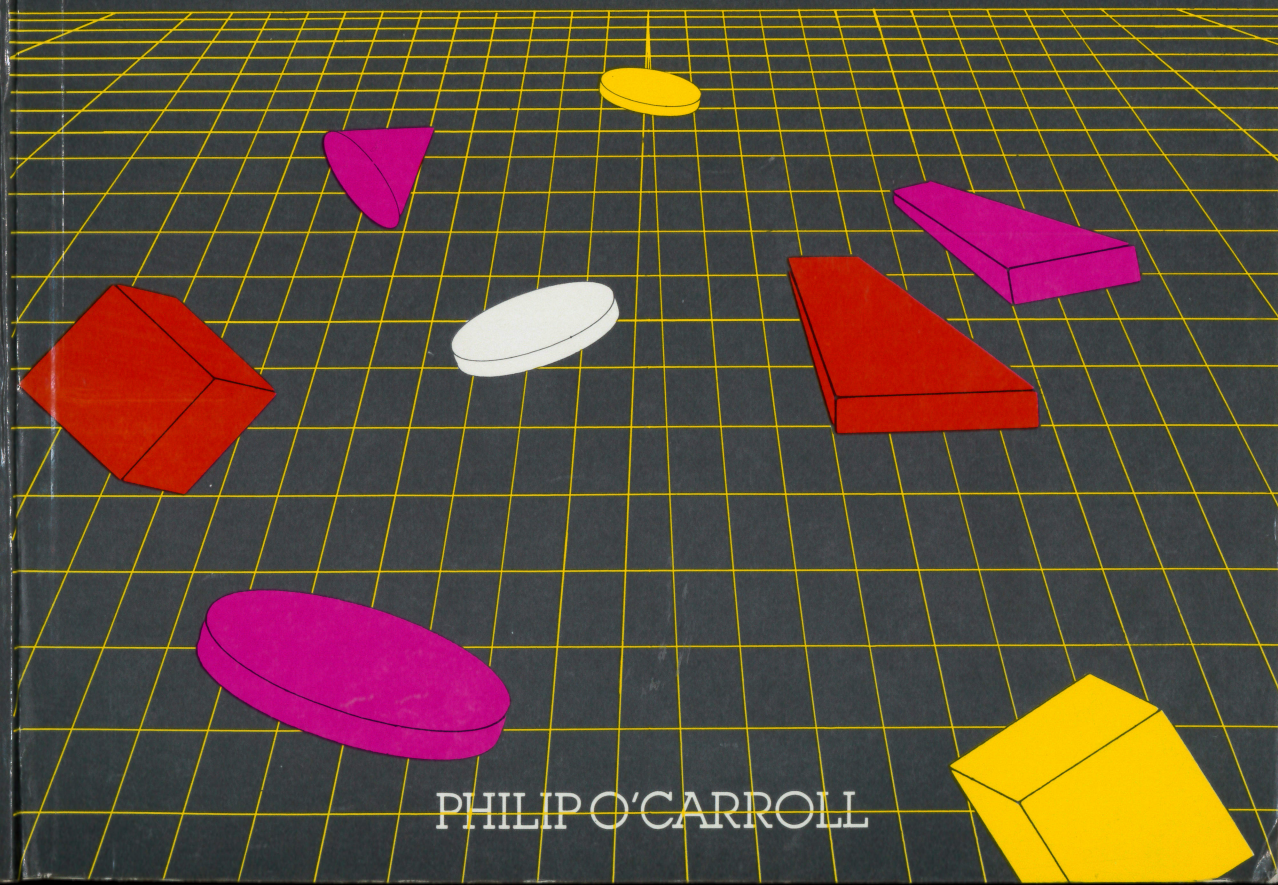
21 FABULOUS PROGRAMS FOR YOUR SEGA SC3000 COMPUTER



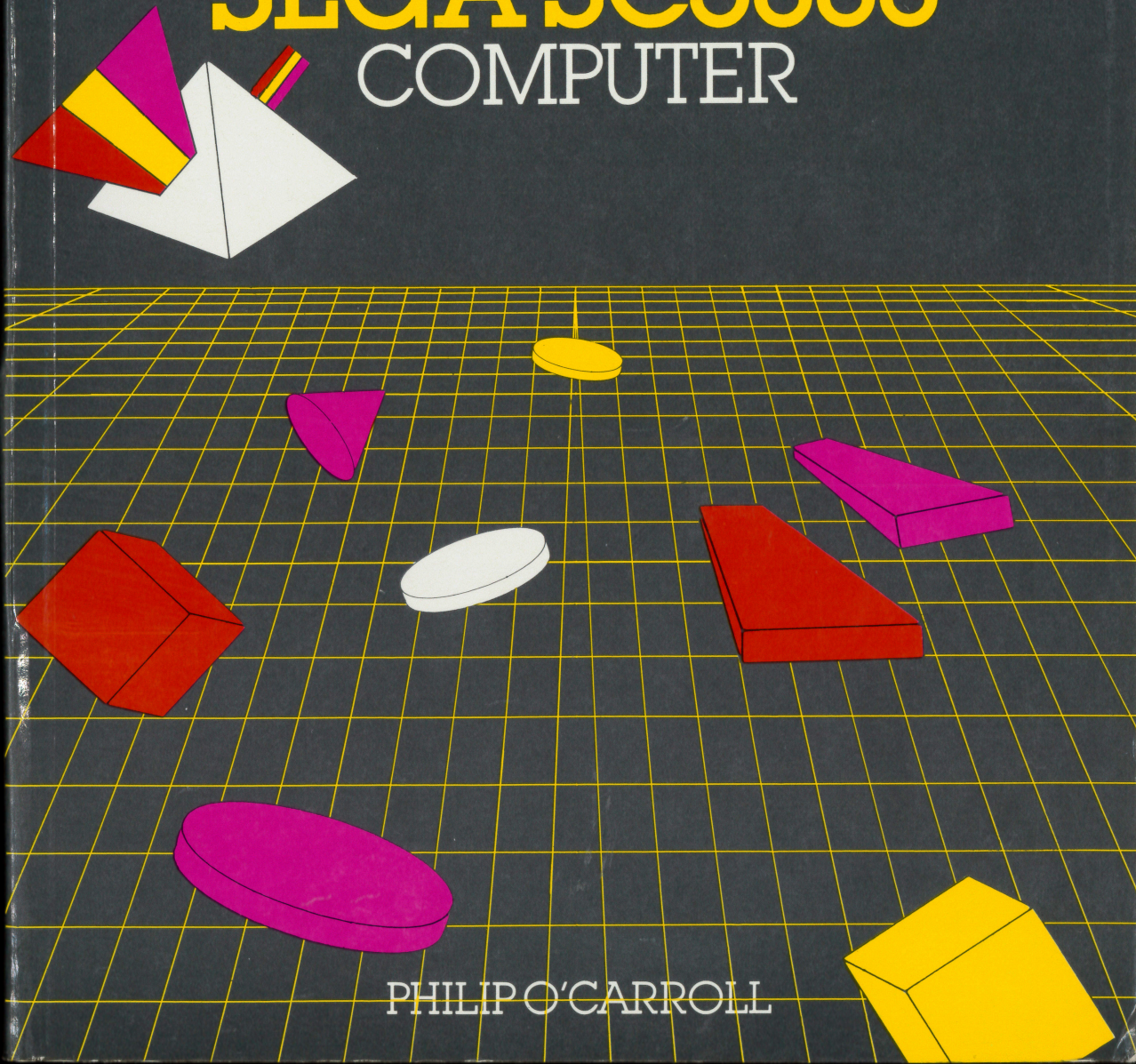
PHILIP O'CARROLL

21 FABULOUS PROGRAMS FOR YOUR SEGA SC3000 COMPUTER

PHILIP O'CARROLL

PITMAN

21 FABULOUS PROGRAMS FOR YOUR SEGA SC3000 COMPUTER



PHILIP O'CARROLL

21 FABULOUS PROGRAMS FOR YOUR SEGA SC3000 COMPUTER

PHILIP O'CARROLL
FITZROY COMMUNITY SCHOOL, MELBOURNE

PITMAN

First published 1985

Pitman Publishing Pty Ltd
(Incorporated in Victoria)

158 Bouverie Street
Carlton
Victoria 3053

Level 12
Town Hall House
452–462 Kent Street
Sydney
New South Wales 2000

9th Floor
National Bank Building
420 George Street
Brisbane
Queensland 4000

National Library of Australia
Cataloguing in Publication data

O'Carroll, Philip, 1945–
21 fabulous programs for your Sega SC3000 computer.

ISBN 0 85896 210 1.

1. Sega SC3000 (Computer). 2. Computer programs. I. Title. II. Title: Twenty-one fabulous programs for your Sega SC3000 computer.
001.64'24

Printed in Australia
by Tripart Marketing

Associated companies

Pitman Publishing Ltd
London

Copp Clark Pitman
Toronto

Pitman Publishing New Zealand Ltd
Wellington

CONTENTS

Mini-checkers	5
Rangefinder	15
Concentration	22
Tic-Tac-Toe	29
Mastermind	36
Hangman	43
Springnet	48
Steering Test	53
Speedcar	58
Duck Shoot	65
Balance Beam	69
Light Show	74
Horse Race	77
Superdice	80
Drawing Pad	84
Sega Poet	87
Mini Piano	92
Quiz Master	97
First Right Guess	102
Countdown	106
Home Budget	111
Word Editor	115

Before you start

This book presents a collection of twenty-one ready-made programs for the Sega SC3000 computer. It's designed to provide enjoyment for people of all ages and interests. The programs range from the simple *First right guess* to the sophisticated *Mini-checkers*. There are challenging action games (like *Springnet*) and useful 'home-office' facilities (like *Word editor*). And there are your old favourites — *Hangman, Mastermind* and *Tic-Tac-Toe* — all in original versions.

All of the programs are ready to use right away. All you have to do is key them into your computer. Notes on *How to use* the program accompany each of the program listings.

For programmers (or for people teaching themselves how to write programs) the main routines in the listings are clearly identified with REM headings. People who can handle a little BASIC programming should be able to make their own changes and additions to them without much trouble. If you're in this category, the sections called *Notes on the program* and *Changes you can make* should prove useful. Adapting programs with your own ideas is a good way to have fun while you're learning BASIC.

How to use this book

Type the programs *exactly* as they are listed. Be sure to include all of the punctuation marks: quotes (''), semi-colons (;), colons (:) and commas (,). Note carefully the *spaces* where they appear inside quotation marks. You can omit other spaces, but not the ones inside quotation marks. (The job is easier if one person reads out the lines of the program and another person types them in.)

Note carefully the difference between " " and "". There is a great difference between "SPACE" and "NOTHING". The space is actually character 32. It puts a space on the screen (erasing anything previously in that position) and moves the cursor one place to the right. The other is not a character at all.

You can save yourself some typing by omitting anything *after* the word REM in any line. It's quite safe and won't affect the program. You can also safely change the wording of anything that appears *inside* quotation marks. For example, for PRINT"PRESS BAR" you could substitute PRINT"PRESS THE SPACE BAR WHEN READY", if you preferred.

Always hit CR at the end of each instruction (ie, before you key the next line number). If you don't your computer won't remember it.

To eliminate an instruction, simply type its line number and hit CR.

If you've lost sight of your program, type LIST and hit CR.

To stop a program, hit the BREAK key. If this doesn't work, press RESET. You won't lose the program unless you type NEW and hit CR.

When you are running several programs, one after the other, it is best to RESET between each program. If you don't, special characters used in earlier programs may appear in later ones!

Saving your programs

When you've typed in a program, SAVE it on tape before you do anything else. After you've SAVEd it, give it a test RUN, correct any bugs in it and then SAVE the new version. You'll never need to type it in again.

To SAVE a program, first REWIND the tape and then STOP it. Type SAVE on your Sega keyboard and hit CR. Now press SAVE on the tape drive. (It's wise to note the names of the programs you've recorded on the label of the cassette.)

If you're not sure what's on a tape, attempt to LOAD"XXX". The screen will display the names of all the programs on the tape.

The text screen and the graphics screen

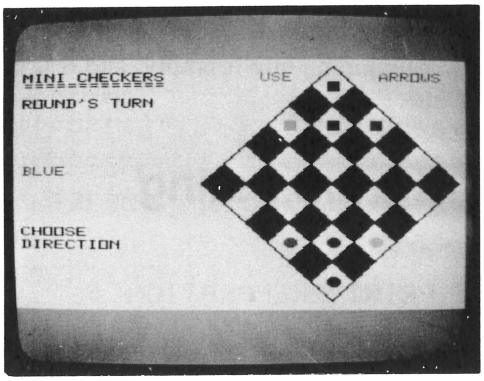
The Sega SC3000 makes two separate screens available to the user: a *text screen* and a *graphics screen*. The text screen is purely for text — ie, words, numbers, and the graphic characters that are shown on the keyboard. It only lets you use one color at a time (in addition to the screen color), although you can create colorful effects by changing the colors during a program. The graphics screen, on the other hand, can display all fifteen colors at the same time. It also permits high resolution graphics and the use of sprites (the user-programmable object blocks).

The programs in this book use both types of screen — sometimes within the same program. Where you have a lot of text and require automatic scroll, you need to use the text screen. The graphic screen can handle a few words but you cannot *print over* them (without first going through a *clear screen*). You can, however, *erase* by printing over in *screen color*. The advantages of the graphic screen, of course, are its abilities to display multicolors and any dot, line, circle or sprite, anywhere on the screen. One or two of the programs in this book that use the text screen could be adapted for the graphic screen with some thoughtful fiddling. There is a challenge for the reader!

Graphic characters and CHR$ codes

A number of the programs in this book make use of the standard graphic characters that appear on the keyboard (eg, card suits, solid blocks, checkered patterns). In the program listings, however, you won't see the symbols themselves. Instead, you'll see PRINT CHR$ and then the code number of the graphic character within brackets. This may make an instruction look cumbersome but it's unavoidable because the Sega SP400 printer will not reproduce graphic characters. If you wanted to, you could look up the list of character codes in your Sega manual and then type the actual graphic character instead of CHR$ and the code number. For example, where you read PRINT CHR$(229) you could type PRINT"■" instead.

MINI-CHECKERS



This game is a variation on traditional checkers. Each player has four pieces, each piece a different color. As is usual with checkers, the aim is to move all your pieces to the opposite end. You can move in any direction, but if you make five pieces in one horizontal row, you lose.

How to play

You are advised on screen when it is your turn. You then indicate which COLOR you will move (by letters) and in which DIRECTION (by arrows).

Notes on the program

For simplicity's sake, we have in line 57 set aside an imaginary grid of 7x7 spaces, although we are only using about half of these for the game. In lines 58 to 60 we begin by outlawing the extreme outside spaces. Our game grid sits squarely within our imaginary 7x7 grid. Then in lines 65 to 67 we have to close off more illegal squares to make our diamond shaped board. Lines 70 and 72 insert certain numbers into certain locations to give us pieces of the right shape and color to begin the game. Lines 90 to 93 give us our square and lines 95 to 98 give us our circle. Lines 674 to 677 read which of the cursor arrows has been pressed and alter the X or Y value accordingly. The variable NY refers to the new Y position and the variable PY refers to the present Y position (same for X). Lines 755 and 756 simply add the contents of the four win positions and if they total the right amount, that player has won. The number of each piece equals five times its color code number (plus one, if it's a square).

Changes you can make

You can make a board with more squares and use more pieces. With a bigger board, you can include a 'jumping' option, just as you have in Chinese Checkers. All you have to do is include a GOSUB instruction to check the legality of the jump move. The GOSUB routine checks whether the square you're jumping into is empty, and if it is, uses a step of two instead of a step of one.

Program listing

```
40 CLS
45 PRINT"PREPARATION
47 REM PX=PRESENT X:PY=PRESENT Y
48 REM NX=NEW X:NY=NEW Y
49 C$(0)=CHR$(236):REM ROUND
50 C$(1)=CHR$(229):REM SQUARE
52 REM BASIC DATA
55 N$(1)="SQUARE"
56 N$(0)="ROUND"
57 DIML(7,7),X(6,7),Y(6,7)
58 FORF=0TO7
59 L(F,0)=2:L(0,F)=2:L(6,F)=2:L(F,7)=2
: REM OUTSIDE SPACES ILLEGAL
60 NEXT
61 FORF=1TO5:FORG=1TO6
62 L(F,G)=0:REM EMPTY SPACES FIRST
63 NEXT:NEXT
65 L(1,6)=2:L(2,6)=2:L(4,6)=2:L(5,6)=2
:REM ILLEGAL PLACES
66 L(1,2)=2:L(5,2)=2:L(1,5)=2:L(5,5)=2
:REM ILLEGAL PLACES
67 L(1,1)=2:L(2,1)=2:L(4,1)=2:L(5,1)=2
:REM ILLEGAL PLACES
68 REM INITIAL OCCUPANTS OF SQUARES
70 L(3,6)=20:L(2,5)=40:L(3,5)=60:L(4,5
)=50:REM FIRST CIRCLE POSITIONS
72 L(3,1)=21:L(2,2)=51:L(3,2)=61:L(4,2
```

```
)=41:REM FIRST SQUARE POSITIONS
80 FORF=1TO5:FORG=1TO6
82 X(F,G)=105+24*F
84 Y(F,G)=-14+30*G
86 NEXT:NEXT
90 PATTERNS#5,"FEFEFEFEFEFEFEFE"
91 PATTERNS#9,"FEFEFEFEFEFEFEFE"
92 PATTERNS#11,"FEFEFEFEFEFEFEFE"
93 PATTERNS#13,"FEFEFEFEFEFEFEFE"
95 PATTERNS#4,"3C7EFFFFFFFF7E3C"
96 PATTERNS#8,"3C7EFFFFFFFF7E3C"
97 PATTERNS#10,"3C7EFFFFFFFF7E3C"
98 PATTERNS#12,"3C7EFFFFFFFF7E3C"
100 REM INSTRUCTIONS
110 SCREEN1,1:CLS
115 COLOR1,15
120 PRINT"MINI-CHECKERS
130 PRINT"-------------
132 PRINT:PRINT
134 PRINT"EACH PLAYER HAS 4 PIECES.
135 PRINT
136 PRINT"ONE PLAYER HAS ROUND PIECES;
  ";C$(0);" ";C$(0)
138 PRINT"THE OTHER HAS SQUARE PIECES.
  ";C$(1);" ";C$(1)
142 BEEP
143 PRINT
144 PRINT"ONE PLAYER PRESS THE BAR NOW

146 PRINT"TO SEE WHICH PIECES YOU WILL
 HAVE:
147 PRINT
148 PRINT"PRESS BAR WHEN READY......
154 A$=INKEY$
156 IFA$<>" "THEN154
160 GOSUB800
164 K$=RIGHT$(TIME$,1)
```

```
166 IFK$="0"THEN164
168 K=VAL(K$)
170 IFK/2=INT(K/2)THENT=1:GOTO174
172 T=0
174 PRINT:PRINT
175 PRINT"YOU HAVE THE ";N$(T);" PIECE
S:
178 PRINT"YOUR FRIEND HAS THE ";N$(1-T
);" PIECES.
180 PRINT:PRINT
182 PRINT"PRESS BAR TO CONTINUE...
184 A$=INKEY$
186 IFA$<>" "THEN184
188 GOSUB800
190 PRINT:PRINT
192 PRINT"YOUR AIM IS TO GET ALL YOUR
193 PRINT"PIECES TO THE OTHER END.
194 PRINT:PRINT:PRINT"WITH THIS SPECIA
L RULE:
195 PRINT"IF YOU MAKE 5 PIECES IN 1 RO
W,
196 PRINT"YOU LOSE THE GAME!
197 PRINT:PRINT:PRINT"PRESS BAR WHEN R
EADY......":GOSUB 1000
198 A$=INKEY$
199 IFA$<>" "THEN198
200 REM DECIDE FIRST TURN
210 K$=RIGHT$(TIME$,1)
220 IFK$="0"THEN210
230 K=VAL(K$)
240 IFK/2=INT(K/2)THENTN=1:GOTO250
245 TN=0
250 REM
300 REM BOARD SCREEN
310 SCREEN2,2:CLS
320 COLOR1,15,(0,0)-(255,191),3
330 FORF=0TO6
```

```
334 X1=108+F*12
338 Y1=95-F*15
342 X2=X1+72
346 Y2=Y1+90
350 LINE(X1,Y1)-(X2,Y2),1
354 NEXT
360 FORF=0TO6
364 X1=108+F*12
368 Y1=95+F*15
372 X2=X1+72
376 Y2=Y1-90
380 LINE(X1,Y1)-(X2,Y2),1
384 NEXT
400 REM CHEQUERED PATTERN
405 COLOR1
410 FORX=114TO246STEP12
412 PAINT(X,95)
414 NEXT
420 FORX=144TO216STEP24
421 FORG=-30TO30STEP60
422 PAINT(X,95+G)
424 NEXT:NEXT
430 FORX=168TO192STEP24
431 FORG=-60TO60STEP120
432 PAINT(X,95+G)
434 NEXT:NEXT
450 REM FIRST POSITIONS
460 GOSUB900
500 REM TEXT ON SCREEN
505 COLOR1
510 CURSOR10,10
512 PRINT"MINI CHECKERS
516 CURSOR10,16
520 PRINT"=============
530 COLOR12
540 CURSOR140,10
542 PRINT"USE         ARROWS
```

```
544 COLOR1
550 REM DECIDE WHOSE TURN
552 COLOR15:CURSOR10,72:PRINT"RED"
553 CURSOR10,82:PRINT"BLUE"
554 CURSOR10,92:PRINT"GREEN"
555 CURSOR10,102:PRINT"YELLOW"
560 TN=1-TN
572 COLOR13
574 CURSOR10,30+TN*10
576 PRINTN$(TN);"'S TURN"
578 COLOR15
580 CURSOR10,30+(1-TN)*10
582 PRINTN$(1-TN);"'S TURN"
600 REM COLOR
610 COLOR1
612 CURSOR10,49
613 PRINT"CHOOSE COLOR"
614 CURSOR10,57
615 PRINT" (R,B,G,Y)
616 GOSUB 1200
620 A$=INKEY$
624 IFA$="R"THEN630
625 IFA$="B"THEN632
626 IFA$="G"THEN634
627 IFA$="Y"THEN636
628 GOTO620
630 COLOR8:CURSOR10,72:PRINT"RED":N=40
+TN
631 GOTO640
632 COLOR4: CURSOR10,82:PRINT"BLUE":N=
20+TN
633 GOTO640
634 COLOR12:CURSOR10,92:PRINT"GREEN":N
=60+TN
635 GOTO640
636 COLOR10:CURSOR10,102:PRINT"YELLOW"
:N=50+TN
```

```
637 GOTO640
640 GOSUB1000
641 FORF=1TO5:FORG=1TO6
642 IFL(F,G)=NTHENPX=F:PY=G:F=5:G=6
644 NEXT:NEXT
645 COLOR15:CURSOR10,49
646 PRINT"CHOOSE COLOR"
647 CURSOR10,57
648 PRINT" (R,B,G,Y)
649 GOSUB 1200
650 REM DIRECTION
652 NX=PX:NY=PY:REM NEW=OLD
655 COLOR1
660 CURSOR10,129
662 PRINT"CHOOSE"
664 CURSOR10,139
666 PRINT"DIRECTION
670 A$=INKEY$
674 IFA$=CHR$(28)THENNX=PX+1:GOTO 680
675 IFA$=CHR$(29)THENNX=PX-1:GOTO 680
676 IFA$=CHR$(30)THENNY=PY-1:GOTO 680
677 IFA$=CHR$(31)THENNY=PY+1:GOTO 680
678 GOTO670
680 IFL(NX,NY)THEN GOSUB694: TN=1-TN:
GOTO550
690 GOTO700
694 FORF=1TO4
696 CURSOR10,157:PRINT"ILLEGAL!"
697 COLOR15:GOSUB1100
698 CURSOR10,157:PRINT"ILLEGAL!"
699 COLOR1:NEXT:IL=1:GOTO702
700 REM UPDATE DISPOSITION
701 GOSUB1000
702 COLOR15:REM ERASE CHOOSE
703 CURSOR10,129
704 PRINT"CHOOSE"
705 CURSOR10,139
```

