MCS 320 Project Two : Fractal Curves due Friday 11 November 2005, at 2PM.

In this project we will use recursive procedures to plot self similar curves. A good start of the project is to download the companion worksheet to this project description from the class web site.

1. Sierpinski's curves

A triangle is plotted by the command `polygonplot`:

```
[> v := [[0,0],[1,0],[0,1]]: # vertices of the triangle
```

With `style = line`, `polygonplot` only draws the edges of the triangle. If this option is omitted, the triangle is considered as solid. In this case, we may change the default white color into green with the option `color = green`. This is illustrated below:

```
[> plots[polygonplot](v,style=line,axes=none);
[> plots[polygonplot](v,color=green,axes=none);
```

The procedure `gasket` below plots part of Sierpinski's triangular curve, also known as the Sierpinski gasket. The five arguments of gasket are

1. a is the first coordinate of the leftmost corner of the triangle;
2. b is the second coordinate of the leftmost corner of the triangle;
3. L is the length of the horizontal and vertical sides of the triangle;
4. k is a counter, it typically equals 0 when the user calls gasket;
5. m is the depth of the tree of recursive calls.

The triangle is divided into four similar triangles. The central triangle is taken out. The removal of the central triangle is repeated on the other three remaining triangles. As this process goes on recursively for ever, it is not too difficult to show that the area of the removed triangles equals the area of the entire triangle. Therefore, what remains has area equal to zero and counts as a curve.

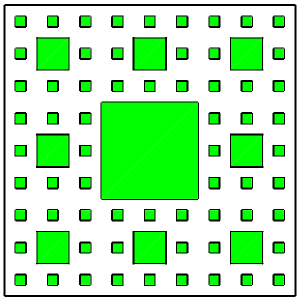
We save the procedure in the file `gasket.mpl`, and load it into a Maple worksheet via the command `read`, and then typically use the procedure as follows:

```
[> read("gasket.mpl");
[> p := gasket(0,0,1,0,2):
[> plots[display](p,axes=none);
```

Assignment One. The goal of this assignment is to write a Maple procedure to plot Sierpinski's rectangular curve, also known as the Sierpinski carpet. Sierpinski's rectangular curve starts by dividing a rectangle (we will take a square) into nine similar rectangles. The central rectangle is removed. The removal process is applied to the eight remaining rectangles.

The parameters `a`, `b`, `L`, `k`, `m` of the procedure `carpet` have the same meaning as in the procedure gasket. The coordinates of the leftmost corner of the carpet are given by `a` and `b`, `L` is the length of each side. The parameter `k` controls the depth of the recursion tree, and should be initially equal to zero. The maximal depth of the recursion tree is given by `m`.

```
[> read("carpet.mpl");
[> p := carpet(0,0,1,0,2):
[> plots[display](p,axes=none);
```



2. The Menger Sponge

There is a three dimensional analogue to the Sierpinski gasket, the so-called Menger sponge. The elementary command from plottools we use is hexahedron, e.g.:

```
[> hx := plottools[hexahedron]([0,0,0],1):
[> plots[display](hx,axes=boxed);
```

defines a cube centered at `[0,0,0]` with sides of length 2 for all directions, illustrated by the plot above.

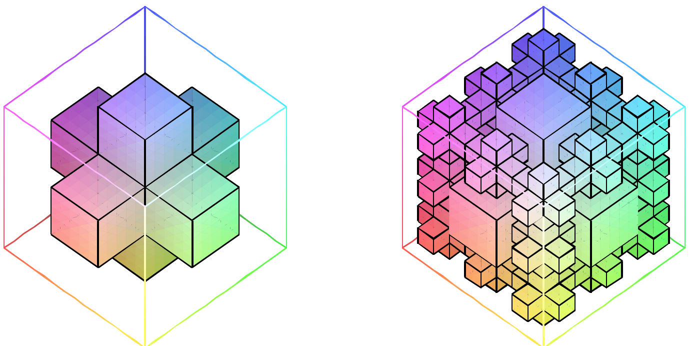
The Menger sponge is obtained by taking a cube and "drilling through" the center of each face. As the cube has six faces, in the first step, we omit from the big cube seven smaller cube whose sides are one third of the original length. We continue applying the same drilling to each of the smaller remaining cubes.

