

ARTE
DE
VIVER.

JACQUES DECONCHAT

102 PROGRAMAS DE JOGOS PARA O AMSTRAD

APRENDA A PROGRAMAR, DIVERTINDO-SE



PUBLICAÇÕES EUROPA-AMÉRICA

**102 PROGRAMAS
DE JOGOS PARA O *AMSTRAD***

Aprenda a programar, divertindo-se

Obras publicadas nesta coleção:

- 1 — *111 Receitas com Ovos*, Etelvina Lopes de Almeida
- 2 — *O Livro do Casal*, Pierre-Marie Brémond
- 3 — *Aprenda a Fotografar*, Antoine Desilets
- 4 — *Guia da Interpretação dos Sonhos*, Louis Stanké
- 5 — *A Arte de bem Receber*, Marguerite du Coffre
- 6 — *Guia do Comportamento Sexual*, Dubois-Caballero
- 7 — *Como Reparar Avarias na Estrada — Manual de Todo o Automobilista*, Miguel de Castro Vicente
- 8 — *Guia Prático e Completo da Costura*, Lise Chartier
- 9 — *Guia Íntimo das Relações Sexuais*, Pierre Valinief
- 10 — *Guia dos Jovens — A Vida e o Amor*, Dr. Benjamin Spock
- 11 — *111 Receitas de Tapas e Entradas*, Etelvina Lopes de Almeida
- 12 — *Guia da Futura Mãe durante a Gravidez*, Dr. José M.^a Carrera
- 13 — *Como Suprimir as Suas Dores com a Simples Pressão de Um Dedo*, Dr. Roger Dalet
- 14 — *O Livro das Boas Maneiras*, Marcelle Fortin-Jacques
- 15 — *111 Receitas de Frango*, Jacky Davin
- 16 — *Doenças Transmítidas pelas Relações Sexuais*, Dr. Lionel Gendron
- 17 — *Hatha-Yoga*, Suzanne Piuze
- 18 — *Os Segredos do Amor Táctil*, A. Vignati e O. Caballero
- 19 — *Como Socorrer o Seu Filho*, Marie Hermand
- 20 — *Métodos Anticonceptivos e Planeamento Familiar*, Santiago Dexeus e Margarita Riviere
- 21 — *A Técnica da Fotografia*, Antoine Desilets
- 22 — *Amor, Sexo e Astrologia*, Teri King
- 23 — *Como Vencer a Timidez*, François Suzzarini
- 24 — *111 Receitas de Coelho*, Anne Vernon
- 25 — *Os Remédios da Avozinha*, Barbara Kamir
- 26 — *A Mulher depois dos 40 Anos*, Santiago Dexeus e Teresa Pámies
- 27 — *Viver bem depois dos 50 Anos*, Dr. Hugues Destrem
- 28 — *Conservas, Comotas e Xaropes*, Maria Emilia Abreu Semedo
- 29 — *Como Proteger a Saúde e a Beleza com a Simples Pressão de Um Dedo*, Dr. Roger Dalet
- 30 — *111 Receitas para Emagrecer*, Dr. Jean-Paul Ostigny
- 31 — *Eu... Tu... e os Outros*, Anna Boyer e Isabelle Nicolas
- 32 — *O Rosto, Espelho do Carácter*, Louis Stanke
- 33 — *O Seu Aquário de Peixes Tropicais*, Brian Ward
- 34 — *Pílula — A Solução Mortal*, Dr. Dominique Chatain
- 35 — *O Seu Futuro nas Caritas*, Louis Stanké
- 36 — *111 Refeições Naturistas*, Maria Cândida de Albuquerque Cardoso
- 37 — *A Bíblia do Bridge*, Claude Derwy
- 38 — *A Congelação dos Alimentos*, Pamela Dotter
- 39 — *A Celulite*, Gerald J. Leonard
- 40 — *Guia Sexual da Moça Moderna*, Wardel B. Pomeroy
- 41 — *101 Conselhos aos Diabéticos*, Prof. Georges Tchobroutsky
- 42 — *A Beleza pela Saúde*, Dr. Pierre Fournier
- 43 — *Plantas de Interior*, Brian Ward e Tom Wellsted
- 44 — *111 Receitas de Cozinha Africana*, Maria de Lourdes Chantre
- 45 — *Saber Maquilhar-Se*, Josette Ghedin
- 46 — *111 Receitas de Massas*, Anne Vernon
- 47 — *Alimentação Natural*, José Lyon de Castro
- 48 — *A Mulher e o Sexo*, Dr. Lionel Gendron
- 49 — *A Menopausa*, Dr. Lionel Gendron
- 50 — *111 Receitas de Arroz*, Deda Frachon
- 51 — *Trate o Seu Cão, o Seu Gato, os Seus Pássaros com a Simples Pressão de Um Dedo*, Roger Dalet
- 52 — *A Chave da Longevidade*, Dr. Hugues Destrem
- 53 — *Tudo sobre Acupuntura*, Dr. Jean Vibes
- 54 — *101 Respostas sobre a Depressão*, Dr.^a Marie Claude Navikoff e Dr. Jean Pierre Olié
- 55 — *111 Receitas para Painel de Pressão*, Janet Warren
- 56 — *Como Vencer as Enxaquecas*, Dr. Claude Loisy e Dr. Sidney Pélage
- 57 — *Como Viajar de Avião sem Ter Medo*, Afra Botteri/Cécile Gateff
- 58 — *Tempo Que Mata, Tempo Que Cura*, Dr. Fernand Attali
- 59 — *Como Manter a Virilidade*, Paul Stanley
- 60 — *A Alimentação da Criança*, Louise Lambert-Lagacé
- 61 — *O Sexo e o Amor no Casamento*, Bernard Delon e Germaine Lanoé
- 62 — *Conheça-Se a Si Próprio — I*
- 63 — *Conheça-Se a Si Próprio — II*
- 64 — *Tênis Prático — Técnica — Conselhos — Campos*, Christian Collin
- 65 — *101 Segredos da Medicina Natural*, Dr. Péron-Antrêt

- 66 — *Manual de Protecção contra o Crime*, Ira A. Lipman
- 67 — *111 Receitas de Caça*, Ana Isabel de Castro
- 68 — *Manual Médico da Família*, Dr. David Kellett Carding
- 69 — *A Cozinha Astrológica*, Marie Gebert e Monique Maine
- 70 — *Doenças de Cães e Gatos Transmissíveis a Crianças*, Silva Leitão
- 71 — *Manual de Sobrevivência na Situação de Guerra Nuclear — Como Viver durante e após um Ataque Nuclear*, Barry Popkess
- 72 — *Conheça os Computadores*, John Shelley
- 73 — *Como Tratar o Seu Filho com a Simples Pressão de Um Dedo*, Dr. Tan Poh Choon
- 74 — *Tudo sobre Astrologia*, H.-M. de Campigny
- 75 — *Horóscopos Árabes*, Paula Delsol
- 76 — *Horóscopos Chineses*, Paula Delsol
- 77 — *Aventuras com o Spectrum*, Tony Bridge e Roy Carnell
- 78 — *Enciclopédia dos Pontos Que Curam*, Dr. Roger Dalet
- 79 — *A Dianética*, L. Ron Hubbard
- 80 — *Auto-Análise*, L. Ron Hubbard
- 81 — *Como Vencer no Trabalho e na Vida*, L. Ron Hubbard
- 82 — *Como Planear e Construir a Sua Lareira*, Margaret e Wilbur F. Eastman Jr.
- 83 — *As Previsões Astrológicas para 1985*, Catherine Aubier
- 84 — *Faça Você Mesmo — I — Alvenaria, Telhados, Carpintaria*
- 85 — *Faça Você Mesmo — II — Electricidade, Canalização, Pintura, Vidraria*
- 86 — *Faça Você Mesmo — III — Revestimentos, Isolamentos, Refrigeração*
- 87 — *8 Exercícios para Um Corpo Perfeito*, Sheri Blair
- 88 — *Guia Prático da Sorte*, Cécile Donner e Jean-Luc Caradeau
- 89 — *Programação Prática para o Spectrum em Linguagem Máquina*, Steve Webb
- 90 — *Aplicações Domésticas no seu Microcomputador*, Mike Grace
- 91 — *Como Fazer Amor com a Simples Pressão de Um Dedo... e não só*, Hsuan Tsai Su-Nu
- 92 — *111 Receitas de Cozinha Indiana*
- 93 — *Como Fazer Amor com Um Homem*, Régine Dumay
- 94 — *Terapêutica Biológica*, Adriano de Oliveira
- 95 — *Inteligência Artificial no Spectrum*, Keith e Steven Brain
- 96 — *111 Receitas de Cozinha Chinesa*
- 97 — *O Spectrum Funcional*, David Lawrence
- 98 — *Domine o Seu ZX Microdrive*, Andrew Pennell
- 99 — *Desenvolvimento de Aplicações no Sinclair QL — Ideias práticas para utilizações domésticas e em negócios*, Mike Grace
- 100 — *Truques de Ilusionismo*, Araújo
- 101 — *Receitas de Refeições para Bebês, Crianças e Jovens — Crescer com Saúde*, Catherine Lewis
- 102 — *Comer Bem e Barato com Saúde — Dieta Para Evitar o Cancro*, Carmel Berman Reingold
- 103 — *A Inteligência Artificial no Sinclair QL — Faça o Seu Micro Pensar*, Keith e Steven Brain
- 104 — *Manual de Defesa Pessoal*, Prof. J. A. Fonseca Gaspar
- 105 — *Ervas — Aplicações Culinárias Decorativas e Cosméticas*, Jack Harvey
- 106 — *A Conservação de Alimentos*, Pamela Dotter
- 107 — *Jogos de Aventuras para o Sinclair QL — O manual do microaventureiro*
- 108 — *Manual de Sobrevivência*
- 109 — *Quill, Easel, Archive e Abacus no Sinclair QL — Como integrar os quatro «pacotes» de software da Psion*, Alison McCallum-Varey
- 110 — *Problemas de Xadrez — História-Antologia-Iniciação Técnica-Regras Básicas do Jogo*, Rui C. Nascimento
- 111 — *102 Programas de Jogos para o Amstrad*, Jacques Deconchat

Título original: 102 Programmes pour Amstrad

Tradução de Costa Vieira

Capa: estúdios P. E. A.

© Éditions du P. S. I., 1985

*Direitos reservados por
Publicações Europa-América, Lda.*

Nenhuma parte desta publicação pode ser reproduzida ou transmitida por qualquer forma ou por qualquer processo, electrónico, mecânico ou fotográfico, incluindo fotocópia, xerocópia ou gravação, sem autorização prévia e escrita do editor. Exceptua-se naturalmente a transcrição de pequenos textos ou passagens para apresentação ou crítica do livro. Esta excepção não deve de modo nenhum ser interpretada como sendo extensiva à transcrição de textos em recolhas antológicas ou similares donde resulte prejuízo para o interesse pela obra. Os transgressores são passíveis de procedimento judicial

Editor: Francisco Lyon de Castro

PUBLICAÇÕES EUROPA-AMÉRICA, LDA.
*Apartado 8
2726 MEM MARTINS CODEX
PORTUGAL*

Edição n.º 133111/4142

*Execução técnica:
Gráfica Europam, Lda.,
Mira-Sintra — Mem Martins*

Índice

Prefácio	Pág. 11
Organização da obra	12

Nível 1

Instruções elementares	15
O número misterioso	22
Cubo	24
Quente, frio	26
Os fósforos	28
Fósforos para vários	30
Calendário perpétuo	32
Batalha naval	34
Golfe	36
Solfejo	38
O jogo do 21	40
Par-ímpar	42
Adivinha um número	44
O jogo de Witthof	46
«Tabelas»	48
«Jackpot»	50
Roleta	52
Corredores	54
Tabuleiro de xadrez	56
Jogo da fava	58
Jogo da glória aleatório	60

Nível 2

Dimensionamento de um quadro — Busca de um elemento num quadro e numa cadeia — A instrução DATA	63
Dado viciado	65
Um dado	67
Visões	69
421	71
O electricista maluco	73

A corrida ao 20	75
Música	77
Póquer	79
O jogo do salta-carneiro	81
Letras	83
Maquinapalavra	85
Espião	87
Totoloto	89
Tiragem de um dado viciado	91
O jogo de Marienbad	93
Compõe	95
Dia da semana	97
Habilidade com cartas	99
O indesejável	101
O jogo da vida	103

Nível 3

Manipulação das cadeias e gráficos de alta resolução	107
Losangos	109
Maior, menor	111
Ordem	113
O enforcado	115
A bisca	117
Um cálculo quebra-cabeças	119
Enforcado invertido	121
«Nicômaco»	123
«Mastermind»	125
Idem	127
Rainhas no tabuleiro de xadrez	129
Palavras	131
Permutas	133
Bridge	135
Hanói	137
Corrida de cavalos	139
Engatem os vagões	141
O lobo e os cordeiros	143

Nível 4

As técnicas de animação elementar	147
Tiragem de dados	149
A bomba	151
Chapéus	153
Desenho de dados	155
O macaco	157
O jogo do 15	159
Teclado	161
Destreza	163
Anagramas	165
«Zombies»	167
Quadro mágico	169
«Decómaco»	171
Evasão	173
Repetições	175
«Mateus»	177
Através	179
Calculadora	181
Caça ao submarino	184

Nível 5

As técnicas particulares do AMSTRAD	188
Relógio 1	189
Relógio 2	191
Combate aéreo	193
Cartaz publicitário	195
O jogo do míssil	197
Submarino	199
Jogo do galo	201
«Smokey»	203
Corrida de carros	205
Perigo espacial	207
«Diago»	209
Ovos	211
Quebra-cabeças com letras	213
Alvos	215
Gotas	217
Obstáculos	219

Capturas no espaço	221
Anomalia	223
Vários em um	225
Percurso do cavalo	227
«Othello»	229
Damas	232
Quatro a quatro	235
Xadrez	238
Muro de tijolos	241
O lagarto de Alice	243

Prefácio

Encontrará neste livro mais de cem programas (essencialmente jogos) redigidos em Basic AMSTRAD. A maioria destes programas, aliás, serão utilizáveis sem grandes modificações noutros computadores que utilizem um Basic Microsoft.

Os programas propostos são muito curtos, e a ordem pela qual são apresentados não é, evidentemente, arbitrária: trata-se de iniciar o leitor, através da realização e da utilização de jogos de boa qualidade, no conhecimento e no domínio progressivo desse extraordinário utensílio que é o computador pessoal.

A brevidade dos programas propostos permitirá teclá-los facilmente, sem risco de erro. Poderá, evidentemente, utilizá-los como base para a realização de jogos mais elaborados.

Os jogos são abundantemente comentados para o ajudar a compreender o seu funcionamento, e as páginas de apresentação permitir-lhe-ão encontrar facilmente as características próprias das instruções Basic que eles utilizam.

O nível 5 dá uma ideia quanto à utilização de certas técnicas um tanto específicas. Estas técnicas poderão ser retomadas com proveito para melhorar os programas precedentes.

Encontrará, no fim de cada texto de apresentação, algumas sugestões para possíveis melhoramentos, ou modificações, que poderão tornar mais atractivo, de mais alta qualidade ou mais completo o jogo proposto.

Organização da obra

Os jogos são classificados por ordem de dificuldade crescente, de acordo com cinco níveis:

- **O nível 1** permite o estudo das instruções elementares da linguagem Basic: **LET** (aqui opcional), **PRINT**, **INPUT**, **IF... THEN**, **FOR... NEXT**, **GOTO**, **GOSUB**, **ON... GOTO**, **ON... GOSUB**, **WHILE... WEND**.
- **No nível 2** utilizaremos as instruções de confecção e de manipulação dos elementos de um quadro e de busca de um elemento numa cadeia: **DIM**, **R(I)**, **MID\$, READ**, **DATA**.
- **No nível 3** utilizaremos a panóplia completa das instruções de tratamento das cadeias de caracteres: **MID\$, RIGHT\$, LEFT\$**.
- **O nível 4** apela para uma instrução muito particular, que facilita a realização de jogos interactivos: **INKEY\$**, e faz-nos descobrir a interacção do modo texto e do modo gráfico com **TAG**.
- Finalmente, **o nível 5** faz-nos descobrir instruções mais específicas: interrupções, leitura do écran, geração de caracteres.

Cada jogo é apresentado de modo muito sistemático, por uma curta descrição, seguida do estudo pormenorizado de cada uma das linhas importantes do programa. Após isto são fornecidas as indicações necessárias ao bom funcionamento do jogo, assim como alguns elementos que possam conduzir a extensões ou melhoramentos do jogo proposto.

O programa, escrito em Basic AMSTRAD, é seguidamente reproduzido. As listagens foram obtidas directamente dos originais do autor, numa impressora matricial, o que deverá garantir contra qualquer risco de mau funcionamento. (A tradução limitou-se a substituir as mensagens a imprimir no écran, o que em nada altera o que atrás fica dito.) As minúsculas não são utilizadas. Em contrapartida, deve fazer-se entrar o programa em minúsculas.

Antes de começar qualquer programa certifique-se de que a máquina foi reinicializada, com a ajuda da sequência de instruções: **MODE 1: INK 0,1: INK 1,24: INK 2,20: INK 3,6.**

Atenção: no écran do AMSTRAD os zeros aparecem quadrados e os «O» aparecem redondos; nas listagens, a diferença entre o zero e o O maiúsculo é evidente: o zero aparece cortado (∅).

Nível 1

<i>Nome do programa</i>	<i>Temas e particularidades</i>
O número misterioso	Jogo de dedução, para um jogador. Vários níveis.
Cubo	Jogo de desenho; perspectiva num écran.
Quente, frio	Lógica e acaso, para um jogador. Vários níveis.
Os fósforos	Jogo de reflexão, contra o computador.
Fósforos para vários	Jogo de reflexão e de estratégia, para vários jogadores.
Calendário perpétuo	Confecção de um calendário, mês após mês.
Batalha naval	Encontrar apenas um navio, mas algumas particularidades.
Golfe	Um percurso de 18 buracos, para um só jogador.
Solfejo	Aprenda solfejo com a ajuda do seu computador.
O jogo do 21	Jogo de azar, o computador é a banca.
Par-ímpar	Um jogo de casino em três tempos (um ou vários jogadores).
Adivinha um número ...	Um jogo muito fácil, para um ou vários jogadores.
O jogo de Witthof	Um jogo de reflexão, bastante simples, para dois jogadores.
Tabelas	Um jogo de destreza muito visual, para um jogador.
«Jackpot»	Jogo de azar, para um jogador.
Roleta	Uma elegante simulação do conhecidíssimo jogo.
Corredores	Um jogo de lógica, para descobrir.
Tabuleiro de xadrez	Uma base gráfica para as suas próprias criações.
Jogo da fava	Um jogo de reflexão, para dois jogadores.
Jogo da glória aleatório	Um jogo muito antigo, revisto e corrigido pelo computador; dois jogadores.

Instruções elementares

Vamos referir aqui apenas as instruções fundamentais, ou instruções de programação; muitos outros comandos, ou funções particulares, serão utilizados ocasionalmente, por exemplo:

- *RND* para obter um número aleatório entre 0 e 1.
- *INT* para tomar a parte inteira de um resultado.
- *REM* para colocar um comentário.
- *END* para terminar um programa.
- *etc.*

As instruções fundamentais da programação, utilizadas logo desde o nível 1, e presentes em praticamente todos os programas, são:

LET (opcional); **PRINT**; **INPUT**; **GOTO** [] **IF** []
THEN [] ; **FOR...NEXT**; **GOSUB...RETURN**

ou seja, somente sete instruções básicas (e duas instruções derivadas), que iremos agora examinar mais pormenorizadamente.

A INSTRUÇÃO LET OU INSTRUÇÃO DE ATRIBUIÇÃO

É a instrução por excelência das linguagens de programação evoluída, porquanto o seu papel essencial é permitir, ao programador, uma gestão transparente da memória. Isto significa, na prática, que para pôr em memória uma quantidade, por exemplo 12, não é necessário saber em que célula da memória ela vai ser colocada: basta atribuir-lhe um nome autorizado e é por esse nome que a quantidade será posteriormente encontrada e manipulada.

- **Se a quantidade é numérica**, o nome será uma simples letra do alfabeto, ou um conjunto de várias cifras ou letras, começando obrigatoriamente por uma letra.
- **Se a quantidade é alfanumérica** aplica-se a mesma regra para o nome, que deverá no entanto ser seguido pelo símbolo \$, e a variável deverá ser escrita entre aspas.

Exemplo:

LET A1 = 12: a variável A1 toma o valor 12.

LET NOME\$ = «RENOIR»; a variável alfanumérica NOME\$ toma o valor RENOIR.

No AMSTRAD, como na maior parte dos Basic, o LET tornou-se opcional, por não ser possível qualquer ambiguidade, pelo que se escreverá: A = 12; NOME\$ = «TOTO»; etc.

A INSTRUÇÃO PRINT OU INSTRUÇÃO DE SAÍDA (ÉCRAN)

É uma instrução de diálogo, que permite que a máquina imprima no écran um texto, o resultado de um cálculo ou uma variável. É o único meio que o computador pessoal possui para fazer conhecer os seus resultados. Assim, PRINT A1 provoca a impressão de 12 (se anteriormente se faz A1 = 12), PRINT A\$ provoca a impressão de BOM DIA (se se faz A\$ = «BOM DIA»). Utilizado após uma limpeza do écran, o PRINT provoca a impressão na primeira linha, ao alto e à esquerda.

Pode ser utilizado certo número de parâmetros para obter uma impressão conveniente no écran:

- **PRINT TAB(N)** provoca, na linha de impressão corrente, a impressão a começar na enésima coluna a partir da esquerda (1N40 no modo 1).
- Um ; (**ponto e vírgula**) no fim do PRINT permite quer encadear uma outra impressão sem repetir a instrução PRINT, quer indicar que a próxima impressão será, na ausência de indicação em contrário, efectuada exactamente a seguir à impressão precedente.
- A , (**vírgula**) provoca uma tabulação horizontal automática pré-definida modificada pela **função ZONE**.

Ao teclar, a **instrução PRINT** poderá ser abreviada teclando ? (**ponto de interrogação**). Aquando de uma impressão, todos os números são precedidos de um sinal; mas o sinal positivo não é impresso, um espaço precederá portanto automaticamente a impressão de qualquer resultado positivo.

A INSTRUÇÃO LOCATE

O AMSTRAD dispõe de uma instrução particular, de uso muito elástico, que *permite posicionar o cursor no écran, tendo em vista a próxima escrita*. Trata-se da instrução LOCATE X,Y. X corresponde a uma tabulação horizontal (entre 1 e 40) e Y a uma tabulação vertical (entre 1 e 25) em modo texto.

Assim, para escrever BOM DIA na 5.^a coluna da 8.^a linha escrever-se-á:

```
LOCATE 5,8 : PRINT «BOM DIA»
```

A INSTRUÇÃO INPUT OU INSTRUÇÃO DE ENTRADA (TECLADO)

Esta instrução, que provoca no aparelho uma paragem e uma espera, *permitirá fazer entrar um número, ou um texto, que será atribuído à variável que se segue à instrução INPUT*. Isto é feito em modo conversacional, isto é, à medida das necessidades; efectivamente, quando o AMSTRAD encontra uma instrução INPUT pára e passamos o controlo para as mãos; pode então fazer-se entrar uma informação de natureza conveniente (número ou texto) e assinalar-lhe que esta entrada está terminada com a ajuda da tecla **ENTER**, o que lhe devolverá o comando das operações.

Se a informação que se fez entrar não tem a natureza conveniente (por exemplo, se teclar um texto em resposta a um INPUT X que espera um número), a mensagem **REDO FROM START** é impressa no écran e o aparelho volta a aguardar uma entrada.

É possível, tal como com o PRINT, imprimir mensagens antes da entrada da variável e utilizar a instrução LOCATE.

Poder-se-á, por exemplo, escrever:

```
INPUT «O SEU NOME, SE FAZ FAVOR»; A$
```

AS JANELAS DE ÉCRAN

O écran pode ser dividido em oito janelas, cada uma com sua cor de tinta e posição de cursor. Por defeito, as oito janelas são rigorosamente sobrepostas. A ordem WINDOW permite limitá-las a uma parte

do écran. Assim, WINDOW # 1,20,25,12,14 abre uma janela de écran entre as colunas 20 e 25 e entre as linhas 12 e 14.

As instruções PRINT, INPUT e LOCATE, para serem executadas numa janela particular, devem utilizar o número respectivo dessa janela.

Assim, PRINT # 1, «TOTO» imprimirá na primeira janela, nas coordenadas (eventualmente) definidas por uma ordem do tipo LOCATE # 1,X,Y. No caso de uma janela, X e Y são contadas em relação ao canto superior esquerdo desta, independentemente da posição no écran habitual de 40 × 25 linhas.

A INSTRUÇÃO GOTO OU INSTRUÇÃO DE SALTO INCONDICIONAL

Uma das particularidades da linguagem Basic, que a tornam facilmente acessível, é a de *cada linha de programa ter de ser precedida de um número*, sendo essas linhas posteriormente executadas, pelo computador pessoal, em sequência, no sentido crescente dos números. Por diversas razões, poderá ser necessário, no decorrer de um programa, não seguir essa rígida ordem sequencial; utilizar-se-á então *a instrução GOTO, que permite desviar o funcionamento do programa do seu curso normal*, voltando aliás a retomá-lo seguidamente por um outro GOTO. Além disso, imperativamente, o salto deve ser feito para um número de linha existente no programa (GOTO 20, por exemplo).

Para dar certa elasticidade a esta instrução de salto existe uma instrução mais poderosa, que permite o salto para qualquer número de linha, segundo o valor de uma variável numérica: é a *instrução ON GOTO*; escrever-se-á, por exemplo:

ON X GOTO 100, 200, 300

e o salto será feito para a linha 100 se X tiver o valor 1, para a linha 200 se X tiver o valor 2 e para a linha 300 se X tiver o valor 3.

A INSTRUÇÃO IF THEN OU INSTRUÇÃO DE TESTE

Pode ser utilizada de duas maneiras: antes de mais, numa rotura de sequência condicional, sob a forma:

IF condição = verdade THEN número de linha

É este o uso mais corrente deste teste, que permite saltar para uma outra parte do programa se, por exemplo, $X = 0$, ou se $A = B$, etc. Os **operadores lógicos NOT, OR, XOR ou AND** podem aliás ser utilizados ao escrever a condição.

Poder-se-á também utilizar o teste sob a forma:

IF condição THEN instrução;

a instrução apresentada a seguir ao THEN pode ser uma atribuição $A = 12$, uma impressão PRINT A, uma entrada INPUT B\$, ou mesmo um outro teste IF THEN .

Pode, por exemplo, escrever-se:

IF $A > 3$ THEN IF $B < 7$ THEN PRINT $A + B$

A INSTRUÇÃO FOR... NEXT OU INSTRUÇÃO DE CICLOS

Em programação é frequentemente necessário executar certo número de vezes uma dada sequência de instruções (por exemplo, fazer avançar um objecto móvel da célula 6 para a célula 20, etc.). Em vez de utilizar um teste e um salto, será preferível apelar para um *ciclo* ou *anel*, com a variável de salto a tomar valores inteiros ou decimais, e o *passo* desta variável (ou seja, a quantidade em que ela aumenta ou diminui a cada passagem) podendo também tomar valores inteiros ou decimais.

Assim se escrevermos:

```
FOR I = {  
NEXT I } 0 TO 20 STEP 2  
Instruções
```

As instruções contidas no ciclo serão executadas para $I = 0$, para $I = 2$, ..., e isto até $I = 20$, ou seja, onze vezes. A variável I pode ser utilizada para fazer um cálculo no ciclo, mas não deverá ser modificada no ciclo. Pode-se, é claro, sair do ciclo antes de $I = 20$, usando um teste contendo uma quantidade qualquer.

Se o *passo do ciclo* (STEP) não for definido, será automaticamente tomado como igual a 1.

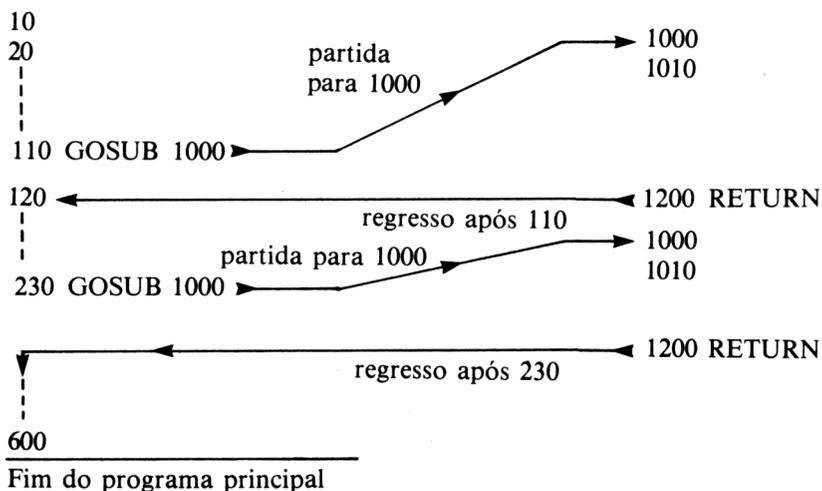
É possível encaixar diversos ciclos de nomes diferentes, e omitir indicação da variável no *NEXT*.

A INSTRUÇÃO GOSUB OU INSTRUÇÃO DE CHAMADA DE ROTINA (OU SUBPROGRAMA)

Quando uma dada seqüência de instruções deve ser executada várias vezes, é chamada a partir de vários locais do programa, evitar-se-á reescrevê-la várias vezes. Redigimo-la uma vez por todas e colocamo-la fora do programa propriamente dito (dando-lhe, por exemplo, um número de linha superior ao número da última linha do programa). Independentemente das dificuldades que poderão ser encontradas ao nível do nome das variáveis, quando se desejar utilizar esta seqüência bastará chamá-la através da **instrução GOSUB número de linha**. O programa irá então «fazer um desvio» à parte do programa chamada, mas não poderá regressar e prosseguir o seu desenvolvimento normal enquanto não encontrar no programa chamado uma instrução **RETURN**, ou instrução de retorno.

Eis o esquema teórico de funcionamento:

Programa principal



1000
1200 RETURN
Subprograma 1

A rotina é escrita apenas uma vez, mas chamada tantas vezes quantas quisermos.

Uma rotina poderá eventualmente ela própria chamar uma outra rotina — e assim sucessivamente.

Uma rotina pode também ser chamada de modo condicional, utilizando a instrução ON `┌` GOSUB `└` .

Por exemplo, ON X GOSUB 1000, 2000, 3000, permitirá executar o subprograma colocado em 1000 se X = 1, em 2000 se X = 2, em 3000 se X = 3.

A INSTRUÇÃO WHILE/WEND

Quando uma sequência de instruções deve ser executada enquanto se verificar uma certa condição, basta colocá-la no meio de um ciclo WHILE/WEND. **Atenção: antes de começar qualquer ciclo WHILE/WEND, é prudente inicializar a variável (ou variáveis) utilizada(s) para o teste de condição.**

EXEMPLO: 10 WHILE TE<200
20 TE = TE + 1 : PRINT TE
30 WEND

A variável TE é aumentada uma unidade e impressa enquanto for inferior a 200.

É possível encaixar entre si diversos ciclos WHILE/WEND (ver o programa «Relógio 2»), o que permite uma grande elegância na escrita dos programas.

O número misterioso

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: jogo de dedução para um jogador.

Dificuldade do jogo: fácil.

O JOGO:

Pede-se ao computador que construa um número de n algarismos (com n compreendido entre 1 e 7).

Dever-se-á seguidamente tentar descobrir, com o mínimo de tentativas possíveis, recebendo para tal somente as informações «DEMASIADO GRANDE» ou «DEMASIADO PEQUENO».

O PROGRAMA:

A linha 60 permite construir o número a adivinhar.

As entradas são feitas pelas linhas 80 e 90.

A linha 100 testa se o resultado é exacto.

A linha 110 assinala uma tentativa demasiado pequena, e a linha 120 uma tentativa demasiado grande.

O contador de tentativas está na linha 130.

As linhas 160 a 180 ocupam-se da impressão do resultado e da gestão da partida seguinte.

PARA JOGAR:

Faz-se trabalhar o programa com RUN.

Obtém-se uma primeira impressão «NÚMERO DE ALGARISMOS A ADIVINHAR?», à qual se responde com um inteiro entre 1 e 7, seguido de ENTER.

Obtém-se então a mensagem «TENTATIVA No. 1», seguida da pergunta «O SEU NÚMERO?».

Faz-se entrar um número e prime-se ENTER.

Obtém-se então uma resposta «DEMASIADO PEQUENO», ou «DEMASIADO GRANDE», ou ainda, se acertamos, uma mensagem de «ACERTOU EM ... TENTATIVAS». Se o resultado não é exacto recomenda-se: «TENTATIVA No. 2», e «O SEU NÚMERO?».

No fim da partida obtém-se a pergunta «OUTRA PARTIDA? (S/N)»; responde-se S ou s (seguido de ENTER) para novo jogo.

Pode-se parar em qualquer momento, premindo duas vezes seguidas a tecla ESC.

EXTENSÕES POSSÍVEIS:

Poder-se-á imaginar fazer o jogador adivinhar outra coisa sem ser um número (por exemplo, uma palavra, utilizando a ordem alfabética, ou qualquer outra coisa...).

```
10 REM O NUMERO MISTERIOSO
20 CLS
30 INPUT "NUMERO DE ALGARISMOS A ADIVINH
AR?";N
40 IF N>7 THEN RUN
50 IF N<=0 THEN RUN
60 X=INT(RND(1)*10^N+1)
70 J=1
80 PRINT:PRINT "TENTATIVA No. ";J
90 INPUT "O SEU NUMERO? ";M
100 IF M=X THEN 160
110 IF M<X THEN PRINT "DEMASIADO PEQUENO
"
120 IF M>X THEN PRINT "DEMASIADO GRANDE"
130 J=J+1
140 PRINT
150 GOTO 80
160 PRINT:PRINT "ACERTOU EM ";J;" TENTAT
IVAS"
170 INPUT "OUTRA PARTIDA? (S/N)";R$
180 IF R$="S" OR R$="s" THEN RUN
```

Cubo

REFERÊNCIAS:

Dificuldade de programação: nenhuma.

Categoria de jogo: desenho.

Dificuldade do jogo: nenhuma.

O JOGO:

O programa proposto contenta-se em realizar, com a ajuda de instruções PRINT, uma vista de um cubo em perspectiva livre.

Poderá ser incorporado, por exemplo, num jogo baseado nas capacidades de reconhecimento das formas geométricas elementares.

O PROGRAMA:

O desenho do cubo é realizado pela impressão de pontos, utilizando dois ciclos: o ciclo I (30-120) e o ciclo I (130-180).

O nome do objecto representado é impresso na linha 200 (neste caso: CUBO).

PARA JOGAR:

Basta fazer correr o programa com a instrução RUN e observar. Far-se-á ESC duas vezes para recuperar o controlo do programa.

EXTENSÕES POSSÍVEIS:

A alta resolução permitirá conseguir um traçado mais preciso.

Poder-se-á imaginar um jogo que utilize paralelepípedos rectangulares de diferentes tamanhos, mostrando por exemplo três lados, e pedir que se determine o seu volume, etc.

```
10 REM CUBO
20 CLS
30 FOR I=1 TO 11
40 LOCATE 7+I,1:PRINT ". "
50 LOCATE I,8:PRINT ". "
60 LOCATE 8+I,12:PRINT ". "
70 LOCATE 1+I,19:PRINT ". "
80 LOCATE 1,8+I:PRINT ". "
90 LOCATE 12,7+I:PRINT ". "
100 LOCATE 8,I+1:PRINT ". "
110 LOCATE 19,I:PRINT ". "
120 NEXT I
130 FOR I=1 TO 6
140 LOCATE 1+I,8:PRINT ". "
150 LOCATE 12+I,8-I:PRINT ". "
160 LOCATE 1+I,19-I:PRINT ". "
170 LOCATE 12+I,19-I:PRINT ". "
180 NEXT I
190 LOCATE 26,19:PRINT CHR$(150);STRING$(4,154);CHR$(156)
200 LOCATE 26,20:PRINT CHR$(149);"CUBO";CHR$(149)
210 LOCATE 26,21:PRINT CHR$(147);STRING$(4,154);CHR$(153)
220 GOTO 220
```

Quente, frio

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: solitário.

Dificuldade do jogo: média.

O JOGO:

Um dos mais velhos jogos do mundo, praticado desde sempre, mas desta vez é o computador que lhe serve de parceiro de jogo: ele vai, com efeito, esconder um objecto num certo lugar de uma grelha cujas dimensões você próprio escolheu. Para o encontrar, ele dar-lhe-á apenas algumas indicações sob a forma de A ESCALDAR, ou COMPLETAMENTE GELADO, etc.

Deverá conseguir descobrir o objecto escondido, com o mínimo de tentativas possível (a sua pontuação será impressa no fim do jogo).

O PROGRAMA:

As linhas 40 e 50 definem a posição do objecto.

A linha 100 calcula a distância em relação às duas coordenadas que você fez entrar nas linhas 70 e 80.

A linha 110 verifica se você ganhou.

As linhas 130-140 normalizam a distância (que deve ter um sentido, sejam quais forem as dimensões da grelha inicial).

A linha 150 dá o salto, se houver motivo para isso, para uma das rotinas de impressão.

PARA JOGAR:

Inicia-se o programa com a ordem RUN, depois introduzem-se as dimensões da grelha (quadrada) e prime-se ENTER.

Após a impressão da mensagem seguinte (ENTRE O VALOR DE X) faz-se entrar a coordenada X que supomos correcta, seguida de ENTER.

Obtém-se a impressão da mensagem ENTRE O VALOR DE Y e faz-se entrar Y, seguido de ENTER.

Em resposta obtém-se um texto, que é variável consoante se estiver mais ou menos longe do objecto procurado (esta resposta vai de COMPLETAMENTE GELADO a A ESCALDAR). Quando se conseguiu encontrar o objecto obtém-se a impressão do número de tentativas efectuadas.

Para sair do programa antes do fim do jogo, basta premir ESC duas vezes seguidas.

EXTENSÕES POSSÍVEIS:

Se se dispõe da cor, será interessante descobrir um meio mais visual para simbolizar a distância, por exemplo quadrados que podem ir do vermelho ao azul, nos locais onde foram feitas tentativas.

```
10 REM QUENTE, FRIO
20 CLS
30 INPUT "DIMENSÃO DA GRELHA: ";D
40 X=INT(RND*D+1)
50 Y=INT(RND*D+1)
60 CLS
70 INPUT "ENTRE O VALOR DE X: ";A
80 INPUT "ENTRE O VALOR DE Y: ";B
90 S=S+1
100 T=ABS(X-A)+ABS(Y-B)
110 IF T=0 THEN 280
120 PRINT:PRINT "ESTA' ";
130 N=INT(SQR(D)/T)+1
140 IF N>5 THEN N=5
150 IF T<=SQR(D) THEN ON N GOTO 180,200,
220,240,260
160 PRINT "COMPLETAMENTE GELADO"
170 GOTO 70
180 PRINT "FRESCO"
190 GOTO 70
200 PRINT "MORNO"
210 GOTO 70
220 PRINT "QUENTE"
230 GOTO 70
240 PRINT "MUITO QUENTE"
250 GOTO 70
260 PRINT "A ESCALDAR"
270 GOTO 70
280 CLS:BORDER 3,4
290 LOCATE 10,12:PRINT "DESCOBRIU EM ";S
;" TENTATIVAS."
300 FOR T=1 TO 5000:NEXT T
310 BORDER 1,1
```

Os fósforos

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: contra o AMSTRAD.
Dificuldade do jogo: média.

O JOGO:

O jogador deverá escolher, no início do jogo, uma certa quantidade de fósforos, assim como o número máximo deles que se pode retirar de cada vez. Em seguida decide se é ele a começar ou se deixa começar o AMSTRAD. Cada qual tira por sua vez tantos fósforos quantos desejar, dentro do limite do número máximo autorizado.

Perde aquele que tirar o último fósforo.

O PROGRAMA:

Para o jogador, o jogo começa na linha 70.

Para o AMSTRAD começa na linha 110.

A linha 80 controla a legalidade da jogada do jogador.

A linha 90 gera a mudança de jogador.

A jogada do AMSTRAD é calculada num ciclo não formal, da linha 120 à linha 150.

O resultado é impresso pela linha 1020.

Se o jogo está terminado, a linha 1030 detecta-o e envia para a linha 2000, para a mensagem de fim.

PARA JOGAR:

A execução começa-se fazendo RUN e, depois de aparecer a mensagem NUMERO TOTAL, faz-se entrar o número de fósforos, seguido de ENTER. Depois faz-se entrar o número máximo de fósforos que se pode retirar de cada vez, seguido de ENTER. Se o jogador quer começar, carrega na tecla 0; se não, carrega na tecla 1, e depois em ENTER.

Após cada jogada, o número de fósforos restantes é impresso no écran, e o AMSTRAD imprime VOCE RETIRA quando é a vez do jogador.

Perde aquele que retirar o último fósforo.

EXTENSÕES POSSÍVEIS:

Poder-se-á, evidentemente, pensar numa apresentação gráfica a cores mais espectacular, imaginar deixar o acaso escolher o número de fósforos e o número máximo que se pode retirar de cada vez, etc.

```

10 REM OS FOSFOROS
20 CLS
30 INPUT "NUMERO TOTAL          :";N
40 INPUT "MAXIMO PERMITIDO RETIRAR :";I
50 INPUT "QUEM COMECA (VOCE: 0/EU: 1):"
;D
60 IF D=1 THEN 110
70 INPUT "VOCE RETIRA :";P
80 IF P<>INT(P) OR P<1 OR P>N OR P>I THE
N 70
90 D=1
100 GOTO 1000
110 S=I+D
120 T=(N-S)/(I+D)
130 IF T=INT(T) THEN 200
140 S=S-D
150 GOTO 120
200 P=ABS(S-D)
210 IF P=0 THEN P=D
220 D=0
1000 PRINT P
1010 N=N-P
1020 PRINT "SOBRAM ";N
1030 IF N=0 THEN 2000
1040 GOTO 60
2000 CLS
2010 IF D=1 THEN PRINT "GANHEI":SOUND 1,
478:SOUND 1,426:SOUND 1,379
2020 IF D<>1 THEN PRINT "BRAVO":SOUND 1,
256,4:SOUND 1,256,4,0:SOUND 1,256,4

```

Fósforos para vários

REFERÊNCIAS:

- Dificuldade de programação: fácil.
- Categoria de jogo: lógica (vários jogadores).
- Dificuldade do jogo: bastante difícil.

O JOGO:

Neste jogo para vários jogadores, o condutor do jogo deverá, no início da partida, escolher um número de fósforos e indicar ao AMSTRAD o número de jogadores.

Em cada jogada, o jogador que está na sua vez de jogar pode retirar do monte de fósforos restantes até o dobro dos que foram retirados pelo jogador precedente; aquele que retirar o último fósforo ganha (ou perde, se se modificarem ligeiramente as regras.)

O PROGRAMA:

A linha 20 permite trabalhar com inteiros, provocando o arredondamento dos valores.

As linhas 40 a 60 permitem as diferentes entradas.

As linhas 100 a 120 realizam as impressões.

A linha 140 controla a jogada de cada jogador (em relação à jogada precedente).

A linha 200 gera a vez dos jogadores (reposição a 1 do número do jogador).

PARA JOGAR:

Depois de fazer correr o programa usando RUN, faz-se entrar o número inicial de fósforos, seguido de ENTER, depois o número de jogadores (e ENTER).

O primeiro jogador só poderá tirar 1 fósforo; o segundo poderá tirar 1 ou 2; o seguinte, 1 ou 2, ou 1, 2, 3 ou 4, consoante a jogada anterior — e assim por diante; o número máximo que se pode tirar de cada vez é impresso pelo AMSTRAD.

Quando o último fósforo for retirado obtém-se a mensagem impressa O JOGADOR... GANHA, e a partida termina.

EXTENSÕES POSSÍVEIS:

Pode-se convencionar que aquele que tira o último fósforo é eliminado do jogo e recomeçar a partida entre os jogadores restantes, que assim serão eliminados um após outro.

Pode-se também imaginar uma representação gráfica dos montes de fósforos, estabelecer ao acaso o número de fósforos a retirar no começo do jogo, etc.

```
10 REM FOSFOROS PARA VARIOS
20 DEFINT F,J,K,N
30 CLS
40 INPUT "NUMERO DE FOSFOROS? ";N
50 PRINT
60 INPUT "NUMERO DE JOGADORES? ";J
70 F=1:K=1:INK 1,0
80 CLS
90 INK 0,K
100 PRINT TAB(12)"VEZ DO JOGADOR";K:PRIN
T
110 PRINT "TOTAL RESTANTE      ";N
120 PRINT "MAXIMO A RETIRAR    ";F
130 INPUT "A SUA JOGADA      ";C
140 IF C<=0 OR C>F THEN 130
150 N=N-C
160 IF N<0 THEN N=N+C:GOTO 130
170 F=C+C
180 IF N=0 THEN 220
190 K=K+1
200 IF K>J THEN K=1
210 GOTO 80
220 CLS
230 LOCATE 12,12:PRINT "O JOGADOR ";K;"G
ANHA."
```

Calendário perpétuo

REFERÊNCIAS:

Dificuldade de programação: média.
Categoria de jogo: brinde publicitário.
Dificuldade do jogo: nenhuma.

O JOGO:

Este programa permitir-lhe-á confeccionar todos os anos o seu próprio calendário; poderá assim servir-se dele para enviar aos seus amigos um calendário do mês do seu nascimento, ou para lhes desejar feliz aniversário, etc.

O PROGRAMA:

As linhas 30 e 50 permitirão dar entrada ao ano e ao mês.

As linhas 60, 70 e 80 calculam o número do primeiro dia do mês.

A linha 90 calcula o número de dias do mês (para simplificar, aqui não se toma em consideração os 29 de Fevereiro).

O ciclo A (160-190) realiza a impressão do calendário propriamente dito.

A linha 140, com a ajuda da ordem **ZONE**, permite separar com cinco espaços em branco os elementos da lista separados por uma vírgula.

A linha 200 evita a aparição da palavra **READY**.

PARA JOGAR:

Depois de fazer correr o programa com a instrução **RUN** deverá introduzir-se o ano, sob a forma de, por exemplo, 1986, seguido de **ENTER**.

Obtém-se então a pergunta **MES**, à qual convém responder com o número do mês, de acordo com a ordem usual (1 para **JANEIRO**, 2 para **FEVEREIRO**, etc.).

Seguidamente faz-se **ENTER**, e o calendário será então impresso no écran.

EXTENSÕES POSSÍVEIS:

Poder-se-á sem dificuldade prever a impressão dos 29 de Fevereiro, prever uma opção que permita a impressão de um ano completo, prever a introdução do nome dos meses, até agora em claro, etc.

```

10 REM CALENDARIO PERPETUO
20 CLS
30 INPUT "PARA QUE ANO?";A
40 PRINT
50 INPUT "PARA QUE MES?";M
60 J=INT(365.25*(A+(M<3)))+1
70 J=J+INT(30.6*(M+1-(M<3)*12))-INT((INT
((A+(M<3))/100)-7)*0.75)
80 J=J-7*INT(J/7)
90 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2
)*3
100 PRINT:PRINT:PRINT
110 PEN 2
120 PRINT "DOM  SEG  TER  QUA  QUI  SEX
SAB"
130 PRINT
140 ZONE 5:PEN 3
150 LOCATE 5*J+1,10
160 FOR A=1 TO M
170 PRINT A,
180 IF J+A-1=6 THEN PRINT:PRINT:J=J-7
190 NEXT A
200 GOTO 200

```

Batalha naval

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: reflexão.

Dificuldade do jogo: fácil.

O JOGO:

Uma batalha naval bastante simples *a priori*, com um único navio escondido ao acaso numa grelha de 10 por 10 casas.

Em cada jogada dever-se-á disparar dando as coordenadas H (horizontal) e V (vertical) da casa visada, e o AMSTRAD responde imprimindo no écran a distância que separa o navio invisível da casa visada. Mas, se o tiro cai muito perto, corre-se o risco de o navio voltar a esconder-se ao acaso.

O PROGRAMA:

O ciclo I (50-80) desenha o campo de jogo.

As linhas 100-110 definem a colocação do navio.

As coordenadas do tiro são feitas entrar nas linhas 130 e 140, e a linha 150 imprime o tiro, no écran. A distância é calculada na linha 160, e a linha 170 verifica se se acertou ou não no alvo.

A linha 190 intervém para deslocar o navio se os tiros são muito próximos do alvo.

A linha 200 permite imprimir a distância.

PARA JOGAR:

O campo de jogo é impresso depois de se ter teclado RUN. Depois de surgir a palavra LINHA? responde-se dando o número da linha horizontal sobre a qual se deseja disparar. A seguir teclia-se ENTER, e à pergunta COLUNA? responde-se dando o número da coluna vertical, seguido de ENTER.

Se se atira muito perto obtém-se a impressão da mensagem PFFFF..., que significa que o navio inimigo mudou de posição.

Se se atirou sobre a casa exacta obtém-se a impressão do número de tiros efectuados para ganhar.

EXTENSÕES POSSÍVEIS:

São inúmeras, tal como a batalha naval é um dos jogos mais espalhados. Pode introduzir-se dois jogadores, jogar contra o AMSTRAD, pôr vários navios, submarinos, etc.

```

10 REM BATALHA NAVAL
20 RANDOMIZE(TIME)
30 CLS
40 PRINT "    0  1  2  3  4  5  6  7  8
9":PRINT
50 FOR I=0 TO 9
60 PRINT I;" . . . . .
"
70 PRINT
80 NEXT I
90 S=1
100 X=INT(RND*10)
110 Y=INT(RND*10)
120 LOCATE 1,23:PRINT "
"
130 LOCATE 1,23:INPUT "LINHA? ";H
140 LOCATE 24,23:INPUT "COLUNA?";V
150 LOCATE 3*V+3,2*H+3:PEN 3:PRINT S:PEN
1
160 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
170 IF D=0 THEN 210
180 S=S+1
190 IF D=1 THEN LOCATE 1,24:PRINT "PFFF.
. . . . .";GOTO 100
200 LOCATE 1,24:PRINT "DISTANCIA      : "
;D:GOTO 120
210 CLS
220 PRINT "DESCOBERTO EM ";S;" JOGADAS"

```

Golfe

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: solitário.

Dificuldade do jogo: bastante fácil.

O JOGO:

Trata-se de tentar realizar, com o menor número de tacadas possível, um percurso de golfe de dezoito buracos. Para isso é necessário bater a bola com maior ou menor força, até obter a sua queda no buraco.

Mas atenção: pode haver um certo elemento casual em cada tacada, e o jogador será levado a voltar para trás [nesse caso será necessário fazer preceder de um sinal de menos (-) a indicação da força].

O PROGRAMA:

A linha 70 permite definir a posição do buraco, e a linha 80 desenha o campo de jogo.

A deslocação a realizar é calculada na linha 120.

O ciclo I (150-210) realiza a deslocação da bola.

