Conhecer Melhor

Uma colecção sem fronteiras temáticas

JOGOS PARA O ZX SPECTRUM

MAIS DE 20 PROGRAMAS, CONCEBIDOS PARA PROPICIAREM MUITAS HORAS DE FASCINANTE DESAFIO

Excalibur contra o inimigo (um combate de naves espaciais), *Ascot* (corrida de cavalos), *Aterragem em Marte*, *Condutor em 3D* (um verdadeiro teste, mesmo para profissionais), *O vendedor de vídeo* (uma iniciação ao mundo dos negócios e do lucro), *A corrida da morte*, *O monstro de Loch Ness*, *A aranha e as moscas*, *Jogo das Damas* — eis alguns dos fascinantes jogos que o presente livro põe ao seu dispor.

Jogos para o ZX Spectrum permitir-lhe-á aperfeiçoar a sua capacidade como programador, além de lhe fornecer um léxico breve de termos de informática, uma bibliografia seleccionada e algumas informações sobre o modo de melhorar e tornar mais complexos os programas apresentados.

O SEU COMPUTADOR AGRADECER-LHE-Á A COMPRA DESTE LIVRO!

15

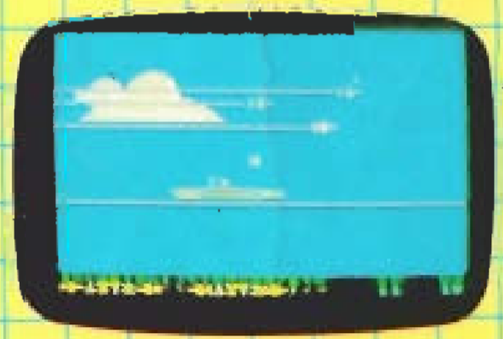
Conhecer Melhor JOGOS PARA O ZX SPECTRUM

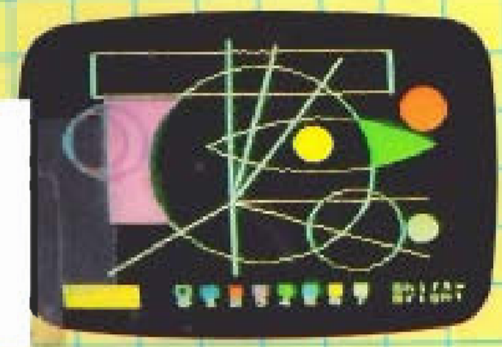
Conhecer Melhor

Peter Shaw

JOGOS PARA O ZX SPECTRUM

Publicações Dom Quixote





Conhecer Melhor

Uma colecção sem fronteiras temáticas...

Títulos publicados:

1. A ÍNDIA
   G. N. S. Raghavan
2. O JAZZ
   Morley Jones
3. JOGOS E PROGRAMAS EM BASIC
   João Carlos Azinhais
4. AS IDEIAS CONTEMPORÂNEAS
   Jean-Marie Domenach
5. CONTRACEPÇÃO, GRAVIDEZ E ABORTO
   P. Bello, C. Dolto e A. Schiffmann
6. O BASIC
   A. Checroun
7. A ARTE DE PERDER TEMPO
   João Esteves da Silva
8. COMO DEIXAR DE FUMAR
   Jean-Luc Roger
9. OS PROBLEMAS SEXUAIS
   G. Zwang e H. Dermange
10. A ASTROLOGIA EGÍPCIA
    François Suzzarini
11. COMO DEIXAR A DROGA
    Pierre Rey
12. MICHAEL JACKSON
    Caroline Latham
13. GUIA PRÁTICO DO VÍDEO
    Jean-Didier Graton e Eric Vincent
14. GUIA DAS DOENÇAS SEXUAIS
    Jean-Claude Bourret
15. JOGOS PARA O ZX SPECTRUM
    Peter Shaw

PETER SHAW

JOGOS PARA O ZX SPECTRUM

PUBLICAÇÕES DOM QUIXOTE
LISBOA
1986

FICHA:

Título: *Jogos para o ZX Spectrum.*

Autor: *Peter Shaw.*

Colecção: *Conhecer Melhor, n.º 15.*

Título Original: *Games for your ZX Spectrum.*

Tradução: *Maria do Carmo Ravara Cary, a partir da edição inglesa publicada por Virgin Books Ltd., Londres.*

Capa: *Fernando Felgueiras.*

Ilustrações: *Sue Walliker.*

1.ª edição: *Abril de 1986.*

Edição n.º: *15 CM 961.*

Depósito legal n.º: *9413/85.*

Fotocomposição, paginação e fotólitos: *Textype — Artes Gráficas, Lda..*

Impressão e acabamento: *Barbosa & Santos, Lda., em Abril de 1986.*

Distribuição: *Diglivro, Rua do Ataíde, Pátio do Pimenta, 28, Lisboa, e Movilivro, Rua do Bonfim, 98, r/c, Porto.*

Para Sue Powell

AGRADECIMENTOS

O autor quer agradecer a Maureen Gates pela sua colaboração na redacção deste livro; a Alan Dennis, Steven Cunning e Michael Merrifield pelas suas sugestões de programas; e a Tim Hartnell e Clive Gifford pelo apoio fornecido. Agradece também a Mark e David Palmer, que contribuíram à sua maneira para que este livro fosse possível.

ÍNDICE

PREFÁCIO DO EDITOR ORIGINAL

O computador é um parceiro que está sempre à sua disposição. Jogos gráficos movimentados, jogos de perícia, jogos de palavras e adivinhas são alguns dos que pode disputar com ele.

Este livro apresenta uma grande variedade de jogos. Os programas foram feitos por alguns dos jovens programadores mais talentosos que trabalham actualmente no nosso país e ilustram diferentes possibilidades de solução dos problemas da programação.

O estudo destas listagens sugerir-lhe-á muitos truques e técnicas que poderá aplicar na sua própria programação. E depois de dominar os programas na forma como são apresentados no livro, pode querer melhorá-los. O «programa perfeito» é uma coisa que não existe e por isso as suas capacidades de programador permitir-lhe-ão certamente aperfeiçoar estes jogos.

E agora volte a página e passe para os programas. Espero que se divirta tanto a jogar estes jogos como nós nos divertimos a preparar o volume.

Londres, Março de 1983

TIM HARTNELL

PREFÁCIO DO AUTOR

O Spectrum é um computador espantoso; em comparação com ele o seu irmão mais novo, o ZX81, é lento e desajeitado, pois a única maneira de jogar um jogo do tipo Invasores do Espaço é estudando o código da máquina. O Spectrum tem a vantagem de permitir programar jogos de acção em BASIC.

Os programas deste livro são principalmente jogos. Evitei os programas usados geralmente para encher espaço como os Biorritmos ou o Bloco de Desenho, não porque tenha alguma coisa contra eles, mas apenas porque são publicados com regularidade em quase todas as revistas de microcomputadores.

Depois de ter experimentado estes programas não se fique por aí — tente melhorá-los. Se achar que algum dos caracteres definidos pelo utilizador ganha em ser corrigido, não hesite em fazê-lo!

Stanwell, Middlesex, Novembro de 1982

PETE SHAW

MÉTODOS E TÉCNICAS DE PROGRAMAÇÃO

A maioria dos meus programas começam de maneira semelhante:

```
10 REM Nome do programa
20 GOSUB 9000: REM UDGs
30 GOSUB 8000: REM Variáveis
40 GOSUB 7000: REM Desenhar no écran
```

Além de se tornar assim mais fácil acrescentar linhas quando se dá ao computador uma instrução GO TO, o computador procura de linha em linha até encontrar a linha certa. Quando se está a programar um jogo de acção — em que a rapidez é essencial — este método de fraccionamento em sub-rotinas acelera o movimento.

Sugestões e truques

O Spectrum tem os seus segredos, que nem sempre são revelados com clareza no manual de instruções, e alguns deles nem sequer são mencionados. Pode usar-se nas instruções

PRINT o símbolo (#) para imprimir em várias posições no écran:

```
PRINT # 1; «Isto é impresso na parte de baixo do
écran»; PAUSE Ø
```

Como se pode ver, # 1 imprime na metade inferior do écran, possibilitando uma imagem de 24 linhas.

Os microcomputadores ZX são os únicos microcomputadores que conheço que só podem INPUT no fundo do écran; mas usando INPUT AT pode introduzir-se informação em qualquer parte do écran.

```
INPUT AT 22,Ø; AT Ø,Ø; «Como se chama»;
LINE a$; AT 1Ø,Ø; «Que idade tem»; (a$)
« »; a; AT 15,Ø; (a$); «tem x anos»; a; AT 2Ø,Ø;
«Carregue em ENTER para continuar»; b$
```

Este método tem alguns inconvenientes. Em primeiro lugar, e como se pode ver, o comprimento das linhas é um problema; em segundo lugar a cor de BORDER tem de ser a mesma do que a de PAPER, dado que a parte de baixo do écran abrange quase toda a parte de cima. (Digo quase toda porque fica uma faixa da parte de cima do écran que é muito visível, a menos que as cores de paper e border sejam iguais.)

Tentei usar UDGs[1] na medida do possível em todo o livro. Servi-me para esse efeito de um acessório muito útil chamado «Print'n'Plotter Jotter». Trata-se de um bloco com duas grelhas em cada página: a grelha inferior tem 32 × 22 e utiliza-se para planear a imagem; a grelha superior tem 64 × 44 e uso-a para definir os meus caracteres. Pode usar também papel milimétrico, mas os quadradinhos são tão pequenos que os UDGs tornam-se indistintos e é difícil defini-los.

[1] UDG — User-Defined Graphic ou Gráfico Definido pelo Utilizador (N. T.).

SALVAR OS OVOS

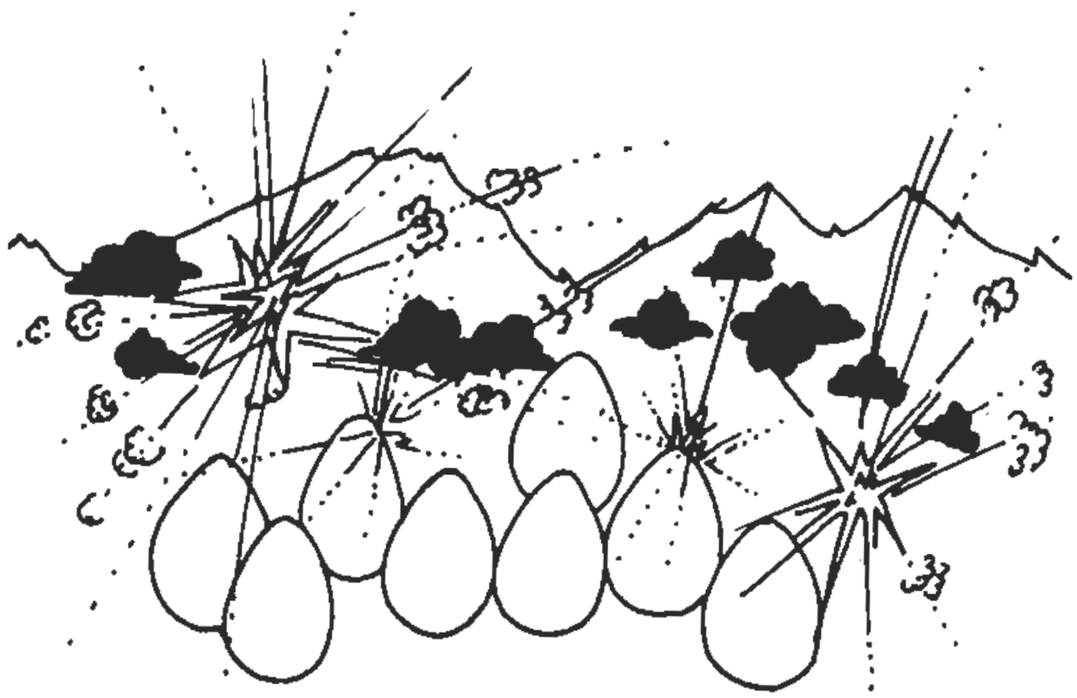
Neste jogo você é o Salvador dos Ovos e a sua tarefa consiste em salvar pobres ovos inocentes das garras de poderes maléficos e tirânicos, que tentam impedir os seus movimentos disparando raios laser. Para recolher os ovos tem de chegar ao 'X' que está por baixo das caixas dos ovos, no alto do écran; depois de ter apanhado os ovos o seu homem fica verde. Tem de escapar aos raios laser e pôr o ovo na caixa passando por cima do '-'. Depois de ter recolhido seis ovos ganha 100 pontos e um novo fornecimento deles.

Este jogo tem determinadas regras:

1. Não pode passar por cima de mais nada além do 'X' ou do '-'; quando isso acontece perde uma vida.
2. Quando é atingido por um raio laser, ou quando perde uma vida de qualquer outra maneira, perde automaticamente o ovo transportado; se não leva nenhum ovo perde só uma vida.
3. Só pode transportar um ovo de cada vez; se leva um ovo e tenta recolher outro passando por cima de um 'X', o 'X' desaparece e não é possível recolher o ovo da caixa que fica acima do 'X'.

4. Tem três vidas, que da primeira vez que joga não duram mais do que dez segundos, mas com a prática conseguem-se resultados razoáveis.

Utilize as teclas do cursor para controlar os seus movimentos.



```
 10 REM Salvar os ovos.
 20 REM ********************
 30 REM Peter Shaw
 40 REM Ideia original de Micheal Merrifield
 50 REM ********************
 55 LET hi = 0
 60 GO SUB 9000
 70 GO SUB 8000
 80 GO SUB 7000
150 PRINT AT h, v; "  "
160 LET h = h + (INKEY$ = "6" AND h <21) - (IN
KEY$="7" AND h >0): LET v=v + (INKEY$="8" AND
v <31) - (INKEY$="5" AND v>0)
```



```
170 IF SCREEN$ (h,v) < >"  " THEN GO SUB 6000
180 IF RND >. 5 THEN GO SUB 6500
190 PRINT AT h, v; INK 3 + ovos; "☤"
191 BEEP .008,0
200 IF Vidas < 1 THEN PRINT AT 21,0; "Está
morto": GO TO 1500
210 PRINT AT 21,0; "Vidas"; Vidas
230 PRINT AT 21,14; "Ovos recolhidos "; tot
998 GO TO 150
999 STOP
1500 BEEP 1,4: BEEP 1,4: BEEP .3,4: BEEP 1.2,4: BEEP
.75,7: BEEP .5,6: BEEP 1,6: BEEP .3,4: BEEP .7,4: BEEP
.5,3: BEEP 1,4
1501 IF sc>hi THEN LET hi = sc
1510 INPUT "Carregue em ENTER para jogar    no
vamente"; LINE a$: CLS: GO TO 70
2000 IF ovos=1 THEN RETURN
2005 PRINT AT 1,v; "    "; AT 2,v; INK 6; "▙▟"
2010 LET ovos=1
2015 BEEP .1,10
2020 RETURN
2500 IF ovos=0 THEN PRINT AT h,v; INK 1; "-": LET
h=h+1: RETURN
2510 PRINT AT 17,v; INK 6; "▲"; AT 18,v; OVER 1;
INK 6; "▀"
2515 LET ovos=0
2516 PRINT AT 19,v; "  "
2517 BEEP .1,20: BEEP .15,15
2520 LET tot=tot+1: IF tot=6 THEN GO TO 9500
2530 RETURN
6000 LET b$=SCREEN$ (h,v)
6010 IF b$="X" THEN GO TO 2000
6020 IF b$="-" THEN GO TO 2500
6030 LET vidas=vidas-1
6031 LET ovos=0
6040 PRINT AT h,v; FLASH 1; INK 3; "  ": BEEP
.1,-10: FOR p=1 TO 20: NEXT P: RETURN
6500 GO TO (INT (RND*3) +1) *100+6500
6600 PLOT 9,m1: DRAW INK 2; BRIGHT 1; 245,0
6605 GO SUB 6900
```

```
6606 PAUSE 2
6610 DRAW OVER 1; -248,0: PLOT OVER 1; 254, m1:
RETURN
6700 PLOT m2,167: DRAW INK 2; BRIGHT 1;0, -120
6705 GO SUB 6900
6706 PAUSE 2
6710 DRAW OVER 1;0, 120: RETURN
6800 PLOT m3,167: DRAW INK 2; BRIGHT 1;0, -120
6805 GO SUB 6900
6806 PAUSE 2
6810 DRAW OVER 1;0,120: RETURN
6900 IF SCREEN$ (h,v) < > "  " THEN GO TO 6000
6910 RETURN
7000 LET n=0: PRINT AT 0,0; "pontos"; sc; TAB 22;
"máximo "; hi
7005  FOR a=1 TO 4
7010 PRINT AT 1,n; "    ";
7020 FOR b=1 TO e (a)
7030 PRINT INK 6; " ▲ ";
7040 NEXT b
7050 PRINT AT 2,n; "    ";
7060 FOR b=1 TO e (a)
7070 PRINT INK 6; " ▼ ";
7080 NEXT b
7090 OVER 1
7095 PRINT AT 2,n; "    ";
7100 FOR b=1 TO e (a)
7110 PRINT INK 6; " ▙▟ ";
7120 NEXT b
7122 PRINT AT 3,n; "     ";
7125 FOR b=1 TO e(a): PRINT INK 6; "X "; :NEXT b
7130 LET n=n + (e (a)*2) +2
7140 NEXT a
7150 PRINT AT 0,9; INK 2; "▼"; AT 0,21; "▼"; AT 4,0;
"▶"
7160 PRINT AT 18,8; INK 1; "▙▟ ▙▟ ▙▟ ▙▟ ▙▟
▙▟"
7165 PRINT AT 19,8; INK 1; "-   -   -   -   -
- "
7170 OVER 0
```

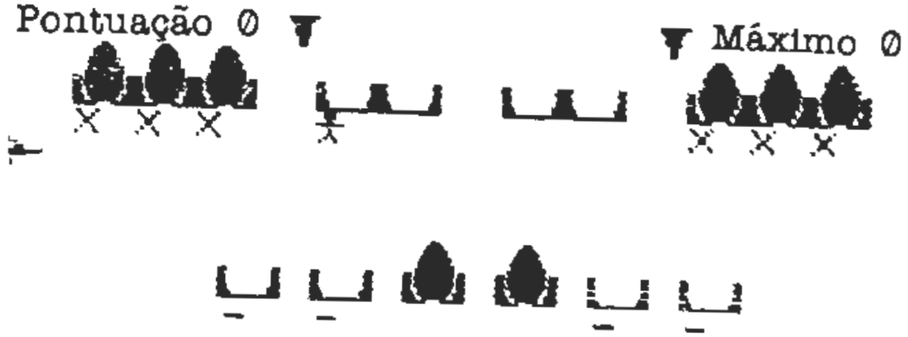


```
7990 RETURN
8000 BORDER 0: INK 7: PAPER 0: CLS
8010 DIM e (4)
8020 RESTORE 9120: FOR z=1 TO 4: READ e (z):
NEXT z
8030 LET h=10: LET v=15
8040 LET sc=0: RANDOMIZE
8060 LET m1=17*8+4: LET m2=9*8+4: LET
m3=21*8+4
8070 LET ovos=0
8080 LET vidas=3
8090 LET tot=0
8100 RETURN
8989 REM **********************************
8990 REM Caracteres gráficos.
8991 REM                ▲ Gráficos ab
8992 REM                ▲ Gráficos Cd
8993 REM               ▙▟ Gráfico ef
8994 REM                ▞ Gráfico g
8995 REM                ▶ Gráfico h
8996 REM                ▼ Gráfico i
8997 REM **********************************
9000 FOR a=USR "a" TO USR "j"-1
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 1,3,7,7,15,15,15,31
9040 DATA 128,192,224,224,240,240,248,248
9050 DATA 31,31,31,31,15,15,7,3
9060 DATA 248,248,248,248,240,240,224,192
9070 DATA 192,192,192,192,192,224,224,255
9080 DATA 3,3,3,3,3,7,7,255
9090 DATA 56,56,16,254,16,16,40,68
9100 DATA 128,192,240,255,255,240,192,128
9110 DATA 255,126,60,60,60,24,24,24
9111 REM **********************************
9120 DATA 3,2,2,3
9500 LET sc=sc+100
9505 LET h=10: LET v=15: LET tot=0: LET ovos=0
9510 PRINT AT 0,0; "Pontuação"; sc
9515 IF sc>hi THEN LET hi=sc
```

```
9520 FOR p=1 TO 31: BEEP .01,p: BEEP .006,-p:
NEXT p
9530 CLS
9540 GO TO 80
```

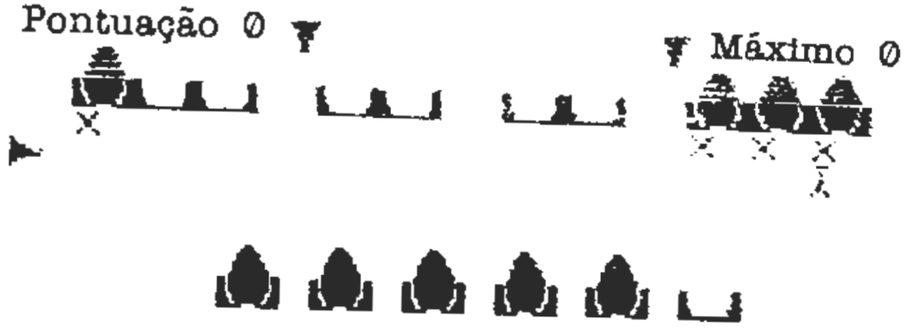
Vidas restantes 1

Ovos recolhidos 2



Vidas restantes 1

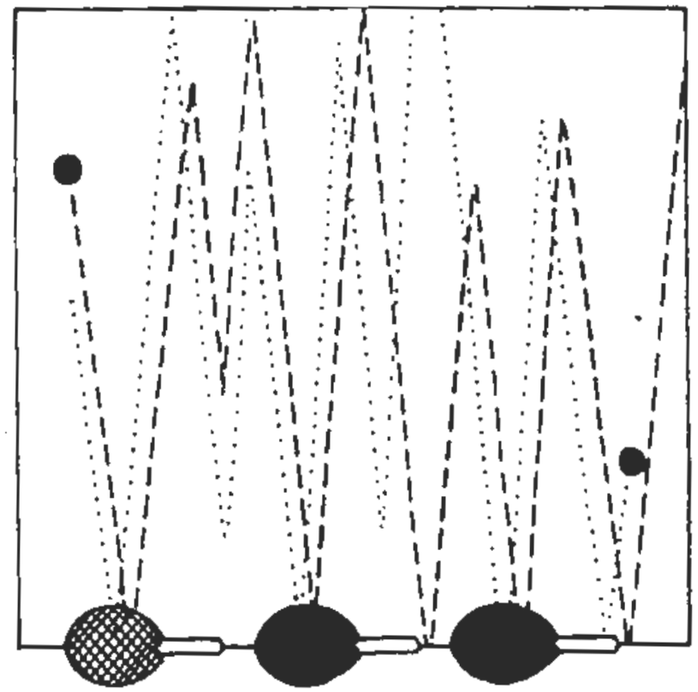
Ovos recolhidos 2

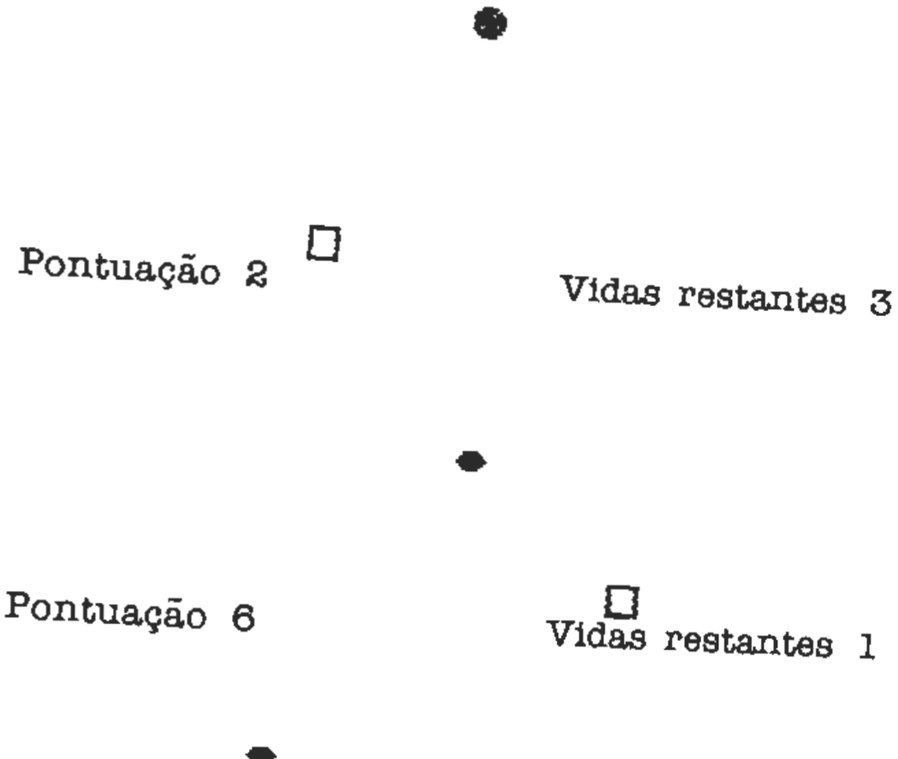


Você foi morto

JOGO DA BOLA

Neste jogo tem de fazer saltar a bola no écran; ganha pontos quando consegue bater na bola com a sua raqueta (o quadrado no fundo do écran). Mas se a bola bate no quadrado do lado perde uma vida. Tem três vidas e desafio-o a tentar bater o meu recorde de 20. Utilize a tecla 5 para os movimentos para a esquerda e a tecla 8 para os movimentos para a direita.





