

```

U 0,36: DRAW -36,0: DRAW 0,16: D
RAW 36,0: DRAW 0,24
9915 PRINT AT 5,8; INK 7; BRIGHT
1;"R I P"
9920 PLOT 76,87: DRAW -40,-24: P
LOT 96,87: DRAW 40,-24
9925 GO SUB 9800: PRINT AT 16,1;
"POUCA SORTE! VOCE PONTUOU ";ST+
DEX+SKI+EXP
9930 PRINT AT 18,3;"OUTRA AVENTU
RA ?(S OU N)"
9935 IF INKEY#="N" THEN NEW
9940 IF INKEY#="S" THEN RUN
9950 GO TO 9935

```

4

LABIRINTO

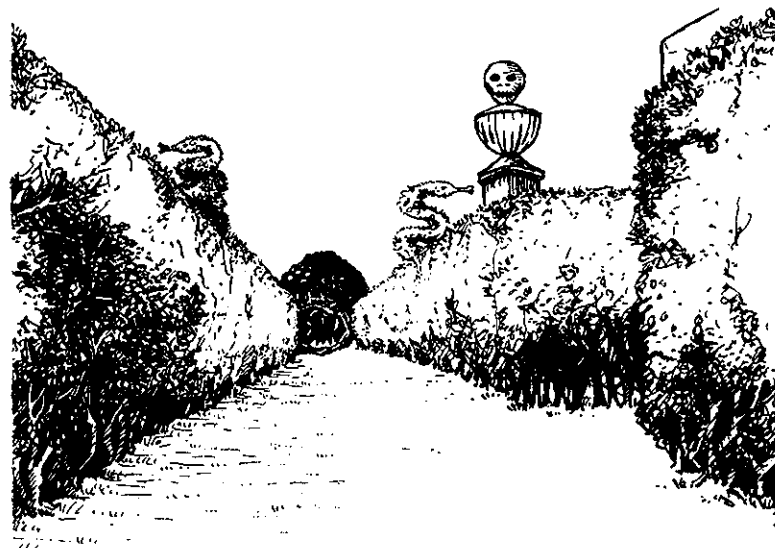


Fig. 6

Argumento

Tendo tentado assassinar o Senhor Negro, você é sentenciado a um destino pior do que a morte. É conduzido aos mais longínquos recessos da sua montanha e fechado numas antigas masmorras, onde nem sequer entram os Espectros do Senhor Negro, que vivem perto delas.

Segundo uma inscrição gravada na porta rochosa, existe apenas uma saída; você deve procurá-la e aventurar-se para além dela a fim de descobrir o que se encontra depois. As arrepiantes passagens sucedem-se umas às outras indefinidamente – conseguirá alguma vez escapar?

Sugestões para a introdução do programa

Neste programa não existem gráficos definidos pelo utilizador dado que são usadas as instruções de alta resolução PLOT e DRAW. A parte principal do jogo foi escrita inteiramente em código-máquina, o que permite obter um jogo muito mais rápido. O código-máquina apresenta-se como uma série de números, que devem ser introduzidos nos endereços mais elevados em memória, usando o programa carregador.

Antes do mais, é necessário colocar o teclado em modo maiúsculas, usando CAPS LOCK, e escrever o programa Basic LABIRINTO. É necessário ter um cuidado especial ao escrever o número que se segue a USR, dado que um número incorrecto provocará um «crash» do sistema. Depois de introduzido na máquina, o programa será gravado em cassete usando a instrução:

SAVE "LABIRINTO" LINE 10.

Este programa deve ser gravado e verificado em seguida, usando a instrução VERIFY". Deve, seguidamente, deixar a fita na mesma posição, a fim de poder gravar a parte de dados que se seguem.

