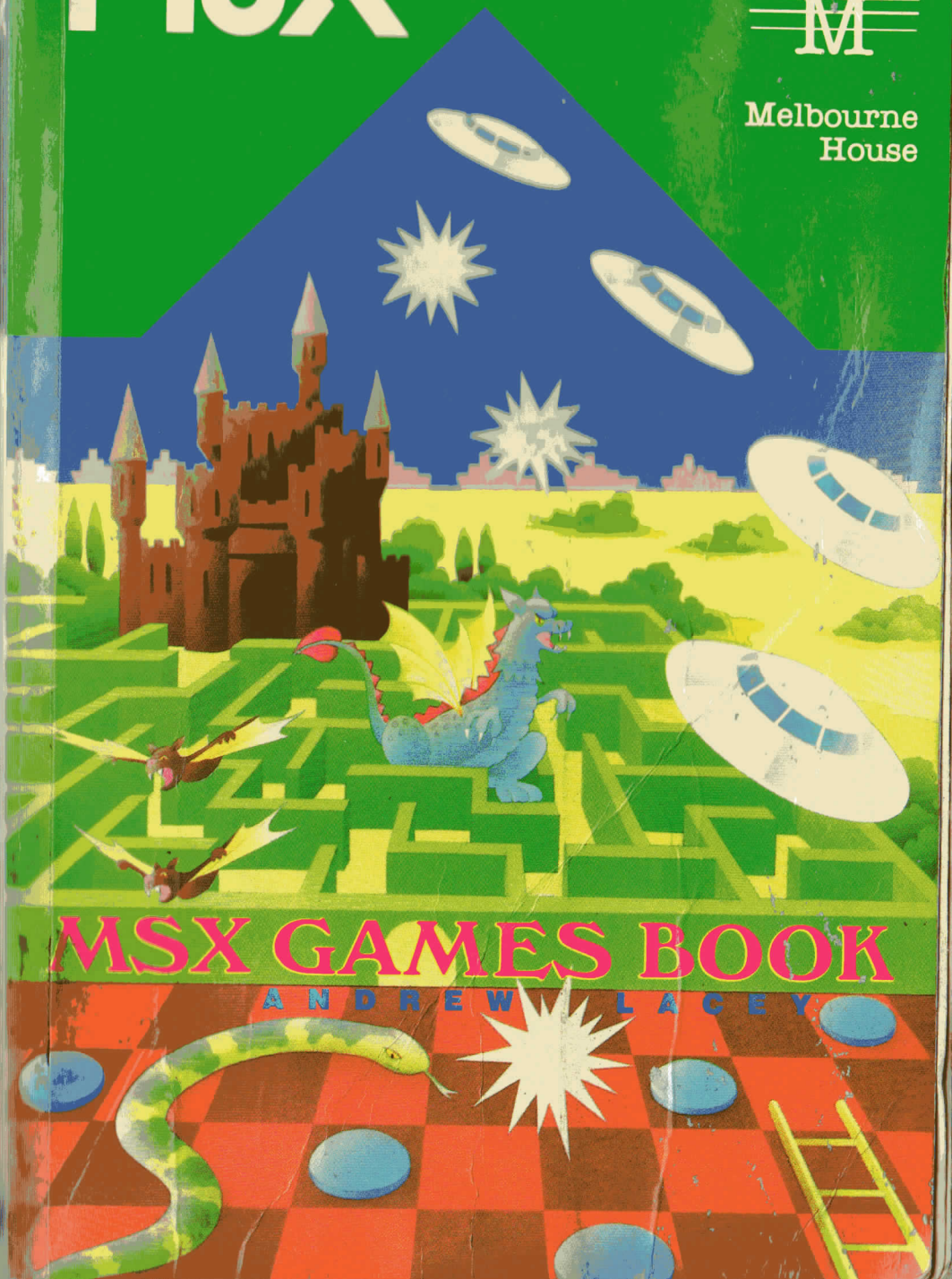
MSX

Melbourne House



MSX GAMES BOOK

ANDREW LACEY

MSX
Games Book

Andrew Lacey

MELBOURNE HOUSE
PUBLISHERS

IN THE UNITED KINGDOM—
Melbourne House (Publishers) Ltd
Castle Yard House
Castle Yard
Richmond, TW10 6TF

IN THE UNITED STATES OF AMERICA—
Melbourne House Software Inc.
347 Reedwood Drive
Nashville TN 37217

IN AUSTRALIA—
Melbourne House (Australia) Pty Ltd
2nd Floor, 70 Park Street
South Melbourne, Victoria 3205

Cataloguing in Publication

Lacey, Andrew.
MSX games book.

ISBN 0 86161 172 1

1. Computer games. 2. MSX Basic (Computer program language).
I. Title

794.8'2

EDITION: 7 6 5 4 3 2 1
PRINTING: F E D C B A 9 8 7 6 5 4 3 2 1
YEAR: 90 89 88 87 86 85 84

Contents

Race Game

Memory Games

Simulation Games

Time-Limit Games

Educational Games

Parlor Games

Appendices

Preface

Computers built to the MSX standard are set to revolutionise the home-computer market. First, they offer a much-improved BASIC with powerful graphics and sound statements, making them ideal for recreational and educational software. Second, they offer software houses the possibility of producing programs which will run on many different computers, which means a wide availability of first-class software for the MSX owner.

With this book, Melbourne House continues its tradition of producing high quality literature and software in support of personal computers.

Arrangement of Programs

The games programs are graded within the book from easy to difficult. However, because different people have preferences for particular kinds of games, they are listed in the Contents page grouped by type. If you enjoy strategy games, you will find them all by consulting the random-access Contents.

Ease of Use

To speed program debugging, the ChexSum utility program enables you to identify incorrectly-typed lines immediately. (This aid is unique to Melbourne House games books.)

But — and this is an important but — read the chapter 'Program Accuracy' first if you want to get the maximum benefit from all the aids to accurate transcription.

Playing the Games

To help you learn to play these games more quickly, game commands have been standardised. Unless otherwise stated in the game title page, the cursor and SPACE bar keys are used as follows:

KEY		USAGE
Cursor up	(↑)	Move sights or object up
Cursor down	(↓)	Move sights or object down
Cursor left	(←)	Move sights or object left
Cursor right	(→)	Move sights or object right
SPACE		Fire button

Programming Projects

I hope that as you read and use the book, you will absorb the principles of programming your MSX computer by osmosis. Lists of variables have been provided and subroutine names inserted in listings to help you discover how programs work.

I have also suggested improvements you may make to the games as a challenge to your programming skills.

Suggestions

Melbourne House is always interested in feedback from readers, whether it be suggestions, praise or complaints — see 'Write to Us' at the end of the book.

Happy computer gaming!

Andrew Lacey
September 1984

Program Accuracy

Programs take a long time to type in and debug. To help make programs easier to read and debug, printouts have been specially formatted and the ChexSum utility has been provided to help you get programs running. Read the rest of this chapter and enter the ChexSum program before you enter any games programs.

FORMATTED LISTINGS

The program listings in this book have been processed through a formatting program to improve their appearance and readability. The formatter:

- Aligns statements at the left, avoiding the margin stepping that occurs as line numbers increase,
- Inserts spaces between keywords, and
- Transforms SPACE characters occuring in strings, i.e. between quote marks (''), into a small black pyramid (▲).

The formatting process makes programs much easier to read but may cause errors in typing unless the hints which follow are obeyed.

TYPING RULES

In the program listings spaces have only been used as an aid to readability. For example, the line

```
100  POKE 54273, 1 : PRINT "▲STRING▲"
```

would be entered into your computer as

```
100 POKE54273,1:PRINT" STRING "
```

Spaces are often used in strings but only enter a space when you see a (▲) symbol. This character does not appear on the keyboard but is used to tell you when to press the SPACE bar.

Character Confusion

In the printouts certain similar characters may be confused and entered incorrectly, and cause your program to crash: they are the letter I and the figure 1, and the letter O and the figure 0. In the text, figure zero (0) is shown cancelled (Ø). In listings these confusing characters are soon recognised:

CHARACTER	PRINTOUT
Letter I	I
Figure 1	1
Letter O	O
Figure Ø	Ø

If you are modifying programs and entering code from your own notes, a number of handwritten characters are easily confused unless clearly distinguished. We offer you this guide:

CONFUSED CHARACTERS	RECOMMENDED FORM
Letter Z, figure 2	Z, 2
Letter O, figure Ø	O, Ø
Letter S, figure 5	S, 5
Letter I, figures 1 and 7	I, 1, 7

Punctuation may also be a trap:

- Commas (,) and full stops or periods (.) are not interchangeable.
- Colons (:) and semi-colons (;) are not interchangeable.
- Apostrophes (') and double quotes (") are not interchangeable.
- Use parentheses ( ) and brackets [ ] appropriately. Always ensure that parentheses and brackets are correctly paired: a quick check is to make sure that you have as many righthand as lefthand parentheses or brackets in a mathematical expression.
- The SHIFT key must be pressed when some characters (such as ! " # $ %, etc.) are entered.

CSAVE Before RUNning

Do you remember the old proverb?

> Programs typed and saved today,
> Live to RUN another day.

Be sure to progressively save the programs onto cassette or disk as you enter the listings. From time to time the keyboard may lock up, especially if you try to RUN a program that contains an error. We strongly recommend that you keep saving copies of the program as you enter it into the computer and debug any typing errors.

Note Some programs contain loaders to place machine language routines in high memory and which may alter certain system pointers. If such a program is run and crashes, the machine-language program and the pointers will remain as they were even when the NEW command has been entered. When debugging programs, save your work before running and always reset the computer after a crash to avoid crashing a program subsequently entered or loaded.

CHEXSUM PROGRAM VERIFICATION

When games programs such as these are keyed in, invariably reading and typing mistakes creep in. You then spend ages trying to sort out where and what is causing the error. Even experienced programmers often cannot easily identify an error and need to do the tedious job of double-checking with the book, especially with DATA statements.

To avoid this major cause of frustration when entering the programs, there are two short routines in this book which you should enter and save before you enter other programs:

- Finder and
- ChexSum.

ChexSum calculates a unique check-sum number for each line of a games program, then prints out a table of program line-numbers and their corresponding line check-sums, plus a grand total for the whole program.

When you have entered a game program, run ChexSum and compare the check-sum table with that in the book. Any discrepancies immediately identify a mistyped line.

For ChexSum to operate correctly, the memory address where BASIC programs start must be inserted into ChexSum. The start of BASIC memory varies with MSX computers having different amounts of RAM. Finder is a program which locates the correct start-of-memory address for your computer.

Saving Finder and ChexSum

Follow these instructions to enter Finder and ChexSum and save them to tape. Do not put games programs on the same tape or you will spend considerable time winding and rewinding the tape to, first, load your program and, second, to load and use ChexSum.

1 Type in the Finder listing below and save it using the command CSAVE "FINDER".

Note Hand check Finder character by character after you have saved it. Finder is the only program in the book which cannot be verified by ChexSum. Why? Because you need Finder to initialise ChexSum to verify Finder and, if Finder is incorrect, this won't happen.

2 RUN Finder; while running, it will display a constantly increasing number labelled LOCATION. When it is finished, it will display the message

Found it! — BASIC starts at location NNNNN

where NNNNN is the address of the start of BASIC programs. Change SMEM in line 62000 of ChexSum to this value.

3 Type in the ChexSum listing below and save it using the command SAVE "CAS: SELCHK"; SelfChek is the self-checking version of ChexSum.

Note The SAVE command causes the ChexSum program to be stored on tape in ASCII alphanumeric characters. This is necessary because the MERGE statement used for adding ChexSum to the end of your games programs for checking will only work with ASCII program files.

4 Load SelfChek with the statement LOAD "CAS: SELCHK".

5 Run SelfCheck by typing RUN. It will first of all process itself, generating a table of line sums and a grand total for the whole program.

6 At this point you may or may not have had a crash. If a table was successfully generated, compare your table with that given below. If the grand totals differ, check the line sums to find and correct the incorrect line.

Note SelfChek and ChexSum will sometimes generate different sums for the same line, even when the program is CORRECT. These differences are caused by the amount of memory in your system — see limitations of ChexSum in 'Debugging Hints' below.

7 Repeat steps 4 to 7 until ChexSum is debugged.

8 In line 62020, alter the value of E = 62500 to E = 61999. Save a copy to tape using the command SAVE "CAS: CHXSUM".

Note The version called ChexSum is your working copy used to check all the other programs in the book. Make sure you don't confuse it with your initial version, SelfChek, which is used only to verify ChexSum itself.

Finder Program

```
  5   REM test 12345
  7   B$ = " 32 116 101 115 116 32 49 50 51 52 53": CLS :
      PRINT : PRINT : PRINT "Location"; CHR$( 11 ) ; TAB(
      11 ) "FINDER": PRINT : PRINT : PRINT
 10   PRINT CHR$( 11 ) ; MEM : PE = PEEK( MEM ) : IF PE
      <> 143 THEN MEM = MEM + 1 : GOTO 10
 15   T = MEM : A$ = "": FOR I = 1 TO 11 : MEM = MEM +
      1 : PE = PEEK( MEM ) : A$ = A$ + STR$( PE ) : NEXT
 20   IF A$ <> B$ THEN MEM = T + 1 : GOTO 10
 30   PRINT : PRINT : PRINT "Found it! - BASIC starts at":
      PRINT "location "; T - 4
```

The value of SMEM = 32769 was obtained on a 16K byte machine and will vary for machines with larger amounts of memory — see also 'Debugging Hints' below.

ChexSum Program

```
62000   SMEM = 32769!
62001   CLS : PRINT  TAB( 10 ) ; "CHEXSUM": PRINT : PRINT :
        PRINT
62004   INPUT  "Line Number to Start"; ST
62005   CLS : PRINT TAB( 10 ) "CHEXSUM": FOR I = 1 TO 6 :
        PRINT : NEXT : PRINT TAB( 4 ) "Output to Printer"
        TAB( 24 ) "<P>": PRINT : PRINT : PRINT TAB( 4 )
        "Output to Screen" TAB( 24 ) "<S>": FOR I = 1 TO
        5 : PRINT : NEXT : PRINT "      Press 'P' or 'S'"
62010   X$ = INKEY$ : IF X$ = "s" OR X$ = "S" THEN 62015
        ELSE IF X$ <> "P" AND X$ <> "p" THEN 62010
62015   REM PRINTER ROUTINE
62020   CLS : PRINT "Check Sum:-": E = 62500! : LINK = SMEM
62100   REM Main Loop
```

```
62120   T = LINK
62130   LINK = PEEK( T + 1 ) * 256 + PEEK( T )
62135   LN = PEEK( T + 3 ) * 256 + PEEK( T + 2 ) : IF LN
        < ST  THEN  T = LINK : GOTO  62130
62136   IF LN > E THEN PRINT : PRINT "Total="; CH : PRINT :
        INPUT "Remove▲Chexsum▲from▲game▲▲▲▲▲program▲(<Y>▲or▲<N>
        )"; X$ : IF X$ = "Y" OR X$ = "y" THEN LINK = SMEM :
        GOTO 62200 ELSE END
62137   PRINT LN ; TAB( 7 ) ;
62140   CS = 0 : N = 0 : C = 0
62150   FOR P = T + 4 TO LINK - 2 : PK = PEEK( P )
62160   IF PK = 143 THEN P = LINK - 2 : GOTO 62190
62165   IF PK = 34 THEN C = ( C = 0 )
62170   IF C = 0 AND PK = 32 THEN 62190
62180   IF PK = 137 THEN N = N + 1 : CS = CS + ( 203 OR
        N ) : PK = 164
62185   N = N + 1 : CS = CS + ( PK OR N )
62190   NEXT  P :  CH = CH + CS : PRINT "="; CS : GOTO 62120
62200   T = LINK
62210   LINK = PEEK( T + 1 ) * 256 + PEEK( T )
62220   LN = PEEK( T + 3 ) * 256 + PEEK( T + 2 )
62230   IF LN <> 62000! THEN 62200 ELSE POKE T, 0 : POKE
        T + 1, 0
62999   REM
```

CHEXSUM TABLE

```
62000 = 811          62130 = 2583         62185 = 2166
62001 = 2048         62135 = 5488         62190 = 2626
62004 = 2469         62136 = 13424        62200 = 638
62005 = 15816        62137 = 724          62210 = 2583
62010 = 5863         62140 = 1217         62220 = 2698
62015 = 0            62150 = 2731         62230 = 2982
62020 = 3365         62160 = 550
62100 = 0            62165 = 1557          Total=78888
62120 = 638          62170 = 1778
```

Using ChexSum

The greatest problem encountered when typing in programs from a book is errors made by the user. Most of these are picked up when the computer responds to the RUN command with the 'Syntax Error' message. The user then has only to LIST the line and compare it with the line in the book. Unfortunately, some errors are more subtle and not fatal to program operation. These types of errors will cause the program to run, but incorrectly, and the computer will not be able to detect them as such.

CheckSum is a special program which generates a unique sum for each line in a program and a grand total of all line sums. After each program listing is a table of check sums. You need only compare the numbers in the ChexSum table for each program with those generated by ChexSum. If two numbers differ, check that particular line.

1 Type in your game program, Patterns, say. Save it to tape with the statement CSAVE "PATTNS".

Note All games may be saved using CSAVE, which stores programs in compressed format, saving space on the tape and allowing programs to load faster.

2 Reload your game program if necessary, using the statement CLOAD "PATTNS" for the first game in the book. Do not RUN the game program at this point.

3 To join ChexSum to the end of your program, enter the statement MERGE "CAS: CHXSUM".

4 When merged, enter RUN 62000 to activate ChexSum. The program will prompt:

Line Number to Start?

Pressing the ENTER key at this point will cause ChexSum to begin with the first line of your program. Entering a line number will start ChexSum at that line.

5 ChexSum next prompts:

Output to Printer <P>
Output to Screen <S>

Press <P> or <S>

Entering a P will cause output to go to the printer, S to go to the screen.

6 ChexSum will now output the check-sum table for the program. To halt the scroll, hit the <STOP> key and to restart the scroll, hit the <STOP> key again. When finished, ChexSum will prompt:

Remove ChexSum From Game Program (<Y> or <N>)?

Entering a Y will delete ChexSum from the end of your game, so select N.

7 Check your grand total with that in the book. If they differ, a line has been typed incorrectly. Compare line numbers until you locate the bad lines and then edit them.

Note ChexSum will sometimes produce erroneous line sums in a CORRECTLY ENTERED program — see 'Debugging Hints' below.

8 Repeat steps 4 to 7 until the games program is debugged. In step 4, enter the number of the first bad line to avoid ChexSum verifying the games program from the first line.

9 When the game program is running satisfactorily, in step 6 enter Y to remove ChexSum from the end of your game program.

10 Save your debugged game program to another tape, using this statement:

```
CSAVE "PATTNS".
```

DEBUGGING HINTS

Limitations of ChexSum

ChexSum is not entirely foolproof and cannot detect two errors which give a (seemingly) correct output:

1 If your grand total is correct but the program crashes, two lines have been typed in incorrectly and, although their individual line sums are incorrect, they cancel. Please check that all line sums are correct.

2 At certain column positions in a statement line, a full point (.) and a comma (,) may be interchanged and yet still give a correct line sum. When occurring in a DATA statement a (.) for (,) substitution will cause two adjacent integer numbers to be read as a single real number and will usually generate an 'Out of Data' message. When either possible substitution occurs in a program statement, a 'Syntax Error' message will be generated.

Lines containing GOSUB, GOTO and implied GOTO statements (i.e. after an IF. . . THEN. . . statement), may cause line sums different from those in this book even though the program is PERFECTLY CORRECT.

This effect depends on the amount of memory available in the machine and causes the line sums to alter from machine to machine. All the programs in this book were created on a National CF2000 MSX computer with 16K bytes of memory. .

Sometimes you may find that a line has been typed in completely correctly, the program runs perfectly and, yet, the ChexSum is different.

This effect arises from the way MSX computers work and seems to depend on the amount of memory in a particular machine and on the operation of the line editor. At the time of publication it has not been possible to modify ChexSum to eliminate this effect reliably.

So, if your games look right and run correctly, ignore discrepancies in the ChexSum tables.

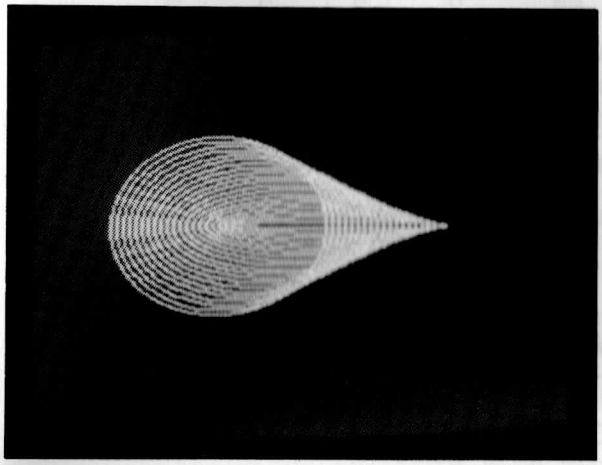
Note It is impossible for ChexSum to completely eliminate typing errors — it merely reduces the probability of them occurring to a very low level.

Pointer Corruption

Sometimes a program will run incorrectly even when thoroughly debugged by ChexSum and by hand. In this case there are two possibilities:

- The program is wrong — please write and tell us (see 'Write to Us' at the end of the book).
- A previous run has left system pointers corrupted and they continue to crash the program despite the use of the NEW command. Always reset the computer after a crash to avoid this problem.

Patterns



CLASSIFICATION: Interactive Graphics

Create fantastic patterns with circles and rectangles! This program demonstrates the powerful graphic capabilities of the MSX. Just answer the questions and watch the shapes form!

The circle patterns are formed by moving the circles horizontally, altering their shapes and sizes.

Rectangles are generated from a line joining two points, P1 and P2.

Press a key when you are finished with a pattern.

PROGRAMMING SUGGESTIONS

Here are some combinations to get you started:

CIRCLES	5Ø,	3,	−Ø.Ø5,	Ø	
RECTANGLES	6Ø,	−2,	−2,	−2,	−2

Simple additions would be sections for drawing lines instead of rectangles or even plotting simple points.

Even easier changes: change the numbers in line 2ØØ or line 1ØØ.

PROGRAM

Variables

NC	Number of circles
IN	Increment in X-directions
CA	Change in aspect ratio
CR	Change in radius
CX, CY	Centre co-ordinates
R	Radius
NR	Number of rectangles
IX, IY	Increments for P1
SX, SY	Increments for P2 (multiples of IX, IY)

Listing

Initialise

```
 5   C1 = 15
10   SCREEN 1 : KEY OFF : PRINT "▲▲▲▲▲▲▲▲▲▲▲PATTERNS▲▲▲▲▲▲▲
     ▲": PRINT "▲▲▲▲▲▲▲▲▲▲▲^^^^^^^^^": PRINT : PRINT :
     PRINT
11   PRINT : PRINT TAB( 4 ) "CIRCLES": PRINT : PRINT :
     PRINT TAB( 4 ) "RECTANGLES": PRINT : PRINT : PRINT
     TAB( 4 ) "SET▲COLOUR"
12   FOR I = 1 TO 6 : PRINT : NEXT : PRINT "Use▲cursor▲keys
     to▲move▲arrow": PRINT : PRINT "▲▲Press▲'RETURN'▲to▲sele
     ct"
13   K1 = 60 : GOSUB 19
14   X$ = INKEY$ : IF X$ = CHR$( 30 ) THEN IF K = 0 THEN
     14 ELSE K1 = 32 : GOSUB 19 : K1 = 60 : K = K - 1 :
     GOSUB 19 : GOTO 14
15   IF X$ = CHR$( 31 ) THEN IF K = 2 THEN 14 ELSE K1
     = 32 : GOSUB 19 : K1 = 60 : K = K + 1 : GOSUB 19 :
     GOTO 14
16   IF X$ <> CHR$( 13 ) THEN 14 ELSE IF K = 0 THEN 20
     ELSE IF K = 1 THEN 40 ELSE CLS : INPUT "Foreground▲(1-1
     5)"; X$ : C1 = VAL( X$ ) : IF C1 < 1 OR C1 > 15
     THEN RUN
17   INPUT "Background▲(1-15)"; X$ : K2 = VAL( X$ ) :
     IF K2 < 1 OR K2 > 15 THEN 17 ELSE INPUT "Border▲(1-15)";
     X$
18   K3 = VAL( X$ ) : IF K3 < 1 OR K3 > 15 THEN PRINT
     "Out▲of▲range": GOTO 17 ELSE COLOR C1, K2, K3 :
     GOTO 10
19   VPOKE 6356 + 96 * K, K1 : VPOKE 6357 + 96 * K, K1 :
     VPOKE 6358 + 96 * K, K1 : RETURN
20   CLS : INPUT "NUMBER▲OF▲CIRCLES"; NC
25   PRINT : INPUT "INCREMENT▲IN▲X-DIRECTION"; X$ : IN
     = VAL( X$ )
30   PRINT : INPUT "CHANGE▲IN▲ASPECT▲RATIO"; X$ : IF
     VAL( X$ ) < - 1 OR VAL( X$ ) > 1 THEN 30 ELSE CA
     = VAL( X$ )
35   PRINT : INPUT "CHANGE▲IN▲RADIUS"; CR : GOTO 100
40   CLS : INPUT "NUMBER▲OF▲RECTANGLES"; NR
45   PRINT : INPUT "INCREMENT▲IN▲X-DIRECTION"; IX : PRINT :
     INPUT "INCREMENT▲IN▲Y-DIRECTION"; IY
50   PRINT : INPUT "INCREASE▲OR▲DECREASE▲P2▲▲▲▲▲▲X-CO-ORDINA
     TES▲(1▲OR▲-1)"; SX
55   PRINT : INPUT "INCREASE▲OR▲DECREASE▲P2▲▲▲▲▲▲Y-CO-ORDINA
     TES▲(1▲OR▲-1)"; SY
90   GOTO 200
```

Circles

```
100   SCREEN 2 : CX = 90 : CY = 90 : IF CR < 0 THEN R
      = 100 ELSE IF CR > 0 THEN R = 5 ELSE R = 50
105   AR = 1
115   FOR I = 1 TO NC
120   IF CX < 0 OR CX > 255 OR CY < 0 OR CY > 192 OR R
      < = 0 OR AR < = 0 THEN 400 ELSE CIRCLE( CX, CY )
      , R, C1, 0, 6.28, AR
125   CX = CX + IN : AR = AR + CA : R = R + CR
130   NEXT
190   GOTO 400
```

Rectangles

```
200   SCREEN 2 : X1 = 100 : Y1 = 100 : X2 = 150 : Y2 = 90
210   FOR I = 1 TO NR : X1 = X1 + IX : Y1 = Y1 + IY :
      X2 = X2 + SX * IX : Y2 = Y2 + SY * IY
220   LINE( X1, Y1 ) - ( X2, Y2 ) , C1, B
230   NEXT
240   GOTO 400
```

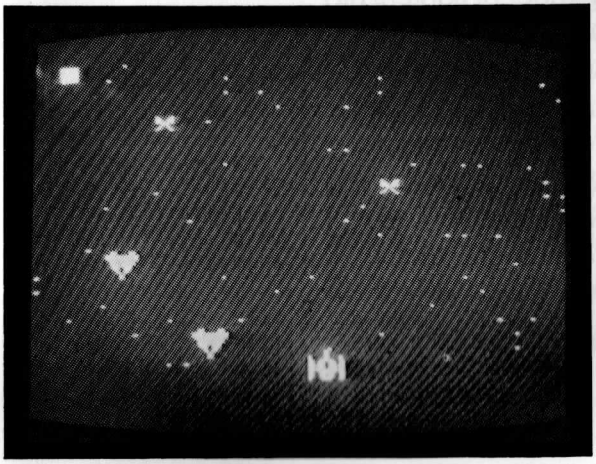
Wait

```
400   IF INKEY$ = "" THEN 400
410   GOTO 10
```

ChexSum Table

```
5     = 387
10    = 5293
11    = 5648
12    = 7919
13    = 697
14    = 6340
15    = 5494
16    = 9928
17    = 7051
18    = 6414
19    = 4582
20    = 2026
25    = 3798
30    = 7126
35    = 2552
40    = 2315
45    = 6125
50    = 5004
55    = 5005
90    = 593
100   = 4971
105   = 408
115   = 837
120   = 7932
125   = 2925
130   = 131
190   = 537
200   = 2582
210   = 6152
220   = 1645
230   = 131
240   = 537
400   = 1091
410   = 403

Total=124579
```

Astral Attack



CLASSIFICATION: Evasion Game

You have pursued an enemy fleet of transport and fighter craft into a vortex where no weapons function. Crash into the smaller fighters but avoid the big transports!

The battle is over when a fighter is allowed to escape.

Use the cursor keys to move left or right.

PROGRAMMING SUGGESTIONS

Try a spectacular explosion effect when there is a collision — lines 800-830. You can also have a different fleet formation, as line 200 controls the vertical spacing between enemy ships and line 215 the horizontal range.

PROGRAM

Variables

NF	Number of fighters hit
DT	Distance to move all ships
SF	Sprite flags for being on screen
LS	Last sprite on screen
FS	Front sprite (enemy ships)
NS	Number of ships destroyed
GE	Game ended
LD	Level of difficulty

Listing

Initialise

```
 5   SCREEN 1, 2 : KEY OFF : COLOR 15, 1, 1 : PRINT "▲▲▲▲▲▲
     ▲ASTRAL▲ATTACK"
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : A$ = ""
25   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 2 ) = A$
30   DEFUSR = 60000! : DEFUSR1 = 60118! : POKE 59996!, 10
35   FOR I = 1 TO 5 : VPOKE 6914 + 4 * I, 8 : NEXT
40   FS = 1 : PRINT : PRINT : PRINT : INPUT "Level▲of▲Diffic
     ulty▲(1-4)"; LD$ : DT = 2 * VAL( LD$ ) + 2 : IF
     DT > 10 THEN 40
50   FOR I = 1 TO 8 : PRINT : NEXT : PRINT "▲▲▲▲▲HIT▲ANY▲KEY
     ▲TO▲START"
55   D = RND( 1 ).: IF INKEY$ = "" THEN 55
60   SPRITE ON : ON SPRITE GOSUB 800
65   FOR I = 1089 TO 1095 : VPOKE I, 0 : NEXT : VPOKE
     1088, 1
70   CLS : FOR I = 1 TO 60 : VPOKE 6144 + INT( RND( 1 )
     * 700 ) , 136 : NEXT
95   TIME = 0 : PUT SPRITE 0, ( 120, 160 ) , 11
```

Control

```
100  POKE 59999!, 7 : D = USR( D )
110  GOSUB 200
120  GOSUB 300
190  GOTO 100
```

New Ship

```
200  IF TIME < 105 - 25 * LD THEN RETURN
210  K = LS + 1 : IF K = 6 THEN K = 1
212  IF SF AND 2 ^K THEN RETURN ELSE LS = K : TIME =
     0 : IF RND( 1 ) < .6 THEN K1 = 8 : K2 = 7 ELSE K1
     = 4 : K2 = 8
```

```
215   VPOKE 6914 + 4 * LS, K1 : PUT SPRITE LS, ( INT(
      RND( 1 ) * ( 150 - 20 * LD ) + 40 + 20 * LD ) ,
      0 ) , K2 : SF = ( SF ) OR 2 ^LS
220   RETURN
```

Move Ships

```
300   POKE 59999!, DT : POKE 59998!, 2 : FOTI = 1 TO 5 :
      IF SF AND 2 ^I THEN POKE 59997!, I : D = USR1( D )
      : IF VPEEK( 6912 + 4 * I ) > 190 AND( SF AND 2 ^
      I ) THEN SF = SF AND( 255 - 2 ^I ) : PUT SPRITE
      I, ( 10 + 30 * I, 200 ) : IF VPEEK( 6914 + 4 * I )
      = 8 THEN GE = 1 : GOTO 900 ELSE FS = FS + 1 : IF
      FS = 6 THEN FS = 1
305   NEXT
310   RETURN
```

Collision

```
800   SPRITE OFF : IF VPEEK( 6914 + 4 * FS ) = 4 THEN 900
810   PUT SPRITE FS, ( 10 + 30 * FS, 200 ) : SF = SF AND(
      255 - 2 ^FS ) : NS = NS + 1 : PLAY "l54m1700s10n54n32"
820   FS = FS + 1 : IF FS = 6 THEN FS = 1
830   SPRITE ON : RETURN
```

Game Over

```
900   SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲▲GAME▲OVER"
910   FOR I = 1 TO 10 : PRINT : NEXT : PRINT "You▲Destroyed";
      NS ; "▲Enemy▲Ships"
920   PRINT : PRINT : IF GE = 1 THEN PRINT "▲▲▲ENEMY▲FIGHTER
      ESCAPED!!!" ELSE PRINT "DESTROYED▲BY▲TRANSPORT▲SHIP!"
940   IF INKEY$ = "" THEN END ELSE 940
```

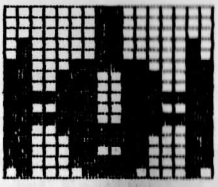
Sprite Data

```
10000   DATA 1, 1, 1, 129, 129, 199, 206, 260, 254, 206,
        255, 207, 199, 198, 199, 131, 128, 128, 128, 130,
        130, 227, 115, 115, 127, 115, 255, 243, 227, 99,
        227, 194
10010   DATA  56, 56, 255, 255, 255, 127, 63, 63, 31, 30,
        14, 6, 2, 1, 1, 1, 28, 28, 255, 255, 255, 254, 252,
        252, 248, 120, 112, 96, 64, 128, 128, 128
10020   DATA 129, 195, 231, 255, 24, 36, 66, 129
```

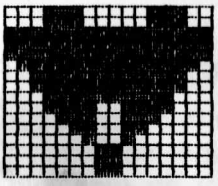
ChexSum Table

```
5     = 3132
10    = 0
11    = 0
12    = 0
15    = 3995
20    = 3996
25    = 3500
30    = 2746
35    = 1951
40    = 7427
50    = 3476
55    = 1893
60    = 926
65    = 2013
70    = 3625
95    = 1549
100   = 1493
110   = 362
120   = 207
190   = 489
200   = 1768
210   = 1865
212   = 6341
215   = 7573
220   = 143
300   = 22634
305   = 131
310   = 143
800   = 2626
810   = 6568
820   = 2113
830   = 551
900   = 1706
910   = 4509
920   = 7117
940   = 1481
10000 = 10608
10010 = 8870
10020 = 1733

Total=131260
```

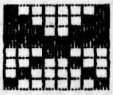
Sprite Shapes



PLAYER'S SHIP

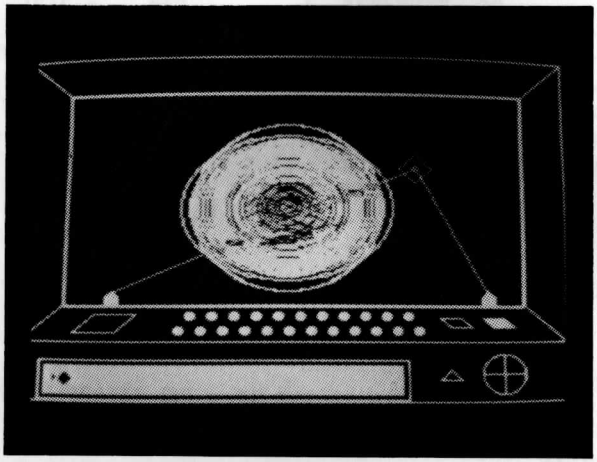


TRANSPORT



FIGHTER SHIP

Planetoid



CLASSIFICATION: Target Practice Game COLOUR ILLUST. OBC

Venture through a space corridor into a zone full of destructive planetoids. Your lasers must strike a hidden weak spot on an approaching planetoid in order to save yourself from being converted into astral dust!

Watch the spectacular graphics create a unique atmosphere in this strange space zone.

PROGRAMMING SUGGESTIONS

The planetoid's approach is controlled by lines 45Ø-49Ø and this is one area where making some modifications could be very rewarding! Make R increase in bigger steps or change the behaviour of AS to get a completely different appearance for the planetoid.

This area could also be used to implement levels of difficulty.

PROGRAM

Variables

PL	Planetoid on screen?
R	Planetoid's radius
WX, WY	Weak point co-ordinates
CX, CY	Centre of planetoid
AS	Aspect ratio
NP	Number of planetoids destroyed

Listing

Initialise

```
  1   CLS : PRINT "▲▲▲▲▲▲▲▲Hit▲a▲key"
  2   IF INKEY$ = "" THEN D = RND( 1 ) : GOTO 2
  5   COLOR 15, 1, 1 : SCREEN 2, 2
 10   REM RUN MACHINE CODE
 11   REM SUPPORT PROGRAM
 12   REM SEE APPENDICES
 15   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 0 ) = A$
 20   DEFUSR = 60000! : POKE 59996!, 15 : POKE 59999!, 8
 30   A$ = "": FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q)
      : NEXT : SPRITE$( 1 ) = A$
```

Set Up Screen

```
100   LINE( 20, 140 ) - ( 235, 20 ) , 15, B : LINE( 0,
      160 ) - ( 255, 0 ) , 15, B : LINE( 0, 192 ) - (
      255, 160 ) , 15, B : LINE( 10, 190 ) - ( 180, 170 )
      , 15, B
110   LINE( 13, 187 ) - ( 177, 173 ) , 15, B : PAINT(
      50, 186 )
120   LINE( 0, 160 ) - ( 20, 140 ) : LINE( 255, 160 )
      - ( 235, 140 ) : LINE( 0, 0 ) - ( 20, 20 ) : LINE(
      255, 0 ) - ( 235, 20 )
130   FOR I = 80 TO 180 STEP 10 : CIRCLE( I, 146 ) , 2 :
      PAINT( I, 146 ) : NEXT : FOR I = 75 TO 185 STEP
      10 : CIRCLE( I, 154 ) , 2 : PAINT( I, 154 ) : NEXT
140   DRAW "bm35,145r20g10l20e10bm195,147r10f5l10h5bm215,147r
      10f5l10h5": PAINT( 220, 150 )
150   DRAW "bm200,175g5r10h5": CIRCLE( 225, 177 ) , 10 :
      LINE( 225, 167 ) - ( 225, 187 ) : LINE( 215, 177 )
      - ( 235, 177 )
160   LINE( 21, 141 ) - ( 234, 21 ) , 15, B : DRAW "bm40,139u
      4e3f3d4bm215,139u4e3f3d4": PAINT( 42, 138 ) : PAINT(
      217, 138 )
165   IF NP = 0 THEN 170 ELSE DRAW "bm35,8": FOR I = 1
      TO NP : DRAW "g3f3e3h3br9": NEXT
170   PRESET( 17, 178 ) : PRESET( 17, 179 ) : PRESET(
      17, 180 ) : PRESET( 18, 179 )
190   IF NP <> 0 THEN 270
```

Corridor

```
200   FOR J = 1 TO 3 : X1 = 120 : X2 = 141 : Y1 = 80 :
      Y2 = 90
210   FOR I = 1 TO 10 : B$ = "n"+ STR$( I + 20 ) : PLAY
      "m299s1119xb$;": LINE( X1, Y1 ) - ( X2, Y2 ) , 15,
      B : X1 = X1 - I : X2 = X2 + I : Y1 = Y1 - I : Y2
      = Y2 + I : NEXT : LINE( 45, 130 ) - ( 211, 24 ) , 15, B
220   X1 = 120 : X2 = 141 : Y1 = 80 : Y2 = 90 : FOR I
      = 1 TO 10 : B$ = "n"+ STR$( 30 - I ) : PLAY "m299s1119x
      b$;": LINE( X1, Y1 ) - ( X2, Y2 ) , 1, B : X1 =
      X1 - I : X2 = X2 + I : Y1 = Y1 - I : Y2 = Y2 + I :
      NEXT : LINE( 45, 130 ) - ( 211, 24 ) , 1, B : NEXT
270   PUT SPRITE 0, ( 120, 70 ) , 9
280   WX = 0 : WY = 0 : STRIG( 0 ) ON : ON STRIG GOSUB 900
```

Control

```
300   A = PEEK( 59996! ) : X = VPEEK( 6913 ) : Y = VPEEK(
      6912 ) : IF X < 30 THEN POKE 59996!, PEEK( 59996! )
      AND 13 ELSE POKE 59996!, PEEK( 59996! ) OR 2
305   IF X > 209 THEN POKE 59996!, PEEK( 59996! ) AND
      7 ELSE POKE 59996!, PEEK( 59996! ) OR 8
310   IF Y < 30 THEN POKE 59996!, PEEK( 59996! ) AND 14
      ELSE POKE 59996!, PEEK( 59996! ) OR 1
315   IF Y > 100 THEN POKE 59996!, PEEK( 59996! ) AND
      11 ELSE POKE 59996!, PEEK( 59996! ) OR 4
320   D = USR( D )
330   IF PL = 0 AND RND( 1 ) < .1 THEN GOSUB 400
340   IF PL = 1 THEN GOSUB 450
390   GOTO 300
```

Introduce Planetoid

```
400   CX = INT( RND( 1 ) * 90 + 80 ) : CY = INT( RND( 1)
      * 30 + 60 ) : R = 3 : CIRCLE( CX, CY ) , R : PAINT(
      CX, CY )
410   PL = 1 : PUT SPRITE 1, ( 170, 176 ) , 1
420   WX = CX + INT( RND( 1 ) * 80 - 40 ) : WY = CY +
      INT( RND( 1 ) * 70 - 35 )
440   RETURN
```

Increase Size of Planetoid

```
450   IF RND( 1 ) < .2 THEN R = R + 2
```

```
460   AS = RND( 1 ) / 3 + .8 : CIRCLE( CX, CY ) , R, 15,
      0, 6.28, AS
470   VPOKE 6917, 160 - 3 * R
480   IF VPEEK( 6917 ) < 15 THEN 2000
490   RETURN
```

Blow Planetoid

```
700   STRIG( 0 ) OFF : FOR I = 1 TO 50 : PLAY "140m380s8n24":
      X1 = INT( RND( 1 ) * 90 - 45 + CX ) : Y1 = INT(
      RND( 1 ) * 90 - 45 + CY ) : X2 = INT( RND( 1 ) *
      200 + 25 ) : Y2 = INT( RND( 1 ) * 110 + 25 )
710   LINE( X1, Y1 ) - ( X2, Y2 ) , 1 : CIRCLE( X1, Y1 )
      , 1 : CIRCLE( X2, Y2 ) , 2 : PAINT( X2, Y2 ) : NEXT
720   PUT SPRITE 1, ( 100, 200 ) : PL = 0
730   NP = NP + 1 : CLS : GOTO 100
```

Fire Lasers

```
900   IX = VPEEK( 6913 ) + 8 : IY = VPEEK( 6912 ) + 8 :
      LINE( 44, 132 ) - ( IX, IY ) , 11 : LINE( 217, 132 )
      - ( IX, IY ) , 11
905   PLAY "110m1000s14n33"
910   LINE( 44, 132 ) - ( IX, IY ) , 1 : LINE( 217, 132 )
      - ( IX, IY ) , 1
920   IF( WX - CX ) ^2 + ( WY - CY ) ^2 > = R * R THEN
      RETURN
930   IF ABS( IX - WX ) < 15 AND ABS( IY - WY ) < 15 THEN
      700
970   RETURN
```

Game Over

```
2000  FOR I = 1 TO 100 : PLAY "164m200s14n23": X1 = INT(
      RND( 1 ) * 255 ) : Y1 = INT( RND( 1 ) * 192 ) :
      X2 = INT( RND( 1 ) * 255 ) : Y2 = INT( RND( 1 )
      * 192 ) : LINE( X1, Y1 ) - ( X2, Y2 ) : NEXT
2010  SCREEN 1 : PRINT "COLLISION▲WITH▲PLANETOID!!!":
      PRINT : PRINT : PRINT : PRINT "PLANETOIDS▲DESTROYED:";
      NP
2099  IF INKEY$ = "" THEN END ELSE 2099
```

Sprite Data

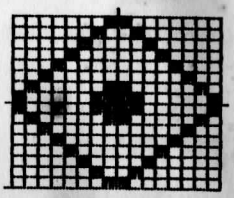
```
10000 DATA 1, 2, 4, 8, 16, 32, 65, 131, 131, 65, 32, 16,
      8, 4, 2, 1, 128, 64, 32, 16, 8, 4, 130, 193, 193,
```

```
        130, 4, 8, 16, 32, 64, 128
10010   DATA 24, 60, 126, 126, 60, 24, 0, 0
```

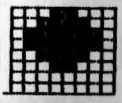
ChexSum Table

```
1       = 1671
2       = 2253
5       = 723
10      = 0
11      = 0
12      = 0
15      = 3482
20      = 2455
30      = 4010
100     = 8141
110     = 2342
120     = 5656
130     = 6409
140     = 5772
150     = 6368
160     = 6637
165     = 4450
170     = 2646
190     = 1069
200     = 3128
210     = 13684
220     = 18153
270     = 883
280     = 2719
300     = 9391
305     = 5207
310     = 5098
315     = 5158
320     = 691
330     = 2309
340     = 1155
390     = 433
400     = 6874
410     = 1554
420     = 5288
440     = 143
450     = 1928
460     = 3663
470     = 1115
480     = 1454
490     = 143
700     = 15021
710     = 4038
720     = 1397
730     = 1623
900     = 6442
905     = 1301
910     = 3119
920     = 3651
930     = 3666
970     = 143
2000    = 14040
2010    = 6422
2099    = 1362
10000   = 7039
10010   = 1398
Total= 224917
```

Sprite Shapes

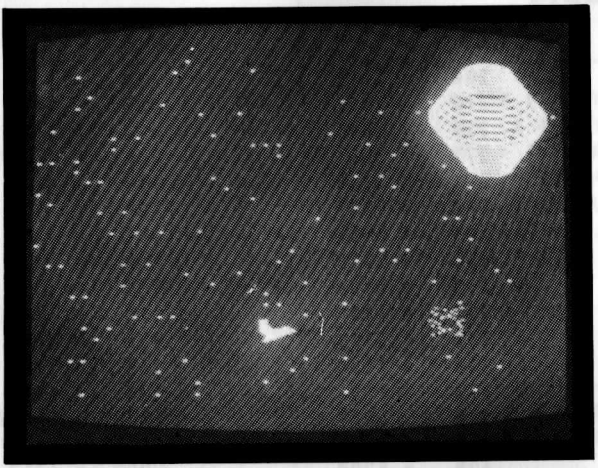


SITES



PLANETOID ON
RADER SCREEN

U.F.O.



CLASSIFICATION: Shoot-Out Game

Chase the retreating fleet of U.F.O.s in your special astral fighter equipped with rockets. You will eventually leave Earth if you don't crash into the landscape or a U.F.O. and, when you see the massive planet-ship, your time is running out.

Use the cursor keys to move up or down and to alter speed; press the <SPACE> bar to fire.

PROGRAMMING SUGGESTIONS

Make the time spent in each sceen whatever you like — S1 and S2 are set in line 6Ø.

If you're feeling energetic, you could add a third screen of your own design so that screen 2 leads to your new screen.

PROGRAM

Variables

FF	Fired rocket flag
S1, S2	Screen time limits
NU	Number of U.F.O.s hit
HE	How game ended (1 - 3)

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   COLOR 15, 1, 1 : SCREEN 2, 2
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : SPRITE$( 2 ) = CHR$( 255)
     + CHR$( 255 ) : FOR I = 14432 TO 14464 : VPOKE I,
     2 ^( INT( RND( 1 ) * 8 ) )
27   NEXT
30   DRAW "BM10,30D134R1D4R2D3R3D2R4D1R55U1R4U2R3U3R2U4R1U13
     4L21D110L1D4L1D3L1D2L1D1L25U1L1U2L1U3L1U4L1U110L20":
     PAINT( 20, 40 )
35   DRAW "BM99,30D145R20U60R30U20L30U45R55U20L75": PAINT(
     100, 100 )
40   DRAW "BM195,30R40D1R4D2R3D3R2D4R1D126L1D4L2D3L3D2L4D1L4
     0U1L4U2L3U3L2U4L1U126R1U4R2U3R3U2R4U1"
45   DRAW "BM208,51R14D1R2D2R1D98L1D2L2D1L14U1L2U2L1U98R1U2R
     2U1": PAINT( 200, 60 )
50   PUT SPRITE 1, ( 0, 5 ) , 8 : DEFUSR = 60000! : DEFUSR1
     = 60118! : POKE 59996!, 13 : POKE 59997!, 1 : POKE
     59998!, 3 : POKE 59999!, 1
55   IF INKEY$ = "" THEN D = USR1( D ) : GOTO 55
60   S1 = 60 * 60 : S2 = 60 * 30 : TIME = 0
```

Screen 1

```
100  CLS : COLOR 15, 1, 1 : DRAW "bm0,170r5d12r2u3r6d4r2u10r
     3d2r2d3r4u6r2d14r2u20r7d15r3u2r2d2r1u6r4u3r7d9r2u18r5d1
     0r3u1r2u3r2d2r3u4r2d3r3d14r2u11r6u2r2d5r2u2r2u3r3d7"
105  DRAW "r3d5r5u8r3d1r3d2r3d6r5u9r4d8r2u12r1d12r2d5r4u1r3u
     19r5d13r4d4r2u8r4d2r3d1r3d5r3u7r3d2r2d5r1u8r3u9r5d3r1d4
     r4u2r4d4r3d1r2d1r3d2r2u6r3d2r1d2r4d3r2u7r3"
110  DRAW "d8r5u3r1u2r4u1r3d1r2d3r4u6r2d3r1d5r4u16r6d8r2d5r4
     u3r4u2r2d3r2d2r2u6r4d6m255,192"
120  PUT SPRITE 1, ( 120, 70 ) , 8 : PUT SPRITE 0, ( 0,80)
     , 11
130  STRIG( 0 ) ON : ON STRIG GOSUB 400
140  SPRITE ON : ON SPRITE GOSUB 450
```

Control 1

```
200   IF VPEEK( 6912 ) > 175 THEN POKE 59996!, 12 ELSE
      POKE 59996!, 13
205   D = USR( D ) : IF VPEEK( 6912 ) > 160 AND VPEEK(
      6912 ) < 175 THEN HE = 2 : GOTO 800
210   POKE 59998!, 3 : POKE 59999!, 4 : POKE 59997!, 0 :
      D = USR1( D )
220   GOSUB 300
230   IF FF = 1 THEN GOSUB 350
240   IF TIME > 51 THEN 500
290   GOTO 200
```

Move U.F.O.

