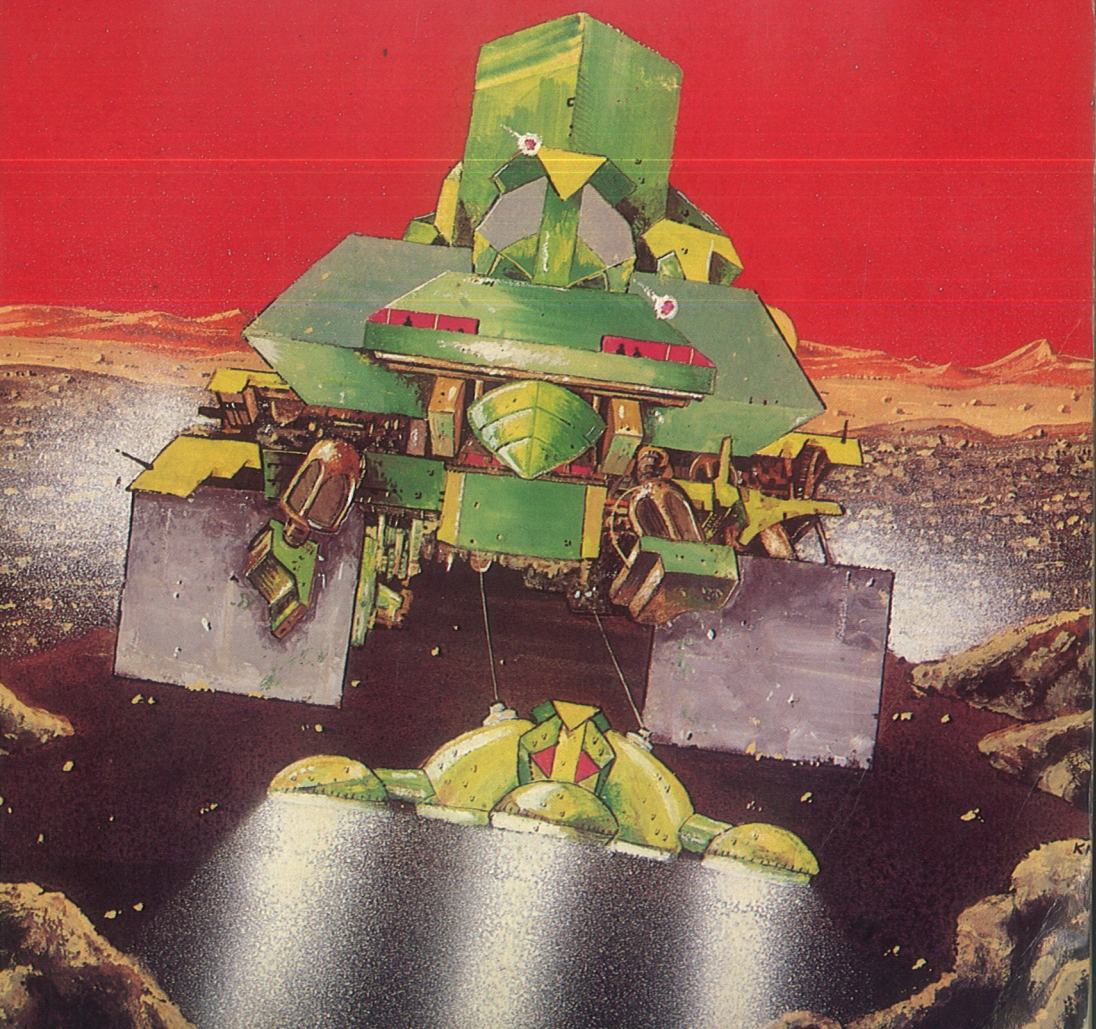


JIM GREGORY
THE
MSX
GAMES
BOOK



Other books of interest

Working with MSX BASIC

Ian Sinclair

0 00 383103 5

*Introducing MSX Assembly
Language and Machine Code*

Ian Sinclair

0 00 383146 9

The MSX Games Book

Jim Gregory

COLLINS
8 Grafton Street, London W1

Collins Professional and Technical Books
William Collins Sons & Co. Ltd
8 Grafton Street, London W1X 3LA

First published in Great Britain by
Collins Professional and Technical Books 1985

Distribution in the United States of America
by Sheridan House, Inc.

Copyright © Jim Gregory 1985

British Library Cataloguing and Publication Data

Gregory, Jim

The MSX games book.

I. Electronic games 2. Microcomputer

I. Title

794.8'028'5404 GV1469.2

ISBN 0-00-383083 7

Typeset by V & M Graphics Ltd, Aylesbury, Bucks
Printed and bound in Great Britain by
Mackays of Chatham, Kent

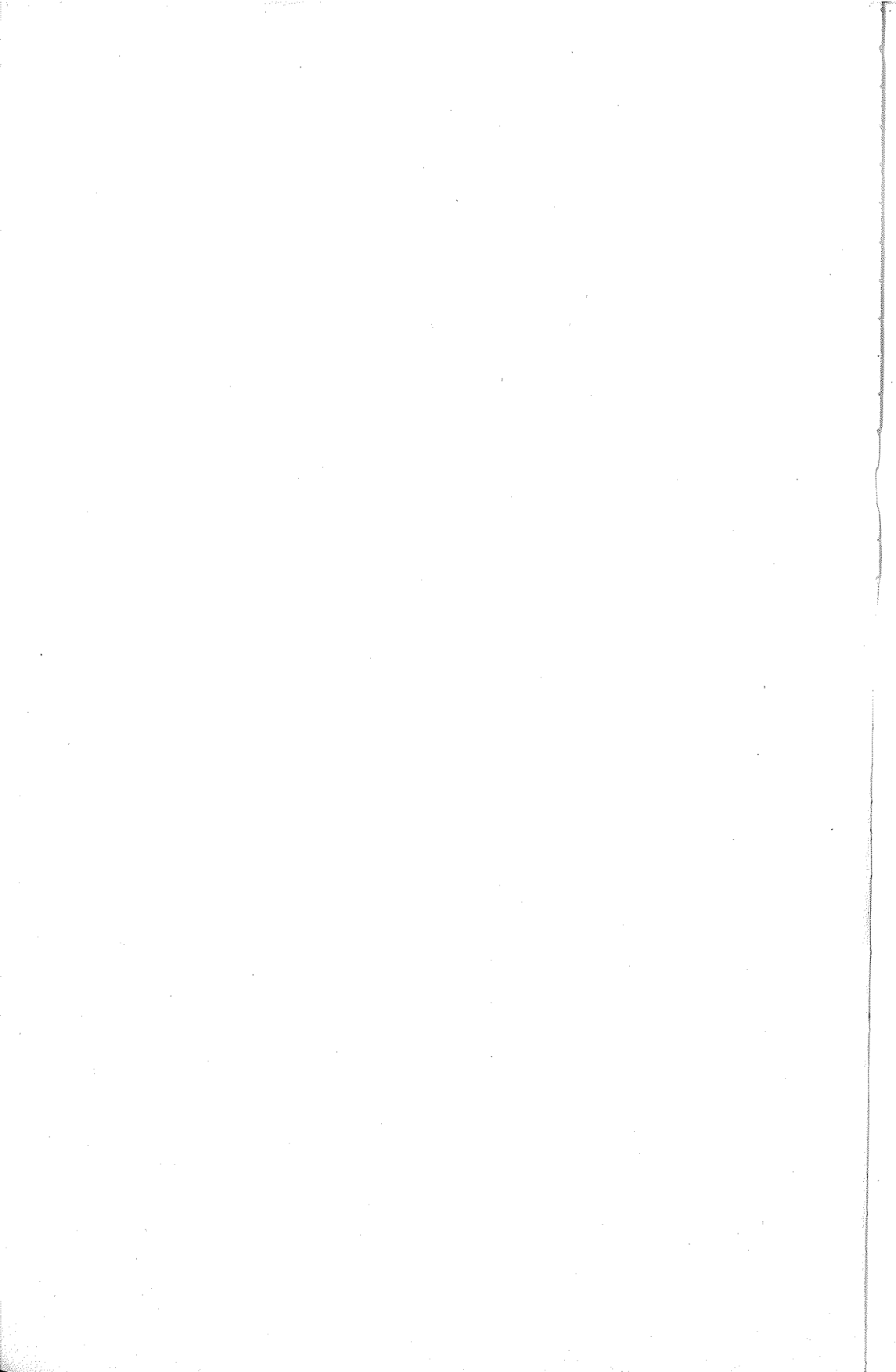
All rights reserved. No part of this publication may
be reproduced, stored in a retrieval system or transmitted,
in any form, or by any means, electronic, mechanical, photocopying,
recording or otherwise, without the prior permission of the
publishers.

Contents

<i>Preface</i>	ix
<i>Introduction</i>	1
Techniques Used in the Programs	6
Text Games	8
1 <i>Marie Celeste: A text adventure based upon discovering the legendary ghost ship.</i>	8
2 <i>Please Sir: A novel guessing game in which teacher has to say 'yes'.</i>	18
3 <i>I Accuse: A new detective game in which the criminal must be deduced from clues.</i>	23
Classic Games	28
4 <i>Star Trek: Using long and short range scanners engage the Klingons and fight to the death.</i>	28
5 <i>Rainbow Breakout: Quick reflex. game of breakout with bat size option.</i>	38
6 <i>Maze Maniac: Find your way around a 3D maze as fast as you can to catch the maniac.</i>	41
Two-player Games	47
7 <i>Duo Draughts: The traditional game with move checking and game save, load feature.</i>	47
8 <i>Chess Duel: A beautiful board layout. Move checking, game store and replay.</i>	58
9 <i>Gustavus: New implementation of a two-player spacial awareness game.</i>	74

Dice Games	78
10 Dice Derby: <i>Race along at the speed determined by the show of the dice.</i>	78
11 Snake Eyes: <i>Competitive dice game based around the 'Dicerama' subroutines.</i>	83
12 Craps: <i>The Las Vegas favourite, including on screen betting.</i>	86
Card Games	90
13 Pontoon: <i>The computer banker deals the cards the bets are placed, and one or two players try to break the bank.</i>	90
14 Hi-Lo: <i>Will the next card be higher or lower? Play the odds and build up a fortune in micro money.</i>	102
15 Concentration: <i>Two players match memory power to find pairs of cards.</i>	109
Mind Games	117
16 Micro Mind: <i>Guess the colour code in ten or less, using the hints given.</i>	117
17 O'Grady Says: <i>Follow his instructions to move your figure - only when O'Grady says.</i>	122
18 Parrot: <i>Repeat the sound sequence produced by the four parrots.</i>	127
Arcade Style Games	132
19 Pick Man: <i>Guide your gobbler round the maze to collect dots.</i>	132
20 Skippy: <i>Help your kangaroo cross the busy road and jump across the river.</i>	139
21 Kinkey Dong: <i>Lead the hero against falling barrels, up ladders through three different screens.</i>	144
Educational Games	150
22 Word Splash: <i>Guess the word quickly to prevent the little chap getting wet.</i>	150
23 Sum Fun: <i>Arithmetic quiz for primary to adult.</i>	154
24 Today England: <i>Geographical quiz based around Britain. Option to develop further.</i>	158

Utility Collection	165
25 Character Builder: <i>Design your own characters and build up more complex blocks of graphics.</i>	165
26 Picture Builder: <i>Use the on-screen pixel pencil to draw directly to the screen – save and load.</i>	175
27 MSX Monitor: <i>Look into RAM, examine, modify memory or provide checksum.</i>	180
Final Words	184
Appendices	
<i>Appendix 1: Sprite/Character Design Aid</i>	187
<i>Appendix 2: Screen Layout Planner</i>	188
<i>Appendix 3: MSX Memory Map</i>	189
<i>Appendix 4: MSX Colour Table</i>	190



Preface

Microsoft Extended BASIC or MSX has made a book such as this possible.

Thanks to the intercompatibility of MSX, we know that every owner of an MSX machine can type in these programs and produce a good, enjoyable game.

Game playing is not only fun but educational. Computer literacy is first of all encouraged by the rewards of a finished working game. Familiarisation with the computer language is then developed by seeing all the syntax in use. Next, since programs seldom work first time due to typing errors, it becomes necessary to follow the logic of a program and thus learn why it works. The last two stages of programming development are modifying existing programs and the final 'I can do better than that!' stage. I sincerely hope that as a result of purchasing this book of programs and learning from the examples, you will be able to reach the stage where you can produce even better ones for yourself in MSX BASIC.

However, this book does not set out to teach or to explain. The primary objective has been to produce good programs which would be worth ten times the price of the book if they were sold separately on tape. This is a games book and we have worked very hard to include all types of games which should appeal to owners of all types of machine.

Unfortunately, although MSX BASIC is fast and includes some very useful commands to make games easier to program, BASIC sometimes needs a little help. We have therefore included short machine code routines within some programs to help produce a better end result. There is no need to worry about these, since they are simply typed in as part of a listing. The finished result will be a hybrid of BASIC and machine code which will meet our primary objective of producing a game which is playable and worth the effort of typing in.

The development of the programs contained in this book has been a team effort which required a great deal of application and dedication by all. I am pleased to record my thanks and praise for all who worked with me to finish the project. I list them as 'ISSI', 'BOOTSY', 'ANDY', whose programming skills and ability to put up with my 'guidance' have been tremendous.

The other key members of the team are Janet and my wife Val, who 'word processed' and 'word processed' until the manuscript was complete.

My last and enduring thanks go to Richard Miles of Collins for commissioning the book and guiding it through the dark passages of publishing to reach yourself – the reason for the whole exercise – and the person whom I thank most of all for purchasing it.

Jim Gregory

Introduction

Each game featured in these pages has been specially written for this book. This has resulted in a range of programs covering all types of games. These games incorporate many useful techniques.

Each listing has been printed with a width setting of 37 columns to help reduce entry errors. This means that if you use Screen \emptyset to enter the programs then a line on the page will match a line on the screen; this feature is an example of the effort which has gone into ensuring your complete success with each and every program.

After extensive testing all the listings have been produced from actual working programs. Each has been reproduced directly from the computer printout. This means that the most likely reason for a program not working is that the reader has typed it in wrong!

Speaking as someone who has entered many programs from books and magazines, I know that at some stage you will not believe that you have got it wrong. Please read the following notes on programming carefully. They will help you to avoid the most common mistakes as you enter the programs. Follow the advice given and you will find that your microcomputer hobby can be more enjoyable and less frustrating.

Although most of the following guidelines may seem obvious, each will save you time or trouble. If you are prepared and relaxed then few errors will be made.

First program yourself

1. Set up your equipment on a desk or table, make sure that you are seated properly and try to keep your back straight. Ensure that lighting is good and that the book is supported where it can be read easily.

2. Before you start a long typing session ensure that everything is plugged in!

3. Make sure that you have a good supply of blank tapes.

4. If your blank tapes have a 'leader' on them, it is a good idea to prepare them for saving by turning the hub until the brown of the tape shows.

5. To make certain everything is OK, get a long audio tape such as a C90, which will be your working tape. Next complete a 'Save and load test', e.g. type in

```
1Ø REM - TEST -  
2Ø PRINT"TEST O.K."  
3Ø GOTO 1Ø
```

or a similar short program.

Save Test, rewind the tape, then load Test, and RUN it. If this is OK, do not rewind the tape; leave it where it stopped. If all does not operate correctly then check everything and try again, until all is working. You will then be able to use this tape to store your program, either when you have finished or part way through to be continued later.

Experiment first with the MSX fast saving speed to ensure that it is reliable with your computer/tape combination.

6. When you save a part version give it a title which will help you later, such as 'V1', 'V2', etc. You may prefer to use the date and time of saving, e.g. 8.15-14/10/85.

As you save a version, remember to make a note on the cassette or insert of the title and the counter number. These will enable you to retrieve the program quickly when you wish to resume entering the program or to 'debug' your completed program.

7. Always save onto tape *before* you type RUN. The microworld is full of keen folk who lost hours of work because they typed RUN before saving. (If you have saved earlier versions as directed, at least only part will be lost.)

The reasons for losing a program in this way are various. Examples are: mistyping words so that the interpreter performs a 'new' instruction, or gets into a loop from which there is no escape except by switching off. Programs which incorporate either machine code, POKE statements or System calls are most liable to 'crash', due to incorrect values being encountered by the program.

8. When a program is finished and working it is recommended that one copy is kept on a C90 back-up tape along with other

programs. The main copy should be stored on a suitable short length cassette such as a C15. Only one program per side should be allowed, to enable it to be loaded easily and quickly when required. Remember to write the title on the label.

This two-tape system should help to prevent re-typing if a cassette is lost, damaged or develops a fault.

9. Many users become very careless about storage of cassettes and then wonder why they sometimes become faulty. If you want to keep what you've saved follow these rules:

- Identify all tapes and cases clearly.
- Keep them in cases in a rack or storage system.
- Store away from heat.
- Store away from any magnetic fields such as TV sets, computers, loudspeakers, motors, fans and very definitely telephones, all of which may damage or erase your program.
- Don't leave tapes in the deck when they are not required, and do not leave the deck switched into 'play' unless the program requires it.
- Remember to break out the plastic tabs at the back of the tape to prevent accidental erasure of the program. If ever you wish to re-record over a protected tape temporarily cover the tab hole with adhesive tape.

10. Don't spend longer than about three hours in a session. Have a break, walk about, have a shower, play a game!

Other problems

PARIS IN THE THE SPRING

Read the above out loud! You will be surprised to know that most people read it as 'Paris in the spring'. Perhaps you did too! However, it actually says 'Paris in the the spring'. The word 'the' is printed twice, but because it is not expected and because the phrase is familiar, the second 'the' is not 'seen' by the mind.

A similar problem occurs when programs are typed in from listings. You often type what you expect to see instead of what is printed. This is why the letter O is sometimes typed when it should be 'zero' (printed as 0), or S is incorrectly typed instead of the dollar symbol (printed as \$).

4 *The MSX Games Book*

A moment's lapse of concentration can lead to lines becoming jumbled as the eye jumps from one line to the next. For example:

```
1Ø PRINT"THIS IS THE START"  
2Ø PRINT"THIS IS NEXT"
```

becomes:

```
1Ø PRINT"THIS IS THE NEXT"
```

Look at the screen to check that your shifted letters have been correctly entered. Failure to do this produces classics such as:

```
1Ø PRINT 2 WHERE ARE THE QUOTES!2
```

Under no circumstances should you attempt to alter a program as it is entered. If the numbers go: 1Ø, 2Ø, 3Ø, etc., then enter them like that. Changes are best made after a program is known to be working. This applies to everything – do not be tempted to miss out lines because you think that they may be optional. Otherwise program control could attempt to go to a line number that you have changed or deleted, and crash the program.

'Rem' statements are remark statements and have been used to help you to understand the program. They should be entered as printed. Only when a working program has been produced and saved should you perhaps produce a copy which has no 'REMS'. This could save a little memory, but there is no real benefit to be obtained. On the programs listed in this book it is safe to omit all 'REMS'. This is because care has been taken not to send control to a 'REM' line. Be careful with programs from other sources – there are some authors who think that it is good practice to send control to a 'REM' line.

Spaces are always of vital importance in a computer program. Care should be taken to get them right, to enable spaces to be counted and entered accurately. Simply align a piece of paper or card with the characters above and below, and count as the edge is moved along.

The width of printout has been set to match the 37 column Screen Ø display of the computer. This should help you to check that everything is right – for example, if the listing shows a line ending with four spaces. Then your line on the screen should also have four spaces at the end. Please be particularly careful to enter each line as a complete line from the line number commands. Confusion could arise when the line includes numbers which wrap onto the next line. Do not enter these lines separately, or the computer will take it as a line number which will cause problems.

You may find it useful to place a ruler along the line you are up to. This will further reduce the risk of errors.

If you follow all the foregoing then you should not have any great problems. However, it is inevitable that some bugs will creep in. If your program fails to operate properly then the last chapter contains some helpful advice on debugging.

That's enough preparation for you, let's now start entering some programs.

Techniques Used in the Programs

Colour pointers (logical colours)

The system of colour allocation used in the MSX can help produce the illusion of very fast movement. The system works by defining each element of a picture separately with different colour pointers. Next the pointers are swapped under program control. When each colour is changed the effect on screen is instantaneous and produces the impression of movement.

Maze Maniac uses the technique to great effect. The result is a maze which changes views very rapidly. This speed could not have been achieved otherwise, unless extensive machine code routines had been used.

Exclusive oring (XOR)

This is another technique which is used in this book. The use of XOR enables information to be superimposed upon the image which is already on screen. Performing the XOR again will remove the image, leaving the background as it was before.

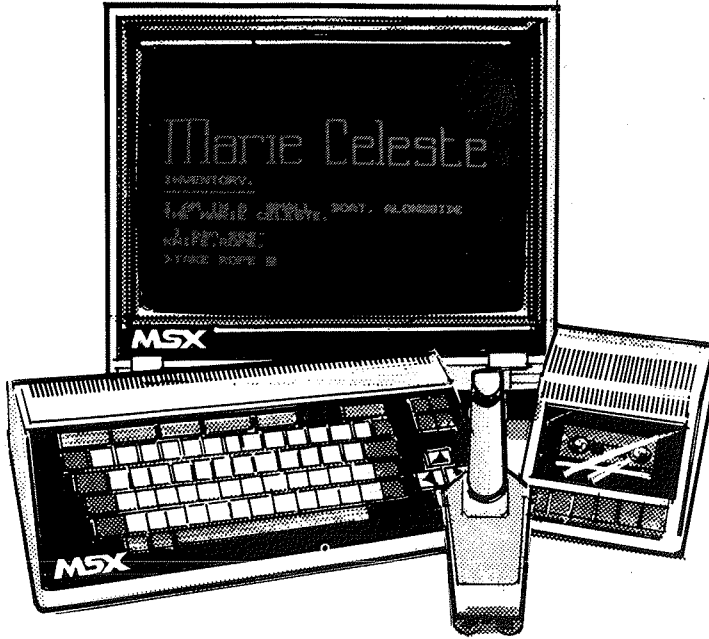
Examples of the commands required to implement this powerful feature can be found throughout the listings. The *Chess Duel* program contains some superb examples.

The primary aim of this book is to provide good playable programs. To explain in detail how each one works would halve the number of games we could include.

If you take the time to follow through the workings of each program, then it should be possible to use the listings as a source for solutions to programming problems. Try to understand the reason that certain routines are used and how they work. Once the programs are fully working and saved on tape then it is recommended that you

experiment. Change the colours or parameters and see what happens. This way you can learn more about programming and be able to incorporate what you learn into your own programs. This is the most satisfying aspect of having your own computer.

Marie Celeste



An Adventure at Sea

In this text adventure you find yourself moored alongside the legendary ghost ship.

What is the secret of the *Marie Celeste*? Where are the crew?

By giving commands to the computer you can solve this marine mystery. By mixing up the text in the listing, care has been taken not to spoil your fun. In this way it is hoped that you will not be able to guess the solutions to the puzzles as they are typed in.

It is usual for adventures to require North, South, East and West as directions, but since this one has a nautical setting directions are: Forard, Aft, Starboard or Port. Which, in case you do not know, are equal to front of ship, rear of ship, right looking forward and left looking forward respectively.

As you move it is a good idea to draw a map. This map will help you to find your way around or identify locations that you have not yet visited.

Two-word instructions such as 'Take bucket' are accepted or the direction may be shortened to FOR, AFT, STA and POR. Adventures

often require the player to enter :'INVENTORY' to see what is being carried. In this adventure the inventory is displayed after each move. One of the enjoyable (but often frustrating) aspects of adventures is trying lots of different 'verbs' to see if they have any effect! Common verbs for adventures are: get, take, down, up, climb and unlock. Whilst you are on board, try lots of different ones. Try 'Examine cat' at the right time and see what happens.

Once you have solved the adventure invite others to play it. It is just as satisfying watching someone else make the same mistakes and perhaps have the same success as you.

```

1000 REM MARIE CELESTE-ISSI/JIM/PAULG
1010 SCREEN0:COLOR 15,4,7:KEY OFF
1020 WIDTH 38:GOTO2530
1030 RESTORE1120
1040 FOR N=1 TO 35
1050 FOR M=1 TO 6
1060 READ A:LOCATE N,M
1070 PRINT CHR$(1);CHR$(A);
1080 NEXT M
1090 NEXT N
1100 PRINT:PRINT
1110 RETURN
1120 DATA 82,86,86,86,86,81,87,32,32,
32,32,32,82,86,86,86,86,81,87,32,32,3
2,32,32,89,86,86,86,86,81
1130 DATA 32,32,83,86,86,90,32,32,87,
87,32,87,32,32,89,83,86
1140 DATA 81,32,32,84,86,86,81,32,32,
87,32,32,32,32,32,89,32,32,32
1150 DATA 32,32,89,86,86,81
1160 DATA 32,32,88,84,86,90,32,32,87,
87,32,87,32,32,89,91,32,91
1170 DATA 32,32,32,32,32,32
1180 DATA 88,86,86,86,86,90,87,32,32,
32,32,87,89,32,32,32,32,91
1190 DATA 32,32,88,84,86,90,32,32,87,
87,32,87,32,32,89,91,32,91
1200 DATA 89,86,86,86,86,81
1210 DATA 32,32,88,84,86,90,32,32,87,
87,32,87,32,32,89,91,32,91
1220 DATA 32,32,88,90,32,90,32,32,87,
87,32,87,32,32,89,89,86,91
1230 DATA 32,88,84,86,86,90,32,32,87,
32,32,87,32,32,32,23,32,91

```

```

1240 DATA 32,32,88,84,86,90,32,32,87,
87,32,87,32,32,89,91,32,91
1250 REM <<<<<< initialise >>>>>>
1260 DIM D(10,7)
1270 DIM O$(11)
1280 DIM M(10,4)
1290 DIM O(11)
1300 RESTORE 1400
1310 FOR N=1 TO 10
1320 READ D(N,1),D(N,2),D(N,3),D(N,4)
1330 READ D(N,5),D(N,6),D(N,7)
1340 READ M(N,1),M(N,2),M(N,3),M(N,4)
1350 NEXT N
1360 FOR N=1 TO 11
1370 READ O$(N),O(N)
1380 NEXT N
1390 RETURN
1400 DATA 1,8,10,11,20,0,0
1410 DATA 0,0,0,0
1420 DATA 12,36,23,0,0,0,0
1430 DATA 1,1,0,0
1440 DATA 13,16,1,29,4,30,20
1450 DATA 1,0,1,0
1460 DATA 13,14,3,23,0,0,0
1470 DATA 0,1,1,0
1480 DATA 1,18,35,3,33,0,0
1490 DATA 0,1,0,0
1500 DATA 1,8,9,11,20,0,0
1510 DATA 0,0,0,0
1520 DATA 2,30,20,13,15,22,16
1530 DATA 1,0,1,1
1540 DATA 5,30,20,0,0,0,0
1550 DATA 0,0,1,0
1560 DATA 1,8,34,29,5,0,0
1570 DATA 1,0,0,1
1580 DATA 1,17,27,33,3,30,20
1590 DATA 0,0,1,0
1600 DATA KNIFE,6 FACA
1610 DATA TARPAULIN,1 TRAPPOLIA
1620 DATA WATER ON DECK,2 CONVES
1630 DATA ROPE,6 CORDA, EFORÇAMENTO
1640 DATA NOTE,8 NOTA, OBSERVAÇÃO
1650 DATA BUCKET,4 BALDE
1660 DATA KEY,0 CHAVE
1670 DATA CAT,9 GATO

```

```

1680 DATA MOP,5 VASSOURA DE LAVABEN
1690 DATA WELLIES,5 POGOS,RELEVATORIOS
1700 DATA HOOKS,0 GACHOS,AN2015
1710 REM <<<<<<< input >>>>>>>>>>
1720 M$="":PRINT:PRINT:PRINT">";
1730 A=ASC(INPUT$(1)) AND 223:A=A OR
32 AND A=0:A$=CHR$(A)
1740 IF A=8 AND M$>" " THEN M$=LEFT$(
M$,LEN(M$)-1):PRINT CHR$(127);
1750 IF A=13 AND M$>" " THEN PRINT A$:
RETURN
1760 IF (A$<"A" OR A$>"Z") AND A<>32
OR LEN(M$)=40 THEN 1730
1770 M$=M$+A$:PRINT A$;:GOTO 1730
1780 REM <<<<<<< inventory >>>>>>>>
1790 PRINT:PRINT
1800 PRINT "INVENTORY."
1810 PRINT " _____ "
1820 PRINT
1830 FOR N=1 TO 11
1840 IF O(N)=13 THEN PRINT O$(N)
1850 NEXT N
1860 RETURN
1870 REM <<<<<<< split >>>>>>>>>
1880 V$="":N$=""
1890 FOR N=1 TO LEN(M$)
1900 IF MID$(M$,N,1)=" " THEN GOTO 19
40
1910 NEXT N
1920 V$=M$
1930 RETURN
1940 V$=LEFT$(M$,N-1)
1950 N$=RIGHT$(M$,LEN(M$)-N)
1960 RETURN
1970 REM <<<<<<< command >>>>>>>>>
1980 GOSUB 1720
1990 GOSUB 1880
2000 C$=LEFT$(V$,2)
2010 IF C$="PO" THEN GOSUB2570
2020 IF C$="ST" THEN GOSUB2640 ESTIBARDO
2030 IF C$="FO" THEN GOSUB 2710 FACUTE
2040 IF C$="AF" THEN GOSUB 2780 RE
2050 IF C$="GE" OR C$="TA" THEN GOSUB 2850
2850
2060 IF C$="GO" OR C$="EN" THEN GOSUB
2960

```

ARREBENTAR DE RISO, FENDER CASCAR

BARBARDO

ESTIBARDO

FACUTE

RE

PEBUE

```

2070 IF C$="EX" OR C$="IN" OR C$="RE"
  THEN GOSUB 3000 EXAMINE
2080 IF C$="UN" OR C$="OP" THEN GOSUB
  3200. ADRA
2090 IF C$="CU" THEN GOSUB 3250
2100 IF C$="CL" THEN GOSUB 3320
2110 IF C$="KI" THEN GOSUB 3390
2120 IF C$="LO" THEN GOSUB 3450 F6CHAA
2130 IF C$="SH" OR C$="FU" OR C$="RA"
  THEN GOSUB 3540
2140 C$=LEFT$(V$,3)
2150 IF C$="SWA" OR C$="MOP" THEN GOS
  UB 3570
2160 IF C$="MGV" OR C$="LIF" THEN GOS
  UB 3630
2170 IF V$="QUIT" THEN GOTO 3730
2180 IF V$="*" THEN GOTO 2200
2190 PRINT:PRINT "I don't understand.
  ..":PRINT
2200 GOSUB 2390:GOSUB1790
2210 GOTO 1980
2220 REM <<<<<<<< start >>>>>>>>>
2230 GOSUB 2300
2240 GOSUB 1260
2250 GOSUB 1030
2260 GOSUB 1790
2270 PO=6
2280 RETURN
2290 REM <<<<<<<< strings >>>>>>>>>
2300 DIM W$(36)
2310 RESTORE 2360
2320 FOR N=1 TO 36
2330 READ W$(N)
2340 NEXT N
2350 RETURN
2360 DATA I AM ,ASTERN,FORARD ,PORT ,
  STARBOARD ,UP ,DOWN ,IN A ,SMALL ,LAR
  GE
2370 DATA "BOAT, ALONGSIDE",THERE IS
  ,THE WIND ,HOWL3 ,SINGS
2380 DATA SOUNDS LIKE LAUGHTER.,OUTSI
  DE ,INSIDE ,IT IS ,THE MARIE CELESTE.
2390 DATA A ,AND ,HERE.,AND I ,SLIDE
  ,ACROSS ,LOCKED ,UNLOCKED ,TO ,OF,IS
  ,I SEE ,STORES.

```

```

2400 DATA GAP ,THE ,WATER ON THE DECK
.,DEVIL,BERMUDA TRIANGLE,ATLANTIS
2410 REM <<<<<<< describe >>>>>>>
2420 IF PO=2 AND O(10)<>13 THEN PRINT
"You slip through to the next deck
  section !":PO=3
2430 IF PO=1 AND O(2)=0 THEN GOTO 380
0
2440 FOR N=1 TO 7
2450 PRINT W$(D(PO,N));" ";
2460 NEXT N
2470 PRINT:PRINT:PRINT" I can see- "
2480 FOR N=1 TO 11
2490 IF O(N)=PO THEN PRINT O$(N);", ";
2500 NEXT N
2510 RETURN
2520 REM <<<<<<< begin game >>>>>>>
2530 GOSUB 2230
2540 GOSUB 2420
2550 GOTO 1980
2560 REM <<<<<<< commands >>>>>>>>
2570 V$="*"
2580 IF M(PO,4)<>0 THEN GOTO 2600
2590 PRINT "You cannot go that way.":
RETURN
2600 IF M(PO,4)=1 THEN PO=PO-5:RETURN
2610 A=M(PO,4):A=A-1:IF O(A)=13 THEN
PRINT "You use the ";O$(A):PRINT "and
  go to port." ELSE GOTO 2590
2620 FOR N=1 TO 3000:NEXT N
2630 PO=PO-5:RETURN
2640 V$="*"
2650 IF M(PO,2)<>0 THEN GOTO 2670
2660 PRINT "You cannot go that way.":
RETURN
2670 IF M(PO,2)=1 THEN PO=PO+5:RETURN
2680 A=M(PO,2):A=A-1:IF O(A)=13 THEN
PRINT "You use the ";O$(A):PRINT "and
  go to starboard." ELSE GOTO 2660
2690 FOR N=1 TO 3000:NEXT N
2700 PO=PO+5:RETURN
2710 V$="*"
2720 IF M(PO,1)<>0 THEN GOTO 2740
2730 PRINT "You cannot go that way.":
RETURN

```

```

2740 IF M(PO,1)=1 THEN PO=PO+1:RETURN
2750 A=M(PO,1):A=A-1:IF O(A)=13 THEN
PRINT "you use the ";O$(A):PRINT"and
go forard." ELSE GOTO 2730
2760 FOR N=1 TO 3000:NEXT N
2770 PO=PO+1:RETURN
2780 V$="*"
2790 IF M(PO,3)<>0 THEN GOTO 2810
2800 PRINT "You cannot go that way.":
RETURN
2810 IF M(PO,3)=1 THEN PO=PO-1:RETURN
2820 A=M(PO,3):A=A-1:IF O(A)=13 THEN
PRINT "You use the ";O$(A):PRINT "and
go aft." ELSE GOTO 2800
2830 FOR N=1 TO 3000:NEXT N
2840 PO=PO-1:RETURN
2850 V$="*"
2860 FOR N=1 TO 10
2870 IF N$=O$(N) THEN GOTO 2910
2880 NEXT N
2890 PRINT "Don't be silly !"
2900 RETURN
2910 IF N=4 THEN PRINT "It's tied too
tightly for me !":RETURN
2920 IF O(N)<>PO THEN GOTO 2890
2930 O(N)=13
2940 PRINT "You take the ";O$(N)
2950 RETURN
2960 V$="*"
2970 IF N$<>"STORE" AND N$<>"STORES"
THEN PRINT "You cannot go there ...":
RETURN
2980 IF M(10,4)=0 THEN PRINT "It's lo
cked !":RETURN
2990 PO=5:RETURN
3000 V$="*"
3010 FOR N=1 TO 11
3020 IF N$=O$(N) THEN GOTO 3060
3030 NEXT N
3040 PRINT "I don't see that here !"
3050 RETURN
3060 IF O(N)<>PO THEN GOTO 3040
3070 IF N=1 THEN PRINT "It's sharp !"
3080 IF N=2 THEN PRINT "It's green !"
3090 IF N=3 THEN PRINT "It's wet & sa
lty !"

```



```

3100 IF N=4 THEN PRINT "It's tied tig
htly !"
3110 IF N=5 THEN PRINT "It says -":PR
INT "'No need to go below'"
3120 IF N=6 AND O(7)=0 THEN PRINT "It
's rattling !":O(7)=PO
3130 IF N=6 AND O(7)<>0 THEN PRINT "I
t's empty !"
3140 IF N=7 THEN PRINT "it says 'STOR
E' !"
3150 IF N=8 THEN PRINT "It runs off w
ith a dead rat !":O(8)=0
3160 IF N=9 THEN PRINT "It's soggy !"
3170 IF N=10 THEN PRINT "They're smel
ly !"
3180 IF N=11 THEN PRINT "They seem to
be for a rope !"
3190 RETURN
3200 V$="*"
3210 IF PO<>10 THEN PRINT "I can't.."
:RETURN
3220 IF O(7)<>13 THEN PRINT "I have n
o key !":RETURN
3230 M(PO,4)=1:PRINT "The store is op
en."
3240 D(10,3)=28:RETURN
3250 V$="*":IF O(1)<>13 THEN PRINT "W
ith what ?":RETURN
3260 IF N$=O$(4) THEN GOTO 3290
3270 PRINT "You sadist !"
3280 RETURN
3290 PRINT "You cut the rope and take
it."
3300 O(4)=13
3310 RETURN
3320 V$="*":IF N$(>O$(4) THEN V$="BI"
:RETURN
3330 IF O(4)<>PO AND O(4)<>13 THEN PR
INT "Where's the ladder ?":RETURN
3340 PRINT "You climb the rope ladder
."
3350 IF PO=6 THEN PO=7:O(4)=7:RETURN
3360 IF PO=7 THEN PO=6:O(4)=6:RETURN
3370 IF PO=1 THEN PO=2:O(4)=2:RETURN
3380 RETURN

```

```

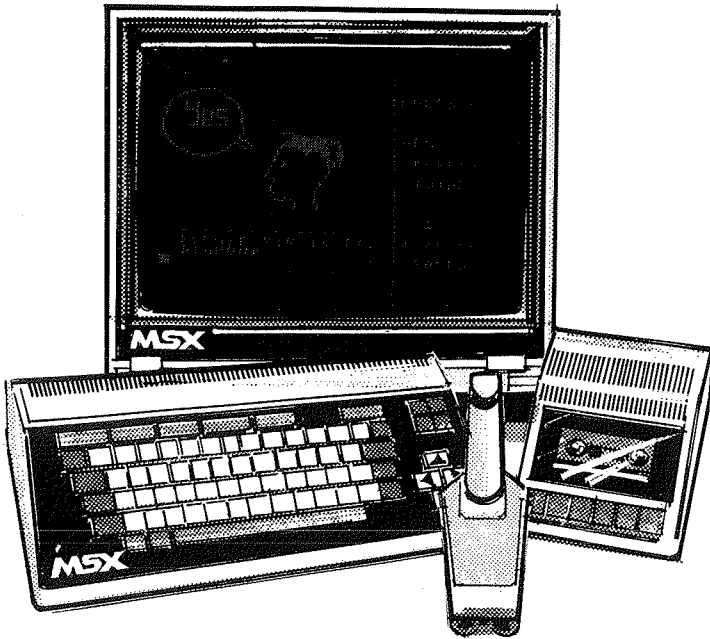
3390 PRINT " An Albatross flies down
"
3400 PRINT "and attacks you for "
3410 PRINT "being destructive.
3420 PRINT " You Are Dead !!!"
3430 FOR N=1 TO 5000:NEXT N
3440 GOTO 3730
3450 V$="*"
3460 PRINT:PRINT "EXITS ARE :"
3470 IF M(PO,1)<>0 THEN PRINT "FORARD
";
3480 IF M(PO,3)<>0 THEN PRINT "AFT ";
3490 IF M(PO,2)<>0 THEN PRINT "STARBO
ARD ";
3500 IF M(PO,4)<>0 THEN PRINT "PORT "
3510 IF M(PO,1)=0 AND M(PO,2)=0 AND M
(PO,3)=0 AND M(PO,4)=0 THEN PRINT "NO
NE."
3520 PRINT
3530 RETURN
3540 V$="*"
3550 PRINT " Not now sailor !"
3560 RETURN
3570 V$="*"
3580 IF O(6)<>13 OR O(9)<>13 THEN PRI
NT "With what you fool ?...":RETURN
3590 IF PO<>2 THEN PRINT "Not here yo
u idiot...":RETURN
3600 O(3)=0:PRINT "Well done ! Now it
is safe.":M(2,3)=5
3610 D(2,1)=1:D(2,2)=2:D(2,3)=20:D(2,
4)=32:D(2,5)=21:D(2,6)=11:D(2,7)=0
3620 O(11)=2:RETURN
3630 V$="*"
3640 FOR N=1 TO 10
3650 IF N#=0$(N) THEN GOTO 3690
3660 NEXT N
3670 PRINT "Move what ? ! ?"
3680 RETURN
3690 PRINT "You move it, and the"
3700 PRINT "wind blows it away !"
3710 O(N)=0
3720 RETURN
3730 PRINT "Another game ?"
3740 GOSUB 1720

```

```
3750 IF M$="YES" OR M$="Y" THEN RUN
3760 PRINT "Well you're playing anywa
y !"
3770 FOR N=1 TO 2500
3780 NEXT N
3790 RUN
3800 PRINT:PRINT:PRINT"SUPRISE SURPRI
SE !!!!!":PRINT "We hid in "W$(21)" "
W$(11)" "W$(20)
3810 PRINT "We thought you would"
3820 PRINT "NEVER solve the adventure
!"
3830 GOTO3730
```

2

Please Sir



A Teaser for Everyone

This program is not quite what it seems! You may think that you are able to guess what the secret of the program is as you enter it, but you will still find the game a challenge when you play it.

The idea is that the on-screen 'teacher' selects a subject and you must then give examples which fit the subject. For example, if 'Films' was displayed then you might enter 'Please sir, is SUPERMAN one?' and teacher may say 'yes' or 'no'! You must try to decide why and then enter another example such as 'STAR WARS' to which the reply will once again be 'yes' or 'no'! The game continues until you achieve three 'yes' replies in a row, when it is assumed that you have realised why some get a 'yes' and others 'no'. You can cheat, but that most certainly would be a shallow victory.

Later, when you have mastered playing the program, you may wish to change the rules or add more subjects, and that may be a challenge in itself!

Invite a friend to list the program, and he may be surprised to find

that there is a distinct lack of data, whereupon you may either give your own reason or offer the following test of gullibility:

'The program uses an advanced text compacting system based upon the mathematical characteristics of letters within English words. Thus, careful selection of parameters enables specific target answers to be 'tuned' into apparently imprecise test routines.'

