**Programming Foundations 1 – Project 3 Report**

Name: Riley Nadwodny

Date: 10/9/19

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement**

The objective of this programming project was to create and output color images of five different country flags from around the world in C++, with a bonus of outputting the flags to ppm files. The purpose was to give us experience using iterative statements in C++, which was composed primarily of nested for loops and if statements. The inputs needed for this program required the user to select one of five supported country flags listed out to the user, as well as the length and height, columns and rows respectively, and use those inputs to print out the corresponding flag with the specified dimensions. In regards to error handling, the range for the length and height was 100-1000 pixels, so if the user input a number less than 100, their input was set to 100 instead. If the user input a value greater than 1000, their input was set to 1000 instead to stay within the range.

**Design**

For outputting the flags, I decided to make them void functions for the purposes of this project. So each flag, such as the Thailand flag, had a corresponding void function, void flag\_Thailand, receives the number of rows and columns from the user as const int’s. It opens a new file using ofstream, outputs the required headers for the .ppm file, and prints out the flag to the ppm file for the user to view. I really liked this design decision as I could work with and debug each flag individually outside of the main function, as it was easier to read and work with the code, rather than looking at multiple nested loops within the main statement. Each function had different algorithms from one another, but some had just slightly different algorithms from another one flag.

**Implementation**

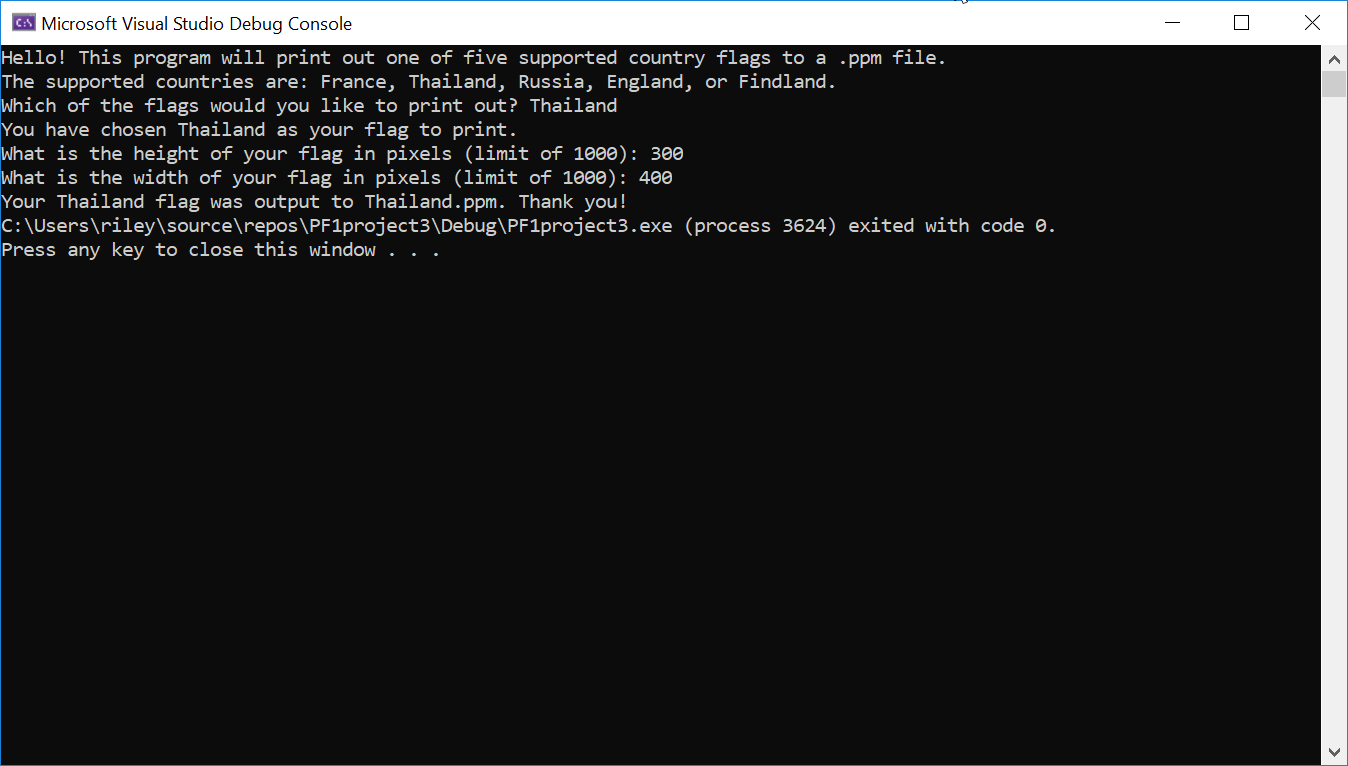
Implementing this project took almost the full two weeks allotted for this project. I began by declaring the initial strings and variables, and receiving inputs from the user. Then came error checking the users inputs, and figuring out how to correct their inputs to keep them within the range, as well as making sure that they input a valid flag for the program. For creating the flags, I began with some sample code from Gauch’s website, the draw3.cpp file. I used the drawbox function as the basis for most of my flags, as well as code for drawing a plus sign that we did in class, and modified the code from there. I had to derive equations to use for the flags and decide what colors to print in given areas of the flags.

