

## Assignment 1 — Graphics and Fractals

Due: 12 noon on Monday 28th August, 2006

---

### Outline

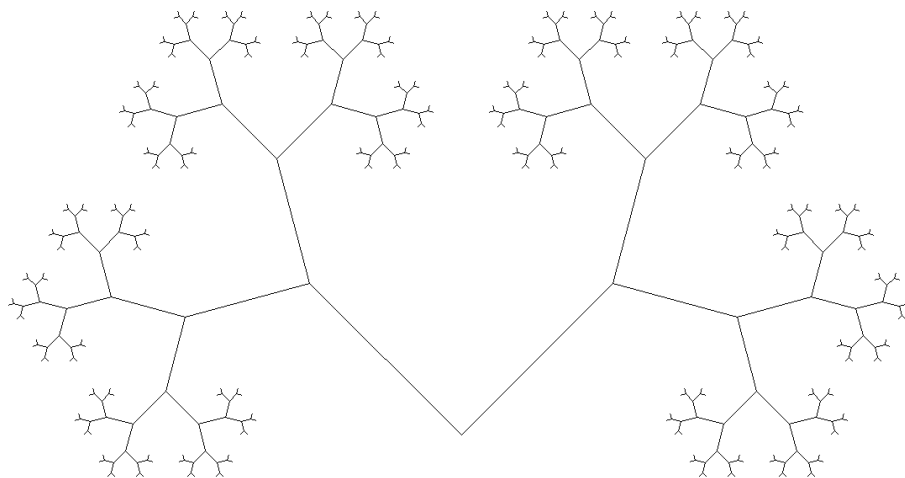
Your aim in this assignment is to write several programs to display recursively generated animations. You should use the ANUPlot graphics library which was discussed in lectures.

The ANUPlot library includes a number of demonstration programs which will serve as a useful starting point for this assignment. It is suggested that you experiment with several of these demos before starting on your own programs.

ANUPlot also includes online documentation describing each of the major functions and data structures provided by the library. This documentation can be accessed by using a web-browser (eg Konquerer, Mozilla) to open the `doc/index.html` file from the root directory of the library. Not all of this documentation will be relevant to this assignment as it also includes a description of the internal structure of the library along with the parts needed by client programs. The most useful pages will be the ones describing the `Graphics.Plot.Display` and `Graphics.Plot.Picture` modules.

