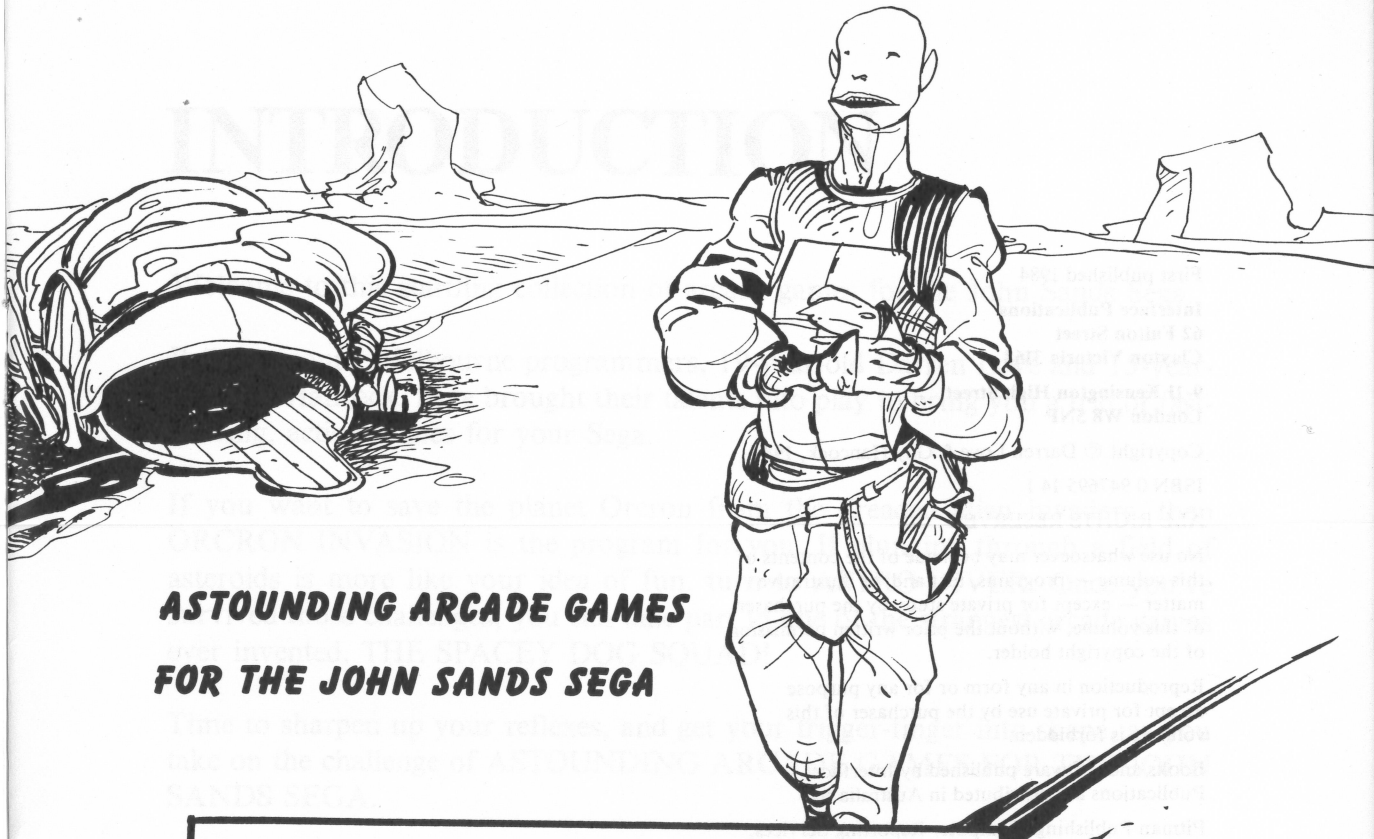
ASTOUNDING ARCADE GAMES FOR THE JOHN SANDS SEGA

DARREN LOVE & GUY HANCOCK



ASTOUNDING ARCADE GAMES

FOR THE JOHN SANDS SEGA

CONTENTS

Introduction 3

Orcron Invasion 5

Astro-Diver 11

The Spacey Dog Squad 19

Write Your Own Games 26

INTERFACE PUBLICATIONS

London and Melbourne

First published 1984

Interface Publications
62 Fulton Street
Clayton Victoria 3168

9-11 Kensington High Street
London W8 5NP

ISBN 0 947695 14 1

Books and software published by Interface Publications are distributed in Australia by:

Pitman Publishing/Computer Reporting Services, Level 12, Town Hall House, 452 Kent Street, Sydney, 2000; (02) 267 1066

And in the UK by:

WHS Distributors, St John's House, East Street, Leicester, LE1 6NE (0533) 551 1196

Any enquiries regarding the contents of this volume should be directed to Interface Publications.

Printed in Australia
by Impact Printing Pty Ltd

INTRODUCTION

Welcome to this startling collection of arcade games for the John Sands Sega.

Brilliant young Melbourne programmers, 19-year-old Darren Love and 15-year-old Guy Hancock, have brought their talents into play to bring you a trio of fast-moving, action games for your Sega.

If you want to save the planet Orcron from the dreaded alien invaders, then ORCRON INVASION is the program for you. If plunging through a field of asteroids is more like your idea of fun, turn to ASTRO-DIVER. Once you've survived those challenges, you can take part in one of the strangest arcade games ever invented, THE SPACEY DOG SQUAD!

Time to sharpen up your reflexes, and get your trigger-finger into shape, as you take on the challenge of ASTOUNDING ARCADE GAMES FOR THE JOHN SANDS SEGA.

Tim Hartnell,
Melbourne, 1984

Tim Hartnell, head of Interface Publications, is one of the world's most widely-published computer book authors. Recent works include EXPLORING ARTIFICIAL INTELLIGENCE ON YOUR COMPUTER and THE BIG FAT BOOK OF COMPUTER GAMES. Both are available from the store where you bought this booklet.

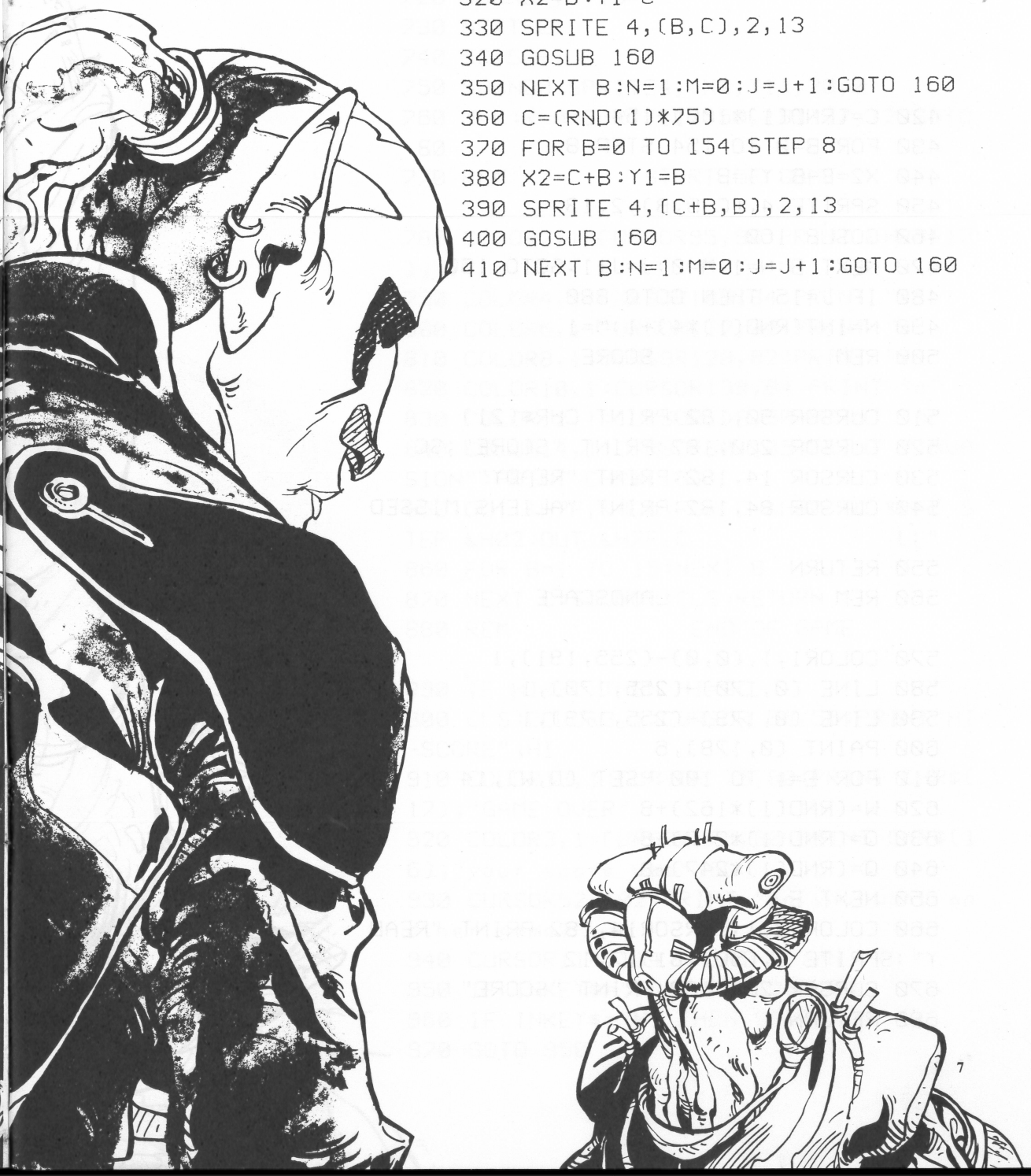
ORCRON INVASION

Orcron Invasion is a fast-moving game for those of you with an eye for accuracy. You are cast upon Orcron's lunar crust in a brave attempt to save Orcron from the evil aliens of the planet "Ectomorphia".

