Graphics Assignment 05 | 21/18

Title: fractal pattern using koch curve

Problem statement: Write c++ program to generate fractal patterns by using koch curves

Objective: Learn & understand how koch curves generate fract patterns.

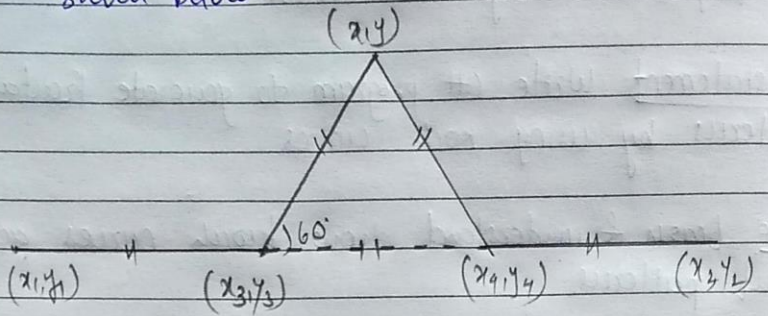
Outcomes: Generate fractal pattern by using koch curve

Hardware & Software requirements:
64-bit opensource qt-creator
Windows-10 operating system
8 GB RAM, 512 GB SSD with intel core i5 8th gen quad core processor.

Theory:
Koch Curves: The koch curve fractal was one of the first fractal objects to be described.
To create a koch curve.
1> Create a line & divide it into 3-parts.
2> The second part is now rotated by 60°
3> Add another part which goes from the end of part 2 to the begining of part 3.
4> Repeat step-1 & step-3 with each part of the line.
We will get the koch curve as number of interation goes on increasing.

Algorithm:
Step1: In iteration 0 we have a horizontal line
Step2: In iteration 1 the line is divided into 3-equal parts. Midde part is rotated by 60° so that it forms a perfect equilateral triangle as

shown below.



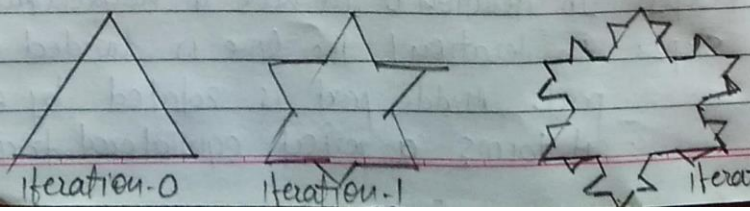
Here (x_1, y_1) & (x_2, y_2) is accepted from user.
Now, we can see line is divided into 3 equal segment.

Step3: $x_3 = (2 * x_1 + x_2)/3;$
$y_3 = (2 * y_1 + y_2)/3;$
$x_4 = (x_1 + 2x_2)/3$
$y_4 = (y_1 + 2y_2)/3;$

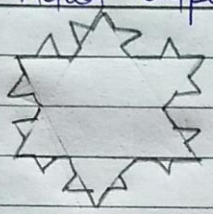
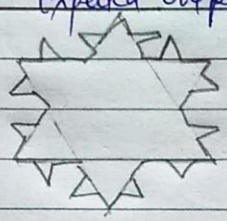
Step4: $x = x_4 + (x_4 - x_3)\cos 60^\circ + (y_4 - y_3)\sin 60^\circ$
$y = y_3 + (x_4 - x_3)\sin 60^\circ + (y_4 - y_3)\cos 60^\circ$

Step5: In iteration 2, you will repeat step 2 for every segment obtained in iteration 1

Snowflake Curve:
It is drawn using koch curves. One koch curve gives us one side of snowflake. Three such koch curves forming an equilateral triangle. 4 iteration would give us a snowflake curve.



