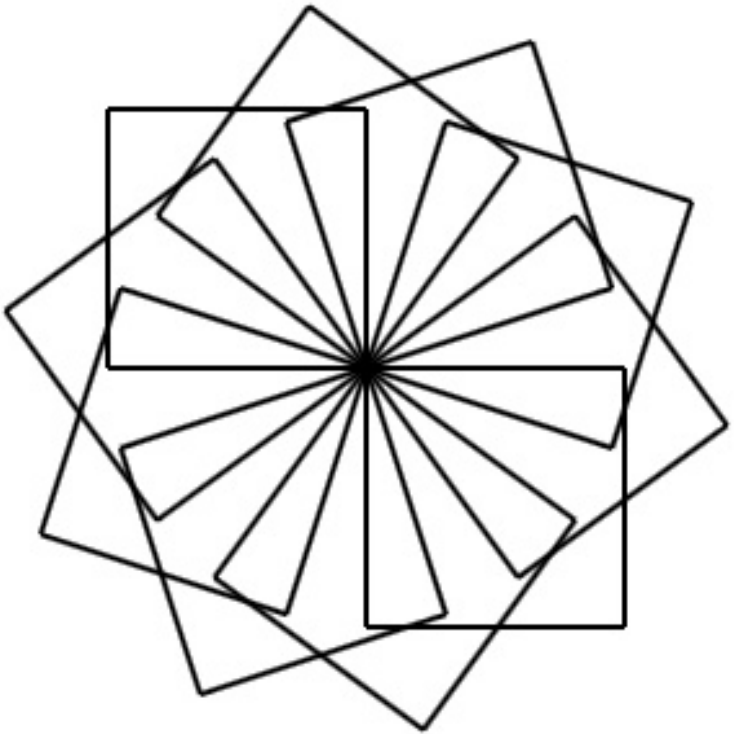
Getting graphical with turtles

Turtles have had a long association with education. First appearing as floor ‘robots’ that could be programmed to move around the floor, and with the addition of a pen they can be made to draw squares or more complicated shapes. Eventually the turtles made their way on to a computer screen where they were programmed to draw fabulous patterns using a language called ‘Logo’, which was developed in the 1960s. Today there are many variants of Logo including a version with Small Basic.

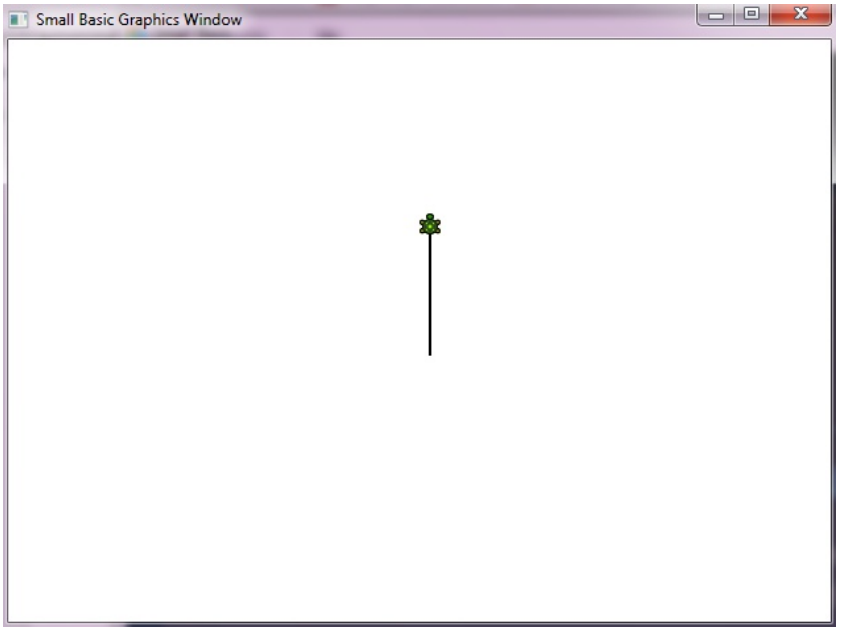
Small Basic provides a Turtle object, which can be moved around the screen, using the Turtle's functions, to draw beautiful pictures. Here is an example.



