

CRIANDO EM LINGUAGEM DE MÁQUINA

Não basta se conhecer as instruções do ASSEMBLY para se criar um programa interessante: existem truques e "dicas" indispensáveis para o bom programador.

- . Gerenciando a tela
- . Utilizando o teclado
- . Usando o "Joystick"
- . Gerando movimentos múltiplos
- . Criando opções
- . Implementando "requintes"

Em cada capítulo um jogo original, totalmente explicado, ilustrando os truques de programação utilizados.

Uma obra indispensável para quem programa em micros da linha SINCLAIR: CP 200 – RINGO – TK 82/83/85 – AS 1000.

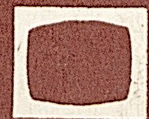
Samuel Ejchel formou-se em Engenharia Mecânica na Poli (USP) em 1963 e voltou todos os seus estudos para uma carreira coerente com sua opção profissional. Fez pós graduação em Engenharia de Produção e frequentou vários cursos de atualização no Brasil e no exterior.

É por isso surpreendente se ver o diretor da Divisão de Amortecedores da COFAP (que conta atualmente com mais de 1500 funcionários) publicar um livro de computação onde ensina com muito bom humor e didática quais as "manhas" para se elaborar um jogo divertido em Assembly.

A explicação é simples: ao adquirir seu computador em fins de 82, o Samuel também adquiriu um hobby apaixonante e envolvente.

Para quem gosta de pensar e usar a inteligência, a arte de programação é mais estimulante que a Matemática ou o Xadrez.

Para compartilhar deste entusiasmo com o público cada vez maior de usuários, o Samuel escreveu este livro. Usar a inteligência é gostoso e o Samuel faz isto como poucos!



SAMUEL EJCHEL

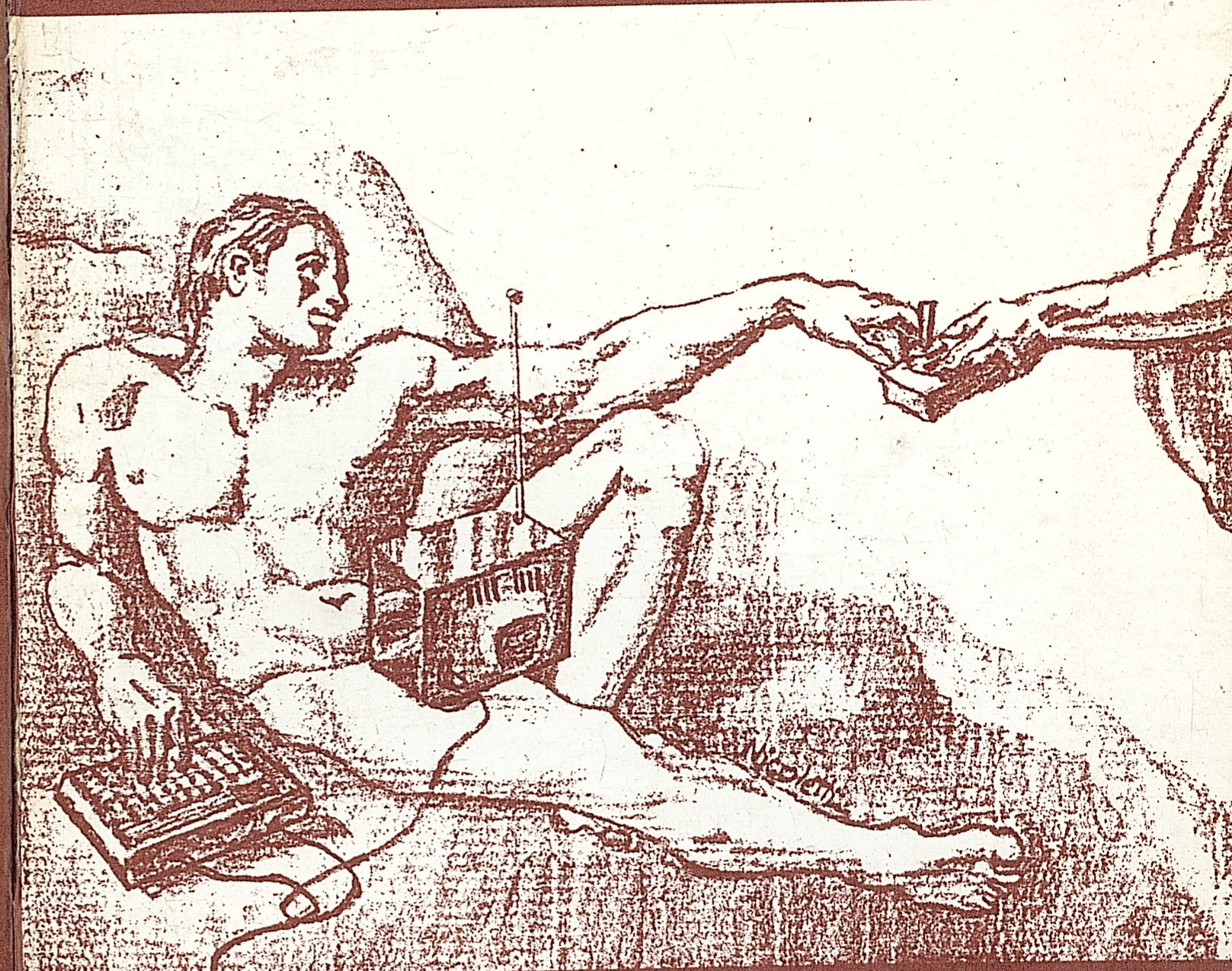
CRIANDO EM LINGUAGEM DE MÁQUINA

URANIA

EDITORA MODERNA

CRIANDO EM LINGUAGEM DE MÁQUINA

SAMUEL EJCHEL



MONTAGENS DE JOGOS EM ASSEMBLY Z 80
PARA CP 200, RINGO, TK 82/83/85, SINCLAIR ZX81

URANIA

EDITORA MODERNA

CRIANDO EM
LINGUAGEM DE MÁQUINA

SAMUEL EJCHEL

URANIA

EM
EDITORA
MODERNA

Coordenação Editorial: Pierluigi Piazzi
Avaliação e Revisão Técnica: Nancy Mitie Ariga
Ilustrações e capa: Durvaly Odilon Nicoletti
Arte: Mario Dimov Mastrotti
Revisão e copy-desk: Lúcia Kairovsky
Produção: Rosa Kogan Fromer
Digitação: Todos os programas foram digitados e testados

no depto. de CURSOS DE COMPUTAÇÃO DO
NÚCLEO DE ORIENTAÇÃO DE ESTUDOS
(813-4555)
sob a supervisão da professora Betty F. Piazzi.



© 1985 URANIA Publicações e Assessoria Pedagógica Ltda.
Av. Brig. Faria Lima, 1451 conj. 31
01451 S. Paulo SP tel: (011) 813-4555

Distribuição exclusiva em lojas e livrarias
Editora Moderna
Rua Afonso Brás, 431
04511 S. Paulo SP tel: (011) 531-5099

CIP-Brasil, Catalogação-na-Publicação
Câmara Brasileira do Livro, SP

E37c	Ejchel, Samuel. Criando em linguagem de máquina / Samuel Ejchel. — São Paulo: Urania: Ed. Moderna, 1985.
	1. Jogos eletrônicos 2. Microcomputadores — Programação I. Título.
	17. CDD-651.8
	18. -001.6425
84-2163	18. -001.642

Índices para catálogo sistemático:

1. Jogos: Computadores: Programas: Processamento de dados 651.8 (17.) 001.6425 (18.)
2. Microcomputadores: Linguagem de programação: Processamento de dados 651.8 (17.) 001.642 (18.)

NOTA DO EDITOR

Enquanto fui diretor editorial da revista MICROHOBBY, procurei sempre dar ênfase ao aspecto "desafio à inteligência" que o microcomputador apresenta.

Com vista nisso, concebi desde o primeiro número da revista uma seção "Quebra-Cabeça" na qual eram propostos problemas que deveriam ser resolvidos em "simbiose" com o microcomputador. Foi a maneira que encontrei para tentar estimular tanto a inteligência artificial quanto a natural de meus leitores.

Às vezes eu exagerava um pouco na dificuldade dos problemas e o pessoal da redação ficava ansioso pois não chegavam cartas com a resposta correta para que a pudessemos publicar no número seguinte.

Não uma, mas várias vezes fomos "salvos pelo gongo" por um leitor de São Paulo que nos enviava sempre a resposta correta e elaborada da maneira mais elegante possível.

As cartas que ele enviava mostravam nitidamente uma inteligência brilhante que aceitava e gostava de desafios. O nome dele tornou-se conhecido na redação tanto quanto se ele fosse um de nós: SAMUEL EJCHEL.

Alguém que sente prazer em pensar (espécie atualmente em extinção) só podia ser um amador entusiasta: realmente o Samuel não é um profissional do ramo e o que ele aprendeu de linguagem de máquina, foi como auto-didata.

Assim sendo ele tem bem nítidas na mente as dificuldades que um iniciante na arte da computação pode enfrentar. Ao mostrar como ele contornava essas dificuldades quando estava elaborando alguns jogos, eu senti a tremenda importância que essa experiência poderia representar para outros programadores.

Mais importante que o resultado, a elaboração dos jogos tinha que ser transmitida aos outros usuários. O melhor caminho para isso era colocar o Samuel dando cursos ou escrevendo um livro.

Optamos pela segunda hipótese e o resultado aí está. Confesso que, re lendo a obra, sinto orgulho.

É um orgulho "obstétrico": o filho não é meu, mas contribuí para que ele visse a luz. Meu "faro" não se enganou: o Samuel mais uma vez aparece com uma solução brilhante e elegante.

Parabéns!
Pierluigi Piazzi

INTRODUÇÃO

*À Frida, Lilian, Beatriz e Flávia pelo apoio e compreensão.
À Rita e Luiz pela ajuda.*

ÍNDICE

Nota do Editor	3
Introdução	5
Cap. 1 – A TELA	7
Cap. 2 – O TECLADO	29
Cap. 3 – O JOYSTICK	51
Cap. 4 – MOVIMENTOS MÚLTIPLOS	75
Cap. 5 – OPÇÕES	101
Cap. 6 – REQUINTES	135
APÊNDICE A – BIBLIOGRAFIA	183
APÊNDICE B – MONITOR	184
APÊNDICE C – MAPA DA ROM	187
APÊNDICE D – TABELA	188

Este livro é destinado àqueles que, conhecedores de linguagem de máquina, desejam aprender ou aprimorar-se na técnica de criar jogos no computador. Esta atividade, mais que outras, revela-se como um atraente e apaixonante passatempo, tipo quebra-cabeças, que exige de quem o faz um grande esforço de atenção e concentração, constituindo-se portanto num excelente exercício para a mente. A satisfação que se sente ao terminar um desenvolvimento, e ver na tela a materialização de uma idéia, é contagiante e altamente gratificante.

Programar em linguagem de máquina já é de per si um desafio. Para jogos, o uso desta linguagem é indispensável, pelo menos nos microcomputadores da linha Sinclair, por ser a única compatível com as velocidades exigidas no movimentos.

Para usufruir deste livro o leitor deve conhecer as instruções de máquina, ou Assembly, do microprocessador Z-80. Algumas destas instruções são explicadas no texto, porém a maioria assume-se como conhecidas e, em caso de dúvida, recomenda-se consultar a bibliografia sobre o assunto. No apêndice A existem algumas indicações para isto.

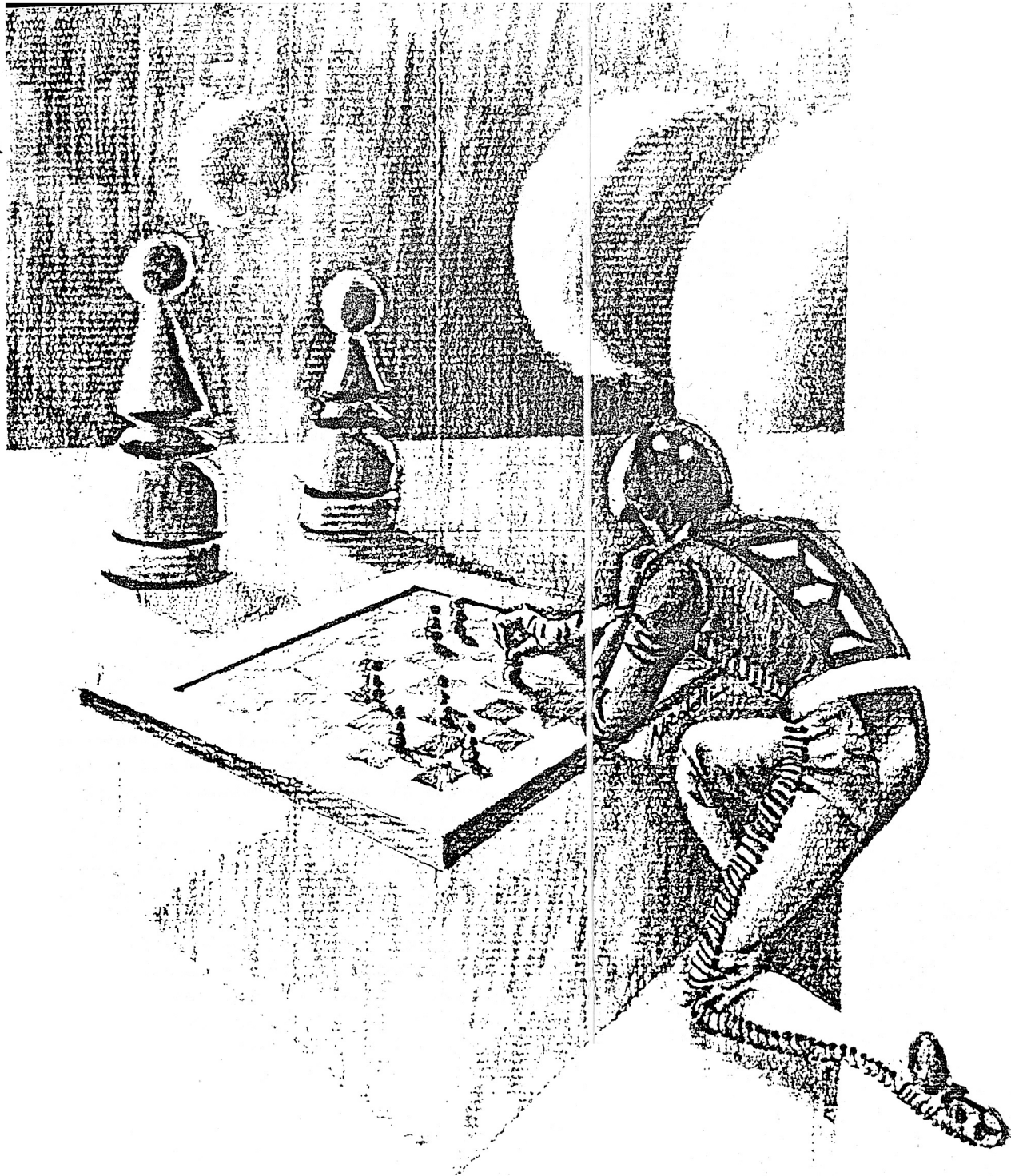
O equipamento a usar deve ser da linha Sinclair ou compatíveis, Ringo, ZX 81, CP 200, AS 1000, NEZ 8000, TK 82C, TK 85, TK 83, com capacidade maior ou igual a 2 Kbytes.

Os programas apresentados necessitam para rodar, em sua maioria, algo perto deste limite. Assim se sua máquina tiver somente 2 Kbytes, convém conseguir uma expansão emprestada para permitir a digitação e carregamento dos programas.

Para carregar os programas, separados em grupos, qualquer monitor ou carregador de linguagem de máquina, que o leitor disponha, pode ser usado. Após cada grupo, devidamente desassemblado, temos a mesma listagem em heptassintático. Querendo usar esta opção, consulte o apêndice B.

Este livro é dividido em 6 capítulos. Em cada um é enfatizada uma característica relacionada com os jogos e é apresentado um programa ilustrativo detalhado e explicado em pequenos blocos. Vários destes contém rotinas de uso geral, que poderão ser copiadas e utilizadas de imediato em novas aplicações. Faça bom uso delas!

A TELA



Neste capítulo discutimos como gerenciar a tela do computador e como utilizá-la da mais eficiente forma. Aproveitamos também para exemplificar o que foi dito, elaborando um jogo deveras interessante denominado "Revertendo" onde também é ensinado como criar pseudo-registros.

CAPÍTULO 1 – A TELA

Indiscutivelmente a tela é o elemento mais importante quando se pretende programar jogos. Enquanto que numa aplicação mais "séria", a tela é o meio para dar saída de resultados de processamento ou listagens da maneira mais clara possível, nos jogos a estética e o dinamismo da imagem são a própria essência dos mesmos.

Nos computadores da linha Sinclair o conteúdo da tela está armazenado na memória em forma de um arquivo seqüencial de bytes, cada um correspondendo a um de seus pontos. Este arquivo é conhecido como **Dfile**, abreviatura do inglês Display File. O endereço onde tem início é uma das chamadas Variáveis do Sistema (como referência ver o manual do seu equipamento) e se encontra armazenado no par de bytes de endereço 16396 e 16397 (400C e 400D em hexa).

Para se obter com estes, o endereço X de início do arquivo da tela basta fazer

```
LET X=PEEK 16396+256*PEEK 16397
```

A organização deste arquivo é função da capacidade de memória do equipamento. Para RAMs acima de 3,5 Kbytes tem a configuração esquemática apresentada na figura 1.

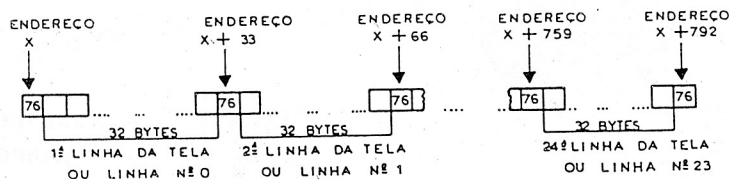


FIGURA 1

Isto significa que a tela ocupa 793 bytes de memória do computador, ou seja 24 vezes 32 bytes de cada linha da tela mais o inicial e final contendo a indicação de New Line (118 ou 76 em hexa), somando 25 bytes. Todos eles ficam reservados somente para a tela, e podem se "mover" ao longo da memória, em função do tamanho do programa que acompanha seu uso. Todavia o endereço aonde o conjunto encontra-se em cada situação é sempre indicado pelos valores contidos nos bytes 16396 e 16397.

Ora, 793 bytes representam uma parcela ponderável do total, quando a capacidade da RAM é inferior a 3,5 Kbytes. Neste caso, para economizar memória, a organização do arquivo de tela é diferente: quando se liga a máquina, o arquivo contém apenas os 25 New Lines (118 ou 76 em hexa), e cresce na medida em que se vai "escrevendo" na tela. Por exemplo, partindo da tela vazia, após comandar:

```
PRINT AT 2,0;"TESTE"
```

o arquivo teria a configuração esquemática indicada na fig. 2.

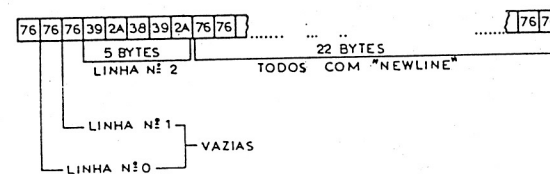



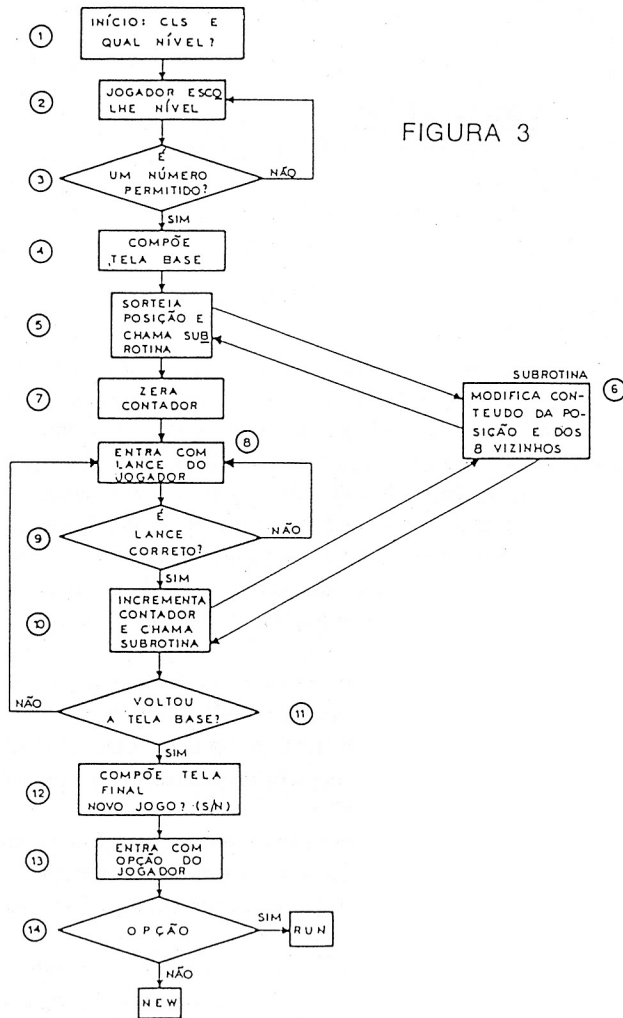
FIGURA 2

Cabe esclarecer que nesta explicação está sendo omitido o que ocorre com as duas últimas linhas da tela, as quais, quando se trabalha em Basic, ficam sob controle do Sistema gerenciador desta linguagem. Porém ao passar para linguagem de máquina temos possibilidade de utilizar todas as 24 linhas como quisermos. O indicador do número de linhas reservadas pelo Basic na parte inferior da tela, é a variável de Sistema DFSZ, de endereço 16418 (4022 em hexa). Mais adiante veremos como utilizar esta informação.

A fim de ilustrar os conceitos acima e introduzir novos, vamos apresentar e analisar o programa de um jogo chamado Revertendo. A idéia do jogo consiste no seguinte:

- * O jogador, de início, define um número — o "Nível de Dificuldade" — que será utilizado no programa para compor a tela ou tabuleiro do jogo.
- * O computador inicialmente constrói a tela base, que consiste em uma moldura de orientação, com letras no sentido horizontal e números no vertical, totalmente cheia de símbolos "+".
- * Em seguida o programa sorteia um número de posições da tela igual ao Nível de Dificuldade eleito pelo jogador, e faz uma modificação do conteúdo de cada uma delas e das oito posições que cercam a sorteada (se lá havia uma + passa a um  e vice-versa).
- * Cabe ao jogador, através de lances, definindo a coluna (letra) e a linha (número), indicados nesta ordem, reverter (daí o nome do jogo) o tabuleiro às condições primitivas.

* Finalizando, o computador relembra o nível de dificuldade escolhido, e informa o número de jogadas ou lances feitos pelo jogador. A comparação destes dois valores dá a noção do desempenho atingido na partida. Embora não seja indispensável, é recomendável que se faça e use o diagrama de blocos do programa. Isto serve para fixar o raciocínio e evitar erros.



O programa em Basic correspondente é o seguinte.

```

3 CLS
5 PRINT "NÍVEL DE DIFICULDADE
?" : PT 0/0;
10 INPUT F#
15 FOR D=1 TO LEN F#
20 IF CODE F$(D) < 28 OR CODE F#
(D) > 37 THEN GOTO 10
25 NEXT D
28 IF D > 4 THEN GOTO 10
30 LET S=VAL F#
35 FAST
40 PRINT " ABCDEFGHIJKLMNOPQ
RSTUVWXYZ"
42 PRINT "+++++"
44 FOR Z=1 TO 18
45 LET M$(Z+100) = (2 TO
)
46 PRINT M$; "+++++"
+++++"; M$
47 NEXT Z
48 PRINT "+++++"
49 PRINT " ABCDEFGHIJKLMNOPQ
RSTUVWXYZ"
50 LET V=PEEK 16396+256*PEEK 1
6397
60 RAND
100 FOR Z=1 TO 5
105 LET X=INT (RAND*26)+4
106 LET Y=INT (RAND*18)+2
110 GOSUB 1000
120 NEXT Z
125 LET N=0
130 SLOW
200 INPUT X#
205 IF CODE X# < 38 OR CODE X# > 63
THEN GOTO 200
207 IF LEN X# < 2 OR LEN X# > 3 THE
N GOTO 200
210 LET X=CODE X#-34
212 FOR M=2 TO LEN X#
214 IF CODE X$(M) < 28 OR CODE X#
(M) > 37 THEN GOTO 200
216 NEXT M
220 LET Y=VAL X$(2 TO )+1
230 IF Y < 2 OR Y > 19 THEN GOTO 20
0
235 LET N=N+1
240 GOSUB 1000
241 FAST
  
```

```

242 FOR I=1 TO 726
244 IF PEEK (V+I)=149 THEN GOTO
130
248 NEXT I
250 GOTO 1050
1000 LET P=U+X+Y#33
1010 POKE P-34,PEEK (P-34)-128
1011 POKE P-33,PEEK (P-33)-128
1012 POKE P-32,PEEK (P-32)-128
1013 POKE P+1,PEEK (P+1)-128
1014 POKE P+34,PEEK (P+34)-128
1015 POKE P+33,PEEK (P+33)-128
1016 POKE P+32,PEEK (P+32)-128
1017 POKE P-1,PEEK (P-1)-128
1018 POKE P,PEEK P-128
1019 RETURN
1050 PRINT AT 6,3;" NIVEL DE DIF
ICULDADE ";S;"
1060 PRINT AT 6,8;" VOCE CONSEGU
IU "
1070 PRINT AT 9,8;" EM ";N;" JOG
ADAS "
1080 PRINT AT 13,7;" NOVO JOGO?
(S/N) "
1085 SLOW
1090 IF INKEY#="" THEN GOTO 1090
1100 IF INKEY#="S" THEN RUN
1110 NEW
1180 SAVE "RE"
1190 RUN

```

A correlação entre as instruções do programa e os blocos é a seguinte:

BLOCO	INSTRUÇÕES
1	3 e 5
2	10
3	15 a 28
4	30 a 49
5	60 a 106
6	1000 a 1019
7	125
8	200
9	205 a 230
10	235
11	241 a 250
12	1050 a 1080
13 e 14	1090 a 1110

Observe agora o programa, a partir da linha 50 até a sub-rotina 1000 (inclusive), e veja como a tela está sendo alterada. A variável V da linha 50 na da mais é senão o endereço que se encontra na Variável de Sistema DFilê que falamos acima, e portanto corresponde ao início do arquivo da tela.

Para entender as instruções seguintes, convém fazer um esboço da tela como o que apresentamos na figura 4. Aliás, quando se programa em linguagem de máquina, a confecção deste esboço é praticamente indispensável. Então, caso você deseje prosseguir é bom já ir providenciando um bloco ou caderno com papel quadriculado, ou tirando cópias do apêndice D.

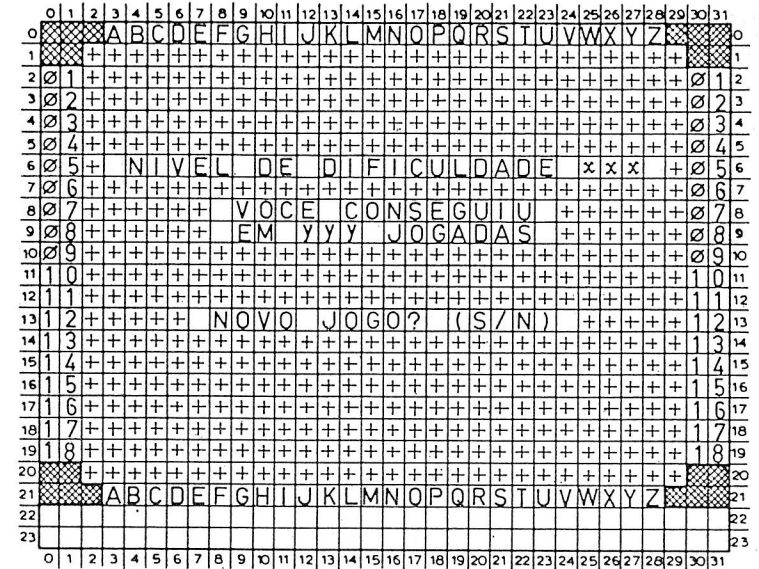



FIGURA 4

Com o auxílio do esboço torna-se mais fácil observar que a variável x, gerada na instrução 105 e variando de 4 a 29, corresponde a uma das colunas identificadas pelas letras, e a Y, gerada na 106 variando de 2 a 19, está ligada às linhas. Com a variável P da instrução 1000, obtida a partir das anteriores, chega-se ao endereço no arquivo da tela da memória do ponto a ser convertido. Correlacione a figura 1 com a 4 e entenda como isto se processa. Note que o que está sendo dito é válido tanto para máquinas com mais ou com menos que 3,5 Kbytes que, neste último caso, como a tela está totalmente ocupada, o seu arquivo na memória estará completo e será igual ao daquele com mais de 3,5 Kbytes.

As instruções seguintes, de 1010 a 1018, fazem a conversão, já que elas

“pegam” (Peek) o que está numa posição de memória, e subtraindo 128 (somar 128 seria a mesma coisa), inverte o “fundo” do símbolo gráfico que lá estava, e o “recoloca” (Poke) no mesmo lugar.

Outro bloco que merece uma análise é o 11, onde o computador verifica se a tela voltou ao que era, ou seja, se já não existe nenhum  (código 149) nela. Para isto parte-se novamente de V e percorre-se todo o arquivo útil da tela (726 = 33 x 22), conforme indicam as instruções 242 a 248.

Digite agora o programa e faça-o rodar. Frustrante! O tempo que a tela fica apagada enquanto se processa a rotina do bloco 11 é realmente muito grande para ser agradável. O primeiro impulso é eliminar a rotina, substituindo-a por um comando manual. Afinal o jogador está vendo que já “limpou” a tela e pode dar este comando.

Faça então o seguinte: digite uma nova linha como segue

```
1 REM ERND=CHR$ "Y";GOSUB TAN
```

Elimine as instruções 241 a 250 (inclusive) e mais a 1085, e introduza as seguintes novas instruções

```
244 LET K=USR 16514
248 IF K=0 THEN GOTO 1050
250 GOTO 200
```


Faça o programa “rodar” de novo e observe como tudo mudou! Vejamos o que foi feito:

Na instrução REM temos

2A OC 40	LD HL (DFile)
01 D6 02	LD BC, 726 d
3E 95	LD A, 149 d
ED B1	CPIR
C9	RET

Consultando-se a bibliografia constata-se que CPIR funciona do seguinte modo: compara o conteúdo do endereço apontado por HL com o valor de A e decrementa BC; a instrução é interrompida em duas situações, ou quando A = (HL) ou se BC atinge zero (neste último caso o “flag” zero é “setado” e instruções condicionais são possíveis). Ou seja, este pequeno conjunto de bytes faz exatamente o que a rotina anterior fazia, porém em velocidade muito maior.

Quando retorna ao Basic, o sistema traz junto com o argumento do USR o conteúdo do par BC, ou seja, após a instrução 244 a variável K fica

igual ao último conteúdo de BC. Com isso sabe-se se não existe mais  na tela, já que nesta situação K = 0.

Vamos agora reformular o programa de modo a utilizar muito mais as rotinas em linguagem de máquina. O novo Basic é o seguinte:

```
710 CLS
720 PRINT "NIVEL DE DIFICULDADE";
730 INPUT F$
740 LET V=USR 16577
750 IF V<>0 THEN GOTO 730
760 LET S=VAL F$
770 RAND
780 FOR Z=1 TO S
790 LET X=INT (RND*26)+4
800 LET Y=INT (RND*18)+2
810 POKE 16444,X
820 POKE 16445,Y
830 LET V=USR 16980
840 NEXT Z
850 SLOW
855 CLEAR
860 INPUT F$
870 LET V=USR 16837
880 IF V=0 THEN GOTO 855
890 IF V<>0 THEN PRINT AT 9,12;
V;" JOGADAS "
900 IF INKEY$="" THEN GOTO 900
910 IF INKEY$="S" THEN RUN
920 NEW
930 SAVE "REV"
940 RUN
```

O diagrama de blocos, como era de se esperar, é o mesmo que foi apresentado na figura 3, porém com uma alteração — a utilização de duas sub-rotinas distintas para modificar o conteúdo da posição e dos 8 vizinhos (bloco 6), uma para acompanhar o bloco 10 e outra, bem mais rápida, para atuar junto ao bloco 5.

A correlação entre blocos e instruções passa a ser:

BLOCO	INSTRUÇÕES
1	710 a 720
2	730
3 e 4	740 a 750
5, 6 e 7	760 a 840
8	860
9, 10, 11 e 12	870 a 890
14	900 a 940

No programa em linguagem de máquina vamos nos utilizar de quatro conjuntos de bytes, que funcionam como registradores, escolhidos para manterem dados ou participarem na lógica de programação. Este tipo de "registrador improvisado" é comumente usado quando se programa em Assembly.

Lugares bons para serem escolhidos são os bytes vazios na região das variáveis do Sistema, sejam o 16417 (4021 em hexadecimal), o par 16507/8 (407B/C) e os bytes do "buffer" da impressora 16444 a 16475 (403C a 405B). Note-se que o buffer da impressora é apagado quando se retorna para o Basic, mas se mantém quando vai para a linguagem de máquina. Os motivos para escolher esses bytes são:

* Eles não apresentam desvantagem em relação a qualquer outro que fosse escolhido.

* Normalmente um registrador não precisa (e talvez nem deva) ficar no meio programa.

* Estes bytes representam pontos fixos na memória e assim, quando se move o programa de uma região a outra, não é necessário mudar o endereço de chamada.

* Os três primeiros bytes citados são preservados pelo SAVE e não são apagados por CLEAR ou RUN.

Quando se utiliza este tipo de registrador é comum batizá-lo com um nome sugestivo para identificá-lo e facilitar a programação. Os quatro registradores que definimos são os seguintes:

NOME	CÓDIGO	ENDEREÇO
Contador	Cont	16507/8 (407B/C em hexa)
Transferência de X	TrX	16444 (403C em hexa)
Transferência de Y	TrY	16445 (403D em hexa)
Registro de Nível de	TEX	16536 a 16538 (4098 a 40A0
Dificuldade	(3 bytes)	em hexa)

O programa em linguagem de máquina será apresentado em grupos e sub-rotinas correlacionadas tanto quanto possível com os blocos do diagrama que apresentamos (fig. 3). Para carregar o mesmo, bem como todos os demais programas apresentados neste livro, o leitor poderá utilizar qualquer carregador que disponha, assumindo sempre que o bloco em linguagem de máquina esteja concentrado numa única linha REM, partindo do endereço 16514 (4082 em hexa). Como foi dito na falta de um carregador próprio, apresentamos um no apêndice B. Como simbologia, espaços em branco serão apresentados por `▯`.

Grupo 1 – 16514 a 16576 (4082 a 40CO em hexa)

`▯ NÍVEL ▯ DIFICULDADE ▯▯▯▯▯ VOCÊ ▯ CONSEGUIU ▯▯ NOVO ▯`
`JOGO? (S/N) ▯`

Este grupo é todo especial. Ele contém o texto que será usado na finalização (bloco 12), e apresenta mais uma particularidade interessante: os três bytes centrais dos 5 espaços que seguem a frase "Nível de Dificuldade" correspondem justamente ao registro TEX que foi criado. Note (pelo basic) que após o uso, no bloco 5, do nível escolhido pelo jogador esta informação só será necessária na finalização. Assim, nada melhor do que já deixá-la num lugar favorável para a tarefa final.

```

16514 00 00 00 00 00 00 01 00 00 00 00 00
16515 00 01 00 00 00 00 00 00 00 00 00 00
16516 00 00 00 00 00 00 00 00 00 00 00 00
16517 00 00 00 00 00 00 00 00 00 00 00 00
16518 00 04 00 00 00 00 00 00 00 00 00 00
16519 00 04 00 00 00 00 00 00 00 00 00 00
16520 00 04 00 00 00 00 00 00 00 00 00 00
16521 00 04 00 00 00 00 00 00 00 00 00 00
16522 00 04 00 00 00 00 00 00 00 00 00 00
16523 00 04 00 00 00 00 00 00 00 00 00 00
16524 00 04 00 00 00 00 00 00 00 00 00 00
16525 00 04 00 00 00 00 00 00 00 00 00 00
16526 00 04 00 00 00 00 00 00 00 00 00 00
16527 00 04 00 00 00 00 00 00 00 00 00 00
16528 00 04 00 00 00 00 00 00 00 00 00 00
16529 00 04 00 00 00 00 00 00 00 00 00 00
16530 00 04 00 00 00 00 00 00 00 00 00 00
16531 00 04 00 00 00 00 00 00 00 00 00 00
16532 00 04 00 00 00 00 00 00 00 00 00 00
16533 00 04 00 00 00 00 00 00 00 00 00 00
16534 00 04 00 00 00 00 00 00 00 00 00 00
16535 00 04 00 00 00 00 00 00 00 00 00 00
16536 00 04 00 00 00 00 00 00 00 00 00 00
16537 00 04 00 00 00 00 00 00 00 00 00 00
16538 00 04 00 00 00 00 00 00 00 00 00 00
16539 00 04 00 00 00 00 00 00 00 00 00 00
16540 00 04 00 00 00 00 00 00 00 00 00 00
16541 00 04 00 00 00 00 00 00 00 00 00 00
16542 00 04 00 00 00 00 00 00 00 00 00 00
16543 00 04 00 00 00 00 00 00 00 00 00 00
16544 00 04 00 00 00 00 00 00 00 00 00 00
16545 00 04 00 00 00 00 00 00 00 00 00 00
16546 00 04 00 00 00 00 00 00 00 00 00 00
16547 00 04 00 00 00 00 00 00 00 00 00 00
16548 00 04 00 00 00 00 00 00 00 00 00 00
16549 00 04 00 00 00 00 00 00 00 00 00 00
16550 00 04 00 00 00 00 00 00 00 00 00 00
16551 00 04 00 00 00 00 00 00 00 00 00 00
16552 00 04 00 00 00 00 00 00 00 00 00 00
16553 00 04 00 00 00 00 00 00 00 00 00 00
16554 00 04 00 00 00 00 00 00 00 00 00 00
16555 00 04 00 00 00 00 00 00 00 00 00 00
16556 00 04 00 00 00 00 00 00 00 00 00 00
16557 00 04 00 00 00 00 00 00 00 00 00 00
16558 00 04 00 00 00 00 00 00 00 00 00 00
16559 00 04 00 00 00 00 00 00 00 00 00 00
16560 00 04 00 00 00 00 00 00 00 00 00 00
16561 00 04 00 00 00 00 00 00 00 00 00 00
16562 00 04 00 00 00 00 00 00 00 00 00 00
16563 00 04 00 00 00 00 00 00 00 00 00 00
16564 00 04 00 00 00 00 00 00 00 00 00 00
16565 00 04 00 00 00 00 00 00 00 00 00 00
16566 00 04 00 00 00 00 00 00 00 00 00 00
16567 00 04 00 00 00 00 00 00 00 00 00 00
16568 00 04 00 00 00 00 00 00 00 00 00 00
16569 00 04 00 00 00 00 00 00 00 00 00 00
16570 00 04 00 00 00 00 00 00 00 00 00 00
16571 00 04 00 00 00 00 00 00 00 00 00 00
16572 00 04 00 00 00 00 00 00 00 00 00 00
16573 00 04 00 00 00 00 00 00 00 00 00 00
16574 00 04 00 00 00 00 00 00 00 00 00 00
16575 00 04 00 00 00 00 00 00 00 00 00 00
16576 00 04 00 00 00 00 00 00 00 00 00 00

```

Grupo 2 – 16577 a 16590 (40C1 a 40CE em hexa)

```

40C1 01 00 00 00 LD BC,0000
40C4 ED 43 7B 40 LD (CONT),BC
40C8 ED 43 99 40 LD (TEX+1),BC
40CC 01 98 40 LD BC,TEX

```

Agora zeramos o contador (equivalente ao bloco 7) e os dois últimos bytes de TEX. Isto é necessário para evitar que ao reiniciarmos um jogo, permaneçam as informações do anterior.

```

16577 01 00 00 ED 43 7B 40 402
16584 ED 43 99 40 01 98 40 708

```

Grupo 3 – 16591 a 16618 (40CF a 40EA em hexa)

```

40CF 2A 10 40 LD HL,(VARS)
40D2 23 INC HL
40D3 7E LD A,(HL)
40D4 7E 04 CP 04
40D5 70 RET P
40D7 A7 AND A
40D8 C8 RET Z
40D9 57 LD D,A
40DA 23 INC HL
40DB 7E LD A,(HL)
40DC A7 AND A

```

```

40DD 00      RET  NZ
40DE 03      INC  HL
40DF 7E      LD   A, (HL)
40E0 7E 28    CP   28
40E2 70      RET  P
40E3 7E 10    CP   10
40E5 78      RET  M
40E6 00      LD   (BC), A
40E7 03      INC  BC
40E8 15      DEC  D
40E9 20 F3    JR   NZ, 40DE

```

Este grupo corresponde ao bloco 3, De início tomou-se o valor registrado no campo das variáveis (VARS) que acabou de "entrar" (Input F\$).

A rotina do Sinclair localiza a String através da Variável de Sistema VARS (16400 ou 4010 em hexa) que informa o endereço onde ela foi armazenada. O byte deste endereço contém o código da letra associada ao string, somado com 40, os dois seguintes são o comprimento da string, primeiro o menos significativo e o segundo mais significativo. Então em nossa rotina, ao fazer Inc HL — LD A, (HL) — CP 4 — Ret P — And A — Ret Z — Inc HL — LD A, (HL) — And A — RetNZ, estamos verificando se o comprimento é maior que zero, mas menor que 3, e em caso contrário, retornando ao basic. Neste caso tornamos o V da instrução 740 diferente de zero já que, nesta situação, BC já contém o endereço de TEX.

Os bytes seguintes contém os dados da própria "String" que entrou, e as demais instruções de grupo verificam se eles são realmente números e os carregam no registro TEX.

```

166901 2A 10 40 23 7E FE 04 541
166908 F0 A7 08 57 23 7E A7 1022
166905 00 23 7E FE 06 F0 FE 1139
166912 10 78 0E 03 15 20 F3 577

```

Grupo 4 — 16619 a 16752 (40EB a 4170 em hexa)

```

40EB 0D 23 0F      CALL  FAST
40EE 21 06 02      LD   HL, 784(D)
40F1 3E 15          LD   A, 21(D)
40F3 07           RST  16
40F4 0B          DEC  HL
40F5 7C          LD   A, H
40F6 59          OR   L
40F7 20 F8       JR   NZ, 40F1
40F9 2A 0C 40    LD   HL, (D+FILE)
40FC 0D 2F 41    CALL  412F
40FF 23          INC  HL
4100 0D 47 41    CALL  4147
4103 01 53 02    LD   BC, 595(D)

```

```

4106 09          ADD  HL, BC
4107 0D 47 41    CALL  4147
410A 23          INC  HL
410B 0D 2F 41    CALL  412F
410E 01 92 02    LD   BC, 658(D)
4111 07          AND  A
4112 0D 42      SBC  HL, BC
4114 3E 10      LD   A, 28(D)
4116 0D 57 41    CALL  4157
4119 3C          INC  A
411A 0D 57 41    CALL  4157
411D 01 51 02    LD   BC, 593(D)
4120 07          AND  A
4121 0D 42      SBC  HL, BC
4123 0D 65 41    CALL  4165
4126 3E 10      LD   A, 28(D)
4128 0D 65 41    CALL  4165
412B 01 00 00    LD   BC, 0000
412E 00          RET
412F 06 03      LD   B, 03
4131 23          INC  HL
4132 36 08      LD   (HL), 08
4134 10 78      DJNZ  4131
4136 06 1A      LD   B, 26(D)
4138 3E 25      LD   A, 37(D)
413A 23          INC  HL
413B 3C          INC  A
413C 77          LD   (HL), A
413D 10 FB      DJNZ  413A
413F 06 03      LD   B, 03
4141 23          INC  HL
4142 36 08      LD   (HL), 08
4144 10 78      DJNZ  4141
4146 0D          RET
4147 23          INC  HL
4148 36 08      LD   (HL), 08
414A 23          INC  HL
414B 36 08      LD   (HL), 08
414D 01 1D 00    LD   BC, 29(D)
4150 00          ADD  HL, BC
4151 36 08      LD   (HL), 08
4153 23          INC  HL
4154 36 08      LD   (HL), 08
4156 00          RET
4157 06 09      LD   B, 09
4159 11 1E 00    LD   DE, 30(D)
415C 77          LD   (HL), A
415D 10          ADD  HL, DE
415E 77          LD   (HL), A
415F 23          INC  HL
4160 00          INC  HL
4161 23          INC  HL

```

```

4160 10 F8      DJNZ 415C
4161 00      RET
4162 00 00    LD  B,00
4163 00 00    LD  (HL),A
4164 00 00    ADD HL,DE
4165 00 00    LD  (HL),A
4166 00 00    INC HL
4167 00 00    INC HL
4168 00 00    INC HL
4169 00 00    INC A
4170 00 00    DJNZ 4167
4171 00 00    RET

```

Como vê-se este grupo está dividido em um básico de 16619 a 16686 (40EB a 412E em hexa) e quatro sub-rotinas de 412F a 4146, de 4147 a 4156, de 4157 a 4164 e, finalmente, de 4165 a 4170.

Existem alguns pontos que merecem comentários: a primeira instrução do grupo básico — CD230F — é uma chamada à rotina da ROM OF23, que corresponde ao comando Fast. Este tipo de procedimento, o uso de rotinas prontas da ROM, é muito comum em programação em linguagem de máquina. Por isso temos no apêndice B a lista de rotinas disponíveis na ROM.

O conjunto seguinte, LD HL, 704 — LD A, 21 — RST 10 — Dec HL — LD A, H — ORL — JRNZ, é também muito importante. Nele é usada a instrução RST 10 que nada mais é do que uma chamada (CALL) à sub-rotina que se inicia no endereço 10 da ROM, e que corresponde ao comando de imprimir na tela o que se encontra em A. A sub-rotina já garante que não será ignorado o byte contendo 118 (76 em hexa) ao fim de cada linha; por isso é que é suficiente carregar HL com 704 (= 32 x 22) e não 726 (= 33 x 22). Com esta seqüência também garante-se, caso o equipamento tenha menos que 3,5 k, que o arquivo da tela corresponda a mesma totalmente "cheia".

As demais instruções e sub-rotinas não são difíceis de entender. Tendo o arquivo da tela totalmente "cheio" como se garantiu no conjunto anterior, podemos compor a tela através de "cargas" (Load) adequadas, sem necessidade do uso de RST10. É o que se está fazendo por exemplo, com a seqüência LD HL (DFile) — LD B, 3 — Inc HL — LD (HL), 08 — DJNZ, em que se coloca na tela 3 vezes o sinal gráfico correspondente ao código 08.

