Reynolds Okyere Boakye

70972024

Intermediate Computer Programming

Reflective Report of C++ RouteFinder Implementation

To be honest, replicating the route finder algorithm in C++ has not been all that easy for me, yet this has been a good experience. I took up the challenge to explore the C++ without relying on a teammate, therefore I did the work alone. This project has exposed me to the proper practice of separating the header files which have the abstract layouts and the actual implementation in the cpp file. Also, I was able to compile the seven cpp files I used in my program using the code "*g++ RunProgram.cpp Computation.cpp FileReader.cpp FileWriter.cpp Map.cpp Node.cpp ProblemSpace.cpp -o routefinder*". Generally, I am proud it works very well and faster than some of my mates' own. But personally, I am proud I was able to do this on my own through exploration.

The Route finder algorithm works with seven cpp files namely RunProgram.cpp Computation.cpp FileReader.cpp FileWriter.cpp Map.cpp Node.cpp ProblemSpace.cpp and their corresponding headerfiles with the exception of RunProgram which has the main function. Through the program, I used an unordered map to key track of the starting city as the key and the list of the corresponding destination as the value. I used a vector as an array in many instances. I also utilized queue as the frontier of my breadth-first search.

I made use of a class Node that represented the current city, the parent node and the number of trips it took to get there. I made sure to take care of edge cases, for instance, I used regex to control the ',' in places they were not supposed to be.