

4. Word Find Tester

General Description

This program runs on a BBC Model 'B' but without REMs and a restricted amount of data it should run on a model 'A'. However, this has not been tested.

The program tests the ability of children to see words within words, randomly accessing a data file of 'test' words. It subsequently displays a series of randomly chosen smaller words inviting a 'yes/no' response from the user as to whether the smaller word can be made from the test word. It checks each word before accepting the user's response and provides laudatory sounds and flashing congratulations if the user is successful. The user is invited each time to progress to the next word or may exit to the results display. This provides a display of the words, along with three stars for the correct answers. Users are meant to add up how many correct answers they scored and the computer will indicate whether it agrees with their assessment. The 'game' may be replayed.

It is important that the educational notes are read with this program as it can, with suitably chosen data, be made to do many things. The data that is given is sample data, and is a little too extensive for proper educational value.

Detailed Description

Lines 1-34 This is the main structure of the program calling several dependent procedures. Line 19 seeds the random number generator. In line 17 the array is limited deliberately to 40 elements, as this allows all results to be seen on one screen page. 'Testwords' in line 24 is the displayed key word.

35-67 This displays the introductory messages for the program. Change the envelope if you find it offensive, but I have found that sound is important as visual stimulation in keeping interest levels high. Line 51 clears the title strip from the centre of the screen, leaving the title in normal size type. Notice that the delays passed into the delay procedure in lines 49, 54, 57 and 61 are two and a half seconds. A delay of ' 10 gives five seconds. Also note that the program assumes that you are operating in normal ' CAPSLOCK' environment. No validation on lower case is done in line 66.

69-80 This chooses the test word that is going to be displayed on the screen. The data is kept at line 280. The value in the brackets, line 76, determines how much of the data is read. It would be possible to keep an extensive data file and through judicious use of the ' RESTORE' command and the value for the random number generator in line 76, access restricted subsets of the data.

81-93 This hardly needed to be placed in a separate procedure but does allow the respond routine to be altered without affecting this routine.

94-117 This displays the remainder of the instructions and then invites a reply, (yes or no) as to whether the smaller word can be made from the larger. A flag ' REPLY' is set which is used in the check procedure in line 115.

118-135 This procedure randomly accesses the data for a smaller word that can be made from the larger. Note the data starts at line 293 as shown by the RESTORE 293 in line 125. The first item of data read is a number which is used as the parameter in the random number generator. This item of data at line 298 determines the range of data read and if judiciously used with the RESTORE in line 125 can be made very selective.

136-159 This routine contains the only real processing in the program. Each letter in the smaller word is checked off against the ' testword'. If an unused letter in the testword is found it is exchanged for another character, a ' 9' and the routine continues the search. Lines 154-158 sets ' checflag' depending on the reply from the user and whether there exists a usable letter in the testword. The checkflag is used to signal a suitable success or failure sound in the picture routine, as well as deciding what message is displayed and putting the relevant stars into the array WORD\$ - all in the next routine.

160-178 This uses checkflag, called ' cflag in the procedure, to provide the sounds and messages outlined above. Notice that lines 175 and 176 generate double size characters for ' excellent' .

179-194 This routine is used to decide whether to progress to the next testword. I could have arranged automatic progression, but feel the user must be encouraged to interact with the machine at all times. Moreover an assumption is made if progression is automatic, that the user will not be bored in the middle of the ' game' . Concentration spans vary enormously.

195-227 The display of the results for the teacher: again the user is asked to interact with the computer. Not only are there stars against correct results but the user is required to add up the number of words he thinks were right. The computer adds a discrete ' also' line 222, if it agrees. No range check is done on the user' s response.

228-256 The menu hardly merits the title ' menu' It is an attempt to make the computer seem ' friendly and to discourage the user from repeatedly coming back to the same program by indicating that it, the computer, gets ' tired' .

257-268 Again self-evident . . . a ' BYEBYE' band is drawn in double size characters across the screen.

279-292 Main testword data.

293-324 Smaller testword data. Please note that the RESTORE command restores to the REM statement which heads the routine and to use the program without REMs this must be altered accordingly. For the MODEL ' Bit may be typed in as it stands.

Educational Notes

As it stands, with the data given the program is entertaining and varied but lacks a structured progression for learning. This can be provided through sensible use of the data.