```
300   POKE 59997!, 1 : POKE 59999!, 7 : IF FF = 0 THEN
      POKE 59998!, 3 : GOTO 310
305   IF RND( 1 ) < .6 THEN POKE 59998!, 3 ELSE IF RND( 1)
      < .52 THEN POKE 59998!, 0 ELSE POKE 59998!, 2
310   D = USR1( D )
320   RETURN
```

Move Rocket

```
350   IF VPEEK( 6921 ) > 238 THEN FF = 0 : PUT SPRITE
      2, ( 101, 200 ) : RETURN
360   VPOKE 6921, VPEEK( 6921 ) + 16
370   RETURN
```

Fire Rocket

```
400   IF FF = 1 OR VPEEK( 6913 ) > 243 THEN RETURN
405   FF = 1 : PUT SPRITE 2, ( VPEEK( 6913 ) + 16, VPEEK(
      6912 ) )
410   RETURN
```

Collision

```
450   SPRITE OFF : IF FF = 0 AND ABS( VPEEK( 6913 ) -
      VPEEK( 6917 ) ) < 17 THEN HE = 1 : VPOKE 6914, 12 :
      FOR I1 = 1 TO 200 : NEXT : VPOKE 6918, 12 : FOR
      I1 = 1 TO 800 : NEXT : GOTO 800
455   IF ABS( VPEEK( 6916 ) - VPEEK( 6920 ) ) < 8 THEN
      NU = NU + 1 : VPOKE 6918, 12 : PUT SPRITE 2, ( 101,
```

```
      200 ) : PLAY "l4m2000s14n15": FOR I1 = 1 TO 500 :
      NEXT : VPOKE 6918, 4 : FF = 0 : PUT SPRITE 1, (
      0, 70 + INT( RND( 1 ) * 20 ) ) : SPRITE ON : RETURN
460   FF = 0 : PUT SPRITE 2, ( 101, 220 ) : SPRITE ON :
      RETURN
```

Screen 2

```
500   COLOR 15, 1, 1 : CLS
505   CIRCLE( 220, 5 ) , 5, 15, 0, 6, .5
510   FOR I = 1 TO 9 : CIRCLE( 220, 5 + 3 * I ) , 10 +
      2 * I, 15, 0, 6.28, .5 : NEXT : FOR I = 1 TO 9 :
      CIRCLE( 220, 33 + 3 * I ) , 28 - 2 * I, 15, 0, 6.28,
      .5 : NEXT
590   FOR I = 0 TO 255 STEP 6 : FOR J = 0 TO 190 STEP
      6 : IF RND( 1 ) < .1 THEN PSET( I, J ) , 15
595   NEXT : NEXT : TIME = 0 : SPRITE ON
```

Control 2

```
600   D = USR( D )
610   POKE 59998!, 3 : POKE 59999!, 5 : POKE 59997!, 0 :
      D = USR1( D )
620   GOSUB 300
630   IF FF = 1 THEN GOSUB 350
640   IF TIME > 52 THEN HE = 3 : GOTO 800
690   GOTO 600
```

Game Over

```
800   SCREEN 1 : PRINT "▲▲▲▲▲▲G▲A▲M▲E▲▲O▲V▲E▲R": PRINT :
      PRINT
805   PRINT "▲▲You▲have▲destroyed▲"; NU ; "of▲": PRINT
      "▲▲the▲invading▲UFO's": PRINT : PRINT
810   IF HE = 1 THEN PRINT "Collision▲with▲one▲of▲UFO's"
820   IF HE = 2 THEN PRINT "Crashed▲into▲the▲buildings!"
830   IF HE = 3 THEN PRINT "TIME-UP:▲INVADING▲FLEET▲":
      PRINT "RETURNED▲TO▲THEIR▲PLANET-SHIP"
890   IF INKEY$ = "" THEN END ELSE 890
```

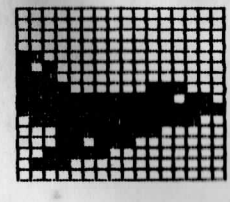
Sprite Data

```
10000   DATA 0, 0, 0, 224, 224, 224, 240, 248, 255, 255,
        127, 31, 31, 62, 112, 0, 0, 0, 0, 0, 0, 0, 0, 56,
        254, 255, 252, 224, 128, 0, 0, 0
10010   DATA 0, 0, 0, 3, 5, 15, 126, 255, 240, 127, 63,
        7, 0, 0, 0, 0, 0, 0, 0, 192, 64, 240, 102, 255,
        0, 254, 252, 224, 0, 0, 0, 0
```

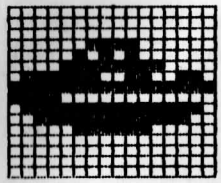
ChexSum Table

```
10    = 0
11    = 0
12    = 0
15    = 723
20    = 3995
25    = 10470
27    = 131
30    = 10098
35    = 3590
40    = 7700
45    = 5085
50    = 6967
55    = 2190
60    = 2166
100   = 16663
105   = 17809
110   = 7830
120   = 1863
130   = 1653
140   = 1126
200   = 3129
205   = 4518
210   = 3160
220   = 207
230   = 1027
240   = 1219
290   = 593
300   = 3798
305   = 6024
310   = 715
320   = 143
350   = 3185
360   = 1181
370   = 143
400   = 2227
405   = 2575
410   = 143
450   = 9393
455   = 14962
460   = 2070
500   = 572
505   = 1179
510   = 9701
590   = 4655
595   = 1275
600   = 691
610   = 3162
620   = 207
630   = 1027
640   = 1838
690   = 483
800   = 2345
805   = 5772
810   = 3815
820   = 3897
830   = 6434
890   = 1429
10000 = 7545
10010 = 6637

Total= 223135
```

Sprite Shapes

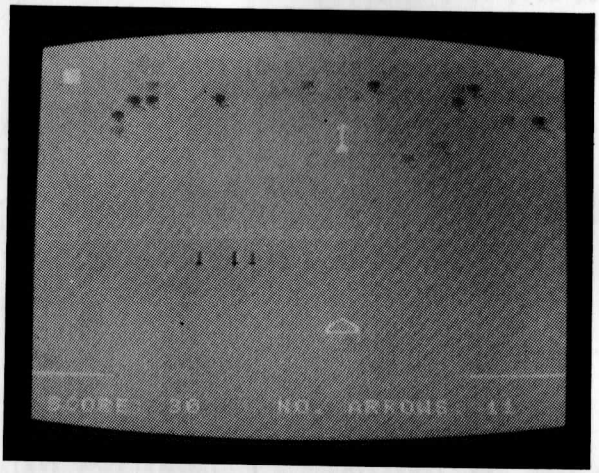


FIGHTER



U.F.O.

Balloons



CLASSIFICATION: Target Practice Game

Test your judgement by attempting to shoot arrows through a mass of balloons without popping any. A good shot scores 10 points but you only have 15 shots.

This game uses a machine-code subroutine to scroll the screen.

PROGRAMMING SUGGESTIONS

The number of shots allowed is arbitrary and is controlled by NS in line 125.

To adapt this game to an arcade-style game, make K–3 become K–5 in line 300, and change 15 to 8 in line 140.

PROGRAM

Variables

SC	Score
NS	Number of shots
S	Shots remaining
F	Fired flag
AC, AR	Arrow's column; arrow's row

Listing

Initialise

```
  5   COLOR 15, 7, 7 : SCREEN 1, 2 : KEY OFF
 10   REM RUN MACHINE CODE
 11   REM SUPPORT PROGRAM
 12   REM SEE APPENDICES
 15   FOR I = 1072 TO 1079 : READ Q : VPOKE I, Q : VPOKE
      I +.48, Q : VPOKE 80 + I, Q : NEXT : FOR I = 1088
      TO 1119 : READ Q : VPOKE I, Q : NEXT : FOR I = 1128
      TO 1135 : READ Q : VPOKE I, Q : NEXT : FOR I = 1280
      TO 1311 : READ Q : VPOKE I, Q : NEXT : FOR I = 1216
      TO 1231 : READ Q : VPOKE I, Q : NEXT
 17   VPOKE 8208, 135 : VPOKE 8209, 231 : VPOKE 8210,
      167 : VPOKE 8211, 23 : VPOKE 8212, 231
 20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 0 ) = A$ : A$ = ""
 25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 1 ) = A$ : A$ = ""
 30   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 2 ) = A$
 35   DEFUSR0 = 60000! : POKE 59996!, 10 : POKE 59999!, 8
 40   DEFUSR2 = 60350! : FOR I = 60350! TO 60377! : READ
      Q : POKE I, Q : NEXT : POKE 60346!, 33 : POKE 60347!,
      24 : POKE 60348!, 255 : POKE 60349!, 24
 55   FOR I = 6441 TO 6454 : VPOKE I, 134 : NEXT : FOR
      I = 6569 TO 6582 : VPOKE I, 134 : NEXT : VPOKE 6473,
      134 : VPOKE 6486, 134 : VPOKE 6537, 134 : VPOKE
      6550, 134 : FOR I = 1 TO 11 : PRINT : NEXT : PRINT
      TAB( 7 ) CHR$( 134 ) "▲▲BALLOONS▲▲" CHR$( 134 )
 60   IF INKEY$ = "" THEN D = RND( 1 ) : GOTO 60
```

Set Up Screen

```
100   CLS
105   FOR I = 6816 TO 6820 : VPOKE I, 23 : NEXT : FOR
      I = 6842 TO 6847 : VPOKE I, 23 : NEXT : FOR I =
      6542 TO 6790 STEP 31 : VPOKE I, 162 : NEXT
110   FOR I = 6544 TO 6808 STEP 33 : VPOKE I, 163 : NEXT :
      VPOKE 6821, 160 : VPOKE 6841, 161
115   FOR I = 6496 TO 6527 : VPOKE I, INT( RND( 1 ) *
      4 + 136 ) : NEXT
120   VPOKE 6592, 141 : VPOKE 6590, 141 : VPOKE 6600,
      141 : VPOKE 6615, 141 : VPOKE 6563, 141 : VPOKE
      6587, 141
125   NS = 15 : S = NS
```

```
130   PUT SPRITE 0, ( 117, 140 ) : PUT SPRITE 2, ( 118,
      139 )
135   GOSUB 1000
140   INTERVAL ON : ON INTERVAL = 15 GOSUB 550
150   STRIG( 0 ) ON : ON STRIG GOSUB 800
160   FOR I = 6240 TO 6303 : IF RND( 1 ) < .1 THEN VPOKE
      I, 134
165   NEXT
```

Control

```
200   D = USR( D ) : IF F = 0 THEN VPOKE 6921, VPEEK( 6913 )
205   K = VPEEK( 6913 ) : IF K < = 69 THEN POKE 59996!,
      8 ELSE IF K > = 165 THEN POKE 59996!, 2 ELSE POKE
      59996!, 10
210   IF RND( 1 ) < .03 + SC / 2000 THEN GOSUB 500
220   IF F = 1 THEN GOSUB 300
290   GOTO 200
```

Move Arrow

```
300   K = VPEEK( 6920 ) : VPOKE 6920, K - 3 : IF K > 55
      THEN RETURN
```

Check for Collision

```
310   IF K < 7 THEN PLAY "s1m200019n55n53": VPOKE 6570
      + NS - S, 152 : SC = SC + 10 : S = S - 1 : GOSUB
      1000 : F = 0 : VPOKE 6920, 139 : VPOKE 6914, 0 :
      RETURN
320   AR = INT( K / 8 ) : J = 6144 + AR * 32 + AC : K1
      = VPEEK( J ) : IF K1 <> 134 AND K1 <> 140 AND K1
      <> 144 THEN RETURN
330   INTERVAL OFF : VPOKE J, 153 : F = 0 : PLAY "l10s10m90n2
      0": VPOKE 6920, 139 : VPOKE 6921, VPEEK( 6913 )
      + 1 : VPOKE 6914, 0 : S = S - 1 : GOSUB 1000 : VPOKE
      J, 32 : INTERVAL ON : RETURN
```

Add Balloons

```
500   IF RND( 1 ) < .5 THEN K1 = 134 ELSE IF RND( 1 )
      < .5 THEN K1 = 144 ELSE K1 = 140
505   VPOKE 6207 + INT( RND( 1 ) * 6 ) * 32, K1
520   RETURN
```

Scroll Screen

```
550   D = USR2( D ) : RETURN
```

Fire Arrow

```
800   IF F = 1 THEN RETURN
802   IF S = 0 THEN 900
805   F = 1 : VPOKE 6914, 4 : VPOKE 6920, 130
810   AC = INT( VPEEK( 6921 ) / 8 ) + 1
820   RETURN
```

Game Over

```
900   ON INTERVAL = 7 GOSUB 550 : INTERVAL ON : POKE 60348!,
      223 : POKE 60349!, 26
905   PUT SPRITE 0, ( 100, 200 ) : PUT SPRITE 2, ( 100,
      200 ) : CLS : FOR I = 1 TO 30 : VPOKE 6177 + INT(
      RND( 1 ) * 700 ) , 134 : PLAY "l64s8m500n50": NEXT
910   PRINT "    Your Score is:"; SC
920   IF INKEY$ = "" THEN 930 ELSE 920
930   IF INKEY$ = "" THEN 930
990   SCREEN 1 : END
```

Update Score

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT "SCORE:";
      SC ; TAB( 13 ) ; "NO. ARROWS:"; S ; CHR$( 11 ) ;
1010  RETURN
```

Character Data

```
10000  DATA 120, 252, 252, 252, 120, 48, 26, 5
10002  DATA 0, 0, 0, 49, 123, 255, 255, 255
10004  DATA 0, 0, 131, 207, 255, 255, 255, 255
10006  DATA 48, 121, 251, 255, 255, 255, 255, 255
10008  DATA 0, 0, 0, 0, 129, 201, 255, 255
10010  DATA 81, 213, 85, 54, 28, 8, 8, 8
10012  DATA 1, 2, 4, 248, 0, 0, 0, 0
10014  DATA 128, 64, 32, 31, 0, 0, 0, 0
10016  DATA 1, 2, 4, 8, 16, 32, 64, 128
10018  DATA 128, 64, 32, 16, 8, 4, 2, 1
10020  DATA 16, 16, 16, 16, 16, 16, 56, 16
10022  DATA 146, 4, 32, 9, 128, 16, 69, 8
```

Sprite Data

```
10100   DATA 1, 6, 24, 48, 96, 64, 192, 128, 128, 64, 32,
        16, 8, 4, 2, 1, 128, 96, 24, 12, 6, 2, 3, 1, 1,
        2, 4, 8, 16, 32, 64, 128
10110   DATA 1, 6, 24, 48, 96, 192, 192, 255, 0, 0, 0, 0,
        0, 0, 0, 0, 128, 96, 24, 12, 6, 3, 3, 255, 0, 0,
        0, 0, 0, 0, 0, 0
10120   DATA 1, 3, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 5, 3,
        5, 0, 128, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 128, 64,
        128, 64
```

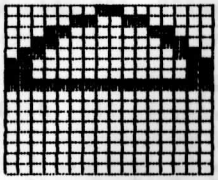
Machine-Code Data

```
10200   DATA 42, 186, 235, 43, 35, 205, 27, 235, 43, 205,
        44, 235, 35, 237, 91, 188, 235, 123, 189, 194, 194,
        235, 122, 188, 194, 194, 235, 201
```

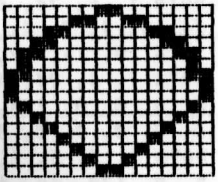
ChexSum Table

```
5     = 1244
10    = 0
11    = 0
12    = 0
15    = 15285
17    = 2786
20    = 3995
25    = 3996
30    = 3484
35    = 2481
40    = 7478
55    = 13895
60    = 2297
100   = 159
105   = 6581
110   = 3858
115   = 3409
120   = 4706
125   = 989
130   = 1935
135   = 397
140   = 2885
150   = 1496
160   = 2969
165   = 131
200   = 2410
205   = 6237
210   = 2681
220   = 904
290   = 593
300   = 2661
310   = 8917
320   = 7716
330   = 10354
500   = 5129
505   = 2334
520   = 143
550   = 917
800   = 835
802   = 865
805   = 1306
810   = 1971
820   = 143
900   = 4665
905   = 7875
910   = 2027
920   = 1526
930   = 1111
990   = 407
1000  = 4819
1010  = 143
10000 = 1645
10002 = 1448
10004 = 1635
10006 = 1850
10008 = 1377
10010 = 1314
10012 = 1028
10014 = 1225
10016 = 1205
10018 = 1229
10020 = 1408
10022 = 1358
10100 = 6404
10110 = 5544
10120 = 4771
10200 = 8518

Total= 207104
```

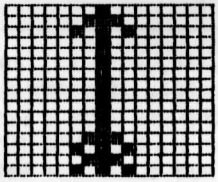
Sprite Shapes



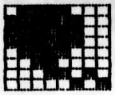
BOW: RELEASED



BOW: EXTENDED

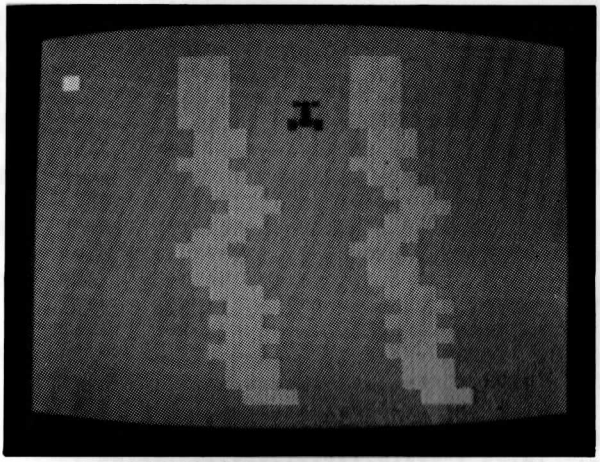


ARROW



BALLOON
(CHARACTER)

Dragster



CLASSIFICATION: Race Game

In this game, you drive as far as you can before you crash into the sides of the track or slide on an oil slick.

PROGRAMMING SUGGESTIONS

The speed is controlled by the value 9 in line 97. To make it easier, you could also widen the track; see the 'New Line' routine.

PROGRAM Variables

DT	Distance
LN$(I)	Line for printing on screen
L(I), R(I)	Left and right co-ordinates of road
SF	Sprite (oil patch) flag
C	Column of left part of road

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   SCREEN 1, 2 : KEY OFF : COLOR 15, 4, 4
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : PUT SPRITE 0, ( 123, 0 ) , 1
22   A$ = "": FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q)
     : NEXT : SPRITE$( 1 ) = A$
25   DEFUSR = 60000! : DEFUSR1 = 60118! : POKE 59996!,
     10 : POKE 59999!, 3 : POKE 59997!, 0 : POKE 59998!, 0
27   FOR I = 60300! TO 60331! : READ Q : POKE I, Q :
     NEXT : DEFUSR2 = 60300! : POKE 60296!, 223 : POKE
     60297!, 26 : POKE 60298!, 0 : POKE 60299!, 24
30   PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***": PRINT "▲HIT▲ANY▲▲***▲▲▲***
     ": PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***": PRINT "▲KEY▲TO▲▲***▲D▲*
     **": PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***"
35   PRINT "▲BEGIN▲▲▲▲***▲R▲***": PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***
     ": PRINT "▲▲▲▲▲▲▲▲▲▲***▲A▲***": PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲
     ***": PRINT "▲▲▲▲▲▲▲▲▲▲***▲G▲***": PRINT "▲▲▲▲▲▲▲▲▲▲***
     ▲▲▲***"
40   PRINT "▲▲▲▲▲▲▲▲▲▲***▲S▲***": PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***
     ": PRINT "▲▲▲▲▲▲▲▲▲▲***▲T▲***": PRINT "▲▲▲▲▲▲▲▲***▲▲▲**
     *": PRINT "▲▲▲▲▲▲▲▲▲▲***▲E▲***": PRINT "▲▲▲▲▲▲▲▲***▲▲▲*
     **": PRINT "▲▲▲▲▲▲▲▲▲▲***▲R▲***"
42   FOR I = 1 TO 5 : PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***": NEXT :
     PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***";
45   IF INKEY$ <> "" THEN 60
50   D = RND( 1 ) : D = USR1( D ) : GOTO 45
```

Set Up Screen

```
60   PRINT : PUT SPRITE 0, ( 123, 176 ) , 1 : FOR I =
     1 TO 23 : PRINT "▲▲▲▲▲▲▲▲▲▲***▲▲▲***": NEXT
65   PRINT CHR$( 11 ) : FOR I = 1 TO 6 : PRINT : NEXT :
     PRINT "▲▲▲▲+-+": FOR I = 1 TO 7 : PRINT "▲▲▲▲▲!▲!":
     NEXT : PRINT "▲▲▲▲▲+-+": FOR I = 1 TO 7 : PRINT
     "▲▲▲▲▲▲!": NEXT
70   VPOKE 6439, 133 : VPOKE 8208, 148
75   FOR I = 1 TO 800 : NEXT : VPOKE 6503, 133 : VPOKE
     8208, 164 : VPOKE 6439, 32 : FOR I = 1 TO 800 :
     NEXT : VPOKE 6503, 32 : VPOKE 6567, 133 : VPOKE
     8208, 36
```

```
80   IF INKEY$ = "" THEN 80 ELSE POKE 59999!, 2 : PLAY
     "12s8m40000n2"
85   D = USR1( D ) : IF VPEEK( 6912 ) > 25 THEN 85
90   PRINT  CHR$( 11 ) ; : FOR I = 1 TO 4 : LN$ ( I )
     = "▲▲▲▲▲▲▲▲▲▲**▲▲▲▲▲▲**▲▲▲▲▲▲▲▲▲▲▲▲▲": PRINT LN$
     ( I ) ; : PRINT "▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲";
     L ( I ) = 110 : R ( I ) = 136 : NEXT
95   POKE 59999!, 7 : C = 11 : ON SPRITE GOSUB 900 :
     SPRITE ON
97   ON INTERVAL = 9 GOSUB 300 : INTERVAL ON
98   FOR I = 152 TO 175 STEP 2 : VPOKE I, 170 : VPOKE
     I + 1, 85 : NEXT : VPOKE 8194, 228
```

Do Nothing

```
100  GOTO 100
```

Create New Line

```
200  FOR I = 4 TO 2 STEP - 1 : LN$ ( I ) = LN$ ( I -
     1 ) : R ( I ) = R ( I - 1 ) : L ( I ) = L ( I -
     1 ) : NEXT
210  I = INT( RND( 1 ) * 3 - 1 ) : C = C + I : IF C <
     2 THEN C = 2 ELSE IF C > 12 THEN C = 12
215  LN$ ( 1 ) = "▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲": MID$(
     LN$ ( 1 ) , C ) = "***▲▲▲▲▲▲▲***": L ( 1 ) = 8 *
     C + 28 : R ( 1 ) = L ( 1 ) + 48
220  IF RND( 1 ) < .05 + DT / 1000 AND SF = 0 THEN PUT
     SPRITE 1, ( L ( 1 ) + INT( RND( 1 ) * 35 + 5 ) ,
     0 ) , 1 : SF = 1
225  IF VPEEK( 6916 ) > 190 THEN PUT SPRITE 1, ( 200,
     200 ) : SF = 0
250  RETURN
```

Control

```
300  D = USR2( D )
305  GOSUB 200 : PRINT CHR$( 11 ) LN$ ( 1 ) : PLAY "110m5900
     0s10n4"
310  IF SF = 1 THEN VPOKE 6916, VPEEK( 6916 ) + 8
320  DT = DT + 1
330  GOSUB 800
340  D = USR( D )
350  RETURN
```

Check Collision

```
800   IF VPEEK( 6913 ) < L ( 4 ) OR VPEEK( 6913 ) > R
      ( 4 ) THEN 900
810   RETURN
```

Game Over

```
900   INTERVAL OFF : PLAY "l1m60000s8n20": FOR I = 6144
      TO 6303 : VPOKE I, 32 : NEXT : PRINT CHR$( 11 ) ;
      "▲▲▲▲▲▲C▲R▲A▲S▲H▲E▲D▲!▲!▲!": PRINT : PRINT "Distance:▲";
      DT ; "km"
910   IF INKEY$ = "" THEN END ELSE 910
```

Sprite Data

```
10000 DATA 57, 63, 59, 3, 3, 3, 3, 3, 3, 247, 247,
      255, 243, 240, 240, 156, 252, 220, 192, 192, 192,
      192, 192, 192, 192, 239, 239, 255, 207, 15, 15
10010 DATA 112, 236, 88, 248, 204, 0, 0, 0
```

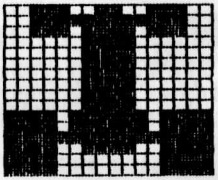
Scroll Data

```
10100 DATA 42, 136, 235, 35, 43, 205, 27, 235, 17, 32,
      0, 25, 205, 44, 235, 237, 82, 237, 91, 138, 235,
      123, 189, 194, 144, 235, 122, 188, 194, 144, 235, 201
```

ChexSum Tables

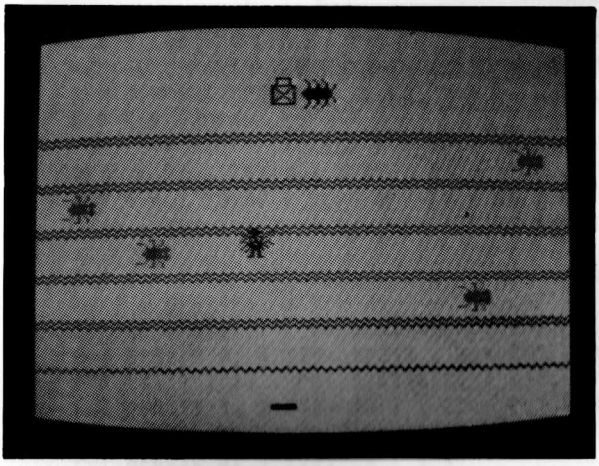
```
10    = 0
11    = 0
12    = 0
15    = 1220
20    = 4519
22    = 4010
25    = 5685
27    = 7288
30    = 9478
35    = 13017
40    = 17019
42    = 3504
45    = 1243
50    = 2186
60    = 3483
65    = 7866
70    = 1031
75    = 6078
80    = 3306
85    = 2173
90    = 11087
95    = 2355
97    = 2767
98    = 3578
100   = 489
200   = 5321
215   = 7843
220   = 6682
225   = 3066
250   = 143
300   = 716
305   = 2859
310   = 1958
320   = 817
330   = 158
340   = 691
350   = 143
800   = 2927
810   = 143
900   = 10755
910   = 1449
10000 = 9121
10010 = 1462
10100 = 9933

Total= 186129
```

Sprite Shapes



DRAG RACER

Cosmic Critters



CLASSIFICATION: Evasion Game

Harry the Hungry, Hairy Mountain Man has arrived at the domain of the Cosmic Critters in search of food. Guide Harry through the tunnels to the top where the Queen Critter guards the food. Harry must then return with a food parcel to his camp at the bottom.

Make as many trips as you can before Harry is caught by a critter.

PROGRAMMING SUGGESTIONS

Line 21Ø determines how often the closest critter moves towards Harry so, by changing this line, you can make the game as hard or easy as you like.

If you're not happy with the speed of the critters then vary the appropriate numbers in line 25Ø.

PROGRAM

Variables

CC	Closest critter
DR(I)	Directions of critters
NT	Number of complete trips
LD	Harry loaded with food?

Listing

Initialise

```
 5   CLEAR 700 : COLOR 1, 14, 14 : SCREEN 2, 2
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : FOR I = 1 TO 5 : SPRITE$( I ) = A$ : NEXT :
     A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 6 ) = A$
30   DEFUSR = 60000! : DEFUSR1 = 60118! : POKE 59996!, 15
35   PUT SPRITE 0, ( 120, 173 ) , 1 : PUT SPRITE 1, (
     255, 46 ) , 12 : PUT SPRITE 2, ( 255, 72 ) , 12 :
     PUT SPRITE 3, ( 255, 98 ) , 12 : PUT SPRITE 4, (
     255, 124 ) , 12 : PUT SPRITE 5, ( 255, 150 ) , 12 :
     PUT SPRITE 6, ( 255, 10 ) , 1
40   FOR I = 1 TO 6 : IF RND( 1 ) < .5 THEN DR ( I )
     = 1 ELSE DR ( I ) = 3
45   NEXT : CC = 5
50   B$ = "e1f2e1": A$ = "": FOR I = 1 TO 32 : A$ = A$
     + B$ : NEXT
52   DRAW "bm0,41xa$;xa$;bm0,44xa$;xa$;"
54   DRAW "bm0,67xa$;xa$;bm0,70xa$;xa$;"
56   DRAW "bm0,93xa$;xa$;"
58   DRAW "bm0,96xa$;xa$;"
60   DRAW "bm0,119xa$;xa$;"
62   DRAW "bm0,122xa$;xa$;"
64   DRAW "bm0,145xa$;xa$;"
66   DRAW "bm0,148xa$;xa$;"
68   DRAW "bm0,171xa$;xa$;"
70   GOSUB 78
75   DRAW "bm115,192r10u2110"
77   GOTO 90
78   DRAW "bm115,14r10d10l10u10f10bl10e10bl8u4r6d4":
     RETURN
79   DRAW "bm115,178r10d10l10u10f10bl10e10bl8u4r6d4":
     RETURN
90   SPRITE ON : ON SPRITE GOSUB 800
```

Control

```
100   POKE 59999!, 10 : D = USR( D ) : GOSUB 300
110   GOSUB 200
190   GOTO 100
```

Set Directions of Critters

```
200   GOSUB 250
205   I = INT( RND( 1 ) * 6 + 1 ) : IF DR ( I ) = 1 THEN
      DR ( I ) = 3 ELSE DR ( I ) = 1
210   IF RND( 1 ) < .9 - NT / 10 THEN 220 ELSE IF VPEEK(
      6913 + 4 * CC ) < VPEEK( 6913 ) THEN DR ( CC ) =
      3 ELSE DR ( CC ) = 1
220   RETURN
```

Move Critters

```
250   POKE 59999!, 9 : FOR I = 1 TO 6 : POKE 59998!, DR
      ( I ) : POKE 59997!, I : D = USR1( D ) : NEXT
260   RETURN
```

Check Move

```
300   K = VPEEK( 6912 ) : CC = INT( ( K - 20 ) / 26 )
305   IF K < 72 THEN CC = 6
320   IF K > 180 THEN K1 = 11 ELSE IF K < 10 THEN K1 =
      14 ELSE K1 = 15
325   POKE 59996!, K1
330   IF K > 25 AND K < 165 THEN RETURN
335   IF VPEEK( 6913 ) < 115 OR VPEEK( 6913 ) > 125 THEN
      RETURN
340   IF K < = 25 THEN IF LD = 1 THEN RETURN ELSE PLAY
      "l24m160s8n60n60": LD = 1 : DRAW "c14": GOSUB 78 :
      GOSUB 79 : RETURN
345   IF LD = 1 THEN PLAY "s8m200140n20": NT = NT + 1 :
      GOSUB 1000 : LD = 0 : DRAW "c1": GOSUB 79 : GOSUB
      78 : RETURN
350   RETURN
```

Game Over

```
800   FOR I = 1 TO 7 : PLAY "150m1300s10n64r10n74": NEXT
810   SCREEN 1 : PRINT "COSMIC▲CRITTERS▲STRIKE▲AGAIN!"
820   PRINT : PRINT : PRINT : IF NT > 1 THEN PRINT "BUT▲-▲Har
```

```
        ry▲the▲Mountain▲Man": PRINT "managed▲"; NT ; "trips▲for
        ▲food" ELSE IF NT = 1 THEN PRINT "AND▲-▲Poor▲Harry▲only
        ▲made▲": PRINT "▲one▲trip" ELSE PRINT "AND▲-▲Harry▲has
        no▲food!!"
  850   IF  INKEY$ = "" THEN END ELSE 850
```

Update Score

```
1000    A$ = "bm"+ STR$( NT * 4 ) + ",1": DRAW "xa$;c1d4":
        RETURN
```

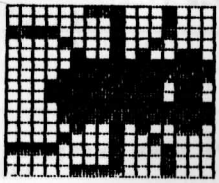
Sprite Data

```
10000   DATA 18, 10, 5, 82, 47, 21, 37, 87, 14, 23, 39,
        75, 18, 2, 2, 14, 144, 161, 74, 148, 104, 178, 164,
        234, 113, 232, 228, 208, 64, 64, 64, 112
10010.  DATA 3, 0, 240, 8, 4, 7, 15, 31, 31, 15, 15, 4,
        8, 240, 0, 3, 136, 136, 136, 144, 166, 255, 255,
        246, 246, 255, 255, 166, 144, 136, 136, 136
10020   DATA 68, 34, 17, 17, 17, 63, 127, 255, 255, 127,
        63, 17, 17, 17, 34, 68, 64, 32, 16, 17, 17, 250,
        252, 252, 252, 252, 250, 17, 17, 16, 32, 64
```

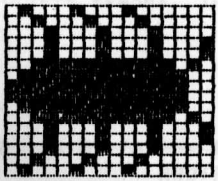
ChexSum Table

```
5     = 1169
10    = 0
11    = 0
12    = 0
15    = 3995
20    = 5120
25    = 3484
30    = 2747
35    = 10823
40    = 3710
45    = 588
50    = 3498
52    = 2587
54    = 2595
56    = 1385
58    = 1388
60    = 1435
62    = 1438
64    = 1442
66    = 1438
68    = 1441
70    = 238
75    = 1522
77    = 483
78    = 3573
79    = 3641
90    = 926
100   = 1837
110   = 362
190   = 489
200   = 410
205   = 4615
210   = 7425
220   = 143
250   = 4786
260   = 143
300   = 2650
305   = 1187
320   = 3424
325   = 786
330   = 1621
335   = 2727
340   = 5989
345   = 5783
350   = 143
800   = 2820
810   = 2873
820   = 20412
850   = 1397
1000  = 3629
10000 = 8605
10010 = 8600
10020 = 8785

Total= 162307
```

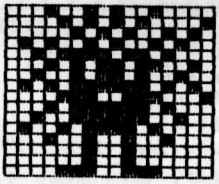
Sprite Shapes



CRITTER

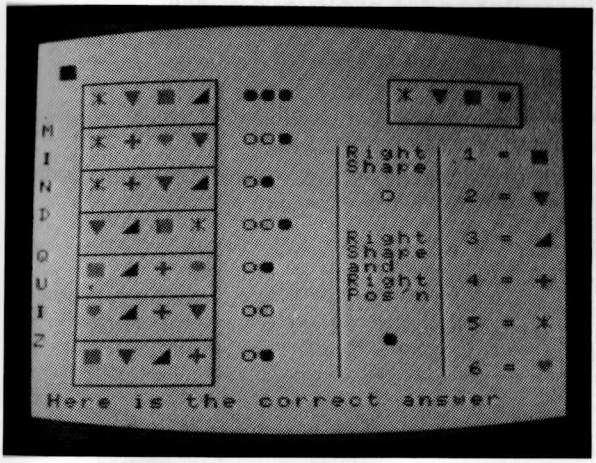


QUEEN CRITTER



HARRY

Mind Quiz



CLASSIFICATION: Memory Game COLOUR ILLUSTRATION OBC

The secret code is a sequence of four shapes chosen from six. Crack the code in less than 8 attempts and you have won the game!

A circle means that you have the right shape in the wrong position and a coloured circle signifies the right shape and position.

Instructions are included in the program.

PROGRAMMING SUGGESTIONS

If you find this game a bit difficult, you could allow extra guesses by replacing the 'Game Lost' routine with one that gives the player a choice of further guesses (to be printed as though the game had just begun) or to give up.

On the other hand, an extra symbol to choose from would increase the difficulty.

PROGRAM
Variables

AN(I)	Character codes of answer
GS(I)	Character codes of player's guess
LV	Level (Ø-6)
C	Column (Ø-3)
M$	Message
CS	Number of correct shapes
SP	Number of correct shapes with correct position

Listing

Initialise

```
  5   SCREEN 1 : KEY OFF : COLOR 1, 15, 15 : PRINT "▲▲▲▲▲▲▲▲
      MIND▲QUIZ"
 10   VPOKE 8209, 79 : VPOKE 8210, 207 : VPOKE 8212, 111 :
      VPOKE 8214, 159 : FOR I = 136 TO 184 STEP 8 : FOR
      J = 0 TO 7 : READ Q : VPOKE 8 * I + J, Q : NEXT : NEXT
 15   FOR I = 1 TO 5 : PRINT : NEXT : PRINT "▲▲Use▲the▲keys▲'
      1'▲to▲'6'▲to▲▲put▲the▲6▲possible▲sha6pes▲▲▲▲on▲the▲scre
      en.▲The▲'del'▲▲▲▲▲▲key▲can▲be▲used▲to▲change▲▲▲▲the▲last
      ▲shape."
 20   PRINT : PRINT "▲▲▲" CHR$( 136 ) "▲▲▲" CHR$( 144 )
      "▲▲▲" CHR$( 152 ) "▲▲▲" CHR$( 160 ) "▲▲▲" CHR$( 168)
      "▲▲▲" CHR$( 176 )
 30   PRINT : PRINT : PRINT "▲▲▲▲▲Hit▲a▲key▲to▲start"
 35   IF INKEY$ = "" THEN D = RND( 1 ) : GOTO 35
```

Set Up Screen

```
100   CLS : FOR I = 1 TO 6 : PRINT : NEXT : PRINT TAB( 16)
      "¦Right¦" TAB( 24 ) "1▲=▲" CHR$( 136 ) ;
105   PRINT TAB( 16 ) "¦Shape¦": PRINT TAB( 16 ) "¦▲▲▲▲▲¦"
       : PRINT TAB( 16 ) "¦▲▲" CHR$( 132 ) "▲▲¦" TAB( 24)
      "2▲=▲" CHR$( 144 ) ; : PRINT TAB( 16 ) "¦▲▲▲▲▲¦":
      PRINT TAB( 16 ) "¦▲▲▲▲▲▲¦" : PRINT TAB( 16 ) "¦Right¦"
      TAB( 24 ) "3▲=▲" CHR$( 152 ) ;
110   PRINT TAB( 16 ) "¦Shape¦": PRINT TAB( 16 ) "¦and▲▲¦":
       PRINT TAB( 16 ) "¦Right¦" TAB( 24 ) "4▲=▲" CHR$(
      160 ) ; : PRINT TAB( 16 ) "¦Pos'n¦": PRINT TAB( 16)
      "¦▲▲▲▲▲¦" : PRINT TAB( 16 ) "¦▲▲" CHR$( 133 ) "▲▲¦"
      TAB( 24 ) "5▲=▲" CHR$( 168 ) ;
115   PRINT TAB( 16 ) "¦▲▲▲▲▲¦": PRINT TAB( 16 ) "¦▲▲▲▲▲¦"
       : PRINT TAB( 16 ) "¦▲▲▲▲▲¦" TAB( 24 ) "6▲=▲" CHR$(
      176 ) CHR$( 11 ) ;
140   VPOKE 6273, 77 : VPOKE 6337, 73 : VPOKE 6401, 78 :
      VPOKE 6465, 68 : VPOKE 6561, 81 : VPOKE 6625, 85 :
      VPOKE 6689, 73 : VPOKE 6753, 90
145   FOR I = 6179 TO 6755 STEP 96 : FOR J = I + 97 TO
      I + 103 : VPOKE J, 23 : NEXT : VPOKE I, 20 : VPOKE
      I + 8, 19 : VPOKE I + 32, 22 : VPOKE I + 64, 22 :
      VPOKE I + 40, 22 : VPOKE I + 72, 22 : NEXT
150   VPOKE 6179, 24 : VPOKE 6187, 25. : VPOKE 6851, 26 :
      VPOKE 6859, 27 : FOR I = 6180 TO 6186 : VPOKE I,
      23 : NEXT
```

```
505   FOR I = 6197 TO 6261 STEP 32 : FOR J = 0 TO 6 :
      VPOKE I + J, 32 : NEXT : NEXT : FOR J = 6198 TO
      6204 : VPOKE J, 23 : NEXT : VPOKE 6197, 24 : VPOKE
      6205, 25 : VPOKE 6229, 22 : VPOKE 6261, 22 : VPOKE
      6237, 22 : VPOKE 6269, 22 : VPOKE 6293, 26 : VPOKE
      6301, 27 : FOR I = 6294 TO 6300 : VPOKE I, 23 : NEXT
510   FOR I = 1 TO 4 : PLAY "n20s8m160n30": VPOKE 6228
      + 2 * I, AN ( I ) : NEXT
515   FOR I = 1 TO 2500 : NEXT : M$ = "Hit▲any▲key▲for▲anothe
      r▲try": GOSUB 1000
520   IF INKEY$ = "" THEN 520 ELSE RUN
```

Game Won

```
900   M$ = "Superb!!!!!!!!!!!!!": GOSUB 1000 : PLAY "s14m2000l
      10n30n32n34n36n38n40n42n44m1000n46n48n49n50n51m700n52n5
      3n54n55m500n56n58n60"
910   FOR I = 1 TO 3000 : NEXT : M$ = "You▲did▲it▲in"+
      STR$( L ) + "▲tries!": GOSUB 1000 : BEEP
920   FOR I = 1 TO 3000 : NEXT : M$ = "Hit▲a▲key▲for▲another
      game": GOSUB 1000 : BEEP
930   IF INKEY$ = "" THEN 930 ELSE RUN
```

Print Message

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT M$ ; CHR$(
      11 ) ;
1010  RETURN
```

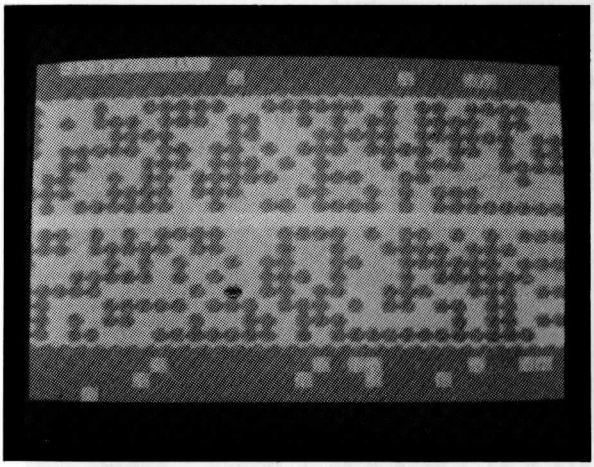
Character Data

```
10000  DATA 255, 255, 255, 255, 255, 255, 255, 255
10002  DATA 255, 255, 126, 126, 60, 60, 24, 24
10004  DATA 1, 3, 7, 15, 31, 63, 127, 255
10006  DATA 24, 24, 24, 255, 255, 24, 24, 24
10008  DATA 146, 84, 56, 16, 56, 84, 146, 0
10010  DATA 108, 254, 254, 254, 124, 56, 16, 0
10012  DATA 255, 255, 255, 255, 255, 255, 255, 255
```

ChexSum Tables

```
5     = 2741
10    = 6786
15    = 17294
20    = 6544
30    = 2730
35    = 2286
100   = 4111
105   = 14238
110   = 14677
115   = 5306
140   = 5550
145   = 9809
150   = 4124
155   = 4353
160   = 5511
165   = 3055
200   = 4410
205   = 6073
210   = 5626
215   = 6078
290   = 593
300   = 1698
305   = 2112
310   = 910
315   = 6007
320   = 1171
400   = 954
410   = 2824
415   = 980
420   = 3283
425   = 4611
430   = 977
435   = 593
500   = 8987
505   = 16070
510   = 3736
515   = 5055
520   = 1320
900   = 11357
910   = 5780
920   = 5172
930   = 1482
1000  = 2190
1010  = 143
10000 = 1888
10002 = 1654
10004 = 1326
10006 = 1515
10008 = 1469
10010 = 1645
10012 = 1888

Total=230692
```

Stream Hopper



CLASSIFICATION: Simulation Game

Life is tough for a little frog trying to cross a steam! In this game you get credit for reaching the floating log in the middle. You must traverse the stream as many times as you can, and your score is calibrated to the total number of moves you need to cross each time.

A machine-code subroutine is used to scroll the screen.

PROGRAMMING SUGGESTIONS

The flow of the stream is controlled by the value 2Ø in line 15Ø so, to speed it up, change 2Ø to 1Ø. The density of the lily-pads is determined by the value Ø.65 in line 125 and Ø.6Ø in line 135.

Why not have different floating objects in one half of the stream? You could even have the odd boat (as a sprite) floating down.

PROGRAM Variables

D	Directions of frog (Ø or 2)
PP	Player's (frog) position
NM	Number of moves
SC	Score
PR, PC	Player's row, player's column
CU	Character under frog
F1	Flag for scoring at log

Listing

Initialize

```
 5   REM RUN MACHINE CODE
 6   REM SUPPORT PROGRAM
 7   REM SEE APPENDICES
10   COLOR 4, 7, 1 : SCREEN 1, 2 : KEY OFF
15   FOR I = 60350! TO 60401! : READ Q : POKE I, Q : NEXT
20   FOR I = 1072 TO 1079 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1104 TO 1135 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1152 TO 1175 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1216 TO 1223 : READ Q : VPOKE I, Q : NEXT
25   FOR I = 1 TO 11 : PRINT : NEXT : PRINT "▲▲▲▲▲▲▲▲▲▲Hit▲a
     ▲Key"
35   PLAY "124m160s8n67n67": PRINT CHR$( 11 ) TAB( 8 )
     "STREAM-HOPPER": FOR I = 1 TO 300 : NEXT
40   VPOKE 6208 + INT( RND( 1 ) * 600 ) , 134 : IF INKEY$
     = "" THEN FOR I = 6148 TO 6170 : VPOKE I, 32 : NEXT :
     FOR I = 1 TO 300 : NEXT : GOTO 35
50   DEFUSR = 60350! : SA = 6240 : FA = 6783 : POKE 60346!,
     SA MOD 256 : POKE 60347!, SA ¥ 256 : POKE 60348!,
     FA MOD 256 : POKE 60349!, FA ¥ 256
```

Set Up Screen

```
100  CLS : FOR I = 6155 TO 6207 : VPOKE I, 141 : NEXT :
     VPOKE 6144, 141 : VPOKE 6145, 141
105  FOR I = 6208 TO 6239 : VPOKE I, 140 : NEXT : FOR
     I = 6784 TO 6815 : VPOKE I, 139 : NEXT
110  FOR I = 6816 TO 6911 : VPOKE I, 141 : NEXT
115  FOR I = 6496 TO 6525 : J = INT( RND( 1 ) * 3 + 144 )
     : VPOKE I, J : NEXT
120  GOSUB 1000
125  FOR I = 6240 TO 6464 STEP 32 : FOR J = 0 TO 31 :
     IF RND( 1 ) < .65 THEN VPOKE I + J, 138
130  NEXT : NEXT
135  FOR I = 6528 TO 6752 STEP 32 : FOR J = 0 TO 31 :
     IF RND( 1 ) < .6 THEN VPOKE I + J, 138
140  NEXT : NEXT
145  CU = 139 : PP = 6799 : VPOKE PP, 134 : PC = 15
150  ON INTERVAL = 20 GOSUB 800 : INTERVAL ON
155  FOR I = 1 TO 5 : VPOKE 6176 + INT( RND( 1 ) * 31 )
     , 152 : NEXT : FOR I = 1 TO 12 : VPOKE 6816 + INT(
     RND( 1 ) * 95 ) , 152 : NEXT
160  VPOKE 8208, 19 : VPOKE 8209, 55 : VPOKE 8210, 231 :
     VPOKE 8211, 179
```

Control

```
200   X$ = INKEY$ : IF X$ <> "" THEN GOSUB 300
210   IF RND( 1 ) < .1 THEN GOSUB 400
290   GOTO 200
```

Read Keyboard

```
300   PR = INT( ( PP - 6144 ) / 32 ) : PC = PP - 32 *
      PR - 6144
305   K = ASC( X$ ) - 27 : IF K > 0 AND K < 5 THEN PLAY
      "l24m160s8n67": ON K GOTO 310, 315, 320, 325 ELSE
      RETURN
310   IF PC = 31 THEN RETURN ELSE INTERVAL OFF : VPOKE
      PP, CU : PP = PP + 1 : GOTO 340
315   IF PC = 0 THEN RETURN ELSE INTERVAL OFF : VPOKE
      PP, CU : PP = PP - 1 : GOTO 340
320   IF PR < = 2 THEN RETURN ELSE INTERVAL OFF : VPOKE
      PP, CU : PP = PP - 32 : GOTO 340
325   IF PR > = 20 THEN RETURN ELSE INTERVAL OFF : VPOKE
      PP, CU : PP = PP + 32
340   NM = NM + 1 : CU = VPEEK( PP ) : VPOKE PP, 134 :
      INTERVAL ON : IF CU = 32 THEN 900 ELSE 500
```

Alter Screen

```
400   IF RND( 1 ) < .5 THEN I = 6240 ELSE I = 6528
405   VPOKE I + 32 * ( INT( RND( 1 ) * 8 ) ) , 138 : VPOKE
      I + 32 * ( INT( RND( 1 ) * 8 ) ) , 32
450   RETURN
```

Check Position

```
500   IF D = 2 THEN 520
505   IF PR = 2 THEN SC = SC + 200 - NM : D = 2 : F1 =
      0 : GOSUB 1000 : NM = 0 : RETURN
510   IF PR = 11 AND F1 = 0 THEN F1 = 1 : SC = SC + 50 :
      GOSUB 1000
515   RETURN
520   IF PR = 20 THEN D = 0 : SC = SC + 200 - NM : F1
      = 0 : GOSUB 1000 : NM = 0 : RETURN
525   GOTO 510
```

Scroll Screen

```
800   D = USR( D ) : IF PP < = 6783 AND PP > = 6240 THEN
      PP = PP - 1 ELSE RETURN
810   IF PP / 32 = INT( PP / 32 ) THEN 900 ELSE RETURN
```

Game Over

```
900   CLS : FOR I = 1 TO 50 : PLAY "164s14m50n50": VPOKE
      6176 + INT( RND( 1 ) * 700 ) , 134 : NEXT
910   PRINT CHR$( 11 ) ; "Score:"; SC
990   END
```

Update Score

```
1000  PLAY "12s10m700n44": PRINT "Score:"; SC ; CHR$( 11 ) ;
1010  RETURN
```

Machine Code Data

```
10000 DATA 6, 31, 42, 186, 235, 43, 35, 205, 27, 235,
      120, 254, 31, 202, 215, 235, 4, 43, 205, 44, 235,
      35, 195, 227, 235, 6, 0, 17, 31, 0, 25, 205, 44,
      235, 183, 237, 82, 237, 91, 188, 235, 123, 189,
      194, 196, 235, 122, 188, 194, 196, 235, 201
```

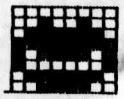
Character Data

```
10100 DATA 0, 36, 90, 255, 255, 189, 66, 60
10102 DATA 60, 124, 254, 255, 255, 255, 126, 124
10104 DATA 0, 131, 207, 255, 255, 255, 255, 255
10106 DATA 255, 255, 255, 255, 255, 247, 97, 0
10108 DATA 255, 255, 255, 255, 255, 255, 255, 255
10110 DATA 32, 255, 255, 255, 255, 255, 255, 11
10112 DATA 8, 255, 255, 255, 255, 255, 255, 1
10114 DATA 97, 251, 255, 255, 255, 255, 255, 204
10116 DATA 56, 56, 254, 254, 214, 16, 16, 16
```

ChexSum Tables

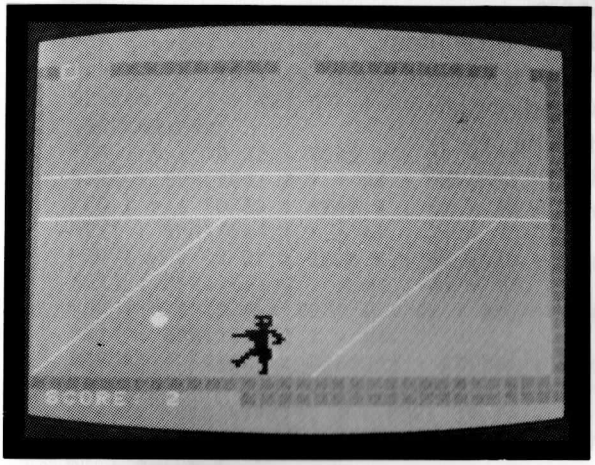
```
5      = 0
6      = 0
7      = 0
10     = 1228
15     = 2217
20     = 9678
25     = 2843
35     = 5332
40     = 7138
50     = 8130
100    = 3158
105    = 3798
110    = 1966
115    = 3923
120    = 397
125    = 4578
130    = 320
135    = 4642
140    = 320
145    = 1057
150    = 2742
155    = 7006
160    = 2050
200    = 1981
210    = 1587
290    = 593
300    = 3517
305    = 6014
310    = 4723
315    = 4671
320    = 4957
325    = 4298
340    = 5757
400    = 2613
405    = 4913
450    = 143
500    = 733
505    = 4523
510    = 3364
515    = 143
520    = 4566
525    = 643
800    = 4361
810    = 2554
900    = 4985
910    = 1659
990    = 129
1000   = 3084
1010   = 143
10000  = 21635
10100  = 1538
10102  = 1836
10104  = 1775
10106  = 1712
10108  = 1888
10110  = 1781
10112  = 1661
10114  = 1857
10116  = 1587
Total= 186447
```

Sprite Shapes



FROG
(CHARACTER)

Soccer Pro



CLASSIFICATION: Target Practice Game

How long can you keep the ball in the air? Try to kick it through the holes in the roof for extra points! You can head the ball or use the <SPACE> bar to kick it.

