

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



ISBN 85-7001-201-2

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**

EDITORAS 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, 4<sup>a</sup> ed. – *J. C. Pereira Filho*  
BASIC PARA APLICAÇÕES COMERCIAIS – *D. Hergert*  
BASIC PARA MICROS PESSOAIS, 2<sup>a</sup> ed. – *J. C. Pereira Filho*  
BASIC SINCLAIR – *R. U. Christmann*  
BRINCANDO COM O COMPUTADOR – *J. A. Moreira*  
COBOL PARA ESTUDANTES, 4<sup>a</sup> 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, 2<sup>a</sup> 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, 2<sup>a</sup> 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**, 2<sup>a</sup> ed. – *M. Bohl*  
**IMPLANTAÇÃO DE MICROS E MINICOMPUTADORES COMERCIAIS** – *P. A. Knight*  
**INTRODUÇÃO À ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES** – *H. Lorin*  
**INTRODUÇÃO À CIÉNCIA DA COMPUTAÇÃO COM WATFIV E FORTRAN**, 2<sup>a</sup> ed. – *S. E. R. Carvalho*  
**INTRODUÇÃO À PROGRAMAÇÃO COM PASCAL**, 2<sup>a</sup> ed. – *S. E. R. Carvalho*  
**INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES**, 3<sup>a</sup> 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**, 2<sup>a</sup> ed. – *E. A. Garbin*  
**JCL/SISTEMA 370**, 3<sup>a</sup> ed. – *G. D. Brown*  
**LCP – LÓGICA DE CONSTRUÇÃO DE PROGRAMAS**, 2<sup>a</sup> 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**, 2<sup>a</sup> 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**, 4<sup>a</sup> ed. – *A. L. Furtado e C. S. Santos*  
**PRINCÍPIOS DE SISTEMAS OPERACIONAIS**, 3<sup>a</sup> 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**, 2<sup>a</sup> 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. McGinley*  
**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:  
**EDITORARIA 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

EDITORIA 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 teográfico: CAMPUSRIO

ISBN 85-7001-201-2

## 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.*

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

CDD — 001.6425  
001.6404

84-0700

# ***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 BRINCARDO 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 as 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	3'20,70	8.183	9.022
	VELHA	2'14,90	5.517	5.846
	PAR	1'14,42	3.028	3.121
	PAR-C	1'12,08	2.958	3.164
6	*TREK ICOOP	5'17,14 2'35,75	12.903 6.354	14.432 6.414
7	NAVAL TOTO	5'42,44 1'39,01	13.910 4.066	15.598 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 freqüência da parada código 5 (tela cheia). Torna-se pois muito freqüente 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	b) Nome: <u>GAMALETE</u>	c) Versão: <u>1</u>	d) Revisão: <u>0</u>	e) Data: <u>22/10/83</u>
Sub-rotina	f) Objetivo: <u>Jogo</u>			
Função				
g) Linguagem: PL/ <input type="checkbox"/> FORTRAN <input type="checkbox"/> COBOL <input type="checkbox"/> C <input type="checkbox"/>	BASIC <input checked="" type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/>			
h) Compatibilidade:	i) Recursos:	j) Instruções/ Comandos Especiais:		
NE Z80	VÍDEO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>		
NE Z8000	TECLADO <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>		
TK 80	MEMÓRIA: 2K <input type="checkbox"/>	LPRINT <input type="checkbox"/>		
TK 82-C/83	16K <input checked="" type="checkbox"/>	LLIST <input type="checkbox"/>		
TK 85	48K <input type="checkbox"/>	COPY <input type="checkbox"/>		
SINCLAIR ZX 80	64K <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
SINCLAIR ZX 81	CASSETTE <input checked="" type="checkbox"/>	DSAVE <input type="checkbox"/>		
SINCLAIR SPECTRUM	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>		
TIMEX 1000	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>		
CP-200	5½" <input type="checkbox"/>	DMLOAD <input type="checkbox"/>		
CP-300	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>		
APPLY 300	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>		
OUTROS (quais?) <u>RINGO R-470 e</u> <u>AS - 1000</u>	OUTROS (quais?) <input type="checkbox"/>			
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

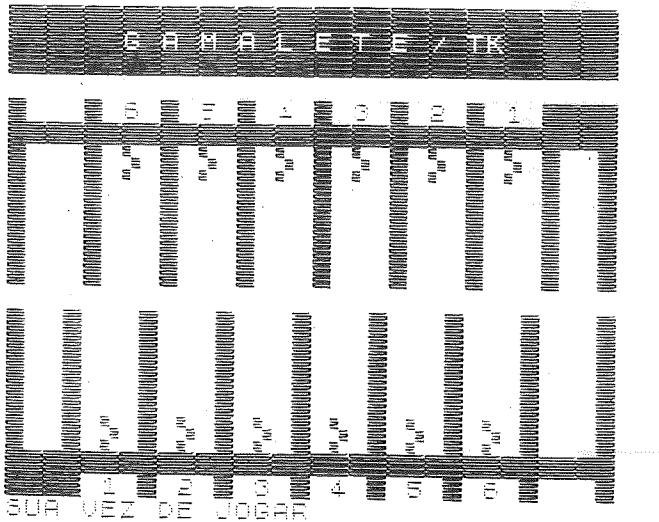
Junto a cada número temos *casas*. Nos ambientes X e Y temos *prisões*. Para começar, distribuem-se as pedras (que podem ser caroços de feijão ou milho) que serão colocadas nas *casas* (3 em cada). Cada jogador tem 6 *casas* numeradas de 1 a 6. Os jogadores se alternam para o início de cada partida.

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.

Tecle 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

Tecle 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

Tecle 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 movimenta 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 fizeram modificações na sequência 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 sequência. 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
2 DIM B(14)
3 LET BU=-1
4 LET AC=0
5 LET N=1
6 DIM G(14)
7 DIM F(30)
8 LET ME=0
9 LET TE=0
10 LET TU=0
11 LET RA=0
12 LET SLOW
13 CLS
14 PRINT AT 21,0;"QUEM COMEÇA?"
15 INPUT A$
16 LET N=N+1
17 LET BU=F(N)
18 LET AC=G(N)
19 LET N=N-1
20 LET BU=F(N)
21 LET AC=G(N)
22 LET N=N-1
23 LET BU=F(N)
24 LET AC=G(N)
25 LET N=N-1
26 LET BU=F(N)
27 LET AC=G(N)
28 LET N=N-1
29 LET BU=F(N)
30 LET AC=G(N)
31 LET N=N-1
32 LET BU=F(N)
33 LET AC=G(N)
34 LET N=N-1
35 LET BU=F(N)
36 LET AC=G(N)
37 LET N=N-1
38 LET BU=F(N)
39 LET AC=G(N)
40 LET N=N-1
41 LET BU=F(N)
42 LET AC=G(N)
43 LET N=N-1
44 LET BU=F(N)
45 LET AC=G(N)
46 LET N=N-1
47 LET BU=F(N)
48 LET AC=G(N)
49 LET N=N-1
50 LET BU=F(N)
51 LET AC=G(N)
52 LET N=N-1
53 LET BU=F(N)
54 LET AC=G(N)
55 LET N=N-1
56 LET BU=F(N)
57 LET AC=G(N)
58 LET N=N-1
59 LET BU=F(N)
60 LET AC=G(N)
61 LET N=N-1
62 LET BU=F(N)
63 LET AC=G(N)
64 LET N=N-1
65 LET BU=F(N)
66 LET AC=G(N)
67 LET N=N-1
68 LET BU=F(N)
69 LET AC=G(N)
70 LET N=N-1
71 LET BU=F(N)
72 LET AC=G(N)
73 LET N=N-1
74 LET BU=F(N)
75 LET AC=G(N)
76 LET N=N-1
77 LET BU=F(N)
78 LET AC=G(N)
79 LET N=N-1
80 LET BU=F(N)
81 LET AC=G(N)
82 LET N=N-1
83 LET BU=F(N)
84 LET AC=G(N)
85 LET N=N-1
86 LET BU=F(N)
87 LET AC=G(N)
88 LET N=N-1
89 LET BU=F(N)
90 LET AC=G(N)
91 LET N=N-1
92 LET BU=F(N)
93 LET AC=G(N)
94 LET N=N-1
95 LET BU=F(N)
96 LET AC=G(N)
97 LET N=N-1
98 LET BU=F(N)
99 LET AC=G(N)
100 LET N=N-1
101 LET BU=F(N)
102 LET AC=G(N)
103 LET N=N-1
104 LET BU=F(N)
105 LET AC=G(N)
106 LET N=N-1
107 LET BU=F(N)
108 LET AC=G(N)
109 LET N=N-1
110 LET BU=F(N)
111 LET AC=G(N)
112 LET N=N-1
113 LET BU=F(N)
114 LET AC=G(N)
115 LET N=N-1
116 LET BU=F(N)
117 LET AC=G(N)
118 LET N=N-1
119 LET BU=F(N)
120 LET AC=G(N)
121 LET N=N-1
122 LET BU=F(N)
123 LET AC=G(N)
124 LET N=N-1
125 LET BU=F(N)
126 LET AC=G(N)
127 LET N=N-1
128 LET BU=F(N)
129 LET AC=G(N)
130 LET N=N-1
131 LET BU=F(N)
132 LET AC=G(N)
133 LET N=N-1
134 LET BU=F(N)
135 LET AC=G(N)
136 LET N=N-1
137 LET BU=F(N)
138 LET AC=G(N)
139 LET N=N-1
140 LET BU=F(N)
141 LET AC=G(N)
142 LET N=N-1
143 LET BU=F(N)
144 LET AC=G(N)
145 LET N=N-1
146 LET BU=F(N)
147 LET AC=G(N)
148 LET N=N-1
149 LET BU=F(N)
150 LET AC=G(N)
151 LET N=N-1
152 LET BU=F(N)
153 LET AC=G(N)
154 LET N=N-1
155 LET BU=F(N)
156 LET AC=G(N)
157 LET N=N-1
158 LET BU=F(N)
159 LET AC=G(N)
160 LET N=N-1
161 LET BU=F(N)
162 LET AC=G(N)
163 LET N=N-1
164 LET BU=F(N)
165 LET AC=G(N)
166 LET N=N-1
167 LET BU=F(N)
168 LET AC=G(N)
169 LET N=N-1
170 LET BU=F(N)
171 LET AC=G(N)
172 LET N=N-1
173 LET BU=F(N)
174 LET AC=G(N)
175 LET N=N-1
176 LET BU=F(N)
177 LET AC=G(N)
178 LET N=N-1
179 LET BU=F(N)
180 LET AC=G(N)
181 LET N=N-1
182 LET BU=F(N)
183 LET AC=G(N)
184 LET N=N-1
185 LET BU=F(N)
186 LET AC=G(N)
187 LET N=N-1
188 LET BU=F(N)
189 LET AC=G(N)
190 LET N=N-1
191 LET BU=F(N)
192 LET AC=G(N)
193 LET N=N-1
194 LET BU=F(N)
195 LET AC=G(N)
196 LET N=N-1
197 LET BU=F(N)
198 LET AC=G(N)
199 LET N=N-1
200 LET BU=F(N)
201 LET AC=G(N)
202 LET N=N-1
203 LET BU=F(N)
204 LET AC=G(N)
205 LET N=N-1
206 LET BU=F(N)
207 LET AC=G(N)
208 LET N=N-1
209 LET BU=F(N)
210 LET AC=G(N)
211 LET N=N-1
212 LET BU=F(N)
213 LET AC=G(N)
214 LET N=N-1
215 LET BU=F(N)
216 LET AC=G(N)
217 LET N=N-1
218 LET BU=F(N)
219 LET AC=G(N)
220 LET N=N-1
221 LET BU=F(N)
222 LET AC=G(N)
223 LET N=N-1
224 LET BU=F(N)
225 LET AC=G(N)
226 LET N=N-1
227 LET BU=F(N)
228 LET AC=G(N)
229 LET N=N-1
230 LET BU=F(N)
231 LET AC=G(N)
232 LET N=N-1
233 LET BU=F(N)
234 LET AC=G(N)
235 LET N=N-1
236 LET BU=F(N)
237 LET AC=G(N)
238 LET N=N-1
239 LET BU=F(N)
240 LET AC=G(N)
241 LET N=N-1
242 LET BU=F(N)
243 LET AC=G(N)
244 LET N=N-1
245 LET BU=F(N)
246 LET AC=G(N)
247 LET N=N-1
248 LET BU=F(N)
249 LET AC=G(N)
250 LET N=N-1
251 LET BU=F(N)
252 LET AC=G(N)
253 LET N=N-1
254 LET BU=F(N)
255 LET AC=G(N)
256 LET N=N-1
257 LET BU=F(N)
258 LET AC=G(N)
259 LET N=N-1
260 LET BU=F(N)
261 LET AC=G(N)
262 LET N=N-1
263 LET BU=F(N)
264 LET AC=G(N)
265 LET N=N-1
266 LET BU=F(N)
267 LET AC=G(N)
268 LET N=N-1
269 LET BU=F(N)
270 LET AC=G(N)
271 LET N=N-1
272 LET BU=F(N)
273 LET AC=G(N)
274 LET N=N-1
275 LET BU=F(N)
276 LET AC=G(N)
277 LET N=N-1
278 LET BU=F(N)
279 LET AC=G(N)
280 LET N=N-1
281 LET BU=F(N)
282 LET AC=G(N)
283 LET N=N-1
284 LET BU=F(N)
285 LET AC=G(N)
286 LET N=N-1
287 LET BU=F(N)
288 LET AC=G(N)
289 LET N=N-1
290 LET BU=F(N)
291 LET AC=G(N)
292 LET N=N-1
293 LET BU=F(N)
294 LET AC=G(N)
295 LET N=N-1
296 LET BU=F(N)
297 LET AC=G(N)
298 LET N=N-1
299 LET BU=F(N)
300 LET AC=G(N)
301 LET N=N-1
302 LET BU=F(N)
303 LET AC=G(N)
304 LET N=N-1
305 LET BU=F(N)
306 LET AC=G(N)
307 LET N=N-1
308 LET BU=F(N)
309 LET AC=G(N)
310 LET N=N-1
311 LET BU=F(N)
312 LET AC=G(N)
313 LET N=N-1
314 LET BU=F(N)
315 LET AC=G(N)
316 LET N=N-1
317 LET BU=F(N)
318 LET AC=G(N)
319 LET N=N-1
320 LET BU=F(N)
321 LET AC=G(N)
322 LET N=N-1
323 LET BU=F(N)
324 LET AC=G(N)
325 LET N=N-1
326 LET BU=F(N)
327 LET AC=G(N)
328 LET N=N-1
329 LET BU=F(N)
330 LET AC=G(N)
331 LET N=N-1
332 LET BU=F(N)
333 LET AC=G(N)
334 LET N=N-1
335 LET BU=F(N)
336 LET AC=G(N)
337 LET N=N-1
338 LET BU=F(N)
339 LET AC=G(N)
340 LET N=N-1
341 LET BU=F(N)
342 LET AC=G(N)
343 LET N=N-1
344 LET BU=F(N)
345 LET AC=G(N)
346 LET N=N-1
347 LET BU=F(N)
348 LET AC=G(N)
349 LET N=N-1
350 LET BU=F(N)
351 LET AC=G(N)
352 LET N=N-1
353 LET BU=F(N)
354 LET AC=G(N)
355 LET N=N-1
356 LET BU=F(N)
357 LET AC=G(N)
358 LET N=N-1
359 LET BU=F(N)
360 LET AC=G(N)
361 LET N=N-1
362 LET BU=F(N)
363 LET AC=G(N)
364 LET N=N-1
365 LET BU=F(N)
366 LET AC=G(N)
367 LET N=N-1
368 LET BU=F(N)
369 LET AC=G(N)
370 LET N=N-1
371 LET BU=F(N)
372 LET AC=G(N)
373 LET N=N-1
374 LET BU=F(N)
375 LET AC=G(N)
376 LET N=N-1
377 LET BU=F(N)
378 LET AC=G(N)
379 LET N=N-1
380 LET BU=F(N)
381 LET AC=G(N)
382 LET N=N-1
383 LET BU=F(N)
384 LET AC=G(N)
385 LET N=N-1
386 LET BU=F(N)
387 LET AC=G(N)
388 LET N=N-1
389 LET BU=F(N)
390 LET AC=G(N)
391 LET N=N-1
392 LET BU=F(N)
393 LET AC=G(N)
394 LET N=N-1
395 LET BU=F(N)
396 LET AC=G(N)
397 LET N=N-1
398 LET BU=F(N)
399 LET AC=G(N)
400 LET N=N-1
401 LET BU=F(N)
402 LET AC=G(N)
403 LET N=N-1
404 LET BU=F(N)
405 LET AC=G(N)
406 LET N=N-1
407 LET BU=F(N)
408 LET AC=G(N)
409 LET N=N-1
410 LET BU=F(N)
411 LET AC=G(N)
412 LET N=N-1
413 LET BU=F(N)
414 LET AC=G(N)
415 LET N=N-1
416 LET BU=F(N)
417 LET AC=G(N)
418 LET N=N-1
419 LET BU=F(N)
420 LET AC=G(N)
421 LET N=N-1
422 LET BU=F(N)
423 LET AC=G(N)
424 LET N=N-1
425 LET BU=F(N)
426 LET AC=G(N)
427 LET N=N-1
428 LET BU=F(N)
429 LET AC=G(N)
430 LET N=N-1
431 LET BU=F(N)
432 LET AC=G(N)
433 LET N=N-1
434 LET BU=F(N)
435 LET AC=G(N)
436 LET N=N-1
437 LET BU=F(N)
438 LET AC=G(N)
439 LET N=N-1
440 LET BU=F(N)
441 LET AC=G(N)
442 LET N=N-1
443 LET BU=F(N)
444 LET AC=G(N)
445 LET N=N-1
446 LET BU=F(N)
447 LET AC=G(N)
448 LET N=N-1
449 LET BU=F(N)
450 LET AC=G(N)
451 LET N=N-1
452 LET BU=F(N)
453 LET AC=G(N)
454 LET N=N-1
455 LET BU=F(N)
456 LET AC=G(N)
457 LET N=N-1
458 LET BU=F(N)
459 LET AC=G(N)
460 LET N=N-1
461 LET BU=F(N)
462 LET AC=G(N)
463 LET N=N-1
464 LET BU=F(N)
465 LET AC=G(N)
466 LET N=N-1
467 LET BU=F(N)
468 LET AC=G(N)
469 LET N=N-1
470 LET BU=F(N)
471 LET AC=G(N)
472 LET N=N-1
473 LET BU=F(N)
474 LET AC=G(N)
475 LET N=N-1
476 LET BU=F(N)
477 LET AC=G(N)
478 LET N=N-1
479 LET BU=F(N)
480 LET AC=G(N)
481 LET N=N-1
482 LET BU=F(N)
483 LET AC=G(N)
484 LET N=N-1
485 LET BU=F(N)
486 LET AC=G(N)
487 LET N=N-1
488 LET BU=F(N)
489 LET AC=G(N)
490 LET N=N-1
491 LET BU=F(N)
492 LET AC=G(N)
493 LET N=N-1
494 LET BU=F(N)
495 LET AC=G(N)
496 LET N=N-1
497 LET BU=F(N)
498 LET AC=G(N)
499 LET N=N-1
500 LET BU=F(N)
501 LET AC=G(N)
502 LET N=N-1
503 LET BU=F(N)
504 LET AC=G(N)
505 LET N=N-1
506 LET BU=F(N)
507 LET AC=G(N)
508 LET N=N-1
509 LET BU=F(N)
510 LET AC=G(N)
511 LET N=N-1
512 LET BU=F(N)
513 LET AC=G(N)
514 LET N=N-1
515 LET BU=F(N)
516 LET AC=G(N)
517 LET N=N-1
518 LET BU=F(N)
519 LET AC=G(N)
520 LET N=N-1
521 LET BU=F(N)
522 LET AC=G(N)
523 LET N=N-1
524 LET BU=F(N)
525 LET AC=G(N)
526 LET N=N-1
527 LET BU=F(N)
528 LET AC=G(N)
529 LET N=N-1
530 LET BU=F(N)
531 LET AC=G(N)
532 LET N=N-1
533 LET BU=F(N)
534 LET AC=G(N)
535 LET N=N-1
536 LET BU=F(N)
537 LET AC=G(N)
538 LET N=N-1
539 LET BU=F(N)
540 LET AC=G(N)
541 LET N=N-1
542 LET BU=F(N)
543 LET AC=G(N)
544 LET N=N-1
545 LET BU=F(N)
546 LET AC=G(N)
547 LET N=N-1
548 LET BU=F(N)
549 LET AC=G(N)
550 LET N=N-1
551 LET BU=F(N)
552 LET AC=G(N)
553 LET N=N-1
554 LET BU=F(N)
555 LET AC=G(N)
556 LET N=N-1
557 LET BU=F(N)
558 LET AC=G(N)
559 LET N=N-1
560 LET BU=F(N)
561 LET AC=G(N)
562 LET N=N-1
563 LET BU=F(N)
564 LET AC=G(N)
565 LET N=N-1
566 LET BU=F(N)
567 LET AC=G(N)
568 LET N=N-1
569 LET BU=F(N)
570 LET AC=G(N)
571 LET N=N-1
572 LET BU=F(N)
573 LET AC=G(N)
574 LET N=N-1
575 LET BU=F(N)
576 LET AC=G(N)
577 LET N=N-1
578 LET BU=F(N)
579 LET AC=G(N)
580 LET N=N-1
581 LET BU=F(N)
582 LET AC=G(N)
583 LET N=N-1
584 LET BU=F(N)
585 LET AC=G(N)
586 LET N=N-1
587 LET BU=F(N)
588 LET AC=G(N)
589 LET N=N-1
590 LET BU=F(N)
591 LET AC=G(N)
592 LET N=N-1
593 LET BU=F(N)
594 LET AC=G(N)
595 LET N=N-1
596 LET BU=F(N)
597 LET AC=G(N)
598 LET N=N-1
599 LET BU=F(N)
600 LET AC=G(N)
601 LET N=N-1
602 LET BU=F(N)
603 LET AC=G(N)
604 LET N=N-1
605 LET BU=F(N)
606 LET AC=G(N)
607 LET N=N-1
608 LET BU=F(N)
609 LET AC=G(N)
610 LET N=N-1
611 LET BU=F(N)
612 LET AC=G(N)
613 LET N=N-1
614 LET BU=F(N)
615 LET AC=G(N)
616 LET N=N-1
617 LET BU=F(N)
618 LET AC=G(N)
619 LET N=N-1
620 LET BU=F(N)
621 LET AC=G(N)
622 LET N=N-1
623 LET BU=F(N)
624 LET AC=G(N)
625 LET N=N-1
626 LET BU=F(N)
627 LET AC=G(N)
628 LET N=N-1
629 LET BU=F(N)
630 LET AC=G(N)
631 LET N=N-1
632 LET BU=F(N)
633 LET AC=G(N)
634 LET N=N-1
635 LET BU=F(N)
636 LET AC=G(N)
637 LET N=N-1
638 LET BU=F(N)
639 LET AC=G(N)
640 LET N=N-1
641 LET BU=F(N)
642 LET AC=G(N)
643 LET N=N-1
644 LET BU=F(N)
645 LET AC=G(N)
646 LET N=N-1
647 LET BU=F(N)
648 LET AC=G(N)
649 LET N=N-1
650 LET BU=F(N)
651 LET AC=G(N)
652 LET N=N-1
653 LET BU=F(N)
654 LET AC=G(N)
655 LET N=N-1
656 LET BU=F(N)
657 LET AC=G(N)
658 LET N=N-1
659 LET BU=F(N)
660 LET AC=G(N)
661 LET N=N-1
662 LET BU=F(N)
663 LET AC=G(N)
664 LET N=N-1
665 LET BU=F(N)
666 LET AC=G(N)
667 LET N=N-1
668 LET BU=F(N)
669 LET AC=G(N)
670 LET N=N-1
671 LET BU=F(N)
672 LET AC=G(N)
673 LET N=N-1
674 LET BU=F(N)
675 LET AC=G(N)
676 LET N=N-1
677 LET BU=F(N)
678 LET AC=G(N)
679 LET N=N-1
680 LET BU=F(N)
681 LET AC=G(N)
682 LET N=N-1
683 LET BU=F(N)
684 LET AC=G(N)
685 LET N=N-1
686 LET BU=F(N)
687 LET AC=G(N)
688 LET N=N-1
689 LET BU=F(N)
690 LET AC=G(N)
691 LET N=N-1
692 LET BU=F(N)
693 LET AC=G(N)
694 LET N=N-1
695 LET BU=F(N)
696 LET AC=G(N)
697 LET N=N-1
698 LET BU=F(N)
699 LET AC=G(N)
700 LET N=N-1
701 LET BU=F(N)
702 LET AC=G(N)
703 LET N=N-1
704 LET BU=F(N)
705 LET AC=G(N)
706 LET N=N-1
707 LET BU=F(N)
708 LET AC=G(N)
709 LET N=N-1
710 LET BU=F(N)
711 LET AC=G(N)
712 LET N=N-1
713 LET BU=F(N)
714 LET AC=G(N)
715 LET N=N-1
716 LET BU=F(N)
717 LET AC=G(N)
718 LET N=N-1
719 LET BU=F(N)
720 LET AC=G(N)
721 LET N=N-1
722 LET BU=F(N)
723 LET AC=G(N)
724 LET N=N-1
725 LET BU=F(N)
726 LET AC=G(N)
727 LET N=N-1
728 LET BU=F(N)
729 LET AC=G(N)
730 LET N=N-1
731 LET BU=F(N)
732 LET AC=G(N)
733 LET N=N-1
734 LET BU=F(N)
735 LET AC=G(N)
736 LET N=N-1
737 LET BU=F(N)
738 LET AC=G(N)
739 LET N=N-1
740 LET BU=F(N)
741 LET AC=G(N)
742 LET N=N-1
743 LET BU=F(N)
744 LET AC=G(N)
745 LET N=N-1
746 LET BU=F(N)
747 LET AC=G(N)
748 LET N=N-1
749 LET BU=F(N)
750 LET AC=G(N)
751 LET N=N-1
752 LET BU=F(N)
753 LET AC=G(N)
754 LET N=N-1
755 LET BU=F(N)
756 LET AC=G(N)
757 LET N=N-1
758 LET BU=F(N)
759 LET AC=G(N)
760 LET N=N-1
761 LET BU=F(N)
762 LET AC=G(N)
763 LET N=N-1
764 LET BU=F(N)
765 LET AC=G(N)
766 LET N=N-1
767 LET BU=F(N)
768 LET AC=G(N)
769 LET N=N-1
770 LET BU=F(N)
771 LET AC=G(N)
772 LET N=N-1
773 LET BU=F(N)
774 LET AC=G(N)
775 LET N=N-1
776 LET BU=F(N)
777 LET AC=G(N)
778 LET N=N-1
779 LET BU=F(N)
780 LET AC=G(N)
781 LET N=N-1
782 LET BU=F(N)
783 LET AC=G(N)
784 LET N=N-1
785 LET BU=F(N)
786 LET AC=G(N)
787 LET N=N-1
788 LET BU=F(N)
789 LET AC=G(N)
790 LET N=N-1
791 LET BU=F(N)
792 LET AC=G(N)
793 LET N=N-1
794 LET BU=F(N)
795 LET AC=G(N)
796 LET N=N-1
797 LET BU=F(N)
798 LET AC=G(N)
799 LET N=N-1
800 LET BU=F(N)
801 LET AC=G(N)
802 LET N=N-1
803 LET BU=F(N)
804 LET AC=G(N)
805 LET N=N-1
806 LET BU=F(N)
807 LET AC=G(N)
808 LET N=N-1
809 LET BU=F(N)
810 LET AC=G(N)
811 LET N=N-1
812 LET BU=F(N)
813 LET AC=G(N)
814 LET N=N-1
815 LET BU=F(N)
816 LET AC=G(N)
817 LET N=N-1
818 LET BU=F(N)
819 LET AC=G(N)
820 LET N=N-1
821 LET BU=F(N)
822 LET AC=G(N)
823 LET N=N-1
824 LET BU=F(N)
825 LET AC=G(N)
826 LET N=N-1
827 LET BU=F(N)
828 LET AC=G(N)
829 LET N=N-1
830 LET BU=F(N)
831 LET AC=G(N)
832 LET N=N-1
833 LET BU=F(N)
834 LET AC=G(N)
835 LET N=N-1
836 LET BU=F(N)
837 LET AC=G(N)
838 LET N=N-1
839 LET BU=F(N)
840 LET AC=G(N)
841 LET N=N-1
842 LET BU=F(N)
843 LET AC=G(N)
844 LET N=N-1
845 LET BU=F(N)
846 LET AC=G(N)
847 LET N=N-1
848 LET BU=F(N)
849 LET AC=G(N)
850 LET N=N-1
851 LET BU=F(N)
852 LET AC=G(N)
853 LET N=N-1
854 LET BU=F(N)
855 LET AC=G(N)
856 LET N=N-1
857 LET BU=F(N)
858 LET AC=G(N)
859 LET N=N-1
860 LET BU=F(N)
861 LET AC=G(N)
862 LET N=N-1
863 LET BU=F(N)
864 LET AC=G(N)
865 LET N=N-1
866 LET BU=F(N)
867 LET AC=G(N)
868 LET N=N-1
869 LET BU=F(N)
870 LET AC=G(N)
871 LET N=N-1
872 LET BU=F(N)
873 LET AC=G(N)
874 LET N=N-1
875 LET BU=F(N)
876 LET AC=G(N)
877 LET N=N-1
878 LET BU=F(N)
879 LET AC=G(N)
880 LET N=N-1
881 LET BU=F(N)
882 LET AC=G(N)
883 LET N=N-1
884 LET BU=F(N)
885 LET AC=G(N)
886 LET N=N-1
887 LET BU=F(N)
888 LET AC=G(N)
889 LET N=N-1
890 LET BU=F(N)
891 LET AC=G(N)
892 LET N=N-1
893 LET BU=F(N)
894 LET AC=G(N)
895 LET N=N-1
896 LET BU=F(N)
897 LET AC=G(N)
898 LET N=N-1
899 LET BU=F(N)
900 LET AC=G(N)
901 LET N=N-1
902 LET BU=F(N)
903 LET AC=G(N)
904 LET N=N-1
905 LET BU=F(N)
906 LET AC=G(N)
907 LET N=N-1
908 LET BU=F(N)
909 LET AC=G(N)
910 LET N=N-1
911 LET BU=F(N)
912 LET AC=G(N)
913 LET N=N-1
914 LET BU=F(N)
915 LET AC=G(N)
916 LET N=N-1
917 LET BU=F(N)
918 LET AC=G(N)
919 LET N=N-1
920 LET BU=F(N)
921 LET AC=G(N)
922 LET N=N-1
923 LET BU=F(N)
924 LET AC=G(N)
925 LET N=N-1
926 LET BU=F(N)
927 LET AC=G(N)
928 LET N=N-1
929 LET BU=F(N)
930 LET AC=G(N)
931 LET N=N-1
932 LET BU=F(N)
933 LET AC=G(N)
934 LET N=N-1
935 LET BU=F(N)
936 LET AC=G(N)
937 LET N=N-1
938 LET BU=F(N)
939 LET AC=G(N)
940 LET N=N-1
941 LET BU=F(N)
942 LET AC=G(N)
943 LET N=N-1
944 LET BU=F(N)
945 LET AC=G(N)
946 LET N=N-1
947 LET BU=F(N)
948 LET AC=G(N)
949 LET N=N-1
950 LET BU=F(N)
951 LET AC=G(N)
952 LET N=N-1
953 LET BU=F(N)
954 LET AC=G(N)
955 LET N=N-1
956 LET BU=F(N)
957 LET AC=G(N)
958 LET N=N-1
959 LET BU=F(N)
960 LET AC=G(N)
961 LET N=N-1
962 LET BU=F(N)
963 LET AC=G(N)
964 LET N=N-1
965 LET BU=F(N)
966 LET AC=G(N)
967 LET N=N-1
968 LET BU=F(N)
969 LET AC=G(N)
970 LET N=N-1
971 LET BU=F(N)
972 LET AC=G(N)
973 LET N=N-1
974 LET BU=F(N)
975 LET AC=G(N)
976 LET N=N-1
977 LET BU=F(N)
978 LET AC=G(N)
979 LET N=N-1
980 LET BU=F(N)
981 LET AC=G(N)
982 LET N=N-1
983 LET BU=F(N)
984 LET AC=G(N)
985 LET N=N-1
986 LET BU=F(N)
987 LET AC=G(N)
988 LET N=N-1
989 LET BU=F(N)
990 LET AC=G(N)
991 LET N=N-1
992 LET BU=F(N)
993 LET AC=G(N)
994 LET N=N-1
995 LET BU=F(N)
996 LET AC=G(N)
997 LET N=N-1
998 LET BU=F(N)
999 LET AC=G(N)
1000 LET N=N-1
1001 LET BU=F(N)
1002 LET AC=G(N)
1003 LET N=N-1
1004 LET BU=F(N)
1005 LET AC=G(N)
1006 LET N=N-1
1007 LET BU=F(N)
1008 LET AC=G(N)
1009 LET N=N-1
1010 LET BU=F(N)
1011 LET AC=G(N)
1012 LET N=N-1
1013 LET BU=F(N)
1014 LET AC=G(N)
1015 LET N=N-1
1016 LET BU=F(N)
1017 LET AC=G(N)
1018 LET N=N-1
1019 LET BU=F(N)
1020 LET AC=G(N)
1021 LET N=N-1
1022 LET BU=F(N)
1023 LET AC=G(N)
1024 LET N=N-1
1025 LET BU=F(N)
1026 LET AC=G(N)
1027 LET N=N-1
1028 LET BU=F(N)
1029 LET AC=G(N)
1030 LET N=N-1
1031 LET BU=F(N)
1032 LET AC=G(N)
1033 LET N=N-1
1034 LET BU=F(N)
1035 LET AC=G(N)
1036 LET N=N-1
1037 LET BU=F(N)
1038 LET AC=G(N)
1039 LET N=N-1
1040 LET BU=F(N)
1041 LET AC=G(N)
1042 LET N=N-1
1043 LET BU=F(N)
1044 LET AC=G(N)
1045 LET N=N-1
1046 LET BU=F(N)
1047 LET AC=G(N)
1048 LET N=N-1
1049 LET BU=F(N)
1050 LET AC=G(N)
1051 LET N=N-1
1052 LET BU=F(N)
1053 LET AC=G(N)
1054 LET N=N-1
1055 LET BU=F(N)
1056 LET AC=G(N)
1057 LET N=N-1
1058 LET BU=F(N)
1059 LET AC=G(N)
1060 LET N=N-1
1061 LET BU=F(N)
1062 LET AC=G(N)
1063 LET N=N-1
1064 LET BU=F(N)
1065 LET AC=G(N)
1066 LET N=N-1
1067 LET BU=F(N)
1068 LET AC=G(N)
1069 LET N=N-1
1070 LET BU=F(N)
1071 LET AC=G(N)
1072 LET N=N-1
1073 LET BU=F(N)
1074 LET AC=G(N)
1075 LET N=N-1
1076 LET BU=F(N)
1077 LET AC=G(N)
1078 LET N=N-1
1079 LET BU=F(N)
1080 LET AC=G(N)
1081 LET N=N-1
1082 LET BU=F(N)
1083 LET AC=G(N)
1084 LET N=N-1
1085 LET BU=F(N)
1086 LET AC=G(N)
1087 LET N=N-1
1088 LET BU=F(N)
1089 LET AC=G(N)
1090 LET N=N-1
1091 LET BU=F(N)
1092 LET AC=G(N)
1093 LET N=N-1
1094 LET BU=F(N)
1095 LET AC=G(N)
1096 LET N=N-1
1097 LET BU=F(N)
1098 LET AC=G(N)
1099 LET N=N-1
1100 LET BU=F(N)
1101 LET AC=G(N)
1102 LET N=N-1
1103 LET BU=F(N)
1104 LET AC=G(N)
1105 LET N=N-1
1106 LET BU=F(N)
1107 LET AC=G(N)
1108 LET N=N-1
1109 LET BU=F(N)
1110 LET AC=G(N)
1111 LET N=N-1
1112 LET BU=F(N)
1113 LET AC=G(N)
1114 LET N=N-1
1115 LET BU=F(N)
1116 LET AC=G(N)
1117 LET N=N-1
1118 LET BU=F(N)
1119 LET AC=G(N)
1120 LET N=N-1
1121 LET BU=F(N)
1122 LET AC=G(N)
1123 LET N=N-1
1124 LET BU=F(N)
1125 LET AC=G(N)
1126 LET N=N-1
1127 LET BU=F(N)
1128 LET AC=G(N)
1129 LET N=N-1
1130 LET BU=F(N)
1131 LET AC=G(N)
1132 LET N=N-1
1133 LET BU=F(N)
1134 LET AC=G(N)
1135 LET N=N-1
1136 LET BU=F(N)
1137 LET AC=G(N)
1138 LET N=N-1
1139 LET BU=F(N)
1140 LET AC=G(N)
1141 LET N=N-1
1142 LET BU=F(N)
1143 LET AC=G(N)
1144 LET N=N-1
1145 LET BU=F(N)
1146 LET AC=G(N)
1147 LET N=N-1
1148 LET BU=F(N)
1149 LET AC=G(N)
1150 LET N=N-1
1151 LET BU=F(N)
1152 LET AC=G(N)
1153 LET N=N-1
1154 LET BU=F(N)
1155 LET AC=G(N)
1156 LET N=N-1
1157 LET BU=F(N)
1158 LET AC=G(N)
1159 LET N=N-1
1160 LET BU=F(N)
1161 LET AC=G(N)
1162 LET N=N-1
1163 LET BU=F(N)
1164 LET AC=G(N)
1165 LET N=N-1
1166 LET BU=F(N)
1167 LET AC=G(N)
1168 LET N=N-1
1169 LET BU=F(N)
1170 LET AC=G(N)
1171 LET N=N-1
1172 LET BU=F(N)
1173 LET AC=G(N)
1174 LET N=N-1
1175 LET BU=F(N)
1176 LET AC=G(N)
1177 LET N=N-1
1178 LET BU=F(N)
1179 LET AC=G(N)
1180 LET N=N-1
1181 LET BU=F(N)
1182 LET AC=G(N)
1183 LET N=N-1
1184 LET BU=F(N)
1185 LET AC=G(N)
1186 LET N=N-1
1187 LET BU=F(N)
1188 LET AC=G(N)
1189 LET N=N-1
1190 LET BU=F(N)
1191 LET AC=G(N)
1192 LET N=N-1
1193 LET BU=F(N)
1194 LET AC=G(N)
1195 LET N=N-1
1196 LET BU=F(N)
1197 LET AC=G(N)
1198 LET N=N-1
1199 LET BU=F(N)
1200 LET AC=G(N)
1201 LET N=N-1
1202 LET BU=F(N)
1203 LET AC=G(N)
1204 LET N=N-1
1205 LET BU=F(N)
1206 LET AC=G(N)
1207 LET N=N-1
1208 LET BU=F(N)
1209 LET AC=G(N)
1210 LET N=N-1
1211 LET BU=F(N)
1212 LET AC=G(N)
1213 LET N=N-1
1214 LET BU=F(N)
1215 LET AC=G(N)
1216 LET N=N-1
1217 LET BU=F(N)
1218 LET AC=G(N)
1219 LET N=N-1
1220 LET BU=F(N)
1221 LET AC=G(N)
1222 LET N=N-1
1223 LET BU=F(N)
1224 LET AC=G(N)
1225 LET N=N-1
1226 LET BU=F(N)
1227 LET AC=G(N)
122
```

**Figura 2.3** LISTAGEM DO P

**Figura 2.3** LISTAGEM DO PROGRAMA - FONTE (GAMALETE) (cont.)