Firstly the larger testwords should be kept in a family of words: eg FAMILIAR, FAMILIES etc. The smaller words should be a subset of the same letters with some unacceptable words added: eg FAME, FAMINE, FAIR, LIES, FILE, FLIES, FLAME, FLEES, FILL etc. From using the program it seems that children respond better to being able to say ' ye it does not fit' rather than ' n ot does not' Experiment for yourself, but as

with the anagram program, the power of the program depends very much on the data used. Build up a series of similar programs using different data. They can be suitably graded. Remedial 11 year olds were not bored for 30 minutes work, but towards the end tended to look for letter agreement between the two words rather than word shape agreement.

Program Listing

```

1 REM *****
2 REM *
3 REM *
4 REM *      WORD FIND TESTER      *
5 REM *      developed by Ian Murray *
6 REM *      JANUARY 1983          *
7 REM *
8 REM *
9 REM *****
10 REM
11 REM =====
12 REM

13 REM      main structure
14 REM
15 REM .....
16 REM
17 DIM WORD$(40):MODE7
18 REPEAT:CLS
19   FLAG1 = 0 : FLAG2 = 0 :X=RND(-TIME)
20   CHECKFLAG = 0:REPLY=0:SLOT=0:SCORE=0
21   PROCintro
22   REPEAT :CLS
23     PROCword
24     PROCseeword(TESTWORD$)
25     PROCrespond
26     PROCdelay(10)
27     FLAG3 = FALSE
28     PROCgiveup
29   UNTIL FLAG1 = TRUE OR SLOT > 40
30   PROCseeresult
31   PROCmenu
32 UNTIL FLAG2 = TRUE
33 PROCend
34 END
35 REM
36 REM =====
37 REM
38 REM      introduction to prog
39 REM .....
40 REM
41 DEF PROCintro
42 LOCAL REPLY$,X
43 REPLY$=""
44 ENVELOPE 1,130,4,-4,4,5,-10,-3,20,0,0,-5,16,126
45 VDU 31,0,10
46 PRINT CHR$(132)CHR$(157) CHR$(141) CHR$(133) CHR$(136)
TAB(15)"WORD SPOT"
47 PRINT CHR$(132) CHR$(157) CHR$(141) CHR$(133) CHR$(136)
) TAB(15)"WORD SPOT"
48 SOUND 1,1,100,200
49 PROCdelay(9)

```

```

50 SOUND &11,0,0,1
51 VDU 31,0,8:PRINT SPC(130) " "
52 VDU 31,0,2
53 PRINT TAB(5) CHR$(131) "You will see a word"
54 PROCdelay(5)
55 PRINT TAB(5) CHR$(131) "and then you will see"
56 PRINT TAB(5) CHR$(131) "other words which" CHR$(135) "
can"
57 PROCdelay(5)
58 PRINT TAB(5) CHR$(131) "or even";CHR$(136) CHR$(134) "
CANNOT" CHR$(137) CHR$(131) "fit in"
59 PRINT TAB(5) CHR$(131) "the word that you see."
60 VDU 31,0,15
61 PROCdelay(5)
62 PRINT CHR$(157) CHR$(147) CHR$(129) "WHEN YOU UNDERSTA
ND TYPE "CHR$(136) CHR$(132) "Y" CHR$(137) CHR$(129) "'
63 REPEAT
64   *FX15,1
65   LET REPLY$ = GET$
66   UNTIL REPLY$ = "Y"
67 ENDPROC
68 REM
69 REM =====
70 REM
71 REM   word select procedure
72 REM .....
73 DEF PROCword
74 RESTORE 280
75 LOCAL X,Y
76 X = RND(20)
77 REPEAT :READ TESTWORD$
78   Y=Y+1
79 UNTIL Y>=X
80 ENDPROC
81 REM
82 REM =====
83 REM
84 REM display testword and
85 REM message for the luckless
86 REM .....
87 REM
88 DEF PROCseeword(TESTWORD$)
89 CLS
90 PRINT CHR$(141) CHR$(131) TAB(15) TESTWORD$
91 PRINT CHR$(141) CHR$(131) TAB(15) TESTWORD$
92 SOUND 1,-6,40,20
93 ENDPROC
94 REM
95 REM =====
96 REM
97 REM get child's responses
98 REM to randomly selected words
99 REM which could fit in the
100 REM displayed word.
101 REM .....
102 REM
103 DEF PROCrespond
104 LOCAL X,Y,Z,REPLY$
105 PROCgetword
106 REPLY$ = ""
107 VDU 31,0,5
108 PRINTCHR$(134) "Can the word" CHR$(136) CHR$(135) GOTW
ORD$
109 PRINT CHR$(134) "be made from "TESTWORD$" above?";
110 REPEAT
111   *FX15,1
112   REPLY$=GET$
113 UNTIL REPLY$ = "Y" OR REPLY$ = "N"

