command will cause the turtle to move without leaving a trace. The origin (0,0) is in the centre of the screen.

The following procedures perform the same task, and illustrate the difference between turtle and co-ordinate geometry:

```
TO SQUARE1
  REPEAT 4 [FD 50 RT 90]
END

TO SQUARE2
  SETXY 0 50
  SETXY 50 50
  SETXY 50 0
  SETXY 0 0
END
```

Typing SQUARE1 or SQUARE2 after returning to DRAW will give exactly the same result. But what happens if you wish to rotate the two squares through 30 degrees? RT 30 SQUARE1 works correctly in the first case, but the second procedure needs to be completely rewritten (because it specifies absolute screen co-ordinates), and this is not a trivial task. But there are times when the use of co-ordinates in LOGO is helpful: as you will see from these examples, SETXY is considerably faster at drawing lines than FORWARD is.

Another feature of turtle geometry is its 'short-sightedness'. The turtle concerns itself only with a single movement at a time; it builds up shapes by taking a series of short 'steps'. Let's play turtle and consider how a circle is constructed. Imagine that you are the turtle — what would you need to do to produce a circle shape? You would move forward for a short distance and turn a little — and you would repeat this sequence many times. In LOGO terms, this translates into something like:

```
TO CIRCLE
REPEAT 360 [FD 1 RT 1]
END
```

This procedure works, but it runs very slowly. It can be made to run faster if the turtle is not drawn at every step. The LOGO command HIDETURTLE is used to make the turtle invisible — SHOWTURTLE draws it again. Our example is really drawing a 360-sided polygon: the lines that form it are so short that they give the appearance of a smooth curve. In fact, 360 sides are more than enough to give the illusion of a circle — a 36-sided polygon is sufficient. So the following procedure draws an acceptable circle, and draws it at a considerably greater speed:

```
TO CIRCLE
  REPEAT 36 [FD 10 RT 10]
END
```

Now try writing procedures that will result in a semi-circle, a quarter-circle, and a sixth of a circle; and combine two of these to make a petal shape.

Playing turtle again, how would you move if you wished to spiral in towards a point, rather than travelling in a circle around it? You would still move a short distance before turning, but in this case you would turn more and more each time (or travel a shorter distance for each turn, which comes to the same thing). The spiral procedure here is a little long-winded, as we have restricted it to use only the commands we have introduced to date, but it will demonstrate that principle:

```
TO SPIRAL
  FD 10 RT 10 FD 10 RT 20 FD 10 RT 30
  FD 10 RT 40 FD 10 RT 50 FD 10 RT 60
  FD 10 RT 70 FD 10 RT 80 FD 10 RT 90
END
```

ALAN COODE COURTESY OF VALIANT DESIGNS LTD.

**LOGO Demo**
Anthony Ginn's Demolition Turtle game introduces the floor turtle and helps to encourage spatial awareness in young children. One player places a toy tower somewhere on the board, and chooses the turtle's start position and heading. The other player then programs the turtle to hit the tower, using as few commands as possible

## Logo Flavours

LOGO editors are very similar, but each has its own peculiarities because of the particular keyboard of the machine. Consult the LOGO manual for your machine.

For all LCSI versions the command to edit the procedure SQUARE is EDIT "SQUARE (quotation mark before but not after the name of the procedure).

To set the turtle at (20,30) use SETPOS [2 30] on all LCSI versions.

## Exercise Answers

**Triangle** REPEAT 3 [FD 50 RT 120]
**Pentagon** REPEAT 5 [FD 50 RT 72]
**Hexagon** REPEAT 6 [FD 60 RT 60]
**Rectangle** REPEAT 2 [FD 25 RT 90 FD 50 RT 90]
**Parallelogram** REPEAT 2 [FD 25 RT 70 FD 50 RT 110]
**Rhombus** REPEAT 2 [FD 50 RT 70 FD 50 RT 110]
**Stars:**

1. REPEAT 5 [FD 50 RT 144]
2. REPEAT 8 [FD 50 RT 135]
3. RT 30 REPEAT 3 [FD 50 RT 120]
   PU LT 30 FD 29 RT 90 PD
   REPEAT 3 [FD 50 RT 120]
4. REPEAT 10 [FD 50 RT 108]