PROGRAMMING SUGGESTIONS

It is easy to vary ball speed — check SP in line 8Ø. But I suggest putting a few obstacles on the walls so that the ball bounces at a random angle when it strikes one of them.

You could also add to lines 7ØØ-75Ø to allow 'kneeing' of the ball or some other deflections.

PROGRAM
Variables

DR	Direction of ball (Ø - 7)
SC	Score
X, Y	Ball co-ordinates
PX	Player's X co-ordinate
SP	Speed of ball
HH	Hole hit (1 - 3)
HT	Ball hit player?

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   SCREEN 1, 3 : KEY OFF : COLOR 15, 3, 3 : PRINT "▲▲▲▲▲▲
     ▲SOCCER▲PRO"
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : A$ = ""
30   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 2 ) = A$ : A$ = ""
35   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 3 ) = A$
40   DEFUSR = 60000! : DEFUSR1 = 60118! : POKE 59996!, 10
45   FOR I = 1 TO 20 : PRINT : NEXT : PRINT "▲▲▲▲▲▲▲▲Hit▲any
     ▲Key": POKE 59997!, 3 : POKE 59998!, 3 : POKE 59999!,
     1 : PUT SPRITE 0, ( 120, 100 ) , 1 : PUT SPRITE
     3, ( 0, 20 )
47   FOR I = 1088 TO 1103 : READ Q : VPOKE I, Q : NEXT
50   D = USR1( D ) : IF VPEEK( 6925 ) = 255 THEN POKE
     59998!, 1 ELSE IF VPEEK( 6925 ) = 1 THEN POKE 59998!,
     3
55   IF INKEY$ = "" THEN 50
```

Set Up Screen

```
60   CLS : K = 134 : VPOKE 1072, 170 : VPOKE 1073, 127 :
     VPOKE 1074, 254 : VPOKE 1075, 127 : VPOKE 1076,
     254 : VPOKE 1077, 127 : VPOKE 1078, 254 : VPOKE
     1079, 85 : FOR I = 6144 TO 6175 : VPOKE I, K : NEXT :
     VPOKE 6147, 32 : VPOKE 6148, 32 : VPOKE 6159, 32 :
     VPOKE 6160, 32 : VPOKE 6172, 32 : VPOKE 6173, 32
65   FOR I = 6176 TO 6880 STEP 32 : VPOKE I, K : VPOKE
     I + 31, K : NEXT : FOR I = 6849 TO 6879 : VPOKE
     I, K : NEXT
70   FOR I = 6893 TO 6911 : VPOKE I, K : NEXT
75   VPOKE 8208, 195 : GOSUB 1000 : PUT SPRITE 0, ( 120,
     142 ) : PUT SPRITE 3, ( 11, 11 )
80   DR = 6 : SP = 7 : STRIG( 0 ) ON : ON STRIG GOSUB 800
85   K = 6817 : FOR I = 1 TO 11 : VPOKE K, 136 : VPOKE
     K + 16, 136 : K = K - 31 : NEXT : FOR I = 6465 TO
     6494 : VPOKE I, 137 : VPOKE I - 96, 137 : NEXT
95   SPRITE ON : ON SPRITE GOSUB 700
```

Control

```
100   POKE 59999!, 6 : D = USR( D )
105   IF PEEK( 59997! ) = 0 THEN VPOKE 6914, 0
110   GOSUB 200
120   GOSUB 300
190   GOTO 100
```

Move Ball

```
200   IF DR > 3 THEN SPRITE OFF : GOTO 210 ELSE POKE 59999!,
      SP : POKE 59997!, 3 : POKE 59998!, DR : D = USR1( D)
      : GOTO 220
210   POKE 59997!, 3 : POKE 59999!, SP - 1 : K1 = DR -
      4 : K2 = ( DR - 3 ) MOD 4 : POKE 59998!, K1 : SPRITE
      ON : D = USR1( D ) : POKE 59998!, K2 : D = USR1( D )
220   X = VPEEK( 6925 ) : Y = VPEEK( 6924 ) : RETURN
```

Check Position

```
300   IF Y < 8 OR Y > 192 THEN HT = 0 : PLAY "l9s14m1900n30":
      GOTO 400
305   IF X < 8 OR X > 248 THEN PLAY "s8m20019n20": GOTO 450
310   IF X > 237 THEN HT = 0 : PLAY "m30000s819n35": GOTO
      500
315   IF Y > 168 THEN 900
320   RETURN
```

Hit Roof

```
400   IF X > 19 AND X < 35 THEN HH = 1 : GOTO 420 ELSE
      IF X > 115 AND X < 131 THEN HH = 2 : GOTO 420 ELSE
      IF X > 219 AND X < 235 THEN HH = 3 : GOTO 420
410   IF DR = 7 THEN DR = 6 : K = 2 ELSE IF DR = 0 THEN
      DR = 2 : K = - 2 ELSE IF DR = 4 THEN DR = 5 : K
      = - 2 ELSE RETURN
415   VPOKE 6925, VPEEK( 6925 ) + K : RETURN
420   PLAY "m1000s814n33": FOR I = 1 TO 1000 : NEXT :
      IF HH = 1 OR HH = 3 THEN SC = SC + 50 ELSE SC =
      SC + 100
425   GOSUB 1000 : PUT SPRITE 3, ( 9, 11 ) : DR = 6 :
      RETURN
```

Hit Left Wall

```
450   IF DR = 4 THEN DR = 7 : K = - 2 ELSE IF DR = 1 THEN
      DR = 3 : K = - 2 ELSE DR = 6 : K = - 2
455   VPOKE 6924, VPEEK( 6924 ) + K
460   RETURN
```

Hit Right Wall

```
500   IF DR = 7 THEN DR = 4 : K = - 2 ELSE IF DR = 6 THEN
      DR = 5 : K = 2 ELSE DR = 1 : K = - 2
505   VPOKE 6924, VPEEK( 6924 ) + K
510   RETURN
```

Head/Kick Ball

```
700   IF HT = 1 THEN RETURN ELSE SPRITE OFF : SC = SC
      + 1 : HT = 1 : GOSUB 1000 : PX = VPEEK( 6913 ) :
      IF Y > 160 THEN 720
705   IF DR = 2 THEN IF RND( 1 ) < .5 THEN DR = 4 ELSE
      DR = 7 ELSE IF DR = 6 THEN DR = 0 ELSE IF DR = 5
      THEN DR = 0 ELSE IF DR = 1 THEN DR = 7 ELSE IF DR
      = 3 THEN DR = 4
710   GOTO 750
720   IF VPEEK( 6914 ) = 8 THEN IF X < 170 THEN DR = 4 :
      GOTO 750 ELSE DR = 1 : GOTO 750
725   IF VPEEK( 6914 ) = 4 THEN IF X > 162 THEN DR = 7 :
      GOTO 750 ELSE DR = 3 : GOTO 750
750   SPRITE ON : RETURN
```

Try to Kick

```
800   PX = VPEEK( 6913 ) : IF PX + 4 < X THEN VPOKE 6914,
      4 ELSE VPOKE 6914, 8
810   RETURN
```

Game Over

```
900   STRIG( 0 ) OFF : SPRITE OFF : CLS : PRINT "▲▲▲▲▲▲▲▲▲▲▲G
      AME▲OVER": PRINT : PRINT : PRINT
910   PRINT "YOUR▲SCORE▲WAS▲"; SC
990   IF INKEY$ <> "" THEN END ELSE 990
```

Update Score

```
1000    FOR I = 1 TO 23 : PRINT : NEXT : PRINT "SCORE:";
        SC ; CHR$( 11 ) ;
1010    RETURN
```

Sprite Data

```
10000   DATA 3, 5, 7, 3, 15, 31, 27, 19, 19, 23, 6, 6, 6,
        6, 6, 14, 128, 64, 192, 128, 224, 240, 176, 144,
        144, 208, 192, 192, 192, 192, 192, 224
10010   DATA 3, 2, 3, 1, 3, 7, 13, 9, 11, 11, 3, 1, 1, 1,
        1, 1, 192, 128, 192, 128, 224, 240, 220, 192, 224,
        224, 240, 152, 142, 4, 0, 128
10020   DATA 1, 0, 1, 0, 7, 63, 3, 1, 1, 3, 7, 12, 56, 16,
        0, 0, 224, 160, 224, 192, 240, 216, 204, 216, 192,
        224, 224, 224, 192, 64, 64, 192
10030   DATA 48, 120, 120, 48, 0, 0, 0, 0
```

Character Data

```
10100   DATA 1, 2, 4, 8, 16, 32, 64, 128
10102   DATA 0, 0, 0, 0, 0, 0, 0, 255
```

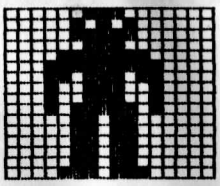
ChexSum Tables

```
10     = 0
11     = 0
12     = 0
15     = 2822
20     = 3995
25     = 3996
30     = 3997
35     = 3497
40     = 2746
45     = 9027
47     = 1899
50     = 5648
55     = 997
60     = 14735
65     = 5073
70     = 1944
75     = 2903
80     = 2532
85     = 7998
95     = 1114
100    = 1500
```

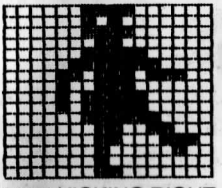
```
105    = 1955
110    = 362
120    = 207
190    = 489
200    = 6852
210    = 9132
220    = 2109
300    = 4083
305    = 3496
310    = 3521
315    = 1045
320    = 143
400    = 9463
410    = 6676
415    = 1443
420    = 6679
425    = 1829
450    = 5567
455    = 1221
460    = 143
500    = 5303
```

```
505    = 1221
510    = 143
700    = 6179
705    = 10306
710    = 629
720    = 4836
725    = 4826
750    = 551
800    = 3236
810    = 143
900    = 3953
910    = 1606
990    = 1777
1000   = 2916
1010   = 143
10000  = 8271
10010  = 7438
10020  = 7802
10030  = 1273
10100  = 1205
10102  = 1006
Total=217601
```

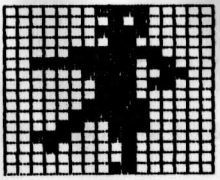
Sprite Shapes



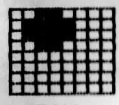
MAN: STANDING



MAN:KICKING RIGHT

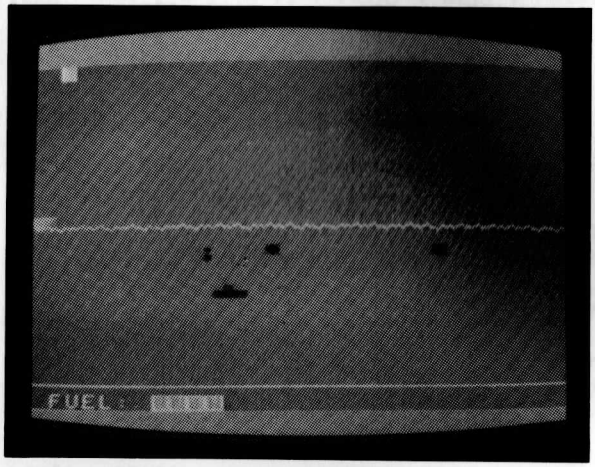


MAN: KICKING LEFT



BALL

Sea Harrier



CLASSIFICATION: Target Practice Game

Your mission: drop depth charges from your harrier jet onto a submarine without detonating a nuclear mine. You land on the aircraft carrier to refuel and the game ends when the carrier runs out of fuel.

Controls:

<UP>	Move up or take-off
<DOWN>	Move down
<LEFT>	Commence landing
<RIGHT>	Speed up
<SPACE>	Drop depth charge

PROGRAMMING SUGGESTIONS

If you find it difficult to land, line 522 contains the requirements for a good landing.

Adding several mines to the sea would make it more challenging to drop charges without blowing yourself to pieces!

PROGRAM
Variables

DC	Depth charge dropped?
FL	Fuel left
TF	Total fuel left
LD	Landing?
ST	Stopped?
RF	Refueled?
SB	Number of the sprite colliding with ship
NS	Number of submarines blown
BM	Blown mine

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   COLOR 15, 4, 7 : SCREEN 1, 2 : KEY OFF
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : PUT SPRITE 0, ( 120,
     35 ) : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : A$ = ""
30   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 2 ) = A$ : SPRITE$( 3 ) = A$ : A$ = ""
35   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 4 ) = A$ : A$ = ""
40   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 5 ) = A$ : A$ = ""
45   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 6 ) = A$
50   FOR I = 1088 TO 1095 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1152 TO 1175 : READ Q : VPOKE I, Q : NEXT :
     DEFUSR0 = 60000! : DEFUSR1 = 60118! : POKE 59996!, 15
55   FOR I = 1 TO 10 : PRINT : NEXT : PRINT "␣␣␣␣Hit␣Any␣Key
     ␣to␣Start"
60   FOR I = 1 TO 11 : D = RND( 1 ) : IF INKEY$ <> ""
     THEN 70 ELSE READ A$ : VPOKE 6153 + I, ASC( A$ ) :
     READ N : B$ = "n"+ STR$( N ) : PLAY "s1m255l15xb$;":
     FOR T = 1 TO 100 : NEXT : NEXT
65   FOR I = 1 TO 500 : NEXT : FOR I = 6154 TO 6164 :
     VPOKE I, 32 : NEXT : RESTORE 10200 : GOTO 60
```

Set Up Screen

```
70   CLS : FOR I = 6496 TO 6527 : VPOKE I, INT( RND( 1)
     * 3 + 144 ) : NEXT : FOR I = 6848 TO 6879 : VPOKE
     I, 23 : NEXT : PUT SPRITE 0, ( 0, 16 )
72   FOR I = 6848 TO 6879 : VPOKE I, 23 : NEXT : PUT
     SPRITE 0, ( 0, 16 ) , 8
75   FOR I = 1 TO 23 : PRINT : NEXT : PRINT "FUEL:";
     CHR$( 11 ) ;
80   PUT SPRITE 4, ( 0, 75 ) , 14 : PUT SPRITE 1, ( 0,
     140 ) , 1
85   K1 = INT( RND( 1 ) * 150 + 40 ) : PUT SPRITE 2,
     ( K1, 100 ) , 1 : K2 = INT( RND( 1 ) * 50 + 30 ) :
```

```
    PUT SPRITE 3, ( K1 + K2, 100 ) , 1
 90 STRIG( 0 ) ON : ON STRIG GOSUB 900 : SPRITE ON :
    ON SPRITE GOSUB 500
 95 ON INTERVAL = 180 GOSUB 1000 : INTERVAL ON : TF
    = 30 : FL = 16 : GOSUB 1100 : SB = 100
```

Control

```
100 D = USR( D ) : IF PEEK( 59997! ) = 0 THEN IF PEEK(
    59998! ) = 1 THEN LD = 1
105 K1 = VPEEK( 6912 ) : IF K1 > 75 AND K1 < 190 THEN 700
110 GOSUB 200
115 GOSUB 250
120 IF DC = 1 THEN GOSUB 300
130 IF LD = 1 THEN POKE 59996!, 13 ELSE POKE 59996!, 15
190 GOTO 100
```

Move Harrier

```
200 IF ST = 1 AND VPEEK( 6912 ) > 65 THEN RETURN
205 IF RF = 1 AND VPEEK( 6912 ) < 65 THEN RF = 0 : ST
    = 0 : LD = 0 : SPRITE ON
215 IF LD = 1 AND RF = 0 THEN K1 = 2 : K2 = 1 : GOTO 240
220 K1 = 3 : K2 = 4
240 POKE 59997!, 0 : POKE 59998!, K1 : POKE 59999!,
    K2 : D = USR1( D ) : RETURN
```

Move Submarine

```
250 K1 = RND( 1 ) : IF K1 < .5 THEN K2 = 3 ELSE IF K1
    < .75 THEN K2 = 0 ELSE K2 = 2
255 IF VPEEK( 6916 ) < 110 THEN K2 = 2 ELSE IF VPEEK(
    6916 ) > 160 THEN K2 = 0
260 POKE 59997!, 1 : POKE 59998!, K2 : POKE 59999!,
    2 : D = USR1( D )
265 RETURN
```

Move Depth Charge

```
300 POKE 59997!, 6 : POKE 59998!, 2 : POKE 59999!, 3 :
    D = USR1( D )
305 IF VPEEK( 6936 ) > 175 THEN DC = 0 : PUT SPRITE
    6, ( 100, 200 )
310 RETURN
```

Collision

```
500   SPRITE OFF : IF DC = 1 AND( VPEEK( 6912 ) < 65 OR
      VPEEK( 6912 ) > 190 OR VPEEK( 6913 ) > 16 ) THEN
      IF VPEEK( 6936 ) < 90 THEN SB = 6 : GOTO 550 ELSE
      IF VPEEK( 6936 ) < 110 THEN 650 ELSE 600
510   IF LD = 0 AND ST = 0 THEN SB = 0 : GOTO 550
515   IF LD = 1 THEN ST = 1 : LD = 0 ELSE IF ST = 1 THEN
      RETURN
520   LD = 0 : RF = 1 : TF = TF - 16 + FL : IF TL < 0
      THEN 700 ELSE IF TF > 0 THEN FL = 16 : GOSUB 1100
522   IF VPEEK( 6913 ) <> 5 AND VPEEK( 6913 ) <> 6 THEN
      SB = 0 : GOTO 550
525   SPRITE ON : RETURN
```

Blow Ship

```
550   VPOKE 6914 + SB * 4, 20 : GOSUB 1200 : FOR K1 =
      1 TO 200 : NEXT : VPOKE 6930, 20 : GOSUB 1200 :
      FOR K1 = 1 TO 1000 : NEXT : GOTO 2000
```

Blow Submarine

```
600   NS = NS + 1 : VPOKE 6919, 15 : VPOKE 6918, 20 :
      GOSUB 1200 : FOR I4 = 1 TO 1000 : NEXT : PUT SPRITE
      6, ( 100, 200 ) : PUT SPRITE 1, ( 0, 140 ) , 1 :
      VPOKE 6918, 4
640   SPRITE ON : RETURN
```

Blow Mine

```
650   VPOKE 6938, 20 : VPOKE 6939, 15 : GOSUB 1200 : FOR
      K1 = 1 TO 1000 : NEXT : BM = 1
660   GOTO 2000
```

Blow Harrier

```
700   VPOKE 6912, 70 : VPOKE 6914, 20 : GOSUB 1200 : FOR
      K1 = 1 TO 1000 : NEXT
710   GOTO 2000
```

Drop Depth Charge

```
900   IF DC = 1 THEN RETURN
905   DC = 1 : I1 = VPEEK( 6912 ) : I2 = VPEEK( 6913 ) :
      IF I2 < 8 THEN DC = 0 : RETURN ELSE PUT SPRITE 6,
      ( I2 - 8, I1 ) , 1
910   RETURN
```

Update Fuel Gauge

```
1000  FL = FL - 1 : VPOKE 6888 + FL, 32
1010  IF FL < 0 THEN 700 ELSE RETURN
```

Refresh Fuel Gauge

```
1100  FOR I3 = 6888 TO 6903 : VPOKE I3, 136 : NEXT
1120  RETURN
```

Explosion

```
1200  PLAY "l3m200s8n27"
1210  RETURN
```

Game Over

```
2000  SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲GAME▲OVER"
2010  PRINT : PRINT
2015  PRINT "Number▲of▲subs▲destroyed:"; NS : PRINT : PRINT
2020  IF SB = 0 THEN PRINT "THAT▲WAS▲A▲NASTY▲LANDING▲-▲":
      PRINT "YOU▲HAVE▲BEEN▲DEFEATED!!!": GOTO 2100
2025  IF SB = 6 THEN PRINT "YOU▲DROPPED▲A▲DEPTH▲CHARGE":
      PRINT "ON▲YOUR▲OWN▲SHIP!!": GOTO 2100
2030  IF BM = 1 THEN PRINT "NUCLEAR▲MINES▲DETONATED!!!":
      GOTO 2100
2035  IF TF < 0 OR FL < 0 THEN PRINT "▲▲▲▲▲▲▲OUT▲OF▲FUEL!!!":
      GOTO 2100
2100  IF INKEY$ = "" THEN END ELSE 2100
```

SpriteData

```
10000   DATA 0, 0, 0, 0, 0, 192, 224, 241, 255, 127, 127,
        31, 63, 124, 0, 0, 0, 0, 0, 0, 0, 0, 112, 236, 242,
        255, 238, 216, 0, 0, 0, 0
10010   DATA 0, 0, 0, 0, 0, 2, 7, 6, 7, 127, 255, 255, 127,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 128, 128, 192, 254, 255,
        255, 254, 0, 0, 0
10020   DATA 0, 90, 60, 126, 126, 60, 90, 0
10030   DATA 0, 0, 0, 0, 192, 192, 240, 240, 240, 255, 255,
        255, 255, 255, 255, 255, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 255, 254, 252, 248, 240, 224, 192
10040   DATA 0, 4, 66, 136, 2, 64, 18, 42, 10, 165, 43,
        144, 40, 1, 64, 18, 128, 16, 4, 32, 1, 72, 33, 146,
        204, 22, 200, 92, 64, 9, 0, 32
10050   DATA 40, 56, 16, 16, 56, 56, 56, 16
```

Character Data

```
10100   DATA 255, 171, 213, 171, 213, 171, 213, 255
10102   DATA 0, 0, 4, 154, 97, 0, 0, 0
10104   DATA 0, 0, 0, 13, 210, 32, 0, 0
10106   DATA 0, 0, 8, 181, 66, 0, 0, 0
```

Front Screen Data

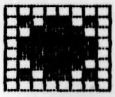
```
10200   DATA S, 30, E, 32, A, 35, "▲", 34, H, 35, A, 34,
        R, 33, R, 32, I, 33, E, 34, R, 35
```

ChexSum Table

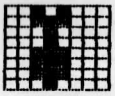
```
10    = 0        65    = 3890     130   = 2464
11    = 0        70    = 6993     190   = 489
12    = 0        72    = 2731     200   = 2093
15    = 1236     75    = 2560     205   = 3904
20    = 5019     80    = 1794     215   = 3106
25    = 3996     85    = 7394     220   = 850
30    = 4903     90    = 2947     240   = 3703
35    = 3995     95    = 5410     250   = 4849
40    = 3996     100   = 4651     255   = 3840
45    = 3500     105   = 2960     260   = 3303
50    = 7907     110   = 362      265   = 143
55    = 3649     115   = 410      300   = 3164
60    = 11102    120   = 973      305   = 2913
```

```
310   = 143
500   = 11347
510   = 2469
515   = 2913
520   = 5593
522   = 3852
525   = 551
550   = 6292
600   = 6979
640   = 551
650   = 3207
660   = 603
700   = 2669
710   = 603
900   = 903
905   = 5833
910   = 143
1000  = 1865
1010  = 1360
1100  = 2188
1120  = 143
1200  = 1130
1210  = 143
2000  = 2155
2010  = 350
2015  = 3582
2020  = 6960
2025  = 6022
2030  = 3680
2035  = 3698
2100  = 1366
10000 = 7050
10010 = 6280
10020 = 1411
10030 = 8226
10040 = 7419
10050 = 1411
10100 = 1873
10102 = 1110
10104 = 1156
10106 = 1112
10200 = 3662

Total=257202
```

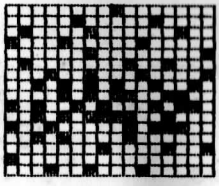
Sprite Shapes



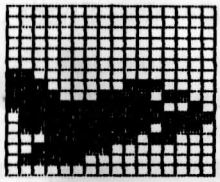
MINE



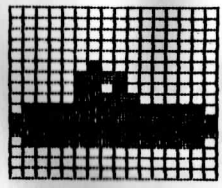
DEPTH CHARGE



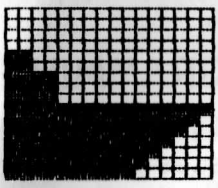
EXPLOSION



HARRIER

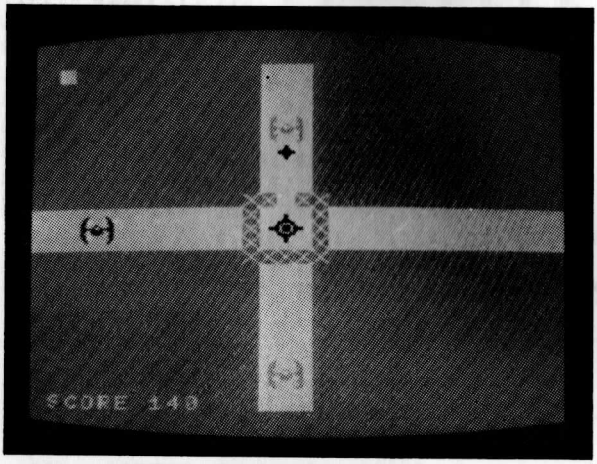


SUBMARINE



SHIP

Robot Raiders



CLASSIFICATION: Shoot-Out

Swarms of suicidal robot ships are attacking your gun ship. You must use the rotating cannon to blast the raiders before they reach the ship. Some of the robot ships disappear at random, so they are worth more points than the others.

Use the <I> key to rotate the cannon and the <SPACE> bar to fire.

PROGRAMMING SUGGESTIONS

You will notice that the speed of the bomb and the raiders increases as the game proceeds. This effect is caused by line 155 which sets the distance each sprite is moved. Try changing the expression that is poked into 59999 to alter the overall speed of the game.

PROGRAM

Variables

R, SC	Round number; score
D(I)	Direction for path I
P	Path number (1 - 4)
F	Fired?
SM, S(I)	Sprite moved; sprite I there?
AB	All ships blown?

Listing

Initialise

```
 1   TIME = 0
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   COLOR 15, 4, 7 : SCREEN 1, 2 : CLS : KEY OFF : PRINT
     "▲▲R▲O▲B▲O▲T▲▲R▲A▲I▲D▲E▲R▲S": PRINT : PRINT : PRINT :
     PRINT
20   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : SPRITE$( 2 ) = A$ : SPRITE$(
     3 ) = A$ : SPRITE$( 4 ) = A$ : A$ = ""
30   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 5 ) = A$ : A$ = ""
35   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 6 ) = A$
40   DEFUSR = 60118!
45   D ( 1 ) = 2 : D ( 2 ) = 3 : D ( 3 ) = 0 : D ( 4 )= 1
50   INPUT "▲▲ARE▲YOU▲READY▲TO▲START"; X$
60   P = 1
```

Set Up Screen

```
100  CLS : COLOR 11, 6, 6
105  FOR I = 6158 TO 6894 STEP 32 : FOR J = 0 TO 2 :
     VPOKE I + J, 255 : NEXT : NEXT
110  FOR I = 6464 TO 6559 : VPOKE I, 255 : NEXT
115  FOR I = 6445 TO 6449 : VPOKE I, 28 : NEXT : FOR
     I = 6464 TO 6528 STEP 32 : VPOKE I + 13, 28 : VPOKE
     I + 17, 28 : NEXT
120  FOR I = 6573 TO 6577 : VPOKE I, 28 : NEXT
125  PUT SPRITE 5, ( 116, 83 ) , 1 : GOSUB 1000
```

New Round

```
150  AB = 0 : R = R + 1 : PUT SPRITE 1, ( 116, 0 ) ,
     8 : PUT SPRITE 2, ( 0, 83 ) , 1 : PUT SPRITE 3,
     ( 116, 192 ) , 8 : PUT SPRITE 4, ( 255, 83 ) , 1
155  POKE 59999!, INT( R / 3 ) + 1
160  GOSUB 350 : STRIG( 0 ) ON :.ON STRIG GOSUB 320
170  SPRITE ON : ON SPRITE GOSUB 500
175  FOR I = 1 TO 4 : S ( I ) = 1 : NEXT
```

Control

```
200   IF F = 1 THEN 220
210   GOSUB 300
220   IF F = 1 THEN GOSUB 400
230   SPRITE OFF : GOSUB 450 : SPRITE ON
235   ST$ = STR$( TIME ) : DG = VAL( RIGHT$( ST$, 1 ) )
240   IF DG > 1 OR RND( 1 ) < .95 THEN 200
245   IF RND( 1 ) < .5 THEN PUT SPRITE 2, ( 116, 200 ) :
      S ( 2 ) = 0 : GOTO 260
250   PUT SPRITE 4, ( 200, 200 ) : S ( 4 ) = 0
260   AB = 1 : FOR I = 1 TO 4 : IF S ( I ) = 1 THEN AB = 0
265   NEXT : GOTO 200
```

Read Keyboard

```
300   KP$ = INKEY$ : IF KP$ <> "i" AND KP$ <> "I" THEN
      RETURN
305   P = P + 1 : IF P = 5 THEN P = 1
310   GOSUB 350 : RETURN
320   IF F = 1 THEN RETURN
322   PLAY "l19m380s10n37"
325   F = 1 : ON P GOTO 330, 335, 340, 345
330   PUT SPRITE 0, ( 120, 70 ) , 1 : RETURN
335   PUT SPRITE 0, ( 103, 87 ) , 1 : RETURN
340   PUT SPRITE 0, ( 120, 103 ) , 1 : RETURN
345   PUT SPRITE 0, ( 137, 87 ) , 1 : RETURN
```

Print Bunker

```
350   VPOKE 6447, 28 : VPOKE 6509, 28 : VPOKE 6575, 28 :
      VPOKE 6513, 28
360   ON P GOTO 365, 370, 375, 380
365   VPOKE 6447, 255 : RETURN
370   VPOKE 6509, 255 : RETURN
375   VPOKE 6575, 255 : RETURN
380   VPOKE 6513, 255 : RETURN
```

Move Bomb

```
400   TP = PEEK( 59999! ) : POKE 59997!, 0 : POKE 59999!, 2
405   POKE 59998!, P - 1 : D = USR( D ) : POKE 59999!, TP
410   ON P GOTO 415, 420, 425, 430
415   IF VPEEK( 6912 ) < 5 THEN F = 0 : PUT SPRITE 0,
      ( 200, 200 ) : RETURN ELSE RETURN
```

```
420   IF VPEEK( 6913 ) < 5 THEN F = 0 : PUT SPRITE 0,
      ( 200, 200 ) : RETURN ELSE RETURN
425   IF VPEEK( 6912 ) > 187 THEN F = 0 : PUT SPRITE 0,
      ( 200, 200 ) : RETURN ELSE RETURN
430   IF VPEEK( 6913 ) > 250 THEN F = 0 : PUT SPRITE 0,
      ( 200, 200 ) : RETURN ELSE RETURN
```

Move Enemy

```
450   SM = SM + 1 : IF SM = 5 THEN SM = 1
455   IF AB = 1 THEN 150
460   IF  S ( SM ) = 0 THEN 450
465   POKE 59997!, SM : POKE 59998!, D ( SM ) : D = USR( D )
470   ON SM GOTO 475, 480, 485, 490
475   IF VPEEK( 6916 ) > 62 THEN 600 ELSE RETURN
480   IF VPEEK( 6921 ) > 95 THEN 600 ELSE RETURN
485   IF VPEEK( 6924 ) < 111 THEN 600 ELSE RETURN
490   IF VPEEK( 6929 ) < 145 THEN 600 ELSE RETURN
```

Blow Ship

```
500   STRIG( 0 ) OFF : SPRITE OFF : F = 0 : IF P = 1 OR
      P = 3 THEN SC = SC + 20 ELSE SC = SC + 100
510   GOSUB 1000 : S ( P ) = 0 : PUT SPRITE P, ( 10, 200 )
      : VPOKE 6914, 24
520   PLAY "t255l64r64m70s10n26n2"
530   FOR T = 1 TO 150 : NEXT : VPOKE 6914, 0 : PUT SPRITE
      0, ( 200, 200 )
535   AB = 1 : FOR I = 1 TO 4 : IF S ( I ) = 1 THEN AB = 0
540   NEXT : STRIG( 0 ) ON : SPRITE ON : RETURN
```

Game Over

```
600   STRIG( 0 ) OFF
630   CLS : SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲GAME▲OVER": PRINT :
      PRINT : PRINT : PRINT : PRINT "▲▲▲▲▲YOUR▲SCORE▲WAS▲";
      SC
640   END
```

Update Score

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT "SCORE";
      SC ; CHR$( 11 ) ; : RETURN
```

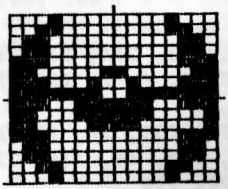
Sprite Data

```
10000   DATA 24, 24, 60, 255, 255, 60, 24, 24
10010   DATA 167, 48, 96, 96, 224, 193, 226, 254, 231, 199,
        195, 224, 96, 96, 48, 16, 8, 12, 6, 6, 7, 131, 71,
        127, 231, 227, 195, 7, 3, 3, 6, 4
10020   DATA 1, 1, 1, 3, 15, 12, 27, 251, 251, 27, 12, 15,
        3, 1, 1, 1, 128, 128, 128, 192, 240, 48, 216, 223,
        223, 216, 48, 240, 192, 128, 128, 128
10030   DATA 128, 4, 64, 0, 144, 5, 40, 134, 67, 177, 68,
        16, 68, 0, 128, 16, 4, 0, 129, 16, 6, 129, 146,
        34, 25, 162, 16, 65, 0, 8, 0, 129
```

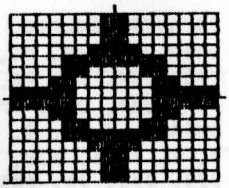
ChexSum Table

```
1      = 461
10     = 0
11     = 0
12     = 0
15     = 4659
20     = 4011
25     = 6939
30     = 3996
35     = 3484
40     = 972
45     = 2088
50     = 2247
60     = 339
100    = 582
105    = 3564
110    = 1874
115    = 4969
120    = 1777
125    = 1395
150    = 5882
155    = 1832
160    = 1945
170    = 1176
175    = 1499
200    = 938
210    = 207
220    = 1012
230    = 1278
235    = 2960
240    = 2330
245    = 3553
250    = 1581
260    = 2624
265    = 784
300    = 3338
305    = 1813
310    = 460
320    = 835
322    = 1260
325    = 1663
330    = 1088
335    = 1081
340    = 1121
345    = 1376
350    = 2154
360    = 1360
365    = 821
370    = 885
375    = 949
380    = 889
400    = 3024
405    = 2815
410    = 1572
415    = 3989
420    = 3399
425    = 3574
430    = 3639
450    = 2155
455    = 939
460    = 1189
465    = 2849
470    = 1914
475    = 1756
480    = 1797
485    = 1822
490    = 1875
500    = 5300
510    = 2587
520    = 1911
530    = 2681
535    = 2624
540    = 1513
600    = 766
630    = 5272
640    = 129
1000   = 3084
10000  = 1519
10010  = 8179
10020  = 8480
10030  = 7606

Total= 184010
```

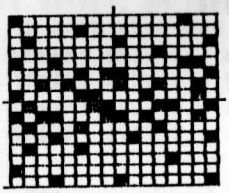
Sprite Shapes



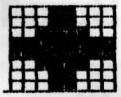
ENEMY SHIP



PLAYER'S GUNSHIP

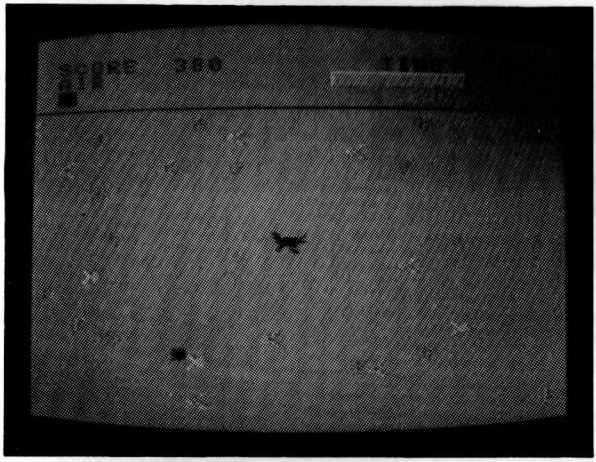


EXPLOSION



BOMB

Alligator!



CLASSIFICATION: Time-limit Game

Help the starving alligator to find food by guiding it to plants, exotic fish and the delicious yellow fish. All of the yellow fish must be eaten and the alligator must reach the right hand side of the screen within the given time limit.

If you forget that alligators can't stay under water forever then the poor thing will drown! Keep an eye on the air gauge at the top of the screen.

PROGRAMMING SUGGESTIONS

In line 35, a number is poked into 59999 to control the speed of the alligator. Change it to 2 for a hard game, 4 to make it easy.

What if the alligator had bigger lungs? Increase the number 20 in line 165 and the alligator will be able to stay under for longer.

PROGRAM

Variables

AI	Air left initially
TM	Overall time used
AR	Air left per dive
PN	Sprite Ø pattern number
R, SC	Round number, score
FL	Flag: air refreshed?
NM	Number of alligators left
TL	Time left to reach other side

Listing

Initialise

```
10    REM RUN MACHINE CODE
11    REM SUPPORT PROGRAM
12    REM SEE APPENDICES
15    KEY OFF : COLOR 1, 4, 4 : SCREEN 1, 2 : PRINT "▲▲▲▲▲▲▲▲
      ▲▲ALLIGATOR!" : PRINT : PRINT : PRINT : PRINT
20    FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25    FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : FOR I = 1 TO 4 : SPRITE$( I ) = A$ : NEXT :
      A$ = ""
30    FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 5 ) = A$
35    NM = 3 : DEFUSR = 60000! : POKE 59996!, 13 : POKE
      59999!, 3 : SPRITE ON : ON SPRITE GOSUB 500
40    FOR I = 60350! TO 60401! : READ Q : POKE I, Q :
      NEXT : DEFUSR1 = 60350! : POKE 60346!, 96 : POKE
      60347!, 24 : POKE 60348!, 255 : POKE 60349!, 26
45    FOR I = 1088 TO 1280 STEP 64 : FOR J = 0 TO 7 :
      READ Q : VPOKE I + J, Q : NEXT : NEXT
50    PRINT "▲▲▲▲▲HIT▲ANY▲KEY▲TO▲START"
55    IF INKEY$ = "" THEN 55
```

New Round

```
100   CLS : IF NM = 0 THEN 600
105   FOR I = 6240 TO 6271 : VPOKE I, 23 : NEXT
110   FOR I = 6274 TO 6882 STEP 32 : FOR J = 0 TO 29 :
      K = 10 * ( TIME / 10 - INT( TIME / 10 ) ) : IF RND(
      1 ) < .95 THEN 140
115   IF K < 1 THEN VPOKE I + J, 144 : GOTO 140
120   IF K < 6 THEN VPOKE I + J, 136 : GOTO 140
125   VPOKE I + J, 152
140   NEXT : NEXT
145   K = K * 3 + 80 : PUT SPRITE 1, ( 30, K + 30 ) ,
      11 : PUT SPRITE 2, ( 80, K + 77 ) , 11 : PUT SPRITE
      3, ( 120, K + 20 ) , 11 : PUT SPRITE 4, ( 200, K
      + 60 ) , 11
155   R = R + 1
160   PUT SPRITE 0, ( 0, 80 ) , 1
165   TIME = 0 : AI = 20 - R : TM = 0 : GOSUB 1000 : PRINT
      CHR$( 11 ) : PRINT "AIR": GOSUB 1100
170   ON INTERVAL = 15 GOSUB 650 : INTERVAL ON
180   VPOKE 8209, 36 : VPOKE 8210, 20 : VPOKE 8211, 148 :
      VPOKE 8212, 164
```

Control

```
200   D = USR( D ) : SPRITE ON : IF PN = 0 THEN PN = 20
      ELSE PN = 0
205   GOSUB 1000 : VPOKE 6914, PN : IF AR = AI THEN 700
210   IF TL < 1 THEN 600
220   IF VPEEK( 6912 ) < 20 THEN GOSUB 400 : POKE 59996!,
      12 : GOTO 230 ELSE IF VPEEK( 6912 ) < 185 THEN POKE
      59996!, 13 ELSE POKE 59996!, 9
225   FL = 0
230   IF VPEEK( 6913 ) < 253 THEN 240
235   IF VPEEK( 6916 ) = 200 AND VPEEK( 6920 ) = 200 AND
      VPEEK( 6924 ) = 200 AND VPEEK( 6928 ) = 200 THEN
      100 ELSE CLS : INTERVAL OFF : FOR I = 0 TO 5 : PUT
      SPRITE I, ( 30 * I, 200 ) : NEXT : PRINT "▲▲▲YOU▲MISSED
      ▲SOME▲FISH!!!!": PRINT : PRINT "▲You▲Have"NM - 1
      "Alligators▲Left": FOR T = 1 TO 2500 : NEXT : NM
      = NM - 1 : GOTO 100
240   K1 = INT( ( VPEEK( 6913 ) + 14 ) / 8 ) + 32 * (
      INT( ( VPEEK( 6912 ) + 4 ) / 8 ) ) + 6144 : IF VPEEK(
      K1 ) = 32 THEN 200
250   IF VPEEK( K1 ) = 152 THEN GOSUB 450
260   IF VPEEK( K1 ) = 144 THEN 550
270   IF VPEEK( K1 ) = 136 THEN GOSUB 300
280   GOTO 200
```

Eat Plant

```
300   INTERVAL OFF : SC = SC + 10 : VPOKE K1, 32 : GOSUB
      1000
310   INTERVAL ON : RETURN
```

Alligator Breathes

```
400   IF FL = 1 THEN 420
410   TM = TM + INT( TIME / 60 ) : TIME = 0 : GOSUB 1100 :
      FL = 1
420   RETURN
```

Eat Exotic Fish

```
450   INTERVAL OFF : SC = SC + 20 : VPOKE K1, 32 : GOSUB
      1000
455   PLAY "l54m1200s10n54"
460   INTERVAL ON : RETURN
```

Eat Yellow Fish

```
500   SPRITE OFF : I1 = VPEEK( 6913 ) : IF I1 < 39 THEN
      510 ELSE IF I1 < 89 THEN 520 ELSE IF I1 < 129 THEN
      530 ELSE GOTO 540
510   IF I1 > 22 THEN RETURN
512   PUT SPRITE 1, ( 30, 200 ) : GOTO 549
520   IF I1 > 72 THEN RETURN
522   PUT SPRITE 2, ( 80, 200 ) : GOTO 549
530   IF I1 > 112 THEN RETURN
532   PUT SPRITE 3, ( 120, 200 ) : GOTO 549
540   IF I1 > 192 THEN RETURN
542   PUT SPRITE 4, ( 200, 200 )
549   SC = SC + 150 : PLAY "124m160s8n27n27": GOSUB 1000 :
      RETURN
```

Eat Mine

```
550   INTERVAL OFF : FOR I = 1 TO 4 : COLOR 1, 15, 15 :
      PLAY "m8000s112n30": FOR J = 1 TO 150 : NEXT : COLOR
      1, 4, 4 : FOR J = 1 TO 150 : NEXT : NEXT
555   CLS : FOR I = 0 TO 5 : PUT SPRITE I, ( 30 * I, 200 )
      : NEXT : PRINT "▲LAND▲MINES▲GIVE▲ALLIGATORS▲": PRINT
      "▲VERY▲BAD▲INDIGESTION!!!!"
560   PRINT : PRINT : PRINT : PRINT "▲You▲Have"NM - 1
      "Alligators▲Left": FOR J = 1 TO 2500 : NEXT
580   NM = NM - 1 : GOTO 100
```

Game Over

```
600   SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲▲GAME▲OVER": PRINT :
      PRINT : PRINT : PRINT
610   PRINT "▲▲▲▲▲Your▲Score▲was▲"; SC
620   IF INKEY$ = "" THEN END ELSE 620
```

Scroll

```
650   D = USR1( D ) : RETURN
```

Alligator Drowns

```
700   CLS : INTERVAL OFF : FOR I = 0 TO 5 : PUT SPRITE
      I, ( 30 * I, 200 ) : NEXT : PRINT "▲▲▲▲▲▲ALLIGATOR▲DROW
      NED"
```

```
705   PRINT : PRINT : PRINT : PRINT "▲You▲Have"; NM -
      1 ; "Alligators▲Left"
710   PLAY "l1m59000s8n2": FOR T = 1 TO 2500 : NEXT
720   NM = NM - 1 : GOTO 100
```

Update Score/Time

```
1000  AR = INT( TIME / 60 ) : VPOKE 6183 + AR, 32
1010  TL = 50 - TM - AR : PRINT CHR$( 11 ) ; "SCORE▲";
      SC ; TAB( 19 ) ; "TIME:▲"; TL
1020  RETURN
```

Refresh Air Gauge

```
1100  FOR I = 6183 TO 6183 + AI : VPOKE I, 160 : NEXT :
      RETURN
```

Sprite Data

```
10000 DATA 0, 0, 0, 0, 192, 96, 96, 119, 63, 63, 31, 15,
      24, 48, 48, 0, 0, 0, 0, 0, 0, 1, 2, 180, 248, 255,
      240, 224, 96, 192, 96, 48
10010 DATA 128, 196, 110, 63, 63, 110, 196, 128
10020 DATA 0, 0, 0, 128, 192, 192, 96, 119, 63, 63, 31,
      15, 24, 48, 97, 97, 0, 0, 0, 0, 0, 0, 16, 184, 255,
      255, 240, 224, 96, 192, 128, 128
```

Scroll Data

```
10100 DATA 6, 31, 42, 186, 235, 43, 35, 205, 27, 235,
      120, 254, 31, 202, 215, 235, 4, 43, 205, 44, 235,
      35, 195, 227, 235, 6, 0, 17, 31, 0, 25, 205, 44,
      235, 183, 237, 82, 237, 91, 188, 235, 123, 189,
      194, 196, 235, 122, 188, 194, 196, 235, 201
```

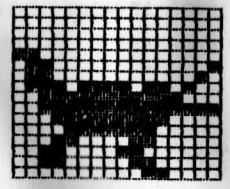
Character Data

