Most educators today struggle with teaching computer science students about parallel programming. The concept of a thread, parallel code block, and other key points of parallel programming are often too abstract to explain and so require a visual aid of sorts in order to help students understand them better. However, most visualizations are rarely enough in order to help students understand other concepts of parallel programming; putting a process on a thread, what it means to have livelock/deadlock amongst threads, and many others. Educators could benefit from having a way to visualize these concepts and students could benefit from being able to see the hard abstract concepts in parallel programming. TSGL fills this role in providing visualizations of parallel programming concepts as well as others.

TSGL stands for Thread-Safe Graphics Library. The goal was to provide a 2D thread-safe graphics library. As it stands, there are no real graphics libraries out there that allow multithreading. TSGL fills the void by providing multithreading capabilities, such as having multiple threads being able to draw different parts of the same image, and many more. It is extremely useful to educators who wish to show better visualizations of hard abstract topics in parallel programming.

Various research methods were used as me and Patrick developed this library. We did trials to see if new code that we added worked and that it didn’t break any existing code. We also researched concepts that we did not know about via the internet such as: how to make and write an installer for this library, makefile syntax, and any other concept we ran into that we just couldn’t figure out ourselves. Analysis of problems and researching similar problems in an attempt to find a working solution were also methods that we used in our research.

As it stands, the library is almost complete. We have already published a working version of the code on GitHub, and we are in the process of finishing up the tutorials and code needed in order to have people try it out on their own personal machines. Throughout the summer, we implemented various examples of parallel programming concepts as well as other Computer Science concepts that we decided to include and multithread. These include the bottom-up mergesort algorithm and recursive splitting.

The research dedicated to this project benefitted me immensely. I learned more about various parallel programming topics that I had struggled with prior to this internship. I also learned more about 2D computer graphics, which I had absolutely no experience with before. In general, I gained more experience with various computing concepts that I had no experience with before. These include: bash scripting, libraries and linking them to source code, creating a makefile, interacting with a GitHub repository, command-line arguments in C++, and much more. I gained a lot of experience from this internship and I am very thankful that I was able to work on this project.