A linha 220 verifica se a tacada acertou ou não; se acertou, recomeça-se (linha 270) até se ter realizado 18 buracos [ciclo S (40-270); fim de ciclo em 290, com impressão da mensagem de fim].

PARA JOGAR:

Teclando RUN obtém-se a impressão do campo de jogo, assim como o desenho da bola na sua posição de partida e o texto: BURACO 1. FORÇA?; deve-se então responder dando a força desejada (um número entre 1 e 300...).

Desde que se tenha teclado ENTER, a bola começa a deslocar-se; pode então parar antes do buraco (caso em que se recomeçará), cair nele (passa-se então ao buraco seguinte), ou ultrapassá-lo (deverá então bater a bola com uma força negativa).

O jogo termina com a impressão da mensagem 18 BURACOS EM ... TENTATIVAS.

EXTENSÕES POSSÍVEIS:

Será possível ampliar o jogo para fazer participar vários jogadores, melhorar a representação gráfica, utilizar uma representação espacial do campo (tomar como referência certos jogos de arcada), criar um campo mais acidentado, mudar de taco...

```

10 REM GOLFE
20 RANDOMIZE(TIME)
30 INK 0,9:INK 1,18:INK 2,0:INK 3,26
40 FOR S=1 TO 18
50 CLS
60 X=1
70 T=INT(29*RND+9)
80 PEN 1:LOCATE 1,12:PRINT STRING$(40,20
8)
90 PEN 3:LOCATE T,12:PRINT " ":LOCATE 1,
11:PRINT "0";
100 PEN 2:LOCATE 1,17:PRINT "BURACO ";S;
" FORCA ?"
110 LOCATE 16,17:PRINT " "
120 PEN 2:LOCATE 16,17:INPUT "",R
130 F=INT(R*0.16-(ABS(R)*(SQR(RND)+0.1)^
2)/1000)
140 IF F=0 THEN 110
150 FOR I=1 TO MIN (ABS(F),40)
160 X=X+SGN(F)
170 PEN 3
180 IF X<=40 AND X>=1 THEN LOCATE X-SGN(
F),11:PRINT " ":LOCATE X,11:PRINT "0"
190 IF X>=40 THEN X=40
200 IF X<=1 THEN X=1
210 NEXT I
220 IF X=T THEN 250
230 E=E+1
240 GOTO 110
250 LOCATE X,11:PRINT " ":LOCATE X,12:PR
INT "0"
260 FOR P=1 TO 1000:NEXT P
270 NEXT S
280 CLS
290 LOCATE 10,12:PRINT "18 BURACOS EM ";
E;" TACADAS"

```

Solfejo

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: reflexão.

Dificuldade do jogo: fácil.

O JOGO:

Este programa propõe-lhe uma iniciação elementar ao solfejo, permitindo-lhe aprender a reconhecer facilmente as diferentes notas, que aparecem uma após outra sobre uma pauta; após a apresentação da nota, deve fazer entrar o nome dela; a sua pontuação será exposta permanentemente.

O programa está previsto, inicialmente, para a clave de sol, mas será muito facilmente adaptado a outra clave.

O PROGRAMA:

O ciclo I (50-80) desenha uma pauta.

A nota é escolhida na linha 100 e impressa pelas linhas 130 a 160 com tinta transparente (linha 120).

A linha 190 imprime a pontuação.

A linha 200 imprime o nome da clave (sol, neste caso) e pergunta qual é a nota.

A linha 220 ocupa-se do fim do jogo.

As linhas 230 e 240 efectuam a descodificação do nome da nota que se fez entrar.

A linha controla o resultado.

PARA JOGAR:

Depois de fazer correr o programa com RUN será impressa a pauta, assim como a primeira nota.

Responderá à pergunta QUAL É A NOTA? fornecendo o nome da nota (dó, ré, mi, fá, sol, lá ou si), seguido de ENTER.

A pontuação indicar-lhe-á se acertou ou não, e será impressa uma nova nota.

Para parar bastará responder 0, seguido de ENTER, à pergunta QUAL É A NOTA?.

EXTENSÕES POSSÍVEIS:

Poder-se-á antes de mais modificar facilmente o programa para lhe permitir trabalhar com as diferentes claves. Poder-se-á também

melhorar o grafismo, introduzir a cor, o som (por exemplo, quando se tiver respondido correctamente ouvir-se-á a nota). Seguindo o mesmo princípio, poder-se-á também realizar um jogo que permita reconhecer acordes, etc.

```
10 REM SOLFEJO
20 RANDOMIZE(TIME)
30 S=0:P=0
40 CLS
50 FOR I=1 TO 5
60 PEN 2
70 LOCATE 1,2*I+3:PRINT STRING$(40,210)
80 NEXT I
90 PEN 3
100 N=INT(RND*12)+1
110 X=N+3
120 PRINT CHR$(22)+CHR$(1)
130 LOCATE 21,X-2:PRINT CHR$(211):LOCATE
  21,X-2:PRINT CHR$(92)
140 LOCATE 21,X-1:PRINT CHR$(211)
150 LOCATE 19,X:PRINT CHR$(214);CHR$(215
);CHR$(211)
160 LOCATE 19,X+1:PRINT CHR$(213);CHR$(2
12)
170 PRINT CHR$(22)+CHR$(0)
180 PEN 1
190 LOCATE 10,21:PRINT "PONTUACAD ";S;"
EM ";P;" TENTATIVAS":PRINT
200 INPUT "CLAVE DE SOL - QUAL E' A NOTA
?";R$
210 R$=UPPER$(R$)
220 IF R$="" THEN END
230 Y=(1 AND R$="SOL")+(2 AND R$="FA")+
(3 AND R$="MI")+(4 AND R$="RE")
240 Y=Y+(5 AND R$="DO")+(6 AND R$="SI")+
(7 AND R$="LA")
250 P=P+1
260 IF Y=N OR (Y=N-7 AND N>9) THEN S=S+1
270 GOTO 40
```

O jogo do 21

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria do jogo: jogo contra o AMSTRAD.

Dificuldade do jogo: fácil.

O JOGO:

Dever-se-á fazer entrar uma parada no início do jogo, e poder-se-á lançar os dados teclando 1; a cada lançamento obter-se-á um novo total, que aumentará pouco a pouco e se aproximará de 21. Dever-se-á tentar chegar o mais próximo possível de 21, sem ultrapassar esse número. Quando se decidir parar bastará teclar 0 para deixar jogar o AMSTRAD, que tentará então atingir melhor pontuação.

Ao fim de cada partida será impresso o ganho, assim como o total dos ganhos (ou das perdas).

O PROGRAMA:

As linhas 60 e 70 realizam o controlo de quem deve jogar (o jogador ou o AMSTRAD).

A rotina (410-420) lança os dados.

As linhas 90 a 110 calculam o total conseguido pelo jogador e imprimem-no.

O total do AMSTRAD é calculado na linha 180 e impresso pela linha 200.

A linha 230 controla o fim do jogo para o AMSTRAD.

As linhas 260-280 imprimem os resultados.

O ganho ou a perda serão impressos pela linha 330, e o total pela linha 360.

A linha 390 controla o fim da partida.

PARA JOGAR:

Teclar RUN para começar o jogo.

À pergunta SUA PARADA? fazer entrar a parada inicial, seguida de ENTER, e teclar 1 para lançar o dado. A primeira tiragem é impressa sob a forma VOCE TEM..., e dever-se-á teclar 1 seguido de ENTER enquanto se desejar fazer lançamentos.

Quando se terminou teclou-se 0 e ENTER, para deixar jogar o AMSTRAD (que desempenha aqui o papel de banca).

O ganho (ou a perda, se tiver o sinal —) será impresso no fim do jogo, assim como o total dos ganhos (ou perdas) realizados desde o iní-

cio. Se se deseja deixar de jogar dever-se-á teclar N em resposta à pergunta CONTINUAMOS (S/N)?.

EXTENSÕES POSSÍVEIS:

Este jogo será evidentemente mais interessante se for jogado por várias pessoas, e será fácil modificar o programa nesse sentido. Poder-se-á também desenhar o dado, ou, melhor, utilizar cartas, o que permitirá então fazer um jogo de *blackjack*.

```
10 REM JOGO DO 21
20 RANDOMIZE(TIME)
30 CLS
40 T=0
50 INPUT "A SUA APOSTA :";M
60 LOCATE 3,25:INPUT "ENTRE 1 PARA LANCA
R O DADO, 0 PARA ACABAR";R
70 IF R=0 THEN 140
80 GOSUB 410
90 T=T+X
100 PAPER 3:PEN 2
110 LOCATE 10,12:PRINT USING "O SEU TOTA
L E' DE ##";T
120 PAPER 0:PEN 1
130 GOTO 60
140 CLS
150 PRINT "A BANCA JOGA"
160 S=0
170 GOSUB 410
180 S=S+X
190 PAPER 1:PEN 0
200 PRINT USING "TOTAL DA BANCA ##";S
210 PAPER 0:PEN 1
220 FOR I=1 TO 500:NEXT I
230 IF S>T OR S>17 THEN 250
240 GOTO 170
250 CLS
260 PAPER 3:PEN 2:PRINT USING "VOCE TEM
##";T
270 PRINT
280 PAPER 1:PEN 0:PRINT USING "A BANCA T
EM ##";S
290 PAPER 0:PEN 1
300 PRINT
310 IF S>21 OR (S<T AND T<=21) THEN 330
320 M=-M
330 PRINT "GANHO ";M
340 PRINT
350 G=G+M
360 PRINT "O SEU TOTAL ACTUAL ";G
370 PRINT
380 INPUT "CONTINUAMOS (S/N)? ";R#
390 IF R#="N" THEN END
400 GOTO 30
410 X=INT(6*RND)+1
420 RETURN
```

Par-ímpar

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: lançamento de dados.
Dificuldade do jogo: fácil.

O JOGO:

Trata-se de um jogo de azar, para um ou vários jogadores. O AMSTRAD lança simultaneamente dois dados, não visíveis, e faz três perguntas: qual é a paridade do total, qual é o total e, finalmente, qual o valor de um dos dados.

Pode-se em seguida prosseguir o jogo ou interrompê-lo em qualquer momento.

O PROGRAMA:

Os dados são lançados por uma rotina colocada nas linhas 380 e 390.

A linha 80 conta o número de partidas e as linhas 180, 250 e 290 contabilizam os pontos marcados.

A paridade do total é determinada pelas linhas 130 e 140 (a linha 140 compara com a resposta do jogador).

A linha 300 e as seguintes realizam a impressão do fim de partida: total, número de pontos, número de partidas jogadas.

A linha 360 provoca a paragem se se responde N à pergunta CONTINUAMOS (S/N)?.

PARA JOGAR:

Faz-se correr o programa com a instrução RUN.

Em seguida basta responder às diversas perguntas feitas, observando escrupulosamente as indicações dadas e teclando ENTER após cada resposta.

Quando se terminou uma partida obtém-se o resultado impresso do total actual e do número de partidas sob a forma de por exemplo, SEU TOTAL: 6 EM 10 PARTIDAS.

Aparece então impressa a pergunta CONTINUAMOS (S/N)? e dever-se-á responder N seguido de ENTER para acabar.

EXTENSÕES POSSÍVEIS:

Se se deseja jogar com vários jogadores será interessante prever a possibilidade de fazer paradas e de acrescentar uma gestão da pontuação alcançada por cada um dos jogadores. Poder-se-á também prever uma melhoria no plano da apresentação (utilização da cor, etc.).

```

10 REM PAR-IMPAR
20 RANDOMIZE (TIME)
30 INK 3,3,6
40 T=0:R=0
50 GOSUB 380
60 A=X
70 GOSUB 380
80 R=R+1
90 S=A+X
100 CLS
110 INPUT "TOTAL PAR (0) OU IMPAR (1)?";
N
120 PRINT
130 P=S/2-INT(S/2)
140 IF N=P OR N-0.5=P THEN 170
150 PEN 3:PRINT "FALSO":PEN 1
160 GOTO 190
170 PEN 2:PRINT "EXACTO":PEN 1
180 T=T+1
190 PRINT
200 INPUT "TOTAL CALCULADO?";N:PRINT
210 IF N=S THEN 250
220 PEN 3:PRINT "PERDEU":PEN 1
230 FOR I=1 TO 1500:NEXT I
240 GOTO 310
250 T=T+2
260 PEN 2:PRINT "BRAVO! ";:PEN 1
270 INPUT "VALOR DE UM DADO: ";N:PRINT
280 IF N<>A AND N<>S-A THEN 220
290 T=T+3
300 PEN 2:PRINT "BRAVO!":PEN 1:GOTO 230
310 CLS
320 PRINT "SEU TOTAL: ";T;" EM ";R;" PAR
TIDAS";
330 IF R>1 THEN PRINT "S"
340 PRINT
350 INPUT "CONTINUAMOS (S/N)?";R$
360 IF R$="N" THEN END
370 GOTO 50
380 X=INT(RND*6)+1
390 RETURN

```

Adivinha um número

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: lançamento de dados.

Dificuldade do jogo: fácil.

O JOGO:

É um jogo de azar, de concepção muito simples, mas no entanto bastante apaixonante: o jogador introduz a sua parada e aposta num número, de 1 a 6; são então lançados três dados; se o número aparece duas vezes ele duplica a sua parada; se aparece três vezes, ele triplica a parada.

O PROGRAMA:

O lançamento dos dados é realizado por uma rotina colocada nas linhas 1000 e 1010.

A impressão é realizada pelas linhas 150 e 160, tendo a linha 145 deslocado as tabulações.

As linhas 180 a 210 calculam o ganho das diversas configurações possíveis.

A totalização é efectuada na linha 230 para os ganhos e na linha 510 para as perdas.

A variável R\$, testada na linha 550, determina a eventual continuação do jogo.

PARA JOGAR:

A iniciação de execução do programa utilizando RUN determina a impressão de uma primeira mensagem: SUA APOSTA; responde-se com um número seguido de ENTER. Obtém-se então a impressão de uma segunda mensagem: ESCOLHA UM NUMERO DE 1 A 6; faz-se entrar um número seguido de ENTER; os três dados, referenciados pelas letras A, B e C, são então impressos, assim como uma mensagem indicando o seu ganho ou a sua perda. Aparece então uma indicação sobre o seu total actual, assim como a pergunta CONTINUAMOS (S/N)?; teclar S para continuar, N para acabar (seguido de ENTER).

EXTENSÕES POSSÍVEIS:

Facilmente se poderá introduzir controlos (aposta mínima, jogada proposta de 1 a 6, etc.). Poder-se-á também desenhar dados com um grafismo mais elaborado, introduzir a possibilidade de vários jogadores, até mesmo fazer participar o AMSTRAD.

```

10 REM ADIVINHA UM NUMERO
20 RANDOMIZE(TIME)
40 G=0
50 CLS
60 INPUT "SUA APOSTA: ",N
70 PRINT
80 PRINT "ESCOLHA UM NUMERO DE 1 A 6:":
PEN 2:INPUT "",D:PEN 1
90 GOSUB 1000
100 A=X
110 GOSUB 1000
120 B=X
130 GOSUB 1000
140 C=X
145 ZONE 10
150 PRINT:PRINT:PRINT ," A"," B"," C"
160 PRINT:PEN 2:PRINT ,A,B,C:PEN 1
170 T=0
180 IF D=A THEN T=T+N
190 IF D=B THEN T=T+N
200 IF D=C THEN T=T+N
210 IF D=0 THEN 500
220 PRINT:PRINT:PRINT "GANHOU ";T
230 G=G+T
240 GOTO 520
500 PEN 3:PRINT:PRINT:PRINT "PERDEU":PEN
1
510 G=G-N
520 PRINT:PRINT:PRINT "TOTAL ACTUAL: ";G
530 PRINT:PRINT
540 INPUT "CONTINUAMOS (S/N)?":R#
550 IF R#="N" OR R#="n" THEN END
560 GOTO 50
1000 X=INT(RND*6)+1
1010 RETURN

```

O jogo de Witthof

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: reflexão (dois jogadores).

Dificuldade do jogo: média.

O JOGO:

Dispõe-se de uma grelha quadrada de 10 por 10 casas. Aparece um ponto, fora da diagonal, e os jogadores deverão deslocá-lo um de cada vez, seguindo somente três direcções, direita, baixo e diagonal, baixo e direita. O primeiro jogador a chegar a (9,9) será considerado vencedor ou vencido, consoante o que for convencionalizado no início do jogo.

O PROGRAMA:

As linhas 40, 50 e 60 permitem definir uma posição de partida, fora da diagonal.

A linha 80 e o ciclo I (90-110) desenham o campo de jogo.

A linha 130 imprime a posição do ponto móvel [uma face que se torna sorridente quando atinge a casa de chegada (9,9)].

O teste de fim de jogo é efectuado na linha 140.

A mudança de jogador é assegurada pela linha 200.

As direcções de deslocação disponíveis, 1, 2 e 3, são descodificadas por três rotinas que começam nas linhas 220, 240 e 250.

PARA JOGAR:

Teclar RUN para iniciar a execução do programa. O campo de jogo é impresso no écran, assim como a posição de partida.

A seguir surge a mensagem JOGADOR 1 e por baixo a pergunta DIRECCAO (1, 2, 3)?, à qual se responde teclando um desses três números, seguido de ENTER.

Vem em seguida a vez do jogador 2.

O jogo termina quando um dos jogadores atinge a casa (9,9). Teclar então ESC duas vezes para recomeçar o programa.

EXTENSÕES POSSÍVEIS:

Pode-se antes de mais melhorar os controlos (a saída do campo de jogo não é interdita, por exemplo), melhorar em seguida a impressão (grafismo, cor), alargar o jogo a vários jogadores, prever um programa que permitiria que o AMSTRAD jogasse, etc.

```

10 REM JOGO DE WYTHOF
20 RANDOMIZE (TIME)
30 DEFINT X,Y
40 X=RND*5
50 Y=RND*5
60 IF Y=X THEN 50
70 CLS
80 PRINT "      0  1  2  3  4  5  6  7  8
  9"
90 FOR I=0 TO 9
100 PRINT:PRINT " ";I;" . . . . .
      "
110 NEXT I
120 H=1
130 LOCATE 3*X+6,2*Y+3:PRINT CHR$(225+(X
=9 AND Y=9))
140 IF X=9 AND Y=9 THEN PEN 1: GOTO 140
150 PEN H+1
160 LOCATE 1,23:PRINT STRING$(30,32)
170 LOCATE 1,23:PRINT "JOGADOR ";H;:INPU
T "DIRECCAO (1, 2, 3)?";D$
180 D=VAL(D$)
190 ON D GOSUB 220,240,250
200 H=2-ABS(H-1)
210 GOTO 130
220 X=X+1
230 RETURN
240 GOSUB 220
250 Y=Y+1
260 RETURN

```

«Tabelas»

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: jogo de destreza.
Dificuldade do jogo: média.

O JOGO:

Utiliza a ideia da simulação de um bilhar.

De um lado e outro do écran encontram-se a bola e o alvo a visar; uma série de algarismos e letras, no alto do écran, permitem escolher o local onde se quer fazer tabela.

A bola vai então partir para o alto do écran, fazer tabela nesse sítio, eventualmente volta a bater em baixo, ressalta, e assim por diante até atingir a linha de chegada; se o fizer suficientemente perto do alvo, teremos ganho e será impresso o número de tabelas.

O PROGRAMA:

A linha 60 imprime os algarismos e as letras no alto do écran.

As linhas 70 e 80 definem as posições iniciais da bola e do alvo.

As linhas 100 e 110 efectuam a sua impressão.

O símbolo corresponde à direcção escolhida é feito entrar na linha 120.

O ciclo J (180-220) realiza a deslocação da bola.

A linha 210 imprime a nova posição.

A mudança de direcção, aquando do encontro com uma tabela, é efectuada pela linha 220.

A linha 240 imprime o texto de fim de partida.

O número de tabelas é contado na linha 200.

PARA JOGAR:

Faz-se correr o programa teclando RUN.

O campo de jogo é impresso no écran, assim como as posições iniciais da bola e do alvo. Escolhe-se o local onde se quer fazer tabela com a bola de (1 a Z), e faz-se ENTER.

A trajectória é então impressa, seguida (se for caso disso) da mensagem BRAVO! X TABELAS.

Pode-se começar outra partida premindo uma tecla qualquer, e parar em qualquer momento teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Será interessante, utilizando um grafismo mais apurado, desenhar um verdadeiro bilhar, com o seu pano verde, e indicar a direcção por um segmento de recta, simbolizando o taco do bilhar, deslocando-se em redor da bola de bilhar. Podem também ser acrescentados buracos, usar várias bolas, sonorizar as tabelas ou os choques com outras bolas, contabilizá-los para fazer uma partida com vários jogadores, etc.

```
10 REM "TABELAS"
20 RANDOMIZE(TIME)
30 DEFINT X,Y
40 INK 0,9:INK 3,3
50 CLS
60 PRINT " 0123456789ABCDEFGHIJKLMNOPQR
STUVWXYZ"
70 X=4+RND*18
80 Y=2+RND*18
90 PEN 3
100 LOCATE 1,X:PRINT CHR$(164)
110 LOCATE 40,Y:PRINT "]"
120 PEN 1:LOCATE 1,25:INPUT "A SUA JOGAD
A: ",R$:PEN 3
130 R$=UPPER(R$)
140 V=3+ASC(R$)-ASC("0")+7*(R$>="A")
150 S=0
160 T=X
170 DT=(2-X)/(V-1)
180 FOR J=2 TO 40
190 T=T+DT
200 IF T<2 OR T>25 THEN DT=-DT:T=25+23*(
DT>0):S=S+1
210 LOCATE J,T:PRINT CHR$(164);
220 NEXT J
230 LOCATE 1,12
240 IF ABS(T-Y)<2 THEN PRINT "BRAVO!";S;
"TABELAS";STRING$(-(S>1),"S")
250 INPUT "OUTRA PARTIDA (S/N)?";R$:PEN
1
260 IF R$="S" THEN RUN
```

«Jackpot»

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: azar (um jogador).

Dificuldade do jogo: fácil.

O JOGO:

É uma variante simplificada de um jogo de azar clássico, no qual o jogador deve fazer uma entrada inicial. Se obtém determinadas combinações de símbolos, ganha (mais ou menos, consoante a aposta e a combinação obtida).

Aqui, os símbolos são substituídos por algarismos, e o facto de haver três algarismos idênticos permite ganhar. Existe mais de uma combinação particular que permite fazer *jackpot*.

O PROGRAMA:

O ciclo I (70-90) permite o desenho da máquina; o ciclo I (110-130) faz a impressão dos lugares (vazios) dos algarismos.

A aposta far-se-á entrar na linha 150: se se aposta 0, a partida termina (salto para a linha 350).

As linhas 160, 180 e 200 permitem a tiragem de três algarismos, chamando uma rotina que se encontra na linha 330.

O resultado é impresso pela linha 220, a seguir testado na linha 240, e depois na linha 310, se se ganhou.

PARA JOGAR:

Fazer correr o programa com RUN; ao ser impressa a pergunta A SUA ENTRADA?, responde-se fazendo entrar um número suficientemente elevado, do qual as vossas apostas serão descontadas.

Depois tecléia-se ENTER e faz-se entrar uma primeira aposta (diferente de 0), seguida de ENTER.

Os três algarismos tirados são então impressos durante um breve instante, assim como a aposta feita e o resultado obtido (ganho ou perda).

Pode-se então apostar de novo, fazer ENTER e assim por diante. Para terminar basta responder 0 à pergunta A SUA ENTRADA?.

O restante do vosso «monte» inicial será então apresentado no écran.

EXTENSÕES POSSÍVEIS:

Poder-se-á encarar a hipótese de melhorar seriamente os controlos, utilizar a cor e possibilidades gráficas mais elaboradas para simular melhor um verdadeiro «Jackpot».

```
10 REM "JACKPOT"
20 DEFINT X
30 RANDOMIZE (TIME)
40 INK 0,9:BORDER 9
50 INK 1,1:INK 2,16,23:INK 3,3
60 CLS
70 FOR I=1 TO 5
80 LOCATE 10,I+4:PAPER 3:PRINT STRING$(2
1,32)
90 NEXT I
100 PAPER 0
110 FOR I=1 TO 3
120 LOCATE 14,5+I:PRINT "      ":LOCATE 19
,5+I:PRINT "      ":LOCATE 24,5+I:PRINT "
"
130 NEXT I
140 LOCATE 1,14:INPUT "A SUA ENTRADA?";M
150 IF M=0 THEN 350
160 GOSUB 330
170 A=X
180 GOSUB 330
190 B=X
200 GOSUB 330
210 PEN 2
220 LOCATE 14,7:PRINT A:LOCATE 19,7:PRIN
T B:LOCATE 24,7:PRINT X
230 PEN 1
240 IF A=B AND A=X THEN 300
250 N=-M
260 LOCATE 7,14:PRINT USING "APOSTA: ###
# ! RESULTADO: #####";M,N
270 LOCATE 4,25:INPUT "CARREGUE EM <ENTE
R> PARA CONTINUAR";R$
280 P=P+N
290 GOTO 60
300 N=M*50
310 IF A=7 THEN N=M*500
320 GOTO 260
330 X=9*RND+1
340 RETURN
350 CLS
360 PRINT "TEM ";P;". ADEUS!"
```

Roleta

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: azar (vários jogadores).

Dificuldade do jogo: fácil.

O JOGO:

É espectacular sobretudo pelo seu aspecto gráfico: uma bola desloca-se por um campo de jogo com casas numeradas de 1 a 36, assim como três casas suplementares, o 00, uma casa P (como par) e uma casa I (como ímpar).

A bola passa pelo menos uma vez por cada casa e depois pára, à segunda volta. Pode-se fazê-la rolar de novo para outra partida.

O PROGRAMA:

O ciclo K (60-120) desenha o campo de jogo, ao qual a linha 130 acrescenta os três símbolos, P, 00 e I.

O ciclo K (150-230) realiza a deslocação da bola, sendo a paragem controlada pela linha 200.

O ciclo J (linha 210) realiza uma ligeira temporização variável para simular a bola a começar a parar.

A linha 210 apaga a bola, deslocando o cursor de impressão para a esquerda.

A linha 240 tira a posição de paragem.

A linha 260 origina as jogadas seguintes.

PARA JOGAR:

Basta fazer RUN e olhar para o écran.

Quando a bola parar bastará premir uma tecla qualquer (excepto o «N») para fazer rolar de novo a bola. Premir N para sair do programa.

EXTENSÕES POSSÍVEIS:

Poder-se-á tentar de diferentes maneiras uma aproximação maior de uma verdadeira roleta: desenhar melhor o campo de jogo, introduzir cores, simular o retardamento da bola, introduzir diferentes possibilidades de apostar e assegurar a gestão dos ganhos correspondentes, manter as contas de vários jogadores e anunciar os resultados de uma mais usual.

```

10 REM ROLETA
20 RANDOMIZE(TIME)
30 DEFINT X
40 INK 0,9:BORDER 9:INK 2,26
50 CLS
60 FOR K=1 TO 4
70 FOR I=1 TO 9
80 S=I+I
90 PRINT TAB(4*I) (S AND K=1)+(S+18 AND
K=2)+(S-1 AND K=3)+(S+17 AND K=4);
100 NEXT I
110 PRINT:PRINT:PRINT:PRINT
120 NEXT K
130 LOCATE 1,3:PRINT "P":LOCATE 1,7:PRIN
T "00":LOCATE 1,11:PRINT "I"
140 PEN 2
150 FOR K=1 TO 4
160 FOR I=1 TO 10+(K=4)
170 IF I=10 AND K<4 THEN LOCATE 1,4*K EL
SE LOCATE 4*I+1,4*K-1
180 PRINT CHR$(231);
190 X=X-1
200 IF X=0 THEN GOTO 260
210 FOR J=1 TO 30*(1-(39/X)*(X>0)):NEXT
J:PRINT CHR$(8);" "
220 NEXT I
230 NEXT K
240 X=1+RND*39
250 GOTO 150
260 PEN 1:PAPER 3:LOCATE 1,20:INPUT "OUT
RA JOGADA (S/N)?";R$:PAPER 0
270 IF R$="N" THEN END
280 GOTO 50

```

Corredores

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: reflexão.

Dificuldade do jogo: difícil.

O JOGO:

Apela para as noções de lógica elementar. O jogador pode apresentar-se à entrada quer do corredor número 1 quer do corredor número 2. Nestas duas entradas existem torniquetes que rodam à passagem do jogador e o deixam passar, uma vez para a direita e a vez seguinte para a esquerda. Além disso existe um torniquete análogo no corredor central. Poderá prever onde (saída A ou B) se vai efectuar a sua próxima saída?

O PROGRAMA:

As linhas 30 a 140 desenham os corredores (e os três torniquetes) com a ajuda da rotina 360, que desloca o cursor de impressão (códigos ASCII 8 e 10).

A linha 180 aguarda a sua escolha.

A descodificação do caminho é feita pelo AMSTRAD utilizando três variáveis lógicas, I, J K, representando os estados dos três torniquetes (linhas 210 a 350).

PARA JOGAR:

Depois de fazer RUN, o corredor é desenhado no écran, sendo visíveis os torniquetes, e dever-se-á teclar 1 ou 2 em resposta à pergunta do AMSTRAD.

Na versão aqui apresentada, a resposta é imediatamente impressa, e o caminho seguido desde o início do jogo, assim como as saídas correspondentes, são mostrados no écran. Isto apresenta a vantagem de permitir a vários jogadores exercer a sua sagacidade tentando prever que vai responder o AMSTRAD a uma dada entrada.

EXTENSÕES POSSÍVEIS:

Serão muito numerosas: é muito fácil imaginar um corredor mais complexo com outras entradas e saídas, e outros torniquetes. Pode-se imaginar uma apresentação gráfica dos resultados desenhando, durante um breve instante, o estado dos torniquetes após a resposta do jogador. Pode-se também acrescentar um teste de resposta (jogo com vários jogadores), descontar os pontos de cada jogador, etc.

```

10 REM CORREDORES
20 CLS
30 PRINT STRING$(40,154)
40 PRINT
50 PRINT "B";SPACE$(6);STRING$(26,154);S
PACE$(6);"2"
60 PRINT
70 PRINT STRING$(14,154);SPACE$(12);STRI
NG$(14,154)
80 PRINT
90 PRINT "A";SPACE$(6);STRING$(26,154);S
PACE$(6);"1"
100 PRINT
110 PRINT STRING$(40,154)
120 PEN 3:LOCATE 33,3:GOSUB 360
130 LOCATE 14,6:GOSUB 360
140 LOCATE 33,9:GOSUB 360
150 I=0:J=0:K=0
160 R$="":T$=""
170 PEN 2
180 LOCATE 1,17:PRINT CHR$(18);:INPUT "E
NTRE 1 OU 2: ";C$
190 IF C$="" OR C$>="3" OR C$<"1" OR LEN
(C$)<>1 THEN 180
200 R$=R$+C$
210 PRINT ;PRINT R$
220 IF C$="1" THEN 270
230 J=NOT(J)
240 IF J<>0 THEN 330
250 S$="B"
260 GOTO 300
270 I=NOT(I)
280 IF I<>0 THEN 330
290 S$="A"
300 T$=T$+S$
310 LOCATE 1,21:PRINT "SAIDA EM ";S$:PRI
NT:PRINT T$
320 GOTO 180
330 K=NOT(K)
340 IF K<>0 THEN 250
350 GOTO 290
360 PRINT CHR$(205);" ";CHR$(204);
370 PRINT STRING$(2,8);CHR$(10);CHR$(203
);
380 PRINT STRING$(2,8);CHR$(10);CHR$(204
);" ";CHR$(205)
390 RETURN

```

Tabuleiro de xadrez

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: grafismo.

Dificuldade do jogo: variável, consoante a utilização.

O JOGO:

O programa proposto serve de suporte ao programa «Xadrez»; efectivamente, dá a possibilidade de realizar uma apresentação gráfica de boa qualidade para jogos que utilizem um tabuleiro de 8 casas (poder-se-á facilmente ampliá-lo para 10 casas). Foi previsto o posicionamento de letras (de A a H) horizontalmente, e, verticalmente, de números (de 1 a 8).

O PROGRAMA:

As linhas 50, 60 e 70 definem três variáveis, A\$, B\$ e C\$, que contêm os elementos de desenho.

A linha 80 imprime as letras horizontais numa janela, com a sua própria cor de impressão.

O ciclo I (90-150) permite a impressão, sendo o motivo a imprimir apresentado ou não consoante o valor de $(I + J)/2$; a linha 120 realiza a impressão.

PARA JOGAR:

A ordem RUN provoca a apresentação do tabuleiro no écran.

Para utilizar o programa convirá ter em conta a localização das casas utilizáveis, que são os centros dos quadrados negros ou cinzentos. As casas são obtidas em largura por uma tabulação da forma $3 * J$, variando J de 1 a 8, e em altura por $2 * I$.

EXTENSÕES POSSÍVEIS:

Poder-se-á utilizar este programa para qualquer espécie de jogos que se desenrolem num tabuleiro (xadrez, damas, etc.).

```

10 REM TABULEIRO DE XADREZ
20 CLS
30 INK 0,0:BORDER 0:INK 1,26:INK 2,2
40 PEN #1,2:LOCATE #1,1,1
50 A#=STRING$(3,140)
60 B#=STRING$(3,143)
70 C#=STRING$(3,131)
80 PRINT #1,"      A B C D E F G H"
:PRINT #1
90 FOR I=1 TO 8
100 FOR J=1 TO 8
110 IF (I+J)/2=INT((I+J)/2) THEN 130
120 LOCATE 3*J+1,2*I:PRINT A#:LOCATE 3*J
+1,2*I+1:PRINT B#:LOCATE 3*J+1,2*I+2:PRI
NT C#
130 NEXT J
140 PRINT #1,I:PRINT #1
150 NEXT I
160 GOTO 160

```

Jogo da fava

REFERÊNCIAS:

Dificuldade de programação: média.
Categoria de jogo: reflexão (dois jogadores).
Dificuldade do jogo: média.

O JOGO:

Apresentando uma certa analogia com o «Jogo de Wytthof», esta versão oferece de facto quatro possibilidades de deslocação; aqui, a posição de partida é fixa (0,0) e poderá eventualmente ser modificada.

As direcções de deslocação disponíveis são: direita, baixo, diagonal baixo e direita, diagonal alto e direita.

Em princípio joga-se com a regra: ganha o primeiro a chegar a (9,9).

O PROGRAMA:

A linha 60 e o ciclo I (70-90) desenham o campo de jogo.

A linha 130 realiza o teste de fim de partida.

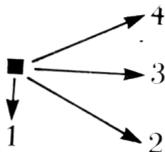
A linha 180 permite executar uma das rotinas de deslocação.

Cada jogador dispõe de uma janela de impressão com uma cor, janela que é sobreposta ao desenho do campo de jogo.

A linha 140 assegura a mudança de jogador.

PARA JOGAR:

Faz-se correr o programa com RUN e o campo de jogo é apresentado, assim como a pergunta DIRECCAO (1, 2, 3, 4)?; as quatro direcções propostas correspondem às seguintes deslocações:



O jogador 1 faz entrar a sua deslocação e teclia ENTER.

O movimento correspondente é realizado e depois obtém-se a impressão das mensagens JOGADOR 2 e DIRECCAO.

O jogador 2 procede do mesmo modo e assim por diante, até que um dos jogadores chegue a (9,9), caso em que o jogo terminará.

EXTENSÕES POSSÍVEIS:

Pode-se imaginar a introdução, no começo da partida, de casas-armadilhas (tentando então cada um fazer o outro ir para uma dessas casas).

Pode-se também jogar com diversos símbolos, se possível de cores diferentes, deslocados simultaneamente, encerrar a hipótese de campos de jogo triangulares, de dimensões arbitrárias, etc.

```
10 REM JOGO DA FAVA
20 DEFINT I,X,Y
30 CLS
40 INK 0,13:INK 1,9:INK 2,15:INK 3,3
50 PEN #1,2:PEN #2,1
60 PEN 3:PRINT "      0  1  2  3  4  5  6
   7  8  9"
70 FOR I=1 TO 9
80 PRINT:PRINT I;"
   "
90 NEXT I
100 X=0:Y=1
110 J=2
120 LOCATE #J,3*X+6,2*Y+1:PRINT #J,CHR$(
248+J)
130 IF X=9 AND Y=9 THEN 200
140 J=1+J MOD 2
150 LOCATE #J,1,23:PRINT CHR$(20);
160 PRINT #J,"JOGADOR ";J;"- DIRECCAO (1
";CHR$(241);", 2";CHR$(205);", 3";CHR$(2
43);", 4";CHR$(204);"?)?";:INPUT #J,"",I$
170 I=VAL(I$)
180 ON I GOSUB 220,240,250,270
190 GOTO 120
200 LOCATE #J,1,23:PRINT #J,"O JOGADOR "
;J;" GANHOU...."
210 GOTO 210
220 Y=Y+1
230 RETURN
240 GOSUB 220
250 X=X+1
260 RETURN
270 GOSUB 250
280 Y=Y-1
290 RETURN
```

Jogo da glória aleatório

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: azar (dois jogadores).
Dificuldade do jogo: fácil.

O JOGO:

O jogo retoma a antiquíssima ideia do jogo da glória mas acrescentando-lhe a possibilidade oferecida pelo computador de escolher ao acaso as casas «com problemas».

A regra do jogo não tem grande interesse, porque os jogadores não intervêm em qualquer momento no desenrolar da partida, que é completamente gerida pelo AMSTRAD; bastará premir qualquer tecla após cada lançamento de dado e observar o que se passa.

O PROGRAMA:

A linha 80 permite chamar a rotina 280 a 230, que realiza o controlo da posição dos jogadores e a apresentação dessa posição. Esta apresentação será, aliás, refeita no decorrer da partida, para tomar em consideração a deslocação realmente efectuada (que não é necessariamente a indicada pelo dado).

A linha 90 ocupa-se da mudança de jogador, e a linha 100 efectua o lançamento do dado.

A linha 370 trata do caso em que um dos jogadores termina.

A linha 140 tira um número aleatório entre 0 e 3, que servirá (se for o caso) para determinar a verdadeira deslocação do jogador, remetendo para uma das rotinas de «chance».

As linhas 280 e 290 fazem avançar as peças, e as linhas 300 a 350 redesenham inteiramente o campo de jogo, a cada apresentação.

PARA JOGAR:

Inicia-se o jogo com RUN. O campo de jogo é então desenhado e depois vê-se aparecer a mensagem JOGADOR 1:...

Deve então premir-se qualquer tecla para que seja tomada em conta a jogada do jogador número 1; não se inquiete com o tempo de reacção, que poderá parecer um pouco longo.

A deslocação indicada pelos dados será executada; depois, ao cabo de algum tempo, será eventualmente efectuada uma segunda deslocação: é a deslocação «CHANCE». Obter-se-á então a mensagem JOGADOR:...

O jogador 2 prime por sua vez uma tecla qualquer, e assim por diante...

O jogador que ganha será assinalado por uma mensagem BRAVO no fim da partida.

Pode também parar em qualquer momento teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se aumentar o campo de jogo, introduzir mais acasos, mudar a natureza deles. Poder-se-ia também escolher as casas «CHAN-CE» no início do jogo, acrescentar apresentações gráficas de acordo com isso, etc.

```
10 REM JOGO DA GLORIA ALEATORIO
20 DEFINT I,J,X,Y
30 RANDOMIZE(TIME)
40 INK 0,3:INK 1,8:INK 2,18:INK 3,11
50 CLS
60 J=2
70 LOCATE 1,3:PRINT " 0123456789ABCDEFGHIJ
HIJKLMNOPQRSTUVWXYZ"
80 GOSUB 280
90 J=1+J MOD 2
100 I=1+RND*5
110 PEN J+1:PRINT CHR*(7)::LOCATE 8,14:P
RINT "JOGADOR ";J;"- A SUA JOGADA: ";I
120 PRINT TAB(10)"CARREGUE EM <ENTER>";
CHR*(20)::INPUT " ",R#
130 GOSUB 280
140 I=RND*3
150 F=1+RND*4
160 ON F GOSUB 180,200,220,260,250
170 GOTO 80
180 I=I+3
190 RETURN
200 I=-I
210 RETURN
220 I=2*I
230 RETURN
240 I=0
250 RETURN
260 IF J=1 THEN X=RND*30 ELSE Y=RND*30
270 RETURN
280 IF J=1 THEN X=X+I*((X+I)<=35)*((X+I)
>=0)
290 IF J=2 THEN Y=Y+I*((Y+I)<=35)*((Y+I)
>=0)
300 FOR I=0 TO 35
310 LOCATE 3+I,5:PRINT " ";
320 IF I=X THEN LOCATE 3+I,5:PEN 2:PRINT
"1";
330 LOCATE 3+I,7:PRINT " ";
340 IF I=Y THEN LOCATE 3+I,7:PEN 3:PRINT
"2";
350 NEXT I
360 FOR I=1 TO 600:NEXT I
370 IF X=35 OR Y=35 THEN LOCATE 1,20:PRI
NT "BRAVO":GOSUB 390:END
380 RETURN
390 SOUND 1,478:SOUND 1,319:SOUND 1,379
400 RETURN
```

Nível 2

Nome do programa

Temas e particularidades

Dado viciado	Para utilizar noutros jogos, se você for desonesto.
Um dado	Tiragem de um dado, com grafismo.
Visões	Um jogo com as palavras, no computador.
421	Uma simulação simplificada, vários jogadores.
O electricista maluco	Lógico e difícil, para um só jogador.
A corrida ao 20	Jogo de reflexão, para dois jogadores ou mais.
Música	Reveja as suas lições de solfejo, graças ao computador.
Póquer	Uma base de partida para afinar o seu programa.
O jogo do salta-carneiros	Um jogo de reflexão, para dois jogadores.
Letras	Um jogo com palavras, muito simples.
Maquinapalavra	Deixe o computador criar vocábulos novos.
Espião	Diga-lhe que a ama — em código.
Totoloto	Ganhe sem problemas, na próxima tiragem.
Tiragem de um dado viciado	Reveja os seus conhecimentos de estatística.
O jogo de Marienbad	Um jogo de lógica para dois jogadores, numa versão alargada.
Compõe	Fazer música é, antes de mais, escrevê-la...
Dia da semana	Aproveite para fazer o ponto dos aniversários.
Habilidade com cartas	O seu computador tem mais de um truque no saco.
O indesejável	Você é proprietário. Conseguirá pôr as mãos neste locatário indesejável?
O jogo da vida	Um grande clássico, em duas versões: manual e automática.

Dimensionamento de um quadro. Busca de um elemento num quadro e numa cadeia. A instrução DATA

A INSTRUÇÃO DIM OU INSTRUÇÃO DE DIMENSIONAMENTO DE UM QUADRO

As estruturas de quadros são correntemente utilizadas em programação, revelando-se mesmo indispensáveis num grande número de casos. Se, por exemplo, você quiser fazer entrar as suas receitas e despesas para todos os dias do mês (suporemos, para simplificar, meses de trinta dias) poderia definir trinta variáveis chamadas PRIMEIRO DIA, SEGUNDO DIA, etc. É mais fácil e mais racional utilizar um ciclo I, com I a variar de 1 a 30, e uma variável R(I) para as receitas e outra variável D(I) para as despesas. Agindo assim, você teria, sem o saber, utilizado uma estrutura de quadro a uma dimensão; isto só poderá ser admitido no AMSTRAD se você previamente tiver prevenido que irá utilizar um tal quadro, dando uma instrução DIM R(30) para as receitas e DIM D(30) para as despesas.

Poder-se-ia também imaginar conservar despesas e receitas num mesmo quadro, dando-lhe desta vez duas dimensões; por exemplo, DIM RD(30,2) [ou DIM RD(2,30)]. Assim, com DIM RD(30,2) poder-se-á convencionar que RD(I,1) representa as receitas e RD(I,2) as despesas. São também possíveis quadros de mais que duas dimensões.

Podem também conceber-se quadros cujos elementos são cadeias de caracteres, e já não números, com o máximo de 255 caracteres.

Assim, DIM NOME\$(12) reserva doze posições na memória para palavras com o máximo de 255 caracteres.

No AMSTRAD não é obrigatório dimensionar um quadro que contenha menos de dez elementos.

A BUSCA DE UM ELEMENTO NUM QUADRO: R(I)

Se tivermos um quadro a uma dimensão, numérico ou alfanumérico, definido por uma instrução como DIM R(30) ou DIM N\$(20), o elemento de ordem I deste quadro é determinado por R(I) [ou N\$(I)]; será então manipulado exactamente como uma variável ordinária (atribuição, cálculo, testes, etc.).

Assim, INPUT R(3) permite fazer entrar manualmente o valor do terceiro elemento numérico do quadro R. PRINT R\$(5) permite a apresentação do quinto elemento do quadro alfanumérico R\$, etc.

A BUSCA DE UM ELEMENTO NUMA CADEIA: MID\$

Para extrair o elemento de ordem I da cadeia alfanumérica A\$ utilizar-se-á a instrução MID\$.

Assim, se A\$ = «BOM DIA», MID\$(A\$,3,1) dá-nos a letra M; além disso, MID\$(A\$,3,3) dá-nos a subcadeia «M D».

O TRATAMENTO DE DADOS MÚLTIPLOS. INSTRUÇÃO READ, DATA, RESTORE

Para encher facilmente um quadro, numérico ou alfanumérico, será interessante utilizar linhas de DATA, que contém os dados inscritos uns a seguir aos outros, separados por uma vírgula.

Assim, poder-se-á escrever:

```
DATA VIEGAS,JULIO,13,Rua Madressilva...
```

Estes dados podem ser lidos em ciclos FOR...NEXT por intermédio da instrução READ. Pode-se recomeçar a leitura no início dos DATA, com a ajuda da instrução RESTORE.

Dado viciado

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: lançamento de dados.
Dificuldade do jogo: fácil.

O JOGO:

Queremos fabricar um dado viciado segundo as nossas próprias conveniências. O método aqui proposto consiste em definir à partida uma probabilidade de realização para cada tipo de tiragem (ou seja, a possibilidade de tirar um 1, um 2, um 3, etc., de cada uma destas probabilidades estar compreendida entre 0 e 1, e de a soma de todas as probabilidades ser um número muito próximo de 1).

O PROGRAMA:

Utiliza o gerador de números aleatórios do AMSTRAD, que se supõe ter a qualidade conveniente (poder-se-á facilmente utilizar qualquer outro gerador, se for necessário). O ciclo I (60-130) ocupa-se da entrada de cada uma das probabilidades, com verificação em cada entrada e no total (linha 100); cada probabilidade é conservada numa variável E(I): linha 80 do programa.

O lançamento do dado é efectuado com a ajuda de um outro ciclo I (180-200), e a linha que permite a batota é a 190.

Pode fazer-se nova tiragem premindo uma tecla (linha 220).

PARA JOGAR:

Depois de se fazer correr o programa com RUN obtém-se a apresentação da pergunta **PROBABILIDADE DE SAIR 1?**, à qual se deve responder com um número entre 0 e 1. Convém não esquecer que o total de todas as probabilidades entradas não deverá em caso algum ultrapassar 1.1 (tolerância de 0.1).

Poderão surgir mensagens de erro e o programa será eventualmente reiniciado. Quando todas as probabilidades tiverem sido definidas será automaticamente efectuada a primeira tiragem e bastará premir qualquer tecla para fazer as tiragens seguintes. Para-se o programa tecando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Será, evidentemente, muito tentador utilizar sorratamente este pequeno programa para melhorar as suas probabilidades de ganhar

num jogo de azar que utilize lançamentos de dados. Poder-se-á então encarar a ideia de «congelar» os dados definindo o dado viciado numa variável A\$, etc. Poder-se-á também pensar em apresentar os resultados correspondentes a uma série de lançamentos de dados de forma espectacular, que põe em evidência a batota...

```
10 REM DADO VICIADO
20 DEFINT N
30 RANDOMIZE(TIME)
40 CLS
50 DIM E(6)
60 FOR I=1 TO 6
70 LOCATE 25,3:PRINT CHR$(18);
80 LOCATE 1,3:PRINT "PROBABILIDADE DE SA
IR ";I;:INPUT "?",E(I)
90 P=P+E(I)
100 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 24
0
110 LOCATE 6*I-3,10:PRINT "DADO ";I
120 LOCATE 6*I-3,12:PRINT E(I)
130 NEXT I
140 IF P<0.9 THEN PRINT CHR$(7);:RUN
150 LOCATE 1,3:PRINT CHR$(18);
160 FOR I=1 TO 500:NEXT I
170 N=1+RND*5
180 FOR I=1 TO 6
190 IF N=I AND RND>E(I) THEN 170
200 NEXT I
210 LOCATE 15,17:PAPER 2:PEN 3:PRINT "TI
RAGEM: ";N;:PAPER 0:PEN 1:GOSUB 290
220 LOCATE 3,25:INPUT "CARREGUE EM <ENTE
R> PARA OUTRA TIRAGEM",R$
230 GOTO 170
240 LOCATE 10,25:PAPER 3:PRINT CHR$(7);"
IMPOSSIVEL: 0<=P<=1";:PAPER 0
250 F=P-E(I)
260 FOR J=1 TO 1500:NEXT J
270 LOCATE 10,25:PRINT STRING$(19,32);
280 GOTO 70
290 SOUND 1,478,20:SOUND 1,319,20:SOUND
1,379,20:RETURN
```

Um dado

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: tiragem de um dado (com grafismo).

Dificuldade do jogo: nenhuma.

O JOGO:

O programa propõe simplesmente o lançamento de um dado aleatório e a sua representação em grande formato.

Poderá facilmente ser utilizado como rotina, ou subprograma, em qualquer tipo de jogo que utilize o lançamento de um ou mais dados.

O PROGRAMA:

A linha 50 utiliza o gerador de números aleatórios do AMSTRAD para tirar um número de 1 a 6.

O ciclo J (70-90) desenha os bordos do dado.

As linhas 120 a 170 são utilizadas para desenhar os pontos.

As linhas 180 e 190 permitirão, se for o caso, refazer uma tiragem.

PARA JOGAR:

Fazer correr o programa com RUN.

Faz-se entrar um número inteiro para inicializar o gerador e será apresentado um dado, a cores e em formato grande.

À pergunta: OUTRA TIRAGEM (S/N)? responde-se S para voltar a lançar o dado.

EXTENSÕES POSSÍVEIS:

Poder-se-á introduzir uma pequena melodia para enfeitar cada lançamento, fazer desfilar os pontos como se o dado rolasse sobre uma mesa, etc.

```

10 REM UM DADO
20 RANDOMIZE(TIME)
30 CLS
40 C=INT(RND*27):INK 1,C+(C=1)
50 A=INT(RND*6+1)
60 LOCATE 12,5:PRINT CHR$(150);STRING$(1
1,154);CHR$(156)
70 FOR J=0 TO 10
80 LOCATE 12,6+J:PRINT CHR$(149):LOCATE
24,6+J:PRINT CHR$(149)
90 NEXT J
100 LOCATE 12,17:PRINT CHR$(147);STRING$
(11,154);CHR$(153)
110 RESTORE 1000:FOR I=1 TO 3:READ N:SOU
ND 1,N:NEXT I
120 IF A<>1 THEN LOCATE 15,8:PRINT CHR$(
231)
130 IF A>=4 THEN LOCATE 21,8:PRINT CHR$(
231)
140 IF A=6 THEN LOCATE 15,11:PRINT CHR$(
231):LOCATE 21,11:PRINT CHR$(231)
150 IF A MOD 2 THEN LOCATE 18,11:PRINT C
HR$(231)
160 IF A>=4 THEN LOCATE 15,14:PRINT CHR$
(231)
170 IF A<>1 THEN LOCATE 21,14:PRINT CHR$
(231)
180 PEN 2:LOCATE 1,25:INPUT "OUTRA TIRAG
EM (S/N)?" :R$
190 IF UPPER$(LEFT$(R$,1))="S" THEN PEN
1:GOTO 30
1000 DATA 478,319,379

```

Visões

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: desenho.

