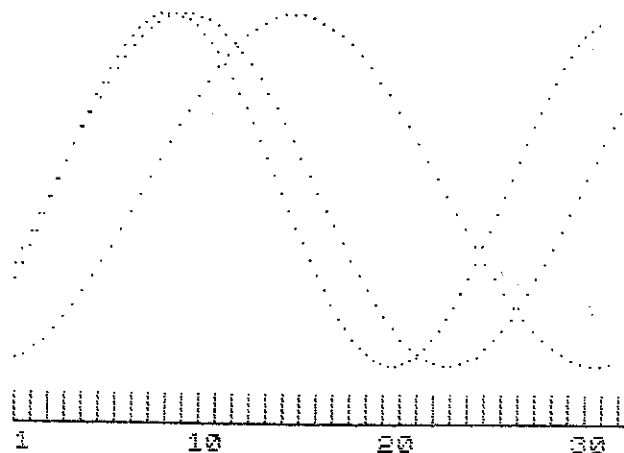
Existem três ciclos, que se iniciam quando você nasce e se mantêm durante toda a sua vida, que muitos pensam ter um efeito sobre esta. Os três ciclos são os seguintes:

Físico — Um ciclo de 23 dias que controla a sua força e resistência físicas.

Emocional — Um ciclo de 28 dias, que governa as suas sensações de optimismo e pessimismo.

Mental — Ciclo de 33 dias que comanda a sua capacidade de raciocínio lógico.

Físico Mental Emocional



O programa far-lhe-á as perguntas necessárias, produzindo em seguida uma imagem bastante esclarecedora.

```

1 RESTORE
2 PRINT AT 0,0;"          BI
ORRITMO."
10 INPUT "Indique data de nasc
imento:";"Dia ";a;" Mes (no.) "
;b;" Ano ";c
20 INPUT "Indique data actual:
";"Mes ";d;" Ano ";e
25 CLS
30 LET t=INT (((e-c)*365.25)+(
(1-b)*365.25)-a)
35 FOR r=0 TO 255
40 PLOT r,10
45 IF r=INT (r/b)*b THEN FOR u
=10 TO 30: PLOT r,u: NEXT u
50 NEXT r
60 PRINT AT 21,0;"1          10
20          30"
65 PRINT AT 0,0; INK 1;"Físico
"; INK 2;"Mental "; INK 4;"
Emocional"
70 FOR r=1 TO 3
80 READ u
90 LET l=2*PI*(t-(INT (t/u)*u)
)/u
100 LET k=2*PI*(33-u)*.50
105 FOR a=1 TO k+(1+(2*PI) STEP
.1
110 PLOT INK ((1 AND u=23)+(2 A
ND u=28)+(4 AND u=33));(a-1)*(100
-20+u),90+SIN a*60
120 NEXT a
130 NEXT r
140 DATA 20,28,30
150 INPUT "Novo biorritmo? ";a#
: IF a#(1)="e" THEN GO TO 1

```

PINTURA

Este programa de Graham Charlton preenche uma forma qualquer com uma determinada cor. As linhas 10 a 60 desenharam a forma, e a subrotina que se inicia na linha 9000 preenche-a. Torne as cadeias A\$ e B\$ tão compridas quanto possível no seu computador; note que ocupam bastante memória.

Se quiser preencher outra parte do visor, altere as linhas 9030 e 9040, onde x e y se referem às coordenadas em pixels do ponto por onde se inicia o enchimento das formas.

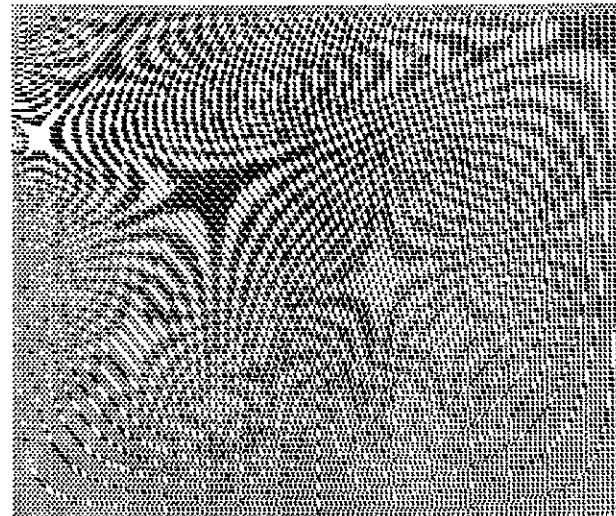
```

10 CIRCLE 125,80,15
20 PLOT 125,80
30 DRAW 45,80
40 DRAW 5,45
50 DRAW 5,45
60 DRAW 5,145
70 GOTO 9000
80 GOTO 9000
9000 GOTO 9000
9001 GOTO 9000
9002 GOTO 9000
9003 GOTO 9000
9004 GOTO 9000
9005 GOTO 9000
9006 GOTO 9000
9007 GOTO 9000
9008 GOTO 9000
9009 GOTO 9000
9010 GOTO 9000
9011 GOTO 9000
9012 GOTO 9000
9013 GOTO 9000
9014 GOTO 9000
9015 GOTO 9000
9016 GOTO 9000
9017 GOTO 9000
9018 GOTO 9000
9019 IF POINT (x+c,y+d)=0 THEN G
9020 GOTO 9000
9021 NEXT d
9022 NEXT c
9023 LET x=x+c
9024 LET y=y+d
9025 GO TO 9000