Test cases

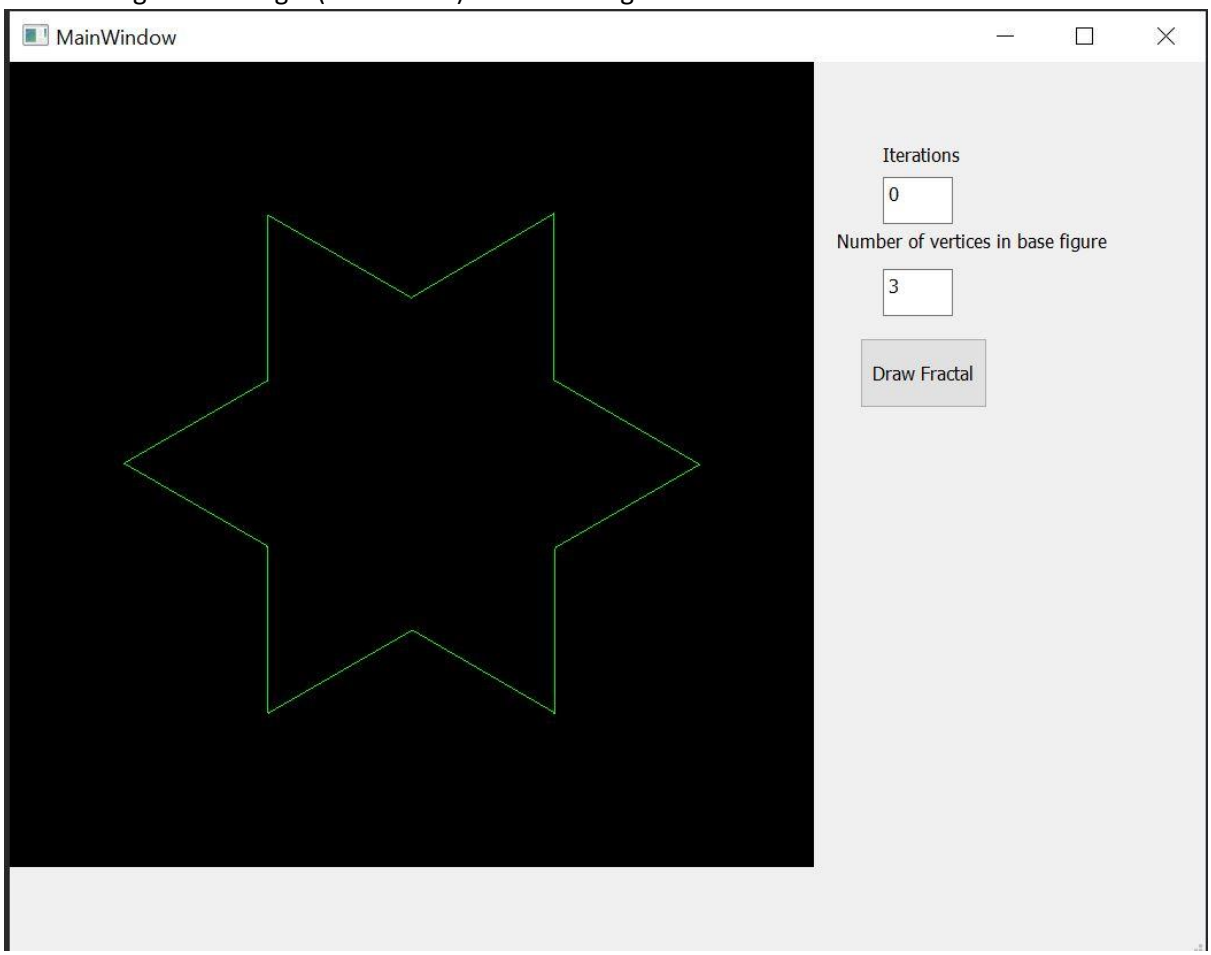
S.NO	Input	Actual Output	Expected Output	Result
1	(100, 100, 100) (200, 100) (150, 186)			pass.

