

SÉRIE
SOFTWARE

DESAFIO

OS MAIS EXCITANTES
JOGOS EM BASIC

FAÇA UMA JOGADA DE MESTRE!

Veja no encarte o formulário de pedido para as 7 fitas que complementam este livro.

Fita 1:

- Gamalete
- Vida

Fita 2:

- Jogo da Velha Tridimensional
- Palitos Chineses

Fita 3:

- Liq-4
- Resta 1

Fita 4:

- Senha
- Forca

Fita 5:

- Damas
- Jogo da Velha
- Palitos
- Palitos com o computador

Fita 6:

- *Trek-Aventura Espacial
- Aventura Administrativa

Fita 7:

- Batalha Naval
- Gerenciador de Tabelas de Jogos

Botelho **DESAFIO OS MAIS EXCITANTES JOGOS EM BASIC**



PROGRAMAS TAMBÉM EM K-7

Contém listagens dos programas para
TK 82-C/83/85 • Sinclair ZX81 • CP-200
Ringo R-470 • AS-1000 e compatíveis.

DESAFIO

OS MAIS EXCITANTES
JOGOS EM BASIC

Antonio Julio Botelho

SÉRIE
SOFTWARE

EDITORA CAMPUS

Títulos de Computação da Editora Campus

ADA[®]; UMA INTRODUÇÃO – *H. Ledgard*
ADMINISTRAÇÃO DE SISTEMAS DE INFORMAÇÃO – *P. Ein-Dor & E. Segev*
ANÁLISE DE SISTEMAS PARA SISTEMAS DE INFORMAÇÃO POR COMPUTADOR – *J. C. Wetherbe*
APLICAÇÕES DE MICROPROCESSADORES – *J. A. Kuecken*
APLICAÇÕES DO COMPUTADOR NA MEDICINA – *N. F. Kember*
APLICALC, UM SOFTWARE EDUCACIONAL, PESSOAL E PROFISSIONAL EM BASIC – *E. A. Meili*
BASIC BÁSICO, 4ª ed. – *J. C. Pereira Filho*
BASIC PARA APLICAÇÕES COMERCIAIS – *D. Hergert*
BASIC PARA MICROS PESSOAIS, 2ª ed. – *J. C. Pereira Filho*
BASIC SINCLAIR – *R. U. Christmann*
BRINCANDO COM O COMPUTADOR – *J. A. Moreira*
COBOL PARA ESTUDANTES, 4ª ed. – *A. Parkin*
COMO LIDAR COM O COMPUTADOR – *H. C. Lucas Jr.*
COMPUTADORES BRASILEIROS; INDÚSTRIA, TECNOLOGIA E DEPENDÊNCIA – *P. B. Tigre*
COMPUTADORES PARA USUÁRIOS – *J. C. Pereira Filho, Coordenador*
Vol. I – Aplicação de Computadores
Vol. II – Equipamentos e Sistemas de Computação
Vol. III – Programas e Programação de Computadores
Vol. IV – Seleção de Sistemas de Computação
CONCEITOS DE PROCESSAMENTO DE DADOS COM BASIC – *R. A. Stern & N. Stern*
A CONSTRUÇÃO DE UM COMPILADOR – *V. W. Setzer & I. S. H. Melo*
CRIANÇA TAMBÉM FAZ PROGRAMAS, 2ª ed. – *J. A. Moreira*
DESAFIO. OS MAIS EXCITANTES JOGOS EM BASIC – *A. J. L. Botelho*
DOCUMENTAÇÃO DE SOFTWARE – *J. D. Lomax*
ESPECIFICAÇÃO DE SISTEMAS – *S. J. Waters*
ESTRUTURAS DE DADOS, 2ª ed. – *P. S. Veloso et al.*
FUNDAMENTOS DE ESTRUTURAS DE DADOS – *E. Horowitz & S. Sahni*
FUNDAMENTOS DE PROCESSAMENTO DE DADOS – *W. T. Price*
GERÊNCIA DE BASES DE DADOS PARA MICROCOMPUTADORES – *E. G. Brooner*

GRAFOS E ALGORITMOS COMPUTACIONAIS – J. L. Szwarcfiter
GUIA DE LINGUAGENS DE COMPUTADORES – H. L. Helms Jr.
GUIA PARA PROGRAMADORES, 2ª ed. – M. Bohl
IMPLANTAÇÃO DE MICROS E MINICOMPUTADORES COMERCIAIS – P. A. Knight
INTRODUÇÃO À ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES – H. Lorin
INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO COM WATFIV E FORTRAN, 2ª ed. – S. E. R. Carvalho
INTRODUÇÃO À PROGRAMAÇÃO COM PASCAL, 2ª ed. – S. E. R. Carvalho
INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES, 3ª ed. – H. V. R. Corrêa Silva et al.
INTRODUÇÃO À PROGRAMAÇÃO FORTRAN – J. C. Pereira Filho
INTRODUÇÃO À SEGURANÇA DO COMPUTADOR – M. B. Wood
INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS – C. J. Date
INTRODUÇÃO AO PROCESSAMENTO DE TEXTOS – G. L. Simons
INTRODUÇÃO AO VISICALC, 2ª ed. – E. A. Garbin
JCL/SISTEMA 370, 3ª ed. – G. D. Brown
LCP – LÓGICA DE CONSTRUÇÃO DE PROGRAMAS, 2ª ed. – J. D. Warnier
LCS – LÓGICA DE CONSTRUÇÃO DE SISTEMAS – J. D. Warnier
LDR – GUIA DOS USUÁRIOS DE SISTEMAS DE INFORMAÇÃO – J. D. Warnier
LINGUAGEM DE PROGRAMAÇÃO ALGOL, 2ª ed. – L. M. Segre
LINGUAGEM PASCAL – V. L. Strube de Lima
MANUAL DE CP/M INCLUINDO MP/M – R. Zaks
MANUAL DE COBOL ESTRUTURADO – D. D. McCracken
MANUAL DO MICROCOMPUTADOR Z-80 – W. Barden Jr.
MANUAL DE LINGUAGEM C – L. Hancock e M. Krieger.
MATEMÁTICA PARA MICROCOMPUTADORES – W. Barden Jr.
MICRO-MINICOMPUTADORES BRASILEIROS – E. L. Passos
MICROCOMPUTADORES PARA APLICAÇÕES COMERCIAIS – W. Barden Jr.
MICROPROCESSADORES DE 16 BITS – C. A. Titus et al.
MICROPROCESSADORES/MICROCOMPUTADORES – A. J. Khambata
Vol. I – Arquitetura - Vol. II – Software e Sistemas
1001 APLICAÇÕES PARA O SEU COMPUTADOR PESSOAL – M. Sawusch
ORGANIZAÇÃO DA INFORMÁTICA NA EMPRESA – E. Rios

ORGANIZAÇÃO DE BANCOS DE DADOS, 4ª ed. – A. L. Furtado e C. S. Santos
PRINCÍPIOS DE SISTEMAS OPERACIONAIS, 3ª ed. – C. C. Guimarães
PRODUTIVIDADE DO PROGRAMADOR – L. J. Arthur
PROGRAMAÇÃO EM ASSEMBLER E LINGUAGEM DE MÁQUINA – D. C. Alexander
PROGRAMAÇÃO ESTRUTURADA EM COBOL – A. B. Furtado
PROGRAMAÇÃO SISTEMÁTICA EM PASCAL, 2ª ed. – N. Wirth
PROGRAMAS ADMINISTRATIVOS EM BASIC SINCLAIR – L. Karsten
RPG-II – J. C. Pereira Filho
REDES DE COMPUTADORES; ASPECTOS TÉCNICOS E OPERACIONAIS – D. A. Menascé e D. Schwabe
ROTINAS MATEMÁTICAS EM BASIC PARA MICROS – C. R. A. Loiola
O SEU COMPUTADOR PESSOAL – M. Waite e M. Pardee
SISTEMA OPERACIONAL UNIX – K. Christian
SISTEMAS DE VIDEOCASSETE; TEORIA E MANUTENÇÃO – G. P. McGinty
SISTEMAS OPERACIONAIS PARA MICROCOMPUTADORES – M. Dahmke
TÉCNICAS E PRÁTICA DE PROGRAMAÇÃO – A. Chantler
TECNOLOGIA DA INFORMAÇÃO: UM GUIA PARA EMPRESAS, GERENTES E ADMINISTRADORES – J. Eaton e J. Smithers
30 PROGRAMAS EM BASIC PARA COMPUTADORES PESSOAIS – D. Chance
USANDO CP/M: UM GUIA EM ENSINO PROGRAMADO – J. N. Fernandez & R. Ashley
MICROBITS
Macroinformações e Programas para Micros TK 82-83-85, CP-200 e compatíveis. Uma publicação bimestral com análises de hardware, aplicativos, artigos sobre linguagem de máquina, dicas de programação e respostas para suas dúvidas.
MICROINFORMÁTICA INTERNACIONAL
Uma publicação bimestral sobre Segurança e Prevenção de Fraudes em Computadores, Aplicação de Microprocessadores e Microcomputadores e Microeletrônica.
Assinaturas: 10 ORTN's

Procure nossas publicações nas boas livrarias ou comunique-se diretamente com:
EDITORA CAMPUS LTDA.
Livros Científicos e Técnicos
Rua Barão de Itapagipe, 55
20261 Rio de Janeiro – RJ – Brasil
Telefone: (021) 284 8443
Atendemos também pelo reembolso postal.

DESAFIO

**OS MAIS EXCITANTES
JOGOS EM BASIC**

SÉRIE
SOFTWARE

DESAFIO

**OS MAIS EXCITANTES
JOGOS EM BASIC**

Antonio Julio Botelho

COORDENADOR
Jorge da Cunha Pereira Filho

EDITORA CAMPUS LTDA.
Rio de Janeiro

© 1985, Editora Campus Ltda.

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/1973. Nenhuma parte deste livro poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Todo o esforço foi feito para fornecer a mais completa e adequada informação. Contudo a editora não assume responsabilidades pelo uso da mesma.

A Editora Campus não é filiada a nenhum fabricante de sistemas computacionais.

Capa
Otavio Studart

Diagramação, composição, paginação e revisão
Editora Campus Ltda.
Rua Barão de Itapagipe 55 Rio Comprido
Tel.: (021) 284 8443
20261 Rio de Janeiro RJ Brasil
Endereço telegráfico: CAMPUSRIO

ISBN 85-7001-201-2

Ficha Catalográfica
CIP-Brasil. Catalogação-na-fonte
Sindicato Nacional dos Editores de Livros, RJ.

B76j Botelho, Antonio Julio Lossio.
Desafio. Os mais excitantes jogos em BASIC / Antonio Julio Lossio Botelho. — Rio de Janeiro : Campus, 1985.
(Série Software ; 1)

Apêndices
ISBN 85-7001-201-2

1. Microprocessadores — Programação 2. Jogos eletrônicos I. Título II. Série

84-0700

CDD — 001.6425
001.6404

Agradecimentos

A Denise, minha esposa, pelo incentivo e ajuda na composição e datilografia do texto.

Aos meus filhos Samya, Marjorie, Danielle e Antônio Júlio Júnior, pelo uso que fizeram dos jogos e pela ajuda na digitação dos programas.

Sumário

APRESENTAÇÃO, 13

CAPÍTULO 1 – *INTRODUÇÃO*, 15

CAPÍTULO 2 – *GAMALETE, O GAMÃO LÓGICO*

A – Ficha, 20

B – Manual de Utilização, 21

C – Manual de Operação, 24

D – Manual Técnico, 24

CAPÍTULO 3 – *UMA VISÃO DO DESENVOLVIMENTO DA VIDA*

A – Ficha, 34

B – Manual de Utilização, 35

C – Manual de Operação, 40

D – Manual Técnico, 40

CAPÍTULO 4 – *O DESAFIO DO JOGO DA VELHA TRIDIMENSIONAL*

A – Ficha, 46

B – Manual de Utilização, 47

C – Manual de Operação, 51

D – Manual Técnico, 52

CAPÍTULO 5 – *O DESAFIO DOS PALITOS CHINESES – NIM*

A – Ficha, 62

B – Manual de Utilização, 63

C – Manual de Operação, 64

D – Manual Técnico, 65

CAPÍTULO 6 – *LIGAÇÕES DESCENDENTES*

A – Ficha, 72

B – Manual de Utilização, 73

- C – Manual de Operação, 74
- D – Manual Técnico, 74

CAPÍTULO 7 – *SERÁ POSSÍVEL SOBRAR SÓ 1?*

- A – Ficha, 82
- B – Manual de Utilização, 83
- C – Manual de Operação, 84
- D – Manual Técnico, 84

CAPÍTULO 8 – *DESCUBRA A SENHA, SE FOR CAPAZ*

- A – Ficha, 90
- B – Manual de Utilização, 91
- C – Manual de Operação, 95
- D – Manual Técnico, 96

CAPÍTULO 9 – *DIVIRTA-SE BRINCANDO DE FORÇA*

- A – Ficha, 104
- B – Manual de Utilização, 105
- C – Manual de Operação, 108
- D – Manual Técnico, 108

CAPÍTULO 10 – *DAMAS TRADICIONAL*

- A – Ficha, 116
- B – Manual de Utilização, 117
- C – Manual de Operação, 118
- D – Manual Técnico, 119

CAPÍTULO 11 – *JOGO DA VELHA TRADICIONAL*

- A – Ficha, 130
- B – Manual de Utilização, 131
- C – Manual de Operação, 134
- D – Manual Técnico, 134

CAPÍTULO 12 – *APRENDA A JOGAR COM PALITOS*

- A – Ficha, 142
- B – Manual de Utilização, 143
- C – Manual de Operação, 144
- D – Manual Técnico, 144

CAPÍTULO 13 – *ENSINE AO COMPUTADOR A JOGAR PALITOS*

- A – Ficha, 150

- B – Manual de Utilização, 151
- C – Manual de Operação, 152
- D – Manual Técnico, 152

CAPÍTULO 14 – *UMA AVENTURA ESPACIAL*

- A – Ficha, 156
- B – Manual de Utilização, 157
- C – Manual de Operação, 168
- D – Manual Técnico, 168

CAPÍTULO 15 – *UMA AVENTURA ADMINISTRATIVA*

- A – Ficha, 184
- B – Manual de Utilização, 185
- C – Manual de Operação, 185
- D – Manual Técnico, 185

CAPÍTULO 16 – *APAIXONANTE E TRADICIONAL BATALHA NAVAL*

- A – Ficha, 194
- B – Manual de Utilização, 195
- C – Manual de Operação, 202
- D – Manual Técnico, 202

CAPÍTULO 17 – *UM GERENCIADOR DE TABELA DE JOGOS*

- A – Ficha, 220
- B – Manual de Utilização, 221
- C – Manual de Operação, 221
- D – Manual Técnico, 225

APÊNDICE A , 231

APÊNDICE B , 241

APÊNDICE C , 245

APÊNDICE D , 249

Apresentação

Quando recebi a tarefa de coordenar a edição da Série *SOFTWARE*, a primeira preocupação foi com a abertura da série, embora muitas áreas de interesse desde logo se apresentassem como candidatas. Um assunto palpitante, do interesse de jovens e adultos, de fácil assimilação eram as características que se impunham. E a escolha recaiu sobre o tema Jogos, por ser, indubitavelmente, o de maior popularidade, porque os jogos divertem enquanto ensinam e educam a inteligência, constituindo uma motivação forte para o uso do microcomputador, o que vem de encontro ao nosso objetivo.

Escolhido o tema, a segunda preocupação foi com a escolha do autor, que deveria ser um profissional de processamento de dados experimentado, com, se possível, uma verdadeira paixão pelos jogos de computador, vivenciada por uma bagagem de realizações concretas nessa área. Seria ideal se esse profissional tivesse uma educação formal sobre computação, de preferência em nível superior, para que pudesse transmitir, no texto do livro, a sua experiência, de uma maneira clara, correta e direta.

Revolvendo os arquivos da minha memória, a pessoa de Antônio Júlio Lossio Botelho foi evocada como aquela que, além de preencher os requisitos, até mesmo os excedia. Além de bacharel em Matemática, também é Mestre em Ciências de Informática pela PUC/RJ.

Meu primeiro contato com o autor foi na época em que ele era o Chefe do Departamento de Computação Eletrônica do LCC, Laboratório de Computação Científica, órgão do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico. Apesar de todos os seus afazeres, ainda encontrava disposição, fora do expediente, geralmente à noite, para desenvolver jogos em seu microcomputador. Fui convidado, então, a jogar sua *Batalha Naval*, e, desnecessário dizer, saí de lá a altas horas da noite, sem jantar ...

Fiquei muito satisfeito – os leitores vão experimentar a mesma sensação ao utilizar os jogos deste livro – quando o meu convite para realizar a produção do livro e dos jogos foi aceito pelo autor.

CAPÍTULO 1

Introdução

De utilidade discutida no passado, os jogos em computador adquiriram muita importância com o advento dos microprocessadores. Essas máquinas permitiram o uso do computador como passatempo e aí os jogos deixaram de ser uma experiência acadêmica para se tornarem negócio — veja-se os vídeo jogos existentes no mercado.

Os vídeo jogos são, na sua maioria, muito bem feitos, porém inacessíveis a quem tenha interesse em estudá-los — exceto para alguns especialistas. O baixo custo dos microcomputadores atualmente existentes no mercado dá condição ao não-especialista a experimentos nessa área.

Os programas dos vídeo jogos são escritos em *linguagem de máquina* e são dependentes do seu próprio sistema, sendo portanto sem utilidade para quem possua microcomputadores pessoais, normalmente programáveis, quase que exclusivamente em BASIC.

Sendo o *BASIC* uma linguagem de *fácil aprendizado*, rapidamente tornou-se quase que universal. Qual o usuário de microcomputador que não sabe programar em BASIC?

Tornou-se assim importante, a nosso ver, que pudéssemos apresentar aqui uma coletânea de jogos, exclusivamente escritos em BASIC.

Estes programas foram feitos para micros compatíveis com o SINCLAIR (Microdigital TK82/C, TK83 e TK85; Prológica CP 200 etc.). Evitou-se o uso de solução exclusiva desta máquina, de forma a permitir, o mais facilmente possível, a adaptação destes programas para outros equipamentos. Por exemplo, **FAST/SLOW** são usados sempre por problemas de velocidade de processamento. Em sistemas sem esta opção os programas funcionarão igualmente. A maioria porém deve ter ajustados os comandos **PRINT**, caso a tela seja diferente da linha SINCLAIR (22 linhas por 32 colunas).

Procuramos também tomar cuidados de medição de tempo e memória.

```

10 LET A = 1
20 LET M = A * 1024
30 DIM X$(M, 1)
40 SAVE "TESTE"

```

Figura 1.1 ASSOCIAÇÃO MEMÓRIA/TEMPO

O programa da Figura 1.1 foi executado em um TK82/C. Na Figura 1.2 você pode ver a variação do tempo de execução da instrução **SAVE** para diferentes valores de A na linha 10.

Valor de A	1ª Medida	2ª Medida	3ª Medida
1	47,11	47,01	46,99
2	1'07,97	1'07,88	1'08,00
3	1'28,92	1'28,90	1'28,87
4	1'50,25	1'50,05	1'49,78
5	2'11,18	2'11,23	2'11,05
6	2'32,42	2'32,28	2'32,35

Figura 1.2 VARIAÇÃO TEMPO/MEMÓRIA

As medidas da Figura 1.2 foram feitas com cronômetro, manualmente, pelo autor, carecendo pois de precisão. Porém são suficientes para concluirmos que, para 1K de dados acrescentados ao programa, cerca de 21 segundos adicionais tornam-se necessários para que a instrução **SAVE** se processe.

Conclui-se então que o programa da Figura 1.1, se executado com $A=0$ (ou seja, **SAVE sem dados**), levaria 26 segundos. Descontados 5 segundos de silêncio antes do **SAVE** conforme padrão SINCLAIR, sobram 21 segundos. Somos pois levados a acreditar que 1K foi transferido para fita.

Todos os programas descritos a seguir são salvos com os comandos que se seguem:

```

9000 FAST
9010 CLEAR
9020 SAVE "nome do programa"
9030 RUN ou GOTO 10

```

Observe que o **CLEAR** libera toda a memória de dados, fazendo com que o **SAVE** seja apenas do texto do programa. Assim, na Figura 1.3 temos a tabela de todos os programas deste livro, com o tempo de **SAVE** e memória utilizada.

Por outro lado,

```
PEEK 16404 + 256 * PEEK 16405 - 16509
```

calcula quanto de memória está em uso naquele instante.

Usando isto durante a execução e após um **CLEAR**, obtivemos os resultados da Figura 1.3.

Estes programas estão disponíveis em 7 (sete) fitas cassete, legíveis por equipamento compatível com o TK82/C.

O nome para a carga (**LOAD**) é o indicado na Figura 1.3.

FITA	NOME	TEMPO DE LOAD	MEMÓRIA	
			TEXTO	TOTAL
1	GAMALETE VIDA	2'35,38 1'00,00	6.385 2.445	7.023 3.463*
2	VELHA-4 NIM	3'03,97 1'54,12	7.487 4.671	9.988 5.151
3	LIQ-4 RESTA-1	1'59,94 1'13,05	4.878 2.983	5.206 3.330
4	SENHA FORCA	1'55,13 1'38,40	4.694 3.924	5.022* 4.143
5	DAMAS VELHA PAR PAR-C	3'20,70	8.183	9.022
		2'14,90	5.517	5.846
		1'14,42	3.028	3.121
		1'12,08	2.958	3.164
6	*TREK ICOOP	5'17,14	12.903	14.432
		2'35,75	6.354	6.414
7	NAVAL TOTO	5'42,44	13.910	15.598
		1'39,01	4.066	4.516*

* Memória pode variar com parâmetro de entrada.

Figura 1.3

RELAÇÃO DOS JOGOS

Tomou-se cuidado com a instrução **STOP**. O usuário de computadores compatíveis SINCLAIR tem uma tendência a utilizar-se do comando **CONT** devido à grande frequência da parada código 5 (tela cheia). Torna-se pois muito frequente teclar-se **CONT** aos demais códigos. Sendo código de erro, o código volta, pois o erro certamente continua. No caso de código 9 – **STOP**, a próxima instrução é executada e o programa vai continuar, agora completamente fora de controle. De forma geral inserimos uma mensagem e/ou controles para que o programa seja reiniciado ou que a parada se repita.

CAPÍTULO 2

*Gamalete,
O Gamão Lógico*

A FICHA

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: <u>GAMALETE</u>	c) Versão: <u>1</u>	d) Revisão: <u>0</u>	e) Data: <u>22/10/83</u>
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>		SLOW <input checked="" type="checkbox"/>	
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>		FAST <input checked="" type="checkbox"/>	
TK 80 <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>		LPRINT <input type="checkbox"/>	
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>		LLIST <input type="checkbox"/>	
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>		COPY <input type="checkbox"/>	
SINCLAIR ZX 80 <input type="checkbox"/>	64K <input type="checkbox"/>		DLOAD <input type="checkbox"/>	
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>		DSAVE <input type="checkbox"/>	
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>		DMLOAD <input type="checkbox"/>	
TIMEX 1000 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>		DMSAVE <input type="checkbox"/>	
CP-200 <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>	
CP-300 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>			
APPLY 300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>			
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
<u>RINGO R-470 e</u>				
<u>AS - 1000</u>				
k) Observações: _____ _____ _____				

B MANUAL DE UTILIZAÇÃO

O GAMALETE é jogado normalmente por duas pessoas e é necessário ter um lápis e 36 pedras. Com o lápis desenha-se como mostra a Figura 2.1.

	6	5	4	3	2	1	
X							Y
	1	2	3	4	5	6	

Figura 2.1
TABULEIRO DESENHADO

Junto a cada número temos *casas*. Nos ambientes X e Y temos *prisões*. Para começar, distribuem-se as pedras (que podem ser caroços de feijão ou milho) que serão colocadas nas *casas* (3 em cada). Cada jogador tem 6 *casas* numeradas de 1 a 6. Os jogadores se alternam para o início de cada partida.

Cada jogador seleciona uma *casa* (não vazia) e pega todas as pedras nela existentes e as distribui, uma a uma, nas *casas* subsequentes, no sentido da numeração crescente das *casas* (1 para 6). Durante o movimento, se terminarem as *casas* do jogador a seqüência continua nas *casas* do oponente. Ou seja, o movimento é circular. Contudo, ao passar para as *casas* do oponente, a *prisão* é tratada como *casa*, isto é, recebe uma pedra como as demais *casas*. A *prisão* não pode ser selecionada. O objetivo na verdade é pôr o maior número possível de pedras na *prisão*. Sempre que se executar um movimento, se a última pedra cair na *prisão*, tem-se o direito a um segundo movimento. Nunca porém a um terceiro. E, o mais importante, sempre que a última pedra cair em uma *casa vazia*, se e somente se a *casa oposta* estiver *não vazia*, esta última pedra, mais todas as da *casa oposta* são levadas para a *prisão*. O jogo termina quando um dos jogadores não possuir *casa não vazia* para jogar. Vence quem tiver então maior número de pedras na *prisão*. A Figura 2.2 mostra a representação do tabuleiro, feita na tela, pelo programa.

O programa começa perguntando:

QUEM COMECA?

Responda EU se você deseja fazer o primeiro movimento.

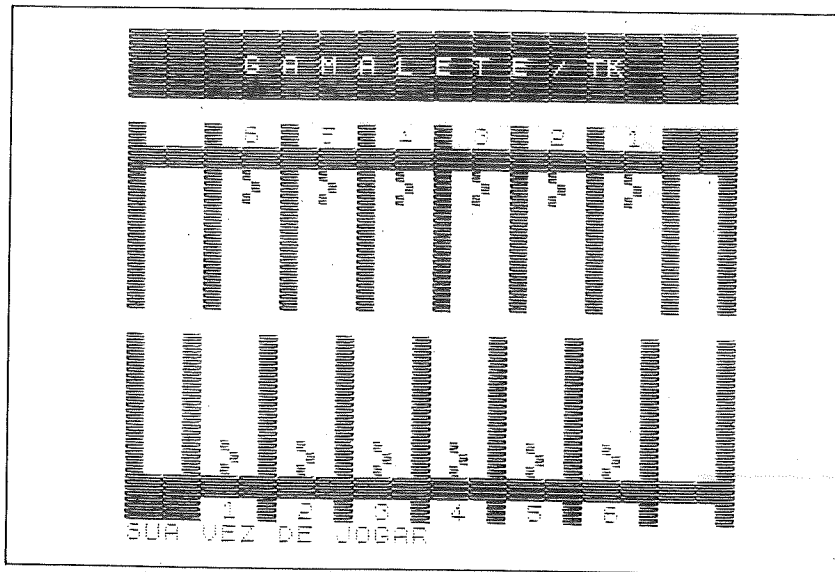


Figura 2.2
TABULEIRO DESENHADO PELO TK

Responda TU ou VOCE se deseja que o computador faça o primeiro movimento.

Tecla NEWLINE.

A tela então se apagará e o tabuleiro será desenhado.

Se o computador deve começar, ele fará o movimento a seguir.

Vem a seguir o pedido:

SUA VEZ DE JOGAR

Tecla o número identificador da casa que você escolheu. O computador se encarregará de mover as pedras por você, segundo as regras. Se você ganhou direito a um segundo movimento, ele dirá

VOCE NOVAMENTE

Tecla o número identificador da casa selecionada para o segundo movimento.

Quando for a vez dele jogar, a tela se apaga. Ao selecionar a jogada dele, na posição N, a tela volta com a mensagem

JOGAREI COM N

A seguir movimentar as pedras. Se ele ganhar o direito a um segundo movimento, novamente a tela se apaga, repetindo-se o processo.

Este jogo fez muito sucesso pela sua sensibilidade e *inteligência*. Entenda aí *inteligência* como a capacidade de aprender. Ele inicia sabendo apenas as regras e vai sendo uma presa fácil a qualquer jogador com um mínimo de experiência. A sensibilidade dele já se faz sentir na segunda partida. Se você tentar repetir os movimentos da primeira, verá que ele tentará soluções diferentes da anterior. Após cinco ou seis partidas, torna-se um contendor rigorosamente a sua altura.

Para exemplificar, damos a seguir três jogos:

1ª Partida – humano começa

movimento	1	2	3	4	5	6	7	8
humano joga	6	5	4/3	2/5	1/2	3	4	5
computador	1	2/6	5	6/3	6/5	6/4	5	—

O humano venceu por 9 pedras.

Na segunda partida o computador começa e não se pode reproduzir, ou tentar, a partida anterior. Na terceira o humano começa novamente e podemos tentar repetir a vitória da primeira.

3ª Partida – humano começa

movimento	1	2	3	4	5	6	7
humano joga	6	5	4/3	2/5	1/2	3/4	6
computador	1	2/5	6	4	3	5	—

O humano venceu por 11 pedras

Observe que o computador já modificou seu movimento a partir do seu segundo lance. Esta mudança afetou a tentativa de repetir a primeira seqüência no 6º movimento, fazendo com que o 7º fosse antecipado e o final fosse com a casa 6 e não mais com a 5. Porém ele continua perdendo e mais feio ainda. A 4ª partida ele começou, não permitindo, portanto, uma tentativa válida de reprodução da seqüência vitoriosa. Vejamos então o que ocorre na 5ª partida.

5ª Partida – humano começa

movimento	1	2	3	4	5	6	7	8	9
humano joga	6	5	4/3	2	5	1	3	4	5/6
computador	6	3	4	2	6/5	6/4	3	1	—

O computador venceu por 11 pedras.

Veja que desde o primeiro lance o computador vem dando respostas diferentes, que porém só forçaram modificações na seqüência até então vitoriosa no 4º movimento.

C MANUAL DE OPERAÇÃO

Quase 3 minutos são necessários para a carga deste programa de 7K a partir da fita cassete. Veja Apêndices para detalhes sobre o uso de fitas.

D MANUAL TÉCNICO

Dois vetores armazenam as informações básicas do GAMALETE. B contém o tabuleiro. F a *inteligência*.

O programa divide-se em 3 blocos distintos:

- 1 – Formatação da tela;
- 2 – Controle do jogo e
- 3 – *Inteligência*.

A formatação da tela é feita, após a inicialização, pela rotina 5000. Até aí ela é bem simples. Ao final, para cada *casa* a rotina 7900 é chamada. Esta rotina é que já merece algum comentário adicional.

A variável S\$ contém as colunas onde devem ser impressas as pedras de cada casa. Assim, a primeira coisa a fazer em 7900 é chamar 7700 para as conversões. Aí as pedras são impressas nas linhas correspondentes, de cima para baixo ou de baixo para cima, dependendo de ser *casa* inferior ou superior.

Observe-se a importância da direção do movimento porque uma *casa* ou está sendo esvaziada (rotina 7800) ou tendo o número de pedras aumentada, não sendo pois necessário apagar posições adicionais na tela. Observe na listagem a semelhança entre as rotinas 7800 e 7900.

A rotina 7000 é que efetua o movimento da *casa* M para as *casas* subsequentes. Observe que a rotina 7800 é chamada para refazer as B(M) *casas* subsequentes.

O controle sobre o jogo é feito da linha 310 à 940.

A rotina 1000 controla a jogada humana.

O controle da *inteligência* inicia-se na rotina 1000 ao se anotar a jogada humana na linha 1250.

Como o fundamental no GAMALETE são as 9 primeiras jogadas (raramente se passa de 10 movimentos), elas são anotadas em F(N) onde N é o contador de partidas. Observe-se que se usa o sistema sexagesimal para armazenagem, de forma a minimizar o uso de memória.

Na rotina 2000 anota-se o resultado. Positivo, se quem começou ganhar. Negativo, se quem começou perder. Aí também pede-se para jogar novamente. Parar será desastroso, pois a *inteligência* dele será destruída.

Para determinar-se o movimento do computador, salva-se o tabuleiro no vetor G e joga-se com cada uma das *casas não vazias* (a rotina 6000 move sem mostrar na tela, como a 7000) e determina qual delas tem melhor avaliação. Até o 9º movimento, examina-se se esse movimento foi efetuado na mesma seqüência. Se não foi, vai-se em frente. Se foi e quem fez o movimento ganhou, então melhora-se a avaliação. Se foi, e quem fez o movimento perdeu, diminui-se a avaliação. Ao final, seleciona-se a *casa* de melhor avaliação.

A avaliação básica é pelo ganho de pedras na *prisão*.

O tempo que o TK82-C leva para escolher a jogada é cerca de 10 segundos.

Na Figura 2.3 encontra-se a listagem do programa-fonte.

```
1 REM GAMALETE VERSAO TK82
10 DIM B(14)
20 LET EU=-1
25 LET AC=0
30 LET N=1
40 DIM G(14)
50 DIM F(30)
60 LET ME=0
70 LET P#=""
80 LET N#=""
90 LET M#=""
100 LET S#="0509131721252926221
814100502"
110 LET TU=0
120 LET A=0
130 SLOW
140 CLS
150 PRINT AT 21,0;"QUEM COMECA?"
160 INPUT A#
```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE)

```

170 IF A#="EU" THEN GOTO 200
180 IF A#="TU" THEN GOTO 300
190 IF A#="VOCE" THEN GOTO 340
2000 SCROLL
210 PRINT AT 21,0;"NAD ENTENDI"
220 SCROLL
230 GOTO 150
240 LET EU=1
250 REM INICIO DO JOGO
310 LET E=0
320 FOR I=1 TO 13
330 LET B(I)=3
340 NEXT I
350 LET C=0
360 LET F(N)=0
370 LET B(7)=0
380 LET B(14)=0
390 LET EU=-EU
400 GOSUB 5000
410 IF EU=1 THEN GOTO 600
420 GOSUB 1050
430 IF E=0 THEN GOTO 800
440 IF M<>H THEN GOTO 600
450 GOSUB 1000
460 IF E=0 THEN GOTO 800
470 GOSUB 4000
480 PRINT AT 21,0;"MINHA VEZ",
490 GOSUB 6000
500 IF E=0 THEN GOTO 800
510 IF M<>H THEN GOTO 750
520 GOSUB 4000
530 PRINT AT 21,0;"EU NOVAMENTE
540 GOSUB 6000
550 IF E>0 THEN GOTO 420
560 GOSUB 4000
570 LET D=B(7)-B(14)
580 LET AC=AC+D
590 IF D>0 THEN GOTO 910
600 LET G=-1
610 IF D<0 THEN GOTO 880
620 PRINT AT 21,0;"EMPATAMOS,"
630 GOTO 2000
640 PRINT AT 21,0;"EU GANHEI PO
650 R " -D;" PONTOS,"
660 LET ME=ME+1
670 GOTO 2000
680 LET TU=TU+1
690 LET G=1
700 PRINT AT 21,0;"VOCE GANHOU
710 FOR " )D;" PONTOS,"
720 GOTO 2000

```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE) (cont.)

```

1000 REM JOGADA HUMANA.
1010 GOSUB 4000
1020 PRINT AT 21,0;"VOCE JOGA NO
1030 VARA?"
1040 GOTO 1070
1050 GOSUB 4000
1060 PRINT AT 21,0;"SUA VEZ DE J
10700 LET A#=INKEY#
1074 IF A#="D" THEN GOTO 2400
1076 IF A#="P" THEN GOTO 2500
1080 IF A#="0" THEN GOTO 1070
1090 IF A#="8" THEN GOTO 1070
1100 LET M=VAL A#
1110 IF B(M)=0 THEN GOTO 1070
1120 LET H=7
1130 GOSUB 1200
1140 RETURN
1150 LET K=M-1
1160 GOSUB 7000
1170 LET E=0
1180 IF K>6 THEN LET K=K-7
1190 LET C=C+1
1200 IF C<9 THEN LET F(N)=F(N)+E
+C
1210 FOR I=1 TO 6
1220 IF B(I)<>0 THEN GOTO 1300
1230 NEXT I
1240 RETURN
1250 FOR I=6 TO 13
1260 IF B(I)<>0 THEN GOTO 1340
1270 NEXT I
1280 RETURN
1290 LET E=1
1300 RETURN
1310 REM MEMORIZA OS RESULTADOS
1320 LET F(N)=F(N)+E*EU
1330 IF N<30 THEN LET N=N+1
1340 FOR I=1 TO 50
1350 NEXT I
1360 GOSUB 4000
1370 PRINT AT 21,0;"VOCE QUER JO
1380 GAR NOVAMENTE?"
1390 LET A#=INKEY#
1400 IF A#="S" THEN GOTO 300
1410 IF A#="N" THEN GOTO 2500
1420 IF A#="D" THEN GOTO 2400
1430 GOTO 2080
1440 FOR I=1 TO 30 STEP 2
1450 SCROLL
1460 PRINT AT 21,0;I;" - ";F(I);
AT 21,16;I+1;" - ";F(I+1)

```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE) (cont.)


```

2430 NEXT I
2500 SCROLL
2510 SCROLL
2520 PRINT AT 21,0;"EU GANHEI
ME;" PARTIDAS"
2530 SCROLL
2540 PRINT AT 21,0;"VOCE GANHOU
";TU;" PARTIDAS"
2550 SCROLL
2560 IF AC=0 THEN GOTO 2700
2570 PRINT AT 21,0;"FICAMOS COM
UM SALDO DE "
2580 SCROLL
2590 IF AC<0 THEN GOTO 2800
2600 PRINT AT 21,0;AC;" A SEU FA
VOR"
2610 GOTO 2810
2700 PRINT AT 21,0;"EMPATAMOS EM
NUMERO DE PONTOS"
2710 GOTO 2810
2800 PRINT AT 21,0;"-AC;" A MEU FA
VOR"
2810 STOP
0820 RUN
4000 PRINT AT 21,0;"
4010 RETURN

5000 REM FORMATAÇÃO DA TELA
5010 CLS
5020 PRINT AT 0,0;P#
5030 PRINT AT 1,0;"          G A M A L E T E
L E T E 2 1 K "
5040 PRINT AT 2,0;P#
5050 PRINT AT 4,0;"          ■ 6 ■ 5 ■
4 ■ 3 ■ 2 ■ 1 ■
5060 PRINT AT 5,0;P#
5070 FOR I=6 TO 11
5080 PRINT AT I,0;M#
5090 NEXT I
5100 FOR I=13 TO 18
5110 PRINT AT I,0;N#
5120 NEXT I
5130 PRINT AT 19,0;P#
5140 PRINT AT 20,0;"          ■ 1 ■ 2 ■
0 ■ 4 ■ 5 ■ 6 ■
5150 FOR X=1 TO 14
5170 GOSUB 7900
5180 NEXT X
5190 RETURN
6000 REM TESTA MOVIMENTO
6010 LET P=B(M)

```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE) (cont.)

```

6020 LET B(M)=0
6030 FOR I=1 TO P
6040 LET M=M+(M<14)-(M=14)*13
6050 LET B(M)=B(M)+1
6060 NEXT I
6070 IF B(M)<>1 THEN RETURN
6080 IF M=7 OR M=14 THEN RETURN
6090 IF B(14-M)=0 THEN RETURN
6110 LET B(H)=B(H)+B(14-M)+1
6120 LET B(14-M)=0
6130 LET B(M)=0
6140 RETURN
7000 REM MOVE
7010 LET X=M
7020 GOSUB 7800
7030 LET P=B(M)
7040 LET B(M)=0
7050 FOR I=1 TO P
7060 LET M=M+(M<14)-(M=14)*13
7070 LET B(M)=B(M)+1
7080 LET X=M
7090 GOSUB 7900
7100 NEXT I
7110 IF B(M)<>1 THEN RETURN
7120 IF M=7 OR M=14 THEN RETURN
7130 IF B(14-M)=0 THEN RETURN
7140 LET X=14-M
7150 LET B(H)=B(H)+B(M)+B(X)
7160 GOSUB 7800
7170 LET B(X)=0
7180 LET X=M
7190 GOSUB 7800
7200 LET B(M)=0
7210 LET X=H
7220 GOSUB 7900
7230 RETURN
7700 LET Z=X*2-1
7710 LET CC=VAL (S$(Z TO Z+1))
7720 RETURN
7800 GOSUB 7700
7805 LET Y=(B(X)+1)/2
7810 IF X>7 THEN GOTO 7860
7820 FOR U=19-Y TO 18
7830 PRINT AT U,CC;" "
7840 NEXT U
7850 RETURN
7860 FOR U=6 TO 5+Y
7870 PRINT AT U,CC;" "
7880 NEXT U
7895 RETURN

```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE) (cont.)

```

7900 GOSUB 7700
7905 LET Y=INT (B(X)/2)
7910 IF X>7 THEN GOTO 7960
7915 IF Y#2<B(X) THEN PRINT AT 1
8-Y,CC;"# "
7920 FOR W=19-Y TO 18
7930 PRINT AT W,CC;"# "
7940 NEXT W
7950 RETURN
7960 FOR W=6 TO 5+Y
7970 PRINT AT W,CC;"# "
7980 NEXT W
7990 IF Y#2<B(X) THEN PRINT AT 6
+Y,CC;"# "
7995 RETURN
8000 REM JOGADA DO TK
8020 LET D=-99
8030 LET H=14
8040 FOR I=1 TO 14
8050 LET G(I)=B(I)
8060 NEXT I
8065 FAST
8070 FOR J=8 TO 13
8080 IF B(J)=0 THEN GOTO 8800
8090 LET Q=0
8100 LET M=J
8110 GOSUB 8000
8120 FOR I=1 TO 6
8130 IF B(I)=0 THEN GOTO 8400
8140 LET L=B(I)+I
8150 LET R=0
8160 IF L<=14 THEN GOTO 8200
8170 LET L=L-14
8180 LET R=1
8190 GOTO 8160
8200 IF B(L)<>0 THEN GOTO 8300
8210 IF L=7 OR L=14 THEN GOTO 83
00
8220 LET R=B(14-L)+R
8300 IF R>0 THEN LET Q=R
8400 NEXT I
8410 LET Q=B(14)-B(7)-Q
8420 IF Q>8 THEN GOTO 8700
8425 IF N=1 THEN GOTO 8700
8430 LET K=J-1
8440 IF K>6 THEN LET K=K-7
8450 FOR I=1 TO N-1
8460 IF F(N)*6+K=INT (ABS (F(I))
/6** (7-C)) THEN LET Q=Q-(SGN (F(
I))=EU)*2
8470 NEXT I
8700 FOR I=1 TO 14

```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE) (cont.)

```

8710 LET B(I)=G(I)
8720 NEXT I
8730 IF Q<D THEN GOTO 8800
8740 LET A=J
8750 LET D=Q
8800 NEXT J
8805 SLOW
8810 LET M=A
8830 GOSUB 4000
8840 PRINT AT 21,0;"JOGAREI COM
";M-7;"."
8850 PAUSE 200
8870 GOSUB 1200
8890 RETURN
9000 CLEAR
9010 SAVE "GAMALETE"
9020 GOTO 10

```

Figura 2.3

LISTAGEM DO PROGRAMA-FONTE (GAMALETE) (cont.)

CAPÍTULO **3**

***Uma Visão do
Desenvolvimento
da Vida***

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: VIDA	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>		SLOW <input checked="" type="checkbox"/>	
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>		FAST <input checked="" type="checkbox"/>	
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>		LPRINT <input type="checkbox"/>	
SINCLAIR ZX 8Ø <input type="checkbox"/>	64K <input type="checkbox"/>		LLIST <input type="checkbox"/>	
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>		COPY <input type="checkbox"/>	
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>		DLOAD <input type="checkbox"/>	
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>		DSAVE <input type="checkbox"/>	
CP-2ØØ <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>		DMLOAD <input type="checkbox"/>	
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>		DMSAVE <input type="checkbox"/>	
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>	
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
RINGSO R-47Ø e				
AS - 1ØØØ				
k) Observações:				

Após o **RUN NEWLINE** ou **LOAD " " NEWLINE** o programa pede que seja teclada a letra **I** para iniciar. Se você deseja modificar algum parâmetro, teclae **BREAK** – no manual técnico você encontrará mais detalhes. Veja a Figura 3.1.

Após teclar **I** o sistema apaga a tela. Alguns segundos depois a tela volta apresentando a geração **0** (zero), distribuída no quadro ao acaso. **2Ø** segundos depois, apaga a tela para calcular a geração seguinte. Este processo repete-se indefinidamente, até que não haja alteração no quadro.

O jogo da vida foi descrito na revista *Scientific America* de outubro de 197Ø, em artigo de Martim Gardner. Nesse artigo é discutido o jogo criado por John Conway – Universidade de Cambridge, Inglaterra. A teoria deste jogo consiste na existência de organismos em um grande tabuleiro. Esses organismos morrem, sobrevivem e se reproduzem segundo leis genéticas. O critério de Conway exige que as regras permitam que ocorra o seguinte:

- Não deve existir uma distribuição inicial de organismos tal que possa se provar que a população possa crescer sem limites;
- Deve existir pelo menos uma distribuição inicial de organismos tal que se tenha a impressão de desenvolvimento contínuo;
- Devem existir configurações iniciais que levem a população à situação de estabilidade, isto é, não há mortes nem nascimentos entre gerações sucessivas ou que, após um número finito de gerações, se volte a uma configuração anterior.

As regras discutidas por Gardner, que satisfazem a estes critérios, são:

- Ambiente** – Os organismos estão distribuídos em um plano reticulado;
- Sobrevivência** – Um organismo sobrevive de uma geração a outra se e somente se existirem dois ou três outros organismos na sua vizinhança (retículos em volta) imediata;
- Nascimento** – Um organismo nasce em um ponto vazio do tabuleiro se e somente se existirem exatamente três organismos na vizinhança desse ponto;
- Morte** – Um organismo morre sempre que estiver só (sem vizinhos) ou em ambientes superpovoados (mais que três vizinhos).

É importante que uma geração não influa sobre si mesma, isto é, a morte ou nascimento de um organismo não influi na morte, sobrevivência ou

nascimento de outro organismo na mesma geração. Bonitas figuras se formarão na tela ao longo do tempo, como mostrado no exemplo da Figura 3.2.

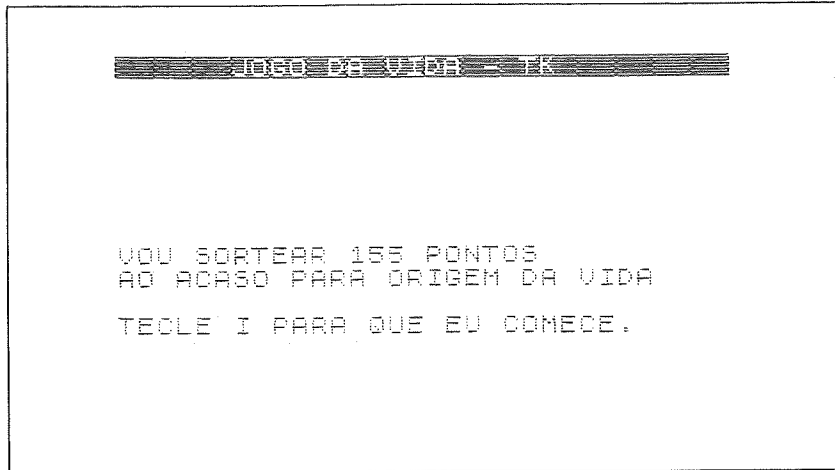


Figura 3.1
TELA INICIAL

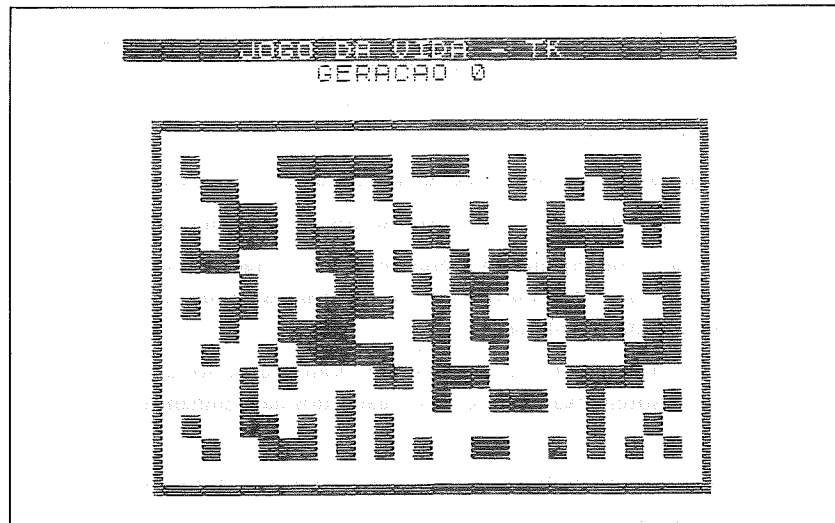


Figura 3.2
EXEMPLO

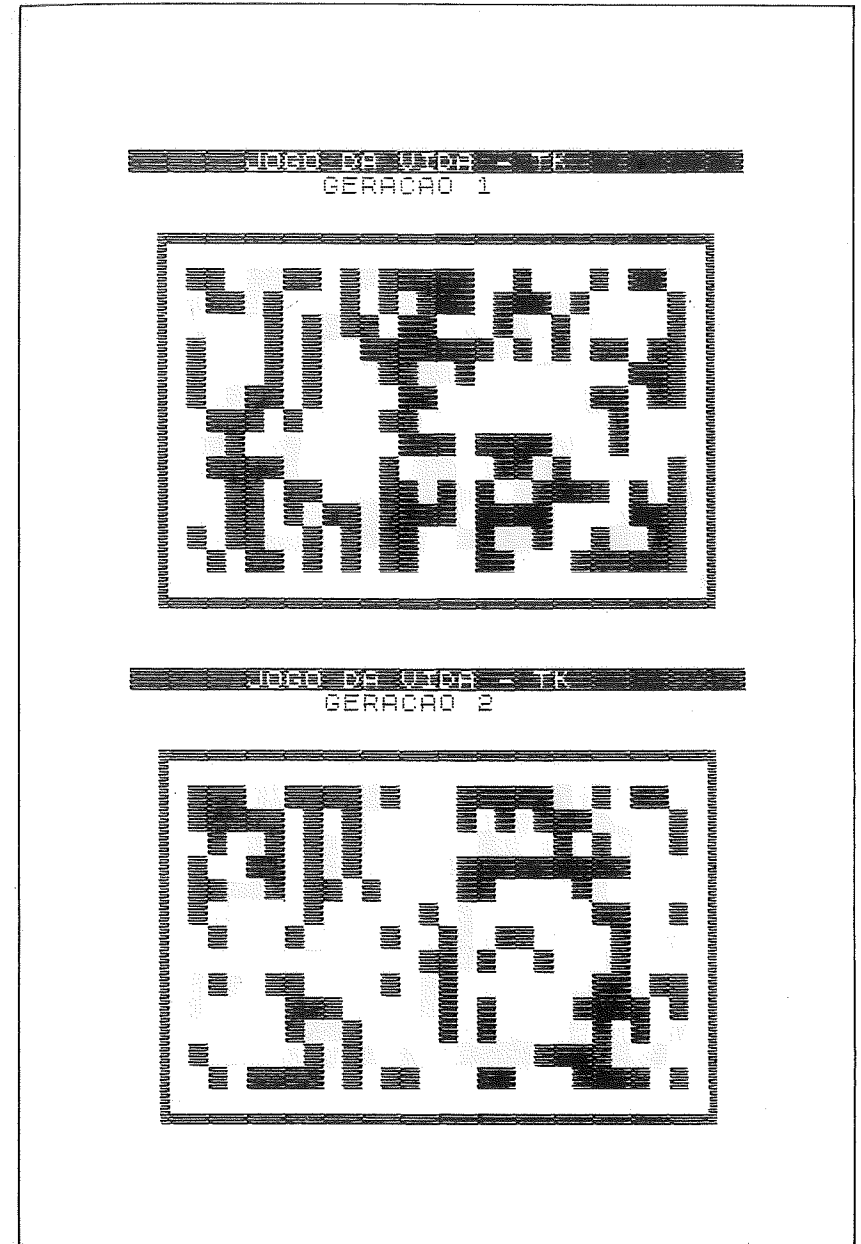


Figura 3.2
EXEMPLO (cont.)