```

TIMOTHY LEARY

O programa que se segue, de Damian Steele, cria algumas imagens de bastante impacto, que demonstram bem (se ainda necessitar de ser convencido) a eficácia do tratamento gráfico do seu computador.



```

10 PAPER 6: BORDER 1: INK 1: L
ET d=1: LET e=1: LET f=0: LET x=
0: LET y=175: LET a=a: LET b=0:
LET c=0: OVER 1
20 GOTO 9000: PRINT #0; AT 0,1: "Car
egue numa tecla se estiver abor
rido..."
100 FOR a=x TO y STEP a
110 IF a>=125 THEN GO TO 400
120 IF a>=87 THEN GO SUB 200
130 PLOT 0,a: DRAW (45+457)-b,
a-c
135 IF INKEY$="" THEN GO TO 10
80
140 NEXT a
200 LET b=b+6.1: LET c=c+2
300 RETURN

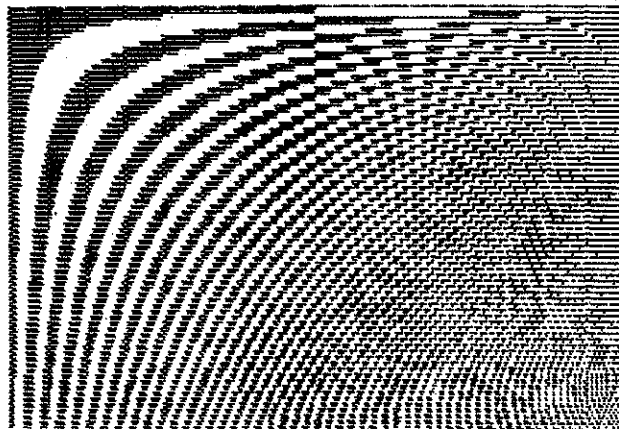
```

TESTE DE CORES

```

400 LET s=2
450 GO SUB 700
500 OVER 1
510 FOR a=1 TO 255 STEP 3
515 IF INKEY#<>" " THEN GO TO 10
50
520 PLOT a,0: DRAW 0,175
530 NEXT a
535 IF b=1 THEN GO TO 500
537 GO SUB 700
540 FOR a=1 TO 175 STEP 3
545 IF INKEY#<>" " THEN GO TO 10
50
550 PLOT 0,a: DRAW 255,0
555 NEXT a
560 GO SUB 700
570 IF b<>1 THEN LET b=1: GO TO
450
580 IF e=1 THEN LET e=0: GO TO
537
590 LET f=f+1: IF f=1 THEN LET
e=1,4: LET b=0: GO TO 410
595 GO SUB 700: GO TO 1000
700 FOR a=1 TO 10: BEEP .007, (R
ND*(30)+35: NEXT a
800 RETURN
1000 PAUSE 0: CLS
1001 LOAD ""
5500 STOP
5550 PRINT FLASH 1: AT 10,0: "PARE
O GRAVADOR": PRINT AT 21,0: "CAR
REGUE NUMA TECLA.": PAUSE 0: GO
TO 1

```



Este programa, bastante simples, de David Perry, imprime todas as cores produzidas pelo seu Spectrum, permitindo-lhe por exemplo sintonizar o receptor de televisão do melhor modo possível.

```

10 BORDER 2: PAPER 0: BRIGHT 1
INVERSE 0: OVER 0: FLASH 0: CL
5
20 LET i=-1: FOR n=0 TO 31 STE
4
30 LET i=i+1
40 FOR y=0 TO 3
50 FOR x=0 TO 21
60 PRINT AT x,y+9: INK i: "███"
70 NEXT x: NEXT y: NEXT n
80 INK 7

```

DEMONSTRAÇÃO DE PORTAS LÓGICAS

Este programa, escrito por Neville Predebon, mostra-nos o resultado do uso de portas lógicas AND, NAND, OR e NOR num circuito. Para usar o programa, responda às perguntas que faz.

```




1 REM Portas
5 REM © Predebon 1983
7 BORDER 5
10 PRINT INK 1: AT 2,13: "PORTAS
"; INK 0: AT 10,2: "Ilustracao de
portas logicas"
10 INK 0
40 PRINT AT 10,6: "Cerreque, num
a tecla.": PAUSE 0: CLS
20 PRINT AT 5,1: "Indique a por
ta": PRINT
35 PRINT TAB 5: "AND = 1"
35 PRINT TAB 5: "NAND = 2"
35 PRINT TAB 5: "OR = 3"
40 PRINT TAB 5: "NOR = 4"
40 IF INKEY#<>" " THEN GO TO 42
40 IF INKEY#="" THEN GO TO 40
45 LET Z#=INKEY#: IF Z#<"1" OR
Z#>"4" THEN GO TO 42
45 CLS

```


Alguns programas que recorrem este efeito utilizam um carácter gráfico com um padrão em xadrez para a mistura das cores. Também o apresentamos aqui, mas em relação às séries de linhas do resultado, reduzindo o evidente movimento dos pontos impressos.

Carregue em qualquer tecla.

