Jack Doherty

CSCE 155N

Final Project Report

5/3/21

Final Project Report

My project was the example project mentioned in the final project folder. I may have taken the time to brainstorm and design a different, but I simply did not have the time, and decided to do the example. Because of this, I did not email any LA asking for authorization, as this was the example put in the instructions for the folder. However, I do not think that this project was any easier than anything I could have thought up on my own. This was a very difficult process overall, as I had limited experience with several parts of the project.

The design process was nothing out of the ordinary, but I did have to educate myself thoroughly on the GUI designing. I started by creating the basic figure, with only a plot, two input boxes, and the plot button. After setting up the first text boxes and buttons, I had to figure out a way for the code to decipher what was written in the input box. After several different methods of trial and error, I ended up using a method involving regular expressions, because other methods of testing whether the input was the correct format were not working with my code. Checking to see if the input was a matrix did not work, as phrases count as a 1x1 matrix, and checking that all inputs were numeric would accept inputs that did not have brackets, which my code needed in the input. I used a two-step regular expression that first made sure the input was a list of numbers, and then divided the failures into two sections; one where the input was a number list, just missing the brackets, and every other combination of symbols that the code would not accept. These then returned error codes instructing the used to add brackets or use numbers only, respectively. Once the inputs were proven valid, the code made sure both arrays were of equal length. If not, an error was displayed which mentioned the issue. If they were of equal length (and not empty), the values were plotted onto the graph, with the icon shape, color, and line style set as variables that could be manipulated later.

At this point, the bulk of the calculations were done, but not the bulk of the code. I spent much longer than I thought I would be learning how to implement the button groups and radio buttons. I thought it would be like the other UI inputs, but I struggled getting them implemented into the code. They also require a significant number of lines of UI setup, and it was annoying searching through the code for the slight issues that were causing my errors. Once the were setup, I mapped each button option to its own function that changed the look of the graph. This was probably the least efficient way to go about this, and there are a lot of functions in the code because of it, but I could not find any way to condense it, and it still works. I also added a title editor that, for the graph. For some reason, whether the title has been edited or not, the title is deleted every time a new plot is formed. I put a significant amount of time into attempting to remedy this, but nothing I changed fixed the issue.

Easily the most difficult part of this assignment was finding a way to manipulate the inputs to make them all similar enough format to be ran through the code. I spend several hours attempting different methods to achieve this, with many, many retries and even a couple fresh starts, where I deleted everything and redid it from scratch. I had a lot of issues with variables, especially ones inside nested functions that needed to be used by the main function. I eventually tried assigning the variable blank values before the functions were executed, and it ended up working. There were several other areas that tripped me up pretty hard, and I spent a good few hours working out several other bugs scattered around my code, using trial and error until something finally worked.

GUIs are very useful for getting anybody to be able to use the code you have written in a way that is easy for them to understand. Every single type of technology used today, from computers to smartphones, all contains a GUI that is intended to be as streamlined and easy to use as possible. While I would probably never make something that sophisticated, I may end up using MATLAB to automate something in the future, and make it easier for myself and others to use by weaving the code into a GUI. While it is more difficult to create in the first place, a GUI makes up for its difficulty by being much, much easier to use once the code is fully operational.

I can say with some degree of certainty that the code that I wrote is not very concise and is probably a needlessly convoluted way of designing it, but the basic components work well, and I could not find any combination of symbols that did not have an intended outcome, error or not. There were no scenarios that I could find where the code crashed or became unresponsive. Overall, this was a difficult but enjoyable project, and I can say that I have learned a lot about programming both on this project and through the entire semester.