For this project, I chose to go with the default option of creating a GUI with the purpose graphing a bunch of data. My main goal when designing this GUI was to make it simplistic and easy to use. I did this by moving the edit boxes in their individual areas with labels and a puah button to enter them. The uibuttongroup looks clean and the buttons display their respective properties right next to them. The title of the plot fits nicely above the graph. Everything just kind of lines up and has around the same proportions to each other. I also added error messages so that if something was entered wrong, it was easy for the user to know what went wrong. Another objective I had for the design of this project was to make sure that everything does its job and that it all works well. I think I've ironed out a lot of the chinks in the code, such that most of the time the data should go through and that everything should work as intended.

The coding process for this project was at times really confusing and during others, really irritating. To make it short, I had to learn a ton about various functions that I worked with and various logical puzzles to go through. Callbacks, edit boxes, and dealing with the strings from the edit boxes were all new to me and not from lab 10, so I had a lot to work with. If there was one thing that I really got stuck on though, it had to be the uibuttongroup. Apart from the format of the coding of uibuttongroup functions, I had a really hard time implementing some of the ideas I wanted to do with them. One of things at the top of my list for the uibuttongroups was a pushbutton that closed them on the figure, but I just couldn't find a way to implement it. A couple things I tried to do was make the group invisible, and just adding a huge text box over them, but it didn't really work out with the reset button and just clicking randomly on the figure.

On the other hand with the coding process, one of the things I kind of just didn't like about it was copy and pasting various lines from the top of the function to update them in a line below. I think that it makes my code look really cluttered, and it makes the small if/else statements look weird. I found condensing my code in general to be really tough as if there was a line of code I thought I could do better on or wasn't needed, when I replaced it, my new code just wouldn't work like I intended it to. I think that condensing my code would have really helped out with making the program run faster and making the code look better. Another thing I really struggled with were logical puzzles that trying to get the error messages to work caused me. Having to make them work in a way that would allow the user less trouble was really irritating. With the limit boxes for instance, I made it so that you could do only the x limit and leave the y limit blank, and it caused a bunch of headaches.

Other than just simplifying and condensing the code, I wish that I could have added more functionality to the GUI. I know there isn't too much one can do with a graph and inputting data, but perhaps adding a menu with way more options, or adding a way to plot the graph in 3d would have made it better. As far as using GUI's in the future, I think they can be used in a lot of useful ways like programming calculators or playing turn-based games like Pokémon or chess. It makes visualizing and just adding data a whole lot easier than declaring a variable in the command window and running the program and such. The problem is that it just takes so much work to create GUI's, especially with the small stuff like the positions of every box and retrieving the information that was entered into them. Maybe I could get really fast at creating these things, but it takes so much effort even for a GUI like plotting data that I can't see myself creating these things for many purposes. Unless I'd have a specific purpose that creating a GUI would make so much easier, it's a hard no. I think that once the big part of the GUI is done, it's quite a bit of fun just tinkering around with some of the small aspects of the GUI and adding small little things to improve the GUI. Although it's probably not the case, I can see why some software developers may like to create things like GUI's, especially if they're actually good at doing it.