Franklin Thomas

Final Project Report

I chose the default option project since I would be working alone. The approach I took was first laying out a skeleton of sorts for the GUIs. I figured it would be easiest to fully complete my interface then work based on that to create my other functions. Lab 10 was a valuable resource for referral. I borrowed the global function idea from it along with the layout for my display labels. Creating the edit boxes, the reset button and the plot button was easy enough. The challenge was formatting the graph and the radio button groups in a way that made sense and fit on the interface page. I had a problem early on with elements overlapping but that was quickly fixed with some position tweaking. I had a similar formatting issue with the radio buttons within the button groups. That was also fixed through several trial and error attempts with positioning.

The only issue I had with creating the radio buttons was making sure only one of each could be selected at any given time. I went through several ideas, none which worked. I eventually consulted the internet and found some matlab documentation that fixed my problem quickly. A short function could be inserted within my main function and could work for both my button groups at the same time. This said function removes the previous choice and only displays the current choice within the group. Now only one option in each of the button groups could be selected at any given time.

Once I had done all of this, I ran my code several times and edited positioning until I found a layout that I liked. Once I had complete skeleton of my interface, I could move on to the inner functions.

The actual callback and moving parts of the function were more difficult. My idea was to convert each of the inputs in the edit boxes into arrays. From there, the function could analyze them and make sure they were valid inputs (no letters or special characters, both inputs were the same length, etc.). The process would be triggered by the plot button, which would act as the signal for my function to start. I planned to use the numel function to count the number of elements in the input arrays. For simplicity’s sake, having the inputs need brackets and be formatted as an array would be easiest. That would make for less if statements and/or loops needed to sort out possible errors. From there, an if statement would compare if all the requirements were met and allow the function to continue and plot. The final plot would take in both x and y inputs, the color selected from the color button group, and the line style selected from the line button group. The reset button, to make things simple, would only reset the graph. It is not necessary to reset the edit bokes or the radio buttons because they need the plot button to activate. Simply telling the graph to re-plot (0,0) would be the simplest option.

Once I started actually implementing these ideas, it proved to be a lot more difficult than expected. Various syntax errors and callback errors required some changes to be made to my skeleton function. I tried switching to a non-global variable function set, but that set me back even further. I eventually returned to my global variable function and consulted lab 10 for extra referral information on callback functions and global variables. Overall, the design and creating the interface was the easy part. It was the callbacks and all the problems and situations within those callbacks that made this project difficult.

A GUI would most likely be used by me to solve formulas. An easily resettable interface with input options for variables would make solving certain physics problems, for example, so much easier. If I program the entire formula into a code and just require input for variables, solving said problems would be faster. A GUI could also be good for presenting data. Using live graphs that respond to inputs would help with presenting information.