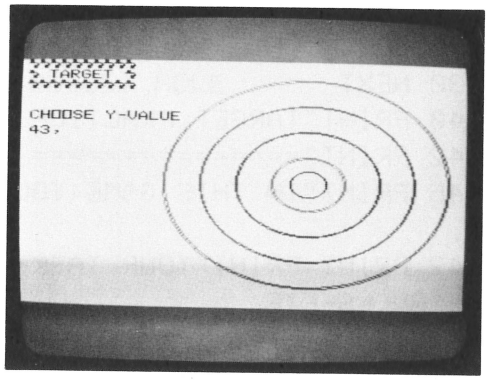
```
706 PRINT"DIRECTION
708 IFILTHENIL=0:RETURN
710 L(PX,PY)=0 :REM EMPTY OLD SPACE
711 L(NX,NY)=N :REM INSERT NEW VALUE
713 REM CHECK FOR LOSE
714 NO=1:MO=1:FORF=1TO5:IFL(F,3)=0THEN
NO=0
715 NEXT:FORF=1TO5:IFL(F,4)=0THENMO=0
716 NEXT:IFNO OR MO THENCOLOR1: CURSOR
10,165 : PRINT"5 IN A ROW! " ; N$(TN);
" LOSES." : FORF=1TO4:GOSUB1100: NEXT:
GOTO792
720 GOSUB950:REM NEW DISPLAY
730 PX=NX:PY=NY:REM UPDATE PRESENT VAL
UES
750 REM CHECK FOR WIN
755 IFL(3,1)+L(2,2)+L(3,2)+L(4,2)=170
THEN770
760 IFL(2,5)+L(3,5)+L(4,5)+L(3,6)=174
THEN770
765 GOTO550:REM NEXT TURN
770 FORF=1TO8
771 COLOR15
772 CURSOR10,165
773 SOUND1,110,15
774 PRINTN$(TN);" WINS!"
775 FORG=1TO10:NEXT
776 COLOR13
777 SOUND0
778 CURSOR10,165
780 PRINTN$(TN);" WINS!"
782 NEXT
790 COLOR1
792 CURSOR10,180
793 PRINT"Press BAR for more...
794 A$=INKEY$
795 IFA$<>" "THEN794
```

```
796 BEEP2
797 GOTO 300
800 REM EFFECTS
810 FORG=1TO15
820 COLOR1,G
830 SOUND1,110+RND(4)*110,15
840 NEXT:SOUND0
850 RETURN
900 REM DISPLAY CURRENT DISPOSITION
910 FORF=1TO5:FORG=1TO6
914 Z=L(F,G)
918 IFZ=2THEN930:REM ILLEGAL
920 X=X(F,G):Y=Y(F,G)
921 CURSORX,Y
922 COLORZ/5
923 IFZ=0THEN930:REM EMPTY
924 W=Z-10*INT(Z/10):REM SHAPE
925 S=W+INT(Z/10)*2
926 SPRITES,(X,Y),S,Z/5
928 BEEP
930 NEXT:NEXT
940 RETURN
950 REM MOVE 1 SPRITE
955 C=2*INT(N/10):REM COLOR NUMBER
963 COLORC
967 W=N-10*INT(N/10):REM SHAPE
979 S=W+C
981 SPRITES,(X(NX,NY),Y(NX,NY)),S,C
983 BEEP
985 RETURN
1000 REM DING
1010 FORF=15TO0STEP-1
1020 SOUND1,500,F
1030 NEXT
1040 SOUND0
1050 RETURN
1100 REM ERROR SOUND
```

```
1110 SOUND1,110,15
1115 SOUND2,150,15
1120 FORGG=1TO30:NEXT
1130 SOUND0
1140 RETURN
1200 REM YOUR TURN SOUND
1210 FORF=1TO8
1220 SOUND1,350,15
1230 SOUND1,400,15
1240 NEXT
1245 SOUND0
1250 RETURN
```

RANGEFINDER

This is an excellent test of spatial intelligence. You are confronted with a target, and the object, of course, is to score a bull's-eye. You 'aim' your rifle by naming a point which is the intersection of two coordinates — the horizontal and the vertical. Your first shots will be a matter of pure chance because you have no idea of the scale of either of the coordinates. You don't know how much real distance a single increment along each coordinate will produce — it's different in every new game. If you don't get the bull's-eye within seven shots, you're out of ammunition.



How to play

Using only the numbers between 10 and 99, you give a number for the horizontal position (the X value) and then a number for the vertical position (the Y value). The X and Y axes of your choice flash on and off leaving only the bullet hole to indicate the location of your shot.

Notes on the program

Lines 242 to 248 create the concentric circles of the target. The innermost circle is made separately with a radius of 10 (line 248). To make a genuinely different challenge every time, lines 909 to 937 randomly select the uppermost, leftmost, rightmost and bottom positions afresh for each new game. The centre position is derived mathematically from these figures (lines 938 to 939) and then the actual X and Y pixel addresses of these coordinates are shown in lines 1010 and 1020. Lines 938, 939, 1010 and 1020 are the arithmetic heart of the game. If you can follow these lines you should be a good programmer.

Changes you can make

You can draw a smaller or larger bull's-eye and alter the winning limits. You can attach legs to your target. To make the game harder, you can shorten or omit the display of the horizontal and vertical axes.

Program listing

```
100 REM INSTRUCTIONS
102 SCREEN1,1:CLS
105 COLOR1,15
110 FORF=1TO15
120 COLOR1,F
130 NEXT
140 PRINT"TARGET PRACTICE
142 PRINT"===============
146 PRINT"IN THIS GAME YOU SEE A TARGE
T.
147 PRINT:PRINT"YOUR TASK IS TO FIND T
HE BULLS-EYE.
148 PRINT
150 PRINT"YOU MUST ESTIMATE ITS POSITI
ON.
151 PRINT:BEEP2
152 PRINT"============================
===
153 PRINT
154 PRINT"FIRST TYPE IN THE HORIZONTAL
 POSITION.";
158 PRINT"THIS IS CALLED THE X-VALUE.
160 PRINT
162 PRINT"THEN TYPE IN THE VERTICAL PO
SITION.
164 PRINT"THIS IS CALLED THE Y-VALUE.
166 PRINT:BEEP
168 PRINT"============================
===
170 PRINT"EACH NUMBER IS BETWEEN 10 AN
D 99.
173 PRINT
174 PRINT"YOU HAVE 7 SHOTS WITH WHICH
178 PRINT"TO HIT THE BULLSEYE.
179 PRINT
```

```
180 PRINT"PRESS BAR TO CONTINUE..."
182 BEEP
184 IFINKEY$<>" "THEN184
190 BEEP
192 PATTERNC#64,"6060F8041C181800":REM
 ALPHA CHARACTER BECOMES DECORATION
194 PATTERNC#35,"C0C0000000000000":REM
 HASH  CHARACTER BECOMES BULLET HOLE
200 REM SCREEN
210 SCREEN2,2:CLS
220 COLOR1,15,(0,0)-(255,191),2
230 PRINT"@@@@@@@@@@"
232 PRINT"@ TARGET @"
233 PRINT"@@@@@@@@@@"
240 FORF=20TO80STEP20
242 CIRCLE(165,95),F,F/10,1,0,1
244 CIRCLE(165,95),F+1,F/10,1,0,1
246 NEXT
248 CIRCLE(165,95),10,1,1,0,1
260 GOTO600
400 REM INPUT DIRECTIONS
408 COLOR1
410 CURSOR10,30
414 PRINT"CHOOSE X-VALUE"
418 RETURN
420 COLOR15
422 CURSOR10,30
424 PRINT"CHOOSE X-VALUE"
428 RETURN
430 COLOR1
432 CURSOR10,40
434 PRINT"CHOOSE Y-VALUE"
438 RETURN
440 COLOR15
442 CURSOR10,40
444 PRINT"CHOOSE Y-VALUE"
448 RETURN
```

```
450 COLOR1
452 CURSOR10,160
454 PRINT"OUT OF RANGE
458 RETURN
460 COLOR15
462 CURSOR10,160
464 PRINT"OUT OF RANGE
468 RETURN
500 REM INPUT DATA
505 IP$=""
510 A$=INKEY$
514 IFA$<"0"ORA$>"9"THEN510
518 BEEP
519 FORG=1TO20:NEXT
520 IP$=IP$+A$
522 IFLEN(IP$)=1THEN510
524 BEEP
526 TE=VAL(IP$)
527 IFM$="Y"AND(TE<UPORTE>BP)THEN580
528 IFM$="X"AND(TE<LPORTE>RP)THEN580
530 RETURN
580 REM OUTSIDE RANGE
584 GOSUB450
586 SOUND1,110,15
588 FORF=1TO100:NEXT
589 SOUND0
590 GOSUB460
592 GOTO500
600 REM SEQUENCE CONTROL
604 GOSUB900
605 TY=0
607 TY=TY+1
608 IFTY=8THEN1100:REM END
610 GOSUB400:REM INVITE INPUT
615 M$="X"
620 GOSUB500:REM INPUT 2-DIGIT NUMBER
625 XG=VAL(IP$)
```

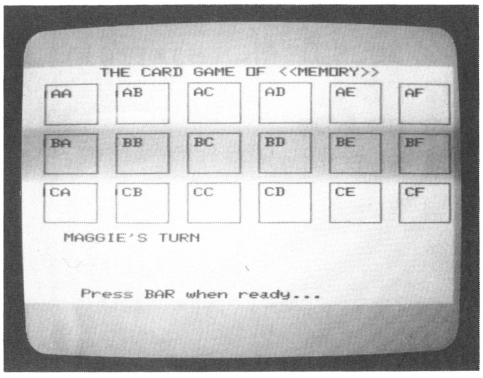
```
630 GOSUB720:REM DISPLAY INPUT
640 GOSUB420:REM ERASE X INVITE
642 GOSUB430:REM Y INVITE
643 M$="Y"
644 GOSUB500:REM INPUT GUESS
646 YG=VAL(IP$)
648 GOSUB739:REM DISPLAY Y-GUESS
650 GOSUB440:REM ERASE Y INVITE
654 GOSUB1000:REM DISPLAY DOT
656 GOSUB1300:REM CHECK FOR WIN
660 GOTO607:REM NEXT GO
700 REM DISPLAY INPUT NUMBERS
719 REM DISPLAY X-GUESS
720 CURSOR10,40+12*TY
722 COLOR1
724 PRINTIP$;","
728 RETURN
739 REM DISPLAY Y-GUESS NUMBER
740 CURSOR34,40+12*TY
742 COLOR1
744 PRINTIP$
748 RETURN
900 REM CHOOSE PARAMETERS
903 REM GET TRULY RANDOM K
905 K=VAL(RIGHT$(TIME$,1)):IFKTHEN910
906 GOTO905
909 REM CHOOSE RIGHTMOST POSITION
910 RP=INT(RND(K)*30)+70
913 REM CHOOSE LEFTMOST POSITION
914 LP=INT(RND(3)*40)
919 REM CHOOSE UPPERMOST POSITION
920 UP=INT(RND(7)*40)
923 REM CHOOSE BOTTOM POSITION
924 BP=INT(RND(7)*40)+60
937 REM DERIVE CENTER POSITION
938 CP=(RP-LP)/2+LP:REM X-VALUE LARGE
939 CQ=(BP-UP)/2+UP:REM Y-VALUE LARGE
```

```
940 REM CP,CQ ARE DESIRED ANSWERS
990 RETURN
1000 REM DISPLAY DOT
1005 COLOR1
1009 REM YA = ACTUAL Y-POSITION
1010 YA=INT((191*(YG-UP))/(BP-UP))
1019 REM XA = ACTUAL X-POSITION
1020 XA=INT(((XG-LP)/(RP-LP))*160+85)
1030 REM
1032 LINE(XA+1,0)-(XA+1,191),1
1033 LINE(80,YA+1)-(255,YA+1),1
1034 SOUND1,110,15:FORGF=1TO250:NEXT:S
OUND0
1035 BLINE(XA+1,0)-(XA+1,191)
1036 BLINE(80,YA+1)-(255,YA+1)
1037 CURSORXA,YA:PRINT"#":REM BULLET H
OLE
1038 SOUND3,5000,0
1039 FORG=15TO7STEP-1
1040 SOUND4,3,G
1044 NEXT
1046 SOUND0
1050 FORF=1TO200:NEXT
1080 RETURN
1100 REM END
1110 COLOR1
1120 CURSOR10,170
1124 PRINT"OUT OF SHOTS
1140 CURSOR10,180
1145 PRINT"BAR FOR MORE
1150 IFINKEY$<>" "THEN1150
1160 BEEP:BEEP:BEEP
1170 GOTO100
1300 REM CHECK FOR WIN
1310 IF XA<151THENRETURN
1312 IF XA>169THENRETURN
1314 IF YA>104THENRETURN
```

```
1316 IF YA<86THENRETURN
1320 CURSOR10,170
1324 PRINT"CONGRATULATIONS!
1330 FORF=1TO16
1340 SOUND1,RND(6)*100+110,15
1342 NEXT
1350 SOUND0
1360 GOTO1140
```

CONCENTRATION

This is a traditional card game which tests the memory of the players. Eighteen cards are displayed labelled AA to CE. You pick two cards and they are 'turned over' to reveal the number and the suit. If you pick a pair you score. After a while you should remember the positions of several cards. The player with most pairs wins.



How to play

The program dictates whose turn it is. When ready, you press BAR, then type in your card labels on the text screen. The cards you choose will be flipped. Good luck.

Notes on the program

Line 55 sets up *arrays* for what's in the cards and for remembering which cards have been used. The key to the program is the card shuffling routine in lines 400 to 499. The pack consists of eighteen cards numbered 2 to Jack in Diamonds and Clubs. Lines 440 to 460 decide the location for each card. The subroutine from 500 to 532 decides the turn by the simple device of subtracting from 3 (in line 510). Lines 520 to 532 are required because we can't erase on the graphics screen in a simple way.

Changes you can make

The present program creates nine different pairs in the pack. To make pair finding easier, you could include some triples or quadruples, or to make the whole game more challenging, you could allow three card flips and score only for triple finds! Another change you may desire is to allow a second turn after a player gets a pair. Some readers may like to adapt the program so that a player may choose and flip the first card before choosing the second one.

If you want to, you can put the input part of the program onto the graphics screen, rather than the text screen. You can do this if you can set aside a separate place for every possible input.

Program listing

```
50 REM ARRAY
55 DIMCD$(6,3),U(6,3),D(6,3)
60 SC(1)=0:SC(2)=0
62 FORF=1TO3:FORG=1TO6
64 U(G,F)=0:D(G,F)=0
66 NEXT:NEXT
100 REM GET PLAYER'S NAMES
105 COLOR1,11
110 CLS
120 CURSOR3,3:PRINT"THIS IS A GAME FOR
 TWO PLAYERS"
130 PRINT
140 PRINT"     TYPE IN YOUR NAMES:"
150 PRINT"     ==================="
160 PRINT
170 FORF=1TO2
175 A$="PLAYER"+STR$(F)
177 PRINT
180 PRINT"NAME OF ";A$;":"
183 BEEP2
185 INPUTPL$(F)
190 NEXT
195 CURSOR3,20
196 PRINT"   SEGA NOW SHUFFLING CARDS"

198 GOSUB400
200 REM SCREEN
201 COLOR1,15
210 SCREEN2,2:CLS
220 COLOR1,15,(0,0)-(255,191),14
250 REM WHOSE 1ST TURN
```

```
260 TN=INT(RND(7)*2)+1
300 REM DRAW CARDS
305 CURSOR50,2
307 PRINT"THE CARD GAME OF <<MEMORY>>"

310 FORF=0TO5
320 FORG=0TO2
330 X=F*41+18:Y=G*40+14
335 LINE(X,Y)-(X+30,Y+32),12,B
340 COLOR6
345 CURSORX+4,Y+4
347 PRINTCHR$(65+G);CHR$(65+F)
350 NEXT
360 NEXT
399 GOTO500
400 REM SHUFFLE CARDS
410 FORN=2TO10
420 FORSU=1TO2
422 IFSU=1THENSU$=CHR$(245)
424 IFSU=2THENSU$=CHR$(247)
426 NU$=RIGHT$(STR$(N),1)
428 IFN=10THENNU$="J"
430 CA$=NU$+SU$
440 X=INT(RND(4)*6)+1
450 Y=INT(RND(5)*3)+1
452 SOUND1,300+100*X*Y,15
455 IFU(X,Y)THEN440
460 CD$(X,Y)=CA$
470 U(X,Y)=1
480 NEXT:NEXT
490 SOUND0
499 RETURN
500 REM WHOSE TURN
510 TN=3-TN
515 COLOR1
517 ONTNGOTO520,530
520 CURSOR30,135:PRINTPL$(1);"'S TURN"
```

```
521 COLOR15
522 CURSOR30,145:PRINTPL$(2);"'S TURN"

525 GOTO534
530 CURSOR30,145:PRINTPL$(2);"'S TURN"

531 COLOR15
532 CURSOR30,135:PRINTPL$(1);"'S TURN"

534 FORF=1TO8
536 SOUND1,200+RND(8)*500,15
537 FORG=1TO10:NEXT
538 NEXT:GOSUB1000
539 SOUND0
540 SCREEN1
541 PR=0
542 CLS
544 COLOR1,7
545 CURSOR3,3
546 PRINTPL$(TN);"'S TURN"
547 PRINT"============================
="
548 PRINT:PRINT
549 PRINT"NAME YOUR FIRST CARD"
550 INPUTC1$
551 T=LEN(C1$)
552 IFT<>2THEN540
553 PRINT:PRINT
554 PRINT"NAME YOUR SECOND CARD"
560 INPUTC2$
561 T=LEN(C2$)
562 IFT<>2THEN540
570 T$=LEFT$(C1$,1)
571 Y1=ASC(T$)-64
572 T=ASC(T$):IFT<65ORT>67THEN540
574 T$=RIGHT$(C1$,1)
575 X1=ASC(T$)-64
```