1234

1. Linhas horizontais 
2. Xadrez 
3. Linhas quebradas 

Carregue numa das opções 1, 2, 3

A combinação de cores de cada tom pode ser determinada através dos eixos da tabela. Convém notar que as cores de tinta do papel, das cores de impressão do papel, e das cores de impressão do papel são diferentes de cores inversas.

```

100  FOR C=0 TO 7 STEP 1: POKER C
110  FOR A=0 TO 7: FOR B=0 TO 7: POKER C: A+1: NEXT B
120  FOR A=0 TO 7: FOR B=0 TO 7: POKER C: A+1: NEXT B
130  FOR A=0 TO 7: FOR B=0 TO 7: POKER C: A+1: NEXT B
140  FOR A=0 TO 7: FOR B=0 TO 7: POKER C: A+1: NEXT B
150  FOR A=0 TO 7: INK 1: BORDER 7: C
160  FOR A=0 TO 7: INK 7: BORDER 0: C

```

```

10  PRINT AT 10,0: "O 1 2"
20  FOR C=0 TO 7: POKER C
30  FOR A=0 TO 7: POKER C: A+1: NEXT B
40  FOR A=0 TO 7: POKER C: A+1: NEXT B
50  FOR A=0 TO 7: POKER C: A+1: NEXT B
60  FOR A=0 TO 7: POKER C: A+1: NEXT B
70  FOR A=0 TO 7: POKER C: A+1: NEXT B
80  FOR A=0 TO 7: POKER C: A+1: NEXT B
90  FOR A=0 TO 7: POKER C: A+1: NEXT B
100 FOR A=0 TO 7: POKER C: A+1: NEXT B
110 FOR A=0 TO 7: POKER C: A+1: NEXT B
120 FOR A=0 TO 7: POKER C: A+1: NEXT B
130 FOR A=0 TO 7: POKER C: A+1: NEXT B
140 FOR A=0 TO 7: POKER C: A+1: NEXT B
150 FOR A=0 TO 7: POKER C: A+1: NEXT B
160 FOR A=0 TO 7: POKER C: A+1: NEXT B
170 FOR A=0 TO 7: POKER C: A+1: NEXT B
180 FOR A=0 TO 7: POKER C: A+1: NEXT B
190 FOR A=0 TO 7: POKER C: A+1: NEXT B
200 FOR A=0 TO 7: POKER C: A+1: NEXT B
210 PRINT AT 10,1: "Este programa"
220 PRINT AT 10,2: "demonstra como a"
230 PRINT AT 10,3: "possível obter"
240 PRINT AT 10,4: "tintas diferentes a"
250 PRINT AT 10,5: "partir das oito"
260 PRINT AT 10,6: "cores do MX 50"
270 PRINT AT 10,7: "color."
280 PRINT "Alguns programas que"
290 PRINT "recorrem este efeito utilizam"
300 PRINT "um carácter gráfico com um"
310 PRINT "padrão em xadrez para a mistura"
320 PRINT "das cores. Também o apresentamos"
330 PRINT "aqui, mas em relação às séries"
340 PRINT "de linhas do resultado, reduzindo"
350 PRINT "o evidente movimento dos"
360 PRINT "pontos impressos."
370 PRINT AT 11,0: "Carregue em"
380 PRINT AT 11,1: "qualquer tecla."
390 FOR A=0 TO 7: INK 1: BORDER 7: C
400 FOR A=0 TO 7: INK 7: BORDER 0: C

```

```

15 PRINT AT 1,13: INVERSE 1: "M
MENU"
160 PRINT AT 5,8: "1. Linhas por
montais: "; AT 7,8: "2. Xadrez
"; AT 9,8: "3. Linhas quebradas
"; AT 11,8: "Carregue numa das co
rres 1, 2, 3": DIM graficos 3,
3, 0
170 PRINT AT 13,1: "A combinaçao
de cores de cada tom pode ser
determinada atravessando eixos de
tabelas. Convém notar que as co
res de tinta de papel não p
odem ser trocadas, isto é, tint
s azul sobre papel verde não é d
iferente de combi - peças inverti
as"
180 PRINT #0: AT 0,1: INK 1: "As
rotinas que geram os LOG's enco
ntre-se nas linhas 2 a 4."
190 IF INKEY#<>"" THEN GO TO 19
0
200 LET a#=INKEY#
205 IF a#>"3" OR a#<"1" THEN GO
TO 200
210 LET c1=VAL a#: GO TO 19

```

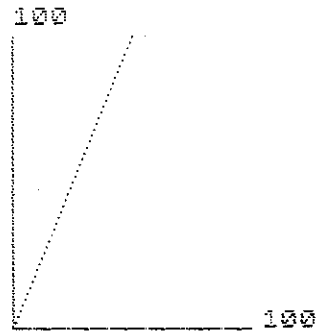
REGRESSÃO LINEAR

Este útil programa, escrito por Neal Cavalier-Smith, permite-lhe indicar um certo número de pontos (à sua escolha) e obter a recta que melhor os representa, dando a equação correspondente.

