Testing Documentation

For testing, our team did solely manual testing, checking each part of our code before moving on. Because of the GUI and format of our program, it was easier to test the code by hand instead of making scripts or bots.

Inside the '*test'* folder, we had our test input (temp_input.csv) which we used to import example lineups. We optimized the lineup in our '*lineups.py'* file to verify the optimizer was optimizing the lineups correctly. For the header buttons File, CSV, Help, after they were written we ran the commands and made sure the correct display appeared. For the variables, we would try different inputs and check the results when they were recorded in our configurations.txt file. From this, we added the checks to make sure only float numbers were added and numbers surpassing a several 9's (example: 999999) would not work as no lineup would need that many. This would help cut down on users breaking the program by adding unexceptable values. For the Import and Export CSV buttons, we tried many different files to see how they would work, and compared the created files and changed variables to the expected results. The optimizer was tested by itself (mentioned above in 'lineups.py') and in the GUI, and besides the known issues the optimizer works properly. We compared the expected output to our actual output to verify it was correct. Due to the intricate design of our optimizer code and GUI portion, when testing the optimizer we needed to add a 'Stage Changes' button in order to ensure all changes are added and ready before the optimization.

At the end of our testing, we had found several problems which we have documented in our known problems report.