They can swoop horizontally and diagonally past you, although there is only one on the screen at a time. You use the cursor keys to move left and right and key "A" to fire. If fifteen "Ectomorphians" manage to land, they will overpower you and the game will be over.

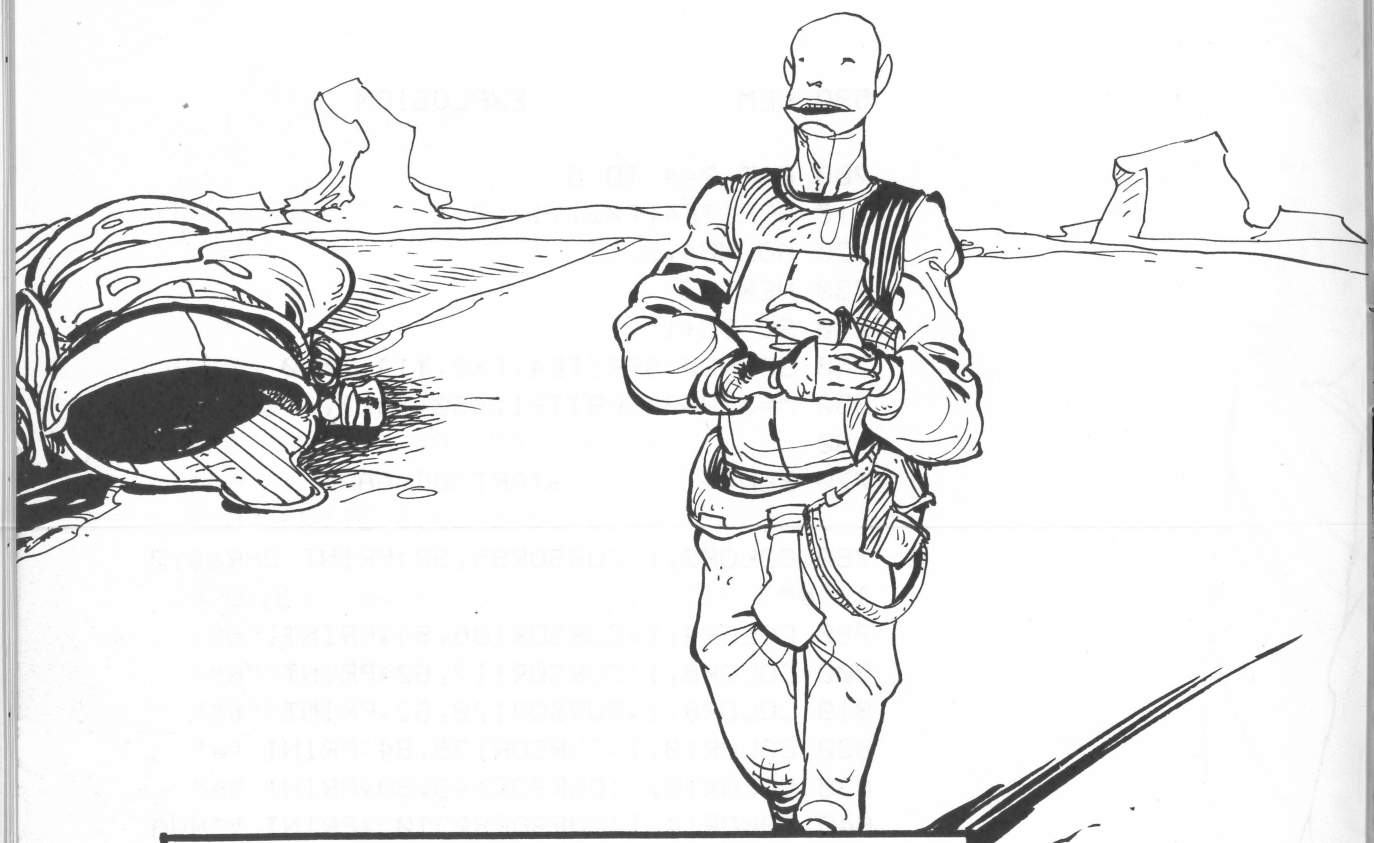
```
5 HI=0
10 SCREEN 2,2:CLS
20 X=120:M=0:J=0:SC=0

10 HI=0
20 SCREEN 2,2:CLS
30 X=120:M=0:J=0:SC=0
40 GOSUB 780
50 GOSUB 990
60 GOSUB 570
70 REM                  MOVEMENT

80 IF  INKEY$=CHR$(29) THEN X=X-10
90 IF  INKEY$=CHR$(28) THEN X=X+10
100 IF R=1 THEN GOTO 120
110 IF  INKEY$="A" THEN R=1:Y=153:X1=X

120 IF M=1 THEN GOTO 150
130 GOSUB 480
140 ON N GOTO  240,300,360,420
150 RETURN
160 SPRITE 0,(X,161),1,4
170 IF R=0  THEN GOTO 220
180 SPRITE 1,(X1,Y),3,5
190 IF X1-7<=X2 AND X1+7>=X2 AND Y>=Y1
-5 AND Y<=Y1+5 THEN GOTO 700
200 Y=Y-8
210 IF Y<=3 THEN R=0:SPRITE 1,(50,181)
,7,12:GOTO 220
220 GOTO 80
230 REM         ALIEN'S MOVEMENT

240 C=RND(1)*154
250 FOR B=0 TO 255 STEP 8
260 X2=B:Y1=C
270 SPRITE 4,(B,C),2,13
280 GOSUB 160
```

```
290 NEXT B:N=1:M=0:J=J+1:GOTO 160
300 C=RND(1)*154
310 FOR B=255 TO 0 STEP -8
320 X2=B:Y1=C
330 SPRITE 4,(B,C),2,13
340 GOSUB 160
350 NEXT B:N=1:M=0:J=J+1:GOTO 160
360 C=(RND(1)*75)
370 FOR B=0 TO 154 STEP 8
380 X2=C+B:Y1=B
390 SPRITE 4,(C+B,B),2,13
400 GOSUB 160
410 NEXT B:N=1:M=0:J=J+1:GOTO 160
```



```
420 C=(RND(1)*100)+154
430 FOR B=0 TO 154 STEP 8
440 X2=C-B:Y1=B
450 SPRITE 4,(C-B,B),2,13
460 GOSUB 160
470 NEXT B:N=1:M=0:J=J+1:GOTO 160
480 IF J=15 THEN GOTO 880
490 N=INT(RND(1)*4)+1:M=1
500 REM              SCORE

510 CURSOR 90,182:PRINT CHR$(21)
520 CURSOR 200,182:PRINT "SCORE";SC
530 CURSOR 14,182:PRINT "READY"
540 CURSOR 84,182:PRINT "ALIENS MISSED
";J
550 RETURN
560 REM            LANDSCAPE

570 COLOR1,1,(0,0)-(255,191),1
580 LINE (0,170)-(255,170),1
590 LINE (0,179)-(255,179),1
600 PAINT (0,178),6
610 FOR E=1 TO 100:PSET (Q,W),14
620 W=(RND(1)*162)+8
630 Q=(RND(1)*247)+8
640 Q=(RND(1)*247)+8
650 NEXT E
660 COLOR10,1:CURSOR14,182:PRINT "READ
Y":SPRITE 1,(50,181),7,12
670 CURSOR 200,182:PRINT "SCORE"
680 RETURN
```

```
690 REM            EXPLOSION

700 FOR Z=4 TO 6
710 SPRITE4,(X2,Y1),Z,Z
720 SOUND4,2,15
730 NEXT Z
740 SC=SC+1
750 SOUND0:SPRITE4,(X2,Y1),20,0
760 M=0:R=0:SPRITE1,(50,181),7,12:GOTO
 80
770 REM          START OF GAME

780 COLOR2,1:CURSOR95,90:PRINT CHR$(17
);"o"
790 COLOR4,1:CURSOR106,84:PRINT "r"
800 COLOR6,1:CURSOR117,82:PRINT "c"
810 COLOR8,1:CURSOR128,82:PRINT "r"
820 COLOR10,1:CURSOR139,84:PRINT "o"
830 COLOR12,1:CURSOR149,90:PRINT "n"
840 COLOR14,1:CURSOR80,105:PRINT "INVA
SION":PRINT CHR$(16)
850 OUT &H7F,&HD0:FOR C=&HB0 TO &HFF S
TEP &H02:OUT &H7F,C
860 FOR B=1 TO 15:NEXT B
870 NEXT C:SOUND0:CLS:RETURN
880 REM              END OF GAME

890 IF HI<SC THEN HI=SC
900 CLS:COLOR3,1:CURSOR 96,55:PRINT"HI
-SCORE";HI
910 COLOR13,1:CURSOR 75,80:PRINT CHR$(
17);"GAME OVER"
920 COLOR3,1:CURSOR 75,90:PRINT CHR$(1
6);"your score was";SC
930 CURSOR52,102:PRINT "do you want an
other game?"
940 CURSOR114,114:PRINT "(Y/N)"
950 IF INKEY$="Y" THEN GOTO 20
960 IF INKEY$="N" THEN STOP
970 GOTO 950
```