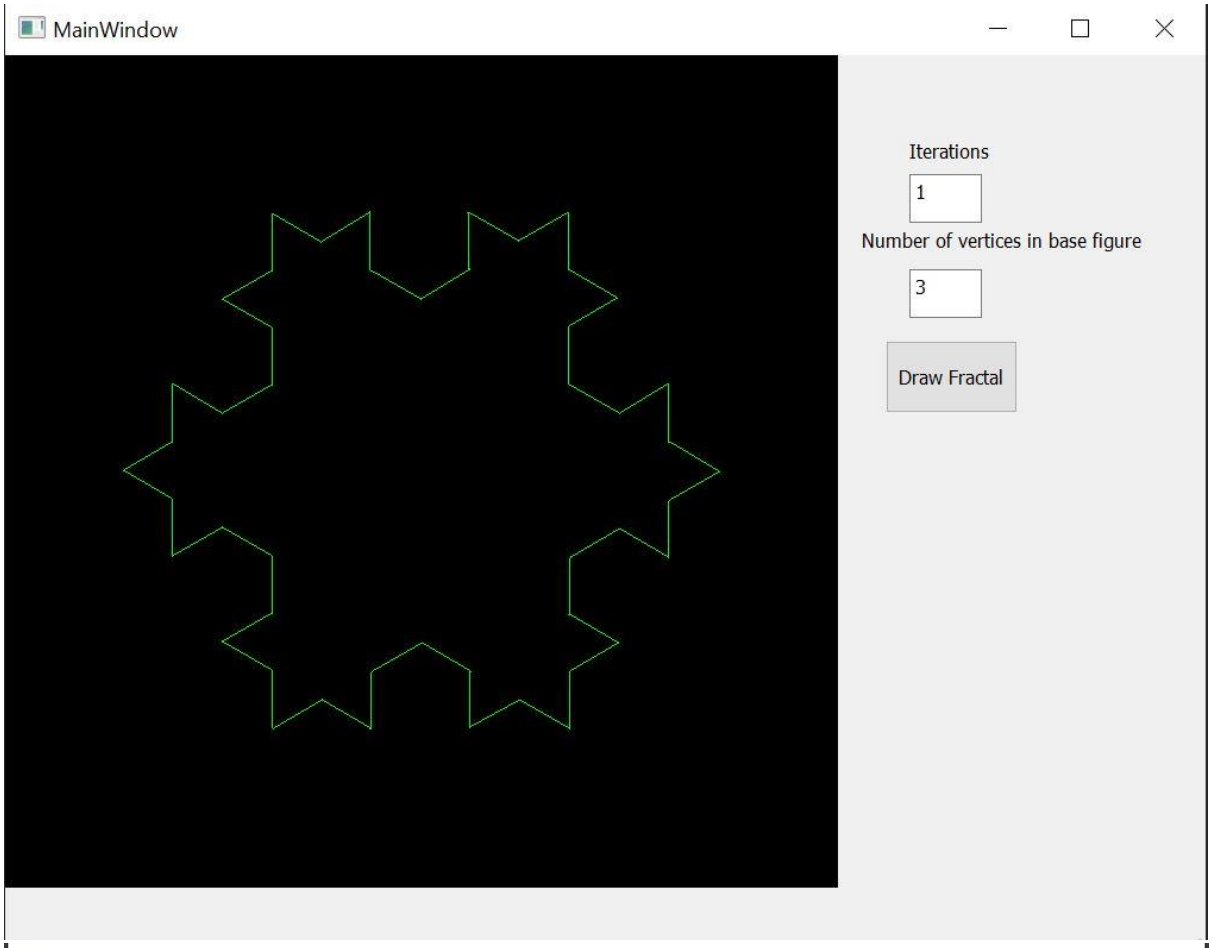
Conclusion

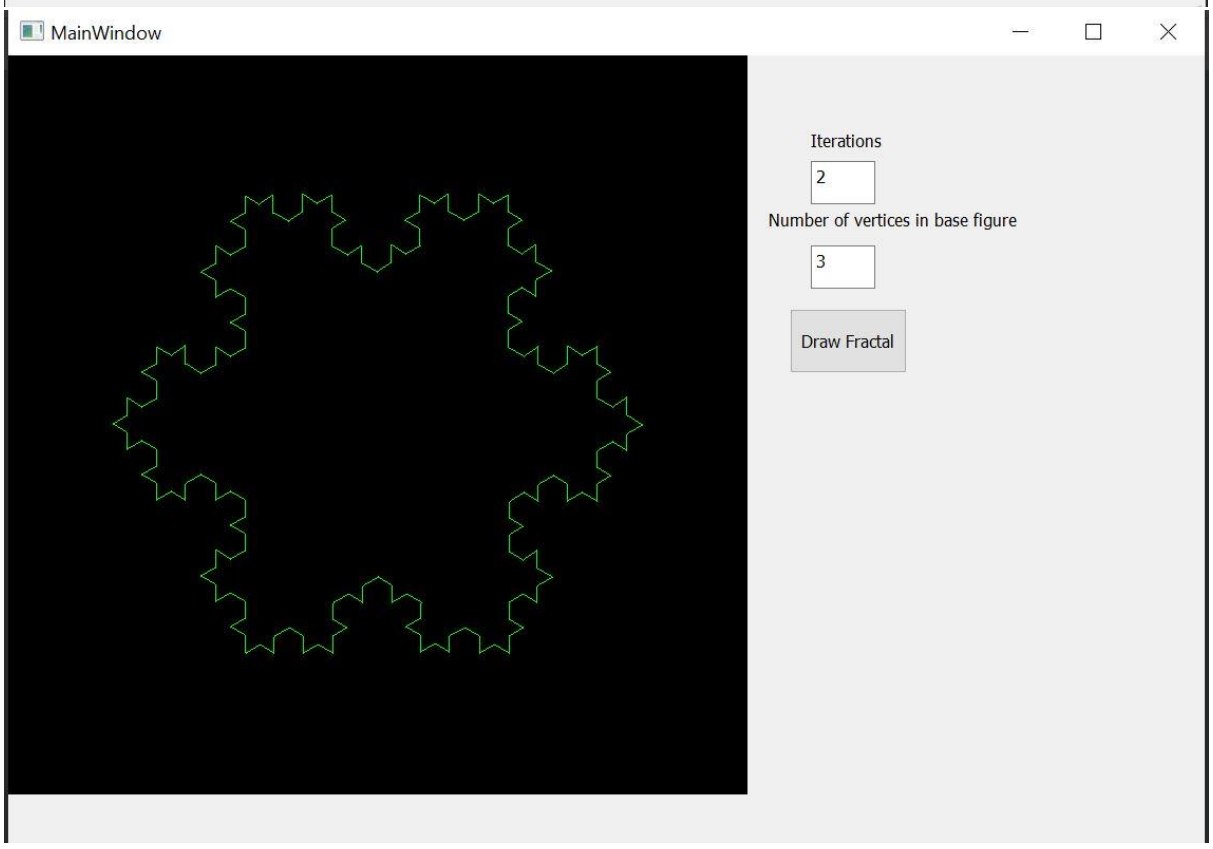
Hence we succesfully generated snowflake & tree pattern using koch curve. in gf creator.