Apresentamos alguns exemplos de execução, seguidos da listagem do programa:

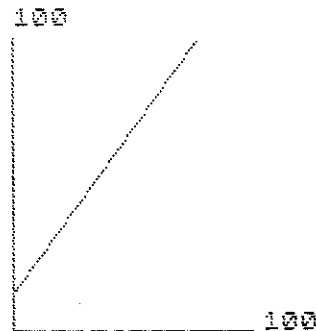
A equação da melhor recta é:

$$Y=2X+0$$



A equação da melhor recta é:

$$Y=1.12X+12.7$$



```

10 LET a=0: LET b=0: LET n=0
20 LET k=0: INK 7: PAPER 0
30 CLS: BORDER 0
40 INPUT "Quantos pontos? " n
50 DIM x(n)
60 DIM y(n)
70 FOR c=1 TO n
75 INPUT "X("; c); "): "; x(c), "Y
("); c); "): "; y(c)
76 PLOT x(c), y(c)
80 LET a=a+x(c)
90 LET b=b+y(c)
100 NEXT c
110 LET a=a/n: LET b=b/n
120 FOR c=1 TO n

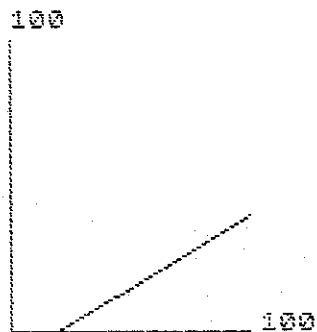
```

```

130 LET M=#+(((X(C)-A)*(Y(C)-B)
)/(ABS(X(C)-A)+2))
140 NEXT C
145 LET M=M/N
150 LET K=B-M*A
160 LET C=0
190 PLOT 0,0: DRAW 100,0: PRINT
AT 21,13:"100"
195 PLOT 0,0: DRAW 0,100: PRINT
AT 3,0:"100"
200 PRINT AT 0,0:"A equacao da
melhor recta e':"
210 PRINT AT 2,0:"Y=";INT (M*10
0+.5)/100:"X+(";INT (K*10+.5)/10
:")."
```

A equacao da melhor recta e':

$$Y = 0.5X + (-0.9)$$



QUADRO DE PONTUAÇÕES

Este quadro de pontuações pode ser utilizado em qualquer jogo de pericia por si programado. Como exemplo, escreva um número maior do que 100, seguido do seu nome, e em pouco tempo o computador calculará e imprimirá a nova tabela de pontuações. O programa foi escrito por David Perry.

```

1 REM QUADRO DE PONTUACOES
2 REM © DAVID PERRY 1983
3 REM
4 REM Num=Numero de pontuacoes
5
6 REM
7 REM
8 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: OLS
10 LET num=10: DIM n(num+1): D
IM n#(num+1,0): FOR n=1 TO num:
LET n(n)=1100-(n*100): LET n#(n)
="SPECTRUM": NEXT n
15 OLS: INPUT "Para exemplo,
indique pontuacao Pontuacao
:";sc
20 IF sc<=n(10) THEN GO TO 200
30 LET num=11: IF sc>n(num) T
HEN INPUT "8 iniciate! ";P$: IF
LEN P#>8 THEN GO TO 20
35 PRINT AT 7,0: FLASH 1: BRIG
HT 0: INK 7: PAPER 2:"
Um momento .
40 IF sc>n(num) THEN LET n(nu
m)=sc: LET n#(num)=P#
50 FOR a=1 TO (num-1): LET b#=
n#(a): LET c#=n#(a+1): LET b=n(a
): LET c=n(a+1): IF b<c THEN LET
n(a)=c: LET n(a+1)=b: LET n#(a)
=c#: LET n#(a+1)=b#
60 NEXT a: FOR n=1 TO num-1: I
F n(n)<n(n+1) THEN GO TO 50
70 NEXT n
80 OLS
90 PRINT AT 2,6:"QUADRO DE PON
TUACOES"
100 PRINT AT 3,6:"=====
=====
110 FOR n=1 TO num-1: PRINT AT
n+5,7: INK 6:"("; INK 2;n: INK 6
:")"AT n+5,12: INK 7;n(n): PRIN
T AT n+5,17: INK 6;n#(n): NEXT n
120 PLOT 0,0: DRAW 000,0: DRAW
```

```

0,175: DRAW -255,0: DRAW 0,-175:
PRINT AT 17,0: PAPER 2: INK 7)
Carregue num# tecla.": LET i=0
130 LET i=i+1: IF i>7 THEN LET
i=0
140 PRINT AT 2,6: INK 1:"QUADRO
DE PONTUACOES"
150 BORDER 0,0,1,1*7: PAUSE 2: IF I
NKEY#="" THEN GO TO 130
160 GO TO 10
170 PRINT AT 10,4:"Pontuacao #U
180 BRIGHT 1:
190 PRINT AT 10,6:"Carregue num
# tecla."
200 PAUSE 0: RUN

```

FICHEIROS

Este interessante programa de ficheiros foi escrito por Michael Briggs. Armazena dados de uma forma bastante compacta, assegurando um uso bastante eficaz da memória disponível.

Quando o programa é executado, é-lhe perguntado o seu nome, o título do ficheiro, os títulos dos campos, etc. Note que nenhuma entrada pode ter um comprimento superior a 256 caracteres, e que um campo deve ter 28 caracteres ou menos.