```

```

114 IF REPLY$="Y" THEN PRINT CHR$(131) " YES":REPLY = TRU
E:SOUND 1,-10,120,5 ELSE PRINT CHR$(131) " NO":SOUND 1,-10,2
0,5: REPLY = FALSE
115 PROCcheck
116 PROCpicture(CHECKFLAG)
117 ENDPROC
118 REM
119 REM =====
120 REM
121 REM     get small inword
122 REM .....
123 REM
124 DEF PROCgetword
125 RESTORE 293
126 LOCAL X,Y
127 Y=0
128 READ X
129 Y = RND(X)
130 X=0
131 REPEAT
132     X=X+1
133     READ GOTWORD$
134 UNTIL X>=Y
135 ENDPROC
136 REM
137 REM =====
138 REM
139 REM     check the chosen word
140 REM .....
141 REM
142 DEF PROCcheck
143 LOCAL X,Y,L,LL
144 CHECKFLAG = TRUE
145 L = LEN(GOTWORD$)
146 LL=LEN(TESTWORD$)
147 TWORD$=TESTWORD$
148 X=0
149 REPEAT
150     X=X+1
151     L$=MID$(GOTWORD$,X,1)
152     Y=INSTR(TWORD$,L$)
153     TWORD$ = LEFT$(TWORD$,Y-1)+"9"+RIGHT$(TWORD$,LL-Y)
154     IF Y=0 AND REPLY = TRUE THEN CHECKFLAG = FALSE :X=L
155     IF Y=0 AND REPLY = FALSE THEN CHECKFLAG = TRUE :X=L
156 UNTIL X>=L
157 IF Y AND NOT REPLY THEN CHECKFLAG = FALSE
158 IF Y AND REPLY THEN CHECKFLAG = TRUE
159 ENDPROC
160 REM
161 REM =====
162 REM
163 REM     display computer analysis
164 REM     of child's response and
165 REM     for future display for
166 REM     the teacher
167 REM .....
168 REM
169 DEF PROCpicture(CFLAG)
170 ENVELOPE 2,140,-5,5,-5,2,33,2,2,2,-1,0,10,200
171 IF CFLAG THEN SOUND 2,1,20,13:SCORE = SCORE + 1 ELSE S
OUND 2,-12,20,5
172 SLOT=SLOT+1
173 IF CFLAG THEN PRINT CHR$(141) CHR$(136) TAB(15) "EXCEL
LENT" ELSE PRINT TAB(15) "< WRONG >"
176 IF CFLAG THEN PRINT CHR$(141) CHR$(136) TAB(15) "EXCEL
LENT"
177 SOUND &11,0,0,1
178 ENDPROC
179 REM =====

```

```

180 REM
181 REM    give up procedure
182 REM .....
183 REM
184 DEF PROCgiveup
185 PRINT:LOCAL REPLY$
186 PRINT CHR$(130) "Have you had enough tries ?";
187 REPEAT
188     *FX 15,0
189     REPLY$=GET$
190 UNTIL REPLY$ = "Y" OR REPLY$ = "N"
191 SOUND 1,-10,100,4
192 IF REPLY$ = "Y" THEN PRINT " YES":FLAG1 = TRUE ELSE PR
INT " NO" : FLAG1 = FALSE
193 PROCdelay(1)
194 ENDPROC
195 REM
196 REM =====
197 REM
198 REM    display results for teacher
199 REM .....
200 REM
201 DEF PROCseeresult
202 LOCAL X,ADD$,REPLY$,Y,Z
203 Y=0 :ADD$=""
204 CLS : VDU 14
205 PRINT CHR$(141) CHR$(131) CHR$(129) TAB(17) "RESULTS"
206 PRINT CHR$(141) CHR$(131) CHR$(129) TAB(17) "RESULTS"
207 PRINT
208 PRINT "Three stars mean you got the answer"
210 PRINT "correct in the test."
211 FOR X= 1 TO SLOT STEP 3
212     FOR Z = 0 TO 2
213         PRINT TAB(15*Z) WORD$(X+Z);
214         IF X+Z = SLOT THEN Z=2
215     NEXT
216 NEXT
217 PRINT
218 PRINT CHR$(134) "Now add up the ones you got right"
219 INPUT ">>> "KSCORE
220 PRINT
221 PRINT; "I think you got "SCORE;
222 IF KSCORE = SCORE THEN PRINT" also!"
223 PRINT:PRINT
224 PRINT CHR$(141) CHR$(134) TAB(16) "WELL DONE"
225 PRINT CHR$(141) CHR$(134) TAB(16) "WELL DONE"
226 PROCdelay(5)
227 ENDPROC
228 REM
229 REM =====
230 REM
231 REM    menu
232 REM .....
233 REM
234 DEF PROCmenu
235 CLS
236 PRINT CHR$(141) TAB(18) "MENU"
237 PRINT CHR$(141) TAB(18) "MENU"
238 PRINT CHR$(131) TAB(16) "-----"
239 FOR X = 0 TO 19
240 PRINT CHR$(149) "/" ;NEXT
241 PRINT:PRINT
242 PRINT "You have now finished the test but"
243 PRINT "of course you may have another go"
244 PRINT "with another word."
245 PROCdelay(5)
246 PRINT
247 PRINT CHR$(131) "I get tired after too long at this"
248 PRINT CHR$(131) "game so what do you think....."