Escreva em seguida o programa carregador, verificando, uma vez mais, se passou ao modo «maiúsculas». Depois de estar terminado, deve pô-lo em execução. Deve aparecer no visor um endereço, sendo-lhe pedido que indique um byte, que é um dos números da Lista Decimal que encontrará no final do capítulo. Deve escrever todos os números desta lista pela ordem em que são indicados, um de cada vez, sendo impresso no final de cada linha um endereço, tal como na lista. No final de cada secção, o computador cumpre uma

rotina para procura de erros. Se descobre algum, permite-lhe verificar os números que escreveu. A secção é impressa linha a linha e ENTER permite-lhe continuar, enquanto C permite corrigir uma linha. O processo continuará até o computador deixar de encontrar erros. Será necessário ter cuidado ao introduzir os números, porque o computador não é capaz de encontrar todos os erros e bastará um só, para impedir a execução do programa.

O leitor terá depois de gravar em cassete os dados, em quatro partes, devendo fazê-lo no ponto da fita imediatamente a seguir ao programa Basic LABIRINTO. Quando a fita estiver pronta, carregue em ENTER para gravar o primeiro bloco de dados. Será necessário repetir este procedimento três vezes, depois do que todas as secções devem ser verificadas.

O jogo fica assim completo. Escreva:

LOAD "LABIRINTO"

e todas as secções serão carregadas automaticamente.

Técnicas

Num computador, pode-se pensar na memória como sendo formada por uma comprida linha de caixas, cada uma delas com um endereço próprio, e capaz de conter um número entre 0 e 255.

Só se pode alterar o conteúdo destas caixas, quando estão situadas na memória de acesso aleatório (RAM). Consegue-se fazê-lo, usando a instrução POKE. Por exemplo:

POKE 60000, 100

colocará o número 100 na caixa com o endereço 60000.

No Spectrum, o primeiro endereço em RAM é 16384 porque a memória ROM utiliza os primeiros endereços, entre 0 e 16383.

A fim de descobrir o que se encontra em qualquer destas «caixas», incluindo as da ROM, utiliza-se a instrução PEEK. Assim, PEEK seguida de um número produz o conteúdo da

caixa requerida, podendo este valor ser usado de qualquer modo que se pretenda. Por exemplo:

PRINT PEEK 60000

colocará no visor o número que está na caixa 60000.

A ROM contém o programa em código-máquina que o computador utiliza constantemente, a menos que tenha sido obrigado a executar rotinas em código-máquina escritas em RAM pelo utilizador.

Sem aprofundarmos mais estas questões relacionadas com a linguagem-máquina, diremos que esta misteriosa linguagem é uma série de instruções simples que podem ser executadas imediatamente pelo processador. Como cada instrução é muito simples, pode ser bastante difícil programar uma tarefa complexa, que seria fácil de programar em Basic.

Utiliza-se a função USR para executar rotinas em código-máquina. O que acontece de facto quando se utiliza esta instrução, é o seguinte:

- 1) O número que se segue a USR é copiado para o registo BC (um tipo de variável em código-máquina).
- 2) O comando é transferido para o endereço especificado a seguir a USR.
- 3) Ao encontrar uma instrução RETURN no final da rotina em código-máquina, o valor que se encontra no registo BC será atribuído à função USR permitindo o seu uso em Basic.

Por exemplo:

PRINT USR 60000

Ao sair da rotina em código-máquina que se inicia no endereço 60000, o conteúdo do registo BC surgirá no visor.

Instruções de jogo

Ao iniciar o jogo, veremos uma representação tridimensional do terreno à nossa frente. Podemos ver sete unidades de terreno, a menos que exista uma parede mais perto.

O movimento pelo labirinto é realizado usando o joystick da direita no Interface 2 ou as teclas indicadas na Tabela 4.

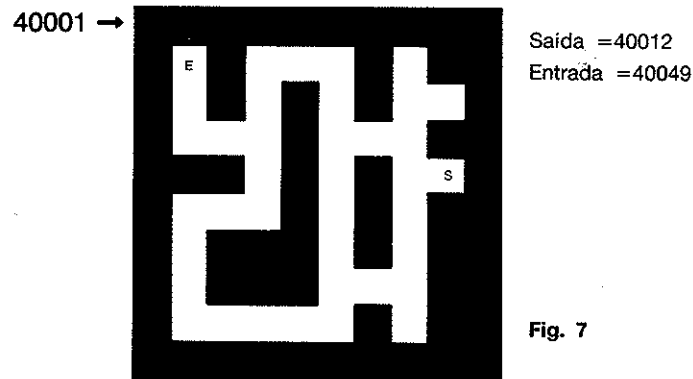
TABELA 4

Comando	Direcção
8	Em frente
6	90° à esquerda
7	90° à direita

Nota: Para parar o jogo, carregue em CAPS SHIFT 8 e SPACE ao mesmo tempo.