My development timeline, as mentioned, was about two full weeks, or 13 days to be exact. The basics of the program were done in the first week so that I could spend some time figuring out how to program the flag functions, and derive the equations needed for those. The England and Finland flags only took an hour for the two flags, as I had most of the code for those from the in-class example we did of drawing a plus. France and Russia actually took me the longest time to code, as the outputs were not what I desired for the longest time, and then Thailand took about three hours to figure out.

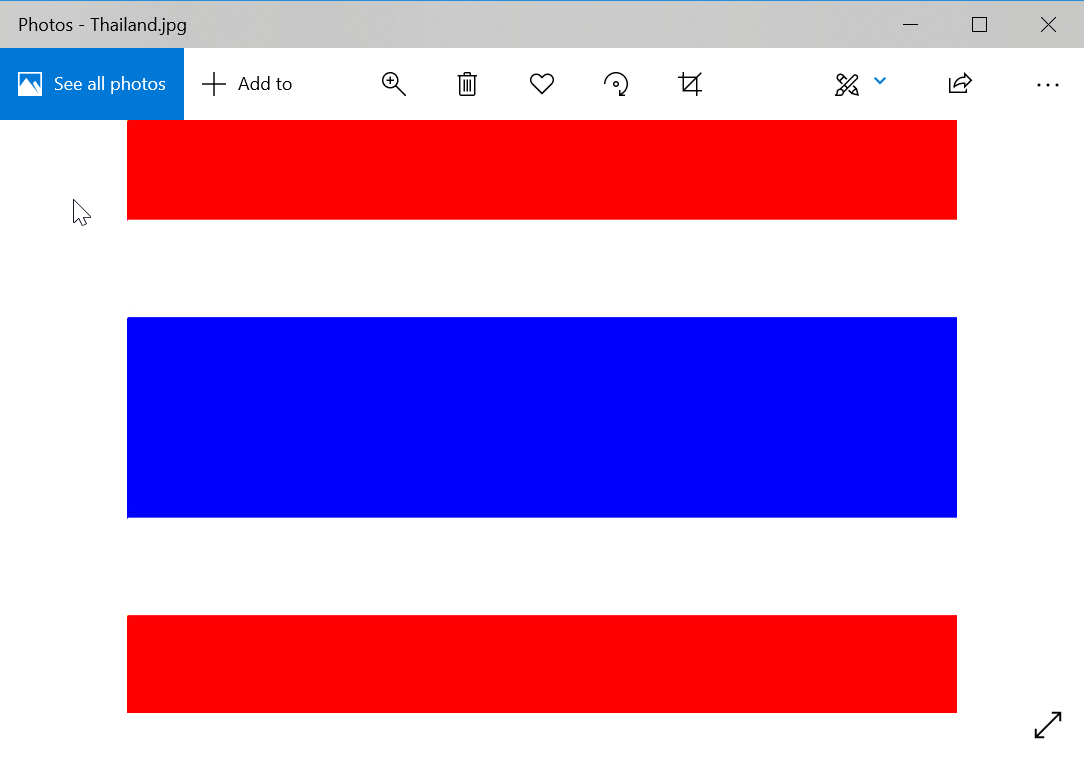
**Testing**

For testing purposes, I worked on coding one flag at a time and debugging that flag before moving on to the next flag. For testing the flag outputs, I used a 4:3 ratio and would use 300 for my rows and 400 for my columns so that the flag would be a rectangle, as country flags are. For a few days, not everything worked as expected. The France and Russia flags would only output two colors for the longest time, omitting the color in the middle for a decent amount of time until I finally figured out the right equation for it.

**Sample input**



**Sample Output**



**Conclusion**

Overall, the assignment was ultimately a success, albeit taking 13 days to complete. The biggest challenge I faced was figuring out the right equations for each flag in order to get the program to print out the correct colors in the right areas. It was definitely a lot of trial and error and a bit of frustration, but when the flags were finally correct, it was the biggest feeling of success and very rewarding. Next time, I would spend more time planning before working on the flags and figuring out the right equations to get the program to print the colors I want in the correct areas. I would definitely do the flags as functions again though, as that made working with individual flags a lot easier to read and comprehend rather than looking at a mess of nested loops inside of the main statement.