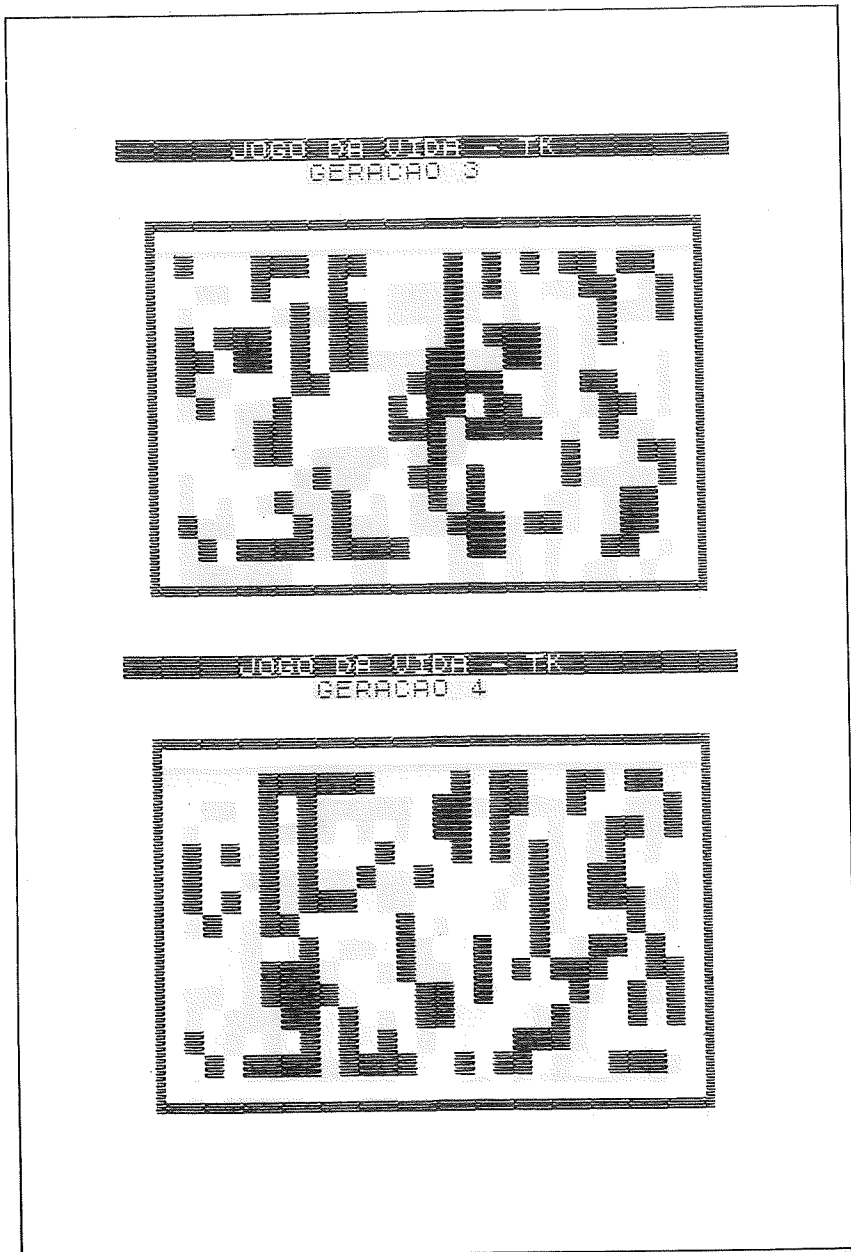


Figura 3.2 *EXEMPLO (cont.)*

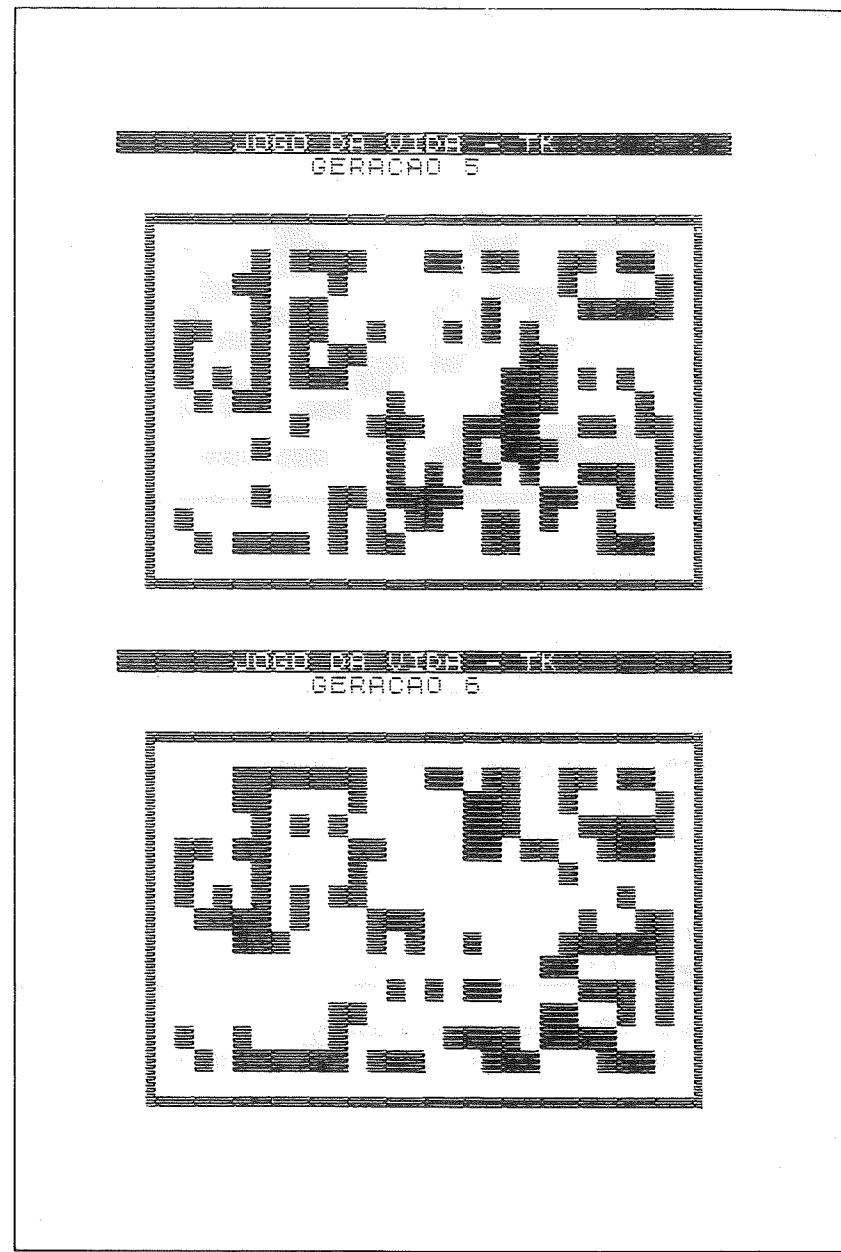


Figura 3.2 *EXEMPLO (cont.)*

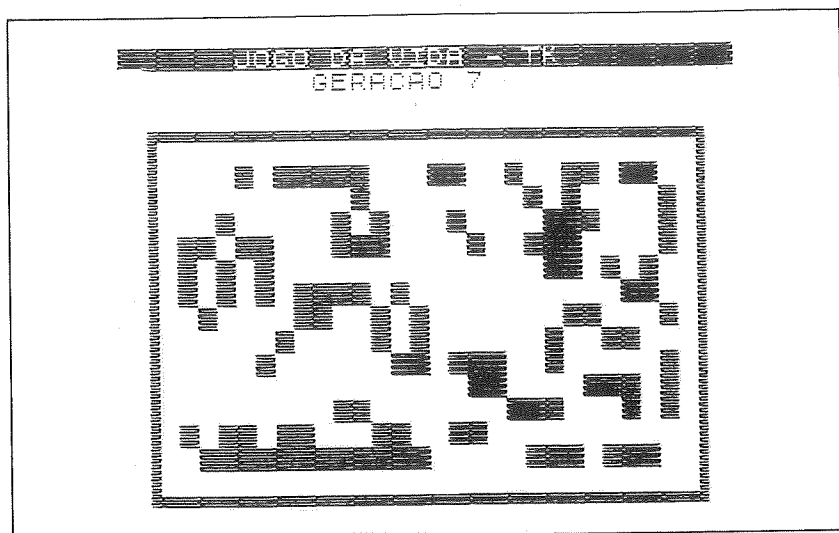


Figura 3.2 EXEMPLO (cont.)

C MANUAL DE OPERAÇÃO

Menos de 1 minuto e meio é necessário para a carga deste jogo, que ocupa cerca de 3K a 8K, dependendo de parâmetros entrados nas linhas 21 e 22 (veja manual técnico).

Observe os Apêndices, como usar fitas cassete.

D MANUAL TÉCNICO

Duas matrizes V\$ e W\$ mantêm cópias da geração anterior e geração atual. O programa listado a seguir acompanha rigorosamente as regras de Conway descritas anteriormente.

As linhas 21 e 22 contêm a inicialização do tamanho do tabuleiro, respectivamente;

```
21 LET DI = 28
22 LET DJ = 15
```

Com estes valores temos um tabuleiro de $28 \times 15 = 420$ pontos. No TK82-C, consomem-se aproximadamente 50 segundos no processamento de uma

geração sendo 42 segundos mais 8 segundos constante — portanto cerca de 10 pontos/segundo. Diminuir o tamanho do tabuleiro aumenta a velocidade, dando porém menos efeitos interessantes. Experimente modificar estes valores. Na linha 20 estão anotados os limites para estas duas variáveis.

A rotina 200 foi originalmente codificada como na Figura 3.3.

```
240 LET X = 0
245 FOR K = J - 1 TO J + 1
250 FOR L = I - 1 TO I + 1
255 IF W$(K, L TO L) = "" THEN LET X = X + 1
257 NEXT L
258 NEXT K
```

Figura 3.3 ROTINA 200 ORIGINAL

A velocidade de processamento era então de 4,2 pontos, por segundo.

Modificada a linha:

```
255 LET X = X + CODE W$(K, L TO L)
```

A velocidade passou para 5,5 pontos por segundo. A solução final praticamente dobrou a velocidade. Acreditamos que outras otimizações fossem possíveis neste jogo tão dependente de tempo para tornar-se interessante. Descartada a hipótese de programá-lo em linguagem de máquina, restam otimizações deste tipo, que aumentam sensivelmente o uso da memória.

Na Figura 3.4 encontra-se a listagem do programa-fonte.

```
10 REM JOGO DA VIDA
12 SLOW
15 CLEAR
18 LET T5 = "
19
20 REM 4(=LI(=30 E 4(=LJ(=17 -
-> DEF. QUADRO
21 LET LI=28
22 LET LJ=15
23 LET DI=(32-LI)/2
25 LET DJ=(19-LJ)/2+2
26 CLS
27 DIM V$(LJ,LJ)
```

Figura 3.4 LISTAGEM DO PROGRAMA-FONTE (VIDA)

```

28 LET UU=INT (LI*LJ/4)
29 LET UU=INT (UU+RND*UU)
30 PRINT T#;
31 DIM U$(LJ,LI)
40 PRINT AT 0,0;"VOU SORTEAR "
;UU;" PONTOS"
41 PRINT "AO ACASO PARA ORIGEM
DA VIDA"
42 PRINT
43 PRINT "TECLE I PARA QUE EU
COMECE."
50 LET A#=INKEY$
60 IF A#(<>)"I" THEN GOTO 50
100 CLS
101 FAST
102 LET Y=1
115 FOR I=1 TO LJ
119 LET U$(I)=" "
120 LET V$(I)=" "
122 NEXT I
123 LET G=0
130 FOR K=1 TO UU
133 LET I=INT (RND*(LI-2))+2
134 LET J=INT (RND*(LJ-2))+2
135 IF U$(J,I TO I)="█" THEN GO
TO 133
140 LET U$(J,I TO I)="█"
145 NEXT K
146 PRINT AT 0,0;T#
150 LET H=DJ+LJ
151 LET I=DI-1
152 LET J=DI+LI
153 FOR K=DJ-1 TO H
160 PRINT AT K,I;"█";AT K,J;"█"
165 NEXT K
166 PRINT AT H,I;"█";AT H,J;"█"
167 PRINT AT DJ-1,I;"█";AT DJ-1
J;"█"
168 LET J=DI+LI-1
169 LET I=DJ-1
170 FOR K=DI TO J
188 PRINT AT I,K;"█";AT H,K;"█"
191 NEXT K
195 GOTO 325
200 FAST
205 LET Y=0
210 FOR J=2 TO LJ-1
220 FOR I=2 TO LI-1
230 LET U$(J,I TO I)=" "
240 LET X=CODE U$(J-1,I-1)+CODE
U$(J-1,I)+CODE U$(J-1,I+1)+CODE
U$(J+1,I-1)+CODE U$(J+1,I)+CODE

```

Figura 3.4

LISTAGEM DO PROGRAMA - FONTE (VIDA) (cont.)

```

U$(J+1,I+1)+CODE U$(J,I-1)+CODE
U$(J,I+1)
250 IF U$(J,I TO I)=" " THEN GO
TO 280
260 IF X>=128 AND X<=384 THEN G
OTO 300
270 GOTO 310
280 IF X<>384 THEN GOTO 310
300 LET U$(J,I TO I)="█"
310 NEXT I
320 NEXT J
325 SLOW
326 LET K=DJ-1
328 PRINT AT 1,10;"GERACAO ";G
329 LET G=G+1
330 FOR J=1 TO LJ
350 PRINT AT K+J,DI;U$(J)
360 LET Y=Y+(U$(J)<>U$(J))
370 LET V$(J)=U$(J)
410 NEXT J
420 FOR I=1 TO 200
430 NEXT I
440 REM COPY
700 IF Y<>0 THEN GOTO 200
710 STOP
720 RUN
9000 CLEAR
9010 SAVE "UID█"
9020 GOTO 10

```

Figura 3.4

LISTAGEM DO PROGRAMA - FONTE (VIDA) (cont.)

CAPÍTULO 4

***O Desafio do
Jogo da Velha
Tridimensional***

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: VELHA TRIDIMENSIONAL	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 80 <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
SINCLAIR ZX 80 <input type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>		
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1000 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-200 <input checked="" type="checkbox"/>	5 1/4" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-300 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
RINGO R-470 e A5-1000				
k) Observações:				

O jogo da VELHA é extremamente conhecido.

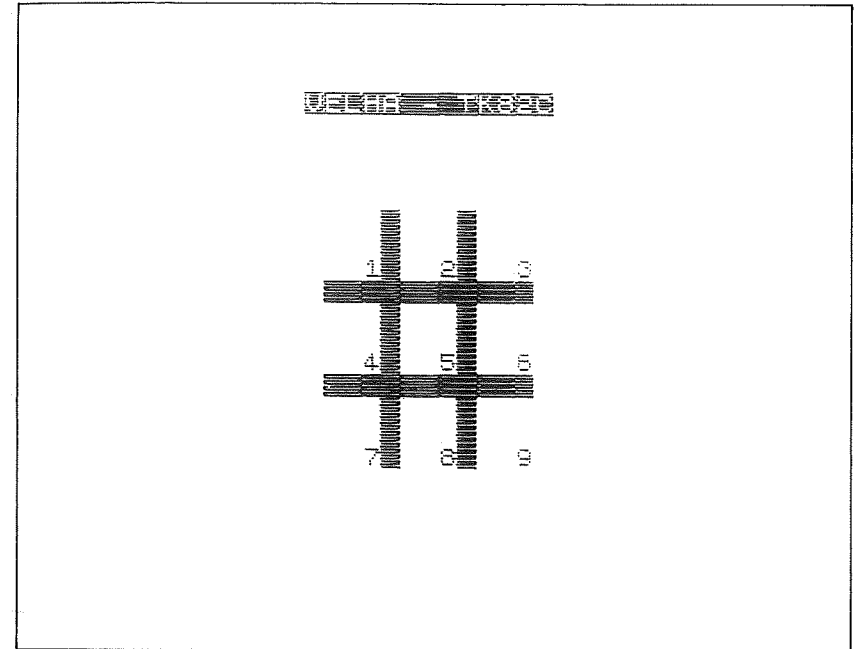


Figura 4.1 TABULEIRO DO JOGO DA VELHA TRADICIONAL

Usa uma matriz 3 x 3 e dois oponentes com um mínimo de experiência entram de imediato no resultado VELHA. Todo jogo da VELHA em computador tem se mostrado ruim – ou é simples demais, e existe uma forma de você vencê-lo sempre, ou é sabido e você nunca consegue vencê-lo. Em ambos os casos joga-se uma vez e perde a graça. O tabuleiro tradicional é mostrado na Figura 4.1.

Agora imagine que ao invés de um tabuleiro 3 x 3 temos um de 4 x 4.

Isto aumenta um pouco as alternativas. Agora ponha 4 destes tabuleiros, um sobre o outro, formando um cubo. Temos então 4 tabuleiros horizontais, 6 diagonais e 8 verticais, como mostra a Figura 4.2.

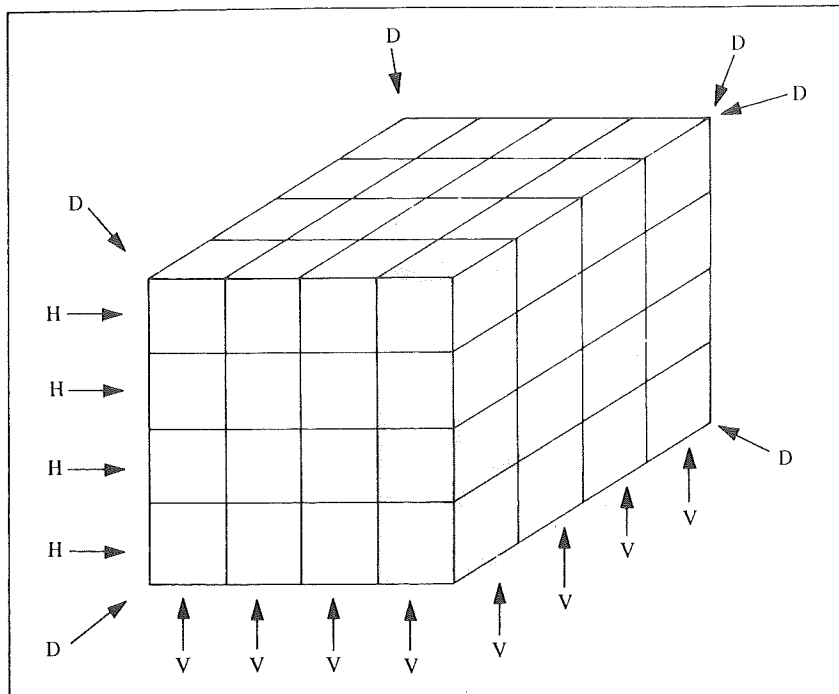


Figura 4.2
PLANOS DO CUBO

Conseguiu visualizar? — ótimo! Enquanto você não conseguir imaginar este cubo, você não está pronto para jogar. Isto é muito importante, pois apenas as fatias horizontais aparecem na tela da TV.

A primeira pergunta feita pelo sistema é sobre as pedras a serem utilizadas para identificar os jogadores (veja Figura 4.3). Normalmente se usa X e O. Escolha um deles para você. Outro símbolo qualquer pode ser utilizado, exceto o —, porém os de melhor efeito visual são O e X.

Escolhidas as pedras, pergunta-se quem começa (veja Figura 4.4). Tecla H (humano) se você deseja o privilégio. Tecla C (computador) se você não quer ser o primeiro a jogar.

Se o computador é o primeiro a jogar, ele inicia o jogo automaticamente. Se o humano é o primeiro a jogar, deverá fazer sua primeira jogada.

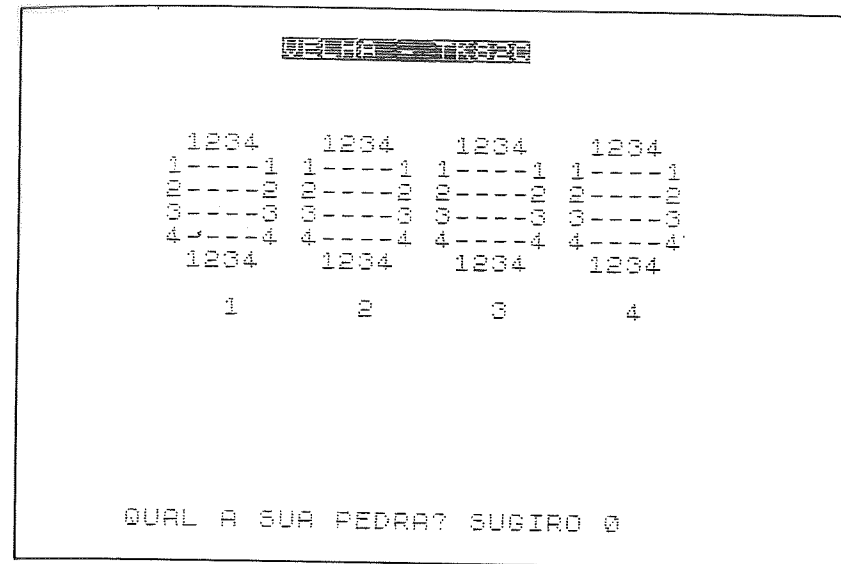


Figura 4.3
TABULEIRO DO CUBO

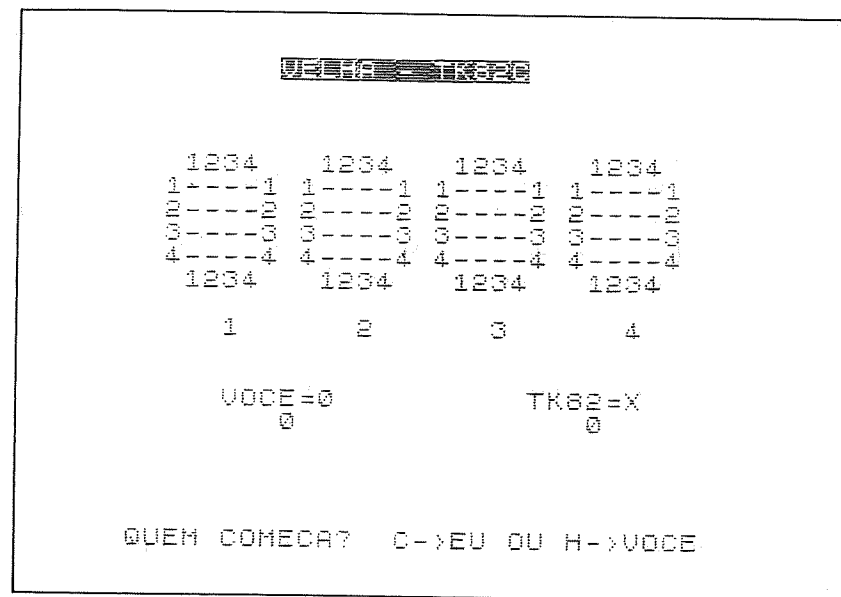


Figura 4.4
ABERTURA

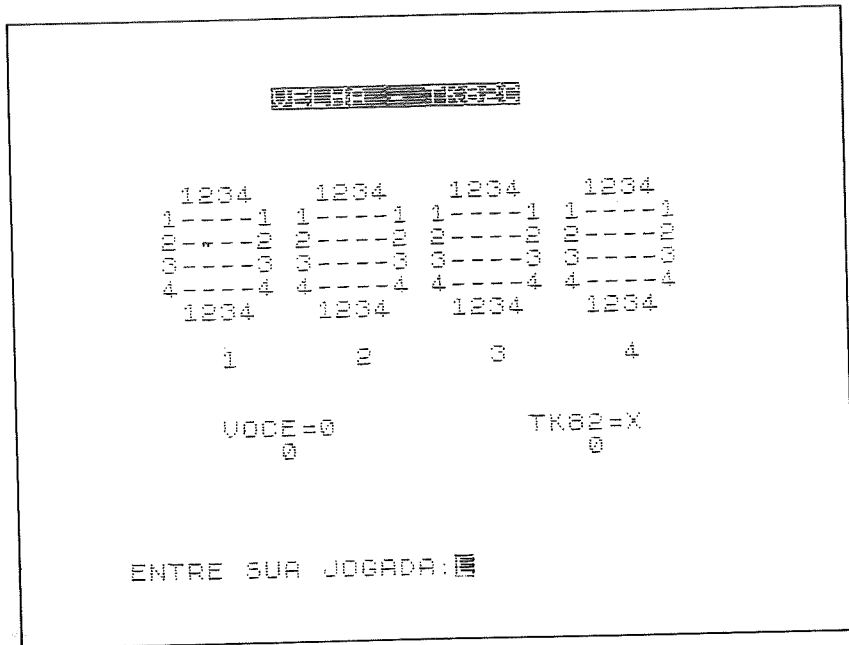


Figura 4.5 PEDIDO DE JOGADA

Cerca de 10 segundos depois a tela volta a apagar. 30 segundos depois a tela volta, agora com a mensagem indicativa de jogada dele.

As seqüências completadas são contabilizadas automaticamente.

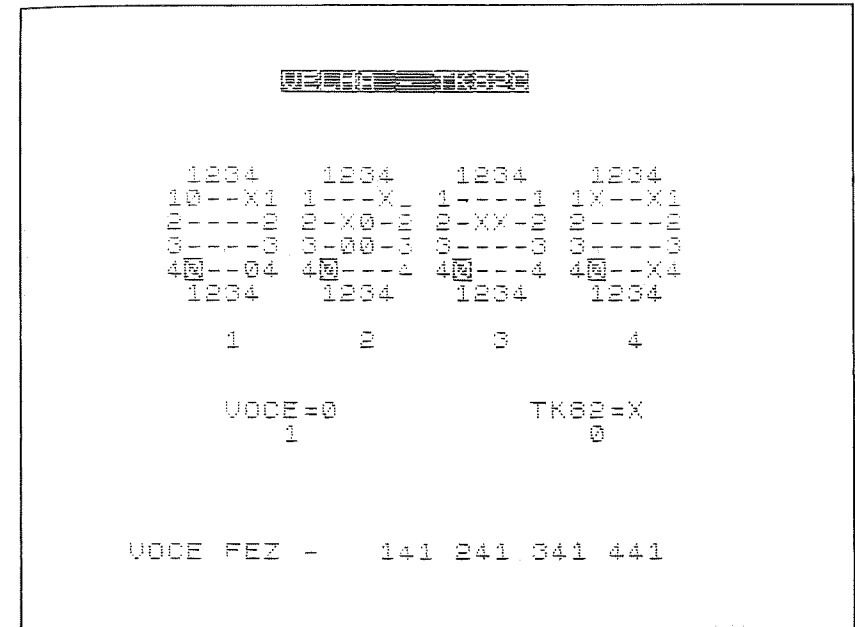


Figura 4.6 SEQUÊNCIA VENCEDORA

Para jogar, teclé as coordenadas do ponto onde você quer jogar. Observe na Figura 4.5 que, após o pedido, está o cursor (L REVERSO). Ao teclar as coordenadas, elas aparecem no local do cursor e este avança. Ao final (após o terceiro dígito), o cursor muda para E REVERSO. Teclé NEWLINE. Enquanto o cursor estiver visível, você pode teclar 0 (zero) que ele voltará uma posição. Observe que o SHIFT não precisa ser utilizado. Use esta facilidade para apagar jogadas erradas.

A primeira coordenada é identificadora do tabuleiro.

A segunda coordenada identifica a linha no tabuleiro.

A terceira coordenada identifica a coluna no tabuleiro. Os quatro tabuleiros que aparecem na tela são os horizontais.

Após o NEWLINE o cursor desaparece e, cerca de 3 segundos depois (tempo para crítica do que você teclou), aparece sua jogada no tabuleiro. A tela apaga-se por 5 segundos, voltando a seguir com a mensagem:

AGUARDE UM POUCO ENQUANTO PENSO

Como é óbvio, o vencedor é quem tiver maior número de seqüências quando não mais existir onde jogar no tabuleiro.

Toda a seqüência vencedora é posta em *reverso* na tela e as coordenadas apresentadas. Isto ajudará na visualização, como mostra a Figura 4.6.

C MANUAL DE OPERAÇÃO

Veja nos Apêndices como utilizar fitas cassette.

Para efetuar-se a carga deste jogo são necessários cerca de 3 minutos. Ocupa aproximadamente 10K.

A lógica deste programa é muito mais simples do que se pode esperar pela aparente complexidade do problema como visto no manual de utilização.

Observe na Figura 4.2 que temos na verdade 18 tabuleiros planos. Em cada tabuleiro plano temos 4 horizontais, 4 verticais e 2 diagonais, ou seja, 10 alternativas. Assim, nos 18 tabuleiros, temos 180 alternativas. Ocorre que haverá duplicidade. Exatamente, são 104 as interseções entre tabuleiros. Sobram portanto 76 alternativas distintas. Fazemos então uma lista com todas elas. Seriam 3 vetores de 76×4 posições, caso os tabuleiros fossem armazenados em uma matriz $4 \times 4 \times 4$. Se porém linearizarmos esta matriz em um vetor de 64 posições, apenas um vetor de 76×4 poderá armazenar todas as alternativas.

Veja no programa a linha 140. Ela contém todas as alternativas. Observe o uso da função VAL para substituição do uso de READ e DATA, não suportados pelo BASIC do TK82C.

O programa inicializa estes vetores na rotina 10-190, o tabuleiro e tela na rotina 200-630. A rotina 700-820 controla a seqüência de jogo através de chamadas sucessivas às rotinas 1000 (humano) e 3000 (computador). Se o computador começa, para aumentar a velocidade da resposta vai-se direto ao final da rotina do computador (4210).

A rotina 1000 – jogada humana – nada mais faz que ler as coordenadas onde o adversário vai jogar. O trecho iniciado em 1400 conta quantas pedras existem em cada uma das 76 alternativas e anota se alguma tem 4 do humano, ou 3 de qualquer um dos jogadores. No primeiro caso, o computador perdeu. Se o computador tiver uma alternativa com 3, vai jogar lá e ganhará. Se o humano tem uma com 3, vai jogar lá para não perder. Estas decisões tomadas já na rotina 3000. A rotina 2000 é de controle e atualização dos tabuleiros na tela. Recebe M e M\$ como parâmetros – M como índice no vetor X dos tabuleiros e M\$ o símbolo do jogador que jogou na posição M. Converte M para a forma de coordenadas e imprime em NPP (Next Print Position). Põe M\$ na tela, na posição correta.

A rotina 3000 – jogada do computador – toma a decisão indicada em 1400 como visto anteriormente. Se 1400 não indicar solução, vai para a rotina 4000 onde é feito o processo de avaliação.

O vetor C contém três tipos de informação básica: 0 se o ponto estiver vazio, 1 tem uma pedra do humano, 5 tem uma pedra do computador. Por que 5 e não 2 ou 1 ou outra coisa qualquer? Porque a soma de 4 uns dá 4 que é

menor que 5. Podia ser 10 ou 100. Porém 5 bastava. Se um ponto merece algum motivo para escolha, colocamos lá 0.125 – observe que 4 vezes 0.125 dá 0.5, ou seja, não faz com que uma alternativa pareça ser o que não é. Além do mais, antes de liberar para a jogada humana, todos os 0.125 são apagados de X.

Na rotina 4000, primeiro procura-se alternativa com duas pedras no computador. Observe que 2 vezes 5 são 10: Passa-se 10 como parâmetro para a rotina 5000 que analisaremos logo mais. Depois procuram-se alternativas com duas pedras do adversário. A rotina 5000 devolve qual a rotina a ser executada, na variável Z, onde recebeu o parâmetro.

Se nas duas chamadas de 5000 não foi encontrada uma decisão, então procura-se uma posição chave. Se todas estão ocupadas (1 ou 5) ou se foram analisadas por 5000 (0.125), procura-se um ponto qualquer para jogar. Se não se achar, desiste-se – vai dar VELHA.

Observe que entre chamadas de rotinas 5000, a rotina 8000 é chamada, para apagar todas as 0.125 que tenham sido colocada: em X pela rotina 5000 na primeira chamada.

Para cada alternativa, a rotina 5000 verifica se ela já tem duas pedras do mesmo parâmetro com dois vazios. Neste caso, marca os vazios como candidato (põe 0.125). Se um dos vazios já era candidato (a soma deu maior que 2 e menor que 3, ou maior que 10 e menor que 11 – depende do parâmetro Z), então temos uma tentativa de trama, se for pedra humana, ou chance de dar o golpe, se for pedra do computador. Seleciona-se a rotina 6000 para tomar esta decisão.

Se não foi selecionada a rotina 6000, então procura-se alternativa com uma pedra colocada e três vazios candidatos (0.125 vezes 3 dá 0.375) ou quatro vazios candidatos (soma igual a 0.5). Neste caso seleciona-se 7000 que escolhe um dos pontos para jogar. Dá-se preferência aos cantos.

A listagem do programa-fonte encontra-se na Figura 4.7

```

10 REM ANTONIO JULIO LOSSIO BO
TELHO
20 REM JOGO DA VELHA CUBICO
30 FAST
40 DIM X(65)
50 DIM L(75)
60 DIM H(75,4)
70 DIM B(15)
80 FOR Y=2 TO 32 STEP 2

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE
(VELHA TRIDIMENSIONAL)

```

90 LET B(Y/2)=VAL "01495204136
164162239233826422743"(Y-1 TO Y)
95 NEXT Y
100 LET A=1
110 FOR I=1 TO 76
120 LET L(I)=0
130 FOR J=1 TO 4
140 LET M(I,J)=VAL "01020304 05
060708 09101112 13141516 17181920
0 21222324 25262728 29303132 333
43536 37383940 41424344 45464748
49505152 53545556 57585960 6162
6364 65666768 69707172 737475 7677
13284581 82183456 666223854 10264
858 14384682 83183551 87833955 1
1274859 15814763 84283852 888440
58 10084450 16324864 91850913 17
812829 33874145 49535781 8288181
4 18222530 34384248 50545882 533
71115 19222731 35394347 51555883
04881216 20242832 36404443 5655
5864 01881116 17222732 33384348
49545954 13180704 26282320 43423
936 61585552 01214161 82224202 0
3234363 04244454 40372513 503820
14 51392715 52402816 01183552 85
823955 00264360 13364764 4934192
4 53382308 57422712 61463116 012
04384 18273840 04284251 13283852
"(A TO A+1)
150 LET A=A+2
160 NEXT J
170 LET A=A+1
180 NEXT I
190 LET TT=0
200 FOR I=1 TO 65
210 LET X(I)=0
220 NEXT I
230 CLS
250 LET K1=0
260 LET K2=0
270 LET K3=0
280 LET HU=0
290 LET CO=0
300 PRINT AT 1,8;"ESCAR - TK828"
305 FAST
310 FOR I=3 TO 27 STEP 7
320 PRINT AT 5,I;"1234"
330 NEXT I
340 FOR I=1 TO 4
350 FOR J=1 TO 4

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE (VELHA TRIDIMENSIONAL) (cont.)

```

360 PRINT AT I+5,(J-1)*7+2;I;"-
---";I);
370 NEXT J
380 NEXT I
390 FOR I=3 TO 27 STEP 7
400 PRINT AT 10,I;"1234"
410 PRINT AT 12,I+2;INT ((I+4)/
7);
420 NEXT I
430 SLOW
500 PRINT AT 21,0;"QUAL A SUA P
EDRA? SUGIRO 0"
510 LET H#=INKEY#
520 IF H#="" THEN GOTO 510
530 LET C#="X"
540 IF H#=0# THEN LET C#="0"
570 PRINT AT 15,6;"VOCE=";H#;AT
15,21;"TK82";C#
575 PRINT AT 16,8;HU;AT 16,24;C
0
580 PRINT AT 21,0;"QUEM COMECA?
C->EU OU H->VOCE"
600 LET A#=INKEY#
610 IF A#="C" THEN GOTO 600
620 IF A#="H" THEN GOTO 700
630 GOTO 600
700 IF TT>63 THEN GOTO 2500
710 GOSUB 1000
720 LET TT=TT+1
740 IF TT>63 THEN GOTO 2500
750 GOSUB 3000
760 LET TT=TT+1
790 GOTO 700
800 GOSUB 3200
810 GOSUB 4210
820 LET TT=TT+1
830 GOTO 700
1000 REM HUMANO JOGA
1005 GOSUB 3200
1010 PRINT AT 21,0;"ENTRE SUA JO
GADA:";
1020 LET A=17
1030 LET M=0
1040 PRINT AT 21,A;"■"
1050 LET A#=INKEY#
1060 IF A#>"4" THEN GOTO 1050
1070 IF A#="0" THEN GOTO 1250
1080 IF A#<"1" THEN GOTO 1050
1090 LET M=M#4+VAL A#-1
1100 PRINT AT 21,A;A#;
1110 LET A=A+1

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE (VELHA TRIDIMENSIONAL) (cont.)

```

1120 IF A<20 THEN GOTO 1040
1125 PRINT AT 21,A;"[E]"
1130 LET A#=INKEY#
1140 IF A#="" THEN GOTO 1250
1150 IF CODE A#(<)116 THEN GOTO 1
130
1155 LET M=M+1
1160 IF X(M)=0 THEN GOTO 1300
1170 PRINT AT 21,20;"-->OCUPADO"
1180 PAUSE 200
1190 GOSUB 2000
1200 GOTO 1000
1250 IF A<18 THEN GOTO 1040
1260 PRINT AT 21,A;" ";
1270 LET A=A-1
1280 LET M=INT (M/4)
1290 GOTO 1040
1300 LET X(M)=1
1310 LET M#=M#
1320 PRINT AT 21,A;" ";
1330 PRINT AT 21,17;" ";
1340 GOSUB 2000
1400 FAST
1410 LET K1=0
1420 LET K2=0
1430 LET K3=0
1440 FOR A=1 TO 76
1450 LET L(A)=X(M(A,1))+X(M(A,2)
)+X(M(A,3))+X(M(A,4))
1460 IF L(A)=4 THEN LET K1=A
1470 IF L(A)=15 THEN LET K2=A
1480 IF L(A)=3 THEN LET K3=A
1490 NEXT A
1500 RETURN

2000 LET G1=INT ((M-1)/16)
2010 LET J2=M-16+G1
2020 LET G2=INT ((J2-1)/4)
2030 LET G3=J2-4+G2
2040 PRINT G1+1;G2+1;G3;
2050 PRINT AT G2+8,G1*7+2+G3;M#;
2060 PAUSE 100
2070 RETURN
2500 SCROLL
2510 PRINT AT 21,0;"VOCE QUER OU
TRA? (S/N)"
2520 SCROLL
2530 LET A#=INKEY#
2540 IF A#="" THEN GOTO 2530
2550 IF A#="N" THEN GOTO 3800
2560 IF A#="S" THEN RUN
2570 PRINT AT 21,0;"TECLE S PARA

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE
(VELHA TRIDIMENSIONAL) (cont.)

```

RESPONDER SIM"
2580 SCROLL
2590 PRINT AT 21,0;"OU TECLE N P
ARA RESPONDER NAO"
2600 GOTO 2500
3000 PRINT AT 21,0;"AGUARDE UM P
OUCO ENQUANTO PENSO,";
3005 SLOW
3010 LET I=K1
3020 IF I<>0 THEN GOSUB 3300
3040 LET I=K2
3050 IF I<>0 THEN GOTO 3400
3070 LET I=K3
3080 IF I<>0 THEN GOTO 3600
3090 PAUSE 100
3100 GOTO 4000
3200 PRINT AT 21,0;" ";
;
3210 SLOW
3220 RETURN
3300 GOSUB 3200
3310 PRINT AT 21,0;"VOCE FEZ - "
;
3315 LET HU=HU+1
3320 LET M#=CHR# (CODE M#+128)
3330 FOR J=1 TO 4
3340 LET M=M(I,J)
3345 PRINT AT 21,8+J*4;" ";
3350 GOSUB 2000
3355 LET H(I,J)=65
3360 NEXT J
3370 PRINT AT 16,8;HU;AT 16,24;C
0
3380 RETURN
3400 FOR J=1 TO 4
3410 LET M=M(I,J)
3420 IF X(M)=0 THEN GOTO 3440
3430 NEXT J
3440 LET M#=C#
3450 GOSUB 3200
3460 PRINT AT 21,0;"JOGAREI EM :
";
3470 LET X(M)=5
3480 GOSUB 2000
3490 LET M#=CHR# (CODE C#+128)
3500 PAUSE 200
3505 GOSUB 3200
3510 PRINT AT 21,0;"EU FIZ :";
3515 LET CO=CO+1
3520 GOTO 3330
3600 GOSUB 3200
3610 PRINT AT 21,0;"LINDO LANCE

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE
(VELHA TRIDIMENSIONAL) (cont.)

```

- JOGAREI:");
3620 FOR J=1 TO 4
3630 LET M=M(I,J)
3640 IF X(M)=0 THEN GOTO 3660
3650 NEXT J
3660 LET M#=C#
3670 LET X(M)=5
3680 GOTO 2000
3800 STOP
3810 RUN

4000 FAST
4010 LET Z=10
4020 GOSUB 5000
4030 IF Z<>10 THEN GOTO Z
4040 GOSUB 3000
4050 LET Z=2
4060 GOSUB 5000
4070 IF Z<>2 THEN GOTO Z
4080 FOR Y=1 TO 18
4090 LET Z=0
4100 FOR I=4*Y-3 TO 4*Y
4120 LET Z=Z+X(M(I,1))+X(M(I,2))
+X(M(I,3))+X(M(I,4))
4140 NEXT I
4150 IF Z<4 THEN GOTO 4190
4160 IF Z<5 THEN GOTO 4500
4170 IF Z<9 THEN GOTO 4190
4180 IF Z<10 THEN GOTO 4500
4190 NEXT Y
4200 GOSUB 3000
4210 FOR Y=1 TO 16
4220 LET Z=B(Y)
4230 IF X(Z)=0 THEN GOTO 4400
4240 NEXT Y
4250 FOR Z=1 TO 64
4260 IF X(Z)=0 THEN GOTO 4400
4270 NEXT Z
4290 GOSUB 3200
4300 PRINT AT 21,0;"DEU VELHA..."
"
4310 GOTO 2500
4400 LET M=Z
4410 LET X(M)=5
4420 LET M#=C#
4430 GOSUB 3000
4440 GOSUB 3200
4450 PRINT AT 21,0;"JOGAREI COM
"
4460 GOTO 2000
4500 LET S=.125
4510 FOR I=4*Y-3 TO 4*Y

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE
(VELHA TRIDIMENSIONAL) (cont.)

```

4520 GOTO 7010
4530 NEXT I
4540 LET S=0
4550 GOTO 4510
5000 FOR I=1 TO 76
5020 LET L(I)=X(M(I,1))+X(M(I,2))
+X(M(I,3))+X(M(I,4))
5050 IF L(I)<Z OR L(I)>=Z+1 THEN
GOTO 5110
5070 IF L(I)>Z THEN GOTO 5200
5080 FOR J=1 TO 4
5090 IF X(M(I,J))=0 THEN LET X(M
(I,J))=.125
5100 NEXT J
5110 NEXT I
5120 LET Y=Z/2+.375
5130 FOR I=1 TO 76
5135 LET L(I)=X(M(I,1))+X(M(I,2))
+X(M(I,3))+X(M(I,4))
5140 IF L(I)=.5 OR L(I)=Y THEN G
OTO 5300
5160 NEXT I
5170 RETURN
5200 LET Z=6000
5210 RETURN
5300 LET Z=7000
5310 RETURN

6000 FOR J=1 TO 4
6010 IF X(M(I,J))<>.125 THEN GOT
O 6090
6020 GOSUB 3200
6030 IF L(I)<5 THEN GOTO 6070
6040 PRINT AT 21,0;"QUERO VER 50
VOCE SAIR DESSA"
6050 PAUSE 200
6060 GOTO 7100
6070 PRINT AT 21,0;"TA PENSANDO
QUE SOU BURRO?"
6080 GOTO 6050
6090 NEXT J
6100 GOSUB 3200
6110 PRINT AT 21,0;"VOCE FOI MUI
TO VIVO - DESISTO."
6120 GOTO 2500
7000 LET S=.125
7010 LET A=1
7020 IF I-INT (I/4)*4>1 THEN LET
A=2
7030 FOR J=A TO 5-A STEP 5-2*A
7040 IF X(M(I,J))=5 THEN GOTO 71
00

```

Figura 4.7 LISTAGEM DO PROGRAMA-FONTE
(VELHA TRIDIMENSIONAL) (cont.)


```
7050 NEXT J
7060 GOTO 4530
7100 LET M=M(I,J)
7110 LET X(M)=5
7120 LET M#=C#
7130 GOSUB 8000
7140 GOSUB 3200
7150 PRINT AT 21,0;"HI HI - VOU
COM ";
7160 GOTO 2000
8000 FOR I=1 TO 64
8010 IF X(I)=.125 THEN LET X(I)=
0
8020 NEXT I
8030 RETURN
9000 FAST
9010 CLEAR
9020 SAVE "VELHA-4"
9030 RUN
```

Figura 4.7 *LISTAGEM DO PROGRAMA-FONTE*
(VELHA TRIDIMENSIONAL) (cont.)

CAPÍTULO 5

O Desafio dos Palitos Chineses-Nim

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: NIM	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 80 <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	LLIST <input type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	COPY <input checked="" type="checkbox"/>		
SINCLAIR ZX 80 <input type="checkbox"/>	64K <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETE <input checked="" type="checkbox"/>	DSAVE <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
TIMEX 1000 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
CP-200 <input checked="" type="checkbox"/>	5 1/4" <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
CP-300 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>			
APPLY 300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>			
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
RINGO R-470 e AS-1000				
k) Observações:				

Muito embora NIM seja uma palavra de inglês arcaico, que significa *permanecer* ou *tomar no vazio*, o jogo consta na literatura como sendo originário da China. Várias pilhas de objetos são postos à disposição de dois jogadores, que selecionam uma pilha e dela retiram pelo menos um objeto, alternadamente. Perde quem pegar o último objeto.

Ao iniciar-se o programa, são apresentadas instruções como na Figura 5.1.

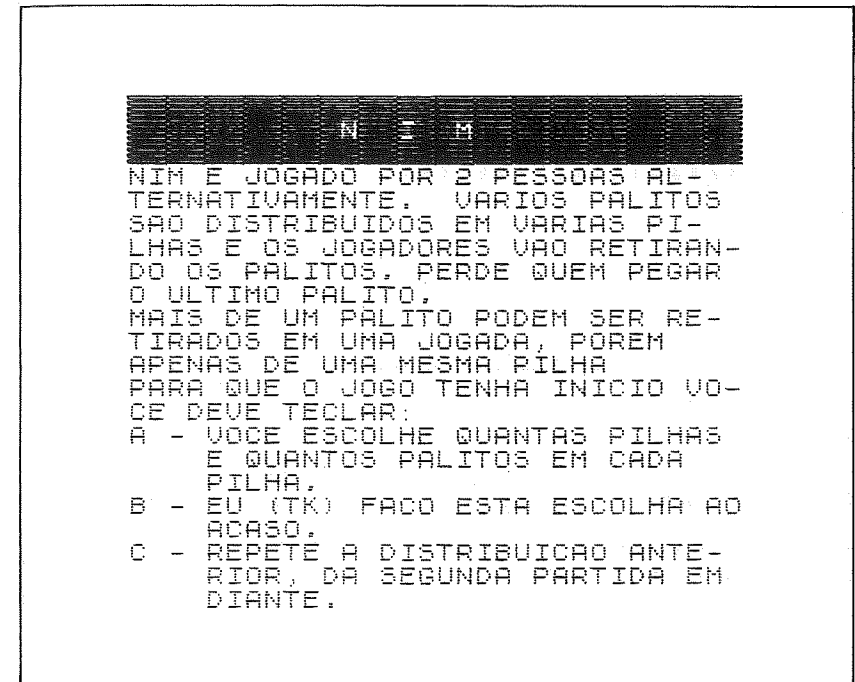


Figura 5.1 TELA DE INSTRUÇÕES

Neste ponto o sistema está em fase de inicialização. Teclé a letra (A, B ou C) da opção desejada. Observe que na primeira vez, após o RUN ou LOAD, a opção C não se aplica, pois não existe distribuição anterior.

A tela então se apaga por alguns segundos, voltando como no exemplo da Figura 5.2.

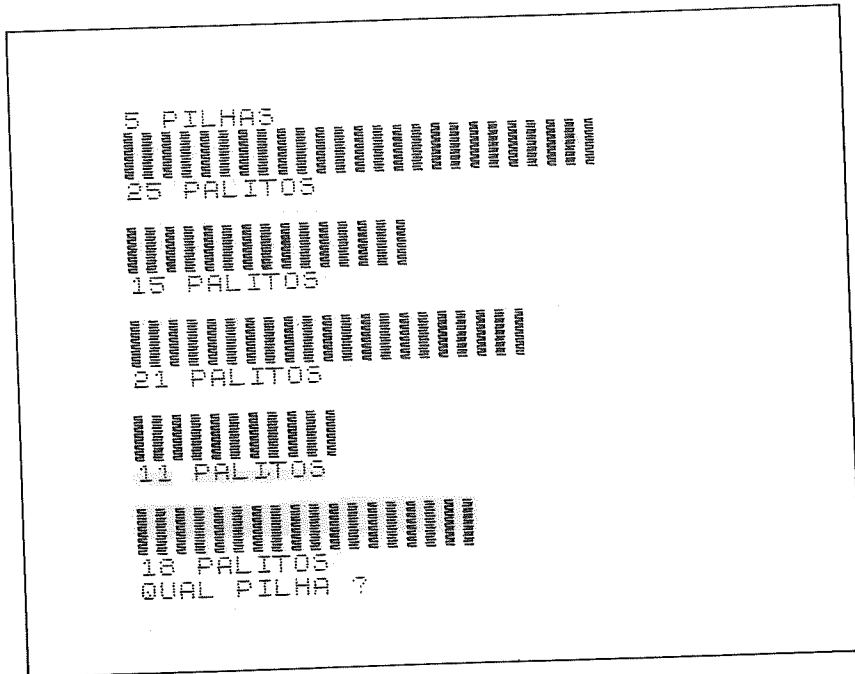


Figura 5.2 EXEMPLO DE DISTRIBUIÇÃO

É perguntado qual a pilha de onde você vai retirar objetos (que o programa chama de *palitos*). Tecele o número identificador da pilha, seguido de NEWLINE. A pilha 1 é a de cima, a 2 logo abaixo, e assim sucessivamente. Depois escolha quantos palitos você quer retirar da pilha selecionada. Tecele o número seguido de NEWLINE. O sistema então apagará a tela por alguns segundos enquanto seleciona a jogada dele, e assim sucessivamente. Ao final, ele informa quem venceu, e volta à tela inicial.

C MANUAL DE OPERAÇÃO

Aproximadamente 2 minutos são necessários para carregá-lo a partir da fita. Ocupa cerca de 5K e é auto-explicativo. Observe os Apêndices, como operar com fitas cassette.

Este jogo é extremamente conhecido e existem várias versões. Todas, porém, baseadas no mesmo algoritmo:

1. Escreve-se o número de objetos de cada pilha em binário. Por exemplo:

- Pilha 1 – 5 objetos – 101
- Pilha 2 – 3 objetos – 011
- Pilha 3 – 5 objetos – 101
- Pilha 4 – 4 objetos – 100
- Pilha 5 – 2 objetos – 010

2. Somam-se o dígitos binários por coluna. No exemplo temos 3, 2 e 3. A primeira coluna ímpar é a 1. Tem 1 nesta coluna nas pilhas 1, 3 e 4. Seleciona-se uma destas ao acaso. Digamos a 3.
3. Apaga-se o 1 desta coluna na pilha escolhida. No exemplo coluna 1 da pilha 3. O número 101 vira 001.
4. Invertem-se os dígitos de menos significância das colunas pares. O número 001 vira 011. Converte-se de volta para decimal. Este é o número de objetos a permanecer na pilha. No exemplo: 3
5. Pega-se pois a diferença, no exemplo, pega-se 2 objetos da pilha 3.
6. Se não existir soma ímpar no passo 2, sorteia-se uma pilha ao acaso e dela retira-se 1 objeto.

No livro *The Explorer Guide to the ZX81* de Mike Lord existe uma outra versão deste jogo por Lee Gentry, na qual nos inspiramos para a formatação da tela. A variante deste algoritmo por ele usada é, porém, diferente.

Na Figura 5.3 encontra-se a listagem do programa-fonte

```

1 REM NIM - ANTONIO JULIO LOS
10 BOTEHO
20 REM VERSAO TK82C
10 DIM X(5,5)
15 DIM S(5)
20 DIM L(5)
25 DIM R(5)
30 DIM Z(5)
40 DIM C(5)
50 DIM T(5)

```

Figura 5.3 LISTAGEM DO PROGRAMA-FONTE (NIM)

```

80 DIM U(5)
80 SLOW
85 LET U=0
90 CLS
100 PRINT AT 0,0;"
105 PRINT "NIM E JOGADO POR 2 P
ESSOAS AL-TERNATIVAMENTE. VAR
IOS PALITOS"
107 PRINT "SAO DISTRIBUIDOS EM
VARIAS PI-LHAS E OS JOGADORES
VAO RETIRAN-DO OS PALITOS. PERDE
QUEM PEGAR O ULTIMO PALITO."
109 PRINT "MAIS DE UM PALITO PO
DEM SER RE-TIRADOS EM UMA JOGAD
A, POREM APENAS DE UMA MESMA
PILHA"
111 PRINT "PARA QUE O JOGO TENH
A INICIO VO-CE DEVE TECLAR:"
113 PRINT "A - VOCE ESCOLHE QUAR
NTAS PILHAS E QUANTOS PALITO
S EM CADA PILHA."
115 PRINT "B - EU (TK) FAÇO EST
A ESCOLHA AO ACASO."
117 PRINT "C - REPETE A DISTRIB
UICAO ANTE-RIOR, DA SEGUNDA
PARTIDA EM DIANTE."
119 LET A#=INKEY#
120 IF A#<>"A" AND A#<>"B" AND
A#<>"C" THEN GOTO 119
125 IF A#="C" AND U=0 THEN GOTO
119
128 CLS
130 IF A#="A" THEN PRINT AT 2,0
;AT 2,0;"2,3,4,OU 5 PILHAS ?"
140 IF A#="B" THEN LET P=INT (R
ND*3+3)
150 IF A#="A" THEN INPUT P
160 IF P<2 OR P<>INT P OR P>5 T
HEN GOTO 130
165 CLS
167 LET G=0
170 PRINT AT 1,0;P;" PILHAS"
180 FOR B=1 TO P
190 IF A#="B" THEN LET L(B)=INT
(RND*16+10)
200 IF A#="A" THEN PRINT AT 21,
0;"PILHA ";B;" - POE QUANTOS(10-
25)?"

