RECURSION, COMPUTERS AND ART

Andy Kemp describes how *Geomlab* can introduce pupils visually to the concept of recursion.

I was recently introduced to a piece of software called *Geomlab*[1], which is a functional programming language used to describe pictures that are made up of tiles. In itself, that doesn't sound very interesting or very mathematical, but when you see that it can be used to create pictures such as those shown on these pages – in a meaningful and quite simple way, which reads like mathematical notation – you start to wonder if there might be something here that our students should know about.

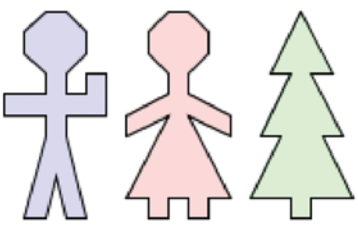
The art of MC Escher is not new to the mathematics classroom, but for the most part all we do is show it to the students and expect them to see the same beauty and mathematics that we see. But do they? Do most of our students have any sensible framework to interpret the concept of 'recursion'? Where would it fit into our already overfull curriculum? The idea of recursion is mentioned in passing when looking at term-to-term rules for sequences, but is quickly rejected in favour of the 'more powerful' n^{th} term rule, but we do the concept an injustice if we leave it there. Recursion is a very powerful mathematical concept; in fact, it is the foundation of our number system! We define the natural numbers using recursion as follows:

0 is in $\mathbb{N}$

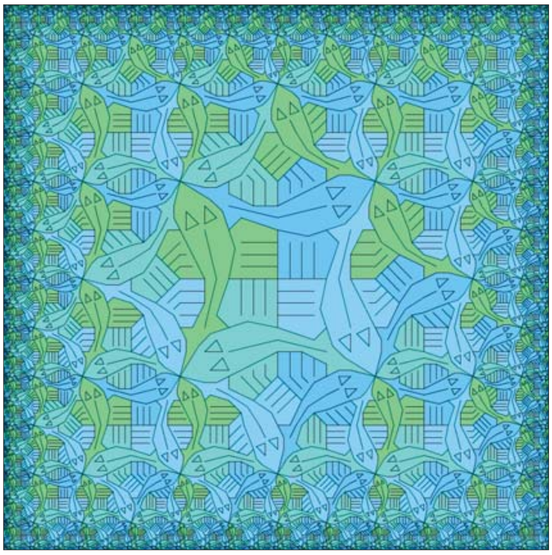
If n is in $\mathbb{N}$, then $n + 1$ is in $\mathbb{N}$

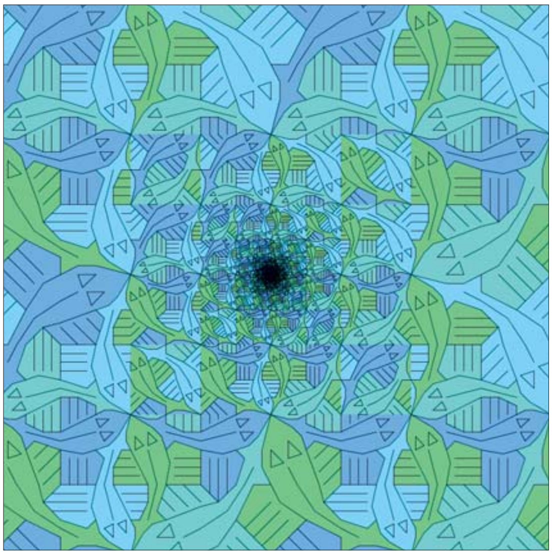
The set of natural numbers is the smallest set satisfying the previous two properties.

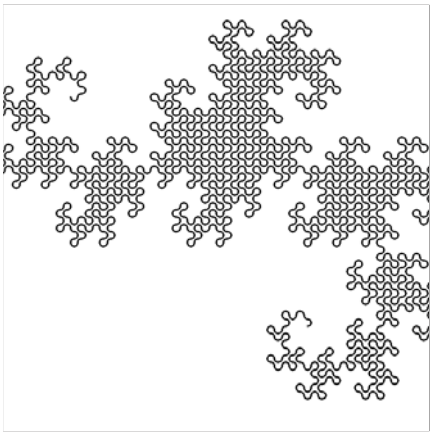
The beauty of *Geomlab* is that it introduces students to recursion through a very simple and enticing graphical environment. Alongside the software is a series of eight worksheets which lead into producing some very sophisticated images. The first three worksheets introduce students to the basics of the software. Students start by learning the syntax of this new 'microworld', in which everything is made up of tiles which can be combined using various functions. The program comes with various preprogrammed tiles; including graphics for a man, woman, tree, and the tiles required to generate the Escher style image.

