CSCN 155N

Quinn Lanik

5/3/21

Jack Sather and Max Gengler

This project linked most everything we learned this semester and showed us the cool stuff you can do with it. This project helped us to see the uses of our coding in an interactive way.

Our code involves a large amount of GUI operators and modifiers. Our global is defined as ‘goo’ and the global proved to be one of the most convenient things we have learned this semester. The ability to write “global goo,” and have access to all global variables provided an efficient way to write the code. This project challenged our knowledge in a couple different ways. The challenges we ran into were getting callback functions to interact in a button group, testing the x and y input values to determine if they should plot or output a modal error message. Lastly, we hit a wall when trying to get the button group functions to affect the plot.

            In the button groups we ran into the issue of the buttons within the parent group, as we could not interact with them inside of the figure. This was easily solved by inserting the parent within the buttons when writing the uicontrol. If the parent is not defined in the components, the button will not be in the group. Also, when giving the buttons within the group functions we did not have a “SelectionChangedfcn” CallBack. This is necessary because it makes it so the buttons can have duties and change per each button. Because we used radio buttons, a “SelectionChangedfcn” callback was necessary.

            When testing x and y values to determine a valid input, we tried regular expressions, but they had to be character arrays and we had a list of scalars. So, we opted to write an if statement that tested the length of the inputs to determine if it were to be plotted or not. If not, the plot would give a modal error. The modal error was something we had never really worked with before. It seems to be a very useful message to the user of the GUI because it will not display an error in the command window. Rather than the error, the message will pop up on the figure and give the user an idea of what the problem was when inputting their information. Writing the callback function to plot our scatter plot was our first step of many, but once we had that function done; we felt very optimistic about the rest of our code.

            After writing our code, we have all the components of the scatter plot, 2 button groups, a plot button, a reset button, and x y input boxes. When our values are inputted into the boxes, they are numbers separated by spaces ex. 1 2 3 4. Our button groups can change the appearance of the graph and our reset button resets the text edit boxes when pressed once, and the plot when pressed a second time. Although at first we wanted to have the reset button reset everything at once, we found that clearing them separately was sort of a perk to the figure. The code functions with no errors to our knowledge, which is something we were very proud of. In order to ensure that, we would test every part of our code to see if there were any inputs that would prompt an error. This took a lot of time, but paid off in the end.

Looking forward, we believe the guide function through the command window could be very useful when creating functions for calculus. We played around with guide and found some cool things that make it a lot easier for us to establish our uicontrols. One of the calculus functions we thought could be useful is integration by parts. Integration by parts could be very done easily and quite efficiently through gui. There also aren’t a lot of recources for that type of process, which is why people who do not know how to code could benefit from the gui. We are excited for our future with gui’s and felt we grew a lot during this project.

We did not have an LA approve our project because we used the option in the final project pdf.