Of course, you will know differently, because you typed in the lines which verify the entries! Why then, is a correct answer for the 'Birds' category: Sparrow, Chaffinch and Swallow?

```

1000 REM >>> PLEASE SIR      ANDY <<<
1010 CLEAR 2000
1020 DIM S$(11)
1030 GOSUB 2130
1040 RESTORE 1770
1050 SCREEN 1,3:COLOR 15,0,0:KEY OFF
1060 FOR S=0 TO 9
1070 FOR A=1 TO 32
1080 READ A#
1090 S$(S)=S$(S)+CHR$(VAL("&H"+A#))
1100 NEXT A
1110 SPRITE$(S)=S$(S)
1120 NEXT S:Y=56
1130 FOR X=0 TO 160
1140 GOSUB 1970
1150 NEXT
1160 GOSUB 2020
1170 LOCATE 11,5:PRINT "PLEASE"
1180 LOCATE 12,7:PRINT "SIR!"
1190 FOR T=1 TO 2500:NEXT T
1200 CLS
1210 X=80:Y=40:GOSUB 1970:GOSUB 2020
1220 FOR YY=0 TO 23:LOCATE 20,YY:PRIN
T "!":NEXT YY
1230 FOR X=21 TO 30:LOCATE X,8:PRINT
"=":LOCATE X,16:PRINT "=":NEXT X
1240 LOCATE 21,2:PRINT "GUESS A"
1250 LOCATE 22,10:PRINT "TOTAL"
1260 LOCATE 22,18:PRINT "GAMES"
1270 GOSUB 2070
1280 LOCATE 2,16:PRINT "PLEASE SIR, IS
IT"
1290 REM >>>> START GAME    <<<<<

```

```

1300 WI=0:GO=GO+1
1310 S=INT(RND(1)*20)+1
1320 LOCATE 21,6:PRINT T$(S)
1330 LOCATE 22,14:PRINT TR
1340 A=T(S,1):B=T(S,2):C=T(S,3)
1350 DEF FN OK=ASC(LEFT$(A$,1))
1360 IF A=2 THEN DEF FN OK=ASC(RIGHT$(A$,1))
1370 IF A=3 THEN DEF FN OK=ASC(MID$(A$,B,1))
1380 IF A=4 THEN DEF FN OK=LEN(A$)
1390 IF A=5 THEN DEF FN OK=SP+B
1400 IF A=6 THEN DEF FN OK=ASC(LEFT$(RIGHT$(A$,2),1))
1410 REM >>>>      LOOP      <<<<<
1420 GOSUB 1490
1430 IF A$="QUIT" THEN RUN
1440 IF RE=C THEN GOSUB 1630 ELSE GOSUB 1660
1450 LOCATE 22,14::PRINT WIN
1460 IF WIN=3 THEN 1690
1470 GOTO 1420
1480 REM >>>>      INPUT      <<<<<
1490 A$="":SP=0
1500 LOCATE 2,17:PRINT A$
1510 W$=INKEY$:IF W$="" THEN 1510
1520 IF W$=" " THEN SP=SP+1:GOTO 1550
1530 IF W$=CHR$(13) AND LEN(A$)>1 THEN 1580
1540 IF W$<"A" OR W$>"Z" THEN GOTO 1490
1550 A$=A$+W$
1560 IF LEN(A$)=17 THEN 1580
1570 LOCATE 2,17:PRINT STRING$(15," "):GOTO 1500
1580 A=LEN(A$)
1590 IF A<1 THEN 1490
1600 RE=FN OK
1610 RETURN
1620 REM >>>>>      YES      <<<<<<
1630 PUT SPRITE 8,(30,10),15,8
1640 WIN=WIN+1:RETURN
1650 REM >>>>>      NO      <<<<<<
1660 PUT SPRITE 8,(30,10),15,9
1670 WIN=0:RETURN

```

```

1680 REM >>>>          WIN          <<<<<<
1690 SCREEN 1:LOCATE 8,3:PRINT "Congr
atulations !"
1700 LOCATE 7,4:PRINT STRING$(18,"0")
1710 LOCATE 5,10:PRINT "You solved th
e puzzle !"
1720 LOCATE 4,20:PRINT "Press 'Y' to
play again."
1730 A$=INKEY$
1740 IF A$="Y"THEN 1200
1750 GOTO 1730
1760 GOTO 1760
1770 DATA 0,0,0,0,0,0,0,0,0,0,0,1,1,1
,2,2
1780 DATA 0,0,7C,7,F,1F,3F,7F,41,80,8
0,0,0,60,30,60
1790 DATA 0,0,0,C6,FD,FF,FF,FF,FF,7F,
20,0,0,0,0
1800 DATA 0,0,0,0,E0,DE,BA,C2,E2,E7,E
7,67,27,2A,2A,2A
1810 DATA 4,4,8,8,7,1,2,2,2,1,2,2,1,0
,0,0
1820 DATA 0,0,0,0,0,0,0,0,0,0,0,18,E4
,2,2,2
1830 DATA 0,0,0,0,0,1,1,2,4,4,4,4,4,4
,2,2
1840 DATA 20,20,40,40,80,0,0,0,0,0,0,
0,0,0,0,0
1850 DATA 0,0,0,0,0,0,1,6,8,10,20,20,
40,40,40,40
1860 DATA 0,0,0,0,F,70,80,0,0,0,0,0,0,
0,0,0
1870 DATA 0,0,0,0,E0,10,8,4,3,0,0,0,0,
0,0,0
1880 DATA 0,0,0,0,0,0,0,0,0,0,80,40,20,
20,10,10,10
1890 DATA 80,80,80,80,80,80,40,40,20,
20,10,8,6,1,0,0
1900 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,8
0,60,1F
1910 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,1,6
,38,C0
1920 DATA 10,10,10,8,8,8,8,10,10,C,62
,99,7,0,0,0
1930 DATA 0,0,0,0,44,44,44,44,7D,5,5,
5,5,5,3D,0

```

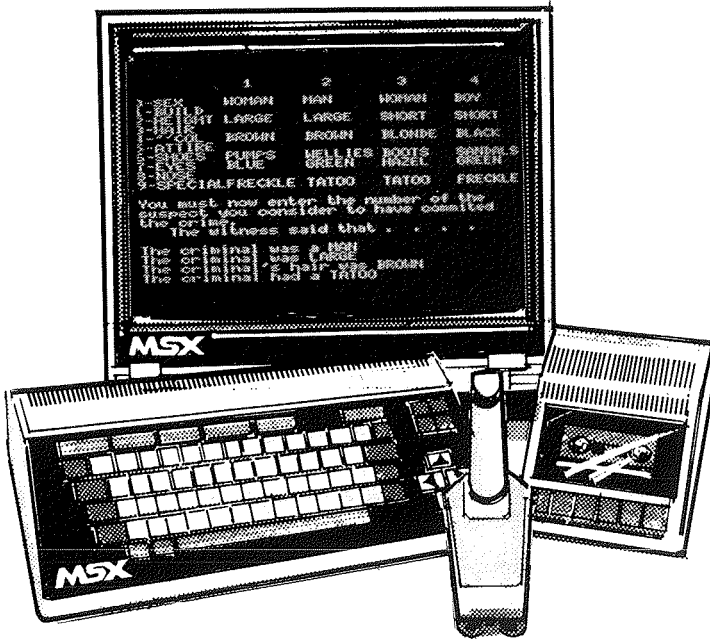
```

1940 DATA 0,0,0,0,0,0,0,0,0,EF,28,28,2F
,E1,1,EF,0
1950 DATA 0,0,0,0,1F,10,10,10,10,10,1
0,10,10,10,10,0
1960 DATA 0,0,0,0,80,80,80,80,80,BC,A
4,A4,A4,A4,BC,0
1970 PUT SPRITE 0,(X,Y),15,0
1980 PUT SPRITE 1,(X+32,Y),15,1
1990 PUT SPRITE 2,(X,Y+32),15,2
2000 PUT SPRITE 3,(X+32,Y+32),15,3
2010 RETURN
2020 PUT SPRITE 4,(X-64,Y-40),15,4
2030 PUT SPRITE 5,(X-32,Y-40),15,5
2040 PUT SPRITE 6,(X-64,Y-8),15,6
2050 PUT SPRITE 7,(X-32,Y-8),15,7
2060 RETURN
2070 LOCATE 1,15:PRINT "000000000000000
000000"
2080 FOR Y=16 TO 18:LOCATE 1,Y:PRINT
"!
!"
2090 NEXT Y
2100 LOCATE 1,19:PRINT "000000000000000
000000"
2110 RETURN
2120 REM >>>> SUBJECTS <<<<<
2130 RESTORE 2230
2140 DIM T$(20)
2150 FOR N=1 TO 20
2160 READ T$(N)
2170 NEXT N
2180 DIM T(20,3)
2190 FOR N=1 TO 20
2200 READ T(N,1),T(N,2),T(N,3)
2210 NEXT N
2220 RETURN
2230 DATA DOG,ACTOR,STAR,TV PROG,FILM
,PAY,SONG
2240 DATA MICRO,COLOUR,FLOWER,TREE,GE
M,BOOK,SPORT,BEAST,BIRD,PLACE
2250 DATA FISH,NAME,CAR
2260 DATA 2,1,78,1,2,82,3,2,79,5,1,2
2270 DATA 1,1,83,5,0,0,5,5,6,4,1,8
2280 DATA 4,1,4,2,1,80,4,1,3,2,1,68
2290 DATA 5,2,4,3,3,67,3,4,83,3,3,65
2300 DATA 2,1,82,3,2,65,3,2,65,4,1,6

```


3

I Accuse



A Suspicious Listing

This is a test of deduction which will challenge all budding sleuths.

You are Inspector Gibbs of CID (Computer Investigations Department) and are presently undertaking an extensive training course. This training is designed to test your courage and endurance. The course also tests your intelligence and ability to use the computer in criminal detection. Since we cannot cover the courage and endurance section in this book, we have included the computer deduction test.

The computer has the ability to generate thousands of different permutations from a set of data. In the blink of an eye it will decide on the criminal and his corroborative details, together with the statements of a witness that will conclusively point out the guilty party. You are allowed to ask for only six sets of details. This is done by entering the number corresponding to the category on the left of the screen. The computer will display them and then display the statements of key witnesses. This could be enough to identify the correct suspect, if you requested the right data!

The number of the suspect that you believe to be the criminal is entered and the accuracy of your judgement is reported back.

This program could well have educational value, but most of all it is designed to be different and fun.

```

1000 REM >>> I ACCUSE *JIM/ANDY<<<<
1010 REM
1020 KEY OFF
1030 SCREEN 0,0,0:COLOR 15,0,0:WIDTH
36
1040 GOSUB 1340
1050 GOSUB 1550
1060 GOSUB 1690
1070 FOR G=1 TO 6
1080 GOSUB 2080
1090 NEXT G
1100 LOCATE 0,18:PRINT STRING$(39," ")
)
1110 LOCATE 0,13:PRINT "You must now
enter the number of the suspect yo
u consider to have committed the cri
me."
1120 PRINT " The witness said that
. . . .":PRINT
1130 RESTORE 1320
1140 FOR N=1 TO 10
1150 READ A$
1160 IF W(1)=N OR W(2)=N OR W(3)=N OR
W(4)=N THEN PRINT "The criminal";A$;
" ";S$(S,N)
1170 NEXT N
1180 A$=INKEY$:IF A$<"1" OR A$>"4" TH
EN 1180
1190 CR=VAL(A$)
1200 FOR Y=12 TO 21:LOCATE 0,Y:PRINT
STRING$(39," "):NEXT Y
1210 IF CR=S THEN 1280
1220 LOCATE 0,15:PRINT "You're pathet
ic, 'Inspector' Gibbs . You will b
e demoted to P.C. Gibbs !"
1230 PLAY "116fedc12c
1240 PRINT:PRINT" It was suspect numb
er ";S;"!"
1250 LOCATE 6,20:PRINT "Press any key
to repeat game"

```

```

1260 A$=INKEY$:IF A$="" THEN 1260
1270 GOTO 1050
1280 LOCATE 0,15:PRINT "Congratulations, Inspector Gibbs !"
1290 PLAY "116cdefcdefcdel2f
1300 PRINT:PRINT "You correctly deduced the true criminal from the clues given."
1310 GOTO 1250
1320 DATA " was a", " was", " was", 's hair was, 's hair was, " wore a", 's shoes were, 's eyes were, 's nose was, " had a"
1330 REM >>> INITIALISE <<<<<
1340 DIM S$(4,10)
1350 DIM W(4)
1360 DIM L$(10,4)
1370 RESTORE 1440
1380 FOR N=1 TO 10
1390 FOR M=1 TO 4
1400 READ L$(N,M)
1410 NEXT M
1420 NEXT N
1430 RETURN
1440 DATA MAN,BOY,WOMAN,GIRL
1450 DATA FAT,THIN,SLIM,STUBBY
1460 DATA TALL,LARGE,SHORT,STOOPED
1470 DATA NORMAL,SHORT,CURLY,LONG
1480 DATA BROWN,BLACK,BLONDE,GINGER
1490 DATA JEANS,SUIT,JACKET,CLOAK
1500 DATA PUMPS,BOOTS,SANDALS,WELLIES
1510 DATA BLUE,BROWN,GREEN,HAZEL
1520 DATA SMALL,LARGE,BROKEN,CROOKED
1530 DATA TATOO,FRECKLE,DAGGER,GUN
1540 REM >>> SCREEN <<
1550 NO=1:SCREEN 0:WIDTH 40
1560 FOR N=10 TO 34 STEP 8
1570 LOCATE N,0:PRINT NO
1580 NO=NO+1
1590 NEXT N
1600 RESTORE 1650
1610 FOR N=2 TO 11
1620 LOCATE 0,N
1630 READ A$:PRINT A$
1640 NEXT N

```

```

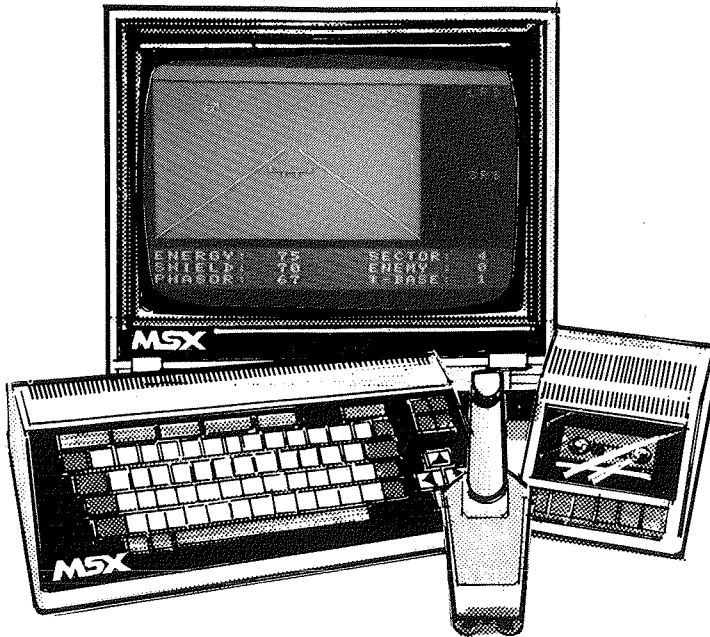
1650 DATA 0-SEX,1-BUILD,2-HEIGHT,3-HA
IR,4-' 'COL,5-ATTIRE,6-SHOES,7-EYES,8-
NOSE,9-SPECIAL
1660 LOCATE 0,18:PRINT "Choose your q
uestion (0-9). . . ."
1670 RETURN
1680 REM >>>          SET UP          <<<
1690 S=INT(RND(-TIME)*4)+1
1700 FOR N=1 TO 4
1710 W(N)=0
1720 CH=0
1730 A=INT(RND(1)*10)+1
1740 FOR Q=1 TO 4
1750 IF W(Q)=A THEN CH=1
1760 NEXT Q
1770 IF CH=1 THEN 1720
1780 W(N)=A
1790 NEXT N
1800 FOR N=1 TO 10
1810 FOR Q=1 TO 4
1820 A=INT(RND(1)*4)+1
1830 S$(Q,N)=L$(N,A)
1840 NEXT Q
1850 NEXT N
1860 N=W(1):GOSUB 1890:N=W(2):GOSUB 1
890
1870 N=W(3):GOSUB 1890:N=W(3):GOSUB 1
970
1880 RETURN
1890 A=INT(RND(1)*4)+1
1900 S$(S,N)=L$(N,A)
1910 T=S+2:IF T>4 THEN T=T-4
1920 S$(T,N)=L$(N,A)
1930 IF A<>1 THEN B=A-1 ELSE B=4
1940 T=T-2:IF T<1 THEN T=T+4
1950 S$(T,N)=L$(N,B)
1960 RETURN
1970 A=INT(RND(1)*4)+1
1980 S$(S,N)=L$(N,A)
1990 IF S<>1 THEN T=S-1 ELSE T=4
2000 S$(T,N)=L$(N,A)
2010 IF A<>4 THEN B=A+1 ELSE B=1
2020 T=T-1:IF T=0 THEN T=4
2030 S$(T,N)=L$(N,B)
2040 T=T-1:IF T=0 THEN T=4

```

```
2050 S$(T,N)=L$(N,B)
2060 RETURN
2070 REM >>>>      QUESTION      <<<<
2080 A$=INKEY$:IF A$="" THEN 2080
2090 IF A$<"0" OR A$>"9" THEN 2080
2100 PLAY "L8F":QW=VAL(A$)
2110 QW=QW+1
2120 FOR PE=1 TO 4
2130 LOCATE 8+(PE*8-7),1+QW
2140 PRINT S$(PE,QW);
2150 NEXT PE
2160 RETURN
```

4

Star Trek



The Program that Goes Boldly On

There have been versions of *Star Trek* written for just about every computer ever made. This says a lot for its attraction as a game and makes this *the* classic game.

Now you can have an MSX Federation Star Ship complete with scanners and phasers. Your mission is to clear the quadrant of the evil Klingons, without using up all your energy or being vaporised.

This version incorporates a zooming star field and 'real time' action during encounters with the enemy or when docking with star bases.

There follows an extract from the 32,544 page mega-manual entitled: 'How to be Captain of a Star Ship in 4000 easy lessons':

1. *Keep calm.* Remember the most you can lose is: your own life, 20 billion Galactic Dollars worth of hardware and the lives of everyone in the known universe.

2. *Watch where you are going!* Use the ship's eyes to reveal the nature of the universe. Pressing 'L' for Long Range Scan (LRS) will

display the position of your ship and the whereabouts of the enemy. Klingons are represented on the LRS by a circle with an arrow. This is the ancient symbol of Mars, the war bringer.

Star bases are represented by a circle above a cross. This is the ancient symbol for Venus, the bringer of love.

Those sectors which are occupied by *both* enemy and star bases have a square symbol in them. The square is an ancient symbol for a square.

3. *See what you can do!* Pressing 'S' will activate the short range scanners. This indicates the total number of enemy and star bases in the present sector.

4. *Keep an eye on your energy.* Docking with a star base will give you more. To dock press 'D' on your control panel. If there is a star base in that sector the docking sequence then requires you to steer your ship using left and right controls until the star base rests in the centre of your screen. Once docked you will receive 20 extra units to the overall ship's energy bank and a special 20 unit boost to your phasers and shields. Maximum capacity is 95 units. Failure to dock could result in the loss of the base. So be careful out there!

5. *Zap em', Cap'n!* A Star Fleet Captain has to do what he has to do, and that means eliminating the evil menacing Klingons.

If there are Klingons around, hitting the 'B' key sounds Battle Stations and you can start zapping. First get your enemy in the centre of the screen using the cursor pad or joystick. When you are ready, stab the copy key or fire button, sending phaser bolts out to destroy the target.

If you are not successful quickly you will lose the element of surprise and they will start to shoot back, thus draining your shield power. Do not give in, keep zapping until the last one is cleared from the sector. Then move on to win more battles.

6. *Warp carefully.* Movement to any sector is achieved by pressing 'W' followed by a number between 1 and 5. This is equal to the warp factor or distance you wish to travel.

Next the warp direction is required. This is a number between 1 and 8. Each number corresponds to a point on a compass, starting with one at the top and counting clockwise. Note: clocks have been retained in stardate 2806 to help Captains steer their Star Ships.

7. *The semi-final frontier.* On commencement of mission you can select the level. The easy level has only 10 enemies to seek and destroy. The average level has 20 and the hard level has 30.

As each level is completed, news of your success reaches Central Klingon Control and a replacement force, increased by 10 each time,

is sent out to do further battle. The ultimate challenge consists of 200 adversaries to wipe out. Are you ready Captain? Then let programming commence ...

```

1000 REM <<<<< STAR TREK <.> ISSI >>>
1010 REM
1020 MAXFILES=2:OPEN "GRP:S" FOR OUTP
UT AS 1
1030 SCREEN 1:KEY OFF
1040 RESTORE 3400
1050 S$="":FOR N=0 TO 7:READ A:S$=S$+
CHR$(A):NEXTN
1060 SPRITE$(1)=S$
1070 S$="":FOR N=0 TO 7:READ A:S$=S$+
CHR$(A):NEXTN
1080 SPRITE$(2)=S$
1090 DIM S(15,2)
1100 GOSUB 1770
1110 GOSUB 1870
1120 GOSUB 1550
1130 GOTO 1250
1140 A$=INKEY$
1150 IF A$="" THEN GOTO 1140
1160 A$=CHR$(ASC(A$) AND 223)
1170 IF A$="L" THEN GOSUB 2030
1180 IF A$="S" THEN GOSUB 2200
1190 IF A$="W" THEN GOSUB 2310
1200 IF A$="D" THEN GOSUB 2590
1210 IF A$="B" THEN GOSUB 2870:PUTSPR
ITE 2,(0,0),14,2
1220 FOR N=1 TO 20:NEXT N
1230 GOSUB 1520
1240 IF DEAD=1 OR ENER<1 THEN GOTO 13
80
1250 COLOR 15,13
1260 LINE (80,156)-(120,191),13,BF
1270 PRESET (80,160):PRINT #1,ENER
1280 PRESET (80,170):PRINT #1,SHIE
1290 PRESET (80,180):PRINT #1,PHAS
1300 LINE (208,156)-(268,191),13,BF
1310 PRESET (208,160):PRINT #1,SEC
1320 PRESET (208,170):PRINT #1,EME
1330 PRESET (208,180):PRINT #1,S(SEC,
2)
1340 IF EME=ALIEN THEN GOTO 3450

```



```
1350 GOTO 1140
1360 >>>>>>>>>> DEAD <<<<<<<<<<<<
1370 REM
1380 COLOR 10,6,6:SCREEN 1
1390 LOCATE 10,2:PRINT "BAD LUCK !"
1400 LOCATE 9,3:PRINT "-----"
1410 LOCATE 3,7:PRINT " The Starship
Enterprise"
1420 LOCATE 2,9:PRINT "has been destr
oyed by the"
1430 LOCATE 3,11:PRINT "Klingons, but
the Admiral"
1440 LOCATE 3,13:PRINT "Yamasheta has
decided to"
1450 LOCATE 3,15:PRINT "give you comm
and of yet"
1460 LOCATE 3,17:PRINT "another Stars
hip."
1470 LOCATE 2,21:PRINT "PRESS 'P' FOR
YOUR MISSION"
1480 A$=INKEY$:IF A$="" THEN GOTO 148
0
1490 A$=CHR$(ASC(A$)AND 223)
1500 IF A$<>"P" THEN GOTO 1480
1510 GOTO 1100
1520 RETURN
1530 REM >>>>>>>>> SCREEN <<<<<<<<<
1540 REM
1550 SCREEN 2:COLOR 11,1,1:CLS
1560 LINE (16,16)-(176,148),4,BF
1570 LINE (184,28)-(248,80),6,BF
1580 LINE (184,100)-(248,148),6,BF
1590 LINE (0,2)-(256,14),14,BF
1600 LINE (8,156)-(256,191),13,BF
1610 PRESET (208,18):PRINT #1,"LRS"
1620 PRESET (208,90):PRINT #1,"SRS"
1630 COLOR 6
1640 PRESET (64,4),14:PRINT #1,"CONDI
TION ~ RED ~"
1650 COLOR 15
1660 PRESET (16,160):PRINT #1,"ENERGY
:"
1670 PRESET (16,170):PRINT #1,"SHIELD
:"
1680 PRESET (16,180):PRINT #1,"PHASOR
:"
```

```

1690 PRESET (144,160):PRINT #1,"SECTO
R:"
1700 PRESET (144,170):PRINT #1,"ENEMY
:"
1710 PRESET (144,180):PRINT #1,"*-BAS
E:"
1720 PRESET (80,88),4:COLOR 10
1730 PRINT #1,CHR$(1);CHR$(90);CHR$(1
);CHR$(82);CHR$(1);CHR$(82);CHR$(1);C
HR$(91)
1740 RETURN
1750 REM >>> DIFFICULTY <<<
1760 REM
1770 COLOR 15,4,4:SCREEN 1
1780 PRINT "Hard/Easy/Average ?"
1790 A$=INKEY$:IF A$="" THEN GOTO 179
0
1800 A$=CHR$(ASC(A$)AND 223)
1810 IF A$="H" THEN ALIEN=20:RETURN
1820 IF A$="E" THEN ALIEN=10:RETURN
1830 IF A$="A" THEN ALIEN=15:RETURN
1840 GOTO 1790
1850 REM >>> INITIALIZE <<<
1860 REM
1870 FOR N=1 TO 15:S(N,1)=0:S(N,2)=0:
NEXT N
1880 FOR N=1 TO ALIEN
1890 SEC=INT(RND(-TIME)*15)+1
1900 IF S(SEC,1)>3 THEN GOTO 1890
1910 S(SEC,1)=S(SEC,1)+1
1920 NEXT N
1930 FOR N=1 TO 3
1940 PO=INT(RND(1)*15)+1
1950 IF S(PO,2)=1 THEN GOTO 1940
1960 S(PO,2)=1
1970 NEXT N
1980 PHAS=25:ENER=25:SHIE=25:BAD=0:EM
E=0
1990 SEC=INT(RND(1)*15)+1
2000 RETURN
2010 REM >>>> LONGSCAN <<<<
2020 REM
2030 PLAY "L8CDECDE":PO=1
2040 FOR N=1 TO 3
2050 FOR M=1 TO 5

```

```

2060 PRESET (188+M*8,18+N*16),6:COLOR
  10
2070 CH=71:IF PO=SEC THEN CH=79:GOTO
2110
2080 IF S(PO,1)>0 THEN CH=75
2090 IF S(PO,2)=1 THEN CH=76
2100 IF CH=76 AND S(PO,1)>0 THEN CH=7
  2
2110 PRINT #1,CHR$(1);CHR$(CH)
2120 PO=PO+1:NEXT M
2130 NEXT N
2140 FOR N=1 TO 5000:NEXT N
2150 ENER=ENER-5:PLAY "EDCEDC"
2160 LINE (184,28)-(248,80),6,BF
2170 RETURN
2180 REM >>>>>>>> SHORT SCAN <<<<<<<<
2190 REM
2200 PLAY "L8CDECDE"
2210 COLOR 15:PRESET (204,110),6
2220 PRINT #1,CHR$(1);CHR$(75);":":S(
SEC,1)
2230 PRESET (204,130),6
2240 PRINT #1,CHR$(1);CHR$(76);":":S(
SEC,2)
2250 FOR N=1 TO 5000:NEXT N
2260 ENER=ENER-5:PLAY "EDCEDC"
2270 LINE (184,100)-(248,148),6,BF
2280 RETURN
2290 REM >>>>>>>>> WARP <<<<<<<<<<<<
2300 REM
2310 LINE (0,2)-(256,14),14,BF:COLOR
  2
2320 PRESET (16,4),14:PRINT #1,"Warp
Factor (mod 1 to mod 5) ?"
2330 A$=INKEY$
2340 IF A$<"1" OR A$>"5" THEN GOTO 23
  30
2350 DIS=VAL(A$)
2360 LINE (0,2)-(256,14),14,BF
2370 PRESET (16,4),14:PRINT #1,"Mega-
Warp Direction (1 to 8) ?"
2380 A$=INKEY$
2390 IF A$<"1" OR A$>"8" THEN GOTO 23
  80
2400 DR=VAL(A$)

```

```

2410 COLOR 11
2420 FOR Q=60 TO 16 STEP -4
2430 LINE (Q,Q)-(192-Q,164-Q),INT(RND
(1)*15),B
2440 NEXT Q
2450 RESTORE 2560
2460 FOR N=1 TO DR:READ XY:NEXT N
2470 FOR N=1 TO DIS:SEC=SEC+XY
2480 IF SEC<1 THEN SEC=SEC+15
2490 IF SEC>15 THEN SEC=SEC-15
2500 PLAY "L8"+CHR$(N+66):NEXT N
2510 FOR Q=60 TO 16 STEP -4
2520 LINE (Q,Q)-(192-Q,164-Q),4,B
2530 NEXT Q
2540 ENER=ENER-INT(DIS*1.5)
2550 LINE (0,2)-(256,14),14,BF:RETURN
2560 DATA -5,-4,1,6,5,4,-1,-6
2570 REM >>>>>>>> DOCKING <<<<<<<<<
2580 REM
2590 LINE (0,2)-(256,14),14,BF
2600 PRESET (16,4),14:COLOR 6
2610 IF S(SEC,2)=0 THEN PRINT #1,"No
starbase in this sector .":PLAY "L2GE
C":RETURN
2620 X=INT(RND(-TIME)*100)+50:Y=18:DI
S=1
2630 PUT SPRITE 1,(X,INT(Y)),11,1
2640 GOSUB 3310
2650 IF RI=1 THEN DIS=DIS-1
2660 IF LE=1 THEN DIS=DIS+1
2670 Y=Y+.5:X=X+DIS
2680 IF X<15 THEN X=15
2690 IF X>170 THEN X=170
2700 IF Y<95 THEN GOTO 2630
2710 IF X>85 AND X<102 THEN GOTO 2770
2720 IF X<65 OR X>125 THEN PRESET (16
,4),14:PRINT #1,"STARBASE OUT OF RANG
E . .":GOTO 2810
2730 PRESET (16,4),14:PRINT #1,"THE S
TARBASE IS DESTROYED !"
2740 PLAY "S10EDCC"
2750 S(SEC,2)=0
2760 GOTO 2810
2770 PRESET (16,4),14:PRINT #1,"WE AR
E NOW DOCKING . . ."

```

```

2780 ENER=75
2790 PHAS=75
2800 SHIE=75
2810 FOR N=1 TO 700:NEXT N:BEEP
2820 LINE (0,2)-(256,14),14,BF
2830 PUT SPRITE 1,(0,0),11,1
2840 RETURN
2850 REM >>>>>>> BATTLE <<<<<<<<<<
2860 REM
2870 LINE (0,2)-(256,14),14,BF
2880 DEAD=0
2890 XD=2:YD=2
2900 COLOR 13
2910 IF S(SEC,1)=0 THEN PRESET (16,4)
,14:PRINT #1," THIS SECTOR IS CLEA
R.":PLAY "L4GEC":RETURN
2920 X=INT(RND(1)*140)+30:Y=INT(RND(1)
)*120)+30
2930 GOSUB 3310
2940 PUTSPRITE 2,(X,Y),7,2
2950 IF FI=1 AND PHAS>1 THEN GOTO 319
0
2960 IF KIL=1 THEN S(SEC,1)=S(SEC,1)-
1:KIL=0:GOTO 2880
2970 IF LE=1 THEN XD=2
2980 IF RI=1 THEN XD=-2
2990 IF UP=1 THEN YD=2
3000 IF DO=1 THEN YD=-2
3010 IF EX=1 THEN RETURN
3020 X=X+XD:Y=Y+YD
3030 IF X<20 THEN XD=20:XD=2
3040 IF X>165 THEN X=165:XD=-2
3050 IF Y<18 THEN Y=18:YD=2
3060 IF Y>135 THEN Y=135:YD=-2
3070 IF RND(1)<.95 THEN GOTO 3110
3080 PLAY "S1402L4C"
3090 FOR M=1 TO 50:NEXT M:BEEP
3100 SHIE=SHIE-1:IF SHIE<0 THEN DEAD=
1:RETURN
3110 LINE (80,170)-(120,190),13,BF:CO
LOR 15
3120 PRESET (80,170):PRINT #1,SHIE
3130 PRESET (80,180):PRINT #1,PHAS
3140 LINE (208,170)-(268,179),13,BF
3150 PRESET (208,170):PRINT #1,EME

```

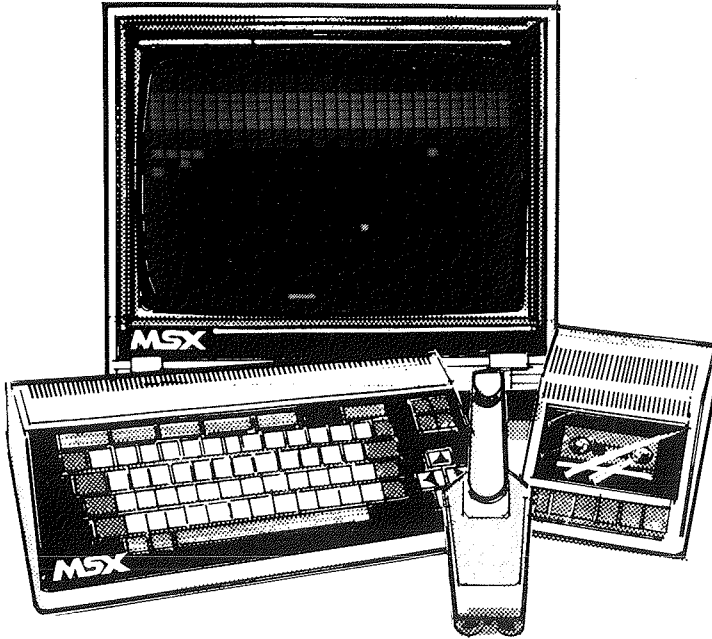
```

3160 GOTO 2930
3170 REM >>>>>>>> FIRE <<<<<<<<<<<
3180 REM
3190 COLOR 14
3200 PSET (16,148):DRAW "E75"
3210 PSET (176,148):DRAW "H75"
3220 PLAY "S14L64CDEFGAB"
3230 FOR N=1 TO 50:NEXT N
3240 PHAS=PHAS-1
3250 IF X>86 AND X<98 AND Y>64 AND Y<
70 THEN EME=EME+1:KIL=1:PLAY "S11L8CD
EL2G"
3260 COLOR 4
3270 PSET (16,148):DRAW "E75"
3280 PSET (176,148):DRAW "H75"
3290 BEEP:GOTO 2960
3300 REM >>>>>> INKEYS <<<<<<<
3310 VA=STICK(0) OR STICK(1) OR STICK
(2)
3320 IF VA=3 THEN RI=1 ELSE RI=0
3330 IF VA=7 THEN LE=1 ELSE LE=0
3340 IF VA=1 THEN UP=1 ELSE UP=0
3350 IF VA=5 THEN DO=1 ELSE DO=0
3360 FI=ABS(STRIG(0) OR STRIG(1) OR S
TRIG(2))
3370 A$=INKEY$
3380 IF A$=CHR$(13) THEN EX=1 ELSE EX
=0
3390 RETURN
3400 DATA 56,68,68,68,56,16,124,16
3410 DATA 15,3,5,121,136,136,136,112
3420 GOTO 1140
3430 REM >>>>>>>> NEXT <<<<<<<<<<<
3440 REM
3450 COLOR 10,6,6:SCREEN 1
3460 LOCATE 10,2:PRINT "WELL DONE ,"
3470 LOCATE 9,3:PRINT "-----"
3480 LOCATE 3,7:PRINT " Captain James
Kirk ,"
3490 LOCATE 3,9:PRINT "Now you must f
ight an"
3500 LOCATE 3,11:PRINT "even fiercer
group of"
3510 LOCATE 3,13:PRINT "evil KLINGONS
. . . ."

```

```
3520 LOCATE 2,21:PRINT "PRESS 'P' FOR  
YOUR MISSION"  
3530 A#=INKEY#:IF A#="" THEN GOTO 353  
0  
3540 A#=CHR$(ASC(A#)AND 223)  
3550 IF A#<>"P" THEN GOTO 3530  
3560 ALIEN=ALIEN+10  
3570 GOTO 1110
```

5 Rainbow Breakout



An Arcade Classic

This second program classic has great attraction for players of all ages. It is one of the shorter programs in the book and so may be a good first attempt for those new to programming.

At the start you are asked to select a large or small bat, then the game begins. The object of the game is to eliminate all the blocks by hitting them with the ball. Your bat is moved left or right using the cursor pad or joystick. It must be moved in advance to anticipate the path of the ball as it bounces back from the blocks.

If you lose a ball then there are three supplied before the game restarts. The score is shown and can reach very high levels in the hands of a skilful player.

```
1000 REM <<<<< Rainbow Breakout >>>>>
1010 REM <<<<<<<<< bootsy >>>>>>>>>>
1020 SCREEN 1,0,0:COLOR 15,0,0:CLS
1030 KEY OFF
1040 ON SPRITE GOSUB 1490:SPRITE ON
```



```

1050 CLS:LOCATE 5,10:PRINT "Small or
Large ?";:I$=INPUT$(1)
1060 SZ=0:IF I$="1" OR I$="L" THEN SZ
=2
1070 SCREEN ,SZ:CLS
1080 LM=240-(16 AND SZ=3):X=120-(8 AN
D SZ>0)
1090 RESTORE:SC=0:LI=5:GOSUB 1610
1100 GOSUB 1210
1110 CT=0
1120 B=100
1130 C=INT(RND(-TIME)*2)
1140 C=C+INT(RND(-TIME)*30)+16
1150 BD=2:AD=-2
1160 GOSUB 1280
1170 GOSUB 1360
1180 IF CT=280 THEN GOTO 1100
1190 GOTO 1160
1200 REM <<<<<<<< screen >>>>>>>>>
1210 LOCATE0,0
1220 FOR N=216 TO 248 STEP 8
1230 PRINT STRING$(28,CHR$(N))
1240 PRINT STRING$(28,CHR$(N))
1250 NEXT N
1260 RETURN
1270 REM <<<<<<<< move bat >>>>>>>
1280 ST=STICK(0) OR STICK(1) OR STICK
(2)
1290 TG=STRIG(0) OR STRIG(1) OR STRIG
(2)
1300 I$=INKEY$
1310 IF ST=7 AND X>4 THEN X=X-4
1320 IF ST=3 AND X<LM THEN X=X+4
1330 PUT SPRITE 0,(X,188),15,0
1340 RETURN
1350 REM <<<<<<<< move ball >>>>>>>>
1360 A1=C:B1=B
1370 A1=A1+AD:B1=B1+BD
1380 IF A1<16 THEN A1=16:AD=-AD:PLAY"
t255c16"
1390 IF A1>240 THEN A1=240:AD=-AD:PLA
Y*t255c16"
1400 IF B1<0 THEN B1=1
1410 IF B1>190 THEN GOTO 1520
1420 CP=&H1800+INT(INT(B/8)*32+C/8)

```

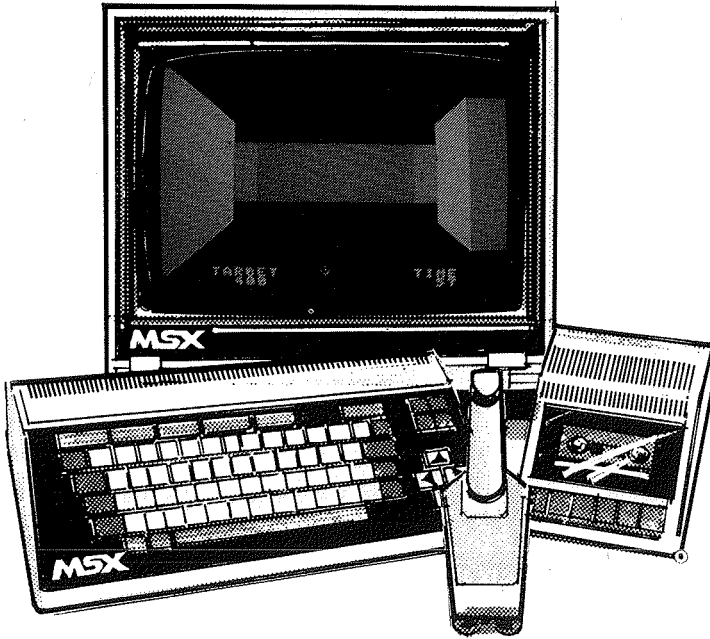
```

1430 CH=VPEEK(CP)
1440 IF CH<>32 THEN VPOKE CP,32:BEEP:
SC=SC+10:BD=-BD:CT=CT+1
1450 IF B1=0 THEN BD=ABS(BD):PLAY"t25
5c16"
1460 C=C+AD:B=B+BD
1470 PUT SPRITE 1,(C,B),15,1
1480 RETURN
1490 BEEP:BD=-ABS(BD)
1500 GOTO 1460
1510 REM <<<<<<<<<< dead >>>>>>>>>>>>
1520 PLAY "t255116edc"
1530 LI=LI-1:IF LI>0 THEN RETURN 1120
1540 CLS:PUT SPRITE 0,(255,208),0,0
1550 LOCATE6,8:PRINT"CONGRATULATIONS"
1560 LOCATE0,12:PRINT"You have scored
: ";SC;"points"
1570 LOCATE6,17:PRINT"Another game ?"
;:I$=INPUT$(1)
1580 IF I$="Y" OR I$="y" THEN RETURN
1050
1590 SCREEN 0:END
1600 REM <<<<<<<< def shapes >>>>>>>>>
1610 GOSUB1680:SPRITE$(0)=SP$:GOSUB16
80:SPRITE$(1)=SP$
1620 FOR N=1728 TO 1984 STEP 64:RESTO
RE 1780:FOR M=0 TO 7
1630 READ H$:VPOKE N+M,VAL("&h"+H$)
1640 NEXT M,N
1650 FOR N=&H201B TO &H201F
1660 READ H$:VPOKE N,VAL("&h"+H$)
1670 NEXT N:RETURN
1680 SP$="":FOR N=1 TO 32:READ H$:SP$
=SP$+CHR$(VAL("&h"+H$)):NEXT N:RETURN
1690 REM <<<<<<<< prog data >>>>>>>>>
1700 DATA ff,ff,ff,0,0,0,0,0
1710 DATA 0,0,0,0,0,0,0,0
1720 DATA ff,ff,ff,0,0,0,0,0
1730 DATA 0,0,0,0,0,0,0,0
1740 DATA f0,f0,f0,f0,0,0,0,0
1750 DATA 0,0,0,0,0,0,0,0
1760 DATA 0,0,0,0,0,0,0,0
1770 DATA 0,0,0,0,0,0,0,0
1780 DATA 0,7f,7f,7f,7f,7f,7f,7f
1790 DATA 80,90,a0,20,40

```

6

Maze Maniac



A Game with Perspective

This third classic is a 3D maze game in which you have a perspective view of the paths open to you.

The object of the game is to find your way around the maze quickly. If you do it fast enough then you will catch the maniac. If you are too slow then *he* will catch *you*!

On screen are shown the passages that are open to you. The arrow at the bottom indicates the true direction that you are facing.

The left or right cursor changes the direction you are facing. The up cursor takes one step into the picture each time it is pressed. The down cursor takes you back one step and leaves you facing the same direction. The joystick can also be used to move.

Once you are successful in reaching the end of the maze then the target time is reduced and the best time must be beaten.

This game uses a sophisticated technique, based upon the MSX colour pointers, to achieve the very fast changes to the picture. The technique is worthy of your close examination since you can use it to produce high speed effects in your own programs.

```

1000 REM>>Maze Maniac      BOOTSY <<
1010 REM
1020 SCREEN 1,,0:CLEAR 500,&HEEFF
1030 DEFUSR=&HEF00:KEY OFF:WIDTH 32
1040 ON INTERVAL=50 GOSUB 1230
1050 DIM S$(11),C$(13),W%(18,2,1),M%(
9,9),MT%(9,9):TT%=400
1060 LOCATE 14,11:PRINT "WAIT":GOSUB
2000
1070 GOSUB 1720:GOSUB 1180:INTERVAL 0
N
1080 GOSUB 1390
1090 I$=INKEY$:FOR I%=0 TO 2
1100 IP%=STICK(I%):IF IP% THEN I%=2
1110 NEXT I%:IF IP%=0 THEN 1090
1120 IF IP%=1 AND M%(X%,Y%)=2 THEN GO
TO 1310
1130 IF IP%=1 AND Y%<9 THEN IF M%(X%,
Y%+1)>0 THEN Y%=Y%+1:GOSUB 1180
1140 IF IP%=5 AND Y%>0 THEN IF M%(X%,
Y%-1)>0 THEN Y%=Y%-1:GOSUB 1180
1150 IF IP%=3 THEN SP%=(SP%+1) AND 3:
GOSUB 1180:GOSUB 1520
1160 IF IP%=7 THEN SP%=(SP%-1) AND 3:
GOSUB 1180:GOSUB 1570
1170 GOTO 1080
1180 IF IP%=3 OR IP%=7 THEN 1210
1190 XS%=XS%+(3 AND (IP%=1 AND SP%=1
OR IP%=5 AND SP%=3))-(3 AND (IP%=5 AN
D SP%=1 OR IP%=1 AND SP%=3))
1200 YS%=YS%+(3 AND (IP%=1 AND SP%=2
OR IP%=5 AND SP%=0))-(3 AND (IP%=5 AN
D SP%=2 OR IP%=1 AND SP%=0))
1210 PUT SPRITE 0,(XS%,YS%),4,SP%:RET
URN
1220 REM ----- timer -----
1230 TM%=TM%+1:LOCATE 23,22:PRINT USI
NG "####"; TM%
1240 IF TM%<TT% THEN RETURN
1250 INTERVAL OFF
1260 CLS:LOCATE 1,11
1270 PRINT "THE MAZE MANIAC HAS CAUGH
T YOU"
1280 PRINT " FROM BEHIND. YOU WERE TO
O SLOW!"