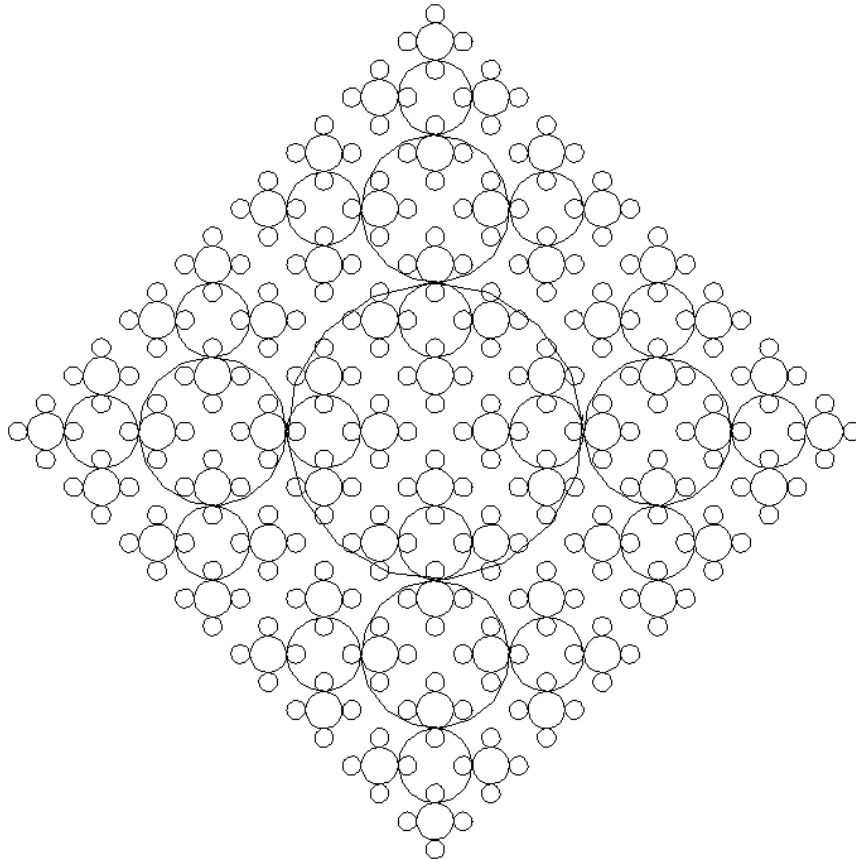
### Required Task 1: Coral Fractal

Write a program to display a fractal similar to the one below. Your initial program should not use animation, color or any other features of the library. Once you have produced this program, write a new version which animates the fractal in some way and uses an interesting color scheme.



## Required Task 2: Circle Fractal

Write a program to display a fractal similar to the one below. As per the previous task, you should produce two versions of the program - one with animation and color and one without. ANUPlot does not provide a constructor for the `Component` data type to draw circles. You will need to write a function to draw a circle first, or else salvage one from one of the demonstration programs.



## Free Choice Tasks

Successful completion of the *Required Tasks* above is sufficient to achieve a *Pass* grade on this assignment. For marks better than a *Pass*, you will need produce some animations of your own design.

Many interesting animations can be obtained simply by applying `Scale` and `Rotate` transformations to the components of a fractal while it is being generated. Interesting effects can also be obtained by using functions such as `sin` and `cos` to modulate color values and animation speeds.

Submit a small selection of your own experiments, where 'small selection' means no more than three or four. Higher marks will be given to animations which use recursion and have a different overall structure than the ones in the required tasks section.

## Using the ANU Graphics Library

The ANU Graphic Library is not installed with the GHC libraries, so it is most convenient if you put it in the same directory as the scripts you are developing. For example, if your working directory is `comp1100/Asst1/`, then you should copy the entire contents of the directory `/dept/dcs/comp1100/public/ANUPlot/` including all subdirectories into your working directory.

There are various examples in the directory, including the snowflake fractal `Snowflake.hs`. You can compile the programs, or load them into GHCi and evaluate `main`. The graphics library uses some non-standard extensions to Haskell, so the command must be

```
ghci -fglasgow-exts -isrc src/Demo/Snowflake.hs
```

or

```
ghc -fglasgow-exts -isrc --make src/Demo/Snowflake.hs -o bin/snowflake
```

Be sure to take a look at the README file in ANUPlot's root directory. Ask your tutor during the lab session this week if you need help getting started.

## Submission

You should submit all the modules that you have written and wish to be included in your assessment, by the published deadline. Late submissions will be penalised 10% for each day (or part thereof) past the deadline.

Assignment submission will be electronic. Submit your assignment with the command:

```
submit comp1100 Asst1 <file names>
```

Only submit your own modules. You should not have modified any of the files from the ANUPlot library.

Make sure you practice submitting your files well before the deadline. It is possible and perfectly acceptable to submit updated versions of your files. Our system will collect them all and your most recent submission will be the one assessed.

You should check that your submissions have been successful by clicking the **View Marks** button in **StReaMS** (<http://cs.anu.edu.au/streams/>). You should see a list of files together with the time and date of submission.

## Assessment

This assignment forms 10% of your total assessment for COMP1100.

There will be two components to the assessment. During week 7 practical classes you will present and demonstrate your programs to your tutor and other members of the class. This will form 50% of your marks for this assignment. The second 50% will be based on your tutor inspecting your submitted modules. Marks will be allocated for programming style, comments, layout, technique and such.