Teddy Jackson

Code report project 6

The major basis of my design is limiting the pathways of communication between threads. For the first version, this took the form of a pseudo-master thread, which launches each generation of threads. When the threads are done processing data, they are collected at the pseudo-master thread, at which point the data is rendered. Version two runs all threads once. The processing threads are launched once by the pseudo-master, and from that point on, they communicate through their shared data. This means that the pseudo-master reads each process state through its thread info, once the master confirms the data is ready for publishing, it signals the renderer. Once the renderer is signaled, the data processing starts again, and the data is displayed on screen.

I had some weird behaviours early on, in the form of the initial rendering outrunning the processing threads. Easily fixed. There weren't any test conditions that threw out unexpected behaviour. Obviously large dimensions mean longer load times, but that's the most of it. Other than that it should work fine.