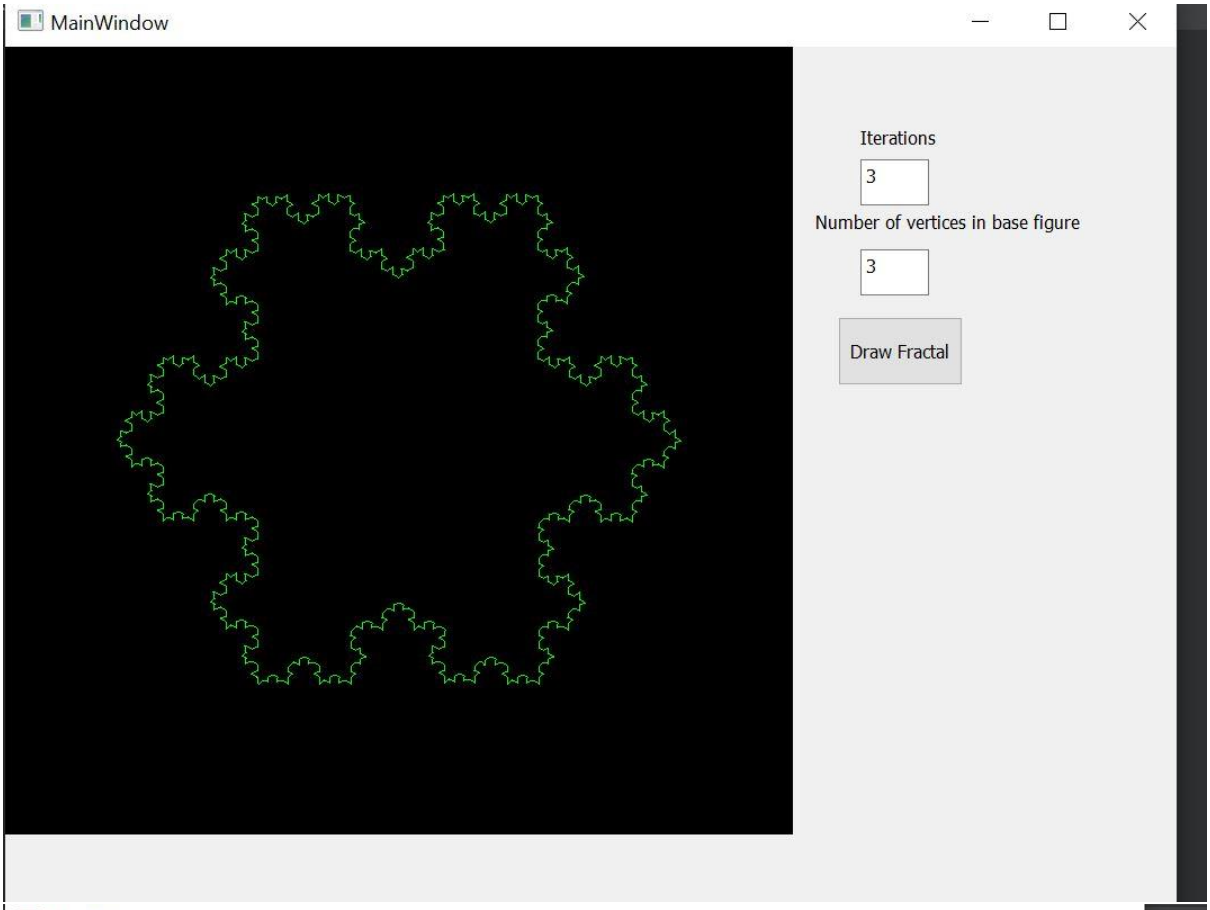
Outputs: for 1, 2, 3, 4, 5 iterations.