Dificuldade do jogo: fácil.

O JOGO:

O AMSTRAD permite-lhe, pela primeira vez, entrar na sua intimidade, deixando-lhe entrever a sua própria representação do mundo! Efectivamente, você irá propor-lhe palavras, não importa quais, e ele vai responder-lhe propondo-lhe uma representação gráfica do que essas palavras representam para ele. O jogo será muito mais divertido se praticado em grupo, e por vezes poderá haver surpresas.

O PROGRAMA:

Um ciclo I (40-120) encarrega-se de realizar os desenhos.

A linha 50 permite transformar em número cada letra da palavra que inicialmente se faz entrar (linha 30). Este número será então traduzido para a base 2 por um ciclo J (linhas 60 a 110); consoante o resultado desta transformação, será ou não impresso um ponto (linha 90).

PARA JOGAR:

Começa-se a execução com RUN, o que conduz à apresentação da pergunta QUE DESENHO?. Responde-se fazendo entrar uma palavra qualquer, que será transformada pelo AMSTRAD depois de se fazer ENTER.

Para recomeçar basta carregar em qualquer tecla; para terminar, carregar em ESC duas vezes.

EXTENSÕES POSSÍVEIS:

A mesma ideia pode ser retomada introduzindo elementos de simetria nos desenhos, ou usando várias cores. Pode-se também imaginar codificar por este meio um certo número de desenhos, que serão introduzidos na máquina quando quisermos e pode-se então criar um jogo em que o AMSTRAD se tornará cada vez mais sábio. Pode-se também realizar um jogo para dois jogadores em que o AMSTRAD lhe pede que encontre a palavra correspondente a um certo desenho, etc.

```

10 REM VISQES
20 CLS
30 INPUT "QUE DESENHO? ";R#
40 FOR I=1 TO LEN(R#)
50 N=9*ASC(MID$(R#,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2)
90 IF S<>0 THEN LOCATE J+15,I+5:PAPER 2:
PRINT " ":PAPER 0
100 J=J+1
110 IF J<10 THEN 70
120 NEXT I
130 LOCATE 11,25:INPUT "CARREGUE EM <ENT
ER>",R#
140 F=INT(RND*27):F=F+(F=1):INK 2,F
150 GOTO 20

```

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: lançamento de dados.
Dificuldade do jogo: média.

O JOGO:

Cada um dos jogadores deverá, à vez, lançar três dados e pode depois escolher, nas duas jogadas seguintes, se deseja ou não voltar a lançar um, dois ou três dados. Marca então uma pontuação correspondente ao número indicado pelos dados, de acordo com as regras escolhidas, e o jogador seguinte procede da mesma maneira.

O PROGRAMA:

Antes de mais faz-se entrar o número de jogadores, na linha 40.

A linha 130 imprime o número do jogador e o número de tentativas já efectuadas.

As novas tiragens são efectuadas pelas linhas 1000, 2000 e 3000.

A variável R\$ gera as segundas e terceiras tiragens. A impressão do resultado é realizado, em qualquer caso, pelas linhas 140-150.

Os testes para o número de tentativas e para a mudança de jogador são efectuados nas linhas 260 e 280.

PARA JOGAR:

O programa é posto a funcionar com RUN, o que leva à apresentação da mensagem NUMERO DE JOGADORES?. Faz-se entrar o número de jogadores, seguido de ENTER. O écran é limpo e surge a mensagem JOGADOR NUMERO 1. TENTATIVA K, PARTIDA P, assim como a tiragem realizada e os nomes dos dados (A, B, C).

Se não se deseja refazer a tiragem basta premir ENTER. A mão passa então ao jogador seguinte.

Para efectuar outra tiragem basta teclar o nome do dado ou dados a retirar e fazer ENTER; por exemplo, para retirar A e C teclia-se AC seguido de ENTER.

EXTENSÕES POSSÍVEIS:

A mais evidente consiste em assegurar a gestão dos resultados dos jogadores, tomando como referência as regras usuais do jogo do 421. Poder-se-á pensar na impressão de um quadro, após cada mão, recapitulando os resultados dos jogadores.

Poder-se-á também melhorar a apresentação dos dados (tomar como referência os diversos programas de apresentação de dados que se encontram nesta mesma obra).

```
10 REM 421
20 RANDOMIZE(TIME)
30 INK 0,0
40 INPUT "NUMERO DE JOGADORES?";N
50 P=1
60 I=1
70 K=0
80 CLS
90 R$="ABC"
100 GOTO 200
110 K=K+1
120 LOCATE 17,1:PEN 3:PAPER 1:PRINT "-42
1-":PAPER 0
130 INK 2,2+I MOD 26:PEN 2:LOCATE 1,4:PR
INT "JOGADOR NUMERO ";I;" TENTATIVA ";K
;" PARTIDA ";P
140 LOCATE 7,7:PRINT A:LOCATE 17,7:PRINT
B:LOCATE 27,7:PRINT C:PEN 1
150 LOCATE 8,9:PRINT "A":LOCATE 18,9:PRI
NT "B":LOCATE 28,9:PRINT "C"
160 R$="":LOCATE 7,14:PRINT CHR$(20);
170 LOCATE 1,14:INPUT "ABC: ";R$
180 IF R$="" THEN 270
190 R$=R$+"DDD"
200 FOR J=1 TO 3
210 GOSUB 4000
220 X$=MID$(R$,J,1)
230 X=ASC(X$)-64:IF X<0 THEN X=4
240 ON X GOSUB 1000,2000,3000
250 NEXT J
260 IF K<3 THEN 110
270 I=I+1
280 IF I<=N THEN 70
290 P=P+1
300 GOTO 60
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=1+INT(RND*6)
4010 RETURN
```

O electricista maluco

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria do jogo: solitário.

Dificuldade do jogo: bastante difícil.

O JOGO:

No écran aparece um prédio com dez janelas, numeradas de 0 a 9. O jogador deverá, a cada tentativa, manobrar um dos interruptores, indicando o seu número (de 0 a 9).

Em princípio a janela correspondente deverá apagar-se, mas é também possível que uma outra se acenda (ou se apague). Trata-se, claro, de obter a extinção das dez janelas com o menor número possível de tentativas.

O PROGRAMA:

Um quadro A, na linha 30, conterà o estado das dez janelas (acesa ou apagada).

O ciclo I (50-70) e o outro ciclo I (90-140) desenham o prédio no seu estado inicial.

A linha 150 verifica se tudo está apagado, caso em que irá à linha 250 para imprimir o número de tentativas.

A linha 160 aguarda a indicação do interruptor que se quer utilizar.

As linhas 170 a 230 definem o resultado da manobra.

PARA JOGAR:

RUN provoca a apresentação do prédio com todas as janelas iluminadas.

À pergunta INTERRUPTOR NUMERO? responde-se indicando o número do interruptor a manobrar, seguido de ENTER.

O resultado da manobra é então mostrado no écran (janelas acendem-se ou apagam-se).

Prosssegue-se até que todas as janelas estejam apagadas, e obtém-se então a apresentação, no écran, do número de tentativas realizadas.

EXTENSÕES POSSÍVEIS:

Poder-se-á pensar num grafismo muito melhorado, com a possível utilização da cor. Poder-se-á também definir um jogo para dois jogadores, aumentar o número de janelas, modificar a fórmula que define se acende ou se apaga...

```

10 REM O ELECTRICISTA MALUCO
20 RANDOMIZE(TIME)
30 DIM A(10)
40 CLS
50 FOR I=1 TO 7
60 PAPER 2:LOCATE 10,I:PRINT STRING$(21,
32):PAPER 0
70 NEXT I
80 T=0
90 FOR I=1 TO 10
100 T=T+A(I)
110 LOCATE 2*I+9,4:PRINT " "
120 IF A(I)=0 THEN LOCATE 2*I+9,4:PRINT
CHR$(224)
130 PAPER 2:PEN 0:LOCATE 2*I+8,6:PRINT I
-1:PAPER 0:PEN 1
140 NEXT I
150 IF T=10 THEN 250
160 PEN 3:LOCATE 1,14:PRINT CHR$(7);CHR$(
20);:INPUT "INTERRUPTOR NUMERO? ";N:PEN
1
170 N=N+1
180 M=M+1
190 A(N)=1-A(N)
200 R=TAN(RND+N/RND-N)-SIN(RND/N)+336*SI
N(8*N)
210 N=INT(10*(R-INT(R)))
220 IF N=0 THEN N=INT(RND*10+1)
230 IF A(N)=1 THEN A(N)=0:GOTO 200
240 A(N)=1:GOTO 80
250 LOCATE 1,14:PRINT "CONSEGUIU EM ";M;
" TENTATIVAS "

```

A corrida ao 20

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: lógica (dois jogadores, ou mais).

Dificuldade do jogo: média.

O JOGO:

É possível imaginar diversas regras para este jogo; aqui usamos apenas a regra usual para dois jogadores, que faz dele um jogo aparentado com os jogos de deslocação, ou de fósforos: quatro grupos de vinte estrelas vão aparecer no início da partida, e cada jogador pode retirar quantas estrelas quiser — mas apenas de um grupo de cada vez. Perde aquele que retira a última estrela.

O PROGRAMA:

O ciclo I (30-50) desenha quatro filas de vinte estrelas.

A variável F, dimensionada na linha 80, servirá para registar o número de estrelas retirado de cada fila.

A variável T conterá o número total de estrelas retiradas (linha 70).

A linha 140 controla se restam estrelas suficientes na linha indicada.

O ciclo J (150-180) apaga as estrelas; a linha 220 testa o fim da partida.

PARA JOGAR:

Fazer correr o programa com RUN; o campo de jogo é apresentado, seguido das inscrições JOGADOR 1 e LINHA, depois CASAS. Responde-se fazendo entrar um número de linha (entre 1 e 4) seguido de ENTER, e o número de estrelas a retirar seguido de ENTER. As estrelas são então suprimidas e passa à vez do segundo jogador.

A partida termina, quando a última estrela tiver sido retirada, com a apresentação do número do jogador que ganha.

EXTENSÕES POSSÍVEIS:

Além da melhoria da qualidade do grafismo e dos controlos, pode-se encarar diversas variantes, modificando, se for necessário, as regras de início: pode-se autorizar a retirada de apenas um número ímpar de estrelas, ou alternar par e ímpar, ou tirar ao acaso o número a retirar, ou ainda jogar com vários jogadores...

```

10 REM A CORRIDA AO 20
20 CLS:PEN 3
30 FOR I=1 TO 4
40 LOCATE 1,2*I:PRINT I;STRING$(20,238)
50 NEXT I
60 PEN 1
70 T=0
80 DIM F(4)
90 B=1
100 LOCATE 1,13:PEN B:PRINT CHR$(20);JOG
ADOR ";B
110 PRINT:INPUT "LINHA: ";L
120 IF L<1 OR L>4 THEN 100
130 PRINT:INPUT "CASAS: ";C
140 IF C+F(L)>20 OR C<=0 THEN 100
150 FOR J=1 TO C
160 LOCATE 24-F(L)-J,2*L:PRINT " ";CHR$(
7);
170 FOR TE=1 TO 200:NEXT TE
180 NEXT J
190 T=T+C
200 F(L)=F(L)+C
210 B=1+B MOD 2
220 IF T<80 THEN 100
230 LOCATE 12,25:PAPER 3:PEN B:PRINT "O
JOGADOR ";B;" GANHOU":SOUND 1,478:SOUND
1,319:SOUND 1,379
240 GOTO 240

```

Música

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo educativo.

Dificuldade do jogo: média.

O JOGO:

Apresenta-se como um exercício de aprendizagem de solfejo: o aparelho anuncia uma nota (em clave de sol), seguida da menção GRAVE ou AGUDA, consoante o seu lugar na pauta. É necessário responder indicando em que linha a nota será impressa.

A nota é então desenhada na pauta, seguida de uma mensagem de BRAVO ou PERDEU, conforme se tenha acertado ou não.

O PROGRAMA:

O ciclo I (60-80) desenha a pauta. Um segundo ciclo I (90-120) imprime os números das linhas.

A linha 130 define o nome das notas e a linha 140 realiza a tira-gem aleatória de uma nota.

O nome da nota é calculado na linha 150 e é impresso pela linha 160, seguido da menção GRAVE ou AGUDA, conforme o caso.

A resposta faz-se entrar na linha 180.

A linha 190 trata da descodificação da resposta.

As linhas 200 a 210 desenhavam a nota.

A linha 230 verifica se se ganhou ou perdeu e imprime a mensagem correspondente.

A linha 240 permite fazer nova tentativa.

PARA JOGAR:

Faz-se correr o programa usando RUN.

Obtém-se então a apresentação da pauta, assim como o nome da nota a colocar na pergunta NUMERO DA LINHA?.

Responde-se fornecendo um número da linha (de 1 a E), seguido de ENTER.

A nota é então desenhada na pauta e aparece no écran a mensagem BRAVO ou PERDEU, conforme o caso.

Carrega-se numa tecla qualquer para continuar e duas vezes em ESC para parar.

EXTENSÕES POSSÍVEIS:

Pode alargar-se a outra clave que não a de sol, pode-se introduzir cor, sons, etc.

```
10 REM MUSICA
20 DEFINT X
30 RANDOMIZE(TIME)
40 INK 0,22:INK 1,7:INK 2,24:INK 3,15
50 CLS
60 FOR I=1 TO 5
70 LOCATE 4,2*I+4:PRINT STRING$(31,154)
80 NEXT I
90 FOR I=1 TO 14
100 LOCATE 1,I+2
110 IF I<=9 THEN PRINT I ELSE PRINT " ";
CHR$(I+55)
120 NEXT I
130 N$="SI LA SOLFA MI RE DO "
140 X=1+RND*13
150 T=(X-1) MOD 7
160 LOCATE 1,18:PRINT "NOTA ";MID$(N$,3*
T+1,3);
170 IF X<=7 THEN PRINT " AGUDA" ELSE PRI
NT " GRAVE"
180 PRINT:INPUT "NUMERO DA LINHA? ";L$
190 L=-((ASC(L$)-48)*(L$<"")-(ASC(L$)-55
)*(L$)="A")
200 PEN 3:LOCATE 18,X:PRINT CHR$(22)+CHR
$(1);CHR$(211);CHR$(8);CHR$(205);CHR$(8)
;CHR$(195)
210 LOCATE 18,X+1:PRINT CHR$(211);CHR$(8
);CHR$(193):LOCATE 17,X+2:PRINT CHR$(143
);CHR$(211);CHR$(22)+CHR$(0);:PEN 1
220 LOCATE 18,24
230 PAPER 3:PEN 2:IF L=X THEN PRINT "BRA
VO";ELSE PRINT "PERDEU";
240 PAPER 0:PEN 1:LOCATE 5,25:INPUT "CAR
REGUE EM <ENTER> PARA CONTINUAR ";R$:GOT
O 50
```

Póquer

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo contra o AMSTRAD.

Dificuldade do jogo: fácil.

O JOGO:

É uma simulação de uma partida de póquer tradicional, efectuada com um baralho de 32 cartas, em que o jogador terá o AMSTRAD como opositor.

O jogador recebe cinco cartas, referenciadas pelos números de 1 a 5, e poderá decidir mudar algumas cartas anunciando quais as que rejeita. A sua nova mão será então apresentada e passa à vez de o AMSTRAD jogar.

O PROGRAMA:

As linhas 50 e 60 definem as cartas. Uma variável N é dimensionada na linha 40, para receber dez cartas do jogador e cinco cartas do AMSTRAD.

A tiragem será efectuada por um ciclo I (80-140) e o controlo por um ciclo J (110-130).

A apresentação das cartas é feita pela rotina 260-370.

Um outro ciclo I (190-210) realiza a segunda tiragem do jogador. A linha 240 permite a segunda apresentação e o ciclo I (260-330), onde se entra directamente após essa apresentação, realizará a apresentação do jogo da máquina.

PARA JOGAR:

A ordem RUN faz correr o programa e são então afixadas as cinco primeiras cartas do jogador.

À pergunta CARTAS REJEITADAS? o jogador responderá 0, seguido de ENTER, se quer acabar a partida. Para jogar responderá simplesmente ENTER se acha que lhe serve o jogo que tem, ou fará entrar os números das cartas que rejeita, seguidos de ENTER.

Por exemplo, se ele teclia 235 e ENTER, o seu novo jogo conservará as cartas 1 e 4. Após isto o seu novo jogo é apresentado, assim como o do AMSTRAD que, neste programa, não tem possibilidade de pedir novas cartas.

Bastará depois carregar em qualquer tecla para iniciar outra partida.

EXTENSÕES POSSÍVEIS:

Serão, evidentemente, muito numerosas, desde uma apresentação gráfica colorida das cartas até à melhoria do jogo do AMSTRAD, introdução das regras sobre as apostas e mesmo, por que não?, possibilidades de *bluff*.

```
10 REM POQUER
20 RANDOMIZE(TIME)
30 INK 0,22:INK 1,2:INK 2,6:INK 3,0
40 DIM N(32)
50 V$="7 8 9 10 VADAREAS"
60 C$=CHR$(226)+CHR$(227)+CHR$(228)+CHR$(229)
70 CLS
80 FOR I=1 TO 15
90 N(I)=INT(RND*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF N(I)=N(J) THEN 90
130 NEXT J
140 NEXT I
150 K=0:P=0
160 LOCATE 1,12:PRINT "CARTAS REJEITADAS
?"
170 GOSUB 260
180 IF R$="" THEN END
190 FOR I=1 TO MIN(5,LEN(R$))
200 N(VAL(MID$(R$,I,1)))=N(I+5)
210 NEXT I
220 CLS
230 P=1
240 GOSUB 260
250 K=K+10
260 FOR I=1 TO 5
270 C=INT(N(I+K)/8)
280 F=N(I+K) MOD 8
290 R$=MID$(V$,2*F+1,2)
300 LOCATE 7*I-2,K+1:PRINT I
310 LOCATE 7*I-3,K+3:PRINT R$;" ";";PEN
(3+(C=1 OR C=2)):PRINT MID$(C$,C+1,1);;P
EN 1
320 LOCATE 7*I-3,K+4:PRINT STRING$(5,208
);
330 NEXT I
340 IF P=1 THEN P=0:RETURN
350 IF K=10 THEN LOCATE 6,25:INPUT "CARR
EGUE EM <ENTER> PARA CONTINUAR",R$:GOTO
70
360 LOCATE 18,12:INPUT R$
370 RETURN
```

*O jogo do salta-carneiro*¹

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: reflexão (dois jogadores).

Dificuldade do jogo: fácil.

O JOGO:

Embora a ideia seja muito simples, não será forçosamente evidente descobrir uma estratégia conveniente. Dispõe-se de três carneiros azuis e de três carneiros encarnados, representados por símbolos gráficos, que ocupam oito casas; os carneiros podem deslocar-se de uma casa para outra casa livre. Poderão ser usadas várias regras, consoante o vosso próprio cálculo do interesse desta ou daquela: quer obrigando os carneiros a colocarem-se na primeira casa disponível, eventualmente saltando por cima dos carneiros adversários, quer deixando-lhes a liberdade de escolher a sua casa de chegada — o que pode tornar intermináveis as partidas. A escolha é sua.

O PROGRAMA:

O ciclo I (70-100) inicializa um quadro A(I), que contém as posições dos carneiros. O ciclo I (120-170) desenha o campo de jogo e os carneiros.

As linhas 190 e 210 ocupam-se do texto e das entradas.

A janela #1, definida na linha 50, serve para preservar o desenho do campo de jogo durante a entrada da jogada na linha 210.

As linhas 220 e 230 controlam a validade da jogada. A mudança de jogador é realizada na linha 260.

PARA JOGAR:

Começa-se com RUN a execução do programa. Obtém-se então a apresentação da mensagem JOGADOR 1, que indica o número do jogador, seguido da pergunta INICIO, FIM?.

Responde-se indicando o número da peça a deslocar, seguido de ENTER, depois o número da casa de chegada, seguido de ENTER.

Realiza-se então a deslocação; depois o segundo jogador procede da mesma forma. Em princípio ganha aquele dos jogadores que conseguir colocar os seus três carneiros lado a lado nas posições inicialmente ocupadas pelo adversário.

¹ Embora *saute-mouton* corresponda ao nosso jogo do eixo, usou-se a tradução literal por motivos que o próprio texto evidencia. (N. do T.)

EXTENSÕES POSSÍVEIS:

Começar-se-á por acrescentar um controlo do jogador que ganha.

Pode-se igualmente melhorar a apresentação desenhando verdadeiros carneiros, tornar o jogo mais atraente numerando-os e impondo uma certa ordem de chegada.

Finalmente, poder-se-á definir uma estratégia para o computador, fixando as regras.

```
10 REM O JOGO DO SALTA-CARNEIRO
20 CLS
30 INK 0,9:INK 3,3
40 DIM A(8)
50 WINDOW #1,1,40,13,13
60 A(1)=0:A(8)=0
70 FOR I=1 TO 3
80 A(I+1)=1
90 A(I+4)=-1
100 NEXT I
110 H=1
120 FOR I=1 TO 8
130 LOCATE 4*I,8:PEN 1:PRINT I
140 LOCATE 4*I+1,6:PRINT " "
150 IF A(I)=1 THEN LOCATE 4*I+1,6:PEN 2:
PRINT CHR$(189)
160 IF A(I)=-1 THEN LOCATE 4*I+1,6:PEN 3
:PRINT CHR$(191)
170 NEXT I
180 J=1-(H=-1)
190 LOCATE 1,11:PEN J+1:PRINT "JOGADOR "
;J
200 WINDOW SWAP 0,1:PEN J+1
210 INPUT "INICIO, FIM? ";D,F
220 IF D<1 OR D>8 OR F<1 OR F>8 THEN 210
230 IF A(D)<>H OR A(F)<>0 THEN 210
240 A(D)=0
250 A(F)=H
260 H=-H
270 WINDOW SWAP 1,0:GOTO 120
```

Letras

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: reflexão (dois jogadores).
Dificuldade do jogo: média.

O JOGO:

A regra inspira-se fortemente num jogo de letras muito conhecido em França, praticado nomeadamente numa emissão televisiva («Os números e as letras»). O AMSTRAD escolhe ao acaso um certo número de símbolos (vogais e consoantes) do alfabeto, e cada jogador deverá indicar o comprimento da palavra que conseguiu obter (o aparelho não verificará se a palavra é correcta!).

Seguidamente é apresentada a pontuação total de cada jogador.

O PROGRAMA:

O ciclo I (90-120) escolhe no alfabeto entre sete e doze letras, em função da variável aleatória L, tirada na linha 60. Para evitar a existência de demasiadas consoantes utiliza-se na linha 110 uma pequena astúcia, que consiste em refazer ao acaso uma tiragem no quadro A\$, que contém vogais (podemos ter mais ou menos vogais utilizando outros RND).

O ciclo 130-170 pergunta a cada jogador qual o comprimento da sua palavra.

As linhas 180 e 190 comparam os resultados e totalizam a pontuação de cada jogador, apresentada no écran pelas linhas 200 e 210.

PARA JOGAR:

Inicia-se a execução do jogo com RUN.

As letras são então apresentadas e a seguir a pergunta JOGADOR 1. QUE COMPRIMENTO?, à qual o primeiro jogador responderá dando o comprimento da palavra que obteve (no entanto o outro jogador deverá verificar se a palavra é correcta); depois o segundo jogador fará o mesmo e a pontuação de cada jogador será então impressa no écran.

Para continuar a partida basta carregar em qualquer tecla.

Para acabar deve-se fazer ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á, bem entendido, pensar em introduzir uma temporização antes de pedir as respostas aos jogadores.

Poder-se-á também fazer entrar não somente o comprimento da palavra mas a própria palavra, devendo o adversário responder à pergunta CORRECTO OU NAO?.

Pode-se também alargar o jogo a mais de dois jogadores. E por que não tentar definir um jogo contra o AMSTRAD, ao qual se forneceria anteriormente uma lista de palavras bastante longa, de que ele imprimiria as letras desordenadas, eventualmente com algumas letras de complemento escolhidas ao acaso?

```
10 REM LETRAS
20 RANDOMIZE(TIME)
30 DEFINT L
40 S=0:J=0
50 CLS
60 L=RND*5+7
70 A$="AEIOUEA"
80 LOCATE 1,6:PRINT "LETRAS : "
90 FOR I=1 TO L
100 LOCATE 8+2*I,6:PRINT CHR$(INT(RND*26
+65))
110 IF RND<0.4 THEN LOCATE 8+2*I,6:PRINT
MID$(A$,INT(RND*7+1),1)
120 NEXT I
130 I=1
140 LOCATE 1,6+4*I:PEN I+1:PRINT "JOGADO
R ";I;:INPUT ". QUE COMPRIMENTO: ";L:PEN
1
150 IF I=1 THEN P=L
160 I=I+1
170 IF I<3 THEN 140
180 S=S+(P AND P>L)
190 J=J+(L AND L>P)
200 LOCATE 1,18:PRINT "PONTUACAO - JOGAD
OR 1: ";:PEN 2:PRINT S;:PEN 1
210 PRINT "JOGADOR 2: ";:PEN 3:PRINT J:PE
N 1
220 LOCATE 5,25:PAPER 3:INPUT "CARREGUE
EM <ENTER> PARA CONTINUAR ";R$:PAPER 0
230 GOTO 50
```

Maquinapalavra

REFERÊNCIAS

Dificuldade de programação: fácil.

Categoria de jogo: jogo de azar (com palavras).

Dificuldade do jogo: nenhuma.

O JOGO:

Este curtíssimo programa permite que o AMSTRAD demonstre as suas capacidades de manipular cadeias de caracteres: parte-se simplesmente de um prefixo ou de um sufixo de origem grega, que se vai juntar ao acaso; de quando em quando aparecerão, sem dúvida, palavras absurdas ou mesmo ligeiramente equívocas.

Em cada tentativa é confeccionada uma nova palavra, mas é possível obter várias vezes a mesma palavra.

O PROGRAMA:

Duas variáveis, A\$ e B\$, são dimensionadas com o comprimento 14 nas linhas 60 e 70: uma conterà os prefixos, outra os sufixos (se se desejar aumentar as possibilidades bastará modificar esta limitação de 14).

O ciclo I (80-100) lê as linhas DATA (180-200) que se referem aos prefixos seguidos dos sufixos.

A linha 110 efectua a escolha de um prefixo ao acaso.

A linha 120 efectua a escolha de um sufixo ao acaso.

A palavra obtida na linha 130 é impressa no écran pela linha 140, e depois o processo recomeça (linha 150).

O ciclo WHILE/WEND em 160 provoca uma espera que utiliza o relógio interno (TIME) do AMSTRAD.

PARA JOGAR:

Fazer correr o programa usando RUN. Será imediatamente impressa no écran uma palavra, depois uma nova palavra será apresentada a seguir à precedente.

Teclar ESC duas vezes para sair do programa.

EXTENSÕES POSSÍVEIS:

Poder-se-á muito facilmente aumentar a lista das palavras disponíveis.

Poder-se-á também utilizar um procedimento análogo para confeccionar frases no computador, fazê-lo inventar poemas, etc. Poder-

-se-á também transformar o programa inicial num jogo de sociedade que consistiria, por exemplo, em fazer apostas sobre a próxima palavra que vai aparecer. E confiar, eventualmente, ao AMSTRAD a gestão das apostas...

```
10 REM MAQUINAPALAVRA
20 RANDOMIZE(TIME)
30 DEFINT I,J,P
40 MODE 0
50 CLS
60 DIM A$(14)
70 DIM B$(14)
80 FOR I=0 TO 14
90 READ A$(I),B$(I)
100 NEXT I
110 I=RND*14
120 J=RND*14
130 C#=A$(I)+B$(J)
140 P=1+RND*12:PEN P:PRINT C#
150 X=TIME
160 WHILE TIME-X<100:WEND
170 GOTO 110
180 DATA ANDRO,CRATA,COSMO,FILO,MICRO,FO
BO,HELIO,TROPO,SEXO,GINO,ESTEREO,MANO
190 DATA PALEO,NAUTA,MACRO,FAGO,HIDRO,FU
GO,ESTENO,PATA,PATO,CARDIO,OLEO,DERMO
200 DATA MITO,ESCOFICO,HOMEO,VORO,OCTO,F
ONO
```

Espião

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: jogo de reflexão.
Dificuldade do jogo: fácil.

O JOGO:

O programa pode servir simultaneamente de codificador e de decodificador de mensagens secretas, seguindo um método facilimo usado neste tipo de aparelhos.

Um pequeno gerador aleatório situado no fim do programa é utilizado para modificar de modo arbitrário cada letra da mensagem. Deverá ter-se o cuidado de utilizar sempre o mesmo número (núcleo) para começar a codificação ou a decodificação de duas mensagens idênticas.

O PROGRAMA:

O núcleo inicial (que será um inteiro qualquer) é feito entrar na linha 30; a mensagem faz-se entrar na linha 50.

A linha 60 permite escolher entre as opções de codificação e de decodificação.

O ciclo I (80-150) é utilizado quer para efectuar a codificação quer para efectuar a decodificação.

A linha 100 chama o gerador aleatório.

A linha 110 realiza a transposição das letras.

As linhas 120 e 130 verificam que se permanece dentro dos limites permitidos.

Os resultados serão impressos pela linha 160.

As linhas 1000 e 1030 contêm o gerador.

PARA JOGAR:

Fazer correr o programa com RUN.

Em seguida fazer entrar um inteiro N, que servirá de núcleo ao gerador (será necessário tomar nota dele, para a decodificação).

Depois teclar a mensagem, seguida de ENTER.

Escolher a opção C para CODIFICAR ou a opção D para DESCODIFICAR, e teclar ENTER.

A mensagem codificada ou decodificada, consoante o caso, será então impressa no écran.

EXTENSÕES POSSÍVEIS:

Poder-se-á alargar a codificação a outros sinais além das letras do alfabeto, utilizar eventualmente vários geradores, introduzir na mensagem um código do ou dos números que tenham servido de geradores, etc.

```
10 REM ESPIAO
20 CLS
30 INPUT "ENTRE O SEU CODIGO: ";N
40 A=1/N
50 PRINT:INPUT "A SUA MENSAGEM: ";A$
60 PRINT:PRINT "DESEJA ";PEN 2:PRINT "(C
)ODIFICAR";:PEN 1:PRINT " OU ";:PEN 3:PR
INT "(D)ESCODIFICAR?";:PEN 1:INPUT ": ";
R$
70 B$=""
80 FOR I=1 TO LEN(A$)
90 T=ASC(MID$(A$,I,1))
100 GOSUB 1000
110 T=T-A*((R$="C")-(R$="D"))
120 IF T>90 THEN T=T-26
130 IF T<65 THEN T=T+26
140 B#=B#+CHR$(T)
150 NEXT I
160 PRINT:PRINT "RESULTADO: ";:PEN (2-(R
$="D")):PRINT B$:PEN 1:END
1000 A=(A+PI)^8
1010 A=A-INT(A)
1020 A=INT(26*A+1)
1030 RETURN
```

Totoloto

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: azar (vários jogadores).

Dificuldade do jogo: fácil.

O JOGO:

Trata-se de realizar uma tiragem completa do totoloto, isto é, de obter seis números, entre 1 e 49, mais um número suplementar, todos diferentes, que poderão ser utilizados, por exemplo, para preencher o seu próximo boletim do Totoloto.

O PROGRAMA:

Utilizar-se-á um quadro N(7), dimensionado na linha 40, para registar os resultados.

A tiragem dos números é efectuada num ciclo I (60-120), sendo utilizado um segundo ciclo, J (90-110), para controlar que não haja duas tiragens idênticas.

A tiragem (ou a retirada) é efectuada na linha 70.

O ciclo I (150-180) imprime o resultado e a linha 190 imprime o número suplementar.

As linhas 210 a 230 ocupam-se do caso de se querer fazer outra partida.

PARA JOGAR:

Basta teclar RUN para fazer correr o programa e obter a impressão no écran de um primeiro conjunto de números.

A pergunta OUTRA TIRAGEM (S/N)? é impressa na base do écran; responder-se-á S no caso de se desejar continuar.

EXTENSÕES POSSÍVEIS:

Pode-se imaginar um programa que possa tomar em consideração quer dos seus próprios resultados quer dos resultados fornecidos por um jogador, para melhorar a tiragem, tendo em conta a probabilidade de sair um dado número.

Pode-se também encarar a possibilidade de um programa que realize cartões para o jogo do loto tradicional, tal como é ainda correntemente praticado um pouco por toda a parte.

```

10 REM TOTOLOTO
20 RANDOMIZE(TIME)
30 MODE 0
40 DIM N(7)
50 CLS
60 FOR I=1 TO 7
70 N(I)=1+INT(RND*45)
80 IF I=1 THEN 120
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 70
110 NEXT J
120 NEXT I
130 LOCATE 4,6:PEN 1:PRINT "TIRAGEM DO T
OTOLOTO"
140 PRINT STRING$(3," ");STRING$(14,"-")
150 FOR I=1 TO 6
160 LOCATE 3*I-2,12:PEN 3+I
170 PRINT N(I)
180 NEXT I
190 LOCATE 1,18:PEN 1:PRINT "COMPLEMENTA
R: ";PEN 15:PRINT N(7):PEN 1
200 SOUND 1,478:SOUND 1,426:SOUND 1,379
210 LOCATE 1,24:PAPER 1:PEN 0:PRINT "OUT
RA TIRAGEM (S/N)?":PAPER 0:PEN 1
220 A#=INKEY$:IF A#="" THEN 220
230 IF UPPER$(A#)="S" THEN 50

```

Tiragem de um dado viciado

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: reflexão.

Dificuldade do jogo: média.

O JOGO:

Retoma a ideia de um programa anterior («Dado viciado») numa aplicação estatística: fabrica-se um dado voluntariamente viciado, que se vai lançar um certo número de vezes, e desenha-se um quadro dos resultados.

O PROGRAMA:

As probabilidades desejadas para cada tiragem são introduzidas por um ciclo I (70-140), com todas as verificações de validade necessárias. O número de tiragens pedidas é introduzido na linha 160.

O ciclo J (170-250) contabiliza as tiragens.

O ciclo I (190-210) assegura a conformidade de cada tiragem com as probabilidades introduzidas no início do jogo.

Os resultados são contabilizados na linha 220 e impressos no écran graças às linhas 230 e 240.

PARA JOGAR:

Começar a execução teclando RUN e depois, à pergunta PROBABILIDADE DO DADO 1?, introduzir um número entre 0 e 1.

Fazer o mesmo para os dados 2, 3, etc., sem esquecer que o total das probabilidades introduzidas deverá ser cerca de 1.

Em seguida, após a impressão da mensagem NUMERO DE TIRAGENS?, introduzir-se-á um número inteiro suficientemente grande, seguido de ENTER. A impressão dos resultados levará tanto mais tempo a aparecer quanto maior for o número inteiro introduzido.

Os resultados são impressos sob os números das tiragens; assim, 32.3 aparecendo sob o 1 significa que a tiragem de 1, à qual fora atribuída uma probabilidade de 0.3, foi realizada 32 vezes.

EXTENSÕES POSSÍVEIS:

Pode-se encarar a possibilidade de melhorar a representação dos resultados, utilizando por exemplo um histograma ou qualquer outro método mais eloquente graficamente.

```

10 REM TIRAGEM DE UM DADO VICIADO
20 RANDOMIZE(TIME)
30 INK 0,2:INK 1,17
40 DIM E(6)
50 CLS
60 P=0
70 FOR I=1 TO 6
80 LOCATE 1,3:PRINT STRING$(40,32)
90 LOCATE 1,3:PRINT "PROBABILIDADE DO DA
DO ";I:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>=1 OR P>1.1 THEN 3
00
120 LOCATE 6*I-3,6:PRINT "DADO ";I
130 LOCATE 6*I-3,8:PRINT E(I)
140 NEXT I
150 IF P<0.9 THEN RUN
160 PRINT:PRINT:INPUT "NUMERO DE TIRAGEN
S: ";P
170 FOR J=1 TO P
180 N=INT(RND*6+1)
190 FOR I=1 TO 6
200 IF N=I AND RND>E(I)-INT(E(I)) THEN 1
80
210 NEXT I
220 E(N)=E(N)+1
230 LOCATE 6*N-1,15:PRINT N
240 LOCATE 6*N-3,17:PRINT E(N)
250 NEXT J
260 END
300 PAPER 3:LOCATE 11,25:PRINT CHR$(7);"
IMPOSSIVEL: 0<=P<1":PAPER 0
310 P=P-E(I)
320 FOR J=1 TO 1000:NEXT J
330 LOCATE 10,25:PRINT STRING$(22,32);
340 GOTO 80

```

O jogo de Marienbad

REFERÊNCIAS:

- Dificuldade de programação: média.
- Categoria de jogo: lógico (dois jogadores).
- Dificuldade do jogo: média.

O JOGO:

Um jogo tornado célebre graças a um filme que o popularizou, aqui no entanto apresentado numa versão mais conforme com as suas origens, a saber, o jogo de Nim.

Efectivamente é possível, no início da partida, escolher o número de fósforos que se quer pôr em cada monte.

Cada jogador tem a sua vez de jogar e perde aquele que retirar o último fósforo.

O PROGRAMA:

A linha 60 dimensiona uma variável A(I) que conterà o número de fósforos de cada monte.

O ciclo I (70-100) permite fazer entrar esse número.

O ciclo I (130-160) imprime os montes.

A linha 170 assegura a impressão do número do jogador.

A linha 190, assim como a linha 200, controlam se a jogada é permitida.

A linha 260 controla o fim da partida.

PARA JOGAR:

Faz-se correr o programa com RUN, depois faz-se entrar o número de montes (entre 1 e 7), seguido de ENTER.

A seguir faz-se entrar o número de fósforos em cada monte (fazer em cada entrada N seguido de ENTER). Obter-se-á então a impressão no écran dos diferentes montes, com o número do jogador respectivo (1 ou 2), assim como a mensagem DE QUE MONTE?; responde-se tecendo o número do monte de onde se deseja retirar fósforos, seguido de ENTER. Obtém-se em seguida a pergunta QUANTIDADE A RETIRAR?, e responde-se indicando o número de fósforos que se deseja retirar do monte referido.

Depois será a vez do segundo jogador, e jogar-se-á alternadamente até que já não haja mais fósforos disponíveis.

No fim da partida obter-se-á a impressão no écran do número do jogador que ganhou.

EXTENSÕES POSSÍVEIS:

Pode-se designar cada jogador pelo seu nome, fazer um grafismo mais elaborado, escrever um programa que permita que o AMSTRAD seja um dos jogadores, etc.

```
10 REM O JOGO DE MARIENBAD
20 CLS
30 PRINT "NUMERO DE MONTES ( <=7 )?";:PA
PER 3:INPUT "",N:PAPER 0
40 IF N>=8 OR N<=0 THEN RUN
50 S=0
60 DIM A(N)
70 FOR I=1 TO N
80 PRINT:PRINT "MONTE ";I;:INPUT "-NUMER
O DE FOSFOROS: ";A(I)
90 S=S+A(I)
100 NEXT I
110 H=1
120 CLS
130 FOR I=1 TO N
140 LOCATE 5*I+3,5:PAPER 1:PEN 0:PRINT U
SING "####";A(I)
150 PAPER 0:PEN 1:PRINT:PRINT "MONTE":LO
CATE 5*I+3,7:PRINT I
160 NEXT I
170 LOCATE 1,10:PEN 1+H:PRINT "JOGADOR "
; H
180 PRINT:INPUT "DE QUE MONTE? ";T
190 IF T>N THEN PRINT CHR$(7);:GOTO 120
200 IF A(T)=0 THEN PRINT CHR$(7);:GOTO 1
20
210 LOCATE 1,14:PRINT CHR$(20);:INPUT "Q
UANTIDADE A RETIRAR? ";O
220 IF O>A(T) OR O<=0 THEN PRINT CHR$(7)
::GOTO 200
230 A(T)=A(T)-O
240 S=S-O
250 H=1+H MOD 2
260 IF S=0 THEN PRINT:PRINT "O JOGADOR "
;H;" GANHOU.":SOUND 1,478:SOUND 1,319:SO
UND 1,379:END
270 GOTO 120
```

Compõe

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: composição musical.

Dificuldade do jogo: nenhuma, *a priori*.

O JOGO:

Trata-se, com efeito, de um programa capaz de compor música, de modo muito elementar, e isso de um modo completamente aleatório.

O computador vai, num primeiro tempo, desenhar uma pauta, depois imprime uma sucessão de notas (que vão do DÓ grave ao SI agudo) e dá o seu nome. Em seguida será impressa uma nova série de notas, e assim por diante.

O PROGRAMA:

O ciclo I(50-70) desenha a pauta. A variável N\$, na linha 80, contém o nome das notas e V\$ contém as notas sob uma forma codificada.

O segundo ciclo I (90-150) permite escolher uma nota de forma aleatória, na linha 100; o seu nome é calculado na linha 110 e a nota é desenhada pela linha 120.

A linha 130 toca a nota, enquanto o ciclo WHILE/WEND proíbe continuar até ao fim da sua execução.

A linha 160 permite recomeçar teclando ENTER.

PARA JOGAR:

Fazer correr o programa teclando RUN.

A pauta será então rapidamente apresentada no écran, depois são impressas as notas, com o seu nome escrito na base do écran.

Um toque em qualquer tecla provocará a impressão de uma nova série de notas, e uma pressão dupla de ESC provoca o fim da partida.

EXTENSÕES POSSÍVEIS:

Poder-se-á afinar as regras de composição, imprimir mais notas, introduzir sustenidos e bemóis, melhorar o grafismo.

```

10 REM COMPOE
20 RANDOMIZE(TIME)
30 PRINT CHR$(22);CHR$(1)
40 CLS
50 FOR I=1 TO 5
60 LOCATE 1,4+2*I:PEN 3:PRINT STRING$(40
,154)
70 NEXT I
80 N$="SI LA SOLFA MI RE DO ":V$="127142
159179190213239253284319358379426478"
90 FOR I=1 TO 10
100 X=INT(RND*14+1)
110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
120 PEN 2:LOCATE 4*I-2,X:PRINT CHR$(211)
;CHR$(10);CHR$(8);CHR$(211);CHR$(10);CHR
$(8);CHR$(8);"0"
130 SOUND 1,VAL(MID$(V$,3*X-2,3)):WHILE
SQ(1)>127:WEND
140 PEN 1:LOCATE 4*I-3,20:PRINT D$
150 NEXT I
160 PRINT CHR$(22)+CHR$(0);:PAPER 1:PEN
0:LOCATE 7,24:INPUT "PREMIR <ENTER> PARA
CONTINUAR",R$:PAPER 0
170 GOTO 40

```

Dia da semana

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: utilitário.

Dificuldade do jogo: fácil.

O JOGO:

O programa permite achar, sem qualquer cálculo, o dia da semana correspondente a uma certa data.

Far-se-á entrar a data sob a forma de três números, por exemplo 11, 3, 1956 para 11 de Março de 1956, e o resultado será impresso no écran.

O PROGRAMA:

As linhas 20 a 60 controlam as entradas. A linha 40 ocupa-se do fim da partida. A linha 90 permite a correcção para as datas anteriores a 1582 (data em que se começou a utilizar o calendário actual).

As linhas 100 a 110 permitirão a impressão no écran do nome do mês (só quatro letras) na resposta.

A fórmula utilizada aparece nas linhas 130 e 140.

As linhas 150 a 180 permitirão a apresentação do dia por extenso, na resposta. A linha 180 imprime os resultados centrados no écran.

PARA JOGAR:

Fazer correr o programa com RUN, depois entrar o dia seguido de ENTER, o mês (entre 1 e 12), seguido de ENTER, e o ano (sob a forma 1845, por exemplo) seguido de ENTER.

O resultado é rapidamente impresso no écran.

Carrega-se numa tecla qualquer e o programa volta a fazer a pergunta DIA?.

Para continuar basta teclar um dia, um mês, um ano.

Para terminar é preciso responder 0 à pergunta DIA?.

EXTENSÕES POSSÍVEIS:

Poder-se-á fazer entrar os meses por extenso e fazer imprimir por extenso o nome do dia.

Pode-se pensar em fazer imprimir as datas e festas particulares (Natal, Páscoa, Domingo de Ramos, Pentecostes...).

Pode-se escrever um programa que imprimirá os dias feriados de um dado ano, etc.

```

10 REM DIA DA SEMANA
20 CLS:INK 3,6,5
30 PRINT:INPUT "DIA. . . . . ?";
J
40 IF J=0 THEN END
50 PRINT:INPUT "MES. . . . . ?";
M
60 PRINT:INPUT "ANO. . . . . ?";
A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THEN S=2
100 M$="JANEFEVEMARCBRIMAIJUNHJULHAGOS
SETEOUTUNOVEZE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THEN GOSUB 220
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S MOD 7
150 J$="DOMINGO SEGUNDA TERCA   QUARTA
QUINTA  SEXTA   SABADO  "
160 J$=MID$(J$,8*S+1,8):WHILE RIGH$(J$,
1)=" ":J$=LEFT$(J$,LEN(J$)-1):WEND
170 P$="D "+STR$(J)+" "+M$+STR$(B)+" E'
UM "
180 PEN 2:LOCATE (40-LEN(P$)-LEN(J$))/2,
20:PRINT P$;;PEN 3:PRINT J$:PAPER 1:PEN
0
190 SOUND 1,478:SOUND 1,319:SOUND 1,379
200 LOCATE 6,24:INPUT "PREMIR <ENTER> PA
RA CONTINUAR",R$:PAPER 0:PEN 1
210 RUN
220 M=M+12
230 A=A-1
240 RETURN

```

Habilidade com cartas

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de prestidigitação.

Dificuldade do jogo: fácil.

O JOGO:

É uma habilidade muito fácil, mas, aqui, as cartas são manipuladas sob os vossos olhos pelo AMSTRAD, que desempenha o papel de prestidigitador; são mostradas no écran 21 cartas, tomadas ao acaso num baralho de 52, e o jogador deverá reter mentalmente uma das cartas apresentadas. Depois deverá indicar ao AMSTRAD em que coluna (primeira, segunda ou terceira a partir da esquerda) se encontra a carta escolhida; as cartas são então redistribuídas e o jogador indica de novo a coluna correspondente à sua carta; isto é feito mais uma vez e, à quarta apresentação, o AMSTRAD está em condições de adivinhar qual é a carta escolhida.

O PROGRAMA:

O ciclo I (60-100) permite escolher, ao acaso, 21 cartas das 52 do baralho, e a linha 80 permite assegurar que não seja feita duas vezes a mesma tiragem. A linha 120 contém os nomes das cartas (2, 3, ..., VA, DA, RE, AS) e a sua cor nas duas variáveis alfanuméricas V\$ e C\$.

O ciclo I (130-210) realiza a apresentação do jogo no écran.

A linha 140 descodifica a cor da carta.

A linha 150 escolhe a posição de impressão.

A linha 160 descodifica a altura.

A impressão propriamente dita é efectuada da linha 170 à linha 200.

A linha 230 permite ao jogador fazer entrar o número da coluna onde se encontra a carta que escolheu. O ciclo J (270-290) baralha as cartas, e os ciclos encadeados J (320-360) e I (330-350) fabricam a nova variável A\$, a ser utilizada na próxima apresentação no écran.

PARA JOGAR:

Faz-se correr o programa com RUN.

Obtém-se uma primeira apresentação de 21 cartas repartidas por 3 colunas.

Indica-se então o número (1, 2 ou 3) da coluna onde se encontra a carta escolhida, seguido de ENTER.

Obtém-se uma segunda apresentação e faz-se o mesmo; depois uma terceira apresentação e volta a fazer-se o mesmo.

À quarta apresentação, a carta escolhida é mostrada automaticamente.

Sai-se do programa teclando duas vezes ESC em vez do número da coluna.

EXTENSÕES POSSÍVEIS:

Pode-se pensar numa melhoria da apresentação escrevendo os nomes das cartas por extenso, ou mesmo desenhando verdadeiras cartas, a cores.

Poder-se-á também tentar programar outras habilidades, utilizando o mesmo princípio básico mas com mais cartas, etc.

```
10 REM HABILIDADE COM CARTAS
20 RANDOMIZE(TIME)
30 BORDER 9:INK 0,9:INK 1,26:INK 2,0:INK
3,6
40 CLS
50 A$=""
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND*52+38))
80 IF INSTR(A$,B$)<>0 THEN 70
90 A$=A$+B$
100 NEXT I
110 H=3:P=1
120 V$="2 3 4 5 6 7 8 9 10VADAREAS":C$=C
HR$(226)+CHR$(227)+CHR$(228)+CHR$(229)
130 FOR I=1 TO LEN(A$)*P-11*(P=0)
140 S=1+INT((ASC(MID$(A$,I,1))-38)/13)
150 J=INT((I-1)/7)
160 T=(ASC(MID$(A$,I,1))-38) MOD 13
170 IF P=0 THEN LOCATE 1,12:PRINT "A SUA
CARTA E' : "
180 LOCATE 1+17*J*P-17*(P=0),1+(I-7*J)*P
-11*(P=0)
190 PRINT MID$(V$,2*T+1,2);" ";
200 PEN (2-(S=2 OR S=3)):PRINT MID$(C$,S
,1):PEN 1
210 NEXT I
220 IF P=0 THEN END
230 LOCATE 1,16:INPUT "QUAL COLUNA? ";N
240 CLS
250 H=H-1
260 R$="20120":B$=""
270 FOR J=0 TO 2
280 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,1))+
1,7)
290 NEXT J
300 IF H=0 THEN A$=B$:P=0:GOTO 120
310 A$=""
320 FOR J=0 TO 2
330 FOR I=1 TO 7
340 A$=A$+MID$(B$,3*I-2+J,1)
350 NEXT I
360 NEXT J
370 GOTO 120
```

O indesejável

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: azar e reflexão (um jogador).

Dificuldade do jogo: média.

O JOGO:

Você é proprietário de um prédio com 16 divisões vazias e não consegue apanhar o seu locatário indesejável. Visita as divisões ao acaso mas, se visitar a divisão vizinha, o seu locatário, prevenido pelo ruído, muda-se. Pode mesmo reinstalar-se numa divisão já visitada e marcada como estando vazia. Para ganhar, terá de o desalojar em menos de nove jogadas.

O PROGRAMA:

Na linha 50 um quadro B é dimensionado a 16, para receber os números das casas já visitadas.

O ciclo I (70-100) desenha o campo de jogo usando tinta transparente.

A posição inicial do locatário é tirada ao acaso na linha 110.

O número da casa a visitar é dado na linha 120 e é verificado na linha 130.