É então impresso um menu, indicando as opções seguintes:

1. Iniciar ficheiro: Esta opção faz correr o programa, eliminando todos os dados que contenha.
2. Introduzir registos: Esta opção pede-lhe que introduza um registo, campo a campo. Em seguida colocará esse registo em ordem alfabética relativamente ao resto do ficheiro. Se cometer um erro ao introduzir qualquer registo, escreve simplesmente «Q» para voltar ao menu.
3. Procura de registos: Esta opção verifica todo o ficheiro, procurando uma cadeia de letras ou um registo numerado.
4. Imprimir todos os registos: Esta opção imprime o primeiro campo de todos os registos, atribuindo um número específico a cada um deles.
5. Gravar ficheiro: Esta opção grava ficheiro em cassette, usando o nome anteriormente atribuído àquele. Se o título não for válido, a gravação é realizada sob o nome «Index».

6. Estado: Esta opção indicar-lhe-á vários factos sobre o estado do programa, como o espaço disponível, a quantidade de registos, etc.

O leitor verificará que apesar de tudo isto parecer complexo, o programa é, de facto, bastante fácil de utilizar:

```

1 DIM Michael Briggs
2 POKR 23500,40: BORDER 0: IN
K 1: PAPER 7: CLS: DIM #$(256)
3: LET N=1: LET C=0
4: INPUT "Nome...": N#
5: INPUT "Apelido...": P#
6: INPUT "Titulo do ficheiro:
": T#
7: INPUT "Numero de campos: "
8:
9: DIM #$(1,20): FOR a=1 TO F:
PRINT "Titulo do campo: ";a: IN
PUT b#: LET #$(a,2 TO )=b#: LET
#$(a,1)=CHR# LEN b#: PRINT ":-"
INK 2:b#: PRINT : NEXT a
10 PAUSE 50
110 POKR 23500,40: BORDER 0: CL
S: INK 0: PLOT 0,0: DRAW 255,0:
DRAW 0,175: DRAW -255,0: DRAW 0
,-175: INK 0
120 PRINT INK 0: BRIGHT 1: AT 1,
10-LEN T#/2:T#:"MENU"
130 PRINT AT 3,0:" No.1 Inicia
r ficheiro."
140 PRINT AT 5,0:" No.2 Acresc
entar registo."
150 PRINT AT 7,0:" No.3 Procur
ar registo."
160 PRINT AT 9,0:" No.4 Imprim
ir registos."
170 PRINT AT 11,0:" No.5 Grava
r ficheiro."
180 PRINT AT 13,0:" No.6 Estad
o do ficheiro."
190 PRINT INK 2: AT 15,0:"© Mich
ael Briggs"
200 PRINT INK 1: AT 17,11-(LEN C
#+LEN #)/2:"FICHEIRO DE "C#":
"15#
210 INPUT "Opcao...": C
220 CLS
230 INK 1
240 IF 0<C AND C<8 THEN GO TO C
*1000
250 GO TO 100

```

```

1000 PRINT AT 10,0;"Confirma, ";
C#;"?";
1010 INPUT C#
1020 IF C#(1)="0" OR C#(1)="5" T
HEN RUN
1030 GO TO 100
1040 INK 0; FOR a=1 TO 7
1050 PRINT AT a*N-1,13-(CODE a#(
a,1)/2);"Indicave ";a#(a,2 TO )
1060 NEXT a
1070 PRINT AT 21,0; INK 1;"Escri
va o nome do jogador a seguir."
1080 LET a#=""
1090 FOR a=1 TO 7
1100 PRINT AT a*N-1,13-(CODE a#(
a,1)/2);"Indicave ";a#(a,2 TO 1+
CODE a#(a))
1110 INPUT F#
1120 IF F#="0" THEN GO TO 100
1130 LET a#=#+CHR# LEN F#+F#
1140 PRINT AT a*N-1,13-(CODE a#(
a,1)/2);"Indicave ";a#(a,2 TO 1+
CODE a#(a))
1150 PRINT INK 0; AT a*N,16-LEN F
#/#);F#
1160 NEXT a
1170 LET X#=1: FOR a=1 TO co: LE
T X#=#(X1+2 TO X1+CODE Z#(X1+1)
+1)
1180 IF X#>a#(2 TO ) THEN GO TO
1190
1200 LET X1=X1+CODE Z#(X1)
1210 NEXT a
1220 LET a#=#CHR# (LEN a#+1)+a#;
LET Z#(2 TO Z+(LEN a#))=a#
1230 LET Z=#Z+LEN a#
1240 LET co=co+1: CLS : GO TO 20
60
1250 LET a#=#CHR# (LEN a#+1)+a#
1260 LET Z#(X1+LEN a# TO Z+(LEN a
#))=#Z#(X1 TO Z)
1270 LET Z#(X1 TO X1+LEN a#-1)=
a#
1280 GO TO 2131
1290 REM Busto
1300 PRINT TAB 0;"PROCURA DE B00
ASTOS"; AT 2,0;"B00 creva..."; AT 3
,0;" 0" se quiser o busto de 100
lato" se quiser o busto de 1000
lato"; INPUT "Nome ";K#
1310 LET B#="" : IF K#(1)="0" TH

