Tyler Gersbach, Alexis Benter, Elana Bobelis,

Patrick Bradley, Ryan Lieblein

**COP 3503 Project: Report**

This portion of our project records the variety of lessons learned in said project, from new C++ terminology to wider reaching concepts, like object orientation. These lessons will be separated into sections.

**Lessons learned, Professional:**

Delegation: To properly finish our project, each function and class needed to be delegated to a specific few people, allowing rapid development of the program. We coded one class a week, delegating that class’s functions to specific people. After the major classes were done, we each suggested different ways to improve the game, and that person would code that portion.

Planning and flexibility: Much can be done over the web in our modern age, but meeting in person is often necessary to make significant progress on work. To achieve this, our group had to plan specific times and locations to design functions and actually code. This required a high amount of communication, and consideration for other member’s obligations.

Utilization of third party resources: In order to effectively merge different versions of the game together, we had to use third party, cooperative programs. We used git; software which is heavily used in professional settings. Git enables the merging of different forms of a program, using an origin-master system. In order to properly use git, each member was required to do their own research in order to use the software properly.

**Lessons learned, Design and Coding:**

Encapsulation: Rather than sloppily putting every function in the same file, we separated each job into specific functions. Each class has its own getter and setter functions, reinforcing encapsulation.

Object Orientation: Our program is specifically designed to separate the functionality into two objects, the UI and the Board. This was great practice on how object orientation works.

Specific coding tricks: In order to get through the project’s specific hurdles, our group had to learn niche coding tricks like clearing the command window, allowing input without having to hit ‘enter’ every time, and even changing the color of the output. These small skills improved our overall toolbox of coding tricks for each member.

**Future Work:**

Coming into this class, every member knew little about the C++ interface but after this project we each gained a sense of confidence in the implementation of C++ through our team collaboration. Each individual member broadened his/her coding knowledge and is now able to understand how to apply techniques like object orientation and encapsulation to future programming projects. Additionally, because of our extensive GitHub use, each individual member was able to demonstrate a strong understanding of the software, which will allow us to excel in future group projects as well as in a professional setting. Finally, every member understands the importance of communication for future projects and with our lessons learned, we are now able to not only apply effective C++ techniques to future code but we also understand how and what makes a team so efficient.