Aussia Stander and Helen Philbrick

CSCE 155N Sec 210

Quinn Lanik

May 3, 2021

**Final Project Report**

The first step of this project was to come up with an idea that would meet all the requirements. It needed to have three different styles of UI control elements, at least two callbacks, and was to be non-trivial. A unit-converter program was sketched and proposed to Learning Assistant Laurel Hilger. One would run the program, type in a number, select the units they were starting with, select the units wanted, then click the “convert” button. A new number would be displayed in the desired output unit equal to that of the input unit. If there was a non-numerical input, or if two units were selected which did not measure the same thing (for example, units of energy and units of temperature), an error message would display. Learning Assistant Laurel Hilger approved of this program.

The next step was to write the program. The code to initialize the GUI was first, since it was where the interactive screen would come from. The first issue with the GUI was the position of the interactive units -- it was different on each screen. This was solved by adding normalized units to the GUI code. However, this did not solve everything with the position issues. The radio buttons seemed to be fine, along with the edit-text box. On the other hand, the pushbutton and the text box seemed to follow a different rule for placement. This was solved by guessing random numbers for the position array and stopping when the GUI item was in the desired location. Along with the GUI coding, there was an issue with the radio buttons. The grouping of the radio buttons was tedious, but not impossible. The issue was solved by adding a bit about the selection changed function. The positioning of the radio button group was, surprisingly enough, not as challenging as the text box was. The callback functions were not much of a problem at this point, as they had not been finished.

After the main GUI figure was written, the callbacks and other technicalities of the code were written. A few new functions were typed up for callbacks, one for the radiobutton groups and one for the pushbutton, and new errors arose. First, it would not display the answer and would send error messages before it was supposed to. There were also a few display issues. This was a relatively easy fix, done by making the logical statements for checking if the input was valid in the pushbutton callback and making the display string in the radiobutton callback. The answer box was not positioned nicely, so the position was manipulated until it was in a good spot. MATLAB disliked using the str2num function to change a string to a number, so it was changed to be a string to double using the str2double function. There was recursive call problem as well which made the code inefficient. A recursive call to the radioSelect function was added to the dispConversion function because if no new radiobuttons were selected, the answer text would not change as no new conversion would be performed. However, this caused the calculation to be run twice regardless of whether the radioSelect function had already performed the calculation or not. Multiple if statements and even adding an additional callback to radioSelect in the pushbutton were attempted to remedy the issue, but they did not work. After consulting with Quinn Lanik, this issue was fixed by adding a callback to radioSelect to the textbox so the calculations ran whenever the user typed a number into the editbox, thus eliminating the need for a recursive call. Once the code was bug-free, its functions were tested to make sure it could perform the proper conversions when correct units and inputs were used and displayed an error message when an improper input or an incorrect conversion was attempted. The display strings were also tested so that it displayed the word degrees in front of the units Fahrenheit and Celsius.

In the future, GUIs could be used to help non-programmers complete tasks requiring complicated code through the GUI interface with ease. It could also be used to have fun with games such as tic-tac-toe or battleship. Additionally, GUIs could be used for multitudinous reasons and applications in engineering and STEM careers in the future. Some applications include plotting data, performing mathematical computations, analyzing large data sets, and image manipulation implemented into the GUI interface. If coded properly, GUIs can perform multiple functions without needing to understand or manipulate the code behind them. Gaining additional practice with MATLAB and GUI coding would be one of the best ways to use GUIs in the future.