```
10200 DATA 146, 84, 185, 82, 181, 24, 16, 40
10202 DATA 0, 90, 60, 126, 126, 60, 90, 0
10204 DATA 3, 6, 108, 176, 240, 108, 6, 3
10206 DATA 255, 255, 255, 255, 255, 255, 255, 255
```

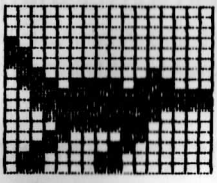
ChexSum Table

```
10    = 0
11    = 0
12    = 0
15    = 4021
20    = 3995
25    = 5137
30    = 3483
35    = 4477
40    = 7435
45    = 3576
50    = 2010
55    = 1002
100   = 1107
105   = 1640
110   = 7003
115   = 2170
120   = 2155
125   = 814
140   = 320
145   = 7496
155   = 673
160   = 780
165   = 4167
170   = 2876
180   = 1936
200   = 3166
205   = 2203
210   = 907
220   = 7227
225   = 409
230   = 1726
235   = 26305
240   = 7006
250   = 1778
260   = 1476
270   = 1602
280   = 593
300   = 3161
310   = 1304
400   = 968
410   = 3281
420   = 143
450   = 3155
455   = 1304
460   = 1304
500   = 6270
510   = 905
512   = 1366
520   = 961
522   = 1422
530   = 1001
532   = 1466
540   = 1081
542   = 1016
549   = 3260
550   = 7556
555   = 8369
560   = 5403
580   = 1392
600   = 2594
610   = 2161
620   = 1416
650   = 916
700   = 6357
705   = 4031
710   = 2465
720   = 1392
1000  = 2267
1010  = 4514
1020  = 143
1100  = 2334
10000 = 7116
10010 = 1790
10020 = 7901
10100 = 21635
10200 = 1579
10202 = 1411
10204 = 1403
10206 = 1888

Total= 254072
```

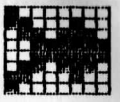
Sprite Shapes



ALLIGATOR: MOUTH OPEN

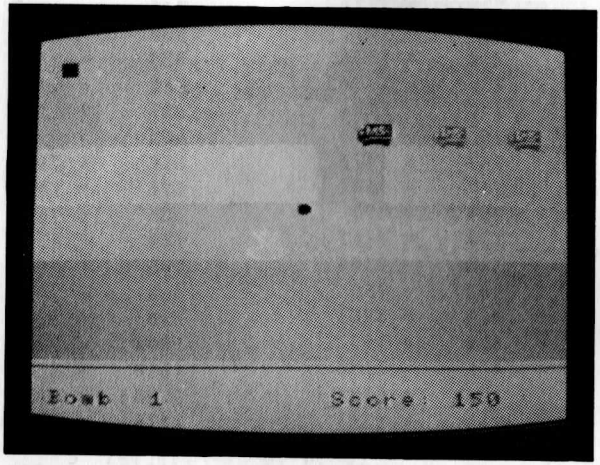


ALLIGATOR: MOUTH CLOSED



GOLDFISH

Convoy



CLASSIFICATION: Time-limit Game

You are a saboteur and your mission is to stop a convoy of enemy trucks from passing you. The idea is to set time bombs underneath the road by positioning your boat on the underground river.

The problem: your own trucks (painted red) must be allowed to pass and you have no control over the fuses on your bombs!

Press the <SPACE> bar to set a bomb or patch up the road — and beware of falling trucks!

PROGRAMMING SUGGESTIONS

The falling trucks would look great if there was a big splash and noise when they hit the water. The code to achieve this would go in the 'Drop Truck' routine, lines 55Ø-58Ø.

PROGRAM
Variables

NT	Last new truck
SC	Score
MD	Minimum distance between trucks
C(I)	Sprite colours
RD, RS	Random number; random number seed
FI, FS	Fuse length; fuse-set flag
LT	Last truck moved
CS	Current speed of trucks
PC	Player's column
BP	Bomb's position
BS	Bomb set flag

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   COLOR 1, 7, 7 : SCREEN 1, 2 : KEY OFF : PRINT "▲▲▲▲▲▲▲▲
     ▲▲▲CONVOY▲": FOR I = 1 TO 9 : PRINT : NEXT
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : FOR I = 1 TO 4 : SPRITE$( I ) = A$ : NEXT
30   PUT SPRITE 1, ( 20, 50 ) , 1 : PUT SPRITE 2, ( 80,
     50 ) , 8 : PUT SPRITE 3, ( 160, 50 ) , 1 : PUT SPRITE
     4, ( 220, 50 ) , 8
35   DEFUSR0 = 60000! : DEFUSR1 = 60118! : POKE 59996!, 10
40   PRINT "▲▲▲▲Hit▲Any▲Key▲to▲Start"
50   IF INKEY$ = "" THEN 50
60   CS = 2
70   FOR I = 1088 TO 1095 STEP 2 : VPOKE I, 170 : VPOKE
     I + 1, 85 : NEXT : FOR I = 1152 TO 1159 : VPOKE
     I, 255 : NEXT : VPOKE 8209, 231 : VPOKE 8210, 71
```

Set Up Screen

```
100  CLS : FOR I = 6336 TO 6463 : VPOKE I, 136 : NEXT :
     FOR I = 6592 TO 6815 : VPOKE I, 144 : NEXT
105  FOR I = 6816 TO 6847 : VPOKE I, 23 : NEXT
110  FOR I = 1 TO 23 : PRINT : NEXT : PRINT "Bomb:";
     CHR$( 11 ) ; : GOSUB 1000
120  PUT SPRITE 1, ( 10, 200 ) : PUT SPRITE 2, ( 40,
     200 ) : PUT SPRITE 3, ( 70, 200 ) : PUT SPRITE 4,
     ( 100, 200 ) : NT = 1 : MD = 25 : GOSUB 400
130  PUT SPRITE 0, ( 120, 95 ) , 11
140  STRIG( 0 ) ON : ON STRIG GOSUB 950
150  RS = 10001 * ( TIME / 60 - INT( TIME / 60 ) )
160  ON SPRITE GOSUB 750
```

Control

```
200  POKE 59999!, 5 : D = USR0( D )
205  GOSUB 900 : IF FS = 0 AND RD < .2 THEN GOSUB 900 :
     FL = INT( RD * 5 + 1 ) : FS = 1 : TIME = 0
207  GOSUB 1100
```

```
210   LT = LT + 1 : IF LT = 5 THEN LT = 1
212   IF VPEEK( 6912 + 4 * LT ) = 200 THEN 210 ELSE POKE
      59997!, LT : POKE 59998!, 1 : POKE 59999!, CS :
      D = USR1( D ) : IF( VPEEK( 6913 + 4 * LT ) < 2 +
      CS ) AND C ( LT ) = 1 THEN 800
213   IF VPEEK( 6913 + 4 * LT ) < 2 + CS THEN PUT SPRITE
      LT, ( 30 * LT - 20, 200 )
215   C = INT( ( VPEEK( 6913 + 4 * LT ) + 8 ) / 8 ) :
      IF VPEEK( 6368 + C ) = 32 AND VPEEK( 6367 + C )
      = 32 AND VPEEK( 6369 + C ) = 32 THEN GOSUB 550
220   IF FS = 1 AND BS = 0 AND INT( TIME / 60 ) > FL THEN
      700
225   IF BS = 1 AND INT( TIME / 60 ) > FL THEN GOSUB 450
230   GOSUB 900 : IF RD < .05 THEN GOSUB 400
240   GOTO 200
```

Patch Up Road

```
350   PLAY "l9m1000s14n53": IF PC < 3 THEN 380
352   IF PC > 28 THEN 392
355   FOR I1 = 6336 TO 6432 STEP 32 : FOR I2 = PC - 2
      TO PC + 2
360   IF VPEEK( I1 + I2 ) = 32 THEN VPOKE I1 + I2, 136
365   NEXT : NEXT : RETURN
380   FOR I1 = 6336 TO 6432 STEP 32 : FOR I2 = 0 TO PC
385   IF VPEEK( I1 + I2 ) = 32 THEN VPOKE I1 + I2, 136
390   NEXT : NEXT : RETURN
392   FOR I1 = 6336 TO 6432 STEP 32 : FOR I2 = 29 TO 31
394   IF VPEEK( I1 + I2 ) = 32 THEN VPOKE I1 + I2, 136
396   NEXT : NEXT : RETURN
```

New Truck

```
400   J = 0 : FOR I = 1 TO 4 : IF VPEEK( 6912 + 4 * I )
      = 200 THEN J = I
405   NEXT
410   IF J = 0 THEN RETURN
415   XP = VPEEK( 6913 + 4 * NT ) : IF 255 - XP < MD THEN
      RETURN
420   NT = J : GOSUB 900 : IF RD < .4 THEN C ( NT ) =
      1 ELSE C ( NT ) = 8
425   PUT SPRITE NT, ( 255, 34 ) , C ( NT )
430   RETURN
```

Blow Bomb

```
450   PLAY "s8m200l64n30n28n26n24": FS = 0 : VPOKE 6464
      + BP, 32 : BS = 0 : IF BP = 31 THEN 470
```

```
455   IF BP = 0 THEN 480
460   FOR J = 6336 TO 6432 STEP 32 : FOR K = BP - 1 TO BP + 1
465   VPOKE J + K, 32 : NEXT : NEXT : RETURN
470   FOR J = 6336 TO 6432 STEP 32 : FOR K = 29 TO 31
475   VPOKE J + K, 32 : NEXT : NEXT : RETURN
480   FOR J = 6336 TO 6432 STEP 32 : FOR K = 0 TO 2
485   VPOKE J + K, 32 : NEXT : NEXT : RETURN
```

Place Bomb

```
500   BS = 1 : BP = PC : VPOKE 6464 + PC, 133
510   PLAY "154m1200s10n35"
520   RETURN
```

Drop Truck

```
550   SPRITE ON : POKE 59997!, LT : POKE 59998!, 2 : POKE
      59999!, 3
555   D = USR1( D ) : IF VPEEK( 6912 + 4 * LT ) < 100
      THEN 555
560   IF C ( LT ) = 8 THEN 800
565   SC = SC + 50 : VPOKE 6912 + 4 * LT, 200 : VPOKE
      6913 + 4 * LT, LT * 30 - 20
570   IF SC < 300 THEN CS = 3 ELSE IF SC < 500 THEN CS
      = 4 ELSE IF SC < 700 THEN CS = 5 ELSE IF SC < 900
      THEN CS = 6 ELSE IF SC < 1400 THEN CS = 7 ELSE CS = 8
580   GOSUB 400 : SPRITE OFF : GOSUB 1000 : RETURN
```

Blow Player

```
700   FOR I = 1 TO 40 : GOSUB 900 : A$ = "n"+ STR$( INT(
      10 * RD + 1 ) ) : PLAY "164m350s8xa$;": VPOKE 14336
      + INT( 60 * RD ) , RD * 255 : NEXT
740   GOTO 800
```

Man Squashed

```
750   CLS : PRINT "YOU▲HAVE▲JUST▲BEEN▲SQUASHED": FOR I
      = 1 TO 2500 : NEXT
```

Game Over

```
800   STRIG( 0 ) OFF : SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲▲GAME▲OVER";
      FOR I = 1 TO 9 : PRINT : NEXT : PRINT "▲▲▲▲▲▲Your▲Score
      ▲was"; SC
```

```
810   IF INKEY$ = "" THEN END ELSE 810
```

Random Number

```
900   RD =  ( 9999 * RD + RS ) MOD 2997! : RD = RD / 2997!
910   IF RD < .5 THEN RS = RD * 10000 + 1
920   RETURN
```

Space Bar Pressed

```
950   PC = INT( ( VPEEK( 6913 ) + 8 ) / 8 )
955   IF VPEEK( 6432 + PC ) <> 32 AND BS = 0 THEN GOSUB
      500 : RETURN
960   IF VPEEK( 6432 + PC ) = 32 THEN GOSUB 350 : RETURN
965   RETURN
```

Update Score

```
1000  FOR I = 1 TO 22 : PRINT : NEXT : PRINT TAB( 16 )
      CHR$( 10 ) "Score:"; SC ; CHR$( 11 ) ;
1010  RETURN
```

Print Time Left for Fuse

```
1100  I = FL - INT( TIME / 60 ) : IF I < 0 THEN I = 0
1110  IF FS = 0 THEN VPOKE 6888, 32 : RETURN ELSE VPOKE
      6888, 48 + I : RETURN
```

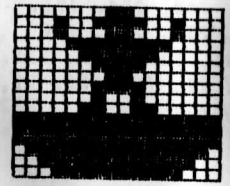
Sprite Data

```
10000 DATA 1, 19, 17, 15, 7, 3, 1, 3, 2, 6, 255, 255,
      255, 127, 63, 31, 128, 200, 136, 240, 224, 192,
      128, 192, 64, 96, 255, 255, 255, 254, 252, 248
10010 DATA 0, 15, 23, 18, 114, 149, 181, 255, 248, 231,
      95, 63, 56, 0, 0, 0, 0, 255, 69, 21, 75, 101, 21,
      255, 0, 255, 254, 254, 14, 0, 0, 0
```

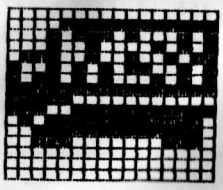
ChexSum Table

```
10     = 0
11     = 0
12     = 0
15     = 4362
20     = 3995
25     = 4580
30     = 4196
35     = 2781
40     = 2333
50     = 997
60     = 412
70     = 6615
100    = 4236
105    = 1768
110    = 3182
120    = 6020
130    = 899
140    = 1687
150    = 2701
160    = 750
200    = 1540
205    = 5332
207    = 238
210    = 2131
212    = 11034
213    = 4245
215    = 8831
220    = 3431
225    = 2877
230    = 1704
240    = 593
350    = 2316
352    = 956
355    = 2965
360    = 2590
365    = 525
380    = 2263
385    = 2590
390    = 525
392    = 2199
394    = 2590
396    = 525
400    = 3692
405    = 131
410    = 838
415    = 3426
420    = 3581
425    = 1459
430    = 143
450    = 5520
455    = 1026
460    = 2826
465    = 1256
470    = 2076
475    = 1256
480    = 1988
485    = 1256
500    = 2085
510    = 1305
520    = 143
550    = 2898
555    = 2924
560    = 965
565    = 4287
570    = 9650
580    = 1476
700    = 8346
740    = 391
750    = 4083
800    = 6613
810    = 1349
900    = 3332
910    = 2168
920    = 143
950    = 2111
955    = 3193
960    = 2069
965    = 143
1000   = 4035
1010   = 143
1100   = 2857
1110   = 2995
10000  = 8898
10010  = 7900

Total=230712
```

Sprite Shapes

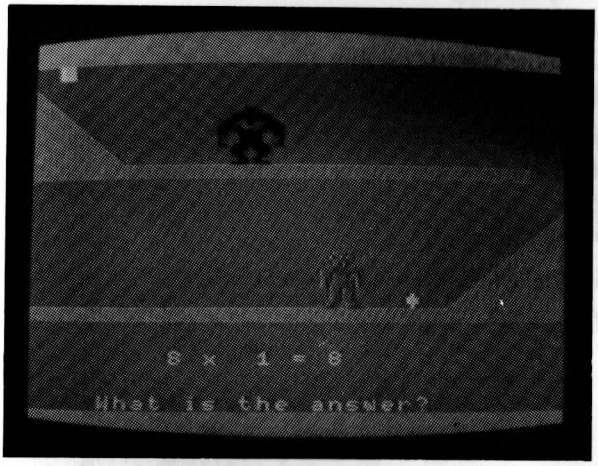


MAN IN BOAT



TRUCK

Gorilla Maths



CLASSIFICATION: Educational Game

Fred is keen to climb the hill to meet the maths gorilla. When you answer a question correctly, the falling rock doesn't push Fred back to the start and he moves closer to the gorilla.

You can have questions about addition or multiplication and there are two levels of difficulty. When you have answered a question, press <RETURN>.

PROGRAMMING SUGGESTIONS

Perhaps you could animate the gorilla while the rock is falling or have it shake hands when she meets Fred. How about a sharp cry of pain when the rock hits Fred?

PROGRAM

Variables

J	Jumped flag
LD	Level of difficulty
Q	Question type (1 - 2)
RP	Rock's pattern (8 or 12)
N1, N2	Numbers to add/multiply
NB	New 'ball' flag
C	Count of digits of player's answer
PA	Player's answer
SP	Speed of rock
M$	Message

Listing

Initialise

```
1   COLOR 15, 4, 7 : SCREEN 1, 3 : KEY OFF : ON INTERVAL
    = 1 GOSUB 4 : INTERVAL ON
2   GOTO 5
4   D = RND( 1 ) : RETURN
5   FOR I = 6145 TO 6148 : VPOKE I, 23 : NEXT : FOR
    I = 6174 TO 6174 : VPOKE I, 23 : NEXT : VPOKE 6144,
    24 : VPOKE 6175, 25 : VPOKE 6880, 26 : VPOKE 6911,
    27 : FOR I = 6176 TO 6848 STEP 32 : VPOKE I, 22 :
    VPOKE I + 31, 22 : NEXT : FOR I = 6881 TO 6910 :
    VPOKE I, 23 : NEXT
10  PRINT "▲▲G▲O▲R▲I▲L▲L▲A▲▲▲M▲A▲T▲H▲S": FOR I = 1 TO
    9 : PRINT : NEXT
20  FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
    NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25  FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
    NEXT : SPRITE$( 1 ) = A$ : A$ = ""
30  FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
    NEXT : SPRITE$( 2 ) = A$ : A$ = ""
35  FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
    NEXT : SPRITE$( 3 ) = A$
37  FOR I = 1088 TO 1119 : READ Q : VPOKE I, Q : NEXT
40  PUT SPRITE 1, ( 116, 30 ) , 1
45  INPUT "Would▲you▲like▲to▲add▲(a)▲▲▲▲or▲multiply▲(m)▲the
    ▲numbers"; X$
50  IF X$ = "m" OR X$ = "M" THEN Q = 2 : GOTO 60
55  IF X$ = "a" OR X$ = "A" THEN Q = 1 ELSE 45
60  PRINT : INPUT "How▲hard▲(1▲or▲2)"; X$ : IF X$ <
    "1" OR X$ > "2" THEN 60
65  LD = VAL( X$ )
70  CLS : FOR I = 6374 TO 6397 : VPOKE I, 138 : NEXT :
    FOR I = 6688 TO 6712 : VPOKE I, 138 : NEXT
75  VPOKE 8209, 228 : PUT SPRITE 1, ( 92, 23 ) : PUT
    SPRITE 0, ( 0, 104 ) , 9
80  FOR I = 0 TO 5 : VPOKE 6176 + 33 * I, 136 : NEXT :
    FOR I = 0 TO 6 : VPOKE 6495 + 31 * I, 137 : NEXT
85  FOR I = 6208 TO 6368 STEP 32 : FOR J = 0 TO K :
    VPOKE I + J, 139 : NEXT : K = K + 1 : NEXT
90  K = 0 : FOR I = 6527 TO 6719 STEP 32 : FOR J = -
    K TO 0 : VPOKE I + J, 139 : NEXT : K = K + 1 : NEXT
95  INTERVAL OFF : SPRITE ON : ON SPRITE GOSUB 700
```

Print Question

```
100   C = 0 : NB = 0 : RP = 8 : PA = 0 : PUT SPRITE 2,
      ( 0, 1 )
102   M$ = "What▲is▲the▲answer?": GOSUB 1000
105   IF Q <> 1 THEN 120
110   IF LD = 1 THEN N1 = INT( RND( 1 ) * 10 + 1 ) : N2
      = INT( RND( 1 ) * 10 + 1 ) ELSE N1 = INT( RND( 1 )
      * 20 + 1 ) : N2 = INT( RND( 1 ) * 20 + 1 )
115   K = 43 : GOTO 140
120   IF LD = 1 THEN N1 = INT( RND( 1 ) * 5 + 1 ) : N2
      = INT( RND( 1 ) * 5 + 1 ) ELSE N1 = INT( RND( 1 )
      * 10 + 1 ) : N2 = INT( RND( 1 ) * 10 + 1 )
125   K = 120
140   FOR I = 1 TO 20 : PRINT : NEXT : PRINT TAB( 6 ) ;
      : PRINT USING "##"; N1 ; : PRINT TAB( 9 ) ; CHR$( K)
      ; TAB( 11 ) ;: PRINT USING "##"; N2 ; : PRINT TAB(
      14 ) ; "=▲???"; CHR$( 11 ) ;
150   J = 0
```

Control

```
300   GOSUB 400 : IF NB = 1 THEN 100
310   GOTO 500
390   GOTO 300
```

Move Rock

```
400   IF LD = 1 THEN SP = 2 ELSE SP = 3
405   K1 = VPEEK( 6921 ) : K2 = VPEEK( 6920 ) : IF K1
      / 4 <> INT( K1 / 4 ) THEN IF RP = 8 THEN RP = 12
      ELSE RP = 8
410   VPOKE 6922, RP : IF K1 < 49 AND K2 < 46 THEN VPOKE
      6921, K1 + SP : VPOKE 6920, K2 + SP : RETURN
420   IF K2 < 50 AND K1 < 240 THEN VPOKE 6921, K1 + SP :
      IF K1 > 62 AND K1 < 125 THEN VPOKE 6916, 0 : RETURN
      ELSE VPOKE 6916, 23 : RETURN
425   IF K2 < 80 THEN VPOKE 6920, K2 + SP : RETURN
430   IF K2 < 125 THEN VPOKE 6920, K2 + SP : VPOKE 6921,
      K1 - SP : RETURN
440   IF K1 < 5 THEN NB = 1 : RETURN ELSE VPOKE 6921,
      K1 - SP : RETURN
```

Read

```
500   X$ = INKEY$ : IF X$ = "" THEN 300
505   IF( X$ < "0" OR X$ > "9") AND X$ <> CHR$( 13 ) THEN
      300
510   IF X$ = CHR$( 13 ) THEN 600 ELSE IF C = 0 THEN VPOKE
      6802, 32 : VPOKE 6803, 32 : VPOKE 6804, 32
515   C = C + 1 : IF C > 3 THEN 600 ELSE PA = PA * 10
      + VAL( X$ ) : VPOKE 6801 + C, 48 + VAL( X$ ) : GOTO
      300
```

Check Answer

```
600   IF C = 0 THEN 300 ELSE M$ = "▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲":
      GOSUB 1000 : IF Q <> 1 THEN 650
605   IF N1 + N2 <> PA  THEN 300
610   GOTO 660
650   IF N1 * N2 <> PA  THEN 300
660   IF NB = 1 THEN 300 ELSE J = 1 : SPRITE OFF : K1
      = VPEEK( 6920 ) : K2 = VPEEK( 6921 ) : IF K1 < 50
      THEN VPOKE 6912, 23 : IF K2 < 200 THEN VPOKE 6913,
      K2 ELSE VPOKE 6913, 200 : PUT SPRITE 2, ( 100, 200 )
      : NB = 1 ELSE IF K2 > 195 THEN VPOKE 6913, 165 ELSE
      VPOKE 6913, K2
665   SPRITE ON : GOTO 300
```

Sprite Collision

```
700   SPRITE OFF : IF J = 1 THEN GOSUB 1100 : RETURN
705   IF VPEEK( 6912 ) < 70 AND VPEEK( 6913 ) < 115 THEN900
710   FOR I2 = 1 TO 10 : VPOKE 6912, VPEEK( 6912 ) - 10 :
      PLAY "m380s8l64n56": FOR I1 = 1 TO 100 : NEXT :
      VPOKE 6912, VPEEK( 6912 ) + 10 : PLAY "m700l64s8n50":
      NEXT
720   J = 0 : PUT SPRITE 0, ( 0, 104 ) : PUT SPRITE 2,
      ( 0, 0 ) : NB = 1 : SPRITE ON : RETURN
```

Game Over

```
900   SCREEN 1 : FOR I = 1 TO 10 : PRINT : NEXT : PRINT
      "HOORAY!!▲-▲You▲have▲met▲the": PRINT "Maths▲Gorilla"
910   PLAY "m255s118o5edco4bo5co4b5agao5co4bo5dl1c"
990   END
```

Print Message

```
1000    FOR I = 1 TO 23 : PRINT : NEXT : PRINT "▲▲▲"; M$ ;
        CHR$( 11 ) ;
1010    RETURN
```

Player Jumps

```
1100    PLAY "l64n60": NB = 1 : J = 0 : K3 = VPEEK( 6913 )
        : K4 = VPEEK( 6912 ) : IF K4 > 50 THEN IF K3 < 165
        THEN VPOKE 6913, K3 + 20 : SPRITE ON : RETURN ELSE
        VPOKE 6912, 23 : SPRITE ON : RETURN
1110    VPOKE 6913, K3 - 30 : SPRITE ON : RETURN
```

Sprite Data

```
10000   DATA 1, 2, 1, 1, 7, 15, 13, 8, 9, 11, 3, 3, 3, 3,
        6, 14, 192, 160, 192, 64, 240, 248, 216, 8, 200,
        232, 96, 96, 96, 96, 48, 56
10010   DATA 3, 5, 31, 59, 113, 248, 237, 199, 135, 135,
        207, 30, 28, 12, 6, 30, 128, 64, 240, 184, 28, 62,
        110, 198, 194, 194, 230, 240, 112, 96, 192, 240
10020   DATA 96, 240, 96, 0, 0, 0, 0, 0
10030   DATA 64, 224, 224, 64, 0, 0, 0, 0
```

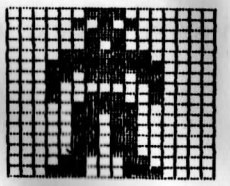
Character Data

```
10100   DATA 128, 64, 160, 80, 168, 84, 170, 85
10102   DATA 1, 2, 5, 10, 21, 42, 85, 170
10104   DATA 255, 170, 85, 170, 85, 170, 85, 170
10106   DATA 85, 170, 85, 170, 85, 170, 85, 170
```

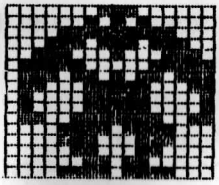
ChexSum Table

```
1     = 4266
2     = 394
4     = 1018
5     = 13556
10    = 3373
20    = 3995
25    = 3996
30    = 4013
35    = 3497
37    = 1915
40    = 827
45    = 5871
50    = 2652
55    = 2420
60    = 3992
65    = 1016
70    = 4133
75    = 2375
80    = 4875
85    = 4389
90    = 5026
95    = 2403
100   = 2594
102   = 2886
105   = 1092
110   = 10117
115   = 978
120   = 9978
125   = 453
140   = 8086
150   = 333
300   = 1287
310   = 633
390   = 433
400   = 1893
405   = 6734
410   = 4454
420   = 6227
425   = 1926
430   = 2976
440   = 2747
500   = 1569
505   = 3352
510   = 4176
515   = 6334
600   = 4489
605   = 1588
610   = 539
650   = 1590
660   = 15461
665   = 851
700   = 1646
705   = 2719
710   = 8801
720   = 3121
900   = 6011
910   = 3837
990   = 129
1000  = 2513
1010  = 143
1100  = 9916
1110  = 1365
10000 = 6990
10010 = 9496
10020 = 1153
10030 = 1279
10100 = 1642
10102 = 1279
10104 = 1721
10106 = 1656

Total= 251195
```

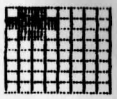
Sprite Shapes



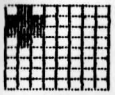
PLAYER STANDING



GORILLA

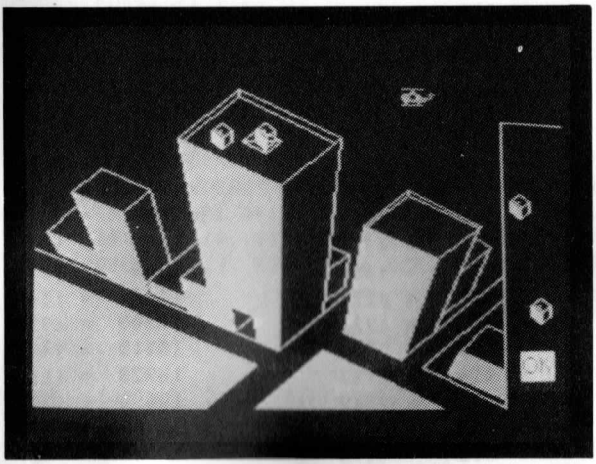


ROCK:
POSITION 1



ROCK:
POSITION 2

Chopper Landing



CLASSIFICATION: Simulation Game

Special packages are needed desperately in a highrise building. Guide your chopper onto the two lower buildings to transport the packages to the tallest building. Don't land on this building as it doesn't have a heliport, so that landing there while loaded causes a crash.

Two packages must be taken from each building, and the scoring is:

hit target	1ØØ
hit building	5Ø
successful pick-up	1Ø

Your score will be better if you work quickly. Use the <SPACE> bar to drop a package.

PROGRAMMING SUGGESTIONS

Tamper with the numbers assigned to J and in the RND(1) statements in lines 200-220 to make the chopper sway more or less and speed up or slow down.

A good exercise in graphics programming would be to improve the buildings and landscape. How about some windows or parked cars?

To make the game longer, you could add a few packages and make the player drop them in a certain order.

PROGRAM

Variables

SC	Score
NP	Number of packages left
SP	Speed for chopper
DR	Directions (up or down bias)
LD	Loaded
B1, B2	Number of packages moved from each building

Listing

Initialise

```
 1   REM   CHOPPER LANDING
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
30   COLOR 11, 1, 1 : SCREEN 2, 2
35   FOR I = 1 TO 75 : READ X1, Y1, X2, Y2 : LINE( X1,
     Y1 ) - ( X2, Y2 ) , 11 : NEXT
40   PAINT( 100, 100 ) : PAINT( 170, 130 ) : PAINT( 85,
     135 ) : PAINT( 70, 130 ) : PAINT( 40, 100 ) : PAINT(
     20, 100 ) : PAINT( 210, 170 ) : PAINT( 140, 180 )
     : PAINT( 40, 160 )
50   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : PUT SPRITE 0, ( 240,
     0 ) , 10 : A$ = ""
55   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : FOR I = 1 TO 4 : SPRITE$( I ) = A$ : NEXT :
     A$ = "": PUT SPRITE 1, ( 226, 70 ) , 14 : PUT SPRITE
     2, ( 241, 70 ) , 14 : PUT SPRITE 3, ( 226, 100 )
     , 14 : PUT SPRITE 4, ( 241, 100 ) , 14
60   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 5 ) = A$ : A$ = ""
65   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 6 ) = A$ : A$ = ""
67   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 7 ) = A$
70   DEFUSR = 60000! : DEFUSR1 = 60118! : POKE 59996!,
     15 : POKE 59997!, 0 : NP = 4 : DR = 2 : SP = 1 :
     TIME = 0
80   STRIG( 0 ) ON : ON STRIG GOSUB 800
```

Control

```
100  POKE 59999!, SP : D = USR( D )
110  GOSUB 200
120  GOSUB 300
190  GOTO 100
```

Move Chopper

```
200  IF RND( 1 ) < .7 THEN K = DR : J = 1 : GOTO 215
205  J = 4 : IF RND( 1 ) < .5 THEN K = 1 : GOTO 215
```

```
210   K = 3
215   POKE 59999!, J : POKE 59998!, K : D = USR1( D )
220   RETURN
300   K = VPEEK( 6918 ) : IF K < 4 OR K > 191 THEN VPOKE
      6912, 4 : RETURN
```

Check Position

```
305   IF K > 160 THEN 500
310   IF K < 30 THEN RETURN
315   X = VPEEK( 6913 ) : IF X > 209 THEN VPOKE 6913,
      209 : IF K > 160 THEN 500 ELSE RETURN
320   IF X < 150 THEN 330 ELSE IF K < 80 THEN RETURN ELSE
      IF X > 200 THEN 500 ELSE IF X < 165 THEN RETURN
      ELSE IF LD = 1 THEN 500 ELSE B2 = B2 + 1 : GOTO 400
330   IF K > 140 THEN 500 ELSE IF X < 60 THEN 340 ELSE
      IF K > 50 THEN RETURN ELSE 500
340   IF K < 60 THEN RETURN ELSE IF K > 100 THEN 500 ELSE
      IF K > 80 THEN RETURN ELSE IF LD = 1 THEN 500 ELSE
      B1 = B1 + 1 : GOTO 400
390   RETURN
```

Pick Up Package

```
400   LD = 1 : PUT SPRITE NP, ( 235, 130 ) : PUT SPRITE
      6, ( 233, 160 ) , 15 : DR = 0 : SC = SC + 10 : VPOKE
      6912, VPEEK( 6912 ) - 2
410   IF B1 > 2 OR B2 > 2 THEN 700
440   RETURN
```

Crash

```
500   PUT SPRITE 6, ( 100, 200 ) : PUT SPRITE 7, ( 233,
      160 ) , 15 : VPOKE 6914, 20 : FOR I = 1 TO 30 :
      PLAY "l60m1000s14n50": NEXT : FOR I = 1 TO 2500 :
      NEXT : PUT SPRITE 7, ( 100, 200 )
550   GOTO 900
```

Wrong Building

```
700   SCREEN 1 : PRINT "TWO▲PACKAGES▲MUST▲BE▲TAKEN▲":
      PRINT "FROM▲EACH▲BUILDING": FOR I = 1 TO 3000 : NEXT
710   PRINT "Please▲try▲again": IF INKEY$ = "" THEN END
      ELSE 710
```

Drop Package

```
800   IF LD = 0 THEN RETURN
805   PUT SPRITE 6, ( 100, 200 ) : PLAY "l8m1000s14n40":
      LD = 0 : DR = 2 : I1 = VPEEK( 6913 ) : IF I1 < 150
      THEN 820 ELSE IF I1 < 170 THEN PUT SPRITE NP, (
      I1, 160 ) : NP = NP - 1 : GOTO 850 ELSE IF I1 <
      200 THEN PUT SPRITE NP, ( I1, 90 ) : NP = NP - 1 :
      GOTO 850 ELSE PUT SPRITE NP, ( I1, 160 ) : NP =
      NP - 1 : GOTO 850
820   IF I1 > 140 THEN PUT SPRITE NP, ( I1, 145 ) : NP
      = NP - 1 : GOTO 850 ELSE IF I1 < 60 THEN 840 ELSE
      IF I1 < 72 THEN PUT SPRITE NP, ( I1, 145 ) : NP
      = NP - 1 : GOTO 850 ELSE IF I1 < 100 OR I1 > 106
      THEN PUT SPRITE NP, ( I1, 32 ) : SC = SC + 50 :
      NP = NP - 1 : GOTO 850
830   PUT SPRITE NP, ( I1, 32 ) : SC = SC + 100 : PLAY
      "m1100s813n50": NP = NP - 1 : GOTO 850
840   IF I1 < 22 OR I1 > 55 THEN PUT SPRITE NP, ( I1,
      110 ) : NP = NP - 1 : GOTO 850 ELSE PUT SPRITE NP,
      ( I1, 70 ) : NP = NP - 1
850   IF NP = 0 THEN FOR I1 = 1 TO 2000 : NEXT : GOTO 900
860   RETURN
```

Game Over

```
900   SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲▲GAME▲OVER": PRINT :
      PRINT : PRINT
910   PRINT "▲▲▲▲▲▲YOUR▲SCORE▲WAS"; SC
915   IF NP <> 0 THEN 940 ELSE K = 150 - TIME / 60 : IF
      K < 0 THEN K = 0
920   PRINT : PRINT : PRINT "Plus▲Bonus▲for▲Moving▲all▲of":
      PRINT "the▲Packages:"; INT( 8 * K ) : PRINT : PRINT
      "This▲gives▲a▲total▲of▲"; SC + INT( 8 * K )
940   IF INKEY$ = "" THEN END ELSE 940
```

Drawing Data

```
10000 DATA 100, 14, 100, 18, 100, 14, 148, 41, 100, 14,
      72, 41, 100, 18, 145, 44, 100, 18, 75, 42, 120,
      68, 148, 41, 120, 68, 72, 41, 120, 68, 120, 160,
      72, 41, 91, 145, 148, 41, 140, 140
10002 DATA 0, 100, 120, 160, 0, 112, 110, 168, 88, 190,
      110, 168, 110, 190, 140, 160, 140, 160, 195, 190,
      120, 160, 153, 127, 149, 150, 176, 165, 186, 169,
      225, 190, 186, 169, 225, 131, 176, 165, 225, 118
```

```
10004   DATA 98, 136, 109, 142, 98, 136, 100, 150, 100,
        150, 108, 142, 109, 142, 110, 155, 102, 43, 120,
        43, 102, 43, 113, 50, 102, 43, 109, 36, 113, 50,
        120, 43, 113, 50, 109, 36, 109, 36, 120, 43
10006   DATA 180, 70, 159, 91, 180, 70, 214, 90, 180, 73,
        161, 91, 180, 73, 213, 91, 180, 70, 180, 73, 159,
        91, 194, 109, 159, 91, 150, 150, 194, 109, 177,
        164, 214, 90, 192, 149, 194, 109, 215, 90
10008   DATA 79, 139, 75, 120, 75, 120, 87, 126, 75, 120,
        84, 111, 78, 137, 61, 128, 61, 128, 59, 122, 59,
        122, 76, 130, 59, 122, 82, 100, 60, 130, 46, 83,
        46, 83, 21, 70, 46, 83, 59, 70
10010   DATA 59, 70, 35, 57, 35, 57, 21, 70, 21, 70, 40,
        119, 40, 117, 16, 105, 16, 105, 10, 93, 10, 93,
        35, 105, 10, 93, 24, 78, 0, 190, 88, 190, 110, 190,
        195, 190, 59, 70, 70, 112
10012   DATA 143, 103, 152, 107, 152, 107, 150, 125, 150,
        125, 140, 134, 142, 116, 152, 107, 205, 116, 219,
        102, 219, 102, 212, 97, 219, 102, 209, 128, 209,
        128, 195, 141, 198, 172, 225, 186, 198, 172, 204,
        160, 204, 160, 225, 171, 204, 160, 217, 146, 217,
        146, 225, 150
10014   DATA 225, 190, 225, 30, 225, 30, 255, 30
```

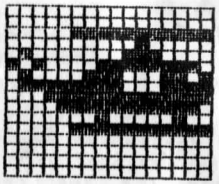
Sprite Data

```
10100   DATA 0, 0, 255, 4, 14, 127, 177, 241, 255, 127,
        32, 255, 0, 0, 0, 0, 0, 0, 224, 0, 2, 133, 194,
        254, 248, 240, 128, 224, 0, 0, 0, 0
10110   DATA 1, 2, 4, 8, 21, 26, 29, 30, 31, 31, 31, 15,
        7, 3, 1, 0, 128, 64, 32, 16, 8, 24, 40, 200, 136,
        136, 136, 144, 160, 192, 128, 0
10120   DATA 0, 4, 66, 136, 2, 64, 18, 42, 10, 165, 43,
        144, 40, 1, 64, 18, 128, 16, 4, 32, 1, 72, 33, 146,
        204, 22, 200, 92, 64, 9, 0, 32
10130   DATA 255, 255, 255, 227, 221, 190, 190, 190, 190,
        190, 221, 227, 255, 255, 255, 255, 255, 255, 255,
        187, 183, 175, 159, 175, 183, 187, 187, 187, 255,
        255, 255, 255
10140   DATA 192, 160, 198, 170, 170, 198, 0, 0, 0, 0, 0,
        255, 0, 0, 0, 0, 0, 0, 198, 170, 170, 166, 2, 2,
        10, 4, 0, 255, 0, 0, 0, 0
```

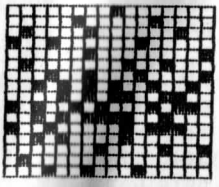
ChexSum Table

```
1     = 0          215   = 2498       840   = 6806
10    = 0          220   = 143        850   = 2666
11    = 0          300   = 3084       860   = 143
12    = 0          305   = 1131       900   = 2359
30    = 719        310   = 868        910   = 1851
35    = 3584       315   = 4179       915   = 3931
40    = 7597       320   = 8579       920   = 11264
50    = 5263       330   = 4059       940   = 1481
55    = 12463      340   = 6853       10000 = 13270
60    = 3996       390   = 143        10002 = 16393
65    = 3997       400   = 6259       10004 = 14772
67    = 3481       410   = 1595       10006 = 14760
70    = 6037       440   = 143        10008 = 12789
80    = 1496       500   = 8697       10010 = 12062
100   = 1661       550   = 523        10012 = 27062
110   = 362        700   = 6488       10014 = 1717
120   = 207        710   = 3653       10100 = 7539
190   = 489        800   = 913        10110 = 7438
200   = 3030       805   = 21980      10120 = 7419
205   = 2867       820   = 17120      10130 = 11545
210   = 337        830   = 4999       10140 = 6674

                                      Total= 345434
```

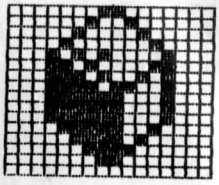
Sprite Shapes



CHOPPER

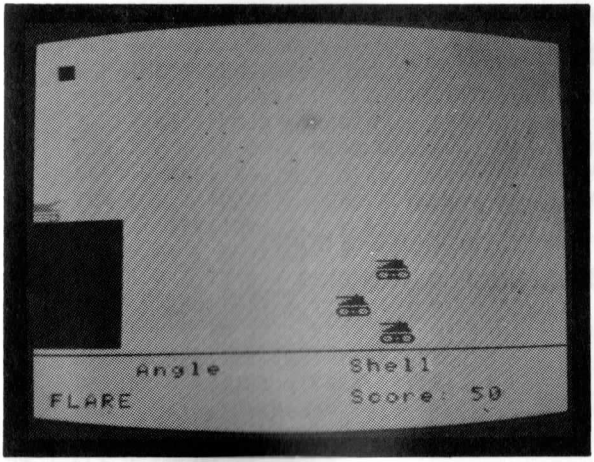


EXPLOSION



PACKAGE

Tank Ambush



CLASSIFICATION: Simulation Game

An ambush has been planned for a squadron of enemy tanks. Your tank is positioned on a cliff and you fire at the enemy on the right. Using keys '0' to '9', give angle of projection of your bomb and shell (muzzle velocity 0 - 9). Then press the space bar to fire, or fire without keying in any numbers to send up a flare.

The game ends when an enemy tank reaches the bottom of the cliff and you score 50 points for each hit.

PROGRAMMING SUGGESTIONS

Like some more tanks? Just create the additional sprites and alter the 'Move Enemy Tanks' routine to incorporate the extra tanks.

The distance each tank is moved determines the difficulty of the game, so changing lines 216-22 would have some interesting effects.

PROGRAM

Variables

RN, SC	Round number; score
M$	Message
NR	New round?
LT	Last tank moved
F	Fire?
A, SH	Angle; shell
T	Time parameter
XI, YI	Initial co-ordinates of bomb
X, Y	Calculated co-ordinates

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   SCREEN 1, 2 : KEY OFF : PRINT "▲▲▲▲T▲A▲N▲K▲▲A▲M▲B▲U▲S▲H
     ": PRINT : PRINT : PRINT : PRINT
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : SPRITE$( 2 ) = A$ : SPRITE$(
     3 ) = A$ : SPRITE$( 4 ) = A$ : A$ = ""
30   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 5 ) = A$ : A$ = ""
35   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 6 ) = A$
40   DEFUSR = 60118! : POKE 59998!, 1
45   INPUT "Are▲You▲Ready▲to▲Start▲▲▲▲▲▲▲▲▲▲(Press▲'RETURN')
     ▲"; X$
50   CLS : COLOR 14, 1, 1
55   FOR I = 6496 TO 6752 STEP 32 : FOR J = 0 TO 5 :
     VPOKE I + J, 255 : NEXT : NEXT : FOR I = 6784 TO
     6815 : VPOKE I, 23 : NEXT
60   FOR I = 1 TO 21 : PRINT : NEXT : PRINT "▲▲▲▲▲Angle▲▲▲▲▲
     ▲▲Shell":  PRINT CHR$( 11 ) ; : M$ = "ready▲▲▲▲▲▲▲▲▲▲▲▲":
     GOSUB  1000
65   FOR I = 6144 TO 6431 : IF RND( 1 ) < .07 THEN V7
     POKE I, 46
67   NEXT : FOR I = 368 TO 375 : VPOKE I, 0 : NEXT :
     VPOKE 369, 1
```

New Round

```
70   PUT SPRITE 0, ( 2, 74 ) , 10 : LT = 1 : F = 0 :
     K1 = 35 * ( TIME / 10 - INT( TIME / 10 ) ) + 60 :
     PUT SPRITE 1, ( 235, K1 ) , 1 : PUT SPRITE 2, (
     255, K1 + 20 ) , 1 : PUT SPRITE 3, ( 235, K1 + 40 )
     , 1 : PUT SPRITE 4, ( 255, K1 + 55 ) , 1
75   NR = 0 : SPRITE ON : ON SPRITE GOSUB 300
```

Control

```
80   IF F = 0 THEN 95
85   GOSUB 150
```

```
 95   GOSUB 200 : IF NR = 1 THEN RN = RN + 1 : GOTO 70
100   X$ = INKEY$ : IF X$ = "" OR F <> 0 THEN 80
105   IF X$ = "▲" THEN M$ = "FLARE▲▲▲▲▲▲▲▲▲▲▲▲": GOSUB
      1000 : F = 2 : A = .628 : SH = 9 : GOSUB 150 : GOTO80
110   IF X$ < "0" OR X$ > "9" THEN 100
115   K1 = VAL( X$ ) : VPOKE 6829, K1 + 48
117   X$ = INKEY$ : IF X$ = "" THEN GOSUB 200 : GOTO 117
120   IF X$ < "0" OR X$ > "9" THEN 117
125   A = 10 * K1 + VAL( X$ ) : VPOKE 6830, VAL( X$ )
      + 48 : A = A * .01745
130   X$ = INKEY$ : IF X$ = "" THEN GOSUB 200 : GOTO 130
135   IF X$ < "0" OR X$ > "9" THEN 130
140   SH = VAL( X$ ) : VPOKE 6841, SH + 48
145   F = 0 : X$ = INKEY$ : IF X$ = "" THEN GOSUB 200 :
      GOTO 145
147   IF X$ = "▲" THEN M$ = "FIRE▲▲▲▲▲▲▲▲▲▲▲▲": GOSUB
      1000 : F = 1 : GOSUB 150 : GOTO 80
148   GOTO 145
```

Move Bomb

```
150   T = T + 3 : XI = SH * COS( A ) / 2.1 : YI = SH *
      SIN( A ) / 2.1
152   IF T <> 3 AND T <> 0 THEN 155
153   IF F = 0 THEN RETURN ELSE PUT SPRITE 5, ( 20, 55 )
      , 15 : PLAY "m80s1l64n30n20"
155   X = INT( 20 + T * XI )
160   Y = INT( 55 - T * YI + .049 * T * T )
165   IF X < 8 OR X > 247 OR Y > 154 THEN GOSUB 400 :
      RETURN
170   IF Y < 8 THEN PUT SPRITE 5, ( 100, 210 ) ELSE PUT
      SPRITE 5, ( X, Y ) , 15
175   IF F <> 2 THEN 185
180   IF T > = 30 THEN 250
185   RETURN
```

Move Enemy Tanks

```
200   SPRITE OFF : LT = LT + 1 : IF LT = 5 THEN LT = 1
210   IF VPEEK( 6912 + LT * 4 ) > 200 THEN RETURN
215   POKE 59997!, LT : ON RN GOTO 216, 217, 217, 219,
      219, 221, 222
216   K = 1 : GOTO 225
217   K = 2 : GOTO 225
219   K = 3 : GOTO 225
221   K = 4 : GOTO 225
222   K = 5
225   POKE 59999!, K : D = USR( D )
```

```
230   IF VPEEK( 6913 + LT * 4 ) < 41 THEN 600
235   SPRITE ON : RETURN
```

Blow Flare

```
250   PUT SPRITE 5, ( 100, 210 ) : FOR I = 1 TO 5 :  COLOR
      1, 15, 15 : FOR J = 1 TO 70 : NEXT : COLOR 14, 1,
      1 : FOR J = 1 TO 70 : NEXT : NEXT
255   M$ = "Ready▲▲▲▲▲▲▲▲▲▲▲▲": GOSUB 1000 : F = 0 : T
      = 0 : VPOKE 6829, 32 : VPOKE 6830, 32 : VPOKE 6841,32
270   RETURN
```

Blow Enemy Tank

```
300   IF F <> 1 THEN RETURN ELSE VPOKE 6934, 24 : COLOR
      1, 15, 15 : PLAY "m13000s118n40n35n40n35": FOR I1
      = 1 TO 200 : NEXT : COLOR 14, 1, 1
305   I2 = 0 : FOR I1 = 1 TO 4 : IXD = VPEEK( 6913 + I1
      * 4 ) - VPEEK( 6933 ) : IYD = VPEEK( 6912 + I1 *
      4 ) - VPEEK( 6932 ) : IF IXD < 0 THEN IXD = - IXD
310   IF IYD < 0 THEN IYD = - IYD
315   IF VPEEK( 6933 ) + 6 > = VPEEK( 6913 + I1 * 4 )
      AND IXD < = 16 AND VPEEK( 6932 ) + 6 > = VPEEK(
      6912 + I1 * 4 ) AND IYD < = 14 THEN I2 = I1
320   NEXT : VPOKE 6914 + 4 * I2, 24 : FOR I1 = 1 TO 200 :
      NEXT
325   SC = SC + 50 : VPOKE 6914 + 4 * I2, 4 * I2 : VPOKE
      6934, 20 : PUT SPRITE 5, ( 100, 210 ) : PUT SPRITE
      I2, ( I2 * 25 - 25, 210 ) : F = 0 : T = 0 : M$ =
      "Ready▲▲▲▲▲▲▲▲▲▲▲▲": GOSUB 1000
330   IF VPEEK( 6916 ) > 200 AND VPEEK( 6920 ) > 200 AND
      VPEEK( 6924 ) > 200 AND VPEEK( 6928 ) > 200 THEN NR = 1
350   VPOKE 6829, 32 : VPOKE 6830, 32 : VPOKE 6841, 32 :
      RETURN
```

Blow Bomb

```
400   VPOKE 6934, 24 : T = 0 : F = 0 : M$ = "Ready▲▲▲▲▲▲▲▲▲▲
      ": GOSUB 1000
405   PLAY "m80164s8n10n12": VPOKE 6829, 32 : VPOKE 6830,
      32 : VPOKE 6841, 32 : FOR I = 1 TO 400 : NEXT
410   VPOKE 6934, 20 : PUT SPRITE 5, ( 100, 210 )
450   RETURN
```

Game Over

```
600   SCREEN 1 : PRINT "▲▲▲▲▲▲▲▲▲GAME▲OVER": PRINT : PRINT :
      PRINT : PRINT
610   PRINT "YOUR▲SCORE▲WAS▲"; SC : PRINT
620   PRINT "YOU▲SURVIVED▲"; RN ; "▲WAVES▲": PRINT "OF▲TANKS"
630   END
```

Update Score/Message

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT M$ ; TAB( 17)
      ; "Score:"; SC ; CHR$( 11 ) ; : RETURN
```

Sprite Data

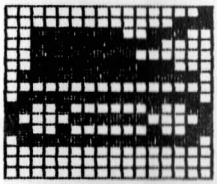
```
10000 DATA 0, 0, 63, 63, 63, 63, 127, 0, 127, 200, 135,
      200, 127, 0, 0, 0, 1, 7, 156, 240, 192, 248, 254,
      0, 254, 19, 225, 19, 254, 0, 0, 0
10010 DATA 0, 0, 255, 1, 7, 31, 127, 0, 127, 199, 146,
      199, 127, 0, 0, 0, 0, 64, 248, 248, 248, 252, 254,
      0, 254, 227, 73, 227, 254, 0, 0, 0
10020 DATA 0, 0, 24, 60, 60, 24, 0, 0
10030 DATA 128, 4, 64, 0, 144, 5, 40, 134, 67, 177, 68,
      16, 68, 0, 128, 16, 4, 0, 129, 16, 6, 129, 146,
      34, 25, 162, 16, 65, 0, 8, 0, 129
```

ChexSum Table

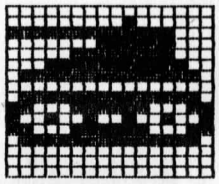
```
10    = 0
11    = 0
12    = 0
15    = 3727
20    = 3995
25    = 6939
30    = 4012
35    = 3484
40    = 1767
45    = 4749
50    = 578
55    = 5731
60    = 6936
65    = 2725
67    = 2349
70    = 13277
75    = 1473
80    = 809
85    = 310
95    = 2685
100   = 2467
105   = 6131
110   = 1814
115   = 2004
117   = 2501
120   = 1814
125   = 4264
130   = 2535
135   = 1844
140   = 2083
145   = 2984
147   = 4492
148   = 538
150   = 4921
152   = 1980
153   = 3437
155   = 1605
160   = 2787
165   = 2905
170   = 2888
175   = 1144
180   = 1246
185   = 143
200   = 2651
210   = 2250
215   = 4062
216   = 1025
217   = 1025
219   = 1029
221   = 1029
222   = 337
225   = 1544
```

```
230    = 2093
235    = 551
250    = 5620
255    = 5049
270    = 143
300    = 6683
305    = 9158
310    = 1848
315    = 9139
320    = 2518
325    = 10530
330    = 6188
350    = 1875
400    = 3417
405    = 4720
410    = 1424
450    = 143
600    = 2537
610    = 1819
620    = 3353
630    = 129
1000   = 3927
10000  = 7777
10010  = 7899
10020  = 1158
10030  = 7606
Total= 246329
```

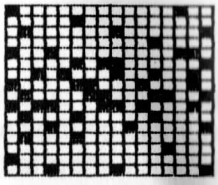
Sprite Shapes



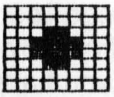
PLAYER'S TANK



ENEMY TANK

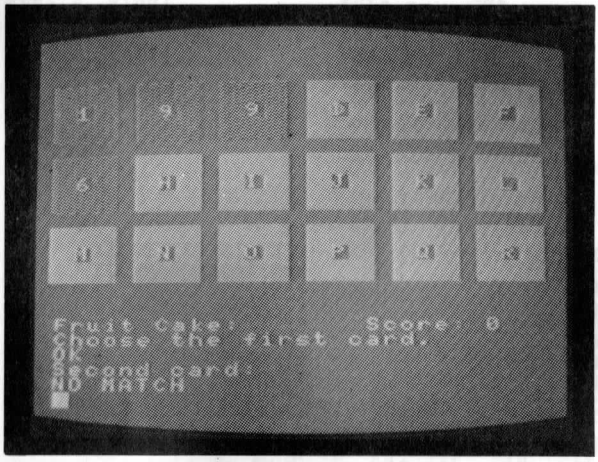


EXPLOSION



BOMB

Concentration



CLASSIFICATION: Memory Game

Challenge your opponent to a tough brain-battle! You select two cards from 18 and, if they have the same number on their opposite sides, you have a match. Otherwise, they are turned face-down again and a chance is given to your opponent.

