For my final project, I have decided to make a pitch counter a coach my use to keep track of the number of pitches his/her pitcher is at currently. This would be the sole application of the user interface, but it is more advanced than what one would find in a sporting goods store. This was approved by Quinn Lanek via email and was requested to have the ability to graph the information it was given.

I, initially, had little to no problems with coding the pitch counter. I have taken book and charted pitchers in the past; giving me the logic to complete the assignment. If there were any problems in the beginning stages, it was dealing with the amount of focus used to write uicontrols. Compared to previous functions written in class, uicontrols takes a wealth of information and requires a lengthy trial-and-error process because I had a vision for the interface, while the computer is incapable, forcing me to tell it every detail. The trial-and-error process, mainly, refers to the placement of words, numbers, and buttons and how, at first, it is plugging numbers in to see if the placement looks good. After that, other placements can be made off the first one making the process easier.

Eventually, I stumbled onto a problem with writing my first callback function. The function “addPitch” is meant to threefold. Firstly, it increments the total pitch counter by one each time a button is pressed. Secondly, depending on the button pressed, a dialog box with two choices is displayed asking if the pitch was a “ball or strike?”. Depending on that answer, either the strikes for that pitch and the total strikes would increment by one or the balls would increment by one. Finally, each row of strikes and balls would be added together to make the subtotal for each pitch. This led to my first problem, I could only add one strike and one ball to each pitch. This was easy fix: I had to save the variable I was incrementing by, so I could add one each time to get a larger number. The next problem was that all the pitches would follow the same number as the total pitch instead of increasing by one. For example, if pitch #1 was strike there would be a 1 displayed under the strike column, but if pitch #3 was strike after pitch #1 then it would display a 2 under the strike column for pitch #3. To remedy this problem, I added an index variable to my function that made sure to call a specific pitch instead of applying it to all pitches.

A final problem I had was graphing the data inputted into the pitch counter. I imagined this displayed bar graph with groups of bars denoting the type of pitch. For some reason, which I never figured out, the bar function would not take in the values of each pitch and would display a default graph of preset values. After some research, I took out the subtotal variable and used the bar function’s stacked ability to represent a subtotal. This solved problem and is more efficient.

I greatly enjoyed making this UI because I can see how it applies to websites, video games, and computer usage in general. What comes to mind immediately is dialog boxes for talking characters in video games and responses from myself or non-player characters. I can only imagine the length and breadth of the code needed for 20-hour long game heavy in story and character development.

In the future, I can see myself using them like a graphing function that is more visual and applicable than only writing code. Especially if others would like to use it to implement real time data and then be able to graph its trend. I can also see it as a way to do difficult or time-consuming math properties like partial fraction decomposition. This takes some time to do by hand is not always perfect, but I could write a UI that takes in whatever needs to be decomposed in a textbox and returns the correct decomposition, so I can then solve the differential equation. I suppose, the same could be said about differential equations.

I greatly enjoyed this project and the difficulties I had along the way. This gives me an appreciation for programmers who optimize user interfaces and the sheer time they spend making simple things even easier to use. Though, computer programming is not my favorite thing to do, this project, and this class, has definitely shown me qualities of my future that I will enjoy.