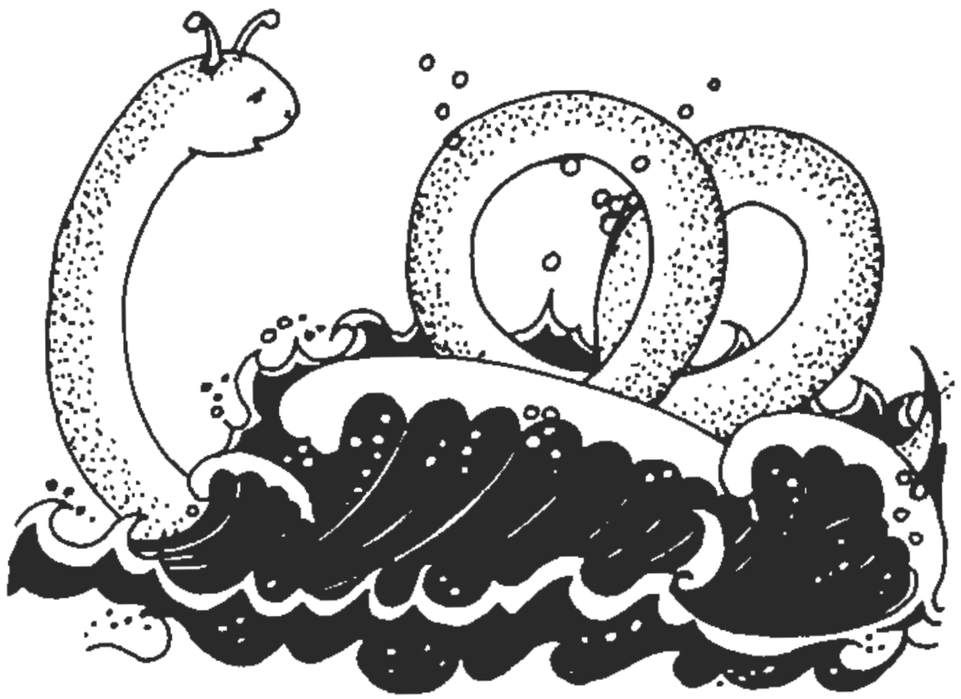

```
 10 REM Jogo da bola
 20 REM Peter Shaw
 30 REM Ideia original de Alan Dennis
 40 LET hi=0
 50 GO SUB 9000
 60 GO SUB 8000
 65 PRINT AT 21,15; "Vidas restantes"; Vidas
 70 PRINT AT 20,v; "  "
 80 LET v=v + (INKEY$="8" AND v <31) - (IN
KEY$="5" AND v>0)
 85 LET v=v+2* (INKEY$=CHR$ 9 AND v<31)
-2*(INKEY$=CHR$ 8 AND v>0)
 90 PRINT AT 20,v; INK 5; BRIGHT 1; "□"
100 PRINT AT a, b; "  "
110 LET a=a+m: IF a >19 OR a <1 THEN BEEP
.05,10: LET m=-m
```



```
120 LET b=b+n: IF b>30 OR b<1 THEN BEEP
.05,15: LET n=-n
130 PRINT AT a,b; INK c; "●"
140 IF a=0 OR a=20 THEN PRINT AT a,b; INK c;
"●"
150 IF b=0 OR b=31 THEN PRINT AT a,b; INK c;
"●"
160 IF a=19 AND b=v THEN PRINT AT a,b; INK c;
"●": BEEP .05,15: LET sc=sc+2: LET m=-m
170 IF a=20 AND b=v THEN GO SUB 500
180 PRINT AT 21,0; "Pontuação"; sc
190 GO TO 70
500 PRINT AT 20,v; FLASH 1; INK 2; "□"
510 LET vidas=vidas-1
515 PRINT AT 21,15; "Vidas restantes "; vidas
520 FOR p=1 TO 20: BEEP .008, -p: NEXT p
525 IF vidas=0 THEN GO TO 600
526 LET a=INT (RND*6) +2: LET b=INT (RND*27)
+2
530 RETURN
600 PRINT AT 2,10; FLASH 1; BRIGHT 1; "FIM DO
JOGO"
610 PRINT "Você marcou "; sc
620 IF sc>hi THEN LET hi=sc
630 PRINT "Pontuação máxima de hoje "; hi
640 INPUT "Carregue em ENTER para jogar    nova
mente"; LINE a$: GO TO 60
8000 BORDER 1: PAPER 0: INK 7: CLS
8005 LET v=15: LET a=INT (RND*5) +2: LET b=INT
(RND*27) +2
8010 LET sc=0:LET m=1: LET n=2
8020 LET c=6: LET vidas=3
8990 RETURN
9000 FOR a=USR "a" TO USR "d" +7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 60, 126, 255, 255, 255, 255,126,60
9040 DATA 0,0,0,60,126,255,126,60
9050 DATA 8,28,62,62,62,62,26,6
9060 DATA 255,129,129,129,129,129,129,255
```

O MONSTRO DE LOCH NESS

Este jogo é uma variante do tiro aos pratos; os alvos passam no écran acima de si. Marca pontos acertando nos alvos, e quanto mais alvos atingir, mais pontos marca. Se conseguir acertar numa fila inteira ganha um bónus. Utilize a tecla 5 para os movimentos para a esquerda, a tecla 8 para os movimentos para a direita e 0 para fazer fogo.




```
BÓNUS 2000 PONTOS

PONTUAÇÃO 3160          MÍSSEIS 0

FIM DO JOGO

Você marcou 3160

Pontuação máxima de hoje 3160
```

```
 10 REM NESSIE
 20 LET hi=0: GO SUB 9000
 30 GO SUB 8000
 40 LET s=0: LET mis=20: GO SUB 7000
 50 PRINT AT 5,0; a$' b$
 60 LET h=h + (INKEY$="8" AND h<30) - (IN
KEY$="5" AND h>0)
 70 IF INKEY$="0" THEN LET mis=mis-1: GO SUB
1000
 75 IF mis < 1 THEN GO TO 500
 80 PRINT AT 20,h; INK 6; "  "
 90 LET b$=b$ (2 TO) + b$(1)
100 LET a$=a$ (LEN a$) + a$(TO (LEN a$) -1)
101 PRINT AT 0,0; PAPER 5; INK 0; "PONTUAÇÃO ";
s, "MÍSSEIS "; mis; "  "
110 IF b$="
          "  THEN GO SUB 2000
120 IF a$="
          "  THEN GO SUB 3000
140 GO TO 50
500 PRINT AT 10,12; FLASH 1; "GAME OVER"
```

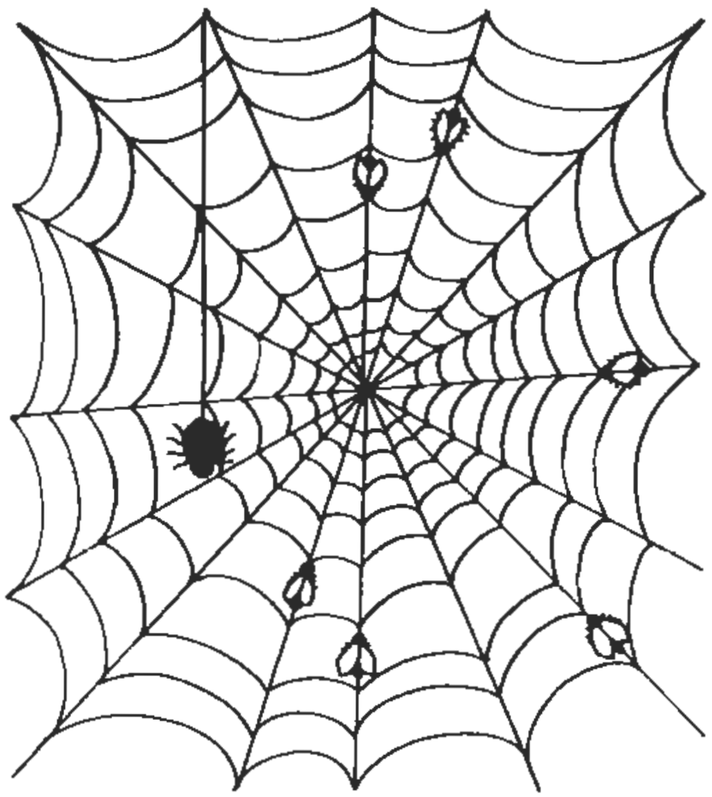
```
 501 PRINT AT 0,0; PAPER 5; INK 0; "PONTUAÇÃO ";
s, "MÍSSEIS "; mis; "  "
 510 PRINT AT 12,10; "Você marcou "; s
 520 IF s >hi THEN LET hi=s
 525 PRINT'' TAB 4; "Pontuação máxima de hoje "; hi
 530 INPUT ''CARREGUE EM ENTER PARA JO
GAR   NOVAMENTE"; LINE a$: GO TO 30
1000 LET b=h+2
1010 FOR f=19 TO 4 STEP -1
1012 PRINT AT f,b; "*"
1019 BEEP .001, - (f-30)
1020 PRINT AT 5,0; a$' ' b$
1030 LET h=h + (INKEY$="8" AND h <30) - (IN
KEY$="5" AND h>0)
1040 PRINT AT 20,h; INK 6; "  "
1050 LET b$=b$(2 TO) + b$(1)
1060 LET a$=a$(LEN a$) + a$(TO (LEN a$) -1)
1070 IF f=5 THEN IF a$ (b) < >"  " THEN GO TO
4000
1080 IF f=7 THEN IF b$(b) < >"  " THEN GO TO
5000
1090 PRINT AT f, b; "  "
1100 NEXT f: RETURN
2000 PRINT AT 10,7; FLASH 1; "BONUS 1000
PONTOS"
2010 LET b$="
   "
2011 LET s=s+1000
2015 BEEP .1,20: BEEP .2,15
2020 FOR p=1 TO 150: NEXT p
2040 PRINT AT 10,7; "                    ":RETURN
3000 PRINT AT 10,7; FLASH 1; "BONUS 2000
PONTOS"
3010 LET a$="
   "
3011 LET s=s+2000
3015 BEEP .1,20: BEEP .2,15
3020 FOR p=1 TO 150: NEXT p
3030 PRINT AT 10,7;"                    ":RETURN
4000 LET a$(b-4 TO b+4)="          "
```

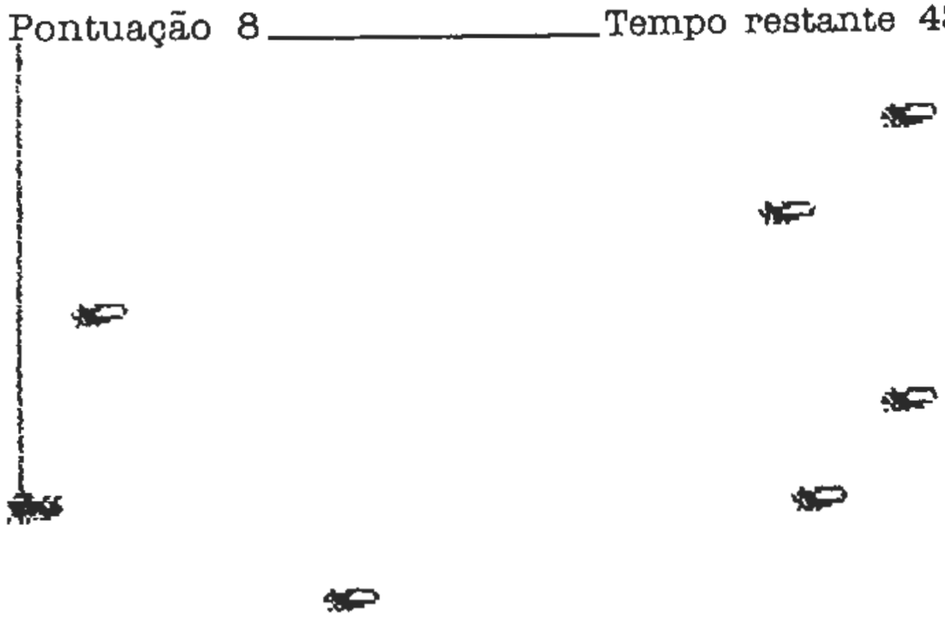


```
4010 LET s=s+20
4011 BEEP .05, -5
4020 RETURN
5000 LET b$(b-4 TO b+4) ="          "
5010 LET s=s+10
5011 BEEP .05,5 -5
5020 RETURN
7000 BORDER 0: PAPER 1: INK 4: CLS
7010 FOR a=0 TO 3: PRINT PAPER 5; "
                    ";: NEXT a
7020 RETURN
8000 LET a$="
   "
8010 LET b$="
   "
8020 LET h=15
8022 LET s=0
8030 RETURN
8900 REM ********************************
8910 REM
8920 REM              Modo gráfico
8930 REM
8940 REM        —        abc
8950 REM
8960 REM        —        defg
8970 REM
8980 REM    *   —        h
8990 REM
8999 REM ********************************
9000 FOR a=USR "a" TO USR "h" +7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,0,0,127,255,255,255,255
9040 DATA 0,24,60,255,255,255,255,255
9050 DATA 0,0,0,254,255,255,255,255
9060 DATA 0,96,248,252,127,63,15,7
9070 DATA 1,15,31,63,255,252,240,192
9080 DATA 128,240,248,255,255,63,31,7
9090 DATA 2,15,31,255,255,252,248,192
9100 DATA 24,60,60,24,60,36,0,0
```

A ARANHA E AS MOSCAS

Você é uma aranha muito esfomeada pendurada num fio à esquerda do écran, e as seis moscas gordas que aparecem são o seu alvo principal. Tem 99 segundos para comer o maior número possível de moscas. Utilize a tecla 6 para subir e a tecla 7 para descer.






```
10 REM A aranha e as moscas
20 LET hi=0: GO SUB 9000
30 GO SUB 8000
40 PRINT AT 0,0; "Pontuação ";sc
50 PRINT AT v,0; "  "
60 LET v=v+(INKEY$="6" AND v <21) - (IN
KEY$="7" AND v>2)
70 IF SCREEN$ (v,2) < >"  " THEN GO SUB 1000
80 PRINT AT v-1,0; "¦"
90 PRINT AT v,0; INK 6; "  "
100 FOR a=1 TO 6
110 LET f(a) = f(a) - INT (RND*2): IF f(a) < 1
THEN PRINT AT a*3,1; "  ": LET f(a)=28
120 PRINT AT a*3,f(a); INK 4; "  "
130 NEXT a
140 PRINT AT 0,14; "Tempo restante "; INT tempo;
"  ": LET tempo=tempo-.5:IF tempo<0THEN GO TO
2000
150 GO TO 50
990 STOP
1000 LET sc=sc+2
1010 PRINT AT v,2; "    "
1020 LET f(v/3) = 28
```

```
1030 PRINT AT 0,0; INK 7; "Pontuação "; sc
1040 RETURN
2000 PRINT AT 5,9; "FIM DO JOGO!"
2010 PRINT'' TAB 6; "Você marcou "; sc
2020 IF sc>hi THEN LET hi=sc
2030 PRINT'' "Pontuação máxima de hoje "; hi
2040 INPUT "Carregue em"; FLASH 1; " ENTER";
FLASH 0; "para jogar novamente"; LINE a$: GO TO 30
8000 BORDER 1: PAPER 1: INK 7: CLS
8010 LET v=10
8020 DIM f(6)
8030 FOR a=1 TO 6: LET f(a)=28: NEXT a
8040 LET sc=0
8050 RANDOMIZE
8060 FOR a=1 TO v-1: PRINT AT a,0; " | ": NEXT a
8070 PLOT 0,168: DRAW 255,0
8080 LET tempo=99
8990 RETURN
8991 REM
8992 REM                    Modo gráfico
8993 REM
8994 REM                    gráfico c
8995 REM                    gráfico ab
8996 REM
8997 REM                    gráfico de
8998 REM
8999 REM ************************
9000 FOR a=USR "a" TO USR "e" +7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,127,127,149,148
9040 DATA 0,108,248,220,252,248,96,0
9050 DATA 16,16,16,16,16,16,6,16
9060 DATA 0,17,63,94,111,23,45,0
9070 DATA 0,252,2,2,252,224,0,0
```


ASCOT

Neste programa pode ser o dono de um cavalo de corrida sem os inconvenientes de ter de o alimentar ou limpar a cavalariça. Recebe um cavalo de corrida e 50 libras, bem como a possibilidade de ganhar uma fortuna na pista de corridas com essas 50 libras.

Pode jogar este jogo com um máximo de 4 amigos; ganha o jogador que acabar o jogo com mais dinheiro. Para ganhar dinheiro aposte no cavalo que escolheu (só pode apostar nesse cavalo durante as cinco corridas): as probabilidades de cada cavalo ganhar aparecem antes de cada corrida. Se já não tem dinheiro e o seu cavalo ganha uma corrida, recebe um bónus em dinheiro, para poder continuar a jogar na corrida seguinte.

Corrida número 1

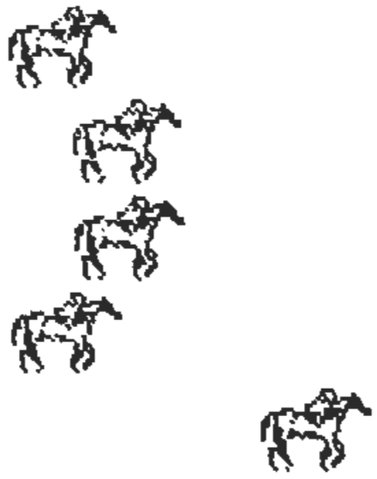
As probabilidades de Red Gin
ganhar são de 1:1

As probabilidades de Sumley Gilds
ganhar são de 1:1

As probabilidades de Sparkle
ganhar são de 6:1

As probabilidades de Danny Boy
ganhar são de 8:1

As probabilidades de Shergart
ganhar são de 3:1



Cavalo: — Red Gin

Dono: — Peter

Você tem £50

O vencedor é Danny Boy

Dono: Micheal

que ganha 200

Cavalo: — Shergart

Dono: — Bill

Você tem 0

Não tem dinheiro por isso não pode apostar

```
 10 REM Ascot
 20 GO SUB 9000: REM UDG's
 30 GO SUB 8000
 40 FOR r=1 TO 5
 50 CLS
 60 PRINT AT 0,0; "Número de corrida "; r
 70 DIM d(p)
 80 FOR a=1 TO p: LET d(a)=INT (RND*10) +1
 90 PRINT' "As probabilidades de"; h$(a)
100 PRINT " ganhar são "; d(a); ":1"
110 NEXT a
120 GO SUB 5000
130 PRINT AT 0,0; #1; "Carregue em qualquer tecla
para  continuar"
140 PAUSE 0
150 CLS: CIRCLE INK 2;240,165,10: PLOT 240,155:
DRAW INK 2;0, - 140: FOR a=1 TO p: PRINT AT a*3+
3,31; a: NEXT a
155 DIM c (p)
160 FOR a=1 TO p
```

```
 170 PRINT AT a*3+2,c(a); a$; AT a*3+3,c(a); b$; AT
a*3+4,c(a); c$
 180 LET c(a)=c(a)+(1/d(a)+INT (RND*2))
 185 IF c(a) >25 THEN GO TO 250
 190 NEXT a
 200 FOR a=1 TO p: BEEP .008,c(a)
 210 PRINT AT a*3+2,c (a); a$; AT a*3+3, c (a); d$;
AT a*3+4, c (a); c$
 220 LET c(a)=c(a)+(1/d (a) +INT (RND*2))
 225 IF c(a) >25 THEN GO TO 250
 230 NEXT a
 240 GO TO 160
 250 LET w$=h$ (a)
 260 PAUSE 100: CLS
 270 PRINT "O vencedor é "; w$
 280 PRINT "Dono "; n$(a)
 290 LET ws=d(a)*s (a)
 300 PRINT' "que ganha "; ws: LET m (a)=m
(a)+ws+s (a)
 310 FOR z=1 TO 50: BEEP .008,z: BEEP .008, -z:
NEXT z
 320 NEXT r
 330 CLS
 340 LET tot=0: PRINT "No fim do jogo "
 350 FOR a=1 TO p
 360 PRINT n$(a); " tem "; m(a)
 370 IF m(a) > tot THEN LET tot= m(a): LET win=a
 380 NEXT a
 390 PRINT' "O vencedor é "; n$ (win)
 400 PRINT "em "; h$ (win)
 410 PRINT "Com "; m(win)
 420 INPUT "Carregue em enter para outro jogo"; LI
NE a$: GO TO 30
 990 STOP
5000 FOR z=1 TO 50: BEEP .002*p,z: BEEP .008, -z:
NEXT z
5010 FOR a= 1 TO p
5020 CLS
5030 PRINT "Cavalo: — "; h$(a)
5040 PRINT' "Dono — "; n$(a)
```



```
5050 PRINT' "Você tem "; m(a)
5055 IF m(a) <1 THEN PRINT "Você não tem dinhei
ro por isso não pode apostar": PAUSE 150: NEXT a:
RETURN
5060 INPUT "Quanto vai apostar na corrida seguin
te "; s(a): IF s(a) >m (a) THEN GO TO 5060
5070 LET m(a)= m(a) - s(a)
5080 NEXT a
5090 RETURN
8000 BORDER 0: PAPER 0: INK 7: CLS: BEEP .1,10:
BEEP .2,15
8010 LET a$="                ": REM abcd
8020 LET b$="                ": REM efgh
8030 LET c$="                ": REM ijkl
8040 LET d$="                ": REM emno
8050 LET c$="                ": REM ipq
8060 PRINT AT 1,12; "ASCOT"
8070 INPUT "Quantos jogadores " (Max 5)"; LINE p$:
IF p$<"1" OR p$>"5" THEN GO TO 8070
8080 LET p=VAL p$: DIM s(p): DIM m(p): DIM n$
(p,10): DIM h$(p,15)
8090 FOR a=1 TO p
8100 PRINT AT 5,3; "Imprimir nome do jogador #"; a
8110 INPUT "Max 10 letras "; LINE n$(a)
8120 LET m(a)=50: NEXT a
8125 CLS
8130 RESTORE 9200: FOR a=1 TO p
8140 READ h$(a): PRINT n$(a); "o vencedor é" 'h$(a)
8150 NEXT a
8160 PRINT AT 0,0; #1; "Carregue em qualquer tecla
para continuar": PAUSE 0
8990 RETURN
9000 FOR a=USR "a" TO USR "q"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,0,0,0,0,0,0,3
9040 DATA 0,0,0,0,0,1,113,207
9050 DATA 0,28,18,42,87,132,15,238
9060 DATA 0,0,16,248,60,124,199,131
9070 DATA 5,13,13,9,9,9,9,0
```

```
9080 DATA 9,16,0,144,157,178,192,192
9090 DATA 221,209,33,1,194,242,11,10
9100 DATA 128,0,0,0,0,0,0,128
9110 DATA 3,2,2,2,2,1,0,0
9120 DATA 192,192,96,32,16,32,0,0
9130 DATA 13,28,25,19,18,8,0,0
9140 DATA 128,128,128,0,0,0,0,0
9150 DATA 16,0,16,144,177,167,64,128
9160 DATA 69,1,1,1,194,246,20,20
9170 DATA 128,0,0,0,0,0,0,0
9180 DATA 128,128,128,128,128,64,0,0
9190 DATA 28,28,24,24,20,12,10,0
9200 DATA "Red Gin","Sumley Gilds","Sparkle", "Danny
Boy","Shergart"
```


O COMILÃO

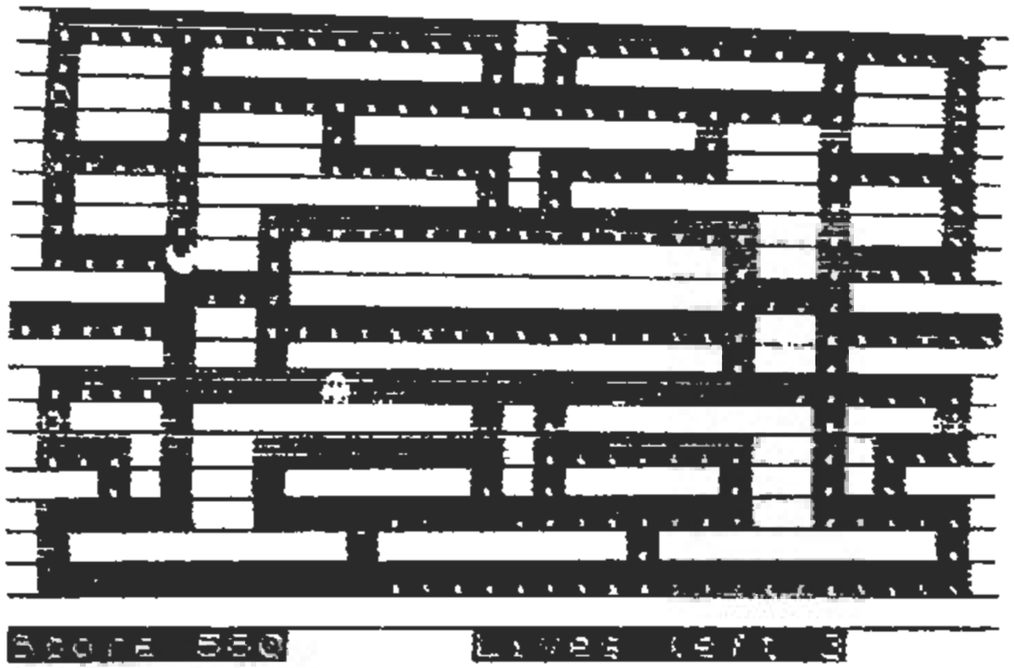
Este jogo é talvez o mais sensacional desta série. Trata da versão de um conhecido jogo de salão de jogos. Difere no entanto dele em dois aspectos principais, se bem que em todos os outros seja basicamente igual.

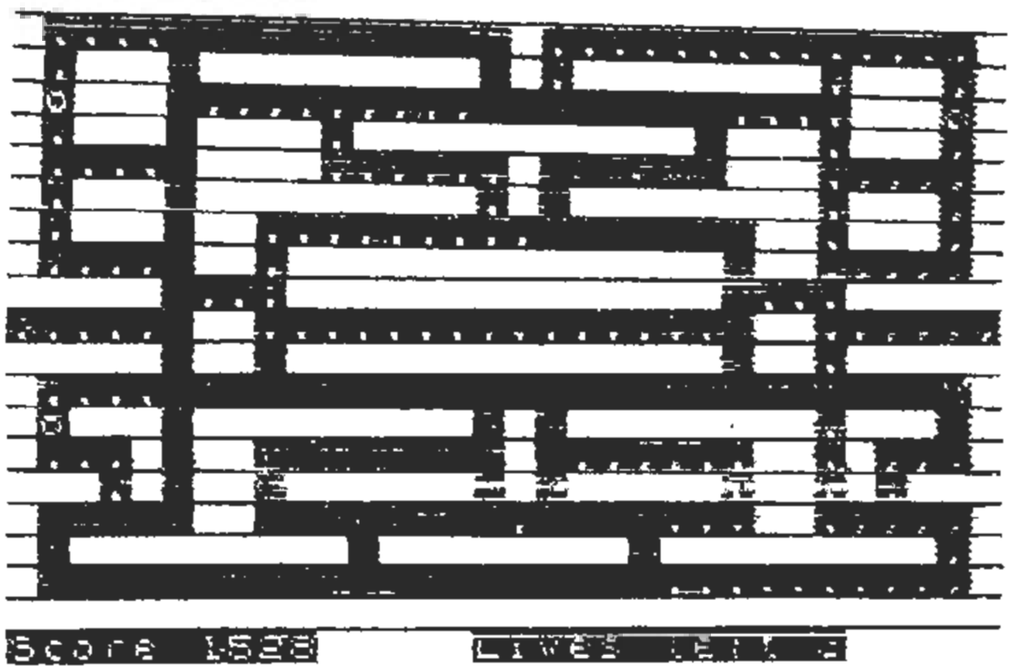
1. Só há um fantasma.
2. Quando come as bolas de energia dos cantos ganha um prémio em pontuação, e não a possibilidade de comer o fantasma.

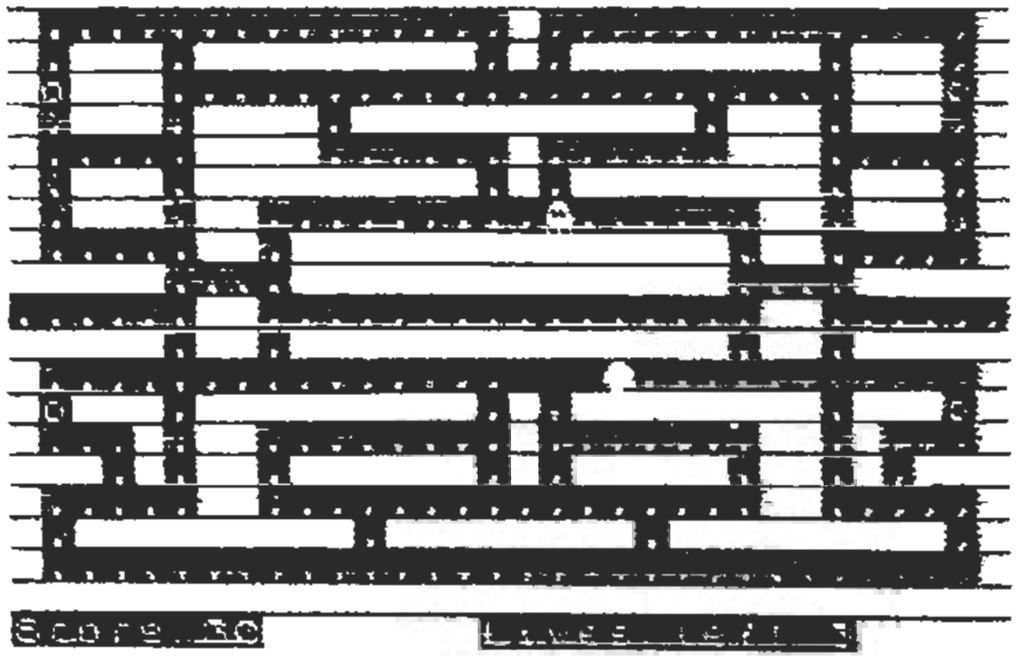
Utilize as teclas do cursor para controlar os seus movimentos.

Nota sobre este programa

Os gráficos das linhas 7010 e 7020 obtêm-se pressionando alternadamente Inv. video e True video, para obter um ponto final invertido e um traço true video (simbol shift 0).