May the best memory win!

PROGRAMMING SUGGESTIONS

For an easier game, use a smaller range of numbers on the backs of the cards.

For a harder game, instead of having two sets of numbers from 1 to 9, have three sets of numbers from 1 to 6 and make each player choose three cards.

The display would look great with some more colouring for the cards!

PROGRAM
Variables

P$(1), P$(2)	Players' names
S(1), S(2)	Scores
C(I, J)	Card numbers; visible flags
CP	Current player
CF	Card to flip
LC	Last card

Listing

Initialise

```
10   SCREEN 1 : COLOR 15, 4, 4 : KEY OFF : PRINT TAB( 8)
     "CONCENTRATION"
15   DIM C ( 18, 2 )
20   FOR I = 1072 TO 1143 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1152 TO 1223 : READ Q : VPOKE I, Q : NEXT :
     VPOKE 8208, 164 : VPOKE 8209, 164 : VPOKE 8210,
     228 : VPOKE 8211, 228
25   FOR I = 1 TO 5 : PRINT : NEXT : PRINT "  Enter Name for
     Player 1:"
35   X$ = INKEY$ : IF X$ = "" THEN D = RND( 1 ) : GOTO
     35 ELSE IF X$ = CHR$( 13 ) THEN P$ ( 1 ) = "Fruit Cake":
     GOTO 50 ELSE P$ ( 1 ) = X$
40   L = LEN( P$ ( 1 ) ) : BEEP : VPOKE 6403 + L, ASC( X$)
     : IF L > 15 THEN 50
45   X$ = INKEY$ : IF X$ = "" THEN D = RND( 1 ) : GOTO
     45 ELSE IF X$ = CHR$( 13 ) THEN 50 ELSE P$ ( 1 )
     = P$ ( 1 ) + X$ : GOTO 40
50   FOR I = 1 TO 5 : PRINT : NEXT : PRINT "  Enter Name for
     Player 2"
60   X$ = INKEY$ : IF X$ = "" THEN 60 ELSE IF X$ = CHR$(
     13 ) THEN P$ ( 2 ) = "Nut Case": GOTO 75 ELSE P$
     ( 2 ) = X$
65   L = LEN( P$ ( 2 ) ) : BEEP : VPOKE 6595 + L, ASC( X$)
     : IF L > 15 THEN 75
70   X$ = INKEY$ : IF X$ = "" THEN 70 ELSE IF X$ = CHR$(
     13 ) THEN 75 ELSE P$ ( 2 ) = P$ ( 2 ) + X$ : GOTO 65
75   FOR I = 1 TO 5 : PRINT : NEXT : PRINT "OK "; P$
     ( 1 ) ; " -": PRINT "you can go first. Hit a a key"
80   IF INKEY$ = "" THEN 80
```

Set Up Screen

```
100  CLS : FOR CF = 1 TO 18 : GOSUB 400 : NEXT
140  CP = 1 : GOSUB 600 : I = 0 : J = 0
150  I = I + 1 : IF I = 10 THEN I = 1 : J = J + 1 : IF
     J = 2 THEN 200
155  K = INT( RND( 1 ) * 18 + 1 ) : IF C ( K, 1 ) = 0
     THEN C ( K, 1 ) = I  ELSE 155
160  GOTO 150
```

Editor

```
200   PRINT CHR$( 11 ) ; : FOR I = 1 TO 19 : PRINT : NEXT :
      PRINT "Choose▲the▲first▲card."
205   X$ = INKEY$ : IF X$ = "" OR( ( X$ < "A" OR X$ >
      "R") AND( X$ < "a" OR X$ > "r") ) THEN 205
210   IF X$ > "R" THEN CF = ASC( X$ ) - 96 ELSE CF = ASC(
      X$ ) - 64
215   IF C ( CF, 2 ) = 1 THEN 205 ELSE GOSUB 300 : PRINT
      "OK": LC = CF
220   PRINT "Second▲card:▲"
225   X$ = INKEY$ : IF X$ = "" OR( ( X$ < "A" OR X$ >
      "R") AND( X$ < "a" OR X$ > "r") ) THEN 225
230   IF X$ > "R" THEN I = ASC( X$ ) - 96 ELSE I = ASC(
      X$ ) - 64
235   IF I = CF THEN 225 ELSE CF = I : IF C ( CF, 2 )
      = 1 THEN 225 ELSE GOSUB 300
240   IF C ( CF, 1 ) = C ( LC, 1 ) THEN GOSUB 500 : GOTO250
245   PLAY "14m6168Øs1Øn2": PRINT "NO▲MATCH": FOR T =
      1 TO 2000 : NEXT : GOSUB 400 : CF = LC : GOSUB 400
250   IF CP = 1 THEN CP = 2 ELSE CP = 1
255   PRINT CHR$( 11 ) ; : FOR I = 1 TO 19 : PRINT : NEXT :
      FOR I = 1 TO 3 : PRINT "▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲":
      NEXT : PRINT "▲▲▲▲▲▲▲▲▲▲▲▲";
260   GOSUB 600 : GOTO 200
```

Flip Card

```
300   PLAY "s8m1000l20n20r30n35": K = INT( ( CF - 1 )
      / 6 ) + 1 : ON K GOTO 305, 310, 315
305   AD = 6172 + 5 * CF : GOTO 320
310   AD = 6332 + 5 * ( CF - 6 ) : GOTO 320
315   AD = 6492 + 5 * ( CF - 12 )
320   VPOKE AD, 134 : VPOKE AD + 1, 138 : VPOKE AD + 2,
      138 : VPOKE AD + 3, 138 : VPOKE AD + 4, 136 : VPOKE
      AD + 32, 140 : VPOKE AD + 64, 140 : VPOKE AD + 96,
      140 : VPOKE AD + 128, 135 : VPOKE AD + 129, 139 :
      VPOKE AD + 130, 139 : VPOKE AD + 131, 139 : VPOKE
      AD + 132, 137 : VPOKE AD + 100, 141 : VPOKE AD +
      68, 141 : VPOKE AD + 36, 141
325   FOR I = AD + 33 TO AD + 97 STEP 32 : FOR J = Ø TO
      2 : VPOKE I + J, 142 : NEXT : NEXT : VPOKE AD +
      66, C ( CF, 1 ) + 48
390   RETURN
```

Reset Card

```
400   PLAY "s8150m1000n35n20": K = INT( ( CF - 1 ) / 6 )
      + 1 : ON K GOTO 405, 410, 415
405   AD = 6172 + 5 * CF : GOTO 420
410   AD = 6332 + 5 * ( CF - 6 ) : GOTO 420
415   AD = 6492 + 5 * ( CF - 12 )
420   VPOKE AD, 144 : VPOKE AD + 1, 148 : VPOKE AD + 2,
      148 : VPOKE AD + 3, 148 : VPOKE AD + 4, 146 : VPOKE
      AD + 32, 150 : VPOKE AD + 64, 150 : VPOKE AD + 96,
      150 : VPOKE AD + 128, 145 : VPOKE AD + 129, 149 :
      VPOKE AD + 130, 149 : VPOKE AD + 131, 149 : VPOKE
      AD + 132, 147 : VPOKE AD + 100, 151 : VPOKE AD +
      68, 151 : VPOKE AD + 36, 151
425   FOR I = AD + 33 TO AD + 97 STEP 32 : FOR J = 0 TO
      2 : VPOKE I, + J, 152 : NEXT : NEXT : VPOKE AD +
      66, 64 + CF
490   RETURN
```

Match

```
500   S ( CP ) = S ( CP ) + 1
510   C ( LC, 2 ) = 1 : C ( CF, 2 ) = 1
520   K1 = 132 : PLAY "m2000110s8n35n40": GOSUB 580 :
      K1 = 133 : PLAY "110s8m1000n30n45": GOSUB 580 :
      K1 = 32 : GOSUB 580
530   IF S ( 1 ) + S ( 2 ) = 9 THEN 900
550   RETURN
580   FOR I = 6144 TO 6175 : VPOKE I, K1 : NEXT : FOR
      I = 6176 TO 6656 STEP 32 : VPOKE I, K1 : FOR J =
      1 TO 50 : NEXT : PLAY "s8164m500n25": VPOKE I +
      31, K1 : NEXT : FOR I = 6656 TO 6687 : VPOKE I,
      K1 : NEXT
585   RETURN
```

Update Name/Score

```
600   PRINT CHR$( 11 ) ; : FOR I = 1 TO 18 : PRINT : NEXT :
      PRINT P$ ( CP ) ":▲" TAB( 18 ) "Score:"; S ( CP )
690   RETURN
```

Game Over

```
900   CLS : PRINT "Congratulations to the winner -":
      IF S ( 1 ) > S ( 2 ) THEN PRINT P$ ( 1 ) : PRINT
      "with a a score of  "; S ( 1 ) ELSE PRINT P$ ( 2 ) :
      PRINT "with a a score of  "; S ( 2 )
910   PLAY "v10o4l4a18gel8egl2f11a", "o3l4a18egl8gel2f11f"
      , "o5l4c18gel8gel2f11c"
920   FOR I = 1 TO 3000 : NEXT : PRINT : PRINT : PRINT
      "Bad Luck, "; : IF S ( 1 ) < S ( 2 ) THEN PRINT
      P$ ( 1 ) : PRINT "Your score was  "; S ( 1 ) ELSE
      PRINT P$ ( 2 ) : PRINT "Your score was "; S ( 2 )
930   PLAY "v15l40s8m60000n2r40n1r30n4r20n1r30n2r10n5r10n4r10
      n3r20l3n1"
990   END
```

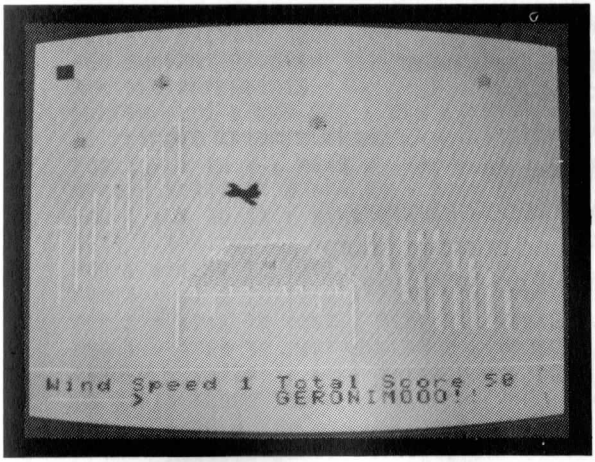
Character Data

```
10000   DATA 0, 0, 0, 0, 10, 5, 10, 5
10002   DATA 10, 5, 10, 5, 0, 0, 0, 0
10004   DATA 0, 0, 0, 0, 160, 64, 160, 64
10006   DATA 160, 64, 160, 64, 0, 0, 0, 0
10008   DATA 0, 0, 0, 0, 170, 85, 170, 85
10010   DATA 170, 85, 170, 85, 0, 0, 0, 0
10012   DATA 10, 5, 10, 5, 10, 5, 10, 5
10014   DATA 160, 64, 160, 64, 160, 64, 160, 64
10016   DATA 170, 85, 170, 85, 170, 85, 170, 85
10018   DATA 0, 0, 0, 0, 15, 15, 15, 15
10020   DATA 15, 15, 15, 15, 0, 0, 0, 0
10022   DATA 0, 0, 0, 0, 240, 240, 240, 240
10024   DATA 240, 240, 240, 240, 0, 0, 0, 0
10026   DATA 0, 0, 0, 0, 255, 255, 255, 255
10028   DATA 255, 255, 255, 255, 0, 0, 0, 0
10030   DATA 15, 15, 15, 15, 15, 15, 15, 15
10032   DATA 240, 240, 240, 240, 240, 240, 240, 240
10034   DATA 255, 255, 255, 255, 255, 255, 255, 255
```

ChexSum Tables

```
10     = 2979
15     = 393
20     = 7430
25     = 3905
35     = 8295
40     = 3774
45     = 7044
50     = 3838
60     = 6378
65     = 4023
70     = 5569
75     = 5303
80     = 1027
100    = 1603
140    = 1544
150    = 3623
155    = 4245
160    = 539
200    = 4508
205    = 5522
210    = 4092
215    = 2940
220    = 1445
225    = 5510
230    = 3900
235    = 3790
240    = 2602
245    = 5543
250    = 1860
255    = 6476
260    = 914
300    = 5209
305    = 1698
310    = 2272
315    = 1591
320    = 21802
325    = 5973
390    = 143
400    = 5335
405    = 1798
410    = 2368
415    = 1591
420    = 21617
425    = 5706
490    = 143
500    = 1179
510    = 1371
520    = 6390
530    = 1444
550    = 143
580    = 9830
585    = 143
600    = 4457
690    = 143
900    = 12566
910    = 6550
920    = 11303
930    = 4938
990    = 129
10000  = 1025
10002  = 1017
10004  = 1264
10006  = 1272
10008  = 1272
10010  = 1280
10012  = 1150
10014  = 1628
10016  = 1640
10018  = 1134
10020  = 1150
10022  = 1372
10024  = 1404
10026  = 1384
10028  = 1416
10030  = 1404
10032  = 1864
10034  = 1888

Total= 283010
```

Stunt Man



CLASSIFICATION: Simulation Game

A team of daredevils is attempting a daring leap from an aeroplane onto a trampoline. Press the <SPACE> bar to make a daredevil jump, but watch the wind speed and the height!

You lose points for hitting the ground or the picket fence and your man dies if he lands on the high-voltage wires.

PROGRAMMING SUGGESTIONS

You can alter the effect of the wind on the man's flight path and speed by changing the use of variables WS and T in line 31Ø.

PROGRAM

Variables

LD	Level of difficulty
WS	Wind Speed
T	Time parameter
F	Flying?
NW, NM	Number of wounds; number of men
M$	Message
X, Y	Man's co-ordinates
RN, SC	Round number; score
XI, YI	Initial co-ordinates

Listing

Initialise

```
 1   REM RUN MACHINE CODE
 2   REM SUPPORT PROGRAM
 3   REM SEE APPENDICES
10   SCREEN 1, 2 : KEY OFF : CLS : PRINT "▲▲▲▲▲▲▲▲▲▲STUNTMAN
     ": PRINT : PRINT : PRINT
15   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$
25   FOR I = 1088 TO 1095 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1152 TO 1191 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1216 TO 1223 : READ Q : VPOKE I, Q : NEXT :
     FOR I = 1280 TO 1287 : READ Q : VPOKE I, Q : NEXT
30   DEFUSR = 60118! : POKE 59997!, 1 : POKE 59998!, 1
40   ON STRIG GOSUB 900
50   NM = 3 : SC = 100
60   PRINT "Level▲of▲Difficulty▲(1-4)?";
65   K$ = INKEY$ : IF K$ = "" THEN 65
70   LD = VAL( K$ ) : IF LD < 1 OR LD > 4 THEN 65 ELSE
     PRINT LD
75   POKE 59999!, LD
90   PRINT : PRINT : PRINT : PRINT : PRINT : PRINT "▲▲▲▲▲Hit
     ▲any▲Key▲to▲Start"
95   IF INKEY$ = "" THEN 95
```

New Round

```
100  STRIG( 0 ) ON : CLS : COLOR 1, 5, 5
105  IF NM = 0 THEN 600
110  FOR I = 6816 TO 6847 : VPOKE I, 23 : NEXT
115  PRINT : PRINT "▲▲▲▲▲▲" CHR$( 160 ) "▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲"
      CHR$( 160 ) : PRINT : PRINT : PRINT "▲▲▲▲▲▲▲!▲▲▲▲▲▲▲"
     CHR$( 160 ) :  PRINT "▲" CHR$( 160 ) "▲▲▲▲!!": PRINT
     "▲▲▲▲▲!!!": PRINT "▲▲▲▲!!!!"
120  PRINT "▲▲▲!!!!": PRINT "▲▲!!!!": PRINT "▲!!!!!":
     PRINT  "!!!!!": PRINT "!!!!▲▲▲▲▲▲" CHR$( 146 ) CHR$(
     146 ) CHR$( 146 )  CHR$( 146 ) CHR$( 146 ) "▲▲▲"
     CHR$( 148 ) CHR$( 148 ) CHR$( 148 ) CHR$( 148 )
      CHR$( 148 )
125  PRINT "!!!▲▲▲▲▲▲" CHR$( 144 ) CHR$( 136 ) CHR$( 136)
     CHR$( 136 )  CHR$( 136 ) CHR$( 136 ) CHR$( 145 )
127  PRINT "▲▲" CHR$( 147 ) CHR$( 148 ) CHR$( 147 ) CHR$(
     147 ) CHR$( 147 )  CHR$( 148 ) : PRINT "!!!▲▲▲▲▲"
```

```
      CHR$( 144 ) CHR$( 136 ) CHR$( 136 )  CHR$( 136 )
      CHR$( 152 ) CHR$( 136 ) CHR$( 136 ) CHR$( 136 )
      CHR$( 145 ) ;
129   PRINT "▲▲" CHR$( 147 ) CHR$( 148 ) "▲▲" CHR$( 147 )
      CHR$( 148 ) :  PRINT "¦¦▲▲▲▲▲" CHR$( 144 ) ; : FOR
      I = 1 TO 9 : PRINT CHR$( 136 ) ; : NEXT :  PRINT
      CHR$( 145 ) "▲▲" CHR$( 147 ) CHR$( 148 )
130   PRINT "¦▲▲▲▲▲▲¦¦¦¦¦¦¦¦¦¦¦¦▲▲" CHR$( 147 ) CHR$( 147)
      ; : FOR I = 1  TO 5 : PRINT CHR$( 148 ) ; : NEXT :
      PRINT : PRINT "▲▲▲▲▲▲▲¦▲¦▲▲▲▲▲¦▲¦¦▲▲" ; : FOR I
      = 1 TO 7 : PRINT CHR$( 147 ) ; : NEXT : PRINT
135   PRINT "▲▲▲▲▲▲▲¦▲▲▲▲▲▲▲▲▲▲¦▲▲▲▲"; : FOR I = 1 TO
      5 : PRINT  CHR$( 147 ) ; : NEXT : PRINT : PRINT
      "▲▲▲▲▲▲▲¦▲▲▲▲▲▲▲▲▲▲¦": PRINT :  PRINT : PRINT :
      PRINT "¦¦¦¦¦>"; CHR$( 11 ) ;
140   WS = 5 : F = 0 : T = 0 : NW = 0 : GOSUB 1000
150   PUT SPRITE 1, ( 255, 40 ) , 1
160   VPOKE 8194, 229 : VPOKE 8209, 165 : VPOKE 8210,
      229 : VPOKE 8211, 149 : VPOKE 8212, 197
```

Control

```
200   D = USR( D )
205   IF VPEEK( 6917 ) > ( 255 - LD ) THEN VPOKE 6916,
      INT( RND( 1 ) * 85 ) : SC = SC - 10 : GOSUB 1000
210   IF F = 0 THEN 200
220   GOSUB 300
230   IF X < 5 OR X > 250 THEN 550
240   IF Y < 105 THEN 200
250   IF X > 85 AND X < 140 THEN GOSUB 350 : GOTO 200
260   IF X > 160 AND X < 210 THEN GOSUB 450 : GOTO 200
270   IF X < 30 THEN 400
280   GOSUB 500 : GOTO 200
```

Move Man

```
300   XV = 3 + LD / 3 : M = 50
310   T = T + 1 : X = WS * T * T / M - XV * T + XI : Y
      = 4.9 * T * T / M + YI
320   X = INT( X ) : Y = INT( Y ) : A$ = "n"+ STR$( N )
      : IF T / 3 = INT( T / 3 ) THEN PLAY "m650s8615s10xa$;":
      N = N - 1
325   IF X > 255 THEN X = 255
327   IF X < 0 THEN X = 0
330   PUT SPRITE 0, ( X, Y ) , 10
340   RETURN
```

Man Hits Target

```
350   IF X > 105 AND X < 117 THEN SC = SC + 200 : M$ =
      "SUPERB!!!!!▲▲▲▲": PLAY "18n5018n4512n50": GOTO 360
355   SC = SC + 100 : M$ = "GOOD▲JUMP▲▲▲"
360   WS = INT( RND( 1 ) * 10 ) : GOSUB 1000 : GOSUB 1300 :
      F = 0 : T = 0
370   RETURN
```

Man Hits Wires

```
400   GOSUB 1200 : CLS : PRINT "▲▲▲STUNT▲MAN▲ELECTROCUTED!!":
      NM = NM -▲1 : PUT SPRITE 0, ( 100, 200 ) : PUT SPRITE
      1, ( 255, 200 )
410   FOR TM = 1 TO 3000 : NEXT : SC = SC - 100 : GOTO 100
```

Man Hits Fence

```
450   M$ = "HIT▲THE▲FENCE!!": SC = SC - 50 : PLAY "154m1200s1
      0n74r20n74"
460   WS = INT( RND( 1 ) * 10 ) : GOSUB 1000 : F = 0 :
      T = 0 : NW = NW + 1 : GOSUB 1300
470   IF NW > 2 THEN 700
480   RETURN
```

Man Hits Ground

```
500   M$ = "OUCH-TRY▲AGAIN!": SC = SC - 30 : NW = NW +
      1 : PLAY "12n7"
510   WS = INT( RND( 1 ) * 10 ) : GOSUB 1000 : GOSUB 1300 :
      F = 0 : T = 0
520   IF NW > 2 THEN 700
530   RETURN
```

Man Out Side

```
550   GOSUB 1200 : CLS : PRINT "STUNT▲MAN▲DROWNED▲IN▲SWAMP▲-":
       PRINT "A▲LONG▲WAY▲FROM▲THE▲TARGET!!"
560   PUT SPRITE 0, ( 200, 200 ) : PUT SPRITE 1, ( 200,
      200 ) : FOR TM = 1 TO 3000 : NEXT
570   NM = NM - 1 : GOTO 100
```

Game Over

```
600   CLS : PRINT "▲▲▲▲▲▲▲▲▲GAME▲OVER": PRINT : PRINT :
      PRINT
610   PRINT "▲▲YOUR▲SCORE▲WAS"; SC
620   PUT SPRITE 1, ( 100, 200 ) : PUT SPRITE 0, ( 100,
      200 )
630   END
```

Man Out of Action

```
700   F = 1 : CLS : PRINT "STUNT▲MAN▲OUT▲OF▲ACTION▲-":
      PRINT "TOO▲MANY▲ACCIDENTS"
710   PUT SPRITE 0, ( 200, 200 ) : PUT SPRITE 1, ( 200,
      200 ) : NM = NM - 1 : FOR TM = 1 TO 3000 : NEXT
720   GOTO 100
```

Space Bar Pressed

```
900   IF F = 1 THEN RETURN
910   K1 = VPEEK( 6916 ) : K2 = VPEEK( 6917 ) : PUT SPRITE
      0, ( K2, K1 ) , 10 : F = 1 : T = 0 : M$ = "GERONIMOOO!!
      ▲▲▲": GOSUB 1300
920   YI = K1 : XI = K2 : N = 50 : RETURN
```

Update Score

```
1000  FOR I = 1 TO 22 : PRINT : NEXT : PRINT "Wind▲Speed";
      WS ; TAB( 13 ) ; "Total▲Score"; SC ; CHR$( 11 ) ;
1010  FOR J = 6895 TO 6910 : VPOKE J, 32 : NEXT
1020  RETURN
```

Man Dead

```
1200  CLS : PLAY "l9m1000s14n33n33n33n33n33n33n33n33n33n33":
      FOR I = 15 TO 1 STEP - 1 : FOR T = 1 TO 100 : NEXT :
      COLOR 15, I, I : NEXT
1250  RETURN
```

Message

```
1300  PRINT CHR$( 11 ) ; : FOR I = 1 TO 23 : PRINT : NEXT :
      FOR I = 1 TO 13 : PRINT CHR$( 28 ) ; : NEXT
```

```
1310    PRINT M$ ; CHR$( 11 ) ; : RETURN
```

Sprite Data

```
10000   DATA 153, 255, 60, 24, 60, 36, 102, 195
10010   DATA 0, 0, 0, 112, 120, 60, 30, 127, 255, 255, 127,
        1, 0, 0, 0, 0, 0, 0, 0, 4, 14, 30, 62, 254, 255,
        255, 254, 254, 240, 120, 60, 28
```

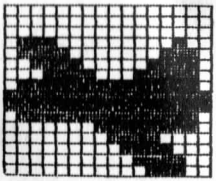
Character Data

```
10100   DATA 170, 85, 170, 85, 170, 85, 170, 85
10102   DATA 1, 2, 4, 10, 17, 34, 85, 138
10104   DATA 128, 64, 32, 80, 136, 68, 170, 81
10106   DATA 0, 0, 0, 0, 0, 0, 255, 36
10108   DATA 56, 56, 56, 56, 56, 56, 56, 56
10110   DATA 16, 56, 56, 56, 56, 56, 56, 56
10112   DATA 24, 60, 126, 126, 126, 60, 24, 0
10114   DATA 16, 56, 84, 186, 84, 186, 84, 16
```

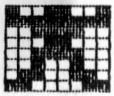
ChexSum Table

```
1     = 0
2     = 0
3     = 0
10    = 3283
15    = 4011
20    = 3483
25    = 9710
30    = 2571
40    = 864
50    = 1002
60    = 2670
65    = 1572
70    = 3287
75    = 855
90    = 3579
95    = 1034
100   = 1329
105   = 892
110   = 1768
115   = 11684
120   = 15549
125   = 6094
127   = 14521
129   = 13949
130   = 12354
135   = 9326
140   = 2280
150   = 975
160   = 2990
200   = 691
205   = 5451
210   = 921
220   = 207
230   = 1634
240   = 1043
250   = 2529
260   = 2797
270   = 916
280   = 1069
300   = 1562
310   = 5718
320   = 7813
325   = 1564
327   = 1074
330   = 832
340   = 143
350   = 6462
355   = 2357
360   = 3386
370   = 143
400   = 6920
410   = 2884
450   = 4400
460   = 4468
470   = 1001
480   = 143
500   = 4350
510   = 3386
520   = 1001
530   = 143
550   = 6423
560   = 3636
570   = 1392
600   = 2207
610   = 1645
620   = 2042
630   = 129
700   = 5121
710   = 4739
720   = 489
900   = 835
910   = 6864
920   = 1800
1000  = 5591
1010  = 1883
1020  = 143
1200  = 7838
1250  = 143
1300  = 3966
1310  = 1111
10000 = 1630
10010 = 7656
10100 = 1640
10102 = 1289
10104 = 1575
10106 = 1084
10108 = 1424
10110 = 1420
10112 = 1516
10114 = 1525

Total= 287396
```

Sprite Shapes

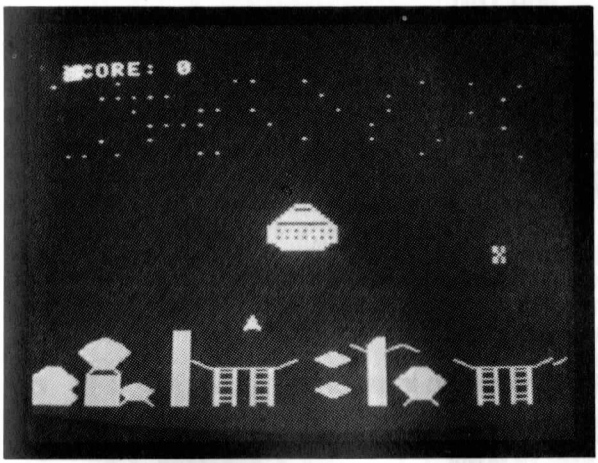


PLANE



STUNTMAN

Secret Ship-Shuttle Search



CLASSIFICATION: Target Practice Game COLOUR ILLUST. OBC

Spies from the planet Zircon have stolen the formula for a valuable compound. The plan is to deposit counter-espionage agents in small shuttle-craft onto special landing sites on planet Zircon so that the formula can be recovered.

You must deposit as many shuttle-craft as possible within the time limit and prevent the mother ship from colliding with enemy craft!

A machine-code subroutine is used to scroll the screen.

PROGRAMMING SUGGESTIONS

For some changes in speed modify these lines:

MOTHER SHIP	line 45
SCROLL	line 160

The time limit TL is set in line 160, so you can change the duration of the game.

A more ambitious alteration would be to add an enemy craft, so that avoiding collisions would be quite a challenge. The relevant section is lines 300-320.

PROGRAM

Variables

SH	Shuttle screen location
SC	Score
TL	Time limit
GE	Game ended (0 - 2)

Listing

Initialise

```
 10   REM RUN MACHINE CODE
 11   REM SUPPORT PROGRAM
 12   REM SEE APPENDICES
 15   SCREEN 1, 3 : KEY OFF : PRINT "▲SECRET▲SHIP▲SHUTTLE▲SEA
      RCH": FOR I = 1 TO 12 : PRINT : NEXT
 20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 0 ) = A$ : A$ = ""
 25   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 1 ) = A$
 30   FOR I = 1072 TO 1231 : READ Q : VPOKE I, Q : NEXT
 35   FOR I = 60350! TO 60401! : READ Q : POKE I, Q : NEXT
 40   POKE 60346!, 224 : POKE 60347!, 24 : POKE 60348!,
      255 : POKE 60349!, 26 : DEFUSR2 = 60350!
 45   POKE 59999!, 4 : POKE 59996!, 15 : DEFUSR = 60000!
 50   PUT SPRITE 0, ( 112, 55 ) , 11 : PRINT "▲▲▲▲Hit▲any▲Key
      ▲to▲Begin".
 55   IF INKEY$ = "" THEN D = USR2( D ) : FOR I = 1 TO
      50 : NEXT : D = RND( 1 ) : GOTO 55
```

Set Up Screen

```
100   COLOR 15, 1, 1 : CLS : VPOKE 8208, 145 : VPOKE 8209,
      145 : VPOKE 8210, 145 : PUT SPRITE 0, ( 114, 80 ), 11
105   VPOKE 6753, 137 : VPOKE 6768, 144 : VPOKE 6769,
      145 : VPOKE 6773, 134
110   VPOKE 6784, 136 : VPOKE 6785, 134 : VPOKE 6786,
      138 : VPOKE 6787, 136 : VPOKE 6799, 139 : VPOKE
      6800, 134 : VPOKE 6801, 134 : VPOKE 6802, 141 :
      VPOKE 6805, 134 : VPOKE 6813, 144 : VPOKE 6814, 145
115   VPOKE 6817, 134 : VPOKE 6819, 137 : VPOKE 6820,
      140 : VPOKE 6822, 136 : VPOKE 6823, 148 : VPOKE
      6824, 149 : VPOKE 6825, 148 : VPOKE 6826, 149 :
      VPOKE 6827, 138 : VPOKE 6829, 137 : VPOKE 6830,
      140 : VPOKE 6832, 142 : VPOKE 6833, 143 : VPOKE
      6837, 134 : VPOKE 6838, 136
120   VPOKE 6839, 148 : VPOKE 6840, 149 : VPOKE 6841,
      148 : VPOKE 6842, 149 : VPOKE 6843, 138 : VPOKE
      6845, 146 : VPOKE 6846, 147
125   VPOKE 6849, 134 : VPOKE 6850, 139 : VPOKE 6851,
      134 : VPOKE 6852, 134 : VPOKE 6853, 141 : VPOKE
      6855, 150 : VPOKE 6856, 151 : VPOKE 6857, 150 :
      VPOKE 6858, 151 : VPOKE 6860, 139 : VPOKE 6861,
```

```
      134 : VPOKE 6862, 134 : VPOKE 6863, 141 : VPOKE
      6864, 134 : VPOKE 6865, 134
130   VPOKE 6866, 144 : VPOKE 6867, 145 : VPOKE 6869,
      134 : VPOKE 6871, 150 : VPOKE 6872, 151 : VPOKE
      6873, 150 : VPOKE 6874, 151 : VPOKE 6877, 144 :
      VPOKE 6878, 145
135   VPOKE 6881, 134 : VPOKE 6883, 142 : VPOKE 6884,
      143 : VPOKE 6887, 150 : VPOKE 6888, 151 : VPOKE
      6889, 150 : VPOKE 6890, 151 : VPOKE 6892, 139 :
      VPOKE 6893, 134 : VPOKE 6894, 134 : VPOKE 6895,
      141 : VPOKE 6896, 134 : VPOKE 6897, 134
140   VPOKE 6898, 142 : VPOKE 6899, 143 : VPOKE 6901,
      134 : VPOKE 6903, 150 : VPOKE 6904, 151 : VPOKE
      6905, 150 : VPOKE 6906, 151 : VPOKE 6909, 146 :
      VPOKE 6910, 147
145   FOR I = 6176 TO 6367 : IF RND( 1 ) < .2 THEN VPOKE
      I, 153
150   NEXT
155   GOSUB 1000 : PUT SPRITE 1, ( 255, 100 ) , 13
160   TIME = 0 : ON INTERVAL = 10 GOSUB 500 : INTERVAL
      ON : TL = 100
170   STRIG(0)ON:ON STRIG GOSUB700
175   SPRITE ON : ON SPRITE GOSUB 600
```

Control

```
200   D = USR( D ) : IF VPEEK( 6912 ) > 136 THEN GE =
      1 : GOTO 900
202   X = VPEEK( 6913 ) : IF X < 57 THEN POKE 59996!,
      13 ELSE IF X > 184 THEN POKE 59996!, 7 ELSE POKE
      59996!, 15
210   GOSUB 300
220   IF SH <> 0 THEN GOSUB 400
290   GOTO 200
```

Move Enemy Craft

```
300   SPRITE OFF : IF VPEEK( 6917 ) < 9 THEN VPOKE 6916,
      INT( RND( 1 ) * 128 + 24 ) : VPOKE 6917, 255
310   SPRITE ON : VPOKE 6917, VPEEK( 6917 ) - 7
320   RETURN
```

Move Shuttle

```
400   INTERVAL OFF : STRIG( 0 ) OFF : K = VPEEK( SH +
      32 ) : INTERVAL ON : IF K = 148 OR K = 149 THEN
```

```
      VPOKE 8208, 241 : VPOKE 8209, 241 : VPOKE 8210,
      241 : SC = SC + 10 : PLAY "S1M2000L14N50N45": GOSUB
      1000 : VPOKE SH, 32 : SH = 0 : VPOKE 8208, 145 :
      VPOKE 8209, 145 : VPOKE 8210, 145 : STRIG( 0 ) ON :
      RETURN
410   IF K = 32 THEN SH = SH + 32 : VPOKE SH, 152 : VPOKE
      SH - 32, 32 : STRIG( 0 ) ON : RETURN
```

Crash

```
450   VPOKE SH, 135 : FOR I = 1 TO 100 : NEXT : SC = SC
      - 10 : GOSUB 1000 : VPOKE SH, 32 : SH = 0 : STRIG( 0)
      ON : RETURN
```

Scroll Screen

```
500   D = USR2( D ) : IF TIME / 60 > TL THEN 900
505   IF SH <> 0 THEN SH = SH - 1
510   RETURN
```

Collision

```
600   INTERVAL OFF : STRIG( 0 ) OFF : FOR I = 1 TO 20 :
      COLOR 15, 1, 1 : PLAY "l64s8m20000n21n12": COLOR
      1, 15, 15 : NEXT
610   GE = 2 : GOTO 900
```

Deposit Shuttle

```
700   IF SH <> 0 THEN RETURN
710   I3 = INT( ( VPEEK( 6913 ) + 20 ) / 8 ) : SH = 6144
      + 32 * INT( ( ( VPEEK( 6912 ) + 20 ) / 8 ) ) + I3 :
      IF SH < 6368 THEN SH = 0
740   RETURN
```

Game Over

```
900   COLOR 15, 4, 7 : SCREEN 1 : PRINT "Your score was";
      SC
910   PRINT : PRINT : PRINT : ON( GE + 1 ) GOTO 920, 930,
      940
920   PRINT "YOUR TIME IS UP!": GOTO 990
930   PRINT "MOTHER SHIP CRASHED!": GOTO 990
940   PRINT "MOTHER SHIP COLLIDED WITH": PRINT "ENEMY CRAFT"
990   IF INKEY$ = "" THEN END ELSE 990
```

Update Score

```
 1000   PRINT "SCORE:"; SC ; CHR$( 11 ) ;
 1010   RETURN
```

Sprite Data

```
10000   DATA 3, 4, 15, 31, 63, 64, 255, 213, 255, 213, 255,
        63, 31, 0, 0, 0, 192, 32, 240, 248, 252, 2, 255,
        85, 255, 85, 255, 252, 248, 0, 0, 0
10010   DATA 160, 160, 64, 160, 160, 0, 0, 0
```

Character Data

```
10100   DATA 255, 255, 255, 255, 255, 255, 255, 255
10102   DATA 68, 16, 130, 40, 84, 130, 16, 68
10104   DATA 192, 48, 12, 3, 0, 0, 0, 0
10106   DATA 0, 0, 0, 0, 15, 31, 63, 127
10108   DATA 3, 12, 48, 192, 0, 0, 0, 0
10110   DATA 1, 3, 7, 15, 15, 7, 3, 1
10112   DATA 0, 0, 0, 0, 240, 248, 252, 254
10114   DATA 128, 192, 224, 240, 240, 224, 192, 128
10116   DATA 255, 127, 63, 31, 31, 32, 64, 128
10118   DATA 255, 254, 252, 248, 248, 4, 2, 1
10120   DATA 0, 0, 0, 0, 3, 15, 63, 255
10122   DATA 0, 0, 0, 0, 192, 240, 252, 255
10124   DATA 255, 63, 15, 3, 0, 0, 0, 0
10126   DATA 255, 252, 240, 192, 0, 0, 0, 0
10128   DATA 0, 0, 0, 0, 255, 24, 31, 24
10130   DATA 0, 0, 0, 0, 255, 24, 248, 24
10132   DATA 24, 24, 31, 24, 24, 31, 24, 24
10134   DATA 24, 24, 248, 24, 24, 248, 24, 24
10136   DATA 16, 16, 40, 40, 56, 124, 254, 130
10138   DATA 0, 0, 0, 32, 0, 0, 0, 0
```

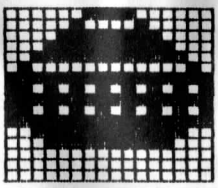
Machine-Code Data

```
10200   DATA 6, 31, 42, 186, 235, 43, 35, 205, 27, 235,
        120, 254, 31, 202, 215, 235, 4, 43, 205, 44, 235,
        35, 195, 227, 235, 6, 0, 17, 31, 0, 25, 205, 44,
        235, 183, 237, 82, 237, 91, 188, 235, 123, 189,
        194, 196, 235, 122, 188, 194, 196, 235, 201
```

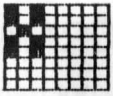
ChexSum Table

```
10      = 0
11      = 0
12      = 0
15      = 4914
20      = 3995
25      = 3499
30      = 2011
35      = 2217
40      = 4607
45      = 2521
50      = 3370
55      = 4291
100     = 3435
105     = 2878
110     = 3089
115     = 12852
120     = 5615
125     = 15431
130     = 8073
135     = 2116
140     = 1341
145     = 2900
150     = 131
155     = 1576
160     = 4187
170     = 1684
175     = 1014
200     = 3257
202     = 5778
210     = 207
220     = 1347
290     = 593
300     = 4717
310     = 1556
320     = 143
400     = 19648
410     = 4136
450     = 5407
500     = 2295
505     = 1881
510     = 143
600     = 5947
610     = 1007
700     = 1168
710     = 7586
740     = 143
900     = 2808
910     = 2490
920     = 2160
930     = 2531
940     = 3757
990     = 1529
1000    = 1612
1010    = 143
10000   = 8092
10010   = 1463
10100   = 1888
10102   = 1514
10104   = 1156
10106   = 1195
10108   = 1149
10110   = 1035
10112   = 1386
10114   = 1888
10116   = 1563
10118   = 1541
10120   = 1136
10122   = 1378
10124   = 1147
10126   = 1410
10128   = 1213
10130   = 1274
10132   = 1398
10134   = 1523
10136   = 1568
10138   = 974
10200   = 21635

Total= 240262
```

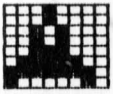
Sprite Shapes



SECRET SHIP

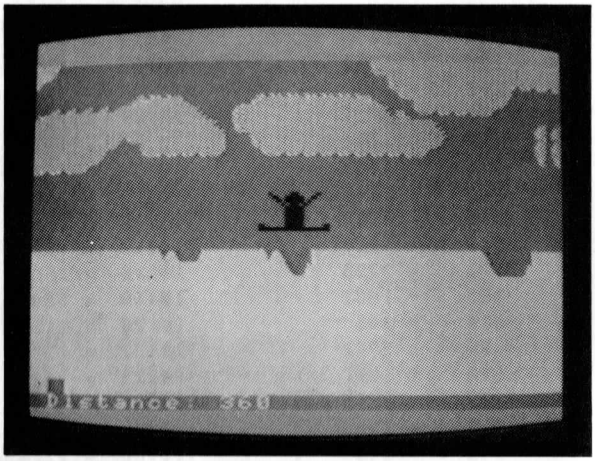


ENEMY CRAFT



SHUTTLE
(CHARACTER)

Cross-Country



CLASSIFICATION: Simulation Game

Here is a game for keen joggers and other runners. Go for your Personal Best on the 1000 m hazardous track. Press the space bar to jump over pot-holes and control your speed with the cursor keys, but keep your eyes open for lightning!

A machine-code subroutine is used for scrolling.

PROGRAMMING SUGGESTIONS

If you're not happy with the ground or clouds then fix the set-up screen routine in lines 100-190.

The usual speed changes can be made in line 90 (to move man further) and line 180 (to speed up the scroll).

