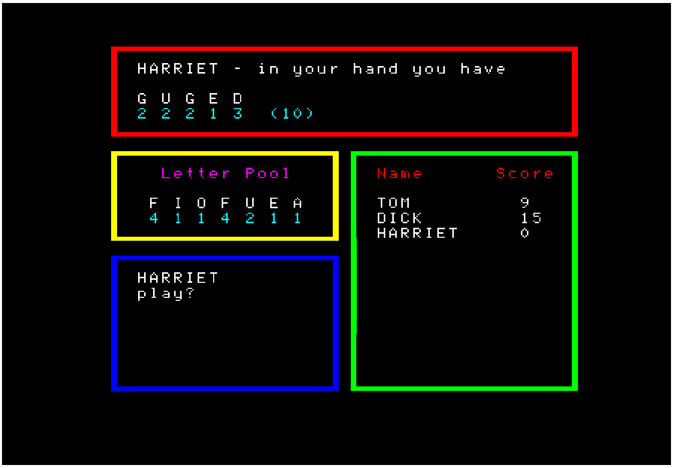
24

Scribble

Scribble is a game for any number of players up to six, and has much in common with Scrabble, Lexicon and similar word games. The object is to make words, for which the computer awards points, the first player reaching a score of 200 points being the winner.

On his or her turn, the player may do one of three things: (a) shuffle the letter pool, (b) take a letter from the pool and add it to his hand, or (c) make a word from the letters in his hand. To shuffle the pool, the player simply presses RETURN. To take up a letter from the pool, the letter is typed. The computer removes the letter from the pool and adds it to those in the player's hand, then selects another letter for the pool. To play a word, the player types the whole word



```
HARRIET - in your hand you have

G U G E D
2 2 2 1 3   (10)

  Letter Pool          Name        Score

 F I O F U E A         TOM           9
 4 1 1 4 2 1 1         DICK         15
                       HARRIET       0

HARRIET
play?
```

and presses RETURN. The computer checks that the word is possible from the letters in the hand, and if so, asks if any other player objects to the word or to its spelling. If all is well, the score is calculated and the letters are removed from the player's hand.

Please note very carefully that the score for a word depends upon (a) the individual values of the letters making up the word, (b) the length of the word, with longer words scoring heavily, and (c) the number of letters remaining in the hand; the more letters left unused, the greater the penalty. At the extreme, it is just possible to play a short word, have a lot of letters left over and so be awarded a negative score!

There are 'jokers' in the form of '*', which have no value but which may be used in place of any letter. To pick up such a joker, the '*' must be typed. On the other hand, to play a word containing a joker, the player does not include '*', but types the whole word as he or she wishes it to be. The computer first checks for normal letters and if one or more short, checks for jokers.

Before and after each player's move there is a timed delay for thought and observation, but players may move the game on swiftly by pressing RETURN. Finally, the number of times that each player may shuffle the letter pool is limited, dependent upon the number of players.

Lines 30 to 50 scatter various letters around the screen in a variety of colours and may well be omitted. They just make a pretty introduction to the game. The title screen appears on lines 60 to 80 and is moved on automatically after five seconds, or sooner if desired by pressing any key. The next lines seek information on the number of players, calculate the number of times each may shuffle the letter pool, and seek players' names. We then call PROCDISPLAY which in turn calls PROCBOX several times, creating an attractive playing screen. PROCINIT selects a starting pool of seven letters and also selects five letters for each player. We then enter the game loop.

It will be seen that the game loop is very simple. On each pass through it will call PROCPLAY, PROCPOOL, PROCPANEL and PROC2 before checking if the current player PL% has reached a winning score.

PROCPLAY calls PROCPANEL, which prints the player's name and lists the letters in his or her hand, with values printed below. The player is then prompted for his play, in line 1020. If G$ is empty, RETURN has been pressed on an empty set and so the player wants to shuffle the letter pool. If the length of the input string is 1, the player wishes to take a letter from the pool, and this is checked on line 1090. The presence of the letter in the pool is looked for in line 1120 and, all being well, line 1180 takes it into the player's hand. A new letter for the pool is found in line 1200.

The player's hand is checked, in lines 1220 onward, when he wishes to make a word. Each character of the input string is checked against the letters held, but notice that in order to save memory space we save the ASCII value of the letters, not the letters themselves. If we are short of a letter or two, lines 1300 to 1310 check for the presence of a joker. We have kept a running score of letter values in T% while the checking was going on, and in line 1440 we add a bonus for the length of the word and subtract three points for each letter left in the hand. Change this if you don't like it. The rest of the listing should be fairly straightforward.