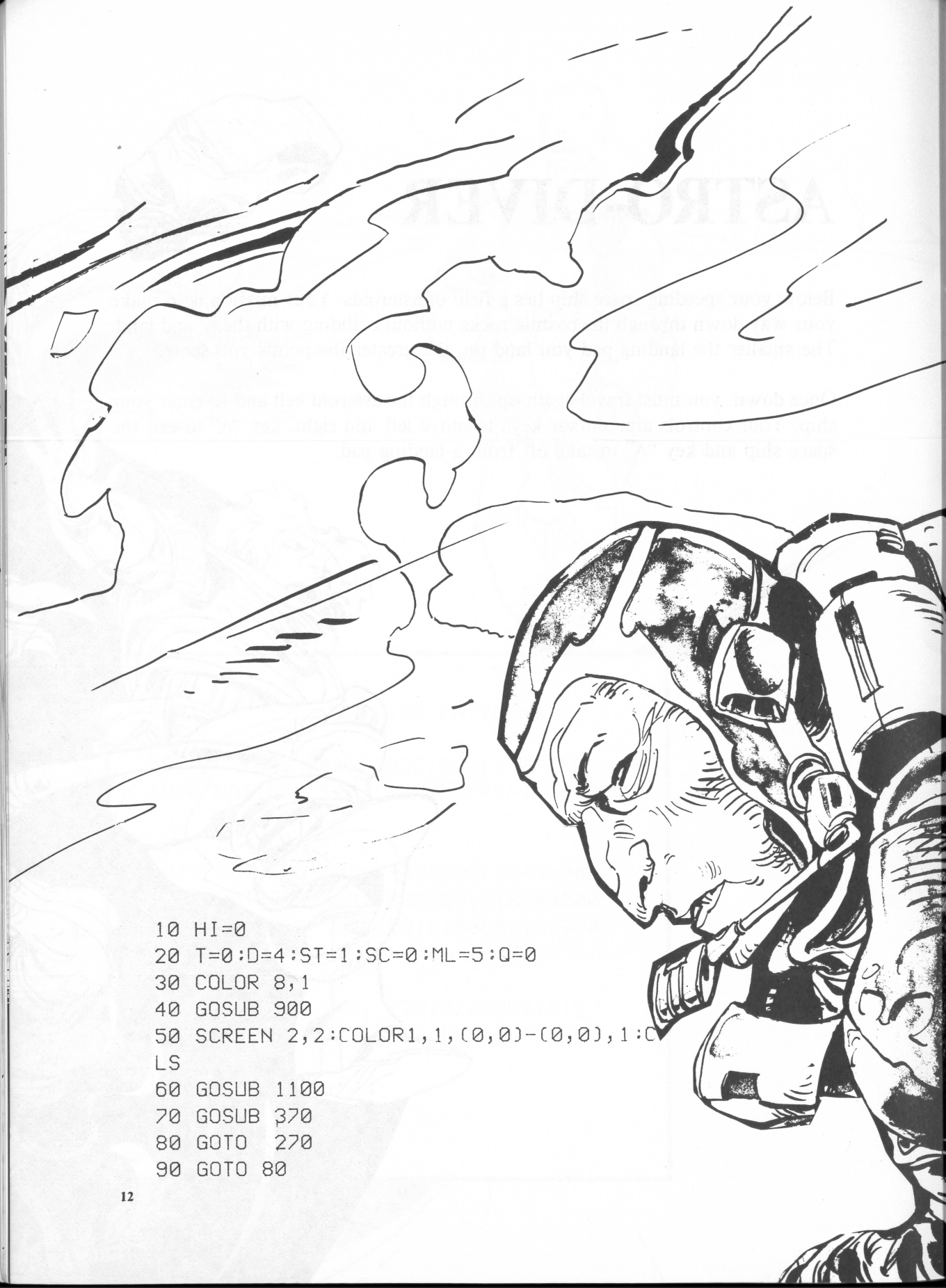


```
980 REM          SPRITE PATTERNS

990 PATTERN S#1,"183C183C187EFFFF"
1000 PATTERN S#2,"247EDBFF5A426600"


1010 PATTERN S#3,"183C3C3C1844104A"
1020 PATTERN S#4,"0000241818240000"
1030 PATTERN S#5,"0018186666181800"


1040 PATTERN S#6,"E7818100008181E7"
1050 PATTERN S#7,"FFFFFFFFFFFFFF00"

1060 RETURN
```

ASTRO-DIVER

Below your speeding space ship lies a field of asteroids. Your mission is to make your way down through the cosmic rocks without colliding with them, and land. The smaller the landing pad you land on, the greater the points you score.

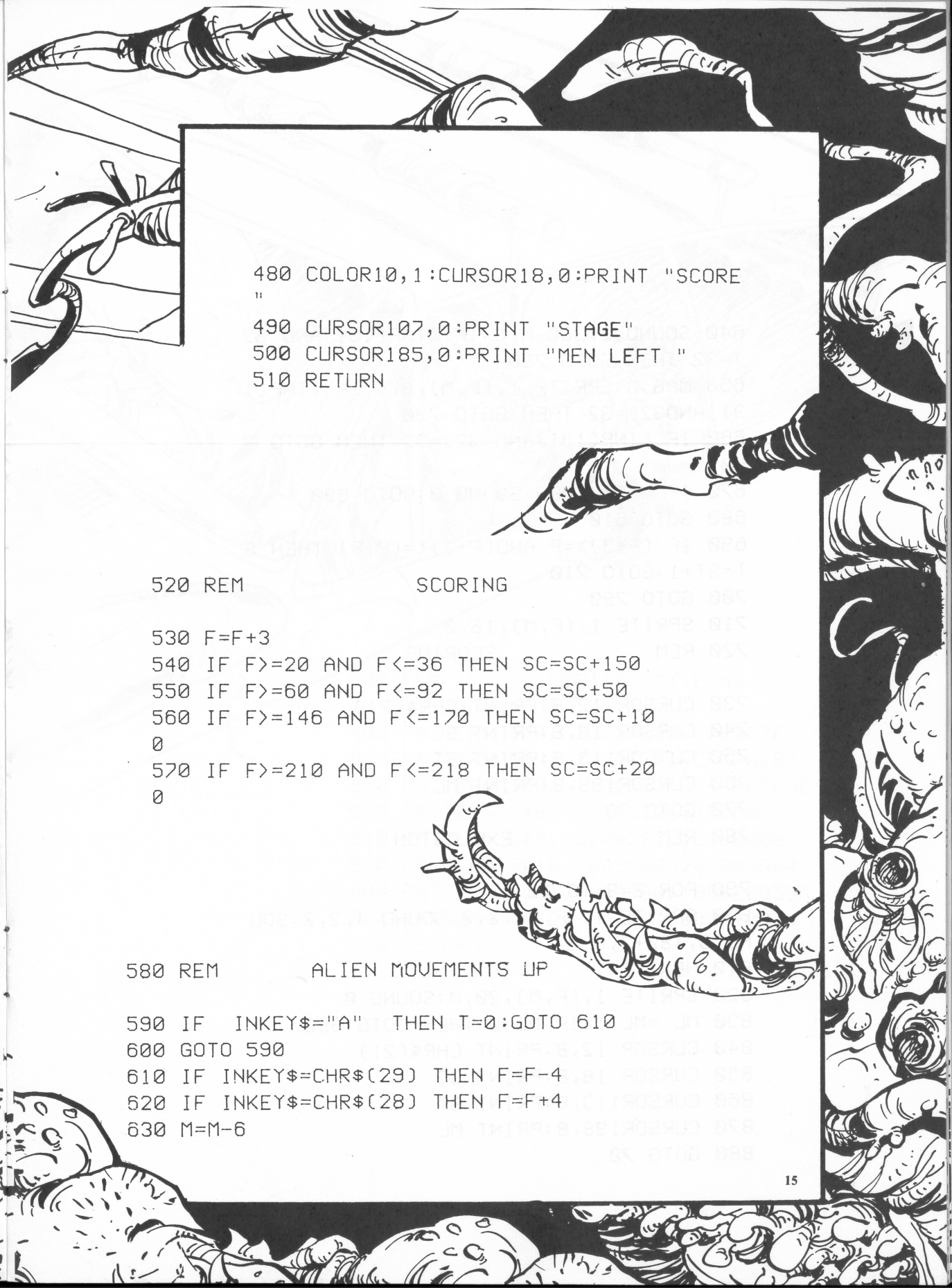
Once down, you must travel again up through the asteroid belt and re-enter your ship. Your controls are: cursor keys to move left and right, key "A" to exit the space ship and key "A" to take off from a landing pad.



```
10 HI=0
20 T=0:D=4:ST=1:SC=0:ML=5:Q=0
30 COLOR 8,1
40 GOSUB 900
50 SCREEN 2,2:COLOR1,1,(0,0)-(0,0),1:C
LS
60 GOSUB 1100
70 GOSUB 370
80 GOTO  270
90 GOTO 80
```