```

Figura 5.3

LISTAGEM DO PROGRAMA-FONTE (NIM) (cont.)

```

210 IF A#="A" THEN INPUT L(B)
220 IF L(B)<10 OR L(B)<>INT L(B)
OR L(B)>25 THEN GOTO 190
230 FOR C=0 TO L(B)-1
240 PRINT AT 4*B-2,C;" ";TAB C;
" "
250 NEXT C
260 PRINT L(B);" PALITOS"
270 LET S(B)=L(B)
275 LET G=G+L(B)
280 NEXT B
290 SLOW
300 PRINT AT 21,0;"QUAL PILHA ?"
310 INPUT N
320 IF N<1 OR N<>INT N OR N>P T
HEN GOTO 310
325 IF S(N)<1 THEN GOTO 310
330 PRINT AT 21,0;"QUANTOS PALI
TOS ?"
340 INPUT T
350 IF T<1 OR T<>INT T OR T>S(N)
THEN GOTO 340
360 PRINT AT 21,0;"VOCE TIROU "
;T;" DA PILHA ";N
362 GOSUB 370
365 GOTO 430
370 LET G=G-T
380 LET S(N)=S(N)-T
390 FOR B=S(N) TO S(N)+T
400 PRINT AT 4*B-2,B;" ";TAB B;
" "
410 NEXT B
415 IF S(N)<1 THEN PRINT "PILHA
VAZIA"
420 IF S(N)=1 THEN PRINT "SO UM
PALITO"
425 IF S(N)>1 THEN PRINT S(N);"
"
428 RETURN
430 IF G>2 THEN GOTO 460
440 LET G=G-1
450 GOTO 900
460 REM JOGADA DO TK
463 PRINT AT 21,0;"VOU PENSAR U
M POUCO. AGUARDE"
465 PAUSE 200
468 FAST
470 FOR I=1 TO 5
480 LET U(I)=0
490 NEXT I

```

Figura 5.3

LISTAGEM DO PROGRAMA-FONTE (NIM) (cont.)

```

495 LET Z=P
500 FOR I=1 TO P
505 IF S(I)<1 THEN LET Z=Z-1
510 LET C(I)=S(I)
515 LET R(I)=0
520 FOR E=S TO 1 STEP -1
525 LET Y=E-1
530 IF C(I)<2**Y THEN GOTO 550
540 LET R(I)=R(I)+INT (10**Y+.1)
)
550 LET C(I)=C(I)-INT (2**Y+.1)
560 NEXT E
570 FOR E=S TO 1 STEP -1
575 LET Y=E-1
580 LET X(I,E)=INT (R(I)/INT (1
0**Y+.1))
590 LET R(I)=R(I)-X(I,E)*INT (1
0**Y+.1)
600 LET U(E)=U(E)+X(I,E)
610 NEXT E
620 NEXT I
630 LET W=1
640 FOR E=S TO 1 STEP -1
645 IF W=2 THEN GOTO 690
650 IF U(E)/2=INT (U(E)/2) THEN
GOTO 710
660 LET Q=INT (P*RND+1)
665 IF X(Q,E)<>1 THEN GOTO 680
670 LET X(Q,E)=0
675 LET U=2
680 GOTO 710
690 IF U(E)/2(<)INT (U(E)/2) THE
N GOTO 710
700 LET X(Q,E)=X(Q,E)=0
710 NEXT E
712 FOR I=1 TO P
714 LET R(I)=0
716 FOR E=S TO 1 STEP -1
718 LET Y=E-1
720 LET R(I)=R(I)+X(I,E)*INT (1
0**Y+.1)
724 NEXT E
726 FOR E=S TO 1 STEP -1
728 LET Y=E-1
730 IF R(I)<INT (10**Y+.1) THEN
GOTO 738
732 LET C(I)=C(I)+INT (2**Y+.1)
734 LET R(I)=R(I)-INT (10**Y+.1)
)
736 NEXT E
740 NEXT I
742 IF W=2 THEN GOTO 1000

```

Figura 5.3

LISTAGEM DO PROGRAMA - FONTE (NIM) (cont.).

```

744 LET Q=INT (P*RND+1)
746 IF C(Q)=0 THEN GOTO 744
747 PRINT AT 20,20;"AA"
748 LET C(Q)=C(Q)-1
750 LET D=S(Q)-C(Q)
755 IF D=0 THEN GOTO 744
756 SLOW
760 PRINT AT 21,0;"EU PEGUEI ";
D;" DA PILHA "/Q;" "
810 LET N=0
820 LET T=0
830 GOSUB 370
840 IF G>1 THEN GOTO 300

900 CLS
910 PRINT AT 8,9;"*****
":AT 10,9;"*** VENCI ***";AT 10,
12;(" PERDI " AND G(>1);AT 12,9;
"*****")
920 PAUSE 300
940 GOTO 90
1000 IF Z>1 THEN GOTO 750
1010 LET C(Q)=1
1020 GOTO 750
9000 FAST
9010 CLEAR
9020 SAVE "NIM"
9030 GOTO 10

```

Figura 5.3

LISTAGEM DO PROGRAMA - FONTE (NIM) (cont.)

CAPÍTULO **6**

**Ligações
Descendentes**

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: LIGAÇÕES DESCENDENTES	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
SINCLAIR ZX 8Ø <input type="checkbox"/>	64K <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	DMLOAD <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
CP-2ØØ <input checked="" type="checkbox"/>	5 1/4" <input type="checkbox"/>			
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>			
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>			
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
Ringo R-47Ø e AS - 1ØØØ				
k) Observações:				

Existe um brinquedo, fabricado pela Estrela, com o nome LIQ-4. Foi inspirado nele que nasceu este jogo.

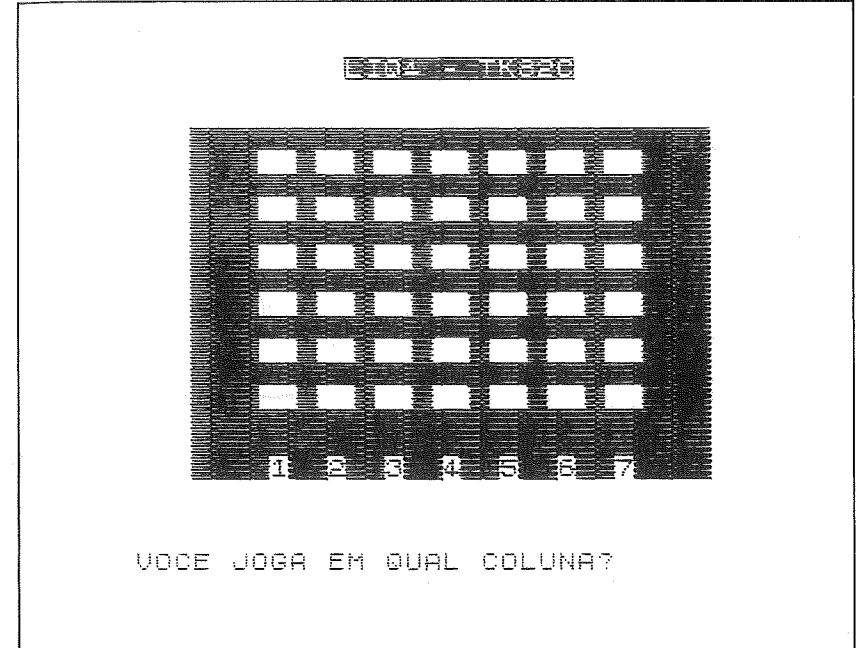


Figura 6.1 TABULEIRO

O quadro tem 7 colunas por 6 linhas (ver Figura 6.1). Jogam dois jogadores. A jogada consiste na escolha de uma coluna. Nela solta-se uma pedra que desce até a primeira linha disponível. Alternadamente vai-se jogando até que um jogador consiga colocar 4 pedras em posições consecutivas, na horizontal (linha) ou vertical (coluna) ou diagonal.

Após o **RUN NEWLINE** ou **LOAD** " " **NEWLINE** (veja Apêndices) o computador desenha o quadro do LIQ-4 na tela e pergunta em qual coluna você vai jogar. Nesta versão é sempre você que começa. Tecele o número da coluna desejada. Nesse momento um O aparece no topo da coluna e inicia sua queda. Pára na última linha vazia e a tela se apaga por cerca de 13 segundos,

quando então volta a aparecer, contendo a informação de qual a jogada do computador e a seguir um X inicia sua queda de forma semelhante ao seu O. Agora é a sua vez novamente. O processo se repete até que você desista, perca ou empate. Se o empate for determinado quando na jogada do computador, ele a detecta e termina. Se você desiste ou detecta o empate, pressione a tecla BREAK.

C

MANUAL DE OPERAÇÃO

Veja nos Apêndices como operar com fitas cassette. O tempo de carga deste programa é de 2 minutos, e ele ocupa cerca de 5K.

D

MANUAL TÉCNICO

A inicialização e impressão da tela é feita até a linha 290. As variáveis H\$, B\$ e P\$ contêm respectivamente os símbolos a serem utilizados para pedras do adversário (humano), pedras do computador, espaços vazios e pretos (colunas e bases).

O tabuleiro fica armazenado no vetor Q\$ (63,3) simulando uma matriz de 9 colunas por 7 linhas. Isto se deve ao fato de armazenarmos as duas colunas externas e a linha de base.

A rotina 5000 desenha a queda de uma pedra. Recebe a coluna real na variável A e a pedra a cair em S\$.

A rotina 7000 simula a queda de uma pedra. Recebe em A a coluna real, em S\$ a pedra a cair e em K retorna a posição em Q\$ onde foi parar a pedra. Esta informação é importante, pois esta rotina é usada para simular jogadas que depois devem ser desfeitas.

Assim, em 3000 lê-se a jogada do humano. Usa-se a rotina 5000 para fazer a pedra cair. Usa-se então a rotina 6000 para avaliação da jogada. Em A vai a coluna humana. Na posição A do vetor C vem a avaliação e em V retorna 0 (zero) se avaliado ou 1 se houve vitória de quem usou a pedra informada em S\$. No caso de vitória, S\$ é posto em reverso e a jogada apagada e refeita com a rotina 5000. Isto dará ao humano melhor visibilidade da ligação vitoriosa.

A jogada do computador é escolhida na seguinte forma:

1. 440 a 510

Para cada coluna, se possível de jogar, simula-se a jogada usando-se a rotina 7000, avalia-se com a rotina 6000 e apaga a jogada. Se a avaliação indicar vitória — joga-se e acabou.

2. 520 a 610

Semelhante ao caso anterior, só que simulando a futura jogada do humano. Se ele vencesse de alguma forma — joga-se lá para fechar-lhe a porta.

Consolida a avaliação dando peso 4 à simulação da jogada do computador e peso 6 à simulação da jogada do humano.

3. 620 a 1060

Seleciona a coluna de melhor avaliação.

Simula jogar nela.

Simula o humano nela também, após o computador.

Se o humano ganhar, zera a avaliação e repete o processo 3.

Se não, apaga as duas simulações e joga.

Observe que se a melhor avaliação for zero é porque estamos sem ter onde jogar — ou empatamos ou vamos perder no próximo lance.

A rotina 8000 é de depuração — apague o REM, que os valores de avaliação aparecerão na base da tela. A rotina 8500 é complementação e dá continuidade a jogos interrompidos para depuração. Estas rotinas permaneceram para ajudar a quem queira melhorar este jogo.

Isto parece factível, pois o tempo de resposta de 13 segundos para examinar um lance à frente é muito bom. Código para olhar dois lances à frente poderia ser incluído.

Na Figura 6.2 encontra-se a listagem do programa-fonte.

```

10 REM ANTONIO JULIO LOSSIO BO
TELHO
20 REM ██████████ ██████████
25 FAST
30 DIM Q$(63,3)
32 DIM H$(3)
33 LET H#=" O "
35 DIM C$(3)
38 LET C#=" X "
40 DIM O(7)

```

Figura 6.2

LISTAGEM DO PROGRAMA-FONTE (LIGAÇÕES DESCENDENTES)


```

43 DIM B$(3)
45 LET B$=" "
47 DIM P$(3)
49 LET P$=" "
50 LET U=0
55 CLS
60 FOR I=1 TO 7
70 FOR J=1 TO 9
80 LET U=U+1
100 IF I=7 OR J=1 OR J=9 THEN G
OTO 130
110 LET Q$(U)=B$
120 GOTO 140
130 LET Q$(U)=P$
140 LET L=I#2+2
150 LET C=J#3
160 PRINT AT L,C;Q$(U)
170 PRINT AT L+1,C;P$
180 NEXT J
190 NEXT I
200 PRINT AT 0,11;"
210 FOR I=1 TO 7
220 LET C=I#3+4
230 PRINT AT 17,C;I
240 NEXT I
250 FOR I=1 TO 9
260 LET C=I#3
270 PRINT AT 3,C;P$
280 NEXT I
290 SLOW
300 PRINT AT 21,0;"VOCE JOGA EM
QUAL COLUMNA?
310 LET A$=INKEY$
320 IF A$<"1" OR A$>"7" THEN GO
TO 310
330 LET A=VAL A$+1
340 IF Q$(A)=B$ THEN GOTO 380
350 PRINT AT 21,0;"NESTA COLUMNA
NAO PODE SER.
360 PAUSE 200
370 GOTO 300
380 LET S#=H$
390 GOSUB 5000
395 LET O=J
400 FAST
410 LET A=A-1
420 GOSUB 6000
430 IF U<>0 THEN GOTO 6190
432 FOR I=2 TO 8
434 IF Q$(I)=B$ THEN GOTO 440

```

Figura 6.2. LISTAGEM DO PROGRAMA-FONTE
(LIGAÇÕES DESCENDENTES) (cont.)

```

436 NEXT I
437 PRINT AT 21,0;"EMPATAMOS ES
TA.
438 STOP
439 GOTO 438
440 LET S#=C#
450 FOR A=1 TO 7
455 LET C(A)=0
460 IF Q$(A+1)<>B$ THEN GOTO 51
0
470 GOSUB 7000
480 GOSUB 5000
490 LET Q$(K)=B$
500 IF U<>0 THEN GOTO 6190
510 NEXT A
520 LET S#=H$
530 FOR A=1 TO 7
540 LET E=C(A)
545 LET C(A)=0
550 IF Q$(A+1)<>B$ THEN GOTO 51
0
560 GOSUB 7000
570 GOSUB 6000
580 LET Q$(K)=B$
590 IF U<>0 THEN GOTO 1000
600 LET C(A)=E*4+C(A)*6
610 NEXT A
620 LET A=1
630 FOR I=2 TO 7
640 IF C(A)>C(I) THEN GOTO 680
650 IF C(A)<C(I) THEN GOTO 670
660 IF RND<.5 THEN GOTO 680
670 LET A=I
680 NEXT I
685 GOSUB 8000
690 IF C(A)<1 THEN GOTO 850
700 LET S#=C#
710 GOSUB 7000
720 LET X=K
730 IF X<10 THEN GOTO 990
740 LET S#=H$
750 GOSUB 7000
760 LET H=K
770 GOSUB 6000
780 LET Q$(X)=B$
790 LET Q$(H)=B$
800 IF U=0 THEN GOTO 1000
810 LET C(A)=0
820 FOR I=1 TO 7
830 IF C(I)>0 THEN GOTO 820
840 NEXT I
850 PRINT AT 21,0;"DOU ESTA POR

```

Figura 6.2 LISTAGEM DO PROGRAMA-FONTE
(LIGAÇÕES DESCENDENTES) (cont.)

```

PERDIDA
860 STOP
870 GOTO 860

990 LET Q$(X)=B$
1000 LET A=A+1
1010 PRINT AT 21,0;"JOGAREI NA O
OLUNA ";A-1);
1020 LET S#=C#
1030 PAUSE 100
1040 GOSUB 5000
1050 GOTO 300
5000 SLOW
5005 LET S=(CODE S$(2)>128)
5006 LET S=((S=0)-(S=1))*128
5007 LET Z#="█"+CHR#(CODE S$(2)
+S)+"█"
5010 FOR I=A TO A+64 STEP 8
5020 IF Q$(I)<>B# THEN GOTO 5110
5030 LET J=I
5040 LET L=INT(I/8)
5050 LET C=(I-L*8)*3
5060 LET L=L*2+3
5070 PRINT AT L,C;Z#
5080 PRINT AT L,C;P#
5090 LET L=L+1
5095 PRINT AT L,C;S#
5095 PRINT AT L,C;Q$(I)
5100 NEXT I
5110 LET Q$(J)=S#
5120 PRINT AT L,C;S#
5130 PAUSE 100
5140 RETURN
6000 LET V=0
6010 FOR I=A+1 TO A+55 STEP 9
6060 IF Q$(I)=S# THEN GOTO 6090
6070 NEXT I
6080 RETURN
6090 LET B=0
6095 LET M=(I>3)+(I>2)+(I>1)
6100 FOR J=I-1 TO I-M STEP -1
6110 IF Q$(J)<>S# THEN GOTO 6140
6120 LET B=B+1
6130 NEXT J
6140 FOR J=I+1 TO I+3
6150 IF Q$(J)<>S# THEN GOTO 6180
6160 LET B=B+1
6170 NEXT J
6180 IF B<3 THEN GOTO 6300
6185 GOTO 8500
6190 PRINT AT 21,0;"ACABOU."
6200 LET A=A+1

```

Figura 6.2 LISTAGEM DO PROGRAMA-FONTE
(LIGAÇÕES DESCENDENTES) (cont.)

```

6210 IF S#=H# THEN GOTO 6250
6220 PRINT "VOCE PERDEU."
"
6230 LET S#="███"
6240 GOTO 6280
6250 PRINT "VOCE VENCEU."
"
6260 LET S#="███"
6270 LET Q$(0)=B$
6280 GOSUB 5000
6290 STOP
6295 GOTO 6290

6300 FOR W=8 TO 10
6310 LET C(A)=(B>0)+((B>1)*3)+C(
A)
6320 LET B=0
6330 FOR J=I+W TO I+3*W STEP W
6340 IF Q$(J)<>S# THEN GOTO 6370
6350 LET B=B+1
6360 NEXT J
6370 FOR J=I-W TO I-3*W STEP -W
6380 IF J<1 THEN GOTO 6410
6390 IF Q$(J)<>S# THEN GOTO 6420
6400 LET B=B+1
6410 NEXT J
6420 IF B>2 THEN GOTO 6500
6430 NEXT W
6440 LET C(A)=(B>0)+((B>1)*3)+C(
A)
6450 RETURN
6500 LET V=1
6510 RETURN
7000 FOR I=A+1 TO A+55 STEP 9
7010 IF Q$(I)<>B# THEN GOTO 7040
7020 LET K=I
7030 NEXT I
7040 LET Q$(K)=S#
7050 RETURN
8000 RETURN
8005 PRINT AT 18,0;" "
8010 FOR Z=1 TO 7
8020 PRINT "*" ;C(Z);
8030 NEXT Z
8040 PRINT " "
8050 RETURN
8500 LET W=0
8505 FOR F=1 TO 7
8510 FOR G=1 TO 9
8520 LET W=W+1
8540 LET L=F*2+2
8550 LET C=G*3

```

Figura 6.2 LISTAGEM DO PROGRAMA-FONTE
(LIGAÇÕES DESCENDENTES) (cont.)

```
8560 PRINT AT L,C;Q$(W)
8570 PRINT AT L+1,C;P#
8580 NEXT G
8590 NEXT F
8600 GOSUB 8000
8610 PRINT AT 0,11; "LIG4 - F820"
"
8620 FOR F=1 TO 7
8630 LET C=F*3+4
8640 PRINT AT 17,C;F
8650 NEXT F
8660 FOR F=1 TO 9
8670 LET C=F*3
8680 PRINT AT 3,C;P#
8690 NEXT F
8700 CONT
9000 FAST
9010 CLEAR
9020 SAVE "LIG-4"
9030 RUN
```

Figura 6.2 LISTAGEM DO PROGRAMA-FONTE
(LIGAÇÕES DESCENDENTES) (cont.)

CAPÍTULO 7

*Será possível
Sobrar Só 1?*

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:
Programa <input checked="" type="checkbox"/>	<u>RESTA-1</u>	<u>1</u>	<u>Ø</u>	<u>22/10/83</u>
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
SINCLAIR ZX 8Ø <input type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>		
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-2ØØ <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
<u>RINGO R-47Ø e</u>				
<u>AS-1ØØØ</u>				
k) Observações: _____				

Este programa é uma simulação do jogo RESTA-1, sem seu maior inconveniente: pôr as pedras nos lugares – o computador faz isso por você. Aliás, é praticamente só isto que ele faz. O que mais que ele faz? Verifica se você está jogando dentro das regras, move pedras e guarda as pedras comidas.

Na Figura 7.1 temos o tabuleiro onde o RESTA-1 deve ser jogado. O jogo consiste em, segundo as regras descritas a seguir, retirar o máximo possível de pedras (representadas por O na Figura 7.1) do tabuleiro. No melhor dos casos: deixar apenas uma.

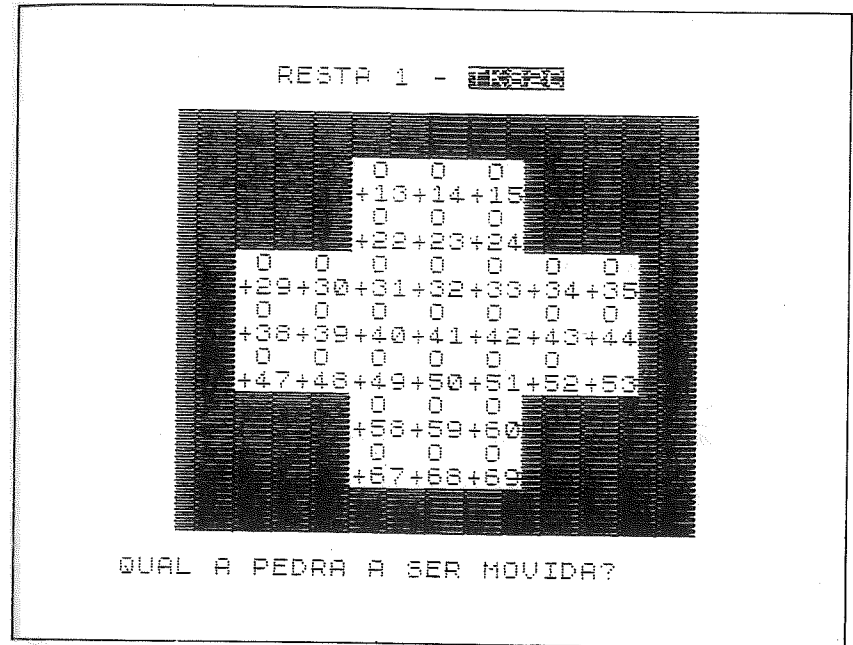


Figura 7.1

TABULEIRO DO RESTA-1

Neste tabuleiro em forma de cruz deve-se jogar. Apenas uma pessoa joga. O movimento de uma pedra é aos saltos. Ou seja, você somente pode mover uma pedra se ela, saltando outra pedra, cair em um buraco vazio. A pedra saltada sai do tabuleiro. O objetivo é retirar todas as pedras do tabuleiro. Obviamente, quando só restar uma pedra, esta não poderá ser retirada, pois

não existirá outra para saltar sobre ela. Daí o nome do jogo, que consta na literatura como sendo de origem européia.

Você pode se classificar como *muito bom* se conseguir deixar *menos de 5* pedras no tabuleiro. *Menos de 3*, você é o *máximo*.

A primeira pedra é retirada ao acaso pelo computador.

Para cada início, existe pelo menos uma seqüência de movimentos que deixa apenas uma pedra no tabuleiro.

Desenhado o tabuleiro, o computador pergunta qual a pedra a ser movida. Escolha e tecele o número que se encontra sob ela, seguido de NEWLINE. A seguir vem a pergunta de para onde a pedra deve ser movida. Tecele o número de uma posição vazia seguida de NEWLINE. Observe que o movimento deve ser tal que uma outra pedra tenha sido saltada e que o movimento tenha sido na horizontal ou vertical — nunca em diagonal.

Se por algum motivo você escolheu mover uma pedra que não pode ser movida e o sistema ficar perguntando sem parar para onde você quer que ela seja movida, responda uma posição vazia qualquer que o programa volta à pergunta de seleção de pedra.

O computador determina o momento em que o jogo termina — falta de pedras que possam ser movimentadas.

Para uma segunda partida, entre **RUN NEWLINE**.

C

MANUAL DE OPERAÇÃO

Um minuto e 20 segundos é o tempo necessário para a carga deste programa de 3K.

Veja nos Apêndices como operar fitas cassete.

D

MANUAL TÉCNICO

O tabuleiro é armazenado em um vetor de 81 posições de 3 caracteres onde simulamos uma matriz 9 x 9. O número posto no tabuleiro é exatamente o índice neste vetor.

Um movimento válido na horizontal exige que a diferença entre os índices seja 2. Na vertical 18. O ponto da média deve estar ocupado. O ponto de destino vazio.

A dúvida sobre circular de um lado do tabuleiro para o outro pode ser respondida facilmente: O tabuleiro real é 7 x 7, donde a diferença de um extremo ao outro na linha inferior é maior que 2 na horizontal e menor que 18 na vertical. Estes pontos extras na matriz são preenchidos com *preto* (*reverso de espaço*) para formatar a tela e portanto não usáveis para movimentos.

Na Figura 7.2 encontra-se a listagem do programa-fonte.

```
10 REM ANTONIO JULIO LOSSIO BO
TEHO
20 REM RESTA 1 - ████████
30 DIM Q$(81,3)
40 FAST
50 LET W=0
60 FOR I=1 TO 9
70 FOR J=1 TO 9
80 LET W=W+1
90 IF I=1 OR J=1 OR I=9 OR J=9
THEN GOTO 120
100 IF I>=4 AND I<=6 THEN GOTO
140
110 IF J>=4 AND J<=6 THEN GOTO
140
120 LET Q$(W) = "███"
130 GOTO 150
140 LET Q$(W) = "  "
150 NEXT J
160 NEXT I
170 LET W=AND#57+13
180 IF Q$(W) = "███" THEN GOTO 17
0
190 LET Q$(W) = "

200 CLS
210 PRINT AT 0,8;"RESTA 1 - ██████
███"
220 LET W=0
230 FOR I=1 TO 9
240 FOR J=1 TO 9
250 LET W=W+1
260 LET C=J*3
270 LET L=I*2
280 PRINT AT L,C;Q$(W)
290 IF Q$(W) = "███" THEN GOTO 32
0
300 PRINT AT L+1,C;"+";W
310 GOTO 330
320 PRINT AT L+1,C;Q$(W)
330 NEXT J
```

Figura 7.2

LISTAGEM DO PROGRAMA-FONTE (RESTA-1)

```

340 NEXT I
350 PRINT AT 21,0;"QUAL A PEDRA
A SER MOVIDA?"
355 SLOW
360 INPUT N
370 IF N<=69 AND N>=13 THEN GOT
O 410
380 PRINT AT 21,0;"ACHO QUE VOC
E NAO ENTENDEU "
390 PAUSE 200
400 GOTO 350
410 IF Q$(N)<>" 0 " THEN GOTO 3
80
420 PRINT AT 21,0;"MOVE PARA ON
DE?"
430 INPUT M
440 IF M<=69 AND M>=13 THEN GOT
O 480
450 PRINT AT 21,0;"PUXA VIDA. T
A DIFICIL HEIM?"
460 PAUSE 200
470 GOTO 420
480 IF Q$(M)<>" " THEN GOTO 4
50
490 LET Z=ABS (M-N)
500 IF Z=2 OR Z=18 THEN GOTO 54
0
510 PRINT AT 21,0;"???? QUAL E
CARA? ASSIM NAO, "
520 PAUSE 200
530 GOTO 350
540 LET U=(M+N)/2
550 LET Q$(U)=" "
560 GOSUB 2000
570 LET U=N
580 LET Q$(U)=" "
590 GOSUB 2000
600 LET U=M
610 LET Q$(U)=" 0 "
620 GOSUB 2000

670 FAST
680 LET M=0.
690 FOR N=13 TO 69
700 IF Q$(N)<>" 0 " THEN GOTO 7
80
710 LET M=M+1
720 FOR U=-1 TO 1 STEP 2
730 IF Q$(N+U)=" 0 " AND Q$(N-U
)=" " THEN GOTO 350
740 NEXT U
750 FOR U=-9 TO 9 STEP 18

```

Figura 7.2

LISTAGEM DO PROGRAMA-FONTE (RESTA-1) (cont.)

```

760 IF Q$(N+U)=" 0 " AND Q$(N-U
)=" " THEN GOTO 350
770 NEXT U
780 NEXT N
790 IF M<7 THEN GOTO 820
800 PRINT AT 21,0;"PUXA COMO VO
CE E FRACO. BOBAO "
810 GOTO 890
820 IF M<5 THEN GOTO 850
830 PRINT AT 21,0;"TREINANDO TA
LVEZ APRENDA "
840 GOTO 890
850 IF M=1 THEN GOTO 880
860 PRINT AT 21,0;"MUITO BOM -
NA PROXIMA ...
870 GOTO 890
880 PRINT AT 21,0;"PARABENS. VO
CE E O BOM.
890 STOP
900 RUN
2000 LET L=INT (U/9)
2010 LET C=(U-L*9)*3
2020 LET L=(L+1)*2
2030 PRINT AT L,C;Q$(U)
2040 RETURN
9000 FAST
9010 CLEAR
9020 SAVE "RESTA-1"
9030 GOTO 10

```

Figura 7.2

LISTAGEM DO PROGRAMA-FONTE (RESTA-1) (cont.)

CAPÍTULO 8

***Descubra
a Senha,
se For Capaz***

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: <u>SENHA</u>	c) Versão: <u>1</u>	d) Revisão: <u>φ</u>	e) Data: <u>22/10/85</u>
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 80 <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 85 <input type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
SINCLAIR ZX 80 <input type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>		
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1000 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-200 <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-300 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
<u>RINGO R-470 e</u> <u>AS-1000</u>				
k) Observações:				

Existem vários jogos com o nome de SENHA, e que são completamente diferentes. Além do nome, eles têm em comum o fato de que alguém tem que descobrir algo. Alguns trabalham com números, outros com letras, objetos e até cores. O mais conhecido deles é fabricado pela GROW em três formas – *Minissenha*, *Senha* e *Supersenha*. Trata-se de um tabuleiro cheio de furos e um espaço com proteção. Nessa proteção um dos jogadores põe uma seqüência de pinos coloridos. O outro jogador deverá descobrir essa seqüência colocando pinos coloridos nos furos. O primeiro responde com pinos pretos e brancos indicando quantas cores certas estão no lugar certo e quantas cores certas estão no lugar errado. O número de cores erradas não é informado, mas deduzido de imediato. Ao fim do jogo, contam-se quantos palpites foram necessários para que a *senha* fosse descoberta. Invertem-se os jogadores de posição e joga-se novamente. Vence aquele que precisou de um número menor de lances para descobrir a *senha* do outro.

A idéia original era de simplesmente simular o tabuleiro. Porém, concluímos que o algoritmo para descobrir a *senha* não era dos mais complicados de ser programado e nasceu uma versão que descobria *senhas*. Nada mais natural que juntar os dois programas. Nasceu a SENHA-DUPLA.

Escolhemos números e não cores ou letras para a formação da SENHA, pura e simplesmente por ser de armazenagem mais simples em microcomputador.

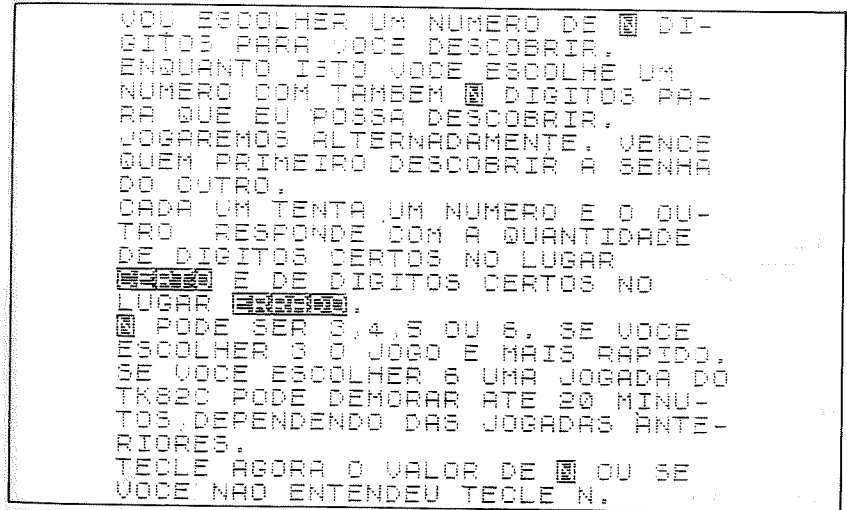


Figura 8.1. TELA DE INSTRUÇÕES

Assim, este jogo tornou-se uma disputa sobre quem consegue descobrir a SENHA do outro mais rapidamente. Você tem a vantagem de jogar primeiro.

Iniciado o processo (RUN NEWLINE ou após LOAD) a tela da Figura 8.1 é apresentada pedindo o grau de dificuldades do jogador com 3, 4, 5 ou 6 dígitos na SENHA. Com 3 dígitos fica bem rápido e é bem fácil descobrir a SENHA. Com 6 dígitos o grau de dificuldade é bem grande e o tempo de resposta após 2 ou 3 lances fica muito longo. Nas primeiras vezes selecione 3 para ganhar experiência. Depois aumente gradativamente.

Escolhido o número de dígitos, tecle-o. Automaticamente a tela mudará veja Figura 8.2.

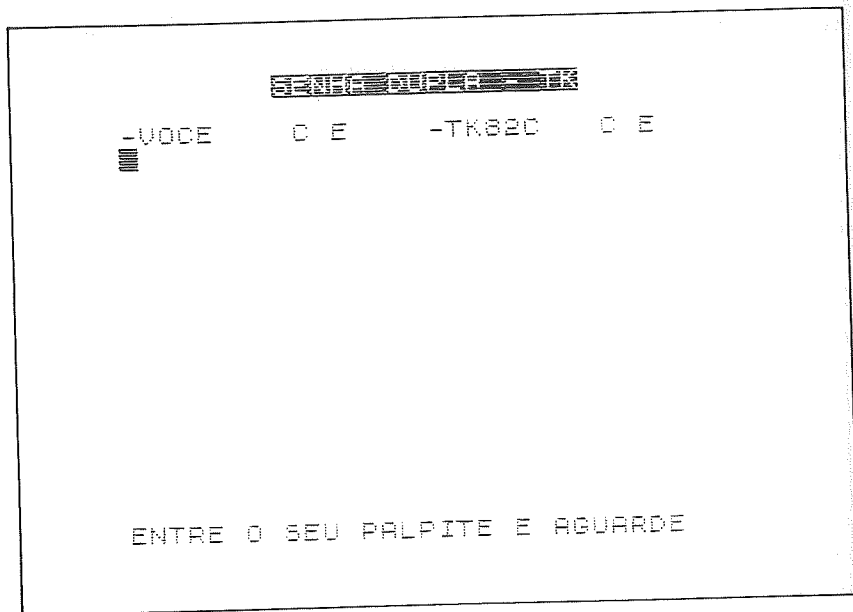


Figura 8.2
TELA INICIAL

Neste ponto escolha sua senha e anote-a em uma folha de papel. Na sua senha, bem como em seus palpites, não poderão ocorrer dígitos repetidos, isto é, 101 não é válido. Ao pedido (veja Figura 8.2)

ENTRE O SEU PALPITE E AGUARDE

tecle os dígitos de seu palpite. Eles aparecerão diretamente na parte superior da tela, abaixo da coluna VOCE. Após entrar todos os dígitos, a linha inferior mudará. Veja Figura 8.3.

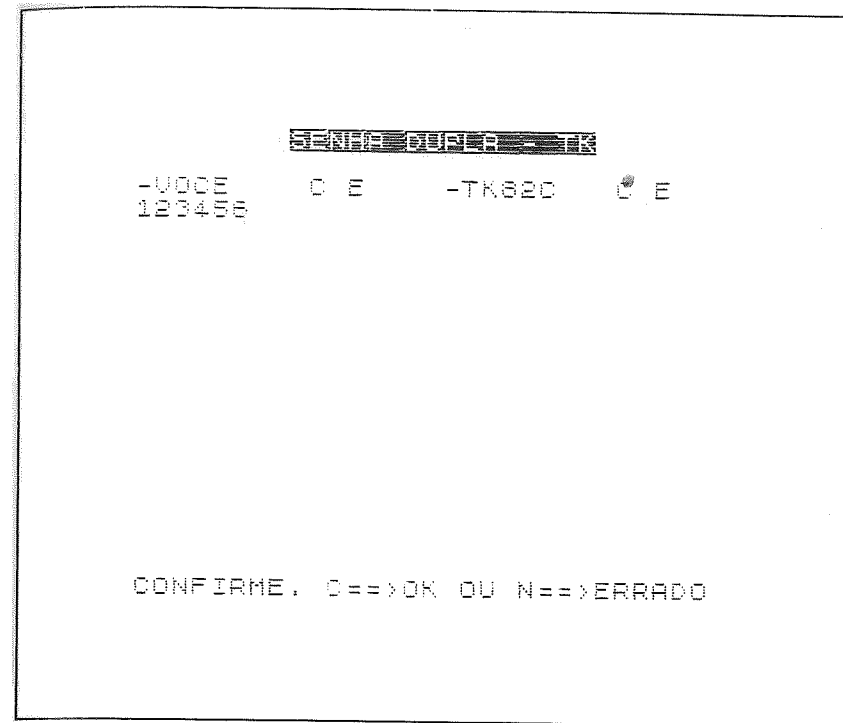


Figura 8.3
TELA COM PALPITES

Isto permite que você modifique seu palpite. Se você teclar N o programa apagará seu palpite e voltará à pergunta anterior.

Se você teclar a letra C, o programa apaga a tela e em poucos segundos escreve nas colunas C e E o número de dígitos certos nos lugares certos, e o número de dígitos certos porém nos lugares errados existentes no seu palpite. Apaga a tela novamente, para formar o palpite dele.

O primeiro palpite dele é rápido. O segundo já demora alguns segundos. O terceiro demora mais ainda, e assim sucessivamente. O número de dígitos na senha também influi no tempo de resposta. Não se assuste se com 6 dígitos o 6º palpite demorar 20 minutos. O computador joga para ganhar.

Determinado o palpite dele, a tela volta – veja Figura 8.4 – o palpite dele aparece na coluna TK82C. Vem então a pergunta:

MEU PALPITE – QUAL O MEU SCORE?

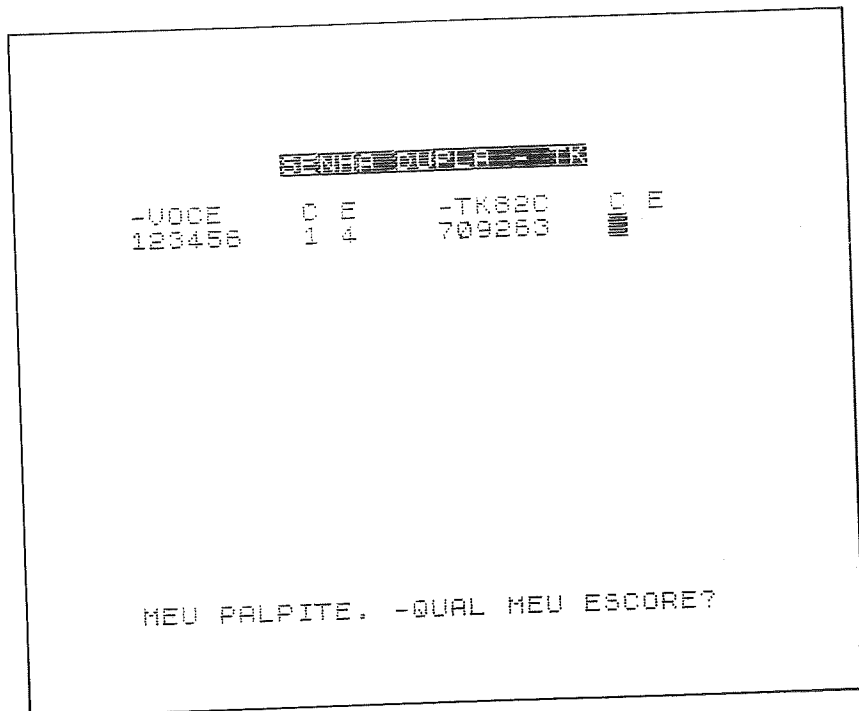


Figura 8.4 PALPITE DO COMPUTADOR

Compare o palpite dele com a sua senha. Conte quantos dígitos estão certos e na mesma posição. Agora conte quantos estão certos mas em posição errada. Tecele o número. Por exemplo:

3478 — sua senha
 2480 — o palpite dele
 errado (3, 7, 8)
 certo (4, 8)

deverá ter como resposta 1 C (o 4) e 1 E (o 8).

Outro exemplo:

3478 — senha
 3870 — palpite dele
 errado (3, 7, 8)
 certo (8)

Resposta — 2 C (o 3 e o 7) e 1 E (o 8).

Neste ponto é novamente pedida a confirmação. Confirme sua resposta, pois se teclar o 'C' não poderá mais voltar atrás. Se teclar N sua resposta será apagada e a pergunta anterior volta. Se responder com C vem a pergunta:

ENTRE SEU PALPITE E AGUARDE

E o processo todo se repete até que um descubra a senha do outro. Vence quem descobrir primeiro, como na Figura 8.5.

Vale salientar que é muito pouco freqüente o computador gastar mais de 6 (seis) palpites sem descobrir a senha. É bem verdade que é raro acertar com menos de 4.

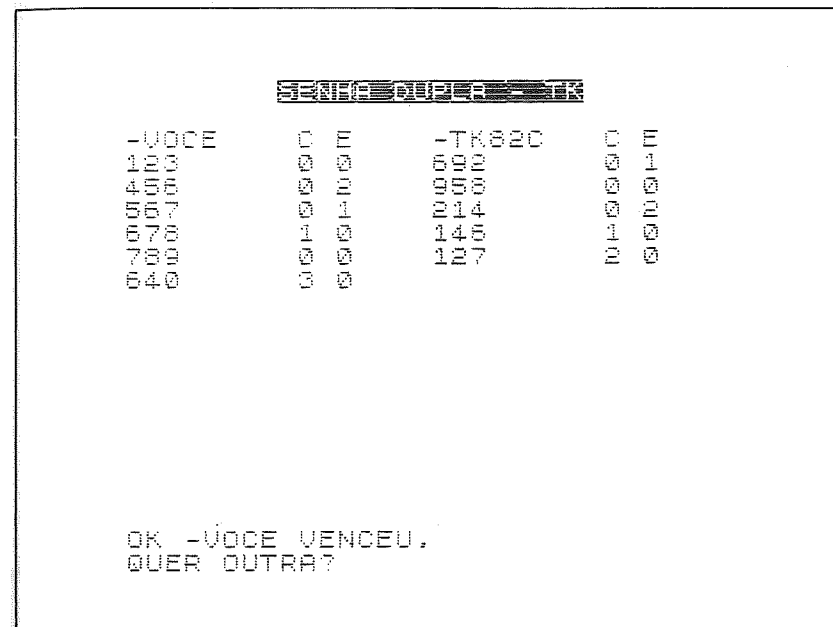


Figura 8.5 TELA DE UMA PARTIDA COM 3 DÍGITOS

C MANUAL DE OPERAÇÃO

Veja nos Apêndices como utilizar-se de fitas cassete. Este programa leva cerca de 2 minutos para ser carregado a partir da fita e ocupa cerca de 5K.

O programa listado a seguir é razoavelmente simples até a linha 4000 – TK joga. Até então trata-se do controle sobre quem joga, formatação da tela, teste de resposta do adversário e resposta de escores.

O primeiro palpite do computador é ao acaso. Daí em diante é montado a partir dos palpites anteriores. Na matriz D(R, S) são armazenados os palpites e em B e C os escores obtidos. Observe o dinamismo da matriz D. Depende de quantos dígitos tem a SENHA. Na primeira linha de D está a SENHA do computador. Na segunda linha está o primeiro palpite. O palpite atual, G, é feito da seguinte forma:

1. $P = 1$
 $G(P) = D(J, P)$
 onde J é o índice da jogada anterior.
 2. Se G(P) é repetido, *substitui-se* e volta-se a 2.
 3. Para cada palpite anterior:
 - 3.1 Comparam-se os P primeiros dígitos com os palpites anteriores;
 - 3.2 Se o número de dígitos comuns é maior que o escore do palpite, *substitui-se* G(P) e volta-se a 2.
 - 3.3 Se o número de certos no lugar certo é maior que o escore do palpite, *substitui-se* G(P) e volta-se a 2.
 - 3.4 Se o número de dígitos que falta ao palpite em análise for menor do que o que falta para completar os escores do palpite anterior, então *substitui-se* G(P) e volta-se a 2.
 4. $P = P + 1$
 Se P é maior que o número de dígitos da SENHA, está preparado o palpite. Incrementa-se J (contador de palpites) e salva-se o palpite G em D(J). Fim.
 Senão volta-se a 2.
- Observe que sublinhamos a palavra *substitui-se* na descrição anterior. Isto deve-se ao fato de não ser uma simples substituição ao acaso. É feita da seguinte maneira:
1. Incrementa-se o dígito de 3; se ultrapassar 9, subtrai-se 10.
 2. Se G(P) difere de D(J, P), ou seja, se não completamos um ciclo todo, testando todos os dígitos possíveis, então a substituição está feita.
 3. Se todos os dígitos foram testados, isto é, G(P) voltou a ser igual a D(J, P) então $P = P - 1$.

Isto é, voltamos ao dígito anterior, pois o mesmo não permite bom futuro. Se P regrediu abaixo do primeiro dígito ($P < 1$), então há algo errado nas respostas recebidas.

Se não, *substitui-se* G(P), isto é, volta-se a 1.

A limitação de 6 dígitos é devida à velocidade do equipamento utilizado. A limitação de 3 é pela dificuldade do jogo – com 1 ou 2 dígitos é fácil demais. Se você quiser aumentar a dificuldade e tiver paciência de esperar o TK passar mais de uma hora calculando, altere a linha 160. Com um computador mais rápido que o TK, você pode ir até 9 dígitos – mais que isso várias outras modificações seriam necessárias.

A listagem do programa se encontra na Figura 8.6.

```

10 REM SENHA DUPLA
20 REM ANTONIO JULIO LOSSIO BO
TELHO
30 FAST
35 CLS
37 PRINT
40 PRINT "VOU ESCOLHER UM NUME
RO DE 0 DI- GITOS PARA VOCE DESC
OBRIR."
50 PRINT "ENQUANTO ISTO VOCE E
SCOLHE UM NUMERO COM TAMBEM 0
DIGITOS PA- RA QUE EU POSSA DESC
OBRIR."
60 PRINT "JOGAREMOS ALTERNADAM
ENTE. VENCE QUEM PRIMEIRO DESCOB
RIR A SENHA DO OUTRO."
80 PRINT "CADA UM TENTA UM NUM
ERO E O OU- TRO RESPONDE COM A
QUANTIDADE DE DIGITOS CERTOS NO
LUGAR ERRADO E DE DIGITOS C
ERTOS NO LUGAR ERRADO."
100 PRINT "0 PODE SER 3,4,5 OU
6. SE VOCE ESCOLHER 3 O JOGO E
MAIS RAPIDO. SE VOCE ESCOLHER 6 U
MA JOGADA DOTK82C PODE DEMORAR A
TE 20 MINU- TOS, DEPENDENDO DAS U
OGADAS ANTE- RIORES."
110 PRINT "TECLE AGORA O VALOR
DE 0 OU SE VOCE NAO ENTENDEU TE
CLE N."

120 SLOW
130 LET A$=INKEY$
140 IF A$="N" THEN STOP
150 IF A$<"3" THEN GOTO 130

```

Figura 8.6

LISTAGEM DO PROGRAMA - FONTE (SENHA DUPLA)

```

160 IF A#>"6" THEN GOTO 130
165 FAST
170 LET S=VAL (A#)
175 LET R=0*S+1
180 DIM D(R,S)
185 DIM B(R)
190 DIM C(R)
195 DIM G(S)
200 CLS
210 PRINT AT 2,8;"SENHA DUPLA"
220 PRINT AT 4,0;"-VOCE C E"
230 PRINT AT 4,16;"-TK820 C E"
235 LET A=1
240 GOSUB 2000
250 LET J=1
260 LET L=5
270 REM HUMANO JOGA
280 SLOW
290 PRINT AT 21,0;"ENTRE O SEU
PALPITE E AGUARDE "
300 FOR I=1 TO 3
310 PRINT AT L,I-1;"■"
320 LET A#=INKEY#
340 IF A#<"0" THEN GOTO 320
350 IF A#>"9" THEN GOTO 320
360 LET G(I)=VAL A#
370 IF I=1 THEN GOTO 410
380 FOR K=1 TO I-1
390 IF G(I)=G(K) THEN GOTO 320
400 NEXT K
410 PRINT AT L,I-1;A#
420 NEXT I
430 LET P=5
440 LET A=1
442 GOSUB 3000
444 IF A<>1 THEN GOTO 290
450 GOSUB 3000
460 PRINT AT L,9;V;" ";U-U
470 IF V<5 THEN GOTO 540
480 PRINT AT 21,0;"OK -VOCE VEN
CEU."
490 SCROLL
492 PRINT "QUER OUTRA? "
495 SLOW
500 LET A#=INKEY#
510 IF A#="S" THEN GOTO 200
520 IF A#<>"N" THEN GOTO 500
530 STOP
535 RUN

```

Figura 8.6

LISTAGEM DO PROGRAMA-FONTE (SENHA DUPLA) (cont.)

```

540 PRINT AT 21,0;"POR FAVOR AG
UARDE ENQUANTO PENSO"
550 PAUSE 200
560 FAST
570 IF J>1 THEN GOTO 610
580 LET A=2
590 GOSUB 2000
600 GOTO 620
610 GOSUB 4000
620 PRINT AT 21,0;"MEU PALPITE.
-QUAL MEU ESCORE? "
630 FOR I=1 TO 3
640 PRINT AT L,I+15;D(J,I)
650 NEXT I
660 PRINT AT L,25;"■";
670 SLOW
680 GOSUB 3500
690 LET B(J)=VAL A#
700 PRINT AT L,25;A#;" ■"
710 GOSUB 3500
720 LET C(J)=B(J)+VAL A#
730 PRINT AT L,27;A#
732 LET A=2
734 GOSUB 3000
736 IF A<>2 THEN GOTO 630
740 IF C(J)<=5 THEN GOTO 790
750 PRINT AT 21,0;"NAO SEJA RID
ICULO"
760 PAUSE 400
780 GOTO 630
790 IF B(J)=3-1 AND C(J)=5 THEN
GOTO 750
792 IF L<20 THEN GOTO 800
794 GOSUB 8070
796 SCROLL
798 GOTO 810
800 LET L=L+1
810 IF B(J)<5 THEN GOTO 280
820 PRINT AT 21,0;"HI HI - VIU?
GANHEI HI HI "
830 PAUSE 500
840 GOSUB 8070
850 PRINT AT 21,0;"MEU NUMERO E
RA "
860 FOR I=1 TO 3
870 PRINT D(1,I);
880 NEXT I
890 PRINT " "
900 GOTO 490
2000 FOR P=1 TO 5
2010 LET D(R,P)=INT (10*RND)
2020 IF P<1 THEN GOTO 2060

```

Figura 8.6

LISTAGEM DO PROGRAMA-FONTE (SENHA DUPLA) (cont.)

```

2030 FOR I=1 TO P-1
2040 IF D(A,I)=D(A,P) THEN GOTO
2010
2050 NEXT I
2060 NEXT P
2070 IF A=1 THEN RETURN
2080 LET J=2
2090 RETURN
3000 REM CONTRA
3010 LET V=0
3020 LET W=0
3030 FOR I=1 TO P
3040 IF D(A,I)=G(I) THEN LET V=V
+1
3050 FOR K=1 TO S
3060 IF D(A,K)=G(I) THEN LET W=W
+1
3070 NEXT K
3080 NEXT I
3090 RETURN
3500 LET A$=INKEY$
3510 IF A$>CHR$(S+28) THEN GOTO
3500
3520 IF A$<"0" THEN GOTO 3500
3530 RETURN

4000 REM TK JOGA
4010 LET P=1
4020 LET G(P)=D(J,P)
4030 IF P=1 THEN GOTO 4070
4040 FOR I=1 TO P-1
4050 IF G(I)=G(P) THEN GOTO 4100
4060 NEXT I
4070 FOR A=2 TO J
4080 GOSUB 3000
4090 IF V<=B(A) THEN IF W<=C(A)
THEN IF S-P>=C(A)-W THEN IF S-P>
=B(A)-V THEN GOTO 4170
4100 LET G(P)=G(P)+3
4110 IF G(P)>9 THEN LET G(P)=G(P)
)-10
4120 IF P=1 THEN IF G(P)=D(2,P)
THEN GOTO 4500
4130 IF G(P)<>D(J,P) THEN GOTO 4
030
4140 LET P=P-1
4150 IF P<1 THEN GOTO 4500
4160 GOTO 4100
4170 NEXT A
4180 LET P=P+1
4190 IF P<=S THEN GOTO 4020
4200 LET J=J+1

```

Figura 8.6

LISTAGEM DO PROGRAMA - FONTE (SENHA DUPLA) (cont.)

```

4210 FOR I=1 TO S
4220 LET D(J,I)=G(I)
4230 NEXT I
4240 RETURN
4500 PRINT AT 21,0;"VOCE ME DEU
RESPOSTA ERRADA"
4530 SCROLL
4540 PRINT AT 21,0;"LOGO VOCE PE
RDEU."
4550 GOTO 490
6000 PRINT AT 21,0;"CONFIRME, C=
=>OK OU N=>ERRADO"
6010 LET A$=INKEY$
6020 IF A$="C" THEN GOTO 6070
6030 IF A$<>"N" THEN GOTO 6010
6040 IF A=1 THEN PRINT AT L,0;"
"
6050 IF A<>1 THEN PRINT AT L,25;
"
6060 LET A=0
6070 PRINT AT 21,0;"
"
6080 RETURN
9000 FAST
9010 CLEAR
9020 SAVE "SENHA"
9030 GOTO 10

```

Figura 8.6

LISTAGEM DO PROGRAMA - FONTE (SENHA DUPLA) (cont.)

CAPÍTULO 9

***Divirta-se
Brincando
de Força***

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:
Programa <input checked="" type="checkbox"/>	<u>FORCA</u>	<u>1</u>	<u>φ</u>	<u>22/10/83</u>
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>	
NE Z8000 <input checked="" type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input type="checkbox"/>	
TK 80 <input type="checkbox"/>	48K <input type="checkbox"/>	64K <input type="checkbox"/>	LPRINT <input type="checkbox"/>	
TK 82-C/83 <input checked="" type="checkbox"/>	CASSETE <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>	
TK 85 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	5 1/4" <input type="checkbox"/>	DSAVE <input type="checkbox"/>	
SINCLAIR ZX 80 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	DMLOAD <input type="checkbox"/>	
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>	DMSAVE <input type="checkbox"/>	
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>			OUTROS (quais?) <input type="checkbox"/>	
TIMEX 1000 <input checked="" type="checkbox"/>				
CP-200 <input checked="" type="checkbox"/>				
CP-300 <input type="checkbox"/>				
APPLY 300 <input type="checkbox"/>				
OUTROS (quais?) <input checked="" type="checkbox"/>				
<u>RINGO R-470 e</u>				
<u>A3 - 1000</u>				
k) Observações:				

O jogo da FORCA é muito conhecido. Trata-se de descobrir uma palavra. Informa-se quantas letras tem a palavra, e o adversário vai tentando letras. Se uma letra está na palavra, é informada a posição da letra na palavra (ou posições, se tiver mais que uma ocorrência).