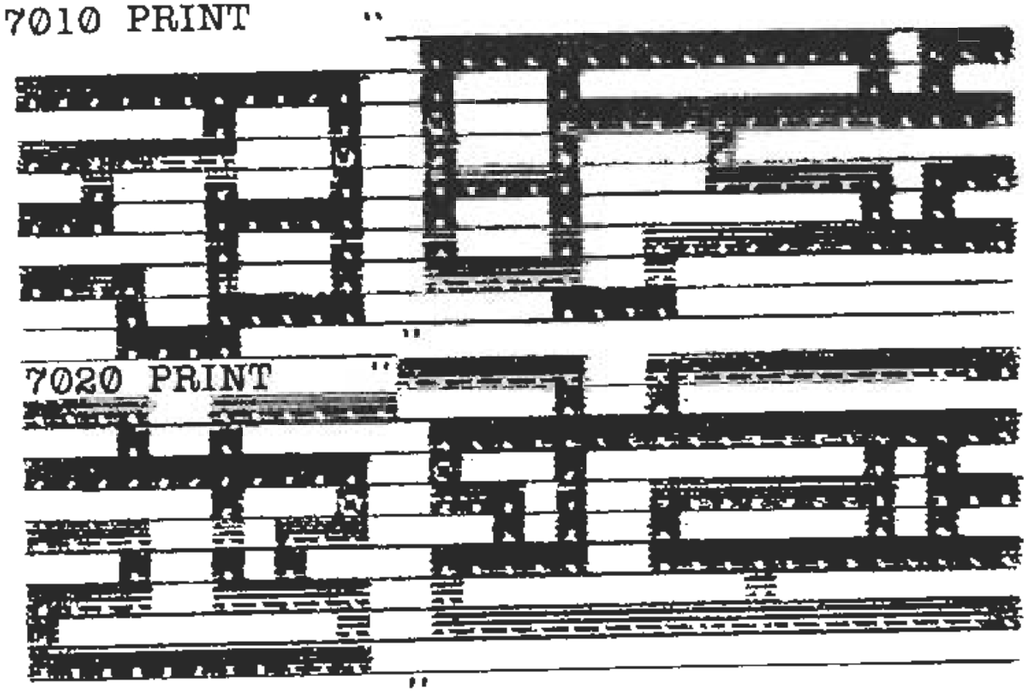


```
  10 REM O Comilão
  20 LET hi=0: GO SUB 9000
  30 GO SUB 8000
  40 LET w=12: LET h=16: Go SUB 7000: INVERSE
1: PRINT AT 20,15; "Vidas restantes "; vidas
  45 PRINT AT w,h; INVERSE 1;m$
  50 LET a$=INKEY$: IF a$<"5" OR a$>"8" THEN
GO TO 80
  60 IF a$< >"" THEN LET c1=VAL a$
  70 LET m$=("■" AND c1=7) + ("■" AND c1=8)
+("■" AND c1=6) + ("■" AND c1=5)
  80 PRINT AT v,h; " "
  85 IF h<1 THEN LET h=30
  86 IF h>30 THEN LET h=0
  90 LET v1=v: LET h1=h
 100 LET v=v + (c1=6) - (c1=7)
 110 LET h=h + (c1=8) - (c1=5)
 120 IF SCREEN$ (v,h)="—" THEN LET v=v1: LET
h=h1: PRINT AT v,h; m$: GO TO 150
```

```
 130 IF SCREEN$ (v,h)="." THEN LET sc=sc+10:
BEEP .008,10: LET count=count+1
 135 IF SCREEN$ (v,h) ="o" THEN GO SUB 6000
 140 IF SCREEN$ (v,h) =" " THEN GO TO 5000
 145 PRINT AT v,h; "●"; AT 20,0; "Pontuação "; sc
 150 PRINT AT z,x; " "
 160 LET z1=z: LET x1=x
 170 LET z=z+(m1=6) - (m1=7): LET
x=x+(m1=8)-(m1=5)
 180 IF SCREEN$ (z,x)=" " THEN LET m1=INT
(RND*4) +5: LET z=z1: LET x=x1: GO TO 160
 185 PRINT AT v,h; m$
 190 IF SCREEN$ (z,x) ="." THEN PRINT AT z1,x1;
"."
 195 IF SCREEN$ (z,x) ="o" THEN PRINT AT z1,x1;
"o"
 200 IF SCREEN$ (z,x) ="" THEN GO TO 5000
 210 PRINT AT z,x; PAPER 4; "■"
 230 IF vidas < 1 THEN GO TO 2000
 240 IF count=tot THEN LET count=0: GO TO 40
 250 GO TO 50
 990 STOP
2000 CLS
2010 INVERSE 0: PRINT ''TAB 5; "GAME OVER»
2020 PRINT'' "              Você marcou "; sc
2030 IF sc>hi THEN LET hi=sc
2040 PRINT "Pontuação máxima de hoje "; hi
2050 INPUT "Carregue em ENTER para jogar nova
mente"; LINE a$: GO TO 30
5000 INVERSE 1
5005 PRINT AT z,x; " "
5010 PRINT AT v,h; "●"
5020 BEEP .3,15
5030 PRINT AT v,h; "■"
5040 BEEP .3,11
5050 PRINT AT v,h; "●"
5060 BEEP .3,7
5070 PRINT AT v,h; "▲"
5080 BEEP .3,3
5090 PRNT AT v,h; "><"
```

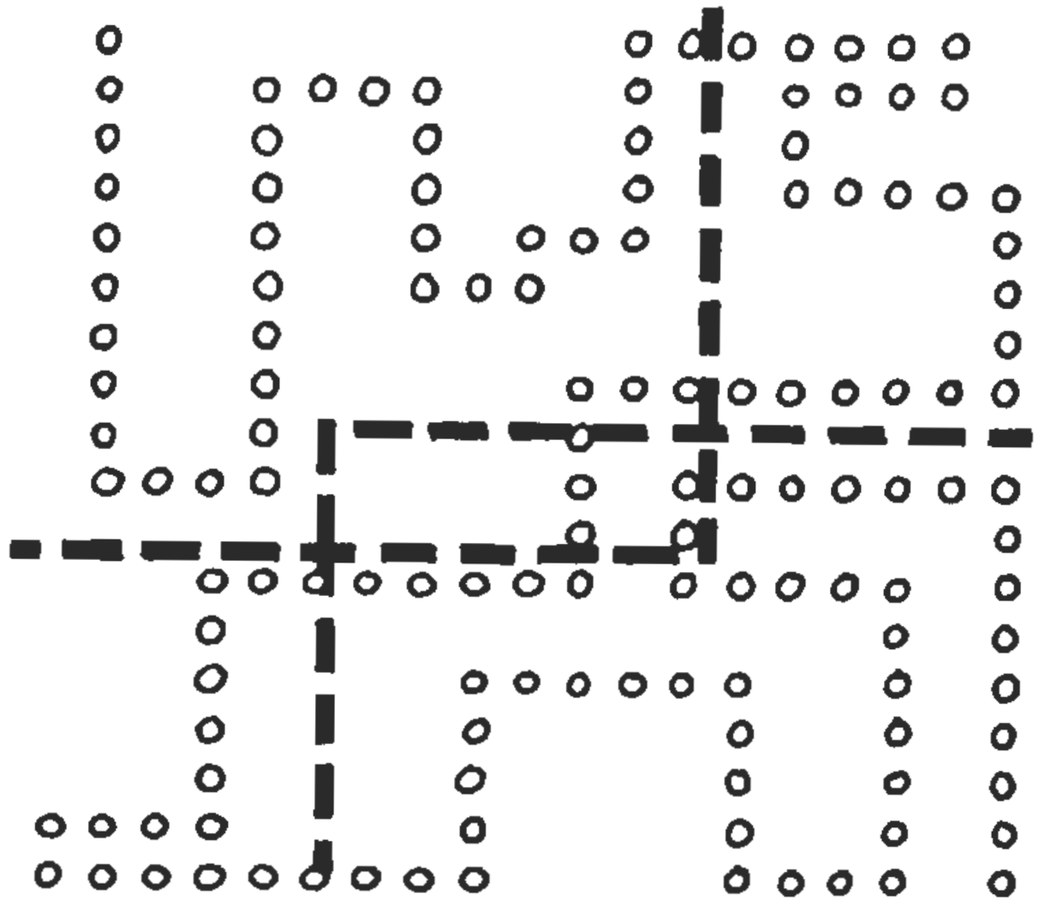


```
5100 BEEP .3,0
5110 PRINT AT v,h; " "
5120 BEEP .5, -5
5130 LET count=0; LET count=0: LET v=12: LET
h=16: LET vidas=vidas-1
5140 INVERSE 0: GO TO 40
6000 BEEP .008,15: BEEP .005, -15: LET sc=sc+INT
(RND*100) +200
6010 RETURN
7000 PAPER 0: CLS: INK 0: PAPER 6: BORDER 0
7010 PRINT      "
```



```
7020 PRINT      "
                "
7030 INK 0
7040 PAPER 6
7100 RETURN
8000 LET m$=" ■ "
8020 LET c1=8: LET m1=8
8030 LET sc=0
8040 LET z=7: LET x=15
8050 LET vidas=3
8060 LET count=0
8070 LET tot=266
8990 RETURN
9000 FOR a=USR "a" TO USR "i" +7
```

```
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255,126,60
9040 DATA 0,66,195,231,255,255,225,60
9050 DATA 60,126,248,240,240,248,125,60
9060 DATA 60,126,255,255,231,195,66,0
9070 DATA 60,126,31,15,15,31,126,60
9080 DATA 0,0,126,255,255,255,126,60
9090 DATA 0,0,0,0,16,24,60,60
9100 DATA 129,66,36,0,0,36,66,129
9110 DATA 28,62,42,107,127,127,109,73
9980 REM a  b  c  d  e  f  g  h  i
9990 REM ●  ●  ●  ●  ●  ●  ▲  ✕  ●
```



PISAR O RASTO

Neste jogo o computador e você deslocam-se no écran deixando um rasto e competindo entre si pelo espaço. Se pisar o rasto do computador ou bater no lado do écran ou se pisar o próprio rasto o computador marca um ponto. Mas o computador não é invencível e, com a prática, vai ter o prazer de o bater. Utilize as teclas do cursor para controlar os seus movimentos.

```
 10 REM Pisar o rasto
 20 LET hi=0
 30 GO SUB 8000
 40 LET a$=INKEY$
 50 IF a$>"8" OR a$<"5" THEN GO TO 70
 60 LET m=VAL a$: LET j=0
 70 LET h1=h1+(m=6) - (m=7)
 80 LET h2=h2+(m=8) - (m=5)
 90 LET b$=SCREEN$ (h1,h2)
100 BEEP .005,10
110 IF b$="_" THEN GO SUB 2000
120 LET j=1: PRINT AT h1,h2; INVERSE 1; "_"
130 PRINT AT h1,h2; INVERSE 1; INK 6; "_"
140 LET u=0
150 IF u=5 THEN GO SUB 1000
```

```
 160 LET c3=c1: LET c4=c2
 170 LET c1=c1+(n=6) - (n=7)
 180 LET c2=c2+(n=8) - (n=5)
 190 LET c$=SCREEN$ (c1, c2)
 200 IF c$="  " THEN LET n=u+6: LET c1=c3: LET
c2=c4: LET u=u+1: GO TO 150
 220 PRINT AT c1,c2; INVERSE 1; "_"
 240 GO TO 40
1000 PRINT AT c1, c2; FLASH 1; "_"
1010 LET hs=hs+1: FOR p=1 TO 30: BEEP .008,p:
NEXT p
1020 PRINT AT 0,0; "PONTUAÇÃO COMP "; cs; "
   HUMANO"; hs
1030 PRINT INVERSE 1; "_____________________
_____________"
1040 FOR i=1 TO 39
1050 PRINT INVERSE 1; "_"; INVERSE 0; "
                          "; INVERSE 1; "_"
1060 NEXT i
1070 PRINT INVERSE 1; "_____________________
_____________"
1080 LET c1=10: LET c2=10
1090 LET h1=10: LET h2=20
1100 IF hs=10 THEN GO TO 3000
1110 RETURN
2000 PRINT AT h1,h2; FLASH 1; "_"
2010 LET cs=cs+1: FOR p=1 TO 30: BEEP .008,p:
NEXT p
2020 PRINT AT 0,0; "PONTUAÇÃO COMP "; cs;
"      HUMANO"; hs
2030 PRINT INVERSE 1; "_____________________
_____________"
2040 FOR i=1 TO 19
2050 PRINT INVERSE 1; "_"; INVERSE 0; "
                          "; INVERSE 1; "_"
2060 NEXT i
2070 PRINT INVERSE 1; "_____________________
_____________"
2080 LET c1=10: LET c2=10
2090 LET h1=10: LET h2=20
```



```
2100 IF cs=10 THEN GO TO 3000
2110 RETURN
3000 PRINT AT 5,8;("COMPUTADOR " AND cs>hs) +
("HUMANO " AND hs>cs); "GANHA!!"
3010 INPUT "Carregue em enter para jogar nova
mente"; LINE a$
8000 LET j=1: LET c1=10: LET c2=10
8010 LET h1=10: LET h2=20
8020 LET hs=0: LET cs=0
8030 BORDER 0: PAPER INT (RND+6): INK 9: CLS
8040 PRINT "PONTUAÇÃO ";
8050 PRINT "COMP 00     HUMANO 00"
8060 PRINT INVERSE 1; "_____________________
_____________"
8070 FOR i=1 TO 19
8080 PRINT INVERSE 1; "_"; TAB 31; INVERSE 1; "_"
8090 NEXT i
8100 PRINT INVERSE 1; "_____________________
_____________"
8110 LET m=5: LET n=8: RETURN
```

VINTE-E-UM

Este jogo é uma versão ligeiramente simplificada do conhecido jogo de cartas do mesmo nome, mas igualmente emocionante. O objectivo do jogo consiste em aproximar-se o mais possível de uma pontuação de 21, mas sem a ultrapassar.

Começa por ser a sua vez de jogar. O computador mostra as suas duas primeiras cartas e você tem de decidir se quer outra carta — «Voltar» ou se quer ficar como está — «Ficar-se». Tem de decidir duas, três ou mais vezes. Se o seu total exceder os 21 pontos, o computador anuncia «você foi eliminado» e você perde 10 £ (a aposta por jogo). Depois de você ter decidido «Ficar-se» com uma pontuação inferior a 21, o computador dá as suas próprias cartas (que não ficam à vista). O computador anuncia então que foi eliminado ou que se fica («Eliminado» ou «Ficar-se») antes de anunciar quem ganhou o jogo.

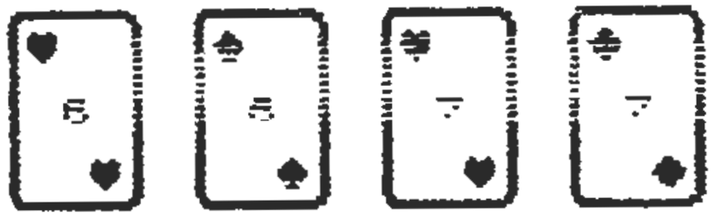
Um ás vale sempre um e os grupos de cinco cartas não contam. Quando se fartar de jogar interrompa o programa quando o computador vai dar cartas.


```
D — Dar
V — Voltar
F — Ficar-se
```

```
Dinheiro 90
```

```
ELIMINADO
Ganho o jogo!
        T — Voltar
        S — Ficar-se
```

```
Dinheiro 90
```



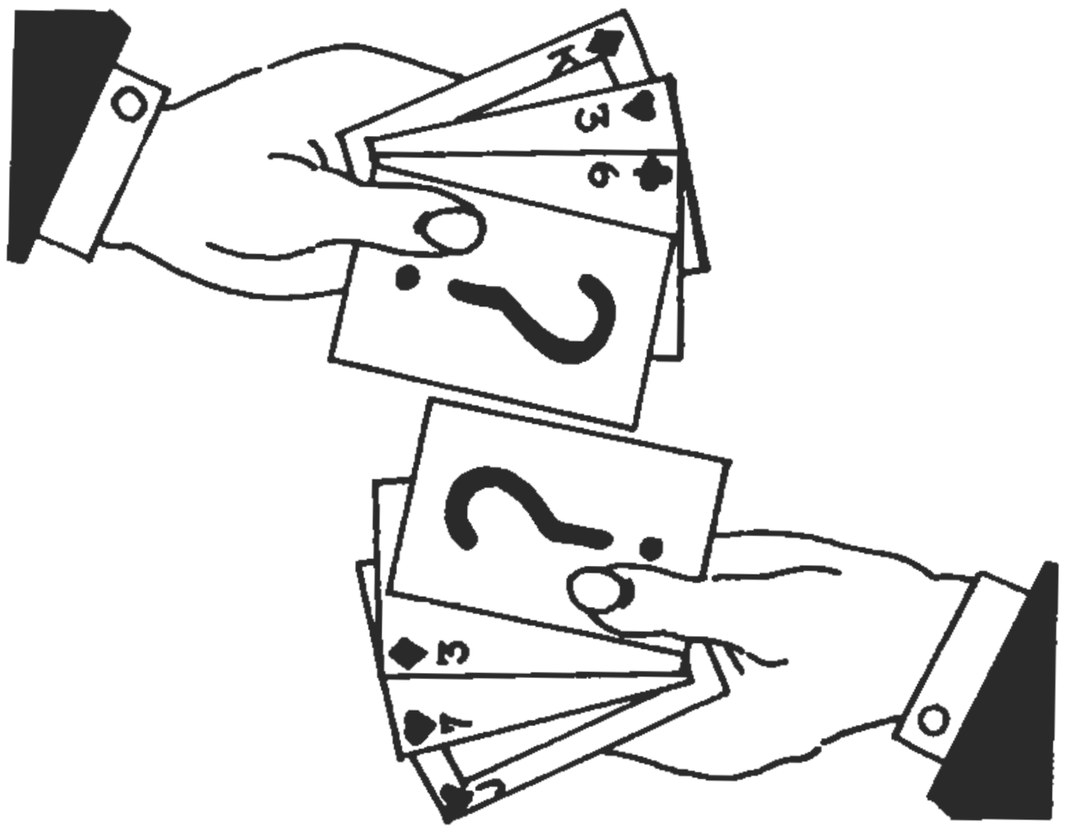
```
Tenho 21
Você tem 19
VINTE-E-UM!
```

```
 10 REM Vinte-e-um
 20 GO SUB 9000
 30 LET dinheiro=50
 40 GO SUB 7000
 50 LET c1=INT (RND*10) +1
 60 LET c2=INT (RND*10) +1
 70 PRINT AT 11,2; "┌──────┐"
 80 LET b$=CHR$ (144+ ( INT (RND*4)))
 90 PRINT AT 12,2; "│ "; b$; "   ▌  "
100 PRINT AT 13,2; "│      ▌"
110 PRINT AT 14,2; "▌      "; (CHR$ 8 AND c1=10);
c1; "  ▌"
120 PRINT AT 15,2; "▌      ▌"
130 PRINT AT 16,2; "▌      "; b$; "▌"
140 PRINT AT 17,2; "└──────┘"
150 LET b$=CHR$ (144+(INT (RND*4)))
160 PRINT AT 11,8; "┌──────┐"
170 PRINT AT 12,8; "│ ";b$; "   ▌"
180 PRINT AT 13,8; "▌      ▌"
190 PRINT AT 14,8; "▌      "; (CHR$ 6 AND c2=10);
c2; "  ▌"
200 PRINT AT 15,8; "▌      ▌"
210 PRINT AT 16,8; "▌      "; b$; "▌"
220 PRINT AT 17,8; "└──────┘"
225 LET tot=c1+c2: LET c3=INT (RND*10) +1
230 PRINT AT 5,5; "T – voltar"
235 PRINT AT 7,5; "S – Ficar-se"
240 IF INKEY$="t" OR INKEY$="s" THEN GO TO
250
245 GO TO 240
250 IF INKEY$="s" THEN GO TO 1000
260 LET tot=tot+c3
270 LET b$=CHR$ (144+ (INT (RND*4)))
275 IF p >26 THEN LET p=2: CLS
280 PRINT AT 11,p; "┌──────┐"
290 PRINT AT 12,p; "│ "; b$; "   ▌"
300 PRINT AT 13,p; "▌      ▌"
310 PRINT AT 14,p; "▌      "; (CHR$ 6 AND c3=10);
c3; "  ▌"
320 PRINT AT 15,p; "▌      ▌"
```



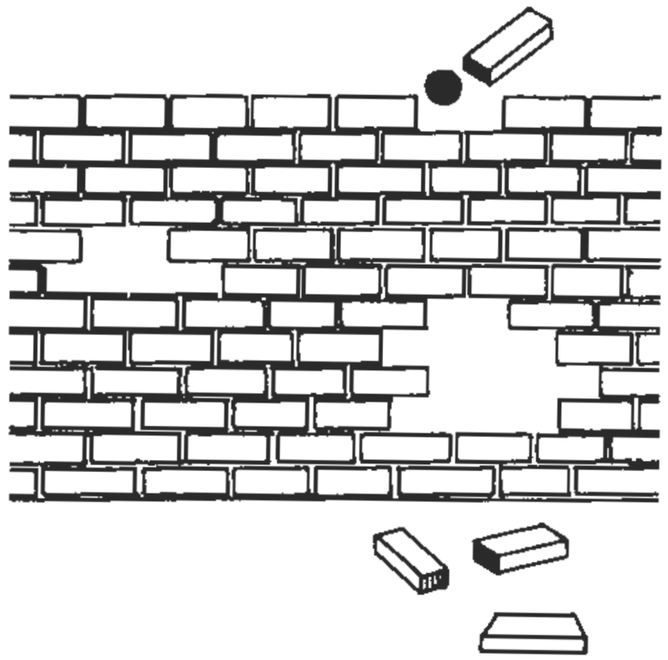
```
330 PRINT AT 16,p; "▌      "; b$; "▌"
340 PRINT AT 17,0; "└──────┘"
350 IF tot>21 THEN PRINT AT 1,1; "Eliminado": GO
TO 2000
360 LET p=p +6
365 LET c3=INT (RND*10) +1
370 GO TO 240
990 STOP
1000 LET cs=INT (RND*13) +10
1010 CLS
1020 PRINT' "Eu tenho"; cs'' "Você tem "; tot''
1030 IF cs >21 THEN PRINT "Fiquei eliminado, você
ganha": GO TO 1070
1040 IF cs=21 THEN PRINT "VINTE-E-UM": GO TO
1070
1050 IF cs>=tot THEN PRINT "Eu ganho": GO TO
1070
1060 PRINT "Tu ganhas 50 £!": LET dinheiro=dinhei
ro +50
1070 FOR k=1 TO 60: BEEP .008,1: BEEP .008,-k:
NEXT k: GO TO 40
2000 PRINT' "Eu ganho este jogo!"
2010 GO TO 1070
3000 CLS
3010 PRINT AT 10,2; "Você não tem mais dinheiro"
3020 INPUT "Carregue em enter para jogar nova-
mente"; LINE a$: RUN
7000 BORDER 0: PAPER 4: INK 9: CLS
7005 LET p=14: LET dinheiro=dinheiro -10: IF
dinheiro<0 THEN GO TO 3000
7010 PRINT AT 1,24; INK 1; "┌──────┐"
7020 FOR a=1 TO 6
7030 PRINT TAB 24; INK 1; "▌▒▒▒▒▌"
7040 NEXT a
7050 PRINT TAB 24; INK 1; "└──────┘"; AT 9,23;
"Cash"; dinheiro
7060 PRINT AT 3,5; "D — Dar"
7070 IF INKEY$< > "d" THEN GO TO 7070
7080 RETURN
7990 RETURN
```

```
9000 FOR a=USR "a" TO USR "k" +7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 24,60,126,255,255,126,60,24
9040 DATA 24,60,90,255,255,90,24,60
9050 DATA 16,56,124,254,254,64,16,56
9060 DATA 68,238,254,254,254,124,56,16
9070 DATA 204,204,51,51,204,204,51,51,
9080 DATA 0,0,0,7,15,12,24,24
9090 DATA 0,0,0,192,240,48,24,24
9100 DATA 24,24,12,15,7,0,0,0
9110 DATA 24,24,48,240,192,0,0,0
9120 DATA 0,0,0,255,255,0,0,0
9130 DATA 24,24,24,24,24,24,24,24,
9990 REM a  b  c  d  e  f  g  h  i  j  k
9991 REM ♦  ♠  ♣  ♥  ✲  ◢  ◣  ◤  ◥  ▬  ▮
```



A EVASÃO

É uma versão de um jogo muito conhecido. O objectivo do jogo é marcar pontos derrubando tijolos de uma parede com uma bola que é atirada com a sua raqueta. Tem dez bolas para jogar. O jogo acaba quando a décima bola fica fora de jogo. A tecla 5 movimenta a raqueta para a esquerda e a tecla 8 para a direita. Pode escolher-se também a velocidade, input 1 para rápido e qualquer número acima de 10 para lento.



```
  10 REM A evasão
  20 GO SUB 9000
  30 GO SUB 8000
  35 GO SUB 7000
  40 PRINT AT 20,v; "    ▬▬▬    "
  50 LET v=v +2*(INKEY$="8" AND v <27)
-2*(INKEY$="5" AND v>0): IF INKEY$="5" AND
v<0 THEN PRINT AT 20,1; "▬▬▬    "
  55 PRINT AT 0,0; "PONTUAÇÃO "; sc: FOR a=1 TO
s: NEXT a
  60 PRINT AT e,f; "  "
  70 LET e=e+c: IF e<2 THEN LET c=-c: BEEP
.008,20
  80 LET f=f+d: IF f<1 OR f>30 THEN LET d=-d:
BEEP .008,10
  90 IF SCREEN$ (e,f)="▁" THEN LET sc=sc+1: LET
c=-c: BEEP .008,15
  95 IF SCREEN$ (e,f)="▁" THEN LET coun
t=count+1: IF count=(32*3) THEN LET count=0: LET
s=s - (s + 1): GO TO 35
 100 IF SCREEN$ (e,f) =" " THEN LET c=-c: BEEP
.008,15
 110 PRINT AT e,f; INK 6; "●": IF e >20 THEN LET
b=b+1: GO SUB 1000
 115 IF b=10 THEN GO TO 2000
 130 GO TO 40
 990 STOP
1000 PRINT AT e,f; "  "
1010 LET e=20: LET f=INT (RND*16) +10: LET
c=-1
1020 PRINT AT 0,25; "BOLA "; b
1030 RETURN
2000 PRINT AT 10,12; "FIM DO JOGO"
2010 PRINT" "            Você marcou "; sc
2020 INPUT "Carregue em Enter para jogar      nova
mente"; LINE a$: GO TO 30
7000 PRINT AT 0,0; "PONTUAÇÃO 0"; AT 0,25; "BOLA
1": FOR a=1 TO 3
7010 PRINT INK a+2; INVERSE 1;
"                                                    "
```



```
7020 NEXT a
7030 RETURN
8000 BORDER 0: PAPER 1: INK 7: CLS
8010 LET v=15
8020 LET c=-1: LET d=1
8030 LET sc=0
8040 LET e=20: LET f=INT (RND*15)+10
8050 LET b=1
8060 INPUT "Velocidade (1 — Rápido 10 — Muito
lento)"; s
8070 LET count=0
8990 RETURN
9000 FOR a=USR "a" TO USR "b"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255,126,60
9040 DATA 0,0,0,0,0,0,255,255
9990 REM a  b
9991 REM ● ▬
```

EXCALIBUR CONTRA O INIMIGO

Este jogo é jogado numa grelha de sete por sete. Você comanda a nave Excalibur e tem de defender-se contra um certo número de naves espaciais inimigas. Se as naves espaciais inimigas o encurralarem de tal maneira que você não possa movimentar-se, ganham elas. Para você ganhar tem de capturar pelo menos 17 naves inimigas saltando por cima delas — antes de 50 jogadas. Pode andar para cima e para baixo, para a direita e para a esquerda, mas não em diagonal. Passa para a posição a seguir na direcção escolhida.