```

```

1290 RETURN 1360
1300 REM ----- out -----
1310 CLS:INTERVAL OFF:COLOR 15
1320 LOCATE 1,11:PRINT "WELL DONE YOU
HAVE ESCAPED THE"
1330 PRINT " MAZE MANIAC THIS TIME ..
.."
1340 PRINT " YOU TOOK ";TM%;"SECONDS"
1350 IF TM%<TT% THEN TT%=TM%
1360 PUT SPRITE 0,(128,208),4,0
1370 FOR I=1 TO 10000:NEXT:GOTO 1070
1380 REM ----- generate view -----
1390 FOR W%=2 TO 0 STEP -1:V%=Y%+3-W%
1400 IF V%=10 THEN GOSUB 1650:K%=W%:W
%=0
1410 IF V%<10 THEN IF M%(X%,V%)=0 THE
N GOSUB 1650:K%=W%:W%=0
1420 NEXT W%:O%=0:IF M%(X%,Y%+2-K%)=2
THEN O%=1:W%=K%:GOSUB 1650
1430 IF X%=9 THEN FOR D%=3 TO 1+K% ST
EP -1:OC%=0:GOSUB 1670:NEXT D%:GOTO 1
470
1440 V%=0:FOR D%=3 TO 1+K% STEP -1:OC
%=0
1450 IF M%(X%+1,Y%+V%)>0 THEN OC%=1
1460 GOSUB 1670:V%=V%+1:NEXT D%
1470 IF X%=0 THEN FOR D%=6 TO 4+K% ST
EP -1:OC%=0:GOSUB 1670:NEXT D%:GOTO 1
500
1480 V%=0:FOR D%=6 TO 4+K% STEP -1:OC
%=0:IF M%(X%-1,Y%+V%)>0 THEN OC%=1
1490 GOSUB 1670:V%=V%+1:NEXT D%
1500 RETURN
1510 REM ----- turn right -----
1520 FOR I%=0 TO 9:FOR J%=0 TO 9
1530 MT%(J%,I%)=M%(I%,9-J%)
1540 NEXT J%,I%
1550 SWAP X%,Y%:X%=9-X%:GOTO 1610
1560 REM ----- turn left -----
1570 FOR I%=0 TO 9:FOR J%=0 TO 9
1580 MT%(J%,9-I%)=M%(I%,J%)
1590 NEXT J%,I%
1600 SWAP X%,Y%:Y%=9-Y%
1610 FOR I%=0 TO 9:FOR J%=0 TO 9
1620 M%(J%,I%)=MT%(J%,I%)

```

```

1630 NEXT J%,I%:RETURN
1640 REM ----- set wall -----
1650 A=USR(VARPTR(W%(0,W%,0))):RETUR
N
1660 REM ----- set door -----
1670 C1%=0:C2%=128-(32 AND(D%=2 OR D%
=5))
1680 I%=8204+D%:J%=8210+D%
1690 IF OC% THEN VPOKE I%,C1%:VPOKE J
%,C2% ELSE C1%=144:C2%=C1%:OC%=1:GOTO
1690
1700 RETURN
1710 REM ----- draw screen -----
1720 RESTORE 2480:O%=0:SP%=1:TM%=0:XS
%=113:YS%=159
1730 FOR I%=0 TO 9:FOR J%=0 TO 9
1740 READ M%(I%,J%)
1750 NEXT J%,I%:X%=0:Y%=0
1760 COLOR 0,0,0:CLS
1770 RESTORE 2590
1780 GOSUB 1970:FOR I%=0 TO 11
1790 LOCATE 0,I%:FOR J%=1 TO 16
1800 K%=ASC(MID$(S$(I%),J%,1))
1810 PRINT C$(K%);:NEXT J%,I%
1820 GOSUB 1970:FOR I%=0 TO 11
1830 LOCATE 16,I%:FOR J%=16 TO 1 STEP
-1
1840 K%=ASC(MID$(S$(I%),J%,1))
1850 PRINT C$(K%);:NEXT J%,I%
1860 GOSUB 1970:FOR I%=11 TO 0 STEP-1
1870 LOCATE 0,23-I%:FOR J%=1 TO 16
1880 K%=ASC(MID$(S$(I%),J%,1))
1890 PRINT C$(K%);:NEXT J%,I%
1900 GOSUB 1970:FOR I%=11 TO 0 STEP-1
1910 LOCATE 16,23-I%:FOR J%=16 TO 1 S
TEP-1
1920 K%=ASC(MID$(S$(I%),J%,1))
1930 PRINT C$(K%);:NEXT J%,I%
1940 LOCATE 5,21:COLOR 15:PRINT "TARG
ET" TAB(23) "TIME"
1950 PRINT TAB(6);TT%
1960 RETURN
1970 FOR I%=1 TO 13:READ K%:C$(I%)=CH
R$(K%):NEXT I%
1980 RETURN

```

```

1990 REM ----- def shapes -----
2000 FOR I%=832 TO 1984 STEP 64
2010 FOR J%=0 TO 7
2020 K%=256-2^(7-J%):L%=2^(J%+1)-1
2030 M%=2^(8-J%)-1:N%=256-2^J%
2040 VPOKE I%+J%,255:VPOKE I%+J%+8,K%
2050 VPOKE I%+J%+16,L%:VPOKE I%+J%+24
,M%
2060 VPOKE I%+J%+32,N%:VPOKE I%+J%+40
,0
2070 NEXT J%,I%
2080 FOR I%=0 TO 11:FOR J%=1 TO 16
2090 READ K%:S$(I%)=S$(I%)+CHR$(K%)
2100 NEXT J%,I%
2110 FOR I%=0 TO 1:FOR J%=0 TO 2:FOR
K%=0 TO 18
2120 READ W%(K%,J%,I%):NEXT K%,J%,I%
2130 FOR I%=0 TO 3:S$="":FOR J%=0 TO
7
2140 READ K%:S$=S$+CHR$(K%):NEXT J%
2150 SPRITE$(I%)=S$:NEXT I%
2160 I%=&HEF00
2170 READ H$:IF H$="x" THEN RETURN
2180 POKE I%,VAL("&h"+H$)
2190 I%=I%+1:GOTO 2170
2200 REM ----- prog data -----
2210 DATA 8,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1
2220 DATA 9,8,1,1,1,1,1,1,1,1,1,1,1,1
,1,1
2230 DATA 9,9,8,1,1,1,1,1,1,1,1,1,1,1
,1,1
2240 DATA 9,9,9,8,1,1,1,1,1,1,1,1,1,1
,1,1
2250 DATA 5,5,5,5,10,2,2,2,2,2,2,2,2,2,2
,2,2,2
2260 DATA 5,5,5,5,11,10,2,2,2,2,2,2,2,2
,2,2,2
2270 DATA 5,5,5,5,11,11,10,2,2,2,2,2,2,2
,2,2,2
2280 DATA 5,5,5,5,6,6,6,12,3,3,3,3,3,3
,3,3,3
2290 DATA 5,5,5,5,6,6,6,13,12,3,3,3,3,3
,3,3,3
2300 DATA 5,5,5,5,6,6,6,7,7,4,4,4,4,4
,4,4

```

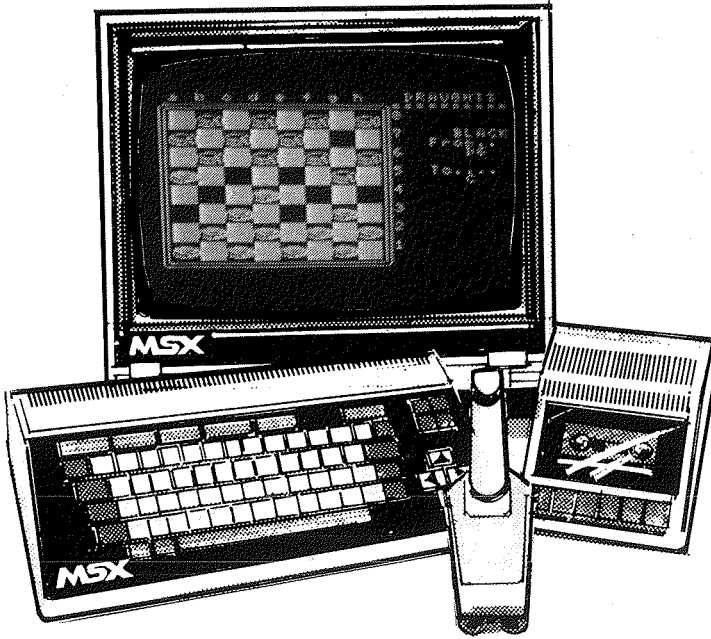
```

2310 DATA 5,5,5,5,6,6,6,7,7,4,4,4,4,4
,4,4
2320 DATA 5,5,5,5,6,6,6,7,7,4,4,4,4,4
,4,4
2330 REM ----- pat cols -----
2340 DATA 144,144,144,144,144,144,144
,144,144,144,144,0,0,0,0,0,0,96
2350 DATA 102,144,144,102,144,144,96,
144,144,96,144,144,6,0,0,6,0,0,96
2360 DATA 102,102,144,102,102,144,96,
96,144,96,96,144,6,6,0,6,6,0,96
2370 DATA 144,144,144,144,144,144,144
,144,144,144,144,144,0,0,0,0,0,0,0
2380 DATA 0,144,144,0,144,144,0,144,1
44,0,144,144,0,0,0,0,0,0,0
2390 DATA 0,0,144,0,0,144,0,0,144,0,0
,144,0,0,0,0,0,0,0
2400 REM ----- arrow -----
2410 DATA 16,56,68,130,16,16,16,16
2420 DATA 8,4,2,255,2,4,8,0
2430 DATA 16,16,16,16,130,68,56,16
2440 DATA 16,32,64,255,64,32,16,0
2450 REM ----- patch -----
2460 DATA 23,23,5e,23,56,21,d,20,cd,5
3,0,e3,e3,eb,6,13,7e,d3,98,23,23,10,f
9,c9,x
2470 REM ----- maze pattern -----
2480 DATA 1,1,1,0,1,1,1,0,1,1
2490 DATA 1,0,1,1,1,0,1,1,1,0
2500 DATA 1,1,0,1,0,1,1,0,1,1
2510 DATA 0,1,0,1,1,1,0,0,0,1
2520 DATA 1,1,1,0,0,0,1,1,1,0
2530 DATA 0,0,1,0,1,1,1,0,1,0
2540 DATA 1,1,1,0,0,1,0,1,1,1
2550 DATA 1,0,0,1,1,1,0,1,0,1
2560 DATA 1,1,1,0,1,0,1,1,1,0
2570 DATA 0,0,1,1,1,0,1,0,1,2
2580 REM ----- screen pats -----
2590 DATA 245,237,229,248,192,184,176
,145,144,137,136,129,128
2600 DATA 245,237,229,248,168,160,152
,122,120,114,112,106,104
2610 DATA 221,213,205,248,192,184,176
,148,144,140,136,132,128
2620 DATA 221,213,205,248,168,160,152
,123,120,115,112,107,104

```


7

Duo Draughts



Computer Moderated Micro Draughts

The idea is very simple – you must get rid of all your opponent's pieces, by jumping over them. To jump over a piece there has to be an empty space on the other side.

At the start you can only move in one direction, that is up the board if you are White or down if you are Black. To move a piece the letter for its column and the number for the row is entered. Next the destination square is given in the same way.

If it is not possible to jump over a piece then any one of your own draughts is advanced by one square to another square of the same colour. If a piece reaches the other side then it becomes a 'king' or 'crown'. This means that it can move up and down the board and so is more powerful.

The fact that you have to take a piece when possible does not apply according to some sets of rules. Some authorities require that should an opponent's piece not be taken when possible, then the offender loses that piece. This is known as 'huffing'.

In this micro version the program will not accept an instruction

which simply moves a piece when there is a jump move available. The indicator will continue to show the same colour player until the jump sequence is complete.

A game can be saved to tape at any stage by pressing [grph] S. A previously saved game is loaded at any time by [grph] L. At any time, either during a game or after loading, the step mode can be selected by pressing [grph] 3. Pressing 'Space' will then step through from the beginning. Pressing [grph] 3 will end step mode and allow the game to be played on from that position.

```

1000 REM <<<<< DRAUGHTS <> ISSI >>
1010 REM
1020 SCREEN 1,0,0
1030 COLOR 15,6,6
1040 CLEAR 2000
1050 KEY OFF
1060 RESTORE
1070 FOR CH=182 TO 199
1080 FOR N=0 TO 7
1090 READ B
1100 IF CH<190 THEN B=B XOR 255
1110 VPOKE 8*CH+N,B
1120 NEXT N
1130 NEXT CH
1140 VPOKE &H2016,31:VPOKE &H2017,31
1150 GOTO 1370
1160 REM -----
1170 DATA 255,255,255,255,240,143,127,
,31
1180 DATA 255,255,255,255,15,241,254,
206
1190 DATA 66,80,85,85,133,240,255,255
1200 DATA 194,30,226,30,241,15,255,25
5
1210 DATA 255,255,255,255,240,128,0,0
1220 DATA 255,255,255,255,15,1,0,48
1230 DATA 67,80,85,85,133,240,255,255
1240 DATA 194,28,224,0,1,15,255,255
1250 REM
1260 DATA 0,1,9,87,99,118,125,255
1270 DATA 0,128,144,234,198,110,58,18
7
1280 DATA 0,1,9,86,98,86,68,128
1290 DATA 0,128,144,106,70,106,26,1

```

```

1300 DATA 0,0,15,16,32,35,36,36
1310 DATA 0,0,240,8,4,196,36,36
1320 DATA 36,36,35,32,16,15,0,0
1330 DATA 36,36,196,4,8,240,0,0
1340 DATA 0,0,255,0,0,255,0,0
1350 DATA 36,36,36,36,36,36,36,36
1360 REM -----
1370 CLS:GOSUB 2160
1380 FOR N=1 TO 100
1390 Z$(N,1)=" "
1400 Z$(N,2)=" "
1410 NEXT N
1420 DIM C(8,8)
1430 GOSUB 1810
1440 GOSUB 2210:GOSUB 2380
1450 LOCATE 5,19:PRINT STRING$(14," "
)
1460 LOCATE 5,20:PRINT STRING$(14," "
)
1470 COU=1
1480 GOTO 1620
1490 GAM=0
1500 FOR N=1 TO 8
1510 FOR M=1 TO 8
1520 IF M$(N,M)="B" OR M$(M,N)="D" TH
EN GAM=1
1530 NEXT M:NEXT N
1540 IF GAM=0 THEN GOTO 1660
1550 GAM=0
1560 FOR N=1 TO 8
1570 FOR M=1 TO 8
1580 IF M$(N,M)="A" OR M$(M,N)="C" TH
EN GAM=1
1590 NEXT M:NEXT N
1600 IF GAM=0 THEN GOTO 1710
1610 RETURN
1620 GOSUB 3770
1630 GOTO 1620
1640 REM >>>>>>> BLACK WIN <<<<<<<
1650 REM
1660 CLS
1670 LOCATE 14,9:PRINT "BLACK won."
1680 GOTO 1730
1690 REM >>>>>>> WHITE WIN <<<<<<<
1700 REM

```

```

1710 CLS
1720 LOCATE 14,9:PRINT "WHITE won."
1730 PLAY "L8CDEFGCDEFG"
1740 LOCATE 4,23:PRINT "Do you want a
      replay (y/n) ?"
1750 A$=INPUT$(1):A$=CHR$(ASC(A$)AND
223)
1760 IF A$="Y" THEN GOSUB 4690:GOTO 1
430
1770 IF A$="N" THEN RUN
1780 GOTO 1750
1790 REM >>>>>>> SCREEN <<<<<<<<<
1800 REM
1810 COLOR 15,4,4
1820 FOR A=0TO7:VPOKE8*214+A,255:NEXT
      A
1830 FOR Y=2 TO 17 STEP 4
1840 FOR X=2 TO 17 STEP 4
1850 LOCATE X,Y:PRINTCHR$(214);CHR$(2
14);"  "
1860 LOCATE X,Y+1:PRINTCHR$(214);CHR$
(214);"  "
1870 LOCATE X,Y+2:PRINT "  ";CHR$(214
);CHR$(214)
1880 LOCATE X,Y+3:PRINT "  ";CHR$(214
);CHR$(214)
1890 C(X/2,Y/2)=1:C(X/2+1,Y/2)=0
1900 C(X/2,Y/2+1)=0:C(X/2+1,Y/2+1)=1
1910 NEXT X
1920 NEXT Y
1930 FOR X=2 TO 17
1940 LOCATE X,1:PRINT CHR$(198)
1950 LOCATE X,18:PRINT CHR$(198)
1960 NEXT X
1970 FOR Y=2 TO 17
1980 LOCATE 1,Y:PRINT CHR$(199)
1990 LOCATE 18,Y:PRINT CHR$(199)
2000 NEXT Y
2010 LOCATE 1,1:PRINT CHR$(194)
2020 LOCATE 18,1:PRINT CHR$(195)
2030 LOCATE 1,18:PRINT CHR$(196)
2040 LOCATE 18,18:PRINT CHR$(197)
2050 FOR Y=2 TO 16 STEP 2
2060 LOCATE Y,0:PRINT CHR$((Y/2)+96)
2070 LOCATE 19,Y:PRINT CHR$(48+9-Y/2)

```

```

2080 NEXT Y
2090 LOCATE 19,0:PRINT" DRAUGHTS "
2100 LOCATE 19,1:PRINT"===== "
2110 LOCATE 22,5:PRINT"From.."
2120 LOCATE 22,8:PRINT"To...."
2130 RETURN
2140 REM >>>>> INITIALISE <<<<<<<
2150 REM
2160 DIM M$(8,8)
2170 DIM Z$(100,2)
2180 RETURN
2190 REM >>>>> GAME START <<<<<<<
2200 REM
2210 FOR N=1 TO 8
2220 FOR M=1 TO 8
2230 M$(N,M)="0"
2240 X$(N,M)="0"
2250 NEXT M
2260 NEXT N
2270 FOR N=1 TO 8 STEP 2
2280 M$(1,N+1)="A"
2290 M$(2,N)="A"
2300 M$(3,N+1)="A"
2310 M$(8,N)="B"
2320 M$(7,N+1)="B"
2330 M$(6,N)="B"
2340 NEXT N
2350 RETURN
2360 REM >>>>>> DRAW PIECES <<<<<<
2370 REM
2380 X=2:Y=2
2390 FOR N=1 TO 8
2400 FOR M=1 TO 8
2410 C$=M$(N,M)
2420 IF C$="0" THEN 2450
2430 LOCATE X,Y
2440 GOSUB 2520
2450 X=X+2
2460 NEXT M
2470 X=2:Y=Y+2
2480 NEXT N
2490 RETURN
2500 REM >>>>>>>> PIECE <<<<<<<<<
2510 REM
2520 LOCATE X,Y

```

```

2530 D$=CHR$(182)+CHR$(183):D1$=CHR$(
184)+CHR$(185)
2540 IF C$="B" THEN D$=CHR$(186)+CHR$
(187):D1$=CHR$(188)+CHR$(189)
2550 IF C$="C" THEN D$=CHR$(192)+CHR$
(193)
2560 IF C$="D" THEN D$=CHR$(190)+CHR$
(191):D1$=CHR$(188)+CHR$(189)
2570 IF C$="Ø" THEN D$=" ":D1$=" "
2580 PRINT D$;
2590 LOCATE X,Y+1
2600 PRINT D1$;
2610 RETURN
2620 REM >>>>>>>>>> INPUT <<<<<<<<<<<
2630 LOCATE 24,4:IF PLA=1 THEN PRINT "
WHITE" ELSE PRINT"BLACK"
2640 LOCATE 25,6:PRINT " "
2650 LOCATE 25,9:PRINT " "
2660 LOCATE 25,6
2670 GOSUB 2750
2680 F$=N$
2690 LOCATE 25,9
2700 GOSUB 2750
2710 S$=N$
2720 RETURN
2730 REM >>>>>>>>>> LOCATION <<<<<<<<
2740 REM
2750 N$=" "
2760 A$=INKEY$:IF A$=""THEN GOTO 2760
2770 A$=CHR$(ASC(A$)AND 223)
2780 IF A$="Q"THEN RUN
2790 IF A$=CHR$(210) THEN GOSUB 3860
2800 IF A$=CHR$(200) THEN GOSUB 4080
2810 IF A$=CHR$(154) AND COUNT>1 THEN
Z$(COU,1)="FF":PLAY "L8CDE":GOTO 469
Ø
2820 IF A$<"A" OR A$>"H"THEN 2760
2830 PLAY "L8C":N$=A$:PRINT N$;
2840 A$=INKEY$
2850 IF A$<"1" OR A$>"9" THEN 2840
2860 PLAY "L8F":N$=N$+A$:PRINT A$;
2870 LOCATE Ø,22:PRINT STRING$(12,32)
2880 RETURN
2890 REM >>>>>>>>>> MOVE <<<<<<<<<<
2900 REM

```

```

2910 GOSUB 3140
2920 RO=S1:NC=S:GOSUB 3000
2930 M$(S1,S)=M$(F1,F)
2940 IF S1=1 AND M$(1,S)="B" THEN M$(
1,S)="D":HUFF=0
2950 IF S1=8 AND M$(8,S)="A" THEN M$(
8,S)="C":HUFF=0
2960 RO=S1:NC=S:GOSUB 3000
2970 M$(F1,F)="0"
2980 RO=F1:NC=F:GOSUB 3000
2990 RETURN
3000 X=2:Y=2
3010 FOR N=1 TO 8
3020 FOR M=1 TO 8
3030 IF N=RO AND M=NC THEN GOSUB 3090
3040 X=X+2
3050 NEXT M
3060 X=2:Y=Y+2
3070 NEXT N
3080 RETURN
3090 C#=M$(N,M)
3100 REM IF C#="" THEN RETURN
3110 GOSUB 2520
3120 RETURN
3130 REM >>>>>>>>> CONVERT <<<<<<<<
3140 F=ASC(LEFT$(F$,1))
3150 F=F-64
3160 F1=VAL(RIGHT$(F$,1))
3170 F1=9-F1
3180 S=ASC(LEFT$(S$,1))
3190 S=S-64
3200 S1=VAL(RIGHT$(S$,1))
3210 S1=9-S1
3220 RETURN
3230 REM >>>>>>>>> CONVERT BACK <<<<<<
3240 R#=CHR$(NC+64)
3250 R#=R#+STR$(9-LIN)
3260 RETURN
3270 REM >>>>>>>>> VALIDATE <<<<<<<
3280 REM
3290 GOSUB 3140:HUFF=0
3300 P#="B":P1#="D"
3310 IF PLA=2 THEN P#="A":P1#="C"
3320 FOR Y=1 TO 8
3330 FOR X=1 TO 8

```

```

3340 IF HUFF=1 THEN GOTO 3370
3350 IF M$(Y,X)<>P$ AND M$(Y,X)<>P1$
THEN GOTO 3370
3360 GOSUB 4340
3370 NEXT X
3380 NEXT Y
3390 CHEC=1
3400 C$=M$(F1,F):D$=M$(S1,S)
3410 IF C$="" THEN RETURN
3420 IF (C$="A" OR C$="C") AND PLA=1
THEN RETURN
3430 IF (C$="B" OR C$="D") AND PLA=2
THEN RETURN
3440 IF D$<>"0" THEN RETURN
3450 IF C$="A" AND S1<F1 THEN RETURN
3460 IF C$="B" AND S1>F1 THEN RETURN
3470 XD=S-F:YD=S1-F1
3480 IF SGN(XD)*(XD)=1 AND SGN(YD)*(Y
D)=1 THEN GOTO 3590
3490 IF SGN(XD)*(XD)<>SGN(YD)*(YD) TH
EN RETURN
3500 X$=M$(F1+(YD/2),F+(XD/2))
3510 IF (X$="A" OR X$="C") AND PLA=2
THEN RETURN
3520 IF (X$="B" OR X$="D") AND PLA=1
THEN RETURN
3530 IF X$="" THEN RETURN
3540 RO=F1+(YD/2):NC=F+(XD/2)
3550 M$(RO,NC)="0":GOSUB 3000
3560 GOSUB 3240
3570 Z$(COU,1)=R$:Z$(COU,2)=R$
3580 COU=COU+1:GOTO 3600
3590 IF HUFF=1 THEN RETURN
3600 CHEC=0:RETURN
3610 REM >>>>>>>>> TURN <<<<<<<<<<<
3620 REM
3630 GOSUB 2630:HUFF=0
3640 GOSUB 3290
3650 IF CHEC=0 THEN GOTO 3680
3660 LOCATE 0,22:PRINT "INVALID MOVE"
3670 PLAY "L4GEC":GOTO 3630
3680 GOSUB 2910
3690 Z$(COU,1)=F$:Z$(COU,2)=S$
3700 COU=COU+1:IF COU=101 THEN COU=1
3710 IF HUFF=0 THEN GOTO 3740

```



```
3720 Y=S1:X=S:GOSUB 4340
3730 IF HUFF=1 THEN GOTO 4500
3740 HUFF=0:RETURN
3750 REM >>>>>> BOTH PLAYERS <<<<<<
3760 REM
3770 PLA=1
3780 GOSUB 3630
3790 GOSUB 1490
3800 PLA=2
3810 GOSUB 3630
3820 GOSUB 1490
3830 RETURN
3840 REM >>>>>>>>> SAVE <<<<<<<<<<<
3850 REM
3860 Z$(COU,1)="FF"
3870 G$=" ":PLAY "CDCDCDE"
3880 FOR N=1 TO 8
3890 FOR M=1 TO 8
3900 G$=G$+M$(N,M)
3910 NEXT M
3920 NEXT N
3930 H$=" ":I$=" "
3940 FOR N=1 TO 100
3950 H$=H$+Z$(N,1)
3960 I$=I$+Z$(N,2)
3970 NEXT N
3980 H$=H$+STR$(COU)
3990 OPEN "CAS:DRA"FOR OUTPUT AS 1
4000 PRINT #1,G$
4010 PRINT #1,H$
4020 PRINT #1,I$
4030 CLOSE #1
4040 PLAY "EDEDED"
4050 RETURN
4060 REM >>>>>>>>> LOAD <<<<<<<<<<<
4070 REM
4080 G$=" "
4090 PLAY "CDCDCDE":FOR N=1 TO 999:NE
XT
4100 OPEN "CAS:DRA"FOR INPUT AS 1
4110 LINE INPUT #1,G$
4120 LINE INPUT #1,H$
4130 LINE INPUT #1,I$
4140 CLOSE #1
4150 FOR N=1 TO 8
```

```

4160 FOR M=1 TO 8
4170 PLAY "L8"+CHR$(66+M/2)
4180 M$(N,M)=LEFT$(G$,1)
4190 G$=RIGHT$(G$,LEN(G$)-1)
4200 NEXT M
4210 NEXT N
4220 FOR N=1 TO 100
4230 Z$(N,1)=LEFT$(H$,2)
4240 H$=RIGHT$(H$,LEN(H$)-2)
4250 Z$(N,2)=LEFT$(I$,2)
4260 I$=RIGHT$(I$,LEN(I$)-2)
4270 NEXT N
4280 COU=VAL(H$)
4290 PLAY "EDEDED"
4300 GOSUB 1810:GOSUB 2380
4310 LOCATE 25,6:RETURN
4320 REM >>>>>>>>> HUFF <<<<<<<<<<
4330 REM
4340 HUFF=0
4350 H1$="A":H2$="C"
4360 IF PLA=2 THEN H1$="B":H2$="D"
4370 IF Y<3 OR M$(Y,X)="A" THEN GOTO
4420
4380 IF X<3 THEN GOTO 4400
4390 IF (M$(Y-1,X-1)=H1$OR M$(Y-1,X-1)
)=H2$) AND M$(Y-2,X-2)="0" THEN HUFF=
1
4400 IF X>6 THEN GOTO 4420
4410 IF (M$(Y-1,X+1)=H1$ OR M$(Y-1,X+
1)=H2$) AND M$(Y-2,X+2)="0" THEN HUFF
=1
4420 IF Y>6 OR M$(Y,X)="B" THEN GOTO
4470
4430 IF X<3 THEN GOTO 4450
4440 IF (M$(Y+1,X-1)=H1$ OR M$(Y+1,X-
1)=H2$) AND M$(Y+2,X-2)="0" THEN HUFF
=1
4450 IF X>6 THEN GOTO 4470
4460 IF (M$(Y+1,X+1)=H1$ OR M$(Y+1,X+
1)=H2$) AND M$(Y+2,X+2)="0" THEN HUFF
=1
4470 RETURN
4480 REM >>>>>>>>> GO AGAIN <<<<<<<
4490 REM
4500 LOCATE 25,6:PRINT S$

```

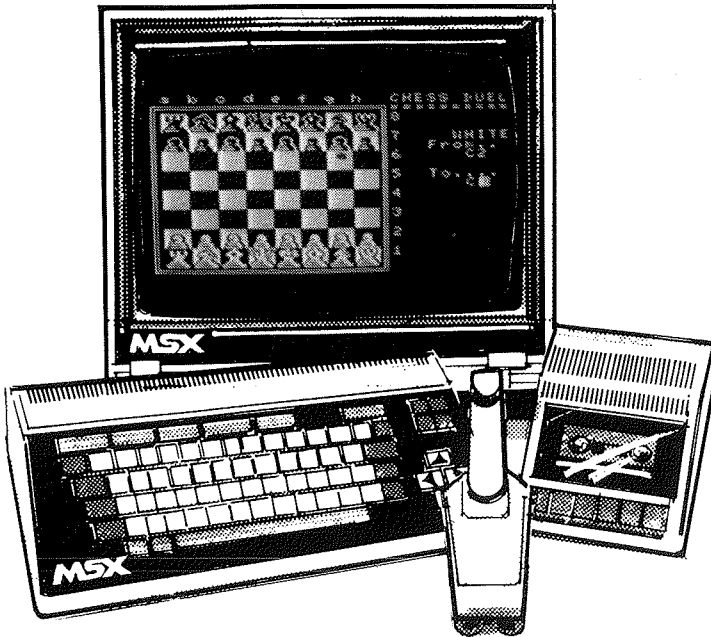
```

4510 F$=S$
4520 LOCATE 25,9:PRINT "  "
4530 LOCATE 25,9
4540 GOSUB 2750
4550 S$=N$
4560 LOCATE 0,22:PRINT STRING$(12,32)
4570 GOSUB 3140:GOSUB 3390
4580 IF CHEC=0 THEN GOTO 4610
4590 LOCATE 0,22:PRINT "INVALID MOVE"
4600 PLAY "L8GECGEC":GOTO 4520
4610 GOSUB 2910
4620 Z$(COU,1)=F$:Z$(COU,2)=S$
4630 COU=COU+1:IF COU=101 THEN COU=1
4640 Y=S1:X=S:GOSUB 4340
4650 IF HUFF=1 THEN GOTO 4500
4660 HUFF=0:GOTO 3740
4670 REM >>>>>>>>> REPLAY <<<<<<<<<<<
4680 REM
4690 COU=0:PLA=1
4700 GOSUB 2210
4710 GOSUB 1810:GOSUB 2380
4720 COU=COU+1
4730 IF Z$(COU,1)="FF" THEN 4910
4740 F$=Z$(COU,1):S$=Z$(COU,2)
4750 GOSUB 3140:GOSUB 2910
4760 IF F$=S$ THEN GOTO 4720
4770 COU=COU+1
4780 A$=INKEY$
4790 IF A$="" THEN 4780
4800 PLA=2
4810 IF A$=CHR$(186) THEN 4910
4820 IF Z$(COU,1)="FF" THEN GOTO 4910
4830 F$=Z$(COU,1):S$=Z$(COU,2)
4840 GOSUB 3140:GOSUB 2910
4850 IF F$=S$ THEN GOTO 4720
4860 A$=INKEY$
4870 IF A$="" THEN 4860
4880 PLA=1
4890 IF A$=CHR$(186) THEN 4910
4900 GOTO 4720
4910 PLAY "L8EDC"
4920 IF PLA=1 THEN GOTO 3770
4930 GOTO 3800

```

8

Chess Duel



Computer Moderated Micro Chess

It will come as a surprise to many that before the age of chess computers and the microcomputer program, the game of chess was originally played by two human beings.

An intelligent chess playing program would take up an entire book on it's own, so it was decided to develop a program which would be of genuine use to anyone who enjoyed playing chess or would like to learn more about it. Chess clubs could also find the program helpful.

The program can be used simply as a replacement for a chess board and pieces. The two players enter their moves in usual standard chess notation. However, extra features have been included to produce worthwhile advantages over the traditional board and pieces.

The computer will examine all entries for legality of the move. Only moves which are the correct colour, are from occupied squares and move to a square which can be reached by that piece, will be accepted. Pieces which are 'taken' in the process are removed from the board.

The program does not indicate 'check' or 'mate' and will allow

moves which reveal check. It is up to the humans to watch out for these. *En passant* is not automatically dealt with but the 'remove' option may be used by a player to take the pawn if required on his move.

Castling is handled by the program. The present position of the king and rook have to be correct, with no pieces between them, for castling to be implemented. The routine will still castle if a piece has previously been moved but returned to the correct square. Moving through check is not monitored. These latter conditions must be detected by the players.

A useful feature is that the initial board may be set up for chess problems or chess variants and games of up to 100 moves can be stored and analysed step by step. Play can continue from any stage of a saved game or the present game can be re-started and then played from any subsequent move. Pawns are automatically promoted to a queen when they reach the appropriate rank.

Summary of commands and options

1. Set up board. Enter 'Y' when asked. Then sequentially from the top left, place each piece using a single letter. Move on using 'Space'.

White pieces: P=pawn, R=rook, N=knight, K=king, Q=queen, B=bishop.

Black pieces: the same but with 'grph' held.

White always plays up the board.

2. Remove piece. 'Grph' 'R' then enter the position. Play continues with the next player.

3. Castle. 'Grph' 'C' then direction 'L' or 'R'.

4. Save game. 'Grph' 'S'. You must ensure cassette Record and Play are on before saving as prompts are not displayed.

5. Load Game. 'Grph' 'L', then press 'Play'. This loads the first file found.

6. Step through game. 'Grph' '3' initiates step mode. 'Space' steps through each move. 'Grph' '3' then ends mode.

7. Quit. 'Grph' 'Q' resets for new game.

```
1000 REM >>CHESS ISSI/BOOTS/ANDY<<
1010 REM
1020 SCREEN 1,0,0: CLEAR 2000
1030 COLOR 15,1,1: KEY OFF
1040 DIM C(8,8), M$(8,8), X$(8,8), Z$(10
0,2)
```

```

1050 GOSUB 5490:GOSUB 4940
1060 GOSUB 4390
1070 LOCATE 5,19:PRINT"DO YOU WISH TO
"
1080 LOCATE 5,20:PRINT"SET A BOARD ?"
;
1090 A$=INPUT$(1):BEEP
1100 LOCATE 5,19:PRINT STRING$(14," "
)
1110 LOCATE 5,20:PRINT STRING$(14," "
)
1120 IF A$="y" OR A$="A" THEN GOSUB 3
800:IF CHE=0 THEN 1140 ELSE 1060
1130 GOSUB 1190:GOSUB 1420
1140 COU=1
1150 GOSUB 3740
1160 GOTO 1150
1170 END
1180 REM >>> GAME START <<<
1190 FOR N=1 TO 8
1200 FOR M=1 TO 8
1210 M$(N,M)="0"
1220 X$(N,M)="0"
1230 NEXT M
1240 NEXT N
1250 FOR N=1 TO 8
1260 M$(2,N)="p":X$(2,N)="p"
1270 M$(7,N)="P":X$(7,N)="P"
1280 NEXT N
1290 RESTORE 6090
1300 FOR N=1 TO 8
1310 READ A$
1320 M$(8,N)=A$
1330 X$(8,N)=A$
1340 C$=CHR$(ASC(A$)+32):A$=C$
1350 M$(1,N)=A$
1360 X$(1,N)=A$
1370 NEXT N
1380 GOSUB 5440
1390 RETURN
1400 REM >>>> DRAW PIECES <<<<<
1410 REM
1420 X=2:Y=2
1430 FOR N=1 TO 8
1440 FOR M=1 TO 8

```

```

1450 C#=M$(N,M)
1460 IF C#="" THEN 1490
1470 LOCATE X,Y
1480 GOSUB 1550
1490 X=X+2
1500 NEXT M
1510 X=2:Y=Y+2
1520 NEXT N
1530 RETURN
1540 REM >>>> PIECE <<<<<
1550 LOCATE X,Y:XX=X:X=X/2:YY=Y:Y=Y/2
:IF C#="K"AND C(X,Y)=1 THEN D#=K#:D1#
=#K1# ELSE IF C#="k" THEN D#=K2#:D1#=K
3#
1560 IF C#="k"AND C(X,Y)=1 THEN D#=K4
#:D1#=K5# ELSE IF C#="k" THEN D#=K6#:
D1#=K7#
1570 IF C#="Q"AND C(X,Y)=1 THEN D#=Q#
:D1#=Q1# ELSE IF C#="Q" THEN D#=Q2#:D
1#=Q3#
1580 IF C#="q"AND C(X,Y)=1 THEN D#=Q4
#:D1#=Q5# ELSE IF C#="q" THEN D#=Q6#:
D1#=Q7#
1590 IF C#="B"AND C(X,Y)=1 THEN D#=B#
:D1#=B1# ELSE IF C#="B" THEN D#=B2#:D
1#=B3#
1600 IF C#="b"AND C(X,Y)=1 THEN D#=B4
#:D1#=B5# ELSE IF C#="b" THEN D#=B6#:
D1#=B7#
1610 IF C#="T"AND C(X,Y)=1 THEN D#=T#
:D1#=T1# ELSE IF C#="T" THEN D#=T2#:D
1#=T3#
1620 IF C#="t"AND C(X,Y)=1 THEN D#=T4
#:D1#=T5# ELSE IF C#="t" THEN D#=T6#:
D1#=T7#
1630 IF C#="R"AND C(X,Y)=1 THEN D#=R#
:D1#=R1# ELSE IF C#="R" THEN D#=R2#:D
1#=R3#
1640 IF C#="r"AND C(X,Y)=1 THEN D#=R4
#:D1#=R5# ELSE IF C#="r" THEN D#=R6#:
D1#=R7#
1650 IF C#="P"AND C(X,Y)=1 THEN D#=P#
:D1#=P1# ELSE IF C#="P" THEN D#=P2#:D
1#=P3#
1660 IF C#="p"AND C(X,Y)=1 THEN D#=P4

```

```

P:P5$ ELSE IF C$="p" THEN D$=P6$:
D1$=P7$
1670 IF C$="0" AND C(X,Y)=1 THEN D$=C
HR$(214)+CHR$(214):D1$=D$ ELSE IF C$=
"0" THEN D$=" ":D1$=D$
1680 PRINT D$;
1690 X=XX:Y=YY
1700 LOCATE X,Y+1
1710 PRINT D1$;
1720 RETURN
1730 REM >>> INPUT <<<
1740 LOCATE 24,4:IF PLA=1 THEN PRINT"
WHITE" ELSE PRINT"BLACK"
1750 LOCATE 25,6:PRINT" "
1760 LOCATE 25,9:PRINT" "
1770 LOCATE 25,6
1780 GOSUB 1850
1790 F$=N$
1800 IF CAS=1 THEN RETURN
1810 LOCATE 25,9
1820 GOSUB 1850
1830 S$=N$
1840 RETURN
1850 N$=" ":CAS=0
1860 A$=INPUT$(1)
1870 IF A$="Q" OR A$="q" THEN RUN
1880 IF A$=CHR$(188) THEN GOSUB 3200
1890 IF CAS=1 THEN RETURN
1900 IF A$=CHR$(200) THEN GOSUB 4730:
LOCATE 25,9
1910 IF A$=CHR$(210) THEN GOSUB 4610
1920 IF A$="R" OR A$="r" THEN GOTO 22
60
1930 A$=CHR$(ASC(A$) AND 223 )
1940 IF A$<"A" OR A$>"H" THEN GOTO 186
0
1950 PLAY "L8C":N$=A$:PRINT N$;
1960 A$=INPUT$(1)
1970 IF A$=CHR$(8) THEN PRINT CHR$(12
7);:GOTO 1850
1980 IF A$<"1" OR A$>"9" THEN 1960
1990 PLAY "L8G":N$=N$+A$:PRINT A$;
2000 RETURN
2010 REM >>> MOVE <<<
2020 GOSUB 2430

```



```

2030 CH#=M$(F1,F)
2040 M$(S1,S)=M$(F1,F)
2050 IF PLA=1 AND F1=1 AND M$(F1,F)="
p" THEN M$(F1,F)="q"
2060 IF PLA=2 AND F1=8 AND M$(F1,F)="
P" THEN M$(F1,F)="Q"
2070 C#=CH#
2080 RO=S1:CR=S:GOSUB 2130
2090 M$(F1,F)="0"
2100 RO=F1:CR=F:GOSUB 2130
2110 M$(F1,F)="0"
2120 RETURN
2130 X=2:Y=2
2140 FOR N=1 TO 8
2150 FOR M=1 TO 8
2160 IF N=RO AND M=CR THEN GOSUB 2220
2170 X=X+2
2180 NEXT M
2190 X=2:Y=Y+2
2200 NEXT N
2210 RETURN
2220 C#=M$(F1,F):LOCATE X,Y:GOSUB 155
0
2230 RETURN
2240 RETURN
2250 REM >>> REMOVE <<<
2260 LOCATE 20,19:PRINT"Piece "
2270 LOCATE 20,20:F#=INPUT$(1):A#=CHR
$(ASC(F#) AND 223)
2280 IF A#<"A" OR A#>"H" THEN GOTO 22
70 ELSE PRINT A#
2290 LOCATE 21,20:G#=INPUT$(1)
2300 IF G#<"1" OR G#>"9" THEN GOTO 22
90 ELSE PRINT G#
2310 F#=A#+G#
2320 LOCATE 20,19:PRINT " "
2330 LOCATE 20,20:PRINT " "
2340 S#="A1":GOSUB 2430
2350 M$(F1,F)="0"
2360 RO=F1:CR=F:GOSUB 2130
2370 Z$(COU,1)=F#:Z$(COU,2)=F#
2380 COU=COU+1
2390 IF COU=101 THEN COU=1
2400 IF PLA=1 THEN PLA=2:GOSUB 3590
2410 RETURN 1150

```

```

2420 REM >>>   CONVERT   <<<
2430 F=ASC(LEFT$(F$,1))
2440 F=F-64
2450 F1=9-VAL(RIGHT$(F$,1))
2460 S=ASC(LEFT$(S$,1))
2470 S=S-64
2480 S1=9-VAL(RIGHT$(S$,1))
2490 RETURN
2500 F#=CHR$(F+64)+STR$(9-F1)
2510 S#=CHR$(S+64)+STR$(9-S1)
2520 Z$(COU,1)=F#
2530 Z$(COU,2)=S#
2540 COU=COU+1:RETURN
2550 REM >>>   VALIDATE   <<<
2560 GOSUB 2430
2570 CHE=1
2580 C#=M$(F1,F):D#=M$(S1,S)
2590 IF C#="" THEN RETURN
2600 IF C#>"Z" AND PLA=1 THEN RETURN
2610 IF C#<"a" AND PLA=2 THEN RETURN
2620 IF D#="" THEN 2650
2630 IF D#>"Z" AND PLA=2 THEN RETURN
2640 IF D#<"a" AND PLA=1 THEN RETURN
2650 IF ASC(C#)>ASC("Z") THEN C#=CHR$(
ASC(C#)-32)
2660 IF ASC(D#)>ASC("Z") THEN D#=CHR$(
ASC(D#)-32)
2670 IF D#="K" THEN RETURN
2680 IF C#="P" THEN 2790
2690 IF C#="R" THEN 2920
2700 IF C#="T" THEN 2950
2710 IF C#="B" THEN 3060
2720 IF C#="Q" THEN 3110
2730 IF C#="K" THEN 3170
2740 CHE=0
2750 IF C#="K" OR C#="P" OR C#="T" TH
EN RETURN
2760 GOSUB 3480
2770 RETURN
2780 REM >>>   PAWN   <<<
2790 IF PLA=2 AND F1>S1 THEN RETURN
2800 IF PLA=1 AND S1>F1 THEN RETURN
2810 DIS=SGN(S1-F1)*(S1-F1):IF DIS=1
THEN 2870
2820 IF DIS>2 THEN RETURN

```

```

2830 IF S<>F THEN RETURN
2840 IF PLA=2 AND F1=2 AND M$(3,F)=""
" THEN 2900
2850 IF PLA=1 AND F1=7 AND M$(6,F)=""
" THEN 2900
2860 RETURN
2870 IF S=F AND D$="" THEN 2900
2880 IF SGN(S-F)*(S-F)=1 AND D$<>"0"
THEN 2900
2890 RETURN
2900 GOTO 2740
2910 REM >>> ROOK <<<
2920 IF F1<>S1 AND F<>S THEN RETURN
2930 GOTO 2740
2940 REM >>> KNIGHT <<<
2950 IF S1=F1+2 AND S=F-1 THEN 3040
2960 IF S1=F1+2 AND S=F+1 THEN 3040
2970 IF S1=F1-2 AND S=F-1 THEN 3040
2980 IF S1=F1-2 AND S=F+1 THEN 3040
2990 IF S1=F1+1 AND S=F-2 THEN 3040
3000 IF S1=F1+1 AND S=F+2 THEN 3040
3010 IF S1=F1-1 AND S=F-2 THEN 3040
3020 IF S1=F1-1 AND S=F+2 THEN 3040
3030 RETURN
3040 GOTO 2740
3050 REM >>> BISHOP <<<
3060 XD=F-S:XD=SGN(XD)*XD
3070 YD=F1-S1:YD=SGN(YD)*YD
3080 IF XD<>YD THEN RETURN
3090 GOTO 2740
3100 REM >>> QUEEN <<<
3110 XD=F-S:XD=SGN(XD)*XD
3120 YD=F1-S1:YD=SGN(YD)*YD
3130 IF XD=YD THEN 3150
3140 IF F1<>S1 AND F<>S THEN RETURN
3150 GOTO 2740
3160 REM >>> KING <<<
3170 IF SGN(F1-S1)*(F1-S1)<>1 AND SGN
(F-S)*(F-S)<>1 THEN RETURN
3180 GOTO 2740
3190 REM >>>> CASTLING <<<<<<<
3200 PRINT "C";
3210 A=ASC(INPUT$(1)):A=A AND 223
3220 IF A<>76 AND A<>82 THEN 3210
3230 A$=CHR$(A):PRINT A$;

```

```

3240 N$="C"+A$:CAS=1:RETURN
3250 REM >>> castle move <<<<<
3260 S$="A8":A#=RIGHT$(F$,1):GOSUB 24
30:CHEC=1
3270 IF (PLA=1 AND A$="L" AND M$(8,1)
<>"R") OR (PLA=1 AND A$="R" AND M$(8,
8)<>"R") THEN RETURN
3280 IF (PLA=2 AND A$="L" AND M$(1,1)
<>"r") OR (PLA=2 AND A$="R" AND M$(1,
8)<>"r") THEN RETURN
3290 IF PLA=2 AND M$(1,5)<>"k" THEN R
ETURN
3300 IF PLA=1 AND M$(8,5)<>"K" THEN R
ETURN
3310 AU=1:EN=7:IF PLA=1 THEN LIN=8 EL
SE LIN=1
3320 IF A$="L" THEN AU=-1:EN=2
3330 FOR N=5+AU TO EN STEP AU
3340 IF M$(LIN,N)<>"0" THEN N=EN:RETU
RN
3350 NEXT N
3360 PLAY "L8CDEGFCEEG"
3370 F1=LIN:F=5+AU:S1=F1:S=EN
3380 IF AU=-1 THEN S=S+1
3390 M$(F1,F)="R":M$(S1,S)="K"
3400 M$(F1,5)="0":M$(S1,EN+AU)="0"
3410 IF LIN=1 THEN M$(F1,F)="r":M$(S1
,S)="k"
3420 Z$(COU,1)="CC":Z$(COU,2)="CC":CO
U=COU+1
3430 FOR F=1 TO 8
3440 RO=F1:CR=F:GOSUB 2130
3450 NEXT F
3460 CHEC=0:RETURN
3470 REM >>> CHECK PATH <<<
3480 ND=0:MD=0
3490 IF F1>S1 THEN ND=-1
3500 IF S1>F1 THEN ND=1
3510 IF F>S THEN MD=-1
3520 IF S>F THEN MD=1
3530 CN=F1:CM=F
3540 CN=CN+ND:CM=CM+MD
3550 IF CN=S1 AND CM=S THEN RETURN
3560 IF M$(CN,CM)<>"0" THEN CHE=1:RET
URN

```

```

3570 GOTO 3540
3580 REM >>> TURN <<<
3590 GOSUB 1740
3600 LOCATE 22,11:PRINT " "
3610 LOCATE 24,12:PRINT " "
3620 IF CAS=1 THEN GOSUB 3260 ELSE GO
SUB 2560
3630 IF CHE=0 THEN 3670
3640 LOCATE 22,11:PRINT "INVALID"
3650 LOCATE 24,12:PRINT"MOVE"
3660 GOTO 3590
3670 IF CAS=1 THEN 3720
3680 Z$(COU,1)=F$
3690 Z$(COU,2)=S$
3700 COU=COU+1:IF COU=101 THEN COU=1
3710 GOSUB 2020
3720 RETURN
3730 REM >>> BOTH PLAYERS <<<
3740 PLA=1
3750 GOSUB 3590
3760 PLA=2
3770 GOSUB 3590
3780 RETURN
3790 REM >>> SET BOARD <<<
3800 X=2:Y=2
3810 FOR N=1 TO 8
3820 FOR M=1 TO 8
3830 M$(N,M)="0":X$(N,M)="0"
3840 GOSUB 3980
3850 X=X+2
3860 NEXT M
3870 X=2:Y=Y+2
3880 NEXT N
3890 LOCATE 20,18:PRINT"IS THIS"
3900 LOCATE 20,19:PRINT"O.K. ?"
3910 LOCATE 20,20:PRINT " "
3920 A$=INKEY$:IF A$=""THEN 3920
3930 IF A$>"Z" THEN A$=CHR$(ASC(A$)-3
2)
3940 IF A$="Y" THEN CHE=0:LOCATE 20,1
8:PRINT " ":LOCATE 20,19:PRINT "
":RETURN
3950 IF A$="N" THEN 3800
3960 GOTO 3920
3970 REM >>> INPUT <<<