```

0430 NEXT I
0460 SCROLL
0490 SCROLL
0520 PRINT AT 21,0;"SEU GANHEI"
0550 ME," PARTIDAS"
0580 SCROLL
0610 PRINT AT 21,0;"VOCÊ GANHOU"
0640 TU," PARTIDAS"
0650 SCROLL
0660 IF AC=0 THEN GOTO 2700
0670 PRINT AT 21,0;"FICAMOS COM"
0680 SALDO DE "
0690 SCROLL
0700 IF AC<0 THEN GOTO 2600
0710 PRINT AT 21,0;AC;" A SEU FA
COR"
0720 GOTO 2610
0730 PRINT AT 21,0;"EMPATAMOS EM"
0740 NUMERO DE PONTOS"
0750 GOTO 2610
0760 PRINT AT 21,0;-AC;" A MEU F
ACOR"
0770 STOP
0780 RUN
4000 PRINT AT 21,0; ""
4010 RETURN

6000 REM FORMATAÇÃO DA TELA
6010 CLS
6020 PRINT AT 0,0;P#
6030 PRINT AT 1,0;[REDACTED]
6040 PRINT AT 2,0;P#
6050 PRINT AT 3,0;[REDACTED]
6060 PRINT AT 4,0;[REDACTED] ■ 6 ■ 5 ■
6070 PRINT AT 5,0;P#
6080 PRINT AT 6,0;[REDACTED]
6090 FOR I=6 TO 11
6100 PRINT AT I,0;M#
6110 NEXT I
6120 FOR I=10 TO 16
6130 PRINT AT I,0;N#
6140 NEXT I
6150 PRINT AT 19,0;P#
6160 PRINT AT 20,0;[REDACTED] 1 ■ 0 ■
6170 FOR X=1 TO 14
6180 GOSUB 7900
6190 NEXT X
6200 RETURN
6600 REM TESTA MOVIMENTO
6100 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 W=19-Y TO 16
7830 PRINT AT W,CC;" "
7840 NEXT W
7850 RETURN
7860 FOR W=6 TO 5+Y
7870 PRINT AT W,CC;" "
7880 NEXT W
7890 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;""
7990 NEXT W
7995 IF Y<2<(B(X)) THEN PRINT AT 6
    +Y,CC;""
7999 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 6000
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=R+1
8190 GOTO 8160
8200 IF B(L)<>0 THEN GOTO 8300
8210 IF L=7 OR L=14 THEN GOTO 83
8220 LET R=B(14-L)+R
8300 IF R>Q THEN LET Q=R
8400 NEXT I
8410 LET Q=B(14)-B(7)-Q
8420 IF C>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))
    *EUN)*2
8470 NEXT I
8700 FOR I=1 TO 14

```

```

8710 LET S(I)=G(I)
8720 NEXT I
8730 IF Q<D THEN GOTO 8800
8740 LET A=J
8750 LET D=0
8800 NEXT J
8805 BLOW
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**

## FICHA

a) Tipo: Programa	b) Nome: <u>VIDA</u>	c) Versão: <u>1</u>	d) Revisão: <u>Ø</u>	e) Data: <u>22/10/83</u>																																									
Sub-rotina	f) Objetivo: <u>Jogo</u>																																												
Função																																													
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:	<table border="1"> <tr> <td>i) Recursos:</td> <td>j) Instruções/ Comandos Especiais:</td> </tr> <tr> <td>NE Z8Ø</td> <td>VÍDEO <input checked="" type="checkbox"/></td> <td>SLOW <input checked="" type="checkbox"/></td> </tr> <tr> <td>NE Z8ØØØ</td> <td>TECLADO <input checked="" type="checkbox"/></td> <td>FAST <input checked="" type="checkbox"/></td> </tr> <tr> <td>TK 8Ø</td> <td>MEMÓRIA: 2K <input type="checkbox"/></td> <td>LPRINT <input type="checkbox"/></td> </tr> <tr> <td>TK 82-C/83</td> <td>16K <input checked="" type="checkbox"/></td> <td>LLIST <input type="checkbox"/></td> </tr> <tr> <td>TK 85</td> <td>48K <input type="checkbox"/></td> <td>COPY <input type="checkbox"/></td> </tr> <tr> <td>SINCLAIR ZX 8Ø</td> <td>64K <input type="checkbox"/></td> <td>DLOAD <input type="checkbox"/></td> </tr> <tr> <td>SINCLAIR ZX 81</td> <td>CASSETE <input checked="" type="checkbox"/></td> <td>DSAVE <input type="checkbox"/></td> </tr> <tr> <td>SINCLAIR SPECTRUM</td> <td>DISQUETE <input checked="" type="checkbox"/></td> <td>DMLOAD <input type="checkbox"/></td> </tr> <tr> <td>TIMEX 1000</td> <td>DISCO: 8" <input checked="" type="checkbox"/></td> <td>DMSAVE <input type="checkbox"/></td> </tr> <tr> <td>CP-20Ø</td> <td>5½" <input type="checkbox"/></td> <td>OUTROS (quais?) <input type="checkbox"/></td> </tr> <tr> <td>CP-30Ø</td> <td>ALAVANCA ("JOYSTICK") <input type="checkbox"/></td> <td></td> </tr> <tr> <td>APPLY 30Ø</td> <td>IMPRESSORA <input type="checkbox"/></td> <td></td> </tr> <tr> <td>OUTROS (quais?)</td> <td>OUTROS (quais?) <input type="checkbox"/></td> <td></td> </tr> </table>				i) Recursos:	j) Instruções/ Comandos Especiais:	NE Z8Ø	VÍDEO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>	NE Z8ØØØ	TECLADO <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>	TK 8Ø	MEMÓRIA: 2K <input type="checkbox"/>	LPRINT <input type="checkbox"/>	TK 82-C/83	16K <input checked="" type="checkbox"/>	LLIST <input type="checkbox"/>	TK 85	48K <input type="checkbox"/>	COPY <input type="checkbox"/>	SINCLAIR ZX 8Ø	64K <input type="checkbox"/>	DLOAD <input type="checkbox"/>	SINCLAIR ZX 81	CASSETE <input checked="" type="checkbox"/>	DSAVE <input type="checkbox"/>	SINCLAIR SPECTRUM	DISQUETE <input checked="" type="checkbox"/>	DMLOAD <input type="checkbox"/>	TIMEX 1000	DISCO: 8" <input checked="" type="checkbox"/>	DMSAVE <input type="checkbox"/>	CP-20Ø	5½" <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>	CP-30Ø	ALAVANCA ("JOYSTICK") <input type="checkbox"/>		APPLY 30Ø	IMPRESSORA <input type="checkbox"/>		OUTROS (quais?)	OUTROS (quais?) <input type="checkbox"/>	
i) Recursos:	j) Instruções/ Comandos Especiais:																																												
NE Z8Ø	VÍDEO <input checked="" type="checkbox"/>	SLOW <input checked="" type="checkbox"/>																																											
NE Z8ØØØ	TECLADO <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>																																											
TK 8Ø	MEMÓRIA: 2K <input type="checkbox"/>	LPRINT <input type="checkbox"/>																																											
TK 82-C/83	16K <input checked="" type="checkbox"/>	LLIST <input type="checkbox"/>																																											
TK 85	48K <input type="checkbox"/>	COPY <input type="checkbox"/>																																											
SINCLAIR ZX 8Ø	64K <input type="checkbox"/>	DLOAD <input type="checkbox"/>																																											
SINCLAIR ZX 81	CASSETE <input checked="" type="checkbox"/>	DSAVE <input type="checkbox"/>																																											
SINCLAIR SPECTRUM	DISQUETE <input checked="" type="checkbox"/>	DMLOAD <input type="checkbox"/>																																											
TIMEX 1000	DISCO: 8" <input checked="" type="checkbox"/>	DMSAVE <input type="checkbox"/>																																											
CP-20Ø	5½" <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>																																											
CP-30Ø	ALAVANCA ("JOYSTICK") <input type="checkbox"/>																																												
APPLY 30Ø	IMPRESSORA <input type="checkbox"/>																																												
OUTROS (quais?)	OUTROS (quais?) <input type="checkbox"/>																																												
<u>RINGO R-47Ø e</u> <u>AS - 1ØØØ</u>																																													
k) Observações:	<hr/> <hr/> <hr/>																																												

**B**

## MANUAL DE UTILIZAÇÃO

Após o **RUN NEWLINE** ou **LOAD " " NEWLINE** o programa pede que seja teclada a letra **I** para iniciar. Se você deseja modificar algum parâmetro, tecle **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:

- a) Não deve existir uma distribuição inicial de organismos tal que possa se provar que a população possa crescer sem limites;
- b) Deve existir pelo menos uma distribuição inicial de organismos tal que se tenha a impressão de desenvolvimento contínuo;
- c) 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:

- 1) **Ambiente** – Os organismos estão distribuídos em um plano reticulado;
- 2) **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;
- 3) **Nascimento** – Um organismo nasce em um ponto vazio do tabuleiro se e somente se existirem exatamente três organismos na vizinhança desse ponto;
- 4) **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 influí 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.

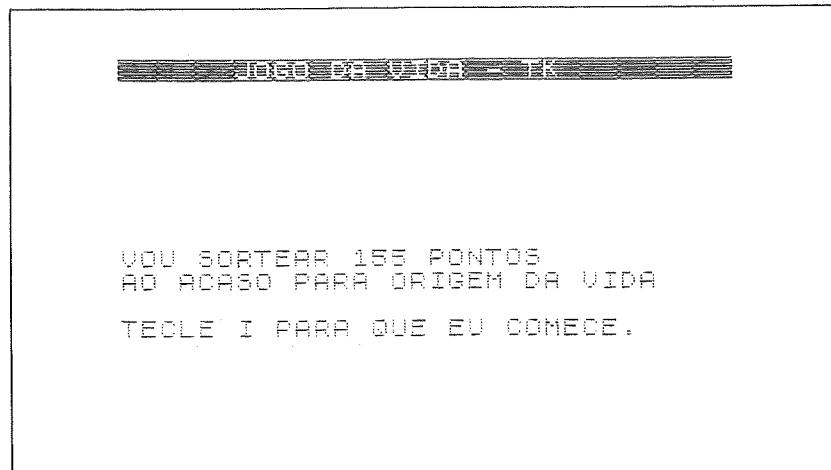


Figura 3.1  
TELA INICIAL

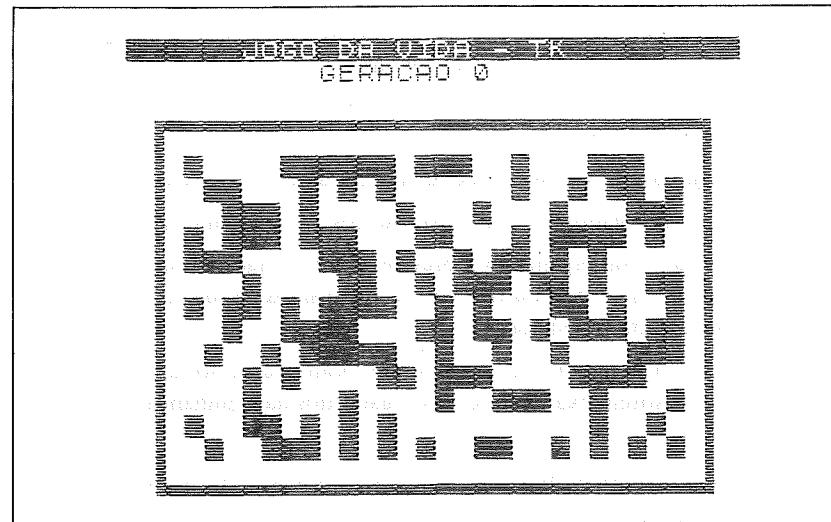


Figura 3.2  
EXEMPLO

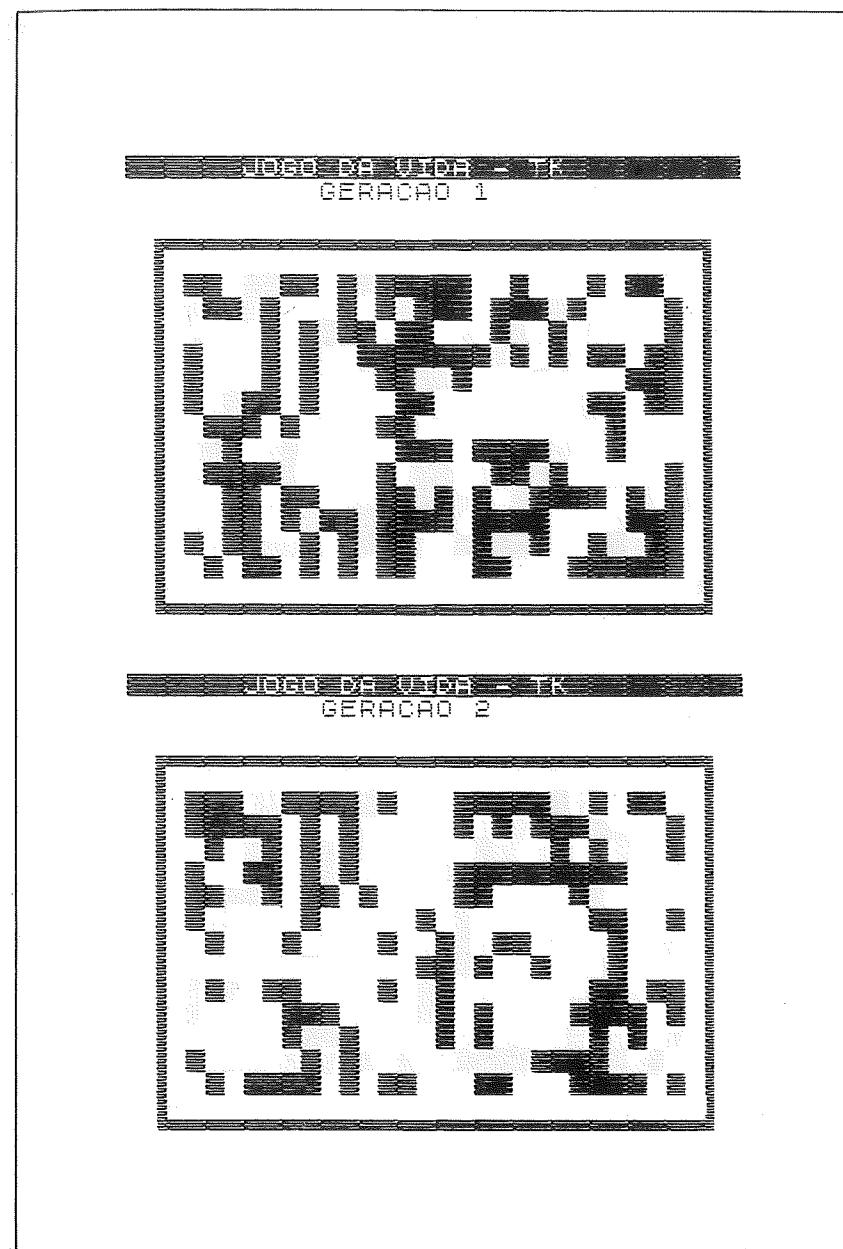


Figura 3.2  
EXEMPLO (cont.)

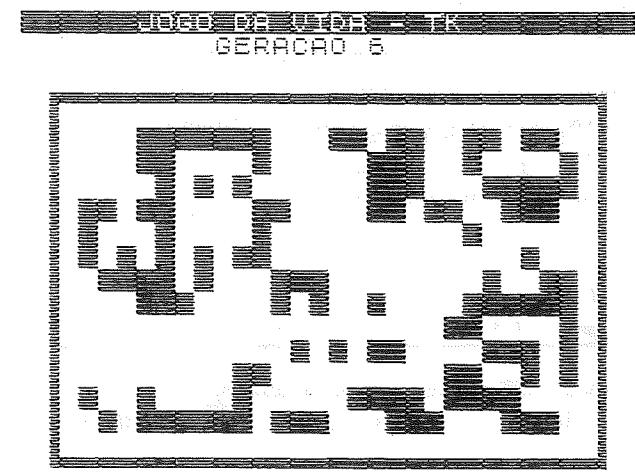
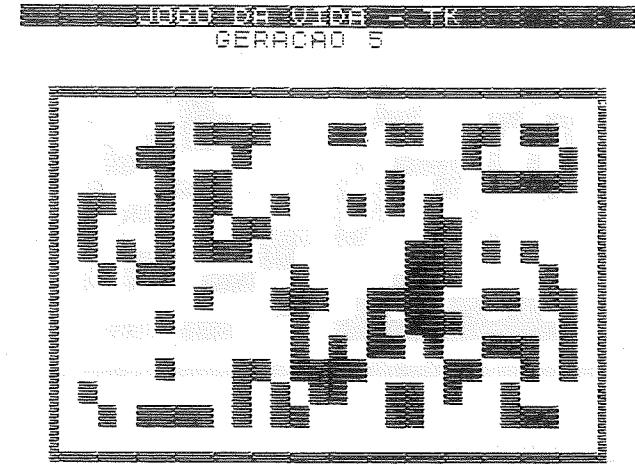
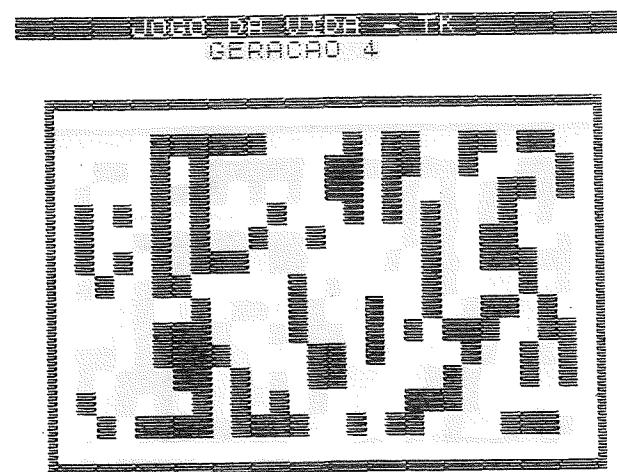
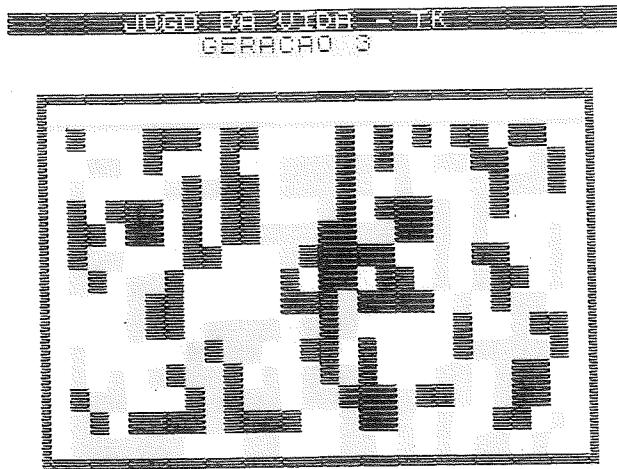


Figura 3.2  
EXEMPLO (cont.)

Figura 3.2  
EXEMPLO (cont.)

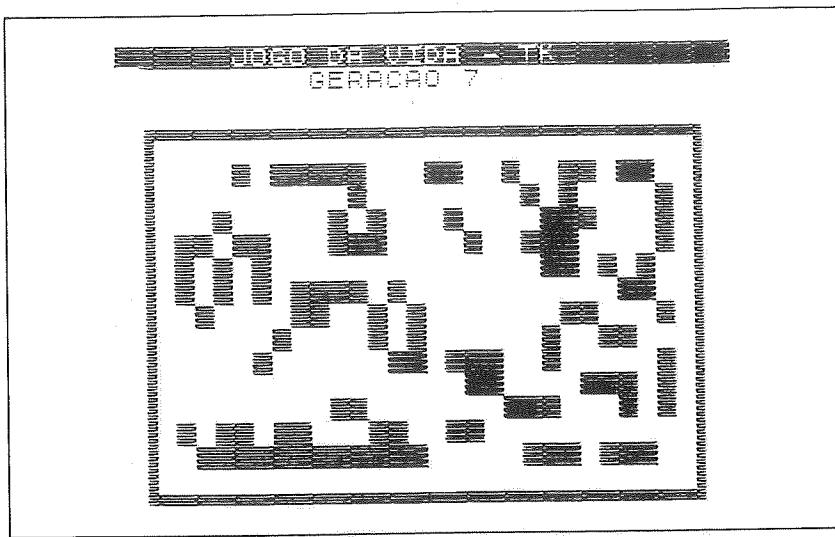


Figura 3.2 EXEMPLO (cont.)

## 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
13 CLEAR
14 LET T$ = " "
15 LET V$ = " "
16 REM 4 <= LI <= 30 E 4 <= LJ <= 17 -
17 DEF QUADRO
18 LET LI = 28
19 LET LJ = 15
20 LET DI = (32-LI)/2
21 LET DJ = (19-LJ)/2+2
22 CLS
23 DIM U$(LJ,LI)
```

Figura 3.4  
LISTAGEM DO PROGRAMA-FONTE (VIDA)

```

28 LET UU=INT ((LI*LU/4)
29 LET UU=INT (UU+RND*UU)
30 PRINT T$)
31 DIM U$(LU,LI)
32 PRINT AT 8,0;"UUU SORTEAR "
;UU;" PONTOS"
33 PRINT "AO ACASO PARA ORIGEM
DA VIDA"
34 PRINT
35 PRINT "TECLE I PARA QUE EU
COMECE."
36 LET A$=INKEY$
37 IF A$<>"I" THEN GOTO 50
100 CLS
101 FAST
102 LET Y=1
110 FOR I=1 TO LU
111 LET U$(I)=""
112 LET U$(I)=""
113 NEXT I
114 LET G=0
115 FOR K=1 TO UU
116 LET I=INT (RND*(LI-2))+2
117 LET J=INT (RND*(LU-2))+2
118 IF U$(J,I TO I)="■" THEN GO
TO 130
119 LET U$(J,I TO I)=""
120 NEXT K
121 PRINT AT 0,0/T$
122 LET H=DJ+LU
123 LET I=DI-1
124 LET J=DI+LI
125 FOR K=DI-1 TO H
126 PRINT AT K,I;"■";AT K,J;"■"
127 NEXT K
128 PRINT AT H,I;"■";AT H,J;"■"
129 PRINT AT DJ-1,I;"■";AT DJ-1
;J;"■"
130 LET J=DI+LI-1
131 LET I=DI-1
132 FOR K=DI TO J
133 PRINT AT I,K;"■";AT H,K;"■"
134 NEXT K
135 GOTO 325
200 FAST
205 LET Y=0
210 FOR J=2 TO LU-1
215 FOR I=2 TO LI-1
220 LET U$(J,I TO I)=""
225 LET X=CODE U$(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

```