Para fazer a sua jogada tem de introduzir o número do quadrado para o qual pretende ir. Imprima o número adequado da coluna lateral, seguido pelo número transversal, mas tudo pegado — ou seja, sem deixar espaço entre os números e sem vírgula. O computador acusa imediatamente qualquer jogada ilícita. Pode ir para qualquer sector que contenha estrelas mas não para um lugar já ocupado por uma nave inimiga ou um quadrado.

```
10 REM Excalibur contra o inimigo
20 RANDOMIZE: GO SUB 9000
30 GO SUB 600
40 GO SUB 80
50 GO SUB 380
60 GO SUB 80
70 GO TO 50
80 REM — Jogadas do computador —
90 LET y=0; LET p=p+1
100 LET k=INT (RND*63) +13
110 LET y=y+1
120 IF h(k) < > 144 THEN GO TO 100
130 LET j=1
140 IF h(k+z (j))=146 THEN GO TO 190
150 LET j=j+1
160 IF j<3 THEN GO TO 140
170 IF y<100 THEN GO TO 100
180 PRINT "Excalibur ganha!!    ":STOP
190 LET h(k)=146: LET h(k+z(j))=144
200 RETURN
210 REM — Aceitar a jogada do jogador —
220 PRINT
230 REM o jogador carrega em 99 para q se não
jogar
240 LET q=qm
250 REM Input ao alto do écran —
260 INPUT AT 4,0; AT 0,0; "Excalibur move-se
para ";l
270 IF l=99 THEN GO TO 750
280 REM — Cercado —
290 LET r=ABS (l-g)
300 IF h(q) < >69 OR h(l) < >146 THEN PRINT
"Jogada ilícita": GO TO 230
310 IF r=9 OR r=18 OR r=18 OR r=22 THEN
PRINT "Os movimentos em diagonal são    ilícitos.":
GO TO 230
330 LET h(q) =146: LET h(l) =69
340 IF r=20 OR r=2 THEN LET h ((l+q)/2) =146:
LET s=s+1
350 LET qm=l
360 RETURN
370 REM — Imprimir o tabuleiro —
380 PRINT
```

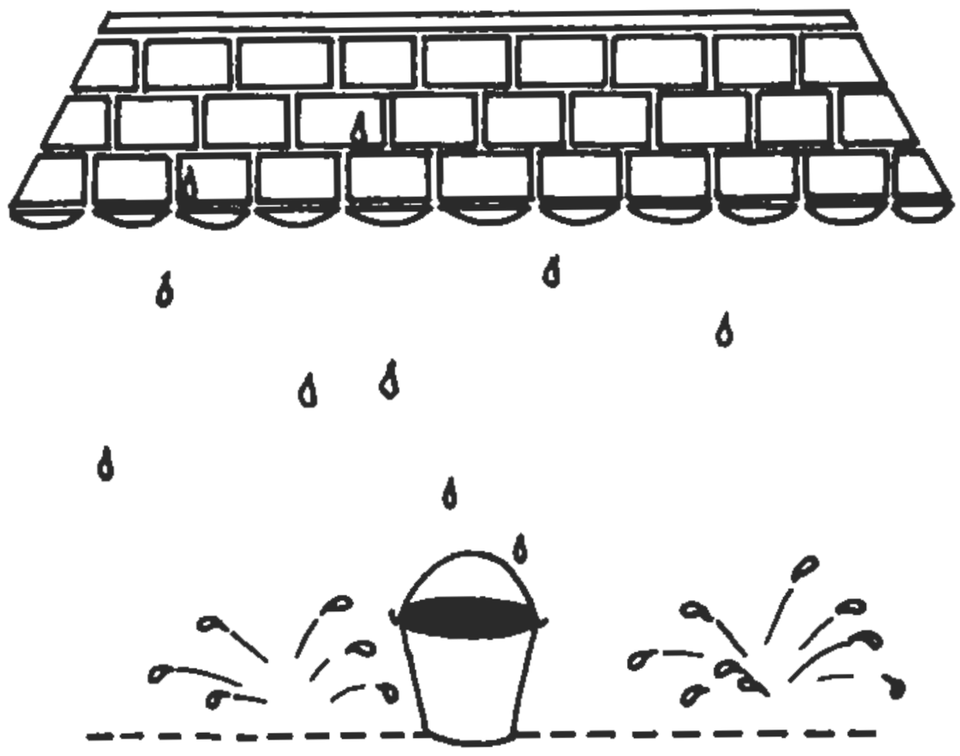
```
 390 PRINT: PRINT TAB 6; "jogada número "; p
 400 IF k=0 OR k+z (j) =10 OR k=k+z (j) THEN GO
TO 420
 410 PRINT
 420 PRINT 50-p; "Movimento esquerda ": PRINT
 430 PRINT TAB 4; "1234567"
 440 FOR j=70 TO 10 STEP -10
 450 LET a=h (j+1): LET b=h (j+2): LET c=h (j+3):
LET d=h (j+4): LET e=h (j+5): LET f=h (j+6): LET
g=h (j+7)
 470 PRINT TAB 3; j/10;
 480 PRINT CHR$ (a); CHR$ (b); CHR$ (c); CHR$
(d); CHR$ (e); CHR$ (f); CHR$ (g); j/10
 490 NEXT j
 500 PRINT TAB 4; "1234567"
 510 PRINT
 520 IF p=50 THEN GO TO 740
 530 PRINT "Contagem do inimigo:"; 17-s; "Naves que
faltam!"
 540 IF s=17 THEN GO TO 160
 550 GO SUB 210
 560 IF k=0 THEN GO TO 580
 570 RETURN
 580 FOR j=1 TO 750: NEXT j
 590 RETURN
 600 BORDER 0: PAPER 0: INK 6: BRIGHT 1: CLS
 610 DIM h (87): DIM z (5): LET k=0: LET s=0: LET
p=0
 620 RESTORE 630: FOR j=1 TO 3: READ q: LET
z(j)=q: NEXT j
 630 DATA 10,1, -1
 640 LET j=1: LET /=100
 650 FOR a=1 TO 87: LET h(a)=145
 660 IF a > 72 AND a < 76 OR a > 62 AND a <66 OR
a > 51 AND a <57 OR a >41 AND a< 47 THEN LET h
(a) =146
 680 IF a=51 OR a=41 OR a=57 OR a=47 THEN LET
h(a)=144
 690 IF a > 30 AND a<38 OR a>22 AND a<26 OR
a>12 AND a<16 THEN LET h(a) =144
```



```
 700 NEXT a
 710 LET h(44)=69
 720 LET qm=44
 730 RETURN
 740 PRINT "Acabou o tempo!"
 750 PRINT "A contagem do inimigo é "; s
 760 PRINT
 770 INPUT AT 20,0; AT 0,0; "outro jogo (s ou n)",
w$
 780 IF w$="s" THEN GO TO 30
 790 PRINT
 800 PRINT "Ainda bem que é capaz de reco   nhecer
que perdeu"
 810 PRINT
 820 PRINT, "Excalibur"
 830 STOP
9000 RESTORE 9000: FOR a=USR "a" TO USR "c"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 24,24,60,90,153,189,195,129
9040 DATA 255,129,129,129,129,129,129,255
9050 DATA 0,0,1,0,4,0,0,16
```

ENTRA ÁGUA PELO TELHADO

O telhado deixa entrar água e você só tem um balde. Tente apanhar as gotas de água no balde carregando na tecla 5 para andar para a esquerda e na 8 para andar para a direita. Os pingos são trinta ao todo, por isso tem de apanhar o maior número possível no balde, para não ser obrigado a chamar o pedreiro.

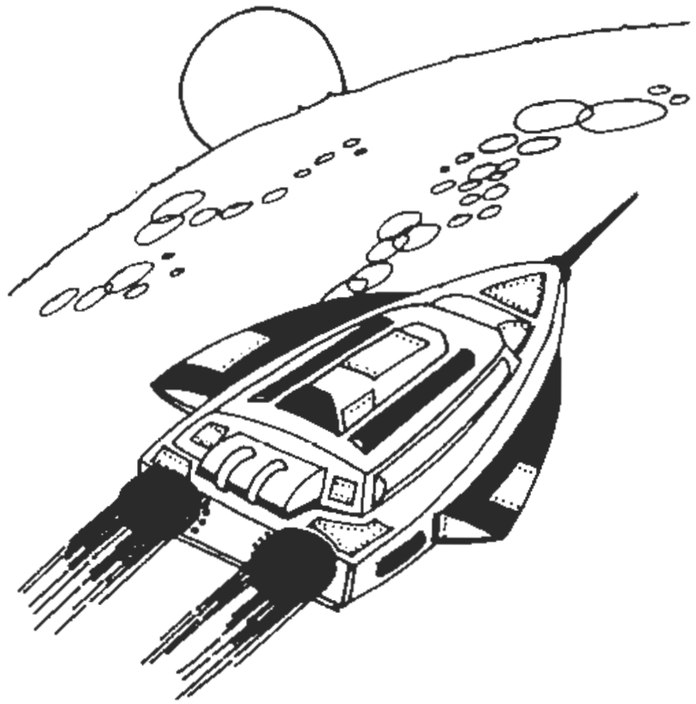



```
 10 REM Entra água pelo telhado
 20 GO SUB 9000: LET hi=0
 30 GO SUB 8000
 40 GO SUB 7000
 50 LET p=INT (RND*30) + 1: LET d=d+1: PRINT
AT 0,25; "Gota "; d
 60 FOR l=6 TO 18
 70 PRINT AT 20,v; INK 6; "  "; AT 21,v;
"  "
 80 PRINT AT l,p; INK 5; "  "; AT l+1,p; "  ";
AT l+2,p; "  "
 90 IF l+3=20 AND v+1=p+1 OR l+3=20 AND
v+2=p+1 OR l+3=20 AND v+2=p THEN LET
sc=sc+1: PRINT AT 0,0; "PONTUAÇÃO "; sc
 100 LET v=v+(INKEY$="8" AND v<28) - (IN
KEY$="5" AND v >0)
 110 NEXT l
 120 PRINT AT l,p; "  "; AT l+1,p; "  "
 130 IF d=30 THEN GO TO 150
 140 GO TO 50
 150 PRINT AT 5,11; "FIM DO JOGO"
 160 PRINT'''' "Você marcou "; sc
 170 IF sc>hi THEN LET hi=sc
 180 PRINT''' "Pontuação máxima de hoje "; hi
 190 INPUT "Carregue em ENTER para jogar      no
vamente"; LINE a$: GO TO 30
7000 CLS
7010 PRINT "PONTUAÇÃO "; sc
7020 PRINT INK 2; "
"
7025 PRINT INK 2; "
"
7030 PRINT INK 2; "
"
7035 PRINT INK 2; "
"
7990 RETURN
8000 BORDER 0: PAPER 0: INK 7: CLS: BRIGHT 0
8010 LET sc=0
8020 LET v=15
```

```
8030 LET d=0
8990 RETURN
9000 FOR a=USR "a" TO USR "j"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 255,255,255,255,255,255,0,0
9040 DATA 252,252,252,252,252,252,0,0
9050 DATA 0,0,0,0,0,1,3,7
9060 DATA 0,0,0,0,0,128,192,224
9070 DATA 7,7,15,15,15,7,7,3
9080 DATA 224,224,240,240,240,224,224,192
9090 DATA 0,0,127,63,95,111,119,69
9100 DATA 0,0,254,254,253,251,251,246
9110 DATA 62,63,63,63,31,31,31,31
9120 DATA 14,254,254,254,252,252,252,252
9130 REM a  b  c  d  e  f  g  h  i  j
9140 REM ■  ■  ▗  ▖  ▚  ▞  ▙  ▟  ▜  ▛
```


ATERRAGEM EM MARTE

Será capaz de pousar o seu módulo espacial no campo de aterragem? A entrada é estreita e é necessário ter um bom golpe de vista para aterrar em segurança. Depois da primeira aterragem recebe outro módulo que desce mais depressa do que o primeiro; por isso tenha cuidado — faça boa pontaria, pois de outra maneira pode ter um desastre e fica em Marte para sempre. Utilize a tecla 5 para andar para a esquerda e a tecla 8 para andar para a direita.



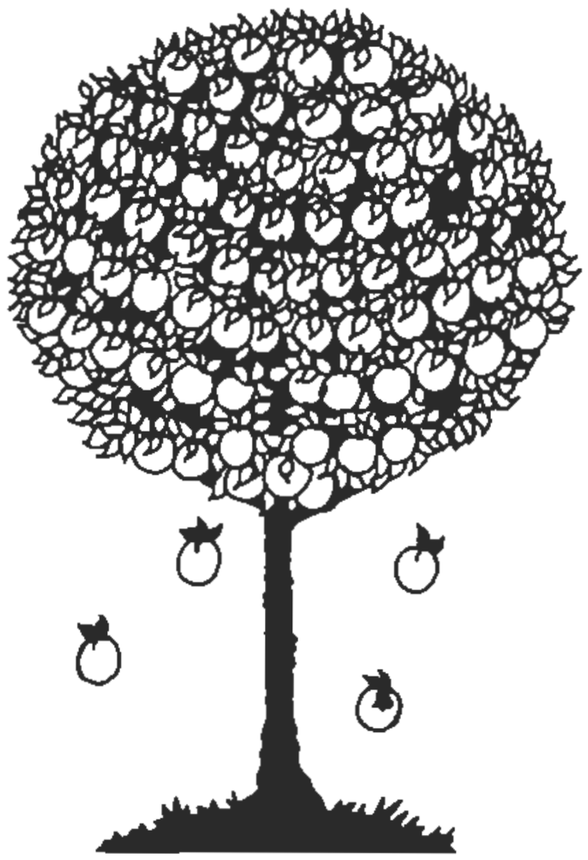
```
  10 REM Aterragem em Marte
  20 GO SUB 9000: LET hi=0
  30 GO SUB 8000
  40 GO SUB 7000
  50 PRINT AT v,h; "   "; AT v+1,h; "   "
  60 LET v=v+(((sc/5) +1)/5): LET h=h+(IN
KEY$="8" AND h<31) - (INKEY$="5" AND h>0)
  70 IF SCREEN$ (v+1,h) < >"  " THEN GO TO 1000
  80 PRINT AT v,h; INK 6; " ▟▙ "; AT v+1,h; " ▽▽ "
  90 IF v>20 THEN GO TO 2000
 100 PRINT AT 21,22; "PONTUAÇÃO "; sc
 110 LET t=t+.1
 120 PRINT AT 21,0; "TEMPO "; INT t
 150 GO TO 50
 990 STOP
1000 PRINT AT 2,5; "FIM DO JOGO EM "; t; " SE
GUNDOS"
1010 PRINT AT v,h; FLASH 1; INK 6; " ▟▙ "; AT
v+1,h; " ▽▽ "
1020 FOR p=1 TO 100: NEXT p
1030 IF sc>hi THEN LET hi=sc
1040 PRINT AT 7,9; "Você marcou "; sc
1050 PRINT AT 9,5; "Pontuação máxima de hoje "; hi
1060 INPUT "Carregue em ENTER para jogar      nova
mente"; LINE a$: GO TO 30
2000 LET v=0: LET h=0: LET sc=sc +1
2010 CLS
2020 GO TO 40
7000 LET b=INT (RND*16) +10: LET b=b*8
7010 FOR a=0 TO b-10: PLOT a,0: DRAW W, (INT
(RND*50) +50)
7020 NEXT a
7040 FOR a=b+10 TO 255: PLOT a,0: DRAW 0, INT
(RND*50) +50
7050 NEXT a
7990 RETURN
8000 BORDER 0: PAPER 0: INK 2: CLS
8010 LET h=0: LET v=0
8020 LET t=0: LET sc=0
8990 RETURN
```



```
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,0,7,15,31,53,53,53
9040 DATA 0,0,224,240,248,172,172,172
9050 DATA 31,13,16,16,8,4,14,0
9060 DATA 248,176,72,8,16,32,112,0
9070 REM a  b  c  d
9080 REM ▟  ▙  ▽  ▽
```

ROUBAR O POMAR

Apanhe o maior número possível de maçãs do pomar antes que o dono chegue e o jogo acabe. Marca três pontos por cada maçã que come e se conseguir comer as 50 maçãs no limite de tempo de 30 segundos ganha um bónus em pontos.

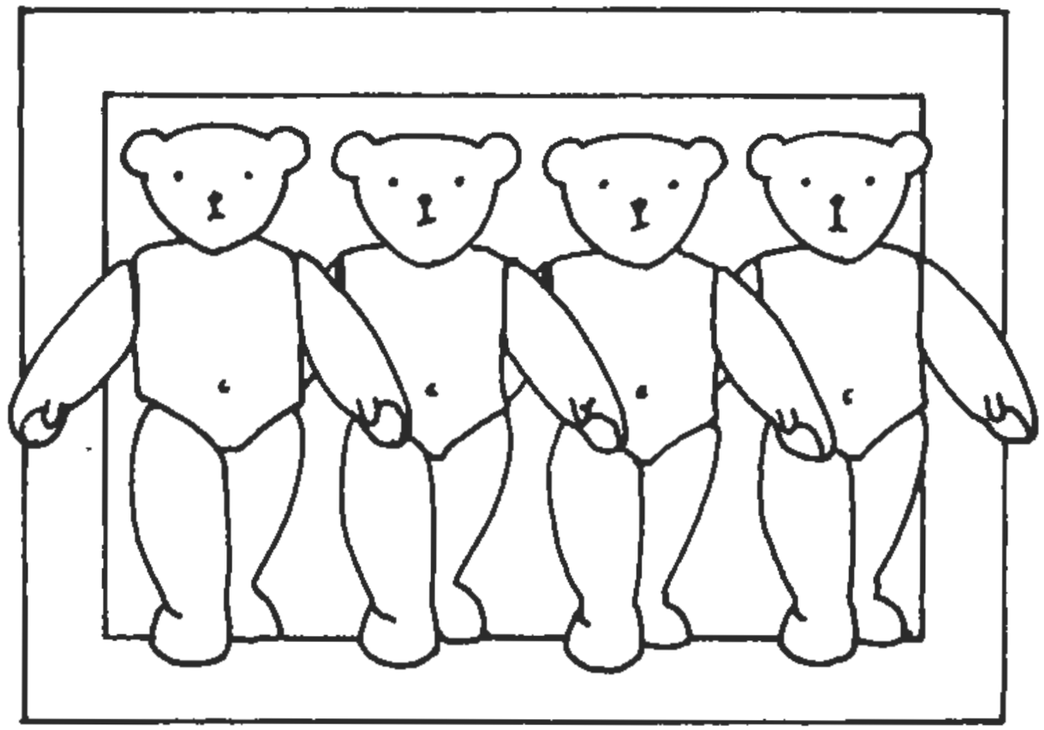



```
  10 REM Roubar o pomar
  20 GO SUB 9000: LET hi=0
  30 GO SUB 8000: GO SUB 7000
  40 PRINT AT v,h; "    "
  45 LET v1=v: LET h1=h
  50 LET v=v+(INKEY$="6" AND v<20)-(IN
KEY$="7" AND v>1)
  60 LET h=h+(INKEY$="8" AND h<31)-(IN
KEY$="5" AND v>0)
  70 IF SCREEN$ (v,h)=" " THEN LET v=v1: LET
h=h1: GO TO 45
  80 IF SCREEN$ (v,h) =" " THEN LET sc=sc+3: LET
co=co+1
  90 PRINT AT v,h; INK 6; "웃"
 100 LET ti=ti+.1: PRINT AT 0,0; PAPER 2; "PONTUA
ÇÃO"; sc, "TEMPO"; INT ti
 105 BEEP .008,co
 110 IF ti>30 THEN GO TO 1000
 120 IF co=50 THEN GO TO 2000
 130 GO TO 40
1000 PRINT AT v,h; FLASH 1; INK 2; "웃"
1010 PRINT AT 1,12; PAPER 1; "FIM DO JOGO"
1020 PRINT AT 5,7; PAPER 2; ''ACABOU O
TEMPO"
1030 PRINT AT 12,10; "VOCÊ MARCOU"; sc
1040 IF sc>hi THEN LET hi=sc
1050 PRINT AT 20,6; "Pontuação máxima de hoje"; hi
1060 INPUT "Carregue em ENTER para jogar nova
mente"; LINE a$: GO TO 30
2000 PRINT AT v,h; FLASH 1; INK 6; "웃"
2010 PRINT AT 2,10; PAPER 1; "FIM DO JOGO"
2020 PRINT AT 5,0; PAPER 2; "VOCÊ FEZ ISSO EM";
INT ti; «UNIDADES DE TEMPO»
2030 PRINT AT 7,10; PAPER 4; FLASH 1; "BÓNUS";
INT (100-ti)*3
2040 LET sc=sc+INT (100-ti)*3
2050 GO TO 1030
7040 PRINT AT 0,0; PAPER 2; "PONTUAÇÃO "; sc;
"TEMPO "; ti,
7050 PRINT INVERSE 1; INK 6; "
```

```
_____"
7060 FOR a=1 TO 18
7070 PRINT INVERSE 1; INK 6; "_"; INVERSE 0; TAB
31; INVERSE 1; "_"
7080 NEXT a
7090 PRINT INVERSE 1; INK 6; " _____
_____"
7095 FOR a=1 TO 50
7100 LET j=INT (RND*18) +2: LET k=INT
(RND*29) +2
7110 IF j=v AND k=h THEN GO TO 7100
7115 IF SCREEN$ (j,k)=" "THEN GO TO 7100
7120 PRINT AT j,k; INK 4; "●": NEXT a
7130 RETURN
8000 BORDER 0: PAPER 0: INK 9: CLS
8010 LET v=INT (RND*16) +2: LET h=INT
(RND*29) +2
8020 LET co=0: LET ti=0
8030 LET sc=0
8040 RETURN
9000 FOR a=USR "a" TO USR "b" +7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 12,24,62,127,127,127,127,62
9040 DATA 28,28,8,63,8,28,34,65
9050 REM a  b
9060 REM ●  ♙
```


SALVAR O URSINHO

O ursinho quer entrar no écran, mas infelizmente há muitas pistolas que disparam raios laser para lhe impedir o caminho. Se conseguir evitar os raios laser, o ursinho chega são e salvo. Ganha três pontos por cada ursinho que chega salvo a casa. O jogo começa com cinco ursinhos. Utilize as teclas do cursor para controlar os seus movimentos.

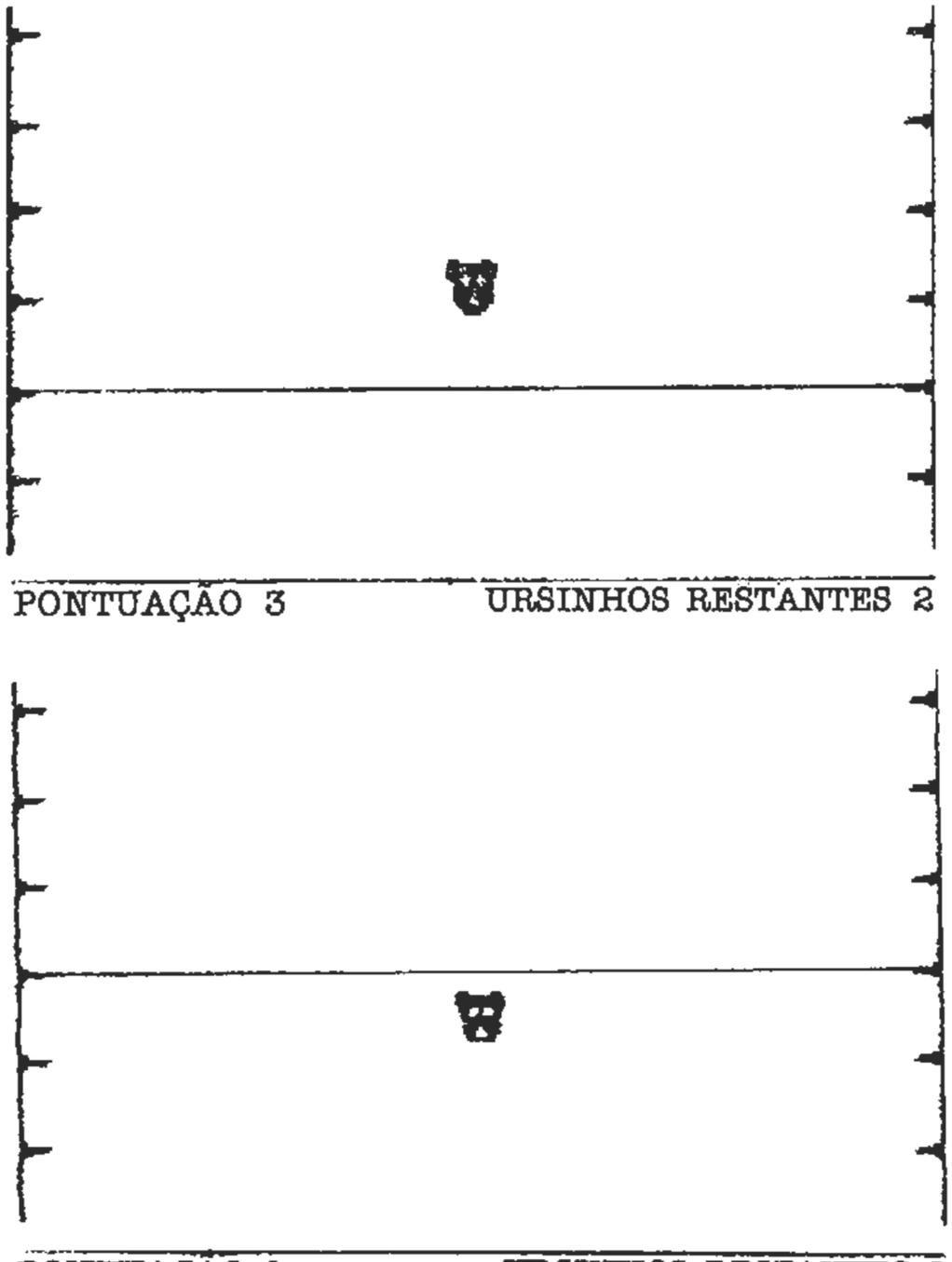


```
  10 REM Salvar o ursinho
  20 REM Peter Shaw
  30 REM Ideia original de Steven Gunning
  40 GO SUB 9000: LET hi=0
  50 GO SUB 8000: GO SUB 7000
  60 PRINT AT v,h; OVER 1; PAPER 8; INK 6; "  ";
AT v+1,h; "  "
  70 LET v=v+2*(INKEY="6" AND v <19)
-2*(INKEY$="7" AND v>0)
  80 LET h=h+2*(INKEY$="8" AND h <29)
-2*(INKEY$="5" AND h>1)
  90 PRINT AT v,h; OVER 1; PAPER 8; INK 6; "  ";
AT v+1,h; "  "
 100 GO SUB 1000
 110 IF v=0 THEN GO SUB 2000
 115 PRINT AT 21,0; PAPER 2; "PONTOS "; sc; " UR-
SI NHOS RESTANTES "; td,
 120 BEEP .008, sc
 130 GO TO 60
 990 STOP
1000 LET r=INT (RND*6) +1
1010 PLOT 8,m(r): DRAW INK 8; 240,0
1020 LET p=(21 - ((m(r) -4)/8))
1030 IF v=p OR v=p-1 THEN GO TO 3000
1035 PLOT 8,m(r): DRAW INK 0; 240,0
1040 PLOT 8,m(r): DRAW OVER 1; 240,0
1050 RETURN
2000 PRINT AT v,h; PAPER 8; "   "; AT v+1,h;"   "
2005 BEEP .1,20
2010 LET sc=sc+3
2020 LET v=18: LET h=15
2030 PRINT AT v,h; INK 6; PAPER 8; OVER 1; "  ";
AT v+1,h; "  "
2040 RETURN
3000 PRINT AT v,h; INK 2; FLASH 1; "  "; AT
v+1,h; "  "
3001 BEEP .5,-10
3005 LET td=td-1: IF td=-1 THEN GO TO 4000
3010 PRINT AT v,h; "    "; AT v+1,h; "    "
3020 LET v=18: LET h=15
```



```
3030 PRINT AT v,h; INK 6; PAPER 8; OVER 1; "  ";
AT v+1,h; "  "
3040 GO TO 1035
4000 PRINT AT 2,12; PAPER 6; "FIM DO JOGO"
4010 PRINT'' TAB 10; PAPER 1; "VOCÊ MARCOU "; sc
4020 IF sc>hi THEN LET hi=sc
4030 PRINT AT 18,3; PAPER 6; "Pontuação máxima
de hoje "; hi
4040 INPUT "Carregue em "; PAPER 2; "ENTER"; PA
PER 0; " para jogar novamente"; LINE a$
4050 GO TO 50
4990 STOP
7000 CLS
7010 PRINT PAPER 1,,
7020 PLOT 0,14: DRAW 0,152
7030 PLOT 255,14: DRAW 0,152
7040 FOR a=2 TO 17 STEP 3
7050 PRINT AT a,0; "  " AT a,31; "  "
7060 NEXT a
7070 PRINT AT 20,0; PAPER 1,,
7080 PLOT 0,8: DRAW 255,0
7090 PRINT AT 21,0; "PONTOS "; sc, "URSINHOS RES
TANTES "; td
7100 PRINT AT v,h; OVER 1; PAPER 8; INK 6; "  ";
AT v+1,h; "  "
7990 RETURN
8000 BORDER 0: PAPER 0: INK 9: CLS
8010 LET sc=0: DIM m(6): LET v=18: LET h=15
8020 RESTORE 9090: FOR a=1 TO 6
8030 READ m(a): NEXT a
8040 LET co=0
8050 LET td=6
8990 RETURN
9000 FOR a=USR "a" TO USR "f" +7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 0,56,127,127,127,125,56,29
9040 DATA 0,28,254,254,254,222,140,220
9050 DATA 31,31,31,31,30,15,7,3
9060 DATA 252,252,124,124,60,248,240,224
```

```
9070 DATA 128,128,224,255,255,224,128,128
9080 DATA 1,1,7,255,255,7,1,1
9090 DATA 36,60,84,108,132,156
9998 REM
9999 REM
```

