Scott Elliott
Report

Drawing Fractals Using Recursion

This document provides instructions for compiling and running the Koch Curve Generator program, which generates a Postscript file representing a Koch curve of a specified level. The program is implemented in C++ using three files: Koch.h, Koch.cpp, and driver.cpp. The functionality of the separate Turtle class has been consolidated into the Koch class.

Compilation Instructions

The program requires a C++ compiler (such as g++) and the C++ standard library.

Step 1: Compile the Source Files

Use this command to compile all source files and link them into a single excecutable named koch:

```
g++ -std=c++11 -o koch driver.cpp Koch.cpp -lm
```

- -std=c++11: Ensures compatibility with modern C++ features.
- -o koch: Specifies the output executable name as koch.
- -lm: Links the math library (required for std::sqrt and std::atan2).

Step 2: Verify Compilation

If compilation is successful, an executable file named koch will be created in the current directory.

Execution Instructions

The program takes five command-line arguments: the starting point (x_1, y_1), the ending point (x_2, y_2), and the curve level.

Usage Syntax (add ".exe" after "koch" if in Windows)

```
./koch x1 y1 x2 y2 level
```

Example Usage (add ".exe" after "koch" if in Windows)

```
./koch 25 400 575 400 5
```

Argument	Description	Example
x1, y1	Coordinates of the starting point of the level 0 segment. 25 unit left margin.	25 400
x2, y2	Coordinates of the ending point of the level 0 segment. 25 unit right margin.	575 400
level	The desired iteration level of the Koch curve (integer ≥ 0).	5

Example Execution (Level 5)

To generate the Level 4 Koch curve from $(25, 400)$ to $(575, 400)$ and save the output to a Post-script file (if using Windows, replace "./koch" with "koch.exe"):

```
./koch 25 400 575 400 5 > output.ps
```

Viewing the Output

A PDF can be generated with the following

Mac/Linux command prompt:

```
./koch 25 400 575 400 5 | ps2pdf - output.pdf
```

On Windows, navigate to the .exe directory, and enter the command prompt:

```
koch.exe 25 400 575 400 5 | gswin64c -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutput-File=output.pdf -
```

Program Report and Assumptions

Problem Clarification and Assumptions

1. **File Structure**: The program uses Koch.h, Koch.cpp, driver.cpp. The Turtle functionality is implemented as a private nested class (TurtleHelper) within the Koch class.
2. **Euclidean Distance and Angle**: The program calculates the initial length of the level 0 segment using the Euclidean distance formula and the initial angle using atan2 to determine the turtle's starting orientation.
3. **Postscript Output**: The program outputs a Postscript program to stdout, including the %!PS-Adobe-2.0 header, moveto, rlineto commands, and the stroke/showpage finalization.

4. **Memory Safety**: The destructor (~Koch()) is defined to safely deallocate the dynamically created TurtleHelper object, and the copy constructor and copy assignment operator are explicitly deleted to prevent unintended resource duplication and memory leaks.
5. **Error Handling**: The driver.cpp includes robust error detection for:
    - Incorrect number of command-line arguments.
    - Non-numeric input for coordinates or level.
    - Negative values for the curve level.
    - Exceeding a practical maximum of 10 for the curve level.

Potential Problems and Limitations

1. **Floating Point Precision**: Due to the recursive nature and the use of floating-point arithmetic (double), small errors in position and angle calculations may accumulate at very high levels (e.g., $level > 8$).
2. **Performance for High Levels**: The time complexity is $O(4^{level})$. Generating a curve with a level greater than 10 will be extremely time-consuming and will produce a Postscript file exceeding 1 million lines, which may strain system resources or Postscript viewers.
3. **File Dependency**: The implementation relies on the user having all three source files (driver.cpp, Koch.cpp, Koch.h) in the same directory for successful compilation.

References

AI assistance was provided by Claude (Anthropic's Claude Sonnet 4.5) for grammatical review, structural organization, and documentation formatting. All technical content, design decisions, algorithm specifications, and implementation strategies are the original work of Scott Elliott.

Original prompt to Claude: "Review this fractal drawing design for grammatical errors and suggest improvements to document structure.