Ainda é neste grupo de instruções que pode ser visto a importância de fazer o esboço da tela. A contagem dos espaços para sair de uma posição e chegar a outra, onde se "carregará" um sinal gráfico qualquer, é bastante facilitada com o uso do esboço. Note que um erro de posicionamento pode desorganizar o arquivo da tela. Isto é especialmente verdadeiro quando se carrega na posição de fim de linha, um valor que não seja 118 (76 em hexa).

```

16686 00 00    DJNZ 1667F
16687 00 00    RET
16688 00 00    LD  B,00
16689 00 00    LD  (HL),A
16690 00 00    ADD HL,DE
16691 00 00    LD  (HL),A
16692 00 00    INC HL
16693 00 00    INC HL
16694 00 00    INC HL
16695 00 00    INC A
16696 00 00    DJNZ 16695
16697 00 00    RET
16698 00 00    LD  B,00
16699 00 00    LD  (HL),A
16700 00 00    ADD HL,DE
16701 00 00    LD  (HL),A
16702 00 00    INC HL
16703 00 00    INC HL
16704 00 00    INC HL
16705 00 00    INC A
16706 00 00    DJNZ 16705
16707 00 00    RET
16708 00 00    LD  B,00
16709 00 00    LD  (HL),A
16710 00 00    ADD HL,DE
16711 00 00    LD  (HL),A
16712 00 00    INC HL
16713 00 00    INC HL
16714 00 00    INC HL
16715 00 00    INC A
16716 00 00    DJNZ 16715
16717 00 00    RET
16718 00 00    LD  B,00
16719 00 00    LD  (HL),A
16720 00 00    ADD HL,DE
16721 00 00    LD  (HL),A
16722 00 00    INC HL
16723 00 00    INC HL
16724 00 00    INC HL
16725 00 00    INC A
16726 00 00    DJNZ 16725
16727 00 00    RET
16728 00 00    LD  B,00
16729 00 00    LD  (HL),A
16730 00 00    ADD HL,DE
16731 00 00    LD  (HL),A
16732 00 00    INC HL
16733 00 00    INC HL
16734 00 00    INC HL
16735 00 00    INC A
16736 00 00    DJNZ 16735
16737 00 00    RET
16738 00 00    LD  B,00
16739 00 00    LD  (HL),A
16740 00 00    ADD HL,DE
16741 00 00    LD  (HL),A
16742 00 00    INC HL
16743 00 00    INC HL
16744 00 00    INC HL
16745 00 00    INC A
16746 00 00    DJNZ 16745
16747 00 00    RET

```

Grupo 5 — 16753 a 16836 (4171 a 41C4 em hexa)

```

4171 21 3C 40    LD  HL,TRX
4172 00 00      LD  B,00
4173 4E 00      LD  C,(HL)
4174 10 00      LD  D,00
4175 00 00      INC HL
4176 0E 00      LD  E,(HL)
4177 0A 00 40    LD  HL,(D-FILE)
4178 00 00      ADD HL,BC
4179 06 21      LD  B,33(D)
4180 00 00      ADD HL,DE
4181 10 FD      DJNZ 4181
4182 3E 00      LD  A,128(D)
4183 1E 21      LD  E,33(D)
4184 00 00      AND A
4185 0D 52      SBC HL,DE
4186 2B 00      DEC HL
4187 0D B2 41    CALL 41B2
4188 23 00      INC HL
4189 0D B2 41    CALL 41B2
4190 23 00      INC HL
4191 0D B2 41    CALL 41B2
4192 23 00      INC HL
4193 0D B2 41    CALL 41B2
4194 23 00      INC HL
4195 0D B2 41    CALL 41B2
4196 23 00      INC HL
4197 2B 00      DEC HL
4198 0D B2 41    CALL 41B2
4199 2B 00      DEC HL
4200 0D B2 41    CALL 41B2
4201 2B 00      DEC HL

```

```

41A7 00 00 00 00 AND A
41A8 00 00 00 00 SBC HL,DE
41A9 00 00 00 41 CALL 41B2
41AA 00 00 00 00 INC HL
41AB 00 00 00 41 CALL 41B2
41AC 00 00 00 00 RET
41AD 00 00 00 00 PUSH AF
41AE 00 00 00 00 ADD A,(HL)
41AF 00 00 00 77 LD (HL),A
41B0 00 01 00 00 LD BC,0000
41B1 00 00 00 00 DEC BC
41B2 00 00 00 70 LD A,C
41B3 00 00 00 00 OR B
41B4 00 00 00 FB JR NZ,41B8
41B5 00 00 00 00 POP AF
41B6 00 00 00 00 RET
41B7 00 00 00 00 NOP
41C0 00 00 00 00 NOP
41C1 00 00 00 00 NOP
41C2 00 00 00 00 NOP
41C3 00 00 00 00 NOP
41C4 00 00 00 00 NOP

```

Quando este programa foi inicialmente criado, pretendia-se usar uma única sub-rotina — a do bloco 6 — para gerar a tela (anexa ao bloco 5) como para “revertê-la” (anexa ao bloco 10). Posteriormente, para ganhar tempo e melhorar o programa, foi introduzida uma sub-rotina própria — com início em 4254 — para gerar a tela. O grupo 5 corresponde então ao bloco 6 associado à reversão (bloco 10).

Três pontos merecem destaque neste grupo:

* A forma como está sendo feita a comunicação entre o basic (Poke 16447, x) e a linguagem de máquina [LD HL, TRX — LDB, 00 — LDC, (HL)].

* A sistemática de “consumir” tempo que é extremamente usada em programas, aplicável a qualquer par de registradores (no caso usamos o BC) — LD BC, 0300 — DEC BC — LD A, C — OR B — JRNZ — onde o valor que se carrega os registradores, 0300 no caso, deve ser ajustado de forma a causar o efeito desejado.

* Os 6 bytes vagos ao fim da sub-rotina ou do grupo são uma reserva para a eventualidade de ser necessária uma correção no programa, e prevê-la é uma prática muito recomendável mesmo para programadores bastante experientes.

```

16753 21 30 40 05 00 4E 15 253
16754 00 03 5E 2A 0C 40 09 256
16757 05 21 19 10 FD 3E 80 323
16774 1E 21 A7 ED 52 2B CD 767
16781 B2 41 23 CD B2 41 23 781
16788 CD 52 41 19 CD B2 41 921

```

```

16779 00 00 00 00 41 00 00 00
16780 00 00 00 00 00 00 00 00
16781 00 00 00 00 00 00 00 00
16782 00 00 00 00 00 00 00 00
16783 00 00 00 00 00 00 00 00
16784 00 00 00 00 00 00 00 00
16785 00 00 00 00 00 00 00 00
16786 00 00 00 00 00 00 00 00
16787 00 00 00 00 00 00 00 00
16788 00 00 00 00 00 00 00 00
16789 00 00 00 00 00 00 00 00
16790 00 00 00 00 00 00 00 00
16791 00 00 00 00 00 00 00 00
16792 00 00 00 00 00 00 00 00
16793 00 00 00 00 00 00 00 00
16794 00 00 00 00 00 00 00 00
16795 00 00 00 00 00 00 00 00
16796 00 00 00 00 00 00 00 00
16797 00 00 00 00 00 00 00 00
16798 00 00 00 00 00 00 00 00
16799 00 00 00 00 00 00 00 00

```

Grupo 6 — 16837 a 16927 (41C5 a 421F em hexa)

```

41C5 00 00 10 40 LD HL,(VAR5)
41C6 00 00 00 00 INC HL
41C7 00 00 00 00 LD A,(HL)
41C8 00 00 00 00 CP 00
41C9 00 00 00 00 LD BC,0000
41CA 00 00 00 00 RET M
41CB 00 00 00 00 LD A,A
41CC 00 00 00 00 INC HL
41CD 00 00 00 00 LD A,(HL)
41CE 00 00 00 00 AND A
41CF 00 00 00 00 RET NZ
41D0 00 00 00 00 INC HL
41D1 00 00 00 00 LD A,(HL)
41D2 00 00 00 00 AND A
41D3 00 00 00 00 RET NZ
41D4 00 00 00 00 INC HL
41D5 00 00 00 00 LD A,(HL)
41D6 00 00 00 40 CP 04(D)
41D7 00 00 00 00 RET P
41D8 00 00 00 26 CP 38(D)
41D9 00 00 00 00 RET M
41DA 00 00 00 22 SUB 34(D)
41DB 00 00 00 40 LD (TRX),A
41DC 00 00 00 00 INC HL
41DD 00 00 00 00 LD A,(HL)
41DE 00 00 00 00 DEC A
41DF 00 00 00 00 DEC A
41E0 00 00 00 00 JR NZ,41FE
41E1 00 00 00 10 CP 29(D)
41E2 00 00 00 00 RET M
41E3 00 00 00 26 CP 38(D)
41E4 00 00 00 00 RET P
41E5 00 00 00 16 SUB 27(D)
41E6 00 00 00 07 JR 4201
41E7 00 00 00 00 DEC A
41E8 00 00 00 00 RET NZ
41E9 00 00 00 10 CP 29(D)
41EA 00 00 00 00 RET NZ
41EB 00 00 00 00 INC HL
41EC 00 00 00 00 LD A,(HL)
41ED 00 00 00 10 CP 28(D)
41EE 00 00 00 00 RET M
41EF 00 00 00 25 CP 37(D)
41F0 00 00 00 00 RET P
41F1 00 00 00 11 SUB 17(D)
4201 00 00 00 40 LD (TRY),A
4204 00 00 00 41 CALL 4171

```

```

42007 ED 5B 7B 40 LD DE,(CONT)
42008 ED 13 INC DE
4200C ED 53 7B 40 LD (CONT),DE
42010 0A 00 40 LD HL,(D-FILE)
42013 01 05 02 LD BC,726(D)
42018 03 05 LD A,05
42019 ED 01 CP IR
4201A 00 04 JR NZ,4220
4201C 01 00 00 LD BC,0000
4201F 00 RET

```

Neste grupo está se repetindo rotinas já conhecidas: do início até 16899 (4203 em hexa) tem-se o mesmo teste básico usado no grupo 3 e que corresponde ao bloco 9 do diagrama de blocos; em seguida há a chamada à subrotina de reversão (grupo 5) e o incremento no contador, para finalmente chegar-se ao bloco 11 de verificação de retorno à tela básica, cujas explicações e instruções foram dadas quando "melhoramos" o programa original totalmente em basic.

```

160007 00 00 40 40 00 7F 7F 00 00 00 00
160008 00 00 44 44 00 00 00 00 00 00 00
160009 00 00 41 41 00 00 00 00 00 00 00
16000A 00 00 47 47 00 00 00 00 00 00 00
16000B 00 00 40 40 00 00 00 00 00 00 00
16000C 00 00 44 44 00 00 00 00 00 00 00
16000D 00 00 40 40 00 00 00 00 00 00 00
16000E 00 00 47 47 00 00 00 00 00 00 00
16000F 00 00 40 40 00 00 00 00 00 00 00
160010 00 00 44 44 00 00 00 00 00 00 00
160011 00 00 47 47 00 00 00 00 00 00 00
160012 00 00 40 40 00 00 00 00 00 00 00
160013 00 00 44 44 00 00 00 00 00 00 00
160014 00 00 47 47 00 00 00 00 00 00 00
160015 00 00 40 40 00 00 00 00 00 00 00

```

Grupo 7 — 16928 a 16979 (4220 a 4253 em hexa)

```

42020 D5 PUSH DE
42021 DA 00 40 LD HL,(D-FILE)
42024 11 0A 00 LD DE,000A(D)
42027 40 00 ADD HL,DE
42028 00 00 EX DE,HL
42029 00 00 LD HL,4080
4202A 01 10 00 LD BC,20(D)
4202B 00 00 LDIR
4202C 00 00 EX DE,HL
4202D 01 0E 00 LD BC,40(D)
4202E 00 00 ADD HL,BC
4202F 00 00 EX DE,HL
42030 01 10 00 LD BC,10(D)
4203A 00 00 LDIR
4203C 00 00 EX DE,HL
4203D 01 11 00 LD BC,17(D)
42040 00 00 ADD HL,BC

```

```

42441 EB EX DE,HL
42442 01 04 00 LD BC,4(D)
42445 ED B0 LDIR
42447 EB EX DE,HL
42448 01 7F 00 LD BC,127(D)
42449 00 00 ADD HL,BC
4244C ED EX DE,HL
4244D 01 12 00 LD BC,12(D)
42450 ED B0 LDIR
42452 C1 POP BC
42453 C0 RET

```

Este grupo corresponde ao bloco 12. Trata-se apenas de compor a tela final, usando os dizeres preparados no grupo 1, que como foi visto, já contém a informação do nível de dificuldade eleito pelo jogador.

Adicionalmente cabe chamar atenção sobre a técnica utilizada na transcrição, com o uso da instrução LDIR, bem como o fato de carregarmos em BC o valor contido no contador de forma a levar para o basic, através da variável V, o número de jogadas utilizadas.

```

160030 05 00 00 00 40 11 0A 00 00 00
160031 00 10 00 00 00 01 10 00 00
160032 00 00 00 00 00 01 00 00 00
160033 00 00 00 00 00 00 00 00 00
160034 00 00 00 00 10 00 00 00 00
160035 00 00 00 00 01 00 00 00 00
160036 00 00 00 00 01 00 00 00 00
160037 00 00 00 00 01 12 00 00 00
160038 00 00 00 00 01 12 00 00 00
160039 00 00 00 00 01 12 00 00 00

```

Grupo 8 — 16980 a 17047 (4254 a 4297 em hexa)

```

42054 01 0C 40 LD HL,TRX
42057 08 00 LD B,00
42058 4E 00 LD C,(HL)
42059 15 00 LD D,00
4205C 00 00 INC HL
4205D 0E 00 LD E,(HL)
4205E 0A 00 40 LD HL,(D-FILE)
42061 00 00 ADD HL,BC
42062 00 01 LD B,01
42064 10 00 ADD HL,DE
42065 10 7D DJNZ 42064
42067 1E 01 LD E,01
42068 0E 00 LD A,00
4206B 00 00 ADD A,(HL)
4206C 77 00 LD (HL),A
4206D 0B 00 DEC HL
4206E 0E 00 LD A,00
42070 00 00 ADD A,(HL)
42071 77 00 LD (HL),A
42072 A7 00 AND A
42073 ED 52 SBC HL,DE

```

```

4275 3E 80 LD A,80
4277 86 ADD A,(HL)
4278 77 LD (HL),A
4279 23 INC HL
427A 3E 80 LD A,80
427C 86 ADD A,(HL)
427D 77 LD (HL),A
427E 23 INC HL
427F 3E 80 LD A,80
4281 86 ADD A,(HL)
4282 77 LD (HL),A
4283 19 ADD HL,DE
4284 3E 80 LD A,80
4286 86 ADD A,(HL)
4287 77 LD (HL),A
4288 19 ADD HL,DE
4289 3E 80 LD A,80
428B 86 ADD A,(HL)
428C 77 LD (HL),A
428D 2B DEC HL
428E 3E 80 LD A,80
4290 86 ADD A,(HL)
4291 77 LD (HL),A
4292 2B DEC HL
4293 3E 80 LD A,80
4295 86 ADD A,(HL)
4296 77 LD (HL),A
4297 09 RET

```

Como já foi explanado, aqui temos uma repetição da sub-rotina 6 (grupo 5) que foi feita sem dispêndio de tempo, visando agilizar a montagem da tela inicial.

Com este grupo de instruções finalizamos o programa em linguagem de máquina. Após carregar e rodá-lo notar-se-á que ele não apresenta grande vantagem em desempenho se for comparado com versão em basic, "melhorada" através do acréscimo da sub-rotina de verificação do final. O ponto de maior realce, além do cunho didático, é a extensão do programa, que na última versão cabe perfeitamente em 2 Kbytes e nas primeiras exigia capacidade além desta grandeza.

```

160080 01 00 00 4E 16 00
160087 00 00 00 4E 16 00
160094 00 01 00 7D 16 00
170001 00 00 00 00 00 00
170008 00 77 00 00 00 00
170015 00 77 00 00 00 00
170022 00 00 00 77 00 00
170029 00 00 00 77 00 00
170036 00 00 00 77 00 00
170043 00 00 00 77 00 00

```

RESUMO

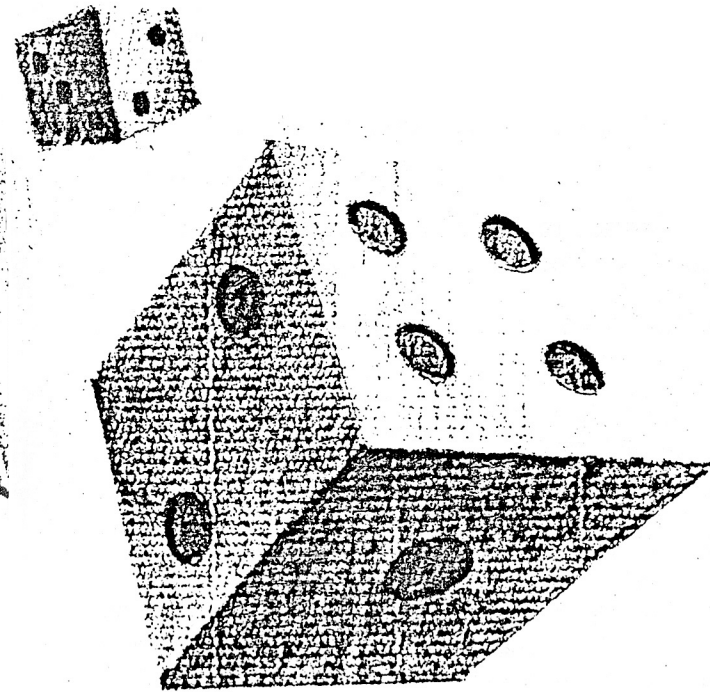
Foi abordado no presente capítulo o seguinte:

- * O arquivo da tela
- * Variáveis do Sistema: DFile, Vars e DFSZ
- * Técnicas de passagem de dados do Basic para rotinas em Linguagem de máquina e vice-versa.
- * Definição de "registros improvisados" e locais interessantes para sua armazenagem
- * Uso de diagrama de blocos e do esboço da tela.
- * Apresentação de rotina para consumir tempo

SUGESTÕES

- 1 - Procurar entender porque é indispensável a instrução 855 CLEAR no basic 2.
- 2 - Se seu equipamento tiver somente 2K procurar modificar o programa de forma a englobar, na linguagem de máquina, as instruções 710 CLS e 720 PRINT "NÍVEL DE DIFICULDADE"; AT 0,0;. Caso tenha mais que 2K, introduzir também telas com instruções do jogo, usando somente linguagem de máquina.

O TECLADO



Aqui vemos como utilizar a rotina de leitura do teclado simulando o INKEY\$ em linguagem de máquina. Tudo que ora se discute é exemplificado na criação do jogo "Crepe" onde se mostra também uma maneira muito original de se criar números ao acaso sem que a sorte seja ludibriada. Alea jacta est.

CAPÍTULO 2 – O TECLADO

A pesquisa ou "varredura" do teclado é feita, nos computadores da linha Sinclair, através de uma sub-rotina da ROM conhecida por KScan, abreviatura do inglês Keyboard Scan.

Esta sub-rotina acha-se situada no endereço 699, ou 02BB em hexadecimal, e similarmente a qualquer outra, pode ser chamada através da instrução Call (CDBB02). Como em sua utilização todos os registradores são afetados, se o conteúdo de algum deles precisar ser preservado, é necessário acumulá-lo no "stack" através da instrução PUSH, e após o Call, buscar a informação de volta através do POP.

O resultado da aplicação do KScan é encontrado no par HL, que funciona como um apontador ou indicador de qual tecla foi digitada, se é que alguma foi. Para entender como achar a tecla pressionada através do resultado acumulado em HL devemos considerar o teclado dividido em grupos. No caso do registrador L tem-se os seguintes grupos e correspondentes valores do mesmo:

GRUPO DE TECLAS	VALOR DE L
Nenhuma foi pressionada	FF
Z – X – C – V	FE
A – S – D – F – G	FD
Q – W – E – R – T	FB
1 – 2 – 3 – 4 – 5	F7
6 – 7 – 8 – 9 – 0	EF
Y – U – I – O – P	DF
H – J – K – L – (ENTER ou NEW LINE)	BF
B – N – M – (.,) – (Space)	7F

Como vê-se a tecla "Shift" não participa de grupo algum dos citados. Isto significa que o fato dela estar pressionada ou não, em nada altera o valor de L. O mesmo conceito se aplica ao registrador H, só que neste caso, seu valor muda se a tecla "Shift" estiver ou não pressionada.

GRUPOS DE TECLAS

	VALOR DE H	
	Sem "Shift"	Com "Shift"
Nenhuma (além do "Shift")	FF	FE
A – Q – 1 – 0 – P – (Enter) – (Space)	FD	FC
Z – S – W – 2 – 9 – O – L – (.,)	FB	FA
X – D – E – 3 – 8 – I – K – M	F7	F6
C – F – R – 4 – 7 – U – J – N	EF	EE
V – G – T – 5 – 6 – Y – H – B	DF	DE

Observando o teclado do equipamento e localizando os grupos citados, será fácil notar que o sistema é organizado como uma rede sobreposta ao conjunto. Dessa forma, só existe uma tecla que pertence simultaneamente a dois dos grupos citados, o que corresponde a uma relação direta entre a tecla e o resultado em HL. Exemplificando, pressionar a tecla A, sem "Shift", resulta em HL = FDFD, e em correspondência, não é possível chegar-se a este valor pressionando qualquer outra, ou conjunto de outras teclas.

O funcionamento desta sub-rotina é similar ao da função INKEY\$ do Basic, a qual aliás faz uso da KScan. Assim, num programa no qual se deseje que o sistema aguarde o pressionar de uma tecla, é preciso incluir um conjunto de instruções equivalente a

```
100 IF INKEY$="" THEN GOTO 100
```

Caso isso não seja feito, ao atingir o KScan ter-se-á apenas o valor FFFF em HL e as instruções seguintes serão executadas sem maiores delongas.

No programa que vem a seguir, chamado "CREPE", existe uma rotina de espera incluída além da ilustração de outros conceitos que foram apresentados.

O "CREPE" é uma versão para o computador do famoso jogo de azar homônimo, também conhecido por "seven-eleven". Para quem não frequenta cassinos descrevemos as regras básicas.

* Um dos participantes do jogo assume a posição de "banqueiro" e enfrenta apostas dos demais.

* Cabe à banca ou banqueiro o lançamento de um par de dados.

* A rodada se define em duas situações:

** No primeiro lance de dados, se a soma dos pontos for 2, 3 ou 12 a banca perde e se for 7 ou 11 a banca ganha.

** Não se resolvendo a rodada no primeiro lance, ou seja, se a soma dos pontos obtidos tiver sido um dos seguintes 4, 5, 6, 8, 9 ou 10, ele será chamado de "ponto" e a banca joga com o mesmo contra o 7. Isto significa que o banqueiro irá lançar os dados tantas vezes quantas forem necessárias até sair

o valor do "ponto" novamente, caso em que ele ganha, ou sair o 7, situação em que a banca perde.

Isto posto, cabe seguir a sistemática recomendada e já exposta no capítulo anterior, com a apresentação do diagrama de blocos, figura 5, e o esboço da tela na figura 6.

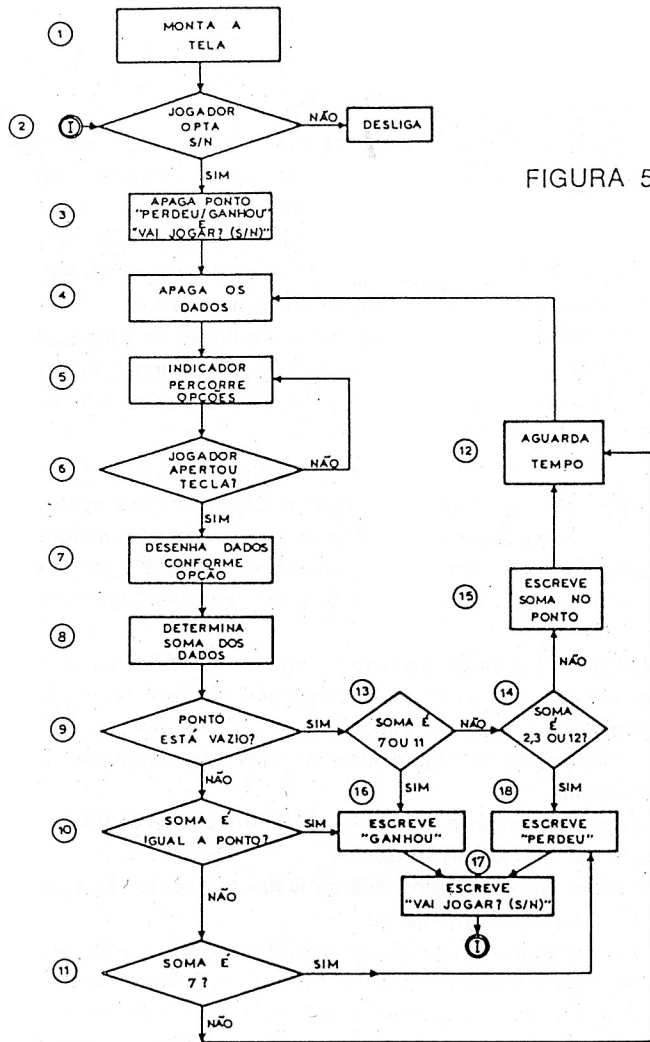


FIGURA 5

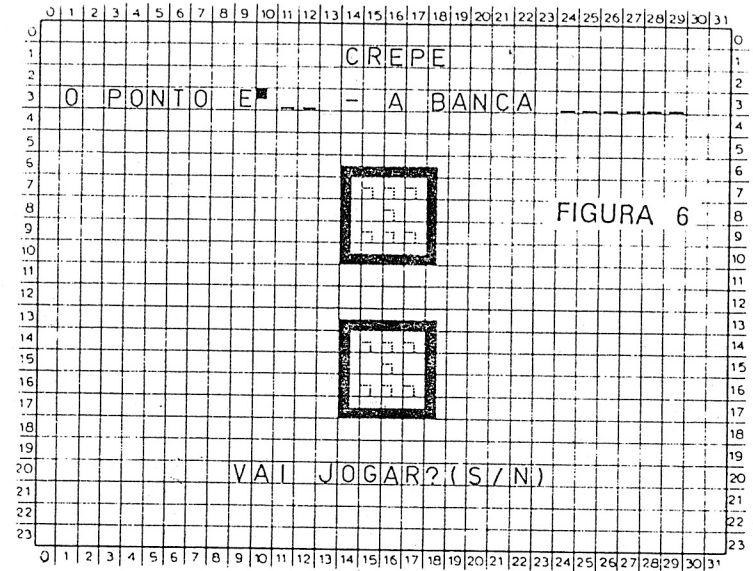


FIGURA 6

O basic correspondente se resume a

```

0 REM ( 714 BYTES )
10 SAVE "CREPE"
20 RAND USR 16566
  
```

É fácil notar que o programa em si é inteiramente escrito em linguagem de máquina, e não necessita mais que 2 Kbytes de capacidade para rodar. Apesar do desempenho ser simples, foram utilizadas técnicas requintadas em sua elaboração o que o torna bastante didático.

Na concepção do programa foi assumido que o equipamento tenha capacidade menor que 3,5 Kbytes, ou mais exatamente 2 Kbytes. Como foi explicado no capítulo anterior, a diferença básica ocasionada pelas diferentes capacidades é a forma de manipular a tela. Justamente para enfatizar este ponto é que adotou-se este critério. Assim se o microcomputador tiver mais que 2 Kbytes é necessário alterar a Variável do Sistema conhecida por Ram Top ou RTP, cuja localização está em 16388/9, dando uma instrução direta Poke 16389, 72 seguido por NEW. Com isso o comportamento da máquina será similar ao de uma com somente 2 Kbytes.

Aqui cabe uma explicação — a variável RTP guarda o endereço do último byte disponível na memória. Daí a razão do nome Ram Top. Este endereço está armazenado na forma tradicional, ou "invertida", usada na lógica Sinclair. Assim se houver 16 Kbytes disponíveis, em 16388 haverá um zero e em

16389 o valor 128. Ora $128 \times 256 + 0 = 32768$ que subtraído do início da Ram (16384), resulta em 16384 bytes disponíveis, que nada mais são que 16 vezes 1 Kbyte, já que 1 Kbyte = 1024 bytes. Ao dar o comando Poke 16389, 72 obtém-se $72 \times 256 + 0 = 18432$ que subtraído do início da Ram (16384) resulta em 2048 bytes ou 2 Kbytes.

O comando NEW que segue é indispensável para que o sistema se consolide na nova configuração, agindo como tendo somente os 2 Kbytes disponíveis.

Após os comandos citados carrega-se o programa, usando o LOAD. Se então gravarmos a fita usando o GOTO 10 toda a vez que voltarmos a carregar o programa com o LOAD, ele entrará em execução automática e se comportará adequadamente mesmo que não se tenha previamente executado a alteração da RTP descrita. Isto já não será verdade, se ao invés do GOTO 10, for utilizado o comando direto SAVE "CREPE". Convém analisar o que foi dito e entender bem a razão desta forma de ser.

Como no exemplo do capítulo anterior vai-se agora desmembrar o programa em grupos de instruções, correlacionando-os com os blocos do diagrama da figura 5.

Grupo 1 — 16514 a 16549 (4082 a 40A5 em hexa)

Aqui estão contidos os 36 resultados possíveis, ou opções, do lançamento dos dois dados, armazenados da seguinte forma: em cada byte os primeiros 4 bits contêm o resultado de um dos dados, que é um número de 1 a 6, e os 4 restantes o do outro dado.

Uma das opções será sorteada e utilizada para montar a figura dos dados na tela.

```

16514 14 10 00 05 09 01 03 04 07
16515 01 04 05 04 02 06 00 03 00
16516 00 01 00 03 14 00 04 00 00
16517 00 00 10 00 01 04 00 04 00
16518 00 04 00 04 01 01 00 04 00
16519 00 00 04 01 01 01 00 00 07
16520 00 00 00 00 00 01 00 00 01

```

Grupo 2 — 16550 a 16555 (40A6 a 40AB em hexa)

Nestes bytes está a palavra "Ganhou" que será utilizada para finalização da rodada, no bloco 16.

É uma prática comum e conveniente esta de armazenar no início do programa informações que não participam da lógica, e que são apenas usadas para consulta ou composição das telas de abertura e finalização. Outra alternativa adotada é fazer a armazenagem após o fim do programa, deixando porém, por cautela, alguns bytes de reserva para o caso de ser necessário alguma alteração imprevista no programa.

```
16550 20 26 33 2D 34 3A 255
```

Grupo 3 — 16556 a 16854 (40AC a 41D6 em hexa)

```

40AC E1          POP  HL
40AD 7E          LD   A, (HL)
40AE 23          INC  HL
40AF E5          PUSH HL
40B0 FE FF      CP   FF
40B2 C8          RET  Z
40B3 CD 08 08   CALL PRINT
40B6 18 F4      JR   40AC
40B8 CD AC 40   CALL 40AC

```

```

DO ENDEREÇO 16571 (40BB)
AO ENDEREÇO 16854 (41D6)
ESTA2 ARMazenado A TELA
DE ABERTURA DO JOGO

```

Este grupo contém as instruções e dados para montagem da tela de abertura e corresponde ao bloco 1 da figura 5, ou início do programa.

A lógica utilizada nesta tarefa é bastante interessante. A instrução 20 do Basic faz uma chamada ao endereço 16568, e lá encontramos um Call para o início da lógica, em 16556 (40AC em hexa). Toda vez que um Call é feito, o endereço do byte imediatamente após a instrução é armazenado no "stack" para ser usado por ocasião do retorno. Surgindo a instrução Ret o endereço é retirado do "Stack" (como se fosse um "Pop") e a próxima instrução que é executada é a deste endereço.

Ora, após o CDAC40 está o primeiro byte a ser "escrito" na tela. Ele é então armazenado no "stack", de lá retirado (E1 — Pop HL) e registrado em

A [7E — LD A, (HL)]. O endereço é então incrementado e novamente armazenado no "Stack", substituindo assim o anterior (23/E5 — Inc HL/ Push HL). O valor de A é comparado com FF, que é o indicador escolhido de fim do arquivo de compor a tela (FEFF — CP FF); sendo igual é dado o retorno (C8 — Ret Z) ou um "jump" para a instrução cujo endereço está no stack, que é exatamente a primeira após FF, e corresponde ao início do bloco 2. O próximo passo é novamente um Call (CD0808), que é uma chamada a uma rotina da ROM de endereço 2056 (0808 em hexa), responsável por escrever o conteúdo do registrador A na tela; esta rotina faz uso do RST 10 citado no capítulo anterior. Finalmente tem-se um "jump" relativo para o reinício do ciclo.

Em relação aos dados usados para compor a tela existem vários pontos dignos de nota:

* Toda a vez que o valor de A for 118 (76 em hexa) a rotina CD0808 bem como a RST10 entende que deve interromper a linha e pular para o início da seguinte.

** Existe uma superabundância de zeros a serem impressos. É claro que se poderia reduzir o número de bytes necessários para compor a tela, usando rotinas auxiliares. Mas em favor da didática e por não ser necessário economizar bytes, adotou-se esta apresentação. Como resultado, sabendo que se irá teoricamente trabalhar com uma capacidade de 2 Kbytes, tem-se entre 16571 e 16853 uma cópia fiel do arquivo da tela, o que facilitará o entendimento das etapas seguintes do programa.

*** Também a título ilustrativo de recursos, e não para economizar memória neste caso, escolheu-se armazenar as palavras "Perdeu" e "Vai jogar? (S/N)" nos mesmos locais em que se reservaria espaço no arquivo da memória para escrevê-las, ou seja, nos endereços 16617 (40E9 em hexa) a 16839 (41C7 em hexa) respectivamente. Como conseqüência, ao montar a tela ela aparece com a indicação equivalente a de já ter havido uma rodada anterior em que a banca perdeu.

```

16617 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16618 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16619 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16621 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16622 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16623 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16624 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16625 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16626 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16627 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16628 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16629 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16630 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16631 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16632 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16633 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16634 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16635 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16636 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16637 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16638 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16639 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```

16640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16641 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16642 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16643 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16644 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16645 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16646 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16647 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16648 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16649 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16651 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16652 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16653 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16654 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16655 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16656 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16657 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16658 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16659 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16660 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16661 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16662 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16663 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16664 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16665 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16666 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16667 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16668 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16669 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16670 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16671 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16672 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16673 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16674 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16675 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16676 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16677 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16678 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16679 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Grupo 4 — 16855 a 16879 (41D7 a 41EF em hexa)

```

41D7  CD  BB  02          CALL KSCAN
41DA  11  7F  EF          LD DE,EF7F
41DD  A7                    AND A
41DE  ED  52          SBC HL,DE
41E0  CA  00  00          JP Z,0000
41E3  11  7E  0C          LD DE,0C7E
41E6  A7                    AND A
41E7  ED  52          SBC HL,DE
41E9  BC                    JR NZ,41D7
41EB  00                    NOP
41EC  00                    NOP
41ED  00                    NOP
41EE  00                    NOP
41EF  00                    NOP

```

Eis o bloco onde se utiliza a KScan, que é o de número 2.

De início se compara o valor de HL correspondente a tecla pressionada (se é que houve alguma), com EF 7F que corresponde a letra N, através do SBC HL, DE. Sendo zero a diferença, o N teria sido acionado e o "jump" se daria (CA 0000). Pular para o endereço 0000 é equivalente ao RST 0, e a consequência é a mesma que desligar e voltar a ligar o computador, ou seja, a memória é totalmente apagada além mesmo do que o NEW executa, já que com esta última instrução o valor da RamTop e tudo que lhe fica acima é preservado.

Após a subtração o HL fica alterado e o valor que se carrega então em DE é a diferença entre o valor associado à tecla S (FBFD) e o da N (EF7F). Então executando nova subtração, se o resultado for zero é sinal que a tecla S foi pressionada e o programa segue para o bloco seguinte. Em caso contrário o JR NZ faz retornar o sistema para oKScan, e assim fica formado um laço ou "loop" que só é "quebrado" quando uma das duas teclas S ou N, for acionada.

Os cinco bytes em branco seguintes são reserva para eventuais alterações.

```
168855 0D 5B 00 11 7F EF A7 044
168860 ED 5D 0F 00 00 11 7E 064
168865 0C D7 ED 5D 00 FC 00 766
168870 00 00 00 00 00 00 00 0
```

Grupo 5 – 16880 a 16910 (41F0 a 420E em hexa)

```
41F0 2A 0C 40 LD HL, (D-FILE)
41F3 11 22 00 LD DE, 34 (D)
41F5 AF XOR A
41F7 19 ADD HL, DE
41F8 77 LD (HL), A
41F9 23 INC HL
41FA 77 LD (HL), A
41FB 11 0C 00 LD DE, 12 (D)
41FE 19 ADD HL, DE
41FF 06 06 LD B, 06
4201 77 LD (HL), A
4202 23 INC HL
4203 10 FC DJNZ 4201
4205 11 D8 00 LD DE, 216 (D)
4208 19 ADD HL, DE
4209 06 0F LD B, 15 (D)
420B 77 LD (HL), A
420C 23 INC HL
420D 10 FC DJNZ 420B
```

Este grupo corresponde ao bloco 3 da figura 5, onde alguns dizeres da tela devem ser apagados.

De sugestivo convém enfatizar o endereço ou localização do que deve ser apagado. Se a tela estivesse "cheia" como comumente ocorre, a segunda instrução ao invés de ser 112200 – LD DE, 34 decimal seria 116F00 – LD DE, 111 decimal. Procure entender bem o porquê.

```
168860 2A 0C 40 11 22 00 AF 344
168867 19 77 23 77 11 0C 00 327
168864 19 06 06 77 23 10 FC 400
168861 11 D8 00 19 06 0F 77 308
168868 23 10 FC 00 00 00 00 300
```

Grupo 6 – 16911 a 16954 (420F a 423A em hexa)

```
420F 2A 0C 40 LD HL, (D-FILE)
4210 11 22 00 LD DE, 31 (D)
4213 19 ADD HL, DE
4215 AF XOR A
4217 77 LD (HL), A
4218 23 INC HL
4219 77 LD (HL), A
421A 23 INC HL
421B 77 LD (HL), A
421C 23 INC HL
421D 11 18 00 LD DE, 19 (D)
421F 19 ADD HL, DE
4220 77 LD (HL), A
4221 19 ADD HL, DE
4222 77 LD (HL), A
4223 23 INC HL
4224 77 LD (HL), A
4225 23 INC HL
4226 77 LD (HL), A
4227 11 0C 00 LD DE, 50 (D)
4229 19 ADD HL, DE
422B 77 LD (HL), A
422C 23 INC HL
422D 77 LD (HL), A
422E 23 INC HL
422F 77 LD (HL), A
4230 11 13 00 LD DE, 19 (D)
4233 19 ADD HL, DE
4234 77 LD (HL), A
4235 19 ADD HL, DE
4236 77 LD (HL), A
4237 23 INC HL
4238 77 LD (HL), A
4239 23 INC HL
423A 77 LD (HL), A
```

Temos aqui uma repetição da lógica do grupo anterior. Trata-se do bloco 4 figura 5 em que, ao invés de dizeres, apagam-se os dados.

```

160011 BA 00 40 11 5B 00 10 051
160018 BF 77 03 77 20 77 11 0810
160025 13 00 10 77 10 77 03 0400
160032 77 20 77 11 00 00 10 0700
160039 77 03 77 03 77 11 10 4500
160046 00 10 77 10 77 03 77 4400
160053 03 77 10 77 03 77 10 4404

```

Grupo 7 – 16955 a 16982 (423B a 4256 em hexa)

```

423B CD BB 02 CALL KSCAN
423E 20 INC L
423F 20 FA JR NZ,423B
4241 11 AD 40 LD DE,40A5
4244 10 09 JR 424F
4246 21 7D BF LD HL,BF7D
4249 10 ADD HL,DE
424A 00 F5 JR NC,4241
424C 1B DEC DE
424D 05 PUSH BC
424E 04 POP BC
424F 05 PUSH DE
4250 CD BB 02 CALL KSCAN
4253 D1 POP DE
4254 20 INC L
4255 28 EF JR Z,4246

```

Aí está um grupo muito interessante. Ele corresponde aos dois blocos seguintes da figura 5, quais sejam o 5 e o 6.

De início tem-se uma rotina equivalente à seguinte instrução em Basic:
100 IF INKEY\$ = " " THEN GOTO 100

Isto porque, incrementando o L após o KScan (CDBB02), só teremos zero se HL for FFFF, ou seja, se nenhuma tecla estiver sendo pressionada. Caso contrário o JRNZ obriga o sistema a voltar ao KScan e tem-se um "loop" formado. Com isso, se após a opção do bloco 2 ou no retorno via bloco 12 da opção feita no 6, o jogador mantiver a tecla pressionada, o programa entrará neste "loop", e ficará nele até o jogador liberar a tecla.

A seguir tem-se um novo laço ou "loop" percorrendo as 36 opções do grupo 1, passando por uma varredura do teclado. A varredura está no endereço 4250, e as instruções Call KScan – Inc L – JRZ (CD BB02 – 2C – 28 EF), garantem que enquanto uma tecla não for pressionada o sistema permanece no "loop".

As instruções Push DE antes do KScan e a Pop DE após são indispensáveis pelo que dissemos anteriormente: o KScan altera os valores de todos

registradores e o DE deve ser preservado, já que nele está armazenado um endereço entre os 36 do grupo 1, que representará o sorteio feito.

A forma de percorrer as 36 alternativas é a seguinte: de início carrega-se DE com o último dos 36 endereços (LD DE, 40A5), e vai-se a cada ciclo diminuindo-o de uma unidade (Dec DE) e somando-o a BF 7D (LD HL, BF 7D, ADD HL, DE). Ora, só não ocorrerá o "carry" quando DE for igual a 4082, que é justamente a primeira das 36 possibilidades, situação em que o JRNC faz carregar de novo em DE o 40A5.

Para que o jogo seja "honesto" é preciso que nossos dados eletrônicos não sejam viciados. Teoricamente teríamos um vício ao substituir o lance de dados pelo sorteio dentro de uma seqüência de 36 valores fixos. O que estamos fazendo eletronicamente equivale a ter 36 resultados colocados na periferia de um disco que gira frente a uma seta indicadora; ao parar o disco o resultado sob a seta é o escolhido.

É fácil perceber que, teoricamente, a seqüência em que os valores são colocados, e o ponto que está sob a seta ao se iniciar a rotação do disco têm influência no resultado do sorteio. Porém a velocidade de rotação do disco é extremamente alta. Por menor que seja o intervalo entre duas pressionadas do teclado feito pelo jogador, o número de voltas será ainda suficientemente grande para poder-se com tranqüilidade desprezar este vício.

Agora, é indispensável que o tempo que se permaneça em cada opção, bem como aquele necessário para se passar de uma para a seguinte sejam rigorosamente igual para as 36 posições. A lógica e as instruções envolvidas para o percurso de 35 delas, sejam de 40A5 a 4083, são exatamente as mesmas.

Ilustrando isso melhor tem-se, para as 35 primeiras a seqüência

```
LDHL, BF7D – ADD HL, DE (o resultado é "não carry") – JRNC – DEC DE – Varredura
```

E para a última

```
LD HL, BF7D – ADD HL, DE (o resultado é "carry") – JRNC – LD DE, 40A5 – JR – Varredura
```

Em todo computador existe um "clock" (relógio), que nada mais é que um circuito oscilante, que vai emitindo impulsos elétricos ritmados, cada um dos quais é chamado de "clock cycle" (ciclo do relógio). Nos equipamentos comuns da linha Sinclair, este relógio tem 3,25 MHz, ou 3,25 milhões de ciclos por segundo. Cada instrução requer um certo número de ciclos para ser executada. Repetindo, não tempo, mas sim ciclos. Então um computador é tanto mais rápido quanto maior for o número de ciclos que seu relógio executar por segundo.

Existe na literatura a informação de quantos ciclos leva cada instrução. Pesquisando-a encontrou-se para a primeira seqüência.

INSTRUÇÃO	Nº DE CICLOS
LD HL, BF 7 D	10
ADD HL, DE	11
JR NC	7
DEC DE	6
TOTAL	34

e para a segunda

INSTRUÇÃO	Nº DE CICLOS
LD HL, BF 7 D	10
ADD HL, DE	11
JRNC	12
LD DE, 40 A5	10
JR	12
TOTAL	55

Ora, existe uma diferença de 21 ciclos, e este é um vício que precisa ser eliminado. Para isso, incluiu-se na primeira seqüência um Push BC seguido de um Pop BC que, sem alterar nada na lógica do sistema, ocupam 11 e 10 ciclos respectivamente, compensando os 21 da diferença.

Outro detalhe a observar é o número de ciclos associados as instruções condicionais — conforme o resultado da condição varia o número de ciclos necessários para a sua execução. É o que está ocorrendo no caso em questão, com a instrução JR NC — quando ocorre o "carry" não é necessário dar o salto ("Jump") e a instrução gasta 7 ciclos, e quando não ocorre o "carry" o salto deve ser feito, e o número de ciclos associado passa a 12.

```

16955 CD 5B 02 2C 20 FA 11 737
16956 AB 40 18 00 21 7D BF 611
16959 19 30 F5 1B 05 01 D5 948
16975 CD 5B 02 D1 2C 28 EF 925

```

Grupo 8 — 16983 a 17103 (4257 a 42CF)

```

4257 1A LD A, (DE)
4258 75 PUSH AF
4259 2A 0C 40 LD HL, (D-FILE)
425C 11 5B 00 LD DE, 91(D)
425F 19 ADD HL, DE
4260 0B AND OF
4262 4F LD C, A

```

```

4266 CD 76 42 CALL 4276
4267 F1 POP AF
4268 00 AND P0
4269 1F RRA
426A 1F RRA
426B 1F RRA
426C 1F RRA
426D 11 5C 00 LD DE, 50(D)
426E 19 ADD HL, DE
4271 CD 76 42 CALL 4276
4274 0B 00 JR NZ, 4283
4275 0B 04 LD B, 04
4276 01 CP 01
4277 00 07 JR NZ, 4283
4278 11 15 00 LD DE, 21(D)
4279 19 ADD HL, DE
427A 70 LD (HL), B
427B 19 ADD HL, DE
427C 00 RET
427D 0B 02 CP 02
427E 00 07 JR NZ, 428E
427F 70 LD (HL), B
4280 11 2A 00 LD DE, 42(D)
4281 19 ADD HL, DE
4282 70 LD (HL), B
4283 00 RET
4284 0B 03 CP 03
4285 00 00 JR NZ, 429B
4286 70 LD (HL), B
4287 11 15 00 LD DE, 21(D)
4288 19 ADD HL, DE
4289 70 LD (HL), B
428A 19 ADD HL, DE
428B 70 LD (HL), B
428C 00 RET
428D 0B 04 CP 04
428E 00 00 JR NZ, 42AC
428F 70 LD (HL), B
4290 03 INC HL
4291 03 INC HL
4292 70 LD (HL), B
4293 11 26 00 LD DE, 38(D)
4294 19 ADD HL, DE
4295 70 LD (HL), B
4296 03 INC HL
4297 03 INC HL
4298 70 LD (HL), B
4299 00 RET
429C 0B 05 CP 05
429D 00 07 JR NZ, 42BF
429E 70 LD (HL), B
429F 03 INC HL

```

```

400000 00 INC HL
400001 70 LD (HL),B
400002 11 10 00 LD DE,10(D)
400003 10 ADD HL,DE
400004 70 LD (HL),B
400005 10 ADD HL,DE
400006 70 LD (HL),B
400007 00 INC HL
400008 00 INC HL
400009 70 LD (HL),B
40000A 00 RET
40000B 70 LD (HL),B
40000C 00 INC HL
40000D 70 LD (HL),B
40000E 00 INC HL
40000F 70 LD (HL),B
400010 11 00 00 LD DE,00(D)
400011 10 ADD HL,DE
400012 70 LD (HL),B
400013 00 INC HL
400014 70 LD (HL),B
400015 00 INC HL
400016 70 LD (HL),B
400017 00 RET
400018 01 ADD A,C
400019 4F LD C,A

```

Apesar de extenso este grupo não apresenta dificuldades. Ele corresponde aos blocos 7 e 8 da figura 5.

De início carrega-se no acumulador o valor existente no endereço indicado por DE, proveniente do bloco anterior. Através do AND OF e AND FO – RRA – RRA – RRA – RRA identifica-se os números originalmente armazenados naquele endereço.

A sub-rotina chamada, do endereço 17014 (4276 em hexa) se encarrega de desenhar cada um dos dados em sua vez. A lógica usada é de fácil compreensão, lembrando apenas que a tela segue o conceito de máquina com capacidade menor que 3,5 Kbytes.

Finalmente, nas duas últimas instruções deste grupo, é obtido no acumulador a soma dos dados que é em seguida armazenado no registrador C.