```

```

3980 LOCATE X,Y:PRINT"?";
3990 D$=INKEY$:IF D$=""THEN GOTO 3990
4000 IF C$=" " THEN D$="0":GOTO 4050
4010 IF D$>"Z" THEN C#=CHR$(ASC(D$)-3
2) ELSE C#=D$
4020 IF C#<>"R" AND C#<>"N" AND C#<>"
B" AND C#<>"Q" AND C#<>"K" AND C#<>"P
" THEN GOTO 3990
4030 IF D$="n" THEN D$="t"
4040 IF D$="N" THEN D$="T"
4050 PLAY "LBC":M$(N,M)=D$
4060 X$(N,M)=D$
4070 C#=D$
4080 LOCATE X,Y
4090 GOSUB 1550
4100 RETURN
4110 REM >>>  REPLAY  <<<
4120 PLAY "L8CEGCEG":COU=0:PLA=1
4130 FOR N=1 TO 8:FOR M=1 TO 8
4140 M$(N,M)=X$(N,M)
4150 NEXT M:NEXT N
4160 GOSUB 4390:GOSUB 1420
4170 COU=COU+1
4180 IF Z$(COU,1)="FF" THEN 4390
4190 F#=Z$(COU,1):S#=Z$(COU,2)
4200 GOSUB 2430:GOSUB 2020
4210 COU=COU+1
4220 IF Z$(COU,1)="CC" THEN 4170
4230 A$=INKEY$
4240 IF A$="" THEN 4230
4250 PLA=2:PLAY "L8DE"
4260 IF A#=CHR$(154) THEN 4390
4270 IF Z$(COU,1)="FF" THEN4390
4280 F#=Z$(COU,1):S#=Z$(COU,2)
4290 GOSUB 2430:GOSUB 2020
4300 IF Z$(COU+1,1)="CC" THEN COU=COU
+2:GOTO 4270
4310 A$=INKEY$
4320 IF A$="" THEN 4310
4330 PLA=1:PLAY "L8DE"
4340 IF A#=CHR$(154) THEN 4390
4350 GOTO 4170
4360 IF PLA=1 THEN 1150
4370 GOSUB 3590:GOTO 1150
4380 REM >>>>  SCREEN  <<<<<

```

```

4390 COLOR 15,1,1
4400 FOR Y=2 TO 16 STEP 2
4410 LOCATE 1,Y:PRINT CHR$(220);BD$;C
HR$(220)
4420 LOCATE 1,Y+1:PRINT CHR$(220);BD$
;CHR$(220)
4430 BD$=RIGHT$(BD$,14)+LEFT$(BD$,2)
4440 NEXT Y
4450 LOCATE 1,1:PRINT CHR$(215);STRIN
G$(16,CHR$(219));CHR$(216)
4460 LOCATE 1,18:PRINT CHR$(217);STRI
NG$(16,CHR$(219));CHR$(218)
4470 FOR Y=2 TO 16 STEP 2
4480 LOCATE Y,0:PRINT CHR$((Y/2)+96)
4490 LOCATE 19,Y:PRINT CHR$(48+9-Y/2)
4500 NEXT Y
4510 LOCATE 19,0:PRINT"CHESS DUEL"
4520 LOCATE 19,1:PRINT"======"
4530 LOCATE 22,5:PRINT"From.."
4540 LOCATE 22,8:PRINT"To...."
4550 FOR X=1 TO 7 STEP 2:FOR Y=1 TO 7
STEP 2
4560 C(X,Y)=1:C(X+1,Y)=0
4570 C(X,Y+1)=0:C(X+1,Y+1)=1
4580 NEXT Y,X
4590 RETURN
4600 REM >>>>> SAVE <<<<<<<<<<<<<<<<<<<<<<
4610 Z$(COU,1)="FF":G$="":H$="":I$=""
4620 FOR N=1 TO 8:FOR M=1 TO 8
4630 G$=G$+M$(N,M)+X$(N,M)
4640 NEXT M,N
4650 FOR N=1 TO 100
4660 H$=H$+Z$(N,1):I$=I$+Z$(N,2)
4670 NEXT N
4680 H$=H$+STR$(COU):BEEP
4690 OPEN "CAS:CHESSF" FOR OUTPUT AS
1
4700 PRINT#1,G$:PRINT#1,H$:PRINT#1,I$
4710 CLOSE:BEEP:BEEP:RETURN
4720 REM >>>>> LOAD <<<<<<<<<<<<<<<<<<<<<<
4730 OPEN "CAS:CHESSF" FOR INPUT AS 1
4740 G$="":H$="":I$=""
4750 LINE INPUT#1,G$
4760 LINE INPUT#1,H$
4770 LINE INPUT#1,I$

```

```

4780 CLOSE
4790 FOR N=1 TO 8:FOR M=1 TO 8
4800 M$(N,M)=LEFT$(G$,1):X$(N,M)=MID$(
(G$,2,1)
4810 G$=RIGHT$(G$,LEN(G$)-2)
4820 NEXT M,N
4830 FOR N=1 TO 100
4840 Z$(N,1)=LEFT$(H$,2)
4850 H$=RIGHT$(H$,LEN(H$)-2)
4860 Z$(N,2)=LEFT$(I$,2)
4870 I$=RIGHT$(I$,LEN(I$)-2)
4880 NEXT N
4890 COU=VAL(H$)
4900 GOSUB 4390
4910 GOSUB 1420
4920 RETURN
4930 REM >>>>> initialise <<<<<<<<
4940 K$=CHR$(182)+CHR$(183)
4950 K1$=CHR$(192)+CHR$(193)
4960 K2$=CHR$(150)+CHR$(151)
4970 K3$=CHR$(160)+CHR$(161)
4980 K4$=CHR$(166)+CHR$(167)
4990 K5$=CHR$(176)+CHR$(177)
5000 K6$=CHR$(198)+CHR$(199)
5010 K7$=CHR$(208)+CHR$(209)
5020 Q$=CHR$(184)+CHR$(185)
5030 Q1$=CHR$(192)+CHR$(193)
5040 Q2$=CHR$(152)+CHR$(153)
5050 Q3$=CHR$(160)+CHR$(161)
5060 Q4$=CHR$(168)+CHR$(169)
5070 Q5$=CHR$(176)+CHR$(177)
5080 Q6$=CHR$(200)+CHR$(201)
5090 Q7$=CHR$(208)+CHR$(209)
5100 B$=CHR$(186)+CHR$(187)
5110 B1$=CHR$(192)+CHR$(193)
5120 B2$=CHR$(154)+CHR$(155)
5130 B3$=CHR$(160)+CHR$(161)
5140 B4$=CHR$(170)+CHR$(171)
5150 B5$=CHR$(176)+CHR$(177)
5160 B6$=CHR$(202)+CHR$(203)
5170 B7$=CHR$(208)+CHR$(209)
5180 T$=CHR$(188)+CHR$(189)
5190 T1$=CHR$(192)+CHR$(193)
5200 T2$=CHR$(156)+CHR$(157)
5210 T3$=CHR$(160)+CHR$(161)

```



```

5220 T4$=CHR$(172)+CHR$(173)
5230 T5$=CHR$(176)+CHR$(177)
5240 T6$=CHR$(204)+CHR$(205)
5250 T7$=CHR$(208)+CHR$(209)
5260 R$=CHR$(190)+CHR$(191)
5270 R1$=CHR$(192)+CHR$(193)
5280 R2$=CHR$(158)+CHR$(159)
5290 R3$=CHR$(160)+CHR$(161)
5300 R4$=CHR$(174)+CHR$(175)
5310 R5$=CHR$(176)+CHR$(177)
5320 R6$=CHR$(206)+CHR$(207)
5330 R7$=CHR$(208)+CHR$(209)
5340 P$=CHR$(194)+CHR$(195)
5350 P1$=CHR$(196)+CHR$(197)
5360 P2$=CHR$(162)+CHR$(163)
5370 P3$=CHR$(164)+CHR$(165)
5380 P4$=CHR$(178)+CHR$(179)
5390 P5$=CHR$(180)+CHR$(181)
5400 P6$=CHR$(210)+CHR$(211)
5410 P7$=CHR$(212)+CHR$(213)
5420 BD$=STRING$(2,CHR$(214))+SPACE$(
2)
5430 BD$=BD$+BD$:BD$=BD$+BD$
5440 FOR N=1 TO 100
5450 Z$(N,1)=SPACE$(2):Z$(N,2)=SPACE$(
2)
5460 NEXT N
5470 RETURN
5480 REM >>>>> def shapes <<<<<<<<
5490 FOR CH=150 TO 181
5500 FOR N=0 TO 7
5510 READ B
5520 IF CH>=166 THEN B=B XOR 255
5530 VPOKE 8*CH+N,B
5540 NEXT N
5550 IF CH=165 THEN RESTORE
5560 NEXT CH:RESTORE 5870
5570 FOR CH=182 TO 213
5580 FOR N=0 TO 7
5590 READ B
5600 IF CH>=198 THEN B=B XOR 255
5610 VPOKE 8*CH+N,B
5620 NEXT N
5630 IF CH=197 THEN RESTORE 5870
5640 NEXT CH

```

```

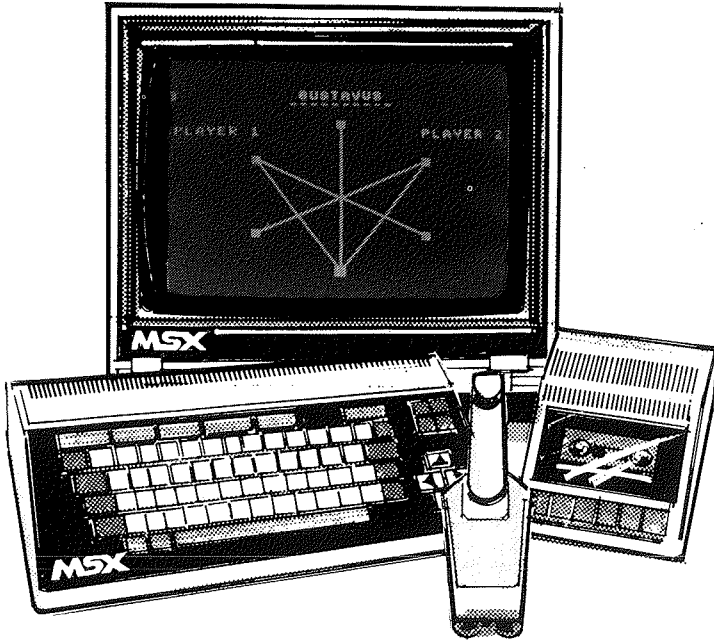
5650 FOR CH=215 TO 220
5660 FOR N=0 TO 7
5670 READ B:VPOKE 8*CH+N,B
5680 NEXT N,CH
5690 FOR A=0 TO 7:VPOKE8*214+A,255:NE
XT A
5700 RETURN
5710 DATA 1,3,29,63,63,57,28,7
5720 DATA 0,128,56,252,252,156,56,224
5730 DATA 1,19,81,91,109,55,27,15
5740 DATA 128,200,138,218,182,236,216
,240
5750 DATA 1,1,7,14,28,28,14,7
5760 DATA 128,128,224,240,120,120,240
,224
5770 DATA 0,0,3,7,15,31,9,3
5780 DATA 0,192,64,224,176,208,232,12
0
5790 DATA 0,0,25,25,25,31,9,3
5800 DATA 0,0,152,152,152,248,240,240
5810 DATA 7,7,4,31,16,127,64,127
5820 DATA 224,224,224,248,120,254,30,
254
5830 DATA 0,0,0,0,0,3,7,7
5840 DATA 0,0,0,0,0,192,224,224
5850 DATA 5,4,3,15,8,31,16,31
5860 DATA 224,224,192,240,112,248,56,
248
5870 DATA 253,227,221,191,191,187,222
,231
5880 DATA 127,131,59,253,253,157,59,2
31
5890 DATA 237,147,85,91,109,173,219,2
39
5900 DATA 183,201,170,218,182,239,219
,247
5910 DATA 253,249,247,238,220,220,238
,247
5920 DATA 191,159,239,247,251,251,247
,239
5930 DATA 255,254,249,247,239,223,233
,243
5940 DATA 63,223,95,239,183,215,235,1
23
5950 DATA 255,230,217,217,217,223,236
,239

```

5960 DATA 255, 103, 155, 155, 155, 251, 247
 , 247
 5970 DATA 247, 247, 228, 223, 144, 127, 64,
 127
 5980 DATA 239, 239, 231, 251, 121, 254, 30,
 254
 5990 DATA 255, 255, 255, 255, 252, 251, 247
 , 247
 6000 DATA 255, 255, 255, 255, 63, 223, 239,
 239
 6010 DATA 245, 250, 243, 239, 232, 239, 208
 , 223
 6020 DATA 239, 239, 207, 247, 119, 251, 59,
 251
 6030 DATA 0, 0, 63, 32, 47, 40, 40, 40
 6040 DATA 0, 0, 248, 8, 224, 40, 40, 40
 6050 DATA 40, 40, 47, 32, 63, 0, 0, 0
 6060 DATA 40, 40, 224, 8, 248, 0, 0, 0
 6070 DATA 0, 0, 255, 0, 255, 0, 0, 0
 6080 DATA 40, 40, 40, 40, 40, 40, 40, 40
 6090 DATA R, T, B, Q, K, B, T, R

9

Gustavus



A Novel Strategy Game for Two

As far as I am aware this is the first computer implementation of *Simple Simmons*, a game usually played with pencil and paper.

Simple Simmons or *Sim* was invented by a chap called Gustavus Simmons. We have decided to call this computer version of the game *Gustavus* as it sounds more grand than its paper cousin.

The game requires players alternately to draw a line between two points. The first to form a triangle of his own colour, whose points each rest on one of the original points, is the loser. The trick is to force your opponent into making the fatal triangle first.

The computer will take care of the turns together with the colour changes and will indicate the winner when a triangle is formed.

Lines are drawn by moving the marker clockwise or anti-clockwise, using either left or right cursor keys or joystick. When the marker is on the first point from which a line is required the space bar or fire button is pressed. A bleep signifies acceptance of the line origin. The marker is next moved to the destination location, then as the space bar or fire button is pressed the line will be drawn. If an invalid line is

attempted a sound will indicate the fact. New start and finish points will then be required.

The game cannot end in a draw and there is no apparent advantage gained by being the first player. This game is very good for developing spacial awareness!

```

1000 REM <<<<< GUSTAVUS >>>>>
1010 REM          ISSI
1020 SCREEN 2,0,0
1030 MAXFILES=2:OPEN"GRP:S"FOR OUTPUT
AS 1
1040 GOSUB 1280
1050 GOSUB 1170
1060 GOSUB 1370
1070 GOSUB 1840
1080 FOR N=1 TO 1000:NEXT N
1090 IF PLY=1 THEN WIN=2 ELSE WIN=1
1100 PRESET (80,182)
1110 IF WIN=2 THEN CO=7 ELSE CO=9
1120 COLOR CO,0:PRINT #1,"PLAYER ";WI
N;" WON !"
1130 A$=INKEY$:IF A$<>"P" THEN 1130
1140 GOTO 1050
1150 REM <<<<< SCREEN >>>>>
1160 REM
1170 KEY OFF
1180 COLOR 11,0,0:CLS
1190 PRESET (100,0):PRINT #1,"GUSTAVU
S":PRESET (92,8):PRINT #1,"-----
"
1200 FOR N=1 TO 6
1210 PUT SPRITE N,(P(N,1),P(N,2)),3,1
1220 NEXT N
1230 PRESET (8,32):COLOR 9:PRINT #1,"
PLAYER 1"
1240 PRESET (192,32):COLOR 7:PRINT #1
,"PLAYER 2"
1250 RETURN
1260 REM <<< INITIALIZE >>>
1270 REM
1280 DIM P(6,2)
1290 RESTORE
1300 FOR N=1 TO 6:READ P(N,1):READ P(
N,2):NEXT N
1310 S$="":FOR N=0 TO 7:READ A$:S$=S$

```

```

+CHR$(VAL("&H"+A$)):NEXT:SPRITE$(1)=S
$
1320 S$="":FOR N=0 TO 7:READ A$:S$=S$
+CHR$(VAL("&H"+A$)):NEXT:SPRITE$(2)=S
$
1330 DIM D(6,6)
1340 RETURN
1350 REM <<<<<< START GAME >>>>>>
1360 REM
1370 FOR N=1 TO 6:FOR M=1 TO 6
1380 D(N,M)=0
1390 NEXT M:NEXT N
1400 RETURN
1410 REM <<<<<<< INPUT >>>>>>>>
1420 REM
1430 LE=0:RI=0:FI=0
1440 A=STICK(0) OR STICK(1) OR STICK(
2):FI=STRIG(0) OR STRIG(1) OR STRIG(1
):A$=INKEY$
1450 IF A=3 THEN RI=1
1460 IF A=7 THEN LE=1
1470 FI=ABS(FI)
1480 FOR W=1 TO 70:NEXT W
1490 RETURN
1500 REM <<<<<<<< MOVE >>>>>>>>>>>>
1510 REM
1520 PNT=1:IF PLY=1 THEN CO=9 ELSE CO
=7
1530 COLOR CO
1540 PUTSPRITE 7,(P(PNT,1),P(PNT,2)),
CO,2
1550 GOSUB 1430
1560 IF FI=1 THEN GOTO 1620
1570 IF LE=1 THEN PNT=PNT-1
1580 IF RI=1 THEN PNT=PNT+1
1590 IF PNT=7 THEN PNT=1
1600 IF PNT=0 THEN PNT=6
1610 GOTO 1530
1620 PLAY "L20C"
1630 FOR W=1 TO 300:NEXT W
1640 RETURN
1650 PUT SPRITE 7,(P(DES,1),P(DES,2))
,CO,2
1660 RETURN
1670 REM <<<<<<<< TURN >>>>>>>>>>>>

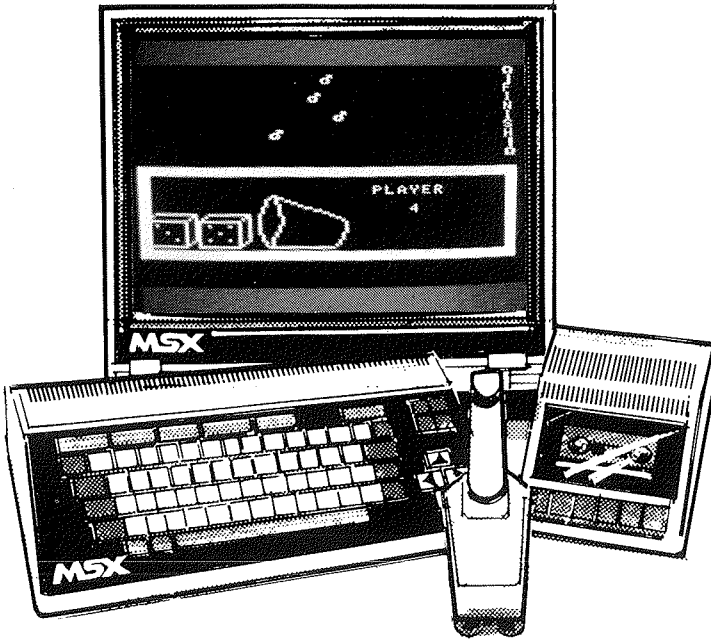
```

```

1680 REM
1690 GOSUB 1520
1700 FRO=PNT:PUT SPRITE 8,(P(FRO,1),P
(FRO,2)),13,2
1710 GOSUB 1520
1720 DES=PNT:PUT SPRITE 8,(0,0),13,2
1730 GOSUB 1650
1740 IF FRO=DES OR D(FRO,DES)<>0 THEN
GOTO 1810
1750 LINE (P(FRO,1)+4,P(FRO,2)+4)-(P(
DES,1)+4,P(DES,2)+4),CO
1760 D(DES,FRO)=PLY:D(FRO,DES)=PLY
1770 FOR N=1 TO 6
1780 IF D(DES,N)=PLY AND D(N,FRO)=PLY
THEN DD=1:GOSUB 1920
1790 NEXT N
1800 RETURN
1810 PLAY "CDE":GOTO 1690
1820 REM <<<<<<<< BOTH >>>>>>>>>>>>
1830 REM
1840 DD=0
1850 PLY=1:GOSUB 1690
1860 IF DD=1 THEN RETURN
1870 PLY=2:GOSUB 1690
1880 IF DD=1 THEN RETURN
1890 GOTO 1840
1900 REM <<<<<<< HIGHLIGHT >>>>>>>>
1910 REM
1920 LINE (P(FRO,1)+4,P(FRO,2)+4)-(P(
N,1)+4,P(N,2)+4),15
1930 LINE (P(N,1)+4,P(N,2)+4)-(P(DES,
1)+4,P(DES,2)+4),15
1940 LINE (P(DES,1)+4,P(DES,2)+4)-(P(
FRO,1)+4,P(FRO,2)+4),15
1950 PLAY "T140L4CL8EFGFEL4CL8CEFGFE"
1960 FOR W=1 TO 200:NEXT W
1970 RETURN
1980 DATA 127,148,190,117,190,55
1990 DATA 127,24,64,55,64,117
2000 DATA 00,7E,42,5A,5A,42,7E,00
2010 DATA FF,81,BD,A5,A5,BD,81,FF

```

10 Dice Derby



Shake!

This is the first of a dice trio, to be followed by *Snake Eyes* and *Craps*. All three use the same central routines and a menu is provided to select between the games. This will produce an enjoyable set of dice games and save a lot of typing.

Dice Derby is a race game in which the dice total is used to determine how far each horse travels. It can be played by two to four players and has an attractive shaking cup feature.

The fire button is pressed to start the roll and again to finish. This means that each player can exercise control over the duration of each shake.

The characters will race along until one is declared the winner. The winner is not determined until the last throw of the round and so some very close 'photo finishes' are possible.

It is possible to end the game and return to the menu by pressing 'grph' 'Q' for quit. Another game can then be selected.

Notes on typing in: the three dice games that follow use the same routine called *Dicerama!* This should be typed first and then

saved onto tape. The games must then be added one by one. It will not be possible to select a game from the menu until all have been entered.

```

1000 REM >>> DICERAMA   ANDY/JIM   <<<
1010 REM
1020 ON INTERVAL=10 GOSUB 1160:GOSUB
1470
1030 SCREEN 1,3:WIDTH 30:KEY OFF
1040 GOSUB 1190:COLOR 15,0,13
1050 GOSUB 3220
1060 LOCATE 4,3:PRINT"1> Dice Derby"
1070 LOCATE 4,6:PRINT"2> Snake Eyes"
1080 LOCATE 4,9:PRINT"3> Craps."
1090 INTERVAL ON
1100 A=VAL(INKEY$):IF A<1 OR A>3 THEN
  1100
1110 PLAY"o4a64"
1120 ON A GOTO 1710,2250,2670
1130 COLOR 14:LOCATE 4,10:PRINT"Press
  any key"
1140 A$=INKEY$:IF A$="" THEN 1140
1150 GOTO 1040
1160 INTERVAL OFF:A$=INKEY$
1170 IF A$="q" OR A$="Q" THEN RUN ELSE
  INTERVAL ON:RETURN
1180 REM >>>          SCREEN          <<<
1190 COLOR 14:FOR X=0 TO 31:LOCATE X,
11:PRINT"[":LOCATE X,20:PRINT"[":NEXT
  X
1200 FOR Y=12 TO 19:LOCATE 0,Y:PRINT"
  [":LOCATE 31,Y:PRINT"[":NEXT Y
1210 RETURN
1220 REM >>>>          RATTLE          <<<<
1230 SOUND 0,0:SOUND 1,3:SOUND 8,10
1240 PUT SPRITE 1,(25,90),15,11
1250 PUT SPRITE 2,(57,90),15,12
1260 PUT SPRITE 3,(25,122),15,13
1270 PUT SPRITE 4,(57,122),15,14
1280 R%=RND(-TIME)*6
1290 D1=INT(RND(R%)*6)+1
1300 SOUND 8,0
1310 A$=INKEY$:IF A$=CHR$(32) THEN 13
  90
1320 PUT SPRITE 1,(35,90),15,11

```

```

1330 PUT SPRITE 2, (67,90), 15, 12
1340 PUT SPRITE 3, (35, 122), 15, 13
1350 PUT SPRITE 4, (67, 122), 15, 14
1360 R%=RND(-TIME)*6
1370 D2=INT(RND(R%)*6)+1
1380 A#=INKEY#: IF A#<>CHR$(32) THEN 1
230
1390 PUT SPRITE 1, (76, 95), 15, 7
1400 PUT SPRITE 2, (108, 95), 15, 8
1410 PUT SPRITE 3, (76, 127), 15, 9
1420 PUT SPRITE 4, (108, 127), 15, 10
1430 PUT SPRITE 5, (12, 127), 15, D1
1440 PUT SPRITE 6, (44, 127), 15, D2
1450 RETURN
1460 REM >>> INITIALISE <<<
1470 DIM S$(14), P(4, 3), O(5)
1480 RETURN
1490 REM >>> CLEAR <<<
1500 FOR Y=0 TO 10: LOCATE 0, Y
1510 PRINT STRING$(30, " "): NEXT Y
1520 PLAY "o3a64": RETURN
1530 REM >>> PLAYERS <<<
1540 GOSUB 1500
1550 LOCATE 0, 0: PRINT "PLAYERS ? (2-4
)"
1560 A#=INKEY#: IF A#="" THEN 1560
1570 IF VAL(A#)<2 OR VAL(A#)>4 THEN 1
560
1580 PLAY "o5g64": PL=VAL(A#)
1590 GOSUB 1500
1600 FOR N=1 TO 4
1610 P(N, 1)=0
1620 NEXT N
1630 FOR N=1 TO PL
1640 O(N)=1
1650 NEXT N
1660 FOR Q=N TO 4
1670 O(Q)=0
1680 NEXT Q
1690 RETURN
1700 REM >>> DICE DERBY <<<
1710 COLOR ,, 12: GOSUB 1540
1720 FOR N=1 TO 4
1730 P(N, 1)=0: P(N, 2)=2*N-1
1740 NEXT N

```

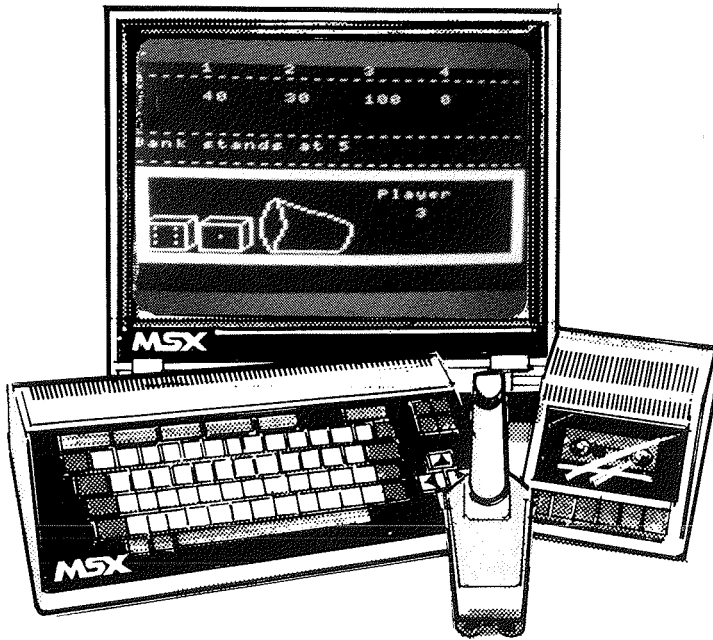
```

1750 RESTORE 1810
1760 FOR N=0 TO 9
1770 READ A
1780 LOCATE 31,N
1790 PRINT CHR$(A)
1800 NEXT N
1810 DATA 200,202,70,73,78,73,83,72,2
02,200
1820 GOSUB 2200
1830 FOR W=1 TO PL
1840 LOCATE 18,13:PRINT "PLAYER";
1850 LOCATE 20,15:PRINT W
1860 PLAY "o3f64"
1870 A$=INKEY$:IF A$="" THEN 1870
1880 GOSUB 2080
1890 GOSUB 1230
1900 FOR D=1 TO (D1+D2)*2
1910 GOSUB 2160
1920 P(W,1)=P(W,1)+1
1930 GOSUB 2180
1940 NEXT D
1950 NEXT W
1960 WI=0
1970 FOR N=1 TO 4
1980 IF WI>0 THEN 2000
1990 IF P(N,1)>240 THEN WI=N
2000 NEXT N
2010 IF WI=0 THEN 1830
2020 GOSUB 1500
2030 GOSUB 2080
2040 LOCATE 0,0:PRINT " Player ";WI;"
won !"
2050 FOR T=1 TO 2000:NEXT T
2060 GOSUB 1500:GOTO 1060
2070 END
2080 PUT SPRITE 5,(100,100),1,0
2090 PUT SPRITE 6,(100,100),1,0
2100 RETURN
2110 END
2120 FOR N=1 TO 4
2130 LOCATE INT(P(N,1)/8),P(N,2)
2140 PRINT " ";NEXT N
2150 RETURN
2160 LOCATE INT(P(W,1)/8),P(W,2)
2170 PRINT " ":RETURN

```

```
2180 LOCATE INT(P(W,1)/8),P(W,2)
2190 PRINT CHR$(201):RETURN
2200 FOR N=1 TO 4
2210 LOCATE INT(P(N,1)/8),P(N,2)
2220 PRINT CHR$(201):NEXT N
2230 RETURN
```

II Snake Eyes



Rattle!

This is the second of the three dice games. *Dice Derby* must be entered first and then *Snake Eyes* can be added. The instructions for throwing the dice are the same.

The game is for two to four players and includes on-screen betting. At the start the target pot is entered, which may be between £40 and £400. The bank starts at £0.

Each player then rolls the dice. If a double is thrown then the player wins the bank amount. If 'Snake Eyes' (two ones) is thrown then the player gets twice the bank total. The bank resets to zero when it pays out.

After each round the bank gets £5 if the target set is below £200 and £10 if the target is above £200.

The game continues until a player's cash reaches the target score.

It may be interesting to change the amount to the bank after each round to provide a larger 'pot'!

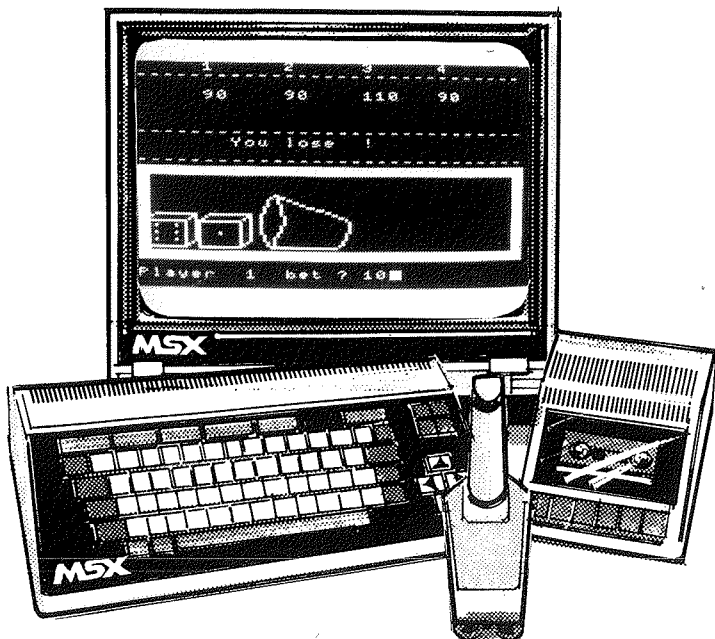
```

2240 REM >>>          SNAKE EYES          <<<
2250 COLOR,,6:GOSUB 1540
2260 LOCATE 0,22:PRINT "Enter win sco
re ";
2270 INPUT LI
2280 LOCATE 0,22:PRINT STRING$(28," ")
)
2290 IF LI>400 OR LI<40 THEN 2250
2300 UA=5:GOSUB 1500:IF LI>200 THEN U
A=10
2310 LOCATE 5,0:PRINT "1"
2320 LOCATE 11,0:PRINT "2"
2330 LOCATE 17,0:PRINT "3"
2340 LOCATE 23,0:PRINT "4"
2350 LOCATE 0,1:PRINT STRING$(30,"-")
;
2360 LOCATE 0,10:PRINT STRING$(30,"-")
);
2370 LOCATE 0,7:PRINT STRING$(30,"-")
;
2380 KI=0
2390 KI=KI+UA
2400 PE=1
2410 FOR N=1 TO 4:LOCATE N*6-2,3
2420 PRINT P(N,1):NEXT N
2430 LOCATE 18,13:PRINT"Player"
2440 LOCATE 20,15:PRINT PE
2450 GOSUB 2480:IF P(PE,1)>LI THEN 26
40
2460 PE=PE+1:IF PE>PL THEN 2390
2470 GOTO 2410
2480 A$=INKEY$
2490 IF A$="" THEN 2480
2500 LOCATE 0,8:PRINT "Bank stands at
";KI
2510 GOSUB 2080
2520 GOSUB 1230
2530 IF D1=1 AND D2=1 THEN GOSUB 2560
2540 IF D1=D2 AND D1<>1 THEN GOSUB 26
00
2550 RETURN
2560 PLAY"o4a64b64c64"
2570 P(PE,1)=P(PE,1)+KI*2
2580 KI=0
2590 RETURN

```

```
2600 PLAY"o4g64e64d64"  
2610 P(PE,1)=P(PE,1)+KI  
2620 KI=0  
2630 RETURN  
2640 WI=PE  
2650 GOTO 2020
```

12 Craps



... and Roll!

Here it is, the micro version of the American dice game featured in gangster movies and TV shows.

Two to four players can now pretend that they are in Las Vegas courtesy of MSX.

At the start of the game each player has £100 (you can change it to dollars if you want more realism!). A bet is placed up to the total holding of each player on his turn. Then the fun begins.

If a 7 or 11 is thrown first then they are paid evens. If 2, 3 or 12 is thrown the turn ends and the stake is lost. Any other total becomes the *point*. This is then the total to roll for.

The player continues to roll until the point is reached, paying evens, or a 7 is thrown in which case it is the next player's turn.

If a player loses all his money then he drops out and the game continues until only one player is left.

Pressing 'grph' 'Q' will quit it any time.

This program requires that the *Dice Derby* and *Snake Eyes* listings are entered first.


```

2660 REM >>>    CRAPS    <<<
2670 COLOR,,4:GOSUB 1540
2680 LOCATE 5,0:PRINT "1"
2690 LOCATE 11,0:PRINT "2"
2700 LOCATE 17,0:PRINT "3"
2710 LOCATE 23,0:PRINT "4"
2720 LOCATE 0,1:PRINT STRING$(30,"-")
;
2730 LOCATE 0,10:PRINT STRING$(30,"-")
);
2740 LOCATE 0,7:PRINT STRING$(30,"-")
;
2750 FOR N=1 TO 4
2760 P(N,1)=100
2770 NEXT N
2780 FOR W=1 TO PL
2790 FOR N=1 TO 4
2800 LOCATE N*6-2,3:PRINT P(N,1)
2810 NEXT N
2820 IF P(W,1)<1 THEN 2990
2830 LOCATE 0,22:PRINT "Player ";W;"
bet ";
2840 INPUT B$
2850 LOCATE 0,22:PRINT STRING$(20," ")
)
2860 BE=VAL(B$):IF BE<1 OR BE>P(W,1)T
HEN PLAY"o4a64":GOTO 2830
2870 GOSUB 2080:GOSUB 1230
2880 T=D1+D2
2890 IF T=7 OR T=11 THEN 3130
2900 IF T=2 OR T=3 OR T=12 THEN 3170
2910 PO=T
2920 LOCATE 0,8:PRINT "Roll again . .
. .
"
2930 A$=INKEY$:IF A$="" THEN 2930
2940 GOSUB 2080:GOSUB 1230
2950 T=D1+D2
2960 IF T=PO THEN 3130
2970 IF T=7 THEN 3170
2980 GOTO 2920
2990 NEXT W
3000 FL=0:CO=0
3010 FOR N=1 TO PL
3020 IF CO=1 THEN 3050
3030 IF P(N,1)>0 AND FL=1 THEN CO=1

```

```

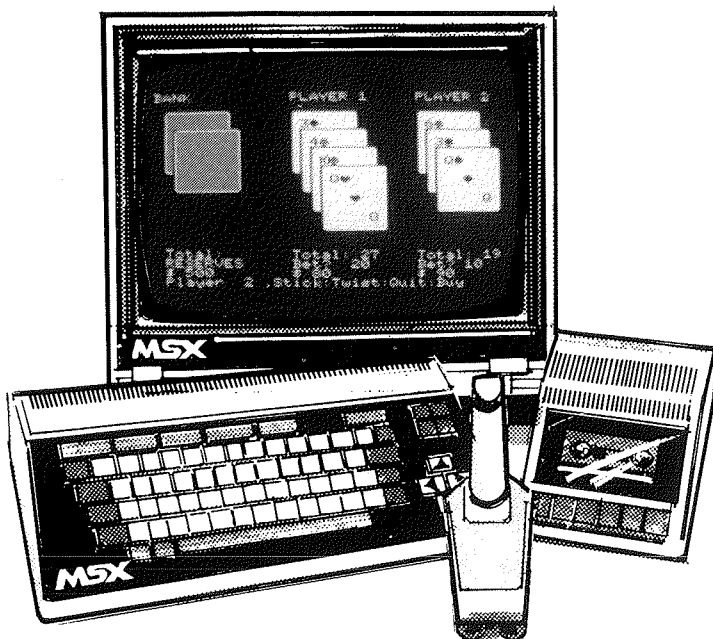
3040 IF P(N,1)>0 THEN FL=1
3050 NEXT N
3060 IF CO=1 THEN 2780
3070 WI=0
3080 FOR N=1 TO PL
3090 IF WI<>0 THEN 3110
3100 IF P(N,1)>0 THEN WI=N
3110 NEXT N
3120 GOTO 2020
3130 PLAY "o4g64f64e64"
3140 LOCATE 0,8:PRINT "          You win
!          "
3150 P(W,1)=P(W,1)+BE
3160 GOTO 2990
3170 PLAY "o4e64g64"
3180 LOCATE 0,8:PRINT "          You los
e !          "
3190 P(W,1)=P(W,1)-BE
3200 GOTO 2990
3210 REM > DEFINE SPRITES/CHARS <
3220 RESTORE 3330
3230 FOR S=1 TO 14:S$(S)=" "
3240 FOR DD=1 TO 32
3250 READ DA:S$(S)=S$(S)+CHR$(DA)
3260 NEXT DD:SPRITE$(S)=S$(S)
3270 NEXT S
3280 FOR CH=200 TO 202
3290 FOR DD=0 TO 7
3300 READ DA:VPOKE CH*8+DD,DA
3310 NEXT DD:NEXT CH
3320 RETURN
3330 DATA 31,32,64,255,128,128,128,12
8,130,128,128,128,128,128,255,0
3340 DATA 255,3,5,249,9,9,9,9,9,9,9
,10,12,248,0
3350 DATA 31,32,64,255,128,144,128,12
8,128,128,128,128,128,128,255,0
3360 DATA 255,3,5,249,9,9,9,9,9,9,9,7
3,10,12,248,0
3370 DATA 31,32,64,255,128,144,128,12
8,130,128,128,128,128,128,255,0
3380 DATA 255,3,5,249,9,9,9,9,9,9,9,7
3,10,12,248,0
3390 DATA 31,32,64,255,128,144,128,12
8,128,128,128,144,128,128,255,0

```

3400 DATA 255, 3, 5, 249, 9, 73, 9, 9, 9, 9, 9,
 73, 10, 12, 248, 0
 3410 DATA 31, 32, 64, 255, 128, 144, 128, 12
 8, 130, 128, 128, 144, 128, 128, 255, 0
 3420 DATA 255, 3, 5, 249, 9, 73, 9, 9, 9, 9, 9,
 73, 10, 12, 248, 0
 3430 DATA 31, 32, 64, 255, 128, 144, 128, 12
 8, 144, 128, 128, 144, 128, 128, 255, 0
 3440 DATA 255, 3, 5, 249, 9, 73, 9, 9, 73, 9, 9,
 73, 10, 12, 248, 0
 3450 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 5, 9, 9, 8, 1
 6, 16, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 192, 48, 12, 1
 31, 128, 128, 128
 3460 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2
 24, 28, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0
 3470 DATA 32, 32, 32, 32, 32, 32, 32, 32, 32, 33,
 17, 18, 18, 10, 10, 7, 0, 128, 64, 64, 64, 64, 12
 8, 128, 128, 0, 0, 0, 0, 0, 0, 0, 255, 0
 3480 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 255, 0, 192, 48, 8, 4, 4, 4, 4, 4, 8, 8, 8, 16, 16
 , 32, 192, 0
 3490 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 1, 0, 0, 0, 0, 15, 16, 32, 32, 64, 64, 128, 12
 8, 128, 0, 0
 3500 DATA 0, 0, 0, 0, 240, 8, 4, 4, 2, 2, 2, 2, 1
 , 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 128
 3510 DATA 1, 2, 2, 2, 2, 2, 4, 4, 4, 4, 4, 2, 1, 0
 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 128, 112,
 15, 0
 3520 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1
 4, 240, 0, 128, 128, 64, 64, 64, 64, 64, 32, 32,
 32, 32, 64, 128, 0, 0, 0
 3530 DATA 0, 124, 198, 198, 198, 108, 238, 0
 3540 DATA 15, 7, 13, 120, 204, 204, 204, 120
 3550 DATA 24, 24, 24, 24, 24, 24, 24, 24

13

Pontoon



A Pair can Break the Bank

In this computer version of the well known card game the bank starts with £500 and each player has £100. One or two players each play against the bank and aim to get a total of 21, which is Pontoon.

One card is first dealt face up to each player and one face down to the bank. A bet up to the total cash available may then be placed. The next card is dealt and various options are given to each player on their turn.

'S'tick. If the total is 16 or over then pressing 'S' will pass play on.

'T'wist. If another card is required this option may be taken by pressing 'T'.

'B'uy. Used when a hand looks good, to double the stake.

An ace will be taken to have the best value, i.e. 1 or 11 dependent upon the other cards. If at any time the card total exceeds 21 then play passes on. When both players have finished, the bank will then play its own hand to determine payout. It should be remembered that play is between the players and the bank, not against each other.

Wins and losses are calculated as follows: highest score is Pontoon with two cards, e.g. 10 and a Queen. The next highest score is a 5-card *trick*, which may be any five cards which do not exceed 21 in total. Pontoon with any card permutation is the third high score. If none of the above exists then the highest score wins. Where the bank's total equals that of a player then the bank wins.