PROGRAM
Variables

J	Jumped? (used as a count)
DT	Distance run
PC	Player's column
LT	Lightning (count)
LD	Level of difficulty
GE	Game ended

Listing

Initialise

```
 5   REM RUN MACHINE CODE
 6   REM SUPPORT PROGRAM
 7   REM SEE APPENDICES
10   KEY OFF : COLOR 15, 4, 7 : SCREEN 1, 3 : FOR I =
     1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) : NEXT :
     SPRITE$( 0 ) = A$ : A$ = ""
15   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : A$ = ""
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 2 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 3 ) = A$
30   FOR I = 60350! TO 60401! : READ Q : POKE I, Q :
     NEXT : DEFUSR2 = 60350! : POKE 60346!, 0 : POKE
     60347!, 24 : POKE 60348!, 223 : POKE 60349!, 25
35   FOR I = 1072 TO 1215 : READ Q : VPOKE I, Q : NEXT
50   FOR I = 1 TO 11 : PRINT : NEXT : PRINT TAB( 8 )
     "CROSS-COUNTRY"
55   DEFUSR = 60000! : DEFUSR1 = 60118! : POKE 59999!,
     1 : POKE 59996!, 10 : POKE 59997!, 0 : POKE 59998!, 3
60   PUT SPRITE 0, ( 0, 55 ) , 1
65   IF INKEY$ = "" THEN IF VPEEK( 6914 ) = 0 THEN VPOKE
     6914, 4 ELSE VPOKE 6914, 0 ELSE GOTO 80
70   D = USR1( D ) : GOTO 65
80   PRINT : PRINT : INPUT "Level▲of▲Difficulty▲(1-3)";LD
85   IF LD < 1 OR LD > 3 THEN 80
90   POKE 59999!, LD + 2
```

Set Up Screen

```
100  PUT SPRITE 0, ( 100, 200 ) : CLS : FOR I = 6609
     TO 6879 : VPOKE I, 134 : NEXT
105  VPOKE 6592, 134 : VPOKE 6593, 134 : VPOKE 6594,
     134 : VPOKE 6595, 139 : VPOKE 6596, 138 : FOR I
     = 6597 TO 6606 : VPOKE I, 134 : NEXT : VPOKE 6607,
     139 : VPOKE 6609, 138
110  VPOKE 6560, 135 : VPOKE 6561, 135 : VPOKE 6562,
     136 : VPOKE 6563, 137 : FOR I = 6565 TO 6574 : VPOKE
     I, 135 : NEXT : VPOKE 6575, 137 : FOR I = 6578 TO
     6586 : VPOKE I, 135 : NEXT : VPOKE 6587, 136 : VPOKE
     6588, 137 : VPOKE 6589, 138
115  VPOKE 6590, 135 : VPOKE 6591, 135
```

```
120  VPOKE 6152, 142 : FOR I = 6153 TO 6163 : VPOKE I,
     140 : NEXT : VPOKE 6164, 146 : VPOKE 6185, 142 :
     FOR I = 6186 TO 6192 : VPOKE I, 140 : NEXT : VPOKE
     6193, 144 : VPOKE 6194, 144 : VPOKE 6195, 145
125  VPOKE 6209, 150 : FOR I = 6210 TO 6215 : VPOKE I,
     149 : NEXT : VPOKE 6216, 147 : VPOKE 6218, 143 :
     FOR I = 6219 TO 6223 : VPOKE I, 140 : NEXT : VPOKE
     6224, 146 : VPOKE 6233, 150 : VPOKE 6234, 149 :
     VPOKE 6235, 149 : VPOKE 6236, 147
130  VPOKE 6240, 141 : FOR I = 6241 TO 6248 : VPOKE I,
     140 : NEXT : VPOKE 6249, 148 : VPOKE 6250, 147 :
     VPOKE 6251, 143 : VPOKE 6252, 144 : VPOKE 6253,
     144 : VPOKE 6254, 144 : VPOKE 6255, 145 : VPOKE
     6259, 150 : FOR I = 6260 TO 6263 : VPOKE I, 151 : NEXT
135  VPOKE 6264, 151 : FOR I = 6265 TO 6268 : VPOKE I,
     140 : NEXT : VPOKE 6269, 148
140  VPOKE 6272, 142 : FOR I = 6273 TO 6283 : VPOKE I,
     140 : NEXT : VPOKE 6284, 148 : VPOKE 6290, 141 :
     FOR I = 6291 TO 6301 : VPOKE I, 140 : NEXT : VPOKE
     6302, 148
145  VPOKE 6305, 142 : FOR I = 6306 TO 6315 : VPOKE I,
     140 : NEXT : VPOKE 6316, 146 : VPOKE 6322, 141 :
     FOR I = 6323 TO 6327 : VPOKE I, 140 : NEXT : VPOKE
     6328, 146 : VPOKE 6329, 143 : FOR I = 6330 TO 6334 :
     VPOKE I, 140 : NEXT
150  VPOKE 6338, 143 : FOR I = 6339 TO 6346 : VPOKE I,
     144 : NEXT : VPOKE 6347, 145 : VPOKE 6354, 142 :
     FOR I = 6355 TO 6358 : VPOKE I, 140 : NEXT : VPOKE
     6359, 146 : VPOKE 6362, 143 : VPOKE 6363, 144 :
     VPOKE 6364, 144 : VPOKE 6365, 144 : VPOKE 6366, 145
155  VPOKE 6387, 143 : VPOKE 6388, 144 : VPOKE 6389,
     144 : VPOKE 6390, 145
170  PUT SPRITE 0, ( 154, 72 )
180  INTERVAL  ON : ON INTERVAL = 15 - 2 * LD GOSUB 500
185  SPRITE ON : ON SPRITE GOSUB 800
190  STRIG( 0 ) ON : ON STRIG GOSUB 700 : TIME = 0
```

Control

```
200  D = USR( D ) : K = INT( ( VPEEK( 6913 ) + 8 ) /
     8 ) : IF K - PC > 0 THEN DT = DT + 8 * ( K - PC )
205  PC = K : IF VPEEK( 6560 + PC ) <> 135 AND J = 0
     AND PC <> 32 THEN GOSUB 600
210  IF J <> 0 THEN J = J + 1 : IF J = 10 - 2 * LD THEN
     J = 0 : VPOKE 6914, 0
220  GOSUB 1000
230  IF RND( 1 ) < .07 THEN GOSUB 300
240  IF LT <> 0 THEN GOSUB 320
290  GOTO 200
```

Lightning

```
300   IF LT <> 0 THEN RETURN
305   PUT SPRITE 3, ( INT( RND( 1 ) * 255 ) , 50 ) , 11 :
      LT = 1 : PLAY "12m2000s10n45": RETURN
320   LT = LT + 1
330   IF LT = 5 THEN PUT SPRITE 3, ( 100, 200 ) , 11 :
      LT = 0 : RETURN
350   RETURN
```

Scroll Screen

```
500   D = USR2( D )
510   DT = DT + 1
515   IF J <> 0 THEN RETURN
520   STRIG( 0 ) OFF : I1 = VPEEK( 6914 ) : IF I1 = 4
      THEN I1 = 0 ELSE I1 = 4
530   VPOKE 6914, I1 : STRIG( 0 ) ON : RETURN
```

Player In Pothole

```
600   SPRITE OFF : INTERVAL OFF : STRIG( 0 ) OFF : VPOKE
      6914, 8 : VPOKE 6912, 94 : PLAY "12m40000s8n4":
      FOR I2 = 1 TO 1500 : NEXT : VPOKE 6912, 72 : VPOKE
      6914, 0
610   VPOKE 6913, ( VPEEK( 6913 ) + 1 ) MOD 255 : PC =
      INT( ( VPEEK( 6913 ) + 8 ) / 8 ) : IF VPEEK( 6560
      + PC ) <> 135 THEN 610
620   SPRITE ON : INTERVAL ON : STRIG( 0 ) ON : RETURN
```

Jump

```
700   IF J <> 0 THEN RETURN
710   J = 1 : VPOKE 6914, 8 : RETURN
```

Player Struck By Lightning

```
800   INTERVAL OFF : FOR I = 1 TO 20 : PLAY "164m800s14n50":
      FOR T = 1 TO 50 : NEXT : COLOR 15, 1, 1 : FOR T
      = 1 TO 50 : NEXT : COLOR 15, 4, 7 : NEXT
810   GE = 1
```

Game Over

```
900   CLS : IF GE = 1 THEN PRINT "You▲have▲▲been▲electrocuted
      .": PRINT : PRINT "Distance:"; DT ; "metres▲in":
      PRINT USING "#####.##"; TIME / 60 ; : PRINT "▲seconds.":
      GOTO 990
 910  PRINT "The▲race▲is▲over!!!!": PRINT : PRINT "Your▲time
      was"; : PRINT USING "#####.##"; TIME / 60 ; : PRINT
      "▲seconds."
 990  IF INKEY$ <> "" THEN END ELSE 990
```

Update Distance

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT "Distance:";
      DT ; CHR$( 11 ) ;
1010  IF DT > 1000 THEN 900
1020  RETURN
```

Sprite Data

```
10000  DATA 1, 3, 3, 1, 7, 11, 19, 11, 7, 3, 195, 70, 44,
       24, 0, 0, 128, 192, 192, 128, 192, 196, 232, 208,
       192, 192, 96, 48, 24, 16, 32, 96
10010  DATA 1, 3, 3, 1, 3, 3, 3, 3, 3, 3, 1, 1, 1, 2, 4,
       2, 128, 192, 192, 128, 192, 192, 240, 192, 192,
       128, 128, 128, 128, 128, 128, 192
10020  DATA 33, 19, 11, 5, 3, 3, 3, 3, 3, 131, 255, 0,
       0, 0, 0, 0, 132, 200, 208, 160, 192, 192, 192, 192,
       192, 193, 255, 0, 0, 0, 0, 0
10030  DATA 6, 12, 24, 48, 12, 3, 0, 0, 0, 0, 0, 0, 0,
       1, 3, 6, 0, 0, 0, 0, 0, 0, 192, 48, 12, 24, 48,
       96, 192, 128, 0, 0
```

Machine-Code Data

```
10100  DATA 6, 31, 42, 186, 235, 43, 35, 205, 27, 235,
       120, 254, 31, 202, 215, 235, 4, 43, 205, 44, 235,
       35, 195, 227, 235, 6, 0, 17, 31, 0, 25, 205, 44,
       235, 183, 237, 82, 237, 91, 188, 235, 123, 189,
       194, 196, 235, 122, 188, 194, 196, 235, 201
```

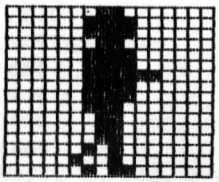
Character Data

```
10200   DATA 255, 255, 255, 255, 255, 255, 255, 255
10202   DATA 0, 170, 255, 255, 255, 255, 255, 255
10204   DATA 0, 192, 225, 227, 243, 247, 247, 255
10206   DATA 0, 0, 128, 192, 192, 224, 224, 240
10208   DATA 0, 2, 3, 3, 7, 7, 7, 15
10210   DATA 240, 240, 248, 248, 252, 252, 254, 255
10212   DATA 170, 85, 170, 85, 170, 85, 170, 85
10214   DATA 1, 5, 10, 13, 21, 26, 13, 21
10216   DATA 10, 21, 10, 13, 6, 10, 5, 2
10218   DATA 170, 85, 106, 53, 10, 0, 0, 0
10220   DATA 170, 85, 170, 205, 134, 0, 0, 0
10222   DATA 85, 170, 84, 172, 80, 0, 0, 0
10224   DATA 85, 170, 86, 168, 84, 168, 80, 160
10226   DATA 0, 0, 0, 160, 80, 176, 84, 170
10228   DATA 128, 80, 160, 64, 168, 80, 170, 84
10230   DATA 0, 0, 140, 198, 85, 170, 85, 170
10232   DATA 0, 0, 0, 2, 1, 5, 10, 13
10234   DATA 1, 5, 10, 85, 170, 85, 170, 85
```

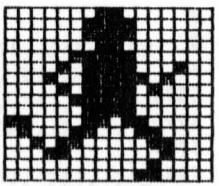
ChexSum Tables

```
5      = 0
6      = 0
7      = 0
10     = 5594
15     = 3996
20     = 3997
25     = 3481
30     = 7334
35     = 1995
50     = 2926
55     = 5636
60     = 754
65     = 3961
70     = 1255
80     = 3280
85     = 1566
90     = 1133
100    = 3295
105    = 8634
110    = 13438
115    = 1362
120    = 8848
125    = 3696
130    = 4661
135    = 3257
140    = 7762
145    = 8856
150    = 512
155    = 560
170    = 836
180    = 3782
185    = 926
190    = 2240
200    = 6088
205    = 4705
210    = 3934
220    = 397
230    = 1589
240    = 1274
290    = 593
300    = 1175
305    = 4449
320    = 833
330    = 2619
350    = 143
500    = 716
510    = 817
515    = 1080
520    = 3762
530    = 1423
600    = 8074
610    = 6950
620    = 2515
700    = 1080
710    = 931
800    = 7565
810    = 403
900    = 12612
910    = 7144
990    = 1529
1000   = 3411
1010   = 1194
1020   = 143
10000  = 7802
10010  = 7586
10020  = 7182
10030  = 5457
10100  = 21635
10200  = 1888
10202  = 1778
10204  = 1780
10206  = 1625
10208  = 965
10210  = 1881
10212  = 1640
10214  = 1265
10216  = 1202
10218  = 1331
10220  = 1470
10222  = 1344
10224  = 1657
10226  = 1393
10228  = 1651
10230  = 1518
10232  = 1029
10234  = 1399

Total= 285199
```

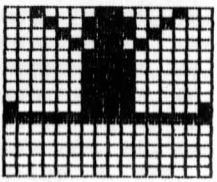
Sprite Shapes



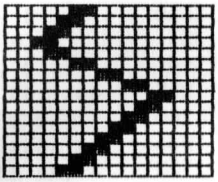
MAN: POSITION 2



MAN: POSITION 1

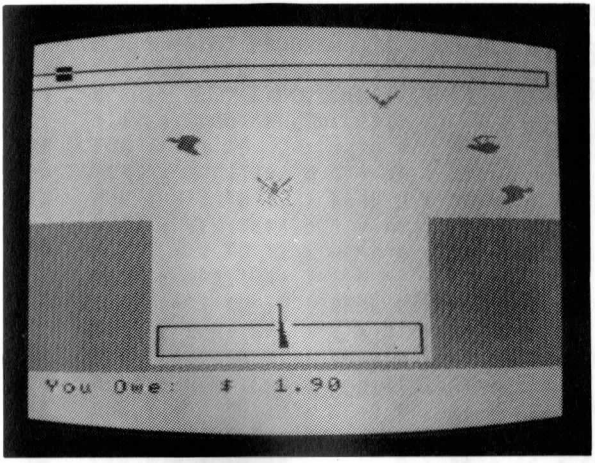


MAN: FALLEN



LIGHTNING

Shooting Gallery



CLASSIFICATION: Target Practice Game

Step right up and have a go! Only **$1**.00 for five shots — and you get five shots free!

Scoring:

front ducks	2Ø ¢
middle ducks	5Ø ¢
crazy bird	**$1**.ØØ

PROGRAMMING SUGGESTIONS

Lines 71Ø-75Ø can be varied to change the difficulty of hitting a bird, and flight speeds are controlled by line 35Ø (middle birds) and line 41Ø (front birds).

Try putting in an extra row of birds with different values to add some variety.

PROGRAM

Variables

M	Money left
NS	Number of shots (Ø - 5)
DH	Bird hit (−1 if none)
C1, C2	Timing of middle and front birds
UM	Update money?

Listing

Initialise

```
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   SCREEN 1, 2 : COLOR 1, 15, 15 : KEY OFF : PRINT
     "▲▲▲▲▲▲SHOOTING▲GALLERY"
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : A$ = ""
30   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 2 ) = A$ : A$ = ""
35   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 3 ) = A$ : A$ = ""
40   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 4 ) = A$ : A$ = ""
45   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 5 ) = A$ : A$ = ""
50   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 6 ) = A$ : A$ = ""
55   FOR I = 1072 TO 1119 : READ Q : VPOKE I, Q : NEXT
60   DEFUSR = 60118! : POKE 59997!, 2 : POKE 59998!,
     1 : POKE 59999!, 5 : PUT SPRITE 2, ( 255, 80 ) , 4
65   D = USR( D ) : IF INKEY$ <> "" THEN 80
70   D = RND( 1 ) : IF VPEEK( 6922 ) = 8 THEN VPOKE 6922,
     12 ELSE VPOKE 6922, 8
75   FOR I = 1 TO 120 : NEXT : GOTO 65
80   DH = - 1
```

Set Up Screen

```
100  CLS : VPOKE 6922, 8 : PUT SPRITE 2, ( 255, 36 )
     , 12 : PUT SPRITE 3, ( 140, 36 ) , 12 : PUT SPRITE
     0, ( 100, 16 ) , 8 : PUT SPRITE 4, ( 0, 64 ) , 4 :
     PUT SPRITE 5, ( 115, 64 ) , 4
110  FOR I = 6145 TO 6206 : VPOKE I, 23 : NEXT : FOR
     I = 6729 TO 6742 : VPOKE I, 23 : NEXT : FOR I =
     6793 TO 6806 : VPOKE I, 23 : NEXT
115  FOR I = 6816 TO 6847 : VPOKE I, 139 : NEXT : FOR
     I = 6496 TO 6784 STEP 32 : FOR J = 0 TO 7 : VPOKE
     I + J, 134 : VPOKE I + J + 24, 134 : NEXT : NEXT
120  VPOKE 6144, 24 : VPOKE 6728, 24 : VPOKE 6176, 26 :
     VPOKE 6792, 26 : VPOKE 6175, 25 : VPOKE 6743, 25 :
     VPOKE 6207, 27 : VPOKE 6807, 27
```

```
125   VPOKE 6760, 22 : VPOKE 6775, 22 : VPOKE 6703, 135 :
      VPOKE 6735, 136 : VPOKE 6767, 137
130   STRIG( 0 ) ON : ON STRIG GOSUB 700
140   GOSUB 800
```

Control

```
200   GOSUB 300 : GOSUB 350 : GOSUB 400
210   IF DH > = 0 THEN GOSUB 450
220   IF UM = 1 THEN GOSUB 800 : UM = 0
290   GOTO 200
```

Move Crazy Bird

```
300   POKE 59999!, 7 : POKE 59997!, 0 : IF RND( 1 ) <
      .5 THEN POKE 59998!, 1 ELSE POKE 59998!, 3
320   IF VPEEK( 6913 ) > 194 THEN POKE 59998!, 1 ELSE
      IF VPEEK( 6913 ) < 65 THEN POKE 59998!, 3
330   D = USR( D )
335   IF RND( 1 ) < .5 THEN VPOKE 6914, 4 ELSE VPOKE 6914,0
340   RETURN
```

Move Middle Birds

```
350   POKE 59999!, 13 : POKE 59997!, 2 : K1 = VPEEK( 6922 )
      : VPOKE 6922, VPEEK( 6926 ) : VPOKE 6926, K1
360   POKE 59998!, 1 : D = USR( D ) : POKE 59997!, 3 :
      D = USR( D )
370   RETURN
```

Move Front Birds

```
400   C2 = ( C2 + 1 ) MOD 2 : POKE 59997!, 4 : IF C2 =
      0 THEN K1 = VPEEK( 6930 ) : VPOKE 6930, VPEEK( 6934 )
      : VPOKE 6934, K1
410   POKE 59999!, 4 : POKE 59998!, 3 : D = USR( D ) :
      POKE 59997!, 5 : D = USR( D )
430   RETURN
```

New Bird

```
450   ON( DH + 1 ) GOTO 465, 469, 470, 475, 480, 485
465   PUT SPRITE 0, ( 100, 16 ) : GOTO 490
470   VPOKE 6922, 8 : VPOKE 6926, 12
472   GOTO 490
```

```
475   VPOKE 6922, 8 : VPOKE 6926, 12
477   GOTO 490
480   VPOKE 6930, 16 : VPOKE 6934, 20
483   GOTO 490
485   VPOKE 6930, 16 : VPOKE 6934, 20
490   DH = - 1 : RETURN
```

Bird Hit

```
500   PLAY "150m1200s10n74r30n74r30n74r30n74r30n74r30n74r30n7
      4": ON( DH + 1 ) GOTO 505, 508, 510, 515, 520, 525
505   M = M + 1 : VPOKE 6914, 24 : FOR I2 = 1 TO 500 : NEXT
507   VPOKE 6914, 0 : GOTO 540
510   M = M + .5 : VPOKE 6922, 24 : FOR I2 = 1 TO 500 : NEXT
512   VPOKE 6922, 8 : GOTO 540
515   M = M + .5 : VPOKE 6926, 24 : FOR I2 = 1 TO 500 : NEXT
517   VPOKE 6926, 12 : GOTO 540
520   M = M + .2 : VPOKE 6930, 24 : FOR I2 = 1 TO 500 : NEXT
522   VPOKE 6930, 16 : GOTO 540
525   VPOKE 6934, 24 : M = M + .2 : FOR I2 = 1 TO 500 : NEXT
527   VPOKE 6934, 20
540   UM = 1 : STRIG( 0 ) ON : RETURN
```

Bird Miss

```
600   STRIG( 0 ) ON : VPOKE 6191, 138 : FOR I1 = 1 TO
      100 : NEXT : VPOKE 6191, 23
620   UM = 1 : RETURN
```

Fire Gun

```
700   STRIG( 0 ) OFF : PLAY "164m60s8n27": NS = NS + 1
710   IF ABS( VPEEK( 6929 ) - 114 ) < 3 THEN DH = 4 :
      GOTO 500
720   IF ABS( VPEEK( 6933 ) - 114 ) < 3 THEN DH = 5 :
      GOTO 500
730   IF ABS( VPEEK( 6921 ) - 114 ) < 3 THEN DH = 2 :
      GOTO 500
740   IF ABS( VPEEK( 6925 ) - 114 ) < 3 THEN DH = 3 :
      GOTO 500
750   IF ABS( VPEEK( 6913 ) - 114 ) < 3 THEN DH = 0 :
      GOTO 500
760   GOTO 600
```

Update Money

```
800   IF M > = Ø THEN A$ = "You▲Have:▲$" ELSE A$ = "You▲Owe:
      ▲$"
805   IF M < Ø THEN I3 = INT( - M ) : I4 = - ( M + I3 )
      ELSE I3 = INT( M ) : I4 = M - I3
807   FOR I = 1 TO 22 : PRINT : NEXT : PRINT A$ ; : PRINT
      USING "###"; I3 ; : PRINT USING ".##"; I4 : PRINT
      CHR$( 11 ) ;
810   IF NS > 4 THEN 850 ELSE RETURN
```

More Shots?

```
850   PLAY "l10s14m2000n30n40n20n27": STRIG( Ø ) OFF :
      FOR I = 1 TO 23 : PRINT : NEXT : PRINT "Another▲5▲Shots
      ?▲(Y/N)"; CHR$( 11 ) ;
855   X$ = INKEY$ : IF X$ = "" THEN 855 ELSE IF X$ <>
      "Y" AND X$ <> "y" AND X$ <> "N" AND X$ <> "n" THEN855
860   IF X$ = "N" OR X$ = "n" THEN 900
865   M = M - 1 : NS = Ø : GOSUB 800
870   FOR I = 6882 TO 6910 : VPOKE I, 32 : NEXT : STRIG( Ø)
      ON : RETURN
```

Game Over

```
900   SCREEN 1 : IF M < Ø THEN PRINT "YOU▲OWE:$"; : PRINT
      USING "###.##"; - M : GOTO 990
920   PRINT "YOU▲HAVE▲WON:▲$"; : PRINT USING "###.##";
      M : GOTO 990
990   IF INKEY$ <> "" THEN END ELSE 990
```

Sprite Data

```
10000   DATA 128, 64, 96, 56, 28, 13, 7, 3, 1, 2, Ø, Ø,
        Ø, Ø, Ø, Ø, 1, 2, 6, 28, 56, 176, 224, 192, 128,
        64, Ø, Ø, Ø, Ø, Ø, Ø
10010   DATA Ø, Ø, Ø, Ø, Ø, Ø, 1, 7, 15, 25, 50, 32, 32, 32,
        Ø, Ø, Ø, Ø, Ø, Ø, Ø, Ø, 128, 224, 240, 152, 76,
        4, 4, 4, Ø, Ø, Ø
10020   DATA Ø, Ø, Ø, Ø, 3, 14, 24, 12, 6, 115, 255, 63,
        7, Ø, Ø, Ø, Ø, Ø, Ø, Ø, 252, 32, 64, 32, 248, 255,
        255, 254, 252, 248, Ø, Ø
10030   DATA Ø, Ø, Ø, Ø, 1, 115, 255, 63, 3, 7, 7, 3, Ø,
        Ø, Ø, Ø, Ø, Ø, Ø, Ø, 248, 254, 255, 255, 252, 248,
        248, 248, 252, 62, 15, Ø
```

```
10040   DATA 0, 0, 0, 0, 63, 4, 2, 4, 63, 255, 255, 127,
        63, 31, 0, 0, 0, 0, 0, 0, 192, 112, 24, 48, 224,
        206, 255, 252, 224, 0, 0, 0
10050   DATA 0, 0, 0, 0, 31, 127, 255, 255, 63, 31, 31,
        31, 63, 124, 240, 0, 0, 0, 0, 0, 128, 206, 255,
        252, 192, 224, 224, 192, 0, 0, 0, 0
10060   DATA 130, 200, 96, 49, 25, 45, 39, 75, 19, 5, 66,
        4, 65, 144, 4, 34, 1, 35, 6, 140, 216, 177, 224,
        201, 196, 144, 66, 32, 4, 18, 128, 1
```

Character Data

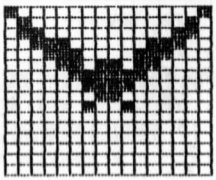
```
10100   DATA 170, 85, 170, 85, 170, 85, 170, 85
10102   DATA 96, 32, 32, 32, 32, 32, 32, 32
10104   DATA 32, 32, 48, 48, 48, 48, 112, 56
10106   DATA 48, 56, 56, 56, 60, 60, 60, 60
10108   DATA 0, 0, 24, 36, 82, 133, 64, 18
10110   DATA 170, 85, 170, 85, 0, 0, 0, 0
```

ChexSum Table

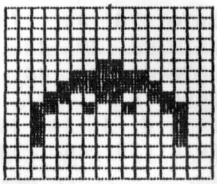
```
10    = 0
11    = 0
12    = 0
15    = 3436
20    = 3995
25    = 3996
30    = 3997
35    = 3994
40    = 3995
45    = 3996
50    = 3997
55    = 1899
60    = 4763
65    = 2046
70    = 3220
75    = 1569
80    = 651
100   = 6885
110   = 5843
115   = 7267
120   = 4733
125   = 3345
130   = 1684
140   = 158
200   = 918
210   = 1376
220   = 1457
290   = 593
300   = 5130
320   = 4736
330   = 691
335   = 2329
340   = 143
350   = 4466
360   = 3114
370   = 143
400   = 5834
410   = 4013
430   = 143
450   = 2989
465   = 1457
470   = 759
472   = 627
475   = 759
477   = 627
480   = 810
483   = 627
485   = 810
490   = 856
500   = 8016
505   = 2432
507   = 829
510   = 2664
512   = 845
515   = 2664
517   = 843
520   = 2572
522   = 855
525   = 2594
527   = 366
540   = 1382
600   = 3028
620   = 630
700   = 2971
710   = 3319
720   = 3322
730   = 3303
740   = 3306
750   = 3303
760   = 483
800   = 3960
805   = 5472
807   = 4368
810   = 1268
850   = 8241
855   = 6687
860   = 1916
865   = 1392
```

```
870    = 2891
900    = 4117
920    = 3279
990    = 1529
10000  = 5771
10010  = 5371
10020  = 6652
10030  = 6647
10040  = 6784
10050  = 7559
10060  = 8039
10100  = 1640
10102  = 1412
10104  = 1472
10106  = 1417
10108  = 1330
10110  = 1280

Total=271097
```

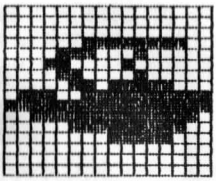
Sprite Shapes



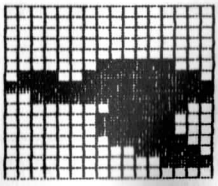
CRAZY BIRD: POSITION 1



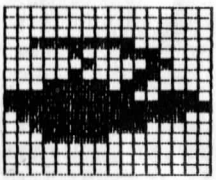
CRAZY BIRD: POSITION 2



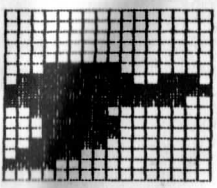
MIDDLE BIRD: POSITION 1



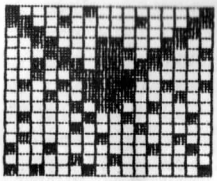
MIDDLE BIRD: POSITION 2



FRONT BIRD: POSITION 1

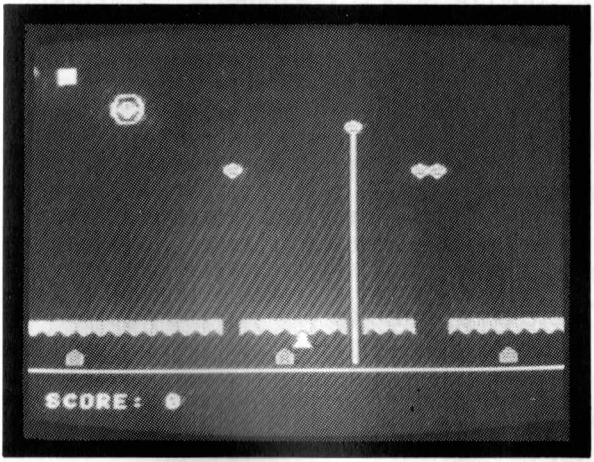


FRONT BIRD: POSITION 2



HIT BIRD

Laser Battle



CLASSIFICATION: Shoot-Out Game

This is one of the fastest games, with dazzling laser effects and super sounds that will thrill the arcade enthusiasts. Waves of guard ships try to obliterate your planet by striking one of the three energy stores or your laser.

The mother ship adds to the enemy's firepower and is worth more points. The small machine code routine is used to create the laser effect.

Use the cursor keys to move left or right and the <SPACE> bar to fire.

PROGRAMMING SUGGESTIONS

To add to the speed of the mother ship, change the poke to 59999 in line 255.

More guard ships could be added, but make sure you update the 'Position Guard Ships', 'Guard Fires' and 'Blow Guard Ship' routines.

PROGRAM

Variables

PH	Phase (Ø - 2)
GS(N, I)	Guard ship attributes: N = ship number I = 1 (column), I = 2 (row), I = 3 (stopped flag)
PG	Column of player's gun
IH	Ship hit flag
ISP	Starting point for firing laser
FG	Number of the guard ship that is firing
NM	Number of guns left

Listing

Initialise

```
 5   KEY OFF : SCREEN 1, 2 : COLOR 15, 1, 1
10   REM RUN MACHINE CODE
11   REM SUPPORT PROGRAM
12   REM SEE APPENDICES
15   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$
25   DEFUSR0 = 60000! : DEFUSR1 = 60118! : POKE 59996!,
     10 : POKE 59997!, 1 : POKE 59998!, 3 : POKE 59999!,
     4 : DEFUSR2 = 60220! : FOR I = 60220! TO 60248! :
     READ Q : POKE I, Q : NEXT
27   FOR I = 1088 TO 1344 STEP 64 : FOR J = 0 TO 7 :
     READ Q : VPOKE I + J, Q : NEXT : NEXT
30   PUT SPRITE 1, ( 100, 45 ) , 7 : PUT SPRITE 0, (
     127, 100 ) , 11 : NM = 3 : ON STRIG GOSUB 150
35   CLS : PRINT "▲▲▲▲▲▲▲▲LASER▲BATTLE": FOR I = 1 TO
     18 : PRINT : NEXT : PRINT "▲▲▲▲HIT▲ANY▲KEY▲TO▲BEGIN"
40   IF INKEY$ = "" THEN D = USR1( D ) : FOR I = 1 TO
     100 : NEXT : PLAY "L9M1000S14N33": GOTO 35
```

Set Up Round

```
50   CLS : PUT SPRITE 1, ( 100, 200 )
52   VPOKE 8209, 225 : VPOKE 8210, 49 : VPOKE 8211, 177 :
     VPOKE 8212, 97 : VPOKE 8213, 241
55   FOR I = 6720 TO 6751 : VPOKE I, 136 : NEXT : FOR
     I = 6816 TO 6847 : VPOKE I, 23 : NEXT : VPOKE 6787,
     160 : VPOKE 6799, 160 : VPOKE 6812, 160
57   PG = 15 : GOSUB 1000 : PUT SPRITE 0, ( 120, 150 )
     , 15 : STRIG( 0 ) ON
```

Control

```
60   POKE 59999!, 8 : D = USR0( D ) : PG = INT( VPEEK(
     6913 ) / 8 )
65   IF PH = 2 THEN GOSUB 400 ELSE IF PH = 0 THEN GOSUB
     220 : GOTO 70
67   GOSUB 250
70   IF RND( 1 ) < .01 AND PH = 1 THEN PH = 2
75   IF PH = 0 THEN 80
```

```
77   IF RND( 1 ) < .8 OR PH = 2 THEN 80
78   GOSUB 190
80   IF VPEEK( 6912 ) = 200 THEN 500
90   GOTO 60
```

Player Fires

```
150  D = RND( 1 ) : STRIG( 0 ) OFF : PLAY "l9m1000s14n33"
155  IH = 0 : I2 = 0 : FOR I1 = 1 TO 4 : IF GS ( I1,
     1 ) = PG THEN I2 = I1
156  NEXT : POKE 59991!, 152
157  IF I2 <> 0 THEN ISP = 6144 + 32 * GS ( I2, 2 ) +
     PG : IH = 2 : GOTO 165
160  IM = VPEEK( 6917 ) : IP = PG * 8
162  IF IP < IM + 2 OR IP - IM > 6 OR VPEEK( 6916 ) =
     200 THEN ISP = 6144 + PG : GOTO 165
163  IH = 1 : ISP = 6208 + PG
165  POKE 59992!, ISP MOD 256 : POKE 59993!, ISP ¥ 256 :
     I3 = 6720 + PG : POKE 59994!, I3 MOD 256 : POKE
     59995!, I3 ¥ 256 : D = USR2( D )
170  POKE 59991!, 32 : FOR I3 = 1 TO 20 : NEXT : D =
     USR2( D )
175  IF IH = 1 THEN GOSUB 300
177  IF IH = 2 THEN GOSUB 350
180  STRIG( 0 ) ON : RETURN
```

Guard Ship Fires

```
190  IF GS ( 1, 1 ) = 0 AND GS ( 2, 1 ) = 0 AND GS ( 3,1)
     = 0 AND GS ( 4, 1 ) = 0 THEN RETURN ELSE STRIG( 0 ) OFF
191  FG = FG + 1 : IF FG = 5 THEN FG = 1
192  IF GS ( FG, 1 ) = 0 THEN 191
193  FL = 0 : K1 = 0 : POKE 59991!, 152 : I = 6176 +
     32 * GS ( FG, 2 ) + GS ( FG, 1 ) : POKE 59992!,
     I MOD 256 : POKE 59993!, I ¥ 256
195  IF VPEEK( 6720 + GS ( FG, 1 ) ) = 136 THEN J = 6752
     + GS ( FG, 1 ) : GOTO 200
196  J = 6816 + GS ( FG, 1 ) : IF GS ( FG, 1 ) = PG THEN
     K1 = 1
197  K = GS ( FG, 1 ) : IF K = 3 OR K = 15 OR K = 28
     THEN FL = 1
200  POKE 59994!, J MOD 256 : POKE 59995!, J ¥ 256 :
     D = USR2( D ) : IF FL = 1 THEN FOR J = 1 TO 10 :
     COLOR 1, 15, 15 : FOR T = 1 TO 30 : NEXT : PLAY
     "l39m59000s8n2": COLOR 15, 1, 1 : FOR T = 1 TO 30 :
     NEXT : NEXT : GOTO 500
202  IF K1 = 1 THEN PUT SPRITE 0, ( 100, 200 ) : FOR
     K = 1 TO 7 : PLAY "l64m1000s14n20n21n20n21": NEXT
```

```
205   PLAY "l19m380s10n50": POKE 59991!, 32 : D = USR2( D )
210   STRIG( 0 ) ON : RETURN
```

Position Guard Ships

```
220   IF GS ( 1, 1 ) <> 0 THEN 225
221   I = INT( RND( 1 ) * 31 + 1 ) : IF I = GS ( 2, 1 )
      OR I = GS ( 3, 1 ) OR I = GS ( 4, 1 ) THEN 221 ELSE
      GS ( 1, 1 ) = I : GS ( 1, 2 ) = 0 : VPOKE 6144 +
      I, 144 : RETURN
225   IF GS ( 2, 1 ) <> 0 THEN 230
226   I = INT( RND( 1 ) * 31 + 1 ) : IF I = GS ( 1, 1 )
      OR I = GS ( 3, 1 ) OR I = GS ( 4, 1 ) THEN 226 ELSE
      GS ( 2, 1 ) = I : GS ( 2, 2 ) = 0 : VPOKE 6144 +
      I, 144 : RETURN
230   IF GS ( 3, 1 ) <> 0 THEN 235
231   I = INT( RND( 1 ) * 31 + 1 ) : IF I = GS ( 1, 1 )
      OR I = GS ( 2, 1 ) OR I = GS ( 4, 1 ) THEN 231 ELSE
      GS ( 3, 1 ) = I : GS ( 3, 2 ) = 0 : VPOKE 6144 +
      I, 144 : RETURN
235   IF GS ( 4, 1 ) <> 0 THEN 240
236   I = INT( RND( 1 ) * 31 + 1 ) : IF I = GS ( 1, 1 )
      OR I = GS ( 2, 1 ) OR I = GS ( 3, 1 ) THEN 236 ELSE
      GS ( 4, 1 ) = I : GS ( 4, 2 ) = 0 : VPOKE 6144 +
      I, 144 : RETURN
240   IF GS ( 1, 3 ) = 1 AND GS ( 2, 3 ) = 1 AND GS ( 3,3)
      = 1 AND GS ( 4, 3 ) = 1 THEN PH = 1 : RETURN
242   J = INT( RND( 1 ) * 4 + 1 ) : IF GS ( J, 3 ) = 1
      THEN 242
243   VPOKE GS ( J, 1 ) + 32 * GS ( J, 2 ) + 6144, 32 :
      GS ( J, 2 ) = GS ( J, 2 ) + 1
244   IF GS ( J, 1 ) <> 0 THEN VPOKE GS ( J, 1 ) + 32
      * GS ( J, 2 ) + 6144, 144
246   IF RND( 1 ) < .1 OR GS ( J, 2 ) > = 7 THEN GS (
      J, 3 ) = 1
248   RETURN
```

Move Mother Ship

```
250   IF VPEEK( 6916 ) <> 200 THEN 255
253   MC = 2 : PUT SPRITE 1, ( 0, 15 ) : RETURN
255   POKE 59997!, 1 : POKE 59998!, 3 : POKE 59999!, 6 :
      D = USR1( D ) : IF VPEEK( 6917 ) > 251 THEN PUT
      SPRITE 1, ( 200, 200 ) : RETURN
260   MC = INT( ( VPEEK( 6917 ) + 8 ) / 8 ) : IF RND( 1)
      < .85 THEN RETURN
265   IF MC <> GS ( 1, 1 ) AND MC <> GS ( 2, 1 ) AND MC
      <> GS ( 3, 1 ) AND MC <> GS ( 4, 1 ) THEN GOSUB 280
270   RETURN
```

Mother Ship Fires

```
280   PLAY "l24m160s8n67"
282   STRIG( 0 ) OFF : FL = 0 : K1 = 0 : POKE 59991!,
      152 : I = 6240 + MC : POKE 59992!, I MOD 256 : POKE
      59993!, I ¥ 256
284   IF VPEEK( MC + 6720 ) = 136 THEN J = 6752 + MC :
      GOTO 290
286   J = 6816 + MC : IF MC = 3 OR MC = 15 OR MC = 28
      THEN FL = 1
287   IF MC = PG THEN K1 = 1
290   POKE 59994!, J MOD 256 : POKE 59995!, J ¥ 256 :
      D = USR2( D )
291   IF FL = 1 THEN FOR J = 1 TO 10 : COLOR 1, 15, 15 :
      FOR T = 1 TO 30 : NEXT : PLAY "l39m59000s8n2": COLOR
      15, 1, 1 : FOR T = 1 TO 30 : NEXT : NEXT : GOTO 500
292   PLAY "l24m160s8n67"
293   IF K1 = 1 THEN PUT SPRITE 0, ( 100, 200 ) : FOR
      J = 1 TO 7 : PLAY "l64m1000s14n20n21n20n21": NEXT
294   POKE 59991!, 32 : D = USR2( D )
299   STRIG( 0 ) ON : RETURN
```

Blow Mother Ship

```
300   SC = SC + 50 : GOSUB 1000
310   PLAY "l64m60000s8n20n21n24n28n40n45n43n29n20n1513n10"
320   FOR I4 = 1 TO 60 : VPOKE 14368 + INT( RND( 1 ) *
      30 ) , INT( RND( 1 ) * 255 ) : NEXT : SPRITE$( 1 )= A$
330   PUT SPRITE 1, ( 200, 200 )
340   RETURN
```

Blow Guard Ship

```
350   VPOKE 6144 + GS ( I2, 1 ) + 32 * GS ( I2, 2 ) ,
      168 : SC = SC + 10 : GOSUB 1000 : FOR T = 1 TO 30 :
      NEXT
355   VPOKE 6144 + GS ( I2, 1 ) + 32 * GS ( I2, 2 ) ,
      32 : GS ( I2, 1 ) = 0
360   RETURN
```

Position Guard Ships (Last Phase)

```
400   IF GS ( 1, 1 ) <> 0 OR GS ( 2, 1 ) <> 0 OR GS ( 3,1)
      <> 0 OR GS ( 4, 1 ) <> 0 THEN STRIG( 0 ) OFF : GOTO
      405
402   FOR J = 1 TO 4 : GS ( J, 3 ) = 0 : NEXT : PH = 0 :
      RETURN
405   J = INT( RND( 1 ) * 4 + 1 ) : IF GS ( J, 1 ) = 0
      THEN 405 ELSE STRIG( 0 ) ON
407   VPOKE 6144 + GS ( J, 1 ) + 32 * GS ( J, 2 ) , 32 :
      IF GS ( J, 2 ) > 0 THEN GS ( J, 2 ) = GS ( J, 2 )- 1
410   VPOKE 6144 + GS ( J, 1 ) + 32 * GS ( J, 2 ) , 144 :
      IF GS ( J, 2 ) = 0 THEN VPOKE 6144 + GS ( J, 1 )
      , 32 : GS ( J, 1 ) = 0
415   RETURN
```

New Round

```
500   IF VPEEK( 6912 ) <> 200 THEN 560
502   STRIG( 0 ) OFF : FOR T = 1 TO 1500 : NEXT
510   CLS : PUT SPRITE 0, ( 100, 200 ) : PUT SPRITE 1,
      ( 100, 200 ) : PRINT "▲▲▲▲▲▲▲▲LASER▲BATTLE": PRINT :
      PRINT : PRINT
515   NM = NM - 1 : IF NM = 0 THEN 600
520   PRINT "▲▲▲▲▲Men▲Left:▲"; NM
530   FOR J = 1 TO 2500 : NEXT
535   FOR J = 1 TO 4 : GS ( J, 1 ) = 0 : NEXT : PH = 0
550   GOTO 50
560   STRIG( 0 ) OFF : FOR T = 1 TO 1500 : NEXT
565   CLS : PUT SPRITE 0, ( 100, 200 ) : PUT SPRITE 1,
      ( 100, 200 ) : PRINT "▲▲▲▲▲▲▲▲▲LASER▲BATTLE": PRINT :
      PRINT : PRINT
570   PRINT "YOUR▲POWER▲PLANT▲HAS▲BEEN▲": PRINT "DESTROYED":
      PRINT : PRINT : GOTO 515
```

Game Over

```
600   PRINT : PRINT : PRINT
610   PRINT "▲▲▲▲▲Your▲score▲was"; SC
620   IF INKEY$ = "" THEN END ELSE 620
```

Update Score

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT "SCORE:";
      SC ; CHR$( 11 ) ;
1010  RETURN
```

Sprite Data

```
10000   DATA 24, 60, 24, 60, 24, 52, 122, 255
10010   DATA 15, 16, 32, 32, 99, 198, 203, 254, 255, 207,
        199, 99, 33, 48, 16, 15, 240, 8, 4, 4, 198, 99,
        211, 127, 255, 243, 227, 198, 132, 12, 8, 240
```

Machine-Code Data

```
10100   DATA 42, 88, 234, 58, 87, 234, 79, 205, 44, 235,
        17, 32, 0, 25, 237, 91, 90, 234, 124, 186, 194,
        67, 235, 125, 187, 194, 67, 235, 201
```

Character Data

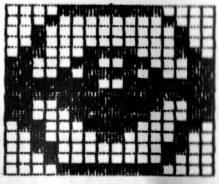
```
10200   DATA 90, 165, 90, 165, 90, 165, 66, 129
10202   DATA 24, 126, 213, 171, 255, 66, 60, 24
10204   DATA 24, 24, 24, 24, 24, 24, 24, 24
10206   DATA 24, 36, 90, 165, 219, 165, 219, 255
10208   DATA 2, 144, 4, 17, 64, 4, 161, 8
```

ChexSum Table

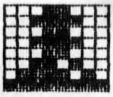
```
5     = 1217
10    = 0
11    = 0
12    = 0
15    = 4011
20    = 3483
25    = 10365
27    = 3640
30    = 3648
35    = 5847
40    = 4677
50    = 1145
52    = 2806
55    = 4952
57    = 2901
60    = 3460
65    = 3195
67    = 410
70    = 2408
75    = 890
77    = 2213
78    = 350
80    = 1663
90    = 449
150   = 3068
155   = 3679
156   = 935
157   = 4301
160   = 1883
162   = 5524
163   = 1523
165   = 7655
170   = 2535
175   = 986
177   = 1036
180   = 883
190   = 5101
191   = 2077
192   = 1302
193   = 7013
195   = 4348
196   = 3156
197   = 3325
200   = 13212
202   = 4889
205   = 2819
210   = 883
220   = 1475
221   = 9526
225   = 1474
226   = 9512
230   = 1473
231   = 9520
235   = 1488
236   = 9526
240   = 4633
242   = 3360
243   = 3549
244   = 3625
246   = 3305
248   = 143
250   = 1914
253   = 1329
```

```
255   = 6489
260   = 3902
265   = 5858
270   = 143
280   = 1195
282   = 6327
284   = 3313
286   = 3751
287   = 1301
290   = 3018
291   = 7810
292   = 1195
293   = 4888
294   = 1399
299   = 883
300   = 1335
310   = 3793
320   = 5613
330   = 1012
340   = 143
350   = 5156
355   = 2966
360   = 143
400   = 6530
402   = 2373
405   = 4230
407   = 4875
410   = 5668
415   = 143
500   = 1724
502   = 2018
510   = 4972
515   = 1787
520   = 1528
530   = 1123
535   = 2158
550   = 443
560   = 2018
565   = 5040
570   = 4811
600   = 561
610   = 2142
620   = 1416
1000  = 2916
1010  = 143
10000 = 1529
10010 = 9019
10100 = 8191
10200 = 1634
10202 = 1649
10204 = 1404
10206 = 1689
10208 = 1275
Total = 368357
```

Sprite Shapes

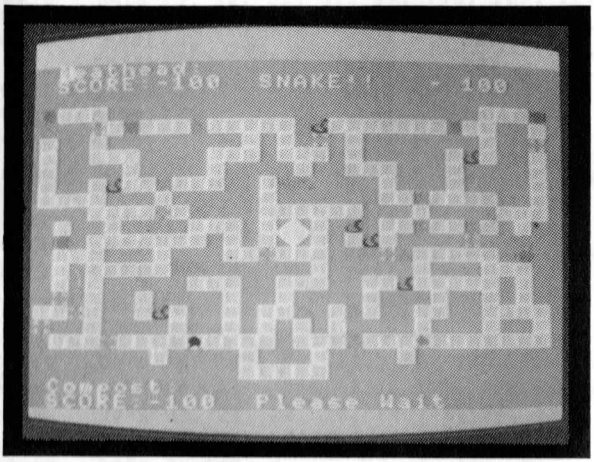


LASER



MOTHER SHIP

Treasure Trek



CLASSIFICATION: Parlor Game COLOUR ILLUSTRATION OBC

Race to reach the diamonds before your opponent, gathering rubies, emeralds and opals along the way. It is often better to head for the rubies however, as tactics are the best way to win this game!