```

100000 10 FF 00 00 40 11 58 497
100001 00 10 00 0F 4F CD 76 672
100002 40 F1 00 F0 1F 1F 1F 670
170004 1F 11 00 00 19 CD 76 455
170011 40 18 00 00 04 FE 01 443
170018 00 07 11 10 00 19 70 214
170025 10 00 FE 00 00 07 70 603

```

```

170000 11 00 00 10 70 00 FE 601
170001 00 00 00 70 11 10 00 104
170002 10 00 70 10 00 FE 04 700
170003 00 00 00 70 00 70 11 000
170004 00 00 10 70 00 70 00 007
170005 00 00 00 00 00 00 70 004
170006 00 00 11 00 10 70 00 000
170007 10 70 00 00 10 70 00 000
170008 00 70 00 11 00 00 040
170009 10 70 00 70 00 00 000
171000 01 4F 00 70 00 00 000

```

Grupo 9 – 17104 a 17135 (42D0 e 42EF em hexa)

```

42D0 2A 00 40 LD HL,(D-FILE)
42D1 11 22 00 LD DE,34(D)
42D2 10 20 ADD HL,DE
42D3 7E LD A,(HL)
42D4 07 AND A
42D5 06 00 LD B,00
42D6 08 03 JR Z,42E0
42D7 06 10 SUB 10(D)
42D8 47 LD B,A
42D9 03 INC HL
42DA 7E LD A,(HL)
42DB 07 AND A
42DC 08 10 JR Z,42FB
42DD 06 10 SUB 20(D)
42DE 00 ADD A,B
42DF 00 CP C
42E0 00 04 JR Z,431F
42E1 70 LD A,C
42E2 FE 07 CP 07
42E3 08 40 JR Z,433D

```

Os blocos da figura 5 associados a este grupo são os de números 9, 10 e 11. O que existe de interessante nele é o uso de informações escritas na tela como parte dos dados de programação. Isto é dito porque o valor do “ponto” está registrado no arquivo da tela e na forma decimal.

Para operar com este registro improvisado, o “ponto”, de início acha-se sua localização (LD LH, (DFile) – LD DE, 34 – ADD HL, DE) e verifica-se se há algo registrado (AND A – JRZ). Havendo, só pode ser o número 1, cujo código é 29 (1D em hexa) e que corresponde a 10. Daí a razão de subtrair 19 e acumular o resultado no registrador B, que havia sido previamente “zerado”. A mesma lógica repete-se no byte seguinte, só que lá poderá estar qual-

quer símbolo entre 0 e 9, cujos códigos subtraídos de 28 resultam em seu valor nominal. Por exemplo, o código do símbolo 2 é 30 (1E em hexa) do qual, se subtrairmos 28 (1C em hexa), resulta o valor 2.

Assim, tendo em A o resultado registrado no "ponto", fazer as comparações e dar os saltos requeridos pelos blocos 10 e 11 é bastante simples.

```

17104 2A 0C 40 11 20 00 10 104
17111 7E A7 05 00 20 00 08 556
17118 10 47 03 7E A7 20 15 460
17125 D6 1C 00 00 00 34 70 768
17132 FE 07 28 4D          378

```

Grupo 10 – 17136 a 17146 (42FO a 42FA em hexa)

```

42F0 21 00 60          LD HL,6000
42F3 08          DEC HL
42F4 7C          LDI A,H
42F5 00          OR L
42F8 00 FB          JR NZ,42F3
42F9 03 0F 42          JP 420F

```

Esta é uma rotina de controle do tempo em que os dados ficam desenhados na tela, e está associada ao bloco 12 da figura 5. A explicação sobre a lógica envolvida em rotinas deste tipo já foi apresentada no capítulo anterior.

```

17136 21 00 60 26 7C B5 20 600
17143 FB 03 0F 42          527

```

Grupo 11 – 17147 a 17182 (42FB a 431E)

```

42FB 70          LD A,C
42FC FE 07          CP 07
42FE 20 1F          JR Z,431F
4300 FE 05          CP 11(D)
4302 20 1B          JR Z,431F
4304 FE 02          CP 02
4306 20 35          JR Z,433D
4308 FE 03          CP 03
430A 20 31          JR Z,433D

```

```

430C FE 0C          CP 0C
430E 20 2D          JR Z,433D
4310 FE 0A          CP 10(D)
4312 30 0E          JR C,432E
4314 D6 0A          SUB 10(D)
4316 28          DEC HL
4317 36 1D          LD (HL),29(D)
4319 20          INC HL
431A C6 1C          ADD A,28(D)
431C 77          LD (HL),A
431D 18 D1          JR 42F0

```

Este grupo corresponde a situação em que o "ponto" está vazio, e se relaciona com os blocos 13, 14 e 15 do diagrama da figura 5.

De interessante cabe ressaltar apenas a forma como se registra o resultado dos dados no "ponto" da tela — é exatamente a lógica ao inverso daquela apresentada no grupo 9.

```

17147 70 FE 07 28 1F FE 0B 718
17154 26 1B FE 02 26 35 FE 670
17161 03 06 31 FE 0C 28 2D 443
17168 FE 0A 36 05 D6 0A 2B 593
17175 36 1D 20 C6 1C 77 18 487
17182 D1          209

```

Grupo 12 – 17183 a 17227 (431F a 434B em hexa)

```

431F 11 0C 00          LD DE,12(D)
4322 10          ADD HL,DE
4323 41 A6 40          LD DE,40A6
4326 EB          EX DE,HL
4327 01 06 00          LD BC,0006
432A ED B0          LDIR
432C EB          EX DE,HL
432D 11 D6 00          LD DE,216(D)
4330 10          ADD HL,DE
4331 11 C7 41          LD DE,4107
4334 EB          EX DE,HL
4335 01 0F 00          LD BC,15(D)
4338 ED B0          LDIR
433A C3 D7 41          JP 41D7
433D 11 0C 00          LD DE,12(D)

```

```

4340 19 ADD HL,DE
4341 11 F0 40 LD DE,40E9
4344 EB EX DE,HL
4345 01 06 00 LD BC,0006
4348 ED 00 LD IR
434F 18 F0 JR 432C

```

Esta é a última etapa. Corresponde aos blocos 16, 17 e 18 e se resume em compor a tela final usando rotinas e lógica já apresentadas no capítulo anterior.

Como comentário final cabe alertar que o jogador, ao apertar alguma tecla para sorteio de novos resultados dos dados, ou seja, ao executar o prescrito no bloco 6 do diagrama da figura 5, deve evitar o uso das teclas S ou N pois, se a opção conduzir ao final, o programa irá com muita rapidez para o bloco 2, onde ou os resultados são apagados da tela (tecla S) ou o programa é eliminado da memória (tecla N).

```

17183 11 0C 00 19 11 A6 40 001
17190 EB 01 06 00 ED 00 EB 000
17197 11 08 00 19 11 C7 41 000
17204 ED 01 07 00 ED 00 C3 000
17211 D7 41 11 0C 00 19 11 001
17218 E9 40 EB 01 06 00 ED 776
1722F 00 18 F0 424

```

RESUMO

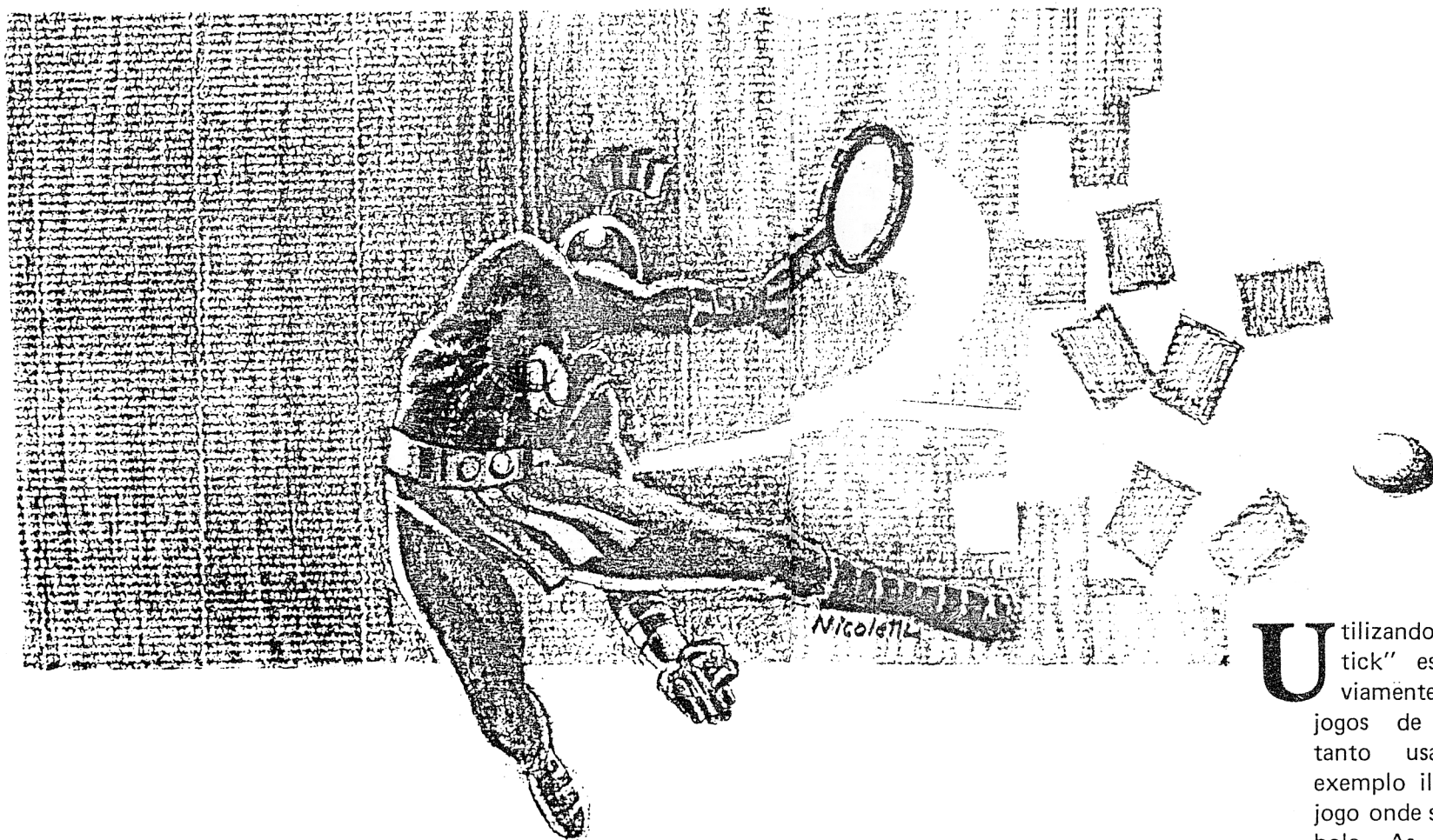
Foi abordado neste capítulo o seguinte

- * Uso de tela em configuração correspondente a capacidade menor que 3,5 Kbytes
- * Variável do Sistema — Ram Top ou RTP
- * Rotinas da Rom — RST ou JP0000, Impressão na tela (0808 em hexa ou 2056 em decimal), Pesquisa do Teclado KScan (02BB em hexa ou 699 em decimal)
- * Resultado em HL do uso de KScan e rotinas correlatas (espera para pressionar e liberar teclas)
- * Ciclos do relógio do computador
- * Forma de sorteio de um elemento numa seqüência de valores
- * Uso de valores registrados na tela na lógica do programa.

SUGESTÕES

- 1 — Converter o programa apresentado de forma a aplicá-lo em computadores com qualquer capacidade de RAM.
- 2 — Criar telas de instruções para o programa apresentado.
- 3 — Alterar o programa de forma que o resultado da rodada, Perdeu ou Ganhou, fique piscando na tela, ou seja, passando alternadamente o fundo branco com letras pretas para fundo preto e letras brancas.
- 4 — Alterar o programa de forma a “bloquear” o uso das teclas S e N quando o bloco 6 da figura 5 estiver em execução, de modo a evitar o problema citado no fim do capítulo.

O JOYSTICK



Utilizando o "Joystick" estaremos obviamente estudando jogos de ação. Para tanto usamos como exemplo ilustrativo um jogo onde se rebate uma bola. As particularidades do gerenciamento da raquete são discutidas detalhada e exaustivamente.

CAPÍTULO 3 – O “JOYSTICK”

Finalmente os jogos dinâmicos! E para melhor manejá-los nada mais indicado do que o “Joystick”. Nem todos os computadores da linha Sinclair, existentes no mercado, vêm equipados de fábrica com soquete próprio para receber este acessório, o que é uma pena.

Em essência o “Joystick” é uma extensão do teclado com cinco posições, quatro das quais acionadas por uma alavanca e uma por um botão, também chamado de botão de disparo. As quatro posições da alavanca geram sinais equivalentes aos das teclas 5, 6, 7 e 8 sem o “SHIFT”. A razão da escolha é óbvia, em função das setas associadas a estas teclas. Já o disparo é equivalente à tecla zero.

Como conseqüência é uma tradição nos programas da linha Sinclair, com movimentação, usar os valores 5 a 8 para gerar os deslocamentos, de forma a permitir o uso do Joystick, o que não impede, àquele que não o disponha, de jogar utilizando as próprias teclas do teclado.

O jogo ilustrativo que é apresentado a seguir, chamado “Rebatedor”, pode ser praticado com o “Joystick”. Diferentemente dos dois anteriores, tanto a lógica quanto o número de passos envolvidos são razoavelmente grandes, o que o torna de compreensão mais difícil. Apesar disto a apresentação será feita como naqueles casos, e o ideal é que o leitor vá acompanhando e entendendo cada passo na medida em que digita o programa. A absorção das informações será desta forma mais profunda.

Mas se houver muita dificuldade no entendimento do que o programa deve fazer, tem-se a alternativa de digitá-lo por inteiro, sem maiores atenções além dos cuidados básicos para evitar enganos, e rodá-lo. Com isso ter-se-á na mente a idéia exata de como o jogo opera. Em seguida deve-se voltar a ler o texto, etapa por etapa. Cabe alertar sobre dois perigos — a tendência de abandonar-se a tarefa após ver o resultado e a impressão, às vezes falsa, de ter-se entendido aquilo que de fato não se entendeu.

Para iniciar a apresentação do jogo tem-se na figura 7 o esboço da tela.

Pressionando o zero ou o disparo, surgirá num dos lados, junto a uma das laterais, uma bola que se moverá no sentido da base, numa das direções possíveis.

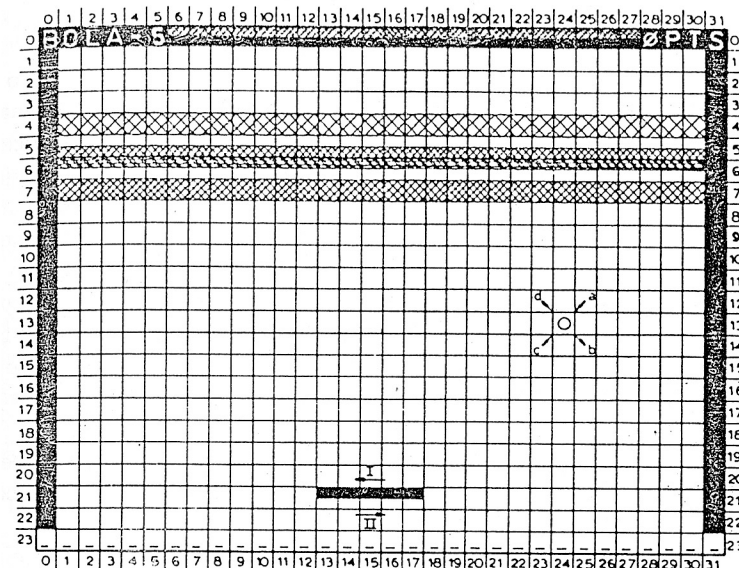


FIGURA 7

Movendo a alavanca do Joystick para à direita ou à esquerda, ou então pressionando uma das teclas 5 ou 8, a raquete move-se e deverá ser posicionada de forma a impedir que a bola atinja a base marcada com pequenos pontos. Existem três formas da bola ser rebatida pela raquete:

* Se atingir um dos cantos da raquete ela retornará na mesma direção em que vinha, porém em sentido oposto; usando a figura 7, se a bola movia-se segundo c, após tocar o canto da raquete ela passa para a.

** Atingindo a face da raquete, com esta movendo-se no mesmo sentido que a bola, ela “escorregará” (deslocar-se-á uma posição para o lado) antes de repicar; assim, na figura 7, se a bola vinha segundo b, atingindo a raquete que se movia na direção II, ela deslocar-se-á uma posição para à direita e, em seguida, assumirá o movimento na direção a.

*** A raquete estando parada, ou se movendo em sentido oposto ao do que vem a bola, esta repicará sem escorregar; usando outra vez a figura 7, se a bola vinha se movendo segundo c, atingindo a raquete parada ou se movendo na direção II, ela passa a seguir o movimento conforme d.

As laterais e o topo produzem um repique “normal” (sem escorregamento) na bola. Novamente, observando a figura 7, se a bola vier segundo d e

tocar a lateral esquerda ela passará para a, e se tocar o topo em seguida, assumirá a direção b.

A barreira é constituída por quatro camadas de 30 blocos cada: o de baixo com quadriculado de código 136 (ver o manual de seu equipamento), o imediatamente acima com o semi-quadriculado de código 10, o terceiro com o semi-quadriculado de código 9, e finalmente, o de cima com quadriculado de código 8. A bola batendo na barreira faz "desaparecer" o bloco atingido, aumenta de um o número de pontos conseguidos e sofre um repique "normal", da forma já explicada anteriormente. Além disso, em função do bloco atingido, a velocidade da bola (e da raquete por consequência) é alterada: se for um da camada de baixo, de código 136, a velocidade será a mais baixa, que é também aquela com que o jogo se inicia; se for da segunda, código 10, a velocidade sobe um pouco; como o 9 sobe mais ainda, e finalmente, com o bloco de código 8 a velocidade será máxima.

Se os 120 blocos forem "apagados" a barreira se recompõe, a bola desaparece, a raquete volta à posição de partida, e o jogo pode ser reiniciado com o botão de disparo.

Se o jogador não tiver habilidade suficiente para impedir que a bola atinja a base, perderá a mesma. A cada bola perdida a raquete é reposicionada e uma nova bola volta a surgir ao aperto do disparo. Obviamente a situação da barreira não é alterada nesta circunstância.

Após a perda das 5 bolas o jogo chega ao fim e aparece na tela a mensagem "Novo Jogo? (S/N)". Pressionando o S o jogo reinicia e com N a memória RAM é apagada.

O Basic correspondente ao jogo é o seguinte

```

00 REM ( 706 BYTES )
10 SAVE "REBAT"
20 RAND USA 16528

```

Isto indica que o programa é todo escrito em linguagem de máquina e ocupa menos que 2 Kbytes, como o CREPE do capítulo anterior. Porém o Rebatador é algo mais longo por utilizar a tela cheia, o que ocupa 792 bytes enquanto o Crepe só o faz parcialmente.

Antes de analisar o diagrama de blocos da figura 8 cabe apresentar os Registros Improvisados utilizados pelo programa;

NOME	ENDEREÇO (HEXA — DEC.)	ABREVIATURA
Número de bolas	DFile + 6	—
Sentido da Raquete	403C/D — 16444/5	SENT
Contagem	403E — 16446	CONT
Posição	4040/1 — 16448/9	POSBOL
Posição da Raquete	4042/3 — 16450/1	POSRAQ
Direção da Bola	4044/5 — 16452/3	DIRBOL
Velocidade	4046/7 — 16454/5	VEL

Para o Número de Bolas está sendo usada a própria tela, à semelhança do feito com o Ponto do Crepe. Os demais registros foram todos colocados no "Buffer" da impressora.

O Sentido da Raquete é um registro que indica se a raquete está sendo movida para a direita ou esquerda ou então parada, e é utilizado como indicador para gerar o efeito de escorregamento lateral da bola, no repique contra a raquete, como foi descrito anteriormente. A indicação é obtida copiando simplesmente neste registro o conteúdo do par HL logo após a chamada ao KScan.

O registro Contagem contém a cada momento o número de blocos ainda existentes na barreira. Isto é conseguido carregando-o inicialmente com o valor 120, e diminuindo uma unidade a cada bloco que é atingido. Sua finalidade é indicar o momento que a barreira foi totalmente apagada (CONT = 0), quando então o programa será interrompido para que a barreira seja refeita, e o jogo possa recomeçar. Poder-se-ia pensar em utilizar os Pontos acumulados na tela para atuar como este registrador. Porém lembrando que a barreira pode ser totalmente apagada várias vezes, a utilização de um registro especial para este controle é uma solução mais prática.

Os registros de Posição, seja da Bola como da Raquete, como o próprio nome diz, contém o endereço onde o elemento em questão se encontra. No caso da Raquete a referência é sua posição central — ela é representada por cinco símbolos gráficos de código 3.

O registro Direção da Bola tem uma correspondência direta com as quatro setas a, b, c e d representadas na figura 7. Isto é, nele está registrada a direção que a bola vem se deslocando. Isto é feito da seguinte forma: sabe-se que se a bola segue na direção a e encontra-se numa certa posição, o próximo endereço que ela ocupará será o da posição atual subtraído de 32; se a direção for b o próximo será o atual aumentado de 34 e assim por diante. Em resumo

Direção	Valor a Somar ao End atual	Conteúdo do DIRBOL
a	- 32	FFEO
b	34	0022
c	32	0020
d	- 34	FFDE

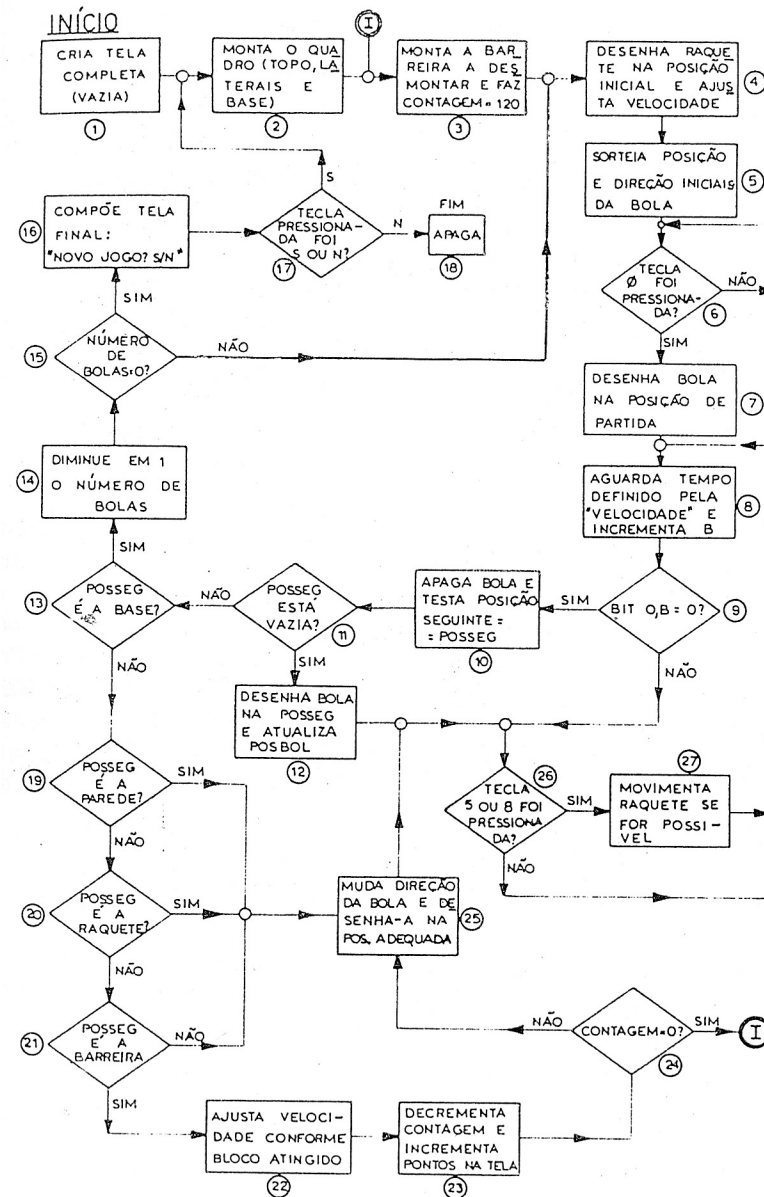
Como se vê o conteúdo de DIRBOL é, em hexadecimal, exatamente o valor a somar ao endereço atual de forma a encontrar o próximo que a bola ocupará, a menos que haja algo que o impeça, como a raquete, a lateral ou o topo, por exemplo.

Finalmente o registro Velocidade contém o valor correspondente ao último bloco da barreira que foi "apagado", e que deverá ser decrementado até atingir zero, em cada ciclo do movimento. Quanto menor for este valor, mais rapidamente os eventos se sucedem e mais dinâmico se torna o jogo. Os valores possíveis de VEL em hexadecimal são

Condição inicial do jogo	-	0500
Bloco apagado tem código 136	-	0500
Bloco apagado tem código 10	-	0400
Bloco apagado tem código 9	-	0300
Bloco apagado tem código 8	-	0150

Conhecendo os Registros Improvisados usados é mais fácil examinar e entender o diagrama de blocos da figura 8. Após tê-lo feito, é o momento de detalhar o programa correlacionando-o com os blocos apresentados, como já foi feito nos capítulos anteriores.

FIGURA 8



Grupo 1 – 16514 a 16527 (4082 a 408F em hexa)

Aí está o texto para compor a tela final: “NOVO JOGO? S/N”

```
16514 33 34 3B 34 00 2F 34 317
16521 2C 34 0F 00 3B 18 33 241
```

Grupo 2 – 16528 a 16541 (4090 a 409D em hexa)

```
4090 FD 36 22 00 LD (IY,+22),00
4094 21 00 03 LD HL,0300
4097 AF XOR A
4098 D7 RST 10
4099 2B DEC HL
409A 7C LD A,H
409B 8B OR L
409C 20 F8 JR NZ,4097
```

Este corresponde ao bloco 1 da figura 8, e consiste numa rotina padrão para gerar o espaço correspondente à tela completa, “escrevendo” na mesma espaços vazios.

A rotina se inicia com um LD (IY + 22), 00 (FD362200) – ora, pela lógica da ROM, ao colocar o Basic à disposição do usuário, o Registro IY é feito igual a 4000 em hexa. Então IY + 22 coincide com o endereço 4022, ou 16418 em decimal, que é o do DFSZ citado no 1º capítulo. Colocando zero neste endereço as 24 linhas da tela ficam disponíveis para o usuário, e pode-se “escrever” os espaços vazios em 24 x 32 = 768 posições ou 0300 em hexa, que é feito pelo restante da rotina.

```
16528 FD 36 22 00 21 00 03 377
16535 AF D7 2B 7C B5 20 F8 1018
```

Grupo 3 – 16542 a 16603 (409E a 40DB em hexa)

```
409E 2A 0C 40 LD HL,(D-FILE)
40A1 23 INC HL
40A2 35 A7 LD (HL),A7
40A4 23 INC HL
40A5 35 B4 LD (HL),B4
40A7 23 INC HL
40A8 35 B1 LD (HL),B1
40AA 23 INC HL
```

```
40AB 35 A5 LD (HL),A5
40AD 23 INC HL
40AF 35 95 LD (HL),95
40B0 23 INC HL
40B1 35 A1 LD (HL),A1
40B3 23 LD B,18
40B4 23 INC HL
40B8 35 8A LD (HL),8A
40BA 23 DJNZ 40B5
40BB 23 INC HL
40BD 35 9C LD (HL),9C
40BE 23 INC HL
40BF 35 85 LD (HL),85
40C0 23 INC HL
40C1 35 B9 LD (HL),B9
40C3 23 INC HL
40C4 35 B8 LD (HL),B8
40C5 23 INC HL
40C7 23 INC HL
40C8 11 1F 00 LD DE,001F
40CB 01 00 18 LD BC,1800
40CE 71 LD (HL),0
40CF 19 ADD HL,DE
40D0 71 LD (HL),0
40D1 23 INC HL
40D2 23 INC HL
40D3 19 F0 DJNZ 40CE
40D5 00 20 LD B,20
40D7 3B 1B LD (HL),1B
40D8 23 INC HL
40DA 18 FB DJNZ 40D7
```

A simplicidade do exposto dispensa comentários. É o próprio bloco 2 da figura 8.

```
16604 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16605 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16606 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16607 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16608 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16609 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1660A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1660B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1660C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1660D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1660E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1660F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16610 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16611 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16612 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16613 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16614 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16615 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16616 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16617 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16618 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16619 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16621 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16622 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16623 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16624 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16625 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16626 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16627 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16628 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16629 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16630 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16631 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16632 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16633 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16634 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16635 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16636 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16637 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16638 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16639 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16641 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16642 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16643 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16644 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16645 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16646 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16647 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16648 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16649 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16651 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16652 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16653 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16654 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16655 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Grupo 4 – 16604 a 16655 (40DC a 410F em hexa)

```
40DC 3E 78 LD A,120(D)
40DE 3D 3E 40 LD (CONT),A
40E1 3D 0C 40 LD HL,(D-FILE)
40E4 11 85 00 LD DE,133(D)
```

```

40F7 19      ADD HL,DE
40F8 01 08 1E LD BC,1E08
40F9 00 02 41 CALL 4102
40FA 01 09 1E LD BC,1E09
40FB 00 02 41 CALL 4102
40FC 01 0A 1E LD BC,1E0A
40FD 00 02 41 CALL 4102
40FE 01 0B 1E LD BC,1E0B
4100 00 02 41 CALL 4102
4101 18 41 1E JR 4143
4102 03 1E INC HL
4103 7E 1E LD R,(HL)
4104 7E 78 CP 78
4105 7E 7A JR NZ,4102
4106 7E 80 CP 80
4107 7E 78 JR NZ,4102
4108 28 78 LD (HL),C
4109 71 1E DJNZ 4102
410A 10 F3 RET
410B 09

```

Outro grupo bem simples. Corresponde ao bloco 3, ou seja, compõe a barreira e carrega 120 no Registro Improvisado "Contagem".

```

1666 04 0E 78 03 0E 40 0A 00 410
1667 11 40 00 00 41 01 00 00 00
1668 10 1E 00 00 41 01 0E 04 00
1669 00 00 00 00 41 01 00 00 00
166A 00 00 41 01 00 00 00 00
166B 00 41 18 41 00 00 78 00 77
166C 00 00 00 00 00 00 71 00 71
166D 00 10 70 00 00 00 40 00

```

Grupo 5 – 16656 a 16706 (4110 a 4142 em hexa)

```

4110 11 05 00 LD DE,0005
4111 0A 0C 40 LD HL,(D-FILE)
4112 10 00 ADD HL,DE
4113 7E 1E LD R,(HL)
4114 0D DEC R
4115 77 LD (HL),R
4116 7E 0C CP 0C
4117 00 00 JR NZ,4143
4118 00 00 LD DE,02B7
4119 11 07 02 ADD HL,DE
411A 11 02 40 LD DE,4082
411B 7E 1E EX DE,HL
411C 01 0E 00 LD BC,000E
411D 00 00 LDIR
411E 00 00 CALL KSCAN
411F 07 00 AND A
4120 11 7F EF LD DE,EF7F

```

```

4132 ED 52 SBC HL,DE
4133 20 01 JR NZ,4137
4134 07 00 RST 00
4135 10 00 ADD HL,DE
4136 07 00 AND A
4137 11 7D FB LD DE,FBFD
4138 ED 00 SBC HL,DE
4139 20 00 JR NZ,412B
4140 03 0E 40 JP 408E

```

Por que simplificar se é possível complicar? Por incrível que pareça este grupo corresponde aos blocos 14, 15, 16, 17 e 18 do diagrama da figura 8. A idéia foi evitar o "salto endereçado" do final da rotina, C39E40, usando em seu lugar um "salto relativo" JR, o que acabou não sendo possível pelo grande número de bytes a pular.

Isto posto, pouco resta a comentar. De início localiza-se o registro "No de Bolas" que é DFile + 6, reduz-se de um seu conteúdo e compara-se o resultado com 9C que é o zero em fundo reverso ou preto. Não sendo igual vai-se para o início do jogo, isto é, para o bloco 4 que vem a seguir. Em caso contrário, aplica-se a mesma rotina de finalização já vista anteriormente.

```

16706 11 05 00 00 40 10 10 00
16707 7E 0D 77 7D 00 00 00 70 00
16708 11 07 00 10 11 02 40 40 00
16709 7E 01 00 00 ED 00 00 00 00
1670A 7E 02 07 11 7F 7F ED 00 70
1670B 00 00 01 07 10 07 11 00 03
1670C 7D 7D ED 02 00 00 03 10 00
1670D 0E 40 00 00 00 00 00 00

```

Grupo 6 – 16707 a 16746 (4143 a 416A em hexa)

```

4143 0A 0C 40 LD HL,(D-FILE)
4144 11 07 02 LD DE,0007(D)
4145 00 00 ADD HL,DE
4146 00 1E LD B,00(D)
4147 00 00 LD (HL),00
4148 00 00 INC HL
4149 10 FB DJNZ 414C
414A 11 12 00 LD DE,12(D)
414B 07 00 AND A
414C 00 00 SBC HL,DE
414D 00 00 LD R,00
414E 77 00 LD (HL),R

```



```

418A 03      INC HL
418B 77      LD (HL),A
418C 03      INC HL
418D 77      LD (HL),A
418E 00 40 40 LD (POSRAQ),HL
418F 00      INC HL
4190 77      LD (HL),A
4191 03      INC HL
4192 77      LD (HL),A
4193 03      INC HL
4194 77      LD (HL),A
4195 01 00 05 LD HL,0500
4196 02 45 40 LD (VEL),HL

```

Como já foi dito, este é o próprio bloco 4. No grupo começa-se apagando totalmente o conteúdo da linha 21, com exceção, é claro, dos blocos das laterais. Assim qualquer vestígio da posição anterior da raquete, bem como da frase "Novo Jogo? S/N" é eliminado.

Desenha-se então a raquete, carrega-se o endereço de sua posição central no Registro POSRAQ — 224240 — e finalmente carrega-se 0500 no VEL.

```

16707 2A 00 40 11 B7 02 19 345
16714 05 1E 38 00 23 10 7B 382
16721 11 12 00 A7 ED 52 3F 583
16728 03 77 23 77 23 77 23 484
16735 42 40 23 77 23 77 21 471
16742 00 05 22 45 40      173

```

Grupo 7 — 16747 a 16799 (416B a 419F em hexa)

```

416B 3A 34 40      LD A,(QUADROS)
416E 06 05      SUB 05
4170 30 FD      JR NC,416E
4172 06 0E      ADD A,14(D)
4174 47      LD B,A
4175 11 21 00      LD DE,33(D)
4178 0B 40      BIT 0,B
417A 20 11      JR NZ,418D
417C 21 22 00      LD HL,0022
417F 22 44 40      LD (DIRBOL),HL
4182 2A 0C 40      LD HL,(D-FILE)
4185 19      ADD HL,DE
4186 10 FD      DJNZ 4185
4188 23      INC HL
4189 22 40 40      LD (POSBOL),HL
418C 10 12      JR 41A0
418E 21 20 00      LD HL,0020
4191 22 44 40      LD (DIRBOL),HL

```

```

4194 2A 0C 40      LD HL,(D-FILE)
4197 10      ADD HL,DE
4198 10 FD      DJNZ 4197
419A 19      ADD HL,DE
419B 2B      DEC HL
419C 2B      DEC HL
419D 22 40 40      LD (POSBOL),HL

```

Aqui tem-se novidades — o sorteio da posição inicial da bola, usando a variável de Sistema Quadros ou "Frames" em inglês, de acordo com o prescrito no bloco 5 da figura 8.

A Variável Quadros situada em 16436/7 (4034/5 em hexa) "conta" o número de quadros ou impulsos de imagens enviadas à tela. É portanto, num instante arbitrário, uma boa fonte para buscar um número ao acaso.

Tomou-se o byte menos significativo desta variável que é o primeiro, no endereço 4034, consistindo num número de 0 a 255.

Subtraíndo repetidamente 6 deste valor até ocorrer um "carry", chega-se a um número entre — 1 a — 6, que somado a 14 resulta finalmente num número aleatório entre 8 e 13.

Se o número assim conseguido for 8, 10 ou 12 o programa fará com que a bola surja nas linhas 9, 11 ou 13 respectivamente logo após a tecla Ø ter sido pressionada, tendo a direção b da figura 7 — em DIRBOL haverá 0022. Se o número for 9, 11 ou 13 a bola surgirá em 10, 12 ou 14 com direção c, ou seja, em DIRBOL haverá 0020.

```

16747 3A 34 40 06 05 30 FD 894
16754 05 0E 47 11 21 00 0B 538
16761 40 20 11 21 22 00 22 214
16768 44 40 2A 0C 40 19 10 291
16775 FD 23 22 40 40 18 12 402
16782 21 20 00 22 44 40 2A 273
16789 0C 40 10 10 FD 19 2B 438
16796 2B 22 40 40      205

```

Grupo 8 — 16800 a 16823 (41A0 a 41B7 em hexa)

```

41A0 0D 5B 02      CALL KSCAN
41A3 A7      AND A
41A4 11 EF FD      LD DE,FDEF
41A7 ED 52      SBC HL,DE
41A9 20 F5      JR NZ,41A0
41AB 2A 40 40      LD HL,(POSBOL)
41AE ED 5B 44 40 LD DE,(DIRBOL)

```

```

41B2 10      ADD HL,DE
41B3 36 34   LD (HL),34
41B5 22 40 40 LD (POSBOL),HL

```

Aí está o KScan de novo. Só que verificando se a tecla 0 foi pressionada, conforme prescrito no bloco 6, em caso positivo desenhando a bola na posição informada anteriormente de acordo com o bloco 7.

```

16800 CD 88 02 A7 11 EF FD 1070
16807 ED 52 20 75 23 40 40 785
16814 ED 58 44 40 10 36 34 5001
16821 ED 40 40      182

```

Grupo 9 — 16824 a 16837 (41B8 a 41C5 em hexa)

```

41B8 2A 48 40   LD HL,(VEL)
41B9 2B      DEC HL
41BC 7C      LD A,H
41BD 80      OR L
41BE 20 FB   JR NZ,41BB
41C0 04      INC B
41C1 CB 40   BIT 0,B
41C3 02 FD 42  JP NZ,42FD

```

Este grupo corresponde aos blocos 8 e 9 da figura 8. Quanto ao primeiro, nada importante a comentar. Já no bloco 9 cabe observar que, através do incremento de B e o teste do bit 0 ser 0 ou 1, se consegue que a raquete tenha o dobro da velocidade da bola. Isto porque se Bit 0, B = 0 após passar por algum dos blocos alternativos chega-se ao 26, que permite mover a raquete. Num instante imediatamente posterior, B é incrementado, Bit 0, B será 1 e o caminho é direto para o 26 novamente. Isto significa dois impulsos para o bloco 26 para cada um que vai mover a bola.

Isto é indispensável para permitir ao jogador acompanhar a máquina. Mesmo com esta "desvantagem", o desempenho da máquina é bastante difícil de ser seguido pelo homem.

```

16824 2A 48 40 2B 7C 85 20 555
16831 FB 04 CB 40 C2 7D 42 1035

```

Grupo 10 — 16838 a 16904 (41C6 a 4208 em hexa)

```

41C6 2A 40 40   LD HL,(POSBOL)
41C7 2B 00 00   LD (HL),00
41C8 ED 55 44 40 LD DE,(DIRBOL)
41C9 2A 40 40   ADD HL,DE
41D0 2A 40 40   LD A,(HL)
41D1 2B      AND A
41D2 2A 40 40   JR NZ,41DC
41D3 2A 40 40   LD (HL),34
41D4 2A 40 40   LD (POSBOL),HL
41D5 2A 40 40   JP 42FD
41D6 2A 40 40   CP 10
41D7 2A 40 40   JP NZ,4110
41D8 2A 40 40   CP 07
41D9 2A 40 40   JP NZ,4270
41DA 2A 40 40   CP 00
41DB 2A 40 40   JP NZ,4230
41DC 2A 40 40   CP 04
41DD 2A 40 40   JP NZ,4235
41DE 2A 40 40   CP 0D
41DF 2A 40 40   JP NZ,423A
41E0 2A 40 40   CP 00
41E1 2A 40 40   JP NZ,423F
41E2 2A 40 40   CP 00
41E3 2A 40 40   JP NZ,4285
41E4 2A 40 40   DEC HL
41E5 2A 40 40   LD A,(HL)
41E6 2A 40 40   CP 00
41E7 2A 40 40   JP NZ,4286
41E8 2A 40 40   INC HL
41E9 2A 40 40   INC HL
41EA 2A 40 40   LD A,(HL)
41EB 2A 40 40   CP 00
41EC 2A 40 40   JR NZ,4286

```

Este consiste num grupo de instruções bastante interessante. Ele está envolvido com vários blocos da figura 8 e ainda faz uso do próprio estado da tela para decisão dos passos a seguir.

De início ele carrega em HL o que está em POSBOL e coloca zero neste endereço, o que equivale "apagar" a bola. Somando a HL o que está em DIRBOL chega-se ao endereço que a bola deve atingir na seqüência, ou seja, o que foi chamado de "posseg" na figura 8. Através da série de instruções LDA, (HL) — AND A — JRNZ, testa-se se a possej está vazia, usando para isso a própria tela. Cobriu-se assim os blocos 10 e 11 da figura citada.

Com a série LD (HL), 34 — LD (POSBOL), HL — JP completa-se a sequência com o bloco 12 e atinge-se a 26.

Em seguida é feita a verificação se a possege contém 1B, 80, 08, 09, enfim qualquer um dos símbolos gráficos existentes ou possíveis de estarem presentes abaixo da linha zero, em caso positivo, é feito um salto até a rotina correspondente ao conteúdo. Note-se que no teste da raquete (figura de código 03) é necessário não só verificar a possege como também as vizinhas a esta posição — DEC HL; LD A, (HL); CP03; JPZ e em seguida INC HL; INC HL; LD A, (HL); CPO2; JRZ. Passou-se assim pelos blocos 13, 19, 20 e 21 da figura 8.

```

16 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
17 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
18 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
19 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
21 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
22 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
23 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
24 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
25 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
26 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
27 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
28 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
29 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
31 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
32 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
33 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
34 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
35 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
36 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
37 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
38 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
39 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00000000 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Se a possege não está vazia e ao mesmo tempo, não contém nenhum dos elementos abaixo da linha zero, resulta que a bola vai de encontro ao topo.

Contrariamente às situações anteriores, tem-se aqui o caso no qual vários grupos de instruções correspondem a um só bloco: o 25. O grupo 11 em questão é um deles. Nele tem-se a lógica de mudança de direção da bola quando encontra um “obstáculo horizontal”, isto é, se ela vier na direção a esta é convertida na b, se vier segundo c passa a d e assim por diante. É obviamente isto que deve ser feito se a possege é um elemento do topo.

Este mesmo grupo será “chamado” em várias circunstâncias verificadas adiante.

```

16 000000 03A 44 40 A7 17 38 0E 450
17 000100 7FD 40 20 05 21 00 7F 070
18 000100 7FD 40 20 05 21 00 7F 070
19 000100 7FD 40 20 05 21 00 7F 070
20 000100 7FD 40 20 05 21 00 7F 070
21 000100 7FD 40 20 05 21 00 7F 070
22 000100 7FD 40 20 05 21 00 7F 070
23 000100 7FD 40 20 05 21 00 7F 070
24 000100 7FD 40 20 05 21 00 7F 070
25 000100 7FD 40 20 05 21 00 7F 070
26 000100 7FD 40 20 05 21 00 7F 070
27 000100 7FD 40 20 05 21 00 7F 070
28 000100 7FD 40 20 05 21 00 7F 070
29 000100 7FD 40 20 05 21 00 7F 070
30 000100 7FD 40 20 05 21 00 7F 070

```

Grupo 11 – 16905 a 16943 (4209 a 422F em hexa)

```

420900 03D 44 40 LD A, (DIRBOL)
420A00 D17 JRN A
420B00 D17 JRN A
420C00 030 0FF JRA C,421E
420D00 700 400 CP 40
420E00 011 000 JR NZ,4219
420F00 011 000 JR NZ,4219
421000 011 000 JR NZ,4219
421100 011 000 JR NZ,4219
421200 011 000 JR NZ,4219
421300 011 000 JR NZ,4219
421400 011 000 JR NZ,4219
421500 011 000 JR NZ,4219
421600 011 000 JR NZ,4219
421700 011 000 JR NZ,4219
421800 011 000 JR NZ,4219
421900 011 000 JR NZ,4219
421A00 011 000 JR NZ,4219
421B00 011 000 JR NZ,4219
421C00 011 000 JR NZ,4219
421D00 011 000 JR NZ,4219
421E00 011 000 JR NZ,4219
421F00 011 000 JR NZ,4219
422000 011 000 JR NZ,4219
422100 011 000 JR NZ,4219
422200 011 000 JR NZ,4219
422300 011 000 JR NZ,4219
422400 011 000 JR NZ,4219
422500 011 000 JR NZ,4219
422600 011 000 JR NZ,4219
422700 011 000 JR NZ,4219
422800 011 000 JR NZ,4219
422900 011 000 JR NZ,4219
422A00 011 000 JR NZ,4219
422B00 011 000 JR NZ,4219
422C00 011 000 JR NZ,4219
422D00 011 000 JR NZ,4219
422E00 011 000 JR NZ,4219
422F00 011 000 JR NZ,4219

```

Grupo 12 – 16944 a 17007 (4230 a 426F em hexa)

```

423000 11 50 01 LD DE,0150
423100 18 00 00 JR 4242
423200 11 00 03 LD DE,0300
423300 18 00 00 JR 4242
423400 11 00 04 LD DE,0400
423500 18 00 00 JR 4242
423600 11 00 05 LD DE,0500
423700 18 00 40 LD (VEL),DE
423800 0E 03 40 LD (HL),34
423900 02 40 40 LD (POSBOL),HL
423A00 0A 00 40 LD HL,(0-FILM)
423B00 11 10 00 LD DE,20(D)
423C00 700 ADD HL,DE
423D00 700 LD A,(HL)
423E00 700 CP 0A
423F00 0E 02 02 JR NZ,4259
424000 0E 00 00 LD A,9C
424100 00 00 00 INC A
424200 700 AB CP A6
424300 00 00 00 JR NZ,4253
424400 00 00 00 LD (HL),9C

```

```

444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000
444444 000000 000000 000000 000000 000000

```

Se a bola encontrou a barreira é para este grupo que se dá o salto. A correspondência é com os blocos 22, 23 e 24.

O ajuste da velocidade (bloco 22) é feito através de LD DE, 0150 ou um dos outros três valores seguido por LD (VEL), DE. O acerto de contagem vem a seguir e consiste numa lógica interessante, que deve ser bem analisada, pelo potencial de uso que tem em muitos jogos.

Para concluir o grupo, é feita a redução na contagem, com a seqüência LD (HL), A; LD A, (CONT); DEC A; JPZ; LD (CONT), A.

```

100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000
100000 000000 000000 000000 000000 000000

```

Grupo 13 — 17008 a 17029 (4270 a 4285 em hexa)

```

40070 00 44 40 LD A, (DIRBOL)
40073 00 44 40 AND A
40074 00 44 40 RLA
40075 00 04 40 JR C, 427B
40077 00 40 40 LD A, 42
40078 00 00 40 JR 427D
40079 00 00 40 LD A, BE
4007D 44 40 LD HL, (DIRBOL)
40080 00 44 40 SUB L
40081 00 44 40 LD (DIRBOL), A
40084 10 77 40 JR 42FD

```

Eis mais uma parcela do bloco 25. É a lógica a ser percorrida no caso da posse ser a parede.

```

170000 00 44 40 07 17 08 04 440
170010 00 44 40 08 17 08 04 440
170020 44 40 08 17 40 08 04 440
170030 77 40 00 00 44 40 08 110

```

Grupo 14 — 17030 a 17148 (4286 a 42FC em hexa)

```

40000 00 44 40 LD A, (DIRBOL)
40003 00 00 40 CP B
40006 00 00 40 JR NZ, 42C6
40009 00 40 40 LD HL, (POSBOL)
40012 11 01 00 LD DE, 33 (D)
40015 11 00 00 LD HL, DE
40018 00 00 40 LD A, (HL)
40021 00 00 40 CP B
40024 00 00 40 JR NZ, 42C2
40027 11 00 00 LD HL, FFDE
40030 00 44 40 LD (DIRBOL), DE
40033 10 00 40 JR 42FD
40036 11 00 44 40 LD DE, FFDE
40039 00 44 40 LD (DIRBOL), DE
40042 00 40 40 LD HL, (SENT)
40045 00 70 40 LD A, L
40048 00 04 40 AND H
40051 00 07 40 CP D7
40054 00 00 40 JR NZ, 42B4
40057 10 40 40 JR 42FD
40060 00 40 40 LD HL, (POSBOL)
40063 00 00 40 DEC HL
40066 00 00 40 LD A, (HL)
40069 00 00 40 CP B
40072 00 00 40 JR NZ, 42BF
40075 00 40 40 LD (POSBOL), HL
40078 00 04 40 LD (HL), 34
40081 10 07 40 JR 42FD
40084 11 40 40 LD HL, (POSBOL)
40087 11 01 00 LD DE, 0021
40090 10 70 40 ADD HL, DE
40093 00 00 40 LD A, (HL)
40096 00 00 40 CP B
40099 00 00 40 JR NZ, 42DB
40102 11 00 40 LD DE, FFDE

```

```

42D05 ED 50 14 40 LD (DIRBOL),DE
42D09 10 2E JR 42FD
42D0B 11 E0 5F LD DE,FFE0
42D0E ED 50 44 40 LD (DIRBOL),DE
42D10 0A 30 40 LD HL,(SENT)
42D15 7D LD A,L
42D18 04 AND H
42D1A 7E CP E7
42D1C 00 00 JR NZ,42ED
42D1E 10 10 JR 42FD
42D20 0A 40 40 LD HL,(POSBOL)
42D22 03 INC HL
42D24 7E CP A,(HL)
42D26 00 00 CP 00
42D28 00 00 JR NZ,42FB
42D2A 10 00 JR 42FD
42D2C 00 40 LD (POSBOL),HL
42D2E 00 34 LD (HL),34

```

Este longo grupo de instruções é o último relacionado com o bloco 25. Trata-se da lógica no caso da posse ser a raquete. Ela é longa em face das alternativas a considerar:

* Se a bola vai atingir o canto da raquete, ela deve retornar de onde veio, ou seja, se a direção era c passa a ser a.

* Se a bola for atingir a face da raquete é necessário verificar a hipótese de ocorrer ou não um "escorregamento" lateral. Porém isso só deve ser feito se a posição ao lado não for a parede.