As linhas 140 a 210 controlam as diversas respostas possíveis.

O ciclo I (180-200) desenha as casas visitadas.

A linha 220 desenha o locatário, apagado na linha 250.

As linhas 260 e 270 asseguram a sua deslocação.

PARA JOGAR:

Faz-se correr o programa com RUN.

À pergunta CASA VISITADA?, responde-se teclando uma letra, de A a P. Se o locatário é encontrado, é mostrado no écran e a partida termina; se não, podem ocorrer dois casos: estar-se bastante longe, e o locatário não se mexe, ou estar na casa vizinha — caso em que aparece durante um curto instante e volta a esconder-se, aleatoriamente, numa casa vizinha. As casas já visitadas ficam de cor azul e não pode voltar a elas. Se, ao fim de nove jogadas, não encontrou o locatário, a posição dele é mostrada e você perde.

EXTENSÕES POSSÍVEIS:

Poder-se-á, claro, jogar com mais de 16 casas, utilizar caracteres gráficos para representar o locatário e o proprietário, utilizar um algoritmo mais subtil para a deslocação do locatário, eventualmente interditar-lhe o regresso a casas já visitadas, etc.

```
10 REM O INDESEJAVEL
20 RANDOMIZE(TIME)
30 PRINT CHR$(22)=CHR$(1)
40 B#=STRING$(3,127)
50 DIM B(16)
60 CLS
70 FOR I=1 TO 16
80 LOCATE 4+4*(I-1-4*INT((I-1)/4)),2+4*
INT((I-1)/4):PEN 3:PRINT B#
90 LOCATE 4+4*(I-1-4*INT((I-1)/4)),3+4*
INT((I-1)/4):PRINT B#:CHR$(8);CHR$(8);:PE
N 2:PRINT CHR$(64+I)
100 NEXT I
110 X=INT(RND*16)+1:S=1:K=0:PRINT CHR$(2
2)+CHR$(0)
120 PEN 1: LOCATE 1,20:PRINT CHR$(20);:I
NPUT "CASA VISITADA? ";C#:C=ASC(C#+ " ")-
64
130 IF C<1 OR C>16 THEN 120
140 IF B(C)=1 THEN LOCATE 14,22:PAPER 1;
PEN 0:PRINT "JA' VISITADA":PRINT CHR$(7)
:PAPER 0:GOTO 120
150 IF C=X THEN LOCATE 3,22:PEN 2:PRINT
"CONSEGUIU EM";S;"JOGADAS. BRAVO!":SOUND
1,478:SOUND 1,379:SOUND 1,319:K=1:GOTO
220
160 IF X-5=C OR X-4=C OR X-3=C OR X-1=C
OR X+1=C OR X+3=C OR X+4=C OR X+5=C THEN
GOTO 220
170 LOCATE 12,22:PRINT "NAO ESTA AQUI":S
OUND 1,478:SOUND 1,426:SOUND 1,379
180 S=S+1:B(C)=1:FOR I=1 TO 16
190 LOCATE 4+4*(I-1-4*INT((I-1)/4)),2+4*
INT((I-1)/4):PEN 3-B(I):PRINT B#
200 NEXT I
210 IF S=9 THEN LOCATE 5,22:PAPER 3:PEN
0:PRINT "PERDEU, ELE ESTAVA EM ";CHR$(X+
64):K=1:PAPER 0:PEN 1:ELSE GOTO 120
220 LOCATE 5+4*(X-1-4*INT((X-1)/4)),1+4*
INT((X-1)/4):PRINT "% "
230 HE=TIME:WHILE TIME-HE<20:WEND
240 IF K=1 THEN SOUND 1,319:SOUND 1,379:
SOUND 1,478:LOCATE 1,24:END
250 LOCATE 5+4*(X-1-4*INT((X-1)/4)),1+4*
INT((X-1)/4):PRINT " "
260 Z=INT(10*RND+1)-5:IF ABS(Z)=2 THEN 2
60
270 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:COTO
260
280 GOTO 180
```

O jogo da vida

REFERÊNCIAS:

- Dificuldade de programação: difícil.
- Categoria de jogo: solitário (duas versões).
- Dificuldade do jogo: fácil.

O JOGO:

A versão aqui apresentada respeita as regras criadas originalmente pelo matemático J. H. Conway.

No princípio do jogo é definida uma certa população, quer pelo AMSTRAD (versão automática) quer pelo próprio jogador (versão manual). Esta população vai evoluir diante dos olhos do jogador, respeitando certas regras, muito elementares, de nascimento, vida e morte. A evolução prosseguirá até que se deseje parar o programa.

O PROGRAMA:

São utilizados dois quadros (B e C), na linha 50, para registar o estado da população. Os ciclos F (230-290) e G (240-280) são utilizados no início do jogo para definir a população inicial (linha 250). O cálculo da nova população é realizado em seguida com o auxílio de um ciclo quádruplo: 80-210, 90-200, 110-160, 120-150. A linha 130 realiza a gestão do novo estado, de acordo com as regras.

A apresentação no écran deste novo estado é obtida realizando o duplo ciclo de impressão F (230-290) e G (240-280).

A linha 300 permite recomeçar.

PARA JOGAR:

Na versão automática basta fazer correr o programa com RUN e observar o écran. Para parar basta teclar ESC duas vezes.

Na versão manual, depois de fazer RUN para correr o programa, far-se-á entrar a configuração de partida, teclando 0 se não se quer pôr nada na linha e na coluna indicadas e 1 se se deseja pôr aí uma célula. Depois a evolução desenrolar-se-á de modo automático.

EXTENSÕES POSSÍVEIS:

Será fácil, se se dispuser de memória suficiente, aumentar as dimensões dos quadros B e C, redefinindo a variável A. Poder-se-á também introduzir a cor e melhorar a precisão do desenho substituindo os PRINT por uma apresentação de alta resolução.

```

10 REM JOGO DA VIDA (MANUAL)
30 INK 0,14:INK 1,18
40 INPUT "DIMENSAO: ";A:IF A<1 OR A>12 T
HEN RUN
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 260
255 LOCATE 1,25:PEN 3:PRINT "LINHA ";F;"
CDLUNA ";G;" PREMIR 1 OU 0 ";
257 LOCATE 33,25:INPUT C(F,G):C(F,G)--(C
(F,G)=1):PEN 1
260 LOCATE 2*G,2*F:IF C(F,G)=1 THEN PRIN
T "*"
270 B(F,G)=C(F,G)
280 NEXT G
290 NEXT F
300 GOTO 80

```

```

10 REM JOGO DA VIDA (AUTOMATICO)
20 RANDOMIZE(TIME)
30 INK 0,14:INK 1,18
40 INPUT "DIMENSÃO: ";A:IF A<1 OR A>12 T
HEN RUN
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I=0 THEN C(F,G)=- (RND<0.5)
260 LOCATE 2*G,2*F:IF C(F,G)=1 THEN PRIN
T "*"
270 B(F,G)=C(F,G)
280 NEXT G
290 NEXT F
300 GOTO 80

```

Nível 3

Nome do programa

Temas e particularidades

Losangos	Para o ajudar a adormecer.
Maior, menor	Um interessante jogo de dedução,
Ordem	Um pouco de astúcia para repor a ordem o mais depressa possível.
O enforcado	Um jogo de letras clássico (dois jogadores).
A bisca	Simulação de uma partida, com um jogo de 32 cartas.
Um cálculo quebra-cabeças	Um bom treino mental (dois jogadores).
Enforcado invertido	Desta vez é o computador que tenta adivinhar a palavra.
«Nicómaco»	Para se aperfeiçoar no cálculo mental (um ou dois jogadores).
«Mastermind»	Com números, mas muito clássico. Um só jogador.
Idem	Um jogo de cálculo mental com vários níveis, para vários jogadores.
Rainhas no tabuleiro de xadrez	Um velho problema, para o ajudar a reflectir.
Palavras	Uma ideia muito simples, mas não muito fácil de resolver!
Permutas	Um jogo de reflexão, nem sempre evidente.
Bridge	Aperfeiçoe os seus próprios problemas de bridge.
Hanói	Um conselho: comece com um número pequeno!
Corrida de cavalos	Muito visual; um jogo de azar, para vários jogadores.
Engatem os vagões	Para os mais jovens: muito espectacular.
O lobo e os cordeiros	Um jogo de reflexão para dois jogadores.

Manipulação das cadeias e gráficos de alta resolução

TRATAMENTO DAS CADEIAS DE CARACTERES:
MID\$, LEFT\$, RIGHT\$

Tivemos ocasião de encontrar, a partir do nível 2, a função essencial utilizada no Basic AMSTRAD para manipular as cadeias: trata-se de **MID\$(A\$,I,J)**, que permite extrair da cadeia A\$ uma subcadeia de comprimento J a começar no carácter de ordem I.

Assim, se:

A\$ = "SENHOR"

ter-se-á:

MID\$(A\$,2,3) = "ENH"

Para facilitar as manipulações, é possível utilizar as duas funções LEFT\$ e RIGHT\$.

LEFT\$(A,I) permite extrair da palavra A\$ os I primeiros caracteres a partir da esquerda.

Assim, usando o exemplo precedente, LEFT\$(A\$,2) = "SE" [poder-se-ia, aliás, utilizar MID\$(A\$,1,2) para obter o mesmo resultado].

RIGHT\$(A\$,J) permite extrair os J últimos caracteres a partir da direita da palavra A\$.

Assim, usando o exemplo precedente, RIGHT\$(A\$,3) = "HOR".

É mais difícil passar sem a função RIGHT\$, mas seria possível fazer MID\$(A\$, LEN(A\$)—3,3) para obter o mesmo resultado.

O Basic AMSTRAD permite ainda a substituição directa de uma subcadeia por uma atribuição do tipo MID\$(—, —) = "—".

Assim, poder-se-á escrever MID\$(A\$,I,1) = "—", o que permite simplificar consideravelmente certas escritas.

O GRAFISMO COM A INSTRUÇÃO PLOT

O AMSTRAD possui três larguras de écran (20, 40 e 80 colunas), às quais correspondem três definições gráficas.

Estes três modos têm a particularidade de possuir as mesmas posições no écran.

Assim, no modo 0, 1 ou 2, a instrução PLOT X,Y,C imprime um ponto na mesma posição do écran X,Y. Só varia a dimensão do ponto.

Um parâmetro opcional C permite variar a cor de impressão.

AS INSTRUÇÕES DE POSICIONAMENTO: MOVE E MOVER

A origem dos eixos de posicionamento das coordenadas de um ponto gráfico é, por defeito, o canto inferior esquerdo do écran. A instrução ORIGIN permite deslocá-la.

No posicionamento assim definido, MOVE X,Y posiciona o cursor gráfico no sítio onde as impressões podem ser feitas.

As deslocações podem ser efectuadas relativamente à posição actual do cursor. Assim, MOVER 8,4 desloca o cursor oito unidades para a direita e quatro para cima.

O DESENHO DE LINHAS: DRAW E DRAWR

A instrução DRAW X,Y desenha uma linha desde a posição do cursor gráfico até ao ponto de coordenadas absolutas X e Y.

A instrução DRAWR X,Y desenha o traço até ao ponto deslocado X unidades horizontais e Y unidades verticais.

Losangos

REFERÊNCIAS:

Dificuldade de programação: fácil.
Categoria de jogo: interlúdio.
Dificuldade do jogo: nenhuma.

O JOGO:

O programa vai construir progressivamente no écran um losango, depois vai apagá-lo progressivamente, reconstruí-lo de novo — e assim por diante.

Pode pensar em utilizá-lo para o adormecer, se contar carneiros já não lhe basta.

É-lhe proposta uma outra versão, muito diferente, que lhe permitirá compreender o funcionamento da poderosa instrução DRAW.

O PROGRAMA:

A linha 30 coloca no centro do écran a origem dos eixos.

Um primeiro ciclo K (50-130) define as dimensões dos traçados sucessivos.

O segundo ciclo, I (60-120), traça faixas de pontos, utilizando instruções PLOT (linhas 80 a 110).

Se a variável S (linha 70) estiver a -1 , então haverá apagamento aleatório (continua-se a usar PLOT, mas utilizando cores aleatórias para apagar).

O ciclo WHILE/WEND, inicializado em 140, permite escolher quatro cores diferentes.

O segundo programa utiliza DRAW para desenhar os losangos e depois para os apagar.

PARA JOGAR:

Começa-se a execução do programa com RUN. Se necessário, faz-se entrar um núcleo para o gerador (primeiro programa) e observa-se o écran.

EXTENSÕES POSSÍVEIS:

Poder-se-á introduzir o som, utilizar a instrução LINE (que permitirá um traçado muito mais rápido), imaginar diversos programas baseados em princípios análogos (caleidoscópio, tapeçaria...).

```

10 REM LOSANGOS1
20 RANDOMIZE(TIME)
30 MODE 0:INK 0,1:INK 1,0:INK 5,3:ORIGIN
 320,200
40 CLS:S=0
50 FOR K=0 TO 130 STEP 6
60 FOR I=0 TO K
70 IF S<>0 THEN ON INT(1*5*RND) GOTO 200
,210,220,230,240
80 PLOT -K+I,-I,1
90 PLOT -K+I,I,1
100 PLOT K-I,I,1
110 PLOT K-I,-I,1
120 NEXT I
130 NEXT K
140 FOR I=1 TO 4:C(I)=0:NEXT I
150 WHILE ((C(1)-C(2))*(C(1)-C(3))*(C(1)
-C(4))*(C(2)-C(3))*(C(2)-C(4))*(C(3)-C(4)
))=0
160 FOR I=1 TO 4:C(I)=2+INT(RND*14):NEXT
I
170 WEND
180 S=NOT S
190 GOTO 50
200 PLOT -K+I,-I,C(1)
210 PLOT -K+I,I,C(2)
220 PLOT K-I,I,C(3)
230 PLOT K-I,-I,C(4)
240 GOTO 120

10 REM LOSANGOS2
20 RANDOMIZE(TIME)
30 MODE 0:INK 0,1:INK 1,0:INK 5,3:ORIGIN
 320,200
40 CLS:S=0
50 FOR K=0 TO 130
70 IF S<>0 THEN ON INT(1*5*RND) GOTO 200
,210,220,230,240
80 MOVE 0,K
90 DRAW K,0,1
100 DRAW 0,-K,1
110 DRAW -K,0,1
120 DRAW 0,K,1
130 NEXT K
140 FOR I=1 TO 4:C(I)=0:NEXT I
150 WHILE ((C(1)-C(2))*(C(1)-C(3))*(C(1)
-C(4))*(C(2)-C(3))*(C(2)-C(4))*(C(3)-C(4)
))=0
160 FOR I=1 TO 4:C(I)=2+INT(RND*14):NEXT
I
170 WEND
180 S=NOT S
190 GOTO 50
200 MOVE 0,K:DRAW K,0,C(1)
210 MOVE K,0:DRAW 0,-K,C(2)
220 MOVE 0,-K:DRAW -K,0,C(3)
230 MOVE -K,0:DRAW 0,K,C(4)
240 GOTO 120

```

Maiores, menores

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: jogo de dedução (um só jogador).

Dificuldade do jogo: difícil.

O JOGO:

Inspira-se num princípio ligeiramente diferente do de «Mastermind»; trata-se igualmente de adivinhar um número (de um a sete algarismos), mas as indicações de que dispomos são agora, por um lado o número de algarismos que, no número tentado, é superior aos do número a adivinhar, e por outro lado o número de algarismos que lhe é inferior. Se, por exemplo, o número a adivinhar é 3054, e se tentamos 4251, a máquina responderá 2 maiores (o 4 e o 2) e 1 mais pequeno (o 1).

O PROGRAMA:

A linha 60 proíbe o uso de números de mais de oito algarismos.

O número a adivinhar é colocado na linha 80, numa variável alfanumérica X\$.

As comparações com o número tentado são feitas na linha 130 (que testa se acertamos), e no ciclo I (150-180) é feita a comparação «letra a letra».

A apresentação do resultado no écran é realizado graças à linha 190, com apagamento de uma eventual tentativa precedente.

O contador do número de tentativas está na linha 200.

PARA JOGAR:

O uso de RUN para fazer correr o programa faz aparecer a mensagem NUMERO DE ALGARISMOS?.

Faz-se entrar um número inferior a 8 e prime-se a tecla ENTER.

O número da tentativa é impresso, assim como a pergunta NUMERO TENTADO?; responde-se fazendo entrar um número, seguido de ENTER.

O resultado é então apresentado sob a forma > e <.

Se não acertámos, recomeça-se (atenção: entrar apenas números com tantos algarismos quantos o do número a adivinhar).

Se acertamos, o AMSTRAD imprimirá ADIVINHOU ... EM ... TENTATIVAS.

A linha 240 forma um ciclo de espera. Depois surge a mensagem OUTRO JOGO (S/N)?; carrega-se em S para iniciar novo jogo.

EXTENSÕES POSSÍVEIS:

Poder-se-á, como no «Mastermind», imaginar um jogo análogo com letras, cores ou mesmo utilizando outros símbolos.

Pode-se também pôr a hipótese de deixar jogar o AMSTRAD, que desempenhará então o papel de decodificador.

```
10 REM MAIOR, MENOR
20 RANDOMIZE(TIME)
30 INK 0,22:BORDER 22:INK 1,1:INK 2,9
40 CLS
50 INPUT "NUMERO DE ALGARISMOS? ";C
60 IF C<1 OR C>8 THEN PRINT CHR$(7):GOTO
 40
70 E=1:F=0
80 X$=STR$(INT(RND*9^C+10^(C-E)))
90 X$=RIGHT$(X$,LEN(X$)-1)
100 LOCATE 1,3:PRINT "TENTATIVA No. NUME
R0 TENTADO > <"
110 LOCATE 1,4:PRINT STRING$(40,154)
120 LOCATE 1,F+5:PRINT TAB(4);CHR$(18);E
;TAB(18-(C/2));INPUT "",P$
130 P$=LEFT$(P$,C):IF P$=X$ THEN 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A=
A+1
170 IF MID$(P$,I,1)<MID$(X$,I,1) THEN A=
B+1
180 NEXT I
190 LOCATE 28,F+5:PRINT A:LOCATE 36,F+5:
PRINT B
200 F=E MOD 17:E=E+1
210 GOTO 120
220 LOCATE 1,23:PRINT "ADIVINHOU ";:PEN
3:PRINT X$;:PEN 1:PRINT " EM ";:PEN 2:PR
INT E;:PEN 1:PRINT "TENTATIVAS"
230 SOUND 1,478:SOUND 1,319:SOUND 1,379
240 X=TIME:WHILE TIME-X<1000:WEND
250 INPUT "OUTRO JOGO (S/N)? ";R$
260 IF UPPER$(LEFT$(R$,1))="S" THEN 40
270 CLS
```

A utilizar somente no
Continente e Regiões
Autônomas

RSF

IMPRESSO — RESPOSTA
Autorizado pelos CTT

Não carece de selo.
O porte será pago
pelo destinatário.



PUBLICAÇÕES EUROPA-AMÉRICA
Apartado 8
2726 MEM MARTINS CODEX

no écran de-
o de jogadas
nas isso tem
is que as ro-

uma variável
abeto, desor-

n A\$ com le-
do alfabeto

Para se entrar a permuta na linha 100.

As linhas 190 e 200 transformam numa variável numérica a variável alfanumérica entrada.

A linha 210 chama uma primeira vez a rotina de inversão.

A linha 240 chama segunda vez a rotina de inversão.

A linha 250 contabiliza as tentativas.

A linha 170 controla os resultados.

PARA JOGAR:

Fazer correr o programa com RUN.

A palavra inicial é apresentada, sendo cada letra referenciada por um número de 0 a 9.

A linha
OUTRO JOG

EXTENSÃO:
Poder-se
com letras, c
Pode-se
que desempe

```
10 REM M
20 RANDO
30 INK 0
40 CLS
50 INPUT
60 IF C<
  40
70 E=1:F
80 X$=ST
90 X$=RI
100 LOCA
R0 TENTA
110 LOCA
120 LOCA
;TAB(18-
130 F$=L
140 A=0:
150 FOR
160 IF M
A+1
170 IF M
B+1
180 NEXT
190 LOCA
PRINT B
200 F=E MOD 17:E=E+1
210 GOTO 120
220 LOCATE 1,23:PRINT "ADIVINHOU ";:PEN
3:PRINT X$;;PEN 1:PRINT " EM ";:PEN 2:PR
INT E;;PEN 1:PRINT "TENTATIVAS"
230 SOUND 1,478:SOUND 1,319:SOUND 1,379
240 X=TIME:WHILE TIME-X<1000:WEND
250 INPUT "OUTRO JOGO (S/N)? ";R$
260 IF UPPER$(LEFT$(R$,1))="S" THEN 40
270 CLS
```

As Publicações e as Livrarias EUROPA-AMÉRICA

Convidam-no a juntar-se
àquele grupo de leitores exigentes
para quem um livro,
além de uma companhia agradável,
é uma companhia útil.

Assim, teremos muito gosto
em passar a enviar-lhe, regularmente,
informações sobre os livros
que publicamos.

o Editor

Nome: _____

Profissão: _____

Morada: _____ Tel.: _____

Cód. Postal: _____ Localidade: _____

Encontrei este postal no livro: _____

Que adquiri numa livraria Que encomendei pelo correio

NÓS EDITAMOS O LIVRO QUE VOCÊ PROCURA!

IMPORTANTE: Se desejar alguns livros de nossa edição, também poderá usar este postal. Bastará preencher o espaço que se segue. Todas as encomendas são confidenciais.

Referência Título e Autor

Ordem

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexão.

Dificuldade do jogo: média.

O JOGO:

As dez primeiras letras do alfabeto são apresentadas no écran desordenadamente; há que pô-las por ordem com o mínimo de jogadas possíveis, por permutas sucessivas de duas das letras; mas isso tem também o inconveniente de fazer permutar as duas letras que as rodeiam, de acordo com o mecanismo explicado a seguir:

```
B A D G H I J C F E
0 1 2 3 4 5 6 7 8 9: permutar 8 e 3
      ┌──────────┬──────────┐
      │           │           │
B A D F H I J E G C
0 1 2 3 4 5 6 7 8 9: resultado
```

O PROGRAMA:

Os ciclos encaixados I (60-120) e J (80-100) definem uma variável alfanumérica A\$, contendo as dez primeiras letras do alfabeto, desordenadas.

A linha 140 chama a rotina 600, que imprime no écran A\$ com letras multicores. A linha 150 imprime os números de letras do alfabeto para a permuta.

Faz-se entrar a permuta na linha 180.

As linhas 190 e 200 transformam numa variável numérica a variável alfanumérica entrada.

A linha 210 chama uma primeira vez a rotina de inversão.

A linha 240 chama segunda vez a rotina de inversão.

A linha 250 contabiliza as tentativas.

A linha 170 controla os resultados.

PARA JOGAR:

Fazer correr o programa com RUN.

A palavra inicial é apresentada, sendo cada letra referenciada por um número de 0 a 9.

À pergunta INVERSÃO (IJ)? responde-se fazendo entrar um número de dois algarismos, xy ; x é o número da primeira letra a permutar e y o da segunda. As duas inversões são então sucessivamente efectuadas, e recomeça-se.

O número de jogadas será apresentado no fim do jogo.

EXTENSÕES POSSÍVEIS:

Pode-se imaginar um jogo contra o computador, que calcularia a pontuação do jogador em função dos seus próprios resultados; pode-se também modificar as regras que regem as permutas.

```
10 REM ORDEM
20 RANDOMIZE(TIME)
30 MODE 0:FOR I=2 TO 11:INK I,I:NEXT I
40 CLS
50 A$=""
60 FOR I=1 TO 10
70 R$=CHR$(INT(RND*10+65))
80 FOR J=1 TO I-1
90 IF R$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A4+R$
120 NEXT I
130 S=1
140 LOCATE 1,4:PRINT "RESULTADO: ";:GOSU
B 500
150 PRINT TAB(11)"0123456789"
160 LOCATE 1,9:PRINT "TENTATIVA NUMERO";
S
170 IF A$="ABCDEFGHIJ" THEN LOCATE 2,24:
PRINT "CONSEGUIU EM ";S;" JOGADAS":SOUND
1,478:SOUND 1,319:SOUND 1,379:END
180 LOCATE 1,12:PRINT CHR$(20);:INPUT "I
NVERSAO (IJ)? ";R$
190 I=ASC(LEFT$(R$,1))-47
200 J=ASC(RIGHT$(R$,1))-47
210 GOSUB 500
220 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9
" THEN 240
230 I=I-1:J=I+2
240 GOSUB 500
250 S=S+1
260 GOTO 140
500 B$=""
510 N$=MID$(A$,I,1):M$=MID$(A$,J,1)
520 FOR T=1 TO LEN(A$)
530 IF T=I THEN B$=B$+M$:GOTO 560
540 IF T=J THEN B$=B$+N$:GOTO 560
550 B$=B$+MID$(A$,T,1)
560 NEXT I
570 A$=B$
580 RETURN
600 FOR T=1 TO 10
610 AF$=MID$(A$,T,1):PEN (ASC(AF$)-63):P
RINT AF$;
620 NEXT T
630 PEN 1:RETURN
```

O enforcado

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de letras (versão para dois jogadores).

Dificuldade do jogo: média.

O JOGO:

Será sem dúvida difícil descobrir o nome do inventor deste jogo tão em voga, mas é certo que ele existe desde tempos imemoriais, por vezes praticado de formas bastante diferentes (por exemplo, reconstituir um porquinho lançando dados).

A aparição dos jogos em microcomputadores e a magia dos écrans de televisão deram-lhe um certo renovo: na versão retida, um dos jogadores deve fazer entrar uma palavra de qualquer comprimento. O computador apresentará então, no écran, traços em vez das letras dessa palavra e pede ao segundo jogador que proponha uma letra. Cada resposta correcta faz aparecer a letra no seu local exacto na palavra, e cada resposta falsa faz progredir o desenho do enforcamento.

Para ganhar basta conseguir descobrir a palavra antes de o boneco do desenho ser enforcado.

O PROGRAMA:

A palavra a adivinhar é introduzida no início do programa (linha 30); a variável alfanumérica B\$, na linha 40, contém todos os elementos do desenho, em notação hexadecimal.

O ciclo I (60-80) fabrica uma palavra P\$, que contém traços no lugar de cada uma das letras da palavra inicial.

O ciclo I (120-180) ocupa-se do desenho (mais ou menos completo, consoante o valor de P). O ciclo K (140-160) transforma os valores hexadecimais em elementos de desenho.

As linhas 190-200 imprimem o resultado e param o jogo.

A letra tentada é feita entrar na linha 210 e testada na linha 230; a rotina 260-280 substitui cada traço pela letra correspondente.

PARA JOGAR:

Depois de fazer correr o programa com RUN, passar para maiúsculas. Um dos jogadores deve fazer entrar a sua palavra e teclar ENTER. Aparecem no écran traços no lugar de cada letra da palavra e é impressa a ordem ENTRE UMA LETRA. O outro jogador responde então, propondo uma letra seguida de ENTER.

Se a letra servir irá substituir uma ou mais traços; se não servir, a força começa a ser desenhada.

No caso de perder, a palavra exacta será apresentada.

EXTENSÕES POSSÍVEIS:

Será interessante elaborar uma lista de palavras com que o AMS-TRAD poderá trabalhar à vontade; poder-se-á também melhorar o desenho (redefinindo os caracteres, por exemplo) e modificar o número de tentativas permitidas consoante a dificuldade da palavra.

```
10 REM O ENFORCADO
20 CLS:INK 2,0,9
30 INPUT "A SUA PALAVRA: ";R$
40 B$="9B2209B952095952095969B9C20952095
9595959595959595969F9C20E120202020"
50 P$=""
60 FOR I=1 TO LEN(R$)
70 P$=P$+"_"
80 NEXT I
90 P=11
100 CLS
110 LOCATE 1,3:PRINT P$
120 FOR I=11 TO P STEP -1
130 PE$=""
140 FOR K=0 TO 2
150 PE$=PE$+CHR$(VAL("&" + MID$(B$,1+6*(I-1)+2*K,2)))
160 NEXT K
170 PEN 3:LOCATE 22,12-I:PRINT PE$;;PEN
1
180 NEXT I
190 IF P$=R$ THEN PEN 2:LOCATE 1,12:PRINT
"BRAVO!":SOUND 1,478:SOUND 1,319:SOUND
1,379:GOTO 290
200 IF P=1 THEN INK 2,3,15:PEN 2:PRINT "
PERDEU! A PALAVRA ERA: ";R$:SOUND 1,478:
GOTO 290
210 LOCATE 1,20:PRINT CHR$(20);;INPUT "E
NTR E UMA LETRA: ";S$:S$=LEFT$(UPPER$(S$+
" "),1)
220 T=0:I=0
230 I=INSTR(I+1,R$,S$):IF I<>0 THEN GOSU
B 260:GOTO 230
240 IF T=0 THEN P=P-1
250 GOTO 110
260 T=1
270 P$=LEFT$(P$,I-1)+S$+RIGHT$(P$,LEN(P$
)-I)
280 RETURN
290 GOTO 290
```

A bisca

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de cartas.

Dificuldade do jogo: fácil.

O JOGO:

O jogo proposto é sobretudo uma simulação, realizada com 32 cartas, da distribuição de quatro mãos de cinco cartas. Poderá ser utilizado para o estudo de problemas em que intervenham mãos de 20 cartas em 32.

O PROGRAMA:

Na linha 50 é criado um quadro que possa conter as 20 cartas.

O ciclo I (70-130) assegura a tiragem aleatória de 20 números, tirados entre 32; o ciclo J (100-120) verifica que se não realiza duas vezes a mesma tiragem.

O segundo ciclo I (150-290) assegura seguidamente a apresentação no écran, sendo os nomes das cartas apresentadas sob a forma de números ou abreviaturas e a cor é descodificada e apresentada em claro graças às linhas 200 a 260.

PARA JOGAR:

Basta teclar RUN para obter imediatamente a apresentação de quatro maços de cinco cartas cada um (uma mão). Uma batida dupla em ESC permite recomeçar.

EXTENSÕES POSSÍVEIS:

Poder-se-á facilmente preparar uma apresentação mais gráfica redefinindo os caracteres do AMSTRAD; poder-se-á igualmente usar este programa como núcleo para a criação de jogos de cartas no computador (que poderá, se necessário, servir de opositor; ver o exemplo do jogo «Póquer», neste mesmo livro).

```

10 REM A BISCA
20 RANDOMIZE (TIME)
30 INK 0,9:INK 2,0:INK 3,3
40 CLS
50 DIM A(20)
60 LOCATE 12,12:PRINT "AGUARDE, ESTOU A
FAZER A TIRAGEM"
70 FOR I=1 TO 20
80 A(I)=1+INT(RND*32)
90 IF I=1 THEN 130
100 FOR J=1 TO I-1
110 IF A(I)=A(J) THEN 80
120 NEXT J
130 NEXT I
140 CLS
150 FOR I=1 TO 20
160 S#=CHR$(227)
170 S=1+A(I) MOD 8
180 R#=STR$(S+6)+" ":R#=RIGHT$(R#,LEN(R#
)-1)
190 R#=LEFT$(R#,2)
200 IF S=5 THEN R#="VA"
210 IF S=6 THEN R#="DA"
220 IF S=7 THEN R#="RE"
230 IF S=8 THEN R#="AS"
240 IF A(I)<=24 THEN S#=CHR$(226)
250 IF A(I)<=16 THEN S#=CHR$(229)
260 IF A(I)<=9 THEN S#=CHR$(228)
270 LOCATE 3+10*INT((I-1)/5),2*K+1:PEN (
3+((ASC(S#) MOD 3)=1)):PRINT R#;" ";S#
280 K=(K+1) MOD 5
290 NEXT I
300 GOTO 300

```

Um cálculo quebra-cabeças

REFERÊNCIAS:

Dificuldade de programação: média.
Categoria de jogo: reflexão (dois jogadores).
Dificuldade do jogo: difícil.

O JOGO:

Pratica-se com dois jogadores, um desempenhando o papel de codificador e o outro o de decodificador.

O codificador parte de um número (proposto pelo AMSTRAD) e deve transformá-lo com a ajuda de cinco operações que ele próprio escolhe, tal como os operadores.

O resultado que ele obteve e as operações que utilizou serão expostos no écran; o decodificador deverá então tentar encontrar o número inicial efectuando as operações em sentido inverso.

O PROGRAMA:

A linha 50 permite obter o número inteiro tomado como ponto de partida, compreendido entre 5 e 20.

Um ciclo I (110-180) permite fazer entrar sucessivamente as cinco operações.

Quatro rotinas são utilizadas para os cálculos (nas linhas 1000 a 4000).

Um ciclo WHILE/WEND (120-140) assegura a validade das operações.

Após cada cálculo o resultado é apresentado no écran (só o codificador deverá ver o écran durante esta operação) e o resultado definitivo permanecerá exposto para uso do decodificador no fim dos cálculos (linha 170).

A linha 190 apresenta as cinco operações utilizadas, assim como os correspondentes operadores.

PARA JOGAR:

Faz-se correr o programa com RUN.

O número inicial é apresentado no écran, depois a ordem ENTRE CINCO OPERACOES; dever-se-á fazê-las entrar uma após outra, sob a forma, por exemplo: + 4 seguido de ENTER.

O resultado será apresentado após cada operação que se fez entrar. No fim, as operações são todas apresentadas numa só linha, e é enviada a mensagem CORAGEM!... dirigida ao decodificador (não há controlo na decodificação).

EXTENSÕES POSSÍVEIS:

Poder-se-á, claro, pensar em melhorar a parte de decodificação do programa introduzindo controlos sobre o resultado; pode-se também pensar em deixar o AMSTRAD participar mais, confiando-lhe o papel de codificador, etc.

```
10 REM UM CALCULO QUEBRA-CABECAS
20 RANDOMIZE (TIME)
30 INK 0,3:INK 3,18,2
40 CLS
50 X=INT(RND*16+5)
60 PRINT "NUMERO INICIAL: ";X
70 PRINT "ENTRE CINCO OPERACOES"
80 PRINT "NA FORMA +5, /2, *4..."
90 F=X
100 DIM R$(5)
110 FOR I=1 TO 5
120 WHILE INSTR("*/-/",LEFT$(R$(I)+" ",1))=0
130 LOCATE 1,6:PRINT CHR$(18);:INPUT "A
SUA OPERACAO: ";R$(I)
140 WEND
150 A=INSTR("*/-/",LEFT$(R$(I),1))
160 ON A GOSUB 1000,2000,3000,4000
170 LOCATE 1,9:PRINT "NUMERO OBTIDO: ";C
HR$(20);F
180 NEXT I
190 PRINT:PRINT:PRINT "OPERACOES: ";:PEN
2:PRINT R$(4);" ";R$(1);" ";R$(5);" ";R
$(3);" ";R$(2)
200 PEN 3:PRINT:PRINT:PRINT "CORAGEM!..."
"
210 SOUND 1,478:SOUND 1,319:SOUND 1,379:
PEN 1:END
1000 F=F*VAL(RIGHT$(R$(I),LEN(R$(I))-1))
1010 RETURN
2000 F=F+VAL(RIGHT$(R$(I),LEN(R$(I))-1))
2010 RETURN
3000 F=F-VAL(RIGHT$(R$(I),LEN(R$(I))-1))
3010 RETURN
4000 F=F/VAL(RIGHT$(R$(I),LEN(R$(I))-1))
4010 RETURN
```

Enforcado invertido

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo com palavras.

Dificuldade do jogo: fácil.

O JOGO:

«O Enforcado» é um jogo de letras muito conhecido, aliás apresentado neste livro, no qual se trata de adivinhar uma palavra escondida pelo computador e em que cada letra não descoberta nos aproxima cada vez mais do enforcamento.

Aqui é o AMSTRAD que vai procurar adivinhar a nossa palavra, mas ele joga um tanto ao acaso e não possui a nossa intuição; necessitará, portanto, regra geral, de bastantes mais tentativas para descobrir a solução. Para aumentar as suas capacidades usa-se uma fórmula que lhe permite tirar primeiro as vogais, depois as consoantes mais correntes e, por fim, as consoantes mais raras.

O PROGRAMA:

A linha 40 permite registar, numa variável alfanumérica D\$, as letras do alfabeto pela seguinte ordem: vogais, consoantes usuais, outras consoantes. A palavra a adivinhar é introduzida na linha 60.

A linha 70 imprime traços no lugar das letras a encontrar.

O ciclo WHILE/WEND (100-220) assegura a continuidade da busca.

O ciclo na linha 120 serve unicamente para simular o tempo de busca de uma letra.

A fórmula colocada na linha 130 permite modificar ligeiramente a probabilidade de tiragem, a favor das vogais e das consoantes usuais.

A linha 150 provoca nova tiragem, e o ciclo I (180-200) reimprime os traços e as letras descobertas.

A linha 210 incrementa o contador de tentativas.

A linha 230 imprime o número de tentativas realizadas, no fim da partida.

PARA JOGAR:

Fazer correr o programa com RUN e passar para maiúsculas.

Fazer entrar a palavra a descobrir, seguida de ENTER.

São então impressos traços (um por cada letra) e o AMSTRAD propõe uma primeira letra, verifica se ela serve e continua automaticamente, até ter descoberto a palavra.

EXTENSÕES POSSÍVEIS:

Será sem dúvida interessante, para crianças, prever uma versão automática em que o jogador responderá SIM ou NÃO a cada letra proposta pelo AMSTRAD; poder-se-á também pensar em acrescentar uma apresentação gráfica, por exemplo uma forca ou uma viatura que avança para um precipício, etc.

```
10 REM ENFORCADO INVERTIDO
20 RANDOMIZE(TIME)
30 CLS
40 D$="AEIOUBCDFGHJKLMNPQRSTVXZ"
50 C=1:T$=""
60 INPUT "ENTRE A PALAVRA: ";R$
70 CLS
80 T$=STRING$(LEN(R$)," ")
90 LOCATE 1,6:PEN 3:PRINT T$:PEN 1
100 WHILE T$<>R$
110 LOCATE 1,9:PRINT "ESTOU A PROCURAR..
. TENTATIVA ";C
120 FOR J=1 TO 1000:NEXT J
130 X=-6*(RND<0.7)-13*(RND<0.5)-7*(RND<0
.4)
140 X=INT(X*RND+1)
150 E#=MID$(D$,X,1):IF E#="-" THEN 130
160 D$=LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
170 PRINT:PRINT:PRINT "EU ESCOLHO: ";:PE
N 2:PRINT E#:SOUND 1,239:SOUND 1,213:SOU
ND 1,190:PEN 1
180 FOR I=1 TO LEN(R$)
190 IF E#=MID$(R$,I,1) THEN T#=LEFT$(T$,
I-1)+E#+RIGHT$(T$,LEN(T$)-I):PEN 2:LOCAT
E I,6:PRINT MID$(T$,I,1):PEN 1
200 NEXT I
210 C=C+1
220 WEND
230 LOCATE 11,20:PAPER 3:PEN 2:PRINT "AC
ABEI EM ";C-1;" TENTATIVAS"
240 SOUND 1,478:SOUND 1,319:SOUND 1,379
250 GOTO 250
```

«*Nicómaco*»

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexão (cálculo mental).

Dificuldade do jogo: média.

O JOGO:

Utiliza uma propriedade um pouco particular dos inteiros de 1 a 100: poderem ser identificados sem ambiguidades pelos restos das suas divisões por 3, 5 e 7.

O que aqui propomos é um programa codificador que simplesmente apresenta no écran os restos da divisão de um dado inteiro N pelos divisores indicados.

O jogo consiste em tentar descobrir o inteiro (mas, nesta versão, você só tem direito a uma tentativa).

O PROGRAMA:

A linha 50 permite confeccionar o inteiro.

O ciclo I (80-110) calcula os três restos sucessivos, e a linha 100 permite apresentá-los sob uma forma bastante visual, bem colocados.

As linhas 60 e 70 existem para indicar de que divisão se trata.

O número proposto pelo jogador entra na linha 120; é impresso e testado na linha 130 (impressão da mensagem de fim de partida: BRAVO ou PERDEU).

A forma de envelope sonoro definida na linha 150 permite tocar as notas (SOUND) da forma indicada na linha 160.

As linhas 170 e 180 ocupam-se da geração da próxima partida, conforme a resposta do jogador.

PARA JOGAR:

Faz-se correr o programa teclando RUN.

Os três restos serão então apresentados na mesma linha, simbolizados por estrelas (a ausência de quadrados significa «resto 0»).

À pergunta SUA TENTATIVA? responder-se-á fornecendo um inteiro compreendido entre 0 e 100, seguido de ENTER.

Obter-se-á em resposta uma mensagem de PERDEU ou BRAVO, e a seguir a pergunta OUTRO JOGO (S/N)?.

Para fazer nova tentativa responde-se S, ou N no caso contrário, seguido de ENTER.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em realizar o programa decodificador (damos os restos e o AMSTRAD deverá fornecer-nos N) e verificar se a propriedade pode ser ampliada (efectuando mais divisões, por exemplo).

```
10 REM "NICOMACO"
20 RANDOMIZE(TIME)
30 ZONE 8
40 CLS
50 X=INT(RND*100+1)
60 PRINT TAB(6)"RESTO DA DIVISAO POR"
70 PRINT:PRINT ", "3", "5", "7"
80 FOR I=1 TO 3
90 A=X MOD (2*I+1)
100 PAPER 3:PEN 0:LOCATE 8*I,5:PRINT STR
ING$(A,233)
110 NEXT I
120 PAPER 0:PEN 1:LOCATE 1,10:INPUT "SUA
TENTATIVA? ",N
130 PAPER 2:PEN 3:PRINT:IF N=X THEN PRIN
T "BRAVO!" ELSE PRINT "PERDEU!"
140 PAPER 0:PEN 1
150 ENV 1,1,15,10,1,-15,10
160 FOR I=1 TO 3:SOUND 1,413,20,0,1:NEXT
I
170 PRINT:INPUT "OUTRO JOGO (S/N)? ",R#
180 IF LEFT$(UPPER$(R#+ " "),1)="S" THEN
40
```

«*Mastermind*»

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: lógica (um só jogador).
Dificuldade do jogo: difícil.

O JOGO:

Tendo-se tornado um grande clássico dos jogos de micromputador, este jogo desenvolve o raciocínio dedutivo; trata-se de descobrir o mais rapidamente possível um número de n algarismos (poderá tomar-se n compreendido entre 1 e 6), dispondo de informações em cada tentativa.

Efectivamente, o AMSTRAD responderá indicando por um lado o número de algarismos que, no número proposto, estão exactamente no mesmo lugar que no número a descobrir, e, por outro lado, o número de algarismos que figuram no número a descobrir mas não exactamente no local correcto.

O PROGRAMA:

A escolha do número a descobrir é efectuada nas linhas 40 a 60. O número é fabricado na linha 80 (poderá haver algarismos repetidos).

A apresentação é preparada na linha 100.

O ciclo WHILE/WEND (120-140) trata do fim da partida.

Um primeiro ciclo, I (160-210), testa os algarismos que estão no lugar certo.

Um segundo ciclo, J (180-20), testa os algarismos que estão presentes e mal colocados.

Os contadores estão nas linhas 170 e 190, ao mesmo tempo que o emprego da variável EB\$ permite evitar contagens duplas.

A linha 220 imprime o resultado, e a linha 230 contabiliza as tentativas.

PARA JOGAR:

Teclia-se RUN e escolhe-se o comprimento do número desejado (de 1 a 6), seguido de ENTER.

Após a impressão do cabeçalho «No. NUMERO NO LUGAR CERTO» introduz-se o número próprio, seguido de ENTER.

Obtém-se imediatamente a resposta (em caso de algarismo duplo, os resultados podem ser de interpretação um tanto delicada).

Em seguida introduz-se a nova tentativa, e assim por diante.

Quando se descobriu o número obtém-se a mensagem O NÚMERO ERA X. FOI DESCOBERTO EM... JOGADAS.

Pode-se sair do programa em qualquer momento teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se procurar evitar a presença de algarismos duplos no número a descobrir, pode-se também prever uma mudança de página automática, com chamada dos resultados precedentes; poder-se-á também realizar jogos análogos com palavras ou outros símbolos (cor, tamanho...)

```
10 REM "MASTERMIND"
20 DEFINT N
30 RANDOMIZE(TIME)
40 WHILE N<1 OR N>6
50 INPUT "QUANTOS ALGARISMOS? ";N
60 WEND
70 E=1:F=1:X#=STR$(INT(RND*9^N+10^(N-1)))
)
80 X#=RIGHT$(X$,LEN(X$)-1)
90 CLS
100 PRINT "   No.          NUMERO          LUGAR
CERTO"
110 PRINT STRING$(40,154)
120 WHILE E#<>X#
130 LOCATE 1,3+F:PRINT E;CHR$(18);TAB(13)
);:INPUT " ",E#
140 X=0:Y=X
150 EB#=E#
160 FOR I=1 TO N
170 IF MID$(X$,I,1)=MID$(E#,I,1) THEN X=X+1:MID$(EB#,I,1)=" ":GOTO 210
180 FOR J=1 TO N
190 IF MID$(X$,I,1)=MID$(EB#,J,1) THEN Y=Y+1:MID$(EB#,J,1)=" ":GOTO 210
200 NEXT J
210 NEXT I
220 LOCATE 24,3+F:FEN 2:PRINT X;TAB(33);
:PEN 3:PRINT Y:PEN 1
230 E=E+1:F=(F+1) MOD 18
240 WEND
250 SOUND 1,478:SOUND 1,319:SOUND 1,379
260 LOCATE 11-(N/2),22:PAPER 3:PRINT "O
CODIGO ERA ";E#;" . DESCOBERTO EM ";E-1;"
TENTATIVAS"
270 GOTO 270
```

Idem

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de cálculo mental (dois jogadores).

Dificuldade do jogo: consoante o nível escolhido.

O JOGO:

O AMSTRAD escolhe um número de n algarismos (consoante o nível de dificuldade indicado no início). Este número (o mesmo inicialmente para os dois jogadores) será, a cada vez que se jogue, regularmente decrementado de uma quantidade igualmente escolhida ao acaso pelo AMSTRAD no início do jogo; em cada jogada deve-se escolher uma coluna (das dezenas, das centenas, etc.) e acrescentar nesta coluna um certo inteiro; o objectivo é obter em dado momento a apresentação de um número com todos os algarismos idênticos.

Cada resultado será mostrado durante um breve momento, após o que passa a vez ao jogador seguinte, que voltará a partir do seu resultado precedente.

O PROGRAMA:

Uma variável numérica, N, é dimensionada a 2 na linha 40, para conter os resultados sucessivos dos dois jogadores.

O ciclo WHILE/WEND (45-55) permitirá fazer entrar o nível de dificuldade escolhido.

O número inicial é confeccionado pela linha 70. A quantidade que intervirá nas decrementações é obtida na linha 90.

A linha 120 testa se o resultado não é demasiado pequeno.

A linha 130 imprime o número do jogador.

O número é apresentado na linha 140 e o do jogador entra na linha 160 (número de dois algarismos).

A linha 170 incrementa a coluna escolhida, e a linha 180 decrementa o número da quantidade escolhida no início.

O resultado será apresentado durante um curto momento (linha 210). O ciclo I (260-280) permite verificar se todos os algarismos são idênticos.

A impressão e o fim da partida são efectuadas na linha 290. As linhas 300 e 310 tratam da mudança de jogador.

PARA JOGAR:

Teclar RUN para fazer correr o programa.

O número inicial será impresso no écran para o jogador número 1, que joga teclando xy , seguido de ENTER; x é o algarismo a adicionar e y é o número da coluna onde ele deve ser adicionado.

O resultado (provisório, se for demasiado pequeno) será então apresentado durante um curto instante, e passa à vez do segundo jogador, e assim por diante. Pode-se parar em qualquer momento teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á alargar o jogo a mais de dois jogadores, melhorar a escolha do número que intervém na decretaçãoção, melhorar os diversos controlos, deixar impressos no écran os diferentes resultados, etc.

```
10 REM IDEM
20 RANDOMIZE(TIME)
40 CLS:H=1:DIM N(2)
45 WHILE N<1 OR N>8
50 INPUT "NIVEL (2-7): ";N
55 WEND
60 U=10:P=0
70 N(1)=INT((U^N-U^(N-1))*RND+U^(N-1))
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND+U^(N-1))
100 IF H=1 THEN P=P+1
110 CLS
120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(H))
130 LOCATE 1,3:PRINT "JOGADOR ";PEN 1+H
:PRINT H:;PEN 1:PRINT "TENTATIVA ";P
140 LOCATE 1,6:PRINT "NUMERO: ";LOCATE 1
5-N,6:;PEN 1+H:PRINT N(H):PEN 1
150 LOCATE 9,7:PRINT "7654321"
160 LOCATE 1,10:PRINT "O SEU JOGO (N E C
) NA FORMA NC ";PEN H+1:INPUT "",R#:PEN
1
170 N(H)=N(H)+VAL(LEFT$(R#,1))*U^(VAL(RI
GHT$(R#,1))-1)
180 N(H)=N(H)-A
190 IF N(H)>U^N THEN N(H)=N(H)-U^N
200 N(H)=INT(N(H))
210 LOCATE 1,13:PRINT "RESULTADO: ";N(H)
220 SOUND 1,478:SOUND 1,426:SOUND 1,379
230 FOR J=1 TO 1200:NEXT J
240 R#=STR$(N(H))
250 R#=RIGHT$(R#,LEN(R#)-1)
260 FOR I=1 TO LEN(R#)-1
270 IF MID$(R#,I,1)<>MID$(R#,I+1,1) THEN
300
280 NEXT I
290 PRINT:PRINT:;PEN H+1:PRINT "BRAVO! JO
GADOR ";H:SOUND 1,478:SOUND 1,319:SOUND
1,379:END
300 H=1+(H MOD 2)
310 GOTO 100
```

Rainhas no tabuleiro de xadrez

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: reflexão.
Dificuldade do jogo: difícil.

O JOGO:

O programa utiliza um algoritmo de alto poder de realização para resolver um problema que deu dores de cabeça a investigadores, mesmo a alguns tão célebres como Gauss, que não pôde descobrir todas as soluções. Trata-se de descobrir todas as maneiras de posicionar rainhas num tabuleiro de xadrez, de tal modo que nenhuma possa tomar outra.

O programa proposto é muito flexível, porque você pode dispor mesmo da possibilidade de escolher o tamanho do seu tabuleiro de xadrez, e os resultados serão apresentados simultaneamente de forma visual e de forma numérica.

O PROGRAMA:

A linha 30 permite-lhe escolher o tamanho do tabuleiro de xadrez.

Uma variável D será dimensionada de acordo com essa escolha, na linha 40.

Um ciclo de exploração principal I (50-110) contém um subciclo J (80-100) que realiza o controlo de não alinhamento (linha 90).

Se, no decorrer da exploração, se encontra um alinhamento, volta-se para trás, utilizando as linhas 230 a 270, e recomeça-se.

O ciclo F (130-210) assegura a impressão do tabuleiro de xadrez, e o ciclo K (160-180), que eventualmente poderá ser suprimido, assegura a impressão do resultado sob a forma de um código numérico correspondente à posição das peças em cada coluna.