You will see that the values of the 26 letters are stored as DATA statements in line 710, with the most commonly used letters having a value 1, rising up to 9 for Q and Z. The letters themselves are held in lines 720 to 800, and at first sight one might wonder why they are there at all - why not just pick a random number 1 to 26? In practice, however, such a course would have the J, X or Z occurring as often as E or A; clearly not a desirable state of affairs. Of course, one could write a RND statement and hedge it about with all sorts of IFs and other things to increase the frequency of chosen letters, but in the end one has a whole lot of BASIC statements that take up far more space than the DATA lines shown. I wrote the alphabet several times and then altered infrequently occurring letters to others over a large number of games until I achieved what I thought to be the right mix. Change it by all means!

Variables

C$	Row of spaces for clearance purposes
X%	General counter
C%	Random colour
Q$	General-purpose print string; also dummy
P%	Number of players
MAX%	Shuffles permitted per player

P%(X,Y)	Player's data, where X = player number and
	Y = 0 = score
	Y = 1 to 12 = ASCII value of letters held
	Y = 13 = shuffles usd

P$(X)	Players' names
POOL$	Letters in pool
C%(X)	Current player's letters
PL%	Current player number

In PROCINIT:

X%	General counter
L%	Randomly chosen letter, 1 to 178
G$	Chosen letter

In PROCPOOL:

X%	General counter
V%	Letter value
C%	Alphabetic position
A%	General counter

In PROCPLAY:

G$	Player' s input
X$	Message string
Q$	Dummy
X%	General counter
Y%	Number of first empty place in player' s hand
C%	Randomly chosen letter
T%	Running total of score
Z%	General counter
S%	Letter value

```
 10 REM - Scribble
 20 C$=STRING$(15," ")
 30 CLS:FORX%=1 TO 200:C%=RND(26)+64
 40 Q$=CHR$(RND(7)+128)+CHR$C%
 50 PRINTTAB(RND(37),RND(22));Q$:NEXT
 60 PROCDBL(12,13,135,"SCRIBBLE   ")
 70 PRINTTAB(13);"__________ "
 80 PROCDBL(2,9,135," THE  COMPUTER  WORD  GAME  ")
 90 Q$=INKEY$(500):MODE7
100 PRINT''''"How many players (1-6)";:PROCBOX(1,1,4)
110 REPEAT:P%=GET-48:UNTIL P%>0 AND P%<=6:PRINT;P%
120 MAX%=12/(P%+1)
130 IF P%=1 PRINT''"You";ELSE PRINT''"Each player";
140 PRINT;" may shuffle the letters"
150 IF MAX%=1 PRINT"once."''ELSE PRINT;MAX%;" times."''
160 INPUT" Press RETURN..."G$
170 DIM P%(P%,13),P$(P%),POOL$(7),C%(12)
180 CLS:FOR X%=1 TO P%
190 PRINT'"Player #";X%;" -"
200 PRINT"please type your name ";:PROCBOX(12,1,4)
210 INPUT P$(X%):NEXT:PROCDISPLAY:PROCINIT:PL%=1
220
230 REM - GAME LOOP ***************
240
250 PROCPLAY
260 PROCPOOL
270 PROCPANEL
```

```
280 PROCL2
290 IF P%(PL%,0)<200 PL%=(PL%MODP%)+1:GOTO250
300 PRINTTAB(21,19);CHR$135;CHR$136;P$(PL%);" WINS!";
310 END
320
330 DEFPROCDBL(X%,Y%,C%,X$)
340 PRINTTAB(X%,Y%);CHR$141;CHR$C%;X$
350 PRINTTAB(X%,Y%+1);CHR$141;CHR$C%;X$:ENDPROC
360
370 DEFPROCBOX(L%,H%,C%)
380 V%=VPOS:W%=POS
390 PRINTTAB(W%,V%-H%);CHR$(C%+144);"7";
400 FOR I%=0 TO L%+1:PRINT"£";:NEXT:PRINT"k"
410 PRINTTAB(W%,V%+1);CHR$(C%+144);"u";
420 FOR I%=0 TO L%+1:PRINT"p";:NEXT:PRINT"z"
430 FOR J%=V%-H%+1 TO V%
440 PRINTTAB(W%,J%);CHR$(C%+144);"5";CHR$135:NEXT
450 FOR J%=V%-H%+1 TO V%
460 PRINTTAB(W%+L%+3,J%);CHR$(C%+144);"j":NEXT
470 PRINTTAB(W%+3,V%);"";:ENDPROC
480
490 DEFPROCDISPLAY
500 CLS:PRINTTAB(0,4);:PROCBOX(35,4,1)
510 PRINTTAB(0,11);:PROCBOX(15,4,3)
520 PRINTTAB(4,8);CHR$133;"Letter Pool"
530 PRINTTAB(20,21);:PROCBOX(15,14,2)
540 PRINTTAB(22,8);CHR$129;"Name      Score"
550 PRINTTAB(0,21);:PROCBOX(15,7,4)
560 ENDPROC
570
580 DEFPROCINIT
590 PROCMIX:REM - Shuffle pool
600 FOR X%=1 TO P%:FOR C%=1 TO 5
610 RESTORE 720:L%=RND(178)
620 FOR Y%=1 TO L%:READ G$:NEXT
630 P%(X%,C%)=ASC(G$):NEXT:NEXT
640 PROCUPDATE:PROCPOOL:ENDPROC
650
660 DEFPROCMIX:REM - Shuffle pool
670 FOR X%=1 TO 7:RESTORE 720
680 L%=RND(178)
690 FOR Y%=1 TO L%:READ POOL$(X%)
700 NEXT:NEXT:ENDPROC
710 DATA 1,5,3,3,1,4,2,2,1,6,5,2,3,3,1,3,9,3,2,2,2,6,4,7,2,9
720 DATA A,B,C,D,E,F,E,E,I,J,K,L,E,A,O,P,*,R,S,T,U
730 DATA V,I,X,Y,U,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P
740 DATA O,R,S,T,U,V,A,X,Y,E,A,B,C,D,E,F,G,H,I,E,*
750 DATA L,M,N,O,P,R,S,T,U,E,W,X,Y,A,B,C,D,E,F,G,H
760 DATA I,J,K,L,M,N,O,P,R,S,T,U,E,W,Y,*
770 DATA A,B,C,D,E,F,G,H,I,E,L,M,N,O,P,R,S,T,E,W,Y,*
780 DATA A,C,D,E,F,G,H,I,L,E,N,O,P,R,S,T,U,W,Y,*
790 DATA A,C,D,E,G,H,I,L,M,N,O,P,R,S,T,U,Y,*,A,E,G
800 DATA H,I,L,M,O,S,T,U,Y,*,A,E,I,O,E
810
```

```
 820 DEFPROCUPDATE:REM - Print scores
 830 FOR X%=1 TO P%
 840 PRINTTAB(22,9+X%);CHR$135;P$(X%);TAB(35,9+X%);P%(X%,0)
 850 NEXT:ENDPROC
 860
 870 DEFPROCPOOL:REM - Print pool