```

17030 00 44 40 7E 00 00 00 00 00 00
17037 00 40 40 11 00 10 00 00 40 00
17044 7E 7E 00 00 00 11 00 00 70 00
17051 7E ED 00 44 40 10 00 00 00 00
17058 11 0E 00 00 44 40 00 00 40 00
17065 00 00 00 00 7D 04 7E 07 00 00 40
17072 00 00 00 40 00 40 40 00 00 00 40
17079 00 00 7E 7E 00 00 00 10 00 00 00
17086 00 00 00 40 40 00 04 10 00 00 40
17093 07 00 40 40 11 01 00 00 00 00 40
17100 10 7E 7E 00 00 00 11 47 47 40
17107 DE 7E ED 00 44 40 10 00 00 00
17114 00 11 00 7E ED 00 44 10 00 00 40
17121 40 00 00 40 7D 04 7E 77 00 00 40
17128 7 00 00 10 10 00 40 40 00 40 00
17135 40 00 7E 7E 00 00 00 00 00 40 00
17142 10 00 00 00 40 40 00 04 00 00 00

```

Grupo 15 – 17149 a 17219 (42FD a 4343 em hexa)

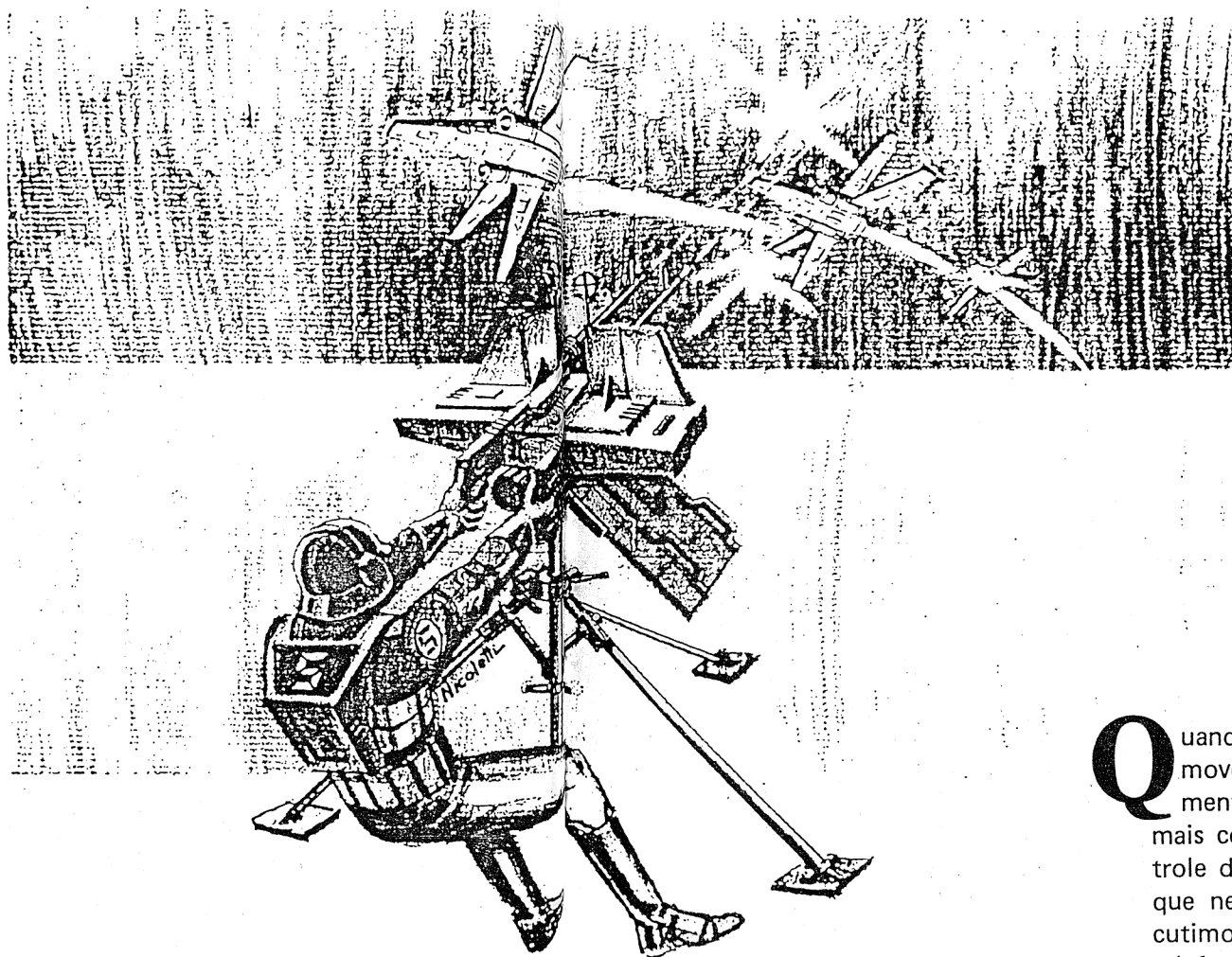
```

42FD 05 PUSH BC
42FE CD BB 02 CALL KSCAN
4301 01 POP BC
4302 02 30 40 LD (SENT),HL
4305 7D LD A,L
4306 7E CP A,L
4307 7E CP A,L
4308 00 0F AND 0F
4309 00 17 JR Z,4323
430A 00 40 40 LD HL,(POSR0)
430F 00 00 LD HL
4310 00 00 DEC HL
4311 00 00 DEC HL
4312 7E CP A,(HL)
4313 00 00 CP 00
4314 00 00 JR Z,4340
4315 00 00 LD (HL),00
4316 00 00 INC HL
4317 00 00 INC HL
4318 00 42 40 LD (POSR0),HL
4319 00 00 INC HL
431A 00 00 INC HL
431B 00 00 INC HL
431C 00 00 INC HL
431D 00 00 LD (HL),00
431E 7E CP AF
431F 00 10 AND 10
4320 00 40 40 LD HL,(POSR0)
4321 00 00 INC HL
4322 00 00 INC HL
4323 00 00 INC HL
4324 7E CP A,(HL)
4325 00 00 CP 00
4326 00 00 JR Z,4341
4327 00 00 LD (HL),00
4328 00 00 DEC HL
4329 00 00 DEC HL
432A 00 42 40 LD (POSR0),HL
432B 00 00 DEC HL
432C 00 00 DEC HL
432D 00 00 DEC HL
432E 00 00 LD (HL),00
432F 7E CP HL
4330 00 00 CP 00
4331 00 00 JR Z,4341
4332 00 00 LD (HL),00
4333 00 00 DEC HL
4334 00 00 DEC HL
4335 00 42 40 LD (POSR0),HL
4336 00 00 DEC HL
4337 00 00 DEC HL
4338 00 00 DEC HL
4339 00 00 LD (HL),00
433A 00 00 PUSH HL
433B 00 00 POP HL
433C 03 BB 41 JP 41B8

```

Finalmente completamos o programa. Este grupo está associado aos blocos 26 e 27, e inicia-se com um PUSH BC, antes do já conhecido CALL KScan, a fim de preservar o valor de B que é usado no bloco 9.

MOVIMENTOS MÚLTIPLOS



Quando várias coisas se movem simultaneamente na tela fica mais complicado o controle das trajetórias. Eis que neste capítulo discutimos como realizar tal façanha ao criar um jogo onde coexistem bombas, canhões, aviões e tiros.

CAPÍTULO 4 — MOVIMENTOS MÚLTIPLOS

Os conceitos básicos envolvidos na concepção e elaboração de jogos no computador foram cobertos nos capítulos anteriores. Agora trata-se de ganhar prática e enfatizar aspectos particulares que podem estar contidos num programa. No exemplo a seguir ver-se-á vários objetos movimentando-se na tela simultaneamente. Daí o nome do capítulo.

Trata-se de um jogo que, a semelhança de muitos outros existentes no mercado, chama-se Invasores. A fim de melhor apresentá-lo cabe examinar o esboço da tela que constitui a figura 9.

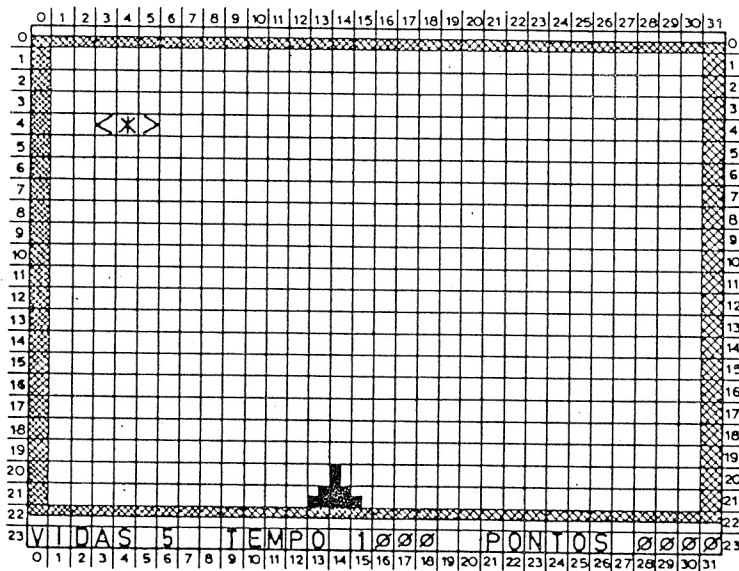


FIGURA 9

O invasor é um avião, representado pelo asterisco e os sinais de “maior” e “menor”, que “nasce” na tela a partir de uma das laterais, e desloca-se em direção à outra. A cada instante não há mais que um avião na tela.

Cabe ao canhão destruir o avião disparando tiros verticais. O canhão move-se para à direita e à esquerda, através das teclas 5 e 8 ou do “Joystick”, e dispara com a tecla 0. Cada vez que o canhão destrói o avião há um ganho

de 10 pontos, porém cada tiro disparado reduz a contagem em um ponto.

O avião também não é inofensivo — ele atira, segundo a diagonal, contra o canhão. Atingindo-o é perdida uma vida. O avião não dispara mais de um tiro por vez segundo cada diagonal. Numa mesma direção, o segundo tiro é disparado somente quando o primeiro já “sumiu”. Um detalhe: os tiros, seja do avião, como do canhão, deslocam-se com o dobro da velocidade dos outros elementos da tela.

Além disso existe uma contagem do tempo remanescente. Ou seja, o jogo termina quando ou o tempo ou as vidas chegam a zero. A meta é fazer o maior número de pontos neste intervalo de tempo.

O Basic associado ao programa é o seguinte.

```

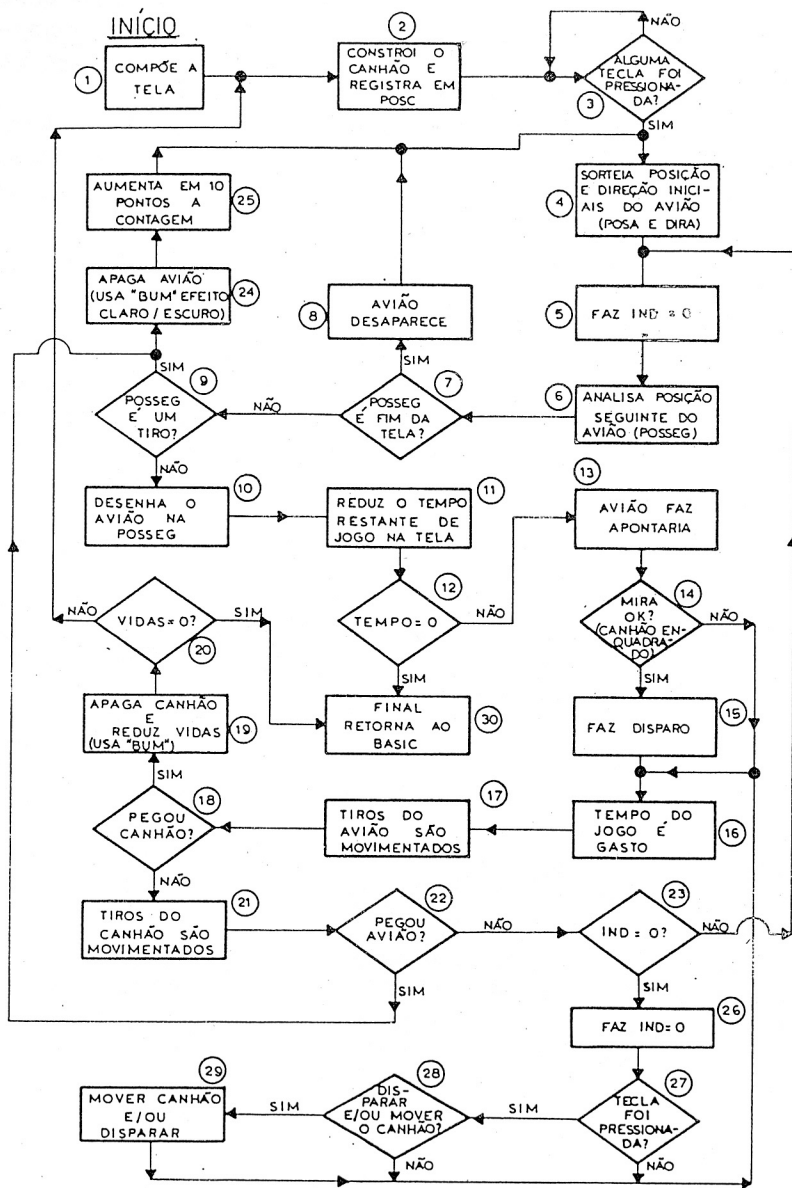
REM          949 BYTES )
CLS
20 RAND JSR 16546
30 IF INKEY="" THEN GOTO VAL
30
40 RUN
    
```

Como se vê o programa é bastante extenso, ocupando quase totalmente os 2 Kbytes disponíveis; daí a necessidade de economia da memória.

Antes de examinar o diagrama de blocos apresentado na figura 10, cabe definir os Registros Improvisados envolvidos:

NOME	ENDEREÇO (HEXA — DEC)	ABREVIATURA
VIDAS	DFile + 766	
TEMPO	DFile + 778	
PONTOS	DFile + 791	
Posição Avião	403C/D — 16444/5	POSA
Posição Tiro 1 Avião	403E/F — 16446/7	POSTA1
Posição Tiro 2 Avião	4040/1 — 16448/9	POSTA2
Posição Tiros Canhão	4042a — 16450a	POSTC
(10 tiros)	4055 — 16469	
Direção do Avião	4021 — 16417	DIRA
Indicador de Saída	4056 — 16470	IND
Posição Canhão	407B/C — 16507/8	POSC

FIGURA 10



Os três primeiros são os próprios campos correspondentes da tela. Os indicadores de posição, quer do Avião, do Canhão e dos respectivos tiros, informam a cada instante onde os mesmos se encontram. No caso do Avião é o asterisco que serve de posicionamento, para o Canhão é a figura de código 128 (80 em hexa), uma das que compõe a base do mesmo.

O Registro Direção do Avião – DIRA pode assumir apenas dois valores, zero ou um, indicando movimento do Avião indo da direita para esquerda ou vice-versa, respectivamente. O indicador de Saída está fazendo as vezes do registro B do programa Rebatedor, apresentado no capítulo anterior, ou seja a de fazer com que os tiros se movam duas vezes mais rapidamente que o avião e o canhão, e assume também somente dois valores – zero ou um.

Agora deve-se examinar e entender o diagrama de blocos. Um dos pontos a realçar é justamente a forma como funciona o IND. Na seqüência dos blocos, chegando ao 5, IND é feito igual a zero. Ao atingir o 25, passando portanto por 17 e 22, sendo zero, a seqüência é o 26 onde IND torna-se um e é aberta a possibilidade de mover-se o canhão. O retorno é para o bloco 16, passando novamente pelos blocos 17 e 22 onde os tiros são movimentados novamente, e daí, em 25 é fechado o ciclo, pois como $IND = 1$ a seqüência será o retorno ao 5.

Outro ponto a destacar é o efeito claro/escuro, também chamado “BUM”, utilizado nos blocos 19 e 23. Trata-se de uma sub-rotina que é “chamada” nos dois blocos citados, e que faz a inversão da tela toda, de fundo claro em escuro e vice-versa, por três vezes.

A próxima etapa é analisar os grupos com as instruções em linguagem de máquina.

Grupo 1 – 16514 a 16545 (4082 a 40A1 em hexa)

Aqui estão somente os dados para compor a tela inicial, o “rodapé” da figura 9.

16514	00	0F	00	00	00	00	01	070
16515	00	00	00	0D	00	00	04	001
16516	00	10	10	10	10	00	00	111
16517	00	00	04	00	00	04	00	001
16518	10	10	10	10				110

Grupo 2 – 16546 a 16560 (40A2 a 40B0 em hexa)

```

40A2 FD 36 22 00 LD. (IY+22),00
40A5 21 00 03 LD HL,0300
40A8 AF XOR A
40AB D7 RST 10
40AE 00 DEC HL
40B0 70 LD A,H
40B1 00 OR L
40B2 00 F9 JR NZ,40A9
40B3 00 NOP

```

Eis a mesma rotina de “encher” a tela para o caso de micros com RAM menor que 3,5 Kbytes.

```

16546 FD 36 22 00 21 00 03 377
16553 AF D7 2B 7C B5 20 F9 1019
16560 00 0

```

Grupo 3 – 16561 a 16603 (40B1 a 40DB em hexa)

```

40B1 2A 00 40 LD HL,(D-FILE)
40B4 06 20 LD B,32(D)
40B6 23 INC HL
40B7 38 09 LD (HL),09
40B9 10 FB DJNZ 40B5
40BB 06 15 LD B,21(D)
40BD 11 17 00 LD DE,31(D)
40C0 23 INC HL
40C1 23 INC HL
40C2 38 08 LD (HL),08
40C4 10 DE ADD HL,DE
40C5 38 08 LD (HL),08
40C7 10 F7 DJNZ 40C0
40C9 23 INC HL
40CA 06 20 LD B,32(D)
40CC 23 INC HL
40CD 38 0A LD (HL),0A
40CF 10 FB DJNZ 40CC
40D1 23 INC HL
40D2 23 INC HL
40D3 EB EX DE,HL
40D4 01 20 00 LD BC,32(D)
40D7 21 00 40 LD HL,4082
40DA ED B0 LDIR

```

Neste está a lógica necessária para compor a tela, usando entre outras figuras, os dizeres do grupo 1. A conjunção dos 3 grupos vai constituir-se no bloco 1 da figura 10.

```

16561 0A 0C 40 00 20 03 00 40
16568 00 00 10 70 00 10 11 00 40
16575 00 00 10 70 00 00 10 00 40
16582 00 00 10 70 00 00 00 00 40
16589 00 00 10 70 00 00 00 00 40
16596 00 00 00 00 00 00 00 00 40
16603 00 00 00 00 00 00 00 00 170

```

Grupo 4 – 16604 a 16630 (40DC a 40F6 em hexa)

```

40DC 2A 00 40 LD HL,(D-FILE)
40DF 11 C5 02 LD DE,02C5
40E2 19 00 ADD HL,DE
40E3 20 7B 40 LD (POSC),HL
40E5 38 00 LD (HL),00
40E8 23 INC HL
40E9 38 04 LD (HL),04
40EB 2B DEC HL
40EC 2B DEC HL
40ED 38 81 LD (HL),81
40EF 11 20 00 LD DE,32(D)
40F2 A7 AND A
40F3 ED 52 SBC HL,DE
40F5 38 05 LD (HL),05

```

Nada além do bloco 2. Sem mais comentários.

```

16604 2A 00 40 11 C5 02 10 35 00
16611 22 7B 40 38 00 20 00 40 00
16618 04 2B 2B 38 81 11 00 00 00
16625 00 A7 ED 52 38 05 50 40 00

```

Grupo 5 – 16631 a 16636 (40F7 a 40FC em hexa)

```

40F7 CD BB 02 CALL KSCAN
40FA 2C INC L
40FB 28 FA JR Z,40F7

```

É o bloco 3. Também sem comentários.

16631 CD 8B 02 2C 28 FA 72E

Grupo 6 – 16637 a 16669 (40FD a 411D em hexa)

```

40FD 3A 34 40 LD A,(QUADROS)
4100 CB 47 BIT 0,A
4102 21 21 40 LD HL,DIRA
4105 38 00 LD (HL),00
4107 28 02 JR Z,410B
4109 38 01 LD (HL),01
410B D8 0B SUB 0B
410D 30 FC JR NC,410B
410F 08 0D ADD A,13(D)
4111 47 LD B,A
4112 2A 0C 40 LD HL,(D-FILE)
4113 11 21 00 LD DE,33(D)
4115 10 DE ADD HL,DE
4116 10 FD DJNZ 4118
411B 22 3C 40 LD (POSA),HL
  
```

Neste grupo, que corresponde ao bloco 4, é utilizada a mesma técnica de sorteio já apresentada anteriormente.

```

16637 3A 34 40 CB 47 21 21 514
16644 40 36 00 28 02 38 01 215
16651 D8 0B 30 FC 06 0D 47 807
16658 2A 0C 40 11 21 00 19 19
16665 10 FD 22 3C 40 427
  
```

Grupo 7 – 16670 a 16797 (411E a 419D em hexa)

```

411E 21 56 40 LD HL,IND
4121 38 00 LD (HL),00
4123 2A 3C 40 LD HL,(POSA)
4126 3A 21 40 LD A,(DIRA)
4129 A7 AND A
  
```

```

412A 00 3A JR Z,4166
412C 00 70 INC HL
412E 00 70 LD A,(HL)
4130 00 06 CP 00
4132 00 00 JR NZ,413E
4134 00 78 CP 78
4136 00 06 JR NZ,413C
4138 00 00 DEC HL
413A 00 00 DEC HL
413C 00 00 LD (HL),00
413E 00 01 JR 40FD
4140 00 17 LD (HL),17
4142 00 30 LD (POSA),HL
4144 00 70 DEC HL
4146 00 70 LD A,(HL)
4148 00 17 CP 17
414A 00 0B JR NZ,4152
414C 00 13 LD (HL),13
414E 00 00 DEC HL
4150 00 13 LD A,(HL)
4152 00 13 CP 13
4154 00 02 JR NZ,4151
4156 00 00 LD (HL),00
4158 00 00 INC HL
415A 00 00 INC HL
415C 00 00 INC HL
415E 00 00 LD A,(HL)
4160 00 00 CP 00
4162 00 49 JR Z,419E
4164 00 78 CP 78
4166 00 41 JR Z,419E
4168 00 1B CP 1B
416A 00 42 JP Z,4209
416C 00 10 LD (HL),12
416E 00 38 JR 419E
4170 00 00 DEC HL
4172 00 7E LD A,(HL)
4174 00 00 CP 00
4176 00 78 JR Z,4178
4178 00 06 CP NZ,4176
417A 00 03 INC HL
417C 00 03 INC HL
417E 00 00 LD (HL),00
4180 00 07 JR 40FD
4182 00 17 LD (HL),17
4184 00 30 LD (POSA),HL
4186 00 70 INC HL
4188 00 7E LD A,(HL)
418A 00 17 CP 17
418C 00 0B JR NZ,418C
418E 00 12 LD (HL),12
  
```

```

4183  LD  A,(HL)
4184  CP  10
4185  JR  NZ,418B
4186  LD  (HL),00
4187  DEC HL
4188  DEC HL
4189  LD  A,(HL)
4190  CP  00
4191  JR  NZ,419E
4192  CP  70
4193  JR  NZ,419E
4194  CP  10
4195  LD  A,40
4196  LD  (HL),10
4197  LD  A,40

```

42

Este é um grupo bastante longo, o que tem sua razão de ser, já que ele está associado aos blocos 5, 6, 7, 8, 9 e 10. O bloco 5 é constituído apenas pelas duas primeiras instruções. Já as demais estão entrelaçadas, e por sua relativa simplicidade, foram analisadas como um só grupo.

Um fato a observar é a duplicação das instruções necessárias em função do Avião estar se deslocando para à esquerda ou à direita. Assim até 4166 a lógica se aplicaria para a última hipótese, e a partir daí para a primeira.

```

16670  LD  A,01
16671  CP  40
16672  JR  NZ,10
16673  LD  A,01
16674  CP  10
16675  JR  NZ,10
16676  LD  A,01
16677  CP  10
16678  JR  NZ,10
16679  LD  A,01
16680  CP  10
16681  JR  NZ,10
16682  LD  A,01
16683  CP  10
16684  JR  NZ,10
16685  LD  A,01
16686  CP  10
16687  JR  NZ,10
16688  LD  A,01
16689  CP  10
16690  JR  NZ,10
16691  LD  A,01
16692  CP  10
16693  JR  NZ,10
16694  LD  A,01
16695  CP  10
16696  JR  NZ,10
16697  LD  A,01
16698  CP  10
16699  JR  NZ,10
16700  LD  A,01
16701  CP  10
16702  JR  NZ,10
16703  LD  A,01
16704  CP  10
16705  JR  NZ,10
16706  LD  A,01
16707  CP  10
16708  JR  NZ,10
16709  LD  A,01
16710  CP  10
16711  JR  NZ,10
16712  LD  A,01
16713  CP  10
16714  JR  NZ,10
16715  LD  A,01
16716  CP  10
16717  JR  NZ,10
16718  LD  A,01
16719  CP  10
16720  JR  NZ,10
16721  LD  A,01
16722  CP  10
16723  JR  NZ,10
16724  LD  A,01
16725  CP  10
16726  JR  NZ,10
16727  LD  A,01
16728  CP  10
16729  JR  NZ,10
16730  LD  A,01
16731  CP  10
16732  JR  NZ,10
16733  LD  A,01
16734  CP  10
16735  JR  NZ,10
16736  LD  A,01
16737  CP  10
16738  JR  NZ,10
16739  LD  A,01
16740  CP  10
16741  JR  NZ,10
16742  LD  A,01
16743  CP  10
16744  JR  NZ,10
16745  LD  A,01
16746  CP  10
16747  JR  NZ,10
16748  LD  A,01
16749  CP  10
16750  JR  NZ,10
16751  LD  A,01
16752  CP  10
16753  JR  NZ,10
16754  LD  A,01
16755  CP  10
16756  JR  NZ,10
16757  LD  A,01
16758  CP  10
16759  JR  NZ,10
16760  LD  A,01
16761  CP  10
16762  JR  NZ,10
16763  LD  A,01
16764  CP  10
16765  JR  NZ,10
16766  LD  A,01
16767  CP  10
16768  JR  NZ,10
16769  LD  A,01
16770  CP  10
16771  JR  NZ,10
16772  LD  A,01
16773  CP  10
16774  JR  NZ,10
16775  LD  A,01
16776  CP  10
16777  JR  NZ,10
16778  LD  A,01
16779  CP  10
16780  JR  NZ,10
16781  LD  A,01
16782  CP  10
16783  JR  NZ,10
16784  LD  A,01
16785  CP  10
16786  JR  NZ,10
16787  LD  A,01
16788  CP  10
16789  JR  NZ,10
16790  LD  A,01
16791  CP  10
16792  JR  NZ,10
16793  LD  A,01
16794  CP  10
16795  JR  NZ,10
16796  LD  A,01
16797  CP  10
16798  JR  NZ,10
16799  LD  A,01
16800  CP  10
16801  JR  NZ,10
16802  LD  A,01
16803  CP  10
16804  JR  NZ,10
16805  LD  A,01
16806  CP  10
16807  JR  NZ,10
16808  LD  A,01
16809  CP  10
16810  JR  NZ,10
16811  LD  A,01
16812  CP  10
16813  JR  NZ,10
16814  LD  A,01
16815  CP  10
16816  JR  NZ,10
16817  LD  A,01
16818  CP  10
16819  JR  NZ,10
16820  LD  A,01
16821  CP  10
16822  JR  NZ,10
16823  LD  A,01
16824  CP  10
16825  JR  NZ,10
16826  LD  A,01
16827  CP  10
16828  JR  NZ,10
16829  LD  A,01
16830  CP  10
16831  JR  NZ,10
16832  LD  A,01
16833  CP  10
16834  JR  NZ,10
16835  LD  A,01
16836  CP  10
16837  JR  NZ,10
16838  LD  A,01
16839  CP  10
16840  JR  NZ,10
16841  LD  A,01
16842  CP  10
16843  JR  NZ,10
16844  LD  A,01
16845  CP  10
16846  JR  NZ,10
16847  LD  A,01
16848  CP  10
16849  JR  NZ,10
16850  LD  A,01
16851  CP  10
16852  JR  NZ,10
16853  LD  A,01
16854  CP  10
16855  JR  NZ,10
16856  LD  A,01
16857  CP  10
16858  JR  NZ,10
16859  LD  A,01
16860  CP  10
16861  JR  NZ,10
16862  LD  A,01
16863  CP  10
16864  JR  NZ,10
16865  LD  A,01
16866  CP  10
16867  JR  NZ,10
16868  LD  A,01
16869  CP  10
16870  JR  NZ,10
16871  LD  A,01
16872  CP  10
16873  JR  NZ,10
16874  LD  A,01
16875  CP  10
16876  JR  NZ,10
16877  LD  A,01
16878  CP  10
16879  JR  NZ,10
16880  LD  A,01
16881  CP  10
16882  JR  NZ,10
16883  LD  A,01
16884  CP  10
16885  JR  NZ,10
16886  LD  A,01
16887  CP  10
16888  JR  NZ,10
16889  LD  A,01
16890  CP  10
16891  JR  NZ,10
16892  LD  A,01
16893  CP  10
16894  JR  NZ,10
16895  LD  A,01
16896  CP  10
16897  JR  NZ,10
16898  LD  A,01
16899  CP  10
16900  JR  NZ,10
16901  LD  A,01
16902  CP  10
16903  JR  NZ,10
16904  LD  A,01
16905  CP  10
16906  JR  NZ,10
16907  LD  A,01
16908  CP  10
16909  JR  NZ,10
16910  LD  A,01
16911  CP  10
16912  JR  NZ,10
16913  LD  A,01
16914  CP  10
16915  JR  NZ,10
16916  LD  A,01
16917  CP  10
16918  JR  NZ,10
16919  LD  A,01
16920  CP  10
16921  JR  NZ,10
16922  LD  A,01
16923  CP  10
16924  JR  NZ,10
16925  LD  A,01
16926  CP  10
16927  JR  NZ,10
16928  LD  A,01
16929  CP  10
16930  JR  NZ,10
16931  LD  A,01
16932  CP  10
16933  JR  NZ,10
16934  LD  A,01
16935  CP  10
16936  JR  NZ,10
16937  LD  A,01
16938  CP  10
16939  JR  NZ,10
16940  LD  A,01
16941  CP  10
16942  JR  NZ,10
16943  LD  A,01
16944  CP  10
16945  JR  NZ,10
16946  LD  A,01
16947  CP  10
16948  JR  NZ,10
16949  LD  A,01
16950  CP  10
16951  JR  NZ,10
16952  LD  A,01
16953  CP  10
16954  JR  NZ,10
16955  LD  A,01
16956  CP  10
16957  JR  NZ,10
16958  LD  A,01
16959  CP  10
16960  JR  NZ,10
16961  LD  A,01
16962  CP  10
16963  JR  NZ,10
16964  LD  A,01
16965  CP  10
16966  JR  NZ,10
16967  LD  A,01
16968  CP  10
16969  JR  NZ,10
16970  LD  A,01
16971  CP  10
16972  JR  NZ,10
16973  LD  A,01
16974  CP  10
16975  JR  NZ,10
16976  LD  A,01
16977  CP  10
16978  JR  NZ,10
16979  LD  A,01
16980  CP  10
16981  JR  NZ,10
16982  LD  A,01
16983  CP  10
16984  JR  NZ,10
16985  LD  A,01
16986  CP  10
16987  JR  NZ,10
16988  LD  A,01
16989  CP  10
16990  JR  NZ,10
16991  LD  A,01
16992  CP  10
16993  JR  NZ,10
16994  LD  A,01
16995  CP  10
16996  JR  NZ,10
16997  LD  A,01
16998  CP  10
16999  JR  NZ,10
17000  LD  A,01
17001  CP  10
17002  JR  NZ,10
17003  LD  A,01
17004  CP  10
17005  JR  NZ,10
17006  LD  A,01
17007  CP  10
17008  JR  NZ,10
17009  LD  A,01
17010  CP  10
17011  JR  NZ,10
17012  LD  A,01
17013  CP  10
17014  JR  NZ,10
17015  LD  A,01
17016  CP  10
17017  JR  NZ,10
17018  LD  A,01
17019  CP  10
17020  JR  NZ,10
17021  LD  A,01
17022  CP  10
17023  JR  NZ,10
17024  LD  A,01
17025  CP  10
17026  JR  NZ,10
17027  LD  A,01
17028  CP  10
17029  JR  NZ,10
17030  LD  A,01
17031  CP  10
17032  JR  NZ,10
17033  LD  A,01
17034  CP  10
17035  JR  NZ,10
17036  LD  A,01
17037  CP  10
17038  JR  NZ,10
17039  LD  A,01
17040  CP  10
17041  JR  NZ,10
17042  LD  A,01
17043  CP  10
17044  JR  NZ,10
17045  LD  A,01
17046  CP  10
17047  JR  NZ,10
17048  LD  A,01
17049  CP  10
17050  JR  NZ,10
17051  LD  A,01
17052  CP  10
17053  JR  NZ,10
17054  LD  A,01
17055  CP  10
17056  JR  NZ,10
17057  LD  A,01
17058  CP  10
17059  JR  NZ,10
17060  LD  A,01
17061  CP  10
17062  JR  NZ,10
17063  LD  A,01
17064  CP  10
17065  JR  NZ,10
17066  LD  A,01
17067  CP  10
17068  JR  NZ,10
17069  LD  A,01
17070  CP  10
17071  JR  NZ,10
17072  LD  A,01
17073  CP  10
17074  JR  NZ,10
17075  LD  A,01
17076  CP  10
17077  JR  NZ,10
17078  LD  A,01
17079  CP  10
17080  JR  NZ,10
17081  LD  A,01
17082  CP  10
17083  JR  NZ,10
17084  LD  A,01
17085  CP  10
17086  JR  NZ,10
17087  LD  A,01
17088  CP  10
17089  JR  NZ,10
17090  LD  A,01
17091  CP  10
17092  JR  NZ,10
17093  LD  A,01
17094  CP  10
17095  JR  NZ,10
17096  LD  A,01
17097  CP  10
17098  JR  NZ,10
17099  LD  A,01
17100  CP  10
17101  JR  NZ,10
17102  LD  A,01
17103  CP  10
17104  JR  NZ,10
17105  LD  A,01
17106  CP  10
17107  JR  NZ,10
17108  LD  A,01
17109  CP  10
17110  JR  NZ,10
17111  LD  A,01
17112  CP  10
17113  JR  NZ,10
17114  LD  A,01
17115  CP  10
17116  JR  NZ,10
17117  LD  A,01
17118  CP  10
17119  JR  NZ,10
17120  LD  A,01
17121  CP  10
17122  JR  NZ,10
17123  LD  A,01
17124  CP  10
17125  JR  NZ,10
17126  LD  A,01
17127  CP  10
17128  JR  NZ,10
17129  LD  A,01
17130  CP  10
17131  JR  NZ,10
17132  LD  A,01
17133  CP  10
17134  JR  NZ,10
17135  LD  A,01
17136  CP  10
17137  JR  NZ,10
17138  LD  A,01
17139  CP  10
17140  JR  NZ,10
17141  LD  A,01
17142  CP  10
17143  JR  NZ,10
17144  LD  A,01
17145  CP  10
17146  JR  NZ,10
17147  LD  A,01
17148  CP  10
17149  JR  NZ,10
17150  LD  A,01
17151  CP  10
17152  JR  NZ,10
17153  LD  A,01
17154  CP  10
17155  JR  NZ,10
17156  LD  A,01
17157  CP  10
17158  JR  NZ,10
17159  LD  A,01
17160  CP  10
17161  JR  NZ,10
17162  LD  A,01
17163  CP  10
17164  JR  NZ,10
17165  LD  A,01
17166  CP  10
17167  JR  NZ,10
17168  LD  A,01
17169  CP  10
17170  JR  NZ,10
17171  LD  A,01
17172  CP  10
17173  JR  NZ,10
17174  LD  A,01
17175  CP  10
17176  JR  NZ,10
17177  LD  A,01
17178  CP  10
17179  JR  NZ,10
17180  LD  A,01
17181  CP  10
17182  JR  NZ,10
17183  LD  A,01
17184  CP  10
17185  JR  NZ,10
17186  LD  A,01
17187  CP  10
17188  JR  NZ,10
17189  LD  A,01
17190  CP  10
17191  JR  NZ,10
17192  LD  A,01
17193  CP  10
17194  JR  NZ,10
17195  LD  A,01
17196  CP  10
17197  JR  NZ,10
17198  LD  A,01
17199  CP  10
17200  JR  NZ,10

```

Grupo 8 — 16798 a 16827 (419E a 41BB em hexa)

```

419E  LD  HL,(D-FILE)
419F  LD  DE,778(D)
41A0  ADD HL,DE
41A1  LD  A,(HL)
41A2  AND A
41A3  JR  NZ,41B1
41A4  LD  B,04
41A5  INC HL
41A6  LD  (HL),10
41A7  DJNZ 41A6
41A8  RET
41A9  DEC A
41AA  CP  10
41AB  JR  NZ,41BB
41AC  LD  (HL),20
41AD  DEC HL
41AE  JR  41A5
41AF  LD  (HL),A

```

Este grupo corresponde a uma rotina muito útil, que pode vir a ser usada em outros jogos. Trata-se do controle do tempo restante na tela, dos blocos 11 e 12, que se inicia de um número qualquer, no caso 1000, e termina com zeros, na mesma quantidade que o número de casas do número inicial. A mesma rotina poderia ser usada se o número inicial fosse 2000 por exemplo, porém se fosse 30.000, deveríamos mudar a instrução LD B, 04 do endereço 41A9 para LD B, 05. Pelo potencial de uso futuro vale a pena saber o porquê, e entender o conjunto.

```

16798  LD  A,00
16799  CP  00
16800  JR  NZ,10
16801  LD  A,00
16802  CP  00
16803  JR  NZ,10
16804  LD  A,00
16805  CP  00
16806  JR  NZ,10
16807  LD  A,00
16808  CP  00
16809  JR  NZ,10
16810  LD  A,00
16811  CP  00
16812  JR  NZ,10
16813  LD  A,00
16814  CP  00
16815  JR  NZ,10
16816  LD  A,00
16817  CP  00
16818  JR  NZ,10
16819  LD  A,00
16820  CP  00
16821  JR  NZ,10
16822  LD  A,00
16823  CP  00
16824  JR  NZ,10
16825  LD  A,00
16826  CP  00
16827  JR  NZ,10

```

Grupo 9 — 16828 a 16917 (41BC a 4215 em hexa)

```

41BC  LD  HL,(POSTA1)
41BD  LD  A,H
41BE  OR  L
41BF  JR  NZ,41EA
41C0  LD  HL,(POSTA)

```

```

4106 11 22 00 LD DE,34(D)
4109 11 22 00 ADD HL,DE
410B 77 77 00 DEC A,(HL)
410D 77 77 00 OR 00
410E 77 77 00 NZ,41E8
410F 77 77 00 CP 00
4110 77 77 00 NZ,41E8
4111 77 77 00 CP 00
4112 77 77 00 NZ,41DF
4113 77 77 00 CP 00
4114 77 77 00 NZ,41DF
4115 77 77 00 CP 00
4116 77 77 00 NZ,4106
4117 77 77 00 LD HL,(POSTA)
4118 77 77 40 ADD HL,DE
4119 77 77 40 LD LD,(POSTA1),HL
411A 77 77 40 LD (HL),27(D)
411B 77 77 40 JR 4216
411C 77 77 40 LD HL,(POSTA2)
411D 77 77 40 A,H
411E 77 77 40 OR L
411F 77 77 40 NZ,4216
4120 77 77 40 LD HL,(POSTA)
4121 77 77 40 DE,32(D)
4122 77 77 40 ADD HL,DE
4123 77 77 40 DEC A,(HL)
4124 77 77 40 OR 00
4125 77 77 40 NZ,4216
4126 77 77 40 CP 00
4127 77 77 40 NZ,4216
4128 77 77 40 CP 00
4129 77 77 40 NZ,420D
412A 77 77 40 CP 04
412B 77 77 40 NZ,420D
412C 77 77 40 CP 00
412D 77 77 40 NZ,41F7
412E 77 77 40 LD HL,(POSTA)
412F 77 77 40 ADD HL,DE
4130 77 77 40 LD (POSTA2),HL
4131 77 77 40 LD (HL),27(D)

```

Eis outro grupo bastante longo. Como o 7 ele repete a mesma lógica duas vezes, em função do Avião estar movendo-se para à direita ou à esquerda e está associado aos blocos 13, 14 e 15.

Como comentário merece destaque a forma usada para determinar se o canhão está “enquadrado” na mira, através do exame de todas as posições da tela, ao longo de uma reta a 45º partindo do avião.

Isto é feito usando os Registros POSTA1 e POSTA2 que, se estiverem “ocupados” (com conteúdo diferente de zero), não seriam usados. Este cui-

dado garante que não se tenha mais de um tiro em cada direção, e que um segundo parta apenas quando o primeiro já “sumiu”.

```

1600 77 77 00 NZ,4216
1601 77 77 00 CP 00
1602 77 77 00 NZ,4216
1603 77 77 00 CP 00
1604 77 77 00 NZ,4216
1605 77 77 00 CP 00
1606 77 77 00 NZ,4216
1607 77 77 00 CP 00
1608 77 77 00 NZ,4216
1609 77 77 00 CP 00
160A 77 77 00 NZ,4216
160B 77 77 00 CP 00
160C 77 77 00 NZ,4216
160D 77 77 00 CP 00
160E 77 77 00 NZ,4216
160F 77 77 00 CP 00
1610 77 77 00 NZ,4216
1611 77 77 00 CP 00
1612 77 77 00 NZ,4216
1613 77 77 00 CP 00
1614 77 77 00 NZ,4216
1615 77 77 00 CP 00
1616 77 77 00 NZ,4216
1617 77 77 00 CP 00
1618 77 77 00 NZ,4216
1619 77 77 00 CP 00
161A 77 77 00 NZ,4216
161B 77 77 00 CP 00
161C 77 77 00 NZ,4216
161D 77 77 00 CP 00
161E 77 77 00 NZ,4216
161F 77 77 00 CP 00
1620 77 77 00 NZ,4216
1621 77 77 00 CP 00
1622 77 77 00 NZ,4216
1623 77 77 00 CP 00
1624 77 77 00 NZ,4216
1625 77 77 00 CP 00

```

Grupo 10 – 16918 a 16925 (4216 a 421D em hexa)

```

4216 01 00 02 LD HL,0200
4219 0B 00 DEC HL
421A 7C 00 LD A,H
421B 00 00 OR L
421C 00 FB JR NZ,4219

```

Nada além da já conhecida rotina de consumo de tempo, que neste caso corresponde ao bloco 16.

```

16918 01 00 02 2B 7C B5 20 415
16925 7B 251

```

Grupo 11 – 16926 a 16997 (421E a 4265 em hexa)

```

421E 2A 3E 40 LD HL,(POSTA1)
4221 7C 00 LD A,H
4222 0B 00 OR L
4223 00 1C JR NZ,4241
4225 00 00 LD (HL),00
4227 11 22 00 LD DE,34(D)
422A 10 00 ADD HL,DE
422B 7E 00 LD A,(HL)

```

```

40000 CP    0A
40001 LJR  NZ,4235
40002 LDY  HL,0000
40003 LJR  423E
40004 CP   1B
40005 LJR  Z,423E
40006 AND  A
40007 LJR  NZ,4265
40008 LDY  (HL),1B
40009 LD  (POSTA1),HL
40010 LD  HL,(POSTA2)
40011 LJR  A,H
40012 OR  L
40013 JR  Z,428D
40014 LD  (HL),00
40015 LD  DE,32(D)
40016 ADD  HL,DE
40017 LD  A,(HL)
40018 CP  0A
40019 JR  NZ,4258
40020 LD  HL,0000
40021 JR  4261
40022 CP  1B
40023 JR  Z,4261
40024 AND  A
40025 JR  NZ,4265
40026 LDY (HL),1B
40027 LD  (POSTA2),HL
40028 JR  428D

```

Novamente a duplicidade, em função agora de haver 2 tiros possíveis do Avião, o registrado no POSTA1 e outro no POSTA2. Associado a este estão os blocos 17 e 18.

Cabe notar, que se na próxima posição do tiro houver a base (figura código 10) ou a lateral (figura código 8) ele some, e no POSTA correspondente carrega-se zero.

```

10000 1B 00 00 00 00 00 00 00 00 00 00
10001 1B 00 00 00 00 00 00 00 00 00 00
10002 1B 00 00 00 00 00 00 00 00 00 00
10003 1B 00 00 00 00 00 00 00 00 00 00
10004 1B 00 00 00 00 00 00 00 00 00 00
10005 1B 00 00 00 00 00 00 00 00 00 00
10006 1B 00 00 00 00 00 00 00 00 00 00
10007 1B 00 00 00 00 00 00 00 00 00 00
10008 1B 00 00 00 00 00 00 00 00 00 00
10009 1B 00 00 00 00 00 00 00 00 00 00
10010 1B 00 00 00 00 00 00 00 00 00 00
10011 1B 00 00 00 00 00 00 00 00 00 00
10012 1B 00 00 00 00 00 00 00 00 00 00
10013 1B 00 00 00 00 00 00 00 00 00 00
10014 1B 00 00 00 00 00 00 00 00 00 00
10015 1B 00 00 00 00 00 00 00 00 00 00
10016 1B 00 00 00 00 00 00 00 00 00 00
10017 1B 00 00 00 00 00 00 00 00 00 00
10018 1B 00 00 00 00 00 00 00 00 00 00
10019 1B 00 00 00 00 00 00 00 00 00 00

```

Grupo 12 – 16998 a 17036 (4266 a 428C em hexa)

```

4266 0D 01 43 CALL 4301
4267 2A 7B 40 LD HL,(P03C)
4268 0B 00 LD (HL),00
4269 20 INC HL
4270 0B 00 LD (HL),00
4271 2B DEC HL
4272 2B DEC HL
4273 0B 00 LD (HL),00
4274 27 AND A
4275 11 20 00 LD DE,32(D)
4276 2D SBC HL,DE
4277 0B 00 LD (HL),00
4278 2A 0C 40 LD HL,(D-FILE)
4279 11 FE 02 LD DE,766(D)
4280 19 ADD HL,DE
4281 7E LD A,(HL)
4282 3D DEC A
4283 77 LD (HL),A
4284 7E 1C CP 1C
4285 0B RET Z
4286 0C DC 40 JP 40DC

```

Este está associado aos blocos 19 e 20. De início há uma chamada (CALL) à sub-rotina já citada, de nome ‘BUM”, no endereço 17153 (4301 em hexa) que será vista adiante em conexão com o bloco 24. O resto, “apagar” o canhão, reduzir o número de vidas na tela, e ver se chegou a zero, não merece maiores comentários.