```

```

EN LET B#=#(1); LET K#=#(2 TO
3300 PRINT TAB 0;"Nome=");K#
3310 LET ret=3340
3320 LET x=1: FOR a=1 TO co
3330 LET x1=x; LET v#=#(x TO (C
ODE Z#(x)+x-1))+"; IF B#="N" T
HEN GO TO 3360
3340 FOR J=1 TO LEN v#-LEN K#
3350 IF v#(J TO J-1+LEN K#)=K# T
HEN GO TO 3360
3360 NEXT J
3370 LET X=x1+LEN v#-1: NEXT a
3380 PRINT TAB 0; INK 0;"Carregue n
um tecla."; PAUSE 0; GO TO 100
3390 PRINT TAB 0; INK 1;"Entrada no.
.";a; LET x2=x1: LET x1=x1+1: F
OR k=1 TO 7
3400 PRINT INK 2;a#(k,2 TO CODE
a#(k)+1);";"; PRINT INK 1;TAB
3410 CODE Z#(X1);Z#(X1+1 TO X1+COD
E Z#(X1))
3420 LET X1=X1+1+CODE Z#(X1)
3430 NEXT k
3440 LET X1=X2: GO TO ret
3450 IF a<>VAL K# THEN GO TO 334
0
3460 LET ret=3350
3470 GO TO 3300
4000 PRINT "Qualquer tecla inter
ompe o rela- mento de ima
gem."
4010 PAUSE 0
4020 PRINT AT 10,16-LEN t#/#); IN
K 2);t#
4030 INK 1; PRINT
4040 PRINT
4050 LET x=1: FOR a=1 TO co
4060 PRINT AT 21,0; POKE 20602,-
1
4070 PRINT "0000"; AT 21,4-LEN (C
ODE a#(a));";";Z#(x+2 TO x+1+CODE
Z#(x+1))
4080 LET x=#x+(CODE Z#(x))
4090 IF INKEY#<>" " THEN GO TO 40
50
4070 NEXT a
4075 PRINT TAB 0;"Carregue num
um tecla."
4080 PAUSE 0; GO TO 100
4090 PRINT "O ficheiro sera gra
vado sob o nome: "; PRINT AT 10

```

```

16:LEN t#>2; FLASH 4;t#
0000 IF t#="" OR LEN t#>10 THEN
PRINT AT 10,1;"          IND
EX          ": SAVE "INDEX" L
LINE 100: GO TO 100
0010 SAVE t# LINE 100
0020 GO TO 100
0030 PLOT 0,0: DRAW 250,0: DRAW
0,175: DRAW -250,0: DRAW 0,-175
0040 PRINT AT 1,10-(LEN t#)/2; 0
RIGHT 1; INK 2;t#;" ESTADO"
0050 PRINT AT 0,0;"No momento: "
;INT ((PEEK 23672+256*PEEK 2367
70)+55536*PEEK 23674)/255);" segs.
0060 PRINT AT 6,10-(LEN s#+LEN c
#)/2;"FICHEIRO DE ";c#" ";s#
0070 PRINT AT 7,0;"No. de regist
os: ";c
0080 PRINT AT 9,2;"Espaco livre: "
;05000-N
0090 PRINT AT 11,4;"Caracteres n
o ficheiro: ";N
0070 PRINT AT 13,3;"Programa © M
ICHAEL BRIGGS"
0075 PRINT AT 20,6;"Carregue num
e tecla."
0080 PRINT AT 3,2;"No momento: "
;INT ((PEEK 23672+256*PEEK 2367
3)+55536*PEEK 23674)/255);" segs.
0085 IF INKEY$<>"" THEN GO TO 61
00
0090 GO TO 0080
6100 GO TO 100

```

PROGRAMAÇÃO ESTRUTURADA

ESBOÇO DE UM PROGRAMA

Escrevi já muitos artigos e capítulos de livros com o objectivo de melhorar a capacidade de programar daqueles que os lêem. Mas de cada vez que apresento uma lista das «coisas que devem ser feitas quando se programa», ou explico o conteúdo desses textos a alguém, recorro a uma velha história de um lavrador.

Uma vez foi abordado por um jovem que vendia cursos por correspondência sobre agricultura. Disse-lhe este: «Não compreende que se tirar este curso ficará a saber muito mais sobre o seu trabalho?». O lavrador respondeu: «Nem sequer consigo trabalhar tão bem como sei...»

O mesmo acontece, parece-me, com os conselhos sobre programação. Sei perfeitamente que não programa «tão bem quanto sei». Muitas vezes contrário as regras todas, entrando directamente na codificação de um programa complexo sem sequer me lembrar de que existe uma «programação estruturada».

Sugiro portanto que o leitor tenha presente, ao ler este apêndice, que não acredito verdadeiramente que você — ou qualquer outra pessoa — vá aproveitar as ideias aqui apresentadas como regras fixas, que devam ser obedecidas em todas as circunstâncias. A melhor forma de considerar o que aqui se apresenta, ou qualquer outro material do mesmo tipo que encontre, consiste em lê-lo cuidadosamente e fazer uma avaliação própria do interesse de cada sugestão dada. Depois, aplique as ideias que lhe parecem mais apropriadas.

A ideia básica que desejo transmitir neste apêndice é que se pode estruturar um programa a partir de um esboço prévio deste.

