Math 198 Report

Rohan Shah
Math 198 - 2010

Abstract

My project will explore fractals and the key math ideas that go behind creating and coloring them. The focus of the project will also be to use math to zoom into a fractal and create a proper algorithm for variable zooming and open editing of the fractal. The key issue with the project will be to figure out how to draw each pixel (Raster graphics, essentially) and create a wonderful picture that can be viewed on the screen without having too much detail/sharpness. The goal of my project will be to work with a pocket program, as the project is focused on 2D creation, and ultimately create a manipulateable fractal function, abilities to zoom in or out, and work with coloring.

1 Focus

I am interested in creating a program that generates Julia and Mandlebrot fractals. This program will involve a certain amount of iterations and a max that will limit the program's iterations from going to infinity for both sets.

Depending on the number of iterations, I plan on coming up with my own coloring system that best represents the fractal and allowing the user to essentially plug in their own numbers for the fractals. My idea for this program is to make the fractal in 2-space. However, since the cube is in 3-D, I am thinking about making this a pseudo-3-space program. Essentially, even if the user is zooming in, the perspective will remain at the same Z axis (i.e., the user is not moving, only the range of the sight is).

As the numbers can get ugly as the user keeps zooming in, I will put a limit on how much the user can zoom in.[1] For creating the actual fractal, I will be using the common fractal equation that I found through some research (Equation 1):

$$z_{n+1} = z_n^2 + c \text{ (Equation 1)}$$

Here, z_{n+1}, z_n and c are all complex numbers with an $a + bi$ root/formula.

What I plan on doing is starting with the x and y axis and plugging in every point (scaled) to calculate the number of iterations for that (x, y) coordinate. Essentially, the fractal.

The number that will change in equation 1 for the Julia Sets will be the c constant. For the Mandlebrot Set, I will be changing the

2 Goals

If all goes well, the program will do the following:

* Calculate the iterations for each point
* Associate the color through the iterations
* Draw a pixel/rectangle for that location
* Calculate zoom in/zoom out scales
* Create a fixed perspective.

[1] For zooming itself, I will be using very small decimal approximations, a reason why I will not be able to zoom in a lot.