```

16998 0D 01 43 2A 7B 40 0B 00
17005 00 03 0B 00 2B 0B 0B 00
17012 00 27 11 20 00 2D 02 5000
17019 3B 00 2A 0C 40 11 FE 443
17026 02 19 7E 3D 77 FE 1C 615
17033 0B 0C DC 40 00 00 676

```

Grupo 13 – 17037 a 17086 (428D a 42BE em hexa)

```

428D 06 0A LD B,0A
428E 11 42 40 LD DE,POSTC1
428F 1D LD A,(DE)
4290 8F LD L,A
4291 13 INC DE
4292 1D LD A,(DE)
4293 67 LD H,A

```

```

42007 A7 AND A
42008 20 00 JR Z,4200C
42009 30 00 LD (HL),00
4200C 08 PUSH DE
4200D 11 21 00 LD DE,33(D)
4200E 0D 00 SBC HL,DE
4200F 01 00 POP DE
42010 7E LD A,(HL)
42011 FE 00 CP 00
42012 20 00 JR NZ,4201D
42013 21 00 00 LD HL,0000
42014 18 00 JR 42016
42015 FE 10 CP 10
42016 20 00 JR Z,42019
42017 A7 AND A
42018 20 15 JR NZ,4201B
42019 36 15 LD (HL),1B
4201A 7C LD A,H
4201B 12 LD (DE),A
4201C 18 DEC DE
4201D 7D LD A,L
4201E 12 LD (DE),A
4201F 13 INC DE
42020 13 INC DE
42021 10 03 DJNZ 42022

```

Trata-se de um grupo bastante simples, ligado aos blocos 21 e 22, apesar de movimentar os 10 tiros possíveis de existir na tela, partidos do Canhão. A facilidade decorre do fato dos tiros serem verticais, e qualquer coisa que esteja na frente de um deles, não sendo o topo (figura código 9), ou um outro tiro (necessariamente disparado pelo avião), terá que ser o próprio Avião. Se for o topo, apaga-se o tiro em questão e zera-se o POSTC correspondente, abrindo registro ou espaço para um novo disparo do Canhão.

```

17037 06 0A 11 42 40 1A 6F 300
17044 13 1A 07 A7 28 22 36 443
17051 00 05 11 21 00 0D 5D 500
17056 01 7E 7E 09 20 05 21 000
17065 00 00 18 00 7E 1B 08 054
17072 05 A7 26 19 36 1B 7C 400
17079 12 1B 7D 12 13 13 10 042
17086 03 211

```

Grupo 14 – 17087 a 17096 (42BF a 42C8 em hexa)

```

42BF 3A 56 40 LD A,(IND)
42C2 A7 AND A
42C3 CA 52 43 JP Z,4352
42C6 03 1E 41 JP 411E

```

Eis mais um que pela simplicidade, não cabe comentar. É apenas o grupo relacionado com o bloco 23.

```

17087 3A 56 40 A7 CA 52 43 74E
17094 03 1E 41 290

```

Grupo 15 – 17097 a 17249 (42C9 a 4361 em hexa)

```

42C9 0D 01 43 CALL 4301
42CA 2A 0C 40 LD HL,(D-FILE)
42CB 11 16 03 LD DE,780(D)
42CC 10 ADD HL,DE
42CD 7E LD A,(HL)
42CE A7 AND A
42CF 08 RET Z
42D0 00 INC A
42D1 7E 28 CP 28
42D2 00 05 JR NZ,42E0
42D3 35 1C LD (HL),1C
42D4 28 DEC HL
42D5 10 73 JR 42D3
42D6 77 LD (HL),A
42D7 03 7D 40 JP 40FD
42D8 7E 75 PUSH HL
42D9 2D 0C 40 LD HL,(D-FILE)
42DA 08 18 LD B,18
42DB 23 INC HL
42DC 7E LD A,(HL)
42DD 7E 76 CP 76
42DE 00 05 JR Z,42FB
42DF 08 00 ADD A,80
42E0 77 LD (HL),A
42E1 10 75 JR 42EA
42E2 10 73 DJNZ 42EA
42E3 7E 71 POP HL
42E4 11 00 10 LD DE,1000
42E5 18 DEC DE
42E6 7A LD A,D
42E7 03 OR E
42E8 00 7B JR NZ,42FB
42E9 00 RET
42EA 00 DEC HL
42EB 06 27 LD (HL),27
42EC 0D E4 42 CALL 42E4
42ED 03 INC HL

```

```

43008 06 BA LD (HL),BA
43009 0D E4 42 CALL 42E4
4300D 03 INC HL
4300E 06 32 LD (HL),32
43010 0D E4 42 CALL 42E4
43013 0D 0B DEC HL
43014 0D 0B DEC HL
43015 06 80 LD (HL),80
43017 03 INC HL
43018 0D E4 42 CALL 42E4
4301B 06 00 LD (HL),00
4301D 03 INC HL
4301E 0D E4 42 CALL 42E4
43021 06 80 LD (HL),80
43023 0D E4 42 CALL 42E4
43028 06 00 LD B,00
43029 01 3E 40 LD HL,POSTA1
4302B 03 0E LD E,(HL)
4302C 03 INC HL
4302D 06 06 LD D,(HL)
4302E 06 06 EX DE,HL
4302F 07 0C LD A,H
43030 06 00 OR L
43031 06 00 JR Z,4335
43032 06 00 LD (HL),00
43033 03 13 INC DE
43034 06 06 EX DE,HL
43037 10 F2 DJNZ 4325
43038 0D F8 42 CALL 42F8
4303C 0D 3C 40 LD HL,(POSA)
4303E 0D 0B DEC HL
43040 07 0E LD A,(HL)
43041 07 13 CP 13
43043 06 02 JR NZ,4347
43045 06 00 LD (HL),00
43047 03 03 INC HL
43048 07 0E LD A,(HL)
43049 07 17 CP 17
4304B 06 02 JR NZ,434F
4304D 06 00 LD (HL),00
4304F 03 03 INC HL
43050 07 0E LD A,(HL)
43051 07 12 CP 12
43053 06 02 JR NZ,4357
43055 06 00 LD (HL),00
43057 06 18 LD B,18
43059 01 3E 40 LD HL,POSTA1
4305C 06 00 LD (HL),00
4305E 03 03 INC HL
4305F 10 FB DJNZ 435C
43061 09 09 RET

```

Este é outro dos tais, do qual pode ser dito — “porque simplificar se é possível complicar”, já que, estando ligado aos blocos 24 e 25, foi montado em ordem totalmente inversa a destes. Além disso ele faz mais funções do que as discriminadas pelos blocos. Cabe então desmembrar o grupo em sub-grupos.

O primeiro sub-grupo, do 42C9 a 42E3, inicia chamando a sub-rotina “BUM”, e aumenta de 10 a contagem da tela. A lógica desta contagem é diferente da usada no capítulo 3, e merece ser estudada e entendida perfeitamente, já que poderá ser usada em outros jogos. O que ela tem de característico é partir de quatro zeros, e prever um retorno ao basic, ou seja, um final, caso a contagem supere o valor 9999. Isto é conseguido através dos sucessivos “vai-um”, se for encontrada uma casa vazia; este fato é verificado pela seqüência LD A, (HL) — AND A — RET Z. Como pela característica do jogo esta contagem é praticamente inatingível, não foi incluída a alternativa no diagrama de blocos.

O segundo sub-grupo é o da sub-rotina “BUM”, não localizada imediatamente após o primeiro, mas sim entre os endereços 4301 a 4315 inclusive. Ele é iniciado por uma série de seis chamadas (CALL) a uma segunda sub-rotina, a do efeito “claro/escuro”, intercaladas pela escrita e apagamento das letras B — U — M no local onde o tiro encontrou o seu alvo (avião ou canhão) — daí o nome da sub-rotina. Na seqüência, o Avião e os tiros são apagados, inclusive os correspondentes Registros (POSA, POSTC e POSTA). Cabe notar que o Avião é apagado quer seja ele o atingido, quer seja o Canhão. Já este, só é apagado se for o atingido.

Finalmente o terceiro sub-grupo é o intercalado entre os outros dois, no endereço 42E4 a 4300, e constitui a sub-rotina “claro/escuro”. Esta é outra que pelo impacto visual que causa pode ser usada em outros programas. O que ela faz é muito simples — soma 128 (80 em hexa) ao código de cada um dos símbolos na tela e como consequência, tudo que tinha fundo preto passa a ter fundo branco e vice-versa. Na seqüência existe um tradicional e conhecido programa que consome tempo — o efeito claro/escuro deve ter uma velocidade de apresentação mais lenta que a do deslocamento do avião.

```

17097 0D 01 43 0A 0C 40 11 408
17104 10 03 10 7E A7 08 0C 603
17111 FE 00 00 00 36 1C 0C 608
17118 10 73 77 00 7D 40 08 1107
17125 0A 0C 40 00 10 03 7E 3007
17132 FE 78 00 00 06 00 77 862
17139 10 75 10 73 E1 11 00 770
17146 10 10 7D 03 00 7B 00 828

```



```

17165 00 00 07 00 04 40 00 67 00
17166 00 00 00 00 04 40 00 00 00
17167 00 00 00 04 40 00 00 00 00
17174 00 00 00 00 04 40 00 00 00
17181 00 00 00 04 40 00 00 00 00
17188 00 04 40 00 00 00 00 00 00
17195 00 00 00 00 00 00 00 00 00
17202 00 00 00 00 00 00 00 00 00
17209 00 00 00 00 00 00 00 00 00
17216 00 00 00 00 00 00 00 00 00
17223 00 00 00 00 00 00 00 00 00
17230 00 00 00 00 00 00 00 00 00
17237 00 00 00 00 00 00 00 00 00
17244 00 00 00 00 00 00 00 00 00

```

AND, ou seja, pressionando o 5 (seta à esquerda) e o zero (disparo) teremos em HL

```

Devido ao 5      DF 7
Devido ao zero   FD EF
Com ambos        DF F7 AND FDEF ou DDE7

```

Fazendo LD A, H – AND L resulta em A o valor C5 que foi usado na comparação. Como foi visto no capítulo 3, esta prática de combinar o H com o L tem a desvantagem que o comando não é somente conseguido com as teclas eleitas para tal fim. Porém a simplicidade da programação e o uso do joystick compensam esta deficiência. Recomenda-se repetir o raciocínio a fim de constatar como chegou-se ao E5 associado a Direita + Disparo.

O resultado da comparação comanda chamadas às sub-rotinas de disparo (4393 em hexa), esquerda (43CD) e direita (4406).

Grupo 16 – 17250 a 17298 (4362 a 4391 em hexa)

```

4362 21 56 40      LD    HL,4056
4365 00 01 00      LD    (HL),01
4367 00 00 00      CALL 02BB
4368 70 00 00      LD    A,H
436B A9 00 00      AND  L
436C FE 07 00      CP   D7
436E 00 00 43      CALL Z,43CD
4371 FE E7 00      CP   E7
4373 00 06 44      CALL Z,4406
4376 FE ED 00      CP   ED
4378 00 03 43      CALL Z,4393
437B FE 05 00      CP   05
437D 00 06 00      JR   NZ,4385
437F 00 03 43      CALL 4393
4382 00 00 43      CALL 43CD
4385 FE E5 00      CP   E5
4387 02 15 42      JP   NZ,4216
438A 00 03 43      CALL 4393
438D 00 06 44      CALL 4406
4390 03 16 42      JP   4216

```

Este é o penúltimo grupo e corresponde aos blocos 26, 27 e 28. Fazer IND = 1 (bloco 26) é extremamente simples e é atendido pelas duas primeiras instruções do grupo.

Já verificar se uma ou mais teclas foram pressionadas, identificá-las em caso positivo ou saltar para o bloco 16 em caso contrário, é a tarefa das demais instruções. A novidade é aprender o que acontece em HL se, usando o KScan, forem pressionadas duas teclas simultaneamente. A resposta é um

```

17250 21 56 40 00 01 00 00 630
17257 00 00 70 00 FE 07 00 00 1169
17264 43 00 FE E7 00 06 44 FE 1084
17271 ED 00 03 43 FE 05 20 1168
17278 06 00 00 43 00 00 43 902
17285 FE E5 00 16 42 00 03 1117
17292 43 00 06 44 03 16 42 829

```

Grupo 17 – 17299 a 17462 (4393 a 4436 em hexa)

```

4393 F5 00 00      PUSH AF
4394 0A 75 40      LD    HL,(POSC)
4397 11 42 00      LD    DE,66(D)
439A A7 00 00      AND  A
439B ED 52 00      SBC  HL,DE
439D 06 1B 00      LD    (HL),1B
439F 11 41 40      LD    DE,POSTC1
43A2 EB 00 00      EX   DE,HL
43A3 03 00 00      INC  HL
43A4 03 00 00      INC  HL
43A5 7E 00 00      LD  A,(HL)
43A6 A7 00 00      AND  A
43A7 00 FA 00      JR   NZ,43A3
43A9 72 00 00      LD  (HL),D
43AA 0B 00 00      DEC  HL
43AB 73 00 00      LD  (HL),E
43AC 0A 00 40      LD  HL,(D-FILE)
43AF 11 17 03      LD  DE,0317
43B2 10 00 00      ADD  HL,DE
43B3 7E 00 00      LD  A,(HL)
43B4 A7 00 00      AND  A

```

4308	00	00	JR	NZ,4300
4309	00	04	LD	B,04
430A	00	03	INC	HL
430B	00	10	LD	(HL),10
430C	00	00	DJNZ	43B9
430D	00	00	POP	AF
430E	00	00	RET	
430F	00	00	DEC	A
4310	00	1B	CP	1B
4311	00	00	JR	NZ,430A
4312	00	00	LD	(HL),25
4313	00	00	DEC	HL
4314	00	00	JR	43B3
4315	00	00	LD	(HL),A
4316	00	00	POP	AF
4317	00	00	RET	
4318	00	00	PUSH	AF
4319	00	7B 40	LD	HL,(POSC)
431A	00	00	DEC	HL
431B	00	00	DEC	HL
431C	00	00	LD	A,(HL)
431D	00	00	CP	00
431E	00	04	JR	Z,43FC
431F	00	10	CP	1B
4320	00	00	JR	Z,43FE
4321	00	01	LD	(HL),81
4322	00	00	INC	HL
4323	00	00	LD	(HL),80
4324	00	7B 40	LD	(POSC),HL
4325	00	04	INC	HL
4326	00	04	LD	(HL),04
4327	00	00	INC	HL
4328	00	00	LD	(HL),00
4329	00	21 00	LD	DE,34(D)
432A	00	07	AND	A
432B	00	52	SBC	HL,DE
432C	00	00	LD	(HL),00
432D	00	00	DEC	HL
432E	00	00	LD	A,(HL)
432F	00	1B	CP	1B
4330	00	00	JR	Z,43FE
4331	00	00	LD	(HL),05
4332	00	00	NOP	
4333	00	00	NOP	
4334	00	00	POP	AF
4335	00	00	RET	
4336	00	00	POP	AF
4337	00	11 66 42	LD	DE,4266
4402	00	00	EX	DE,HL
4403	00	00	EX	(SP),HL
4404	00	00	EX	DE,HL
4405	00	00	RET	

4405	00	00	PUSH	AF
4406	00	7B 40	LD	HL,(POSC)
4407	00	00	INC	HL
4408	00	00	INC	HL
4409	00	00	LD	A,(HL)
440A	00	00	CP	00
440B	00	00	JR	Z,43FC
440C	00	1B	CP	1B
440D	00	00	JR	Z,43FE
440E	00	04	LD	(HL),04
440F	00	00	DEC	HL
4410	00	00	LD	(HL),80
4411	00	7B 40	LD	(POSC),HL
4412	00	81	LD	(HL),81
4413	00	00	DEC	HL
4414	00	00	LD	(HL),00
4415	00	11 20 00	LD	DE,32(D)
4416	00	07	AND	A
4417	00	52	SBC	HL,DE
4418	00	00	LD	(HL),00
4419	00	00	INC	HL
441A	00	00	LD	A,(HL)
441B	00	1B	CP	1B
441C	00	00	JR	Z,4401
441D	00	05	LD	(HL),05
441E	00	00	NOP	
441F	00	00	NOP	
4420	00	F1	POP	AF
4421	00	00	RET	

Este último grupo, associado ao bloco 29, contém exatamente as três sub-rotinas citadas, no grupo anterior.

A primeira, entre 4393 e 43CC, resulta no Disparo do Canhão. De início é desenhado um "tiro" na boca do Canhão. Em seguida a localização deste tiro é registrada em um dos POSTC, mais precisamente, no primeiro que for encontrado vago. Por fim é utilizada a mesma lógica de redução de contagem que foi aplicada no bloco 11 para o tempo restante de jogo desde que a contagem não seja 0000.

O PUSH AF no início da sub-rotina e o POP AF no final visa preservar o valor de A para as comparações múltiplas previstas no grupo anterior.

As duas outras sub-rotinas, Esquerda (43CD a 4405) e Direita (4406 a 4436) são muito semelhantes entre si. De início é verificado se na direção a ser movimentado o canhão existe a parede ou um tiro; não havendo, ele é movido. Se houver a parede, dá-se o retorno sem movimentação, e no caso de

um tiro, o endereço de retorno é alterado de forma a ocorrer um salto para o bloco 19. Isto é feito através da seqüência LD DE, 16998 (116642 – 16998 ou 4266 em hexa é o endereço do bloco 19) – EX DE, HL – EX (SP), HL – EX DE, HL – RET – com que o endereço em SP passa a ser do bloco 19 e o valor de HL é preservado.

Para finalizar, cabe alertar que existem alguns poucos bytes vagos nos dois últimos sub-grupos de instruções, que funcionam como reserva apenas.

```

172000 F0 00 00 07 40 11 40 00 00 05 07
173000 07 00 00 00 00 11 11 00 00 00 04
173100 40 00 00 00 00 07 07 00 00 04 00
173200 07 00 00 00 00 00 00 00 00 04 00
173300 11 17 00 10 07 00 00 00 00 00 00
173400 00 00 00 00 00 00 10 00 00 00 00
173410 00 00 00 00 00 00 00 00 00 00 00
173420 00 00 00 00 00 00 00 00 00 00 00
173430 00 00 00 00 00 00 00 00 00 00 00
173440 00 00 00 00 00 00 00 00 00 00 00
173450 00 00 00 00 00 00 00 00 00 00 00
173460 00 00 00 00 00 00 00 00 00 00 00
173470 00 00 00 00 00 00 00 00 00 00 00
173480 00 00 00 00 00 00 00 00 00 00 00
173490 00 00 00 00 00 00 00 00 00 00 00
173500 00 00 00 00 00 00 00 00 00 00 00
173510 00 00 00 00 00 00 00 00 00 00 00
173520 00 00 00 00 00 00 00 00 00 00 00
173530 00 00 00 00 00 00 00 00 00 00 00
173540 00 00 00 00 00 00 00 00 00 00 00
173550 00 00 00 00 00 00 00 00 00 00 00
173560 00 00 00 00 00 00 00 00 00 00 00
173570 00 00 00 00 00 00 00 00 00 00 00
173580 00 00 00 00 00 00 00 00 00 00 00
173590 00 00 00 00 00 00 00 00 00 00 00
174000 00 00 00 00 00 00 00 00 00 00 00
174010 00 00 00 00 00 00 00 00 00 00 00
174020 00 00 00 00 00 00 00 00 00 00 00
174030 00 00 00 00 00 00 00 00 00 00 00
174040 00 00 00 00 00 00 00 00 00 00 00
174050 00 00 00 00 00 00 00 00 00 00 00
174060 00 00 00 00 00 00 00 00 00 00 00
174070 00 00 00 00 00 00 00 00 00 00 00
174080 00 00 00 00 00 00 00 00 00 00 00
174090 00 00 00 00 00 00 00 00 00 00 00
174100 00 00 00 00 00 00 00 00 00 00 00
174110 00 00 00 00 00 00 00 00 00 00 00
174120 00 00 00 00 00 00 00 00 00 00 00
174130 00 00 00 00 00 00 00 00 00 00 00
174140 00 00 00 00 00 00 00 00 00 00 00
174150 00 00 00 00 00 00 00 00 00 00 00
174160 00 00 00 00 00 00 00 00 00 00 00
174170 00 00 00 00 00 00 00 00 00 00 00
174180 00 00 00 00 00 00 00 00 00 00 00
174190 00 00 00 00 00 00 00 00 00 00 00
174200 00 00 00 00 00 00 00 00 00 00 00
174210 00 00 00 00 00 00 00 00 00 00 00
174220 00 00 00 00 00 00 00 00 00 00 00
174230 00 00 00 00 00 00 00 00 00 00 00
174240 00 00 00 00 00 00 00 00 00 00 00
174250 00 00 00 00 00 00 00 00 00 00 00
174260 00 00 00 00 00 00 00 00 00 00 00
174270 00 00 00 00 00 00 00 00 00 00 00
174280 00 00 00 00 00 00 00 00 00 00 00
174290 00 00 00 00 00 00 00 00 00 00 00
174300 00 00 00 00 00 00 00 00 00 00 00
174310 00 00 00 00 00 00 00 00 00 00 00
174320 00 00 00 00 00 00 00 00 00 00 00
174330 00 00 00 00 00 00 00 00 00 00 00
174340 00 00 00 00 00 00 00 00 00 00 00
174350 00 00 00 00 00 00 00 00 00 00 00
174360 00 00 00 00 00 00 00 00 00 00 00
174370 00 00 00 00 00 00 00 00 00 00 00
174380 00 00 00 00 00 00 00 00 00 00 00
174390 00 00 00 00 00 00 00 00 00 00 00
174400 00 00 00 00 00 00 00 00 00 00 00
174410 00 00 00 00 00 00 00 00 00 00 00
174420 00 00 00 00 00 00 00 00 00 00 00
174430 00 00 00 00 00 00 00 00 00 00 00
174440 00 00 00 00 00 00 00 00 00 00 00
174450 00 00 00 00 00 00 00 00 00 00 00
174460 00 00 00 00 00 00 00 00 00 00 00
174470 00 00 00 00 00 00 00 00 00 00 00
174480 00 00 00 00 00 00 00 00 00 00 00
174490 00 00 00 00 00 00 00 00 00 00 00
174500 00 00 00 00 00 00 00 00 00 00 00

```

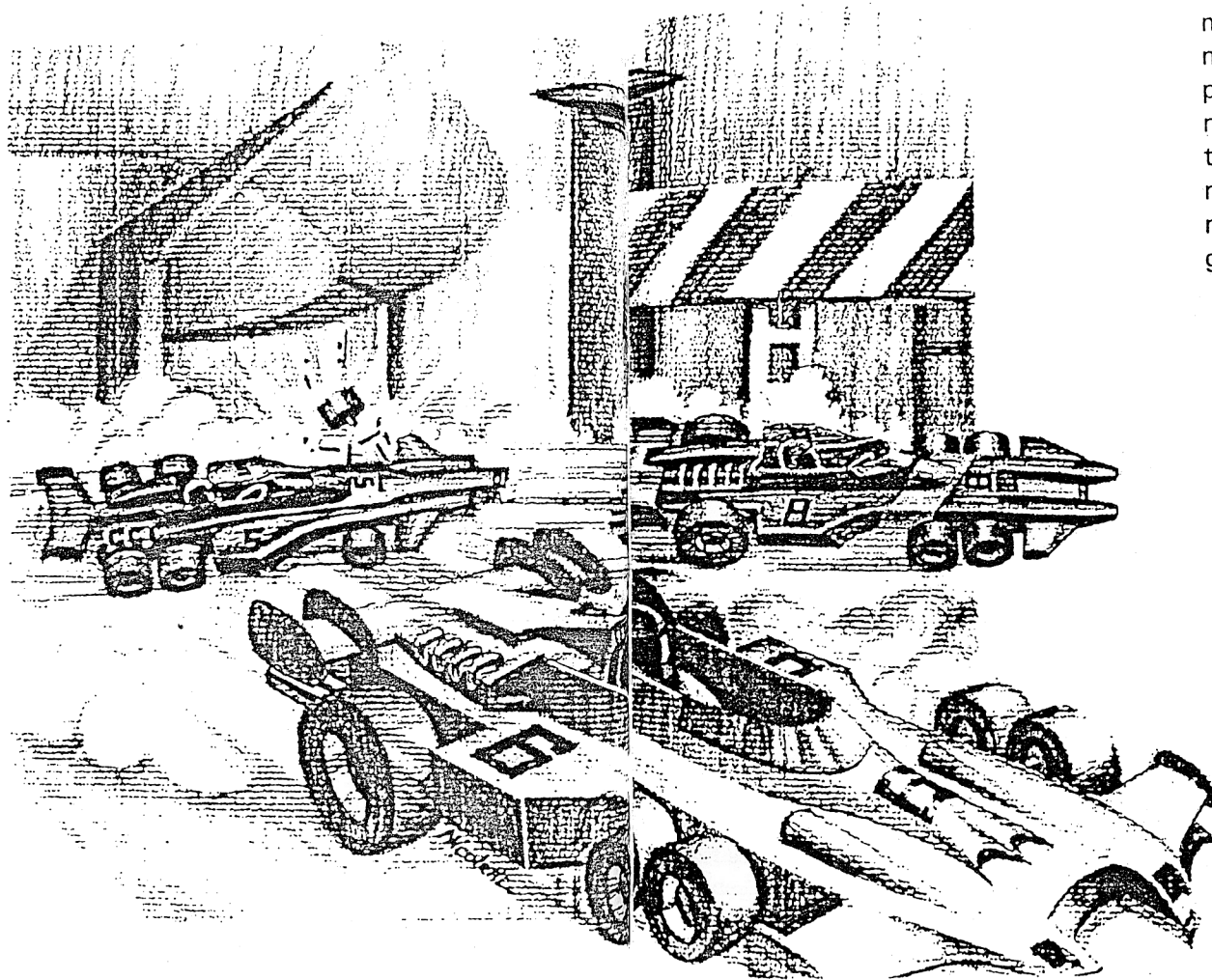
RESUMO

- Foi abordado no presente capítulo o seguinte:
- * Rotinas para contagem crescente e decrescente na tela.
 - * Efeito de "impacto" tipo claro/escuro.
 - * Uso da rotina da ROM KScan no caso de mais de uma tecla ser pressionada ao mesmo tempo.
 - * Movimentação de múltiplos símbolos na tela.

SUGESTÕES

- 1 – À semelhança dos programs anteriores alterar o Basic deste, e incluir dizeres do tipo "Novo Jogo? (S/N)" na parcela em linguagem de máquina. Criar também telas com instruções.
- 2 – Alterar o programa em linguagem de máquina de forma que o canhão, ao invés de mover-se lateralmente de uma posição de memória ou da tela ao ser comandado, que o faça de meia posição. A fim de melhor entender a sugestão deve-se observar que o canhão é composto do cano (figura código 5) e base (figuras 129 – 128 – 4); ao "mover-se" meio passo à direita ele passaria a ser programado por cano (figura 133) e base (figuras 135 – 128 – 130). Com isso seu movimento seria mais suave e uniforme, porém mais lento.

OPÇÕES



Permitir opções no correr de um jogo é uma maneira de torná-lo mais versátil e interessante. Uma emocionante corrida de automóveis é utilizada para mostrar ao leitor como predispor várias opções num mesmo jogo. Aqui também se discutem mais detalhes sobre os movimentos e a contagem de tempo.

CAPÍTULO 5 – OPÇÕES

A característica dos jogos que vai ser enfocada neste capítulo é o oferecimento de escolha ao jogador frente a diferentes opções. A existência destas alternativas é um fator que torna o jogo mais atraente e sem dúvida, representa uma sofisticação do programa.

É possível distinguir três formas de concepção das alternativas:

- * A primeira em que se mantém a tela, ou apresentação básica, e se oferece opção quanto ao programa em si, o qual é modificado de forma a alterar o desempenho em termos de velocidade ou de dificuldade. Um exemplo clássico seria o jogo de xadrês, que pode ser praticado em diferentes níveis, desde o de principiante até os mais avançados.
- * Outra forma seria a de utilizar uma mesma rotina, fazendo variar as telas onde a mesma é praticada.
- * A última consistiria apenas na combinação das duas anteriores.

Em qualquer um dos casos haveria sempre uma alteração em características do programa, condicionada a escolhas do jogador. As formas de implementar as opções são as mais variadas possíveis e dependem da diferença entre elas. Assim para modificar uma velocidade pode ser suficiente alterar um número em um certo endereço, algo similar ao que é feito com a Ramtop quando se deseja limitar a capacidade do computador. Já em outros casos faz-se necessário utilizar longas rotinas distintas condicionadas à escolha feita.

O exemplo que se segue apresenta uma opção de nível de dificuldade. Trata-se de uma corrida que foi intitulada de "Sam's Race". Depois de ler a descrição e fazê-lo rodar no computador, o leitor talvez verifique não ser esta a primeira vez que vê este jogo.

O esboço da tela correspondente é o apresentado na figura 11.

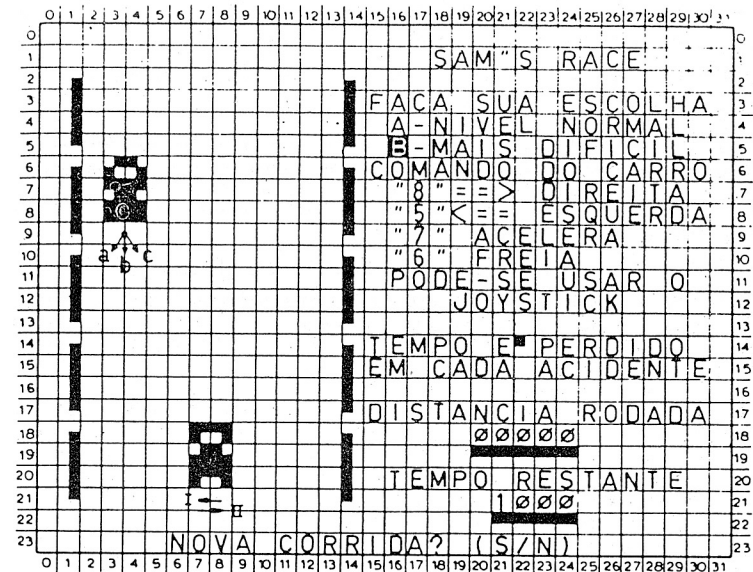


FIGURA 11

Logo de início surge a necessidade de optar: as letras A e B ficam "pisando" na tela, ou seja, cada uma alternadamente muda de fundo preto para branco. Pressionada uma das teclas o jogo inicia-se. Começam então a surgir, em intervalos regulares, figuras representando os carros "inimigos", que se movem de encontro ao carro do jogador.

Os carros inimigos deslocam-se segundo três direções, representadas por a, b e c na figura 11. Encontrando uma lateral, a direção é alterada de a para c ou vice-versa. O ponto de origem e a direção inicial são sorteadas. A diferença entre os dois níveis a escolher, o normal e o mais difícil, é justamente a frequência de ocorrência das direções — no nível normal 75% dos carros deslocam-se segundo b e 25% conforme a ou c; no mais difícil ocorre o oposto.

A fim de evitar a colisão ou acidente, o jogador pode movimentar seu carro nos sentidos I e II. Para dar a sensação de movimento, as laterais "movem-se" também na direção vertical, no mesmo ritmo que os carros inimigos.

A meta do jogo é conseguir "percorrer" a maior distância no tempo disponível. Para um maior controle sobre o jogo, é possível alterar a velocidade dos carros através das teclas 6 (freio) ou 7 (acelerador). Como o comando "direita" ou "esquerda" é feito com o 5 e o 8 respectivamente, pode-se também utilizar o joystick.

Ocorrendo uma colisão, a figura do carro do jogador é modificada para representar um veículo acidentado. Os demais carros são apagados e dá-se uma interrupção de 10 unidades de tempo, após os quais a corrida reinicia-se na velocidade original.

Findo o tempo restante, ou seja, quando este chega a zero, a corrida é definitivamente suspensa e aparece na tela a indicação de "NOVA CORRIDA (S/N)". Com "S" há um reinício e com "N" a memória RAM é apagada.

O Basic associado é o seguinte:

```

0 REM ( 1642 BYTES )
10 SAVE "RACE"
20 RAND USR 16888

```

Depreende-se daí que o programa é praticamente todo escrito em assembly e mais, como a tela está integralmente ocupada (792 bytes), a capacidade do computador deve ser bem superior a 2 Kbytes.

Quando da elaboração deste programa não houve a preocupação de economizar-se memória. E, de fato, existem claramente condições de alterá-lo de forma a que se possa rodá-lo num computador com somente 2 Kbytes de capacidade. Fica aqui a tarefa para aqueles que dispõem de computadores de menor capacidade: de converter o programa adequadamente para isso. Dois pontos a considerar na conversão são:

* A tela não deve ser totalmente preenchida, e sim tratada na forma peculiar aos computadores com capacidade menor que 3,5 Kbytes. Uma sugestão seria, observando a figura 11, a de eliminar todos os dizeres da tela à direita da pista de corrida, deslocá-la uma coluna à esquerda (da 1 para a zero) e finalmente, manter o controle do tempo restante, a distância rodada e as opções, todos na primeira e/ou segunda linha. Com isto, ao invés de 792 bytes a tela ocuparia algo mais que 300 bytes, economizando quase 490. Adicionalmente a isto, observando o grupo que é apresentado adiante, verifica-se ser desneces-

sária a existência do texto para compor a tela, o que ocupa 374 bytes dos 1642 originalmente reservados.

* Só as duas medidas citadas já reduzem a necessidade de capacidade a um nível aceitável para os 2Kbytes, porém existem outros aspectos a explorar — a eliminação dos vários bytes de reserva que foram mantidos ao longo do programa, a simplificação de algumas rotinas, como a de composição da tela por exemplo. Com isso pode-se inclusive reformular o aspecto da tela de modo a torná-la mais atraente.

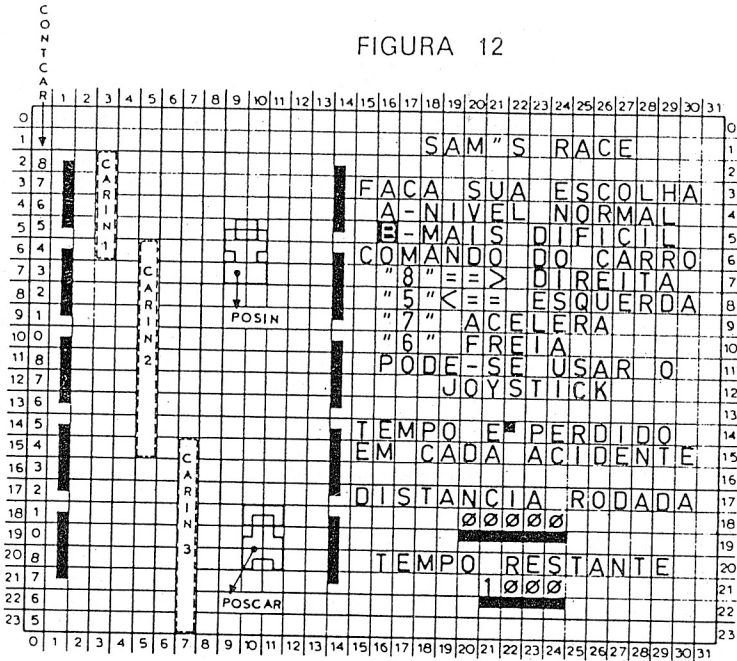
O controle do jogo é baseado nos seguintes Registros Improvisados:

NOME	ENDEREÇO	ABREVIATURA
DISTÂNCIA RODADA	DFile + 619	—
TEMPO TELA	DFile + 718	—
Posição Carro Jogador	407B/C — 16507/8	POSCAR
Contador de Carros	4021 — 16417	CONTCAR
Posição Carro Inimigo 1	403C/D — 16444/5	POSIN1
Direção Carro Inimigo 1	403E — 16446	DIRIN1
Posição Carro Inimigo 2	403F/40 — 16447/8	POSIN2
Direção Carro Inimigo 2	4041 — 16449	DIRIN2
Posição Carro Inimigo 3	4042/3 — 16450/1	POSIN3
Direção Carro Inimigo 3	4044 — 16452	DIRIN3
Velocidade Ajustável	4045/6 — 16453/4	VEL
Tempo Imutável	4047/8 — 16455/6	TEMP
Indicador Posição Jogador	4049 — 16457	INDCAR
Indicador Paridade	404A — 16458	PARI

Para melhor explicar a função dos Registros e a dinâmica do sistema, foi feito um esquema ilustrativo sobre o esboço da tela que é apresentado na figura 12.

Os indicadores de posição seja do carro do jogador como dos inimigos (POSCAR e POSIN) dizem respeito aos pontos de referência indicados na figura.

FIGURA 12



O CONTCAR é registro mais importante para o comando do movimento dos carros inimigos. A cada ciclo ele decresce de uma unidade, a partir de nove até atingir zero. O valor 9 é assumido apenas transitoriamente, no instante em que é feito o sorteio da posição inicial do carro inimigo 1 e, ainda no mesmo ciclo, o CONTCAR é alterado para 8.

Existe uma correspondência direta entre a linha da tela em que se encontra cada carro e o valor de CONTCAR. Por exemplo, se CONTCAR for 7 o carro 1 estará em algum ponto ao longo da linha 3, o carro 2 na linha 12 e o 3 em algum ponto da linha 21. Só podem coexistir três carros inimigos na tela. Quando CONTCAR é igual a 4 o carro 1 passa a ser o 2, e este passa a ser o 3. Aí ficam somente 2 carros na tela até que, com CONTCAR novamente igual 8 ressurgem o carro 1 e o carro 3 começa a "sumir".

O DIRIN dá indicação da direção dos carros inimigos e pode assumir um entre os já conhecidos três valores — 33 para seguir reto, 32 para à esquerda e 34 para à direita.

Os registradores da velocidade ajustável e do tempo imutável terão suas funções facilmente entendidas quando for exposto o programa grupo a grupo.

O INDCAR registra a posição do carro do jogador. Ele varia de zero a dez de forma que, no valor mais baixo, o carro encontra-se na extrema esquerda, e com 10 na direita.

Finalmente o PARI é um indicador usado para permitir que o carro do jogador se mova duas vezes mais rapidamente que os inimigos, seguindo a mesma rotina já usada anteriormente.

Resta ainda apresentar, antes da análise do diagrama de blocos, as principais sub-rotinas utilizadas no programa. A maior parte delas não é expressamente indicada em bloco algum, mas o conhecimento de sua existência e das tarefas que executam serão de boa valia na compreensão geral do esquema. São elas:

- * *Repique* no endereço 17627 (44DB) responsável em testar se o carro inimigo está indo de encontro a Lateral e, em caso positivo, alterar sua direção.
- * *FazCarIn* no endereço 17647 (44EF) cuja função é desenhar o carro inimigo.
- * *ApCar* no endereço 17670 (4506) que se incumbem de "apagar" os carros, tanto os inimigos quanto o do jogador.
- * *TestTrom* no endereço 17689 (4519) que testa se o carro inimigo está indo de encontro ao carro do jogador, ocorrendo uma trombada.
- * *LatSobe* no endereço 17759 (455F) responsável por mover as laterais.
- * *FazCar* no endereço 17785 (4579) desenha o carro do jogador.
- * *Teste* no endereço 17811 (4593) verifica se ao mover-se o carro do jogador não se choca com um inimigo.
- * *Trombada* no endereço 17835 (45AB) bastante importante pois, além de desenhar o carro batido, apaga os carros inimigos da tela, zera os registros correspondentes e mantém o jogo suspenso por 10 unidades de tempo.
- * *TemDim* no endereço 17942 (4616) é a sub-rotina mais explicitada no diagrama de blocos, responsável pela redução do tempo-tela e conseqüente verificação do final, desenhando neste caso a tradicional frase "Nova Corrida? (S/N)".

Posto isto cabe apresentar e analisar a figura 13, onde se encontra o diagrama de blocos que, com tudo o que já foi descrito, não deve causar maiores dificuldades de compreensão.

Seguindo a mesma sistemática de apresentação dos capítulos anteriores tem-se agora os grupos de instruções.


```

110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00
110000000400 00 00 00 00

```

```

40040 00 84 LD A,100(D)
40044 77 LD (HL),A
40048 11 00 LD DE,00(D)
4004C 11 00 XOR A
40050 11 00 SBC HL,DE
40054 11 00 LD A,00
40058 11 00 LD (HL),A
4005C 00 01 DEC HL
40060 77 01 LD A,01
40064 77 01 LD (HL),A
40068 00 02 XOR A
4006C 00 02 SBC HL,DE
40070 00 07 LD A,07
40074 77 07 LD (HL),A
40078 00 07 LD A,07
4007C 00 07 LD (HL),A
40080 00 07 LD A,07
40084 00 07 LD (HL),A
40088 00 07 LD A,07
40092 00 07 LD (HL),A
40096 00 07 LD A,07
400A0 00 07 LD (HL),A
400A4 00 07 LD A,07
400A8 00 07 LD (HL),A
400AC 00 07 LD A,07
400B0 00 07 LD (HL),A
400B4 00 07 LD A,07
400B8 00 07 LD (HL),A
400BC 00 07 LD A,07
400C0 00 07 LD (HL),A
400C4 00 07 LD A,07
400C8 00 07 LD (HL),A
400CC 00 07 LD A,07
400D0 00 07 LD (HL),A
400D4 00 07 LD A,07
400D8 00 07 LD (HL),A
400DC 00 07 LD A,07
400E0 00 07 LD (HL),A
400E4 00 07 LD A,07
400E8 00 07 LD (HL),A
400EC 00 07 LD A,07
400F0 00 07 LD (HL),A
400F4 00 07 LD A,07

```

Grupo 2 – 16888 a 17012 (41F8 a 4274 em hexa)

```

41F0 00 2A 0A CALL CFS
41F4 00 40 LD HL,(D-FILE)
41F8 11 00 00 LD DE,40(D)
4200 10 10 00 ADD HL,DE
4204 00 00 10 LD A,00(D)
4208 00 00 00 EX DE,HL
420C 00 00 40 LD HL,4000
4210 00 00 11 00 LD BC,17(D)
4214 00 00 00 LD IR
4218 00 00 00 EX DE,HL
421C 00 00 10 00 LD BC,16(D)
4220 00 00 00 ADD HL,BC
4224 00 00 00 EX DE,HL
4228 00 00 00 DEC A
422C 00 00 00 JNZ,4200
4230 00 00 40 LD HL,(D-FILE)
4234 00 00 44 00 LD DE,00(D)
4238 00 00 10 00 ADD HL,DE
423C 00 00 05 00 LD A,05
4240 00 00 11 01 00 LD DE,0021
4244 00 00 00 13 00 LD B,13
4248 00 00 00 00 LD (HL),A
424C 00 00 10 00 ADD HL,DE
4250 00 00 00 00 DJNZ,4224
4254 00 00 00 05 00 LD B,05
4258 00 00 11 04 00 LD DE,132(D)
425C 00 00 00 02 00 LD A,02
4260 00 00 00 00 AND A
4264 00 00 00 00 LD (HL),A
4268 00 00 00 00 SBC HL,DE
426C 00 00 00 00 DJNZ,422F
4270 00 00 11 02 00 LD DE,633(D)
4274 00 00 00 00 ADD HL,DE
4278 00 00 00 00 LD (POSCAR),HL
427C 00 00 00 07 00 LD A,07
4280 00 00 00 00 LD (HL),A
4284 00 00 00 00 INC HL

```

Com este compõe-se a tela, incluindo o carro do jogador, e se estabelece o valor inicial do INDCAR que é 5. Para finalizar é feita uma chamada ao endereço 18049 (4681 em hexa) e aparecem dois bytes de reserva.

```

18000000 00 2A 00 00 0A 00 40 11 00 00
18000000 00 00 00 10 00 10 00 00 00 00
18000000 00 00 00 40 00 11 00 00 00 00
18000000 00 00 00 00 40 10 00 00 00 00
18000000 00 00 70 00 00 00 40 11 44 477
18000000 00 00 10 00 00 00 11 01 00 070
18000000 00 00 10 00 00 00 10 10 00 00 00
18000000 00 00 00 11 00 00 00 04 00 00 00
18000000 00 00 00 00 10 00 00 10 00 00 00
18000000 00 00 00 10 00 00 70 40 00 07 017
18000000 00 00 00 00 00 00 04 07 11 01 017
18000000 00 00 00 00 00 00 00 00 07 00 00

```

```

18007 00 00 01 77 AF BC 52 04 7
18007 00 00 77 77 00 04 77 00 00
18007 00 00 07 3E 07 07 00 00 00
18007 00 00 00 1E 04 07 00 00 00
17000 00 00 10 1E 04 07 00 00 00
17007 40 00 01 4E 00 00 40 00 00

```

Grupo 3 – 18031 a 18156 (466F a 46EC em hexa)

```

4666F 7E LD A, (HL)
46670 00 80 ADD A, 80
46672 78 00 PUSH AF
46673 77 LD (HL), A
46674 01 00 08 LD BC, 0800
46677 0B DEC BC
46678 78 LD A, B
46679 01 OR C
4667A 78 7E JR NZ, 46677
4667C 71 00 POP AF
4667D 06 80 SUB 80
4667F 77 LD (HL), A
46680 00 RET
46681 00 XOR A
46682 01 40 LD (CONTCAR), A
46683 00 00 LD B, 00
46684 01 40 LD HL, POSIN1
46685 77 LD (HL), A
46686 03 INC HL
46687 10 7C DJNZ 4668A
46688 00 00 LD HL, (D-FILE)
46689 11 00 40 LD DE, 149(D)
4668A 10 00 ADD HL, DE
4668B 11 01 00 LD DE, 33(D)
4668C 00 0F 4E CALL 4666F
4668D 10 ADD HL, DE
4668E 00 0F 4E CALL 4666F
4668F 00 0B 02 CALL KSCAN
46690 7C LD A, H
46691 05 AND L
46692 11 40 LD DE, 42AD
46693 01 00 00 LD BC, 13(D)
46694 7E 7D CP FD
46695 20 05 JR Z, 466B4
46696 7E 07 CP 5F
46697 20 00 JR Z, 466BE
46698 18 0A JR 4668E
46699 21 03 46 LD HL, 46D3
4669A 00 00 LDIR
4669B 11 00 00 LD DE, 0000

```

```

466BC 18 08 JR 46C5
466BE 21 00 46 LD HL, 46E0
466C1 0D 00 LDIR
466C3 11 21 00 LD DE, 33(D)
466C5 2A 0C 40 LD HL, (D-FILE)
466C9 01 95 00 LD BC, 149(D)
466CC 08 ADD HL, BC
466CD 19 ADD HL, DE
466CE 7E LD A, (HL)
466CF 06 80 ADD A, 80
466D1 77 LD (HL), A
466D2 09 RET

```

DO ENDEREÇO 46D3
 AO ENDEREÇO 46EC
 ESTÃO OS DADOS
 PARA ALTERAR
 O PROGRAMA ORIGINAL

Cuidado! Contrariando a forma que vinha sendo utilizada na apresentação, há aqui um pulo na seqüência dos endereços. Para clareza da explicação achou-se por bem seguir este caminho. Este grupo deve ser o último a ser digitado. Ele, em conjunto com os dois anteriores, vai constituir o bloco 1 do diagrama da figura 13.

Percebe-se claramente que o programa, após sua concepção original, sofreu uma alteração onde, entre outras coisas, foi introduzida a rotina de opção. Esta é uma forma de utilizar os bytes de reserva, quando seu número não é suficiente para conter os dados a crescer, qual seja a de fazer um CALL para o fim do programa onde existe normalmente espaço em abundância. Para aqueles que, por limitação de capacidade, irão reescrever o programa, eis a oportunidade de economizar alguns bytes, bastando colocar este grupo em seqüência com o anterior, formando um só bloco, sem necessidade do CALL.

O grupo se inicia por uma sub-rotina, entre os endereços 18031 a 18048 (466F a 4680 em hexa), responsável pela inversão do "fundo", usando para isso a técnica de somar 128 (80 em hexa) já conhecida. Em complementação existe uma rotina de consumo de tempo correspondente ao intervalo em

que a inversão persiste, para imediatamente após ocorrer a reinversão.

O restante do grupo, a partir do endereço 18049 (4681 em hexa), é exatamente o chamado pelo CALL. Em primeiro lugar o CONTCAR e os demais registradores são zerados. Isto se faz necessário para evitar problemas na situação de reinício do jogo, após o término de uma corrida. Então se dá o "pisca" das letras A e B com o uso da sub-rotina descrita anteriormente, e o teclado é pesquisado através do mais conhecido KScan. Se A for a tecla pressionada, uma série de 13 bytes a partir do endereço 17069 (42AD em hexa) serão substituídos pelos 13 existentes a partir do 18131 (46D3 em hexa) e se B for a tecla, a troca será feita pelos bytes do endereço 18144 (46E0 em hexa). Por fim a letra correspondente a tecla pressionada se manterá com o "fundo" invertido. A forma como isto está sendo feito é interessante e merece alguma atenção.

