
# ROUTE PLANNING PROJECT WITH JAVA

Mercy Onyinyechukwu Chimezie

58142024

Department of Computer and Information Systems, Ashesi University.

Intermediate Computer Programming, Semester II.

David Ebo Adjepon & Banahene Osei

September 30, 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 java, fields, constructors, and 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 Javadoc.

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. Then I created the main class in another file(my driver class), which included the main method and a class called "theObjects", where I created objects for the databases to use effectively. In this main class file, too, I created the "readnWrite" class that was used in reading from the file and writing to it. 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 wrote the algorithm for route pathfinding. I used the Uniform Cost Search algorithm, which finds a route to a place based on its path cost. It uses 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 uniform cost search algorithm is identical to the general graph search algorithm, except for using a priority queue 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 priority queue and a hash table (Russell and Stauart, 2010).

# References

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