Adaptação

O jogo foi concebido de tal modo que possa ser alterado pelo utilizador. No entanto, deve ter-se bastante cuidado no sentido de garantir que as rotinas em código-máquina funcionem correctamente.



Em primeiro lugar, desenhamos um labirinto semelhante ao apresentado na figura 7, tendo o cuidado de respeitar as regras seguintes:

- 1) O labirinto deve ser quadrado.
- 2) O labirinto deve ser rodeado por paredes.
- 3) Os corredores devem ter todos uma largura unitária.
- 4) Não pode ser esquecida uma posição de partida (S).
- 5) Deve-se obviamente incluir uma posição de saída (E).

Depois de o labirinto ter sido desenhado, devem definir-se as variáveis do código-máquina, do seguinte modo:

```
POKE 60141,x+2
POKE 60165,x+2
POKE 60175,x
POKE 60189,x
POKE 60202,(2*x)-1
POKE 60212,x
POKE 60226,x
POKE 60239,(2*x)-1
POKE 60252,x
POKE 60260,x
```

onde x é a dimensão do labirinto.

Pode-se agora introduzir o labirinto em memória começando na posição 41000 e introduzindo a informação linha a linha, usando os códigos indicados na Tabela 5.

TABELA 5

Código	Elemento
0	Um espaço
1	Uma parede
2	Saída

Quando o labirinto tiver sido completamente introduzido, as linhas do programa relacionadas com o endereço de partida devem ser alteradas do seguinte modo:

```
50 POKE 40006, INICIO-256*INT(START/256)
60 POKE 40007, INT(INICIO/256)
```

e a saída deve ser alterada usando:

```
POKE 60286,INT(SAÍDA/256)
POKE 60292,SAÍDA-256*INT(SAÍDA/256)
```

Quando tudo isto estiver feito, o jogo estará terminado e poderá ser gravado, se assim se quiser, usando as instruções:

```
SAVE "CODIGO1" CODE 40500,19
SAVE "CODIGO2" CODE 41000,900
SAVE "CODIGO3" CODE 51000,5162
SAVE "CODIGO4" CODE 60000,532
```

LISTAGEM

IMPORTANTE	
Toda a informação deve ser introduzida usando	
CAPS LOCK	

Labirinto

```
10 LOAD ""CODE
20 LOAD ""CODE
30 LOAD ""CODE
40 LOAD ""CODE
50 POKE 40000,71
60 POKE 40001,160
70 POKE 40002,1
80 CLS
```

```

90 GO TO 100
100 CLS: FOR C=1 TO LEN A#
110 PRINT AT 10,15-(LEN A#/2)+C
FLASH 1: INVERSE C-C*INT (C/2)
A#(C):
120 NEXT C
130 PAUSE 200
140 RETURN
150 LET A#="SO MAZE": INK 1: PA
PAPER 7: GO SUB 100
160 LET A#="LIBERTE-SE DO LABIR
INTO OU": INK 2: PAPER 8: GO SUB
100
170 LET A#="SOFR A MISERIA DE
ERRAR": INK 1: PAPER 7: GO SUB 1
00
180 LET A#="NOS SEUS CORREDORES
VAZIOS": INK 2: PAPER 8: GO SUB
100
190 LET A#="SOA SORTE": INK 5:
PAPER 1: GO SUB 100
200 LET A#="QUALQUER TECLA PARA
COMEÇAR": INK 5: PAPER 2: GO SU
B 100
210 PAUSE 0
220 INK 0: PAPER 7: CLS
230 RANDOMIZE USR 00000
240 CLS
250 LET A#="PARABENS!": INK 2:
PAPER 8: GO SUB 100
260 FOR N=0 TO 99 STEP .5
270 BEEP .005,N
280 NEXT N
290 LET A#="VOCE CONSEGUIU": IN
K 1: PAPER 7: GO SUB 100