```

180031 7E C6 80 F5 77 01 00 817
180038 08 0B 78 B1 00 F1 040
180045 06 00 77 00 00 21 000
180050 40 06 00 01 00 40 77 000
180059 00 10 70 00 00 40 11 400
180066 00 00 10 11 00 00 00 400
180070 00 40 10 00 40 00 70 7
180080 00 00 70 00 11 00 42 734
180087 00 00 00 00 00 00 00 007
180094 7E 97 00 00 00 10 00 21 070
181001 00 40 00 00 11 00 00 711
181008 10 00 00 01 00 40 00 770
181110 11 00 00 00 00 40 01 100
181120 00 00 00 10 7E C6 80 00 000
181130 77 00 10 7E 07 00 00 00400
181136 0E 01 10 00 00 00 00 10000
181140 00 01 00 00 00 00 00 10000
181143 00 04 7E 07 00 00 00 00007
181150 01 10 00 C6 47 00 04 00000

```

Grupo 4 – 17013 a 17024 (4275 a 4280 em hexa)

```

4275 01 00 02 LD HL,0200
4276 02 45 40 LD (VEL),HL
427B 01 00 03 LD HL,0300
427E 02 47 40 LD (TEMP),HL

```

Nada além do prescrito pelo bloco 2. Sem comentários.

```

17013 01 00 02 22 45 40 01 035
17020 00 03 02 47 40 177

```

Grupo 5 – 17025 a 17090 (4281 a 42C2 em hexa)

```

4281 0A 21 40 LD A,(CONTCAR)
4284 07 00 AND A
4285 00 30 JR NZ,42C3
4287 0E 00 LD A,09
4288 00 01 40 LD (CONTCAR),A
428C 00 40 40 LD A,(INDCAR)
428F 06 45 ADD A,59(D)
4291 00 00 40 LD HL,(D-FILE)
4294 05 00 ADD A,L
4295 07 00 LD L,A
4296 0E 00 LD A,00
4298 00 00 ADC A,H
4299 07 00 LD H,A
429A 00 30 40 LD (POSINI),HL
429D 00 34 40 LD A,(QUADROS)
429E 00 40 40 40 LD BC,(INDCAR)
42A4 01 00 ADD A,C
42A5 00 00 NOP
42A6 00 00 NOP
42A7 00 00 NOP
42A8 00 00 NOP
42A9 00 00 NOP
42AB 00 07 AND 07
42AC 00 04 JR NZ,42B2
42AD 07 07 CP 07
42B0 00 04 JR C,42B6
42B2 0E 01 LD A,33(D)
42B4 10 00 JR 42C0
42B6 00 47 BIT 0,A
42B8 00 04 JR NZ,42BE
42BA 0E 00 LD A,34(D)
42BC 10 00 JR 42C0
42BE 0E 00 LD A,22
42C0 02 0E 40 LD (DIRINI),A

```

É neste grupo que é feita a alteração comandada pela opção descrita no grupo 3. Há uma correspondência entre este grupo e os blocos 3 e 4 da figura 13.

De início verifica-se se o CONTCAR é zero. Não sendo, há um salto, e na hipótese contrária, CONTCAR é feito igual a 9 e o sorteio da posição

inicial do carro inimigo 1 (POSIN1) e a respectiva direção (DIRIN1) é feito. Só que este não é um sorteio "honesto". Verificou-se que o jogo fica mais atraente se o jogador for forçado a sempre mover seu carro, fugindo do acidente. Por isso POSIN é determinado em função do INDCAR, ou seja, da posição do carro do jogador.

Para o estabelecimento do DIRIN é que são feitas as alterações, conforme opção descritas no grupo 3 — o endereço 42AC está aí envolvido. O acumulador A é feito igual ao Quadros, e a ele é somado o conteúdo de INDCAR. Após o AND 07, com o qual A terá um valor de 0 a 7 — isto é, 8 possibilidades — chega-se ao 42AC. Se a opção for pelo nível normal, seis dos oito valores possíveis de acumulador conduzirão o conteúdo de DIRIN a ser 33 (reto), um conduzirá para 34 (direita) e o outro restante para 32 (esquerda) ou seja, como dito, seis em oito representa 75% de possibilidade do carro seguir reto. Já se a opção for pelo mais difícil, faz-se a inversão das porcentagens.

Os cinco bytes com conteúdo zero que aparecem imediatamente antes do AND 07 são de reserva, e podem portanto ser eliminados. A existência destes no meio de uma rotina é mais um indício de programa modificado.

```

170005 0A 01 40 A7 00 00 0E 470
170006 00 00 01 40 00 40 40 001
170009 00 40 00 00 00 40 07 000
17000B 00 00 00 00 00 00 40 000
17000C 00 00 00 00 00 00 40 000
17000D 00 00 00 00 00 00 40 000
17000E 01 00 00 00 00 00 00 000
17000F 07 00 04 7E 07 00 04 070
170010 0E 01 10 0A 0B 47 00 00 440
170011 04 0E 00 10 00 0E 00 000
170012 00 0E 40 00 00 00 170

```

Grupo 6 — 17091 a 17220 (42C3 a 4344 em hexa)

```

420C3 0A 21 40 LD A, (CONTCAR)
420C6 00 DEC A
420C7 32 21 40 LD (CONTCAR), A
420CA 7E 04 CP, 04
420CC 70 77 JR C, 4345
420CE 21 3E 40 LD HL, DIRIN1
420D1 10 00 LD 0, 00
420D3 5E 00 LD E, (HL)
420D4 21 3C 40 LD HL, (POSIN1)
420D7 7E 00 CP 00
420D9 00 00 JR C, 42E6

```

```

420DB 00 NOP
420DD 00 LD (HL), 00
420DE 00 LD (POSIN1), HL
420E1 00 INC HL
420E2 00 LD (HL), 00
420E4 00 JR 4345
420E6 07 CP 07
420E8 00 JR C, 430D
420EA 00 LD (HL), 00
420EC 00 INC HL
420ED 00 LD (HL), 00
420F7 0D 0E 44 CALL REPIQUE
420F8 00 LD (POSIN1), HL
420FA 07 AND A
420FB 00 JR Z, 42FB
420FC 00 LD (DIRIN1), A
420FE 00 LD (HL), 00
420FF 00 INC HL
42100 00 LD (HL), 00
42102 11 21 00 LD DE, 33(D)
42103 07 AND A
42104 0D 5E SBC HL, DE
42106 00 LD (HL), 07
42108 0D DEC HL
42109 00 LD (HL), 84
4210B 00 JR 4345
4210D 00 CP 00
4210E 00 JR C, 4333
42111 00 LD BC, 33(D)
42114 00 LD (HL), 00
42116 07 AND A
42117 0D 4E SBC HL, BC
42119 00 LD (HL), 00
4211B 00 INC HL
4211D 00 LD (HL), 00
4211F 00 ADD HL, BC
42121 00 LD (HL), 00
42124 00 DEC HL
42126 0D 0E 44 CALL REPIQUE
42128 0D 0E 44 CALL FAZCARIND
4212B 00 LD (POSIN1), HL
4212E 07 AND A
42130 00 JR Z, 4345
42132 00 LD (DIRIN1), A
42134 10 10 JR 4345
42136 0D 0E 45 CALL APCAR
42138 0D 0E 44 CALL REPIQUE
4213A 0D 0E 44 CALL FAZCARIN
4213C 00 LD (POSIN1), HL
4213E 07 AND A
42140 00 JR Z, 4345
42142 00 LD DIRIN1, A

```



```

17300 17 44 44 16 00 00 00 00
17301 17 44 44 16 00 00 00 00
17302 17 44 44 16 00 00 00 00
17303 17 44 44 16 00 00 00 00
17304 17 44 44 16 00 00 00 00
17305 17 44 44 16 00 00 00 00
17306 17 44 44 16 00 00 00 00
17307 17 44 44 16 00 00 00 00
17308 17 44 44 16 00 00 00 00
17309 17 44 44 16 00 00 00 00
17310 17 44 44 16 00 00 00 00

```

```

4300 20 00 00 00 00 00 00 00
4301 20 00 00 00 00 00 00 00
4302 20 00 00 00 00 00 00 00
4303 20 00 00 00 00 00 00 00
4304 20 00 00 00 00 00 00 00
4305 20 00 00 00 00 00 00 00
4306 20 00 00 00 00 00 00 00
4307 20 00 00 00 00 00 00 00
4308 20 00 00 00 00 00 00 00
4309 20 00 00 00 00 00 00 00
4310 20 00 00 00 00 00 00 00
4311 20 00 00 00 00 00 00 00
4312 20 00 00 00 00 00 00 00
4313 20 00 00 00 00 00 00 00
4314 20 00 00 00 00 00 00 00
4315 20 00 00 00 00 00 00 00
4316 20 00 00 00 00 00 00 00
4317 20 00 00 00 00 00 00 00
4318 20 00 00 00 00 00 00 00
4319 20 00 00 00 00 00 00 00
4320 20 00 00 00 00 00 00 00
4321 20 00 00 00 00 00 00 00
4322 20 00 00 00 00 00 00 00
4323 20 00 00 00 00 00 00 00
4324 20 00 00 00 00 00 00 00
4325 20 00 00 00 00 00 00 00
4326 20 00 00 00 00 00 00 00
4327 20 00 00 00 00 00 00 00
4328 20 00 00 00 00 00 00 00
4329 20 00 00 00 00 00 00 00
4330 20 00 00 00 00 00 00 00
4331 20 00 00 00 00 00 00 00
4332 20 00 00 00 00 00 00 00
4333 20 00 00 00 00 00 00 00
4334 20 00 00 00 00 00 00 00
4335 20 00 00 00 00 00 00 00
4336 20 00 00 00 00 00 00 00
4337 20 00 00 00 00 00 00 00
4338 20 00 00 00 00 00 00 00
4339 20 00 00 00 00 00 00 00
4340 20 00 00 00 00 00 00 00
4341 20 00 00 00 00 00 00 00
4342 20 00 00 00 00 00 00 00
4343 20 00 00 00 00 00 00 00
4344 20 00 00 00 00 00 00 00
4345 20 00 00 00 00 00 00 00
4346 20 00 00 00 00 00 00 00
4347 20 00 00 00 00 00 00 00
4348 20 00 00 00 00 00 00 00
4349 20 00 00 00 00 00 00 00
4350 20 00 00 00 00 00 00 00
4351 20 00 00 00 00 00 00 00
4352 20 00 00 00 00 00 00 00
4353 20 00 00 00 00 00 00 00
4354 20 00 00 00 00 00 00 00
4355 20 00 00 00 00 00 00 00
4356 20 00 00 00 00 00 00 00
4357 20 00 00 00 00 00 00 00
4358 20 00 00 00 00 00 00 00
4359 20 00 00 00 00 00 00 00
4360 20 00 00 00 00 00 00 00
4361 20 00 00 00 00 00 00 00
4362 20 00 00 00 00 00 00 00
4363 20 00 00 00 00 00 00 00
4364 20 00 00 00 00 00 00 00
4365 20 00 00 00 00 00 00 00
4366 20 00 00 00 00 00 00 00
4367 20 00 00 00 00 00 00 00
4368 20 00 00 00 00 00 00 00
4369 20 00 00 00 00 00 00 00
4370 20 00 00 00 00 00 00 00
4371 20 00 00 00 00 00 00 00
4372 20 00 00 00 00 00 00 00
4373 20 00 00 00 00 00 00 00
4374 20 00 00 00 00 00 00 00
4375 20 00 00 00 00 00 00 00
4376 20 00 00 00 00 00 00 00
4377 20 00 00 00 00 00 00 00
4378 20 00 00 00 00 00 00 00
4379 20 00 00 00 00 00 00 00
4380 20 00 00 00 00 00 00 00
4381 20 00 00 00 00 00 00 00
4382 20 00 00 00 00 00 00 00
4383 20 00 00 00 00 00 00 00
4384 20 00 00 00 00 00 00 00
4385 20 00 00 00 00 00 00 00
4386 20 00 00 00 00 00 00 00
4387 20 00 00 00 00 00 00 00
4388 20 00 00 00 00 00 00 00
4389 20 00 00 00 00 00 00 00
4390 20 00 00 00 00 00 00 00
4391 20 00 00 00 00 00 00 00
4392 20 00 00 00 00 00 00 00
4393 20 00 00 00 00 00 00 00
4394 20 00 00 00 00 00 00 00
4395 20 00 00 00 00 00 00 00
4396 20 00 00 00 00 00 00 00
4397 20 00 00 00 00 00 00 00
4398 20 00 00 00 00 00 00 00
4399 20 00 00 00 00 00 00 00
4400 20 00 00 00 00 00 00 00
4401 20 00 00 00 00 00 00 00
4402 20 00 00 00 00 00 00 00
4403 20 00 00 00 00 00 00 00
4404 20 00 00 00 00 00 00 00
4405 20 00 00 00 00 00 00 00
4406 20 00 00 00 00 00 00 00
4407 20 00 00 00 00 00 00 00
4408 20 00 00 00 00 00 00 00
4409 20 00 00 00 00 00 00 00
4410 20 00 00 00 00 00 00 00
4411 20 00 00 00 00 00 00 00
4412 20 00 00 00 00 00 00 00
4413 20 00 00 00 00 00 00 00
4414 20 00 00 00 00 00 00 00
4415 20 00 00 00 00 00 00 00
4416 20 00 00 00 00 00 00 00
4417 20 00 00 00 00 00 00 00
4418 20 00 00 00 00 00 00 00
4419 20 00 00 00 00 00 00 00
4420 20 00 00 00 00 00 00 00
4421 20 00 00 00 00 00 00 00
4422 20 00 00 00 00 00 00 00
4423 20 00 00 00 00 00 00 00
4424 20 00 00 00 00 00 00 00
4425 20 00 00 00 00 00 00 00

```

Grupo 9 – 17303 a 17448 (4397 a 4428 em hexa)

```

4300 17 44 44 16 00 00 00 00 AND A
4301 17 44 44 16 00 00 00 00 JR Z,43DF
4302 17 44 44 16 00 00 00 00 LD A,(CONTCAR)
4303 17 44 44 16 00 00 00 00 CP 07
4304 17 44 44 16 00 00 00 00 JR NZ,43B8
4305 17 44 44 16 00 00 00 00 CALL APCAR
4306 17 44 44 16 00 00 00 00 CALL TESTROM
4307 17 44 44 16 00 00 00 00 CALL REPIQUE
4308 17 44 44 16 00 00 00 00 CALL FAZCARIN
4309 17 44 44 16 00 00 00 00 LD (POSIN3),HL
4310 17 44 44 16 00 00 00 00 AND A
4311 17 44 44 16 00 00 00 00 JR Z,43DF
4312 17 44 44 16 00 00 00 00 LD (DIRIN3),A
4313 17 44 44 16 00 00 00 00 JR 43DF
4314 17 44 44 16 00 00 00 00 CP 05
4315 17 44 44 16 00 00 00 00 JR NZ,43E1
4316 17 44 44 16 00 00 00 00 CALL APCAR
4317 17 44 44 16 00 00 00 00 AND A
4318 17 44 44 16 00 00 00 00 SBC HL,BC
4319 17 44 44 16 00 00 00 00 CALL TESTROM
4320 17 44 44 16 00 00 00 00 CALL REPIQUE
4321 17 44 44 16 00 00 00 00 LD (HL),84
4322 17 44 44 16 00 00 00 00 INC HL
4323 17 44 44 16 00 00 00 00 LD (HL),07
4324 17 44 44 16 00 00 00 00 AND A
4325 17 44 44 16 00 00 00 00 SBC HL,BC
4326 17 44 44 16 00 00 00 00 LD (HL),86
4327 17 44 44 16 00 00 00 00 DEC HL
4328 17 44 44 16 00 00 00 00 LD (HL),06
4329 17 44 44 16 00 00 00 00 ADD HL,BC
4330 17 44 44 16 00 00 00 00 LD (POSIN3),HL
4331 17 44 44 16 00 00 00 00 AND A
4332 17 44 44 16 00 00 00 00 JR Z,43DF
4333 17 44 44 16 00 00 00 00 LD (DIRIN3),A
4334 17 44 44 16 00 00 00 00 JR 4429
4335 17 44 44 16 00 00 00 00 CP 05
4336 17 44 44 16 00 00 00 00 JR NZ,4405
4337 17 44 44 16 00 00 00 00 LD BC,33(D)
4338 17 44 44 16 00 00 00 00 LD (HL),00

```

Aqui começam as trombadas. Elas se dão sempre entre o carro inimigo 3 e o do jogador. Os blocos ligados a este grupo são os 14, 15 e 16. As duas primeiras instruções constituem o 14 e os dois outros blocos vão sendo repetidos a cada valor de CONTCAR. De uma certa forma a rotina aqui envolvida é a inversa da do grupo 6, já que o carro vai gradualmente desaparecendo.

O teste de trombada é feito a cada vez com o uso da sub-rotina Testrom.

```

17303 A7 28 45 3A 21 40 FE 666
17310 07 28 17 CD 06 45 CD 547
17317 19 45 CD DB 44 CD EF 1030
17324 44 22 42 40 A7 28 2C 463
17331 00 44 40 18 07 FE 08 505
17338 20 25 CD 08 45 A7 ED 753
17345 40 CD 19 45 CD DB 44 857
17352 06 84 20 06 07 A7 ED 866
17359 40 06 06 0B 06 06 09 366

```



```

4476 20 1E JR Z,4495
4477 30A 4A 40 LD A,(PARI)
4478 30C 4A 40 INC A
4479 30E 4A 40 LD (PARI),A
447E 0B 47 BIT 0,A
4480 00 0B JR Z,444D
4481 00 01 40 JP 4281
4482 0A 45 40 LD HL,(VEL)
4483 11 10 00 LD DE,0010
4484 07 AND A
4485 00 00 SBC HL,DE
4486 00 07 JR Z,4477
4487 22 45 40 LD (VEL),HL
4488 10 00 JR 4477
4489 2D 45 40 LD HL,(VEL)
4490 11 00 00 LD DE,0020
4491 10 ADD HL,DE
4492 70 LD A,H
4493 00 CP 00
4494 20 00 JR Z,4477
4495 22 45 40 LD (VEL),HL
4496 10 D1 JR 4477
4497 3D 40 40 LD A,(INDCAR)
4498 07 AND A
4499 20 0B JR Z,4477
44A0 2D 7B 40 LD HL,(POSCAR)
44A1 00 06 40 CALL APCAR
44A2 2B DEC HL
44A3 00 03 45 CALL TESTE
44A4 3D DEC A
44A5 3D 40 40 LD (INDCAR),A
44A6 00 70 45 CALL FAZCAR
44A7 10 00 JR 4477
44A8 3D 40 40 LD A,(INDCAR)
44A9 00 0A OP 0A
44AA 20 01 JR Z,4477
44AB 2D 7B 40 LD HL,(POSCAR)
44AC 00 06 45 CALL APCAR
44AD 00 INC HL
44AE 00 INC HL
44AF 00 03 45 CALL TESTE
44B0 30 INC A
44B1 3D 40 40 LD (INDCAR),A
44B2 2B DEC HL
44B3 00 70 45 CALL FAZCAR
44B4 10 00 JR 4477

```

Este grupo está ligado a todo um conjunto de blocos: 19, 20, 21, 22, 23, 24 e 25. Assim é conveniente analisá-lo por sub-grupos.

Do 4462 a 4476 é utilizado o frequentemente citado KScan e ao sub-grupo correspondem blocos 19, 21 e 23.

O subgrupo seguinte é do endereço 4477 ao 4484 e faz exatamente o prescrito pelo bloco 20, ou seja, incrementa o PARI e dá os saltos em função do bit zero deste registrador.

Prosseguindo vem o 4485 a 44A5 que acelera ou breca os carros, isto é, é o associado ao bloco 22. Como observação digna de nota é o fato de só haver limitação na redução da velocidade e não no seu aumento.

Por fim vem o último sub-grupo, do 44A6 a 44DA ligado aos blocos 24 e 25. Nele são usadas várias sub-rotinas que o simplificam, a ApCar e a FazCar que apaga o carro de uma posição e o desenha na seguinte, e a Teste utilizada para detectar a ocorrência de trombada.

```

175006 00 0B 00 70 05 7E 07 1150
175010 00 0B 7E 07 00 00 7E 00
175020 00 07 00 10 7E 07 00 1E 00
175027 3A 4A 40 30 3E 4A 40 44
175034 0B 47 00 0B 00 01 40 90
175041 0A 45 40 11 10 00 07 37
175048 00 00 00 07 00 45 40 75
175050 10 00 00 40 40 11 00 47
175059 00 10 70 7E 00 00 00 00
175060 00 40 40 10 01 00 40 80
175076 40 07 00 0B 00 7B 40 70
175080 00 06 40 00 00 00 40 74
175090 00 30 40 40 00 70 40 64
175097 10 00 00 40 40 7E 00 60
1750A4 00 01 00 7D 40 00 00 60
1750B1 40 00 00 00 00 00 40 60
1750B8 00 40 40 00 00 70 40 60
1750C5 10 00 00 00 00 00 70 10

```

Grupo 13 – 17627 a 17646 (44DB a 44EE em hexa)

```

44DB 10 ADD HL,DE
44DC 20 INC HL
44DD 7E LD A,(HL)
44DE 2B DEC HL
44DF 07 AND A
44E0 20 05 JR Z,44E7
44E2 2B DEC HL
44E3 2B DEC HL
44E4 3E 20 LD A,32(D)
44E5 09 RET
44E7 7E LD A,(HL)

```



```

44E8 A7 AND A
44E9 00 RET Z
44EA 22 LD A,34 (D)
44EB 00 INC HL
44EC 03 INC HL
44ED 00 RET

```

Éis a sub-rotina Repique, primeira de uma série não associada diretamente a bloco algum da figura 13. Ela apenas verifica se o carro inimigo vai de encontro a lateral e, em caso positivo o Acumulador deixa de ser zero e passa a conter a direção a seguir.

```

17627 10 03 7F 28 A7 20 05 441
17634 20 00 0F 00 09 70 05 474
17641 00 0E 00 00 03 00 05 567

```

Grupo 14 — 17647 a 17669 (44EF a 4505 em hexa)

```

44EF 36 00 LD (HL),00
44F1 00 INC HL
44F2 36 00 LD (HL),00
44F4 A7 AND A
44F5 ED 40 SBC HL,BC
44F7 36 07 LD (HL),07
44F9 A7 AND A
44FB ED 40 SBC HL,BC
44FD 36 06 LD (HL),06
44FF 00 DEC HL
4501 36 06 LD (HL),06
4503 00 ADD HL,BC
4505 36 04 LD (HL),04
4507 00 ADD HL,BC
4509 00 RET

```

É a sub-rotina FazCarIn, ou seja, desenha a figura do carro inimigo. Sem comentários.

```

17647 36 00 03 06 00 A7 ED 003
17654 40 00 07 A7 ED 42 06 051
17661 00 00 06 06 09 06 04 432
17668 09 00 00 00 00 00 04 210

```

Grupo 15 — 17670 a 17688 (4506 a 4518 em hexa)

```

4506 01 21 00 LD BC,33 (D)
4508 70 LD (HL),B
450A 70 AND A
450C ED 42 SBC HL,BC
450E 70 LD (HL),B
4510 ED 42 SBC HL,BC
4512 70 LD (HL),B
4514 20 INC HL
4516 70 LD (HL),B
4518 00 ADD HL,BC
451A 70 LD (HL),B
451C 00 ADD HL,BC
451E 70 LD (HL),B
4520 20 DEC HL
4522 00 RET

```

Outra sub-rotina que não merece comentários. É a ApCar que apenas apaga o desenho dos carros.

```

17670 01 21 00 70 A7 ED 42 016
17677 70 ED 42 70 23 70 09 083
17684 70 09 70 2B 09 477

```

Grupo 16 — 17689 a 17758 (4519 a 455E em hexa)

```

4519 20 DEC HL
451B 70 LD A,(HL)
451D 70 CP 84
451F 20 04 JR NZ,4523
4521 20 02 CP 82
4523 20 14 JR NZ,4537
4525 30 44 40 LD A,(DIRIN3)
4527 00 CP 32 (D)
4529 20 00 JR NZ,455A
452B 00 CP 34 (D)
452D 20 00 JR NZ,4537
452F 00 INC HL
4531 00 INC HL
4533 00 INC HL
4535 20 LD A,(HL)
4537 70 AND A
4539 20 JR NZ,455A
453B 10 00 JR 4557
453D 20 INC HL
453F 00 INC HL
4541 00 INC HL
4543 70 LD A,(HL)
4545 07 CP 07
4547 04 JR NZ,4543
4549 01 CP 81

```

```

4541 00 14      JR    NZ,4557
4543 00 44 40    LD    A,(DIRING)
4545 00 00 00    CP    34(D)
4546 00 10      JR    NZ,455A
4548 00 00 00    CP    32(D)
454C 00 00      JR    NZ,4557
454E 00 00      DEC  HL
454F 00 00      DEC  HL
4550 00 00      DEC  HL
4551 00 7E      LD    A,(HL)
4552 00 03      INC  HL
4553 00 17      AND  A
4554 00 04      JR    NZ,455A
4555 00 00 00    RET
4557 00 00      DEC  HL
4558 00 00      DEC  HL
4559 00 00      RET
455A 01 AB 45    LD    HL,TROMBADA
455D 03 00      EX   (SP),HL
455E 00 00      RET

```

Esta é a TesTrom, ou seja, a sub-rotina que verifica, no movimento do carro inimigo 3, se ele vai trombar com o carro do jogador. Não é das mais complicadas, e merece alguma atenção para uma perfeita compreensão.

```

17660 00 7E 7E 04 20 04 7E 00 00
17662 00 00 00 14 3D 44 40 7E 00 00
17700 00 00 00 00 7E 02 00 00 44 00
17710 00 03 03 03 7E A7 00 00 45 7
17717 10 00 00 00 00 00 7E 00 41
17724 07 00 04 7E 01 00 14 43 00
17734 0D 44 40 7E 00 00 10 00 44
17738 7E 00 00 00 00 00 00 45 00
17748 7E 00 00 07 00 04 00 00 00
17752 00 00 01 00 45 03 00 00 45

```

Grupo 17 – 17759 a 17784 (455F a 4578 em hexa)

```

455F 11 21 00    LD    DE,33(D)
4562 06 13      LD    B,19(D)
4564 05 00      PUSH HL
4566 07 00      AND  A
4568 0D 52      SBC  HL,DE
456A 7E 00      LD    A,(HL)
456C 03 00      EX   (SP),HL
456E 77 00      LD    (HL),A
456F 01 00      POP  HL
4570 05 00      PUSH HL
4572 10 F6      DJNZ 4565
4574 00 00      NOP
4576 00 00      NOP

```

```

4571 10 00      ADD  HL,DE
4572 10 00      ADD  HL,DE
4573 10 00      ADD  HL,DE
4574 10 00      ADD  HL,DE
4576 7E 00      LD    A,(HL)
4578 01 00      POP  HL
457A 77 00      LD    (HL),A
457C 00 00      RET

```

Outra sub-rotina bastante simples. Através desta as laterais são movidas. É a que foi chamada LatSobe.

```

17750 11 21 00 06 13 55 A7 47 1
17752 0D 52 7E 03 77 01 00 12 40
17773 10 F6 00 00 10 10 10 33 7
17780 10 7E 01 77 00 00 00

```

Grupo 18 – 17785 a 17810 (4579 a 4592 em hexa)

```

4579 06 07      LD    (HL),07
457B 03 00      INC  HL
457D 06 84      LD    (HL),84
457F 07 00      AND  A
4581 0D 42      SBC  HL,BC
4583 0D 82      LD    (HL),82
4585 07 00      AND  A
4587 0D 42      SBC  HL,BC
4589 06 84      LD    (HL),84
458B 00 00      DEC  HL
458D 06 07      LD    (HL),07
458F 06 00      ADD  HL,BC
4591 06 81      LD    (HL),81
4593 06 00      ADD  HL,BC
4595 02 7B 40    LD    (POSCAR),HL
4597 00 00      RET

```

Esta é a FazCar que desenha o carro do jogador. Também sem comentários.

```

17785 06 07 23 36 84 A7 0D 58 6
17792 42 06 02 A7 0D 42 36 77 4
17798 04 00 36 07 00 36 01 42 6
17805 09 02 7B 40 00 00 00 43 1

```

Grupo 19 – 17811 a 17834 (4593 a 45AA em hexa)

```

4593 F5 00      PUSH AF
4594 7E 00      LD    A,(HL)
4595 A7 00      AND  A

```

```

45006 20 10 JR NZ,45A8
45008 ED 42 SBC HL,BC
4500A 7E LD A,(HL)
4500C A7 AND A
4500E 20 0A JR NZ,45A8
45010 ED 42 SBC HL,BC
45012 7E LD A,(HL)
45014 A7 AND A
45016 00 04 JR NZ,45A8
45018 00 00 ADD HL,BC
4501A 00 00 ADD HL,BC
4501C F1 POP AF
4501E C9 RET
45020 F1 POP AF
45022 E1 POP HL
45024 00 NOP

```

Esta é a sub-rotina Teste utilizada tanto no movimento do carro inimigo 3 como no do jogador para verificar se haverá uma trombada.

```

17811 F5 7E A7 20 10 ED 42 000
17813 7E A7 00 00 ED 42 7E 764
17815 A7 00 04 00 09 F1 C9 663
17817 F1 E1 00 400

```

Grupo 20 — 17835 a 17941 (45AB a 4615 em hexa)

```

45AB 2A 0C 40 LD HL,(D-FILE)
45AD 11 45 00 LD DE,59(D)
45B1 10 ADD HL,DE
45B2 11 15 00 LD DE,20(D)
45B6 01 00 14 LD BC,1400
45BA 3E 0C LD A,12(D)
45BC 71 LD (HL),C
45BE 23 INC HL
45C0 3D DEC A
45C2 A7 AND A
45C4 20 FA JR NZ,45BA
45C6 10 75 ADD HL,DE
45C8 10 F5 DJNZ 45B8
45CA AF XOR A
45CC 30 01 40 LD (CONTCAR),A
45CE 00 09 LD B,09
45D0 01 3C 40 LD HL,POSIN1
45D2 77 LD (HL),A
45D4 23 INC HL
45DE 10 7C DJNZ 45C0
45E0 2A 7B 40 LD HL,(POSCAR)
45E2 3E 85 LD (HL),85
45E4 11 01 00 LD DE,33(D)
45E6 A7 AND A

```

```

45F0 ED 52 SBC HL,DE
45F2 3E 07 LD (HL),07
45F4 A7 AND A
45F6 ED 52 SBC HL,DE
45F8 3E 06 LD (HL),06
45FA 00 05 INC HL
45FC 10 05 LD (HL),05
45FE 10 81 ADD HL,DE
4600 10 81 LD (HL),81
4602 10 85 ADD HL,DE
4604 3E 85 LD (HL),85
4606 06 0A LD B,10(D)
4608 ED 10 14 LD HL,1400
460A 70 DEC HL
460C 7C LD A,H
460E 05 OR L
4610 00 7B JR NZ,45F0
4612 0D 16 46 CALL TEMDIM
4614 10 73 DJNZ 45ED
4616 2A 7B 46 LD HL,(POSCAR)
4618 01 01 00 LD BC,33(D)
461A 0D 7B 45 CALL FAZCAR
461C 03 7B 42 JP 427B

```

Aqui retorna-se ao programa básico. Este grupo está associado aos blocos 26, 27 e 28. De início todos os carros são totalmente apagados da pista. Todos os registradores de posicionamento e direção, com exceção do POSCAR, são então zerados. Baseado no POSCAR o desenho do carro trombado é executado.

O efeito de consumir 10 unidades de tempo é implementado através das chamadas sucessivas à sub-rotina TemDim e, ao final, tem-se o carro do jogador redesenhado com o uso da FazCar.

Os 16 bytes que estão no final da série são de reserva e poderiam ser eliminados.

```

17835 2A 0C 40 11 45 00 10 000
17837 11 15 00 01 00 14 00 000
17839 3E 0C 71 23 3D A7 00 00 000
1783B 00 09 10 75 10 75 00 00 000
1783D 01 3C 40 77 01 00 00 00 000
1783F 0D 7B 45 01 00 14 00 00 000
17841 03 7B 42 00 00 00 00 00 000
17843 00 00 00 00 00 00 00 00 000
17845 00 00 00 00 00 00 00 00 000
17847 00 00 00 00 00 00 00 00 000
17849 00 00 00 00 00 00 00 00 000
1784B 00 00 00 00 00 00 00 00 000
1784D 00 00 00 00 00 00 00 00 000
1784F 00 00 00 00 00 00 00 00 000
17851 00 00 00 00 00 00 00 00 000
17853 00 00 00 00 00 00 00 00 000
17855 00 00 00 00 00 00 00 00 000
17857 00 00 00 00 00 00 00 00 000
17859 00 00 00 00 00 00 00 00 000
1785B 00 00 00 00 00 00 00 00 000
1785D 00 00 00 00 00 00 00 00 000
1785F 00 00 00 00 00 00 00 00 000
17861 00 00 00 00 00 00 00 00 000
17863 00 00 00 00 00 00 00 00 000
17865 00 00 00 00 00 00 00 00 000
17867 00 00 00 00 00 00 00 00 000
17869 00 00 00 00 00 00 00 00 000
1786B 00 00 00 00 00 00 00 00 000
1786D 00 00 00 00 00 00 00 00 000
1786F 00 00 00 00 00 00 00 00 000
17871 00 00 00 00 00 00 00 00 000
17873 00 00 00 00 00 00 00 00 000
17875 00 00 00 00 00 00 00 00 000
17877 00 00 00 00 00 00 00 00 000
17879 00 00 00 00 00 00 00 00 000
1787B 00 00 00 00 00 00 00 00 000
1787D 00 00 00 00 00 00 00 00 000
1787F 00 00 00 00 00 00 00 00 000
17881 00 00 00 00 00 00 00 00 000
17883 00 00 00 00 00 00 00 00 000
17885 00 00 00 00 00 00 00 00 000
17887 00 00 00 00 00 00 00 00 000
17889 00 00 00 00 00 00 00 00 000
1788B 00 00 00 00 00 00 00 00 000
1788D 00 00 00 00 00 00 00 00 000
1788F 00 00 00 00 00 00 00 00 000
17891 00 00 00 00 00 00 00 00 000
17893 00 00 00 00 00 00 00 00 000
17895 00 00 00 00 00 00 00 00 000
17897 00 00 00 00 00 00 00 00 000
17899 00 00 00 00 00 00 00 00 000
1789B 00 00 00 00 00 00 00 00 000
1789D 00 00 00 00 00 00 00 00 000
1789F 00 00 00 00 00 00 00 00 000
178A1 00 00 00 00 00 00 00 00 000
178A3 00 00 00 00 00 00 00 00 000
178A5 00 00 00 00 00 00 00 00 000
178A7 00 00 00 00 00 00 00 00 000
178A9 00 00 00 00 00 00 00 00 000
178AB 00 00 00 00 00 00 00 00 000
178AD 00 00 00 00 00 00 00 00 000
178AF 00 00 00 00 00 00 00 00 000
178B1 00 00 00 00 00 00 00 00 000
178B3 00 00 00 00 00 00 00 00 000
178B5 00 00 00 00 00 00 00 00 000
178B7 00 00 00 00 00 00 00 00 000
178B9 00 00 00 00 00 00 00 00 000
178BB 00 00 00 00 00 00 00 00 000
178BD 00 00 00 00 00 00 00 00 000
178BF 00 00 00 00 00 00 00 00 000
178C1 00 00 00 00 00 00 00 00 000
178C3 00 00 00 00 00 00 00 00 000
178C5 00 00 00 00 00 00 00 00 000
178C7 00 00 00 00 00 00 00 00 000
178C9 00 00 00 00 00 00 00 00 000
178CB 00 00 00 00 00 00 00 00 000
178CD 00 00 00 00 00 00 00 00 000
178CF 00 00 00 00 00 00 00 00 000
178D1 00 00 00 00 00 00 00 00 000
178D3 00 00 00 00 00 00 00 00 000
178D5 00 00 00 00 00 00 00 00 000
178D7 00 00 00 00 00 00 00 00 000
178D9 00 00 00 00 00 00 00 00 000
178DB 00 00 00 00 00 00 00 00 000
178DD 00 00 00 00 00 00 00 00 000
178DF 00 00 00 00 00 00 00 00 000
178E1 00 00 00 00 00 00 00 00 000
178E3 00 00 00 00 00 00 00 00 000
178E5 00 00 00 00 00 00 00 00 000
178E7 00 00 00 00 00 00 00 00 000
178E9 00 00 00 00 00 00 00 00 000
178EB 00 00 00 00 00 00 00 00 000
178ED 00 00 00 00 00 00 00 00 000
178EF 00 00 00 00 00 00 00 00 000
178F1 00 00 00 00 00 00 00 00 000
178F3 00 00 00 00 00 00 00 00 000
178F5 00 00 00 00 00 00 00 00 000
178F7 00 00 00 00 00 00 00 00 000
178F9 00 00 00 00 00 00 00 00 000
178FB 00 00 00 00 00 00 00 00 000
178FD 00 00 00 00 00 00 00 00 000
178FF 00 00 00 00 00 00 00 00 000
17901 00 00 00 00 00 00 00 00 000
17903 00 00 00 00 00 00 00 00 000
17905 00 00 00 00 00 00 00 00 000
17907 00 00 00 00 00 00 00 00 000
17909 00 00 00 00 00 00 00 00 000
1790B 00 00 00 00 00 00 00 00 000
1790D 00 00 00 00 00 00 00 00 000
1790F 00 00 00 00 00 00 00 00 000
17911 00 00 00 00 00 00 00 00 000
17913 00 00 00 00 00 00 00 00 000
17915 00 00 00 00 00 00 00 00 000
17917 00 00 00 00 00 00 00 00 000
17919 00 00 00 00 00 00 00 00 000
1791B 00 00 00 00 00 00 00 00 000
1791D 00 00 00 00 00 00 00 00 000
1791F 00 00 00 00 00 00 00 00 000
17921 00 00 00 00 00 00 00 00 000
17923 00 00 00 00 00 00 00 00 000
17925 00 00 00 00 00 00 00 00 000
17927 00 00 00 00 00 00 00 00 000
17929 00 00 00 00 00 00 00 00 000
1792B 00 00 00 00 00 00 00 00 000
1792D 00 00 00 00 00 00 00 00 000
1792F 00 00 00 00 00 00 00 00 000
17931 00 00 00 00 00 00 00 00 000
17933 00 00 00 00 00 00 00 00 000
17935 00 00 00 00 00 00 00 00 000
17937 00 00 00 00 00 00 00 00 000
17939 00 00 00 00 00 00 00 00 000
1793B 00 00 00 00 00 00 00 00 000
1793D 00 00 00 00 00 00 00 00 000
1793F 00 00 00 00 00 00 00 00 000
17941 00 00 00 00 00 00 00 00 000

```

Grupo 21 – 17942 a 18030 (4616 a 466E em hexa)

```

4616 05          PUSH BC
4617 2A 00 40    LD HL,(D-FILE)
4618 11 0E 02    LD DE,718(D)
4619 19          ADD HL,DE
461E 7E          LD A,(HL)
461F D7          AND A
4620 29 2A      JR NZ,464C
4621 08 04      LD B,04
4622 23          INC HL
4623 38 1C      LD (HL),1C
4624 10 7B      DJNZ 4624
4625 0D 01      POP BC
4626 0D 01      POP HL
4627 2A 00 40    LD HL,(D-FILE)
4628 11 7E 02    LD DE,766(D)
4629 19          ADD HL,DE
462A EB          EX DE,HL
462B 01 5C 46    LD HL,465C
462C 01 13 00    LD BC,19(D)
462D ED B0      LDIR
462E CD BB 02    CALL K60AN
462F 7C          LD A,H
4630 29        AND L
4631 7E 07      CP B7
4632 CC 00 00    CALL NZ,0000
4633 7E 70 00    CP 70
4634 0A 70 41    JP NZ,41F6
4635 18 77 41    JR 463B
4636 3D          DEC A
4637 7E 1B      CP 1B
4638 29 05 46    JR NZ,4656
4639 38 25      LD (HL),25
463A 2B          DEC HL
463B 18 C8      JR 461E
463C 77          LD (HL),A
463D C1          POP BC
463E 21 00 08    LD HL,0800
463F C9          RET

```

DO ENDEREÇO 465C

AO ENDEREÇO 466E

ESTÃO OS DIZERES

"NOVA CORRIDA (S/N)?"

Este é o último grupo do programa. É a sub-rotina TemDim que no diagrama de blocos da figura 13 corresponde ao número 29. Apesar de sua importância os comentários não se fazem necessários já que ela é composta de uma seqüência de instruções já conhecidas e explicadas anteriormente.

```

17942 05 2A 00 40 11 0E 02 540
17949 19 7E A7 20 2A 05 04 402
17956 23 36 1C 10 FB C1 E1 802
1795B 2A 0C 40 11 FE 02 10 416
17970 EB 21 5C 46 01 13 00 450
17977 ED B0 CD BB 02 7C A5 1006
17984 FE 87 CC 00 00 7E 70 1070
17991 CA 76 41 18 E7 3D 7E 1060
17998 1B 20 09 36 05 0B 18 2000
18005 C8 77 C1 21 00 00 C0 704
18012 33 34 38 26 00 00 34 2000
18019 37 37 DE 20 26 07 00 2000
18026 10 38 18 33 11          184

```

RESUMO

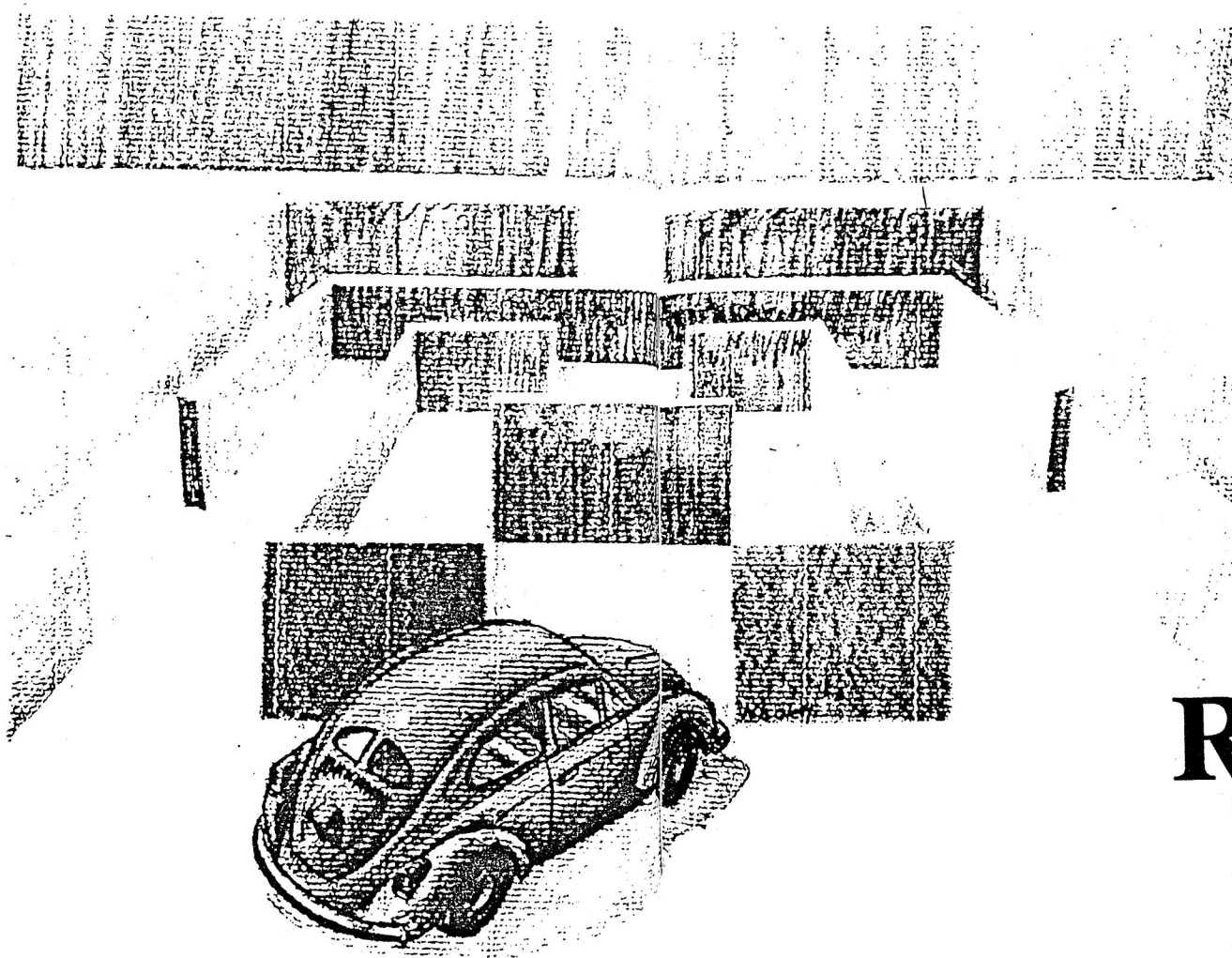
Foi abordado no presente capítulo o seguinte:

- * Forma de alterar o programa pelo uso de chamada a sub-rotina.
- * Conveniência e modo de fazer um sorteio orientado.
- * Técnica para manter concomitantemente dois ritmos distintos, no caso a redução uniforme do tempo na tela e o movimento variado dos carros.

SUGESTÕES

- 1 – Introduzir no programa uma rotina que, ao fim de cada partida, compare a distância rodada com um valor anteriormente conseguido, e registre o maior deles como "Record do Dia". Sendo mais explícito – eliminar da tela a frase "tempo é perdido em cada acidente" e em seu lugar criar um registro "Record do Dia" contendo o valor da distância conseguido em partidas anteriores e que servirá para comparação com o valor da distância rodada conseguida na última jogada.
- 2 – Este jogo foi desenvolvido basicamente pensando no uso do joystick e assim não foi prevista a possibilidade do jogador apertar duas teclas simultaneamente, ou seja, o comando de deslocar o carro à direita e acelerar ao mesmo tempo, por exemplo. Introduzir estas alternativas no programa.

REQUINTES



Requintes e sofisticação dão o verdadeiramente feliz acabamento a um jogo que se preze. Paradoxalmente aqui se discutem os requintes com um provinciano "Fuscão Preto". O festival de estranhos SCROLLs e apresentações do jogo são aqui discutidos em todos seus detalhes técnicos.

CAPÍTULO 6 – REQUINTES

Enfim chegou-se ao último capítulo. Isto não significa que o assunto, programação de jogos em linguagem de máquina para computadores da linha Sinclair, tenha se esgotado. Nem era essa a intenção do livro. Pretendeu-se, isto sim, dar uma visão geral e alguns exemplos práticos e claros relativos ao tema, o que se espera ter alcançado. E para finalizar, o enfoque dado nos requintes.

Entende-se por requintes, os detalhes e características não essenciais para o funcionamento do programa, visando melhorar o visual ou a sua apresentação. Nesta ordem de idéias, vários requintes já foram sugeridos em capítulos anteriores: criar telas com instruções, premiar o jogador caso atinja um certo desempenho, apresentar ao final da partida o registro do recorde do dia, etc. Destes, sem dúvida, as telas com instruções são os mais úteis; com elas, gastando apenas poucos metros a mais de fita, evita-se a necessidade de instruções escritas em papel que nem sempre são achados quando é necessário. O problema é que a capacidade de memória do equipamento pode não ser suficiente para conter o programa e as telas em questão. Cada uma destas ocupa quase 800 bytes e, em geral, contém tão poucas informações...