The game continues until each player has lost all available cash or until the bank is bust. 'Q' for Quit may be used to end a game prematurely.

```

1000 REM >>>>    PONTOON    <<<<<
1010 REM >>>>    ANDY/JIM    <<<<<
1020 SCREEN 0:COLOR 15,12:KEY OFF
1030 DEFINT A-Z
1040 DIM PA(52),AV(52),PAC$(5,13),CC(
3)
1050 DIM CX(4),CY(4)
1060 DIM NOC(4),P(2)
1070 DIM PLC(3,5),BC(5)
1080 DIM MO(2),P$(2)
1090 GOSUB 3960
1100 GOSUB 3120
1110 MO(1)=100:MO(2)=100:MC=500
1120 CC(1)=20:CC(2)=33:CC(3)=6
1130 SC(1)=0:SC(2)=0
1140 GOSUB 2570
1150 GOSUB 2690
1160 CLS:IF NP=1 THEN P(2)=0
1170 IF P(1)=0 AND P(2)=0 THEN 4650
1180 IF P(1)=0 THEN BE(1)=0
1190 IF P(2)=0 THEN BE(2)=0
1200 GOSUB 3200
1210 CX(1)=15:CX(2)=28
1220 CY(1)=2:CY(2)=2
1230 BX=2:BY=2
1240 GOSUB 3400
1250 FOR Q=1 TO NP
1260 IF P(Q)=0 THEN 1290
1270 GOSUB 3450
1280 GOSUB 4390
1290 NEXT Q
1300 CX=BX:CY=BY:FOR Q=1 TO NC

```

```

1310 GOSUB 3030:CY=CY+2:CX=CX+1:NEXT
Q
1320 GOSUB 3750
1330 FOR Q=1 TO NP:IF P(Q)=0 THEN 139
0
1340 CY=CY(Q):CX=CX(Q)
1350 CY=CY+2:CX=CX+1
1360 GOSUB 3890
1370 GOSUB 4390
1380 CX(Q)=CX
1390 NEXT Q
1400 CX=BX+1:CY=BY+2:GOSUB 2990
1410 GOSUB 3030
1420 GOSUB 3530
1430 REM >>>>      SCORE LOGIC <<<<<
1440 Q=3
1450 IF NP=2 THEN 1460 ELSE IF SC(1)>
21 THEN P$(1)="L":GOTO 1620
1460 IF SC(1)>21 AND SC(2)>21 THEN P$
(1)="L":P$(2)="L":GOTO 1620
1470 FOR W=1 TO NC
1480 CX=BX:CY=(BY-2)+W*2
1490 PLC(3,W)=BC(W)
1500 GOSUB 3330:GOSUB 2810
1510 BX=BX+1
1520 NEXT W
1530 BX=CX:BY=CY
1540 FOR W=1 TO NC
1550 PLC(3,W)=BC(W):NEXT W
1560 Q=3:NOC(3)=NC:CX(3)=BX:GOSUB 439
0
1570 IF SC(3)<16 THEN GOSUB 2990 ELSE
GOTO 1620
1580 BX=BX+1:BY=BY+2:PLC(3,NC)=BC(NC)
:Q=3:CX=BX:CY=BY:W=NC:GOSUB 3330:GOSU
B 2810
1590 NOC(3)=NC:GOSUB 4390
1600 IF NC=5 THEN 1620 ELSE GOTO 1570
1610 F=(RND(1)*2)+17:IF SC(3)>F THEN
GOSUB 2990:GOTO 1580
1620 Q=1
1630 IF P(Q)=0 THEN 1780
1640 IF SC(Q)>21 THEN P$(Q)="L":GOTO
1780
1650 IF SC(3)>21 THEN P$(Q)="W":GOTO
1780

```

```

1660 IF SC(3)=21 AND NC=2 THEN P$(Q)=
"L":GOTO 1780
1670 IF SC(Q)=21 AND NOC(Q)=2 THEN P$
(Q)="W":GOTO 1780
1680 IF NC=5 AND NOC(Q)=5 THEN P$(Q)=
"L":GOTO 1780
1690 IF NOC(Q)=5 THEN P$(Q)="W":GOTO
1780
1700 IF NC=5 THEN P$(Q)="L":GOTO 1780
1710 IF SC(3)=21 AND NC=2 AND SC(Q)=2
1 AND NOC(Q)=2 THEN P$(Q)="L":GOTO 17
80
1720 IF NOC(Q)=5 AND NC<5 THEN P$(Q)=
"W":GOTO 1780
1730 IF SC(Q)=21 AND NOC(Q)=2 THEN P$
(Q)="":GOTO 1780
1740 IF SC(Q)>SC(3) THEN P$(Q)="W":GO
TO 1780
1750 IF SC(Q)=SC(3) THEN P$(Q)="L":GO
TO 1780
1760 IF SC(3)>SC(Q) THEN P$(Q)="L"
1770 IF SC(Q)>SC(3) THEN P$(Q)="W"
1780 IF NP=2 AND Q=1 THEN Q=2:GOTO 16
30
1790 IF NP=1 THEN 1840
1800 IF P$(1)="L" AND P$(2)="L" THEN
2130
1810 IF P$(1)="W" AND P$(2)="W" THEN
2040
1820 IF P$(1)="W" AND P$(2)="L" THEN
2240
1830 IF P$(1)="L" AND P$(2)="W" THEN
2350
1840 IF P$(1)="W" THEN 1880
1850 IF P$(1)="L" THEN 1970
1860 END
1870 REM >>>>      PLAYER WIN      <<<<<
1880 MC=MC-BE(1)
1890 MO(1)=MO(1)+BE(1)*2
1900 GOSUB 4670
1910 IF MC<1 THEN 2450
1920 GOSUB 4510
1930 PLAY "O1L64ERC"
1940 LOCATE 1,20:PRINT "Player wins":
GOSUB 4520

```

```

1950 GOTO 1150
1960 REM >>>>      BANK WIN   <<<<<<<
1970 MC=MC+BE(1)
1980 GOSUB 4510:GOSUB 4680
1990 PLAY "04L64CEG"
2000 LOCATE 1,20:PRINT "Bank wins":GO
SUB 4520
2010 IF MO(1)<1 THEN P(1)=0
2020 GOTO 1150
2030 REM >>>>      BOTH WIN   <<<<
2040 MO(1)=MO(1)+BE(1)*2
2050 MO(2)=MO(2)+BE(2)*2
2060 GOSUB 4670
2070 MC=MC-BE(1)-BE(2)
2080 IF MC<1 THEN 2450
2090 GOSUB 4510
2100 LOCATE 1,20:PRINT "Both players
win"
2110 GOTO 1150
2120 REM >>>>      BOTH LOST  <<<<<
2130 MC=MC+BE(1)+BE(2)
2140 GOSUB 4510:GOSUB 4680
2150 IF MO(1)<1 THEN P(1)=0
2160 IF MO(2)<1 THEN P(2)=0
2170 IF P(1)=0 AND P(2)=1 THEN LOCATE
1,20:PRINT "Player 2 loses":PLAY "01
L8C"
2180 IF P(1)=1 AND P(2)=0 THEN LOCATE
1,20:PRINT "Player 1 loses":PLAY "01
L8C"
2190 IF MO(1)<1 THEN P(1)=0
2200 IF P(1)=1 AND P(2)=1 THEN LOCATE
1,20:PRINT "Both players lose":PLAY
"01L8EDC"
2210 GOSUB 4520
2220 GOTO 1150
2230 REM >>>>      PLAYER 1 WIN <<<<<
2240 MC=MC+BE(2)
2250 MO(1)=MO(1)+BE(1)*2
2260 MC=MC-BE(1)
2270 GOSUB 4670
2280 IF MO(2)<1 THEN P(2)=0
2290 IF MC<1 THEN 2450
2300 GOSUB 4510
2310 PLAY "04L64CDEFGABAGFEDC"

```



```

2320 LOCATE 1,20:PRINT "Player 1 wins
":GOSUB 4520
2330 GOTO 1150
2340 REM >>>>    PLAYER 2 WIN <<<<<
2350 MC=MC+BE(1)
2360 MO(2)=MO(2)+BE(2)*2
2370 MC=MC-BE(2)
2380 GOSUB 4670
2390 IF MC<1 THEN 2450
2400 GOSUB 4510
2410 PLAY "04L64CDEFGABAGFEDC"
2420 LOCATE 1,20:PRINT "Player 2 wins
":GOSUB 4520
2430 GOTO 1150
2440 REM >>>>    BANK BROKE <<<<<
2450 GOSUB 4510
2460 LOCATE 1,20
2470 PLAY "04L64CDEFGABAGFEDCCDEFGAB"
2480 PRINT "Well done,you broke the b
ank"
2490 GOSUB 4670:GOSUB 4670
2500 GOSUB 4520
2510 CLS
2520 PRINT "Player 1:";MO(1)
2530 IF NP=2 THEN PRINT "Player 2:";M
O(2)
2540 GOSUB 4520:GOSUB 4520
2550 GOTO 1100
2560 REM >>>>    GENERATE PACK <<<
2570 FOR SUIT=1 TO 4
2580 FOR CAR=1 TO 13
2590 PAC$(SUIT,CAR)=CHR$(SUIT+217)
2600 IF CAR<11 AND CAR>1 THEN PAC$(SU
IT,CAR)=PAC$(SUIT,CAR)+CHR$(CAR+220)
2610 IF CAR=1 THEN PAC$(SUIT,CAR)=PAC
$(SUIT,CAR)+CHR$(231)
2620 IF CAR=11 THEN PAC$(SUIT,CAR)=PA
C$(SUIT,CAR)+CHR$(232)
2630 IF CAR=12 THEN PAC$(SUIT,CAR)=PA
C$(SUIT,CAR)+CHR$(233)
2640 IF CAR=13 THEN PAC$(SUIT,CAR)=PA
C$(SUIT,CAR)+CHR$(234)
2650 NEXT CAR
2660 NEXT SUIT
2670 RETURN

```

```

2680 REM >>>> SHUFFLE PACK <<<<<
2690 FOR N=1 TO 52
2700 PA(N)=N:AV(N)=1
2710 NEXT N
2720 FOR A=1 TO 30
2730 N1=INT(RND(1)*51)+1
2740 N2=INT(RND(1)*51)+1
2750 CC=PA(N1)
2760 PA(N1)=PA(N2)
2770 PA(N2)=CC
2780 NEXT A
2790 RETURN
2800 REM >>>> DRAW CARD <<<<<
2810 LOCATE CX-1,CY
2820 PRINT CHR$(200);:FOR Z=1 TO 5:PR
INT CHR$(204);:NEXT Z:PRINT CHR$(203)
2830 LOCATE CX-1,CY+1
2840 PRINT CHR$(206);RIGHT$(PAC$(SUIT
,CAR),1);
2850 PRINT LEFT$(PAC$(SUIT,CAR),1);:P
RINT STRING$(3,CHR$(208));CHR$(207)
2860 LOCATE CX-1,CY+2
2870 PRINT CHR$(206);STRING$(5,CHR$(2
08));CHR$(207)
2880 LOCATE CX-1,CY+3
2890 PRINT CHR$(206);CHR$(208);CHR$(2
08);
2900 PRINT LEFT$(PAC$(SUIT,CAR),1);CH
R$(208);CHR$(208);CHR$(207)
2910 LOCATE CX-1,CY+4
2920 PRINT CHR$(206);STRING$(5,CHR$(2
08));CHR$(207)
2930 LOCATE CX-1,CY+5
2940 PRINT CHR$(206);CHR$(208);CHR$(2
08);CHR$(208);CHR$(208);RIGHT$(PAC$(S
UIT,CAR),1);CHR$(207)
2950 LOCATE CX-1,CY+6
2960 PRINT CHR$(201);STRING$(5,CHR$(2
05));CHR$(202)
2970 RETURN
2980 REM >>>> COMPUTERS CARD <<<<<
2990 C=(RND(1)*51)+1:IF AV(C)=0 THEN
2990
3000 NC=NC+1:BC(NC)=C
3010 RETURN

```

```

3020 REM >>> DRAW CARD BACK <<<<
3030 LOCATE CX-1,CY
3040 PRINT CHR$(209);STRING$(5,CHR$(2
13));CHR$(210)
3050 FOR NN=1 TO 5:LOCATE CX-1,CY+NN
3060 PRINT CHR$(215);STRING$(5,CHR$(2
17));CHR$(216)
3070 NEXT NN
3080 LOCATE CX-1,CY+6:PRINT CHR$(211)
;STRING$(5,CHR$(214));CHR$(212)
3090 RETURN
3100 REM >>>> NUMBER OF PLAYERS <<<<
3110 RETURN
3120 CLS
3130 LOCATE 15,1:PRINT"PONTOON"
3140 PRINT:PRINT "1 or 2 players ";
3150 NP=VAL(INPUT$(1)):IF NP<1 OR NP>
2 THEN 3150
3160 BEEP
3170 FOR Q=1 TO NP:P(Q)=1:NEXT Q
3180 RETURN
3190 REM >>>> DEAL CARDS <<<<<
3200 FOR Q=1 TO NP
3210 C1=(RND(-TIME)*51)+1
3220 IF AV(C1)=0 THEN 3210
3230 AV(C1)=0
3240 PLC(Q,1)=C1
3250 NEXT Q
3260 C1=(RND(-TIME)*51)+1
3270 IF AV(C1)=0 THEN 3260
3280 AV(C1)=0
3290 BC(1)=C1
3300 NOC(1)=1:NOC(2)=1:NC=1
3310 RETURN
3320 REM >>>> DECODE CARD <<<<<
3330 S=PLC(Q,W)/13:CAR=PLC(Q,W) MOD 1
3:IF CAR=0 THEN CAR=13
3340 IF S>=0 THEN SUIT=1
3350 IF S>=1 THEN SUIT=2
3360 IF S>=2 THEN SUIT=3
3370 IF S>=3 THEN SUIT=4
3380 RETURN
3390 REM >>>> DRAW SCREEN <<<<
3400 LOCATE 0,0:PRINT"BANK","PLAYER 1
",:IF NP=2 THEN LOCATE 27,0:PRINT"PLA
YER 2"

```

```

3410 LOCATE 1,19:PRINT"#":LOCATE 14,1
9:PRINT"#":LOCATE 27,19:PRINT"#"
3420 LOCATE 1,18:PRINT"RESERVES":LOCA
TE 1,17:PRINT"Total:":LOCATE 14,17:PR
INT"Total:":IF NP=2 THEN LOCATE 27,17
:PRINT"Total:"
3430 RETURN
3440 REM >>>>      DRAW HAND      <<<<<
3450 CX=CX(Q):CY=CY(Q)
3460 FOR W=1 TO NOC(Q)
3470 GOSUB 3330
3480 GOSUB 2810
3490 CY=CY+2:CX=CX+1:NEXT W
3500 LOCATE CC(Q)-5,19:PRINT MO(Q)
3510 RETURN
3520 REM >>>>      STICK      <<<<<
3530 FOR Q=1 TO NP:IF P(Q)=0 THEN 366
0
3540 CX(Q)=CX(Q)+1
3550 LOCATE 1,20:PRINT STRING$(39," "
)
3560 LOCATE 1,20:PRINT "Player ";Q;",
";:PRINT"Stick:Twist:Quit:Buy"
3570 A$=INKEY$:IF A$="S" OR A$="T" OR
A$="Q" OR A$="B" THEN 3580 ELSE 3570
3580 CX=CX(Q):CY=CY(Q)+(2*NOC(Q))
3590 IF A$="B" THEN 4540
3600 IF A$="Q" THEN END
3610 IF A$="S" AND SC(Q)>15 THEN GOSU
B 3680:CX=CX+1:GOTO 3660
3620 IF A$="T" THEN GOSUB 3700:CX=CX+
1:CX(Q)=CX(Q)+1
3630 GOSUB 4390
3640 IF SC(Q)>21 THEN LOCATE CC(Q),20
:PRINT "BUST":GOTO 3660
3650 IF NOC(Q)<>5 THEN 3570
3660 NEXT Q
3670 LOCATE 1,20:PRINT STRING$(39," "
):RETURN
3680 GOSUB 4390:LOCATE CC(Q),17:PRINT
" ":LOCATE CC(Q),17:PRINT SC(Q):RE
TURN
3690 REM >>>>      TWIST      <<<<<
3700 C=(RND(1)*51)+1:IF AV(C)=0 THEN
3700

```

```

3710 AV(C)=0:NOC(Q)=NOC(Q)+1
3720 PLC(Q,NOC(Q))=C:W=NOC(Q):GOSUB 3
330:GOSUB 2810
3730 RETURN
3740 REM >>>> PLACE BETS <<<<<
3750 LOCATE 2,19:PRINT MC:FOR Q=1 TO
NP
3760 IF P(Q)=0 THEN 3860
3770 LOCATE 1,20:PRINTSTRING$(39," ")
3780 LOCATE 1,20:PRINT "Player ";Q;"
Your bet";
3790 LOCATE CC(Q)-6,18:PRINT "Bet";
3800 INPUT BE(Q)
3810 IF BE(Q)<1 THEN 3770
3820 IF BE(Q)>MO(Q) THEN 3770
3830 LOCATE CC(Q),19:PRINT " "
3840 MO(Q)=MO(Q)-BE(Q)
3850 LOCATE CC(Q)-4,19:PRINT " ":LO
CATE CC(Q)-5,19:PRINT MO(Q)
3860 NEXT Q
3870 RETURN
3880 REM >>> DEAL ANOTHER CARD <<<
3890 C=(RND(1)*51)+1
3900 IF AV(C)=0 THEN 3890
3910 PLC(Q,NOC(Q)+1)=C:AV(C)=0
3920 NOC(Q)=NOC(Q)+1
3930 W=NOC(Q):GOSUB 3330:GOSUB 2810
3940 RETURN
3950 REM >>>> CHARACTERS <<<<<
3960 RESTORE 4030
3970 FOR C=200 TO 234
3980 FOR B=0 TO 7
3990 READ A:VPOKE 2048+C*8+B,A
4000 NEXT B
4010 NEXT C
4020 RETURN
4030 DATA 0,60,124,124,124,124,124,124,12
4
4040 DATA 124,124,124,124,124,124,124,60,
0
4050 DATA 248,248,248,248,248,248,248,240
,0
4060 DATA 0,240,248,248,248,248,248,248,2
48
4070 DATA 0,252,252,252,252,252,252,252,2
52

```

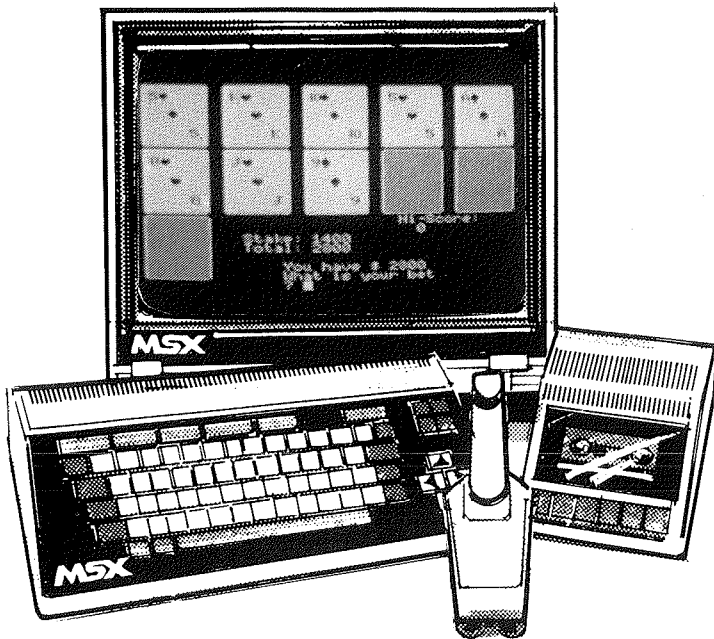
4080 DATA 252, 252, 252, 252, 252, 252, 252, 252, 0
 4090 DATA 124, 124, 124, 124, 124, 124, 124, 124, 124
 4100 DATA 248, 248, 248, 248, 248, 248, 248, 248, 248
 4110 DATA 252, 252, 252, 252, 252, 252, 252, 252, 252
 4120 DATA 0, 60, 84, 104, 84, 104, 84, 104
 4130 DATA 0, 240, 88, 168, 88, 168, 88, 168
 4140 DATA 84, 104, 84, 104, 84, 104, 60, 0
 4150 DATA 88, 168, 88, 168, 88, 168, 240, 0
 4160 DATA 0, 252, 84, 168, 84, 168, 84, 168
 4170 DATA 84, 168, 84, 168, 84, 168, 252, 0
 4180 DATA 85, 106, 85, 106, 85, 106, 85, 106
 4190 DATA 88, 168, 88, 168, 88, 168, 88, 168
 4200 DATA 84, 168, 84, 168, 84, 168, 84, 168
 4210 DATA 252, 140, 140, 84, 4, 84, 220, 140
 4220 DATA 220, 140, 4, 4, 4, 140, 220, 252
 4230 DATA 252, 180, 0, 0, 0, 132, 204, 252
 4240 DATA 220, 140, 4, 4, 4, 4, 220, 140
 4250 DATA 140, 116, 244, 204, 188, 124, 4, 252
 4260 DATA 4, 244, 236, 204, 244, 116, 140, 252
 4270 DATA 236, 204, 172, 108, 4, 236, 236, 252
 4280 DATA 4, 124, 12, 244, 244, 116, 140, 252
 4290 DATA 204, 188, 124, 12, 116, 116, 140, 252
 4300 DATA 4, 244, 236, 220, 188, 188, 188, 252
 4310 DATA 140, 116, 116, 140, 116, 116, 140, 252
 4320 DATA 140, 116, 116, 132, 244, 236, 156, 252
 4330 DATA 100, 88, 88, 88, 88, 88, 100, 252
 4340 DATA 220, 172, 116, 116, 4, 116, 116, 252
 4350 DATA 196, 236, 236, 236, 236, 108, 156, 252
 4360 DATA 140, 116, 116, 116, 84, 108, 148, 252
 4370 DATA 116, 108, 92, 60, 92, 108, 116, 252

```

4380 REM >>>> PLAYER'S SCORE <<<<<
4390 SC(Q)=0:AC=0:FOR Z=1 TO NOC(Q):W
=Z
4400 GOSUB 3330:IF RIGHT$(PAC$(SUIT,C
AR),1)=CHR$(232) OR RIGHT$(PAC$(SUIT,
CAR),1)=CHR$(233) OR RIGHT$(PAC$(SUIT
,CAR),1)=CHR$(234) THEN SC(Q)=SC(Q)+1
0:GOTO 4430
4410 IF RIGHT$(PAC$(SUIT,CAR),1)=CHR$(
231) THEN AC=AC+1:GOTO 4430
4420 PU=ASC(RIGHT$(PAC$(SUIT,CAR),1))
-220:SC(Q)=SC(Q)+PU
4430 NEXT Z:IF AC>0 THEN 4470
4440 LOCATE CC(Q),17:PRINT "    ":LOCAT
E CC(Q),17:PRINT SC(Q)
4450 RETURN
4460 REM >>>> ACE IN HAND <<<<<
4470 FOR R=1 TO AC
4480 IF SC(Q)+11>21 THEN SC(Q)=SC(Q)+
1 ELSE SC(Q)=SC(Q)+11
4490 NEXT R
4500 GOTO 4440
4510 LOCATE 1,20:PRINT STRING$(39," "
):RETURN
4520 FOR T=1 TO 1400:NEXT T:RETURN
4530 REM >>>> BUY <<<<<<<
4540 IF MO(Q)<BE(Q) THEN 3630
4550 BE(Q)=BE(Q)*2
4560 MO(Q)=MO(Q)-BE(Q)/2
4570 GOSUB 3700
4580 LOCATE CC(Q)-1,18:PRINT "    "
4590 LOCATE CC(Q)-1,18:PRINT BE(Q)
4600 LOCATE CC(Q)-5,19:PRINT "    "
4610 LOCATE CC(Q)-5,19:PRINT MO(Q)
4620 CX=CX+1:CX(Q)=CX(Q)+1
4630 GOTO 3600
4640 REM >>>> END OF GAME <<<
4650 GOSUB 4510:LOCATE 1,20:PRINT"All
players money gone"
4660 GOSUB 4520:GOTO 1100
4670 RETURN
4680 RETURN

```

14 Hi-Lo



Nothing for a Pair

Here is the computer version of the card game featured in a well known TV programme.

Each player starts with £150 and eleven cards are dealt out face down.

The first is turned over and a bet is requested. The bet can be anything up to the player's total holding. When the bet has been placed, 'higher' or 'lower' will be requested. If this is correct then the bet is paid evens. If the guess is wrong or the card is the same value then the stake for that card is lost. This then continues until all the cards have been turned and a new game can then be played.


```

1000 REM >>>> HI-LO ANDY / JIM <<
<<
1010 SCREEN 0:COLOR 15,12:KEY OFF
1020 DEFINT A-Z
1030 DIM PA(52),AV(52),PAC$(5,13),BC(
11)
1040 DIM CX(11),CY(11)
1050 HI=0
1060 SCREEN0:WIDTH 40
1070 GOSUB 2480
1080 CLS
1090 LOCATE 17,10:PRINT"HI - LO"
1100 GOSUB 1390
1110 GOSUB 1510
1120 GOSUB 1890
1130 LOCATE 11,16:PRINT"Stake:"
1140 LOCATE 11,17:PRINT"Total:"
1150 LOCATE 27,14:PRINT"Hi-Score:"
1160 TT=150
1170 BE=0
1180 GOSUB 2060
1190 GOSUB 2210
1200 C=1
1210 CX=CX(C):CY=CY(C)
1220 GOSUB 2060
1230 IF C=11 THEN 2910
1240 PLC(1,1)=BC(C):Q=1:W=1
1250 GOSUB 1990:GOSUB 1630
1260 GOSUB 2140
1270 GOSUB 2290
1280 GOSUB 2060
1290 CX=CX(C+1):CY=CY(C+1)
1300 PLC(1,1)=BC(C+1):CB=CAR
1310 Q=1:W=1:GOSUB 1990
1320 PLAY "O2L64CRERF"
1330 GOSUB 1630
1340 IF CAR>CB AND A$="H" THEN 2430:G
OTO 1210
1350 IF CAR<CB AND A$="L" THEN 2430:G
OTO 1210
1360 GOTO 2390
1370 GOTO 1210
1380 REM >>>> GENERATE PACK <<<<
<
1390 FOR SUIT=1 TO 4

```

```

1400 FOR CAR=1 TO 13
1410 PAC$(SUIT,CAR)=CHR$(SUIT+217)
1420 IF CAR<11 AND CAR>1 THEN PAC$(SUIT,CAR)=PAC$(SUIT,CAR)+CHR$(CAR+220)
1430 IF CAR=1 THEN PAC$(SUIT,CAR)=PAC$(SUIT,CAR)+CHR$(231)
1440 IF CAR=11 THEN PAC$(SUIT,CAR)=PAC$(SUIT,CAR)+CHR$(232)
1450 IF CAR=12 THEN PAC$(SUIT,CAR)=PAC$(SUIT,CAR)+CHR$(233)
1460 IF CAR=13 THEN PAC$(SUIT,CAR)=PAC$(SUIT,CAR)+CHR$(234)
1470 NEXT CAR
1480 NEXT SUIT
1490 RETURN
1500 REM >>>> SHUFFLE PACK <<<<<
1510 FOR N=1 TO 52
1520 PA(N)=N:AV(N)=1
1530 NEXT N
1540 FOR A=1 TO 30
1550 N1=INT(RND(-TIME)*51)+1
1560 N2=INT(RND(-TIME)*51)+1
1570 CC=PA(N1)
1580 PA(N1)=PA(N2)
1590 PA(N2)=CC
1600 NEXT A
1610 RETURN
1620 REM >>>> DRAW CARD <<<<<
1630 LOCATE CX-1,CY
1640 PRINT CHR$(200);:FOR Z=1 TO 5:PRINT CHR$(204);:NEXT Z:PRINT CHR$(203)
1650 LOCATE CX-1,CY+1
1660 PRINT CHR$(206);RIGHT$(PAC$(SUIT,CAR),1);
1670 PRINT LEFT$(PAC$(SUIT,CAR),1);:PRINT STRING$(3,CHR$(208));CHR$(207)
1680 LOCATE CX-1,CY+2
1690 PRINT CHR$(206);STRING$(5,CHR$(208));CHR$(207)
1700 LOCATE CX-1,CY+3
1710 PRINT CHR$(206);CHR$(208);CHR$(208);
1720 PRINT LEFT$(PAC$(SUIT,CAR),1);CHR$(208);CHR$(208);CHR$(207)
1730 LOCATE CX-1,CY+4

```

```

1740 PRINT CHR$(206);STRING$(5,CHR$(2
08));CHR$(207)
1750 LOCATE CX-1,CY+5
1760 PRINT CHR$(206);CHR$(208);CHR$(2
08);CHR$(208);CHR$(208);RIGHT$(PAC$(S
UIT,CAR),1);CHR$(207)
1770 LOCATE CX-1,CY+6
1780 PRINT CHR$(201);STRING$(5,CHR$(2
05));CHR$(202)
1790 RETURN
1800 REM >>> DRAW CARD BACK <<<<
1810 LOCATE CX-1,CY
1820 PRINT CHR$(209);STRING$(5,CHR$(2
13));CHR$(210)
1830 FOR NN=1 TO 5:LOCATE CX-1,CY+NN
1840 PRINT CHR$(215);STRING$(5,CHR$(2
17));CHR$(216)
1850 NEXT NN
1860 LOCATE CX-1,CY+6:PRINT CHR$(211)
;STRING$(5,CHR$(214));CHR$(212)
1870 RETURN
1880 REM >>>> DRAW CARDS <<<<<
1890 RESTORE 1950:FOR A=1 TO 11
1900 READ CX(A),CY(A)
1910 CX=CX(A):CY=CY(A)
1920 GOSUB 1810
1930 NEXT A
1940 RETURN
1950 DATA 2,0,10,0,18,0,26,0,34,0
1960 DATA 2,7,10,7,18,7,26,7,34,7
1970 DATA 2,14
1980 REM >>>> DECODE CARD '<<<<<
1990 S=PLC(Q,W)/13:CAR=PLC(Q,W) MOD 1
3:IF CAR=0 THEN CAR=13
2000 IF S>=0 THEN SUIT=1
2010 IF S>=1 THEN SUIT=2
2020 IF S>=2 THEN SUIT=3
2030 IF S>=3 THEN SUIT=4
2040 RETURN
2050 REM >>>> PRINT HI,TOTAL,BET <<<
2060 LOCATE 17,16:PRINT " "
2070 LOCATE 17,16:PRINT BE
2080 LOCATE 17,17:PRINT " "
2090 LOCATE 17,17:PRINT TT
2100 LOCATE 28,15:PRINT " "
```

```

2110 LOCATE 28,15:PRINT HI
2120 RETURN
2130 REM >>>> PLACE BETS <<<<<
2140 LOCATE 15,19:PRINT "You have $";
TT
2150 LOCATE 15,20:PRINT "What is your
bet "
2160 LOCATE 15,21:INPUT BE
2170 IF BE<1 OR BE>TT THEN LOCATE 15,
21:PRINT STRING$(10," "):GOTO 2160
2180 TT=TT-BE:GOSUB 2060
2190 RETURN
2200 REM >>>> GENERATE CARDS <<<<
2210 FOR A=1 TO 11
2220 C=INT(RND(-TIME)*51)+1
2230 IF AV(C)=0 THEN 2220
2240 BC(A)=PA(C)
2250 AV(C)=0
2260 NEXT A
2270 RETURN
2280 REM >>> HIGHER / LOWER <<<
2290 LOCATE 15,20:PRINT STRING$(25,"
")
2300 LOCATE 15,19:PRINT STRING$(20,"
")
2310 LOCATE 15,21:PRINT STRING$(20,"
")
2320 LOCATE 15,19:PRINT "Higher/Lower
"
2330 A$=INKEY$
2340 IF A$="h" THEN A$="H"
2350 IF A$="l" THEN A$="L"
2360 IF A$<>"H" AND A$<>"L" THEN 2330
2370 RETURN
2380 REM >>>> LOSE <<<<<
2390 IF TT<1 THEN 2920
2400 PLAY "02L64BAGFEDCO1BAGFEDC"
2410 C=C+1:GOTO 1210
2420 REM >>>> WIN <<<<<
2430 TT=TT+BE*2
2440 C=C+1
2450 PLAY "04L64CDEFGABO5CDEFGAB"
2460 GOTO 1210
2470 REM >>>> CHARACTERS <<<<<
2480 RESTORE 2550

```

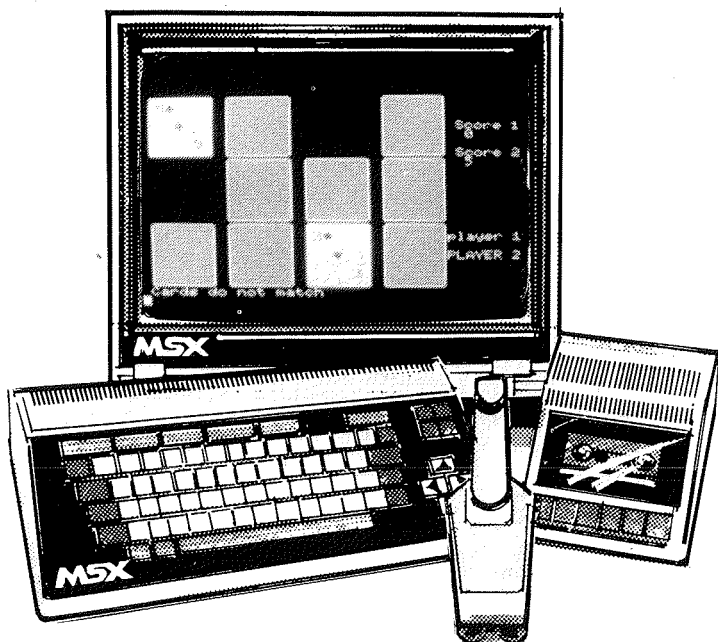
```

2490 FOR C=200 TO 234
2500 FOR B=0 TO 7
2510 READ A:VPOKE 2048+C*8+B,A
2520 NEXT B
2530 NEXT C
2540 RETURN
2550 DATA 0,60,124,124,124,124,124,124,124
2560 DATA 124,124,124,124,124,124,60,0
2570 DATA 248,248,248,248,248,248,240,0
2580 DATA 0,240,248,248,248,248,248,248,248
2590 DATA 0,252,252,252,252,252,252,252,252
2600 DATA 252,252,252,252,252,252,252,252,0
2610 DATA 124,124,124,124,124,124,124,124,124
2620 DATA 248,248,248,248,248,248,248,248,248
2630 DATA 252,252,252,252,252,252,252,252,252
2640 DATA 0,60,84,104,84,104,84,104
2650 DATA 0,240,88,168,88,168,88,168
2660 DATA 84,104,84,104,84,104,60,0
2670 DATA 88,168,88,168,88,168,240,0
2680 DATA 0,252,84,168,84,168,84,168
2690 DATA 84,168,84,168,84,168,252,0
2700 DATA 85,106,85,106,85,106,85,106
2710 DATA 88,168,88,168,88,168,88,168
2720 DATA 84,168,84,168,84,168,84,168
2730 DATA 252,140,140,84,4,84,220,140
2740 DATA 220,140,4,4,4,140,220,252
2750 DATA 252,180,0,0,0,132,204,252
2760 DATA 220,140,4,4,4,4,220,140
2770 DATA 140,116,244,204,188,124,4,252
2780 DATA 4,244,236,204,244,116,140,252
2790 DATA 236,204,172,108,4,236,236,252
2800 DATA 4,124,12,244,244,116,140,252

```

```
2810 DATA 204,188,124,12,116,116,140,
252
2820 DATA 4,244,236,220,188,188,188,2
52
2830 DATA 140,116,116,140,116,116,140
,252
2840 DATA 140,116,116,132,244,236,156
,252
2850 DATA 100,88,88,88,88,88,100,252
2860 DATA 220,172,116,116,4,116,116,2
52
2870 DATA 196,236,236,236,236,108,156
,252
2880 DATA 140,116,116,116,84,108,148,
252
2890 DATA 116,108,92,60,92,108,116,25
2
2900 FOR T=1 TO 5000:NEXT T:RETURN
2910 IF TT>HI THEN HI=TT:GOSUB 2900:G
OTO 1080
2920 LOCATE 15,20:PRINT STRING$(25,"
")
2930 LOCATE 15,20:PRINT "Run out of m
oney"
2940 GOSUB 2900
2950 GOTO 1080
```

15 Concentration



Find the Pair

This deceptively simple game requires a good memory and of course great concentration.

Each player has one turn to select two cards. If they match then they score and are removed from the screen. The next player then tries to find a pair until all the cards are gone. The winner is the one who correctly matches the most pairs.

In some card versions of this game a successful player is allowed to turn over another two cards. However, the game plays better in this 12-card version if one player cannot clear the decks so quickly.

Here is a money raising tip. Those readers who are involved in raising cash for charities or school funds may like to consider how the micro can help. *Concentration* could be used as a fund raiser, just as

many other programs featured in the book can. Simply set up your computer and charge visitors for the privilege of pressing the buttons!

```

1000 REM >> CONCENTRATION <<
1010 REM << ANDY/JIM >>
1020 SCREEN0:KEY OFF
1030 WIDTH 40
1040 COLOR 15,12
1050 DEFINT A-Z
1060 DIM PAC$(5,13),CC(3),SC(2)
1070 DIM PA(52),AV(52)
1080 DIM CX(12),CY(12)
1090 DIM CA(12),BC(12)
1100 DIM PLC(1,2),AC(12)
1110 GOSUB 2830
1120 CLS:GOSUB 1260
1130 SC(1)=0:SC(2)=0
1140 GOSUB 1260
1150 GOSUB 1380
1160 GOSUB 1820
1170 GOSUB 2070
1180 FOR A=1 TO 12:AC(A)=1:NEXT A
1190 P=1
1200 GOSUB 2560
1210 GOSUB 2250
1220 GOTO 2440
1230 IF P=1 THEN P=2 ELSE P=1
1240 GOTO 1200
1250 REM >>>> GENERATE PACK <<<<
1260 FOR SUIT=1 TO 4
1270 FOR CAR=1 TO 13
1280 PAC$(SUIT,CAR)=CHR$(SUIT+217)
1290 IF CAR<11 AND CAR>1 THEN PAC$(SU
IT,CAR)=PAC$(SUIT,CAR)+CHR$(CAR+220)
1300 IF CAR=1 THEN PAC$(SUIT,CAR)=PAC
$(SUIT,CAR)+CHR$(231)
1310 IF CAR=11 THEN PAC$(SUIT,CAR)=PA
C$(SUIT,CAR)+CHR$(232)
1320 IF CAR=12 THEN PAC$(SUIT,CAR)=PA
C$(SUIT,CAR)+CHR$(233)
1330 IF CAR=13 THEN PAC$(SUIT,CAR)=PA
C$(SUIT,CAR)+CHR$(234)
1340 NEXT CAR
1350 NEXT SUIT
1360 RETURN

```



```

1370 REM >>>> SHUFFLE PACK <<<<<
1380 FOR N=1 TO 52
1390 PA(N)=N:AV(N)=1
1400 NEXT N
1410 FOR A=1 TO 30
1420 N1=INT(RND(-TIME)*51)+1
1430 N2=INT(RND(-TIME)*51)+1
1440 CC=PA(N1)
1450 PA(N1)=PA(N2)
1460 PA(N2)=CC
1470 NEXT A
1480 RETURN
1490 REM >>>> DRAW CARD <<<<<
1500 LOCATE CX-1,CY
1510 PRINT CHR$(200);:FOR Z=1 TO 5:PR
INT CHR$(204);:NEXT Z:PRINT CHR$(203)
1520 LOCATE CX-1,CY+1
1530 PRINT CHR$(206);RIGHT$(PAC$(SUIT
,CAR),1);
1540 PRINT LEFT$(PAC$(SUIT,CAR),1);:P
RINT STRING$(3,CHR$(208));CHR$(207)
1550 LOCATE CX-1,CY+2
1560 PRINT CHR$(206);STRING$(5,CHR$(2
08));CHR$(207)
1570 LOCATE CX-1,CY+3
1580 PRINT CHR$(206);CHR$(208);CHR$(2
08);
1590 PRINT LEFT$(PAC$(SUIT,CAR),1);CH
R$(208);CHR$(208);CHR$(207)
1600 LOCATE CX-1,CY+4
1610 PRINT CHR$(206);STRING$(5,CHR$(2
08));CHR$(207)
1620 LOCATE CX-1,CY+5
1630 PRINT CHR$(206);CHR$(208);CHR$(2
08);CHR$(208);CHR$(208);RIGHT$(PAC$(S
UIT,CAR),1);CHR$(207)
1640 LOCATE CX-1,CY+6
1650 PRINT CHR$(201);STRING$(5,CHR$(2
05));CHR$(202)
1660 RETURN
1670 REM >>> DRAW CARD BACK <<<<
1680 LOCATE CX-1,CY
1690 PRINT CHR$(209);STRING$(5,CHR$(2
13));CHR$(210)
1700 FOR NN=1 TO 5:LOCATE CX-1,CY+NN

```

```

1710 PRINT CHR$(215);STRING$(5,CHR$(2
17) );CHR$(216)
1720 NEXT NN
1730 LOCATE CX-1,CY+6:PRINT CHR$(211)
;STRING$(5,CHR$(214));CHR$(212)
1740 RETURN
1750 REM >>>> ERASE CARD <<<<<
1760 LOCATE CX-1,CY
1770 FOR Y=1 TO 7
1780 LOCATE CX-1,CY+Y-1:PRINT"
"
1790 NEXT Y
1800 RETURN
1810 REM >>>> GENERATE CARDS <<<<
1820 RESTORE 2020:FOR A=1 TO 12
1830 CA(A)=1:NEXT A
1840 FOR A=1 TO 10 STEP 2
1850 READ C1,C2
1860 C=INT(RND(-TIME)*11)+1
1870 IF CA(C)=0 THEN 1860
1880 BC(C)=C1:CA(C)=0
1890 C=INT(RND(-TIME)*11)+1
1900 IF CA(C)=0 THEN 1890
1910 BC(C)=C2:CA(C)=0
1920 NEXT A
1930 A=0
1940 A=A+1
1950 IF A<13 THEN IF CA(A)=1 THEN REA
D BC(A):CA(A)=0 ELSE 1940
1960 A=A+1
1970 IF A<13 THEN IF CA(A)=1 THEN REA
D BC(A):CA(A)=0 ELSE 1960
1980 FOR A=1 TO 12
1990 READ CX(A),CY(A)
2000 NEXT A
2010 RETURN
2020 DATA 1,14,2,15,3,16,4,17,5,18,6,
19
2030 DATA 2,0,10,0,18,0,26,0
2040 DATA 2,7,10,7,18,7,26,7
2050 DATA 2,14,10,14,18,14,26,14
2060 REM >>>> DRAW SCREEN <<<<<
2070 CLS
2080 FOR A=1 TO 12
2090 CX=CX(A):CY=CY(A)

```

```

2100 GOSUB 1680
2110 NEXT A
2120 LOCATE 33,3:PRINT "Score 1"
2130 LOCATE 33,6:PRINT "Score 2"
2140 LOCATE 32,15:PRINT "Player 1"
2150 LOCATE 32,17:PRINT "Player 2"
2160 RETURN
2170 REM >>>>      DECODE CARD  '<<<<<
2180 S=PLC(Q,W)/13:CAR=PLC(Q,W) MOD 1
3: IF CAR=0 THEN CAR=13
2190 IF S>=0 THEN SUIT=1
2200 IF S>=1 THEN SUIT=2
2210 IF S>=2 THEN SUIT=3
2220 IF S>=3 THEN SUIT=4
2230 RETURN
2240 REM >>>>      INPUT NUMBER  <<<<<
2250 LOCATE 1,21:PRINT STRING$(35," ")
)
2260 LOCATE 1,21:PRINT "Enter first n
o.";
2270 INPUT N1
2280 PLAY "03L64CRE"
2290 IF N1<1 OR N1>12 THEN 2260
2300 IF AC(N1)=0 THEN 2260
2310 PLC(1,1)=BC(N1):Q=1:W=1
2320 GOSUB 2180:CX=CX(N1):CY=CY(N1)
2330 GOSUB 1500
2340 LOCATE 1,21:PRINT "Enter second
no.";
2350 INPUT N2
2360 PLAY "03L64CRE"
2370 IF N2<1 OR N2>12 OR N1=N2 THEN 2
340
2380 IF AC(N2)=0 THEN 2340
2390 PLC(1,2)=BC(N2):Q=1:W=2
2400 GOSUB 2180:CX=CX(N2):CY=CY(N2)
2410 GOSUB 1500
2420 RETURN
2430 REM >>>>      CHECK PAIR  <<<<<
2440 Q=1:W=1:GOSUB 2180
2450 A1=CAR:Q=1:W=2:GOSUB 2180
2460 A2=CAR
2470 IF A1=A2 AND N1<>N2 THEN 2620
2480 LOCATE 1,21:PRINT STRING$(35," ")
)

```

```

2490 PLAY "02L64BAGFEDCO1BAGFEDC"
2500 LOCATE 1,21:PRINT "Cards do not
match"
2510 GOSUB 3250
2520 CX=CX(N1):CY=CY(N1)
2530 GOSUB 1680:CX=CX(N2):CY=CY(N2)
2540 GOSUB 1680:GOTO 1230
2550 REM >>>> PRINT SCORES <<<<<
2560 LOCATE 33,4:PRINT SC(1)
2570 LOCATE 33,7:PRINT SC(2)
2580 IF P=1 THEN LOCATE 32,15:PRINT "
PLAYER 1":LOCATE 32,17:PRINT "player
2"
2590 IF P=2 THEN LOCATE 32,17:PRINT "
PLAYER 2":LOCATE 32,15:PRINT "player
1"
2600 RETURN
2610 REM >>>>> PAIR FOUND <<<<<<
2620 LOCATE 1,21:PRINT STRING$(35," "
)
2630 PLAY "05L64CDEFGAB06CDEFGAB"
2640 LOCATE 1,21:PRINT "Pair found"
2650 AC(N1)=0:AC(N2)=0
2660 GOSUB 3250
2670 CX=CX(N1):CY=CY(N1)
2680 GOSUB 1760:CX=CX(N2):CY=CY(N2)
2690 GOSUB 1760:SC(P)=SC(P)+5
2700 IF SC(1)+SC(2)=30 THEN 2730
2710 GOTO 1230
2720 REM >>>>> END OF GAME <<<<<
2730 CLS
2740 PRINT:PRINT "Player 1 score: ";SC
(1)
2750 PRINT:PRINT "Player 2 score: ";SC
(2)
2760 PRINT:PRINT:LOCATE 5,10:PRINT "P
ress 'SPACE' for another game,"
2770 LOCATE 10,12:PRINT "or 'Q' to qu
it."
2780 A$=INKEY$
2790 IF A$=" " THEN 1120
2800 IF A$="Q" OR A$="q" THEN END
2810 GOTO 2780
2820 REM >>>> CHARACTERS <<<<<<
2830 RESTORE 2900

```

```

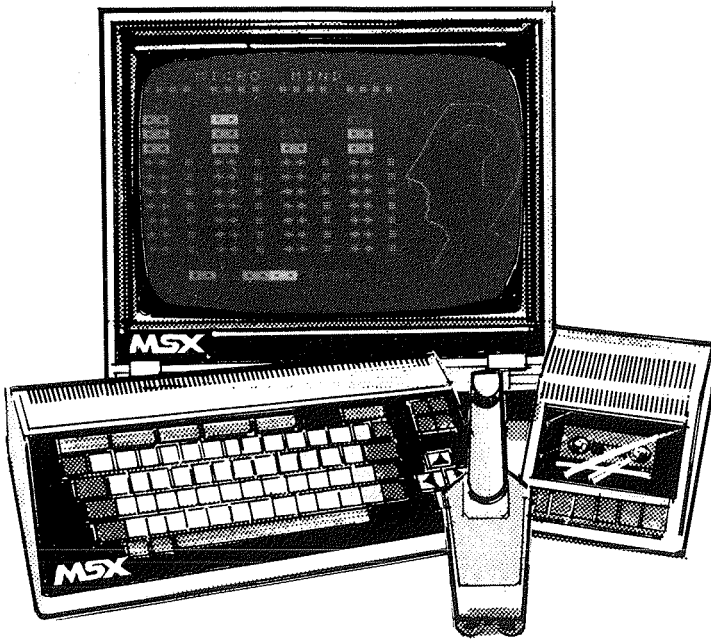
2840 FOR C=200 TO 234
2850 FOR B=0 TO 7
2860 READ A:VPOKE 2048+C*8+B,A
2870 NEXT B
2880 NEXT C
2890 RETURN
2900 DATA 0,60,124,124,124,124,124,12
4
2910 DATA 124,124,124,124,124,124,60,
0
2920 DATA 248,248,248,248,248,248,240
,0
2930 DATA 0,240,248,248,248,248,248,2
48
2940 DATA 0,252,252,252,252,252,252,2
52
2950 DATA 252,252,252,252,252,252,252
,0
2960 DATA 124,124,124,124,124,124,124
,124
2970 DATA 248,248,248,248,248,248,248
,248
2980 DATA 252,252,252,252,252,252,252
,252
2990 DATA 0,60,84,104,84,104,84,104
3000 DATA 0,240,88,168,88,168,88,168
3010 DATA 84,104,84,104,84,104,60,0
3020 DATA 88,168,88,168,88,168,240,0
3030 DATA 0,252,84,168,84,168,84,168
3040 DATA 84,168,84,168,84,168,252,0
3050 DATA 85,106,85,106,85,106,85,106
3060 DATA 88,168,88,168,88,168,88,168
3070 DATA 84,168,84,168,84,168,84,168
3080 DATA 252,140,140,84,4,84,220,140
3090 DATA 220,140,4,4,4,140,220,252
3100 DATA 252,180,0,0,0,132,204,252
3110 DATA 220,140,4,4,4,4,220,140
3120 DATA 140,116,244,204,188,124,4,2
52
3130 DATA 4,244,236,204,244,116,140,2
52
3140 DATA 236,204,172,108,4,236,236,2
52
3150 DATA 4,124,12,244,244,116,140,25
2

```

```
3160 DATA 204,188,124,12,116,116,140,  
252  
3170 DATA 4,244,236,220,188,188,188,2  
52  
3180 DATA 140,116,116,140,116,116,140  
,252  
3190 DATA 140,116,116,132,244,236,156  
,252  
3200 DATA 100,88,88,88,88,88,100,252  
3210 DATA 220,172,116,116,4,116,116,2  
52  
3220 DATA 196,236,236,236,236,108,156  
,252  
3230 DATA 140,116,116,116,84,108,148,  
252  
3240 DATA 116,108,92,60,92,108,116,25  
2  
3250 FOR T=1 TO 2000:NEXT T:RETURN
```

16

Micro Mind



Colour Your Reasoning

Versions of this well known and widely enjoyed game can be found on all microcomputers. In writing this implementation of the game, care has been taken to produce a good looking layout that makes play enjoyable.