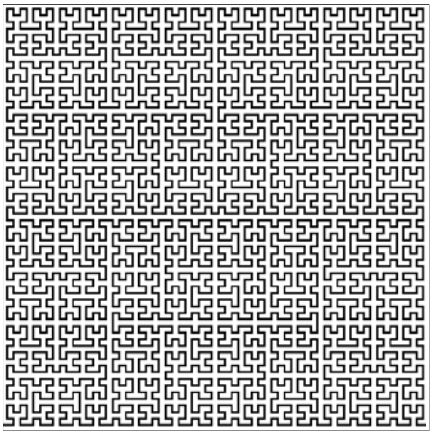


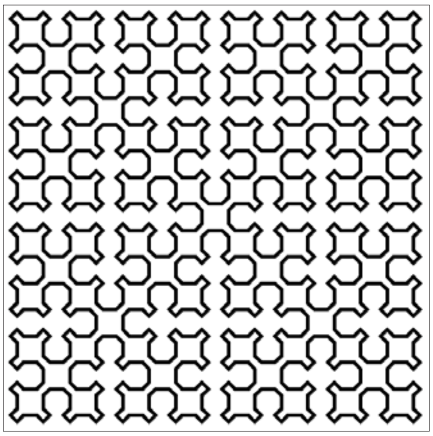
Each of these tiles can be joined together in a number of ways;
eg, `man $ tree` puts a tree next to a man:

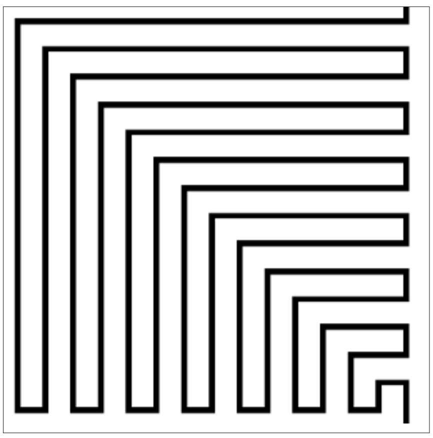


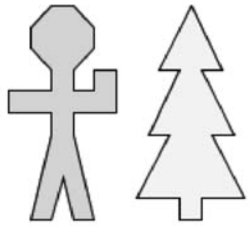




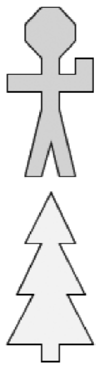




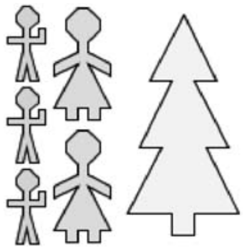




But `man & tree` puts a tree *under* a man:



Using just these commands, we can create complicated images such as:



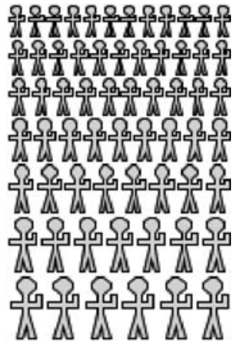
This one is made using the command:
`(man & man & man) $ (woman & woman) $ tree`

Notice how it automatically resizes the men so that three men are the same height as two women, which are the same height as one tree.

Next, students learn about the flip and rot (rotate) commands, which enable them to manipulate tiles in more complex ways. Then students learn how to define new functions within the software. The fourth worksheet begins to introduce recursion properly by talking about men standing in a row. You can say that n men in a row can be defined as 'take a row of $n - 1$ men and add 1 man to the end'. To make this work, however, you need a starting point; so we say '1 man standing in a row is just one man". In formal notation this is written as:

```
define manrow(n) = manrow(n-1) $ man
when n > 1 | manrow(1) = man
```

Extending this idea, we can define a crowd, adding the effect of perspective by taking a row of men and adding a row of $n + 1$ men behind and $n + 2$ behind that and so on. To make this work, we have to tell it where to start and stop, so `crowd(6,12)` would have 6 men on the first row, then 7, 8, 9, ... and then 12 on the back row. This produces the image:



Next, students are encouraged to try to define spirals and zigzags recursively, before they are led through the process of producing their own Escher image. The final two worksheets introduce them to the concept of space-filling curves and fractals. The self-similar nature of fractals lends itself well to the recursive concept, and getting students to think about their construction gives them a much greater insight into their nature.

When I used the software with a Y9 group, we spent around 4-5 hours over two weeks working with the software, and over this time virtually all the students managed to gain a good grasp of it and produce their own brightly coloured Escher image. They really enjoyed the process, and looked forward to the lessons. As the software is freely available on the internet, some students even continued with the work at home! I am optimistic that when some of them begin to meet the ideas of recursion at A-level they will do so with the added insight they have gained through this process.

So if you are looking for something different to do with your classes, you could do a lot worse than introduce your students to the exciting and powerful world of recursion through the medium of *Geomlab*!

Andy Kemp teaches mathematics at Warwick School.

Note

1 http://web.comlab.ox. ac.uk/geomlab/, created by Michael Spivey of Oxford University with funding from NAGTY.

Go to www.atm.org.uk/mt/ for a set of functions which can be used to generate these images.