Adam Roth

CSCE-155N

Quinn Lanik

May 3, 2021

Computer Science Final Project

For my final project, I just did the example general project, with the x and y coordinate graph. I did this for a couple of reasons. The main reason I chose this was because I am not the best at writing code. I have tried very hard to understand, but it is just hard for me. I have enjoyed it to a certain extent and definitely learned a lot from this class. I know way more about coding now than I did before this class.

To start my project, I defined the global variable, and created my GIU figure with, at the time, undefined callbacks. There are two callbacks in my main function. The two callbacks were hard for me to write, but the plotting function was the most difficult by far. One of the callbacks is colorSelect. This function allows the user to pick the color of the data that they are plotting. It is nice to have this option when you plot more than one dataset, so that it is easier to compare the two datasets. The other callback is typeSelect. This is very similar to colorSelect, but it lets the user choose between different styles of lines for their data. This is also helpful in analyzing multiple datasets. Both of these callback functions are essentially identical. They both string compare to the radio buttons and if they match, then I set graph.color equal to what it needed to be for each specific case. It was the same idea for both callbacks. At the time of writing this paper, my code works. It plots points and gives an error for unacceptable data. The only thing I have left to workout is the color and line style. I have the radio buttons in for them, and the colors worked before I added the line type function. I always get a green dot/dash line no matter which buttons I select. I am meeting with Quinn on the Monday that this is due to work that small detail out, but for the most part, my code is complete.

The hardest part of this project was the plotStuff function. This function checks to make sure that the data for x and y inputted by the user are the same lengths, and that they are numerical only. If they are not the same length or not all numerical, then a modal error box pops up and tells the user what is wrong with what they put in. If both of these conditions are met, then the function plots the x and y coordinates and uses the two callback functions to decide color and line type for the data that was put in by the user. I am not sure why this function was so hard for me to write, because looking back at it, it all makes sense. I just had a hard time with it at the time.

As an engineering major, it is essential for me to be able to make and analyze graphs. I usually use my calculator for this task, but there is one problem with that… my calculator can only graph functions. This code can take in any set of numbers and plot them, even if they are not a function. If I have some extra time over the summer, I might try to create more functions inside this file to make it so I can type in functions and have functions be plotted. Having a feature like that, where inputs do not have to be in terms of y like they do on most calculators, would be extremely helpful for visualizing functions that are given to me in higher level math courses that I can not necessarily visualize in my head. I wish I would have taken this class last semester while I was in Calculus III because a lot of my problems in that class came with not being able to visualize what I was doing or learning about. Having this would have been extremely helpful, but oh well. I plan to use this graphing code often in my educational career and maybe even into my professional career.