The object of the game is to work out what the secret 6-colour combination is, by placing coloured markers in a row.

The computer will then indicate how accurate the placing was:

A face indicates the right colour, right place.

An exclamation mark indicates the wrong colour, wrong place.

A question mark indicates the right colour, wrong place.

A joystick or the cursor pad is used to select the colours by moving an arrow to the colour required. The fire button or space bar then places it. The next ones are then entered in the same manner until the row is full and the result is shown.

There are ten chances to guess the correct sequence before each

game ends. The computer will then display the correct combination at the top of the screen. This will remain in view to allow analysis of where you went wrong! Pressing P starts play again.

The game can be played competitively by each person aiming to achieve a lower number of lines to work out the combinations.

```

1000 REM <<<<<< MICROMIND ISSI >>>>
1010 REM
1020 SCREEN 2,0,0:COLOR 10,1,0:CLS
1030 MAXFILES=2:OPEN "GRP:S" FOR OUTP
UT AS 1
1040 PRESET (50,175):PRINT #1,"COLOUR
  DISPLAY (Y/N) ?"
1050 A$=INPUT$(1):A=ASC(A$) AND 223
1060 IF A=89 THEN GR=0:GOTO 1090
1070 IF A=78 THEN GR=1:GOTO 1090
1080 GOTO 1050
1090 CLS
1100 CX=2:CY=180
1110 PRESET (8,4):PRINT #1,"      MICR
O MIND"
1120 W$="####"
1130 PRESET (8,15):PRINT #1,W$;W$;W$;
W$
1140 W$=" "+CHR$(1)+CHR$(79)+CHR$(1)+
CHR$(79)+" "+CHR$(240)
1150 FOR N=40 TO 150 STEP 12:COLOR 9
1160 PRESET (8,N):PRINT #1,W$;W$;W$;W
$
1170 NEXT N
1180 GOSUB 1530
1190 GOSUB 1260
1200 GOSUB 1330
1210 GOSUB 1630
1220 IF A$="YYYY" THEN GOTO 1950 ELSE
  A$=""
1230 TRY=TRY+1:IF TRY=10 THEN GOTO 18
30 ELSE GOTO 1200
1240 REM >>>> RANDOM SEQUENCE <<<<
1250 REM
1260 FOR PLA=1 TO 4
1270 N=INT(RND(-TIME)*4)+2
1280 NN(PLA)=N
1290 NEXT PLA
1300 RETURN

```



```

1310 REM >>>>>> INPUT COLOURS <<<<<
1320 REM
1330 RESTORE:PRESET (40,170):FOR N=1
TO 6:READ C,R:COLOR C:C#=CHR$(1)+CHR$(
R)
1340 IF GR=0 THEN PRINT #1,CHR$(1)+CH
R$(72)+CHR$(1)+CHR$(72); ELSE PRINT #
1,C#;" ";
1350 NEXT N
1360 TX=1:CY=180:W#=CHR$(1)+CHR$(75)
1370 COLOR 9:PRESET (CX*16+8,CY):PRIN
T #1,W#;;FOR W=1 TO 10:NEXT W
1380 COLOR 1:PRESET (CX*16+8,CY):PRIN
T #1,W#;
1390 GOSUB 2110
1400 IF LE=1 THEN CX=CX-1
1410 IF RI=1 THEN CX=CX+1
1420 IF CX=8 THEN CX=2
1430 IF CX=1 THEN CX=7
1440 IF FI=1 THEN PLAY "L64CDE":GOTO
1460
1450 GOTO 1370
1460 PRESET (TX*40-24,TRY*12+40):COLO
R 1:PRINT #1,CHR$(1)+CHR$(79)+CHR$(1)
+CHR$(79)
1470 RESTORE:FOR W=1 TO CX-1:READ C,R
:NEXT W:COLOR C,1:C#=CHR$(1)+CHR$(R)
1480 PRESET (TX*40-24,TRY*12+40)
1490 IF GR=0 THEN PRINT #1,CHR$(1)+CH
R$(72)+CHR$(1)+CHR$(72) ELSE PRINT #1
,C#;C#
1500 PE(TX)=CX-1:TX=TX+1
1510 IF TX=5 THEN RETURN ELSE GOTO 13
70
1520 REM >>>>>>>>> HEAD <<<<<<<<<<<
1530 COLOR 7:PSET (250,160)
1540 RESTORE 2180
1550 FOR N=1 TO 25:READ A,B:A=A/2.5:B
=192-B/2
1560 IF A<180 THEN A=A+25
1570 D$="M"+STR$(INT(A))+", "+STR$(INT
(B))
1580 DRAW D$:NEXT N
1590 PSET (220,100):DRAW "U20E10U10H1
0L10G10"

```

```

1600 RETURN
1610 REM >>>>>>>>> CHECK <<<<<<<<<<<
1620 REM
1630 A$=""
1640 FOR X=1 TO 4:COLOR 1
1650 PRESET (X*40,TRY*12+40):PRINT #1
,CHR$(240):COLOR 4
1660 IF PE(X)=NN(X) THEN Z=66 ELSE Z=
33
1670 PRESET (X*40,TRY*12+40):IF Z=66
THEN PRINT #1,CHR$(1);
1680 PRINT #1,CHR$(Z)
1690 IF Z=66 THEN Z$="Y" ELSE Z$="N"
1700 A$=A$+Z$:NEXT X
1710 TX=1:DO=0
1720 FOR X=1 TO 4
1730 IF MID$(A$,X,1)="Y" THEN PLAY "E
":W=4:GOTO 1790
1740 FOR W=1 TO 4
1750 IF PE(X)=NN(W) AND MID$(A$,W,1)=
"N" THEN GOTO 1770
1760 GOTO 1780
1770 COLOR 1:PRESET (X*40,TRY*12+40):
PRINT #1,CHR$(33):COLOR 10:PRESET (X*
40,TRY*12+40):PRINT #1,CHR$(63):DO=1
1780 NEXT W:IF DO=0 THEN PLAY "A"
1790 TX=TX+1:PLAY "F":NEXT X
1800 RETURN
1810 REM >>>>>>>>>> LOST <<<<<<<<<<<
1820 REM
1830 LINE (8,15)-(170,25),1,BF
1840 FOR N=1 TO 4
1850 RESTORE:FOR W=1 TO NN(N):READ C,
R:NEXT W:C$=CHR$(1)+CHR$(R)
1860 IF GR=0 THEN C$=CHR$(1)+CHR$(72)
1870 PRESET (N*40-24,15):COLOR C:PRIN
T #1,C$;C$;C$;C$
1880 NEXT N
1890 PRESET (208,20):PRINT #1,"SORRY!
"
1900 PLAY "L8GFL4EL8FEL4DL8EDL4CD"
1910 FOR N=1 TO 4000:NEXT N
1920 CLS:GOTO 2050
1930 REM >>>>>>>>>>> WON <<<<<<<<<<<
1940 REM

```

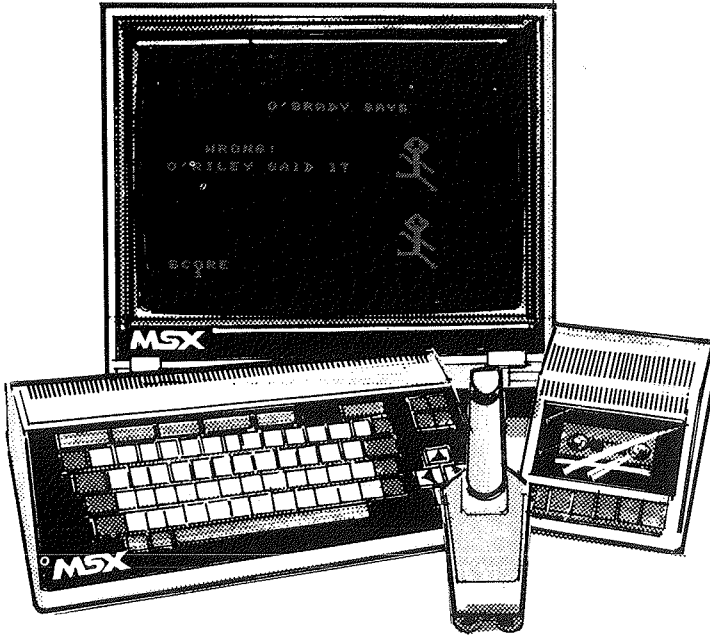
```

1950 FOR N=1 TO 4
1960 RESTORE:FOR W=1 TO NN(N):READ C,
R:C#=CHR$(1)+CHR$(R):NEXT W
1970 LINE(N*40-24,15)-(N*40+16,25),1,
BF
1980 IF GR=0 THEN C#=CHR$(1)+CHR$(72)
1990 PRESET (N*40-24,15):COLOR C:PRIN
T #1,STRING$(4,CHR$(1)+C#)
2000 NEXT N
2010 PLAY "L8D#F#G#L4A#L8R8A#G#F#G#FD
#L4C#"
2020 CLS
2030 PRESET (90,50):PRINT #1,"WELL DO
NE !"
2040 PRESET (65,60):PRINT #1,"IT TOOK
";TRY+1;" TRIES !"
2050 PRESET (80,150):PRINT #1,"AGAIN
(Y/N) ?"
2060 A#=INPUT$(1):A=ASC(A#) AND 223
2070 IF A=89 THEN TRY=0:GOTO 1090
2080 GOTO 2060
2090 REM >>>>>>>> INPUT <<<<<<<<<
2100 REM
2110 LE=0:RI=0:FI=0
2120 DI=STICK(0) OR STICK (1) OR STIC
K(2)
2130 IF DI=7 THEN LE=1
2140 IF DI=3 THEN RI=1
2150 FI=ABS(STRIG(0) OR STRIG(1) OR S
TRIG(2))
2160 RETURN
2170 DATA 7,77,2,67,10,69,15,70,8,68,
13,78
2180 DATA 560,110,605,180,615,220,615
,250,605,290,555,320,500,320
2190 DATA 415,290,395,250,365,200,375
,195,390,190,400,170,395,165
2200 DATA 400,160,395,155,400,150,400
,145,395,140,395,130,400,125
2210 DATA 415,120,425,110,430,95,405,
50

```

17

O'Grady Says



O'Grady Says: Type This In

As far as I know this is the world's first implementation of the well known game, sometimes also known as Simple Simon Says.

This program can be most helpful in teaching the concept of left and right, but be warned, as it stands the program means *its* left or right, not yours!

You start each game with just one point. The speech bubble will then announce the action required together with the name of the person giving the order. If O'Grady says 'do it' then you do it fast! If O'Riley says 'do it' then you do *not*! The little chap at the top follows all commands.

There are four possible actions which you may be asked to do. Up and Down are actioned using the up or down cursor or joystick as normal. Left or Right are obtained using the respective key in relation to the on-screen character. This means that pressing the left cursor will actually lift up the chap's right arm.

This swap has been incorporated to provide more of a mental challenge. It can be quite a struggle to work out which is left, which is

right and of course whether O'Grady said 'do it'. Play the game and you will see what I mean.

Speedy mental processing is important in *O'Grady Says* or lack of action may be taken as intentional. This could mean that you lose points or, if you are lucky, doing nothing could be just the right response.

```

1000 REM > O'GRADY SAYS <>JIM/ANDY <
1010 SCREEN 1,1:KEY OFF
1020 FOR A=1 TO 9
1030 FOR B=1 TO 8
1040 READ A$
1050 S$(A)=S$(A)+CHR$(VAL("&H"+A$))
1060 NEXT B
1070 SPRITE$(A)=S$(A)
1080 NEXT A
1090 DATA 18,24,42,A5,81,5A,24,18
1100 DATA 31,9,5,3,1,1,1,1
1110 DATA 1,1,1,3,5,9,11,21
1120 DATA 8C,90,A0,C0,80,80,80,80
1130 DATA 80,80,80,C0,A0,90,88,84
1140 DATA 41,7F,0,0,0,0,0,0
1150 DATA 1,1,2,4,8,10,20,60
1160 DATA 82,FE,0,0,0,0,0,0
1170 DATA 80,80,40,20,10,8,4,6
1180 REM >>> GAME STARTS HERE <<<<<<
1190 GOSUB 1600
1200 SC=1:GOSUB 1900
1210 FOR AA=1 TO 500:NEXT AA
1220 GOSUB 1780
1230 GOSUB 1660
1240 IF GO=PO AND OG=1 THEN MO=MO+1:S
C=SC+1:GOSUB 1900:GOTO 1210
1250 IF TI=200 AND OG=2 THEN 1290
1260 IF TI=200 THEN 1280
1270 IF OG=2 OR PO<>GO THEN SC=SC-1:M
O=MO+1:GOSUB 2000
1280 IF SC<1 THEN 2080
1290 GOSUB 1900:GOTO 1210
1300 GOTO 1300
1310 REM >>>> PUT 'DOWN' FIGURE <<<<
1320 PUT SPRITE M,(X,Y),15,1
1330 PUT SPRITE M+1,(X-8,Y+16),15,3
1340 PUT SPRITE M+2,(X+8,Y+16),15,5
1350 PUT SPRITE M+3,(X-8,Y+32),15,7

```

```

1360 PUT SPRITE M+4, (X+8,Y+32), 15, 9
1370 RETURN
1380 REM >>>> PUT 'UP' FIGURE <<<<<<
1390 PUT SPRITE M, (X,Y), 15, 1
1400 PUT SPRITE M+1, (X-8,Y+16), 15, 2
1410 PUT SPRITE M+2, (X+8,Y+16), 15, 4
1420 PUT SPRITE M+3, (X-8,Y+32), 15, 6
1430 PUT SPRITE M+4, (X+8,Y+32), 15, 8
1440 RETURN
1450 REM >>>> PUT 'UP LEFT' FIGURE <
1460 PUT SPRITE M, (X,Y), 15, 1
1470 PUT SPRITE M+1, (X-8,Y+16), 15, 3
1480 PUT SPRITE M+2, (X+8,Y+16), 15, 4
1490 PUT SPRITE M+3, (X-8,Y+32), 15, 7
1500 PUT SPRITE M+4, (X+8,Y+32), 15, 8
1510 RETURN
1520 REM >>> PUT 'UP RIGHT' FIGURE <
1530 PUT SPRITE M, (X,Y), 15, 1
1540 PUT SPRITE M+1, (X-8,Y+16), 15, 2
1550 PUT SPRITE M+2, (X+8,Y+16), 15, 5
1560 PUT SPRITE M+3, (X-8,Y+32), 15, 6
1570 PUT SPRITE M+4, (X+8,Y+32), 15, 9
1580 RETURN
1590 REM >>>>>> DRAW SCREEN <<<<<<<
1600 LOCATE 10,0:PRINT "O'GRADY SAYS"
1610 LOCATE 2,18:PRINT "SCORE"
1620 X=180:Y=30:M=1:GOSUB 1320
1630 X=180:Y=100:M=6:GOSUB 1320
1640 RETURN
1650 REM >>>> KEYBOARD INPUT <<<<
1660 TI=0
1670 AA=STICK(0) OR STICK(1)
1680 IF AA=0 THEN 1730
1690 IF AA=8 THEN GO=1:GOTO 1750
1700 IF AA=2 THEN GO=2:GOTO 1750
1710 IF AA=1 THEN GO=3:GOTO 1750
1720 IF AA=5 THEN GO=4:GOTO 1750
1730 TI=TI+1:IF TI=200 THEN 1950
1740 GOTO 1670
1750 X=180:Y=100:M=6
1760 ON GO GOTO 1460,1530,1390,1320
1770 REM >>>> CHOOSE POSITION <<<<<<
1780 OG=INT(RND(1)*5)+1:IF OG<3 THEN
OG=2:GOTO 1800
1790 OG=1

```

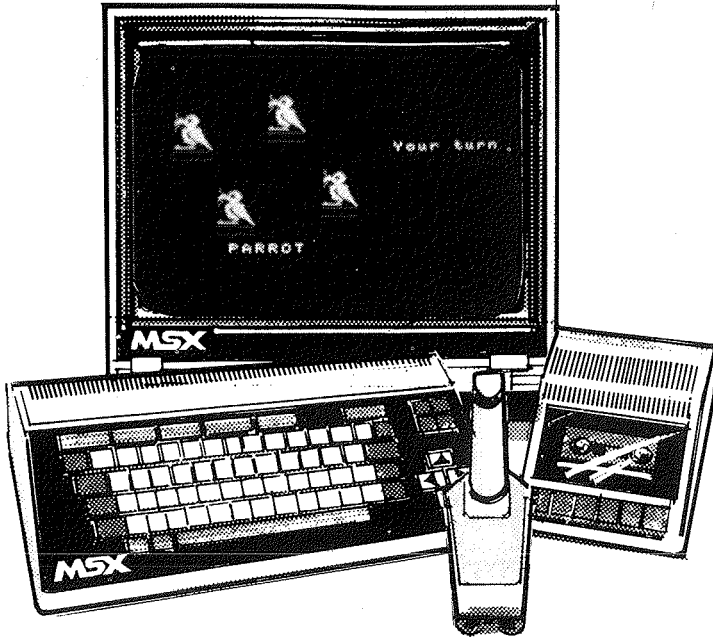
```

1800 PO=INT(RND(1)*4)+1
1810 X=180:Y=30:M=1
1820 IF OG=1 THEN LOCATE 3,5:PRINT "O
'GRADY SAYS"
1830 IF OG=2 THEN LOCATE 3,5:PRINT "O
'RILEY SAYS"
1840 IF PO=1 THEN LOCATE 3,7:PRINT "U
P LEFT"
1850 IF PO=2 THEN LOCATE 3,7:PRINT "U
P RIGHT"
1860 IF PO=3 THEN LOCATE 3,7:PRINT "M
OVE UP"
1870 IF PO=4 THEN LOCATE 3,7:PRINT "M
OVE DOWN"
1880 ON PO GOTO 1460,1530,1390,1320
1890 REM >>>> PRINT SCORE <<<<<<<<<
1900 LOCATE 3,5:PRINT " "
1910 LOCATE 3,7:PRINT " "
1920 LOCATE 3,19:PRINT SC
1930 RETURN
1940 REM >>> TIME OUT <<<<<<<<<<<<
1950 LOCATE 3,5:PRINT " "
1960 LOCATE 3,7:PRINT " "
1970 IF OG=2 THEN LOCATE 3,5:PRINT "W
ELL DONE,":LOCATE 3,7:PRINT "O'RILEY
SAID IT":SC=SC+1:FOR T=1 TO 800:NEXT
T:LOCATE 2,7:PRINT " "
:RETURN
1980 LOCATE 3,7:PRINT "OUT OF TIME!":
SC=SC-1:FOR T=1 TO 800:NEXT T:RETURN
1990 REM >>>> WRONG ANSWER <<<<<<<<<
2000 LOCATE 3,5:PRINT " "
2010 LOCATE 3,7:PRINT " "
2020 LOCATE 5,5:PRINT "WRONG!"
2030 IF OG=2 THEN LOCATE 2,7:PRINT "O
'RILEY SAID IT"
2040 FOR T=1 TO 800:NEXT T
2050 LOCATE 2,7:PRINT "
"
2060 RETURN
2070 REM >>>> END OF GAME <<<<<<<<<
2080 LOCATE 3,5:PRINT " "
2090 LOCATE 3,7:PRINT " "
2100 LOCATE 3,7:PRINT "OUT OF TRIES"
2110 FOR T=1 TO 800:NEXT T

```

```
2120 SC=0:LOCATE 3,7:PRINT "  
"  
2130 LOCATE 3,7:PRINT "TRY AGAIN"  
2140 FOR T=1 TO 800:NEXT T  
2150 LOCATE 3,7:PRINT "  
2160 GOTO 1190
```


18 Parrot



Who's a Pretty Program?

Here are four pretty parrots, each a different colour and each holding a 'string' in his mouth.

When the program runs the parrots will commence pulling the strings in sequence. The first set will consist of only one pull. Then it will be your turn to copy the sequence.

The perches of the birds have been arranged to correspond to the arrangement of the cursor keys. Thus by pressing 'up' the top bird will pull a string, 'down' will activate the bottom bird and so on. As usual a joystick may be used instead of the cursor pad.

The computer produces a sequence which starts off simply but gets very long. During testing of this program no one was able to remember more than a sequence of 25. Amnesia usually sets in at around 12. As it stands the longest sequence that the parrots will produce is 50 long. If you remember that length then you must have some sort of photographic memory – or perhaps you are an android!

This simple game can be great fun at parties where all sorts of forfeits or penalties may be devised for the losers.

```

1000 REM >>>> PARROT  ANDY/JIM <<<
1010 REM
1020 X=RND(-TIME)
1030 DIM NO(50)
1040 SCREEN 1,0,0
1050 GOSUB 1710
1060 GOSUB 1820
1070 SC=0:HI=0
1080 GOSUB 2050
1090 FOR A=1 TO 4
1100 GOSUB 1530
1110 NEXT A
1120 PO=0
1130 REM >>>          MAIN LOOP          <<<
1140 N=INT(RND(1)*4)+1
1150 PO=PO+1
1160 NO(PO)=N
1170 GOSUB 2160:LOCATE 20,10:PRINT "M
y turn"
1180 FOR B=1 TO PO
1190 GOSUB 2210
1200 NEXT B
1210 FOR T=1 TO 500:NEXT T
1220 B=1
1230 GOSUB 2160:LOCATE 20,10:PRINT "Y
our turn"
1240 A$=INKEY$:IF A$="" THEN 1240
1250 AA=ASC(A$)
1260 IF AA=29 THEN A=4:GOSUB 1650:GOS
UB 1590:FOR T=1 TO 100:NEXT T
1270 IF AA=28 THEN A=2:GOSUB 1650:GOS
UB 1590:FOR T=1 TO 100:NEXT T
1280 IF AA=30 THEN A=1:GOSUB 1650:GOS
UB 1590:FOR T=1 TO 100:NEXT T
1290 IF AA=31 THEN A=3:GOSUB 1650:GOS
UB 1590:FOR T=1 TO 100:NEXT T
1300 IF AA<28 OR AA>31 THEN 1240
1310 GOSUB 1650
1320 GOSUB 1530
1330 IF NO(B)=A THEN 1340 ELSE 1370
1340 IF B=PO THEN GOSUB 2160:LOCATE 2
0,10:PRINT "Well done":LOCATE 20,12:P
RINT PO;"so far"
1350 IF B=PO THEN FOR T=1 TO 1000:NEX
T T:GOSUB 2160:GOTO 1140

```

```

1360 B=B+1:GOTO 1240
1370 GOSUB 2160
1380 LOCATE 20,10:PRINT "Wrong":LOCAT
E 20,11:PRINT "Sequence":LOCATE 20,13
:PRINT "After ";PO-1:LOCATE 20,14:PRI
NT "moves."
1390 FOR T=1 TO 1000:NEXT T
1400 GOSUB 2160
1410 LOCATE 20,10:PRINT "Sequence":LO
CATE 20,12:PRINT "   was."
1420 FOR T=1 TO 1000:NEXT T
1430 FOR B=1 TO PO
1440 GOSUB 2210
1450 NEXT B
1460 GOSUB 2160
1470 LOCATE 20,10:PRINT "Try again":X
=RND(-TIME)
1480 FOR T=1 TO 1000:NEXT T
1490 GOSUB 2160
1500 GOTO 1120
1510 END
1520 REM >>> PRINT UPRIGHT PARROT <<<
1530 LOCATE PX(A),PY(A):PRINT CHR$(20
0);CHR$(201);CHR$(202)
1540 LOCATE PX(A),PY(A)+1:PRINT CHR$(
203);CHR$(204);CHR$(205)
1550 LOCATE PX(A),PY(A)+2:PRINT CHR$(
206);CHR$(207);CHR$(208)
1560 LOCATE PX(A),PY(A)+3:PRINT CHR$(
209);CHR$(210);CHR$(211)
1570 RETURN
1580 REM>>> PRINT BENT DOWN PARROT <<
1590 PLAY "L8"+CHR$(67+A)
1600 LOCATE PX(A)-1,PY(A)+2:PRINT CHR
$(212);CHR$(213);CHR$(214);CHR$(215)
1610 LOCATE PX(A)-1,PY(A)+3:PRINT CHR
$(209);CHR$(210)
1620 LOCATE PX(A),PY(A):PRINT CHR$(21
6)
1630 RETURN
1640 REM >>>>      BLANK PARROT <<<<
1650 LOCATE PX(A),PY(A):PRINT "   "
1660 LOCATE PX(A),PY(A)+1:PRINT "   "
1670 LOCATE PX(A)-1,PY(A)+2:PRINT "
"

```

```

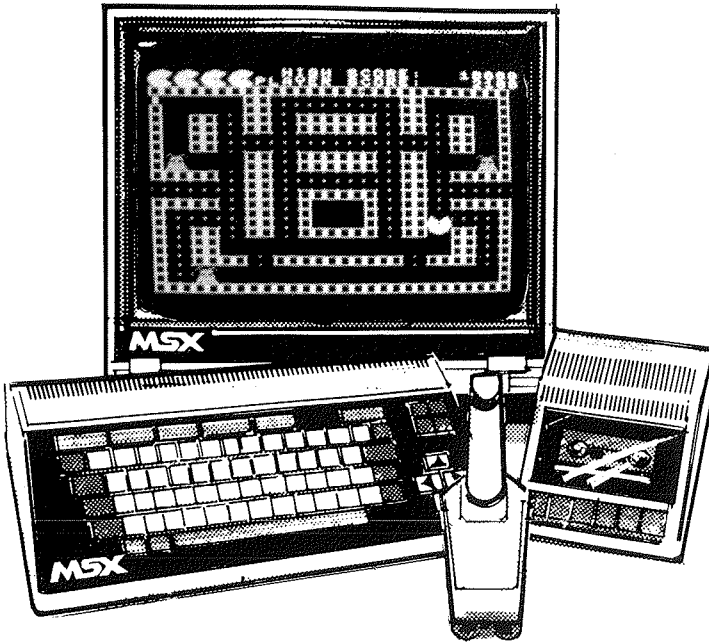
1680 LOCATE PX(A)-1,PY(A)+3:PRINT "
      "
1690 RETURN
1700 REM > DEFINE CHARACTERS <
1710 RESTORE 1860
1720 FOR C=1 TO 18
1730 READ CC
1740 FOR A=0 TO 7
1750 READ NN
1760 VPOKE CC*8+A,NN
1770 NEXT A
1780 NEXT C
1790 VPOKE &H2019,48:VPOKE &H201A,48:
VPOKE &H201B,96
1800 RETURN
1810 REM >>> SET UP VARIABLES <<<<
1820 PX(1)=10:PX(2)=14:PX(3)=6:PX(4)=
3
1830 PY(1)=5:PY(2)=13:PY(3)=15:PY(4)=
7
1840 RETURN
1850 REM >>> CHARACTERS <<<<
1860 DATA 200,16,16,16,16,0,14,31,63
1870 DATA 201,0,0,28,126,127,207,207,
207
1880 DATA 202,0,0,0,0,0,128,128
1890 DATA 203,63,49,40,8,16,33,35,3
1900 DATA 204,255,255,254,124,126,255
,255,255
1910 DATA 205,0,0,0,0,128,192,192
1920 DATA 206,3,3,3,3,3,1,1,1
1930 DATA 207,223,191,191,223,223,239
,247,251
1940 DATA 208,192,192,224,224,224,240
,240,248
1950 DATA 209,1,0,0,0,56,127,0,0
1960 DATA 210,253,126,12,24,48,252,0,
0
1970 DATA 211,248,120,60,28,14,6,3,0
1980 DATA 212,7,31,63,127,99,99,127,6
3
1990 DATA 213,144,225,247,251,139,115
,251,255
2000 DATA 214,0,254,255,255,255,255,2
55,255

```

```
2010 DATA 215,3,14,252,252,248,240,22
4,128
2020 DATA 216,16,16,16,16,16,16,16,16
2030 DATA 217,255,170,85,170,85,170,8
5,255
2040 REM >>>> DRAW SCREEN <<<<
2050 FOR Y=1 TO 4:LOCATE 10,Y:PRINT C
HR$(216):NEXT Y
2060 FOR Y=1 TO 12:LOCATE 14,Y:PRINT
CHR$(216):NEXT Y
2070 FOR Y=1 TO 14:LOCATE 6,Y:PRINT C
HR$(216):NEXT Y
2080 FOR Y=1 TO 6:LOCATE 3,Y:PRINT CH
R$(216):NEXT Y
2090 FOR X=0 TO 28:LOCATE X,0:PRINT C
HR$(217):NEXT X
2100 LOCATE 10,9:PRINT CHR$(217);CHR$(
217);CHR$(217)
2110 LOCATE 14,17:PRINT CHR$(217);CHR
$(217);CHR$(217)
2120 LOCATE 6,19:PRINT CHR$(217);CHR$(
217);CHR$(217)
2130 LOCATE 3,11:PRINT CHR$(217);CHR$(
217);CHR$(217)
2140 LOCATE 7,21:PRINT "PARROT"
2150 RETURN
2160 FOR Y=10 TO 15
2170 LOCATE 20,Y:PRINT STRING$(9," ")
2180 NEXT Y
2190 RETURN
2200 REM >>>> MOVE PARROT <<<<
2210 A=NO(B):GOSUB 1650:GOSUB 1590
2220 FOR T=1 TO 100:NEXT T
2230 GOSUB 1650:GOSUB 1530
2240 RETURN
```

19

Pick Man



An Arcade Gobble

This is the first of three arcade type games included in this book. It is a variation on a well known theme in which you guide your character around a maze eating up the dots as you go.

Life is never easy in the arcade world and so there are a couple of nasties out to eat up the dots before you do.

To make matters even worse they will kill you if they get you! The object of this variant is to pick up as many dots as possible before all the dots are gone, without being killed. Once all the dots are gone the next screen will be displayed.

Controls are, as usual, joystick or cursor pad, which change the direction of the Pick Man.

In the hands of a skilled and patient player, incredible high scores can be achieved.

```

1000 REM <<<<<< pick man >>>>>>>
1010 REM
1020 SCREEN 1,2,0:CLEAR500
1030 COLOR 15,0,0:CLS:KEY OFF
1040 RESTORE:DIM S$(11):HS%=10000
1050 ON SPRITE GOSUB 1930
1060 ON INTERVAL=35 GOSUB 1440
1070 GOSUB 2330:DEFUSR=342
1080 GOTO 1600
1090 REM ----- input -----
1100 FOR I%=0 TO 2
1110 ST%=STICK(I%):IF ST% THEN I%=2
1120 NEXT I%:RETURN
1130 A=USR(0):IP=ASC(INPUT$(1))
1140 RETURN
1150 REM ----- xychar -----
1160 C%=6144+INT(Y%/8)*32+X%/8:RETURN
1170 REM ----- over char -----
1180 Y%=Y%-1:GOTO 1210
1190 REM ----- under char -----
1200 Y%=Y%+16
1210 GOSUB 1160:CT%=C%+1:CL%=VPEEK(C%
):CR%=VPEEK(CT%):GOTO 1270
1220 REM ----- left char -----
1230 X%=X%-2:GOTO 1260
1240 REM ----- right char -----
1250 X%=X%+17
1260 GOSUB 1160:CT%=C%+32:CL%=VPEEK(C
%):CR%=VPEEK(CT%)
1270 IF CL%=251 THEN VPOKE C%,32:DT%=
DT%+1:IF PM% THEN PS%=PS%+50
1280 IF CR%=251 THEN VPOKE CT%,32:DT%
=DT%+1:IF PM% THEN PS%=PS%+50
1290 IF CL%=250 OR CR%=250 THEN RETUR
N
1300 C%=0:RETURN
1310 REM ----- rand dir -----
1320 H%=INT(RND(-TIME)*3-1)*4:IF H% T
HEN V%=0:RETURN ELSE V%=INT(RND(-TIME
)*3-1)*4:RETURN
1330 REM ----- place char -----
1340 PUT SPRITE 1,(X1%,Y1%),4,PB%
1350 PUT SPRITE 2,(X2%,Y2%),4,PB%
1360 PUT SPRITE 3,(X3%,Y3%),4,PB%
1370 PUT SPRITE 0,(X0%,Y0%),10,P0%

```

```

1380 RETURN
1390 REM ----- print score -----
1400 LOCATE 22,0:PRINT USING "#####
";HS%
1410 LOCATE 22,1:PRINT USING "#####
";PS%
1420 RETURN
1430 REM ----- move pickman -----
1440 PLAY"t255116o1c"
1450 T0%=C%:T1%=CT%:T2%=X%:T3%=Y%:T4%
=CL%:T5%=CR%:PM%=1:GOSUB 1100
1460 IF ST%=1 OR ST%=5 THEN P0%=0:H0%
=0:IF ST%=1 THEN V0%=-4:S0%=2 ELSE V0
%=4:S0%=4
1470 IF ST%=3 OR ST%=7 THEN P0%=0:V0%
=0:IF ST%=3 THEN H0%=4:S0%=3 ELSE H0%
=-4:S0%=1
1480 X%=X0%:Y%=Y0%
1490 IF V0%<0 THEN GOSUB 1180
1500 IF V0%>0 THEN GOSUB 1200
1510 IF H0%>0 THEN GOSUB 1250
1520 IF H0%<0 THEN GOSUB 1230
1530 IF C% THEN 1550
1540 X0%=X0%+H0%:Y0%=Y0%+V0%
1550 P0%=P0% XOR S0%
1560 IF X0%>248 THEN X0%=-8
1570 IF X0%<-8 THEN X0%=248
1580 C%=T0%:CT%=T1%:X%=T2%:Y%=T3%:CL%
=T4%:CR%=T5%:PM%=0:GOSUB 1400:GOTO 13
70
1590 REM ----- game -----
1600 GOSUB 2040
1610 X0%=120:Y0%=144:H0%=-8:V0%=0:S0%
=1
1620 GOSUB 1990
1630 P0%=1:PB%=5:SPRITE ON:INTERVAL O
N
1640 GOSUB 1340
1650 GOSUB 1130
1660 GOSUB 1340
1670 IF DT%=356 THEN INTERVAL OFF:GOT
O 1600
1680 PB%=PB% XOR 3
1690 REM ----- move baddies -----
1700 X%=X1%:Y%=Y1%

```



```

1710 IF H1%<0 THEN GOSUB 1230:GOTO 17
50
1720 IF H1%>0 THEN GOSUB 1250:GOTO 17
50
1730 IF V1%<0 THEN GOSUB 1180:GOTO 17
50
1740 IF V1%>0 THEN GOSUB 1200
1750 IF C% THEN GOSUB 1320:H1%=H%:V1%
=V%:GOTO 1700
1760 X1%=X1%+H1%:Y1%=Y1%+V1%
1770 X%=X2%:Y%=Y2%
1780 IF H2%<0 THEN GOSUB 1230:GOTO 18
20
1790 IF H2%>0 THEN GOSUB 1250:GOTO 18
20
1800 IF V2%<0 THEN GOSUB 1180:GOTO 18
20
1810 IF V2%>0 THEN GOSUB 1200
1820 IF C% THEN GOSUB 1320:H2%=H%:V2%
=V%:GOTO 1770
1830 X2%=X2%+H2%:Y2%=Y2%+V2%
1840 X%=X3%:Y%=Y3%
1850 IF H3%<0 THEN GOSUB 1230:GOTO 18
90
1860 IF H3%>0 THEN GOSUB 1250:GOTO 18
90
1870 IF V3%<0 THEN GOSUB 1180:GOTO 18
90
1880 IF V3%>0 THEN GOSUB 1200
1890 IF C% THEN GOSUB 1320:H3%=H%:V3%
=V%:GOTO 1840
1900 X3%=X3%+H3%:Y3%=Y3%+V3%
1910 GOTO 1660
1920 REM ----- lose life -----
1930 SPRITE OFF:INTERVAL OFF
1940 IF ABS(X0%-X1%)>15 AND ABS(X0%-X
2%)>15 AND ABS(X0%-X3%)>15 OR ABS(Y0%
-Y1%)>15 AND ABS(Y0%-Y2%)>15 AND ABS(
Y0%-Y3%)>15 THEN GOSUB 1340:SPRITE ON
:INTERVAL ON:GOTO 1990
1950 PLAY"t255116o3bagfedc"
1960 LF=LF-1:IF LF=0 THEN RETURN 1600
1970 PUT SPRITE LF+3,(LF*16,0),0,3
1980 RETURN 1610
1990 X1%=24:Y1%=24:H1%=8:V1%=0

```

```

2000 X2%=216:Y2%=24:H2%=-8:V2%=0
2010 X3%=120:Y3%=168:H3%=8:V3%=0:IF R
ND(-TIME)>.5 THEN H3%=-H3%
2020 RETURN
2030 REM ----- initialise -----
2040 GOSUB 2120:LF=5:DT%=0
2050 FOR I=4 TO 7:J=(I-3)*16
2060 PUT SPRITE I,(J,0),10,3
2070 NEXT I
2080 LOCATE 10,0:PRINT"HIGH SCORE:";
2090 LOCATE 8,1:PRINT"PLAYER SCORE:";
2100 RETURN
2110 REM ----- draw maze -----
2120 LOCATE 0,2
2130 PRINT S$(0)
2140 PRINT S$(1):PRINT S$(1)
2150 PRINT S$(2):PRINT S$(2)
2160 PRINT S$(3):PRINT S$(3)
2170 PRINT S$(4):PRINT S$(4)
2180 PRINT S$(5):PRINT S$(6)
2190 PRINT S$(7):PRINT S$(8)
2200 PRINT S$(9):PRINT S$(9)
2210 PRINT S$(2)
2220 PRINT S$(3):PRINT S$(3)
2230 PRINT S$(10)
2240 PRINT S$(11):PRINT S$(11)
2250 PRINT S$(0);
2260 RETURN
2270 REM ----- def shapes -----
2280 SP$=""
2290 FOR J=1 TO 32
2300 READ H$:SP$=SP$+CHR$(VAL("&h"+H$
))
2310 NEXT J
2320 RETURN
2330 FOR I=250 TO 251:K=I*8
2340 FOR J=0 TO 7
2350 READ H$:VPOKE J+K,VAL("&h"+H$)
2360 NEXT J,I:VPOKE 8223,96
2370 FOR I=0 TO 6
2380 GOSUB 2280:SPRITE$(I)=SP$
2390 NEXT I
2400 S1$=CHR$(250):S2$=CHR$(251)
2410 S3$=S1$+S1$:S4$=S2$+S2$
2420 S5$=S1$+S3$:S6$=S2$+S4$

```

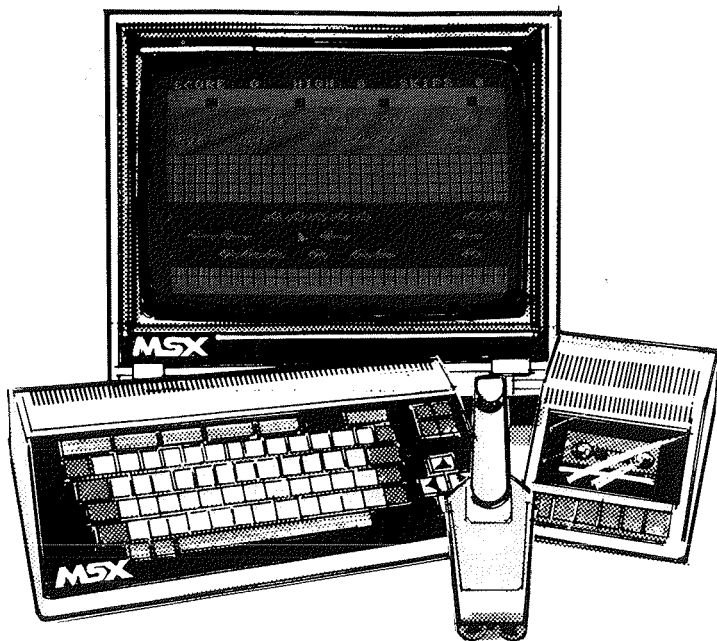
```

2430 S7$=S3$+S3$:S8$=S4$+S4$
2440 S9$=S3$+S7$:S0$=S4$+S8$
2450 S$(0)=STRING$(28,S1$)
2460 S$(1)=S1$+S0$+S3$+S0$+S8$+S3$+S0$
$+S1$
2470 S$(2)=S1$+S4$+S3$+S4$+S3$+S4$+S9$
$+S4$+S3$+S4$+S3$+S4$+S1$
2480 S$(3)=S1$+S4$+S3$+S0$+S0$+S0$+S3$
$+S4$+S1$
2490 S$(4)=S1$+S0$+S3$+S4$+S9$+S4$+S3$
$+S0$+S1$
2500 S$(5)=S1$+S7$+S4$+S3$+S8$+S0$+S3$
$+S4$+S1$+S7$
2510 S$(6)=S2$+S0$+S3$+S8$+S0$+S3$+S2$
$+S0$
2520 S$(7)=S2$+S0$+S3$+S4$+S9$+S4$+S3$
$+S0$+S2$
2530 S$(8)=S1$+S7$+S4$+S3$+S4$+S1$+SP
ACE$(4)+S1$+S4$+S3$+S4$+S7$+S1$
2540 S$(9)=S1$+S0$+S3$+S4$+S1$+SPACE$
(4)+S1$+S4$+S3$+S0$+S1$
2550 S$(10)=S1$+S4$+S7$+S4$+S7$+S9$+S
4$+S7$+S4$+S1$
2560 S$(11)=S1$+STRING$(26,S2$)+S1$
2570 RETURN
2580 REM ----- prog data -----
2590 DATA ff,c3,c3,c3,c3,c3,ff
2600 DATA 81,0,0,0,0,0,0,81
2610 REM
2620 DATA 3,f,3f,3f,7f,7f,ff,fe
2630 DATA fe,ff,7f,7f,3f,3f,f,3
2640 DATA c0,f0,fc,fc,fe,fe,ff,7f
2650 DATA 7f,ff,fe,fe,fc,fc,f0,c0
2660 DATA 3,f,3f,3f,1f,f,7,2
2670 DATA 2,7,f,1f,3f,3f,f,3
2680 DATA c0,f0,fc,fc,fe,fe,ff,7f
2690 DATA 7f,ff,fe,fe,fc,fc,f0,c0
2700 DATA 0,0,30,38,7c,7e,ff,fe
2710 DATA fe,ff,7f,7f,3f,3f,f,3
2720 DATA 0,0,c,1c,3e,7e,ff,7f
2730 DATA 7f,ff,fe,fe,fc,fc,f0,c0
2740 DATA 3,f,3f,3f,7f,7f,ff,fe
2750 DATA fe,ff,7f,7f,3f,3f,f,3
2760 DATA c0,f0,fc,fc,f8,f0,e0,40
2770 DATA 40,e0,f0,f8,fc,fc,f0,c0

```

2780 DATA 3, f, 3f, 3f, 7f, 7f, ff, fe
2790 DATA fe, ff, 7e, 7c, 38, 30, 0, 0
2800 DATA c0, f0, fc, fc, fe, fe, ff, 7f
2810 DATA 7f, ff, 7e, 3e, 1c, c0, 0, 0
2820 DATA 0, 18, 25, 47, f, f, 9, 9
2830 DATA 1f, 1f, 1f, 3f, 3f, 7f, 67, 43
2840 DATA 0, 18, 64, e2, f0, f0, 90, 90
2850 DATA f8, f8, f8, fc, fc, fe, e6, c2
2860 DATA e0, 18, 5, 7, f, f, 9, 9
2870 DATA 1f, 1f, 1f, 3f, 3f, 7e, 3c, 18
2880 DATA 7, 18, a0, e0, f0, f0, 90, 90
2890 DATA f8, f8, f8, fc, fc, 7e, 3c, 18
2900 REM
2910 DATA ff, c3, c3, c3, c3, c3, c3, ff
2920 DATA 81, 0, 0, 0, 0, 0, 0, 81

20 Skippy



One Jump Ahead

'Once upon a time in Australia, which is a long way from Britain, there lived a little kangaroo. This little chap was called Skippy and he was very happy for many years.

'Skippy had his home by the banks of a fast flowing river on which logs floated. He used to enjoy sitting in his little hut looking out across the beautiful countryside, but one day things changed.

'The nasty humans came, and built a car expressway right next to the river. This meant that every day when Skippy and his friends returned from foraging for food, they had to cross the busy road, jump onto the logs to cross the river and then jump off as they passed their home. This is very dangerous and so they would like you, dear readers, to help.'

