

**THE  
MICRO  
USER**

No. 1 £1.95

*Games  
Special*

**14**  
**ORIGINAL  
GAMES**

**Arcade Action**

Thrill packed games  
to test even the  
veteran player

**Old Time  
Favourites**

Traditional games  
from bowling green,  
pub and fairground

**Brainteasers**

Challenging and  
entertaining games  
to strain your brain

**PLUS**

Games of strategy,  
skill – and survival!

*All with full-length  
listings and detailed  
explanations*

**All these games are  
also available on a  
REAL cassette.  
See our bargain  
offer on Page 73**

**Golden  
Games**  
*for the BBC Micro*



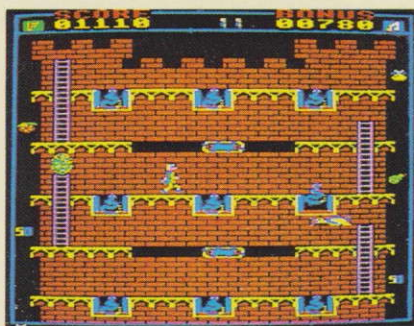
# GAMES

FOR THE BBC 32K O.S. 1.2

Q\*MAN



CASTLE ASSAULT



Q\*MAN'S BROTHER



NIGHTMARE MAZE



SCREWBALL



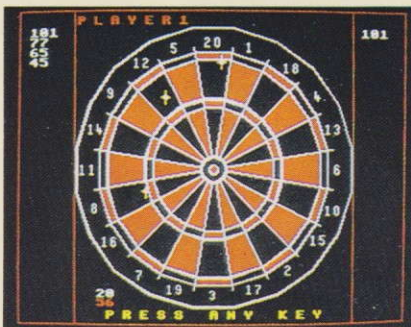
CASTLE OF GEMS



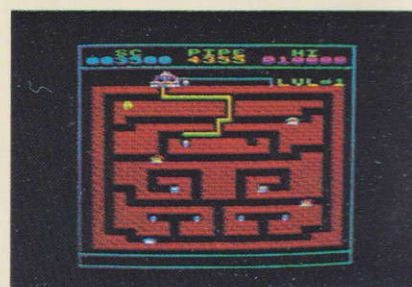
3-D MUNCHY



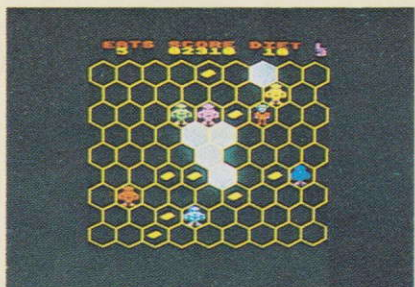
DARTS



DIAMOND MINE



GUY IN THE HAT



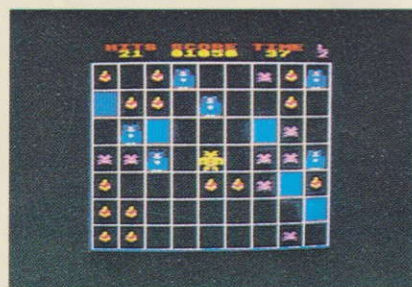
ARTIST

£6.95



(ALSO AVAILABLE FOR ELECTRON)

BANANA MAN



**ALL GAMES ONLY**  
**£5.70** FULLY INCLUSIVE



17 Cross Coates Road, Grimsby, South Humberside. Tel: 0472 44304.



**Managing Editor**  
**Derek Meakin**

**Features Editor**  
**Mike Bibby**

**Production Editor**  
**Peter Glover**

**Layout Design**  
**Heather Sheldrick**

**Associate Editor**  
**Alan McLachlan**

**Advertisement Manager**  
**John Riding**

**Advertising Sales**  
**John Snowden**

**Editor in Chief,**  
**Database Publications**  
**Peter Brameld**

Tel: 061-456 8383 (Editorial)  
061-456 8500 (Advertising)  
061-480 0171 (Subscriptions)  
Telex: 667664 SHARETG  
Prestel Mailbox: 614568383

**Published by:**  
**Database Publications Ltd,**  
**Europa House, 68 Chester Road,**  
**Hazel Grove, Stockport SK7 5NY.**

**ABC** Member of Audit  
Bureau of Circulations

The Micro User welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a stamped, self-addressed envelope, otherwise the return of material cannot be guaranteed. Contributions accepted for publication will be on an all-rights basis.

© 1984 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles, listings or advertisements.

*The Micro User is an independent publication and neither the BBC nor Acorn Computers Ltd are responsible for any of the articles in this issue or for any of the opinions expressed.*

News trade distribution:  
Europress Sales and Distribution Limited,  
11 Brighton Road, Crawley, West Sussex  
RH10 6AF. Circulation 0293 27053.

## **SPECIAL OFFER!**

**All the programs in this issue are available on tape or disc – and it's the best value ever in educational software.**

**– See coupon on Page 73**

## **CONTENTS**

### **4 Explorer Eddie**

Trapped in the deepest recesses of a monster-infested mine, you have to battle your way upwards through 12 levels. Superb graphics and cunningly constructed screens set new standards.

### **10 Masterspy**

Based on Mastermind, this intriguing brainteaser is set in the deadly world of counter espionage.

### **14 Draw**

Brilliant simulation of draw poker guarantees hours of enjoyment for would-be Mavericks.

### **22 China Drop**

A baby-sitting program might seem the ideal solution for insomniacs, but not when the baby is two-year-old Tobermory, intent on destroying his parents' precious china...

### **28 White Widow**

There aren't fairies at the bottom of your garden, but there is Arachnida. This aged spider needs your help in capturing the fleas she needs for food.

### **34 Bill Bounces Back**

Continuing the saga of our intrepid Spaceman Bill. This time he's trying to make his fortune from the teapots of the planet Neegus. Of course he hadn't reckoned on the cloudhoppers...

### **41 Software Guide**

Comprehensive survey of all the latest games software for the BBC Micro will let you find exactly the program you want.

### **45 Splat**

A cat and mouse maze game for all ages, Splat's simple layout hides a compulsive, action-packed game.

### **47 Fair Play**

All the fun of the fair in this simulation of a shooting gallery. To achieve a respectable score you'll have to earn your ammunition – and that's no simple matter.

### **50 Crown Green**

A vivid reconstruction of the gentle game of crown green bowling. This program captures all the grace and skill of the original.

### **54 Sevens**

Two player simulation of the standard card game. It needs skill and downright cunning if you are to rid yourself of all your cards before your opponent.

### **60 Cribbage**

Our version of the classic pub card game is certain to give you all the fun and frustration of the real thing.

### **68 Which Way?**

Perplexing, puzzling, but always highly pleasurable, this original game of strategy for two players will really get you thinking.

### **74 Yams**

Based on the traditional dice game Yahtzee, it's easy to learn, but its subtle tactics will keep you intrigued.

### **80 Patience**

Simulation of the solo card game. Like the original it offers the same need for careful thought, the same excitement as you near the end.



# NOW GET OUT

HAVE you got problems! Trapped in the darkest recesses of a monster-infested mine, you have to battle your way upwards through 12 levels.

There is a lift to help, but you'll have to collect all the keys on each level before you can use it safely.

You can walk on and jump off the ledges and conveyor belts found on all levels, but the bricks cover the cavern floor completely. Be careful though – if you jump too far you're dead!

You'll have to watch the conveyor belts as well – you can't walk against the direction of travel. You'll stand still if you try!

And, of course, the final level has its own special trick, but you wouldn't want us to spoil it by telling, would you?

You can even tailor the game to your own design.

You'll have to change the user-defined characters 239 to 254 at the correct place in the program for the level being altered, then redefine logical colours 2 and 1.

Next, give the start position and direction of movement for the two hazards.

You must make sure that if the hazard moves horizontally there is some sort of ground beneath it and that there is a gap in the ground where it is to stop.

If the hazard moves vertically there must be pieces of ground where the hazard is to stop.

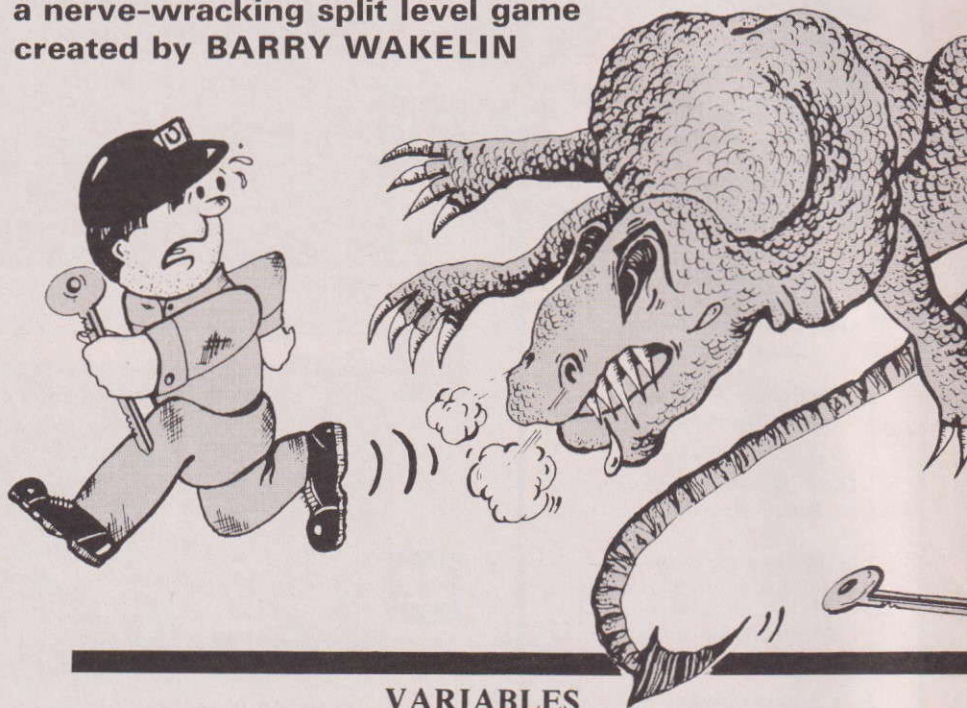
Next, give the name of this level. Then set the number of pieces of ground. Print, at their appropriate positions, the keys(K\$), stalactites(SS\$) and trees(T\$).

Now go to the line where the positions of that level's pieces of ground are stored and enter your own data. Each piece of ground is stored as X position, Y position, length and type of ground.

Type 1 is a ledge, type 2 is brick, type 3 is a left conveyor and type 4 is a right conveyor.

There are 24 different monsters, each of 2 x 2 character size. All this takes up rather a lot of memory, so take care not to add unnecessary spaces and so on when typing it in.

You've got monster-size problems in Explorer Eddie – a nerve-wracking split level game created by BARRY WAKELIN



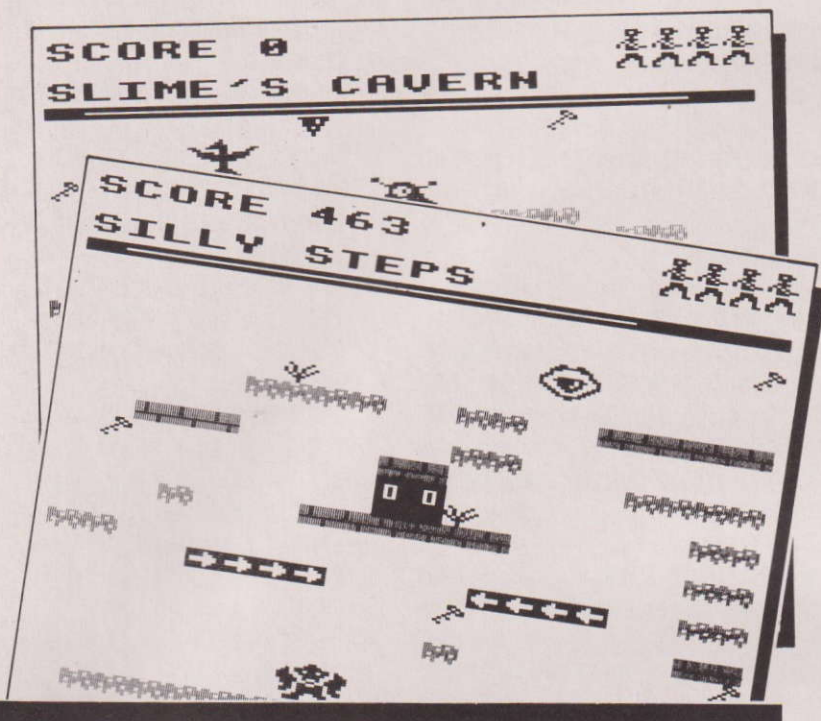
## VARIABLES

PS(n)	Character used for the player.
MS(n)	First character for each monster.
NS(n)	Second character for the monsters.
D%(n)	Horizontal direction of monster n.
E%(n)	Vertical direction of monster n.
A%(n)	X position of monster n.
B%(n)	Y position of monster n.
HIS(n)	Name of high scorer n.
HI%(n)	Score of high scorer n.
I%,I	General variables.
K\$	Key character.
SS	Stalactite character.
T\$	Tree character.
G\$	Puts CGOL3,3 in the PS(n) characters.
C\$	Puts cursor down and left in characters.
L%	Character number being used for monster 0.
LEV%	Level being played.
LIV%	Number of lives.
SCR%	Score.
FI%	True if level 13 is reached.
W%	Number of keys collected.
O%	Last character printed for the player's character.
C%	Character being used for the player's character.
J%	True if you are jumping.
V%	Vertical movement of player.
DE%	True if hit by monster.
TIM%	Time remaining.
D%	Used to store direction of motion before a jump.
X%	X position of player.
Y%	Y position of player.
Q%	Number of pieces of ground used on this level.
S%	Start of piece of ground.
F%	Length of piece of ground.





# OF THAT!



K%	Type of ground.
T%	X position of lift.
U%	Y position of lift.
K%	When elsewhere in the program it is used as the character for monster 1.
H%	Horizontal move of player.
Z%	Stage of jump.
A%	This is two if the player is going left.
R%	Ranking on table.
WS	String to be printed in double height.
NS	Name of level.

## DESCRIPTION OF LINES

10-80	Set up characters, arrays, variables and envelopes.
90-110	Set up screen.
120-150	Move the monsters if they move vertically.
160-170	Move the monsters if they move horizontally.
180-190	Print monsters at their new positions.
200	End if player has been hit.
210	Decrement clock.
220	Check if the player is jumping.
230-240	Check for left, right input.
250-260	Check if player is on a conveyor belt.
270	Check for jump input.
280-330	Act on move.
340-370	Move player and restart loop.
380-480	Checks and moves if jumping.
490-500	Subroutine if a key is grabbed.
510-520	Lose a life.
530-540	End of game.
550-580	Input name if on score board.
590-620	Print up high score table.

```

5REM (C) The Micro User
10MODE5:DIMP$(3),M$(1),M$(1),DX(1)
,EX(1),AX(1),BX(1),HI$(8),HI$(8):FORI
X=1TO8:HI$(IX)="The Micro User":HI$(I
X)=100:NEXT
20ENVELOPE1,1,5,0,0,25,0,0,126,0,0
,-126,126,126:ENVELOPE2,1,0,0,0,0,0
,127,-3,-2,-1,126,0
30VDU23,224,28,22,28,8,28,40,84,91
,23,225,28,0,28,30,54,98,66,99,23,226
,92,0,28,28,20,20,20,22,23,227,56,104
,56,16,56,20,42,218,23,228,56,0,56,12
0,108,70,66,198,23,229,56,104,56,16,5
6,16,104,180,23,255,12,10,5,11,16,40,
64,32
40VDU23,233,28,22,28,8,28,8,22,45,
23,234,58,0,56,56,40,40,40,104,23,235
,0,255,169,171,137,171,169,255
50VDU23,230,66,36,169,170,84,56,16
,16,23,231,255,239,207,129,129,207,23
9,255,23,232,255,247,243,129,129,243,
247,255,23,236,206,251,239,255,189,24
7,166,34,23,237,223,223,223,223,0,251
,251,251,23,238,127,59,58,62,20,28,8,
8
60K$=CHR$255:S$=CHR$238:T$=CHR$230
:6$=CHR$18+CHR$3+CHR$3:C$=CHR$10+CHR$
8:P$(0)=6$+CHR$224+C$+CHR$225:P$(1)=6
$+CHR$233+C$+CHR$226:P$(2)=6$+CHR$227
+C$+CHR$228:P$(3)=6$+CHR$229+C$+CHR$2
34:C$=C$+CHR$8:R$=" "+C$+" ":GOTO59
0
70LEVX=1:LIVX=5:SCRX=0
806$=CHR$17+CHR$2:M$(0)=6$+CHR$239
+CHR$240+C$+CHR$241+CHR$242:M$(1)=6$+
CHR$243+CHR$244+C$+CHR$245+CHR$246:LX
=0:N$(0)=6$+CHR$247+CHR$248+C$+CHR$24
9+CHR$250:N$(1)=6$+CHR$251+CHR$252+C$
+CHR$253+CHR$254
90RESTORE(LEVX*20+640):COLOUR1:PRI
NTTAB(0,30)STRING$(20,CHR$236):FI$=0
:DX(0)=0:DX(1)=0:EX(0)=0:EX(1)=0:COLO
UR3:GOSUB(LEVX*60+910):IFFIX=180TO530
100WX=0:DX=0:CX=0:AX=0:JX=0:VX=0:DE
X=0:TINX=1200:DX=0:XZ=0:YX=124:FORIX=
1TOQX:READS$,P$,F$,K$:GOSUB(KX*10+900
):NEXT:PROC1FT(TX,UX):KX=0:BCOL3,3:VD
U5:MOVEX$,Y$:PRINTTAB(X$,Y$)P$(0):VDU
4
110COLOUR3:COLOUR128:PRINTTAB(0,1)"
SCORE ":SCRXTAB(16,1)STRING$(LIVX-1,P
$(0)+CHR$11)TAB(0,3)N$:VDU23,1;0;0;0;
0;
120KZ=(KX+1)MOD2:LZ=(LX+1)MOD2:FORI
X=0TO1:IFDX(IX)>0 GOTO160
130IFEX(IX)=-1ANDPOINT(AX(IX)*64,(3
2-BX(IX))*32+16)>0 EX(IX)=1
140IFEX(IX)=1ANDPOINT(AX(IX)*64,(32
-BX(IX))*32-80)>0 EX(IX)=-1

```



## From Page 5

```

15060T0180
160IFDZ(IX)=-1ANDPOINT(AZ(IX)*64-64
,(32-BZ(IX))*32-76)=0DZ(IX)=1
170IFDZ(IX)=1ANDPOINT(AZ(IX)*64+128
,(32-BZ(IX))*32-76)=0DZ(IX)=-1
180AZ(IX)=AZ(IX)+DZ(IX);BZ(IX)=BZ(I
X)+EZ(IX);IFAZ(IX)*64)=XZ-64ANDAZ(IX)
*64<XZ+64AND(32-BZ(IX))*32>YZ-36AND(3
2-BZ(IX))*32<YZ+68DEX=1:VDU5:6COL3,3:
MOVEXZ,YZ:PRINTP$(OZ):YZ=YZ-92:MOVEXZ
,YZ:PRINTP$(OZ):VDU4
190NEXT:PRINTTAB(AZ(O)-DZ(O),BZ(O)-
EZ(O))R$TAB(AZ(O),BZ(O))M$(KX)TAB(AZ(
1)-DZ(1),BZ(1)-EZ(1))R$TAB(AZ(1),BZ(1
))M$(LX)
200IFDEX=160T0510
2106COL0,3:PL0T69,TIMZ,876:TIMZ=TIM
Z-4:IFTIMZ<7560T0510
220IFJZ=160T0380
230HZ=0:IFINKEY(-98)HZ=-64
240IFINKEY(-67)HZ=64
250IFPOINT(XZ+HZ+32,YZ-84)=2HZ=HZ-6
4:IFHZ<-64HZ=-64
260IFPOINT(XZ+HZ+24,YZ-84)=2HZ=HZ+6
4:IFHZ>64HZ=64
270IFINKEY(-74)JZ=1:Z=0:DZ=HZ:HZ=0
:60T0380
280IFXZ+HZ<0ORXZ+HZ>1216HZ=0
290IFPOINT(XZ+HZ,YZ-64)=0JZ=1:Z=4:
DZ=0
300IFLEVZ=12ANDPOINT(XZ+HZ,YZ-76)=3
60SUB630
310IFHZ=0ANDVZ=0FORI=1TO150:NEXT:60
T0120
320IF(XZ+HZ)=TZANDXZ+HZ<TZ+128ANDYZ
+VZ=UZ)ANDWX<360T0640
330IFPOINT(XZ+HZ+8,YZ+VZ-32)=360T05
10
340VDU5:AX=0:IFHZ=-64ORDZ=-64AZ=2
350CZ=(CX+1)MOD2:XX=XZ+HZ:YZ=YZ+VZ:
MOVEXZ-HZ,YZ-VZ:PRINTP$(OZ):MOVEXZ,YZ
:PRINTP$(CX+AZ):OZ=CX+AZ:VDU4,23,1;0;
0;0;0;
360IF(AZ(O)*64)=XZ-64ANDAZ(O)*64<XZ
+64AND(32-BZ(O))*32>YZ-36AND(32-BZ(O)
)*32<YZ+68)OR(AZ(1)*64)=XZ-64ANDAZ(1)
*64<XZ+64AND(32-BZ(1))*32>YZ-36AND(32
-BZ(1))*32<YZ+68)60T0510
37060T0120
380ZZ=ZZ+1:SOUND1,-10,ZZ*4,1:HZ=0:I
FZZ>2ANDZZ<6HZ=DZ
390IFXZ+HZ<0ORXZ+HZ>1216HZ=0
400VZ=32:IFZZ>3VZ=-32
410IFYZ+VZ<92ORYZ+VZ>860VZ=0
420IF(VZ=32ANDPOINT(XZ+HZ+8,YZ+VZ)=
3)OR(VZ=-32ANDPOINT(XZ+HZ+8,YZ-64)=3)
60T0510
430IFVZ=-32ANDPOINT(XZ+HZ,YZ-64)>0J

```

```

Z=0:VZ=0:DZ=0:60T0230
440IFVZ=32ANDPOINT(XZ+HZ+32,YZ+VZ)=
360SUB490:MOVEXZ+HZ,YZ+VZ:VDU255
450IFVZ=32ANDPOINT(XZ+HZ+8,YZ+16)=2
VZ=0:ZZ=3
460IFVZ=-32ANDPOINT(XZ+HZ+32,YZ-64)
=360SUB490:MOVEXZ+HZ,YZ-64:VDU255
470IFZZ=1160T0510
48060T0310
490SOUND2,2,150,1:WZ=WZ+1:SCRZ=SCRZ
+LEVZ+25:COLOUR3:PRINTTAB(6,1);SCRZ:V
DU5:6COL0,0:TIMZ=TIMZ+64:IFTIMZ>1264T
IMZ=1264
500RETURN
510VDU5:6COL3,3
520REPEAT:YZ=YZ-32:MOVEXZ,YZ+32:PRI
NTP$(OZ):MOVEXZ,YZ:PRINTP$(OZ):SOUND1
,-12,YZ/4,2:FORI=1TO150:NEXT:UNTILYZ<
128:SOUND0,2,6,1:LIVZ=LIVZ-1:IFLIVZ>0
MODE5:60T080
530FORI=1TO3000:NEXT:MODE7:IFSCRZ<H
IZ(8)60T0590
540*FX15,1
550PRINTTAB(8,3)CHR$136"THAT'S A NE
W HI-SCORE"TAB(10,7)CHR$130"What is y
our name"
560INPUTTAB(0,9)P$:IFLEN(P$)>24PRIN
TTAB(15,12)CHR$129"Too long.":FORI=1T
05000:NEXT:CLS:60T0550
570RZ=9:REPEAT:RZ=RZ-1:UNTILRZ=10RH
IZ(RZ-1)>SCRZ
580FORIZ=BTORZ+1STEP-1:HI$(IZ)=HI$(
IZ-1):HIZ(IZ)=HIZ(IZ-1):NEXT:HI$(RZ)=
P$:HIZ(RZ)=SCRZ
590MODE7:FORI=0TO1:PRINTTAB(4,IZ)C
HR$141CHR$129"EXPLORER EDDIES HALL OF
FAME":NEXT:FORI=1TO8:PRINTTAB(0,IZ*
2+3);IZTAB(3,IZ*2+3)STRING$(28,".")TA
B(5,IZ*2+3)HI$(IZ)TAB(33,IZ*2+3);HIZ(
IZ):NEXT
600*FX15,1
610PRINTTAB(1,3)CHR$130"Collect the
keys to escape the mine."TAB(2,21)CH
R$133"Z - Left X - Right RETURN - J
ump"TAB(5,23)CHR$131"Press any key to
play again."
620QZ=BET:MODE5:60T070
630DZ(O)=0:DZ(1)=0:EZ(O)=0:EZ(1)=0:
SOUND0,2,6,1:VDU19,0,3;0;0:FORI=1TO100
:NEXT:VDU19,0,0;0;0;5:6COL0,0:MOVEXZ+HZ
,YZ-64:VDU235,4:RETURN
6406COL0,3:COLOUR3:VDU4:FORI=TIMXT
075STEP-8:PL0T69,IZ,876:SCRZ=SCRZ+5:P
RINTTAB(6,1);SCRZ:SOUND1,-13,75,.1:NE
XT:LEVZ=LEVZ+1:FORI=1TO3000:NEXT:MODE
5:60T080
650DEFPROCdb1(W$,YZ):W$=CHR$141+W$:
TZ=(40-LEN(W$))/2-1:PRINTTAB(TZ,YZ)CH
R$130W$TAB(TZ,YZ+1)CHR$134W$:ENDPROC
660DATA0,26,8,1,9,27,11,1,5,23,3,3,

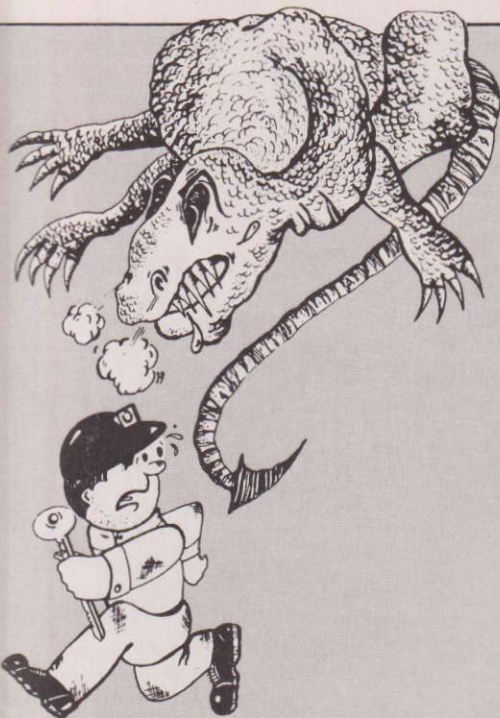
```

```

14,21,6,2,7,20,3,1,0,19,3,1,10,18,6,1
,4,17,2,2
670DATA16,17,2,2,0,14,4,1,16,15,2,1
,16,13,2,1,16,11,2,1,2,11,2,1,6,10,9,
1
680DATA0,27,2,1,3,25,4,1,9,25,4,2,1
5,25,3,1,18,22,2,2,3,20,4,1,9,20,4,3,
15,20,3,1
690DATA0,17,2,1,2,15,2,1,6,15,7,2,1
5,15,5,1,11,12,2,1,15,12,3,2,5,10,2,1
,9,10,2,1,0,7,4,2,11,8,9,1
700DATA15,28,2,1,1,26,13,1,18,26,2,
1,16,24,2,2,18,21,2,2,0,18,2,1,4,19,4
,4,12,19,4,3,18,19,2,1
710DATA18,17,2,1,3,16,1,1,7,16,6,2,
18,15,2,1,9,13,2,2,16,13,4,1,2,12,3,2
,11,12,2,1,5,10,4,1,11,10,2,1,15,10,5
,2,11,22,1,1
720DATAB,27,1,2,5,26,1,1,1,25,2,1,1
1,25,1,2,0,22,1,1,8,23,1,1,13,23,3,1,
11,21,1,1,16,21,4,2,9,18,1,1
730DATA0,17,4,1,12,16,1,1,15,16,4,1
,7,15,1,1,19,15,1,2,0,14,1,1,4,13,2,1
,0,11,1,1,6,10,13,4,4,9,1,1,0,8,1,1,1
,8,3,4
740DATA13,27,4,1,15,24,1,1,1,23,4,1
,6,23,1,1,9,23,1,1,12,23,1,1,0,22,1,2
,0,19,4,1,1,17,8,4,11,17,8,4
750DATA0,16,1,1,9,14,2,2,19,14,1,2,
0,13,1,1,3,12,2,1,16,12,2,3,4,10,1,1,
15,10,1,1,0,8,2,1,5,8,3,1,9,8,5,1,16,
7,4,2
760DATA0,27,1,1,4,25,13,2,18,27,1,1
,1,24,2,1,3,22,14,2,9,22,2,4,17,21,2,
3,5,18,9,1
770DATA16,18,2,1,4,16,1,1,1,15,1,1,
3,12,2,4,6,12,3,1,11,12,3,4,17,14,3,2
,13,9,1,1,13,7,5,1,18,9,1,2
780DATA2,27,1,1,5,26,1,1,15,26,1,1,
9,25,3,4,16,24,1,1,3,23,2,1,12,23,2,1
,17,22,1,1,13,21,2,1,6,20,2,1,18,20,1
,1,9,19,1,1,2,17,1,1,5,17,1,2,9,17,1,
1,12,17,2,4,15,17,1,1,16,17,2,4,19,15
,1,1,2,14,1,1,16,14,1,2,5,13,1,1,5,11
,1,1,6,11,10,4
790DATA0,8,2,2,2,8,3,3
800DATA0,27,1,1,1,27,17,4,11,27,2,2
,13,23,1,3,2,21,7,3,10,21,1,1,0,19,2,
1,2,16,1,1,5,16,2,4,11,16,1,2,10,14,1
,1
810DATA2,13,1,1,7,13,1,2,3,10,1,1,0
,8,2,1,4,7,12,3,16,10,4,1,16,13,1,1,1
6,16,1,1,18,19,1,1,18,22,1,1,18,24,1,
2
820DATA1,28,1,1,0,25,1,1,3,25,1,1,1
3,25,1,3,16,25,1,3,10,24,1,1,6,23,1,1
,17,23,1,1,2,21,1,2,6,21,1,1,8,20,8,4
,18,20,1,1
830DATA3,17,7,3,13,17,1,1,16,15,1,1
,19,15,1,1,7,14,1,1,3,13,1,1,19,12,1,

```





```

1,4,11,4,3,1,9,2,4,16,9,4,1
840DATA8,27,2,2,11,25,1,1,14,25,5,4
,3,23,1,1,6,23,3,4,2,22,1,1,19,22,1,2
,14,20,5,3,3,19,10,3,3,16,7,2,11,16,2
,2,1,14,1,1,9,13,1,4,11,13,1,1,14,13
,2,4,1,12,2,4,5,12,1,1,18,12,1,1,8,10
,1,1,5,9,1,1,15,9,1,1,18,9,1,1,0,8,1,1
850DATA2,8,1,3,12,8,1,1,9,7,2,1
860DATA9,27,1,2,11,27,1,2,9,24,1,1,
11,24,1,1,0,22,1,2,1,21,18,4,19,19,1,
1,1,17,18,3,1
870DATA13,8,3,10,13,9,4,0,10,3,2,16
,10,2,3,13,9,1,1,10,8,1,3,8,8,1,3,6,8
,1,3,5,8,1,2,4,8,1,4,10,16,1,2
880DATA7,27,1,2,10,26,1,1,16,26,1,1
,8,23,1,3,12,23,4,4,17,23,2,3,1,22,5
,3,1,19,2,2,4,19,2,2,9,19,1,1,12,19,1
,3,13,18,4,4
890DATA19,17,1,2,1,16,1,1,3,15,1,1,
6,15,6,4,14,18,2,3,1,14,2,4,15,14,1,3
,18,14,2,3,0,10,3,2,5,11,10,2,3,10,1,
4
900DATA0
910COLOUR1:COLOUR128:PRINTTAB(SX,PZ
)STRIN$ (FX,CHR$236);:RETURN
920COLOUR1:COLOUR130:PRINTTAB(SX,PZ
)STRIN$ (FX,CHR$237);:RETURN
930COLOUR2:COLOUR128:PRINTTAB(SX,PZ
)STRIN$ (FX,CHR$231);:RETURN
940COLOUR2:COLOUR128:PRINTTAB(SX,PZ
)STRIN$ (FX,CHR$232);:RETURN
950DEFPROC1ft(TZ,UZ):BCOLO,3:MOVETZ
,UZ:MOVETZ+120,UZ:PLOT85,TZ+120,UZ-64
:MOVETZ,UZ-64:PLOT85,TZ,UZ:BCOLO,0:MO
VETZ+16,UZ-12:DRAWTZ+32,UZ-12:DRAWTZ+
32,UZ-32:DRAWTZ+16,UZ-32:DRAWTZ+16,UZ
-12:MOVETZ+104,UZ-12:DRAWTZ+104,UZ-32
960DRAWTZ+92,UZ-32:DRAWTZ+92,UZ-12:
DRAWTZ+104,UZ-12:BCOLO,3:FORIZ=868TO8
84STEP4:MOVE0,IZ:DRAW1279,IZ:NEXT:BCO
LO,0:MOVE79,876:DRAW1200,876:ENDPROC
970VDU23,239,0,0,0,195,199,62,13,13

```

```

,23,240,0,6,6,200,240,96,32,32,23,241
,30,31,60,120,251,63,30,12,23,242,112
,240,216,60,126,255,239,70
980VDU23,243,0,96,96,19,15,6,4,4,23
,244,0,0,0,195,227,124,176,176,23,245
,14,15,27,56,124,255,247,98,23,246,12
0,248,124,62,223,252,120,48
990VDU23,247,1,3,5,15,1,25,63,127,2
3,248,0,0,128,128,128,152,252,254,23,
249,199,131,1,1,3,7,12,16,23,250,227,
193,128,128,192,224,48,8
1000VDU23,251,1,3,7,207,97,61,63,31,
23,252,0,0,128,131,134,188,252,248,23
,253,7,1,1,1,3,7,12,8,23,254,224,128,
128,128,192,224,48,16
1010TX=1152:UX=124:VDU19,2,2;0;:N$="
SLIME'S CAVERN":DZ(0)=1:AZ(0)=6:BX(0)
=8:EX(1)=-1:AZ(1)=4:BX(1)=15:QZ=15:PR
INTTAB(7,5)S$TAB(5,18)S$TAB(0,8)K$TAB
(14,5)K$TAB(13,14)K$TAB(18,15)K$
1020RETURN
1030VDU23,239,3,14,61,123,247,206,4,
0,23,240,192,112,188,222,239,115,32,1
28,23,241,0,0,0,0,0,0,0,23,242,128,
128,128,128,128,144,80,32
1040VDU23,243,1,2,5,13,11,11,27,21,2
3,244,128,64,160,176,208,208,216,168,
23,245,25,8,0,0,0,4,5,2,23,246,152,16
,128,128,128,128,0,0
1050VDU23,247,7,31,60,56,116,226,193
,194,23,248,224,248,60,30,46,71,131,6
7,23,249,194,193,226,116,56,60,31,7,2
3,250,67,131,71,46,30,60,248,224
1060VDU23,251,7,31,60,48,112,224,193
,194,23,252,224,248,188,140,142,135,1
31,127,23,253,254,193,225,113,49,61,3
1,7,23,254,67,131,7,14,12,60,248,224
1070VDU19,2,6;0;19,1,4;0;:AZ(0)=7:BX
(0)=16:EX(0)=1:AZ(1)=13:BX(1)=28:EX(1)
=-1:N$="LOOPY LEVELS":QZ=18:PRINTTAB
(19,18)K$TAB(0,13)K$TAB(18,12)K$TAB(1
7,5)K$TAB(4,19)T$TAB(15,7)T$T$=0:UX=
860
1080RETURN
1090VDU23,239,0,6,15,28,16,48,103,79
,23,240,0,96,240,56,8,12,230,178,23,2
41,71,103,51,25,12,7,1,0,23,242,162,1
02,204,152,48,224,128,0
1100VDU23,243,0,0,0,6,15,31,51,23,
244,0,0,0,96,240,248,204,23,245,28,
15,7,1,0,0,0,0,23,246,56,248,240,128,
0,0,0,0
1110VDU23,247,1,3,47,109,231,243,250
,121,23,248,128,192,244,182,231,207,9
5,158,23,249,118,55,119,103,14,60,60,
0,23,250,110,236,238,230,112,48,60,60
1120VDU23,251,1,3,47,109,231,243,251
,248,23,252,128,192,244,182,231,207,2
23,31,23,253,119,55,119,103,14,12,60,
60,23,254,238,236,238,230,112,60,60,0

```

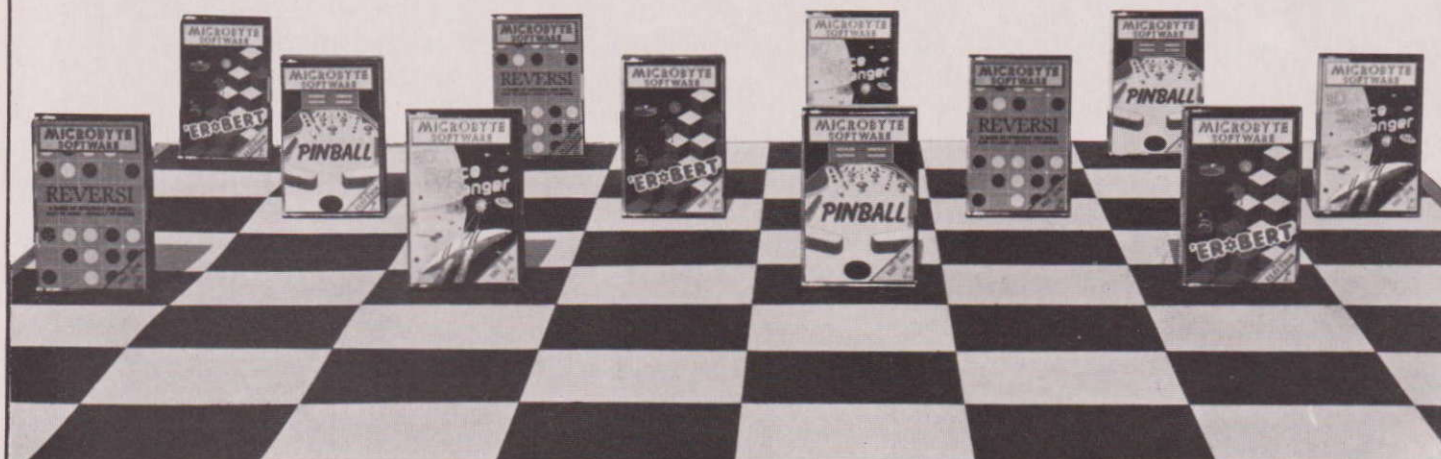
```

1130VDU19,2,1;0;19,1,2;0;:QZ=21:TZ=5
76:UX=572:AZ(0)=13:BX(0)=5:EX(0)=1:AZ
(1)=1:BX(1)=24:DZ(1)=1:N$="SILLY STEP
S":PRINTTAB(19,22)K$TAB(11,20)K$TAB(1
,13)K$TAB(19,6)K$TAB(6,9)T$TAB(11,15)
T$
1140RETURN
1150VDU23,239,3,15,63,113,111,249,24
9,255,23,240,192,240,252,142,246,159,
159,255,23,241,239,223,219,104,124,62
,15,3,23,242,247,251,219,22,62,124,24
0,192
1160VDU23,243,3,15,63,127,113,239,24
9,255,23,244,192,240,252,254,142,247,
159,255,23,245,255,247,239,124,127,63
,15,3,23,246,255,239,247,62,254,252,2
40,192
1170VDU23,247,0,0,0,4,12,30,15,7,23,
248,0,0,0,0,0,0,6,142,23,249,3,0,3,15
,31,63,31,7,23,250,58,250,246,252,248
,240,192,0
1180VDU23,251,0,0,0,0,0,0,96,113,23,
252,0,0,0,32,48,120,240,224,23,253,92
,95,111,63,31,15,3,0,23,254,192,0,192
,240,248,252,248,224
1190VDU19,2,3;0;19,1,5;0;:AZ(0)=9:BX
(0)=19:EX(0)=1:AZ(1)=6:BX(1)=8:DZ(1)=
1:N$="SUNSHINE CAVERNS":TZ=1152:UX=41
2:QZ=22:PRINTTAB(0,18)K$TAB(12,12)K$T
AB(0,5)K$TAB(11,5)K$TAB(3,7)T$TAB(16,
15)T$
1200RETURN
1210VDU23,239,15,31,24,24,57,57,48,4
8,23,240,254,255,31,15,206,142,62,14,
23,241,112,115,103,103,224,224,255,12
7,23,242,12,204,156,156,24,24,248,240
1220VDU23,243,15,31,24,24,57,57,48,4
8,23,244,254,255,31,15,206,142,62,14,
23,245,112,115,103,103,224,224,255,12
7,23,246,12,204,156,156,24,24,248,240
1230VDU23,247,15,31,28,28,56,57,51,5
1,23,248,254,255,31,15,198,230,254,25
4,23,249,115,103,103,103,240,240,255,
127,23,250,252,252,204,156,24,56,248,
240
1240VDU23,251,15,31,28,28,56,57,51,5
1,23,252,254,255,31,15,198,230,254,25
4,23,253,115,103,103,103,240,240,255,
127,23,254,252,252,204,156,24,56,248,
240
1250VDU19,2,5;0;19,1,6;0;:AZ(0)=7:BX
(0)=15:DZ(0)=-1:AZ(1)=11:BX(1)=15:DZ(
1)=1:N$="ADVERTISING HALL":TZ=1152:UX
=860:QZ=22:PRINTTAB(3,13)K$TAB(10,10)
K$TAB(17,13)K$TAB(1,5)K$TAB(3,29)T$="
T$" "T$" "T$TAB(2,20)S$
1260PRINTTAB(0,5)S$TAB(5,7)T$TAB(11,
7)T$TAB(13,5)S$:PRINTTAB(9,15)N$(0):R

```



# YOUR MOVE...



## TAKE A PIECE OF OUR SOFTWARE!

### SUPER GAMES

ONLY  
**£4.95**  
INC VAT  
&  
p+p

### SUPER VALUE

#### ER\*BERT



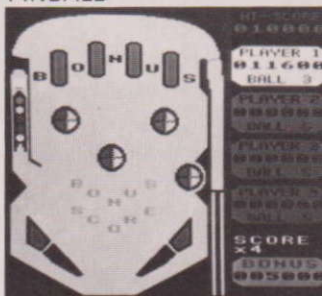
**Electron & BBC 32K**  
Join Erbert in his cubic domain – fast and funny. Avoid his unwelcome guests. Many features – alternative screen displays – addictive!  
MACHINE CODE GAME

#### 3D SPACE RANGER



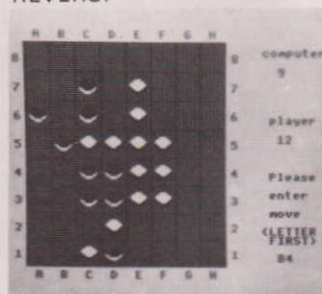
**BBC 32K (OS 1.0 or 1.2)**  
Excellent 3D graphics four different scenes. Battle to the death star and destroy it.  
MACHINE CODE GAME

#### PINBALL



**Electron & BBC 32K**  
The classic arcade game up to four players with bonus features.  
MACHINE CODE GAME

#### REVERSI



**Electron & BBC 32K**  
A game of strategy and skill – 2 levels

## MICROBYTE SOFTWARE

Games available now at many computer shops – or by fast mail order from Microbyte Software, S.A.E. for illustrated brochure. Trade enquiries welcome. Access 24 hour hot line **06373 6886**.

**MICROBYTE SOFTWARE (Dept. E10)**  
18 Hilgrove Road, Newquay, Cornwall TR7 2QZ



# From Page 7

## ETURN

1270VDU23,239,112,136,168,120,60,60,  
30,30,23,240,14,17,21,30,60,60,120,12  
0,23,241,63,58,114,120,61,31,50,50,23  
,242,252,92,78,30,188,248,76,76  
1280VDU23,243,28,42,34,60,120,240,12  
4,30,23,244,56,84,68,60,30,31,62,120,  
23,245,63,63,114,120,63,31,13,13,23,2  
46,252,252,78,30,252,248,176,176  
1290VDU23,247,3,7,7,76,126,63,15,7,2  
3,248,192,224,241,159,190,248,240,240  
,23,249,3,1,1,1,3,7,14,24,23,250,240,  
224,224,192,192,128,0,0  
1300VDU23,251,3,7,143,249,125,31,15,  
15,23,252,192,224,224,50,126,252,240,  
224,23,253,15,7,7,3,3,1,0,0,23,254,19  
2,128,128,128,192,224,112,24  
1310VDU19,2,6;0;19,1,1;0;:AZ(0)=3:BZ  
(0)=20:DX(0)=1:AZ(1)=5:BZ(1)=16:DX(1)  
=1:N\$="SPOOKY SIDE-CHAMBER":TX=1088:U  
Z=636:QZ=18:PRINTTAB(19,27)K\$TAB(7,19  
)K\$TAB(10,19)K\$TAB(9,14)K\$TAB(6,29)T\$  
" T\$ " T\$ " T\$  
1320PRINTTAB(4,26)S\$ " S\$ " S\$ " "  
S\$ " S\$TAB(6,11)T\$TAB(12,11)T\$:RETUR  
N  
1330VDU23,239,1,3,7,15,31,25,53,49,2  
3,240,128,192,224,240,248,248,252,236  
,23,241,123,127,127,3,7,63,30,7,23,24  
2,246,246,246,238,220,184,112,224  
1340VDU23,243,1,3,7,15,31,31,63,55,2  
3,244,128,192,224,240,248,152,172,140  
,23,245,111,111,111,119,59,29,14,7,23  
,246,222,254,254,192,224,248,112,224  
1350VDU23,247,1,3,7,7,7,7,3,1,23,248  
,128,192,224,160,160,96,192,128,23,24  
9,1,15,17,1,7,4,12,0,23,250,240,128,1  
28,128,224,32,32,48  
1360VDU23,251,1,3,3,3,3,3,1,17,23,25  
2,128,192,192,64,64,192,128,128,23,25  
3,15,1,1,1,7,4,4,12,23,254,128,240,13  
6,128,224,32,48,0  
1370VDU19,2,3;0;19,1,4;0;:AZ(0)=3:BZ  
(0)=9:EX(0)=1:AZ(1)=14:BZ(1)=9:DX(1)=  
-1:N\$="MINERS DINER":TX=0:UX=828:QZ=2  
6:PRINTTAB(2,5)S\$ " S\$TAB(9,21)K\$TAB(  
16,11)K\$TAB(10,7)K\$TAB(3,5)K\$TAB(12,2  
9)T\$  
1380RETURN  
1390VDU23,239,3,3,49,31,7,3,3,7,23,2  
40,192,192,142,248,96,192,64,224,23,2  
41,14,220,120,48,0,0,0,0,23,242,112,5  
9,30,12,0,0,0,0  
1400VDU23,243,0,0,3,3,1,15,31,51,23,  
244,0,0,192,192,128,240,120,204,23,24  
5,99,71,30,56,48,24,56,112,23,246,70,  
226,120,28,12,24,28,14  
1410VDU23,247,7,15,31,63,115,99,99,1

27,23,248,192,224,240,248,156,140,140  
,252,23,249,61,63,28,16,21,31,15,3,23  
,250,120,248,112,16,80,240,224,128  
1420VDU23,251,7,15,31,63,115,99,99,1  
27,23,252,192,224,240,248,156,140,140  
,252,23,253,61,63,31,24,15,3,0,0,23,2  
54,120,248,240,48,224,128,0,0  
1430VDU19,2,1;0;19,1,3;0;:AZ(0)=3:BZ  
(0)=19:DX(0)=1:AZ(1)=11:BZ(1)=17:EX(1  
)=1:N\$="ANCIENT ARCH":TX=1152:UX=764:  
QZ=22:PRINTTAB(0,5)K\$TAB(8,22)K\$TAB(7  
,17)K\$TAB(7,8)K\$TAB(4,22)S\$TAB(5,26)T  
\$ " T\$ " T\$  
1440RETURN  
1450VDU23,239,0,0,0,0,0,7,15,14,23,2  
40,48,120,120,60,6,195,230,44,23,241,  
14,14,15,8,12,7,3,15,23,242,184,48,22  
4,32,96,192,128,224  
1460VDU23,243,0,0,0,7,15,8,10,8,23,2  
44,0,48,120,126,179,198,236,248,23,24  
5,15,12,13,7,3,3,3,15,23,246,240,96,9  
6,192,128,128,128,224  
1470VDU23,247,1,1,15,1,81,115,37,63,  
23,248,128,128,240,128,138,206,164,25  
2,23,249,2,3,0,54,0,94,64,27,23,250,6  
4,192,0,216,2,122,0,108  
1480VDU23,251,1,1,3,97,97,35,37,63,2  
3,252,128,128,192,134,134,196,164,252  
,23,253,3,3,0,27,64,94,0,54,23,254,19  
2,192,0,108,0,122,2,216  
1490VDU19,2,6;0;19,1,2;0;:AZ(0)=3:BZ  
(0)=15:DX(0)=1:AZ(1)=14:BZ(1)=18:DX(1  
)=-1:N\$="RAMPAGING ROBOTS":TX=1024:UX  
=796:QZ=22:PRINTTAB(2,5)K\$TAB(5,7)K\$T  
AB(16,11)K\$TAB(10,21)K\$TAB(12,29)T\$ "  
T\$  
1500RETURN  
1510VDU23,239,16,16,24,8,12,12,6,6,2  
3,240,16,16,48,32,96,96,192,192,23,24  
1,3,1,3,6,14,14,6,2,23,242,128,0,128,  
192,224,224,192,128  
1520VDU23,243,0,0,64,96,48,56,28,14,  
23,244,0,0,4,12,24,56,112,224,23,245,  
3,1,3,14,12,24,24,8,23,246,128,0,128,  
224,96,48,48,32  
1530VDU23,247,0,0,0,252,252,104,120,  
108,23,248,0,0,0,0,0,60,230,23,249,  
127,117,255,255,255,0,248,248,23,250,  
242,251,255,254,248,3,127,30  
1540VDU23,251,252,252,104,120,108,12  
7,117,255,23,252,0,0,0,60,230,242,251  
,255,23,253,255,255,0,248,248,0,0,0,2  
3,254,254,248,3,127,30,0,0,0  
1550VDU19,2,3;0;19,1,1;0;:AZ(0)=3:BZ  
(0)=5:EX(0)=1:AZ(1)=3:BZ(1)=17:DX(1)=  
1:N\$="ESCAPED EQUIPMENT":TX=1152:UX=1  
24:QZ=26:PRINTTAB(6,20)S\$ " S\$TAB(14,  
21)S\$ " S\$TAB(7,20)K\$TAB(15,21)K\$TAB(  
0,5)K\$TAB(8,7)K\$TAB(18,24)T\$  
1560PRINTTAB(6,15)T\$:RETURN  
1570VDU23,239,3,15,63,255,1,1,1,1,23

,240,192,240,252,255,128,128,128,128,  
23,241,127,15,39,112,219,219,115,32,2  
3,242,254,240,228,14,219,219,206,4  
1580VDU23,243,0,0,0,3,15,63,255,1,23  
,244,0,0,0,192,240,252,255,128,23,245  
,127,15,39,112,219,219,115,32,23,246,  
254,240,228,14,219,219,206,4  
1590VDU23,247,0,0,0,0,0,0,25,23,24  
8,24,60,126,231,195,231,126,188,23,24  
9,61,126,231,195,231,126,60,24,23,250  
,152,0,0,0,0,0,0,0  
1600VDU23,251,24,60,126,231,195,231,  
126,61,23,252,0,0,0,0,0,0,152,23,25  
3,25,0,0,0,0,0,0,23,254,188,126,231  
,195,231,126,60,24  
1610VDU19,2,2;0;19,1,4;0;:AZ(0)=17:B  
Z(0)=19:DX(0)=-1:AZ(1)=10:BZ(1)=11:DX  
(1)=1:N\$="PROCESSING PLANT":TX=0:UX=7  
64:QZ=19:PRINTTAB(4,18)STRING\$(7,S\$)T  
AB(14,18)S\$S\$S\$TAB(5,16)STRING\$(4,T\$+  
" ")TAB(5,7)T\$  
1620PRINTTAB(3,12)T\$ " T\$TAB(10,29)  
T\$TAB(3,18)K\$TAB(6,18)K\$TAB(17,18)K\$T  
AB(5,5)K\$:RETURN  
1630VDU23,239,1,3,3,6,5,12,63,255,23  
,240,128,192,192,96,160,48,252,255,23  
,241,234,106,63,31,15,9,16,0,23,242,8  
7,86,252,248,224,144,8,0  
1640VDU23,243,1,3,3,7,7,12,63,255,23  
,244,128,192,192,224,224,48,252,255,2  
3,245,245,117,63,31,7,25,0,0,23,246,1  
75,174,252,248,224,152,0,0  
1650VDU23,247,0,0,0,4,34,2,0,24,23,2  
48,0,0,0,36,64,128,152,36,23,249,4,2,  
0,0,3,3,3,3,23,250,64,0,8,0,192,192,1  
92,192  
1660VDU23,251,0,0,0,0,12,16,32,1,23,  
252,0,0,0,4,0,16,8,8,23,253,0,0,32,64  
,67,67,3,3,23,254,8,0,32,0,196,196,19  
6,192  
1670VDU19,2,3;0;19,1,5;0;:AZ(0)=5:BZ  
(0)=9:DX(0)=1:AZ(1)=13:BZ(1)=9:DX(1)=  
-1:N\$="THE GRAND FINALE":TX=0:UX=764:  
QZ=23:PRINTTAB(12,6)K\$TAB(11,12)K\$TAB  
(3,19)K\$TAB(17,19)K\$TAB(4,18)T\$TAB(13  
,22)T\$TAB(15,29)T\$  
1680PRINTTAB(14,19)S\$TAB(10,12)S\$TAB  
(2,29)T\$ " T\$TAB(3,14)CHR\$235:RETURN  
1690MODE7:FORIX=1TO4:SOUND1,1,40,12:  
NEXT:PROCdb1("CONGRATULATIONS",2):PRO  
Cdb1("YOU HAVE ESCAPED",5):PROCdb1("T  
HE LOST URANIUM MINE",8):PROCdb1("WIT  
H A FORTUNE",11):PROCdb1("IN GOLD KEY  
S !",14)  
1700SCRX=SCRX+5000\*LIVX:PROCdb1("BOM  
US "+STR\$(5000\*LIVX),18):FIZ=1:FORI=1  
TO17500:NEXT:RETURN  
1710REM  
1720REM #### EXPLORER EDDIE ####  
1730REM  
1740REM #### BY B.WAKELIN ####



# If you're devious, deft and deadly -



## PROCEDURES

<b>PROctitle</b>	Sets up MasterSpy title.
<b>PROCerr</b>	Error trap. If ERR=17 PROCreport ELSE REPORT ERROR with my name and address.
<b>PROCinit</b>	Sets DIM arrays for Messages (Mes\$), Colours (Col%), Answers (Ans%), Codes (Code\$) and Random (rnd%).
<b>PROCagent</b>	Asks for the player's name, waits for a mission.
<b>PROCinstruction</b>	Gives briefing to agent.
<b>PROCscore</b>	Keeps score.
<b>PROCsetcode</b>	Sets all codes, and makes sure that Files 1-3 do not duplicate any colour.
<b>PROCplayingboard</b>	Sets screen up for play.
<b>PROCplay</b>	Prints colour key and changes numbered input to the right coloured ★. Keeps tries to 10 per file.
<b>PROCreport</b>	Gives a mission report.
<b>PROCcheck</b>	Checks answer and assigns clues.
<b>PROCans</b>	Reveals correct code.
<b>PROCnewpage</b>	Press space bar to continue.
<b>PROCclues</b>	Mixes up answers and prints them as on screen 'X'.

## PROCnewsflash

If score is too good a newsflash from mission control.

## PROCscoretable

There are five places on the agent's league table. You have to be good to beat James Bond.

## PROCinsert

If an error other than 17(Escape) then this will list the line in which the error has occurred.

## VARIABLES

<b>HIS(5)</b>	James Bond.
<b>H% (5)</b>	Minimum score to enter score table.
<b>level%(6)</b>	File level in play.
<b>attempts%</b>	Total goes, all files.
<b>of%</b>	Number of files opened.
<b>err%</b>	Flag. err%=1 or 2. If 1 program terminated. If 2 program completed so that correct message is printed.
<b>Cn%</b>	Number of colours in code (1-5).
<b>N%</b>	Number of tries in each file (max 10).
<b>M%</b>	Counts how many '/' (right colour right place) in your answer. If M%=Cn% it is correct.
<b>Tries%</b>	Puts number of tries on screen.
<b>Col%</b>	Colours used in each file.



## But you have to be good to beat James Bond!

MASTERSPY is based on the well-known colour code game Mastermind. In this version you are a secret agent who, once accepted into the agency, is sent on a mission by mission control. You must uncover the secret colour codes of six files — but it is not easy.

File 1 contains any three of seven colours. File 2 contains any four of seven colours. File 3 contains any five of seven colours. These files do not contain any duplication of colours.

Once you have opened File 3 (five colours) you get on to the more difficult agency work. The next three files, although reverting to three, four and five colours of seven respectively, now may contain any number of colour duplications.

As you have to be good to be an agent you are only given 10 attempts at each file. If you don't crack by the tenth try the code will be revealed to you and you lose the file you previously opened, so you go back by one file. On File 1 should you fail to uncover the code it is rescrambled and you can try again.

During your spying activities mission control keeps an undercover watch over

your progress. If M decides you are likely to be caught he instructs that your mission is terminated and you will receive a mission report to that effect.

If on the other hand you complete your mission by opening all six files you will receive a report from your mission controller and may be duly promoted.

To join the ranks of James Bond you have to be very good. You must open all six files in fewer than 50 attempts. If you can do this you are invited to enter your name on the agents' league table.

To help you in your mission your controller gives you clues for each colour entered — but these do not relate to the correct position of the colours in the code. To enter your selection you must depress the relevant number key followed by Return for each colour in the code.

When the program runs you enter your name and if accepted for a mission you are asked: "Do you require a briefing?". This is a must for a newly enrolled agent as it contains mission control's clues, which must be memorised.