Assignment Two. A first method to plot the Menger sponge proceeds in the same fashion as with the Sierpinski carpet: we plot the cubes omitted from the original cube.

The procedure `sponge` has six parameters: `a`, `b`, `c`, `L`, `v`, and `m`. The center of the cube is defined by the first three parameters: `a`, `b`, `c`. The length of the first cube is `L`. The parameter `v` is a counter, used to control the level of recursion in the tree and initially called as 0. The `m` is the maximal depth of the recursion tree. Provide an implementation for the procedure `sponge`.

```
[> read("sponge.mpl"):
[> p0 := sponge(0,0,0,1,0,0): plots[display](p0);
[> p1 := sponge(0,0,0,1,0,1): plots[display](p1);
```

The two plots are displayed next to each other below:



Assignment Three. A second method to plot the Menger sponge is to plot what remains after drilling the holes in the cube. The elementary plotting command we use is

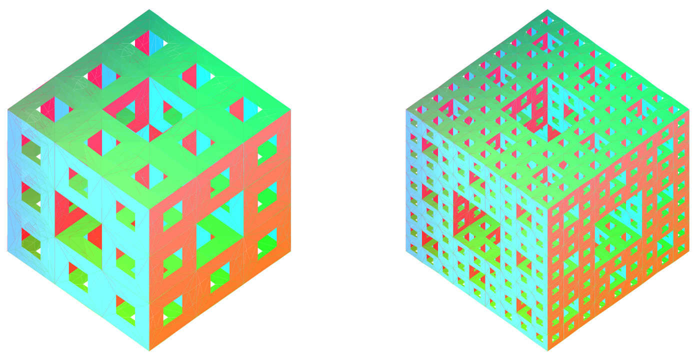
```
[> p := plottools[cutout](plottools[hexahedron]([0,0,0],1),0.4):
[> plots[display](p);
```

which shows the cube centered at `[0,0,0]`, all sides have length 2, and their area is reduced with 40%.

This reduction factor is an additional parameter in the procedure `menger`, as fifth parameter `f`, occurring after the length `L`, and before the parameter `v` which controls the recursion level. Following are some examples:

```
[> read("menger.mpl"):
[> p1 := menger(0,0,0,1,0.4,0,1): plots[display](p1);
[> p2 := menger(0,0,0,1,0.4,0,2): plots[display](p2);
```

The two plots are displayed next to each other below:



Provide an implementation for the procedure `menger`, which has as parameters `a`, `b`, `c`, `L`, `f`, `v`, and `m`.

3. Dimension Calculations

The Sierpinski gasket is a curve because the area of what we remove from the triangle equals the area of the entire triangle. So the area of the Sierpinski gasket is zero, whence we call it a curve. The purpose of the next assignment is to validate this statement.

Assignment Four. Use Maple to show that the Sierpinski gasket and carpet have zero area.

To show that the area is zero, add up the area of all the pieces which are removed from the Sierpinski curves, gasket and carpet, and show that the area of the removed pieces sums up to the area of the respective original triangle and rectangle. Use Maple's `sum` command for this.

Also show that the volume of the Menger sponge equals zero, using the same method.

4. The deadline is Friday 11 November 2005 at 2PM

Bring *your* solution to the project to class. The *your* is emphasized to stress that your solution is the result of an *individual* effort. Collaborations are **not** permitted.

The solution to this project consists in three parts:

1. The Maple procedures to make the plots. As done in the worksheet, save the procedures on separate files. Either you could print out the procedures separately, or show them as the output of the `read` command in the print out of the worksheet.
2. A print out of the Maple worksheet that you bring to class.

   Deliver a well written document, with grammatically correct and complete sentences, without spelling mistakes, appropriately structured into sections and subsections.
3. The (**only one!**) Maple worksheet that you eMail as an attachment to me.

   The worksheet should run as a computer program, from top to bottom with consistent output and without errors.

If you have questions or difficulties with the assignments, feel free to come to my office for help.