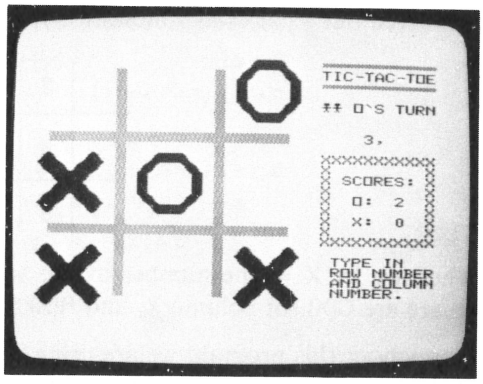
```
576 T=ASC(T$):IFT<65ORT>70THEN540
580 T$=LEFT$(C2$,1)
581 Y2=ASC(T$)-64
582 T=ASC(T$):IFT<65ORT>67THEN540
584 T$=RIGHT$(C2$,1)
585 X2=ASC(T$)-64
586 T=ASC(T$):IFT<65ORT>70THEN540
590 IFC2$=C1$THEN540
600 REM DISPLAY CARDS
602 SCREEN2,2
605 COLOR1
610 X=(X1-1)*41+26:Y=(Y1-1)*40+34
620 CURSORX,Y
630 PRINTCD$(X1,Y1)
640 X=(X2-1)*41+26:Y=(Y2-1)*40+34
650 CURSORX,Y
660 PRINTCD$(X2,Y2)
674 FORF=1TO32
676 SOUND1,110+RND(8)*100,10
677 FORG=1TO20:NEXT
678 NEXT
679 SOUND0
700 REM CHECK SCORE
710 IFVAL(CD$(X1,Y1))<>VAL(CD$(X2,Y2))
THEN800
720 IFD(X1,Y1)ORD(X2,Y2)THEN800
730 D(X1,Y1)=1
740 D(X2,Y2)=1
745 COLOR1
750 CURSOR24,160:PRINT"YES A PAIR!"
760 SC(TN)=SC(TN)+1
770 FORF=1TO8
772 SOUND1,1000,15
774 SOUND2,1500,15
776 FORH=1TO50:NEXT
777 SOUND0:FORH=1TO10:NEXT
778 NEXT
```

```
785 COLOR15
790 CURSOR24,160:PRINT"YES A PAIR!"
791 IFSC(2)THENCOLOR6:FORI=1TOSC(2):CU
RSOR200+4*I,145:PRINTCHR$(224):NEXT
792 IFSC(1)THENCOLOR4:FORI=1TOSC(1):CU
RSOR200+4*I,135:PRINTCHR$(224):NEXT
793 IFSC(1)+SC(2)<9THEN500
796 IFSC(1)>SC(2)THENWP$=PL$(1)
797 IFSC(2)>SC(1)THENWP$=PL$(2)
798 COLOR1
799 CURSOR50,170:PRINT"GAME OVER:";WP$
;" WINS":GOTO900
800 REM ERASE DISPLAYS
801 COLOR15
802 IFD(X1,Y1)THEN835
810 X=(X1-1)*41+26:Y=(Y1-1)*40+34
820 CURSORX,Y
830 PRINTCD$(X1,Y1)
835 IFD(X2,Y2)THEN890
840 X=(X2-1)*41+26:Y=(Y2-1)*40+34
850 CURSORX,Y
860 PRINTCD$(X2,Y2)
890 GOTO 500
900 REM RESTART
910 FORF=1TO300:NEXT
915 COLOR4
920 CURSOR50,180
925 PRINT"PRESS BAR FOR ANOTHER GAME"
930 IFINKEY$<>" "THEN930
935 SCREEN1:CLS
940 GOTO60
1000 REM BAR PRESS
1010 COLOR13
1020 CURSOR40,180
1022 SOUND0
1025 PRINT"Press BAR when ready...
1030 IFINKEY$<>" "THEN1030
```

```
1034 FORG=1TO4
1035 SOUND1,110,15
1037 FORF=1TO20:NEXT
1038 SOUND0
1039 NEXT
1040 RETURN
```

TIC-TAC-TOE

This game is also called *Noughts and Crosses*. Most people know how to play it with pen and paper, but with this program there is an extra dimension of thinking required. Each player must name the *row* and *column* in which they want to place their O or X. It's very good practice for higher maths. The first player to get a full row, column or diagonal of their symbol wins.



How to play

The row and column values for each square are as follows:

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

You must type in one of these pairs of numbers when it is your turn. The score is automatically displayed. If you are clever enough, you can learn to play so that you never lose.

Notes on the program

Line 12 introduces abbreviations for several block characters which will be used to create the very large *nought* and the very large *cross*. Line 13 gives us T$ and U$, parts of the large noughts and crosses. If you care to look up the character code list in your manual, you can greatly simplify lines 24 and 28 by using keyboard characters within quotes.

The presence of each cross is stored as a 5 and the presence of each nought is stored as a 2. This is how (in line 620) we can tell if a win has occurred. That is, if T or U (the sum of three symbols in a row) equals 6 or 15. Lines 630 to 636

take care of the diagonal case. A draw is recognised simply because there has been no win, and the number of turns equals 9. See line 705.

In order to know where to display our big noughts and crosses we have numbered our squares as follows:

1	2	3
4	5	6
7	8	9

When we let X = the number of the square, the cursor positions for each square are C(X) for column X, and R(X) for row X. See lines 52 and 54.

Throughout this program we are using only the text screen — there being no need for high resolution graphics. Because we've stuck to the text screen we can use CURSOR LEFT and CURSOR DOWN commands such as CHR$(31), etc. See lines 22, 24 and 28.

Changes you can make

You could try to get the game onto the graphics screen. For this you need a different location for each message. Remember that you can't *erase* text on the graphics screen — you can only white over it. A grand enhancement would be to use much smaller symbols and create three-dimensional *Noughts and Crosses:* You play imagining grid one is on top of grid two, and grid two is on top of grid three!

Program listing

```
10 REM TIC TAC TOE
11 REM GRAPHIC ABBREVS
12 B$=CHR$(229):F$=CHR$(152):L$=CHR$(1
50):S$=CHR$(151):A$=CHR$(149):REM BLOC
K,TRIANGLES
13 T$=A$+B$+B$+L$:U$=S$+B$+B$+F$:REM G
RAPHIC GROUPS
14 CH$=CHR$(144):REM CHECKED BLOCK
15 UL$="":FORF=1TO11:UL$=UL$+CHR$(233)
:NEXT:REM UNDERLINE
```

```
16 ME$=CHR$(253)+CHR$(253):REM 2 MEN
20 S$(2)="O":S$(5)="X"
22 CR$=CHR$(31)+CHR$(29)+CHR$(29)+CHR$
(29)+CHR$(29)+CHR$(29)+CHR$(29)
24 L$(2)=" "+T$+" "+CR$+A$+F$+"  "+S$+
L$+CR$+B$+"    "+B$+CR$+B$+"    "+B$+C
R$+S$+L$+"  "+A$+F$+CR$+" "+U$+CHR$(30
)
28 L$(5)=A$+L$+"  "+A$+L$+CR$+S$+B$+L$
+A$+B$+F$+CR$+" "+U$+" "+CR$+" "+T$+"
"+CR$+A$+B$+F$+S$+B$+L$+CR$+S$+F$+"  "
+S$+F$+CHR$(30)
50 REM CURSOR POSITIONS
52 FORF=1TO9:X=INT((F-1)/3):R(F)=X*9
54 Y=F-X*3:C(F)=(Y-1)*9:NEXT
60 REM SQUARE NUMBERS
62 FOR R=1TO3:FOR C=1TO3:N(R,C)=R+3*(C
-1):NEXT :NEXT
100 REM PLAYERS CHOOSE SYMBOLS
105 FORF=1TO5:BEEP:NEXT
110 CLS
120 CURSOR5,5:PRINT"A GAME FOR 2 PLAYE
RS"
130 CURSOR5,15:PRINT"DECIDE WHO SHALL
BE
140 CURSOR5,17: PRINT"X AND WHO SHALL
BE O.
150 CURSOR5,20:PRINT"THEN PRESS BAR
160 A$=INKEY$:IFA$<>" "THEN160
170 BEEP2
180 X=VAL(RIGHT$(TIME$,1)):PRINTX
182 TN=5:IFX/2=INT(X/2)THENTN=2
200 REM DISPLAY GRID
202 CLS:COLOR1,10
205 FORG=1TO22
210 FORF=7TO16STEP9
212 CURSORF,G:PRINTCH$
```

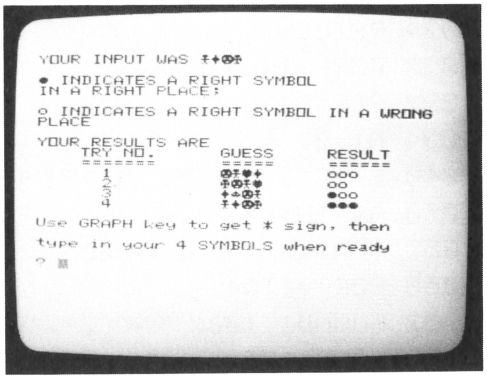
```
220 CURSORG,F:PRINTCH$:NEXT :NEXT
230 CURSOR26,0:PRINTUL$
231 CURSOR26,2:PRINTUL$
232 CURSOR26,1:PRINT"TIC-TAC-TOE"
240 CURSOR26,4:PRINTME$;"  'S TURN"
242 CURSOR26,9:PRINT"XXXXXXXXXXX"
244 CURSOR26,17:PRINT"XXXXXXXXXXX"
246 FORF=10TO16:FORG=26TO36STEP10
248 CURSORG,F:PRINT"X":NEXT:NEXT
250 CURSOR29,13:PRINT"O:"
252 CURSOR29,15:PRINT"X:"
254 CURSOR28,11:PRINT"SCORES:
256 GOSUB1020
260 CURSOR27,19:PRINT"TYPE IN"
262 CURSOR27,20:PRINT"ROW NUMBER"
264 CURSOR27,21:PRINT"AND COLUMN"
266 CURSOR27,22:PRINT"NUMBER."
300 REM INDICATE WHOSE TURN
305 CURSOR27,6:PRINT"          "
306 CURSOR27,7:PRINT"          "
310 TN=7-TN:REM CHANGE TURN
320 CURSOR29,4:PRINTS$(TN)
330 FORF=1TO4
332 SOUND1,2000+RND(7)*2000,10
334 NEXT:SOUND0
400 REM INPUT TURN
405 NT=NT+1:REM NO. OF TURNS
410 A$=INKEY$
420 IFA$<"1"ORA$>"3"THEN410
424 R=VAL(A$):CURSOR29,7
426 FORF=1TO10:SOUND2,300+RND(6)*300,7
:NEXT
427 PRINTR;",":SOUND0
428 A$=INKEY$
430 IFA$<"1"ORA$>"3"THEN428
432 C=VAL(A$):CURSOR32,7
433 FORF=1TO10:SOUND2,300+RND(6)*300,7
```

```
:NEXT
434 PRINTC:SOUND0
436 IFP(R,C)=0THEN450:REM SPACE EMPTY
438 FORG=1TO4
440 CURSOR29,6:PRINT"ILLEGAL"
441 SOUND2,150,15
442 FORF=1TO10:NEXT
444 CURSOR29,6:PRINT"       "
445 SOUND0
446 FORF=1TO10:NEXT
448 NEXT : CURSOR29,7: PRINT"       "
GOTO410
450 P(R,C)=TN
500 REM DISPLAY O OR X
504 Q=N(R,C)
508 CURSORR(Q),C(Q):PRINTL$(TN)
510 FORF=1TO99:NEXT
512 CURSOR29,7:PRINT"       ":REM ERASE

600 REM CHECK FOR WIN
602 WN=0
605 CURSOR27,6:PRINT"CHECKING"
606 CURSOR27,7:PRINT"  NOW   "
610 FORR=1TO3
612 T=0:U=0
614 FORC=1TO3
616 T=T+P(R,C)
618 U=U+P(C,R)
620 IFT=60RT=150RU=60RU=15THENR=3:C=3:
WN=1
622 NEXT :NEXT
624 IFWNTHEN800
628 T=0:U=0
630 FORF=1TO3
632 T=T+P(F,F)
634 U=U+P(4-F,F)
636 NEXT:IFT=60RT=150RU=60RU=15THENWN=
```

```
1:GOTO800
700 REM CHECK FOR DRAW
702 DW=0
705 IFNT=9THEN900
710 GOTO300
800 REM WIN ROUTINE
805 CURSOR27,6:PRINT"          "
806 CURSOR27,7:PRINT"          "
810 FORG=1TO8
814 CURSOR29,6:PRINTS$(TN)+" WINS"
818 SOUND2,1500,15:SOUND1,2500,15
822 FORF=1TO10:NEXT
826 CURSOR29,6:PRINTS$(TN)+" wins"
830 SOUND0
834 FORF=1TO10:NEXT
838 NEXT:CURSOR29,7:PRINT"          "
860 GOTO1000
900 REM DRAW ROUTINE
905 CURSOR27,6:PRINT"          "
906 CURSOR27,7:PRINT"          "
910 FORG=1TO8
914 CURSOR29,6:PRINT"  DRAW"
918 SOUND2,500,15:SOUND1,750,15
922 FORF=1TO10:NEXT
926 CURSOR29,6:PRINT"  draw"
930 SOUND0
934 FORF=1TO10:NEXT
938 NEXT:CURSOR29,7:PRINT"          "
940 DW=1
960 GOTO1000
1000 REM UPDATE SCORE
1005 IFWNTHENSC(TN)=SC(TN)+2
1007 IFDWTHENSC(2)=SC(2)+1:SC(5)=SC(5)
+1
1010 CURSOR27,19:PRINT"PRESS   "
1012 CURSOR27,20:PRINT"BAR FOR    "
1014 CURSOR27,21:PRINT"NEXT GRID "
```

```
1016 CURSOR27,22:PRINT"          "
1018 GOSUB1020:GOTO1030
1020 CURSOR32,13:PRINTSC(2)
1022 CURSOR32,15:PRINTSC(5)
1024 RETURN
1030 FORF=1TO3:FORG=1TO3
1032 P(F,G)=0:NEXT:NEXT
1034 NT=0
1040 IFSC(2)>8ORSC(5)>8THEN1100
1050 A$=INKEY$
1055 IFA$<>" "THEN1050
1060 FORF=1TO4
1062 SOUND1,1000+RND(7)*1000,10
1063 FORG=1TO70:NEXT
1064 NEXT:SOUND0
1080 GOTO200
1100 REM NEXT GAME?
1105 NT=0
1110 CLS:COLOR1,11
1120 CURSOR15,5:PRINT"GAME OVER"
1125 CURSOR15,10:PRINT"O SCORE:";SC(2)

1130 CURSOR15,12:PRINT"X SCORE:";SC(5)

1140 CURSOR5,18:PRINT"IF YOU DESIRE AN
OTHER GAME"
1150 SC(2)=0:SC(5)=0:GOTO150
```

MASTERMIND



This game is the classic form of *Mastermind*. One player inputs four symbols and the other tries to match them. The computer indicates after each try how many symbols were correct and how many in their correct position. The logical thinker will deduce the correct combination in fewer tries.

How to play

Press the GRAPHIC key and select four different symbols from the seven displayed and type them in. If you type an illegal set, you get a BLUE and are allowed another turn. A black dot indicates a correct place and a white dot means right symbol, wrong place. All previous attempts and their indicators are displayed.

Notes on the program

Line 50 sets aside one hundred spaces in the S$ array and the R$ array for a possible hundred tries. We hope that nobody will require that many! Line 55 holds the pattern for the checkered underline. For line 126 you can simply print the characters as listed in the REM statement. Lines 430 to 450 check every symbol in the answer given against the original string of symbols. This is to calculate how many hollow symbols we should award. This is going to include, of course, symbols which are in exactly the right place, so we shall have to subtract the correct place count from the correct symbol count. This is done in lines 462 to 466.

Changes you can make

If this game is too easy, you can make it tougher by:

- increasing the range of usable symbols, and/or
- increasing the length of the symbol set.

And if you want to, you can create your own symbols (using PATTERN C#).

Program listing

```
40 CLS
42 PRINT"PREPARING GAME"
50 DIMTS$(100),RS$(100):REM UP TO TR (
TRIES)
55 PATTERNC#64,"0000102844800000"
60 TR=1
70 BEEP2
90 FORF=1TO100
91 TS$(F)="":RS$(F)=""
92 NEXT
100 REM INPUT NAMES
110 CLS
120 COLOR1,15
122 PRINT"MASTER MIND"
123 PRINT"for 2 players
124 PRINT"using these symbols:
125 PRINT
126 PRINTCHR$(245);" ";CHR$(246);" ";C
HR$(247);" ";CHR$(248);" ";CHR$(249);"
 ";CHR$(253);" ";CHR$(250);" (7 symbol
s)":REM SPADE,HEART,DIAMOND,CLUB,FACE,
MAN,UFO
130 PRINT:PRINT:PRINT
140 FORF=1TO2
141 PRINT
142 PRINT"PLAYER";F;" NAME:"
143 BEEP
144 INPUT PL$(F)
146 NEXT
150 REM 1ST TURN
155 TN=INT(RND(8)*2)+1
```

```
200 REM DECIDE WHOSE TURN
210 TN=3-TN
220 PRINT
221 CLS:BEEP2
222 PRINTPL$(TN);"'S TURN
224 PRINT"TO CHOOSE A SET OF 4 SYMBOLS

226 PRINT"ALL DIFFERENT"
230 PRINT
232 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"

234 PRINTPL$(3-TN);", DON'T LOOK!"
235 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"

236 GOTO238
237 GOSUB800
238 PRINT
239 PRINT"Use GRAPH key to get * sign,
 then
240 PRINT
241 PRINT"type in your SYMBOL SET when
 ready
242 PRINT
243 INPUTSS$
244 IFLEN(SS$)<>4THEN237
246 T$=SS$:GOSUB900:IFOK=0THEN237
247 CLS:BEEP2
248 FORF=1TO4
250 S$(F)=MID$(SS$,F,1)
258 NEXT
300 REM INPUT TRIES
310 PRINT"NOW, ";PL$(3-TN);", GUESS WH
AT SET"
314 PRINT"OF 4 SYMBOLS ";PL$(TN)
315 PRINT"HAS TYPED IN!
316 GOTO319
317 GOSUB800
```

```
319 PRINT
320 PRINT"Use GRAPH key to get * sign,
 then
322 PRINT
324 PRINT"type in your 4 SYMBOLS when
ready
326 PRINT
330 INPUTTS$(TR):REM TR=TRY NUMBER
344 IFLEN(TS$(TR))<>4THEN317
346 T$=TS$(TR):GOSUB900:IFOK=0THEN317
347 CLS:GOSUB850
348 FORF=1TO4
350 U$(F)=MID$(T$,F,1):REM U$=TRIED SY
MBOLS
358 NEXT
400 REM ASSESS INPUT
405 REM PRODUCE RS$(TR)
406 R$="":CP=0:CS=0
410 FORII=1TO4
415 IFU$(II)=S$(II)THENR$=R$+CHR$(236)
:CP=CP+1
420 NEXT
430 FORJJ=1TO4
434 FORKK=1TO4
442 IFS$(JJ)=U$(KK)THENCS=CS+1:R$=R$+C
HR$(235)
450 NEXT:NEXT
460 IFCP=0THEN470
462 FORF=1TOCP
464 CS=CS-1:R$=LEFT$(R$,LEN(R$)-1)
466 NEXT
470 RS$(TR)=R$
505 CLS
510 PRINT"YOUR INPUT WAS ";TS$(TR)
520 PRINT
530 PRINTCHR$(236);" INDICATES A RIGHT
 SYMBOL
```

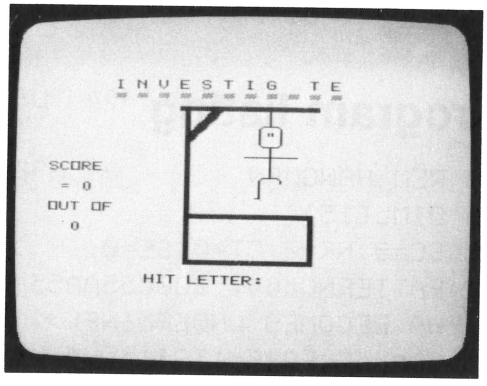
```
531 PRINT"IN A RIGHT PLACE;
532 PRINT
533 PRINTCHR$(235);" INDICATES A RIGHT
 SYMBOL IN A WRONG PLACE"
536 PRINT
540 PRINT"YOUR RESULTS ARE
542 PRINT"    TRY NO.      GUESS     R
ESULT"
543 PRINT"    =======      =====     =
=====""
544 FORFF=1TOTR
546 PRINTTAB(5);FF;TAB(17);TS$(FF);TAB
(27);RS$(FF)
550 NEXT
552 IFTS$(TR)=SS$THEN700
560 PRINT
562 PRINT"Press BAR when ready for nex
t try"
563 IFINKEY$<>" "THEN563
564 BEEP2
566 FORFH=1TO60:NEXT
570 PRINTCHR$(30);CHR$(30);CHR$(30)
590 TR=TR+1
599 GOTO319
700 REM WIN
710 FORTT=1TO15
720 SOUND1,RND(7)*1300+200,15
722 FORTG=1TO25:NEXT
723 COLOR1,TT
724 NEXT
740 PRINT
742 PRINT"CONGRATULATIONS, ";PL$(3-TN)

743 PRINT
744 PRINT"YOU MADE IT IN";TR;" TRIES!"

750 PRINT
```

```
751 SOUND0
752 PRINT"Press BAR for new game"
754 IFINKEY$<>" "THEN754
760 GOTO200
800 REM ODD GOSUBS
810 REM DUD TURN
820 COLOR1,4
830 SOUND1,120,15
832 FORGG=1TO59
834 NEXT
836 SOUND0
838 COLOR1,15
840 RETURN
850 REM NEXT TURN
852 COLOR1,11
853 SOUND1,620,15
854 FORGG=1TO59
855 NEXT
856 SOUND0
857 COLOR1,15
860 RETURN
900 REM TEST 4-SET T$ FOR VALID SYMBOL

902 ST=0
905 OK=1
910 FORFF=1TO4
920 TL$=MID$(T$,FF,1)
922 GOSUB960
924 IFSY=0THENOK=0
940 NEXT
942 IFOKTHENS(FF)=SY
944 RETURN
960 REM TEST SINGLE SYMBOL
965 SY=0
970 IFTL$=CHR$(245)THENSY=2:GOTO990
972 IFTL$=CHR$(246)THENSY=4:GOTO990
974 IFTL$=CHR$(247)THENSY=8:GOTO990
```

```
976 IFTL$=CHR$(248)THENSY=16:GOTO990
987 IFTL$=CHR$(249)THENSY=32:GOTO990
988 IFTL$=CHR$(253)THENSY=64:GOTO990
989 IFTL$=CHR$(250)THENSY=128:GOTO990
990 IFSYANDSTTHENOK=0
992 ST=ST+SY
1020 RETURN
```

HANGMAN

This game conforms to the classic style of *Hangman*. One player types in a word and the other guesses the letters. Every error adds another piece to the scaffold. Too many errors and you're hanged. It can make spelling fun. A parent could insert a child's current problem words and thus help remedy spelling difficulties.