*Good luck in your mission.*

```
5 REM (C) The Micro User
10 MODE7
20 *FX200,1
30 VDU23;8202;0;0;0;CLS:PROctitle
40 CLEAR
50 *FX4,1
60 *KEY10 0.1MSOUND1,-15,75,5:SOUN
D2,-15,100,5:IM RUN IM
70 REM ON ERROR PROCerr
80 VDU23;8202;0;0;0;
90 DIM H1$(5),H2$(5),P$(5):FORA=1TO
5:H1$(A)="James Bond"+STRING$(13,".")
:H2$(A)=50:P$(A)=A:NEXT A
100 PROCinit
110 level%=1:attempts%=0:of%=0:err%
=0:file1%=0:file2%=0:file3%=0:file4%
=0:file5%=0:file6%=0
120 PROCtitle:PROCagent:PROCtitle:P
RINTTAB(13,8);CHR$131;"Agent ";Name$
130 PRINTTAB(2,10)CHR$131;" you hav
e been accepted for a mission"
140 REPEATPRINTTAB(4,12)CHR$131;"Do
you require a briefing? (Y/N)";:Z$=6
ET$
150 X%=INSTR("YyNn",Z$)
160 UNTIL (X%>0)
170 IF X%=1 OR X%=2 THEN PROCinstru
ctions ELSE 180
180 IF level%>6 THEN err%=2:of%=6:C
LS:PROCscore
190 IF level%=6 OR level%>3 THEN Cn
```

```
X%=level%-1 ELSE CnX=level%+2
200 M%=0:N%=0:X=2:Y=6:Tries%=0:VDU2
3;8202;0;0;0;
210 PROCsetcode
220 PROCtitle:PROCplayingboard
230 REPEAT
240 Tries%=Tries%+1
250 N%=N%+1:Y=Y+1
260 IF M%=Cn% THEN 360
270 PROCplay
280 attempts%=attempts%+1
290 IF of%<5 AND attempts%>50 THEN
300 ELSE 320
300 err%=1:CLS:PROCreport:SOUND1,1,
136,50:ENVELOPE1,1,-7,7,0,10,10,0,126
,0,0,-126,126,126:Z=INKEY(200)
310 PROCtitle:PROCreport:PROCplayin
gboard:PROCans:PROCscore
320 UNTIL N%=10:IF M%=Cn% THEN 360
330 PROCans:PRINTTAB(0,23);SPC(39);
:PRINTTAB(11,23);CHR$133;"Incorrect c
ode ":VDU7:Z=INKEY(200)
340 PROCnewpage
350 IF level%=1 THEN of%=0:GOTO 180
ELSE level%=level%-1:of%=level%:GOTO
180
360 PROCans:PRINTTAB(0,23);SPC(39);
:PRINTTAB(12,23);CHR$131;"Correct cod
e ":SOUND1,2,4,50:ENVELOPE2,2,6,0,0,2
55,0,0,126,0,0,-126,126,126:Z=INKEY(2
50)
```

```
370 of%=level%-1:level%=level%+1:GO
TO 180
380 END
390 REM ***** init *****
400 DEFPROCinit
410 DIM Mes$(23):FORA=1TO23:Mes$(A)
=" ":NEXT
420 DIM Col%(5),Ans%(5),Code$(5),rn
d%(5),A(7)
430 A=0:REPEAT:A=A+1
440 Col%(A)=0:Ans%(A)=0:Code$(A)=".
":UNTIL A=5
450 filec%=135:filel%=129
460 ENDPROC
470 REM ***** setcode *****
480 DEFPROCsetcode
490 IF level%=3 THEN 530 ELSE 500
500 C%=0:REPEAT:C%=C%+1
510 Col%(C%)=RND(7)
520 UNTIL C%=Cn%:ENDPROC
530 FOR Z=1TO7:A(Z)=0:NEXT
540 C=0:Q=0:REPEAT:Q=Q+1:IF C=7 END
PROC
550 X=RND(-TIME)
560 X=RND(7)
570 IF A(X)=1 THEN Q=Q-1:GOTO 590 E
LSE A(X)=1
580 Col%(Q)=X
590 UNTIL Q=Cn%
600 ENDPROC
```



## From Page 11

```

610 REM ***** playingboard *****
620 DEFPROCplayingboard
630 PRINTTAB(11,3);CHR$(file%);CHR$
157;CHR$(file%);CHR$136;"F I L E "
;level%; "CHR$156
640 PRINTTAB(0,5);CHR$132;CHR$157;C
HR$131;"Selection";TAB(17)"Tries";TAB
(31)"Clues "CHR$156
650 ENDPROC
660 REM ***** play *****
670 DEFPROCplay
680 *FX15,0
690 PRINTTAB(13,19);CHR$134;"Colour
key"
700 A%=0:A=128:x%=10:REPEAT:A%=A%+1
:A=A+1:x%=x%+2
710 PRINTTAB(x%,20);CHR$(A);CHR$255
;" "
720 UNTIL A%=7
730 N=0:x1%=11:REPEAT:N=N+1:x1%=x1%
+2
740 PRINTTAB(x1%,21);N
750 UNTIL N=7
760 PRINTTAB(0,23);SPC(39);
770 A=0:X=18:REPEAT:A=A+1:X=X+2
780 PRINTTAB(0,23);CHR$131;"Enter "
;Cn%; " colour code"
790 VDU21
800 INPUTTAB(18,23);Ans%(A)
810 VDU6
820 IF Ans%(A)<1 OR Ans%(A)>7 THEN
SOUND1,-15,75,5:GOTO 780
830 VDU6:PRINTTAB(X,23)CHR$(Ans%(A)
+128);" "; "
840 UNTIL A=Cn%:Q=INKEY(50)
850 A=5-Cn%:B=0:REPEAT:A=A+2:B=B+1:
PRINTTAB(A,Y);CHR$(Ans%(B)+128);" ";:
UNTIL B=Cn%
860 PROCcheck
870 PRINTTAB(18,Y);CHR$135;Tries%;
880 PROCclues
890 ENDPROC
900 REM ***** check *****
910 DEFPROCcheck:M%=0
920 FOR A=1 TO Cn%
930 IF Ans%(A)=Col%(A) THEN Code$(A)
)="/":M%=M%+1:NEXT A:GOTO 980 ELSE
940 B=0:REPEAT:B=B+1
950 IF Ans%(A)=Col%(B) THEN Code$(A)
)="X":GOTO 960 ELSE
960 UNTIL B=Cn%
970 NEXT A
980 ENDPROC
990 REM ***** ans *****
1000 DEFPROCans
1010 PRINTTAB(5,3);SPC(20);:A=0:T=17
-Cn%
1020 REPEAT:A=A+1:T=T+2
1030 PRINTTAB(T,3);CHR$(Col%(A)+128)

```

```

+"*"; " ":UNTIL A=Cn%
1040 ENDPROC
1050 REM ***** clues *****
1060 DEFPROCclues
1070 xx%=31-Cn%:REPEAT:xx%=xx%+2
1080 X=RND(Cn%)
1090 IF rnd%(X)=1 THEN xx%=xx%-2:GO
TO 1120 ELSE rnd%(X)=1
1100 PRINTTAB(xx%,Y);CHR$131;Code$(X
)
1110 C=C+1
1120 UNTIL C=Cn%
1130 C=0
1140 FORA=1TDCn%:rnd%(A)=0:NEXT
1150 X=RND(Cn%)
1160 FORA=1TDCn%:rnd%(X)=0:NEXT
1170 FOR A=1 TO Cn%:Ans%(A)=0:Code$(
A)=".":NEXT
1180 ENDPROC
1190 REM ***** score *****
1200 VDU6
1210 DEFPROCscore
1220 *FX15,1
1230 IF attempts%<30 AND of%=6 PROC
newsflash:GOTO 1570
1240 IF err%=1 THEN 1260
1250 FORh%=10T011:PRINTTAB(10,h%);CH
R$130;CHR$136;CHR$141;"Mission Comple
ted":NEXT
1260 Z=INKEY(300):CLS
1270 FORh%=0T01:PRINTTAB(10,h%);CHR$
131;CHR$141;"Mission Report":NEXT
1280 RESTORE 2570:A=0:REPEAT:A=A+1
1290 READ Mes$(A)
1300 UNTIL A=23
1310 PRINTTAB(0,4);CHR$134;"Ref Agen
t";CHR$134;Name$
1320 PRINTTAB(10,6);CHR$134;"Classi
fied";CHR$133;"Secret"
1330 PRINTTAB(0,9);CHR$134;Mes$(1);
:IF err%=1 THENPRINTTAB(20,9);CHR$13
3;Mes$(2);CHR$134;Mes$(4) ELSE PRINT
TAB(20,9);CHR$130;Mes$(3);CHR$134;M
es$(4)
1340 PRINTTAB(0,10);CHR$134;Mes$(5)
;CHR$131;of%;CHR$134;Mes$(6)
1350 PRINTTAB(0,11);CHR$134;Mes$(7)
:PRINTTAB(4,11);CHR$131;attempts%;CH
R$134;Mes$(8)
1360 PRINTTAB(0,12);CHR$134;CHR$13
4;Mes$(9)
1370 IF err%=1 THEN PRINTTAB(17,12);
;CHR$133;Mes$(10);CHR$134;Mes$(12) E
LSE PRINTTAB(17,12);CHR$130;Mes$(11)
;CHR$134;CHR$134;Mes$(12)
1380 PRINTTAB(0,13);CHR$134;Mes$(13)
1390 IF of%<6 THEN 1400 ELSE GOTO 14
40
1400 d=RND(3)
1410 IF d=1 THEN D$=Mes$(14):GOTO 14
80

```

```

1420 IF d=2 THEN D$=Mes$(15):GOTO 14
80
1430 IF d=3 THEN D$=Mes$(16):GOTO 14
80
1440 IF of%=6 AND attempts%>80 THEN
D$=Mes$(17):GOTO 1480
1450 IF of%=6 AND attempts%<71 AND a
ttempts%>60 THEN D$=Mes$(18):GOTO 148
0
1460 IF of%=6 AND attempts%<61 AND a
ttempts%>50 THEN D$=Mes$(19):GOTO 148
0
1470 IF attempts%<50 THEN D$=Mes$(20
):GOTO 1480
1480 PRINTTAB(3,13);CHR$131;D$
1490 PRINTTAB(0,14);CHR$134;Mes$(21
)
1500 PRINTTAB(6,17);CHR$131;Mes$(22)
1510 PRINTTAB(0,19);CHR$129;CHR$13
6;Mes$(23)
1520 Z=INKEY(2000)
1530 CLS:PROCtitle
1540 FORh%=10T011:PRINTTAB(8,h%);CHR
$131;CHR$136;CHR$141;"Message destroy
ed":NEXT
1550 Z=INKEY(200)
1560 PROCscoretable
1570 REPEAT:PRINTTAB(4,21);CHR$130;"
There's another mission for you":TAB(
10);CHR$130;"Do you accept Y/N";Z$=G
ET$
1580 Z%=INSTR("YyNn",Z$)
1590 UNTIL (Z%>0)
1600 IF Z%=1 OR Z%=2 THEN GOTO 110 E
LSE CLS:PRINT"Basic":END
1610 END
1620 REM ***** agent *****
1630 DEFPROCagent
1640 PROCtitle
1650 PRINTTAB(0,4);CHR$134;"Before y
ou can become an Agent you must";CHR$
134;"state your name for our files. T
hen you ";CHR$134;"will be considered
for a mission."
1660
1670 PRINTTAB(1,10);CHR$131;"Please
state your name."
1680 INPUTTAB(27,10)Name$:Name$=LEFT
$(Name$,18)
1690 PRINTTAB(1,13);CHR$130;Name$;"
you are being considered"
1700 Z=INKEY(400)
1710 ENDPROC
1720 REM ***** title *****
1730 DEFPROCtitle:CLS
1740 FOR h%=0 T01:PRINTTAB(7,h%);CHR$
129;CHR$157;CHR$135;CHR$141" M a s t
e r S p y "CHR$156:NEXT h%
1750 ENDPROC
1760 REM ***** report *****
1770 DEFPROCreport

```



```

1780 FOR h% = 10 TO 11: PRINT TAB(10, h%); CH
R$131; CHR$136; CHR$141; "Mission Report
": NEXT
1790 ENDPROC
1800 REM ***** newflash *****
1810 DEFPROC newflash
1820 Name$ = "john": attempts% = 23
1830 CLS: FOR h% = 10 TO 11: PRINT TAB(12,
h%); CHR$141; CHR$131; CHR$136; "NEWSFLASH
": NEXT h%
1840 Z = INKEY(200): CLS
1850 PROC title
1860 PRINT TAB(13, 3); CHR$131; CHR$136;
"Newsflash"
1870 PRINT TAB(0, 5); CHR$134; "Ref: Age
nt"; CHR$131; Name$
1880 PRINT TAB(0, 7); CHR$134; "Congrat
ulation You opened all 6 files"; CHR$
134; "in only "; attempts% " attempts.
How ever 'M' is "; CHR$134; "puzzled as
it is almost impossible to "; CHR$134
; "do that. Therefore you are"; CHR$133
; "Dismissed";
1890 PRINT CHR$134; "from the service
immediately."
1900 Z = INKEY(200): PRINT "; CHR$131; "H
ow ever 'M' thinks you should have" TAB
(12); CHR$131; "a second chance": GOTO 1
570
1910 END
1920 REM ***** score table *****
1930 DEFPROC scoretable
1940 PROC title: PRINT TAB(9, 3); CHR$134
; "Agent's League Table": PRINT TAB(9, 4)
; CHR$134; STRING$(20, "-")
1950 FOR N% = 1 TO 5
1960 PRINT TAB(5, (N% * 2) + 6); CHR$134; P%
(N%); CHR$131; HI$(N%); H%(N%): NEXT
1970 COLOUR 3: IF attempts% > H%(5) OR a
ttempts% < 6 ENDPROC
1980 N% = -1: REPEAT: N% = N% + 1: UNTIL atte
mpts% < H%(N%) OR N% = 5: IF N% = 5 OR err% =
1 ENDPROC
1990 H%(N%) = attempts%: PRINT TAB(12, 19
); VDU 141, 136: PRINT CHR$134; "Hi-score
!!!": TAB(12, 20); VDU 141, 136: PRINT CHR$
134; "Hi-score !!!"
2000 PRINT TAB(12); CHR$134; "Enter yo
ur name-" TAB(13);
2010 VDU 31, 32: INPUT A$: IF LEN(A%) > 18
A% = LEFT$(A%, 18): REPEAT: A% = A% + ".": UNTI
L LEN(A%) = 24 ELSE REPEAT: A% = A% + ".": UN
TIL LEN(A%) = 23
2020 HI$(N%) = A%: attempts% = 100: CLS: GO
TO 1940
2030 ENDPROC
2040 REM ***** instructions *****
2050 DEFPROC instructions
2060 PROC title
2070 PRINT TAB(0, 4); CHR$130; "Briefing
for Agent "; Name$;

```

```

2080 PRINT "" Your mission is to g
ain access to 6"" files held in the
Udanddy Embassy."
2090 PRINT "" Each file is in a sep
arate safe with"" its own colour-co
de lock and is graded by the classif
ication of the file it"" is protect
ing."
2100 PRINT "; CHR$131; "File No." TAB(12
) "Colours in the code-lock"
2110 PRINT TAB(2); CHR$134; "1" TAB(10)
"3 colours No duplication"
2120 PRINT TAB(2); CHR$134; "2" TAB(10) "
4 colours No duplication"
2130 PRINT TAB(2); CHR$134; "3" TAB(10) "
5 colours No duplication"
2140 PRINT TAB(2); CHR$134; "4" TAB(10) "
3 colours May be duplicated"
2150 PRINT TAB(2); CHR$134; "5" TAB(10) "
4 colours May be duplicated"
2160 PRINT TAB(2); CHR$134; "6" TAB(10) "
5 colours May be duplicated"
2170 PROC newpage
2180 CLS: PROC title
2190 PRINT TAB(0, 4); CHR$130; "Briefing
for Agent "; Name$;
2200 PRINT "" The colours to choose
from are as shown below together with
their appropriate"" number-key."
2210 A% = 0: A% = 128: X% = 10: REPEAT: A% = A% + 1
: A% = A% + 1: X% = X% + 2
2220 PRINT TAB(X%, 11); CHR$(A%); CHR$255
; " ";
2230 UNTIL A% = 7
2240 N% = 0: X% = 11: REPEAT: N% = N% + 1: X% = X% + 2
2250 PRINT TAB(X%, 12); N%: UNTIL N% = 7
2260 PRINT "" You are given 10 tries
at each code. If you do not open a f
ile by the 10th try the code is reve
aled, but you go BACK a file. If you
are taking too many goes for the nu
mber of open files or if it"" looks
like you could";
2270 PRINT " be caught then the missi
on controller will terminate your mi
ssion."
2280 PROC newpage
2290 PROC title
2300 PRINT TAB(0, 4); CHR$130; "Briefing
for Agent "; Name$;
2310 PRINT "" There are clues to hel
p you crack the codes. Memorize the
following."" CHR$131; " / = righ
t colour right place" CHR$131; "
X = right colour wrong place" CHR$13
1; " . = colour not used."
2320 PRINT "" However these clues are
mixed up e.g. clue 1 may not relat
e to colour 1, etc. If you are good y
ou can be promoted."
2330 PRINT "" That concludes the brie

```

```

fing for this"" mission "; Name$; TAB
(15); "Good luck"
2340 PRINT TAB(7); CHR$131; "Signed Mi
ssion controller"
2350 PROC newpage
2360 ENDPROC
2370 END
2380 REM ***** newpage *****
2390 DEFPROC newpage
2400 PRINT TAB(3, 23); CHR$130; "Press t
he <Space-bar> to continue"
2410 Z = GET$: IF Z <> " " VDU 7: GOTO 240
0
2420 ENDPROC
2430 REM ***** err *****
2440 DEFPROC err: VDU 6: CLS
2450 SOUND 1, -15, 75, 5: SOUND 2, -15, 100,
5
2460 IF ERR = 17 THEN err% = 1: PROC repor
t: PROC score: END
2470 *FX 4, 0
2480 CLS: PRINT: REPORT: PRINT " at line
No. "; ERL
2490 PRINT "; CHR$131; "Dear Player"";
CHR$131; " I am sorry that an error ha
s appeared"; CHR$131; "in the program.
If you cannot correct"; CHR$131; "it
please let me know. Tell me the";
CHR$131; "error and the line number in
which it";
2500 PRINT CHR$131; "occurred." CHR$1
31; "Write to me at this address:-"
2510 PRINT TAB(12); CHR$134; "Mr J. Hol
lings" TAB(12); CHR$134; "156 Templewoo
d" TAB(12); CHR$134; "Walter Ash" TAB(1
2); CHR$134; "High Wycombe" TAB(12); CHR
$134; "Bucks."
2520 PRINT: PROC insert("LIST" + STR$(ER
L) + CHR$13): STOP
2530 END
2540 DEFPROC insert(I%): REPEAT: A% = 138
: X% = 0: Y% = ASC(LEFT$(I$, 1)): CALL &FFFF4:
I% = RIGHT$(I$, LEN(I%) - 1): UNTIL I% = "" : E
NDPROC
2550 END
2560 REM ***** DATA *****
2570 DATA "As your mission was", "Term
inated", "Completed", "and the", "number
of files opened was", "which took", "y
ou", " attempts. It is the decision of
"
2580 DATA "M' that you are", "Demoted
", "Promoted", "to the rank", "of", "Clea
ner (at Hoover's)", "Supervisor (in a
Siberian Salt mine)", "Builder (for Le
go)", "Assistant Spy", "Spy", "Super Spy
", "Master Spy"
2590 DATA "This is effective immediat
ely.", "Signed Mission Controller", "Th
is message self-destructs in 30 Sec"
2600 END

```



Draw up a chair and join in this poker  
simulation game by **ALAN GORNALL**

# Deal your way to a right royal flush

**DRAW** is a computer simulation of draw poker, from which all other forms of the game are derived.

First each player chips in his or her ante or stake into the kitty or pot. Then each is dealt five cards face down.

In real life the player to the dealer's left either opens the betting or folds, so losing his ante. The next player in turn may "see" the amount previously put into the pot, or raise it, or fold. And so on until we get back to the dealer.

If the pot has been raised the betting goes around a second time. Next comes the draw.

Each player can change any or all of the five cards in his hand. Once everyone has been dealt the final round of betting follows.

The same process as before applies and the last person left in collects the pot.

*Poker hands are ranked like this, with the best hand first:*

Royal flush (ace to ten in the same suit), straight flush (five cards in sequence of the same suit), four of a kind (say, four kings), full house (three cards of one kind and two of another), Flush (all of the same suite), Straight (consecutive cards, different suits), three of a kind (say three fives), two pairs, one pair.

The simulation, however, involves only two players with the computer acting as dealer. So if one player drops out the hand will be over.

*A few tips for the player:*

The only way to play poker, other than by cheating – which incidentally the simulation is quite adept at when called upon – is to make the most out of your winning hands, and lose as little as possible on your losing ones. Unfortunately, this is hard to do.

One way is to glean as much as possible from your opponent's style of play and act accordingly.

Of course he may play erratically, which would not be much help.

You will be pleased to hear that the simulation does play to a fixed system,

the nature of which you'll have to work out for yourself.

There are two pretty good indicators as to the strength of an opponent's hand. One is how much money he is prepared to put on it – although he may bluff – and the other is how many cards he takes at the draw.

If he takes three cards, the best hand he could have had before the draw would be a pair, and his odds of improving this are about 5-2 against. So if you have two pairs, chances are you'll beat him in the showdown.

In the game the ante is 1, the maximum open 10 and the maximum raise 10.



## PROCEDURES

Values any poker hand in order of the accepted ranks. This uses other procedures, such as: PROCstraight, PROCswap, PROCflush, PROCpremanip, PROCa (deals with hands of one pair), PROCflush2, PROCb (hands of two pairs), PROCc (three of a kind), PROCD (full house), PROCflush1, PROCit, PROCe (four of a kind). Deals the specified hand. Starting at the left of the screen at a Y – coordinate of Y%. This uses PROCcard(X%,Y%) to draw the card in the correct place.

Draws the value of the card on its face. Displays the value in the pot, and how much you are in credit.

Control the betting rounds. Which one is used depends on whose turn it is to open first. Decide the computer betting scheme. The former is used before the draw has taken place, and the latter after it has taken place.

Works out the percentage of hands that the computer's opponent has bluffed on.

Called when someone has been seen after the draw has taken place.

Controls which cards you wish to change, and how many the computer wishes to change.

Called when the name entered at the beginning of the program matches up with one of those in one of the data statements at the end of the program. It also sets its betting scheme, and if its hand is not as good as its opponent's hand it will alter it using PROCalter, and if it still isn't as good, it will change it using PROCalter1.

PROCvalue(Z\$)

PROCdeal(HAND\$,Y%)

PROCval  
PROCinfo

PROCbet  
PROCbet1  
PROCpre\_bet  
PROCpre\_bet1

PROCbluff

PROCwin

PROCdraw

PROCcheat





```

10 REM Draw-Poker by Alan Gornall
20 REM Copyright Micro User
30 REM 1984
80 ON ERROR GOTO 5650
90 MODE 5
100 VDU19,2,2,0,0,0
110 VDU 23,240,8,28,28,107,127,107,
8,28
120 VDU 23,241,8,28,62,127,62,28,8,
0
130 VDU 23,242,54,127,127,127,62,28
,8,0
140 VDU 23,243,8,28,62,127,127,127,
28,62
150 DIM ST$(4)
160 ST$(1)=CHR$ 18+CHR$ 0+CHR$ 0+CH
R$ 240
170 ST$(2)=CHR$ 18+CHR$ 0+CHR$ 1+CH
R$ 241
180 ST$(3)=CHR$ 18+CHR$ 0+CHR$ 1+CH
R$ 242
190 ST$(4)=CHR$ 18+CHR$ 0+CHR$ 0+CH
R$ 243
200 DIM YY$(3),A$(52),Z$(5),type$(1
4),local$(2),ZX$(5),CX$(52),AZ$(4,5)
210 STAKE%=50:bet%=0:raise%=0:bluff
%=FALSE:no%=10:BLUFF%=1:POT%=0
220 Pack$="AC2C3C4C5C6C7C8C9CTCJCQC
KCAD2D3D4D5D6D7D8D9DJDJDDKDAH2H3H4H5
H6H7H8H9HTHJHQHKHAS2S3S4S5S6S7S8S9STS
JSQSKS"
230
240 PRINT "What is your name":INPUT
Opponent$
250 REPEAT
260 READ F$
270 UNTIL F$="****"OR F$=Opponent$
280 IF F$=Opponent$ THEN cheat%=1 E
LSE cheat%=FALSE
290
300 REPEAT
310 IF STAKE% > 500 THEN cheat%=1
320 fold%=FALSE
330 REPEAT
340 FOLD%=FALSE:flush%=FALSE:BLOP%=
FALSE:FOLDED%=FALSE:taken%=FALSE:draw
%=FALSE:stiff%=0:run%=FALSE:ace%=FALS
E:position%=FALSE
350 CLS:PRINT TAB(2,26);"Shuffling
Cards"
360 FOR AX= 1 TO 52
370 CX(AZ)=FALSE
380 NEXT
390 SEED%=RND(-TIME)
400 FORAX=1TO52
410 REPEAT
420 BX=RND(52)
430 UNTIL CX(BX) = FALSE
440 EX=BX*2-1:CX(BX)=CX(BX)+1:A$(AX
)=MID$(Pack$,EX,2):NEXT
450 Hand1$=A$(1)+A$(3)+A$(5)+A$(7)+

```

## VARIABLES

**Pack\$** Contains a pack of cards.

**STAKE%** Opponent's credit.

**POT%** Amount of money currently in the pot.

**bluff%** Percentage of hands that the opponent has bluffed on.

**no%** Number of hands that have so far been played, with nine added on, to make calculations easier.

**BLUFF%** Number of hands that the computer's opponent has bluffed on.

**Opponent\$** Name of the computer's opponent.

**FOLD%** Variable by which computer decides whether it will fold or not, if 1, it will fold, if 0, it will not.

**cheat%** Variable by which computer decides whether it will cheat or not, if 1, it will cheat, if 0, it will not.

**opt%** What the computer's opponent has decided to do.

**draw%** Whether the draw has taken place or not, if 1, it has, if 0, it has not.

**Count%** Card at the top of the pack.

**A\$(52)** After the cards have been shuffled they are placed here, one in each location.

**Hand1\$** Opponent's hand.

**Hand2\$** Computer's hand.

**fold%** Determines who has the write to fold and not lose their ante. If 1, it is the computer's turn, if 0, it is the opponent's turn.

**value** The value of the hand last used in PROCvalue.

All of the other variables are used in FOR...NEXT loops, are restricted to a small part of the program, or are of little importance.



## From Page 15

```

A$(9)
460 Hand2$=A$(2)+A$(4)+A$(6)+A$(8)+
A$(10)
470 CountX=11
480 STAKEX=STAKEX-1:POTX=POTX+2
490 aboX=FALSE
500
510 COLOUR 130
520 CLS
530 PRINT:COLOUR0
540 PRINT"Your hand : "
550
560 PROCdeal(Hand1$,700)
570
580 PROCvalue(Hand2$)
590
600 PROCinfo
610 IF foldX THEN 3380 ELSE 3470
620 PROCinfo
630
640 noX=noX+1
650
660 IF foldX THEN PROCbet ELSE PROC
bet1
670
680 IF (FOLDX=1 AND cheatX=FALSE) O
R (FOLDEDX = 1 AND foldX=0) THEN 800
690 IF foldX=1 AND FOLDEDX=1 THEN 8
20
700
710 VDU 26
720 PROCdraw
730
740 drawX=1
750
760 PROCvalue(Hand2$)
770
780 IF foldX THEN PROCbet ELSE PROC
bet1
790
800 PROCbluff
810
820 foldX = foldX +1
830 UNTIL foldX>1
840 foldX=0
850 UNTIL FALSE
860 END
870
880 DEFPROCvalue(Z$)
890
900 flushX=0
910 Y$=Z$
920 value=0
930
940 PROCpremanip
950
960 PROCflush
970
980 NX=FALSE
990 FOR JX=1 TO 4

```

```

1000 FOR KX=JX+1 TO 5
1010 IF MID$(Z$,2*JX-1,1)=MID$(Z$,2*
KX-1,1) THEN NX=NX+1:AX(JX,KX)=1
1020 NEXT
1030 NEXT
1040 IF NX=1 THEN PROCa:ENDPROC
1050 IF NX=2 THEN PROCb:ENDPROC
1060 IF NX=3 THEN PROCc:ENDPROC
1070 IF NX=4 THEN PROCd:ENDPROC
1080 IF NX=6 THEN PROCe:ENDPROC
1090
1100 PROCflush
1110
1120 REM flush
1130
1140 FOR JX= 1 TO 4
1150 FOR KX= JX+1 TO 5
1160 IF MID$(Z$,2*JX,1) = MID$(Z$,2*
KX,1) THEN NX=NX+1:AX(JX,KX)=1
1170 NEXT
1180 NEXT
1190 IF NX=10 THEN flushX=1
1200
1210 PROCstraight
1220 IF value > 0 THEN ENDPROC
1230
1240 REM bust
1250
1260 ZZ$=""
1270 FOR JX=1 TO 9 STEP 2
1280 IF MID$(Z$,JX,1)="" THEN value
= 13:ZZ$=ZZ$+"K"+MID$(Z$,JX+1,1)
1290 IF MID$(Z$,JX,1)="" OR MID$(Z$,
JX,1)="" THEN value = 14:ZZ$=ZZ$+"A"+
MID$(Z$,JX+1,1)
1300 NEXT
1310 IF value<13 AND cheatX=0 THEN F
OLDX = 1
1320
1330 ENDPROC
1340
1350 DEF PROCstraight
1360
1370 FOR JX=1 TO 5:Z$(JX)="" :Z$(JX)=
MID$(Z$,2*JX-1,2):NEXT
1380 FOR KX=1 TO 4
1390 leastX=KX
1400 FOR testX = KX+1 TO 5
1410 IF Z$(testX) < Z$(leastX)
THEN leastX = testX
1420 NEXT
1430
1440 PROCswap
1450
1460 NEXT
1470 Z$=""
1480 FOR JX=1 TO 5
1490 Z$=Z$+Z$(JX)
1500 NEXT
1510 ZZ$=Z$
1520 BBX=0
1530 AA$=ASC(MID$(Z$,1,1))

```

```

1540 FOR JX=5 TO 2 STEP -1
1550 BBX=BBX+(ASC(MID$(Z$,2*JX-1,1))
-AA$)
1560 NEXT
1570 IF BBX=10 AND flushX=0 THEN val
ue = (AA$-44)*20000
1580 IF BBX=10 AND flushX=1 THEN val
ue = (((AA$-44)*10)+30000200)
1590 IF BBX>10 AND flushX=1 THEN val
ue = (ASC(Z$(5))-48)*200000+(ASC(Z$(4
))-48)*10000+(ASC(Z$(3))-48)*500+(ASC
(Z$(2))-48)*25+(ASC(Z$(1))-48)
1600 IF BBX>10 AND flushX=1 AND aceX
=1 AND runX=FALSE THEN Z$=LEFT$(IT$,p
ositionX-1)+"1"+RIGHT$(IT$,10-positio
nX):runX=1:PROCstraight:runX=FALSE
1610 IF BBX>10 AND flushX=FALSE AND
aceX=1 AND runX=FALSE THEN Z$=LEFT$(I
T$,positionX-1)+"1"+RIGHT$(IT$,10-pos
itionX):runX=1:PROCstraight:runX=FALS
E
1620 ENDPROC
1630
1640 DEF PROCpremanip
1650 FOR JX=1 TO 9 STEP 2
1660 z$=Z$
1670 IF MID$(z$,JX,1)="" THEN Z$=LEF
T$(z$,JX-1)+"":RIGHT$(z$,10-JX)
1680 IF MID$(z$,JX,1)="" THEN Z$=LEF
T$(z$,JX-1)+"":RIGHT$(z$,10-JX)
1690 IF MID$(z$,JX,1)="" THEN Z$=LEF
T$(z$,JX-1)+"<"+RIGHT$(z$,10-JX)
1700 IF MID$(z$,JX,1)="" THEN Z$=LEF
T$(z$,JX-1)+"="+RIGHT$(z$,10-JX)
1710 IF MID$(z$,JX,1)="" THEN Z$=LEF
T$(z$,JX-1)+">"+RIGHT$(z$,10-JX):IT$=
Z$:aceX=1:positionX=JX
1720 NEXT
1730 y$=Z$
1740 ENDPROC
1750
1760 DEF PROCflush
1770
1780 FOR JX=1 TO 4
1790 FOR KX=1 TO 5
1800 AX(JX,KX)=FALSE
1810 NEXT
1820 NEXT
1830 ENDPROC
1840
1850 DEF PROCflush1
1860 FOR JX=0 TO 14
1870 typeX(JX) = FALSE
1880 NEXT
1890 ENDPROC
1900
1910 DEF PROCflush2
1920 FOR JX=1 TO 5:Z$(JX)=FALSE:NEXT
1930 ENDPROC
1940
1950 DEF PROCit
1960 FOR JX=1 TO 5

```



```

1970 KZ=(ASC(MID$(Z$,2*JX-1,1)))-48
1980 typeZ(KZ) = typeZ(KZ) + 1
1990 NEXT
2000 ENDPROC
2010
2020 DEF PROCswap
2030 temp$ = Z$(KZ)
2040 Z$(KZ) = Z$(leastZ)
2050 Z$(leastZ) = temp$
2060 ENDPROC
2070
2080 DEF PROCa
2090
2100 PROCflush2
2110
2120 FOR JX=1 TO 4
2130 FOR KZ=1 TO 5
2140 IF AZ(JX,KZ)=1 THEN CX=JX:DX=KZ

2150 NEXT
2160 NEXT
2170 ZZ$=MID$(Y$,2*CX-1,2)+MID$(Y$,2
+DX-1,2)
2180 ZX(CX)=1:ZX(DX)=1:YY$=""
2190 FOR JZ= 1 TO 5
2200 IF ZX(JZ)=FALSE THEN YY$=YY$+MI
D$(Y$,2*JZ-1,1)
2210 NEXT
2220 FOR JX=1 TO 3
2230 YY$(JX)=MID$(YY$,JZ,1)
2240 NEXT
2250 IF YY$(1)<YY$(2) THEN T$=YY$(2)
:YY$(2)=YY$(1):YY$(1)=T$
2260 IF YY$(2)<YY$(3) THEN T$=YY$(3)
:YY$(3)=YY$(2):YY$(2)=T$
2270 IF YY$(1)<YY$(2) THEN T$=YY$(2)
:YY$(2)=YY$(1):YY$(1)=T$
2280 value = (ASC(MID$(z$,2*CX-1,1))
-48)*100+(ASC(YY$(1))-48)+(ASC(YY$(2)
)-48)*.01+(ASC(YY$(3))-48)*.0001
2290 ENDPROC
2300
2310 DEF PROCb
2320
2330 PROCflush2
2340
2350 ZZ$="":ABCZ=FALSE
2360 FOR JZ=1 TO 4
2370 FOR KZ=1 TO 5
2380 IF AZ(JZ,KZ)=1 THEN ZZ$=ZZ$+MID
$(Y$,2*JZ-1,2)+MID$(Y$,2*KZ-1,2):ZX(J
Z)=1:ZX(KZ)=1:ABCZ=ABCZ+1:localZ(ABCZ
)=ASC(MID$(Z$,2*JZ-1))-48
2390 NEXT
2400 NEXT
2410 FOR JZ=1 TO 5
2420 IF ZX(JZ)=FALSE THEN localZ = A
SC(MID$(Z$,2*JZ-1,1))-48
2430 NEXT
2440 IF localZ(1)>localZ(2) THEN val
ue = localZ(1)*500+localZ(2)*20+local
Z ELSE value = localZ(2)*500+localZ(1
)*20+localZ
2450 ENDPROC
2460
2470 DEF PROCc
2480 ZZ$=""
2490 FOR JX=1 TO 5
2500 ZX(JX)=FALSE
2510 NEXT
2520 FOR JX=1 TO 4
2530 FOR KX=1 TO 5
2540 IF AX(JX,KX)=1 THEN ZX(JX)=1:ZX
(KX)=1
2550 NEXT
2560 NEXT
2570 FOR JX=1 TO 5
2580 IF ZX(JX)=1 THEN ZZ$=ZZ$+MID$(Y
$,2*JX-1,2):CBAZ=JX
2590 NEXT
2600 value = (ASC(MID$(Z$,2*CBAZ-1,1
))-48)*5000
2610 ENDPROC
2620
2630 DEF PROCd
2640
2650 PROCflush1
2660
2670 PROCit
2680
2690 FOR JX=1 TO 14
2700 IF typeZ(JX) = 3 THEN value = J
X*2E6
2710 NEXT
2720 ZZ$=Y$
2730
2740 ENDPROC
2750
2760 DEF PROCe
2770
2780 PROCflush1
2790
2800 PROCit
2810
2820 FOR JX=1 TO 14
2830 IF typeZ(JX) = 4 THEN value = (
JX*10)+3E7
2840 NEXT
2850 ZZ$=Y$
2860
2870 ENDPROC
2880
2890 DEF PROCdeal(HAND$,YX)
2900 blipZ = 1
2910 FOR XX=100 TO 900 STEP 200
2920 K$=MID$(HAND$,blipZ,2)
2930 PROCcard(XZ,YX)
2940 PROCval
2950 blipZ = blipZ + 2
2960 NEXT XX
2970 VDU 4
2980 ENDPROC
2990
3000 DEF PROCcard(XZ,YX)
3010 BCOL 0,3
3020 MOVE XZ,YX
3030 MOVE XZ+160,YX
3040 PLOT 85,XZ,YX+200
3050 PLOT 85,XZ+160,YX+200
3060 BCOL 0,0
3070 ENDPROC
3080
3090 DEF PROCval
3100 X$=MID$(K$,1,1)
3110 VDU 5
3120 IF X$="T" THEN MOVE XZ+24,YX+11
2:PRINT"10":GOTO3140
3130 MOVE XZ+8,YX+180:PRINT X$:MOVE
XZ+104,YX+40:PRINT X$
3140 X$=MID$(K$,2,1)
3150 IF X$="C" THEN NX=1
3160 IF X$="D" THEN NX=2
3170 IF X$="H" THEN NX=3
3180 IF X$="S" THEN NX=4
3190 MOVE XZ+88,YX+180:PRINT ;ST$(NX
):MOVE XZ+8,YX+40:PRINT ;ST$(NX)
3200 ENDPROC
3210
3220 DEF PROCinfo
3230 VDU28,7,30,18,29
3240 COLOUR 131
3250 CLS
3260 COLOUR 0
3270 PRINT"pot ";POTX
3280 PRINT"credit ";
3290 IF STAKEZ < 0 THEN COLOUR 1
3300 PRINT;STAKEZ;
3310 COLOUR 0
3320 COLOUR 130
3330 VDU 26
3340 ENDPROC
3350
3360 REM 1 fold
3370
3380 *FX15
3390 IF value < 500 AND cheatZ=FALSE
THEN PRINT TAB(0,25);"I fold":O$=INK
EY$(500):GOTO 820
3400 PRINT TAB(0,25);"I am not going
to fold"
3410 O$=INKEY$(500)
3420 PRINT TAB(0,25);SPC(30)
3430 GOTO 630
3440
3450 REM you fold
3460
3470 PRINT TAB(0,25);"Do you want to
fold"
3480 REPEAT:O$=GET$:UNTIL O$="Y" OR
O$="N"OR O$="y"OR O$="n"
3490 IF O$="Y"OR O$="y" THEN 820
3500 PRINT TAB(0,25);SPC(30)
3510 GOTO 630
3520

```





# SATAN'S CHALLENGE

or (Nevil Rides Out) A Black Magic Adventure



Dare you take up the challenge laid down by the most evil and sinister of all beings... the devil himself.

If you do you will find yourself alone and at the mercy of the twisted fancies and whims of a cold and calculating mind.

Occult forces are threatening the lives of those near and dear to you. Their only salvation rests in your hands but in accepting the challenge your own existence is put in severe peril.

In taking up the challenge you have to find the TALISMAN and locate a pentagram which then has to be prepared for the final rite. In the mean time dark forces will be opposing you making a difficult task almost impossible.

Do not allow yourself to be lulled into a sense of security for it will be short lived.

Many have gone before only to swell the ranks of the damned.

This is the latest adventure from the stables of Microtest and has been written with the acclaimed features of other adventures in mind eg save facility, quick response, simple but extensive commands, a mixture of logical and fiendish problems to solve.

Be warned this is an easy adventure to get into but devilishly difficult to end.

**CASSETTE £7.95 inc VAT**

**DISC 40 or 80 tk £9.45 inc VAT (add 50p p&p)**

## DAIRYFILE FOR DAIRY FARMERS

Keep on that economic line between over and under feeding!

Save time recording milk yield and calculating feed amounts!

Quickly decide feeding policy with the 'Monthly Calving Group' Performance Graph!

Print out a recording sheet with cows in numerical order. Print out graphs or tables of individual cows or Monthly Groups showing serving and pregnancy details, illness record, Calving Index, weekly and running total Margin over Concentrate. See at a glance cows due for serving, pregnancy diagnosis and drying off. Keep track of

weekly total feed cost and milk sale price.

All this and more with DAIRYFILE.

Predict cow or Monthly Group total lactation yield. Compare with Standard Lactation Curves. All old data preserved - compare Monthly Groups over the years. Which is the most profitable month?

Find out with DAIRYFILE - for up to 200 lactating cows.

Available on dual 40 tk disc (For dual 40 tk drives only)

Dual or Single 80 tk drives **£69 inc VAT (add 50p p&p)**

Please state on ordering which disc format and printer model.

## MICROTEST FONT ROM

This exciting new ROM from Microtest will enable you to get all sorts of new characters and fonts from your BBC Computer. Once you have produced your masterpiece on the screen, all you have to do is use the inbuilt screen-dump utility to produce a hard copy onto paper.

Typing '\*HELP FONTS' gives a list of available fonts and the blocks of characters which they replace.

Available fonts are:

- \*Accents Accents and miscellaneous.
- \*Block Small capitals.
- \*Data Like the bottoms of cheques.
- \*Greek It's all Greek to me too!
- \*Joined Standard capitals with joined up lower case.
- \*Maths A mix of until now unobtainable Mathematical symbols.
- \*Miscellaneous A few oddities which often are very necessary.
- \*Thick Thick text (for MODEs 0&3) to enhance 80 column mode.

### \*Thin

Thin text (for MODEs 2&5) which makes modes 2 & 5 much more readable or perhaps "READABLE".

### \*Vertical

For labelling graphs.

### \*MODE 8

Ten column multicolour, memory-miserly mode.

### \*S DUMP

A dump facility which will produce a screen dump of any MODE from 0 to 8 (including a text only dump in MODE 7) on an Epson, Star, CTI CP80 or MT80.

The ROM uses absolutely NO user memory and can be used with word processors etc. as well as normal BASIC programs. The DUMP utility will produce positive or negative graphics and will also rotate the picture produced through 90° as well as being able to position the picture anywhere laterally across the sheet of paper.

**£17.50 inc VAT (add 50p p&p)**

## Microtest Starstick ROM & Joystick Package

Now available the Starstick ROM and Joystick. This comes in three forms:



### (A) The Starstick ROM and Quickshot I Joystick

**Price 17.95 + VAT = 19.84**

### (B) The Starstick ROM and Quickshot II Joystick

**Price 19.25 + VAT = 22.14**

### (C) The Starstick ROM and patch lead, choose your own Spectrum/Atari style joystick

**Price 15.25 + VAT = 17.54**

**Post and Packing £1.00 inc. VAT per item**

This enables you to plug the Rapid action self centring joysticks until now only available for the Spectrum/Atari/CBM machines into the user port of the BBC. Model A users please note NO ANALOGUE INTERFACE REQUIRED.

Disc Users Note - pressing BREAK, SHIFT-BREAK or CONTROL BREAK does not modify or destroy the STARSTICK software so Disc Users please feel free to Boot!

The software patch provided in the ROM is interrupt driven and adds the following commands to your computer.

- \* STICK turn on the STARSTICK ROM
- \* NSTICK turn off the STARSTICK ROM
- \* SETSTICK set up joystick to users spec
- \* SAVE "NAME" 140 160 saves your user-key protocols
- \* ADVAL emulate standard analogue joysticks
- \* PAUSE define key to Freeze game
- \* NPAUSE turn off ability to freeze game
- \* "NAME" predefined key protocols set up for software houses programs
- \* HELP KEYS displays currently selected key protocols
- \* REPEAT enables auto-repeat fire
- \* NREPEAT disables auto-repeat fire

**Enables you to use our joysticks on most programs even those that do not offer joystick capability.**



available from

**MICROTEST LTD**

**18 Normandy Way, Bodmin, Cornwall PL31 1EX**

**Telephone: 0208 3812**

OR ANY GOOD DEALER

Large picture shows BBC Computer System and a Quickshot II Joystick. Small inset just a few of the joysticks that will work with the patch lead. Screenshot by kind permission of SUPERIOR SOFTWARE

**All MICROTEST ROMs are fully TUBE® compatible. (TUBE® is a reg. trademark of Acorn Computers).**

DEALER ENQUIRIES  
and  
EXPORT ORDERS  
WELCOME

VISA





## From Page 17

```

3530 DEF PROCbet
3540
3550 IF cheat% THEN PROCcheat ELSE I
F draw% THEN PROCpre_bet1 ELSE PROCpr
e_bet
3560
3570 VDU 28,1,27,18,22
3580 COLOUR 128:CLS:COLOUR3
3590
3600 IF FOLD% = 1 AND cheat%=FALSE T
HEN PRINT "I fold":STAKE%=STAKE%+POT%
:POT%=FALSE:PROCinfo:0$=INKEY$(500):E
NDPROC
3610 PRINT"I open with ";BET%=bet%+
INT(RND(bet%)):PRINT;BET%
3620 POT%=POT%+BET%:PROCinfo
3630 PROC2nd_bit:ENDPROC
3640
3650 DEF PROC2nd_bit
3660 VDU 28,1,27,18,24
3670 COLOUR3:COLOUR128
3680 PRINT"1. See"
3690 PRINT"2. Raise"
3700 PRINT"3. Fold"
3710 REPEAT:opt%=BET:UNTIL opt%>48 A
ND opt%<52
3720 opt%=opt%-48
3730 IF opt%=1 AND draw% = 1 THEN ST
AKE%=STAKE%-BET%:POT%=POT%+BET%:COLOU
R130:CLS:VDU26:PROCinfo:PROCwin:ENDPR
OC
3740 IF opt%=1 THEN STAKE%=STAKE%-BE
T%:POT%=POT%+BET%:PROCinfo:ENDPROC
3750 IF opt% = 3 THEN FOLDED% = 1:PO
T%=0:COLOUR 130:COLOUR 0:VDU26:CLS:EN
DPROC
3760 REPEAT
3770 CLS:INPUT"How much",raise1%
3780 UNTILraise1%>0 AND raise1%<=10
3790 BET1%=BET%+raise1%:STAKE%=STAKE
%-BET1%:POT%=POT%+BET1%
3800 IF BET1% <= (raise%+bet%) OR bl
uff% > 25 OR cheat%=1 OR POT%>20 PRIN
T"I see you":POT%=POT% + BET1%:BLOP%=
1
3810 IF BLOP%=1 AND draw% = FALSE TH
EN PROCinfo:0$=INKEY$(500):BLOP%=FALS
E:ENDPROC
3820 IF BLOP%=1 AND draw% = 1 THEN P
ROCinfo:0$=INKEY$(500):PROCwin:BLOP%=
FALSE:ENDPROC
3830 STAKE%=STAKE%+POT%:POT%=FALSE:F
OLD%=1
3840 PRINT"I fold"
3850 PROCinfo
3860 0$=INKEY$(500)
3870 ENDPROC
3880
3890 DEF PROCpre_bet
3900

```

```

3910 IF value < 13 AND draw% = FALSE
THEN FOLD% = 1:ENDPROC
3920 IF value < 300 AND draw% = 1 TH
EN FOLD% = 1:ENDPROC
3930 IF value < 500 THEN bet% = 1: r
aise% = 1:ENDPROC
3940 IF value < 1000 THEN bet% = 1:
raise% = 2:ENDPROC
3950 IF value < 1500 THEN bet% = 2: r
aise% = 2:ENDPROC
3960 IF value < 4000 THEN bet% = 3:
raise% = 3:ENDPROC
3970 IF value < 8000 THEN bet% = 3:
raise% = 4:ENDPROC
3980 IF value < 40000 THEN bet% = 4:
raise% = 5:ENDPROC
3990 IF value < 70000 THEN bet% = 4:
raise% = 6:ENDPROC
4000 bet% = 5:raise% = 7:ENDPROC
4010
4020 DEF PROCbluff
4030
4040 PROCvalue(Hand1$)
4050
4060 IF value < 500 THEN BLUFF%=BLUF
F%+1
4070 bluff% = BLUFF% * 100 DIV no%
4080 ENDPROC
4090
4100 DEF PROCwin
4110
4120 VDU 28,1,27,18,22
4130 COLOUR130:COLOUR0:CLS
4140 VDU26
4150
4160 PROCvalue(Hand1$)
4170 value1=value
4180
4190 PROCvalue(Hand2$)
4200 value2=value
4210
4220 PROCdeal(Hand2$,400)
4230
4240 IF value1>value2 THEN PRINT TAB
(0,21);"you win ";Opponent$:STAKE%=ST
AKE%+POT%:POT%=0:PROCinfo:0$=INKEY$(5
00):ENDPROC
4250 IF value1<value2 THEN PRINT TAB
(0,21);"you lose ";Opponent$:POT%=0:P
ROCinfo:0$=INKEY$(500):ENDPROC
4260 IF value1=value2 THEN PRINT TAB
(0,21);"both hands are of equal value
";Opponent$:STAKE%=STAKE%+(POT%/2):P
OT%=0:PROCinfo:0$=INKEY$(500):ENDPROC
4270
4280 DEF PROCdraw
4290 VDU 26
4300 VDU 28,1,27,18,22:COLOUR128:COL
OUR3
4310 CLS
4320 len%=LEN(ZZ$)
4330 IF len% = 10 THEN PRINT TAB(0,2

```

```

2);"I take no cards" ELSE IF (10-len%
)/2=1 THEN PRINT TAB(0,22);"I take 1
card" ELSE PRINT TAB(0,22);"I take ";
(10-len%)/2;" cards"
4340 IF len%=10 THEN 4410
4350 a%=(10-len%)/2+Count%
4360 REPEAT
4370 ZZ%=ZZ%+A$(Count%)
4380 Count%=Count%+1
4390 UNTIL Count% = a%
4400 Hand2%=ZZ%
4410 PRINT"Do you want to change
any cards"
4420 REPEAT:0$=BET$:UNTIL 0$="Y" OR
0$="N"OR 0$="y"OR 0$="n"
4430 IF 0$="N"OR 0$="n" THEN COLOUR1
30:COLOUR0:CLS:VDU26:ENDPROC
4440
4450 PROCflush2
4460
4470 FOR J%=1 TO 5
4480 PRINT"Change card ";J%;" ?";
4490 REPEAT:0$=BET$:UNTIL 0$="Y" OR
0$="N"OR 0$="n"OR 0$="y"
4500 PRINT0$
4510 IF 0$="Y"OR 0$="y" THEN Z%(J%)=
1:taken%=taken%+1
4520 NEXT
4530 len%=0:0$=""
4540 FOR J%=1 TO 5
4550 IF Z%(J%)=0 THEN 0$=0$+MID$(Han
d1$,2+J%-1,2)
4560 IF Z%(J%)=1 THEN len%=len%+1
4570 NEXT
4580 IF LEN(0$)>9 THEN 4640
4590 a%=len%+Count%
4600 REPEAT
4610 0$=0$+A$(Count%)
4620 Count%=Count%+1
4630 UNTIL a%=Count%
4640 Hand1%=0$
4650 PROCdeal(Hand1$,700)
4660 COLOUR130:COLOUR0:CLS:VDU26:END
PROC
4670
4680 DEF PROCcheat
4690
4700 FOLD%=0
4710
4720 PROCvalue(Hand1$)
4730 value1=value
4740
4750 PROCvalue(Hand2$)
4760 value2=value
4770
4780 IF value2>value1 THEN bet%=5:ra
ise%=7:ENDPROC
4790
4800 PROCalter
4810 PROCvalue(Hand2$)

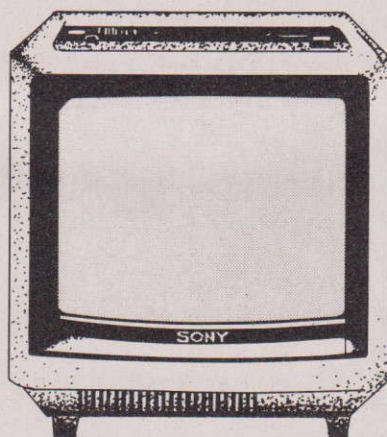
```



The best software store in the country  
(now boasting over 2,500 good titles in stock)  
with accessories, joysticks and books. We have  
somehow made room for hardware stocks.  
Home computers, disc drives, printers, monitors,  
and now a very special offer

## SONY<sup>®</sup> R.G.B. MONITOR\*

Ideal for BBC/Electron and many others.



- \* 14" colour monitors
- \* Resolution equivalent to 480 x 400 pixels
- \* Full 12 month guarantee

- \* These monitors using Trinitron tubes have been converted to RGB

Our  
unbeatable  
price

# £149.50

inc. VAT.

Or mail order £159.50 including VAT, p&p & BBC leads  
Postal and telephone orders welcomed.

All major credit cards accepted.  
(Offer subject to availability)

Soft Shop Ltd.  
55 St Peter's Court  
Chalfont St Peter  
Bucks. SL9 9QQ

Tel: (0753) 889010

Easy Road Access  
Free Parking and  
Open Mon. — Sat.  
9.00—5.30  
Sun. 9.30 - 12.30



## From Page 19

```

4820 IF value<10000 THEN PROCalter1
4830 ENDPROC
4840
4850 DEF PROCalter
4860
4870 abo%=abo%+1
4880 IF abo%=2 THEN ENDPROC
4890 S$=MID$(Hand2$,1,2)
4900 naughty%=16
4910 REPEAT
4920 F$=A$(naughty%)
4930 IF MID$(S$,1,1)=MID$(F$,1,1) TH
EN S$=S$+F$
4940 naughty%=naughty%+1
4950 UNTIL naughty%=53
4960 abbo%=LEN(S$)
4970 abbo$=Hand2$
4980 Hand2$=S$+MID$(abbo$,3,10-abbo%
)
4990 ZZ$=Hand2$
5000 bet%=5:raise%=7
5010
5020 ENDPROC
5030
5040 DEF PROCbet1
5050
5060 IF cheat% THEN PROCcheat ELSE I
F draw% THEN PROCpre_bet1 ELSE PROCpr
e_bet
5070
5080 VDU28,1,27,18,22
5090 COLOUR128:COLOUR3:CLS
5100 PRINT"Do you want to:"
5110 PRINT"1. Open"
5120 PRINT"2. Fold"
5130 REPEAT:opt%=BET:UNTIL opt%>48 A
ND opt%<51
5140 IF opt%=50 THEN FOLDED%=1:COLOUR
R130:COLOUR0:VDU26:CLS:POT%=0:ENDPROC
5150 REPEAT
5160 PRINTTAB(0,4);SPC(14)
5170 INPUT TAB(0,4)"How much",BET%
5180 UNTIL BET%>0 AND BET%<=10
5190 *FX15
5200 STAKE%=STAKE%-BET%:POT%=POT%+BE
TX%
5210 IF ((bet%*raise%)>=BET% OR bluf
f%>25 OR POT%>20 OR cheat%=1) AND FOL
D%=0 THEN CLS:PRINT"I see you":POT%=P
OT%+BET%:PROCinfo:BLOP%=1 ELSE FOLD%=
1
5220 IF FOLD%=1 AND cheat%=FALSE THE
N PRINT"I fold":STAKE%=STAKE%+POT%:PO
T%=0:PROCinfo:VDU26:0$=INKEY$(500):EN
DPROC
5230 g=(raise%*bet%-BET%)/2
5240 IF g>10 THEN g=10
5250 IF g>1 THEN g=INT(g):COLOUR128:
COLOUR3:PRINTTAB(1,23)"and raise you
";g:POT%=POT%+g:BET%=g:PROCinfo:PROC

```



```

2nd_bit
5260 IF draw% AND g>=1 AND FOLDED%=0
PROCwin:VDU26
5270 IF FOLDED%=0 0$=INKEY$(500)
5280 ENDPROC
5290
5300 DEF PROCpre_bet1
5310
5320 IF value<13 AND draw%=0 THEN FO
LD%=1:ENDPROC
5330 IF value<300 AND draw%=1 THEN F
OLD%=1:ENDPROC
5340 IF value<500 THEN bet%=1:raise%
=1:ENDPROC
5350 IF value<1000 THEN bet%=1:raise
%=2:ENDPROC
5360 IF value<1200 THEN bet%=2:raise
%=2:ENDPROC
5370 IF value<1500 AND taken%>2 THEN
bet%=4:raise%=2:ENDPROC ELSE IF valu
e<1500 AND taken%<3 THEN bet%=2:raise
%=2:ENDPROC
5380 IF value<4000 AND taken%>2 THEN
bet%=5:raise%=3:ENDPROC ELSE IF valu
e<4000 AND taken%<3 THEN bet%=3:raise
%=3:ENDPROC
5390 IF value<8000 AND taken%>2 THEN
bet%=5:raise%=5:ENDPROC ELSE IF valu
e<8000 AND taken%<3 THEN bet%=2:rais
e%=3:ENDPROC
5400 IF value<40000 AND taken%>2 THE
N bet%=5:raise%=5:ENDPROC ELSE IF val
ue<40000 AND taken%<3 THEN bet%=4:ra
ise%=4:ENDPROC

```

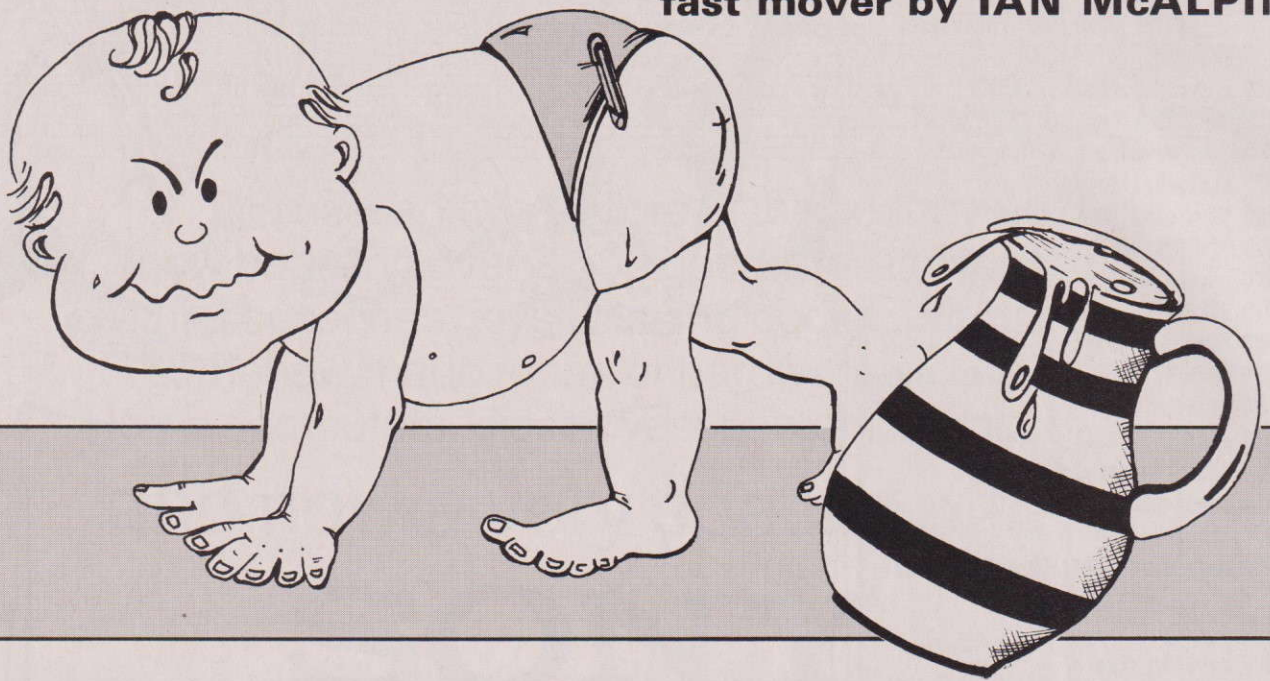
```

5410 IF value<70000 AND taken%>2 THE
N bet%=5:raise%=5:ENDPROC ELSE IF val
ue<70000 AND taken%<3 THEN bet%=4:ra
ise%=4:ENDPROC
5420 bet%=5:raise%=7:ENDPROC
5430
5440 DEF PROCalter1
5450
5460 stiff%=stiff%+1
5470 IF stiff%=2 THEN ENDPROC
5480 naughty%=16
5490 F$=MID$(Hand2$,1,2)
5500 REPEAT
5510 naughty%=naughty%+1
5520 0$=A$(naughty%)
5530 IF MID$(F$,2,1)=MID$(0$,2,1) TH
EN F$=F$+0$
5540 UNTIL naughty%=52 OR LEN F$=10
5550 IF LEN F$<10 THEN Hand2$=F+RIG
HT$(F$,10-LEN(F$)):ENDPROC
5560 Hand2$=F$
5570 ZZ$=Hand2$
5580 ENDPROC
5590
5600 REM these DATA statements shoul
d be filled with names of
people you want the computer
to cheat against.
5610
5620 DATA ****
5630
5640 REM Escape Key has been Pressed
5650 MODE7
5660 END

```



Catch crashing crockery in this  
fast mover by IAN McALPINE



# BABY, YOU'RE QUITE A HANDFUL!

**YOUR** rich aunt and uncle have asked you to babysit with their two-year-old son, Tobermory, who is quite a mischievous chap for his age.

No sooner have your aunt and uncle left than he jumps up on to the china cabinet and runs along the top of it, jumping up and down as he goes.

The expensive china starts to fall and you run frantically left and right trying to catch the falling pieces.

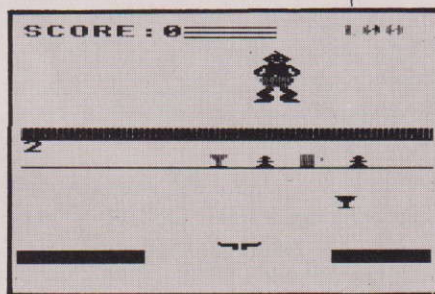
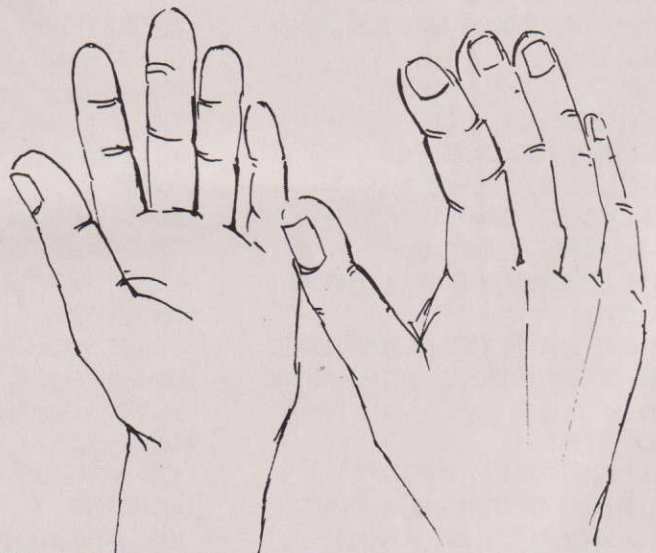
It's easy to start with, but soon the crockery falls thick and fast.

Because of the Law of Frustrating Games the china will occasionally turn into a red anti-matter blob, heralded by a ping.

Never try to catch an anti-matter blob or your hands will explode and you will lose a life – as you do when you drop a piece of china.

You have three lives to begin with, but get an extra one on completion of Level 3. The number of lives left is printed in yellow two-thirds up the left hand side of the screen. This number turns red when you only have one life left.

Once you have lost all of your lives, your aunt and uncle return home, see their beautiful china smashed, and uncle promptly shoots you!



The program's features include ability to turn the sound on/off, use of keyboard or joystick for movement, pause game control, SAVE and LOAD high score from cassette, "gruesome consequences" and full use of the Acorn Speech System if fitted.

The Speech System is used for prompting input, speaking level number, the "thud" in the gruesome consequences and for announcing the score.

