Robert
Fractal Project

Mandelbrot and Julia patterns are very fascinating. Being a professional computer artist, I've seen these things before but had yet to realize where they came from or the math behind it. This past summer I went to the Siggraph convention in LA, which showcases a collection of the best work in the industry among other things. One piece was simply an animation of a constant zoom through a fractal with every level showing up yet further detail. The artist took liberties and had the animation culminate with the fractal finally arriving at an 'omega point' which turned out to be a rendering of Arthur C Clarke's head within a sphere, who had been narrating the whole journey. Rather surreal, but more meaningful now that I know what a fractal actually means.

As I looked at fractals and saw how they are constructed, I noticed how the method requires using imaginary numbers to compute the 2d pattern. I imagined what a fractal would look like extended into 3 dimensions and what kind of math would describe it. I originally tried some equations along the line of f(x)=a+bi+cf, where f stands for 'fantasy' number equaling 1 over 0. Eventually I decided this was not the right way to go about it and instead I looked on the internet for '3d fractals'. After spending lots of time agog at the plethora of artwork produced fractally, I came across some bona fide 3d fractals, which are called quaternions. These are essentially 4d complex numbers developed by a mathematician in the 1840s. It is of the form f(x)=a+bi+cj+dk, where:
$i^2 = j^2 = k^2 = -1$
However, even though they are all 'equal' they do not follow the usual rules of multiplicative commutativity.
ij=k jk=i ki=j
ji=-k kj=-i ik=-j
The rules for addition and multiplication are similarly bizarre. As for actually generating the fractals, the usual displayed quaternion Julia is essentially only what would be the black area in a 2d Julia. For simplicities sake, anything that heads to infinity is left out of the image. To get a scope of how complex these patterns are, it takes 16 networked graphics machines 2 seconds to render every iteration.

To add to the abstractness, it turns out that quaternions are actually 4d shapes, like a hypercube, and that each image is only a 3d slice of the fully realized function. So this explains somewhat why rendering out the full Julia set would require such monumental computing power.

After I found the quaternions I also investigated things like Lorenz and Duffing attractors, but they did not seem as remarkable as the 4d Julia sets. They are based on differential equations that are represented as looping strings around a number of foci in 3d space.