How to play

One player types in five words and the other tries to guess the correct letters for a score out of five. Then the players swap roles. If a letter occurs more than once in a word, one correct guess will display it in all its positions.

Notes on the program

Lines 23 and 24 form strings of graphics for use as parts of the scaffold. Line 25 abbreviates the triangle characters and line 26 gives us vertical line, horizontal line and cross-over characters. You can of course translate these into 'direct' keyboard character symbols for your own copy of the program. Lines 109 and 112 contain a row of the symbol *alpha*. Line 22 converts this to a graphic character, namely an underline.

Lines 30—36 abbreviate cumbersome cursor controls for use later in building the scaffolding.

Line 441 checks that the letter is actually in the word. Line 442 then prints it in its correct position, while keeping a tally of the number of letters correctly guessed. This count is used in line 445 to recognise when the spelling is completed.

To cause the word RIGHT or the word HUNG to blink on and off, we print spaces over them as in line 655.

Changes you can make

You can create more steps in the building of the scaffold — or fewer. You could put the whole game onto the graphics screen — in color — using three or more sprites to represent the hanged one.

Program listing

```
10 REM HANGMAN
15 DIML(15)
20 SC=0:NR=0:CT=0:SE=0
22 PATTERNC#64,"000055AA55AA0000":REM
ALPHA BECOMES UNDERLINE
23 LL$="":FORF=1TO11:LL$=LL$+CHR$(155)
:NEXT:REM LOWER LINE
24 UU$="":FORF=1TO11:UU$=UU$+CHR$(156)
:NEXT:REM UPPER LINE
25 FT$=CHR$(152):AT$=CHR$(149):REM TRI
ANGLE CHARACTERS
26 VL$=CHR$(128):HL$=CHR$(129):CC$=CHR
$(146):REM LINE CHARACTERS
30 D$=CHR$(29)+CHR$(31):REM DOWN
32 L$=CHR$(29)+CHR$(29):REM LEFT
33 DL$=D$+L$:REM LEFT & DOWN
34 UR$="":FORF=1TO5:UR$=UR$+CHR$(226)+
D$:NEXT:REM UP RIGHT
36 UL$="":FORF=1TO5:UL$=UL$+CHR$(225)+
D$:NEXT:REM UP LEFT
100 REM INPUT WORDS
105 CLS
107 COLOR1,11
109 CURSOR3,2:PRINT"@@@@@@@@@@@@@@@@@@@
@"
110 CURSOR3,3:PRINT"NOW TYPE IN 5 WORD
S"
112 CURSOR3,4:PRINT"@@@@@@@@@@@@@@@@@@@
@"
```

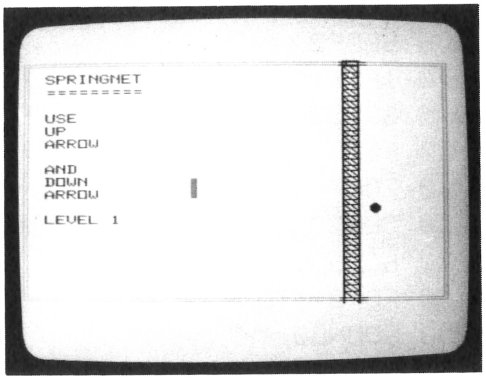
```
115 PRINT:PRINT"NOT LONGER THAN 15 LET
TERS
120 PRINT:PRINT"(OTHER PLAYER MUST NOT
 LOOK)
125 PRINT:PRINT
130 FORF=1TO5
140 PRINT"WORD";F;:    INPUTW$(F):PRINT
CHR$(13):NEXT
200 REM DISPLAY SCORE & SPACES
210 CLS
212 LT=0:REM LETTERS GOT
213 NR=NR+1
214 FORF=1TOLEN(W$(NR)):L(F)=1:NEXT
220 COLOR1,14
230 CURSOR1,10:PRINT"SCORE"
235 CURSOR1,12:PRINT" =";SE
237 CURSOR1,14:PRINT"OUT OF
238 CURSOR1,16:PRINT" ";NR-1
250 FORF=1TOLEN(W$(NR))
252 CURSOR5+2*F,3:PRINT"@"
254 NEXT
290 SC=0
300 REM INPUT ROUTINE
310 CURSOR10,21:PRINT"HIT LETTER:";
315 SOUND2,1000,10
316 FORG=1TO10:NEXT
317 SOUND0
320 A$=INKEY$
330 IFA$=""THEN320
332 CURSOR10,21:PRINT"              ";
335 OK=0:REM WRONG LETTER FLAG
400 REM CHECK LETTER
440 FORF=1TOLEN(W$(NR))
441 IFA$=MID$(W$(NR),F,1) THEN  BEEP:
OK=1
442 IFA$=MID$(W$(NR),F,1) AND L(F)THEN
CURSOR 5+2*F,2: PRINTA$: BEEP2:LT=LT+1
```

```
:L(F)=0:REM USED
444 NEXT
445 IFLT=LEN(W$(NR))THEN600
446 IFOKTHEN300
500 REM SCAFFOLD
505 SC=SC+1
511 IFSC=1THENCURSOR14,19:PRINTLL$
512 IFSC=2THENCURSOR13,15:PRINTUR$:CUR
SOR25,15:PRINTUL$
513 IFSC=3THENCURSOR14,15:PRINTUU$
514 IFSC=4THENCURSOR13,5:PRINTUR$;UR$
515 IFSC=5THENCURSOR16,5:PRINTFT$+ L$+
AT$+D$+FT$+L$+AT$+D$+FT$
516 IFSC=6THENCURSOR13,4:PRINTLL$;CHR$
(155);CHR$(155)
517 IFSC=7THENCURSOR21,5:PRINTUL$+D$+
L$+" ";CHR$(138);CHR$(130);CHR$(140)+D
L$+UL$+CHR$(34)+UL$+DL$+CHR$(139);CHR$
(131)+CHR$(141)
518 IFSC=8THENCURSOR19,9:PRINTHL$;HL$;
CC$;HL$;HL$
519 IFSC=9THENCURSOR21,10:PRINTUL$+DL$
;" ";CHR$(138);CHR$(130);CHR$(140)
520 IFSC=10THENCURSOR20,12:PRINTUL$+D$
+CHR$(137)
521 IFSC=11THENCURSOR22,12:PRINTUL$+D$
+CHR$(135):GOTO700
525 SOUND1,200,15:FORG=1TO30:NEXT :
SOUND0
530 GOTO310:REM NEXT LETTER
600 REM COMPLETION
605 SE=SE+1
610 FORF=1TO6
620 CURSOR31,15:PRINT"RIGHT!
630 SOUND2,3120,15
640 FORG=1TO10:NEXT
650 SOUND0
```

```
655 CURSOR31,15:PRINT"        "
657 FORG=1TO20:NEXT
660 NEXT
666 CURSOR10,23:PRINT"BAR FOR NEXT";
667 IFINKEY$<>" "THEN667
668 BEEP2
670 IFNR=5THEN800
675 GOTO200
700 REM  HUNG ROUTINE
710 FORF=1TO3
720 CURSOR32,15:PRINT"HUNG!"
730 SOUND2,120,15
740 FORG=1TO50:NEXT
750 SOUND0
755 CURSOR32,15:PRINT"      "
757 FORG=1TO20:NEXT
760 NEXT
765 CURSOR10,21:      PRINT"word was ";
W$(NR)
766 CURSOR10,23:PRINT"BAR FOR NEXT";
767 IFINKEY$<>" "THEN767
768 BEEP2
769 IFNR=5THEN800
770 GOTO200
800 REM NEW GAME
805 COLOR1,11
810 CLS
815 CURSOR3,3:PRINT"GAME OVER!
817 CURSOR3,6:PRINT"SCORE =";SE;" OUT
OF 5
820 CURSOR3,15:PRINT"WANT A NEW GAME?"

830 CURSOR3,18:PRINT"IF SO, PRESS BAR"

840 IFINKEY$<>" "THEN840
850 BEEP2
860 GOTO 20
```

SPRINGNET

Here is an exciting real-time game to test your reflexes. It is like solo squash but with the extra complication of a special net stretched across the court. This net *sometimes* catches and returns the ball. This gives you less time to react. Despite appearances, it *is* possible to hit the ball at CHAMPION level. Hint: you need every millimetre of bat.



How to play

Use up and down arrows to control your bat. Sometimes it is wise to use the upper half of your bat and sometimes the lower half. You have three balls in all. Your score equals the number of times you successfully strike the ball.

Notes on the program

Depending on the level of difficulty chosen by the user in line 17, a *chance* level is determined in line 80. This will decide how often the ball will hit the net. Lines 151 to 156 form the bat, which is a quadruple sprite although we are only using the two left edges of it. Line 158 makes our ball. Lines 420 and 424 assign new values to the ball position based on the old values plus the X distance and the Y distance. Both of these are set at ten in line 110. You can change this if you wish to make a faster or slower game. Line 487 decides whether the ball hits the net or not. If RND(5) (which is always a fraction less than 1) is greater than the chance level, CH, then the ball will miss the net. Otherwise we reverse the X direction (line 488). Lines 900 to 950 weave the intriguing net.

Changes you can make

You can use smaller steps to make a slower game or larger steps to make a faster one. If you make such alterations you may need to adjust the bat

column, net column or wall column so that the ball just strikes. A more dramatic alteration is to shift the net to centre and to introduce a second bat — on the right-hand side — to create a version of *tennis*.

Program listing

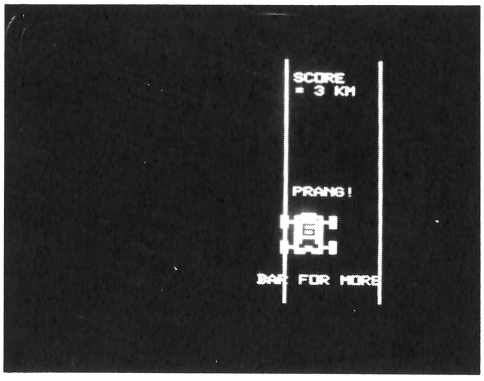
```
40 REM SPRINGNET
42 REM MADE BY PHILIP OCARROLL
50 REM DEGREE OF DIFFICULTY
51 COLOR1,15
52 SCREEN1,1:CLS
53 FORF=2TO15:COLOR1,F:NEXT
54 PRINT"SPRINGNET"
55 PRINT"=========
56 PRINT"THIS IS A NEW BALL GAME FROM
MARS      "
57 PRINT"IN WHICH THERE IS A RANDOMLY
ACTIVATED
58 PRINT"SPRINGNET WHICH SOMETIMES RET
URNS YOUR
59 PRINT"BALLS, MAKING FOR SOME DIFFIC
ULT SHOTS
60 PRIN
61 BEEP2
62 PRINT"SELECT LEVEL
64 PRINT"------------
66 PRINT:PRINT"1. BEGINNER
67 PRINT:PRINT"2. JUNIOR
68 PRINT:PRINT"3. SENIOR
69 PRINT:PRINT"4. CHAMPION
70 PRINT:PRINT"PRESS 1-4
72 A$=INKEY$
74 A=VAL(A$):LV=A
76 IFA<1ORA>4THEN72
78 BEEP
80 CH=((A-1)*3)/10
100 REM INITIAL POSITIONS OF BALL
110 BX=90:BY=0:YD=10:XD=10:AY=10:AX=10
```

```
120 UL=10:LL=170:RL=230
130 SC=3:REM PERMISSIBLE BALLS
150 REM MAKE BAT
151 PATTERNS#0,"F0F0F0F0F0F0F0F0"
152 PATTERNS#1,"F0F0F0F0F0F0F0F0"
154 PATTERNS#2,"0000000000000000"
156 PATTERNS#3,"0000000000000000"
158 PATTERNS#4,"0C1E3F3F3F3F1E0C"
160 MAG1
170 Y=90:REM INITIAL BAT POSITION
200 REM HANDBALL COURT
210 SCREEN2,2:CLS
218 IFBGTHEN222
220 COLOR2,15,(0,0)-(255,191),15
222 BG=1
240 LINE(10,3)-(245,185),3,B
242 LINE(08,1)-(247,187),3,B
250 GOSUB 900
254 COLOR1
258 RESTORE
260 FORF=1TO10
262 READW$
264 CURSOR20,F*10
266 PRINTW$
267 IFF=2THENCOLOR4:BEEP:BEEP:BEEP
268 IFF=6THENCOLOR9:BEEP
269 NEXT:BEEP
270 DATASPRINGNET,=========,,USE,UP,AR
ROW,,AND,DOWN,ARROW,
280 CURSOR20,120
282 COLOR13
284 PRINT"LEVEL";LV
300 REM POSITION BAT
310 A$=INKEY$
320 IFA$=CHR$(30)THENY=Y-10:IFY=-10THE
NY=0
322 IFA$=CHR$(31)THENY=Y+10:IFY=180THE
```

```
NY=170
350 SPRITE0,(100,Y),0,6
400 REM BALL POSITION
410 BL=BL+1
420 BX=BX+XD
424 BY=BY+YD
430 IFBX >RLTHENXD=-XD:BP=1
440 IFBY<ULTHENYD=AY:BP=1
442 IFBY>LLTHENYD=-AY:BP=1
443 IFBX=180THEN486
444 IFBX<0THEN700:REM MISSED
450 SPRITE2,(BX,BY),4,1
452 IFBPTHENBEEP:BP=0
454 IFBX=100THEN470
460 GOTO 310
470 IFY>BYTHEN310
474 IFY<=BYTHENIFY+20>BYTHEN480
478 GOTO 310
480 XD=-XD:ST=ST+1:BP=1
482 YD=(INT(RND(5)*3)-1)*10
484 GOTO310
486 IFXD<0THEN450
487 IFRND(5)>CHTHEN450
488 BP=1:XD=-AX
490 GOTO450
700 REM SCORE
702 SOUND1,110,15
704 FORF=1TO100:NEXT
706 SOUND0
710 SC=SC-1
720 IFSC=0THEN800:REM GAME END
725 BX=INT(RND(6)*5)*10+90
730 BY=0:YD=10:XD=10
740 GOTO310
800 REM GAME END
810 CURSOR20,140
815 COLOR1
```

```
820 PRINT"YOU MADE";ST;" GOOD STROKES!

830 ST=0
880 CURSOR20,160
881 PRINT"PRESS BAR FOR MORE"
882 CURSOR20,170
883 PRINT"PRESS N FOR NEW LEVEL
884 A$=INKEY$
885 IFA$<>" "ANDA$<>"N"THEN884
886 BEEP
888 IFA$=" "THENGOTO100
890 GOTO 50
900 REM BUILD NET
904 COLOR13
906 X=186
910 FORW=0TO182STEP4.5
920 LINE(X,W)-(X+9,W)
922 LINE(X+9,W)-(X+9,W+9)
924 LINE(X+9,W+9)-(X,W)
926 LINE(X,W)-(X,W+9)
930 NEXT
950 RETURN
```

STEERING TEST



You are driving a fast car in the dark. There are occasional white strips on your left and right to guide you. The harder the level you choose, the narrower the road.

How to play

Use the keys *less than* < and *greater than* > to stay on the road. It is wise to stay near the middle so that you are not caught out by sudden curves.

Notes on the program

In lines 160 to 166 we design our racing car and in line 170 we say MAG 3 to increase its scale. Lines 360 to 362 erase the road side lines and lines 420 to 450 randomly shift them three pixels to the left or the right. Now, we don't exactly alter the numbers in the line drawing commands. We shift the POSITION of our (X, Y) axes. We do this in line 455 simply by adjusting the X value three positions to the left or right. In line 505 we have to compensate for this shift of POSITION by placing the car three pixels in the opposite direction (SX = SX-SH) to make it appear to be travelling in its previous position.

Changes you can make

If you find you are too good for this test, make a narrower road, or if you want to give a little friend a fair go, make a wider road. You can also use a screen color change to enhance the crash, should it occur.

Program listing

```
30 REM INSTRUCTIONS
35 CLS
40 PRINT"STEERING TEST"
42 PRINT"-------------"
44 PRINT:PRINT:PRINT
50 PRINT"YOU ARE DRIVING THROUGH THE N
IGHT
51 PRINT
52 PRINT"IN A HIGH-SPEED RACING CAR. Y
OUR
53 PRINT
54 PRINT"ONLY GUIDES ARE THE INTERMITT
ENT
55 PRINT
56 PRINT"WHITE LINES AT THE SIDES OF T
HE ROAD.
64 PRINT:PRINT:PRINT
70 PRINT"TO STEER LEFT, PRESS <
72 PRINT
74 PRINT"TO STEER RIGHT, PRESS >
76 PRINT:PRINT:PRINT
80 PRINT"PRESS BAR WHEN READY
82 IFINKEY$<>" "THEN82
84 BEEP
86 SCREEN1
100 REM DEGREE OF DIFFICULTY
110 X1=40:X2=89:P=100:REM ROAD SIDE IN
ITIAL POSITIONS
112 SX=44:SY=120:REM SPRITE INITIAL PO
SITIONS
114 SD=120:REM INITIAL SOUND PITCH
120 REM CHOICE
123 SCREEN1,1
124 CLS
125 COLOR1,15
```

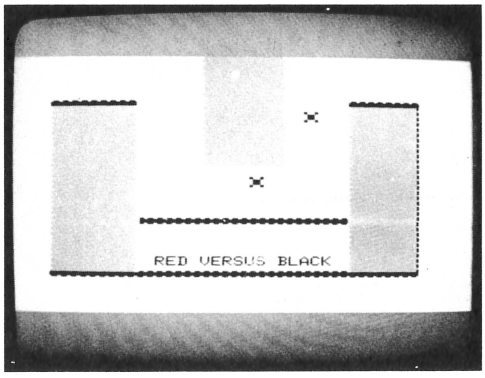
```
126 PRINT"CHOOSE:
128 PRINT"=======
129 PRINT
130 PRINT"1. JUNIOR
131 PRINT
132 PRINT"2. SENIOR
133 PRINT
134 PRINT"3. ACE
135 PRINT
137 PRINT
138 PRINT"PRESS 1-3
140 A$=INKEY$
142 L=VAL(A$)
144 IFL<1ORL>3THEN140
146 IFL=1THENX2=X2+6
147 IFL=3THENX2=X2-6
150 REM DESIGN CAR
160 PATTERNS#0,"03CFCBFFCCCD0C0D"
162 PATTERNS#1,"0C0F0CCCC8F9CFCE"
164 PATTERNS#2,"C0F3D3FF33F330B0"
166 PATTERNS#3,"30F03033139FF373"
170 MAG3
200 REM TRACK
205 SC=0:REM 0 SCORE
210 SCREEN2,2:CLS
218 IFBGTHEN300
220 COLOR15,1,(0,0)-(255,191),1
222 BG=1
300 REM DRAW TRACK
360 BLINE(X1,0)-(X1,191)
362 BLINE(X2,0)-(X2,191)
400 REM SHIFT
420 R=INT(RND(7)*2)+1
430 ONRGOTO432,434
432 SH=+3:GOTO450
434 SH=-3:GOTO450
450 P=P+SH:IFP<0THENP=1
```

```
452 IFP>252THENP=252
455 POSITION(P,0),0,0
460 LINE(X1,0)-(X1,191),15
462 LINE(X2,0)-(X2,191),15
500 REM CAR POSITION
505 SX=SX-SH
510 A$=INKEY$
512 IFA$=","THENSX=SX-3
514 IFA$="."THENSX=SX+3
530 SPRITE0,(SX,SY),0,9
600 REM TEST FOR PRANG
610 IFSX<40THEN800
620 IFSX>X2-31THEN800
700 REM NOISE
705 SV=RND(9)*3-1
707 SD=SD+SV
710 SOUND3,SD,10
750 REM SCORE
755 SC=SC+1
790 GOTO 300
800 REM END
820 CURSOR45,100
825 COLOR3
830 PRINT"PRANG!"
831 COLOR7
832 CURSOR45,10
833 PRINT"SCORE
834 CURSOR45,20
835 PRINT"=";SC/10;" KM"
840 CURSOR25,170
842 COLOR10
850 SV=RND(9)*1000
852 SOUND3,3000+SV,0
854 SOUND4,3,15
856 FORF=1TO100:NEXT
858 SOUND0
860 PRINT"BAR FOR MORE"
```

```
862 IFINKEY$<>" "THEN862
864 BEEP2
866 SCREEN1
870 GOTO 100
```

SPEEDCAR

This is a race for two players. One uses the keyboard and the other uses the joystick. You must take every corner before you can win. If you drive into the oil slick, you lose traction. You will get off, but it will take time. In order to win, you must properly take each bend of the circuit without cutting the corner.