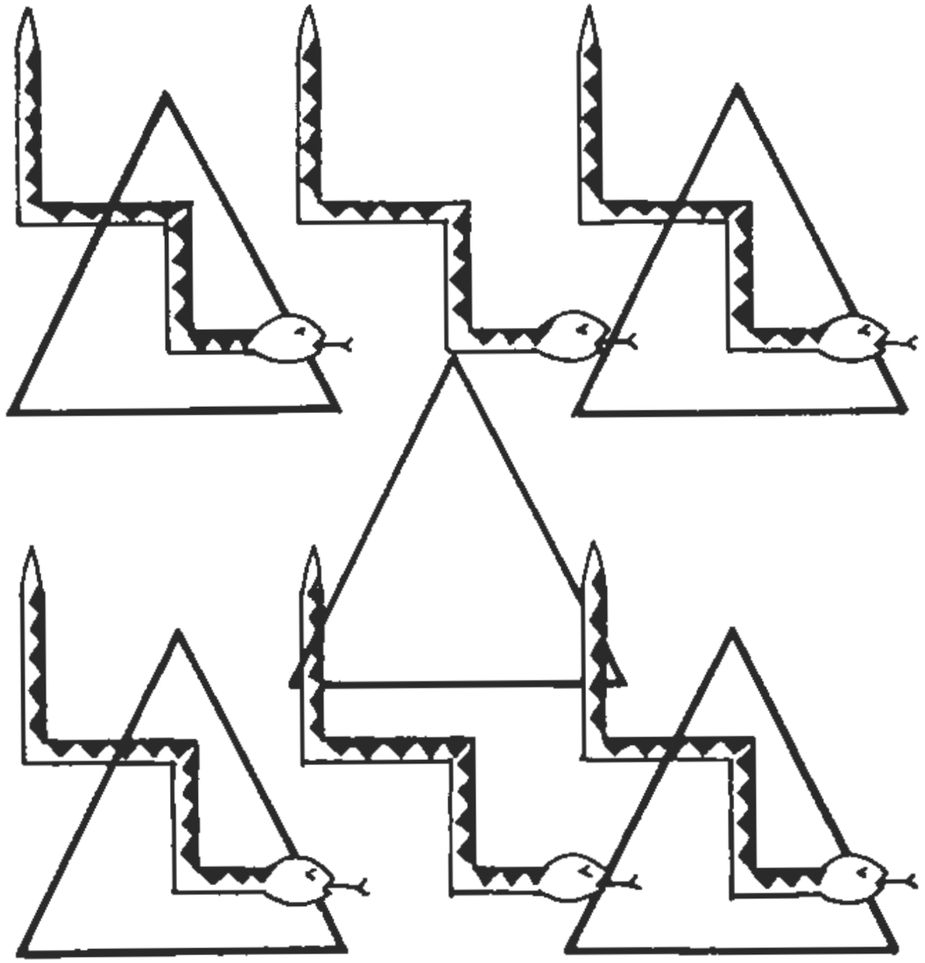


A SERPENTE ENTRE TRIÂNGULOS

Você é uma serpente e serpenteia em movimentos angulosos, descendo o écran. Um certo número de triângulos azuis venenosos vêm a toda a velocidade na sua direcção; tem de os evitar a todo o custo, pois se bater num deles o jogo acaba. Anda automaticamente para a esquerda, por isso utilize a tecla 8 para controlar os seus movimentos para a direita.

```
  10 REM A serpente entre triângulos
  20 GO SUB 9000: LET hi=0
  30 GO SUB 8000
  40 PRINT AT 9,h; " "
  50 LET h=h+2*(INKEY$="8" AND v <29): IF h>0
THEN LET h=h -1
  54 IF SCREEN$ (10,h)=" " THEN GO TO 1000
  55 PRINT AT 9,h; INK 6; "";  AT 10,h; ""
  60 PRINT AT 21,0;
  70 PRINT TAB INT (RND*31); INK 1; ""
  80 POKE 23692,255
  90 LET sc=sc+1: PRINT'
 100 GO TO 40
1000 PRINT AT 0,0; OVER 1; PAPER 8; INK 2; v$(1)
1010 PRINT AT 1,10; FLASH 1; "FIM DO JOGO"
1020 PRINT AT 5,7; "Você marcou "; sc
```

```
1030 IF sc>hi THEN LET hi=sc
1040 PRINT AT 10,2; "Pontuação máxima de hoje"; hi
1050 INPUT "Carregue em "; PAPER 1; "ENTER "; PA
PER 0; "para jogar      novamente"; LINE a$: GO TO 30.
1090 STOP
8000 BORDER 0: PAPER 0: INK 9: CLS: LET v=10:
LET h=15
8010 LET sc=0: RANDOMIZE
8020 DIM v$(1,704)
8990 RETURN
9000 FOR a=USR "a" TO USR "c" +7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 255,126,126,60,60,24,24,0
9040 DATA 0,99,119,127,62,28,0,0
9050 DATA 99,119,73,73,127,62,28,0
```



A CORRIDA DA MORTE

O objectivo deste jogo é atropelar o maior número possível de peões — como no filme do mesmo nome — e marca pontos por cada peão atropelado. Quando atinge o seu alvo aparece um túmulo no écran. Tem um limite de tempo de 60 segundos para jogar, por isso não perca tempo. Utilize as teclas do cursor para controlar o seu carro.

```
  10 REM A Corrida da Morte
  20 GO SUB 9000: LET hi=0
  30 GO SUB 8000
  40 GO SUB 7000
  50 PRINT AT v,h; "  "
  51 IF INKEY$<"5" OR INKEY$>"8" THEN GO TO
55
  53 LET a$=INKEY$
  55 LET v1=v: LET h1=h
  60 LET v=v+(a$="6") - (a$="7")
  70 LET h=h+(a$="8") - (a$="5")
  80 IF SCREEN$ (v,h)="  " THEN GO TO 1000
  90 IF SCREEN$ (v,h)="_" THEN LET v=v1: LET
h=h1: LET a$="  ": GO TO 55
 100 PRINT AT v,h; INK 3; c$
 105 LET d$=c$
 110 LET c$=("  " AND a$="7") + ("  " AND
a$="8") + ("  " AND a$="6") + ("  " AND
a$="5")
 120 IF c$="  " THEN LET c$=d$
 125 PRINT AT 21,16; "TEMPO "; INT ti; "  ": LET ti
=ti-.2: IF ti<0 THEN GO TO 2000
 130 BEEP .008,sc
 140 PRINT AT j,k; "  "
 150 LET j1=j: LET k1=k
 160 LET j=j+INT (RND*3) -1: LET k=k+INT
(RND*3) -1
 170 IF SCREEN$ (j,k)="  " THEN LET j=j1: LET
k=k1: GO TO 150
 175 IF SCREEN$ (j,k)="  " THEN GO TO 1000
 180 PRINT AT j,k; INK 5; "  "
 200 GO TO 50
1000 PRINT AT j,k; "  "
1010 FOR a=1 TO 3
1020 LET sc=sc+1
1030 PRINT AT 21,0; "PONTUAÇÃO "; sc
1040 BEEP .7,sc
1050 NEXT a
1060 LET j=4: LET k=20: LET v=10: LET k=16: GO
TO 40
```



```
2000 PRINT AT 21,16; PAPER 2; "TEMPO ": BEEP 2,
-10: CLS
2010 PRINT AT 2,12; PAPER 1; "FIM DO JOGO"
2020 PRINT AT 5,10; PAPER 2; "VOCÊ MARCOU "; sc
2030 IF sc>hi THEN LET hi=sc
2040 PRINT AT 18,6; PAPER 6; "PONTUAÇÃO MÁXI
MA DE HOJE "; hi
2050 INPUT PAPER 1; "CARREGUE EM"; PAPER 2;
"ENTER"; PAPER 1; "PARA JOGAR NOVAMENTE"; LI
NE a$: GO TO 30.
7000 CLS: PRINT INVERSE 1; INK 6; "  _____
_____"
7010 FOR a=1 TO 19
7020 PRINT INVERSE 1; INK 6; "_"; AT a,31; "_"
7030 NEXT a
7040 PRINT INVERSE 1; INK 6; "  _____
_____"
7050 PRINT AT 21,0; "PONTUAÇÃO"; sc
7060 RETURN
8000 BORDER 0: PAPER 0: INK 9: CLS
8010 LET v=10: LET h=16
8020 LET j=4: LET k=20
8030 LET sc=0: LET ti=60
8040 LET c$="  "
8050 LET a$="  "
8990 RETURN
9000 FOR a=USR "a" TO USR "f"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 60,153,255,153,24,169,255,189
9040 DATA 238,68,229,255,255,229,68,238
9050 DATA 189,255,189,24,153,255,153,60
9060 DATA 119,34,167,255,255,167,34,119
9070 DATA 56,124,238,198,238,238,254,254
9080 DATA 56,56,16,124,16,16,40,68
9998 REM a  b  c  d  e  f
9999 REM    
```

JOGO DAS DAMAS

O jogo das damas é jogado segundo regras convencionadas. Cada jogador começa com 12 pedras e joga-se num tabuleiro quadriculado com oito quadrados de lado. Os movimentos são feitos em diagonal nos quadrados pretos. Só pode deslocar as suas pedras subindo no tabuleiro e o computador descendo — a não ser que chegue ao lado oposto do tabuleiro e faça «dama», nessa altura a dama pode andar nas duas direcções. «Comem-se» todas as pedras por cima das quais se salta.

JOGA COM AS PEDRAS BRANCAS E O COMPUTADOR COM AS PRETAS

Para fazer uma jogada tem de introduzir no computador a letra e o número da casa onde está a pedra que vai deslocar (por exemplo, C6); depois carregue na tecla ENTER e introduza o número e a letra da casa para onde quer deslocar a pedra (por exemplo, E4). Quando se come uma das pedras do adversário esta desaparece automaticamente do écran e o computador pergunta (aparece um ponto de interrogação) se você quer comer outra pedra. Carregue em 'Y' e carregue em ENTER para comer a outra pedra.

PARA JOGAR ESTE JOGO AS LETRAS INTRODUZIDAS TÊM QUE SER TODAS MAIÚSCULAS; SERÁ PREFERÍVEL, PORTANTO, PÔR O COMPUTADOR EM CAPS-LOCK ANTES DE CORRER O PROGRAMA.

```
  10 REM Jogo das Damas
  20 GO SUB 9000
  30 GO SUB 1620
  40 GO TO 1340
  50 CLS
  60 PRINT PAPRER 1; TAB 12; "DAMAS",
  70 PRINT''
  80 PRINT TAB 2; PAPER 6; "ABCDEFGH"
  90 GO SUB 460
 100 PRINT "A sua pontuação "; sm, "Pontuação da
máquina "; si; "   "
 110 PRINT
 120 IF u$="N" THEN GO TO 680
 130 IF si=12 THEN PRINT "Ganho eu": STOP
 140 IF sm=12 THEN PRINT "Ganha você": STOP
 150 REM
 160 IF q=2 THEN GO TO 410
 170 PRINT "Última jogada para "; f$;
 180 INPUT "De (Let., No.)"; c$; " para "; b$
 220 LET f$=b$
 230 LET d$=c$
 240 GO SUB 1550
 250 LET c=m(i)
 260 IF i=0 THEN GO TO 180
 270 LET d$=b$
 280 GO SUB 1560
 290 LET b=m (i)
 300 IF i=0 THEN GO TO 180
 310 IF ABS (c-b)=10 OR ABS (c-b)=8 THEN LET
sm=sm+1
 320 LET u$="   "
 330 IF b-c=10 THEN LET a(b-5)=0
 340 IF b-c=8 THEN LET a(b-4)=0
```

```
 350 IF c-b=10 THEN LET a(c-5)=0
 360 IF c-b=8 THEN LET a(c-4)=0
 370 LET a(b)=a(c)
 380 LET a(c)=0
 390 LET q=2
 400 GO TO 50
 410 LET u$=" ": LET q=0
 420 IF ABS (c-b)=10 OR ABS (c-b)=8 THEN
PRINT: PRINT: PRINT: PRINT: INPUT u$
 440 IF u$< >"Y" THEN GO TO 680
 450 GO TO 50
 470 LET m1=0: LET k=1
 480 LET j=-1
 490 FOR i=40 TO 6 STEP -1
 500 IF a(i)=1 AND i>37 THEN LET a(i)=2
 510 IF a(i)=-1 AND i<10 THEN LET a(i)=-2
 520 IF i=14 OR i=32 OR i=23 THEN GO TO 580
 530 IF m1=0 THEN PRINT k; " ";: LET m1=0: LET
k=k+1: LET j=-1*j: IF j=1 THEN PRINT INK 2; "■";
 540 LET a=a(1)
 550 GO SUB 610
 560 IF m1< >3 OR j=-1 THEN PRINT INK 2; " ■ ";
 570 LET m1=m1+1: IF m1>3 THEN LET m1=0:
PRINT
 580 NEXT i
 590 PRINT
 600 RETURN
 610 REM Imprimir as pedras
 620 IF a=0 THEN PRINT "  ";
 630 IF a=1 THEN PRINT " ●";
 640 IF a=-1 THEN PRINT "O";
 650 IF a=-2 THEN PRINT INVERSE 1; "O";
 660 IF a=2 THEN PRINT INVERSE 1; "●";
 670 RETURN
 680 LET u$="    ": LET q=0
 690 LET z=6
 700 IF z<9 THEN GO TO 740
 710 IF a(z) <0 AND (a (z-4)=1 OR a(z-4)=2) AND
a (z-8)=0 THEN GO TO 930
 720 IF z<1 THEN GO TO 740
```



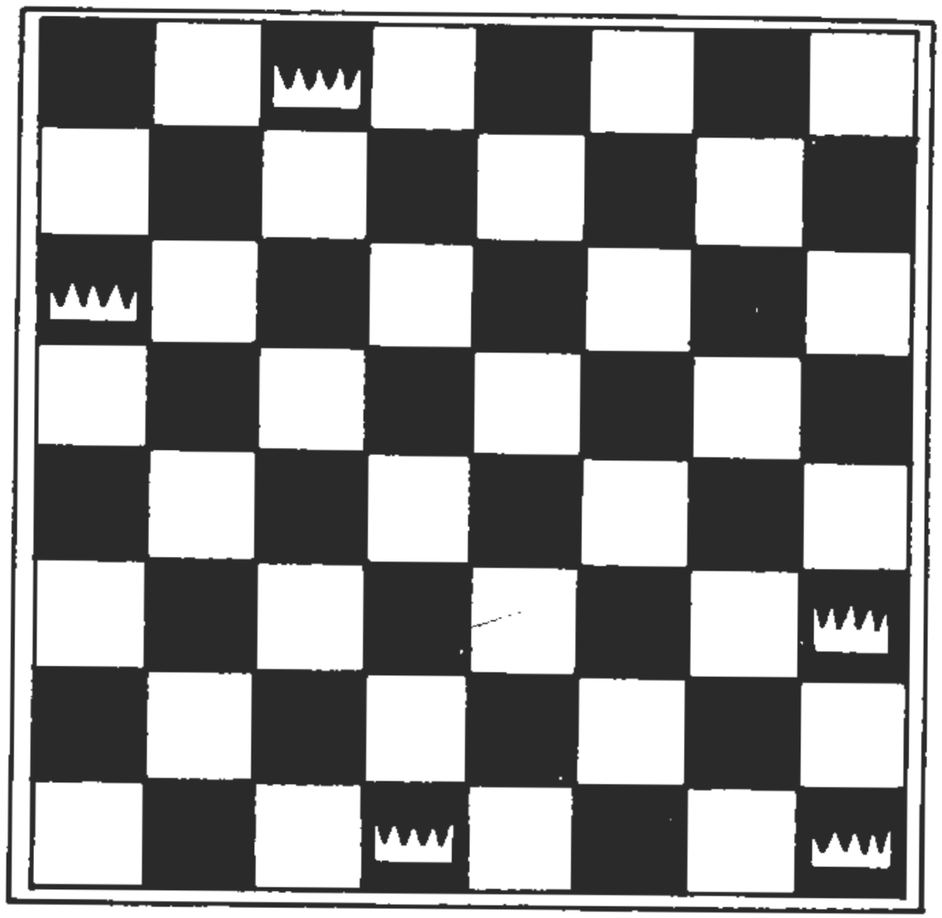
```
 730 IF a(z) <0 AND (a (z-5)=1 OR a (z-5)=2)
AND a (z-10)=0 THEN GO TO 1030
 740 IF z>25 THEN GO TO 770
 750 IF a(z)=-2 AND (a (z+4)=1 OR a (z+4)=2)
AND a(z+8)=0 THEN GO TO 1140
 760 IF a(z)=-2 AND (a (z+5)=1 OR a(Z+5)=2)
AND a(z+10)=0 THEN GO TO 1250
 770 LET z=z+1: IF z<=40 THEN GO TO 700
 780 REM * RANDOM *
 790 LET u=0
 800 LET z=6+INT (RND*34)+1
 810 LET k=0
 820 LET u=u+1
 830 IF a(z) <0 AND a(z-4)=0 THEN LET k=1
 840 IF a(z) <0 AND a(z-5)=0 AND k=0 THEN LET
k=2
 850 IF k=0 AND z<26 AND a(z)=-2 AND
a(z+4)=0 THEN LET k=-7
 860 IF z<10 THEN GO TO 880
 870 IF (k=1 OR k=2) AND u<200 AND (a(z-(10
AND z>10))=1 OR a(z-8)=1) THEN GO TO 800
 880 IF k=0 AND u<400 THEN GO TO 800
 890 IF k=0 THEN LET sm=12: GO TO 50
 900 LET a(z-(3+k))=a(z)
 910 LET a(z)=0
 920 GO TO 50
 930 LET a(z-8)=a(z)
 940 LET a(z)=0
 950 LET a(z-4)=0
 960 LET si=si+1
 970 IF z<24 THEN GO TO 50
 980 IF (a(z-13)=1 OR a(z-13)=2) AND a(z-18)
=0 THEN LET p=2
 990 IF p=1 THEN LET a(z-18)=a(z-8): LET
a(z-13)=00
1000 IF p=2 THEN LET a(z-12)=0: LET a(z-12)=0
1010 IF p>0 THEN LET a(z-8)=0
1020 GO TO 50
1030 LET a(z-10)=a(z)
1040 LET a(z)=0
```

```
1050 LET a(z-5)=0
1060 LET si=si+1
1070 IF z<25 THEN GO TO 50
1080 IF (a(z-15)=1 OR a(z-15)=2) AND
a(z-20)=0 THEN LET p=1
1090 IF (a(z-14)=1 OR a (z-14)=2) AND
a(z-18)=0 THEN LET p=2
1100 IF p=1 THEN LET a(z-15)=0: LET
a(z-20)=a(z-10)
1110 IF p=2 THEN LET a(z-14)=0: LET
a(z-18)=a(z-10)
1120 IF p>0 THEN LET a(z-10)=0
1130 GO TO 50
1140 LET a(z+8)=-2
1150 LET a(z+4)=0
1160 LET a(z)=0
1170 LET si=si+1
1180 IF z<32 AND (a(z+3)=1 OR a (z+3)=2) AND
a(z-2)=0 THEN LET p=1
1190 IF z<23 AND (a(z+14)=1 OR a(z+14)=2) AND
a(z+16)=2 THEN LET p=2
1200 IF z<23 AND (a(z+13)=1 OR a(z+13)=2) AND
a(z+16)=0 THEN LET p=3
1210 IF p=1 THEN LET a(z+3)=0: LET a(z-2)=-2
1220 IF p=2 THEN LET a(z+14)=0: LET a(z+16)=0
1230 IF p=3 THEN LET a(z+13)=0: LET
a(z+18)=-2
1240 IF p>0 THEN LET a(z+8)=0
1250 LET a(z+10)=-2
1260 LET a(z+5)=0
1270 LET a(z)=0
1280 LET si=si+1
1290 GO TO 50
1300 PRINT : PRINT
1310 PRINT : PRINT
1320 RETURN
1330 REM * INDICAR PELAS INICIAIS *
1340 DIM a(45)
1350 PRINT
1360 FOR z=1 TO 45
```



```
1370 IF z<6 THEN LET a(z)=9
1380 IF z>5 AND z<19 THEN LET a(z)=1
1390 IF z>18 AND z<28 THEN LET a(z)=0
1400 IF z>27 AND z<41 THEN LET a(z)=-1
1410 IF z>40 THEN LET a(z)=9
1420 NEXT z
1430 LET a(14)=9: LET a(23)=9: LET a(32)=9
1440 LET f$="--"
1450 LET p=0: LET q=0
1460 LET si=0: LET sm=0
1470 PRINT : PRINT
1480 INPUT "Quer fazer a primeira jogada y/n"; LINE
q$
1490 LET u$=" "
1500 PRINT
1510 IF q$ < > "Y" THEN GO TO 680
1520 LET u$="".
1530 PRINT
1540 GO TO 50
1550 REM Descodificar a jogada
1560 LET i=1
1570 IF m$(i)=d$ THEN GO TO 1610
1580 LET i=i+1
1590 IF i=33 THEN LET i=0: GO TO 1610
1600 GO TO 1570
1610 RETURN
1620 DIM m$(32,2): DIM m(32)
1630 DATA "B1", "D1", "F1", "H1", "A2", "C2", "E2",
"G2", "B3", "D3", "F3", "H3", "A4", "C4", "E4", "G4",
"B5", "D5", "F5", "H5", "A6", "C6", "E6"
1640 DATA "G6", "B7", "D7", "F7", "H7", "A8", "C8",
"E8", "G8"
1650 DATA 40,39,38,37,36,35,34,33,31,30
1660 DATA 29,28,27,26,25,24,22,21,20,19
1670 DATA 18,17,16,15,13,12,11,10,9,8,7,6
1675 RESTORE 1630
1680 FOR i=1 TO 32
1690 READ m$(i)
1700 NEXT i
1710 FOR i=1 TO 32
```

```
1720 READ m(i)
1730 NEXT i
1740 RETURN
9000 BORDER 0: PAPER 0: INK 9: CLS
9005 RESTORE 9030: FOR a=USR "a" TO USR "a"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,60,126,126,126,126,60,0
9998 REM a
9999 REM ■
```

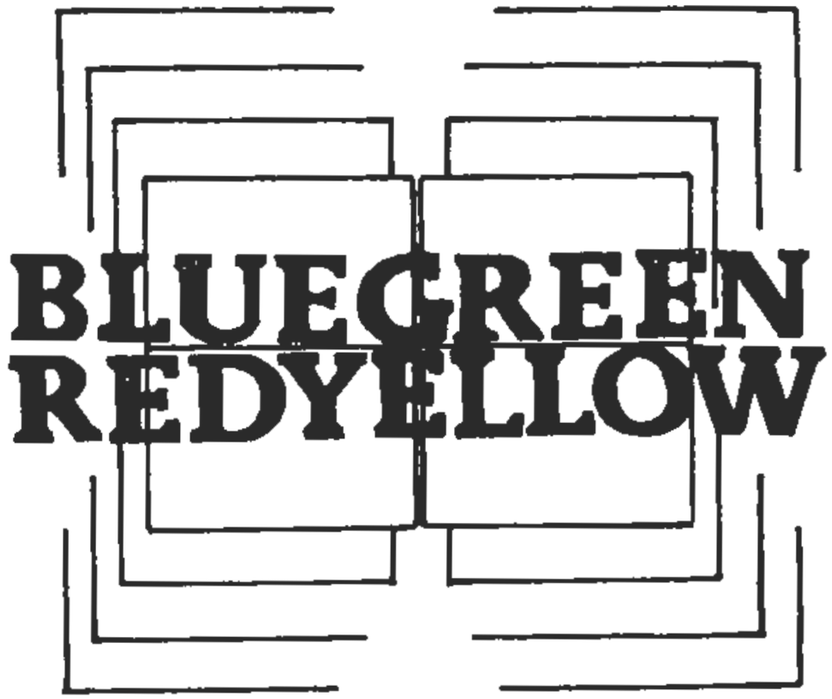


SIMÃO DISSE

Este jogo é uma versão de um conhecido brinquedo com o mesmo nome. O computador selecciona uma sequência de cores com acompanhamento de sons, e você tem de repetir exactamente a sequência. Há quatro cores diferentes. A sequência começa com uma cor escolhida ao acaso e sempre que se repete a sequência é adicionada mais uma cor. O comprimento máximo da sequência é de um a dez.

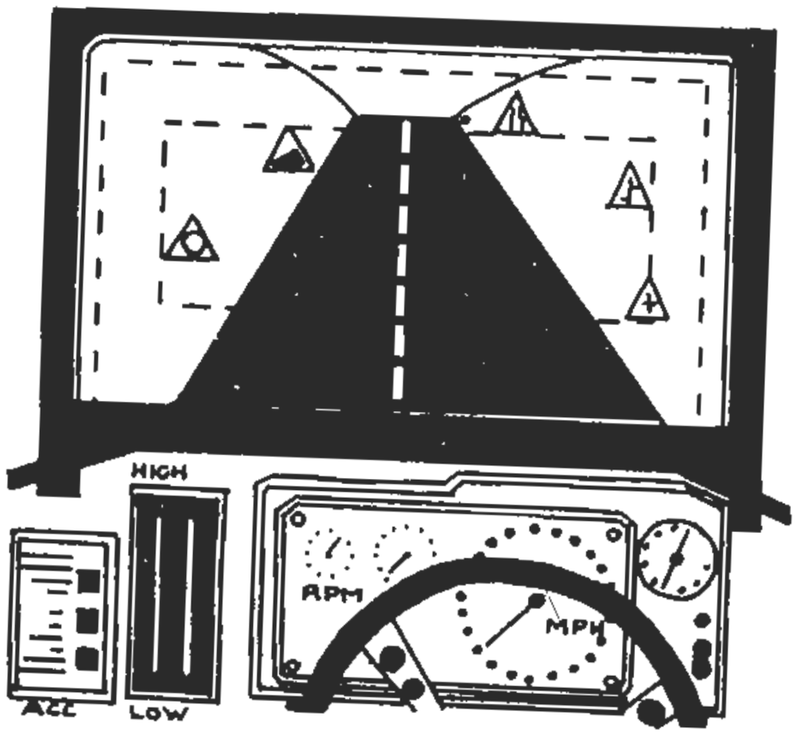
```
  10 REM Simão disse...
  20 REM Tim Hartnell
         Peter Shaw
  30 GO SUB 9000: GO SUB 8000
  40 FOR a=1 TO 10
  50 LET c$=c$+STR$ (INT (RND*4) +1)
  60 NEXT a
  70 FOR q=1 TO x
  80 LET l=4*(CODE c$(q) -48)
  90 LET t=VAL c$(q)
 100 BEEP .05,10*t
 110 PRINT AT 1,7; INK 6; " ■ "; INT l/4; AT
1-1,7; " ▬ "; AT 1+1,7; " ▬ "
 120 PAUSE 4
 130 CLS
```

```
 140 NEXT q
 150 FOR b=1 TO x
 160 IF INKEY $< > "" THEN GO TO 160
 170 LET t$=INKEY$
 180 IF CODE t$=0 THEN GO TO 170
 190 CLS
 200 LET y=4*(CODE t$-48)
 210 PRINT AT y,7; INK y/4; "  "; INK 6; t$; AT
y-1,7; INK y/4; "  "; AT y+1,7; "  "
 220 BEEP .04,2.5*y
 230 IF CODE t$< > CODE (c$(b)) THEN GO TO 1000
 240 PAUSE 7
 250 CLS
 260 NEXT b
 270 IF x=10 THEN PRINT "Você bateu a máquina!":
INPUT "Carregue em Enter para jogar novamente"; LI
NE a$: RUN
 280 LET x=x+1
 290 PAUSE 50
 300 GO TO 70
1000 PRINT "Você marcou"; x-1
1010 BEEP .02,RND*30
1020 INPUT "Carregue em Enter para jogar nova
mente"; LINE a$: RUN
8000 BORDER 0: PAPER 0: INK 7: CLS : BRIGHT 1
8010 LET c$=""
8020 LET x=1
8990 RETURN
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,0,0,0,3,7,15,15
9040 DATA 0,0,0,0,192,224,240,240
9050 DATA 15,15,7,3,0,0,0,0
9060 DATA 240,240,224,192,0,0,0,0
```

CONDUTOR EM 3D

Este jogo é uma adaptação esquemática dos jogos de condução existentes nos salões de jogos. Você tem uma perspectiva tridimensional do seu carro e da estrada. A estrada anda para cá e para lá, e você tem de evitar que o seu carro saia da estrada. Use as teclas 5 e 8 do cursor para controlar o seu carro.