Enter the program exactly as listed, as it runs very tight on memory.

When entering the program disc users would be advised to SAVE it each time before they run it because of the relocater at the start of the program.

Cassette users can leave out lines 60, 1330 and 1340. All users should initially enter line 70 as:

70 PROCi

Once the program is running correctly the rest of line 70 can be entered.



## CONTROLS

- To move your hands left or right use the left or right cursor keys.
- To accelerate hold down either Shift key.
- To pause the game press Shift Lock.
- To continue press Caps Lock.
- To turn the sound on/off press S/Q.
- The sound controls only take effect at the end of each drop of china.
- To return to the title page press Escape.

## PROCEDURES

**PROCi**

Initialises characters and envelopes.

**PROCgh**

Asks whether to load recorded high score.

**PROChp**  
**PROCa**

Prints title page.  
Presses space or fire button.

**PROCi1**

Initialises strings and secondary variables.

**PROCre**  
**PROCw**

Prepares to start game.  
Checks whether using keys or joystick.

**PROCb**  
**PROcf**

Prints bouncing baby.  
Drops a piece of china.

**PROCwt(num)**  
**PROCu**

Delay.  
Updates score, high score and pieces of china left to drop.

**PROcff**  
**PROCc**

Plays title tune.  
Gruesome consequences.

**PROChp**

Prints your score and high score at end of game.

**PROCs**  
**PROCKm**

Prints screen display.  
Checks which keys pressed.

**PROChp**

Checks for joystick movement.

**PROChp**

Pauses game or sound on/off?

**PROChp**

You caught the piece of china.

**PROChp**

You dropped the piece of china.

**PROChp**

Erases piece of china from bottom of screen.

**PROChp**  
**PROCr**

Records high score.  
Restores everything to default.

**PROCas**

Assembles machine code to print the baby and your hands.

**PROCi2**

Initialises which piece of china goes where on the cabinet.

**PROCspr**

Reads data, then speak.

**PROCd(num)** Prints "China Drop", in teletext graphics.

**PROCdo(num)** Plot door.

**FNrc(num)** Reads character.

```

10 REM ...China Drop...
20 REM by Ian McAlpine.
30 REM Theme, Graphics and Speech
40 REM consultant, Keith McAlpine.
50 REM (C) The Micro User
60 IFPAGE<>E00THEN1330
70 PROCi:ONERRORMODE7:CLEAR:VDU23;
10,32,0;0;0;:PROChp:PROCa:GOTO90
80 MODE7:VDU23;10,32,0;0;0;:PROChp
:CLS:PROChp:PROCa
90 PROCi1:PROCas:MODE2:PROCre:SOUN
D3,5,50,-1:REPEAT:PROCw:PROCb:PROCi:I
Fd=8ANDcr<>0sk=sk+1:ll=ll+1:IFsk=5cr=
cr+1:PROCwt(50):SOUND1,4,200,5:PROCu:
PROCwt(150):PROcff:PROCwt(150)
100 IFsk=5sk=5
110 IFd=8ANDcr<>0MODE2:PROCre:SOUN
D3,5,50,-1
120 UNTILcr=0:*FX15,0
130 CLS:PROCc:MODE7:VDU23;10,32,0;0
;0;:PROChp:END
140 DEFPROCi:RESTORE1360:FORC%=223T
O254:VDU23,c%:FORd%=0TO7:READd:VDUd:M
EXT,:ENVELOPE1,4,-4,-1,-1,20,20,20,1,
0,0,0,1,1:ENVELOPE2,133,8,4,8,3,1,1,1
26,0,0,-10,126,0:ENVELOPE3,136,-1,-1,
-1,30,2,2,127,0,-127,-10,60,60
150 ENVELOPE4,1,0,0,0,0,0,126,-1,
0,-3,126,126:ENVELOPE5,1,40,-4,40,6,1
2,6,50,0,0,-50,75,5:H%=100:*FX4,1
160 ENDPROC
170 DEFPROCi1:DIMO$(9),CZ9:*FX11,1
180 *FX12,100
190 sc=0:cr=3:sk=2:ll=1:pp=3:RESTOR
E1410:FORd%=&0A40TO&0A84:READd:?D%&D:
NEXT:ENDPROC
200 DEFPROCi2:FORch=1TO8:O$(ch)=CHR
$(244+RND(4)):CZ?ch=RND(7):NEXT:O$(9)
=CHR$244:CZ?9=1:xh=8:yh=30:xb=8:yb=2:
wa=2:d=0:y=FALSE:rep=FALSE:ENDPROC
210 DEFPROCa:PROcff:PROCwt(150):VDU
31,0,23,129,157,135,136:PRINT"PRESS <
RETURN> OR 'FIRE' BUTTON ..";RESTORE
1420:PROCspr:TIME=0:REPEAT:k=INKEY(-7
4):j=ADVAL(0):UNTILk=TRUE ORj=10TIME
>=1500:IFk=TRUE m$="K"ELSEm$="J"
220 IFTIME>=1500PROCa
230 ENDPROC
240 DEFPROCs:VDU19,7,0;0;19,1,0;0;1
9,3,0;0;:GCOLOR,7:MOVE0,736:DRAW1279,7
36:MOVE0,670:DRAW1279,670:MOVE0,30:DR
AW1279,30:MOVE400,992:DRAW700,992:MOV
E400,1007:DRAW700,1007:MOVE400,1019:D
RAW700,1019
250 BCOL44,1:FORyc=740TO764STEP4:PL
OT77,640,yc:NEXT:BCOL43,3:FORyc=0TO26
STEP4:PLOT77,640,yc:NEXT:PRINTTAB(6,1
5)SPC(7):VDU20:FORpr=1TO8:COLOURC%pr
:VDU31,pr*2,10:PRINTO$(pr):NEXT:*FX9,
5

```



## From Page 23

```

260 *FX10,5
270 COLOUR14:PRINTTAB(0,0);"SCORE:"
:COLOUR12:PRINTTAB(11,0);"HIGH:";COLO
UR4:PRINTTAB(6,31);"LEVEL ";:COLOUR2
:PRINTTAB(6,0);sc:COLOUR5:PRINTTAB(16
,0);HZ:COLOURpp:PRINTTAB(0,9);cr:COLO
UR4:PRINTTAB(13,31);11;
280 DEFPROCw:IFm$="K"PROCKm ELSEPRO
Cjm
290 ENDPROC
300 DEFPROCKm:IFINKEY(-26)AND(xh>0)
xh=xh-1
310 IFINKEY(-122)AND(xh<15)xh=xh+1
320 ?&0A43=xh:CALL&A05:IFrep=TRUE r
ep=FALSE:ENDPROC
330 IFINKEY(-1)rep=TRUE:PROCKm
340 ENDPROC
350 DEFPROCjm:ov=32750:nv=ADVAL(1):
IFnv<(ov-30000)AND(xh<15)xh=xh+1ELSEI
Fnv>(ov+30000)AND(xh>0)xh=xh-1
360 ?&0A43=xh:CALL&A05:IFrep=TRUE r
ep=FALSE:ENDPROC
370 fi=ADVAL(0)AND3:IFI=1rep=TRUE:
PROCjm
380 ENDPROC
390 DEFPROCspk:REPEAT:READs:SOUND-1
,s,0,0:UNTILs=1:ENDPROC
400 DEFPROCb:PROCpa:IFyb=2yb=1ELSEy
b=2
410 ONwa 60T0430,420
420 IF(xb<15)xb=xb+1:60T0450
430 IF(xb>0)xb=xb-1
440 IFxb=0wa=2
450 IFxb=15wa=1
460 ?&0A4D=xb:??&0A4E=yb:CALL&A20:EN
DPROC
470 DEFPROCwt(de):TIME=0:REPEATUNTIL
TIME>de:ENDPROC
480 DEFPROCf:f_ch=RND(5):IFF_ch<3EN
DPROC
490 o=RND(8):PROCw:PROCb:IF0$(o)=""
THEN490
500 of$=CHR$237+CHR$8+STRING$(sk,CH
R$10)+0$(o):0$(o)="" :xa=o*2:b=FALSE:IF
Fsk=3bo=27ELSEbo=28
510 FORdp=10T0bo STEPsk:IFb=FALSE:IF
FRND(50)<3ANDdp<=23of$=CHR$237+CHR$8+
STRING$(sk,CHR$10)+0$(9):0$(o)="" :o=9
:b=TRUE:SOUND2,-15,200,1
520 PROCw:PROCb:COLOURC?o:PRINTTAB
(xo,dp);of$:PROCb:PROCw:NEXT:IFo<>9AN
D FNrc(xo)=1300Ro<>9AND FNrc(xo)=129P
ROCct ELSEIFo<>9AND FNrc(xo)<>1300Ro<
>9AND FNrc(xo)<>129PROCdp
530 IFo=9AND FNrc(xo)=1300Ro=9AND F
Nrc(xo)=129y=TRUE:CZ?o=7:PROCdp:CZ?o=
1 ELSEIFo=9AND FNrc(xo)<>1300Ro=9AND
FNrc(xo)<>129PROCe:PROCct
540 IFxb<16COLOUR7:VDU31,xb+1,yb,23
8,239,240:PROCwt(60):VDUB,8,8,237,237

```

```

,237
550 d=d+1:IFd<>8SOUND3,5,50,-1:ENDP
ROC
560 DEFPROCu:COLOUR2:PRINTTAB(6,0);
sc:COLOUR5:PRINTTAB(16,0);HZ:IFcr=<=1
pp=1ELSEpp=3
570 COLOURpp:PRINTTAB(0,9);cr:ENDPR
OC
580 DEFFNrc(x):LOCALAZ,cZ:VDU31,x,y
h:AZ=135:=(USR(&FFF4)AND&FFFF)DIV&100
590 DEFPROCct:*FX15,0
600 SOUND2,2,100,10:IFsk=3PRINTTAB(
xo,28);" "
610 IFo<>9sc=sc+sk*10ELSEsc=sc+sk*5
620 IFsc>HZHZ=sc
630 PROCu:ENDPROC
640 DEFPROCdp:*FX15,0
650 PROCe:cr=cr-1:PROCu:ye=0:VDU19,
0,7,0;:PROCwt(5):SOUND1,1,100,1:SOUND
0,-15,7,30:VDU19,0,0,0;:VDU29,(xo*64)
+32,32;
660 FORex=1T0350STEP10:MOVE0,0:6COL
0,CZ?o:PLOT65,0,ye:MOVE0,0:PLOT65,-ex
,32:MOVE0,0:PLOT65,-ex,ye:MOVE0,0:PLD
T65,ex,32:MOVE0,0:PLOT65,ex,ye:MOVE0,
0:6COL0,0
670 PLOT65,-ex,ye:MOVE0,0:PLOT65,ex
,ye:MOVE0,0:PLOT65,0,ye:MOVE0,0:PLOT6
5,-ex,32:MOVE0,0:PLOT65,ex,32:ye=ye+5
:PLOT65,ex,ye:MOVE0,0:NEXT:ENDPROC
680 DEFPROCp:*FX12,0
690 *FX15,0
700 IFsc=HZ:VDU31,8,2,129,141:PRINT
"Congratulations!!";TAB(8,3);:VDU130,
141:PRINT"Congratulations!!":SOUND-1,
279,0,0:SOUND-1,202,0,0:PROCwt(150)
710 PRINTTAB(8,6);:VDU131,141:PRINT
"Your Score "CHR$129;sc:TAB(8,7);:VDU
131,141:PRINT"Your Score "CHR$129;sc;
TAB(8,10);:VDU134,141:PRINT"High Scor
e "CHR$136;HZ:TAB(8,11);:VDU134,141:P
RINT"High Score "CHR$136;HZ
720 RESTORE1440:PROCspk:sc$=STR$sc:
IFLENsc$<4:REPEATsc$="0"+sc$:UNTILLE
Nsc$=4:IFsc$="0000"SOUND-1,139,0,0:6D
T0760
730 SOUND-1,VAL(LEFT$(sc$,1))+48,0,
0:SOUND-1,141,0,0:SOUND-1,VAL(MID$(sc
$,2,1))+48,0,0:SOUND-1,140,0,0:SOUND-
1,165,0,0:IFMID$(sc$,3,1)="1"ANDMID$(
sc$,4,1)="0"SOUND-1,264,0,0:6T0760
740 IFMID$(sc$,3,1)="1"SOUND-1,VAL(
MID$(sc$,4,1))+32,0,0:SOUND-1,135,0,0
:6T0760ELSEIFMID$(sc$,3,1)="0"SOUND-
1,VAL(MID$(sc$,4,1))+48,0,0ELSE SOUND
-1,VAL(MID$(sc$,3,1))+32,0,0:SOUND-1,
137,0,0
750 IFMID$(sc$,4,1)<>"0"SOUND-1,VAL(
MID$(sc$,4,1))+48,0,0:*FX21,0
760 CLEAR:PROCwt(500):VDU31,2,17,14
1,134:PRINT"Do you want to play again
?";TAB(2,18);:VDU141,132:PRINT"Do yo

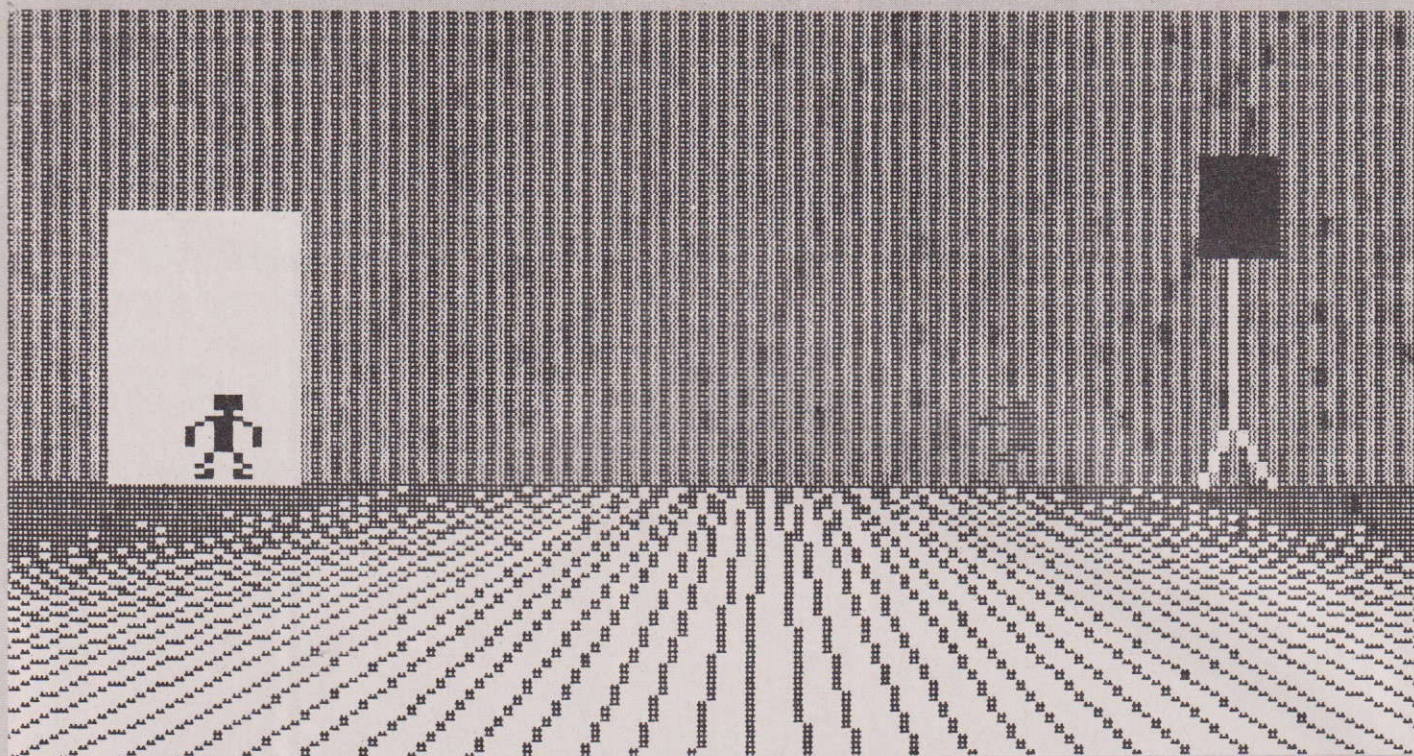
```

```

u want to play again ?":RESTORE1450:P
ROCspk:A=8ET:*FX21,8
770 IFA=780R A=110PROCrh:*FX21,8
780 IFA=780R A=110PRINTTAB(0,22)SPC
8"Good_bye one and all!"SPC11:RESTORE
1460:PROCspk:PROCwt(400):PROCcr:END
790 IFA=890R A=121:PROCa:60T090ELSE
760
800 ENDPROC
810 DEFPROCre:*FX21,0
820 VDU23,10,32,0,0,0;:*FX9,25
830 *FX10,25
840 *FX21,8
850 IF1<10SOUND-1,229,0,0:SOUND-1,
11+48,0,0
860 COLOUR12:PRINTTAB(6,15);"READY!
!":PROCwt(150):PROCi2:PROCs:*FX15,0
870 ENDPROC
880 DEFPROCe:*FX21,0
890 IFy=TRUE PRINTTAB(xo,28);" " :y=
FALSE:ENDPROC
900 PRINTTAB(xo,yh);" ";TAB(xo,28);
" ":ENDPROC
910 DEFPROCpa:*FX15,1
920 *FX12,0
930 IFINKEY(-81)REPEATUNTILINKEY(-6
5)
940 IFINKEY(-17):*FX210,1
950 IFINKEY(-82):*FX210,0
960 *FX12,100
970 ENDPROC
980 DEFPROCff:RESTORE1350:*FX21,0
990 REPEAT:READV,PP,D:IFV=1V=-15
1000 SOUND1,V,PP,D:SOUND2,V,PP-48,D:
SOUND3,V,PP+48,D:UNTILD=99:*FX15,1
1010 ENDPROC
1020 DEFPROCbp:LOCALi:FORi=0T07:VDU3
1,0,i,154,RND(7)+144:NEXT:PROCcd(0):F
ORI=7T014:VDU31,0,i,136,133,RND(7)+14
4:NEXT:PROCcd(7):FORi=14T020:VDU31,0,
i,154,RND(7)+144:NEXT:PROCcd(14):ENDP
ROC
1030 DEFPROCcd(ee):*FX21,8
1040 PRINTTAB(4,ee)"x/tj5"SPC12CHR$2
55"/m0"TAB(4,ee+1)CHR$255"/j5_0"SP
C8CHR$255" j5"TAB(4,ee+2)CHR$255" j5
p b10p _p0 "CHR$255" "CHR$34")_0_p0
_p0"TAB(4,ee+3)CHR$255" j";
1050 PRINTCHR$255CHR$163"j5"CHR$255
CHR$163"((ss"CHR$255" "CHR$255" ?j7"
CHR$163CHR$255CHR$163CHR$255"j7"CHR$1
63CHR$255TAB(4,ee+4)CHR$255" _z5 "CHR
$255"j5"CHR$255" "CHR$255"j7"CHR$163C
HR$255" "CHR$255" j5j5 "CHR$255" "CHR
$255"j)?
1060 PRINTTAB(4,ee+5)+1'k5 "CHR$255
":u"CHR$255" ";CHR$255":)"CHR$255" "
CHR$255"!":j5 o1?j5"TAB(31,ee+6)"j5":
ENDPROC
1070 DEFPROCc:VDU29,640,0;:VDU24,-64
0,500,639,1023;:6COL33,134:CL6:VDU24,
-640,0,639,1023;:6COL0,6:MOVE-640,500

```





```
:DRAW640,500:FORBZ=-640TO640STEP12:MD
VEBZ,500:DRAWB*6Z,0:NEXT:VDU26:MOVE10
00,500:BCOLO,0:DRAW1030,550:DRAW1060,
500:MOVE1030,550
```

```
1080 DRAW1030,675:MOVE1000,675:MOVE1
060,675:BCOLO,3:PL0T85,1000,745:PL0T8
5,1060,745:MOVE800,564:VDU19,10,0;:
BCOLO,10:VDU5:VDU242,10,8,243:PROCdo(
5):PROCwt(100):PROCdo(0):MOVE165,564:
BCOLO,7:VDU224,10,8,227:PROCwt(50)
```

```
1090 PROCdo(5):MOVE300,564:BCOLO,0:V
DU228,10,8,241:MOVE370,600:BCOLO,0:VD
U251,252:SOUND16,4,3+RND(3),2:PROCwt(
50):MOVE370,600:BCOL33,6:VDU251,252:F
ORxs=330TO800STEP30:MOVExs,560:BCOLO,
0:VDU255,8:PROCwt(5):BCOL33,6:VDU255:
NEXT:PRINT">"
```

```
1100 PROCwt(10):MOVE840,560:BCOL33,6
:PRINT">":MOVE760,600:BCOLO,0:VDU253,
254:VDU19,10,9;0:SOUND1,3,200,20:PRO
Cwt(50):MOVE760,600:BCOL33,6:VDU253,2
54:MOVE800,564:BCOLO,0:VDU242,10,8,24
3:BCOL33,6:MOVE800,564:VDU242,10,8,24
3:BCOLO,1
```

```
1110 SOUND-1,7,0,0:MOVE800,532:VDU24
9,250:PROCwt(200):ENDPROC
```

```
1120 DEFPROCdo(U):BCOLO,U:MOVE100,50
0:MOVE250,500:PL0T85,250,700:MOVE100,
700:PL0T85,100,500:IFU=58BCOLO,7:PL0T6
9,220,610
```

```
1130 ENDPROC
```

```
1140 DEFPROCgh:RESTORE1430:PROCspk
```

```
1150 PRINTTAB(1,4);CHR$134"Do you wa
nt to load the High score?"A=INSTR(
"YNyn",GET$):IFA=20R A=4ENDPROC
```

```
1160 IFA=10R A=3VDU28,0,10,39,6:H=OP
ENIN"HIGH":INPUT#H,HX:CLOSE#H:VDU26:E
```

```
NDPROC ELSE1150
```

```
1170 DEFPROCrh:RESTORE1470:PROCspk
```

```
1180 LOCALA:PRINTTAB(1,22);CHR$134"D
o you want to record the High score ?
```

":A=INSTR("YNyn",GET\$):IFA=20R A=4END
PROC

```
1190 IFA=10R A=3VDU28,0,24,19,23:H=0
PENOUT"HIGH":PRINT#H,HX:CLOSE#H:CLS:V
DU26:ENDPROC ELSE1180
```

```
1200 DEFPROCas:PX=&A05
```

```
1210 [OPT0:LDA#0:LDX#0:LDA&0A40,X
```

```
1220 .write JSR&FFEE:INX:LDA&0A40,X:
```

```
CMP#&FF:BNEwrite
```

```
1230 RTS:]
```

```
1240 PX=&A20
```

```
1250 [OPT0:LDA#0:LDX#0:LDA&0A4A,X
```

```
1260 .write2 JSR&FFEE:INX:LDA&0A4A,X
```

```
:CMP#&FF:BNEwrite2
```

```
1270 RTS:]ENDPROC
```

```
1280 DEFPROCcr:#FX4,0
```

```
1290 #FX12,0
```

```
1300 #FX15,0
```

```
1310 #FX210,0
```

```
1320 ENDPROC
```

```
1330 PRINT"Please wait...":TAPE
```

```
1340 FORIX=0TO TOP-PAGE STEP4:IX!&EO
0=IX!PAGE:NEXT:??&13=??&13-(PAGE-&EO0)D
IV256:PAGE=&EO0:RUN
```

```
1350 DATA1,53,4,0,0,.25,1,53,4,1,73,
4,0,0,.25,1,73,4,1,81,4,0,0,.25,1,81,
4,1,89,9,1,73,5,0,0,.25,1,53,3,5,0,0,
.25,1,53,3,5,1,73,4,0,0,.25,1,73,4,1,
89,2,5,1,73,2,5,1,89,2,5,1,73,15,0,0,
99
```

```
1360 DATA0,0,0,24,24,0,0,0,28,28,28,
8,60,90,153,153,0,0,0,64,127,63,3,3,0
,0,0,1,127,126,96,96,153,153,24,36,36
```

```
,66,36,102,112,120,112,39,124,112,112
,112,1,0,3,7,13,31,13,6,0,128,192,224
,176,248,176,96,7,31,63,111,239,207,2
23,126,224,248,252
```

```
1370 DATA246,247,243,251,254,31,63,6
3,63,31,31,15,14,248,252,252,252,248,
248,240,112,30,60,120,124,62,30,126,2
52,120,60,30,62,124,120,126,63,0,0,0,
0,0,0,0,0,0,0,0,251,170,171,170,0,0,0
,0,190,170,170,170,0,0,0,0,228,164,22
8,160,4,112,112,80
```

```
1380 DATA72,72,72,72,108,7,15,7,34,6
3,7,7,7,7,7,9,17,17,17,51,28,60,254
,255,126,63,125,56,255,126,126,60,24,
24,24,126,248,250,253,249,249,250,252
,248,250,253,249,114,124,112,32,248,0
,24,60,24,126,24,60,126,0,0,0,0,0,129
,135,255,0,0,0,0,2
```

```
1390 DATA231,247,255,238,170,170,206
,170,170,234,0,238,170,168,171,170,17
0,174,0,174,170,168,171,170,170,238,0
,165,165,165,229,165,160,165,0
```

```
1400 DATA241,246,252,1
```

```
1410 DATA17,7,31,8,30,32,225,226,32,
255,17,7,31,8,2,237,237,237,237,8,8,8
,8,10,237,229,230,237,8,8,8,8,10,237,
231,232,237,8,8,8,8,10,17,1,237,233,2
34,237,8,8,8,8,10,17,7,237,235,236,23
7,8,8,8,8,10,237,237,237,237,255
```

```
1420 DATA241,246,252,237,267,174,1
```

```
1430 DATA184,275,281,267,257,200,176
,1
```

```
1440 DATA290,257,209,1
```

```
1450 DATA162,289,237,226,1
```

```
1460 DATA265,275,147,254,270,247,1
```

```
1470 DATA184,275,281,267,257,235,143
,176,1
```



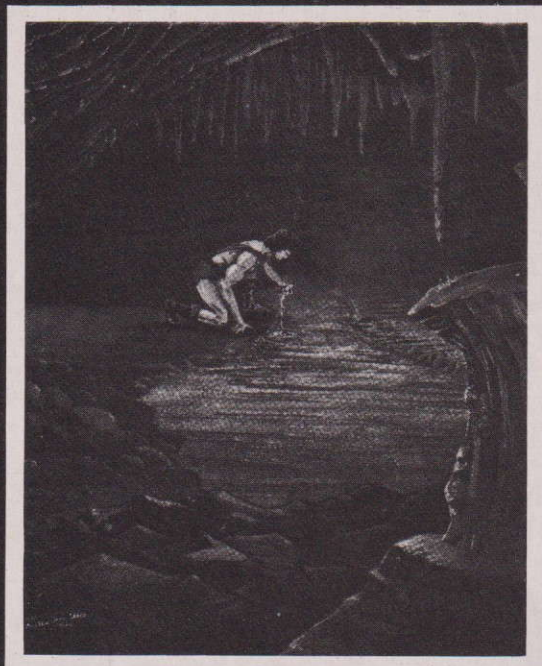


# THE NUMBER ONE ADVENTURE SERIES



Triffid Software Research

Triffid Software Research



**BBC-B**  
versions  
£6-95  
each



**BBC-B** RUNEMAGIC SERIES  
ADVENTURE ONE  
THE SECRET RIVER

**BBC-B** RUNEMAGIC SERIES  
ADVENTURE TWO  
THE WIZARD'S CITADEL

1. Load Character Generator (on side one of Adventure One).
2. Dice Character; buy weapon and armour and choose spells.
3. Load Character into Adventure One.
4. If you complete Adventure, then your character will progress in experience, so:
5. Load Character into Generator, rest up, relearn spells and buy better armour (only if you have found enough money on your travels); Save Character.
6. Buy next Adventure in series and load Character in.
7. Go to step 4. Until You have finished Adventure Eight then put your feet up for a well deserved rest.

Available by M/Order (p+p free)  
Send cheques or Postal Orders to:

Triffid Software Research  
12 Cullwood Lane  
Ashley  
New Milton  
Hampshire  
BH25 5QJ



**SPECTRUM 48K,  
ELECTRON and  
COMMODORE 64**  
versions coming soon

Available from good computer  
software stores nationwide

Send your software to us.  
If accepted,  
generous royalties offered.



**AVAILABLE NOW!**

Super colour graphics with six skill levels & 48 screen locations

# 1 WARP

**NEW!  
FOR BBC  
& ELECTRON**

COMING NOV 1st  
BBC/ELECTRON/48K SPECTRUM



arcade fun as Santa delivers the presents on Xmas eve... overcome hostile snowmen, flying snowballs, ice & icicles etc. whilst consuming as much cake & sherry as possible on the way. Hi-res full colour graphics £7.95



BBC OR ELECTRON

SPACE STATION ALPHA

Zorakk the Conqueror... a strategical fantasy adventure with full colour graphics. £7.95

Space station Alpha... A full colour space battle with 3D graphics £7.95



Space...the final frontier...

as commander of a federation starship, your mission is to seek out a fellow captain who is lost in space. You must explore strange new galaxies, fight and destroy the enemy and boldly go where no man has gone before.

BBC £7.95 ELECTRON £7.95



**SOFTWARE**

65 High Street, Gosforth, Tyne & Wear Tel: (091) 2845965

Available from most major stockists, all good computer shops or direct from us. Dealers contact Centresoft, Express.

Access/Barclaycard or send cheque/P.O.

**AVAILABLE NOW**

**ORDER BY POST**





# Flee the gardener ...catch the fleas

Help a senior spider  
round the garden in  
**ALAN SERGEANT's**  
lunchtime lark-about

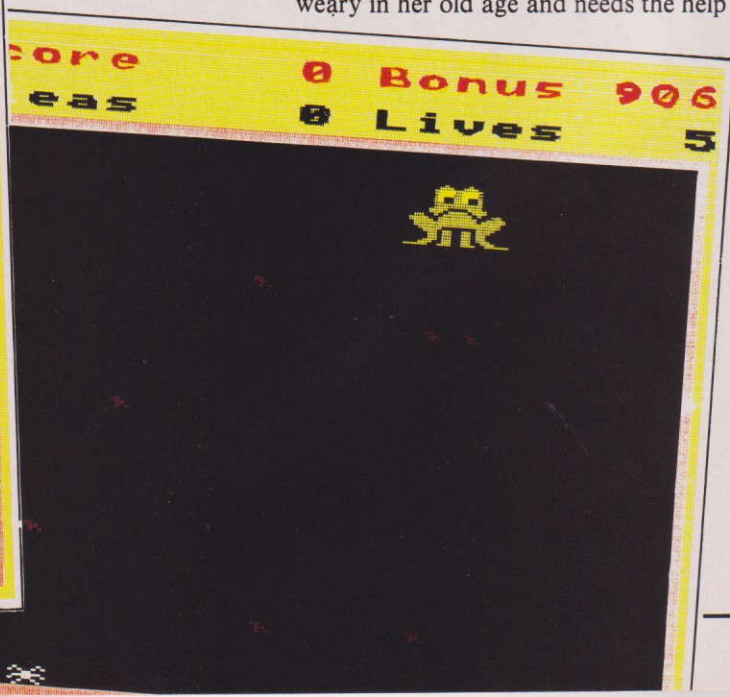
ME mam always told me that there were fairies at the bottom of our garden — well, I checked the other day and believe me it's a load of old rubbish. The place is full of insects and other horrible creepy crawlies.

Down in the bottom corner, just to the left of the garden roller, there's the most gi-normous white lady spider I've ever seen in my life. I can tell she's a lady by the way she keeps smashing everything in sight with her handbag.

Alas and alack, she is now a widow and her daily diet of fresh fleas is getting more difficult to come by, what with inflation as it is, and VAT (vermin added tax) escalating out of all proportion.

To cap it all the gardener, in his wisdom, has sprayed everything with this horrible yellow stuff, fatal to spiders I'm told.

Anyway the lady is getting a bit weary in her old age and needs the help





of your nimble fingers to get her around the garden.

You move Arachnida — she didn't pick the name — round the screen using:

A (up) . (left)  
Z (down) , (right)

You must catch a minimum of 20 fleas before you move on to another screen.

There are seven different screens in each round and on the first run through there are a maximum of eight fleas on the screen. This number decreases by one each round to make the game a little more difficult.

You score 10 points for each flea you catch, plus whatever bonus points are left if you clear a screen.

Just be careful of anything yellow — the walls, the frog, the plant pots, and even the gardener's boots are poisonous.

One touch means certain death, although you do have five lives to play with. There is a time limit on each screen so don't hang about 'cos the fleas won't.



*THE following routines are a little more complicated and require further explanation:*

**PROCtun(P%,D%,DD%)** — Writes the music that is passed to PROCmusic. The parameters are passed because without this, each time the PROC is called, the pitch resets.

It is this procedure which keeps the music playing throughout the game by using the ADVAL(-8) test. This checks for free spaces in the channel three sound buffer and, if there are any, continues playing the tune.

**PROCscroll** — Inserts a space in the bottom right hand corner of the screen forcing it to scroll upwards. If you prefer not to have the scroll, replace PROCscroll in lines 4190, 4260, 4380 and 4490 with CLS and omit lines 4620-4720.

**PROCenv(e%)** — This is necessary to cater for the 0.1 OS which only allows the user to define four envelopes. This game needs five and gets round this by defining two envelopes with the same number, (1), and toggling between them according to the value of e%.

**PROCscoretbl** — In order to control the input to the high score table, I have used an OSWORD call set at &FFF1 (line 700), with A% set to 0.

This routine takes a specified number of characters from the currently selected

## PROCEDURES

Most of the procedures are adequately described by their names, but the following require a little more description.

**PROCinit\_arrays**

Initialises all arrays which are called but only once.

**PROCstartup**

Initialises screen and restores data for music for each round.

**PROCinit**

Initialises variables and envelopes.

**PROCsetup**

Initialises certain variables and strings.

**PROCsoundoff**

Empties the sound queue at "Game over".

**PROCtun (P%,D%,DD%)**

Creates the music and passes parameters to PROCmusic, P% — offset, D% — note length, DD% — pause length.

**PROCmusic**

Called throughout the game with updated parameters.

**PROCdbl (msg\$,X%,Y%,C%)**

Double height characters passing the string, X, Y coordinates and colour.

**PROCsgl (msg\$,Y%,C%)**

Single height centralised characters passing the string, Y coordinates and colour.

**PROCenv(e%)**

Toggles between two envelopes with the same number.

**FNp (X%,Y%,C%)**

Hit check for colour C% at location X%, Y%.

## VARIABLES

**BO%**

Bonus.

**BPITCH%**

Base pitch for music.

**co%**

Count for number of lines scrolled up.

**dead%**

Number of fleas caught.

**F%**

Random number for feet to move.

**frog%**

Count for delay in froghop.

**FOOTflag%**

Flag for deciding which foot to move on screen 1.

**HI%**

Hiscore.

**lives%**

Lives.

**MAX%**

Maximum number of fleas on screen.

**NEWRIGHTX%, NEWLEFTX%**

**OLDRIGHTX%, OLDLEFTX%**

**NEWRIGHTY%, NEWLEFTY%**

**OLDRIGHTY%, OLDLEFTY%**

Coordinates of feet.

**NEWX%, NEWY%**

**OLDX%, OLDY%**

Coordinates of spider.

**NFX%, NFY%**

**OFX%, OFY%**

**P%, D%, DD%**

Coordinates of frog.

Offset from BPITCH%, note length, pause length.

**snuffedit%**

True or false for dead spider.

**SCREEN%**

Screen number.

**XX%,YY%**

Coordinates of teletext spider.

input stream. Input is terminated with a Return, the last character can be deleted and Ctrl+U deletes the entire line.

If the maximum number of characters is exceeded a VDU7 prompts. X% and Y% are the low and high bytes of the address of the parameter block. The parameters are set up using the following:

**block%?0** and **block%?1** are the low and high bytes of the start address of the

input buffer.

**block%?2** is the maximum length of the input stream.

**block%?3** and **block%?4** are the minimum and maximum acceptable Ascii values of the input.

**\*FX12,0** resets the auto repeat delay and **\*FX15,1** flushes the keyboard buffer. OSWORD is called and finally the string in **buff%** is allocated to **names\$(N)** using the string indirection operator \$.



## From Page 29

```

5 REM (C) The Micro User
10 PROCinit_arrays
20 MODE7
30 VDU23;8202;0;0;0;
40 PROCinst
50 *FX12,2
60 REM*****
70 REM START OF MAIN LOOP
80 REM*****
90 REPEAT
100 MODE5
110 PROCstartup
120 MAX%=8
130 PROCinit
140 PROCsetup
150 REPEAT
160 PROCscreen1
170 PROCscreen_num
180 PROCMUSIC
190 IF SCREEN%=2 PROCscreen2
200 IF SCREEN%=3 PROCscreen3
210 IF SCREEN%=4 PROCscreen4
220 IF SCREEN%=5 PROCscreen5
230 IF SCREEN%=6 PROCscreen6
240 IF SCREEN%=7 PROCscreen7
250 PROCMUSIC
260 PROCfleas
270 REPEAT
280 PROCTun(P%,D%,DD%)
290 PROCbonus
300 FLEA%=FLEA%+1
310 IF FLEA%=MAX% FLEA%=0
320 PROCmove_flea
330 PROCspider
340 IF Fnp(NEWX%,NEWY%,2) snuffedit
%:TRUE:PROCdeadspider:lives%=lives%-1
350 F%=RND(ABS(12-SCREEN%))
360 IFF%=1 AND SCREEN%=1 PROCmove_f
eet
370 IF SCREEN%=7 PROCfroghop
380 UNTIL dead%=20 OR snuffedit% 0
R BOX<1
390 IF BOX<1 AND lives%=1 THEN CLG
:PROCsetup
400 IF dead%=20 THEN CLG:SOUND&11,
4,0,40:PROCdelay(2):SC%=SC%+BOX:PROCs
core:PROCsetup:SCREEN%=SCREEN%+1:IF S
CREEN%>7 SCREEN%=1:MAX%=MAX%-1
410 IF lives%=1 AND snuffedit% THE
N CLG:PROCscore:PROClives:PROCsetup
420 UNTIL lives%<1:PROCgameover:MOD
E7:VDU23;8202;0;0;0;0;:PROCsoundoff:P
ROCscoretbl
430 CLS
440 PROCchiscores
450 UNTIL 0
460 VDU22,7:REPORT:PRINT" at line",
ERL
470 END
480 REM*****

```

```

490 REM END OF MAIN LOOP
500 REM*****
510 DEFPROCstartup
520 RESTORE3830
530 VDU23;8202;0;0;0;
540 VDU24,20;20;1260;880;
550 COLOUR130:8COL0,130:8COL0,129:C
LS:CLG
560 VDU24,34;36;1246;866;
570 8COL0,128:CLG:8COL0,1
580 ENDPROC
590
600 REM*****
***
610 DEFPROCinit_arrays
620 DIM block% 6:DIM buff% 20
630 DIMFLEAS(36),sc%(11),name$(11)
640 FOR NX=0 TO 11:sc%(NX)=ABS(10-N
%)*100:name$(NX)="The Micro User":NEX
T
650 ENDPROC
660
670 REM*****
***
680 DEFPROCinit
690 X=0:lives%=5:musicoft%=0
700 OSWORD=&FFF1
710 BPITCH%=43
720 FOOTflag%=1
730 ENVELOPE1,1,20,-20,10,10,0,0,0,
0,0,0,0,0
740 ENVELOPE2,128,30,-10,0,10,3,0,1
26,-10,0,0,126,0
750 ENVELOPE3,6,95,0,0,137,0,0,92,0
,0,-84,82,100
760 ENVELOPE4,1,1,0,0,200,0,0,126,0
,0,-126,126,126
770 ENVELOPE5,1,43,0,0,100,0,0,126,
0,0,-126,126,126
780 SC%=0:HIZ=0:SCREEN%=1
790 PROCchars
800 PROCfeet
810 ENDPROC
820
830 REM*****
*
840 DEFPROCfleas
850 FOR FLEA%=0 TO MAX%-1:PROCnew_f
lea(FLEA%):NEXT
860 FLEA%=0
870 ENDPROC
880
890 REM*****
**
900 DEFPROCsoundoff
910 SOUND&11,0,0,0
920 SOUND&12,0,0,0
930 SOUND&13,0,0,0
940 ENDPROC
950 REM*****
960 REM PLACE FEET
970 REM*****

```

```

980 DEFPROCfeet
990 FOOTflag%=1:VDU5
1000 NEWRIGHTx%=1800:OLDRIGHTx%=NEWR
IGHTx%
1010 NEWRIGHTy%=RND(24)*32+64
1020 OLDRIGHTy%=NEWRIGHTy%
1030 NEWLEFTy%=NEWRIGHTy%-64
1040 OLDLEFTy%=NEWLEFTy%
1050 NEWLEFTx%=NEWRIGHTx%-100:OLDLEF
Tx%=OLDRIGHTx%-100
1060 MOVE NEWRIGHTx%,NEWRIGHTy%:8COL
3,2:VDU226,227:MOVE NEWLEFTx%,NEWLEF
Ty%:VDU226,227,4
1070 ENDPROC
1080
1090 REM*****
*
1100 DEFPROCsetup
1110 TIME=0
1120 snuffedit%=FALSE
1130 dead%=0:FLEA%=0
1140 frog%=0:W%=40
1150 SOUND&11,5,17,15
1160 BOX=0:NEWX%=160:NEWY%=64:VDU5:
MOVENEWX%,NEWY%:8COL3,3:VDU224
1170 time_limit=RND(40)+40
1180 SC%=STR$(SC%):BO%=STR$(BOX):HI$
=STR$(HIZ):SCREEN%=STR$(SCREEN%):dead
%=STR$(dead%):lives%=STR$(lives%)
1190 OLDRIGHTx%=-1500:OLDLEFTx%=-150
0:NEWRIGHTx%=-1500:NEWLEFTx%=-1500
1200 ENDPROC
1210
1220 REM*****
1230 DEFPROCbonus
1240 COLOUR1
1250 VDU4
1260 BOX=999-TIME/5
1270 IF BOX<0 THEN BOX=0
1280 BO%=STR$(BOX)
1290 PRINTTAB(19-LEN(BO%),1);" ";BOX
;
1300 VDU5
1310 ENDPROC
1320 REM*****
**
1330 REM CREATE SCREENS 1 TO 7
1340 REM*****
**
1350 DEFPROCscreen1
1360 VDU4
1370 COLOUR1:PRINTTAB(0,1)"Score ";T
AB(10-LEN(SC%),1)SC%
1380 PRINTTAB(11,1)"Bonus ";TAB(20-L
EN(BO%),1)BOX
1390 COLOUR0:PRINTTAB(0,3)"Fleas ";T
AB(10-LEN(dead%))dead%:PRINTTAB(11,3)
"Lives";TAB(20-LEN(lives%),3)lives%
1400 VDU5
1410 ENDPROC
1420 DEFPROCscreen2
1430 8COL0,2

```



```

1440 VDU5
1450 wall$=STRING$(5,CHR$(228))
1460 MOVE32,704:PRINTwall$
1470 MOVE480,704:PRINTwall$
1480 MOVE920,704:PRINTwall$
1490 MOVE32,340:PRINTwall$
1500 MOVE480,340:PRINTwall$
1510 MOVE928,340:PRINTwall$
1520 MOVE256,544:PRINTwall$
1530 MOVE704,544:PRINTwall$
1540 MOVE256,192:PRINTwall$
1550 MOVE704,192:PRINTwall$
1560 ENDPROC
1570 DEFPROCscreen3
1580 GCOL0,2
1590 FORP%=0 TO 10:MOVE RND(34)*32+3
2,RND(26)*32+32:VDU229:NEXT
1600 FORP%=0 TO 10:MOVE RND(34)*32+3
2,RND(26)*32+32:VDU230:NEXT
1610 ENDPROC
1620 DEFPROCscreen4
1630 GCOL0,2
1640 MOVE160,700:MOVE350,700:PLOT85,
160,500
1650 MOVE350,700:MOVE350,500:PLOT85,
160,500
1660 MOVE930,700:MOVE1114,700:PLOT85
,930,500
1670 MOVE930,500:MOVE1114,500:PLOT85
,1114,700
1680 MOVE550,700:MOVE734,700:PLOT85,
550,500
1690 MOVE550,500:MOVE734,500:PLOT85,
734,700
1700 MOVE160,400:MOVE350,400:PLOT85,
160,200
1710 MOVE350,400:MOVE350,200:PLOT85,
160,200
1720 MOVE930,400:MOVE1114,400:PLOT85
,930,200
1730 MOVE930,200:MOVE1114,200:PLOT85
,1114,400
1740 MOVE550,400:MOVE734,400:PLOT85,
550,200
1750 MOVE550,200:MOVE734,200:PLOT85,
734,400
1760 ENDPROC
1770 DEFPROCscreen5
1780 GCOL0,2
1790 LOCAL X%,Y%
1800 FORX%=192 TO 1024 STEP256
1810 FORY%=100 TO 804 STEP 192
1820 MOVEX%,Y%:MOVEX%,Y%+96:PLOT85,X
%+92,Y%+96
1830 MOVEX%,Y%:MOVEX%+92,Y%:PLOT85,X
%+92,Y%+96
1840 NEXT Y%:NEXTX%
1850 ENDPROC
1860 DEFPROCscreen6
1870 GCOL0,2
1880 block$=CHR$228
1890 MOVE160,128:PRINTSTRING$(3,block$)

```

```

k$)
1900 MOVE160,800:PRINTSTRING$(3,block$)
k$)
1910 FORY%=128 TO 400 STEP32:MOVE160
,Y%:PRINTblock$:NEXT
1920 FORY%=512 TO 800 STEP32:MOVE160
,Y%:PRINTblock$:NEXT
1930 MOVE960,128:PRINTSTRING$(3,block$)
k$)
1940 MOVE960,800:PRINTSTRING$(3,block$)
k$)
1950 FORY%=128 TO 400 STEP32:MOVE108
8,Y%:PRINTblock$:NEXT
1960 FORY%=512 TO 800 STEP32:MOVE108
8,Y%:PRINTblock$:NEXT
1970 FORY%=224 TO 704 STEP32:MOVE320
,Y%:PRINTblock$:NEXT
1980 FORY%=224 TO 704 STEP32:MOVE928
,Y%:PRINTblock$:NEXT
1990 MOVE384,224:PRINTSTRING$(3,block$)
k$)
2000 MOVE384,704:PRINTSTRING$(3,block$)
k$)
2010 MOVE736,224:PRINTSTRING$(3,block$)
k$)
2020 MOVE736,704:PRINTSTRING$(3,block$)
k$)
2030 MOVE512,352:PRINTSTRING$(5,block$)
k$)
2040 MOVE512,576:PRINTSTRING$(5,block$)
k$)
2050 FORY%=384 TO 448 STEP32:MOVE512,Y%
:PRINTblock$:NEXT:MOVE768,384:PRINTblock$
2060 MOVE512,544:PRINTblock$:FORY%=4
80 TO 576 STEP32:MOVE768,Y%:PRINTblock$:
NEXT
2070 ENDPROC
2080 DEFPROCscreen7
2090 GCOL3,2
2100 NFX%=RND(800)+100:NFY%=RND(640)
+200
2110 MOVENFX%,NFY%
2120 VDU231,232,233,10,8,8,8,234,235
,236,10,8,8,8,237,238,239
2130 MOVENFX%,NFY%:GCOL0,3:VDU9,240
2140 OFX%=NFX%:OFY%=NFY%
2150 ENDPROC
2160 REM*****
*
2170 REM      MOVE FROG
2180 REM*****
*
2190 DEFPROCfroghop
2200 frogZ=frogX+1:IF frogZ < 20ENDPROC:
ELSE frogZ=0
2210 GCOL3,2
2220 MOVEOFX%,OFY%
2230 VDU231,232,233,10,8,8,8,234,235
,236,10,8,8,8,237,238,239
2240 MOVEOFX%,OFY%:GCOL3,3:VDU9,240
2250 GCOL3,2

```

```

2260 NFX%=RND(800)+100:NFY%=RND(640)
+200
2270 MOVENFX%,NFY%
2280 VDU231,232,233,10,8,8,8,234,235
,236,10,8,8,8,237,238,239
2290 MOVENFX%,NFY%:GCOL0,3:VDU9,240
2300 OFX%=NFX%:OFY%=NFY%
2310 ENDPROC
2320
2330 REM*****
*
2340 DEFPROCscore
2350 VDU4
2360 COLOUR1
2370 SC%=STR$(SCZ)
2380 PRINTTAB(10-LEN(SC%),1);SCZ
2390 VDU5
2400 ENDPROC
2410
2420 REM*****
*
2430 DEFPROCclives
2440 VDU4
2450 lives%=STR$(livesZ)
2460 PRINTTAB(20-LEN(lives%),3);live
s%
2470 ENDPROC
2480
2490 REM*****
**
2500 DEFPROCscreen_num
2510 VDU4
2520 SCREEN%=STR$(SCREENZ)
2530 COLOUR 130:COLOUR0
2540 PRINTTAB((14-LEN(SCREEN%))DIV 2
,16);"Screen:";SCREENZ:PROCdelay(2):P
RINTTAB((14-LEN(SCREEN%))DIV 2,16);:V
DU4:COLOUR128:PRINTTAB((14-LEN(SCREEN
%))DIV 2,16);SPC(8):COLOUR130:VDU5
2550 PROCMUSIC
2560 ENDPROC
2570
2580 REM*****
**
2590 DEFPROCgameover
2600 VDU4
2610 PRINTTAB(6,16)"Game over"
2620 VDU5
2630 ENDPROC
2640
2650 REM*****
***
2660 DEFPROCfleanum
2670 VDU4
2680 dead%=STR$(deadZ)
2690 PRINTTAB(10-LEN(dead%),3);deadZ
2700 ENDPROC
2710 DEFPROCmove_flea
2720 LOCAL X%,Y%,TX
2730 X%=FLEAS(FLEAX+0)
2740 Y%=FLEAS(FLEAY+12)

```



## From Page 31

```

2750 TX=FLEAS(FLEAZ+24)
2760 GCOL3,1
2770 MOVEXZ,YZ:VDU5:VDU225
2780 TX=TX+1
2790 IF TX= time_limit THEN SOUND&1
2,2,200,20:PROCnew_flea(FLEAZ):ENDPROC
C
2800 MOVEXZ,YZ:VDU5:VDU225
2810 FLEAS(FLEAZ+0)=XZ
2820 FLEAS(FLEAZ+12)=YZ
2830 FLEAS(FLEAZ+24)=TX
2840 ENDPROC
2850
2860 REM*****
2870 DEFPROCnew_flea(FLEAZ)
2880 GCOL3,1
2890 REPEAT
2900 FLEAS(FLEAZ+0)=RND(34)*32+32
2910 FLEAS(FLEAZ+12)=RND(26)*32+32
2920 FLEAS(FLEAZ+24)=RND(100)
2930 UNTIL POINT(FLEAS(FLEAZ+0),FLEA
S(FLEAZ+12))=0 AND POINT(FLEAS(FLEAZ+
0)+64,FLEAS(FLEAZ+12)+32)=0 AND POINT
(FLEAS(FLEAZ+0)-8,FLEAS(FLEAZ+12)-8)=
0
2940 MOVEFLEAS(FLEAZ+0),FLEAS(FLEAZ+
12):VDU5,225
2950 ENDPROC
2960
2970 REM*****
**
2980 DEFPROCcrunch
2990 SOUND&11,1,90,255
3000 SOUND&10,-15,7,3
3010 ENDPROC
3020 REM*****
3030 REM MOVE FEET
3040 REM*****
3050 DEFPROCmove_feet
3060 FOOTflagZ=FOOTflagZ EOR 1
3070 IF FOOTflagZ=0 NEWRIGHTXZ=NEWRI
GHTXZ-200
3080 IF FOOTflagZ=1 NEWLEFTXZ=NEWLEF
TXZ-200
3090 IF NEWRIGHTXZ<=-300 NEWRIGHTXZ=
0:PROCfeet:ENDPROC
3100 IF FOOTflagZ=1 GCOL3,2:MOVE NEW
LEFTXZ,NEWLEFTYZ:VDU226:MOVE NEWLEFTX
Z+64,NEWLEFTYZ:VDU227:PROCcrunch:MOVE
OLDLEFTXZ,OLDLEFTYZ:VDU226:MOVE OLDL
EFTXZ+64,OLDLEFTYZ:VDU227
3110 IF FOOTflagZ=0 GCOL3,2:MOVE NEW
RIGHTXZ,NEWRIGHTYZ:VDU226:MOVE NEWRI6
HTXZ+64,NEWRIGHTYZ:VDU227:PROCcrunch:
MOVE OLDRIGHTXZ,OLDRIGHTYZ:VDU226:MOV
E OLDRIGHTXZ+64,OLDRIGHTYZ:VDU227
3120 OLDRIGHTXZ=NEWRIGHTXZ:OLDLEFTXZ
=NEWLEFTXZ
3130 ENDPROC
3140 REM*****
*
```

```

3150 REM MOVE SPIDER
3160 REM*****
*
3170 DEFPROCspider
3180 OLDXZ=NEWXZ:OLDYZ=NEWYZ
3190 IF INKEY(-17) musicoffZ=-1
3200 IF INKEY(-102) musicoffZ=0
3210 IF INKEY(-66) NEWYZ=OLDYZ+32
3220 IF INKEY(-98) NEWYZ=OLDYZ-32
3230 IF INKEY(-103) NEWXZ=OLDXZ-32
3240 IF INKEY(-104) THEN NEWXZ=OLDXZ
+32
3250 IF NEWXZ<34 THEN NEWXZ=34
3260 IF NEWXZ>1186 THEN NEWXZ=1186
3270 IF NEWYZ<64 THEN NEWYZ=64
3280 IF NEWYZ>868 THEN NEWYZ=868
3290 IF NOT(NEWXZ=OLDXZ AND NEWYZ=OL
DYZ)MOVE NEWXZ,NEWYZ:GCOL3,3:VDU224:M
OVE OLDXZ,OLDYZ:VDU224
3300 IF POINT(NEWXZ+32,NEWYZ-16)=2 T
HEN PROCwhichflea
3310 ENDPROC
3320
3330 REM*****
*
3340 DEF PROCwhichflea
3350 LOCALCZ,NZ
3360 REPEAT
3370 IF POINT(FLEAS(NZ)+32,FLEAS(NZ+
12)-16)=2 THEN SOUND0,-15,4,2:GCOL3,1
:VDU5:MOVE FLEAS(NZ),FLEAS(NZ+12):VDU
225,5:PROCnew_flea(NZ):CZ=1:SCZ=SCZ+1
0:deadZ=deadZ+1:VDU4:PROCscore:PROCF1
eanum:VDU5
3380 NZ=NZ+1
3390 UNTIL CZ OR NZ>=MAXZ
3400 ENDPROC
3410
3420 REM*****
**
3430 DEFPROCdelay(SECS)
3440 LOCAL time
3450 time=TIME
3460 REPEAT:PROCmusic:UNTILTIME>=tim
e+(100*SECS)
3470 ENDPROC
3480
3490 REM*****
**
3500 DEFPROCchars
3510 VDU23,224,0,66,165,24,126,153,3
6,66
3520 VDU23,225,0,0,48,56,24,20,0,0
3530 VDU23,226,0,63,127,255,255,127,
63,0
3540 VDU23,227,0,128,206,207,207,206
,128,0
3550 VDU23,228,123,123,123,0,222,222
,222,222
3560 VDU23,229,255,255,255,126,126,1
26,60,60
3570 VDU23,230,60,60,126,126,126,255
,255,255
```

```

,255,255
3580 VDU23,231,0,0,0,1,1,1,1,1
3590 VDU23,232,0,66,231,255,24,24,24
,24
3600 VDU23,233,0,0,0,128,128,128,128
,128
3610 VDU23,234,1,1,3,3,25,60,62,31
3620 VDU23,235,255,255,195,189,126,2
55,255,255
3630 VDU23,236,128,128,192,192,152,6
0,124,248
3640 VDU23,237,31,15,7,3,3,7,14,124
3650 VDU23,238,255,128,102,102,102,1
02,102,231
3660 VDU23,239,248,240,224,192,192,2
24,112,62
3670 VDU23,240,0,0,0,0,231,231,132,1
32
3680 VDU23,241,0,0,0,0,99,99,0,0
3690 VDU23,242,0,0,0,0,0,99,99
3700 VDU23,243,0,0,60,66,129,0,0,0
3710 ENDPROC
3720
3730 REM*****
*
3740 DEFPROCtun(PZ,DZ,DDZ)
3750 IF musicoffZ ENDPROC
3760 IF ADVAL(-8)<8 ENDPROC
3770 IF BPITCHZ>101 BPITCHZ=43
3780 READPZ,DZ,DDZ
3790 SOUND3,-5,BPITCHZ+PZ,DZ
3800 SOUND3,0,0,DDZ
3810 IF PZ=0 AND DZ=0 AND DDZ=0 REST
ORE 3830:BPITCHZ=BPITCHZ+5:SOUND3,0,0
,40
3820 ENDPROC
3830 DATA 0,8,4,28,4,8,28,8,0,16,4,0
,8,8,0,16,4,0,0,12,0,28,12,0,28,16,8
,48,8,0,28,8,4,28,8,0,36,4,0,28,8,0,20
,4,0,16,12,0,0,8,0,8,16,8
3840 DATA 36,8,0,32,4,0,36,8,0,44,4
,0,48,8,0,44,4,0,48,8,0,56,4,0,64,8,0
,56,4,0,64,8,0,68,4,0,76,16,8,76,8,0,6
4,4,0,48,8,0,56,4,0,64,8,0,56,4,0,64
,4,0,76,4,16,28,8,0,48,16,0,0,0,0
3850
3860 REM*****
*
3870 DEFPROCmusic
3880 PROCtun(PZ,DZ,DDZ)
3890 ENDPROC
3900
3910 REM*****
*
3920 DEF FNp(XZ,YZ,CZ)
3930 =POINT(XZ,YZ-16)=CZ OR POINT(XZ
,YZ)=CZ OR POINT(XZ+60,YZ)=CZ OR POI
NT(XZ+32,YZ)=CZ
3940
3950 REM*****
**
3960 DEFPROCdeadspider
```



```

3970 SOUND#11,3,100,16
3980 ENDPROC
3990
4000 REM*****
***
4010 DEFPROCinst
4020 LOCALXXZ,YYZ
4030 FORYZ=1 TO 4
4040 PRINT TAB(6,YZ)CHR$129;CHR$157;
TAB(35,YZ)CHR$156
4050 NEXT
4060 PROCdbl(" W H I T E W I D O W
",7,2,7)
4070 RESTORE 4160
4080 REPEAT
4090 READXXZ,YYZ
4100 PRINTTAB(XXZ,YYZ)CHR$255
4110 UNTILXXZ=24 AND YYZ=18
4120 FORYZ=21 TO 24
4130 PRINTTAB(6,YZ);CHR$132;CHR$157;
TAB(35,YZ);CHR$156;
4140 NEXT
4150 PROCdbl("by Alan Sergeant",11,2
2,7)
4160 DATA 16,7,25,7,15,8,17,8,20,8,2
1,8,24,8,26,8,14,9,18,9,20,9,21,9,23,
9,27,9,16,10,17,10,19,10,20,10,21,10,
22,10,24,10,25,10,15,11,18,11,19,11,2
0,11,21,11,22,11,23,11,26,11,14,12,17
,12,18,12,19,12,20,12,21,12,22,12,23,
12,24,12,27,12
4170 DATA 16,13,18,13,19,13,20,13,21
,13,22,13,23,13,25,13,15,14,19,14,20,
14,21,14,22,14,26,14,15,15,18,15,20,1
5,21,15,23,15,26,15,17,16,24,16,17,17
,24,17,17,18,24,18,999,999
4180 A$=INKEY$(200)
4190 PROCscroll
4200 RESTORE 3830
4210 FORYZ=10 TO 13:PRINTTAB(11,YZ);
CHR$129;CHR$157;TAB(28,YZ);CHR$156;N
EXT
4220 PROCdbl(CHR$136+" INSTRUCTIONS?
",10,11,3)
4230 REPEAT :G=GET
4240 UNTIL G=78 OR G=89
4250 IF G=78 THEN ENDPROC
4260 PROCscroll
4270 PRINTTAB(14,2)CHR$141;CHR$130;"
ARACHNIDA"
4280 PRINTTAB(14,3);CHR$141;CHR$131;
"ARACHNIDA"
4290 PROCsng("is a lady and lives al
one.",5,3)
4300 PROCsng("She must survive on a
diet of fleas.",7,3)
4310 PROCsng("Life won't be easy. Th
e garden walls",9,3)
4320 PROCsng("and all obstacles, hav
e been painted",11,3)
4330 PROCsng("with a deadly yellow p
oison.",13,3)

```

```

4340 PROCsng("She must eat 20 fleas
before moving",16,1)
4350 PROCsng("to a different locatio
n.",18,1)
4360 PROCsng("Press any key to conti
nue",22,3)
4370 A=GET
4380 PROCscroll
4390 PROCsng("Move Arachnida around
using these keys",4,3)
4400 PROCdbl("A - UP",11,7,1)
4410 PROCdbl("Z - DOWN",11,9,1
)
4420 PROCdbl("< - LEFT",11,11,
1)
4430 PROCdbl("> - RIGHT",11,13
,1)
4440 PROCdbl("Q/M turn OFF/ON the mu
sic",6,16,1)
4450 PROCsng("REMEMBER, touch anythi
ng yellow",19,3)
4460 PROCsng("or run out of time and
you lose a life.",21,3)
4470 PROCsng("Press any key to conti
nue",24,6)
4480 A=GET
4490 PROCscroll
4500 PROCsng("You score 10 for each
flea collected",3,3)
4510 PROCsng("plus the bonus left on
a cleared screen",5,3)
4520 PROCsng("8 fleas appear on scre
en at the start.",8,3)
4530 PROCsng("As you succeed on the
first 7 screens",10,3)
4540 PROCsng("this total is reduced
by one per round.",12,3)
4550 PROCsng("Don't expect the fleas
to hang about!",16,3)
4560 PROCsng("They don't particularl
y like spiders!",19,1)
4570 PROCsng("Press any key to start
",23,6)
4580 A=GET
4590 CLS
4600 ENDPROC
4610
4620 REM*****
*
4630 DEFPROCscroll
4640 coZ=0
4650 REPEAT
4660 PRINTTAB(39,24)" "
4670 coZ=coZ+1
4680 FORDELAY=1 TO 200:NEXT
4690 UNTILcoZ>16
4700 VDU31,0,0
4710 ENDPROC
4720
4730 REM*****
**

```

```

4740 DEFPROCdbl(msg$,XZ,YZ,CZ)
4750 FOR NZ=0 TO 1:PRINT TAB(XZ,YZ+N
Z);CHR$141;CHR$(128+CZ);msg$;NEXT
4760 ENDPROC
4770 DEFPROCsng(msg$,YZ,CZ)
4780 XZ=(40-LEN(msg$))/2
4790 PRINTTAB(XZ,YZ)CHR$(128+CZ);msg
$;
4800 ENDPROC
4810
4820 REM*****
***
4830 DEFPROCscoretbl
4840 N=0:REPEATN=N+1
4850 UNTIL SCZ>scZ(N) OR N=11
4860 IF N=11 ENDPROC
4870 FOR N1=11 TO N STEP-1:scZ(N1)=s
cZ(N1-1):name$(N1)=name$(N1-1):NEXT
4880 scZ(N)=SCZ
4890 CLS
4900 PROCdbl(CHR$(136)+"Congratulati
ons",10,6,7):PROCsng("Your score is i
n the top ten",12,6):PROCsng("Please
enter your name",14,6)
4910 PRINTTAB(10,16)CHR$(134)CHR$(15
7)CHR$(129);SPC(18);CHR$(156);TAB(13,
16);
4920 blockZ?0=buffZ MOD 256
4930 blockZ?1=buffZ DIV 256
4940 blockZ?2=17:REM maximum number
of characters.
4950 blockZ?3=32:REM minimum ASCII v
alue = 32
4960 blockZ?4=126:REM maximum ASCII
value = 126
4970 XZ=blockZ MOD 256
4980 YZ=blockZ DIV 256
4990 AZ=0
5000 PROCdelay(1)
5010 *FX12,0
5020 *FX15,1
5030 CALL OSWORD
5040 name$(N)=$buffZ
5050 ENDPROC
5060
5070 REM*****
**
5080 DEFPROCchiscores
5090 REPEAT
5100 PROCdbl(CHR$(157)+CHR$(131)+"
White Widow Hall Of Fame.",0,0,1):P
RINTTAB(38,0);CHR$(156);TAB(38,1);CHR
$(156);
5110 FOR NZ=1 TO 10:PRINT TAB(8,NZ*2
+2);CHR$(RND(6)+128);scZ(NZ);TAB(16,N
Z*2+2);CHR$(RND(6)+128);name$(NZ);NE
XT
5120 PROCsng("Press space bar for a
new game",24,7)
5130 *FX15,1
5140 UNTIL INKEY(-99)
5150 ENDPROC

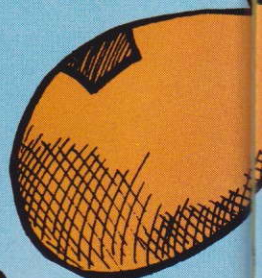
```



# The continuing saga of Bill the Spaceman and his quest for fame

## Bill bow bac

By  
**SIMON  
PHIPPS**



WE thought he had gone, but Simon Phipps returns yet again with a sequel to the highly exciting *Dug-Dig* (*Micro User*, May 1984). This time though, he had incorporated a section of machine code in his latest marvel to make the game a little faster and obviously that much harder.