How to play

Use the joystick in the usual way to guide the speed car (joystick port 2). Keyboard operator uses A and Z (for up and down) and S and W (for left and right). Driving on the grass is possible but never an advantage.

Notes on the program

The aim of the game is to complete the circuit in a shorter time than the existing best score. To get the ball rolling, line 2 sets the best score at a very high level.

The program 'knows' whether we've covered the course legally because a 'flag' is set as each corner is passed. Initially, the corner flags are set at zero. Lines 52 to 54 set up the flags F, G and H for the red and black cars. Line 105 tests for entry to the oil slick area. Line 297 tests for a collision between the two cars. The ABS function means 'plus or minus' so the test is whether the distance between the two cars is within eight pixels in any direction. Line 298 checks that all three flags have a value (namely, 1).

The subroutine beginning at line 1000 prints the circular characters just four pixels apart (eg, line 1010). This ensures that the discs overlap and that no color will 'leak out' when we PAINT the field areas and the oil slick area. All paint areas must be absolutely closed. That's why we have the two orange *I*'s at the top of the oil slick.

Changes you can make

You can design your own cars. You can use a different car sprite for up-down movement or even four different sprites. You can design bikes, boats or planes. You can alter the speed.

Program listing

```
2 HI=9999 :REM SET POOR TIME RECORD
5 REM SPEEDCAR GAME
6 O$=CHR$(236):REM OIL BARREL CHARACTE
R
7 GOSUB900
8 TB=TI:REM TIME BEGIN
9 WN=0:MO=2000
10 GOSUB1500:REM INSTRUCTIONS
12 SCREEN2,2:CLS
15 COLOR1,14,(0,0)-(255,191),12
17 GOSUB1000:REM RACE TRACK
20 PATTERNS#4,"C3C33C3C3C3CC3C3"
50 RX=20:RY=10:REM RED CAR  POSITION
51 BX=20:BY=19:REM BLACK CAR  POSITION

52 FR=0:FB=0:REM CORNER FLAGS
53 GR=0:GB=0:REM CORNER FLAGS
54 HR=0:HB=0:REM CORNER FLAGS
56 SP=10:REM HORIZ & VERT SPEED
60 MAG1
100 REM RED DRIVER INPUT
105 IFRY<80 THENIFRX>105 THEN IFRX<145
THENSP=1:REM OIL SLICK
110 A=STICK(2)
120 IFA=1THEN160
125 IFA=7THEN170
130 IFA=3THEN180
135 IFA=5THEN190
140 GOTO195
160 RY=RY-SP
```

```
161 IFRY<0THEN190
162 GOTO195
170 RX=RX-SP
171 IFRX<0THEN180
172 IFRY>165THEN IFRX<35THEN HR=1:REM
LOWER LEFT FLAG
173 GOTO195
180 RX=RX+SP
181 IFRX>245THEN170
182 IFRX>215THEN IFRY<35THEN GR=1:REM
UPPER RIGHT FLAG
183 GOTO195
190 RY=RY+SP
191 IFRY>185THEN160
193 GOTO195
195 SPRITE3,(RX,RY),4,6:SP=10
196 IFRY>165THEN IFRX>215THENFR=1:REM
DISTANCE FLAG
197 IFABS(RX-BX)<8AND ABS(RY-BY)<8THEN
1200:REM CRASH
198 IFFRTHEN IFGRTHEN IFHRTHEN IFRY<35
THENWN=1:GOTO 1300:REM WIN
199 MO=MO+20:SOUND3,MO,0:SOUND4,3,9
200 REM BLACK DRIVER INPUT
205 IFBY<80 THEN IFBX>105 THENIFBX<145
THENSP=1:REM OIL SLICK
210 A$=INKEY$
220 IFA$="W"THEN260
225 IFA$="A"THEN270
230 IFA$="S"THEN280
235 IFA$="Z"THEN290
240 GOTO295
260 BY=BY-SP
261 IFBY<0THEN290
262 GOTO295
270 BX=BX-SP
271 IFBX<0THEN280
```

```
272 IFBY>165THEN IFBX<35THEN HB=1:REM
LOWER LEFT FLAG
273 GOTO295
280 BX=BX+SP
281 IFBX>245THEN270
282 IFBX>215THEN IFBY<35THEN GB=1:REM
UPPER RIGHT FLAG
283 GOTO295
290 BY=BY+SP
291 IFBY>185THEN260
293 GOTO295
295 SPRITE2,(BX,BY),4,1:SP=10
296 IFBY>165THEN IFBX>215THENFB=1:REM
DISTANCE FLAG
297 IFABS(RX-BX)<8AND ABS(RY-BY)<8THEN
1200:REM CRASH
298 IFFBTHEN IFGBTHEN IFHBTHEN IFBY<35
THEN1300:REM WIN
299 MO=MO+10:SOUND3,MO,0:SOUND4,3,9
300 GOTO 100
900 REM GET TIME IN SECONDS
910 MI$=MID$(TIME$,4,2)
920 MI=VAL(MI$)
930 SE$=MID$(TIME$,7,2)
940 SE=VAL(SE$)
950 TI=60*MI+SE
960 RETURN
1000 REM RACE TRACK
1010 FORF=34TO160STEP4
1015 CURSOR25,F:PRINTO$
1020 CURSOR220,F:PRINTO$
1025 NEXT
1030 Y=160:FORX=25TO220STEP5
1032 CURSORX,Y:PRINTO$
1034 NEXT
1040 Y=32:FORG=0TO160STEP160:FORX=25+G
TO60+GSTEP5
```

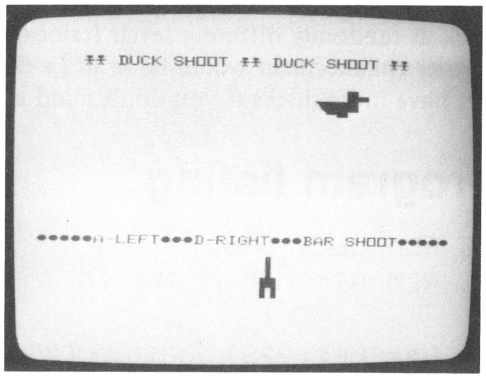
```
1042 CURSORX,Y:PRINTO$
1044 NEXT:NEXT
1050 Y=120:FORX=65TO185STEP5
1052 CURSORX,Y:PRINTO$
1054 NEXT
1060 FORF=32TO158STEP4
1065 CURSOR65,F:PRINTO$
1067 CURSOR185,F:PRINTO$
1068 NEXT
1070 COLOR9:REM HAY BALES
1072 FORF=0TO80STEP5
1074 CURSOR105,F:PRINTO$
1078 CURSOR145,F:PRINTO$
1082 NEXT
1090 FORF=105TO145STEP4
1094 CURSORF,80:PRINTO$
1096 NEXT
1098 CURSOR120,10:PRINT"OIL"
1099 CURSOR114,20:PRINT"SLICK
1100 REM PAINT TRACK
1110 PAINT(50,50),2:REM LEFT FIELD
1115 PAINT(200,50),2:REM RIGHT FIELD
1117 CURSOR105,0: PRINT"I": REM CLOSE
FOR PAINT
1119 CURSOR145,0:PRINT"I"
1120 PAINT(125,5),5:REM OIL PATCH
1130 COLOR1
1132 CURSOR80,150:PRINT"RED VERSUS BLA
CK"
1199 RETURN
1200 REM CRASH
1205 COLOR1
1210 CURSOR100,135:PRINT"**CRASH**"
1220 FORF=15TO0STEP-.3
1222 SOUND4,RND(7)*3,F
1224 NEXT
1240 FORF=1TO500:NEXT
```

```
1250 GOTO1430
1300 REM WIN
1310 FORF=1TO4
1312 SOUND0
1315 SOUND1,841,15
1316 SOUND2,1682,15
1318 COLOR14: CURSOR80,150: PRINT"RED
VERSUS BLACK":COLOR1
1320 IFWNTHENCOLOR6:CURSOR80,180:PRINT
"□□RED CAR WINS□□":GOTO 1340
1330 COLOR1:CURSOR80,180:PRINT"□BLACK
CAR WINS□"
1340 FORG=1TO50:NEXT:SOUND0
1345 COLOR14
1350 IFWNTHENCURSOR80,180:PRINT"□□RED
CAR WINS□□" :GOTO 1370
1360 -
1370 FORG=1TO50:NEXT
1380 NEXT:COLOR1
1400 REM SCORE DISPLAY
1402 BEEP2
1420 GOSUB900:TM=TI-TB
1423 CURSOR80,140 : PRINT"YOUR TIME ="
;TM
1426 CURSOR80,130: PRINT"BEST TIME =";
HI
1428 IFTM<HITHENHI=TM
1430 CURSOR80,100:PRINT"PRESS BAR
1432 CURSOR80,110:PRINT"WHEN READY
1434 IFINKEY$<>" "THEN1434
1444 SC=0:SCREEN1,1:GOTO5
1500 REM INSTRUCTIONS
1510 SCREEN1,1
1520 CLS
1530 PRINT"SPEEDCARS
1532 PRINT"=========
1534 PRINT
```

```
1536 PRINT"FOR 2 PLAYERS.
1538 PRINT
1540 PRINT
1542 PRINT"ONE USES JOYSTICK
1544 PRINT
1546 PRINT"OTHER USES KEYS A,W,S,Z
1548 PRINT:PRINT:PRINT
1550 PRINT"Press BAR when ready.
1560 IFINKEY$<>" "THEN1560
1565 BEEP2
1570 RETURN
```

DUCK SHOOT

A duck flaps its way (rather woodenly) across the screen while you position your shotgun at the bottom, firing when ready. If your shot meets the duck — not as easy as it sounds — you get a flash of color, a *squawk* and some points. The easy shots are on the right where you have plenty of time to see the duck coming.



How to play

Use A for left, D for right and SPACE BAR for fire. Anticipate where the duck will be by the time your shot reaches it. To get higher points, shoot from a position further left.

Notes on the program

Lines 24 to 34 assemble various keyboard characters to build parts of the duck and other screen decoration. You can translate these directly into graphic characters within quotes.

Lines 110 and 115 make a *wing-down* duck and a *wing-up* duck.

Line 120 assembles the spaces that will erase the duck when it reaches right-of-screen.

Line 140 makes the gun. Note the two spaces on each side of the gun. These automatically erase the gun as it moves left and right (two spaces at a time).

Line 340 erases the duck when it reaches column 31 and resets the duck position to column 1.

Line 605 awards a higher score the closer the bird is to the left of the screen.

Lines 607 to 609 create the squawk by combining two slightly dissonant pitches.

Changes you can make

If the game becomes too easy, you can spice it up by making a random delay between duck disappearance on the right and the new duck appearance on the left. This way high scoring shots are much trickier. You can also introduce the duck at randomly different levels (relocating the score listings, of course). A further enhancement would be to use a duck sprite on a colored screen. You can have more ducks if you don't mind a slower game.

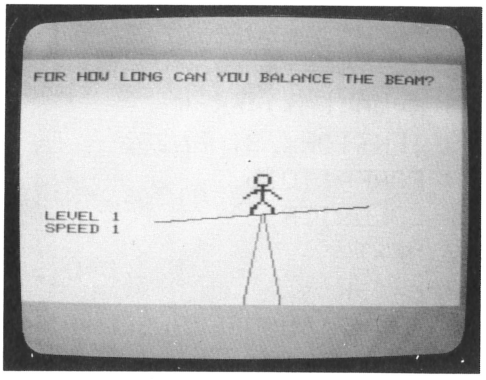
Program listing

```
20 REM DUCK SHOOT
22 REM GRAPHIC ABBREVIATIONS,DUCK PART
S
23 B$=CHR$(229):REM BLOCK
24 D1$="  "+CHR$(151)+B$+B$+B$+"  "
26 D2$="     "+B$+CHR$(155)
28 D3$="   "+CHR$(157)+CHR$(156)+CHR$(1
52)+"  "
30 D4$="   "+CHR$(157)+CHR$(156)+"   "
32 ME$=CHR$(253)+CHR$(253):REM 2 MEN
34 Z$=CHR$(236)+CHR$(236)+CHR$(236):RE
M DECORATIVE BALLS
40 C$=""
41 FORF=1TO6
42 C$=C$+CHR$(29)
43 NEXT
44 C$=C$+CHR$(31)
50 COLOR1,14
60 GN=0:BY=19:GX=6:GY=19:SC=0:ST=0
90 CLS
100 REM MAKE DUCK
110 B1$=D2$+ C$ + D1$+ C$ + D3$
115 B2$=D2$+ C$ + D1$+ C$ + D4$
120 CL$="       "+C$+"       "+C$+"
  ":REM 6 SPACES EACH
130 REM GUN
132 L$=CHR$(225):R$=CHR$(226):REM LEFT
```

```
 AND RIGHT HALF BLOCKS
140 GU$="  "+R$+"    "+C$+"  "+R$+"    "
+C$+"  "+B$+L$+"  "+C$+"  "+L$+L$+"  "

150 BU$="."+CHR$(29)+CHR$(31)+" "
160 BC$=CHR$(31)+" ":REM BULLET CLEAR
200 REM SCREEN INSTRUCTIONS
210 CURSOR4,0: PRINTME$;" DUCK SHOOT "
;ME$;" DUCK SHOOT ";ME$
220 CURSOR0,17:   PRINTƹ$;CHR$(236);CH
R$(236);"A-LEFT";ƹ$;"D-RIGHT";ƹ$;"BAR
SHOOT";ƹ$;CHR$(236);CHR$(236)
250 CURSORGX,GY:PRINTGU$
300 REM MOVE DUCK
310 DX=DX+1
314 CURSORDX,3:PRINTB1$
318 CURSORDX,3:PRINTB2$
340 IFDX=31THENCURSOR31,3:PRINTCL$:DX=
1
400 REM MOVE GUN
410 A$=INKEY$
420 IFA$="A"THEN460
430 IFA$="D"THEN470
440 IFA$=" "ANDGN=0THENBX=GX+3: BY=18:
GN=1
450 GOTO510
460 GX=GX-2
462 IFGX<6THEN470
464 GOTO 480
470 GX=GX+2
472 IFGX>30THEN460
480 CURSORGX,19:PRINTGU$
500 REM BULLET MOVE
510 IFGN=0THEN310
515 BY=BY-1
520 IF  BY=3 THENIFBX>DXANDBX<DX+6THEN
GOTO600
```

```
525 IFBY=2 THENGN=0:CURSORBX,BY: PRINT
BC$:BY=19:GOTO310
530 CURSORBX,BY:PRINTBU$
590 GOTO310
600 REM SQUAWK
605 SE=INT((36-BX)/2): REM SCORE THIS
SHOT
606 COLOR1,SE
607 SOUND1,2000,15
608 SOUND2,2500,15
609 FORT=1TO10:NEXT:SOUND0
610 CURSORBX,3:PRINTCL$
615 SC=SC+1 :REM SUCCESSFUL SHOTS
617 ST=ST+SE:REM SCORE TOTAL
620 CURSOR0,SC+5:PRINT"SQUAWK";SE
625 COLOR1,14
630 IFSC=10THENPRINT:       PRINT"GOOD
SHOOTING!                     ":  GOTO
660
640 GOTO310
660 BEEP2
663 PRINT"YOUR TOTAL =";ST
664 PRINT
665 PRINT"press bar for more"
670 IFINKEY$<>" "THEN670
680 GOTO40
```

BALANCE BEAM



In this game the player is a gymnast who moves back and forth along a see-saw beam. When it tilts you must move to counter-balance it. If you move too far in the wrong direction (or if you step off the end), you fall off!

How to play

Use the *less than* < and *greater than* > keys for movement left and right. There are three levels of play. The first level is for beginners: you only crash if you step off the beam. The second level is more difficult: you crash if your position is the exact reverse of what it should be. The third level requires still better balance. Points are awarded for the number of times you can reach the extremity of the beam.

Notes on the program

Lines 70 to 73 contain the data to make the man. We choose the large size sprite in line 78 with our MAG 3 command. Lines 112 and 113 create the fulcrum or permanent lines in the picture. The movable beam is displayed by line 407, which contains the variable V representing the vertical displacement from level. The score is displayed not in numerals, but in little 'scratch marks'. This is because you cannot erase text on the graphic screen. The little routine beginning with line 900 gives us vertical display of text.

Changes you can make

You can have more positions for the beam. You can have more positions for the man. You can require the man's feet always to touch the beam.

Program listing

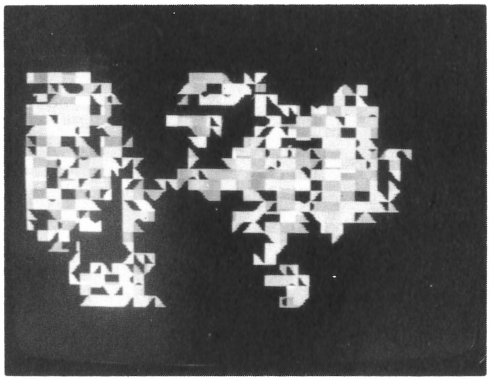
```
20 REM CHOOSE LEVEL
22 CLS:SC=0
24 PRINT"CHOOSE LEVEL (1-3)
26 A$=INKEY$
28 K=VAL(A$):KK=K
29 IFK<1ORK>3THEN26
30 FORF=1TO15
31 COLOR1,F
32 NEXT
33 K=4-K
34 PRINT"CHOOSE SPEED (1-9)
36 A$=INKEY$
38 S=VAL(A$):SS=S
40 IFS<1ORS>9THEN36
41 S=100-10*S
42 FORF=1TO15
44 COLOR1,F
46 NEXT
50 REM DISPLAY
55 SCREEN2,2:CLS
57 COLOR1,15,(0,0)-(255,191),7
60 CURSOR12,10:COLOR6
62 PRINT"FOR HOW LONG CAN YOU BALANCE
THE BEAM?"
63 COLOR4:CURSOR20,120:PRINT"LEVEL";KK

64 COLOR12:CURSOR20,130:PRINT"SPEED";S
S
70 PATTERNS#0,"030404040301070D"
71 PATTERNS#2,"804040408000C020"
72 PATTERNS#1,"1121010204OC0808"
73 PATTERNS#3,"1008008040202020"
78 MAG3
80 X=125:Y=88
90 SPRITE1,(X,Y),0,6
```

```
112 LINE(130,191)-(140,120),4
113 LINE(150,191)-(140,120),4
300 REM MAN SHIFT
305 A$=INKEY$
306 IFA$=","THENBEEP:GOTO 318:REM LEFT

307 IFA$="."THENBEEP:GOTO 310:REM RIGH
T
309 GOTO400
310 IFX=75 THENX=100:Y= 81:Q=2:GOTO350

312 IFX=100THENX=125:Y= 88:Q=3:GOTO350

314 IFX=125THENX=150:Y= 81:Q=4:GOTO350

316 IFX=150THENX=175:Y= 73: Q=5:GOSUB
600:GOTO350
317 IFX=175THEN740
318 IFX=100THENX= 75:Y= 73: Q=1:GOSUB
600:GOTO350
320 IFX=125THENX=100:Y= 81:Q=2:GOTO350

322 IFX=150THENX=125:Y= 88:Q=3:GOTO350

324 IFX=175THENX=150:Y= 81:Q=4:GOTO350

326 IFX=75THEN700
350 SPRITE1,(X,Y),0,6
355 FORG=1TO5:NEXT
360 IFABS(P-Q)>KTHEN800
400 REM BAR SHIFT
405 T=INT(RND(8)*2)
407 LINE(80,120+V)-(200,120-V),15
408 IFV=-15THENV=-7:P=2:GOTO 450
410 IFV=15THENV=7:P=4:GOTO 450
411 IFTTHEN418
412 IFV=7THENV=15:P=5:GOTO450
```

```
414 IFV=0THENV=7:P=4:GOTO450
416 IFV=-7THENV=0:P=3:GOTO450
418 IFV=7THENV=0:P=3:GOTO450
420 IFV=0THENV=-7:P=2:GOTO450
422 IFV=-7THENV=-15:P=1
450 LINE(80,120+V)-(200,120-V),1
455 FORG=1TOS:NEXT
499 GOTO300
600 REM SCORE
604 BEEP
608 COLOR1
610 SC=SC+6
612 CURSOR50+SC,30:PRINTCHR$(225)
620 RETURN
700 REM CRASH LEFT
701 SPRITE1,(X,Y),0,15
702 FORF=152TO120STEP-50
703 SOUND1,F,15:NEXT
704 SOUND3,2000,0:SOUND4,3,15
705 FORF=1TO10:NEXT:SOUND0
710 Y=160:X=90
711 COLOR1
712 CURSOR50,160:PRINT"CRASH"
720 SPRITE1,(X,Y),0,6
739 GOTO770
740 REM CRASH RIGHT
741 SPRITE1,(X,Y),0,15
742 FORF=152TO120STEP-50
743 SOUND1,F,15:NEXT
744 SOUND3,2000,0:SOUND4,3,15
745 FORF=1TO10:NEXT:SOUND0
750 Y=160:X=190
760 SPRITE1,(X,Y),0,6
761 COLOR1
762 CURSOR220,160:PRINT"CRASH"
770 GOTO900
799 GOTO 799
```

```
800 REM SIDE
810 IFQ<3THEN700
820 IFQ>3THEN740
830 T=INT(RND(8)*2)+1
840 ONTGOTO700,740
900 REM RESTART
902 RESTORE
905 COLOR1
910 FORF=1TO5
920 CURSOR20,10*F+15
922 DATAPRESS,BAR,TO,PLAY,AGAIN
925 READW$
930 PRINTW$
935 NEXT
940 IFINKEY$<>" "THEN940
950 GOTO20
```

LIGHT SHOW



This is a space age light show designed to intrigue your friends. You can set this program running as a 'side show' when there is a party or gathering. Your Sega will draw pretty designs of its own making.