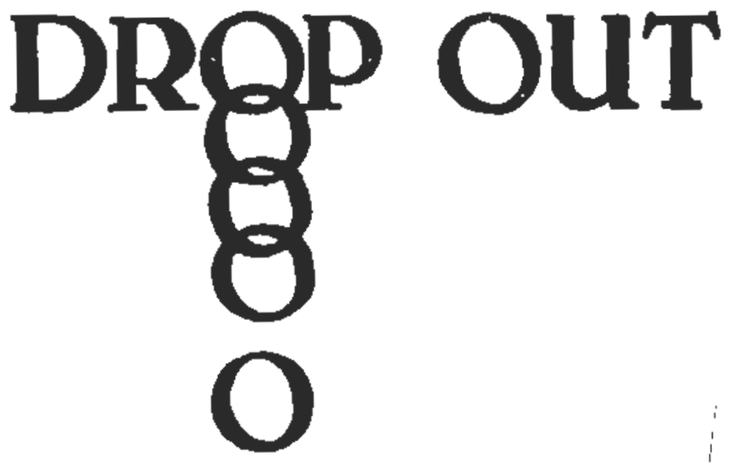



```
  10 REM Condutor em 3D
  20 GO SUB 9000
  30 GO SUB 8000
  40 PRINT AT 15,15; INK 4; "  ▀▄  "; AT 16,15; "▄▀▀▄"
  50 IF p<108 OR p>150 THEN GO TO 1000
  60 PLOT p-10,159: DRAW -170+(p-10), -159
  70 PLOT p+10,159: DRAW -70+(p-10), -159
  80 PLOT OVER 1; p-10, 159: DRAW OVER 1;
-170+(p-10), -159
  90 PLOT OVER 1; p+10,159: DRAW OVER 1;
-70+(p-10), -159
 100 LET p=p+4*(INKEY$="5") -4*(INKEY$="8")
 110 LET p=p + (INT (RND*20) -10)
 115 LET sc=sc+1
 120 GO TO 50
 990 STOP
1000 PLOT p-10,159: DRAW -170+(p-10) -159
1010 PLOT p+10,159: DRAW -70+(p-10), -159
1020 PRINT AT 15,15; FLASH 1; INK2; "  ▀▄  "; AT
16,15; "▄▀▀▄"
1030 FOR a=0 TO 50: BEEP .05, -a: BEEP. 008, a:
NEXT a
1040 PRINT AT 1,10; PAPER 1; "Você marcou "; sc
1050 INPUT "Carregue em"; INK 6; " ENTER "; INK
7; "para jogar      novamente"; LINE a$: GO TO 30.
8000 BORDER 0: PAPER 0: INK 7: CLS
8010 LET p=127
8020 LET sc=0
8040 PRINT AT 0,0; PAPER 1; " ____________________
____________________"
8990 RETURN
9000 FOR a=USR "a" TO USR "d"+7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,0,0,0,13,11,13,3
9040 DATA 0,0,0,0,176,208,176,192
9050 DATA 6,117,207,223,208,240,12,3
9060 DATA 102,174,243,251,19,15,48,192
9998 REM ab
         cd
9999 REM ▄▀
```

METER A BOLA NO BURACO

Este jogo é relativamente simples. Coloca-se uma bola em cima de uma plataforma alta; tem que calcular a força necessária para a fazer entrar num buraco no fundo do écran. Para o conseguir introduza-se no computador um número entre um e nove.

Este programa ajudá-lo-á a calcular as distâncias, e vai ser-lhe útil quando quiser fazer programas em que o texto é alinhado no centro do écran.

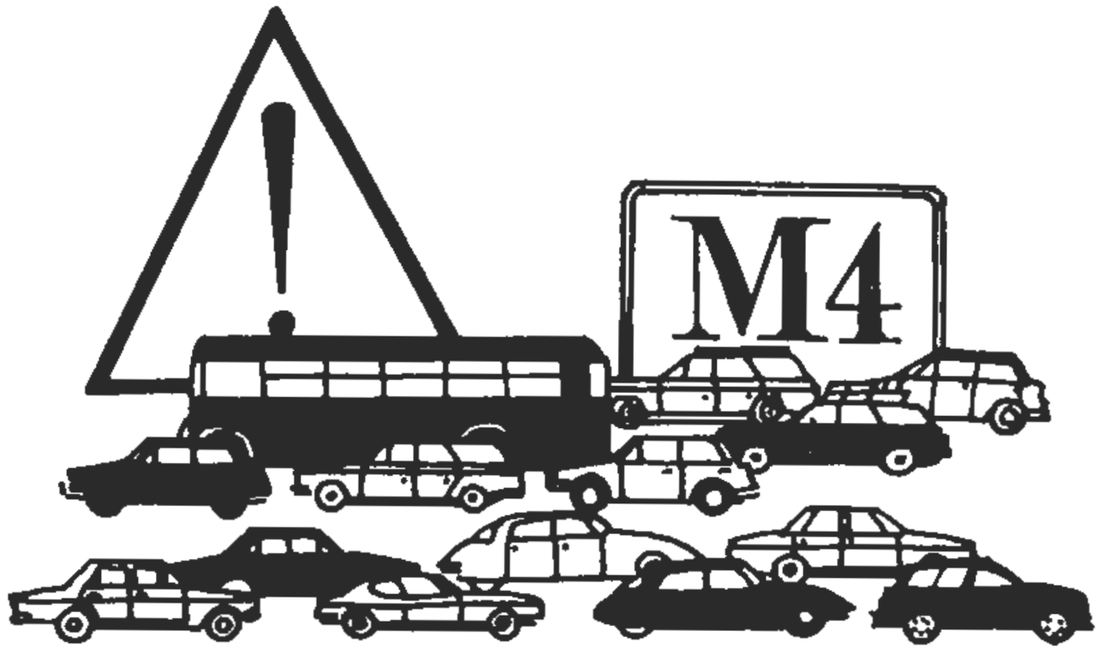



```
 10 REM Meter a bola no buraco
 20 GO SUB 9000
 30 GO SUB 8000
 40 GO SUB 7000
 50 INPUT "Força ? "; LINE a$: IF a$>"9" OR
a$<"1" THEN GO TO 50
 60 FOR a=0 TO 8
 70 PRINT AT 3,a; "■"
 80 BEEP .1,a
110 PRINT AT 3,a; "  ": NEXT a
120 FOR a=4 TO 15
130 PRINT AT a,8;"■"
140 BEEP .1,a
170 LET b=b+.5
180 PRINT AT a,8; "  "
190 LET a=a+b
200 NEXT a
210 PRINT AT 15,8; "■"
220 BEEP .2,5
230 PRINT AT 15,8; INK 1; "▙"
240 FOR a=1 TO VAL a$*2
250 PRINT AT 15,8+a; "●"
260 BEEP .1,a
290 PRINT AT 15,8+a; "  "
300 NEXT a
310 IF VAL a$ = h THEN PRINT AT 16,8 + (h*2);
"●": LET sc = sc+1: PRINT AT 21,17; PAPER 6; INK 9;
"PONTUAÇÃO "; sc: BEEP 1,30
315 BEEP 1,10
320 GO TO 40
7000 REM
7010 FOR a=4 TO 15
7020 PRINT AT a,0; INK 1; "[redacted]"
7030 NEXT a
7040 FOR a=16 TO 21
7050 PRINT INK 1; "[redacted]
[redacted]"
7060 NEXT a
7070 PRINT AT 15,8; INK 1; "▙"
7080 LET h=INT (RND*9) +1
```

```
7090 PRINT AT 16,8+(h*2); " "
7100 RETURN
8000 LET sc=0
8010 BORDER 0: PAPER 0: INK 6: CLS
8020 LET b=.5
8990 RETURN
9000 FOR a=USR "a" TO USR "b" +7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 60,126,255,255,255,255,126,60
9050 DATA 128,128,128,128,128,128,224,255
9980 REM a  b
9990 REM ●  ⌊_
```


TRAVESSIA DA AUTO-ESTRADA

Atravessar a auto-estrada à hora de ponta não é a coisa mais agradável do mundo, mas é o objectivo deste jogo. Tem de evitar os carros que circulam em duas faixas, em direcções opostas; se for atropelado por um carro perde uma das suas cinco vidas. Marca 10 pontos por cada homem que atravessa a auto-estrada sem ser atropelado. Toma-se em consideração a pontuação máxima. Utilize as teclas do cursor para controlar os seus movimentos.



```
  10 REM Travessia da auto-estrada
  20 GO SUB 9000: LET hi=0
  30 GO SUB 8000
  40 LET v=14: LET h=16: GO SUB 7000
  50 PRINT AT v,h; PAPER 8; "  "
  60 LET v=v+(INKEY$="6" AND v<21)-(IN
KEY$="7" AND v>0)
  70 LET h=h+(INKEY$="8" AND h<31)-(IN
KEY$="5" AND h>0)
  80 PRINT AT 4,0; INK 6; a$; AT 21,0; b$; AT 5,0;
b$; AT 12,0; a$
  90 IF SCREEN$ (v,h)="  "THEN GO TO 1000
 100 PRINT AT v,h; PAPER 8; "  "
 110 LET b$=b$(32) +b$(TO 31)
 120 LET a$=a$ (2 TO) +a$(1)
 130 IF v=2 THEN GO TO 2000
 150 GO TO 50
1000 PRINT AT v,h; FLASH 1; "  "
1010 DIM s$ (1,704)
1020 PRINT AT 0,0; FLASH 8; OVER 1; PAPER 8; INK
2; s$(1)
1025 IF ml>0 THEN LET ml=ml-1: GO TO 40
1026 IF sc>hi THEN LET hi=sc
1030 INPUT "Carregue em ENTER para jogar nova
mente"; LINE b$
1040 GO TO 30
2000 LET sc=sc+10
2010 GO TO 40
7000 CLS
7010 PRINT' TAB 12; "PONTUAÇÃO MÁXIMA "; hi;
TAB 23; "PONTUAÇÃO "; sc; PAPER 1; "
                                    "
7020 PRINT INK 6 'a$' b$
7030 PRINT ' PAPER 4; "
                      "
7050 PRINT INK 6'b$' a$
7060 PRINT ' PAPER 1; "
                      "
7070 PRINT AT v,h; PAPER 8; "  "
7080 FOR a=1 TO ml: PRINT AT 1,a; INK 5; "  ";:
```

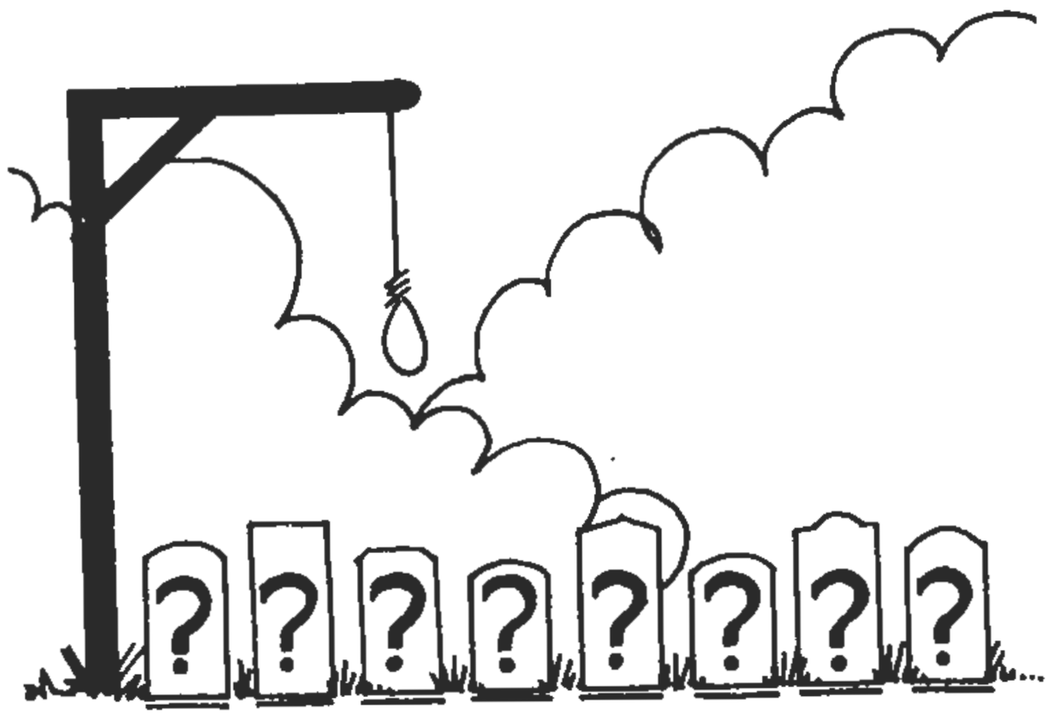


```
NEXT a
7090 RETURN
8000 BORDER 0: PAPER 0: INK 9: CLS
8010 LET sc=0
8030 LET a$="                    "
8040 LET b$="                    "
8060 LET ml=5
8990 RETURN
9000 FOR a=USR "a" TO USR "e" +7
9010 READ user: POKE a, user
9020 NEXT a RETURN
9030 DATA 0,1,2,127,235,253,26,6
9040 DATA 0,240,16,252,215,167,56,16
9050 DATA 0,15,8,63,235,221,26,6
9060 DATA 0,128,64,254,215,191,28,8
9070 DATA 28,28,8,8,62,8,28,34
```

A FORCA

Este jogo é uma versão de um jogo de palavras muito conhecido. O computador faz aparecer no écran um certo número de traços que correspondem ao número de letras da palavra. Você introduz uma letra, e se essa letra fizer parte da palavra aparece no écran e você pode adivinhar outra letra. Se o seu palpite falhou o computador começa a desenhar o enforcado. Tem 13 hipóteses antes do homem ser enforcado, por isso não se ponha a recitar por ordem as letras do alfabeto, pois por esse processo não chega lá.

Quando escolhe uma letra e a introduz no computador, o computador põe na palavra todas as letras iguais a essa que ela contém; por exemplo, carro tem dois *rr*, e por isso o computador imprime os dois *rr* no écran. Como pode ver pela listagem deste programa, o dicionário de palavras admitidas é muito extenso, e por isso a pessoa que introduz o programa pode jogar sem saber logo qual é a palavra escolhida.




```
10 REM A Forca
20 GO SUB 9000
30 GO SUB 8000
40 GO SUB 7000
50 FOR a=1 TO LEN w$
60 PRINT "_ ";
70 NEXT a
80 INPUT "Adivinhe (só 1 letra)"; LINE n$
90 LET r=0: FOR a=1 TO LEN w$
100 IF w$(a)=n$ THEN LET r=r+1
110 NEXT a
120 IF r< >0 THEN GO TO 1000
130 LET t$=t$+n$+"  "
140 GO SUB 6000
150 LET c$="": FOR a=1 TO LEM w$: LET
d$=SCREEN$ (0, (a*2) -2): LET c$=c$+d$: NEXT a:
IF c$=2$ THEN GO TO 5000
160 GO TO 30
1000 FOR a=1 TO LEN w$
1010 IF w$(a)=n$ THEN PRINT AT 0 (a*2) -2; n$
1020 NEXT a
```

```
1030 GOTO 150
2000 PRINT AT 5,19; "l": BEEP 1, -10
2010 PRINT AT 5,0; "Você matou-o!"
2011 PRINT AT 7,0; "a palavra era"; AT 8,0; w$
2020 INPUT "Carregue em enter para jogar nova
mente"; LINE j$: GO TO 30
5000 PRINT AT 5,0; "Acertou!"
5010 INPUT "Carregue em enter para jogar nova
mente"; LINE j $: GO TO 30
6000 LET d=d+2: LET h=h+2: IF h>31 THEN LET
v=v+1: LET h=18
6010 PRINT AT v,h; INK 4; n$
6020 PRINT AT INT (d/2)+2-5,17; INK 6; b$(INT
((d-16) /2))
6030 IF INT ((d-16) /2) =13 THEN GO TO 2000
6040 RETURN
7000 RESTORE 7000
7005 FOR a=1 TO INT (RND*193) +1
7010 READ w$
7020 NEXT a
7030 DATA "avião", "aeroplano", "antrópode", "anão",
"afluxo", "aarónidas", "abacate", "abalumar", "bigorna",
"binário", "bétula", "bizarro", "bocado", "bolha", "buril",
"brota", "bobo", "borla", "carrinho", "caracol", "cére
bro", "câmara", "canoa", "capuz", "cuidado", "castanho
las", "caçar", "carroça", "carruagem"
7040 DATA "duplo", "duna", "dual", "dobre", "desalo
jar", "encontro", "empregar", "engenho", "envelope",
"equinóxio", "empalidecer", "fada", "fenda", "faúlha",
"fastio", "fascista", "fateixa", "feitiço"
7050 DATA "geometria", "guru", "grau", "golo", "gomo
so", "golfinho", "gordo", "governo", "hospital", "heráldi
ca", "helicóptero", "heliogravura", "hexagrama",
"hidrosfera", "império", "insidioso", "inspector", "juve
nil", "jardineira", "jarreta", "jerico", "kantiano",
"kantista", "kneippista", "lázaro", "landau", "lábio",
"lua", "labirinto", "laranja", "luneta"
7060 DATA "massa", "macio", "médio", "montada",
moinho", "motor", "mistificar", "marceneiro", "ma
nual", "montar", "mar", "nostalgia", "nervo", "nicho",
```



```
"órfão", "objecto", "obtuso", "ozono", "órbita", "outro-
ra", "ordinário", "outorgante", "ourives", "ostrogodo"
7070 DATA "pálio", "panqueca", "pantógrafo", "parce
la", "parlamento", "partícula", "pelourinho", "prega",
"poesia", "portagem", "projecto", "planear", "planalto",
"peixe", "peixeira", "pneumonia", "podar", "plural",
"quadrilha", "quadrado", "quaresma", "queimadela",
"rusga", "recíproco", "reencarcerar"
7080 DATA "remanescer", "residência", "renda", "rui
vo", "rádio", "sábio", "sentença", "semáforo", "seara",
"seareiro", "sedativo", "senhorial", "selvagem", "sáti
ro", "saber"
7090 DATA "tetina", "ténis", "tétano", "tiquetaque",
"tigre", "titânico", "torrada", "tónico", "tórrido", "tri
bo", "teatro", "torno", "tareia", "tendão", "tribunal",
"triunvirato", "universo", "unha", "ultramarino", "um
bigo", "umbreira", "ultrajante"
7100 DATA "vácuo", "valentim", "vagante", "válvula",
"variável", "veia", "visita", "variante", "wagneriano",
"walkéria", "wernerite", "wilsónia", "xabregas", "xaile",
"xadrez", "xarofagia", "xeta", "xexuão", "xiloma", "xo
dó", "xistoso", "yverdoniano", "yorkshiriano", "yersia
no", "zebra", "zoólogo", "zagaia", "zangorro"
7110 RETURN
8000 BORDER 0: PAPER 0: INK 9: CLS
8010 DIM b$(13,5)
8020 LET b$(1)= "     "
8030 LET b$(2)= "     "
8040 LET b$(3)= "     "
8050 LET b$(4)= "     "
8060 LET b$(5)= "     "
8070 LET b$(6)= "     "
8080 LET b$(7)= "     "
8090 LET b$(8)= "     "
8100 LET b$(9)= "     "
8110 LET b$(10)="     "
8120 LET b$(11)="     "
8130 LET b$(12)="     "
8140 LET b$(13)="     "
8150 LET t$="     "
```

```
8160 LET h=16:  LET  v=0
8170 LET d=16
8990 RETURN
9000 RESTORE 9000: FOR a=USR "a" TO USR "g" +7
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 0,0,0,7,15,28,24,24
9040 DATA 24,24,28,15,7,0,0,0
9050 DATA 0,0,0,224,240,56,24,24
9060 DATA 24,24,56,240,224,0,0,0
9070 DATA 0,0,0,255,255,0,0,0
9080 DATA 24,24,24,24,24,24,24,24
9090 DATA 0,0,102,102,0,0,0,0
```


O VENDEDOR DE VIDEO

Quanto tempo consegue aguentar-se na selva que é o mundo da venda de aparelhagem video? Neste programa, você compra e vende gravadores video para ganhar dinheiro. Compra os gravadores video a 200 £ cada, e vende-os ao preço que entende. O número de gravadores vendidos em cada semana depende do preço de venda e da situação do mercado do video (estes dados são apresentados pelo computador antes de cada jogada).

Tem de ter cuidado quando compra material, pois se não vender os gravadores nessa semana já os não pode vender na semana seguinte. O jogo dura cinco semanas e começa com um capital de 1000 £ .

```
  10 REM O Vendedor de Video
  20 REM
  30 GO SUB 8000
  40 INPUT AT 22,0; AT 0,6; "Quantos jogadores "; LI
NE p$; AT 10,2; "Carregue em ENTER para jogar"; AT
22,30; LINE a$
  50 LET p=VAL p$: IF p < 1 THEN GO TO 40
  55 DIM s(p): DIM c (p): FOR a=1 TO p: LET c(a)=
1000: NEXT a
```

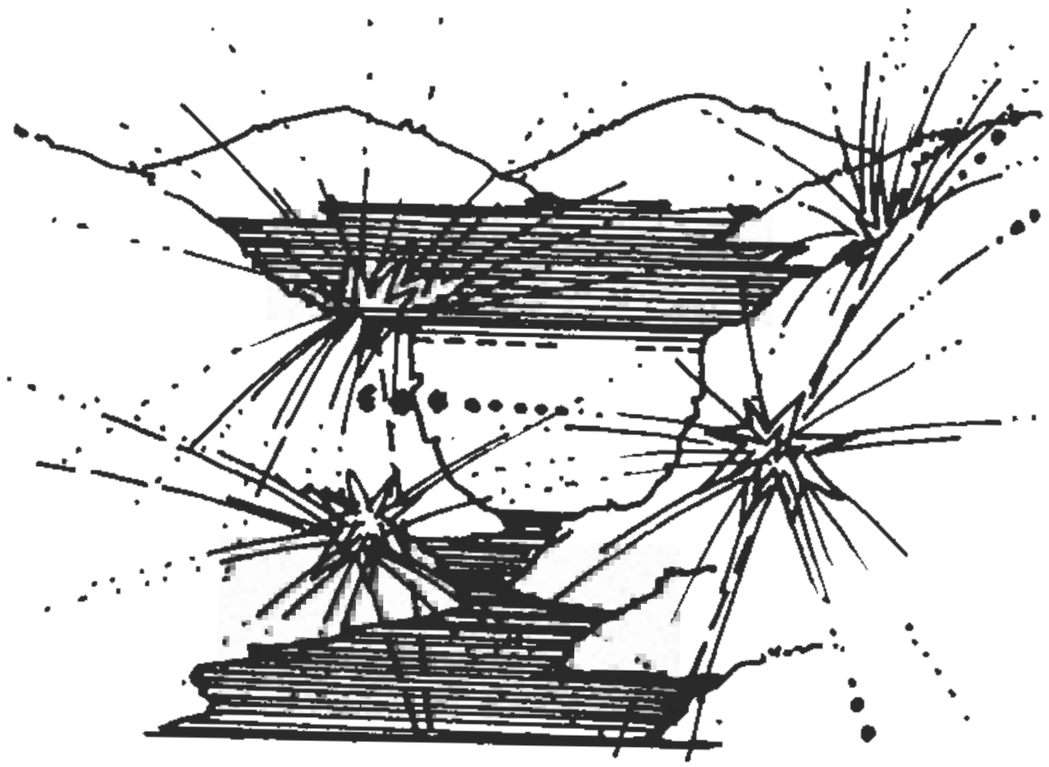
```
  60 FOR w=1 TO 5
  70 FOR l=1 TO p
  75 IF c(l)<200 THEN PRINT "vendedor " l„ "Já
não tem dinheiro para comprar mais material": NEXT
l: NEXT w: GO TO 270
  80 CLS
  90 PRINT "Vendedor "; l, "Vendas "; s (1)
  95 PRINT "Dinheiro em caixa "; c(1)
 100 GO SUB 2000
 110 PRINT' "Notícias locais e nacionais: —"
 120 PRINT h$
 130 INPUT AT 15,0; AT 0,0; "Quantos gravadores vai
ter em   stock esta semana ( £200 p.uni.)"; re; AT 5,0;
"Quanto vai pedir por cada grava dor "; ch; AT 15,5;
"Carregue em ENTER para continuar"; LINE a$
 135 IF re*200>c (1) THEN GO TO 130
 140 CLS
 150 PRINT "Vendedor "; l
 160 LET sa=(h*(10) / (ch/10))
 170 IF sa>re THEN LET sa=re
 180 PRINT' "Dinheiro em caixa "; c(l)
 190 LET s(l)=INT sa
 200 PRINT "Vendas esta semana "; s (1)
 210 LET pr=(s (1)*ch) - (re*200)
 220 PRINT "Lucro "; pr
 230 LET c(l)=c (1)+pr
 240 PRINT "Novo balanço "; c(l)
 250 INPUT "Carregue em ENTER para continuar";
LINE a$: NEXT l
 260 NEXT w
 990 STOP
2000 LET h=INT (RND*20) + 1
2010 IF h>10 THEN GO TO 2500
2020 RESTORE 2000
2030 FOR a=1 TO INT (RND*6) + 1: READ h$: NEXT
a
2040 RETURN
2050 DATA "É a semana da Taça das Nações!", "O E.T.
acabou de ser lançado oficialmente em video", "Baixa
de preço dos gravadores video", "Oferta especial de
```



```
cassettes virgens", "O Natal aproxima-se", "Abriu na
cidade uma nova loja de aluguer de cassettes video"
2500 RESTORE 2500
2510 FOR a=1 TO INT (RND*5) +1: READ h$: NEXT
a
2520 RETURN
2530 DATA "A inflação prejudica a indústria do vi
deo", "Os discos video baratos vendem-se cada vez
mais", "A televisão por cabo está a prejudicar as
vendas de video", "O cinema tem cada vez mais espec
tadores", "Depressão na indústria de video»
8000 BORDER 0: PAPER 0: INK 9: CLS
9990 RETURN
```

O BOMBARDEAMENTO DO DIQUE

Quando regressa de uma missão à Europa tem problemas no motor e começa a perder altura. O único lugar onde pode aterrar em segurança é o rio. Mas infelizmente há um dique que o impede de aterrar, e portanto tem de rebentar com o dique. Use a tecla 0 para largar as bombas.




```
 10 REM O Bombardeamento do Dique
 20 GO SUB 9000: LET hi=0
 30 GO SUB 8000
 40 GO SUB 7000
 50 FOR v=1 TO 15
 55 PRINT AT v-1,0; PAPER 8; "       "
 60 FOR h=0 TO 31
 70 PRINT AT v,h; INK 6; PAPER 8; "  ■■■■ "
 80 IF SCREEN$ (v,h+6) < >"   " THEN GO TO 1000
 90 IF INKEY$="0" THEN GO SUB 200
 95 LET sc=sc+1
100 BEEP (.008 AND ch=1) + (.01 AND ch=0) ,0
105 IF ch=0 THEN GO TO 120
110 IF (SCREEN$ (13,16)= "  " AND SCREEN$
(13,17)="  " AND SCREEN$ (13,18) ="  ") THEN
LET ch=0: PRINT AT 13,0;"                    "AT
14,20; INK 2; PAPER 1; "▙            "; AT 13,16;
PAPER 0; "    "
180 NEXT h
190 NEXT v
195 GO TO 1500
200 PRINT AT v+1,h; INK 5; PAPER 8; "  ▀ "
210 PRINT AT v,h; PAPER 8; INK 6; "  ■■■■ "
220 LET h=h+1: IF h>30 THEN LET v=v+1: LET
h=0
240 PRINT AT v+1,h; INK 5; PAPER 8; "  ▟ "
250 PRINT AT v,h; PAPER 8; INK 6; "  ■■■■ "
260 LET h=h+1: IF h>31 THEN LET v=v+1: LET
h=0
275 PRINT AT v+2,h; PAPER 8; "  "
280 LET b1=h
290 FOR b=v+1 TO 14
300 PRINT AT b,b1; PAPER 8; INK 5; "  "; AT
b+1,b1; "▚ "
310 PRINT AT v,h; INK 6; PAPER 8;"  ■■■■ "
320 LET h=h+1: IF h>30 THEN LET v=v+1: LET
h=0
330 IF SCREEN$ (b+2, b1) < >"    " THEN GO TO
500
335 PRINT AT v-1, 31; "  "
```

```
 340 NEXT b
 350 PRINT AT b,b1; PAPER 8; "  "
 360 LET h=h-1
 380 RETURN
 500 FOR b=b TO b+1
 510 PRINT AT b,b1; PAPER 8; "  "; AT b+1,b1; INK 5;
"▚"
 520 NEXT b
 524 BEEP .005, -b
 525 LET h=h-1: PRINT AT b,b1; PAPER 8; "  "
 530 RETURN
1000 FOR a=v TO 15
1010 PRINT AT a,h+1; PAPER 1; "▚▚▚▚"; AT
a+1,h+1; INK 6; "          "
1020 BEEP .5,-a: NEXT a
1030 GO TO 1510
1500 LET sc=sc+100: PRINT AT 0,12; "FIM DO JOGO"
1510 PRINT AT 5,10; "Você marcou "; sc
1520 IF sc>hi THEN LET hi=sc
1530 PRINT AT 10,6; "Pontuação máxima de hoje ";
hi
1540 INPUT "Carregue em ENTER para jogar   nova
mente"; LINE a$: GO TO 30
7000 CLS
7010 PRINT AT 13,0; PAPER 1; "                    "
7020 PRINT AT 14,0: PAPER 1; "                    "
7030 PRINT AT 15,0; PAPER 1; "                    "
7040 PRINT AT 12,16; INK 2; "▙"; AT 13,16; PAPER
1; "▀"; PAPER 0; "▙"; AT 14,16; PAPER 1; "▀▀"; PA
PER 0; "▙"; AT 15,16; PAPER 1; "▀▀▀▀▙"
7050 FOR a=1 TO 4
7060 PRINT PAPER 2; "
      "
7070 NEXT a
7990 RETURN
8000 BORDER 0: PAPER 0: INK 9: CLS
8010 LET sc=0
8020 LET ch=1
8990 RETURN
9000 FOR a=USR "a" TO USR "i" +7
```



```
9010 READ user: POKE a, user
9020 NEXT a: RETURN
9030 DATA 56,60,63,63,63,7,0,0
9040 DATA 0,0,128,255,255,255,255,0
9050 DATA 0,0,63,255,255,255,255,0
9060 DATA 0,0,224,248,248,248,240,0
9070 DATA 0,0,220,126,126,220,0,0
9080 DATA 0,0,20,254,30,14,0,0
9090 DATA 36,60,24,60,60,24,0,0
9100 DATA 128,192,224,240,246,252,254,255
9110 DATA 254,254,255,255,255,255,255,255,255
9990 REM a  b  c  d  e  f  g  h  i
9999 REM ▚ ▚▚ ▚▚ ▚▚ ▚▚ ▚▚ ▚▚ ▚▚ ▚▚
```

COMO FAZER PROGRAMAS MELHORES

por Tim Hartnell, editor da colecção

Este livro contém uma série de excelentes programas, e pode encontrar muitos outros igualmente bons em diversas revistas de computadores. Por muito bons que sejam porém os programas publicados em livros ou revistas, o leitor divertir-se-á muito mais executando programas que foram parcial ou totalmente escritos por si. Imprimir aos programas a sua marca pessoal, alterando-os de modo a reflectirem os seus desejos e a sua criatividade, é uma boa maneira de melhorar os programas e por outro lado ajuda-o a ser um programador mais eficiente e imaginativo.