```

Carregador

```

10 REM *****
20 REM *FOLDER DEMONIAL*
30 REM *****
40 DATA 400000,400010,4700
50 DATA 410000,410000,4400
60 DATA 420000,411001,4004
70 DATA 430000,430101,4040
80 DATA 440000,441001,40411
90 DATA 450000,441001,40300
100 DATA 460000,461001,07000
110 DATA 000000,000001,00040

```

```

100 RESTORE 40
110 READ A,B,C
140 LET D=0
145 CLS
150 PRINT "ORG: ";A
160 PRINT
170 FOR N=A TO B STEP 6
180 PRINT N;";";
190 FOR M=0 TO 5
200 PRINT TAB M*4+7;
210 INPUT "BYTE=";Z
220 POKE N+M,Z
230 PRINT Z;
240 NEXT M
250 PRINT
260 NEXT N
270 PRINT
280 IF C<>D THEN PRINT FLASH 1:
"ERRO NA SECCAO DE CODIGO": PAUS
E 2000: GO TO 1000
2900 PRINT
300 PRINT
310 IF A<00000 THEN GO TO 100
320 CLS
330 PRINT "INTRODUZA A CASSETE."
340 PRINT "O CODIGO E GRAVADO E
M 4 PARTES, CARRIQUEM ENTER AN
TES DE GRAVAR CADA PROGRAMA
"
350 SAVE "CODE1"CODE 40000,10
360 SAVE "CODE2"CODE 41000,000
370 SAVE "CODE3"CODE 51000,9162
380 SAVE "CODE4"CODE 00000,532
390 CLS
400 PRINT "TERMINADA A GRAVACAO
"
410 PRINT "COLOQUE A CASSETE NO
INICIO"
420 PRINT "PRIMA (ENTER) PARA U
ERIFICAR"
430 PAUSE 0
440 FOR N=1 TO 4: VERIFY ""CODE
: NEXT N
450 CLS
460 PRINT "CODIGO GRAVADO SEM E
RROS"
470 STOP
1000 CLS
1010 PRINT "VERIFICAR CODIGO,PAI

```

```

MA <ENTER> PARA USR A PROXIMA L
INHA.PRIMA <C> PARA CORRIGIR A
LINHA.
1020 PRINT
1030 PRINT
1040 PRINT
1045 LET D=0
1050 FOR N=A TO B STEP 6
1060 PRINT N;" ";
1070 FOR M=0 TO 5
1080 PRINT TAB M*4+7;PEEK (N+M);
1085 LET D=D+PEEK (N+M)
1090 NEXT M
1100 PAUSE 0
1110 IF INKEY#="C" THEN GO TO 20
00
1115 PRINT
1120 NEXT N
1130 IF C<>D THEN GO TO 1000
1140 GO TO 130
20000 CLS
20010 PRINT "ESCREVA A LINHA CORR
MOTA MENTE"
20020 PRINT AT 10,0;
20030 PRINT N;" ";
20040 FOR M=0 TO 5
20050 INPUT "BYTE=";Z
20060 POKE N+M,Z
20070 PRINT TAB M*4+7;PEEK (N+M);
20080 NEXT M
20090 GO TO 1000

```

Lista Decimal

ORG: 40500

405000:	00	10	24	18	1	60
405005:	00	1	00	00	0	70
405010:	00	1	04	10	0	00
405015:	0	0	0	0	0	0

ORG: 41000

410000:	1	1	1	1	1	1
410005:	1	1	1	1	1	1
410010:	1	1	1	1	1	1
410015:	1	1	1	1	1	1
410020:	1	1	1	1	1	1