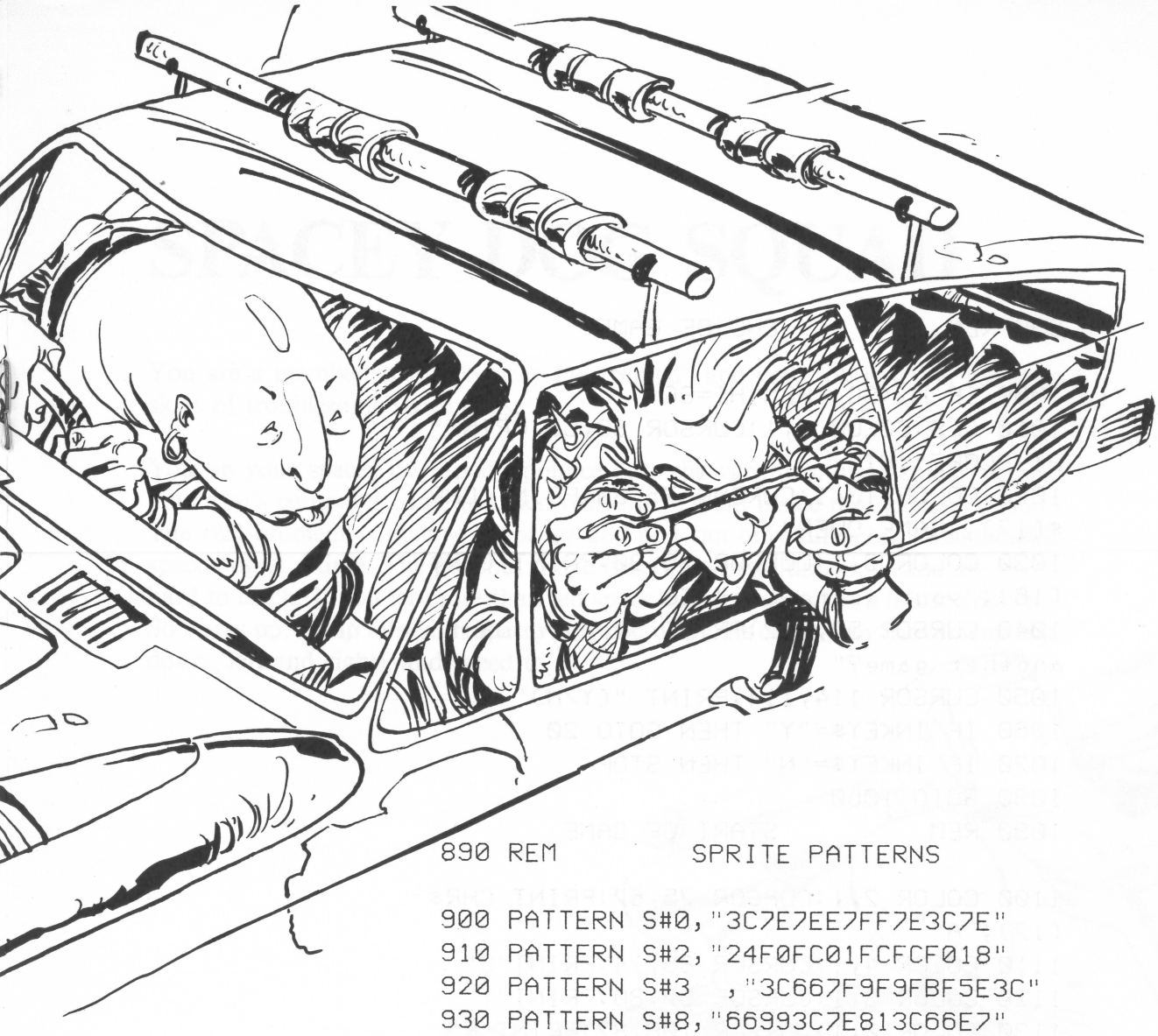
```
100 REM        ALIEN MOVEMENTS DOWN

110 IF Q=1 THEN GOTO 150
120 IF INKEY$="A" THEN Q=1:F=P:M=26
130 IF T=0 THEN GOTO 300
140 IF T=3 THEN GOTO 340
150 IF INKEY$=CHR$(29) THEN F=F-4
160 IF INKEY$=CHR$(28) THEN F=F+4
170 SOUND 1,500-M,7
180 MAG0:SPRITE1,(F,M),8,4:IF (INP(191
)AND 32)=32  THEN GOTO 790
190 IF (INP(191)AND 32)=32 THEN GOTO 7
90
```

```
200 IF M>=176THEN SOUND 0:GOTO  240
210 M=M+6
220 IF T=0 THEN GOTO 300
230 IF T=3 THEN GOTO 340
240 IF VPEEK(INT(188/8)*256+INT((F+3)/
8)*8+188 MOD8+8192)=129 THEN GOTO 520
250 GOTO 790
260 REM       SPACE POD'S MOVEMENT

270 SOUND 4,3,8:T=0:FOR P=8 TO 247 STE
P 8
280 MAG0:SPRITE0,(P,16),0,5
290 GOTO 110
300 NEXT P
310 T=3:FOR P=247 TO 8 STEP -8
320 MAG0:SPRITE0,(P,16),0,5
330 GOTO 110
340 NEXT P
350 GOTO 270
360 REM                 LANDSCAPE

370 FOR Z=3 TO 8+INT(1.5*ST):SPRITE Z,
(0,0),90,0
380 FOR Z=3 TO 8+INT(1.5*ST):Q=INT(RND
(9)*247)+1:W=INT(RND(9)*150)+25
390 SPRITE Z,(Q,W),3,6:NEXT Z
400 LINE (20,188)-(36,188),8
410 LINE (20,190)-(36,190),8
420 LINE (60,188)-(92,188),8
430 LINE (60,190)-(92,190),8
440 LINE (146,188)-(170,188),8
450 LINE (146,190)-(170,190),8
460 LINE (210,188)-(218,188),8
470 LINE (210,190)-(218,190),8
```

```
480 COLOR10,1:CURSOR18,0:PRINT "SCORE
"
490 CURSOR107,0:PRINT "STAGE"
500 CURSOR185,0:PRINT "MEN LEFT "
510 RETURN

520 REM              SCORING

530 F=F+3
540 IF F>=20 AND F<=36 THEN SC=SC+150
550 IF F>=60 AND F<=92 THEN SC=SC+50
560 IF F>=146 AND F<=170 THEN SC=SC+10
0
570 IF F>=210 AND F<=218 THEN SC=SC+20
0

580 REM       ALIEN MOVEMENTS UP

590 IF  INKEY$="A"  THEN T=0:GOTO 610
600 GOTO 590
610 IF INKEY$=CHR$(29) THEN F=F-4
620 IF INKEY$=CHR$(28) THEN F=F+4
630 M=M-6
```

```
640 SOUND 2,360-M,6:IF (INP(191)AND 32
)=32 THEN GOTO 790
650 MAG 0:SPRITE 1,(F,M),8,4:IF (INP(1
91)AND32)=32 THEN GOTO 790
660 IF (INP(191)AND 32)=32 THEN GOTO 7
90
670 IF M=26 THEN SOUND 0:GOTO 690
680 GOTO 610
690 IF (F+3)>=P AND(F+3)<=(P+8) THEN S
T=ST+1:GOTO 710
700 GOTO 790
710 SPRITE 1,(F,M),18,0
720 REM              SCORING

730 CURSOR 12,8:PRINT CHR$(21)
740 CURSOR 18,8:PRINT SC
750 CURSOR113,8:PRINT ST
760 CURSOR198,8:PRINT ML
770 GOTO 70
780 REM             EXPLOSION

790 FOR Z=8 TO 12
800 SPRITE 1,(F,M),Z,Z:SOUND 4,2,Z:SOU
ND 3,180+Z,6
810 NEXT Z
820 SPRITE 1,(F,M),20,0:SOUND 0
830 ML =ML-1:IF ML=0 THEN GOTO 990
840 CURSOR 12,8:PRINT CHR$(21)
850 CURSOR 18,8:PRINT SC
860 CURSOR113,8:PRINT ST
870 CURSOR198,8:PRINT ML
880 GOTO 70
```



