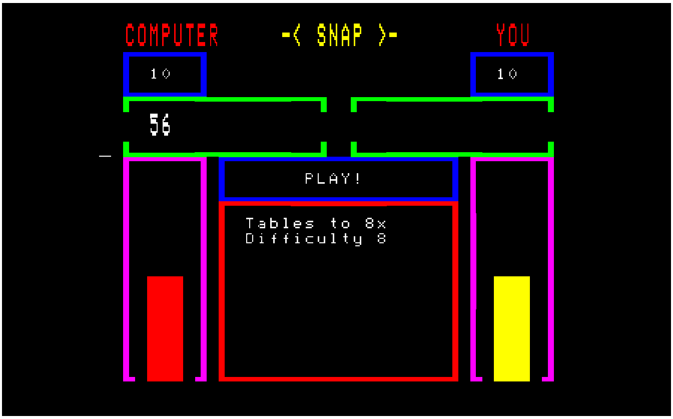
22

Snap

In this game of Snap, the computer displays a word or numeric value left and right alternately - 'playing', as it were, both hands. To win the point, the user must tap a key within the time limit set by the difficulty level. If a key is pressed when the two sides do not match, or if the user is too late pressing a key, a point is lost. The scoring is rather different from usual. Both the computer and the player start with ten points and points are exchanged, one side increasing as the other decreases. The game ends when either side reaches twenty, the other side then having nil. As an added interest, the scores on both sides are shown numerically, and underneath is a bar-graph or column representing the score graphically.

The user has a choice of subject matter, given in lines 120 to 140 of the listing. These are alphabetic or numeric. If for example the



user chooses three-letter words, then a match is simple; the word on each side must be identical for the Snap point to be won. On the other hand, if a numeric subject is chosen it is the total value on each side that must match. That is to say, if one side displays 12, then the other side might show 3 + 9, or 4 * 3, for a Snap. As an improvement, readers might like to devise simple graphics, but as the program is written in Mode 7 we cannot utilise user-defined graphics or drawn pictures of any kind. On the other hand, there might be scope for a few letters printed in a variety of colours on different backgrounds.

As will be seen from the illustration PROCBOX is used a lot in this game, which helps to concentrate attention. The screen is cleared in line 20 and then down to line 150 we are concerned with preparing the display. The DATA lines pertain to information regarding the position, size and colour of the various boxes. Although you may not understand how it works, the listing down to line 390 should be easy enough to follow.

PROCGRAPH is responsible for printing the left and right bargraphs, while PROCCLR clears the centre text box. When we reach the game loop, the user is warned with a flashing prompt (line 430) and then abruptly the word ' PLAY!' appears. The computer immediately starts playing characters into the left and right play areas.

PROCCHECK checks for a user input - Snap claim - and also checks to see if the two displays are the same. The scores and graphs are adjusted if necessary, as are the graphic displays.

Of all the listing, PROCCHOICE is the most interesting, as this is the procedure that prepares what is to appear in the two play areas. There is no problem if a choice of words was made, as line 1190 simply adjusts the READ pointer to the appropriate DATA line 1080 to 1120, but numeric values are entirely different and must be recalculated on every appearance. If you glance back at the user' s original choices, you will see that in the numeric selection there are numbers to 10 (total), numbers to 20, tables to 4 times, tables to 8 times and tables to 12 times. Line 1200 splits the tables away from the others and in each case there is a 50 per cent chance that either the program will print a complete equation as in ' 4 + 9', or print the value as in ' 13'. Follow this part of the listing carefully and you will see the point. Remember that in BBC BASIC, the EVAL command causes the evaluation of the numeric parts of a string.

