



ROUTE PLANNING PROJECT WITH C++

Mercy Onyinyechukwu Chimezie

58142024

Department of Computer and Information Systems, Ashesi University.

Intermediate Computer Programming, Semester II.

Robert Sowah & Banahene Osei

November 29, 2022

This project is a route-planning project where a database of airports, airlines and routes was provided. I was tasked with writing a programme which outputs a series of flights that take the passenger when given a start city and an end from the start city to the destination city. In this comprehensive project, I could define classes in c++, fields, constructors, use and apply pointers effectively and implement methods. I also understood how to use appropriate types, efficient algorithms, and data structures to solve problems. I wrote the main method, including console I/O and appropriate commenting.

To fully understand the processes in solving the problem, I wrote down my steps in solving this project problem to visualise it; then, I started writing the code from there. I began by creating classes for each database we were provided with. These classes had constructors and getter methods. Different data structures were helpful in some of these cases. I used the Haversine method, which computed the distance between two points given the latitude and longitude coordinates. Periodically, I tested the code to ensure everything was going well. Finally, I co-wrote the algorithm for route pathfinding. I used the Depth first algorithm, which finds a route to a place based on its path cost. In this project, for efficiency, we used a priority queue as its frontier, which helps get the state with the lowest path cost for expansion.

In conclusion, this pathfinding project was used to find the shortest path to a destination, given a source location and a destination location. The depth first search algorithm is identical to the general graph search algorithm, except for using a stack and adding an extra check in case a shorter path to a frontier state is discovered. The data structure for frontier needs to support efficient membership testing, combining the capabilities of a stack and a hash table (Russell and Stuart, 2010). I got better understanding of syntax in C++, pointers, and OOP in C++.

It was quite challenging having to implement some data structures in C++, but we figured it out anyway. I learnt how to use Hashmaps, and sets in C++, more importantly I saw the differences between C++ and Java.

References

Russell, Stuart J. (Stuart Jonathan). (2010). Artificial intelligence: a modern approach. Upper Saddle River, N.J.:Prentice Hall,