Os programas publicados em revistas e em livros como este são ideais como ponto de partida para as suas próprias criações. Os anúncios dos «packages» de software podem ser também um bom ponto de partida para as suas próprias criações. Basta ler a descrição do que faz o programa comercial proposto, e está dado o primeiro passo para a criação do seu próprio programa. É claro que tem de ter cuidado para não infringir a legislação de protecção dos direitos de autor tanto no que se refere à visualização gráfica, como ao nome do programa ou aos nomes dos «caracteres» do programa. Mas vai ver que quando o programa atinge um certo estádio de

desenvolvimento parece adquirir vida própria, desenvolvendo-se e evoluindo de modo a diferenciar-se do cenário original, de tal modo que acaba por surgir um jogo completamente novo no conceito e na prática.

De qualquer maneira nunca deve tentar apresentar como sendo da sua autoria trabalhos de outros. Isto não quer dizer que não tente adaptar ou melhorar programas publicados, mas sim que os não deve enviar a revistas como se fossem originais. Já perdi a conta do número de vezes que me enviaram para publicação programas feitos por mim e extraídos de um dos meus livros.

Mantenha-se sempre atento e procure novas ideias quando lê livros ou revistas de jogos e de computadores, ou quando vai a salas de jogos electrónicos. Pode valer a pena tomar nota das ideias de jogos, figuras, sons, fins dramáticos, etc., que lhe vão surgindo. Se assim proceder nunca lhe faltarão ideias e poderá combinar esse material de modo a produzir jogos melhores, que interessarão os jogadores durante mais tempo.

De uma maneira geral podemos classificar os jogos em três categorias, e vale a pena definir a categoria em que se integrará o seu novo programa *antes* de começar a programar, pois a categoria do jogo determinará o método de programação. O que não quer dizer que não seja possível mudar de uma categoria para outra no decurso da programação, ou que um determinado jogo não possa pertencer simultaneamente a duas categorias; no entanto vale a pena ter uma ideia definida dos diferentes grupos, para saber bem o que se está a fazer. Essas três categorias são:

1. Jogos de tabuleiro.
2. Jogos de «sala de jogos» (ou seja, jogos predominantemente visuais, de movimentação rápida, ruidosos, actuais).
3. Jogos de azar (como a roleta ou o poker).

Nos jogos de tabuleiro a qualidade da jogada é mais importante do que uma resposta-relâmpago, enquanto que nos programas do tipo «sala de jogos» a movimentação tem de ser constante, ainda que se tenham de sacrificar por vezes algumas vidas na luta contra os invasores de Marte. Os jogos de azar dependem mais da oportunidade das jogadas (inputs «favoráveis ao utilizador») e do puro acaso do que qualquer uma das outras categorias.

Verificará que os programas de jogos podem geralmente ser classificados em tipos que são subdivisões das três categorias atrás mencionadas. Muitos jogos de tabuleiro são variantes do xadrez ou das damas; muitos jogos de acção começaram como jogos do tipo Invasores do Espaço; e os jogos de azar derivam do «mundo real» dos dados e das cartas. Se examinar a descrição de um programa ou uma máquina de jogos, tentando classificar o jogo numa categoria, pode obter ideias novas adequadas ao tipo específico a que ele pertence.

Há na programação uma escola de pensamento — a que se dá geralmente o nome de «programação estruturada» — que considera que é essencial proceder metodicamente quando se redige um programa. Se bem que este método seja menos interessante do que o procedimento que consiste em sentar-se directamente em frente do computador, o programa elaborado será muito melhor. Fiz uma vez um programa chamado «Habitantes de uma Estação Lunar», um programa de simulação em que o jogador é responsável por uma «estação lunar» e tem de decidir quais os produtos a manufacturar e vender para comprar oxigénio e comida para os habitantes da estação. (Este programa foi publicado no meu livro *The Book of Listings*, redigido em colaboração com Jeremy Ruston e editado pela BBC). Depois de ter escolhido o cenário geral, come-

cei a trabalhar na visualização gráfica, e tive a seguinte ideia:

As reservas de oxigénio são baixas
Há 96 pessoas a viver na estação no ano 3
O crédito é de $5.693
O custo anual de manutenção é de $226
O tanque de oxigénio tem uma capacidade de 811 unidades
O preço do oxigénio é de $8 por unidade
Cada um dos habitantes da estação precisa de 5 unidades por ano
As reservas de comida montam a 2122 unidades
Cada um dos habitantes precisa de 3 unidades por ano (a $6 por unidade, $576 para a estação. Para a população actual estas condições manter-se-ão durante 7 anos).
Pode vender as suas esculturas lunares originais às pessoas que vivem nas outras estações. Para fabricar uma escultura gastam-se 2 unidades de oxigénio, e o preço de venda de cada escultura é de $30.

Como o leitor pode ver por este «esquema», a finalidade do programa consiste em decidir quantas «esculturas lunares originais» é preciso fabricar e vender para comprar oxigénio e comida e para pagar o custo «anual de manutenção». O problema deste programa específico reside no facto de se ter de gastar oxigénio para fabricar uma escultura, sendo portanto necessário tomar em consideração por um lado o desejo de ganhar dinheiro, e por outro a necessidade de gastar inteligentemente o oxigénio.

O leitor pode querer fazer um programa semelhante. Além de obter assim um programa divertido, o trabalho de elaboração do programa vai contribuir para desenvolver as suas capacidades de programação. A primeira coisa a fazer

consiste em elaborar uma lista daquilo que o programa tem que fazer:

Definir as variáveis necessárias
Informar o jogador sobre o «estado da estação»
Perguntar qual a quantidade de oxigénio que tem de ser adquirida
Verificar se há disponibilidades para essa compra, se a resposta é sim comprar, se é não voltar atrás e perguntar novamente
Perguntar qual a quantidade de comida a comprar
Verificar se há disponibilidades para essa compra, se a resposta é sim comprar, se é não voltar atrás e perguntar novamente
Actualizar a quantidade de oxigénio
Actualizar a quantidade de comida
Reduzir o total de dinheiro disponível
Perguntar quantas unidades de escultura devem ser fabricadas
Verificar se há oxigénio suficiente para fabricar essa quantidade, se a resposta é não voltar atrás e perguntar novamente
Deduzir da quantidade de oxigénio a quantidade necessária para fabricar o número de esculturas especificado, aumentar o total de dinheiro disponível de modo a reflectir o valor das esculturas fabricadas
Aumentar ligeiramente o total da população, juntar um ao «ano em curso»
Verificar se há reservas de comida suficientes para alimentar toda a população
Verificar se ainda há dinheiro
Se algumas destas condições for negativa (por exemplo, não há comida suficiente) conduzir a acção para uma rotina «fim do jogo»

Se forem todas positivas, encerrar o ciclo, informando novamente o jogador sobre o estado da estação e iniciar um novo ciclo

O leitor seria provavelmente capaz de fazer um programa «Habitantes de uma Estação Lunar» com base na lista apresentada e na informação contida no esquema. Mas quero revelar-lhe um segredo que lhe permitirá resolver quase instantaneamente os problemas da programação. Pode escrever em poucos minutos todas as instruções principais do programa, o que lhe permite idealizar a execução desse programa esquemático muito antes de definir os pormenores do programa. E a partir do momento em que se elaborou este programa esquemático, pode-se desenvolver o esquema tanto quanto se queira, pois tem-se sempre — em qualquer momento do desenvolvimento do programa — um programa funcional. Não é preciso esperar que o programa esteja acabado para o executar e ver como é que funciona. O «segredo» reside em fazer um programa inteiramente constituído por uma série de chamadas de sub-rotina, incluídas por sua vez num ciclo perpétuo. Vejamos como é que isto se pode aplicar ao nosso programa. As primeiras linhas a introduzir no computador serão:

```
10 REM HABITANTES DA ESTAÇÃO LUNAR
20 GOSUB 1000: REM DEFINIR VARIÁVEIS
30 GOSUB 2000: REM IMPRESSÃO DO ESTADO
DA ESTAÇÃO
40 GOSUB 3000: REM OXIGÉNIO
50 GOSUB 4000: REM COMIDA
60 GOSUB 5000: REM ESCULTURA
70 GOSUB 6000: REM ACTUALIZAR POPULAÇÃO
80 GOSUB 7000: REM VERIFICAR ESTADO DA
ESTAÇÃO
```



```
90 IF (todas as condições positivas, desde GOSUB 7000)
THEN GOTO 30
100 REM Fim do jogo...
```

Como se pode ver, depois de se ter esquematizado desta maneira o «ciclo principal», é relativamente fácil introduzir as sub-rotinas uma por uma, testando cada uma delas antes de passar à seguinte, e aperfeiçoando-as até obter um excelente programa. A única coisa que resta fazer é elaborar uma lista das variáveis a usar no programa.

Sou de opinião que a melhor maneira de o fazer consiste em atribuir nomes explícitos às variáveis, afim de evitar as perdas de tempo no decurso da programação, por exemplo, para verificar se AA corresponde à população ou ao número de unidades de oxigénio gastas para fabricar uma unidade de escultura. Se quiser fazer programas que possam ser transferidos o mais facilmente possível para computadores diferentes, pode optar por nomes de variáveis constituídos por duas letras ou (caso o seu computador o permita) pode usar nomes compridos (tais como OXIGASTO para a quantidade de oxigénio gasto) para as variáveis. Dessa maneira não terá dúvidas sobre o significado do nome de cada variável. Para exemplificar como é que este método funciona e para ilustrar uma outra vantagem da atribuição de nomes explícitos às variáveis, apresento em seguida as variáveis usadas em Habitantes da Estação Lunar:

```
POVO — população da estação
DINHEIRO — dinheiro em caixa
COMIDA — reservas de alimentos disponíveis
CUSTCOMIDA — custo de uma unidade de comida
COMIDANEC — quantidade de unidades de comida
                consumidas por pessoa e por ano
```

CUSTOARTE — quantidade de oxigénio gasto para fabricar cada exemplar de escultura
OXI — reservas de oxigénio disponíveis
OXINEC — quantidade de unidades de oxigénio consumidas por pessoa e por ano
CUSTOXI — preço de compra de cada unidade de oxigénio
REPARAÇÃO — custo de reparação anual da estação
ANO — ano de vida da estação

Utilizando assim nomes explícitos para as variáveis — apesar de se ocupar mais espaço de memória do que quando se usam variáveis com nomes constituídos por uma ou duas letras — é mais simples interpretar o programa, e compreender para que serve cada secção do programa. Além disso, e é esta a outra vantagem a que nos referimos, é mais fácil inserir as fórmulas necessárias para os cálculos quando se redige o programa. O que significa, por exemplo, que caso se queira incluir (como o faço neste programa) uma indicação da quantidade de oxigénio necessária por ano, basta multiplicar o número de pessoas da estação (POVO) pelo número de unidades de oxigénio que cada pessoa consome por ano (OXINEC). Pode incluir-se então nas instruções de impressão do estado da estação uma linha como:

```
PRINT «HÁ»; POVO; «NA ESTAÇÃO»
PRINT «NO ANO»; ANO
PRINT «CADA PESSOA NECESSITA»; OXINEC;
      «UNIDADES DE»
PRINT «OXIGÉNIO POR ANO»; OXINEC*POVO;
      «NECESSÁRIO»
PRINT «PARA TODA A ESTAÇÃO»
```


Desta maneira é também muito fácil verificar se as compras são possíveis. Por exemplo, para comprar comida pode dizer-se:

```
PRINT «QUANTA COMIDA VAI COMPRAR?»
INPUT A
IF A*CUSTCOMIDA  DINHEIRO THEN GOTO (in
dicar outro valor de A)
```

As sugestões que lhe apresentamos no sentido de melhorar a sua programação recorrendo à «programação estruturada» podem resumir-se assim:

— faça um esquema do programa, ou da visualização gráfica definitiva;
— faça uma lista do que o programa tem de fazer em cada «ciclo principal de controle»;
— transforme esta lista numa série de instruções de sub-rotina;
— se possível use variáveis com nomes explícitos.

Quando se estão a fazer programas que se destinam a ser executados por outras pessoas convém verificar se aquilo que o jogador deve fazer quando executa o programa está bem explicado. Não interessa incluir no programa um conjunto de instruções muito longo, especialmente se a memória é limitada, mas devem anotar-se essas instruções. Além disso as instruções dirigidas ao utilizador devem ser explícitas (tais como INTRODUZA O NUMERO DE REPETIÇÕES DESEJADAS) e devem incluir-se indicações referentes aos limites a estabelecer para o input (COM QUANTAS CARTAS VAI COMEÇAR: 1, 2 OU 3?, por exemplo).

Não podemos estar presentes sempre que alguém executa um dos nossos programas e por isso temos de nos esforçar

para que o programa seja o mais possível à prova de erros. Se possível devem introduzir-se no programa rotinas de detecção de erros, para evitar que uma opção errada introduzida mais atrás no programa o não inutilize ou não produza mais adiante resultados disparatados.

Se ler várias vezes este capítulo do livro e tentar aplicar estas sugestões ao seu trabalho de programação, verificará que a qualidade desse trabalho melhora muito, o que lhe permitirá passar mais tempo a melhorar e aperfeiçoar o programa e gastar menos tempo com a tarefa puramente mecânica de elaborar um programa funcional.

GLOSSÁRIO

A

Acumulador — a parte do computador em que se efectuam os cálculos aritméticos e em que os resultados desses cálculos são armazenados.

Alfanumérico — este termo é geralmente usado em relação a um teclado, como por exemplo na frase «é um teclado alfanumérico», que significa que o teclado tem simultaneamente letras e números. É também usado em referência ao «conjunto de caracteres» do computador. O conjunto de caracteres inclui os números e as letras que o computador pode imprimir no écran.

Álgebra Booleana — um sistema algébrico criado pelo matemático George Boole, que utiliza expressões algébricas para exprimir relações lógicas (vide AND).

Algoritmo — a série de operações sucessivas que o computador executa para resolver um determinado problema.

ALU (Aritmética/Unidade Lógica) — a parte do computador que efectua operações aritméticas (tais como adição, subtracção) e onde são tomadas as decisões.

AND — uma operação de lógica booleana que o computador usa no seu processo de tomada de decisões. Baseia-se na álgebra booleana, um sistema criado pelo matemático George Boole (1815-64). Na álgebra booleana as variáveis de uma expressão representam uma operação lógica como OU e NEM.

ASCII — ou American Standard Code for Information Exchange (Código Americano Estandardizado para a Troca de Informação), o sistema de código mais geralmente usado nos teclados alfanuméricos de língua inglesa. O teclado tem 128 letras grandes e pequenas, dígitos e caracteres especiais. O ASCII converte os símbolos e as instruções de controlo em combinações binárias de sete bits.

Assemblador — um programa que converte outros programas escritos numa linguagem simbólica de baixo nível em linguagem-máquina (que o computador compreende imediatamente). Uma linguagem simbólica de baixo nível é uma linguagem de programação de baixo nível que utiliza combinações mnemónicas de duas ou três letras para representar uma determinada instrução que o assemblador converte de modo a que a máquina a possa compreender. São exemplos dessas combinações grupos de letras como ADD (somar) e SUB (subtrair). Um computador programado numa linguagem de baixo nível tem tendência para trabalhar mais depressa do que outro programado numa linguagem de alto nível como o BASIC.

B

BASIC — um acrónimo de Beginners All-Purpose Symbolic Instruction Code (Código Simbólico de Instruções para todos os fins para Principiantes). É a linguagem de computador mais usada no domínio dos microcomputadores. Apesar de ter sido criticada por muita gente, tem a vantagem de se aprender com facilidade. Um grande número de afirmações em Basic assemelham-se ao inglês vulgar.

Baud — do nome de Baudot, um pioneiro das comunicações telegráficas. O Baud é uma medida da velocidade de transmissão da informação e corresponde aproximadamente a um bit por segundo.

BCD — abreviatura de Binary Coded Decimal ou representação decimal codificada em binário.

Benchmark — um teste que permite medir certas funções do computador. Há vários «testes Benchmark standard», que servem geralmente para testar a velocidade. Este aspecto do microcomputador raramente é o que mais interessa ao possível comprador.

Binário — um sistema de numeração que usa apenas zeros e uns.

Bit — abreviatura de Binary Digit (dígito binário). É a mais pequena quantidade de informação que um circuito de computador pode reconhecer.

Bootstrap — um programa ou rotina curto que é introduzido no computador quando se liga o aparelho. Prepara o computador para receber o programa mais comprido que se segue.

Bug — um erro num programa de computador que faz com que o programa não seja correctamente executado. Se bem que designe geralmente um erro do programa, o termo de *bug* pode também ser usado para designar um defeito no *hardware* do computador.

Bus — condutores usados para transmitir sinais como instruções sobre dados ou passagem ou não de corrente no computador.

Byte — um grupo de dígitos binários que constituem uma palavra de computador. O número mais vulgar de bits de um byte são oito.

C

CAI — Computer Assisted Instruction (Instrução Assistida de Computador).

CAL — Computer Assisted Learning (Aprendizagem Assistida de Computador). O termo é geralmente usado para designar programas usados pelo utilizador no processo de aprendizagem.

Chip — o termo geral que designa todo o circuito gravado numa placa de silício. O *chip* é o coração do microcomputador.

Circuito Integrado — um circuito electrónico completo impresso numa superfície semicondutora.

COBOL — uma linguagem de alto nível cuja designação é derivada das palavras Common Business Orientated Language (Linguagem Vulgar Orientada para os Negócios) que foi concebida principalmente para a organização de ficheiros e registos.

Comparador — um dispositivo que compara duas coisas e produz um sinal relacionado com a diferença entre elas.

Compilador — um programa de computador que converte uma linguagem de programação de alto nível numa linguagem-máquina binária, que o computador pode tratar.

Complemento — um número derivado de outro segundo regras específicas.

Computador — uma máquina com três capacidades ou funções principais:
1) receber dados
2) resolver problemas
3) fornecer resultados

Computador Digital — um computador que processa informação expressa em forma descontínua.

Computador de grande porte — os computadores classificam-se geralmente em três grupos, e o grupo a que um computador pertence depende em grande medida das suas dimensões. O computador que o leitor quer comprar é um microcomputador; os computadores de dimensões médias são designados pelo nome de minicomputadores; e os computadores gigantes que se vêm por vezes nos filmes de ficção científica são computadores de grande porte. Até há quinze anos atrás os únicos computadores disponíveis na prática eram os de grande porte.

Consola — uma máquina semelhante a uma máquina de escrever que pode introduzir informação, assim como recebê-la e imprimi-la.

CPU — abreviatura de Central Processing Unit (Unidade Processadora Central). É onde reside a inteligência do computador, onde são tratados os dados e executadas as instruções.

Cursor — um carácter que aparece no écran da TV quando o computador está a funcionar. Mostra onde vai ser impresso o carácter seguinte. Há geralmente no computador «teclas de controle do cursor» que permitem ao utilizador deslocar o cursor no écran.

D

Dados — informação numa forma que o computador pode processar.

Debug — termo utilizado para designar o processo de execução de um programa para detectar e corrigir os erros que possa conter, ou seja, detectar e eliminar os *Bugs* (vide esta palavra).

Disco — é um disco de plástico magnetizado, um pouco mais pequeno do que um disco de single. É utilizado para registar programas e para a obtenção de dados. É muito mais rápido meter um disco do que uma cassette com um programa com o mesmo comprimento. Pode consultar-se muito rapidamente o disco durante a execução de um programa, para a obtenção de dados adicionais.

Disco magnético — vide Disco e Diskette.

Diskette — um tipo de disco magnético relativamente barato utilizado para registar informação para o computador e que é feito de um material flexível.

E

Editor — é o termo geralmente usado para designar a rotina do computador que permite alterar as linhas de um programa quando se está a fazê-lo.

Editor de texto — uma máquina de escrever muito inteligente que permite ao dactilógrafo manipular o texto, mudá-lo de lugar, corrigir as margens e deslocar se necessário palavras inteiras num écran antes de a informação ser enviada para o exterior por uma impressora. Os editores de texto geralmente têm memórias, podendo assim armazenar cartas estandardizadas e o texto de cartas escritas anteriormente.

EPROM — abreviatura de Erasable Programmable Read-Only Memory (Memória só para Leitura Programável e Obliterável). É semelhante ao ROM do computador, com a diferença de que é bastante fácil inserir material numa EPROM e que este não desaparece quando se desliga o aparelho. Para obliterar o que está registado numa EPROM tem de se submeter a memória à acção de uma radiação forte de ultravioletas.

F

Ficheiro — um conjunto de elementos de informação organizados de forma sistemática.

Firmware — há três espécies de «ware» nos computadores: programas «temporários» de software; hardware como a ROM, que contém informação permanente; e firmware, em que a informação é relativamente permanente, como na EPROM (vide EPROM).

Flip-Flop — um circuito que se mantém num estado eléctrico até mudar para o estado oposto por acção de um sinal de input.

Fluxograma — um diagrama desenhado antes de redigir um programa e em que as principais operações são enquadradas em rectângulos ou noutras formas geométricas e ligadas por linhas, com setas representando os ciclos e com as decisões escritas em ramificações. O fluxograma facilita muito a redacção do programa, pois permite detectar num estádio precoce erros como os ciclos infinitos ou as variáveis não definidas. Pode não valer a pena fazer um fluxograma para programas muito curtos, mas geralmente o fluxograma ajuda a criar os programas.

FORTRAN — um acrónimo de FORmula TRANslation que designa uma linguagem de alto nível usada para fins científicos e matemáticos e orientada para a resolução de problemas.

G

Gate — um circuito eléctrico que, se bem que receba um ou mais sinais exteriores, só emite um único sinal.

Gráficos — informação fornecida sob a forma de imagens, por oposição às letras e números.

H

Hard Copy — output do computador com forma permanente.

Hardware — os elementos físicos do computador (vide também software e firmware).

Hexadecimal (Hex) — um sistema de numeração de base 16. Utilizam-se os dígitos de zero a nove, assim como as letras A, B, C, D, E e F para representar números. A é igual a 10, B a 11, C a 12, e assim por diante. O sistema hexadecimal é muito usado pelos utilizadores de microprocessadores.

Hex Pad — um teclado concebido especificamente para o uso da notação hexadecimal.

I

Impressora de matriz de pontos — uma impressora que imprime quer a listagem de um programa, quer o que aparece no écran da TV. Todas as letras e caracteres são constituídos por um certo número de pontos. Quanto maior é o número de pontos por carácter, melhor é a resolução da impressora.

Input — a informação introduzida no computador através do teclado, de um microfone, de uma cassette ou de um disco.

Instrução — dados que determinam uma única operação de processamento da informação pelo computador (também designado por ordem).

Instrução de salto — uma instrução que diz ao computador para passar para outra parte do programa quando a finalidade dessa operação depende do resultado do cálculo efectuado imediatamente antes.

Interface — o limite entre o computador e um periférico como uma impressora.

Intérprete — um programa que traduz a linguagem de alto nível introduzida no computador pelo operador humano numa linguagem que a máquina compreende.

Inversor — um *gate* lógico que transforma o sinal introduzido no sinal contrário.

K

K — designação relacionada com as dimensões da memória. A memória é geralmente medida em blocos de 4K. 1K contém 1024 bytes.

L

LCD — iniciais de Liquid Crystal Diode. Alguns computadores como o TRS-80 Pocket Computer têm écrans de LCD.

LED — ou Light Emitting Diode (díodo luminoso). Os números vermelhos luminosos que aparecem nos mostradores de alguns relógios de pulso e outros são constituídos por LED.

Linguagem — as linguagens de computador dividem-se em três categorias: linguagens de alto nível como o BASIC, que se assemelham bastante ao inglês e que são de utilização bastante fácil pelo homem; linguagens de baixo nível, tais como a Assembler, que utilizam frases curtas relacionadas até certo ponto com a língua inglesa (ADD para *add* (somar) e RET para *return* (retorno), por exemplo); e linguagem máquina, que comunica mais ou menos directamente com a máquina.

Linguagem de Alto Nível — uma linguagem de programação que permite que o utilizador fale com o computador numa língua muito semelhante ao inglês. Geralmente quanto mais alto é o nível da linguagem (ou seja, quanto mais semelhante ao inglês), mais tempo é necessário ao computador para a traduzir para uma linguagem utilizável pela máquina. As linguagens de baixo nível são muito mais difíceis para os operadores humanos, mas geralmente são executadas com muito mais rapidez.

Linguagem máquina — uma linguagem operativa que o computador compreende e a que obedece directamente.

Lógica — forma matemática do estudo das relações entre acontecimentos.

Loop — ciclo. Uma sequência de instruções de um programa que é executada repetidamente até que se verifique uma dada condição.

M

Memória — há dois tipos de memória num computador. O primeiro é designado pelo nome de ROM (read-only memory, memória só para leitura) é a memória que vem já programada no computador, que diz ao computador como deve tomar decisões e executar operações aritméticas. Esta memória não é afectada quando se desliga o computador. O segundo tipo é a RAM (random acess memory, memória de acesso aleatório). Esta memória regista o programa que é dactilografado no teclado ou introduzido sob a forma de cassette ou disco. Na maioria dos computadores o computador «esquece» o que está na RAM quando é desligado.

Memória Dinâmica — uma unidade de memória incluída no computador que «esquece» o seu conteúdo quando se desliga o aparelho.

Memória estática — uma memória não-volátil que conserva a informação enquanto o aparelho está ligado, mas que não precisa de um suplemento de energia para conservar o que está na memória.