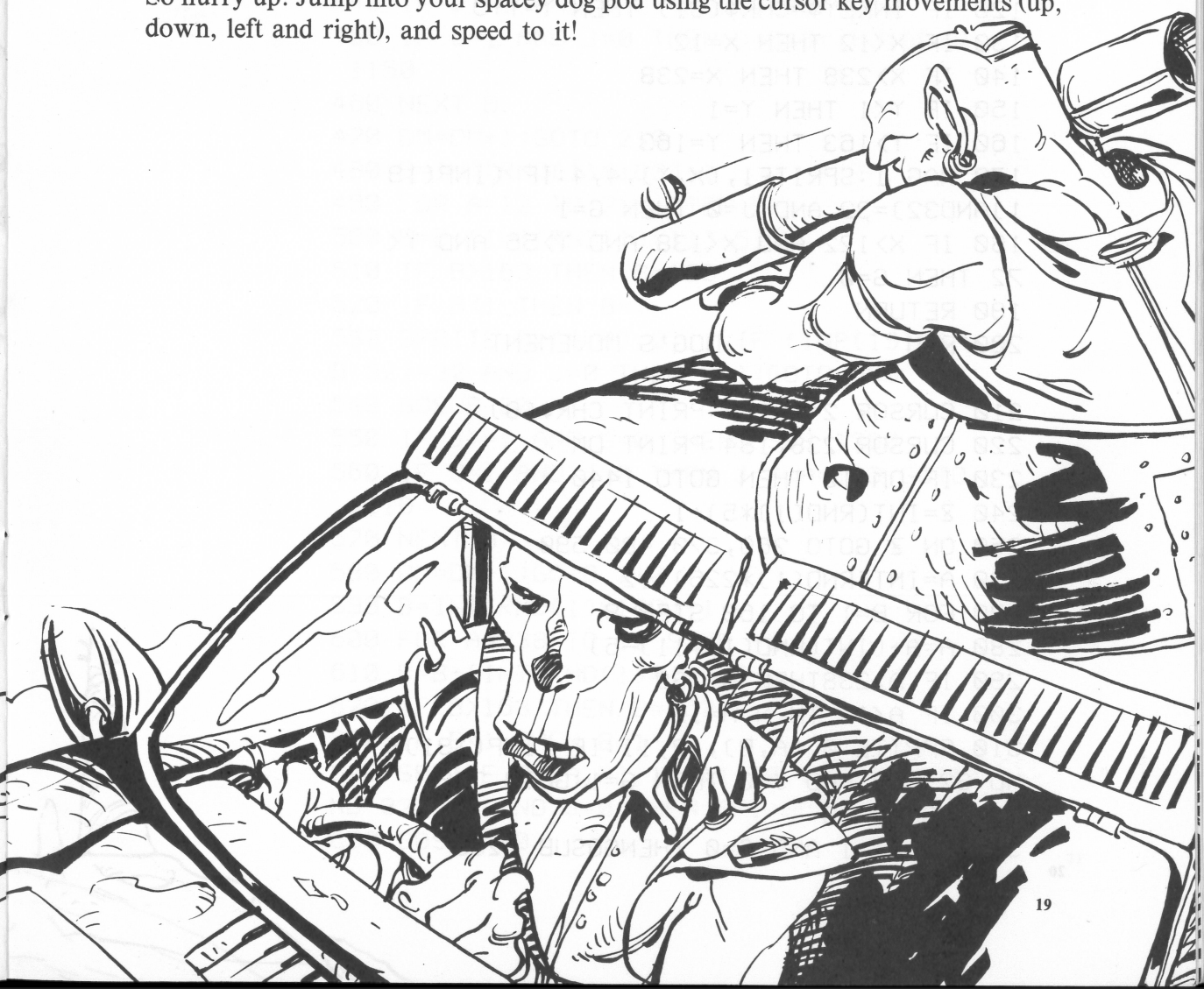
```
890 REM          SPRITE PATTERNS

900 PATTERN S#0,"3C7E7EE7FF7E3C7E"
910 PATTERN S#2,"24F0FC01FCFCF018"
920 PATTERN S#3  ,"3C667F9F9FBF5E3C"
930 PATTERN S#8,"66993C7E813C66E7"
940 PATTERN S#9,"0000283810280000"
950 PATTERN S#10,"005A004242005A00"
960 PATTERN S#11,"AA01801188018055"
970 PATTERN S#12,"2400810000810024"
980 RETURN
```

```
990 REM          END OF GAME

1000 IF HI<SC THEN HI=SC
1010 CLS:COLOR 3,1:CURSOR 90,55:PRINT"
HI-SCORE";HI
1020 COLOR 13,1:CURSOR 75,80:PRINT CHR
$(17);"GAME OVER"
1030 COLOR 3,1:CURSOR 75,90:PRINT CHR$
(16);"your score was";SC
1040 CURSOR 52,102:PRINT "do you want
another game?"
1050 CURSOR 114,114:PRINT "(Y/N)"
1060 IF INKEY$="Y" THEN GOTO 20
1070 IF INKEY$="N" THEN STOP
1080 GOTO 1060
1090 REM          START OF GAME

1100 COLOR 2,1:CURSOR 75,67:PRINT CHR$
(17);"A"
1110 COLOR 4,1:CURSOR 86,74:PRINT"S"
1120 COLOR 8,1:CURSOR 97,80:PRINT"T"
1130 COLOR10,1:CURSOR 108,82:PRINT"R"
1140 COLOR12,1:CURSOR 119,84:PRINT"0"
1150 COLOR3,1:CURSOR 129,90:PRINT"D"
1160 COLOR7,1:CURSOR 134,99:PRINT"I"
1170 COLOR15,1:CURSOR139,111:PRINT"V"
1180 COLOR13,1:CURSOR143,123:PRINT"E"
1190 COLOR14,1:CURSOR146,136:PRINT"R":
PRINT CHR$(16)
1200 OUT &H7F,&HF0:FOR C=&HD0 TO  &HFF
 STEP &H04:OUT&H7F,C
1210 FOR B=1 TO 15:NEXT B
1220 NEXT C:SOUND0
1230 CLS
1240 RETURN
```

SPACEY DOG SQUAD

You are a member of the "Spacey Dog Squad" and your job is to rid the solar skies of troublesome space dogs.

You, in your spacey dog "pod", pounce onto the varmints, disassemble them in your pod's computer bank and then reassemble them inside the "galactic pound". The real problem is that your spacey dog pod can only carry one disassembled space dog at a time. The pooches move randomly in all directions and are pretty hard to catch. If you let more than ten space dogs get away, you'll lose your job. So hurry up! Jump into your spacey dog pod using the cursor key movements (up, down, left and right), and speed to it!