To create a turtle and make it move by up the screen by 100 pixels we can use the following command.

```
1 Turtle.Show()
2 Turtle.Move(100)
```

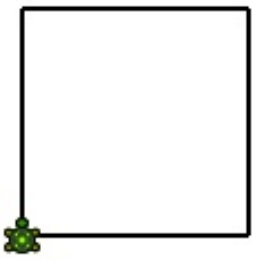
When this program is run, the black TextWindow disappears. This is because the turtle is a graphical object and needs to be displayed in a GraphicsWindow object.



Drawing a square

We can use the Turtle object's, functions and properties to draw a square.

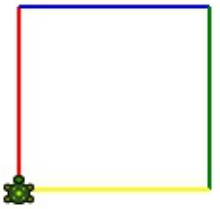
```
1  'This program draws a square.
2  Turtle.Show()
3  Turtle.Move(100)
4  Turtle.TurnRight()
5  Turtle.Move(100)
6  Turtle.TurnRight()
7  Turtle.Move(100)
8  Turtle.TurnRight()
9  Turtle.Move(100)
10 Turtle.TurnRight()
```



Using Colour

The turtle object does not have any colour properties, however the GraphicsWindow object does and can be used to change the pen colour.

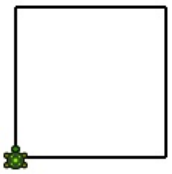
```
1  Turtle.Show()
2  GraphicsWindow.PenColor = "red"
3  Turtle.Move(100)
4  Turtle.TurnRight()
5  GraphicsWindow.PenColor = "blue"
6  Turtle.Move(100)
7  Turtle.TurnRight()
8  GraphicsWindow.PenColor = "green"
9  Turtle.Move(100)
10 Turtle.TurnRight()
11 GraphicsWindow.PenColor = "yellow"
12 Turtle.Move(100)
13 Turtle.TurnRight()
```



Looping and repeating with turtles

When we examine our first square drawing program we notice that after the first statement Turtle.Show(), that there are just 2 unique lines which are repeated 4 times. This is a very inefficient way to program, fortunately we already know a way to repeat this section of code, the For Loop. The next program uses Small Basic's For Loop to reduce our program from 10 to 5 lines.

```
1 Turtle.Show()
2 For count = 1 To 4
3   Turtle.Move(100)
4   Turtle.TurnRight()
5 Endfor
```



In this program we use the For keyword followed by a variable which is initially assigned the value 1. The instructions between For and Endfor are repeatedly executed (i.e. the program loops between lines 2 and 5) in this case 4 times.

How does the computer know when to stop looping? Initially the variable count is assigned the value 1 but on each repeat it is increased by 1 until it is equal to more than 4, after which the looping stops and the program moves onto line 6.

Activity 5: drawing with turtles

Task 1

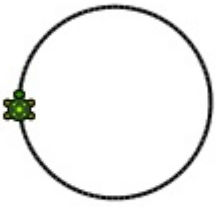
Write a program that uses the turtle draw a pentagon. Pentagons have 6 sides and the turtle will need to turn 60 degrees each time so use Turtle.Turn(60) rather than Turtle.TurnRight().

Task 2

Draw a picture of anything you like: a house , a train or an alien.

Task 3

Can you write a program that uses the turtle to draw a circle. Use a many sided polygon to approximate a circle.



Task 4

Try making common polygons such as equilateral triangles, kites and pentagons.

Task 5

What shape will the following code make?

```
Turtle.Show()
xPos = Turtle.X
yPos = Turtle.Y
Turtle.MoveTo(xPos, yPos + 30)
Turtle.MoveTo(xPos + 40, yPos)
Turtle.MoveTo(xPos, yPos)
```