A ideia é simples, mas muito valiosa na medida em que nos permite escrever programas complicados, como muitos dos apresentados neste livro. Como é óbvio, o leitor talvez esteja já a criar programas complexos sem precisar da ideia aqui sugerida. Mas mesmo que tal aconteça, sugiro que pense cuidadosamente no que está contido neste apêndice, tentando descobrir algo que possa facilitar o seu trabalho.

A ideia fundamental da programação estruturada está contida na frase «programação de cima para baixo». Isto sugere que se começa a programar definindo o objectivo geral do programa que se quer escrever. Em seguida tomam-se algumas notas — cada uma das quais acabará por ser transformada numa subrotina — que cobrirão cada uma das tarefas que o computador deverá realizar.

Para aclarar melhor este último parágrafo vou dar um exemplo concreto. Vejamos então os passos que devem ser dados quando se pretende escrever um programa sobre o Jogo do Galo (note que a programação estruturada produz listagens muito mais extensas do que o habitual. No entanto, este maior comprimento do programa é contrariado pelo facto de os programas assim escritos serem muito fáceis de emendar e melhorar, e de a sua estrutura ser fácil de compreender por outros programadores. É provável que o leitor não possa dizer o mesmo de alguns dos seus programas actuais.

DEFINIR TABULEIRO INICIAL

Lance do Computador

- (A) VERIFICAR SE O QUADRADO CENTRAL ESTÁ VAZIO; SE NÃO COLOCAR NELE UMA PEÇA
- (B) VERIFICAR SE EXISTE UM MOVIMENTO GANHANTE DO COMPUTADOR, E SE HOVER EXECUTÁ-LO
- (C) VERIFICAR SE O JOGADOR GANHARÁ NO LANCE SEGUINTE, E SE SIM BLOQUEAR O SEU MOVIMENTO
- (D) SE NÃO HOVE MOVIMENTO DO JOGADOR, VERIFICAR SE O COMPUTADOR PODE FAZER UM MOVIMENTO. SE SIM MOVER, SE NÃO DECLARAR UM EMPATE.

IMPRIMIR TABULEIRO

ACEITAR O LANCE DO JOGADOR

IMPRIMIR TABULEIRO

VERIFICAR SE O JOGADOR GANHOU. SE SIM PARAR

«GO TO» LANCE DO COMPUTADOR.

Como o leitor pode verificar se dedicar alguns momentos ao estudo deste esboço, pode saber exactamente quais os passos da execução feita pelo computador; as decisões que deverá tomar; e os resultados alternativos destas decisões.

O leitor já deu o seu primeiro passo no sentido da escrita de um programa do Jogo do Galo. A propósito, note que não interessa se já sabe ou não a forma concreta como o computador irá executar as funções necessárias. Apenas interessa que saiba por agora quais as operações gerais que deverão ser executadas por uma dada ordem.

O segundo passo deste processo consiste em transformar o esboço feito numa série de chamamentos de subrotinas. Em linguagens mais estruturadas do que a Basic empregue no Spectrum torna-se relativamente fácil chamar um certo número de subrotinas no interior de um ciclo sem fim, usando uma estrutura como a DO/WHILE ou a REPEAT/UNTIL. Estas permitem ao programador repetir uma série de passos de programa indefinidamente, até ser satisfeita uma determinada condição ou se alterar o estado de um indicador («pointer» ou «flag»).

Na Basic por nós empregue, no entanto, teremos de nos contentar com o humilde GO TO. Grande parte das críticas à GO TO, que os «programadores sérios» vêem como o maior insulto à verdadeira arte da programação alguma vez feito, devem-se às primeiras e relativamente primitivas versões da Basic, nas quais só era possível usar uma IF/THEN seguida de GO TO. Isto significava que o programa saltava repetidamente de uma instrução para outra, conduzindo a uma codificação «spaghetti» extremamente difícil de interpretar.

A segunda fase consiste portanto em transformar o esboço já preparado numa série de chamamentos de subrotinas, realizados num ciclo sem fim produzido por uma GO TO da seguinte forma:

```

10 REM Nome
20 GO SUB 9000: REM inicializa
30 GO SUB 1000: REM lance do c
computador
40 GO SUB 6000: REM imprimir i
mensagem
50 GO SUB 2000: REM aceitar la
nce do jogador
60 GO SUB 8000: REM imprimir i
mensagem
70 IF jogador-nao-ganha AND co
mputador-nao-ganha THEN GO TO 30
80 PRINT "PARABENS"
```

Como se pode ver, temos já a estrutura geral de um programa do jogo do Galo, apesar de ainda não termos qualquer noção de como funcionará de facto o programa. Podemos começar agora a escrever o programa partindo do princípio.

Existem outras vantagens deste método de «esboço». Se existir alguma coisa que não possamos, nesta fase, programar (como o modo como o computador descobrirá quem ganha), poderemos simplesmente colocar uma instrução PRINT no local apropriado, por exemplo PRINT «Verificar vitória», e continuar a desenvolver outras partes do programa. Depois, de cada vez que o computador deva verificar a vitória, imprimirá esta mensagem. Isto significa que se pode desenvolver o programa sem ficar retido por uma qualquer dificuldade secundária que não possa ser resolvida no momento.

