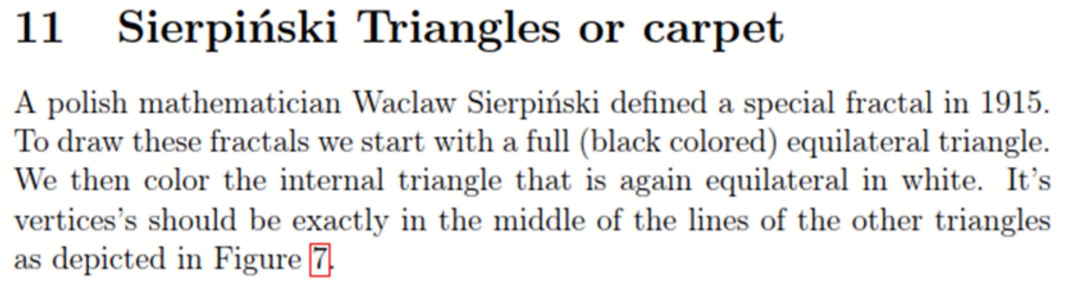

11 Sierpiński Triangles or carpet

A polish mathematician Waclaw Sierpiński defined a special fractal in 1915. To draw these fractals we start with a full (black colored) equilateral triangle. We then color the internal triangle that is again equilateral in white. It's vertices's should be exactly in the middle of the lines of the other triangles as depicted in Figure 7.

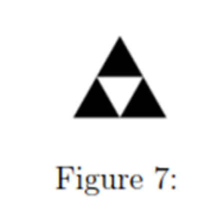

Figure 7:

We obtain three new full(black) equilateral triangles. We perform the same procedure on those triangles as illustrated in Figure 8.

- Length of the lines of the starting triangle
- Stopping condition (number of recursive calls)

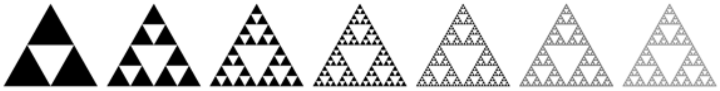

Figure 8:

Develop an algorithm that draws the fractals. The algorithm should draw the final result using a method DrawLine. The processes should compute the position and the distance of all lines that form a fractal and insert them in a data structure. When the algorithm reaches a maximum number of recursive calls, the main process draws all lines in the data structure to obtain a visual representation of the fractal.

The example describes Sierpiński Triangles. However, you can also choose to compute the Sierpiński Carpet where the process is very similar with the exception of starting with a square.

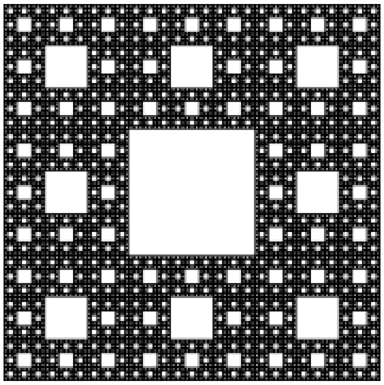

Figure 9:

11.1 Implementation guidelines

1. Implementation of the graphical interface:

   - The program should have parameters with which the graphical interface can be toggled to achieve best performance
   - The drawing of the graphics should be done independently of the computational threads.
   - The default size of the window is 800x600px but can be adjusted manually.
   - The graphical interface can scale, zoom and move. Initially, it scales to fit all points in the view-port.

2. Running the program:

   - The program can be ran in different modes (sequential, parallel, distributed) by specifying a parameter.
   - The program measures run-time needed to complete.

3. Problem specific implementation requirements

   - Rendering should not be considered in measuring runtime.
   - The implementation must adapt automatically to the hardware it is being ran on (Physical CPU's, Cores, Memory, etc..);

11.2 Testing

The report must include extensive testing and explanation of results (numeric and graphical). All three versions must be tested. The parameter that influence the runtime is the number of recursive calls the fractal is computed. Consequently, multiple configurations must be tested to observe how the implementations scale. The first configuration is 5 recursive calls, to obtain the next configuration, increase the recursive calls by 1. Stop testing new configurations when the sequential implementation takes more then 10 minutes to compute. Each configuration should be ran at least three times with the average running time as the final result.