A linha 220 permite uma paragem após a apresentação no écran de cada uma das soluções.

PARA JOGAR:

Faz-se correr o programa com RUN, depois teclia-se a dimensão do tabuleiro de xadrez desejada, e faz-se ENTER.

Basta esperar, com paciência (quanto maior for o tabuleiro mais longa será a espera). A primeira solução será exposta no écran e depois

o programa pára; é preciso carregar em qualquer tecla para voltar a funcionar. Para terminar a exploração sem ter atingido todas as soluções basta teclar ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em encontrar algoritmos para todos os problemas análogos (posição de torres, deslocação de um cavalo, etc.), tentando utilizar o máximo de ciclos WHILE/WEND.

```
10 REM RAINHAS NO TABULEIRO DE XADREZ
20 CLS:INK 0,9:INK 1,21:INK 2,0:INK 3,26
,15:SPEED INK 5,5
30 INPUT "TAMANHO DO TABULEIRO? :",S
40 DIM D(S)
50 I=I+1
60 D(I)=1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J)
THEN 240
100 NEXT J
110 I=I+1:IF I<=S THEN 60
120 CLS
130 FOR F=1 TO S
140 LOCATE 2*F+1,1:PRINT F
150 LOCATE 1,2*F+1:PRINT F
160 FOR K=1 TO S
170 PEN 2:LOCATE 2*K+2,2*F+1:PRINT ".":P
EN 1
180 NEXT K
190 LOCATE 2*D(F)+2,2*F+1:PEN 3:PRINT "*"
":PEN 1
200 LOCATE F+1,20:PEN 3:PRINT RIGHT$(STR
$(D(F)),1):PEN 1
210 NEXT F
220 SOUND 1,478:SOUND 1,319:SOUND 1,379:
LOCATE 6,23:INPUT CARREGUE EM <ENTER> PA
RA CONTINUAR",R#
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN 70
260 I=I-1
270 IF I<>0 THEN 240
```

Palavras

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: jogo de letras (para um ou vários jogadores).

Dificuldade do jogo: média.

O JOGO:

É difícil imaginar jogo mais simples que este, no seu princípio: contenta-se com imprimir no écran, desordenadamente, letras retiradas de uma trintena de palavras. Mas tente jogá-lo e verá que não é assim tão fácil, variando a dificuldade, evidentemente, consoante a lista de palavras que previamente tenha introduzido como DATA.

O PROGRAMA:

A linha 40 permite dimensionar a variável A\$(I), que conterà as palavras.

O ciclo I (50) toma a lista das palavras (inscritas como DATA nas linhas 240 a 280).

O ciclo WHILE/WEND das linhas 70-90 escolhe três cores de tinta diferentes entre si e diferentes da do fundo.

A linha 120 escolhe uma palavra ao acaso; o ciclo I (150-220) controla se uma letra não foi já tirada, e retoma as letras desordenadas.

A linha 230 imprime no écran as letras, e a linha 240 aguarda a vossa proposta, que é testada na linha 250.

O número de tentativas é incrementado na linha 260.

O fim da partida é controlado pela linha 270, e a pontuação é impressa na linha 290.

PARA JOGAR:

Faz-se correr o programa com RUN. Uma série de letras desordenada é impressa no écran, e é preciso descobrir a palavra correspondente (o tempo é controlado).

Tecléia-se a palavra suposta, depois ENTER, e será impressa no écran uma nova série de letras se se responder S à pergunta OUTRA TENTATIVA (S/N)?.

A pontuação e o número de tentativas são apresentadas no écran no fim da partida.

EXTENSÕES POSSÍVEIS:

Pode-se imaginar minutar o tempo de aparição da palavra no ecrã, ou fazer desfilas as letras uma após outra.

Pode-se também avolumar as dificuldades introduzindo *jokers*, jogar com vários jogadores, contabilizar os pontos, etc.

```
10 REM PALAVRAS
20 RANDOMIZE(TIME)
30 DEFINT E
40 DIM A$(30):S=0:E=0
50 FOR I=1 TO 30:READ A$(I):NEXT I
60 F=0:E1=0:E2=0:E3=0
70 WHILE ((F-E1)*(F-E2)*(F-E3)*(E1-E2)*
E1-E3)*(E2-E3)=0
80 E1=RND*26:E2=RND*26:E3=RND*26
90 WEND
100 CLS:INK 0,F:INK 1,E1:INK 2,E2:INK 3,
E3
110 WHILE A$(Z)=""
120 Z=INT(RND*30+1)
130 WEND
140 R$=A$(Z):A$="":A$(Z)="" :V$=R$
150 FOR I=1 TO LEN(R$)
160 Z=1
170 WHILE MID$(R$,Z,1)="-"
180 Z=INT(RND*LEN(R$)+1)
190 WEND
200 A$=A$+MID$(R$(R$,Z,1)
210 MID$(R$,Z,1)="-"
220 NEXT I
230 PRINT "AS LETRAS SAO: ";:PEN 3:PRINT
A$:SOUND 1,478:SOUND 1,426:SOUND 1,379:
PEN 1
240 LOCATE 1,6:PRINT "A SUA SOLUCAO: ";:
PEN 2:INPUT "",S$:PEN 1
250 IF UPPER$(S$)=V$ THEN S=S+1
260 E=E+1
270 LOCATE 1,11:INPUT "OUTRA TENTATIVA (
S/N)? ",R$
280 IF LEFT$(UPPER$(R$+" "),1)="S" AND E
<30 THEN 60
290 LOCATE 1,16:PRINT "A SUA PONTUACAO :
";S;"EM";E;"TENTATIVAS"
300 SOUND 2,478:SOUND 2,319:SOUND 2,379
310 DATA PRESUNTO,ENERGUMENO,FILIADO,ERU
DITO,INSIGNIFICANTE,FATALISTA
320 DATA IMPRESSIONANTE,ATRAENTE,ANTICON
STITUCIONAL,ARANHA,INTREPIDO,COMPUTADOR
330 TREGUA,BANDEIROLA,SENADOR,ACTUALIDAD
E,CANIBAL,DIEDRO
340 DATA BEBADO,BARCO,ESFANADOR,SIGNATAR
IO,CONTRATUAL,PROCESSO
350 DATA CALOR,INVERNAL,DESODORIZANTE,LU
XACAO,EBONITE,NOVICO
```

Permutas

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: reflexão.

Dificuldade do jogo: média.

O JOGO:

O AMSTRAD pede-lhe antes de mais o número de letras que deseja permutar (não mais de 12). A partir daí vai imprimir no écran o número correspondente de letras tomadas no início do alfabeto, primeiro desordenadas (INICIO) e depois ordenadas (FIM).

O PROGRAMA:

A linha 40 permite escolher o número de letras, e a linha 50 controla que se não pede mais de 12.

A linha 80 realiza a mistura, tirando ao acaso a nova posição de cada letra. As letras são depois guardadas, à medida que vão sendo tiradas, numa variável B\$ (linha 100), suprimidas em A\$ e recolocadas em C\$, na ordem correcta (linhas 110-120).

A linha 140 controla o número de letras de B\$.

As impressões no écran são efectuadas pelas linhas 160, 170 e 180.

A viragem é feita, letra por letra, no ciclo I (210-230); o resultado é colocado em B\$ na linha 240, apresentado no écran pela linha 250 e comparado com o resultado final na linha 260.

O número de tentativas será impresso no écran no fim da partida, graças à linha 270.

PARA JOGAR:

Depois de ter teclado RUN, obtém-se a apresentação no écran da pergunta NUMERO DE LETRAS (DE 1 A 12)?; responde-se teclando N, seguido de ENTER. A impressão seguinte apresenta a ordem inicial (INICIO) e a ordem a obter (FIM).

Joga-se indicando o número de letras a permutar a partir da esquerda; por exemplo: se o texto apresentado fosse GEABDFC, teclando o 3 e ENTER obter-se-ia AEGBDFC.

Prosegue-se até à obtenção da ordem desejada.

A pontuação será impressa no fim da partida, assim como a pergunta OUTRO JOGO (S/N)?; responder S se se deseja fazer outro jogo.

EXTENSÕES POSSÍVEIS:

Poder-se-á utilizar o mesmo tipo de jogo para, por exemplo, reconstituir um desenho escondido, cujas diversas partes sejam referenciadas por letras. Pode-se também realizar permutas mais complexas (por exemplo, o facto de permutar duas letras faz permutar as duas vizinhas da letra mais à esquerda, etc.).

```
10 REM PERMUTAS
20 RANDOMIZE(TIME)
30 CLS
40 INPUT "NUMERO DE LETRAS (DE 1 A 12)? ";N
50 IF N<2 OR N>12 THEN 40
60 A$="ABCDEFGHJKLMN":C$=STRING$(12," ");
B$=""
70 CLS
80 X=INT(RND*N+1)
90 IF MID$(A$,X,1)=" " THEN 80
100 B$=MID$(A$,X,1)+B$
110 MID$(C$,X,1)=MID$(A$,X,1)
120 MID$(A$,X,1)=" "
130 S=1
140 IF LEN(B$)<>N THEN 80
150 C$=LEFT$(C$,LEN(B$))
160 LOCATE 1,5:PRINT "INICIO   :";:PEN 2
:PRINT B$:PEN 1
170 LOCATE 1,7:PRINT "FIM     :";:PEN 3
:PRINT C$:PEN 1
180 LOCATE 1,11:PRINT "PERMUTA NUMERO";S
:PRINT:PRINT "QUANTIDADE? ";CHR$(18);:IN
PUT " ",L:IF L>N OR L<1 THEN 180
190 S=S+1
200 A$=""
210 FOR I=1 TO L
220 A$=A$+MID$(B$,L-I+1,1)
230 NEXT I
240 B$=A$+RIGHT$(B$,LEN(B$)-L)
250 SOUND 1,478:SOUND 1,426:SOUND 1,379:
LOCATE 1,17:PRINT "RESULTADO :";:PAPER 2
:PEN 3:PRINT B$:PAPER 0:PEN 1
260 IF B$<>C$ THEN 180
270 SOUND 1,478:SOUND 1,319:SOUND 1,379:
PRINT:PRINT "CONSEGUIDO EM";S-1;"TENTATI
VAS"
280 INPUT "OUTRO JOGO (S/N)? ";R$
290 IF LEFT$(R$,1)="S" THEN 30
```

Bridge

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: reflexão.

Dificuldade do jogo: variável, consoante o problema posto.

O JOGO:

Não encontrará, neste programa, um parceiro para os seus jogos de bridge, mas sim, mais exactamente, uma simulação para a apresentação de uma mão; esta apresentação é realizada jogo a jogo. Pode-se, a partir daí, pôr todas as espécies de problemas: declarações, jogar cartas, etc.

O PROGRAMA:

As linhas 50 e 60 definem um jogo de 52 cartas.

O quadro B\$, na linha 70, permitirá a apresentação no écran dos símbolos das cores e também melhorar a designação de certas cartas (10, valete, dama, rei e ás).

O ciclo I (80-240) realiza a tiragem de quatro jogos de 13 cartas.

O ciclo J (90-210) realiza tiragem e impressão.

A carta é escolhida, entre as 52 disponíveis, pelo ciclo WHILE/WEND (110-130) inicializado em 100.

A linha 140 calcula a cor da carta tirada.

As linhas 150 e 160 ocupam-se da impressão complementar no caso de se tirar um 10, um valete, uma dama, um rei ou um ás.

A impressão é realizada pelas linhas 170 a 190.

A linha 200 permite eliminar as cartas já tiradas, e a linha 220 realiza uma paragem entre cada apresentação.

PARA JOGAR:

Faz-se correr o programa com RUN.

Uma primeira série de 13 cartas é impressa no écran, verticalmente. Bastará carregar em qualquer tecla para obter a segunda série de 13 cartas. Proceder-se-á do mesmo modo para a terceira e para a quarta séries.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em representar as 13 cartas sob uma forma mais convencional.

Pode-se também acrescentar a possibilidade de fazer declarações (e compará-las com as do AMSTRAD, por exemplo), ou introduzir possibilidades de jogar cartas, mesmo para o AMSTRAD.

```
10 REM BRIDGE
20 RANDOMIZE(TIME)
30 INK 2,0:INK 3,6
40 CLS
50 R$=234567891VDRA"
60 R$=R$+R$+R$+"0"
70 B$=CHR$(226)+CHR$(227)+CHR$(228)+CHR$(
(229)+"0AAES"
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=53
110 WHILE MID$(R$,X,1)="0"
120 X=INT(RND*LEN(R$)+1)
130 WEND
140 S=INT((X-1)/13)+1
150 T=X MOD 13
160 T=T-13*(T=0)
170 LOCATE 10*J-9,I+1:PRINT MID$(R$,X,1)
;
180 IF T>=9 THEN PRINT MID$(B$,T-4,1)
190 LOCATE 10*J-5,I+1:FEN 2-(S=2 OR S=3)
:PRINT MID$(B$,S,1):FEN 1
200 MID$(R$,X,1)="0"
210 NEXT I
220 LOCATE 4,23:INPUT "CARREGUE EM <ENTE
R> PARA CONTINUAR";S$
230 LOCATE 4,23:PRINT CHR$(20);
240 NEXT J
```

Hanói

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: reflexão.

Dificuldade do jogo: difícil.

O JOGO:

Este jogo é aqui apresentado numa versão para um jogador. Dispõe-se de um certo número de discos (sete no máximo) e deve-se transferi-los da torre número 1 para a torre número 3, respeitando uma única regra: em momento algum um disco pode ser coberto por um disco de diâmetro superior.

O número de jogadas não é contabilizado por este programa mas, em contrapartida, as jogadas ilícitas serão recusadas.

O PROGRAMA:

A variável D\$, na linha 40, servirá para desenhar os discos e as bases.

O número de discos é feito entrar na linha 50 e recusado se tiver mais de sete discos.

O ciclo I (70-90) permite fabricar uma variável A\$(1), que contém o estado da primeira pilha sob forma numérica (na linha 30 são previstas três variáveis para conservar o estado das três pilhas).

As bases são desenhadas pela linha 110 e as três pilhas pelos ciclos encaixados I (130-180) e J (140-170).

O controlo da jogada é efectuado pela linha 270 e o estado das pilhas é modificado pelas linhas 280 e 290.

PARA JOGAR:

Depois de se ter feito correr o programa com RUN, em resposta ao pedido que é impresso no écran faz-se entrar o número de discos, seguido de ENTER. Obtém-se o desenho do estado inicial e, à pergunta TORRE DE PARTIDA?, responde-se dando o número da torre que escolhemos. Este número (1, 2 ou 3) será recusado se a torre estiver vazia. Depois teclia-se ENTER e, à pergunta TORRE DE CHEGADA?, responde-se com o número da torre de chegada que escolhemos.

Se a jogada não é possível, ser-se-á obrigado a recomeçar na TORRE DE PARTIDA. Se a jogada for possível, a deslocação é realizada e recomeça-se.

EXTENSÕES POSSÍVEIS:

Poder-se-á desenhar discos de diferentes cores, introduzir um contador de jogadas e, sobretudo, definir um jogo para o computador.

Pode-se também tentar complicar o jogo com mais duas pilhas e dois jogadores, cada um deles dispondo de discos de cor diferente, por exemplo.

```
10 REM HANOI
20 CLS
30 DIM A(3):A(1)=9:A(2)=9:A(3)=9
40 D$="":D$=STRING$(9,233)
50 INPUT "NUMERO DE DISCOS (MENOS DE 8)?
";N
60 IF N<2 OR N>7 THEN 50
70 FOR I=0 TO N-1
80 A(1)=VAL(STR$(A(1))+STR$(N-I))
90 NEXT I
100 CLS
110 LOCATE 1,15:PEN 5:PRINT D$:LOCATE 13
,15:PRINT D$:LOCATE 25,15:PRINT D$
120 LOCATE 5,15:PAPER 2:PRINT "1":LOCATE
17,15:PRINT "2":LOCATE 29,15:PRINT "3":
PAPER 0:PEN 3
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B$=RIGHT$(STR$(A(J)),LEN(STR$(A(J)))
-1)
160 IF LEN(B$)>I+1 THEN LOCATE 12*J-10,1
4-I:PRINT LEFT$(D$,VAL(MID$(B$,I+2,1)))
170 NEXT J
180 NEXT I
190 PEN 1
200 LOCATE 1,18:PRINT CHR$(18);:INPUT "T
ORRE DE PARTIDA? ";D
210 IF D>3 OR D<1 THEN 200
220 IF A(D)=9 THEN 200
230 LOCATE 1,20:PRINT CHR$(18);:INPUT "T
ORRE DE CHEGADA? ";A
240 IF A=D THEN 200 ELSE IF (A<1 OR A>3)
THEN 230
250 B$=RIGHT$(STR$(A(D)),LEN(STR$(A(D)))
-1)
260 C$=RIGHT$(STR$(A(A)),LEN(STR$(A(A)))
-1)
270 IF RIGHT$(B$,1)>RIGHT$(C$,1) THEN 20
0
280 A(A)=VAL(C$+RIGHT$(B$,1))
290 A(D)=VAL(LEFT$(B$,LEN(B$)-1))
300 SOUND 1,478:SOUND 1,426:SOUND 1,379:
GOTO 100
```

Corrida de cavalos

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: azar (vários jogadores).

Dificuldade do jogo: nenhuma.

O JOGO:

Far-se-á entrar um número de cavalos compreendido entre 1 e 9, e a corrida começará imediatamente.

Poderá haver chegadas *ex aequo*, caso em que se deve ter em conta que o AMSTRAD se ocupa de fazer avançar os cavalos uns após outros, começando pelo número 1; a classificação será impressa no écran à medida que os cavalos vão chegando (e tendo em conta os *ex aequo*.)

O PROGRAMA:

Um primeiro ciclo não formal (80-170) imprime os cavalos na sua posição de partida.

Um segundo ciclo, I (100-180), fá-los-á avançar; a posição de cada cavalo será referenciada por um número ou por uma letra numa variável alfanumérica A\$.

A linha 120 fará avançar os cavalos.

A linha 140 faz a gestão das chegadas.

A linha 150 realizará as impressões.

A rotina 240-280 faz avançar o contador das chegadas e imprime os resultados.

PARA JOGAR:

Faz-se correr o programa com RUN.

À pergunta NUMERO DE CAVALOS?, responde-se com um inteiro de 1 a 9, seguido de ENTER.

Em seguida a corrida desenrolar-se-á automaticamente.

EXTENSÕES POSSÍVEIS:

Poder-se-á desenhar os cavalos com um grafismo mais cuidado, e diferenciá-los por números.

Poder-se-á também imaginar todo um sistema de apostas, a realizar antes do início de cada corrida.

```

10 REM CORRIDA DE CAVALOS
20 RANDOMIZE(TIME)
30 INK 0,9:INK 2,0,26
40 CLS
50 INPUT "NUMERO DE CAVALOS? ";N:IF N<0
OR N>9 THEN 50
60 A$="":P=0:A=0:T=1:K=0
70 J=T
80 A#=A#+ "0":I=J
90 GOTO 150
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="K" THEN 180
120 B#=CHR$(ASC(MID$(A$,I,1))+INT(RND*4+
1))
130 A#=LEFT$(A$,I-1)+B#+RIGHT$(A$,LEN(A$
)-I)
140 IF B#>="K" THEN GOSUB 240
150 LOCATE ASC(MID$(A$,I,1))-46,2*I+3:PR
INT " ";:PAPER 3:PRINT I:PAPER 0
160 SOUND 1,159,1,4,0,0,1:FOR D=1 TO 60:
NEXT D
170 IF K=0 THEN J=J+1:IF J<=N THEN 80 EL
SE K=1:GOTO 100
180 NEXT I
190 P=P+T
200 T=0
210 IF A<N THEN 100
220 SOUND 1,478:SOUND 1,319:SOUND 1,379
230 LOCATE 1,24:END
240 A#=LEFT$(A$,I-1)+"K"+RIGHT$(A$,LEN(A
$)-1)
250 LOCATE 1,2*I+3:PEN 2:PRINT P:PEN 1
260 T=1:SOUND 1,239,2
270 A=A+T
280 RETURN

```

Engatem os vagões

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: solitário.

Dificuldade do jogo: fácil.

O JOGO:

É necessário engatar, o mais rapidamente possível, a locomotiva ao seu vagão, que aparecem ambos, no início da partida, em dois pontos do écran mais ou menos afastados. Para isso deverá deslocar-se a locomotiva indicando a sequência dos movimentos a efectuar (sob uma forma codificada: A para uma casa para o alto; B para uma casa para baixo; D para uma casa para a direita; E para uma casa para a esquerda).

Teclar-se-á, por exemplo, BBEDAAA.

O PROGRAMA:

As linhas 40 a 70 desenham a locomotiva e o vagão.

A posição inicial da locomotiva é definida pelas linhas 90 e 100, a do vagão pelas linhas 110 e 120.

A linha 130 evita que essas posições sejam inicialmente muito próximas uma da outra.

As linhas 150 a 190 realizam a impressão do vagão e da locomotiva.

O teste de fim de partida é feito na linha 200.

A utilização da janela 1 permite proteger o écran contra um *scrolling* (rolamento do écran) provocado pela impressão no écran de uma cadeia R\$ (linhas 25 e 210) demasiado longa.

Os movimentos a efectuar são feitos entrar numa variável alfanumérica R\$, na linha 210, e são descodificados num ciclo I (240-280), para serem executados na linha 270.

PARA JOGAR:

A ordem RUN permite imprimir no écran as posições de partida da locomotiva e do vagão.

Tecléia-se então a série de deslocações a efectuar, sob a forma DDDAAABAEE, seguida de ENTER.

A locomotiva será então apresentada na sua posição, e recomeça-se até que ela esteja engatada ao vagão.

Obter-se-á então a impressão da pontuação, no écran.

EXTENSÕES POSSÍVEIS:

Será interessante prever a perda da partida se, por acidente, no decorrer da deslocação, a locomotiva colidir com o vagão.

Poder-se-á também prever vários vagões e uma ordem de engate, jogar com vários jogadores com várias locomotivas, desenhar vias férreas, introduzir obstáculos a contornar obrigatoriamente por um lado determinado, etc.

```
10 REM ENGATEM OS VAGÕES
20 RANDOMIZE(TIME)
25 WINDOW #1,1,40,24,25
30 DEFINT L,W
40 D$=CHR$(139)+CHR$(140)+CHR$(141)
50 A$=CHR$(142)+CHR$(143)+CHR$(143)
60 B$=CHR$(141)+CHR$(140)+CHR$(142)
70 C$="O O"
80 N=0
90 LY=1+RND*32
100 LX=2+RND*19
110 WY=1+RND*32
120 WX=2+RND*19
130 IF ABS(LX-WX)<4 THEN 90
140 CLS:PRINT CHR$(22)+CHR$(1)
150 PEN 2:LOCATE WY,WX:PRINT B$
160 LOCATE WY,WX+1:PRINT C$
170 PEN 3:LOCATE LY,LX-1:PRINT D$
180 LOCATE LY,LX:PRINT A$
190 LOCATE LY,LX+1:PRINT C$
200 IF LX=WX AND WY=LY-3 THEN 300
210 CLS #1:INPUT #1,"CAMINHO A SEGUIR (D
EEABB...)? ";R$:R$=UPPER$(R$)
220 X=0:Y=0
230 N=N+1
240 FOR I=1 TO LEN(R$)
250 X--(MID$(R$,I,1)="B" AND LX<22)+(MID
$(R$,I,1)="A" AND LX>2)
260 Y--(MID$(R$,I,1)="D" AND LY<38)+(MID
$(R$,I,1)="E" AND LY>1)
270 LX=LX+X:LY=LY+Y
280 NEXT I
290 GOTO 140
300 CLS #1:PRINT #1,TAB(9) "BRAVO! CONSE
GUIU EM";N;"JOGADAS"
310 SOUND 1,478:SDUND 1,319:SOUND 1,379
```

O lobo e os cordeiros

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: reflexão (dois jogadores).

Dificuldade do jogo: média.

O JOGO:

Trata-se de um jogo bastante conhecido que pode ser praticado indiferentemente num tabuleiro de xadrez de 8 por 8 ou de 10 por 10. É aqui apresentado numa versão para um tabuleiro de 8 por 8 casas; um dos jogadores escolhe ser o lobo, que se desloca casa a casa em cada jogada, segundo as diagonais, permanecendo nas casas pretas, e que pode avançar ou recuar. Os cordeiros deslocam-se igualmente sobre as casas pretas, em diagonal, mas não têm o direito de recuar. As casas já ocupadas são interditas (não se pode tomar peças) e não há saltos.

O lobo ganha se consegue chegar à linha de base antes de todos os cordeiros terem ocupado a linha do topo; os cordeiros ganham neste último caso, ou se conseguirem bloquear o lobo.

O PROGRAMA:

Uma janela com o número 1, definida na linha 20, permite proteger a apresentação no écran.

Uma variável A\$, na linha 40, contém as posições de partida.

O ciclo I (80-170) desenha o tabuleiro, e as linhas 190 a 220 posicionam o lobo e os cordeiros.

A linha 230 faz a gestão da mudança de jogador.

A linha 250 imprime no écran JOGO DO LOBO, se é a vez do jogador que manipula o lobo, e JOGO DOS CORDEIROS no caso contrário.

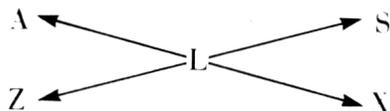
As linhas 280 a 300 ou 310 a 320 analisam a resposta, em função do respectivo jogador.

PARA JOGAR:

RUN inicia o jogo, que fica parado aguardando uma entrada: 0 e ENTER, se queremos que sejam os cordeiros a começar, ou 1 e ENTER, se quisermos que seja o lobo. Depois disto é apresentado o tabuleiro e é impressa a frase JOGO DO LOBO (A, S, Z, X), para o lobo, ou JOGO DOS CORDEIROS (ND e NE), para os cordeiros.

Os cordeiros respondem com um número, seguido de D ou E (para direita ou esquerda). Assim, 2D provoca a deslocação do cordeiro

número 2, que sobe em diagonal uma casa para a direita. A deslocação é efectuada, após o que se obtém uma pergunta idêntica para o lobo, que responderá teclando A, S, Z ou X, consoante a direcção escolhida:



Deverá ter-se o cuidado de não sair do tabuleiro e de não cair em cima do adversário, porque não estão previstos controlos para esses casos.

EXTENSÕES POSSÍVEIS:

Poder-se-á prever uma referenciação das casas, um grafismo mais cuidado, controlos para não ultrapassar os limites do tabuleiro, para não cair numa casa já ocupada...

Poder-se-á também pensar em jogar num tabuleiro de 10 por 10 casas, o que aumenta ligeiramente, para os cordeiros, a dificuldade do jogo.

Mas a extensão que tem mais interesse executar será certamente a realização de um programa que permita ao AMSTRAD comportar-se como um adversário à altura, quer como lobo quer como cordeiros (e, neste caso, será possível definir uma estratégia que permita que o AMSTRAD consiga ganhar).

```

10 REM O LOBO E OS CORDEIROS
20 WINDOW #1,28,40,9,15
30 INPUT "CORDEIROS (O) OU LOBO (1)?";S
40 A$="8183858714"
50 CLS
60 LOCATE 1,25:PRINT "  A B C D E
F G H";
70 FOR I=1 TO 8:LOCATE 1,3*I-1:PRINT I:N
EXT I
80 FOR I=1 TO 8
90 FOR J=1 TO 8
100 LOCATE 3*J,3*I-2
110 IF (I+J) MOD 2=0 THEN PAPER 2:ELSE P
APER 3
120 FOR N=1 TO 3
130 PRINT STRING$(3,233);CHR$(10);STRING
$(3,8);
140 NEXT N
150 PAPER 0
  
```

```

160 NEXT J
170 NEXT I
180 PEN 3
190 FOR I=1 TO 5
200 LOCATE 3*VAL(MID$(A$,2*I,1))+1,3*VAL
(MID$(A$,2*I-1,1))-1
210 IF I<>5 THEN PRINT RIGHT$(STR$(I),1)
;:ELSE PRINT "L";
220 NEXT I
230 PEN 1:CLS #1:S=ABS(S-1)
240 PRINT #1,"      JOGO"
250 IF S=0 THEN PRINT #1," DO LOBO":PRI
NT #1," (A,S,Z,X)" ELSE PRINT #1,"DOS CO
RDEIROS":PRINT #1," ND DU NE"
260 LOCATE #1,6,5:INPUT #1,"",R$:IF S=1
THEN GOTO 310
270 C=10
280 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")
+(C-1 AND R$="Z")
290 A=A-(C+1 AND R$="A")-(C-1 AND R$="S"
)
300 GOTO 330
310 C=2*VAL(LEFT$(R$,1))
320 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(
R$,1)="D")-(11 AND RIGHT$(R$,1)="E")
330 MID$(A$,C-1,2)=RIGHT$(STR$(A),2)
340 GOTO 80

```

Nível 4

<i>Nome do programa</i>	<i>Temas e particularidades</i>
Tiragem de dados	Suficientemente flexível para ser utilizado onde for necessário.
A bomba	Poderá descobri-la a tempo? (Um jogador.)
Chapéus	Seja bom observador e terá êxito (dois jogadores).
Desenho de dados	Melhorar os seus jogos de dados incorporando este programa.
O macaco	Um jogo de letras para dois jogadores, em que é necessário ter bons reflexos.
O jogo do 15	Um jogo de reflexão utilizando números.
Teclado	Aprenda a escrever à máquina, divertindo-se ao mesmo tempo.
Destreza	Jogo de reflexos para um jogador, muito visual.
Anagramas	Um bom jogo de reflexos, para dois jogadores.
«Zombies»	Um pouco fantástico, mas muito visual (um jogador).
Quadro mágico	Prepare as suas criações gráficas, sem papel nem lápis.
«Decómaco»	Para lhe evitar uma grande «seca» no jogo «Nicomaco».
Evasão	Precisará de muita sorte para conseguir.
Repetições	Descobrir uma sequência de números cada vez mais longa.
Mateus	Não sabe desenhar? Deixe que o seu computador o faça.
Através	Um jogo de estratégia para dois jogadores.
Calculadora	Um programa de simulação muito realista.
Caça ao submarino	Precisará de bastante reflexão e de muita sorte.

As técnicas de animação elementar

DESCOBRIR UM CARÁCTER TECLADO: INKEY\$

O princípio do jogo gráfico, em animação, é deixar o jogo desenrolar-se sozinho, de certa maneira, durante certo tempo, enquanto não houver qualquer intervenção do jogador.

Logo, é preciso dispor da possibilidade de observar se houve ou não intervenção do jogador, sem com isso interromper o desenrolar do programa.

No Basic AMSTRAD, esta propriedade é oferecida pela **função INKEY\$**, que efectua uma perscrutação das teclas do teclado e regista o nome da última tecla usada.

INKEY\$ poderá ser utilizada de duas maneiras:

- Num ciclo (realização de um movimento). Toma-se INKEY\$ e faz-se um teste para uma eventual saída do ciclo.
- Numa pausa: enquanto INKEY\$ estiver vazia (não foi usada qualquer tecla) mantém-se a pausa, que acaba quando for premida uma tecla.

Por vezes será necessário premir várias vezes a tecla para ser tomada em conta.

DESLOCAÇÃO DE OBJECTOS DE BAIXO PARA CIMA NO ÉCRAN: «SCROLL»

É difícil, trabalhando em Basic, gerar a deslocação simultânea de vários objectos no écran (acaba-se por se obter um movimento aos sacões, devido a um proibitivo tempo de deslocação).

O AMSTRAD realiza automaticamente um «**scrolling**» (quer dizer uma deslocação do conjunto da imagem) para cima, quando se escreve na última linha da parte inferior do écran.

Esta propriedade foi aproveitada em diversos jogos. Com efeito, ela permite deslocar uma linha para cima os diversos objectos (textos ou gráficos) presentes no écran num dado momento. A linha do topo, evidentemente, perde-se.

Se um objecto tem de permanecer imóvel no écran, será necessário apagá-lo na sua nova posição e recolocá-lo na posição inicial.

Tiragem de dados

REFERÊNCIAS

Dificuldade de programação: nenhuma.
Categoria de jogo: lançamento de dados.
Dificuldade do jogo: nenhuma.

O JOGO:

O programa proposto não realiza um jogo propriamente dito, mas poderá ser utilizado como rotina em numerosos jogos.

Oferece simplesmente a possibilidade de lançar de 1 a 9 dados, pela simples pressão de uma tecla. Os dados são, seguidamente, apresentados no écran com um grafismo muito simplificado.

O PROGRAMA:

O ciclo WHILE/WEND (60-80) toma o carácter premido no teclado e a linha 100 transforma-o em número (se isso não for possível, o programa recusa a entrada).

Uma variável D, não dimensionada, recebe o valor indicado por cada dado (linha 150).

Um ciclo, J (120-140), é utilizado para desenhar um fundo e um outro ciclo, I (110-170), serve para a tiragem (tiragem na linha 150 e impressão na linha 160).

A linha 190 oferece a possibilidade de refazer uma tiragem; não é indispensável.

PARA JOGAR:

Faz-se correr o programa com a ordem RUN e obtém-se a impressão no écran da mensagem NUMERO DE DADOS A LANCAR?, a que se responde tecendo um algarismo.

O resultado é então automaticamente apresentado no écran; basta premir qualquer tecla para recomeçar o programa.

Para parar é necessário premir ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á, evidentemente, utilizar a base fornecida por este programa para a realização de qualquer jogo que use dados («421», «Yam's», «Craps», etc.).

```

10 REM TIRAGEM DE DADOS
20 RANDOMIZE(TIME)
30 INK 0,9:INK 1,26:INK 2,0
40 CLS
50 LOCATE 1,1:PRINT "NUMERO DE DADOS A L
ANCAR?";A$=""
60 WHILE A$="" OR A$<"1" OR A$>"9"
70 A$=INKEY$
80 WEND
90 PRINT CHR$(20);:PAPER 1:PEN 3:PRINT A
$:PAPER 3:PEN 2
100 N=VAL(A$)
110 FOR I=1 TO N
120 FOR J=1 TO 3
130 LOCATE 4*I-3,8+J:PRINT "  "
140 NEXT J
150 D(I)=1+INT(RND*6)
160 LOCATE 4*I-2,10:PRINT RIGHT$(STR$(D(
I)),1)
170 NEXT I
180 PAPER 0:PEN 1
190 GOTO 50

```

A bomba

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: solitário.

Dificuldade do jogo: média.

O JOGO:

Deverá descobrir, com a ajuda de um detector especializado, um caixote radioactivo armadilhado, colocado num quarto de hotel por um viajante mal-intencionado. Apenas um problema, mas de vulto: o hotel tem um milhão de quartos (100*100*100).

O seu detector apresenta no écran, permanentemente, um sinal que o informa da proximidade da bomba, e o tempo que lhe resta é também permanentemente apresentado no écran, decrescendo de 10 em 10.

Poderá descobrir a bomba antes da explosão?

O PROGRAMA:

As janelas 1 e 2, definidas nas linhas 40 e 50, servirão para apresentar o sinal e o tempo.

Uma variável numérica R, não dimensionada, contém as três coordenadas que definem a posição do jogador.

A posição do quarto que contém o caixote armadilhado é definida por uma tiragem aleatória de três variáveis, A, B e C, na linha 70.

O ciclo I (90-110) desenha o detector.

O sinal de proximidade é calculado na linha 130 e a linha 140 permite a apresentação das coordenadas do quarto a descobrir, no caso de se ter perdido o jogo.

A linha 150 permite a apresentação simultânea do sinal e do tempo restante a decorrer.

A linha 160 provoca cada vez mais frequentemente a chamada à rotina de som na linha 350.

A linha 170 desconta o tempo.

A linha 180 controla se o caixote foi ou não encontrado.

O ciclo I (220-260) permite fazer entrar novas coordenadas, utilizando a rotina colocada nas linhas 290 a 340.

PARA JOGAR:

Depois de se fazer correr o programa com a ordem RUN, são impressos no écran o sinal e o tempo (em princípio 200).

O jogador está então na posição $x = 0$, $y = 0$, $z = 0$, e dever-se-á introduzir sucessivamente as coordenadas x , y , z do quarto a explorar. Far-se-á, por exemplo, 13 e depois ENTER, 45 e depois ENTER, etc.

O novo sinal é então apresentado, e o tempo é decrementado de 10.

Continuar-se-á assim, quer até ao bom resultado (paragem do jogo e fim do programa), quer até à aparição de três números (x , y , z) sem saída do programa, o que significa que se perdeu a partida.

EXTENSÕES POSSÍVEIS:

Será sem duvida indispensável melhorar o grafismo, acrescentar uma explosão (sonora e visual) para simbolizar o rebentamento da bomba, apresentar simbolicamente a posição do hotel, etc.

```
10 REM A BOMBA
20 RANDOMIZE(TIME)
30 DEF FN SS*(X)=RIGHT*(STR*(X),LEN(STR*(X))-1)
40 WINDOW #1,13,19,7,7:WINDOW #2,22,24,7,7
50 PAPER #1,1:PEN #1,3:PAPER #2,2:PEN #2,0
60 L=100:T=200
70 A=INT(RND*L):B=INT(RND*L):C=INT(RND*L)
80 CLS:PEN 3
90 FOR I=1 TO 3
100 LOCATE 11,5+I:PRINT STRING*(16,233)
110 NEXT I
120 PEN 1
130 S=ABS((A/L+B*C*L)-(R(1)/L+R(2)+R(3)*L)):S=ROUND(S,2)
140 IF T=0 THEN PRINT "PERDEU. A SOLUCAO
: ";A;B;C:SOUND 1,253:SOUND 1,323:SOUND
1,478,20:END
150 CLS #1:PRINT #1,FN SS*(S);:CLS #2:PR
INT #2,FN SS*(T);
160 EVERY INT(T/2) GOSUB 350
170 T=T-INT(SQR(L))
180 IF S=0 THEN LOCATE 1,21:PAPER 2:PEN
3:PRINT "BRAVO":SOUND 1,478:SOUND 1,319:
SOUND 1,379:END
190 LOCATE 1,14:PRINT " - X: "
200 LOCATE 1,16:PRINT " - Y: "
210 LOCATE 1,18:PRINT " - Z: "
220 FOR I=1 TO 3
230 R$=""
240 GOSUB 280
250 R(I)=VAL(R#)
260 NEXT I
270 GOTO 130
280 LOCATE 7,2*I+12:PRINT CHR*(243);CHR*
(18);
290 WHILE A$<>CHR*(13)
300 A$=INKEY$
310 PRINT A$;
320 R$=R$+A$
330 WEND
340 A$="" :LOCATE 7,2*I+12:PRINT " ":RETU
RN
350 SOUND 2,159,1:RETURN
```

Chapéus

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexos, para dois jogadores.

Dificuldade do jogo: fácil.

O JOGO:

A ideia é muito simples: duas personagens deslocam-se no écran, uma em direcção à outra, e cruzam-se. No momento em que se cruzam é possível que troquem de chapéus. Trata-se de ser o primeiro a dizer se houve ou não troca de chapéus aquando do cruzamento.

O PROGRAMA:

A linha 30 contém o desenho dos dois chapéus [nas duas variáveis alfanuméricas C\$(1) e C\$(2)].

A linha 40 contém os símbolos que permitirão desenhar as personagens, sob a forma do seu código ASCII de três algarismos.

A linha 60 permite a troca dos chapéus, se ocorrer.

A variável T da linha 70 serve para referenciar se houve ou não troca de chapéus.

O desenho das duas personagens é realizado pelo ciclo L (90-80). O ciclo encaixado I (120-170) realiza o desenho linha por linha, e o ciclo N (140-160) assegura o desenho de cada linha.

A linha 190 permite a apresentação simultânea dos dois chapéus.

A sua deslocação é realizada pela linha 200.

O teste que permite ou não a mudança dos chapéus é efectuado na linha 220.

As linhas 250 a 310 determinam se o primeiro jogador que premiu uma tecla ganhou ou perdeu.

PARA JOGAR:

Fazer correr o programa com RUN.

As personagens começam imediatamente a deslocar-se uma em direcção à outra.

O jogador da esquerda dispõe das teclas 1 e 2 e o da direita das teclas 9 e 0.

Se houve troca de chapéus, as teclas a premir serão 1 (à esquerda) e 0 (à direita).

Se não houve troca de chapéus será necessário, para ganhar, ser o primeiro a carregar no 2 (à esquerda) e no 9 (à direita).

A mensagem impressa no écran indicará então BRAVO, ESQUERDA ou BRAVO, DIREITA, ou, ainda, PERDEU, ESQUERDA ou PERDEU, DIREITA.

EXTENSÕES POSSÍVEIS:

Será possível prever apresentação a cores, com mais complexa animação das personagens.

Poder-se-á também complicar o jogo a nosso bel-prazer com vários objectos trocados de modo aleatório (será necessário, nesse caso, encontrar a lista completa deles).

```
10 REM CHAPEUS
20 RANDOMIZE(TIME):PRINT CHR$(22);CHR$(1
)
30 C$(1)=CHR$(222)+CHR$(207)+CHR$(223):C
HR$(2)=CHR$(214)+CHR$(203)+CHR$(215)
40 A$="032224032150159156149149149155149
155150155156149032149149032149153032147"
50 X(1)=1:X(2)=33
60 D#=C$(1):C$(1)=C$(2):C$(2)=D#
70 T=ABS(T-1)
80 CLS:INK 1,1
90 FOR L=1 TO 2
100 PEN 1:LOCATE X(L),2:PRINT C$(L)
110 PEN 1+L
120 FOR I=1 TO 8
130 LOCATE X(L),I+2
140 FOR N=0 TO 2
150 PRINT CHR$(VAL(MID$(A$,9*I-8+3*N,3)
));
160 NEXT N
170 NEXT I
180 NEXT L
190 INK 1,24
200 X(1)=X(1)+2:X(2)=X(2)-2:INK 1,24
210 TE=0:WHILE TE<200:TE=TE+1:WEND
220 IF X(1)=X(2) AND RND<0.5 THEN 60
230 IF X(2)=-1 THEN PRINT CHR$(7):;T=0:G
OTO 50
240 R$=INKEY$:IF R$="" THEN 80
250 PRINT CHR$(22);CHR$(0):;IF X(1)<=X(
2) THEN 280
260 PAPER 2:PEN 3:LOCATE 13,23:IF (R$="1
" AND T=0) OR (R$="2" AND T=1) THEN PRIN
T "BRAVO, ESQUERDA":SOUND 1,478:SOUND 1,
379:SOUND 1,319:PAPER 0:END
270 IF (R$="0" AND T=0) OR (R$="9" AND T
=1) THEN PRINT "BRAVO, DIREITA":SOUND 1,
478:SOUND 1,379:SOUND 1,319:PAPER 0:END
280 PAPER 3:PEN 1:LOCATE 14,23:IF R$="2"
OR R$="1" THEN PRINT "PERDEU, ESQUERDA"
290 IF R$="9" OR R$="0" THEN PRINT "PERD
EU, DIREITA"
300 FOR I=1 TO 200 STEP 2:SOUND 1,I,1:NE
XT I
310 PAPER 0
```

Desenho de dados

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: lançamento de dados.
Dificuldade do jogo: fácil.

O JOGO:

O programa proposto é interessante sobretudo pelo seu lado gráfico, porque permite representar no écran, de modo muito conveniente, de um a cinco dados, utilizando uma técnica de desenho um tanto particular.

Cada resposta à pergunta NUMERO DE DADOS? provoca imediatamente um novo lançamento do número de dados indicado.

O PROGRAMA:

A linha 70 permite controlar se houve ou não acção sobre uma tecla de 1 a 15.

As linhas 90 e 100 inicializam as duas variáveis de desenho dos dados.

No ciclo I (110-200), as linhas 120, 130 e 180 desenharam o quadro do dado.

A linha 140 realiza a tiragem, e o ciclo J (150-190) desenha os pontos relacionados com a tiragem realizada (chamando as variáveis A\$ e B\$).

PARA JOGAR:

Faz-se correr o programa com a ordem RUN, e à pergunta NUMERO DE DADOS? responde-se carregando numa tecla de 1 a 5. Os dados pedidos são então representados e volta a ser feita a pergunta NUMERO DE DADOS?.

Bastará teclar ESC duas vezes para sair do programa.

EXTENSÕES POSSÍVEIS:

Poder-se-á conservar a parte de desenho utilizada neste programa para realizar todas as espécies de jogos de dados, com uma representação gráfica de qualidade conveniente e fácil de modificar se houver necessidade.

```

10 REM DESENHO DE DADOS
20 RANDOMIZE(TIME)
30 INK 0,9:INK 1,26:INK 2,15:INK 3,0
40 CLS
50 PEN 1:LOCATE 1,15:PRINT "NUMERO DE DA
DOS? ";
60 A$=INKEY$:IF A$="" THEN 60
70 N=VAL(A$):IF N<1 OR N>5 THEN 40
80 CLS:PEN 3
90 A$=STRING$(7,32)+CHR$(231)+STRING$(2,
32)+CHR$(231)+STRING$(8,32)+CHR$(231)+CH
R$(231)+STRING$(3,32)+CHR$(231)
100 B$="121314324515525555"
110 FOR I=1 TO N
120 LOCATE 8*I-7,3:PRINT CHR$(150);STRIN
G$(5,154);CHR$(156)
130 LOCATE 8*I-7,9:PRINT CHR$(147);STRIN
G$(5,154);CHR$(153)
140 X=INT(RND*6+1)
150 FOR J=1 TO 5
160 IF J<=3 THEN S=5*VAL(MID$(B$,3*(X-1)
+J,1))-4
170 IF J<=3 THEN PEN 2:LOCATE 8*I-6,2*J+
2:PRINT MID$(A$,S,5):PEN 3
180 LOCATE 8*I-7,J+3:PRINT CHR$(149):LOC
ATE 8*I-1,J+3:PRINT CHR$(149)
190 NEXT J
200 NEXT I
210 GOTO 50

```

O macaco

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexos (dois jogadores).

Dificuldade do jogo: difícil.

O JOGO:

Inspira-se na já antiga ideia do macaco que, batendo ao acaso nas teclas de uma máquina de escrever, tem hipóteses não nulas de escrever um texto significativo.

O papel do macaco é aqui desempenhado pelo AMSTRAD, que vai imprimir no écran grupos de três a seis letras durante um tempo relativamente curto.

Se um dos jogadores calcula que pode conseguir fazer uma palavra com as letras propostas deve carregar rapidamente na sua tecla de jogo (1 ou 0).

A palavra deverá então ser aceite ou recusada pelo adversário, que responderá à pergunta CERTO OU ERRADO?

Cada palavra certa dá um ponto, cada palavra errada tira um ponto.

O PROGRAMA:

A linha 40 contém uma lista V\$ das vogais e uma lista C\$ das consoantes mais correntes.

Será possível modificar estas duas listas para aumentar, por exemplo, a frequência de aparição de certas letras (poder-se-á, assim, pôr dois E em V\$ ou dois R em C\$, etc.).

A linha 50 define o comprimento da palavra (em princípio, de três a seis letras, mas pode-se igualmente modificar este comprimento).

O ciclo I (80-130) permite a confecção de uma «palavra» tomando uma letra quer no alfabeto completo, quer na lista C\$ (linha 100), quer na lista V\$ (linha 110).

A linha 150 temporiza a aparição do resultado.

A linha 170 faz a gestão do fim da partida, com a impressão da pontuação.

A linha 180 verifica se um dos jogadores reagiu.

Por fim, as linhas 210 e 220 contabilizam os pontos de cada jogador.

PARA JOGAR:

Faz-se correr o programa teclando RUN.

A primeira série de letras é então apresentada no écran. O jogador da esquerda deverá teclar 1 se vê que pode realizar uma palavra, o jogador da direita deverá teclar 0 para o mesmo fim.

Se nenhum dos dois carregar na sua tecla, passado algum tempo será apresentada outra série de letras.

Se um dos jogadores usou a sua tecla aparece a pergunta (C)ERTO ou (E)RRADO?, impressa a azul para o jogador da esquerda e a encarnado para o da direita.

Responder-se-á C se a palavra estiver correcta ou E se estiver errada, seguido de ENTER.

Os pontos são então calculados pelo AMSTRAD e é apresentada a série de letras seguinte.

Para sair do programa basta teclar P em vez de 1 ou 0; a pontuação final é então apresentada.

EXTENSÕES POSSÍVEIS:

Poder-se-á modificar o comprimento das palavras, a frequência de aparição das letras, o desconto dos pontos, a regra do jogo, etc.

```
10 REM O MACACO
20 RANDOMIZE(TIME)
30 MODE 0
40 A=0:B=0:V$="AEIOU":C$="BCDGLMNPSTV"
50 X=INT(RND*4+3)
60 M$=""
70 CLS
80 FOR I=1 TO X
90 E$=CHR$(INT(RND*26+65))
100 IF RND<.4 THEN E$=MID$(C$,INT(LEN(C$)*RND+1),1)
110 IF RND<.5 THEN E$=MID$(V$,INT(LEN(V$)*RND+1),1)
120 M$=M$+E$
130 NEXT I
140 LOCATE (20-LEN(M$))/2,6:PEN 6:PAPER
17:PRINT M$:SOUND 1,478:SOUND 1,426:SOUND
1,379:PAPER 0:PEN 1
150 TE=0:WHILE TE<500:TE=TE+1:WEND
160 A$=INKEY$
170 IF A$="P" THEN MODE 1:LOCATE 4,12:PE
N 2:PRINT "PONTUACAO DE A: ";A:;PEN 3:PRI
NT "- PONTUACAO DE B: ";B:SOUND 1,478:SOU
ND 1,379:SOUND 1,319:TE=0:WHILE TE<1000:
TE=TE+1:WEND:END
180 IF A$<>"0" AND A$<>"1" THEN 50
190 LOCATE 2,18:PEN 3-VAL(A$):PRINT "(C)
ERTO OU (E)RRADO?"
200 R$=INKEY$:IF R$="" THEN 200
210 A=A-(R$="C" AND A$="1")+(R$="E" AND
A$="1")
220 B=B-(R$="C" AND A$="0")+(R$="E" AND
A$="0")
230 GOTO 50
```

O jogo do 15

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: reflexão (dois jogadores).
Dificuldade do jogo: média.

O JOGO:

É um jogo para dois jogadores, muito antigo, outrora praticado nas feiras.

Dispõe-se de um terreno de jogo de nove casas e cada jogador põe, à vez, um marcador com a sua cor numa casa disponível.

Ganhará o primeiro jogador que, com exactamente três marcadores, conseguir atingir um total de 15.

O PROGRAMA:

O ciclo I (50-80) desenha o campo de jogo, e as linhas 90 e 100 ocupam-se da gestão da vez dos jogadores (número 1 ou 2 por cada jogador, e marcador X ou O correspondente).

A linha 110 aguarda o lance.

O ciclo I (120-140) verifica a validade do lance.

A linha 150 testa o fim da partida, em caso de partida nula, e a linha 160 recusa os lances que não estejam de acordo com as regras (com retorno à situação inicial, linha 90).

A linha 170 ocupa-se da impressão do lance, e a linha 180 regista-o numa variável R\$.

A parte mais interessante do programa, que poderia servir para definir uma estratégia da máquina, vai da linha 200 à linha 290; são utilizados três ciclos encaixados para verificar se um dos jogadores obteve, com três marcadores, um total de 15

A mudança de jogador é realizada na linha 300.