Variables

GR$	A short block-graphic string for the graphs
X	General counter

B$ and C$	Strings of spaces for clearance purposes
A%	Horizontal position of box
B%	Vertical position of box
C%	Length of box
D%	Height of box
E%	Colour of box
L%	Number of left play display
R%	Ditto, right
L1%, R1%	Copies of above
PS%	Player's score
CS%	Computer's score
Q$	Player's choice input
CH%	ASCII value of Q$
D	Difficulty level
K	Dummy; also used for Snap input
Z$	Numeric string to be evaluated or printed
Z%	Value of Z$

```
 10 REM - Snap
 20 MODE7:PROCDBL(0,0,129,"COMPUTER")
 30 GR$=CHR$255+CHR$255+CHR$255+CHR$149
 40 PROCDBL(31,0,129,"YOU")
 50 PROCDBL(13,0,131,"_< SNAP >_")
 60 FORX=1TO8
 70 READA%,B%,C%,D%,E%:PROCbox(A%,B%,C%,D%,E%):NEXT
 80 B$=STRING$(8," "):C$=B$+B$:L%=0:R%=0:L1%=0:R1%=0
 90 FORX=1TO2:PRINTTAB(0,5+X);C$;C$;B$:NEXT
100 DATA1,3,3,1,4,30,3,3,1,4,1,7,13,2,2,20,7,13,2
110 DATA2,9,10,16,1,4,1,22,3,13,5,30,22,3,13,5,9,22,16,10,1
120 DATAA  2 letters,B  3 letters,C  4 letters
130 DATAD  5 letters,E  Numbers to 10,F  Numbers to 20
140 DATAG  Tables to 4x,H  Tables to 8x,I  Tables to 12x
150 PS%=10:CS%=10:PROCGRAPH
160 RESTORE120:PRINTTAB(11,10);
170 PRINTCHR$136;CHR$135;"Please choose";CHR$137
180 FORX=1TO9:READQ$:PRINTTAB(11,12+X);CHR$135;Q$:NEXT
190 PRINTTAB(11,22);CHR$130;"Your choice?";
200 REPEAT:Q$=GET$:UNTIL Q$>="A" AND Q$<="I"
210 CH%=ASC(Q$)-64
220 VDU7:PROCCLR:RESTORE120
230 FORX=1TO CH%:READX$:NEXT
240 PRINTTAB(11,15);CHR$135;"How difficult?"
250 PRINTTAB(11,17);CHR$135;"from 1 (easy)"
260 PRINTTAB(11,18);CHR$135;"to 9 (hard)..."
270 PRINTTAB(11,22);CHR$130;"Your choice?";
280 REPEAT:D=GET:UNTIL D>48 AND D<=58
290 PROCCLR:VDU7:PRINTTAB(11,10);C$
300 D=D-48:X$=RIGHT$(X$,LEN(X$)-3)
310 PRINTTAB(11,13);CHR$130;"You have chosen"
320 PRINTTAB(11,15);CHR$135;X$
```

```
330 PRINTTAB(11,16);CHR$135;"Difficulty ";D
340 PRINTTAB(11,18);CHR$130;"OK? (Y-N)";
350 REPEAT:G$=GET$:UNTIL G$="Y"OR G$="N"
360 IFG$="N"PROCCLR:GOTO160
370 VDU7:PROCCLR
380 PRINTTAB(11,13);CHR$135;X$
390 PRINTTAB(11,14);CHR$135;"Difficulty ";D:D=600-D*63
400
410 REM - Game loop
420
430 PRINTTAB(11,10);CHR$135;CHR$136;"READY !";CHR$137
440 K=INKEY(300):*FX15
450 PRINTTAB(11,10);C$;TAB(16,10);CHR$135;"PLAY!":VDU7
460 PROCCHOICE:IF CH%>4 L%=Z%:I$=Z$:GOTO490
470 L%=RND(15):IFL%=L1%GOTO470
480 L1%=L%:FORX%=1TOL%:READ I$:NEXT
490 PROCDBL(2,6,135,I$+B$):K=INKEY(D)
500 IF L%=R% OR K<>-1 PROCCHECK:GOTO430
510 PROCCHOICE:IF CH%>4 R%=Z%:J$=Z$:GOTO540
520 R%=RND(15):IF R%=R1% GOTO520
530 R1%=R%:FORX%=1TOR%:READ J$:NEXT
540 PROCDBL(21,6,135,J$+B$):K=INKEY(D)
550 IF L%=R% OR K<>-1 PROCCHECK:GOTO430
560 GOTO460
570