How to use the program

You decide how many different colors will be used in the design by selecting a number at the beginning. This allows all the colors up to that color code number to be used. When the display reaches a particularly interesting stage, onlookers can press the bar to freeze it. Another bar press will start it off again.

Notes on the program

Line 111 uses the ON. . .GOTO command to select the next direction the painting will take. ON. . .GOTO is an abbreviation for a whole series of IF. . . THEN instructions. Lines 192 to 196 contain the characters used in the display. You can of course change these to suit yourself. The little routine from 200 to 228 gives us a randomly selected sound with *decay,* which is easier on the ear than full-on, full-off sound.

Changes you can make

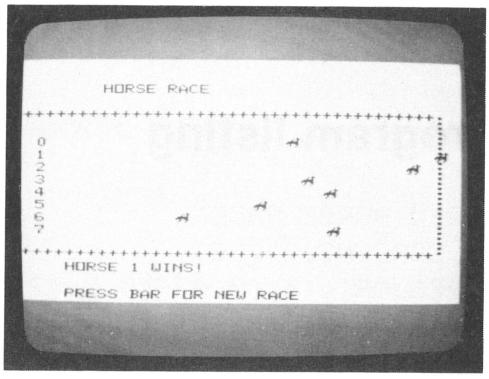
To give full artistic control to the operator you can invite users to input the precise colors of their choice by listing the colors against their code numbers on an initial text screen. You can also control what characters your Sega uses in creating the design. For example, you can specify all blocks, or blocks and half-blocks, or use all the 'stick' graphics to see how they knit together. If you want to, you can omit the black color options so that there are no 'holes' in your design.

Program listing

```
10 REM MARTIAN LIGHT SHOW
20 CLS
22 PRINT"INPUT COLOR VARIETY (1-14)
23 PRINT
24 INPUTCL
26 IFCL<1ORCL>14THEN20
28 CL=CL+1
60 REM SCREEN
62 SCREEN2,2:CLS
64 COLOR1,1,(0,0)-(255,191),1
72 X=19:Y=11:F=15
100 REM DISPLAY
110 D=INT(RND(7)*4)+1
111 ONDGOTO113,114,115,116
113 X=X+1:GOTO122
114 X=X-1:GOTO120
115 Y=Y+1:GOTO124
116 Y=Y-1:GOTO126
120 IFX<0THENX=0
122 IFX>41THENX=41
124 IFY>22THENY=22
126 IFY<1THENY=1
130 IFINKEY$=" "THEN400
140 REM COLOR
144 C=INT(RND(5)*CL)+1
148 COLORC
190 CURSORX*6,Y*8
191 ONINT(RND(8)*5)+1GOTO 192,193,194,
195,196
192 PRINTCHR$(229):GOTO199
193 PRINTCHR$(149):GOTO199
194 PRINTCHR$(150):GOTO199
195 PRINTCHR$(151):GOTO199
196 PRINTCHR$(152):GOTO199
199 SOUND0
```

```
200 REM SOUND
210 A=INT(RND(9)*150)+110
214 FORF=15TO3STEP-3
224 SOUND1,A,F
228 NEXT
240 GOTO100
400 REM STOP PICTURE
402 BEEP
405 SOUND0
410 FORF=1TO100:NEXT
420 A$=INKEY$
422 IFA$=" "THENBEEP2:GOTO 10
424 GOTO420
```

HORSE RACE



This program takes you to the races. Lay your bets on your favourite color or number and then see who makes it first across the finish line. Genuine randomisation — a thriller for people who like to bet.

How to play

Just press the space bar.

Notes on the program

The artistic work of the program is done in line 20 where we describe the horse. Lines 65 and 70 print the rails of the racetrack using the + character. Lines 410 and 420 generate the random effects of the program. Notice that randomisation applies both to the next horse chosen to move and also to the size of the step it takes. The size of the step may vary between six and eight places.

Changes you can make

You can change the number of horses in the race to suit the number of players (or punters!). You can increase or decrease the speed of the race by changing the size of the basic step. You can, if you want to, place the speed under the control of the user by using a step variable INPUT before the race starts. Hurdle races are also possible. You just have to place vertical bars along the track for each horse and then have the horse jump them or baulk at them — at random.

Handicapping and the creation of true 'odds' are also possible with this program. To create the effect of a handicap, you simply vary the size of the

step for each horse (or vary the frequency with which each horse takes a step). To call the odds, you work out the *average* step for each horse. So, for example, if a horse's step randomly varies between six and eight, his average step will be seven. If another's step randomly varies between two and five, his average will be three and a half. The odds between these two horses will be two to one.

Program listing

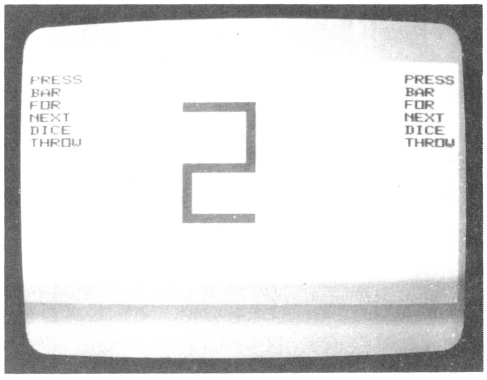
```
5 REM HORSE RACE
10 SCREEN2,2:CLS
12 COLOR1,15,(0,0)-(0,0),2
16 REM MAKE HORSES
18 FORF=0TO7
20 PATTERNS#F,"020B127EFEA62244"
22 NEXT
24 REM PLACE AT STARTER
25 FORF=0TO7:X(F)=20:Y(F)=F*10+60
27 COLOR1:CURSOR7,F*10+62:PRINTF
30 SPRITEF,(X(F),Y(F)),0,F+1
40 NEXT
50 REM MARK TRACK
60 CURSOR50,20:PRINT"HORSE RACE"
65 CURSOR0,40:PRINT"++++++++++++++++++
++++++++++++++++++++
70 CURSOR0,150:PRINT"++++++++++++++++++
++++++++++++++++++++
80 FORG=38TO146STEP4
82 CURSOR240,G:PRINT"."
84 NEXT
100 REM START
105 COLOR1
110 CURSOR30,170 : PRINT"PRESS BAR TO
START"
115 IFINKEY$<>" "THEN115
117 FORF=1TO4:BEEP:NEXT
120 COLOR15
```

```
125 CURSOR30,170 : PRINT"PRESS BAR TO
START"
390 J=VAL(RIGHT$(TIME$,1)): IF J=0THEN
390
395 K=VAL(MID$(TIME$,7,1)): IF K=0THEN
395
400 REM PICK HORSE
410 H=INT(RND(J)*8):REM HORSE NUMBER
420 S=INT(RND(K)*3)+6:REM SIZE STEP
430 SOUND3,(RND(6)*3)*510+500,0
434 SOUND4,3,15
500 REM UPDATE POSN
510 X(H)=X(H)+S
520 CURSORX(H),Y(H)
530 SPRITEH,(X(H),Y(H)),0,H+1
540 SOUND0
550 IFX(H)>237THEN600
590 GOTO400
600 REM WIN
605 BEEP2:BEEP2
610 COLORH+1
620 CURSOR30,161:PRINT"HORSE";H;" WINS
!"
630 COLOR1
640 CURSOR30,182: PRINT"PRESS BAR FOR
NEW RACE
650 IFINKEY$<>" "THEN650
660 GOTO5
```

SUPERDICE

We couldn't decide whether readers would appreciate this one, so we've thrown it in free. It is not counted among your twenty-one programs!

Superdice is a glamorous addition to your games of chance such as *Ludo, Monopoly* and *Snakes and Ladders.* Instead of throwing a dice from a dice cup (the old fashioned way!), the computer displays in bright color a huge numeral that all players and onlookers can clearly see — eliminating all disputes.



How to use superdice

Press the space bar!

Notes on the program

Line 130 reads the data from lines 150 to 160 describing the X and Y coordinates. Line 130 contains a double subscript for a single number because each number actually consists of six lines of dots. In the subroutine starting at line 600, the dots are simply spun from one X position to another X position (line 630) and from one Y position to another Y position (line 664). The lines creating the numbers are either horizontal or vertical. Lines 634 and 640 create the sound effect by tying the pitch of the sound to the number of dots.

Changes you can make

You can increase the number of 'faces' on your superdice to any number you want for the game you're playing (say, twelve or twenty if you're playing *Dungeons and Dragons*). You can even offer the user a choice of a six, eight, twelve or twenty service! To make a traditional dice display, simply substitute the appropriate number of colored blobs for the numerals. The superdice display will appear even faster.

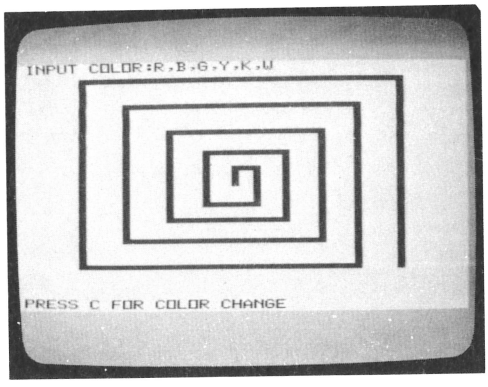
Program listing

```
10 REM SUPER DICE
50 DIMX(6,6),Y(6,6)
100 REM CO-ORDINATES
110 FORF=1TO6
120 FORG=0TO6
130 READX(F,G),Y(F,G)
140 NEXT:NEXT
150 DATA17,4,19,4,19,10,19,15,17,15,21
,15,19,15
152 DATA16,4,22,4,22,10,16,10,16,15,22
,15,16,15
154 DATA16,4,22,4,22,9,17,9,22,9,22,15
,16,15
156 DATA16,4,16,10,22,10,19,10,19,7,19
,15,19,10
158 DATA 22,4,16,4,16,9,22,9,22,15,16,
15,22,15
160 DATA22,4,16,4,16,15,22,15,22,10,16
,10,22,10
180 FORF=1TO6
182 READW$(F)
184 NEXT
186 DATAPRESS,BAR,FOR,NEXT,DICE,THROW
188 DATAPRESS,BAR,FOR,NEXT,DICE,THROW
190 SCREEN2,2:CLS
192 COLOR1,15,(0,0)-(0,0),14
200 REM INVITE THROW
205 COLOR1,15
218 FORF=1TO6
219 COLORF*2
220 CURSOR10,F*10:PRINTW$(F)
221 SOUND0
222 CURSOR224,F*10:PRINTW$(F)
223 SOUND1,110+F*50,15
224 NEXT
```

```
225 FORF=1TO10:NEXT
227 SOUND0
300 REM INPUT BAR
310 A$=INKEY$
320 IFA$=" "THEN390
330 GOTO 310
390 BEEP2
392 CLS
395 K=VAL(RIGHT$(TIME$,1)) : IFK=0THEN
395
400 REM SELECT COLOR & NUMBER
410 C=INT(RND(K)*6)+1
420 C=C*2
440 N=INT(RND(K)*6)+1
500 REM ERASE INVITE
505 COLOR15
518 FORF=1TO6
520 CURSOR10,F*10:PRINTW$(F)
522 CURSOR224,F*10:PRINTW$(F)
524 NEXT
600 REM DISPLAY NUMBER
610 COLORC
620 FORL=0TO5
630 IFX(N,L)=X(N,L+1)THEN660
632 D=1:IFX(N,L)>X(N,L+1)THEND=-1
634 FORI=X(N,L)TOX(N,L+1)STEPD
636 CURSORI*6,Y(N,L)*8
638 PRINTCHR$(229)
640 SOUND1,110+I*100,15
642 NEXT:GOTO674
660 D=1:IFY(N,L)>Y(N,L+1)THEND=-1
664 FORI=Y(N,L)TOY(N,L+1)STEPD
666 CURSORX(N,L)*6,I*8
668 PRINTCHR$(229)
670 SOUND1,110+I*100,15
672 NEXT
674 NEXT
```

```
690 SOUND0
693 RESTORE
694 FORF=11084
695 READA:NEXT
698 GOTO200
```

DRAWING PAD



This is a program for artists. You can choose from six colors to draw the picture of your choice. You can also erase parts of what you've drawn.

How to use drawing pad

To control the direction of the brush you use the arrow keys. When the program starts, the color is black. To change color, press C then the initial letter of the color you want. R = red; G = green; B = blue; Y = yellow; K = black; W = white. White paint on a white canvas serves as an *eraser*. To draw diagonals, you use the arrow keys alternately: for example, ↑ → ↑ → creates a sloping line. Where two colors meet, you may find some 'leakage' because one character area can only accommodate one color (besides screen color).

Notes on the program

Line 50 creates the small square which is the character for a single brush stroke. Lines 220 to 226 alter the X and Y position of our brush according to the input. Lines 620 to 670 respond to key characters and yield color values. Line 690 looks very much like line 605 but the essential difference is the color 15. This erases line 605.

Changes you can make

You can change the size of your brush and you can cause it to create a variety of different textures — for example, a checkered pattern — just by changing the character. You can allow a PAINT option letting the user nominate a color to fill in the whole area surrounding the brush. You can change the program so

that it will draw fine lines in any direction, but to do that you'll have to switch from *character printing* (which we're doing in this program) to *pixel plotting* using the POINT command. This will create fine lines, but the action will be slower.

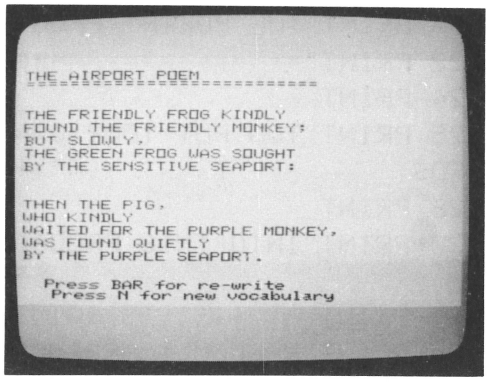
Program listing

```
50 PATTERNC#46,"F0F0F0F000000000"
60 C=1
100 REM SCREEN
110 SCREEN2,2:CLS
120 COLOR1,15,(0,0)-(0,0),7
130 X=124:Y=92
140 COLOR14 :CURSOR10,183:PRINT"PRESS
C FOR COLOR CHANGE"
200 REM DRAW
210 A$=INKEY$
220 IFA$=CHR$(28)THENX=X+4
221 IFX>248THENX=248
222 IFA$=CHR$(29)THENX=X-4
223 IFX<8THENX=8
224 IFA$=CHR$(30)THENY=Y-4
225 IFY<0THENY=0
226 IFA$=CHR$(31)THENY=Y+4
227 IFY>180THENY=180
230 IFA$="C"THENGOSUB 600
240 COLOR3
242 CURSORX,Y:PRINT"."
244 COLORC
250 CURSORX,Y:PRINT"."
290 GOTO210
600 REM GET COLOR
605 COLOR1:CURSOR10,0:PRINT"INPUT COLO
R:R,B,G,Y,K,W"
610 B$=INKEY$
620 IFB$="R"THENC=6:GOTO690
630 IFB$="B"THENC=4:GOTO690
640 IFB$="G"THENC=12:GOTO690
```

```
650 IFB$="Y"THENC=10:GOTO690
660 IFB$="K"THENC=1:GOTO690
670 IFB$="W"THENC=15:GOTO690
680 GOTO610
690 COLOR15:CURSOR10,0:PRINT"INPUT COL
OR:R,B,G,Y,K,W"
695 RETURN
```

SEGA POET

This program invites you to type in several sets of words of your own choosing and then creates a rhyming verse from them. The results are quite unpredictable — and frequently hilarious. Your Sega will try other arrangements of your poem at the press of the bar.



How to play

You will be asked to type in sets of nouns, verbs, adjectives, adverbs and rhyming nouns. The more words you put in, the more unpredictable and fascinating the resulting poem will be. With a little trial and error you'll soon find the right sorts of verbs to use. You need two different sets of rhyming nouns in addition to the first (non-rhyming) list of nouns. To get variety, you need to input at least a few rhyming words of each sound.

Notes on the program

The hard work of this program begins at line 338. Here we select a number between 1 and the number of a certain type of word (GOSUB 900 to get the random number). Then in line 341 we print the variable V$ with the double subscript for that word type (in this case 3) and the nth word (got from the subroutine at 900). We repeat this process many times with different parts of speech through to line 393.

Changes you can make

The program produces its 'poems' by slotting the different sorts of words into a standard structure (or skeleton). To create different sorts of poems, you simply change the skeleton. A good one to start with is the traditional limerick. You work out the structure and then fill the 'holes' with words from the different sorts of lists.

Program listing

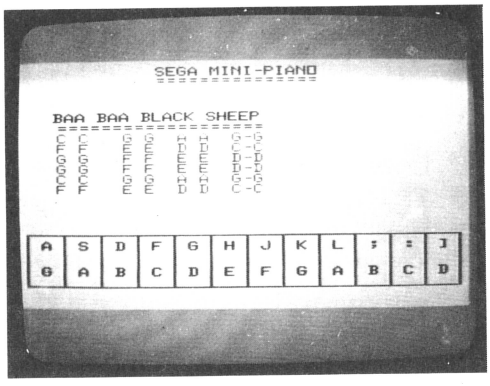
```
100 REM THE POETRY PROGRAM
110 SCREEN1,1:CLS
120 PRINT"THE POETRY PROGRAM
122 PRINT"==================
124 PRINT
126 PRINT"YOU PUT YOUR OWN CHOICE OF W
ORDS
128 PRINT
130 PRINT"INTO THIS PROGRAM AND IT WIL
L
132 PRINT
134 PRINT"RANDOMLY ARRANGE THEM INTO
136 PRINT
138 PRINT"A POEM FORMAT.
139 PRINT
140 PRINT"============================
======
142 PRINT
144 PRINT"YOU CAN RE-RUN THE PROGRAM A
ND GET
146 PRINT
148 PRINT"MANY DIFFERENT POEMS FROM TH
E SAME
150 PRINT
152 PRINT"SELECTION OF WORDS.
154 PRINT
156 PRINT"============================
======
158 PRINT
160 PRINT
162 PRINT"Press BAR when ready.......
"
163 GOSUB800
164 A$=INKEY$
166 IFA$<>" "THEN164
```

```
168 BEEP
200 REM INPUT
210 CLS
212 PRINT"INPUT YOUR WORDS"
213 RESTORE
214 PRINT"================"
215 PRINT
216 FORFF=1TO6:READW$(FF)
217 PRINT"HOW MANY ";W$(FF)
218 INPUTNU(FF)
219 IFNU(FF)>20THEN218
220 PRINT:FORF=1TONU(FF)
221 SW$=LEFT$(W$(FF),LEN(W$(FF))-1)
222 PRINT;SW$;F
223 INPUTV$(FF,F)
224 IFV$(FF,F)=""THEN223
225 NEXT:PRINT
280 NEXTFF
290 GOSUB 800
295 DATANOUNS,(PAST TENSE) VERBS,ADJEC
TIVES,ADVERBS,RHYMING NOUNS,(RHYME 2)
RHYMING NOUNS
300 REM DISPLAY
310 SCREEN2,2:CLS
320 COLOR1,15,(0,0)-(255,191),14
330 CURSOR10,10
331 M=NU(6):GOSUB900
332 PRINT"THE ";V$(6,N);" POEM"
335 CURSOR10,16
337 PRINT"=========================
338 M=NU(3):GOSUB900
339 COLOR1
340 CURSOR10,40
341 PRINT"THE ";V$(3,N);" ";
342 M=NU(1):GOSUB900
343 PRINTV$(1,N);" ";
344 M=NU(4):GOSUB900
```

```
345 PRINTV$(4,N)
346 CURSOR10,50
347 COLOR4
348 M=NU(2):GOSUB900
349 PRINTV$(2,N);" THE ";
350 M=NU(3):GOSUB900
351 PRINTV$(3,N);" ";
352 M=NU(5):GOSUB900
353 PRINTV$(5,N);";"
354 M=NU(4):GOSUB900
355 CURSOR10,60:COLOR6
356 PRINT"BUT ";V$(4,N);","
357 M=NU(3):GOSUB900
358 CURSOR10,70:COLOR12
359 PRINT"THE ";V$(3,N);" ";
360 M=NU(1):GOSUB900
361 PRINTV$(1,N);" WAS ";
362 M=NU(2):GOSUB900
363 PRINTV$(2,N)
364 CURSOR10,80:COLOR1
365 M=NU(3):GOSUB900
366 PRINT"BY THE ";
367 PRINTV$(3,N);" ";
368 M=NU(6):GOSUB900
369 PRINTV$(6,N);":"
370 CURSOR10,110:COLOR1
371 PRINT"THEN THE ";
372 M=NU(1):GOSUB900
373 PRINTV$(1,N);","
374 CURSOR10,120:COLOR4
375 M=NU(4):GOSUB900
376 PRINT"WHO ";V$(4,N)
377 CURSOR10,130:COLOR6
378 M=NU(2):GOSUB900
379 PRINTV$(2,N);
380 M=NU(3):GOSUB900
381 PRINT" THE ";V$(3,N);" ";
```

```
382 M=NU(5):GOSUB900
383 PRINTV$(5,N);","
384 CURSOR10,140:COLOR12
385 M=NU(2):GOSUB900
386 PRINT"WAS ";V$(2,N);
387 M=NU(4):GOSUB 900
388 PRINT" ";V$(4,N)
389 CURSOR10,150:COLOR1
390 M=NU(3):GOSUB900
391 PRINT"BY THE ";V$(3,N);
392 M=NU(6):GOSUB900
393 PRINT" ";V$(6,N);"."
394 COLOR13
395 CURSOR22,172
396 PRINT"Press BAR for re-write"
397 PRINT"   Press N for new vocabular
y
398 GOSUB800
399 A$=INKEY$
400 IFA$=" "THEN300
402 IFA$="N"THEN100
404 GOTO399
800 REM SOUND
810 FORF=0TO15
814 SOUND1,RND(8)*100+110,15
815 COLOR1,F
816 NEXT
818 SOUND0
820 RETURN
900 REM GET RANDOM NUMBER
905 REM M=MAX:N=OUTCOME
910 K=VAL(RIGHT$(TIME$,1))
920 IFK=0THEN910
930 N=INT(RND(K)*M)+1
940 RETURN
```

MINI PIANO



This program turns the Sega keyboard into a small piano keyboard. The ASD. . . row of the keyboard is used (because it has sufficient keys). You can play most simple melodies.