Without further delay we continue the saga of Spaceman Bill and his quest

for fame, fortune and an Acorn Electron. (How did that get in?)

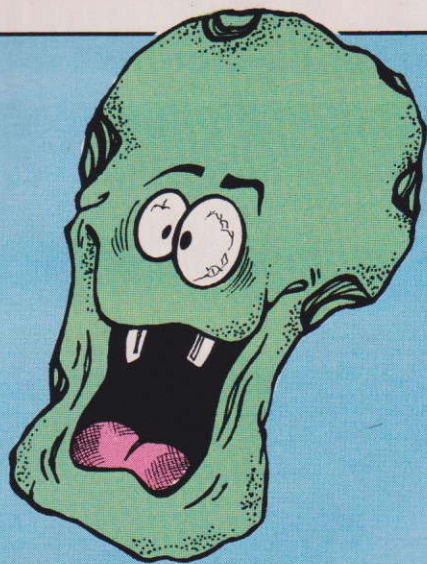
There he was with a handful of krystals on a strange asteroid away in the Hilton system. It was quite risky this mining business and Bill was a little worried about the fact that he might not be able to spend all of his profits, what with those nasty Scrugolds and all.

So packing another gem into his

starship and zapping a last alien he blasted off for territories unknown and relaxed with a jinantonnix and his copy of the book that caused him all the trouble in the first place.

He had reached chapter 21,884 by the time the warp drive had cut in when suddenly he found what he thought might be a safer way of making money. Gugvunt Bleuch, author of the best





nces  
k!



selling "Jewels of the Universe" had made a killing on his book, but Bill was no author, so turning the page he found a section on the sacred Green Teapots of Quarg.

Wasting no time Bill punched the right coordinates into his flight computer and headed for the Betel-joolien star cluster and once more (hopefully) fortune. (He was starting to

give up on the fame bit.)

Bill was now certain about everything. Surely nothing dangerous could happen. According to the book the strange teapots were piled high all over the planet Neeguss – the umptifirst moon of the star cluster.

However he was a trifle worried about the fact that the last page of that chapter had been torn out.

As it happened the last page did give a warning along the lines of "Keep away from it mate, it's far too dangerous", but as Bill didn't know this he landed his starship with an ungainly thud on the alien terrain.

He had noticed on the way down that there was a great deal of low cloud cover



## From Page 35

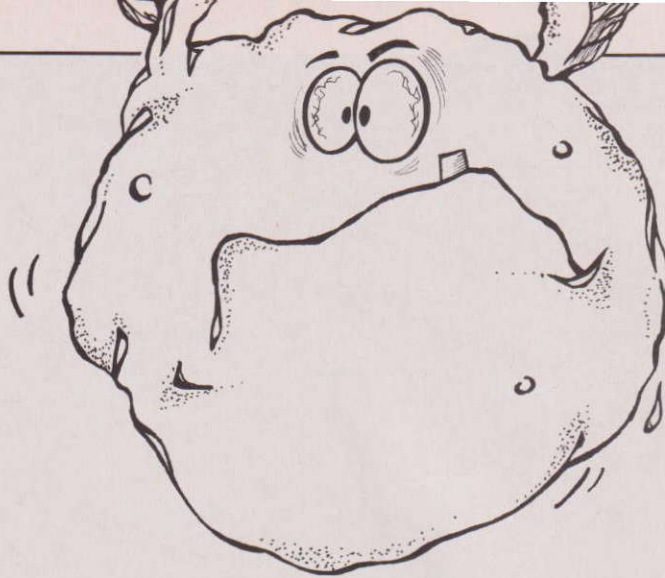
and that it moved quickly and erratically across the planet's surface.

Had Bill realised that the clouds were really vaporoids, a strange life form only found in a few video games across the galaxy, he probably wouldn't have landed. But land he did.

Of course that wasn't all that was stopping Bill from having an easy life. The planet's population mainly consisted of Uecchian Cloudhoppers, small cute and cuddly nasties that spent their whole lives jumping up and down, annoying unwary space travellers and consuming the little oxygen that remains in the planet's atmosphere.

Bill dived out of his starship. Taking a large gulp of air he ran towards the teapots. Night was falling and he knew that any collision with a Cloudhopper would mean certain death. Can he make it? Get typing and find out...

It really is quite simply a matter of collecting some objects and returning



them to your waiting starship, avoiding numerous space nasties and only having one minute's supply of air to do it in – a theme that has become rather popular recently.

The game hinges around the large section of assembly language (lines 370-520) which controls no less than 18 separate objects on the screen at once.

This technique uses the rather

“illegal” method of directly accessing the screen, but seeing that many leading software houses use this method it is quite acceptable.

Just to make the game harder, as each level is progressed night begins to fall (literally) as the stars come out and the Cloudhoppers refuse to jump quite as high into the cold night air...

Have fun...!

## VARIABLES

<b>A%</b>	Variable used in PROC-MAN to find out what must be added to Bill's x coordinate value depending upon the keys pressed and in FNPEEK(x%,y%) to aid in the calculation of the Ascii code of the character position (x%,y%).	<b>H%</b>	otherwise it will go down. Used in the generation of the starfield.	<b>o%</b>	Variable passing on OPT value to the assembler routine.
<b>C%</b>	The value of the address from which the alien and cloud definitions are stored.	<b>HI%</b>	Table of the hi-bytes of the cloud's positions.	<b>x%</b>	Variable used to pass various x values to PROC\$.
<b>CAR%</b>	A flag indicating if Bill is carrying a teapot, if zero then no object is being carried.	<b>HY%</b>	Table of the hi-bytes of the alien's positions.	<b>y%</b>	Variable used as x% to pass various y values to PROC\$.
<b>CODE%</b>	Value of the address from which the assembler code is assembled into.	<b>J%</b>	Used to generate the starfield.	<b>MAN\$</b>	Holds the multicoloured definition for the Bill character.
<b>D%</b>	Used in the generation of the starfield to calculate the base y coordinate for the stars being plotted.	<b>KIL%</b>	A flag indicating if Bill is still alive, 0 indicates yes, otherwise it's splat!	<b>a\$</b>	Used in PROCP(x%,y%,a\$) to enable a string to be passed to that procedure.
<b>DIR%</b>	Value of the address from which the cloud direction table is stored. A value in the table of 2 indicates the cloud moves left, a value of 1 moves it right.	<b>LO%</b>	Table of the lo-bytes of the cloud's positions.	<b>A</b>	Calculates various screen addresses in PROCMEM.
<b>DY%</b>	Value of the address of the start of the alien direction table. A value in this table of 2 moves the alien up,	<b>LY%</b>	Table of the lo-bytes of the alien's positions.	<b>B</b>	Execution address of cloud movement routine.
		<b>P%</b>	Program counter, where the assembler code is presently being assembled into.	<b>C</b>	Execution address of cloud initialisation routine.
		<b>Q%</b>	Bill's x coordinate.	<b>D</b>	Execution address of alien in initialisation routine.
		<b>S%</b>	Player's present score.	<b>E</b>	Execution address of alien movement routine.
		<b>SC%</b>	The number of screens successfully completed.	<b>JJ</b>	} Loop variables.
		<b>T%</b>	General variable used for virtually everything – in fact my favourite variable.	<b>KK</b>	
		<b>TEA%</b>	Number of teapots collected.	<b>X</b>	Used in PROCMEM to generate random screen addresses.
		<b>X%</b>	Start address of table of x values of the cloud's positions.	<b>Y</b>	Used as X in PROCMEM to generate random screen addresses.
		<b>Y%</b>	Start address of table storing Cloudhopper y coordinates.		



## FUNCTIONS

**FNPEEK** Taken almost exactly from the User Guide, checks the character at the position (x%,y%) and returns its Ascii code.



## CONTROLS

Z Left  
/ Right

The game is not very tight on memory and will run quite happily on a standard BBC Micro with 32k. Disc users please note that to run the game with PAGE set to &1900 all unnecessary spaces must be removed and some of the lines may need packing together.

## PROCEDURES

### PROCASS

Contains the assembler code for the aliens and clouds.

### PROCBONUS

Calculates the bonus points given for completing one screen.

### PROCCED

Splats Bill about the bottom of the screen.

### PROCMAN

Moves Bill around the screen.

### PROCMEM

Initialises the cloud and alien positions.

### PROCP

Yet again simulates the VDU:PRINT TAB(x,y)a\$ fea-

### PROCScore

ture available on the 0.1 OS BBC Micro.

Prints out the player's score in the box at bottom of the screen.

### PROCScreen

Draws out screen.

### PROCTITLE

Prints out start up display - "The Micro User presents" etc.

### PROCVARI

Sets up variables Q%, TEA%, CAR% and KIL%.

### PROCcentre

Centres text on screen, as passed to procedure by a\$.

1REM BILL BOUNCES B  
ACK!

2REM By Simon Phipps (C) The Micr  
o User

10\*FX9,7

20\*FX10,7

30VDU23,224,127,127,127,0,247,247,  
247,0

40VDU23,225,8,28,192,95,125,125,62  
,0

50VDU23,226,8,28,0,0,0,64,32,0

60VDU23,227,24,24,60,126,255,255,2

55,153,23,228,24,0,0,24,24,0,0,23,2  
29,0,0,0,0,129,129,153,153

70VDU23,230,56,56,16,254,16,40,40,  
108,23,231,0,0,0,56,16,40,40,0

80VDU23,232,128,72,1,34,132,52,110  
,255

90ENVELOPE3,1,-10,-37,-89,24,13,46  
,43,106,64,64,135,0

100ENVELOPE1,1,12,-5,15,1,1,60,127,  
0,127,127,-127,-127

110MODE7

120DIMCODEZ%1D0

130DIMXZ8

140DIMDIRZ8

150DIMHIZ8

160DIMLOZ8

170DIMCX95

180DIMYZ8

190DIMHYZ8

200DIMLYZ8

210DIMDYZ8

220PROCASS(0):PROCASS(2)

230MODE2:SZ=0:SCZ=0:PROCTITLE

240REPEAT:UNTILGET=32

241 MENX=3

250PROCMEM

260PROCVARI

270MODE2:VDU23;8202;0;0;0;

280VDU28,0,31,19,28,17,128+2,12,20,

26

290PROCScreen

300?&73=8:TIME=0

310CALLC:CALLD:REPEAT:CALLB:CALLE:P  
ROCMAN:DIRZ?(RND(9)-1)=RND(2):UNTILKI  
LX<>0OR (TEAZ=10AND QZ=0)

320IFTEAZ=10THENPROCBONUS:GOTO250

330PROCCED

331 MENX=MENX-1:IF MENX>0 THEN PROC  
VARI:PROCMEM:GOTO270

340COLOUR3:PRINTTAB(5,15)"GAME OVER  
":FORKK=0TO4000:NEXT:GOTO230

350DEFPROCASS(oz)

360PZ=CODEZ:[OPTOZ

370.C:LDX&73:.L1:LDA#CXDIV256:STA&7  
1:LDA#CXMOD256:STA&70:LDAHIZ,X:STA&81

:LDALOX,X:STA&80:JSR0n:DEX:BPL1:RTS  
380.B:LDX&73:.L2:LDADIRZ,X:CMP#2:BE

Q LEFT:LDADIRZ,X:CMP#1:BEQ RIGHT:JMP  
RET1:.RET:.RET1:DEX:BPL2:RTS

390.LEFT:LDAHIZ,X:STA&81:LDALOX,X:S  
TA&80:JSR0ff:LDAZX,X:CMP#0:BEQOFF:CL  
C:LDAZX,X:SBC#0:STAXZ,X:CLC:LDA LOZ,X  
:SBC#7:STA LOZ,X:LDA HIZ,X:SBC#0:STA  
HIZ,X:LDA#CXDIV256:STA&71:LDA#CXMOD25  
6:STA&70:LDAHIZ,X:STA&81:LDALOX,X:STA  
&80:JSR0n:JMPRET

400.RIGHT:LDAHIZ,X:STA&81:LDALOX,X:  
STA&80:JSR0ff

410LDAZX,X:CMP#68:BEQLOFF:CLC:LDAZX  
,X:ADC#1:STAXZ,X:CLC:LDA LOZ,X:ADC#8:  
STA LOZ,X:LDA HIZ,X:ADC#0:STA HIZ,X:L  
DA#(CX+64)DIV256:STA&71:LDA#(CX+64)MO  
D256:STA&70:LDAHIZ,X:STA&81:LDALOX,X:  
STA&80:JSR0n:JMPRET:.ROFF:LDA#1:STA D  
IRZ,X:JMP RET

420.LOFF:LDA#2:STA DIRZ,X:JMP RET

430.on:LDY#31:.L3:LDA(&70),Y:STA(&8  
0),Y:DEY:BPL L3:RTS:.off:LDY#31:.L4:L  
DA#0:STA(&80),Y:DEY:BPL L4:RTS

440.D:LDX&73:.L5:LDA#(CX+32)DIV256:  
STA&71:LDA#(CX+32)MOD256:STA&70:LDAH  
Y,X:STA&81:LDALYZ,X:STA&80:JSR0n:DEX:

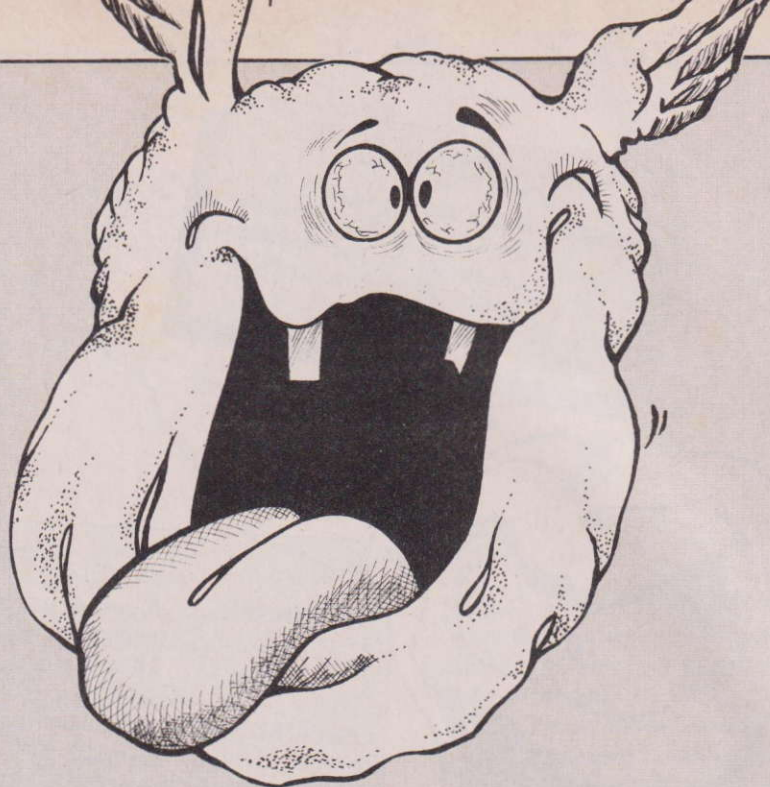


# From Page 37

```

BPLL5:RTS
450.E:LDX#73;.L6
460LDA DYZ,X:CMPI#2:BEQ UP:JMP DOWN:
.RET3
470LDA#(CX+32)DIV256:STA#71:LDA#(CX
+32)MOD256:STA#70:LDANYZ,X:STA#81:LDA
LYZ,X:STA#80:JSR#DEX:BPLL6:RTS
480.UP:LDAYZ,X:CMPI#0:BEQ C1:CLC:LDA
LYZ,X:SBC#7F:STA#76:LDANYZ,X:SBC#2:S
TA#77:LDY#31;.C2:LDA(&76),Y:CMPI#0:BNE
C1:DEY:BPLC2:CLC:LDAYZ,X:SBC#0:STAYZ
,X:LDANYZ,X:STA#81:LDALYZ,X:STA#80:JS
Roff:CLC:LDA#76:STALYZ,X:LDA#77:STAHY
Z,X:JMP RET3;.C1
490LDA#1:STA DYZ,X:JMP RET3
500.DOWN:LDAYZ,X:CMPI#27:BEQ D10:CLC
:LDALYZ,X:ADC#80:STA#76:LDANYZ,X:ADC
#2:STA#77:LDY#31;.D2:LDA(&76),Y:CMPI#0
:BNE D1:DEY:BPLD2:CLC:LDAYZ,X:ADC#1:S
TAYZ,X:LDANYZ,X:STA#81:LDALYZ,X:STA#8
0:JSRoff:CLC:LDA#76:STALYZ,X:LDA#77:S
TANYZ,X:JMPRET3
510.D1:LDA#2:STA DYZ,X:JMP RET3
520.D10:JMP D1
530RTS:J:ENDPROC
540DEFPROCMEH
550FORTX=0T08
560X=RND(60)
570Y=TX+15
580A=&3000+Y#640+X#8
590XX?TX=X
600HIX?TX=ADIV256
610LOX?TX=AMOD256
620DIRX?TX=RND(2)
630Y=25:X=TX#2+1:X=X#4
640YZ?TX=Y
650A=&3000+Y#640+X#8
660HYX?TX=ADIV256
670LYX?TX=AMOD256
680DYX?TX=1
690MAN#=#CHR$18+CHR$0+CHR$7+CHR$230+
CHR$8+CHR$18+CHR$0+CHR$4+CHR$231+CHR$
4
700NEXT
710RESTORE:FORTX=0T095:READCX?TX:NE
XT:ENDPROC
720ENDPROC
730DATA0,0,21,63,63,63,21,0,0,63,63
,63,62,63,63,62,0,0,63,63,63,61,62,40
,0,0,0,42,62,63,42,0
740DATA85,149,1,1,1,1,0,0,1,19,63,2
13,35,0,19,1,34,51,63,213,19,0,51,2,1
70,106,34,34,34,34,0,0
750DATA0,0,0,21,61,63,21,0,0,21,63,
63,63,62,61,20,0,42,63,63,61,63,63,61
,0,0,42,63,63,63,42,0
760DEFPROCSCREEN
770VDU19,14,0;0;:COLOUR142:VDU28,0,
SCX,19,0,12,17,128:FORHX=0T0SCX:DZ=(3
1-HX)*32-4:GCOL0,7:FORJX=0T010:PL0T69

```



```

,RND(1280),RND(32)+DX:NEXT:NEXT
780COLOUR128
790VDU28,0,31,19,28,17,130,12,17,12
8,26
800COLOUR4:COLOUR134
810PRINTTAB(0,28)STRING$(60,CHR$224
);
820COLOUR128
830COLOUR7:PRINTTAB(7,31);"000000";
840FORTX=0T09:MOVE1216,(31-TX-15)*3
2-4:VDU5,18,0,2,225,8,18,0,3,226,4:NE
XT
850MOVE0,256:VDU5,18,0,3,227,18,0,2
,8,228,18,0,1,8,229,4
860PROCP(0,27,MAN$)
861FORTX=1 TO MENX:PROCP(16+TX,31,M
AN$):NEXT
870ENDPROC
880DEFPROC(xZ,yZ,a$):VDU5:MOVEX#6
4,(32-YZ)*32-4:PRINT;a$;VDU4:ENDPROC
890DEFPROCMAN
900#FX15,1
910COLOUR7:PRINTTAB(0,31);" ";(60-T
IMEDIV100);" ";
920AX=INKEY(-98)-INKEY(-105)
930PRINTTAB(QZ,27);" ";
940IFFNPEEK(QZ,26)<>32THENKILX=1
950IFAX=-1ANDQZ<>0THENQZ=QZ-1:IFFNP
EEK(QZ,27)<>32THENKILX=1
960IFAX=1ANDQZ<>19THENQZ=QZ+1:IFFNP
EEK(QZ,27)<>32THENKILX=1
970IFAX<>0SOUND#11,1,65,1
980PROCP(QZ,27,MAN$)
990IFQZ=19ANDCARX=0THENCARX=1:SOUND
1,-15,160,1:TEAX=TEAX+1:PRINTTAB(19,1
5+TEAX);" ";
1000IFQZ=0ANDCARX=1THENCARX=0:SOUND1
,-15,35,1:SX=SX+10:PROCScore
1010IFTIME>6000THENKILX=1
1020ENDPROC
1030DEFFNPEEK(xZ,yZ)

```

```

1040LOCALAX,CX
1050VDU31,xZ,yZ
1060AX=135
1070CX=USR(&FFF4)
1080CX=CXAND&FFFF
1090CX=CXDIV&100
1100=CX
1110DEFPROCVAR1:QZ=0:TEAX=0:CARX=0:K
ILX=0:ENDPROC
1120DEFPROCSCORE:COLOUR7:PRINTTAB(7,
31);STRING$(6-LEN(STR$(SX)),"0");SX;:
ENDPROC
1130DEFPROCDED: SOUND3,3,119,45:FORKK
=0T0100:CALLB:CALC:COLOUR1:PRINTTAB(
QZ,27);CHR$232:FORJJ=0T020:NEXT:NEXT:
ENDPROC
1140DEFPROCbonus:TX=60-TIMEDIV100:CO
LOUR3:PRINTTAB(3,15)"TIME BONUS=";TX*
10*(SCX+1):SCX=SCX+1:IFSCX=15SCX=14
1150SX=SX+TX*SCX:PROCScore:FORKK=0T0
4000:NEXT:ENDPROC
1160DEFPROCTITLE
1170VDU23;B202;0;0;0;
1180PRINT''
1190COLOUR5
1200PROCcentre("The Micro User")
1210PRINT''
1220COLOUR1
1230PROCcentre("presents")
1240PRINT''''
1250COLOUR11
1260PROCcentre("BILL BOUNCES BACK!")
1270PRINT''':COLOUR1
1280PROCcentre("by")
1290PRINT''':COLOUR10
1300PROCcentre("Simon Phipps")
1310COLOUR15:PRINTTAB(0,30);:PROCcen
tre("PRESS SPACE TO PLAY")
1320ENDPROC
1330DEFPROCcentre(a$):PRINTTAB(10-LE
N(a$)/2);a$:ENDPROC

```



# Three programs everyone's talking about!

**Play the most exciting,  
challenging Olympic  
simulation of them all!**



**MICRO OLYMPICS** is riding high in the charts – and no wonder. You are challenged to beat the world record in 11 realistic track and field events, with the packed stadium cheering you on to victory – or defeat! Just like the real thing, you'll want to keep on improving your sporting skill until you're up there among the greats!



**Turn your home micro into  
a true office workhorse!**

**MINI OFFICE** is the most incredibly priced introduction to business computing ever offered. The package contains four full-scale programs – word processor, database manager, spreadsheet and graphics. They include advanced features not yet available on programs costing many times as much. Plus a concise 32-page how-to-do-it booklet.

**Let your children embark on a  
magical micro adventure!**

**THE MAGIC SWORD** breaks new ground in programming for the young – a complete adventure on cassette accompanied by a 48-page full colour storybook. Stunning sound and colourful graphics bring a completely new dimension to the classic computer adventure. And there's a very helpful animated compass to help you find your way around!



**Send  
for  
them  
today**

Please supply software as ticked in price boxes

☐ I enclose cheque made payable to  
Database Publications Ltd. for £ \_\_\_\_\_

I wish to pay by ☐ Access ☐ Visa

No. \_\_\_\_\_ Expiry date \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

Signed \_\_\_\_\_

Post to: Database Publications, Europa House,  
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

	Micro Olympics	Mini Office	Magic Sword
BBC 'B' cassette	£5.95 <input type="checkbox"/>	£5.95 <input type="checkbox"/>	£8.95* <input type="checkbox"/>
Electron cassette	£5.95 <input type="checkbox"/>	£5.95 <input type="checkbox"/>	
BBC 40 track disc	£7.95 <input type="checkbox"/>	£7.95 <input type="checkbox"/>	£9.95 <input type="checkbox"/>
BBC 80 track disc	£7.95 <input type="checkbox"/>	£7.95 <input type="checkbox"/>	£9.95 <input type="checkbox"/>
Spectrum cassette	£5.95 <input type="checkbox"/>	Coming soon	£8.95* <input type="checkbox"/>
Commodore cassette	£5.95 <input type="checkbox"/>	Coming soon	

\* Versions for both machine on same cassette



# Electronequip

(Authorised BBC Micro Dealer, and Econet service centre)

BBC Cassette Software	Exc VAT	Inc VAT
BCAAFRK Aardvark Frak (Caveman game)	7.00	8.05
BCAAZALA Aardvark Zalaga	7.00	8.05
BCADFOOT Addictive Football Manager	5.87	6.75
BCAFBOUN A & F Software Bouncer	5.92	6.80
BCAFCHIC A & F Software Chuckie Egg	5.84	6.72
BCAFCYLO A & F Software Cylon Attack	5.84	6.72
BCAFFROG A & F Software Frogger	5.92	6.80
BCAFPAIN A & F Software Painter	5.92	6.80
BCAFSHRI A & F Software Shrinking Professor	5.92	6.80
BCAIAOVE Adventure International AdventurLand	7.35	8.46
BCAIPAIR Adventure International Pirate Adventure	7.35	8.46
BCAISECR Adventure International Pirate Mission	7.35	8.46
BCAIVOOD Adventure International Voodoo Castle	7.35	8.46
BCAKCHIL ASK Children from Space (XBE16)	7.35	8.46
BCAKCRAN ASK Cranky (XBE17)	7.35	8.46
BCAKFACE ASK Facemaker (XBE10)	7.35	8.46
BCAKHIDE ASK Hide & Seek (XBE11)	7.35	8.46
BCAKLETS ASK Let's Count (XBE12)	7.35	8.46
BCAKNUMC ASK Number Chase (XBE15)	7.35	8.46
BCAKNUMG ASK Number Gunner (XBE13)	5.82	6.70
BCAKNUMP ASK Number Puzzler (XBE14)	7.35	8.46
BCAKWORD ASK Words, Words, Words (XBE19)	7.35	8.46
BCALBUGB Alligata Software Bug Blaster	5.88	6.76
BCALCOSM Alligata Software Cosmic Asteroids	4.39	5.05
BCALDABL Alligata Software Dambusters	5.87	6.75
BCALEAGL Alligata Software Eagle Empire	5.87	6.75
BCALFRUI Alligata Software Fruit Machine (BO06)	4.39	5.05
BCALLUNCL Alligata Software Uncle Claude	7.00	8.05
BCAMALGE Ampsoft Chessmate Cat Algebra 0 level	5.13	5.90
BCAMARIT Ampsoft Chessmate Cat Arithmetic 0 level	5.13	5.90
BCAMCALC Ampsoft Chessmate Cat Calculus 0 level	5.13	5.90
BCAMCONC Ampsoft Chessmate Cat Construction "0"	5.13	5.90
BCAMGRAMP Ampsoft Chessmate Cat Graphs 0 level	5.13	5.90
BCAMTRIG Ampsoft Chessmate Cat Trigonometry "0"	5.13	5.90
BCAS100P Acornsoft 100 Programs for BBC Micro	8.50	9.78
BCASACT Acornsoft Arcade Action (SBG06)	8.80	10.12
BCASARCA Acornsoft Arcadians (SBG14)	7.35	8.46
BCASAVIA Acornsoft Avatar flight simulator SBG02	11.05	12.71
BCASBUSI Acornsoft Business Games (SBG03)	7.35	8.46
BCASCARO Acornsoft Carousel (SBG24)	7.35	8.46
BCASCAST Acornsoft Castle of Riddles (SBG17)	7.35	8.46
BCASCHAN Acornsoft Chemical Analysis (SBG12)	10.20	11.73
BCASCHES Acornsoft Chess (SBG10)	7.35	8.46
BCASCHSI Acornsoft Chemical Simulations (SBG13)	10.20	11.73
BCASCHST Acornsoft Chemical Structures (SBG14)	10.20	11.73
BCASCOLL Acornsoft ICL Collectors Cat SBX05	7.35	8.46
BCASCOUN Acornsoft Countdown to Doom (SBG19)	7.35	8.46
BCASCRAZ Acornsoft Crazy Tracer (SBG26) (B)	7.35	8.46
BCASCREA Acornsoft Creative Graphics (SBX01)	7.35	8.46
BCASDESK Acornsoft Desk Diary (SBG01)	7.35	8.46
BCASDRAU Acornsoft Draughts & Reversi (B) (SBG20)	7.35	8.46
BCASDROG Acornsoft Droga (SBG27)	7.78	8.95
BCASELIT Acornsoft Elite (SBG38)	12.35	14.20
BCASEXAM Acornsoft Examiner Cassette (SBG17) (B)	7.35	8.46
BCASFORE Acornsoft Forecast (SBG02)	8.80	10.12
BCASFORT Acornsoft Fort Cassette (SBG01)	12.45	14.32
BCASFREE Acornsoft Free Fall (SBG28)	7.35	8.46
BCASFREN Acornsoft French (Linkword) (SBX13)	11.05	12.71
BCASGERM Acornsoft German (Linkword) (SBX16)	11.05	12.71
BCASGRAM Acornsoft Graphics on the BBC	8.50	9.78
BCASGRAP Acornsoft Graphs & Charts (SBX02)	7.35	8.46
BCASHOOK Acornsoft Hooked on Numbers (SBX08)	7.35	8.46
BCASHOPP Acornsoft Hopper (SBG23)	7.35	8.46
BCASITAL Acornsoft Italian (Linkword) (SBX14)	11.05	12.71
BCASJARS Acornsoft Jars (SBG15)	8.83	10.16
BCASJCB0 Acornsoft JCB Digger (SBG09)	7.35	8.46
BCASKING Acornsoft Kingdom of Hamill (SBG25)	7.35	8.46
BCASLISD Acornsoft Lisp Demo Progs SBL09	7.35	8.46
BCASLISP Acornsoft Lisp Cassette (SBL02)	12.45	14.32
BCASMAGI Acornsoft Magic Garden (B) (SBX04)	7.35	8.46
BCASMAMB Acornsoft M/M ship Manager (SBX06)	7.35	8.46
BCASMETE Acornsoft Meteors (SBG13)	7.35	8.46
BCASMETM Acornsoft Meteor Mission (SBG29)	7.78	8.95
BCASMICR Acornsoft Microtext (SBL04)	36.85	42.37
BCASMISB Acornsoft Missile Base (SBG18)	7.35	8.46
BCASMISB Acornsoft Missing Signs (SBG09)	8.83	10.16
BCASMONS Acornsoft Monsters (SBG03)	7.35	8.46
BCASNUMB Acornsoft Number Balance (SBG08)	8.83	10.16
BCASONET Acornsoft/ICL One to Nine (SBX07)	7.35	8.46
BCASPEEK Acornsoft Peeko-Computer Pack (SBG02)	7.35	8.46
BCASPERP Acornsoft Personal Money M'ment SBG5	8.80	10.12
BCASPHIL Acornsoft Philosophers Quest (SBG01)	7.35	8.46
BCASPICT Acornsoft Picture Maker (SBX03)	7.35	8.46
BCASPLAN Acornsoft Planetoid (Defender) (SBG15)	7.35	8.46
BCASROCK Acornsoft Rocket Raid (SBG05)	7.35	8.46
BCASSENT Acornsoft Sentence Sequencing (SBG07)	8.83	10.16
BCASSINV Acornsoft Super Invaders (SBG16)	7.35	8.46
BCASSLID Acornsoft Sliding block puzzles (SBG12)	7.35	8.46
BCASSNAP Acornsoft Snapper (SBG04)	7.35	8.46
BCASSNOO Acornsoft Snooker (SBG21)	7.35	8.46
BCASSPAS Acornsoft Spanish (Linkword) (SBX15)	11.05	12.71
BCASSPAS Acornsoft S Pascal (SBL08)	12.45	14.32
BCASSPHI Acornsoft Sphinx Adventure (SBG07)	7.35	8.46
BCASSSTAR Acornsoft Starship Command (SBG22)	7.35	8.46
BCASTETR Acornsoft Tetrapod (SBG32)	7.78	8.95
BCASTREE Acornsoft Tree of Knowledge (SBG04)	7.35	8.46
BCASTURT Acornsoft Turtle Graphics (SBL07)	12.45	14.32
BCASTVIEW Acornsoft View Printer Driver (SBG04)	7.35	8.46
BCASVOLC Acornsoft Volcano (SBG33)	7.78	8.95
BCASWORH Acornsoft Word Hunt (SBG05)	8.83	10.16

BBC Cassette Software	Exc VAT	Inc VAT
BCASWORS Acornsoft Word Sequencing (SBG06)	8.83	10.16
BCBEAVM BES Animal Vegetable Mineral (XBE03)	6.63	7.62
BCBEHAPL BES Happy Letters (XBE07)	6.63	7.62
BCBEHAPN BES Happy Numbers (XBE04)	6.63	7.62
BCBEMAPR BES Map Rally (XBE08)	6.63	7.62
BCBETIM1 BES Timemán One (XBE05)	6.63	7.62
BCBETIM2 BES Timemán Two (XBE06)	6.63	7.62
BCBEWORH BES Wordhang (XBE02)	6.63	7.62
BCBEWORW BES World-Wise (XBE01)	6.63	7.62
BCBSBEYO BBCsoft Beyond Basic Cassette (B)	8.50	9.78
BCBSCANY BBCsoft Canyon (B)	7.39	8.50
BCBSCDRAW BBCsoft Drawing	7.40	8.50
BCBSCDRWH BBCsoft Dr Who: The First Adventure (B)	7.39	8.50
BCBSEARL BBCsoft Early Learning	7.40	8.50
BCBSFUNG BBCsoft Fun Games	7.40	8.50
BCBSGAMS BBCsoft Games of Strategy	7.40	8.50
BCBSHOMF BBCsoft Home Finance	7.40	8.50
BCBSMAKM Making the most of your Micro (Inc book)	9.57	11.01
BCBSMOTO BBCsoft Motorists' Log	7.40	8.50
BCBSPAIN BBCsoft Painting	7.40	8.50
BCBSPRO1 BBCsoft Programs Vol 1	7.40	8.50
BCBSPRO2 BBCsoft Programs Vol 2	7.40	8.50
BCBSRECO BBCsoft Record Keeper (B)	10.20	11.73
BCBSTAXC BBCsoft Taxcalc (B)	8.16	9.38
BCBSTOOL BBCsoft Tool Box (B)	15.52	17.85
BCBSVUTY BBCsoft VU Type (Typing Tutor) (B)	11.90	13.69
BCBSWHIT BBCsoft White Knight Mark Eleven (B)	8.50	9.78
BCBUTWIN Bug-Byte Twin Kingdom Valley	7.02	8.07
BCCCANDR Computer Concepts Android Attack	6.61	7.60
BCCCASTE Computer Concepts Asteroid belt	5.87	6.75
BCCCCHES Computer Concepts Chess	6.61	7.60
BCCCLOGO Computer Concepts Logo II	8.50	9.78
BCCCSNAK Computer Concepts Snake	5.87	6.75
BCCCSPAC Computer Concepts Spacehaws	5.87	6.75
BCCCSWAR Computer Concepts Swarm	6.61	7.60
BCDST747 Doctor Soft 747	5.87	6.75
BCGEBEEB Gemini Beebeak (Spreadsheet)	14.75	16.96
BCGEBEEP Gemini Beeplot (Graph Plot)	14.75	16.96
BCGECATE Gemini Caterpillar	7.35	8.46
BCGGFAIR Griffin & George Fairshare	7.35	8.46
BCGGGETS Griffin & George Getset	7.35	8.46
BCGGNUMB Griffin & George Numberfun	7.35	8.46
BCGGTABL Griffin & George Tablesums	7.35	8.46
BCGGWORG Griffin & George Wordgames	7.35	8.46
BCGGWORS Griffin & George Wordspell	7.35	8.46
BCGHTALP Good Houseke g Mr T's Alphabet	9.57	11.01
BCGHTMEA Good Houseke g Mr T's Measuring Games	9.57	11.01
BCGHTMON Good Houseke g Mr T's Money Box	9.57	11.01
BCGHTNUM Good Houseke g Mr T's Number Games	9.57	11.01
BCGHTSHA Good Houseke g Mr T's Shape Games	9.57	11.01
BCGHTTEL Good Houseke g Mr T Tells the Time	9.57	11.01
BCHOADVB Honeyfold Advanced basic course	12.10	12.71
BCHOBEGA Honeyfold Beginners Assembly Language	12.10	12.71
BCHOBEGB Honeyfold Beginners basic course	12.10	12.71
BCIBCRIM Ivan Berg Crime & Detection Quiz (XBX02)	9.35	10.75
BCIBDATI Ivan Berg Dating Game (XBX08)	9.35	10.75
BCIBHIST Ivan Berg History Quiz (XBX04)	9.35	10.75
BCIBID0 Ivan Berg 'I Do' (XBX07)	9.35	10.75
BCIBMUSI Ivan Berg Music Quiz (XBX03)	9.35	10.75
BCIRROYA Ivan Berg Royalty Quiz (XBX06)	9.35	10.75
BCISBSCIE Ivan Berg Science Fiction Quiz (XBX05)	9.35	10.75
BCIBTHER Ivan Berg Theatre Quiz (XBX01)	9.35	10.75
BCIJHYPE UK Hyperdrive for BBC (13)	4.80	5.52
BCIUNIVA UK Invaders for BBC (9)	5.54	6.37
BCIJSTRA UK Jstrabomber for BBC (14)	5.54	6.37
BCIJSUPE UK Super Hangman	3.32	3.82
BCJOYSTI Joystick Utility program for BBC	5.06	5.82
BCKAGALA Kansas Galactic Firebird	7.01	8.06
BCKOFREA Kosmos Software French Mistress A	7.35	8.46
BCKOFREB Kosmos Software French Mistress B	7.35	8.46
BCKOGERA Kosmos Software German Master A	7.35	8.46
BCKOGERB Kosmos Software German Master B	7.35	8.46
BCKOSPAA Kosmos Software Spanish Tutor A	7.35	8.46
BCKOSPAB Kosmos Software Spanish Tutor B	7.35	8.46
BCLCOLO Level 9 Colossal Adventure (B)	7.32	8.42
BCLDUNG0 Level 9 Dungeons (B)	7.32	8.42
BCLDUNG1 Level 9 Lords of Time (b)	7.32	8.42
BCLDUNG2 Level 9 Snowball (B)	7.32	8.42
BCMHOB0B Melbourne House Hobbit	11.05	12.71
BCMSQUIC Mirrosoft Quick Thinking (B)	5.13	5.90
BCOCMRWY Ocean Mr. Wimpy	5.10	5.87
BCPAFORT Pace Software Fortress	6.61	7.60
BCPPADVE Program Power Adventure (B)	5.87	6.75
BCPPALID Program Power Alien Destroyers (Invader)	5.91	6.79
BCPPALIE Program Power Alien Swirl	5.13	5.90
BCPPASTE Program Power Asteroid Storm (B)	5.87	6.75
BCPPASTR Program Power Astro Navigator	4.21	4.84
BCPPBAND Program Power Bandits at 3 o'clock	5.14	5.91
BCPPBARR Program Power Barrage for BBC	5.13	5.90
BCPPBEEB Program Power Beeb-Tote (B)	4.40	5.06
BCPPBUMB Program Power Bumble Bee (B)	5.87	6.75
BCPPCAVA Program Power Caveman Adventure	5.06	5.82
BCPPCHEM Program Power Chemistry	5.13	5.90
BCPPCHES Program Power Chess	5.87	6.75
BCPPCOSM Program Power Cosmic Combat (B)	5.14	5.91
BCPPCOWB Program Power Cowboy Shoot-out	5.06	5.82
BCPPCROA Program Power Croaker	5.87	6.75
BCPPCYBE Program Power Cybertron Mission (B)	5.87	6.75
BCPPDANG Program Power Danger! UXB	5.87	6.75
BCPPDEMD Program Power Demon Decorator (Painter)	5.06	5.82

## BUY 3 CASSETTES GET 1 FREE

BBC Cassette Software	Exc VAT	Inc VAT
BCPPDIS Program Power BBC Disassembler	5.06	5.82
BCPPDUNE Program Power Dune Rider	6.22	7.15
BCPPELDO Program Power Eldorado Gold (Adventure)	5.06	5.82
BCPPESCA Program Power Escape from M'base Alpha	5.87	6.75
BCPPFFAC Program Power Felix in the Factory	5.87	6.75
BCPPFFRU Program Power Felix & the Fruit Monsters	5.87	6.75
BCPPGHOU Program Power Ghoulies	5.87	6.75
BCPPHELL Program Power Hell Driver (B)	5.87	6.75
BCPPINTE Program Power Intergalactic Trader (B)	6.61	7.60
BCPPJETP Program Power Jet Power Jack	5.87	6.75
BCPPKILL Program Power Killer Gorilla	5.87	6.75
BCPPLABC Program Power Labyrinths of la Coshe	5.87	6.75
BCPPPLAS Program Power Laser Command	5.87	6.75
BCPPMART Program Power Martian Attack (B)	5.87	6.75
BCPPMINE Program Power Mine (B)	5.87	6.75
BCPPMOON Program Power Moon Raider	5.87	6.75
BCPPNEME Program Power Nemesis (B)	5.87	6.75
BCPPPHYS Program Power Physics	5.06	5.82
BCPPPOKE Program Power Poker Dice (B)	4.40	5.06
BCPPPOSJ Program Power Positron (B)	5.14	5.91
BCPPPREV Program Power Reversi (32K)	4.21	4.84
BCPPRROL Program Power Roulette for BBC	4.21	4.84
BCPPSEEK Program Power Seek	5.06	5.82
BCPPSPAC Program Power Space Jailer (B)	5.14	5.91
BCPPSTOC Program Power Stock Car (B)	6.56	7.45
BCPPSWOO Program Power Swoop	5.87	6.75
BCPPTIMT Program Power Timetrek (Star Trek Plus)	5.87	6.75
BCPPWALL Program Power Wall (B)	4.40	5.06
BCPPWHER Program Power Where (B)	5.14	5.91
BCPPWORM Program Power World Geography (B)	5.14	5.91
BCPPZARM Program Power Zarm (B)	5.87	6.75
BCPSALO Pasion Saloon Sally	4.39	5.05
BCPSBEEB Quicksilver Beeb Art (B)	11.05	12.71
BCPSMUSI Quicksilver Music Processor (B)	11.05	12.71
BCRHGALA RH Software Galactic Wipeout (RHS001C)	6.61	7.60
BCRHSKIS RH Software Ski Slalom (RHS002C)	6.61	7.60
BCSHGBLT Simon W Hessel GB Ltd	5.13	5.90
BCSHINTE Simon W Hessel Interference	4.39	5.05
BCSHTRAV Simon W Hessel Travel Game	4.39	5.05
BCSI3DBO Software Invasion 3D Bomb Alley	5.87	6.75
BCSIATTA Software Invasion Attack on Alphacenturi	5.87	6.75
BCSIEAGL Software Invasion Eagles Wing (B)	5.87	6.75
BCSIGUNS Software Invasion Guns smoke	5.87	6.75
BCSISPOO Software Invasion Spooks & Spiders (B)	5.87	6.75
BCSIVORT Software Invasion Vortex	5.87	6.75
BCSMDIS Simonsoft Disassembler	5.87	6.75
BCSS2002 Superior Software 2002	5.87	6.75
BCSSALID Superior Software Alien Dropout	5.87	6.75
BCSSBATT Superior Software Battle Tank	5.87	6.75
BCSSCENT Superior Software Centipede	5.87	6.75
BCSSCOLD Superior Software Colditz Adventure	5.53	6.35
BCSSCRAZ Superior Software Crazy Painter	5.87	6.75
BCSSFAIR Superior Software Fairground	5.87	6.75
BCSSFRUG Superior Software Frogger	5.87	6.75
BCSSFRUI Superior Software Fruit Machine	5.87	6.75
BCSSGALA Superior Software Galaxians	5.87	6.75
BCSSHUNC Superior Software Hungback	5.87	6.75
BCSSINVA Superior Software Invaders	5.87	6.75
BCSSOVER Superior Software Overdrive	5.87	6.75
BCSSPONT Superior Software Pontoon	5.13	5.90
BCSSOBER Superior Software Obert	5.87	6.75
BCSSROAD Superior Software Road Runner	5.87	6.75
BCSSSPAF Superior Software Space Fighter	5.87	6.75
BCSIJUNW Sulis Software Junior Wordspits	7.35	8.46
BCSIJUST Sulis Software Just a Mot	7.35	8.46
BCSUOPEN Sulis Software Open Sesame	7.35	8.46
BCSUSTAW Sulis Software Starter Wordspits	7.35	8.46
BCSUTENF Sulis Software Tense French	7.35	8.46
BCSUTIME Sulis Software Time Traveller	7.35	8.46
BCSUWORD Sulis Software Wordpower	7.35	8.46
BCVIDARE Visions Daredevil Dennis	7.00	8.05
BCVIDEMO Visions Demolator	7.00	8.05
BCVIPENG Visions Pengi	7.00	8.05
BCVISNOO Visions Snooker	6.61	7.60
BCVMDLT Voltmace Delta Driver Cassette	4.39	5.05

Prices subject to variation without prior notification

TRADE ENQUIRIES WELCOME  
Access & Barclaycard Accepted  
Large Stocks - 24 Hour Despatch  
Carriage 46p

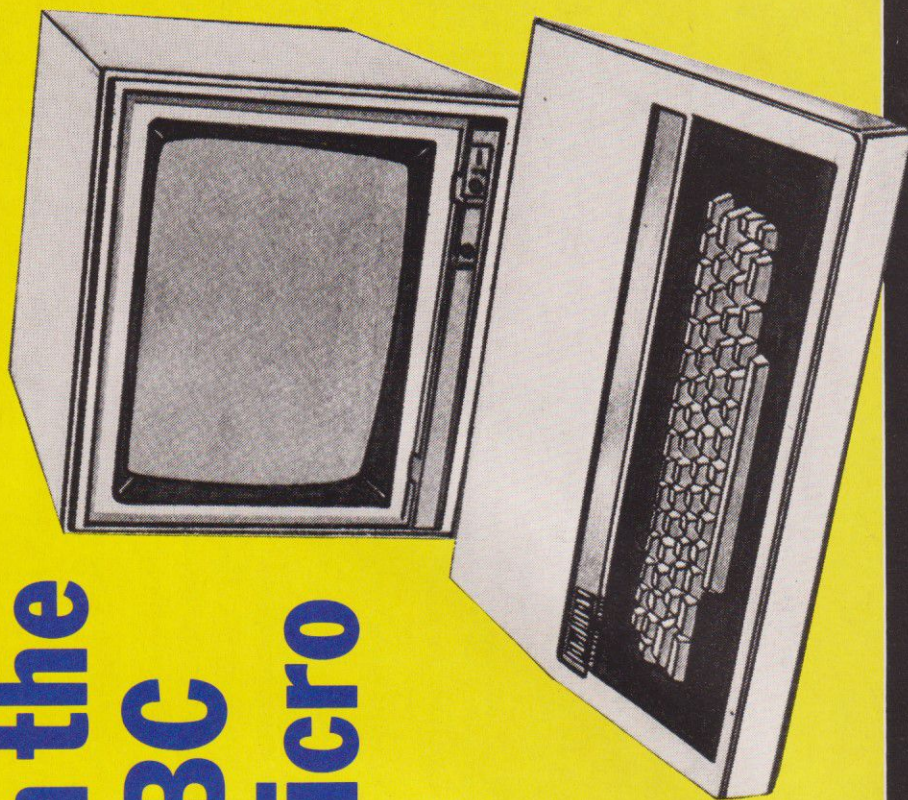
# BBC



# Electronequip



# A-Z guide to Games Software on the BBC Micro



Title	Supplier	Description
Trafalgar	Squirrel	Ships of the line in sea battle.
Transistors Revenge	Softspot	Arcade fun in the CPU.
Trench	Virgin	Fight to save the universe.
Twin Kingdom Valley	Bugbyte	Highly complex graphic adventure.
Valley of the Kings	M.P.	Adventure game.
Vortex	S. Invasion	3D voyage into outer space with aliens and asteroids.
Wallaby	Superior	Rescue your wallaby baby from the evil monkeys.
Wheel of Fortune	Epic	Graphic adventure. 250 locations and moving chars.
Wizard	Quicksilva	Use your magic to save the princess.
Woodland Terror	M.P.	Adventure game.
Zalaga	Aardvark	Arcade action.

## Suppliers of programs featured in this Guide

**A&F**, Unit 8, Canal Side Industrial Estate, Woodbine Street East, Rochdale; **Aardvark**, 100 Ardleigh Green Road, Hornchurch, Essex; **Acornsoft**, c/o Vector Marketing, Denington Estate, Wellingborough, Northants; **Alligata**, 1 Orange Street, Sheffield; **Amcom**, 92 New Cross Street, Bradford; **ASP**, 1 Golden Square, London W1R 3AB; **Atari**, Atari House, Railway Terrace, Slough, Berks; **Bridge**, 36 Fernwood, Marple Bridge, Stockport, Cheshire; **Bug-Byte**, Mulberry House, Canning Place, Liverpool; **Chalksoft**, Lowmoor Cottage, Tonedale, Wellingdon, Somerset; **Comsoft**, 7 Roman Drive, Leeds, West Yorkshire; **D. Kindersley**, 1-2 Henrietta Street, London; **Dial Software**, 72 Downend Road, Downend, Bristol; **Dynabyte**, PO Box 10, Barnsley, South Yorkshire; **Epic**, 10 Gladstone Street, Kibworth, Beauchamp, Leicester; **Ganymede**, Huntsmans Walk, Rugeley, Staffs; **H&H**, 53 Holloway, Runcorn, Cheshire; **IJK**, Unit 3C, Moorfields, Moorpark Avenue, Bispham, Blackpool, Lancs; **Ixon**, 10 The Crescent, Lymington, West-on-Super-Mare, Somerset; **Kay-Ess**, 11 Buttercup Close, Romleighs Park, Harold Wood, Essex; **Leisure Genius**, 3 Montagu Row, London W1H 1AB; **Level 9**, 229 Hughenden Road, High Wycombe, Bucks; **Melbourne**, 39 Milton Trading Estate, Abingdon, Oxon; **Microbyte**, 18 Hilgrove Road, Newquay, Cornwall; **Microdeal**, 41 Truro Road, St Austell, Cornwall; **Micro Power**, Northwood House, North Street, Leeds; **MP**, 165 Spital Road, Bromborough, Merseyside; **MRM**, 17 Cross Coates Road, Grimsby, South Humberside; **Peaksoft**, 48 Queen Street, Balderton, Newark, Notts; **Postern**, PO Box 2, Andoversford, Cheltenham, Glos; **Quicksilva**, Palmerston Park House, 13 Palmerston Road, Southampton; **Red Giant**, 3A Oakcroft Close, Pinner, Middlesex; **Salamander**, 27 Ditchling Rise, Brighton, Sussex; **Senator**, 44 Worcester Crescent, London NW7 4LL; **Serin**, PO Box 163, Slough; **Silverlind**, 156 Newton Road, Burton-On-Trent, Staffs; **Simonsoft**, 25 Tatham Road, Abingdon, Oxon; **Softspot**, 29 South Crescent, Prittlewell, Southend, Essex; **Software Comm**, Martech House, Bay Terrace, Pevensey Bay, East Sussex; **Software Invasion**, 50 Elborough Street, Southfields, London; **Squirrelsoft**, 4 Bindloss Avenue, Eccles, Manchester; **Starcade**, 2 Elworthy Avenue, Halewood Village, L26 7AA; **Superior**, Regent House, Skinner Lane, Leeds; **Virgin**, 2-4 Vernon Yard, Portobello Road, London W11 2DX; **Visions**, 1 Felgate Mews, Studland Street, London W6 9JF; **Watford Electronics**, 250 High Street, Watford, Herts.



Title	Supplier	Description
Adventure Quest	Level 9	Pure text adventure. 200+ locations. Overcome umpteenth obstacles guarding the Dark Tower in Middle Earth.
Air Traffic Control	Microdeal	Simulated Air Traffic Control situation where the user is responsible for take off departure of remotely controlled vehicles.
Alien Dropout	Superior Software	Arcade style shooting game where killer moths are invading earth.
Arena 3000	Microdeal	Fifteen waves of robots with which to do battle.
Artist	MRM	Computer aided design package which will enable you to create your own pictures.
Atlantis	IJK	This includes normal Atlantis/Scramble features include depth charges, rockets, jelly fish etc.
Attack on Alpha Centauri	Software Invasion	A 3D battle for supremacy over swarms of deadly, screaming, sting dropping bug-eyed wasps.
Aviator	Acornsoft	Flight simulator.
Banana Man	MRM	Your mission is to eat 40 banana sarnies. Watch the spiders!
Base 10	Dial	How fast can you add to 10, to receive your bonus star and improve your score.
Battletank	Superior Software	A 3D tank battle game.
Beat the Bug	Bridge	A new virus is decimating mankind. You must deduce the structure of its molecule.
Beebmunch	IJK	Another version of Pacman, including ghosts, fruits, super points, screams, etc.
Beep-Beep	IJK	A version of Simon. Includes choice of number of colours and sounds.
Blitzkrieg	Software Invastor.	3D tank battle with revolving, exploding tanks, 3D shell fire, etc.
Blue Dragon	MP	Adventure game.
Boris in the Underworld	Superior Software	Help Boris recover his possessions from the underworld creatures.
Brainstorm	Virgin	Unique game designed to test skill, logic, memory and nerve of two players.
Brain Teasers	Dynabyte	Six games of logic including Reversi, 3D oxa, Forty Two, Mastermind etc.
Bridgeman	Bridge	Bridge Software's version of the popular gobbler game.
Bridgemaster	Serim	Bridge tutor with world authority Terence Reese. Contains two commentary tapes, two computer tapes plus book.
Bridge to the East	Ixon	Multi screen arcade-adventure.
Bunfun	Squirrel	Frenetic assembly line happenings in a bun factory. Icing and nuts everywhere. 1-2 younger players.
Bug Bomb	Virgin	Fight off the aliens by throwing bombs at them.
Casino	IJK	Version of two popular casino games, full rules.
Castaway	Simon	A graphic adventure. Battle against giant spiders, irate chemistry teachers, etc.
Castle Assault	MRM	Scale the heavily defended castle walls to obtain your golden reward.
Castle Frankenstein	Epic	M/C adventure with over 250 locations. Find and destroy the evil Frankenstein monster.
Castle of Riddles	Acornsoft	Text adventure.

Title	Supplier	Description
Snakepit	Postern	Gobble 'em up type arcade game.
Snowball	Level9	Pure text adventure.
Space Fighter	Superior	Pilot the fighter craft against the aliens.
Space Adventure	Virgin	Graphics adventure.
Star Hawks	Kay-Ess	Hold off the hovering mutant firing hawks.
Space Invaders	Bug Byte	Arcade classic.
Space Shuttle	Microdeal	Launch, park, retrieve and land, simulation.
Space Tank	Kay-Ess	Sideways scroll arcade game.
Traffic Controller	Kay-Ess	Control landing of up to 20 robot spacecraft.
Spitfire Command	Superior	Fight the battle against the Luftwaffe.
Spooks and Spiders	S. Invasion	Using stairs and platforms rescue damsel - avoiding deadly spiders.
Staircase Stampede	Comsoft	Paint the stairs, diffuse the bomb and avoid the stampede.
Star Battle	Superior	Two-player deep space dog fight.
Star Maze	S. Invasion	Discover rare jewels, dock with mothership, avoid aliens.
Star Striker	Superior	Control a three stage rocket and fight the aliens.
Star Trek	Superior	Space adventure.
Star Trek	IJK	Space adventure.
Stranded	Superior	Graphic adventure.
Stratobomber	IJK	Keep the enemy at bay and destroy the reactor.
Stockcar	Micropower	Car driving fun on the track.
Superfruit	Simonsoft	Ultimate fruit machine program.
Super Golf	Squirrel	18 hole simulation cross section view.
Superpool	S. Invasion	Real time graphic simulation.
Survivor	M.P.	Adventure game.
The Avaunting	Ixon	Adventure game with 13 magic segments.
The Cliche	Ixon	Ludicrous send up of the computer industry.
The Fallen Eagle	M.P.	Adventure game.
The Greedy Dwarf	D. Kindersley	Text adventure.
3D Maze	IJK	Battle against the clock to escape.
3D Bomb Alley	S. Invasion	Air/sea battle in 3D simulation.
3D Munchy	MRM	Pacman type game with the ghosts digging holes.
3D Space Ranger	Microbyte	Action game with on-screen ship, asteroids and trench.
3D Tank Zone	Dynabyte	Wire graphics tank battle.
3 Deep Space	Postern	3D space action with special glasses.
Theseus	Silverlind	A million mappable 3D mazes with treasure and monsters.
2002	Superior	3D flight simulation docking to space station.