Para cada letra tentada e que não ocorre na palavra, uma parte de uma forca ou do enforcado é desenhada. O jogo termina com o descobrimento da palavra ou com o enforcamento (Figura 9.2).

O jogo se inicia – veja Figura 9.1 – com a tela em branco. Na antepenúltima linha temos tantos traços quantos forem as letras da palavra. Na penúltima linha temos as letras do alfabeto. Na última o pedido:

ENTRE SEU PALPITE

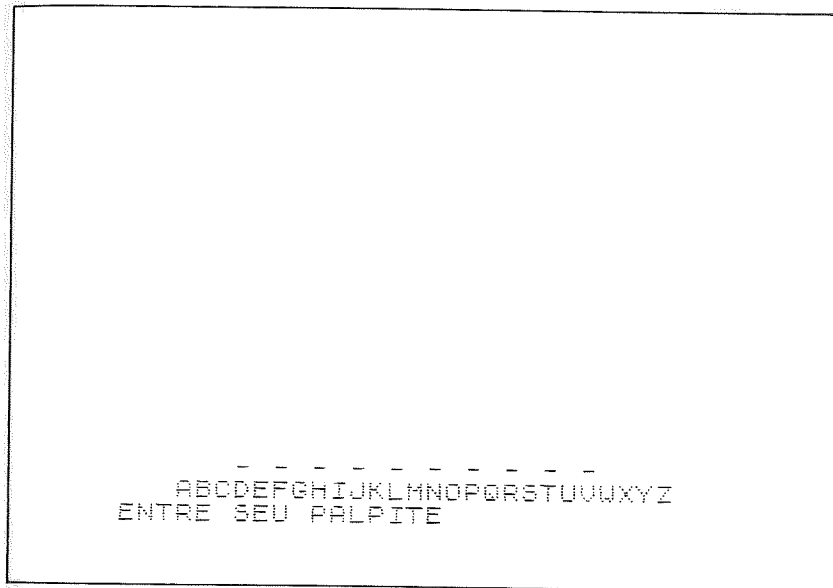


Figura 9.1 TELA INICIAL

Tecele uma das letras da penúltima linha. Ela sairá de lá. A última linha se apagará. Se a letra estiver na palavra, aparecerá nos locais corretos, em substituição aos traços correspondentes. Se não estiver na palavra, a FORCA será iniciada.

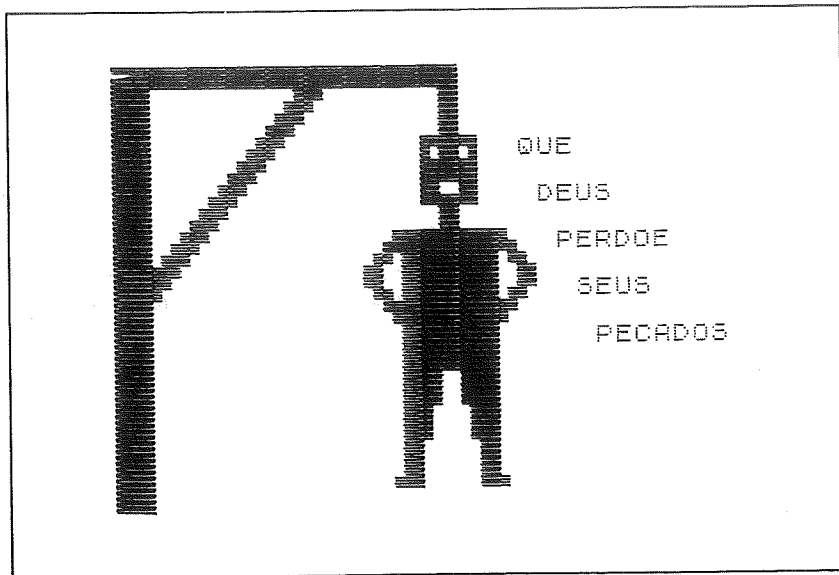


Figura 9.2 ENFORCAMENTO

Se você for enforcado, a palavra ser-lhe-á informada como consolo.

O banco de palavras fornecido na fita é de 80 palavras que são apresentadas na Figura 9.3.

1	-	PETROLEO
2	-	PESTANA
3	-	PESTILENTO
4	-	PETELECO
5	-	REBOQUE
6	-	CADAFALSO
7	-	CADEIRA
8	-	COLA
9	-	REBENTO
10	-	DEFUMADO
11	-	DISCURSO
12	-	ELETRDOMESTICO
13	-	ELETRONICA
14	-	ENGRENAGEM
15	-	ESFOMEADO
16	-	FERROLHO
17	-	FERROVIA

Figura 9.3 RELAÇÃO DE PALAVRAS

18	-	FERTIL
19	-	FORTE
20	-	FORTALEZA
21	-	FORTIFICANTE
22	-	GALOCHA
23	-	OFICINA
24	-	OGIVA
25	-	ORNAMENTO
26	-	PALADIO
27	-	PALACIO
28	-	PALADINO
29	-	PALAVRA
30	-	PREOCCUPACAO
31	-	PULGA
32	-	RADIOLOGIA
33	-	RADIO
34	-	GALVANIZADO
35	-	RADIOGRAFIA
36	-	GRANJA
37	-	REBOTALHO
38	-	GRANULADO
39	-	INDUZIDO
40	-	INDUMENTARIA
41	-	IMBATIVEL
42	-	INDUSTRIA
43	-	INTERNACIONAL
44	-	LABIRINTO
45	-	LOBISOMEM
46	-	ELEFANTE
47	-	LOCOMOTIVA
48	-	LOCATARIO
49	-	TESOURA
50	-	LAPIS
51	-	LOBO
52	-	MASCARADO
53	-	MASSAGISTA
54	-	MATADOURO
55	-	MOBILIA
56	-	MOCHILA
57	-	MOCOTO
58	-	NORMALISTA
59	-	NORDESTE
60	-	NITROGLICERINA
61	-	POLICIA
62	-	LADRAO
63	-	CASSETE
64	-	FLAMENGO
65	-	VASCO
66	-	MAQUINA
67	-	REVISTA
68	-	HELICOPTERO

Figura 9.3 RELAÇÃO DE PALAVRAS (cont.)


```

60 - LIVRO
70 - FORÇA
71 - MENINOS
72 - CASA
73 - BARCO
74 - LOJA
75 - BONECA
76 - FESTA
77 - TELEVISÃO
78 - APARTAMENTO
79 - COMPUTADOR
80 - ARRABADOR

```

Figura 9.3. **RELAÇÃO DE PALAVRAS** (cont.)

Você pode aumentar o número de palavras. E muito. Menos de 5K estão em uso. Pelo menos mais 160 palavras podem ser incluídas antes de completarem 16K. Veja no manual técnico como incluir ou alterar o banco de palavras.

C MANUAL DE OPERAÇÃO

O tempo de carga a partir da fita cassete é de cerca de 1 minuto e 40 segundos ocupando cerca de 4K.

Veja como operar fita cassete nos Apêndices.

D MANUAL TÉCNICO

O principal problema deste jogo é memória para armazenar-se o banco de palavras. Se as instruções **READ**, **DATA** e **RESTORE** estão disponíveis, fica mais simples. Uma série de instruções **DATA** com as palavras. Sorteia-se a palavra (ou seja, seu número da seqüência nos **DATA** e lê-se (**READ**) tantas vezes quanto necessário até chegar a ela. Dá-se **RESTORE**.

Como o TK não possui estas instruções, usa-se muito freqüentemente dimensionar um vetor e ler do teclado os valores a serem armazenados antes do **SAVE**. Com cuidado. Devem-se evitar os comandos **CLEAR** e **RUN** se este recurso é utilizado. Tornam-se perigosas alterações nestes dados.

A solução que adotamos não é tão econômica em termos de memória, porém mais segura. Guardar no texto do programa como uma constante. Entenda

bem, não inicializando um vetor com as palavras, pois cada palavra ocuparia o dobro do espaço necessário. Veja a linha 8600. Uma constante alfanumérica gigante, da qual pegamos uma parte (da letra $l - L + 1$ até l) ou seja, apenas a palavra sorteada.

Na linha 8000 temos o número de palavras no banco. Na linha 8500 temos os comprimentos de cada uma delas. Observe que a atribuição (**LET**) da linha 8500 é de uma cadeia que inicia com 00 e 00 não é comprimento de nenhuma palavra — o 00 está aí para que a função **VAL** possa funcionar corretamente. Na linha 8500 temos NP (veja linha 8000) números de dois dígitos (além de 00) separados por +. Sorteia-se a palavra P (linha 110). Para localizá-la na constante da linha 8600 basta eliminarmos todas as palavras anteriores — isto é, somamos (veja porque o + para separar os comprimentos na linha 8500) os comprimentos de todas as palavras anteriores — temos então onde começa a palavra sorteada. Pelo comprimento dele, podemos extraí-la de imediato. O somatório é obtido usando-se a função **VAL**. Sem o 00 na frente daria erro quando a palavra sorteada fosse a primeira.

Observe melhor a idéia descrita acima no exemplo da Figura 9.4 com apenas 3 palavras:

```

100 LET NP = 3
110 LET P = RND * NP + 1
120 LET N$ = "00 + 05 + 03 + 06"
130 LET L = VAL N$ (P * 3 + 1 TO)
140 LET I = VAL N$
150 LET P$ = "CASALOVOFERTIL" (I - L + 1 TO I)

```

Figura 9.4. **EXEMPLO**

Suponha que a palavra sorteada seja 1. N\$ passaria a conter 00 + 05, logo L conteria 5 e I também. P\$ seria CASAL.

Suponha agora que seja a segunda palavra sorteada (P = 2). N\$ teria os 8 primeiros caracteres da seqüência, ou seja, 00 + 05 + 03. O valor (**VAL**) dos 2 últimos caracteres da N\$ seria 3 e o valor (**VAL**) de N\$ resulta 8. Assim, da sexta (8 - 3 + 1 TO 8) a oitava letra da cadeia da linha 150 do exemplo temos OVO. A terceira palavra é FERTIL. Verifique.

Entendida a estrutura, fica mais fácil alterar.

Para inserir novas palavras, faça-o na frente (dá muito menos trabalho) — logo após o = " (igual-asps) na linha 8600, ou seja, antes da primeira palavra.

Depois insira os comprimentos na linha 8500, na mesma ordem das palavras inseridas na linha 8600. Observe que isto deve ser feito após o 00.

Finalmente atualize a linha 8000 para que NP reflita o novo número de palavras no banco.

Não esqueça de salvar sua nova *personalizada* versão da FORCA, em fita cassete (ver Apêndices).

A rotina 9500 é apenas para depuração do banco de palavras.

O programa da Figura 9.5 usa estruturalmente exatamente o algoritmo da Figura 9.4. Porém a rotina 9500 exigiria mais para ter a utilidade que tem. Assim, transformaremos a linha 100 na rotina 8000 (chamando em 100 pela FORCA e em 9510 pelo utilitário). As linhas 120 a 150 transformaram-se na rotina 8500, chamado na linha 150 do programa e na 9530 pelo utilitário. Observe que com isto o texto das linhas fundamentais fica sendo usado pelos dois programas, garantindo pois a correção dos dados após a execução do utilitário com sucesso.

Entre RUN 9500 que as palavras serão listadas na tela para você conferir. Ao notar erro, tecele BREAK. Acerte a linha que contém o erro (8500 ou 8600) e repita. Volto a lembrar — se fez correções, não esqueça de salvar o programa, gravando-o em fita cassete (ver Apêndices).

Na Figura 9.5 encontra-se a listagem do programa-fonte.

```

10 REM 8000 ANTONIO JULIO LOS
SIO BOTELHO
20 SLOW
30 CLS
70 CLEAR
80 LET G$="ABCDEFGHIJKLMNPOQRS
TUVWXYZ"
100 GOSUB 8000
110 LET P=INT (RND*NP+1)
150 GOSUB 8500
200 LET Z=INT ((30-L-L)/2-1)
210 FOR I=1 TO L
220 PRINT AT 19,Z+I+I;"-";
230 NEXT I
240 LET E=0
245 LET C=0
250 PRINT AT 20,0;"

255 PRINT AT 20,(31-LEN G$)/2;G
$)

```

Figura 9.5

LISTAGEM DO PROGRAMA-FONTE (FORCA)

```

260 PRINT AT 21,0;"ENTRE SEU PA
LPITE"
270 LET A$=INKEY$
280 IF A$<"A" OR A$>"Z" THEN GO
TO 270
290 PRINT AT 21,0;"

300 FOR I=1 TO LEN G$
310 IF G$(I TO I)=A$ THEN GOTO
340
320 NEXT I
330 GOTO 260
340 LET G$=G$(1 TO I-1)+G$(I+1
TO )
350 LET X=0
360 FOR I=1 TO L
370 IF A$<>P$(I TO I) THEN GOTO
410
380 LET C=C+1
390 LET X=1
400 PRINT AT 19,Z+I+I;A$;
410 NEXT I
420 IF X=0 THEN GOTO 510
430 IF C<L THEN GOTO 250
440 PRINT AT 21,0;"PARABENS - U
OCE ACERTOU."
450 SCROLL
460 PRINT AT 21,0;"OUBR OUTRA?
(S/N)"
470 LET A$=INKEY$
480 IF A$="S" THEN RUN
490 IF A$<>"N" THEN GOTO 470
495 STOP
500 GOTO 460
510 LET E=E+1
520 GOSUB E*1000
530 GOTO 250

1000 FOR J=18 TO 0 STEP -1
1010 PRINT AT J,0;"███"
1020 NEXT J
1030 PRINT AT 0,0;"███████████
███"
1040 FOR J=1 TO 8
1050 PRINT AT J,10-J;"███"
1060 NEXT J
1070 PRINT AT 9,2;"███"
1080 RETURN
2000 PRINT AT 3,16;"███"
2010 PRINT AT 4,16;"███"
2020 PRINT AT 5,16;"███"
2030 PRINT AT 6,17;"███"

```

Figura 9.5

LISTAGEM DO PROGRAMA-FONTE (FORCA) (cont.)

```

2040 RETURN
3000 FOR J=7 TO 12
3010 PRINT AT J,15;"██████████"
3020 NEXT J
3030 RETURN
4000 PRINT AT 13,15;"██████████"
4010 PRINT AT 14,15;"██████████"
4020 PRINT AT 15,15;"██████████"
4030 PRINT AT 16,15;"██████████"
4040 PRINT AT 17,14;"██████████"
4050 RETURN
5000 PRINT AT 7,14;"█";AT 8,13;"█";
█";AT 9,13;"█";AT 10,14;"█";
5010 RETURN
6000 PRINT AT 7,20;"█";AT 8,20;"█";
█";AT 9,20;"█";AT 10,20;"█";
6010 RETURN
7000 PRINT AT 1,17;"█";AT 2,17;"█";
█";AT 3,21;"QUE";AT 4,20;"DEUS";
AT 7,23;"PERDOE";AT 9,24;"SEUS";
AT 11,25;"PECADOS"
7005 RETURN
7010 PRINT AT 21,0;"ERRA:";P#
7020 GOTO 450
8000 LET NP=80
8010 RETURN
8500 LET N#="00+06+07+10+08+07+0
9+07+04+07+08+08+15+10+10+09+08+
08+06+06+09+12+07+07+05+09+07+07
+08+07+11+05+10+05+11+11+06+09+0
9+06+12+09+09+13+09+09+08+10+09+
07+05+04+09+10+09+07+07+05+10+08
+14+07+06+07+08+05+07+07+11+05+0
5+07+04+05+04+05+05+09+11+10+08"
(1 TO P*3+2)
8510 LET L=VAL N#(P*3+1 TO )
8520 LET I=VAL N#
8600 LET P#="PETROLEOPESTANAPEST
ILENTOPETELECOREBOQUECADAFALSOCA
DEIRACOLAREBENTODEFUMADODISCURSO
ELETRODOMESTICOELETRONICAENGRENA
GEMESFOMEADOFERROLHOFERROUIAFERT
ILFORTEFORTALEZAFORTIFICANTEGALO
CHAOFICINADGIVAORNAMENTOPALADIOP
ALACIOPALADINOPALAVRAPREOCUPACAO
PULGARADIOLOGIARADIOGALVANIZADOR
ADIOGRAFIAGRANJAREBOTALHOGUANULA
DOINDUZIDOINDUMENTARIAIMBATIVELI
NDUSTRIAINTERNACIONALLABIRINTOLO
BISOMELEFANTELOCOMOTIVALOCATAR
IOTESOURALAPISLOBOMASCARADOMASSA
GISTAMATADOUROMOBILIAMOCHILAMOCO

```

```

TONORMALISTANORDESTENITROGLICERI
NAPOLICIALADRAOCASSETEFLAMENGOVA
6000MAQUINAREVISTAHELICOPTEROLIVR
OFORCAMENINOSCASABARCOLOJABONECA
FESTATELEVISAOAPARTAMENTOCOMPUTA
DORGRAVADOR"(I-L+1 TO I)
8610 RETURN
9000 CLEAR
9010 FAST
9020 SAVE "FORCA"
9030 RUN
9040 SLOW
9050 GOSUB 8000
9060 FOR P=1 TO NP
9070 GOSUB 8500
9080 SCROLL
9090 PRINT AT 21,P<=9;P;" - ";P#
9100 NEXT P
9170 STOP

```

Figura 9.5

LISTAGEM DO PROGRAMA-FONTE (FORCA) (cont.)

CAPÍTULO **10**

***Damas
Tradicional***

A FICHA

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:
Programa <input checked="" type="checkbox"/>	<u>DAMAS</u>	<u>1</u>	<u>φ</u>	<u>22/10/83</u>
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	TK 80 <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	SINCLAIR ZX 80 <input type="checkbox"/>	64K <input type="checkbox"/>	LPRINT <input type="checkbox"/>
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	SINCLAIR ZX 81 <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	LLIST <input type="checkbox"/>
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	TIMEX 1000 <input checked="" type="checkbox"/>	5 1/4" <input type="checkbox"/>	DLOAD <input type="checkbox"/>
CP-200 <input checked="" type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	CP-300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	DMSAVE <input type="checkbox"/>
APPLY 300 <input type="checkbox"/>	OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>
<u>RINGO R-470 e AS-1000</u>				
k) Observações:				

B MANUAL DE UTILIZAÇÃO

DAMAS é jogado em um tabuleiro plano, reticulado em 8 x 8, retículos alternados em escuros e claros como visto na Figura 10.1

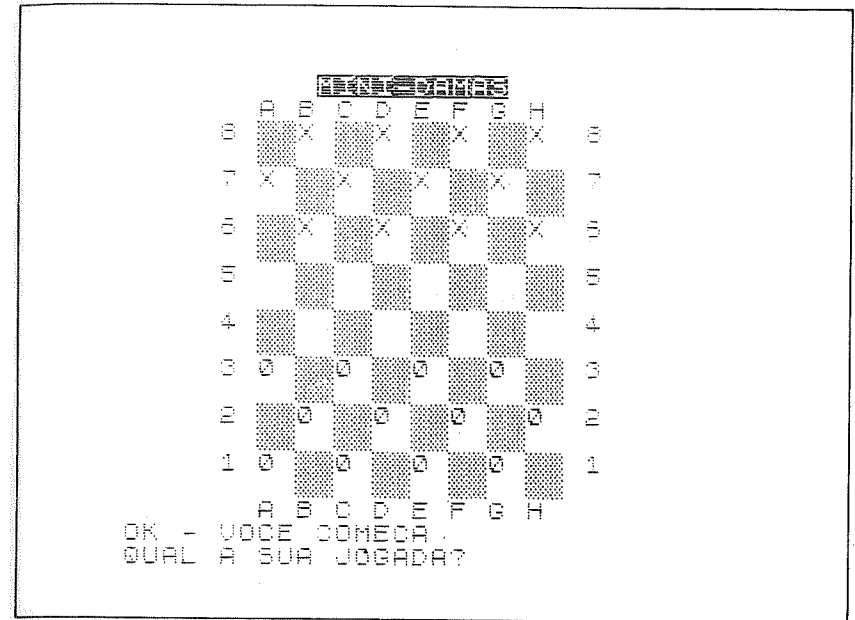


Figura 10.1
TABULEIRO

Cada jogador utiliza-se de 12 pedras, que são distribuídas sobre as casas (retículos) claros em lados opostos do tabuleiro (veja Figura 10.1).

Os jogadores movem suas pedras alternadamente, com movimentos em diagonal sempre em direção ao lado oposto, e se a casa imediatamente seguinte estiver vazia.

Se uma pedra do adversário se encontra em casa imediatamente vizinha e se a casa subsequente na diagonal estiver vazia, você pode mover sua pedra, saltando sobre a pedra do adversário. A pedra saltada é retirada do tabuleiro e dita que foi *comida*. Observe que neste caso foi permitido que sua pedra se movimentasse para trás (na direção contrária ao movimento normal).

Se uma pedra efetuou uma *comida*, é permitido a essa pedra efetuar outras *comidas* possíveis, na mesma seqüência de movimentos.

O objetivo é *comer* todas as pedras do adversário.

Se uma pedra atingir o extremo oposto do tabuleiro, essa pedra recebe o título de *dama* e ganha o direito de movimentar-se indistintamente para frente e para trás, bem como efetuar movimentos longos, isto é, pode mover-se para casas não imediatamente subseqüentes. As pedras comuns são chamadas *peões*.

Para efeitos de comunicação entre você e o computador, as linhas do tabuleiro serão identificadas por dígitos de 1 a 8 e as colunas por uma letra de A a H, e sempre que for necessário referir-se a um retículo o faremos pela letra identificadora da coluna, seguida do dígito identificador da linha. As pedras do computador serão representadas por X e inicialmente serão colocadas em B8, D8, F8, H8, A7, C7, E7, G7, B6, D6, F6 e H6. Suas pedras serão representadas por Ø e estarão em A1, A3, B2, C1, C3, D2, E1, E3, F2, G1, G3 e H2 (veja Figura 1Ø.1).

O programa inicia-se após o **LOAD** ou **RUN** perguntando quem começa. Responda EU, TU ou VOCE. A tela se apagará por alguns segundos, voltando a seguir com o tabuleiro desenhado.

Será então sua vez de mover. Escolha a pedra. Veja em que coluna ela está — tecle a letra. Veja a linha — tecle o número. Veja para onde a peça deve ir — tecle a letra da coluna — depois o número da linha. Se for movimento duplo (comida de mais de uma pedra) tecle a letra e o número do próximo local. Ao final tecle NEWLINE.

A tela se apagará durante algum tempo enquanto o computador avalia a situação — cerca de 5 segundos por peça do computador no tabuleiro, ou seja cerca de 1 minuto no início do jogo.

A tela voltará já com o movimento do computador efetuado e indicado, como por exemplo:

MINHA JOGADA FOI A7B6.

Se você responder errado ou tentar movimento inválido, receberá uma mensagem de erro. Várias mensagens farão o tabuleiro ficar parcialmente destruído na tela — Entre ? NEWLINE que o tabuleiro será refeito.

A qualquer momento você poderá desistir — basta teclar FIM NEWLINE.

C

MANUAL DE OPERAÇÃO

O programa DAMAS leva cerca de três minutos e meio para ser carregado a partir da fita no TK 82C, ocupando cerca de 9K de memória.

Veja nos Apêndices o uso da fita cassette.

D

MANUAL TÉCNICO

Elaborar um programa para jogar DAMAS inteligentemente não é uma tarefa das mais difíceis. Porém, em BASIC temos um problema sério a resolver: o tempo. Nesta versão simplificada que apresentamos a seguir, o tempo que o computador passa avaliando as jogadas é muito grande, principalmente no início. E esta versão é incompleta em termos de DAMAS. É para principiantes. O computador move suas pedras de forma razoável, porém fica em completa desvantagem quando tem que mover uma *dama*. Isto deve-se ao fato de termos utilizado para a *dama* o mesmo algoritmo do *peão*, visto que o tempo de resposta está excessivamente grande.

A seleção de qual o próximo movimento, é feita levando-se em consideração apenas o movimento seguinte do adversário. Pesquisar mais tornaria o jogo muito demorado e programá-lo em linguagem de máquina conflita frontalmente com os princípios deste livro, conforme apresentado na introdução.

Este programa pode ser modificado, mormente na rotina de avaliação de movimentos com *damas*, onde foi utilizado um algoritmo simplista. Um terceiro nível de lances (examinar à frente) pode ser incluído, porém deverá fazer o programa ficar lento demais.

O tabuleiro de DAMAS, semelhante ao xadrez, é uma matriz de 8 x 8.

Este tabuleiro está armazenado na matriz A\$, de 2 caracteres e inicializado entre as linhas 9Ø e 26Ø do programa.

Da linha 3ØØ à 39Ø decide-se quem começa e da 4ØØ à 48Ø temos o controle da seqüência do jogo, através de chamadas alternadas as rotinas 2ØØØ e 6ØØØ, respectivamente jogada do adversário e do computador. A rotina 5ØØØ é de gerência de tela.

A rotina 2ØØØ efetua o movimento indicado em H\$, que por sua vez é lido via instrução INPUT. Antes de efetuar o movimento, é feita a consistência, de forma a não permitir movimentos incorretos. A rotina 6ØØØ seleciona o movimento e transfere para 2Ø3Ø de forma a evitar duplicidade de código. A rotina 3ØØØ é chamada pela 2ØØØ para efetuar o movimento de dama. A série de linhas do milhar 4ØØØ é de complementação à rotina 2ØØØ para emissão de mensagens. Observe-se o uso do PRINT AT 21,Ø em todas as mensagens emitidas, bem como o fato de completar-se a mensagem com espaços para garantir a limpeza da última linha da tela.

A seleção do movimento do computador é feita na rotina 6000. Nos vetores P e R\$ as rotinas 7000 e 8000 armazenam a avaliação de movimentos para cada uma das peças do computador. O desempate entre duas avaliações é feito ao acaso (RND).

A rotina 7000 avalia a jogada de pedra e a 8000 a da dama. Uma pedra ganha 9 pontos se puder ser movida para linha da dama (extremo oposto) ou antecessora, 8 pontos se para as duas linhas anteriores ou 7 pontos se nas linhas defensivas. Ganha 9 pontos se o movimento come uma pedra do adversário e mais 2 se essa pedra for uma *dama*. Perde 10 (DM) pontos se este movimento dá essa pedra para ser comida pelo adversário. Se não mover a pedra, permitindo que esta seja comida, ganha 10 (DM) pontos. Observe-se que não foi analisado o fato de o movimento de uma peça permitir ao adversário comer uma outra pedra, protegida pela peça movida.

A avaliação da *dama* é semelhante, porém as bonificações são de 12 pontos e não 9, a linha onde a peça fique não é considerada e ser comida pelo adversário (DM) é de 15 pontos (a *mais* se não movida, e *menos* se movida).

Na Figura 10.2 encontra-se a listagem do programa-fonte.

```

10 REM DAMAS ████████
20 REM ANTONIO JULIO LOSSIO BD
TELHA
30 FAST
40 REM INICIALIZACAO
42 DIM G(4)
44 DIM F(4)
46 DIM F#(4,2)
50 DIM E#(2)
51 DIM X#(1)
52 DIM O#(1)
60 DIM A#(3,8,2)
63 DIM P(12)
66 DIM R#(12,20)
68 DIM L(12)
70 LET G=-1
80 LET L=0
90 LET B#=""
100 FOR Y=1 TO 8
110 FOR X=1 TO 8 STEP 2
120 LET A#(X,Y)=B#(1 TO 2)
130 LET A#(X+1,Y)=B#(3 TO 4)
140 NEXT X
150 LET B# = B#(3 TO 4) + B#(1 TO 2)
160 NEXT Y

```

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS)

```

170 FOR Y=1 TO 3
180 FOR X=1 TO 8
190 IF A#(X,Y)="" THEN LET A#
(X,Y)="0"
200 NEXT X
210 NEXT Y
220 FOR Y=6 TO 8
230 FOR X=1 TO 8
240 IF A#(X,Y)="" THEN LET A#
(X,Y)="X"
250 NEXT X
260 NEXT Y
300 CLS
310 PRINT AT 21,0;"QUEM COMECA?"

320 INPUT H#
330 IF H#="EU" THEN GOTO 390
340 IF H#="TU" OR H#="VOCE" THE
N GOTO 440
350 SCROLL
360 PRINT AT 21,0;"NAO ENTENDI
O QUE VOCE DISSE."
370 SCROLL
380 GOTO 310
390 GOSUB 5000
400 PRINT "OK - VOCE COMECA"
410 LET J=0
420 LET EU=0
430 GOSUB 2000
440 LET EU=1
450 GOSUB 5000
460 GOSUB 5000
470 PRINT "MINHA JOGADA FOI ";T
#;"."
480 GOTO 410

2000 REM JOGADA HUMANA
2001 LET U=1
2005 IF J<>0 THEN GOTO 4800
2010 PRINT "QUAL A SUA JOGADA?"
2013 LET O#="0"
2016 LET X#="X"
2020 INPUT H#
2025 IF LEN H#<4 THEN GOTO 4600
2030 LET E#=H#(1 TO 2)
2033 LET T#=H#
2035 LET E=CODE H#-37
2040 IF E<1 OR E>8 THEN GOTO 450
0
2060 LET H#=H#(2 TO )
2070 LET H=CODE H#-28
2080 IF H<1 OR H>8 THEN GOTO 450

```

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

```

0
0100 LET H#=H$(2 TO )
0110 IF A$(E,H,1 TO 1)=0$ THEN G
OTO 0150
0120 SCROLL
0130 PRINT T#;"???" NAO TEM PECA
0140 REM "E#"
0140 GOTO 0300
0150 LET E#=H$(1 TO 2)
0160 LET A=CODE H#-37
0170 IF A<1 OR A>8 THEN GOTO 450
0
0210 LET H#=H$(2 TO )
0220 LET B=CODE H#-38
0230 IF B<1 OR B>8 THEN GOTO 450
0
0230 LET H#=H$(2 TO )
0240 IF A$(A,B)=" " THEN GOTO 2
0250 SCROLL
0260 PRINT T#;"???"E#;" JA ESTA
OCUPADO."
0270 GOTO 0300
0280 IF A$(E,H,2 TO 2)=0$ THEN G
OTO 0300
0300 IF ABS (A-E)<>1 THEN GOTO 2
400
0310 IF (B-H)=0 THEN GOTO 0300
0320 SCROLL
0330 PRINT "MOVIMENTO INCORRETO-
:T#)"
0350 GOTO 0300
0400 IF ABS (A-E)<>2 THEN GOTO 2
300
0410 IF ABS (B-H)<>2 THEN GOTO 2
300
0420 LET U=(A+E)/2
0430 LET V=(B+H)/2
0440 IF A$(U,V,1 TO 1)<>X$ THEN
GOTO 0320
0450 LET A$(U,V)=" "
0500 GOTO 0310
0500 IF J<>0 THEN GOTO 4800
0510 LET J=1
0520 LET A$(A,B)=A$(E,H)
0530 LET A$(E,H)=" "
0535 LET T=0
0538 IF O#<>"0" THEN LET T=1
0540 IF B=T AND A$(A,B,2 TO 2)<>
O# AND LEN H#<0 THEN LET A$(A,B)
=O#+O$
0550 IF LEN H#<0 THEN RETURN

```

```

0360 IF LEN H#<2 THEN GOTO 4800
0370 LET E=A
0380 LET H=B
0390 GOTO 0160
0000 REM HUMANO MOVE RAINHA
0010 IF ABS (A-E)<>1 THEN GOTO 3
100
0020 IF ABS (B-H)<>1 THEN GOTO 2
300
0030 GOTO 0300
0100 IF A=E OR B=H THEN GOTO 032
0
0120 LET U=1
0130 LET V=1
0140 IF A=E THEN LET U=-U
0150 IF B=H THEN LET V=-V
0160 LET C=0
0170 LET N=H+U
0180 FOR M=E+U TO A-U STEP U
0190 IF A$(M,N)=" " THEN GOTO 3
0200
0210 IF A$(M,N,1 TO 1)=0$ THEN G
OTO 0320
0220 LET X=M
0230 LET Y=N
0240 LET C=C+1
0250 LET N=N+U
0260 NEXT M
0270 IF C=0 THEN GOTO 0300
0280 IF C>1 THEN GOTO 0320
0290 LET A$(X,Y)=" "
0300 GOTO 0310
4500 SCROLL
4510 PRINT T#;"???" NAO ENTENDI "
:H#
4520 GOTO 0300
4530 IF H#<>"?" THEN GOTO 4700
4540 GOSUB 5000
4550 GOTO 0200
4700 IF H#<>"FIM" THEN GOTO 4500
4710 SCROLL
4720 PRINT "HA HA HA - GANHEI"
4740 STOP
4750 RUN
4800 GOSUB 5000
4805 SCROLL
4810 PRINT T#;"???" ;H#;" NAO FA
Z SENTIDO."
4820 GOTO 0300
5000 REM MOSTRA TABULEIRO

```

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)


```

5010 CLS
5030 PRINT TAB 10;"XXXXXXXXXX"
5040 GOSUB 5500
5050 FOR Y=8 TO 1 STEP -1
5060 PRINT TAB 5;Y;" ";
5070 FOR X=1 TO 8
5080 PRINT A$(X,Y);
5090 NEXT X
5100 PRINT " ";Y
5110 PRINT TAB 6;" ";
5120 IF A$(1,Y)("<>") THEN PRINT
" "
5130 PRINT "███ ███ ███ ███"
5140 NEXT Y
5500 PRINT TAB 7;"A B C D E F G
H"
5510 RETURN
6000 REM XXXXXXXXXX JOGA
6010 LET J=0
6020 FOR Y=1 TO 8
6030 FOR X=1 TO 8
6040 IF A$(X,Y,1 TO 1)("<>")"X" THEN
GOTO 6140
6050 LET J=J+1
6060 LET P(J)=0
6070 LET R$(J)=CHR$( X+37)+CHR$(
(Y+28)
6075 LET L(J)=2
6080 LET E=X
6090 LET H=Y
6100 IF A$(X,Y,2 TO 2)="X" THEN
GOTO 6130
6110 GOSUB 7000
6120 GOTO 6135
6130 GOSUB 8000
6135 REM SCROLL
6135 REM PRINT "P(";J;")=");P(J);
"R#=");R$(J);","
6140 NEXT X
6150 NEXT Y
6155 REM PRINT
6160 IF J("<>") THEN GOTO 6300
6170 GOSUB 5000
6180 PRINT "OK - ESSA EU PERDI"
6190 SCROLL
6200 PRINT "A PROXIMA, QUEM SABE
?"
6210 STOP
6220 RUN
6300 LET U=1
6310 IF J=1 THEN GOTO 6400
6320 FOR I=2 TO J

```

Figura 10.2

LISTAGEM DO PROGRAMA - FONTE (DAMAS) (cont.)

```

6330 IF P(I)("<P(U) THEN GOTO 6350
6335 IF P(I)(">P(U) THEN GOTO 6340
6335 IF RND<.5 THEN GOTO 6350
6340 LET U=I
6350 NEXT I
6400 IF L(U)("<4 THEN GOTO 6170
6410 LET H#=R$(U,1 TO L(U))
6420 IF J("<4 THEN GOSUB 8500
6450 LET X#="0"
6460 LET O#="X"
6470 LET J=0
6480 LET U=-1
6500 GOTO 2030
7000 REM MOVE PEAO

```

```

7010 LET S=0
7020 LET O#="0"
7030 LET X#="X"
7040 LET TK=7100
7050 LET DM=10
7100 LET K=0
7105 FOR I=-1 TO 1 STEP 2
7105 FOR T=-1 TO 1 STEP 2
7110 LET K=K+1
7115 LET F(K)=0
7115 LET F$(K)=" "
7120 LET G(K)=0
7121 LET H=E
7122 LET A=E
7123 LET B=H
7124 LET N=H
7125 GOSUB 7500
7130 LET G(K)=-G(K)
7140 LET A=E+I
7150 LET B=H+T
7160 IF A("<1 OR A(">8 OR B("<1 OR B(">8
THEN GOTO 7270
7180 IF A$(A,B,1 TO 1)="X" THEN
GOTO 7270
7190 IF A$(A,B)("<>") THEN GOTO
7210
7200 IF T(">0 THEN GOTO 7270
7205 GOTO 7250
7210 LET B=B+T
7220 LET A=A+I
7230 IF A("<1 OR A(">8 OR B("<1 OR B(">8
THEN GOTO 7270
7240 IF A$(A,B)("<>") THEN GOTO
7270
7243 LET G(K)=G(K)+9
7244 LET U=(E+A)/2
7245 LET V=(H+B)/2

```

Figura 10.2

LISTAGEM DO PROGRAMA - FONTE (DAMAS) (cont.)

```

7245 IF A$(U,U,2 TO 2)="0" THEN
LET G(K)=G(K)+2
7247 LET F(K)=1
7250 LET G(K)=G(K)+10-INT (B/3)
7260 LET F$(K)=CHR$(A+37)+CHR$(
(B+28)
7262 LET M=A-I
7264 LET N=B-T
7265 GOSUB 7500
7270 NEXT T
7272 NEXT I
7275 LET K=1
7280 FOR I=2 TO 4
7285 IF G(I)<G(K) THEN GOTO 7295
7290 IF G(I)>G(K) THEN GOTO 7295
7292 IF RND<.5 THEN GOTO 7295
7295 LET K=I
7298 NEXT I
7300 IF G(K)=0 THEN RETURN
7301 IF S=1 AND F(K)=0 THEN RETU
RN
7303 IF F$(K)<>" " THEN GOTO 73
08
7304 LET G(K)=0
7305 GOTO 7275
7308 FOR I=1 TO L(J) STEP 2
7310 IF R$(J,I TO I+1)=F$(K) THE
N RETURN
7312 NEXT I
7318 LET R$(J)=R$(J,1 TO L(J))+F
$(K)
7320 LET L(J)=L(J)+2
7330 LET P(J)=P(J)+G(K)
7340 IF F(K)=0 THEN RETURN
7350 LET E=(CODE F$(K))-3
7360 LET H=(CODE F$(K,2 TO
S
7370 LET S=1
7400 GOTO TK

7500 FOR Z=-1 TO 1 STEP 2
7510 FOR O=-1 TO 1 STEP 2
7520 LET U=A-Z
7530 LET V=B-O
7535 IF U=M AND V=N THEN GOTO 75
90
7540 IF U<1 OR U>8 OR V<1 OR V>8
THEN GOTO 7660
7550 IF A$(U,U,1 TO 1)<>" " THEN
GOTO 7660
7590 LET U=A+Z
7600 LET V=B+O

```

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

```

7610 IF U<1 OR U>8 OR V<1 OR V>8
THEN GOTO 7660
7650 IF A$(U,U,1 TO 1)=0$ THEN G
OTO 7690
7660 NEXT O
7670 NEXT Z
7680 RETURN
7690 LET G(K)=G(K)-DM
7700 RETURN
8000 REM MOVE DAMA
8001 REM VERSAO QUEBRA GALHO
8010 LET S=0
8020 LET O$="0"
8030 LET X$="X"
8040 LET TK=8100
8050 LET DM=15
8100 LET K=0
8105 FOR I=-1 TO 1 STEP 2
8108 FOR T=-1 TO 1 STEP 2
8110 LET K=K+1
8115 LET F(K)=0
8118 LET F$(K)=" "
8120 LET G(K)=0
8121 LET H=E
8122 LET A=E
8123 LET B=H
8124 LET N=H
8125 GOSUB 7500
8130 LET G(K)=-G(K)
8140 LET A=A+I
8150 LET B=B+T
8160 IF A<1 OR A>8 OR B<1 OR B>8
THEN GOTO 8270
8180 IF A$(A,B,1 TO 1)="X" THEN
GOTO 8270
8190 IF A$(A,B)=" " THEN GOTO 8
250
8210 LET B=B+T
8220 LET A=A+I
8230 IF A<1 OR A>8 OR B<1 OR B>8
THEN GOTO 8270
8240 IF A$(A,B)<>" " THEN GOTO
8270
8243 LET G(K)=G(K)+12
8244 LET U=(E+A)/2
8245 LET V=(H+B)/2
8246 IF A$(U,U,2 TO 2)="0" THEN
LET G(K)=G(K)+2
8247 LET F(K)=1
8250 LET G(K)=G(K)+12
8260 LET F$(K)=CHR$(A+37)+CHR$(
(B+28)

```

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

```

0060 LET N=A-I
0064 LET N=B-T
0068 GOSUB 7500
0070 NEXT T
0072 NEXT I
0075 GOTO 7275

0500 LET K=0
0510 LET U=0
0520 FOR A=1 TO 8
0530 FOR B=1 TO 8
0540 IF A#(A,B,1 TO 1) <>"X" THEN
  GOTO 0570
0550 IF A#(A,B,2 TO 2)="X" THEN
  LET U=U+1
0560 LET U=U+1
0570 IF A#(A,B,1 TO 1) <>"0" THEN
  GOTO 0600
0580 IF A#(A,B,2 TO 2)="0" THEN
  LET K=K+1
0590 LET K=K+1
0600 NEXT B
0610 NEXT A
0620 IF U=K THEN RETURN
0630 GOSUB 5000
0640 PRINT "ASSIM VAI FICAR MUIT
O MONOTONO."
0650 PRINT "PELO NUMERO DE PONTO
0."
0670 IF U>K THEN GOTO 4710
0680 SCROLL
0690 GOTO 6180
0700 SCROLL
0810 IF EU=0 THEN GOTO 2000
0820 PRINT "ALGO DE POORE NO PRO
GRAMA."
0830 STOP
0840 RUN
9000 FAST
9010 CLEAR
9020 SAVE "DAMAS"
9030 GOTO 1

```

Figura 10.2

LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

CAPÍTULO II

Jogo da Velha Tradicional

A FICHA

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:
Programa <input checked="" type="checkbox"/>	VELHA	1	Ø	22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
SINCLAIR ZX 8Ø <input checked="" type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>		
SINCLAIR ZX 81 <input type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-2ØØ <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
<u>RINGO R-47Ø e AS-1ØØØ</u>				
k) Observações: _____				

B MANUAL DE UTILIZAÇÃO

Você não conhece o jogo da velha? Não acredito! Em todo caso, resumirei as regras.

O jogo da velha joga-se entre duas pessoas em um tabuleiro, como na Figura 11.1.

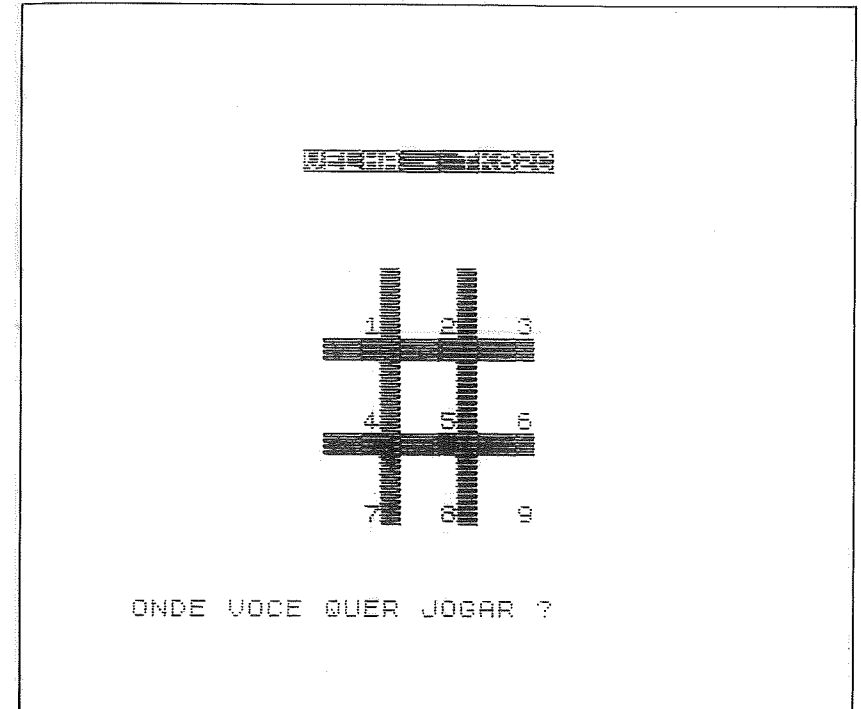


Figura 11.1 TABULEIRO DA VELHA

Um dos jogadores marca uma das casas com X e o outro com Ø. O objetivo é ligar 3 casas consecutivas na horizontal, vertical ou diagonal com marcas de um mesmo jogador, como na Figura 11.2.

Quando o tabuleiro fica sem ter onde jogar, diz-se que deu VELHA.

No nosso caso um dos jogadores é o computador que inicia, perguntando:

QUAL A SUA PEDRA?

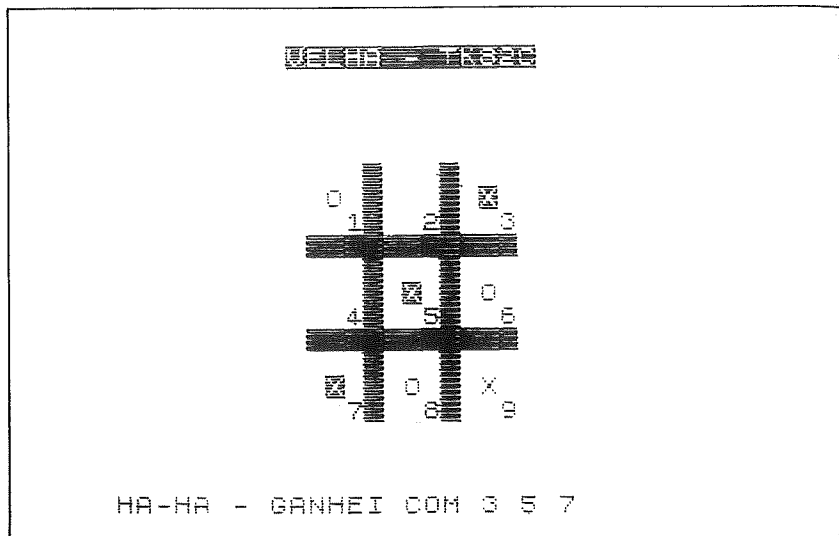


Figura 11.2 *EXEMPLO DE VITÓRIA*

Escolha entre \emptyset e X que dão melhor visual, contudo qualquer caractere pode ser escolhido.

Neste ponto você é informado, como na Figura 11.3.

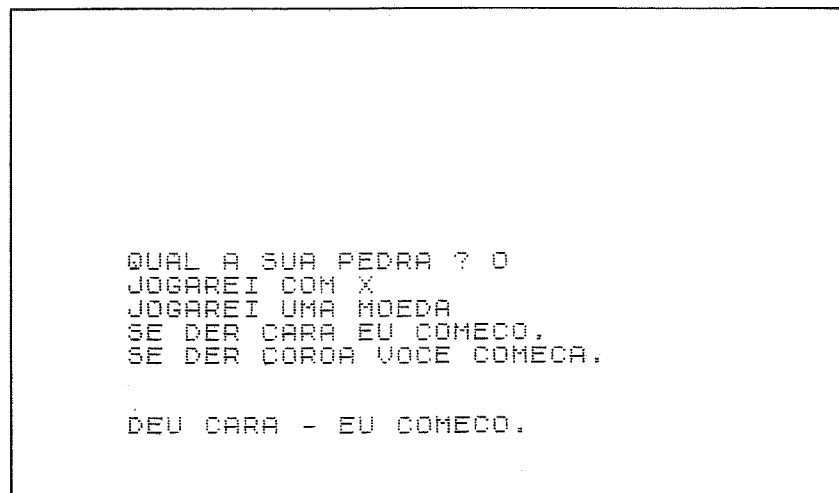


Figura 11.3 *TELA DE INICIAÇÃO*

Então é feito um sorteio, ao acaso, e uma das duas mensagens aparece:

DEU CARA – EU COMEÇO
DEU COROA – VOCE COMECA

Alguns segundos depois, a tela apaga-se e volta a seguir, como mostra a Figura 11.1.

Observe que cada posição do tabuleiro tem um número no canto inferior direito. Esse número identifica a posição. Se você, por exemplo, responder

3 NEWLINE

seu símbolo aparecerá no centro da posição 3 do tabuleiro, como mostra a Figura 11.4.

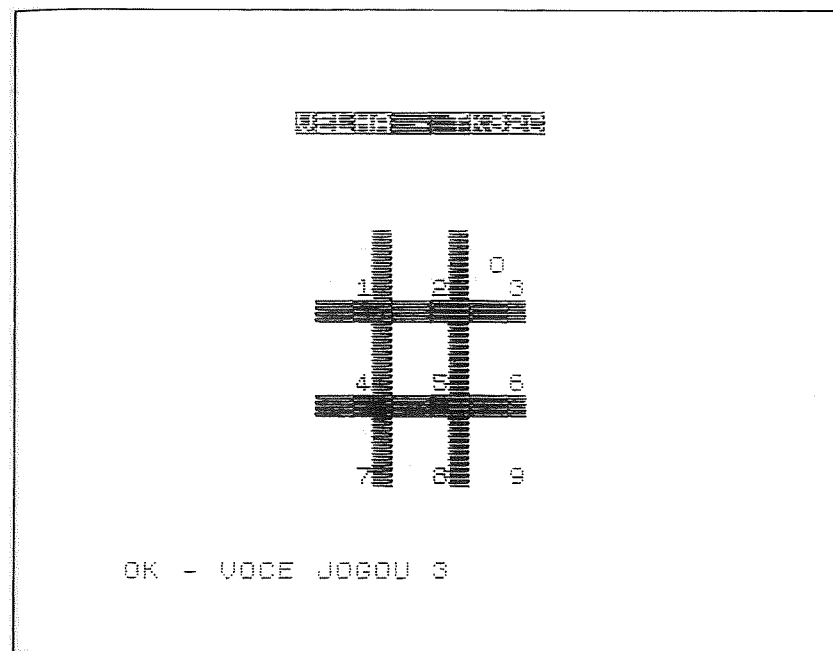


Figura 11.4 *JOGADA DE \emptyset NA POSIÇÃO 3*

Depois de você, ele joga. Depois você, e assim sucessivamente até que não haja mais onde jogar no tabuleiro.

Esta versão é composta de dois programas diferentes do jogo da velha – um sabido, que quase nunca perde, e o outro simples. Um deles é sorteado ao

acaso, no momento de cada jogada do computador. Inicialmente só o simples é chamado. Para cada vez que o computador perde, aumentam-se as chances do outro. Ou seja, dá-se a sensação de que o computador está aprendendo a jogar melhor com o passar do tempo.

C

MANUAL DE OPERAÇÃO

Para carregar-se o jogo da velha leva-se cerca de 2 minutos e meio de fita cassete. Ocupa cerca de 6K. Veja instruções sobre carregamento de programas nos Apêndices.

D

MANUAL TÉCNICO

Até a linha 490 temos a inicialização, sendo que na linha 300 temos o ponto de reinício, isto é, reinicialização do que foi modificado durante o jogo, para que uma nova partida possa ser jogada.

Em 500 inicia-se o controle sobre a jogada humana. Em 793 inicia-se a jogada do computador, juntamente com a análise de jogada humana. O processo é o seguinte:

- Se não há menos que 3 pedras no tabuleiro então:

Para cada alternativa (linha, coluna ou diagonal) conta-se o número de pedras (1 – humano, 4 – computador). Se há uma linha com 3 pedras humanas – acabou. Anota-se ocorrência de 2 pedras.

Ao fim de todas as alternativas, se ocorreram 2 pedras do computador – acabou.

Se ocorreram 2 pedras do humano, fecha-se-lhe a porta.
- Se há menos de 3 pedras no tabuleiro, então vai para 2000, ou seja:

se o humano joga em um canto, joga-se no canto oposto;

se o meio estiver ocupado, joga-se em uma central, se o canto estiver vazio,

se o humano não jogou em um canto, escolhe-se conforme o item *d*.
- Sorteia-se ao acaso. Se o número sorteado for maior que QI escolhe-se conforme o item *d*. Observe que QI cresce na medida em que o computador perde e vai chegar o momento de não haver transferência para o item *d* (veja 1000). Procura-se alternativas com 1 pedra. Se duas alternativas com 1 pedra humana se cruzam num vazio, então o jogador quer dar golpe – fecha-se-lhe a porta, jogando em uma alternativa onde já existe uma pedra

do computador. A melhor defesa é o ataque. Se duas alternativas com 1 pedra do computador se cruzam num vazio, joga-se lá para dar o golpe.

- Tenta-se o meio; se ocupado, tenta-se um canto. Se todos estiverem ocupados procura-se qualquer local.

Se você desejar modificar a velocidade de *aprendizado*, aumente ou diminua o incremento do QI na linha 1005. Lembre-se de que RND é sempre em número menor que 1, ou seja, não use incrementos maiores que 0.5. Se você quiser desligar esta característica, insira REM na linha 1400 (não apague a linha – assim você pode reativar o QI da VELHA).

