program to loop back to the beginning, thus calling the routines that describe a location and its exits. As the value of the location variable, P, is not changed by the LOOK command, the same location will be described. This command is useful if, after the player has performed a series of actions, the original description of the current location has moved off the screen.

ADDING FLEXIBILITY

When issuing movement commands, the player may type in different forms of the same instruction. For example, GO NORTH, MOVE NORTH and GO TOWARDS THE NORTH are asking the same thing. Although it is not vital for an adventure game program to recognise all of these forms, it makes playing the game more interesting if a number of different instruction formats are legal. The three movement commands we just gave have a common structure: they all start with a movement verb, and the direction required is a discrete word. It is possible, therefore, to design a routine that will search the part of the sentence coming after the verb for the direction. The routine scans this part of the sentence for spaces, isolating each word in turn and comparing it with the four direction words sought until a match is found.

```
3630 REM **** SEARCH FOR DIRECTION S/R ****
3640 NN$=NN$+" ":LN=LEN(NN$):C=1
3645 FOR I=1 TO LN
3650 IF MID$(NN$,I,1)<>" " THEN NEXT I:RETURN
3655 W$=MID$(NN$,C,I-C):C=I+1
3660 IF W$="NORTH" OR W$="EAST" THEN NN$=W$:I=LN
3665 IF W$="SOUTH" OR W$="WEST" THEN NN$=W$:I=LN
3670 NEXT I
3675 RETURN
```

In the last instalment of the project we developed a movement routine. To add this new routine to the movement routine we need simply to add the following line:

```
3505 GOSUB3630:REM SEARCH FOR DIRECTION
```

It is worth noting that this routine will not obey instructions such as GO IN A NORTHERLY DIRECTION, since the direction word cannot be isolated by the routine. It would be possible to design a routine that worked on the principle of scanning groups of four and five letters, comparing each group with the four possible direction words. However, such a routine would have a long execution time. On the other hand, our program will accept GO NORTHWARDS, as the movement routine finally uses the first letter of the second part of the sentence, NN$. In this case, the N in NORTHWARDS would be accepted as N for NORTH.

Digitaya Listings

```
1220 GOSUB1700:REM ANALYSE INSTRUCTIONS
1225 IF F=0 THEN 1210:REM INVALID INSTRUCTION
1230 GOSUB 1900:REM NORMAL INSTRUCTIONS
1240 IF VF=0 THENPRINT"I DON'T UNDERSTAND"
1250 IF MF=1 THEN 1160:REM NEW POSITION
1260 IF MF=0 THEN 1210:REM NEW INSTRUCTION

1700 REM **** ANALYSE INSTRUCTION S/R ****
1705 F=0:REM ZERO FLAG
1710 IFIS$="END" OR IS$="LIST" THEN VB$=IS$:F=1:
     RETURN
1720 IF IS$="LOOK" THEN VB$=IS$:F=1:RETURN
1730 :
1740 REM ** SPLIT INSTRUCTION **
1750 VB$="":NN$="":REM ZERO VERB AND NOUN
1770 LS=LEN(IS$)
1780 FOR C=1TO LS
1790 A$=MID$(IS$,C,1)
1800 IF A$=" "THEN VB$=LEFT$(IS$,C-1):NN$=RIGHT$(I
     S$,LS-C):F=1:C=LS
1810 NEXT
1830 IF F=0 THEN PRINT:PRINT"I NEED AT LEAST TWO
     WORDS"
1840 RETURN
1850 :
1900 REM **** NORMAL ACTIONS S/R ****
1910 VF=0
1920 PRINT
1930 IF VB$="GO"ORVB$="MOVE"THENVF=1:GOSUB2000
1940 IF VB$="TAKE"ORVB$="PICK"THEN VF=1:GOSUB2140
1950 IF VB$="DROP"ORVB$="PUT"THENVF=1:GOSUB2360
1960 IF VB$="LIST"ORVB$="INVENTORY"THENVF=1:
     GOSUB2540
1965 IF VB$="LOOK" THEN VF=1:MF=1:RETURN
1970 IF VB$="END"ORVB$="FINISH"THENVF=1:GOSUB2610
1980 RETURN

2015 GOSUB8600:REM SEARCH FOR DIRECTION

2610 REM **** END GAME S/R ****
2620 PRINT:PRINT"ARE YOU SURE (Y/N) ?"
2630 GET A$:IFA$<>"Y"AND A$<>"N"THEN2630
2640 IFA$="N"THEN RETURN
2650 END

8600 REM **** SEARCH FOR DIRECTION S/R ****
8610 NN$=NN$+" ":LN=LEN(NN$):C=1
8620 FORI=1 TO LN
8630 IF MID$(NN$,I,1)<>" " THEN NEXT I:RETURN
8640 W$=MID$(NN$,C,I-C):C=I+1
8650 IF W$="NORTH" OR W$="EAST" THEN NN$=W$:I=LN
8660 IF W$="SOUTH" OR W$="WEST" THEN NN$=W$:I=LN
8670 NEXT I
8680 RETURN
```

Basic Flavours

Spectrum:

In both programs, use I$ for IS$, B$ for VB$, and R$ for NN$ throughout.

Replace the following lines in Digitaya:

```
1790 LET A$=I$(C TO C)
1800 IF A$=" " THEN LET B$=I$(TO C-1):LET R$=I$
     (LEN(I$)-LS+C+1 TO):LET F=1:LET C=LS
2630 LET A$=INKEY$:IF A$<>"Y"
     AND A$<>"N" THEN 2630
8630 IF R$(I TO I)<>" " THEN NEXT I:RETURN
8640 LET W$=R$(C TO I-1):LET C=I+1
```

Replace these lines in Haunted Forest:

```
2550 LET A$=I$(C TO C)
2570 LET B$=I$(TO C-1):LET F=1
2580 LET R$=I$(LEN(I$)-LS+C+1 TO):LET C=LS
3650 IF R$(I TO I)<>" " THEN NEXT I:RETURN
3655 LET W$=R$(C TO I-1):LET C=I+1
4190 LET A$=INKEY$:IF A$<>"Y" AND
     A$<>"N" THEN GOTO 4190
```

BBC Micro:

Replace this line in Digitaya:

```
2630 REPEAT A$=GET$:UNTIL A$="Y" OR
     A$="N"
```

and this line in Haunted Forest:

```
4190 REPEAT A$=GET$:UNTIL A$="Y" OR
     A$="N"
```