Title	Supplier	Description
Ossie	Peaksoft	Machine code arcade game.
Overdrive	Superior Software	3D car race simulation.
'Owzat	Virgin	Fight for the ashes in your own front room.
Peg Leg	IJK	Collect gold bars while fighting off pirates.
Pentiles	Silverlind	Cover a target area with a random collection of shapes.
Pengwyn	Postern	Strategy and arcade action combined.
Percy Penguin	Superior Software	Save Percee from the Snobeels on the antarctic glaciers.
Pinball	MByte	Machine code simulation for up to four players.
Plank Walk	Virgin	Help Scaffolding Sid survive berserk planks.
Pole Position	Atari	3D, full colour Formula One racing car game.
Pontoon	Superior Software	Full feature version of the ever popular card game.
Pontoon & Patience	IJK	Versions of the two popular card games.
Pool	Dynabyte	Two player game using high-res graphics for accuracy and smoothness.
Proctector	Quicksilver	Protect the pods in outer space action.
Q-Man	MRM	Highly entertaining 3D pyramid game.
Q-Mans Brother	MRM	The follow up to Q-Man. Great 3D graphics.
Quest for the Holy Grail	Epic	Find and return the Holy Grail to Camelot in this machine code adventure.
Renegade Robots	Senator Software	Skillfully reassemble three cubes in a maze, avoiding the enemy.
Return to Eden	Level 9	Explore a totally alien planet in this sequel to Snowball.
Reversi	Superior Software	Board game simulation with full instructions.
Reversi	MByte	You play the micro in this machine code simulation.
Robotron 2084	Atari	Stop the robotrons and save the last humans.
Rocky	Superior Software	Drop rocks on monsters in this all action game.
Road Racer	Superior Software	A maze chase game with cars, radar and smokescreens.
Rubble Trouble	Micropower	Arcade rock pushing fun.
Sadim Castle	MP	Adventure game.
SAS Commander	Comsoft	Free the hostages and kill the terrorists.
Savage Pond	Starcade	Survive in the perils of the deep.
Scrabble 1eisure	Genius	A game of strategy involving words.
Screwball	MRM	The human corkscrew changes his surrounds avoiding the deadly black bugs.
Sea Lord	Bug Byte	Undersea fun.
Serpents Lair	Comsoft	Children's graphic adventure.
Secret Sam 1	MRM	Spy-based text adventure.
Secret Sam 2	MRM	Spy-based text adventure – more difficult.
Smash and Grab	Superior	A robber snatching gold with a PC in hot pursuit.

Title	Supplier	Description
Caterpillar	IJK	Rendition of the popular arcade game, features toadstools, spiders, fleas, etc.
Centibug	Superior Software	A game where insects attack you through an ever increasing number of mushrooms.
Champions	Peaksoft	Football management – Take your club from the 4th Division to the European Cup.
Checkout	Virgin	Change colour of squares on a grid by "running" over them twice.
Chieftain	Virgin	It's a fight to the finish in your armoured tank – fight the computer or a friend.
City Defence	Bug Byte	Need we say more.
Classic Arcade Games	Ganymede	Four classic arcade games, with full colour and sound and many extra facilities. Joystick or K/B.
Colditz Adventure	Superior Software	Taunting adventure game in the perilous setting of Colditz Castle.
Colossal Adventure	Level 9	Pure text adventure. 200+ locations full scale version of the original mainframe game, "Adventure".
Community	Ixion	Strategic game where you must care for a farm, beware the famines and bank manager.
Conflict	Software Comm	Computer moderated strategy board game for two players.
Corporate Climber	Dynabyte	A race against time, the taxmen and heart failure in this arcade style game.
Cosmic Kidnap	Superior Software	Prevent the aliens from effecting the release of the captives in this fray.
Crawler	Watford Electronics	A version of the arcade game.
Crazy Painter	Superior Software	Help the monkey paint the squares before the hungry tribesmen catch him.
Cribbage	Superior Software	A version of the card game played against the computer.
Crown of Mardan	MP	Adventure game.
Cruncher	Virgin	Trample the time bombs but avoid the boots or be crunched.
Custard Pie Fight	Comsoft	Knock your opponent over with a pie. A hilarious game for one or two players.
Cybertron Mission	Micro Power	Version of the Atari game Shamus. Walk through a maze filled with robots, collecting objects.
Cylon Attack	A&F	3D space shoot out.
Dambusters	Alligator	Destroy the dam and avoid the flask.
Darts	MRM	Step up to the ocky to enjoy 501, Round the Clock and Cricket.
Death's Head Hole	Peaksoft	Simulation of pothole rescue call-out.
Diamond Mine	MRM	A game in which you have to guide a pipe to diamonds.
Digger	Visions	Allotment arcade capers.
Donkey Kong Jr	Atari	Help DJK rescue papa from Mario.
Draughts	Superior Software	An excellent version of the board game with some nice additions.
Draughts	Ganymede	Play the computer – no death or destruction here!
Dungeon Adventure	Level 9	Pure-text adventure. 200+ locations. Magical treasures abound in the caves of Middle Earth.



Title	Supplier	Description
Eagle's Wing	Software Invasion	Fly through a heavily guarded canyon, avoid missiles, refuel and be ready for the next confrontation.
Elite	Acomsoft	Space strategy and shooting.
Er*bert	M Byte	Help Erbert change the colour of the cubes, but avoiding unwanted guests.
Escape from Moonbase Alpha	Program Power	Graphic adventure.
Fairground	Superior Software	A simulation of a rifle range at a fun fair.
Fall of Rome	ASP	Historical strategy game.
Fantasy Adventure	Dial	Overcome the puzzles and collect to fulfil your quest.
Frirenwood	MP	Adventure game.
Five A Side Socca	IJK	Two player m/c version. Joystick or keyboard.
Flags	IJK	The flags of the world are drawn in hi-res colour. Test's knowledge of flags and geography.
Flints Gold	Micrograf	Adventure game.
Frak	Aardvark	Ladders and levels game.
Franklyn's Tomb	Salamander	Adventure.
Froggy	Superior Software	You must guide Froggy across the roads and rivers to reach his home in time.
Fortress	Amcom	3D shoot out.
Fruit Machine	Superior Software	A fruit machine simulation with nudges, gambling, etc.
Fruity Freddy	Softspot	Arcade action in Mr Meany's garden.
Galaxy Birds	Superior Software	Many formations of Galaxy Birds intent on your destruction are the enemy in this game.
Gatecrasher	Quicksilver	A game of skill and strategy that will test your mind to the limits. Plus you have the opportunity to win 200.
Ghouls	Program Power	Haunted house fun.
Gideon's Gamble	Superior Software	Assist Gideon to find the treasure in this complex adventure with a nautical flavour.
Gisburne's Castle	Software Comm	Graphic arcade adventure with 380 different locations.
Gnasher	Superior Software	A maze chase game including power pills, ghosts and fruits.
Gunsmoke	Software Invasion	A realistic 3D Wild West gun battle.
Guy in the Hat	MRM	A game in which you have to eat "pickles" without getting eaten yourself.
Heist	Softspot	Levels and ladders in the bank.
Hobbit	Melbourne House	The well-known adventure game.
Honeybug	Silverind	Fill the hive with honey and larvae and make the colony swarm.
Horseace	Dynabyte	Betting game for one to six players.
Horses	Kay-Ess	Choice of three arenas and six horses.
Hunchback	Superior Software	A 12 screen version of the arcade game.
Hyperdrive	IJK	Guide your laser tank around the network of passages destroying the drone aliens.

Title	Supplier	Description
Inkosi	Chalksoft	A colourful and addictive simulation with graphics and sound, ages 10 to adult.
Innerzone Shift	Ixon	Defuse enemy bombs while dodging their patrols.
Invaders	IJK	The classic game.
Invaders	Superior Software	Another version of space invaders.
Jungle Jive	Virgin	Wild animals attack you as you stroll through the jungle.
Kensington	Leisure Genius	A game of strategy involving shapes.
Killer Gorilla	Micro Power	A version of the arcade game Donkey Kong.
Kingdom of Klein	Epic	Defeat the wicked witch in this 230+ roomed adventure.
Ladder Maze	Superior Software	Find your way to the matter transmitter in this 3D maze game.
Landfall	Virgin	A space ship landing simulator.
Leap Frog	IJK	Help Froggy to get home.
Lemming Syndrome	Dynabyte	Bounce the people to safety.
Looney Lift	H&H	Operate the lift to catch rushing guests, VIPs and luggage.
Lords of Time	Level 9	Text adventure with over 200 locations.
Lost City	Superior Software	Venture into the unknown to discover the hidden mysteries of the Lost City.
Lost in Space	Salamander	Adventure.
Mazog	Red Giant	Can you reach the other side of the maze before the Mazog gets you?
Mekon Raiders	Micro Byte	3D space fight simulation.
Microbe	Virgin	Graphically stunning all-action arcade game.
Mined-Out	Quicksilver	Rescue Bill the worm from certain death. A strategy game.
MS Pac-man	Atari	Guide our heroine around tricky mazes...
Missile Base	Acomsoft	Defend your cities from the enemy.
Missile Strike	Superior Software	Defend your cities from relentless enemy missiles.
Moon Mission	Superior Software	Avoid the hazards and rescue stranded astronauts.
Mr. Wiz	Superior Software	Guide Mr. Wiz around the maze collecting fruit, avoiding gremlins.
Multi-Family Games	IJK	A games compendium. Hangman, Dice, Beetle and more.
Mutant Invaders	IJK	A version of the arcade classic.
Napoleon	Minerol	A war game where you become Bonaparte.
Neutron	Superior Software	Ride your light cycle skillfully to trap your opponent.
Nightmare Maze	MRM	Help Sleepy Joe escape the evils in his multi-screen nightmare.
Noc-A-Bloc	Virgin	Just when you thought it was safe to go back into the deep freeze.
Numberhang	Dialsoft	
Oblivion	Bug Byte	Addictive arcade action.
OG the Caveman	Simonsoft	Caveman smashes dinosaur eggs.
Old Father Time	Bug Byte	A traditional style text adventure.



# SPLAT!

By  
**MARTIN HOLLIS**

**SPLAT** is a simple cat and mouse maze game for any age.

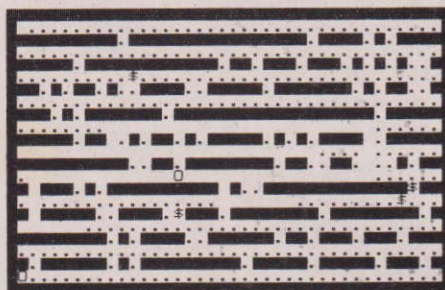
You are the star in the centre of the screen being chased by a monster – 'O' – who starts off in the top left hand corner. You have to eat all the dots, worth one point each, without being caught.

Every 100 dots you eat gives you four \$ signs to eat, each worth 10 points. When your score reaches 500 another 'O' – which has been waiting in the bottom left hand corner – springs to life.

These monster 'O's do not eat the dots, but they do eat the \$ signs – and you! If the two monsters hit each other, one is transported to a random position in the maze.

You have three lives to clear the screen of dots. When you lose a life you re-start from where you died.

At the beginning of the game you are asked to enter your name and to choose



a level. The higher the level, the more holes there are in the walls of the maze.

At the bottom of the screen your name, score, lives left and the hi-score are displayed.

If you qualify, your name will be entered automatically in the hi-score table displayed at the end of the game.

To change the player's name, press Escape which returns you to the beginning of the game without losing the hi-score table.

```

10 REM *****
20 REM *
30 REM * SPLAT! A maze game *
40 REM *
50 REM * By Martin Hollis *
60 REM * Design - J. Barton *
70 REM * (c)1984 Micro user *
80 REM *
90 REM *****
100 ONERROR PROCMORE:GOTO200
110 PROCTITLE
120 PRINT""SPC(12);"By Martin Hollis"
130 PRINT""SPC(9);"Designed by John Barton"
140 PRINT""SPC(14);"Press SPACE"
150 REPEAT:G=GET:UNTILG=32
160 REPEAT
170 PROCTITLE
180 INPUT" What is your name",NAME$
190 UNTILLEN(NAME$)<21
200 DIMP$(39,22),HZ$(9),H$(9)
210 FORAX=0TO9:HZ$(AX)=50+AX*50:H$(A
220 X)=CHR$(129+AXMOD4)+"Micro User.":NEXT
230 REPEAT
240 MODE7:VDU23;8202;0;0;0
250 PROCINST
260 REPEAT
270 PROCTITLE
280 INPUT" Enter level 1-8 (1 is ea
290 sy, 8 is hard)"" then press RETURN"
300 SPC(1)HZ
310 UNTILHZ>0ANDHZ<9:HZ=(HZ-1)*5
320 FORBX=0TO39:FORAX=0TO21STEP2
330 P$(BX,AX)=1:P$(BX,AX+1)=2:NEXT:
340 NEXT
350 FORAX=1TO38:P$(AX,22)=1:NEXT
360 FORAX=0TO22:P$(0,AX)=1:P$(39,AX
370 )=1:NEXT
380 FORAX=2TO20STEP2:FORBX=0TORND(H
390 X)+2:P$(RND(37)+1,AX)=2:NEXT:NEXT
400 MAX=100
410 LEFTX=0
420 CLS
430 FORBX=0TO22:FORAX=0TO39
440 IFP$(AX,BX)=0 PRINT" "; ELSEIFP
450 $(AX,BX)=1 PRINTCHR$(255); ELSEIFP$(A
460 X,BX)=2 PRINT". ";:LEFTX=LEFTX+1
470 NEXT:NEXT
480 XX=20:YY=11:CX=1:DX=1:EX=1:FZ=2
490 1:SZ=0:LX=3:P$(CX,DX)=0:P$(EX,FZ)=0:L
500 EFTX=LEFTX-2:REPEAT
510 YZ=11
520 CX=1
530 EX=1
540 FZ=21
550 SZ=0
560 LX=3
570 P$(CX,DX)=0
580 P$(EX,FZ)=0
590 LEFTX=LEFTX-2
600 REPEAT
610 FORAX=0TO1

```

## PROCEDURES

**PROCSCREEN**  
**PROCDOAR**  
**PROCSPIRINKLE**  
**PROCSCORES**  
**PROCORE**  
**PROCTITLE**  
**PROCINST**

Prints screen.  
Places \$ on screen at every 100 score.  
Gives new dots when screen cleared.  
Hiscore table.  
Another go option.  
Prints header title.  
Prints instruction.

## VARIABLES

**C%,E%** x position of monsters.  
**D%,F%** y position of monsters.  
**H%** Level of difficulty.  
**H\$(9)** High score.  
**L%** Lives.  
**LEFT%** Number of dots remaining on screen.  
**P% (40,24)** Dimension for easy access to chrs on screen.  
**S%** Score.  
**X%,Y%** Coordinates of star.  
**HS(9)** Table of names.

## Keys

A up  
Z down  
< left  
> right



## From Page 45

```

500 PRINTTAB(XZ,YZ); " "
510 IFPZ(XZ,YZ)=2 SOUND1,-15,100,1:
SZ=SZ+1:PZ(XZ,YZ)=0:LEFTZ=LEFTZ-1
520 IFPZ(XZ,YZ)=3 SZ=SZ+10:FORBZ=0T
0255STEP2:SOUND1,-15,BZ,0:NEXT:PZ(XZ,
YZ)=0
530 IFINKEY(-66)ANDPZ(XZ,YZ-1)<>1 Y
Z=YZ-1ELSEIFINKEY(-98)ANDPZ(XZ,YZ+1)<
>1 YZ=YZ+1
540 IFINKEY(-103)ANDPZ(XZ-1,YZ)<>1
XZ=XZ-1ELSEIFINKEY(-104)ANDPZ(XZ+1,YZ
)<>1 XZ=XZ+1
550 FORA=0T0100:NEXT
560 MZ=0
570 PRINTTAB(XZ,YZ); "*"
580 NEXT
590 IFPZ(CZ,DZ)=2 PRINTTAB(CZ,DZ); "
." ELSE PRINTTAB(CZ,DZ); " ":PZ(CZ,DZ)
=0
600 IFDZ<YZANDPZ(CZ,DZ+1)<>1 DZ=DZ+
1:MZ=-1
610 IFDZ>YZANDPZ(CZ,DZ-1)<>1 DZ=DZ-
1:MZ=-1
620 IFCZ>XZANDPZ(CZ-1,DZ)<>1ANDMZ=0
CZ=CZ-1 ELSEIFCZ<XZANDPZ(CZ+1,DZ)<>1
ANDMZ=0 CZ=CZ+1
630 IFSZ<501 80T0670
640 MZ=0:IFPZ(EX,FZ)=2 PRINTTAB(EX,
FZ); "." ELSE PRINTTAB(EX,FZ); " ":PZ(C
Z,DZ)=0
650 IFFZ<YZANDPZ(EX,FZ+1)<>1 FZ=FZ+
1:MZ=-1 ELSEIFFZ>YZANDPZ(EX,FZ-1)<>1
FZ=FZ-1:MZ=-1
660 IFEX>XZANDPZ(EX-1,FZ)<>1ANDNOTM
Z EX=EX-1 ELSEIFEX<XZANDPZ(EX+1,FZ)<>
1ANDNOTMZ EX=EX+1
670 IFLEFTZ<8 PROCSPRINKLE
680 IFCZ=EXANDDZ=FZ REPEAT:EX=RND(3
8)+1:FZ=RND(20)+1:UNTILPZ(EX,FZ)<>1
690 PRINTTAB(CZ,DZ); "O";TAB(EX,FZ);
"O"
700 PRINTTAB(0,23); "SCORE=";SZ;TAB(
12,23); "LIVES=";LZ;TAB(23,23); "HIGH="
;HZ(8);TAB(13,24); "Player=";NAME$;
710 IFSZ>MAX MAX=MAX+100:FORBZ=0T03
:PROCDOLLAR:NEXT
720 IFCZ=XZANDDZ=YZOREZ=XZANDFZ=YZ
LZ=LZ-1:FORAZ=0T080:PRINTTAB(XZ,YZ);C
HR$(RND(10)+64):SOUND1,-15,RND(255),0
:NEXT:SOUND0,-15,4,20:CZ=1:DZ=1:EX=1:
FZ=21:XZ=10:YZ=11:PROCSCREEN
730 FORA=0T0100:NEXT:UNTILLZ=0
740 PROCSCORES:GOTO200
750 DEFPROCSCREEN
760 CLS
770 LEFTZ=0
780 FORBZ=0T022:FORAZ=0T039
790 IFPZ(AZ,BZ)=0 VDU32: ELSEIFPZ(A
Z,BZ)=1 PRINTCHR$(255);

```

```

800 IFPZ(AZ,BZ)=2 PRINT". ";LEFTZ=L
EFTZ+1
810 IFPZ(AZ,BZ)=3 PRINT"$ ";
820 NEXT:NEXT:ENDPROC
830 DEFPROCDOOLLAR
840 IZ=RND(38)
850 JZ=RND(21)
860 PRINTTAB(IZ,JZ); "$"
870 IFPZ(IZ,JZ)=2 LEFTZ=LEFTZ-1
880 PZ(IZ,JZ)=3
890 FORKZ=0T0255:SOUND1,-15,KZ,0:NE
XT
900 ENDPROC
910 DEFPROCSPRINKLE
920 HZ=HZ+1
930 PROCTITLE
940 PRINT" Well done! You deserve a
sprinkling of""SPC(13);CHR$129;CH
R$136"DOTS!!!!""
950 LEFTZ=0
960 FORBZ=0T039:FORAZ=0T021STEP2
970 PZ(BZ,AZ)=1
980 PZ(BZ,AZ+1)=2
990 NEXT:NEXT
1000 FORAZ=1T038
1010 PZ(AZ,22)=1
1020 NEXT
1030 FORAZ=0T022
1040 PZ(0,AZ)=1
1050 PZ(39,AZ)=1
1060 NEXT
1070 FORAZ=2T020STEP2:FORBZ=0T0HZ+2
1080 PZ(RND(37)+1,AZ)=2
1090 NEXT:NEXT
1100 PROCSCREEN
1110 XZ=20:YZ=11
1120 CZ=1:DZ=1
1130 EZ=1:FZ=21
1140 ENDPROC
1150 DEFPROCSCORES:IFSZ<=HZ(0) 80T01
290
1160 *FX15
1170 PROCTITLE
1180 AZ=0
1190 REPEAT
1200 AZ=AZ+1
1210 UNTILSZ<HZ(AZ)ORAZ=9
1220 FORBZ=0T0AZ-2
1230 HZ(BZ)=HZ(BZ+1):H$(BZ)=H$(BZ+1)
1240 NEXT
1250 HZ(AZ-1)=SZ
1260 PRINT" Congratulations ";NAME$
" You are in the high score table"
1270 H$(AZ-1)=CHR$136+NAME$
1280 PROCKEY
1290 PROCTITLE
1300 PRINTSPC(10);CHR$(134);" High
Score Table"
1310 FORAZ=8T00STEP-1
1320 PRINTSPC(8);H$(AZ);STRING$(25-(
LEN(H$(AZ))+LEN(STR$(HZ(AZ)))),".");H
Z(AZ)'

```

```

1330 IFLEFT$(H$(AZ),1)=CHR$136 H$(AZ
)=CHR$(128+RND(4))+MID$(H$(AZ),2)
1340 NEXT
1350 PROCKEY
1360 ENDPROC
1370 DEFPROCSCORE:PROCTITLE:PRINT" An
y more Y/N":REPEAT:G$=GET$:UNTILG$="Y
"ORG$="N":IFG$="N" PROCTITLE:END
1380 PRINT" What is your name?":INPU
TSPC(1)NAME$:ENDPROC
1390 DEFPROCTITLE
1400 CLS
1410 PRINTCHR$(130);CHR$(157);CHR$(1
32);CHR$(141);" S P L A
T !";CHR$(130);CHR$(157);CHR$(132);
CHR$(141);" S P L A T !
"
1420 ENDPROC
1430 DEFPROCKEY
1440 PRINTTAB(14);"Press SPACE";
1450 *FX15
1460 REPEAT:G$=GET$:UNTILG$=32
1470 ENDPROC
1480 DEFPROCINST
1490 CLS
1500 PROCTITLE
1510 PRINT" Instructions Y/N"" ";
1520 REPEAT:G$=GET$:UNTILG$="Y"ORG$=
"N"
1530 IFG$="N" ENDPROC
1540 PROCTITLE
1550 PRINT"You start off as the ""*""
" in the middle""of the screen.You m
ust eat up all of the little dots as y
ou get 1 added to your""score for ev
ery dot eaten. Every time""your scor
e reaches 100, 200, 300 etc."
1560 PRINT"you get four '$' signs wh
ich are each""worth 10 points when e
aten. If you eat""up all the dots yo
u get more dots and a new maze."
1570 PRINT"There is one 'O' chasing
you round the""maze, and another re
ady to be activated when your score r
eaches 500. If the two 'Os' meet one
of them is sent to another part of the
maze."
1580 PRINT"You also get a personali
sed High Score""Table for nine score
s. If you qualify""your name will be
entered automatically."
1590 PROCKEY
1600 CLS
1610 PROCTITLE
1620 PRINTTAB(7);CHR$(136);CHR$(13
4);"Keys for the GAME.""TAB(11);CHR$(
131);"A = Up.""TAB(11);CHR$(132
);"Z = Down.""TAB(11);CHR$(133);
"< = Left.""TAB(11);CHR$(130);">
= Right.""
1630 PROCKEY
1640 ENDPROC

```



# FAIR

# PLAY

By  
**D.H. IBBOTSON**

THE fairground has come to your micro in the form of Happy Harry's shooting gallery. All you have to do is hit the targets with the darts in order to win one of Harry's fantastic prizes. (If the prizes are not quite up to your expectations then simply change the data on line 160.)

Hitting a number will increase your darts. Hitting a face will make all the targets retreat. Hitting anything else will increase your score.

Clear all the targets and a new set will appear in new colours, sometimes making the targets or gun partly invisible. So try to hit the numbers first, followed by everything except the faces. Shoot the faces last of all in order to progress to the next card.

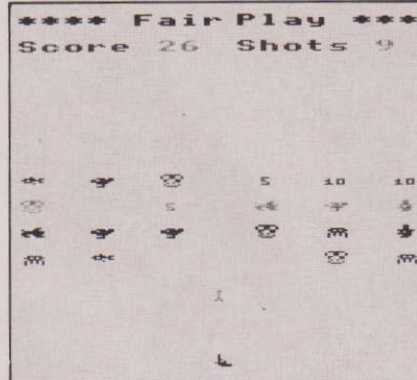
Instructions are contained in the program. Only the space bar is strictly necessary in order to play the game. I

hope you enjoy the Fairground organ music.

The data on line 460 may be changed if the score, or the number of extra darts, is not to your liking.

The formula is simple, 0 to 9 is added to the score, 10 or greater will be added to the darts, and -2 (use -2 only) will force a retreat.

Likewise characters 224 to 232 can be redefined if you dislike the targets.



## ARRAYS

- AS( ) Targets.
- CS( ) Colours in the form of strings.
- DS(1) Dart / DS(0) delete dart.
- prize\$( ) All the prizes.
- R%( ) Temporary stores for random numbers.
- S%( ) Scores for the various targets.
- s%( ) Old scores for score ladder.
- s\$( ) Old names for score ladder.

## VARIABLES

- A% Permanently set 165.
- a% Permanently set to OSBYTE address &FFF4.
- AS Removes bottom line of targets when they retreat.
- B% X axis position of the gun.
- C% Master counter.
- D% Darts flag 1 if fired, 0 if not fired.
- G\$ The gun.
- H% Height of targets.
- I% Jump out of loop flag.
- J% Counts hits.
- L% Level or card.
- M% Time value for each note.
- N% Pitch value for each note.
- O% Removes bottom line of targets if set flag.
- S% Total shots remaining.
- s% Old total shots remaining.
- T% Total scored.
- t% Old total scored.
- X% Darts X axis.
- Y% Darts Y axis.

## PROCEDURES

- PROCmulti\_colour** Displays Mode 6 (normally a single colour mode) in multi-colour by using \*FX19 and then rapidly changing the palette in order to display bands of colour.
- PROCsetup\_characters** Sets the string array AS( ) to random characters or targets.
- PROCcheck (A%)** Finds out which target (A%) has been hit and then pays out accordingly.
- PROCchange FNp (AS)** Selects fresh colour from the palette. PEEKs the screen more or less as described on page 432 of the User Guide. If a hit is detected removes it from the appropriate string, then calls PROCchange.
- FNs** Gets the name of the player for use in PROCscore\_ladder.
- FNT** Resets data pointer to line 1180 if the end of the music, as indicated by M%=0, has been reached.

```

10 REM **** Fair play ****
20 REM (c) The Micro User
30 REM Program by D.H.Ibbotson.
40 REM Music by T.Howes.
50 ENVELOPE1,4,90,-15,-15,10,20,20,
126,0,0,-126,126,126
60 ENVELOPE2,4,-1,1,0,20,20,0,126,
0,0,-126,126,126
70 ENVELOPE3,3,0,0,0,0,0,0,121,-10
,-5,-2,-120,-120
80 ENVELOPE4,0,0,0,0,0,0,0,126,-4,
-3,-3,126,100
90 ENVELOPE5,10,1,-2,1,10,10,10,70
,-10,-8,-5,-50,0
100 DIMA$(3),C$(3),D$(1),R%(3),S%(7)
)
110 REM arrays for score ladder
120 DIMs$(8),s$(8):FORa%=0TO8:s$(a%)
)=0:s$(a%)="Happy Harry ":NEXT
130 REM arrays for prizes
140 DIMprize$(7)
150 RESTORE:FORa%=0TO7:READprize$(a)
:NEXT

160 DATA "a plastic ring","a cardboard
doll's house","a fish in a bag","a
coconut","a rag doll","a plaster duc
k","a toy whistle","any prize on the
stall"
170 MODE6
180 VDU23:8202:0:0:0;
190 PRINTTAB(9,2)"Welcome to the MI
CRO fair"
200 PRINTSPC2"try your luck at the
shooting gallery"
210 PRINTSPC11"A prize every time !
!"
220 PRINTSPC2"everyone wins on Hap
py Harry's pitch"
230 PRINTSPC13"5 darts per go"
240 PRINTSPC14"PRESS ANY KEY"
250 PROCmulti_colour:FORa%=0TO1000:N
EXT
260 PRINTTAB(14,2)"INSTRUCTIONS"
270 PRINTSPC9"Z move gun to the rig
ht"

```



## From Page 47

```

ht"
280 PRINTSPC9"X move gun to the le
ft"
290 PRINTSPC9"SPACE to fire a da
rt"
300 PRINTSPC14"PRESS ANY KEY"
310 PROCmulti_colour
320 MODE5:VDU19,1,1;0;
330 REM define characters
340 VDU23,224,0,6,140,127,255,206,1
03,0
350 VDU23,225,0,0,59,223,254,60,8,5
6
360 VDU23,226,24,24,24,126,44,124,2
8,56
370 VDU23,227,0,16,123,222,115,8,0,
0
380 VDU23,228,0,0,62,107,62,85,85,6
5
390 VDU23,229,0,0,56,32,48,8,56,0
400 VDU23,230,0,0,0,78,202,74,238,0
410 VDU23,231,126,153,153,102,70,40
,66,60
420 VDU23,232,16,56,16,16,16,16,56,
40
430 VDU23,233,16,16,16,24,20,60,31,
15
440 REM set up score array
450 RESTORE460:FORa=0TO7:READS(a):
NEXT
460 DATA 2,1,3,4,5,10,20,-2
470 PROCsetup_characters
480 REM the gun
490 G$=C$(2)+" "+CHR$(233)+" "
500 REM the darts
510 D$(0)=" "+CHR$(8)+CHR$(10)+" ":
D$(1)=C$(1)+CHR$(232)+CHR$(8)+CHR$(10)
)+" "
520 A$=" "
530 VDU19,3,10;0;
540 VDU23,8202;0;0;0;
550 *FX9 100
560 *FX10 100
570 PRINT"**** Fair Play ****"C$(
2)"Score Shots"
580 REM constants
590 A%=135:a%=&FFF4
600 REM U% V% & Z% temp numbers
610 REM U$ V$ & Z$ temp strings
620 REM initial settings for variab
les
630 T%=0:t%=1:S%=5:s%=1:L%=0
640 B%=9
650 TIME=1:M%=0:N%=0:RESTORE1180
660 REPEATJ%=24
670 C%=0:D%=0:H%=16:XX=0:Y%=4
680 I%=0

```

```

690 OX=0
700 REM main loop
710 REPEATC%=CX+1AND7:IF TIME>=M%*1
0READN,M%:IFFNT SOUND&12,5,(N%-17)*4
,255:SOUND&13,4,(N%-25)*4,255:TIME=0
720 PRINTTAB(B%,30)G$:VDU31,X%,Y%:P
RINT;FNp(D$(OX)):IFOXOX=0:PRINTTAB(0,
HX+6);A$:HX=HX-2:IFHX=4HX=6
730 IF(CXAND1)PRINTTAB(0,HX+(CXAND1
4))C$(CXMOD3+1)A$(CXDIV2)ELSEA$(CXDIV
2)=FNmove(A$(CXDIV2),CXAND3)
740 IFY%<>4Y%=Y%-1:IFY%=3Y%=4:D%=0
750 IFD%=0IFINKEY-99D%=1:X%=B%+1:Y%
=29:S%=S%-1:SOUND&10,-15,5,8
760 IFB%>11IFINKEY-98B%=B%-1
770 IFB%<16IFINKEY-67B%=B%+1
780 IFs%<X%=0:PRINTTAB(17,2)C$(1);S%
" ":IFS%<1I%=1 ELSEIFTX<>TXPRINTTAB(6
,2)C$(1);TX
790 UNTILI%:IFS%>0FORu%=100TO200STE
P3:SOUND1,1,u%,3:NEXT:LZ=L%+1:J%=24:I
%=0:S%=5:s%=1:PROCsetup_characters:PR
OCchange:RESTORE1180:UNTIL0
800 UNTILI%:Z%=prize$(7):IFTX<699Z$=
prize$((TXDIV100)AND7)
810 *FX15
820 FORA%=150TO0STEP-1:SOUND1,-AXDI
V10,A%,1:NEXT
830 MODE6:PRINTTAB(0,2)"Well done y
ou have won"Z$""PRESS ANY KEY
TO ENTER YOUR NAME""If your score
is high enough,of course":VDU23,8202
;0;0;0;
840 PROCmulti_colour:PROCscore_ladd
er(TX):GOTO170
850 REM end main loop
860 DEFFNmove(A$,A%)IFAZZ$=LEFT$(A$
,1):=RIGHT$(A$,19)+Z$ ELSEZ$=RIGHT$(A
$,1):=Z$+LEFT$(A$,19)
870 DEFFNp(A%)Z%=USRA%:IF?&469<>32D
%=0:IFY%>HX-1U%=(Y%-HX)DIV2:A$(U%)=LE
FT$(A$(U%),Y%-1)+" "+RIGHT$(A$(U%),
18-X%):Y%=4:s%=1:PROCcheck(S%(?&469-1
28)):J%=J%-1:IFJ%=0:I%=1
880 IFY%=HXs%=1
890 =D$(DX)
900 REM checks on characters hit th
en pays out accordingly
910 DEFPROCcheck(A%)IFAX>9S%=S%+AXD
IV2:SOUND&11,1,100,12:ENDPROC
920 IFAX=-20X=1:SOUND&11,2,120,15:E
NDPROC
930 TX=TX+AX+(3-U%)*10:SOUND&11,3,1
00,4:ENDPROC
940 DEFPROCsetup_characters:FORa=0T
O3:C$(a)=CHR$(17)+CHR$(a):A$(a)=""FO
Rz=0TO6:A$(a)=A$(a)+" "+CHR$(223+RND
(8)):A$(a)=LEFT$(A$(a),20):NEXT,:ENDP
ROC

```

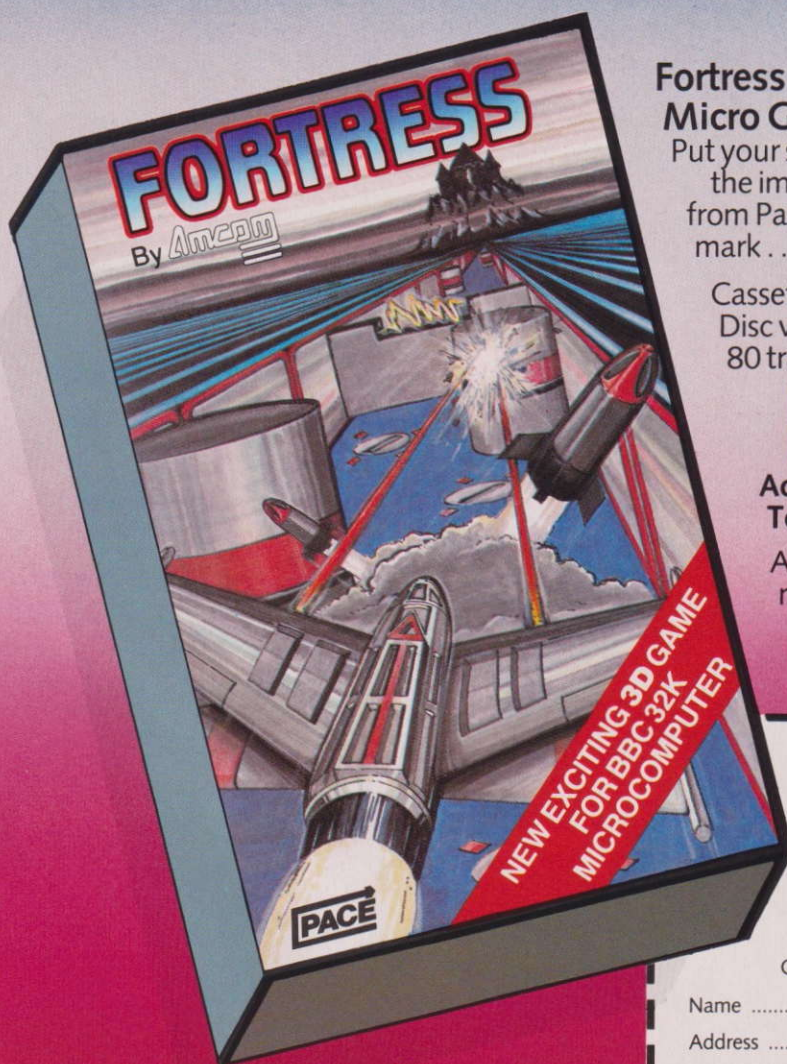
```

950 REM adds multi colour to mode 6
960 DEFPROCmulti_colour
970 REM do not abbreviate or add to
lines 990 & 1000
980 *FX15
990 REPEAT*FX19
1000 FORA%=1TO6:VDU19,1,A%;0;::NEXT
:UNTILINKEY(0)<>-1:CLS:ENDPROC
1010 REM change colours
1020 DEFPROCchange_R%(1)=RND(15):REP
EATR%(2)=RND(15):R%(3)=RND(15):UNTILR
%(1)<>R%(2)ANDR%(2)<>R%(3)ANDR%(1)<>R
%(3):FORa=1TO3:VDU19,a,R%(a);0;:NEXT:
ENDPROC
1030 REM score ladder
1040 DEFPROCscore_ladder(S%):*FX15
1050 VDU4,23,8202;0;0;0;:IFS%<S%(7)P
RINTTAB(11,10)"Please enter""SPC11"Y
our name"":IFFNs A%=-1:REPEATAX=A
%+1:UNTILS%>S%(A%):FORB%=7TOAXSTEP-1:
s%(B%+1)=s%(B%):s%(B%+1)=s%(B%):NEXT:
s%(A%)=S%:s%(A%)=Z$
1060 B$="-----"
1070 CLS:VDU23,8202;0;0;0;:PRINTSPC7
"Harry's Hall of HEROES""SPC11" Last
Score=";S%"" Scores "+B$+" Name
s"":FORA%=0TO7:PRINTSPC4" ";s%(A%)TAB
(11)B$TAB(25)" "s%(A%)":NEXT:PRINTSPC
11"PRESS ANY KEY":*FX15
1080 VDU19,1,6;0;:REPEATUNTILGET:END
PROC
1090 DEFFNs A%=0:Z$=""
1100 VDU7
1110 X%=GET:IFX%=13=1
1120 IFX%=127ANDAX=0GOTO1100
1130 IFX%=127:AX=AX-1:Z$=LEFT$(Z$,AX
)ELSEIFAX<12AX=AX+1:Z$=Z$+CHR%X%ELSEV
DU7
1140 PRINTTAB(11,14)" "Z$":GOTO111
0
1150 DEFFNT IFM%=0RESTORE1180:=0ELSE
=)
1160 UNTIL0
1170 REM mid C=25
1180 DATA 37,6,27,6,35,6,25,6
1190 DATA 34,2,33,2,34,2,37,4,36,2,3
5,2,34,2,33,2,34,4,31,2
1200 DATA 32,2,31,2,32,2,35,4,34,2,2
5,2,30,2,34,2,37,4,38,2
1210 DATA 39,2,38,2,39,2,41,4,41,2,4
1,6,39,4,38,2
1220 DATA 37,6,27,6,35,6,25,6
1230 DATA 34,2,33,2,34,2,37,4,36,2,3
5,2,34,2,31,2,32,6
1240 DATA 33,6,41,4,39,1,41,1,42,6,3
4,4,35,2
1250 DATA 36,2,39,2,37,2,35,4,29,2,3
0,6,34,2,35,2,36,2
1260 DATA 0,0

```



# Once you've got **FORTRESS** you might never see the end of it!



## Fortress sets the pace for other Micro Games

Put your skills to the ultimate test and overcome the impossible with the explosive new game from Pace. Meet the challenge and make your mark . . . **The fortress awaits you.**

Cassette **£8.85** inc. VAT

Disc version available. Usable on both 40 and 80 track drives. **£11.95** inc. VAT.



Access/Barclaycard orders.  
Tel: (0274) 729306

Available direct from Pace or from your nearest Micro Dealer.

**PACE**, 92 New Cross Street,  
Bradford, BD5 8BS.

Please send me ☐ 'Fortress' Cassettes at **£8.95** inc VAT

+55p P+P TOTAL **£9.50**

And/or ☐ 'Fortress' Discs at **£11.95** inc. VAT

+55p P+P. TOTAL **£12.50.**

Total remittance enclosed .....

Access/Barclaycard No .....

Cheque/PO No .....

Name .....

Address .....

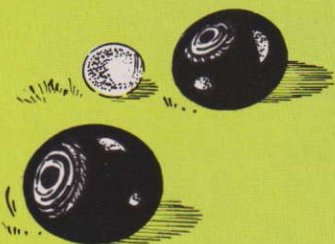
.....

..... Tel .....





# CROWN GREEN



**CROWN Green** is a vivid simulation of the game for two players. The program allows for different weights of bowl, 16 bowling directions and left or right bias. The wood – the name remains, even though many bowls are now made of plastic – even slows down to a halt realistically.

For those of you who like to play rough, full allowance is made for cannoning of woods into one another.

Each player plays two woods alternately from the footer (a non-slip-pable mat). The player who bowls the jack also retains possession of the footer until his bowl has stopped running.

If the jack comes to rest within less than 21 yards – strength 150 in our game – or is bowled directly off the green, the player loses it to the opponent.

If it is knocked off the green during an

end, that end is replayed and the same player retains the jack.

At the completion of each end a player scores one chalk – point – if his bowl is nearer the jack and two points if both are. The first player to 21 points is the winner.

Before each delivery instructions are given in the boxes at the top and bottom of the screen and a hint is given as to which way the run of the bowl will bend.

The lines drawn between each bowl and the jack at the finish of each end represent measuring strings and the player who wins the end bowls the jack at the start of the next one.

It is sometimes better to alter the direction and peg – bias – to avoid knocking your opponent in or yourself out.

## VARIABLES

A%,B%	Coordinates of the running bowl.
C%	Colour of each bowl.
P%	TRUE if a bowl has been hit or 2 if it has gone off the green.
M%,N%	Coordinates of any hit bowl.
H%	Number of bowl being bowled.
A%	Player.
HIT%	TRUE if any bowl has been hit.
S%,T%	Determines direction of running bowl and causes it to curve (bias).
F%	TRUE if jack is being bowled.
X%,Y%	Position of the footer.
A\$,B%,C%	PROCmessage parameters.
D%	Direction of bowl pointer arrow.
G\$	Bias selection (finger or thumb peg).
E%	Strength factor.
MANX%,MANY%	Position of man at start of delivery.
MANX2,MANY2	Position of man at the footer.
L%	Count for running bowl – random factor at line 990 brings an element of chance.
G%	1 if the jack has been knocked off during an end.

## PROCEDURES

PROCbowl	Moves current bowl and calls PROCwhbowlA and tests for a hit.
PROCover	The finish of each end.
PROCbend and PROCacross	Causes the bowl to bend according to the bias (G\$).
PROCmove	Moves only the "hit" bowls.
PROCwhbowlA	Tests for a hit on any still bowl.
PROCscores	Measures the bowls from the jack.
PROCslow	Ensures that each bowl wobbles to a halt instead of stopping dead.



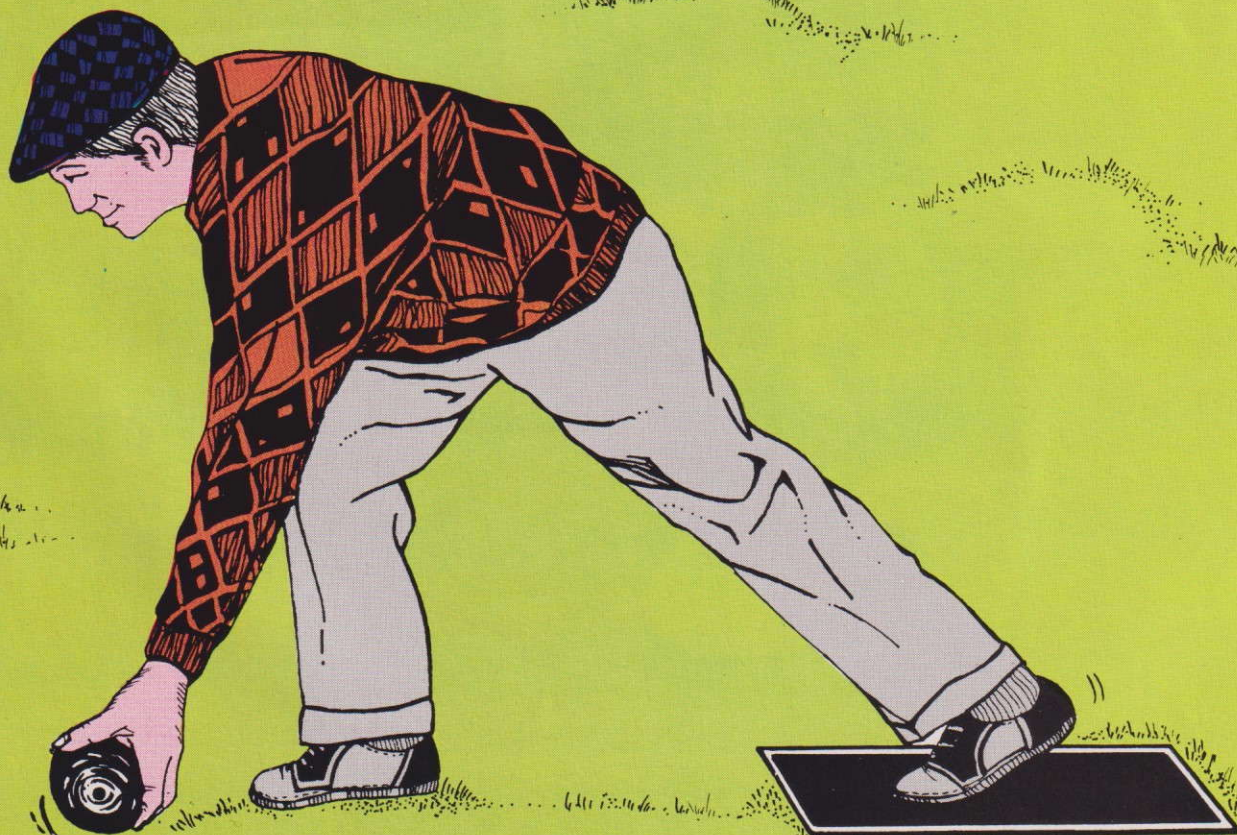




```

10 REM CROWN GREEN BOWLS
20 REM BY ALAN TOPHAM
30 REM THE MICRO USER
40 ON ERROR GOTO2490
50 DIM AX(5),BX(5),CX(5),PX(5),MX(
5),NX(5)
60 PROCinit:MODE1
70 VDU19,1,4,0,0,0,19,3,2,0,0,0
80 COLOUR1:PRINTTAB(1,1);CHR$248:P
RINTTAB(38,30);CHR$248:COLOUR2:PRINTT
AB(38,1);CHR$248:PRINTTAB(1,30);CHR$2
48
90 PROCscreen:PROCprintscores
100 REPEAT
110 CX(HX)=AX
120 IF AX=2 PROCmessage("BLUE BOWL
TO PLAY",11,30) ELSE PROCmessage("YEL
LOW BOWL TO PLAY",10,30)
130 PROCdirection
140 PROCbias
150 PROCstrength
160 PROCposman
170 PROCbowl
180 IF HX=6 THEN PROCover
190 UNTIL score1X>=21 OR score2X>=2
1
200 PROCfin
210 END
220 DEFPROCinit
230 VDU23,240,3,3,3,1,1,7,15,27,23,
241,192,192,192,128,128,224,224,224,2
3,242,51,99,195,131,3,3,3,23,243,22
4,224,240,216,204,198,195,193,23,244,
2,2,2,2,2,2,2,23,245,64,64,64,64,64
,64,64,64,23,246,0,56,124,124,124,56,
0,0

```





## From Page 51

```

240 VDU23,241,192,192,192,128,128,2
24,224,224
250 VDU23,224,16,56,84,146,16,16,16
,16,23,225,124,10,10,17,16,32,32,64,2
3,226,31,3,5,9,17,32,64,128,23,227,8,
6,1,7,25,97,129,0,23,228,8,4,2,255,2,
4,8,0,23,229,0,129,97,25,7,1,6,8,23,2
30,128,64,32,17,9,5,3,31
260 VDU23,231,64,32,32,16,17,10,10,
124,23,232,8,8,8,8,73,42,28,8,23,233,
2,4,4,8,136,80,80,62,23,234,1,2,4,136
,144,160,192,248,23,235,0,129,134,152
,224,128,96,8,23,236,16,32,64,255,64,
32,16,0,23,237,8,96,128,224,152,134,1
29,0
270 VDU23,238,240,192,160,144,136,4
,2,1,23,239,62,80,80,136,8,4,4,2,23,2
47,0,0,24,60,60,24,0,0,23,248,16,48,1
12,240,16,16,16,16
280 HIT%=FALSE:Z=0:F%=TRUE:H%=1:score1%
=0:score2%=0:A%=2:X%=650:Y%=200
290 ENDPROC
300 DEFPROCscreen
310 MOVE20,20:DRAW1259,20:DRAW1259,
1003:DRAW20,1003:DRAW20,20
320 VDU24,75;75;1204;948;
330 GCOL0,131:CL6
340 ENDPROC
350 DEFPROCmessage(A$,B$,C%)
360 VDU4
370 PRINTTAB(B$,C%);A$
380 VDU5
390 ENDPROC
400 DEFPROCdirection
410 *FX15,1
420 PROCmessage("SELECT DIRECTION W
ITH (Z&X) KEYS",4,1)
430 GCOL0,0:D%=224
440 MOVEX%,Y%:VDU42,11,8,D%
450 REPEAT
460 E=INKEY(500)
470 IF E=88 THEN D%=D%+1:IF D%=240
THEN D%=224
480 IF E=90 THEN D%=D%-1:IF D%=223
THEN D%=239
490 VDU127,D%
500 UNTIL E=&0D
510 ENDPROC
520 DEFPROCbias
530 GOSUB2350
540 IF D%=227 OR D%=228 OR D%=229 P
ROCmessage("SELECT BIAS (Z UP X DOWN)
",7,1) ELSE IF D%=235 OR D%=236 OR D%
=237 PROCmessage("SELECT BIAS (X UP Z
DOWN)",7,1) ELSE PROCmessage("SELECT
BIAS (Z LEFT X RIGHT)",7,1)
550 REPEAT:G%=GET$:UNTIL G%="Z" OR

```

```

G%="X"
560 GOSUB2350
570 ENDPROC
580 DEFPROCstreng
590 PROCmessage("STRENGTH (MAX 300)
",7,1)
600 VDU4
610 INPUTTAB(25,1),EX:IF EX>300 THE
N GOSUB2350:PROCstreng
620 GOSUB2340
630 GOSUB2350
640 ENDPROC
650 DEFPROCposman
660 MOVEX%,Y%:VDU9,127,11,9,127
670 RESTORE 710
680 FOR Q=224 TO D%
690 READ MANX%,MANY%,MANX2,MANY2,S%
,T%
700 NEXT
710 DATA-57,-20,-57,60,0,5,-77,-20,
-57,60,2,5,-96,-20,-57,60,4,5,-96,38,
-57,78,5,2,-96,96,0,96,5,0,-96,108,0,
75,5,-2,-96,120,24,55,4,-5,-38,120,24
,55,2,-5
720 DATA20,120,24,55,0,-5,40,120,24
,55,-2,-5,72,120,24,55,-4,-5,72,80,24
,48,-5,-2,72,40,0,40,-5,0,72,10,0,50,
-5,2,72,-20,-57,60,-4,5,10,-20,-57,60
,-2,5
730 MOVEX%+MANX%,Y%+MANY%:GOSUB2360
740 IF D%=224 OR D%=225 OR D%=226 O
R D%=227 OR D%=236 OR D%=237 OR D%=23
8 OR D%=239 MOVE X%+MANX%+60,Y%+MANY%
-60 ELSE MOVE X%+MANX%-24,Y%+MANY%-48
750 IF F% GCOL3,3:VDU247 ELSE GCOL3
,A%:VDU246
760 TIME=0:REPEAT UNTILTIME>90
770 IF F% VDU8,247 ELSE VDU8,246
780 MOVE X%+MANX%,Y%+MANY%:GOSUB236
0
790 MOVE X%+MANX2,Y%+MANY2:GOSUB236
0
800 ENDPROC
810 DEFPROCbowl
820 IF H%=1 THEN GCOL3,3 ELSE GCOL3
,A%
830 AX(H%)=X%:BX(H%)=Y%
840 MOVE AX(H%),BX(H%):IF H%=1 VDU2
47 ELSE VDU246
850 LX=0
860 REPEAT
870 LX=LX+1
880 IF LX MOD 50=0 PROCbend
890 PROCmove
900 IF H%=1 THEN GCOL3,3 ELSE GCOL3
,C%(H%)
910 X1%=AX(H%)+S%:Y1%=BX(H%)+T%
920 MOVE AX(H%),BX(H%):IF H%=1 VDU2
47 ELSE VDU246

```

```

930 MOVE X1%,Y1%:IF H%=1 VDU247 ELS
E VDU246
940 AX(H%)=X1%:BX(H%)=Y1%
950 IF AX(H%)<90 OR AX(H%)>1165 OR
BX(H%)>940 OR BX(H%)<110 THEN SOUND0,
-15,6,1:MOVEAX(H%),BX(H%):GOSUB2370:P
%(H%)=2:L%=EX
960 IF LX<60 THEN990
970 PROCwhbowlA:IF HIT% SOUND0,-12,
3,1
980 PROCmove
990 IF RND(4)=2 LX=LX+2
1000 IF LX>(EX-60) THEN PROCslow
1010 UNTIL LX>=EX
1020 IF P%(1)=2 THEN PROCjackoff:END
PROC
1030 H%=H%+1
1040 REPEAT
1050 HIT%=FALSE
1060 FOR F=1 TO 5
1070 IF P%(F)=2 THEN 1090
1080 P%(F)=FALSE
1090 NEXT
1100 PROCwhbowlA:PROCmove
1110 IF P%(1)=2 G%=1:PROCjackoff:HIT
%=FALSE
1120 UNTIL HIT%=FALSE
1130 IF F%=TRUE AND EX<150 THEN PROC
message("FAILED TO SET A MARK(minimum
150)",4,1):SOUND1,-15,1,3:T=TIME:REP
EAT UNTIL TIME-T>200:CLG:F%=TRUE:H%=1
:GOTO1160
1140 MOVEX%+MANX2,Y%+MANY2:GOSUB2360
1150 IF F%=TRUE THEN F%=FALSE:ENDPRO
C
1160 PROCchplayer
1170 ENDPROC
1180 DEFPROCover
1190 MOVEX%-120,Y%:GOSUB2360:PROCchp
layer:MOVEX%+80,Y%+20:GOSUB2360
1200 PROCmessage("MEASURING NOW",12,
1)
1210 X%=AX(1):Y%=BX(1):IF X%<200 THE
N X%=200 ELSE IF X%>1030 X%=1030
1220 IF Y%<200 Y%=200 ELSE IF Y%>800
Y%=800
1230 PROCscores
1240 PROCprintscores
1250 FOR T=1 TO 3000:NEXT
1260 GOSUB2350
1270 IF score1%>=21 OR score2%>=21 T
HEN ENDPROC
1280 PROCmessage("PRESS SPACE BAR",1
2,30)
1290 FOR T=1 TO 5:P%(T)=0:NEXT
1300 G=GET:IF G<>32 THEN 1300
1310 CLG:H%=1:F%=TRUE
1320 ENDPROC
1330 DEFPROCbend

```



```

1340 IF DX=227 OR DX=228 OR DX=229 O
R DX=235 OR DX=236 OR DX=237 PROCacro
ss:ENDPROC
1350 IF TX>1 TX=TX-1:IF TX<=1 TX=1 E
LSE IF TX<-1 TX=TX+1:IF TX>=-1 TX=-1
1360 IF G$="Z" SX=SX-1
1370 IF G$="X" SX=SX+1
1380 ENDPROC
1390 DEFPROCacross
1400 IF TX=4 TX=4 ELSE IF TX<=-4 TX
=-4
1410 IF SX>1 SX=SX-1:IF SX<=1 SX=1 E
LSE IF SX<-1 SX=SX+1:IF SX>=-1 SX=-1
1420 IF DX=235 OR DX=236 OR DX=237 T
HEN 1460
1430 IF G$="Z" TX=TX+1:SX=SX+1
1440 IF G$="X" TX=TX-1:SX=SX+1
1450 ENDPROC
1460 IF G$="Z" TX=TX-1:SX=SX-1
1470 IF G$="X" TX=TX+1:SX=SX-1
1480 ENDPROC
1490 DEFPROCmove
1500 FOR WX=1 TO HX-1
1510 IF P%(WX)=2 OR P%(WX)=FALSE THE
N 1590
1520 IF WX=1 THEN GCOL3,3 ELSE GCOL3
,CX(WX)
1530 X1%=AX(WX)+MX(WX):Y1%=BX(WX)+NX
(WX)
1540 MOVE X1%,Y1%:IF WX=1 VDU247 ELS
E VDU246
1550 MOVE AX(WX),BX(WX):IF WX=1 VDU2
47 ELSE VDU246
1560 AX(WX)=X1%:BX(WX)=Y1%
1570 LX=LX+1
1580 IF AX(WX)<90 OR AX(WX)>1165 OR
BX(WX)>940 OR BX(WX)<110 THEN SOUND0,
-15,6,1:MOVEAX(WX),BX(WX):GOSUB2370:P
X(WX)=2:IF WX=1 LX=EX:GX=1
1590 NEXT
1600 ENDPROC
1610 DEFPROCscores
1620 FOR Y=2 TO 5
1630 IF P%(Y)=2 THEN 1820
1640 A=0
1650 X1=AX(Y):Y1=BX(Y)
1660 REPEAT
1670 IF X1<AX(1) THEN X1=X1+1 ELSE X
1=X1-1
1680 IF Y1<BX(1) THEN Y1=Y1+1 ELSE Y
1=Y1-1
1690 A=A+1
1700 UNTIL X1=AX(1) OR Y1=BX(1)
1710 MX(Y)=A#4
1720 A=0
1730 REPEAT
1740 IF X1<AX(1) THEN X1=X1+1
1750 IF X1>AX(1) THEN X1=X1-1
1760 IF Y1<BX(1) THEN Y1=Y1+1

```

```

1770 IF Y1>BX(1) THEN Y1=Y1-1
1780 A=A+1
1790 UNTIL X1=AX(1) AND Y1=BX(1)
1800 NX(Y)=A#3
1810 MOVEAX(1)+12,BX(1)-12:GCOL3,CX(
Y):DRAWAX(Y)+12,BX(Y)-12
1820 NEXT
1830 FOR Y=2 TO 5:IF P%(Y)=2 THEN 18
60
1840 MOVEAX(1)+12,BX(1)-12:GCOL3,CX(
Y):DRAWAX(Y)+12,BX(Y)-12
1850 FOR T=1 TO 1000:NEXT
1860 NEXT
1870 FOR Y=2 TO 5:IF P%(Y)=2 THEN P%
(Y)=5000:MX(Y)=Y:NEXT ELSE P%(Y)=(MX(
Y)+NX(Y)):MX(Y)=Y:NEXT
1880 FOR X=1 TO 3
1890 FOR Y=2 TO 4
1900 IF P%(Y)>P%(Y+1) THEN P%(1)=P%(
Y):P%(Y)=P%(Y+1):P%(Y+1)=P%(1):MX(1)=
MX(Y):MX(Y)=MX(Y+1):MX(Y+1)=MX(1)
1910 NEXT
1920 NEXT
1930 IF P%(2)=5000 THEN ENDPROC
1940 IF CX(MX(2))=2 THEN PROCblwin E
LSE PROCylwin
1950 ENDPROC
1960 DEFPROCprintscores
1970 FOR A=80 TO 120 STEP 10:SOUND1,
-11,A,1:NEXT
1980 VDU4
1990 COLOUR1:PRINTTAB(3,30):score1%:
COLOUR2:PRINTTAB(35,30):score2%
2000 VDU5
2010 ENDPROC
2020 DEFPROCjackoff
2030 PROCmessage(" JACK OFF THE GR
EEN ",7,1):SOUND1,-15,1,3:T=TIME:REPE
AT UNTIL TIME-T>200
2040 FOR T=1 TO 5:P%(T)=FALSE:NEXT
2050 HX=1:CLG:F%TRUE
2060 IF GX=1 AX=CX(2) ELSE PROCchpla
yer
2070 GX=0
2080 ENDPROC
2090 DEFPROCwhbow1A
2100 HIT%=FALSE
2110 FOR QX=1 TO HX-1
2120 IF P%(QX)=2 OR P%(QX)=TRUE THEN
2210
2130 IFPOINT(AX(QX)-4,BX(QX)+3)<>3 T
HEN MX(QX)=2:NX(QX)=-3:P%(QX)=TRUE:HI
TX=TRUE
2140 IFPOINT(AX(QX)+12,BX(QX)+6)<>3
THEN MX(QX)=0:NX(QX)=-3:P%(QX)=TRUE:HI
TX=TRUE
2150 IFPOINT(AX(QX)+28,BX(QX)+3)<>3
THEN MX(QX)=-2:NX(QX)=-3:P%(QX)=TRUE:
HITX=TRUE

```

```

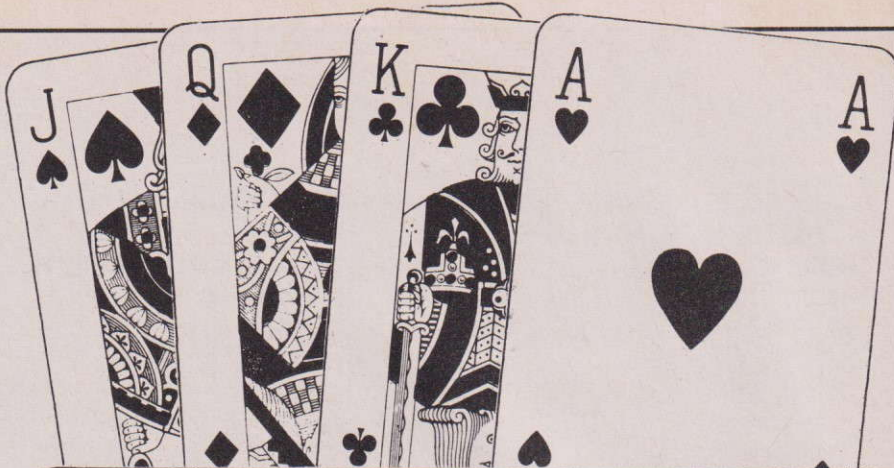
2160 IFPOINT(AX(QX)-6,BX(QX)-12)<>3
THEN MX(QX)=3:NX(QX)=0:P%(QX)=TRUE:HI
TX=TRUE
2170 IFPOINT(AX(QX)+30,BX(QX)-12)<>3
THEN MX(QX)=-3:NX(QX)=0:P%(QX)=TRUE:
HITX=TRUE
2180 IFPOINT(AX(QX)+28,BX(QX)-28)<>3
THEN MX(QX)=-2:NX(QX)=3:P%(QX)=TRUE:
HITX=TRUE
2190 IFPOINT(AX(QX)+12,BX(QX)-30)<>3
THEN MX(QX)=0:NX(QX)=3:P%(QX)=TRUE:HI
TX=TRUE
2200 IFPOINT(AX(QX)-4,BX(QX)-28)<>3
THEN MX(QX)=2:NX(QX)=3:P%(QX)=TRUE:HI
TX=TRUE
2210 NEXT
2220 ENDPROC
2230 DEFPROCblwin
2240 IF CX(MX(2))=CX(MX(3)) THEN PRO
Cmessage("TWO TO BLUE",14,30):score1%
=score1%+2 ELSE PROCmessage("ONE TO B
LUE",14,30):score1%=score1%+1
2250 AX=2
2260 ENDPROC
2270 DEFPROCylwin
2280 IF CX(MX(2))=CX(MX(3)) THEN PRO
Cmessage("TWO TO YELLOW",12,30):score
2%=score2%+2 ELSE PROCmessage("ONE TO
YELLOW",12,30):score2%=score2%+1
2290 AX=1
2300 ENDPROC
2310 DEFPROCchplayer
2320 IF AX=2 THEN AX=1:ENDPROC
2330 IF AX=1 THEN AX=2:ENDPROC
2340 VDU4:PRINTTAB(3,30):SPC(34):VDU
5:RETURN
2350 VDU4:PRINTTAB(3,1):SPC(34):VDU5
:RETURN
2360 VDU18,3,3,240,241,10,8,8,18,3,A
X,242,243,18,3,3,10,8,8,244,245:RETUR
N
2370 IF WX=1 OR HX=1 THEN VDU247:RET
URN ELSE VDU246:RETURN
2380 DEFPROCfin
2390 VDU4,22,7
2400 VDU23:8202;0;0;0;
2410 IF score1%>score2% PRINTTAB(7,1
2):"BLUE WINS 21 chalks to ";score2%
ELSE PRINTTAB(7,12):"YELLOW WINS 21
chalks to ";score1%
2420 TIME=0:REPEAT UNTIL TIME>300
2430 PRINTTAB(6,20)CHR$136;"PRESS AN
Y KEY TO PLAY AGAIN"
2440 G=BET:RUN
2450 DEFPROCslow
2460 IF SX<0 SX=-1 ELSE SX=1
2470 IF TX<0 TX=-1 ELSE TX=1
2480 ENDPROC
2490 REPORT:PRINTERR:PRINTERL

```



# Lay your cards on the table

By **STUART MENEFY**



## PROCEDURES

<b>PROC_DEFINE</b>	Defines characters, envelopes, sets up DIMs and major variables.
<b>PROC_INIT</b>	Sets up variables for each new game.
<b>PROC_SCREEN</b>	Displays screen.
<b>PROC_GO</b>	Decides who starts.
<b>PROC_SHUFFLE</b>	Sets up and shuffles the pack.
<b>PROC_SOUND</b>	Makes a bleep, when an input is not valid.
<b>PROC_DELETE</b>	Removes last character from input string.
<b>PROC_COMMENT</b>	Displays a centralised comment for required period of time.
<b>PROC_HUMAN</b>	Accepts and validates the humans entry, and performs the required function, if possible.
<b>PROC_POSSIBLE</b>	Finds out if a card may be played from the current hand.
<b>PROC_DETAILS</b>	Calculates the representation, value and suit of a card.
<b>PROC_COMPUTER</b>	Plays a card if possible from the micro's hand.
<b>PROC_PLAY</b>	Plays a card, and removes it from the hand.
<b>PROC_FLIP</b>	Attempts to make the sound of a card being put down!
<b>PROC_DIS_HAND</b>	Displays the human's hand.
<b>PROC_ADVICE</b>	Tells the human which card to play, using the micro's criteria.
<b>PROC_SWAP</b>	If possible, swaps the human's and micro's hands.
<b>PROC_INSTRUCTIONS</b>	Displays brief instructions, and asks for the human's name, the time for which messages are displayed, and if hands can be swapped.