```

U$(J+1,I+1)+CODE U$(J,I-1)+CODE
U$(J,I+1)
280 IF U$(J,I TO I)="" THEN GO
TO 280
280 IF X>=128 AND X<=384 THEN G
OTO 300
270 GOTO 310
280 IF X>384 THEN GOTO 310
300 LET U$(J,I TO I)=""
310 NEXT I
320 NEXT J
325 SLOW
330 LET K=DJ-1
335 PRINT AT 1,10;"GERACAO ";G
340 LET G=G+1
350 FOR J=1 TO LU
355 PRINT AT K+J,DI;U$(J)
360 LET Y=Y+(U$(J)<>U$(J))
370 LET U$(J)=U$(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 "VIDA"
9020 GOTO 10

```

Figura 3.4 LISTAGEM DO PROGRAMA-FONTE (VIDA) (cont.)

CAPÍTULO 4

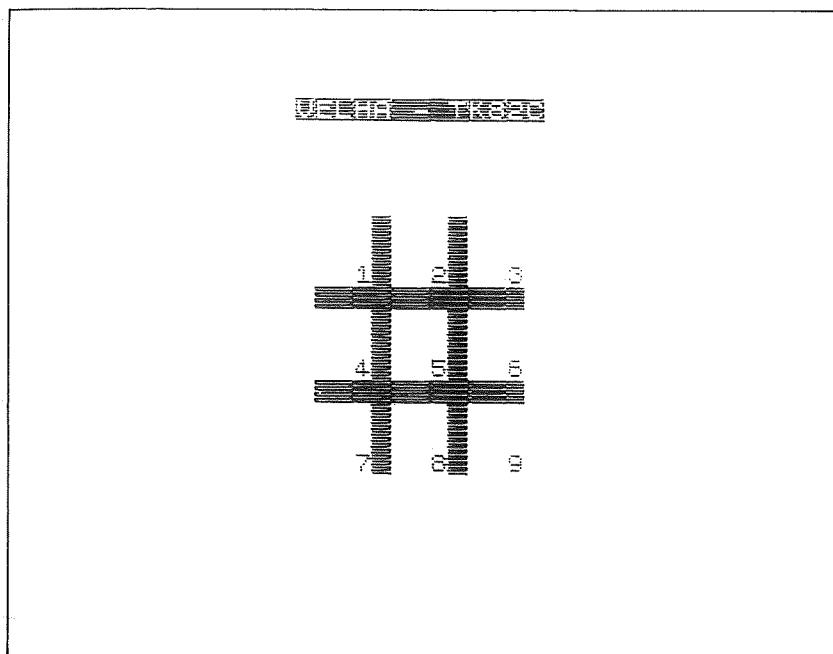
***O Desafio do  
Jogo da Velha  
Tridimensional***

**A****FICHA**

a) Tipo: Programa	b) Nome: <b>VELHA TRIDIMENSIONAL</b>	c) Versão: <b>1</b>	d) Revisão: <b>Ø</b>	e) Data: <b>22/10/83</b>
Sub-rotina	f) Objetivo: <b>Jogo</b>			
Função				
g) Linguagem: <b>BASIC</b> <input checked="" type="checkbox"/> <b>PASCAL</b> <input type="checkbox"/> <b>FORTH</b> <input type="checkbox"/> <b>OUTRAS (quais?)</b> <input type="checkbox"/> <b>PL/</b> <input type="checkbox"/> <b>FORTRAN</b> <input type="checkbox"/> <b>COBOL</b> <input type="checkbox"/> <b>C</b> <input type="checkbox"/>				
h) Compatibilidade:		i) Recursos:	j) Instruções/ Comandos Especiais:	
NE Z80	<input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	SLOW <input type="checkbox"/>	
NE Z8000	<input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	FAST <input type="checkbox"/>	
TK 80	<input type="checkbox"/>	MEMÓRIA: 2K	LPRINT <input type="checkbox"/>	
TK 82-C/83	<input checked="" type="checkbox"/>	16K	LLIST <input type="checkbox"/>	
TK 85	<input checked="" type="checkbox"/>	48K	COPY <input type="checkbox"/>	
SINCLAIR ZX 80	<input type="checkbox"/>	64K	DLOAD <input type="checkbox"/>	
SINCLAIR ZX 81	<input checked="" type="checkbox"/>	CASSETTE	DSAVE <input type="checkbox"/>	
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	DISQUETE	DMLOAD <input type="checkbox"/>	
TIMEX 1000	<input checked="" type="checkbox"/>	DISCO: 8"	DMSAVE <input type="checkbox"/>	
CP-200	<input checked="" type="checkbox"/>	5½ "	OUTROS (quais?)	<input type="checkbox"/>
CP-300	<input type="checkbox"/>	ALAVANCA ("JOYSTICK")		
APPLY 300	<input type="checkbox"/>	IMPRESSORA		
OUTROS (quais?)	<input type="checkbox"/>	OUTROS (quais?)		
<i>RINGO R-47Ø e AS - 16ØØ</i>				
k) Observações:				

**B****MANUAL DE UTILIZAÇÃO**

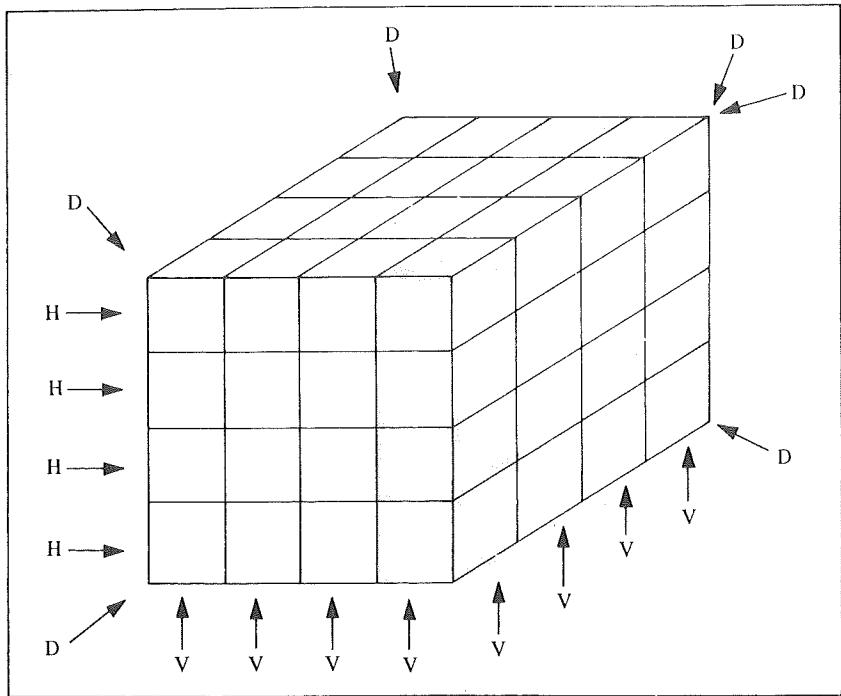
O jogo da VELHA é extremamente conhecido.

**Figura 4.1****TABULEIRO DO JOGO DA VELHA TRADICIONAL**

Usa uma matriz  $3 \times 3$  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  $3 \times 3$  temos um de  $4 \times 4$ .

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.



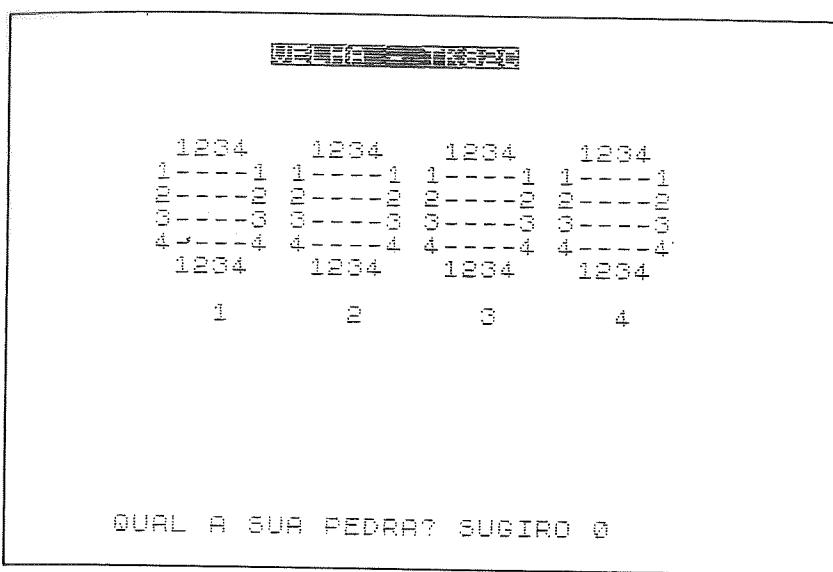
**Figura 4.2**  
PLANOS DO CUBO

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.

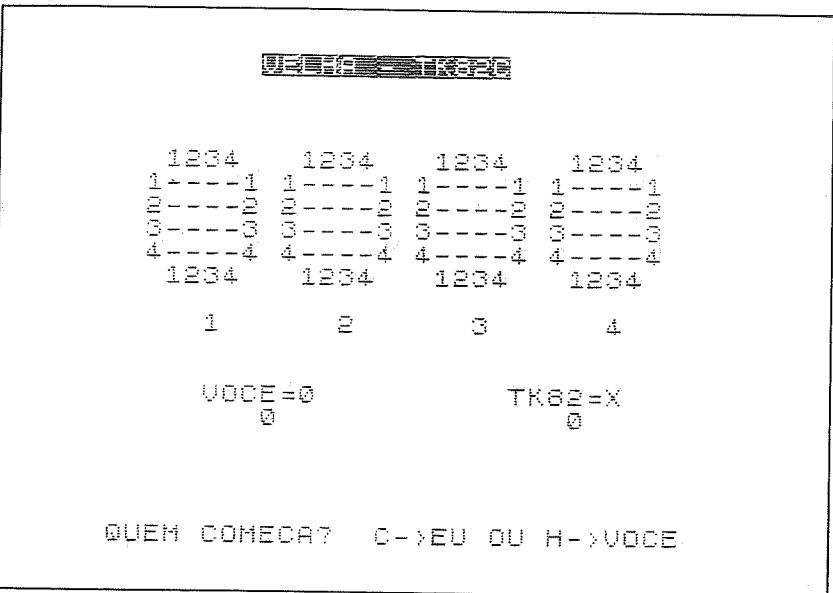
A primeira pergunta feita pelo sistema é sobre as pedras a serem utilizadas para identificar os jogadores (veja Figura 4.3). Normalmente se usa X e Ø. Escolha um deles para você. Outro símbolo qualquer pode ser utilizado, exceto o —, porém os de melhor efeito visual são Ø e X.

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

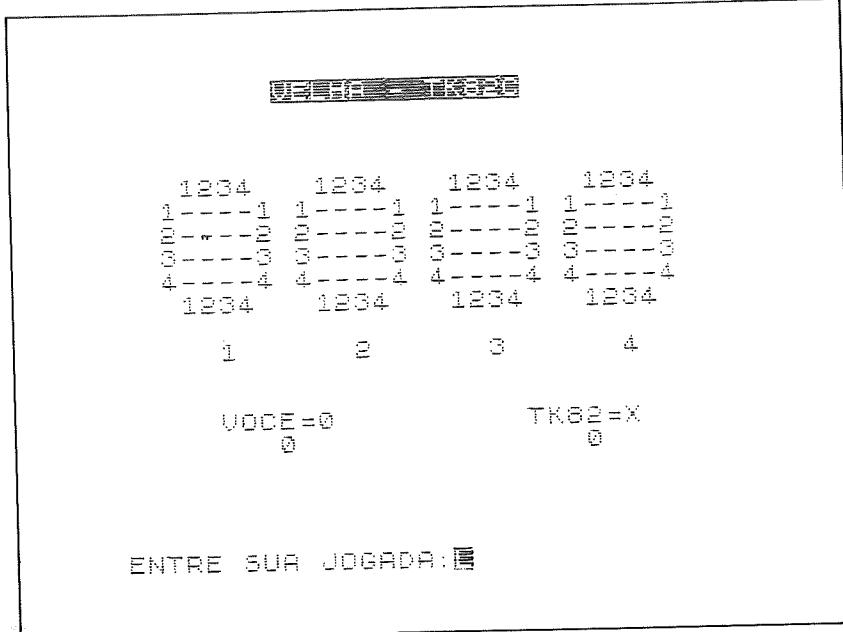
Se o computador é o primeiro a jogar, ele inicia o jogo automaticamente. Se o humano é o primeiro a jogar, deverá fazer sua primeira jogada.



**Figura 4.3**  
TABULEIRO DO CUBO



**Figura 4.4**  
ABERTURA



**Figura 4.5** *PEDIDO DE JOGADA*

Para jogar, tecle 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. Tecle 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.

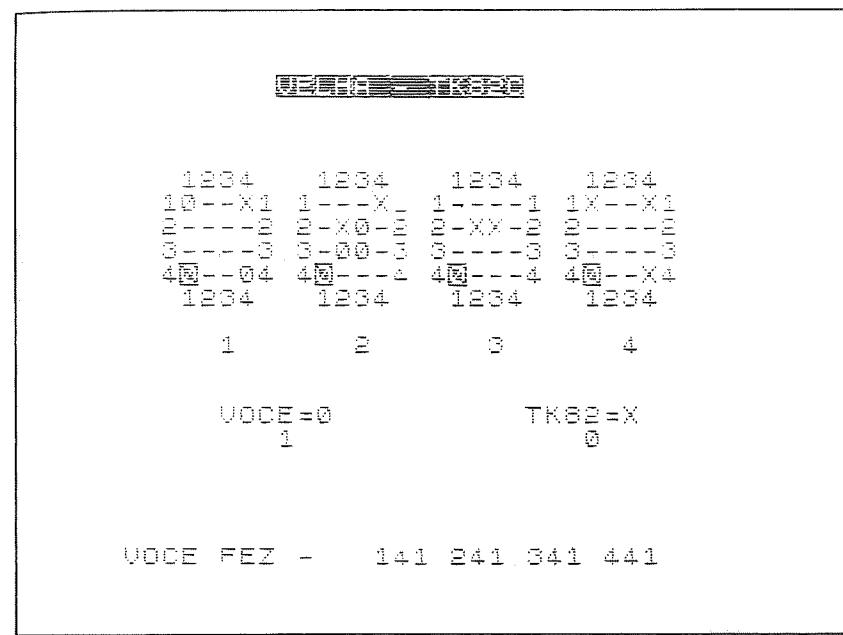
A terceira coordenada identifica a coluna no tabuleiro. Os quatro tabuleiros que aparecem na tela são os horizontais.

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

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*

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 cassete.

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. Façamos então uma lista com todas elas. Seriam 3 vetores de 76 x 4 posições, caso os tabuleiros fossem armazenados em uma matriz 4 x 4 x 4. Se porém linearizarmos esta matriz em um vetor de 64 posições, apenas um vetor de 76 x 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 sequência 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 colocadas 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(85)
50 DIM L(75)
60 DIM H(75,4)
70 DIM B(16)
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 "01080304 06
06070609101112 13141515 17181919
0 0122023324 05262728 09303132 333
43505 07306094 041424344 45464748
4 40505050505050 0750859000 0160
6304 01173049 0608013753 0900941617
1300945681 0801634560 060900654 100064
0000 143046680 001903551 0780339000 1
10743559 155014763 04003660 0800446
56 1000844666 160304664 01050913 17
01 0000 0303741450 495035761 08006101
4 10000 0303840446 500045880 030
71115 100238731 050394347 515555960
040081016 00042638 034044446 00256
6064 010501116 172202732 030064348
49546984 13100 0704 0900620326 484400
9306 061505550 01014161 022042690 0
30340003 04084 4454 49372513 0500000
14 01300715 02402616 01163552 05
203956 090064360 10304764 4934190
4 5030020308 057402718 01463116 012
04364 160273849 040234261 13263952
:(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
240 LET K1=0
250 LET K2=0
260 LET K3=0
270 LET HU=0
280 LET CO=0
290 PRINT AT 1,6; "██████████"
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 "
510 LET H$=INKEY$
520 IF H$="" THEN GOTO 510
530 LET CO$="X"
540 IF H$=C$ THEN LET C$="O"
550 PRINT AT 15,5;"VOCE=";H$;AT
15,21;"TK88=";H$;
560 PRINT AT 16,0;HU;AT 16,24;C
0
580 PRINT AT 21,0;"QUEM COMECA?
C->EU OU H->VOCE"
600 LET A$=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
770 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 1050
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.)

```

1100 IF A<00 THEN GOTO 1040
1105 PRINT AT 21,A;""
1110 LET A$=INKEY$
1115 IF A$="0" THEN GOTO 1250
1120 IF CODE A$<>118 THEN GOTO 1
1130
1155 LET M=M+1
1160 IF X(M)=0 THEN GOTO 1300
1170 PRINT AT 21,20;"-->OCUPADO"
1180 PAUSE 200
1190 GOSUB 3200
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$=H$
1320 PRINT AT 21,A;""
1330 PRINT AT 21,17;"")
1340 GOSUB 3000
1400
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))
1460 +(X(M(A,3))+X(M(A,4)))
1470 IF L(A)=4 THEN LET K1=A
1480 IF L(A)=15 THEN LET K2=A
1490 IF L(A)=3 THEN LET K3=A
1500 NEXT A
1510 RETURN
1520 LET G1=INT ((M-1)/16)
1530 LET J2=M-16*G1
1540 LET G2=INT ((J2-1)/4)
1550 LET G3=J2-4*G2
1560 PRINT G1+1;G2+1;G3;
1570 PRINT AT G2+6,G1*7+2+G3;M$)
1580 PAUSE 100
1590 RETURN
1600 SCROLL
1610 PRINT AT 21,0;"VOCE QUER OU
TRD? (S/N) "
1620 SCROLL
1630 LET A$=INKEY$
1640 IF A$="" THEN GOTO 2500
1650 IF A$="N" THEN GOTO 3600
1660 IF A$="S" THEN RUN
1670 PRINT AT 21,0;"TECLE S PARA

```

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

```

RESPONDER SIM"
2500 SCROLL
2550 PRINT AT 21,0;"OU TECLE N P
ARA RESPONDER NAO"
2600 GOTO 2500
3000 PRINT AT 21,0;"AGUARDE UM P
OUQUANTO 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$+128)
3330 FOR J=1 TO 4
3340 LET M=M(I,J)
3345 PRINT AT 21,8+J*4;""
3350 GOSUB 2000
3355 LET M(I,J)=65
3360 NEXT J
3370 PRINT AT 16,8;HU;AT 16,24;C
0
3380 RETURN
3400 FOR J=1 TO 4
3410 LET M=M(I,J)
3420 IF X(M)=0 THEN GOTO 3440
3430 NEXT J
3440 LET M$=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$=0$
3670 LET X(M)=5
3680 GOTO 2000
3690 STOP
3700 RUN

4000 FAST
4010 LET Z=10
4020 GOSUB 5000
4030 IF Z<>10 THEN GOTO Z
4040 GOSUB 6000
4050 LET Z=2
4060 GOSUB 5000
4070 IF Z<>2 THEN GOTO Z
4080 FOR Y=1 TO 16
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 6000
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
4280 GOSUB 3200
4300 PRINT AT 21,0;"DEU VELHA...
"
4310 GOTO 2500
4400 LET M=Z
4410 LET X(M)=5
4420 LET M$=0$
4430 GOSUB 6000
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))
5060 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 SE
VOCE SAIR DESSA"
6050 PAUSE 200
6060 GOTO 7100
6070 PRINT AT 21,0;"TA PENSANDO
SEU 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 R=1
7020 IF I-INT (I/4)*4>1 THEN LET
R=2
7030 FOR J=R TO 5-A STEP 5-2*A
7040 IF X(M(I,J))=S 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 9200
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)=
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

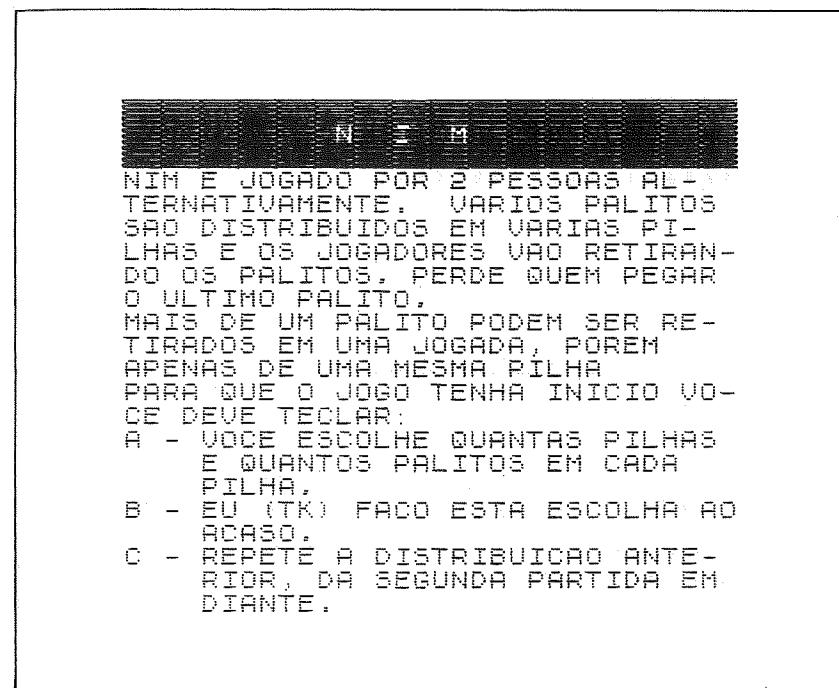
**A****FICHA**

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

**B****MANUAL DE UTILIZAÇÃO**

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 postas à disposição de dois jogadores, que selecionam uma pilha e dela retiram pelo menos um objeto, alternadamente. Perde quem pegar o último objeto.

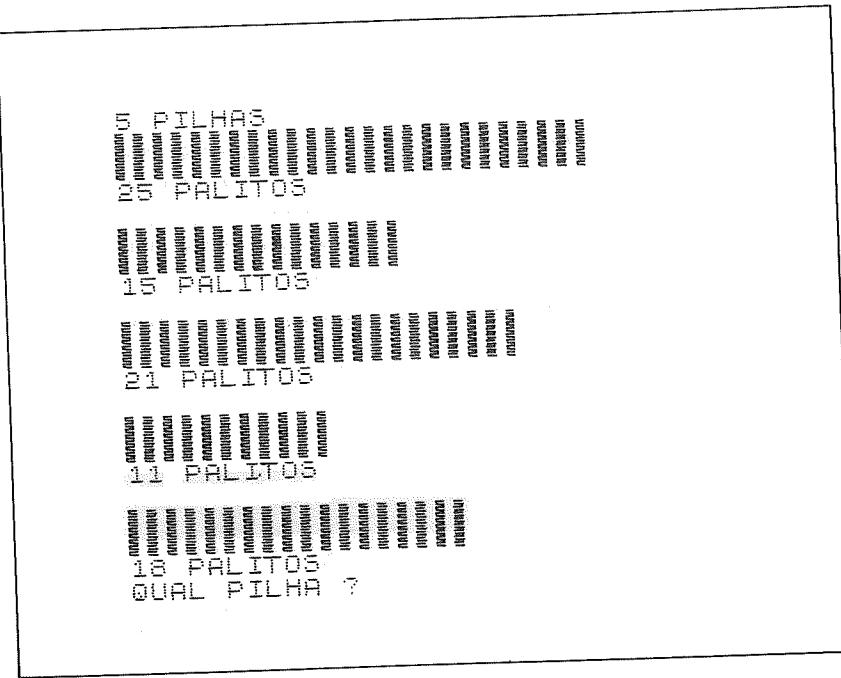
Ao iniciar-se o programa, são apresentadas instruções como na Figura 5.1.



**Figura 5.1**  
**TEL A DE INSTRUÇÕES**

Neste ponto o sistema está em fase de inicialização. Tecle 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*). Tecle 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. Tecle 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 cassete.

## D **MANUAL TÉCNICO**

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
2 REM VERSAO TK820
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)**

```

60 DIM V(5)
60 SLOW
60 LET W=0
60 CLS
100 PRINT AT 0,0;""
105 PRINT "NIM E JOGADO POR 2 P
ESSOAS ALTERNATIVAMENTE. VAR
IOS PALITOS"
107 PRINT "SAO DISTRIBUIDOS EM
VARIAS PILHAS E OS JOGADORES
VAO RETIRANDO OS PALITOS. PERDE
QUEM PEGAR O ULTIMO PALITO."
109 PRINT "MAIS DE UM PALITO PO
DEM SER RETIRADOS 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 QU
ANTAS PILHAS E QUANTOS PALITO
S EM CADA PILHA."
115 PRINT "B - EU (TK) FAO 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 W=0 THEN GOTO
119
126 CLS
130 IF A$="A" THEN PRINT AT 2,0
;BT 2,0;"DOIS PILHAS ?"
140 IF A$="B" THEN LET P=INT (R
ND*3+3)
145 IF A$="A" THEN INPUT P
150 IF P<2 OR P>INT P OR P>5 T
HEN GOTO 130
155 CLS
167 LET G=0
170 PRINT AT 1,0;P;" PILHAS"
180 FOR B=1 TO P
190 IF A$="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*N-B,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); "
426 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
466 FAST
470 FOR I=1 TO 5
480 LET V(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=5 TO 1 STEP -1
525 LET Y=E-1
530 IF C(I)<2**Y THEN GOTO 560
540 LET R(I)=R(I)+INT (10**Y+.1)
545
550 LET C(I)=C(I)-INT (2**Y+.1)
555 NEXT E
560 FOR E=5 TO 1 STEP -1
565 LET Y=E-1
570 LET X(I,E)=INT (R(I)/INT (1
0**Y+.1))
575 LET R(I)=R(I)-X(I,E)*INT (1
0**Y+.1)
580 LET U(E)=U(E)+X(I,E)
585 NEXT E
590 NEXT I
595 LET W=1
600 FOR E=5 TO 1 STEP -1
605 IF W=0 THEN GOTO 690
610 IF U(E)/2=INT (U(E)/2) THEN
GOTO 710
615 LET Q=INT (P*RND+1)
620 IF X(0,E)<>1 THEN GOTO 680
625 LET X(0,E)=0
630 LET W=2
635 GOTO 710
640 IF U(E)/2<>INT (U(E)/2) THE
N GOTO 710
645 LET X(0,E)=X(0,E)=0
650 NEXT E
655 FOR I=1 TO P
660 LET R(I)=0
665 FOR E=5 TO 1 STEP -1
670 LET Y=E-1
675 LET R(I)=R(I)+X(I,E)*INT (1
0**Y+.1)
680 NEXT E
685 FOR E=5 TO 1 STEP -1
690 LET Y=E-1
695 IF R(I)<INT (10**Y+.1) THEN
GOTO 730
700 LET C(I)=C(I)+INT (2**Y+.1)
705 LET R(I)=R(I)-INT (10**Y+.1)
710
715 NEXT E
720 NEXT I
725 IF W=2 THEN GOTO 1000

```

Figura 5.3

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

```

744 LET Q=INT (P*RND+1)
745 IF C(0)=0 THEN GOTO 744
746 PRINT AT 20,20;"AA"
747 LET C(0)=C(0)-1
750 LET D=C(0)-C(0)
755 IF D=0 THEN GOTO 744
760 SLOW
765 PRINT AT 21,0;"EU REGUEI "
D;" DA PILHA ");0)
770 LET N=0
775 LET T=D
780 GOSUB 370
785 IF G>1 THEN GOTO 300
790 CLS
795 PRINT AT 8,9;"*****"
800 AT 10,9;"*** VENCI ***";AT 10,
101;" PERDI ";AND G<>1);AT 12,9;
805 *****"
810 PAUSE 300
815 GOTO 90
820 IF Z>1 THEN GOTO 750
825 LET C(0)=1
830 GOTO 750
835 FAST
840 CLEAR
845 SAVE "NIM"
850 GOTO 10

```

Figura 5.3

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

CAPÍTULO 6

***Ligações  
Descendentes***

**A**

## FICHA

a) Tipo: Programa	b) Nome: <b>LIGAÇÕES DESCENDENTES</b>	c) Versão: <b>1</b>	d) Revisão: <b>Ø</b>	e) Data: <b>22/10/83</b>	
Sub-rotina	f) Objetivo: <b>Jogo</b>				
Função					
g) Linguagem: PL/	<input checked="" type="checkbox"/> BASIC	<input type="checkbox"/> PASCAL	<input type="checkbox"/> FORTH	<input type="checkbox"/> OUTRAS (quais?)	
	<input type="checkbox"/> FORTRAN	<input type="checkbox"/> COBOL	<input type="checkbox"/> C		
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	16K	<input checked="" type="checkbox"/>	FAST	<input checked="" type="checkbox"/>	
TK 85	48K	<input type="checkbox"/>	LPRINT	<input type="checkbox"/>	
SINCLAIR ZX 80	64K	<input type="checkbox"/>	LLIST	<input type="checkbox"/>	
SINCLAIR ZX 81	<input checked="" 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 1000	<input checked="" type="checkbox"/> DISCO: 8"	<input type="checkbox"/>	DSAVE	<input type="checkbox"/>	
CP-200	<input checked="" type="checkbox"/> 5½"	<input type="checkbox"/>	DMLOAD	<input type="checkbox"/>	
CP-300	ALAVANCA ("JOYSTICK")	<input type="checkbox"/>	DMSAVE	<input type="checkbox"/>	
APPLY 300	IMPRESSORA	<input type="checkbox"/>	OUTROS (quais?)	<input type="checkbox"/>	
OUTROS (quais?)	OUTROS (quais?)	<input type="checkbox"/>			
<b>RINGO R-470 e AS - 1000</b>					
k) Observações:					

**B**

## MANUAL DE UTILIZAÇÃO

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

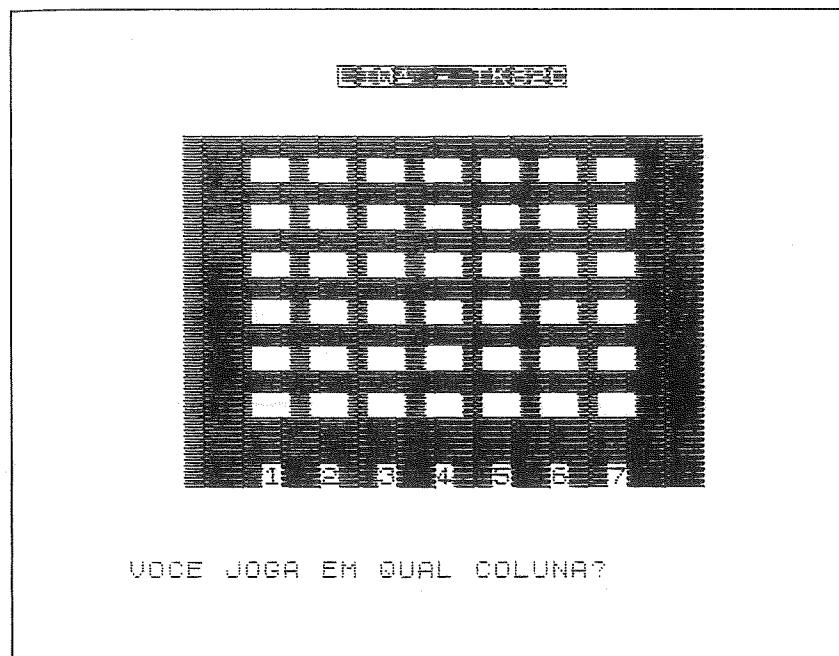


Figura 6.1  
TABULEIRO

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. Tecle 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 cassete. 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 300 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 ████ ████ ████
30 FAST
30 DIM B$(63,3)
32 DIM H$(3)
33 LET H$ = "█ O █"
35 DIM C$(3)
36 LET C$ = "█ X █"
40 DIM Q(7)

```

Figura 6.2

LISTAGEM DO PROGRAMA-FONTE (LIGAÇÕES DESCENDENTES)

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

```

436 NEXT I
437 PRINT AT 21,0;"EMPATAMOS EN
TR.
438 STOP
439 GOTO 438
440 LET S$=C$
441 FOR A=1 TO 7
442 LET C(A)=0
443 IF B$(A+1)<>B$ THEN GOTO 51
444
470 GOSUB 7000
480 GOSUB 6000
490 LET Q$(K)=B$
500 IF V<>0 THEN GOTO 6190
510 NEXT I
520 LET S$=H$
530 FOR A=1 TO 7
540 LET E=C(A)
541 LET C(B)=0
550 IF Q$(A+1)<>B$ THEN GOTO 51
551
560 GOSUB 7000
570 GOSUB 6000
580 LET Q$(K)=B$
590 IF V<>0 THEN GOTO 1000
600 LET C(A)=E*4+C(A)*6
610 NEXT I
620 LET A=1
630 FOR I=2 TO 7
640 IF C(A)>C(I) THEN GOTO 660
650 IF C(A)<C(I) THEN GOTO 670
660 IF RND*.5 THEN GOTO 680
670 LET A=I
680 NEXT I
690 GOSUB 6000
700 IF C(A)<1 THEN GOTO 650
710 LET S$=C$
720 GOSUB 7000
730 LET X=K
731 IF X<10 THEN GOTO 990
740 LET H$=H$
750 GOSUB 7000
760 LET H=K
770 GOSUB 6000
780 LET Q$(X)=B$
790 LET Q$(H)=B$
800 IF V=0 THEN GOTO 1000
810 LET C(A)=0
820 FOR I=1 TO 7
830 IF C(I)>0 THEN GOTO 620
840 NEXT I
850 PRINT AT 21,0;"DOU ESTA POR

```

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

```

PERDIDA
660 STOP
670 GOTO 660

990 LET B$(X1)=B$
1000 LET A=A+1
1010 PRINT AT 21,0;"JOGAREI NA O
OLUNA";A-1;;
1020 LET B$=0#
1030 PAUSE 100#
1040 GOSUB 5000
1050 GOTO 300
1060 SLOW
1070 LET S$=(CODE B$(2)>128)
1080 LET S$=((S$=0)-(S=1))*128
1090 LET N$="■"+CHR$(CODE B$(2))
1100 +
1110 FOR I=A TO A+54 STEP 9
1120 IF B$(I)<>B$ THEN GOTO 5110
1130 LET J=H
1140 LET L=INT(I/9)
1150 LET C=(I-L*9)*0
1160 LET L=L*B+0
1170 PRINT AT L,C;Z#
1180 PRINT AT L,C;D#
1190 LET L=L+1
1200 PRINT AT L,C;B#
1210 PRINT AT L,C;B$(I)
1220 NEXT I
1230 LET D$(J)=B$
1240 PRINT AT L,C;B#
1250 PAUSE 100
1260 RETURN
1270 LET U=0
1280 FOR I=A+1 TO A+55 STEP 9
1290 IF B$(I)=B$ THEN GOTO 6090
1300 NEXT I
1310 RETURN
1320 LET M=0
1330 LET M=(I>3)+(I>2)+(I>1)
1340 FOR J=I-1 TO I-M STEP -1
1350 IF B$(J)<>B$ THEN GOTO 6140
1360 LET B=B+1
1370 NEXT J
1380 FOR J=I+1 TO I+3
1390 IF B$(J)<>B$ THEN GOTO 6180
1400 LET B=B+1
1410 NEXT J
6180 IF B<3 THEN GOTO 6300
6190 GOTO 6500
6200 PRINT AT 21,0;"ACABOU.
6210 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 B$(J)<>S$ 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 B$(J)<>S$ THEN GOTO 6420
6400 LET B=B+1
6410 NEXT J
6420 IF B>W THEN GOTO 6500
6430 NEXT J
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 B$(I)<>B$ THEN GOTO 7040
7020 LET K=I
7030 NEXT I
7040 LET Q$(K)=B$
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 U=0
8505 FOR F=1 TO 7
8510 FOR G=1 TO 9
8520 LET U=U+1
8540 LET L=F*2+2
8550 LET C=G*3

```

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

