Lab Exercise 13
CS 2334

November 19, 2015

Introduction

In this lab, you will use recursive logic to create a fractal snowflake, commonly known as a Koch snowflake. Recursion has many practical uses outside of graphics, but fractals provide a great way to visualize the concept.

Learning Objectives

By the end of this laboratory exercise, you should be able to:

1. Define the base and recursive cases of a recursive formulation
2. Correctly call a method recursively
3. Use an **ActionListener** to update graphics based on a **Timer**
4. Read existing code and documentation in order to complete an implementation

Proper Academic Conduct

This lab is to be done individually. Do not look at or discuss solutions with anyone other than the instructor or the TAs. Do not copy or look at specific solutions from the net.

Preparation

1. Import the existing lab13 implementation into your eclipse workspace.

   (a) Download the lab13 implementation: `http://www.cs.ou.edu/~fagg/classes/cs2334/labs/lab13/lab13-initial.zip`

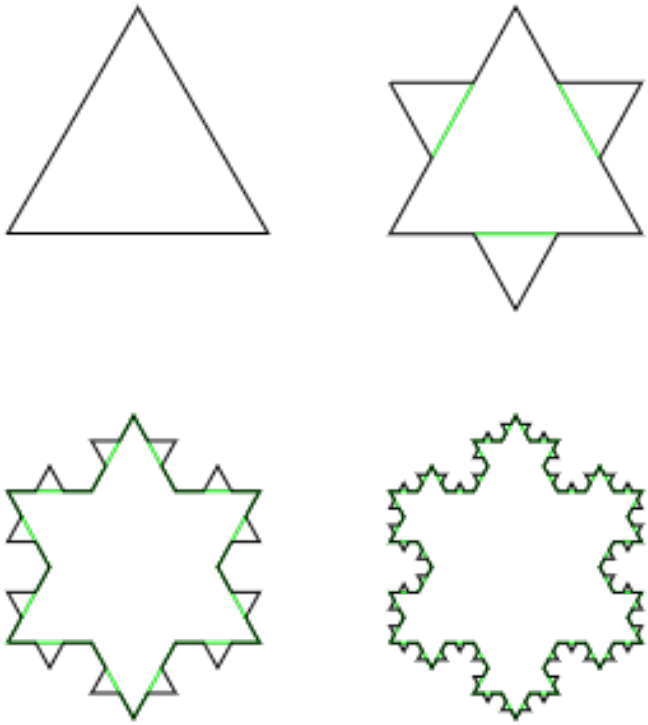
   (b) In Eclipse, select *File/Import*

   (c) Select *General/Existing projects into workspace*. Click *Next*

   (d) Select *Select archive file*. Browse to the lab13-initial.zip file. Click *Finish*

Koch Snowflake

The first four iterations of the Koch snowflake are shown below.
(from https://en.wikipedia.org/wiki/File:KochFlake.svg)



The Koch snowflake can be constructed by starting with an equilateral triangle, then recursively altering each line segment as follows:

1. Divide the line segment into three segments of equal length.
2. Draw an equilateral triangle that has the middle segment from step 1 as its base and points outward.
3. Remove the line segment that is the base of the triangle from step
4. Repeat for each of the resulting line segments

Timer

Once your snowflake is drawn, you will animate it by rotating the orientation slightly and redrawing. This should happen at regular small intervals (not larger than 100 microseconds) to create a smooth and continuous animation.

A Timer can be set to run in your GUI that keeps track of time and raises ActionEvents. Details are provided in the Java API.

Lab 13: Specific Instructions

All of the necessary classes are provided in lab13-initial.zip.

1. Look carefully through the existing code and implement any TODOs.
2. Do not add major functionality to the classes beyond what has been specified, but feel free to use creativity with your snowflakes and animations!
3. Don't forget to document as you go!

Final Steps

1. Generate Javadoc using Eclipse.
   - Select *Project/Generate Javadoc...*
   - Make sure that your project is selected, as well as all of the Java source files
   - Select *Private* visibility
   - Use the default destination folder
   - Click *Finish*
2. Open the *lab13/doc/index.html* file using your favorite web browser or Eclipse (double clicking in the package explorer will open the web page). Check to make sure that that all of your classes are listed and that all of your documented methods have the necessary documentation.
3. If you complete the above instructions during lab, you may have your implementation checked by one of the TAs.

Submission Instructions

- All required components (source code and compiled documentation) are due at 11:59pm on Friday, November 20th.
- Prepare your submission file:
  1. Select the project in the *Package Explorer* window.
  2. Right-click. Select *Export*
  3. Select *General/Archive File*
  4. Expand your project and verify that both the *src* and *doc* folders are selected, as well as all of their contents
  5. Enter the archive file name: *lab13.zip* (note that you may want to browse to a different destination folder)
  6. Select *Save in zip format*
  7. Click *Finish*
- Submit your zip file to the lab13 folder on D2L.

Rubric

The project will be graded out of 100 points. The distribution is as follows:

Implementation: 35 points

Program formatting: 10 points

(10) The program is properly formatted (including indentation, curly brace and semicolon locations).

(5) There is one problem with program formatting.

(0) The program is not properly formatted.

Data types and method calls: 15 points

(15) The program is using proper data types and method calls.

(10) There is one error in data type or method call selection.

(5) There are two errors in data type or method call selection.

(0) There are multiple errors in data type and method call selection.

Required Methods: 10 points

(10) All of the required methods are implemented.

(7) A component of one required method is missing.

(4) A method is not implemented or components are missing from two methods.

(0) Two or more required methods are not implemented or components from three or more methods are missing.

Proper Execution: 30 points

Output: 15 points

(15) The program passes all tests.

(10) The program fails one test.

(5) The program fails two tests.

(0) The program fails three or more tests.

Execution: 15 points

(15) The program executes with no errors.

(8) The program executes, but there is one minor error.

(0) The program does not execute.

Documentation and Submission: 35 points

Project Documentation: 5 points

(5) The java file contains all of the required documentation elements at the top of the file.

(3) The java file is missing one of the required documentation elements.

(2) The java file is missing two of the required documentation elements.

(0) The java file is missing more than two of the required documentation elements.

Method-Level Documentation: 10 points

(10) Every method contains all of the required documentation elements ahead of the method prototype.

(7) The method documentation is missing one of the required documentation elements.

(3) The method documentation is missing two of the required documentation elements.

(0) The method documentation is missing more than two of the required documentation elements.

Inline Documentation: 10 points

(10) Every method contains appropriate inline documentation.

(7) There is one missing or incorrect line of inline documentation.

(3) There are two missing or incorrect lines of inline documentation.

(0) There are more than two missing or incorrect lines of inline documentation.

Submission: 10 points

(10) The correct zip file name is used and has the correct contents.

(5) The correct zip file name is used, but one required component is missing.

(0) An incorrect zip file name is used or more than one required component is missing.