How to use the program

The screen displays the keyboard along with the musical notes. Press the key indicated for the note you want. The Sega will give true pitch. Choose P for piano or O for organ. In *organ mode* you can use the up and down arrows for volume control. For any melody you want to play, first write down the notes in sequence. For most tunes, starting on C, G or E will give the best results. The program, as it is, includes all the notes needed for *Baa Baa Black Sheep* and *Waltzing Matilda.*

Notes on the program

Line 132 holds the important data for this program. The subcripted variable V() converts the ASC value of the key pressed into the pitch value of the note wanted. Thus we use V(number) in our SOUND command at lines 540 and 640.

Changes you can make

For greater musical versatility you can use the keys of the row *above* for sharps and flats. You can also offer the user a screen display like that for *Baa Baa Black Sheep* or *Waltzing Matilda* for your favourite tune.

Program listing

```
20 REM PREPARATION
25 CLS
30 PRINT"PREPARATION OF MINI-PIANO
50 REM BASIC DATA
55 DIML$(25),N$(25),V(256)
100 REM MINI PIANO
110 REM KEYBOARD KEYS
112 FORF=1TO12
114 READL$(F)
116 NEXT
118 DATAA,S,D,F,G,H,J,K,L,";",":",]
122 FORF=1TO12
124 READN$(F)
126 NEXT
128 DATAG,A,B,C,D,E,F,G,A,B,C,D
130 REM NOTE VALUES
131 FORF=1TO255:V(F)=262:NEXT
132 V(65)=196:V(83)=220:V(68)=247:V(70
)=262:V(71)=294:V(72)=330:V(74)=349:V(
75)=392:V(76)=440:V(59)=494:V(58)=523:
V(93)=587
150 CLS
152 COLOR1,10
154 PRINT:PRINT:PRINT:PRINT:PRINT
156 PRINT"CHOOSE
158 PRINT"======
160 PRINT:PRINT
162 PRINT"PIANO or ORGAN
164 PRINT:PRINT
166 PRINT"(Press P or O)
170 CH$=INKEY$
172 IFCH$<>"P"ANDCH$<>"O"THEN170
174 BEEP
176 CLS
178 COLOR1,7
```

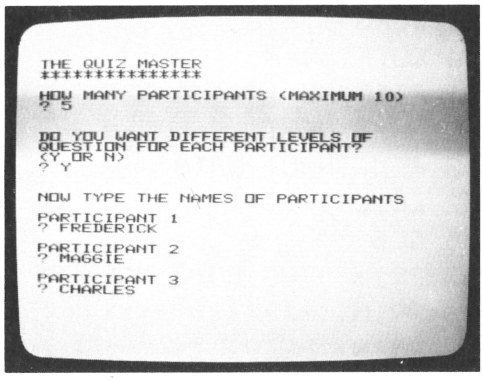
```
180 PRINT:PRINT"DO YOU WANT
182 PRINT:PRINT"BAA-BAA BLACK SHEEP
184 PRINT:PRINT"WALTZING MATILDA
186 PRINT:PRINT"NEITHER
188 PRINT:PRINT"(Press B,W, or N)
190 S$=INKEY$
192 IFS$<>"B"ANDS$<>"W"ANDS$<>"N"THEN1
90
194 BEEP
200 REM SCREEN DISPLAY
210 SCREEN2,2:CLS
215 COLOR4,15,(0,0)-(255,191),14
216 CURSOR80,5
217 PRINT"SEGA MINI-PIANO
218 COLOR6:CURSOR80,12
219 PRINT"==============":COLOR1
220 LINE(10,135)-(250,135),1
222 LINE(10,175)-(250,175),1
224 FORF=1TO13:FF=F*20-10
226 LINE(FF,135)-(FF,175),1
228 NEXT
230 REM KEYBOARD LETTERS
232 FORF=1TO12
234 COLOR9
236 CURSORF*20-2,140
238 PRINTL$(F)
239 COLOR4
240 CURSORF*20-2,160
242 PRINTN$(F)
246 NEXT
260 IFS$="B"THENGOSUB700
262 IFS$="W"THENGOSUB800
490 IFCH$="P"THEN600
500 REM PLAYING THE ORGAN
505 VO=8:REM VOLUME
520 A$=INKEY$
524 IFA$=""THEN520
```

```
526 IFA$=CHR$(30)THENGOSUB920:GOTO520
527 IFA$=CHR$(31)THENGOSUB900:GOTO520
530 NT=V(ASC(A$))
540 SOUND1,NT,VO
560 GOTO520
600 REM PLAYING THE PIANO
601 NT=0
602 NN=1:REM NO NOTE FLAG
605 VO=16:REM VOLUME
620 A$=INKEY$
622 IFA$=""ANDNNTHEN620
624 IFA$=""THEN627
626 IFA$<>""THENNT=V(ASC(A$)):NN=0:VO=
16
627 VO=VO-1:IFVO=-1THEN602
640 SOUND1,NT,VO
660 GOTO620
700 REM BAA BAA
705 COLOR13
710 CURSOR0,40
712 PRINT"    BAA BAA BLACK SHEEP
713 PRINT"   ==================
718 COLOR4
720 PRINT"   C C   G G  A A  G-G
724 PRINT"   F F   E E  D D  C-C
734 PRINT"   G G   F F  E E  D-D
744 PRINT"   G G   F F  E E  D-D
754 PRINT"   C C   G G  A A  G-G
764 PRINT"   F F   E E  D D  C-C
770 RETURN
800 REM WALTZING MATILDA
805 COLOR12
810 CURSOR0,40
812 PRINT"   WALTZING MATILDA
813 PRINT" ===============
815 COLOR6
820 PRINT"  E  EE  D  DD  CD EC AB C
```

```
low A
822 PRINT"  G  CE  G  FE  D  D  C  -
low G and C
826 PRINT"  G  GG  G  E   C  CC B  A
high G
828 PRINT"  G  GG  A  G   G  FE D  CD
830 PRINT"  E  EE  D  DD  CD EC AB C
832 PRINT"  G  CE  G  FE  D  D  C  -
840 RETURN
900 REM DIE
905 IFVO=0THENRETURN
910 VO=VO-1
915 SOUND1,NT,VO
917 RETURN
920 REM INC
925 IFVO=15THENRETURN
930 VO=VO+1
935 SOUND1,NT,VO
937 RETURN
```

QUIZ MASTER

This program enables you to type in a series of questions and answers and the names of the contestants. It indicates whose turn it is and after receiving the answer, reveals the official answer. The judge then awards 0, 1 or 2 points. The total scores are displayed at the end. For the benefit of younger participants you are offered the option of inputting different *levels* of questions and answers for each contestant.



How to play

You need to have two or more participants and a judge. First type in the names of the players and then the questions and answers. Remember to press CR after every input. Players type in their answers in response to the questions that appear on the screen. The judge then awards points as the official answers are displayed. If you are a parent and want to use this program as a homework aid for just one child, simply name the child as both participants!

Notes on the program

Line 50 begins with a DIM for Q$(10,10) to allow for a possible ten questions for a possible ten participants. A$(10,10) is for the answers. QU(10,10) is a numeric array which holds a 1 for each question already asked and 0 for those that haven't been. The first number in the double subscript is the player number and the second number is the question number. To avoid predictability, the routine from 900 on simply swaps question A with question B. It randomly tries this HM x HQ/2 times.

Changes you can make

This is a program you can adapt for the graphics screen, enhancing the interest of the screen display by having the questions and answers appear in different colors. Remember, though, that you'll have to clear the screen for each new question.

Program listing

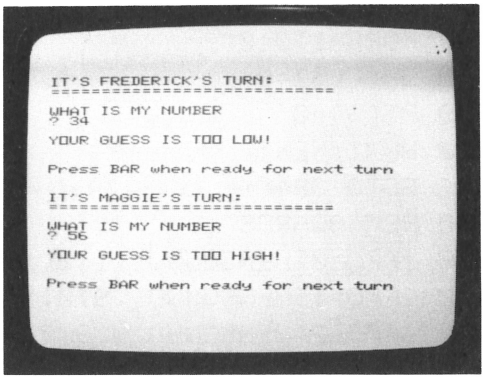
```
50 DIMQ$(10,10),A$(10,10),QU(10,10)
190 FORF=1TO10
191 FORG=1TO10
194 QU(F,G)=0
195 PT(G)=0
196 NEXT:NEXT
200 REM INPUT DATA
202 COLOR1,15
205 CLS
207 PRINT"THE QUIZ MASTER"
208 PRINT"***************"
209 PRINT:BEEP2
210 PRINT"HOW MANY PARTICIPANTS (MAXIM
UM 10)
220 INPUT HM
222 IFHM<2ORHM>10THEN210
225 PRINT
227 PRINT:BEEP2
230 PRINT"DO YOU WANT DIFFERENT LEVELS
 OF        QUESTION FOR EACH PARTICIPAN
T?        (Y OR N)
240 INPUT A$
242 IFA$<>"Y"ANDA$<>"N"THEN230
250 DL=0:IFA$="Y"THENDL=1
255 PRINT
257 'PRINT:BEEP2
260 PRINT"NOW TYPE THE NAMES OF PARTIC
IPANTS"
261 FORF=1TOHM
262 PRINT
263 PRINT"PARTICIPANT";F
264 INPUT PL$(F)
266 NEXT
300 REM INPUT QUESTIONS & ANSWERS
310 PRINT
```

```
311 PRINT:BEEP2
312 PRINT"HOW MANY QUESTIONS PER PARTI
CIPANT    (MAXIMUM 10)
315 INPUTHQ
316 IFHQ<2ORHQ>10THEN312
320 PRINT
322 PRINT"NOW TYPE IN QUESTIONS AND AN
SWERS"
328 FORG=1TOHM
330 FORF=1TOHQ
337 PRINT
338 TA=F:IFDL=0THENTA=F+HQ*(G-1)
340 PRINT"QUESTION";TA
341 SOUND1,200,15:FORY=1TO30:NEXT:SOUN
D0
342 IFDLTHENPRINT"FOR ";PL$(G)
350 INPUTQ$(G,F):REM PL,QN
352 IFQ$(G,F)=""THEN340
358 PRINT
359 TA=F:IFDL=0THENTA=F+HQ*(G-1)
360 PRINT"ANSWER";TA
361 SOUND1,900,15:FORY=1TO30:NEXT:SOUN
D0
362 IFDLTHENPRINT"FOR ";PL$(G)
370 INPUTA$(G,F):REM PL,QN
372 IFA$(G,F)=""THEN360
380 NEXTF:REMHQ
382 NEXTG:REM HM
384 IFDLTHEN400
386 GOSUB900
400 REM QUIZ
405 FORCQ=1TOHQ
410 FORPL=1TOHM
418 PRINT
419 CLS
420 PRINT"HERE IS A QUESTION FOR"
422 PRINTPL$(PL)
```

```
423 PRINT"=======================
424 PRINT
430 QN=INT(RND(6)*HQ)+1
434 IFQU(PL,QN)THEN430
440 PRINTQ$(PL,QN)
442 QU(PL,QN)=1
450 PRINT
452 INPUT AN$
454 PRINT
456 PRINT"Press BAR for official answe
r"
460 IFINKEY$<>" "THEN460
461 PRINT
462 PRINTCHR$(236);" ";A$(PL,QN)
463 PRINT
464 PRINT"Judge award 0,1, or 2 points
"
465 PRINT
466 INPUT PO
468 IFPO>2THEN463
469 GOSUB800
470 PT(PL)=PT(PL)+PO
472 PRINT
480 NEXT
483 NEXT
500 REM FINAL SCORES
505 CLS
520 PRINT"SCORES FOR THE QUIZ ********
"
530 PRINT
532 FORH=1TOHM
534 PRINTCHR$(253);" ";PL$(H);TAB(20);
PT(H)
536 PRINT
538 NEXT
550 PRINT"PRESS BAR FOR SAME QUIZ"
555 PRINT"PRESS N FOR NEW QUIZ"
```

```
560 A$=INKEY$
563 IFA$="N"THEN190
564 IFA$<>" "THEN560
565 BEEP2
566 FORF=1TO10
567 FORG=1TO10
568 QU(F,G)=0
569 PT(G)=0
570 NEXT:NEXT
571 GOTO 384
800 REM AWARDS
810 IFPO=0THENSOUND1,110,15:GOTO850
820 IFPO=1THENSOUND1,910,15:GOTO850
830 IFPO=2THENSOUND1,1910,15:GOTO850
850 FORHH=1TO50:NEXT:SOUND0
860 FORHH=1TO100:NEXT:RETURN
900 REM MIX QUESTIONS
904 PRINT
905 PRINT"** MIXING UP QUESTIONS **"
906 PRINT"** MIXING UP QUESTIONS **"
907 PRINT
910 FORF=1TOHM*HQ/2
920 P1=INT(RND(7)*HM)+1
922 P2=P1+1:IFP2>HMTHENP2=1
924 Q=INT(RND(9)*HQ)+1
930 QS$=Q$(P1,Q)
931 AS$=A$(P1,Q)
932 Q$(P1,Q)=Q$(P2,Q)
933 A$(P1,Q)=A$(P2,Q)
934 Q$(P2,Q)=QS$
935 A$(P2,Q)=AS$
990 NEXT
999 RETURN
```

FIRST RIGHT GUESS



This is a game for one to five people. Players take it in turns to guess the number the computer is 'thinking' of.

How to play

First type in the names of the players. Now your Sega will indicate whose turn it is. All you do is input a number between 10 and 99. Your Sega will tell you TOO HIGH, TOO LOW or JUST RIGHT and let you know how many guesses it took. A logical thinker will need fewer guesses.

Notes on the program

To avoid any possibility of a player getting used to the sequence, line 205 includes a thoroughly randomising element. J equals the final digit of the time which has elapsed since the computer has been turned on or RESET.

Changes you can make

To make the game harder or easier you can increase or decrease the range of numbers between which the mystery number occurs. You can also adapt the program for the graphics screen, enhancing the interest of the screen display by using different colors for the different messages. Remember, though, that you'll have to clear the screen for each new turn.

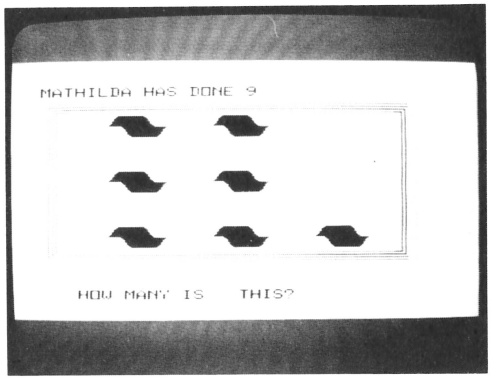
Program listing

```
100 REM INPUT NAMES
110 CLS
115 GOSUB 800
117 PRINT"FIRST RIGHT GUESS"
118 PRINT"================"
119 PRINT
120 PRINT"THIS IS A GAME FOR 1 TO 5 PL
AYERS"
122 PRINT
124 PRINT"THE WINNER IS THE FIRST TO G
UESS THE
126 PRINT"NUMBER THE COMPUTER IS THINK
ING OF
128 PRINT
130 PRINT"(IT'S BETWEEN 11 AND 99)
150 PRINT
152 PRINT"HOW MANY PLAYERS"
154 INPUTHM
156 IFHM<1ORHM>5THEN150
158 GOSUB 800
160 FORF=1TOHM
162 PRINT
164 PRINT"NAME OF PLAYER";F
166 INPUT PL$(F)
168 NEXT
200 REM GET A NUMBER
205 J=VAL(RIGHT$(TIME$,1)):IFJ=0THEN20
5
210 NU=INT(RND(J)*89)+11
220 GOSUB 800
250 TN=INT(RND(7)*HM)+1
300 REM DECIDE TURN
311 FORT=1TO50:NEXT
312 TN=TN+1:IFTN>HMTHENTN=1
315 PRINT:PRINT
```

```
320 PRINT"IT'S ";PL$(TN);"'S TURN:"
321 PRINT"============================

322 CT=CT+1
325 BEEP2
330 PRINT
332 PRINT"WHAT IS MY NUMBER"
334 INPUTGU
336 IF GU<10ORGU>99THEN330
340 GOSUB800
345 PRINT
347 IFGU=NUTHEN370
349 GOSUB820:REM BARP
350 IFGU<NUTHENPRINT"YOUR GUESS IS TOO
 LOW!"
352 IFGU>NUTHENPRINT"YOUR GUESS IS TOO
 HIGH!"
358 PRINT:PRINT
360 PRINT"Press BAR when ready for nex
t turn"
362 A$=INKEY$
363 IFA$<>" "THEN362
364 GOTO 300
370 GOSUB800
372 PRINT
374 PRINT"YES, YOU'VE GOT IT!
375 PRINT
376 PRINTPL$(TN);" is right with";GU
377 PRINT
378 PRINT"- after";CT;" guesses.":CT=0

379 FORT=1TO16:SOUND1,RND(7)*1000+500,
15
380 FORG=1TO40:NEXT:NEXT
381 SOUND0:GOSUB 800:PRINT:PRINT
382 PRINT"Press BAR when ready for nex
t game"
```

```
384 PRINT"Press N for new players
390 A$=INKEY$
392 IFA$<>" "ANDA$<>"N"THEN390
394 GOSUB800
396 IFA$="N"THEN100
398 GOTO200
800 REM GOSUBS
810 FORF=1TO15
812 COLOR1,F
814 NEXT
816 RETURN
820 SOUND1,110,15
822 FORF=1TO34:NEXT
824 SOUND0
826 RETURN
```

COUNTDOWN



This game is especially for little brothers or sisters who would like to use your computer. When you RUN the program, you, the older person, are asked to put in the child's highest counting number (a number between 2 and 9). Then the program asks for the child's name. Some children can type this in themselves, otherwise you do.

How to play

Your Sega displays a set of objects for the child to count and then type in the number. After ten correct answers there are congratulations and the program restarts.

Notes on the program

The large counting objects in this program are constructed of standard graphic characters from the keyboard. Lines 50 to 55 abbreviate the CHR$ codes for the solid blocks to the shorter X$ and Y$. You can see how useful the abbreviations are in lines 420 to 480! Lines 520 to 532 draw four lines to form a rectangle. The four instructions could be replaced by a single instruction ending with ,B. Think about it and give it a try! Line 536 chooses the colors (at random) for the various objects. Lines 544 and 546 choose the screen position (X,Y) for each of the objects — again, at random. Line 548 tests to see whether the position is already filled (in which case P(X,Y) = 1).

Line 405, by the way, contains 38 spaces.