```
0560 PRINT AT L,C;G$(U)
0570 PRINT AT L+1,C;P$
0580 NEXT G
0590 NEXT F
0600 GO SUB 6000
0610 PRINT AT 0,11;"LEIA - TECLADO
"
0620 FOR F=1 TO 7
0630 LET C=F*3+4
0640 PRINT AT 17,C;F
0650 NEXT F
0660 FOR F=1 TO 9
0670 LET C=F*3
0680 PRINT AT 3,C;P$
0690 NEXT F
0700 CONT
0700 FAST
0710 CLEAR
0720 SAVE "LIO-4"
0730 RUN
```

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

## CAPÍTULO 7

*Será possível  
Sobrar Só 1?*

**A**

## FICHA

a) Tipo: Programa	b) Nome: <u>RESTA-1</u>	c) Versão: <u>1</u>	d) Revisão: <u>Ø</u>	e) Data: <u>22/10/83</u>	
Sub-rotina	f) Objetivo: <u>Jogo</u>				
Função					
g) Linguagem: PL/	BASIC <input checked="" type="checkbox"/>	PASCAL <input type="checkbox"/>	FORTH <input type="checkbox"/>	OUTRAS (quais?) <input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<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	MEMÓRIA: 2K	<input type="checkbox"/>	SLOW	<input checked="" type="checkbox"/>	
TK 82-C/83	16K	<input checked="" type="checkbox"/>	FAST	<input checked="" type="checkbox"/>	
TK 85	48K	<input type="checkbox"/>	LPRINT	<input type="checkbox"/>	
SINCLAIR ZX 8Ø	64K	<input type="checkbox"/>	LLIST	<input type="checkbox"/>	
SINCLAIR ZX 81	CASSETE	<input checked="" type="checkbox"/>	COPY	<input type="checkbox"/>	
SINCLAIR SPECTRUM	DISQUETE	<input type="checkbox"/>	DLOAD	<input type="checkbox"/>	
TIMEX 10ØØ	DISCO: 8"	<input type="checkbox"/>	DSAVE	<input type="checkbox"/>	
CP-2ØØ	5½"	<input type="checkbox"/>	DMLOAD	<input type="checkbox"/>	
CP-3ØØ	ALAVANCA ("JOYSTICK")	<input type="checkbox"/>	DMSAVE	<input type="checkbox"/>	
APPLY 3ØØ	IMPRESSORA	<input type="checkbox"/>	OUTROS (quais?)	<input type="checkbox"/>	
OUTROS (quais?)	OUTROS (quais?)	<input type="checkbox"/>			
<u>RINGO R - 47Ø e</u>					
<u>AS - 1ØØØ</u>					
k) Observações:					
<hr/> <hr/> <hr/>					

**B**

## MANUAL DE UTILIZAÇÃO

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.

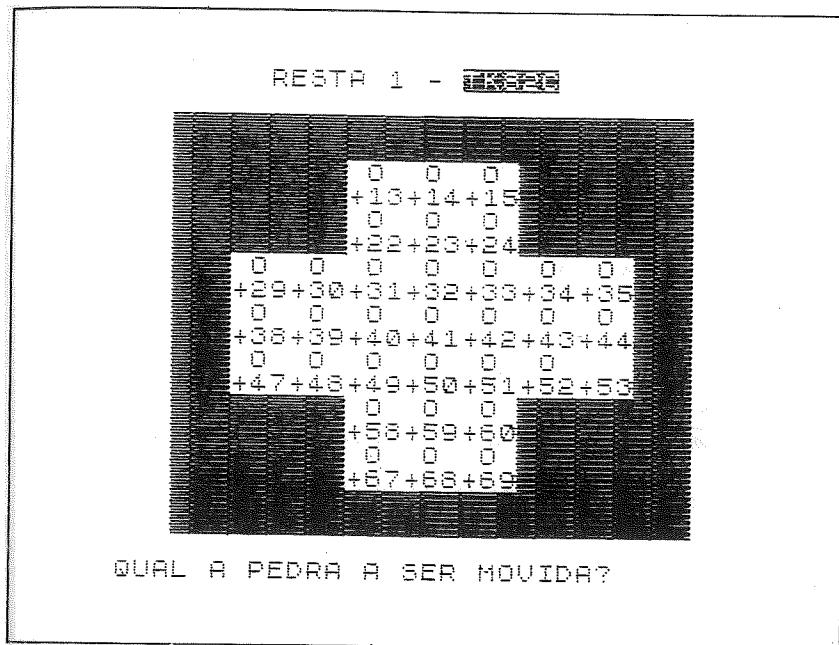


Figura 7.1  
TABULEIRO DO RESTA-1

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 sequência de movimentos que deixa apenas uma pedra no tabuleiro.

Desenhado o tabuleiro, o computador pergunta qual a pedra a ser movida. Escolha e tecle o número que se encontra sob ela, seguido de NEWLINE. A seguir vem a pergunta de para onde a pedra deve ser movida. Tecle 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$(81,3)  
40 FAST  
50 LET U=0  
60 FOR I=1 TO 9  
70 FOR J=1 TO 9  
80 LET U=U+1  
90 IF I=1 OR J=1 OR I=9 OR J=9  
THEN GOTO 120  
100 IF I>=4 AND I<=6 THEN GOTO  
140  
110 IF J>=4 AND J<=6 THEN GOTO  
140  
120 LET Q$(U) = "█"  
130 GOTO 150  
140 LET Q$(U) = " 0 "  
150 NEXT J  
160 NEXT I  
170 LET U=RND*57+13  
180 IF Q$(U) = "█" THEN GOTO 17  
0  
190 LET Q$(U) = "  
  
200 CLS  
210 PRINT AT 0,8;"RESTA 1 - ███  
██████████  
220 LET U=0  
230 FOR I=1 TO 9  
240 FOR J=1 TO 9  
250 LET U=U+1  
260 LET C=J*3  
270 LET L=I*2  
280 PRINT AT L,C;Q$(U)  
290 IF Q$(U) = "█" THEN GOTO 32  
0  
300 PRINT AT L+1,C;"+";U  
310 GOTO 330  
320 PRINT AT L+1,C;Q$(U)  
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 GOTO
0 410
380 PRINT AT 21,0;"ACHO QUE VOC
E NAO ENTENDEU"
390 PAUSE 200
400 GOTO 350
410 IF 0$(N)<>"0" THEN GOTO 3
80
420 PRINT AT 21,0;"MOVE PARA ON
DE?"
430 INPUT M
440 IF M<=69 AND M>=13 THEN GOTO
0 480
450 PRINT AT 21,0;"PUXA VIDA. T
A DIFICIL HEIM?"
460 PAUSE 200
470 GOTO 420
480 IF 0$(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 W=(M+N)/2
550 LET 0$(W)="
560 GOSUB 2000
570 LET W=N
580 LET 0$(W)="
590 GOSUB 2000
600 LET W=M
610 LET 0$(W)="
620 GOSUB 2000

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

```

Figura 7.2

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

```

760 IF 0$(N+W)="0" AND 0$(N-W)
=" " THEN GOTO 350
770 NEXT W
780 NEXT N
790 IF M<7 THEN GOTO 820
800 PRINT AT 21,0;"PUXA COMO VO
CE E FRACO, BOBÃO"
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 (W/9)
2010 LET C=(W-L*9)*3
2020 LET L=(L+1)*3
2030 PRINT AT L,C;0$(W)
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**

## FICHA

a) Tipo: Programa	b) Nome: <u>SENHA</u>	c) Versão: <u>1</u>	d) Revisão: <u>Ø</u>	e) Data: <u>22/10/85</u>	
Sub-rotina					
Função	f) Objetivo: <u>Jogo</u>				
g) Linguagem: PL/	BASIC <input checked="" type="checkbox"/>	PASCAL <input type="checkbox"/>	FORTH <input type="checkbox"/>	OUTRAS (quais?) <input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<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"/> CASSETE	<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 checked="" type="checkbox"/> OUTROS (quais?)	<input type="checkbox"/>			
<u>RINGO R-47Ø e</u> <u>45 - 1000</u>					
k) Observações:					

**B**

## MANUAL DE UTILIZAÇÃO

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.

```

VOU ESCOLHER UM NÚMERO DE  DÍGITOS PARA VOCÊ DESCOBRIR.
ENQUANTO ISTO VOCÊ ESCOLHE UM NÚMERO COM TAMBÉM  DÍGITOS PARA QUE EU POSSA DESCOBRIR.
JOGAREMOS ALTERNADAMENTE. VENCE QUEM PRIMEIRO DESCOBRIR A SENHA DO OUTRO.
CADA UM TENTA UM NÚMERO E O OUTRO RESPONDE COM A QUANTIDADE DE DÍGITOS CERTOS NO LUGAR  E DE DÍGITOS CERTOS NO LUGAR .
 PODE SER 3, 4, 5 OU 6. SE VOCÊ ESCOLHER 6 O JOGO É MAIS RÁPIDO. SE VOCÊ ESCOLHER 6 UMA JOGADA DO TK8200 PODE DEMORAR ATÉ 20 MINUTOS, DEPENDENDO DAS JOGADAS ANTERIORES.
TECLE AGORA O VALOR DE  OU SE VOCÊ NÃO ENTENDEU TECLE N.

```

Figura 8.1

TELDA 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.

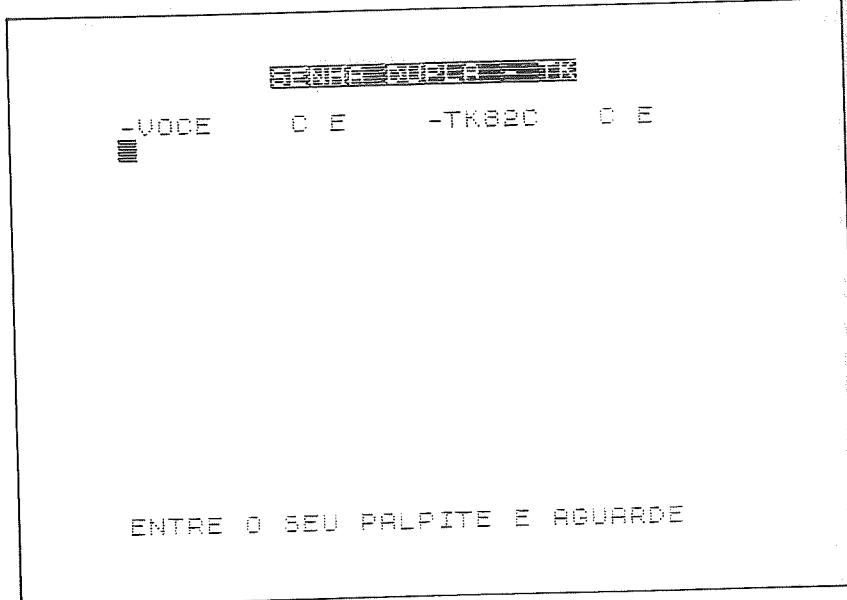


Figura 8.2  
*TELA INICIAL*

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

tecce 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.

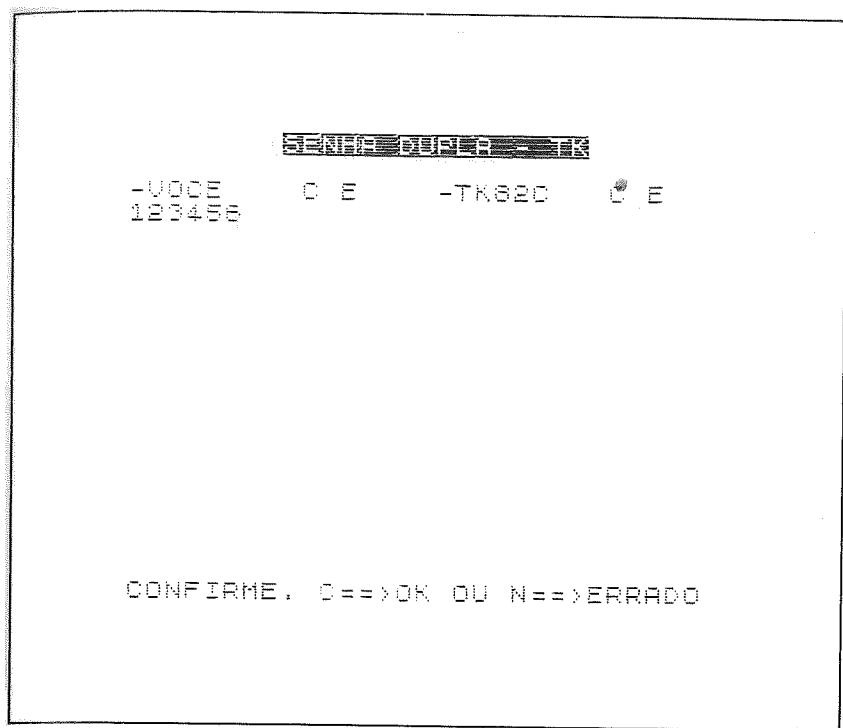


Figura 8.3  
*TELA COM PALPITES*

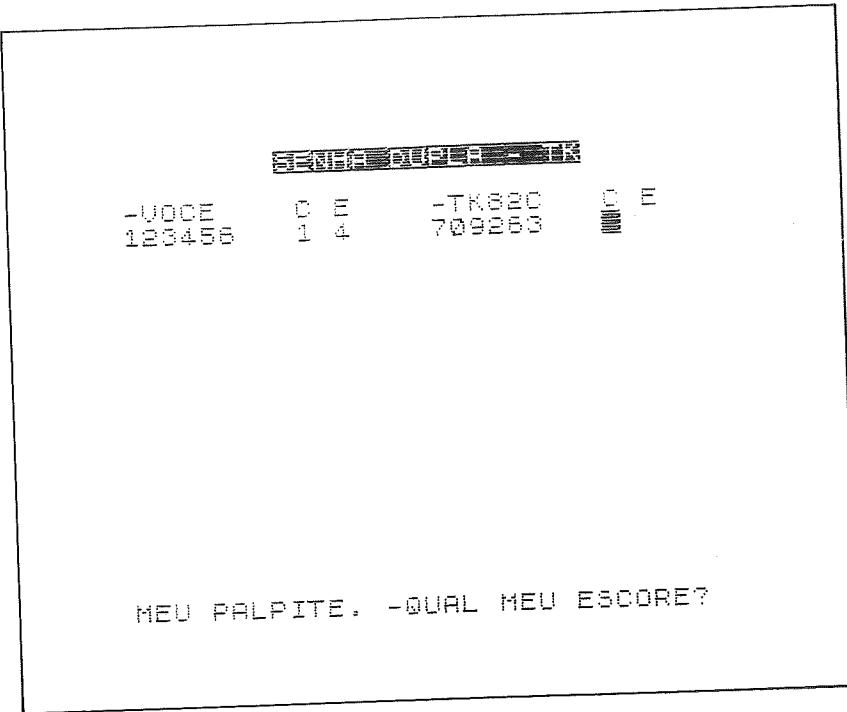
Isto permite que você modifique seu palpite. Se você teclar N o programa apagará seu palpite e voltará à pergunta anterior.

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 ESCORE?



**Figura 8.4**  
*PALPITE DO COMPUTADOR*

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

3478 — sua senha  
2480 — o palpito dele  
errado  
certo

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

Outro exemplo:

3478 — senha  
3870 — palpito dele  
errado  
certo

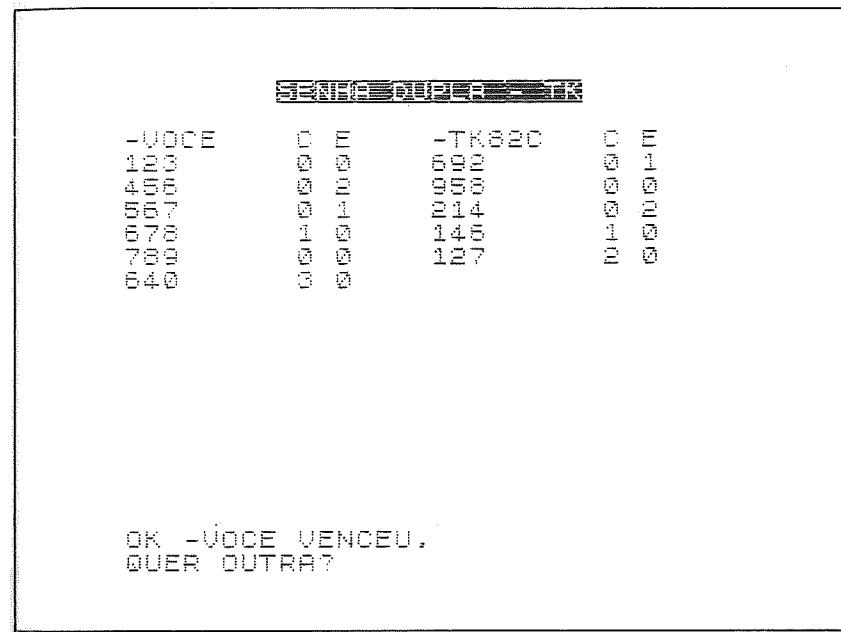
Resposta — 2 C (o 3 e o 7) e 1 E (o 8).

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 frequente 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.

# D MANUAL TÉCNICO

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 palpito 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 palpito. 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 – mas 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 ■ DI- GITOS PARA VOCÊ DESCO
BRIAR."
50 PRINT "ENQUANTO ISTO VOCÊ E
SCOLHE UM NUMERO COM TAMBEM ■
DIGITOS PA- RA QUE EU POSSA DESCO
BRIAR."
60 PRINT "JOGAREMOS ALTERNADAM
ENTE. VENCE QUEM PRIMEIRO DESCO
BRIR A SENHA DO OUTRO."
70 PRINT "CADA UM TENTA UM NUM
ERO E O OU- TRO RESPONDE COM A
QUANTIDADE DE DIGITOS CERTOS NO
LUGAR ■ E DE DIGITOS C
ERTOS NO LUGAR ■."
80 PRINT "■ PODE SER 3,4,5 OU
6. SE VOCÊ ESCOLHER 3 O JOGO E
MAIS RAPIDO. SE VOCÊ ESCOLHER 6 U
MA JOGADA DOTAKE82C PODE DEMORAR A
TE 20 MINU- TOS, DEPENDENDO DAS J
OGADAS ANTE-RIORES."
90 PRINT "TECLE AGORA O VALOR
DE ■ OU SE VOCÊ NÃO ENTENDEU TE
CLE N."
100 SLOW
110 LET A$=INKEY$
120 IF A$="N" THEN STOP
130 IF A$<"3" THEN GOTO 130
140 IF A$>"6" THEN GOTO 130
150 IF A$<"6" 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=S*5+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 ;AT 4,16;"-TK820 C E"
235 LET A=1
240 GO SUB 2000
245 LET J=1
250 LET L=5
255 REM HUMANO JOGA
260 SLOW
265 PRINT AT 21,0;"ENTRE O SEU
270 PALPITE E AGUARDE"
275 FOR I=1 TO 5
280 PRINT AT L,I-1;" "
285 LET A$=INKEY$
290 IF A$<"0" THEN GOTO 320
295 IF A$>"9" THEN GOTO 320
300 LET G(I)=VAL A$
305 IF I=1 THEN GOTO 410
310 FOR K=1 TO I-1
315 IF G(I)=G(K) THEN GOTO 320
320 NEXT K
325 PRINT AT L,I-1;A$
330 NEXT I
335 LET P=S
340 LET A=1
345 GO SUB 6000
350 IF A>1 THEN GOTO 290
355 GO SUB 2000
360 PRINT AT L,9;V;" ";W-V
365 IF V>S THEN GOTO 540
370 PRINT AT 21,0;"OK - VOCE VEN
375 CEU."
380 SCROLL
385 PRINT "QUER OUTRA?"
390 SLOW
395 LET A$=INKEY$
400 IF A$="S" THEN GOTO 200
405 IF A$>"N" THEN GOTO 500
410 STOP
415 RUN

```

Figura 8.6 LISTAGEM DO PROGRAMA-FONTE (SENHA DUPLA) (cont.)

```

540 PRINT AT 21,0;"POR FAVOR AG
545 UARDE ENQUANTO PENSO"
550 PAUSE 200
555 FAST
560 IF J>1 THEN GOTO 610
565 LET A=2
570 GO SUB 2000
575 GOTO 620
580 GO SUB 4000
585 PRINT AT 21,0;"MEU PALPITE,
590 -QUAL MEU ESCORE? "
595 FOR I=1 TO 5
600 PRINT AT L,I+15;D(J,I);
605 NEXT I
610 PRINT AT L,25;" "
615 SLOW
620 GO SUB 3500
625 LET B(J)=VAL A$
630 PRINT AT L,25;A$;" "
635 GO SUB 3500
640 LET C(J)=B(J)+VAL A$;
645 PRINT AT L,27;A$;
650 LET A=2
655 GO SUB 8000
660 IF A>2 THEN GOTO 630
665 IF C(J)<=5 THEN GOTO 790
670 PRINT AT 21,0;"NAO SEJA RID
675 ICULO
680 PAUSE 400
685 GOTO 630
690 IF B(J)=S-1 AND C(J)=5 THEN
695 GOTO 750
700 IF L<20 THEN GOTO 800
705 GO SUB 8070
710 SCROLL
715 GOTO 810
720 LET L=L+1
725 IF B(J)<S THEN GOTO 260
730 PRINT AT 21,0;"HI HI - VIU?
735 GANHEI HI HI"
740 PAUSE 600
745 GO SUB 8070
750 PRINT AT 21,0;"MEU NUMERO E
755 DE"
760 FOR I=1 TO 5
765 PRINT D(1,I);
770 NEXT I
775 PRINT "."
780 GOTO 490
785 FOR P=1 TO 5
790 LET D(A,P)=INT (10*RND)
795 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
2050 NEXT I
2060 NEXT P
2070 IF A=1 THEN RETURN
2080 LET J=2
2090 RETURN
3000 REM CONTA
3010 LET U=0
3020 LET V=0
3030 FOR I=1 TO P
3040 IF D(A,I)=G(I) THEN LET U=U
+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$>CHR$(5+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
4250 PRINT AT 21,0;"VOCE ME DEU"
4260 REPONDEUERRADA
4270 SCROLL
4280 PRINT AT 21,0;"LOGO VOCE PE"
4290 RDEU,
4300 GOTO 490
8000 PRINT AT 21,0;"CONFIRME. C="
+>OK OU N=>ERRADO"
8010 LET A$=INKEY$
8020 IF A$="C" THEN GOTO 8070
8030 IF A$<>"N" THEN GOTO 8010
8040 IF A=1 THEN PRINT AT L,0;"

8050 IF A>>1 THEN PRINT AT L,25;
8060 LET A=0
8070 PRINT AT 21,0;"

8080 RETURN
8090 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 Forca***

**A**

## FICHA

a) Tipo: Programa	b) Nome: <u>FORCA</u>	c) Versão: <u>1</u>	d) Revisão: <u>φ</u>	e) Data: <u>22/10/83</u>
Sub-rotina	f) Objetivo: <u>Jogo</u>			
Função				
g) Linguagem: PL/	BASIC <input checked="" type="checkbox"/>	PASCAL <input type="checkbox"/>	FORTH <input type="checkbox"/>	OUTRAS (quais?) <input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
h) Compatibilidade:	NE Z80 <input type="checkbox"/>	VÍDEO <input checked="" type="checkbox"/>	j) Recursos:	j) Instruções/ Comandos Especiais:
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input type="checkbox"/>	FAST <input checked="" type="checkbox"/>
TK 80 <input type="checkbox"/>	16K <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>	LLIST <input type="checkbox"/>
TK 82-C/83 <input checked="" type="checkbox"/>	64K <input type="checkbox"/>	CASSETE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>	DLOAD <input type="checkbox"/>
TK 85 <input checked="" type="checkbox"/>	DISQUETE <input checked="" type="checkbox"/>	DISCO: 8" <input checked="" type="checkbox"/>	DSAVE <input type="checkbox"/>	DMLOAD <input type="checkbox"/>
SINCLAIR ZX 80 <input type="checkbox"/>	5½" <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>	
OUTROS (quais?) <input checked="" type="checkbox"/>	RINGO R-470 e As - 1000			
k) Observações:				

**B**

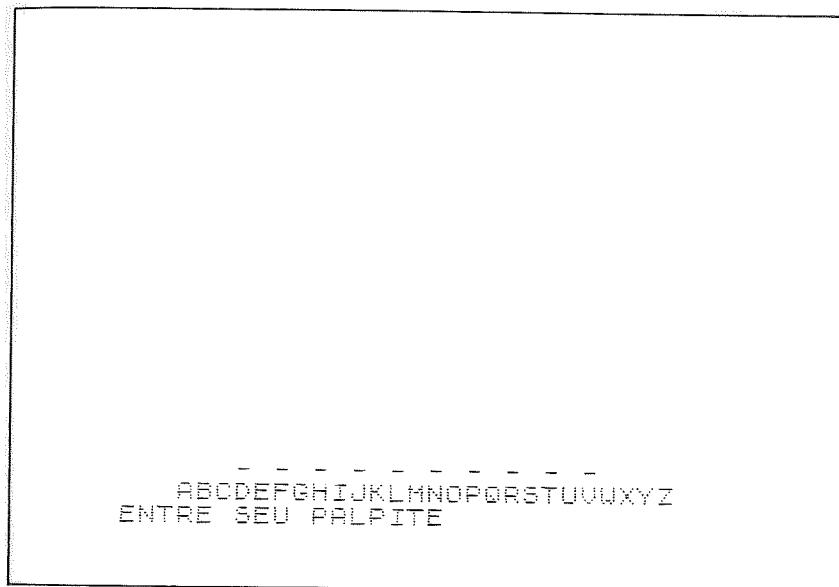
## MANUAL DE UTILIZAÇÃO

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).

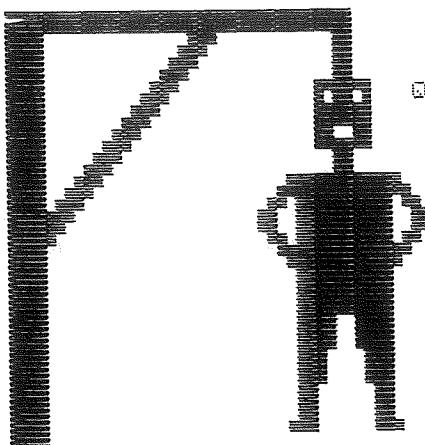
Para cada letra tentada e que não ocorre na palavra, uma parte de uma forca ou do enforcado é desenhada. O jogo termina com o descobrimento da palavra ou com o enforcamento (Figura 9.2).

O jogo se inicia – veja Figura 9.1 – com a tela em branco. Na antepenúltima linha temos tantos traços quantas 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

Tecle 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.



QUE  
DEUS  
PERDOE  
SEUS  
PECADOS

**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 | - ELETRODOMESTICO |
| 13 | - ELETRONICA      |
| 14 | - ENGRANAGEM      |
| 15 | - ESPOMEADO       |
| 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 | - PREOCUPACAO    |
| 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 | - LADRÃO         |
| 63 | - CASSETE        |
| 64 | - FLAMENGO       |
| 65 | - VASCO          |
| 66 | - MAQUINA        |
| 67 | - REVISTA        |
| 68 | - HELICOPTERO    |

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

79	-	EURO
77	-	MORÇA
74	-	MENINOS
73	-	MORADA
74	-	MARCO
74	-	JOUR
74	-	MONTECA
74	-	MULHERES
74	-	ALIVISAO
74	-	DEPARTAMENTO
74	-	COMPUTADOR
80	-	GRAVADOR

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 I – L + 1 até I) 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-aspas) 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, tecle 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 [REDACTED] ANTONIO JULIO LOS
510 BOTELO
520 SLOW
530 CLS
70 CLEAR
80 LET G$="ABCDEFGHIJKLMNOPQRS
TUUVXYZ"
100 GOSUB 3000
110 LET P=INT((RND*NP+1)
150 GOSUB 8500
200 LET Z=INT((30-L-L)/2-1)
210 FOR I=1 TO L
220 PRINT AT 19,Z+I+I;"-"
230 NEXT I
240 LET E=0
245 LET C=0
250 PRINT AT 20,0; ""
255 PRINT AT 20,(31-LEN G$)/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$(I TO I) THEN GOTO
410
380 LET C=C+1
390 LET X=X+1
400 PRINT AT 19,Z+I+I;A$
410 NEXT I
420 IF X=0 THEN GOTO 510
430 IF C<L THEN GOTO 350
440 PRINT AT 21,0;"PARABENS - U
OCE ACERTOU.
450 SCROLL
460 PRINT AT 21,0;"QUIR OUTRA?
(S/N)"
470 LET A$=INKEY$
480 IF A$="S" THEN RUN
490 IF A$<>"N" THEN GOTO 470
495 STOP
500 GOTO 450
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; "[REDACTED]"
1020 NEXT J
1030 PRINT AT 0,0; "[REDACTED]"
[REDACTED]
1040 FOR J=1 TO 8
1050 PRINT AT J,10-J; "[REDACTED]"
1060 NEXT J
1070 PRINT AT 9,2;"#"
1080 RETURN
2000 PRINT AT 6,16;"#"
2010 PRINT AT 4,16;"#"
2020 PRINT AT 5,16;"#"
2030 PRINT AT 6,17;"#"
```

Figura 9.5

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

```

3040 RETURN
3060 FOR J=7 TO 12
3080 PRINT AT J,15;""
3100 NEXT J
3120 RETURN
4000 PRINT AT 13,15;""
4010 PRINT AT 14,15;""
4020 PRINT AT 15,15;""
4030 PRINT AT 16,15;""
4040 PRINT AT 17,15;""
4050 RETURN
5000 PRINT AT 7,14;"";AT 8,13;""
5010 PRINT AT 9,13;"";AT 10,14;""
5020 RETURN
5030 PRINT AT 7,20;"";AT 8,20;""
5040 PRINT AT 9,20;"";AT 10,20;""
5050 RETURN
6000 PRINT AT 1,17;"";AT 2,17;""
6010 PRINT AT 3,21;"QUE";AT 5,20;"DEUS";
6020 PRINT AT 7,23;"PERDOE";AT 9,24;"SEUS";
6030 PRINT AT 11,25;"PECADOS"
6040 RETURN
7000 PRINT AT 21,0;"ERA:";P$
7010 GOTO 450
8000 LET NP=80
8010 RETURN
8020 LET N$="00+08+07+10+08+07+0
8030 +07+04+07+08+08+15+10+10+09+08+
8040 +08+05+05+09+12+07+07+05+09+07+07
8050 +08+07+11+06+10+06+11+11+06+09+0
8060 +08+18+09+09+13+09+09+08+10+09+
8070 07+05+04+09+10+09+07+07+06+10+08
8080 +14+07+06+07+08+05+07+07+11+05+0
8090 +07+04+05+04+06+05+09+11+10+08
8010 LET L=VAL N$(P#3+1 TO )
8020 LET I=VAL N$
8030 LET P$="PETROLEOPESTANAPEST
ICENTOPETELECOREBOQUECADAFALSOCA
DEIRACOLAREBENTODEFUMADODISCURSO
ELETRODOMESTICOELETRONICAREENGRENA
GEMESFOMEADOFERROLHOFERROVIAFERT
ILFORTEFORTALEZAFORTIFICANTEGRALO
CHAOFICINAOGIVADORNAMENTOPALADINOP
ALACIOPALADINOPALAVRAPREOCUPACAO
PULGARADIOLOGIARADIOGALVANIZADOR
ADIOGRAFIAGRANJAREBOTALHOGRANULA
DOINDUZIDOINDUMENTARIAIMBATIVELI
INDUSTRIAINTERNACIONALLABIRINTOLO
BISOMEMELEFANTELOCOMOTIVALOCATAR
IOTESOURALAPISLOBOMASCARADOMASSA
GISTAMATADOUROMOBILIAMOCHILAMOCO

```

Figura 9.5

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

```

TONORMALISTANORDESTENITROGLICERI
NAPOLICIALADRAOCASSETTEFLAMENGOVA
SCOMBUQUINAREVISTAHELICOPTEROLIVR
OFORCAMENINOSCASABARCOLOJABONECA
FESTATELEVISAOAPARTAMENTOCOMPUTA
DORGRAVADOR"(I-L+1 TO I)
9610 RETURN
9620 CLEAR
9630 FAST
9640 SAVE "FORCA"
9650 RUN
9660 SLOW
9670 GOSUB 8000
9680 FOR P=1 TO NP
9690 GOSUB 8500
9700 SCROLL
9710 PRINT AT 21,P<=9;P;" - ";P$
9720 NEXT P
9730 STOP

```

Figura 9.5

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

CAPÍTULO 10

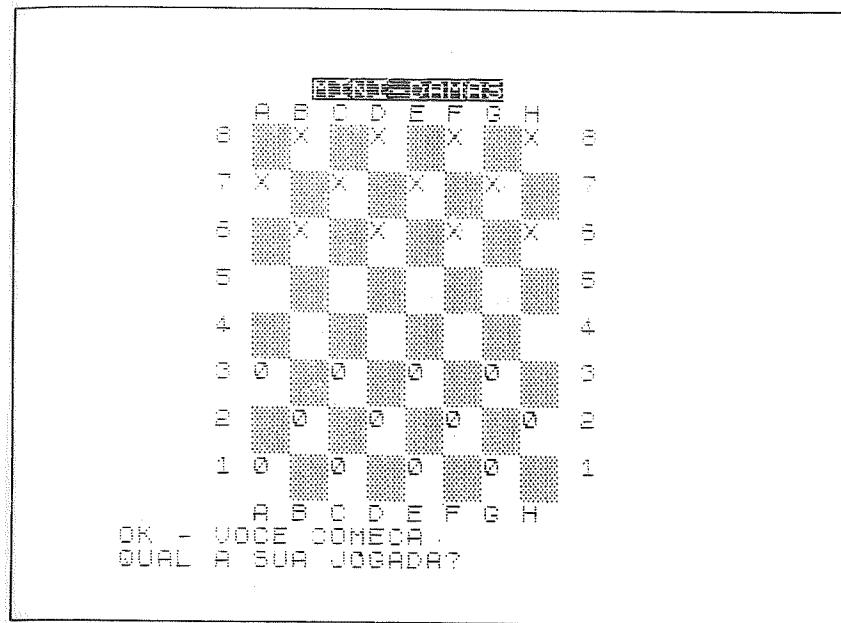
*Damas  
Tradicional*

A FICHA

a) Tipo: Programa	b) Nome: <u>DAMAS</u>	c) Versão: <u>1</u>	d) Revisão: <u>Ø</u>	e) Data: <u>22/10/83</u>
Sub-rotina	f) Objetivo: <u>Jogo</u>			
Função				
g) Linguagem: PL/	<input checked="" type="checkbox"/> BASIC	<input checked="" type="checkbox"/> PASCAL	<input type="checkbox"/> FORTH	<input type="checkbox"/> OUTRAS (quais?)
	<input type="checkbox"/> FORTRAN	<input type="checkbox"/> COBOL	<input type="checkbox"/> C	
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
TK 82-C/83	<input checked="" type="checkbox"/>	16K	<input checked="" type="checkbox"/>	FAST
TK 85	<input checked="" type="checkbox"/>	48K	<input type="checkbox"/>	LPRINT
SINCLAIR ZX 80	<input type="checkbox"/>	64K	<input type="checkbox"/>	LLIST
SINCLAIR ZX 81	<input checked="" type="checkbox"/>	CASSETE	<input checked="" type="checkbox"/>	COPY
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	DISQUETE	<input type="checkbox"/>	DLOAD
TIMEX 1000	<input checked="" type="checkbox"/>	DISCO: 8"	<input type="checkbox"/>	DSAVE
CP-200	<input checked="" type="checkbox"/>	5¼"	<input type="checkbox"/>	DMLOAD
CP-300	<input type="checkbox"/>	ALAVAÑCA ("JOYSTICK")	<input type="checkbox"/>	DMSAVE
APPLY 300	<input type="checkbox"/>	IMPRESSORA	<input type="checkbox"/>	OUTROS (quais?)
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

DAMAS é jogado em um tabuleiro plano, reticulado em  $8 \times 8$ , 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 subsequentes. 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 10.1).