Na Figura 11.5 encontra-se a listagem do programa-fonte.

```

10 REM VELHA ANTONIO JULIO LOS
SIO BOTELHO
20 DIM B(9)
30 DIM R(8,3)
35 DIM A(8)
40 FAST
50 LET K=1
55 LET QI=0
60 FOR I=1 TO 8
70 FOR J=1 TO 3
80 LET R(I,J)=VAL "12345678914
7288389159357" (K TO K)
90 LET K=K+1
100 NEXT J
110 NEXT I
120 SLOW
130 CLS
150 LET C#="X"
160 PRINT AT 21,0;"QUAL A SUA P
EDRA ? ";
170 INPUT H#
190 PRINT H#
200 IF H#=C# THEN LET C#="0"
210 SCROLL
215 IF H#="" OR LEN H#(>)1 THEN
GOTO 150
220 PRINT AT 21,0;"JOGAREI COM
";C#
230 SCROLL
240 PRINT AT 21,0;"JOGAREI UMA
MOEDA"
250 SCROLL
260 PRINT AT 21,0;"SE DER CARR
EU COMEÇO,"
265 SCROLL

```

Figura 11.5

LISTAGEM DO PROGRAMA-FONTE (VELHA)

```

266 PRINT AT 21,0;"SE DER COROA
VOCE COMECA."
270 LET V=INT (RND*2)
275 REM LET V=1
280 SCROLL
285 SCROLL
286 SCROLL
290 PRINT AT 21,0;"DEU "; ("DARA
- EU COMECO." AND V=0); ("COROA
- VOCE COMECA." AND V=1)
295 PAUSE 500
300 FOR I=1 TO 9
310 LET B(I)=0
320 NEXT I
330 FAST
335 CLS
340 PRINT AT 2,9;"SE DER COROA

346 LET CC=0
350 FOR I=1 TO 3
360 FOR J=1 TO 3
370 PRINT AT 4*I+2+J,13;"■ ■"
380 NEXT J
390 IF I<>3 THEN PRINT AT 4*I+6
,10;"SE DER COROA"
400 NEXT I
410 FOR I=0 TO 8
420 PRINT AT 9+4*INT (I/3),12+4
*(I-3*INT (I/3));I+1
430 NEXT I
435 SLOW
440 IF V=0 THEN GOTO 1470

500 GOSUB 7000
502 GOSUB 8500
505 PRINT AT 21,0;"ONDE VOCE QU
ER JOGAR ?"
510 INPUT M
515 GOSUB 7000
520 IF M>0 AND M<10 THEN GOTO 5
60
530 PRINT AT 21,0;"VOCE DEVE RE
SPONDER ENTRE 1 E 9";
540 PAUSE 200
550 GOTO 500
560 IF B(M)=0 THEN GOTO 770
750 PRINT AT 21,0;M;" ESTA OCUP
ADO."
760 GOTO 540
770 LET B(M)=1
775 PRINT AT 21,0;"OK - VOCE JO
GOU ";

```

Figura 11.5

LISTAGEM DO PROGRAMA-FONTE (VELHA) (cont.)

```

780 LET M#=H#
790 GOSUB 8000
791 GOSUB 8500
792 IF CC<3 THEN GOTO 2000
793 LET EU=0
795 LET TU=0
800 FOR I=1 TO 8
810 LET A(I)=B(R(I,1))+B(R(I,2)
)+B(R(I,3))
820 IF A(I)=3 THEN GOTO 1000
830 IF A(I)=8 THEN LET EU=I
840 IF A(I)=2 THEN LET TU=I
850 NEXT I
860 IF EU=0 THEN GOTO 1200
865 LET I=EU
870 FOR J=1 TO 3
880 LET M=R(I,J)
890 IF B(M)=0 THEN GOTO 910
900 NEXT J
910 GOSUB 7000
915 PRINT AT 21,0;"HAM-HAM - JO
GO COM ";
920 LET M#=C#
930 GOSUB 8000
935 GOSUB 8500
940 GOSUB 7000
945 PRINT AT 21,0;"HA-HA - GANH
E COM";
950 GOTO 1020
1000 GOSUB 7000
1005 LET QI=QI+.2
1010 PRINT AT 21,0;"SNIFF - PERD
I COM ";
1020 LET M#=CHR# (CODE M#+128)
1030 FOR J=1 TO 3
1040 LET M=R(I,J)
1050 PRINT AT 21,16+2*J;" ";
1060 GOSUB 8000
1070 NEXT J
1080 PAUSE 100
1090 SCROLL
1100 LET V=V+0
1110 PRINT AT 21,0;"OUER OUTRA ?
(S/N)"
1120 INPUT A#
1130 IF A#="S" THEN GOTO 300
1140 IF A#="N" THEN STOP
1150 SCROLL
1160 PRINT AT 21,0;"RESPONDA S P
ARA OUTRA,"
1170 SCROLL
1180 PRINT AT 21,0;"RESPONDA N P

```

Figura 11.5

LISTAGEM DO PROGRAMA-FONTE (VELHA) (cont.)

```

ARA PARAR."
1130 GOTO 1120
1200 IF TU=0 THEN GOTO 1400
1210 LET I=TU
1220 GOSUB 7000
1230 PRINT AT 21,0;"BOA TENTATIVA
A-";
1240 FOR J=1 TO 3
1250 LET M=R(I,J)
1260 IF B(M)=0 THEN GOTO 1280
1270 NEXT J
1280 PRINT "JOGAREI COM ";
1290 LET M#=0#
1300 LET B(M)=4
1310 GOSUB 8000
1315 GOSUB 8500
1320 GOTO 500
1400 IF OIK=AND THEN GOTO 1470
1405 GOSUB 6500
1410 LET M=4
1420 GOSUB 3000
1430 IF M<>4 THEN GOTO M
1435 GOSUB 3400
1440 LET M=1
1450 GOSUB 3000
1460 IF M<>1 THEN GOTO M
1465 GOSUB 3400
1470 LET M=6
1480 IF B(M)=0 THEN GOTO 1600
1500 FOR I=1 TO 3 STEP 2
1510 FOR J=1 TO 3 STEP 2
1520 LET M=(I-1)*3+J
1530 IF B(M)=0 THEN GOTO 1600
1540 NEXT J
1550 NEXT I
1560 FOR M=1 TO 6
1570 IF B(M)=0 THEN GOTO 1600
1580 NEXT M
1600 GOSUB 7000
1610 PRINT AT 21,0;
1620 GOTO 1280
2000 IF M=2*INT (M/2) THEN GOTO
1400
2010 GOSUB 7000
2020 PRINT AT 21,0;
2030 LET M=10-M
2031 REM IF B(5)=0 THEN LET M=M-
1
2032 IF M=0 THEN LET M=2
2040 IF B(M)<>0 THEN GOTO 1470
2050 PRINT "E EU ";
2060 GOTO 1280

```

Figura 11.5

LISTAGEM DO PROGRAMA-FONTE (VELHA) (cont.)

```

3000 FOR I=1 TO 8
3020 LET A(I)=B(R(I,1))+B(R(I,2)
)+B(R(I,3))
3040 REM PRINT AT I+8,(M-1)*8;
"
3050 REM PRINT AT I+8,(M-1)*8;I
"-";A(I)
3060 IF A(I)<M THEN GOTO 3110
3065 IF A(I)>M+.5 THEN GOTO 3110
3070 IF A(I)>M THEN GOTO 3200
3080 FOR J=1 TO 3
3090 IF B(R(I,J))=0 THEN LET B(R
(I,J))=.125
3100 NEXT J
3110 NEXT I
3170 RETURN
3200 LET M=4000
3210 RETURN
3400 FOR I=1 TO 9
3410 IF B(I)=.125 THEN LET B(I)=0
3420 NEXT I
3430 RETURN
3600 LET M=R(I,J)
3610 GOSUB 7000
3620 PRINT AT 21,0;"VEJA SO - ";
3630 GOSUB 3400
3640 GOTO 1280
4000 GOSUB 7000
4010 IF A(I)<4 THEN GOTO 4100
4020 PRINT AT 21,0;"QUERO VER SO
VOCE SAIR DESSA."
4030 PAUSE 300
4040 FOR J=1 TO 3
4050 IF B(R(I,J))<1 THEN GOTO 36
00
4060 NEXT J
4070 SCROLL
4080 PRINT AT 21,0;"TO DOIDAO. "
4085 PAUSE 200
4090 RUN
4100 PRINT AT 21,0;"TA PENSAND O
UE SOU BURRO?"
4110 FOR I=1 TO 8
4120 IF INT A(I)=4 THEN GOTO 403
0
4130 NEXT I
4170 GOTO 1470
7000 SLOW
7010 PRINT AT 21,0;";

```

Figura 11.5

LISTAGEM DO PROGRAMA-FONTE (VELHA) (cont.)


```
7020 RETURN
8000 PRINT M
8010 LET K=M-1
8020 LET Q=INT (K/3)
8030 PRINT AT 8+4*Q,11+4*(K-3*Q)
;M#
8040 LET CC=CC+1
8050 PAUSE 150
8060 RETURN
8500 IF CC<9 THEN RETURN
8530 SLOW
8540 SCROLL
8550 PRINT AT 21,0;"DEU VELHA"
8560 GOTO 1080
9000 CLEAR
9010 FAST
9020 SAVE "VELHA"
9030 GOTO 10
```

Figura 11.5 LISTAGEM DO PROGRAMA-FONTE (VELHA) (cont.)

CAPÍTULO 12

**Aprenda
a Jogar
com Palitos**

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:
Programa <input checked="" type="checkbox"/>	PAR	1	φ	22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 80 <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>		SLOW <input checked="" type="checkbox"/>	
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>		FAST <input checked="" type="checkbox"/>	
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>		LPRINT <input type="checkbox"/>	
SINCLAIR ZX 80 <input type="checkbox"/>	64K <input type="checkbox"/>		LLIST <input type="checkbox"/>	
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>		COPY <input type="checkbox"/>	
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>		DLOAD <input type="checkbox"/>	
TIMEX 1000 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>		DSAVE <input type="checkbox"/>	
CP-200 <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>		DMLOAD <input type="checkbox"/>	
CP-300 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>		DMSAVE <input type="checkbox"/>	
APPLY 300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>	
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
RINGO R-470 e AS-1000				
k) Observações:				

Um número ímpar de objetos (palitos) é dado a dois oponentes que, alternadamente, vão retirando-os num máximo de 4 por vez. Vence quem terminar com um número PAR de objetos.

A pergunta

QUEM COMECA?

responda

EU, TU ou VOCE.

Neste ponto será desenhada a tela da Figura 12.1.

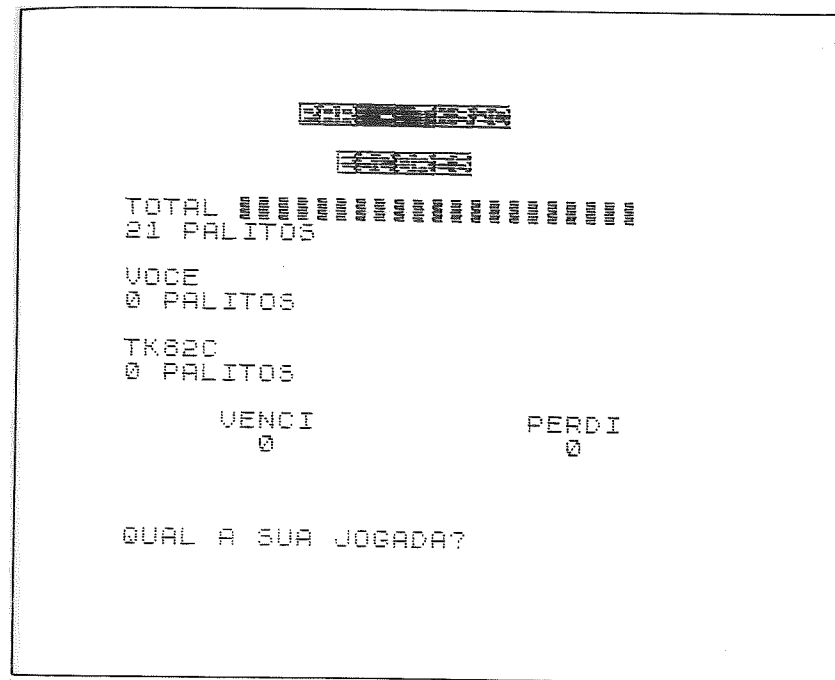


Figura 12.1 TELA DO JOGO PAR

Será então perguntado quantos palitos você deseja tirar. Você deve entrar um número de 1 a 4 como resposta. O objetivo é ficar com um número par de objetos no final.

Este jogo é semelhante ao do capítulo seguinte, PAR-C.

Na verdade o jogo é o mesmo. A diferença é na forma como o computador joga. No PAR é usado um algoritmo determinístico (veja manual técnico) que se torna literalmente impossível vencê-lo (é possível). No PAR-C você verá uma solução diferente.

C

MANUAL DE OPERAÇÃO

Tempo de carga: menos de 1 minuto e meio.

Memória: cerca de 3K.

Observe-se o descrito nos Apêndices para a carga a partir da fita cassete.

D

MANUAL TÉCNICO

O algoritmo utilizado nesta versão é muito eficiente, o que leva a uma certa monotonia, pois o computador ganha quase sempre.

1. Cálculo de $R \leftarrow$ resto da divisão por 6 do número de palitos P da pilha;
2. Se o adversário tem um número par de palitos, então:
 - 2.1 Se $R = 0$ ou 1
então: tira-se 1
senão: tira-se $R - 1$
senão:
 - 2.2 Se a pilha tem menos de 5 (de 1 a 4) palitos, então: tira-se tudo e ganha-se;
senão: se $R < 4$
então: tira-se $R + 1$
senão: se $R = 4$
então: tira-se 4
senão: tira-se 1

Na Figura 12.2 encontra-se a listagem do programa-fonte.

```
10 REM PAR
20 REM VERSAO PROGRAMADA
30 REM ANTONIO JULIO LOSSIO BOT
ELHO
40 LET P$=""
50 LET T$=" PALITOS "
60 FAST
70 LET EU=0
80 LET TU=0
90 LET A=1
100 CLS
110 PRINT
120 PRINT "QUEM COMECA?"
130 INPUT X$
140 IF X$="EU" THEN GOTO 190
150 IF X$="TU" THEN GOTO 200
160 IF X$="VOCE" THEN GOTO 200
170 PRINT "NAO DEU PARA ENTENDE
R"
180 GOTO 120
190 LET A=0
200 LET U=0
210 LET T=0
220 LET P=INT ((13*RND+13)/2) #2
+1
230 IF A=0 THEN GOTO 400
300 GOSUB 1000
320 IF P<1 THEN GOTO 650
330 GOSUB 5000
340 IF P<1 THEN GOTO 650
400 REM HUMANO JOGA
420 GOSUB 1000
430 LET X=18
440 PRINT AT X,0;"QUAL A SUA JO
GADA?"
450 INPUT J
455 IF J>4 THEN GOTO 470
460 IF J>0 AND J<=P THEN GOTO 5
00
470 PRINT AT 18,0;"TA BRINCANDO
???"
480 LET X=19
490 GOTO 440
500 IF J=P THEN GOTO 650
510 LET P=P-J
520 LET U=U+J
530 GOTO 300
650 IF T/2=INT (T/2) THEN GOTO
700
660 PRINT AT 19,0;"OK. ESTA VOC
E GANHOU."
```

Figura 12.2

LISTAGEM DO PROGRAMA-FONTE (PAR)

```

670 LET TU=TU+1
690 GOTO 800
700 PRINT AT 19,0;"VIU? GANHEI.
HA HA HA"
710 LET EU=EU+1

800 PRINT AT 20,0;"QUER OUTRA?
(S/N) "
805 LET A=A+0
810 INPUT X$
820 IF X$="N" THEN STOP
830 IF X$="S" THEN GOTO 200
840 PRINT AT 19,0;"ACHO QUE VOC
E NAO ENTENDEU."
850 PRINT AT 20,0;"ENTRE S DE S
IM OU N DE NFO"
860 PRINT AT 21,0;"CASO VOCE QU
EIRA JOGAR OUTRA"
870 GOTO 810
1000 CLS
1010 PRINT AT 0,9;"PAR = 1000"
1020 PRINT AT 2,11;"ESCORREO"
1030 PRINT AT 4,0;"TOTAL";P$(1 T
O P+1)
1040 PRINT AT 5,0;P;T$
1050 PRINT AT 7,0;"VOCE ";P$(1 T
O U+1)
1060 PRINT AT 8,0;U;T$
1070 PRINT AT 10,0;"TK820";P$(1
TO T+1)
1080 PRINT AT 11,0;T;T$
1090 PRINT AT 13,5;"VENCIA";AT 13
,21;"PERDI"
1100 PRINT AT 14,7;EU;AT 14,23;T
U
1110 RETURN
5000 REM TK JOGA
5010 LET R=P-5*INT (P/5)
5020 IF INT (U/2)=U/2 THEN GOTO
5200
5030 IF P<4.2 THEN GOTO 5080
5040 IF R>3.4 THEN GOTO 5100
5050 LET J=R+1
5060 GOTO 5300
5080 LET J=P
5090 GOTO 5300
5100 IF R<4.7 THEN GOTO 5150
5110 IF R>3.5 THEN GOTO 5150
5120 LET J=1
5140 GOTO 5300
5150 LET J=4
5160 GOTO 5300

```

```

5200 IF R<1.5 THEN GOTO 5120
5210 IF R>5.3 THEN GOTO 5120
5220 LET J=R-1
5300 LET T=T+J
5310 LET P=P-J
5320 PRINT AT 18,0;"MINHA JOGADA
:";J
5330 PAUSE 300
5340 RETURN
9000 FAST
9010 CLEAR
9020 SAVE "PAR"
9030 RUN

```

Figura 12.2 LISTAGEM DO PROGRAMA-FONTE (PAR) (cont.)

Figura 12.2 LISTAGEM DO PROGRAMA-FONTE (PAR) (cont.)

CAPÍTULO **13**

***Ensine ao
Computador
a Jogar Palitos***

A FICHA

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: PAR-C	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	SINCLAIR ZX 8Ø <input type="checkbox"/>	64K <input type="checkbox"/>	LPRINT <input type="checkbox"/>
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	LLIST <input type="checkbox"/>
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	CP-2ØØ <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	DLOAD <input type="checkbox"/>
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	CP-3ØØ <input type="checkbox"/>		DSAVE <input type="checkbox"/>
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input checked="" type="checkbox"/>		DMLOAD <input type="checkbox"/>
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			DMSAVE <input type="checkbox"/>
Ringo R-47Ø e AS-1ØØØ				
k) Observações:				

B MANUAL DE UTILIZAÇÃO

Dois jogadores, com uma pilha contendo um número ímpar de objetos, retiram alternadamente de 1 a 4 objetos. Ao final, com a pilha vazia, apenas um tem um número PAR de objetos retirados da pilha – este é o vencedor.

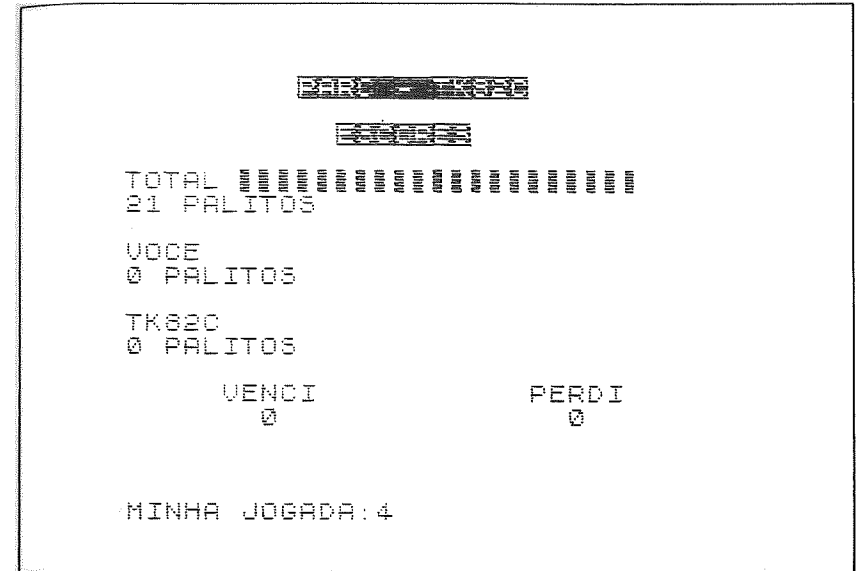


Figura 13.1 TELA DO JOGO PAR-C

Após o RUN NEWLINE ou LOAD, ser-lhe-á perguntado quantos palitos você deseja retirar: tecle um dígito de 1 a 4, seguido do NEWLINE. A tela é redesenhada. O computador joga. A tela é redesenhada, e a pergunta de quantos palitos é repetida até que haja um vencedor.

Este jogo – PAR-C – é semelhante ao descrito no capítulo anterior. Mais precisamente, como jogo, é idêntico.

O que difere PAR-C do PAR é a forma como o computador joga. Enquanto o PAR usa o algoritmo determinístico que quase sempre ganha, o PAR-C inicia o jogo sabendo apenas as regras e vai aprendendo à medida que vai jogando. Após cerca de 12 partidas, começa a ficar imbatível. Não se assuste. Basta um BREAK seguido de RUN NEWLINE que ele se esquece do que aprendeu.

Os jogadores muito experientes neste jogo gostam mais do PAR. A grande maioria porém prefere o PAR-C, pois participa mais, vence algumas vezes, além de sentir como o computador modifica suas respostas à medida que as jogadas vão se repetindo.

C MANUAL DE OPERAÇÃO

Carregável de fita em menos de 1 minuto e meio, ocupa cerca de 3K.

Observe os Apêndices sobre a carga a partir da fita cassete.

D MANUAL TÉCNICO

Uma matriz de 2 linhas por 6 colunas é utilizada para armazenar os resultados.

Esta matriz é indexada pelas seguintes informações:

- linhas → se o adversário tinha um número par ou ímpar;
- colunas → resto da divisão do total de palitos por 6.

Ele é inicializado com 4 em todas suas posições.

Sempre que perde, o computador presume que sua jogada não foi boa. Subtrai um na posição da matriz indicada pelo último lance. Se já era 1, subtrai da matriz na posição indicada no penúltimo lance (se maior que 1). Se a matriz indicar que deve ser retirado mais que o número de palitos existentes, todos são retirados.

Na Figura 13.2 encontra-se a listagem do programa-fonte.

```

10 REM PAR
20 REM VERSAO CYBER
30 REM ANTONIO JULIO LOSSIO BOT
ELHO
40 LET P$="
50 LET T$=" PALITOS "
60 FAST
70 LET EU=0
80 LET TU=0
90 DIM M(2,6)
100 LET L=1

```

Figura 13.2

LISTAGEM DO PROGRAMA-FONTE (PAR-C)

```

110 LET E=1
120 FOR I=1 TO 6
130 FOR J=1 TO 2
140 LET M(J,I)=4
150 NEXT J
160 NEXT I
200 LET U=0
210 LET T=0
220 LET P=INT ((13*RND+13)/2) *2
+1
300 GOSUB 1000
310 LET E1=E
320 LET L1=L
330 LET E=(U/2-INT (U/2))*2+1
340 LET L=INT (P-(INT (P/6))*6+
1)
350 LET J=M(E,L)
360 IF J>=P THEN GOTO 600
370 IF J<=0 THEN GOTO 660
380 PRINT AT 18,0;"MINHA JOGADA
";J
390 PAUSE 300
400 LET P=P-J
410 LET T=T+J
420 GOSUB 1000
430 LET X=18
440 PRINT AT X,0;"QUAL A SUA JO
GADA?"
450 INPUT J
455 IF J>4 THEN GOTO 470
460 IF J>0 AND J<=P THEN GOTO 5
00
470 PRINT AT 18,0;"TA BRINCANDO
???"
480 LET X=19
490 GOTO 440
500 IF J=P THEN GOTO 650
510 LET P=P-J
520 LET U=U+J
530 GOTO 300
600 PRINT AT 18,0;"MINHA JOGADA
";P
620 LET M(E,L)=P
630 LET T=T+P
640 LET P=0
650 IF T/2=INT (T/2) THEN GOTO
700
660 PRINT AT 19,0;"OK. ESTA VOC
E GANHOU."
665 LET TU=TU+1
670 IF M(E,L)=1 THEN GOTO 900
680 LET M(E,L)=M(E,L)-1

```

Figura 13.2

LISTAGEM DO PROGRAMA-FONTE (PAR-C) (cont.)

```

690 GOTO 800
700 PRINT AT 19,0;"UIU? GANHEI,
HA HA HA"
710 LET EU=EU+1

800 PRINT AT 20,0;"QUER OUTRA?
(S/N) "
810 INPUT X#
820 IF X#="N" THEN STOP
830 IF X#="S" THEN GOTO 200
840 PRINT AT 18,0;"ACHO QUE VOC
E NAO ENTENDEU."
850 PRINT AT 19,0;"CASO VOCE QU
EIRA JOGAR OUTRA"
860 PRINT AT 20,0;"ENTRE S DE S
IM, CASO CONTRARIO ENTRE N DE N
AO."
870 GOTO 810
900 IF M(E1,L1)=1 THEN GOTO 800
910 LET M(E1,L1)=M(E1,L1)-1
920 GOTO 800
1000 CLS
1010 PRINT AT 0,9;"PARC = TK82C"
1020 PRINT AT 2,11;"SCORES"
1030 PRINT AT 4,0;"TOTAL";P$(1 T
O P+1)
1040 PRINT AT 5,0;P;T$
1050 PRINT AT 7,0;"VOCE ";P$(1 T
O U+1)
1060 PRINT AT 8,0;U;T$
1070 PRINT AT 10,0;"TK82C";P$(1
TO T+1)
1080 PRINT AT 11,0;T;T$
1090 PRINT AT 13,5;"VENCEI";AT 13
,21;"PERDI"
1100 PRINT AT 14,7;EU;AT 14,23;T
U
1110 RETURN
2000 CLS
2010 PRINT
2020 FOR I=1 TO 9
2030 PRINT M(1,I),M(2,I)
2040 NEXT I
2050 INPUT X#
2060 GOTO 200
9000 FAST
9010 CLEAR
9020 SAVE "PAR-C"
9030 GOTO 10

```

Figura 13.2

LISTAGEM DO PROGRAMA-FONTE (PAR-C) (cont.)

CAPÍTULO 14

Uma Aventura Espacial

A FICHA

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: AVENTURA ESPACIAL	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/81
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
SINCLAIR ZX 8Ø <input type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>		
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETTE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-2ØØ <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
RINGO R-47Ø e AS-1ØØØ				
k) Observações:				

B MANUAL DE UTILIZAÇÃO

A série de televisão Jornada nas Estrelas (*STARTREK*), além de ser precursora de vários outros filmes do gênero, foi inspiradora de vários jogos para computador.

A série de TV e filmes de cinema tratam das aventuras do Capitão James Kirk, comandante da Nave Estelar USS ENTERPRISE e dos seus auxiliares, entre eles o Sr. Spock. A maioria dos filmes da série descreve sua luta contra os *Klingons*, povo de outra civilização, que invade a *Galáxia*, então controlada pela *Federação dos Planetas*, organização nos moldes terráqueos.

A USS ENTERPRISE é uma nave de combate equipada com *torpedos fotônicos* e *phasers*. Protege-se com um eficiente campo de força denominado de *telas defletoras* ou simplesmente *defletores*. Viaja através das *dobras espaciais*. Essas *dobras* têm comportamento geométrico e dividem a Galáxia em *quadrantes*. Os *quadrantes* ficam divididos em *setores*.

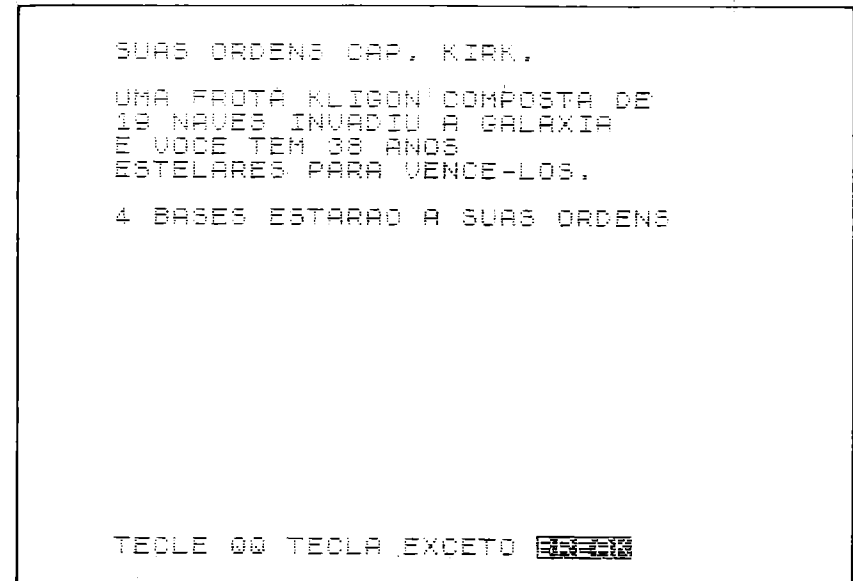


Figura 14.1 TELA DE ABERTURA

A região defendida pela USS ENTERPRISE no jogo é um quadrado composto de 64 (8 x 8) quadrantes (Figura 14.2).

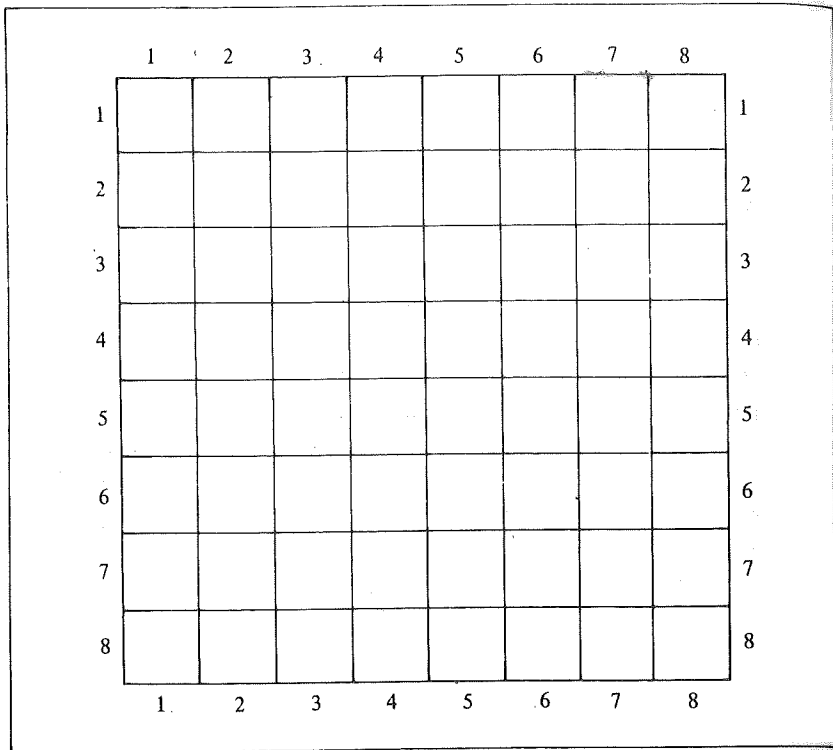


Figura 14.2
RETICULADO DAS DOBRAS ESPACIAIS

Cada quadrante está dividido em 64 setores (Figura 14.2). Dentro de um setor a USS ENTERPRISE move-se de forma lenta. As *dobras intra-setores* são aceleradoras. Estrelas, naves ou bases em setores na direção do movimento da nave tornam-se obstáculos intransponíveis. Já as *dobras intraquadrantes* são desmaterializantes, isto é, ao penetrar nas *dobras* intraquadrantes, a USS ENTERPRISE chegará ao destino sem riscos de choque.

No início do jogo você recebe informações sobre a frota invasora, quantas bases estelares estão disponíveis e quanto tempo você tem para destruir os Klingons antes que estes armem uma segunda e mais poderosa frota, na Figura 14.1.

Os Klingons estão armados apenas de *phasers*. A segunda frota já terá *torpedos fotônicos*. Portanto, se você não vencê-los antes da segunda leva invasora, a Federação terá que se render incondicionalmente. Após ler as

instruções (Figura 14.1), teclê qualquer tecla (exceto a tecla BREAK). A tela se apagará – a USS ENTERPRISE está viajando para iniciar a luta. Irá materializar-se em um ponto qualquer da Galáxia.

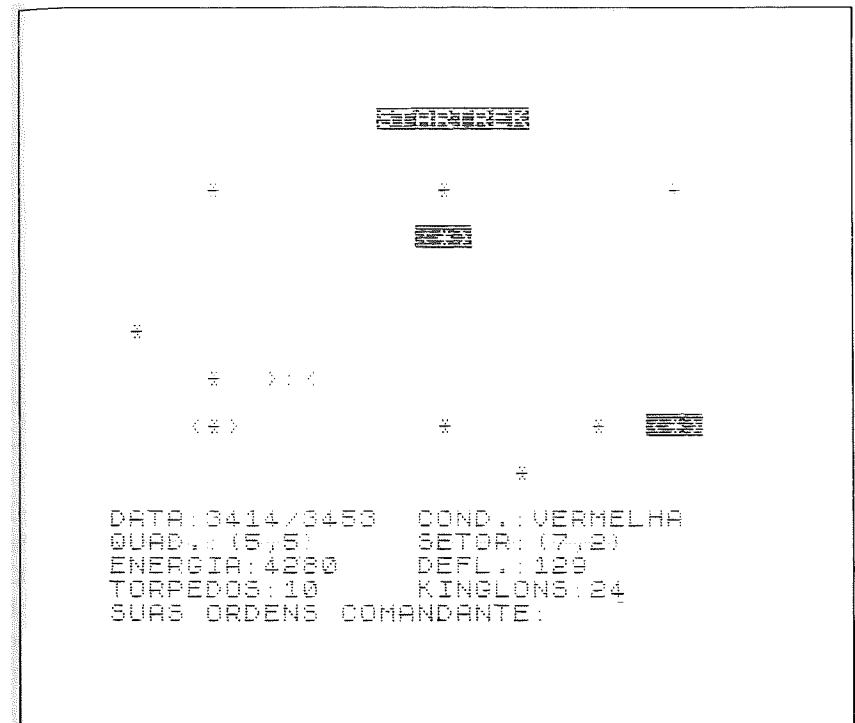


Figura 14.3
TELA DO VISOR

O quadrante onde se encontra a USS ENTERPRISE pode ser visto pelo VISOR, como no exemplo da Figura 14.3. Nesta figura você pode identificar as estrelas (asteriscos), a sua nave – < * > – e as naves Klingons – *reverso* (fundo *preto*) de < * >.

A USS ENTERPRISE está equipada com SENSORES REMOTOS, que descrevem os quadrantes vizinhos àquele onde ela se encontra.

O SENSOR informa números de 3 dígitos que descrevem o quadrante na forma EKB, isto é, o primeiro dígito é o número de estrelas, o segundo o número de naves Klingons e o terceiro o número de bases. Na Figura 14.4, o quadrante do centro é o mesmo visto no VISOR da Figura 14.3.

```

          SENSOR REMOTO

          400 900 900

          900 821 300

          700 920 720

DATA: 3414/3453  COND.: VERMELHA
QUAD.: (5,5)   SETOR: (7,2)
ENERGIA: 4280  DEFL.: 129
TORPEDOS: 10  KINGLONS: 24
SUAS ORDENS COMANDANTE:

```

Figura 14.4 TELA DO SENSOR

Sempre que o SENSOR é ativado, sua resposta é memorizada pelo COMPUTADOR de bordo, que pode lhe informar a situação da Galáxia – Figura 14.5. Observe porém que:

- Informação do COMPUTADOR é a obtida para o quadrante pelo último SENSOR emitido naquele quadrante ou seu vizinho, ou seja, se após a emissão do SENSOR uma base ou nave Klingon for destruída, esta informação não é posta no banco de dados do COMPUTADOR;
- Quadrantes não pesquisados anteriormente por SENSOR aparecem em -11.

Quem defende a USS ENTERPRISE dos PHASERS dos Klingons são as TELAS DEFLETORAS. Viajar com as TELAS carregadas consome muita energia. Viajar com as TELAS descarregadas pode ser perigoso. Uma carga de PHASERS nesse momento seria fatal. *Se a carga dos DEFLETORES for a zero a USS ENTERPRISE explode.*

As fontes de energia dos DEFLETORES e a propulsora são intercambiáveis, isto é, ao aumentar a carga dos defletores, você está diminuindo sua energia disponível para viajar, e vice-versa.

```

          900 500 300 900 500 410 -11 -11
          731 900 501 600 900 900 -11 -11
          -11 -11 -11 -11 -11 -11 -11 -11
          400 500 500 400 900 900 -11 -11
          600 600 500 900 821 300 -11 -11
          500 900 500 730 920 720 -11 -11
          -11 -11 -11 -11 -11 -11 -11 -11
          -11 -11 -11 -11 -11 -11 -11 -11

DATA: 3414/3453  COND.: VERMELHA
QUAD.: (5,5)   SETOR: (7,2)
ENERGIA: 4280  DEFL.: 129
TORPEDOS: 10  KINGLONS: 24
SUAS ORDENS COMANDANTE:

```

Figura 14.5 RELATÓRIO DO COMPUTADOR

Os PHASERS são armas poderosas, porém consomem muita energia e sua eficiência depende da distância. Os PHASERS são controlados pelo COMPUTADOR e sempre acertam o alvo, porém a força com que atingem depende também de interferências cósmicas que podem diminuir-lhe ou potencializar-lhe o efeito.

```

          RELATORIO DE DANOS

          EQUIPAMENTO      DANOS

          MOTOR            -1
          VISOR             0
          SENSOR           0
          FASER            0
          TUBO TORP.       0
          ENGENHARIA       0
          DEFLETORES       0
          COMPUTADOR       0

DATA: 3414/3453  COND.: VERMELHA
QUAD.: (5,5)   SETOR: (7,2)
ENERGIA: 4280  DEFL.: 129
TORPEDOS: 10  KINGLONS: 24
SUAS ORDENS COMANDANTE:

```

Figura 14.6 RELATÓRIO DE DANOS/ENGENHARIA

O uso de *PHASERS* automaticamente fecha o VISOR e o controle passa ao COMPUTADOR que apenas informa os efeitos (Figura 14.8), indicando a localização do KLINGON atingido, quanto de energia acertou o alvo e quanto as telas defletoras do inimigo ainda podem resistir. E se foi destruído. Como durante a emissão de *PHASERS* a USS ENTERPRISE está sob controle do COMPUTADOR, os efeitos de eventuais explosões de algum KLINGON são compensados automaticamente.

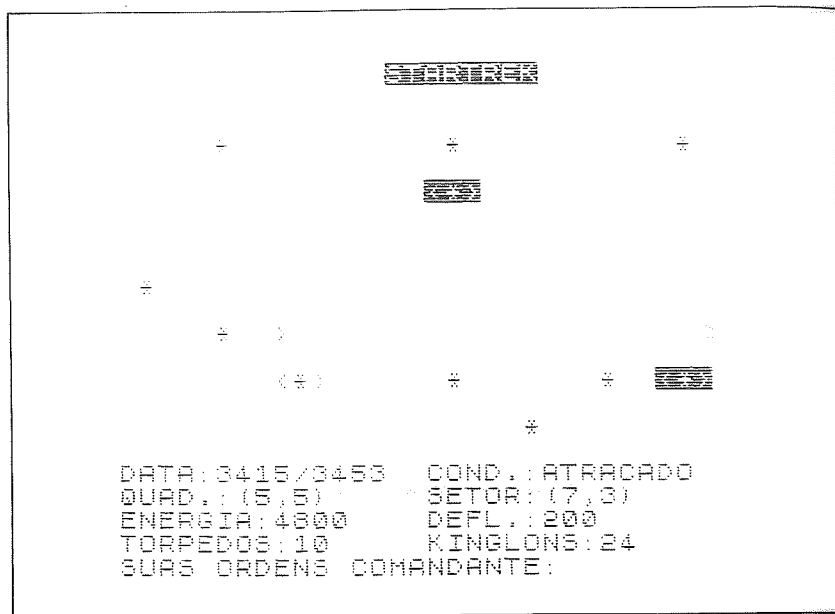


Figura 14.7 NAVE APORTADA

O relatório de acompanhamento do PHASERS informa o SETOR onde se encontra a nave KLINGON atingida, qual a carga de energia que atingiu suas TELAS DEFLETORAS, qual a capacidade de resistência que elas ainda possuem. Se a nave KLINGON foi destruída, a coluna CONDIÇÃO conterà esta informação.

Ao receber um ataque Klingon, a USS ENTERPRISE tem seus DEFLETORES consumidos. Se a carga destes for inferior a 2000 você estará sujeito a avarias variadas. Outra forma da USS ENTERPRISE sofrer avarias é se, durante uma viagem, encontrar uma TEMPESTADE CÓSMICA. A ENGENHARIA pode lhe fornecer um relatório de danos, como na Figura 14.6. Se o equipamento

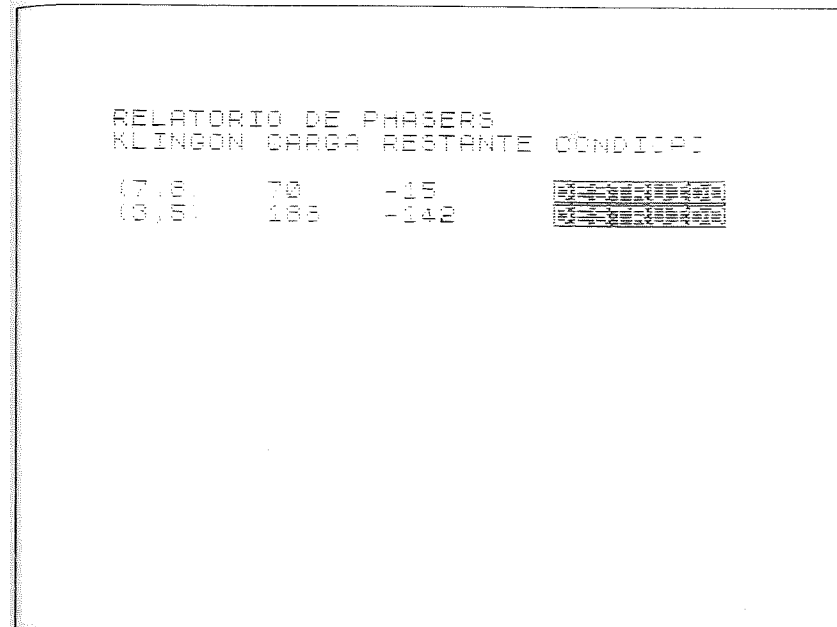


Figura 14.8 RELATÓRIO DE PHASERS

estiver normal, o relatório informa 0. Um número negativo indica o tempo (em anos estelares) necessário para o conserto.

Se você se aproximar de uma BASE – visual no VISOR como < : > – então pode atracar (Figura 14.7). Estando atracado, todas as avarias são consertadas. Os TORPEDOS são renovados e os bancos de energia substituídos.

Os TORPEDOS FOTÔNICOS são armas poderosíssimas. As TELAS DEFLETORAS são absolutamente inoperantes contra os TORPEDOS. Porém estes são carentes de precisão ao alvo. O tubo de lançamento tem que ser apontado sem interferência do COMPUTADOR.

Na Figura 14.9 você pode ver a tabela de direções de CURSO da USS ENTERPRISE (é o que na Terra se usava até o século 20 para medir ângulos em grau). A direção varia continuamente de 0 a 360 e é chamada de CURSO.

Escolhido o CURSO do TORPEDO, este é disparado. Se o VISOR estiver ativo, sua trajetória é rastreada visualmente (Figura 14.10) por um quadro escuro piscando.

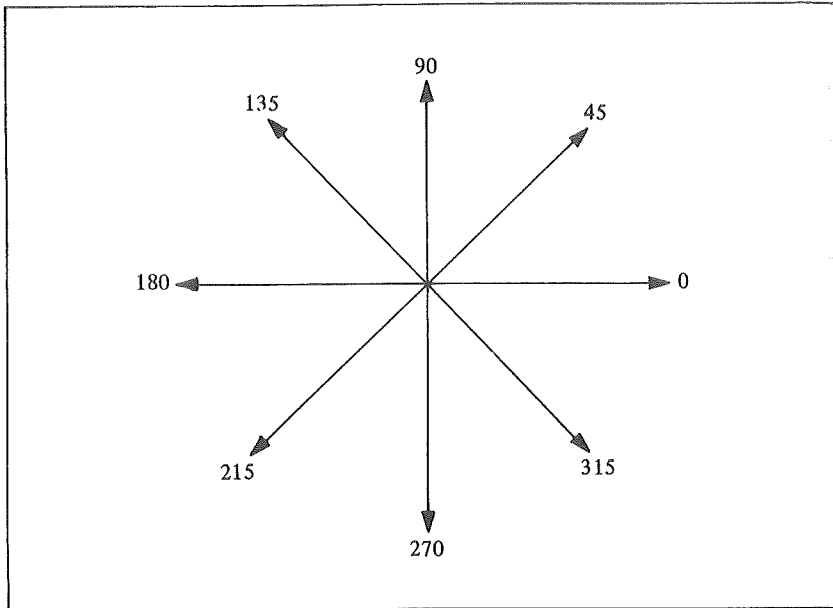


Figura 14.9
TABELA DE DIREÇÕES

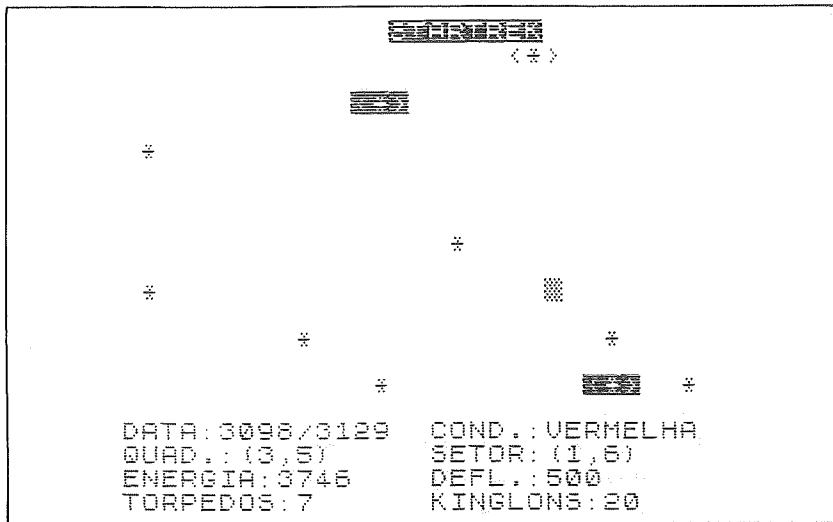


Figura 14.10
RASTREAMENTO DO TORPEDO PELO VISOR

Se o VISOR não estiver ativo, o rastreamento é feito pelo COMPUTADOR (Figura 14.11) desde que este não esteja avariado.

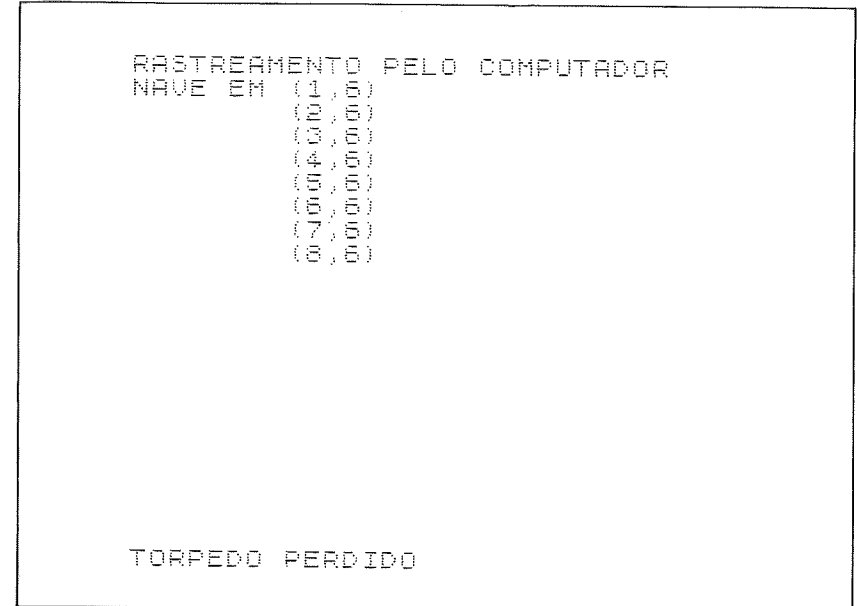


Figura 14.11
RASTREAMENTO DO TORPEDO PELO COMPUTADOR

Nesse caso você receberá como informação as coordenadas de onde passa o TORPEDO. A primeira linha é coincidente com a nave. Na medida em que o TORPEDO se desloca para um outro SETOR, as coordenadas deste aparecem na tela.

Para viajar, basta definir o CURSO, como descrito para os TORPEDOS, e o fator de DOBRA a ser utilizado. Um fator de DOBRA .1 faz com que a USS ENTERPRISE se mova no setor em que se encontra. DOBRA .2 faz ela se mover para o setor imediatamente vizinho, mesmo que seja em outro quadrante. Pode variar continuamente de .1 a 8.

Aqui vale a observação: o programa quando pede o valor do *fator dobra* diz 1 a 8. Na verdade é maior que zero e menor ou igual a 8. Isto é, 0.1 é uma resposta válida. É aconselhável, devido a características de arredondamento do TK, que você sempre use o ponto decimal ao responder CURSO, CURSO DE TORPEDO e FATOR DOBRA.

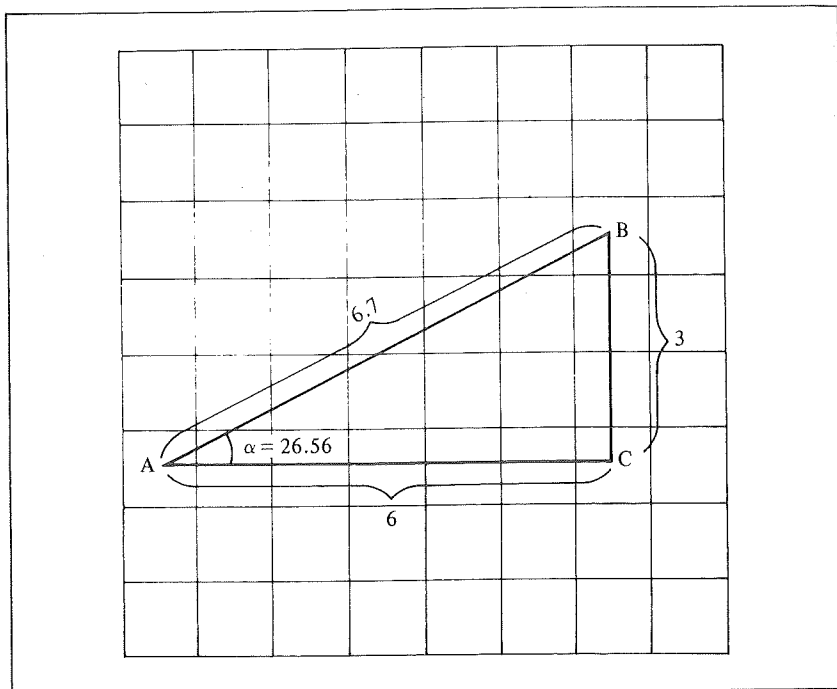


Figura 14.12 EXEMPLO DE CÁLCULO DE CURSO E DOBRA

O que vamos descrever a seguir aplica-se também ao cálculo de CURSO DO TORPEDO.

Suponha que do Setor A você queira mover-se para o setor B da Figura 14.12.

Um pouco de geometria e trigonometria ajuda no cálculo correto do CURSO e da DOBRA. O teorema de Pitágoras pode ser aplicado para calcular-se a DOBRA, que vem a ser a hipotenusa do triângulo ABC. O ângulo pode ser calculado pela função ARCOTANGENTE.

$$\text{CURSO} = \text{ARCTG} \left(\left| \frac{y_B - y_A}{x_B - x_A} \right| \right)$$

$$\text{DOBRA} = \sqrt{(y_A - y_B)^2 + (x_A - x_B)^2}$$

A fórmula do CURSO nos levará a um valor entre 0 e 90 (90 não, pois TAN (90) é indefinido). Dependendo da direção desejada, soma-se ou subtrai-se

a 0, 90, 180 ou 270 de forma a se enquadrar na tabela de CURSO da USS ENTERPRISE (Figura 14.9).

Existem outras maneiras de se calcular o CURSO. Nenhuma delas foi automatizada no COMPUTADOR para não tornar a aventura de vencer os KLINGONS muito fácil.

No VISOR (Figura 14.3), SENSOR (Figura 14.4), relatório do COMPUTADOR (Figura 14.5) e relatório de DANOS/ENGENHARIA (Figura 14.6) você recebe informações adicionais, básicas e da maior importância.

- DATA — data estelar corrente/dada limite
- COND. — condição da USS ENTERPRISE: pode ser ATRACADO, VERDE ou VERMELHA; se em VERMELHA, *you're in danger*.
- QUAD. — localização do *quadrante* onde você se encontra.
- SETOR — localização do *setor* onde você se encontra.
- ENERGIA — qual a carga dos seus bancos de energia.
- DEFL. — quanto de energia você tem nas TELAS DEFLETORAS: *se zero, a nave explode*.
- TORPEDO — número de torpedos prontos para lançamento.
- KLINGON — frota inimiga na Galáxia: *se zero, você venceu*.

Para responder ao pedido:

SUAS ORDENS COMANDANTE:

entre uma das respostas da Figura 14.13.

Curso	— para movimentar a nave
Visor	— para ativar o VISOR
Sensor	— para ativar o SENSOR
COMputador	— para ativar o COMPUTADOR
Final	— para suspender o jogo
Torpedo	— para ativar os TORPEDOS
Phaser	— para ativar os PHASERS
Faser	— sinônimo de PHASERS
PHoton	— sinônimo de TORPEDOS
POton	— sinônimo de TORPEDOS
Engenharia	— ativa o controle de avarias
Defletor	— ativa o controle dos DEFLETORES
Aporte	— para atracar em uma BASE

Figura 14.13 RELAÇÃO DOS COMANDOS

As respostas podem ser abreviadas. As letras *maiúsculas* indicam a *abreviatura mínima*, ou seja, VISOR, VISO, VIS, VI ou V têm o mesmo efeito. CO é abreviatura mínima de COMPUTADOR.

Ao iniciar o jogo, Figura 14.1, tecla qualquer coisa que *não* BREAK. A tela se apagará. Voltará com o VISOR do quadrante onde você se encontra. Usando os comandos anteriores você deve viajar para outros quadrantes para localizar e destruir os KLINGONS. É boa norma emitir o SENSOR sempre que você sair de um quadrante, de forma a deixar armazenado no COMPUTADOR dados sobre aquela região.

Sempre que você der um comando, a tela se apagará. O tempo varia dependendo do que você ordenou.

C MANUAL DE OPERAÇÃO

Cerca de 5 minutos são necessários para carregar este programa a partir da fita cassete. Ocupa cerca de 14K de memória.