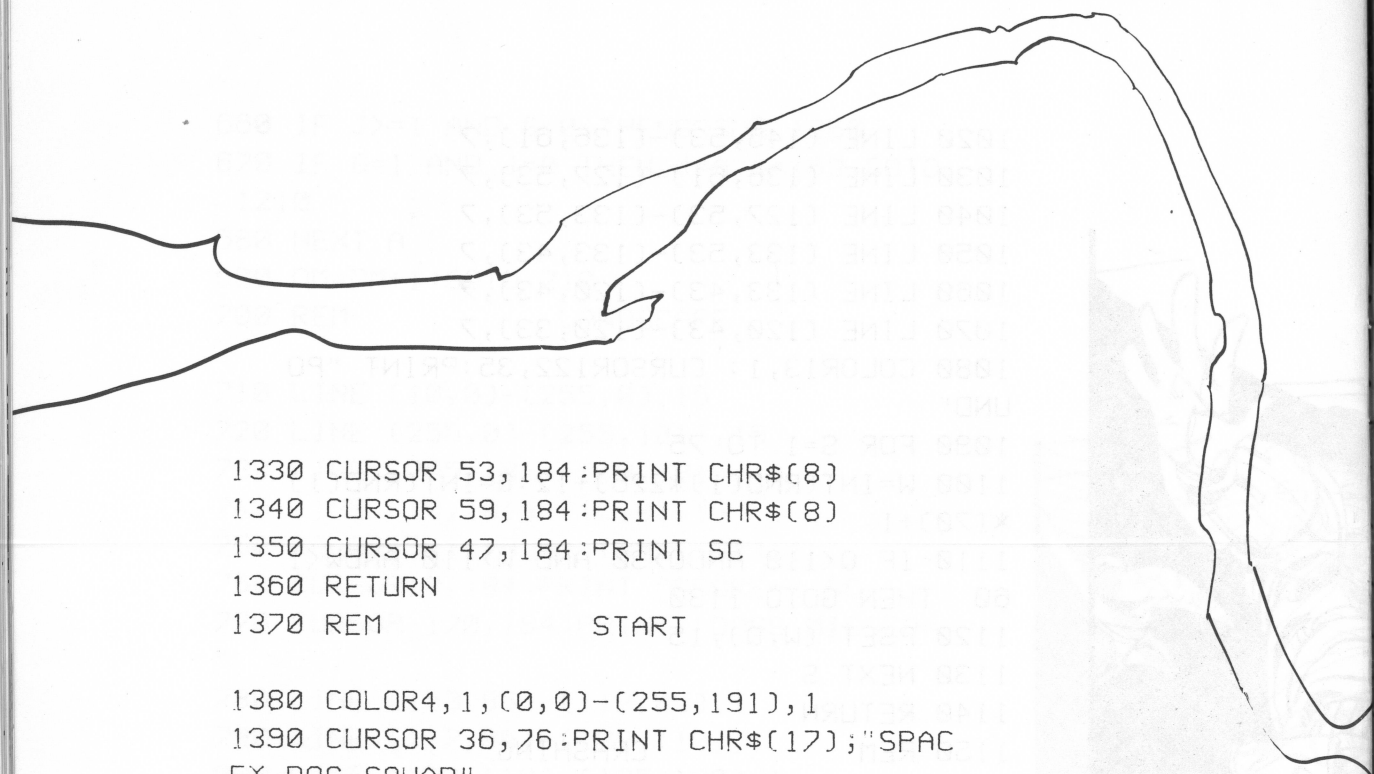


```
10 HI =0
20 X=128:Y=66:DM=0:SC=0
30 SCREEN 2,2:CLS
40 GOSUB 1370
50 GOSUB 1540
60 GOSUB 700
70 GOTO 210
80 REM              POD'S MOVEMENT
90 IF INKEY$=CHR$(28) THEN X=X+9
100 IF INKEY$=CHR$(29) THEN X=X-9
110 IF INKEY$=CHR$(30) THEN Y=Y-9
120 IF INKEY$=CHR$(31) THEN Y=Y+9
130 IF X<12 THEN X=12
140 IF X>238 THEN X=238
150 IF Y<1 THEN Y=1
160 IF Y>163 THEN Y=163
170 MAG 1:SPRITE1,(X,Y),4,4:IF (INP(19
1)AND32)=32 AND J=0 THEN G=1
180 IF X>122 AND X<138 AND Y>56 AND Y<
72 THEN G=0
190 RETURN
200 REM              DOG'S MOVEMENT

210 CURSOR 244,184:PRINT CHR$(8)
220 CURSOR 238,184:PRINT DM
230 IF DM=10 THEN GOTO 1440
240 Z=INT(RND(1)*5)+1
250 ON Z GOTO 260,370,480,590
260 A=INT(RND(1)*226)+12
270 FOR B=1 TO 163 STEP 8
280 A=A+(INT(RND(1)*11)-5)
290 IF A>238THEN A=238
300 IF A<12 THEN A=12
310 SPRITE 0,(A,B),12,11:IF (INP(191)A
ND 32)=32 AND J=0 THEN G=1:GOTO 340
320 GOSUB 80
330 IF J>=1 AND G=0 THENGOSUB 1260
```

```
340 IF G=1 AND J=0 THEN J=B:B=163:GOTO
 1150
350 NEXT B
360 DM=DM+1:GOTO 210
370 A=INT(RND(1)*226)+12
380 FOR B=163 TO 1 STEP -8
390 A=A+(INT(RND(1)*11)-5)
400 IF A>238 THEN A=238
410 IF A<12 THEN A=12
420 SPRITE 0,(A,B),2,11:IF (INP(191)AN
D 32)=32 AND J=0 THEN G=1:GOTO 450
430 GOSUB 80
440 IF J>=1 AND G=0 THENGOSUB 1260
450 IF G=1 AND J=0 THEN J=B:B=163:GOTO
 1150
460 NEXT B
470 DM=DM+1:GOTO 210
480 B=INT(RND(1)*163)+1
490 FOR A=12 TO 238 STEP 8
500 B=B+(INT(RND(1)*11)-5)
510 IF B>163 THEN B=163
520 IF B<1 THEN B=1
530 SPRITE 0,(A,B),2,11:IF (INP(191)AN
D 32)=32 AND J=0 THEN G=1:GOTO 560
540 GOSUB 80
550 IF J>=1 AND G=0 THENGOSUB 1260
560 IF G=1 AND J=0 THEN J=A:A=163:GOTO
 1210
570 NEXT A
580 DM=DM+1:GOTO 210
590 B=INT(RND(1)*163)+1
600 FOR A=238 TO 12 STEP -8
610 B=B+(INT(RND(1)*11)-5)
620 IF B>163 THEN B=163
630 IF B<1 THEN B=1
640 SPRITE 0,(A,B),12,11:IF (INP(191)A
ND 32)=32 AND J=0 THEN G=1:GOTO 670
650 GOSUB 80
```

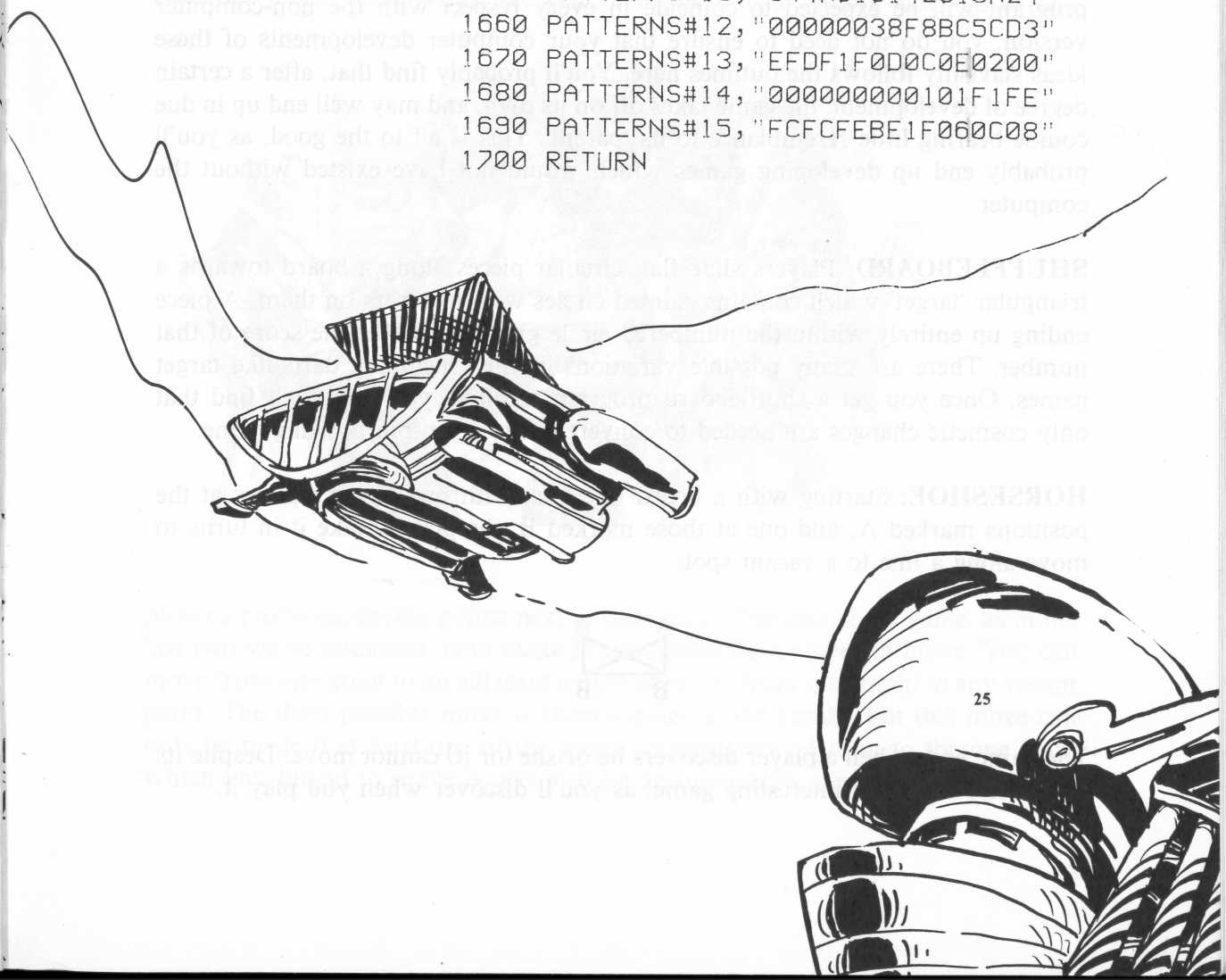
```
660 IF J>=1 AND G=0 THENGOSUB 1260
670 IF G=1 AND J=0 THEN J=A:A=163:GOTO
 1210
680 NEXT A
690 DM=DM+1:GOTO 210
700 REM            LANDSCAPE

710 LINE (10,0)-(255,0),15
720 LINE (255,0)-(255,181),15
730 LINE (255,181)-(10,181),15
740 LINE (10,181)-(10,0),15
750 COLOR15,1,(0,0)-(255,191),1
760 CURSOR10,184:PRINT "SCORE=";SC
770 CURSOR 170,184:PRINT "DOGS MISSED=
"
780 LINE (153,85)-(117,85),1
790 LINE (117,86)-(122,110),1
800 LINE (122,110)-(125,100),1
810 LINE (125,100)-(127,103),1
820 LINE (127,103)-(130,95),1
830 LINE (130,95)-(132,107),1
840 LINE (132,107)-(135,102),1
850 LINE (135,102)-(137,97),1
860 LINE (137,97)-(140,104),1
870 LINE (140,104)-(143,100),1
880 LINE (143,100)-(148,108),1
890 LINE (148,108)-(155,85),1
900 PAINT (121,105),6
910 CIRCLE (135,85),20,3,0.2,0,1,BF
920 COLOR1,14,(130,66)-(140,81),1
930 LINE (131,68)-(131,78),1
940 LINE (134,68)-(134,78),1
950 LINE (137,68)-(137,78),1
960 LINE (140,68)-(140,78),1
970 LINE (120,33)-(152,33),7
980 LINE (152,33)-(152,43),7
990 LINE (152,43)-(139,43),7
1000 LINE (139,43)-(139,53),7
1010 LINE (139,53)-(145,53),7
```

```
1020 LINE (145,53)-(136,61),7
1030 LINE (136,61)-(127,53),7
1040 LINE (127,53)-(133,53),7
1050 LINE (133,53)-(133,43),7
1060 LINE (133,43)-(120,43),7
1070 LINE (120,43)-(120,33),7
1080 COLOR13,1: CURSOR122,35:PRINT "PO
UND"
1090 FOR S=1 TO 75
1100 W=INT(RND(1)*226)+12:Q=INT(RND(1)
*170)+1
1110 IF Q<110 ANDQ>30 AND W>110 ANDW<1
60  THEN GOTO 1130
1120 PSET (W,Q),15
1130 NEXT S
1140 RETURN
1150 REM              CRASHING

1160 FOR D=8 TO 10:SPRITE 1,(X,Y),0,0
1170 MAG 2:SPRITE0,(A,J),D,15:SOUND1,1
30+-D,15
1180 NEXT D
1190 SOUND0
1200 GOTO 210
1210 FOR D=8 TO 10:SPRITE 1,(X,Y),0,0
1220 MAG 2:SPRITE0,(J,B),D,15:SOUND1,1
30+-D,15
1230 NEXT D
1240 SOUND0
1250 GOTO 210
1260 REM              CRASHING

1270 SPRITE 1,(X,Y),4,0
1280 FOR D=8 TO 10
1290 MAG 2:SPRITE0,(X,Y),D,15:SOUND1,1
30+-D,15
1300 NEXT D
1310 SOUND0:G=0:J=0
1320 SC=SC+1
```



```
1330 CURSOR 53,184:PRINT CHR$(8)
1340 CURSOR 59,184:PRINT CHR$(8)
1350 CURSOR 47,184:PRINT SC
1360 RETURN
1370 REM            START

1380 COLOR4,1,(0,0)-(255,191),1
1390 CURSOR 36,76:PRINT CHR$(17);"SPAC
EY DOG SQUAD"
1400 PRINT CHR$(16)
1410 FOR V=1 TO 252 STEP 5:SOUND 5,(IN
T(RND(1)*2)+1),11
1420 PSET (V,80+(RND(1)*20)-10),15:NEX
T V:CLS
1430 SOUND0:RETURN
1440 REM              END

1450 IF HI<SC THEN HI=SC
1460 CLS:COLOR3,1:CURSOR 98,58:PRINT "
HI-SCORE";HI
1470 COLOR13,1:CURSOR75,80:PRINT CHR$(
17);"GAME OVER"
1480 COLOR3,1:CURSOR 75,90:PRINT CHR$(
16);"you caught";SC;" dogs"
1490 CURSOR 52,102:PRINT "do you want
another game?"
1500 CURSOR114,114:PRINT "(Y/N)"
1510 IF INKEY$="Y" THEN GOTO 20
1520 IF INKEY$="N" THEN STOP
1530 GOTO 1510
```

```
1540 REM              PATTERNS

1550 PATTERNS#0,"0000000080809F7F"
1560 PATTERNS#1,"3F3F7F7DF8603010"
1570 PATTERNS#2,"0000001C1F3D3BCA"
1580 PATTERNS#3,"F7FBF8B030704000"
1590 PATTERNS#4,"010F1E1F3F3F7F7F"
1600 PATTERNS#5,"7FFFC9FF7F3F1F00"
1610 PATTERNS#6,"80F03818CCCCC6E6"
1620 PATTERNS#7,"E6FF27FFFEFCF800"
1630 PATTERNS#8,"0018242424241800"
1640 PATTERNS#9,"0834244284422C10"
1650 PATTERNS#10,"344A818142815A24"
1660 PATTERNS#12,"00000038F8BC5CD3"
1670 PATTERNS#13,"EFDF1F0D0C0E0200"
1680 PATTERNS#14,"000000000101F1FE"
1690 PATTERNS#15,"FCFCFEBE1F060C08"
1700 RETURN
```