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 × 8.

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

Da linha 3Ø à 39Ø decide-se quem começa e da 40Ø à 48Ø temos o controle da seqüência do jogo, através de chamadas alternadas as rotinas 20ØØ e 60ØØ, respectivamente jogada do adversário e do computador. A rotina 50ØØ é de gerência de tela.

A rotina 20ØØ 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 60ØØ seleciona o movimento e transfere para 203Ø de forma a evitar duplicidade de código. A rotina 30ØØ é chamada pela 20ØØ para efetuar o movimento de dama. A série de linhas do milhar 40ØØ é de complementação à rotina 20ØØ 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 BO
TELHO
30 FAST
40 REM INICIALIZACAO
42 DIM G(4)
44 DIM P(4)
46 DIM E$(4, 8)
50 DIM X$(10)
51 DIM Y$(10)
52 DIM O$(10)
53 DIM B$(8, 8)
54 DIM D$(10)
55 DIM R$(12, 20)
56 DIM L$(18)
57 LET G=-1
58 LET L=6
59 LET B$=" "
60 FOR Y=1 TO 8
61 FOR X=1 TO 8 STEP 2
62 LET D$(X,Y)=B$(1 TO 2)
63 LET D$(X+1,Y)=B$(3 TO 4)
64 NEXT X
65 LET B$=B$(3 TO 4)+B$(1 TO 2)
66 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)="O"
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
270 CLS
280 PRINT AT 21,0;"QUEM COMECA?"
290 INPUT H$
300 IF H$="EU" THEN GOTO 390
310 IF H$="TU" OR H$="VOCE" THE
N GOTO 440
320 SCROLL
330 PRINT AT 21,0;"NAO ENTENDI
O QUE VOCE DISSE."
340 SCROLL
350 GOTO 310
360 GOSUB 5000
370 PRINT "OK - VOCE COMECA"
380 LET J=0
390 LET EU=0
400 GOSUB 2000
410 LET EU=1
420 GOSUB 6000
430 GOSUB 6000
440 PRINT "MINHA JOGADA FOI ";T
$;";"
450 GOTO 410

2000 REM JOGADA HUMANA
2001 LET W=1
2002 IF J>0 THEN GOTO 4800
2010 PRINT "QUAL A SUA JOGADA?"
2013 LET O$="O"
2016 LET X$="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$-37
2040 IF E<1 OR E>8 THEN GOTO 450
2050 LET H$=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.)*

```

0100 LET H$=H$(8 TO 1)
0110 IF A$(E,H,1 TO 1)=0$ THEN GOTO 8150
0120 SCROLL
0130 PRINT T$;"???" NAO TEM PECH
0140 IF A$=B$ THEN GOTO 8800
0150 LET E$=H$(1 TO 8)
0160 LET A$=CODE H$-37
0170 IF A<1 OR A>8 THEN GOTO 450
0180 LET H$=H$(8 TO 1)
0190 LET B$=CODE H$-26
0200 IF B<1 OR B>8 THEN GOTO 450
0210 LET H$=H$(8 TO 1)
0220 IF A$(A,B)="" THEN GOTO 9
0230 SCROLL
0240 PRINT T$;"??" E$;" JA ESTA
0250 OCUPADO."
0260 GOTO 8800
0270 IF A$(E,H,2 TO 8)=0$ THEN GOTO 3000
0280 IF ABS (A-E)<>1 THEN GOTO 9
0290 IF (B-H)=W THEN GOTO 2800
0300 SCROLL
0310 PRINT "MOVIMENTO INCORRETO"
0320 GOTO 9000
0330 IF ABS (A-E)<>2 THEN GOTO 9
0340 IF ABS (B-H)<>2 THEN GOTO 9
0350 LET U=(A+E)/2
0360 LET V=(B+H)/2
0370 IF A$(U,V,1 TO 1)<>X$ THEN GOTO 8320
0380 LET A$(U,V)=""
0390 GOTO 8810
0400 IF J<>0 THEN GOTO 4600
0410 LET J=1
0420 LET A$(A,B)=A$(E,H)
0430 LET A$(E,H)=""
0440 LET T=8
0450 IF O$<>"0" THEN LET T=1
0460 IF B=1 AND A$(A,B,2 TO 8)<>0$ AND LEN H$=0 THEN LET A$(A,B)=0$+O$
0470 IF LEN H$=0 THEN RETURN

```

Figura 10.2  
LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

```

0480 IF LEN H$>2 THEN GOTO 4800
0490 LET T=9
0500 LET X=0
0510 GOTO 9160
0520 REM HUMANO MOVE RAINHA
0530 IF ABS (A-E)<>1 THEN GOTO 9
0540 IF ABS (B-H)<>1 THEN GOTO 9
0550 GOTO 9800
0560 IF A=E OR B=H THEN GOTO 230
0570 LET U=1
0580 LET U=1
0590 IF A<E THEN LET U=-U
0600 IF B>H THEN LET U=-U
0610 LET O=0
0620 LET N=H+U
0630 FOR M=A+U TO A-U STEP U
0640 IF A$(M,N)="" THEN GOTO 9
0650 IF A$(M,N,1 TO 1)=0$ THEN GOTO 9
0660 LET X=N
0670 LET Y=N
0680 LET C=C+1
0690 LET Z=N+U
0700 NEXT M
0710 IF C=0 THEN GOTO 8800
0720 IF C>1 THEN GOTO 8800
0730 LET A$(X,Y)=""
0740 GOTO 8810
0750 SCROLL
0760 PRINT T$;"???" NAO ENTENDI "
0770 GOTO 6800
0780 IF H$<>"?" THEN GOTO 4700
0790 GOSUB 5000
0800 GOTO 8000
0810 IF H$<>"FIM" THEN GOTO 4500
0820 SCROLL
0830 PRINT "HA HA HA - GANHEI"
0840 STOP
0850 RUN
0860 GOSUB 5000
0870 SCROLL
0880 PRINT T$;"??" H$;" NAO FA
Z SENTIDO."
0890 GOTO 6800
5000 REM MOSTRA TABULEIRO

```

Figura 10.2  
LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

```

6010 CLS
6020 PRINT TAB(10);"XXXXXXXXXX"  

6030 GOSUB 6500
6040 FOR Y=8 TO 1 STEP -1
6050 PRINT TAB(5;Y)
6060 FOR X=1 TO 8
6070 PRINT A$(X,Y);
6080 NEXT X
6090 PRINT " " Y
6100 PRINT TAB(6);"  

6110 IF A$(1,Y) <> "XXXXXX" THEN PRINT
" "
6120 PRINT " " " " " " "
6130 NEXT Y
6140 PRINT TAB(7);"A B C D E F G"
6150 RETURN
6000 REM XXXXXX JOGA
6010 LET J=0
6020 FOR Y=1 TO 8
6030 FOR X=1 TO 8
6040 IF A$(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+38)
6075 LET L(J)=8
6080 LET E=X
6090 LET H=Y
6100 IF A$(X,Y,8 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>0 THEN GOTO 6300
6170 GOSUB 5000
6180 PRINT "OK - ESSA EU PERDI"
6190 SCROLL
6200 PRINT "A PROXIMA, QUEM SABE  
"
6210 STOP
6220 RUN
6230 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
6330 IF P(I)>P(U) THEN GOTO 6360
6340 LET U=I
6350 NEXT I
6410 IF L(U)<4 THEN GOTO 6170
6420 LET H=R$(U,1 TO L(U))
6430 IF J<4 THEN GOSUB 8500
6440 LET X$="X"
6450 LET O$="X"
6460 LET U=
6470 LET W=-1
6480 GOTO 8030
7000 REM MOVE PEAO
7010 LET S=0
7020 LET O$="O"
7030 LET X$="X"
7040 LET TK=7100
7050 LET DM=10
7100 LET K=0
7105 FOR I=-1 TO 1 STEP 0
7106 FOR T=-1 TO 1 STEP 0
7110 LET K=K+1
7115 LET F(K)=0
7116 LET F$(K)=""
7120 LET G(K)=0
7121 LET H=E
7122 LET A=E
7123 LET B=H
7124 LET N=H
7125 GOSUB 7600
7130 LET G(K)=-G(K)
7140 LET A=A+T
7150 LET B=H+T
7160 IF A<1 OR A>8 OR B<1 OR B>8
THEN GOTO 7270
7160 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+T
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+96)
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 7298
7290 IF G(I)>G(K) THEN GOTO 7295
7295 IF RND(.5 THEN GOTO 7298
7298 LET K=I
7300 NEXT I
7305 IF G(K)=0 THEN RETURN
7310 IF S=1 AND F(K)=0 THEN RETU
RN
7303 IF F$(K)<>" " THEN GOTO 73
00
7304 LET G(K)=0
7305 GOTO 7275
7308 FOR I=1 TO L(J) STEP 2
7310 IF R$(J,I TO I+1)=F$(K) THE
N RETURN
7312 NEXT I
7315 LET R$(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
0
7370 LET S=1
7400 GOTO TK

7500 FOR Z=-1 TO 1 STEP 0
7510 FOR O=-1 TO 1 STEP 0
7520 LET U=R-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 7680
7550 IF A$(U,V,1 TO 1)<>" " THEN
GOTO 7660
7590 LET U=R+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$(U,V,1 TO 1)="0" THEN G
OTO 7690
7660 NEXT O
7670 NEXT M
7680 RETURN
7690 LET G(K)=G(K)-DM
7700 RETURN
0000 REM MOVE DAMA
0001 REM VERSAO QUEBRA GALHO
0010 LET S=0
0010 LET O=" "
0030 LET X="X"
0040 LET TK=8100
0060 LET D=15
0100 LET K=0
0105 FOR I=-1 TO 4 STEP 0
0110 FOR T=-1 TO 1 STEP 0
0115 LET K=K+1
0011 LET F(K)=0
0120 LET G(K)=0
0120 LET M=0
0120 LET B=0
0120 LET Z=0
0120 LET N=0
0120 GOSUB 7500
0120 LET G(K)=+G(K)
0140 LET D=D+1
0160 LET B=B+T
0160 IF A<1 OR A>8 OR B<1 OR B>8
THEN GOTO 8270
0160 IF A$(A,B,1 TO 1)="X" THEN
GOTO 8270
0190 IF A$(A,B)=" " THEN GOTO 8
0200
0210 LET B=B+T
0220 LET D=D+1
0230 IF A<1 OR A>8 OR B<1 OR B>8
THEN GOTO 8270
0240 IF A$(A,B)<>" " THEN GOTO
0270
0240 LET G(K)=G(K)+12
0244 LET U=(E+H)/2
0245 LET V=(H+B)/2
0246 IF A$(U,V,2 TO 2)="0" THEN
LET G(K)=G(K)+0
0247 LET P(K)=1
0248 LET G(K)=G(K)+12
0260 LET F$(K)=CHR$(A+37)+CHR$(
B+96)

```

Figura 10.2  
LISTAGEM DO PROGRAMA-FONTE (DAMAS) (cont.)

```

0060 LET M=I
0064 LET N=B-T
0065 GOSUB 7500
0070 NEXT T
0072 NEXT I
0075 GOTO 7275
0080 LET K=0
0084 LET U=S
0090 FOR A=1 TO 8
0094 FOR B=1 TO 8
0098 IF A$(A,B,1 TO 1) <> "X" THEN
0100 GOTO 6570
0104 IF A$(A,B,2 TO 2) = "X" THEN
0108 LET U=U+1
0112 LET U=U+1
0116 IF A$(A,B,1 TO 1) <> "O" THEN
0120 GOTO 6600
0124 IF A$(A,B,2 TO 2) = "O" THEN
0128 LET K=K+1
0132 LET K=K+1
0136 NEXT B
0140 NEXT A
0144 IF U=K THEN RETURN
0148 GOSUB 5000
0152 PRINT "ASSIM VAI FICAR MUIT
0156 O MONOTONO."
0160 PRINT "PELO NUMERO DE PONTO
0164"
0168 IF U>K THEN GOTO 4710
0172 SCROLL
0176 GOTO 6180
0180 SCROLL
0184 IF EU=0 THEN GOTO 2000
0188 PRINT "ALGO DE PODRE NO PRO
GRAMA."
0192 STOP
0196 RUN
0200 FAST
0204 CLEAR
0208 SAVE "DAMAS"
0212 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	<u>1</u>	<u>Ø</u>	<u>22/10/83</u>
Sub-rotina	<input type="checkbox"/>			
Função	<input type="checkbox"/>			
f) Objetivo: <u>Jogo</u>				
g) Linguagem: <input checked="" type="checkbox"/> BASIC <input type="checkbox"/> PASCAL <input type="checkbox"/> FORTH <input type="checkbox"/> OUTRAS (quais?) <input type="checkbox"/> <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 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"/>	LLIST <input type="checkbox"/>	
TK 85	<input checked="" type="checkbox"/>	48K <input type="checkbox"/>	COPY <input type="checkbox"/>	
SINCLAIR ZX 8Ø	<input checked="" type="checkbox"/>	64K <input type="checkbox"/>	DLOAD <input type="checkbox"/>	
SINCLAIR ZX 81	<input 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 10ØØ	<input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DMSAVE <input type="checkbox"/>	
CP-2ØØ	<input checked="" type="checkbox"/>	5¼" <input type="checkbox"/>	OUTROS (quais?) <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"/>		
<u>RINGO R - 47Ø e</u>				
<u>AS - 10ØØ</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.

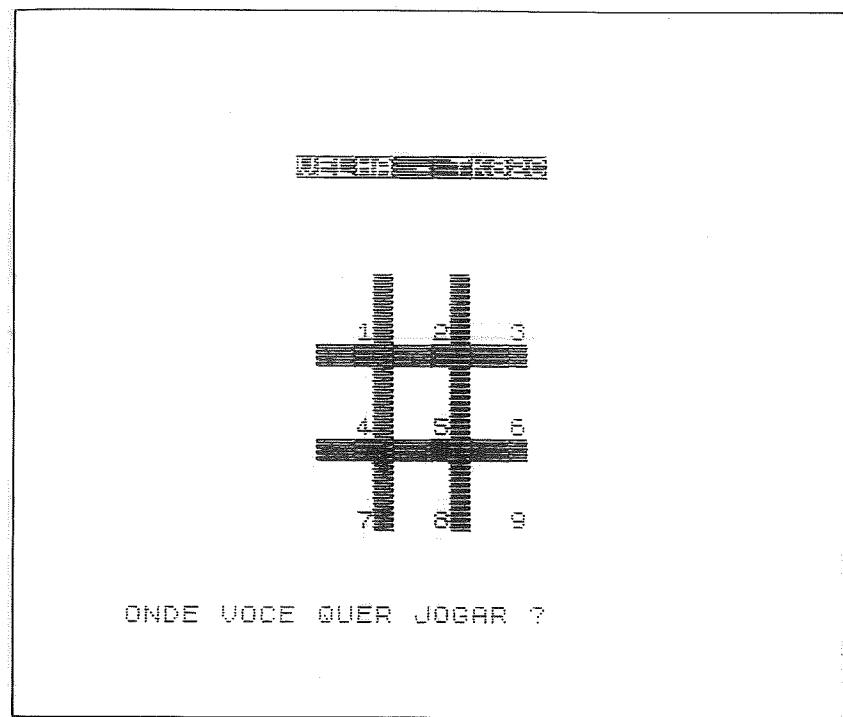


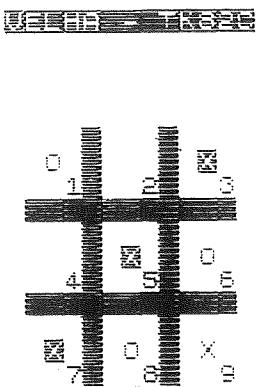
Figura 11.1 TABULEIRO DA VELHA

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?



HA-HA - GANHEI COM 3 5 7

**Figura 11.2 EXEMPLO DE VITÓRIA**

Escolha entre Ø e X que dão melhor visual; contudo qualquer caractere pode ser escolhido.

Neste ponto você é informado, como na Figura 11.3.

QUAL A SUA PEDRA ? Ø  
JOGAREI COM X  
JOGAREI UMA MOEDA  
SE DER CARA EU COMECO,  
SE DER COROA VOCE COMECA.  
  
DEU CARA - EU COMECO.

**Figura 11.3 TELA DE INICIAÇÃO**

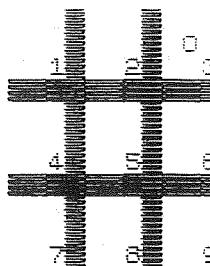
Então é feito um sorteio, ao acaso, e uma das duas mensagens aparece:

DEU CARA – EU COMECO  
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.



OK - VOCE JOGOU 3

**Figura 11.4 JOGADA DE Ø 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:

a) 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.

b) 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.

c) Sorteia-se ao acaso. Se o número sorteado for maior que Q1 escolhe-se conforme o item d. Observe que Q1 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.

d) 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 Q1 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 Q1 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 DI=0
60 FOR I=1 TO 8
70 FOR J=1 TO 3
80 LET R(I,J)=VAL "12345678914
7256369158367"(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$="O"
210 SCROLL
215 IF H$=" " 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 CARA
EU COMECO."
265 SCROLL
```

Figura 11.5  
LISTAGEM DO PROGRAMA-FONTE (VELHA)

```

886 PRINT AT 21,0;"SE DER COROA
896 COMECA."
876 LET V=INT (RND*2)
875 REM LET V=1
886 SCROLL
886 SCROLL
886 SCROLL
886 PRINT AT 21,0;"DEU ");("COROA
- DEU COMECA."); AND V=0); ("COROA
- VOCE COMECA."); AND V=1)
896 PAUSE 500
896 FOR I=1 TO 9
896 LET B(I)=0
896 NEXT I
896 LAST
896 CLS
896 PRINT AT 2,9;"XXXXXXXXXX
896
846 LET CC=0
856 FOR I=1 TO 3
866 FOR J=1 TO 3
876 PRINT AT 4*I+2+J,13;"■ ■"
876 NEXT J
896 IF I<>3 THEN PRINT AT 4*I+6
896
896 NEXT I
896 FOR I=0 TO 8
896 PRINT AT 9+4*INT (I/3),12+4
* (I-3*INT (I/3)); I+1
436 NEXT I
436 SLOW
446 IF V=0 THEN GOTO 1470
446
600 GOSUB 7000
600 GOSUB 6500
605 PRINT AT 21,0;"ONDE VOCE QU
ER JOGAR ?"
610 INPUT M
615 GOSUB 7000
620 IF M>0 AND M<10 THEN GOTO 5
620
630 PRINT AT 21,0;"VOCE DEVE RE
SPONDER ENTRE 1 E 9");
640 PAUSE 200
650 GOTO 500
660 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 60000
791 GOSUB 65000
792 IF OC<0 THEN GOTO 2000
793 LET OC=0
794 LET TU=0
795 LET H=1 TO 6
796 LET B(I)=B(R(I,1))+B(R(I,2))
797 IF H>OC THEN GOTO 1000
798 IF H>TU THEN LET TU=I
799 IF H>OC+TU THEN LET EU=I
800 NEXT H
801 IF EU=0 THEN GOTO 1200
802 LET H=EU
803 FOR J=1 TO 3
804 LET M=R(I,J)
805 IF B(M)=0 THEN GOTO 910
806 NEXT J
807 GOSUB 7000
808 PRINT AT 21,0;"HAM-HAM - JO"
809 COM";
810 LET M$=C$
811 GOSUB 60000
812 GOSUB 65000
813 GOSUB 7000
814 PRINT AT 21,0;"HA-HA - GANH"
815 COM";
816 GOTO 1600
817 GOSUB 7000
818 LET OI=OI+2
819 PRINT AT 21,0;"SNIFF - PERD"
820 COM";
821 LET M$=CHR$(CODE M$+128)
822 FOR J=1 TO 3
823 LET M=R(I,J)
824 PRINT AT 21,16+2*j;" ";
825 GOSUB 60000
826 NEXT J
827 PAUSE 100
828 SCROLL
829 LET U=U=0
830 PRINT AT 21,0;"QUER OUTRA ?"
831 INPUT A$
832 IF A$="S" THEN GOTO 300
833 IF A$="N" THEN STOP
834 SCROLL
835 PRINT AT 21,0;"RESPONDA S P"
836 OUTRA,"
837 SCROLL
838 PRINT AT 21,0;"RESPONDA N P"

```

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

```

0000 PRINT AT 21,0; "BOA TENTATIVA"
0000 GOTO 1100
0000 IF T=0 THEN GOTO 1400
0010 LET I=T
0010 GOSUB 7000
0030 PRINT AT 21,0; "BOA TENTATIVA"
0030 FOR J=1 TO 3
0030 LET M=R(I,J)
0030 IF B(M)=0 THEN GOTO 1200
0030 NEXT J
0030 PRINT "JOGAREI COM "
0030 LET M=0
0030 LET B(M)=4
0030 GOSUB 6500
0030 GOSUB 6500
0030 GOTO 500
0030 IF B(I)=RND THEN GOTO 1470
0030 GOSUB 6500
0030 LET M=4
0030 GOSUB 6500
0030 IF M>4 THEN GOTO M
0030 GOSUB 6400
0030 LET M=1
0030 GOSUB 6500
0030 IF M>1 THEN GOTO M
0030 GOSUB 6400
0030 LET M=5
0030 IF B(M)=0 THEN GOTO 1600
0030 FOR I=1 TO 3 STEP 2
0030 FOR J=1 TO 3 STEP 2
0030 LET M=(I-1)*3+J
0030 IF B(M)=0 THEN GOTO 1600
0030 NEXT J
0030 NEXT I
0030 FOR M=1 TO 6
0030 IF B(M)=0 THEN GOTO 1600
0030 NEXT M
0030 GOSUB 7000
0030 PRINT AT 21,0;
0030 GOTO 1200
0030 IF M=2*INT(M/2) THEN GOTO
0030 GOSUB 7000
0030 PRINT AT 21,0;
0030 LET M=10-M
0031 REM IF B(5)=0 THEN LET M=M-
0032 IF M=0 THEN LET M=8
0040 IF B(M)<>0 THEN GOTO 1470
0050 PRINT "E EU !"
0050 GOTO 1200

```

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

```

3000 FOR I=1 TO 8
3000 LET A(I)=B(R(I,1))+B(R(I,2))
3000 +B(R(I,3))
3040 REM PRINT AT I+8, (M-1)*8; "
3040 "
3050 REM PRINT AT I+8, (M-1)*8; I;
3050 "-"; 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
3080 IF B(R(I,J))=0 THEN LET B(R
3080 (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 1200

4000 GOSUB 7000
4010 IF A(I)<4 THEN GOTO 4100
4020 PRINT AT 21,0;"QUERO VER SO"
4020 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 PENSANDO"
4100 UE SOU BURRO?"
4110 FOR I=1 TO 8
4120 IF INT A(I)=4 THEN GOTO 403
00
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****FICHA**

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:	NE Z80 <input type="checkbox"/>	i) Recursos:	VÍDEO <input checked="" type="checkbox"/>	j) Instruções/ Comandos Especiais:
NE Z8000 <input checked="" type="checkbox"/>	TECLADO <input checked="" type="checkbox"/>	MEMÓRIA: 2K <input type="checkbox"/>	SLOW <input checked="" type="checkbox"/>	
TK 80 <input type="checkbox"/>	16K <input checked="" type="checkbox"/>	48K <input type="checkbox"/>	FAST <input checked="" type="checkbox"/>	
TK 82-C/83 <input checked="" type="checkbox"/>	64K <input type="checkbox"/>		LPRINT <input type="checkbox"/>	
TK 85 <input checked="" type="checkbox"/>			LLIST <input type="checkbox"/>	
SINCLAIR ZX 80 <input type="checkbox"/>			COPY <input type="checkbox"/>	
SINCLAIR ZX 81 <input checked="" type="checkbox"/>	CASSETE <input type="checkbox"/>		DLOAD <input type="checkbox"/>	
SINCLAIR SPECTRUM <input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>		DSAVE <input type="checkbox"/>	
TIMEX 1000 <input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>		DMLOAD <input type="checkbox"/>	
CP-200 <input checked="" type="checkbox"/>	5 1/4 " <input type="checkbox"/>		DMSAVE <input type="checkbox"/>	
CP-300 <input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>		OUTROS (quais?) <input type="checkbox"/>	
APPLY 300 <input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>			
OUTROS (quais?) <input checked="" type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>			
<i>RINGO R-470 e AS - 1000</i>				
k) Observações:	<hr/> <hr/>			

**B****MANUAL DE UTILIZAÇÃO**

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.

À pergunta

QUEM COMECA?

responda

EU, TU ou VOCE.

Neste ponto será desenhada a tela da Figura 12.1.

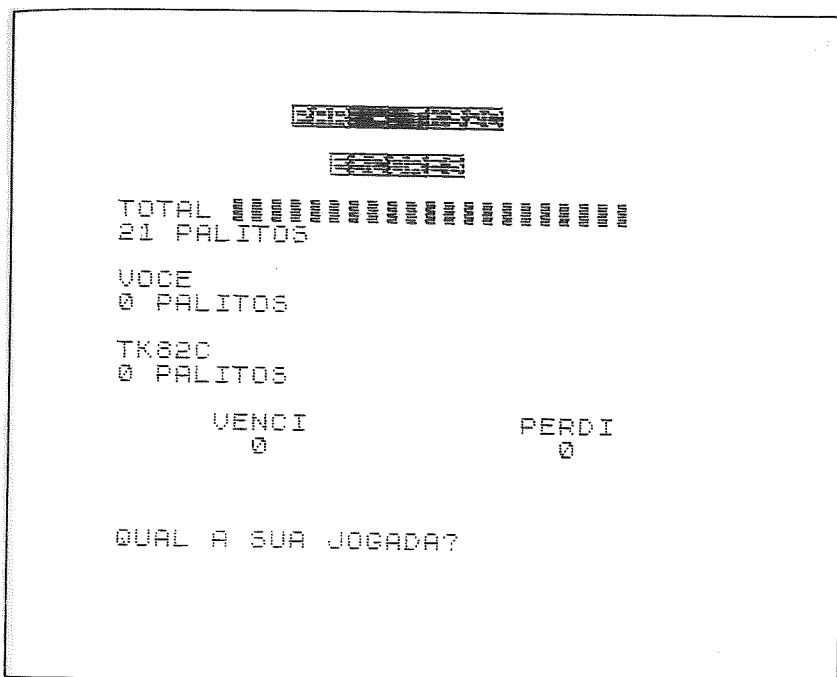


Figura 12.1  
TELA DO JOGO PAR

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 ← 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 [REDACTED]
20 REM VERSAO PROGRAMADA
30 REM ATONIO 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 ENTENDER
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
240 GOSUB 1000
250 IF P<1 THEN GOTO 650
260 GOSUB 5000
270 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
550 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)

```

570 LET TU=TU+1
580 GOTO 800
700 PRINT AT 19,0;"VIU? GANHEI,
HA HA HA"
710 LET EU=EU+1
800 PRINT AT 20,0;"QUER OUTRA?
(N/N)"
805 LET R=R+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 NPO"
860 PRINT AT 21,0;"CASO VOCÊ QU
EIRA JOGAR OUTRA"
870 GOTO 810
1000 CLS
1010 PRINT AT 0,0;"[REDACTED]"
1020 PRINT AT 2,11;"[REDACTED]"
1030 PRINT AT 4,0;"TOTAL";P$(1 T
0 P+1)
1040 PRINT AT 5,0;P/T$
1050 PRINT AT 7,0;"VOCÊ ";P$(1 T
0 U+1)
1060 PRINT AT 8,0;U/T$
1070 PRINT AT 10,0;"TK820";P$(1
TO T+1)
1080 PRINT AT 11,0;T/T$
1090 PRINT AT 13,5;"VENCEI";AT 13
21;"PERDI"
1100 PRINT AT 14,7;EU;AT 14,23;T
0
1110 RETURN
5000 REM TK JOGA
5010 LET R=P-6*INT (P/6)
5020 IF INT (U/2)=U/2 THEN GOTO
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

```

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

