**Programming Project Report #1**

Matthew S. Hartstein

010567629

**Problem Statement:**

The main goal of this assignment is to develop a collection of functions that will let users create and display data using the four expected charts (bar, point, line, and area). Normal inputs to my program are as follows: (1) user enters the name of their data file, (2) user enters the name of the type of chart they want to use, (3) command line arguments. As you will be able to tell, I was not able to properly construct the command line arguments due to other projects in other classes; I simply did not have time to fully complete this assignment.

Moving on, the output of my first program is a point chart instead of the required command line arguments that are used as inputs to my second program. As for my second program, the output should be a graph, but I had an **extremely** hard time understanding the problem statement and did not set this up correctly. As for error checking, you will notice how the user must enter a data file within the same directory as the project file. If the file cannot be opened due to a user error, the program will immediately terminate. The same thing goes for the four types of charts the user must select from. If the user doesn’t enter the correct name of one of the four charts, the program will terminate, as expected.

**Design:**

I thought the design decisions were very important throughout this process. I tried several different techniques to accomplish the program that I’ve submitted. Even though it does not fully work as expected, I am proud of the progress that was made. I used a vector as a data structure to save the contents from the data file, along with numerous functions and global variables. Each function contains its own algorithm to perform a specific task. The pros from my design choices mainly involve clarity and organization; I tried to stay as organized as possible throughout the entire process. I would say not fully comprehending the assignment was a problem that led to several cons. First, it took a very long time to understand the basics of OpenGL, even though I was able to create a graph (x and y axis with horizontal lines and vertical tick marks) and display the coordinate values on the chart/graph. Another con is that I was not able to successfully scale this data to be evenly spaced within the region of my x and y regions.

**Implementation:**

My implementation process mainly involved hours-upon-hours of trial-and-error. Once I started feeling more comfortable, I referred to the source code examples on our class website for basic guidance on how to create a simple OpenGL/GLUT application in C++. The only sample code that was used for this assignment was the “sample” file on our course website. This code was basically used as a skeleton to the functions that are called by the various command line arguments. Even though I was not able to fully complete this assignment, over 30 hours were spent implementing, testing, and debugging my source code.

**Testing:**

To test my program, I added error checking mechanisms for both reading the data file, as well as the type of chart the user wants to create. From here, I was able to pass values into my GLUT display function and was able to create the “background” of each of the four graphs, as well as displaying the point coordinates on top of the graph layout. Normal inputs that were used came directly from the data.txt file that was included on our class website. Special cases correlate with the error checking mechanisms to ensure the user correctly inputs his/her data file name, as well as the type of chart he/she wants to create. However, my program does not output the command line arguments required to create the chart, but instead, creates data points on a chart from the values from the data.txt file; this assignment was pretty confusing to be honest! But I am pleased with the amount of work I put into this project and feel much more comfortable with OpenGL/GLUT than I did before this semester started. The screenshots below display a sample input and a sample output from the program that I developed:



1.) Error check to verify the file has been opened

A close up of a sign

Description automatically generated

2.) Error check to verify the type of chart is one of four options

A close up of a keyboard

Description automatically generated

3.) Point chart output with command line arguments used to create chart.

A close up of a logo

Description automatically generated

4.) Point chart that is produced from the command line arguments from #3 above.

A screenshot of a computer

Description automatically generated

5.) Line chart output with command line arguments used to create chart.

A close up of a logo

Description automatically generated

6.) Line chart that is produced from the command line arguments from #5 above.

**Conclusion:**

By now, it is evident that I was not able to successfully complete this assignment, yet, over 30 hours were spent developing the source code that I’ve submitted and I’m happy with the progress that was made. I was able to only create the point and line graphs, along with their proper outputs. However, I was not able to scale this data. One thing that I would do differently next time is to utilize the T.A. office hours for extra help, as well as starting a little sooner to allocate more time to fully complete the next assignment. Lastly, around ~30 hours were spent developing, testing, and debugging this program and I am happy with the results – this was an enjoyable learning experience!