## FUNCTIONS

<b>FN_DEAL</b>	Moves a card from the top of the pack to one of the hands.
<b>FN_INPUT</b>	Asks the human for his/her instruction.
<b>FN_COMPUTER</b>	Returns the position of the 'best' card to play.

**FN\_END**

Displays which player won, makes an appropriate sound, and asks if the human would like another go.

**FN\_YES\_NO**

Only returns TRUE if "Y" pressed, FALSE if "N" pressed.

## VARIABLES

**CARDS\$(13)**

Contains the representation of each card.

**C%(8)**

The last card played in each section (see below).

**CARDS\$**

A representation of a card, including colour control characters, as determined in PROC\_DETAILS.

**CARD%**

The cards face value, from 1 to 13, as determined in PROC\_DETAILS.

**DELAY%**

Delay while messages are displayed, in centiseconds.

**end%**

Whether the game has finished or not. In which section the most cards can be played, used to determine the micro's go.

**G%(8)**

**GO%**

The current player (see below).

**GO\_POS%**

The position in the current hand of the best card to play

**GOS**

Command entered by human.

**HAND%(1)**

Position in memory where each hand starts.

**HAND\_LEN%(1)**

Length of each hand.

**NAMES\$**

Human's name.

**NUM\_GO%**

Number of cards that can be played.

**PH%(8)**

Position in the hand of each card that can be played.

**PACK%**

Position of the start of the pack in memory.

**PACK\_LEN%**

Number of cards in the pack.

**P1% & P2%**

Used in shuffling the pack.

**PH%**

Position in the human's hand of the card he/she wants to play.

**SCORE0% &**

Number of games won by the human, and micro.

**SCORE1%**

**SNUM%**

Number of times the human may swap hands with the micro.

**SUIT%**

Suit number, from 1 to 4, as determined in PROC\_DETAILS.

**VLU%**

Card's internal value, as determined in PROC\_DETAILS.



**SEVENS** is a traditional card game, depending partly upon luck and partly upon the skill of the players.

The game is played with a normal pack of playing cards, the object being to lay down all your cards before your opponent.

However cards may only be laid down in order, starting from seven, going up to king, and down to ace, and may only be laid down with other cards of the same suit.

Each player starts with seven cards, dealt from the pack, and puts down a card in turn. If he or she is unable to go, then a card must be picked up from the pack, and added to the hand.

In this way, four piles of cards develop, in order, and the winner is the player who is able to put down his or her final card.

In this version you play against the micro, who cannot "see" your hand, although it is displayed on the screen.

The player can also swap hands with the micro a limited number of times – as set by SNUM% – and can also give advice on which card to play.

The logic behind deciding which card to play is simple, as can be seen in FN\_COMPUTER, and is used by the micro for its own goes, and also to give advice.

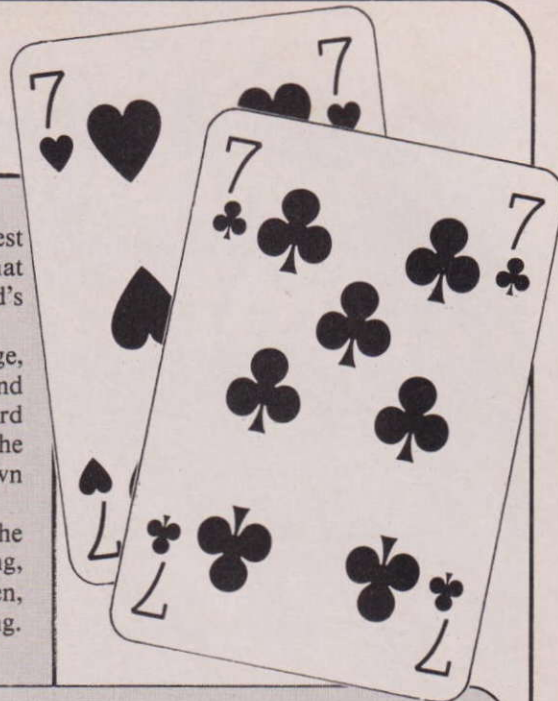
Having first checked to see how many cards can possibly be played (PROC\_POSSIBLE), the micro returns a -1 if no card can be played. If only one card may be played then its position is returned, otherwise the remaining logic comes into play.

This first checks to see how many cards are held in the hand in each 'section'. There are eight sections, two for each suit, which extend from seven (which is included in both sections), either up or down.

The section that has the greatest number of cards in, and has a card that can be played, is used, and that card's position in the hand is returned.

This gives the player one advantage, in that the micro does not understand 'blocking'. This is when you hold a card for as long as possible, to prevent the micro being able to lay down his own cards, which come after it.

For example, if it is thought that the micro holds the jack, queen and king, then by delaying putting down the ten, you stand a greater chance of winning.



```

10 REM SEVENS
20 REM by S.I.Menefy
30 REM for 32K BBC
40 REM (c) Micro User
50 MODE7
60 PROC_DEFINE
70 PROC_INSTRUCTIONS
80 PROC_INIT
90 MODE 1
100 PROC_SCREEN
110 PROC_60
120 PROC_SHUFFLE
130 REPEAT
140 IF GOX PROC_COMPUTER ELSE PROC_
HUMAN
150 UNTIL HAND_LEN%(0)=-1 OR HAND_L
EN%(1)=-1 OR end%
160 IF NOT(end%) THEN IF FN_END THE
N 80
170 MODE 7
180 *FX4
190 END
200 DEF PROC_DEFINE
210 VDU 23,224,54,127,127,127,62,28
,8,0
220 VDU 23,225,8,28,62,127,127,28,6
2,0
230 VDU 23,226,8,28,62,127,62,28,8,
0
240 VDU 23,227,28,28,107,127,107,8,
28,0
250 VDU 23,228,0,0,0,0,0,96,96,0
260 ENVELOPE 1,1,2,0,0,60,0,0,127,0
,0,-127,126,126
270 ENVELOPE 2,3,0,0,0,0,0,121,-1
3,0,-10,120,120
280 ENVELOPE 3,0,-2,0,0,100,0,0,1,0
,0,-1,1,0
290 ON ERROR REPORT:PRINT" at line
";ERL:END
300 SCORE0%=0:SCORE1%=0

310 DIM HAND%(1),HAND_LEN%(1),CARD$
(13),C%(8),G%(8),PH%(8):PACK%=&900
320 HAND%(0)=PACK%+52:HAND%(1)=PACK
%+104
330 FOR AX=1 TO 13:READ CARD$(AX):N
EXT AX
340 ENDPROC
350 DATA ACE,2,3,4,5,6,7,8,9,10,JAC
K,QUEEN,KING
360 DEF PROC_INIT
370 ON ERROR IF ERR=17 THEN 130 ELS
E REPORT:PRINT" at line ";ERL:END
380 end%=FALSE
390 SNUM%=2

400 PACK_LEN%=51:HAND_LEN%(0)=-1:HA
ND_LEN%(1)=-1
410 ENDPROC
420 DEF PROC_SCREEN
430 VDU 19,2,2;0;
440 VDU 23,0,10,32,0;0;0;24,0;336;9
76;944;18,0,130,16,26
450 PRINT"Games to "NAME$ " :SCORE0
%
460 PRINT"Games to computer " :SCORE
1%
470 RESTORE 520
480 FORAX=1 TO 4
490 READ Y%
500 MOVE 0,Y%:DRAW 976,Y%
510 NEXT
520 DATA 880,432,336,208
530 MOVE 0,944:DRAW 1280,944
540 MOVE 976,1024:DRAW 976,0

550 FOR AX=0 TO 3:MOVE AX*224+80,94
4:DRAW AX*224+80,432:NEXT AX
560 VDU 5
570 FOR AX=1 TO 13
580 MOVE 16+(16*(LEN(CARD$(AX))=2))

```

In the arrays and scoring, the human is player 0, and the micro is player 1.

Also, the word "section" in arrays C%(8) and G%(8), section 1 (or element 1) is from seven hearts, to the king hearts, 2 is from the ace hearts to the seven hearts, and similarly 3 and 4 are clubs, 5 and 6 diamonds, and 7 and 8 spades.



## From Page 55

```
,AZ=32+440
590 PRINT LEFT$(CARD$(AZ),1)-(LEN(CARD$(AZ))=2))
600 NEXT AZ
610 FOR AZ=1 TO 4
620 MOVE AZ*224-48,928
630 GCOL0,AZ MOD 2
640 PRINT CHR$(AZ+223)
650 NEXT AZ
660 GCOL 0,3
670 MOVE 984,1000
680 PRINT"Your hand"
690 VDU 4
700 PRINT TAB(0,22)"Number left in pack 52"
710 PRINT"Number in computers hand 7"
720 IF SZ PRINT"Number of swaps left 2" ELSE PRINT"You cannot swap hands"
730 PRINT TAB(13)"KEYS"
740 PRINT TAB(2,27)"Cannot go"
750 PRINT" End game"
760 PRINT" Advice on what to do"
770 IF SZ PRINT" Swap hands with the computer"
780 PRINT"or number of the required card";
790 COLOUR 1
800 PRINT TAB(0,27)"C""E""A""CHR$(83+SZ)
810 COLOUR 3
820 COLOUR 130
830 ENDPROC
840 DEF PROC_60
850 IF SCORE0X=SCORE1X 60X=RND(2)-1 ELSE IF SCORE0X<SCORE1X 60X=0 ELSE 60X=1
860 IF 60X PRINT TAB(10,19)"I go first" ELSE PRINT TAB(8,19)"You go first"
870 ENDPROC
880 DEF PROC_SHUFFLE
890 SOUND 0,-13,3,60
900 SOUND 1,3,255,60
910 AZ=1
920 FOR BZ=0 TO 51
930 IF AZ MOD 20=14 AZ=AZ+7
940 PACKZ?BZ=AZ
950 AZ=AZ+1
960 NEXT BZ
970 FOR AZ=1 TO 50+RND(50)
980 P1X=RND(52)+PACKZ-1:P2X=RND(52)+PACKZ-1
990 TX=?P1X
1000 ?P1X=?P2X
1010 ?P2X=TX
1020 NEXT AZ
```

```
1030 FOR AZ=0 TO 6
1040 FOR BZ=0 TO 1
1050 CZ=FN_DEAL(BZ)
1060 NEXT BZ
1070 NEXT AZ
1080 PRINT TAB(0,20)SPC(30)
1090 PROC_DIS_HAND
1100 ENDPROC
1110 DEF FN_DEAL(CZ)
1120 IF PACK_LENZ=-1 =FALSE
1130 LZ=-1
1140 REPEAT
1150 LZ=LZ+1
1160 UNTIL HANDZ(CZ)?LZ>PACKZ?PACK_LENZ OR LZ>HAND_LENZ(CZ)
1170 FOR KZ=HANDZ(CZ)+HAND_LENZ(CZ) TO HANDZ(CZ)+LZ STEP-1:?(KZ+1)=?KZ:NEXT
1180 HAND_LENZ(CZ)=HAND_LENZ(CZ)+1
1190 HANDZ(CZ)?LZ=PACKZ?PACK_LENZ
1200 PACK_LENZ=PACK_LENZ-1
1210 COLOUR 128
1220 PRINT TAB(20,22);PACK_LENZ+1;"
1230 COLOUR 130
1240 PROC_FLIP
1250 =TRUE
1260 DEF FN_INPUT
1270 *FX15
1280 COLOUR 3
1290 VDU 23,0,10,96,0;0;0;
1300 INPUT TAB(0,19)"Please enter your go"SPC(10)TAB(21,19)input$
1310 VDU 23,0,10,32,0;0;0;
1320 IF VAL(input$)>HAND_LENZ(0)+1 PROC_SOUND:PROC_COMMENT("You don't have "+input$+" cards",FALSE):GOTO 1270
1330 IF input$="" PROC_SOUND:GOTO 1270
1340 =input$
1350 DEF PROC_SOUND
1360 SOUND 1,-15,101,2
1370 ENDPROC
1380 DEF PROC_DELETE
1390 IF input$=""ENDPROC
1400 input$=LEFT$(input$,LEN(input$)-1)
1410 VDU 127
1420 ENDPROC
1430 DEF PROC_COMMENT(COMMENT$,COLZ)
1440 PRINT TAB((30-(LEN(COMMENT$)+COLZ*5))/2,20)COMMENT$;
1450 AZ=INKEY(DELAYZ)
1460 PRINT TAB(0,20)SPC(30)
1470 ENDPROC
1480 DEF PROC_HUMAN
1490 PROC_POSSIBLE
1500 60$=FN_INPUT
1510 IF LEFT$(60$,1)<>"C" THEN 1590
1520 IF NUM_60X>0 PROC_COMMENT("You
```

```
can go",FALSE):GOTO 1500
1530 6Z=FN_DEAL(0)
1540 PROC_DETAILS(LZ,0)
1550 IF 6Z PROC_COMMENT("You picked up "+CARD$,TRUE) ELSE PROC_COMMENT("The pack is empty",FALSE)
1560 PROC_DIS_HAND
1570 60X=1
1580 ENDPROC
1590 IF LEFT$(60$,1)="A" AND NUM_60X>0 PROC_ADVICE:GOTO 1500
1600 IF LEFT$(60$,1)="A" THEN 1650
1610 IF LEFT$(60$,1)="S" PROC_SWAP:GOTO 1490
1620 IF LEFT$(60$,1)="E" endZ=TRUE:ENDPROC
1630 PHZ=VAL(60$)-1
1640 IF PHZ=-1 PROC_SOUND:PROC_COMMENT("Pardon?",FALSE):GOTO 1500
1650 IF NUM_60X=0 PROC_COMMENT("You can't go",FALSE):GOTO 1530
1660 PROC_DETAILS(PHZ,0)
1670 FOR AZ=1 TO NUM_60X
1680 IF PHZ=PHZ(AZ) THEN AZ=9
1690 NEXT AZ
1700 IF AZ<9 PROC_COMMENT("You cannot play that card",FALSE):GOTO 1500
1710 PROC_PLAY(PHZ)
1720 PROC_DIS_HAND
1730 60X=1
1740 ENDPROC
1750 DEF PROC_POSSIBLE
1760 NUM_60X=0
1770 FOR AZ=0 TO HAND_LENZ(60X)
1780 PROC_DETAILS(AZ,60X)
1790 IF CZ((SUITZ*2)+(CARDZ*6))=VLUX-1 AND CARDZ>6 THEN NUM_60X=NUM_60X+1:PHZ(NUM_60X)=AZ
1800 IF CZ((SUITZ*2)+(CARDZ*6))=VLUX+1 AND CARDZ<=6 THEN NUM_60X=NUM_60X+1:PHZ(NUM_60X)=AZ
1810 IF CARDZ=7 THEN NUM_60X=NUM_60X+1:PHZ(NUM_60X)=AZ
1820 NEXT AZ
1830 ENDPROC
1840 DEF PROC_DETAILS(posZ,DZ)
1850 VLUX=HANDZ(DZ)?posZ
1860 SUITZ=(VLUX DIV 20)+1
1870 CARDZ=VLUX MOD 20
1880 CARD$=CHR$ 17+CHR$(SUITZ MOD 2)+CHR$ 18+CHR$ 0+CHR$(SUITZ MOD 2)+CARD$(CARDZ)+CHR$(223+SUITZ)
1890 ENDPROC
1900 DEF FN_COMPUTER
1910 PROC_POSSIBLE
1920 IF NUM_60X=1 =PHZ(1)
1930 IF NUM_60X=0 =-1
1940 FOR AZ=1 TO 8:6Z(AZ)=0:NEXT AZ
```



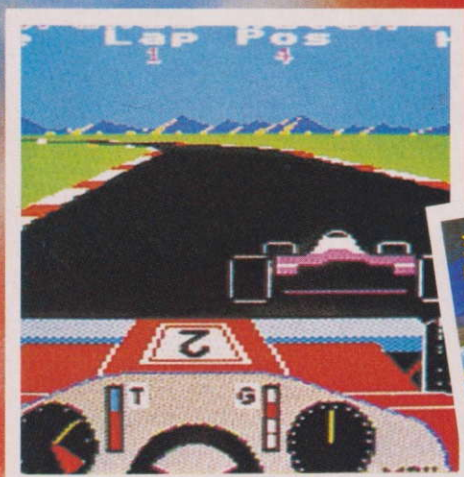
# mixed moods

## two amazing new games

### from Software Invasion...

## 3D Grand Prix

Software Invasion games are available from WH SMITH, HMV, all SPECTRUM SHOPS and over 1,000 retail outlets.



This is the game you've been waiting for! A 3D high speed formula 1 car race round 8 different international circuits.

Just look at these amazing features:-

- \* Bends arranged to simulate 8 actual Grand Prix circuits.
- \* Working cockpit - Rev. counter, Speedometer, Temperature Gauge, gears, moving steering wheel.
- \* Other cars which alter in perspective and size as they overtake.
- \* Signposts.
- \* Realistic sound effects such as engine noise, skids and crunching shunts.
- \* Keyboard or Joystick control.
- \* Volume control.

Runs on O.S. 1.0 or greater. More than a year of man hours have gone into creating this superb game.

BBC MICRO MODEL B

£9.95 CASSETTE

£11.95 DISK

Authors: Mick O'Neill and Dave Mendes

Blitzkrieg is a 3D Tank Battle from the same author as our Best Seller "3D Bomb Alley".

This program is probably unique in the fact that all the graphics were created with the help of models, a video camera, a video digitiser and some very special Light Pen software.

This has resulted in extremely fine detail and realism, with fast moving real time graphics - you don't have to wait for the screen to be updated between each move.

Features include:-

- \* Detailed scrolling landscape.
- \* Rotating tanks.
- \* Detailed explosions.
- \* 3D shell movement.
- \* Recoiling barrel.
- \* 3D sound.
- \* Keyboard/Joystick option.

\* Freeze game facility.

\* Optional sound.

\* Demo' mode.

\* High score & ranking system.

BBC MICRO MODEL B

£7.95 CASSETTE

£9.95 DISK

Author: Simon Vout.

## Blitzkrieg



NOW WITH COLOUR GRAPHICS! (BBC VERSION ONLY)



### HOW TO ORDER

You may purchase any of the games above direct. All you have to do is write your name and address on a piece of paper, item(s) required, enclosing your cheque/P.O. made payable to SOFTWARE INVASION.

Please allow 7 to 14 days for delivery (orders are normally dispatched within 48 hours!)

OVERSEAS ORDERS. Please add £1.00 per game

DISTRIBUTORS, MULTIPLE STORES, RETAILER GROUP AND EXPORT INQUIRIES INVITED.

## SOFTWARE INVASION



50 Elborough Street, Southfields, London SW18 5DN

Do you write your own programs? If you have an unusual program which can meet our standards you could be earning a substantial sum every week! Why not take advantage of our sought after reputation. Write now!

other SOFTWARE INVASION GAMES... send for full list now!





The company which brought you the first self-build Arcade game and the first Adventure with sound, just had to be the one to give you the very first QUEST for the Beeb!

# Swords and Sorcery

a new experience...

QUEST programs originated on the American mainframe computers, and were converted to micros, though requiring so much memory could only be made to work on expanded Apple micros and the like. Now, Kansas have crammed a full feature Quest into 32K, giving every BBC Micro owner the opportunity to play and experience these unique games.

So what is a Quest? Think of an Adventure, then think of very much more. A Quest is operated similar to an Adventure, but no longer are you alone, starting with three helpers. As you progress you will encounter many other characters, but unlike an Adventure program were all characters are passive, these can be either hostile, friendly or indifferent. If hostile you have to fight, and quickly; if friendly they may join your team bringing more strength, magical ability and carrying capability; if indifferent you could perhaps try a bribe (but beware if you offer too little) or you could sell an object and raise yourself more cash as well as lessening your load.

But most of the effort is your own, with treasure, money, magic and strength all having a bearing on your progress. At times it will pay to be vicious and abandon your friends as they become weaker, or even go in fighting before characters show their true colours, the element of surprise using less strength.

As can be expected in Swords and Sorcery there is a story:

Once upon a time in a far away land called Iriuma a magical Princess cast a spell of banishment on the wicked Sorcerer Brogfelt. However, just before the spell took effect, Brogfelt threw his arms into the shape of changing and cast a return spell, changing princess Illear into a diamond. Taking the diamond with him, Brogfelt took sanctuary in the Dungeons of Terror amongst the various monsters and demi-humans. Here he split the diamond into four parts and changed each part into a different crystal. Brogfelt then hid each crystal in the dungeons never to be found again. When the king heard of this he summoned all his faithful Knights to him and offered half his lands for the four crystals. Sadly they all perished in the Dungeons of Terror attempting it. Now the king has offered anyone his other daughter's hand together with half his lands. Hearing this, you set off to the king's palace to offer your services. The king is astounded but nevertheless offers you a party of three prisoners from his jail, promising them a full pardon if they will go and aid you. As you leave the king stops you and thrusts a scrap of paper into your hand explaining that it contains the location of each part of the crystal from the entrance of the dungeons. It was written very shakily and stained in blood. The king says: "Let me introduce you to the three prisoners that I have volunteered to go along with you..."

The characters you meet include a Troll, Orc, Thief, Dwarf, Goblin, Madman, Witch, Hobgoblin, Mad Monk and of course the wicked wizard himself, all in fact you would expect in a magical Swords and Sorcery...

All the objects have a use, but be careful, for picking the Dragons Tooth could be fatal; though if you find the Staff of Healing try and get it; an Idol of a forgotten God should be left well alone; the Ring may help you; the Magic Axe certainly will; the Old Book will give some clues; rub the Glass Ball; used properly the Magic Carpet will get you out of trouble; the Rolled Scroll too is useful; but not so the Fools Gold; be careful with the Bottle of Liquid; but drink the Magic Potion; treat the Golden Orb with care; but of course the Sword is the greatest help of all. And so it goes on, and on and on...

Unlike an Adventure game, were once you have solved the plot, and it is all finished for good, Swords and Sorcery generates an entirely different scenerio every time, even though totally logical! But if you really intend to see a game right through to the end, the game saving facility allows you to do this, playing the same scenerio, time after time, to its end. It is so different from an Adventure, that it actually has nine—yes nine—levels of play, with the ninth having so many locations and of such complexity, that we would be amazed if anybody ever solved it!

If you are an Adventure addict, this will really spoil you! If you do not care for Adventures the activity in this unique game will suit you as well. It is one of those you just will not be able to leave alone...

Though there are many Adventure games using the word Quest in their title, these are not Quests in this, the true sense of the word, but are just ordinary Adventures.

FEATURES a Map routine showing exactly where you are and where you have explored.

FEATURES colour denoting the different aspects of the game.

FEATURES sound on or sound off facility.

FEATURES entirely different scenerio for every game and all totally logical

FEATURES nine levels of difficulty from fairly easy to impossible.

FEATURES game saving routine to be able to replay the same scenerio.

Only available from Kansas at £10.35 Vat and post paid.

Our normal return first class post service applies, with same day postage if ordered on Access or Barclaycard.

This program carries our famed lifetime guarantee.

Available on cassette only for BBC model B.



**Kansas City Systems, Unit 3, Sutton Springs Wood, Chesterfield, S44 5XF.**  
**Tel. 0246 850357**



## From Page 56

```

1950 FOR AZ=0 TO HAND_LEN%(60%)
1960 PROC_DETAILS(AZ,60%)
1970 B%((SUITX*2)+(CARDX*6))=B%((SUITX*2)+(CARDX*6))+1
1980 NEXT AZ
1990 MAX%=0
2000 FOR AZ=1 TO NUM_60%
2010 IF B%(AZ)>MAX% MAX%=B%(AZ)
2020 NEXT AZ
2030 FOR AZ=1 TO NUM_60%
2040 PHX=PHX(AZ):PROC_DETAILS(PHX,60%)
2050 IF (SUITX*2)+(CARDX*6)=MAX% OR (CARDX=7 AND (SUITX*2)=MAX%) AZ=9
2060 NEXT AZ
2070 =PHX
2080 DEF PROC_COMPUTER
2090 PRINT TAB(0,19)SPC(30)
2100 GO_POSX=FN_COMPUTER
2110 COLOUR 3
2120 IF GO_POSX=-1 PRINT TAB(8,19)"I could not go";B%=FN_DEAL(1):GOTO 2160
2130 PROC_DETAILS(GO_POSX,1)
2140 PRINT TAB(13-(LEN(CARD$)/2),19)"I played ";CARD$;
2150 PROC_PLAY(GO_POSX)
2160 COLOUR 128:COLOUR 3
2170 PRINT TAB(25,23);HAND_LEN%(1)+1;" "
2180 COLOUR 130
2190 AZ=INKEY(DELAY%)
2200 B0%=0
2210 ENDPROC
2220 DEF FN_END
2230 COLOUR 3
2240 IF HAND_LEN%(0)=-1 PRINT TAB(31,3)SPC(9);TAB(0,19)" Congratulation s, you win":SCORE0%=SCORE0%+1:SOUND 1,1,100,35:GOTO 2350
2250 PRINT TAB(0,19)SPC(8)"Bad luck I won"SPC(8):SCORE1%=SCORE1%+1
2260 FOR BX=8 TO 0 STEP -8
2270 RESTORE 2420
2280 FOR AX=1 TO 5
2290 READ NX,DZ
2300 IF AX=5 AND BX=0 DZ=30
2310 SOUND &101,2,NX+BX,DZ
2320 SOUND &102,2,NX+BX+48,DZ
2330 SOUND 1,0,0,2
2340 NEXT AX:NEXT BX
2350 COLOUR 128
2360 PRINT TAB(10+LEN(NAME$),0);SCOR E0%
2370 PRINT TAB(18,1);SCORE1%
2380 FOR AX=1 TO 8:CX(AZ)=0:NEXT AX
2390 COLOUR 130
2400 PRINT TAB(0,20)"Do you want to

```

```

play again(Y/N)"
2410 =FN_YES_NO
2420 DATA 4,10,4,10,12,15,4,5,0,20
2430 DEF PROC_PLAY(PLAY_POSX)
2440 PROC_DETAILS(PLAY_POSX,60%)
2450 VDU 5
2460 MOVE ((SUITX-1)*224)+((256-(LEN(CARD$)*32)/2))+16,440+(CARDX*32):PRINTCARD$
2470 VDU 4
2480 CX((SUITX*2)+(CARDX*6))=VLUX
2490 IF CARDX=7 CX(SUITX*2)=VLUX
2500 FOR AX=HANDX(60%)+PLAY_POSX TO HANDX(60%)+HAND_LEN%(60%)-1
2510 ?AX=AX?1
2520 NEXT
2530 HAND_LEN%(60%)=HAND_LEN%(60%)-1
2540 PROC_FLIP
2550 ENDPROC
2560 DEF PROC_FLIP
2570 REPEAT UNTIL ADVAL(-5)=15
2580 SOUND 0,-12,5,0
2590 SOUND 0,-13,4,0
2600 SOUND 0,0,0,6
2610 ENDPROC
2620 DEF PROC_DIS_HAND
2630 VDU 24,980;0;1279;940;:CLG
2640 FOR AZ=0 TO HAND_LEN%(0)
2650 PROC_DETAILS(AZ,0)
2660 COLOUR 3
2670 PRINT TAB(31-(AX*9),AX*3);AX+1;CHR$(228);CARD$
2680 NEXT AZ
2690 VDU 26
2700 ENDPROC
2710 DEF FN_YES_NO
2720 REPEAT
2730 A$=GET$
2740 UNTIL A$="Y" OR A$="N"
2750 =A$="Y"
2760 DEF PROC_ADVISE
2770 GO_POSX=FN_COMPUTER
2780 PROC_DETAILS(GO_POSX,0)
2790 PROC_COMMENT("I advise you to play "+CARD$,TRUE)
2800 ENDPROC
2810 DEF PROC_SWAP
2820 IF SZ=0 OR SNUMX=0 PROC_COMMENT("You cannot swap hands.",FALSE):ENDPROC
2830 TX=HANDX(0):HANDX(0)=HANDX(1):HANDX(1)=TX
2840 TX=HAND_LEN%(0):HAND_LEN%(0)=HAND_LEN%(1):HAND_LEN%(1)=TX
2850 COLOUR 128
2860 PRINT TAB(25,23);HAND_LEN%(1)+1;" "
2870 COLOUR 130
2880 PROC_DIS_HAND
2890 SNUMX=SNUMX-1

```

```

2900 COLOUR 128
2910 COLOUR 3
2920 PRINTTAB(21,24);SNUMX
2930 COLOUR 130
2940 ENDPROC
2950 DEF PROC_INSTRUCTIONS
2960 FOR AZ=0 TO 1
2970 VDU 31,0,AZ,134,157,132,141:PRINTSPC(10)"Sevens"
2980 NEXT AZ
2990 PRINT" The object of the game is to put all"
3000 PRINT" the cards in your hand down before the"
3010 PRINT" computer can. You may only lay one card";
3020 PRINT" down at a time, starting from seven and";
3030 PRINT" going up and down to King and Ace."
3040 PRINT" Your cards are displayed next to a"
3050 PRINT" number, and it is this number that you"
3060 PRINT" enter to play the card. If you cannot"
3070 PRINT" go type ""C"" and you will be dealt a"
3080 PRINT" card."
3090 PRINT" If you want advice on which card to"
3100 PRINT" play type ""A"" and you will be told."
3110 PRINT" If you wish, you can choose to swap"
3120 PRINT" your hand with the computer, but only"
3130 PRINT" two times during the game."
3140 *FX15
3150 *FX4 1
3160 VDU 23;10,96,0;0;0;
3170 PRINTTAB(1,20)"Please type in your name (<19 letters)"
3180 INPUT TAB(1,21)SPC(40)TAB(1,21)NAME$
3190 NAME$=LEFT$(NAME$,18)
3200 PRINTTAB(1,20)"Please type how long messages are to be";
3210 PRINT" displayed for (in seconds) "
3220 INPUTTAB(28,21)SPC(12)TAB(28,21)A$
3230 DELAYX=VAL(A$)*100
3240 PRINTTAB(1,20)"Do want to be able to swap your hand ";
3250 PRINT" with the computer. (Y/N)"
3260 SZ=FN_YES_NO
3270 ENDPROC

```



**THIS** version of cribbage pits one player against the BBC Micro. For those unfamiliar with crib, the object of the game is to "peg" your way round a board by scoring points. There are two main ways to score.

Firstly you can score points as you and your opponent play your cards alternately – this is the play.

Secondly, after the play, you lay down your hand and gain points according to the various scoring combinations you hold – this is the lay.

All cards are worth their face value with jack, queen and king counting ten. Ace is low.

Play alternates between player and computer, with the dealer of the first game being picked at random. In practice the micro deals the card.

However the deal is considered to alternate between micro and player, the non-dealer having to lay the first card in a game.

The player is dealt six cards, from which he has to retain four for his hand. The two cards he discards are thrown into the "box" or "crib". These, together with the computer's discards, form an extra hand.

This hand is scored at the lay and the points awarded to the dealer – that is, this extra hand alternates between player and micro.

Thus when selecting discards you have to bear in mind whose box it is. You must also try to ensure that the hand you are left with is balanced enough to provide points in both play and lay.

Once the discards have been made a starter card is dealt from the pack. This is the "turn up". If it's a jack, the dealer scores two points. This turn up plays no further part in the play, but is considered to be an integral part of the hand during the lay.

The non-dealer then plays a card followed by the dealer, the total points value of the cards being added. Play continues alternately until the total is either exactly 31, or neither player can play a card without exceeding that value – or until all the cards in the hand have been played.

If, after your turn, your opponent cannot go, you are still entitled to if you can do so legally, scoring as normal.

If a position is reached where neither player can go, that round of play is finished and a new one started with the remaining cards, the player who first failed to play a card going first.

The last player to play a card in a round scores points: two if he reaches 31 exactly, otherwise one. However

# Play Cribbage

## Peg your way round ALAN FARMAN's version of the classic card game

there are also other points to be won during the play.

You score two points if you bring the total to 15, or if your card forms a pair with the previous card. If your card forms three the same you score six, and all four identical scores 12.

Also, if the card you play forms a run of three or more (including your opponent's cards), you score one for each card in the run. Note that they don't have to be in order. If you play a four then the micro plays a six and you then play a five, you score three for the

run, even though it's 4, 6, 5.

Incidentally, if your four had been the first card in the round, you would score five, since the total would be 15 – you score two for that and three for the run. All you have to hope is that the micro doesn't have a three – then again you might have a two to follow it...

When all the cards have been played they are laid out and the cards in each hand – together with the turn up – are scored.

You score two for every combination of 15 you can find in your hand, and

## PROCEDURES

**PROCINIT**  
**PROCTAPE**  
**PROCget**  
**PROCT**  
**PROC\_vdu\_chars**  
**PROCplayname.**  
**PROCBOARD**  
**PROCDEALER**  
**PROC\_varis**  
**PROCSHUFFLE**  
**PROCCARDS**  
**PROCALLOCATE**

**PROCSTARTER**  
**PROCSTARTERBACK**  
**PROCBOXCARD**  
**PROCWHOLAY**  
**PROCGAME**  
**PROCCARDFRONT**  
**PROCCARDBACK**  
**PROCUP & PROCDOWN**  
**PROCNUMBER**

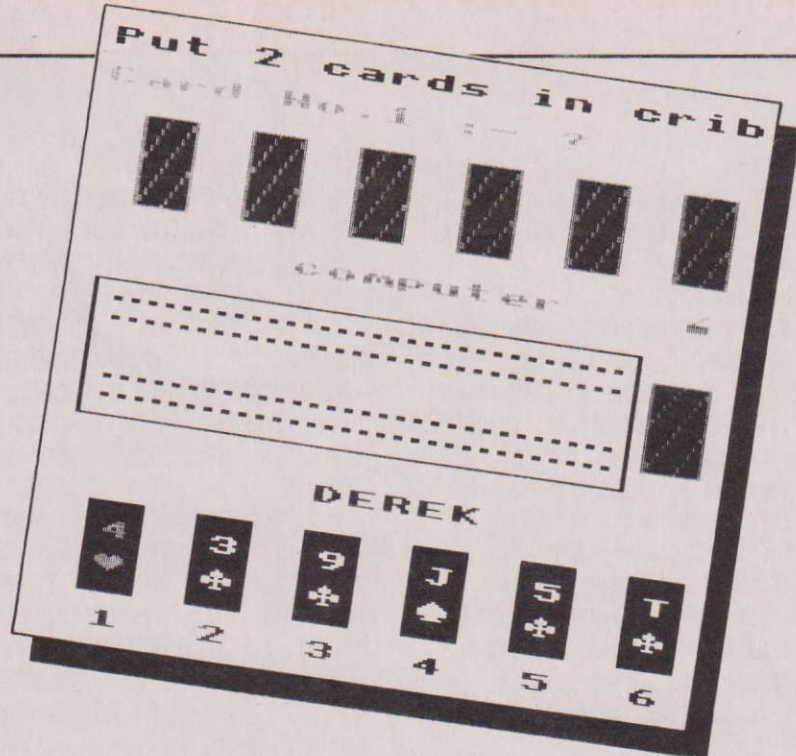
**PROCsound**  
**PROCPACK**

**PROCPLAYERCARD**  
**PROCcardcol**  
**PROCCPBEST &**  
**PROCCPSECBT**

**PROCcompse &**  
**PROCplayse**  
**PROCGAMETOT**

Rules.  
Chains game.  
Pause control.  
Time delay.  
Sets up characters.  
Player's name.  
Draws board.  
Randomly decides first dealer.  
Sets up variables.  
Shuffles cards.  
Draws cards on screen.  
Allocates CARDS type to numeric.  
CARD% for calculation purposes.  
Draws starter or crib card.  
Draws starter card back.  
Allocates the four cards to the crib.  
Sorts out who is to lay first card.  
Plays the game.  
Draws front of card.  
Draws back of card.  
Moves box character about screen.  
Converts random number into card number used when sorting out dealer.  
Makes a sound.  
Reads 52 cards into an array and allocates suit.  
Draws player's cards on screen.  
Sorts out player's suit colour.  
Sorts out computer's best or second best cards according to whether Level 1 or 2 is being played.  
Draws scores on screen for both computer and player as play proceeds.  
Displays game score.





similarly for every pair. Three of a kind count six, and four count 12. Runs of three or more count one for each card involved.

If all four cards in your hand are the same suit you score four for the flush.

(In the box, you need all four cards in the hand plus the turn up to be of the same suit for a flush – and you score five for it.)

Also, if you have a jack in your hand of the same suit as the turn up, you score

“one for his nob”.

The non-dealer scores his lay down first, followed by the dealer, who then also totals his box. This is quite important, as it lets the non-dealer have first chance of pegging out – that is, pegging his way around the scoring board twice.

There are two levels of play. At level one, the computer has limited search facilities and this level should be used to learn the game.

At level two a full search of any four combinations from six cards is carried out, making it suitable for the more experienced player.

#### PROCCOMPCHK

Part of computer's search routine to find best cards.

#### PROC(S)-FIVECARD, FOURCARD, THREECARD, TWO CARD, SAME, PAIR, JACK

These procedures search for runs, 15s, flushes, pairs etc.

#### PROCSWOP

Swops computer/player cards to enable search for runs.

#### PROCKEEP

Decides which four cards computer should keep.

#### PROCCOMPGAMECARD PROCCOMPFIRSTMOVE

Computer plays with these cards.  
Plays computer's first move (only if computer goes first).

#### PROCPLAYERMOVE PROCCOMPNEXTMOVE PROCODSANS PROCPLAYCARDGONE/ COMPCARDGONE

Plays player's moves.  
Plays computer's subsequent moves.  
Clears up a bit of logic.  
Removes cards from game screen.

#### PROCFACECARD PROCPLAYFIRST/ COMPFIRST

Prints card face.  
Decides who scores first, displays cards again and scores.

#### PROCCLEARUP PROCLAY PROCPLAYCARDHAND

Lays a card and sorts out specific variables.  
Lays a card in CLEARUP above.  
Re-allocates player's cards before game commences; enables player's scoring to be carried out by the computer.

#### PROCcorner

Prints arrow character when scoreboard turns corner (useful on B/W sets).

#### PROCWINNER PROCBOX PROCPLCOSCCHECK PROCCOMPBOX PROCscore

Declares winner.  
Displays boxcards.  
Part of score checking routine.  
Sorts out boxcards.  
Shows total score held by each player at the end of each hand.

#### PROCblank\_score

Removes total score from game board.

```

10 REM *** CRIBBAGE ***
20 REM * Alan Farman *
30 REM *(c) Micro User*
40 REM *** 1984 ***
50 DN ERROR GOTO 2380
60 DIM CD$(18),BOX$(5),CDX(18),CHC
DX(18),KP$(3),X1$(4),X2$(4),Gone$(6),
run$(9):MODE7:PROCVD:PROCMD:seed%=(RN
D(-TIME)/LEN(NAME$))
70 MODE5:PROCBD:PROCCL:CYM%=976:PX
M%=976:CYM%=600:PYM%=472:REPEAT:PROCV
A:PROCSH:PROCCD(1):PROCAL(1,7):PROCS
T(" "):PROCSB:PROCB0:PROCW:PROCGM:UNT
IL FALSE
80 DEF PROCVD:VDU23,226,16,32,126,
33,17,1,1,126,23,227,0,0,60,36,36,60,
0,0,23,228,0,126,66,66,66,66,126,0,23
,229,255,129,129,129,129,129,255,
23,230,2,4,8,16,62,62,62,62,23,231,54
,127,127,127,62,28,8,0
90 VDU23,232,8,28,62,127,127,127,2
8,62,23,233,8,28,62,127,62,28,8,0,23,
234,8,28,28,107,127,107,8,28:HEART$=C
HR$231:SPADE$=CHR$232:DIAMOND$=CHR$23
3:CLUB$=CHR$234:ST$=STRING$(59," "):S
T1$=STRING$(19," "):ENDPROC
100 DEF PROCVA:nexcd%=0:plcdpo%=1:c
pnogo%=0:plnogo%=0:fiftn%=0:pair%=0:j
ack%=0:co%=0:pl%=0:npair%=0:newno%=31
:n%=0:point%=0:cpfin%=0:plfin%=0:plco
nt%=0:cpprecd%=0:plprecd%=0:total%=0:
corp%=1:ftpr%=0:twoetc%=0:thiscd%=0:l
aid%=0
110 gotrun%=0:plcorun%=0:apair%=0:c
opr%=0:plpr%=0:FORI%=1TO6:Gone$(I%)=0
:NEXT:VDU28,0,3,19,0,23,11,0,0,0,0:EN
DPROC
120 DEF PROCCF(X%,Y%):VDU25,4,X%,Y%
;25,5,X%;(Y%+150);25,5,(X%+100);(Y%+1
50);25,85,X%;Y%;25,5,(X%+100);Y%;25,8
5,(X%+100);(Y%+150);:ENDPROC

```



## From Page 61

```

130 DEF PROCB:VDU5,18,0,3:FORX=44
OTD600STEP32:FORX=48TO97&STEP32:IFYX
<=472 ORYX>=568THENMOVEX,YX:PRINT" "
140 NEXT:VDU18,0,3,25,4,32;384
;25,5,32;608;25,5,1040;608;25,5,1040;
384;25,5,32;384;25,4,384;688;18,0,1:P
RINT"computer":VDU18,0,2:MOVE 640-(L
EN(NAME$)/2)*64,320:PRINT NAME$:VDU1
8,0,3,4:ENDPROC
150 DEF PROCST(N$):VDU5,18,0,3,25,4
,1079;421;25,5,1079;571;25,5,1179;571
;25,85,1079;421;25,5,1179;421;25,85,1
179;571;25,4,1100;504;18,0,0:PRINT N$
:VDU18,0,3,4:ENDPROC
160 DEF PROCCK(X,Y):VDU5,18,0,1,2
5,4,X,Y;25,5,(X+100);(Y+150);25,5
,X,Y;(Y+150);25,5,X,Y;25,5,(X+100)
;Y;25,5,(X+100);(Y+150);25,4,(X+5
0);(Y+150);25,5,X,Y;(Y+75);25,4,(X+
100);(Y+75);25,5,(X+50);Y;18,0,3,4
:ENDPROC
170 DEF PROCDL:PROCST(" ");PROCSB:C
OLOUR3:PRINT TAB(0,1)"First Jack for
crib":VDU7:PROCT(1.5):REPEAT:PROCP:P
ROCNU:PROCST(N$)
180 IFN$="J"THENPROCT(.5):deal%=1:D
EALER$="computer":GOTO210ELSEPROCT(.5
)
190 PROCDN:PROCNU:PROCST(N$)
200 IFN$="J"THENPROCT(.5):deal%=-1:
DEALER$=NAME$:GOTO210ELSEPROCT(.5)
210 UNTILN$="J":PROCSd:PROCST(" ");
PROCSB:PRINTTAB(0,0):ST$:COLOUR1:PRIN
TTAB(10-(LEN(DEALER$)/2),0):DEALER$:C
OLOUR2:PRINTTAB(0,2)"has the first c
rib!":PROCT(4):ENDPROC
220 DEF PROCSB:PROCCK(1079,421):END
PROC
230 DEF PROCNU:J%=RND(13):IFJ%=1THE
NN$="A"
240 IFJ%>1ANDJ%<10THENN$=STR$(J%)
250 IFJ%=10THENN$="T"
260 IFJ%=11THENN$="J"
270 IFJ%=12THENN$="Q"
280 IFJ%=13THENN$="K"
290 ENDPROC
300 DEF PROCAL(a%,b%):FORN=a%TOb%:
IFRIGHT$(CD$(N%),1)="A"THENCDC$(N%)=1:
CHCD$(N%)=1
310 IFRIGHT$(CD$(N%),1)="T"THENCDC$(
N%)=10:CHCD$(N%)=10
320 IFRIGHT$(CD$(N%),1)="J"THENCDC$(
N%)=10:CHCD$(N%)=11
330 IFRIGHT$(CD$(N%),1)="Q"THENCDC$(
N%)=10:CHCD$(N%)=12
340 IFRIGHT$(CD$(N%),1)="K"THENCDC$(
N%)=10:CHCD$(N%)=13
350 IFVAL(RIGHT$(CD$(N%),1))>1ANDVA

```

```

L(RIGHT$(CD$(N%),1))<10THENCDC$(N%)=VA
L(RIGHT$(CD$(N%),1)):CHCD$(N%)=VAL(RI
GHT$(CD$(N%),1))
360 NEXT:ENDPROC
370 DEF PROCSH:PRINTTAB(0,0):ST$:COL
OUR1:PRINTTAB(2,0)"- Please wait -":C
OLOUR2:PRINTTAB(2,2)"while I shuffle"
;VDU7,18,0,0:PROCCF(1079,421):PROCT(
1):NNX=1:REPEAT
380 copy=0:J%=1:t%=0:T%=RND(52):RES
TORE:REPEAT:READPK$:t%=t%+1:UNTILT%=T
%:CS$=LEFT$(PK$,1):CN$=RIGHT$(PK$,1)
390 IFCS$="H"THENPK$=HEART$+CN$
400 IFCS$="S"THENPK$=SPADE$+CN$
410 IFCS$="D"THENPK$=DIAMOND$+CN$
420 IFCS$="C"THENPK$=CLUB$+CN$
430 REPEAT:CD$(NNX)=PK$:IFJ%=NNX%THE
N450
440 IFCD$(NNX)=CD$(J%)THENcopy=1:UN
TILcopy=1:GOTO380
450 J%=J%+1:UNTILJ%>13:NNX=NNX+1:UN
TILNNX>13:PRINTTAB(0,2):ST$:COLOUR3:P
RINTTAB(4,2)"while I deal":VDU7:PROC
T(1):ENDPROC
460 DATA D9,H4,CJ,ST,CT,D6,S5,S4,CA
,SA,D3,S9,HK,HQ,H8,D8,S6,HA,C9,DT,DK,
S3,SK,H9,H7,DA,SJ,C3,SQ,D5,H3,H2,C4,H
J,H5,D2,C5,HT,C7,S8,H6,C6,C8,S7,C2,D4
,S2,CK,CQ,DQ,D7,DJ
470 DEF PROCPC(XX):YY%=214:VDU5:PR
OCcd(plcdp%):GCOL0,CX:MOVEXX+20,YY%
-56:PRINT;LEFT$(CD$(plcdp%),1):MOVEX
X+25,YY%:PRINT;RIGHT$(CD$(plcdp%),1
):GCOL0,2:MOVEXX+25,64:PRINT;plcdp%
:plcdp%=plcdp%+1:VDU4:ENDPROC
480 DEF PROCB:PRINTTAB(0,0):ST$:COL
OUR3:PRINTTAB(0,0)"Put 2 cards in cri
b":COLOUR1:VDU7:PRINTTAB(0,2)"Card No
.1 :- ":INPUTNo1:IFNo1<10RNo1>6THEN4
80
490 VDU5,18,0,0:y%=96:p1x%=(No1*20
0)-200)+80:PROCCF(p1x%,y%):VDU25,4,(p
1x%+80);64;127,4
500 PRINTTAB(0,2):ST1$:COLOUR2:VDU7:
PRINTTAB(0,2)"Card No.2 :- ":INPUTNo
2:IFNo2=No1 ORNo2<1 ORNo2>6THENPRINTT
AB(0,2)STRING$(19," "):GOTO500
510 VDU5,18,0,0:y%=96:p2x%=(No2*20
0)-200)+80:PROCCF(p2x%,y%):VDU25,4,(p
2x%+80);64;127,4,17,1
520 Gane%(No1)=99:Gane%(No2)=99:PRI
NTTAB(0,0):ST$:PRINTTAB(0,0)"Please wa
it while I":COLOUR2:PRINT:PRINT" thro
w mine in too":PROCCP
530 BOX$(1)=CD$(No1):BOX$(2)=CD$(No
2):BOX$(5)=CD$(7):CD$(18)=CD$(7):CD$(
18)=CD$(7):CHCD$(18)=CHCD$(7)
540 PRINTTAB(0,0):ST$:PRINTTAB(1,1)"
OK, I have chosen.":VDU7,5,18,0,0:y%=
736:c1x%=(NUM1*200)-200)+80:PROCCF(c

```

```

1x%,y%):c2x%=(NUM2*200)-200)+80:PROC
CF(c2x%,y%):PROCT(1):VDU4:ENDPROC
550 DEF PROCVH:IFdeal%<0-1THENSTART
$=NAME$ ELSESTART$="computer"
560 PRINTTAB(0,0):ST$:VDU7:COLOUR2:P
RINTTAB(4,1)"Starter card":PROCT(1):
VDU5,18,0,3:PROCTST(" ");VDU5,25,4,110
0;536;IFLEFT$(CD$(7),1)=HEART$ ORLEF
T$(CD$(7),1)=DIAMOND$THENGCOL0,1 ELSE
GCOL0,0
570 PRINTRIGHT$(CD$(7),1):VDU25,4,1
095;488;PRINTLEFT$(CD$(7),1):VDU4:PR
INTTAB(0,0):ST$:IFCHCD$(7)=11THENPROC
s:PRINTTAB(0,0):ST$:PRINTTAB(0,1)"Two
for his heels!":PROCT(2):jack%=1
580 IFdeal%=1ANDjack%=1THENPROCCs(2
) ELSE IFdeal%=TRUE ANDjack%=1THENPRO
Cps(2):jack%=0
590 VDU7,5,18,0,2,25,4,80;512;PRIN
T"Game total=":PROCGT:VDU4:ENDPROC
600 DEF PROCGT:VDU5,18,0,0,25,4,920
;512;127,127,18,0,1,25,4,800;512;PRI
NT;total%:VDU4:ENDPROC
610 DEF PROCSE:A%=8:B%=11:FORQX=1TO
3:PROCAL(A%,B%):PROCSW(A%,B%):PROCCC(
QX+7,QX+10):KP$(QX)=sc$:PROCAL(A%,B%
):A%=A%+1:B%=B%+1:NEXT:PROCKP:IFkp%=0T
HENkp%=RND(3)
620 IFkp%=1THENBOX$(3)=CD$(12):BOX$(
4)=CD$(13):NUM1=5:NUM2=6:A%=8:B%=11
630 IFkp%=2THENBOX$(3)=CD$(8):BOX$(
4)=CD$(13):NUM1=1:NUM2=6:A%=9:B%=12
640 IFkp%=3THENBOX$(3)=CD$(8):BOX$(
4)=CD$(9):NUM1=1:NUM2=2:A%=10:B%=13
650 ENDPROC
660 DEF PROCT(t):FORT=1TO1500*t:NEX
T:ENDPROC
670 DEF PROCCC(B%,TX):FVR=0:FRR=0:s
c%=0:PROCFR(B%,TX):PROCTH(B%,TX):PROC
TW(B%,TX):PROCPR(B%,TX):PROCSM(B%,TX)
:ENDPROC
680 DEF PROCFV(B%,TX):LOCALHX,IX,J%
,K%,L%:FORHX=B%TOTX-4:FORIX=H%+1TOTX-
3:FORJX=IX+1TOTX-2:FORKX=JX+1TOTX-1:F
ORLX=KX+1TOTX
690 IFCHCD$(HX)+1=CHCD$(IX)ANDCHCD$(
IX)+1=CHCD$(JX)ANDCHCD$(JX)+1=CHCD$(
KX)ANDCHCD$(KX)+1=CHCD$(LX)THENsc%=sc
%+5:FVR=1:FRR=1
700 IFCD$(HX)+CD$(IX)+CD$(JX)+CD$(K
X)+CD$(LX)=15THENsc%=sc%+2
710 NEXT:NEXT:NEXT:NEXT:NEXT:ENDPRO
C
720 DEF PROCFR(B%,TX):LOCALHX,IX,J%
,K%,FORHX=B%TOTX-3:FORIX=H%+1TOTX-2:F
ORJX=IX+1TOTX-1:FORKX=JX+1TOTX
730 IFFVR=0THEN IFCHCD$(HX)+1=CHCD$(
IX)ANDCHCD$(IX)+1=CHCD$(JX)ANDCHCD$(
JX)+1=CHCD$(KX)THENsc%=sc%+4:FRR=1
740 IFCD$(HX)+CD$(IX)+CD$(JX)+CD$(K

```



```

%)=15THENS%=sc%+2
750 NEXT:NEXT:NEXT:NEXT:ENDPROC
760 DEF PROCTH(B%,T%):LOCALH%,I%,J%
:FORH%=B%TOT%-2:FORI%=H%+1TOT%-1:FORJ
%=I%+1TOT%
770 IFFRR=0THEN IFCHCD%(H%)+1=CHCD%
(I%)ANDCHCD%(I%)+1=CHCD%(J%)THENS%=s
c%+3
780 IFCD%(H%)+CD%(I%)+CD%(J%)=15THE
Nsc%=sc%+2
790 NEXT:NEXT:NEXT:ENDPROC
800 DEF PROCTW(B%,T%):LOCALH%,I%:FO
RH%=B%TOT%-1:FORI%=H%+1TOT%
810 IFCD%(H%)+CD%(I%)=15THENS%=sc%
+2
820 IFtotal%+CD%(H%)=15THENiftn%=H
%
830 IFtotal%+CD%(I%)=15THENiftn%=I
%
840 NEXT:NEXT:ENDPROC
850 DEF PROCPR(B%,T%):LOCALH%,I%:FO
RH%=B%TOT%-1:FORI%=H%+1TOT%
860 IFCHCD%(H%)=CHCD%(I%)THENS%=sc
%+2:ftpr%=H%
870 IFCHCD%(H%)=CHCD%(n%)THENpair%=
H%
880 IFCHCD%(I%)=CHCD%(n%)THENpair%=
I%
890 NEXT:NEXT:ENDPROC
900 DEF PROCSM(B%,T%):same%=0:LOCAL
I%:FORI%=B%TOT%-2:IFLEFT$(CD$(I%),1)=
LEFT$(CD$(I%+1),1)THENSame%=same%+1
910 NEXT:IFsame%=3THENS%=sc%+4
920 IFsame%=3ANDLEFT$(CD$(T%),1)=LE
FT$(CD$(B%),1)THENS%=sc%+1:ENDPROC E
LSEENDPROC
930 DEF PROCJK(B%,T%):LOCALI%:FORI%
=B%TOT%-1
940 IFRIGHT$(CD$(I%),1)="J"ANDLEFT$(
CD$(T%),1)=LEFT$(CD$(I%),1)THENS%=s
c%+1
950 NEXT:ENDPROC
960 DEF PROCsd:FORs=50TO100:SOUND1,
-10,S,.7:NEXT:ENDPROC
970 DEF PROCSW(b%,t%):LOCALM%,N%:FO
RM%=b%TOT%-1:FORN%=M%+1TOT%:IFCHCD%(M
%)<=CHCD%(N%)THEN990
980 temp%=CHCD%(M%):CHCD%(M%)=CHCD%
(N%):CHCD%(N%)=temp%
990 NEXT:NEXT:ENDPROC
1000 DEF PROCkp:nk%=0:kp%=0:FORI%=1T
O3:IFKP%(I%)>nk%THENNk%=KP%(I%):kp%=I
%
1010 NEXT:ENDPROC
1020 DEF PROCSM:PROCAL(A%,B%):PROCCG
(A%,B%):laid%=1:IFdeal%=-1THENPROCFM:
laid%=2 ELSEPROCPM:laid%=2:PROCNM:lai
d%=3
1030 PROCPM:npair%=0:IFplnogo%=0THEN
laid%=laid%+1:IFlaid%>3THENPROCn:IFg

```

```

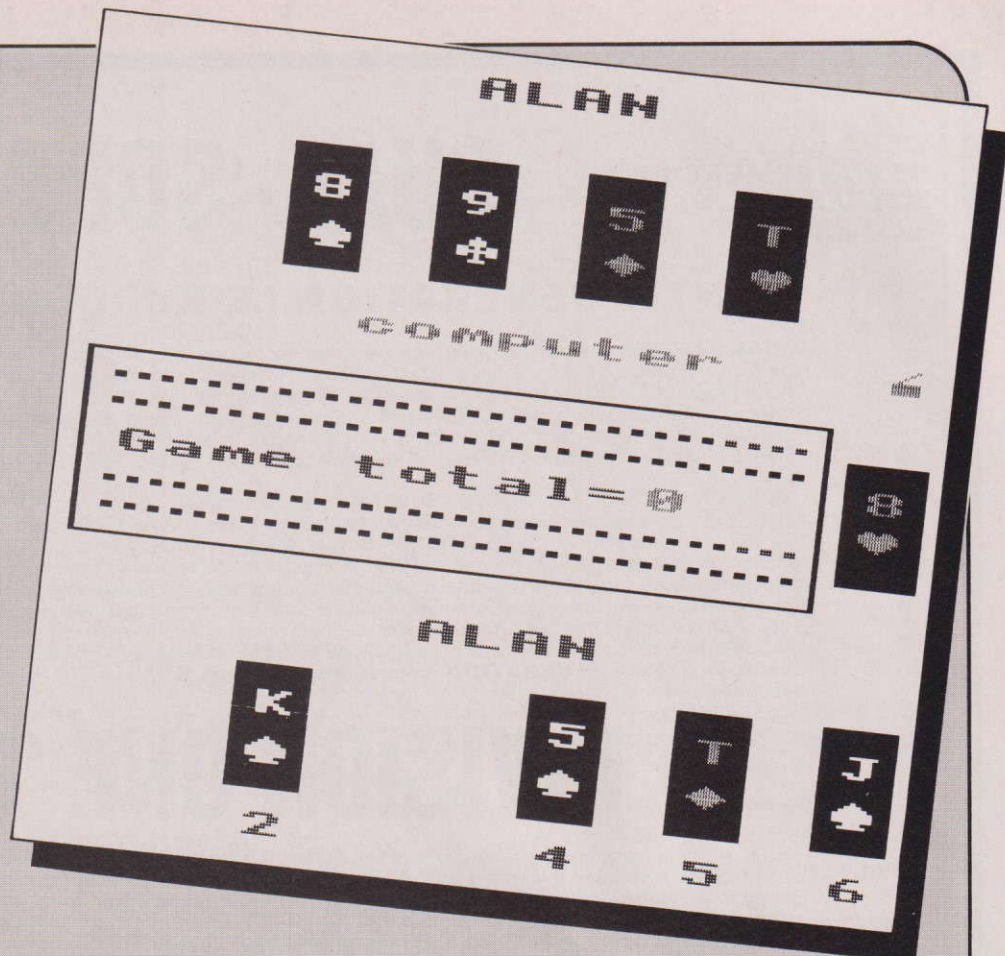
otrun%>3ANDgotrun%>plcorun%THENPROCp
s(gotrun%):plcorun%=gotrun%
1040 IFtotal%=31THENCorp%=2:GOTO1140
1050 IFcpnogo%=0ANDplnogo%=1THENCorp
%=1:GOTO1070
1060 IFtotal%<31PROCCD:IFpoint%=1THE
NPROCps(1):point%=0:GOTO1140
1070 PROCNM:npair%=0:IFcpnogo%=0THEN
laid%=laid%+1:IFlaid%>3THENPROCn:IFg
otrun%>3ANDgotrun%>plcorun%THENPROCc
s(gotrun%):plcorun%=gotrun%
1080 IFtotal%=31THENCorp%=1:GOTO1140
1090 IFpl%=4ANDcpnogo%=1THEN1060
1100 IFplnogo%=0ANDcpnogo%=1THENCorp
%=2:GOTO1120
1110 IFtotal%<31PROCCD:IFpoint%=1THE
NPROCcs(1):point%=0
1120 IFplnogo%=0ANDpl%<4THEN1030
1130 IFcpnogo%=0ANDco%<4THEN1070
1140 PROCPG:PROCGN:plnogo%=0:cpnogo%
=0:plcorun%=0:laid%=1:apair%=0:copr%
=0:plpr%=0:CHCD%(n%)=0:npair%=1:total%
=0:PROCGT:FORC%=1TO8:run%(C%)=0:NEXT
1150 IFpl%=4ANDco%=4THENCorp%=3
1160 ONCorp% GOTO1030,1070,1170
1170 PROCPG:PROCGN:COLOUR3:PRINTTAB(
0,0)ST$:PRINTTAB(3,0)"All cards laid"
:COLOUR1:PRINTTAB(5,2);"- scores -":V
DU7:PROCT(2):PRINTTAB(0,0)ST$:plcdpo%
=1:PROCCD(0):GOTO10
1180 PROCCF(plx%,96):VDU5,25,4,(plx%
+80);64;127:PROCCF(p2x%,96):VDU25,4,(
p2x%+80);64;127,4:PROCCF(14,848):PROC
T(1):IFdeal%=1THENPROCCP ELSEPROCCT
1190 total%=0:PROCCB:PRINTTAB(0,0)ST

```

```

$:COLOUR3:PRINTTAB(0,1)"Next game sta
rting":VDU7:PROCCB:VDU5,18,0,0:PROCC
F(1079,421):VDU4:IFdeal%=-1THENPROCCP
ELSEPROCCD
1200 PROCT(2):deal%=-deal%:ENDPROC
1210 DEF PROCCG(a%,b%):IFa%=8 c%=6:O
%=13
1220 IF a%=9 c%=5:O%=12
1230 IF a%=10 c%=4:O%=11
1240 FORI%=a%TOb%:CD$(I%+c%)=CD$(I%
):CD%(I%+c%)=CD%(I%):CHCD%(I%+c%)=CHCD
%(I%):NEXT:ENDPROC
1250 DEF PROCFM:newno%=newno%+1:COLO
UR1:VDU7:PRINTTAB(0,0)ST$:PRINTTAB(1,
1)"Computer to move":PROCT(1):check%
=0:first%=14:REPEAT:IFCD%(first%)<5TH
ENrun%(laid%)=CHCD%(first%):PROCCL(fi
rst%):check%=1:first%=17
1260 first%=first%+1:UNTIL first%>17
:IFcheck%=1THENENDPROC
1270 ftpr%=0:PROCPR(14,17):IFftpr%>0
THENrun%(laid%)=CHCD%(ftpr%):PROCCL(f
tpr%):ENDPROC
1280 LX=RND(4)+13:run%(laid%)=CHCD%(
L%):PROCCL(L%):ENDPROC
1290 DEF PROCNM:IFco%=4 OR cpnogo%=1
THENENDPROC
1300 COLOUR1:VDU7:PRINTTAB(0,0)ST$:P
RINTTAB(1,1)"Computer to move":PROCT
(1):newno%=newno%+1:nexcd%=0:iftn%=0
:pair%=0:thiscd%=cpnogo%:prdrp%=0:IF
total%<15THENPROCTW(14,17) ELSE1330
1310 IFiftn%<>0THENrun%(laid%)=CHCD

```





DEPT. CG1,  
18 HAZELMERE RD,  
STEVENAGE, HERTS  
SG2 8RX.

