**Title:** CSCE 155N MATLAB Final Project

**Owner:** Zachary Bravo

**LA approval:** Caitlyn Allen Using Default Option

**Introduction:**

In CSCE 155N, we were exposed to a variety of STEM related coding projects that were developed in MATLAB. One of the helpful MATLAB tools are the set of GUI (Graphical User Interface) tools. The GUI provides a visual representation of communication presented to the user to allow easy interaction with the machine. MATLAB provides graphical representations such as buttons, edit boxes and icons. Communication can be performed by interacting with these icons rather than using text-based command-line operations.

**Project requirements**:

Project requirements for the Default Option are stated on the Final Project instruction sheet.

**Scope:**

By using the MATLAB GUI tools, a user can quickly generate a plot and interactively modify it without having to understand the plot commands and how to implement in MATLAB. The GUI should behave in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. The user would not need to redo MATLAB commands or a function many times to get the output they want.

**Problem statement:**

Automate some tasks related to my major or hobby using a MATLAB GUI. The task should not be trivial in nature. The GUI will need 3 or more different styles of UI Control elements and two or more call backs. The finished project should provide a user friendly tool that is easy to use and understand.

**Design and Coding Process:**

The first thing I did was to review the project requirements from the Final Project instruction sheets to understand each requirement. I then wrote down each requirement and what is expected for each. This was used as a checklist to validate all requirements have been implemented in the code and that the program generates the expected outputs.

1. Start simple
2. Determine what is the first item that needs to be created
3. Add one function or capability or line at a time
4. Add one GUI option at a time
5. Test/debug after each added item by displaying or printing.

GUI design:

1. Rough out how it should look
2. Analysis – define key functions, interfaces, and MATLAB functions which can support them.
3. For each GUI element
   1. decide what type of GUI element can support the capability
   2. look at options for that GUI element to decide what to use
   3. Add the control item to the code
   4. Add the callback function to the code
   5. Test good input
   6. Test bad input
4. While writing code
   1. Use descriptive names for functions and variables to make it understandable. If this is done, the program should not require as many comments
   2. Use whitespace to make the code more readable
   3. Add comments to explain the approach
   4. Make sure function parameters are understandable
   5. For GUI entries, make sure the user understands what inputs are allowed
   6. Notify the user if inputs are invalid
5. When adding new capabilities
   1. Check documentation on each function
   2. Review online examples
6. Have someone else run the program to see if they run into issues and fix them

**Difficulties and Troubleshooting:**

1. Determining starting point – I had trouble deciding whether to start with the plot or the GUI controls.
2. Determine how to include a plot and GUI controls in the same figure – I had to look up examples on how to do this.
3. Determine how to update the plot items in the callbacks without erasing the plot – I had to figure out how to reference the same plot elements and update them.
4. I had some difficulties in sizing, determining where to put the user controls and making sure there was enough space for everything.
5. Restarting from scratch 10 separate times was hard, some of the capabilities I was adding were not compatible, so I had to try different approaches.
6. Uifigure and figure are not the same thing and have different functions to work with them. Elements like radio button are implemented differently
7. Trying to use xlimit(ylimit) handle had issues, using a handle would create a new plot and break everything.
8. It’s painful to move stuff around together, using offsets would have helped.
9. It took many hours to complete and to make it function properly, I think the end result was worth it.

**How can I use GUIs in the future:**

I can use GUIs to manipulate, plot and interactively update lab data sets. GUIs can be used to take data set and perform calculations to create new data sets. The great thing about a GUI is you create a simple user interface when you must repeat a task on large data sets that would take longer if it would have to be performed manually. These are opportunities that I will use in the future to make tasks and projects easier in my upcoming classes. I could use MATLAB and GUIs outside of school to look at how much I earn, how much I spend and where I spend money. MATLAB and GUIs could be useful in potential jobs in the future.

**Conclusion:**

This project has been successfully completed by having established a user-friendly interface with the help of MATLAB GUI tools. It consisted of an interactive plot created and modified by using various GUI user controls. At the same time, there is some room for improvement in the future. It can be possible to make it more user friendly by adding more variety of inputs and functions to it. For most cases, it would make sense to have a user enter filenames and read the input data from files rather than having the user enter data points. It would be nice if we could introduce user-defined functions to be plotted, then the user would have the ultimate flexibility to input and analyze whatever data they need to accomplish their tasks. The Final Project has given me the ability to learn more about what I can do with MATLAB, what it takes to create a user-friendly GUI and insight into how I can use MATLAB in the future.