

Development

Adam Nguyen-adammn2 & Rithik Booreddy-rithikb2

*Disclaimer: Weekly updates were pushed every week under the file labeled “Minutes CS 225 Project.pdf” & “Minutes 11-27 CS 225 Final Project.pdf”

Week 1 (11/18/20):

This was our first week so we spent time figuring out what our project was going to be about. We decided to utilize the airline datasets for our project. Our project is going to have a user input a certain shape, and return a route that will mimic that shape. The route will be dependent on airports and feasible routes according to the data. Our next steps include parsing the datasets and learning how to use them.

Week 2 (11/27/20):

We assigned data to variables and used iostream to read and display values on VSCode. There were some issues with committing our files to the GitHub repository through VSCode. We solved this issue by December 1st. We also planned out how we were going to implement our idea. We decided that our next steps for next week include creating a weighted graph with distances on the edges, creating class/struct for airport nodes so that data is more organized, and formulating logic for choosing an initial airport so that a traversal does not need to check unnecessary airport nodes (boolean array that represents possible directional routes).

Week 3 (12/3/20):

We created an edge class that would hold information regarding the edges between airports. The data would be loaded from the routes dataset and saved into different variables including airport name, distance, and angle. Distance was defined using the Haversine formula and is based on latitude and longitude. To maintain our edges, we created an edges map that would map each airport to each other, as well as hold the other required information. We also created a vertex class that holds all of our nodes, as well as the latitude, longitude, and IATA information. We used iostream to determine polygon route characteristics which would dynamically calculate the range of possible internal angles for the user to choose. Finally, we used an array of vectors of Edge pointers of size 72 ($360/5$) and used clock-like iteration to find the user-specified angle.

Week 4 (12/11/20):

This week, we implemented the algorithms (BFS and Landmark Problem) in the `map_route` class. BFS is used to efficiently find a route given the geometric criteria that is defined by the user. The Landmark Problem is used to then further optimize the valid path by considering neighboring airports along a given route. Most of our time this week was spent finishing the implementation of these algorithms in our codebase, as well as testing. We also spent time writing the results report, as well as recording our presentation on its usage.