PARA JOGAR:

Faz-se correr o programa teclando RUN.

O campo de jogo é impresso no écran e depois surge a pergunta JOGADOR 1: X? e o jogador que escolheu os X deverá fornecer o número da casa onde deseja jogar, se ela estiver livre, e o seu lance será registado.

Obtém-se, em seguida, a impressão da pergunta JOGADOR 2: O? e o segundo jogador procede do mesmo modo.

Se um dos jogadores ganha (totaliza 15 com três marcadores) obtém a mensagem BRAVO e a partida pára.

Em caso de partida nula, a máquina imprime no écran a mensagem ACABOU.

Far-se-á outra partida teclando RUN.

EXTENSÕES POSSÍVEIS:

O mais interessante será sem dúvida tentar definir uma estratégia para a máquina (tentando evitar que a máquina não jogue sempre o 5 quando é a sua vez de começar).

Pode-se também, evidentemente, melhorar o grafismo e introduzir efeitos sonoros para as diferentes mensagens.

```
10 REM O JOGO DO 15
20 CLS
30 F=0:R$="123456789"
40 O$="":X$=""
50 FOR I=0 TO 24 STEP 3
60 LOCATE I+6,1:PRINT CHR$(150)+CHR$(154
)+CHR$(156):LOCATE I+6,2:PRINT CHR$(149)
+" "+CHR$(149):LOCATE I+6,3:PRINT CHR$(1
47)+CHR$(154)+CHR$(153)
70 LOCATE I+6,4:PRINT (I+3)/3
80 NEXT I
90 LOCATE 1,10:PEN 2+F:PRINT "JOGADOR";F
+1;
100 PRINT ":"+CHR$(203+F*27)
110 A$=INKEY$:IF A$<"1" OR A$>"9" THEN 1
10
120 FOR I=1 TO LEN(R$)
130 IF A$=MID$(R$,I,1) THEN GOSUB 170
140 NEXT I
150 IF VAL(R$)=0 THEN LOCATE 15,20:PAPER
3:PEN 1:PRINT "ACABOU":TE=0:WHILE TE<20
0:SOUND 1,TE,1:TE=TE+1:WEND:END
160 GOTO 90
170 LOCATE 3*I+4,2:PRINT CHR$(203+F*27):
IF F=1 THEN O$=O$+A$ ELSE X$=X$+A$
180 MID$(R$,I,1)="0"
190 IF F=1 THEN T$=O$ ELSE T$=X$
200 IF LEN(T$)<3 THEN 300
210 FOR I=1 TO 3
220 IF I=LEN(T$) THEN 300
230 FOR J=I+1 TO LEN(T$)
240 IF J=LEN(T$) THEN 300
250 FOR K=J+1 TO LEN(T$)
260 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1
))+VAL(MID$(T$,K,1))=15 THEN 320
270 NEXT K
280 NEXT J
290 NEXT I
300 F=(F+1) MOD 2
310 RETURN
320 LOCATE 12,20:PAPER 2+F:PEN 0:PRINT "
BRAVO! JOGADOR ";CHR$(203+F*27)
330 SOUND 1,478:SOUND 1,478:SOUND 1,379:
SOUND 1,319
340 PEN 1:PAPER 0
```

Teclado

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexos (aprendizagem do teclado).

Dificuldade do jogo: difícil.

O JOGO:

No écran aparece uma reprodução do teclado do seu computador; aparece uma seta por baixo de um carácter, cuja tecla deve ser premida no teclado. Quanto mais depressa se tecléia mais rapidamente se desloca a seta (o tempo decorrido desde o início é apresentado no final da partida).

O PROGRAMA:

Uma varável A\$, dimensionada na linha 50, conterà a lista das teclas respectivas. Os valores das teclas entram na linha 60 para a linha 110 chamados pela rotina situada na linha 300.

Os ciclos encaixados J (120-160) e I (130-150) desenharam o teclado.

A linha 180 define a posição da seta impressa pela linha 190.

A tecla premida é testada nas linhas 220-230 e a seta é apagada pela linha 250.

As mensagens de fim de partida são realizadas nas linhas 270 e 280.

PARA JOGAR:

Faz-se correr o programa com RUN. Dispõe-se então de alguns segundos para premir a tecla que está por cima da seta, depois será apresentada outra seta, etc.

A duração da partida não é limitada; basta teclar 0 para obter a apresentação da pontuação e do tempo total.

Fazer RUN para outra partida.

EXTENSÕES POSSÍVEIS:

Poder-se-á encarar a hipótese de fazer cintilar as letras ou de as apresentar a cores, em vez de as indicar por meio de uma seta. Poder-se-á imprimir na base do écran as letras correctamente tecladas e eventualmente treinar-se a teclar um texto que desfile na parte superior do écran, letra por letra.

```

10 REM TECLADO
20 RANDOMIZE(TIME)
30 MODE 0
40 T=0:S=0:P=0
50 DIM A$(10,3)
60 S$="QWERTYUI":J=1
70 GOSUB 300
80 S$="ASDFGHJK":J=2
90 GOSUB 300
100 S$="ZXCVBNM,":J=3
110 GOSUB 300
120 FOR J=1 TO 3
130 FOR I=1 TO 8
140 PEN I:LOCATE 2*I+J,5*J:PRINT A$(I,J)
150 NEXT I
160 NEXT J
170 T=TIME
180 X=INT(RND*8+1):Y=INT(RND*3+1)
190 LOCATE 2*X+Y,5*Y+2:PEN X:PRINT CHR$(
240)
200 FOR F=1 TO 50+RND*80:A$=INKEY$
210 IF A$="" THEN NEXT F
220 IF A$=A$(X,Y) THEN S=S+1
230 IF A$="0" THEN 270
240 P=P+1
250 LOCATE 2*X+Y,2+5*Y:PRINT " "
260 GOTO 180
270 MODE 1:PEN 3:LOCATE 3,12:PRINT USING
"### RESPOSTAS CERTAS EM ### TENTATIVAS
";S,P
280 LOCATE 11,14:PRINT USING "O SEU TEMP
O: ####";ROUND((TIME-T)/300)
290 SOUND 1,478:SOUND 1,379:SOUND 1,319:
END
300 FOR I=1 TO 8:A$(I,J)=MID$(S$,I,1):NE
XT I:RETURN

```

Destreza

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexos (para um jogador).

Dificuldade do jogo: difícil.

O JOGO:

Números entre 0 e 9 vão aparecer ao acaso em diversos pontos do écran. Permanecem impressos durante um tempo muito curto. O jogador deverá conseguir premir a tecla numérica correspondente enquanto o número está visível.

O número teclado pelo jogador será então impresso em tamanho grande na parte esquerda do écran; cada resposta exacta dá dez pontos. A duração da partida é aleatória e a pontuação será apresentada no final.

O PROGRAMA:

As linhas 50 e 60 contêm, nas duas variáveis alfanuméricas C\$ e B\$, as referências que serão utilizadas para desenhar os números em tamanho grande.

O posicionamento do número tirado pelo AMSTRAD é escolhido pelas linhas 80 e 90, e o número é tirado na linha 110.

A linha 120 realiza a impressão, e a linha 130 determina a duração da apresentação. O ciclo WHILE/WEND (70-170) define a duração do jogo, e o fim da partida é tratado pela linha 190.

O ciclo I (220-270) realiza o desenho em tamanho grande do número teclado pelo jogador e imprime-o na janela # 2, definida na linha 30.

O controlo da exactidão do resultado é efectuado pela linha 280.

PARA JOGAR:

Faz-se correr o programa com RUN e, após uma curta pausa, aparece um primeiro número, perto do centro do écran. O jogador deverá reagir suficientemente depressa, enquanto o número permanece no écran, e premir a tecla respectiva. O número que ele teclou será então desenhado na parte superior esquerda do écran, em tamanho grande.

Se o resultado for exacto, o jogador marcará dez pontos.

O tempo de duração do jogo será função dos números que aparecerão no écran.

No fim da partida será apresentada a pontuação.

É possível parar o jogo em qualquer momento, teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em modificar o jogo para permitir a participação de vários jogadores, eventualmente modificando a apresentação e codificando as teclas que serão postas à disposição de cada um dos jogadores.

Poder-se-á também retirar pontos por cada resultado errado, modificar as regras de cálculo da pontuação e a duração do jogo, etc.

```
10 REM DESTREZA
20 RANDOMIZE(TIME)
30 WINDOW #2,2,10,7,18:PEN 3:PAPER #2,1:
PEN #2,0
40 T=100:S=0:C#=CHR$(143):CLS
50 C#=C#+C#+C#+C#+STRING$(4,32)+C#+C#+
"+C#
60 B#="144413333313121131314413312131121
41133331414114131"
70 WHILE T>=0
80 X=12+INT(RND*20)
90 Y=1+INT(RND*23)
100 C=2+INT(RND*25)
110 N=INT(RND*10)
120 LOCATE X,Y:INK 1,C:PRINT N;CHR$(7)
130 FOR I=1 TO RND*350+400:NEXT I
140 A#=INKEY$:IF A#<>" " THEN 210
150 T=T-N
160 LOCATE X,Y:PRINT " ";
170 WEND
180 CLS
190 PRINT:PRINT "A SUA PONTUACAO E' ";S;
" POR CENTO"
200 SOUND 1,478:SOUND 1,426:SOUND 1,379:
END
210 U=VAL(A#):CLS #2
220 FOR I=1 TO 5
230 A=3*VAL (MID$(B#,5*I+1,1))-2
240 FOR K=1 TO 2
250 LOCATE #2,4,2*I+K-1:PRINT #2,MID$(C#
,A,3)
260 NEXT K
270 NEXT I
280 IF U=N THEN S=S+10
290 GOTO 150
```

Anagramas

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexos (dois jogadores).

Dificuldade do jogo: consoante o nível.

O JOGO:

O AMSTRAD escolhe letras ao acaso, sendo o número das letras função do nível de dificuldade desejado. Depois as letras da palavra assim obtida serão misturadas e apresentadas na base do écran; uma nova palavra é então confeccionada, de modo aleatório, a partir das mesmas letras, e é apresentada sob a primeira, e assim por diante. Os dois jogadores (esquerdo e direito) dispõem cada um de uma tecla (1 ou 0) e devem premi-la quando virem aparecer no fundo do écran uma palavra já existente nele; a sua pontuação é então calculada e será impressa uma mensagem correspondente ao resultado.

O PROGRAMA:

Na linha 50 faz-se entrar o nível do jogo desejado. O ciclo I (80-100) realiza a escolha das letras.

Um segundo ciclo I (120-190) efectua a mistura, e a linha 140 permite detectar a pressão de uma tecla.

A linha 200 imprime nova palavra e faz subir uma linha no écran as palavras já obtidas.

A linha 240 verifica qual dos dois jogadores premiu uma tecla.

O ciclo I (250-270) explora as palavras presentes no écran, linha por linha. Obtém-se assim uma palavra que na linha 260 é comparada com a palavra inicial; se o resultado for exacto aparece a mensagem BRAVO!. Se não, a partida acaba com a mensagem PERDEU... (linha 280).

PARA JOGAR:

Faz-se correr o programa com RUN.

Em seguida faz-se entrar o nível de dificuldade (em princípio de 1 a 3) e teclêia-se ENTER. As palavras são apresentadas a partir da base do écran e vão subindo à medida que o jogo se desenrola; se um dos jogadores crê reconhecer na palavra que aparece uma das palavras já impressas, deverá premir imediatamente a sua tecla (1 ou 0) e o seu resultado será apresentado após um breve instante.

A partida será então terminada. Basta fazer ESC duas vezes para recuperar o controlo do teclado.

EXTENSÕES POSSÍVEIS:

Poder-se-á utilizar outros assuntos além das palavras, modificar os tempos de apresentação, fazer participar mais de dois jogadores, melhorar o som e a cor, etc.

```
10 REM ANAGRAMAS
20 RANDOMIZE(TIME)
30 DIM A$(24)
40 CLS
50 INPUT "NIVEL (DE 1 A 3): ";N
60 IF N<1 OR N>3 THEN 40
70 S$=""
80 FOR I=1 TO N+3
90 S$=S$+CHR$(INT(RND*26+65))
100 NEXT I
110 A$="":B$=S$
120 FOR I=1 TO LEN(S$)
130 N=INT(RND*LEN(S$)+1)
140 D$=INKEY$
150 IF D$="1" OR D$="0" THEN 240
160 IF MID$(B$,N,1)="-" THEN 130
170 A$=A$+MID$(B$,N,1)
180 MID$(B$,N,1)="-"
190 NEXT I
200 LOCATE 1,25:PRINT A$:PRINT CHR$(8)
210 FOR I=1 TO 23:A$(I)=A$(I+1):NEXT I:A
$(24)=A$
220 S$=A$
230 GOTO 110
240 IF D$="1" THEN H$="JOGADOR DA ESQUER
DA" ELSE H$="JOGADOR DA DIREITA"
250 FOR I=1 TO 23
260 IF A$(I)=S$ THEN LOCATE LEN(S$)+4,I:
PEN 2:PRINT "BRAVO! ";H$:SOUND 1,478:SOU
ND 1,379:SOUND 1,319:GOTO 290
270 NEXT I
280 PEN 3:LOCATE LEN(S$)+4,24:PRINT "PER
DEU...";H$:FOR I=1 TO 6:SOUND 1,400,10,
6*(I MOD 2):NEXT I
290 GOTO 290
```

«Zombies»

REFERÊNCIAS:

Dificuldade de programação: bastante difícil.

Categoria de jogo: solitário.

Dificuldade do jogo: bastante fácil.

O JOGO:

Os *zombies* são seres malfazejos que vão tentar destruí-lo, mas você irá tentar utilizá-los em seu proveito, para desembaraçar o seu espaço vital dos buracos negros que o obscurecem. No écran é traçado um quadro que define o seu espaço vital. Neste quadro vão aparecer, de modo aleatório, os *zombies* (são os Z), os buracos negros (quadrados azuis preenchidos a amarelo) e o jogador (uma cabeça). Os *zombies* convergem em direcção ao jogador para tentar capturá-lo, mas este desloca-se mais rapidamente que eles e vai tentar fazê-los engolir pelos buracos negros, sem se deixar apanhar.

A partida termina quer com o desaparecimento do jogador quer por eliminação de todos os *zombies*.

O PROGRAMA:

A posição dos *zombies* é definida por um quadro Z, não dimensionado (abaixo de 10 é inútil fazê-lo).

As linhas 50 a 120 (ciclo I) desenharam o quadro, os *zombies* e os buracos negros.

A linha 130 define a posição inicial do jogador.

Um segundo ciclo I (150-240) realiza a deslocação do jogador.

As linhas 290 a 340 controlam a deslocação do jogador.

PARA JOGAR:

Depois de ter feito correr o programa com RUN, o jogador deverá deslocar-se utilizando as quatro setas e prestando atenção para não sair dos limites do campo de jogo.

Deverá manobrar de modo a fazer que os *zombies* se dirijam aos buracos negros, mas sem se deixar apanhar.

Para parar o jogo antes do fim bastará teclar ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Será interessante ampliar o campo de jogo, aumentar o número de *zombies* ou de buracos negros, melhorar os controlos, colocar o jogador de modo mais aleatório, eventualmente modificar as regras.

Poder-se-á também, claro, melhorar a qualidade do desenho e introduzir a cor.

```

10 REM "ZOMBIES"
20 RANDOMIZE(TIME)
30 BORDER 9:INK 0,9:INK 3,1
40 CLS
50 FOR I=1 TO 22
60 LOCATE I,1:PRINT CHR$(233)
70 LOCATE 1,I:PRINT CHR$(233):LOCATE 23,
I:PRINT CHR$(233)
80 IF RND<0.5 THEN LOCATE I,INT(RND*20+2
):PEN 1:PAPER 3:PRINT CHR$(233):PAPER 0
:PEN 1
90 S=INT(RND*14+2)
100 Z(INT(I/3)+1)=5+S
110 LOCATE I+1,22:PRINT CHR$(233);
120 NEXT I
130 K=INT(RND*18+2)
140 P=0
150 FOR I=1 TO 4
160 P=P+Z(I)
170 IF Z(I)=0 THEN 240
180 LOCATE Z(I+4),Z(I):PRINT " ";
190 Z(I)=Z(I)-SGN(Z(I)-S)
200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
210 IF TEST((Z(I+4)-1)*16+8,(25-Z(I))*16
+8)=1 THEN Z(I)=0:GOTO 230
220 LOCATE Z(I+4),Z(I):PEN 3:PRINT "Z"
230 IF Z(I)=S AND Z(I+4)=K THEN LOCATE 2
5,12:PEN 3:PRINT "PERDEU":WHILE TE<200:S
OUND 1,TE,1:TE=TE+1:WEND:END
240 NEXT I
250 IF P=0 THEN LOCATE 2,23:PRINT "BRAVO
! VOCE MATOU TODOS OS <ZOMBIES>." :SOUND
1,478:SOUND 1,426:SOUND 1,379:SOUND 1,47
8:SOUND 1,426:END
260 LOCATE K,S:PEN 2:PRINT CHR$(224)
270 FOR I=1 TO 1200:NEXT I
280 LOCATE K,S:PRINT " "
290 A#=INKEY#
300 K=K-2*((A#=CHR$(&F3))-(A#=CHR$(&F2))
)
310 IF K<=1 OR K>=23 THEN LOCATE 25,12:P
EN 3:PRINT "QUE PENA! ";CHR$(7):END
320 S=S-2*((A#=CHR$(&F1))-(A#=CHR$(&F0))
)
330 IF S<=1 OR S>=22 THEN K=25:GOTO 310
340 GOTO 140

```

Quadro mágico

REFERÊNCIAS:

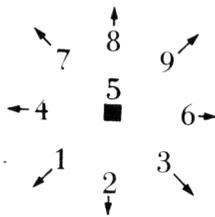
Dificuldade de programação: média.

Categoria de jogo: desenho.

Dificuldade do jogo: fácil.

O JOGO:

Como o título pode sugerir, trata-se muito simplesmente de usar o écran do televisor como um quadro escolar, manipulando um ponto sobre o écran em oito direcções, com a ajuda das teclas 1 a 9, utilizando a convenção seguinte:



Pode-se escolher a cor do ponto premindo uma tecla de A a N; é possível apagar todo o écran premindo ENTER.

Dispõe-se também da possibilidade de deslocar o ponto no écran sem deixar traço, premindo a tecla A.

O PROGRAMA:

A linha 30 transforma as teclas numéricas em teclas de repetição automática do toque.

A linha 40 contém a escolha das possibilidades de cor, sob a forma hexadecimal.

A linha 60 imprime o ponto elementar, a cintilar.

As linhas 80 e 90 controlam as entradas.

As linhas 100 a 130 provocam a deslocação.

A linha 140 verifica se não são ultrapassados os limites do écran.

A linha 160 imprime o ponto.

A rotina 180-190 escolhe a cor.

PARA JOGAR:

Faz-se correr o programa com RUN e no meio do écran aparece um ponto.

Se se premir uma letra de A a N, a cor do ponto muda.

A tecla ENTER apaga tudo e volta-se a começar.
Pode-se sair do programa em qualquer momento, teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á, claro, pensar em escrever um programa idêntico utilizando traçados de segmentos, prever uma rotina que permita a salvaguarda do desenho realizado para uma utilização ulterior, etc.

```
10 REM QUADRO MAGICO
20 MODE 0:INK 15,26,0
30 FOR I=1 TO 20:KEY DEF I,1:NEXT I
40 C$="0123456789ABCD"
50 CLS:ORIGIN 320,200:A=0:B=0:C=4:X=0:Y=
  0
60 PLOT X,Y,15
70 A$=INKEY$:IF A$="" THEN 70
80 IF A$>="A" AND A$<="N" THEN GOSUB 180
:GOTO 60
90 IF A$=CHR$(13) THEN GOTO 50
100 A=VAL(A$):U=(A-1) MOD 3:V=INT((A-1)/
  3) MOD 3
110 U=U-1:V=V-1
120 A=X:B=Y
130 X=X+U:Y=Y+V
140 D=- (ABS(X)<319 AND ABS(Y)<199)
150 IF D=0 THEN X=A:Y=B
160 PLOT A,B,C
170 GOTO 60
180 C=VAL(CHR$(38)+MID$(C$,ASC(A$)-64,1)
  )
190 RETURN
```

«Decómaco»

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: solução do programa «Nicómaco».

Dificuldade do jogo: nenhuma.

O JOGO:

É uma réplica ao programa «Nicómaco», já anteriormente proposto: você fará entrar os restos sucessivos das divisões por 3, 5 e 7 de um número inteiro compreendido entre 0 e 100 e o AMSTRAD calcula e imprime no écran esse inteiro. O processo utilizado pode ser generalizado (descubra você como).

O PROGRAMA:

As linhas 30 a 50 imprimem no écran algumas explicações.

A linha 100 aguarda que você teclie um resto, que será recusado se for incompatível com a divisão pelo número apresentado.

A linha 110 armazena os restos numa variável não dimensionada A(S).

A linha 140 volta a apresentar os restos entrados.

O ciclo I (150-170) desenha pequenos quadrados segundo os restos introduzidos.

As linhas 210 a 220 fazem a gestão do fim da partida.

PARA JOGAR:

Faz-se correr o programa com RUN. É impresso no écran um texto de apresentação, depois é apresentada a mensagem «RESTO DAS DIVISOES: POR 3, POR 5, POR 7».

Deve responder-se 0, 1 ou 2 para 3, depois 0 a 4 para 5 e de 0 a 6 para 7.

A solução é apresentada quando tiver fornecido os três restos.

Para outra partida, teclar S em resposta à pergunta OUTRA TENTATIVA (S/N)?.

Para terminar, premir N.

EXTENSÕES POSSÍVEIS:

Pode-se ampliar o programa aos inteiros de 1 a 1000, utilizando os restos da divisão por 3, 5, 7 e 11; poder-se-á acrescentar som e cor para tornar mais atraente o programa.

```

10 REM "DECOMACO"
20 CLS:INK 2,9:PEN 2
30 PRINT "DEVE ENTRAR OS TRES RESTOS DAS
"
40 PRINT "DIVISOES POR 3, 5, E 7, DE UM
NUMERO"
50 PRINT "INFERIOR A 100."
60 I=3:S=1
70 WHILE I<8
80 LOCATE 5,10:SOUND 1,478:SOUND 1,426:S
OUND 1,379
90 PEN (I-3)/2:PAPER 3:PRINT "RESTO DA D
IVISAO POR ";I;
100 A#=INKEY$:IF A#="" OR A#<"0" OR A#>"C
HR$(47+I) THEN GOTO 100
110 A(S)=ASC(A#)-48
120 S=S+1:I=I+2
130 WEND
140 LOCATE 1,16:PAPER 0:PEN 1:PRINT "RES
TO DAS DIVISOES:   FOR 3   FOR 5   FO
R 7"
150 FOR I=1 TO 3
160 LOCATE 5+9*I,18:PAPER 3:PEN (I-1):PR
INT STRING$(A(I),233)
170 NEXT I
180 T=A(1)*70+A(2)*21+A(3)*15
190 IF T>105 THEN T=T-105:GOTO 190
200 LOCATE 10,22:PAPER 1:PEN 2:PRINT "A
SOLUCAO E' ";T
210 PAPER 0:PEN 1:LOCATE 4,24:INPUT "OUT
RA TENTATIVA (S/N)? ";R$
220 IF UPPER$(R$)="S" THEN RUN

```

Evasão

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: solitário.

Dificuldade do jogo: fácil.

O JOGO:

O seu único problema é conseguir atravessar o écran sem que o AMSTRAD o consiga atingir com um dos numerosos obuses que vai disparar sobre si.

Pode deslocar-se nas quatro direcções (usando as quatro setas), e deve conseguir sair do campo de jogo pela margem direita, sem, claro, ter sido atingido uma única vez.

O PROGRAMA:

A linha 60 define a posição inicial do jogador.

A linha 80 escolhe a coordenada horizontal da casa para onde o AMSTRAD vai disparar (relacionada com a sua posição), e a linha 90 escolhe a coordenada vertical (arbitrária) dessa casa.

A deslocação do jogador é controlada pelas linhas 170 e 180. A sua posição actual (no écran, um símbolo gráfico) é impressa pela linha 190.

A linha 200 verifica se o jogador foi ou não atingido.

O ciclo WHILE/WEND (70-210) efectua o controlo de fim de partida.

PARA JOGAR:

Inicia-se a execução do programa com RUN.

É disparado um primeiro tiro, e só depois disso é que aparece o jogador, representado por um símbolo gráfico.

As quatro teclas de direcção (←, ↓, ↑, →) servem para deslocar o jogador.

Se este é atingido obtém-se, no écran, a mensagem PERDEU, e a partida acaba.

Se ele consegue sair pela direita do écran sem ser atingido obtém-se a mensagem BRAVO, e a partida acaba igualmente.

EXTENSÕES POSSÍVEIS:

A cor e um grafismo mais cuidado melhorarão consideravelmente a atracção do jogo. Poder-se-á também tornar mais aleatório o tiro dos obuses, introduzir obstáculos a contornar, acrescentar controlos, fazer participar vários jogadores, etc.

```

10 REM EVASAO
20 RANDOMIZE(TIME)
30 DEFINT H,K,X,Y
40 CLS
50 ORIGIN 0,200
60 X=1:Y=13
70 WHILE X<38
80 H=16*(X-2)+32*RND
90 K=184-RND*368
100 IF X<4 THEN 150
110 FOR J=1 TO 3+RND*12
120 PLOT H+16*RND,K+16*RND,3
130 SOUND 1,834,1,7,0,0,1
140 NEXT J
150 LOCATE X,Y:PRINT ". "
160 A#=INKEY#
170 X=X-(A#=CHR#(&F3))+ (A#=CHR#(&F2))
180 Y=Y-(A#=CHR#(&F1))+ (A#=CHR#(&F0))
190 LOCATE X,Y:PEN 2:PRINT CHR#(249):PEN
1
200 IF X>3 AND (ABS(H-16*(X-1))+ABS(K-16
*(12-Y))<=15) THEN PEN 3:PRINT "PERDEU!"
:WHILE S<254:SOUND 1,S,1:S=S+1:WEND:END
210 WEND
220 LOCATE 1,23:PRINT "BRAVO!":SOUND 1,4
78:SOUND 1,379:SOUND 1,319

```

Repetições

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: jogo de memória (um ou vários jogadores).

Dificuldade do jogo: média.

O JOGO:

No écran vão aparecer números, um a um.

Em cada jogada o jogador deverá repeti-los todos, desde o início (o que significa que em cada jogada terá mais um para teclear).

Deverá tentar aguentar-se o máximo de tempo possível. A melhor pontuação é memorizada após cada partida, o que permitirá eventualmente jogar com vários jogadores.

O PROGRAMA:

A linha 40 chama uma rotina (linhas 1000 a 1060) que estabelece as regras do jogo.

A linha 70 fabrica os números, que seguidamente são armazenados numa variável alfanumérica A\$.

A linha 80 permite a impressão do último número ao acaso na quarta linha do écran, e a duração da exposição no écran é determinada pela linha 90.

A linha 110 aguarda a resposta do jogador, que na linha 120 é comparada com a resposta correcta.

A linha 150 marca os pontos.

A linha 160 controla a melhor pontuação.

A impressão das mensagens de fim de partida são efectuadas da linha 200 à linha 230.

PARA JOGAR:

A execução do programa com a ordem RUN provoca a apresentação no écran das regras do jogo e, depois, de um número que desaparece. O aparelho faz então a pergunta SUA SEQUENCIA?.

Responde-se fornecendo-lhe o número, seguido de ENTER. Se estiver correcto, aparece durante um breve instante a mensagem CORRECTO. ATENÇÃO!; depois é apresentado um novo número, que desaparece, e de novo a pergunta SUA SEQUENCIA?. Tecleiam-se então os dois números já apresentados, seguidos de ENTER.

Continua-se assim (três numeros, depois quatro, cinco, etc.), até ser cometido um erro, o que provoca a apresentação da pontuação

(e da melhor pontuação obtida até então), seguida da pergunta OUTRO JOGO (S/N)?; tecléia-se S para continuar.

Pode-se também sair do programa a qualquer momento teclando ESC duas vezes antes da apresentação de um número.

EXTENSÕES POSSÍVEIS:

Poder-se-á realizar jogos análogos e mais ou menos aperfeiçoados substituindo os números por letras ou cores ou sons...

```
10 REM REPETICOES
20 RANDOMIZE(TIME)
30 M=0
40 GOSUB 1000
50 A$=""
60 MODE 0:H=0
70 X=INT(RND*10):INK 3,2+INT(25*RND):A$=
A$+RIGHT$(STR$(X),1)
80 CLS:LOCATE 1+RND*20,4:PEN 3:PRINT X:P
EN 1
90 FOR I=1 TO 200+RND*300:NEXT I
100 CLS
110 LOCATE 4,8:PEN 6:PRINT "SUA SEQUENCI
A? ":PRINT:PEN 4:INPUT "":R$
120 IF R$<>A$ THEN 180
130 PRINT:SOUND 1,478:SOUND 1,426:SOUND
1,379:PRINT " CORRECTO. ATENCAO!"
140 FOR I=1 TO 400:NEXT I
150 H=H+1
160 IF M<H THEN M=H
170 GOTO 70
180 MODE 1:INK 0,6:BORDER 6:PEN 2
190 TE=0:WHILE TE<200:SOUND 1,TE,1:TE=TE
+1:WEND
200 INK 0,1:BORDER 1:PRINT "A SUA PONTUA
CAO E' DE ";H
210 PRINT:PRINT "MELHOR PONTUACAO ACTUAL
";M
220 PRINT:INPUT "OUTRO JOGO (S/N)?":R
$
230 IF R$="N" THEN PRINT CHR$(7):;END
240 GOTO 50
1000 CLS:PEN 1
1010 PRINT "DEVE RESPEITAR TODA A SERIE
DOS"
1020 PRINT:PRINT "NUMEROS MOSTRADOS DESD
E O INICIO DO JOGO"
1030 PEN 2:LOCATE 4,12:PRINT "HA' MAIS U
M DE CADA VEZ"
1040 LOCATE 4,25:PAPER 1:PEN 0:PRINT "CA
RREGUE EM <ENTER> PARA COMECAR":;PAPER 0
1050 INPUT "":R$
1060 RETURN
```

Mateus

REFERÊNCIAS:

- Dificuldade de programação: fácil.
- Categoria de jogo: desenho aleatório.
- Dificuldade do jogo: nenhuma.

O JOGO:

Utiliza-se a definição de uma curva em coordenadas polares e a função DRAW para obter a impressão no écran de desenhos aleatórios (do tipo flor), mais ou menos elegantes.

Os dois parâmetros principais, A e B, poderão ser, se necessário, fornecidos à mão (instruções INPUT), para fazer um estudo mais sistemático das diferentes curvas possíveis.

O PROGRAMA:

Os três primeiros parâmetros, Z, K e T, introduzidos na linha 50, definem o raio máximo e a ocupação do écran.

O ciclo WHILE/WEND (70-90) permite escolher as quatro cores de fundo da tinta, validadas na linha 100.

As variáveis aleatórias A e B, escolhidas entre 0 e 2, da linha 110, definem o tipo da curva; poderá ser interessante introduzi-las manualmente (por exemplo: 110 INPUT A, B).

O ciclo H (120-170) realiza o traçado, ponto a ponto. O raio R é calculado na linha 130, e a passagem para coordenadas cartesianas é efectuada nas linhas 140 e 150.

A impressão dos pontos é realizada na linha 160.

A linha 180 imprime uma mensagem de fim.

A linha 210 permite fazer um novo desenho, enquanto as linhas 220 e 230 asseguram um novo traçado sem apagar o anterior.

PARA JOGAR:

Faz-se correr o programa teclando RUN.

O desenho será em seguida executado automaticamente.

A mensagem ACABOU... assinala o fim do traçado.

Se se quer novo traçado teclou-se 0.

Se se quer um traçado sem apagamento do desenho precedente prime-se uma tecla qualquer.

Para parar basta fazer ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á tentar utilizar outras funções para realizar traçados ainda mais extraordinários (por exemplo, com simulação da terceira dimensão).

```
10 REM MATEUS
20 RANDOMIZE(TIME)
30 ORIGIN 320,200
40 DEG
50 Z=7:K=13:T=3:CLS
60 F=0:E(1)=0:E(2)=0:E(3)=0
70 WHILE (E(1)-F)*(E(2)-F)*(E(3)-F)*(E(1)
)-E(2))*(E(1)-E(3))*(E(2)-E(3))=0
80 F=INT(RND*27):E(1)=INT(RND*27):E(2)=I
NT(RND*27):E(3)=INT(RND*27)
90 WEND
100 INK 0,F:INK 1,E(1):INK 2,E(2):INK 3,
E(3)
110 A=3*RND:B=3*RND
120 FOR H=0 TO 360
130 R=Z*SIN(H*T)
140 X=INT(K*R*COS(A*H))
150 Y=INT(K*R*SIN(B*H))
160 DRAW X,Y,C+1
170 NEXT H
180 LOCATE 1,21:PRINT "ACABOU..."
190 SOUND 1,478:SOUND 1,379:SOUND 1,319
200 A$=INKEY$:IF A$="" THEN 200
210 IF UPPER$(A$)="O" THEN 50
220 C=(C+1) MOD 3
230 GOTO 110
```

Através

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexão (dois jogadores).

Dificuldade do jogo: média.

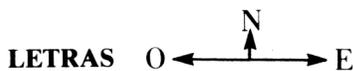
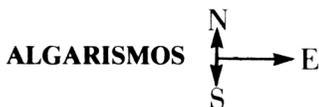
O JOGO:

No écran aparece um quadrado de nove casas, com letras no bordo vertical e números no bordo horizontal.

Um dos jogadores manipula os números, o outro manipula as letras. São possíveis três deslocações para os números e outras três para as letras. As letras podem deslocar-se para cima, para a direita e para a esquerda, e os números podem deslocar-se para a direita, para cima e para baixo.

Ganha o jogador que primeiro conseguir fazer sair todos os seus símbolos (as letras saem por cima, os números saem pela direita).

Não se podem tomar casas e os saltos não são permitidos; também não pode colocar-se numa casa já ocupada.



O PROGRAMA:

As posições dos números e das letras são registadas num quadro N(5,5); as linhas 30 a 70 definem os valores iniciais. Os índices 0 e 5 servem para delimitar o campo de jogo.

As linhas 100 a 140 desenharam o campo de jogo e a linha 170 imprime o jogador que deve jogar.

A linha 190 controla a jogada feita, assim como os ciclos I e J (200 a 240), e a mudança de jogador é assegurada pela linha 330.

PARA JOGAR:

O campo de jogo será apresentado no écran depois de se ter feito RUN, e o computador fica a aguardar a jogada do jogador designado.

Antes de mais faz-se entrar o número ou a letra que representa a peça a deslocar, seguido de uma indicação de direcção sob a forma N (para norte), S (para sul), E (para este) ou O (para oeste).

Assim, poder-se-á fazer 2E seguido de ENTER.

Se a jogada não for válida não será executada, e o código do jogador permanecerá exposto no écran; se for válida, a deslocação será efectuada e obtém-se a mensagem referente ao segundo jogador.

Ganha aquele que primeiro conseguir fazer desaparecer as suas peças.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em ampliar o jogo com mais números ou letras e melhorar os controlos. Pode-se também definir uma estratégia para o AMSTRAD.

```
10 REM ATRAVES
20 DIM N(5,5):K=1
30 FOR I=0 TO 5
40 N(I,1)=ASC("0")+I:N(4,I)=ASC("?")+I
50 N(I,0)=1:N(0,I)=1:N(5,I)=1:N(I,5)=1
60 NEXT I
70 N(4,1)=0
80 MODE 0
90 FOR U=1 TO 4
100 FOR V=1 TO 4
110 LOCATE 4*U,4*V-2:IF N(V,U)=0 THEN PE
N 6:PRINT ". " ELSE PEN(2-(N(V,U)<52)):PR
INT CHR$(N(V,U)):PEN 1
130 NEXT V
140 NEXT U
150 LOCATE 3,18:PEN 2+K
160 PRINT "JOGADOR QUE TEM":LOCATE 6,19:
IF K=0 THEN PRINT "AS LETRAS" ELSE PRINT
"OS ALGARISMOS"
170 LOCATE 1,22:PRINT CHR$(20);"SUA JOGA
DA (";STRING$(-(K=0),"0");"NE";STRING$(-
(K=1),"S");") ";
180 INPUT " ",R$:R$=LEFT$(UPPER$(R$)+" "
,2):L$=LEFT$(R$,1):R$=RIGHT$(R$,1)
190 IF (K=0 AND R$="S") OR (K=1 AND R$="
0") THEN PRINT CHR$(7);:GOTO 170
200 FOR I=1 TO 4
210 FOR J=1 TO 4
220 IF N(I,J)=ASC(L$) THEN 260
230 NEXT J
240 NEXT I
250 PRINT CHR$(7);:GOTO 150
260 A=I:B=J
270 I=I-(R$="S")+(R$="N")
280 J=J-(R$="E")+(R$="0")
290 IF (I=0 AND K=0) OR (J>4 AND K=1) TH
EN 320
300 IF N(I,J)<>0 THEN PRINT CHR$(7);:GOT
O 150
310 N(I,J)=ASC(L$)
320 N(A,B)=0
330 K=ABS(K-1)
340 GOTO 80
```

Calculadora

REFERÊNCIAS:

- Dificuldade de programação: muito difícil.
- Categoria de jogo: simulação de uma calculadora.
- Dificuldade do jogo: média.

O JOGO:

Trata-se da simulação, com apresentação em formato grande, do funcionamento de uma pequena calculadora de bolso, do tipo que só faz quatro operações. As aplicações pedagógicas deste programa são evidentes e é muito simples dominar o modo como funciona.

O PROGRAMA:

Na linha 15 é definida a função VA(I\$) para remediar (em parte) as singularidades da função VAL do Basic AMSTRAD.

As variáveis B\$ e C\$, das linhas 20 e 30, contêm as informações codificadas que permitirão desenhar números em formato grande.

As linhas 60 a 150 ocupam-se da gestão das entradas.

O ciclo I (160-190) realiza a representação dos algarismos no écran, em formato grande, à medida que vão entrando.

A gestão do ponto decimal e do sinal — é feita pelas linhas 200 e 210.

O número total dos símbolos entrados é controlado pela linha 220.

O resultado é inscrito no écran graças a uma rotina de funcionamento automático que se encontra nas linhas 270 a 390 (o automatismo é gerido pela linha 230, no programa principal).

As rotinas colocadas nas linhas 400-410, 420-430, 440-450 e 460-470 efectuem a operação indicada, de acordo com o valor de Z (linha 310).

Cada resultado é memorizado em A(F — 1) (linha 330).

PARA JOGAR:

Fazer correr o programa com RUN.

Depois, teclar os algarismos do primeiro número a entrar, que vão ser impressos um a um, à medida que forem entrando (não se poderá fazer entrar mais de oito símbolos, incluindo o sinal — e o ponto decimal). Teclar um sinal operatório deixa intacta a apresentação no écran, mas a introdução do próximo algarismo ou símbolo provoca o imediato apagamento e a apresentação do novo símbolo.

É de salientar que os algarismos são introduzidos da esquerda para a direita, sentido usual da escrita.

O aparelho aceita como sinais operatórios os símbolos +, -, *, /; o sinal = provocará a apresentação do resultado. As operações são encadeadas automaticamente, sendo o resultado precedente tomado em conta como operador do cálculo seguinte.

Para fazer outra operação será necessário teclar ESC duas vezes, depois RUN.

Teclar ESC duas vezes para sair do programa.

EXTENSÕES POSSÍVEIS:

Proteger o programa contra as entradas erradas (divisão por 0 ou vários pontos decimais seguidos).

Dever-se-á, antes de mais, melhorar a apresentação do sinal — (aqui representado por um traço), depois poder-se-á melhorar o desenho dos algarismos (utilizando, por exemplo, o gerador do AMS-TRAD). Pode-se pensar também numa ampliação para mais de oito algarismos, permitir operações mais complexas (COS, SEN...), acrescentar memórias, etc.

```
10 REM CALCULADORA
15 DEF FN VA(I$)=VAL(I$+STRING$(-(I$="-"
  OR I$="+"),"0"))
20 C$=STRING$(4,233)+STRING$(4,32)+STRIN
  G$(2,233)+" "+CHR$(233)
30 B$="14441333313121131314413312131121
  41133331414114131"
40 F=1:L=0:CLS
50 N=0:T=0:K=0:R$=""
60 A$=INKEY$:IF A$="" THEN 60
70 IF K=1 THEN N=0:R$="":K=0
80 IF A$="" THEN 260
90 X=(1 AND A$="+")+ (2 AND A$="-")+ (3 AN
  D A$="*")+ (4 AND A$="/")
100 R$=R$+A$
110 IF X<>0 THEN 260
120 IF R$="-" THEN A$=R$:GOTO 200
130 IF A$<"." OR A$>"9" OR A$="/" THEN P
  RINT CHR$(7);:GOTO 60
140 IF N=0 THEN CLS
150 IF A$="." OR A$="-" THEN 200
160 FOR I=1 TO 5
170 S=3*VAL(MID$(B$,I+5*FN VA(A$),1))-2
180 LOCATE N+1,I+4:PRINT MID$(C$,S,3)
```

```

190 NEXT I
200 IF A$="-" THEN CLS:LOCATE 2,7:PRINT
STRING$(2,233)
210 IF A$="." THEN LOCATE N+2,9:PRINT CH
R$(233)
220 N=N-4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THE
N 380
240 IF K=1 AND L<>1 THEN 50
250 T=0:GOTO 60
260 IF R$="-" AND A(F)=0 THEN 120
270 N=0:K=1
280 IF (A$="*" OR A$="/") AND FN VA(R$)=
0 THEN A(F)=1:GOTO 300
290 A(F)=FN VA(R$)
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400,420,440,460
320 IF A$<>=" " THEN F=F+1 ELSE F=2
330 A(F-1)=FN VA(R$)
340 IF FN VA(R$)=0 THEN A(F-1)=A(F)
350 R$=STR$(FN VA(R$))
360 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$
,LEN(R$)-1)
370 T=1
380 A$=MID$(R$,T,1)
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=FN VA(R$):
L=1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(
F)=FN VA(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(
F)=FN VA(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(
F)=FN VA(R$):L=1
470 RETURN

```

Caça ao submarino

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexão.

Dificuldade do jogo: média.

O JOGO:

Você está a bordo de um barco de superfície, encarregado de perseguir um submarino escondido algures no seu sector; deve deslocar-se até se colocar por cima do submarino e disparar a uma certa profundidade. Mas atenção: o submarino também é móvel; só quando você estiver exactamente na vertical é que desligará os motores, para não ser referenciado.

O PROGRAMA:

O ciclo WHILE/WEND (30-50) recebe a dimensão da quadrícula.

O ciclo I (80-140) imprime o campo de jogo.

A linha 160 define a posição inicial do submarino; a linha 170 permite a sua deslocação, controlada pela linha 180 (permanece-se nos limites do campo de jogo).

A linha 200 imprime a vossa posição.

As linhas 240 a 260 controlam a vossa deslocação.

A distância é calculada na linha 270, apresentada na linha 280 (numa janela # 1) e testada nas linhas 290 e 300.

O teste de fim de partida é efectuado na linha 330.

PARA JOGAR:

Inicia-se a execução do programa com RUN e faz-se entrar a dimensão; a quadrícula de jogo é impressa. O seu ponto de partida é a posição (0,0) e você desloca-se com as setas.

Após cada deslocação é impressa no écran a nova distância a que se encontra do submarino. Em cada jogada o submarino desloca-se. Se conseguir colocar-se na vertical dele, a distância 0 é mostrada e ele não se move mais.

Você dispara indicando uma profundidade em números inteiros (entre 2 e 4, ambos incluídos). Se não lhe acertar, ele volta a colocar-se algures e recomeça-se; se lhe acertar obtém a mensagem BRAVO! e a partida acaba.

Pode-se parar em qualquer momento o programa utilizando a tecla ESC duas vezes.

EXTENSÕES POSSÍVEIS:

É possível prever uma melhoria do grafismo, melhorar as possibilidades de deslocação do caçador (um tanto lenta, nesta versão), prever percursos mais sofisticados para o submarino, pôr um contador de tempo...

```
10 REM CACA AO SUBMARINO
20 RANDOMIZE(TIME)
30 WHILE H<1 OR H>10
40 PRINT:INPUT "DIMENSÃO DA GRELHA?";H
50 WEND
60 CLS
70 WINDOW #1,4+2*(H+1),39,H+2,H+5:PEN #1
  ,0:PAPER #1,1
80 FOR I=0 TO H
90 PEN 1:LOCATE 2*I+4,1:PRINT I
100 LOCATE 1,2*I+3:PRINT I:PEN 2
110 FOR J=0 TO H
120 LOCATE 2*J+5,2*I+3:PRINT "."
130 NEXT J
140 NEXT I
150 I=1:X=0:Y=0
160 V=INT(RND*H):U=INT(RND*H)
170 U=U+I:V=V+I
180 IF (U<0 OR U>H OR V<0 OR V>H) THEN I
=-I:GOTO 170
190 LOCATE 2*Y+3,2*X+3:IF Y<>0 THEN PRIN
T " "
200 LOCATE 2*Y+5,2*X+3:PEN 3:PRINT CHR$(
255):PEN 2
210 LOCATE 2*Y+7,2*X+3:IF Y<>H THEN PRIN
T " "
220 LOCATE 2*Y+5,2*X+1:IF X<>0 THEN PRIN
T " "
230 LOCATE 2*Y+5,2*X+5:IF Y<>H THEN PRIN
T " "
240 A$=INKEY$:IF A$="" THEN 240
250 Y=Y-(A$=CHR$(&F3) AND Y<H)+(A$=CHR$(
&F2) AND Y>0)
260 X=X-(A$=CHR$(&F1) AND X<H)+(A$=CHR$(
&F2) AND X>0)
270 D=ROUND(SQR(ABS(X-V)^2+ABS(Y-U)^2))
280 CLS #1:LOCATE #1,14-H,1:PRINT #1,"DI
STANCIA":PRINT #1,USING SPACE$(16-H)+"##
":D
290 IF D=0 THEN 320
300 IF D<3 THEN 170
310 GOTO 190
320 LOCATE #1,13-H,3:PRINT #1,"PROFUNDID
ADE":LOCATE #1,18-H,4:INPUT #1,"",D
330 IF D=INT(RND*3+2) THEN LOCATE 1,25:P
RINT "BRAVO!":SOUND 1,478:SOUND 1,379:SO
UND 1,319:END
340 TE=0:WHILE TE<20:SOUND 1,TE,1:TE=TE+
1:WEND
350 GOTO 160
```

Nível 5

<i>Nome do programa</i>	<i>Temas e particularidades</i>
Relógio 1	Preciso, fácil de regular e muito colorido.
Relógio 2	Não deite fora a sua velha televisão; transforme-a em relógio.
Combate aéreo	Não é muito fácil abater um aparelho que está sempre a mover-se.
Cartaz publicitário	Se não sabe o que há-de pôr na sua mostra.
O jogo do míssil	Uma missão bastante perigosa (um jogador).
Submarino	Um clássico dos jogos vídeo (um jogador).
Jogo do galo	Um clássico, mas não desinteressante (dois jogadores).
«Smokey»	Fume sem risco, com um programa original para um jogador.
Corrida de carros	Um jogo de reflexos não demasiado difícil, para um jogador.
Perigo espacial	Será suficientemente hábil para evitar todos os meteoritos?
«Diago»	Está verdadeiramente seguro dos seus reflexos?
Ovos	Não deixe fazer a postura diabólica (um jogador).
Quebra-cabeças com letras	Há um provérbio escondido lá dentro: descubra-o.
Alvos	Um pouco de destreza, bastante sorte: muito visual.
Gotas	Um verdadeiro <i>gag</i> visual. A explorar.
Obstáculos	Bastantes obstáculos a evitar neste jogo de destreza para um jogador.
Capturas no espaço	A sua missão seria fácil, se não fossem os malditos meteoritos.
Anomalia	Exerça a sua perspicácia no reconhecimento dos símbolos.
Vários em um	Um campo de jogo e as peças, para dois jogadores.
Percurso do cavalo	Muito espectacular e instrutivo a diversos títulos.

«Othello»	Para dois jogadores; o jogo move as peças por si.
Damas	Para dois jogadores: clássico.
Quatro a quatro	Um jogo de reflexão um pouco difícil, para dois jogadores.
Xadrez	Um programa assaz simplificado, para dois jogadores.
Muro de tijolos	Um grande clássico numa versão muito completa.
O lagarto de Alice	Muita intervenção do acaso, mas muito espectacular.

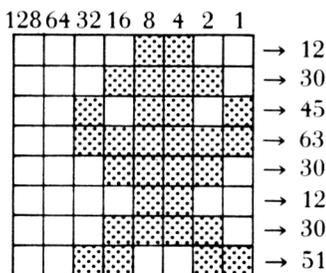
As técnicas particulares do AMSTRAD

UMA INSTRUÇÃO DE CRIAÇÃO DE FORMAS: A INSTRUÇÃO SYMBOL

O Basic AMSTRAD permite redefinir os caracteres afectados às teclas. A ordem SYMBOL AFTER indica a partir de que código ASCII se deseja começar. O carácter de código N é criado pela ordem:

SYMBOL N,12,30,45,63,30,12,30,51

Foi definido assim:



A ordem TAG, indicando que as impressões de texto devem ser efectuadas na posição do cursor gráfico, permite pôr o carácter em qualquer ponto do écran com um simples «PRINT CHR\$(N);» (não esquecer o ponto e vírgula: que evita a aparição de caracteres parasitas).

AS INTERRUPTÕES

Em certos momentos, o computador pode abandonar o programa que executa para realizar uma outra tarefa, antes de retomar o seu trabalho anterior. Controladas pelo relógio interno do AMSTRAD, estas rotinas são úteis quando o tempo desempenha um papel importante. A ordem EVERY chama uma rotina a intervalos regulares (por exemplo, para simular o batimento de um metrónomo). É possível desligar temporariamente a interrupção com uma ordem DI. Será restabelecida por EI.

IDENTIFICAÇÃO DE UM OBJECTO NO ÉCRAN: TEST

A função TEST permite fazer a leitura directa da cor de um ponto do écran de coordenadas X e Y.

Relógio 1

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: simulação.
Dificuldade do jogo: nenhuma.

O JOGO:

O programa simula o funcionamento de um relógio digital, utilizando os caracteres ampliados do AMSTRAD. Apresenta a particularidade de usar o relógio interno do AMSTRAD, para uma boa precisão na hora.

O PROGRAMA:

A linha 60 permite fazer entrar a hora, teclando primeiro as horas, depois os minutos e por fim os segundos. A rotina 1000-1080 é utilizada para definir as zonas de recolha dos dados, controlar as entradas e validar a recolha.

O ciclo WHILE/WEND (110-200) ocupa-se das horas e o ciclo compreendido entre as linhas 120-180 ocupa-se dos minutos.