# QUAL-SOFT

STEV (0438) 721936

## NEW GENERATION LEISURE SOFTWARE

**QUAL-SOFT comments:** In 1982/83 it was only necessary to throw together a few M/C code routines, record them on a cassette, produce a flashy cassette insert, and it sold. But as we approach 1985, BBC users have become much more discriminating in their choice of software. Customers demand something more satisfying than a few hours of high adrenaline action. They demand, not hours, but tens of hours, even a hundred hours plus, of "playability". They demand that the program challenges their intellect and not just their reflexes. They demand **QUALITY** games. For the 84/85 season; for these customers, QUAL-SOFT is proud to offer these games:

"DIVISION ONE '85" and "SOCCER SUPREMO" are direct descendants of the 1983 game "LEAGUE DIVISION ONE", but they are a significant advance on what was described as "by far the best game for the BBC Micro". They are soccer management **SIMULATIONS!** Soccer management is about **ASSESSMENT**; assessment of the opposition strengths, of your players and their skills, of the effectiveness of the teams you assemble, of the value of transfer listed players, of the potential of your own youngsters. Soccer management is about **EXPERIMENTATION**; experimentation with players, with blends of players, with team formations, with tactical ploys. QUAL-SOFT's unique soccer match simulation, with tackles, passes, shots, saves etc. all simulated within your micro, and the results displayed in graphical form on your screen, allows **ASSESSMENT** and **EXPERIMENTATION** to be carried out. **The result is a 2ND GENERATION soccer management SIMULATION, not an exercise in mental arithmetic!!!**

BBC 'B'  
(all OS)

## DIVISION ONE '85'

TAPE £9.95  
DISC £12.95  
(all inc.)

This is an uncompromising strategy/tactics simulation, it will test to the full your knowledge of the game and your ability to use it to build a winning side. As described above, your ability to experiment and judge the results are paramount. Injuries, suspensions, long term "form" changes, short term "form" fluctuations, will all conspire to confuse and defeat you. You will also meet the problems of financial limitations as you choose between the 33 players available to you. You have a contract for 5 seasons, which will be terminated if your side is relegated. This game is the supreme test of your managerial skills, short of the real thing.

ELECTRON  
BBC 'B' (all OS)

## SOCCER SUPREMO

TAPE £9.95  
DISC £12.95  
(all inc.)

Soccer Supremo has been improved over the original game in the direction of a much more realistic "3D", 22 MAN, FULL PITCH match simulation for each of your games. The game is still a significant challenge to your managerial skills, but with greater emphasis on enjoyment and entertainment. But with 30 players to choose from, restricted finances, and just a 5 season contract, the 1st Division Championship is not an easy task.

**COMBO OFFER.** Telephone conversations, and orders, have shown that customers find it very difficult to choose between the two games. Our COMBO tape/disc offers you both games for just the price of one plus £3. £12.95 or £15.95.

**PREVIOUS OWNERS.** Send in your ORIGINAL tape/disc of LEAGUE DIVISION ONE, with manuals, and receive £7 for your tape or £9 for your disc, discount against any order for the above game(s).

BBC 'B'  
(all OS)

## PORTFOLIO

TAPE £9.95  
DISC £12.95  
(all inc.)

The most popular indoor games have always been multi-player board games, and yet designers have insisted on producing single player "me and my micro" computer games. In the USA the balance has already shifted away from the "arcade mentality" towards multi-player programs. **PORTFOLIO** is an investment game for 1-4 players, or teams of players. Each player/team begins with £1,000 and invests his money in a variety of companies against a background of continually changing international, national and commercial news. The news items affect the share values of the companies in a variety of different ways, and the players must shift their money around to maximise their investments; maybe even to become a millionaire.

The game is an ideal family game, or perfect in the classroom with a high pupil/micro ratio. It's suitable for 12 years old and above, though younger do enjoy the game with some help from their friends, and in this game, competitors. For the adult we have made some of our news items somewhat cryptic, occasionally even red herrings, to really make them think about the effect on share values. Because we call it a family game we have tried to see that there is something in it for all members of the family. Teachers can even tailor the news items to suit the capabilities of the pupils.

So why not set your Beeb on the coffee table, connect it to the TV set and play against the rest of the family or your friends on these cold, wet winter nights that are on us. It might even bring back the art of conversation (as against BASIC statements).

All orders **DIRECT SALE**.  
Orders by post delivered  
in 3/4 working days.  
ACCESS telephone  
authorisation 1/2 days.

TAPE	DISC
£9.95 <input type="checkbox"/> DIVISION ONE '85'	<input type="checkbox"/> £12.95
£9.95 <input type="checkbox"/> SOCCER SUPREMO	<input type="checkbox"/> £12.95
£12.95 <input type="checkbox"/> COMBO, DIV.1+SS	<input type="checkbox"/> £15.95
£9.95 <input type="checkbox"/> PORTFOLIO	<input type="checkbox"/> £12.95
Soccer Supremo: ELECTRON or BBC	

Name: .....  
Address: .....  
.....  
.....  
Card No: .....



## From Page 63

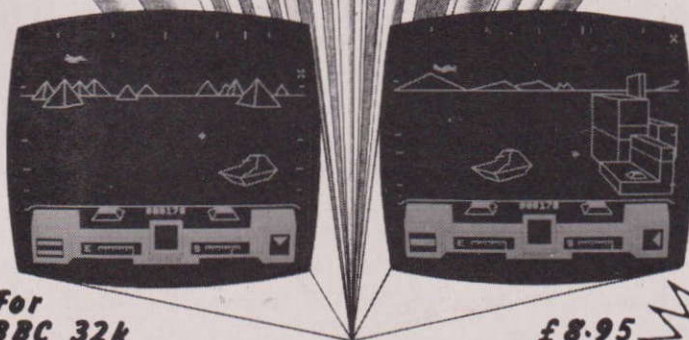
```
% (fiftn%):PROCC (fiftn%):PROCCs(2) EL
SE1330:IFCD$(fiftn%)=CD$(n%) THENPROCC
s(2)
1320 GOTO1430
1330 PROCPR(14,17):IFpair%(>0)THEN IF
total%+CD$(pair%)<=31THENrun%(laid%)=
CHCD$(pair%):PROCC (pair%):PROCCs(2):
apair%=apair%+1:prdrp%=1ELSEapair%=0
1340 IFapair%=2THENPROCCs(4)ELSE IFa
pair%=3THENPROCCs(10):apair%=0
1350 IFprdrp%=1THEN1430
1360 IFlaid%<=2THEN1400ELSEfindrun%=
0:lp%=14
1370 REPEAT:IFCHCD$(lp%)<31THENrun%(
laid%)=CHCD$(lp%) ELSE1390
1380 PROCrn:IFgotrun%=3THEN IFtotal
%+CD$(lp%)<=31THENrun%(laid%)=CHCD$(l
p%):PROCC (lp%):lp%=17:findrun%=1
1390 lp%=lp%+1:UNTILlp%>17:IFfindrun
%=1THEN1430
1400 FORI%=14TO17:IFtotal%+CD$(I%)<=
31THENnexc%:=I%:I%=17
1410 NEXT:IFnexc%(>0)THENrun%(laid%)
=CHCD$(nexc%):PROCC (nexc%):GOTO143
0
1420 VDU4,7,18,0,1:PRINTTAB(0,0)ST$:
PRINTTAB(1,1)"Sorry - can't go!":newn
o%=newno%-1:cpnogo%=1:PROCT(1):ENDPRO
C
1430 IFplnogo%=1ANDnpair%=0ANDRIGHT$(
CD$(thiscd%),1)=RIGHT$(CD$(cpnogo%),
1)THENPROCCs(2):copr%=copr%+1:IFcopr
%=2THENPROCCs(4)
1440 IFpl%=4ANDnpair%=0ANDplfin%=1AN
DRIGHT$(CD$(thiscd%),1)=RIGHT$(CD$(cp
nogo%),1)THENPROCCs(2):copr%=copr%+1
:IFcopr%=2THENPROCCs(4)
1450 IFtotal%=31THENPROCCs(2):cpnogo
%=1
1460 IFpl%=4 plfin%=1
1470 ENDPROC
1480 DEF PROCCL(twoetc%):total%=tota
l%+CD$(twoetc%):PROCLA(twoetc%,736):C
HCD$(twoetc%)=newno%:CD$(twoetc%)=new
no%:X2$(co%)=twoetc%:cpnogo%=twoetc%
:co%=co%+1:PROCGT:ENDPROC
1490 DEF PROCPC:IFpl%=4 ORplnogo%=1T
HENENDPROC
1500 PRINTTAB(0,0)ST$:VDU7,17,3:PRIN
TTAB(8-(LEN(NAME$)/2),0)NAME$:COLOUR
2:PRINT "to":PRINTTAB(1,1)"enter card
number":PRINT "or":COLOUR1:PRINT "G
":COLOUR2:PRINT "if can't go":*FX21
,0
1510 N$=GET$:IFN$="G"ORN$="g"THENpln
ogo%=1:ENDPROC
1520 n%=VAL(N$):IFn%<1 ORn%>6THEN150
0
1530 IFGone%(n%)=99THENPRINTTAB(0,0)
```

```
ST$:COLOUR1:PRINTTAB(0,1)"Sorry-alrea
dy gone":VDU7:PROCT(2):GOTO1500
1540 IFtotal%+CD$(n%)>31THENCOLOR1:
PRINTTAB(0,0)ST$:PRINTTAB(0,1)"Sorry-
more than 31":VDU7:PROCT(2):GOTO1500
1550 IFco%=0THEN1590
1560 IFnpair%=0ANDRIGHT$(CD$(n%),1)=
RIGHT$(CD$(X2$(co%-1)),1)THENPROCCs(2
):apair%=apair%+1ELSEapair%=0
1570 IFapair%=2THENPROCCs(4)ELSE IFa
pair%=3THENPROCCs(10):apair%=0
1580 IFco%=4ANDnpair%=0ANDcpfin%=1AN
DRIGHT$(CD$(n%),1)=RIGHT$(CD$(plnogo
%),1)THENPROCCs(2):plpr%=plpr%+1:IFpl
pr%=2THENPROCCs(4)
1590 IFcpnogo%=1ANDnpair%=0ANDRIGHT$(
CD$(n%),1)=RIGHT$(CD$(plnogo%),1)TH
ENPROCCs(2):plpr%=plpr%+1:IFplpr%=2TH
ENPROCCs(4)
1600 IFtotal%+CD$(n%)=15THENPROCCs(2
)
1610 IFtotal%+CD$(n%)=31THENPROCCs(2
):cpnogo%=1
1620 total%=total%+CD$(n%):PROCGT:X1
$(pl%)=((n%*200)-200)+80:VDU5,25,4,(X
1$(pl%)+80);64;127,4:pl%=pl%+1:Gone%(
n%)=99:plnogo%=n%:IFco%=4THENcpfin%=
1
1630 run%(laid%)=CHCD$(n%):ENDPROC
1640 DEF PROCOD:IFcpnogo%=1ANDpl%=4
ORcpnogo%=1ANDplnogo%=1THENpoint%=1:E
NDPROC
1650 IFco%=4 ANDpl%=4 ORco%=4 ANDpln
ogo%=1THENpoint%=1 ELSE ENDPROC
1660 DEF PROCLA(pos%,YY%)
1670 XX%=((pos%-0%)*200)-200)+80:VD
U5,18,0,3:PROCCF(XXX,YY%):PROCCd(pos%
):VDU25,4,(XX%+24):(YY%+120):GCOLOR,C
%:PRINTRIGHT$(CD$(pos%),1):VDU25,4,(X
X%+24):(YY%+54):PRINTLEFT$(CD$(pos%)
,1):VDU4,18,0,3:ENDPROC
1680 DEF PROCPC:HH%=1:LOCALI%:FORI%=
1TO6:IFI%<>No1 ANDI%<>No2 THENCD$(HH%
)=CD$(I%):HH%=HH%+1
1690 NEXT:CD$(5)=CD$(7):ENDPROC
1700 DEF PROCPC:VDU5,18,0,0:LOCALI%:
FORI%=0TOpl%-1:PROCCF(X1$(I%),96):NEX
T:VDU4:ENDPROC
1710 DEF PROCPC:VDU5,18,0,0:LOCALI%:
FORI%=0TOco%-1:x2%=((X2$(I%)-0%)*200
)-200)+80:PROCCF(x2%,736):NEXT:VDU4:E
NDPROC
1720 DEF PROCCs(points%):VDU5,18,0,2
:R%=1:REPEAT:IFPOINT(1008,550)=2THENV
DU18,0,1
1730 MOVE ABS(CXM%),CYM%:PRINT ".":I
FCXM%=48THENCXM%=-CXM%+32:CYM%=568
1740 IFABS(CXM%)=976ANDCYM%=568THEN
CXM%=1008:CYM%=600
1750 SOUND1,-12,R%*2+50,2:PROCT(.3):
IFPOINT(1008,582)=1THENPROCCn(1,688)
```

```
1760 IFPOINT(1008,550)=1THENPROCCn("
The computer")
1770 CXM%=CXM%-32:R%=R%+1:UNTILR%>po
ints%:VDU4:ENDPROC
1780 DEF PROCSC(B%,TX%):PROCAL(B%,TX%
):PROCSW(B%,TX%):FVR=0:FRR=0:sc%=0:PRO
CV(B%,TX%):PROCFR(B%,TX%):PROCTH(B%,TX%
):PROCTW(B%,TX%):PROCPR(B%,TX%):PROCJK(B
%,TX%):PROCSM(B%,TX%):ENDPROC
1790 DEF PROCCD(F%):FORx%=80TO1080ST
EP200:y%=736:IFF%=0ANDx%=80 ORF%=0AND
x%=1080THENGOLOR,0:PROCCF(x%,y%):GOTO
1810
1800 GCOLOR,3:PROCCF(x%,y%)
1810 IFF%=1THENGOLOR,1:PROCCF(x%,y%)
1820 GCOLOR,3:y%=96:PROCCF(x%,y%):PRO
CPC(x%):NEXT:ENDPROC
1830 DEF PROCPC(score%):VDU5,18,0,2:
R%=1:REPEAT:IFPOINT(1008,422)=2THENV
DU18,0,1
1840 MOVEABS(PXM%),PYM%:PRINT ".":IF
PXM%=48THENPXM%=-PXM%+32:PYM%=440
1850 IFPXM%=48THENPXM%=-PXM%+32:PYM%
=440
1860 IFABS(PXM%)=976ANDPYM%=440THENP
XM%=1008:PYM%=472
1870 SOUND2,-12,100-R%*2,2:PROCT(.3)
:IFPOINT(1008,454)=1THENPROCCn(2,320)
1880 IFPOINT(1008,422)=1THENPROCCn(N
AME%)
1890 PXM%=PXM%-32:R%=R%+1:UNTILR%>sc
ore%:VDU4:ENDPROC
1900 DEF PROCFC(IC%,Y%):LOCALI%:VDU5
:FORI%=280TO880STEP200:PROCCd(IC%):GC
OLOR,C%:MOVE I%+25,Y%-56:PRINT:LEFT$(C
D$(IC%),1):MOVEI%+25,Y%:PRINT:RIGHT$(
CD$(IC%),1):IC%=IC%+1:NEXT:VDU4:ENDPR
OC
1910 DEF PROCUP:VDU5,18,0,1,25,4,110
5;320;9,127,25,4,1105;688;230,7,4:END
PROC
1920 DEF PROCDN:VDU5,18,0,2,25,4,110
5;688;9,127,25,4,1105;320;230,7,4:END
PROC
1930 DEF PROCRC:VDU5,18,0,0:FORx%=80
TO1080STEP200:y%=736:PROCCF(x%,y%):y%
=96:PROCCF(x%,y%):VDU25,4,(x%+96);64;
127:NEXT:VDU4:ENDPROC
1940 DEF PROCWN(W$)
1950 PROCBL:VDU4,17,1:PRINTTAB(0,0)S
T$:PRINTTAB(10-(LEN(W$)/2),0)W$:COLO
UR2:PRINT:PRINTTAB(3);"is the winner!
":TI%=1:VDU5:REPEAT:FORI%=1TO3:VDU18
,0,1,25,4,970;512;(226+I%):PROCCd:NE
XT:VDU25,4,1034;512;127
1960 PROCT(.2):TI%=TI%+1:UNTILTI%=5:
VDU4,17,3:PRINTTAB(0,0)ST$:PRINTTAB(1
,0)"Would you like":COLOUR2:PRINTTAB(
2);"another game of":COLOUR1:PRINTTAB
```



# 3D - TANK ZONE Sensational NEW release!!!



For  
**BBC 32k**

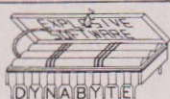
At last, a stunning interpretation of the 3D Wire-Graphics Tank Battle simulation you've all been waiting for brought to you by Dynabyte.

Features include anti-tank missile launcher with realistic 'shell-in-flight' action, radar screen, detailed control panel, anti-aircraft and anti-tank modes (with options to choose one or both), hidden line removal, exploding tanks and many other special effects. There is NO other game like it.

**DYNABYTE**

EXPLOSIVE

Software



Available now at all good computer shops or direct from

**DYNABYTE SOFTWARE**

PO Box 10 Barnsley S. Yorkshire  
0226-287707

£8.95

Now for  
the  
Electron

SPECIAL  
OFFER TO  
READERS  
All 5 games +  
"Brain teasers" on  
one disc. 40 track  
or 80 track.  
Just £29.00

Disc versions available,  
same price for  
Mail Order only

SAE for catalogue.

(All programs require O.S. 1.2.)

Please state BBC or Electron when ordering and include 50p P&P

**POOL**  
£6.95  
Exciting pool hall action all the way in this 2-player game of skill. Excellent ball control and smooth graphics an outstanding feature of this program.  
**BBC and Electron versions**  
(Machine code 32k)

**CORPORATE CLIMBER**  
£6.95  
Avoid the eager taxmen in the lifts ready to hinder your climb to power and beware of too much stress resulting in high blood pressure.  
**BBC and Electron versions**  
(Machine code 32k)

**HORSE RACE**  
£6.95  
All the excitement of the track brought to your own home in this colourful game complete with tumbling jockeys, realistic horses, TV van, tote and leader boards, waving crowds and much more. Suitable for 1-6 players.  
**BBC and Electron versions**  
(Basic + M/C 32k)

**LEMMING SYNDROME**  
£6.95  
Save the panicking population in this highly original game full of special features and options designed to make your task harder as you get better.  
**BBC and Electron versions**  
(Machine code 32k)



# PHILIPS 14" COLOUR TV/MONITOR

£235 inc VAT

(TV only: £185 inc VAT)

TV/Monitor (resolution better than 585 x 450 pixels. Dot-to-dot distance .65mm)

As a company we chose the Philips because we believe it to have the clearest picture as well as being the best value. Take our word for it, or come and see for yourself.

**EASY ON-THE-SPOT PARKING. AND IT'S FREE!**

**Viglen**

COMPUTER SUPPLIES

Personal Service & Quality coupled with Wholesale Prices. Unit 7 Trumpers Way, Hanwell, London W7 2QA. Telephone: 01-843 9903.

For special attention to your orders or enquiries concerning MONITORS, PRINTERS, COMPUTERS, DISC DRIVES, ring 01-843 9903 NOW or fill in the coupon below.

Plus £8 carriage and packing



- Easy to carry • Lightweight • Sparkling picture in seconds • Reliability and performance excellent • Headphone Socket can also be connected to tape recorder • Loudspeaker mute switch • Retractable carrying handle
- Monitor/TV selected by switch • Includes RGB lead connecting direct to your BBC Micro. • Philips Model No. 14CT 2006/05T

Post to: VIGLEN COMPUTER SUPPLIES, UNIT 7, TRUMPERS WAY, HANWELL, LONDON W7 2QA.

MU 11/7

Please send me \_\_\_\_\_ (qty) Philips TV/Monitor at £243 each. I enclose Cheque/P.O. for £\_\_\_\_\_ made out to

VIGLEN COMPUTER SUPPLIES or debit my ACCESS/BARCLAYCARD No. \_\_\_\_\_

Name \_\_\_\_\_



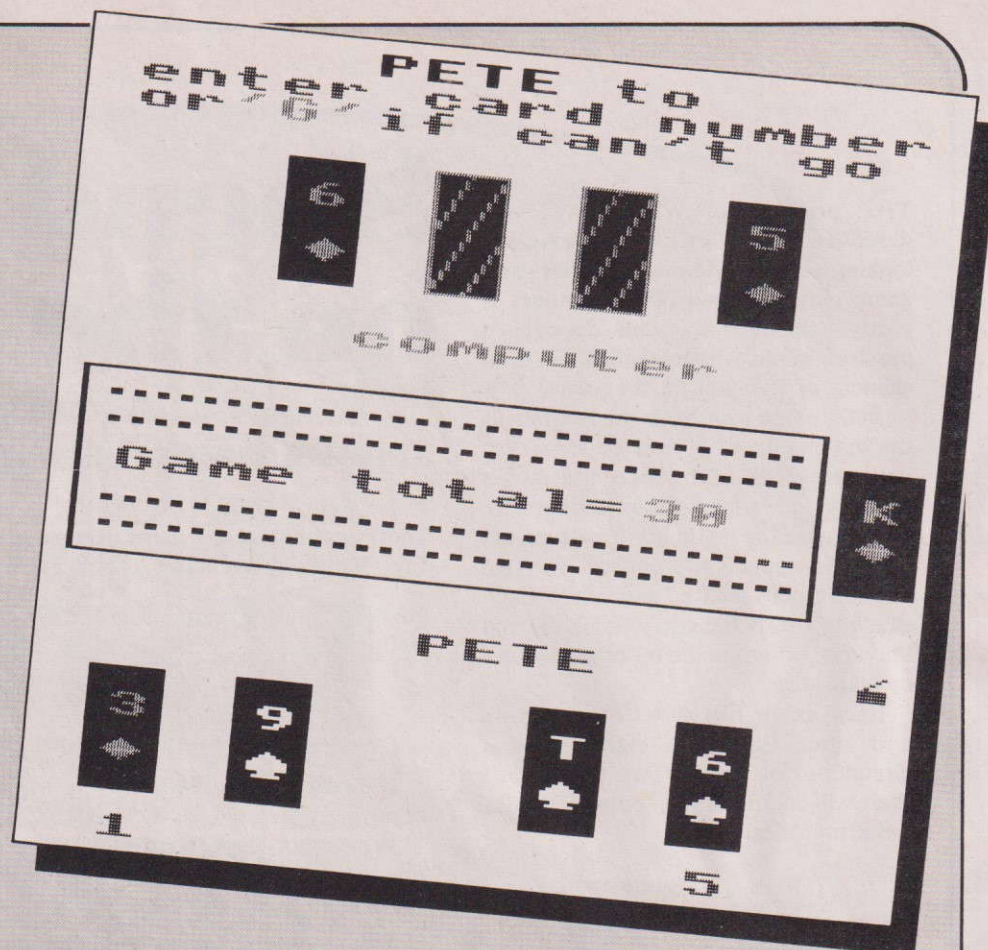
Signature \_\_\_\_\_

Address \_\_\_\_\_



# From Page 65

```
(3);"cribbage - (Y/N)";:yn$=GET$:IFyn
$(>)"Y"THEN END ELSE GOT070
1970 DEF PROC PF:VDU7,17,2:PRINTTAB(1
0-(LEN(NAME$)/2),1) NAME$:PROC PF:PROC
SC(1,5):IFsc%>0THENPROCscore:PROCps(sc%):PROCB
1980 PRINTTAB(0,1)ST1$:COLOUR1:PRINT
TAB(6,1)"computer":VDU7:PROCSC(14,18)
:IFsc%>0THENPROCscore:PROCcs(sc%):PRO
CBL
1990 PROCRC:PROCBX(14,736):PROCSC(14
,18):IFsc%>0THENPROCscore:PROCcs(sc%):
PROCB ELSECOLOUR1:PRINTTAB(0,0)ST$:
PRINTTAB(6,0)"B*!#* it":COLOUR2:PRIN
TTAB(2,2)"my box is empty!":PROCsd:PR
OCT(3):ENDPROC
2000 PROCT(2):ENDPROC
2010 DEF PROCCT:VDU7,17,1:PRINTTAB(6
,1)"computer":PROCSC(14,18):IFsc%>0TH
ENPROCscore:PROCcs(sc%):PROCB
2020 PRINTTAB(0,1)ST1$:COLOUR2:PRINT
TAB(10-(LEN(NAME$)/2),1)NAME$:VDU7:PR
OCPH:PROCSC(1,5):IFsc%>0THENPROCscore
:PROCps(sc%):PROCB
2030 PROCRC:PROCBX(1,96):PROCSC(1,5)
2040 IFsc%>0THENPROCscore:PROCps(sc%
):PROCB ELSECOLOUR1:PRINTTAB(0,0)ST$
:PRINTTAB(7-(LEN(NAME$)/2),0) "Sorry
";:COLOUR3:PRINTNAME$:COLOUR2:PRINTTA
B(2,1)"There's nothing":PRINTTAB(3);"
in your box!";:PROCsd:PROCT(3):ENDPRO
C
2050 PROCT(2):ENDPROC
2060 DEF PROCBX(Z%,Y%):TZ=1:LOCALI%:
FORI%=Z%TOZ%+4:CD$(I%)=BOX$(TZ%):TZ=TZ
+1:NEXT:VDU5,18,0,3:FORx%=280TO880STE
P200:PROCCF(x%,Y%):NEXT:PROCCF(Z%,Y%+
112):ENDPROC
2070 DEF PROC MN:PRINTTAB(13,2);CHR$1
34;CHR$141;"CRIBBAGE":PRINTTAB(13);CH
R$134;CHR$141;"CRIBBAGE"
2080 PRINTTAB(0,7);CHR$133;"NOTE:";C
HR$131;"With the exceptions of enteri
ng":PRINTTAB(6);CHR$131;"your name an
d boxcard selection":PRINTTAB(6);CHR$
131;"all entries are single key."
2090 PRINTTAB(6);CHR$129;"At the end
of each game the":PRINTTAB(6);CHR$12
9;"computer will display the scores":
PRINTTAB(6);CHR$129;"You must then pr
ess a key to":PRINTTAB(6);CHR$129;"co
ntinue."
2100 PRINTTAB(1,17);CHR$134;"Now ent
er your name-(1-14 letters)":PRINT:IN
PUT LINETAB(10)NAME$
2110 IFLEN(NAME$)>14ORNAME$=""THENPR
INTTAB(0,17);STRING$(199," "):VDU7:PR
INTTAB(0,16);CHR$136;CHR$129;"Come on
, please enter correctly!";:PROCT(4):
GOTO2100
```



```
2120 ENDPROC
2130 DEF PROCcd(col%):IFLEFT$(CD$(co
l%),1)=HEART$ ORLEFT$(CD$(col%),1)=DI
AMOND$THENC%=1 ELSEC%=0
2140 ENDPROC
2150 DEF PROCcn(c%,w%):VDU18,0,c%,5,
25,4,96;w%:226;:ENDPROC
2160 DEF PROC CP:kp%=0:LOCALI%,J%,K%,
L%:FORI%=8TO10:FORJ%=I%+1TO11:FORK%=J
%+1TO12:FORL%=K%+1TO13:totno%=0:CD$(1
4)=CD$(I%):CD$(15)=CD$(J%):CD$(16)=CD
$(K%):CD$(17)=CD$(L%)
2170 PROCAL(14,17):PROCSW(14,17):PRO
CCC(14,17):IFsc%>0kp%THENi%=I%:j%=J%:k
%=K%:l%=L%:totno%=i%+j%+k%+l%:PROCCX:
kp%=sc%
2180 NEXT:NEXT:NEXT:NEXT
2190 IFkp%=0THENPROCSE:ENDPROC
2200 CD$(14)=CD$(i%):CD$(15)=CD$(j%):
CD$(16)=CD$(k%):CD$(17)=CD$(l%):CD$(
9)=CD$(14):CD$(10)=CD$(15):CD$(11)=CD
$(16):CD$(12)=CD$(17):NUM1=1:NUM2=6:A
%=9:B%=12:ENDPROC
2210 DEF PROCCX:IFtotno%=38THENBOX$(
3)=CD$(12):BOX$(4)=CD$(13)
2220 IFtotno%=39THENBOX$(3)=CD$(11):
BOX$(4)=CD$(13)
2230 IFtotno%=42THENBOX$(3)=CD$(8):B
OX$(4)=CD$(13)
2240 IFtotno%=45THENBOX$(3)=CD$(8):B
OX$(4)=CD$(10)
2250 IFtotno%=46THENBOX$(3)=CD$(8):B
OX$(4)=CD$(9)
2260 IFtotno%=40ANDk%=10THENBOX$(3)=
```

```
CD$(11):BOX$(4)=CD$(12) ELSE IFtotno%
=40THENBOX$(3)=CD$(10):BOX$(4)=CD$(13
)
2270 IFtotno%=41ANDj%=9THENBOX$(3)=C
D$(10):BOX$(4)=CD$(12) ELSE IFtotno%=
41THENBOX$(3)=CD$(9):BOX$(4)=CD$(13)
2280 IFtotno%=43ANDi%=8THENBOX$(3)=C
D$(9):BOX$(4)=CD$(11) ELSE IFtotno%=4
3THENBOX$(3)=CD$(8):BOX$(4)=CD$(12)
2290 IFtotno%=44ANDi%=8THENBOX$(3)=C
D$(9):BOX$(4)=CD$(10) ELSE IFtotno%=4
4THENBOX$(3)=CD$(8):BOX$(4)=CD$(11)
2300 ENDPROC
2310 DEF PROCBL:VDU5,18,0,0,25,4,103
2;512;127,127,127,127,127,127,127,127
,127,127,127,127,127,127,4:ENDPRO
C
2320 DEF PROCscore:PROCB:VDU5,18,0,
2,25,4,80;512;:PRINT"Total Points=";:
VDU18,0,1:PRINT;sc%:VDU4:GET:=F$X2
1,0
2330 ENDPROC
2340 DEF PROCrn:LOCALI%,J%,K%:gotrun
%=1:FORI%=1TO14:J%=2:FORJ%=I%+1TO14:
J%-1:IFrun%(I%)<run%(J%)THEN2360
2350 run_temp%=run%(I%):run%(I%)=run
%(J%):run%(J%)=run_temp%
2360 NEXT:NEXT:FORK%=1TO14:J%-1:IFru
n%(K%)+1=run%(K%+1)THENgotrun%=gotrun
%+1ELSE IFrun%(K%+1)<>0THENgotrun%=1
2370 NEXT:ENDPROC
2380 IFERR=25THEN70ELSEMODE7:REPORT:
PRINT" at line No. ";ERL:VDU23;11,255
;0;0;0
```

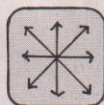


THE principle of Which-Way? was described in 1969 by a research worker looking into psycho-cybernetics and game playing. Each of your moves is partly controlled by your opponent's previous move, which means there is an element of feedback in the game.

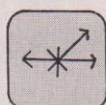
In the same way, your move controls the options open to your opponent, so you may be able to force him into a move which is favourable to yourself.

The board is made up of 64 tiles, set out in the usual 8 x 8 pattern. The two opposing sides are Red and White. Each side has a start tile and a finish tile and the other 60 are made up of arrow tiles or blank tiles.

Each arrow tile shows three arrows and since there are eight possible directions for the arrows this means there will be 56 different possibilities for the arrow tiles.



8 directions

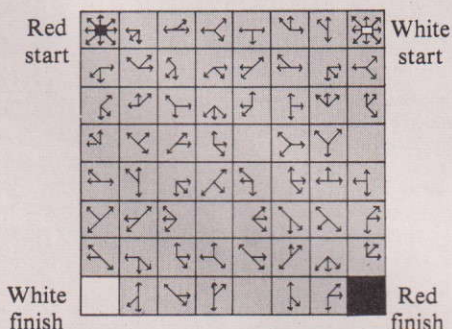


one possible tile

$$\frac{8 \times 7 \times 6}{1 \times 2 \times 3} = 56$$

If we take a completely blank tile into account then there will be 57 possible tiles to choose from. The program chooses at random from these designs and prints them out in the 8 x 8 layout.

On the first run this takes about five or six seconds to be produced, but in further games the board is printed on demand.



The aim of the game is to get from Red start to Red finish or from White start to White finish.

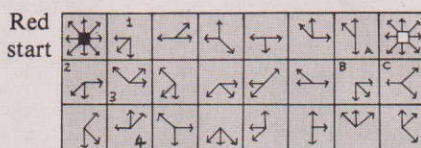
There is one basic rule which controls the moves in the game – you can only move in a direction shown by one of the arrows on your opponent's tile. To make your moves you use the keys shown in Figure I.

Here is part of one possible layout

# Spend a night on the tiles...

... but make sure you head in the right direction with **JERRY LANCASTER's game of strategy for two players**

showing the start of a game. Red moves first. He has three choices, tiles 1, 2 or 3:



If Red moves to tile 1, White has a completely free choice – A, B or C.

White chooses A, Red will have only one legal move (down to tile 3) because the other two directions take him off the board.

White chooses B, Red has a choice of three moves all of which take him in the right direction.

White chooses C, Red will be left with two moves, one of which takes him back to the Red start tile, giving White a free choice next move.

If Red moves to tile 2, White will have only two possible moves, B or C.

White chooses B, and gives Red the advantage of a free choice of three directions.

White chooses C, and Red has two moves possible this time, back up to tile 1 or down to tile 4.

If Red moves to tile 3, White has no legal moves available because all Red's arrows take White off the board.

This move by Red to tile 3 looks like

his best option.

If you attempt an illegal move the computer will not execute it. Instead it will leave you where you are and award the move to your opponent.

Illegal moves are:

- Any move which would take you off the board.
- Any move which does not match one of the arrows on your opponent's tile.
- Any move which would put both of you onto the same tile.

Some tiles are blank. If you move onto a blank tile your opponent will be left with no legal moves, so in effect you will get a free move. When you are forced to stay on the same tile you can do so by pressing the central key of your control group.

Red always starts the game but White can decide whether or not to accept the layout displayed. If Y is pressed you can get an entirely new layout, without losing the current game score.

This option is also available during play, so that if a drawn position is reached you can restart that game.

Be careful when entering the program that you don't inadvertently include any spaces or suchlike as it is very tight on memory. You will certainly have to download it on DFS machines.

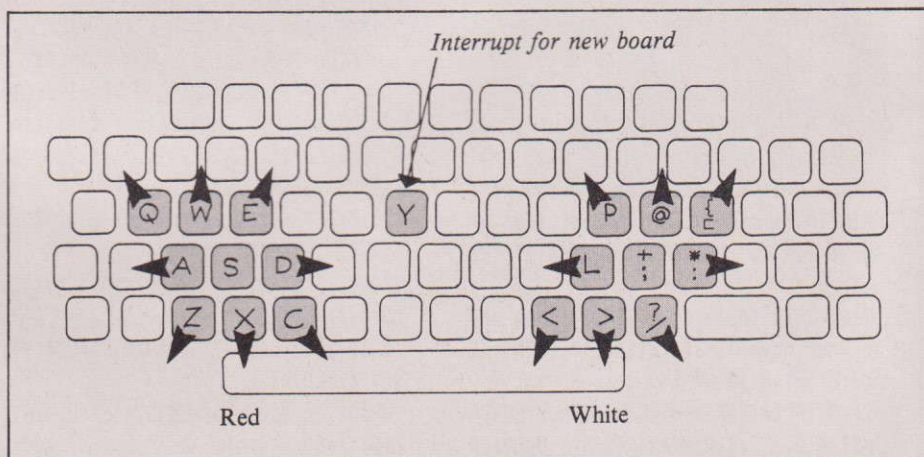
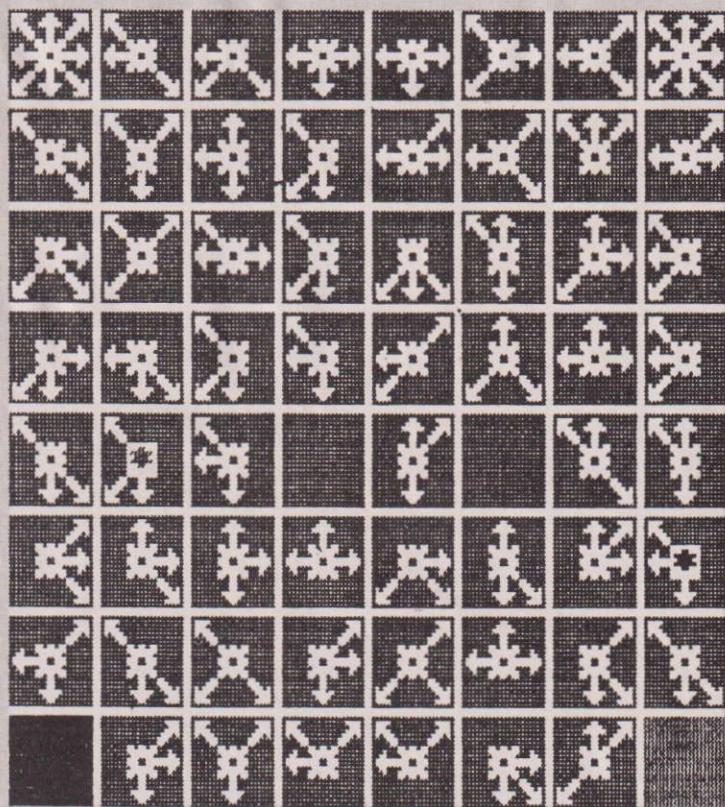


Figure I





WHITE TO MOVE

RED 0

WHITE 0





## PROCEDURES AND MAIN VARIABLES

sh\$(N%)

SHAPE%(N%)

TILE%(X,Y)

red, white

VDU23,224 to 232

VDU23,234 to 242

move1\$, move 2\$

SHAPE%(N%)

SH\$

### DEFPROCinit

This is an entire tile pattern made up from nine user defined characters and several cursor moves, all printed as one unit.

A Binary coded nine-digit number which describes the tile pattern.

Array which records the layout of the tiles on the board for a particular game.

The game scores.

'Arrow' shapes.

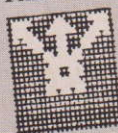
'No arrow' shapes.

### DEFPROCtileshapes

Cursor movement controls which form part of each string describing a tile.

Read as a number from data statements. The data is written in the form 111010000.

This is derived from a tile pattern:



1	1	1
0	1	0
0	0	0

String version of SHAPE%(N%).  
Leading zeros will have been omitted and need to be replaced.

sh\$(N%)

X% Y%

PROCmoveA

PROCmoveB

I%

PROClegalmove

PROCwin

The tile pattern obtained from SH\$.  
MID\$(...) is used to detect the '1' which shows where the arrow should be.

### DEFPROCboard

These correspond to the columns and rows of the 8 x 8 board.

Values are used to identify particular tiles, such as the corners. The remaining tile patterns are chosen at random using N% = RND(57).

RED and WHITE are printed on their start squares using "\*" in the right colours.

### DEFPROCgame

Used to move each 'man'.

Uses INKEY(0) to detect key presses. Acceptable returns are converted into values of I% from 1 to 9. I% then corresponds to tile sectors 1 to 9.

1	2	3
4	5	6
7	8	9

The values are checked using MID\$(...) to determine the legality of a move.

If a winning position is reached a winning message is printed in the appropriate colour and the game score is updated.

```

20 REM (C) The Micro User
30 MODE1
40 VDU23,1,0;0;0;0;
60 *FX4,1
70 *FX11,0
90 PROCinit
100 PROCtileshapes
110 PROCboard
120 PROCgame
130 COLOUR2:CLS:BOTD110
140 END
160 DEFPROCinit
165 ENVELOPE2,2,6,0,0,255,0,0,126,0
,0,-126,126,126
170 DIMsh$(58)
180 DIMTILE%(8,8)
190 DIMSHAPE%(58)
200 red=0:white=0
210 PRINTTAB(7,7);"WHICH-WAY?"
220 VDU23,224,0,127,67,71,67,81,120
,124
230 VDU23,225,0,255,231,195,129,231
,231,231
240 VDU23,226,0,254,194,226,194,138
,30,62
250 VDU23,227,127,119,103,64,64,103
,119,127
260 VDU23,228,36,0,129,24,24,129,0,
36
270 VDU23,229,254,238,230,2,2,230,2
38,254
280 VDU23,230,124,120,81,67,71,67,1
27,0
290 VDU23,231,231,231,231,129,195,2
31,255,0
300 VDU23,232,62,30,138,194,226,194
,254,0
310 VDU23,234,0,127,127,127,127,127
,127,127
320 VDU23,235,0,255,255,255,255,255
,255,255
330 VDU23,236,0,254,254,254,254,254
,254,254
340 VDU23,237,127,127,127,127,127,1
27,127,127
350 VDU23,238,255,255,255,255,255,2
55,255,255
360 VDU23,239,254,254,254,254,254,2
54,254,254
370 VDU23,240,127,127,127,127,127,1
27,127,0
380 VDU23,241,255,255,255,255,255,2
55,255,0
390 VDU23,242,254,254,254,254,254,2
54,254,0
400 PRINTTAB(8,14);"Setting up board"
410 ENDPROC
420 DEFPROCtileshapes
430 move1$=CHR$(8)+CHR$(8)+CHR$(8)+
CHR$(10)
440 move2$=CHR$(11)+CHR$(11)
450 FORNX=1TO58
460 READSHAPE%(NX)
470 SH$=STR$(SHAPE%(NX))
480 REPEAT
490 IF LEN((SH$))<9 SH$="0"+SH$
500 UNTIL LEN((SH$))=9
510 sh$(NX)=""
520 FORAX=1TO9
530 IF MID$(SH$,AX,1)="1"sh$(NX)=sh
$(NX)+CHR$(223+AX)ELSEsh$(NX)=sh$(NX)
+CHR$(233+AX)
540 IFA%=3DRA%=6sh$(NX)=sh$(NX)+mov
el$
550 IFA%=9sh$(NX)=sh$(NX)+move2$
560 NEXTAX
570 NEXTNX
580 ENDPROC
590 DATA111010000,011110000,0011110

```



# WHAT A BARGAIN!

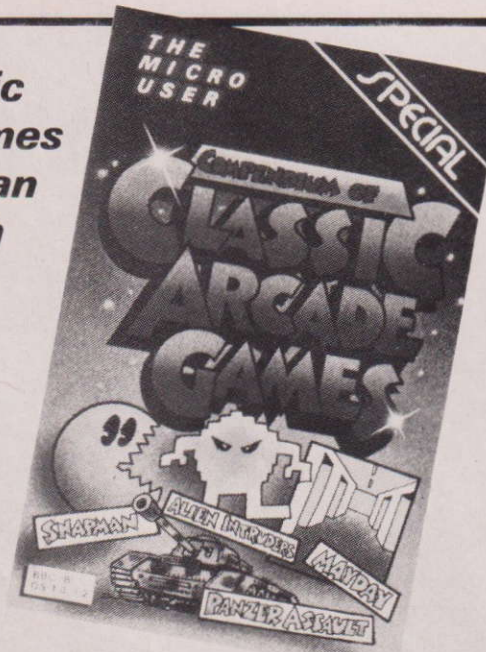
Four classic arcade games for less than £1.50 each

Three of this high-powered collection are top-rate machine-code versions of arcade classics and the fourth is a thrilling real-time adventure game. There's hours of enjoyment and something to suit everyone in this unique value for money collection.

**ALIEN INTRUDERS** – With only your laser for protection you must destroy the waves of aliens who threaten to engulf you. **PANZER ATTACK** – You are a tank commander engaged in vicious combat against encircling enemy forces. **SNAPMAN** – Guide your man through

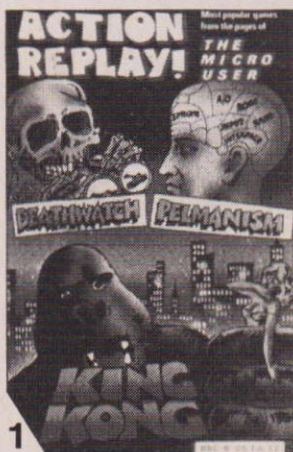
the maze as he munches energy pellets and avoids hostile aliens. **MAYDAY** – A futuristic adventure! As captain of an interstellar cruiser you must guide the sole survivor of a stricken space freighter through the wreckage of his craft.

£5.95 (cassette) £7.95 (disc)



## ACTION REPLAY!

Build your own handy, action-packed cassette library of these all-time Micro User favourites. Only £3.75 per cassette (three games in each).



**Deathwatch** Hold the fort against a combined tank and helicopter assault on your positions. Fast and furious – one of our most popular games.

**Pelmanism** Defeat the computer by remembering more pairs of cards than it can.

**King Kong** Use your helicopter to rescue two maidens from the Empire State building – avoiding a deluge of rocks from King Kong.



**Fruities** Fast-paced, fun-packed extravaganza. Guide your man up ladders, avoiding – and digging traps for – the marauding Fruities!

**Nim** Find out just how nimble your mind is in this intriguing battle of wits against the computer.

**Space Pilot** Desperately fighting off hostile life-forms, you must dock with satellites to replenish dwindling fuel supplies.



**Robin and Marian** Medieval melodrama! Rescue Maid Marian from the sheriff. A hilarious, action-packed game.

**EECGAL** In charge of the EEC's Picture Gallery, you have to put some order into the frames – but it's not so simple as it sounds.

**Air Strike** You need to fly fast and low for this dangerous intrusion into enemy air space.



**Barrel Battles** Fast and furious video action. Dodging an avalanche of deadly barrels you climb towards the telephone.

**Frogs 'n' Dogs** Loved by young and old alike. Simply take Freddy Frog for a walk, avoiding Fido and poisonous lilies!

**Tenpins** Stunning simulation of ten pin bowling. This addictive game will hold you spellbound.



**Tower of London** Ghosts have broken out of the Tower of London and taken the treasure. Can you recover it?

**Blackjack** A true-to-life recreation of the casino favourite. Can you beat the bank?

**Galactic Invaders** The only thing between you and imminent destruction by waves of hostile aliens is your proton gun. A compulsive classic.



**Alien Intruders** Stunning version of the all-time arcade hit. No one can resist this classic!

**Goblins** Frolics in fairyland! Guide Kaplan the pixie home through the forests, avoiding the nasty goblins – a delightful game.

**Gone Fishing** Superb simulation of a day on the riverside. This enthralling program will give hours of enjoyment.

Please send me:

### CLASSIC ARCADE GAMES

BBC 'B' Cassette £5.95  
40 Track Disc £7.95  
80 Track Disc £7.95

### ACTION REPLAY

BBC 'B' Cassette £3.75

1	4
2	5
3	6

☐ I enclose cheque made payable to Database Publications Ltd. for £

☐ I wish to pay by ☐ Access ☐ Visa

No. \_\_\_\_\_ Expiry date \_\_\_\_\_

Signed \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

Post to: Database Publications, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.



## From Page 70

```

00,000111100,000011110,000010111
600 DATA110110000,110011000,1100101
00,110010010,110010001
610 DATA011011000,011010100,0110100
10,011010001
620 DATA001110100,001110010,0011100
01,101110000
630 DATA000111010,000111001,1001110
00,010111000
640 DATA000011101,100011100,0100111
00,001011100
650 DATA100010110,010010110,0010101
10,000110110
660 DATA100010011,010010011,0010100
11,000110011,000011011
670 DATA101011000,101010100,1010100
10,101010001
680 DATA100110100,100110010,1001100
01
690 DATA100011010,100011001,1000101
01
700 DATA010110100,010110010,0101100
01
710 DATA010011010,010011001,0100101
01
720 DATA001011010,001011001,0010101
01,000110101
730 DATA000000000,111111111
740 DEFPROCboard
750 X%=1
760 Y%=1
770 CLS
780 COLOUR2
790 Z%=0
800 REPEAT
810 N%=RND(57)
820 IFX%=1ANDY%=1N%=58
830 IFX%=8ANDY%=1N%=58
840 IFX%=1ANDY%=8 N%=57
850 IFX%=8ANDY%=8 N%=57
860 TILEX(X%,Y%)=SHAPEX(N%)
870 PRINTsh$(N%);
880 X%=X%+1
890 IFX%>8PRINT:PRINT:PRINT:X%=1:Y%
=Y%+1
900 IFX%=1ANDY%=8 COLOUR3 ELSE COLO
UR2
910 IFX%=8ANDY%=8 COLOUR1
920 UNTILY%>8
930 XA=1:YA=1:XB=22:YB=1
940 COLOUR1:PRINTTAB(XA,YA);"*"
950 COLOUR3:PRINTTAB(XB,YB);"*"
960 *FX21,0
970 ENDPROC
980 DEFPROCgame
990 REPEAT
1000 COLOUR1:PRINTTAB(27,4);"RED TO

```

```

MOVE "
1010 PRINTTAB(27,12);"RED "red
1020 COLOUR3:PRINTTAB(27,15);"WHITE
"white
1030 PROCmoveA
1040 SOUND1,-15,97,2
1050 IFZ%=160T01090
1060 COLOUR3:PRINTTAB(27,4);"WHITE T
O MOVE"
1070 PROCmoveB
1080 SOUND1,-15,197,2
1090 UNTILZ%=1
1100 IFXA=22ANDYA=22red=red+1
1110 IFXB=1ANDYB=22white=white+1
1120 COLOUR1:PRINTTAB(27,12);"RED
"red
1130 COLOUR3:PRINTTAB(27,15);"WHITE
"white
1140 PRINT TAB(4,28)"Press space for
next game"
1150 REPEAT UNTIL GET$=" "
1160 ENDPROC
1170 DEFPROCmoveA
1180 REPEAT
1190 xa=XA:ya=YA
1200 I%=INKEY(0)
1210 UNTIL I%<>-1
1220 IF I%=81 I%=1
1230 IF I%=87 I%=2
1240 IF I%=69 I%=3
1250 IF I%=65 I%=4
1260 IF I%=83 I%=5
1270 IF I%=68 I%=6
1280 IF I%=90 I%=7
1290 IF I%=88 I%=8
1300 IF I%=67 I%=9
1310 IF I%=89 Z%=1:ENDPROC
1320 IF I%<1 OR I%>9 THEN SOUND1,-15
,33,7:GOTO1180
1330 PROClegalmove(XB,YB)
1340 IF I%=-1 ENDPROC
1350 IF I%=1 XA=XA-3:YA=YA-3
1360 IF I%=2 YA=YA-3
1370 IF I%=3 XA=XA+3:YA=YA-3
1380 IF I%=4 XA=XA-3
1390 IF I%=5 XA=XA:YA=YA
1400 IF I%=6XA=XA+3
1410 IF I%=7 XA=XA-3:YA=YA+3
1420 IF I%=8 YA=YA+3
1430 IF I%=9 XA=XA+3:YA=YA+3
1440 IFXA<1 XA=xa:YA=ya
1450 IFXA>22 XA=xa:YA=ya
1460 IFYA<1 XA=xa:YA=ya
1470 IFYA>22 XA=xa:YA=ya
1480 IFXA=XB AND YA=YB XA=xa:YA=ya
1490 COLOUR2:PRINTTAB(xa,ya);CHR$(22
B)
1500 COLOUR1:PRINTTAB(XA,YA);"*"
1510 PROCcheck

```

```

1520 ENDPROC
1530 DEFPROCmoveB
1540 REPEAT
1550 I%=INKEY(0)
1560 UNTIL I%<>-1
1570 xb=XB:yB=YB
1580 IF I%=80 I%=1
1590 IF I%=64 I%=2
1600 IF I%=91 I%=3
1610 IF I%=76 I%=4
1620 IF I%=59 I%=5
1630 IF I%=58 I%=6
1640 IF I%=44 I%=7
1650 IF I%=46 I%=8
1660 IF I%=47 I%=9
1670 IF I%=89 Z%=1:ENDPROC
1680 IF I%<1 OR I%>9 THEN SOUND1,-15
,33,7:GOTO1540
1690 PROClegalmove(XA,YA)
1700 IF I%=-1 ENDPROC
1710 IF I%=1 XB=XB-3:YB=YB-3
1720 IF I%=2 YB=YB-3
1730 IF I%=3 XB=XB+3:YB=YB-3
1740 IF I%=4 XB=XB-3
1750 IF I%=5 XB=XB:YB=YB
1760 IF I%=6 XB=XB+3
1770 IF I%=7 XB=XB-3:YB=YB+3
1780 IF I%=8 YB=YB+3
1790 IF I%=9 XB=XB+3:YB=YB+3
1800 IFXB<1 XB=xb:YB=yb
1810 IFXB>22 XB=xb:YB=yb
1820 IFYB<1 XB=xb:YB=yb
1830 IFYB>22 XB=xb:YB=yb
1840 IF XB=XA AND YB=YA XB=xb:YB=yb
1850 COLOUR2:PRINTTAB(xb,yb);CHR$(22
B)
1860 COLOUR3:PRINTTAB(XB,YB);"*"
1870 PROCcheck
1880 ENDPROC
1890 DEFPROCcheck
1900 IFXA=22 AND YA=22 PROCwin
1910 IFXB=1 AND YB=22 PROCwin
1920 ENDPROC
1930 DEFPROCwin
1935 SOUND1,2,4,50
1940 PRINTTAB(27,4);"
1950 PRINTTAB(4,25);"The winner ! *
* * *"
1960 Z%=1
1970 ENDPROC
1980 DEFPROClegalmove(X,Y)
1990 x=(X+2)/3:y=(Y+2)/3
2000 a$="0"
2010 Z%=STR$(TILEX(x,y))
2020 REPEAT
2030 IF LEN(Z%)<9 THEN Z%=a$+Z%
2040 UNTILLEN(Z%)=9
2050 IF MID$(Z%,1%,1)<>"1" I%=-1
2060 ENDPROC

```



# The best software value ever!

**£7.95**  
Cassette/Disc

**£14.00**  
Both Discs

IT'S an offer you can't afford to refuse – all 14 fantastic games from The Micro User's Games Special are now available on tape or disc to save you the trouble of typing them in.

Never before has such high quality fun-packed games been offered at such a low price!

In fact the package is so crammed with goodies that for the DFS version we've had to split the cassette over two discs.

On DISC A (Cassette Side A) you'll find Bounce, Crib, Fair Play, Sevens, Which Way, White Widow and Yams.

DISC B (Cassette Side B) features China Drop, Crown Green, Draw Poker, Explorer Eddie, Masterspy, Patience and Splat.

And every game is of the high standard for which Micro User is renowned!

Never before has so much BBC Micro software been offered for so little.

**Don't miss out – send off for your cassette or discs today**



Please send me Games Special software as indicated here:

☐ I enclose cheque made payable to Database Publications Ltd. for £ \_\_\_\_\_

☐ I wish to pay by ☐ Access ☐ Visa

No. \_\_\_\_\_ Expiry date \_\_\_\_\_

Signed \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

☐ Cassette £7.95

☐ Disc A £7.95

☐ Disc B £7.95

☐ Discs A and B £14.00

☐ 40 track ☐ 80 track

N.B. No Electron version

Post to: Games Special, Database Publications,  
68 Chester Road, Hazel Grove, Stockport SK7 5NY.



**YAMS** is an adaptation of the traditional dice game Yahtzee and can be played by two to four players.

The object of the game is to get the highest score possible by using a set of five dice to obtain certain combinations, called scores. These are outlined below.

Each player takes a turn at throwing the set of dice. During that turn you have a maximum of three throws. After the first throw you may

- ☐ *Throw all the dice again.*
- ☐ *Hold selected dice and throw the rest.*
- ☐ *Accept the throw.*

After the third throw, you must accept the dice shown. If you can select a score from this combination of dice then you must do so – if not you can pass.

Once you have made a valid selection, a summary of your current scores is shown and the game then passes on to the next player. It cycles round all the players until everyone has had 14 turns – enough to cover all the scores.

A final score sheet is then shown and the winner declared.

# Dicey business

Who can get the best score from throwing a set of dice? Play Yams by JULIA AIZPURVS and find out!

## KEYS

<b>Y</b>	<i>Accept a throw</i>
<b>Space</b>	<i>Throw all dice again</i>
<b>H</b>	<i>Hold some of the dice</i>

Once you have pressed H, you are asked how many dice you wish to keep and then which ones they are. To select these, type in the appropriate number (1-5) of the dice, reading from left to right. Once selected, the dice has the

letter H printed under it.

When you accept a throw, type the NAME of the score you have selected. The ones you require are shown at the top of the screen. If you can't make up a score, type PASS and then enter the NAME of the score you wish to forfeit.

When your score summary is shown the computer waits for quite a while to allow you to get a good look at your scores.