Scoring:

Diamonds	1ØØØ
Rubies	75Ø
Emeralds	5ØØ
Opals	25Ø
Snake	−1ØØ

PROGRAMMING SUGGESTIONS

There are two main tactics in this game: going straight for the diamonds or heading for a corner to pounce on some rubies. By rearranging the screen and adding some different gemstones, you could increase the options for tactical playing. This would not be hard to do, as the arrangement of the screen only affects the set-up screen routine in lines 100-195

PROGRAM

Variables

PP(1)	Player 1 position
S(1)	Player 1 score
P$(1)	Player names
M$(1)	Messages to players
CP	Current player
CU(1)	Characters under player

Listing

Initialise

```
  5   COLOR 15, 4, 7 : SCREEN 1, 2 : KEY OFF
  7   VPOKE 8210, 180 : VPOKE 8211, 36 : VPOKE 8212, 132 :
      VPOKE 8213, 20
 10   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
      NEXT : SPRITE$( 0 ) = A$ : SPRITE$( 1 ) = A$
 15   FOR I = 1072 TO 1079 : READ Q : VPOKE I, Q : NEXT :
      FOR I = 1152 TO 1344 STEP 64 : FOR J = 0 TO 7 :
      READ Q : VPOKE I + J, Q : NEXT : NEXT : FOR I =
      1112 TO 1143 : READ Q : VPOKE I, Q : NEXT
 20   PRINT "▲▲▲▲▲▲▲▲TREASURE▲TREK": PRINT : PRINT : PRINT :
      PRINT
 25   PRINT "▲▲Gem▲▲▲▲▲▲▲▲Value▲(Millions)"; : PRINT "▲▲---▲▲
      ▲▲▲▲▲▲▲----------------";
 30   PRINT : PRINT : PRINT "opals" TAB( 10 ) CHR$( 144 )
      TAB( 19 ) "250"
 35   PRINT : PRINT "emeralds" TAB( 10 ) CHR$( 152 ) TAB(
      19 ) "500"
 40   PRINT : PRINT "rubies" TAB( 10 ) CHR$( 160 ) TAB( 19)
      "750"
 45   PRINT : PRINT "diamond" TAB( 10 ) CHR$( 139 ) CHR$(
      140 ) TAB( 19 ) "1000": PRINT TAB( 10 ) CHR$( 141 )
      CHR$( 142 )
 47   PRINT : PRINT "snake" TAB( 10 ) CHR$( 168 ) TAB( 18)
      "-▲100"
 50   PRINT : PRINT : P$ ( 1 ) = "Meathead": INPUT "Player▲1";
      P$ ( 1 ) :  PRINT : P$ ( 2 ) = "Compost": INPUT
      "Player▲2"; P$ ( 2 )
 55   IF LEN( P$ ( 2 ) ) > 9 OR LEN( P$ ( 1 ) ) > 9 THEN
      PRINT "TOO▲LONG": GOTO 50 ELSE PRINT : PRINT "OK:▲"
      P$ ( 1 ) "▲and▲"P$ ( 2 )
 57   FOR I = 1 TO 20 : PRINT : NEXT : PRINT "▲▲▲▲▲▲▲▲▲Hit▲a
      Key"
 60   IF INKEY$ = "" THEN D = RND( 1 ) : GOTO 60
```

Set Up Screen

```
100   CLS : VPOKE 6511, 139 : VPOKE 6512, 140 : VPOKE
      6543, 141 : VPOKE 6544, 142
105   FOR I = 6241 TO 6244 : VPOKE I; 134 : NEXT : FOR
      I = 6267 TO 6270 : VPOKE I, 134 : NEXT : VPOKE 6273,
      134 : FOR I = 6276 TO 6287 : VPOKE I, 134 : NEXT :
      FOR I = 6289 TO 6299 : VPOKE I, 134 : NEXT : VPOKE
      6302, 134
```

```
110   VPOKE 6305, 134 : VPOKE 6308, 134 : VPOKE 6317,
      134 : VPOKE 6319, 134 : VPOKE 6321, 134 : VPOKE
      6323, 134 : VPOKE 6323, 134 : VPOKE 6331, 134 :
      VPOKE 6334, 134
115   VPOKE 6337, 134 : VPOKE 6340, 134 : VPOKE 6341,
      134 : VPOKE 6342, 134 : VPOKE 6347, 134 : VPOKE
      6348, 134 : VPOKE 6349, 134 : VPOKE 6351, 134 :
      VPOKE 6352, 134 : VPOKE 6353, 134 : VPOKE 6355,
      134 : VPOKE 6361, 134 : VPOKE 6362, 134 : VPOKE
      6363, 134 : VPOKE 6366, 134
120   VPOKE 6369, 134 : VPOKE 6374, 134 : VPOKE 6379,
      134 : VPOKE 6384, 134 : VPOKE 6387, 134 : VPOKE
      6388, 134 : VPOKE 6389, 134 : VPOKE 6393, 134 :
      VPOKE 6398, 134
125   VPOKE 6401, 134 : FOR I = 6404 TO 6411 : VPOKE I,
      134 : NEXT : VPOKE 6414, 134 : VPOKE 6415, 134 :
      VPOKE 6416, 134 : VPOKE 6421, 134 : VPOKE 6425,
      134 : VPOKE 6430, 134
130   FOR I = 6433 TO 6436 : VPOKE I, 134 : NEXT : VPOKE
      6440, 134 : VPOKE 6446, 134 : FOR I = 6453 TO 6459 :
      VPOKE I, 134 : NEXT : VPOKE 6462, 134
135   VPOKE 6466, 134 : FOR I = 6468 TO 6474 : VPOKE I,
      134 : NEXT : FOR I = 6478 TO 6483 : VPOKE I, 134 :
      NEXT : VPOKE 6487, 134 : FOR I = 6491 TO 6494 :
      VPOKE I, 134 : NEXT
140   VPOKE 6498, 134 : VPOKE 6501, 134 : VPOKE 6506,
      134 : VPOKE 6507, 134 : VPOKE 6508, 134 : VPOKE
      6510, 134 : VPOKE 6513, 134 : VPOKE 6515, 134 :
      FOR I = 6517 TO 6523 : VPOKE I, 134 : NEXT : VPOKE
      6525, 134
145   VPOKE 6530, 134 : FOR I = 6532 TO 6538 : VPOKE I,
      134 : NEXT : VPOKE 6540, 134 : VPOKE 6542, 134 :
      VPOKE 6545, 134 : VPOKE 6547, 134 : VPOKE 6548,
      134 : VPOKE 6549, 134 : VPOKE 6554, 134 : VPOKE
      6557, 134
150   VPOKE 6562, 134 : VPOKE 6563, 134 : VPOKE 6564,
      134 : VPOKE 6568, 134 : FOR I = 6572 TO 6577 : VPOKE
      I, 134 : NEXT : FOR I = 6581 TO 6587 : VPOKE I,
      134 : NEXT : VPOKE 6589, 134
155   VPOKE 6594, 134 : FOR I = 6596 TO 6602 : VPOKE I,
      134 : NEXT : VPOKE 6609, 134 : VPOKE 6615, 134 :
      VPOKE 6619, 134 : VPOKE 6620, 134 : VPOKE 6621, 134
160   VPOKE 6626, 134 : VPOKE 6628, 134 : VPOKE 6631,
      134 : VPOKE 6634, 134 : VPOKE 6635, 134 : VPOKE
      6636, 134 : VPOKE 6639, 134 : VPOKE 6640, 134 :
      VPOKE 6641, 134 : FOR I = 6644 TO 6651 : VPOKE I,
      134 : NEXT : VPOKE 6653, 134
165   VPOKE 6658, 134 : VPOKE 6660, 134 : VPOKE 6663,
      134 : VPOKE 6668, 134 : VPOKE 6671, 134 : VPOKE
```

```
      6676, 134 : VPOKE 6680, 134 : VPOKE 6683, 134 :
      VPOKE 6685, 134
170   FOR I = 6689 TO 6692 : VPOKE I, 134 : NEXT : VPOKE
      6695, 134 : VPOKE 6696, 134 : VPOKE 6697, 134 :
      VPOKE 6700, 134 : VPOKE 6702, 134 : VPOKE 6703,
      134 : VPOKE 6704, 134 : VPOKE 6706, 134 : VPOKE
      6707, 134 : VPOKE 6708, 134 : VPOKE 6711, 134 :
      VPOKE 6712, 134 : FOR I = 6715 TO 6718 : VPOKE I,
      134 : NEXT
175   VPOKE 6721, 134 : VPOKE 6724, 134 : VPOKE 6729,
      134 : VPOKE 6732, 134 : VPOKE 6734, 134 : VPOKE
      6736, 134 : VPOKE 6738, 134 : VPOKE 6743, 134 :
      VPOKE 6747, 134 : VPOKE 6750, 134
180   FOR I = 6753 TO 6766 : VPOKE I, 134 : NEXT : FOR
      I = 6768 TO 6782 : VPOKE I, 134 : NEXT
185   VPOKE 6792, 134 : VPOKE 6797, 134 : VPOKE 6801,
      134 : VPOKE 6806, 134 : FOR I = 6829 TO 6833 : VPOKE
      I, 134 : NEXT
190   VPOKE 6241, 160 : VPOKE 6270, 160 : FOR I = 6242
      TO 6782 : IF VPEEK( I ) = 134 THEN K = RND( 1 ) :
      IF K < .05 THEN VPOKE I, 168 ELSE IF K < .1 THEN
      VPOKE I, 152 ELSE IF K < .15 THEN VPOKE I, 144
192   NEXT : VPOKE 6530, 160 : VPOKE 6589, 160 : VPOKE
      6278, 160 : VPOKE 6297, 160
195   M$ ( 1 ) = "Your▲Turn!▲▲▲▲▲": M$ ( 2 ) = "Please▲Wait▲▲
      ▲▲" : GOSUB 1000 : CP = 1 : Q = 1 : PUT SPRITE 0,
      ( 64, 160 ) , 1 : PUT SPRITE 1, ( 176, 160 ) , 12 :
      CU ( 1 ) = 134 : CU ( 2 ) = 134 : PP ( 1 ) = 6792 :
      PP ( 2 ) = 6806
```

Editor

```
200   X$ = INKEY$ : C = C + 1
210   IF C = 6 THEN Q = 1 - Q : C = 0
220   IF Q <> 0 THEN 250
230   Q = 1 : K = VPEEK( PP ( CP ) ) : IF K = 32 THEN
      VPOKE PP ( CP ) , CU ( CP ) ELSE VPOKE PP ( CP ) , 32
250   IF X$ = "" THEN 200
260   K = ASC( X$ ) - 27 : IF K < 1 OR K > 4 THEN 200 ELSE
      ON K GOTO 300, 320, 340, 360
```

Recognise Allowed Moves

```
300   IF VPEEK( PP ( CP ) + 1 ) = 32 THEN 200
305   VPOKE PP ( CP ) , CU ( CP ) : CU ( CP ) = VPEEK(
      PP ( CP ) + 1 ) : PP ( CP ) = PP ( CP ) + 1 : VPOKE
      6909 + 4 * CP, VPEEK( 6909 + 4 * CP ) + 8
```

```
310   GOTO 400
320   IF VPEEK( PP ( CP ) - 1 ) = 32 THEN 200
325   VPOKE PP ( CP ) , CU ( CP ) : CU ( CP ) = VPEEK(
      PP ( CP ) - 1 ) : PP ( CP ) = PP ( CP ) - 1 : VPOKE
      6909 + 4 * CP, VPEEK( 6909 + 4 * CP ) - 8
330   GOTO 400
340   IF VPEEK( PP ( CP ) - 32 ) = 32 THEN 200
345   VPOKE PP ( CP ) , CU ( CP ) : CU ( CP ) = VPEEK(
      PP ( CP ) - 32 ) : PP ( CP ) = PP ( CP ) - 32 :
      VPOKE 6908 + 4 * CP, VPEEK( 6908 + CP * 4 ) - 8
350   GOTO 400
360   IF VPEEK( PP ( CP ) + 32 ) = 32 THEN 200
365   VPOKE PP ( CP ) , CU ( CP ) : CU ( CP ) = VPEEK(
      PP ( CP ) + 32 ) : PP ( CP ) = PP ( CP ) + 32 :
      VPOKE 6908 + 4 * CP, VPEEK( 6908 + 4 * CP ) + 8
```

Check Move

```
400   K = CU ( CP ) : IF K = 134 THEN GOSUB 750 ELSE IF
      K = 144 THEN GOSUB 600 ELSE IF K = 152 THEN GOSUB
      650 ELSE IF K = 160 THEN GOSUB 550 ELSE IF K = 168
      THEN GOSUB 700 ELSE GOSUB 900
401   K = CU ( CP ) : ON K GOSUB 750, 600, 650, 550, 700,
      900, 900, 900, 900
410   IF CP = 1 THEN CP = 2 ELSE CP = 1
420   M$ ( CP ) = "Your▲Turn!▲▲▲▲▲": IF CP = 1 THEN M$
      ( 2 ) = "Please▲wait▲▲▲▲" ELSE M$ ( 1 ) = "Please▲Wait▲
      ▲▲▲"
425   GOSUB 1000
490   GOTO 200
```

Rubies

```
550   M$ ( CP ) = "RUBIES!!!▲+▲750": S ( CP ) = S ( CP )
      + 750 : GOSUB 1000 : CU ( CP ) = 134 : VPOKE PP
      ( CP ) , 134
580   PLAY "14s8m30000n4518n4014n45", "14s8m20000n3618n3114n3
      6": FOR T = 1 TO 1000 : NEXT
590   RETURN
```

Opals

```
600   M$ ( CP ) = "OPALS!▲▲▲+▲250▲": S ( CP ) = S ( CP )
      + 250 : GOSUB 1000 : CU ( CP ) = 134 : VPOKE PP
      ( CP ) , 134
630   PLAY "14s14m3000n45": FOR T = 1 TO 1000 : NEXT
640   RETURN
```

Emeralds

```
650   M$ ( CP ) = "EMERALDS!▲+▲500": S ( CP ) = S ( CP )
      + 500 : GOSUB 1000 : CU ( CP ) = 134 : VPOKE PP
      ( CP ) , 134
680   PLAY "14s14m3000n40l2n45": FOR T = 1 TO 1000 : NEXT
690   RETURN
```

Snake

```
700   M$ ( CP ) = "SNAKE!!▲▲▲-▲100": S ( CP ) = S ( CP )
      - 100 : GOSUB 1000 : CU ( CP ) = 134 : VPOKE PP
      ( CP ) , 134
730   PLAY "164s8m59000n45n46n47n48n49n50n51n51n50n49n48n47n4
      6n45n44n45n46n47n48n49n50n51n50n49n48n47n46n45n4413n43":
      FOR T = 1 TO 2000 : NEXT
740   RETURN.
```

Player on Normal Square

```
750   IF RND( 1 ) < .7 THEN RETURN
755   IF RND( 1 ) < .3 THEN M$ ( CP ) = "SILVER!▲+▲50▲▲▲":
      S ( CP ) = S ( CP ) + 50 : GOSUB 1000 : PLAY "16s1m777n
      34n50n23", "12s5m2000n40": FOR T = 1 TO 1000 : NEXT :
      RETURN
760   IF RND( 1 ) < .5 THEN M$ ( CP ) = "AMETHYST!▲+▲20▲":
      S ( CP ) = S ( CP ) + 20 : GOSUB 1000 : PLAY "14s10m500
      n35n30", "12s1m1000n20": FOR T = 1 TO 1000 : NEXT :
      RETURN
765   M$ ( CP ) = "FOOL'S▲GOLD▲-50": S ( CP ) = S ( CP )
      - 50 : GOSUB 1000 : PLAY "11s8m20000n2": FOR T =
      1 TO 1000 : NEXT : RETURN
```

Game Over

```
900   SCREEN 1 : S ( CP ) = S ( CP ) + 1000
910   PLAY "18s2m30000n45n46l4n45l2n46l1n37", "s2m3000018n37r
      8l4n37r2l1n33"
920   IF S ( 1 ) > S ( 2 ) THEN PRINT P$ ( 1 ) "▲has▲won!!"
      ELSE IF S ( 2 ) > S ( 1 ) THEN PRINT P$ ( 2 ) "▲has▲won
      !!" ELSE PRINT "▲▲▲▲▲▲▲▲▲IT'S▲A▲TIE"
930   PRINT : PRINT : PRINT : PRINT "The▲scores▲are:":
      PRINT : PRINT P$ ( 1 ) , S ( 1 ) : PRINT : PRINT
      P$ ( 2 ) , S ( 2 )
980   PRINT : PRINT : PRINT
990   END
```

Update Message/Score

```
1000    PRINT P$ ( 1 ) ":": PRINT "SCORE:"; S ( 1 ) ; TAB(
        12 ) M$ ( 1 ) : FOR I = 1 TO 20 : PRINT : NEXT :
        PRINT P$ ( 2 ) ":": PRINT "SCORE:"; S ( 2 ) ; TAB(
        12 ) M$ ( 2 ) ; CHR$( 11 ) ;
1010    RETURN
```

Sprite Data

```
10000   DATA 24, 60, 126, 126, 60, 36, 66, 129
```

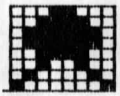
Character Data

```
10100   DATA 255, 213, 171, 213, 171, 213, 171, 255
10102   DATA 66, 231, 66, 0, 0, 66, 231, 66
10104   DATA 32, 112, 248, 116, 46, 31, 14, 4
10106   DATA 60, 126, 255, 255, 255, 255, 126, 60
10108   DATA 2, 7, 12, 76, 140, 134, 195, 126
10110   DATA 1, 3, 7, 15, 31, 63, 127, 255
10112   DATA 128, 192, 224, 240, 248, 252, 254, 255
10114   DATA 255, 127, 63, 31, 15, 7, 3, 1
10116   DATA 255, 254, 252, 248, 240, 224, 192, 128
```

ChexSum Tables

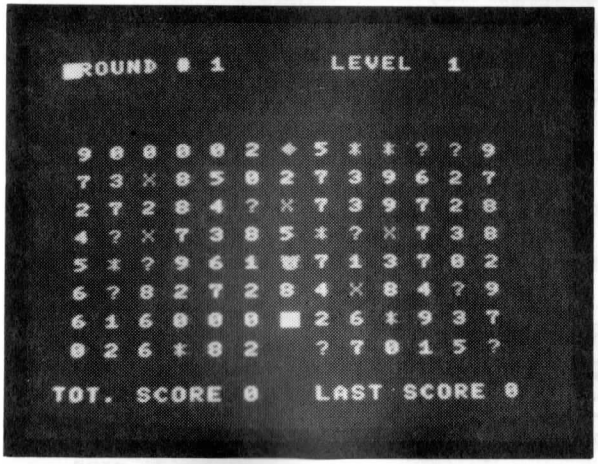
```
5     = 1236
7     = 1940
10    = 4362
15    = 8894
20    = 2536
25    = 4738
30    = 2878
35    = 2982
40    = 2787
45    = 5790
47    = 2777
50    = 7165
55    = 7015
57    = 2816
60    = 2297
100   = 1772
105   = 5850
110   = 6446
115   = 15401
120   = 8095
125   = 6994
130   = 5901
135   = 7883
140   = 9606
145   = 10499
150   = 8947
155   = 7551
160   = 12260
165   = 6289
170   = 15921
175   = 7800
180   = 3771
185   = 5077
190   = 11107
192   = 3109
195   = 14919
200   = 1325
210   = 1793
220   = 1220
230   = 4863
250   = 1032
260   = 4656
300   = 2084
305   = 8596
310   = 537
320   = 2084
325   = 8598
330   = 537
340   = 21069
345   = 8611
350   = 537
360   = 2069
365   = 8603
400   = 9663
401   = 3584
410   = 1860
420   = 7514
425   = 397
490   = 593
550   = 6421
580   = 5622
590   = 143
600   = 6288
630   = 2562
640   = 143
650   = 6465
680   = 3007
690   = 143
700   = 6124
730   = 12076
740   = 143
750   = 1506
755   = 11737
760   = 11539
765   = 7148
900   = 1748
910   = 5112
920   = 8013
930   = 4599
980   = 561
990   = 129
1000  = 8401
1010  = 143
10000 = 1570
10100 = 1885
10102 = 1390
10104 = 1535
10106 = 1775
10108 = 1514
10110 = 1326
10112 = 1891
10114 = 1343
10116 = 1891

Total= 444129
```

Sprite Shapes



PLAYER

Junior Minotaur Mastermind



CLASSIFICATION: Educational Game

Here is a great educational game that is a simplified version of 'Minotuar Mastermind'. The secret number is between 1Ø and 99 and the clues have been simplified to cater for younger players

Your score is the sum of the digits that you step on — so the **lower** scores are better. When you reach the exit at the top of the maze. you get a chance to guess the secret number. The non-numeric symbols have different effects when stepped on.

X	Supplies a clue
?	Has random value
*	Randomly alters your score

You are represented by a flashing point; the controls are:

<P>	–	Up	<:>	–	Down/right
<Ø>	–	Up/left	<@>	–	Right
<;>	–	Down	<L>	–	Down/left
<O>	–	Left	<T>	–	Trapped
<->	–	Up/right			

PROGRAMMING SUGGESTIONS

You may wish to modify the difficulty of the game rather than just playing at different levels of difficulty. One possibility is to simplify the screen grid (lines 1ØØ-16Ø) so that there are fewer numbers and symbols.

As with the senior version of this game, you could add more clues (lines 65Ø-688) or change existing ones.

PROGRAM

Variables

LD	Level of difficulty
R	Round number
SN	Secret number
D1. D2	Digits of secret number
PP	Player's position
M	Minotaur's position
LS. TS	Last score: total score
RN. RS	Random number: random number seed
CU. CP	Character under minotaur: player
PR. PC	Player's row: player's column
NC	Number of clues
T(I)	Clue I used?
G1. G2	Digits of player's guess
MC. MR	Minotaur's column: minotaur's row

Listing

Initialise

```
 5   TM = 1 : INTERVAL ON : ON  INTERVAL = 10 GOSUB 7
 6   GOTO 10
 7   TM = TM * 2 : RETURN
10   KEY OFF : SCREEN 1 : CLS : COLOR 15, 1, 1 : PRINT
     "▲JUNIOR▲MINOTAUR▲MASTER-MIND"
15   PRINT : PRINT : PRINT : PRINT "▲LEVEL▲OF▲DIFFICULTY?(1-
     9)";
20   LD = VAL( INKEY$ ) : IF LD < 1 OR LD > 9 THEN 20
     ELSE PRINT LD
22   FOR I = 1088 TO 1344 STEP 64 : FOR J = 0 TO 7 :
     READ Q : VPOKE I + J, Q : NEXT : NEXT
23   VPOKE 8209, 49 : VPOKE 8210, 97 : VPOKE 8211, 177 :
     VPOKE 8212, 145 : VPOKE 8213, 129
25   FOR I = 1 TO 8 : PRINT : NEXT : PRINT "▲▲▲▲HIT▲ANY▲KEY
     TO▲START"
30   IF INKEY$ = "" THEN 30
35   INTERVAL  OFF :  IF  TM > 1000 THEN TM = TM / 3 :
     GOTO 35
40   TS = 0 : NC = 0 : FOR I = 1 TO 6 : T ( I ) = 0 :
     NEXT : R = 1 : SN = INT( 89 * TM / 1000 + 10 ) :
     RS = SN * 100 : D1 = INT( SN / 10 ) : D2 = INT(
     SN - D1 * 10 )
```

New Round

```
100  CLS : PP = 6799 : M = 6607
110  FOR I = 6336 TO 6784 STEP 64 : FOR J = 3 TO 27 STEP
     2 : GOSUB 900 : K1 = INT( RN * 13 ) : IF K1 < 10
     THEN VPOKE I + J, K1 + 48 : GOTO 130
115  IF K1 = 10 THEN VPOKE I + J, 144 : GOTO 130
120  IF K1 = 11 THEN VPOKE I + J, 152 : GOTO 130
125  VPOKE I + J, 160
130  NEXT : NEXT
140  VPOKE PP, 48 : LS = 0 : CU = VPEEK( M ) : VPOKE
     M, 136 : VPOKE 6351, 168 : CP = 48
150  PRINT "▲ROUND▲#"; R ; "▲▲▲▲▲LEVEL▲"; LD ; CHR$( 11 ) ;
160  GOSUB 1000
```

Editor

```
200  C = 0 : Q = 1
205  X$ = INKEY$ : C = C + 1
```

```
210   IF C = 7 THEN Q = 1 - Q : C = 0
215   IF Q <> 0 THEN 230
220   IF VPEEK( PP ) = 255 THEN VPOKE PP, CP : Q = 1 :
      GOTO 230
225   VPOKE PP, 255 : Q = 1
230   IF X$ = "" THEN 205
240   GOSUB 950
242   IF VPEEK( PP ) = 255 THEN VPOKE PP, CP
245   IF X$ = "p" OR X$ = "P" THEN 300
250   IF X$ = "o" OR X$ = "O" THEN 310
255   IF X$ = "@" OR X$ = "`" THEN 320
260   IF X$ = ";" OR X$ = "+" THEN 330
265   IF X$ = "-" OR X$ = "=" THEN 340
270   IF X$ = "0" THEN 350
275   IF X$ = "1" OR X$ = "L" THEN 360
280   IF X$ = ":" OR X$ = "*" THEN 370
285   IF X$ = "t" OR X$ = "T" THEN 380
290   GOTO 205
```

Recognise Allowed Moves

```
300   IF VPEEK( PP - 64 ) = 32 THEN 205
302   LS = VPEEK( PP ) - 48
304   VPOKE PP, 32 : PP = PP - 64
305   GOTO 400
310   IF VPEEK( PP - 2 ) = 32 THEN 205
312   LS = VPEEK( PP ) - 48
314   VPOKE PP, 32 : PP = PP - 2
315   GOTO 400
320   IF VPEEK( PP + 2 ) = 32 THEN 205
322   LS = VPEEK( PP ) - 48
324   VPOKE PP, 32 : PP = PP + 2
325   GOTO 400
330   IF VPEEK( PP + 64 ) = 32 THEN 205
332   LS = VPEEK( PP ) - 48
334   VPOKE PP, 32 : PP = PP + 64
335   GOTO 400
340   IF VPEEK( PP - 62 ) = 32 THEN 205
342   LS = VPEEK( PP ) - 48
344   VPOKE PP, 32 : PP = PP - 62
345   GOTO 400
350   IF VPEEK( PP - 66 ) = 32 THEN 205
352   LS = VPEEK( PP ) - 48
354   VPOKE PP, 32 : PP = PP - 66
355   GOTO 400
360   IF VPEEK( PP + 62 ) = 32 THEN 205
362   LS = VPEEK( PP ) - 48
364   VPOKE PP, 32 : PP = PP + 62
365   GOTO 400
```

```
370   IF VPEEK( PP + 66 ) = 32 THEN 205
372   LS = VPEEK( PP ) - 48
374   VPOKE PP, 32 : PP = PP + 66
375   GOTO 400
380   PR = INT( PP / 32 ) : PC = PP - PR * 32 : GOSUB
      500 : IF PP <> M THEN 380
385   GOTO 800
```

Check Move

```
400   CP = VPEEK( PP ) : PR = INT( PP / 32 ) : PC = PP
      - PR * 32
402   IF( LS > 9 ) OR( LS < 0 ) THEN LS = 0
405   K1 = VPEEK( PP ) : IF K1 = 152 THEN GOSUB 600
410   IF K1 = 144 THEN GOSUB 650
415   IF K1 = 160 THEN GOSUB 750
420   IF K1 = 168 THEN 2000
422   K1 = VPEEK( PP ) - 48 : IF K1 < 0 OR K1 > 9 THEN K1 = 0
425   IF LS + K1 > = 10 - LD THEN GOSUB 500
430   TS = TS + K1 : GOSUB 1000
435   IF PP = M THEN 800
440   GOTO 200
```

Move Minotaur

```
500   VPOKE 8198, 31 : MR = INT( M / 32 ) : MC = M - MR * 32
502   IF MC <> PC THEN 508
504   IF MR > PRT HENMR = MR - 2 : GOTO 540
506   MR = MR + 2 : GOTO 540
508   IF MR <> PRT HEN515
510   IF MC > PC THEN MC = MC - 2 : GOTO 540
512   MC = MC + 2 : GOTO 540
515   IF M > PP THEN 522
517   IF MC < PC THEN MC = MC + 2 : GOTO 540
520   MC = MC - 2 : GOTO 540
522   MR = MR - 2
540   K = 32 * MR + MC : VPOKE 8198, 241 : IF VPEEK( K )
      = 131 THEN RETURN
542   VPOKE M, CU
544   M = K : CU = VPEEK( M )
546   VPOKE M, 136
550   PLAY "n2"
570   RETURN
```

Player On (?)

```
600   GOSUB 900 : I = INT( RN * 10 ) : VPOKE PP, I + 48
610   CP = I + 48 : RETURN
```

Player On (X)

```
650   VPOKE PP, 48
652   IF NC > = 6 THEN BEEP : GOTO 670
654   GOSUB 900 : K2 = INT( RN * 6 + 1 ) : IF T ( K2 )
      = 1 THEN 654
656   T ( K2 ) = 1 : NC = NC + 1 : BEEP
658   ON  K2  GOTO  660, 665, 670, 675, 680, 685
660   PRINT : PRINT : PRINT "▲First▲Digit▲plus▲Second":
      PRINT "▲Digit▲Equals▲";
662   PRINT D1 + D2 : PRINT CHR$( 11 ) ; : RETURN
665   PRINT : PRINT : PRINT "▲Secret▲Number▲is▲";
666   IF SN > 50 THEN PRINT "Greater▲▲▲▲than▲50": GOTO 668
667   PRINT "Less▲than▲▲▲or▲equal▲to▲50"
668   PRINT CHR$( 11 ) ; : RETURN
670   GOSUB 900 : K = INT( RN * 15 ) + SN : J = K - 15 :
      IF J < 0 THEN J = 0
672   IF K > 99 THEN K = 99
673   PRINT : PRINT : PRINT "▲Number▲is▲Between"; J :
      PRINT "▲and"; K
674   PRINT CHR$( 11 ) ; : RETURN
675   PRINT : PRINT : PRINT "▲First▲Digit▲is▲";
676   IF D1 < 5 THEN PRINT "Less▲than▲5": GOTO 678
677   PRINT "Greater▲than▲or▲Equal▲to▲5"
678   PRINT CHR$( 11 ) ; : RETURN
680   PRINT : PRINT : PRINT "▲Second▲Digit▲is▲";
681   IF D2 < 5 THEN PRINT "Less▲than▲5": GOTO 683
682   PRINT "Greater▲than▲or▲Equal▲to▲5"
683   PRINT CHR$( 11 ) ; : RETURN
685   PRINT : PRINT : PRINT "▲First▲Digit▲times▲Second":
      PRINT "▲Digit▲is▲";
688   PRINT D1 * D2 ; CHR$( 11 ) ; : RETURN
```

Player On (*)

```
750   VPOKE PP, 48
752   IF TS < 11 THEN RETURN
754   GOSUB 900 : I = INT( RN * 100 ) : IF I > 50 THEN
      TS = TS - 10 : PLAY "L64N45N46N50N48": RETURN
756   IF I < 10 THEN TS = TS + 50 : PLAY "L32N10N8N6N4N10N8N6
      N4": RETURN
760   TS = TS + 10 : PLAY "L64N10N8N6N4": RETURN
```

Game Over

```
800   CLS : IF M = PP THEN 820
```

```
805   FOR I = 1 TO 15 : PRINT "▲▲▲▲CONGRATULATIONS!":
      NEXT : PRINT
810   PRINT "YOU▲HAVE▲BEATEN▲THE▲MINOTAUR!": PRINT "YOUR▲SCOR
      E▲IS▲"; TS : PRINT
815   PRINT "NUMBER▲OF▲ROUNDS:"; R ; : GOTO 840
820   FOR I = 1 TO 18 : PRINT "▲▲▲▲▲▲▲MINOTAUR▲WINS!":
      NEXT : PRINT
825   PRINT "The▲secret▲number▲was▲"; SN
840   PRINT : PRINT "HIT▲ANY▲KEY▲FOR▲ANOTHER▲GAME"
845   X$ = INKEY$ : IF X$ = "" THEN 845
846   RUN
```

Random Number

```
900   RN =  ( 9999 * RN + RS ) MOD 5997! : RN = RN / 5997!
910   IF RN < .2 THEN RS = RN * 10000 + 1
920   RETURN
```

Blank Message

```
950   FOR I = 6209 TO 6271 : VPOKE I, 32 : NEXT
970   RETURN
```

Update Score

```
1000  FOR I = 1 TO 23 : PRINT : NEXT : PRINT "TOT.▲SCORE"
      TS ; TAB( 15 ) ; "LAST▲SCORE"; LS ; CHR$( 11 ) ;
1010  RETURN
```

Guess Secret Number

```
2000  PRINT : PRINT : PRINT "▲GUESS▲AT▲THE▲SECRET▲NUMBER":
      PRINT "▲";
2005  INPUT "AND▲HIT▲(RETURN)"; K$
2010  I = VAL( K$ ) : IF I = SN THEN 800
2015  G1 = INT( I / 10 ) : G2 = I - G1 * 10
2020  CLS : PRINT : IF D1 = G1 THEN PRINT "First▲Digit▲Correc
      t": PRINT
2025  IF D2 = G2 THEN PRINT "Second▲Digit▲Correct": PRINT
2035  IF D1 <> G1 AND D2 <> G2 THEN PRINT "No▲Digits▲Correct"
2040  FOR T = 1 TO 2000 : NEXT
2045  R = R + 1 : GOTO 100
```

Character Data

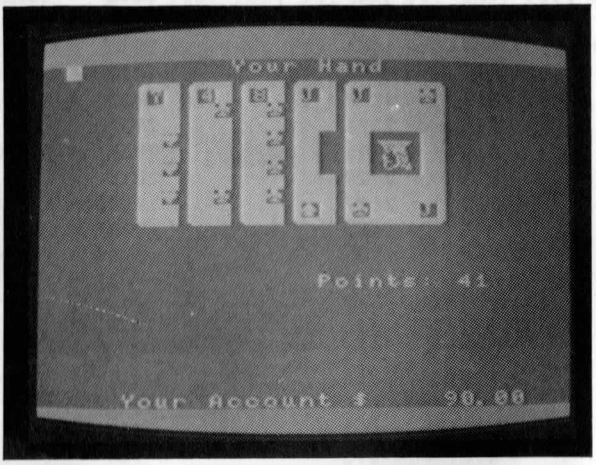
```
10000   DATA 165, 126, 219, 126, 102, 126, 82, 60
10002   DATA 136, 136, 80, 32, 80, 136, 136, 0
10004   DATA 112, 136, 8, 16, 32, 0, 32, 0
10006   DATA 32, 168, 112, 32, 112, 168, 32, 0
10008   DATA 16, 56, 124, 254, 124, 56, 16, 0
```

ChexSum Table

```
5     = 3376
6     = 403
7     = 1035
10    = 4336
15    = 2887
20    = 3348
22    = 3640
23    = 2630
25    = 3398
30    = 969
35    = 3770
40    = 10909
100   = 653
110   = 7694
115   = 2238
120   = 2239
125   = 822
130   = 320
140   = 4133
150   = 2831
160   = 397
200   = 734
205   = 1325
210   = 1798
215   = 1208
220   = 3347
225   = 1095
230   = 1036
240   = 345
242   = 2143
245   = 1850
250   = 1823
255   = 1838
260   = 1779
265   = 1771
270   = 977
275   = 1886
280   = 1802
285   = 1914
290   = 594
300   = 1856
302   = 1398
304   = 1427
305   = 537
310   = 1792
312   = 1398
314   = 1365
315   = 537
320   = 1792
322   = 1398
324   = 1364
325   = 537
330   = 1856
332   = 1398
334   = 1426
335   = 537
340   = 1846
342   = 1398
344   = 1411
345   = 537
350   = 1858
352   =  1398
354   = 1427
355   = 537
360   = 1846
362   = 1398
364   = 1410
365   = 537
370   = 1858
372   = 1398
374   = 1426
375   = 537
380   = 4638
385   = 391
400   = 3919
402   = 2128
405   = 2306
410   = 1197
415   = 1313
420   = 1162
422   = 3290
425   = 2245
430   = 1429
435   = 868
440   = 593
500   = 2962
502   = 1423
504   = 2345
506   = 1325
508   = 1212
510   = 2283
512   = 1294
515   = 884
517   = 2287
520   = 1293
522   = 831
540   = 3410
542   = 481
544   = 1424
546   = 481
550   = 431
570   = 143
600   = 2558
610   = 981
650   = 477
652   = 1831
654   = 3251
656   = 1724
```

```
658   = 2198
660   = 4907
662   = 1663
665   = 2493
666   = 3475
667   = 2592
668   = 932
670   = 4110
672   = 1233
673   = 3488
674   = 932
675   = 2245
676   = 2777
677   = 2816
678   = 932
680   = 2350
681   = 2773
682   = 2816
683   = 932
685   = 4543
688   = 1514
750   = 477
752   = 947
754   = 5284
756   = 3807
760   = 2259
800   = 1151
805   = 30 81
810   = 4837
815   = 2371
820   = 2991
825   = 2729
840   = 2697
845   = 1602
846   = 139
900   = 3412
910   = 2090
920   = 143
950   = 1584
970   = 143
1000  = 5164
1010  = 143
2000  = 3295
2005  = 1608
2010  = 1876
2015  = 2414
2020  = 3830
2025  = 3499
2035  = 4167
2040  = 1157
2045  = 1250
10000 = 1 770
10002 = 1572
10004 = 1340
10006 = 1567
10008 = 1517
Total= 323177
```

Thirty-One



CLASSIFICATION: Parlor Game

Inflation comes even to Blackjack in this game where the MSX challenges you to a game of cards! Add the value of your cards — if you are closer to 31 than the MSX then you win! Of course, a hand over 31 costs you (or the MSX) the game.

The card values are:

King	13
Queen	12
Jack	11
Ace	1
Others	Face value

PROGRAMMING SUGGESTIONS

The MSX has simple tactics — it stops when it reaches 25. If you have mastered this game, then make the MSX use better tactics, such as remembering what cards have been used and calculating probabilities for the next card. You could even let the MSX cheat a bit by peeking at the next card once in a while (see lines 300-310)!

PROGRAM

Variables

M1$, M2$	Messages
CA(I, J)	Card attribute table
DK(I)	Deck of cards
NC	Next card on deck
MB	Money bet this hand
AB	Account balance
PP	Player's points
CP	Computer's points
LC	Location of card
CD	Card to draw at LC
CC	Number of computer's cards
PC	Number of player's cards

Listing

Initialise

```
10   KEY OFF : SCREEN 1, 2 : COLOR 15, 4, 7
15   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : A$ = ""
20   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 1 ) = A$ : A$ = ""
25   FOR I = 1 TO 32 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 2 ) = A$ : A$ = ""
30   FOR I = 1072 TO 1143 : READ Q : VPOKE I, Q : NEXT
35   PRINT : PRINT : PRINT "▲▲▲▲▲▲▲▲▲THIRTY-ONE": VPOKE
     6153, 134 : FOR I = 6154 TO 6165 : VPOKE I, 135 :
     NEXT : VPOKE 6166, 136 : VPOKE 6198, 137 : VPOKE
     6230, 137 : VPOKE 6185, 141 : VPOKE 6217, 141 :
     FOR I = 6282 TO 6293 : VPOKE I, 139 : NEXT
40   VPOKE 6262, 137 : VPOKE 6294, 138 : VPOKE 6249,
     141 : VPOKE 6281, 140
45   FOR I = 384 TO 463 : K = VPEEK( I ) : VPOKE I, INT(
     K / 4 ) : NEXT
50   FOR I = 1 TO 6 : PRINT : NEXT : AB = 100 : PRINT
     "You▲Have▲$100▲to▲St▲▲▲▲art▲With": PRINT : PRINT
     "▲▲▲▲▲▲▲▲▲▲Hit▲a▲Key"
55   IF INKEY$ = "" THEN D = RND( 1 ) : GOTO 55
60   CLS : DIM CA ( 52, 2 ) , DK ( 52 )
65   FOR I = 1 TO 4 : FOR J = 1 TO 13 : K = ( I - 1 )
     * 13 + J : CA ( K, 1 ) = J : CA ( K, 2 ) = I : NEXT :
     NEXT : FOR I = 1 TO 52 : DK ( I ) = I : NEXT
75   AB = 100 : NC = 1
80   PRINT TAB( 10 ) "Your▲Hand"
90   PRINT CHR$( 11 ) ; : M1$ = "Please▲wait▲for▲deck▲to▲be
     ": M2$ = "shuffled.▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲": GOSUB 1000 :
     GOSUB 1100 : GOSUB 1200 : IF NC = 1 THEN GOSUB 400
95   GOSUB 300 : GOSUB 200 : GOSUB 200
```

Control

```
100  IF( ( ST ) AND 1 ) = 1 THEN 102 ELSE GOSUB 350
102  IF VPEEK( 6156 ) = 77 THEN 130 ELSE IF( ( ST ) AND
     1 ) <> 1 THEN 110
105  FOR I1 = 1 TO 5 : BEEP : PRINT CHR$( 11 ) TAB( 20 )
     : FOR I2 = 1 TO 300 : NEXT : PRINT CHR$( 11 ) TAB(
     10 ) "MSX▲Hand▲" CHR$( 11 ) ; : NEXT : FOR I = 6176
     TO 6623 : VPOKE I, 32 : NEXT : FOR I = 0 TO 2 :
     PUT SPRITE I, ( 100, 200 ) : NEXT
107  FOR I1 = 1 TO CC : IF I1 < 7 THEN LC = 6179 + 3
```

```
      * I1 ELSE LC = 6278 + 3 * ( I1 - 7 )
108   CD = CH ( I1 ) : GOSUB 250 : NEXT : FOR I = 1 TO
      15 : PRINT : NEXT : PRINT TAB( 15 ) "Points:"; CP ;
      CHR$( 11 ) ;
110   IF( ( ST ) AND 1 ) = 1 THEN 130
120   M1$ = "Your▲next▲card-▲▲▲▲▲▲▲▲▲▲▲": M2$ = "▲▲▲▲▲▲▲▲▲▲▲▲
      ▲▲▲▲▲▲▲▲▲▲▲▲▲": GOSUB 1000
125   GOSUB 200 : IF PP > 31 THEN 500
130   IF( ( ST ) AND 2 ) = 2 THEN 135 ELSE M1$ = "MSX:▲next▲c
      ard▲▲▲▲▲▲▲▲▲▲▲": M2$ = "▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲":
      GOSUB 1000 : FOR T = 1 TO 1500 : NEXT : GOSUB 300 :
      IF( CP > 24 AND CP < 32 ) THEN ST = ( ( ST ) OR
      2 ) ELSE IF CP > 31 THEN 600
135   IF ST = 3 THEN 450
140   GOTO 100
```

Deal Player

```
200   CD = DK ( NC ) : NC = NC + 1
205   PP = PP + CA ( CD, 1 ) : GOSUB 1100 : PC = PC + 1
210   IF PC < 7 THEN LC = 6179 + 3 * PC ELSE LC = 6278
      + 3 * ( PC - 7 )
215   GOSUB 250
240   RETURN
```

Draw Card On Screen

```
250   VPOKE LC, 134 : FOR I = LC + 1 TO LC + 5 : VPOKE
      I, 135 : NEXT : VPOKE LC + 6, 136 : FOR I = LC +
      33 TO LC + 289 STEP 32 : FOR J = 0 TO 4 : VPOKE
      I - 1, 141 : VPOKE I + 5, 137 : VPOKE I + J, 142 :
      NEXT : NEXT
255   VPOKE LC + 320, 140 : FOR I = LC + 321 TO LC + 325 :
      VPOKE I, 139 : NEXT : VPOKE LC + 326, 138
260   K1 = CA ( CD, 2 ) : IF K1 = 1 THEN K1 = 131 ELSE
      IF K1 = 2 THEN K1 = 129 ELSE IF K1 = 3 THEN K1 =
      128 ELSE K1 = 130
265   K2 = CA ( CD, 1 ) : ON K2 GOTO 270, 272, 274, 276,
      278, 280, 282, 284, 286, 288, 290, 292, 294
270   VPOKE LC + 33, 65 : VPOKE LC + 163, K1 : VPOKE LC
      + 293, 65 : RETURN
272   VPOKE LC + 33, 50 : VPOKE LC + 67, K1 : VPOKE LC
      + 259, K1 : VPOKE LC + 293, 50 : RETURN
274   VPOKE LC + 33, 51 : VPOKE LC + 67, K1 : VPOKE LC
      + 163, K1 : VPOKE LC + 259, K1 : VPOKE LC + 293,
      51 : RETURN
276   VPOKE LC + 33, 52 : VPOKE LC + 293, 52 : VPOKE LC
      + 66, K1 : VPOKE LC + 68, K1 : VPOKE LC + 258, K1 :
```

```
      VPOKE LC + 260, K1 : RETURN
278   VPOKE LC + 33, 53 : VPOKE LC + 293, 53 : VPOKE LC
      + 66, K1 : VPOKE LC + 68, K1 : VPOKE LC + 258, K1 :
      VPOKE LC + 260, K1 : VPOKE LC + 163, K1 : RETURN
280   VPOKE LC + 33, 54 : VPOKE LC + 293, 54 : VPOKE LC
      + 67, K1 : VPOKE LC + 130, K1 : VPOKE LC + 132,
      K1 : VPOKE LC + 194, K1 : VPOKE LC + 196, K1 : VPOKE
      LC + 259, K1 : RETURN
282   VPOKE LC + 33, 55 : VPOKE LC + 293, 55 : VPOKE LC
      + 67, K1 : VPOKE LC + 130, K1 : VPOKE LC + 132,
      K1 : VPOKE LC + 194, K1 : VPOKE LC + 196, K1 : VPOKE
      LC + 258, K1 : VPOKE LC + 260, K1 : RETURN
284   VPOKE LC + 33, 56 : VPOKE LC + 293, 56 : VPOKE LC
      + 66, K1 : VPOKE LC + 68, K1 : VPOKE LC + 130, K1 :
      VPOKE LC + 132, K1 : VPOKE LC + 194, K1 : VPOKE
      LC + 196, K1 : VPOKE LC + 258, K1 : VPOKE LC + 260,
      K1 : RETURN
286   VPOKE LC + 33, 57 : VPOKE LC + 293, 57 : VPOKE LC
      + 163, K1 : VPOKE LC + 66, K1 : VPOKE LC + 68, K1 :
      VPOKE LC + 130, K1 : VPOKE LC + 132, K1 : VPOKE
      LC + 194, K1 : VPOKE LC + 196, K1 : VPOKE LC + 258,
      K1 : VPOKE LC + 260, K1 : RETURN
288   VPOKE LC + 33, 49 : VPOKE LC + 293, 48 : VPOKE LC
      + 292, 49 : VPOKE LC + 34, 48 : VPOKE LC + 99, K1 :
      VPOKE LC + 227, K1 : VPOKE LC + 66, K1 : VPOKE LC
      + 68, K1 : VPOKE LC + 130, K1 : VPOKE LC + 132,
      K1 : VPOKE LC + 194, K1 : VPOKE LC + 196, K1 : VPOKE
      LC + 258, K1 : VPOKE LC + 260, K1 : RETURN
290   I = 8 * INT( ( LC - 6014 ) / 32 ) : J = 8 * ( LC
      - 6014 - 32 * I ) : PUT SPRITE 0, ( J + 4, I + 4 )
      : FOR I = LC + 130 TO LC + 194 STEP 32 : FOR J =
      0 TO 2 : VPOKE I + J, 32 : NEXT : NEXT : VPOKE LC
      + 33, 74 : VPOKE LC + 293, 74 : VPOKE LC + 37, K1 :
      VPOKE LC + 289, K1 : RETURN
292   I = 8 * INT( ( LC - 6014 ) / 32 ) : J = 8 * ( LC
      - 6014 - 32 * I ) : PUT SPRITE 1, ( J + 4, I + 4 )
      : FOR I = LC + 130 TO LC + 194 STEP 32 : FOR J =
      0 TO 2 : VPOKE I + J, 32 : NEXT : NEXT : VPOKE LC
      + 33, 81 : VPOKE LC + 293, 81 : VPOKE LC + 37, K1 :
      VPOKE LC + 289, K1 : RETURN
294   I = 8 * INT( ( LC - 6014 ) / 32 ) : J = 8 * ( LC
      - 6014 - 32 * I ) : PUT SPRITE 2, ( J + 4, I + 4 )
      : FOR I = LC + 130 TO LC + 194 STEP 32 : FOR J =
      0 TO 2 : VPOKE I + J, 32 : NEXT : NEXT : VPOKE LC
      + 33, 75 : VPOKE LC + 293, 75 : VPOKE LC + 37, K1 :
      VPOKE LC + 289, K1 : RETURN
298   RETURN
```

Deal MSX

```
300   CC = CC + 1
305   CD = DK ( NC ) : NC = NC + 1 : CP = CP + CA ( CD,
      1 ) : CH ( CC ) = CD
307   IF( ( ST ) AND 1 ) <> 1 THEN 310 ELSE IF CC < 7
      THEN LC = 6179 + 3 * CC ELSE LC = 6278 + 3 * ( CC - 7 )
308   GOSUB 250 : FOR I = 1 TO 15 : PRINT : NEXT : PRINT
      TAB( 15 ) "Points:"; CP ; CHR$( 11 ) ;
310   RETURN
```

Bet

```
350   GOSUB 1300 : FOR I = 1 TO 18 : PRINT : NEXT : IF
      MB <> 0 THEN 355 ELSE 360
355   INPUT "Change your bet (Y/N)"; X$ : IF X$ = "n" OR
      X$ = "N" THEN 365 ELSE INPUT "How much more"; X$ :
      IF VAL( X$ ) > 1000 THEN PRINT CHR$( 11 ) ; : GOTO
      350 ELSE IF VAL( X$ ) < 0 THEN PRINT CHR$( 11 ) ;
      : GOTO 350 ELSE MB = MB + VAL( X$ ) : AB = AB -
      VAL( X$ ) : GOTO 365
360   INPUT "What is your initial bet"; X$ : IF VAL( X$ )
      > 1000 THEN PRINT CHR$( 11 ) ; : GOTO 350 ELSE MB
      = VAL( X$ ) : AB = AB - VAL( X$ )
365   IF( ( ST ) AND 1 ) = 0 THEN INPUT "Another card (Y/N)";
      X$ : IF X$ <> "y" AND X$ <> "Y" THEN ST = ( ST ) OR 1
370   PRINT CHR$( 11 ) ; : GOSUB 1300 : GOSUB 1200 : RETURN
```

Shuffle Deck

```
400   FOR I = 1 TO 100 : J = INT( RND( 1 ) * 52 + 1 ) :
      K = INT( RND( 1 ) * 52 + 1 ) : SWAP DK ( J ) , DK ( K )
410   NEXT
440   RETURN
```

Show Hands

```
450   GOSUB 1300 : FOR I = 1 TO 18 : PRINT : NEXT : PRINT
      "MSX has"; CP ; "points": PRINT : K4 = 31 - CP :
      K5 = 31 - PP : IF K4 = K5 THEN PRINT "It's a a tie!! New
      DECK!": AB = AB + MB : GOSUB 400 : CLS : NC = 1 :
      GOTO 520
460   FOR T = 1 TO 2000 : NEXT : IF K4 > K5 THEN 600 ELSE
      500
```

Player Loses

```
500   PLAY "m59000s816n2r6n3r6n2": GOSUB 1300 : PRINT
      CHR$( 11 ) ; : FOR I = 1 TO 17 : PRINT : NEXT :
      PRINT "You▲Have▲lost▲this▲hand": PRINT "Your▲points";
      PP ; ",▲MSX:"; CP : INPUT "Another▲hand▲(Y/N)"; X$
505   IF X$ = "Y" OR X$ = "y" THEN IF NC < 25 THEN PRINT
      "OK▲-▲same▲deck": FOR T = 1 TO 2000 : NEXT : CLS :
      GOTO 520 ELSE PRINT "OK▲-▲new▲deck"; : FOR T = 1
      TO 2000 : NEXT : CLS : NC = 1 : GOTO 520
510   GOTO 900
520   MB = 0 : PP = 0 : ST = 0 : CP = 0 : CC = 0 : PC
      = 0 : FOR I = 0 TO 2 : PUT SPRITE I, ( 100, 200 )
      : NEXT : GOTO 80
```

Player Wins

```
600   PLAY "l1s14m3000n44", "l1s14m2000n27": GOSUB 1300 :
      PRINT CHR$( 11 ) ; : FOR I = 1 TO 17 : PRINT : NEXT :
      PRINT "You▲Have▲won▲this▲hand": PRINT "Your▲points";
      PP ; ",▲MSX:"; CP : INPUT "Another▲hand▲(Y/N)"; X$
605   IF X$ = "Y" OR X$ = "y" THEN IF NC < 25 THEN PRINT
      "OK▲-▲same▲deck": FOR T = 1 TO 2000 : NEXT : CLS :
      GOTO 620 ELSE PRINT "OK▲-▲new▲deck"; : FOR T = 1
      TO 2000 : NEXT : CLS : NC = 1 : GOTO 620
610   AB = AB + 2 * MB : GOTO 900
620   AB = AB + 2 * MB : MB = 0 : PP = 0 : ST = 0 : CP
      = 0 : CC = 0 : PC = 0 : FOR I = 0 TO 2 : PUT SPRITE
      I, ( 100, 200 ) : NEXT : GOTO 80
```

Game Over

```
900   SCREEN 1 : IF AB < = 0 THEN PRINT "YOU▲OWE▲$"; :
      PRINT USING "#####.##"; - AB : PRINT : PRINT "The▲Colle
      ctors▲will▲be▲": PRINT "visiting▲you▲shortly.": END
910   PRINT "You▲have▲won▲$"; : PRINT USING "#####.##";
      AB : PRINT : PRINT "▲Bet▲you▲can't▲do▲it▲twice!!": END
```

Print Message

```
1000  FOR I = 1 TO 18 : PRINT : NEXT : PRINT M1$ : PRINT :
      PRINT M2$ ; CHR$( 11 ) ;
1010  RETURN
```

Update Points

```
1100    FOR I = 1 TO 15 : PRINT : NEXT : PRINT TAB( 15 )
        "Points:"; PP ; CHR$( 11 ) ;
1110    RETURN
```

Print Account

```
1200    FOR I = 1 TO 23 : PRINT : NEXT : PRINT TAB( 4 )
        "Your▲Account▲$"; : PRINT USING "######.##"; AB ;
        : PRINT CHR$( 11 ) ;
1210    RETURN
```

Blankout Message Area

```
1300    FOR I = 6722 TO 6879 : VPOKE I, 32 : NEXT
1310    RETURN
```

Sprite Data

```
10000   DATA 255, 127, 42, 31, 19, 29, 9, 16, 32, 56, 16,
        28, 16, 16, 17, 14, 255, 254, 172, 248, 248, 248,
        188, 136, 116, 92, 84, 108, 106, 218, 151, 7
10010   DATA 0, 5, 11, 15, 31, 62, 60, 121, 120, 120, 120,
        124, 124, 124, 62, 31, 128, 80, 232, 248, 28, 12,
        14, 39, 7, 7, 199, 15, 207, 15, 30, 252
10020   DATA 4, 10, 15, 5, 7, 5, 10, 10, 10, 10, 21, 21,
        26, 54, 73, 48, 68, 170, 254, 84, 252, 72, 136,
        152, 132, 130, 28, 8, 24, 8, 8, 240
```

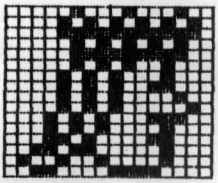
Character Data

```
10100   DATA 0, 0, 0, 0, 3, 7, 15, 15
10102   DATA 0, 0, 0, 0, 255, 255, 255, 255
10104   DATA 0, 0, 0, 0, 192, 224, 240, 240
10106   DATA 240, 240, 240, 240, 240, 240, 240, 240
10108   DATA 240, 240, 224, 192, 0, 0, 0, 0
10110   DATA 255, 255, 255, 255, 0, 0, 0, 0
10112   DATA 15, 15, 7, 3, 0, 0, 0, 0
10114   DATA 15, 15, 15, 15, 15, 15, 15, 15
10116   DATA 255, 255, 255, 255, 255, 255, 255, 255
```

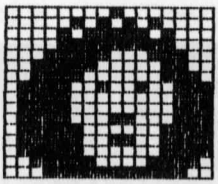
ChexSum Tables

```
10    = 1215
15    = 3995
20    = 3996
25    = 3997
30    = 1931
35    = 13381
40    = 3258
45    = 3631
50    = 6725
55    = 2290
60    = 1118
65    = 7604
75    = 967
80    = 1428
90    = 10893
95    = 1071
100   = 1925
102   = 3452
105   = 12244
107   = 5026
108   = 5255
110   = 1417
120   = 5276
125   = 1534
130   = 16298
135   = 1017
140   = 489
200   = 1636
205   = 2593
210   = 3992
215   = 410
240   = 143
250   = 11749
255   = 4868
260   = 6598
265   = 4132
270   = 3099
272   = 4014
274   = 5330
276   = 6514
278   = 7843
280   = 9481
282   = 10867
284   = 12206
286   = 13803
288   = 18778
290   = 17052
292   = 17067
294   = 17053
298   = 143
300   = 785
305   = 4120
307   = 6144
308   = 4084
310   = 143
350   = 3178
355   = 21896
360   = 9849
365   = 7154
370   = 1640
400   = 6153
410   = 131
440   = 143
450   = 14191
460   = 2802
500   = 14517
505   = 12322
510   = 523
520   = 6059
600   = 15852
605   = 12390
610   = 1797
620   = 7659
900   = 10318
910   = 6659
1000  = 2882
1010  = 143
1100  = 3524
1110  = 143
1200  = 5546
1210  = 143
1300  = 1682
1310  = 143
10000 = 9134
10010 = 8616
10020 = 7679
10100 = 1033
10102 = 1384
10104 = 1375
10106 = 1864
10108 = 1403
10110 = 1416
10112 = 1025
10114 = 1404
10116 = 1888

Total= 524162
```

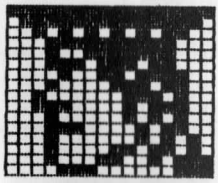
Sprite Shapes



KING

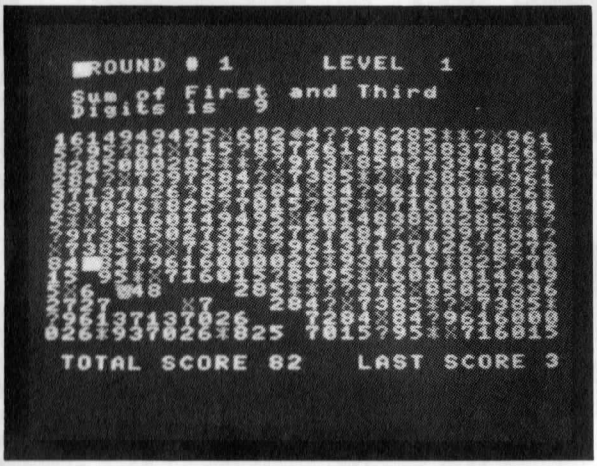


QUEEN



JACK

Minotaur Mastermind



CLASSIFICATION: Educational Game

This is an original strategic game which will test the brain power of even the most experienced masterminds. The challenge is to guess a secret number between 0 and 1000 by working your way through a maze of numbers and symbols, without being gobbled by the minotaur.

