For my final project, I chose to automate the scoring process of trapshooting. Trapshooting has been a hobby of mine since freshman year of high school. In trapshooting, the scorekeepers keep score by using a piece of paper and pencil. Oftentimes, some people's handwriting can be hard to read, and I wanted to make a GUI that would automate this process. If you are not familiar with trapshooting, here is a quick summary. Each round consists of 25 targets. A single target is thrown when the shooter calls pull. If the shooter breaks the target in any way, it is recorded as a hit. If not, it is recorded as a miss. For seasoned professionals, a miss is a rare sight, but for the sake of this project, we are going to assume that hitting 20 targets out of 25 will be the standard. I would also like to begin by thanking Laurel Hilger, who approved my topic, and also Trevor Thomazin, who helped me in the coding process.

By using my GUI, the user is able to record hits and misses. It also keeps track of how many targets the shooter has left out of 25. The user has a few different options: they can either start a practice round or a competitive round by selecting one of two radio buttons. Once an option is selected, it will alert the user as to which kind of round they will be keeping score on. Once the type of round is selected, the user can now start recording their round. This is very easy. For every target "thrown", the user will click either the button titled "hit" or the button titled "miss". With each click of the buttons, a target is added to the corresponding list, either "Hits" or "Misses", and subtracted from the amount of targets remaining. Once the round is over, and all 25 targets have been thrown, a message is displayed depending on how the shooter did. If the shooter hit 20 or more of their targets, a message will display reading, "Great Job!". However, if the shooter hits less than 20, the program will display a message reading, "Keep Practicing!". Finally, the user can choose to either quit the program, or start a new round. If they decide to start a new round, all of the lists will be reset. "Hits" will return to zero, "Misses" will return to zero, and "Targets Remaining" will return to 25. The user can play as many rounds as they like, there is no limit.

This class is the first experience I have had with any sort of coding language, so it was a difficult but fun project to do. At first, it was a little overwhelming, especially with finals right around the corner, but once I had my topic approved, the ideas for my GUI began flowing. I first began by identifying the global variable “pull”. Next, I created the titles, “Hits”, “Misses”, and “Targets Remaining”, and also the corresponding counters for each list. Once the lists were created, I needed to make two buttons, one for hits and one for misses, that each had a callback function within them. Both callback functions essentially did the same thing: add a target to the corresponding list, and subtract a target from the “Targets Remaining”. I am sure that there was a way to combine these two functions, however, I could not figure it out. Once I had the main part of my code finished, I wanted to add an option that would allow the user to select which type of round they wanted to play. Trevor Thomazin showed me how to make a button group as well as some other things that it could do. I made the button group and included two radio buttons for the user to select from. Depending on which button was selected, a message would appear. Finally, I wanted the user to be able to play multiple rounds without closing the GUI and running the function again. I got the idea of a reset button from looking at connect four code provided in class. If the reset button was clicked, the callback function contained within the code would reset the scoreboard.

While writing my code, I did run into some issues. First, as I stated earlier, I could not figure out how to combine the “addHit” and the “addMiss” call back functions. Although I couldn’t combine these functions, I think that it worked out for the better. With the two separate functions, whoever is using my code is able to see exactly what each function does. I also had trouble creating a button group, but thankfully Trevor helped me out. If I had the opportunity to redo this project, I would want to display the user’s score out of 25. I tried to do that for this project, but with finals quickly approaching, I could not figure it out in time.

As a future civil engineer, GUIs are going to be very helpful in my career. Not only can they do fun things like trapshooting or other games, they can also plot different kinds of data. In

class, we plotted weather data, but GUIs can cover a wide variety of topics. The only limit to a GUI is the creator's imagination.