

Luke Bond, Chad Gratton

5/5/2021

CSCE155N

Final Project Report

As a pair, we decided that we were going to use the default option that was provided. The first thing we did was create our master function in order to make this a divide and conquer type of solution. This function contained our global variable that was called gui(yes very original). The function then had all of our edit boxes, buttons, radio buttons, and plot and where they would be located. This function as it was used in the figure would call other functions to run to plot the data. This was the easiest part of the entire code to write, as it was a simple formula to follow to get all of the elements to appear on the figure. We both worked together to get the edit boxes, plot, radio buttons, and everything else to be in the right position before we decided to make any of the callback functions.

The first callback function that we made was the reset button. This was also the easiest callback function that we had to make. We essentially just copy and pasted the main function into the reset function and everytime the function was called, it would essentially just run the function from the start. Additionally, since it took such little time to make this function, we decided to have a little fun with our code. We made the reset button as big as the rest of the screen and made the font size as big as we could to fill the button. We thought it was pretty funny and we decided to keep the code for our final code.

The second function that we were able to make was the plot data function. This was our major function that would end up doing the most work and also taking up the most time. We first made it so that all of the edit boxes, buttons, and radio buttons would go away once the plot button was pushed. This was in order for the massive reset button to be made. We then used `str2num` to make the x and y edit boxes convert the string into an array that matlab could interpret. This was slightly confusing at first but it ended up being a very easy part of the function that we made. We then defined more variables that we would end up using in the scatter function to change the shape and color on the plot. We also created an error message that would pop up if the data was not correctly formatted by the user. At the very end of this function is where the magic happens, because this is where our scatter function, x label, y label, and title were all located. This was tricky at first to implement, but it was very rewarding once it was done.

The third set of functions that we made were the radio buttons. These were especially tedious to make. We first made the radio button callback function first which was very simple in itself. We defined the variables `type` and `type2` as the selected color and shape from the radio buttons in the figure. We then made another function that was a series of if loops that used string compare to essentially tell the main function what the selected color and shape was in the two sets of radio buttons. This would then be stored and used in the scatter function in the plot data callback function.

In conclusion, this project turned out to be actually very useful. We both now have a way to plot data and make graphs for other classes like calculus or chemistry. This is all thanks to GUIs. GUIs are a very neat way to have a common variable

throughout that is easily accessible and can create some very useful graphs and other algorithms. If needed in the future, we can both turn to GUIs to try and automate something like we did in this project. We also thought of something that we could automate in the future while working on this project. A deck of cards has 52 cards in it and we were thinking that we could come up with a way to “shuffle” a deck of cards and everytime you clicked the button to deal a card, you would deal one card at a time until all 52 cards had been dealt. Another idea that we came up with was to create a GUI that would calculate the tip a person should leave based on how well the job was done. For example, if a waiter was rude that would subtract 5% from the total tip. If they brought water to the table or refilled constantly that would be another 5% added on to the tip. Then the final output would give you what a reasonable tip amount should be.