Your score is the sum of the digits that you step on — so the **lower** scores are better. When you reach the exit at the top of the maze, you get a chance to guess the secret number. The non-numeric symbols have different effects when stepped on.

X	Supplies a clue
?	Has random value
*	Randomly alters your score

PROGRAMMING SUGGESTIONS

Minotaur Mastermind could be made into a very complex game by adding more clues (lines 650-708) and different symbols.

You are represented by a flashing point; the controls are:

<P>	–	Up	<:>	–	Down/right
<Ø>	–	Up/left	<@>	–	Right
<;>	–	Down	<L>	–	Down/left
<O>	–	Left	<T>	–	Trapped
<->	–	Up/right			

PROGRAM Variables

LD	Level of difficulty
R, LS, TS	Round number; last score; total score
SN	Secret number
D1, D2, D3	Digits of secret number
PP	Player's position
M	Minotaur's position
C, Q	For flashing effect
TM	Time
RN, RS	Random number; random number seed
CU, CP	Character under minotaur; character under player
PR, PC	Player's row; player's column
NC	Number of clues
T(I)	Clue I used?
G1, G2, G3	Digits of player's guess
MC, MR	Minotuar's column; minotaur's row

Listing

Initialise

```
  5   TM = 1 : INTERVAL ON : ON  INTERVAL = 10 GOSUB 7
  6   GOTO 10
  7   TM = TM * 2 : RETURN
 10   KEY OFF : SCREEN 1 : CLS : COLOR 15, 1, 1 : PRINT
      "▲▲▲▲MINOTAUR▲MASTER-MIND"
 15   PRINT : PRINT : PRINT : PRINT "▲LEVEL▲OF▲DIFFICULTY?(1-
      9)";
 20   LD = VAL( INKEY$ ) : IF LD < 1 OR LD > 9 THEN 20
      ELSE PRINT LD
 22   FOR I = 1088 TO 1344 STEP 64 : FOR J = 0 TO 7 :
      READ Q : VPOKE I + J, Q : NEXT : NEXT
 23   VPOKE 8209, 49 : VPOKE 8210, 97 : VPOKE 8211, 177 :
      VPOKE 8212, 145 : VPOKE 8213, 129
 25   FOR I = 1 TO 8 : PRINT : NEXT : PRINT "▲▲▲▲▲HIT▲ANY▲KEY
      ▲TO▲START"
 30   IF INKEY$ = "" THEN 30
 35   INTERVAL  OFF :  IF  TM > 1000 THEN TM = TM / 3 :
      GOTO 35
 40   TS = 0 : NC = 0 : FOR I = 1 TO 10 : T ( I ) = 0 :
      NEXT : R = 1 : SN = INT( TM ) : RS = SN : D1 = INT(
      SN / 100 ) : D2 = INT( ( SN - D1 * 100 ) / 10 ) :
      D3 = INT( SN - D1 * 100 - D2 * 10 )
```

New Round

```
100   CLS : PP = 6767 : M = 6575
110   FOR I = 6304 TO 6752 STEP 32 : FOR J = 1 TO 30 :
      GOSUB 900 : K1 = INT( RN * 13 ) : IF K1 < 10 THEN
      VPOKE I + J, K1 + 48 : GOTO 130
115   IF K1 = 10 THEN VPOKE I + J, 144 : GOTO 130
120   IF K1 = 11 THEN VPOKE I + J, 152 : GOTO 130
125   VPOKE I + J, 160
130   NEXT : NEXT
140   VPOKE 6767, 48 : LS = 0 : CU = VPEEK( M ) : VPOKE
      M, 136 : VPOKE 6319, 168 : CP = 48
150   PRINT "▲ROUND▲#"; R ; "▲▲▲▲LEVEL▲"; LD ; CHR$( 11 );
160   GOSUB 1000
```

Editor

```
200   C = 0 : Q = 1
205   X$ = INKEY$ : C = C + 1
```

```
210   IF C = 7 THEN Q = 1 - Q : C = 0
215   IF Q <> 0 THEN 230
220   IF VPEEK( PP ) = 255 THEN VPOKE PP, CP : Q = 1 :
      GOTO 230
225   VPOKE PP, 255 : Q = 1
230   IF X$ = "" THEN 205
240   GOSUB 950
242   IF VPEEK( PP ) = 255 THEN VPOKE PP, CP
245   IF X$ = "p" OR X$ = "P" THEN 300
250   IF X$ = "o" OR X$ = "O" THEN 310
255   IF X$ = "@" OR X$ = "`" THEN 320
260   IF X$ = ";" OR X$ = "+" THEN 330
265   IF X$ = "-" OR X$ = "=" THEN 340
270   IF X$ = "0" THEN 350
275   IF X$ = "l" OR X$ = "L" THEN 360
280   IF X$ = ":" OR X$ = "*" THEN 370
285   IF X$ = "t" OR X$ = "T" THEN 380
290   GOTO 205
```

Recognise Allowed Moves

```
300   IF VPEEK( PP - 32 ) = 32 THEN 205
302   LS = VPEEK( PP ) - 48
304   VPOKE PP, 32 : PP = PP - 32
305   GOTO 400
310   IF VPEEK( PP - 1 ) = 32 THEN 205
312   LS = VPEEK( PP ) - 48
314   VPOKE PP, 32 : PP = PP - 1
315   GOTO 400
320   IF VPEEK( PP + 1 ) = 32 THEN 205
322   LS = VPEEK( PP ) - 48
324   VPOKE PP, 32 : PP = PP + 1
325   GOTO 400
330   IF VPEEK( PP + 32 ) = 32 THEN 205
332   LS = VPEEK( PP ) - 48
334   VPOKE PP, 32 : PP = PP + 32
335   GOTO 400
340   IF VPEEK( PP - 31 ) = 32 THEN 205
342   LS = VPEEK( PP ) - 48
344   VPOKE PP, 32 : PP = PP - 31
345   GOTO 400
350   IF VPEEK( PP - 33 ) = 32 THEN 205
352   LS = VPEEK( PP ) - 48
354   VPOKE PP, 32 : PP = PP - 33
355   GOTO 400
360   IF VPEEK( PP + 31 ) = 32 THEN 205
362   LS = VPEEK( PP ) - 48
364   VPOKE PP, 32 : PP = PP + 31
365   GOTO 400
```

```
370   IF VPEEK( PP + 33 ) = 32 THEN 205
372   LS = VPEEK( PP ) - 48
374   VPOKE PP, 32 : PP = PP + 33
375   GOTO 400
380   PR = INT( PP / 32 ) : PC = PP - PR * 32 : GOSUB
      500 : IF PP <> M THEN 380
385   GOTO 800
```

Check Move

```
400   CP = VPEEK( PP ) : PR = INT( PP / 32 ) : PC = PP
      - PR * 32
402   IF( LS > 9 ) OR( LS < 0 ) THEN LS = 0
405   K1 = VPEEK( PP ) : IF K1 = 152 THEN GOSUB 600
410   IF K1 = 144 THEN GOSUB 650
415   IF K1 = 160 THEN GOSUB 750
420   IF K1 = 168 THEN 2000
422   K1 = VPEEK( PP ) - 48 : IF K1 < 0 OR K1 > 9 THEN K1 = 0
425   IF LS + K1 > = 10 - LD THEN GOSUB 500
430   TS = TS + K1 : GOSUB 1000
435   IF PP = M THEN 800
440   GOTO 200
```

Move Minotaur

```
500   VPOKE 8198, 31 : MR = INT( M / 32 ) : MC = M - MR * 32
502   IF MC <> PC THEN 508
504   IF MR > PR  THEN MR = MR - 1 : GOTO 540
506   MR = MR + 1 : GOTO 540
508   IF MR <> PR  THEN 515
510   IF MC > PC THEN MC = MC - 1 : GOTO 540
512   MC = MC + 1 : GOTO 540
515   IF M > PP THEN 522
517   IF MC < PC THEN MC = MC + 1 : GOTO 540
520   MC = MC - 1 : GOTO 540
522   MR = MR - 1
540   K = 32 * MR + MC : VPOKE 8198, 241 : IF VPEEK( K )
      = 131 THEN RETURN
542   VPOKE M, CU
544   M = K : CU = VPEEK( M )
546   VPOKE M, 136
550   PLAY "n2"
570   RETURN
```

Player on (?)

```
600   GOSUB 900 : I = INT( RN * 10 ) : VPOKE PP, I + 48
610   CP = I + 48 : RETURN
```

Player on (X)

```
650   VPOKE PP, 48
652   IF NC > = 9 THEN BEEP : GOTO 670
654   GOSUB900 : K2 = INT( RN * 9 + 1 ) : IF T ( K2 )
      = 1 THEN 654
656   T ( K2 ) = 1 : NC = NC + 1 : BEEP
658   ON  K2  GOTO  660, 665, 670, 675, 680, 685, 690,
      695, 700
660   PRINT : PRINT : PRINT "Sum of Digits of Secret":
      PRINT "Number Equals ";
662   PRINT D1 + D2 + D3 : PRINT CHR$( 11 ) ; : RETURN
665   PRINT : PRINT : PRINT "Secret Number is ";
666   IF SN / 2 = INT( SN / 2 ) THEN PRINT "Even": GOTO 668
667   PRINT "Odd"
668   PRINT CHR$( 11 ) ; : RETURN
670   GOSUB 900 : K = INT( RN * 200 ) + SN : J = K - 200 :
      IF J < 0 THEN J = 0
672   IF K > 1000 THEN K = 1000
673   PRINT : PRINT : PRINT "Number is Between"; J : PRINT
      "and"; K
674   PRINT CHR$( 11 ) ; : RETURN
675   PRINT : PRINT : PRINT "Number is ";
676   IF SN / 5 = INT( SN / 5 ) THEN PRINT "Divisible":
      GOTO 678
677   PRINT "Not Divisible"
678   PRINT "by 5": PRINT CHR$( 11 ) ; : RETURN
680   L = D1 : IF D2 > L THEN L = D2
681   IF D3 > L THEN L = D3
682   PRINT : PRINT : PRINT "Largest Digit in the": PRINT
      "Secret Number is"; L ; CHR$( 11 ) ; : RETURN
685   PRINT : PRINT : PRINT "First Digit in the Secret":
      PRINT "Number is ";
686   IF D1 > 5 THEN PRINT "Larger Than 5": GOTO 689
687   IF D1 < 5 THEN PRINT "Less Than 5": GOTO 689
688   PRINT "Equal to 5"
689   PRINT CHR$( 11 ) ; : RETURN
690   PRINT : PRINT : PRINT "Sum of First and Third":
      PRINT "Digits is ";
692   PRINT D1 + D3 ; CHR$( 11 ) ; : RETURN
695   PRINT : PRINT : PRINT "One of the Digits in the":
      PRINT "Secret Number is ";
696   GOSUB 900 : K = INT( RN * 3 ) : IF K = 0 AND SN
      > 99 THEN PRINT D1 : GOTO 699
697   IF K = 1 AND SN > 9 THEN PRINT D2 : GOTO 699
698   PRINT D3
699   PRINT CHR$( 11 ) ; : RETURN
```

```
700   PRINT : PRINT : PRINT "Product▲of▲the▲Non-Zero":
      PRINT "Digits▲is";
701   IF D1 = 0 AND D2 = 0 THEN PRINT D3 : GOTO 708
702   IF D1 = 0 AND D3 = 0 THEN PRINT D2 : GOTO 708
703   IF D1 = 0 THEN PRINT D2 * D3 : GOTO 708
704   IF D2 = 0 AND D3 = 0 THEN PRINT D1 : GOTO 708
705   IF D2 = 0 THEN PRINT D1 * D3 : GOTO 708
706   IF D3 = 0 THEN PRINT D1 * D2 : GOTO 708
707   PRINT D1 * D2 * D3
708   PRINT CHR$( 11 ) ; : RETURN
```

Player on (*)

```
750   VPOKE PP, 48
752   IF TS < 11 THEN RETURN
754   GOSUB 900 : I = INT( RN * 100 ) : IF I > 50 THEN
      TS = TS - 10 : PLAY "L64N45N46N50N48": RETURN
756   IF I < 10 THEN TS = TS + 50 : PLAY "L32N10N8N6N4N10N8N6
      N4": RETURN
760   TS = TS + 10 : PLAY "L64N10N8N6N4": RETURN
```

Game Over

```
800   CLS : IF M = PP THEN 820
805   FOR I = 1 TO 17 : PRINT "▲▲▲▲▲CONGRATULATIONS!▲":
      NEXT : PRINT
810   PRINT "YOU▲HAVE▲BEATEN▲THE▲MINOTAUR": PRINT "YOUR▲SCORE
      ▲IS▲"; TS
815   PRINT "IN▲"; R ; "▲ROUNDS": GOTO 840
820   FOR I = 1 TO 17 : PRINT "MINOTAUR▲WINS▲▲MINOTAUR▲WINS":
      NEXT : PRINT
825   PRINT "The▲secret▲number▲was▲"; SN
840   PRINT : PRINT "HIT▲ANY▲KEY▲FOR▲ANOTHER▲GAME"
845   X$ = INKEY$ : IF X$ = "" THEN 845
846   RUN
```

Random Number

```
900   RN =  ( 9999 * RN + RS ) MOD 5997! : RN = RN / 5997!
910   IF RN < .2 THEN RS = RN * 10000 + 1
920   RETURN
```

Blank Out Message

```
950   FOR I = 6208 TO 6271 : VPOKE I, 32 : NEXT
970   RETURN
```

Update Score

```
1000    FOR I = 1 TO 21 : PRINT : NEXT : PRINT "TOTAL SCORE"
        TS ; TAB( 17 ) ; "LAST SCORE"; LS ; CHR$( 11 ) ;
1010    RETURN
```

Guess Secret Number

```
2000    PRINT : PRINT : PRINT "GUESS AT THE SECRET NUMBER"
2005    INPUT "AND HIT (RETURN)"; K$
2010    I = VAL( K$ ) : IF I = SN THEN 800
2015    G1 = INT( I / 100 ) : G2 = INT( ( I - G1 * 100 )
        / 10 ) : G3 = I - G1 * 100 - G2 * 10
2020    CLS : PRINT : IF D1 = G1 THEN PRINT "First Digit Correc
        t": PRINT
2025    IF D2 = G2 THEN PRINT "Second Digit Correct": PRINT
2030    IF D3 = G3 THEN PRINT "Third Digit Correct"
2035    IF D1 <> G1 AND D2 <> G2 AND D3 <> G3 THEN PRINT
        "No Digits Correct"
2040    FOR T = 1 TO 2000 : NEXT
2045    R = R + 1 : GOTO 100
```

Character Data

```
10000   DATA 165, 126, 219, 126, 102, 126, 82, 60
10002   DATA 136, 136, 80, 32, 80, 136, 136, 0
10004   DATA 112, 136, 8, 16, 32, 0, 32, 0
10006   DATA 32, 168, 112, 32, 112, 168, 32, 0
10008   DATA 16, 56, 124, 254, 124, 56, 16, 0
```

ChexSum Table

5	= 3376	100	= 1401	215	= 1208
6	= 403	110	= 7235	220	= 3347
7	= 1035	115	= 2238	225	= 1095
10	= 3831	120	= 2239	230	= 1036
15	= 2887	125	= 822	240	= 345
20	= 3348	130	= 320	242	= 2143
22	= 3640	140	= 4161	245	= 1850
23	= 2630	150	= 2754	250	= 1823
25	= 3476	160	= 397	255	= 1838
30	= 969	200	= 734	260	= 1779
35	= 3770	205	= 1325	265	= 1771
40	= 12987	210	= 1798	270	= 977

275 = 1886
280 = 1802
285 = 1914
290 = 594
300 = 1780
302 = 1398
304 = 1349
305 = 537
310 = 1791
312 = 1398
314 = 1365
315 = 537
320 = 1791
322 = 1398
324 = 1364
325 = 537
330 = 1780
332 = 1398
334 = 1348
335 = 537
340 = 1814
342 = 1398
344 = 1380
345 = 537
350 = 1824
352 = 1398
354 = 1396
355 = 537
360 = 1814
362 = 1398
364 = 1379
365 = 537
370 = 1824
372 = 1398
374 = 1395
375 = 537
380 = 4638
385 = 391
400 = 3919
402 = 2128
405 = 2306
410 = 1197
415 = 1313
420 = 1162
422 = 3290
425 = 2245
430 = 1429
435 = 868
440 = 593

500 = 2962
502 = 1423
504 = 2344
506 = 1325
508 = 1212
510 = 2282
512 = 1294
515 = 884
517 = 2286
520 = 1293
522 = 831
540 = 3410
542 = 481
544 = 1424
546 = 481
550 = 431
570 = 143
600 = 2558
610 = 981
650 = 477
652 = 1838
654 = 3251
656 = 1724
658 = 3150
660 = 4854
662 = 2064
665 = 2451
666 = 3357
667 = 497
668 = 932
670 = 4489
672 = 1542
673 = 3430
674 = 932
675 = 1696
676 = 4001
677 = 1548
678 = 1556
680 = 1767
681 = 1251
682 = 5960
685 = 4644
686 = 2960
687 = 2739
688 = 1125
689 = 932
690 = 4204
692 = 1513
695 = 5220

696 = 4300
697 = 2351
698 = 266
699 = 932
700 = 4374
701 = 2407
702 = 2407
703 = 2127
704 = 2408
705 = 2125
706 = 2124
707 = 1018
708 = 932
750 = 477
752 = 947
754 = 5284
756 = 3807
760 = 2259
800 = 1151
805 = 3152
810 = 4559
815 = 1883
820 = 3962
825 = 2729
840 = 2697
845 = 1602
846 = 139
900 = 3412
910 = 2090
920 = 143
950 = 1584
970 = 143
1000 = 5315
1010 = 143
2000 = 2804
2005 = 1608
2010 = 1876
2015 = 5740
2020 = 3830
2025 = 3499
2030 = 3150
2035 = 5189
2040 = 1157
2045 = 1250
10000 = 1770
10002 = 1572
10004 = 1340
10006 = 1567
10008 = 1517

Total= 376804

Appendix A
Using Joysticks

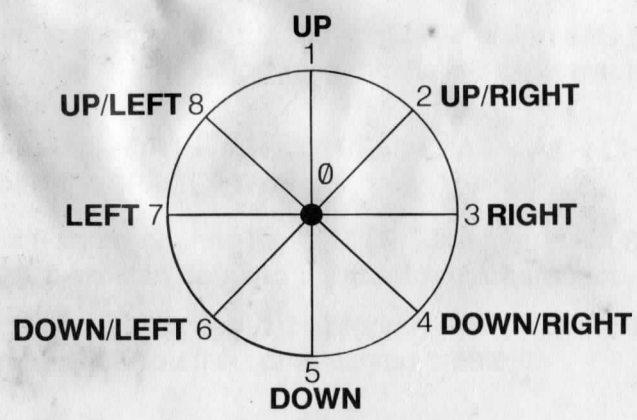
To convert keyboard games to joystick games, you will need to replace USR calls to location 60000 (which reads the keyboard) with a BASIC sub-routine similar to the one below. [Use STICK(1) to read joystick 1 or STICK(2) for joystick 2; STICK(0) reads the keyboard.]

```
A = STICK(1)
IF (A = 0) THEN RETURN [to main program]
IF (A = 1) OR (A = 2) OR (A = 8) THEN [move up]
IF (A = 8) OR (A = 7) OR (A = 6) THEN [move left]
IF (A = 6) OR (A = 5) OR (A = 4) THEN [move down]
IF (A = 2) OR (A = 3) OR (A = 4) THEN [move right]
```

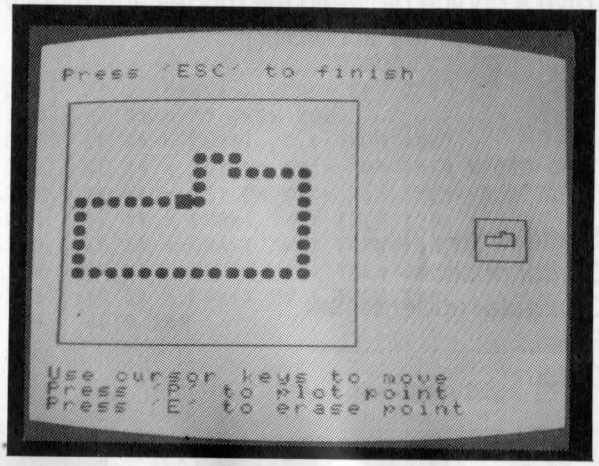
These 'move' statements could be USR calls to location 60118 accompanied by appropriate pokes (see Appendix C, 'Machine Code Support Program').

To alter firing, simply change any references to STRIG(0) to STRIG(1) for joysick 1, or STRIG(2) for joystick 2.

JOYSTICK VALUES, STRIG(n)



Appendix B
Sprite Maker



This program makes the creation of sprites a very easy task. You create a sprite on a large grid by moving the cursor around and plotting points. When you are finished, the program supplies you with the numbers that you need to create the sprite in your own program.

If these numbers are placed in DATA statements in your program, then the following statements will create a 16 × 16 sprite:

```
10 A$ = " ": FOR I = 1 TO 32: READ Q: A$ = A$ + CHR$(Q):
   NEXT I: SPRITE$(O) = A$
```

For an 8 × 8 sprite, change the '32' to '8'. Note that, in 8 × 8 mode, the sprite maker can be used for redefining characters.

Instructions are included in the program and the box to the right of the grid shows you the sprite as you create it.

PROGRAMMING SUGGESTIONS

The colour of the sprite can be set by adding the colour number to the end of the 'PUT SPRITE' statement in lines 120 and 520.

If you want to see your sprite in the enlarged mode, you could make the program give a choice between the modes and modify the boxes surrounding the sprites accordingly.

PROGRAM Variables

SA(I, J)	Sprite Array (1 to Ø)
CP	Cursor postion
CU	Character under cursor
RW	Row
CO	Column

LISTING

Initialise

```
10   KEY OFF : COLOR 1, 15, 15 : SCREEN 1, 2 : PRINT
     "▲▲▲▲▲▲▲▲▲SPRITE▲MAKER"
15   PRINT : PRINT : PRINT : INPUT "Choose▲8x8▲('8')▲or▲▲▲▲▲
     ▲▲▲▲▲▲▲▲▲▲▲▲16x16▲('16')"; X$
20   IF X$ = "16" THEN 500 ELSE IF X$ = "8" THEN 100 ELSE
     PRINT "Pardon?": GOTO 15
```

Set-up Screen (8 × 8 sprite)

```
100  CLS : PRINT "▲▲▲Press▲'ESC'▲to▲finish": FOR I =
     1 TO 5 : PRINT : NEXT
105  PRINT "▲▲▲+--------+": FOR I = 1 TO 8 : PRINT "▲▲▲¦▲▲▲▲
     ▲▲▲▲¦":  NEXT : PRINT "▲▲▲+--------+"
110  PRINT : PRINT : PRINT "▲▲▲Use▲cursor▲keys▲to▲move":
     PRINT : PRINT "▲▲▲Press▲'P'▲to▲plot▲point": PRINT :
```

```
      PRINT "▲▲▲Press▲'E'▲to▲erase▲point"
115   CP = 6473 : CU = 32 : RW = 4 : CO = 4
120   A$ = "": SPRITE$( 0 ) = A$ : PUT SPRITE 0, ( 167, 86 )
125   VPOKE 6484, 24 : VPOKE 6485, 23 : VPOKE 6486, 25 :
      VPOKE 6516, 22 : VPOKE 6518, 22 : VPOKE 6549, 23 :
      VPOKE 6548, 26 : VPOKE 6550, 27
```

Control (8 × 8)

```
200   X$ = INKEY$ : VPOKE CP, 255
250   IF X$ = "" THEN 200
255   IF X$ = CHR$( 27 ) THEN 400
260   IF X$ = "p" OR X$ = "P" THEN VPOKE CP, CU : GOTO 300
265   IF X$ = "E" OR X$ = "e" THEN VPOKE CP, CU : GOTO 310
270   IF X$ = CHR$( 30 ) THEN 320
275   IF X$ = CHR$( 29 ) THEN 330
280   IF X$ = CHR$( 31 ) THEN 340
285   IF X$ = CHR$( 28 ) THEN 350
290   GOTO 200
```

Move/Plot/Erase

```
300   IF VPEEK( CP ) = 133 THEN 200 ELSE CU = 133 : SA
      ( RW, CO ) = 1 : GOTO 380
310   IF VPEEK( CP ) = 32 THEN 200 ELSE CU = 32 : SA (
      RW, CO ) = 0 : GOTO 380
320   IF RW < 2 THEN 200 ELSE RW = RW - 1 : VPOKE CP,
      CU : CU = VPEEK( CP - 32 ) : CP = CP - 32 : GOTO 200
330   IF CO < 2 THEN 200 ELSE CO = CO - 1 : VPOKE CP,
      CU : CU = VPEEK( CP - 1 ) : CP = CP - 1 : GOTO 200
340   IF RW > 7 THEN 200 ELSE RW = RW + 1 : VPOKE CP,
      CU : CU = VPEEK( CP + 32 ) : CP = CP + 32 : GOTO 200
350   IF CO > 7 THEN 200 ELSE CO = CO + 1 : VPOKE CP,
      CU : CU = VPEEK( CP + 1 ) : CP = CP + 1 : GOTO 200
```

Print Sprite (8 × 8)

```
380   K2 = 14335 + RW : K = VPEEK( K2 ) : K1 = 2 ^( 8
      - CO ) : IF SA ( RW, CO ) = 1 THEN K = K + K1 :
      VPOKE K2, K : GOTO 200
385   K = K - K1 : VPOKE K2, K : GOTO 200
```

Finish (8 × 8)

```
400   CLS : PRINT "Numbers▲for▲'DATA'▲statements": PRINT :
      PRINT : PRINT
410   FOR I = 14336 TO 14342 : PRINT USING "###"; VPEEK( I)
      ; : PRINT "▲"; : NEXT
415   PRINT : PRINT : PRINT : PRINT USING "###"; VPEEK(
      14343 )
420   FOR I = 1 TO 4 : PRINT : NEXT : PRINT "▲Hit▲any▲key▲to▲
      start▲again" : PUT SPRITE 0, ( 127, 150 )
430   IF INKEY$ = "" THEN 430 ELSE RUN
```

Set-up Screen (16 × 16)

```
500   CLS : PRINT "Press▲'ESC'▲to▲finish": PRINT
505   PRINT "+----------------+": FOR I = 1 TO 16 : PRINT
      "#▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲#": NEXT : PRINT "+----------------+"
510   PRINT : PRINT "Use▲cursor▲keys▲to▲move": PRINT "Press▲
      P'▲to▲plot▲point": PRINT "Press▲'E'▲to▲erase▲point";
515   DIM SA ( 16, 16 ) : CP = 6474 : CU = 32 : RW = 8 :
      CO = 8
520   A$ = "": SPRITE$( 0 ) = A$ : PUT SPRITE 0, ( 216, 86 )
525   VPOKE 6490, 24 : VPOKE 6491, 23 : VPOKE 6492, 23 :
      VPOKE 6493, 25 : VPOKE 6522, 22 : VPOKE 6525, 22 :
      VPOKE 6554, 22 : VPOKE 6557, 22 : VPOKE 6587, 23 :
      VPOKE 6588, 23 : VPOKE 6586, 26 : VPOKE 6589, 27
```

Control (16 × 16)

```
600   X$ = INKEY$ : VPOKE CP, 255
650   IF X$ = "" THEN 600
655   IF X$ = CHR$( 27 ) THEN 800
660   IF X$ = "p" OR X$ = "P" THEN VPOKE CP, CU : GOTO 700
665   IF X$ = "E" OR X$ = "e" THEN VPOKE CP, CU : GOTO 710
670   IF X$ = CHR$( 30 ) THEN 720
675   IF X$ = CHR$( 29 ) THEN 730
680   IF X$ = CHR$( 31 ) THEN 740
685   IF X$ = CHR$( 28 ) THEN 750
690   GOTO 600
```

Move/Plot Erase (16 × 16)

```
700   IF VPEEK( CP ) = 133 THEN 600 ELSE CU = 133 : SA
      ( RW, CO ) = 1 : GOTO 780
```

```
710   IF VPEEK( CP ) = 32 THEN 600 ELSE CU = 32 : SA (
      RW, CO ) = 0 : GOTO 780
720   IF RW < 2 THEN 600 ELSE RW = RW - 1 : VPOKE CP,
      CU : CU = VPEEK( CP - 32 ) : CP = CP - 32 : GOTO 600
730   IF CO < 2 THEN 600 ELSE CO = CO - 1 : VPOKE CP,
      CU : CU = VPEEK( CP - 1 ) : CP = CP - 1 : GOTO 600
740   IF RW > 15 THEN 600 ELSE RW = RW + 1 : VPOKE CP,
      CU : CU = VPEEK( CP + 32 ) : CP = CP + 32 : GOTO 600
750   IF CO > 15 THEN 600 ELSE CO = CO + 1 : VPOKE CP,
      CU : CU = VPEEK( CP + 1 ) : CP = CP + 1 : GOTO 600
```

Print Sprite (16 × 16)

```
780   IF CO < 9 THEN K1 = 0 ELSE K1 = 16
785   K2 = 14335 + RW + K1 : K = VPEEK( K2 ) : IF CO <
      9 THEN K1 = 2 ^( 8 - CO ) ELSE K1 = 2 ^( 16 - CO )
790   IF SA ( RW, CO ) = 1 THEN K = K + K1 : VPOKE K2,
      K : GOTO 600
795   K = K - K1 : VPOKE K2, K : GOTO 600
```

Finish (16 × 16)

```
800   CLS : PRINT "Numbers▲for▲'DATA'▲statements": PRINT
      "▲Read▲from▲left▲to▲right": PRINT
810   FOR I = 14335 TO 14356 STEP 7 : PRINT : PRINT :
      FOR J = 1 TO 7 : PRINT USING "###"; VPEEK( I + J )
      ; : PRINT "▲"; : NEXT : NEXT
815   PRINT : PRINT : FOR I = 14364 TO 14367 : PRINT USING
      "###"; VPEEK( I ) ; : PRINT "▲"; : NEXT
820   FOR I = 1 TO 3 : PRINT : NEXT : PRINT "▲Hit▲any▲key▲to▲
      start▲again" : PUT SPRITE 0, ( 120, 165 )
830   IF INKEY$ = "" THEN 830 ELSE RUN
```

ChexSum Table

```
10    = 3219
15    = 4148
20    = 4379
100   = 4028
105   = 4018
110   = 10384
115   = 2034
120   = 2258
125   = 5134
200   = 1337
250   = 1032
255   = 1449
260   = 2947
265   = 2916
270   = 1373
275   = 1373
280   = 1389
285   = 1389
290   = 593
300   = 4243
310   = 3938
320   = 6213
330   = 6174
340   = 6210
350   = 6171
380   = 7500
385   = 2007
400   = 4116
410   = 2988
415   = 1934
420   = 5353
430   = 1484
500   = 2698
505   = 6460
510   = 8747
515   = 2718
520   = 2297
525   = 9318
600   = 1337
650   = 920
655   = 1292
660   = 3075
665   = 3076
670   = 1516
675   = 1516
680   = 1532
685   = 1532
690   = 483
700   = 4051
710   = 3750
720   = 5975
730   = 5936
740   = 6031
750   = 6008
780   = 1845
785   = 6451
790   = 3234
795   = 1879
800   = 6481
810   = 5442
815   = 3392
820   = 5345
830   = 1374

Total= 229442
```

Appendix C
Machine-Code Support Program

SUPPORT PROGRAM

This program is required for a number of the games in this book. It must be loaded and RUN before you load your game program.

1. Type this program in and save it. Use ChexSum to check it and, when bug-free, save it as "SUPPRT".
2. Load Support and RUN it.
3. Load your debugged game program and RUN it; it will call and use support as required.

Support Listing

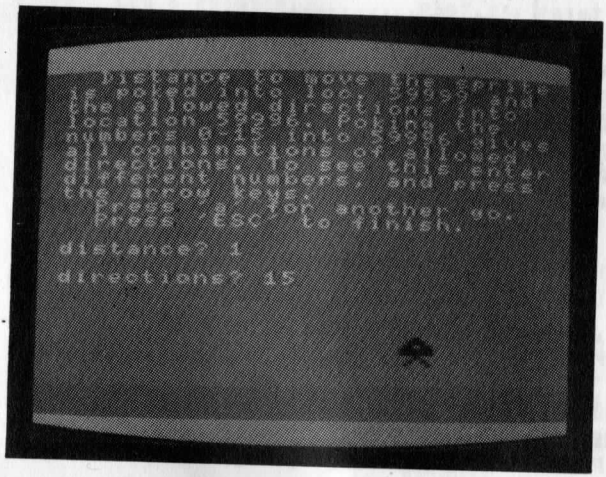
```
  10   CLS : PRINT "MACHINE▲CODE▲SUPPORT▲PROGRAM"
  20   FOR I = 60000! TO 60217! : READ Q : POKE I, Q : NEXT
1000   DATA 219, 170, 230, 240, 33, 92, 234, 94, 246, 8,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 211, 170, 219,
       169, 71, 0, 0, 0, 0, 0, 0, 0, 0
1010   DATA 0, 0, 0, 230, 128, 194, 151, 234, 203, 91,
       202, 151, 234, 62, 3, 50, 94, 234, 195, 209, 234,
       120, 230, 64, 194, 170, 234, 203, 83, 202, 170,
       234, 62, 2
1020   DATA 50, 94, 234, 195, 209, 234, 120, 230, 32, 194,
       189, 234, 203, 67, 202, 189, 234, 62, 0, 50, 94,
       234, 195, 209, 234, 120, 230, 16, 194, 208, 234,
       203, 75, 202, 208, 234, 62, 1, 50, 94, 234, 195,
       209, 234, 201, 62, 0, 50, 93, 234, 58
1030   DATA 93, 234, 203, 39, 203, 39, 79, 6, 0, 33, 0,
       27, 9, 58, 95, 234, 71, 58, 94, 234, 87, 254, 0,
       194, 250, 234, 205, 27, 235, 121, 152, 79, 195,
       44, 235, 254
1040   DATA 1, 194, 6, 235, 35, 205, 27, 235, 195, 244,
       234, 254, 2, 194, 20, 235, 205, 27, 235, 121, 128,
       79, 195, 44, 235, 35, 205, 27, 235, 195, 14, 235,
       125, 243, 211
```

```
1050   DATA 153, 124, 211, 153, 0, 0, 0, 0, 219, 152, 79,
       183, 251, 201, 125, 243, 211, 153, 124, 246, 64,
       211, 153, 121, 211, 152, 251, 201
```

Support ChexSum Table

```
10     = 2712
20     = 2233
1000   = 7068
1010   = 10317
1020   = 21367
1030   = 10298
1040   = 11087
1050   = 8282

Total= 73364
```

DEMONSTRATION PROGRAM



This program gives an explanation and demonstration of the machine-code support program used to move sprites in many of the games in this book. It is divided into two parts, Part 1 involving the reading of the keyboard, and Part 2 the moving of sprites.

All instructions are included in the program.

Demonstration Program Suggestions

Part 2 only allows the user to choose directions for moving the sprite. The '1' in line 90 can be altered to give different speeds, or you can restructure the program to give the user the choice of speeds when the program runs.

Demonstration Variables

X$	Key Pressed
K	Temporary

Demonstration Listing

Initialise

```
 5   REM RUN MACHINE CODE
 6   REM SUPPORT PROGRAM
 7   REM SEE APPENDICES
10   COLOR 15, 4, 7 : SCREEN 1, 1 : KEY OFF
15   FOR I = 1 TO 8 : READ Q : A$ = A$ + CHR$( Q ) :
     NEXT : SPRITE$( 0 ) = A$ : GOTO 200
```

Part 1: Read Keyboards

```
20   CLS : PRINT "  DEMONSTRATION OF MACHINE": PRINT
     "    CODE SUPPORT PROGRAM": PRINT : PRINT "          P
     ART 1": PRINT
25   DEFUSR0 = 60000! : PRINT "  The first routine starts at
     location 60000. Its purpose": PRINT "is to read the key
     board and": PRINT "move sprite 0 if the arrow":
     PRINT "keys are pressed."
30   PRINT "  The 4 allowed directions": PRINT "are numbered
     as follows:": PRINT
35   PRINT "              0": PRINT : PRINT "              U":
     PRINT "              V": PRINT "          1 UWWU 3":
     PRINT "              V": PRINT "              U":
     PRINT : PRINT "              2": PRINT : PRINT "  Hit an
     y key to continue";
40   IF INKEY$ = "" THEN 40
45   CLS : PRINT "  Distance to move the sprite is poked into
     loc. 59999 and": PRINT "the allowed directions into":
     PRINT "location 59996. Poking the": PRINT "numbers 0-1
     5 into 59996 gives";
50   PRINT "all combinations of allowed": PRINT "directions.
     To see this enter different numbers, and press":
     PRINT "the arrow keys.": PRINT "  Press 'a' for another
     go.": PRINT "  Press 'ESC' to finish.": PRINT
55   INPUT "distance"; D : PRINT : INPUT "directions";X$
60   PRINT : PRINT : PUT SPRITE 0, ( 120, 175 ) , 1
65   POKE 59999!, D : POKE 59996!, VAL( X$ ) : I = USR0(
     I ) : A$ = INKEY$ : IF A$ = "a" OR A$ = "A" THEN
     55 ELSE IF A$ = CHR$( 27 ) THEN PUT SPRITE 0, (
     100, 200 ) : GOTO 200 ELSE 65
```

Part 2: Move Sprites

```
 75   CLS : PRINT "▲▲▲▲▲▲▲▲▲▲PART▲2": PRINT : PRINT "▲▲This▲r
      outine▲starts▲at▲loc."; : PRINT "60118▲and▲moves▲a▲sprit
      e.": PRINT "The▲sprite▲number▲must▲be▲▲▲▲poked▲into▲59
      997,▲direction▲▲into▲59998▲and▲distance▲into▲59999."
 80   PRINT "▲▲Hit▲'RETURN'▲to▲change▲▲▲▲▲direction.":
      PRINT "▲▲Hit▲'ESC'▲to▲finish.": PRINT
 85   DEFUSR1 = 60118! : POKE 59997!, 0 : PUT SPRITE 0,
      ( 10, 175 ) , 1
 90   POKE 59998!, 1 : POKE 59999!, 1
 95   I = USR1( I ) : X$ = INKEY$ : IF X$ = "" THEN 95
      ELSE IF X$ = CHR$( 27 ) THEN PUT SPRITE 0, ( 100,
      200 ) : GOTO 200 ELSE IF X$ <> CHR$( 13 ) THEN 95
 96   INPUT "Which▲direction▲(0-3)"; X$ : K = VAL( X$ )
      : IF K < 0 OR K > 3 THEN PRINT "Pardon?": GOTO 96
      ELSE POKE 59998!, K : CLS : PRINT "▲▲Hit▲'RETURN'▲to▲ch
      ange▲▲▲▲▲direction.": PRINT "▲▲Hit▲'ESC'▲to▲finish.":
      GOTO 95
```

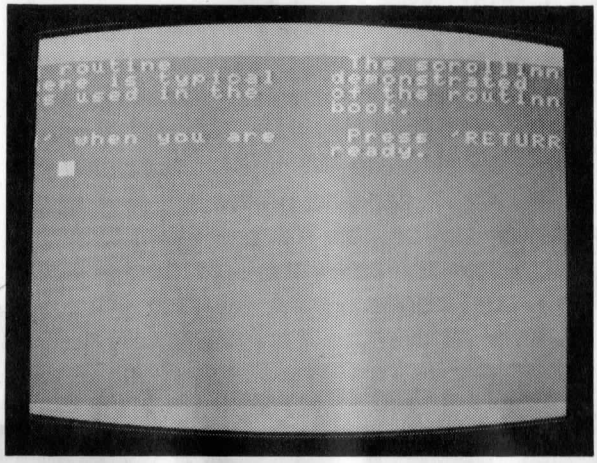
Menu

```
200   CLS : PRINT "▲MACHINE▲CODE▲DEMONSTRATION"
205   FOR I = 1 TO 4 : PRINT : NEXT : PRINT TAB( 18 )
      "Press▲Key": PRINT : PRINT : PRINT "▲▲▲▲DEMO▲PART▲1▲▲▲
      ▲▲▲'1'": PRINT : PRINT : PRINT "▲▲▲▲DEMO▲PART▲2▲▲▲▲▲▲▲'
      2'": PRINT : PRINT : PRINT "▲▲▲▲FINISH▲▲▲▲▲▲▲▲▲▲▲▲'F'"
210   FOR I = 1 TO 4 : PRINT : NEXT : PRINT "▲Press▲one▲of▲th
      e▲above▲keys": PRINT "▲▲to▲make▲your▲selection"
220   X$ = INKEY$ : IF X$ = "1" THEN 20 ELSE IF X$ = "2"
      THEN 75 ELSE IF X$ = "F" OR X$ = "f" THEN SCREEN
      1 : PRINT "▲▲▲▲GOOD-BYE!": END ELSE 220
1000  DATA 24, 60, 102, 255, 255, 24, 36, 66
```

Demonstration ChexSum Table

5	= 0	40	= 979	90	= 1523
6	= 0	45	= 20547	95	= 8437
7	= 0	50	= 23681	96	= 19288
10	= 1235	55	= 3192	200	= 2619
15	= 4280	60	= 1400	205	= 13114
20	= 8057	65	= 10795	210	= 7199
25	= 19414	75	= 24879	220	= 8044
30	= 5884	80	= 6587	1000	= 1583
35	= 20681	85	= 2879	Total=	216297

Appendix D
Machine-Code Scrolling Routine



There are several machine-code scrolling routines used in this book, but all are very similar to the one used in this demonstration. All intructions are included in the program.

PROGRAM

Listing of BASIC Loader

```
 5   REM RUN MACHINE CODE
 6   REM SUPPORT PROGRAM
 7   REM SEE APPENDICES
10   SCREEN 1 : COLOR 15, 4, 7 : KEY OFF : FOR I = 60350!
     TO 60401! : READ Q : POKE I, Q : NEXT
15   DEFUSR = 60350! : POKE 60346!, 0 : POKE 60347!,
     24 : POKE 60348!, 255 : POKE 60349!, 26
```

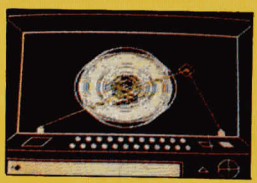
```
   20    PRINT "▲▲▲DEMONSTRATION▲OF▲MACHINE": PRINT "▲▲▲▲CODE▲SC
         ROLLING▲ROUTINE"
   25    FOR I = 1 TO 10 : PRINT : NEXT : PRINT "▲▲▲▲Press▲any▲k
         ey▲to▲start": PRINT : PRINT "▲▲▲▲Press▲'ESC'▲to▲finish"
   30    IF INKEY$ = "" THEN 30
   40    CLS : PRINT "▲▲The▲scrolling▲routine▲▲▲▲▲▲▲demonstrated
         ▲here▲is▲typical▲of▲the▲routines▲used▲in▲the▲▲book.▲"
   45    PRINT : PRINT "▲▲Press▲'RETURN'▲when▲you▲are▲ready."
   50    IF INKEY$ = CHR$( 13 ) THEN 55 ELSE 50
   55    D = USR( D ) : X$ = INKEY$ : IF X$ = CHR$( 27 )
         THEN SCREEN 1 : PRINT "GOOD-BYE!": END
   60    GOTO 55
10000    DATA 6, 31, 42, 186, 235, 43, 35, 205, 27, 235,
         120, 254, 31, 202, 215, 235, 4, 43, 205, 44, 235,
         35, 195, 227, 235, 6, 0, 17, 31, 0, 25, 205, 44,
         235, 183, 237, 82, 237, 91, 188, 235, 123, 189,
         194, 196, 235, 122, 188, 194, 196, 235, 201
```

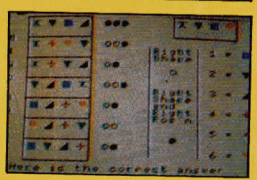
ChexSum Table

```
 5          = 0
 6          = 0
 7          = 0
 10         = 3647
 15         = 4235
 20         = 5324
 25         = 6972
 30         = 969
 40         = 10464
 45         = 3749
 50         = 1782
 55         = 4360
 60         = 444
 10000      = 21635
Total= 63581
```

MSX

Melbourne House





The best new software games book for your MSX!

These easy-to-enter program listings turn your MSX computer into an arcade of electronic fun and thrills. There are games that test your reflexes, your nerve, your logic, your strategy and your intelligence — educational games, simulation games, adventure games and much more!

These games programs make maximum use of the sophisticated MSX features, including its sprite capabilities and excellent sound function.

Use the unique ChexSum verification program to ensure that your games are bug free. Screen shots preview each program. Program structures are outlined, and key variables are given so that you can modify and extend these games for even more thrill-packed action!

This book will open the door for all MSX owners to a fascinating new dimension of computer game action and excitement.

Melbourne House Publishers

ISBN 0-86161-172-1

9 780861 611720