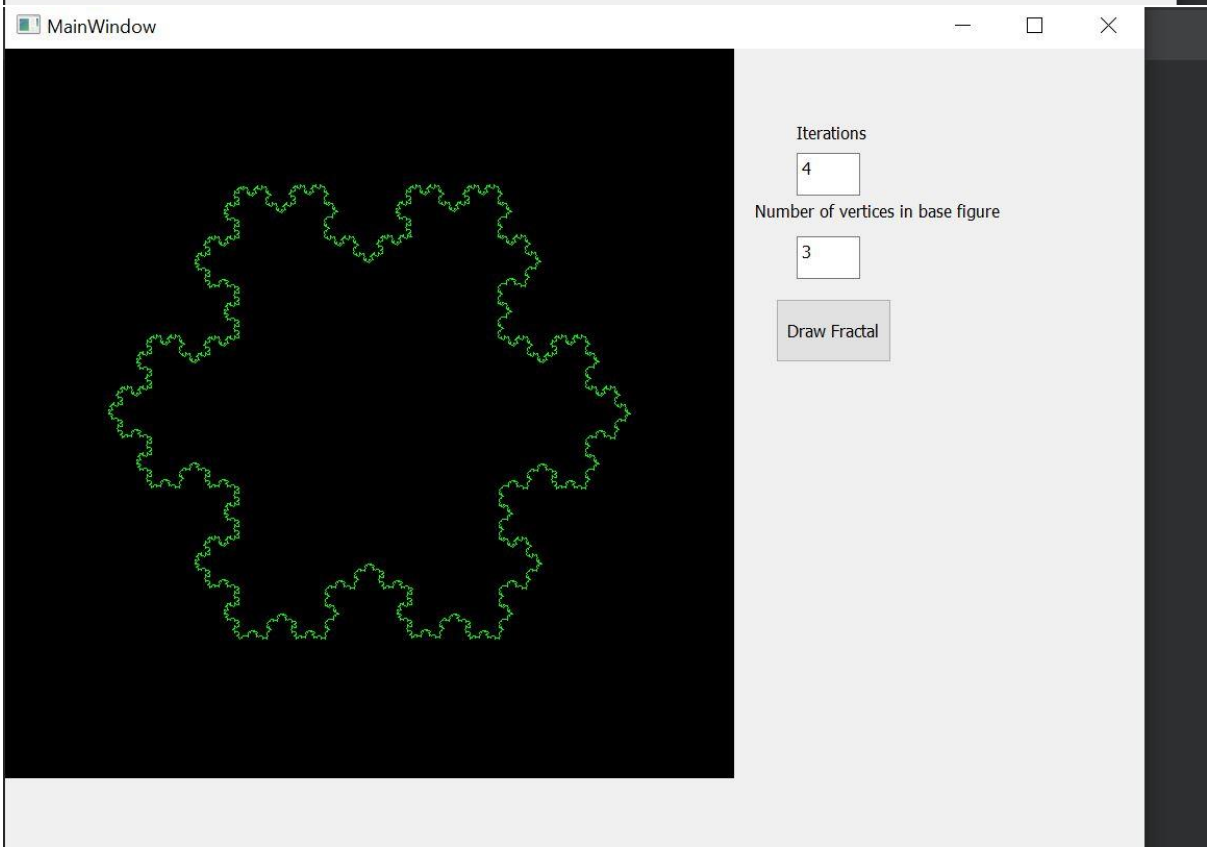
The base figure is triangle (3 - vertices). The counting of iterations is 0 based.




MainWindow
Iterations
1
Number of vertices in base figure
3
Draw Fractal


MainWindow
Iterations
2
Number of vertices in base figure
3
Draw Fractal


MainWindow
Iterations
3
Number of vertices in base figure
3
Draw Fractal


MainWindow
Iterations
4
Number of vertices in base figure
3
Draw Fractal