```

```

249 PRINT:PRINT CHR$(130) "Shall we go on"
250 REPEAT
251   REPLY$ = GET$
252   UNTIL REPLY$ = "Y" OR REPLY$ = "N"
253   IF REPLY$ = "N" THEN FLAG2 = TRUE : PRINT TAB(18) "NO"
ELSE FLAG2 = FALSE :PRINT TAB(18) "YES"
254   SOUND 2,-10,60,4
255   PROCdelay(2)
256 ENDPROC
257 REM =====
258 REM
259 REM      goodbye
260 REM .....
261 REM
262 DEF PROCend
263 CLS
264 VDU 31,0,10
265 PRINT CHR$(150) CHR$(157) CHR$(141) CHR$(131) CHR$(136
) TAB(16) "BYE BYE"
266 PRINT CHR$(150) CHR$(157) CHR$(141) CHR$(131) CHR$(136
) TAB(16) "BYE BYE"

267 PROCdelay(5)
268 ENDPROC
269 REM
270 REM =====
271 REM
272 REM      delay procedure
273 REM .....
274 DEF PROCdelay(DEL)
275   FINISH = TIME + (50*DEL)
276   REPEAT
277   UNTIL TIME >= FINISH
278 ENDPROC
279 REM
280 REM =====

281 REM
282 REM this data may be changed !
283 REM .....
284 REM
285 DATA THEATRES, THERMAL,HEATERS
286 DATA THREESOME, TIRESOME,THREADBARE
287 DATA THROUGHOUT, THOROUGH,THOUGHT
288 DATA TOUGHEN,TERRIBLE,TERRIFIED
289 DATA FRIGHTENED,THREATENED,TREATED
290 DATA FRICTION,FRACTIONS,TRACTION
291 DATA STEALTHY,THIRSTY
292 REM
293 REM =====
294 REM
295 REM and so may this data
296 REM .....
297 REM
298 DATA 104
299 DATA HEAT,TREAT,TREE,EAT
300 DATA ARM,HARM,ERA,REAL
301 DATA MEAL,TERM,MALE,HELM,
302 DATA HER,MARE,HOME,SOME
303 DATA REST,HERE,SHEER,MERE
304 DATA TIE,TRIES,THREAD,READ
305 DATA BARE,TREAD,BREAD,BED
306 DATA BAD,BREAD,BEAD,BED
307 DATA RED,REAR,ROUGH,OUT
308 DATA GOUT,TROUT,ROT,ROUT
309 DATA TOUT,TOR,TROT,TROUGH
310 DATA TUG,UGHT,HEN,THEN
311 DATA TEN,HOG,HUG,HOT
312 DATA RITE,ERR,RIB,BERR

```


313 DATA BIT,FRIED,FED,RIFT
314 DATA FRIGHT,RIGHT,RIG,RIGHT
315 DATA TENT,NET,NEED,TENDER
316 DATA DEN,DEATH,TONIC,TIN
317 DATA TIC,FRICTION,TON,RATION
318 DATA RAT,FAT,CAT,FAST
319 DATA CAST,AFT,RAFT,FONT
320 DATA FIT,ACT,FACT,STEAL
321 DATA SHY,SLY,STY,STEAL
322 DATA HEAL,TEST,LAST,TASTY
323 DATA HASTY,SHIRT,SIR,HIT
324 DATA SIT,NOT,HAD,TAR