580 DEFPROCCHECK
590 IFL%<>R%OR K=-1 GOTO610
600 T$="You got it!":PS%=PS%+1:CS%=CS%-1:GOTO640
610 IFL%=R%ANDK=-1 T$="I got it!":GOTO630
620 T$="Not the same."
630 PS%=PS%-1:CS%=CS%+1
640 PRINTTAB(11,10);CHR$135;T$:VDU7
650 PROCGRAPH:K=INKEY(300):*FX15
660 PRINTTAB(11,10);C$
670 IFPS%<>0 AND CS%<>0 ENDPROC
680 CLS:PROCTITLE("GAME OVER")
690 IF CS%=0 PROCDBL(12,12,131,"YOU WIN!"):PROCWARBLE:END
700 PROCDBL(12,12,131,"I WIN!")
710 FORX=1TO3:PROCBOING:NEXT:END
720
730 DEFPROCbox(X%,Y%,L%,H%,C%)
740 LOCALV%,W%,I%,J%
750 PRINTTAB(X%,Y%);
760 V%=VPOS:W%=POS
770 PRINTTAB(W%,V%-H%);CHR$(C%+144);"7";
780 FORI%=0TOL%+1:PRINT"£";:NEXT
790 PRINT"k":PRINTTAB(W%,V%+1);CHR$(C%+144);"u";
800 FORI%=0TOL%+1:PRINT"p";:NEXT
810 PRINT"z":FORJ%=V%-H%+1TOV%
820 PRINTTAB(W%,J%);CHR$(C%+144);"5":NEXT
830 FORJ%=V%-H%+1TOV%
840 PRINTTAB(W%+L%+3,J%);CHR$(C%+144);"j":NEXT
850 ENDPROC
860
```

```
 870 DEFPROCDBL(X%,Y%,C%,X$)
 880 PRINTTAB(X%,Y%);CHR$141;CHR$C%;X$
 890 PRINTTAB(X%,Y%+1);CHR$141;CHR$C%;X$:ENDPROC
 900
 910 DEFPROCTITLE(X$)
 920 PRINTCHR$132;STRING$(19,"Oo")
 930 PROCDBL((36-LEN(X$))/2,4,131,X$)
 940 PRINTCHR$132;STRING$(19,"Oo")
 950 ENDPROC
 960
 970 DEFPROCGRAPH
 980 FORX%=(23-CS%*.66)TO23:PRINTTAB(3,X%);CHR$145;GR$:NEXT
 990 PRINTTAB(3,23-CS%*.66);"    "
1000 FORX%=(23-PS%*.66)TO23:PRINTTAB(32,X%);CHR$147;GR$:NEXT
1010 PRINTTAB(32,23-PS%*.66);"    "
1020 PRINTTAB(3,3);CHR$135;CS%;" "
1030 PRINTTAB(32,3);CHR$135;PS%;" ":ENDPROC
1040
1050 DEFPROCCLR
1060 FORX%=13TO22:PRINTTAB(12,X%);C$:NEXT:ENDPROC
1070
1080 DATAit,to,at,on,in,by,be,me,my,up,an,am,as,is,so
1090 DATAbun,ban,bin,din,bed,bud,bid,did,dud,nod,and,den
1100 DATAboy,say,day,band,bend,bond,bind,dens,send,sand
1110 DATAnods,fire,find,fine,rind,rent,rend,tire
1120 DATAbrush,shrub,shred,sheds,stabs,drays,brays,burst
1130 DATAstrew,straw,stars,slabs,stubs,blest,blast
1140
1150 DEFPROCCHOICE
1160 REM - If the program is renumbered, the RESTORE
1170 REM - value of next line must be altered manually.
1180
1190 IFCH%<=4 RESTORE(1070+10*CH%):GOTO1320
1200 IFCH%>=7 GOTO1260
1210 IFRND(1)>.5 GOTO1240
1220 Z$=STR$(RND((CH%-3)*2))+" + "+STR$(RND((CH%-3)*2))
1230 Z%=EVAL(Z$):GOTO1320
1240 Z$=STR$(RND((CH%-3)*2)+RND((CH%-3)*2))
1250 Z%=EVAL(Z$):ENDPROC
1260 IFRND(1)>.5 GOTO1300
1270 Z$=STR$(RND(5)-1+(CH%-7)*4)
1280 Z$=Z$+" * "+STR$(RND(5)-1+(CH%-7)*4)
1290 Z%=EVAL(Z$):GOTO1320
1300 Z$=STR$((RND(5)-1+(CH%-7)*4)*(RND(5)-1+(CH%-7)*4))
1310 Z%=EVAL(Z$):GOTO1320
1320 ENDPROC
1330
1340 DEFPROCBOING
1350 SOUND 0,-15,80,2
1360 FOR S%=-15 TO 0:SOUND1,S%,20+S%,2:NEXT:ENDPROC
1370
1380 DEFPROCWARBLE
1390 FORS%=1TO20:SOUND1,-12,30,1
1400 SOUND1,-12,100,1:NEXT:ENDPROC
```