Using the joystick or keyboard guide Skippy and his friends safely between the cars, onto the logs and into their homes. Your assistance to wildlife will be rewarded with points in relation to how well you guide each kangaroo!

```

1000 REM >> SKIPPY          ANDY/BOOTSY <<
1010 REM
1020 SCREEN 1,0,0: CLEAR 1800
1030 COLOR 15,0,0: KEY OFF
1040 WIDTH 32: HS%=5: GOSUB 1680
1050 KA%=3: SC%=0
1060 GOSUB 1610: GOSUB 1550
1070 KX%=128: KY%=160: K%=1
1080 PUT SPRITE 0, (KX%, KY%), 15, K%
1090 REM ----- game loop -----
1100 GOSUB 1360: IF (K% AND 1)=0 THEN
GOSUB 1490: GOTO 1100
1110 CH%=VPEEK(6144+(KX%+4)/8+INT((KY
%+4)/8)*32)
1120 IF CH%<>32 AND CH%<>240 AND (CH%
<208 OR CH%>210) THEN 1180
1130 IF KY%=16 THEN 1300
1140 GOSUB 1320: IF ST% THEN GOSUB 148
0
1150 IF CH%>207 AND CH%<211 THEN KX%=
KX%+(8 AND KY%=32)-(8 AND KY%=48): GOT
O 1080
1160 GOTO 1100
1170 REM ----- lose life -----
1180 KA%=KA%-1: PLAY"o1164bagfedc"
1190 IF KA% THEN GOTO 1060
1200 PUT SPRITE 0, (0, 208), 0, 0
1210 LOCATE 2, 14
1220 PRINT " YOU BLEW IT YOU ROAD HOG
!"
1230 LOCATE 2, 16
1240 PRINT" trigger or space for anot
her "
1250 LOCATE 2, 18
1260 PRINT " attempt "
1270 GOSUB 1320: IF TG%=0 THEN 1270
1280 GOTO 1050
1290 REM ----- made it! -----
1300 PLAY"o1164cdefgabo2cdefgabo3cdef
gabo4cdefgab": SC%=SC%+1: GOTO 1060
1310 REM ----- input -----
1320 FOR I%=0 TO 2: ST%=STICK(I%): TG%=
STRIG(I%)
1330 IF ST% OR TG% THEN I%=2
1340 NEXT I%: RETURN

```

```

1350 REM ----- move chars -----
1360 L2#=RIGHT$(L2$,1)+LEFT$(L2$,31)
1370 L1#=RIGHT$(L1$,31)+LEFT$(L1$,1)
1380 C3#=RIGHT$(C3$,1)+LEFT$(C3$,31)
1390 C2#=RIGHT$(C2$,31)+LEFT$(C2$,1)
1400 C1#=RIGHT$(C1$,1)+LEFT$(C1$,31)
1410 LOCATE 0,4:PRINT L2#;
1420 LOCATE 0,6:PRINT L1#;
1430 LOCATE 0,14:PRINT C3#;
1440 LOCATE 0,16:PRINT C2#;
1450 LOCATE 0,18:PRINT C1#;
1460 RETURN
1470 REM ----- move kangaroo -----
1480 K%=ST%:M%=ST%
1490 K%=K% XOR 1
1500 KX%=KX%+(8 AND M%=3)-(8 AND M%=7
)
1510 KY%=KY%+(8 AND M%=5)-(8 AND M%=1
)
1520 PUT SPRITE 0,(KX%,KY%),15,K%
1530 RETURN
1540 REM ----- score etc. -----
1550 IF SC%>HS% THEN HS%=SC%
1560 LOCATE 7,0:PRINT SC%
1570 LOCATE 17,0:PRINT HS%
1580 LOCATE 28,0:PRINT KA%
1590 RETURN
1600 REM ----- draw screen -----
1610 LOCATE 0,1
1620 PRINT B1#;RV#;B2#
1630 LOCATE 0,20:PRINT B3#;B4#;
1640 LOCATE 1,0
1650 PRINT "SCORE:      HIGH:      'ROO
S:"
1660 GOTO 1410
1670 REM ----- def shapes -----
-
1680 C%=192
1690 READ I%:IF I%<0 THEN 1760
1700 FOR J%=0 TO 7:VPOKE C%*8+J%,I%
1710 READ I%:NEXT J%
1720 VPOKE 8192+C%/8,I%
1730 C%=C%+1
1740 IF C%=194 OR C%=203 OR C%=211 OR
C%=217 OR C%=226 OR C%=233 OR C%=241
THEN C%=(C%+7) AND 248

```

```

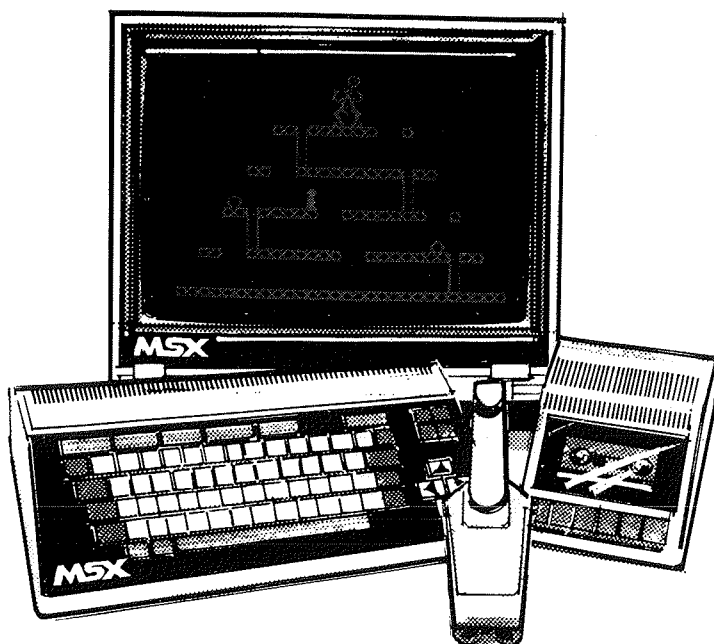
1750 GOTO 1690
1760 FOR I%=0 TO 7
1770 S$="":FOR J%=0 TO 7:READ C%
1780 S$=S$+CHR$(C%):NEXT J%
1790 SPRITE$(I%)=S$:NEXT I%
1800 CR$=CHR$(192)+CHR$(193)
1810 LY$=CHR$(202)+CHR$(201)+CHR$(200)
)
1820 BK$=CHR$(224)+CHR$(225)
1830 LG$=CHR$(208)+CHR$(209)+CHR$(210)
)
1840 C1$=CR$+CR$+SPACE$(8)+CR$+SPACE$(4)+CR$+CR$+CR$+SPACE$(2)+CR$+SPACE$(2)+CR$
1850 C2$=LY$+SPACE$(12)+LY$+SPACE$(3)+LY$+LY$+SPACE$(5)
1860 C3$=BK$+BK$+SPACE$(8)+BK$+BK$+BK$+BK$+BK$+SPACE$(8)+BK$
1870 L1$=LG$+STRING$(3,CHR$(248))+LG$+STRING$(3,CHR$(248))+STRING$(3,CHR$(216))+STRING$(2,CHR$(248))+LG$+STRING$(3,CHR$(248))+LG$+STRING$(2,CHR$(248))+STRING$(2,CHR$(216))+STRING$(2,CHR$(248))
1880 L2$=LG$+STRING$(3,CHR$(248))+LG$+STRING$(3,CHR$(248))+STRING$(3,CHR$(216))+STRING$(3,CHR$(248))+LG$+STRING$(3,CHR$(248))+STRING$(4,CHR$(216))+STRING$(2,CHR$(248))
1890 RV$=STRING$(160,CHR$(248))
1900 B1$=STRING$(32,CHR$(232))
1910 FOR I%=1 TO 16 STEP 4
1920 B1$=B1$+STRING$(4,CHR$(232))+SPACE$(1)+STRING$(3,CHR$(232))
1930 NEXT I%
1940 BN$=LEFT$(BN$,63):BN$=BN$+CHR$(232)
)
1950 B2$=STRING$(160,CHR$(240))
1960 B3$=STRING$(64,CHR$(240))
1970 B4$=STRING$(32,CHR$(232))
1980 RETURN
1990 REM ----- shape defs -----
2000 DATA 7,28,56,127,219,231,36,24,208
)
2010 DATA 128,64,32,252,218,229,38,24,208
)

```


2020 DATA 0,0,0,0,255,255,60,24,96
 2030 DATA 64,64,64,64,255,255,128,0,9
 6
 2040 DATA 7,9,9,63,127,127,7,3,96
 2050 DATA 191,109,182,109,218,125,171
 ,95,196
 2060 DATA 255,181,235,182,47,254,93,1
 90,196
 2070 DATA 250,213,190,91,182,109,182,
 253,196
 2080 DATA 0,0,67,172,240,192,255,0,19
 7
 2090 DATA 0,1,51,14,55,72,72,48,176
 2100 DATA 96,200,244,12,154,164,228,2
 4,176
 2110 DATA 255,255,255,255,255,255,255
 ,255,192
 2120 DATA 0,127,127,127,127,127,127,1
 27,160
 2130 DATA 170,85,170,85,170,85,170,85
 ,84
 2140 DATA -1
 2150 REM ----- sprite info -----
 2160 DATA 36,24,90,60,24,60,60,219
 2170 DATA 36,24,24,60,90,189,126,36
 2180 DATA 8,24,28,24,60,61,58,220
 2190 DATA 4,12,14,24,60,188,184,94
 2200 DATA 219,60,60,24,60,90,24,36
 2210 DATA 36,126,189,90,60,24,24,36
 2220 DATA 16,24,56,24,60,188,92,59
 2230 DATA 32,48,112,24,60,61,29,122

21

Kinkey Dong



All Action Climax

This, the third arcade style game, is loosely inspired by a group of games in which the hero must jump over obstacles as he climbs up the ladder to get higher and higher through the screens.

The evil monster that appears on each screen has in fact abducted the hero's girlfriend and locked her up. He now spends all of his time rolling large barrels down to crush the hero.

There are three different screen layouts and if you are successful then the game continues through from the first screen to give higher scores.

Movement is by the now familiar cursor keys or your old friend the joystick. The little hero can be made to jump by pressing the space bar or the fire button.

```

1000 REM>> KINKY DONG      BOOTSY <<
1010 REM
1020 SCREEN 1,2,0:COLOR 15,0,0:CLS
1030 KEY OFF:GOSUB 2060:HS%=1000
1040 SC%=0:LF%=3:PS%=0
1050 XP%=20:YP%=160
1060 GOSUB 1830:SB%=4:SM%=0
1070 IF YP%=32 THEN GOSUB 1780:SC%=(S
C%+1) AND 3:GOTO 1050
1080 GOSUB 1390:IF C% THEN 1190
1090 X%=XP%:Y%=YP%
1100 IF TG% THEN GOSUB 1540:GOTO 1190
1110 IF ST%=1 THEN GOSUB 1490 ELSE GO
SUB 1480
1120 IF ST%=1 AND C%=0 AND CH%=240 TH
EN UD%=-2:GOSUB 1590:GOTO 1190
1130 IF ST%=5 AND C%=0 AND CH%=240 TH
EN UD%=2:GOSUB 1590:GOTO 1190
1140 IF ST%=3 AND CH%=32 THEN 1630
1150 IF ST%=3 THEN XP%=XP%+4:D%=2:IF
SM% AND 2 THEN SM%=0
1160 IF ST%=7 AND C%=32 THEN 1630
1170 IF ST%=7 THEN XP%=XP%-4:D%=-2:IF
(SM% AND 2)=0 THEN SM%=2
1180 SM%=SM% XOR 1:PUT SPRITE 0,(XP%,
YP%),15,SM%
1190 L%=0:FOR I%=0 TO 3:J%=XP%-BA%(I%
,0):K%=YP%-BA%(I%,1)
1200 IF ABS(J%)<12 AND ABS(K%)<12 THE
N L%=1:I%=3
1210 NEXT I%:IF L% THEN 1630
1220 IF B%<3 THEN IF RND(-TIME)>.95 T
HEN B%=B%+1:BA%(B%,0)=128:BA%(B%,1)=3
2
1230 IF B%=0 THEN 1070
1240 SB%=SB% XOR 1:I%=1
1250 J%=BA%(I%,0):K%=BA%(I%,1)
1260 IF K%=32 OR K%=96 OR K%=160 THEN
J%=J%-4 ELSE J%=J%+4
1270 PUT SPRITE I%,(J%,K%),12,SB%
1280 X%=J%:Y%=K%:GOSUB 1480
1290 IF C%=0 AND CH%=32 THEN GOSUB 17
40:GOTO 1310
1300 IF C%=0 AND CH%=240 THEN IF RND(
-TIME)>.66 THEN GOSUB 1740

```

```

1310 IF J%=16 AND K%=160 THEN 1350
1320 BA%(I%,0)=J%:BA%(I%,1)=K%
1330 I%=I%+1:IF I%>B% THEN 1070
1340 GOTO 1250
1350 FOR L%=I% TO B%:J%=BA%(L%+1,0):B
A%(L%,0)=J%:K%=BA%(L%+1,1):BA%(L%,1)=
K%
1360 PUT SPRITE L%,(J%,K%),12,SB%:NEX
T L%
1370 B%=B%-1:I%=I%-1:GOTO 1330
1380 REM ----- input -----
1390 C%=1:FOR I%=0 TO 2:ST%=STICK(I%)
:TG%=STRIG(I%)
1400 IF ST% OR TG% THEN I%=2:C%=0
1410 NEXT I%:IF ST%=5 THEN 1390
1420 RETURN
1430 A=USR1(0):I$=INPUT$(1)
1440 RETURN
1450 REM ----- xychar -----
1460 CH%=VPEEK(&H1800+INT(Y%/8)*32+X%
/8):RETURN
1470 REM ---- on/over ladder -----
1480 Y%=Y%+8
1490 X%=X%+4:Y%=Y%+8:GOSUB 1460:C%=CH
%
1500 X%=X%+7:GOSUB 1460
1510 IF C%<>CH% THEN RETURN
1520 C%=0:RETURN
1530 REM ----- jump -----
1540 RESTORE 2560:FOR I%=1 TO 12
1550 READ VO%:XP%=XP%+D%:YP%=YP%+VO%
1560 PUT SPRITE 0,(XP%,YP%),15,0
1570 NEXT I%:RETURN
1580 REM ----- climb ladder -----
1590 FOR I%=1 TO 16:YP%=YP%+UD%
1600 PUT SPRITE 0,(XP%,YP%),15,0
1610 NEXT I%:RETURN
1620 REM ----- fall -----
1630 PLAY"L6402BAGFEDCO1BAGFEDC"
1640 FOR I%=YP% TO 192 STEP 2
1650 PUT SPRITE 0,(XP%,I%),15,SP%
1660 NEXT I%
1670 REM ----- lose life -----
1680 LF%=LF%-1:IF LF% THEN 1050
1690 LOCATE 7,12:PRINT " TOUGH MONKEY
S !"

```

```

1700 PRINT " press trigger for anothe
r game"
1710 GOSUB 1390:IF TG%=0 THEN 1710
1720 GOTO 1040
1730 REM ---- barrel drop -----
1740 FOR L%=1 TO 16:K%=K%+2
1750 PUT SPRITE I%,(J%,K%),12,SB%
1760 NEXT L%:RETURN
1770 REM ----- score -----
1780 PS%=PS%+500
1790 IF PS%>HS% THEN HS%=PS%
1800 PLAY "L6404CDEFGAB05CDEFGAB06CDE
FGAB"
1810 RETURN
1820 REM ----- draw screen -----
1830 CLS:FOR I%=0 TO 4
1840 LOCATE 8-I%*2,I%*4+6
1850 PRINT STRING$(I%*4+12,CHR$(248))
1860 NEXT I%
1870 SL%=10-SC%*2:SH%=SC%*2
1880 RESTORE 2540:FOR I%=1 TO SL%
1890 READ LX%,LY%:FOR J%=1 TO 4
1900 LOCATE LX%,LY%:PRINT CHR$(240)
1910 LY%=LY%+1:NEXT J%,I%
1920 IF SH%=0 THEN 1960
1930 FOR I%=1 TO SH%
1940 READ LX%,LY%:LOCATE LX%,LY%
1950 PRINT SPC(2):NEXT I%
1960 PUT SPRITE 0,(XP%,YP%),15,0
1970 C%=224:FOR I%=2 TO 5:FOR J%=13 T
O 15
1980 LOCATE J%,I%:PRINT CHR$(C%);:C%=
C%+1
1990 NEXT J%,I%
2000 LOCATE 1,1:PRINT "SCORE:";HS%
2010 LOCATE 14,1:PRINT "HIGH:";PS%
2020 LOCATE 26,1:PRINT "MEN:";LF%
2030 FOR I%=0 TO 4:BA%(I%,0)=108:NEXT
I%
2040 RETURN
2050 REM ----- def shapes -----
2060 RESTORE
2070 FOR C%=224 TO 235
2080 GOSUB 2160
2090 NEXT C%

```

```

2100 C%=240:GOSUB 2160
2110 C%=248:GOSUB 2160
2120 FOR S%=0 TO 5
2130 GOSUB 2200
2140 NEXT S%
2150 RETURN
2160 CA%=C%*8:FOR I%=CA% TO CA%+7
2170 READ J%:VPOKE I%,J%:NEXT I%
2180 READ J%:VPOKE C%/8+8192,J%
2190 RETURN
2200 S$="":FOR I%=0 TO 31
2210 READ J%:S$=S$+CHR$(J%):NEXT I%
2220 SPRITE$(S%)=S$:RETURN
2230 REM ----- kinky dong -----
2240 DATA 0,0,16,44,62,28,56,112,96
2250 DATA 129,66,126,90,126,106,86,60
,96
2260 DATA 0,0,8,52,124,56,28,14,96
2270 DATA 112,121,127,63,31,7,1,1,96
2280 DATA 60,255,255,255,231,215,235,
213,96
2290 DATA 14,158,254,252,248,224,128,
128,96
2300 DATA 1,3,15,15,31,63,62,62,96
2310 DATA 171,213,235,215,235,255,60,
0,96
2320 DATA 128,192,240,240,248,252,124
,124,96
2330 DATA 30,15,15,3,59,127,127,57,96
2340 DATA 0,0,0,0,129,195,195,129,96
2350 DATA 120,240,240,192,220,254,254
,156,96
2360 REM ----- backdrop -----
2370 DATA 129,255,129,129,129,255,129
,129,128
2380 DATA 255,36,66,129,129,66,36,255
,160
2390 REM ----- man -----
2400 DATA 3,7,7,15,7,3,7,15,31,55,7,7
,14,29,25,12
2410 DATA 128,192,224,128,192,128,192
,224,240,176,128,128,128,128,192,224
2420 DATA 3,7,7,15,7,3,7,6,7,15,15,7,
7,7,13,14
2430 DATA 128,192,224,128,192,128,128

```

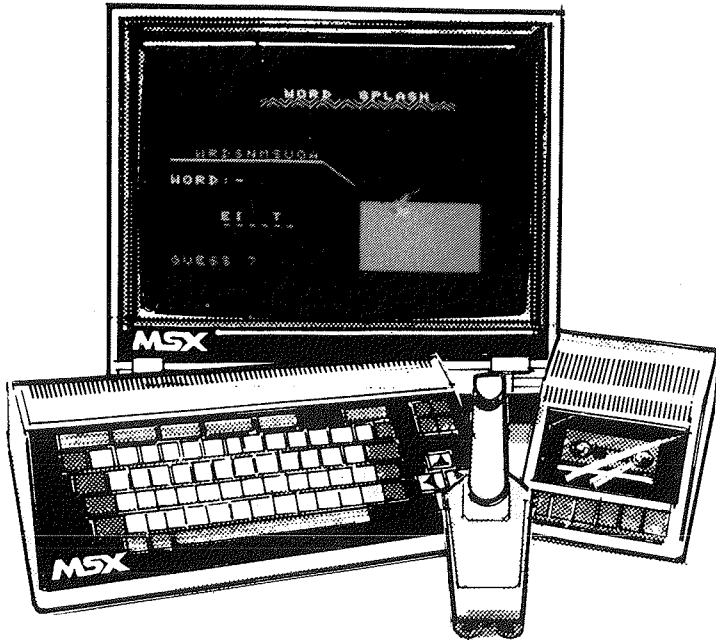
```

,128,0,192,224,128,128,128,192,224
2440 DATA 1,3,7,1,3,1,3,7,15,13,1,1,1
,1,3,7
2450 DATA 192,224,224,240,224,192,192
,240,248,236,224,224,112,184,152,48
2460 DATA 1,3,7,1,3,1,1,1,0,3,7,1,1,1
,3,7
2470 DATA 192,224,224,240,224,192,224
,96,224,240,240,224,224,224,176,112
2480 REM ----- barrel data -----
2490 DATA 0,0,0,0,0,0,3,7,12,24,24,25
,26,12,7,3
2500 DATA 0,0,0,0,0,0,192,224,48,88,1
52,24,24,48,224,192
2510 DATA 0,0,0,0,0,0,3,7,12,26,25,24
,24,12,7,3
2520 DATA 0,0,0,0,0,0,192,224,48,24,2
4,152,88,48,224,192
2530 REM ---- hole/ladder info ----
2540 DATA 10,6,19,10,6,14,23,18,4,18,
21,14,8,10,17,6,14,18,12,14
2550 REM ----- jump info -----
2560 DATA -5,-4,-3,-2,-1,0,0,1,2,3,4,
5
2570 A$=INPUT$(1):RETURN

```

22

Word Splash



Spell or Swim

This game is a more humane, more fun version of the well known hangman game. If the player cannot guess the word in time, then all the wrong letters push a character into the water.

Words can be up to eight letters long but should not contain hyphens. The listing has about 60 words already included. More words may be added or the existing ones changed.

If a letter is guessed, it appears at each correct position in the word. If it is not right then the poor victim is pushed one space along. Should the word be completed in time then the little figure pushes all the wrong letters back and a new mystery word is offered.

The word is not shown if it is not guessed and since words are randomly selected it may be possible to have the same word again later.

The program is very entertaining, educational and can be surprisingly addictive!


```

1000 REM <<<<< WORD SPLASH >> ISSI >
1010 REM
1020 MAXFILES=2:OPEN "GRP:S" FOR OUTP
UT AS 1
1030 GOSUB 1520
1040 W=INT(RND(-TIME)*60)+1
1050 W#=D$(W)
1060 GOSUB 1320
1070 GOSUB 1750
1080 FLAG=0
1090 IF T#=W# THEN GOTO 1180
1100 IF LEN(G#)=10 THEN GOTO 1140
1110 GOTO 1070
1120 REM <<<<<<<< LOSE >>>>>>>>>
1130 REM
1140 GOSUB 1580
1150 GOTO 1280
1160 REM <<<<<<<<< WIN >>>>>>>>>
1170 REM
1180 OCT=8
1190 IF G#="" THEN GOTO 1260
1200 FOR N=LEN(G#) TO 1 STEP -1
1210 G#=RIGHT$(G#,N-1)
1220 M#="S11L640"+STR$(OCT)+"C":PLAY
M#
1230 GOSUB 1840:PRINT " "
1240 FOR W=1 TO 250:NEXT W
1250 NEXT N
1260 PRESET (48,16):PRINT #1," CONGRA
TULATIONS !"
1270 PRESET (48,24):PRINT #1,"-----
-----"
1280 A#=INKEY$:IF A#="" THEN 1280
1290 BEEP:GOTO 1040
1300 REM >>>>>>>>> SCREEN <<<<<<<<<<
1310 REM
1320 SCREEN 2,0,0:KEY OFF
1330 COLOR 15,1,1:CLS
1340 PRESET (80,0):PRINT #1,"WORD SP
LASH":COLOR 3
1350 A#="" :FOR N=1 TO 8:A#=A#+CHR$(&H
CB)+CHR$(&HCC):NEXT N
1360 PRESET (64,8):PRINT #1,A#
1370 PRESET (24,56):COLOR 13:O#=CHR$(
1)+CHR$(&HCC):PRINT #1,O#

```

```

1380 LINE (128,100)-(208,160),4,BF
1390 LINE (120,88)-(128,100),4
1400 LINE (216,88)-(208,100),4
1410 LINE (0,65)-(104,65),14
1420 LINE -STEP(20,20),14
1430 COLOR 11:PRESET (8,80):PRINT #1,
"WORD:-"
1440 PRESET (40,120):T$="":G$=""
1450 COLOR 5:FOR N=1 TO LEN(W$)
1460 PRINT #1,"-";:T$=T$+" "
1470 NEXT N
1480 PRESET (8,152):PRINT #1,"GUESS ?
"
1490 RETURN
1500 REM >>>>>>> INITIALIZE <<<<<<<<
1510 REM
1520 RESTORE 1910
1530 DIM D$(60)
1540 FOR N=1 TO 60:READ D$(N):NEXT N
1550 RETURN
1560 REM >>>>>>>> SPLASH ! <<<<<<<<
1570 REM
1580 FOR N=106 TO 152 STEP 2
1590 COLOR 1:PRESET (N-2,N-50)
1600 PRINT #1,CHR$(1)+CHR$(&HCB)
1610 COLOR 9:PRESET (N,N-48)
1620 PRINT #1,CHR$(1)+CHR$(&HCB);
1630 NEXT N
1640 RESTORE 1720
1650 FOR N=6 TO 13:READ A:SOUND N,A:N
EXTN
1660 LINE -STEP(-8,0),4
1670 FOR I=1 TO 15:J=-RND(-TIME)*20:K
=RND(-TIME)*40-20
1680 LINE -STEP(K,J),4:LINE -STEP(-K,
-J),4
1690 NEXT I
1700 FOR W=1 TO 500:NEXT W
1710 RETURN
1720 DATA 15,135,16,16,16,0,16,0
1730 REM >>>>>>>>> INKEYS <<<<<<<<<<
1740 REM
1750 A$=INKEY$:IF A$="" THEN GOTO 175
0
1760 A$=CHR$(ASC(A$) AND 223):PLAY "O
4S11L16C"

```

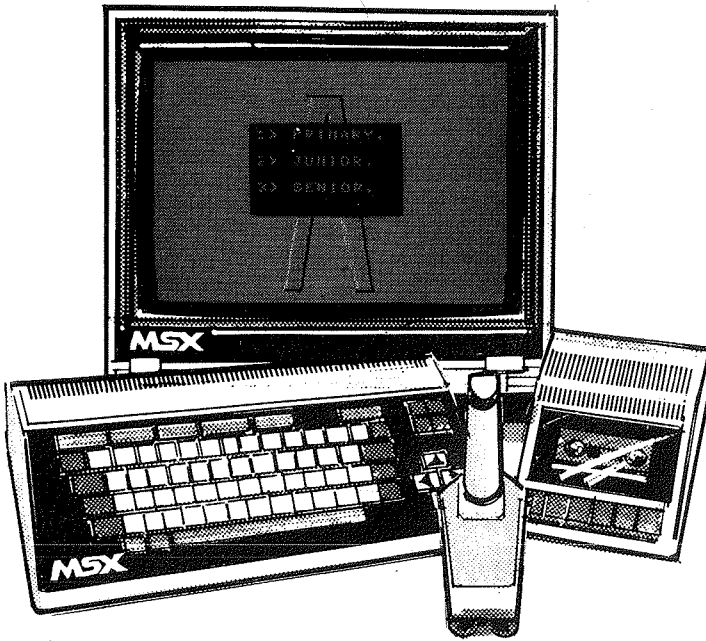
```

1770 IF A$<"A" OR A$>"Z" THEN GOTO 17
50
1780 FLAG=0:COLOR 9
1790 FOR N=1 TO LEN(W$)
1800 IF MID$(W$,N,1)=A$ THEN PRESET (
(N+4)*8,112):PRINT #1,A$:MID$(T$,N,1)
=A$:FLAG=1
1810 NEXT N
1820 IF FLAG=1 THEN PLAY "GAB":RETURN
1830 G$=A$+G$
1840 PRESET (24,56):COLOR 1:PRINT #1,
O$
1850 PRESET (24,56):COLOR 13
1860 IF LEN(G$)=INT(LEN(G$)/2) THEN
M=&HCB ELSE M=&HCC
1870 O$=G$+CHR$(1)+CHR$(M):PRINT #1,O
$;
1880 RETURN
1890 REM >>>>>>>>>> DATA <<<<<<<<<<<<
1900 REM
1910 DATA ACUTE,AGILE,BROKE,CANDY,COU
GH,DRYLY,GNOME,MISTY,NYMPH,PYGMY
1920 DATA BARLEY,BODKIN,CEPTIC,DEVOUR
,EIGHTY,EXOTIC,FLINCH,GLORIA,HYMN,LYR
ICS
1930 DATA METHYL,SPYING,SYRUP,YOGHURT
,VISUAL,WOBBLE,ANCHOVY,RELAX,RHYTHM,R
HUBARB
1940 DATA SKETCH,FASCHIST,BORING,FATI
GUE,FROWN,TRUANT,LEMMING,ZINC,GLAMOUR
,HYDRATED
1950 DATA KETCHUP,MYSTIC,NOISY,OCTOPU
S,SQUID,RAMBLER,SATCHEL,SEXTANT,YULET
IDE,SWINDLE
1960 DATA PLAYER,DOCKER,DESTINY,FUTUR
E,LENTIL,FOREIGN,LOGICAL,WINTER,THIRS
T,NAIVE

```

23

Sum Fun



Ten out of Ten

This single program can offer questions for 'primary', 'junior' and 'senior' levels (that is everyone). The question can be of any one type, selected from plus, minus, multiply and divide. The questions and answers are drawn on a blackboard to help make the presentation attractive.

Selecting the primary level will set up problems which involve counting two sets of 'little men' to produce a total. The junior level provides options for the four types of arithmetic and produces questions which have whole number answers. It should be possible for most adults and older children to work these out mentally.

The senior level would require a very strong mental arithmetic ability and is designed to set problems which can be worked out using pencil and paper or even a calculator.

The answers are required to be typed in and then sent to the computer using the 'Enter' key. This means that mistyped answers can be corrected if spotted before the 'Enter' key is pressed. The

program will allow up to two wrong tries before revealing the correct answer.

At the end of ten questions the computer will print up a 'report' showing the number of correct answers and a percentage score.

```

1000 REM <<<<<< SUM FUN * ISSI >>>>
1010 REM
1020 SCREEN 2,0,0:KEY OFF
1030 MAXFILES=2:OPEN "GRP:S" FOR OUTP
UT AS 1
1040 GOSUB 1130
1050 COLOR 3:PRESET (75,55):PRINT #1,
"1> PRIMARY."
1060 COLOR 9:PRESET (75,75):PRINT #1,
"2> JUNIOR."
1070 COLOR 11:PRESET (75,95):PRINT #1
,"3> SENIOR."
1080 A$=INPUT$(1):A=ASC(A$)
1090 IF A<49 OR A>51 THEN GOTO 1080
1100 PLAY "L64CDECDE"
1110 LEV=A-48
1120 GOTO 1880
1130 COLOR 15,4,1:CLS
1140 COLOR 9:LINE (120,40)-(140,180)
1150 LINE (140,180)-(150,180)
:LINE (150,180)-(130,30)
1160 LINE (130,30)-(110,30):LINE (110
,30)-(90,180)
1170 LINE (90,180)-(100,180):LINE (10
0,180)-(120,40)
1180 PAINT (120,35),9,15
1190 LINE (70,50)-(170,120),1,BF
1200 COLOR ,1
1210 RETURN
1220 REM >>>>>>> INPUT <<<<<<<<<
1230 G$=""
1240 PRESET (75,100):COLOR 15
1250 A$=INKEY$:IF A$="" THEN GOTO 125
0
1260 PLAY "O4L16B"
1270 A=ASC(A$):IF (A>47 AND A<58) OR
A=45 THEN PRINT #1,A$;:G$=G$+A$:GOTO
1250
1280 IF A=13 THEN GOTO 1340
1290 IF A<>8 OR G$="" THEN GOTO 1250

```

```

1300 LINE (70,90)-(170,120),1,BF
1310 G#=LEFT$(G$,LEN(G$)-1)
1320 PRESET (75,100):PRINT #1,G#;
1330 GOTO 1250
1340 IF VAL(G#)=ANS THEN CO=1 ELSE CO
=0
1350 RETURN
1360 REM >>>>>>>>> GUESS <<<<<<<<<<<
1370 REM
1380 FOR G=1 TO 2
1390 GOSUB 1230
1400 IF CO=1 THEN GOTO 1460
1410 PLAY "L802C01C"
1420 LINE (70,90)-(170,120),1,BF
1430 NEXT G
1440 PLAY "L1602FEDFEDFEDC"
1450 GOTO 1480
1460 PLAY "L1604CDECDECDEA"
1470 G=2:TT=TT+1
1480 LINE (70,50)-(170,120),1,BF:RETU
RN
1490 REM >>>>>>>>> TYPE <<<<<<<<<<<
1500 REM
1510 LINE (70,50)-(170,120),1,BF
1520 COLOR 15:PRESET (75,55):PRINT #1
,"1> '+'"
1530 COLOR 9:PRESET (75,70):PRINT #1,
"2> '-'"
1540 COLOR 3:PRESET (75,85):PRINT #1,
"3> 'x'"
1550 COLOR 11:PRESET (75,100):PRINT #
1,"4> '/'"
1560 A#=INPUT$(1):A=ASC(A$)-48
1570 IF A<1 OR A>4 THEN PLAY "L801C":
GOTO 1560
1580 DEF FN SUM=A+B:S$="+"
1590 IF A=2 THEN DEF FN SUM=A-B:S$="-"
"
1600 IF A=3 THEN DEF FN SUM=A*B:S$="x"
"
1610 IF A=4 THEN DEF FN SUM=A/B:S$="/
":DIV=1
1620 RETURN
1630 REM >>>>>>>>>> INIT <<<<<<<<<<<
1640 LIM=8:DIV=0

```

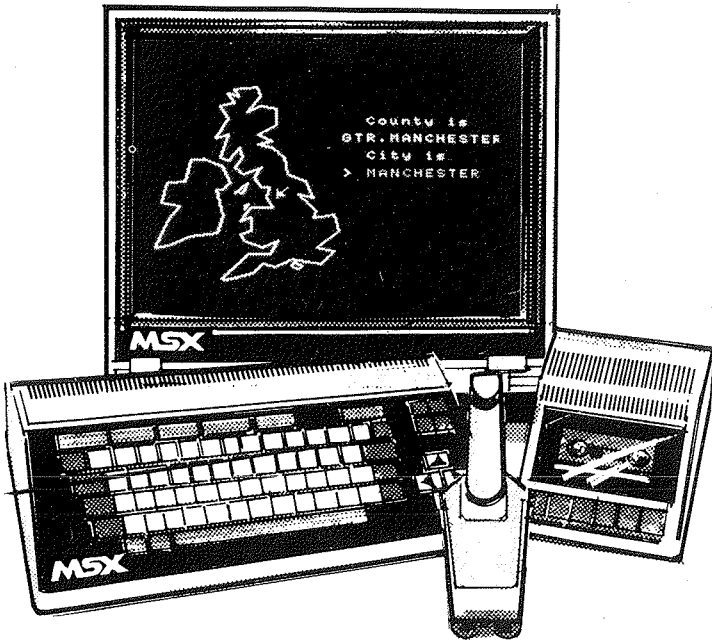
```

1650 IF LEV=2 THEN LIM=12:GOSUB 1510
1660 IF LEV=3 THEN LIM=25:GOSUB 1510
1670 LINE (70,50)-(170,120),1,BF
1680 TT=0:RETURN
1690 REM >>>>>> NUMBERS <<<<<<<<
1700 REM
1710 A=INT(RND(-TIME)*LIM)+1
1720 B=INT(RND(1)*LIM)+1
1730 IF DIV=1 THEN A=A*B
1740 IF LEV=2 THEN DIV=0
1750 LINE (70,50)-(170,120),1,BF
1760 COLOR 15:PRESET (72,52)
1770 IF LEV=1 THEN GOTO 1810
1780 PRINT #1,A;S#;B
1790 ANS=FN SUM
1800 RETURN
1810 FOR N=1 TO A:PRINT #1,CHR$(1)+CHR$(66);:NEXT N:PRINT #1," +"
1820 PRESET (72,70)
1830 FOR N=1 TO B:PRINT #1,CHR$(1)+CHR$(65);:NEXT N:PRINT #1," ="
1840 ANS=A+B
1850 RETURN
1860 REM >>>>>> MAIN LOOP <<<<<<<<
1870 REM
1880 GOSUB 1640
1890 FOR P=1 TO 10
1900 GOSUB 1710:GOSUB 1380
1910 NEXT P
1920 LINE (70,50)-(170,120),1,BF
1930 COLOR 15
1940 PRESET (75,64)
1950 PRINT #1,TT;" CORRECT"
1960 PRESET (75,88)
1970 PRINT #1,"THAT'S";TT*100/10;"%"
1980 PRESET (75,112)
1990 PRINT #1," PRESS 'Y'."
2000 A$=INPUT$(1):IF A$<>"Y" AND A$<>"y" THEN GOTO 2000
2010 LINE (70,50)-(170,120),1,BF
2020 GOTO 1050

```

24

Today England



Tomorrow the World

This is a self-contained geographic quiz based upon the map of Great Britain. The game is very straightforward and consists of questions which relate to key towns or cities, and the county to which they belong. Options are given to guess the county, the town or both. A point is plotted on the map to indicate the location of the town or city.

The central core of the program could be developed to produce either a more detailed map or as the heading suggests, the world could be plotted next!

Another way to develop this program would be to extend the data to include more or all counties and the county towns. In fact you could find that the research of the facts to put into the listing is even more educational than the program itself!


```

1000 REM << TODAY ENGLAND >> ISSI >>
1010 REM
1020 MAXFILES=2:OPEN "GRP:S"FOR OUTPUT AS 1
1030 COLOR 15,1,1:SCREEN 1,0,0:KEY OFF
1040 SC=0:LOCATE 8,0:PRINT "Today England"
1050 LOCATE 5,2:PRINT "Tomorrow the World."
1060 COLOR 7
1070 LOCATE 1,5:PRINT "Choose Option:"
1080 LOCATE 6,7:PRINT "1....Guess Counties"
1090 LOCATE 6,9:PRINT "2....Guess Cities"
1100 LOCATE 6,11:PRINT "3....Guess Both."
1110 A$=INKEY$:IF A$<"1" OR A$>"3" THEN GOTO 1110
1120 OP=VAL(A$)
1130 GOSUB 1800
1140 RESTORE 2470
1150 IF OP=3 THEN GOTO 1490
1160 REM >>>> GUESS CITY/COUNTY <<<
1170 FOR C=1 TO 16
1180 READ N$,X,Y,CO$
1190 PSET (X/2.5,192-Y/2)
1200 PUTSPRITE 1,(X/2.5,182-Y/2),9,1
1210 PRESET (164,32):COLOR 7
1220 IF OP=1 THEN PRINT #1,"City is":PRESET (148,48):PRINT #1,N$
1230 IF OP=2 THEN PRINT #1,"County is":PRESET (148,48):PRINT #1,CO$
1240 TR=3
1250 PRESET (164,64):IF OP=1 THEN PRINT #1,"County is " ELSE PRINT #1,"City is "
1260 GOSUB 2170:IF OP=1 THEN C$=I$ ELSE CI$=I$
1270 IF OP=1 AND C$=CO$ THEN GOTO 1340
1280 IF OP=2 AND CI$=N$ THEN GOTO 1340

```

```

1290 TR=TR-1:PRESET (164,96):COLOR 9:
PRINT #1,"WRONG !";TR:PRESET (164,104
):PRINT #1,"TRIES LEFT."
1300 PLAY "EDC":FOR W=1 TO 500:NEXT W
1310 LINE (164,64)-(255,160),1,BF
1320 IF TR<1 THEN GOTO 2080
1330 GOTO 1250
1340 GOSUB 2030
1350 PRESET (164,96):COLOR 9:PRINT #1
,"CORRECT !"
1360 PLAY "EFG":FOR W=1 TO 500:NEXT W
1370 SC=SC+TR
1380 PSET (X/2.5,192-Y/2),1:GOSUB 203
0
1390 NEXT C
1400 SCREEN 1,0,0
1410 COLOR 15,1,1:CLS
1420 LOCATE 5,5:PRINT "You scored ";S
C;"points."
1430 LOCATE 6,10:PRINT "Press 'P' to
replay."
1440 A$=INPUT$(1):A$=CHR$(ASC(A$) AND
223)
1450 IF A$<>"P" THEN GOTO 1440
1460 GOTO 1030
1470 REM >>> GUESS CITY & COUNTY <<
1480 REM
1490 FOR C=1 TO 16
1500 READ N$,X,Y,C$:COLOR 7
1510 PSET (X/2.5,192-Y/2)
1520 PUTSPRITE 1,(X/2.5,182-Y/2),9,1
1530 PRESET (164,32):PRINT #1,"City i
s"
1540 TR=3
1550 GOSUB 2170:CI$=I$
1560 IF CI$=N$ THEN GOSUB 2030:PRESET
(164,80):PRINT #1,"Correct":GOSUB 20
50:SC=SC+TR:GOSUB 2030:GOTO 1640
1570 TR=TR-1
1580 COLOR 6,1:PRESET (164,90):PRINT
#1,"Wrong";TR:PRESET (164,100):PRINT
#1,"tries left"
1590 GOSUB 2050:PRESET (164,32)
1600 LINE (164,64)-(255,110),1,BF
1610 IF TR<>0 THEN GOTO 1550

```

```

1620 GOSUB 2030:PRESET (164,32):PRINT
#1,"CITY IS"
1630 PRESET (164,40):PRINT #1,N$
1640 PRESET (164,48):PRINT #1,"County
is"
1650 TR=3
1660 GOSUB 2170:CO#=I$
1670 IF CO#=C$ THEN GOSUB 2030:PRESET
(164,80):PRINT #1,"Correct":GOSUB 20
50:SC=SC+TR:GOSUB 2030:GOTO 1750
1680 TR=TR-1
1690 COLOR 6,1:PRESET (164,90):PRINT
#1,"Wrong":TR:PRESET (164,100):PRINT
#1,"tries left"
1700 GOSUB 2050
1710 LINE (164,64)-(255,110),1,BF
1720 IF TR<>0 THEN GOTO 1660
1730 GOSUB 2030:PRESET (164,64):PRINT
#1,"COUNTY IS"
1740 PRESET (148,74)
1750 GOSUB 2050
1760 GOSUB 2030
1770 PSET (X/2.5,192-Y/2),1
1780 NEXT C:GOTO 1400
1790 REM >>>>>>>>> DRAW MAP <<<<<<<
1800 COLOR 15,1,1:SCREEN 2,0,0
1810 RESTORE 2640:S$="":FOR N=0 TO 7:
READ A:S$=S$+CHR$(A):NEXT N:SPRITE$(1
)=S$
1820 RESTORE
1830 READ X,Y:Y=192-Y/2:X=X/2.5:PSET
(X,Y)
1840 FOR N=1 TO 58
1850 READ X,Y:Y=192-INT(Y/2):X=INT(X/
2.5):L$="M"+STR$(X)+", "+STR$(Y):DRAW
L$
1860 NEXT N
1870 READ X,Y:Y=192-Y/2:X=X/2.5:PSET
(X,Y)
1880 FOR N=1 TO 16
1890 READ X,Y:Y=192-INT(Y/2):X=INT(X/
2.5):L$="M"+STR$(X)+", "+STR$(Y):DRAW
L$
1900 NEXT N
1910 READ X,Y:Y=192-Y/2:X=X/2.5:PSET
(X,Y)

```

```

1920 FOR N=1 TO 3
1930 READ X,Y:Y=192-INT(Y/2):X=INT(X/
2.5):L$="M"+STR$(X)+", "+STR$(Y):DRAW
L$
1940 NEXT N
1950 READ X,Y:Y=192-Y/2:X=X/2.5:PSET
(X,Y)
1960 FOR N=1 TO 3
1970 READ X,Y:Y=192-INT(Y/2):X=INT(X/
2.5):L$="M"+STR$(X)+", "+STR$(Y):DRAW
L$
1980 NEXT N
1990 READ X,Y:Y=192-Y/2:X=X/2.5:PSET
(X,Y)
2000 FOR N=1 TO 4
2010 READ X,Y:Y=192-INT(Y/2):X=INT(X/
2.5):L$="M"+STR$(X)+", "+STR$(Y):DRAW
L$
2020 NEXT N
2030 LINE (148,0)-(255,191),1,BF
2040 RETURN
2050 PLAY "L8CEGFEDC"
2060 FOR T=1 TO 1200:NEXT T
2070 RETURN
2080 GOSUB 2030
2090 PRESET (148,24):COLOR 9
2100 PRINT #1,"City is"
2110 PRESET (148,40):PRINT #1,N$
2120 PRESET (148,56):PRINT #1,"County
is"
2130 PRESET (148,72):PRINT #1,C0$
2140 FOR W=1 TO 500:NEXT W
2150 GOSUB 2030
2160 GOTO 1380
2170 COLOR 5
2180 I$="":PRESET (148,80):PRINT #1,"
>"
2190 A$=INKEY$:IF A$="" THEN GOTO 219
0
2200 BEEP:IF A$=CHR$(13) THEN RETURN
2210 X$="":IF LEN(I$)<12 THEN W$=I$ E
LSE W$=LEFT$(I$,11):X$=RIGHT$(I$,LEN(
I$)-11)
2220 COLOR 1:PRESET (164,80):PRINT #1
,W$

```

```

2230 IF X$<>" THEN PRESET (164,88):P
RINT #1,X$
2240 IF A$=CHR$(8) AND I$>" THEN I$=
LEFT$(I$,LEN(I$)-1):GOTO 2280
2250 IF A$=" " OR A$="." THEN GOTO 22
70
2260 IF A$<"A" OR A$>"Z" THEN GOTO 22
80
2270 I$=I$+A$
2280 COLOR 5
2290 X$="":IF LEN(I$)<12 THEN W$=I$ E
LSE W$=LEFT$(I$,11):X$=RIGHT$(I$,LEN(
I$)-11)
2300 PRESET (164,80):PRINT #1,W$
2310 IF X$<>" THEN PRESET (164,88):P
RINT #1,X$
2320 GOTO 2190
2330 REM <<<<< MAINLAND >>>>>>>>>>>>>>>>
2340 DATA 192,371,228,377,234,359,210
,329,234,341,258,335,258,305,228,287,
252,275
2350 DATA 228,269,252,263,270,245,276
,215,306,203,318,173,294,179,324,149,
318,137,354,143,360,113
2360 DATA 324,83,354,77,330,53,282,61
,253,47,234,53,216,35,174,23,165,24,2
10,71,246,71,258,95,234,83
2370 DATA 204,95,198,107,216,113,216,
137,198,131,216,155,252,155,240,173,2
46,191,234,191,222,209
2380 DATA 234,227,204,215,192,227,204
,245,198,263,180,239,168,245,180,287,
162,281,180,317,168,341,192,347,180,3
65,198,359,192,371
2390 REM * * * IRELAND * * *
2400 DATA 142,241,172,229,184,199,172
,187,160,187,172,139,160,109,118,103,
82,85,58,80,80,130,100,157,70,163,82,
199,110,200,120,240,142,241
2410 REM <<<<< MANX >>>>>>>>>>>>>>>>
2420 DATA 192,181,204,181,210,199,192
,181
2430 REM <<<<< ANGLESEY >>>>>>>>>>>>>>>>
2440 DATA 206,157,206,151,210,160,206
,157

```

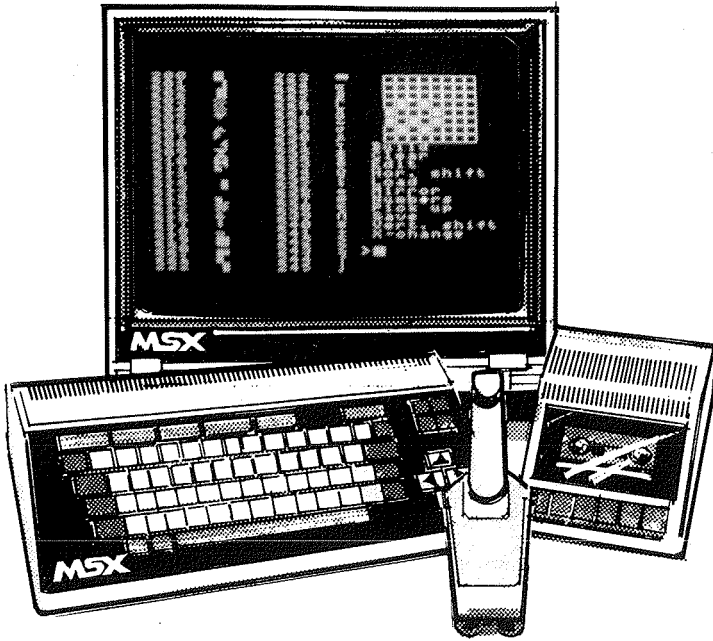
```

2450 REM <<<<< ISLE WIGHT >>>>>>>>>
2460 DATA 286,55,280,49,292,43,298,49
,286,55
2470 REM <<<<< CITIES >>>>>>>>>>>>
2480 DATA MANCHESTER,258,171,GTR.MANC
HESTER
2490 DATA CHESTER,252,149,CHESHIRE
2500 DATA LEEDS,282,175,YORKSHIRE
2510 DATA HULL,308,181,HUMBERSIDE
2520 DATA LIVERPOOL,250,163,MERSEYSID
E
2530 DATA BIRMINGHAM,274,119,MIDLANDS
2540 DATA LONDON,318,85,GTR.LONDON
2550 DATA BRISTOL,260,83,AVON
2560 DATA CARDIFF,244,95,GLAMORGAN
2570 DATA CAMBRIDGE,322,115,CAMBRIDGE
SHIRE
2580 DATA GLASGOW,216,261,STRATHCLYDE
2590 DATA EDINBURGH,238,261,LOTHIAN
2600 DATA ABERDEEN,250,315,GRAMPIAN
2610 DATA NEWCASTLE,266,223,NORTHUMBE
RLAND
2620 DATA BELFAST,170,200,ARMAGH
2630 DATA DUBLIN,160,160,DUBLIN
2640 DATA &H01,&H02,&H84,&H88,&HD0,&H
E0,&HF0,&HFC

```

25

Character Builder



The Create Your Own Monster Kit

Character building can be as much fun as playing a game. This 'utility' enables the MSX characters to be defined easily and then butted together to form more complex pictures.