D MANUAL TÉCNICO

Este programa tem muito código, porém é extremamente simples.

Sua estrutura de dados é basicamente composta de duas matrizes $G(8 \times 8)$ → Galáxia e $Q\$(8,32)$ → quadrante. Observe que a matriz $Q\$\$$ é de 3 caracteres por setor, que é usado diretamente para apresentação.

Um vetor $D(8)$ gerencia os defeitos dos equipamentos e a matriz $K(8,8)$ é a memória do COMPUTADOR de bordo.

A rotina 1000 é concentradora do controle do programa. Para 1000 ou 1100 é direcionado o retorno de todas as rotinas de processamento de comandos.

Observe no programa-fonte (Figura 14.14) que a resposta à pergunta sobre qual o comando a executar é truncada a 5 caracteres, se maior, e comparada com a lista de comandos. Na comparação trunca-se a constante.

Outro detalhe interessante é a indexação de matrizes. No movimento da NAVE, ou do TORPEDO, os indexadores são incrementados de valores fracionários. A indexação feita com números não inteiros é dependente de implementação do interpretador BASIC. Algumas implementações utilizam apenas a parte inteira. SINCLAIR usa a parte inteira se a parte fracionária for menor que .5, se não usa o inteiro imediatamente superior. Para ficar

independente de implementação, usamos um vetor $R(8)$ inicializado com o próprio índice que utilizamos para determinar o índice correto, em substituição à função INT.

O modo **FAST** é usado para que o tempo de resposta do computador se mantenha razoável, para efeitos visuais — viagens subespaciais são não visíveis e **PAUSE** com valor muito pequeno cria tremores na tela.

A listagem do programa se encontra na Figura 14.14.

```
10 REM XXXXXXXX TK82
20 REM ANTONIO JULIO LOSSIO BO
TELHO
25 FAST
30 DIM G(8,8)
35 DIM R(8)
36 DIM C(3)
37 DIM U(8)
40 DIM Q$(8,32)
50 DIM Z(3,3)
60 DIM K(8,8)
70 DIM D(8)
75 LET D$="MOTDR      VISOR
SENSOR      FASER      TUBO TORP.E
NGENHARIADE FLETORES COMPUTADOR"
80 LET T=INT (RND*2000+2000)
90 LET T0=INT (RND*10+35)
95 LET TT=T+T0
100 LET D0=0
110 LET E0=5000
130 LET P0=10
140 LET S0=200
150 LET P=P0
160 LET E=E0-S0
165 LET UI=0
170 LET QX=INT (RND*8+1)
180 LET QY=INT (RND*8+1)
190 LET SX=INT (RND*8+1)
200 LET SY=INT (RND*8+1)
210 LET S=S0
220 LET H=0
230 LET N=0
240 LET K0=0
250 FOR I=1 TO 8
300 LET D(I)=0
305 LET R(I)=I
306 LET C(I)=I*4-3
307 LET U(I)=C(I)+2
310 FOR J=1 TO 8
320 LET K(I,J)=-11
```

Figura 14.14
LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL)

```

330 LET B=AND
340 LET K3=0
350 IF R>.95 THEN LET K3=K3+1
360 IF R>.90 THEN LET K3=K3+1
370 IF R>.80 THEN LET K3=K3+1
380 LET K9=K9+K3
390 LET S3=INT (AND*7+3)
400 LET G(I,J)=S3*100+K3*10
410 NEXT J
420 NEXT I
430 LET BT=INT (AND*6+3)
440 FOR I=1 TO BT
450 LET X=INT (AND*8+1)
460 LET Y=INT (AND*8+1)
470 LET G(X,Y)=G(X,Y)+1
480 NEXT I
490 LET K0=K9
600 LET C#="VERDE"

800 CLS
820 PRINT "SUAS ORDENS CAP. KIR
K."
830 PRINT
840 PRINT "UMA FROTA KLIGON COM
POSTA DE"
850 PRINT K9;" NAVES INVADIU A
GALAXIA"
860 PRINT "E VOCE TEM ";T0;" AN
OS"
870 PRINT "ESTELARES PARA VENCE
-LOS."
880 PRINT
890 PRINT BT;" BASES ESTARAO A
SUAS ORDENS"
900 PRINT AT 21,0;"TECLE 00 TEC
LA EXCETO 00=00"
910 SLOW
920 IF INKEY$="" THEN GOTO 910
930 FAST
940 CLS
990 GOTO 2730
1000 FAST
1010 PRINT AT 17,0;"DATA: ";T;" / "
;TT
1020 PRINT AT 17,16;"COND.:";C#
1030 PRINT AT 18,0;"QUAD. (";OX;
";";OY;")"
1040 PRINT AT 18,16;"SETOR: (";SX
";";SY;")"
1050 PRINT AT 19,0;"ENERGIA: ";E;
"
1060 PRINT AT 19,16;"DEFL.:";S;"

```

```

1070 PRINT AT 20,0;"TORPEDOS: ";P
;"
1080 PRINT AT 20,16;"KINGLONS: ";
K9;"
1100 GOSUB 1500
1101 SLOW
1105 PRINT AT 21,0;"SUAS ORDENS
COMANDANTE:"
1110 INPUT A#
1120 LET A=LEN A#
1130 IF A=0 THEN GOTO 1110
1140 IF A<=5 THEN GOTO 1170
1150 LET A#="
1160 LET A#=A#(1 TO 5)
1170 IF A#="CURSO"(1 TO A) THEN
GOTO 2000
1180 IF A#="VISUR"(1 TO A) THEN
GOTO 3400
1190 IF A#="SENSO"(1 TO A) THEN
GOTO 3000
1200 IF A#="FASER"(1 TO A) THEN
GOTO 6000
1210 IF A#="TORPE"(1 TO A) THEN
GOTO 4000
1220 IF A#="DEFLE"(1 TO A) THEN
GOTO 5000
1230 IF A#="ENGEN"(1 TO A) THEN
GOTO 5200
1240 IF A#="COMPU"(1 TO A) THEN
GOTO 3500
1250 IF A#="FINAL"(1 TO A) THEN
GOTO 7100
1260 IF A#="PHASE"(1 TO A) THEN
GOTO 6000
1270 IF A#="PHOTO"(1 TO A) THEN
GOTO 4000
1280 IF A#="FOTON"(1 TO A) THEN
GOTO 4000
1290 IF A#="APORT"(1 TO A) THEN
GOTO 5500
1300 GOTO 1110

1500 PRINT AT 21,0;"
"
1510 RETURN
2000 GOSUB 1500
2010 PRINT AT 21,0;"CURSO (0-360
)?"
2020 INPUT C1
2030 IF C1<0 THEN GOTO 1000
2040 LET C1=(90+C1)*PI/180
2050 GOSUB 1500

```

Figura 14.14

LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL) (cont.)

Figura 14.14

LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL) (cont.)


```

2060 PRINT AT 21,0;"DOBRA (1-8)
?"
2070 INPUT U
2080 IF U<=0 THEN GOTO 1000
2090 IF U>8 THEN GOTO 2000
2100 IF D(1)>=0 OR U<=.2 THEN GO
TO 2140
2110 PRINT AT 21,0;"MOTOR AVARIA
DO, MAX=.2"
2120 PAUSE 200
2130 GOTO 2000
2140 IF E>0 THEN GOTO 2200
2150 IF S<1 THEN GOTO 7000
2160 GOSUB 1500
2170 PRINT AT 21,0;"ENERGIA INSU
FICIENTE."
2180 PAUSE 200
2190 GOTO 1000
2200 FOR I=1 TO 8
2210 LET D(I)=D(I)+(D(I)<0)
2220 NEXT I
2230 IF RND>.2 THEN GOTO 2350
2240 LET R1=INT (RND*8+1)
2250 IF RND>.5 THEN GOTO 2290
2260 GOSUB 7700
2270 LET D(R1)=D(R1)-INT (RND*5+
1)
2275 GOSUB 1500
2280 PRINT AT 21,0;"TEMPESTADE O
SMICA."
2285 PAUSE 200
2286 GOTO 2350
2290 IF D(R1)>=0 THEN GOTO 2350
2300 GOSUB 1500
2310 PRINT AT 21,0;"REPAROS ACEL
ERADOS"
2320 PAUSE 200
2330 LET D(R1)=D(R1)+INT (RND*5+
1)
2340 IF D(R1)>0 THEN LET D(R1)=0
2350 FAST
2360 CLS
2365 LET UI=0
2370 LET Q$(SX,C(SY) TO U(SY))="
"

2380 LET X1=COS C1
2390 LET Y1=8IN C1
2400 LET X=SX
2410 LET Y=SY
2420 LET N=INT (U*8)
2430 FOR I=1 TO N
2440 LET SX=SX+X1

```

```

2450 LET SY=SY+Y1
2460 IF SX<1 OR SY<1 OR SX>8 OR
SY>8 THEN GOTO 2610
2470 IF Q$(SX,C(SY) TO U(SY))<>"
" THEN GOTO 2540
2480 NEXT I
2490 LET Q$(SX,C(SY) TO U(SY))="
<*>"
2492 LET SX=R(SX)
2494 LET SY=R(SY)

2500 LET E=E-N*5-INT (S/10)
2505 GOSUB 6500
2510 IF U>=.1 THEN LET T=T+1
2520 IF T>TT THEN GOTO 7600
2530 GOTO 3400
2540 GOSUB 7700
2550 GOSUB 1500
2560 PRINT AT 21,0;"PASSAGEM BLO
QUEADA."
2570 PAUSE 200
2580 LET SX=SX-X1
2590 LET SY=SY-Y1
2600 GOTO 2490
2610 LET X=QX*8+X+X1*N
2620 LET Y=QY*8+Y+Y1*N
2630 LET QX=INT (X/8)
2640 LET QY=INT (Y/8)
2650 LET SX=INT (X-QX*8)
2660 LET SY=INT (Y-QY*8)
2670 IF SX>0 THEN GOTO 2700
2680 LET SX=8
2690 LET QX=QX-SGN X1
2700 IF SY>0 THEN GOTO 2721
2710 LET SY=8
2720 LET QY=QY-SGN Y1
2721 IF QX>8 THEN LET QX=8
2722 IF QY>8 THEN LET QY=8
2723 IF QX<1 THEN LET QX=1
2724 IF QY<1 THEN LET QY=1
2730 LET U=1
2740 FOR I=1 TO 8
2760 LET Q$(I)=" "
2780 NEXT I
2785 LET Q$(SX,C(SY) TO U(SY))="
<*>"
2790 LET S3=INT (G(QX,QY)/100)
2800 LET K3=INT ((G(QX,QY)-S3*10
0)/10)
2810 LET B3=G(QX,QY)-S3*100-K3*1
0
2820 LET A$=">:<"

```

Figura 14.14 LISTAGEM DO PROGRAMA - FONTE (AVENTURA ESPACIAL) (cont.)

Figura 14.14 LISTAGEM DO PROGRAMA - FONTE (AVENTURA ESPACIAL) (cont.)

```

00030 LET I=03
00040 GOSUB 0020
00050 LET A#="*"
00060 LET I=03
00070 GOSUB 0020
00080 LET A#="**"
00090 LET I=03
00100 GOSUB 0020
00110 IF I=3 THEN GOTO 02490
00120 FOR J=I+1 TO 3
00130 LET Z(J,3)=-1
00140 NEXT J
00150 GOTO 02490
00160 IF I<1 THEN RETURN
00170 FOR J=1 TO I
00180 LET X=INT (RAND*8+1)
00190 LET Y=INT (RAND*8+1)
00200 IF Q$(X,C(Y) TO U(Y))<>
" THEN GOTO 02940
00210 LET Q$(X,C(Y) TO U(Y))=A#
00220 IF A#<>" " THEN GOTO 02990
00230 LET Z(J,1)=X
00240 LET Z(J,2)=Y
00250 LET Z(J,3)=89
00260 NEXT J
00270 RETURN

3000 IF D(3)>=0 THEN GOTO 3050
3010 GOSUB 1500
3020 PRINT AT 21,0;"SENSOR REMOT
O AVARIADO."
3030 PAUSE 200
3040 GOTO 1100
3050 FAST
3060 CLS
3065 LET VI=0
3070 PRINT AT 0,9;"SENSOR REMOTO
"

3080 FOR I=0X-1 TO 0X+1
3090 FOR J=0Y-1 TO 0Y+1
3100 IF I<1 OR J<1 OR I>8 OR J>8
THEN GOTO 3130
3110 LET K(I,J)=G(I,J)
3120 PRINT AT ((I-0X)*3+8),((J-0
Y)*4+14);K(I,J)
3130 NEXT J
3140 NEXT I
3150 GOTO 1000
3300 FAST
3310 CLS
3312 LET VI=1
3315 PRINT AT 0,14;"XXXXXXXXXX"

```

Figura 14.14

LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL) (cont.)

```

3320 FOR I=1 TO 8
3340 PRINT AT I#2-1,0;Q$(I);
3360 NEXT I
3370 RETURN
3400 IF D(2)>=0 THEN GOTO 3450
3410 GOSUB 1500
3420 SLOW
3430 PRINT AT 21,0;"VISOR DEFEIT
050"
3440 PAUSE 200
3450 GOTO 1100
3460 GOSUB 3300
3470 GOTO 1000
3500 IF D(8)>=0 THEN GOTO 3550
3510 GOSUB 1500
3520 SLOW
3530 PRINT AT 21,0;"COMPUTADOR D
EFEITOSO"
3540 PAUSE 200
3550 GOTO 1100
3560 FAST
3570 CLS
3575 LET VI=0
3580 PRINT AT 0,14;"XXXXXXXXXX"
3590 FOR I=1 TO 8
3595 PRINT AT I#2-1,0;
3600 FOR J=1 TO 8
3610 REM PRINT AT I#2-1,(J-1)*4;
" ";K(I,J);
3615 PRINT " ";K(I,J);
3620 NEXT J
3630 NEXT I
3640 GOTO 1000

4000 SLOW
4010 GOSUB 1500
4020 IF D(5)>=0 THEN GOTO 4060
4030 PRINT AT 21,0;"TUBO DE LANÇ
AMENTO AVARIADO."
4040 PAUSE 200
4050 GOTO 1100
4060 IF P>0 THEN GOTO 4090
4070 PRINT AT 21,0;"VOCE NAO TEM
HAIS TORPEDOS."
4080 GOTO 4040
4090 PRINT AT 21,0;"CURSO DO TOR
PEDO (0-360) ?"
4100 INPUT C1
4110 IF C1<0 THEN GOTO 1000
4120 LET C1=(C1+90)*PI/180
4130 LET X1=COS C1
4140 LET Y1=SIN C1

```

Figura 14.14

LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL) (cont.)

```

4150 LET P=P-1
4160 LET X=3X
4170 LET Y=3Y
4180 IF D(2)>=0 AND VI=1 THEN GO
TO 4500
4181 CLS
4182 LET VI=0
4190 IF D(2)<0 THEN PRINT "VISOR
AVARIADO."
4200 IF D(8)>=0 THEN GOTO 4280
4210 GOSUB 1500
4220 PRINT "COMPUTADOR AVARIADO"
4230 PAUSE 100
4230 LET RE=4240
4240 GOTO 4800
4250 GOTO 4390
4260 GOTO 4350
4280 PRINT "RASTREAMENTO PELO CO
MPUTADOR"
4290 LET RE=4320
4300 PRINT "NAVE EM (";X";",";Y);"
)"
4310 GOTO 4800
4320 PRINT "          (";R(X);",";
R(Y);")"
4325 GOTO 4800
4330 PRINT AT 21,0;"TORPEDO PERD
IDO"
4332 GOTO 4360
4340 GOSUB 7700
4350 IF 0$(X,C(Y) TO U(Y))<>>:X
" THEN GOTO 4400
4360 PRINT AT 21,0;"VOCE DESTRUI
U UMA BASE."
4370 LET BT=BT-1
4375 LET G(OX,OY)=G(OX,OY)-1
4378 LET 0$(X,C(Y) TO U(Y))="
"
4380 PAUSE 100
4390 GOSUB 6500
4395 GOTO 1000
4400 IF 0$(X,C(Y) TO U(Y))=" * "
THEN GOTO 4600
4410 PRINT AT 21,0;"KLINGON DEST
RUIDO"
4411 FOR I=1 TO 3
4412 IF (Z(I,1)=R(X)) AND (Z(I,2
)=Y) THEN LET Z(I,3)=-1
4415 NEXT I
4420 LET K3=K3-1
4425 LET K9=K9-1
4426 IF K9<1 THEN GOTO 7400

```

Figura 14.14 LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL) (cont.)

```

4430 LET G(OX,OY)=G(OX,OY)-10
4440 GOTO 4378
4500 GOSUB 1500
4506 SLOW
4510 LET RE=4530
4520 GOTO 4800
4530 GOTO 4560
4540 GOTO 4330
4550 IF 0$(X,C(Y) TO U(Y))=" * "
THEN GOTO 4340
4552 PRINT AT 2*R(X)-1,(R(Y)-1)*
4;" ";
4554 GOTO 4340
4560 PRINT AT X*2-1,Y*4-3;"███";
4570 PRINT AT X*2-1,Y*4-3;"  "
4580 GOTO 4800
4600 PRINT AT 21,0;"VOCE ACERTOU
UMA ESTRELA"
4610 GOTO 4360
4600 LET X=X+X1
4810 LET Y=Y+Y1
4820 IF X<1 OR X>8 OR Y<1 OR Y>8
THEN GOTO (RE+10)
4830 IF 0$(X,C(Y) TO U(Y))="  "
THEN GOTO RE
4840 GOTO (RE+20)
5000 SLOW
5010 GOSUB 1500
5020 IF D(7)>=0 THEN GOTO 5060
5030 PRINT AT 21,0;"CONTROLE DEF
EITUOSO."
5040 PAUSE 200
5050 GOTO 1100
5060 PRINT AT 21,0;"QUANTA ENERG
IA NOS DEFLETORES?"
5070 INPUT X
5080 IF X<=0 THEN GOTO 1100
5090 IF (S+E)>X THEN GOTO 5120
5100 PRINT AT 21,0;"VOCE NAO TEM
TANTA ENERGIA."
5110 GOTO 5040
5120 LET E=E+S-X
5130 LET S=X
5140 GOTO 1000
5200 SLOW
5210 GOSUB 1500
5220 IF D(6)>=0 THEN GOTO 5250
5230 PRINT AT 21,0;"ENGENHARIA 5
EM COMUNICACAO."
5240 GOTO 5040
5250 LET VI=0

```

Figura 14.14 LISTAGEM DO PROGRAMA-FONTE (AVENTURA ESPACIAL) (cont.)

```

5260 CLS
5270 PRINT "RELATORIO DE DANOS"
5280 PRINT
5290 PRINT "EQUIPAMENTO";TAB 16;
"DANOS"
5295 PRINT
5300 FOR I=1 TO 8
5310 PRINT D$(I*10-9 TO I*10);TA
B 16;D(I)
5320 NEXT I
5330 GOTO 1000
5340 SLOW
5350 FOR J=5Y-1 TO 5Y+1
5360 FOR I=5X-1 TO 5X+1
5370 IF I<1 OR J<1 OR I>8 OR J>8
THEN GOTO 5550
5380 IF Q$(I,C(J) TO U(J))=">:<"
THEN GOTO 5600
5390 NEXT I
5400 NEXT J
5410 GOSUB 1500
5420 PRINT AT 21,0;"APORTAR??? E
A BASE???"
5430 PAUSE 200
5440 GOTO 1100
5450 LET C$="ATACADO"
5460 LET E=50-59
5470 LET P=P0
5480 LET S=59
5490 FOR I=1 TO 8
5500 LET D(I)=0
5510 NEXT I
5520 GOTO 1000

6000 GOSUB 1500
6010 IF D(4)>=0 THEN GOTO 6040
6020 PRINT AT 21,0;"PHASER AVARI
ADO."
6030 GOTO 5590
6040 IF K3>0 THEN GOTO 6070
6050 PRINT AT 21,0;"NAO TEM KLIN
GON NO QUADRANTE."
6060 GOTO 5590
6070 IF D(7)>=0 THEN GOTO 6100
6080 PRINT AT 21,0;"PONTARIA FAL
HA - COMP. AVARIADO";
6090 PAUSE 100
6095 GOSUB 1500
6100 PRINT AT 21,0;"PHASER APONT
ADO. ENERGIA?"
6110 INPUT X
6120 IF X<=0 THEN GOTO 1100

```

Figura 14.14 LISTAGEM DO PROGRAMA - FONTE (AVENTURA ESPACIAL) (cont.)

```

6130 IF X<=E THEN GOTO 6160
6140 PRINT AT 21,0;"VOCE NAO TEM
TANTA ENERGIA"
6150 GOTO 6090
6160 LET E=E-X
6165 CLS
6166 PRINT "RELATORIO DE PHASERS"
"
6167 PRINT "KLINGON CARGA RESTAN
TE CONDICAO "
6170 IF D(7)<0 THEN LET X=X*RND
6180 FOR I=1 TO 3
6181 IF Z(I,3)<0 THEN GOTO 6290
6182 LET X1=Z(I,1)
6183 LET Y1=Z(I,2)
6190 GOSUB 6900
6195 LET H=X/K3/(SQR H)*2*RND
6196 LET H=INT H
6200 LET Z(I,3)=Z(I,3)-H
6210 PRINT "(";X1;";";Y1;")";TAB
8;H;TAB 14;Z(I,3);
6220 IF Z(I,3)>=0 THEN GOTO 6260
6230 PRINT TAB 23;"6230 PRINT";
6250 LET Q$(X1,C(Y1) TO U(Y1))="
"
6260 LET K3=K3-1
6265 LET G(OX,OY)=G(OX,OY)-10
6270 LET K9=K9-1
6271 IF K9<1 THEN GOTO 7400
6275 IF K3<1 THEN GOTO 6300
6280 PRINT ""
6290 NEXT I
6300 IF VI=0 THEN GOTO 6320
6310 PAUSE 100
6320 GOSUB 6500
6330 GOTO 3400

6500 LET C$="VERDE"
6510 IF K3<1 THEN RETURN
6515 LET C$="VERMELHA"
6518 LET X=0
6520 FOR I=1 TO 3
6530 IF Z(I,3)<=0 THEN GOTO 6590
6540 LET X1=Z(I,1)
6550 LET Y1=Z(I,2)
6560 GOSUB 6900
6570 LET H=Z(I,3)/(SQR H)*2*RND
6580 LET X=X+INT H
6590 NEXT I
6600 SLOW
6620 GOSUB 7700
6630 PRINT AT 21,0;"ATAQUE KLING

```

Figura 14.14 LISTAGEM DO PROGRAMA - FONTE (AVENTURA ESPACIAL) (cont.)

```

ON ";X)" UNID."
6640 LET S=S-X
6650 PAUSE 100
6660 IF S>59 THEN RETURN
6670 GOSUB 7700
6680 LET I=INT (RND*8+1)
6690 LET D(I)=D(I)-1
6700 IF S>0 THEN RETURN
6710 GOSUB 7700
6720 CLS
6730 PRINT "DEFLETORES DESTRUIDO
S..."
6740 PRINT
6750 PRINT "SUA NAVE VAI EXPLOD#
#=#"
6770 GOTO 7800
6900 LET H=((X1-SX)*(X1-SX))+((Y
1-SY)*(Y1-SY))
6910 RETURN
7000 CLS
7010 PRINT "VOCE ESTA SEM ENERGI
A PARA MOVER-SE NO ESPACO."
7020 GOTO 7800
7100 CLS
7110 PRINT "LASTIMAVEL QUE VOCE
NAO TENHA FIBRA SUFICIENTE PAR
A DEFENDER AFEDERACAO DA INVAZAO
KINGLON."
7120 PRINT
7130 GOTO 7800
7400 PAUSE 100
7405 CLS
7410 PRINT "PARABENS. VOCE FOI
PROMOVIDO ALMIRANTE DA FROTA E
STELAR POR TER VENCIDO AOS KLIN
GOS."
7420 PRINT "SUA MEDIA FOI DE ";K
0/T0
7430 PRINT "KLINGONS POR ANO EST
ELAR."
7440 GOSUB 7700
7450 PRINT AT 19,0;
7460 GOTO 7910

7600 CLS
7610 PRINT "TERMINOU SEU TEMPO.
UMA SEGUNDA LEVA DE NAVES KLIGON
S PENETROU NA GALAXIA COMO TEMI
DO E VOCE FOI INCAPAZ DE VENCE
-LOS."
7620 PRINT
7630 GOTO 7800

```

```

7700 FAST
7710 FOR D=1 TO 10
7720 PAUSE 2
7730 NEXT D
7740 GOSUB 1500
7750 PRINT AT 21,0;"ATENCAO A
TENCAO ATENCAO"
7760 SLOW
7770 PAUSE 100
7780 GOSUB 1500
7790 RETURN
7800 PRINT
7810 PRINT "SUA INCOMPETENCIA LE
VOU AO DE- SASTRE UNIVERSAL. A
GALAXIA FOI DOMINADA PELOS KLING
ONS E VOCE EUM XXXXXXXXXX"
7820 STOP
7900 CLS
7910 PRINT "PARA OUTRA AVENTURA
TECLE ESC"
7920 STOP
7930 GOTO 7900
9000 FAST
9010 CLEAR
9020 SAVE "*TREK"
9030 RUN

```

Figura 14.14 LISTAGEM DO PROGRAMA - FONTE (AVENTURA ESPACIAL) (cont.)

Figura 14.14 LISTAGEM DO PROGRAMA - FONTE (AVENTURA ESPACIAL) (cont.)

CAPÍTULO 15

***Uma Aventura
Administrativa***

A FICHA

a) Tipo: Programa	<input checked="" type="checkbox"/>	b) Nome: AVENTURA ADMINISTRATIVA	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83	
Sub-rotina	<input type="checkbox"/>	f) Objetivo: Jogo				
Função	<input type="checkbox"/>					
g) Linguagem:						
BASIC		<input checked="" type="checkbox"/>	PASCAL	<input type="checkbox"/>	FORTH	<input type="checkbox"/>
OUTRAS (quais?)		<input type="checkbox"/>	PL/			<input type="checkbox"/>
FORTRAN		<input type="checkbox"/>	COBOL	<input type="checkbox"/>	C	<input type="checkbox"/>
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:		
NE Z8Ø	<input type="checkbox"/>	VÍDEO	<input checked="" type="checkbox"/>			
NE Z8ØØØ	<input checked="" type="checkbox"/>	TECLADO	<input checked="" type="checkbox"/>			
TK 8Ø	<input type="checkbox"/>	MEMÓRIA: 2K	<input type="checkbox"/>	SLOW	<input checked="" type="checkbox"/>	
TK 82-C/83	<input checked="" type="checkbox"/>	16K	<input checked="" type="checkbox"/>	FAST	<input type="checkbox"/>	
TK 85	<input checked="" type="checkbox"/>	48K	<input type="checkbox"/>	LPRINT	<input type="checkbox"/>	
SINCLAIR ZX 8Ø	<input type="checkbox"/>	64K	<input type="checkbox"/>	LLIST	<input type="checkbox"/>	
SINCLAIR ZX 81	<input checked="" type="checkbox"/>	CASSETTE	<input checked="" type="checkbox"/>	COPY	<input type="checkbox"/>	
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	DISQUETE	<input type="checkbox"/>	DLOAD	<input type="checkbox"/>	
TIMEX 1ØØØ	<input checked="" type="checkbox"/>	DISCO: 8"	<input type="checkbox"/>	DSAVE	<input type="checkbox"/>	
CP-2ØØ	<input checked="" type="checkbox"/>	5¼"	<input type="checkbox"/>	DMLOAD	<input type="checkbox"/>	
CP-3ØØ	<input type="checkbox"/>	ALAVANCA ("JOYSTICK")	<input type="checkbox"/>	DMSAVE	<input type="checkbox"/>	
APPLY 3ØØ	<input type="checkbox"/>	IMPRESSORA	<input type="checkbox"/>	OUTROS (quais?)	<input type="checkbox"/>	
OUTROS (quais?)	<input checked="" type="checkbox"/>	OUTROS (quais?)	<input type="checkbox"/>			
RINGO R-47Ø e AS-1ØØØ						
k) Observações:						

B MANUAL DE UTILIZAÇÃO

Você foi eleito administrador de uma ilha de 4000 km², 2000 dos quais ocupados por uma linda floresta. O restante é cultivado. A floresta é grande atração turística e traz muitos visitantes à ilha. Porém, devido à falta de indústria, o volume de importações é muito grande. Ao assumir o governo da ilha você se comprometeu a distribuir aos ilhéus parte dos lucros obtidos, o que os deixou muito felizes. Você garantiu distribuir um mínimo de \$200,00 a cada habitante. O custo de vida na ilha é tal que uma pessoa para sobreviver necessita de no mínimo \$100,00.

O número de habitantes da ilha é escolhido ao acaso, próximo de 1500. A disponibilidade de dinheiro em caixa é sorteado próximo a \$500.000,00.

No início lhe é informado quantos habitantes existem e quanto você tem em caixa. Anote, pois você não tem como rever informações anteriores.

Industriais estrangeiros desejam comprar terras para instalarem-se na ilha. Você pode vender ou não. Se vender, lembre-se de que indústrias poluem a ilha e reduzem sua produção agrícola. Porém, você pode desmatar para substituir as terras vendidas. Isto diminuirá os atrativos turísticos da ilha. Para plantar 1 km² você precisa de pelo menos 2 pessoas.

Ao final do conjunto de perguntas e respostas, é feito um balanço da situação da ilha. Depois você é perguntado se quer parar, continuar ou salvar em fita para continuar posteriormente – neste caso use uma fita rascunho.

C MANUAL DE OPERAÇÃO

Este programa leva cerca de 3 minutos para ser carregado de fita cassete e ocupa aproximadamente 6K. Veja nos Apêndices.

D MANUAL TÉCNICO

Veja na Figura 15.1 o programa-fonte. Trata-se de uma seqüência de perguntas e leituras com alternativas sobre a próxima pergunta e crítica das respostas.

```

10 REM ANTONIO JULIO LOSSIO BC
TELHC
20 REM D=DINHEIRO
30 REM P=PESSOAS
40 REM T=TERRAS
50 REM F=FLORESTA
60 REM E=EXTRANGEIROS
70 REM U=PRECO TERRAS
80 REM M=INDUSTRIAS
85 REM H=PLANTACAO
90 LET M=0
95 LET F=2000
100 SLOW
110 LET D=500000+10000*RND-1000
0*RND
120 LET P=INT (1500+100*RND-100
*RND)
130 LET T=2000
140 LET E=0
150 LET U=RND*100+120
160 PRINT AT 20,0;"VOCE TEM US#
";D;" EM CAIXA"
170 SCROLL
180 PRINT AT 20,0;"SAO ";P;" SO
CIOS COPROPRIETARIOS"
190 SCROLL
200 IF E<1 THEN GOTO 230
210 PRINT AT 20,0;"E ";E;" TRAB
ALHADORES EMPREGADOS"
220 SCROLL
230 PRINT AT 20,0;"INDUSTRIAS
DESEJAM INVESTIR NA ILHA"
235 SCROLL
240 SCROLL
250 PRINT AT 20,0;"PARA ISTO PR
ECISAM COMPRAR TER-RAS"
255 SCROLL
265 SCROLL
270 PRINT AT 20,0;"ESTAO OFEREC
ENDO US#";U;" POR KM2"
280 SCROLL
285 SCROLL
290 PRINT AT 20,0;"QUANTO VOCE
VENDE A ELES ?";
300 INPUT U
310 PRINT U

320 SCROLL
330 IF U<0 THEN GOTO 290
340 IF U<=T THEN GOTO 370
350 PRINT AT 20,0;"VOCE SO TEM
";T;" KM2 DE TERRAS "

```

Figura 15.1 LISTAGEM DO PROGRAMA-FONTE
(AVENTURA ADMINISTRATIVA)

```

360 GOTO 280
370 LET T=T-U
380 LET D=D+U*U
390 LET M=M+U
400 IF F<1 OR D<U/10 THEN GOTO
600
410 PRINT AT 20,0;"VOCE QUER DE
SMATAR A FLORESTA ?"
420 SCROLL
430 PRINT AT 20,0;"LEMBRE-SE QU
E A FLORESTA E ATRA-CAO TURISTIC
A"
435 SCROLL
440 SCROLL
450 PRINT AT 20,0;"QUANTO VAI S
ER DESMATADO ?";
460 INPUT U
470 PRINT U
480 IF U<0 THEN GOTO 440
490 IF U<=F THEN GOTO 520
495 SCROLL
500 PRINT AT 20,0;"VOCE SO TEM
";F;" KM2 DE FLORESTA"
510 GOTO 440
520 SCROLL
525 IF U*U/10<=D THEN GOTO 570
530 PRINT AT 20,0;"DESMATAR CUS
TA US#";U/10;" POR KM2 "
540 SCROLL
545 SCROLL
550 PRINT AT 20,0;" E VOCE NAO
TEM DINHEIRO PARA ISSO ";
555 SCROLL
560 GOTO 440
570 LET D=D-U*U/10
580 LET T=T+U
590 LET F=F-U
595 LET H=0
600 IF T<1 OR D<U/100 THEN GOTO
820
610 SCROLL
620 PRINT AT 20,0;"QUANTO VOCE
QUER PLANTAR ?";
630 INPUT H
640 PRINT H
645 SCROLL
650 IF H<1 THEN GOTO 820
660 IF H*U/100<=D THEN GOTO 710
670 PRINT AT 20,0;"O CUSTO DA S
EMENTE E US#";U/100;" POR KM2"
680 SCROLL
685 SCROLL

```

Figura 15.1 LISTAGEM DO PROGRAMA-FONTE
(AVENTURA ADMINISTRATIVA) (cont.)


```

690 PRINT AT 20,0;"E VOCE NAO T
EM DINHEIRO PARA ISSO"
700 GOTO 610
710 IF P/H>=2 THEN GOTO 610
720 SCROLL
730 PRINT AT 20,0;"VOCE PRECISA
NO MINIMO DE 2"
740 SCROLL

750 PRINT AT 20,0;"PESSOAS PARA
O CULTIVO DE 1 KM2"
760 GOTO 610
810 LET D=D-H*U/100
820 IF H<1 OR RND>.2 THEN GOTO
880
830 LET U=RND*H
840 SCROLL
850 PRINT AT 20,0;"TEMPORAL DES
TRUIU ";U;" KM2 DAS PLANTACOES"
860 SCROLL
865 SCROLL
870 LET H=H-U
880 LET U=(4001-F)*RND/5+H
890 IF U<F OR H<1 THEN GOTO 120
0
900 PRINT AT 20,0;"O DESMATAMEN
TO E AS INDUSTRIAS ESTAO POLUIN
DO A ILHA "
910 SCROLL
915 SCROLL
920 PRINT AT 20,0;"QUANTO VOCE
GASTAR PARA CONTRO- LAR A POLUIC
AO ?";
930 INPUT U
940 PRINT U
950 SCROLL
955 SCROLL
960 IF U<=0 THEN GOTO 990
970 PRINT AT 20,0;"VOCE SO TEM
US#";D
980 GOTO 910
990 LET D=D-U
1000 LET U=U*U-U
1010 IF U<0 THEN GOTO 1040
1020 PRINT AT 20,0;"DEVIDO AOS E
STRAGOS DA AGUA E AR "
1030 LET V=INT (RND*P/50)
1040 SCROLL
1050 LET Z=INT (U*RND)
1060 PRINT AT 20,0;"MORRERAM ";U
-Z;" PESSOAS E ";
1065 SCROLL

```

```

1070 PRINT AT 20,0;Z;" ADOECERAM
, PORTANTO"
1075 SCROLL
1080 LET U=U*RND+U
1090 IF U>D THEN LET U=D
1100 PRINT AT 20,0;"SUAS DESPESA
S COM FUNERAIS E"
1110 SCROLL
1120 PRINT AT 20,0;"HOSPITAL FOR
AM US#";U
1130 LET D=D-U
1140 LET P=P-U+Z
1200 LET E=INT ((F/2+P*RND)/2)
1205 IF E<1 THEN GOTO 1300
1210 SCROLL
1220 PRINT AT 20,0;E;" TURISTAS
VISITARAM A ILHA,"
1230 LET U=E*(100*RND+100)
1240 LET D=D+U
1250 SCROLL
1260 PRINT AT 20,0;"E RENDERAM U
S#";U
1300 LET U=INT (P*(RND*.2+.1))

1305 IF U<1 THEN LET U=1
1310 SCROLL
1320 PRINT AT 20,0;"A POPULACAO
AUMENTOU NATURALMEN-TE. ";U;" PE
SSOAS NASCERAM."
1330 SCROLL
1340 SCROLL
1350 LET P=P+U
1360 GOTO 1700
1400 SCROLL
1410 PRINT AT 20,0;"VAMOS AGORA
A ASSEMBLEIA GERAL."
1415 SCROLL
1420 PRINT AT 20,0;"FEITO SEUS C
ALCULOS, QUAL O VALOR A SER
DISTRIBUIDO AOS"
1422 SCROLL
1425 SCROLL
1430 PRINT AT 20,0;"CO-PROPRIETA
RIOS?"
1435 SCROLL
1440 PRINT AT 20,0;"DE QUANTO E
A COTA?";
1450 INPUT U
1460 PRINT U
1465 SCROLL
1470 IF U<0 THEN GOTO 1440
1480 IF U<=D THEN GOTO 1500

```

Figura 15.1 LISTAGEM DO PROGRAMA-FONTE
(AVENTURA ADMINISTRATIVA) (cont.)

Figura 15.1 LISTAGEM DO PROGRAMA-FONTE
(AVENTURA ADMINISTRATIVA) (cont.)

```

1490 PRINT AT 20,0;"VOCE SO TEM
US#";D
1495 GOTO 1435
1500 IF U/P>100 THEN GOTO 1600
1510 SCROLL
1520 PRINT AT 20,0;"OS RENDIMENT
OS MINIMOS ACERTADOS SAO DE US#10
0 E VOCE NAO PODE"
1525 SCROLL
1530 SCROLL
1540 PRINT AT 20,0;"PAGAR 30 US#
";U/P
1550 SCROLL
1560 PRINT AT 20,0;"OU MORREM TO
DOS."
1570 IF D/P<100 THEN GOTO 5000
1580 GOTO 1435
1600 IF U/P>200 THEN GOTO 1700
1610 SCROLL
1620 PRINT AT 20,0;"A POPULACAO
NAO ESTA SATISFEITA COM O SEU DE
SEMPELHO."
1630 SCROLL
1640 SCROLL
1650 GOTO 1780
1700 SCROLL
1710 PRINT AT 20,0;"UM ANO SE PA
SSOU E A"
1720 SCROLL
1730 PRINT AT 20,0;"COLHEITA REN
DEU US#";H#U/3
1740 SCROLL
1750 LET D=D+H#U/3
1755 GOTO 1910
1780 PRINT AT 20,0;"VAMOS AO PRO
XIMO ANO?"

1770 SCROLL
1780 PRINT AT 20,0;"ENTRE 0 PARA
CONTINUAR AGORA. ENTRE 1 PARA
CONTINUAR DEPOIS"
1790 SCROLL
1800 SCROLL
1820 PRINT AT 20,0;"ENTRE 2 PARA
DESESTIR."
1830 INPUT U
1840 PRINT U
1850 SCROLL
1860 IF U<0 OR U>2 THEN GOTO 178
0
1890 LET U=INT U*100+2000
1900 GOTO U

```

```

1910 PRINT AT 20,0;"TAXACAO SOB R
E A INDUSTRIA RENDEU US#";
M#U/5
1920 SCROLL
1930 SCROLL
1940 LET D=D+M#U/5
1950 GOTO 1400
2000 GOTO 150
2100 SCROLL
2110 PRINT AT 20,0;"POSICIONE A
FITA NO GRAVADOR. TECLE 0 QUAN
DO ESTIVER OK."
2120 INPUT M
2130 SCROLL
2140 SCROLL
2200 SCROLL
2210 PRINT AT 20,0;"QUER PARAR ?
(0=SIM)"
2230 INPUT M
2240 IF M=0 THEN GOTO 5160
2250 SCROLL
2260 GOTO 150
5000 SCROLL
5010 SCROLL
5020 SCROLL
5030 PRINT AT 20,0;"COMO CAO GERA
L"
5040 SCROLL
5050 PRINT AT 20,0;"COM UM SALDO
DE US#";D
5060 SCROLL
5070 PRINT AT 20,0;"VOCE SO PODE
PAGAR US#";D/P
5080 SCROLL
5090 PRINT AT 20,0;"PARA CADA UM
DOS ";P
5100 SCROLL
5110 PRINT AT 20,0;"HABITANTES I
DIOTAS DESTA ILHA."
5120 SCROLL
5130 PRINT AT 20,0;"VOCE FOI DES
TITUIDO E PRESO"
5140 SCROLL
5150 PRINT AT 20,0;"POR MALVERSA
CAO DOS BENS COMUNS."
5160 FAST
5170 STOP
9000 CLEAR
9010 FAST
9020 SAVE "ICOOF"
9030 RUN

```

Figura 15.1 LISTAGEM DO PROGRAMA-FONTE
(AVENTURA ADMINISTRATIVA) (cont.)

Figura 15.1. LISTAGEM DO PROGRAMA-FONTE
(AVENTURA ADMINISTRATIVA) (cont.)

CAPÍTULO 16

***Apasionante e
Tradicional
Batalha Naval***

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:
Programa <input checked="" type="checkbox"/>	BATALHA NAVAL	1	φ	22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: <u>Jogo</u>			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>				
PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8φ <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	TK 8φ <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>
NE Z8φφφ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>
TK 8φ <input type="checkbox"/>		TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>
TK 82-C/83 <input checked="" type="checkbox"/>		SINCLAIR ZX 8φ <input type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>
TK 85 <input checked="" type="checkbox"/>		SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CÁSSETE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>
SINCLAIR ZX 8φ <input type="checkbox"/>		SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>
SINCLAIR ZX 81 <input checked="" type="checkbox"/>		TIMEX 1φφφ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>		CP-2φφ <input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>
TIMEX 1φφφ <input checked="" type="checkbox"/>		CP-3φφ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>
CP-2φφ <input checked="" type="checkbox"/>		APPLY 3φφ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>
CP-3φφ <input type="checkbox"/>		OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>	
APPLY 3φφ <input type="checkbox"/>		<u>RINGO R-47φ</u>		
OUTROS (quais?) <input checked="" type="checkbox"/>		<u>e AS-1φφφ</u>		
k) Observações:				

BATALHA NAVAL é um jogo muito antigo. Em qualquer papelaria encontram-se blocos como o da Figura 16.1.

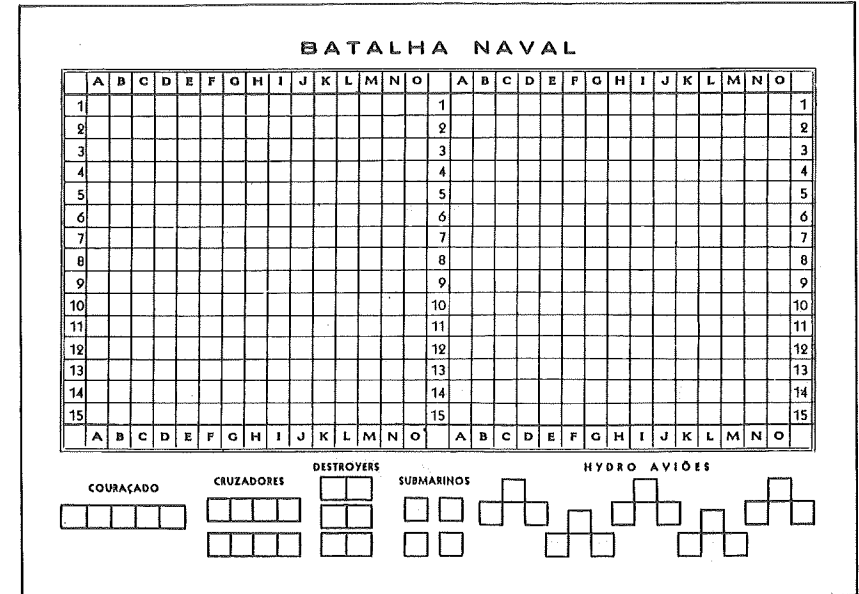


Figura 16.1 BATALHA NAVAL TRADICIONAL

Em cada folha do bloco temos dois quadros reticulados em 15 x 15 com as linhas identificadas por letras e as colunas por números. Estes quadros serão os mapas das suas esquadras. No da esquerda você distribuí a sua esquadra, composta de tantos navios quantos os descritos no rodapé, ou seja, um couraçado (composto de 5 quadrinhos consecutivos na horizontal ou vertical), dois cruzadores (4 quadrinhos) ou fragatas (depende da nomenclatura usada por quem imprimiu o bloco), três destróieres (2 quadrinhos), 4 submarinos e 5 aviões (três quadrinhos em forma de V).

As peças devem ser distribuídas de forma a que os quadrinhos de uma peça não toquem os quadrinhos de outra, isto é, pelo menos um quadrinho vazio separa quadrinhos de peças diferentes.

O outro quadro é reservado para anotações sobre a esquadra do adversário que, em outra folha idêntica, distribui sua esquadra.

Alternadamente os jogadores *atiram* um no outro, dizendo a letra da linha e o número da coluna do quadrinho onde deseja *atirar*. Cada jogador dá 3 (três) *tiros* de cada vez e o adversário responde conforme a regra previamente acertada.

Como a BATALHA NAVAL é um passatempo muito antigo, suas regras sofreram variações ao longo do tempo. Pelo menos quatro variações são bem difundidas e são apresentadas a seguir conforme o grau de dificuldade:

- 1) Responde-se quais e quantos tiros acertaram uma peça e que tipo de peça ficou avariada;
- 2) Responde-se que tipo de peça foi avariada, sem contudo informar qual dos tiros a avariou. Porém informa-se quantos tiros aquela peça já levou;
- 3) Responde-se qual tiro acertou alguma peça, sem informar que tipo de peça foi avariada nem tampouco se a peça foi *afundada* ou não;
- 4) Responde-se apenas quantos tiros atingiram peças, sem informar quais tiros nem qual o tipo das peças avariadas, nem tampouco se alguma peça *afundou*.

Quando um tiro não atinge uma peça é dito que foi *água*.

O quadro da direita é utilizado para marcação dos tiros já dados e para determinação da esquadra do adversário. Vence quem puser *a pique* toda a esquadra inimiga. Ao final, as esquadras são apresentadas para confirmação das respostas pelos jogadores.

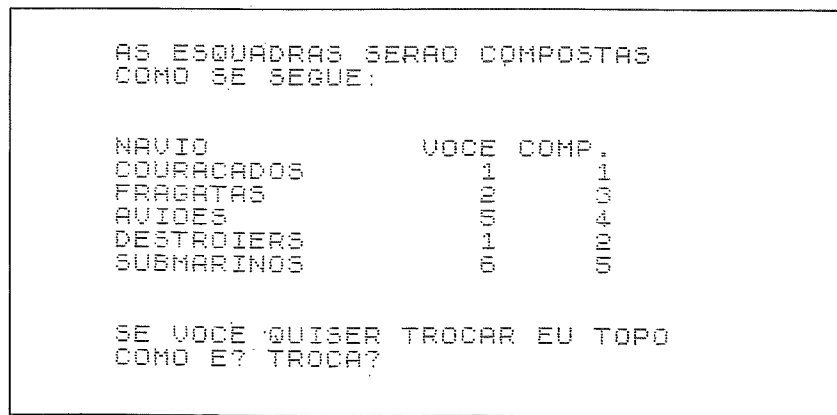


Figura 16.2

TELA DE ABERTURA

Nesta versão de BATALHA NAVAL para o TK optamos pela forma 3 de resposta, com variação no número de peças das esquadras. As esquadras são sorteadas ao acaso e apresentadas como na Figura 16.2.

Neste ponto é dado ao adversário o direito de trocar de esquadra, pois as mesmas não são iguais.

Se você desejar trocar, responda SIM ou S seguido de NEWLINE. A troca é efetuada e o TK volta a perguntar se você gostaria de trocar — é o seu direito de se arrepender.

Se não quer trocar (ou não mais), responda NAO ou N seguido de NEWLINE.

A tela se apagará para que as esquadras sejam distribuídas nos quadros correspondentes. Não é dada a opção de você distribuir a sua esquadra na versão 16K aqui descrita — a memória, 16K, é insuficiente para isto e não interessava usar linguagem de máquina.

Como a distribuição das esquadras é ao acaso, o tempo de inicialização é variável, de um a três minutos. Em média um minuto e quarenta segundos. Após isto o jogo começa, como na Figura 16.3.

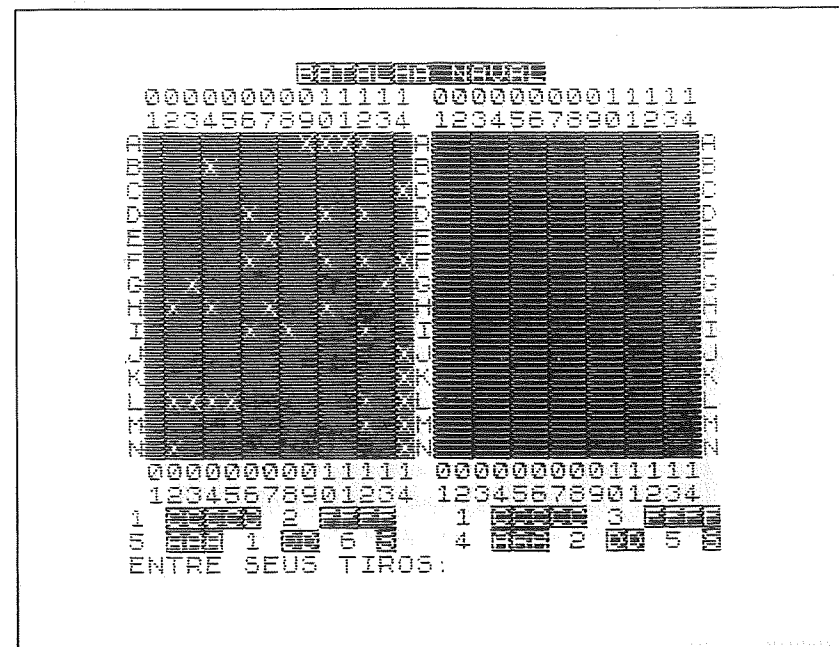


Figura 16.3

INÍCIO DO JOGO

A pergunta

ENTRE SEUS TIROS

você pode dar 3 tipos de resposta:

- 1) Se você deseja suspender a partida e deseja saber qual a esquadra do TK, tecle P e NEWLINE;
- 2) Se você deseja marcar a vizinhança das peças abatidas de forma a garantir não gastar tiros desnecessários, entre A seguido das identificações dos quadrinhos a serem marcados. Até 14 (quatorze) podem ser marcados de uma só vez. Uma vez marcado um quadrinho, não pode ser desmarcado. Porém esta marcação é diferente da resposta ÁGUA dada pelo TK aos seus tiros. Para esta marcação é usada a *vírgula*. Termine com NEWLINE. Veja o exemplo nas Figuras 16.6, 16.7 e 16.8.
- 3) Um T indica tiro. Após o T, entre as coordenadas dos quadrinhos onde você quer atirar. No máximo 3 quadrinhos. Entre NEWLINE. Veja o exemplo nas Figuras 16.4 e 16.5. Se você especificou os 3 tiros, ótimo. Se foi em locais onde você não havia atirado antes, a resposta virá. Se algum dos tiros for em local já atirado, uma mensagem é emitida e a resposta ignorada. Você terá então que entrar novamente.

Se você não especificar os três locais onde atirar, o TK complementar os tiros em quadrinhos escolhidos ao acaso entre aqueles ainda não atirados nem marcados.

O TK responderá a seus tiros pondo um ponto no quadro correspondente se a resposta for ÁGUA. Será um asterisco se a resposta for AVARIA. Observe na Figura 16.4 a resposta a TB04K11E07. Como os quadros são pretos, a resposta é destacada, por vir em branco. No próximo lance estes quadrinhos voltam a preto (sem perder o ponto ou asterisco nele contido) veja Figura 16.5.

As Figuras 16.6, 16.7 e 16.8 mostram a seqüência de respostas do TK à marcação de *água*. Observe que da Figura 16.6 à 16.7 o quadro da esquerda não se alterou. O TK não atira após responder à marcação de água. Na Figura 16.8, em resposta aos tiros dados – veja quadro da direita – temos os tiros do TK.

Você não precisa indicar ao TK as respostas aos tiros dele. Ele mesmo consulta – *com honestidade* – seu quadro. Ele marca *água* com pontos, no seu próprio quadro. A marcação de *água* pelo computador difere do tiro apenas no fato do ponto não ser inicialmente em branco.

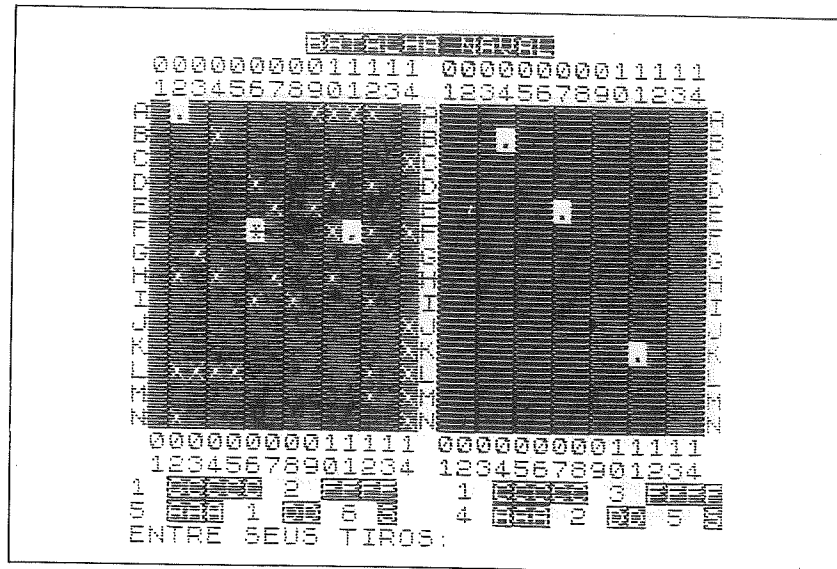


Figura 16.4

RESPOSTA A TB04K11E07.