A segunda vantagem ocorre no final do desenvolvimento do programa, na fase de correcção de erros. Sempre considereei esta como a mais frustrante, e de muitos pontos de vista a menos compensadora fase do trabalho de programação. Se bem que consiga muitas vezes pôr o programa em funcionamento em condições razoáveis com alguma rapidez (se bem que o meu programa de xadrez, em Basic, tenha demorado a corrigir cerca de seis meses), desde um funcionamento em condições razoáveis até um verdadeiro bom funcionamento pode demorar bastante tempo. Quando porém se constrói o programa da forma indicada, a fase de correcção de erros é bastante simplificada.

Por exemplo, o leitor poderá verificar no caso do programa do Jogo do Galo que o computador tende a ignorar o canto inferior do lado direito, quando um movimento para esta posição lhe permitiria ganhar, ou bloquear um movimento ganhante do jogador. Da sequência de rotinas definida anteriormente sabemos que o lance do computador deve ser realizado num qualquer ponto entre as linhas 1000 e 1999. Isto permite-nos imediatamente concentrar a procura do erro. Se cada subrotina tiver sido programada também como uma série de subrotinas, cada uma delas construída da forma já citada, será muito fácil determinar a instrução errada.

Vejamos um pouco melhor esta ideia da transformação de cada subrotina numa série de outras subrotinas. Observemos então a subrotina que se inicia na linha 1000, aquela em que o computador realiza de facto o seu movimento, dado que esta é a mais importante (e a mais difícil de programar) de todas.

A subrotina pode iniciar-se do seguinte modo:

```
2000 REM lance do computador
2010 LET mover=0: REM se passa a
2020 GO SUB 2030: REM Verificar
2030 GO SUB 2040: REM Verificar
2040 GO SUB 2050: REM Verificar
2050 IF mover=1 THEN RETURN
2060 GO SUB 2070: REM Verificar
2070 IF mover=1 THEN RETURN
2080 REM O retorno com mover=0
2090 REM significa que nao e' possivel
2100 RETURN
```

```
2060 IF mover=1 THEN RETURN
2070 GO SUB 2080: REM Verificar
2080 GO SUB 2090: REM Verificar
2090 IF mover=1 THEN RETURN
2100 REM O retorno com mover=0
2110 RETURN
```

Como já disse, os programas construídos usando o «ciclo sem fim de subrotinas», juntamente com «subrotinas dentro de subrotinas» pode aumentar bastante o comprimento do programa. No entanto, não há normalmente necessidade de nos preocuparmos com a memória disponível (ao contrário do que acontecia com o ZX81 ou o TS1000, quando a contagem dos bytes restantes se transformava no aspecto dominante do trabalho de programação). O maior trabalho tido a escrever um maior número de instruções será completamente recompensado pela maior clareza das listagens obtidas, e pelo menor tempo perdido na correcção dos erros que inevitavelmente ocorrerão.

O leitor verificará certamente que, se tiver a preocupação de começar por fazer um esboço da estrutura do programa a escrever, todo o processo de construção do programa em causa se tornará de facto muito mais simples do que se não tiver esse cuidado. Certamente observará também que o tempo gasto na construção do programa será muito mais produtivo do que no caso de se lançar imediatamente no trabalho de programação, sem dedicar algum tempo previamente à definição de um esboço daquilo que lhe interessa fazer.

O melhor conselho que lhe posso dar consiste portanto em ignorar o velho agricultor, e tentar desde já «programar tão bem quanto sabe»...

ÍNDICE

PREFÁCIO	
JOGOS DE PERÍCIA	
Protector	9
Bem alto	14
Bombardeiro	21
Pássaros	23
Helicóptero	32
Serpente	37
Corridas	41
Zombies	46
AVENTURAS	
O Gigante e a Cidade	52
Portas do Destino	68
JOGOS COM GRÁFICOS MÓVEIS	
Corrida ao ouro	91
Tarântula	97
A Selva	104
Lógica	109
Na horta	109
Fantasmas	111
Sapo	114
TEMPOS LIVRES	
Pares de cartas	120
Submarino	123
O Carrasco	132

JOGOS NO ESPAÇO	
Fuga entre as estrelas	135
Space Trek	135
Sonda espacial	141
Tempestade lunar	148
A O PRAZER DO LABIRINTO	
Gerador de labirintos	152
Labirintos irregulares	154
Muros do suspense	158
Muro do suspense	161
Labirinto a 2 níveis	162
Labirinto móvel	165
Labirinto rolante	169
Labirinto a 3 dimensões	177
Labirinto a 3 dimensões — II	184
JOGOS PARA DOIS JOGADORES	
Quadrados	189
Quatro em linha	196
Tanques	198
AUXILIARES EM CÓDIGO-MÁQUINA	
Monitor de memória	203
Guardar Screens em código-máquina	208
Guardar Screens em Basic	209
Irlanda	210
Copiador	212
UTILITÁRIOS E DEMONSTRAÇÕES	
Biorrítmos	216
Pintura	218
Timothy Leary	219
Teste de cores	221
Demonstração de portas lógicas	221
Sons	222
Arco-Íris	223
Regressão Linear	226
Quadro de Pontuações	229
Ficheiros	230
PROGRAMAÇÃO ESTRUTURADA	
Esboço de um programa	235

Este livro foi fotocomposto
 Por José Maria Marques—Lisboa
 e impresso na
 Tipografia Nunes, Lda. — 4000 Porto
 para a
EDITORIAL PRESENÇA, LDA.