```

5200 IF R<1,5 THEN GOTO 5120
5210 IF R>5,3 THEN GOTO 5180
5220 LET J=R-1
5230 LET T=T+J
5240 LET P=P-J
5250 PRINT AT 18,0;"MINHA JOGADA
...";J
5300 PAUSE 300
5340 RETURN
5360 FAST
5310 CLEAR
5380 SAVE "PAR"
5390 RUN

```

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

CAPÍTULO 13

*Ensine ao  
Computador  
a Jogar Palitos*

**A**

## FICHA

a) Tipo: Programa	b) Nome: <u>PAR - C</u>	c) Versão: <u>1</u>	d) Revisão: <u>Ø</u>	e) Data: <u>22/10/83</u>
Sub-rotina	f) Objetivo: <u>Jogo</u>			
Função				
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:				
NE Z80	<input type="checkbox"/>	i) Recursos: VÍDEO <input checked="" type="checkbox"/>	j) Instruções/ Comandos Especiais:	
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"/>	<input type="checkbox"/>
TK 82-C/83	<input checked="" type="checkbox"/>	16K <input checked="" type="checkbox"/>	FAST <input checked="" type="checkbox"/>	<input type="checkbox"/>
TK 85	<input checked="" type="checkbox"/>	48K <input type="checkbox"/>	LPRINT <input type="checkbox"/>	<input type="checkbox"/>
SINCLAIR ZX 80	<input type="checkbox"/>	64K <input type="checkbox"/>	LLIST <input type="checkbox"/>	<input type="checkbox"/>
SINCLAIR ZX 81	<input checked="" type="checkbox"/>	CASSETE <input checked="" type="checkbox"/>	COPY <input type="checkbox"/>	<input type="checkbox"/>
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	DISQUETE <input type="checkbox"/>	DLOAD <input type="checkbox"/>	<input type="checkbox"/>
TIMEX 1000	<input checked="" type="checkbox"/>	DISCO: 8" <input type="checkbox"/>	DSAVE <input type="checkbox"/>	<input type="checkbox"/>
CP-200	<input checked="" type="checkbox"/>	5 1/4 " <input type="checkbox"/>	DMLOAD <input type="checkbox"/>	<input type="checkbox"/>
CP-300	<input type="checkbox"/>	ALAVANCA ("JOYSTICK") <input type="checkbox"/>	DMSAVE <input type="checkbox"/>	<input type="checkbox"/>
APPLY 300	<input type="checkbox"/>	IMPRESSORA <input type="checkbox"/>	OUTROS (quais?) <input type="checkbox"/>	<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

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.

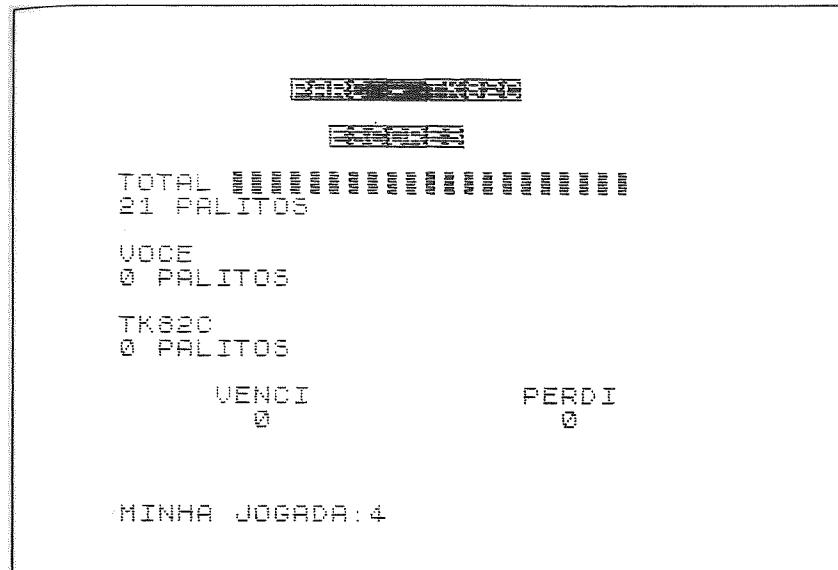


Figura 13.1  
TEL A DO JOGO PAR-C

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 ATONIO 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
170 LET U=0
180 LET T=0
190 LET P=INT (((13*RND+13)/2)*2
+1
200 GOSUB 1000
210 LET E1=E
220 LET L1=L
230 LET E=(U/2-INT (U/2))*2+1
240 LET L=INT (P-(INT (P/6))*6+
1)
250 LET J=M(E,L)
260 IF J>=P THEN GOTO 600
270 IF J<=0 THEN GOTO 660
280 PRINT AT 18,0;"MINHA JOGADA
";J
290 PAUSE 300
300 LET P=P-J
310 LET T=T+J
320 GOSUB 1000
330 LET X=18
340 PRINT AT X,0;"QUAL A SUA JO
GADA?"
350 INPUT J
360 IF J>4 THEN GOTO 470
370 IF J>0 AND J<=P THEN GOTO 5
00
380 PRINT AT 18,0;"TA BRINCANDO
?";
390 LET X=19
400 GOTO 440
410 IF J=P THEN GOTO 650
420 LET P=P-J
430 LET U=U+J
440 GOTO 300
450 PRINT AT 18,0;"MINHA JOGADA
";P
460 LET M(E,L)=P
470 LET T=T+P
480 LET P=0
490 IF T/2=INT (T/2) THEN GOTO
700
500 PRINT AT 19,0;"OK. ESTA VOC
E GANHOU."
510 LET TU=TU+1
520 IF M(E,L)=1 THEN GOTO 900
530 LET M(E,L)=M(E,L)-1

```

Figura 13.2  
LISTAGEM DO PROGRAMA-FONTE (PAR-C) (cont.)

```

590 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"
810 INPUT X$
820 IF X$="N" THEN STOP
830 IF X$="S" THEN GOTO 800
840 PRINT AT 18,0;"ACHO QUE VOC
E NAO ENTENDEU."
850 PRINT AT 19,0;"CASO VOCE QU
ER JOGAR OUTRA"
860 PRINT AT 20,0;"ENTRE S DE S
IM. CASO CONTRARIO ENTRE N DE N
O."
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 0,0;"BRASIL 82"  

1020 PRINT AT 0,11;"BRASIL 82"  

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;"VENCI";AT 13
,21;"PERDI"
1100 PRINT AT 14,7;EU;AT 14,23;T
O
1110 RETURN
0000 CLS
0010 PRINT
0020 FOR I=1 TO 9
0030 PRINT M(1,I),M(2,I)
0040 NEXT I
0050 INPUT X$
0060 GOTO 800
0070 FAST
0080 CLEAR
0090 SAVE "PAR-C"
0099 GOTO 10

```

Figura 13.2

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

## CAPÍTULO 14

### *Uma Aventura Espacial*

**A**

## FICHA

a) Tipo: Programa	b) Nome: <b>AVVENTURA ESPACIAL</b>	c) Versão: <b>1</b>	d) Revisão: <b>Ø</b>	e) Data: <b>22/10/81</b>
Sub-rotina	f) Objetivo: <b>Jogo</b>			
Função				
g) Linguagem:	BASIC <input checked="" type="checkbox"/>	PASCAL <input type="checkbox"/>	FORTH <input type="checkbox"/>	OUTRAS (quais?)
PL/	<input type="checkbox"/> FORTRAN	<input type="checkbox"/> COBOL	<input type="checkbox"/> C	
h) Compatibilidade:	i) Recursos:			
NE Z80	<input type="checkbox"/> VÍDEO	<input checked="" type="checkbox"/> j) Instruções/ Comandos Especiais:		
NE Z8000	<input checked="" type="checkbox"/> TECLADO	<input checked="" type="checkbox"/>		
TK 80	<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 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"/> CASSETE	<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 checked="" type="checkbox"/> OUTROS (quais?)			
<b>RINGO R-470 e AS - 100Ø</b>				
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 terrestres.

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*.

SUAS ORDENS CAP. KIRK.

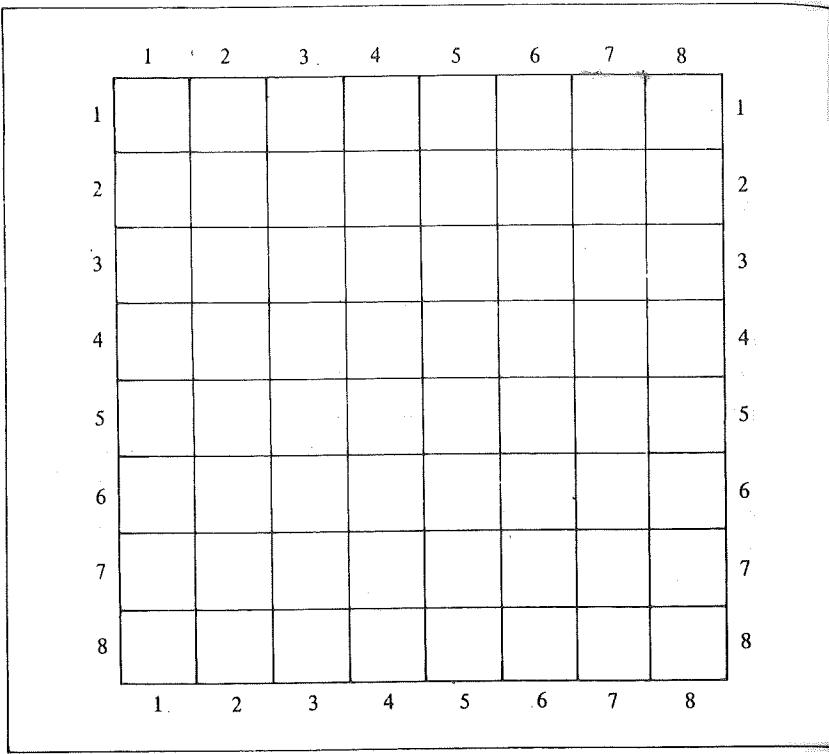
UMA FROTA KLIGON COMPOSTA DE  
18 NAVES INVADIU A GALÁXIA  
E VOCÊ TEM 38 ANOS  
ESTELARES PARA VENCE-LOS.

4 BASES ESTARÃO A SUAS ORDENS

TECLE OO TECLA (EXCETO ~~DEL~~)

Figura 14.1  
TELHA DE ABERTURA

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



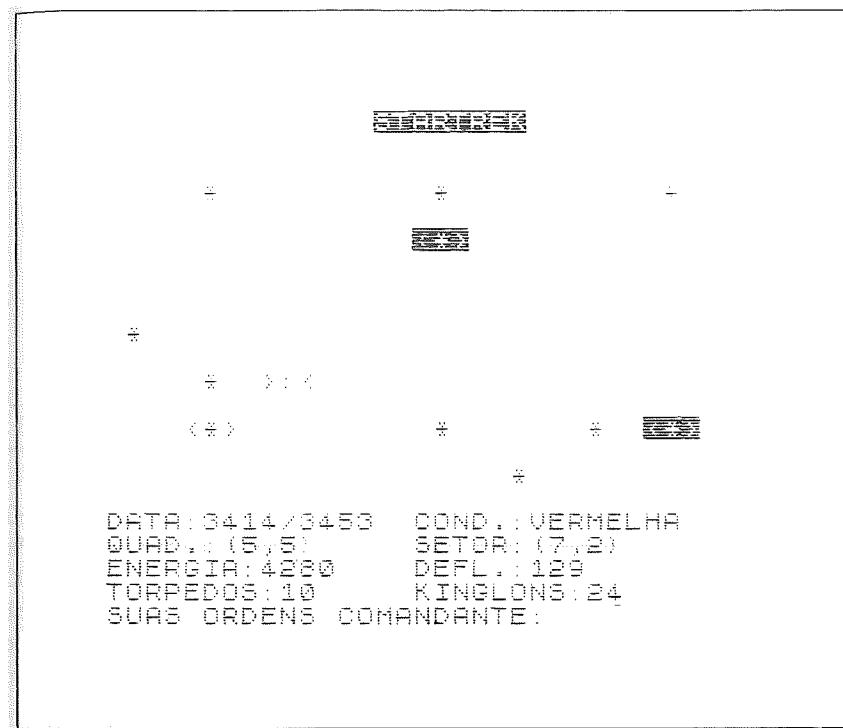
**Figura 14.2**  
**RETICULADO DAS DOBRAS ESPACIAIS**

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), tecle 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.

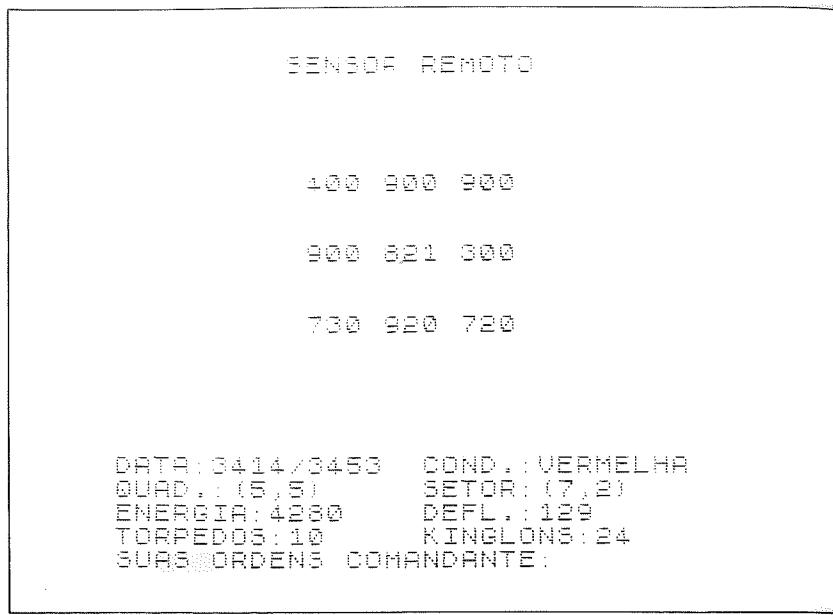


Figura 14.4  
TELA DO SENSOR

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 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.

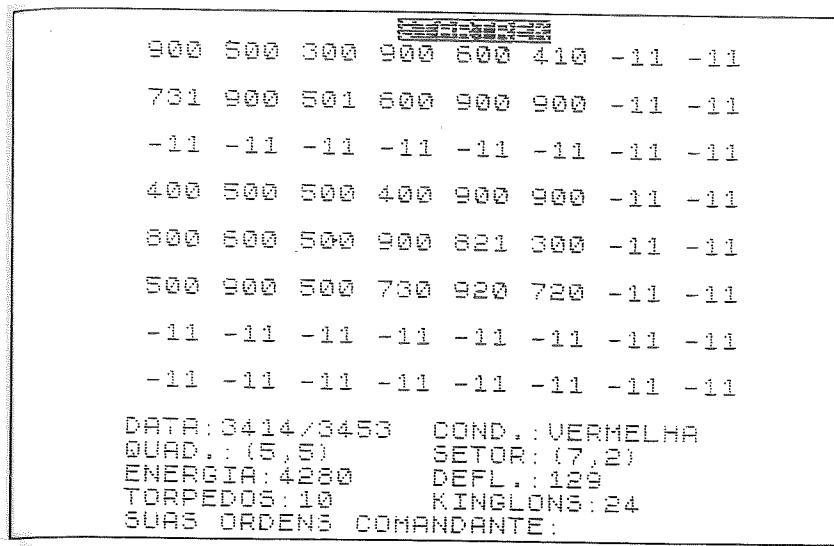


Figura 14.5  
RELATÓRIO DO COMPUTADOR

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.

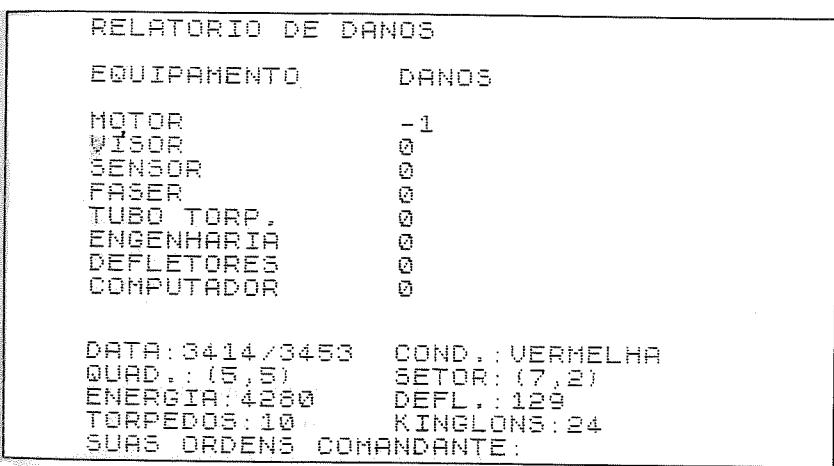


Figura 14.6  
RELATÓRIO DE DANOS/ENGENHARIA

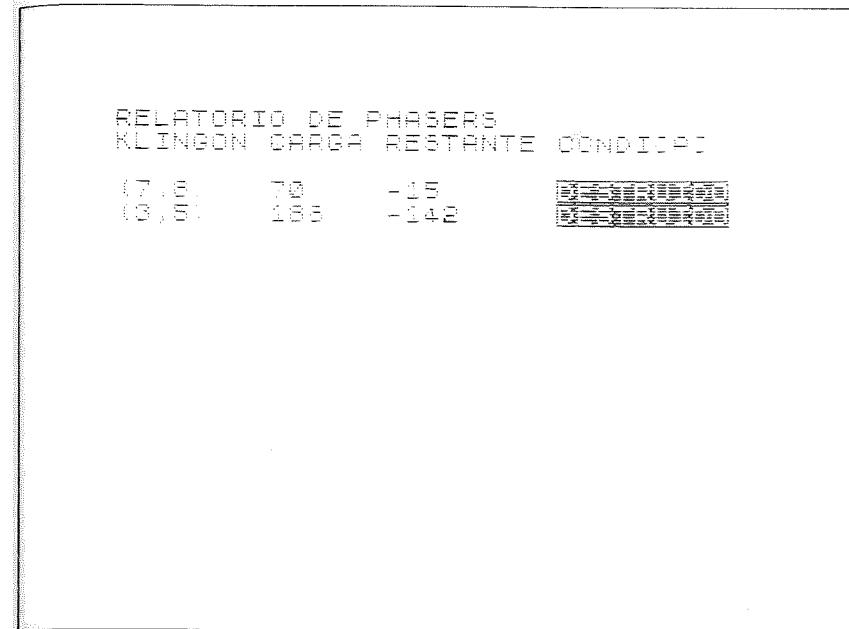
O uso de *PHASERS* automaticamente fecha o VISOR e o controle passa ao COMPUTADOR que apenas informa os efeitos (Figura 14.8), indicando a localização do KLINGON atingido, quanto de energia acertou o alvo e quanto as telas defletoras do inimigo ainda podem resistir. E se foi destruído. Como durante a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

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 conterá esta informação.

Ao receber um ataque Klingon, a USS ENTERPRISE tem seus DEFLETORES consumidos. Se a carga destes for inferior a 200 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



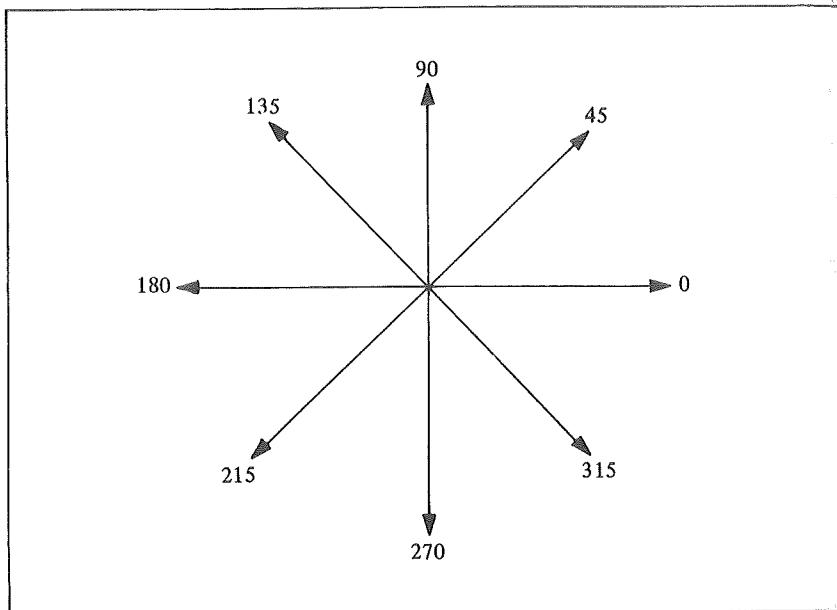
**Figura 14.8**  
**RELATÓRIO DE PHASERS**

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

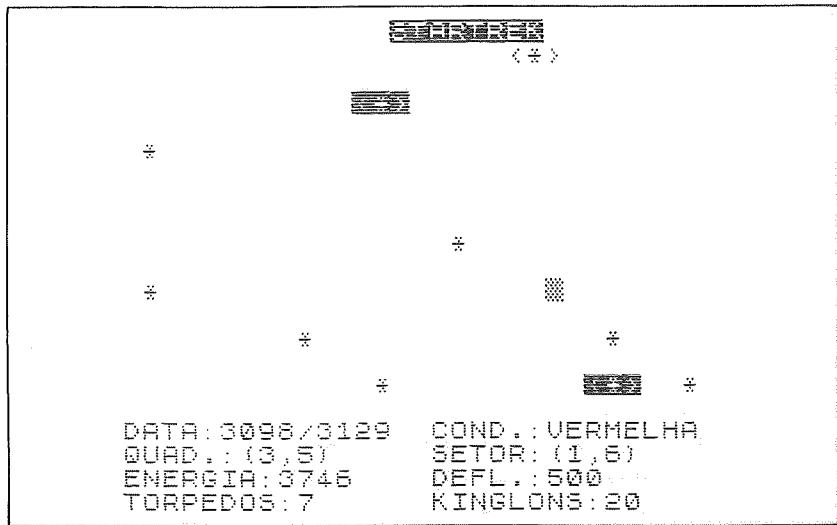
Se você se aproximar de uma BASE – visual no VISOR como < : > – então pode atracar (Figura 14.7). Estando atracado, todas as avarias são consertadas. Os TORPEDOS são renovados e os bancos de energia substituídos.

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.

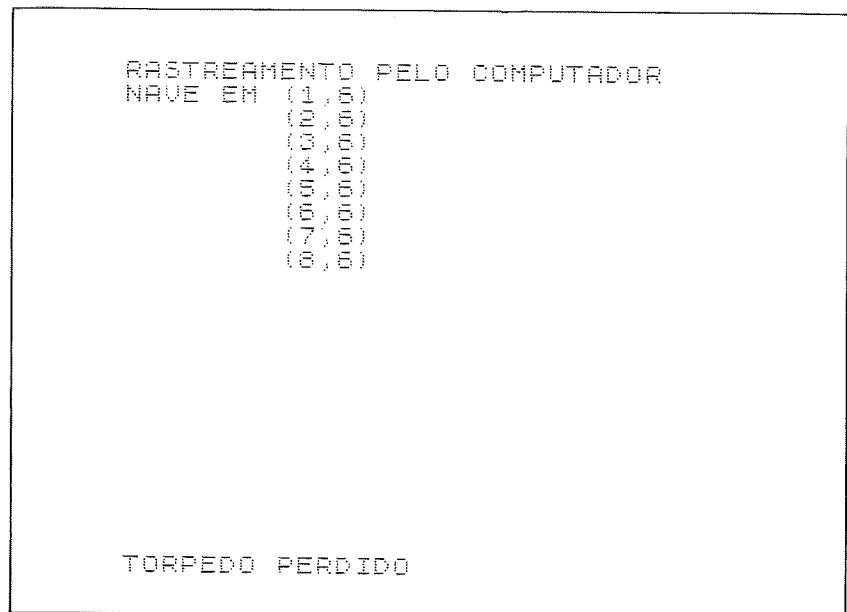


**Figura 14.9**  
**TABELA DE DIREÇÕES**



**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.

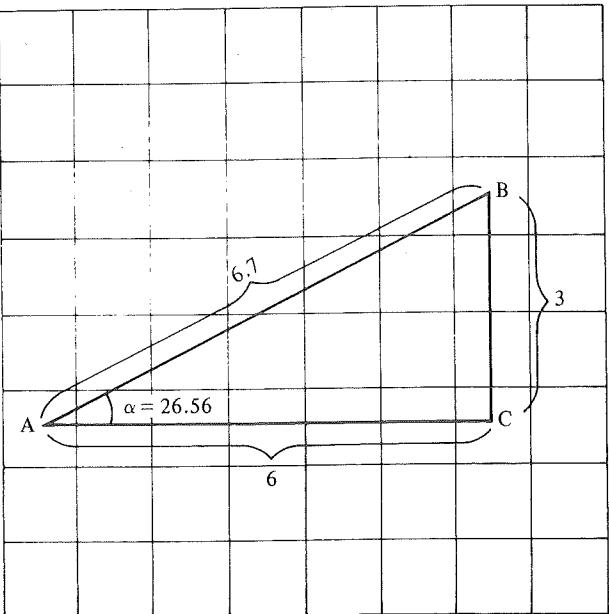


**Figura 14.11**  
**RASTREAMENTO DO TORPEDO PELO COMPUTADOR**

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 ARCTANGENTE.

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

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

A fórmula do CURSO nos levará a um valor entre  $\emptyset$  e  $90^\circ$  ( $90^\circ$  não, pois  $\text{TAN}(90^\circ)$  é indefinido). Dependendo da direção desejada, soma-se ou subtrai-se

a  $\emptyset$ ,  $90^\circ$ ,  $180^\circ$  ou  $270^\circ$  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, *você está correndo perigo*.
- 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 *abbreviatura 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, tecle 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  $G(8 \times 8) \rightarrow$  Galáxia e  $Q\$(8,32) \rightarrow$  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 utilizarmos 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 - TK82
20 REM ANTONIO JULIO LOSSIO BO
TELHO
30 FAST
40 DIM G(8,8)
50 DIM R(8)
60 DIM C(8)
70 DIM U(8)
80 DIM Q$(8,32)
90 DIM Z(3,3)
100 DIM K(8,8)
110 DIM D(8)
120 LET D#="MOTOR VISOR
SENSOR PASER TUBO TORP.E
ENGENHARIA DE FLETORES COMPUTADOR"
130 LET T=INT (RND*2000+2000)
140 LET TO=INT (RND*10+35)
150 LET TT=T+TO
160 LET DO=0
170 LET EO=5000
180 LET PO=10
190 LET SO=200
200 LET PR=PO
210 LET E=EO-S9
220 LET UI=0
230 LET QX=INT (RND*8+1)
240 LET QY=INT (RND*8+1)
250 LET SX=INT (RND*8+1)
260 LET SY=INT (RND*8+1)
270 LET S=S9
280 LET H=0
290 LET NE=0
300 LET K9=0
310 FOR I=1 TO 8
320 LET D(I)=0
330 LET R(I)=I
340 LET C(I)=I*4-3
350 LET U(I)=C(I)+2
360 FOR J=1 TO 8
370 LET K(I,J)=-11
```

Figura 14.14

LISTAGEM DO PROGRAMA-FONTE (AVVENTURA ESPACIAL)

```

330 LET R=RND
340 LET K3=0
350 IF R>.900 THEN LET K3=K3+1
360 IF R>.800 THEN LET K3=K3+1
370 IF R>.600 THEN LET K3=K3+1
380 LET K3=K3+KG
390 LET G0=INT (RND*7+3)
400 LET G(I,J)=G0*100+K3*10
410 NEXT J
420 NEXT I
430 LET BT=INT (RND*6+3)
440 FOR I=1 TO BT
450 LET X=INT (RND*8+1)
460 LET Y=INT (RND*8+1)
470 LET G(X,Y)=G(X,Y)+1
480 NEXT I
490 LET KO=K9
500 LET C$="VERDE"
510 CLS
520 PRINT "SUAS ORDENS CAP. KIR
K."
530 PRINT
540 PRINT "UMA FROTA KLIGON COM
POSTA DE"
550 PRINT K9;" NAVES INVADIU A
GALAXIA"
560 PRINT "E VOCE TEM ";T0;" AN
OS"
570 PRINT "ESTELARES PARA VENCE
-LOS."
580 PRINT
590 PRINT BT;" BASES ESTARAO A
SUAS ORDENS"
600 PRINT AT 21,0;"TECLE OO TEC
LA EXCETO ";KEY$"
610 SLOW
620 IF INKEY$="" THEN GOTO 910
630 FAST
640 CLS
650 GOTO 2730
660 FAST
670 PRINT AT 17,0;"DATA:";T;" /"
;TT
680 PRINT AT 17,16;"COND.:";C$
690 PRINT AT 18,0;"QUAD.: (";0X;
"/";BY;" )"
700 PRINT AT 18,16;"SETOR: (";SX
"/";SY;" )"
710 PRINT AT 19,0;"ENERGIA:";E;
720 PRINT AT 19,16;"DEFL.:";S;

```

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

```

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=A
1160 LET A$=A$(1 TO 5)
1170 IF A$="CURSO" (1 TO A) THEN
GOTO 2000
1180 IF A$="VISUA" (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$="ENGEN" (1 TO A) THEN
GOTO 5000
1240 IF A$="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 (AVVENTURA ESPACIAL) (cont.)

```

2050 PRINT AT 21,0;"DOBRA (1-6)
?
2070 INPUT V
2080 IF V<=0 THEN GOTO 1000
2090 IF V>8 THEN GOTO 2000
2100 IF D(1)>=0 OR V<=.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 C
OSMICA."
2285 PAUSE 200
2290 GOTO 2350
2295 IF D(R1)>=0 THEN GOTO 2350
2300 GOSUB 1500
2310 PRINT AT 21,0;"REPAROS ACEL
ERRADOS"
2320 PAUSE 200
2330 LET D(R1)=D(R1)+INT (RND*5+
1)
2340 IF D(R1)>0 THEN LET D(R1)=0
2350 PAST
2360 CLS
2370 LET UI=0
2380 LET Q$(SX,C(SY)) TO U(SY))=""
2390 LET X1=COS C1
2400 LET Y1=SIN C1
2410 LET X=SX
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
2515 IF T>TT THEN GOTO 7600
2520 GOTO 2400
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*X+S+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
2750 LET Q$(I)=""
2760 NEXT I
2770 LET Q$(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
2820 LET R$="">;<"

```

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

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

```

0030 LET I=0
0040 GOSUB 1000
0050 LET D(0)=0
0060 LET I=0
0070 GOSUB 1000
0080 LET D(1)=0
0090 LET I=0
0100 GOSUB 1000
0110 IF I=3 THEN GOTO 2490
0120 FOR J=I+1 TO 3
0130 LET Z(J,3)=-1
0140 NEXT J
0150 GOTO 2490
0160 IF I<1 THEN RETURN
0170 FOR J=1 TO I
0180 LET X=INT (RND*8+1)
0190 LET Y=INT (RND*8+1)
0200 IF Q$(X,C(Y)) TO U(Y)) <>
0210 " THEN GOTO 2940
0220 LET Q$(X,C(Y)) TO U(Y1)=B$
0230 IF A$<>" " THEN GOTO 2990
0240 LET Z(J,1)=X
0250 LET Z(J,2)=Y
0260 LET Z(J,3)=S9
0270 NEXT J
0280 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
3070 LET VI=0
3080 PRINT AT 0,9;"SENSOR REMOTO
"
3090 FOR I=0X-1 TO 0X+1
3100 FOR J=0Y-1 TO 0Y+1
3110 IF I<1 OR J<1 OR I>8 OR J>8
    THEN GOTO 3130
3120 LET K(I,J)=G(I,J)
3130 PRINT AT ((I-0X)*3+8),((J-0
    Y)*4+14);K(I,J),
3140 NEXT J
3150 NEXT I
3160 GOTO 1000
3170 FAST
3180 CLS
3190 LET VI=1
3200 PRINT AT 0,14;"XXXXXXXXXX"

```

Figura 14.14

*LISTAGEM DO PROGRAMA-FONTE (AVVENTURA ESPACIAL) (cont.)*

```

3300 FOR I=1 TO 8
3340 PRINT AT I*8-1,0,Q$(I);
3360 NEXT I
3370 RETURN
3400 IF D(2)>=0 THEN GOTO 3450
3410 GOSUB 1500
3430 SLOW
3430 PRINT AT 21,0;"VISOR DEFEIT
000"
3440 PAUSE 200
3450 GOTO 1100
3460 GOSUB 3300
3470 GOTO 1000
3500 IF D(8)>=0 THEN GOTO 3560
3510 GOSUB 1500
3530 SLOW
3530 PRINT AT 21,0;"COMPUTADOR D
EFEITO SO"
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
3600 PRINT AT I*8-1,0;
3610 FOR J=1 TO 8
3620 REM PRINT AT I*8-1,(J-1)*4;
3630 ;K(I,J);
3640 PRINT " ";K(I,J);
3650 NEXT J
3660 NEXT I
3670 GOTO 1000

4000 SLOW
4010 GOSUB 1500
4020 IF D(5)>=0 THEN GOTO 4060
4030 PRINT AT 21,0;"TUBO DE LAN
CIMENTO AVARIADO."
4040 PAUSE 200
4050 GOTO 1100
4060 IF P>0 THEN GOTO 4090
4070 PRINT AT 21,0;"VOCE NAO TEM
MAIS 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 (AVVENTURA ESPACIAL) (cont.)*

