Thread Safe Graphics Library

I worked with Professor Adams as my mentor along with Patrick Crain on a programming library called Thread Safe Graphics Library. With ever increasing power in computers, it becomes increasingly harder to teach students how to effectively use that power. This power is often manifested in multiple processors running on the computer, which can be difficult to understand and use correctly. One way to help students understand parallel abstractions is to create visualizations that show the behavior of processes or threads running at the same time.

Our library simplifies the process of drawing objects on the screen. It also allows for the user to create as many threads, or sub-programs, as they want and for all of those threads to draw to the screen at the same time. Without using our library, trying to draw simultaneously would crash the program, similar to multiple cars trying to fit into the same lane. Our library prevents these crashes from happening, which allows the user to draw in parallel.

This allows for visualizing the parallel computations, which will help students to understand what is behind running programs in parallel. A simple example is drawing an image, which can be easily split into small parts and run in parallel. If the user is using a multi-core computer, then it will run much faster than it would with just one thread.

It was important that our library be portable, meaning that it could run on any operating system. This meant careful choosing what libraries to use in our project and extensive testing on many different computers to make sure that everything worked as expected.

While the library by itself is not able to visualize parallel processing, part of our project was to write examples that use our library to do this visualization. The examples show off the capabilities of our project, as well as provide actual examples that could be used in a classroom. One example is the Mandelbrot set, which at its core is a two-dimensional graph. Each point can be calculated by itself, so the graph can be easily broken up into small parts for computation in parallel.

I personally benefited in many different ways during this project. I had never worked on a large programming project with another person before, so learning how to code along with someone else was an interesting challenge. This was also my first project that used many other libraries to build it, where I had to learn how to use other interfaces while building our own.