O ciclo WHILE/WEND (130-160) imprime os segundos. A rotina 2000-2010 incrementa os segundos (a linha 100 autoriza a contagem dos segundos na rotina 2000-2010, chamada graças a uma ordem EVERY na linha 20).

A rotina 3000-3010 assegura a impressão das horas, minutos e segundos com dois algarismos.

PARA JOGAR:

Fazer RUN. No écran é impresso o pedido **FACA ENTRAR A HORA SOB A FORMA HH:MM:SS**, e teclia-se a hora, depois os minutos e por fim os segundos.

A contagem do tempo inicia-se logo que se prima ENTER.

Faz-se parar teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em imprimir os segundos, acrescentar uma possibilidade de despertador, prever uma agenda automática, fazer aparecer um cuco a dar horas, etc.

```

10 RELOGIO 1
20 EVERY 50 GOSUB 2000:DI
30 CLS
40 PRINT:PRINT "ENTRE A HORA SOB A FORMA
  HH:MM:SS"
50 LOCATE 12,4:PRINT "--:--:--":PEN 3
60 FOR N=0 TO 2:LOCATE 12+3*N,4:GOSUB 10
00:T#=T#+R#:NEXT N
70 FOR I=1 TO 3:X(I)=VAL(MID$(T#,2*I-1,2
)):GOSUB 3000:NEXT I
80 MODE 0
90 MOVE 11*16,6*32+8:DRAWR 0,32,2:DRAWR
9*32,0,2:DRAWR 0,-32,2:DRAWR -9*32,0,2
100 EI
110 WHILE X(1)<24
120 WHILE X(2)<60
130 WHILE X(3)<60
140 I=3:GOSUB 3000
150 LOCATE 7,12:PRINT X$(1);" ";X$(2);"
";X$(3)
160 WEND
170 X(3)=0:X(2)=X(2)+1:PEN 1+X(2) MOD 12
:I=2:GOSUB 3000
180 WEND
190 X(2)=0:I=2:GOSUB 3000:X(1)=X(1)+1:I=
1:GOSUB 3000
200 WEND
210 FOR I=1 TO 3:X(I)=0:GOSUB 3000:NEXT
I
220 GOTO 110
1000 R$="":S=0
1010 A$=INKEY$:IF A$="" THEN 1010
1020 IF (N=0 AND S=0 AND A$<>"0" AND A$<
>"E"1" AND A$<>"2") THEN 1010
1030 IF ((N=1 OR N=2) AND S=0 AND A$<>"0
" AND VAL(A$)>=6) THEN 1010
1040 IF (S=1 AND A$<>"0" AND VAL(A$)=0)
THEN S=S-1:GOTO 1010
1050 R$=R#+A$:PRINT A$;;S=S+1
1060 IF S<2 THEN 1010
1070 IF N=0 AND VAL (R$)>24 THEN RUN
1080 RETURN
2000 X(3)=X(3)+1:PRINT CHR$(7);
2010 RETURN
3000 X$(I)="0"+RIGHT$(STR$(X(I)),LEN(STR
$(X(I)))-1):X$(I)=RIGHT$(X$(I),2)
3010 RETURN

```

Relógio 2

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: desenho.
Dificuldade do jogo: nenhuma.

O JOGO:

Transformar o seu velho écran de televisão num relógio digital com números em formato grande, graças a um programa que lhe permitirá animar uma vitrina, fazer um relógio de fantasia, e que facilmente poderá prestar-se a muitos outros usos.

O PROGRAMA:

A linha 30 permite-lhe fazer entrar a hora sob a forma de um inteiro de quatro algarismos (por exemplo: 1024 significará 10 horas e 24 minutos).

A rotina 1000-1030 é executada todos os segundos, ao passo que a rotina 2000-2210 o é todos os minutos.

As linhas 2070-2190 realizam o desenho, substituindo a impressão à medida que o tempo passa (com a ajuda da variável K, incrementada na linha 2180).

As linhas 2000 a 2040 asseguram a progressão da hora.

A linha 1000 efectua a impressão de dois pontos a piscar, para os segundos.

PARA JOGAR:

Depois de ter sido feito correr com RUN, o programa fica a aguardar a hora; teclia-se a hora e aguarda-se que chegue o momento exacto para premir ENTER.

A hora é então rapidamente apresentada e o relógio está a trabalhar. Para o parar bastará premir duas vezes a tecla ESC.

Ao dar entrada à hora terá de se usar imperativamente um inteiro de quatro algarismos; para 5 horas e 6 minutos terá de se teclar 0506 seguido de ENTER.

EXTENSÕES POSSÍVEIS:

Será interessante modificar o programa para fazer dele um despertador, com alarme; pode-se também melhorar o grafismo, introduzindo a cor.

```

10 REM RELOGIO 2
20 CLS
30 INPUT "HORA (FORMA XXXX)";A$
40 GOSUB 2050
50 EVERY 50,1 GOSUB 1000
60 EVERY 50*60,2 GOSUB 2000
70 GOTO 70
1000 LOCATE 20,4:PAPER H:PRINT " ":LOCAT
E 20,6:PRINT " ":PAPER 2
1010 SOUND 1,473,1
1020 H=ABS(H-3)
1030 RETURN
2000 A=1+VAL(A$)
2010 IF A=2360 THEN A=0
2020 A$="000"+RIGHT$(STR$(A),LEN(STR$(A)
)-1)
2030 A$=RIGHT$(A$,4)
2040 IF RIGHT$(A$,2)="60" THEN A$=RIGHT$
(STR$(VAL(LEFT$(A$,2))+1),2)+"00"
2050 CLG 0:PAPER 2
2060 K=1
2070 WHILE K<5
2080 A=VAL(MID$(A$,K,1))
2090 IF A<>1 AND A<>4 THEN LOCATE 6*K+4,
1:PRINT " "
2100 FOR J=1 TO 5
2110 IF A=0 OR (A>3 AND A<>7) THEN LOCAT
E 6*K+4,J:PRINT " "
2120 IF A<>5 AND A<>6 THEN LOCATE 6*K+6,
J:PRINT " "
2130 IF A MOD 2=0 AND A<>4 THEN LOCATE 6
*K+4,J+4:PRINT " "
2140 IF A<>2 THEN LOCATE 6*K+6,J+4:PRINT
" "
2150 NEXT J
2160 IF A>1 THEN LOCATE 6*K+5,5:PRINT "
"
2170 IF A<>1 AND A<>4 AND A<>7 THEN LOCA
TE 6*K+4,9:PRINT " "
2180 K=K+1
2190 WEND
2200 SOUND 1,478:SOUND 1,253:SOUND 1,358
2210 RETURN

```

Combate aéreo

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexos (um jogador).

Dificuldade do jogo: difícil.

O JOGO:

Ao centro do écran encontra-se o visor do seu aparelho, e o avião inimigo desloca-se na sua frente, de um modo que ao princípio pode parecer-lhe incoerente.

Você pode deslocar-se em todas as direcções, graças às setas de movimento, e vai tentar apanhar o inimigo no centro do seu visor.

Dispara sobre ele premindo COPY, e vê os projectéis convergirem sobre o alvo. Mas este continua a deslocar-se, mesmo enquanto você dispara, e não é lá muito evidente que consiga apanhá-lo.

A partida continuará a desenrolar-se até que você o tenha atingido (aparecendo então a mensagem GANHOU!).

O PROGRAMA:

A linha 70 desenha o aparelho inimigo, cuja deslocação aleatória será realizada pelas linhas 90 e 100.

As linhas 120 e 130 permitem-lhe pilotar o seu aparelho.

A linha 170 desenha o visor do seu aparelho.

A rotina de gestão do tiro vai da linha 200 à linha 250, e o teste de fim de partida é feito na linha 240.

A linha 180 verifica se você disparou ou não.

O ruído da metralhadora é provocado pela chamada da rotina 260-270, durante o tempo do disparo, graças à ordem EVERY.

PARA JOGAR:

Depois de o programa ter sido posto em funcionamento com RUN, você deve manobrar para tentar enquadrar no seu visor o aparelho inimigo.

Para isso dispõe das setas de deslocação (teclas ↑, ↓, ←, →), e pode disparar a qualquer momento premindo a tecla COPY.

O aparelho inimigo pode continuar a deslocar-se de modo aleatório, mesmo depois de você ter disparado.

A partida terminará com a impressão da mensagem GANHOU!, se você atingir o avião inimigo.

Pode-se também parar em qualquer momento premindo ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Será bastante fácil fazer uma apresentação no écran com um grafismo mais cuidado, utilizando a cor. Poder-se-á pensar em estabilizar a posição do avião inimigo durante o tiro.

Pode-se também introduzir vários aviões, contar os pontos ou o tempo, tomar em consideração a sua própria reserva de combustível, etc.

```
10 REM COMBATE AEREO
20 RANDOMIZE(TIME)
30 EVERY 4,1 GOSUB 260:DI
40 CLS
50 A=5:B=15:AV$=CHR$(242)+CHR$(198)+CHR$(243)
60 N=21
70 LOCATE B,A:CALL &BD19:PRINT AV$
80 U=A:V=B
90 A=A+2*INT(RND*2)-1
100 B=B+2*INT(RND*2)-1
110 A$=INKEY$
120 B=B-2*(A$=CHR$(&F3))+2*(A$=CHR$(&F2))
130 A=A-2*(A$=CHR$(&F1))+2*(A$=CHR$(&F0))
140 IF B<1 OR B>38 THEN B=ABS(B-37)
150 IF A<1 OR A>24 THEN A=ABS(A-23)
160 LOCATE V,U:CALL &BD19:PRINT " "
170 LOCATE 18,9:PEN 2:PRINT CHR$(150);CHR$(158);CHR$(156):LOCATE 18,10:PRINT CHR$(151);CHR$(144);CHR$(157):LOCATE 18,11:PRINT CHR$(147);CHR$(155);CHR$(153):PEN 1
180 IF A$=CHR$(&E0) OR N<21 THEN GOSUB 200
190 GOTO 70
200 EI:PEN 3:LOCATE 29-N,N:PRINT CHR$(204):LOCATE N+9,N:PRINT CHR$(205):PEN 1
210 LOCATE 29-N,N:PRINT " ":LOCATE N+9,N:PRINT " "
220 N=N-1
230 IF N>9 THEN RETURN
240 IF ((ABS(10-A))<=1) AND ((20-B)<=4 AND B<=20) THEN LOCATE B,A:PEN 3:PRINT AV$:PAPER 2:PEN 3:LOCATE 17,23:PRINT "GANHOU!":PAPER 0:END
250 N=21:DI:RETURN
260 SOUND 1,271,1,5
270 RETURN
```

Cartaz publicitário

REFERÊNCIAS:

- Dificuldade de programação: média.
- Categoria de jogo: cartaz publicitário.
- Dificuldade do jogo: nenhuma.

O JOGO:

Um texto qualquer deverá ser introduzido no início do programa. Poderá ser relativamente longo (mas inferior a 254 caracteres); escolher-se-á em seguida o sentido e a velocidade a que deve desfilar, após o que o texto vai aparecer (só uma linha de cada vez) e começará a desfilar, à velocidade e no sentido escolhidos.

O PROGRAMA:

O texto será introduzido, letra por letra, nas linhas 70 a 100.

A apresentação será feita em formato grande, graças à linha 160.

O sentido de desfile para a esquerda é preparado na linha 190, e o desfile para a direita na linha 200.

O texto será impresso no écran pela linha 180, e a duração da apresentação (velocidade de desfile) é escolhida pelo ciclo 210-240.

A linha 230 permitirá mudar as condições de desfile, se o desejarmos.

Uma ordem EVERY, suspensa durante a recolha dos dados, permite mudar regularmente a cor das impressões (rotina 260-270).

PARA JOGAR:

Far-se-á correr o programa com RUN. Após a apresentação da mensagem ENTRE O SEU TEXTO, introduzir-se-á o texto que se quer fazer desfilar, seguido de ENTER.

Obtém-se em seguida a pergunta DESFILE PARA A DIREITA (D) OU PARA A ESQUERDA (E)?, a que se responderá tecendo D ou E.

Depois, à pergunta VELOCIDADE?, responder-se-á introduzindo um número grande para uma velocidade elevada ou um número pequeno para uma velocidade baixa (por exemplo, de 0 a 40 ou mais).

O texto começará a desfilar logo que se tenha teclado ENTER.

Se se deseja mudar as condições de desfile basta carregar na tecla 0 para reaparecerem as perguntas SENTIDO e VELOCIDADE.

Para interromper o programa teclava-se ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á introduzir a cor, realizar uma impressão ondulante, fazer subir as linhas no écran, etc.

```
10 REM CARTAZ PUBLICITARIO
20 EVERY 250 GOSUB 260:DI
30 CLS
40 B$=""
50 PRINT "ENTRE O SEU TEXTO : "
60 PRINT
70 A$=INKEY$:IF A$="" THEN 70
80 IF A$=CHR$(13) THEN B$=B$+" " :GOTO 11
0
90 B$=B$+A$
100 PEN 3:PRINT A$::GOTO 70
110 PRINT:PRINT:PEN 1:PRINT "DESFILE PAR
A A DIREITA (D) OU PARA A ESQUERDA (E)?"
:PEN 2:INPUT " ",S$:PEN 1
120 PRINT
130 PRINT "VELOCIDADE (DE 0 A 50)?"::PEN
2:INPUT " ",V:PEN 1
140 V=51-V
150 EI
160 MODE 0
170 T=LEN(B$)
180 LOCATE 1,12:PRINT " "+MID$(B$,1,18)+
" "
190 IF LEFT$(S$,1)="E" THEN B$=RIGHT$(B$
,T-1)+LEFT$(B$,1)
200 IF LEFT$(S$,1)="D" THEN B$=RIGHT$(B$
,1)+LEFT$(B$,T-1)
210 FOR J=0 TO 10*V
220 C$=INKEY$
230 IF C$="0" THEN MODE 1:DI:GOTO 110
240 NEXT J
250 GOTO 180
260 C=(C+1) MOD 25:INK 1,1+C
270 RETURN
```

O jogo do míssil

REFERÊNCIAS:

- Dificuldade de programação: média.
- Categoria de jogo: contra o AMSTRAD.
- Dificuldade do jogo: média.

O JOGO:

O AMSTRAD encontra-se em estado de alerta: você está aos comandos de um avião encarregado de uma missão de espionagem no seu território; o seu objectivo é sobrevoar uma base inimiga (representada por um carácter ampliado) e regressar são e salvo à sua base. Mas, quando o AMSTRAD o referenciar, ele vai enviar em sua perseguição um míssil aperfeiçoadíssimo que não descansará enquanto não o destruir.

O seu avião pode deslocar-se em todas as direcções, graças às teclas de 1 a 9. Conseguirá cumprir a sua missão?

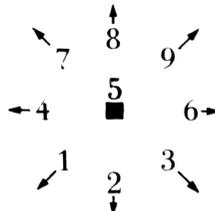
O PROGRAMA:

- A linha 50 desenha a base (sob a forma de um H ampliado).
- A linha 60 desenha a sua base de partida.
- As linhas 80 e 90 escolhem o momento de partida do míssil.
- A linha 100 desenha o seu avião.
- As linhas 120 e 130 definem o seu movimento segundo oito direcções, utilizando as rotinas das linhas 220, 230, 250 e 260.
- A linha 140 assegura a partida do míssil, que é desenhado pela linha 150; a deslocação do míssil é gerida pelas linhas 160 e 170.
- As linhas 180 e 200 controlam o fim da partida.

PARA JOGAR:

Fazer correr o programa com RUN.

O campo de jogo é apresentado no écran e você pode deslocar o seu aparelho, utilizando as teclas de 1 a 9, de acordo com a seguinte convenção:



Deverá fazê-lo contornar o território inimigo (se o míssil toca neste território a partida acaba).

Se o míssil o destruir obterá uma mensagem ATINGIDO!... e a partida acaba.

Para acabar a partida basta teclar ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Será sem dúvida necessário melhorar os diversos controlos do programa (o avião sobrevoa a base inimiga, regressa à sua própria base...).

Pode-se também prever diversos níveis de jogo (fazendo, por exemplo, variar a velocidade do míssil), prever a utilização de engodos pelo avião, a explosão do míssil ao cabo de certo tempo, etc.

```
10 REM O JOGO DO MISSIL
20 RANDOMIZE(TIME)
30 EVERY 10 GOSUB 280:DI
40 CLS:M=0
50 LOCATE 18,6:PEN 3:PRINT CHR$(141);CHR
$(142);CHR$(10);STRING$(2,8);CHR$(135);C
HR$(139)
60 PEN 2:LOCATE 1,21:PRINT STRING$(4,127
):LOCATE 1,22:PRINT STRING$(4,127):PEN 1
70 U=34:V=16:X=2:Y=20
80 S=INT(RND*V+3)
90 S=S-1
100 LOCATE X,Y:PRINT CHR$(239);
110 A*=INKEY$:IF A*="" THEN 110
120 A=VAL(A*):F=SGN(6.5-A)+2*(A=4)
130 ON (A AND A<5)+((10-A) AND A>5) GOSU
B 220,230,250,260
140 IF S<>0 THEN 90 ELSE EI
150 LOCATE U,V:PEN 3:PRINT CHR$(252);:PE
N 1
160 U=U-SGN(U-X)
170 V=V-SGN(V-Y)
180 IF (U=X AND U=Y) THEN LOCATE 1,23:PR
INT "ATINGIDO!...":INK 0,3,0:SOUND 1,319
:SOUND 1,379:SOUND 1,478:FOR I=1 TO 500:
NEXT I:INK 0,1:END
190 IF X>=20 AND Y<=5 THEN M=1
200 IF M=1 AND X<=4 AND Y=20 THEN LOCATE
1,23:PRINT "MISSAO CUMPRIDA":SOUND 1,47
8:SOUND 1,379:SOUND 1,319:END
210 GOTO 100
220 X=X-F
230 Y=Y+F
240 RETURN
250 GOSUB 230
260 X=X+F
270 RETURN
280 SOUND 1,1000,2,2,0,0,1
290 RETURN
```

Submarino

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexos (um só jogador).

Dificuldade do jogo: fácil.

O JOGO:

Você comanda um caça-submarinos (à superfície, portanto) e sob a sua unidade passam regularmente submarinos, a diferentes velocidades.

Pode, em qualquer momento, largar granadas submarinas e pode pôr a hipótese de atingir várias vezes o mesmo submarino.

A sua pontuação estará permanentemente exposta, ao centro do écran (relação entre o número de disparos que atingiram o alvo e o número de submarinos aparecidos).

O PROGRAMA:

A linha 70 permite ao AMSTRAD escolher a profundidade do submarino.

A linha 90 desenha o seu navio.

A linha 100 desenha a superfície da água.

A linha 120 desenha o submarino.

O disparo é efectuado por uma rotina específica colocada nas linhas 200 a 240.

A deslocação (muito rápida) do seu navio é realizada pelas linhas 170 e 180.

O teste que permite saber se atingiu ou não o alvo é efectuado na linha 140.

PARA JOGAR:

Teclar RUN para começar a partida.

O navio será deslocado graças às setas direita e esquerda.

A tecla 0 permite disparar.

Como o navio se desloca três vezes mais depressa que os submarinos, é possível atingir várias vezes o mesmo alvo.

Para terminar a partida é preciso teclar ESC duas vezes ou atingir 99 submarinos.

EXTENSÕES POSSÍVEIS:

É bastante elementar melhorar a qualidade gráfica do jogo, utilizar a cor e os sons, fazer aparecer vários submarinos deslocando-se a velocidades diferentes, eventualmente permitir até que os submarinos desapareçam...

```
10 REM SUBMARINO
20 RANDOMIZE(TIME):EVERY 100 GOSUB 250
30 INK 0,0:INK 1,6:INK 2,18:INK 3,2
40 P=0:C=16:S=5
50 CLS
60 LOCATE 1,S:PRINT STRING$(5," ")
70 T=37:S=INT(RND*8+5):P=P+0.01
80 X=2
90 PEN 2:LOCATE C,3:PRINT "   ";CHR$(213);CHR$(142);CHR$(141);CHR$(212);"   "
100 LOCATE 1,4:PEN 3:PRINT STRING$(40,CHR$(196))
110 PAPER 1:PEN 2:LOCATE 15,19:PRINT USING "PONTUACAO ###.##";P:PAPER 0
120 PEN 2:LOCATE T,S:PRINT CHR$(154);CHR$(231);CHR$(154);"  "
130 IF X=2 THEN Y=C+2
140 IF S=X AND (T+1=Y OR T+2=Y) THEN P=P+1:LOCATE T,S:PRINT "***":SOUND 1,456,20,3,0,0,1
150 T=T-1:IF 100*(P-INT(P))>=99 THEN LOCATE 1,21:PRINT "ACABOU . . . . .":SOUND 1,478:SOUND 1,426:SOUND 1,379:END
160 IF T<1 THEN LOCATE Y,X:PRINT " ":GOTO 60
170 A$=INKEY$
180 C=C+3*((A$=CHR$(&F2) AND C>1)-(A$=CHR$(&F3) AND C<33))
190 IF A$<>"0" AND X=2 THEN 90
200 LOCATE Y,X:IF X=2 THEN PEN 2 ELSE PEN 1
210 PRINT " ":X=X+1
220 IF X>S THEN 80
230 LOCATE Y,X:PRINT CHR$(252)
240 GOTO 90
250 SOUND 1,200,8:RETURN
```

Jogo do galo

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: jogo de reflexão (dois jogadores).

Dificuldade do jogo: fácil.

O JOGO:

Um grande clássico dos jogos de estudantes, aqui proposto numa versão que não usa a máquina como opositor. Cada jogador deverá colocar, à vez, um O ou um X (aqui o símbolo de copas ou o de ouros — um coração ou um losango) numa casa livre da grelha do jogo. Ganha o que primeiro conseguir um alinhamento de três símbolos idênticos (o alinhamento pode ser horizontal, vertical ou segundo uma das diagonais).

O PROGRAMA:

O campo de jogo é desenhado com a ajuda do ciclo I (50-70).

A linha 90 ocupa-se de imprimir o símbolo de copas ou o de ouros correspondente ao jogador que está na sua vez de jogar.

Um símbolo cintilante (linha 120) é impresso no canto superior esquerdo do campo de jogo, mesmo que a casa já esteja ocupada; as linhas 150 e 160 permitem deslocar esse símbolo pelo campo de jogo.

A linha 170 verifica se a casa onde se quer jogar está livre, e se se deseja jogar.

A linha 180 regista as jogadas e a linha 190 realiza a mudança de jogador.

PARA JOGAR:

O campo de jogos será apresentado no écran depois de se ter teclado RUN.

Um símbolo cintilante aparece então no canto superior esquerdo do écran; o jogador pode deslocá-lo no campo de jogo utilizando as quatro setas (teclas ←, →, ↑, ↓).

Quando se chega à casa que se deseja ocupar basta premir a tecla COPY; a jogada será recusada se a casa não estiver livre.

Se a casa estiver livre, o símbolo que corresponde a esse jogador é nela inscrito, e muda a vez do jogador.

Quando a partida termina convém teclar ESC duas vezes para sair do programa. Poder-se-á depois jogar outra partida, teclando RUN.

EXTENSÕES POSSÍVEIS:

Poder-se-á realizar após cada jogada um controlo para verificar se um dos jogadores ganhou. Pode-se também realizar um jogo contra a máquina, mas convém evitar que ele seja demasiado sistemático (por exemplo, evitar que a máquina faça sempre a sua primeira jogada na casa central).

```
10 REM JOGO DO GALD
20 MODE 0
30 INK 0,18:INK 1,16:INK 2,26:INK 3,6:BO
RDER 16
40 CLG 1
50 FOR I=1 TO 9
60 PEN 0:LOCATE 6,I+1:PRINT STRING$(9,14
3)
70 NEXT I
80 K=0
90 LOCATE 6,18:PAPER 6:PEN 2+K:PRINT "JO
GADOR "+CHR$(228-K):PAPER 0
100 I=1:J=1
110 C=TEST ((3*(I+1))*32+16,(25-(3*J))*1
6+8)
120 PEN 2+K:LOCATE 1+3*(I+1),3*J:PRINT C
HR$(143)
130 LOCATE 1+3*(I+1),3*J:IF C<>0 THEN PE
N C:PRINT CHR$(230-C) ELSE PRINT " "
140 A$="":A$=INKEY$
150 I=I-(A$=CHR$(&F3) AND I<3)+(A$=CHR$(
&F2) AND I>1)
160 J=J-(A$=CHR$(&F1) AND J<3)+(A$=CHR$(
&F0) AND J>1)
170 IF C<>0 OR A$<>CHR$(&E0) THEN 110
180 LOCATE 1+3*(I+1),3*J:PEN 2+K:PRINT C
HR$(228-K)
190 K=ABS(K-1)
200 GOTO 90
```

«Smokey»

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: solitário.

Dificuldade do jogo: média.

O JOGO:

Uma coluna de fumo amarelo aparece no écran. O jogador, representado por uma cabeça, torna-se visível à direita. Primeiro o jogador deve posicionar-se sobre a coluna de fumo; depois, a partir daí, vai automaticamente engolindo o fumo, desde que consiga manter-se sobre a coluna.

O PROGRAMA:

A linha 60 faz aparecer o fumo na base do écran; a linha 70 permite mudar (para a direita ou para a esquerda) a direcção da coluna, e a linha 90 faz subir a coluna.

O jogador aparece graças à linha 110, e a sua deslocação é efectuada pelas linhas 160 a 180.

As linhas 230 a 250 encarregam-se da apresentação do resultado.

PARA JOGAR:

Fazer correr o programa com RUN.

O fumo começa a elevar-se, enquanto o jogador, aparecido sob a forma de uma cabeça vermelha num canto do écran, vai sendo deslocado na sua direcção com a utilização das setas. Quando se coloca sobre o fumo começa automaticamente a engoli-lo, e deverá tentar seguir a coluna sem se desviar (usando as setas). Ganha se conseguir chegar à base do écran antes de o fumo atingir o topo do écran.

A mensagem final que aparece no écran indica então BRAVO! ou PERDEU..., assim como a pontuação obtida.

EXTENSÕES POSSÍVEIS:

Poder-se-á, conservando a ideia inicial, impor outras regras: por exemplo, engolir todo o fumo, que poderá também voltar a surgir de outro local se se desvia da coluna inicial.

Poder-se-á também melhorar os controlos de fim de partida e a qualidade do grafismo.

```

10 REM "SMOKEY"
20 RANDOMIZE(TIME)
30 S=0:X=24:Y=1+INT(RND*40)
40 U=1:V=17
50 CLS
60 LOCATE Y,X:PRINT CHR$(228)
70 Y=Y-(RND<0.5 AND Y<24)+(RND<0.5 AND Y
>1)
80 T=1
90 X=X-T
100 IF X=0 THEN 230
110 PEN 3:LOCATE V,U:PRINT CHR$(224);:PE
N 1
120 IF U>=24 THEN 230
130 IF T<>0 THEN FOR I=0 TO 300:A$=INKEY
$:IF A$<>" " THEN 150
140 IF T<>0 THEN NEXT I
150 LOCATE V,U:PRINT " "
160 U=U-(A$=CHR$(&F1) AND T=1 AND X<23)
170 U=U-(T=0)
180 V=V-(A$=CHR$(&F3) AND V<40)+(A$=CHR$
(&F2) AND V>1)
190 C=TEST((V-1)*16+8,(25-U)*16+8)
200 IF C<>1 THEN 60
210 T=0:S=S+1
220 GOTO 110
230 LOCATE 12,12:IF U>=24 THEN PRINT "
BRAVO!":SOUND 1,478:SOUND 1,379:SOUND 1,
319:GOTO 250
240 FOR I=1 TO 6:SOUND 1,278,10,6*(I MOD
2):NEXT I:PRINT "PERDEU..."
250 LOCATE 12,14:PRINT USING "PONTUACAO:
##";S

```

Corrida de carros

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexos (um jogador).

Dificuldade do jogo: média.

O JOGO:

Você circula numa estrada muito sinuosa, sem qualquer obstáculo, e deve tentar segui-la tanto tempo quanto for possível. O traçado do seu percurso permanece impresso permanentemente e a sua pontuação é apresentada quando tiver um acidente.

O PROGRAMA:

A impressão da estrada é assegurada pelo ciclo I (70-100).

A linha 110 faz avançar a estrada e a linha 120 imprime a viatura.

A linha 150 controla o seu movimento e a linha 160 determina se a estrada faz viragens ou não.

A linha 180 verifica se tocou na borda da estrada.

A linha 190 incrementa a pontuação.

A linha 230 imprime o tempo decorrido até ao acidente.

PARA JOGAR:

Iniciar a execução do programa com a ordem RUN. O desenho da estrada começa a evoluir e o seu veículo, representado por uma viatura, aparece.

Para o deslocar para a direita ou para a esquerda deverá utilizar as teclas *e* (setas de deslocação do cursor). Terá de evitar, durante tanto tempo quanto for possível, tocar qualquer das bordas da estrada, o que provocará a paragem do programa e a impressão da mensagem ACIDENTE, seguida da indicação da pontuação realizada.

Em qualquer momento poder-se-á parar o desenrolar da partida premindo duas vezes a tecla ESC.

EXTENSÕES POSSÍVEIS:

Pode-se pensar num jogo com vários jogadores, cada um na sua vez, registando a melhor pontuação obtida por cada qual.

Pode-se também encarar a possibilidade de alargar a pista e de fazer correr várias viaturas.

Poderá também inspirar-se nos jogos de arcada, que são particularmente numerosos sobre este tema.

```

10 REM CORRIDA DE CARROS
20 RANDOMIZE(TIME)
30 SYMBOL AFTER 240
40 SYMBOL 240,93,127,85,20,93,127,93,8
50 T=TIME:CLS:PEN 1
60 X=24:Y=11:A=5:B=10:S=0
70 FOR I=0 TO 6
80 LOCATE Y+I+1,X+1:PRINT CHR$(143);
90 IF I<2 OR I>4 THEN PEN 2:LOCATE Y+I+1
,X+1:PRINT CHR$(207);:PEN 1
100 NEXT I
110 PRINT:PRINT
120 LOCATE B,A:PEN 3:PAPER 0-(N>18):PRIN
T CHR$(240):PAPER 0:PEN 2
130 FOR I=1 TO 10:A$=INKEY$:IF A$="" THE
N NEXT I
140 C=B
150 B=B-(A$=CHR$(&F3) AND B<40)+(A$=CHR$
(&F2) AND B>1)
160 Y=Y-(-2*RND AND Y<25)-(2*RND AND Y>3
)
170 IF N<19 THEN N=N+1:GOTO 200
180 IF TEST((B-1)*16+8,(25-A-1)*16+8)<>1
THEN 220
190 S=S+1:PAPER 1
200 LOCATE C,A:PRINT " "
210 PAPER 0:GOTO 70
220 LOCATE 20,12:PAPER 2:PEN 3:PRINT USI
NG "ACIDENTE. PONTUACAO ###";S
230 LOCATE 20,14:PRINT USING "TEMPOS: ##
#";ROUND ((TIME-T)/300)
240 SOUND 1,478:SOUND 1,426:SOUND 1,379
250 PAPER 0:PEN 1

```

Perigo espacial

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexos (um jogador).

Dificuldade do jogo: média.

O JOGO:

Você comanda um engenho espacial que acaba de penetrar numa nuvem de meteoritos. Deve tentar ultrapassar a nuvem sem chocar com um único meteorito. A sua pontuação será apresentada no fim da partida, quando colidir com um meteorito.

O PROGRAMA:

A posição de cada meteorito é determinada pelas linhas 60 e 80, e é impressa pelas linhas 70 e 90.

O avanço dos meteoritos é realizado pela linha 100.

As linhas 100 e 110 verificam se você colidiu ou não com um meteorito.

A linha 120 desenha a sua nave espacial e a linha 140 permite deslocá-la.

O contador encontra-se na linha 150.

A linha 500 provoca um bipe-bipe durante todo o tempo do jogo.

PARA JOGAR:

O jogo é iniciado com a ordem RUN.

A sua nave aparece pelo meio do écran e os meteoritos vêm ao encontro dela a partir da base do écran. Para se deslocar pode utilizar as teclas de deslocação horizontal (←e→).

Pode-se acabar a partida em qualquer momento teclando ESC duas vezes. De outro modo, o jogo acaba quando você chocar com um meteorito, e a pontuação é então exibida no écran.

EXTENSÕES POSSÍVEIS:

Seria interessante tornar o jogo cada vez mais difícil fazendo aparecer cada vez mais meteoritos à medida que a partida se vai desenrolando.

Pode-se também tentar destruir os meteoritos pulverizando-os à medida que aparecem.

```

10 REM PERIGO ESPACIAL
20 RANDOMIZE (TIME)
30 EVERY 10 GOSUB 500
40 CLS
50 X=18:Z=0:Y=X
60 H=1+INT(RND*38)
70 IF RND<0.7 THEN LOCATE H,25:PEN 2:PRI
NT CHR$(231);
80 K=1+INT(RND*38)
90 IF RND<0.6 THEN LOCATE K,25:PEN 1:PRI
NT CHR$(231);
100 LOCATE 1,25:PRINT CHR$(10);B=TEST((X
-1)*16+8,(25-12)*16+8):C=TEST(X*16+8,(25
-12)*16+8)
110 IF B MOD 3<>0 OR C MOD 3<>0 THEN 190
120 LOCATE Y,11:PRINT STRING$(2,32);
130 LOCATE X,12:PEN 3:PRINT CHR$(242)+CH
R$(243);
140 A$=INKEY$
150 Y=X
160 X=X-(A$=CHR$(&F3) AND X<37)+(A$=CHR$
(&F2) AND X>1)
170 Z=Z+1
180 GOTO 60
190 TE=0:WHILE TE<200:SOUND 1,TE,1:TE=TE
+1:WEND
200 LOCATE 11,25:PAPER 3:PEN 1:PRINT "A
PONTUACAO E ";Z:PAPER 0:PEN 1:END
500 SOUND 2,40,2:RETURN

```

«Diago»

REFERÊNCIAS:

Dificuldade de programação: fácil.

Categoria de jogo: jogo de reflexos (um jogador).

Dificuldades do jogo: difícil.

O JOGO:

Faz-se correr o jogo com RUN e obtém-se uma primeira mensagem que o informa sobre a *performance* a atingir (por exemplo, cinco pontos negros ou dezassete pontos negros...), depois o tempo é impresso e inicia-se a contagem decrescente, a começar em 10 segundos. O jogo principia quando se chega a 0 segundos. Começam então a aparecer círculos na parte inferior esquerda do écran, formando uma linha diagonal, e você deve premir a tecla 0 para parar a impressão dos pontos quando calcular que o número de pontos inicialmente pedido já apareceu no écran.

Obtém-se então uma mensagem de BRAVO! ou de PERDEU!, conforme o resultado obtido.

O PROGRAMA:

O número de pontos negros será escolhido ao acaso, na linha 50.

O tempo é descontado pela rotina 240-260, chamada a cada segundo (linha 309).

A linha 120 imprime os pontos e fabrica a linha diagonal.

A pressão da tecla 0 é detectada na linha 150, e a linha 170 controla o fim da partida (qualquer reflexo).

As mensagens de fim de jogo são executadas pelas linhas 170, 200 e 210.

PARA JOGAR:

Faz-se correr o programa com RUN.

Obtém-se então a impressão no écran da frase O BOM REFLEXO E' O DE..., que indica o número de pontos que permite ganhar.

O tempo será então descontado sob a forma de ATENÇÃO: FALTAM ... SEGUNDOS, que indica o tempo que resta antes da aparição da diagonal.

Quando se atinge 0, a diagonal aparece e começa a deslocar-se no écran.

Se se calcula que foi atingido o número de pontos anunciado deve-se premir a tecla 0.

A mensagem impressa no écran será então BRAVO!, se se ganhou, ou PERDEU!, no caso contrário. Depois aparece a frase OUTRA TENTATIVA (S/N)?; responder S para jogar outra partida.

EXTENSÕES POSSÍVEIS:

Poder-se-á encarar uma adaptação do jogo para dois jogadores ou mais, deixando a diagonal formar-se completamente e testando a coincidência aquando da pressão da tecla afecta a cada jogador.

Pode-se então realizar um grafismo mais cuidado e modificar o princípio da deslocação dos objectos no écran (deslocação horizontal, vertical ou aparição ao acaso...).

```
10 REM "DIAGO"
20 RANDOMIZE(TIME)
30 EVERY 50 GOSUB 240
40 CLS
50 X=INT(RND*15+3)
60 S=10
70 PRINT "O BOM REFLEXO E' O DE ";:PAPER
 3:PRINT STRING$(X,202):PAPER 0
80 WHILE S>0:WEND
90 DI
100 CLS
110 U=1
120 LOCATE 1,26:PRINT TAB(U) CHR$(202)
130 FOR I=1 TO 100:NEXT I
140 U=U+1
150 A$=INKEY$
160 IF A$="0" THEN 190
170 IF U>22 THEN PEN 3:PRINT "PERDEU!":W
HILE TE<200:SOUND 2,TE,1:TE=TE+1:WEND:PE
N 1:GOTO 210
180 GOTO 120
190 IF U<>X+1 THEN U=23:GOTO 170
200 PEN 2:PRINT "BRAVO!":SOUND 1,478:SOU
ND 1,426:SOUND 1,379:PEN 1
210 PRINT:PRINT "OUTRA TENTATIVA (S/N)"
220 A$=INKEY$:IF A$="" THEN 220
230 IF UPPER$(A$)="S" THEN RUN ELSE END
240 LOCATE 5,12:PRINT USING "ATENCAO. FA
LTAM ## SEGUNDOS";S
250 S=S-1:SOUND 1,200
260 RETURN
```

Ovos

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: solitário.

Dificuldade do jogo: média.

O JOGO:

Uma ave misteriosa desloca-se ao acaso pelo écran, batendo as asas. Põe ovos (representados pela letra O) que você (representado por um quadrado de cor) deve tentar apanhar. **Atenção: só os ovos da sua cor dão pontos, os outros tiram-lhe pontos.** A sua pontuação (número de ovos recuperados) será exposta permanentemente e a partida desenrola-se durante um lapso de tempo definido.

O PROGRAMA:

A ave é desenhada em duas partes (por causa do movimento), nas linhas 70 e 110); a sua deslocação aleatória é gerida pela rotina 220-310 (em particular as linhas 270 a 300).

A linha 130 permite fazer o programa regressar à linha 70.

A deslocação do jogador é gerida pelas linhas 150 e 160 e o teste que permite detectar a presença de um ovo é efectuado na linha 170.

A linha 190 permite a apresentação da pontuação.

O fim do jogo é detectado na linha 230.

PARA JOGAR:

Far-se-á correr o programa com RUN.

A ave inicia então a sua deslocação ao acaso pelo écran, e começa a pôr os seus ovos. Quando o símbolo que representa o jogador (um quadrado colorido) surge no écran, ele pode deslocar-se utilizando as quatro teclas de deslocação. É preciso, a partir desse momento, apanhar o máximo possível de ovos antes do fim da partida.

Poder-se-á parar de jogar em qualquer momento premindo ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Com base no mesmo princípio, poder-se-á facilmente imaginar outros temas de jogo. Poder-se-á, claro, melhorar a cor, os sons, o grafismo, e, eventualmente, definir um jogo para vários jogadores, cada um deles tentando apanhar o máximo de ovos.

```

10 REM DVDS
20 RANDOMIZE(TIME)
30 INK 0,9
40 SYMBOL AFTER 240:SYMBOL 240,24,24,126
,255,255,36,36,36
50 CLS:P=200:R=18:S=10:T=0
60 X=1+INT(RND*36):Y=1+INT(RND*21)
70 LOCATE X,Y:PEN 2:PRINT CHR$(198);CHR$(
(240));CHR$(198);
80 FOR I=1 TO 30:A$=INKEY$
90 IF A$="" THEN NEXT I ELSE GOSUB 140
100 GOSUB 210
110 LOCATE X,Y:PEN 2:PRINT CHR$(193);CHR
$(240);CHR$(192);
120 GOSUB 210
130 GOTO 70
140 LOCATE R,S:PRINT " "
150 R=R-(A$=CHR$(&F3) AND R<38)+(A$=CHR$(
&F2) AND R>1)
160 S=S-(A$=CHR$8&F1) AND S<22)+(A$=CHR$(
&F0) AND S>1)
170 G=TEST ((R-1)*16+8,(25-S)*16+8)
180 T=T-2*(G=3)+(G=2)
190 LOCATE 15,23:PAPER 1:PEN 3:PRINT USI
NG "PONTUACAD ###";T;;PAPER 0
200 RETURN
210 A=X:B=Y
220 P=P-1
230 IF P=0 THEN PRINT "ACABOU...":SOUND
1,478:SOUND 1,426:SOUND 1,379:END
240 LOCATE R,S:PAPER 3:PEN 0:PRINT CHR$(
233);:PAPER 0
250 PEN 2-(RND<0.5):LOCATE A,B:PRINT " "
+CHR$(231)+" ";
260 X=ABS(X+INT(1+2*(RND<0.5)))
270 Y=ABS(Y+INT(1+2*(RND<0.5)))
280 IF X>37 OR X<1 THEN X=1
290 IF Y>22 OR Y<1 THEN Y=1
300 RETURN

```

Quebra-cabeças com letras

REFERÊNCIAS:

Dificuldade de programação: média.

Categoria de jogo: reflexão.

Dificuldade do jogo: difícil.

O JOGO:

Qual de nós, um que outro dia, não teve entre mãos um desses pequenos quebra-cabeças em que se tenta deslocar de uma casa para outra pequenos quadrados coloridos de plástico, até à obtenção de um resultado bem definido?

No seu tempo, o «14-15» de Sam Loyd teve tanto êxito como o nosso célebre cubo de Rubik.

O quebra-cabeças que aqui lhe propomos utiliza 35 letras, dispostas ao acaso num quadrado de 6 por 6 casas, e essas letras, colocadas na sua ordem inicial, permitem-lhe reconstituir um provérbio muito conhecido.

Uma casa colorida permite-lhe deslocar as letras, e um contador apresenta permanentemente o número de tentativas. Conseguirá descobrir o provérbio?

O PROGRAMA:

A grelha de jogo é desenhada pelos ciclos encaixados K (70-100) e I (40-120).

A frase a reconstituir (que poderá ser modificada) está colocada numa variável R\$, na linha 130.

A mistura das letras e a impressão são feitas pelo ciclo I (140-220).

A linha 250 realiza a apresentação do quadrado colorido; este é deslocado pelas linhas 290 e 300, utilizando as quatro setas.

A linha 320 reimprime a respectiva letra na posição do quadrado colorido.

A linha 340 contabiliza o número de tentativas, que é impresso graças à linha 240.

PARA JOGAR:

Faz-se correr o programa com RUN.

A partida, evidentemente, será muito mais difícil se não conhecermos o provérbio a reconstituir. Pode-se deslocar as letras e fazê-las passar para a posição do quadrado colorido, com a ajuda das quatro teclas de direção (←, →, ↑, ↓).

O número de tentativas está permanentemente visível, por cima do quebra-cabeças.

Pode-se parar em qualquer momento premindo ESC duas vezes.

EXTENSÕES POSSÍVEIS:

O mesmo princípio pode ser aplicado a toda a espécie de quebra-cabeças (manipulação de números, de cores, eventualmente de sons). Pode-se também prever várias frases, com uma escolha feita por um gerador aleatório.

```
10 REM QUEBRA-CABECAS COM LETRAS
20 RANDOMIZE(TIME)
30 MODE 0
40 FOR I=1 TO 13
50 V=-I MOD 2:LOCATE 4,I+3
60 PRINT CHR$(149-(2*V)+(I=1)+4*(I=13));
70 FOR K=1 TO 6
80 PRINT CHR$(154+122*NOT V);
90 PRINT CHR$(159+(10*NOT V)+(I=1)+4*(I=
13)+2*(K=6 AND V));
100 NEXT K
110 PRINT
120 NEXT I
130 A$="";R$="CANDEIAQUEVAIAFRETEALUMIA
DUASVEZES"
140 FOR I=1 TO 35
150 X=INT(RND*35+1)
160 IF MID$(R$,X,1)="-" THEN 150
170 A$=A$+MID$(R$,X,1)
180 MID$(R$,X,1)="-"
190 X=X MOD 6:Y=INT(I/6)
200 A$(X,Y)=MID$(A$,I,1)
210 LOCATE 2*X+5,2*Y+5:PEN 3:PRINT A$(X,
Y)
220 NEXT I
230 X=0:Y=0:C=32
240 LOCATE 1,21:PEN 6:PRINT USING "NUMER
0 DE TENTATIVAS: ###";
250 LOCATE 2*X+5,2*Y+5:PEN 3:PRINT CHR$(
143)
260 T$=INKEY$
270 IF T$="" THEN LOCATE 2*X+5,2*Y+5:PRI
NT CHR$(32):GOTO 250
280 I=X:J=Y
290 X=X-(T$=CHR$(&F3) AND X<5)+(T$=CHR$(
&F2) AND X>0)
300 Y=Y-(T$=CHR$(&F1) AND Y<5)+(T$=CHR$(
&F0) AND Y>0)
310 A$(I,J)=A$(X,Y):A$(X,Y)=""
320 LOCATE 2*I+5,2*J+5:PRINT A$(I,J)
330 IF I=X AND J=Y THEN PRINT CHR$(7);:G
OTO 250
340 S=S+1:SOUND 1,478:SOUND 1,379:SOUND
1,319
350 GOTO 240
```

Alvos

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: azar.

Dificuldade do jogo: fácil.

O JOGO:

Consiste simplesmente em disparar contra alvos, tentando marcar o máximo de pontos; o número de alvos (de 1 a 6) será escolhido no início da partida, e a pontuação obtida estará permanentemente visível. Se se atinge a borda de um alvo, a partida acaba. Marcar-se-ão tanto mais pontos quanto maior for o número que se atingir.

O PROGRAMA:

O número de alvos é feito entrar na linha 50. O ciclo I (80-100) efectua a impressão dos alvos.

A linha 120 apresentará permanentemente a pontuação.

A bala é impressa pela linha 130.

O ciclo I (150-200) realiza a impressão dos números contidos nos alvos e permite a deslocação da bala (linha 190).

A linha 220 verifica se houve disparo e a linha 230 verifica se a bala está ainda na linha de partida.

A linha 240 faz progredir a bala para a direita e a linha 250 verifica se os alvos foram atingidos ou não; depois as linhas 260 e 270 vêem o que é que foi atingido.

A linha 280 incrementa a pontuação.

As linhas 290 e 300 ocupam-se do fim da partida.

PARA JOGAR:

Fazer correr o programa com RUN e teclar o número de alvos desejado, seguido de ENTER. Os alvos são impressos no écran e a partida começa. A sua bala, representada por um O, é apresentada no canto inferior esquerdo; pode deslocá-la com as setas para cima e para baixo (teclas ↑ e ↓) e pode disparar quando quiser premindo COPY.

A pontuação será modificada em função do resultado obtido, e aparece uma nova bala. É possível parar em qualquer momento premindo ESC duas vezes; de outro modo, o jogo acaba ao fim de certo número de tiros.

EXTENSÕES POSSÍVEIS:

Poder-se-á sem dificuldade aumentar o número de alvos, modificar a regra de aparição dos números nos alvos para a tornar menos casual, providenciar para que a própria bala seja representada por um número que deverá coincidir à chegada com o do alvo.

```
10 REM ALVOS
20 RANDOMIZE(TIME)
30 INK 0,9
40 PRINT:INPUT "NUMERO DE ALVOS: ";N
50 IF N<1 OR N>6 THEN 40
60 P=10:J=30:S=0:B=1
70 CLS
80 FOR I=1 TO N
90 LOCATE J+1,3*I:PRINT CHR$(140);CHR$(1
32):LOCATE J+1,3*I+1:PRINT " ";CHR$(133)
:LOCATE J+1,3*I+2:PRINT CHR$(131);CHR$(1
29)
100 NEXT I
110 K=0:A=18*16
120 LOCATE 8,21:PAPER 1:PEN 3:PRINT USI
NG "FONTUACAO ### - BALA ### ";S;B:PAPER
0
130 TAG:MOVE K,A:PRINT CHR$(231);:TAGOFF
140 D=A
150 FOR I=1 TO N
160 U=INT(RND*P)
170 LOCATE J+1,3*I+1:PAPER 2:PRINT CHR$(
U+48):V(I)=U
180 A#=INKEY$
190 A=A+4*(A#=CHR$(&F1) AND A>6*16)-4*(A
#=CHR$(&F0) AND A<400)
200 NEXT I
210 TAG:MOVE K,D:PRINT " ";:TAGOFF
220 IF A#=CHR$(&EO) AND K=0 THEN B=B+1:G
OTO 240
230 IF K=0 THEN 130
240 K=K+16
250 IF K<16*J THEN 130
260 C=TEST (K+15,A-8)
270 IF C<>2 THEN LOCATE 12,12:PRINT "PER
DEU!":WHILE TE<200:SOUND 1,TE,1:TE=TE+1:
WEND:PAPER 0:END
280 S=S+V(ROUND((25-((A-8)/16))/3))
290 P=P-1:PRINT CHR$(7);
300 IF P=0 THEN SOUND 1,478:SOUND 1,426:
SOUND 1,379:END
310 GOTO 110
```

Gotas

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: desenho.

Dificuldade do jogo: nenhuma.

O JOGO:

É um verdadeiro programa de desenho animado que se pode considerar, de facto, como um passatempo. No écran está desenhado um recipiente, visto em corte, contendo já obstáculos. Uma gota de água, simbolizada por um círculo azul, aparece por cima do recipiente e cai, preenchendo-o pouco a pouco. A sua trajectória, à medida que encontra um obstáculo, é completamente determinada pelo acaso: salta para a direita ou para a esquerda, até já não haver mais nenhuma possibilidade. Então cai uma nova gota, e assim por diante até finalmente transbordar.

O PROGRAMA:

Os dois ciclos encaixados I (60-130) e J (70-120) desenharam o recipiente e os obstáculos.

A linha 170 desenha a gota e a linha 210 envia para uma rotina (370-390) que permite determinar se há ou não choque com um obstáculo.

Se não há choque, passa-se para a linha 290 e a gota continua a cair.

Se há, a gota salta para a direita ou para a esquerda (linha 230), de acordo com o valor definido para K na linha 150.

Verifica-se de novo se há ou não obstáculo ou se a gota pode continuar a cair (linhas 240, 250 ou 260).

Consoante o resultado do teste será utilizada a linha 270, 290, 310 ou 340.

PARA JOGAR:

A ordem RUN desencadeia o desenho do recipiente e dos obstáculos; a gota aparece, começa a cair e acaba por encontrar um lugar. Aparece outra gota e tudo recomeça.