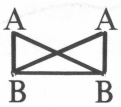
WRITE YOUR OWN GAMES

You may find, after you've been programming for some time, that although you've got many of the skills you need to write games of your own, you're a little short on ideas. If that's the case, you should find this section of the book of interest to you. I'm going to outline a number of games which seem to me to be ideal for conversion to games for the computer.

Unless you're converting a well-known game, such as Checkers, when your program will be expected to coincide in every respect with the non-computer version, you do not need to ensure that your computer developments of these ideas slavishly follows the outlines here. You'll probably find that, after a certain degree of development, the game takes off on its own, and may well end up in due course bearing little resemblance to its 'parent'. This is all to the good, as you'll probably end up developing games which would not have existed without the computer.

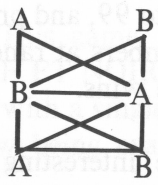
SHUFFLEBOARD. Players slide flat, circular pieces along a board towards a triangular 'target' which contains painted circles with numbers on them. A piece ending up entirely within the numbered circle gives the player the score of that number. There are many possible variations of this, including darts-like target games. Once you get a shuffleboard program running, you may well find that only cosmetic changes are needed to convert it into a ten-pin bowling game.

HORSESHOE: Starting with a board like this, with one player's pieces at the positions marked A, and one at those marked B, the players take it in turns to move along a line to a vacant spot:

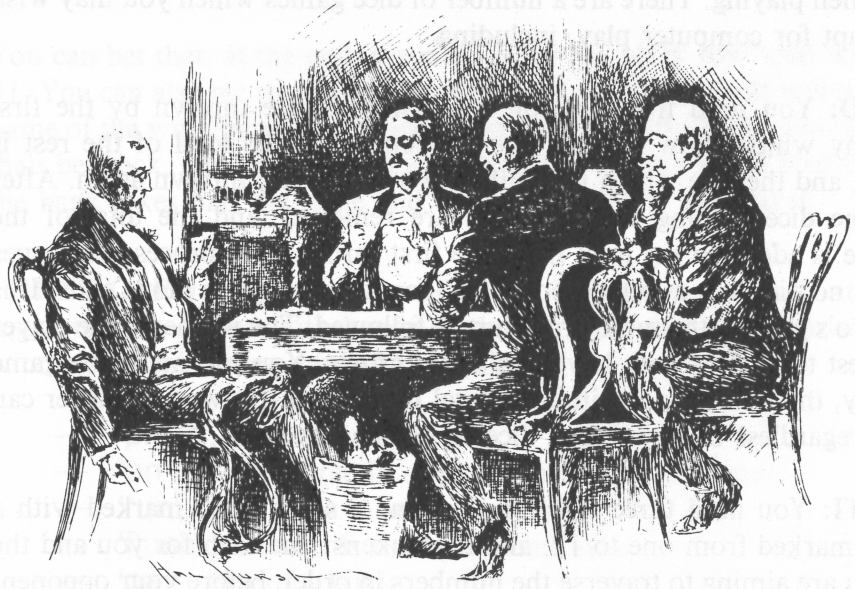


The game ends when a player discovers he or she (or it) cannot move. Despite its simplicity, this is an interesting game, as you'll discover when you play it.

DAMOCLES: This game is like HORSESHOE, but uses a larger board, and three pieces per player. The aim of the game is the same as HORSESHOE, and uses this board:



MU-TORERE: This game, which originated among the Maoris in New Zealand, is played on a board shaped like an eight-pointed star, with each point of the star joined to a circle in the middle. This circle is called the **putahi.** The players start with four pieces each, which are placed on the outside points, with all of one



player's pieces occupying points next to each other. The aim of the game, as in the last two we've discussed, is to make it impossible for a player to move. You can move from one point to an adjacent empty point, or from the putahi to any vacant point. The third possible move is from a point to the putahi, but this move can only be made if at least one of the points immediately adjacent to the one from which you intend to move is occupied by an opponent's piece.

BINGO: This game is usually played with cards marked with numbers, which are crossed off as numbers are drawn at random from a pool, the winner being the player who first crosses off five in a row on their card, in any direction. It can be greatly simplified for computer use, with the computer generating two 'cards' at random, using numbers from zero to 99, and printing them on a five by five grid. The computer can then choose numbers at random from zero to 99, and do the crossings off, and look for possible wins.

The program can be made more interesting to look at by, instead of just generating a random number, including two 'dials' with spinning arrows. The first dial, the 'tens' is marked from zero to nine, and this is spun first. The second dial is marked the same as the first one, except that it represents the 'ones'.