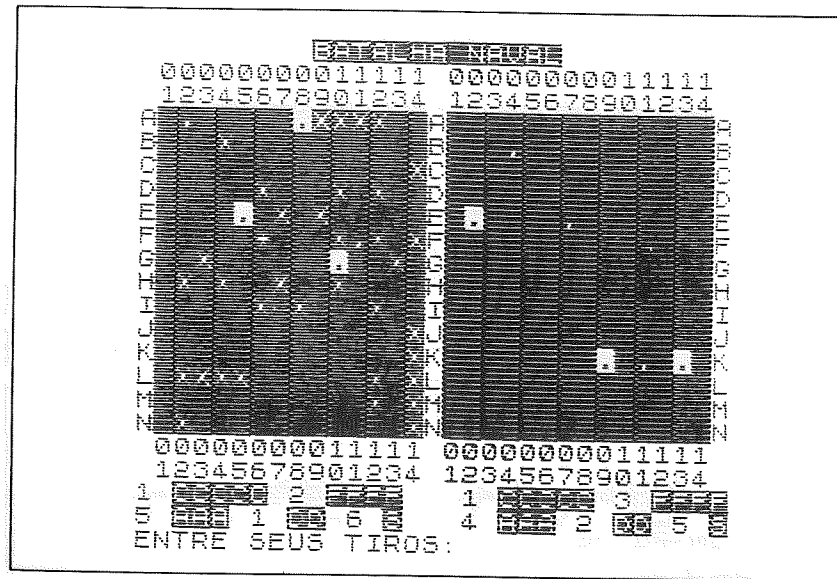


Figura 16.5

RESPOSTA SEGUINTE À FIGURA ANTERIOR

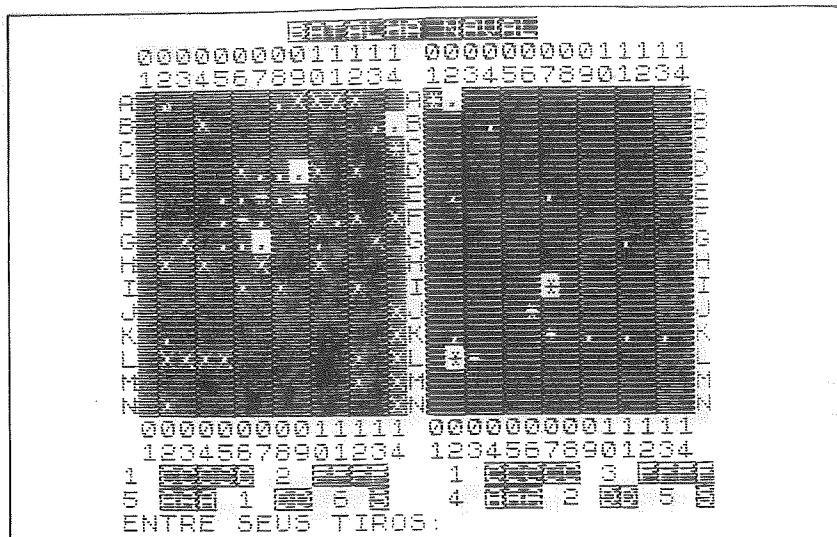


Figura 16.6 MARCAÇÃO DE ÁGUA – ANTES

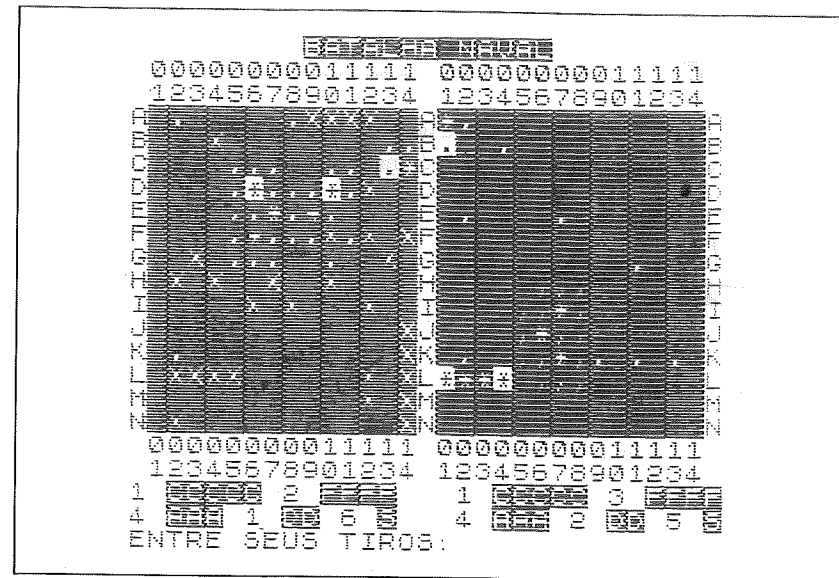


Figura 16.8 MARCAÇÃO DE ÁGUA – APÓS

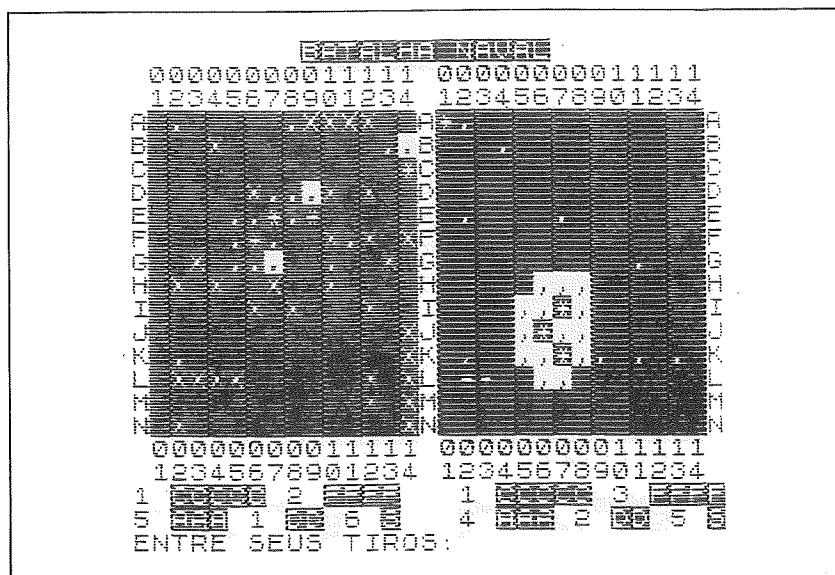


Figura 16.7 MARCAÇÃO DE ÁGUA

Alguns pontos descritos anteriormente devem ser melhor esclarecidos, dada sua importância:

- Se você responder apenas T, o TK sorteará seus tiros. Por isto é importante que você marque os quadrinhos vizinhos a uma peça já *afundada* para que o TK não escolha um destes quando da seleção dos tiros ao acaso por você;
- Se você informar apenas um ou dois quadrinhos, o TK completa com tiros ao acaso;
- Um quadrinho é identificado pela letra da linha e pelo número da coluna. Observe que a coluna é identificada sempre por 2 dígitos. Assim, A1 está *errado*. Você deve usar A01.
- O TK atualiza a relação de peças suas, as quais ele ainda falta *afundar*. Não atualiza por você as peças dele que ainda não *afundaram*. Quando ele atualiza as suas peças, o faz por ter *concluído* que aquela peça foi *afundada*. Você deve fazer o mesmo. Se ele o fizesse por você, estaria informando por *antecipação* que uma peça teria *afundado* e você levaria vantagem e as *regras básicas* não seriam respeitadas.

Cerca de 6 minutos é o tempo de carga deste programa. Ocupa cerca de 14K de texto e 2K de dados. Consulte os Apêndices sobre como usar fitas cassete.

Os quadros ficam armazenados nas variáveis U\$, C\$ e T\$. Em C\$ são anotados tiros e marcações do usuário na sua tentativa de determinar a esquadra do Computador. Em T\$ está a esquadra do Computador. Em U\$ está a esquadra do usuário.

A rotina 6000 – gerência de tela – imprime apenas U\$ e C\$ – veja Figura 6.11.

Para jogar, o computador atira ao acaso. Se foi *água*, nada a fazer. Se algum tiro foi avariado, anota nos vetores T, L e D, respectivamente tipo, local e direção.

- T = 1 ----> afundou
- T = 2 ----> avião
- T = 3 ----> horizontal
- T = 4 ----> vertical
- T = 5 ----> desconhecido

Na próxima vez, antes de atirar ao acaso, verifica se existe algum acerto anterior ainda não afundado ($T > 1$). Se tem, seleciona a rotina apropriada para escolher o tiro. Completa os três tiros ao acaso. A rotina do desconhecido – 5500 – vai cercado o quadrinho anteriormente avariado procurando determinar qual o tipo de peça. Se completar a cercadura sem acertar outro tiro, era *submarino*.

Observe na Figura 16.9 a distância relativa dos pontos que cercam um quadrinho M. O vetor R é inicializado com estes valores. Se durante a cercadura um segundo tiro é acertado, não é submarino. Se a distância relativa é 15 ou 17, em valor absoluto, tem um avião. Se ± 1 , está na horizontal. Se ± 16 , está na vertical. Em qualquer caso o T e o D são ajustados.

Se horizontal ou vertical (rotina 4500 e 5000), resta determinar se é *destróier*, *fragata* ou *couraçado*. Para isto vamos atirando, um de cada vez, na direção D = 1. Isto é, de índice crescente. Se não tem mais nada nesta direção, vai-se em direção D = 2, isto é, índice decrescentes.

	- 34	- 33	- 32	- 31	- 30
	- 18	- 17	- 16	- 15	- 14
	- 12	- 1	M	+ 1	+ 2
	+ 14	+ 15	+ 16	+ 17	+ 18
	+ 30	+ 31	+ 32	+ 33	+ 34

Figura 16.9

CERCADURA DE UM PONTO

Se diagonal, D conterá a diferença entre o primeiro e o segundo tiro.

		- 17				- 15	
	- 17		J		I		- 15
		I		+ 17	+ 15		J
			+ 17			+ 15	

Figura 16.10

TIROS DIAGONAIS

Na Figura 16.10 temos as duas alternativas de direção. Se a diferença entre I e J é ±15, a outra parte do *avião*, só pôde estar em I ± 17 ou J ± 17. Se a diferença é ±17, então a última parte somente pode ser encontrada em I ± 15 e J ± 15.

Para detalhes, veja o programa-fonte na Figura 16.11.

```

10 REM ***** BATALHA NAVAL *****
15 FAST
20 DIM P(9)
17 LET P(1)=-17
18 LET P(2)=-16
19 LET P(3)=-15
20 LET P(4)=-14
21 LET P(5)=-13
22 LET P(6)=-12
23 LET P(7)=-11
24 LET P(8)=-10
25 LET P(9)=-9
26 LET P(10)=0
27 LET P(11)=1
28 LET P(12)=2
29 LET P(13)=3
30 LET P(14)=4
31 LET P(15)=5
32 LET P(16)=6
33 LET P(17)=7
34 LET P(18)=8
35 LET P(19)=9
36 LET P(20)=10
37 LET P(21)=11
38 LET P(22)=12
39 LET P(23)=13
40 LET P(24)=14
41 LET P(25)=15
42 LET P(26)=16
43 LET P(27)=17
44 LET P(28)=18
45 LET P(29)=19
46 LET P(30)=20
47 LET P(31)=21
48 LET P(32)=22
49 LET P(33)=23
50 LET P(34)=24
51 LET P(35)=25
52 LET P(36)=26
53 LET P(37)=27
54 LET P(38)=28
55 LET P(39)=29
56 LET P(40)=30
57 LET P(41)=31
58 LET P(42)=32
59 LET P(43)=33
60 LET P(44)=34
61 LET P(45)=35
62 LET P(46)=36
63 LET P(47)=37
64 LET P(48)=38
65 LET P(49)=39
66 LET P(50)=40
67 LET P(51)=41
68 LET P(52)=42
69 LET P(53)=43
70 LET P(54)=44
71 LET P(55)=45
72 LET P(56)=46
73 LET P(57)=47
74 LET P(58)=48
75 LET P(59)=49
76 LET P(60)=50
77 LET P(61)=51
78 LET P(62)=52
79 LET P(63)=53
80 LET P(64)=54
81 LET P(65)=55
82 LET P(66)=56
83 LET P(67)=57
84 LET P(68)=58
85 LET P(69)=59
86 LET P(70)=60
87 LET P(71)=61
88 LET P(72)=62
89 LET P(73)=63
90 LET P(74)=64
91 LET P(75)=65
92 LET P(76)=66
93 LET P(77)=67
94 LET P(78)=68
95 LET P(79)=69
96 LET P(80)=70
97 LET P(81)=71
98 LET P(82)=72
99 LET P(83)=73
100 LET P(84)=74
101 LET P(85)=75
102 LET P(86)=76
103 LET P(87)=77
104 LET P(88)=78
105 LET P(89)=79
106 LET P(90)=80
107 LET P(91)=81
108 LET P(92)=82
109 LET P(93)=83
110 LET P(94)=84
111 LET P(95)=85
112 LET P(96)=86
113 LET P(97)=87
114 LET P(98)=88
115 LET P(99)=89
116 LET P(100)=90
117 LET P(101)=91
118 LET P(102)=92
119 LET P(103)=93
120 LET P(104)=94
121 LET P(105)=95
122 LET P(106)=96
123 LET P(107)=97
124 LET P(108)=98
125 LET P(109)=99
126 LET P(110)=100
127 LET P(111)=101
128 LET P(112)=102
129 LET P(113)=103
130 LET P(114)=104
131 LET P(115)=105
132 LET P(116)=106
133 LET P(117)=107
134 LET P(118)=108
135 LET P(119)=109
136 LET P(120)=110
137 LET P(121)=111
138 LET P(122)=112
139 LET P(123)=113
140 LET P(124)=114
141 LET P(125)=115
142 LET P(126)=116
143 LET P(127)=117
144 LET P(128)=118
145 LET P(129)=119
146 LET P(130)=120
147 LET P(131)=121
148 LET P(132)=122
149 LET P(133)=123
150 LET P(134)=124
151 LET P(135)=125
152 LET P(136)=126
153 LET P(137)=127
154 LET P(138)=128
155 LET P(139)=129
156 LET P(140)=130
157 LET P(141)=131
158 LET P(142)=132
159 LET P(143)=133
160 LET P(144)=134
161 LET P(145)=135
162 LET P(146)=136
163 LET P(147)=137
164 LET P(148)=138
165 LET P(149)=139
166 LET P(150)=140
167 LET P(151)=141
168 LET P(152)=142
169 LET P(153)=143
170 LET P(154)=144
171 LET P(155)=145
172 LET P(156)=146
173 LET P(157)=147
174 LET P(158)=148
175 LET P(159)=149
176 LET P(160)=150
177 LET P(161)=151
178 LET P(162)=152
179 LET P(163)=153
180 LET P(164)=154
181 LET P(165)=155
182 LET P(166)=156
183 LET P(167)=157
184 LET P(168)=158
185 LET P(169)=159
186 LET P(170)=160
187 LET P(171)=161
188 LET P(172)=162
189 LET P(173)=163
190 LET P(174)=164
191 LET P(175)=165
192 LET P(176)=166
193 LET P(177)=167
194 LET P(178)=168
195 LET P(179)=169
196 LET P(180)=170
197 LET P(181)=171
198 LET P(182)=172
199 LET P(183)=173
200 LET P(184)=174
201 LET P(185)=175
202 LET P(186)=176
203 LET P(187)=177
204 LET P(188)=178
205 LET P(189)=179
206 LET P(190)=180
207 LET P(191)=181
208 LET P(192)=182
209 LET P(193)=183
210 LET P(194)=184
211 LET P(195)=185
212 LET P(196)=186
213 LET P(197)=187
214 LET P(198)=188
215 LET P(199)=189
216 LET P(200)=190
217 LET P(201)=191
218 LET P(202)=192
219 LET P(203)=193
220 LET P(204)=194
221 LET P(205)=195
222 LET P(206)=196
223 LET P(207)=197
224 LET P(208)=198
225 LET P(209)=199
226 LET P(210)=200
227 LET P(211)=201
228 LET P(212)=202
229 LET P(213)=203
230 LET P(214)=204
231 LET P(215)=205
232 LET P(216)=206
233 LET P(217)=207
234 LET P(218)=208
235 LET P(219)=209
236 LET P(220)=210
237 LET P(221)=211
238 LET P(222)=212
239 LET P(223)=213
240 LET P(224)=214
241 LET P(225)=215
242 LET P(226)=216
243 LET P(227)=217
244 LET P(228)=218
245 LET P(229)=219
246 LET P(230)=220
247 LET P(231)=221
248 LET P(232)=222
249 LET P(233)=223
250 LET P(234)=224
251 LET P(235)=225
252 LET P(236)=226
253 LET P(237)=227
254 LET P(238)=228
255 LET P(239)=229
256 LET P(240)=230
257 LET P(241)=231
258 LET P(242)=232
259 LET P(243)=233
260 LET P(244)=234
261 LET P(245)=235
262 LET P(246)=236
263 LET P(247)=237
264 LET P(248)=238
265 LET P(249)=239
266 LET P(250)=240
267 LET P(251)=241
268 LET P(252)=242
269 LET P(253)=243
270 LET P(254)=244
271 LET P(255)=245
272 LET P(256)=246
273 LET P(257)=247
274 LET P(258)=248
275 LET P(259)=249
276 LET P(260)=250
277 LET P(261)=251
278 LET P(262)=252
279 LET P(263)=253
280 LET P(264)=254
281 LET P(265)=255
282 LET P(266)=256
283 LET P(267)=257
284 LET P(268)=258
285 LET P(269)=259
286 LET P(270)=260
287 LET P(271)=261
288 LET P(272)=262
289 LET P(273)=263
290 LET P(274)=264
291 LET P(275)=265
292 LET P(276)=266
293 LET P(277)=267
294 LET P(278)=268
295 LET P(279)=269
296 LET P(280)=270
297 LET P(281)=271
298 LET P(282)=272
299 LET P(283)=273
300 LET P(284)=274
301 LET P(285)=275
302 LET P(286)=276
303 LET P(287)=277
304 LET P(288)=278
305 LET P(289)=279
306 LET P(290)=280
307 LET P(291)=281
308 LET P(292)=282
309 LET P(293)=283
310 LET P(294)=284
311 LET P(295)=285
312 LET P(296)=286
313 LET P(297)=287
314 LET P(298)=288
315 LET P(299)=289
316 LET P(300)=290
317 LET P(301)=291
318 LET P(302)=292
319 LET P(303)=293
320 LET P(304)=294
321 LET P(305)=295
322 LET P(306)=296
323 LET P(307)=297
324 LET P(308)=298
325 LET P(309)=299
326 LET P(310)=300
327 LET P(311)=301
328 LET P(312)=302
329 LET P(313)=303
330 LET P(314)=304
331 LET P(315)=305
332 LET P(316)=306
333 LET P(317)=307
334 LET P(318)=308
335 LET P(319)=309
336 LET P(320)=310
337 LET P(321)=311
338 LET P(322)=312
339 LET P(323)=313
340 LET P(324)=314
341 LET P(325)=315
342 LET P(326)=316
343 LET P(327)=317
344 LET P(328)=318
345 LET P(329)=319
346 LET P(330)=320
347 LET P(331)=321
348 LET P(332)=322
349 LET P(333)=323
350 LET P(334)=324
351 LET P(335)=325
352 LET P(336)=326
353 LET P(337)=327
354 LET P(338)=328
355 LET P(339)=329
356 LET P(340)=330
357 LET P(341)=331
358 LET P(342)=332
359 LET P(343)=333
360 LET P(344)=334
361 LET P(345)=335
362 LET P(346)=336
363 LET P(347)=337
364 LET P(348)=338
365 LET P(349)=339
366 LET P(350)=340
367 LET P(351)=341
368 LET P(352)=342
369 LET P(353)=343
370 LET P(354)=344
371 LET P(355)=345
372 LET P(356)=346
373 LET P(357)=347
374 LET P(358)=348
375 LET P(359)=349
376 LET P(360)=350
377 LET P(361)=351
378 LET P(362)=352
379 LET P(363)=353
380 LET P(364)=354
381 LET P(365)=355
382 LET P(366)=356
383 LET P(367)=357
384 LET P(368)=358
385 LET P(369)=359
386 LET P(370)=360
387 LET P(371)=361
388 LET P(372)=362
389 LET P(373)=363
390 LET P(374)=364
391 LET P(375)=365
392 LET P(376)=366
393 LET P(377)=367
394 LET P(378)=368
395 LET P(379)=369
396 LET P(380)=370
397 LET P(381)=371
398 LET P(382)=372
399 LET P(383)=373
400 LET P(384)=374
401 LET P(385)=375
402 LET P(386)=376
403 LET P(387)=377
404 LET P(388)=378
405 LET P(389)=379
406 LET P(390)=380
407 LET P(391)=381
408 LET P(392)=382
409 LET P(393)=383
410 LET P(394)=384
411 LET P(395)=385
412 LET P(396)=386
413 LET P(397)=387
414 LET P(398)=388
415 LET P(399)=389
416 LET P(400)=390
417 LET P(401)=391
418 LET P(402)=392
419 LET P(403)=393
420 LET P(404)=394
421 LET P(405)=395
422 LET P(406)=396
423 LET P(407)=397
424 LET P(408)=398
425 LET P(409)=399
426 LET P(410)=400
427 LET P(411)=401
428 LET P(412)=402
429 LET P(413)=403
430 LET P(414)=404
431 LET P(415)=405
432 LET P(416)=406
433 LET P(417)=407
434 LET P(418)=408
435 LET P(419)=409
436 LET P(420)=410
437 LET P(421)=411
438 LET P(422)=412
439 LET P(423)=413
440 LET P(424)=414
441 LET P(425)=415
442 LET P(426)=416
443 LET P(427)=417
444 LET P(428)=418
445 LET P(429)=419
446 LET P(430)=420
447 LET P(431)=421
448 LET P(432)=422
449 LET P(433)=423
450 LET P(434)=424
451 LET P(435)=425
452 LET P(436)=426
453 LET P(437)=427
454 LET P(438)=428
455 LET P(439)=429
456 LET P(440)=430
457 LET P(441)=431
458 LET P(442)=432
459 LET P(443)=433
460 LET P(444)=434
461 LET P(445)=435
462 LET P(446)=436
463 LET P(447)=437
464 LET P(448)=438
465 LET P(449)=439
466 LET P(450)=440
467 LET P(451)=441
468 LET P(452)=442
469 LET P(453)=443
470 LET P(454)=444
471 LET P(455)=445
472 LET P(456)=446
473 LET P(457)=447
474 LET P(458)=448
475 LET P(459)=449
476 LET P(460)=450
477 LET P(461)=451
478 LET P(462)=452
479 LET P(463)=453
480 LET P(464)=454
481 LET P(465)=455
482 LET P(466)=456
483 LET P(467)=457
484 LET P(468)=458
485 LET P(469)=459
486 LET P(470)=460
487 LET P(471)=461
488 LET P(472)=462
489 LET P(473)=463
490 LET P(474)=464
491 LET P(475)=465
492 LET P(476)=466
493 LET P(477)=467
494 LET P(478)=468
495 LET P(479)=469
496 LET P(480)=470
497 LET P(481)=471
498 LET P(482)=472
499 LET P(483)=473
500 LET P(484)=474
501 LET P(485)=475
502 LET P(486)=476
503 LET P(487)=477
504 LET P(488)=478
505 LET P(489)=479
506 LET P(490)=480
507 LET P(491)=481
508 LET P(492)=482
509 LET P(493)=483
510 LET P(494)=484
511 LET P(495)=485
512 LET P(496)=486
513 LET P(497)=487
514 LET P(498)=488
515 LET P(499)=489
516 LET P(500)=490
517 LET P(501)=491
518 LET P(502)=492
519 LET P(503)=493
520 LET P(504)=494
521 LET P(505)=495
522 LET P(506)=496
523 LET P(507)=497
524 LET P(508)=498
525 LET P(509)=499
526 LET P(510)=500
527 LET P(511)=501
528 LET P(512)=502
529 LET P(513)=503
530 LET P(514)=504
531 LET P(515)=505
532 LET P(516)=506
533 LET P(517)=507
534 LET P(518)=508
535 LET P(519)=509
536 LET P(520)=510
537 LET P(521)=511
538 LET P(522)=512
539 LET P(523)=513
540 LET P(524)=514
541 LET P(525)=515
542 LET P(526)=516
543 LET P(527)=517
544 LET P(528)=518
545 LET P(529)=519
546 LET P(530)=520
547 LET P(531)=521
548 LET P(532)=522
549 LET P(533)=523
550 LET P(534)=524
551 LET P(535)=525
552 LET P(536)=526
553 LET P(537)=527
554 LET P(538)=528
555 LET P(539)=529
556 LET P(540)=530
557 LET P(541)=531
558 LET P(542)=532
559 LET P(543)=533
560 LET P(544)=534
561 LET P(545)=535
562 LET P(546)=536
563 LET P(547)=537
564 LET P(548)=538
565 LET P(549)=539
566 LET P(550)=540
567 LET P(551)=541
568 LET P(552)=542
569 LET P(553)=543
570 LET P(554)=544
571 LET P(555)=545
572 LET P(556)=546
573 LET P(557)=547
574 LET P(558)=548
575 LET P(559)=549
576 LET P(560)=550
577 LET P(561)=551
578 LET P(562)=552
579 LET P(563)=553
580 LET P(564)=554
581 LET P(565)=555
582 LET P(566)=556
583 LET P(567)=557
584 LET P(568)=558
585 LET P(569)=559
586 LET P(570)=560
587 LET P(571)=561
588 LET P(572)=562
589 LET P(573)=563
590 LET P(574)=564
591 LET P(575)=565
592 LET P(576)=566
593 LET P(577)=567
594 LET P(578)=568
595 LET P(579)=569
596 LET P(580)=570
597 LET P(581)=571
598 LET P(582)=572
599 LET P(583)=573
600 LET P(584)=574
601 LET P(585)=575
602 LET P(586)=576
603 LET P(587)=577
604 LET P(588)=578
605 LET P(589)=579
606 LET P(590)=580
607 LET P(591)=581
608 LET P(592)=582
609 LET P(593)=583
610 LET P(594)=584
611 LET P(595)=585
612 LET P(596)=586
613 LET P(597)=587
614 LET P(598)=588
615 LET P(599)=589
616 LET P(600)=590
617 LET P(601)=591
618 LET P(602)=592
619 LET P(603)=593
620 LET P(604)=594
621 LET P(605)=595
622 LET P(606)=596
623 LET P(607)=597
624 LET P(608)=598
625 LET P(609)=599
626 LET P(610)=600
627 LET P(611)=601
628 LET P(612)=602
629 LET P(613)=603
630 LET P(614)=604
631 LET P(615)=605
632 LET P(616)=606
633 LET P(617)=607
634 LET P(618)=608
635 LET P(619)=609
636 LET P(620)=610
637 LET P(621)=611
638 LET P(622)=612
639 LET P(623)=613
640 LET P(624)=614
641 LET P(625)=615
642 LET P(626)=616
643 LET P(627)=617
644 LET P(628)=618
645 LET P(629)=619
646 LET P(630)=620
647 LET P(631)=621
648 LET P(632)=622
649 LET P(633)=623
650 LET P(634)=624
651 LET P(635)=625
652 LET P(636)=626
653 LET P(637)=627
654 LET P(638)=628
655 LET P(639)=629
656 LET P(640)=630
657 LET P(641)=631
658 LET P(642)=632
659 LET P(643)=633
660 LET P(644)=634
661 LET P(645)=635
662 LET P(646)=636
663 LET P(647)=637
664 LET P(648)=638
665 LET P(649)=639
666 LET P(650)=640
667 LET P(651)=641
668 LET P(652)=642
669 LET P(653)=643
670 LET P(654)=644
671 LET P(655)=645
672 LET P(656)=646
673 LET P(657)=647
674 LET P(658)=648
675 LET P(659)=649
676 LET P(680)=650
677 LET P(681)=651
678 LET P(682)=652
679 LET P(683)=653
680 LET P(684)=654
681 LET P(685)=655
682 LET P(686)=656
683 LET P(687)=657
684 LET P(688)=658
685 LET P(689)=659
686 LET P(690)=660
687 LET P(691)=661
688 LET P(692)=662
689 LET P(693)=663
690 LET P(694)=664
691 LET P(695)=665
692 LET P(696)=666
693 LET P(697)=667
694 LET P(698)=668
695 LET P(699)=669
696 LET P(700)=670
697 LET P(701)=671
698 LET P(702)=672
699 LET P(703)=673
700 LET P(704)=674
701 LET P(705)=675
702 LET P(706)=676
703 LET P(707)=677
704 LET P(708)=678
705 LET P(709)=679
706 LET P(710)=680
707 LET P(711)=681
708 LET P(712)=682
709 LET P(713)=683
710 LET P(714)=684
711 LET P(715)=685
712 LET P(716)=686
713 LET P(717)=687
714 LET P(718)=688
715 LET P(719)=689
716 LET P(720)=690
717 LET P(721)=691
718 LET P(722)=692
719 LET P(723)=693
720 LET P(724)=694
721 LET P(725)=695
722 LET P(726)=696
723 LET P(727)=697
724 LET P(728)=698
725 LET P(729)=699
726 LET P(730)=700
727 LET P(731)=701
728 LET P(732)=702
729 LET P(733)=703
730 LET P(734)=704
731 LET P(735)=705
732 LET P(736)=706
733 LET P(737)=707
734 LET P(738)=708
735 LET P(739)=709
736 LET P(740)=710
737 LET P(741)=711
738 LET P(742)=712
739 LET P(743)=713
740 LET P(744)=714
741 LET P(745)=715
742 LET P(746)=716
743 LET P(747)=717
744 LET P(748)=718
745 LET P(749)=719
746 LET P(750)=720
747 LET P(751)=721
748 LET P(752)=722
749 LET P(753)=723
750 LET P(754)=724
751 LET P(755)=725
752 LET P(756)=726
753 LET P(757)=727
754 LET P(758)=728
755 LET P(759)=729
756 LET P(760)=730
757 LET P(761)=731
758 LET P(762)=732
759 LET P(763)=733
760 LET P(764)=734
761 LET P(765)=735
762 LET P(766)=736
763 LET P(767)=737
764 LET P(768)=738
765 LET P(769)=739
766 LET P(770)=740
767 LET P(771)=741
768 LET P(772)=742
769 LET P(773)=743
770 LET P(774)=744
771 LET P(775)=745
772 LET P(776)=746
773 LET P(777)=747
774 LET P(778)=748
775 LET P(779)=749
776 LET P(780)=750
777 LET P(781)=751
778 LET P(782)=752
779 LET P(783)=753
780 LET P(784)=754
781 LET P(785)=755
782 LET P(786)=756
783 LET P(787)=757
784 LET P(788)=758
785 LET P(789)=759
786 LET P(790)=760
787 LET P(791)=761
788 LET P(792)=762
789 LET P(793)=763
790 LET P(794)=764
791 LET P(795)=765
792 LET P(796)=766
793 LET P(797)=767
794 LET P(798)=768
795 LET P(799)=769
796 LET P(800)=770
797 LET P(801)=771
798 LET P(802)=772
799 LET P(803)=773
800 LET P(804)=774
801 LET P(805)=775
802 LET P(806)=776
803 LET P(807)=777
804 LET P(808)=778
805 LET P(809)=779
806 LET P(810)=780
807 LET P(811)=781
808 LET P(812)=782
809 LET P(813)=783
810 LET P(814)=784
811 LET P(815)=785
812 LET P(816)=786
813 LET P(817)=787
814 LET P(818)=788
815 LET P(819)=789
816 LET P(820)=790
817 LET P(821)=791
818 LET P(822)=792
819 LET P(823)=793
820 LET P(824)=794
821 LET P(825)=795
822 LET P(826)=796
823 LET P(827)=797
824 LET P(828)=798
825 LET P(829)=799
826 LET P(830)=800
827 LET P(831)=801
828 LET P(832)=802
829 LET P(833)=803
830 LET P(834)=804
831 LET P(835)=805
832 LET P(836)=806
833 LET P(837)=807
834 LET P(838)=808
835 LET P(839)=809
836 LET P(840)=810
837 LET P(841)=811
838 LET P(842)=812
839 LET P(843)=813
840 LET P(844)=814
841 LET P(845)=815
842 LET P(846)=816
843 LET P(847)=817
844 LET P(848)=818
845 LET P(849)=819
846 LET P(850)=820
847 LET P(851)=821
848 LET P(852)=822
849 LET P(853)=823
850 LET P(854)=824
851 LET P(855)=825
852 LET P(856)=826
853 LET P(857)=827
854 LET P(858)=828
855 LET P(859)=829
856 LET P(860)=830
857 LET P(861)=831
858 LET P(862)=832
859 LET P(863)=833
860 LET P(864)=834
861 LET P(865)=835
862 LET P(866)=836
863 LET P(867)=837
864 LET P(868)=838
865 LET P(869)=839
866 LET P(870)=840
867 LET P(871)=841
868 LET P(872)=842
869 LET P(873)=843
870 LET P(874)=844
871 LET P(875)=845
872 LET P(876)=846
873 LET P(877)=847
874 LET P(878)=848
875 LET P(879)=849
876 LET P(880)=850
877 LET P(881)=851
878 LET P(882)=852
879 LET P(883)=853
880 LET P(884)=854
881 LET P(885)=855
882 LET P(886)=856
883 LET P(887)=857
884 LET P(888)=858
885 LET P(889)=859
886 LET P(890)=860
887 LET P(891)=861
888 LET P(892)=862
889 LET P(893)=863
890 LET P(894)=864
891 LET P(895)=865
892 LET P(896)=866
893 LET P(897)=867
894 LET P(898)=868
895 LET P(899)=869
896 LET P(900)=870
897 LET P(901)=871
898 LET P(902)=872
899 LET P(903)=873
900 LET P(904)=874
901 LET P(905)=875
902 LET P(906)=876
903 LET P(907)=877
904 LET P(908)=878
905 LET P(909)=879
906 LET P(910)=
```

```

530 PRINT "VOCE DEVE RESPONDER
531H OU NAO"
535 SCROLL
540 GOTO 530
550 CLS
600 PRINT TAB 9;"██████████NEWS"
630 GOSUB 680
640 FOR I=2 TO 15
645 LET J1=I+16-14
646 LET J2=J1+13
650 LET A#=0$(I TO I)
655 PRINT TAB 0;A#;U$(J1 TO J2)
;A#;C$(J1 TO J2);A#
665 NEXT I
680 PRINT " 000000000011111 0000
0000011111 12345678901234 1234
5678901234"
690 RETURN
710 LET AV=1
715 LET NU=2
720 LET V=CA
730 GOSUB 8000
735 LET AV=0
736 LET V=CD
738 GOSUB 8000
740 LET NU=1
750 LET V=CB
760 GOSUB 8000
770 LET NU=5
780 LET V=CC
790 GOSUB 8000
800 LET NU=4
810 LET V=CF
820 GOSUB 8000
830 FOR I=1 TO 15
835 LET T$(I TO I)="█"
840 LET M=257-I
845 LET T$(M TO M)="█"
850 LET M=I+16
855 LET T$(M TO M)="█"
860 LET M=M+1
865 LET T$(M TO M)="█"
870 NEXT I
880 RETURN
900 GOSUB 250
910 GOTO 400

1000 GOSUB 250
1080 GOSUB 710
1090 LET U#=T#
1100 GOSUB 100
1130 GOSUB 250

```

Figura 16.11 LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

```

1140 GOSUB 710
1180 GOSUB 800
1190 GOSUB 6900
1200 GOSUB 5700
1210 GOSUB 6900
1220 PRINT AT 21,0;"ENTRE SEUS T
IROS:"
1230 INPUT A#
1240 IF (LEN A#)<1 THEN GOTO 133
0
1245 LET F#=A$(1 TO 1)
1248 LET P#=A$(2 TO 1)
1250 IF F#="A" THEN GOTO 1500
1255 IF F#="T" THEN GOTO 1400
1258 IF F#="P" THEN GOTO 5820
1260 GOSUB 9900
1262 PRINT AT 21,0;"TIRO OU AGUA
?"
1264 PAUSE 300
1265 GOTO 1220
1280 FOR I=1 TO TT
1290 LET C$(X(I) TO X(I))=","
1295 NEXT I
1298 GOTO 1180
1300 FOR I=1 TO TT
1310 IF T$(X(I) TO X(I))="█" THE
N GOTO 1340
1320 LET C$(X(I) TO X(I))="*"
1325 LET CP=CP-1
1330 GOTO 1350
1340 LET C$(X(I) TO X(I))=","
1350 NEXT I
1360 GOTO 3000
1400 GOSUB 2000
1410 IF (LEN A#)=0 THEN GOTO 160
0
1420 GOTO 1510
1500 GOSUB 3000
1510 GOSUB 2100
1520 GOSUB 2300
1530 IF V=0 THEN GOTO 1540
1532 LET TT=0
1535 GOTO 1180
1540 IF F#="A" THEN GOTO 1280
1550 IF TT<=3 THEN GOTO 1500
1560 GOSUB 9900
1570 PRINT AT 21,0;"NO MAXIMO 3
TIROS."
1580 PAUSE 300
1590 GOTO 1210
1600 LET ER=1690
1610 FOR V=TT+1 TO 3

```

Figura 16.11 LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

```

1630 GOSUB 6500
1635 GOSUB 6540
1630 IF C$(M TO M) <> "■" THEN GOT
0 1625
1635 LET C$(M TO M) = " "
1640 LET X(U) = M
1650 NEXT U
1670 LET TT = 3
1680 GOTO 1300
1690 GOSUB 9900
1700 PRINT AT 21,0;"VOCE MARCOU
■■■■ ERRADAMENTE"
1710 PAUSE 300
1712 GOSUB 9900
1715 PRINT AT 21,0;"NAO TEM ONDE
JOGAR MAIS."
1717 PAUSE 300
1720 LET TT = U - 1
1730 GOTO 1300

2000 IF TT < 1 THEN RETURN
2010 FOR I = 1 TO TT
2015 IF CODE (C$(X(I) TO )) >= 128
THEN GOTO 2030
2020 LET C$(X(I) TO X(I)) = CHR$ (
CODE (C$(X(I) TO )) + 128)
2030 NEXT I
2040 LET TT = 0
2050 RETURN
2100 IF (LEN A$) < 3 THEN GOTO 215
0
2105 LET U = TT + 1
2105 IF U > 14 THEN GOTO 2400
2110 LET K$ = A$(1 TO 1)
2120 FOR I = 2 TO 15
2130 IF K$ = 0$(I TO I) THEN GOTO
2200
2140 NEXT I
2150 GOSUB 9900
2160 PRINT AT 21,0;"VOCE ENTROU
NO FORMATO ERRADO."
2170 PAUSE 300
2180 GOTO 1180
2200 LET X(U) = I - 1
2210 LET K$ = A$(2 TO 3)
2220 FOR I = 0 TO 15
2222 LET J1 = I * 2 + 1
2226 LET J2 = J1 + 1
2230 IF K$ = U$(J1 TO J2) THEN GOT
0 2220
2240 NEXT I
2250 GOTO 2150

```

Figura 16.11

LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

```

2260 LET X(U) = X(U) * 16 + I + 1
2270 LET TT = U
2280 LET A$ = A$(4 TO )
2290 IF (LEN A$) <> 0 THEN GOTO 21
00
2295 RETURN
2300 LET U = 0
2310 FOR I = 1 TO TT
2320 IF C$(X(I) TO X(I)) = "■" THE
N GOTO 2390
2330 IF C$(X(I) TO X(I)) = "■" THE
N GOTO 2390
2340 IF C$(X(I) TO X(I)) = " " THE
N GOTO 2390
2350 GOSUB 9900
2352 LET M = INT (X(I) / 16)
2354 LET N1 = X(I) - M * 16
2356 LET M = M + 1
2358 LET N2 = N1 + 1
2360 PRINT AT 21,0;"VOCE JA HAVI
A ATIRADO EM "; 0$(M TO M); U$(N1
TO N2);
2370 PAUSE 200
2380 LET U = 1
2390 NEXT I
2395 RETURN
2400 GOSUB 9900
2410 PRINT AT 21,0;"NO MAXIMO 14
MARCAC0ES DE AGUA"
2420 PAUSE 300
2430 RETURN

3000 GOSUB 6700
3005 GOSUB 3300
3010 LET AL = 1
3020 IF 0 <= 0 THEN GOTO 3070
3030 FOR K = 1 TO 0
3040 GOSUB (3000 + T(K) * 500)
3050 IF AL > 3 THEN GOTO 1180
3060 NEXT K
3065 LET ER = 3270
3070 FOR K = AL TO 3
3080 GOSUB 6500
3090 GOSUB 6540
3100 IF U$(M TO M) = "■" THEN GOTO
3130
3110 IF U$(M TO M) = "■" THEN GOTO
3150
3120 GOTO 3090
3130 GOSUB 3700
3140 GOTO 3270
3150 GOSUB 3720

```

Figura 16.11

LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

```

3160 LET D=D+1
3170 IF D>20 THEN GOTO 3500
3180 LET L(D)=M
3230 LET T(D)=5
3260 LET D(D)=0
3270 NEXT K
3280 GOTO 1180
3300 IF G<1 THEN RETURN
3310 FOR I=1 TO G
3315 IF CODE (U$(Y(I) TO ))>=128
THEN GOTO 3330
3320 LET U$(Y(I) TO Y(I))=CHR$ (
CODE (U$(Y(I) TO ))+128)
3330 NEXT I
3340 LET G=0
3350 RETURN
3400 GOSUB 9900
3410 PRINT AT 21,0;"ERRO DE LOGI
CA RARO. ABORTO"
3420 STOP
3430 GOTO 3400
3500 RETURN
3500 GOSUB 9900
3610 PRINT AT 21,0;"NUMERO DE OB
JETOS >20"
3620 STOP
3630 GOTO 3600
3700 LET A$="."
3710 GOTO 3730
3720 LET A$="*"
3721 IF U$(M TO M)(<>"#") AND U$(M
TO M)(<>"#") THEN STOP
3725 LET UP=UP-1
3730 LET G=G+1
3740 LET Y(G)=M
3755 LET AL=AL+1
3770 LET U$(M TO M)=A$
3780 RETURN

4000 LET M=L(K)
4010 LET M1=L(K)-D(K)
4065 LET J=15
4070 IF ABS D(K)=17 THEN GOTO 40
80
4075 LET J=17
4080 LET I=M-J
4085 LET D=1
4090 GOTO 4241
4100 LET I=M+J
4110 LET D=2
4115 GOTO 4241
4120 LET I=M1-J

```

Figura 16.11

LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL) (cont.)

```

4132 LET D=3
4135 GOTO 4241
4140 LET I=M1+J
4152 LET D=4
4155 GOTO 4241
4240 GOTO 4450
4241 LET A$=U$(I TO I)
4242 IF A$="#" OR A$="*" THEN GO
TO 4260
4245 IF A$="." OR A$="." THEN GO
TO (4080+D*20)
4248 LET M=I
4250 IF A$="#" THEN GOTO 3700
4255 GOSUB 3720
4260 LET T(K)=1
4265 LET UA=UA-1
4270 IF UA<0 THEN LET UA=0
4300 FOR J=1 TO 9
4310 LET I=R(J)+M
4315 IF U$(I TO I)="#" THEN LET
U$(I TO I)="#"
4320 NEXT J
4340 RETURN
4450 GOSUB 9900
4460 PRINT AT 21,0;"QUE AVIAO E
ESTE?";
4470 STOP
4480 GOTO 4450

4500 IF D(K)=2 THEN GOTO 4700
4550 LET I=L(K)
4555 LET M1=I+5
4565 FOR M=I TO M1
4567 LET A$=U$(M TO M)
4570 IF A$="#" THEN GOTO 4670
4571 IF A$="#" THEN GOTO 4600
4575 IF A$="." OR A$="." THEN GO
TO 4590
4580 LET L(K)=M
4585 NEXT M
4590 LET D(K)=2
4595 RETURN
4600 GOSUB 3720
4620 IF D(K)=2 THEN GOTO 4640
4625 LET M=M-15
4635 GOTO 4650
4640 LET M=M-17
4650 LET U$(M TO M)="#"
4652 LET M=M+32
4655 LET U$(M TO M)="#"
4660 RETURN
4670 GOSUB 3700

```

Figura 16.11

LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL) (cont.)

```

4671 LET M=I
4675 IF D(K) <> 2 THEN GOTO 4590
4680 GOTO 4740
4700 LET I=L(K)
4705 LET M1=I-5
4710 FOR M=I TO M1 STEP -1
4715 LET A#=U$(M TO M)
4715 IF A#="█" THEN GOTO 4670
4720 IF A#="█" THEN GOTO 4600
4725 IF A#="█" OR A#="." THEN GO
TO 4740
4725 LET L(K)=M
4730 NEXT M
4740 LET T(K)=1
4745 LET M=L(K)
4750 LET M=M-1
4755 IF U$(M TO M)="█" OR U$(M T
O M)="*" THEN GOTO 4750
4755 LET M=M+1
4765 LET M1=M+5
4775 LET D=0
4780 FOR I=M TO M1
4785 IF U$(I TO I) <> "█" AND U$(I
TO I) <> "*" THEN GOTO 4840
4790 LET D=D+1
4795 FOR C=1 TO D
4800 IF T(C)=1 THEN GOTO 4820
4810 IF L(C) <> I THEN GOTO 4820
4815 LET T(C)=1
4820 NEXT C
4825 NEXT I
4850 GOTO 5000

5000 IF D(K)=2 THEN GOTO 5200
5010 LET I=L(K)
5020 LET M1=I+80
5030 FOR M=I TO M1 STEP 16
5040 LET A#=U$(M TO M)
5070 IF A#="█" THEN GOTO 5170
5071 IF A#="█" THEN GOTO 5100
5075 IF A#="." OR A#="█" THEN GO
TO 5090
5080 LET L(K)=M
5085 NEXT M
5090 LET D(K)=2
5095 RETURN
5100 GOSUB 3720
5120 IF D(K)=2 THEN GOTO 5140
5125 LET M=M+17
5135 GOTO 5150
5140 LET M=M-15
5150 LET U$(M TO M)="█"

```

Figura 16.11

LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

```

5152 LET M=M-2
5155 LET U$(M TO M)="█"
5160 RETURN
5170 GOSUB 3700
5171 LET M=I
5175 IF D(K) <> 2 THEN GOTO 5090
5180 GOTO 5240
5200 LET I=L(K)
5205 LET M1=I-80
5210 FOR M=I TO M1 STEP -16
5215 LET A#=U$(M TO M)
5215 IF A#="█" THEN GOTO 5170
5220 IF A#="█" THEN GOTO 5100
5225 IF A#="█" OR A#="." THEN GO
TO 5240
5225 LET L(K)=M
5230 NEXT M
5240 LET T(K)=1
5245 LET M=L(K)
5250 LET M=M-16
5255 IF U$(M TO M)="█" OR U$(M T
O M)="*" THEN GOTO 5250
5260 LET M=M+16
5265 LET M1=M+80
5275 LET D=0
5280 FOR I=M TO M1 STEP 16
5285 IF U$(I TO I) <> "█" AND U$(I
TO I) <> "*" THEN GOTO 5340
5290 LET D=D+1
5295 FOR C=1 TO D
5300 IF T(C)=1 THEN GOTO 5320
5310 IF L(C) <> I THEN GOTO 5320
5315 LET T(C)=1
5320 NEXT C
5325 NEXT I
5350 GOTO 5000

5500 LET C=0
5510 FOR I=1 TO 9
5520 LET J=L(K)+R(I)
5525 LET A#=U$(J TO J)
5530 IF A#="█" OR A#="." THEN GO
TO 5550
5535 IF A# <> "█" AND A# <> "*" THEN
GOTO 5600
5540 LET C=C+1
5550 NEXT I
5560 IF C>1 THEN GOTO 5575
5565 LET US=US-1
5570 IF US<0 THEN LET US=0
5575 LET T(K)=1
5580 RETURN

```

Figura 16.11

LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

```

5600 LET M=J
5615 IF A#=" " THEN GOTO 5700
5620 GOSUB 5700
5625 LET M=ABS (L(K)-J)
5635 IF M<>18 THEN GOTO 5750
5640 LET T(K)=4
5645 LET M=L(K)
5650 LET M=M-16
5655 LET A#=U$(M TO M)
5660 IF A#=" " OR A#="*" THEN GO
TO 5650
5665 LET U$(M-1 TO M-1)=" "
5670 LET U$(M+1 TO M+1)=" "
5680 LET M=M+16
5690 LET A#=U$(M TO M)
5700 IF A#=" " OR A#="*" THEN GO
TO 5665
5710 LET U$(M+1 TO M+1)=" "
5720 LET U$(M-1 TO M-1)=" "
5730 RETURN
5750 IF M<>1 THEN GOTO 5850
5755 LET T(K)=3
5760 LET M=L(K)
5765 LET M=M-1
5770 LET A#=U$(M TO M)
5775 IF A#=" " OR A#="*" THEN GO
TO 5765
5780 LET U$(M-16 TO M-16)=" "
5785 LET U$(M+16 TO M+16)=" "
5790 LET M=M+1
5795 LET A#=U$(M TO M)
5800 IF A#=" " OR A#="*" THEN GO
TO 5780
5810 LET U$(M-16 TO M-16)=" "
5820 LET U$(M+16 TO M+16)=" "
5830 RETURN
5850 LET T(K)=2
5855 LET D(K)=L(K)-J
5860 LET I=L(K)
5870 LET U$(I-1 TO I-1)=" "
5875 LET U$(I+1 TO I+1)=" "
5880 LET U$(J-1 TO J-1)=" "
5885 LET U$(J+1 TO J+1)=" "
5890 LET U$(J-16 TO J-16)=" "
5895 LET U$(J+16 TO J+16)=" "
5900 LET U$(I-16 TO I-16)=" "
5910 LET U$(I+16 TO I+16)=" "
5920 LET M=2*J-I
5930 LET U$(M TO M)=" "
5940 LET M=2*I-J
5950 LET U$(M TO M)=" "
5960 RETURN

```

Figura 16.11

LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL) (cont.)

```

6000 LET C=INT ((D+1)/2)
6020 GOTO (6000+C*30)
6030 LET UD=UD-1
6040 IF UD<0 THEN LET UD=0
6050 RETURN
6060 LET UF=UF-1
6070 IF UF<0 THEN LET UF=0
6080 RETURN
6090 LET UC=UC-1
6100 IF UC<0 THEN LET UC=0
6110 RETURN
6500 LET M=INT (RND*224+17)
6530 LET JE=0
6535 RETURN
6540 LET M=M+1
6560 IF M<240 THEN RETURN
6570 LET M=18
6610 IF JE THEN GOTO ER
6620 LET JE=1
6630 RETURN
6700 IF UP>0 THEN GOTO 6780
6710 GOSUB 7000
6730 PRINT AT 21,0;"HA HA HA. GA
NHEI..."
6730 STOP
6740 RUN
6780 IF CP>0 THEN RETURN
6790 GOSUB 7000
6800 PRINT AT 21,0;"PERDI ESSA,
MAS PROXIMA ..."
6810 GOTO 6730
6820 GOSUB 7000
6830 PRINT AT 21,0;"DESISTIU HEI
M?"
6840 GOTO 6730
6900 PRINT AT 19,0;UC;" ";
UF;" ";OC;" ";CF;" ";
6920 PRINT AT 20,0;UA;" ";UD
;" ";US;" ";CA;" ";CD
;" ";CS;" ";
6990 RETURN
7000 FOR I=18 TO 239
7010 IF U$(I TO I)<" " THEN LET
U$(I TO I)=CHR$ (CODE (U$(I TO
)+128)
7020 IF C$(I TO I)<" " THEN LET
C$(I TO I)=CHR$ (CODE (C$(I TO
)+128)
7030 IF U$(I TO I)=" " OR U$(I T
O I)=" " THEN LET U$(I TO I)=CHR
$ (CODE (U$(I TO ))-128)

```

Figura 16.11

LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL) (cont.)

```

7040 IF C#(I TO I)=" " OR C#(I TO
C#(I TO I) THEN LET C#(I TO I)=CHR
CODE (C#(I TO I)+128)
7050 IF T#(I TO I)=" " AND C#(I
TO I)>"X" THEN LET C#(I TO I)="X"

7060 NEXT I
7070 GOSUB 600
7080 GOSUB 6900
7090 RETURN

8000 LET K1=1+AV
8010 LET K14=16-AV
8020 LET ER=3400
8030 FOR K=1 TO U
8040 GOSUB 6500
8050 LET H=M
8060 LET H=H
8061 GOSUB 6540
8065 IF T#(M-17 TO M-15)<>" "
THEN GOTO 8061
8066 IF T#(M-1 TO M+1)<>" " TH
EN GOTO 8061
8067 IF T#(M+15 TO M+17)<>" "
THEN GOTO 8061
8070 LET I=INT (M/16)
8080 LET N=M-I*16
8090 LET H=M
8100 LET M=I+1
8110 IF M<2 OR N<2 OR M>15 OR N>
15 THEN GOTO 8060
8130 IF NU>1 THEN GOTO 8170
8140 LET T#(H TO H)=" "
8150 GOTO 8980
8170 LET P0=0
8180 LET P1=0
8190 LET P2=0
8195 LET P3=0
8195 LET D=INT (RND*4)
8200 LET D=D+1-(D>=3)*4
8210 GOTO (8220+D*80)
8220 IF P0 THEN GOTO 8040
8230 LET P0=1
8240 LET D=0
8250 LET I1=M-1
8260 LET I2=M+NU
8270 LET J1=N-K1
8280 LET J2=N+K1
8290 IF I2<=15 THEN GOTO 8530
8300 IF P1 THEN GOTO 8040
8310 LET D=1
8320 LET P1=1

```

Figura 16.11 LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL) (cont.)

```

8330 LET I1=M-K1
8340 LET I2=M+K1
8350 LET J1=N-1
8360 LET J2=N+NU
8370 IF J2<=15 THEN GOTO 8530
8380 IF P2 THEN GOTO 8040
8390 LET P2=1
8400 LET D=2
8410 LET I1=M-NU
8420 LET I2=M+1
8430 LET J1=N-K1
8440 LET J2=N+K1
8450 IF I1>=2 THEN GOTO 8530
8460 IF P3 THEN GOTO 8060
8465 LET P3=1
8470 LET D=3
8480 LET I1=M-K1
8490 LET I2=M+K1

8500 LET J1=N-NU
8510 LET J2=N+1
8520 IF I1<=1 THEN GOTO 8020
8530 IF I1<2 THEN LET I1=2
8531 IF I1<2 THEN LET I1=2
8533 IF J1<2 THEN LET J1=2
8535 IF I2>15 THEN LET I2=15
8540 IF J2>15 THEN LET J2=15
8570 FOR I=I1 TO I2
8580 LET M1=I*16-16
8590 IF T#(M1+J1 TO M1+J2)<>P#(U
1 TO J2) THEN GOTO 8060
8610 NEXT I
8620 GOTO (8630+D*50)
8630 LET I1=M+AV
8640 LET I2=M+NU-1
8650 LET J1=N-AV
8660 LET J2=N+AV
8670 GOTO 8820
8680 LET I1=M-AV
8690 LET I2=M+AV
8700 LET J1=N+AV
8710 LET J2=N+NU-1
8720 GOTO 8820
8730 LET I1=M-NU+1
8740 LET I2=M-AV
8750 LET J1=N-AV
8760 LET J2=N+AV
8770 GOTO 8820
8780 LET I1=M-AV
8790 LET I2=M+AV
8800 LET J1=N-NU+1
8810 LET J2=N-AV

```

Figura 16.11 LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL) (cont.)

```

0000 IF I1<=1 THEN GOTO 0000
0000 IF J1<=1 THEN GOTO 0000
0000 IF J2>=16 THEN GOTO 0000
0000 IF I2>=16 THEN GOTO 0000
0000 IF AV<>0 THEN GOTO 0000
0000 FOR I=I1 TO I2
0000 LET M1=I*16-16
0000 LET T#(M1+J1 TO M1+J2)=X#(J
1 TO J2)
0000 NEXT I
0000 GOTO 0080
0000 LET M=(M-1)*16+N
0000 LET T#(M TO M)="X"
0000 LET M=(I1-1)*16+J1
0000 LET T#(M TO M)="X"
0000 LET M=(I2-1)*16+J2
0000 LET T#(M TO M)="X"
0000 NEXT K
0000 RETURN
0000 FAST
0000 CLEAR
0000 SAVE "NAVAL"
0000 RUN
0000 PRINT AT*21,0;" ";
0000 RETURN

```

Figura 16.11 LISTAGEM DO PROGRAMA - FONTE (BATALHA NAVAL) (cont.)