```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0 INSTRUÇÕES: TECLAS S SOBE D DESCE
1
2
3 FUSCAO PRETO
4
5 XIRATA-SE DE UMA CORRIDA ONDE
6 O CARRO "0" DEVE APAGAR OS(...)
7 PARA GANHAR PONTOS.
8
9 XO FUSCAO PRETO(X) PERSEGUE O
10 "0". PEGANDO-O PERDE-SE UMA VI-
11 DA. PARA FUGIR, USAR JOYSTICK OU
12 TECLAS 5, 6, 7 E 8 AFIM DE MUDAR
13 DE PISTA.
14
15 XAPERTAR TECLA 8 OU BOTAO DO
16 JOYSTICK FAZ COM QUE "0" AVAN-
17 CE MAIS QUE O X FACILITANDO A
18 FUGA POREM COM ISSO O BONUS DI-
19 MINUE.
20
21 XNAO SE DEVE APERTAR MAIS DE
22 UMA TECLA OU COMANDO POR VEZ.
23 APERTE A TECLA C PARA COMEGAR
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

```

FIGURA 14

Para ilustrar este capítulo escolheu-se um jogo, denominado "Fusão Preto", que tem uma apresentação bastante requintada. Infelizmente para aqueles possuidores de equipamentos com capacidade de 2 Kbytes, o programa em si, mesmo sem as instruções, ocupa mais do que isso.

Logo após carregar o programa aparece uma tela com instruções, cujo esboço é dado na figura 14.

Pressionando a tecla S vai havendo um efeito de Scroll, ou seja, as primeiras linhas vão sumindo no topo e novas linhas surgem na base, até chegar-se ao final, conforme ilustrado na figura 15.

```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0 INSTRUOE : TECLAS S SOBE D DESCE
1
2 CE MAIS QUE O X FACILITANDO A
3 FUGA POREM COM ISSO O BONUS DI
4 MINUE.
5
6 XNAO SE DEVE APERTAR MAIS DE
7 UMA TECLA OU COMANDO POR VEZ.
8 SE ISTO FOR FEITO O MOVIMENTO
9 NAO SERA EXECUTADO.
10
11 XSE TODOS (...) FOREM APAGADOS
12 O BONUS RESTANTE SOMA-SE A CON-
13 TAGEM. A CORRIDA REINICIA COM
14 MAIS VELOCIDADE.
15
16 XPARA OS CARROS PARTIREM BAS-
17 TA APERTAR UMA TECLA QUALQUER.
18
19 XXBOA SORTE E VAMOS LAXX
20 APERTE A TECLA C PARA COMEGAR.
21
22
23 APERTE A TECLA C PARA COMEGAR
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

```

FIGURA 15

Caso queiramos reler alguma instrução, basta pressionar D e o efeito se inverte, ou seja, as últimas linhas vão desaparecendo na base e novas aparecem no topo, gerando o que pode ser chamado de antiscroll.

Pressionando a tecla C, a tela altera-se ficando como o esboço da figura 16.

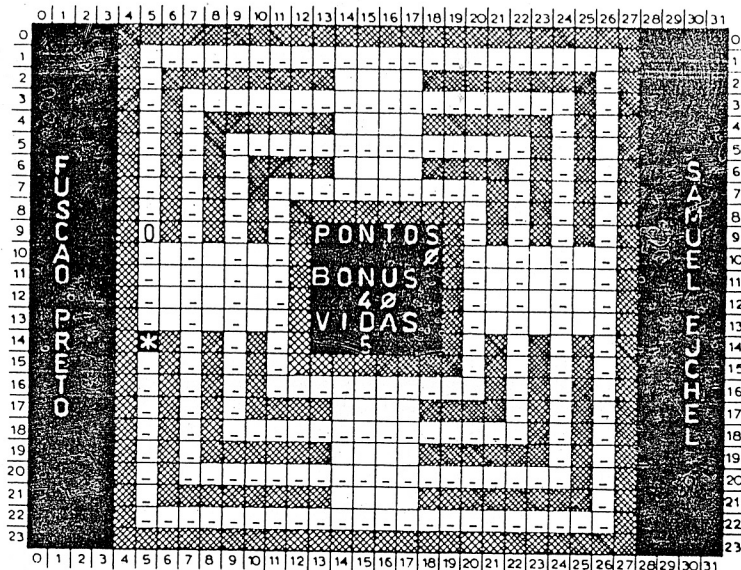


FIGURA 16

Acionando qualquer tecla a figura "O" e o asterisco de fundo escuro começam a mover-se em sentidos contrários — o "O" para cima (sentido horário) e o asterisco para baixo (sentido anti horário).

Ao mover-se, o "O" vai fazendo sumir os pontinhos e a cada um deles os Pontos da tela são incrementados. A passagem do asterisco não afeta o conteúdo da posição da tela.

Se a tecla zero, ou botão de disparo for acionada, o asterisco tem sua velocidade reduzida pela metade, o que equivale ao "O" avançar em relação a ele. A cada posição que o "O" se adiante, uma unidade é reduzida do Bonus.

Se o asterisco se "chocar" com o "O" uma vida é perdida, como já foi descrito nas figuras 14 e 15. Para evitar isto o "O" passando em frente às aberturas pode mudar de pista, desde que a alavanca do joystick ou as teclas 5 e 8 tenham sido acionadas adequadamente (as teclas 5 e 8 nas pistas verticais, à direita ou esquerda, e as 6 e 7 nas horizontais, em cima e em baixo). O asterisco é programado de forma a orientar seu movimento conforme o de "O", e assim aquele também mudará de pista se este o fizer. Daí a importância ou a maior facilidade de fuga se o avanço de "O" tiver sido feito previamente.

O basic associado ao programa é o seguinte:

```

00 REM ( 3400 BYTES )
10 SAVE "FUSCAO"
20 RAND USA 19677
30 RAND USA 19859

```

Isto é, em sua forma completa o programa utiliza mais de 4 Kbytes de memória e mesmo se somente as rotinas do jogo em si fossem consideradas, a capacidade necessária seria próxima dos 3 Kbytes.

Os Registros Improvisados criados são:

NOME	ENDEREÇO (HEXA-DEC)	ABREVIATURA
Pontos Tela	DFile + 349	
Bonus Tela	DFile + 413	
Vidas	DFile + 478	
Pista do "O"	4021 — 16417	PISTAO
Bonus	407B — 16507	BONUS
Contagem	407C — 16508	CONT
Velocidade	403C/D — 16444/5	VELOC
Acelerador	403E — 16446	ACEL
Posição do Asterisco	403F/40 — 16447/8	POSAST
Pista do Asterisco	4041 — 16449	PISTAST
Direção do Asterisco	4042 — 16450	DIRAST
Conteúdo da Posição do Asterisco	4043 — 16451	CONPOAST
Direção de "O"	4044 — 16452	DIRO

Os três primeiros registros são auto explicativos.

O registro Pista do "O" informa a cada instante em qual pista o "O" se encontra, e é utilizado primordialmente no movimento do asterisco para determinar se ele deve ou não mudar de pista. O conteúdo deste registro é um número de 1 a 4 em conformidade com aquele associado às pistas.

A pista 1 é a mais externa (mais comprida) e a 4 é a interna (mais curta) que contorna o centro; a 2 e 3 são as intermediárias.

O registro Bonus corresponde exatamente ao Bonus da tela e seu uso justifica-se apenas para simplificar a lógica de acelerar, (se Bonus = 0 não se pode acelerar) e ao final do jogo, para somar os valores remanescentes aos Pontos Tela. Em suma, seu conteúdo é um número de 0 a 40.

O registro Contagem contém um número de 0 a 239, e serve para controlar os pontinhos ainda existentes na tela. A cada um que é apagado o valor do registro é decrementado.

O registro Velocidade controla a velocidade do jogo como o próprio nome diz. Ou seja, a cada vez que o tempo é consumido para uma espera no jogo, é o valor contido neste registro que é transferido para um dos pares HL, DE ou BC para ser decrementado até zero. Cada vez que os pontinhos são totalmente apagados, o valor deste registro é diminuído para aumentar a velocidade.

O Acelerador é um registro que funciona como indicador para saber se no ciclo imediatamente anterior o Asterisco foi retardado ou não, de forma a garantir que a velocidade deste não fique inferior a metade da do "O".

Seu funcionamento é similar ao PARI do Sam's Race (capítulo 5) ou IND dos Invasores (capítulo 4).

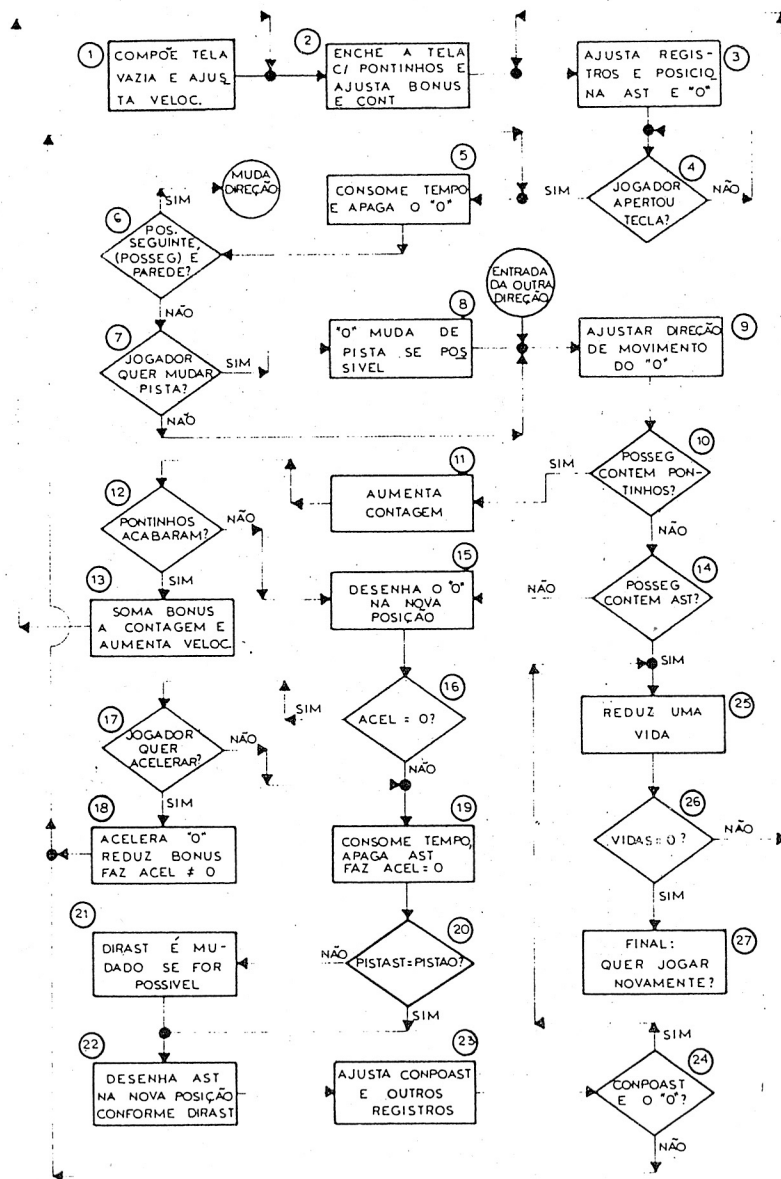
O POSAST é um clássico registro de posição do Asterisco. O PISTAST é o equivalente ao PISTAO.

O registro DIRAST é algo mais complicado. Pode assumir um valor de 1 a 12. Quando o Asterisco está "descendo" pelo lado esquerdo da tela, ele pode ter um entre três valores: 1 se descer direto, 2 se estiver passando de uma pista para outra de número mais baixo (de 3 para 2 por exemplo) e 3 se for de uma pista mais baixa para mais alta (de 3 para 4 por exemplo). Em resumo temos:

DIREÇÃO DO ASTERISCO	VALOR DE DIRAST		
	DIRETO	AUMENTA	DIMINUI
Lado Esquerdo — descendo	1	3	2
Embaixo — da esquerda para direita	4	5	6
Direita — subindo	7	8	9
Em cima — da direita para esquerda	10	11	12

NOTA: a inversão entre 3 e 2 foi acidental e resolveu-se não alterar o programa somente por isso.

FIGURA 17



O CONPOAST informa o conteúdo que havia na posição da tela em que o Asterisco está sendo desenhado de forma que, ao mudar de posição no próximo ciclo, se possa redesenhar o conteúdo anterior sem alterá-lo. Assim o registro pode conter 00 (vazio), 1B (pontinho) e 34 (figura "O"). O último caso indica que uma vida será perdida.

Por último tem-se o DIRO. Este registro é previsto para conter apenas dois valores, ou 01 ou FF. O motivo de sua existência é que o movimento de "O" é governado por duas sub-rotinas — uma para quando ele sobe ou desce (lado esquerdo ou direito da tela) e outra para quando ele se move horizontalmente (da direita para esquerda ou vice-versa). Para distinguir as alternativas DIRO é 01, quando "O" sobe ou vai da esquerda para direita e é FF quando desce ou vai da direita para esquerda.

Isto posto veja agora o diagrama de blocos que é apresentado na figura 17.

Logo no início pode-se observar que o diagrama não é muito complicado, o que, a primeira vista, não justifica os mais de 2 Kbytes necessários para traduzi-los em linguagem de máquina.

O motivo é que, a menos dos blocos de inicialização e finalização e de algumas sub-rotinas comuns, o diagrama que é apresentado refere-se a apenas uma das direções do movimento de "O", o de subida do lado esquerdo por exemplo. Em termos de lógica há uma repetição nas quatro direções, porém em termos de programa, na concepção em que foi feito, há diferenças que implicam em ter que escrevê-lo quatro vezes. A saída de uma direção se dá no bloco 6, e a entrada no grupo de instruções da outra direção se dá no ponto de acesso ao bloco 9.

Conhecendo os Registros Improvisados e os comentários apresentados não haverá dificuldade no entendimento do diagrama da figura 17. Pode-se então partir para o exame dos grupos de instrução em linguagem de máquina.

Grupo 1 — 16514 a 17066 (4082 a 42AA em hexa)

```

40802 00 00 40 LD HL,(D-FILE)
40805 00 18 LD A,18
40807 00 04 LD B,04
40809 00 04 INC HL
4080A 00 80 LD (HL),80
4080C 10 FB DJNZ 40809
4080E 00 80 INC HL
4080F 00 08 LD (HL),08
40811 00 DEC A

```

```

40802 00 05 JR Z,4080A
40804 11 10 LD DE,28(D)
40807 10 00 ADD HL,DE
40808 10 00 JR 40807
4080A 11 00 AND A
4080B 11 00 LD DE,02E0
4080E 10 00 SBC HL,DE
4080F 11 10 LD DE,29(D)
40810 00 18 LD A,24(D)
40813 00 08 LD (HL),08
40817 00 04 LD B,04
40819 00 03 INC HL
4081A 00 80 LD (HL),80
4081C 10 FB DJNZ 40809
4081E 00 00 DEC A
4081F 00 00 JR Z,40804
40821 10 00 ADD HL,DE
40822 10 F1 JR 40825
40824 11 47 AND A
40825 10 02 LD DE,581(D)
40828 10 00 SBC HL,DE
4082A 11 10 LD DE,29(D)
4082D 01 04 LD BC,0004
4082E 00 AB LD (HL),AB
40830 10 00 ADD HL,DE
40833 00 B8 LD (HL),B8
40835 00 00 ADD HL,BC
40838 00 BA LD (HL),BA
4083A 10 00 ADD HL,DE
4083C 00 A6 LD (HL),A6
4083E 00 00 ADD HL,BC
40840 00 B8 LD (HL),B8
40842 10 00 ADD HL,DE
40845 00 B2 LD (HL),B2
40847 00 00 ADD HL,BC
40849 00 A8 LD (HL),A8
4084B 10 00 ADD HL,DE
4084E 00 BA LD (HL),BA
40850 00 00 ADD HL,BC
40853 00 A6 LD (HL),A6
40855 10 00 ADD HL,DE
40858 00 AA LD (HL),AA
4085A 00 00 ADD HL,BC
4085D 00 B4 LD (HL),B4
4085F 10 00 ADD HL,DE
40861 00 B1 LD (HL),B1
40863 00 00 ADD HL,BC
40865 10 00 ADD HL,DE
40868 00 BC ADD HL,BC
4086A 00 B5 LD (HL),B5
4086C 10 00 ADD HL,DE
4086E 00 AA LD (HL),AA

```

```

400000 ADD HL,BC
400001 LD (HL),B7
400002 ADD HL,DE
400003 LD (HL),AF
400004 ADD HL,BC
400005 LD (HL),AA
400006 ADD HL,DE
400007 LD (HL),A8
400008 ADD HL,BC
400009 LD (HL),89
40000A ADD HL,DE
40000B LD (HL),AD
40000C ADD HL,BC
40000D LD (HL),B4
40000E ADD HL,DE
40000F LD (HL),AA
410010 ADD HL,BC
410011 ADD HL,DE
410012 LD (HL),B1
410013 AND A
410014 LD DE,619(D)
410015 SBC HL,DE
410016 LD C,02
410017 LD B,22(D)
410018 LD (HL),08
410019 INC HL
41001A DJNZ 4111
41001B DEC C
41001C JR Z,4111F
41001D LD DE,737(D)
41001E ADD HL,DE
41001F JR 410F
410020 AND A
410021 LD DE,714(D)
410022 SBC HL,DE
410023 LD C,02
410024 LD B,02
410025 LD B,08
410026 LD (HL),08
410027 INC HL
410028 DJNZ 4128
410029 DEC A
41002A JR Z,4139
41002B INC HL
41002C INC HL
41002D INC HL
41002E JR 4129
41002F DEC C
410030 JR Z,4142
410031 LD DE,607(D)
410032 ADD HL,DE

```

```

414000 JR 4127
414001 AND A
414002 LD DE,579(D)
414003 SBC HL,DE
414004 LD C,02
414005 LD B,02
414006 LD B,08
414007 LD (HL),08
414008 INC HL
414009 DJNZ 414E
41400A DEC A
41400B JR Z,415C
41400C INC HL
41400D INC HL
41400E INC HL
41400F INC HL
414010 JR 414C
414011 DEC C
414012 JR Z,4165
414013 LD DE,479(D)
414014 ADD HL,DE
414015 JR 414A
414016 AND A
414017 LD DE,443(D)
414018 SBC HL,DE
414019 LD C,02
41401A LD B,02
41401B LD B,04
41401C LD (HL),08
41401D INC HL
41401E DJNZ 4171
41401F DEC A
414020 JR Z,417F
414021 INC HL
414022 INC HL
414023 INC HL
414024 INC HL
414025 JR 416F
414026 DEC C
414027 JR Z,4188
414028 LD DE,351(D)
414029 ADD HL,DE
41402A JR 416D
41402B AND A
41402C LD DE,307(D)
41402D SBC HL,DE
41402E LD C,02
41402F LD B,08
414030 LD (HL),08
414031 INC HL
414032 DJNZ 4192
414033 DEC C
414034 JR Z,41A0

```

```

4119A 41 DF 00 LD DE,223(D)
4119D 410F ADD HL,DE
4119E 4107 JR 4190
411A0 417F AND A
411A1 4107 LD DE,205(D)
411A4 4107 SBC HL,DE
411A6 4100 LD (HL),08
411A8 4100 INC HL
411AA 4105 LD (HL),B5
411AC 4100 INC HL
411AE 4104 LD (HL),B4
411B0 4100 INC HL
411B2 4103 LD (HL),B3
411B4 4100 INC HL
411B6 4109 LD (HL),B9
411B8 4104 INC HL
411BA 4107 LD (HL),B4
411BC 4100 INC HL
411BE 4103 LD (HL),B8
411C0 4100 INC HL
411C2 4108 LD (HL),08
411C4 4121 00 LD DE,33(D)
411C6 4109 ADD HL,DE
411C8 4100 LD (HL),08
411CA 4100 DEC HL
411CC 4103 LD (HL),9C
411CE 4100 LD B,05
411D0 4100 DEC HL
411D2 4100 LD (HL),80
411D4 4107 DJNZ 4108
411D6 4100 DEC HL
411D8 4103 LD (HL),08
411DA 4100 ADD HL,DE
411DC 4100 LD (HL),08
411DE 4100 INC HL
411E0 4107 LD (HL),A7
411E2 4100 INC HL
411E4 4104 LD (HL),B4
411E6 4100 INC HL
411E8 4103 LD (HL),B3
411EA 4100 INC HL
411EC 410A LD (HL),BA
411EE 4100 INC HL
411F0 4103 LD (HL),B8
411F2 4100 INC HL
411F4 4100 LD (HL),80
411F6 4100 INC HL
411F8 4103 LD (HL),08
411FA 4100 ADD HL,DE
411FC 4100 LD (HL),08
411FE 4100 DEC HL
41200 4100 LD (HL),80

```

```

411FE 4100 DEC HL
41202 4100 LD (HL),80
41204 4100 DEC HL
41206 4100 LD (HL),9C
41208 4100 DEC HL
4120A 4100 LD (HL),A8
4120C 4100 DEC HL
4120E 4100 LD (HL),80
41210 4100 DEC HL
41212 4100 LD (HL),80
41214 4100 DEC HL
41216 4100 LD (HL),08
41218 4100 ADD HL,DE
4121A 4100 LD (HL),08
4121C 4100 INC HL
4121E 4100 LD (HL),B8
41220 4100 INC HL
41222 4100 LD (HL),B8
41224 4100 INC HL
41226 4100 LD (HL),B8
41228 4100 INC HL
4122A 4100 LD (HL),B8
4122C 4100 INC HL
4122E 4100 LD (HL),B8
41230 4100 INC HL
41232 4100 LD (HL),B8
41234 4100 INC HL
41236 4100 LD (HL),B8
41238 4100 INC HL
4123A 4100 LD (HL),B8
4123C 4100 INC HL
4123E 4100 LD (HL),B8
41240 4100 INC HL
41242 4100 LD (HL),B8
41244 4100 INC HL
41246 4100 LD (HL),B8
41248 4100 INC HL
4124A 4100 LD (HL),B8
4124C 4100 INC HL
4124E 4100 LD (HL),B8
41250 4100 INC HL
41252 4100 LD (HL),B8
41254 4100 INC HL
41256 4100 LD (HL),B8
41258 4100 INC HL
4125A 4100 LD (HL),B8
4125C 4100 INC HL
4125E 4100 LD (HL),B8
41260 4100 INC HL
41262 4100 LD (HL),B8
41264 4100 INC HL
41266 4100 LD (HL),B8
41268 4100 INC HL
4126A 4100 LD (HL),B8
4126C 4100 INC HL
4126E 4100 LD (HL),B8
41270 4100 INC HL
41272 4100 LD (HL),B8
41274 4100 INC HL
41276 4100 LD (HL),B8
41278 4100 INC HL
4127A 4100 LD (HL),B8
4127C 4100 INC HL
4127E 4100 LD (HL),B8
41280 4100 INC HL
41282 4100 LD (HL),B8
41284 4100 INC HL
41286 4100 LD (HL),B8
41288 4100 INC HL
4128A 4100 LD (HL),B8
4128C 4100 INC HL
4128E 4100 LD (HL),B8
41290 4100 INC HL
41292 4100 LD (HL),B8
41294 4100 INC HL
41296 4100 LD (HL),B8
41298 4100 INC HL
4129A 4100 LD (HL),B8
4129C 4100 INC HL
4129E 4100 LD (HL),B8
412A0 4100 INC HL
412A2 4100 LD (HL),B8
412A4 4100 INC HL
412A6 4100 LD (HL),B8
412A8 4100 INC HL
412AA 4100 LD (HL),B8
412AC 4100 INC HL
412AE 4100 LD (HL),B8
412B0 4100 INC HL
412B2 4100 LD (HL),B8
412B4 4100 INC HL
412B6 4100 LD (HL),B8
412B8 4100 INC HL
412BA 4100 LD (HL),B8
412BC 4100 INC HL
412BE 4100 LD (HL),B8
412C0 4100 INC HL
412C2 4100 LD (HL),B8
412C4 4100 INC HL
412C6 4100 LD (HL),B8
412C8 4100 INC HL
412CA 4100 LD (HL),B8
412CC 4100 INC HL
412CE 4100 LD (HL),B8
412D0 4100 INC HL
412D2 4100 LD (HL),B8
412D4 4100 INC HL
412D6 4100 LD (HL),B8
412D8 4100 INC HL
412DA 4100 LD (HL),B8
412DC 4100 INC HL
412DE 4100 LD (HL),B8
412E0 4100 INC HL
412E2 4100 LD (HL),B8
412E4 4100 INC HL
412E6 4100 LD (HL),B8
412E8 4100 INC HL
412EA 4100 LD (HL),B8
412EC 4100 INC HL
412EE 4100 LD (HL),B8
412F0 4100 INC HL
412F2 4100 LD (HL),B8
412F4 4100 INC HL
412F6 4100 LD (HL),B8
412F8 4100 INC HL
412FA 4100 LD (HL),B8
412FC 4100 INC HL
412FE 4100 LD (HL),B8

```

```

DEC HL
LD (HL),80
DEC HL
LD (HL),9C
DEC HL
LD (HL),A8
DEC HL
LD (HL),80
DEC HL
LD (HL),80
DEC HL
LD (HL),08
ADD HL,DE
LD (HL),08
INC HL
LD (HL),B8
INC HL
LD (HL),AE
INC HL
LD (HL),A8
INC HL
LD (HL),A8
INC HL
LD (HL),B8
INC HL
LD (HL),80
INC HL
LD (HL),80
INC HL
LD (HL),08
ADD HL,DE
LD (HL),08
LD B,05
DEC HL
LD (HL),80
DJNZ 421D
DEC HL
LD (HL),08
INC HL
LD (HL),A1
AND A
LD DE,372(D)
SBC HL,DE
LD C,02
LD DE,33(D)
LD A,02
LD B,07
LD (HL),08
ADD HL,DE
DJNZ 4239
DEC A
JR Z,4247
ADD HL,DE

```

```

400000 44 11110000 ADD HL,DE
400001 44 11110000 ADD HL,DE
400002 44 11110000 ADD HL,DE
400003 44 11110000 JR 4237
400004 44 11110000 DEC C
400005 44 11110000 JR Z,4252
400006 44 11110000 AND A
400007 44 11110000 LD DE,575 (D)
400008 44 11110000 SBC HL,DE
400009 44 11110000 JR 4232
400010 44 11110000 AND A
400011 44 11110000 LD DE,545 (D)
400012 44 11110000 SBC HL,DE
400013 44 11110000 LD C,02
400014 44 11110000 LD DE,33 (D)
400015 44 11110000 LD A,02
400016 44 11110000 LD B,08
400017 44 11110000 LD (HL),08
400018 44 11110000 ADD HL,DE
400019 44 11110000 DJNZ 4261
400020 44 11110000 DEC A
400021 44 11110000 JR Z,426F
400022 44 11110000 ADD HL,DE
400023 44 11110000 ADD HL,DE
400024 44 11110000 ADD HL,DE
400025 44 11110000 ADD HL,DE
400026 44 11110000 JR 425F
400027 44 11110000 DEC C
400028 44 11110000 JR Z,427A
400029 44 11110000 AND A
400030 44 11110000 LD DE,447 (D)
400031 44 11110000 SBC HL,DE
400032 44 11110000 JR 425A
400033 44 11110000 AND A
400034 44 11110000 LD DE,409 (D)
400035 44 11110000 SBC HL,DE
400036 44 11110000 LD A,02
400037 44 11110000 LD DE,33 (D)
400038 44 11110000 LD (HL),08
400039 44 11110000 ADD HL,DE
400040 44 11110000 LD (HL),08
400041 44 11110000 ADD HL,DE
400042 44 11110000 LD (HL),08
400043 44 11110000 ADD HL,DE
400044 44 11110000 LD (HL),08
400045 44 11110000 ADD HL,DE
400046 44 11110000 ADD HL,DE
400047 44 11110000 ADD HL,DE
400048 44 11110000 ADD HL,DE
400049 44 11110000 ADD HL,DE
400050 44 11110000 ADD HL,DE
400051 44 11110000 LD (HL),08
400052 44 11110000 ADD HL,DE
400053 44 11110000 LD (HL),08
400054 44 11110000 ADD HL,DE
400055 44 11110000 LD (HL),08
400056 44 11110000 ADD HL,DE
400057 44 11110000 LD (HL),08
400058 44 11110000 ADD HL,DE
400059 44 11110000 LD (HL),08
400060 44 11110000 ADD HL,DE

```

```

400060 06 08 LD (HL),08
400061 06 08 DEC A
400062 06 08 JR Z,42A5
400063 06 08 AND A
400064 06 08 LD DE,286 (D)
400065 06 08 SBC HL,DE
400066 06 08 JR 4282
400067 06 08 LD HL,02AA
400068 06 08 LD (VELOC),HL

```

Este grupo corresponde ao bloco 1, ou seja, compõe a estrutura das pistas, porém sem os pontinhos. Além disto ajusta a VELOC.

```

100000 10 00000000
100001 10 00000000
100002 10 00000000
100003 10 00000000
100004 10 00000000
100005 10 00000000
100006 10 00000000
100007 10 00000000
100008 10 00000000
100009 10 00000000
100010 10 00000000
100011 10 00000000
100012 10 00000000
100013 10 00000000
100014 10 00000000
100015 10 00000000
100016 10 00000000
100017 10 00000000
100018 10 00000000
100019 10 00000000
100020 10 00000000
100021 10 00000000
100022 10 00000000
100023 10 00000000
100024 10 00000000
100025 10 00000000
100026 10 00000000
100027 10 00000000
100028 10 00000000
100029 10 00000000
100030 10 00000000
100031 10 00000000
100032 10 00000000
100033 10 00000000
100034 10 00000000
100035 10 00000000
100036 10 00000000
100037 10 00000000
100038 10 00000000
100039 10 00000000
100040 10 00000000
100041 10 00000000
100042 10 00000000
100043 10 00000000
100044 10 00000000
100045 10 00000000
100046 10 00000000
100047 10 00000000
100048 10 00000000
100049 10 00000000
100050 10 00000000
100051 10 00000000
100052 10 00000000
100053 10 00000000
100054 10 00000000
100055 10 00000000
100056 10 00000000
100057 10 00000000
100058 10 00000000
100059 10 00000000
100060 10 00000000

```


4311	23	INC	HL	4340	35	34	LD	(HL),34
4312	23	INC	HL	4341	35		PUSH	HL
4313	35	LD	(HL),00	4342	35	02	CALL	KSCAN
4315	11	LD	DE,14(D)	4350	3C		INC	L
4316	19	ADD	HL,DE	4353	38	7A	JR	NZ,434F
4319	19	DJNZ	42FD	4355	13	09	JR	4355

Faz exatamente o prescrito pelo bloco 2 da figura 17 – Sem comentários.

17067	00	00	70	40	3E	FF	640
17068	00	00	70	40	3E	FF	640
17069	00	00	70	40	3E	FF	640
17070	00	00	70	40	3E	FF	640
17071	00	00	70	40	3E	FF	640
17072	00	00	70	40	3E	FF	640
17073	00	00	70	40	3E	FF	640
17074	00	00	70	40	3E	FF	640
17075	00	00	70	40	3E	FF	640
17076	00	00	70	40	3E	FF	640
17077	00	00	70	40	3E	FF	640
17078	00	00	70	40	3E	FF	640
17079	00	00	70	40	3E	FF	640
17080	00	00	70	40	3E	FF	640
17081	00	00	70	40	3E	FF	640
17082	00	00	70	40	3E	FF	640
17083	00	00	70	40	3E	FF	640
17084	00	00	70	40	3E	FF	640
17085	00	00	70	40	3E	FF	640
17086	00	00	70	40	3E	FF	640
17087	00	00	70	40	3E	FF	640
17088	00	00	70	40	3E	FF	640
17089	00	00	70	40	3E	FF	640
17090	00	00	70	40	3E	FF	640
17091	00	00	70	40	3E	FF	640
17092	00	00	70	40	3E	FF	640
17093	00	00	70	40	3E	FF	640
17094	00	00	70	40	3E	FF	640
17095	00	00	70	40	3E	FF	640
17096	00	00	70	40	3E	FF	640
17097	00	00	70	40	3E	FF	640
17098	00	00	70	40	3E	FF	640
17099	00	00	70	40	3E	FF	640

Grupo 3 – 17179 a 17238 (431B a 4356 em hexa)

431B	01	00	04	LD	HL,2400
431E	00			DEC	HL
431F	7C			LD	A,H
4320	00			OR	L
4321	00	76		JR	NZ,431E
4323	00	01		LD	A,01
4325	00	21	40	LD	(PISTAO),A
4326	00	41	40	LD	(PISTAST),A
4327	00	4B	40	LD	(DIRAST),A
4328	00	44	40	LD	(DIRO),A
432A	00	7F		XOR	A
432B	00	3E	40	LD	(ACEL),A
432C	00	5C	40	LD	HL,(D-FILE)
432D	11	04	01	LD	DE,458(D)
432E	19			ADD	HL,DE
432F	00	3F	40	LD	(403F),HL
4330	7F			LD	A,(HL)
4331	32	43	40	LD	(COMPOST),A
4332	38	97		LD	(HL),97
4333	2D	0C	40	LD	HL,(D-FILE)
4334	11	0F	01	LD	DE,303(D)
4335	19			ADD	HL,DE

O único detalhe a citar é o fato de que o endereço onde o "0" se encontra é acumulado no stack, através da instrução PUSH HL, imediatamente antes do conhecido Call KScan, que constitui o bloco 4 da figura 17. As instruções anteriores estão ligadas ao bloco 3.

17179	00	00	70	40	3E	FF	640
17180	00	00	70	40	3E	FF	640
17181	00	00	70	40	3E	FF	640
17182	00	00	70	40	3E	FF	640
17183	00	00	70	40	3E	FF	640
17184	00	00	70	40	3E	FF	640
17185	00	00	70	40	3E	FF	640
17186	00	00	70	40	3E	FF	640
17187	00	00	70	40	3E	FF	640
17188	00	00	70	40	3E	FF	640
17189	00	00	70	40	3E	FF	640
17190	00	00	70	40	3E	FF	640
17191	00	00	70	40	3E	FF	640
17192	00	00	70	40	3E	FF	640
17193	00	00	70	40	3E	FF	640
17194	00	00	70	40	3E	FF	640
17195	00	00	70	40	3E	FF	640
17196	00	00	70	40	3E	FF	640
17197	00	00	70	40	3E	FF	640
17198	00	00	70	40	3E	FF	640
17199	00	00	70	40	3E	FF	640

Grupo 4 – 17239 a 17276 (4357 a 437C em hexa)

4357	00	3C	40	LD	HL,(VELOC)
4358	00			DEC	HL
4359	7C			LD	A,H
435A	00			OR	L
435B	00	76		JR	NZ,435A
435C	00			RET	
435D	00	57	40	CALL	4357
435E	01			POP	HL
435F	06	00		LD	(HL),00
4360	11	01	00	LD	DE,33(D)
4361	00			AND	P
4362	00	52		SBC	HL,DE
4363	46			LD	B,(HL)
4364	00	52		SBC	HL,DE
4365	4E			LD	C,(HL)
4366	19			ADD	HL,DE
4367	40			ADD	HL,DE
4368	40			LD	A,08
4369	00	06		CP	B
436A	00	57		JR	NZ,436E
436B	00			CP	C
436C	00	00		JR	NZ,4370
436D	00	00		CP	C
436E	00	9B	44	CALL	445B

As seis primeiras instruções do grupo constituem-se numa sub-rotina de consumo de tempo baseada no Registro VELOC. que será "chamada" diversas vezes ao longo do programa. A primeira destas ocorre imediatamente em seguida, e considerando as duas posteriores (POR HL – LD (HL), 00) tem-se atendido o bloco 5.

As instruções seguintes estão associadas ao bloco 6, porém na condição de "O" estar subindo. Isto é facilmente notado pelo fato da POSSEG (Posição Seguinte) ser obtida da anterior subtraída de 33 em decimais. Feito o teste se a posição seguinte é a parede, o salto relativo leva o programa para o grupo onde o "O" se movimenta a direita. A razão de testar duas posições à frente (através do uso do registro C) deve-se a uma alternativa que existia e foi eliminada do programa.

Ao fim deste grupo existe uma chamada a uma sub-rotina. Esta vai se constituir nos blocos 7 e 8.

```

172330 0A 00 40 00 70 00 00 046
172400 7B 00 00 07 40 01 00 10000
172500 00 11 00 01 06 07 00 00000
172600 40 00 00 10 10 00 00 00000
172667 00 00 00 07 00 00 00 047
172774 00 00 44 00 00 00 00 408
  
```

Grupo 5 – 17277 a 17334 (437D a 43B6 em hexa)

```

437D 3E 01 LD A,01
437F 32 44 40 LD (DIRO),A
4382 A7 AND A
4383 11 01 00 LD DE,33(D)
4386 ED 02 SBC HL,DE
4388 7E LD A,(HL)
4389 7E 18 CP 18
438B 00 ED 45 CALL Z,45EC
438E 7E 97 CP 97
4390 00 ED A5 45 JP Z,45A5
4393 ED 00 PUSH HL
4394 06 34 LD (HL),34
4396 3A 3E 40 LD A,(ACEL)
4399 A7 AND A
439B 00 00 JR NZ,43A5
439C 00 7B 45 CALL 457B
439F 3A 3E 40 LD A,(ACEL)
43A2 A7 AND A
43A3 00 00 JR NZ,43B0
43A5 00 00 45 CALL 4555
  
```

```

43B8 00 40 40 LD A,(CONPOAST)
43BB 7E 04 CP 04
43BD 00 01 JR NZ,43C0
43BF 7E 00 XOR A
43C0 40 40 LD (CONPOAST),A
43C3 00 00 POP HL
43C4 00 A5 45 JP 45A5
  
```

Se "O" viesse pelo ramo de baixo, da direita para esquerda, seria neste ponto que ele entraria no programa do movimento de subida. Por isso o grupo inicia-se acertando o DIRO, e carregando DE com 33 em decimal. Isto é o bloco 9.

Em seguida tem-se o bloco 10, com as instruções de CP 1B e a correspondente chamada a sub-rotina, que irá constituir os blocos 11 e 12. Na sequência o CP97 e o salto condicionado a zero constitui o bloco 14, sendo que o endereço do salto leva para onde estão as instruções dos blocos 25, 26 e 27.

A próxima etapa é fazer um PUSH HL que será explicado adiante, e desenha-se o "O", isto é, atingimos o bloco 15. As instruções do bloco 16 é que surgem a seguir e aí, se ACEL não for zero dá-se um salto para o bloco 5, ou seja, grupo 4 onde o endereço do "O" é obtido pelo POP. Eis a razão do PUSH, que é ainda reforçada pelo detalhe de que ao fazer a chamada à sub-rotina do bloco 17, objeto da próxima instrução, haverá uma consulta ao teclado (Call KScan), e caso o jogador tenha acelerado, o salto para a mesma posição citada acima, do bloco 5 grupo 4, ocorrerá.

O final do grupo corresponde a uma chamada à sub-rotina que comanda o movimento do Asterisco, englobando os blocos 19, 20, 21, 22 e 23. A verificação do CONPOAST, objeto do bloco 24, é ainda parte deste grupo, que acaba com um salto ao conjunto dos blocos 25, 26 e 27 caso haja lá o 34. Em caso contrário novamente, se atingirá o bloco 5 do grupo 4.

```

17277 3E 01 32 44 40 A7 11 408
17284 21 00 ED 02 7E 18 10000
17291 00 00 40 7E 07 00 00000
17298 40 00 00 04 00 00 00000
17305 A7 00 00 ED 7B 45 00 00000
17312 3E 40 A7 00 00 00 00000
17319 40 3A 43 40 7E 34 00 00000
17326 B1 AF 3E 40 40 E1 00 00000
17333 A5 45 00 00 00 00 004
  
```

Grupo 6 – 17335 a 17405 (43B7 a 43FD em hexa)

```

43B7 00 57 43 CALL 4357
43BA E1 POP HL
43BB 3E 00 LD (HL),00
  
```

```

43BD  02          INC  HL
43BE  46          LD   B, (HL)
43BF  02          INC  HL
43C0  4E          LD   C, (HL)
43C1  02          DEC  HL
43C2  02          DEC  HL
43C3  00 08      LD   A, 08
43C4  00 08      CP   B, 08
43C5  00 53      JR   Z, 441B
43C6  00 00      CP   C
43C7  00 00      JR   Z, 43CE
43C8  00 04 45   CALL 4304
43C9  00 00      INC  HL
43CA  00 00      LD   A, (HL)
43CB  00 18      CP   18
43CC  00 00 45   CALL  Z, 45EC
43CD  00 00 97   CP   97
43CE  00 00 95   JP   Z, 45A5
43CF  00 00      PUSH HL
43D0  00 34      LD   (HL), 34
43D1  00 3E 40   LD   A, (ACEL)
43D2  00 07      AND  A
43D3  00 00      JR   NZ, 43EC
43D4  00 00 7B 45   CALL 457B
43D5  00 00 3E 40   LD   A, (ACEL)
43D6  00 07      AND  A
43D7  00 00 0B      JR   NZ, 43B7
43D8  00 00 55 45   CALL 4655
43D9  00 00 43 40   LD   A, (CONPOAST)
43DA  00 00 34      CP   34
43DB  00 00 01      JR   NZ, 43B7
43DC  00 00      XOR  A
43DD  00 43 40   LD   (CONPOAST), A
43DE  00 01      POP  HL
43DF  00 03 45   JP   45A5

```

Esta é exatamente a repetição dos grupos 4 e 5 porém aplicada ao ramo de cima, ou seja, naquele em que o "O" se desloca da esquerda para a direita.

```

173305 00 57 43 00 00 00 00 00 873
173310 00 00 00 00 00 00 00 00 300
173315 00 00 00 00 00 00 00 00 740
173320 00 04 40 00 00 00 00 00 710
173325 00 00 40 00 00 00 00 40 1140
173330 00 00 40 00 00 00 00 40 000
173335 00 00 00 00 00 00 40 07 000
173340 00 00 00 00 00 40 00 00 000
173345 00 00 00 00 00 40 00 00 000
173350 00 00 00 00 00 40 00 00 000
173355 00 00 00 00 00 40 00 00 000
173360 00 00 00 00 00 40 00 00 000
173365 00 00 00 00 00 40 00 00 000
173370 00 00 00 00 00 40 00 00 000
173375 00 00 00 00 00 40 00 00 000
173380 00 00 00 00 00 40 00 00 000
173385 00 00 00 00 00 40 00 00 000
173390 00 00 00 00 00 40 00 00 000
173395 00 00 00 00 00 40 00 00 000
174005 45

```

Grupo 7 – 17406 a 17490 (43FE a 4452 em hexa)

```

43FE  00 57 43          CALL 4357
4401  00 00          POP  HL
4402  00 00          LD   (HL), 00
4404  00 21 00      LD   DE, 33(D)
4407  00 19          ADD  HL, DE
4408  00 46          LD   B, (HL)
4409  00 19          ADD  HL, DE
440A  00 4E          LD   C, (HL)
440B  00 A7          AND  A
440C  00 00 52      SBC  HL, DE
440D  00 00 02      SBC  HL, DE
440E  00 00 08      LD   A, 08
4410  00 00          CP   B
4411  00 00 56      JR   Z, 446B
4412  00 00          CP   C
4413  00 00 03      JR   Z, 441B
4414  00 00 00      CALL 449B
4415  00 00 44      LD   A, FF
4416  00 00 44 40   LD   (DIRO), A
4417  00 00 21 00   LD   DE, 33(D)
4418  00 00 19          ADD  HL, DE
4419  00 00 7E          LD   A, (HL)
4420  00 00 1B          CP   1B
4421  00 00 0C 45   CALL  Z, 45EC
4422  00 00 97          CP   97
4423  00 00 A5 45   JP   Z, 45A5
4424  00 00 00          PUSH HL
4425  00 00 34          LD   (HL), 34
4426  00 00 3E 40   LD   A, (ACEL)
4427  00 00 07          AND  A
4428  00 00 00          JR   NZ, 4401
4429  00 00 00          XOR  A
4430  00 00 34          LD   (CONPOAST), A
4431  00 00 3E 40   LD   A, (ACEL)
4432  00 00 A7          AND  A
4433  00 00 20 00   JR   NZ, 4441
4434  00 00 7B 45   CALL 457B
4435  00 00 3E 40   LD   A, (ACEL)
4436  00 00 A7          AND  A
4437  00 00 20 00   JR   NZ, 4401
4441  00 00 55 46   CALL 4655
4444  00 00 3A 43 40   LD   A, (CONPOAST)
4447  00 00 34          CP   34
4448  00 00 20 00   JR   NZ, 4401
444B  00 00 A7          XOR  A
444C  00 00 3E 40   LD   (CONPOAST), A
444F  00 00 01          POP  HL
4450  00 00 03 45   JP   45A5

```

Nova repetição dos grupos 4 e 5, agora aplicado ao ramo da direita em que o "O" desce.


```

44B5 23      INC HL
44B6 00 44 40 LD A, (DIRO)
44B9 47      LD B, A
44BA 00 21 40 LD A, (PISTAO)
44BD 00      SUB A, B
44BE 00 21 40 LD A, (PISTAO), A
44C1 10 10     JR 44D9
44C2 20      INC HL
44C4 7E      LD A, (HL)
44C5 7E 08     CP 08
44C7 20 00     JR NZ, 44CB
44C8 20      DEC HL
44CA 00      RET
44CB 20      INC HL
44CC 00      PUSH HL
44CD 20      DEC HL
44CE 0A 44 40 LD A, (DIRO)
44D1 47      LD B, A
44D2 0A 21 40 LD A, (PISTAO)
44D5 00      ADD B
44D6 00 21 40 LD A, (PISTAO), A
44D9 7E      LD A, (HL)
44DA 7E 07     CP 07
44DC 20 00     JR NZ, 44E4
44DE 00      POP HL
44DF 21 05 45 LD HL, 45A5
44E0 00      EX (SP), HL
44E3 00      RET
44E4 00 34     LD (HL), 34
44E5 00      PUSH HL
44E7 00 55 46 CALL 4655
44EA 00 43 46 LD A, (CONPORST)
44EB 7E 34     CP 34
44ED 20 00     JR NZ, 44FC
44F1 0F      XOR A
44F2 00 43 40 LD A, (CONPORST), A
44F5 00      POP HL
44F6 00      POP HL
44F7 21 05 45 LD HL, 45A5
44FA 00      EX (SP), HL
44FB 00      RET
44FC 00      POP HL
44FD 00 00     LD (HL), 00
44FF 00 57 43 CALL 4357
4502 00      POP HL
4503 00      RET

```

Esta é a sub-rotina correspondente aos blocos 7 e 8 que havia sido citada no grupo 4. Como comentado na explicação do Registro DIRO, este conjunto de instruções aplica-se para os ramos verticais, da esquerda em que "O" sobe e da direita em que ele desce.

Analisando as instruções constata-se que, de início, o teclado é pesquisado pelo KScan e caso as teclas 5 ou 8 estejam sendo pressionadas dá-se um salto para um grupo de instruções correspondentes a cada tecla, e em caso contrário ocorre o retorno à chamada.

Após o salto, nos blocos correspondentes a cada tecla detecta-se se existe a possibilidade de mudar-se de pista, isto é, se na direção que "O" deve mover-se não existe a parede (figura de código 08). Se a parede lá estiver novamente, dá-se o retorno.

Em seguida é feita a alteração da pista, através da série LD A, (DIRO) – LD B, A – LD A, (PISTAO) – SUB A, B – LD (PISTAO), A e atinge-se uma seqüência de instruções que correspondem novamente aos blocos 14, 15, uma chamada a sub-rotina associada aos blocos 19 a 23, bloco 24 e finalmente bloco 5. Isto também foi elaborado deste modo pela facilidade de compreensão, e pelo fato do movimento entre pistas ser feito em duas etapas ou ciclos diferentes daqueles que o "O" faz ao longo da pista.

