Lessons Learned

Due to my pre-existing programing abilities from numerous past projects, I did not learn development and testing skills as much as I improved those I already had through practice. These skills included project planning, code structing, Object-Oriented development, and use of git. However, I did learn a lot about C++ development as it was my first time using the language. While I drew core OOP concepts from other languages (primarily Java), I learned how to implement them in C++ throughout the project. This will be useful for working with lower-level and embedded systems, something I will be doing with various design teams.

I also learned a lot about working with the various structures covered in the course, primarily graphs and heaps. As a graph forms the backbone of the simulation, I learned a lot about developing the graph and implementing Dijkstra's algorithm to navigate it. As part of Dijkstra's algorithm, I learned more about the implementation and usage of a heap as a priority queue. Additionally, I got more practice in learning from online resources in implementing these features as the relevant course material was weeks away.

Code Stats

According to cloc, a command-line utility which counts lines of code, the project contains close to 2000 lines of code and 900 lines of comments.

According to git-time, a Nodejs script which calculates time based on git commits, I have spent around 55 hours on the project.