As the game progresses, there are

## VARIABLES

**AS,ABS,ANS** Strings for INPUTS.

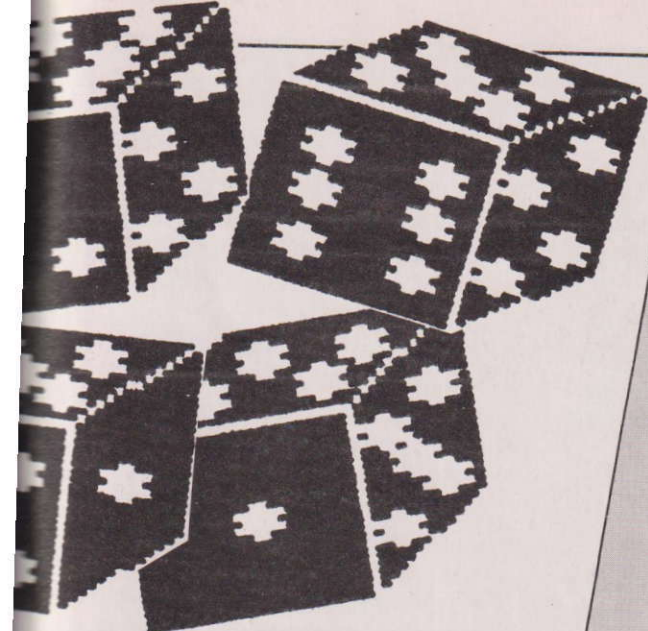
<b>A%</b>	Accept throw flag.
<b>a%</b>	Number of selected 'score'.
<b>B%</b>	Bonus.
<b>b%, BS</b>	General variable used in PROC PASS.
<b>C%</b>	Game loop counter.
<b>d%, dd%</b>	General loop variable.
<b>DR\$</b>	Names of players who have drawn a game.
<b>DS</b>	General string used in PROC got.
<b>e%</b>	Side value in PROC SIDE.
<b>f%</b>	Indicates whether or not you've already selected a score.
<b>F%, ff%</b>	General variable in RUN procedure.
<b>FA%</b>	Value of face of dice.
<b>G%</b>	General loop variable.
<b>g%</b>	Face value in PROC FACE.
<b>H%</b>	General loop variable.
<b>HOS, HO%</b>	Number of dice to be held.
<b>ho%</b>	Identity of dice selected to hold.
<b>I%</b>	Seeds random number generator.
<b>ID%</b>	Flag to indicate validity of throw.
<b>J%</b>	General loop variable.
<b>JS</b>	General string used in PROC PASS.
<b>k%</b>	General loop variable.
<b>K%</b>	Key used in sort routine.
<b>N%</b>	General loop variable.
<b>O%</b>	General variable in RUN procedure.
<b>P%</b>	Number of players.
<b>p%</b>	General loop variable.
<b>pts%</b>	Points assigned to a score.
<b>q%</b>	Controls colours.
<b>Q%, R%</b>	General loop variable.

<b>REPLY%</b>	Indicates a key pressed.
<b>S%</b>	Size of side of dice.
<b>s%</b>	General loop variable.
<b>ss%</b>	General variable used in PROC RUN.
<b>SI%</b>	Value of side of dice.
<b>throw%</b>	Throw counter.
<b>TT%, t%</b>	General variables in PROC RUN.
<b>T%</b>	Controls dice moving down screen.
<b>t%</b>	Value of top of dice in PROC TOP.
<b>to%</b>	Value of top of dice.
<b>U%</b>	General loop variable.
<b>V%</b>	Valid flag.
<b>WINS</b>	Name of winning player.
<b>X%</b>	Value of dice in PROC THDI.
<b>X%</b>	Coordinate (x) of dice.
<b>Y%</b>	Coordinate (y) of dice.
<b>Z%</b>	Delay variable.

## ARRAYS

<b>D%</b>	Holds value for each of the five dice.
<b>DPOS%</b>	Holds starting position for drawing dice on screen.
<b>hold%</b>	Contains dice to be held during each throw.
<b>NA\$</b>	Holds names of players.
<b>SC%</b>	Holds each players total score.
<b>SC\$</b>	Holds names of scores.
<b>SCSH%</b>	Three dimensional array holding scoresheet for each player. That is, SCSH%(A,B,C) where A='score', B=player and C=status of 'score' A for player B. C=0 gives the actual score obtained; C=1 gives details about that score, that is, whether available, forfeited or taken.





quite a lot of them. However, if you find this too long, simply alter line 1190 PROCDEL (1000) to a lower value.

When playing this game, you take your scores from the front face of the dice — that is, the dice facing you on the screen.

Because of the large number of variables in this game, the program had to be written in Mode 4, thus sacrificing colours. The program is a tight fit even then, so be careful not to leave spaces.

**SCORES**

Ones Any number of ones. Score total number of ones (1-5).  
 Twos Any number of twos. Score total number of twos (2-10).  
 Threes Any number of threes. Score total number of threes (3-15).  
 Fours Any number of fours. Score total number of fours (4-20).  
 Fives Any number of fives. Score total number of fives (5-25).  
 Sixes Any number of sixes. Score total number of sixes (6-30).  
 If you total of scores is 60 or more, you are awarded a bonus score of 60 points.

**SPECIAL SCORES**

Two Pairs Any two pairs of dice, such as 2 ones and 2 fives. Score 30 points.  
 Triple Run Any three dice the same. Score 30 points.  
 All dice different, either one to five or two to six. Score 40 points.  
 Four Kind Four of dice the same. Score 40 points.  
 Full House Two dice the same, and the other three the same as each other, such as 2 fours and 3 sixes. Score 40 points.  
 Hi Any fairly high score. It must be higher than Lo. Score total number shown on dice face.  
 Lo Any low score. It must be lower than Hi. Score total number shown on dice face.  
 Yams Five sixes. Score 60 points.

## PROCEDURES

<b>Lines 10-70</b>	Main body of program. Seeds random number generator, defines envelopes and calls procedures to play the game. Turns off cursor after each mode change.	<b>PROCDEL</b>	each players turn. Indicates scores obtained and scores forfeited.
<b>PROCnoplayers</b>	Prints titles. Asks if you want instructions and if so calls PROCINST. Asks for the number of players.	<b>PROCADD</b>	Delay procedure.
<b>PROCINIT</b>	Sets up arrays according to the number of players and initialises variables.	<b>PROCPLTBL</b>	Adds up each players scoresheet and decides whether to award bonus. Calls PROC-PLTBL.
<b>PROCNames</b>	Gets the names of the players.	<b>PROCTHDI</b>	Draws the final scores table and declares the winner. Asks if you want another game.
<b>PROCPLAY</b>	The main procedure, it controls the playing of the game. It calls PROCTHDI, PROCgot, PROCpass, PROCcheck, PROCerr, PROCDEL and PROCSCORES.	<b>PROCDRDI</b>	Throws dice, checking which ones (if any) are to be held. This procedure assigns the values to the dice.
<b>PROCcheck</b>	Checks if the score you've selected is valid. Calls relevant checking procedure.	<b>PROCDRDI</b>	Actually draws dice tumbling down the screen. Also erases dice where necessary. Sets hold% back to 0 before ending.
<b>PROCtot</b>	If a number score is selected, this procedure is called to add up the dice with the number.	<b>PROCplot</b>	Draws a cube at selected position.
<b>PROChi</b>	Checks the dice to see if a Hi score is valid.	<b>PROCLplot</b>	Draws the edges of the cube.
<b>PROClo</b>	Checks the dice to see if a Lo score is valid.	<b>PROCTILT</b>	Draws a tiltedcube and its edges.
<b>PROCvalid</b>	Checks SCSH% after validation. If there is no score for the player concerned, then ID% is returned false, that is not valid.	<b>PROCTOP</b>	Draws the spots on the top of the cube.
<b>PROCTPR</b>	Checks all combinations of dice for a Two Pairs score.	<b>PROCSIDE</b>	Draws the spots on the side of the cube.
<b>PROCTRI</b>	Checks dice for a Triple score.	<b>PROCFACE</b>	Calculates the values for the top and side of the cube given the value for the face.
<b>PROCFKD</b>	Checks dice for a Four Kind score.	<b>PROCSPTS</b>	Asks which dice you wish to hold and sets hold% accordingly.
<b>PROCYAMS</b>	Checks dice for a Yams score.	<b>PROCHOLD</b>	Asks which score you will forfeit and sets SCSH% to indicate this.
<b>PROCRUN</b>	Checks dice for a Run score.	<b>PROCPASS</b>	Erases dice from resting position on screen, ready for next throw.
<b>PROCFH</b>	Checks dice for a Full House score.	<b>PROCERASE</b>	Checks to see if you've selected a score which you've already had.
<b>PROCSCORES</b>	Prints the summary of scores at the end of	<b>PROCgot</b>	Instructions for the game.
		<b>PROCINST</b>	Lets you know if you've entered a name or score that the program does not recognise.
		<b>PROCerr</b>	Data containing names of scores.
		<b>Line 2580</b>	



## From Page 75

```

10 REM *****YAMS*****
20 REM BY J.AIZPURVS
30 REM (C) The Micro User
40 MODE7:VDU23;8202;0;0;0;IX=RND(
-TIME):PROCnoplayers:PROCINIT:PROCNAME
ES:MODE4:VDU23;8202;0;0;0;24,0;160;12
79;840;19,0,4,0,0,0;CLS:CL6
50 ENVELOPE1,1,-128,127,-128,1,1,1
,75,-50,-1,0,126,0
60 PROCPLAY:MODE7:VDU23;8202;0;0;0
;:PROCADD
70 END
80 DEFPROCnoplayers
90 PRINTTAB(12,8);CHR$141;CHR$136;
CHR$133;CHR$157;CHR$132;"YAMS ";CHR$3
2;CHR$156;CHR$137:PRINTTAB(12,9);CHR$
141;CHR$136;CHR$133;CHR$157;CHR$132;"
YAMS ";CHR$32;CHR$156;CHR$137:*FX21,0
100 PRINTTAB(3,14);CHR$131;"Do you
want instructions?";CHR$134;"Y/N":AN
$=GET$
110 IF AN$="Y" THEN PROCINST
120 *FX21,0
130 PRINTTAB(3,14);SPC(37);TAB(3,14
);CHR$130;:INPUT"HOW MANY PLAYERS ",P
%
140 IF P%<=1 OR P%>4 THEN SOUND0,-15,
3,10;SOUND1,0,172,10;GOTO130
150 ENDPROC
160 DEFPROCINIT
170 DIM DX(4),SCSHZ(14,P%-1,1),NA$(
P%-1),DPOSX(4),SC$(14),SCX(P%-1),hold
X(4)
180 SZ=100
190 FOR NX=0TO4:holdX(NX)=FALSE:NE
XT
200 FOR NX=0TO4:DX(NX)=0:NEXT
210 FOR NX=0TOPX-1:FOR pX=0TO14:FOR
QX=0TO1:SCSHZ(pX,NX,QX)=0:NEXT:NEXT:N
EXT
220 FOR dX=0TO4:READ DPOSX(dX):NEXT
230 DATA 400,600,800,1000,1200
240 RESTORE2580:FOR NX=1TO14:READ S
C$(NX):NEXT
250 FOR NX=0TOPX-1:SCX(NX)=0:NEXT
260 ENDPROC
270 DEFPROCNAME$
280 FOR NX=0TOPX-1:*FX21,0
290 PRINTTAB(0,17);SPC(40):PRINTTAB
(3,16);CHR$129;"WHAT IS YOUR NAME PLA
YER ";NX+1:INPUT,AN$:IF AN$="" THEN 2
90
300 NA$(NX)=LEFT$(AN$,6):NEXT
310 ENDPROC
320 DEFPROCPLAY:CX=0:REPEAT:CX=CX+1
330 FORNX=0TOPX-1:CLS:PRINTTAB(5,0)
;"D.K.";NA$(NX);" you need:"
340 FOR QX=1TO14:IFSCSHZ(QX,NX,1)=0

```

```

THENPRINTSC$(QX);",";
350 NEXT:AX=FALSE:throwX=0
360 REPEAT:PROCTHDI:UNTIL AX=TRUE O
R throwX=3
370 *FX21,0
380 PRINTTAB(0,24);SPC(160):INPUTTA
B(5,24)"What will you take from this
throw ",AN$:IF LEFT$(AN$,4)="PASS" TH
EN PROCPASS:GOTO430
390 fX=0:PROCgot(AN$):IF fX<>0 THEN
380
400 IDX=FALSE:PROCHECK(AN$)
410 IFIDX=FALSE AND aX=0 THEN PROCe
rr:PROCDEL(500):GOTO380 ELSEIFIDX=FAL
SE THEN PRINTTAB(5,26);AN$;" is not v
alid with this throw":SOUND0,-15,3,10
;SOUND1,0,172,10:PROCDEL(500):GOTO370
420 PRINTTAB(0,24);SPC(160):PRINTTA
B(5,24);"O.K.":SCSHZ(aX,NX,1)=1:VDU7:
PROCDEL(100)
430 PRINTTAB(0,0);SPC(200);TAB(0,24
);SPC(160):PROCSCORES(NA$(NX)):NEXT
440 UNTIL CX>=14
450 ENDPROC
460 DEFPROCHECK(A$):aX=0
470 IFA$="ONES" THENaX=1:PROCTot:PRO
Cvalid
480 IFA$="TWOS" THENaX=2:PROCTot:PRO
Cvalid
490 IFA$="THREES" THENaX=3:PROCTot:P
ROCvalid
500 IFA$="FOURS" THENaX=4:PROCTot:PR
OCvalid
510 IFA$="FIVES" THENaX=5:PROCTot:PR
OCvalid
520 IFA$="SIXES" THENaX=6:PROCTot:PR
OCvalid
530 IFA$="TWO PAIRS" THENaX=7:ptsX=3
0:PROCTPR:PROCvalid
540 IFA$="TRIPLE" THENaX=8:ptsX=30:P
ROCTRI:PROCvalid
550 IFA$="RUN" THENaX=9:ptsX=40:PROC
RUN:PROCvalid
560 IFA$="FOUR KIND" THENaX=10:ptsX=
40:PROCFKD:PROCvalid
570 IFA$="FULL HOUSE" THENaX=11:ptsX
=40:PROCFH:PROCvalid
580 IFA$="HI" THENaX=12:PROChi:PROCV
alid
590 IFA$="LO" THENaX=13:PROClo:PROCV
alid
600 IFA$="YAMS" THENaX=14:ptsX=60:PR
OCYAMS:PROCvalid
610 IF aX=0 THEN IDX=FALSE
620 ENDPROC
630 DEFPROCTot:JX=0
640 FOR RX=0TO4:IF DX(RX)=aX THEN J
X=JX+DX(RX)
650 NEXT:SCSHZ(aX,NX,0)=JX
660 ENDPROC

```

```

670 DEFPROChi
680 sX=0:IFAX=12AND SCSHZ(13,NX,0)=
0THENFORRX=0TO4:SCSHZ(aX,NX,0)=SCSHZ(
aX,NX,0)+DX(RX):NEXT ELSEFORRX=0TO4:s
X=sX+DX(RX):NEXT
690 IFSX=0THEN ENDPROC ELSEIFsX<=SC
SHZ(13,NX,0)THEN ENDPROC ELSESCSHZ(aX
,NX,0)=SCSHZ(aX,NX,0)+sX:ENDPROC
700 DEFPROClo
710 sX=0:IFSCSHZ(12,NX,0)=0THENFORR
X=0TO4:SCSHZ(aX,NX,0)=SCSHZ(aX,NX,0)+
DX(RX):NEXT ELSEFORRX=0TO4:sX=sX+DX(R
X):NEXT
720 IFSX=0THEN ENDPROC ELSE IFSX>=S
CSHZ(12,NX,0)THEN ENDPROC ELSESCSHZ(a
X,NX,0)=SCSHZ(aX,NX,0)+sX:ENDPROC
730 DEFPROCvalid
740 IFSCSHZ(aX,NX,0)=0THENIDX=FALSE
ELSEIDX=TRUE
750 ENDPROC
760 DEFPROCTPR:VX=FALSE
770 IF(DX(0)=DX(1)AND(DX(2)=DX(3)OR
DX(2)=DX(4)OR DX(3)=DX(4)))OR(DX(0)=
DX(2)AND(DX(1)=DX(3)OR DX(1)=DX(4)OR
DX(3)=DX(4)))OR(DX(0)=DX(3)AND(DX(1)=
DX(2)OR DX(1)=DX(4)OR DX(2)=DX(4)))TH
ENVX=TRUE
780 IF(DX(0)=DX(4)AND(DX(1)=DX(2)OR
DX(1)=DX(3)OR DX(2)=DX(3)))OR(DX(1)=
DX(2)AND DX(3)=DX(4)OR(DX(1)=DX(3)AN
D DX(2)=DX(4)OR(DX(1)=DX(4)AND DX(2)
=DX(3)))THENVX=TRUE
790 IFVX=TRUE THEN SCSHZ(aX,NX,0)=p
tsX
800 ENDPROC
810 DEFPROCTRI:VX=FALSE
820 IF(DX(0)=DX(1)ANDDX(0)=DX(2)OR
(DX(1)=DX(2)ANDDX(1)=DX(3)OR(DX(2)=D
X(3)ANDDX(2)=DX(4)OR(DX(3)=DX(4)AND
X(3)=DX(0)OR(DX(4)=DX(0)AND DX(4)=DX
(1)))THEN VX=TRUE
830 IF(DX(0)=DX(1)ANDDX(0)=DX(3)OR
(DX(1)=DX(2)ANDDX(1)=DX(4)OR(DX(0)=D
X(2)ANDDX(0)=DX(3)OR(DX(0)=DX(2)AND
X(0)=DX(4)OR(DX(1)=DX(3)ANDDX(1)=DX(
4)OR(DX(1)=DX(3)ANDDX(1)=DX(0))THEN
VX=TRUE
840 IFVX=TRUE THEN SCSHZ(aX,NX,0)=p
tsX
850 ENDPROC
860 DEFPROCFKD:VX=FALSE
870 IF(DX(0)=DX(1)ANDDX(0)=DX(2)AND
DX(0)=DX(3)OR(DX(1)=DX(2)ANDDX(1)=DX
(3)ANDDX(1)=DX(4)OR(DX(2)=DX(3)ANDDX
(2)=DX(4)ANDDX(2)=DX(0)OR(DX(3)=DX(4
)ANDDX(3)=DX(0)ANDDX(3)=DX(1)OR(DX(4
)=DX(0)ANDDX(4)=DX(1)ANDDX(4)=DX(2))T
HENVX=TRUE
880 IFVX=TRUE THEN SCSHZ(aX,NX,0)=p
tsX

```



```

890 ENDPROC
900 DEFPROC YAMS
910 IFDZ(0)=6ANDDZ(1)=6ANDDZ(2)=6AND
DZ(3)=6ANDDZ(4)=6THENSESHZ(aZ,NZ,0)=
ptsZ
920 ENDPROC
930 DEFPROC RUN:DZ=0:TTZ=0:tZ=0:FZ=0
:ffZ=0:ssZ=0
940 FORRZ=0TO4:IFDZ(RZ)=1THENDZ=DZ+
1
950 NEXT:IFDZ>1THENENDPROC
960 FORRZ=0TO4:IFDZ(RZ)=2THENTT=TT
Z+1
970 NEXT:IFTT<>1THENENDPROC
980 FORRZ=0TO4:IFDZ(RZ)=3THENTZ=tZ+
1
990 NEXT:IFTZ<>1THENENDPROC
1000 FORRZ=0TO4:IFDZ(RZ)=4THENFZ=FZ+
1
1010 NEXT:IFFZ<>1THENENDPROC
1020 FORRZ=0TO4:IFDZ(RZ)=5THENffZ=ff
Z+1
1030 NEXT:IFFFZ<>1THENENDPROC
1040 FORRZ=0TO4:IFDZ(RZ)=6THENssZ=ss
Z+1
1050 NEXT:IF(DZ=1ANDssZ<>0)OR(DZ=0AND
ssZ<>1)THENENDPROC
1060 SCSHZ(aZ,NZ,0)=ptsZ:ENDPROC
1070 DEFPROC FH:VZ=FALSE
1080 IF(DZ(0)=DZ(1)ANDDZ(0)=DZ(2)AND
DZ(3)<>DZ(0)ANDDZ(3)=DZ(4))OR(DZ(1)=D
Z(2)ANDDZ(1)=DZ(3)ANDDZ(0)<>DZ(1)ANDD
Z(0)=DZ(4))OR(DZ(2)=DZ(3)ANDDZ(2)=DZ(
4)ANDDZ(0)<>DZ(2)ANDDZ(0)=DZ(1))THEN
VZ=TRUE
1090 IF(DZ(3)=DZ(4)ANDDZ(3)=DZ(0)AND
DZ(1)<>DZ(0)ANDDZ(1)=DZ(2))OR(DZ(4)=D
Z(0)ANDDZ(4)=DZ(1)ANDDZ(2)<>DZ(4)AND
DZ(2)=DZ(3))OR(DZ(0)=DZ(1)ANDDZ(0)=DZ
(3)ANDDZ(2)<>DZ(0)ANDDZ(2)=DZ(4))THEN
VZ=TRUE
1100 IF(DZ(1)=DZ(2)ANDDZ(1)=DZ(4)AND
DZ(0)<>DZ(1)ANDDZ(0)=DZ(3))OR(DZ(0)=D
Z(2)ANDDZ(0)=DZ(3)ANDDZ(1)<>DZ(0)ANDD
Z(1)=DZ(4))OR(DZ(0)=DZ(2)ANDDZ(0)=DZ(
4)ANDDZ(1)<>DZ(1)ANDDZ(1)=DZ(3))THEN
VZ=TRUE
1110 IF(DZ(0)=DZ(2)ANDDZ(0)=DZ(4)AND
DZ(1)<>DZ(0)ANDDZ(1)=DZ(3))OR(DZ(1)=D
Z(3)ANDDZ(1)=DZ(4)ANDDZ(0)<>DZ(1)ANDD
Z(0)=DZ(2))OR(DZ(1)=DZ(3)ANDDZ(1)=DZ(
0)ANDDZ(2)<>DZ(1)ANDDZ(2)=DZ(4))THEN
VZ=TRUE
1120 IFVZ=TRUE THEN SCSHZ(aZ,NZ,0)=p
tsZ
1130 ENDPROC
1140 DEFPROC SCORES(NA$(NZ)):PRINTTAB
(0,0):SPC(200)
1150 PRINTTAB(0,0):"O.K.":NA$(NZ):"
Your scores so far are:":FORRZ=1TO14:

```

```

IFSCSHZ(RZ,NZ,1)=1THENPRINT:SC$(RZ):"
";SCSHZ(RZ,NZ,0):";"
1160 NEXT
1170 PRINTTAB(5,24):SPC(160):TAB(5,2
4):"Your forfeits:":PRINTTAB(5,25):F
ORRZ=1TO14:IFSCSHZ(RZ,NZ,1)=-1THEN PR
INT:SC$(RZ):";"
1180 NEXT
1190 PROCDEL(1000):CLS:ENDPROC
1200 DEFPROC DEL(ZZ):TIME=0:REPEAT:UN
TIL TIME=ZZ:ENDPROC
1210 DEFPROC ADD:BZ=60
1220 FORGZ=0TOPX-1:FORHZ=1TO14:SCZ(6
Z)=SCZ(6Z)+SCSHZ(HZ,6Z,0)
1230 IFHZ=6AND SCZ(6Z)>60THENSECZ(6Z
)=SCZ(6Z)+BZ:SCSHZ(0,6Z,0)=60:SCSHZ(0
,6Z,1)=1
1240 NEXT:NEXT:PROCPLTBL:ENDPROC
1250 DEFPROC PLTBL
1260 PRINTTAB(11,0):CHR$(141):CHR$(13
3):CHR$(136):"FINAL SCORES":CHR$(137):PR
INTTAB(11,1):CHR$(141):CHR$(133):CHR$(1
36):"FINAL SCORES":CHR$(137)
1270 FORRZ=1TOPX:PRINTTAB(0,3):CHR$(
135):CHR$(157):TAB(5+RZ*7,3):CHR$(128+R
Z):NA$(RZ-1):NEXT
1280 FORGZ=1TO6:FORRZ=1TOPX:PRINTTAB
(0,6Z+3):CHR$(135):CHR$(157):CHR$(133):
SC$(6Z):TAB(7+RZ*7,6Z+3):CHR$(128+RZ)
:SCSHZ(6Z,RZ-1,0):NEXT:NEXT

```

```

1290 FORRZ=1TOPX:PRINTTAB(0,10):CHR$
135:CHR$(157):CHR$(133):"BONUS":TAB(7+RZ*
7,10):CHR$(128+RZ):SCSHZ(0,RZ-1,0):NE
XT
1300 FORGZ=7TO14:FORRZ=1TOPX:PRINTTA
B(0,6Z+4):CHR$(135):CHR$(157):CHR$(133):SC
(6Z):TAB(7+RZ*7,6Z+4):CHR$(128+RZ):SC
SHZ(6Z,RZ-1,0):NEXT:NEXT
1310 FORRZ=1TOPX:PRINTTAB(0,6Z+4):CH
R$(135):CHR$(157):CHR$(133):"TOTAL":TAB(7+R
Z*7,6Z+4):CHR$(128+RZ):SCZ(RZ-1):NEXT
:PROCVIN:ENDPROC
1320 DEFPROC WIN:WIN$="":DR$="":KZ=0:
FORRZ=1TOPX-1:IFSCZ(KZ)<SCZ(RZ)THENKZ
=RZ
1330 NEXT:FORRZ=0TOPX-1:IFRZ=KZTHEN1
350
1340 IFSCZ(RZ)=SCZ(KZ)THENDR$=DR$+NA
$(RZ)+" and "
1350 NEXT:IFDR$<>""THENDR$=DR$+NA$(K
Z):PRINTCHR$(133):"It's a draw between"
:CHR$(136):CHR$(134):DR$:CHR$(137)ELSEWIN$
=NA$(KZ):PRINTTAB(5):CHR$(133):CHR$(136):
WIN$:CHR$(137):CHR$(134):"wins this game"
1360 *FX21,0
1370 PRINTCHR$(130):"Do you want to pl
ay again?":A$=GET$:IF A$="Y"THENRUN
1380 ENDPROC

```



## From Page 77

```

1390 DEFPROCTHDI:throwX=throwX+1:FOR
ddX=0T04:IF holdX(ddX)=TRUE THEN NEXT
ELSE XX=RND(6):DX(ddX)=XX:NEXT
1400 PROCDRDI:FX21,0
1410 PRINTTAB(10,6);"Y = ACCEPT this
throw";TAB(10,7);"H = HOLD some dice
";TAB(6,8);"SPACE = THROW all dice ag
ain"
1420 REPLYX=GET:IFREPLYX=32THENAX=FA
LSE ELSEIFREPLYX=89THENAX=TRUE ELSE I
FREPLYX=72THEN AX=FALSE:IFthrowX<>3 T
HENPROCHOLD ELSE SOUND0,-15,3,10:SOUN
D1,0,172,10:BOTD1420
1430 PRINTTAB(10,6);SPC(22);TAB(10,7
);SPC(19);TAB(6,8);SPC(29):ENDPROC
1440 DEFPROCARDI:IFthrowX=1THEN1450
ELSEFORpX=0T04:IFholdX(pX)=TRUE THEN
NEXT ELSE PROCERASE(pX):NEXT
1450 FORpX=0T04:IFholdX(pX)=TRUE THE
N 1560
1460 YX=750
1470 FORTX=0T05:FORqX=1T00STEP-1:GCO
L0,qX
1480 XX=DPOSX(pX)-TX*50:VDU29,XX,YX;
:PROCPLOT:GCOL0,0:PROCLPLOT:IFqX=1 TH
EN GCOL0,qX
1490 SOUND1,1,2,5
1500 NEXT
1510 FORqX=1T00 STEP-1:GCOL0,qX
1520 XX=DPOSX(pX)-TX*50:VDU29,XX,YX;
:PROCTILT:NEXT:YX=YX-50
1530 NEXT
1540 XX=DPOSX(pX)-300:VDU29,XX,YX;6
COL0,1:PROCPLOT:GCOL0,0:PROCLPLOT:PRO
CSPOTS(pX)
1550 SOUND1,1,2,5:FX21,5
1560 NEXT
1570 FORkX=0T04:holdX(kX)=FALSE:PRIN
TTAB(kX*6+3,22);" ";NEXT:ENDPROC
1580 DEFPROCPLT
1590 MOVE0,SX:DRWSX,SX:PLOT85,SX/2,
SX/2:DRAW-SX/2,SX-SX/2:PLOT85,0,SX:DR
AWSX,0:DRWSX,SX:PLOT85,SX/2,SX/2:DRA
WSX/2,-SX/2:PLOT85,SX,0:DRAW-SX/2,SX-
SX/2:DRWSX/2,SX/2:PLOT85,-SX/2,-SX/2
:DRWSX/2,-SX/2:PLOT85,SX/2,SX/2
1600 ENDPROC
1610 DEFPROCPLT
1620 MOVE-SX/2,SX-SX/2:DRWSX/2,SX/2
:DRWSX,SX:MOVESX/2,SX/2:DRWSX/2,-SX
/2:ENDPROC
1630 DEFPROCTILT
1640 MOVE-40,0:DRAW-140,0:DRAW-75,-7
5:DRAW25,-75:DRAW85,0:DRAW25,75:DRAW-
75,75:DRAW-140,0:MOVE-40,0:PLOT85,25,
75
1650 PLOT85,85,0:MOVE-140,0:MOVE-75,
75:PLOT85,25,75:MOVE85,0:MOVE25,-75:P

```

```

LOT85,-40,0:PLOT85,-75,-75:PLOT85,-14
0,0:GCOL0,0
1660 DRAW-20,0:DRAW25/2,-(75/2):MOVE
-20,0:DRAW25/2,75/2:IF qX=1 THEN GCOL
0,qX
1670 ENDPROC
1680 DEFPROCTOP(tX):VDU5
1690 IFtX=1THENMOVE10,87:VDU42
1700 IFtX=2THENMOVE-30,80:VDU42:MOVE
45,95:VDU42
1710 IFtX=3THENMOVE10,86:VDU42:MOVE-
32,80:VDU42:MOVE44,95:VDU42
1720 IFtX=4THENMOVE-25,80:VDU42:MOVE
45,97:VDU42:MOVE0,97:VDU42:MOVE27,80:
VDU42
1730 IFtX=5THENMOVE-27,80:VDU42:MOVE
47,100:VDU42:MOVE-2,100:VDU42:MOVE29,
80:VDU42:MOVE11,89:VDU42
1740 IFtX=6THENMOVE-29,76:VDU42:MOVE
54,102:VDU42:MOVE9,102:VDU42:MOVE23,7
6:VDU42:MOVE-12,89:VDU42:MOVE36,89:VD
U42
1750 VDU4:ENDPROC
1760 DEFPROCSDI(eX):VDU5
1770 IFeX=1THENMOVE61,34:VDU42
1780 IFeX=2THENMOVE49,48:VDU42:MOVE7
4,17:VDU42
1790 IFeX=3THENMOVE73,70:VDU42:MOVE4
8,-5:VDU42:MOVE60,32:VDU42
1800 IFeX=4THENMOVE49,45:VDU42:MOVE7
4,70:VDU42:MOVE49,-5:VDU42:MOVE74,20:
VDU42
1810 IFeX=5THENMOVE48,47:VDU42:MOVE7
3,68:VDU42:MOVE48,-3:VDU42:MOVE73,18:
VDU42:MOVE60,32:VDU42
1820 IFeX=6THENMOVE48,45:VDU42:MOVE7
3,70:VDU42:MOVE48,20:VDU42:MOVE73,45:
VDU42:MOVE48,-5:VDU42:MOVE73,20:VDU42
1830 VDU4:ENDPROC
1840 DEFPROCFCF(gX):VDU5
1850 IFgX=1THENMOVE-11,12:VDU42
1860 IFgX=2THENMOVE-38,37:VDU42:MOVE
13,-14:VDU42
1870 IFgX=3THENMOVE-38,37:VDU42:MOVE
13,-14:VDU42:MOVE-11,12:VDU42
1880 IFgX=4THENMOVE-38,37:VDU42:MOVE
13,-15:VDU42:MOVE-38,-15:VDU42:MOVE13
,37:VDU42
1890 IFgX=5THENMOVE-38,37:VDU42:MOVE
13,-15:VDU42:MOVE-38,-15:VDU42:MOVE13
,37:VDU42:MOVE-11,12:VDU42
1900 IFgX=6THENMOVE-38,37:VDU42:MOVE
13,-15:VDU42:MOVE-38,-15:VDU42:MOVE13
,37:VDU42:MOVE-38,12:VDU42:MOVE13,12:
VDU42
1910 VDU4:ENDPROC
1920 DEFPROCSPOTS(NX):GCOL0,0:FAZ=DX
(NX):IFFAX<>6THEN1950
1930 toX=RND(5):IFtoX=1THEN1930
1940 IFtoX=2THENSIX=3 ELSEIFtoX=3THE

```

```

NSIX=5ELSEIFtoX=4THENSIX=2 ELSESESIX=4
1950 IFFAX<>5THEN1980
1960 toX=RND(6):IFtoX=FAZ ORtoX=2THE
N1960
1970 IFtoX=1THENSIX=4 ELSEIFtoX=4THE
NSIX=6 ELSEIFtoX=6THENSIX=3 ELSESESIX=1
1980 IFFAX<>4THEN2010
1990 toX=RND(6):IFtoX=FAZ ORtoX=3THE
N1990
2000 IFtoX=1THENSIX=2ELSEIFtoX=2THE
NSIX=6 ELSEIFtoX=6THENSIX=5 ELSESESIX=1
2010 IFFAX<>3THEN2040
2020 toX=RND(6):IFtoX=FAZ ORtoX=4THE
N2020
2030 IFtoX=1THENSIX=5 ELSEIFtoX=2THE
NSIX=1 ELSEIFtoX=5THENSIX=6 ELSESESIX=2
2040 IFFAX<>2THEN2070
2050 toX=RND(6):IFtoX=FAZ ORtoX=5THE
N2050
2060 IFtoX=1THENSIX=3 ELSEIFtoX=3THE
NSIX=6 ELSEIFtoX=4THENSIX=1 ELSESESIX=4
2070 IFFAX<>1THEN2100
2080 toX=RND(5):IFtoX=FAZ THEN2080
2090 IFtoX=2THENSIX=4 ELSEIFtoX=3THE
NSIX=2 ELSEIFtoX=4THENSIX=5 ELSESESIX=3
2100 XX=DPOSX(NX)-300:VDU29,XX,YX;6
COL0,0:PROCTOP(toX):PROCSIDE(SIX):PRO
CFACE(FAZ):ENDPROC
2110 DEFPROCHOLD:FX21,0
2120 PRINTTAB(0,24);SPC(160);TAB(5,2
4);"How many dice do you want to hold
?":HOX=GET:HOX=VAL(HOX)
2130 IFHOX<1 OR HOX>5 THEN SOUND0,-1
5,3,10:SOUND1,0,172,10:BOTD2120 ELSE
VDU7
2140 FORUX=1TOHOX:FX21,0
2150 PRINTTAB(0,24);SPC(160);TAB(5,2
4);"Which dice would you like to keep
?":hoX=GET:PRINTTAB(0,24);SPC(160):VD
U7:IF hoX<49 OR hoX>53 THEN 2150
2160 holdX(hoX-49)=TRUE:PRINTTAB(ho
X-49)*6+3,22);"H":NEXT
2170 ENDPROC
2180 DEFPROCPCPASS:RESTORE2580:J$="" :b
X=0:FX21,0
2190 PRINTTAB(0,24);SPC(160):INPUTTA
B(5,24)"What will you forfeit",B$
2200 REPEAT:READJ$:bX=bX+1:UNTILJ$=
B$
2210 SCSHX(bX,NX,1)=-1
2220 PRINTTAB(0,24);SPC(160):ENDPROC
2230 DEFPROCERASE(aX)
2240 XX=DPOSX(aX)-300:VDU29,XX,YX;6
COL0,0:PROCPLOT
2250 ENDPROC
2260 DEFPROCgot(D$):RESTORE2580
2270 J$="" :bX=0
2280 REPEAT:READJ$:bX=bX+1:UNTILJ$=
D$ OR bX=14:IF bX=14 AND J$<>D$ THEN
ENDPROC

```



```

2290 IF SCSHX(bX,NZ,1)<>0 THEN fX=S
CSHX(bX,NZ,1):PRINTTAB(0,24);SPC(160)
;TAB(5,24);"You've already had this o
ne":SOUND0,-15,3,10:SOUND1,0,172,10:P
ROCDL(300)

```

```

2300 ENDPROC

```

```

2310 DEFPROCINST:FX21,0

```

```

2320 CLS:PRINTTAB(3,3);"This game i
s for 2 to 4 players.You have five d
ice and you must use them to accrue s
cores (details later).You must also
attempt to get your 'number' scoresto
a total of 60 or more to obtain your
Bonus."

```

```

2330 PRINT;"Player one throws first
.You have three throws,and after the
first throw, you may hold any numb
er of dice.To do this, INPUT how many
dice you want to hold,andthen which
ones they are (1-5) reading left to
right."

```

```

2340 PRINT;"Keep CAPS LOCK on durin
g this game, as DATA is stored in up
per case."''TAB(5);CHR$130;CHR$136;
"PRESS ANY KEY TO CONTINUE.....":AB$=
GET$:CLS

```

```

2350 PRINTTAB(3,3);"To";CHR$129;"ACC
EPT";CHR$135;"a throw press Y":TAB(3)
;"To";CHR$130;"REJECT";CHR$135;"a thr
ow press SPACE BAR":TAB(3);"To";CHR$1
31;"HOLD";CHR$135;"any dice press H"

```

```

2360 PRINT;"Once you've accepted yo
ur throw or had three goes, the Comp
uter asks you to type in the score
you wish to take."

```

```

2370 PRINT;"The Computer will then
check your Answer to see if it is vali
d.If so,the score will be calculate
d and added to your scoresheet.The
game then passes on to"

```

```

2380 PRINT;"the next player,and cont
inues going round until all the s
cores are filled.""You also have the
option to";CHR$134;"PASS""To do thi
s type PASS when you are asked what y
ou want to take from the throw,"

```

```

2390 PRINT;"then decide which score
you will forfeitThese are displayed
at the bottom of thescreen."

```

```

2400 PRINTTAB(5);CHR$133;CHR$136;"P
RESS ANY KEY TO CONTINUE.....":AB$=6
ET$:CLS

```

```

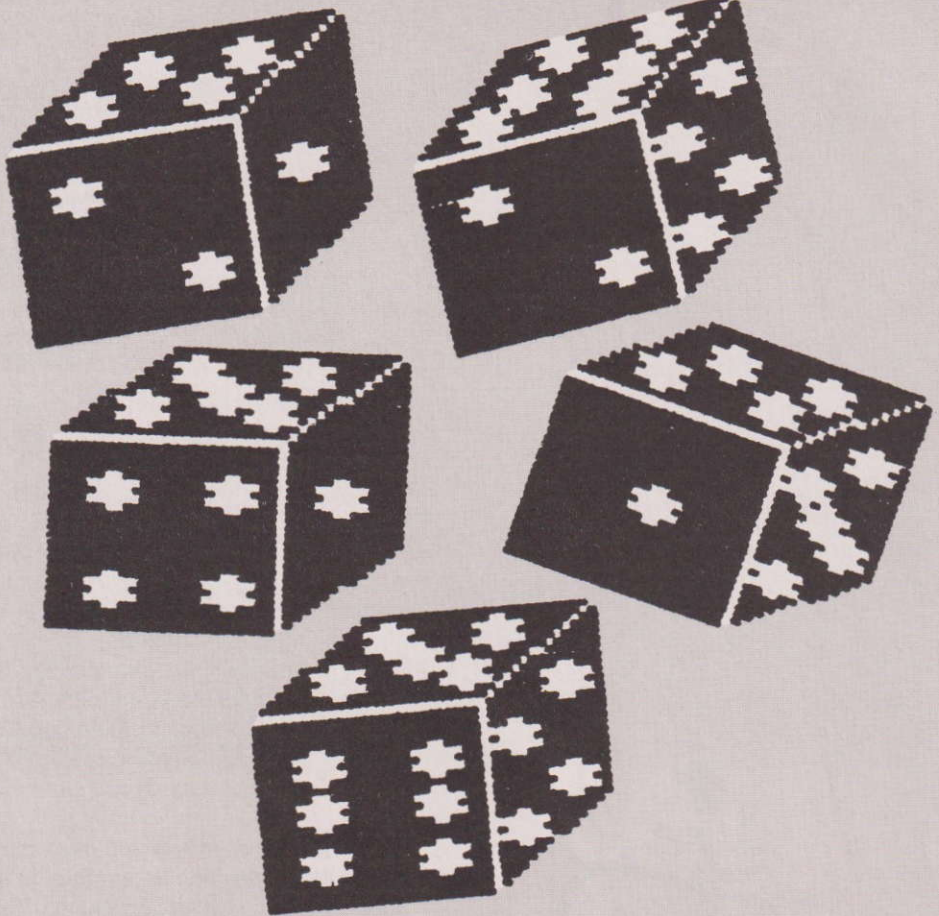
2410 PRINTTAB(8,3);CHR$141;CHR$135;
CHR$157;CHR$132;"number SCORES";CHR$3
2;CHR$156;TAB(8,4);CHR$141;CHR$135;CH
R$157;CHR$132;"number SCORES";CHR$32;
CHR$156

```

```

2420 PRINTTAB(3,7);CHR$129;"ONES...
.points depend on number of"CHR$129;
"ones,from 1 - 5.";TAB(3,10);CHR$130;

```



```

"TWOS....as above,points from 2 - 10.
"

```

```

2430 PRINTTAB(3,12);CHR$131;"THREES
..as above,points from 3 - 15.";TAB(3
,14);CHR$133;"FOURS..as above,points
from 4-20."

```

```

2440 PRINTTAB(3,16);CHR$134;"FIVES.
..as above,points from 5-25.";TAB(3,1
8);CHR$129;"SIXES..as above,points fr
om 6-30.";TAB(5,24);CHR$135;CHR$157;C
HR$132;"PRESS ANY KEY TO CONTINUE....
"CHR$32;CHR$156;AB$=GET$:CLS

```

```

2450 PRINTTAB(8,3);CHR$141;CHR$135;C
HR$157;CHR$132;"special SCORES";CHR$3
2;CHR$156;TAB(8,4);CHR$141;CHR$135;CH
R$157;CHR$132;"special SCORES";CHR$32
;CHR$156

```

```

2460 PRINTTAB(3,6);CHR$130;"TWO PAI
RS..of 5 dice,2 pairs i.e 2"CHR$130;
"ones and 2 fives.Points 30.";TAB(3,8
);CHR$131;"TRIPLE..any three dice the
same."CHR$131;"Points 30.";TAB(3,10
);CHR$133;"RUN....Each dice different
, 1-5 or"

```

```

2470 PRINTCHR$133;"2-6, points 40.";
TAB(3,12);CHR$134;"FOUR KIND..Any fou
r dice the same."CHR$134;"Points 40.
"

```

```

2480 PRINTTAB(3,14);CHR$129;"FULL H
OUSE..two dice the same and"CHR$129;
"the other three the same,i.e.2 threes"CHR$129;"and 3 fives.Points 40.";T
AB(3,17);CHR$130;"HI....Any fairly hi
gh score.Points"

```

```

2490 PRINTCHR$130;"total of all dice
.MUST be higher than"CHR$130;"LO."

```

```

2500 PRINTTAB(3,20);CHR$131;"LO....A
ny low score.Points total of"CHR$131
;"dice.MUST be lower than HI.";TAB(3,
22);CHR$133;"YAMS...all dice showing
sixes.Points"CHR$133;"60."

```

```

2510 PRINTTAB(5,24);CHR$135;CHR$157
;CHR$132;"PRESS ANY KEY TO CONTINUE..
."CHR$32;CHR$156;AB$=GET$:CLS

```

```

2520 PRINTTAB(6,11);CHR$141;CHR$134;
CHR$157;CHR$132;"You take your scores
";CHR$32;CHR$156;TAB(6,12);CHR$141;C
HR$134;CHR$157;CHR$132;"You take your
scores ";CHR$32;CHR$156

```

```

2530 PRINTTAB(1,14);CHR$141;CHR$134;
CHR$157;CHR$132;"from the front face
of the dice.";CHR$32;CHR$156;TAB(1,15
);CHR$141;CHR$134;CHR$157;CHR$132;"fr
om the front face of the dice.";CHR$3
2;CHR$156

```

```

2540 PRINTTAB(3,24);CHR$131;CHR$157;
CHR$136;CHR$129;"PRESS ANY KEY TO CON
TINUE...."CHR$32;CHR$156;AB$=GET$:CL
S;ENDPROC

```

```

2550 DEFPROCerr

```

```

2560 PRINTTAB(0,24);SPC(160);TAB(5,
24);"You've misspelt something here o
r left a space... try again!"

```

```

2570 SOUND0,-15,3,10:SOUND1,0,172,10
:ENDPROC

```

```

2580 DATA ONES,TWOS,THREES,FOURS,FIV
ES,SIXES,TWO PAIRS,TRIPLE,RUN,FOUR KI
ND,FULL HOUSE,HI,LO,YAMS

```



# George Cane puts an old favourite on your monitor

**PATIENCE** is a BBC Micro version of the long established card game which, by its nature, is for one player.

On screen you see seven vertical files of cards, all face down bar the last card of each file. The number of cards placed in each decreases by one as you look from left to right, resulting in there being only one card in file seven.

The remaining 24 cards form the pack. For convenience the pack is also laid in a vertical file on the extreme left of the screen and is numbered 1. The seven files of game cards are numbered 2 to 8.

Using these numbers you must move cards from one file to another in an attempt to finish with every card face up, and all the pack used up.

The all-important rule is that cards

must be placed in the order "low" on "high" and in alternating colours. This means for example that a red 4 can only be placed on a black 5, a black queen can only be placed on a red king and so on.

When any vertical file becomes empty as a result of moving its last card, that empty space can be filled, but only with a king.

Just keep moving the cards until either you succeed in using all the pack and turning all 52 cards face up in ascending order, or you cannot move any card according to the rules.

By the way, the computer won't let you make an illegal move — though we're sure you wouldn't even try!

All you need is a little bit of luck ... and an awful lot of patience.

```

LD 10 REM Patience
20 REM (C) The Micro User
30 *FX11,0
40 MODE1:VDU19,2,2,0,0,0:COLOUR130
:CLS:PROCTEXT:VDU26:VDU23,1,0;0;0;0;:
PROCINTRO:FORJ=0T012:READA,B,C,D,E,F,
6,H:VDU23,224+J,A,B,C,D,E,F,6,H:NEXT:
DIMC$(52),F$(52),L$(7),N$(7),V$(
52),V$(13):FORA=1T013:READV$(A):NEXT
:V$(10)=CHR$236
50 T$=CHR$224+CHR$225+CHR$225+CHR$
226:M$=CHR$227+CHR$228+CHR$228+CHR$22
9:B$=CHR$230+" "+CHR$231:FORA=0T07:R
EADN$(A):NEXT
60 VDU26:COLOUR130:PROCTEXT:PRINT:
K$="Press any key to start":PROC$AY(3
):REPEATQ=RND:UNTILINKEY$(1)<>"
70 K$="Do you like to play strictl
y?":PRINT':PROC$AY(3):REPEATQ=RND:K
$=INKEY$(1):UNTILK$<>"":J=0:IFK$="Y"J
=1
80 PRINT':K$="Would you like a di
fficult game?":PROC$AY(3):REPEATQ=RN
D:K$=INKEY$(1):UNTILK$<>"":K=8:IFK$="
Y"K=9
90 PROC$SHUFFLE:PROC$DEAL:*FX21,0
100 DNERRORGOTO590
110 PROC$PLAY:IFK$="A"ORK$="a"THEN90
:ELSEIFK$="C"ORK$="c"THEN70:ELSEGOTO1
00

```

```

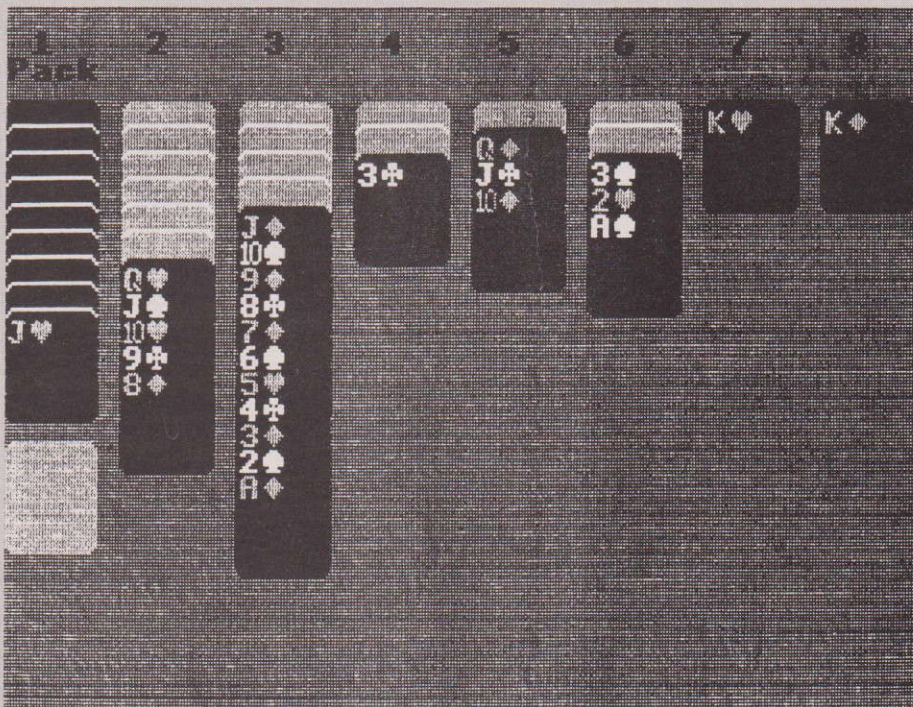
120 DEFPROCWINDOW:B$=W$*5:VDU28,B$,
28,B$+3,3:COLOUR130:CLS:ENDPROC
130 DEFPROC1ONG:VDU28,0,28,3,0:COLO
UR3:COLOUR130:CLS:PRINT' 1 Pack";E
NDPROC
140 DEFPROCTEXT:VDU28,0,31,39,29:CO
LOUR128:CLS:ENDPROC
150 DEFPROC$AY(C):COLOURC:PRINTTAB(
20-(LEN(K$)DIV2));K$:ENDPROC
160 DEFPROC$SHUFFLE:VDU26:COLOUR130:
CLS:Q=RND(3):PROCTEXT:PRINT:IFH=0K$="
Remember, Press A for a new game.":PR
OC$AY(3):PRINT':H=1
170 IFQ<2K$="Just a quick shuffle."
:ELSEIFQ<3K$="I'm shuffling the pack.
":ELSEK$="I'm giving the cards a good
shuffle."
180 PROC$AY(3):FORA=0T052:F$(A)=""
:NEXT:FORA=1T052:REPEAT:FORB=0TQ:D$=R
ND(52):NEXT:UNTILF$(D$)=""C$=(A-1)DI
V13:V$(D$)=A MOD13:F$(D$)=V$(V$(D$)+1
)+CHR$(232+C$)+" ":C$(D$)=C$AND1:NEX
T:ENDPROC
190 DEFPROC$PACK:IFLX(0)=P$ENDPROC:E
LSECOLOUR129:COLOUR2:PRINTT$
"B$":ENDPROC
200 DEFPROC$ONE:COLOUR131:COLOUR2:PR
INTT$:E$=VAL(LEFT$(L$(W$),2)):COLOUR
C$(E$):PRINTF$(E$) " ";COLOUR2
:PRINTB$;ENDPROC

```



$1 \leq i \leq n$

---

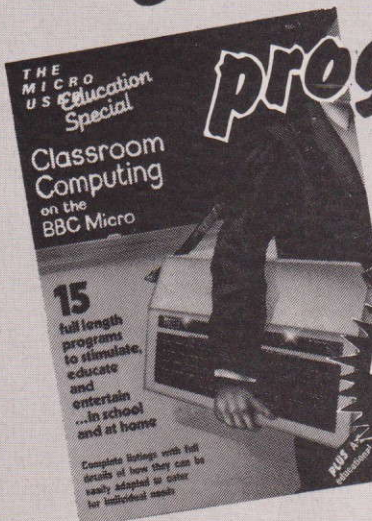


or press 9 to turn the pack.

Micro User Games Special



# The best value ever in educational programs!



**Available now at your newsagent**

A must for every parent and teacher, the Micro User Educational Special will help turn your BBC Micro into a learning centre.

Its 15 original programs have been specially chosen to combine educational validity with sheer good fun.

Tested in the classroom, and ranging in scope from pre-reading to sixth form maths, each program is fully listed and explained in detail – in the clear style for which The Micro User is renowned.

And we've managed to squeeze all the programs from this Micro User Special into a software package that's overflowing with educational value.

Can you or your children afford to miss these vital educational resources?

Please send me the following:

☐ I enclose cheque made payable to Database Publications Ltd. for £ \_\_\_\_\_

☐ I wish to pay by  
☐ Access ☐ Visa

No. \_\_\_\_\_

Expiry date \_\_\_\_\_

Signed \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

- ☐ Education Special No.1 £1.95
- ☐ BBC 'B' cassette £5.95
- ☐ BBC 40-track disc £7.95
- ☐ BBC 80-track disc £7.95

- Magazine PLUS software:
- ☐ Magazine and cassette £7
  - ☐ Magazine and disc £9

Please tick box

Post to: Database Publications, Europa House,  
68 Chester Road, Hazel Grove, Stockport SK7 5NY.

# BINGO

UP TO 4 PLAYERS AT ONCE

A FIVE YEAR OLD OR GRANNY CAN PLAY

1

**£100.00 TO BE WON EACH MONTH**  
FOR FURTHER DETAILS RING TREVOR ON 091 414 4611

66

**ACTUALLY SHOUTS THE NUMBER OUT**  
CURRAH COMPATIBLE FOR CM64 & SPECTRUM

**FUN FOR ALL THE FAMILY**

**TYNESOFT**  
COMPUTER SOFTWARE

**ADDISON INDUSTRIAL ESTATE, BLAYDON-ON-TYNE, TYNE & WEAR. TEL: (091) 414 4611.**

**ALL TITLES AVAILABLE FROM TYNESOFT**  
POSTAGE AND PACKING FREE

**SUPERFILE 64**  
A REALLY USER FRIENDLY DATABASE FOR THE 64.

CM64 CASS **£17.95** DISK **£19.95**

**FIREMAN FRED**  
HELP FIREMAN FRED RESCUE CATS, DOGS & PEOPLE FROM 20 DIFFERENT LOCATIONS.

SPEC. **£5.95**

**AUF WIEDERSEHEN**  
HELP OZ BUILD A WALL, DRINK HIS BEER, THEN FIND HIS WAY HOME.

SPEC. **£5.95** CM64 **£6.95**

BBC/ELECTRON **£6.95**

## NEW! from SQUIRREL Two very original programs for BBC-B and ELECTRON



### POLAR PERILS

**£7.95**

A 100% m/code multiscreen arcade/adventure game set in very hostile Arctic waters. No gun – just your wits and your spear for survival against many sub-zero hazards including ravenous polar bears who will hop on to your tiny ice floe without warning! Great skill is required to get through to the final screen of this very addictive adventure and only then do you realise that your difficulties have only just begun as you try to collect ice blocks in your kayak for your igloo home!

### MONEYWISE

**£9.95**

An up to date yet simple to use personal/home money management system based on not one but TWO main accounts with extensive analysis, budget forecasting, annual summary and 3D BAR CHARTS. Other facilities offered are – SAVE complete program to DISC, PASSWORD protected datafiles (tape or disc), PRINTOUTS of all accounts and 3D Bar chart screen dump.

Other established classics from SQUIRREL:-

### TRAFALGAR

**£8.00**

'A good combination of arcade action and strategy' – C & VG

### SUPERGOLF

**£7.50**

'I do know a good game when I see one and SUPERGOLF is just that!' – Electron User

### BUNFUN

**£6.50**

'Fast and furious – a game calling for quick reactions – fun to play' – The Micro User

All programs available on 40T disc – add £2.00

## SQUIRREL SOFT

Dept E,  
4 BINDLOSS AVENUE, ECCLES, MANCHESTER M30 0DU  
24 Hour answering service – 061-789 4120

Cheques, P.O.s  Same day despatch



**SPECIAL  
OFFER!**

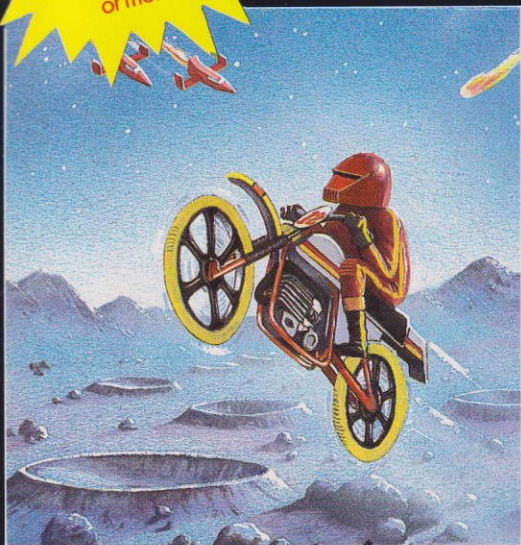
Deduct £1 per  
cassette when  
ordering 2  
or more

# THE BEST BBC MICRO SOFTWARE

\* PRODUCED BY AN INDEPENDENT SOFTWARE HOUSE \*

\* TOP QUALITY MACHINE-CODE PROGRAMS \*

**BBC  
MICRO**



## BMX ON THE MOON (32K)

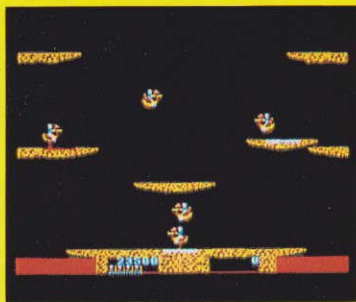
**£7.95**

Smooth scrolling-arcade action. In this game you control a moon cycle which you use to patrol the surface of the moon. You must avoid the rocks on the surface, and also the monsters bouncing high in the low gravity. You should try to shoot down the spaceships which fly overhead dropping bombs onto the moon surface: A further hazard are the flying barrels which are often just too high to shoot or jump over, and just too low to squeeze underneath.

●●●NEW RELEASE●●●



Many of our titles are available in Boots, W.H. Smiths, John Menzies, Rumbelows, Laskys, Currys, HMV, Greens at Debenhams and Spectrum Shops. Also at all major computer dealers – Eltec Computers, Micro Management, West Coast Personal Computers, Microstyle, Electronequip, 3D Computers, Computerama, GTM Computers, etc. Our software is also available through all the major distributors, and directly from us by mail-order.



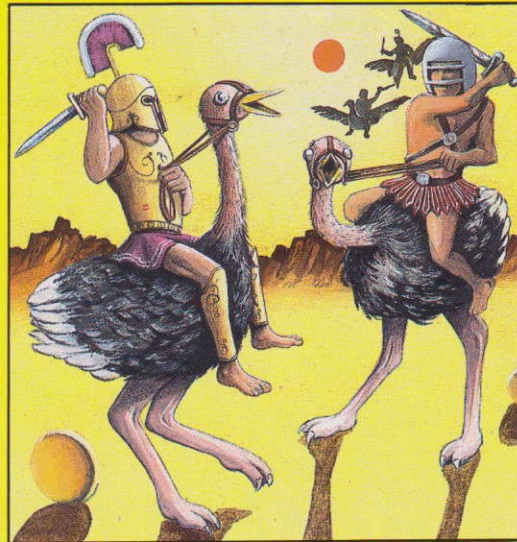
## WINGED WARLORDS (32K)

**£7.95**

A fascinating arcade-style game for one or two players. Each player rides an ostrich, and the aim is to knock the enemy riders off by colliding with them. In a collision the rider with the highest jousting pole will be victorious. If you are successful, the enemy will revert to its original form: an egg. The egg must be collided with to prevent it from hatching. Avoid the lava bath at the bottom of the screen, and remember: you can fly better by using stronger strokes – imagine you are an ostrich trying to fly!

(KEYBOARD OR JOYSTICKS)

●●●NEW RELEASE●●●



## SUPERIOR SOFTWARE LTD.

Dept. TM11, Regent House,  
Skinner Lane, Leeds 7  
Tel: 0532 459453

### OUR GUARANTEE

- (1) All our software is available before we advertise.
- (2) All our software is despatched within 48 hours by first-class post.
- (3) In the unlikely event that any of our software fails to load, return your cassette to us and we will immediately send a replacement.



# WIN! THIS FABULOUS GOLDEN TREASURE, THE FAMOUS GOLDEN JEWELLED HARE OF 'MASQUERADE' OR £30,000

WITH  
HARERAISER  
FROM  
HARESOF



*Hareraiser, the unique computer treasure hunt, a computer puzzle of graphics and text in two parts each part contains vital clues to places or locations which will enable you to discover the whereabouts of this golden treasure. It has not been buried you just have to pin point its location to claim this amazing prize. See cassette inlay card rules for the treasure hunt of 1984. Part one (Prelude) available now. Part two (Female) coming soon! Order yours now.*

Available for

ELECTRON	BBC B	ORIC/ATMOS 48	DRAGON 32
SPECTRUM 48	CBM 64	VIC 20 EX	AMSTRAD
TANDY EXT 32K COL	ATARI 800 XL	MSX	

HARESOF LTD.  
P.O. BOX 365, London NW1 7JD.  
Tel: 01-387 5876

Available at £8.95 each (£17.90) total. From Harrods, W.H. Smith, Boots, Tandy, Curry's Granada TV Rentals, Rumbelows, Lewis's Ltd, Debenhams, Lasky's and good Software outlets or direct from Haresoft Ltd. Access orders Phone 01-388 3910. 48 hr despatch.

Perry