The built-in characters are first displayed and then the opportunity is given to adapt them, swap them, or create completely new ones. The definition values can be noted down and then used in your own programs. Complete sets can be saved on tape to produce a library of useful characters.

Although many of the programs in this book have had characters designed for them, others use standard characters. With this character definer you can dramatically improve the appearance of these games.

Because of the wide range of features in this editor, it may take a little time learning how to drive it. The following list of commands will provide a handy reference guide during your 'Learner' period.

Character builder reference guide

After loading, the grid starts with the first character picked up. This may then be Edited, Cleared, or any of the following can be selected by pressing the first letter of the option.

Clear. This clears the grid completely.

Pick up. Asks for character number, then as the last digit is pressed that character will be transferred to the grid. It can then be Edited, Changed, etc.

Edit. Places the flashing cursor at the top left of the grid. It can then be moved around using cursors or joystick. Fire or space bar will set or unset a point. When finished 'Enter' is pressed. Then the number is required to transfer the pattern to.

Numbers. Asks for character to which the pattern will be assigned and then requires eight values, corresponding to the total for each line, starting at the top. This then generates the image directly. 'Enter' terminates this option.

Mirror. Asks for character number then asks whether horizontal (swap left to right) or vertical (turn upside down). The result is shown in the grid. To store the new character select 'Edit' followed by 'Enter' and then the character number for allocation.

Vert shift. Asks 'up' or 'down'. Moves the definition up or down one row with wraparound.

Hor shift. As for Vert shift, but left or right. Wrapped lines may be edited out.

X-change. Asks for two characters and then swaps their definitions.

Build. Removes the grid and provides a 14 × 10 character area in which to test how the patterns look when fitted together. The character number is requested and then that pattern may be placed anywhere using the cursor pad or the joystick. The fire button or space bar fixes a character. A new character is selected by firmly pressing 'Enter' and then entering the character number required. 'Enter' leaves Build and returns to grid.

Save. Asks for a file name to save with. 'Play' and 'Record' should be pressed next, then 'Enter' will save the current set.

Load. Asks for a file name to load, or if just 'Enter' is pressed the first file found will be loaded.


```

1000 REM >>>> CHARACTER DEFINER <<<
1010 REM >>>>>> ANDY/BOOTSY <<<<<<
1020 SCREEN 1,0,0:COLOR 15,1,1
1030 CLEAR 4000,&HEFFF:WIDTH 31
1040 DIM C$(8):DEFUSR=&HF000:DEFUSR1=
342
1050 ON ERROR GOTO 3440
1060 GOSUB 3470
1070 FOR A=1 TO 8:C$(A)="00000000":NE
XT A
1080 YP=8
1090 GOSUB 2510
1100 GOSUB 2750
1110 ON YP GOSUB 2150,2460,1170,1730,
3390,1930,3090,2300,3280,1580,3010
1120 LOCATE 18,19:PRINT STRING$(10,"
")
1130 LOCATE 18,20:PRINT STRING$(10,"
")
1140 GOTO 1100
1150 END
1160 REM >>>> EDIT <<<<
1170 XC=20:YC=0
1180 LOCATE XC,YC:PRINT CHR$(247):X=X
C-19
1190 FOR T=1 TO 50:NEXT T
1200 A=STICK(0) OR STICK(1)
1210 IF A=1 AND YC>0 THEN 1350
1220 IF A=3 AND XC<27 THEN 1390
1230 IF A=5 AND YC<7 THEN 1430
1240 IF A=7 AND XC>20 THEN 1470
1250 AA=STRIG(0) OR STRIG(1):IF AA TH
EN 1510
1260 A$=INKEY$:IF A$<>CHR$(13) THEN 1
180
1270 LOCATE XC,YC:IF MID$(C$(YC+1),X,
1)="1" THEN PRINT CHR$(248) ELSE PRIN
T CHR$(246)
1280 GOSUB 2890
1290 FOR A=1 TO 8
1300 NU=VAL("&B"+C$(A))
1310 VPOKE CH*8+(A-1),NU
1320 NEXT
1330 LOCATE 18,19:PRINT STRING$(10,"
"):LOCATE 18,20:PRINT STRING$(10," ")

```

```

:LOCATE 18,21:PRINTSTRING$(10," "):RE
TURN
1340 END
1350 Q#=MID$(C$(YC+1),X,1)
1360 LOCATE XC,YC
1370 IF Q#="1" THEN PRINT CHR$(248) E
LSE PRINT CHR$(246)
1380 YC=YC-1:GOTO 1180
1390 Q#=MID$(C$(YC+1),X,1)
1400 LOCATE XC,YC
1410 IF Q#="1" THEN PRINT CHR$(248) E
LSE PRINT CHR$(246)
1420 XC=XC+1:GOTO 1180
1430 Q#=MID$(C$(YC+1),X,1)
1440 LOCATE XC,YC
1450 IF Q#="1" THEN PRINT CHR$(248) E
LSE PRINT CHR$(246)
1460 YC=YC+1:GOTO 1180
1470 Q#=MID$(C$(YC+1),X,1)
1480 LOCATE XC,YC
1490 IF Q#="1" THEN PRINT CHR$(248) E
LSE PRINT CHR$(246)
1500 XC=XC-1:GOTO 1180
1510 X=XC-19:Q#=MID$(C$(YC+1),X,1)
1520 LOCATE XC,YC
1530 IF Q#="1" THEN PRINT CHR$(246):W
#=LEFT$(C$(YC+1),X-1)+"0"+RIGHT$(C$(Y
C+1),(8-X))
1540 IF Q#="0" THEN PRINT CHR$(247):W
#=LEFT$(C$(YC+1),(X-1))+"1"+RIGHT$(C$
(YC+1),(8-X))
1550 C$(YC+1)=W#
1560 GOTO 1180
1570 REM >>>> VERT.SHIFT <<<<<
1580 LOCATE 18,19:PRINT"Up or Down":L
OCATE 19,20:PRINT"U/D"
1590 A#=INKEY$:IFA#="" THEN 1590 ELSE
IF A#<>"U" AND A#<>"u" AND A#<>"D" AN
D A#<>"d" THEN 1590
1600 LOCATE 18,19:PRINTSTRING$(10," "
):LOCATE 18,20:PRINTSTRING$(10," ")
1610 IF A#="U" OR A#="u" THEN 1670
1620 D#=C$(8):FOR A=7 TO 1 STEP -1
1630 C$(A+1)=C$(A):NEXT A
1640 C$(1)=D#

```

```

1650 GOSUB 2820
1660 RETURN
1670 D#=C$(1):FOR A=2TO8
1680 C$(A-1)=C$(A):NEXT A
1690 C$(8)=D#
1700 GOSUB 2820
1710 RETURN
1720 REM >>>>    HOR.SHIFT    <<<<<
1730 LOCATE 18,19:PRINT"Left/Right"
1740 LOCATE 19,20:PRINT"(L/R)"
1750 A#=INKEY$:IF A#="" THEN 1750
1760 IF A#<>"R" AND A#<>"r" AND A#<>"
L" AND A#<>"I" THEN 1750
1770 LOCATE 18,19:PRINTSTRING$(10," "
)
1780 LOCATE 18,20:PRINTSTRING$(10," "
)
1790 IF A#="R" OR A#="r" THEN 1860
1800 FOR A=1 TO 8
1810 D#=LEFT$(C$(A),1)
1820 E#=RIGHT$(C$(A),7)
1830 C$(A)=E#+D#
1840 NEXT A
1850 GOSUB 2820:RETURN
1860 FOR A=1 TO 8
1870 D#=RIGHT$(C$(A),1)
1880 E#=LEFT$(C$(A),7)
1890 C$(A)=D#+E#
1900 NEXT A
1910 GOSUB 2820:RETURN
1920 REM >>>>>    mirror <<<<<<
1930 GOSUB 2890
1940 LOCATE 18,19:PRINT "horiz/vert"
1950 LOCATE 18,20:PRINT "H/V?";
1960 CH=USR1(0):CH#=INPUT$(1)
1970 IF CH#="v" OR CH#="V" THEN 2060
1980 IF CH#<>"h" AND CH#<>"H" THEN 19
60
1990 REM >>>>>    Y-MIRROR    <<<<<
2000 D#=C$(8):C$(8)=C$(1):C$(1)=D#
2010 D#=C$(7):C$(7)=C$(2):C$(2)=D#
2020 D#=C$(6):C$(6)=C$(3):C$(3)=D#
2030 D#=C$(5):C$(5)=C$(4):C$(4)=D#
2040 GOSUB 2820:RETURN
2050 REM >>>>>    X-MIRROR    <<<<<

```

```

2060 FOR A=1 TO 8
2070 D#=RIGHT$(C$(A),1):E#=LEFT$(C$(A),1)
2080 F#=MID$(C$(A),2,1):G#=MID$(C$(A),7,1)
2090 H#=MID$(C$(A),3,1):I#=MID$(C$(A),6,1)
2100 J#=MID$(C$(A),4,1):K#=MID$(C$(A),5,1)
2110 C$(A)=D#+G#+I#+K#+J#+H#+F#+E#
2120 NEXT A
2130 GOSUB 2820:RETURN
2140 REM >>>> BUILD <<<<<
2150 FOR Y=0 TO 7:LOCATE 19,Y:PRINT S
TRING$(10," "):NEXT Y
2160 XX=19:YY=0
2170 GOSUB 2890:IF CH=13 THEN GOSUB 2
460:RETURN
2180 PUT SPRITE 0,(XX*8+8,YY*8),15,0
2190 A=STICK(0) OR STICK(1)
2200 Y=STRIG(0) OR STRIG(1)
2210 CH#=INKEY#:IF CH#=CHR$(13) THEN
2280
2220 IF A=1 THEN YY=YY-(1 AND YY>0)
2230 IF A=3 THEN XX=XX+(1 AND XX<27)
2240 IF A=5 THEN YY=YY+(1 AND YY<7)
2250 IF A=7 THEN XX=XX-(1 AND XX>19)
2260 IF Y THEN LOCATE XX,YY:PRINT CHR
$(CH)
2270 GOTO 2180
2280 PUT SPRITE 0,(0,208),0,0:GOTO 21
70
2290 REM >>>> PICK-UP <<<<<
2300 GOSUB 2890
2310 FORA=1TO8:C$(A)="":NEXT A
2320 FOR A=0 TO 7
2330 AA=VPEEK(CH*8+A)
2340 B=AA AND 128:IF B>0 THEN LOCATE
20,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+
"1" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 2
0,A:PRINTCHR$(246)
2350 B=AA AND 64:IF B>0 THEN LOCATE 2
1,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"
1" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 21
,A:PRINTCHR$(246)

```

```

2360 B=AA AND 32:IF B>0 THEN LOCATE 2
2,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"
1" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 22
,A:PRINTCHR$(246)
2370 B=AA AND 16:IF B>0 THEN LOCATE 2
3,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"
1" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 23
,A:PRINTCHR$(246)
2380 B=AA AND 8:IF B>0 THEN LOCATE 24
,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"1
" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 24,
A:PRINTCHR$(246)
2390 B=AA AND 4:IF B>0 THEN LOCATE 25
,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"1
" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 25,
A:PRINTCHR$(246)
2400 B=AA AND 2:IF B>0 THEN LOCATE 26
,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"1
" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 26,
A:PRINTCHR$(246)
2410 B=AA AND 1:IF B>0 THEN LOCATE 27
,A:PRINT CHR$(248):C$(A+1)=C$(A+1)+"1
" ELSE C$(A+1)=C$(A+1)+"0":LOCATE 27,
A:PRINTCHR$(246)
2420 LOCATE 18,19:PRINTSTRING$(10," "
):LOCATE 18,20:PRINTSTRING$(10," "):L
OCATE 18,21:PRINTSTRING$(10," ")
2430 NEXT A
2440 RETURN
2450 REM >>>> CLEAR <<<<<
2460 FOR A=1 TO 8:C$(A)="00000000"
2470 NEXT A
2480 FOR Y=0 TO 7:LOCATE 19,Y:PRINT C
HR$(32)+STRING$(8,CHR$(246)):NEXT Y
2490 RETURN
2500 REM >>>> SET UP SCREEN <<<<
2510 KEY OFF:A=200:FOR Y=0 TO 22
2520 LOCATE 0,Y:PRINT A;" ";CHR$(A);"
";A+23;" ";CHR$(A+23)
2530 A=A+1:NEXT Y
2540 VPOKE 246*8,255:VPOKE 247*8,255:
VPOKE 248*8,0
2550 FOR B=1 TO 6:VPOKE 246*8+B,129:V
POKE 247*8+B,255:VPOKE 248*8+B,126
2560 NEXT B

```

```

2570 VPOKE 246*8+7,255:VPOKE 247*8+7,
255:VPOKE 248*8+7,0
2580 A$=STRING$(8,CHR$(246))
2590 FOR Y=0 TO 7
2600 LOCATE 20,Y:PRINT A$
2610 NEXT Y
2620 LOCATE 19,8:PRINT "Build"
2630 LOCATE 19,9:PRINT "Clear"
2640 LOCATE 19,10:PRINT "Edit"
2650 LOCATE 19,11:PRINT "Hor. shift"
2660 LOCATE 19,12:PRINT "Load"
2670 LOCATE 19,13:PRINT "Mirror"
2680 LOCATE 19,14:PRINT "Numbers"
2690 LOCATE 19,15:PRINT "Pick up"
2700 LOCATE 19,16:PRINT "Save"
2710 LOCATE 19,17:PRINT "Vert. shift"
2720 LOCATE 19,18:PRINT "X-change"
2730 RETURN
2740 REM >>> SELECT OPTION <<<<
2750 LOCATE 18,20:PRINT "> "+CHR$(2
9)+CHR$(29)+CHR$(29);
2760 A=USR1(0):A=ASC(INPUT$(1)) AND 2
23
2770 PRINT CHR$(A);:YP=0
2780 FOR I=6420 TO 6740 STEP 32
2790 IF A=VPEEK(I) THEN YP=(I-6420)/3
2+1:I=6740
2800 NEXT I:IF YP=0 THEN 2750
2810 RETURN
2820 FOR B=1 TO 8
2830 FOR C=1 TO 8
2840 IF MID$(C$(B),C,1)="1" THEN LOCA
TE (19+C),(B-1):PRINT CHR$(248) ELSE
LOCATE (19+C),(B-1):PRINT CHR$(246)
2850 NEXT C
2860 NEXT B
2870 RETURN
2880 REM >>>> char input <<<<<<
2890 LOCATE 18,19:PRINT "Character":C
H=USR1(0):LOCATE 18,20:PRINT ">"
2900 LOCATE 19,20:PRINT " ":LOCATE
19,20:CC=0
2910 CH=ASC(INPUT$(1)):IF CH=13 AND C
C=0 AND YP=1 THEN RETURN
2920 IF (CH<48 OR CH>57) AND CH<>8 TH

```

```

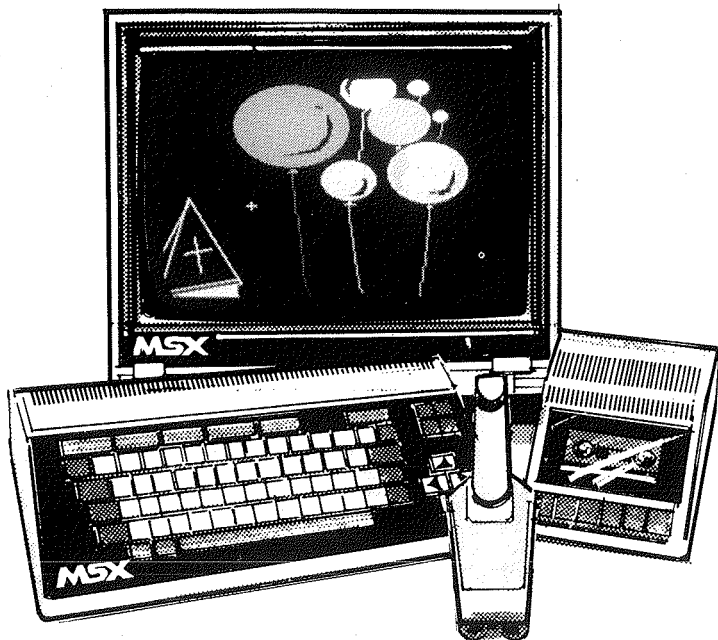
EN 2910
2930 IF CH=8 AND CC=0 THEN 2910
2940 IF CH=8 THEN PRINT CHR$(127);:CC
=CC-1:GOTO 2910
2950 PRINT CHR$(CH);:CC=CC+1
2960 IF CC<3 THEN 2910
2970 CH#=CHR$(VPEEK(6804))+CHR$(VPEEK
(6805))+CHR$(VPEEK(6806)):CH=VAL(CH#)
2980 IF CH<200 OR CH>255 THEN 2900
2990 RETURN
3000 REM >>>>> x change <<<<<<<
3010 GOSUB 2890:C1=CH*8
3020 GOSUB 2890:C2=CH*8
3030 FOR CH=0 TO 7
3040 C3=VPEEK(C1):C4=VPEEK(C2)
3050 SWAP C3,C4
3060 VPOKE C1,C3:VPOKE C2,C4
3070 C1=C1+1:C2=C2+1:NEXT CH:RETURN
3080 REM >>>>> NUMBERS <<<<<<
3090 GOSUB 2890
3100 FOR Y=0 TO 7:CH#=""
3110 LOCATE 19,20:PRINT STRING$(5," "
)
3120 LOCATE 18,20:INPUT CH#
3130 IF CH#="" THEN Y=7:NEXT Y:RETURN
3140 CC=VAL(CH#):IF CC>255 THEN 3120
3150 VPOKE CH*8+Y,CC
3160 NEXT Y:GOTO 3100
3170 REM >>>>> save <<<<<<
3180 REM the chars once saved may be
3190 REM included in another program
3200 REM as follows:-
3210 REM 10 SCREEN 1,
3220 REM 20 CLEAR 1000,&HEFFF
3230 REM 30 DEFUSR=&HF000
3240 REM 40 BLOAD "<name>"
3250 REM 50 A=USR(0)
3260 REM at the start of your
3270 REM program
3280 RESTORE 3450:FOR CH=&HF000 TO &H
F00B
3290 READ CC:POKE CH,CC:NEXT
3300 A=USR(0)
3310 FOR CH=&HF000 TO &HF00B
3320 READ CC:POKE CH,CC:NEXT CH

```

```
3330 LOCATE 18,19:PRINT "Filename"
3340 LOCATE 18,19:INPUT">";CH$
3350 IF CH$="" THEN CH$="CHR"
3360 BSAVE CH$, &HF000, &HF300
3370 RETURN
3380 REM >>>      LOAD      <<<<<<
3390 LOCATE 18,19:PRINT "Filename"
3400 LOCATE 18,20:INPUT">";CH$
3410 IF CH$="" THEN CH$="CHR"
3420 POKE &HF000,0:BLOAD CH$:IF PEEK(
&HF000) THEN A=USR(0)
3430 RETURN
3440 RESUME NEXT
3450 DATA 1,112,1,17,12,240,33,64,6,1
95,89,0,1,112,1,17,64,6,33,12,240,195
,92,0
3460 REM >>>>    DEF CROSS <<<<<<<<
3470 RESTORE 3510:CH$=""
3480 FOR CH=1 TO 8:READ CC
3490 CH$=CH$+CHR$(CC):NEXT CH
3500 SPRITE$(0)=CH$:RETURN
3510 DATA 24,24,24,255,255,24,24,24
```


26

Picture Builder



Your Phosphor Palette

This is hi-tech art with MSX, though it is unlikely that any of your 'masterpieces' will ever fetch thousands of pounds.

The program operates in Screen 2, which is the HI RES screen and provides a lot of useful features which will help you to produce some great pictures.

When the screen appears a graphic 'pencil' shows where the line will be drawn. The pencil is moved using the joystick or cursor pad. Pressing the fire button or space bar changes the pencil to an eraser symbol. This will then rub out any lines that it passes over. If the fire button is pressed again a cross symbol will be displayed. This indicates that the cursor may be moved without drawing a line. Once in position to draw again, pressing the fire button will return to the pencil and draw mode. The cursor can be cycled through these modes by repeated pressing of the fire button.

Whilst in the move mode (cross cursor) special features are available. Experiment with these to discover how powerful they can be. First press the home key 'c' to draw a circle, 'b' to draw a box, and

'insert' to fill; 'l' will draw lines. The technique for each is first to indicate to the computer the start position of the line, radius or base side, and then repeat the procedure to indicate the finish position.

At any time the current drawing ink can be changed using the 'Enter' key. Each press of the key steps through the available inks and the cursor changes colour to show the one selected.

'Select' will change the background colour.

'Graph' 'S' saves the current picture to tape (Play and Record should be pressed first as prompts are not displayed).

'Graph' 'L' then Play on the cassette will load the first picture found.

'Delete' pressed twice will clear the screen.

Although producing a good picture may take a little effort, depending on your skill, producing 'art' in this way can be very enjoyable.

```

1000 REM << PIC.BUILDER- BOOTSY >>>>
1010 REM
1020 SCREEN 2,2,0: CLEAR300,&HD7F3: COL
OR 15,0,0:CLS
1030 DIM S$(2):PC=15:BC=0:L=1:X=128:Y
=96:RESTORE
1040 GOSUB 1800
1050 GOTO 1430
1060 REM ----- input -----
1070 C=1
1080 FOR I=0 TO 2
1090 I$=INKEY$:IP=0:IF I$>" THEN IP=
ASC(I$)
1100 ST=STICK(I):TG=STRIG(I):IF ST OR
TG OR IP THEN C=0:I=2
1110 NEXT I:S=ST OR TG
1120 RETURN
1130 REM ----- put sprite -----
1140 XS=X-4:YS=Y-4
1150 PUT SPRITE 0,(XS,YS),PC,0
1160 RETURN
1170 REM ----- backdrop -----
1180 BC=BC+1 AND 15:COLOR,,BC
1190 RETURN
1200 REM ----- pen -----
1210 PC=PC+1 AND 15:COLOR PC
1220 GOSUB 1140

```

```

1230 RETURN
1240 REM --- line/circle/box -----
1250 IF CR OR L=0 AND X0=X AND Y0=Y O
R L AND IP<>11 THEN RETURN
1260 IF L THEN BEEP:X0=X:Y0=Y:L=0:RET
URN
1270 IF IP=76 OR IP=108 THEN LINE (X0
,Y0)-(X,Y),PC
1280 IF IP=66 OR IP=98 THEN LINE (X0,
Y0)-(X,Y),PC,B
1290 IF IP=67 OR IP=99 THEN CIRCLE (X
0,Y0),SQR((ABS(X0-X))^2+(ABS(Y0-Y))^2
),PC
1300 L=1:RETURN
1310 REM ----- fill -----
1320 PAINT (X,Y),PC:RETURN
1330 REM ----- cls -----
1340 TC=PC
1350 PC=0:GOSUB 1140
1360 GOSUB 1070:IF IP AND IP<>127 THE
N 1410
1370 PC=15:GOSUB 1140
1380 GOSUB 1070:IF C THEN 1350
1390 IF IP<>127 THEN 1410
1400 CLS
1410 PC=TC:GOSUB 1140:RETURN
1420 REM ----- draw -----
1430 GOSUB 1070:IF C THEN 1430
1440 X=X+(1 AND ST>1 AND ST<5 AND X<2
55)-(1 AND ST>5 AND X>0)
1450 Y=Y+(1 AND ST>3 AND ST<7 AND Y<1
91)-(1 AND (ST>0 AND ST<3 OR ST>7) AN
D Y>0)
1460 GOSUB 1140
1470 CR=CR+(1 AND TG) AND 3
1480 CR=CR-(CR AND CR=3)
1490 IF TG THEN SPRITE$(0)=S$(CR)
1500 IF CR=1 THEN PSET(X,Y),PC
1510 IF CR=2 THEN PRESET(X,Y)
1520 IF S THEN 1430
1530 REM ----- command -----
1540 RESTORE 2020:CN=0
1550 READ CC:CN=CN+1:IF CC=0 THEN 142
0
1560 IF CC<>IP THEN 1550

```

```

1570 ON CN GOSUB 1180,1210,1250,1250,
1340,1320,1250,1250,1250,1250,1250,12
50,1600,1700
1580 GOTO 1420
1590 REM ----- save -----
1600 RESTORE 2030:DEFUSR=&HD7F4
1610 FOR I=&HD7F4 TO &HD7FF
1620 READ H$:POKE I,VAL("&h"+H$)
1630 NEXT I
1640 A=USR(0)
1650 BSAVE "picpat",&HD800,&HF000
1660 POKE &HD7FC,&H20:A=USR(0)
1670 BSAVE "piccol",&HD800,&HF000
1680 RETURN
1690 REM ----- load -----
1700 RESTORE 2040:DEFUSR=&HD7F4
1710 FOR I=&HD7F4 TO &HD7FF
1720 READ H$:POKE I,VAL("&h"+H$)
1730 NEXT I
1740 BLOAD "picpat"
1750 A=USR(0)
1760 BLOAD "piccol"
1770 POKE &HD7F9,&H20:A=USR(0)
1780 RETURN
1790 REM ----- def sprites -----
1800 FOR I=0 TO 2
1810 S$="":FOR J=1 TO 32
1820 READ H$:S$=S$+CHR$(VAL("&h"+H$))
1830 NEXT J
1840 S$(I)=S$
1850 NEXT I
1860 SPRITE$(0)=S$(0)
1870 GOSUB 1140
1880 RETURN
1890 REM ----- prog data -----
1900 DATA 10,10,10,fe,10,10,10,0
1910 DATA 0,0,0,0,0,0,0,0
1920 DATA 0,0,0,0,0,0,0,0
1930 DATA 0,0,0,0,0,0,0,0
1940 DATA 0,0,0,10,c,7,7,3
1950 DATA 1,0,0,0,0,0,0,0
1960 DATA 0,0,0,0,0,0,80,c0
1970 DATA +4,fa,fd,7f,3e,1c,0,0
1980 DATA 0,0,0,e,f,7,3,1
1990 DATA 0,0,0,0,0,0,0,0

```

2000 DATA 0,0,0,0,0,80,c0,e0

2010 DATA f0,78,3c,1e,f,0,0,0

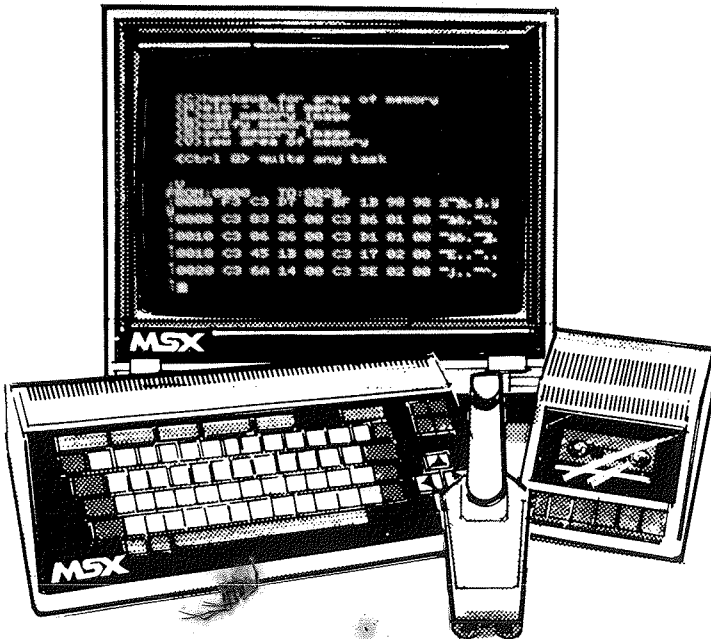
2020 DATA 24,13,11,12,127,18,66,67,76
,98,99,108,210,200,0

2030 DATA 1,0,18,11,0,d8,21,0,0,c3,59
,0

2040 DATA 1,0,18,11,0,0,21,0,d8,c3,5c
,0

27

MSX Monitor



Open RAM Surgery

This utility will help with investigations into memory. The machine's ROM routines can be looked at or any program which can co-reside in RAM can be examined. Memory can be modified, saved or loaded.

Command summary

View Memory. Asks for the start and finish locations, which have to be hexadecimal. The memory contents will then be displayed in blocks of ten. 'Space' steps through to the end. 'Text' can be read from the ASCII columns.

Checksum. Asks for the start and finish. All the values are then totalled in decimal, inclusive of the addresses entered. This option is useful when entering from a listing which provides a 'Checksum' to check if errors have occurred.

Modify mem. Again the start and the end addresses are required. Then the display shows the current value. Pressing 'Enter' will retain

this value and the next location will be displayed. Typing numbers followed by 'Enter' will put what is typed into the current address. If the value is invalid then this will be indicated and no changes will be made to memory. The next location will then be shown. *Great care should be taken*, since *any RAM* may be modified with perhaps surprising results. (All entries are in hexadecimal.)

Save. Asks for start and finish (hex), then the file name. Saving commences when 'Enter' is pressed. 'Record' and 'Play' should be pressed prior to selecting the 'Save' option.

Load. This, when followed by pressing 'Play', loads the first file found.

'Grph' 'Q' will quit any option.

```

1000 REM <<<<<<<< MONITOR >>>>>>>>>
1010 REM <<<<<<<< BOOTSY. >>>>>>>>>
1020 SCREEN0:KEY OFF
1030 GOTO 1190
1040 Q=0:IP$=INPUT$(1):IP=ASC(IP$):IF
  IP>64 THEN IP=IP AND 223:IP$=CHR$(IP
)
1050 IF IP=17 THEN Q=1
1060 RETURN
1070 H$=""
1080 GOSUB 1040:IF Q=1 THEN RETURN
1090 IF IP=8 AND H$>"" THEN H$=LEFT$(
H$,LEN(H$)-1):PRINT CHR$(127);
1100 IF IP=13 THEN RETURN
1110 IF (IP<48 OR IP>57) AND (IP<65 O
R IP>70) THEN GOTO 1080
1120 IF LEN(H$)=4 THEN GOTO 1080
1130 H$=H$+IP$:PRINT IP$;:GOTO1080
1140 PRINT "FROM: ";:GOSUB 1070:IF Q=1
  THEN RETURN
1150 FH=VAL("&h"+H$)
1160 PRINT TAB(12) "TO: ";:GOSUB 1070:
IF Q=1 THEN RETURN
1170 TH=VAL("&h"+H$)
1180 RETURN
1190 PRINT "Basic Monitor":PRINT
1200 PRINT
1210 PRINT " (C)hecksum for area of m
emory""
1220 PRINT " (H)elp - this menu"
1230 PRINT " (L)oad memory image"

```

```
1240 PRINT " (M)odify memory"
1250 PRINT " (S)ave memory image"
1260 PRINT " (V)iew area of memory"
1270 PRINT:PRINT " <Ctrl Q> quits any
task":PRINT:PRINT
1280 PRINT": ";
1290 GOSUB 1040
1300 PRINT CHR$(IP);
1310 RESTORE:K=0:FOR I=1 TO 6
1320 READ J:IF IP=J THEN K=I:I=6
1330 NEXT
1340 IF K=0 THEN PRINT "?":GOTO 1280
1350 PRINT:ON K GOSUB 1380,1440,1550,
1680,1550,1370
1360 GOTO 1280
1370 GOTO 1210
1380 GOSUB 1140:IF Q=1 THEN PRINT:RET
URN
1390 PRINT TAB(24) "CHECKSUM: ";
1400 IF TH<FH THEN PRINT "?":RETURN
1410 H=0:FOR I=FH TO TH:H=H+PEEK(I)
1420 H=H MOD 256:NEXT I
1430 H$=HEX$(H):PRINT STRING$(2-LEN(H
$), "0")+H$:RETURN
1440 GOSUB 1140:PRINT:IF Q=1 THEN RET
URN
1450 IF TH<FH THEN TH=FH+7
1460 L=0:FOR I=FH TO TH STEP 8:L=L+1
1470 H$=HEX$(I):PRINT ": ";STRING$(4-L
EN(H$), "0")+H$; " ";
1480 FOR J=0 TO 7
1490 H$=HEX$(PEEK(I+J)):PRINT STRING$
(2-LEN(H$), "0")+H$; " ";:NEXT J
1500 FOR J=0 TO 7:K=PEEK(I+J)
1510 IF K>31 THEN PRINT CHR$(K); ELSE
PRINT ".";
1520 NEXT J:PRINT:IF L=23 THEN I$=INP
UT$(1):L=0:IF ASC(I$)=17 THEN RETURN
1530 NEXT I
1540 RETURN
1550 S$=IP$
1560 INPUT"Filename: ";F$
1570 IF F$="" THEN RETURN
1580 Q=0:IF S$="S" THEN GOSUB 1140
1590 PRINT:IF Q=1 THEN RETURN
```



```

1600 IF TH<=FH AND S$="S" THEN PRINT
":?":RETURN
1610 IF S$="S" THEN PRINT "Press PLAY
/REC then any key..";
1620 IF S$="L" THEN PRINT "Press PLAY
then any key..";
1630 I$=INPUT$(1):PRINT:IF ASC(I$)=17
THEN RETURN
1640 ON ERROR GOTO 1670
1650 IF S$="S" THEN BSAVE F$,FH,TH
1660 IF S$="L" THEN BLOAD F$
1670 RETURN
1680 GOSUB 1140:PRINT:IF Q=1 THEN RET
URN
1690 IF TH<FH THEN PRINT ":?":RETURN
1700 FOR I=FH TO TH:H$=HEX$(I)
1710 PRINT " ";STRING$(4-LEN(H$),"0")
;H$;" ";H$=HEX$(PEEK(I))
1720 PRINT STRING$(2-LEN(H$),"0");H$;
" ";
1730 H$=""
1740 GOSUB 1040:IF Q=1 THEN I=TH:NEXT
:PRINT:RETURN
1750 IF IP=8 AND H$>" " THEN H$=LEFT$(
H$,LEN(H$)-1):PRINT CHR$(127);
1760 IF IP=13 THEN PRINT IP$:IF H$>" "
THEN POKE I,VAL("&H"+H$):NEXT:RETURN
ELSE NEXT:RETURN
1770 IF LEN(H$)=2 THEN GOTO 1740
1780 IF (IP<48 OR IP>57) AND (IP<65 O
R IP>70) THEN GOTO 1740
1790 H$=H$+IP$:PRINT IP$;:GOTO 1740
1800 DATA 67,86,83,77,76,72

```

Final Words

In producing this book an attempt has been made to adopt a standard of game which is well above the norm.

This naturally leads to longer programs which require greater care to enter. The time will come therefore when you will need to debug your programs. This is when you have the opportunity to try to understand how and why the programs work.

To help you in this task there follows a 'debug guide'.

General points

1. Get someone else to double check that the entry matches the listing. Say it out loud.

2. Remember that in 37-column Screen \emptyset the listing should match the screen line by line.

3. Be careful that an I has not been confused with a 1.

4. Ensure that lower case 'a' is entered as such and not 'A'.

5. Watch out for ';' instead of ':'.

6. Double check spaces by reference to the spaces above and below.

7. Double check line numbers. Don't confuse a number which flows over onto another line as a line number.

8. Treble check data. It is very easy to skip a few numbers. As an overall check, count how many commas and compare with the listing.

9. Make sure that all quotes are in.

10. Is there really a fault? Re-read the instructions for play. Most responses need to be followed by the 'Enter' key.

Specific problems

1. To save space whilst still including substantial programs some programs use sections from other listings, so make certain that everything has been typed in as required. We anticipate some problems from readers in using this technique, but consider the finished results to be worth while for the majority.

2. One of the main sources of program failure is poor handling of bad entries. For example, a program may crash if an unexpected minus number is entered. It is possible to anticipate most 'silly' entries but only at the expense of needlessly longer listings. In writing these programs it has been assumed that the player will not set out to crash the program! They are therefore only protected against class three idiots. If you will be exposing your finished efforts to a hostile environment then it is up to you to protect against class one idiots.

3. All programs should work equally well on a monochrome monitor or TV but change the colours (after saving the correct version) to suit your own taste. *Micro Mind* can use symbols instead of colours to make the game playable in monochrome.

4. It is always good practice to load into a machine which has just been powered up. This can avoid problems if the previous program left something changed deep within the machine.

5. Unfortunately the computer's 'Random' function has been found to be very predictable. Care has been taken to minimise this problem whenever possible but you may be able to remember certain sequences.

Debugging

If none of the foregoing resolves the problem then roll up your sleeves and note some of these tips.

1. If an error message indicates a fault is in a particular line, but it looks right when examined, remember that the fault could be in the line which passes information to that line. Check back related lines.

2. Errors such as 'Out of range', 'Improper argument', etc., may be traced as above. To help type 'PRINT A' or whatever variables are used in that line. This should indicate the problem.

3. Bits being missed out! The 'Tron' function will show the order in which lines are called, often highlighting a rogue 'GOSUB' or similar problem.

4. Work methodically. Consider each section in turn. Set 'Break points' using 'End' or dump out variables by adding a short print line in the middle of the faulty routine.

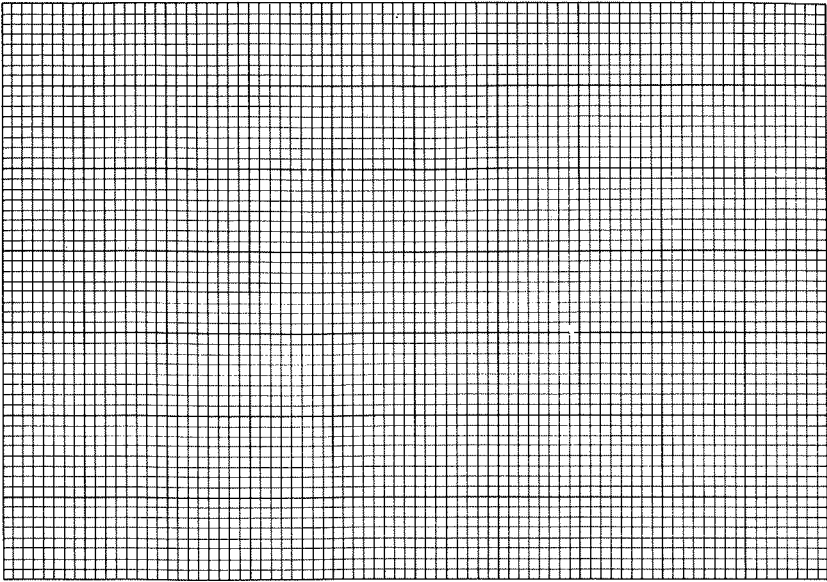
5. Temporarily disable sections of program by adding a 'REM' to the start of a line. See what the effect is.

6. Re-type the section responsible! Some errors can only be mysteriously solved by re-entering a single 'perfect' line.

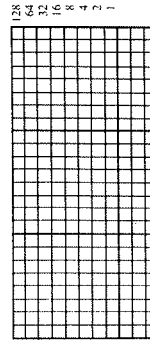
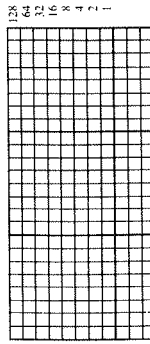
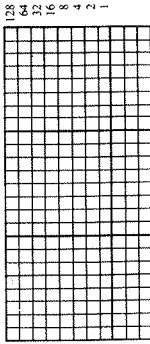
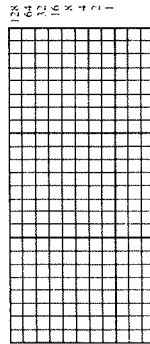
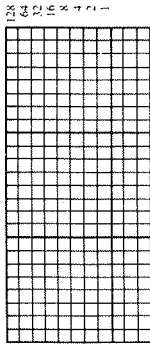
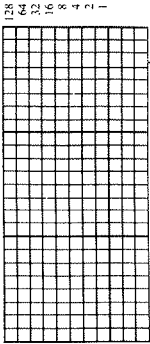
7. Use the 'On error', 'Er!' and 'Err' functions normally as described in the manual.

8. Go to bed – have a sleep. It may work the next day!

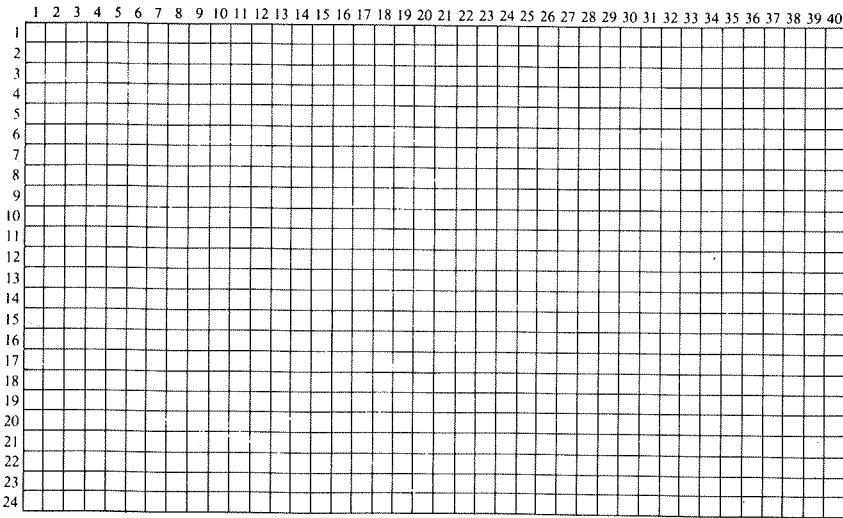
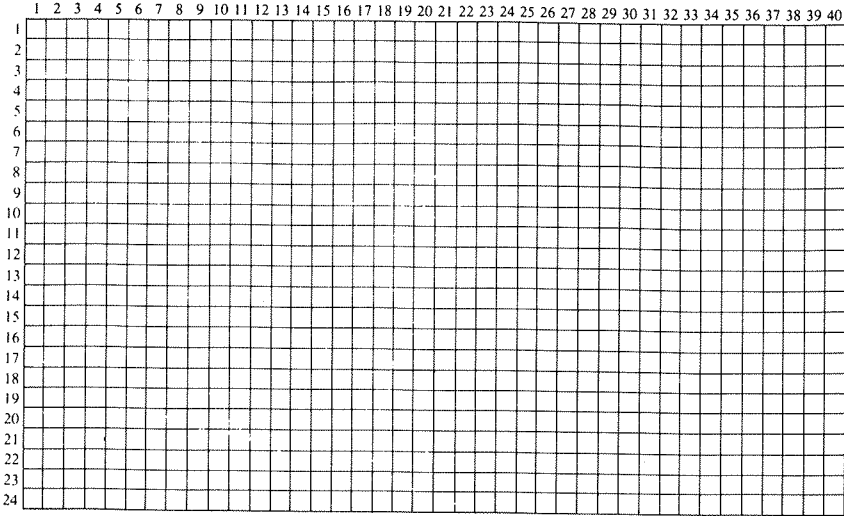
Appendix 1: Sprite/Character Design Aid



DEFINITIONS



Appendix 2: Screen Layout Planner



SCREEN 0 = MAX WIDTH = 40
SCREEN 1 = MAX WIDTH = 32
SCREEN 2 = HI RES 256 × 192 points
SCREEN 3 = LORES 64 × 48 points

Appendix 3: MSX Memory Map

	BASIC WORK AREA	&HFFFF
USER'S PROGRAM AREA	I/O CONTROL	
	STRING STORAGE INITIALLY 200 CHANGE WITH CLEAR	
	STACK	&HF380
	SPARE AREA	
	ARRAY STORAGE	
	VARIABLE STORAGE	
	BASIC PROGRAM TEXT	&H8000
32K MSX-BASIC		&H0000

Appendix 4: MSX Colour Table

	COLOUR		COLOUR
0	Clear	8	Red
1	Black	9	Bright red
2	Green	10	Yellow
3	Light green	11	Light yellow
4	Dark blue	12	Dark green
5	Light blue	13	Purple
6	Dark red	14	Grey
7	Sky blue	15	White