O programa pára por si próprio quando o recipiente transborda; pode-se pará-lo em qualquer momento premindo ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se, claro, acrescentar a cor e o som e também pensar num jogo em que uma pequena personagem tenha de escapar às gotas ou ao afogamento.

```
10 REM GOTAS
20 RANDOMIZE(TIME)
30 SYMBOL AFTER 240:SYMBOL 240,28,62,127
,127,127,62,28
40 EVERY 40 GOSUB 1000:DI
50 CLS
60 FOR I=12 TO 20
70 FOR J=12 TO 27
80 LOCATE 11,I:PEN 1:PRINT CHR$(143):LOC
ATE 28,I:PRINT CHR$(143)
90 LOCATE J,20:PRINT CHR$(143)
100 IF I<>12 AND I<>20 THEN LOCATE J,I:P
EN 3:PRINT CHR$(143):PEN 1
110 IF I<>20 AND RND<0.7 THEN LOCATE INT
(RND*15+12),I:PRINT " "
120 NEXT J
130 NEXT I
140 Y=INT(RND*7+18):X=6
150 K=-2*(RND<0.5)-1
160 I=0:EI
170 LOCATE Y,X:PEN 2:PRINT CHR$(240)
180 FOR J=0 TO 100:NEXT J
190 LOCATE Y,X:PRINT " "
200 U=X+1:V=Y
210 GOSUB 370
220 IF J=0 THEN 290
230 U=X:V=Y+K
240 GOSUB 370
250 IF J=0 THEN 310
260 IF I=0 THEN 340
270 LOCATE Y,X:PRINT CHR$(240)
280 GOTO 140
290 X=X+1
300 GOTO 150
310 Y=Y+K
320 I=1
330 GOTO 170
340 K=-K
350 I=1
360 GOTO 230
370 J=TEST((V-1)*16+8,(25-U)*16+8)
380 IF U=24 THEN LOCATE 14,6:PRINT "ACAB
DU...":SOUND 1,478:SOUND 1,379:SOUND 1,3
19:END
390 RETURN
1000 SOUND 129,0:SOUND 1,20,5:RETURN
```

Obstáculos

REFERÊNCIAS:

Dificuldade de programação: bastante difícil.

Categoria de jogo: jogo de reflexos (para um jogador).

Dificuldade do jogo: média.

O JOGO:

No écran deslocam-se obstáculos para irem chocar com uma pequena viatura que dispõe da possibilidade de se deslocar para a esquerda ou para a direita para os evitar e que deverá fazê-lo durante tanto tempo quanto possível.

A distância percorrida será exibida permanentemente, e o jogo termina quando a viatura choca com a borda do campo de jogo ou com um obstáculo.

O PROGRAMA:

A linha 80 faz avançar os obstáculos para a viatura do jogador, que é impressa graças à linha 70.

As linhas 80 a 100 fabricam a estrada (as linhas 90 e 100, embora idênticas, são ambas indispensáveis, porque o valor de RND não será, em princípio, o mesmo nestas duas linhas).

A linha 120 imprime a pontuação numa janela # 1 protegida contra o desfilar da estrada.

A linha 140 permitirá deslocar o veículo.

As linhas 150 e 160 permitem controlar se há ou não um obstáculo e continuar o programa se for caso disso.

A pontuação é incrementada na linha 160.

Na linha 170, a utilização de uma tinta transparente permite mostrar a viatura esmagando o obstáculo.

PARA JOGAR:

Faz-se correr o programa teclando RUN.

A estrada e os obstáculos aparecem na base do écran e sobem para a viatura. Esta pode deslocar-se com a ajuda das setas de deslocação horizontal (teclas ←→).

O choque com um obstáculo ou com a borda da estrada interrompe o desenrolar do programa e provoca a apresentação da pontuação.

No entanto, pode-se sair do programa em qualquer momento, usando a tecla ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Será muito fácil melhorar o desenho da viatura, da estrada, dos obstáculos. Poder-se-á melhorar o som e a cor, admitir um certo número de acidentes, registar a melhor pontuação (para vários jogadores), eventualmente desenhar várias viaturas, etc.

```
10 REM OBSTACULOS
20 RANDOMIZE(TIME)
30 EVERY 5 GOSUB 1000
40 SYMBOL SFTER 240:SYMBOL 240,93,127,85
,20,93,127,93,8:SYMBOL 241,8,20,8,62,42,
8,28,54
50 CLS:WINDOW #0,1,11,1,25
60 S=0:Y=5
70 LOCATE Y,1:PEN 3:PRINT CHR$(240)
80 LOCATE 1,25:PEN 1:PRINT CHR$(143);STR
ING$(8,32);CHR$(143)
90 IF RND<0.7 THEN LOCATE INT(RND*8+2),2
5:PEN 2:PRINT CHR$(241);
100 IF RND<0.8 THEN LOCATE INT(RND*8+2),
25:PEN 3:PRINT CHR$(241)+CHR$(10)
120 PEN #1,1+(S MOD 3):LOCATE #1,15,15:P
RINT #1,USING "PONTUACAO:###";S
130 A$=INKEY$
140 Y=Y-(A$=CHR$(&F3) AND Y<10)+(A$=CHR$
(&F2) AND Y>1)
150 C=TEST((Y-1)*16+8,(25-1)*16+8)
160 IF C=0 THEN S=S+1:GOTO 70
170 LOCATE Y,1:PRINT CHR$(22);CHR$(1);CH
R$(240)
180 WHILE TE<200:SOUND 1,TE,1:TE=TE+1:WE
ND
190 END
1000 SOUND 1,400,2,,,2:RETURN
```

Capturas no espaço

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexos (um só jogador).

Dificuldade do jogo: média.

O JOGO:

A sua nave espacial navega num perigoso campo de meteoritos e a sua missão é apoderar-se da maior quantidade possível de minério de urânio, um metal indispensável que só se pode encontrar nesses locais particularmente perigosos.

Você deve manobrar para recolher o urânio, sem se deixar atingir por um meteorito.

O PROGRAMA:

A linha 70 permite fazer aparecer o minério de urânio.

A linha 90 fará aparecer os meteoritos e assegura a sua deslocação pelo écran.

As linhas 110 a 130 controlam o desenrolar do jogo (chocou com um meteorito ou capturou um bloco de urânio?).

O choque com um meteorito provoca o fim da partida.

A sua nave espacial é impressa pela linha 150 e a sua deslocação é controlada pela linha 160.

PARA JOGAR:

Faz-se correr o programa com RUN e os meteoritos começam a subir no écran, assim como os blocos de urânio.

As teclas de deslocação horizontal (←e→) serão utilizadas para deslocar a sua nave e recolher o urânio, mas o menor choque com um meteorito provocará o fim da partida e a apresentação da quantidade de urânio capturado.

Poder-se-á também parar o jogo em qualquer momento premindo ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se tornar o jogo cada vez mais difícil aumentando progressivamente o número de meteoritos ou, ainda, minutando a duração da partida.

Pode-se também prever uma exposição permanente da pontuação, introduzir uma segunda nave espacial manobrada por uma potência estrangeira e que procura também apoderar-se do urânio...

```

10 REM CAPTURAS NO ESPACO
20 RANDOMIZE(TIME)
30 INK 1,0
40 CLS:SYMBOL AFTER 240:SYMBOL 240,0,8,8
,54,127,28,8
50 X=15:Z=0
60 S=INT(1+RND*39)
70 LOCATE S,25:PEN 2:PRINT "#";
80 S=INT(1+RND*39)
90 IF RND<0.7 THEN LOCATE S,25:PEN 3:PRI
NT "*";:PRINT CHR$(10)
100 FOR I=1 TO 10:NEXT I
110 C=TEST((X-1)*16+8,(25-12)*16+8)
120 IF C=3 THEN PAPER 3:PEN 1:LOCATE 11,
24:PRINT USING "PONTUACAO OBTIDA: ###";Z
:WHILE TE<200:INK 0,TE MOD 26:SOUND 1,TE
,1:TE=TE+1:WEND:PAPER 0:END
130 IF C=2 THEN Z=Z+1:SOUND 1,478:SOUND
1,426:SOUND 1,379
140 LOCATE X-1,11:PRINT STRING$(3,32)
150 PEN 1:LOCATE X-1,12:PRINT " "+CHR$(2
40)+" "
160 A$=INKEY$
170 X=X-(A$=CHR$(&F3) AND X<38)+(A$=CHR$
(&F2) AND X>2)
180 GOTO 60

```

Anomalia

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: solitário.

Dificuldade do jogo: média.

O JOGO:

Baseia-se essencialmente numa boa capacidade de observação. Duas grelhas, de 25 casas cada, são desenhadas no écran. Símbolos idênticos são apresentados nas duas grelhas, mas um deles foi modificado na grelha da direita. O jogo consiste em descobrir qual, fazendo entrar as suas coordenadas pela ordem linha-coluna.

O PROGRAMA:

O ciclo I (50-80) imprime os números de referência das grelhas e pontos com uma tinta desmaiada.

O ciclo principal I (100-160) imprime os símbolos das duas grelhas, numa ordem diferente.

A linha 110 escolhe o carácter a imprimir.

A rotina 1000-1040 ocupa-se da impressão numa posição livre (a rotina é chamada duas vezes, uma para cada grelha). Aquando da tiragem da letra modificada, a linha 1030 assegura-se de que ela não é idêntica àquela que substitui. A matriz B contém o valor de cada letra da grelha da direita.

As linhas 170 e 180 mudam um dos caracteres, de modo aleatório.

As coordenadas do ponto testado são feitas entrar nas linhas 190 e 200, e o controlo é efectuado pela linha 210.

A linha 220 ocupa-se da apresentação do resultado.

PARA JOGAR:

Depois de se fazer correr o programa com a instrução RUN obtém-se a apresentação das duas grelhas. Dever-se-á então procurar, por comparação, qual o carácter que foi modificado (na grelha da direita).

Faz-se então entrar o seu número de linha, seguido de ENTER, e o número de coluna, seguido de ENTER.

O programa responderá então BRAVO! ... ou PERDEU! ...

EXTENSÕES POSSÍVEIS:

Será possível utilizar símbolos gráficos de diversas cores, o que poderá simplificar a pesquisa.

Pode-se também pensar em utilizar desenhos mais sofisticados, melhorar o controlo do resultado, admitir como boa resposta a da grelha da direita ou a da esquerda, introduzir uma minutagem, etc.

```
10 REM ANOMALIA
20 INK 1,1:INK 2,1:INK 3,1:RANDOMIZE(TIM
E)
30 FOR I=1 TO 5:PO$=PO$+CHR$(144)+" ":NE
XT I
40 T=5:DIM B(5,5):CLS
50 FOR I=1 TO 5
60 LOCATE 2*I+1,1:PRINT I:LOCATE 1,2*I+1
:PRINT I
70 LOCATE 2*I+19,1:PRINT I:LOCATE 19,2*I
+1:PRINT I
75 LOCATE 4,2*I+1:PRINT PO$:LOCATE 22,2*
I+1:PRINT PO$
80 NEXT I
100 FOR I=1 TO 22
110 D=INT(26*RND+1)
120 L=0:S=T
130 GOSUB 1000
140 L=L+1
150 GOSUB 1000
160 NEXT I
170 S=0
180 GOSUB 1000:INK 1,24:INK 2,6:INK 3,9
190 PEN 1:LOCATE 1,16:INPUT "LINHA.....
.":I
200 PRINT:INPUT "COLUNA .....":T
210 LOCATE 1,20:IF B(T,I)=D THEN PRINT "
BRAVO!..." ELSE PRINT "PERDEU!...":WHILE
TE<200:SOUND 1,TE,1:TE=TE+1:WEND
220 PRINT:PRINT "A LETRA A DESCOBRIR ERA
":CHR$(64+D):END
1000 U=3+2*INT(RND*T):V=4+L+2*INT(RND*S)
1010 C=TEST((V-1)*16+8,(25-U)*16+8)
1020 IF C<>1 AND S<>0 THEN 1000
1030 IF S=0 THEN IF B((V-20)/2,(U-1)/2)=
D THEN 1000
1040 IF L<>0 THEN B((V-20)/2,(U-1)/2)=D
1050 LOCATE V,U:PEN 2-(L<>0):PRINT CHR$(
D+64)
1060 RETURN
```

Vários em um

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexão (dois jogadores).

Dificuldade do jogo: variável, consoante a utilização.

O JOGO:

Na verdade são possíveis vários jogos com este programa, que vai comportar-se como um lápis electrónico, que lhe permitirá inscrever símbolos de cor seja onde for da sua folha-écran de televisão. Poder-se-á, por exemplo, utilizá-lo para uma variante mais vasta do jogo do galo em que ganha o primeiro que realiza o alinhamento de cinco peças idênticas, ou para outro em que se realiza o enquadramento das peças adversas, ou para qualquer outro jogo, segundo a sua imaginação.

O PROGRAMA:

O campo de jogo é delimitado por um quadro construído pela linha 30.

A mudança de jogador é efectuada pela linha 50.

O jogador que vai jogar vê o símbolo que escolheu (um quadrado de cor) colocar-se no limite do quadro (linha 70).

As linhas 100 e 110 permitem que os dois jogadores desloquem o seu símbolo pelo campo de jogo.

As linhas 130 e 140 controlam se a deslocação é possível (casa livre e não ultrapassagem dos limites).

A linha 180 permite escrever numa casa livre.

PARA JOGAR:

A ordem RUN permite a apresentação dos limites do campo de jogo e do número do jogador, assim como dos símbolos correspondentes, que deverão ser deslocados pelo écran com a ajuda das quatro teclas de direcção (↑, ↓, →, ←).

Não é possível saltar para uma casa já ocupada, e será necessário premir a tecla COPY para inscrever o seu próprio símbolo numa casa disponível.

Uma vez isso feito, o símbolo do segundo jogador aparece na coluna da esquerda, mais ou menos ao meio, e o jogador procederá da mesma maneira que o seu adversário.

Para sair do programa basta premir ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á utilizar todo o écran como campo de jogo e realizar controlos em função dos diferentes jogos que se pretenda praticar: controlo de alinhamentos de cinco peças, controlo do número de peças capturadas, etc.

```
10 REM VARIOS EM UM
20 MODE 1
30 MOVE 8,8:DRAWR 0,24*16,1:DRAWR 25*16,
0,1:DRAWR 0,-24*16,1:DRAWR -25*16,0,1
40 J=0
50 J=ABS(J-1)
60 X=13:Y=1
70 LOCATE Y,X:PEN 2-(J=1):PRINT CHR$(231
-28*J)
80 A$=INKEY$:IF A$="" THEN 80
90 A=X:B=Y
100 X=X-(A$=CHR$(&F1) AND X<24)+(A$=CHR$
(&F0) AND X>2)
110 Y=Y-(A$=CHR$(&F3) AND Y<25)+(A$=CHR$
(&F2) AND Y>2)
120 LOCATE B,A:PRINT " "
130 C=TEST((Y-1)*16+8,(25-X)*16+8)
140 IF C<>2 AND C<>3 AND C<>1 THEN 160
150 X=A:Y=B
160 CA=INKEY(9)
170 IF CA<>0 OR Y=2 THEN 70
180 LOCATE Y,X:PEN 2-(J=1):PRINT CHR$(23
1-28*J)
190 IF J=0 THEN SOUND 1,200,10 ELSE SDUN
D 1,800,10
200 GOTO 50
```

Percurso do cavalo

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexão.

Dificuldade do jogo: difícil.

O JOGO:

Este jogo é aqui apresentado como sendo totalmente de acaso, mas na realidade provém de um problema bastante difícil de resolver: como deslocar o cavalo num tabuleiro de xadrez de modo que ele passe uma vez, e apenas uma vez, por cada uma das 64 casas do tabuleiro.

O cavalo desloca-se segundo as regras comumente correntes no xadrez e a sua posição de partida é definida pelo jogador; em contrapartida, as suas deslocações são feitas ao acaso. Conseguirá percorrer as 64 casas?

O PROGRAMA:

A linha 60 define as diversas deslocações possíveis, em duas variáveis alfanuméricas, A\$ e B\$; o ciclo encaixado I (70-100) desenha o campo de jogo.

Dá-se entrada às coordenadas da casa de partida (primeiro a linha, depois a coluna) nas linhas 110 e 120.

A linha 140 imprime o número da jogada.

A linha 150 permite a tiragem ao acaso de uma deslocação do cavalo.

O ciclo G (160-210) verifica se a deslocação é possível; se não for, tenta todas as outras (sempre na mesma ordem).

As linhas 170 e 180 efectuam a deslocação.

A linha 190 vela para que se não saia do campo de jogo.

A linha 200 envia para a linha 230 se a jogada é possível.

A linha 220 faz a gestão do fim de partida (cavalo bloqueado) e imprime a pontuação.

As linhas 230, 240 e 250 realizam a jogada e incrementam a pontuação.

PARA JOGAR:

Faz-se correr o programa com a ordem RUN. O campo de jogo é impresso no écran; escolhe-se então a posição de partida teclando primeiro um número de linha (entre 1 e 8), seguido de ENTER, e depois um número de coluna, seguido de ENTER. O jogo começa imediata-

mente e desenrola-se de modo completamente automático até que o cavalo fique bloqueado. Pode-se, evidentemente, parar em qualquer momento teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Podem-se imaginar numerosas variantes sobre este tema, nem que seja começando por fazer variar o tamanho do tabuleiro; pode-se também melhorar consideravelmente o grafismo. Mas o mais interessante é conseguir definir um algoritmo que dê a ou as soluções do problema inicial...

```
10 REM PERCURSO DO CAVALO
20 RANDOMIZE(TIME)
30 INK 0,9: BORDER 9
40 CLS
50 S=1:T=0
60 A$="56653223":B$="653223"+A$+"56":A$=
A$+A$
70 FOR I=1 TO 8
80 LOCATE 3*I+2,1:PRINT I
90 LOCATE 1,2*I+1:PRINT I;"
100 NEXT I
110 LOCATE 1,19:INPUT "LINHA DE PARTIDA?";I
120 PRINT:INPUT "COLUNA DE PARTIDA?";J
130 IF S>9 THEN S=0:T=T+1
140 LOCATE 3*J+3,2*I+1:PAPER 2:PEN 3:PRINT
RIGHT$(STR$(S),1):IF T<>0 THEN LOCATE
3*J+2,2*I+1:PRINT RIGHT$(STR$(T),1)
150 Z=INT(RND*8+1)
160 FOR G=Z TO Z+8
170 A=I+VAL(MID$(A$,G,1))-4
180 B=J+VAL(MID$(B$,G,1))-4
190 IF A<1 OR B<1 OR A>8 OR B>8 THEN 210
200 IF TEST(((3*B+3)-1)*16+8,(25-(2*A+1)
)*16+8)=0 THEN 230
210 NEXT G
220 PAPER 1:PEN 3:LOCATE 1,23:PRINT "TOT
AL :";T*10+S:SOUND 1,478:SOUND 1,426:SQU
ND 1,379:PAPER 0:PEN 1:END
230 I=A:J=B
240 S=S+1:PRINT CHR$(7);
250 GOTO 130
```

«Othello»

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexão (dois jogadores).

Dificuldade do jogo: difícil.

O JOGO:

Tendo-se tornado um grande clássico dos jogos para microcomputador, o jogo «Othello» ou «Reversi» só tem a ganhar com a visualização, mesmo se (como acontece aqui) é jogado por dois jogadores; efectivamente, como o jogo consiste no derrube dos peões do adversário, isto pode tornar-se monótono a longo prazo, sobretudo se você é um jogador assíduo. O princípio é que o jogador que tem os X deve, para poder jogar, conseguir enquadrar uma ou várias peças do seu adversário; se conseguir, então os O serão derrubados e tornar-se-ão X (o derrube pode ser feito horizontalmente, verticalmente ou segundo as diagonais).

O jogador que tem os O procederá do mesmo modo, e isto prossegue até que o tabuleiro esteja cheio ou que ninguém possa mais jogar (se um jogador não pode derrubar peões, deve passar a sua vez).

O PROGRAMA:

Os ciclos encaixados P (50-120) e Q (70-110) desenham o campo de jogo e a linha 140 imprime um X ou um O, conforme o jogador que comece.

O ciclo P (200-320) controla a validade da jogada (a casa está livre: linhas 300 e 310), utilizando uma parte de programa que se encontra nas linhas 360 a 410 (um ou vários inimigos tomados na linha 360 e eventual derrube autorizado pela linha 380: volta a fazer-se um varrimento em todas as direcções, se o enquadramento se revelou possível).

A linha 200 define as oito direcções de exploração.

As linhas 140 e 250 calculam as deslocações a efectuar para explorar essas oito direcções.

A linha 230 realiza as deslocações.

A linha 340 assegura a mudança de jogador, se houve derrube (controlado na linha 330).

PARA JOGAR:

Depois de se ter feito RUN, os jogadores farão entrar as coordenadas, cada um na sua vez (linha, ENTER, depois coluna, ENTER), da casa para onde desejam jogar; se uma jogada não é aceite, faz-se entrar 0 como primeira coordenada para a mudança de jogador.

EXTENSÕES POSSÍVEIS:

Antes de mais, evidentemente, melhorar a apresentação com cores, uma grelha mais arejada.

Poder-se-á também imaginar um jogo-máquina para o AMS-TRAD.

Depois de se ter feito correr o programa com RUN, convém fazer entrar a dimensão da grelha (4, ENTER, ou 6, ou 8...).

```
10 REM "OTHELLO"
15 WINDOW #1,25,40,8,18
20 INPUT "DIMENSÃO DA GRELHA :";A
30 IF A MOD 2<>0 OR A>8 OR A<2 THEN 20
40 CLS
50 FOR P=1 TO A
60 LOCATE 2*P+1,1:PRINT RIGHT$(STR$(P),1
)
70 FOR Q=1 TO A
80 LOCATE 1,2*P+1:PRINT RIGHT$(STR$(P),1
):LOCATE 2*Q+1,2*P+1:PRINT CHR$(144)
90 IF Q=A/2+1 AND P=A/2 THEN LOCATE 2*Q-
1,2*P+1:PEN 2:PRINT CHR$(231);" ";:PEN 3
:PRINT CHR$(203):PEN 1
100 IF Q=A/2+1 AND P=A/2+1 THEN LOCATE 2
*Q-1,2*P+1:PEN 3:PRINT CHR$(203);" ";:PE
N 2:PRINT CHR$(231):PEN 1
110 NEXT Q
120 NEXT P
130 K=1
140 CLS #1:LOCATE #1,2,3:PEN #1,2-(K=1):
PRINT #1,"JOGADOR ";CHR$(231-K*28):SOUND
1,200+200*K,10
160 PRINT #1:INPUT #1,"LINHA :";U
170 IF U=0 THEN 340
180 PRINT #1:INPUT #1,"COLUNA :";V
190 H=1
200 FOR P=1 TO 8
210 S=0
220 W=0
230 T=2*U+1:R=2*V+1
240 F=(P>3 AND P<7)-(P<3 OR P>7)
250 G=(P>1 AND P<5)-(P=6)
260 Q=-203*(S=1)-231*(S=2)-144*(S=0)
270 IF S=0 THEN 290
280 LOCATE R,T:PEN 2-(K=1):PRINT CHR$(Q)
290 T=T+2*F:R=R+2*G
300 C=TEST((R-1)*16+8,(25-T)*16+8)
310 IF C=2 OR C=3 THEN 360
```

```
320 NEXT P
330 IF H<>0 THEN 140
340 K=ABS(K-1)
350 GOTO 140
360 IF (C=2 AND K=1) OR (C=3 AND K=0) TH
EN W=1
370 IF S<>0 THEN LOCATE R,T:FN 2-(K=1):
PRINT CHR$(Q)
380 IF W=1 AND ((C=2 AND K=1) OR (C=3 AN
D K=0)) THEN 290
390 IF W=0 OR S<>0 THEN 320
400 S=1-(K=0)
410 H=-H*(S=0)
420 GOTO 220
```

Damas

REFERÊNCIAS:

Dificuldade de programação: muito difícil.

Categoria de jogo: jogo de reflexão (dois jogadores).

Dificuldade do jogo: difícil.

O JOGO:

É o jogo clássico de damas, mas um tanto incompleto porque não prevê a possibilidade de fazer dama. As regras do jogo são as clássicas, mas os controlos foram um pouco simplificados. Os movimentos das peças são feitos entrar numa única manobra (posição de partida e direcção de deslocação) e a tomada de pedras, se for possível, é realizada automaticamente.

Com a mudança de jogador é realizada em cada jogada, será necessário eventualmente premir 0 para poder voltar a jogar, se houver salto ou se a jogada não era possível.

O PROGRAMA:

Os ciclos I (80-170) e J (90-160) desenham o tabuleiro e definem as posições iniciais das pedras brancas e das pedras pretas. As posições estão um quadro R(X,Y) e a cor num outro quadro, C(X,Y).

A impressão das pedras propriamente dita é efectuada pelos ciclos encaixados J (190-250) e I (200-240).

A linha 270 indica qual jogador tem a vez.

A jogada é feita entrar na linha 270.

A linha 280 realiza a mudança de jogador (tecleia-se 0).

A linha 300 verifica se a deslocação é possível.

A linha 330 trata o caso de uma deslocação e a linha 340 o caso de um salto.

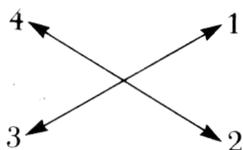
A linha 380 assegura a mudança de jogador.

PARA JOGAR:

Depois de se fazer correr o programa com RUN, o tabuleiro, com as suas pedras, é desenhado no écran e aparece um círculo de cor por baixo, para indicar a que jogador pertence a vez de jogar.

Joga-se indicando, com uma palavra de três letras (palavra que

pode começar por zero), a linha e a coluna da peça a deslocar e a direcção da deslocação (codificada de 1 a 4), segundo a convenção:



Por exemplo, 9D1 significa que a pedra situada na linha 9, coluna D, se desloca na direcção 1.

Do mesmo modo, 0G3 significará pedra negra, coluna G, direcção 3.

Como a mudança de jogador é efectuada automaticamente, será necessário teclar 0 para voltar a jogar se a jogada for recusada ou se foi «comida» uma pedra.

O programa não controla a chegada à posição de dama e não proíbe a deslocação para trás.

EXTENSÕES POSSÍVEIS:

Será bastante fácil começar por melhorar os controlos (mudança de jogador, jogada para trás que não corresponda à tomada de uma peça, saída do tabuleiro. Poder-se-á encarar a criação de damas, sua deslocação, e eventualmente escrever um programa que permita ao AMSTRAD juntar-se às suas apaixonantes partidas com possibilidades de vencer.

```
10 REM DAMAS
20 DIM R(10,10):DIM C(10,10):SYMBOL AFTE
R 240:SYMBOL 240,0,0,0,0,0,0,0,0,0:SYMBOL
241,0,60,195,195,189,129,66,60
30 A#=STRING$(3,140)
40 B#=STRING$(3,143)
50 C#=STRING$(3,131)
60 CLS
70 PRINT "      A B C D E F G H I
J"
80 FOR I=1 TO 10
90 FOR J=1 TO 10
100 IF (I+J) MOD 2=0 THEN 120
110 LOCATE 3*J+1,2*I:PRINT A#:PRINT I-1:
LOCATE 3*J+1,2*I+1:PRINT B#:LOCATE 3*J+1
,2*I+2:PRINT C#
120 R(I,J)=0
130 IF (I+J) MOD 2=1 THEN 160
```

```

140 IF J<5 THEN R(I,J)=1:C(I,J)=2
150 IF J>6 THEN R(I,J)=1:C(I,J)=3
160 NEXT J
170 NEXT I
180 H=1
190 FOR J=1 TO 10
200 FOR I=1 TO 10
210 IF (I+J) MOD 2<>0 THEN 240
220 X=3*J+2:Y=2*I+1
230 LOCATE X,Y:PEN C(I,J):PRINT CHR$(240
+R(I,J))
240 NEXT I
250 NEXT J
260 LOCATE 1,23:PRINT CHR$(20);
270 LOCATE 1,23:PEN 2-(H=1):PRINT "JOGAD
OR ";CHR$(241);:INPUT " - SUA JOGADA? ";
D$
280 IF D$="0" THEN H=ABS(H-1):GOTO 260
290 GOSUB 400
300 IF R(L,C)=0 OR (C(L,C)=2 AND H=1) OR
(C(L,C)=3 AND H=0) THEN 260
310 T=(1 AND (D=1 OR D=2))-(1 AND (D=4 O
R D=3))
320 S=(1 AND (D=3 OR D=2))-(1 AND (D=1 O
R D=4))
330 IF R(L+S,C+T)=0 THEN R(L,C)=0:R(L+S,
C+T)=1:C(L+S,C+T)=2-(H=1):GOTO 380
340 IF C(L+S,C+T)=2-(H=0) AND R(L+S,C+T)
=1 AND R(L+2*S,C+2*T)=0 THEN 360
350 GOTO 260
360 R(L,C)=0:R(L+S,C+T)=0:R(L+2*S,C+2*T)
=1:C(L+2*S,C+2*T)=2-(H=1)
370 GOTO 190
380 H=ABS(H-1)
390 GOTO 190
400 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2
,1))-64:D=VAL(RIGHT$(D$,1))
410 RETURN

```

Quatro a quatro

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexão (um ou dois jogadores).

Dificuldade do jogo: difícil.

O JOGO:

Poderá ser jogado por um ou vários jogadores. Nos dois casos, o objectivo é conseguir trocar os peões X com os peões O; se jogado como solitário, cada mudança deverá ser realizada com o mínimo de jogadas possível.

Os peões são deslocados alternadamente, respeitando as regras de deslocação do cavalo no xadrez; não há tomada de pedras e não se pode passar de vez sem jogar.

O PROGRAMA:

A variável alfanumérica A\$, na linha 40, contém a tabela das jogadas possíveis (salto do cavalo).

O ciclo I (60-100) desenha o campo de jogo e as posições de partida.

A linha 120 permite imprimir X ou O, consoante a vez do jogador.

A linha 150 permite a impressão do cursor a piscar, que indica em que casa se encontra.

As linhas 180 e 190 permitem a deslocação do cursor.

A linha 200 verifica se foi premida uma tecla de 1 a 8, dando a direcção da deslocação.

A linha 210 reimprime a figura que estava sob o cursor. A linha 240 transforma em número o símbolo numérico teclado. A linha 230 realiza o controlo de início de partida.

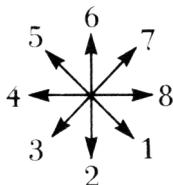
As linhas 250 e 260 efectuam a deslocação. A linha 240 realiza o controlo da chegada. O resultado é impresso pela linha 300, a mudança de jogador é realizada pela linha 310 e um «.» é impresso no local de partida. A rotina 330-340 é utilizada para os diversos controlos.

PARA JOGAR:

Faz-se correr o programa com RUN. O campo de jogo é impresso no écran, assim como o símbolo que permite identificar qual o jogador que está na vez de jogar. O jogador pode escolher o peão que deseja

deslocar manobrando o cursor a piscar graças às teclas de deslocação do cursor.

Para deslocar um peão basta, quando o cursor está sobre esse peão, teclar a direcção da deslocação, de acordo com o código:



A jogada será recusada se a casa de partida não está conforme ou se a casa de chegada está ocupada. Caso contrário, a jogada será realizada e a vez passa ao outro jogador, que desloca o cursor a piscar para o posicionar sobre um dos peões, e joga do mesmo modo.

Ganha o jogador que conseguir ser o primeiro a levar todos os seus peões até às casas inicialmente ocupadas pelo adversário. Pode-se interromper a partida em qualquer momento teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se melhorar os controlos, prever um teste de fim, utilizar um campo de jogo diferente; poder-se-á também pensar na escrita de um programa que permita ao seu AMSTRAD participar no jogo com boas hipóteses de o ganhar.

```
10 REM QUATRO A QUATRO
20 INK 0,15: BORDER 15
30 MODE 0
40 H=1:A$="5665322356"
50 LOCATE 5,2: PRINT 1 2 3 4 5"
60 FOR I=1 TO 5
70 LOCATE 1,3*I+3: PRINT RIGHT$(STR$(I),1
): LOCATE 5,3*I+3: PRINT CHR$(144)+" "+CHR
$(144)+" "+CHR$(144)+" "+CHR$(144)+" "+C
HR$(144)
80 IF I<3 THEN PEN 2: LOCATE 5,3*I+3: PRIN
T CHR$(231);" ";CHR$(231): PEN 1
90 IF I>3 THEN PEN 3: LOCATE 11,3*I+3: PRI
NT CHR$(203);" ";CHR$(203): PEN 1
100 NEXT I
110 X=1: Y=1
120 LOCATE 5,22: PEN 2+H: PRINT "JOGADOR "
;CHR$(231-28*H)
```

```

130 GOSUB 330
140 I=J
150 LOCATE 3+2*X,3+3*Y:PRINT " "
160 A=X:B=Y
170 R#=INKEY$
180 X=X-(R#=CHR$(&F3) AND X<5)+(R#=CHR$(&F2) AND X>1)
190 Y=Y-(R#=CHR$(&F1) AND Y<5)+(R#=CHR$(&F0) AND Y>1)
200 IF R#="9" AND R#>"0" THEN 230
210 LOCATE 3+2*A,3+3*B:PEN J:IF J=1 THEN PRINT CHR$(144) ELSE PRINT CHR$(231-28*(J-2))
220 GOTO 120
230 IF (I=1) OR (I=2 AND H=1) OR (I=3 AND H=0) THEN GOTO 210
240 K=VAL(R#)
250 S=Y:Y=Y+VAL(MID$(A$,K,1))-4
260 T=X:X=X+VAL(MID$(A$,K+2,1))-4
270 IF X<1 OR X>5 OR Y<1 OR Y>5 THEN Y=S:X=T:A=T:B=S:GOTO 210
280 GOSUB 330
290 IF J<>1 THEN X=-1:J=2+H:GOTO 270
300 LOCATE 3+2*X,3*Y+3:PEN (2+H):PRINT CHR$(231-28*H)
310 H=ABS(H-1):I=1
320 GOTO 210
330 J=TEST ((3+2*X-1)*32+16,(25-(3+3*Y))*16+8)
340 RETURN

```

Xadrez

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: jogo de reflexão (dois jogadores).
Dificuldade do jogo: difícil.

O JOGO:

O programa propõe apenas a dois jogadores a disputa de uma partida servindo-se do écran do televisor como tabuleiro de jogo. Não há controlo de validade das jogadas, mas unicamente gestão da posição das peças no tabuleiro. As pedras são desenhadas com a ajuda da instrução SYMBOL. A cor permite distinguir cada um dos campos.

O PROGRAMA:

As linhas 110 a 200 desenharam o tabuleiro. As linhas 150, 160, 170, 180, 210 e 220 definem as condições iniciais, com a ajuda de dois quadros: R(I,J) para os códigos das peças e C(I,J) para a cor.

Os ciclos encaixados I (250-280) e J (240-290) desenharam as peças.

A linha 320 permite fazer entrar a linha e a coluna de partida, com controlo na linha 330.

A linha 360 permite fazer entrar a coluna e a linha de chegada.

A linha 380 toma nota da nova posição.

A linha 390 assegura a mudança de jogador. Para parar basta teclar ESC duas vezes.

PARA JOGAR:

Faz-se correr o programa com RUN.

O tabuleiro de xadrez é desenhado, depois as peças e por fim um texto de cor indica qual o jogador que tem a vez de jogar; faz-se entrar a coluna e a linha de partida, sob a forma E7, por exemplo, e dá-se a coluna e a linha de chegada. A deslocação é então efectuada, e é a vez do outro jogador.

Para parar basta teclar ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Poder-se-á procurar realizar um grafismo mais preciso, introduzir o controlo do movimento das peças. E, por que não?, escrever um programa que permita ao AMSTRAD comportar-se como opositor à altura. Coragem!

```

10 REM XADREZ
20 DIM R(8,8):DIM C(8,8):SYMBOL AFTER 24
0
30 SYMBOL 240,0,16,40,68,108,56,124,0:SY
MBOL 241,0,16,56,16,56,68,56,0
40 SYMBOL 242,0,84,124,56,56,124,124,0:S
YMBOL 243,0,84,40,16,108,124,124,0
50 SYMBOL 244,0,0,16,56,56,16,124,0:SYMB
OL 245,0,16,56,120,24,56,124,0:SYMBOL 24
6,0,0,0,0,0,0,0,0
60 A$=STRING$(3,140)
70 B$=STRING$(3,143)
80 C$=STRING$(3,131)
90 CLS:INK 1,9:INK 3,0
100 PRINT "  A  B  C  D  E  F  G  H"
110 FOR I=1 TO 8
120 FOR J=1 TO 8
130 IF (I+J) MOD 2=0 THEN 150
140 LOCATE 3*J,2*I:PRINT A$:PRINT CHR$(4
8+I):LOCATE 3*J,2*I+1:PRINT B$:;LOCATE 3
*J,2*I+2:PRINT C$
150 IF I=2 OR I=7 THEN R(I,J)=4
160 IF I<>2 AND I<>7 THEN R(I,J)=6
170 C(I,J)=2
180 IF I=7 OR I=8 THEN C(I,J)=3
190 NEXT J
200 NEXT I
210 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:
R(1,5)=1:R(1,6)=0:R(1,7)=5:R(1,8)=2
220 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
230 H=0
240 FOR J=1 TO 8
250 FOR I=1 TO 8
260 X=1+3*J:Y=2*I+1
270 LOCATE X,Y:PEN C(I,J):PAPER (I+J) MO
D 2:PRINT CHR$(240+R(I,J))
280 NEXT I
290 NEXT J
300 WINDOW #1,28,40,9,18
310 CLS #1:PEN #1,2+H:PRINT #1," JOGADOR
";;IF H=0 THEN PRINT #1,"AZUL" ELSE PRI
NT #1,"NEGRO"
320 PRINT #1,"PARTIDA (C,L)":LOCATE #1,7
,3:INPUT #1,"",D$
330 GOSUB 410
340 IF R(L,C)=6 OR C(L,C)=3-H THEN 310
350 T=R(L,C):S=C(L,C):R(L,C)=6
360 LOCATE #1,1,7:PRINT #1,"CHEGADA (C,L
)":LOCATE #1,7,8:INPUT #1,"",D$

```

```
370 GOSUB 410
380 R(L,C)=T:C(L,C)=S
390 H=ABS(H-1)
400 GOTO 240
410 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1))-64
420 RETURN
```

Muro de tijolos

REFERÊNCIAS:

Dificuldade de programação: difícil.

Categoria de jogo: jogo de reflexos (um jogador).

Dificuldade do jogo: difícil.

O JOGO:

Trata-se de um jogo que se pode apresentar como um grande clássico dos jogos de arcade para microcomputadores.

É aqui apresentado numa versão simplificada, mas oferece mesmo assim boas *performances*: afixação da pontuação no fim da partida, ressaltos aleatórios, etc.

A regra é muito elementar: você dispõe de uma raqueta que pode ser deslocada horizontalmente para devolver uma bola que vai demolir os tijolos desenhados na parte de cima do écran.

O seu problema é demolir a maior parte possível do muro, sem faltar a bola, o que provocará o fim da partida e a apresentação da pontuação.

O PROGRAMA:

O ciclo I (50-80) desenha o campo de jogo e os tijolos são postos no lugar pelos ciclos encaixados I (90-130) e J (100-120).

A linha 200 desenha a raqueta, cuja posição é controlada pela linha 180.

As linhas 250 a 270 verificam se foram atingidos o muro, os tijolos ou a raqueta.

Os ressaltos são gerados em consequência, com eventuais saltos aleatórios, pelas linhas 250 e 270.

A linha 210 apresenta permanentemente a pontuação.

PARA JOGAR:

O programa será posto a trabalhar com a instrução RUN, o que provoca a impressão do campo de jogo e depois, muito rapidamente, o envio da bola. O jogador deverá então tentar deslocar a sua raqueta com a ajuda das teclas de deslocação horizontal (←→).

Se, em qualquer momento, a bola não tocar na raqueta, o jogo termina.

Pode-se também terminar quando quisermos, teclando ESC duas vezes.

EXTENSÕES POSSÍVEIS:

Pode-se pensar em ampliar o campo de jogo, aumentar o número de tijolos, prever várias bolas, introduzir efeitos sonoros.

Pode-se também, com base em jogos de arcade, introduzir pequenos trabalhadores que virão reconstruir o muro de tijolos, à medida que você o for demolindo.

```
10 REM MURO DE TIJOLOS
20 DEF FN IN(X)=16*ROUND(X/16)
30 RANDOMIZE(TIME)
40 CLS:PEN 1:A#=CHR$(143):B#=CHR$(233)
50 FOR I=1 TO 18
60 LOCATE 10+I,1:PRINT A#
70 LOCATE 11,I+1:PRINT A#:LOCATE 28,I+1:
PRINT A#
80 NEXT I
90 FOR I=1 TO 6
100 FOR J=11 TO 26
110 LOCATE J+1,I+1:PEN 2+(I+J) MOD 2:PRI
NT B#
120 NEXT J
130 NEXT I
140 TAG
150 ORIGIN 11*16,7*16-1:I=0:J=8*16:K=16:
H=16:Y=6*16
160 A=I:B=J
170 A#=INKEY$
180 Y=Y-16*(A#=CHR$(&F3) AND Y<15*16)+16
*(A#=CHR$(&F2) AND Y>-4*16)
190 IF T=1 THEN MOVE IJ,II:PRINT " ";T=
0:I=A:J=B
200 MOVE Y,-16:PRINT " "+STRING$(3,143)+
" ";
210 LOCATE #1,1,22:PRINT #1,"PONTUACAD:"
;S
220 A=I+H:B=J+K:IA=FN IN(A):IB=FN IN(B):
II=FN IN(I):IJ=FN IN(J)
230 IF II<0 THEN TE=0:WHILE TE<200:SOUND
1,TE,1:TE=TE+1:WEND:END
240 C=TEST(IB+8,IA+8)
250 IF C=1 THEN H=H+2*H*(IA>16*16 OR IA<
0):K=K+2*K*(IB>15*16 OR IB<0):SOUND 1,44
0,1:IF RND<0.5 THEN 280 ELSE GOTO 170
260 IF C=0 THEN MOVE IB,IA:PRINT CHR$(23
3) ;T=1:GOTO 170
270 IF C=2 OR C=3 THEN MOVE IB,IA:PRINT
CHR$(231) ;H=-H:K=K*(1+2*(RND<0.5)):T=1:S
=S+C:SOUND 1,200,6
280 K=SGN(K)*(16+10*(RND<0.5))
290 GOTO 170
```

O lagarto de Alice

REFERÊNCIAS:

Dificuldade de programação: difícil.
Categoria de jogo: solitário.
Dificuldade do jogo: bastante difícil.

O JOGO:

Inspira-se vagamente numa ideia de Lewis Carroll, em *Alice no País das Maravilhas*¹. Na história original, Alice cresce ou diminui, ao passo que no jogo, por razões de ocupação da memória, é um lagarto que muda de tamanho.

No écran aparecem dois copos: um contém o líquido que faz crescer, o outro o líquido que faz diminuir (infelizmente não é sempre o mesmo copo). O jogador deverá indicar a quantidade de líquido que deseja beber e o lagarto mudará de tamanho em função dessa quantidade. O objectivo do jogo é conseguir devolvê-lo ao seu tamanho inicial.

O PROGRAMA:

O ciclo I (50-80) desenha os copos.

A rotina 430 é chamada se a resposta à pergunta QUANTIDADE? não for suficientemente rápida. A linha 220 anula essa chamada.

O ciclo I (230-250) volta a encher os copos com a quantidade indicada na linha 210.

A variação de estatura do lagarto é definida pela linha 290, com controlo pelas linhas 300-310.

O ciclo I (320-340) desenha o corpo do lagarto, as linhas 350 e 360 desenham as patas, a linha 370 a cabeça; os dois ciclos separados I (380) e I (390) formam a boca e os dentes.

A linha 410 imprimirá uma mensagem BRAVO! se conseguir encontrar o tamanho correcto.

PARA JOGAR:

Faz-se trabalhar o programa com RUN. Os dois copos são desenhados, depois aparece a pergunta QUE LADO?, à qual se responde tecendo E ou D; obtém-se a impressão da pergunta QUANTIDADE?; faz-se então entrar uma quantidade (de 0 a 200) e prime-se ENTER.

¹ Editado por Publicações Europa-América, na col. «Livros de Bolso», n.º162. (N. do E.)

O copo escolhido enche-se diante de si e, momentos depois, é apresentado o lagarto. Deve então premir-se uma tecla qualquer; se o lagarto não tiver o tamanho correcto não será impressa qualquer mensagem e volta-se ao desenho dos copos.

EXTENSÕES POSSÍVEIS:

Dado que se trata de um jogo essencialmente gráfico, será interessante melhorar os desenhos e a cor, modificar o carácter aleatório da escolha dos copos, etc.

Por que não reler *Alice no País das Maravilhas*? As ideias não faltam aí.

```
10 REM O LAGARTO DE ALICE
20 RANDOMIZE(TIME)
30 MODE 0:N=33
40 PAPER 5
50 FOR I=1 TO 9
60 LOCATE 1,I+4:PRINT " ":LOCATE 8,I+4:P
RINT " "
70 LOCATE 13,I+4:PRINT " ":LOCATE 20,I+4
:PRINT " "
80 NEXT I
90 FOR I=1 TO 8:LOCATE I,14:PRINT " ":NE
XT I
100 FOR I=13 TO 20:LOCATE I,14:PRINT " "
:NEXT I
110 PAPER 0:PEN 3:LOCATE 2,15:PRINT "ESQ
UERDO":PEN 8:LOCATE 14,15:PRINT "DIREITO
"
120 PEN 1:LOCATE 1,17:PRINT "QUE LADO?"
130 A#=INKEY#:A#=UPPER$(A#):IF A#<>"D" A
ND A#<>"E" THEN 130
140 U=(1-12*(A#="D"))*32:C=3-5*(A#="D")
150 PEN C:LOCATE 12,17:IF A#="D" THEN PR
INT "DIREITO" ELSE PRINT "ESQUERDO"
160 PEN 1:PRINT:PRINT "QUANTIDADE?":PEN
:C:F=0:R#=""
170 AFTER 100,0 GOSUB 430
180 A#=INKEY#:IF F=1 THEN GOTO 260
190 IF A#="" THEN GOTO 180
200 IF A#<>CHR$(13) THEN R#=R#+A#:PRINT
A#;GOTO 180
210 Q=ABS(VAL(R#))
220 Z=REMAIN(0)
230 FOR I=1 TO 8*SQR(Q)
240 MOVE U,12*16+I:DRAWR 6*32-2,0,C
```

```

250 NEXT I:SOUND 1,478:SOUND 1,426:SOUND
1,379
260 TE=0:WHILE TE<600:TE=TE+1:WEND
270 CLS
280 H=1+2*(RND<0.5)
290 N=N+H*INT(SQR(Q))
300 IF N<12 THEN N=12
310 IF N>60 THEN N=60
320 FOR I=1 TO 10*N
330 MOVE I+32,195:DRAWR 0,10,12
340 NEXT I
350 TAG:MOVE 32,192:PRINT CHR$(153);
360 MOVER 10*N-64,0:PRINT CHR$(153);
370 MOVE 32,220:PRINT CHR$(224);:TAGOFF
380 FOR I=0 TO 1:MOVE 0,212+2*I:DRAWR 32
,I:NEXT I
390 FOR I=1 TO 4:MOVE 7*I,211:DRAWR 0,-2
:NEXT I
400 A#=INKEY$:IF A#="" THEN 400
410 IF N=28 THEN LOCATE 8,23:PRINT "BRAV
O!":SOUND 1,478:SOUND 1,379:SOUND 1,319:
END
420 CLS:GOTO 40
430 LOCATE 2,23:PEN 2:PRINT "DEMASIADO T
EMPO, JOGO EU"
440 Q=INT(RND*300+10):F=1
450 RETURN

```



«O melhor romance jamais escrito
acerca da II Guerra Mundial»

Time

NOVA
EDICÃO

OS NUS E OS MORTOS

NORMAN
MAILER



Retrato vivo dos horrores da guerra e dos seus efeitos nos homens que nela lutam. Um romance vivo, avassalador, que arrebatava em inúmeros combates e desvenda o lado humano dos heróis obscuros da campanha do Pacífico.



Do mesmo autor de
Noites Antigas e *O Canto do Carrasco*

À VENDA NAS MELHORES LIVRARIAS E PAPELARIAS



EUROPA-AMÉRICA ...a memória no futuro

Se te sentes só, num mundo de estranhos,
e anseias por alguém que ainda não conheces ...
Se as nuvens onde voas, num céu azul de esperança,
em cinzento se tornam e te sentes cair ...
Se sentes, mesmo assim, que queres abrir as asas
e recuperar o voo que, para ti, é tudo ...

Em A PONTE PARA A ETERNIDADE
encontrarás uma mensagem de amor

RICHARD
BACH

Autor de
«NÃO HÁ LONGE NEM DISTÂNCIA»
e «HISTÓRIA DE FERNÃO CAPELO GAIVOTA»

A Ponte para a Eternidade

2^a
Edição



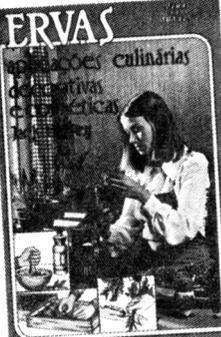
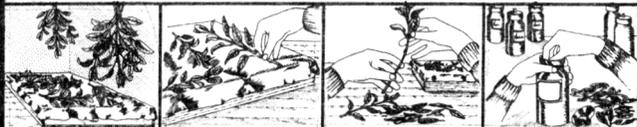
uma
mensagem
de amor



EUROPA-AMÉRICA ... a memória no futuro

BELEZA e ECONOMIA

DOIS LIVROS QUE AJUDAM
A TRANSFORMAR UMA CASA NUM LAR



ERVAS

Decorativas, perfumantes, saborosas, cosméticas. Que sabe sobre elas? Neste livro de Jack Harvey encontra muita informação pormenorizada sobre a utilização das plantas cujas raízes, caules, folhas, flores e sementes servem para dar mais encanto a um lar, ou para perfumar e embelezar e dar mais sabor às refeições.

Um livro que lhe dá «informação» e ensinará «como fazer». Como cultivar, secar e conservar ervas, como utilizá-las na decoração, na culinária, na preparação de perfumes e cosméticos.



A CONSERVAÇÃO DE ALIMENTOS

Pamela Dotter traz-nos um guia prático sobre os métodos caseiros de conservação de alimentos: confecção de compotas, manteigas, queijos e sumos de frutas, conservas de frutos e vegetais, frutas cristalizadas, molhos, temperos, congelação e secagem de alimentos. Tem igualmente conselhos acerca dos utensílios e ingredientes necessários. Pormenorizadas instruções tornam este livro ideal para principiantes. E não só.



Muitas dezenas de ilustrações a preto e a cores
À VENDA NAS LIVRARIAS E BONS ESTABELECIMENTOS DIETÉTICOS



EUROPA-AMÉRICA ... a memória no futuro

O objectivo deste livro é ensinar distraído.
Estes 102 programas de jogos guiá-lo-ão progressivamente na exploração do BASIC Amstrad.

Assimilará novas instruções nível a nível, até conseguir o domínio do seu CPC 464.

Todos os programas contidos neste livro funcionam igualmente no CPC 664 e no CPC 6128.

O livro está dividido em cinco níveis e cada nível começa com uma apresentação pedagógica do conjunto de instruções utilizadas.

Todos os jogos são descritos e os programas são analisados linha por linha, a fim de facilitar a sua modificação.

Torne-se mestre do seu Amstrad!

**ARTE
DE
VIVER.®**



PROGRAMS DE JOGOS PARA O
MASTROD

JACQUES DECOURT E



133111862