18

Bingo

It is no bad thing to make concessions to other members of the family, especially when one has hogged the TV for hours on end! Here is a computer implementation of Bingo that four can play and which should convince some members of the family that the computer can be of interest to all.

All players start with £10 and a 'card' costs £1, the winner being the one first to have a full card. The computer asks for the number of people playing and the time allowance. This latter is the time delay in seconds before the next number is called. To start with, allow about 7 or 8 seconds and decrease it as players become expert. The screen clears and then each player is asked to type in his or her name. When this done, the screen again clears and a Bingo card is drawn for each player, each in his own colour, complete with number and name. Each card has twelve numbers and they are arranged as a normal Bingo card, lowest on the left, with values 11 to 99.

The game now starts. As the computer 'calls' each new number, players' attention is drawn by a short beep, and players must claim any that they have. That is to say, the computer will not automatically tag or cover the numbers on each card (that would be boring), but instead the player(s) must press a number 1, 2, 3 or 4, corresponding to their own player or card number. The computer will then cover the player's number - unless he or she is fibbing or mistaken - and restart the time delay. In this way, two or more players may claim the same called number. When the time allowance is up, with no players claiming, another number is called automatically. Note very carefully that if a player misses a claim, it is still possible that he or she can win, for in lines 220 to 240 the program makes 20 tries at finding an uncalled number before giving up and blanking the whole lot again. This provision is necessary because - at the faster auto speeds - it is conceivably possible that all players may miss at least one called number and therefore that no-one could win unless numbers were called again. There is no penalty for a wrong claim. The computer automatically checks for a winning card and adjusts cash accordingly before presenting a new set of Bingo cards.

The array C(X,Y) is the data concerning the players, with X the player's number and Y the 12 numbers on his or her card. After the usual input of names, PROCSHUFFLE blanks off the 'caller' sheet, as it were, and then PROCARRAY creates and prints the player's card. It does this, starting at line 930, by first choosing 12 random numbers, ensuring that no two are alike, and then arranging them in ascending order (lines 1010 to 1050). The heading is printed by lines 1070 and 1080, and the three lines of the actual card are printed. For each row of data there are nine places to be printed or not, depending upon the next number; if when divided by ten it is greater than the current horizontal position, it is printed. Every third number is taken, with the starting bias depending upon the row number. For each horizontal position, therefore, either the number is printed, with padding spaces (lines 1120 and 1130), or else four graphics blocks are printed (line 1140). This is repeated as many times as necessary, being called by line 180 of the main program.

When RETURN is pressed, we enter the main game loop. The computer calls a number by choosing at random, checking that the N(X) array is blank (the number not having been called before). If it has been called, it makes another try and after 20 unsuccessful tries blanks the sheet off again. If all is well, the number is called, the VDU7 command causing a short beep.

The program now waits for a period of time set by the players, and if no input is detected on line 310, goes to choose another number to call. On an input, PL% is a legal player number, and the next lines check that the called number is on the player's card and not yet blanked off. Lines 380 to 400 calculate the position of the number on the screen and blank it by printing a small magenta marker over it. Line 410 actually blanks the number in the player's array.

Any successfully called number may result in a winning card, so now we check. If any element is found unblanked, that card is not yet full, so the game can proceed. Otherwise the screen is cleared, the winner announced and the cash adjusted. Pressing RETURN clears the screen for another set of cards.

Variables

P$(X)	Players' names
P(X)	Players' cash balances
C(X,Y)	Values on cards, where X = player and Y has 12 elements
N(X)	Caller's 'sheet', where 0 = uncalled number
G%	Number of players
TA%	Time allowance in seconds before another number is called
X%	General counter
X$	Transfer string from PROCbox
TRY%	Computer's tries at finding an uncalled number
N%	Current called number
PL%	Current player claiming a number
Y%	General counter
I%	Horizontal tab position for blanking piece on screen
J%	Vertical tab position for blanking piece on screen
Q%	Counter when checking for winning card

In PROCARRAY:

O%	Current number to be placed on card
N%	Count of numbers to be placed on card or checked
P%	Counter when checking for duplicates
FLAG%	Value = 1 when swap has taken place in sorting
Z%	Simple holding variable for swap process
R%	Number of row to be printed
C%	Character position in row

```
 10 MODE7:PROCTITLE("BINGO")
 20 DIM P$(4),P(4),C(4,12),N(99)
 30 PRINT''"How many players (1-4)?";
 40 REPEAT:G%=GET-48:UNTIL G%>0 AND G%<5:PRINT;G%
 50 INPUT'"Time allowance (1-9 secs)",TA%:TA%=TA%*100
 60 PRINT'"You all start with £10"
 70 PRINT'"Cards cost £1 each.":PROCRET
 80 CLS:PRINT''"Please type your names:"
 90 FOR X%=1 TO G%:P(X%)=10
100 PRINTTAB(0,X%*4);"Player #";X%
110 PROCbox(12,X%*4,12,1,148,0)
120 P$(X%)=X$:NEXT:PROCRET
130
140 REM - Outer loop
150
160 PROCSHUFFLE
170 CLS:FOR X%=1 TO G%
```

```
180 PROCARRAY(X%):NEXT:PROCRET
190
200 REM - Game loop
210
220 TRY%=1
230 N%=RND(89)+10:IF N(N%)=0 GOTO260
240 TRY%=TRY%+1:IF TRY%<=20 GOTO230
250 PROCSHUFFLE:GOTO220
260 N(N%)=N%:VDU7
270 VDU28,0,24,39,21,12
280 PRINTTAB(0,20);"Number called - ";N%
290 *FX15,1
300 PL%=INKEY(TA%)-48
310 IF PL%=-49 GOTO220
320 IF PL%<=0 OR PL%>G% GOTO300
330 Y%=1
340 IF C(PL%,Y%)=N% GOTO380
350 Y%=Y%+1:IF Y%<=12 GOTO340
360 PRINT;"Player ";PL%;" does not have ";N%;" on card."
370 PROCRET:GOTO270
380 I%=C(PL%,Y%)DIV10*4-3
390 J%=(PL%-1)*5+Y%MOD3:IFY%MOD3=0 J%=J%+3
400 VDU26:PRINTTAB(I%,J%);CHR$149;CHR$255;CHR$255
410 C(PL%,Y%)=0
420
430 REM - Check for win
440
450 Q%=1
460 IF C(PL%,Q%)<>0 GOTO270
470 Q%=Q%+1:IF Q%<=12 GOTO460
480 VDU26:CLS:PROCDBL(0,3,131,P$(PL%)+" WINS")
490 FOR Y%=1 TO G%
500 IF PL%=Y% P(Y%)=P(Y%)+G%-1 ELSE P(Y%)=P(Y%)-1
510 NEXT Y%:PRINT"CASH -":FOR X%=1 TO G%
520 PRINT;P$(X%);" - £";P(X%):NEXT
530 PROCRET:GOTO160
540 DEFPROCTITLE(X$)
550 PRINTCHR$132;STRING$(19,"Oo")
560 PROCDBL((36-LEN(X$))/2,3,131,X$)
570 PRINT'CHR$132;STRING$(19,"Oo")
580 ENDPROC
590
600 DEFPROCDBL(X%,Y%,C%,X$)
610 PRINTTAB(X%,Y%);CHR$141;CHR$C%;X$
620 PRINTTAB(X%,Y%+1);CHR$141;CHR$C%;X$
630 ENDPROC
640
650 DEFPROCbox(X%,Y%,L%,H%,C%,F%)
660 LOCALV%,W%,I%,J%:REM - MODE7 ONLY
670 PRINTTAB(X%,Y%);
680 V%=VPOS:W%=POS:PRINTTAB(W%,V%-H%);CHR$C%;"7";
690 FORI%=0TOL%+1:PRINT"£";:NEXT:PRINT"k"
700 PRINTTAB(W%,V%+1);CHR$C%;"u";
710 FORI%=0TOL%+1:PRINT"p";:NEXT:PRINT"z"
```

```
 720 FORJ%=V%-H%+1TOV%:PRINTTAB(W%,J%);CHR$C%;"5":NEXT
 730 FORJ%=V%-H%+1TOV%
 740 PRINTTAB(W%+L%+3,J%);CHR$C%;"j":NEXT
 750 IF F%GOTO860
 760 PRINTTAB(W%+2,V%);CHR$135;
 770 FORI%=1TOL%:PRINT".";:NEXT
 780 PRINTTAB(W%+3,V%);:X$=""
 790 G$=GET$:IF ASCG$=13 GOTO860
 800 IF LENX$=L%OR ASCG$=127 GOTO820
 810 PRINT G$;:X$=X$+G$:GOTO790
 820 IF X$="" GOTO790
 830 X$=LEFT$(X$,LENX$-1):PRINTCHR$8;
 840 IF ASCG$<>127 GOTO810
 850 PRINT".";CHR$8;:GOTO790
 860 ENDPROC
 870
 880 DEFPROCRET
 890 PRINTTAB(5,19);CHR$131;"Press";
 900 PRINTCHR$132;CHR$157;CHR$129;"RETURN ";CHR$156;
 910 G$=GET$:ENDPROC
 920
 930 DEFPROCARRAY(X%)
 940 LOCAL N%,O%,P%,Z%,FLAG%
 950 FOR N%=1 TO 12
 960 O%=RND(89)+10:P%=1
 970 IF C(X%,P%)=O% GOTO960
 980 P%=P%+1:IF P%<N% GOTO970
 990 C(X%,N%)=O%:NEXT
1000 REM - Arrange in order
1010 FLAG%=0:FOR N%=1 TO 11
1020 IF C(X%,N%)<=C(X%,N%+1)GOTO1050
1030 Z%=C(X%,N%):C(X%,N%)=C(X%,N%+1)
1040 C(X%,N%+1)=Z%:FLAG%=1
1050 NEXT N%:IF FLAG%<>0 GOTO1010
1060 REM - Print card
1070 PRINTCHR$150;CHR$157;CHR$132;
1080 PRINT;"PLAYER ";X%;" - ";P$(X%)
1090 FOR R%=1 TO 3:N%=R%:PRINTCHR$(144+X%);
1100 FOR C%=1 TO 9:IF N%>12 GOTO1140
1110 IF C(X%,N%)DIV10>C% GOTO1140
1120 PRINTCHR$135;:IFC(X%,N%)<10 PRINT" ";
1130 PRINT;C(X%,N%);CHR$(144+X%);:N%=N%+3:GOTO1150
1140 PRINTCHR$255;CHR$255;CHR$255;CHR$255;
1150 NEXT:PRINT:NEXT:PRINT
1160 ENDPROC
1170 DEFPROCSHUFFLE
1180 FOR X%=11 TO 99:N(X%)=0:NEXT
1190 ENDPROC
```