Memória não-volátil — uma memória que se não perde quando se desliga o computador. Alguns dos computadores mais pequenos, tais como o TRS-80 Pocket Computer, têm uma memória não-volátil. As pilhas conservam durante várias centenas de horas o programa introduzido.

Mensagens de erro — a informação fornecida pelo computador quando há um erro na codificação de uma parte do programa e que se exprimem geralmente através de uma paragem do computador e da impressão de uma palavra, de uma palavra e de números ou só de uma combinação de números no fundo do écran. A mensagem indica o erro cometido. São erros vulgares o uso da letra O em vez do zero numa linha, a omissão de parêntesis ou de um dos parêntesis numa expressão ou a não definição de uma variável.

Microprocessador — o coração do computador. Precisa das unidades periféricas *interface,* tais como uma fonte de corrente e órgãos de input e output, para poder trabalhar como um microcomputador.

MODEM — ou Modulator Demodulator (modulador desmodulador). É um aparelho que permite que dois computadores falem ao telefone um com o outro. Utiliza-se geralmente um suporte com um receptor telefónico.

Monitor — este termo tem dois significados em termos de computadores. Um dos significados é um écran semelhante ao de uma televisão. O monitor não possui um dispositivo de sintonização de programas televisivos, mas a imagem produzida num monitor é geralmente superior à de

uma televisão vulgar. O segundo significado de monitor relaciona-se com a ROM. O monitor de um computador é então a informação que o computador tem incorporada quando é comprado. É essa informação que lhe permite tomar decisões e efectuar cálculos aritméticos.

MPU — abreviatura de Microprocessor Unit (Unidade Microprocessadora).

N

Nanossegundo — um nanossegundo é um bilionésimo de segundo, a unidade de medição da velocidade de um computador ou de um *chip* da memória.

Not — uma operação de lógica booleana que converte um dígito binário no seu contrário.

Numérico — relativo aos números, por oposição às letras (alfabético). Muitos teclados são designados pelo termo de alfanuméricos, o que significa que apresentam simultaneamente letras e números.

O

Obliteração por ultravioletas — têm de se usar radiações de ultravioletas para obliterar o conteúdo das EPROM (vide EPROM).

Octal — um sistema de numeração de base 8 que utiliza os dígitos 0, 1, 2, 3, 4, 5, 6 e 7. O sistema octal é pouco usado actualmente no domínio dos microcomputadores. O sistema hexadecimal é mais vulgar (vide Hexadecimal).

OR — uma operação aritmética que produz um 1 se um ou mais inputs forem 1.

Oracle — um método de transmissão de mensagens em texto com um sinal emissor televisivo. É necessário um aparelho de teletexto para descodificar as mensagens. O *Oracle* é operado pelo Independent Television Service em Inglaterra, e a BBC tem um serviço semelhante — o Ceefax.

Órgão de Input/Output (I/O) — um dispositivo que recebe informação ou instruções provenientes do exterior, transmitindo-a ao computador e que, depois do processamento, a envia para o exterior numa forma adequada ao armazenamento ou numa forma que pode ser compreendida por um ser humano.

Output — ou saída. Informações ou dados fornecidos pelo computador através de dispositivos como um écran semelhante ao da televisão, uma impressora ou uma unidade de cassette. O output consiste geralmente em informação produzida pelo computador depois de ter executado um programa.

Overflow — um número demasiado grande ou demasiado pequeno para poder ser tratado pelo computador.

P

Página — termo usado com frequência para referir a quantidade de informação necessária para encher um écran de televisão, e portanto pode falar-se numa página de um programa para designar a porção de listagem que aparece ao mesmo tempo no écran.

Palavra — um grupo de caracteres ou uma série de dígitos binários que representam uma unidade de informação e ocupam um único lugar de armazenamento. O computador processa a palavra como uma única instrução.

Palavra de instrução — a palavra que desencadeia a operação comandada por uma linha do programa, que é geralmente a primeira palavra depois do número da linha. São palavras de instrução termos como STOP, PRINT e GOTO.

Palavra reservada — uma palavra que não pode ser usada como variável de um programa, pois o computador interpretá-la-á de outra maneira. Citemos como exemplo a palavra TO. Dado que TO tem um significado específico para o computador, a maioria dos computadores rejeitarão essa palavra como nome de uma variável. O mesmo se aplica a palavras como FOR, GOTO e STOP.

PASCAL — uma linguagem de alto nível.

Periférico — qualquer dispositivo que é ligado a um computador para controlo do computador, tal como uma unidade de disco, uma impressora ou um sintetizador da voz.

Port — ficha de ligação dos periféricos ao computador.

Prestel — o nome inglês de um sistema de telecomunicação em que se pedem por via telefónica páginas de informação a um computador central, recebendo-as num écran de televisão. Existe nos Estados Unidos numa versão comercial semelhante, a que se dá o nome de The Source.

Programa — em termos de computadores a palavra programa significa a lista de instruções que se introduz num computador.

PROM — iniciais de Programmable Read Only Memory (memória programável só para leitura). É um dispositivo que pode ser programado e que depois de ter sido programado conserva permanentemente o programa (vide também EPROM e ROM).

Q

Quadro principal — uma estrutura de base a que se podem acrescentar circuitos extra. Estes circuitos extra conferem frequentemente ao computador capacidades que não foram programadas pelo fabricante, tais como a produção de som ou o controlo da *light pen*.

R

Random Acess Memory (RAM) — uma memória do computador que pode ser alterada à vontade pela pessoa que está a utilizar o computador. O conteúdo da RAM perde-se geralmente quando se desliga o computador. A RAM é o dispositivo da memória que armazena o programa dactilografado no teclado e que armazena também os resultados dos cálculos em curso.

Read-Only Memory (ROM) — ao contrário do que acontece com a RAM, a informação armazenada na ROM não pode ser alterada pelo utilizador do computador e a informação não se perde quando se desliga o computador. Os dados armazenados na ROM foram aí introduzidos pelo fabricante e dizem ao computador como é que deve tomar decisões e efectuar cálculos aritméticos. As dimensões da ROM e da RAM são indicadas em unidades K (vide K).

Recorrência — repetição contínua de uma parte do programa.

Registo — um local específico da memória em que são armazenadas uma ou mais palavras de computador durante o processamento.

Relógio — o dispositivo de medição do tempo do computador que sincroniza as operações efectuadas pela máquina.

Rotina — esta palavra pode ser usada como sinónimo de programa, ou pode designar também uma parte específica de um programa (vide também Subrotina).

Rotina Interactiva — parte de um programa que é repetida indefinidamente até que uma dada condição seja preenchida.

S

Segunda Geração — esta expressão tem dois significados. O primeiro aplica-se aos computadores que utilizam transistores, por oposição aos computadores da primeira geração, que utilizavam válvulas. Segunda geração pode significar também uma segunda cópia de um determinado programa; as gerações seguintes são de qualidade progressivamente inferior, apresentando cada vez mais ruídos.

Semicondutor — um material que é geralmente um isolador eléctrico mas que em condições específicas pode tornar-se num condutor.

Sequencial — informação que é armazenada ou introduzida numa sequência, um bit de cada vez.

Sinal — um impulso eléctrico que transmite dados.

Silicon Valley (Vale do Silício) — nome por que é vulgarmente conhecida uma região da Califórnia onde se situam muitas fábricas de semicondutores.

Sistema de exploração — o software ou firmware fornecido geralmente com a máquina e que permite executar outros programas.

SNOBOL — uma linguagem de alto nível.

Software — o programa que é introduzido no computador por um utilizador e que diz ao computador o que a máquina deve fazer.

Software compatível — expressão aplicável a dois computadores diferentes que aceitam programas escritos para o outro.

String nulo — uma variável *string* que não contém caracteres. É indicado no programa por aspas duplas com um espaço vazio no meio.

Subrotina — parte de um programa que geralmente se repete muitas vezes durante a execução do programa principal. Uma subrotina acaba com a instrução de voltar à linha seguinte àquela que remeteu para subrotina.

T

Teletexto — informação transmitida na faixa superior de uma imagem televisiva. É necessário um aparelho especial para a descodificar e imprimir no resto do écran a informação contida no texto. O serviço correspondente da BBC é designado pelo nome de Ceefax, e o da ITV pelo de Oracle. As mensagens em teletexto podem ser também transmitidas por cabo, como é por exemplo o caso no serviço Prestel em Inglaterra ou no The Source nos Estados Unidos.

Terminal — uma unidade independente da unidade central de processamento. Consiste geralmente num teclado e num écran de raios catódicos.

Time Sharing — um processo pelo qual um certo número de utilizadores têm acesso a um computador grande, que estabelece uma ligação sucessiva muito rápida com um utilizador de cada vez, de tal modo que cada um dos utilizadores tem a impressão de que é a única pessoa que está a usar o computador no momento.

Tabela de verdade — uma tabela matemática que fornece uma lista de todos os resultados possíveis de uma operação de lógica booleana, mostrando os resultados obtidos a partir de diferentes combinações de outputs.

U

UHF — Ultra High Frequency (frequência ultra alta) (300-3000 mega-Hertz).

V

Variável — uma letra ou combinação de letras e símbolos que o computador pode relacionar com um valor ou com uma palavra durante a execução de um programa.

VDU — abreviatura de Visual Display Unit ou Unidade de Visualização Gráfica.

Visualização gráfica — o output visual do computador, que aparece geralmente num écran de TV ou de um monitor.

Volátil — este termo designa uma memória que «esquece» o seu conteúdo quando se desliga o computador.

BIBLIOGRAFIA

Compilada por Tim Hartnell

The A to Z Book of Computer Games (McIntire, Thomas C, Tab Books, Blue Ridge Summit, Pa.).

Um excelente livro Tab que lhe sugere ideias para programas e programas prontos para correrem, apesar de alguns dos jogos serem decepcionantes, como o programa Othello, demasiado comprido e que nem sequer é um jogo, mas apenas o registo das jogadas de dois jogadores humanos. Mas outros jogos, como Fivecard e Hotshot, estão bem feitos e vale a pena experimentá-los no seu microcomputador.

Basic Computer Games (ed. Ahl, David, Creative Computing Press, Morristown, New Jersey).

É uma obra clássica, que contém mais sugestões de programas do que qualquer outro livro de jogos de computador jamais publicado. Jantei uma vez com David Ahl em Londres, depois de um programa PCW, e falei do livro com ele. Disse-me que trabalhava já com computadores pessoais mesmo antes de esses computadores terem sido lançados comercialmente e que apesar de muitos dos jogos publicados no livro não parecerem agora nada de especial, o facto de se poderem fazer e executar programas de jogos de interacção com o computador parecia incrível no fim da década de setenta. O programa de xadrez e Life for Two são dois dos melhores dos que pode encontrar neste excelente livro de programas e sugestões para programas.

BASIC Computer Programs for the Home (Sternberg, Charles D, Hayden Book Company, Inc., Rochelle Park, New Jersey).

Os computadores «domésticos» começam sempre por ser utilizados para jogar jogos (quando são comprados). Uma das razões pelas quais não são usados para finalidades mais sérias é devido à falta de um conjunto de

programas práticos e funcionais que o utilizador possa compreender e aplicar com facilidade para finalidades domésticas práticas. Este livro contém uma série de programas de rentabilização do seu computador. Os programas têm as mais variadas aplicações práticas; e foram concebidos de modo a não ser necessária a utilização de unidades de gravação ou de disco. Os programas são muito variados, e estão agrupados por temas: programas de contabilidade doméstica (incluindo as despesas da casa e registos de impostos); programas referentes ao automóvel (incluindo o consumo de combustível e o planeamento de viagens); programas de cozinha (incluindo programas de planeamento de uma alimentação normal ou de dieta); programas-calendários domésticos (incluindo um calendário de coisas a fazer e alguns programas que devem ter sido concebidos para evitar as discussões acerca dos programas de televisão a ver); e finalmente «Listas para todos os fins» (incluindo cartões de Boas-Festas, colecções de discos e três versões de um programa de moradas).

The BASIC Handbook (Lien, David A, Compusoft Publishing, San Diego, Califórnia).

Este livro é uma enciclopédia da linguagem BASIC. Dado que o BASIC é tão usado actualmente no mundo dos microcomputadores, é necessário explicar os seus numerosos dialectos, de modo a poder transportar os programas para computadores diferentes. É muito desagradável encontrar o programa exacto que se procurava só para verificar que é impossível corrê-lo no nosso computador. Este livro trata esse problema, examinando em pormenor todas as instruções, funções, operandos e comandos usados vulgarmente no BASIC. A maioria das palavras em BASIC tem o mesmo significado para todos os computadores que a aceitam. Se um computador não tem as capacidades correspondentes a uma palavra necessária ou específica, geralmente há maneiras de obter a mesma função usando uma outra palavra ou combinação de palavras. Se bem que o manual exija uma certa aplicação para assimilar utilmente a informação que contém, é uma obra de referência extremamente útil. Encontrará provavelmente no livro todas as palavras de BASIC que conhece (e muitas que não conhece, tais como LE, NE, GOTO-OF, RES e TIME). O manual pode ter pouca utilidade para si nos primeiros tempos em que lida com o computador, mas quando aprofundar os seus conhecimentos ser-lhe-á indispensável.

Beat the Odds, Microcomputer Simulations of Casino Games (Sagan, Gans, Hayden Book Company, Inc., Rochelle Park, New Jersey).

O livro explica como é que se jogam alguns jogos de azar (trente-et-quarante, roleta, chemin-de-fer, *craps* e *blackjack*) apresentando listagens

completas de programas em BASIC, acompanhadas de comentários sobre os sistemas e as estratégias óptimas. O Professor Sagan (professor de matemática na North Carolina State University) diz-nos que escreveu o livro para convencer as pessoas de que a longo prazo nunca podiam ganhar — a não ser talvez no *blackjack* — e para explicar alguns sistemas conhecidos e os seus defeitos, mas principalmente para apresentar simulações muito realistas dos próprios jogos. Essa tentativa foi muito bem sucedida. As listagens são talvez mais compridas do que outras versões computorizadas dos mesmos jogos, mas as versões de Sagan reproduzem exactamente as probabilidades do jogo «na vida real» e abrangem todas as possibilidades que se podem verificar num jogo real. Os programas estão bem estruturados e um estudo das listagens pode sugerir-lhe ideias para melhorar os seus próprios programas.

Beginner's Guide to Chess (Keene, Raymond, Pelham Books Ltd, London).

Um guia ideal das técnicas simples do jogo do xadrez, que podem ser transpostas em algoritmos se você quiser fazer o seu próprio programa de xadrez.

The Calculator Book for Kids of All Ages (Hartman, Arlene, Signet Books, New York).

O título do livro já diz tudo, e os nomes dos jogos (por exemplo, Fibonacci Follies, Stretch to Sixty e Cating Out 9s) ilustram o conteúdo do livro. Há algumas adivinhas inteligentes e cerca de 15 ideias que estão mesmo a pedir para serem transformadas em jogos de computador.

33 Challenging Computer Games for TRS 80-Apple/PET (Chance, David, Tab Books, Blue Ridge Summit, Pa.).

Mesmo que não tenha nenhum dos três computadores citados no título deste livro, encontrará nele excelentes ideias para o seu próprio trabalho de programação e muitos dos jogos podem correr com alterações mínimas em qualquer computador que use o BASIC. Recomendamos em especial os programas Life Support, Scrambled Eggs e Tank Assault.

Communicating with Microcomputers (Witten, Ian H., Academic Press, London).

O livro constitui uma introdução à tecnologia da comunicação homem/computador, dirigindo-se ao não especialista. O livro insiste especialmente nas técnicas de baixo custo associadas aos pequenos sistemas e aos computadores pessoais, chamando assim a atenção do leitor para a natureza positiva da «revolução do microprocessador» — para a maneira como as máquinas podem ajudar as pessoas — mais do que para os aspectos ne-

gativos que são frequentemente referidos na imprensa não-especializada. O nível do livro está indicado para o leitor que sabe qualquer coisa de electrónica. A parte final, relativa à comunicação falada, constitui uma leitura fascinante.

Computer Appreciation (Fry, T. F., Newnes, Butteworths, 1975).

Um manual bastante simples e útil sobre o funcionamento do computador e as suas aplicações para fins administrativos. O livro, que foi concebido como um manual para um curso comercial, trata uma grande variedade de tópicos, que vão de uma história resumida da evolução dos computadores até ao estudo do hardware para computador e da programação, incluindo o dos métodos de organização de um departamento moderno de processamento de dados. Termina com um comentário breve sobre as aplicações práticas dos computadores e a influência dos computadores nas técnicas de gestão. O manual mantém a sua actualidade, apesar de a variedade e as capacidades do hardware terem evoluído muito depois do livro ter sido escrito.

The Computer Book: An Introduction to Computers and Computing (Bradbeer, Robin; De Bono, Peter; Laurie, Peter; BBC Publications).

Este livro foi publicado em associação com a série televisiva da BBC «The Computer Programme», que começou a ser transmitida pela BBC2 em Janeiro de 1982 e que foi produzida por Paul Kriwaczek. Tive ocasião de falar do livro com Robin Bradbeer quando ele o estava a escrever, e o autor disse-me que os editores da BBC estavam sempre a chamar-lhe a atenção para as palavras em «gíria de computadores», que deviam ser eliminadas. Insistiam em que todos os termos deviam ser bem explicados, disse-me Robin. Em consequência dessa insistência o livro é acessível a toda a gente. Parte-se do princípio de que tudo tem de ser explicado, incluindo por exemplo o uso de uma tecla *shift* — ou o efeito desse uso — de um teclado. O texto é ilustrado com numerosas fotografias e ilustrações e constitui uma introdução pormenorizada aos computadores, principalmente aos micro, e às suas aplicações possíveis.

Computer Games for Businesses, Schools and Homes (Nahigian, J. Victor and Hodges, William S. Winthrop Publishers Inc., Cambridge, Mass.).

Alguns programas não estão à altura das aspirações e do preço do livro, mas vale apena adaptar os melhores para correr no seu computador. A inclusão de esquemas simples exemplificativos do que o programa faz permite-lhe saber exactamente em que consiste o programa antes de o correr. Os programas Tennis e Star Trek são particularmente bons.

Dice Games Old and New (Tredd, William E., Oleander Press, Cambridge).

As explicações claras destes jogos permitir-lhe-ão inventar programas de jogos para o seu microcomputador que o vão ocupar durante muito tempo.

The Electronic Calculator in Business, Home and School (Birtwistle, Claude, Elliot Right Way Books, Kingswood, Surrey).

Se quiser tirar o melhor partido possível da sua calculadora tem de compreender os princípios matemáticos que estão na base do seu funcionamento. É este o objectivo do livro, que de uma maneira geral é atingido. Os princípios matemáticos explicados são no entanto bastante simples e básicos, na medida em que o livro se dirige a um público variado — alunos em idade escolar, estudantes universitários, empregados comerciais e donas de casa. É um livro prático, que deve ser lido e estudado com uma calculadora à mão.

Everyman's Indoor Games (Brandreth, Gyles, J. M. Dent and Sons Ltd., London).

Se está interessado em jogos que podem ser convertidos em programas de computadores, salte os capítulos intitulados Parlour Games e Children's Party Games e concentre-se no resto do livro, que constitui um verdadeiro tesouro de jogos que podem ser usados como ponto de partida para programas de computador. Fox and Geese, Poker Dice e Billiards são alguns dos programas que pode fazer com base na descrição destes jogos tal como consta do livro.

Games and Puzzles for Addicts (Millington, Roger, M. and J. Hobbs, Walton-on-Thames).

Estes jogos e adivinhas foram publicados inicialmente na revista semanal de computadores «Datalink», e portanto agradarão muito especialmente aos entusiastas dos computadores. Há muitas ideias que podem ser adaptadas com facilidade em jogos para o computador.

Games for Home, Travel and Parties (Jensen, Helen, Western Publishing Company Inc., Racine, Wisconsin).

Este livro, que se dirige explicitamente às crianças, explica alguns jogos que podem ser facilmente transformados em programas (por exemplo, Snakes, Lift-Off e Fish), incluindo ainda um capítulo que ensina a jogar xadrez.

Home Computers, Questions and Answers, Hardware (Didday, Rich, dilithium Press).

Este livro tem dois objectivos principais. Em primeiro lugar pretende dar aos leitores uma ideia bastante exacta do que é um computador pessoal, para lhes permitir tomar decisões racionais antes de comprar esse material. Em segundo lugar propõe-se fornecer às pessoas que não têm conhecimentos especializados de computadores uma informação geral sobre o assunto, e principalmente sobre os microcomputadores. O livro consegue dar a informação suficiente para que o leitor fique em condições de compreender facilmente os artigos sobre projectos avançados publicados em revistas de computadores para amadores e a publicidade do equipamento computorizado pessoal e para que possa compreender também o que dizem outras pessoas que têm um conhecimento mais especializado de computadores.

Inside BASIC Games (Mateosian, Richard, Sybex).

Este livro é um guia, se bem que talvez um tanto ou quanto verboso, para todos os que querem perceber os jogos de computador. Ensina a fazer programas interactivos em BASIC e como é que se aplicam aos computadores pequenos os princípios do desenvolvimento de sistemas. O livro aborda também as maneiras como as características dos sistemas específicos dos pequenos computadores são tratadas em BASIC. Os capítulos do livro incluem: Arithmetic Games, Guessing Games, Time Games, Date Games, Taxman e ainda a programação em «Free BASIC», um BASIC estruturado que é traduzido manualmente nas instruções do BASIC para ser introduzido no computador. O «Free BASIC» não é uma linguagem; é um método de descrição dos programas (como um fluxograma) sem números de linhas, que utiliza nomes simbólicos para as subrotinas. Outros capítulos são dedicados a jogos como The Match-Up Game, Craps e Alien Life. Apesar da sua verbosidade o livro é uma obra útil.

An Introduction to Personal and Business Computing (Zaks, Rodnay, Sybex).

Almocei com Rodnay em Londres por ocasião de um programa PCW e ele disse-me que achava que as actuais previsões americanas sobre o crescimento no campo do computador pessoal eram muito pessimistas. Observou que as previsões de 1978, época em que escreveu o seu livro, provaram ser tão inexactas que os aspirantes a profetas deviam aprender essa lição e partir do princípio de que as suas previsões teriam uma margem de erro equivalente à multiplicação por um factor 10 ou 100. Apesar de não ser já muito recente — e os livros sobre computadores desactualizam-se muito rapidamente — este livro é uma boa introdução aos computadores, explicando em inglês claro e conciso as bases do funcionamento dos computadores. O Dr. Zaks apresenta também algumas sugestões sobre o que se deve procurar quando se vai comprar um computador.

Microsoft BASIC (Knecht, Ken, dilithium Press, Forest Grove, Oregon).

Este livro constitui uma introdução muito completa à programação em BASIC Microsoft. Os conceitos explicados são ilustrados em programas curtos e funcionais. Começando pelas instruções mais simples e mais frequentemente usadas e avançando até às instruções complexas do BASIC, o Sr. Knecht mostra como as versões mais potentes da linguagem podem poupar muito tempo e esforço na programação.

The Personal Computer Book (Bradbeer, Robin, Input Two-Nine).

O título deste livro já diz tudo. Robin é um conhecido especialista inglês de microcomputadores. Foi o fundador das London Polytechnic Computer Fairs (feiras de computadores), foi consultor do programa televisivo de microcomputadores da BBC (e um dos autores de *The Computer Book*, publicado pela BBC) e era o editor da publicação mensal *Educational Computing*. O livro tem pois bases sólidas. Explica o que é um computador e como funciona; explica os mistérios encerrados nas «caixas pretas» que constituem o computador; e inclui ainda alguns apêndices úteis sobre margens de erro, fabricantes e distribuidores, revistas, uma bibliografia seleccionada (compilada por Richard Ross-Langley, de *Mine of Information*) e um glossário. Mas a parte mais interessante e útil do livro é talvez a descrição pormenorizada da maior parte dos sistemas dos computadores comercializados no mercado inglês, respectivo preço e capacidades. Trata-se de um livro de referência muito útil.

Personal Computers: What They Are and How to Use Them (Wels, Byron G., Trafalgar House Publishing).

O mundo dos computadores evoluiu muito depois de 1978, data da redacção deste livro, mas apesar disso a obra mantém grande parte da sua utilidade e interesse. O livro descreve alguns dos computadores pessoais à venda no mercado e os melhoramentos que sofrerão provavelmente no futuro. Explica em linguagem acessível aos leigos o funcionamento do computador e como trabalhar com ele. Inclui também capítulos sobre a construção e manutenção de pequenos sistemas computorizados.

Play the Game (Love, Brian, Michael Joseph and Ebury Press, London).

É um excelente livro, contendo cerca de 40 reproduções de jogos de tabuleiro vitorianos (e pré-vitorianos) em tamanho natural, sendo muitos de-

les adequados para serem jogados com o computador. Podem inclusive usar-se os tabuleiros do livro (o computador indicará então onde se localiza a jogada no tabuleiro exterior), o que evita que se tenha de introduzir no programa uma rotina para desenhar um tabuleiro.

The Pocket Calculator Games Book (Schlossberg, Edwin and Brockman, John, Wilton House Publications Ltd., London).

O livro contém muitas ideias de jogos que podem ser adaptados em jogos de computador.

57 Practical Programs and Games in BASIC (Tracton, Ken, Tab Books, Blue Ridge Summit, Pa.).

Este livro contém outros programas mais sérios além dos jogos (do tipo Chi-Square Evaluation e Fibonacci Numbers). Os programas são bem feitos e completados com documentação adequada (embora breve) e fluxogramas. Os programas Guerra no Espaço (versões um e dois) do fim do livro são particularmente bons.

Problems for Computer Solution (Rogowski, Stephen J, Creative Computing Press, Morristown, New Jersey).

O livro apresenta 50 problemas simples (e alguns menos simples) que podem ser resolvidos por meio de um programa. Foi publicado em duas versões, uma para os professores e outra para os alunos: a versão do professor contém o programa sugerido para a solução do problema e o respectivo esquema. O livro constitui um excelente instrumento de ensino.

Stimulating Simulations (Engel, C. W., Hayden Book Company, Inc., New Jersey).

Segundo se diz na capa, o livro contém «12 programas originais em BASIC para o amador de computadores». Alguns desses programas são efectivamente fascinantes, Forest Fire, Rare Birds e The Devil's Dungeon são três programas de jogos muito divertidos, e Diamond Thief (ladrão de diamantes, em que o computador decide quem cometeu o crime e desafia depois o outro jogador a descobrir qual dos suspeitos é o culpado) é um excelente programa, funcional e conciso.

TAKE TWO! 32 Board Games for 2 Players (Tapson, Frank, A & C Black, London).

Este livro dirige-se às crianças, mas sugere muitas ideias fascinantes que podem ser facilmente adaptadas em jogos de computador (apesar de algumas serem repetitivas).

24 Tested, Ready-to-Run Game Programs in BASIC (Tracton, Ken, Tab Books, Blue Ridge Summit, Pa.).

Tab Books é uma editora prolífica no campo dos programas para microcomputadores e os seus livros alcançam sempre e justificadamente grande sucesso junto do público. A leitura de um livro como este sugerir-lhe-á ideias para uma boa estruturação dos programas e para os escrever de modo a garantir uma compatibilidade máxima entre diferentes versões de BASIC. Muitos dos jogos, tais como Auto Rally e Capture the Alien, apesar dos seus nomes pouco imaginativos, são programas bem planeados e estruturados.

1001 Things to Do with Your Personal Computer (Sawush, Mark, Tab Books, Blue Ridge Summit, Pa.).

Comprei este livro numa feira de computadores em Atlanta, e li-o no avião, na viagem de regresso a Londres (tomando notas e marcando os cantos das páginas). Quando cheguei ainda o não tinha acabado de ler. Se está convencido de que esgotou todas as aplicações possíveis do seu computador, compre este livro e verifique como está enganado. O livro ensina-o a utilizar o computador para escrever músicas e histórias, para facilitar trabalhos de mecânica ou de carpintaria, para resolver equações simultâneas, em astrologia, e para muitas outras finalidades ainda.

The World Computer Chess Championship (Hayes, Jean E., e Levy, David N. L., Edinburgh University Press, Edinburgh).

É uma descrição fascinante do primeiro campeonato mundial de xadrez máquina contra máquina, que teve lugar em 1974, quando os cerca de doze programas de computador que participaram no campeonato eram ainda os únicos programas de xadrez existentes. Os jogos são analisados em pormenor e a parte final do livro descreve um sistema de numeração do tabuleiro que pode usar se quiser fazer o seu próprio programa de xadrez. O livro evidencia bem o progresso que se verificou em poucos anos no mundo dos computadores.