```

176563 00 00 00 00 00 70 00 00 11 07
176570 7E 07 00 00 00 00 00 00 10 00
176577 40 00 00 00 7E 00 00 00 00
176584 00 00 40 00 00 00 00 00 00
176591 44 40 40 40 00 00 00 00 00
176598 00 00 00 00 00 00 00 00 00
1765A5 7E 00 00 00 00 00 00 00 00
1765B2 00 00 00 00 44 40 47 00 00 00
1765C0 7E 00 00 00 00 00 00 00 00
1765C7 40 00 00 00 00 00 00 00 00
1765D4 00 00 40 00 00 00 00 00 00
1765E1 00 00 00 40 00 00 00 00 00
1765E8 00 00 00 00 00 00 00 00 00
1765F5 00 00 00 00 00 00 00 00 00

```

Grupo 10 – 17668 a 17786 (4504 a 457A em hexa)

```

4504 00      PUSH HL
4505 00 BB 02 CALL KSCAN
4508 70      LD A, H
4509 00      AND L
450A 00      POP HL
450B 7E CF   CP CF
450D 20 05   JR Z, 4514
450F 7E E7   CP EF
4511 20 20   JR Z, 4533
4513 00      RET
4514 11 21 00 LD DE, 33 (D)
4517 10      ADD HL, DE
4518 7E      LD A, (HL)

```

```

45510 CP 08
45510 JR NZ,45521
45510 AND A
45510 SBC HL,DE
45510 RET
45521 ADD HL,DE
45521 PUSH HL
45521 AND A
45521 SBC HL,DE
45524 LD A,(DIRO)
45525 LD B,A
45526 LD A,(PISTAO)
45527 ADD B,B
45528 LD A,(PISTAO),A
45530 JR 45530
45531 LD DE,33(D)
45532 AND A
45533 SBC HL,DE
45534 LD A,(HL)
45535 CP 08
45536 JR NZ,45540
45537 ADD HL,DE
45538 RET
45539 AND A
45541 SBC HL,DE
45542 PUSH HL
45543 ADD HL,DE
45544 LD A,(DIRO)
45545 LD B,A
45546 LD A,(PISTAO)
45547 SUB B
45548 LD A,(PISTAO),A
45549 LD A,(HL)
45551 CP 07
45552 JR NZ,4555B
45553 POP HL
45554 LD HL,455A5
45555 EX (SP),HL
45556 RET
45557 LD (HL),34
45558 PUSH HL
45559 CALL 45555
45561 LD A,(CONPAR8T)
45564 CP 34
45565 JR NZ,45573
45566 XOR A
45569 LD (CONPAR8T),A
45570 POP HL
45571 POP HL
45572 LD HL,455A5
45573 EX (SP),HL
45574 RET

```

```

4573 E1 POP HL
4574 36 00 LD (HL),00
4575 CD 57 43 CALL 4357
4576 E1 POP HL
4577 09 RET

```

Nova repetição, agora do grupo anterior, de número 9, porém aplicado aos movimentos horizontais, dos ramos de cima e de baixo, em que o "O" se desloca da direita para esquerda ou vice-versa.

```

17868 B5 CD B3 02 7C A5 E1 1137
17870 FE CF 20 00 7E FE 00 1000
17882 20 00 11 01 00 10 00 434
17889 FE 00 20 04 A7 00 00 704
17895 00 10 05 07 00 00 000
17700 44 40 47 0A 01 40 00 4000
17710 00 01 40 18 10 11 01 050
17717 00 07 00 00 7E 7E 00 074
17724 00 00 10 00 A7 00 00 746
17731 00 10 00 00 44 40 47 00 070
17738 01 40 00 00 01 40 7E 014
17745 7E 07 00 00 00 01 00 000
17750 00 00 40 00 00 04 00 00 000
17755 00 00 07 00 00 43 40 01 004
17773 E1 01 00 00 00 00 01 1146
17780 36 00 CD 57 43 E1 00 000

```

Grupo 11 — 17787 a 17828 (457B a 45A4 em hexa)

```

457B CD B3 02 CALL KSCAN
457E 3E FD LD A,FD
4580 5C H CP H
4581 00 RET NZ
4582 3A 7B 40 LD A,(BONUS)
4583 A7 AND A
4586 C0 RET Z
4587 3D DEC A
4588 32 7B 40 LD (BONUS),A
458B 0A 0C 40 LD HL,(O-FILE)
458E 11 90 01 LD DE,413(D)
4591 10 ADD HL,DE
4592 7E LD A,(HL)
4593 7E 9C CP 9C
4595 00 04 JR Z,459B
4597 3D DEC A
4598 77 LD (HL),A
4599 18 05 JR 45A1
459B 36 A5 LD (HL),A5

```

```

45000 2B      DEC HL
45001 7E      LD A, (HL)
45002 3D      DEC A
45003 77      LD (HL), A
45004 32 3E 40 LD (ACEL), A
45005 C9      RET

```

Outra sub-rotina, está ligada aos blocos 17 e 18. Como não poderia deixar de ser, ela começa pesquisando o teclado (CALL KScan) e dá-se o retorno, se o jogador não está pressionando o zero, ou se BONUS for zero.

O efeito de acelerar é obtido simplesmente pelo uso do retorno ao bloco 5, evitando o ramo 19, 20 etc., o que faz com que o asterisco se atrase. Os valores do BONUS Tela são diminuídos de uma unidade e dá-se o retorno.

```

17787 CD 8B 02 3E FD BC 00 1000
17788 3A 7B 40 A7 C8 3D 00 7000
17801 7E 40 2A 0C 40 11 00 4700
17808 01 10 7E 7E 0C 00 04 6000
17818 3D 77 18 06 06 10 20 4700
17822 7E 3D 77 32 3E 40 C8 6000

```

Grupo 12 — 17829 a 17873 (45A5 a 45D1 em hexa)

```

45A5 2A 3F 40      LD HL, (POBAST)
45A6 06 B4        LD (HL), B4
45A7 06 10        LD B, 10
45A8 0E 00        LD C, 00
45A9 06 00        DEC BC
45AA 70          LD A, B
45AB 05 01        OR C
45AC 00 01 7E    JR NZ, 45AE
45AD 00 17 00    LD (HL), 17
45AE 00 14 00    LD B, 14
45AF 00 00 00    LD C, 00
45B0 06 00        DEC BC
45B1 70          LD A, B
45B2 05 01        OR C
45B3 00 01 7E    JR NZ, 45B5
45B4 3A 43 40    LD A, (CONPOST)
45B5 77          LD (HL), A
45B6 2A 0C 40    LD HL, (D-FILE)
45B7 11 DE 01    LD DE, 478 (D)
45B8 10          ADD HL, DE
45B9 7E          LD A, (HL)

```

```

45CA 3D      DEC A
45CB 77      LD (HL), A
45CC FE 9C   CP 9C
45CD C8      RET Z
45CE 03 1B 43 JP 431B

```

Note-se que este grupo não se constitui numa sub-rotina. Ele está associado aos blocos 25 e 26 da figura 17.

Para começar é dado um efeito especial no encontro do asterisco com o "O" e, em seguida, é decrementado o registro Vidas. Atingindo o zero dá-se o retorno para o Basic, em caso contrário, um salto para o bloco 3.

```

17829 2A 3F 40 06 B4 06 10 4000
17836 0E 00 0B 70 B1 00 70 6000
17843 3E 17 06 14 0E 00 08 1000
17850 70 B1 00 70 3A 43 40 7000
17857 77 2A 0C 40 11 00 4770
17864 10 7E 3D 77 7E 9C C8 0441
17871 C3 1B 43      JP 431B

```

Grupo 13 — 17874 a 18004 (45D2 a 4654 em hexa)

```

45D2 2A 0C 40      LD HL, (D-FILE)
45D3 11 5D 01      LD DE, 349 (D)
45D4 10          ADD HL, DE
45D5 7E          LD A, (HL)
45D6 7E 00        CP 00
45D7 20 02        JR NZ, 45E0
45D8 3E 9C        LD A, 9C
45D9 0C          INC A
45DA 7E A6        CP A6
45DB 20 05        JR NZ, 45EA
45DC 30 9C        LD (HL), 9C
45DD 2B          DEC HL
45DE 10 EF        JR 45D9
45DF 77          LD (HL), A
45E0 C9          RET
45E1 7E A6        CP A6
45E2 20 05        JR NZ, 45EA
45E3 30 9C        LD (HL), 9C
45E4 2B          DEC HL
45E5 10 EF        JR 45D9
45E6 77          LD (HL), A
45E7 C9          RET
45E8 7E          PUSH AF
45E9 7E          PUSH HL
45EA CD D2 45      CALL 45D2
45EB 3A 7C 40    LD A, (407C)
45EC 3D          DEC A
45ED 20 06        JR Z, 45FD
45EE 32 7C 40    LD (CONT), A
45EF 71          POP HL

```

457FB	F1		POP	AF
457FC	00		RET	
457FD	11		POP	HL
457FE	F1		POP	AF
457FF	36	34	LD	(HL), 34
45801	55		PUSH	HL
45802	21	7B 40	LD	HL, BONUS
45805	7E		LD	A, (HL)
45808	D7		AND	A
45807	20	24	JR	Z, 462D
4580A	47		LD	B, A
4580B	21	00 10	LD	HL, 1000
4580D	2B		DEC	HL
4580E	70		LD	A, H
4580F	55		OR	L
45810	20	F8	JR	NZ, 460D
45813	2A	00 40	LD	HL, (D-FILE)
45814	11	9D 01	LD	DE, 413 (D)
45815	19		ADD	HL, DE
45816	7E		LD	A, (HL)
45817	7E	00	CP	90
45818	20	04	JR	Z, 4622
45819	3D		DEC	A
4581A	77		LD	(HL), A
4581B	20	00	JR	4628
4581C	20	00	LD	(HL), A5
4581D	20	00	DEC	HL
4581E	7E		LD	A, (HL)
4581F	3D		DEC	A
45820	77		LD	(HL), A
45821	CC	D2 45	CALL	46D2
45822	10	DD	DJNZ	460A
45823	11		POP	HL
45824	36	00	LD	(HL), 00
45825	36	00 40	LD	HL, (POSTAST)
45826	36	00	LD	(HL), 00
45827	20	00 40	LD	HL, (VELOC)
45828	70		LD	A, H
45829	20		OR	L
4582A	7E	00	CP	83
4582B	20	00	JR	Z, 4647
4582C	D7		AND	A
4582D	11	52 00	LD	DE, 62 (D)
4582E	ED	52	SBC	HL, DE
4582F	2D	3C 40	LD	(VELOC), HL
45830	2A	00 40	LD	HL, (D-FILE)
45831	11	9C 01	LD	DE, 412 (D)
45832	19		ADD	HL, DE
45833	36	A0 42	LD	(HL), A0
45834	21	B5	LD	HL, 42AB
45835	03		EX	(SP), HL
45836	09		RET	

Este grupo está ligado aos blocos 11, 12 e 13 da figura 17 e constitui uma sub-rotina como já citado.

Do endereço 45D2 a 45EC temos também uma sub-rotina, já conhecida, com a qual a contagem de pontos na tela é incrementada. Esta é utilizada imediatamente adiante, nas 10 instruções que se seguem e correspondem aos blocos 11 e 12.

As demais instruções vão compor o bloco 13. Primeiro o "O" é desenhado na posição final e o BONUS é testado. Sendo zero não há o que somar, e a velocidade é aumentada, desde que não tenha atingido o valor máximo fixado, através da redução do valor colocado no Registro VELOC. Se BONUS não for zero, seu valor é somado ao Ponto Tela, um a um, por sucessivas chamadas à sub-rotina citada no parágrafo anterior e, ao mesmo tempo, o Bonus Tela é reduzido igualmente.

Antes do salto ao bloco 2 o "O" e o asterisco são apagados.

17874	0A	00	40	11	00	00	01	00	00	00
17881	7E	00	00	00	00	00	01	00	00	00
17888	7E	00	00	00	00	00	01	00	00	00
17895	00	00	00	00	00	00	01	00	00	00
17902	00	00	00	00	00	00	01	00	00	00
17909	00	00	00	00	00	00	01	00	00	00
17916	00	00	00	00	00	00	01	00	00	00
17923	00	00	00	00	00	00	01	00	00	00
17930	00	00	00	00	00	00	01	00	00	00
17937	00	00	00	00	00	00	01	00	00	00
17944	10	7E	00	00	00	00	01	00	00	00
17951	77	10	00	00	00	00	01	00	00	00
17958	00	00	00	00	00	00	01	00	00	00
17965	00	00	00	00	00	00	01	00	00	00
17972	00	00	00	00	00	00	01	00	00	00
17979	00	00	00	00	00	00	01	00	00	00
17986	40	00	00	00	00	00	01	00	00	00
17993	41	00	00	00	00	00	01	00	00	00
18000	21	00	00	00	00	00	01	00	00	00

Grupo 14 – 18005 a 18530 (4655 a 4862 em hexa)

4655	0D	57 43	CALL	4357
4658	AF		XOR	A
465B	32	3E 40	LD	(ACEL), A
465C	2A	3F 40	LD	HL, (POSTAST)
465F	3A	43 40	LD	A, (CONPOST)
4662	77		LD	(HL), A
4665	3A	42 40	LD	A, (DIRAST)
4668	FE	01	CP	01
466B	20	4D	JR	NZ, 4667

4666A	11	21	00	LD	DE, 33(D)	4600E	FE	00	CP	03
4666D	19			ADD	HL, DE	46009	00	10	JR	NN, 466E
4666E	7E			LD	A, (HL)	46000	00	01	LD	A, 01
4666F	7E	08		LD	A, (HL)	46004	00	40	LD	(DIRAST), A
46671	00	00		CP	08	46007	00	41	LD	A, (DIRAST)
46673	00	00		JR	NZ, 467F	4600A	00		LD	A, (DIRAST)
46674	00	52		AND	A	4600B	00		INC	A
46676	00	52		SBC	HL, DE	46000	00	41	LD	(PISTAST), A
46677	00	04		INC	HL	4600B	11	22	LD	DE, 34(D)
46679	00	40	40	LD	A, 04	46001	10		ADD	HL, DE
4667C	00	50	40	LD	(DIRAST), A	46000	00	40	JP	4659
4667E	00	41	40	JP	4659	46007	00	04	CP	04
46680	47			LD	A, (PISTAST)	46000	00	40	JR	NN, 4734
46683	00	21	40	LD	B, A	46000	00		INC	HL
46686	00			LD	A, (PISTAST)	46000	00		LD	A, (HL)
46687	00	50	40	SUB	B	46000	11	21	LD	DE, 33(D)
4668A	00			JP	Z, 4659	46000	00	00	CP	08
4668B	00	15		PUSH	HL	46000	00	00	JR	NZ, 46FE
4668D	00			JR	C, 46A2	46000	00		DEC	HL
4668E	00	52		AND	A	46000	00	50	AND	A
46690	00			SBC	HL, DE	46000	00	07	SBC	HL, DE
46692	00			INC	HL	46000	00	40	LD	A, 07
46694	00			LD	A, (HL)	46000	00	50	LD	(DIRAST), A
46696	00	04		AND	A	46000	00	40	JP	4659
46699	00			JR	Z, 4699	46000	00	41	LD	A, (PISTAST)
4669B	00	50	40	POP	HL	46000	00	47	LD	B, A
4669D	00			JP	4659	46000	00	21	LD	A, (PISTAST)
4669F	00	00	40	POP	DE	46000	00	50	SUB	B
466A0	00	00	40	LD	A, 00	46000	00	10	JP	Z, 4659
466A2	00	50	40	LD	(DIRAST), A	46000	00	00	JR	C, 4720
466A4	00	50	40	JP	4659	46000	00	07	DEC	HL
466A6	00	52		AND	A	46000	00	50	AND	A
466A8	00			SBC	HL, DE	46000	00	50	SBC	HL, DE
466AA	00			DEC	HL	46000	00	7E	LD	A, (HL)
466AC	00			LD	A, (HL)	46000	00		AND	A
466AD	00			AND	A	46000	00	00	JR	Z, 4718
466AE	00	04		JR	Z, 46AE	46000	00	40	ADD	HL, DE
466B0	00			POP	HL	46000	00		INC	HL
466B2	00	50	40	JP	4659	46000	00	50	JP	4659
466B4	00			POP	DE	46000	00	00	LD	A, 00
466B6	00	00		LD	A, 00	46000	00	40	LD	(DIRAST), A
466B8	00	40	40	LD	(DIRAST), A	46000	00	50	JP	4659
466BA	00	50	40	JP	4659	46000	00		DEC	HL
466BC	00	02		CP	02	46000	00	10	ADD	HL, DE
466BE	00	13		JR	NZ, 46CE	46000	00	7E	LD	A, (HL)
466C0	00	01		LD	A, 01	46000	00		AND	A
466C2	00	42	40	LD	(DIRAST), A	46000	00	00	JR	Z, 4720
466C4	00	41	40	LD	A, (PISTAST)	46000	00	00	SBC	HL, DE
466C6	00			DEC	A	46000	00		INC	HL
466C8	00	41	40	LD	(PISTAST), A	46000	00	50	JP	4659
466CA	11	00	00	LD	DE, 32(D)	46000	00	00	LD	A, 00
466CB	00	50	40	ADD	HL, DE	46000	00	40	LD	(DIRAST), A
				JP	4659	46000	00	50	JP	4659

47334	7E	05	CP	05	4798B	03	59	48	JP	4859
47336	B0	10	JR	NZ,474D	4798E	10			ADD	HL,DE
47338	0E	04	LD	A,04	4798F	B0			INC	HL
4733A	02	42	LD	(DIRAST),A	47990	7E			LD	A,(HL)
4733D	3A	41	LD	A,(PISTAST)	479A1	A7			AND	A
4740	0C		INC	A	479A2	B0	04		JR	Z,47A8
4741	00	41	LD	(PISTAST),A	479A4	01			POP	HL
4744	11	20	LD	DE,32(D)	479A5	03	59	48	JP	4859
4747	A7		AND	A	479A8	D1			POP	DE
4748	ED	52	SBC	HL,DE	479A9	0E			LD	A,09
474A	03	50	JP	4859	479AB	02	42	40	LD	(DIRAST),A
474D	7E	06	CP	06	479AE	03	59	48	JP	4859
474F	B0	10	JR	NZ,4764	479B1	7E	08		CP	08
4751	0E	04	LD	A,04	479B3	09	15		JR	NZ,47CA
4753	00	42	LD	(DIRAST),A	479B5	0A	07		LD	A,07
4755	0A	41	LD	A,(PISTAST)	479B7	00	42	40	LD	(DIRAST),A
4759	0D		DEC	A	479BA	0A	41	40	LD	A,(PISTAST)
475A	00	41	LD	(PISTAST),A	479BD	0C			INC	A
475D	11	20	LD	DE,34(D)	479BE	02	41	40	LD	(PISTAST),A
4760	10		ADD	HL,DE	479C1	A7	20	00	LD	DE,34(D)
4761	03	59	JP	4859	479C4				AND	A
4764	7E	07	CP	07	479C5	ED	52		SBC	HL,DE
4766	B0	42	JR	NZ,47B1	479C7	03	59	48	JP	4859
4768	11	21	LD	DE,33(D)	479CA	7E	09		CP	09
476B	A7		AND	A	479CC	09	15		JR	NZ,47E3
476C	ED	52	SBC	HL,DE	479CE	0E	07		LD	A,07
476E	7E		LD	A,(HL)	479D0	02	42	40	LD	(DIRAST),A
476F	7E	08	CP	08	479D3	0A	41	40	LD	A,(PISTAST)
4771	B0	0A	JR	NZ,477D	479D6	0D			DEC	A
4773	10		ADD	HL,DE	479D7	02	41	40	LD	(PISTAST),A
4774	B0		DEC	HL	479DA	11	20	00	LD	DE,32(D)
4775	0E		LD	A,10(D)	479DD	A7			AND	A
4777	00	42	LD	(DIRAST),A	479DE	ED	52		SBC	HL,DE
477A	03	59	JP	4859	479E0	03	59	48	JP	4859
477D	0A	41	LD	A,(PISTAST)	479E3	7E	0A		CP	10(D)
4780	47		LD	B,A	479E5	B0	42		JR	NZ,4830
4781	0A	21	LD	A,(PISTAO)	479E7	B0			DEC	HL
4784	00		SUB	B	479E8	7E			LD	A,(HL)
4785	0A	59	JP	Z,4859	479E9	11	21	00	LD	DE,33(D)
4788	05		PUSH	HL	479EC	7E	06		CP	06
4789	08	13	JR	C,479E	479EE	09	0A		JR	NZ,47FA
478B	10		ADD	HL,DE	479F0	03			INC	HL
478C	B0		DEC	HL	479F1	10			ADD	HL,DE
478D	7E		LD	A,(HL)	479F2	0E	01		LD	A,01
478E	A7		AND	A	479F4	00	42	40	LD	(DIRAST),A
478F	B0	04	JR	Z,4795	479F7	03	59	48	JP	4859
4791	0E		POP	HL	479FA	0A	41	40	LD	A,(PISTAST)
4792	03	59	JP	4859	479FD	47			LD	B,A
4795	D1		POP	DE	479FE	0A	21	40	LD	A,(PISTAO)
4796	0E		LD	A,08	4801	00			SUB	B
4798	02	42	LD	(DIRAST),A	4802	0A	59	48	JP	Z,4859
					4805	08	14		JR	C,481B
					4807	B0			INC	HL

```

48008 10 ADD HL,DE
48009 7E LD A,(HL)
4800A A7 AND A
4800B 28 05 JR NZ,4813
4800C E0 52 SBC HL,DE
4800D EB DEC HL
4800E 03 59 48 JP 4859
4800F 3E 0B LD A,0B
48010 32 42 40 LD (DIRAST),A
48011 C3 59 48 JP 4859
48012 23 INC HL
48013 A7 AND A
48014 E0 52 SBC HL,DE
48015 7E LD A,(HL)
48016 A7 AND A
48017 28 05 JR NZ,4828
48018 19 ADD HL,DE
48019 EB DEC HL
4801A 03 59 48 JP 4859
4801B 3E 0C LD A,0C
4801C 32 42 40 LD (DIRAST),A
4801D C3 59 48 JP 4859
4801E 7E LD A,(HL)
4801F 28 05 JR NZ,4847
48020 19 ADD HL,DE
48021 EB DEC HL
48022 03 59 48 JP 4859
48023 3E 13 LD A,13
48024 3E 0A LD A,10(D)
48025 32 42 40 LD (DIRAST),A
48026 3A 41 40 LD A,(PISTAST)
48027 00 INC A
48028 32 41 40 LD (PISTAST),A
48029 11 20 00 LD DE,32(D)
4802A 49 ADD HL,DE
4802B C3 59 48 JP 4859
4802C 3E 0A LD A,10(D)
4802D 32 42 40 LD (DIRAST),A
4802E 3A 41 40 LD A,(PISTAST)
4802F 3D DEC A
48030 32 41 40 LD (PISTAST),A
48031 11 22 00 LD DE,34(D)
48032 A7 AND A
48033 E0 52 SBC HL,DE
48034 28 57 40 JR NZ,4844
48035 3E 04 LD A,(DIRAST),HL
48036 7E LD A,(HL)
48037 32 42 40 LD (COMPOST),A
48038 3D LD (HL),37
48039 00 RET

```

O bloco 19 é atendido pelas 5 primeiras instruções do grupo e onde, adicionalmente, o registro ACEL é feito igual a zero.

Os dois últimos bloco, respectivamente os 22 e 23 são satisfeitos pelas 5 últimas instruções do grupo. Todas as demais se relacionam com os blocos 20 e 21. As rotinas vão se sucedendo em função dos diferentes possíveis valores de DIRAST. Com este e pela utilização do PISTAST, o movimento do asterisco vai sendo monitorado.

```

180005 CD 57 43 DF 32 3E 40 710
180010 BA 37 46 3E 48 46 77 477
180019 3A 42 49 7E 01 28 40 552
180026 11 21 00 19 7E 7E 06 483
180033 00 00 00 00 00 00 00 7
180040 04 02 40 00 09 40 546
180047 3A 41 49 47 3D 01 40 413
180054 00 00 00 40 00 00 19 013
180061 D7 7E 00 00 00 00 00 4
180068 04 00 00 00 40 01 00 000
180075 03 00 40 40 03 00 40 000
180082 D7 7E 00 00 7E D7 00 000
180089 04 01 00 00 40 01 00 000
180096 00 00 40 40 00 00 40 000
180103 7E 00 00 10 3E 01 00 400
180110 4D 40 00 41 40 00 40 400
180117 41 40 00 41 40 00 40 400
180124 00 40 7E 00 00 10 00 411
180131 01 00 40 40 3D 41 40 388
180138 10 00 03 03 40 7E 04 571
180145 40 00 03 7E 00 04 00 546
180152 00 00 00 00 00 00 00 1
180159 00 00 00 00 00 00 00 0
180166 00 07 40 40 40 00 00 0
180173 40 3E 41 40 47 3D 01 401
180180 40 00 00 00 40 38 41 400
180187 20 00 D7 7E 00 00 00 000
180194 00 00 40 00 00 40 00 400
180201 00 00 40 40 00 00 40 541
180208 00 00 7E D7 00 00 00 441
180215 00 00 00 00 40 3E 00 541
180222 40 40 00 00 04 00 00 440
180229 40 00 41 40 00 00 41 400
180236 41 40 00 00 D7 7E 00 000
180243 00 00 00 00 40 40 00 000
180250 04 40 40 40 00 00 41 400
180257 00 04 40 40 00 00 00 000

```

Sem dúvida esta é a maior das sub-rotinas do programa. É a que comanda os movimentos do asterisco e corresponde aos blocos 19, 20, 21, 22 e 23 da figura 17.


```

4D57 36 00 LD (HL),00
4D59 10 FB DJNZ 4D55
4D5B 3D DEC A
4D5C CA 62 40 JP Z,4062
4D5E 23 INC HL
4D60 00 NOP
4D61 18 F1 JR 4D54

```

Este grupo é responsável pela montagem da tela inicial e pelo efeito de "SCROLL" e "ANTI-SCROLL".

Entre 4CBB a 4CDC está a rotina de impressão. Ela é monitorada pelo valor contido em B que vai indicar a partir de que linha o texto deve ser escrito na tela.

A chamada do Basic é para o endereço que vem a seguir, ou seja, para 4CDD. Entre este endereço e 4DAC estão as instruções necessárias para compor a tela inicial. O valor de B foi feito igual a 1 e usou-se a rotina citada.

Em 4D1D há a consulta ao teclado. Se S for pressionado dá-se o salto para 4D34, onde B é incrementado, desde que ainda não tenha atingido seu valor máximo, que é 10. Com D o salto é para 4D41, onde B é decrementado até o mínimo de 1. Em ambos os casos é sempre feita a chamada para o 4CBB, responsável pela escrita.

Finalmente, se C for a tecla pressionada, dá-se o salto para 4D4E onde, após a tela ser "apagada", dá-se o início ao jogo.

```

196643 21 55 48 11 1E 00 10 078
196650 10 FD E5 3E 10 00 00 600
196657 40 11 44 00 10 00 01 604
196664 01 1E 00 00 00 00 700
196671 13 13 13 00 10 70 00 501
196678 00 00 40 10 01 00 40 000
196685 01 00 00 00 00 00 00 001
196692 00 00 00 00 00 10 70 00 502
196699 11 17 00 00 10 00 1400
196706 00 00 10 00 00 10 77 000
196713 00 11 00 40 00 00 01 000
196720 1E 00 00 00 00 00 00 000
196727 00 40 01 00 01 00 1400
196734 00 10 00 70 00 00 70 047
196741 00 00 00 00 70 00 00 040
196748 00 77 00 04 00 00 00 007
196755 00 00 13 00 04 00 00 100
196762 10 00 00 00 00 00 00 700
196769 04 00 00 00 00 40 10 011
196776 04 00 01 00 00 00 00 007
196783 00 00 00 00 40 10 00 000
196790 01 00 00 40 00 10 00 400
196797 00 00 00 00 10 70 00 440
198004 00 00 40 00 00 10 71 000

```

Grupo 17 — 19811a 19858 (4D63 a 4D92 em hexa)

Aí estão os dizeres para o final do jogo.

```

19811A 00 00 00 00 00 00 00 000
19811B 00 00 00 00 00 00 00 000
19811C 00 00 00 00 00 00 00 000
19811D 00 00 00 00 00 00 00 000
19811E 00 00 00 00 00 00 00 000
19811F 00 00 00 00 00 00 00 000
198120 00 00 00 00 00 00 000
198121 00 00 00 00 00 00 000
198122 00 00 00 00 00 00 000
198123 00 00 00 00 00 00 000
198124 00 00 00 00 00 00 000
198125 00 00 00 00 00 00 000
198126 00 00 00 00 00 00 000
198127 00 00 00 00 00 00 000
198128 00 00 00 00 00 00 000
198129 00 00 00 00 00 00 000
19812A 00 00 00 00 00 00 000
19812B 00 00 00 00 00 00 000
19812C 00 00 00 00 00 00 000
19812D 00 00 00 00 00 00 000
19812E 00 00 00 00 00 00 000
19812F 00 00 00 00 00 00 000
198130 00 00 00 00 00 00 000
198131 00 00 00 00 00 00 000
198132 00 00 00 00 00 00 000
198133 00 00 00 00 00 00 000
198134 00 00 00 00 00 00 000
198135 00 00 00 00 00 00 000
198136 00 00 00 00 00 00 000
198137 00 00 00 00 00 00 000
198138 00 00 00 00 00 00 000
198139 00 00 00 00 00 00 000
19813A 00 00 00 00 00 00 000
19813B 00 00 00 00 00 00 000
19813C 00 00 00 00 00 00 000
19813D 00 00 00 00 00 00 000
19813E 00 00 00 00 00 00 000
19813F 00 00 00 00 00 00 000
198140 00 00 00 00 00 00 000
198141 00 00 00 00 00 00 000
198142 00 00 00 00 00 00 000
198143 00 00 00 00 00 00 000
198144 00 00 00 00 00 00 000
198145 00 00 00 00 00 00 000
198146 00 00 00 00 00 00 000
198147 00 00 00 00 00 00 000
198148 00 00 00 00 00 00 000
198149 00 00 00 00 00 00 000
19814A 00 00 00 00 00 00 000
19814B 00 00 00 00 00 00 000
19814C 00 00 00 00 00 00 000
19814D 00 00 00 00 00 00 000
19814E 00 00 00 00 00 00 000
19814F 00 00 00 00 00 00 000
198150 00 00 00 00 00 00 000
198151 00 00 00 00 00 00 000
198152 00 00 00 00 00 00 000
198153 00 00 00 00 00 00 000
198154 00 00 00 00 00 00 000
198155 00 00 00 00 00 00 000
198156 00 00 00 00 00 00 000
198157 00 00 00 00 00 00 000
198158 00 00 00 00 00 00 000
198159 00 00 00 00 00 00 000
19815A 00 00 00 00 00 00 000
19815B 00 00 00 00 00 00 000
19815C 00 00 00 00 00 00 000
19815D 00 00 00 00 00 00 000
19815E 00 00 00 00 00 00 000
19815F 00 00 00 00 00 00 000
198160 00 00 00 00 00 00 000
198161 00 00 00 00 00 00 000
198162 00 00 00 00 00 00 000
198163 00 00 00 00 00 00 000
198164 00 00 00 00 00 00 000
198165 00 00 00 00 00 00 000
198166 00 00 00 00 00 00 000
198167 00 00 00 00 00 00 000
198168 00 00 00 00 00 00 000
198169 00 00 00 00 00 00 000
19816A 00 00 00 00 00 00 000
19816B 00 00 00 00 00 00 000
19816C 00 00 00 00 00 00 000
19816D 00 00 00 00 00 00 000
19816E 00 00 00 00 00 00 000
19816F 00 00 00 00 00 00 000
198170 00 00 00 00 00 00 000
198171 00 00 00 00 00 00 000
198172 00 00 00 00 00 00 000
198173 00 00 00 00 00 00 000
198174 00 00 00 00 00 00 000
198175 00 00 00 00 00 00 000
198176 00 00 00 00 00 00 000
198177 00 00 00 00 00 00 000
198178 00 00 00 00 00 00 000
198179 00 00 00 00 00 00 000
19817A 00 00 00 00 00 00 000
19817B 00 00 00 00 00 00 000
19817C 00 00 00 00 00 00 000
19817D 00 00 00 00 00 00 000
19817E 00 00 00 00 00 00 000
19817F 00 00 00 00 00 00 000
198180 00 00 00 00 00 00 000
198181 00 00 00 00 00 00 000
198182 00 00 00 00 00 00 000
198183 00 00 00 00 00 00 000
198184 00 00 00 00 00 00 000
198185 00 00 00 00 00 00 000
198186 00 00 00 00 00 00 000
198187 00 00 00 00 00 00 000
198188 00 00 00 00 00 00 000
198189 00 00 00 00 00 00 000
19818A 00 00 00 00 00 00 000
19818B 00 00 00 00 00 00 000
19818C 00 00 00 00 00 00 000
19818D 00 00 00 00 00 00 000
19818E 00 00 00 00 00 00 000
19818F 00 00 00 00 00 00 000
198190 00 00 00 00 00 00 000
198191 00 00 00 00 00 00 000
198192 00 00 00 00 00 00 000

```

Grupo 18 — 19859 a 19922 (4D93 a 4DD2 em hexa)

```

4D93 2A 00 40 LD HL,(D-FILE)
4D94 11 05 00 LD DE,0005
4D95 10 ADD HL,DE
4D96 EB EX DE,HL
4D97 21 63 4D LD HL,4D63
4D98 01 10 00 LD BC,24(D)
4D99 ED LDIR
4D9A 00 NOP
4D9B EB EX DE,HL
4D9C 01 DF 02 LD BC,735(D)
4D9D 00 ADD HL,BC
4D9E EB EX DE,HL
4D9F 01 18 00 LD BC,24(D)
4DA0 ED LDIR
4DA1 00 CALL KSCAN
4DA2 CB 4D BIT 1,L
4DA3 20 0A JR Z,4D00
4DA4 CB 6D BIT 5,L
4DA5 20 00 JR Z,4D07
4DA6 CB 7D BIT 7,L
4DA7 20 10 JR Z,4D0E
4DA8 10 0F JR 4DAF
4DA9 CB 54 BIT 2,H
4DAA 20 EB JR NZ,4DAF
4DAB 03 02 4D JP 4062
4DAC CB 5C BIT 3,H
4DAD 20 E4 JR NZ,4DAF
4DAE 03 DD 4C JP 4CDD
4DAF CB 64 BIT 4,H
4DB0 20 DD JR NZ,4DAF
4DB1 07 RST 00

```


APÊNDICE B — MONITOR

Em todos os capítulos deste volume os programas em linguagem de máquina são listados de duas formas:

1) **Desassemblados:** logo após o sub-título de cada grupo estão listados os comandos em três colunas: na primeira o endereço inicial do comando (em hexadecimal), na segunda a seqüência de bytes do comando em códigos também hexadecimais e na terceira os mnemônicos. A análise deste tipo de listagem é que traz maior proveito didático.

2) **Heptassintáticos:** esta é a forma mais rápida de se introduzir os códigos e a mais garantida em relação a erros de digitação. Em compensação a digitação é feita de maneira cega pois não aparecem os mnemônicos.

Para digitar o programa desta forma devemos antes digitar o monitor listado a seguir (quem tem apenas 3 K de RAM deve obter provisoriamente uma expansão até completar a digitação).

```
900 FAST
1000 LET A#="00 00 CD 23 0F 01 0
6 00 2A 00 7F E5 09 44 4D 2A 29
40 09 22 29 40 21 0C 40 3E 09 5E
23 56 D5 EB 09 EB 72 2B 73 23 2
3 3D 28 03 D1 18 EE E1 E5 01 7C
40 A7 ED 42 44 4D E1 ED B8 21 7D
40 36 00 23 36 00 23 C1 03 03 "
1 23 70 23 36 EA 0B 0B 23 11 01
00 EB 19 EB 36 00 ED B0 36 75 31
CD 2B 0F C8 "
1100 FOR I=1 TO LEN A# STEP 3
1200 POKE (32512+INT (I/3)),16#C
ODE A#(I)+CODE A#(I+1)-476
1300 NEXT I
1400 PRINT "N# BYTES ?"
1500 INPUT N
1600 POKE 32512,N-256*INT (N/256
)
1700 POKE 32513,INT (N/256)
1800 RAND USR 32514
1900 CLS
```

```
2000 SLOW
2100 PRINT "ENDERECO INICIAL =
2200 INPUT I
2250 PRINT I
2300 PRINT "ENDERECO FINAL="
2400 INPUT F
2500 PRINT F
2600 PAUSE 200
2700 CLS
2800 FOR M=I TO F STEP 7
2900 PRINT AT 21,0) "-----
-----"
3000 SCROLL
3100 LET T=0
3200 DIM A#(32)
3300 LET A#( TO 5)=STR# M
3400 PRINT AT 21,0)M;" ";
3500 FOR J=1 TO 21 STEP 3
3600 IF (M+INT (J/3))>F THEN GOT
O 5000
3700 FOR K=0 TO 1
3800 IF INKEY#<" THEN GOTO 380
0
3900 IF INKEY#="" THEN GOTO 3900
4000 LET P=CODE INKEY#
4100 IF P<28 OR P>43 THEN GOTO 3
800
4200 LET A#(J+K+6)=CHR# P
4300 PRINT CHR# P;
4400 NEXT K
4500 LET N=16#CODE A#(J+6)+CODE
A#(J+7)-476
4600 LET T=T+N
4700 POKE (M+INT (J/3)),N
4800 PRINT " ";
4900 NEXT J
5000 PRINT TAB 27;T
5100 LET A#(26 TO )=STR# T
5200 IF INKEY#="S" THEN GOTO 560
0
5300 IF INKEY#<>"N" THEN GOTO 52
00
5400 PRINT AT 21,0) "
5500 GOTO 3100
5600 IF (M+INT (J/3))>F THEN STO
P
5700 PRINT AT 20,0)A#
5800 NEXT M
```

Ao digitar RUN o monitor apresenta qual o número de bytes que a linha REM deve ter para que nela sejam introduzidos os códigos. O leitor deve

procurar este número de bytes no texto, no programinha em basic que acompanha o programa principal. Por exemplo, na página 33 o programinha em basic mostra que a linha REM deve ter 714 bytes.

Após introduzir este número, o monitor se encarrega de criar a linha REM com o número de bytes pedido (preenchidos com NOPs).

A seguir o programa pede o endereço inicial e final do grupo que está sendo digitado. Estes endereços, em decimal, deve ser procurado no subtítulo de cada grupo.

O monitor apresenta na tela as mesmas 9 colunas da listagem: a primeira contém o endereço do primeiro byte de linha, a seguir temos sete colunas contendo seqüencialmente sete bytes em hexadecimal e, finalmente, na última coluna temos a soma sintática dos sete bytes da linha. Se a soma coincidir com a listada no livro, isto significa que a linha foi digitada corretamente e o leitor deverá teclar S; os bytes serão introduzidos na linha REM e a próxima linha é iniciada. Se a soma não conferir, deve-se teclar N e sua digitação deve ser refeita.

Se a digitação do grupo seguinte não for feita no mesmo dia, o programa deve ser gravado. Ao ser recolocado no gravador devemos digitar:

GOTO 2000

e os endereços inicial e final do grupo seguintes serão pedidos.

Ao terminar o programa tome o seguinte cuidado: digite antes as linhas de basic do programinha sugerido no capítulo e só depois apague o monitor linha por linha. Isto se deve ao fato da linha REM ser demasiadamente longa e se ficar sozinha faz o sistema operacional entrar em "tilt".

APÊNDICE C

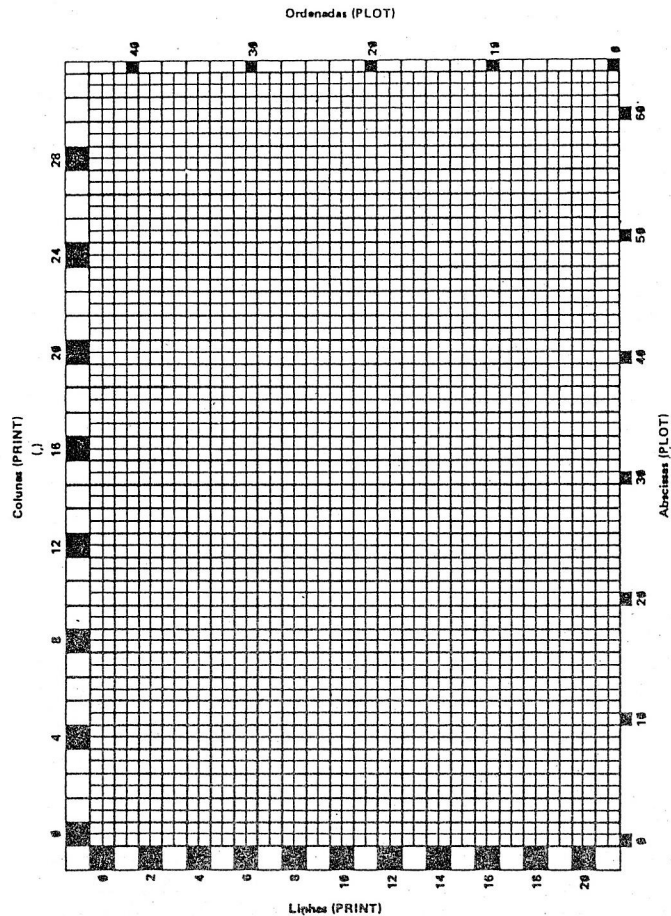
MAPA DA ROM

0000	— Inicialização do Sistema
0008	— Manipulação de erros
0010	— Rotina que imprime um caractere
0018	— Carrega no acumulador o byte apontado pela variável no endereço 16406
0020	— Carrega no acumulador o próximo byte apontado pela variável no endereço 16406
0028	— Rotina de Cálculo
0030	— Incrementa a área de variáveis com os números de byte em BC
0038	— Rotina de interrupção para mostrar na tela uma linha
0066	— Rotina de interrupção para mostrar a tela no modo SLOW
007E	— Tabelas dos caracteres normas do teclado
00A5	— Tabela dos caracteres SHIFT do teclado
00CC	— Tabela das funções do BASIC
00F3	— Tabela dos caracteres gráficos
0111	— Tabela das palavras-chaves do BASIC
01FC	— Rotina de atualização dos comandos SAVE e LOAD
0207	— Rotina de determinação da velocidade (SLOW ou FAST)
0229	— Rotina principal do display
0292	— Rotina do display no modo SLOW
02B5	— Rotina do display no modo FAST
02BB	— Rotina de varredura do teclado
02E7	— Rotina de RESET do SCL
02F4	— Comando SAVE
0340	— Comando LOAD
03A2	— Teste de BREAK do comando LOAD
03C3	— Comando NEW
0419	— Rotina de edição das linhas de programação
0454	— Rotina do cursor
04B2	— Rotina de execução do programa em BASIC
052B	— Rotina de construção do Sistema E-LINE
05C4	— Rotina de ordenação da edição
063E	— Rotina principal de edição

072C — Comando LLIST
 0730 — Comando LIST
 0745 — Rotina de impressão de uma linha BASIC
 07BD — Rotina de decodificação do teclado
 07F1 — Rotina de impressão de um caractere
 0808 — Rotina de impressão de um caractere no vídeo
 0851 — Coloca um caractere no buffer da impressora
 0869 — Comando COPY
 08F5 — Teste dos parâmetros de PRINT AT
 0918 — Rotina de expansão do display
 094B — Rotina de impressão das palavras-chaves do BASIC
 09AD — Rotina de organização das variáveis
 09D8 — Determinação do endereço de uma linha de programa
 0A2A — Comando CLS
 0A98 — Rotina de impressão do número da linha
 0ACB — Comando LPRINT
 0ACF — Comando PRINT
 0B6B — Rotina de impressão de uma string
 0BAF — Comando PLOT e UNPLOT
 0C0E — Comando SCROLL
 0C29 — Tabela de sintaxe dos comandos
 0CDC — Comando STOP
 0DAB — Comando IF
 0DB9 — Comando FOR
 0E2E — Comando NEXT
 0E6C — Comando RAND
 0E7C — Comando CONT
 0E81 — Comando GOTO
 0E92 — Comando POKE
 0ED8 — Comando RETURN
 0F23 — Comando FAST
 0F2B — Comando SLOW
 0F32 — Comando PAUSE
 0F46 — Teste de BREAK do comando SAVE
 1321 — Comando LET
 1409 — Comando DIM
 149A — Comando CLEAR
 151D — Arquiva o acumulador do Stack do calculador
 1520 — Arquiva o par BC no Stack do calculador
 158A — Rotina de manipulação dos cálculos de ponto flutuante
 174C — Rotina de subtração para números de 5 bytes
 1755 — Rotina de adição para números de 5 bytes

1706 — Rotina de multiplicação para números de 5 bytes
 1882 — Rotina de divisão para números de 5 bytes
 1815 — Tabela das funções
 199D — Calculador de ponto flutuante
 1E00 — Tabela de definição dos caracteres
 4000 — Início da RAM
 4082 — Início do programa em BASIC
 DFILE — Início do arquivo de tela
 VARS — Início da região de variáveis
 ELINE — Linha sendo digitada + espaço de trabalho
 PILFUN — Início da pilha de cálculo
 PILFIM — Início da pilha de cálculo
 SP — Início da pilha para endereçar sub-rotinas
 RAMTOP — Início do espaço que pode ser reservado para rotinas em linguagem de máquina

APÊNDICE D — TABELA



NÍVEL #	NÍVEL 1	NÍVEL 2A	NÍVEL 2B	NÍVEL 3	NÍVEL 4
CRIANÇAS	ADULTOS E ADOLESC.	ADULTOS E ADOLESC.	ADOLESC. JOGOS	ADULTOS E ADOLESC.	ADULTOS E ADOLESC.
10 h	20 h	20 h	20 h	40 h	40 h
INICIAÇÃO	INICIAÇÃO	APROF. EM APLICAÇÕES ADM.	APLICAÇÕES EM JOGOS	LINGUAGEM DE MÁQUINA ASSEMBLY Z80	APROF. DE LINGUAGEM DE MÁQUINA



Av. Brig. Faria Lima, 1.451 - 3º - Cj. 31
Tel.: 813-4555 - CEP 01451 - São Paulo

LIBERTE O GENIO DO SEU MICRO



CURSO DE ASSEMBLY
O seu micro de lógica SINCLAIR (RINGO, CP-200, TK82/83/85, AS-1000, etc.) bem como os compatíveis com TRS-80, têm um microprocessador Z-80.

Aprendendo Linguagem de Máquina (ASSEMBLY Z 80) você poderá comandar diretamente o microprocessador

estruturando programas muitas vezes mais rápidos do que em BASIC e gastando muito menos memória.

Você poderá aprender ASSEMBLY como fez o FLAVIO ROSSINI (autor do LINGUAGEM DE MÁQUINA PARA O TK - Ed. MODERNA) assistindo às aulas do professor FRANCISCO A.S. DE OLIVEIRA.

Os cursos LM I (40 horas) e LM II (40 horas) são no NÚCLEO DE ORIENTAÇÃO DE ESTUDOS - Av. Brig. Faria Lima, 1451 conj. 31, em vários horários, inclusive aos sábados. Para informações ligue para (011) 813-4555.

NÃO SE DEIXE LEVAR!

Saiba escolher o melhor material para seu micro. Na URANIA, dirigida pelo professor Pierluigi Piazzì você encontra novidades inteligentes para o seu computador:

Software Laser: Jogos inteligentes e de ação para divertimento de pessoas inteligentes.

Software Didático: Use o seu micro como um eficientíssimo professor para rever e aprender línguas, matemática, ciências, etc., na velocidade que você determinar.

Livros para seu micro: Obras originais editadas pela própria URANIA por autores NACIONAIS escritos para o USUÁRIO BRASILEIRO.

Cursos: Basic (elementar e avançado) linguagem de máquina Z-80 (iniciantes e aprofundamento) sob a orientação de Flavio Rossini, no Núcleo de Orientação de Estudos.



Faça-nos uma visita ou envie cupom anexo (ou xerox) para

Gostaria de receber gratuitamente o boletim informativo da Urania.

Nome: _____

Endereço: _____

CEP: _____ Cidade: _____

Estado: _____ Data: _____ Tel.: _____

Assinatura: _____

URANIA Publicações e Assessoria
Av. Brigadeiro Faria Lima, 1451
3º cj. 31 - CEP 01451 - São Paulo
Tel.: (011) 813.4555

Favor preencher em letra de forma

Se quiser, transmita seus dados por telefone para nossa secretaria eletrônica, fora do horário comercial.

PARA: SINCLAIR ZX-81 RINGO CP-200 TK 82/83/85

- COLEÇÃO DE PROGRAMAS Vol 3
RICARDO DE FAYETTI SIQUEIRA
- TABELA DE MNEMÔNICOS Z 80
- JOGOS EM LINGUAGEM DE MÁQUINA
Vol 1
- JOGOS EM LINGUAGEM DE MÁQUINA
Vol 2
- USANDO LINGUAGEM DE MÁQUINA
MARIO SCHAEFFER
- CRIANDO EM LINGUAGEM DE MÁQUINA
SAMUEL EJCHEL

URANIA

PUBLICAÇÕES E ASSESSORIA PEDAGÓGICA LTDA.

Av. Brig. Faria Lima, 1451 - conj. 31

01451 - Jd. Paulista - São Paulo-SP

tel. (011)813-4555



impresso na
planimpres gráfica e editora
rua anhaia, 247 - s.p.