Changes you can make

For children with different spans of attention you can increase or decrease the number of trials before the congratulations are given. For older children you

can increase the number of objects to be counted. For a multiplication exercise you can display the objects in groups. To enhance the interest of the screen display you can use large sprites to create all sorts of interesting things to be counted.

Program listing

```
10 REM COUNTDOWN
15 SCREEN 2,2:CLS
20 COLOR 0,1,(0,0)-(255,191),4
30 FOR X=1 TO 3:C(X)=55*X
35 R(X)=41*X:NEXT
45 REM MUCH USED GRAPHIC CHARACTERS
50 X$=CHR$(229)+CHR$(229):REM ABBREV F
OR 2 BLOCKS
55 Y$=X$+CHR$(229):REM ABBREV FOR 3 BL
OCKS
100 REM SET LIMITS
110 FOR F=1TO15
120 COLOR F,1
130 CURSOR 30,10+F*10:PRINT"WELCOME TO
 COUNTDOWN"
160 X=RND(5)*500+150
170 SOUND 2,X,15:NEXT:SOUND 0
175 FOR F=1 TO 100:NEXT
177 COLOR 15,1
180 FOR F=1 TO 7:READ T$
182 DATA      "ADULT,",TYPE IN,CHILD'S,
HIGHEST,COUNTING,NUMBER,(2 TO 9)
185 CURSOR 200,20+F*10:PRINT T$:NEXT
187 SOUND 2,1000,15
188 FOR F=1 TO 100:NEXT:SOUND 0
190 CURSOR212,130
192 MN$=INKEY$
193 MN=VAL(MN$):IF MN<2 OR MN>9 THEN
192
194 PRINTMN:BEEP2
196 FOR F=1 TO 200:NEXT
```

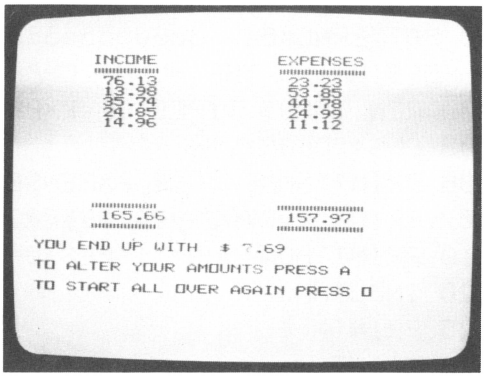
```
200 REM TYPE IN NAME
210 SCREEN 1:COLOR 1,11:CLS:CURSOR 3,3

215 PRINT"TYPE IN YOUR NAME"
217 CURSOR 7,12:PRINT"THEN PRESS CR"
220 CURSOR 6,7:PRINTCHR$(155),CHR$(150
)
221 CURSOR 6,8:PRINTX$
222 CURSOR 6,9:PRINTCHR$(156);CHR$(152
)
231 CURSOR 9,8:INPUT N$
300 REM SELECT A NUMBER
310 PN=INT(RND(8)*MN)+1
400 REM SELECT AN OBJECT
405 CL$="
        ":REM 1 LINE OF SPACES
410 X=INT(RND(7)*5)+1
415 REM YOU CAN TYPE LINES 420-460 USI
NG STRAIGHT GRAPHIC CHARACTERS IF YOU
WISH
420 IF X=1 THEN O$=X$+CL$+CHR$(151)+X$
+CHR$(152)
430 IF X=2 THEN O$=Y$+CL$+Y$
440 IF X=3 THEN O$=CHR$(229)+CHR$(155)
+CHR$(229)+CL$+CHR$(156)+CHR$(229)+CHR
$(156)
450 IF X=4 THEN O$=CHR$(149)+X$+CHR$(1
50)+CL$+CHR$(151)+X$+CHR$(152)
460 IF X=5 THEN O$=CHR$(149)+X$+CHR$(1
50)+CL$+X$
480 REM
500 REM DISPLAY IN RANDOM LOCATIONS
502 SCREEN 2,2:CLS
505 COLOR 1,15,(0,0)-(255,191),12
510 CURSOR 20,20:PRINT N$;" HAS DONE";
NR
515 FOR F=1 TO 5:F2=F*2:COLOR F2
```

```
520 LINE(30-F2,40-F2)-(210+F2,40-F2),
F2
524 LINE(30-F2,40-F2)-(30-F2,140+F2),
F2
528 LINE(30-F2,140+F2)-(210+F2,140+F2)
,F2
532 LINE(210+F2,40-F2)-(210+F2,140+F2)
,F2
534 NEXT
536 CR=(INT(RND(6)*5)+1)*2
538 COLOR CR
540 FOR F=1 TO 3:FOR G=1 TO 3:P(F,G)=0
:NEXT:NEXT
542 FOR F=1 TO PN
544 X=INT(RND(5)*3)+1
546 Y=INT(RND(4)*3)+1
548 IFP(X,Y)THEN 544
552 P(X,Y)=1
556 CURSOR C(X),R(Y):PRINT O$
558 SOUND 2,200,15:FOR G=1TO10:NEXT:
SOUND 0
560 NEXT
600 REM INVITE INPUT
605 COLOR1
610 CURSOR 40,170:PRINT"HOW MANY IS
THIS?"
700 REM GET INPUT
715 A$=INKEY$:IFA$<"1"ORA$>"9"THEN 715

720 A=VAL(A$)
725 IF A>PN THEN GOSUB 800
730 IF A<PN THEN GOSUB 900
735 IF A=PN THEN 1000
740 GOTO 715
800 REM TOO HIGH ROUTINE
810 SOUND 2,3500,15
820 FOR F=1 TO 20:NEXT
```

```
830 SOUND 0:RETURN
900 REM TOO LOW ROUTINE
910 SOUND 2,150,15
920 FOR F=1 TO 20:NEXT
930 SOUND 0:RETURN
1000 REM CORRECT ROUTINE
1010 CURSOR 170,170
1020 PRINT A$;" CORRECT"
1040 FOR F=1 TO 24
1045 X=(RND(4)*1000)+1000
1050 SOUND 2,X,15
1060 NEXT
1070 SOUND 0
1080 FOR F=1 TO 200:NEXT
1100 REM FINISH?
1110 NR=NR+1:IF NR=10 THEN 1200
1120 GOTO 300
1200 REM CONGRATULATIONS
1205 CLS
1210 FOR F=1 TO 15
1220 COLOR F,1
1230 CURSOR 30,10+F*10:PRINT"CONGRATUL
ATIONS,";N$
1240 X=RND(5)*500+150
1250 SOUND 2,X,15:NEXT:SOUND 0
1260 FOR F=1 TO 100:NEXT
1300 REM INVITE ANOTHER GAME
1310 CLS
1320 CURSOR 50,50:PRINT"ANOTHER GAME?"

1330 CURSOR 50,150:PRINT"PRESS ANY
KEY"
1340 A$=INKEY$:IFA$=""THEN 1340
1350 NR=0
1355 RESTORE
1360 GOTO 10
```

HOME BUDGET

This program lets you enter each item of your income for a period and each item of your expenses for the same period. It allows for up to a hundred items of each. Then it lists both the expenses and sources of income by name and shows the totals for both lists. Finally, it tells you whether you will have a credit balance or a debit balance! If you want to, you can then go back over your figures, changing them in the light of what you now know.



How to use the program

When typing in amounts, don't use the dollar sign. All the numbers you type must be in the same units. For example, you could do the whole thing in cents. If you do it in dollars, any amount in cents must come after the decimal point: for example, 50 cents would be .50 (or just .5). When you press A to change your amounts, the names of your items are retained but the old amounts are discarded. Just type in your new amounts.

Notes on the program

Line 50 sets up arrays for income amounts, expenditure amounts, names of income items and names of expenditure items. Line 60 designs our own graphic underlining character. Lines 512 and 514 total the income variables and the expenditure variables.

Changes you can make

You can work out a subroutine to put each amount into a definite column, so that the decimal points are all in a row. You could alter the sequence of this program to total the income first and remember it before you total the expenditure — or vice versa. You could enchance the program by smartening up the alteration procedure, allowing the user to alter only selected items and not have to type in all the amounts again.

Program listing

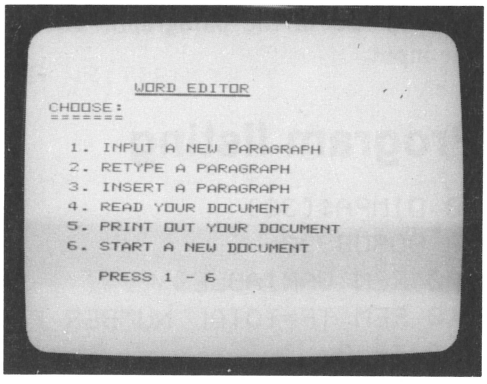
```
50 DIM IN(100), EX(100), IN$(100), EX$
(100)
60 PATTERNC#64,"0000555555550000":REM
RE-DESIGNING THE ALPHA CHARACTER
100 REM INPUT TYPES OF EXPENSE
105 CLS:COLOR1,14
106 PRINT"NAME YOUR EXPENSES"
107 PRINT"@@@@@@@@@@@@@@@@@@@@"
110 PRINT"HOW MANY TYPES OF EXPENSE?"
120 INPUTHM
125 PRINT
126 PRINT
130 FORF=1TOHM:PRINT:PRINT"NAME OF EXP
ENSE";F;":":BEEP
140 INPUTEX$(F):NEXT:BEEP2
150 COLOR1,14
152 CLS
155 PRINT"ADDING EXPENSES"
156 PRINT"@@@@@@@@@@@@@@@@"
160 FORF=1TOHM
162 PRINT`
165 PRINT"AMOUNT SPENT ON ";EX$(F);":"

170 INPUTEX(F)
180 NEXT:BEEP2
190 IF RHTHEN300
200 REM INPUT TYPES OF INCOME
203 PRINT:PRINT
205 CLS:COLOR1,14
206 PRINT"LIST YOUR INCOME"
207 PRINT"@@@@@@@@@@@@@@@@@@"
210 PRINT"HOW MANY TYPES OF INCOME?"
220 INPUTHN
225 PRINT
226 PRINT
```

```
230 FORF=1TOHN:PRINT:PRINT"NAME OF INC
OME";F;":":BEEP
240 INPUTIN$(F):NEXT:BEEP2
300 REM INPUT AMOUNT OF INCOME
302 CLS
304 COLOR1,14
305 PRINT"ADDING INCOME"
306 PRINT"@@@@@@@@@@@@@@"
310 FORF=1TOHN
312 PRINT
315 PRINT"AMOUNT GAINED FROM ";IN$(F);
":"
320 INPUTIN(F)
330 NEXT:BEEP2
500 REM FINANCIAL CONCLUSION
501 CLS
502 PRINT"FINANCIAL CONCLUSION "
503 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@"
512 TI=0:FORF=1TOHN:TI=TI+IN(F):NEXT
514 TE=0:FORF=1TOHM:TE=TE+EX(F):NEXT
520 COLOR1,11
530 PRINT"      INCOME              EXPENS
ES"
531 PRINT"      @@@@@@              @@@@@@
@@"
540 FORF=1TOHN
544 CURSOR5,F+3:PRINTIN(F)
546 NEXT
550 FORF=1TOHM
554 CURSOR22,F+3:PRINTEX(F)
556 NEXT
560 IFHM>HNTHENHO=HM-2
561 IFHN>HMTHENHO=HN-2
562 FORF=1TOHO:PRINT:NEXT
563 PRINT"      @@@@@@              @@@@@@
@@"
564 PRINT"      ";TI
```

```
566 PRINTCHR$(30);CHR$(13);TAB(22);TE
571 PRINT"      @@@@@@              @@@@@@
@@"
580 PRINT
590 PRINT"YOU END UP WITH  $";TI-TE
600 REM ALTER
610 PRINT:RH=0
620 PRINT"TO ALTER YOUR AMOUNTS PRESS
A
622 PRINT
625 PRINT"TO START ALL OVER AGAIN PRES
S O
630 A$=INKEY$
635 IFA$<>"A"ANDA$<>"O"THEN630
637 BEEP2:FORF=1TO100:NEXT:BEEP2
640 IFA$="A"THENRH=1:GOTO 150
650 IFA$="O"THEN100
```

WORD EDITOR

With this program and a Sega printer you can compose a document on screen, rewrite any paragraph of it, insert extra paragraphs, then print it out exactly as you wrote it. You can use it to write letters, make out invitations (without rewriting most of the message), design business documents for special purposes . . . the list is endless.



How to use word editor

Every document is made up of paragraphs. A paragraph can have up to five and a half lines of typing, or it can be no more than a dotted line. You space out your words on the screen as you want them to appear on paper. The printer will faithfully reproduce your layout.

Use the SPACE BAR for all gaps within your paragraph. To make an address on the right-hand side of your letter, for example, simply fill the unused part of each line with spaces. In that way you can write the whole address as one paragraph. It is important not to use quotes (") within a paragraph as this will signal end-of-message. If you must have quotes, use *single quotes* (SHIFT 7). Avoid using graphic characters as the Sega printer will not reproduce them.

Notes on the program

The FOR. . .NEXT loop beginning in line 1060 is the key to the problem of inserting a paragraph between two existing paragraphs. This backward loop goes from the final paragraph number (TP), back to the number to be inserted (NN). In line 1065 we start with a new paragraph number (F + 1) and let this take the contents of the old paragraph (F) — and so on down to the paragraph NN + 1 taking the old paragraph NN, thus leaving room for our new paragraph.

Changes you can make

To save the user starting every paragraph with right cursor, you could change the input mechanism onto an INKEY$ basis and add the quotes yourself. The quotes are needed in case any punctuation such as comma (,) or semi-colon (;) is contained in the paragraph. These symbols would otherwise break up the input.

Program listing

```
50 DIMPA$(30)
70 GOSUB900
100 REM VARIABLES
110 REM TP=TOTAL NUMBER PARAGRAPHS
112 TP=0
180 GOSUB800
190 GOTO300:REM MENU
200 REM INPUT PARAGRAPH
202 CLS
205 COLOR1,15
220 PRINT"NOW TYPE IN PARAGRAPH";TP+1
222 PRINT"--------------------------"

224 PRINT"press right arrow to begin
225 PRINT"press CR when paragraph comp
leted
226 PRINT:PRINT
227 PRINT"
        ";CHR$(34)
228 FORF=1TO2:PRINTCHR$(30); NEXT
230 INPUT"
        ";PA$(TP+1)
235 TP=TP+1
240 PRINT:PRINT:PRINT
242 PRINT"Press BAR for another paragr
aph
244 PRINT"or press M for menu.
254 A$=INKEY$
```

```
256 IFA$<>" "ANDA$<>"M"THEN254
258 BEEP
260 IFA$=" "THEN200
262 IFA$="M"THEN300
264 GOTO254
290 PRINT:PRINT:PRINT
292 PRINT"          NOTHING WRITTEN YET!

297 SOUND1,110,15
298 FORF=1TO500:NEXT
299 SOUND0
300 REM MENU
305 COLOR1,11
310 CLS
320 PRINT:PRINT
330 PRINT"CHOOSE:
332 PRINT"=======
334 PRINT:PRINT
336 PRINT"  1. INPUT A NEW PARAGRAPH
338 PRINT
340 PRINT"  2. RETYPE A PARAGRAPH
341 PRINT
342 PRINT"  3. INSERT A PARAGRAPH
343 PRINT
344 PRINT"  4. READ YOUR DOCUMENT
346 PRINT
347 PRINT"  5. PRINT OUT YOUR DOCUMENT

348 PRINT
349 PRINT"  6. START A NEW DOCUMENT
350 PRINT:PRINT
352 PRINT"      PRESS 1 - 6
354 A$=INKEY$
356 A=VAL(A$)
358 IFA<1ORA>6THEN354
359 IFTPORA=1ORA=6THEN361
360 GOTO290
```

```
361 GOSUB 800:COLOR1,15
362 ONAGOTO200,500,1000,600,700,100
500 REM CHANGE  PARAGRAPH
505 IS$="NEW ":REM NEW FLAG
510 CLS
515 PRINT
520 PRINT"Which paragraph will you re-
write
522 PRINT"(1 -";TP;")"
523 PRINT
524 INPUTNP
526 IFNP<1ORNP>TPTHEN529
528 GOTO530
529 PRINT"NUMBER ERROR":GOTO 515
530 CLS:PRINT"HERE IS THE OLD PARAGRAP
H";NP;":"
531 PRINT"--------------------------

532 PRINT:PRINTPA$(NP)
533 PRINT:PRINT
534 PRINT"NOW TYPE IN ";IS$;"PARAGRAPH
";NP
535 IS$="":REM KILL INSERT FLAG
536 PRINT"------------------------"
540 PRINT"press right arrow to begin
542 PRINT:PRINT
544 PRINT"
        ";CHR$(34)
546 FORF=1TO2:PRINTCHR$(30);:NEXT
548 INPUT"
        ";PA$(NP)
550 PRINT:PRINT:PRINT
552 PRINT"Press BAR to re-write this p
aragraph
554 PRINT"or press M for menu.
556 A$=INKEY$
558 IFA$<>" "ANDA$<>"M"THEN556
```

```
560 BEEP
562 IFA$=" "THEN530
566 IFA$="M"THEN300
600 REM READ DOCUMENT
610 CLS
620 PRINT"Press BAR for each paragraph

622 PRINT
624 PRINT"(paragraph numbers won't be
printed)
630 FORF=0TOTP
632 IFFTHENPRINT"par";F
640 PRINTPA$(F)
642 A$=INKEY$
644 IFA$<>" "THEN642
646 BEEP
648 NEXT
649 PRINT:PRINT:PRINT:GOSUB800
650 PRINT"Press BAR to re-read
652 PRINT"or M for menu
653 FORH=1TO50:NEXT
654 A$=INKEY$
656 IFA$<>" "ANDA$<>"M"THEN654
658 BEEP
660 IFA$=" "THEN600
662 IFA$="M"THEN300
700 REM PRINT OUT
705 CLS
706 PRINT
707 PRINT"Document directed to printer
.
710 LPRINT
715 PRINT:PRINT
720 FORF=1TOTP
722 BEEP
725 PRINT"printing paragraph";F
730 LPRINTPA$(F)
```

```
740 LPRINT
750 NEXT
755 PRINT
756 PRINT"PRINTING COMPLETED"
760 PRINT:PRINT:PRINT
762 GOSUB800
764 PRINT"Press BAR to print again"
765 PRINT"or press M for menu"
766 A$=INKEY$
768 IFA$=" "THEN700
770 IFA$="M"THEN300
774 GOTO766
800 REM SOUND
810 FORF=1TO8
820 R=INT(RND(8)*500)+300
822 SOUND1,R,15
823 NEXT
824 SOUND0
828 RETURN
900 REM INSTRUCTIONS
905 COLOR1,15
910 CLS
919 PRINT
920 PRINT"WITH THIS PROGRAM,
921 PRINT
922 PRINT"YOU CAN COMPOSE
923 PRINT
924 PRINT"AND PRINT OUT
925 PRINT
926 PRINT"ANY DOCUMENT
927 PRINT
928 PRINT"UP TO 30 PARAGRAPHS LONG.
929 PRINT:PRINT
930 PRINT"(MAXIMUM 5 LINES PER PARAGRA
PH)
931 PRINT:PRINT"(USE ' FOR QUOTES - NO
T ";CHR$(34);")"
```

```
932 PRINT:PRINT"Make sure your printer
 is connected
933 PRINT"and switched on..."
934 GOSUB800
935 PRINT:PRINT:PRINT:PRINT
936 PRINT"Press BAR to continue..."
940 IFINKEY$<>" "THEN940
950 RETURN
1000 REM INSERT PARAGRAPH
1005 IS$="NEW ":REM INSERT FLAG
1010 CLS
1020 PRINT"You already have paragraphs
 1 to";TP;"."
1022 PRINT
1024 PRINT"BEFORE WHICH PARAGRAPH (GIV
E NUMBER)
1025 PRINT
1026 PRINT"DO YOU WANT TO INSERT A NEW
 PARAGRAPH?
1028 PRINT:PRINT:PRINT
1032 PRINT"To see document again, type
 S.
1033 PRINT
1034 PRINT"Otherwise TYPE NUMBER.
1035 PRINT
1036 PRINT"Then press CR
1037 PRINT:PRINT:PRINT
1043 INPUTA$
1044 IFVAL(A$)=0THEN600
1050 NN=VAL(A$)
1060 FORF=TPTONNSTEP-1
1065 PA$(F+1)=PA$(F)
1070 NEXT
1072 TP=TP+1
1075 NP=NN
1077 CLS
1078 GOSUB800
1079 GOTO533
```

21 Fabulous Programs for your Sega SC3000 Computer is designed to provide enjoyment for people of all ages and interests. The ready-made programs range from the simple **First right guess** to the sophisticated **Mini-checkers**. There are challenging action games (like **Springnet**) and useful 'home-office' facilities (like **Word editor**). And there are your old favourites – **Hangman**, **Mastermind** and **Tic-Tac-Toe** – all in original versions. All you have to do is key them into your computer.

All of the programs are ready to use right away. Notes on **How to use** the program accompany each of the program listings. And there is also advice for people who can handle a little BASIC programming and who want to adapt the programs.

PITMAN
ISBN 0 85896 210 1