410030:	1	0	0	0	0	0
410035:	0	0	0	0	0	0
410040:	0	1	1	0	0	0
410045:	0	0	0	0	0	0
410050:	0	0	0	0	0	0
410055:	1	1	1	1	1	1
410060:	0	0	0	0	0	0
410065:	0	1	1	1	1	1
410070:	0	1	1	0	0	0
410075:	0	1	1	1	1	1
410080:	1	1	1	1	1	1
410085:	1	0	0	0	0	0
410090:	0	1	1	1	1	1
410095:	0	0	0	0	0	0
410100:	1	1	1	1	1	1
410105:	1	1	1	1	1	1
410110:	1	0	0	0	0	0
410115:	1	1	1	1	1	1
410120:	0	1	1	0	0	0
410125:	0	1	1	1	1	1
410130:	1	1	1	1	1	1
410135:	0	0	0	0	0	0
410140:	1	1	1	1	1	1
410145:	1	1	1	1	1	1
410150:	1	0	0	0	0	0
410155:	0	0	0	0	0	0
410160:	0	0	0	0	0	0
410165:	1	1	1	1	1	1
410170:	1	1	1	1	1	1
410175:	0	0	0	0	0	0
410180:	1	1	1	1	1	1
410185:	1	1	1	1	1	1
410190:	1	1	1	1	1	1
410195:	0	0	0	0	0	0
410200:	1	1	1	1	1	1
410205:	1	1	1	1	1	1
410210:	1	0	0	0	0	0
410215:	1	1	1	1	1	1
410220:	1	1	1	1	1	1
410225:	1	1	1	1	1	1
410230:	0	0	0	0	0	0
410235:	0	0	0	0	0	0
410240:	1	1	1	1	1	1
410245:	0	0	0	0	0	0
410250:	0	0	0	0	0	0
410255:	0	0	0	0	0	0
410260:	0	0	0	0	0	0
410265:	1	1	1	1	1	1
410270:	1	1	1	1	1	1
410275:	0	0	0	0	0	0
410280:	0	0	0	0	0	0
410285:	0	0	0	0	0	0
410290:	0	0	0	0	0	0
410295:	1	1	1	1	1	1
410300:	1	1	1	1	1	1
410305:	0	0	0	0	0	0
410310:	0	0	0	0	0	0
410315:	0	0	0	0	0	0
410320:	1	1	1	1	1	1

4113300	1	0	0	0	1	0	0
4113306	1	1	1	1	1	1	0
4113342	1	1	0	0	0	0	1
4113348	1	0	0	1	1	1	1
4113354	0	0	0	0	0	0	1
4113350	1	0	1	1	0	1	0
4113355	1	1	1	0	0	0	0
4113372	1	1	1	1	0	1	0
4113378	0	0	0	0	0	1	1
4113384	1	0	0	1	1	1	1
4113390	1	0	0	0	0	0	0
4113395	0	0	0	0	1	1	1
4114002	0	0	0	0	0	1	1
4114008	0	1	0	0	1	1	1
4114114	0	0	0	0	0	0	1
4114200	1	0	1	1	0	1	1
4114205	0	1	1	1	0	0	0
4114302	1	0	1	1	1	1	0
4114308	0	0	0	1	1	0	0
4114444	1	0	1	1	0	1	1
4114500	1	0	0	0	0	1	0
4114505	0	0	1	1	0	1	0
4114502	1	0	1	1	0	0	0
4114508	1	0	0	1	1	0	1
4114774	0	0	1	1	0	0	1
4114800	1	1	1	1	1	0	0
4114805	1	0	1	0	1	1	0
4114902	1	0	1	1	1	1	1
4114908	1	0	1	1	1	0	1
4115004	0	1	0	0	1	0	1
4115100	1	0	0	0	0	0	1
4115105	1	0	0	0	0	1	0
4115202	0	0	1	0	0	0	0
4115208	0	0	1	0	0	0	1
4115304	0	1	0	0	0	0	1
4115340	1	1	1	1	0	1	0
4115345	1	0	1	1	0	1	0
4115352	1	0	1	1	1	1	1
4115358	0	1	1	1	0	1	0
4115364	0	0	1	1	1	1	1
4115370	1	0	0	0	1	0	0
4115375	1	0	1	1	0	1	0
4115382	1	0	1	1	1	0	0
4115388	0	0	1	1	1	0	1
4115394	0	1	1	1	1	1	1
4116000	1	1	1	1	0	1	0
4116006	1	0	0	0	0	1	1
4116102	1	1	1	1	1	0	1
4116108	0	1	1	1	0	0	0
4116204	0	0	0	0	0	0	0