CAPÍTULO 17

Um Gerenciador de Tabela de Jogos

A FICHA

a) Tipo: Programa <input checked="" type="checkbox"/>	b) Nome: TOTÓ	c) Versão: 1	d) Revisão: Ø	e) Data: 22/10/83
Sub-rotina <input type="checkbox"/>	f) Objetivo: Jogo			
Função <input type="checkbox"/>				
g) Linguagem: BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z8Ø <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>			
NE Z8ØØØ <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>			
TK 8Ø <input type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
TK 82-C/83 <input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input type="checkbox"/>		
TK 85 <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
SINCLAIR ZX 8Ø <input checked="" type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>		
SINCLAIR ZX 81 <input type="checkbox"/>	CASSETE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1ØØØ <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-2ØØ <input checked="" type="checkbox"/>	5 1/4" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-3ØØ <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 3ØØ <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
RINGO R-47Ø e AS- 1ØØØ				
k) Observações:				

B MANUAL DE UTILIZAÇÃO

Este programa *não* é um jogo. Porém está relacionado a jogos. Trata-se de um construtor de tabelas e classificação de torneios clássicos. A forma de classificação é a normalmente utilizada em futebol.

Inicia perguntando quantas equipes. No mínimo 3, sem o que não seria necessário tabela de jogos. A seguir você deve entrar os nomes das equipes – no máximo 10 letras. Ao acabar de entrar todos os nomes, ele automaticamente sorteia os jogos e pergunta os resultados na forma:

equipe 1 x equipe 2

Entre o *resultado* do jogo iniciando pelo *número de pontos (gols)* da *equipe 1* seguido de NEWLINE. Depois, entre o da *equipe 2* seguido de NEWLINE. A tela se apagará por alguns segundos (o tempo depende do número de equipes) e volta com a classificação atual.

Depois de todos jogarem contra todos, é perguntado se deve haver retorno, isto é, se os jogos devem ser repetidos. Se você responder N, termina o programa. Se você responder S, todos os jogos serão repetidos, porém com *mando de campo* invertido em relação ao *turno*, isto é, se no turno houve o jogo

equipe a x equipe b

no retorno o jogo será

equipe b x equipe a

Se for torneio ao vivo – pequenos jogos familiares como o TOTÓ por exemplo –, as respostas devem ser entradas diretamente. Se o torneio puder ser interrompido, é aconselhável que a tabela seja transcrita e os resultados anotados para serem entrados todos de uma vez.

Na Figura 17.1 temos um exemplo completo com 4 equipes disputando um torneio de TOTÓ.

C MANUAL DE OPERAÇÃO

Cerca de 2 minutos são necessários para a carga a partir da fita cassete. No mínimo 4K é utilizado, dependendo da resposta ao número de equipes. Veja os Apêndices sobre o uso de fitas cassete.

QUANTAS EQUIPES? 1
 NO MINIMO TRES EQUIPES PARA HA-
 VER UM TORNEIO.
 QUANTAS EQUIPES? 4

ENTRE AGORA OS NOMES DAS EQUIPES
 UM A UM, SEGUIDOS DE NEWLINE.

- 1 - SAMYA
- 2 - MARJORIE
- 3 - DANIELLE
- 4 - ~~XXXXXXXXXX~~

VAMOS AGORA A TABELA:

DAREI O JOGO E VOCE ENTRA O
 ESCORE NA ORDEM.

1 - MARJORIE X DANIELLE
 -1 ??? NAO ENTENDI.

1 - MARJORIE X DANIELLE
 1 X -1 ??? ****

1 - MARJORIE X DANIELLE
 1 1

CLASSIFICACAO GERAL

EQUIPE	J	P	V	E	D	GP	GC
MARJORIE	1	1	0	1	0	1	1
DANIELLE	1	1	0	1	0	1	1
SAMYA	0	0	0	0	0	0	0
XXXXXXXXXX	0	0	0	0	0	0	0
2 - SAMYA			X	XXXXXXXXXX			
	2			1			

CLASSIFICACAO GERAL

EQUIPE	J	P	V	E	D	GP	GC
SAMYA	1	2	1	0	0	2	1
MARJORIE	1	1	0	1	0	1	1
DANIELLE	1	1	0	1	0	1	1
XXXXXXXXXX	1	0	0	0	1	1	0
3 - SAMYA				X	MARJORIE		
	2				1		

CLASSIFICACAO GERAL

EQUIPE	J	P	V	E	D	GP	GC
SAMYA	2	4	2	0	0	4	0
DANIELLE	1	1	0	1	0	1	1
MARJORIE	2	1	0	1	1	2	0
XXXXXXXXXX	1	0	0	0	1	1	0
4 - MARJORIE			X	XXXXXXXXXX			
	0			3			

CLASSIFICACAO GERAL

EQUIPE	J	P	V	E	D	GP	GC
SAMYA	0	4	2	0	0	4	0
XXXXXXXXXX	2	2	1	0	1	4	0
DANIELLE	1	1	0	1	0	1	1
MARJORIE	0	1	0	1	0	0	0
5 - SAMYA				X	DANIELLE		
	1				2		

CLASSIFICACAO GERAL

EQUIPE	J	P	V	E	D	GP	GC
SAMYA	0	4	2	0	1	0	4
DANIELLE	0	3	1	1	0	0	2
XXXXXXXXXX	2	2	1	0	1	4	0
MARJORIE	0	1	0	1	0	0	0
6 - DANIELLE			X	XXXXXXXXXX			
	2				3		

CLASSIFICACAO GERAL

EQUIPE	J	P	V	E	D	GP	GC
XXXXXXXXXX	0	4	2	0	1	7	4
SAMYA	0	4	2	0	1	0	4
DANIELLE	0	3	1	1	1	0	4
MARJORIE	0	1	0	1	0	0	0

Figura 17.1 EXEMPLO DE TORNEIO

Figura 17.1 EXEMPLO DE TORNEIO (cont.)

E ASSIM FOI O TURNO,
HAVERA RETORNO? (S/N)

1 - ~~MARJORIE~~ X DANIELLE
2 1

CLASSIFICACAO GERAL

EQUIPE	J	P	U	E	D	GP	GC
MARJORIE	4	0	0	0	4	0	0
SAMYA	0	4	0	0	4	0	0
DANIELLE	4	0	1	1	0	0	4
MARJORIE	0	1	0	1	0	0	0

2 - MARJORIE X SAMYA
2 2

CLASSIFICACAO GERAL

EQUIPE	J	P	U	E	D	GP	GC
MARJORIE	4	0	0	0	4	0	0
SAMYA	4	0	0	0	4	0	0
DANIELLE	4	0	1	1	0	0	4
MARJORIE	4	0	0	1	0	0	0

3 - ~~MARJORIE~~ X MARJORIE
1 1

CLASSIFICACAO GERAL

EQUIPE	J	P	U	E	D	GP	GC
MARJORIE	0	0	0	0	0	0	0
SAMYA	4	0	0	0	4	0	0
DANIELLE	4	0	1	1	0	0	4
MARJORIE	0	0	0	0	0	0	0

4 - DANIELLE X SAMYA
2 2

CLASSIFICACAO GERAL

EQUIPE	J	P	U	E	D	GP	GC
MARJORIE	0	0	0	0	0	0	0
SAMYA	0	0	0	0	0	0	0
DANIELLE	0	4	1	1	0	0	0
MARJORIE	0	0	0	0	0	0	0

5 - ~~MARJORIE~~ X SAMYA
3 1

CLASSIFICACAO GERAL

EQUIPE	J	P	U	E	D	GP	GC
MARJORIE	0	0	4	1	1	10	7
SAMYA	0	0	0	0	0	10	11
DANIELLE	0	4	1	1	0	0	0
MARJORIE	0	0	0	0	0	0	0

6 - DANIELLE X MARJORIE
1 2

CLASSIFICACAO GERAL

EQUIPE	J	P	U	E	D	GP	GC
MARJORIE	0	0	4	1	1	10	7
SAMYA	0	0	0	0	0	10	11
MARJORIE	0	0	1	0	0	7	10
DANIELLE	0	4	1	0	0	0	11

Figura 17.1 EXEMPLO DE TORNEIO (cont.)

D MANUAL TÉCNICO

Os dados são armazenados em vetores distintos

- R - tabela de jogos
- G - gols pró
- C - gols contra
- P - pontos - 2 por vitória

Figura 17.1 EXEMPLO DE TORNEIO (cont.)

- 1 por empate
- 0 por derrota

- V - vitórias
- E - empates
- D - derrotas
- N\$ - nomes das equipes
- O - classificação

Para classificação usa-se o muito conhecido *método da bolha* de forma indireta. É um método lento, porém, como o número de itens a ser classificado é pequeno, não chega a comprometer o tempo de resposta — o mais demorado é a impressão da classificação, pois a impressão de números no TK é naturalmente lenta.

Na Figura 17.2 encontra-se a listagem do programa-fonte.

```

10 REM TABELAO DO TOTO
20 REM ANTONIO JULIO LOSSIO BO
TELHO
25 FAST
30 PRINT AT 3,0;"QUANTAS EQUIP
ES?"
35 CLEAR
40 INPUT T
50 PRINT T
60 IF T>2 THEN GOTO 110
70 PRINT "NO MINIMO TRES EQUIP
ES PARA HA- VER UM TORNEIO."
80 PAUSE 300
90 CLS
100 RUN
110 LET Y=T*(T-1)/2
115 DIM R(Y,3)
120 DIM G(T)
125 DIM O(T)
130 DIM D(T)
140 DIM N$(T,10)
150 DIM P(T)
160 DIM U(T)
170 DIM V(T)
175 DIM E(T)
180 PRINT AT 20,0;"ENTRE AGORA
OS NOMES DAS EQUIPESUM A UM, SEG
UIDOS DE NEWLINE."
190 FOR I=1 TO T
195 LET O(I)=I
200 LET G(I)=0
210 LET C(I)=0

```

Figura 17.2

LISTAGEM DO PROGRAMA-FONTE (TOTO)

```

220 LET P(I)=0
230 LET U(I)=0
240 LET D(I)=0
250 LET E(I)=0
2600 INPUT N$(I)
270 SCROLL
280 PRINT AT 21,0;I;" - ";N$(I)
290 NEXT I
300 SCROLL
310 PRINT AT 21,0;"VAMOS AGORA
A TABELA:"
320 SCROLL
330 SCROLL
340 PRINT AT 20,0;"DAREI O JOGO
E VOCE ENTRA O ESCORE NA OR
DEM."
350 SCROLL
370 LET K=0
380 FOR I=1 TO T-1
390 FOR J=I+1 TO T
400 LET K=K+1
410 LET R(K,1)=I
420 LET R(K,2)=J
430 LET R(K,3)=0
440 NEXT J
450 NEXT I
460 FOR K=1 TO Y
465 IF K>Y*3/4 THEN GOTO 480
470 LET M=INT (RND*Y+1)
475 IF R(M,3)(<>0 THEN GOTO 470
478 GOTO 500
480 FOR M=1 TO Y
485 IF R(M,3)=0 THEN GOTO 500
490 NEXT M
500 LET I=R(M,1)
510 LET J=R(M,2)
520 LET R(M,3)=1
530 SCROLL
535 SCROLL
540 PRINT AT 21,0;(" " AND K<=9
);K;" - ";N$(I);" X ";N$(J)
550 INPUT I1
560 IF I1>=0 THEN GOTO 580
570 SCROLL
580 PRINT AT 21,0;I1;" ??? NAO
ENTENDI."
590 GOTO 530
600 INPUT J1
610 IF J1>=0 THEN GOTO 700
620 SCROLL

```

Figura 17.2

LISTAGEM DO PROGRAMA-FONTE (TOTO) (cont.)

```

630 PRINT AT 21,0;I1;" X ";J1;"
635 *****
640 GOTO 530
700 SCROLL
710 PRINT AT 21,5;I1;AT 21,18;J
1
720 LET G(I)=G(I)+I1
730 LET G(J)=G(J)+J1
740 LET C(I)=C(I)+J1
750 LET C(J)=C(J)+I1
760 IF I1<>J1 THEN GOTO 820
770 LET E(I)=E(I)+1
780 LET E(J)=E(J)+1
790 LET P(J)=P(J)+1
800 LET P(I)=P(I)+1
810 GOTO 900
820 IF I1>J1 THEN GOTO 870
830 LET V(J)=V(J)+1
840 LET D(I)=D(I)+1
850 LET P(J)=P(J)+2
860 GOTO 900
870 LET V(I)=V(I)+1
880 LET D(J)=D(J)+1
890 LET P(I)=P(I)+2
900 SCROLL
905 GOSUB 2000
910 SCROLL
915 PRINT AT 21,6;"CLASSIFICACA
O GERAL"
915 SCROLL
920 PRINT AT 21,0;"EQUIPE";AT 2
1,11;"J P U E D GP GC"
930 FOR H=1 TO T
935 LET L=O(H)
940 SCROLL
950 PRINT AT 21,0;N$(L);AT 21,1
1;V(L)+D(L)+E(L);AT 21,14;P(L);A
T 21,17;U(L);AT 21,20;E(L);AT 21
,23;D(L);AT 21,26;G(L);AT 21,29;
C(L);
955 LET O(H)=H
960 NEXT H
970 NEXT K
980 IF (V(1)+D(1)+E(1))>=T THEN
GOTO 1500
1000 SCROLL
1010 SCROLL
1020 SCROLL
1030 PRINT AT 20,0;"E ASSIM FOI
O TURNO. HAVERA RETUR
NO? (S/N)"

```

Figura 17.2

LISTAGEM DO PROGRAMA-FONTE (TOTO) (cont.)

```

1040 INPUT A$
1050 IF A$="N" THEN GOTO 1500
1060 IF A$="S" THEN GOTO 1200
1070 SCROLL
1080 SCROLL
1090 PRINT AT 20,0;"TECLE S SE V
OCE DESEJA QUE AS EQUIPES SE R
EFRENTEM DENOVO."
1100 GOTO 1040
1200 FOR H=1 TO Y
1210 LET R(H,3)=0
1220 LET U=R(H,1)
1230 LET R(H,1)=R(H,2)
1240 LET R(H,2)=U
1250 NEXT H
1260 GOTO 460
1300 STOP
1510 RUN
2000 LET Q=0
2010 FOR H=1 TO T-1
2020 LET U=O(H)
2030 LET S=O(H+1)
2040 IF P(U)>P(S) THEN GOTO 2500
2050 IF P(U)=P(S) THEN GOTO 2100
2060 LET Q=1
2070 LET O(H+1)=U
2080 LET O(H)=S
2090 GOTO 2500
2100 IF V(U)>V(S) THEN GOTO 2500
2110 IF V(U)<V(S) THEN GOTO 2050
2120 IF (G(U)-C(U))>(G(S)-C(S))
THEN GOTO 2500
2130 IF (G(U)-C(U))<(G(S)-C(S))
THEN GOTO 2050
2140 IF G(U)>G(S) THEN GOTO 2500
2150 IF G(U)<G(S) THEN GOTO 2050
2500 NEXT H
2510 IF Q<>0 THEN GOTO 2000
2520 RETURN
0000 FAST
0010 CLEAR
0020 SAVE "TOTO"
0030 GOTO 10

```

Figura 17.2

LISTAGEM DO PROGRAMA-FONTE (TOTO) (cont.)

APÊNDICE **A**

**Como Utilizar
os Programas
deste Livro**

Autor: Jorge da Cunha Pereira Filho

INTRODUÇÃO

Recomenda-se ao leitor só utilizar cada um dos programas deste livro após um estudo apurado, através da documentação incluída. Este procedimento evitará futuros dissabores.

Os programas deste livro estão apresentados na forma de documentação padronizada e completa, para permitir ao leitor uma fácil utilização do livro, tanto à primeira leitura como, em consultas posteriores, seja para simples aquisição de conhecimento, utilização efetiva dos programas, seja para sua adaptação ou modificação, ou para ampliar sua capacidade para convertê-los para uso em outro computador.

Além da documentação, incluída no texto do livro, estão disponíveis fitas cassete, que contêm os programas gravados, para que o leitor não tenha o trabalho de digitá-los no teclado, tarefa demorada, cansativa e sempre sujeita a erros e falhas.

As fitas cassete, contendo os programas gravados, podem vir junto com o livro ou serem adquiridas separadamente, opcionalmente, à escolha do leitor. A utilização de fitas cassete pré-gravadas poupa tempo e permite uma *imediata* utilização dos programas, assim que o leitor já os tenha estudado no livro. Basta colocá-los no gravador, fazer as ligações necessárias, carregar o programa desejado na memória do microcomputador e, *imediatamente*, iniciar sua execução, sem perda de tempo.

O livro se divide em capítulos e seções (pode haver subseções), por assunto. A documentação de cada programa se encontra dentro destes capítulos e seções, obedecendo ao assunto tratado. Portanto, para ter acesso a um programa, a procura deve ser feita por assunto, através dos títulos de capítulos e seções. Esta documentação obedece sempre ao mesmo formato, para que o leitor se familiarize com ela, e é dividida em quatro partes, como mostra a Figura A.1. Para cada programa são apresentados uma *Ficha* e três *manuals*: de Utilização, de Operação e Técnico. O *manual* de Utilização apresenta ainda duas partes: O Enunciado do Problema e a Descrição de Uso.

Em geral, todas as partes da documentação estarão presentes. Poderá ocorrer que procedimentos, operações e outras informações de caráter comum, isto é, servindo a vários ou todos os programas, ao invés de figurarem em uma determinada parte ou item, se encontrem apresentados, uma única vez, nos apêndices do livro. Este procedimento evita a repetição desnecessária da mesma descrição em vários pontos do livro. Neste caso, o leitor encontrará, sob a parte ou item respectivo, uma referência ao apêndice onde o assunto é apresentado.

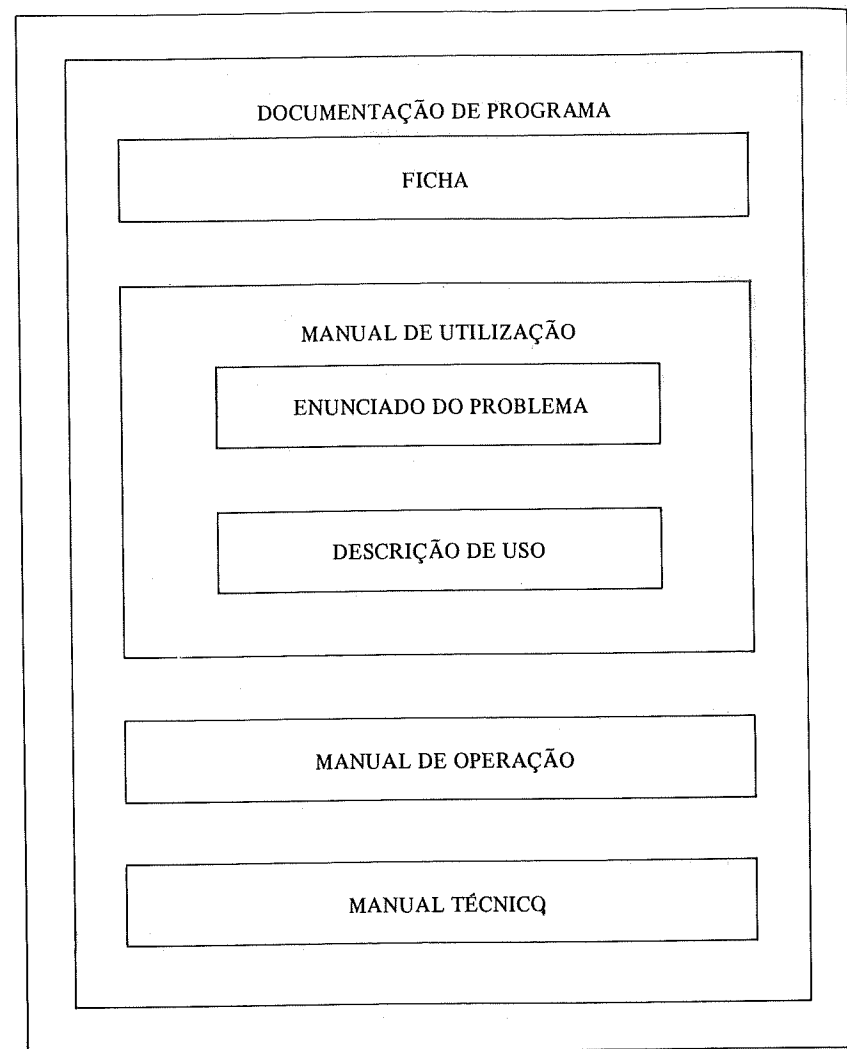


Figura A.1 ESQUEMA DA ORGANIZAÇÃO GERAL DA DOCUMENTAÇÃO DE PROGRAMA

FICHA

A *Ficha* descreve, de forma esquemática e sucinta, as principais características do programa como, por exemplo, seu objetivo, linguagem, compatibilidade com equipamentos, recursos em termos de memória e de periféricos etc. De

um relance, o leitor tem uma primeira visão global do programa, sem gastar tempo com a leitura de detalhes, nos manuais.

A Ficha é auto-explicativa e dispensa maiores comentários sobre o seu conteúdo e o seu formato se encontra na Figura A.2. Todavia, se se necessita de maiores informações sobre o programa, devem-se ler os *manuais*.

O leitor encontrará na Ficha marcados com um "X" os quadros à direita das características que forem pertinentes e os espaços em branco preenchidos com os dados adequados. As versões e revisões de um programa são numeradas de 1 (um) em diante. Não havendo revisão, o código é 0 (zero).

Por exemplo, a primeira versão de um programa, sem revisões, é versão 1 e revisão 0, ou, como se diz corretamente, 1.0 (um ponto zero).

a) Tipo: Programa <input type="checkbox"/> Sub-rotina <input type="checkbox"/> Função <input type="checkbox"/>	b) Nome: _____	c) Versão: _____	d) Revisão: _____	e) Data: ____/____/____
f) Objetivo: _____				
g) Linguagem: BASIC <input type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:		j) Instruções/ Comandos Especiais:
NE Z80 <input type="checkbox"/>		VIDEO <input type="checkbox"/>		
NE Z8000 <input type="checkbox"/>		TECLADO <input type="checkbox"/>		
TK 80 <input type="checkbox"/>		MEMÓRIA: 2K <input type="checkbox"/>		SLOW <input type="checkbox"/>
TK 82-C/83 <input type="checkbox"/>		16K <input type="checkbox"/>		FAST <input type="checkbox"/>
TK 85 <input type="checkbox"/>		48K <input type="checkbox"/>		LPRINT <input type="checkbox"/>
SINCLAIR ZX 80 <input type="checkbox"/>		64K <input type="checkbox"/>		LLIST <input type="checkbox"/>
SINCLAIR ZX 81 <input type="checkbox"/>		CASSETE <input type="checkbox"/>		COPY <input type="checkbox"/>
SINCLAIR SPECTRUM <input type="checkbox"/>		DISQUETE <input type="checkbox"/>		DLOAD <input type="checkbox"/>
TMEX 1000 <input type="checkbox"/>		DISCO: 8" <input type="checkbox"/>		DSAVE <input type="checkbox"/>
CP-200 <input type="checkbox"/>		5 1/4" <input type="checkbox"/>		DMLOAD <input type="checkbox"/>
CP-300 <input type="checkbox"/>		ALAVANCA ("JOYSTICK") <input type="checkbox"/>		DMSAVE <input type="checkbox"/>
APPLY 300 <input type="checkbox"/>		IMPRESSORA <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>
OUTROS (quais?) <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>		
k) Observações: _____				

Figura A.2 **FORMATO DA FICHA DE PROGRAMA**

MANUAIS

Cada um dos três *manuais* de programa tem um objetivo bem definido, ou seja, está orientado para um tipo de leitor: o usuário final, o operador de computador e o programador. São papéis diferentes os desempenhados por cada um destes personagens. Todavia, em microcomputadores, pelo menos dois papéis se confundem: o de operador de computador e o de usuário. Quer dizer que a mesma pessoa, o usuário das informações que serão processadas, é também quem opera o computador. Eventualmente, se a pessoa tem alguma inclinação para a programação, os três papéis podem se confundir: uma só pessoa é o usuário final, o operador e o programador. Daí a necessidade de dividir a documentação de um programa em três partes, destinadas, cada uma, a um destes personagens ou papéis (que podem ser representados pela mesma pessoa física). Os manuais, orientações e objetivos de cada manual estão sumariados no Quadro A.1.

MANUAL	ORIENTAÇÃO	OBJETIVO
de UTILIZAÇÃO	USUÁRIO FINAL DAS INFORMAÇÕES	Permitir realizar trabalho útil com o programa, fornecendo os dados a serem processados, dialogando com o programa e obtendo e analisando os resultados.
de OPERAÇÃO	OPERADOR DE COMPUTADOR	Permitir operar o microcomputador para execução e utilização de programas, realizando as operações de carga (colocar o programa na memória interna) a partir de fita cassete, como realizar o processamento e como salvar (armazenar) o programa e dados em fita cassete, além de outros procedimentos operacionais especiais.
TÉCNICO	PROGRAMADOR	Permitir o conhecimento interno do programa, como a natureza do problema a resolver, detalhes técnicos dos dados que entram e que saem e de como os dados são processados, do fluxo de dados (fluxograma) e do conjunto de instruções que constituem o programa (listagem na impressora).

Quadro A.1 **MANUAIS DE PROGRAMAS, ORIENTAÇÕES E OBJETIVOS**

Embora cada *manual* tenha uma orientação e se destine a um personagem diferente, não quer dizer que eles sejam estanques, da mesma maneira que os três papéis anteriormente descritos também possam ser desempenhados pela

mesma pessoa. As informações contidas nos três *manuals* se integram e complementam, representando, cada *manual*, apenas um ângulo de visão diferente do mesmo objeto, o programa. Por isso, é recomendável a leitura dos três *manuals*, embora com uma visão e grau de interesse diferentes para cada um ou suas partes. Por exemplo, o usuário final, a quem o Manual de Utilização se destina, encontrará neste um ponto de forte interesse, principalmente no Enunciado do Problema, não significando que o Manual Técnico, que contém a descrição da solução lhe interessa, mas apenas que esta última parte lhe interessa em menor grau, de uma forma global, para ter uma idéia geral do programa internamente, sem descer a detalhes. Um simples usuário final, se não tem que operar o computador, deverá todavia ter pelo menos um interesse superficial sobre os problemas operacionais, descritos no Manual de Operação. Como geralmente o usuário final é também o operador do computador, em se tratando de microcomputador, como operador deverá também tomar conhecimento detalhado do Manual de Operação. Tanto a operação como a utilização do programa requerem um bom nível de treinamento. Já o programador terá no Manual Técnico o seu maior interesse, mas não pode ignorar os demais manuais, isto é, os aspectos operacionais e utilitários, descritos, respectivamente, no Manual de Operação e no Manual de Utilização.

Cada *manual* tem em geral um formato próprio, para atender a sua orientação e objetivo específico, como se passa a descrever a seguir.

MANUAL DE UTILIZAÇÃO

O manual de utilização apresenta o enunciado do problema e a descrição de uso do programa, separadamente.

Dispensável esclarecer o que é o enunciado do problema.

A descrição de uso dá ao usuário as principais informações sobre como iniciar o processamento, a continuação normal e como terminar a execução do programa, além de outras informações que o permitam estimar o tempo de duração do processamento e o desempenho do programa.

O formato geral do manual de utilização é apresentado na Figura A.3, onde também se encontram, entre parênteses, sucintas explicações sobre o conteúdo de cada item.

Descrições de procedimentos comuns a diversos programas podem ser omitidas no manual de utilização, para que sejam descritas uma única vez, em apêndices, no final do livro. Haverá indicação de em que apêndice se encontram descritos os procedimentos omitidos no manual.

B – MANUAL DE UTILIZAÇÃO
a) ENUNCIADO DO PROBLEMA:
I) Descrição Geral: (enunciado geral do problema)
II) Entradas: (descrição qualitativa dos principais dados que entram)
III) Regras: (descrição de regras de processamento, métodos e roteiros de cálculo, algoritmos, a serem aplicados)
IV) Saídas: (descrição qualitativa dos principais dados que saem)
b) DESCRIÇÃO DE USO:
I) Preparação de Dados de Entrada: (instruções sobre como preparar dados de entrada, se for necessária uma preparação prévia)
II) Início: (informação sobre como iniciar a execução)
III) Continuação: (informação sobre o processamento normal, mensagens emitidas pelo programa, dados a serem fornecidos e resultados que devem ser separados)
IV) Término: (informações sobre como encerrar a execução do programa)
V) Análise de Resultados: (instruções sobre como analisar os resultados e seu significado)

Figura A.3

FORMATO DO MANUAL DE UTILIZAÇÃO DE PROGRAMA

MANUAL DE OPERAÇÃO

O manual de operação descreve como operar o programa, isto é, como instalá-lo, o processamento normal, os procedimentos em caso de anormalidade (exceções e erros) e como arquivar o programa (se for o caso).

O formato geral do manual de operação é apresentado na Figura A.4, onde também se encontram, entre parênteses, sucintas explicações sobre o conteúdo de cada item.

Como foi mencionado anteriormente, é possível que vários programas tenham procedimentos operacionais comuns, caso em que sua descrição não será incluída em cada manual de cada programa, mas estes procedimentos serão descritos unicamente uma vez, nos apêndices. Neste caso, a parte ou item do

C – MANUAL DE OPERAÇÃO
a) Instalação: (procedimentos e cuidados na instalação)
b) Processamento Normal: (como caracterizar o processamento normal e quais os procedimentos a serem seguidos)
c) Procedimento em Casos Anormais: (características de paradas e término anormais, mensagens e procedimentos a seguir nestes casos)
d) Arquivamento do Programa: (se o programa deve ser arquivado ao término do processamento, quais os procedimentos)
e) Outros Procedimentos: (outros procedimentos e observações são enquadráveis nos itens anteriores e julgados importantes)

Figura A.4 **FORMATO DO MANUAL DE OPERAÇÃO DE PROGRAMA**

manual indicará em que apêndice se encontra a descrição dos procedimentos omitidos no manual (para evitar repetições cansativas).

MANUAL TÉCNICO

O manual técnico apresenta a solução do problema, cujo enunciado já foi apresentado. O formato geral do manual técnico, onde estão também, entre parênteses, sucintas explicações sobre o conteúdo de cada item, encontra-se na Figura A.5.

D – MANUAL TÉCNICO
a) Abordagem: (texto descrevendo o enfoque usado na solução do problema)
b) Estruturas de Dados: (descrição da organização de telas e de arquivos [nomes e tipos de registros], das estruturas, como os registros, com a descrição de variáveis [nomes, características e funções])
c) Algoritmos: (descrição da solução lógica, através de pseudolinguagem ou fluxograma)
d) Programa-Fonte: (listagem da fonte, de preferência na impressora)
e) Restrições: (informações sobre o que o programa <i>não</i> resolve)
f) Diversos: (outras informações e observações julgadas pertinentes)

Figura A.5 **FORMATO DO MANUAL TÉCNICO DO PROGRAMA**

APÊNDICES

Pelo fato de muitos procedimentos operacionais e utilitários serem comuns ao uso de microcomputadores ou a grupos de programas, são descritos uma única vez, em forma de apêndices, no final do livro, como já se disse anteriormente.

Desde já, sugere-se ao leitor fazer uma primeira leitura destes apêndices, contendo informações gerais. Entre outros, o leitor encontrará mais informações nos apêndices relacionados no Quadro A.2.

APÊNDICE	TÍTULO
B	Carregamento de programas a partir de fitas cassete
C	Armazenamento de programas em fita cassete
D	Cuidados com gravadores e fitas cassete

Quadro A.2 **APÊNDICES COM INFORMAÇÕES DE PROCEDIMENTOS GERAIS**

APÊNDICE **B**

***Carregamento
de Programas
a partir de
Fitas Cassete***

Autor: Jorge da Cunha Pereira Filho

Para que seja feito o carregamento de programas a partir de fitas cassete na memória do microcomputador, tipo TK ou NE e similares, para execução, é preciso que:

1. O microcomputador esteja alimentado (conectado na fonte e esta conectada na tomada de eletricidade) e ligado (botão da fonte em ON);
2. O gravador esteja alimentado (conectado na tomada de corrente elétrica).

Obtidas as condições preliminares, descritas anteriormente, o procedimento geral é o seguinte:

1. Ligar o cabo (fio) adequado (que acompanha o equipamento) nos soquetes EAR (auricular) do gravador e do microcomputador;
2. Colocar os controles de *tonalidade* e *volume* do gravador na posição *adequada*. Estes controles têm uma graduação que vai de 0 (zero), correspondendo ao mínimo ou 0%, até 10 (dez), correspondendo ao máximo ou 100%. A posição *adequada* varia de gravador para gravador, com o tipo de fita e até mesmo com as condições ambientais (temperatura e umidade relativa do ar) e depende também de como a fita foi gravada. Em geral, tanto *tonalidade* como *volume* são usados com o mesmo nível. Por exemplo, se 70% (número 7 do controle) para *tonalidade*, então, 70% (número 7 do controle) para o *volume*, e assim sucessivamente. Em muitos gravadores, o nível *adequado* para o carregamento está em cerca de 80% (para *tonalidade* e *volume*), mas a *tonalidade* pode chegar a 100% (máximo do agudo). É preciso experimentar e determinar a posição *adequada* dos controles. Por exemplo, a leitura de programas em fita cassete *normal*, com gravadores NATIONAL modelo RQ-2222MA, só é conseguida após um cuidadoso ajuste dos controles geralmente entre 50% e 70%;
3. Se a fita cassete que contém o programa desejado não se encontra já colocada no gravador, fazê-lo;
4. Correr a fita para a posição inicial, calcando as teclas REWIND (retroceder) e STOP (parar), como for necessário, no caso de a fita já não se encontrar anteriormente colocada na posição de leitura (onde se encontra o início do programa desejado). Se o gravador tem *contador* (marcador de posição da fita), colocá-lo em zero (000). Avançar a fita, usando as teclas FORWARD (avançar) e STOP (parar), até a posição do contador, onde se encontra o programa desejado;
5. Para localizar exatamente o programa, procure ouvir a fita (desconectando *temporariamente* o pino do cabo que está ligado no soquete EAR e, se necessário, reduzindo o *volume*), calcando a tecla PLAY (tocar) e, uma vez encontrado o programa, a tecla STOP (parar). Inicialmente deve-se ouvir

um *zumbido agudo, suave e uniforme*, como um assobio. O início deste *zumbido* é o ponto procurado. Depois vem um trecho de *silêncio*, de 5 segundos. Imediatamente a seguir, vem um *zumbido forte, em tonalidades variadas*, correspondente ao *programa* gravado. Volte atrás, ao início do *zumbido agudo*, usando as teclas REWIND (retroceder) e STOP (parar). Se passar muito do ponto, pode usar, de novo, as teclas FORWARD (avançar), que dá um avanço rápido (sem som) e STOP (parar). Se estiver próximo e antes do ponto, use de novo a tecla PLAY (tocar), até ouvir de novo o início do *zumbido agudo* e calque a tecla STOP (parar), imediatamente. Restaure a ligação do cabo no soquete EAR e o *volume* adequado;

6. Com o cursor K na tela, digitar o comando de *carregar*, LOAD (tecla J), que deverá aparecer na tela, seguido entre aspas (""), do *nome do programa* (*nome* com que foi gravado anteriormente), que também deverão aparecer na tela. Se existe certeza de que, entre vários programas gravados, o desejado é o que se encontra nesse ponto, pode-se dispensar a digitação do nome do programa (mas não as aspas), bastando digitar apenas: **LOAD ""**. O uso do *nome do programa* com o comando **LOAD**, todavia, evita o carregamento do programa errado. Se o *nome* gravado na fita não confere com o *nome* no comando **LOAD**, o programa *não* será carregado. Além disso, a colocação do *nome do programa* no comando **LOAD** permite que, colocada a fita no *início*, seja feita uma busca do programa, até o final da fita (se necessário) até que seja encontrado, *automaticamente*, sem que se tenha que procurar, como descrito no item 5 acima. Embora cômodo para o usuário, este último método é demorado e pode levar muitos minutos, principalmente nos casos em que o programa procurado esteja no final da fita;
7. Colocar a fita em movimento, calcando a tecla PLAY (tocar) do gravador;
8. Imediatamente, teclar NEW LINE (ou ENTER) no teclado do microcomputador. O programa começará a ser carregado.

Durante o carregamento do programa, surgem na tela, inicialmente, *listras estreitas e inclinadas* correspondentes ao *zumbido agudo* e à zona de silêncio da gravação, e depois *listras largas, pretas e brancas, horizontais*, correspondentes estas ao código do programa. Pela nitidez da imagem, pode-se fazer uma avaliação de se os níveis de *tonalidade* e *volume* estão próximos do adequado e se o carregamento está sendo efetuado.

9. Terminado o carregamento, em geral, deverá aparecer na tela o código de reportagem 0/0. Se o programa foi projetado para iniciar a execução *automaticamente*, logo que carregado, surgirão na tela as primeiras

mensagens ou imagens do programa, assim que termina o carregamento. Se o programa *não* foi projetado para entrar em execução *automaticamente*, é preciso que sua execução seja iniciada pelo operador. Este início poderá ser feito, *de acordo com as instruções dos manuais do programa*, através de um comando **RUN** ou de um comando **GOTO**.

Muitas vezes, por um motivo qualquer, como mau contato nos conectores do cabo de ligação ou níveis inadequados dos controles de *tonalidade* e *volume*, ou sujeira do cabeçote de leitura/gravação, o carregamento falha. Ver também o anexo C, sobre armazenamento de programas em fitas cassete, e o anexo D, sobre cuidados com gravadores e fitas cassete.

Em caso de falha no carregamento, o sistema pode perder totalmente o controle ou conseguir um carregamento parcial ou defeituoso do programa. Se o sistema perder o controle, a tela ficará cinza e não aparecerá o cursor K. Neste caso, deve-se *desligar* (colocar o botão da fonte em OFF) e, novamente, *ligar* (colocar o botão da fonte em ON) o microcomputador, para que seja reinicializado. Se outro cursor surgir na tela, que *não* o cursor K, houve um carregamento defeituoso e o procedimento é o mesmo que para o caso anterior: *desligar e ligar* de novo. Se o cursor K aparecer na tela, após uma falha, pode ter ocorrido um carregamento parcial. Neste caso, deve-se limpar a memória, com o comando **NEW**, *antes* de tentar um novo carregamento do programa.

Caso as falhas ocorram com frequência, deve-se procurar verificar a causa. Entre as causas mais comuns, a sujeira do cabeçote de leitura/gravação requer atenção, mas podem haver outras causas. Maiores informações sobre cuidados com gravadores e fitas cassete se encontram no Apêndice D.

Procure se exercitar nestas operações, usando uma fita de *rascunho* e um programa de teste (qualquer programa BASIC correto serve).

APÊNDICE **C**

Armazenamento de Programas em Fitas Cassete

Autor: Jorge da Cunha Pereira Filho

Para que seja feito o armazenamento de programas em fitas cassete, a partir da memória interna do microcomputador, tipo TK e NE ou similar, é preciso que:

1. O microcomputador esteja alimentado (conectado na *fonte* e esta conectada na tomada elétrica) e ligado (botão da fonte em ON);
2. O programa a gravar esteja na memória principal no computador. Caso não esteja, deverá ser ou digitado ou carregado, previamente, a partir de outra fita cassete. Sobre carregamento de programas a partir de fitas cassete, ver o Apêndice B;
3. O gravador esteja alimentado (conectado na tomada de corrente elétrica).

Obtidas as condições preliminares anteriores, o procedimento, então, é o seguinte:

1. Ligar o cabo (fio) adequado (que acompanha o equipamento) nos soquetes MIC (microfone) do gravador e do microcomputador;
2. Colocar os controles de *tonalidade* e *volume* do gravador na posição adequada. Estes controles têm uma graduação de 0 (zero), correspondente ao mínimo ou 0%, até 10 (dez), correspondente ao máximo ou 100%. Se o gravador tem *controle automático de ganho*, o controle de volume é ignorado, porque o sistema se encarrega de controlar o nível adequado da gravação, mas o de *tonalidade* deve ser ajustado. Leia as instruções do manual do gravador que vai utilizar, sobre as características dele (do gravador). O nível *adequado* se situa entre 80% (número 8 no controle) e 100% (número 10 no controle), para os gravadores que não possuem controle automático de ganho. Na maioria dos casos, é preciso usar o nível de 100%. Como exemplo da variedade de características de gravadores cita-se o gravador NATIONAL modelo RQ-2222MA, em que, para armazenar um programa em uma fita *normal*, tanto a *tonalidade* como o *volume* serão controlados *automaticamente*;
3. Se a fita cassete a gravar não estiver no gravador, colocá-la;
4. Correr a fita para a posição *inicial* (calcando a tecla REWIND, para retroceder, e, para parar, a tecla STOP), se a fita não estiver anteriormente colocada na posição de gravação (onde será iniciado o armazenamento do programa). Se o gravador tem *contador* (marcador de posição da fita), colocá-lo a zero (000). Avançar até a posição da fita, pela leitura do contador, onde será gravado o programa, usando a tecla FORWARD para um avanço rápido e a tecla STOP para parar. Se passar da posição, use as teclas REWIND (retroceder) e STOP (parar), para voltar. Para avançar lentamente, as teclas PLAY (tocar) e STOP (parar) podem ser usadas (reduzindo o volume, temporariamente, se necessário);

5. Se o seu gravador *não* tem *contador*, você deve se certificar de que, no trecho em que pretende gravar o programa, não existem outros já gravados (se não quer perdê-los). A solução é ouvir a fita, usando os controles FORWARD (avançar), REWIND (retroceder), STOP (parar) e PLAY (tocar), do gravador, adequadamente, até ter certeza de que, por exemplo, está *além* do trecho em que foi gravado o último programa.
6. Com o cursor K na tela, digitar o comando para *salvar* (ou armazenar), **SAVE** (tecla S), que deverá aparecer na tela, seguido, entre aspas ("), do *nome do programa*, que também deverá aparecer na tela. Alternativamente, *de acordo com as instruções dos manuais do programa*, o programa poderá ou deverá ser armazenado por um comando **GOTO** seguido do número da instrução do programa onde se encontra a instrução **SAVE** (seguida do *nome do programa*, entre aspas) que provocará o armazenamento do programa. Quando o programa é armazenado desta forma, o posterior carregamento, por um comando **LOAD**, provocará não somente o carregamento do programa mas também a *imediate e automática* execução do programa carregado (sem que se tenha que usar os comandos **RUN** ou **GOTO**);
7. Colocar a fita em movimento, calcando, simultaneamente, as teclas PLAY (tocar) e RECORD (gravar) do gravador;
8. Imediatamente, teclar NEW LINE (ou ENTER), no teclado do microcomputador. O programa começará a ser armazenado na fita cassete.

Durante o armazenamento do programa em fita, inicialmente aparecerão *listras estreitas e inclinadas*. Imediatamente estas desaparecem e a tela fica *cinza*, correspondendo à gravação do *zumbido suave, agudo e uniforme*, bem como da zona de *silêncio*, de 5 segundos de duração, *antes* do programa. A seguir, aparecem na tela *listras largas, pretas e brancas, horizontais*, que correspondem ao código do *programa* que está sendo gravado. Pela imagem na tela pode-se avaliar se a gravação está sendo realizada, mas não há garantia disso, pois pode ter ocorrido uma falha qualquer;

9. Terminado o armazenamento do programa, em geral, aparecerá na tela o código de reportagem 0/0, se o armazenamento foi feito com um comando **SAVE**. Se o programa foi armazenado com um comando **GOTO**, como mencionado no item 6, acima, ele foi gravado por uma instrução **SAVE** do próprio programa, geralmente seguida de uma instrução **RUN**, o que provocará sua imediata execução após o armazenamento. Neste caso, surgirão na tela as primeiras mensagens e imagens enviadas pelo próprio programa, resultantes da sua execução.

Pode ocorrer uma falha no armazenamento do programa. Para verificar se o

programa foi gravado, volte ao ponto onde começou a gravação e ouça a fita, usando a tecla PLAY (tocar). Reduza o *volume*, se necessário. Deve-se ouvir o *zumbido suave, agudo e uniforme*, inicialmente, seguido de 5 segundos de *silêncio*, seguido do *zumbido forte e de tonalidades variadas*, correspondente ao *programa* gravado. Não significa, todavia, que a gravação tenha sido adequada. Para maior garantia, é aconselhável fazer mais de uma gravação do mesmo programa, em trechos diferentes da mesma fita (duas já é razoável) ou de fitas diferentes. Para evitar falhas de conexão de cabos de ligação, retire e coloque os pinos dos soquetes MIC, do gravador e do microcomputador, antes de uma nova gravação. Procure ouvir aquele *clíc* ao encaixar o pino no soquete.

Se quer ter certeza de que a gravação foi bem realizada, deverá apagar o programa da memória, com o comando **NEW**, e voltar a carregar o programa na memória, a partir das gravações realizadas em fitas cassete, como descrito no Apêndice B.

As causas de falhas na gravação podem ser de diversas origens, inclusive por sujeira no cabeçote de leitura/gravação. Para maiores informações sobre cuidados com gravadores e fitas cassete, ver o Apêndice D.

Se um programa não foi projetado para iniciar a execução automática logo após o armazenamento, é preciso iniciá-lo com um comando **RUN** ou um comando **GOTO**, de acordo com as instruções nos manuais do programa.

A gravação de programas pode ser feita em ambos os lados da fita cassete (A ou B). Existem fitas cassete com diversos tamanhos, sendo os mais comuns C-46, C-60 e C-120. A preferência deve recair na fita de tamanho médio, C-60. Deve-se dar um espaçamento entre programas, que evite sua superposição. O *contador* do gravador, em geral, tem uma pequena folga e, com o avanço e retrocesso da fita e paradas, dá pequenas diferenças na medição. Se o gravador *não* tem *contador*, a identificação de posição na fita se torna mais difícil. Neste caso, tem-se que recorrer à *escuta* da fita. Deve-se ouvir a fita, com o uso das teclas PLAY (tocar) e STOP (parar). Um procedimento que ajuda a identificar programas, em todos os casos, é, *antes de gravar* um programa, usar o microfone para gravar, à voz, a sua identificação (nome, data da gravação e finalidade; entre outros atributos). O *nome* do programa deve ser repetido, isto é, após gravar o nome por extenso, repetir o nome, *soletrado*. Este procedimento facilita a correta compreensão do nome, letra por letra. Para *carregar* o programa, fica mais fácil sua identificação, por escuta, quando se tem gravada, à voz, a sua identificação.

Procure se exercitar nestas operações, usando uma fita de *rascunho* e um programa de teste (qualquer programa BASIC correto serve).

APÊNDICE D

Cuidados com Gravadores e Fitas Cassete

Autor: Jorge da Cunha Pereira Filho

GRAVADORES

O gravador, no microcomputador, é um equipamento periférico, que tem um papel relevante no armazenamento de programas (e áreas de dados destes programas, como nos micros tipo TK e NE ou similares) e, dependendo do micro, também de dados. Por isso, merece especial atenção, sob pena de se perderem programas gravados ou por gravar, bem como os seus dados, muitas vezes já processados.

O gravador deve ser escolhido, se possível, obedecendo aos seguintes requisitos:

1. *Boa qualidade*, garantida por uma marca tradicional no mercado e, adicionalmente, *garantia* por um prazo longo. O comum é que o prazo de garantia seja de três meses, mas já existe algum gravador com garantia de 1 (um) ano;
2. Ter um *controle de ganho* (volume) *automático*, na gravação, para evitar um inadequado uso do controle manual do *volume*, isto é, um nível insatisfatório de gravação;
3. Dispor de um *contador* (marcador de posição da fita), para facilitar a localização de posições de leitura e gravação de programas.

Antes de adquirir um gravador, consulte outros usuários e também fornecedores (em geral, lojas de eletrodomésticos ou microcomputadores). Procure ler, *antes* de comprar, o manual do equipamento, para constatar quais as suas verdadeiras características. Todo o equipamento vem acompanhado do manual e do certificado de garantia. Verifique.

Depois de adquirir o gravador, utilize-o corretamente. Para isso, deve-se ler o *manual* que o acompanha, *antes* de utilizá-lo.

Quando em uso, se ocorrerem falhas de leitura ou gravação de programas atribuíveis ao gravador (que não sejam por defeito da fita cassete ou erro de operação), deve-se:

1. Limpar o cabeçote de leitura/gravação e as polias de tração da fita. Para tal, desalimentar o gravador (retirar o pino do cabo de alimentação do soquete do gravador ou da tomada de corrente elétrica). Em seguida, pressionar a tecla PLAY (tocar) para expor o cabeçote e as polias de tração (de borracha preta, em geral). Com um pedaço de algodão limpo (que também pode ser enrolado na ponta de um palito) ou um cotonete limpo, embebido em álcool comum, esfregar o cabeçote e as polias. Deixar secar ao ar livre (sem ser ao sol direto), em lugar seco, por um ou dois minutos. Soltar o mecanismo de leitura/gravação, pressionando a tecla STOP (parar). Esta limpeza pode ser necessária freqüentemente, até mesmo após a leitura

ou gravação de uma só fita, dependendo da sensibilidade do gravador. A falha em gravar (armazenar) ou ler (carregar) programas pode ser uma indicação de sujeira no cabeçote de leitura/gravação;

2. Se persistirem as falhas de leitura ou gravação, com freqüência, experimente trocar de tipo ou mesmo de fita cassete. Por exemplo, se está usando fita *normal*, passe para fita de *baixo ruído* (*low noise*) e vice-versa;
3. Se as falhas continuam a ocorrer com freqüência, mesmo com os procedimentos acima, procure uma oficina autorizada no conserto do gravador daquela marca (às vezes, do próprio fabricante), pois o defeito pode ser, por exemplo, de alinhamento do cabeçote (azimute) ou no mecanismo de tração, ou outro, que só pode ser corrigido em uma oficina;
4. Se o conserto por serviço autorizado não resolver o problema de falhas freqüentes, talvez seja aconselhável trocar de gravador. Um outro tipo ou marca de gravador pode ser a solução. Se possível, experimente um outro, por empréstimo, antes de comprar.

FITAS CASSETE

A escolha de fitas cassete é outro ponto importante para que se possa ter um nível de confiança no sistema de armazenamento de programas, reduzindo o risco de perder programas já gravados. Fitas de má qualidade resultam em gravações defeituosas, e em uma precoce desmagnetização.

A fita cassete deve ser escolhida, se possível, obedecendo aos seguintes requisitos:

1. *Boa qualidade*, garantida por uma marca tradicional no mercado;
2. Ser *normal* ou de *baixo ruído* (*low noise*). Não é necessário usar fitas com alto teor de dióxido de cromo. Em geral estas devem mesmo ser evitadas. Fitas com alto teor de dióxido de cromo, em geral, reduzem a vida útil do cabeçote de leitura/gravação, provocando um desgaste prematuro deste. A este respeito, ver instruções nos manuais do gravador que vai usar, sobre qual o tipo de fita recomendável para aquele aparelho.

Depois de gravadas, as fitas cassete devem ser guardadas em local seco, à temperatura ambiente e longe de objetos magnéticos. Não devem ser expostas ao sol nem serem submetidas a variações bruscas de temperatura. Muito cuidado para não colocar a fita cassete perto de objetos magnéticos, como ímãs, televisores, tesouras, alto-falantes, lâmpadas fluorescentes etc. O campo magnético destes dispositivos poderá desmagnetizar a fita cassete e os programas serão desgravados (apagados), causando um grande prejuízo.