```

4150 LET P=P-1
4160 LET X=BX
4170 LET Y=BY
4180 IF D(2)>=0 AND VI=1 THEN GO
TO 4500
4181 CLS
4190 LET VI=0
4195 IF D(2)<0 THEN PRINT "VISOR
AVARIADO."
4200 IF D(3)>=0 THEN GOTO 4280
4205 GOSUB 1500
4210 PRINT "COMPUTADOR AVARIADO"
4215 PAUSE 100
4220 LET RE=4240
4230 GOTO 4800
4235 GOTO 4390
4260 GOTO 4350
4280 PRINT "RASTREAMENTO PELO CO
MPUTADOR"
4290 LET RE=4320
4300 PRINT "NAVE EM (";X;");";Y;""
1"
4310 GOTO 4800
4330 PRINT ("";R(X);");";)
R(Y);")"
4325 GOTO 4800
4330 PRINT AT 21,0;"TORPEDO PERD
IDO"
4332 GOTO 4380
4340 GOSUB 7700
4350 IF 0$(X,C(Y)) TO V(Y))<>":<
" THEN GOTO 4400
4360 PRINT AT 21,0;"VOCE DESTRUI
U UMA BASE."
4370 LET BT=BT-1
4375 LET G(QX,QY)=G(QX,QY)-1
4378 LET 0$(X,C(Y)) TO V(Y))=""
4380 PAUSE 100
4390 GOSUB 6500
4395 GOTO 1000
4400 IF 0$(X,C(Y)) TO V(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 (AVVENTURA ESPACIAL) (cont.)

```

4430 LET G(QX,QY)=G(QX,QY)-10
4440 GOTO 4370
4500 GOSUB 1500
4505 SLOW
4510 LET RE=4530
4520 GOTO 4800
4530 GOTO 4560
4540 GOTO 4330
4550 IF 0$(X,C(Y)) TO V(Y))="* "
THEN GOTO 4340
4552 PRINT AT 2*R(X)-1,(R(Y)-1)*
4553 "
4554 GOTO 4340
4560 PRINT AT X*2-1,Y*4-3;"#";
4570 PRINT AT X*2-1,Y*4-3;"#"
4580 GOTO 4800
4600 PRINT AT 21,0;"VOCE ACERTOU
UMA ESTRELA"
4610 GOTO 4380
4620 LET X=X+X1
4630 LET Y=Y+Y1
4640 IF X<1 OR X>8 OR Y<1 OR Y>8
THEN GOTO (RE+10)
4650 IF 0$(X,C(Y)) TO V(Y))="* "
THEN GOTO RE
4660 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 5130
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 S
EM COMUNICACAO."
5240 GOTO 5040
5250 LET VI=0

```

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

```

5060 CLS
5070 PRINT "RELATORIO DE DANOS"
5080 PRINT
5090 PRINT "EQUIPAMENTO"; TAB 16;
5100 PRINT "DANOS"
5110 PRINT
5120 FOR I=1 TO 8
5130 PRINT D$(I*10-9 TO I*10); TAB
16; D(I)
5140 NEXT I
5150 GOTO 1000
5160 SLOW
5170 FOR J=SY-1 TO SY+1
5180 FOR I=SX-1 TO SX+1
5190 IF I<1 OR J<1 OR I>8 OR J>8
THEN GOTO 5550
5200 IF Q$(I,C(J)) TO V(J)) = ">:<" THEN GOTO 5600
5210 NEXT I
5220 NEXT J
5230 GOSUB 1500
5240 PRINT AT 21,0; "APORTAR???" E
A BASE???""
5250 PAUSE 200
5260 GOTO 1100
5270 LET C$="ATRACADO"
5280 LET E=E0-59
5290 LET P=P0
5300 LET S=S9
5310 FOR I=1 TO 8
5320 LET D(I)=0
5330 NEXT I
5340 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
6100 GOSUB 1500
6110 PRINT AT 21,0; "PHASER APONT
ADO. ENERGIA?"
6120 INPUT X
6130 IF X<=0 THEN GOTO 1100

```

Figura 14.14

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

```

6130 IF X<=0 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
6190 IF Z(I,3)<0 THEN GOTO 6290
6200 LET X1=Z(I,1)
6210 LET Y1=Z(I,2)
6220 GOSUB 6900
6230 LET H=X/K3/(SQR H)*2*RND
6240 LET H=INT H
6250 LET Z(I,3)=Z(I,3)-H
6260 PRINT "(";X1;"/";Y1;")";TAB
8;H;TAB 14;Z(I,3);
6270 IF Z(I,3)>=0 THEN GOTO 6280
6280 PRINT TAB 23; "EXPLORACAO"
6290 LET Q$(X1,C(Y1)) TO V(Y1)) = "
6300 LET K3=K3-1
6310 LET G(OX,OY)=G(OX,OY)-10
6320 LET K9=K9-1
6330 IF K9<1 THEN GOTO 7400
6340 IF K3<1 THEN GOTO 6300
6350 PRINT ""
6360 NEXT I
6370 IF VI=0 THEN GOTO 6320
6380 PAUSE 100
6390 GOSUB 6500
6400 GOTO 3400

6500 LET C$="VERDE"
6510 IF K3<1 THEN RETURN
6520 LET O$="VERMELHA"
6530 FOR I=1 TO 3
6540 IF Z(I,3)<=0 THEN GOTO 6590
6550 LET X1=Z(I,1)
6560 LET Y1=Z(I,2)
6570 GOSUB 6900
6580 LET H=Z(I,3)/(SQR H)*2*RND
6590 LET X=X+INT H
6600 NEXT I
6610 SLOW
6620 GOSUB 7700
6630 PRINT AT 21,0; "ATAQUE KLING

```

Figura 14.14

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

```

ON "1,X;" UNIGO
6640 LET S=S-X
6650 PAUSE 100
6660 IF S>89 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 DESTRUÍDO
6740 PRINT
6750 PRINT "SUA NAVE VAI EXPLODIR"
6760 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 A 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 OS KLIN
GOS."
7420 PRINT "SUA MEDIA FOI DE ";K
8/T8
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
SE PENETROU NA GALAXIA COMO TEIM
DO E VOCE FOI INCAPAZ DE VENCE
-LOS."
7620 PRINT
7630 GOTO 7800

```

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

```

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

```

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

CAPÍTULO 15

***Uma Aventura  
Administrativa***

# A FICHA

---

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

# B MANUAL DE UTILIZAÇÃO

---

Você foi eleito administrador de uma ilha de 4000 km<sup>2</sup>, 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 km<sup>2</sup> 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 LISBIA BC
TELHO
90 REM D=DINHEIRO
90 REM P=PRESSORES
40 REM T=TERRAS
60 REM F=FLORESTA
60 REM E=EXTRANGEIROS
70 REM U=PRECO TERRAS
60 REM M=INDUSTRIAS
80 REM H=PLANTACAO
90 LET M=0
95 LET F=20000
100 SLOW
110 LET D=500000+10000*RND-1000
*RNDO
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$"
";D;" EM CAIXA"
170 SCROLL
180 PRINT AT 20,0;"SAO ";P;" SO
CIOES COPROPRIETARIOS"
190 SCROLL
200 IF E<1 THEN GOTO 230
210 PRINT AT 20,0;"E ";E;" TRAB
ALHADORES EMPREGADOS"
220 SCROLL
230 PRINT AT 20,0;"INDUSTRIAS
DESEJAM INVESTIR NA ILHA"
240 SCROLL
250 PRINT AT 20,0;"PARA ISTO PR
ECISAM COMPRAR TERRAS"
260 SCROLL
270 PRINT AT 20,0;"ESTAO OFEREC
ENDO US$";W;" POR KM2"
280 SCROLL
290 SCROLL
300 PRINT AT 20,0;"QUANTO VOCE
VENDE A ELES ?"
310 INPUT U
320 PRINT U
330 SCROLL
340 IF U<0 THEN GOTO 290
350 IF U<T THEN GOTO 370
360 PRINT AT 20,0;"VOCE SO TEM
";U;" KM2 DE TERRAS"

```

**Figura 15.1 LISTAGEM DO PROGRAMA-FONTE  
(AVVENTURA 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
";P;" 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$";U/10;" POR KM2 "
540 SCROLL
545 SCROLL
550 PRINT AT 20,0;" E VOCE NAO
TEM DINHEIRO PARA ISSO ?"
555 SCROLL
560 GOTO 440
570 LET D=D-U*U/10
580 LET T=T+U
590 LET F=F-U
595 LET H=0
600 IF T<1 OR D<U/10 THEN GOTO
610
615 SCROLL
620 PRINT AT 20,0;"QUANTO VOCE
QUER PLANTAR ?"
630 INPUT H
640 PRINT H
645 SCROLL
650 IF H<1 THEN GOTO 620
660 IF H*U/100<=D THEN GOTO 710
670 PRINT AT 20,0;"O CUSTO DA S
EMENTE E US$";U/100;" POR KM2"
680 SCROLL
685 SCROLL

```

**Figura 15.1 LISTAGEM DO PROGRAMA-FONTE  
(AVVENTURA 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
OCULTIVO DE 1 KM2"
760 GOTO 610
810 LET D=D-H*RND/100
820 IF H<1 OR RND>.2 THEN GOTO
830
830 LET U=RND*D
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
900
900 PRINT AT 20,0;"O DESMATAMENT
O 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<D THEN GOTO 990
970 PRINT AT 20,0;"VOCE SO TEH
U$";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

```

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

```

1070 PRINT AT 20,0;Z;" ADOECERAM
1075 MORTANTO"
1080 SCROLL
1085 LET U=U*RND+U
1090 IF U>D THEN LET U=D
1100 PRINT AT 20,0;"SUSP DESPESA
1105 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
1150 LET E=INT ((F/2+F*RND1/2))
1160 IF E<1 THEN GOTO 1300
1170 SCROLL
1180 PRINT AT 20,0;E;" TURISTAS
1185 VISTARAM A ILHA,"
1190 LET U=E*(100*RND+100)
1200 LET D=D+U
1205 SCROLL
1210 PRINT AT 20,0;"E RENDERAM U
$";U
1220 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
1330 NASCERAM,"
1340 SCROLL
1345 SCROLL
1350 LET P=P+U
1360 GOTO 1700
1370 SCROLL
1410 PRINT AT 20,0;"VAMOS AGORA
A ASSEMBLEA GERAL."
1415 SCROLL
1420 PRINT AT 20,0;"FEITO SEUS C
ALCULOS, QUAL O VALOR A SER
DISTRIBUIDO AOS"
1425 SCROLL
1430 SCROLL
1435 PRINT AT 20,0;"CO-PROPRIETA
RIO?!"
1440 SCROLL
1445 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  
(AVVENTURA ADMINISTRATIVA) (cont.)

```

1490 PRINT AT 20,0; "VOCE SO TE"
1500 ;D
1505 GOTO 1435
1510 IF U/P>100 THEN GOTO 1600
1515 SCROLL
1520 PRINT AT 20,0; "OS RENDIMENT
1525 OS MINIMOS ACERTADOS SÃO DE US$10
1530 E VOCE NAO PODE"
1535 SCROLL
1540 SCROLL
1545 PRINT AT 20,0; "PAGAR SO US$"
1550 ;U/P
1555 SCROLL
1560 PRINT AT 20,0; "OU MORREM TO
1565 DOIS."
1570 IF D/P<100 THEN GOTO 5000
1575 GOTO 1435
1580 IF U/P>200 THEN GOTO 1700
1585 SCROLL
1590 PRINT AT 20,0; "A POPULACAO
1595 NAO ESTA SATISFEITA COM O SEU DE
1600 SERPENHO."
1605 SCROLL
1610 SCROLL
1615 GOTO 1760
1620 SCROLL
1625 PRINT AT 20,0; "UM ANO SE PA
1630 E A"
1635 SCROLL
1640 PRINT AT 20,0; "COLHEITA REN
1645 DEU US$";H*N/3
1650 SCROLL
1655 LET D=D+H*N/3
1660 GOTO 1910
1665 PRINT AT 20,0; "VAMOS AO PRO
1670 XIMO ANO?"
1675 SCROLL
1680 PRINT AT 20,0; "ENTRE 0 PARA
1685 CONTINUAR AGORA, 1 PARA
1690 CONTINUAR DEPOIS"
1695 SCROLL
1700 SCROLL
1705 PRINT AT 20,0; "ENTRE 0 PARA
1710 DESESTIR:"
1715 INPUT U
1720 PRINT U
1725 SCROLL
1730 IF U<0 OR U>2 THEN GOTO 178
1735
1740 LET U=INT U*100+2000
1745 GOTO U

```

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

```

1910 PRINT AT 20,0; "TAXACAO SOBR
1915 E A INDUSTRIA RENDEU US$";
1920 M*N/5
1925 SCROLL
1930 SCROLL
1935 LET D=D+M*N/5
1940 GOTO 1400
1945 GOTO 150
1950 SCROLL
1955 PRINT AT 20,0; "POSICIONE A
1960 FITA NO GRAVADOR. TECLE 0 QUAN
1965 DO ESTIVER OK."
1970 INPUT M
1975 SCROLL
1980 SCROLL
1985 SCROLL
1990 PRINT AT 20,0; "QUER PARAR ?"
1995 ;SIM"
2000 INPUT M
2005 IF M=0 THEN GOTO 5160
2010 SCROLL
2015 GOTO 150
2020 SCROLL
2025 SCROLL
2030 SCROLL
2035 PRINT AT 20,0; "COMOCAO GERA
2040 L"
2045 SCROLL
2050 PRINT AT 20,0; "COM UM SALDO
2055 DE US$";D
2060 SCROLL
2065 PRINT AT 20,0; "VOCE SO PODE
2070 PAGAR US$";D/P
2075 SCROLL
2080 PRINT AT 20,0; "PARA CADA UM
2085 DOS"
2090 ;P
2095 SCROLL
2100 PRINT AT 20,0; "HABITANTES I
2105 DIOTAS DESTA ILHA."
2110 SCROLL
2115 PRINT AT 20,0; "VOCE FOI DES
2120 TITUIDO E PRESO"
2125 SCROLL
2130 PRINT AT 20,0; "POR MALUVERA
2135 O DOS BENS COMUNS."
2140 FAST
2145 STOP
2150 CLEAR
2155 FAST
2160 SAVE "ICOOP"
2165 RUN

```

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

CAPÍTULO 16

*Apaixonante e  
Tradicional  
Batalha Naval*

**A**

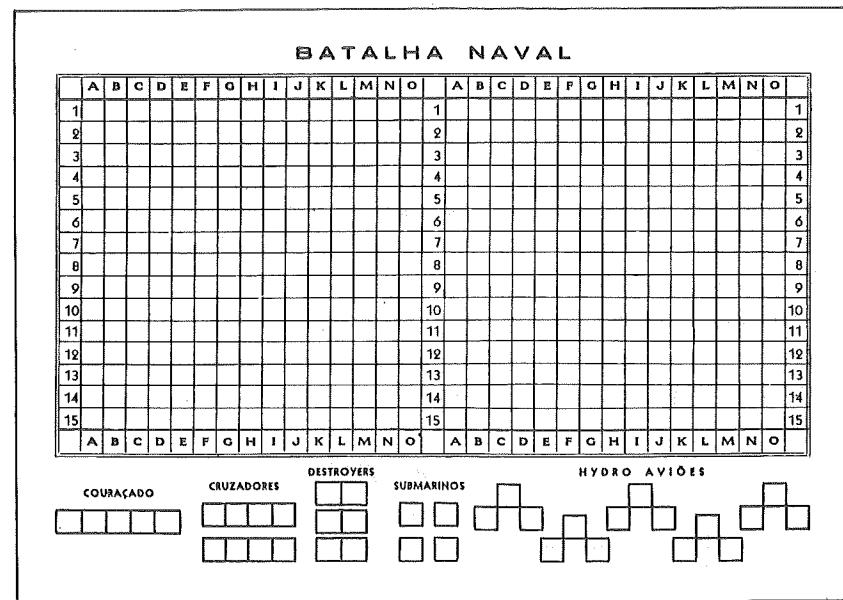
## FICHA

a) Tipo:	b) Nome:	c) Versão:	d) Revisão:	e) Data:	
Programa	<input checked="" type="checkbox"/> BATALHA NAVAL	1	Ø	22/10/83	
Sub-rotina					
Função					
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"/> CASSETE	<input checked="" type="checkbox"/>	COPY	<input type="checkbox"/>	
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/> DISQUETE	<input type="checkbox"/>	DLOAD	<input type="checkbox"/>	
TIMEX 10ØØ	<input checked="" type="checkbox"/> DISCO: 8"	<input type="checkbox"/>	DSAVE	<input type="checkbox"/>	
CP-20Ø	<input checked="" type="checkbox"/> 5½ "	<input type="checkbox"/>	DMLOAD	<input type="checkbox"/>	
CP-30Ø	<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"/>			
<b>RINGO R - 47Ø e AS - 1ØØØ</b>					
k) Observações:					

**B**

## MANUAL DE UTILIZAÇÃO

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 distribui 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 destroiers (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.

AS ESQUADRAS SERAO COMPOSTAS  
COMO SE SEGUIM:

NAUJO	VOCE	COMP.
COURACADOS	140100-140100	
FREGATAS		
AUVIORES		
DESTRÓIER		
DESTRUÍDORES		

SE VOCÊ QUISER TROCAR EU TOPO  
COMO É? TROCA?

**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.

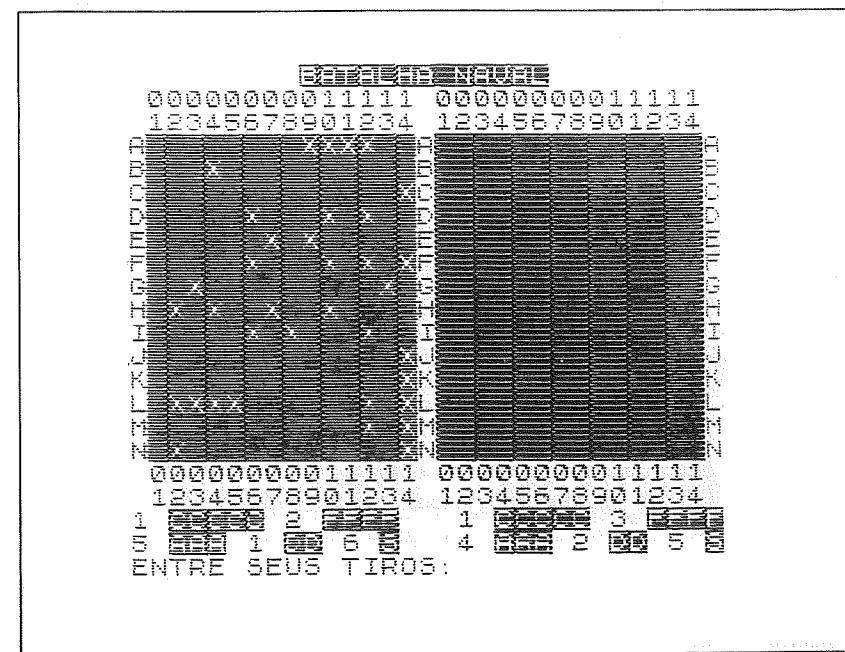
Neste ponto é dado ao adversário o direito de trocar de esquadra, pois as mesmas não são iguais.

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 complementará 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.

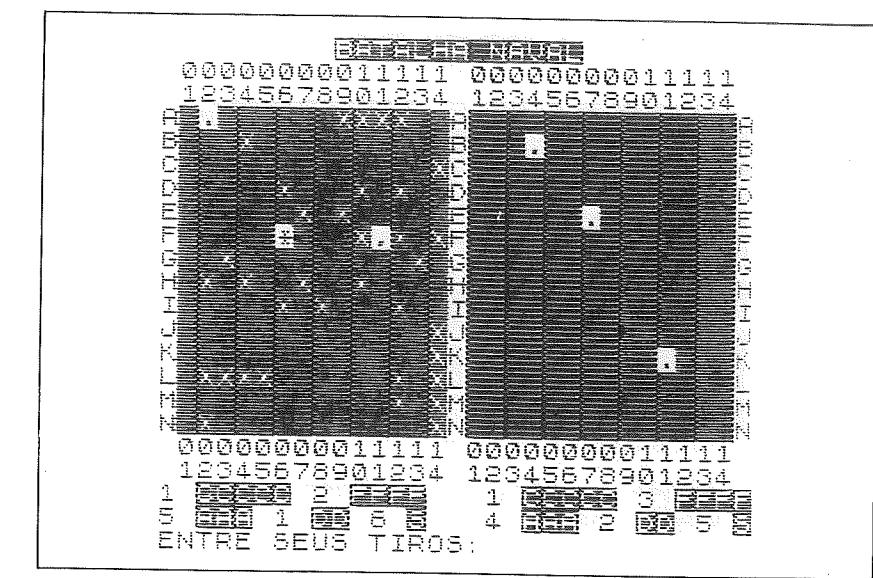


Figura 16.4  
RESPOSTA A TB04K11E07

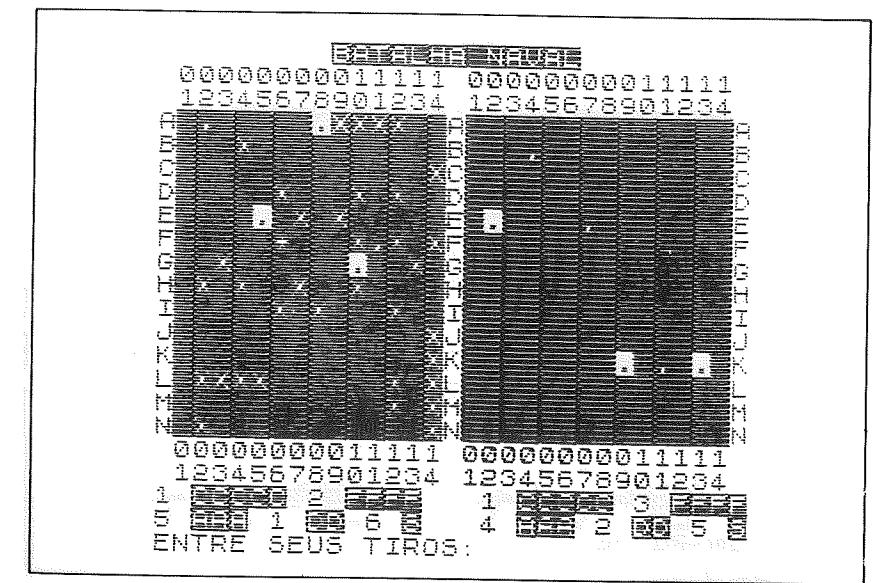
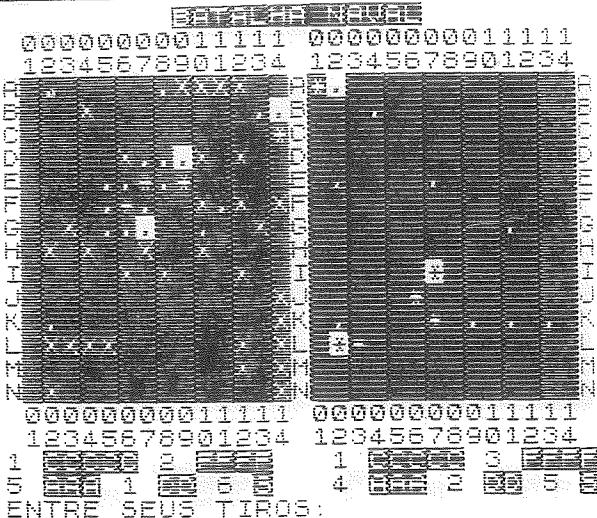
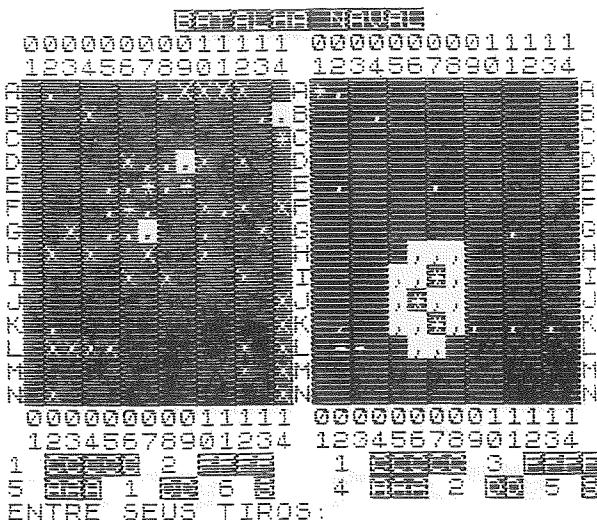


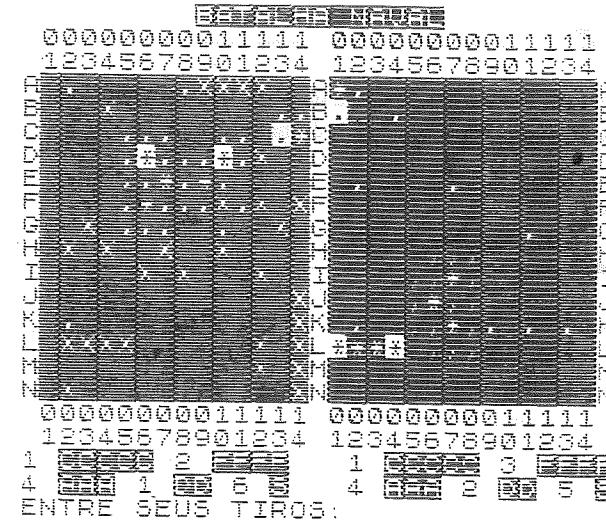
Figura 16.5  
RESPOSTA SEGUINTE À FIGURA ANTERIOR



**Figura 16.6**  
*MARCAÇÃO DE ÁGUA – ANTES*



**Figura 16.7**  
*MARCAÇÃO DE ÁGUA*



**Figura 16.8**  
*MARCAÇÃO DE ÁGUA – APÓS*

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.

## C MANUAL DE OPERAÇÃO

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.

## D MANUAL TÉCNICO

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 600 – 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 avaria, 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 cercando 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  $\pm 15$ , a outra parte do avião, só pode estar em  $I \pm 17$  ou  $J \pm 17$ . Se a diferença é  $\pm 17$ , então a última parte somente pode ser encontrada em  $I \pm 15$  e  $J \pm 15$ .

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

```

1000 LET UD=INT ((RND*4+1))
1010 LET CD=INT ((RND*4+1))
1020 LET UD=7-UD
1030 LET CD=7-CD
1040 LET UD=INT ((RND*4+1))
1050 LET CD=INT ((RND*4+1))
1060 LET UD=7-UD
1070 LET CD=7-CD
1080 LET UD=INT ((RND*4+1))
1090 LET CD=INT ((RND*4+1))
1100 LET UD=7-UD
1110 LET CD=7-CD
1120 LET UD=INT ((RND*4+1))
1130 LET CD=INT ((RND*4+1))
1140 LET UD=7-UD
1150 LET CD=7-CD
1160 FOR I=1 TO 256 STEP 16
1170 LET T#(I) TO I+15) =P#
1180 NEXT
1190 RETURN
1200 LET USUB=100
1210 LET U$=T#
1220 LET USUB=800
1230 LET U$=T#
1240 LET UD=INT ((RND*2+1))
1250 LET CD=INT ((RND*2+1))
1260 LET UD=INT ((RND*2+1))
1270 LET CD=INT ((RND*2+1))
1280 LET UD=UD-UD-U#
1290 LET CD=CD-CD-U#

```

**Figura 16.11** LISTAGEM DO PROGRAMA-FONTE (BATALHA NAVAL)

```

1300 LET UD=INT ((RND*4+1))
1310 LET CD=INT ((RND*4+1))
1320 LET UD=7-UD
1330 LET CD=7-CD
1340 LET UD=INT ((RND*4+1))
1350 LET CD=INT ((RND*4+1))
1360 LET UD=7-UD
1370 LET CD=7-CD
1380 LET UD=INT ((RND*4+1))
1390 LET CD=INT ((RND*4+1))
1400 LET UD=7-UD
1410 LET CD=7-CD
1420 LET UD=INT ((RND*4+1))
1430 LET CD=INT ((RND*4+1))
1440 LET UD=7-UD
1450 LET CD=7-CD
1460 LET UD=INT ((RND*4+1))
1470 LET CD=INT ((RND*4+1))
1480 LET UD=7-UD
1490 LET CD=7-CD
1500 LET UD=INT ((RND*4+1))
1510 LET CD=INT ((RND*4+1))
1520 LET UD=7-UD
1530 LET CD=7-CD
1540 LET UD=INT ((RND*4+1))
1550 LET CD=INT ((RND*4+1))
1560 LET UD=7-UD
1570 LET CD=7-CD
1580 LET UD=INT ((RND*4+1))
1590 LET CD=INT ((RND*4+1))
1600 LET UD=7-UD
1610 LET CD=7-CD
1620 LET UD=INT ((RND*4+1))
1630 LET CD=INT ((RND*4+1))
1640 LET UD=7-UD
1650 LET CD=7-CD
1660 LET UD=INT ((RND*4+1))
1670 LET CD=INT ((RND*4+1))
1680 LET UD=7-UD
1690 LET CD=7-CD
1700 LET UD=INT ((RND*4+1))
1710 LET CD=INT ((RND*4+1))
1720 LET UD=7-UD
1730 LET CD=INT ((RND*4+1))
1740 LET UD=UD-UD-U#
1750 LET CD=CD-CD-U#
1760 RETURN
1770 LET UP=UD*5+UF*4+UA*3+UD*2+
1780 LET CP=CD*5+CF*4+CA*3+CD*2+
1790 CLS
1800 PRINT "AS EQUADRADAS SERAO AO
1810 COMPOSTAS COMO SE SEGUISE:"
1820 PRINT "NAOVO", "VEO COMP."
1830 PRINT "COURACADOS", " "; UC
1840 ";"; UC
1850 PRINT "FREGATAS", " "; UF;
1860 PRINT "AVIOES", " "; UA;
1870 PRINT "DESTROIERS", " "; UD;
1880 PRINT "SUBMARINOS", " "; US;
1890 PRINT "SE VOCE QUISER TROCAR
1900 TOPON"
1910 PRINT "COMO E? TROCA?"
1920 INPUT A#
1930 IF A$="S" OR A$="SIM" THEN
1940 GOTO 1960
1950 IF A$="N" OR A$="NAO" THEN
1960 GOTO 1980
1970 SCROLL

```

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

```

500 PRINT "VOCE DEVE RESPONDER"
510 OU_NAO
520 SCROLL
530 GOTO 500
540 CLS
550 PRINT TAB(9) "BATALHA NAVAL"
560 DOOR
570 FOR I=1 TO 15
580 LET J1=I*16-14
590 LET J2=J1+13
600 LET A$=TAB(I TO I)
610 PRINT TAB(8);A$(J1 TO J2)
620 NEXT I
630 PRINT " 00000000011111 0000
640 11111 12345678901234 1234
650 1234"
660 RETURN
670 LET NU=1
680 LET NO=0
690 LET U=CA
700 GOSUB 8000
710 LET NU=0
720 LET U=CD
730 GOSUB 8000
740 LET NU=1
750 LET U=CB
760 GOSUB 8000
770 LET NU=0
780 LET U=CC
790 GOSUB 8000
800 LET NU=1
810 LET U=CF
820 GOSUB 8000
830 FOR I=1 TO 15
840 LET T$(I TO I)=" "
850 LET M=87-I
860 LET TM=M TO M=" "
870 LET TH=I*16
880 LET TM=(M TO M)=" "
890 LET M=M+1
900 LET TM=(M TO M)=" "
910 NEXT M
920 RETURN
930 GOSUB 250
940 GOTO 400
950 GOSUB 250
960 GOSUB 710
970 LET U=T$
980 GOSUB 160
990 GOSUB 250

