

# TREAD LIGHTLY

In this short series of articles we shall be looking at the construction of a graphics game using BBC BASIC. The game is designed to run on the Model A, Model B and the Electron. As each phase of the game is developed, the appropriate section of the program will be listed, allowing you to build up the game with each instalment.

**Routine Procedure**

Unlike a flow chart, this structure diagram shows the procedural structure of the program rather than the flow of control through it. A capsule indicates the start of a REPEAT...UNTIL loop; the lozenges are decision boxes — when the test fails, the enclosing loop continues. The Level numbers show the program's block structure: all loop starts and procedure calls open a new block of program and a lower logical level — compare this with the diagram on page 387

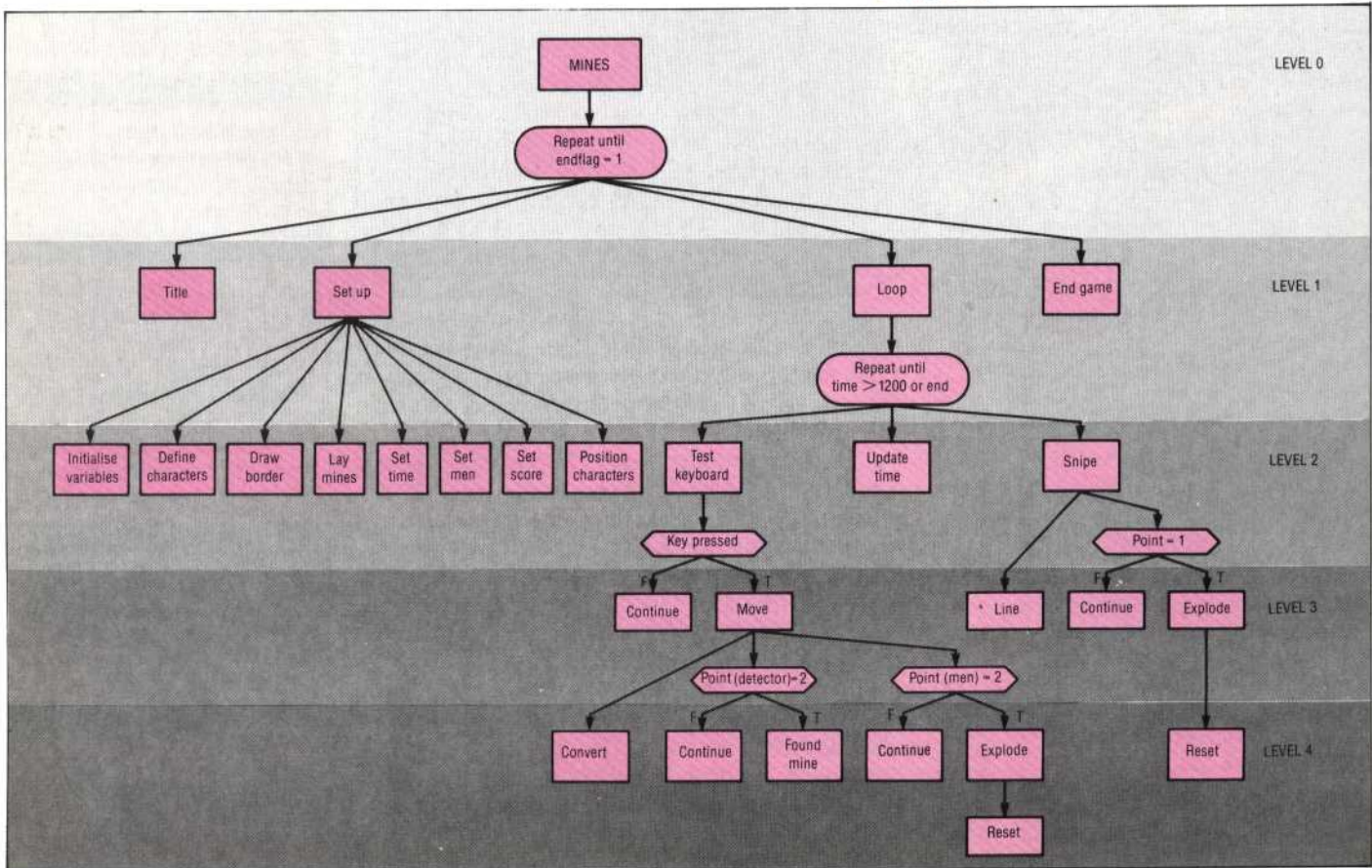
BBC BASIC has two major advantages for the programmer over 'standard' Microsoft BASIC: it is fast in execution and has features that allow you to structure programs. The essence of developing a structured program is to develop small, independent sections of code that can be individually debugged before assembly into a larger program. Any BASIC program can be structured to a certain extent by the use of subroutines to code each module of the program, but BBC BASIC has special types of subroutines, known as *procedures*. These can be thought of as blocks of code that are designed to do a specific job within a program. For example, let us imagine a piece of program that has to pause between each

instruction for a given time. In standard BASIC this may be written using a *dummy loop*; that is, a loop that does nothing except take time to execute:

```
10 PRINT "FIRST SECTION"
20 FOR I=1TO100: NEXT I
30 PRINT "SECOND SECTION"
40 FOR I=1TO100: NEXT I
50 PRINT "THIRD SECTION"
60 FOR I=1TO100: NEXT I
70 PRINT "FOURTH SECTION"
80 END
```

A better approach, however, would be to place the delay loop in a subroutine:

```
10 PRINT "FIRST SECTION"
20 GOSUB100
30 PRINT "SECOND SECTION"
40 GOSUB100
50 PRINT "THIRD SECTION"
60 GOSUB100
70 PRINT "FOURTH SECTION"
80 END
100 REM ** SUBROUTINE **
110 FOR I=1TO100:NEXT I
120 RETURN
```



JANET BARRANCE