416000:	1	0	0	1	0	0	1	0
416008:	0	0	0	1	1	0	0	0
416016:	0	0	0	0	0	0	0	0
416024:	1	1	0	0	0	1	0	1
416032:	1	0	0	1	1	1	1	1
416040:	1	1	0	0	0	0	0	1
416048:	0	0	0	0	0	0	0	0
416056:	1	1	0	0	0	0	0	1
416064:	1	1	0	0	0	0	0	0
416072:	1	0	0	1	1	1	0	1
416080:	0	1	0	0	1	1	0	0
416088:	0	1	0	0	1	1	0	0
416096:	1	1	0	0	0	1	1	1
416104:	1	1	1	1	0	1	0	0
416112:	0	0	0	0	0	0	0	0
416120:	1	1	0	0	0	0	0	0
416128:	0	0	0	0	0	0	0	0
416136:	1	1	1	1	1	1	1	1
416144:	1	1	0	0	0	0	0	0
416152:	1	1	0	0	0	0	0	0
416160:	0	1	0	0	0	0	0	0
416168:	1	1	1	1	1	1	1	1
416176:	1	1	0	0	0	0	0	0
416184:	1	1	0	0	0	0	0	0
416192:	1	1	0	0	0	0	0	0
416200:	0	1	1	1	1	1	1	1
416208:	0	1	1	0	0	0	0	0
416216:	1	1	1	1	1	1	1	1
416224:	1	1	0	0	0	0	0	0
416232:	1	1	0	0	0	0	0	0
416240:	1	1	0	0	0	0	0	0
416248:	1	1	0	0	0	0	0	0
416256:	1	1	0	0	0	0	0	0
416264:	1	1	0	0	0	0	0	0
416272:	1	1	0	0	0	0	0	0
416280:	1	1	0	0	0	0	0	0
416288:	1	1	0	0	0	0	0	0
416296:	1	1	0	0	0	0	0	0
416304:	1	1	0	0	0	0	0	0
416312:	1	1	0	0	0	0	0	0
416320:	1	1	0	0	0	0	0	0
416328:	1	1	0	0	0	0	0	0
416336:	1	1	0	0	0	0	0	0
416344:	1	1	0	0	0	0	0	0
416352:	1	1	0	0	0	0	0	0
416360:	1	1	0	0	0	0	0	0
416368:	1	1	0	0	0	0	0	0
416376:	1	1	0	0	0	0	0	0
416384:	1	1	0	0	0	0	0	0
416392:	1	1	0	0	0	0	0	0
416400:	1	1	0	0	0	0	0	0
416408:	1	1	0	0	0	0	0	0
416416:	1	1	0	0	0	0	0	0
416424:	1	1	0	0	0	0	0	0
416432:	1	1	0	0	0	0	0	0
416440:	1	1	0	0	0	0	0	0
416448:	1	1	0	0	0	0	0	0
416456:	1	1	0	0	0	0	0	0
416464:	1	1	0	0	0	0	0	0
416472:	1	1	0	0	0	0	0	0
416480:	1	1	0	0	0	0	0	0
416488:	1	1	0	0	0	0	0	0
416496:	1	1	0	0	0	0	0	0
416504:	1	1	0	0	0	0	0	

ORG: 51000

51000: 75 108 1 2 1 1

5 QUADRILÂNDIA



Fig. 8

Argumento

Como último dos grandes Cavaleiros Quadriliões, é seu dever libertar a Quadrilândia do poder do Senhor da Guerra que trouxe centenas de anos de morte, fome e miséria a este planeta, em tempos belo e próspero.

Depois de alcançar o poder há cerca de quinhentos anos, o temível Senhor da Guerra dividiu-se em três seres de diferente poder, cada um deles governando um reino próprio a partir de fortalezas escondidas e inatacáveis.