```

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

```

1140 GOSUB 710
1150 GOSUB 1600
1160 GOSUB 00900
1170 GOSUB 00700
1180 GOSUB 00900
1190 PRINT AT 21,0;"ENTRE SEUS T
1200 INPUT A$
1210 IF (LEN A$)<1 THEN GOTO 120
1220 LET E=A$&A$(1 TO 1)
1230 LET E=E&" "
1240 LET E=E&" "
1250 LET E=E&" "
1260 GOSUB 9900
1270 PRINT AT 21,0;"TIRO OU AGUA
1280 PAUSE 300
1290 GOTO 1030
1300 FOR I=1 TO TT
1310 LET O$(X(I) TO X(I))=" "
1320 NEXT I
1330 GOTO 1160
1340 FOR I=1 TO TT
1350 IF T$(X(I) TO X(I))=" " THE
N GOTO 1340
1360 LET O$(X(I) TO X(I))="*"
1370 LET O=O-1
1380 GOTO 1350
1390 LET O$(X(I) TO X(I))=" "
1400 NEXT I
1410 GOTO 3000
1420 GOSUB 0000
1430 IF (LEN A$)=0 THEN GOTO 160
1440 GOTO 1510
1450 GOSUB 0000
1460 GOSUB 0100
1470 GOSUB 00300
1480 IF U=0 THEN GOTO 1540
1490 LET TT=0
1500 GOTO 1160
1510 IF F$="0" THEN GOTO 1280
1520 IF TT<3 THEN GOTO 1600
1530 GOSUB 00900
1540 PRINT AT 21,0;"NO MAXIMO 3
TIROS."
1550 PAUSE 300
1560 GOTO 1210
1570 LET E=1690
1580 FOR U=TT+1 TO 3

```

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

```

1620 GOSUB 6500
1630 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
1700 ERRADAMENTE"
1710 PAUSE 300
1710 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
2105 LET U=TT+1
2106 IF U>14 THEN GOTO 2400
2110 LET K$=A$(1 TO 1)
2115 FOR I=2 TO 15
2130 IF K$=0$(I TO I) THEN GOTO
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
2230 LET J1=I*2+1
2235 LET J2=J1+1
2238 IF K$=U$(J1 TO J2) THEN GOT
0 2260
2240 NEXT I
2250 GOTO 2150

```

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

```

0260 LET X(U)=X(U)+16+I+1
0270 LET TT=U
0280 LET A$=A$(4 TO )
0290 IF (LEN A$)<0 THEN GOTO 21
0300
0305 RETURN
0310 LET U=0
0315 FOR I=1 TO TT
0320 IF C$(X(I) TO X(I))="■" THE
N GOTO 2390
0330 IF C$(X(I) TO X(I))="■" THE
N GOTO 2390
0340 IF C$(X(I) TO X(I))="," THE
N GOTO 2390
0350 GOSUB 9900
0355 LET M=INT (X(I)/16)
0360 LET N1=X(I)-M*16
0365 LET M=M+1
0370 LET N2=N1+1
0375 PRINT AT 21,0;"VOCE JA HAVI
A ATIRADO EM ";Q$(M TO M);U$(N1
TO N2)
0378 PAUSE 300
0380 LET U=1
0390 NEXT I
0395 RETURN
0400 GOSUB 9900
2410 PRINT AT 21,0;"NO MAXIMO 14
MARCACAO DE AGUA"
2420 PAUSE 300
2430 RETURN

3000 GOSUB 6700
3005 GOSUB 6300
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 LR=3070
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 3780

```

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

```

3160 LET O=O+1
3170 IF O>20 THEN GOTO 3600
3180 LET L(O)=M
3190 LET T(O)=S
3200 LET D(O)=0
3210 NEXT K
3220 GOTO 1180
3230 IF G<1 THEN RETURN
3240 FOR I=1 TO G
3250 IF CODE(U$(Y(I) TO ))>=128
    THEN GOTO 3330
3260 LET U$(Y(I) TO Y(I))=CHR$(
    CODE(U$(Y(I) TO ))+128)
3270 NEXT I
3280 LET G=0
3290 RETURN
3300 GOSUB 9900
3310 PRINT AT 21,0;"ERRO DE LOGI
CA RARO. ABORTO"
3320 STOP
3330 GOTO 3400
3340 RETURN
3350 GOSUB 9900
3360 PRINT AT 21,0;"NUMERO DE OB
JETOS >20"
3370 STOP
3380 GOTO 3600
3390 LET A$="."
3400 GOTO 3730
3410 LET A$="*"
3420 LET U$(M TO H)<>"." AND U$(H
    TO M)<>"*" THEN STOP
3430 LET UP=UP-1
3440 LET G=G+1
3450 LET Y(G)=M
3460 LET AL=AL+1
3470 LET U$(M TO H)=A$
3480 RETURN

4000 LET M=L(K)
4010 LET M1=L(K)-D(K)
4020 LET J=15
4030 IF ABS(D(K))=17 THEN GOTO 40
4040 LET J=17
4050 LET I=M-J
4060 LET D=1
4070 GOTO 4241
4080 LET I=M+J
4090 LET D=2
4100 GOTO 4241
4110 LET I=M1-J

```

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

```

4120 LET D=3
4130 GOTO 4241
4140 LET I=M1+J
4150 LET D=4
4160 GOTO 4241
4170 GOSUB 4450
4180 LET A$=U$(I TO I)
4190 IF A$="." OR A$="*" THEN GO
    TO 4260
4200 IF A$="." OR A$="," THEN GO
    TO (4080+D*20)
4210 LET M=I
4220 IF A$="." THEN GOTO 3700
4230 GOSUB 3720
4240 LET T(K)=1
4250 LET UA=UA-1
4260 IF UA<0 THEN LET UA=0
4270 FOR J=1 TO 9
4280 LET I=I+J+M
4290 IF U$(I TO I)="." THEN LET
    U$(I TO I)=""
4300 NEXT J
4310 RETURN
4320 GOSUB 9900
4330 PRINT AT 21,0;"QUE AVIADO E
ESTE?"/>
4340 STOP
4350 GOTO 4450

4460 IF D(K)=2 THEN GOTO 4700
4470 LET I=L(K)
4480 LET M1=I+5
4490 FOR M=I TO M1
4500 LET A$=U$(M TO M)
4510 IF A$="." THEN GOTO 4570
4520 IF A$="*" THEN GOTO 4600
4530 IF A$="," OR A$="'" THEN GO
    TO 4590
4540 LET L(K)=M
4550 NEXT M
4560 LET D(K)=2
4570 RETURN
4580 GOSUB 3720
4590 IF D(K)=2 THEN GOTO 4640
4600 LET M=M-15
4610 GOTO 4650
4620 LET M=M-17
4630 LET U$(M TO M)=""
4640 LET M=M+32
4650 LET U$(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 4690
4680 GOTO 4740
4685 LET I=L(K)
4690 LET M1=I-10
4695 FOR M=I TO M1 STEP -1
4700 LET R$=U$(M TO M)
4705 IF R$="" THEN GOTO 4670
4710 IF R$="" THEN GOTO 4600
4715 IF R$="" OR R$=".," THEN GO
TO 4740
4720 LET L(K)=M
4725 NEXT M
4730 LET T(K)=1
4735 LET M=L(K)
4740 LET M=M-1
4745 IF U$(M TO M) = "■" OR U$(M T
O M) = "*" THEN GOTO 4750
4750 LET M=M+1
4755 LET M=M+6
4760 LET D=0
4765 FOR I=M TO M1
4770 IF U$(I TO I) <> "■" AND U$(I
TO I) <> "*" THEN GOTO 4840
4775 LET D=D+1
4780 FOR C=1 TO 0
4785 IF T(C)=1 THEN GOTO 4820
4790 IF L(C)<>I THEN GOTO 4820
4795 LET T(C)=1
4800 NEXT C
4805 NEXT I
4810 GOTO 6000

5000 IF D(K)=2 THEN GOTO 5200
5010 LET I=L(K)
5015 LET M1=I+60
5020 FOR M=I TO M1 STEP 16
5025 LET R$=U$(M TO M)
5030 IF R$="" THEN GOTO 5170
5035 IF R$="" OR R$=".," THEN GO
TO 5020
5040 LET L(K)=M
5045 NEXT M
5050 LET D(K)=2
5055 RETURN
5100 GOSUB 3790
5105 IF D(K)=2 THEN GOTO 5140
5110 LET M=M+17
5115 GOTO 5150
5120 LET M=M-15
5125 LET U$(M TO M) = "■"

```

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

```

5150 LET M=M-2
5155 LET U$(M TO M) = "■"
5160 RETURN
5170 GOSUB 3790
5175 LET M=I
5180 IF D(K) <> 2 THEN GOTO 5090
5185 GOTO 5240
5190 LET I=L(K)
5195 LET M1=I-80
5200 FOR M=I TO M1 STEP -16
5205 LET R$=U$(M TO M)
5210 IF R$="" THEN GOTO 5170
5215 IF R$="" OR R$=".," THEN GO
TO 5140
5220 LET L(K)=M
5225 NEXT M
5230 LET T(K)=1
5235 LET M=L(K)
5240 LET M=M-16
5245 IF U$(M TO M) = "■" OR U$(M T
O M) = "*" THEN GOTO 5250
5250 LET M=M+16
5255 LET M=M+80
5260 LET D=0
5265 FOR I=M TO M1 STEP 16
5270 IF U$(I TO I) <> "■" AND U$(I
TO I) <> "*" THEN GOTO 5340
5275 LET D=D+1
5280 FOR C=1 TO 0
5285 IF T(C)=1 THEN GOTO 5320
5290 IF L(C)<>I THEN GOTO 5320
5295 LET T(C)=1
5300 NEXT C
5305 NEXT I
5310 GOTO 6000

5320 LET C=0
5325 FOR I=1 TO 9
5330 LET J=L(K)+R(I)
5335 LET R$=U$(J TO J)
5340 IF R$="" OR R$=".," THEN GO
TO 5320
5345 IF R$ <> "■" AND R$ <> "*" THEN
GOTO 5500
5350 LET C=C+1
5355 NEXT I
5360 IF C>1 THEN GOTO 5575
5365 LET US=US-1
5370 IF US<0 THEN LET US=0
5375 LET T(K)=1
5380 RETURN

```

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

```

10 LET M=0
20 IF A$="" THEN GOTO 3700
30 GOSUB 5700
40 LET M=LBB-(L(K)-J)
50 IF M>16 THEN GOTO 5750
60 LET T(K)=4
70 LET H=L(K)
80 LET M=H-16
90 LET D$=U$(H TO M)
100 IF D$="" OR D$="*" THEN GO
110 TO 5700
120 LET U$(M-1 TO M+1)="
130 LET U$(M+1 TO M+1)="
140 LET M=M+16
150 LET D$=U$(M TO M)
160 IF D$="" OR D$="*" THEN GO
170 TO 5700
180 LET U$(M+1 TO M+1)="
190 LET U$(M-1 TO M-1)="
200 RETURN
210 IF M>16 THEN GOTO 5650
220 LET T(K)=3
230 LET H=L(K)
240 LET M=M-1
250 LET D$=U$(M TO M)
260 IF D$="" OR D$="*" THEN GO
270 TO 5700
280 LET U$(M-16 TO M-16)="
290 LET U$(M+16 TO M+16)="
300 LET M=M+1
310 LET D$=U$(M TO M)
320 IF D$="" OR D$="*" THEN GO
330 TO 5700
340 LET U$(M-16 TO M-16)="
350 LET U$(M+16 TO M+16)="
360 RETURN
370 LET T(K)=2
380 LET D$=L(K)-J
390 LET H=L(K)
400 LET D$(I-1 TO I-1)="
410 LET D$(I+1 TO I+1)="
420 LET D$(J-1 TO J-1)="
430 LET D$(J+1 TO J+1)="
440 LET D$(J-16 TO J-16)="
450 LET D$(J+16 TO J+16)="
460 LET D$(I-16 TO I-16)="
470 LET D$(I+16 TO I+16)="
480 LET D$(M TO M)="
490 LET D$(2*I-J TO M)="
500 LET D$(2*I-J TO M)="

```

```

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
6120 RETURN
6140 LET M=INT (RND*224+17)
6160 LET JE=0
6180 RETURN
6200 LET M=M+1
6220 IF M>240 THEN RETURN
6240 LET M=18
6260 IF JE THEN GOTO ER
6280 LET JE=1
6300 RETURN
6320 IF UP>0 THEN GOTO 6780
6340 GOSUB 7000
6360 PRINT AT 21,0;"HA HA HA. GA
NHEI."
6380 STOP
6400 RUN
6420 IF CP>0 THEN RETURN
6440 GOSUB 7000
6460 PRINT AT 21,0;"PERDI ESSA,
MAS PROXIMA ."
6480 GOTO 6730
6500 GOSUB 7000
6520 PRINT AT 21,0;"DESISTIU HEI
M?"
6540 GOTO 6730
6560 PRINT AT 19,0;UC;" ████ " ;
UF;" ████ ";C0;" ████ ";CF;" █
███ "
6580 PRINT AT 20,0;UA;" ████ " ;UD
;" ████ ";US;" ████ ";CA;" ████ " ;CD
;" ████ ";CS;" ████ "
6600 RETURN
7000 FOR I=18 TO 239
7010 IF U$(I TO I)<"█" THEN LET
U$(I TO I)=CHR$ (CODE (U$(I TO I)
)+128)
7020 IF C$(I TO I)<"█" THEN LET
C$(I TO I)=CHR$ (CODE (C$(I TO I)
)+128)
7030 IF U$(I TO I)="█" OR U$(I T
O I)="█" THEN LET U$(I TO I)=CHR$
(CODE (U$(I TO I))-128)

```

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

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

```

7040 LET H=H+1 THEN TO I=1 TO H DO H=H+1
7050 IF H>100 THEN LET H=100 AND H=H-X
7060 TO H>"X" THEN LET H=H+1 TO H DO H=H+1
7070 NEXT I
7080 GOSUB 6000
7090 GOSUB 6000
7100 RETURN

6000 LET K1=1+AV
6010 LET K14=16-AV
6020 LET ER=3400
6030 FOR K=1 TO V
6040 GOSUB 6000
6050 LET H=H
6060 LET M=H
6070 LET UB=6840
6080 IF T$(M-17 TO M-15)<>"■■■■■"
6090 THEN GOTO 6061
6100 IF T$(M-1 TO M+1)<>"■■■■■" TH
6110 EN GOTO 6061
6120 IF T$(M+15 TO M+17)<>"■■■■■"
6130 THEN GOTO 6061
6140 LET I=INT (M/16)
6150 LET N=M-I*16
6160 LET H=M
6170 LET H=I+1
6180 IF M<2 OR N<2 OR M>15 OR N>
6190 THEN GOTO 6060
6200 IF NU>1 THEN GOTO 6170
6210 LET T$(H TO H)="■"
6220 GOTO 6060
6230 LET D=0
6240 LET P1=0
6250 LET P2=0
6260 LET P3=0
6270 LET D=INT (RND*4)
6280 LET D=D+1-(D>3)*4
6290 GOTO (8200+D*80)
6300 IF P0 THEN GOTO 6040
6310 LET P0=1
6320 LET D=0
6330 LET I1=M-1
6340 LET I2=M+NU
6350 LET J1=N-K1
6360 LET J2=N+K1
6370 IF I2<=15 THEN GOTO 6530
6380 IF P1 THEN GOTO 6040
6390 LET D=1
6400 LET P1=1

```

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

```

6410 LET I1=M-1
6420 LET I2=M+NU
6430 LET J1=N-K1
6440 LET J2=N+K1
6450 LET D=0
6460 LET P0=1
6470 LET T$(H TO H)=P0
6480 LET H=H+1 THEN GOTO 6040
6490 LET H=H+1 THEN GOTO 6040
6500 LET H=H+1 THEN GOTO 6040
6510 LET H=H+1 THEN GOTO 6040
6520 LET H=H+1 THEN GOTO 6040
6530 LET H=H+1 THEN GOTO 6040
6540 LET H=H+1 THEN GOTO 6040
6550 LET H=H+1 THEN GOTO 6040
6560 LET H=H+1 THEN GOTO 6040
6570 LET H=H+1 THEN GOTO 6040
6580 LET H=H+1 THEN GOTO 6040
6590 LET H=H+1 THEN GOTO 6040
6600 LET H=H+1 THEN GOTO 6040
6610 LET H=H+1 THEN GOTO 6040
6620 LET H=H+1 THEN GOTO 6040
6630 LET H=H+1 THEN GOTO 6040
6640 LET H=H+1 THEN GOTO 6040
6650 LET H=H+1 THEN GOTO 6040
6660 LET H=H+1 THEN GOTO 6040
6670 LET H=H+1 THEN GOTO 6040
6680 LET H=H+1 THEN GOTO 6040
6690 LET H=H+1 THEN GOTO 6040
6700 LET H=H+1 THEN GOTO 6040
6710 LET H=H+1 THEN GOTO 6040
6720 LET H=H+1 THEN GOTO 6040
6730 LET H=H+1 THEN GOTO 6040
6740 LET H=H+1 THEN GOTO 6040
6750 LET H=H+1 THEN GOTO 6040
6760 LET H=H+1 THEN GOTO 6040
6770 LET H=H+1 THEN GOTO 6040
6780 LET H=H+1 THEN GOTO 6040
6790 LET H=H+1 THEN GOTO 6040
6800 LET H=H+1 THEN GOTO 6040
6810 LET H=H+1 THEN GOTO 6040

```

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

```

6690 IF I1<=1 THEN GOTO 6900
6695 IF J1<=1 THEN GOTO 6900
6700 IF I0>=16 THEN GOTO 6900
6705 IF J0>=16 THEN GOTO 6900
6710 IF I0<>0 THEN GOTO 6900
6715 FOR I=1 TO 16
6720 LET M1=I*16+16
6725 LET T$(M1+J1 TO M1+J2)=X$(J
6730 TO J2)
6735 NEXT I
6740 GOTO 6980
6745 LET M=(M-1)*16+N
6750 LET T$(M TO M)=“ ”
6755 LET M=(I1-1)*16+J1
6760 LET T$(M TO M)=“ ”
6765 LET M=(I2-1)*16+J2
6770 LET T$(M TO M)=“ ”
6775 NEXT K
6780 RETURN
6785 FAST
6790 CLERR
6795 SAVE “NAVAL”
6800 RUN
6805 PRINT AT 21,0;“ ”
6810 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: <u>TOTÓ</u>	c) Versão: <u>1</u>	d) Revisão: <u>Ø</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 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 10ØØ	<input checked="" type="checkbox"/>	DISCO: 8"	<input type="checkbox"/>	DSAVE	<input type="checkbox"/>
CP-2ØØ	<input checked="" type="checkbox"/>	5½ "	<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:					
<hr/> <hr/>					

**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 × 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 × equipe b*

no retorno o jogo será

*equipe b × 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? 2  
NO MÍNIMO TRES EQUIPES PARA FAZER UM TORNEIO.  
QUANTAS EQUIPES? 4

ENTRE AGORA OS NOMES DAS EQUIPES DA UM, SEGUIDOS DE NEWLINE.  
 1 - SAMYA  
 2 - MARJORIE  
 3 - DANIELLE  
 4 - ~~RODRIGO~~

VERMOS AGORA A TABELA:

DAREI O JOGO E VOCÊ ENTRA O SCORE NA ORDEM.

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

CLASSIFICACAO GERAL	
EQUIPE	GP
SAMYA	0
DANIELLE	0
MARJORIE	0
1 - SAMYA	0

CLASSIFICACAO GERAL	
EQUIPE	GP
SAMYA	0
MARJORIE	0
DANIELLE	0
1 - SAMYA	0

Figura 17.1  
EXEMPLO DE TORNEIO

CLASSIFICACAO GERAL	
EQUIPE	GP
SAMYA	0
DANIELLE	0
MARJORIE	0
4 - MARJORIE	0

CLASSIFICACAO GERAL	
EQUIPE	GP
SAMYA	0
DANIELLE	0
MARJORIE	0
5 - SAMYA	0

CLASSIFICACAO GERAL	
EQUIPE	GP
SAMYA	0
DANIELLE	0
MARJORIE	0
6 - DANIELLE	0

CLASSIFICACAO GERAL	
EQUIPE	GP
SAMYA	0
DANIELLE	0
MARJORIE	0
7 - SAMYA	0

Figura 17.1  
EXEMPLO DE TORNEIO (cont.)

QUEM FOI O TURNO,  
QUEM RETORNO? (S/N)

4 - [REDACTED] X DANIELLE

CLASSIFICACAO GERAL

EQUIPE	G	C	P
[REDACTED]	0	0	0
SANYA	0	0	0
DANIELLE	0	0	0
MARJORIE	0	0	0
4 - [REDACTED]	0	0	0
X DANIELLE	0	0	0

CLASSIFICACAO GERAL

EQUIPE	G	C	P
[REDACTED]	0	0	0
SANYA	0	0	0
DANIELLE	0	0	0
MARJORIE	0	0	0
6 - [REDACTED]	0	0	0
X DANIELLE	0	0	0

CLASSIFICACAO GERAL

EQUIPE	G	C	P
[REDACTED]	0	0	0
SANYA	0	0	0
DANIELLE	0	0	0
MARJORIE	0	0	0
4 - DANIELLE	0	0	0
X DANIELLE	0	0	0

CLASSIFICACAO GERAL

EQUIPE	G	C	P
[REDACTED]	0	0	0
SANYA	0	0	0
DANIELLE	0	0	0
MARJORIE	0	0	0
4 - [REDACTED]	0	0	0
X DANIELLE	0	0	0

CLASSIFICACAO GERAL

EQUIPE	G	C	P
[REDACTED]	0	0	0
SANYA	0	0	0
DANIELLE	0	0	0
MARJORIE	0	0	0
6 - DANIELLE	0	0	0
X DANIELLE	0	0	0

CLASSIFICACAO GERAL

EQUIPE	G	C	P
[REDACTED]	0	0	0
SANYA	0	0	0
MARJORIE	0	0	0
DANIELLE	0	0	0

Figura 17.1 EXEMPLO DE TORNEIO (cont.)

Figura 17.1 EXEMPLO DE TORNEIO (cont.)

## MANUAL TÉCNICO

Os dados são armazenados em vetores distintos

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

— 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
30 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 C(T)
140 DIM N$(T,10)
150 DIM P(T)
160 DIM U(T)
170 DIM D(T)
175 DIM E(T)
180 PRINT AT 20,0;"ENTRE AGORA
OS NOMES DAS EQUIPESUM A UM, SEC
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
260 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;"DARREI O JOGO
VOCE ENTRA O ESCORE NA OR
DEM."
350 SCROLL
360 LET K=0
370 FOR I=1 TO T-1
380 FOR J=I+1 TO T
390 LET K=K+1
400 LET R(K,1)=I
410 LET R(K,2)=J
420 LET R(K,3)=0
430 NEXT J
440 NEXT I
450 FOR K=1 TO Y
460 IF K>Y*3/4 THEN GOTO 480
470 LET M=INT((RND*Y+1))
475 IF R(M,3)<>0 THEN GOTO 470
480 GOTO 500
485 FOR M=1 TO Y
490 IF R(M,3)=0 THEN GOTO 500
495 NEXT M

500 LET I=R(M,1)
510 LET J=R(M,2)
520 LET R(M,3)=1
530 SCROLL
540 SCROLL
540 PRINT AT 21,0;(" " AND K<=9
);K;" - ";N$(I);" X ";N$(J)
550 INPUT I1
560 IF I1>0 THEN GOTO 600
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.)*

```

530 PRINT AT 21,0;I1;" X ";J1;
570 ****
640 GOTO 530
700 SCROLL
710 PRINT AT 21,5;I1;AT 21,18;J
720 LET G(I)=G(I)+I1
730 LET G(J)=G(J)+J1
740 LET O(I)=O(I)+I1
750 LET O(J)=O(J)+J1
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 U(J)=U(J)+1
840 LET D(I)=D(I)+1
850 LET P(J)=P(J)+2
860 GOTO 900
870 LET U(I)=U(I)+1
880 LET D(J)=D(J)+1
890 LET P(I)=P(I)+2
900 SCROLL
900 GOSUB 2000
910 SCROLL
910 PRINT AT 21,6;"CLASSIFICACAO
0 GERAL"
910 SCROLL
910 PRINT AT 21,0;"EQUIPE");AT 2
1,14;"J P U E D GP GC"
910 FOR H=1 TO T
910 LET L=0(H)
910 SCROLL
910 PRINT AT 21,0;N$(L);AT 21,1
15 U(L)+D(L)+E(L));AT 21,14;P(L);AT
21,17;V(L);AT 21,20;E(L));AT 21,
23;D(L);AT 21,26;G(L);AT 21,29;
29;
915 LET O(H)=H
916 NEXT H
917 NEXT K
918 IF (U(1)+D(1)+E(1))>T THEN
GOTO 1500
1000 SCROLL
1010 SCROLL
1020 SCROLL
1030 PRINT AT 20,0;"E ASSIM FOI
0 TURNO,
0 (S/N)" HAVERA RETUR

```

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 1080
1070 SCROLL
1080 PRINT AT 20,0;"TECLH
0000 DESENHA QUE TA
0000 PRESENTEM DENOVOU."
1100 GOTO 1040
1100 FOR H=1 TO Y
1110 LET R(H,0)=0
1120 LET R(H,1)=R(H,0)
1130 LET R(H,2)=R(H,1)
1140 LET R(H,3)=R(H,2)
1150 LET R(H,4)=R(H,3)
1160 LET R(H,5)=R(H,4)
1170 LET R(H,6)=R(H,5)
1180 LET R(H,7)=R(H,6)
1190 LET R(H,8)=R(H,7)
1200 LET R(H,9)=R(H,8)
1210 STOP
1210 RUN
1220 LET O=0
1230 FOR H=1 TO T-1
1240 LET O=O(H)
1250 LET O=O(H+1)
1260 IF P(O)>P(S) THEN GOTO 2500
1270 IF P(O)=P(S) THEN GOTO 2100
1280 LET O=1
1290 LET O(H+1)=0
1300 LET O(H)=0
1310 GOTO 2500
1320 IF U(W)>U(S) THEN GOTO 2500
1330 IF U(W)<U(S) THEN GOTO 2050
1340 IF (G(W)-O(W))>(G(S)-O(S))
1350 THEN GOTO 2500
1360 IF (G(W)-O(W))<(G(S)-O(S))
1370 THEN GOTO 2060
1380 IF G(W)>G(S) THEN GOTO 2500
1390 IF G(W)<G(S) THEN GOTO 2060
1400 NEXT H
1410 IF O<>0 THEN GOTO 2000
1420 RETURN
1430 CLAST
1440 CLEAR
1450 SAVE "TOTO"
1460 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 evitárá 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 cassette, 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 cassette, contendo os programas gravados, podem vir junto com o livro ou serem adquiridas separadamente, opcionalmente, à escolha do leitor. A utilização de fitas cassette 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 *manuais*: 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 encontrarem 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.

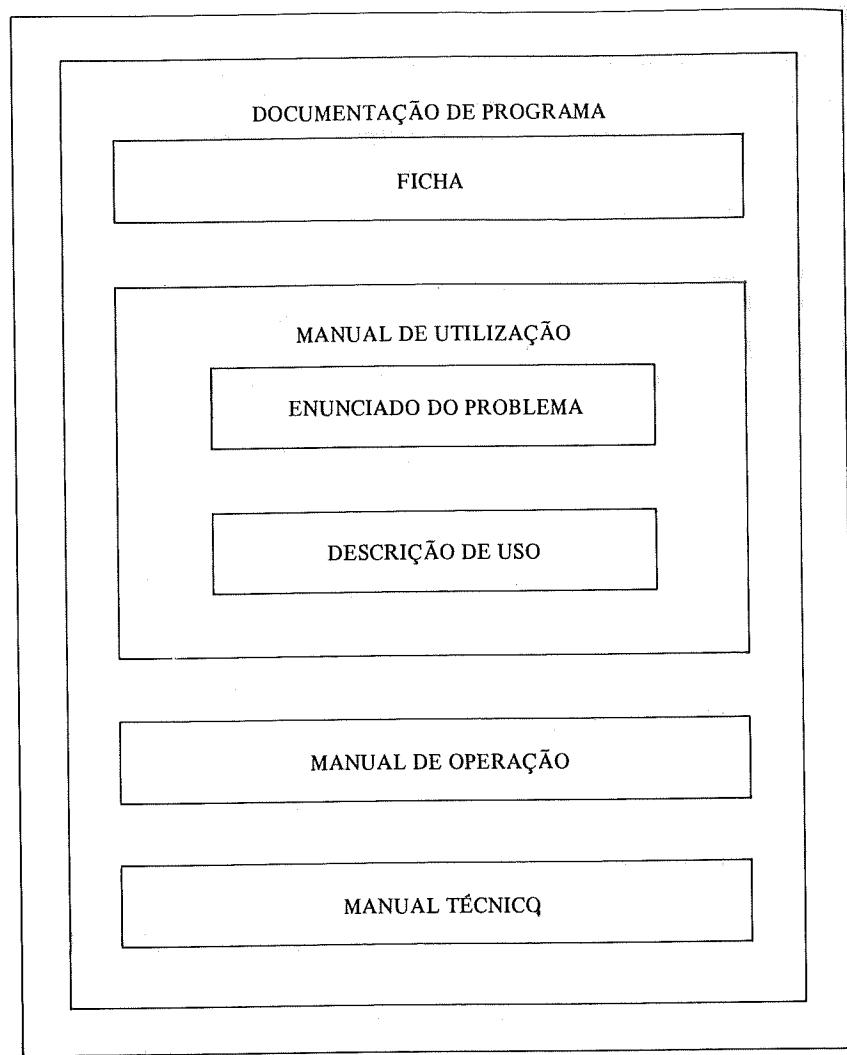


Figura A.1 ESQUEMA DA ORGANIZAÇÃO GERAL DA DOCUMENTAÇÃO DE PROGRAMA

### 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	b) Nome:	c) Versão:	d) Revisão:	e) Data: //
Sub-rotina	f) Objetivo:			
Função				
g) Linguagem: PL/	<input type="checkbox"/> BASIC	<input type="checkbox"/> PASCAL	<input type="checkbox"/> FORTH	<input type="checkbox"/> OUTRAS (quais?)
	<input type="checkbox"/> FORTRAN	<input type="checkbox"/> COBOL	<input type="checkbox"/> C	
h) Compatibilidade:	i) Recursos:	j) Instruções/Comandos Especiais:		
NE 280	<input type="checkbox"/> VÍDEO	<input type="checkbox"/> SLOW		
NE 28000	<input type="checkbox"/> TECLADO	<input type="checkbox"/> FAST		
TK 80	<input type="checkbox"/> MEMÓRIA: 2K	<input type="checkbox"/> LPRINT		
TK 82-C/83	<input type="checkbox"/> 16K	<input type="checkbox"/> LLIST		
TK 85	<input type="checkbox"/> 48K	<input type="checkbox"/> COPY		
SINCLAIR ZX 80	<input type="checkbox"/> 64K	<input type="checkbox"/> DLOAD		
SINCLAIR ZX 81	<input type="checkbox"/> CASSETE	<input type="checkbox"/> DSAVE		
SINCLAIR SPECTRUM	<input type="checkbox"/> DISQUETE	<input type="checkbox"/> DMLOAD		
TIMEX 1000	<input type="checkbox"/> DISCO: 8"	<input type="checkbox"/> DMSAVE		
CP-200	<input type="checkbox"/> 5½"	<input type="checkbox"/> OUTROS (quais?)		
CP-300	<input type="checkbox"/> ALAVANCA ("JOYSTICK")			
APPLY 300	<input type="checkbox"/> IMPRESSORA			
OUTROS (quais?)	<input type="checkbox"/> OUTROS (quais?)			
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 *manuais* 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 *manuais*, 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 interesse, 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, suscintas 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>B – MANUAL DE UTILIZAÇÃO</b>
<b>a) ENUNCIADO DO PROBLEMA:</b>
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>b) DESCRIÇÃO DE USO:</b>
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, suscintas 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 pseudolínguagem 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 cassette
C	Armazenamento de programas em fita cassette
D	Cuidados com gravadores e fitas cassette

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 cassette 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 Ø (zero), correspondendo ao mínimo ou Ø%, até 1Ø (dez), correspondendo ao máximo ou 1ØØ%. 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 7Ø% (número 7 do controle) para *tonalidade*, então, 7Ø% (número 7 do controle) para o *volume*, e assim sucessivamente. Em muitos gravadores, o nível *adequado* para o carregamento está em cerca de 8Ø% (para *tonalidade* e *volume*), mas a *tonalidade* pode chegar a 1ØØ% (máximo do agudo). É preciso experimentar e determinar a posição *adequada* dos controles. Por exemplo, a leitura de programas em fita cassette *normal*, com gravadores NATIONAL modelo RQ-2222MA, só é conseguida após um cuidadoso ajuste dos controles geralmente entre 5Ø% e 7Ø%;
3. Se a fita cassette 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 (ØØØ). 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 Ø/Ø. 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* é 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 freqüência, 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), **SAVÉ** (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 *imediata 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 *clic* 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 cassette, 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 cassette, 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 cassette (A ou B). Existem fitas cassette com diversos tamanhos, sendo os mais comuns C-46, C-6 $\varnothing$  e C-12 $\varnothing$ . A preferência deve recair na fita de tamanho médio, C-6 $\varnothing$ . 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.

