DO THE LOGOMOTION

In this instalment of our LOGO course, we will develop a simple game in which the turtle gets lost in space. To do this, we will first need to look more closely at various input and output methods.

In our 'Space Turtle' game, the turtle is stranded in the depths of space, a long distance from its base, to which it must return. The game will require us to print various messages on the screen. The necessary command for this is, not surprisingly, PRINT. Once a message has been printed, the cursor is moved to the beginning of the next line.

To print a single word, PRINT is followed by the word itself — thus, PRINT "HELLO prints the word 'HELLO' on the screen. PRINT " is used to print the 'null word' (a 'word' that has no characters). The effect of this command is simply to print a blank line. If more than one word is to be printed, the text is enclosed in square brackets to indicate that it forms a *list*:

```
PRINT [YOUR TIME HAS RUN OUT]
```

PRINT is also used to display the contents of a variable, so PRINT :SCORE will take the value held in the variable "SCORE and display it. Messages and variable values may be combined in the same PRINT statement by enclosing the complete instruction in *round* brackets, as in:

```
(PRINT [YOUR SCORE WAS] :SCORE)
```

PRINT1 behaves in exactly the same fashion as PRINT does, except that in this case the cursor will remain at the end of the printed text and will not be moved to the next line. This can be demonstrated by entering:

```
PRINT1 [WHAT IS YOUR NAME?]
```

OUTPUT OPERATIONS

LOGO commands, such as HIDETURTLE or PRINT, cause something to happen — they may be said to have an effect on the turtle. However, other LOGO primitives — XCOR, for example — do not have an effect, but instead output a value. This value is then normally used as the input to a command. So, for example, typing:

```
PRINT XCOR
```

would cause XCOR to output the value corresponding to the turtle's current x co-ordinate to the command PRINT, which then displays the result. Thus, if the current value of XCOR is 20, PRINT XCOR will cause the number 20 to appear on the screen. If XCOR is typed on its own, the message RESULT: 20 will appear. This is actually an error message (LCSI versions are somewhat less polite and would print YOU DON'T SAY WHAT TO DO WITH 20).

The procedures we have so far written have all been commands. To create *operations* we must make use of the primitive OUTPUT. As a simple example, here's a procedure that outputs the distance of the turtle from the origin; this procedure uses SQRT to return the square root of a number:

```
TO DISTANCE
  OUTPUT SQRT (XCOR*XCOR + YCOR*YCOR)
END
```

Try moving the turtle to different screen positions and use DISTANCE to determine how far it is from the origin. For example, SETXY 30 40 PRINT DISTANCE should give the answer 50.

When LOGO executes an OUTPUT instruction it stops running the current procedure, returning control to the procedure that called it. This can be seen in the procedure MAX, which outputs the larger of two numbers:

```
TO MAX :X :Y
  IF :X > :Y THEN OUTPUT :X
  OUTPUT :Y
END
```