 880 PRINTTAB(3,10);CHR$135;
 890 FOR X%=1 TO 7:PRINT;POOL$(X%);" ";:NEXT
 900 PRINTTAB(3,11);CHR$134;
 910 FOR X%=1 TO 7:IF POOL$(X%)="*" V%=0:GOTO950
 920 C%=ASC(POOL$(X%))-64
 930 RESTORE 710
 940 FOR A%=1 TO C%:READ V%:NEXT
 950 PRINT;V%;" ";
 960 NEXT:ENDPROC
 970
 980 DEFPROCPLAY
 990 *FX15,1
1000 PROCPANEL:VDU7
1010 PRINTTAB(2,15);CHR$135;P$(PL%);
1020 PRINTTAB(2,16);CHR$135;"play";:INPUT G$
1030 IF G$<>""GOTO1090
1040 IF P%(PL%,13)<MAX% GOTO1080
1050 X$="All shuffles used"
1060 PRINTTAB(2,17);CHR$129;X$
1070 PROCBOING:Q$=INKEY$(500):PROCCLIP:GOTO990
1080 PROCMIX:PROCPOOL:ENDPROC
1090 IF LEN(G$)>1 GOTO1220 ELSE X%=1
1100
1110 REM - Take letter from pool
1120 IF POOL$(X%)=G$ GOTO1150
1130 X%=X%+1:IF X%<=7 GOTO1120
1140 X$="Not in the pool":GOTO1060
1150 IF P%(PL%,12)<>0 X$="Hand full":GOTO1060
1160 Y%=1
1170 IF P%(PL%,Y%)<>0 Y%=Y%+1:GOTO1170
1180 P%(PL%,Y%)=ASC(G$)
1190 RESTORE 720:C%=RND(178)
1200 FOR D%=1 TO C%:READ POOL$(X%):NEXT:ENDPROC
1210
1220 REM - Make a word
1230 FOR X%=1 TO 12:C%(X%)=P%(PL%,X%):NEXT:FL%=1
1240 T%=0:FOR X%=1 TO LEN(G$)
1250 C%=ASC(MID$(G$,X%,1)):Y%=1
1260 IF C%=C%(Y%) GOTO1320 ELSE Y%=Y%+1
1270 IF Y%<=12 GOTO1260 ELSE Y%=1
1280
1290 REM - Letter not found. Check *'s
1300 IF C%(Y%)=42 GOTO1320 ELSE Y%=Y%+1
1310 IF Y%<=12 GOTO1300 ELSE X$="Impossible":GOTO1060
1320 C%(Y%)=0:IF C%(Y%)=42 GOTO1340
1330 RESTORE 710:FOR Z%=1 TO C%-64:READ S%:NEXT
1340 T%=T%+S%:NEXT X%
1350 PRINTTAB(2,17);CHR$130;"Objections? ";
```

```
1360 REPEAT:Q$=GET$:UNTIL Q$="Y" OR Q$="N":PRINTQ$
1370 IF Q$<>"Y"GOTO1430
1380 P%(PL%,0)=P%(PL%,0)-10:PROCCLIP
1390 PRINTTAB(2,16);CHR$129;"Penalty 10"
1400 Q$=INKEY$(500):PROCCLIP:GOTO1010
1410
1420 REM - Calculate score
1430 C%=0:FOR X%=1 TO 12:IF C%(X%)<>0 C%=C%+1
1440 NEXT X%:T%=T%+(LEN(G$)-1)^2-C%
1450 P%(PL%,0)=P%(PL%,0)+T%
1460 PRINTTAB(2,18);CHR$133;"Score ";T%
1470 PROCUPDATE:PROCWARBLE:PROCL2:Y%=1
1480 FOR X%=1 TO 12:P%(PL%,X%)=0:NEXT
1490 FOR X%=1 TO 12
1500 IF C%(X%)=0 GOTO1520
1510 P%(PL%,Y%)=C%(X%):Y%=Y%+1
1520 NEXT X%
1530 ENDPROC
1540
1550 DEFPROCPANEL:REM - Player's panel
1560 PROCL3
1570 PRINTTAB(2,1);CHR$135;P$(PL%);
1580 PRINT" - in your hand you have"
1585 PRINTTAB(2,3);CHR$135;:X%=1
1590 IF P%(PL%,X%)=0 GOTO1620
1600 PRINT;CHR$(P%(PL%,X%));" ";
1610 X%=X%+1:IF X%<=12 GOTO1600
1620 PRINTTAB(2,4);CHR$134;:X%=1:T%=0
1630 IF P%(PL%,X%)=0 GOTO1680
1640 IF P%(PL%,X%)=42 V%=0:GOTO1670
1650 RESTORE 710
1660 FOR A%=1 TO P%(PL%,X%)-64:READ V%:NEXT
1670 PRINT;V%;" ";:T%=T%+V%:X%=X%+1:IF X%<=12 GOTO1630
1680 PRINT;" (";T%;")":ENDPROC
1690
1700 DEFPROCCLIP:REM Clear input
1710 PRINTTAB(2,16);C$:PRINTTAB(3,17);C$:ENDPROC
1720 DEFPROCL2:REM Clear input panel
1730 G$=INKEY$(500)
1740 FOR X%=15 TO 18:PRINTTAB(3,X%);C$:NEXT:ENDPROC
1750 DEFPROCL3:REM Clear player's panel
1760 FORX%=1 TO 4:PRINTTAB(2,X%);STRING$(36," "):NEXT
1770 ENDPROC
1780
1790 DEFPROCWARBLE
1800 FORS%=1TO20:SOUND1,-12,30,1
1810 SOUND1,-12,100,1:NEXT:ENDPROC
1820
1830 DEFPROCBOING
1840 SOUND 0,-15,80,2
1850 FOR S%=-15 TO 0:SOUND1,S%,20+S%,2:NEXT:ENDPROC
```