

Robert Young - 3176240

COSC326

Etude 6 Report

Fractals with Recursion and Iteration

Fractals provide a good visual example of recursion, so when writing a program that creates one, recursion seems to be the best way to go about it. In my RecursiveFractal program, I have a function that draws a square and recursively calls for eight more squares to be drawn around it. This method of drawing the fractal can be understood quite easily, but each square creates a new function call, which could require a lot of overhead in higher fractal generations.

My iterative solution for this etude, IterativeFractal, fills the window with a grid of uniformly spaced squares. Each iteration increases the quantity and decreases the size of the squares, so the final product is made up of many grids drawn on top of each other. This method produces the same image as the recursive one but draws a lot more squares to produce it.

Overall, using recursion would mean more function calls but it is the fractal generation solution that would come naturally to most people. Iteration avoids these function calls but will draw many more squares to the screen, which could cost a lot of resources depending on what the computer must do to draw a pixel to the screen.

One more thing to note about my solution is how it is limited by screen size. The smallest square that can be drawn is one pixel in width and height. The programs are designed to not even attempt to draw squares beyond this point, as it simply wouldn't work. Because of this, I didn't get to try out any fractals beyond the fourth generation with my solution.