DICE GAMES: You should find dice games relatively easy to program on your computer, especially if the computer is not expected to exhibit too much intelligence when playing. There are a number of dice games which you may wish to try and adapt for computer play, including:

DROP DEAD: You need five dice for this. The dice are thrown by the first player, and any which show two or five are removed. The total of the rest is written down, and the dice which fell with a two or five are thrown again. After this throw, the dice landing two and five are removed, and the total of the remaining dice is added to the total from the first throw. This process continues until there are no dice left, at which point that particular player's move ends. It is then player two's go, and the same procedure is followed. The winner is the player with the biggest total after both have had their throws. You can make the game the best of, say, three rounds, or limit the total number of throws any player can have per go, regardless of how many dice he or she has left.

MARTINETTI: You need three dice for this, and a score sheet marked with a row of boxes marked from one to 12, and two tokens, one each for you and the computer. You are aiming to traverse the numbers in order, before your opponent manages to do so. Each player takes it in turn to throw the dice. If the throw contains a one, the player's token can be placed in the one square. If the first throw also contained a two, the token could be moved into the two. After box one, players can either take the pip shown on an individual die for a move, or can add together the pips showing to get a desired number. A variation of this game (often called EVEREST) is to 'climb down' through the numbers, after one through to 12 has been achieved. The winner of EVEREST is the first player getting back to base camp on one.

CARD GAMES: Once you've worked out a routine to get the computer hold, shuffle and deal the cards you have the raw bones of a host of games. Any book of card games will give you more ideas than you can possibly cope with. Here are a few to start you off:

TRENTE AND QUARANTE: This game, which originated in 17th century Europe, is also known as ROUGE ET NOIR. Six decks of cards are used in the casino version, but it works well with a single deck. The cards are dealt into two rows, with the total of each row being examined after each row has had an additional card.

Court cards (jack, queen, king) count as ten, with all other cards (including the ace) counting as their face value. The moment the total of a row equals or exceeds 31, no more cards are added to that row. However, the dealer keeps adding to the other row until it, too, equals or exceeds 31. The first row is called 'black', regardless of the suits of the cards involved, and the second row is 'red'.

You can bet that, at the end, the red row or the black row total will be closer to 31. You can also bet that the color of the **first** card to be dealt will match the color name of the winning row. All bets are made against the bank, and winners receive their bet back, plus that amount again (odds of one to one). If both rows equal 31, the bank takes half the stake and returns the rest to the player.

THREE CARD BRAG: This game was a forerunner of poker, and as its name suggests, it is played with hands of three, rather than five, cards. Here are the winning combinations, ranked from highest to lowest:

— 'pryle', three of a kind;
— 'on a bike', three in sequence from the same suit;
— 'run', three cards in sequence;
— 'flush', three cards from the same suit;
— 'pair', two of a kind;
— 'high card'

A normal deck is used, and aces rank high, except for 'on a bike' and 'run', when 3, 2, ace beats ace, king, queen.

ACE-DEUCE-JACK: This is a simple game, heavily loaded in the dealer's favor, in which players bet on the likelihood that the next card which appears will be an ace, a two (a 'deuce') or a jack.

FURTHER READING

I make a point of reading any book on games I can get my hands on. It doesn't matter whether the book is about computer games or the old-fashioned 'play against another human' type. So long as it is a book of games, I want to see it.

Games books are superb as idea-starters. Read about a game and you've got the seed of a program. And it doesn't matter if the program doesn't end up playing precisely the game you read about. Many great computer games started their lives as an attempt to computerise a more traditional game, and somehow got side-tracked.

I hope some of the following list of books prove as valuable to you as they have to me.

* **DISCOVERING OLD BOARD GAMES** — R C Bell (Shire Publications, Aylesbury, UK, 1980)

* **DISCOVERING DICE AND DOMINOES** — R C Bell (Shire Publications, Aylesbury, UK, 1980)

* **DICE GAMES NEW AND OLD** — William E Tredd (The Oleander Press, London and New York, 1981)

* **GAME PLAYING WITH BASIC** — Donald D Spencer (Hayden Book Co., Inc., Rochelle Park, New Jersey, 1977)

* **THE TURING CRITERION — MACHINE INTELLIGENT PROGRAMS FOR THE 16K ZX81** — Graham Charlton, Mark Harrison & Dilwyn Jones, ed. Tim Hartnell (Interface Publications, London and Melbourne, 1982)

* **GAMES ANCIENT AND ORIENTAL AND HOW TO PLAY THEM** — Edward Falkener (Dover Publications, Inc., New York, 1961 — a reprint of an original work published by Longmans, Green and Co. in 1892)

* **BASIC COMPUTER GAMES** — Ed. David H Ahl (Creative Computing Press, Morristown, New Jersey, 1978)

* **MORE BASIC COMPUTER GAMES** — Ed. David H Ahl (Creative Computing Press, Morristown, New Jersey, 1979)

* **26 BASIC PROGRAMS FOR YOUR MICRO** — Derrick Daines (Newnes Technical Books, Butterworths & Co., London and Boston, 1982)

* **TABLE GAMES OF GEORGIAN AND VICTORIAN DAYS** — F R B Whitehouse (Priority Press, Royston, Herts., UK, 1971)

* **PET FUN AND GAMES** — Ron Jeffries and Glen Fisher (Osborne/McGraw-Hill, Berkeley, California, 1981)

* **COMPUTER GAMES FOR BUSINESSES, SCHOOLS AND HOMES** — J Victor Nahigian and William S Hodges (Winthrop Publishers, Inc., Cambridge, Massachusetts, 1979)

* **GAMES YOU MAKE AND PLAY** — Pia Hsiao, Neil Lorimer and Nick Williams (Macdonald and Jane's, London, 1975)

* **WHAT TO DO AFTER YOU HIT RETURN** — People's Computer Company (Hayden Book Co., Inc., Rochelle Park, New Jersey, 1980)

* **EVERYMAN'S INDOOR GAMES** — Gyles Brandreth (Dent, Everyman's Library, London, Melbourne and Toronto, 1982)

* **A PLAYER'S GUIDE TO TABLE GAMES** — John Jackson (Stackpole Books, Harrisburg, Pennsylvania, 1975)

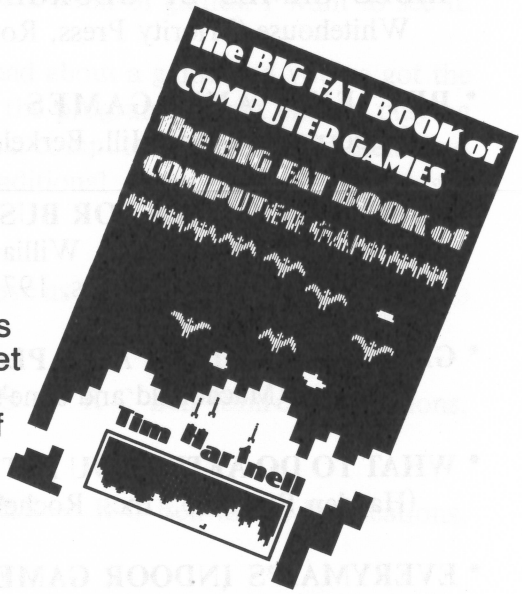
* **GAMES FOR TWO** — John Wasley (Proteus, London and New York, 1981)

* **THE ADDISON-WESLEY BOOK OF APPLE COMPUTER SOFTWARE 1982** — Jeffrey Stanton and John Dickey, editors (The Book Company, Lawndale, California, 1982)

* **DICING WITH DRAGONS** — Ian Livingstone (Routledge & Kegan Paul, London, Melbourne and Henley, 1982)

* **FANTASY ROLE PLAYING GAMES** — J Eric Holmes (Hippocrene Books Inc., New York, 1981)

The Big Fat Book of Computer Games!



Here's a massive collection of games programs which you'll find easy to get up and running on your John Sands Sega. Written by Tim Hartnell, one of the world's most widely-published authors of popular computer books, the games are all new, original, and lots of fun.

No matter what kind of games you're interested in, you'll find them in this 400-page collection. Board games, adventures, space games, brain teasers and simulations – they're all here, just waiting to explode inside your Sega.

Programs in this giant collection include:

Wall Street Trader; Quest for Riemannian; Voyages of Magellan; Starbase 2000; Gettysburg Dice; Magician's Maze; Tryst with Mephistopheles; Detective Q; Cube of Eleusis; 3-D Tic Tac Toe; Go Fish and Heyday Seven.

There are more than 30 magnificent games for your Sega in this collection – all tested, and all guaranteed to be great fun to run.

The Big Fat Book of Computer Games is published by Interface Publications, and distributed in Australia and New Zealand by Pitman Publishing.

Welcome to this startling collection of arcade games for the John Sands Sega.

Brilliant young Melbourne programmers, 19-year-old Darren Love and 15-year-old Guy Hancock, have brought their talents into play to bring you a trio of fast-moving, action games for your Sega.

If you want to save the planet Orcron from the dreaded alien invaders, then ORCRON INVASION is the program for you. If plunging through a field of asteroids is more like your idea of fun, turn to ASTRO-DIVER. Once you've survived those challenges, you can take part in one of the strangest arcade games ever invented, THE SPACEY DOG SQUAD!

Time to sharpen up your reflexes, and get your trigger-finger into shape, as you take on the challenge of ASTOUNDING ARCADE GAMES FOR THE JOHN SANDS SEGA.

Another great book from INTERFACE

ISBN 0 947695 14 1