Evaluation Report: EleNa

Group: RedCoders

1. Functional Requirements:
    a. Given a start and end location, determine a route that maximizes or minimizes elevation gain, while limiting the total distance between the two locations to x% of the shortest path.
    b. Algorithms: Dijkstra, A-star
    c. Display route information, for instance, distance and elevation gain.

2. Algorithm Analysis:
    a. We implemented two different algorithms namely Dijkstra and A-star. Dijkstra is a special case of A star where we set the heuristic value to 0.
    b. A star finds the shorter path faster than Dijkstra as we use the heuristic of the distance till the destination node to reduce the time complexity. But Dijkstra is guaranteed to find the shortest path.
    c. We use binary search algorithm to find the best scaling factor which is used to construct the weighted edges of the graph using below equation:
        i. Edge_weight = scaling_factor * edge_length + elevation_gain.
3. Usability:
    a. The current implementation is valid for Amherst, Massachusetts as the geographical area.
    b. The implementation ensures usability. For instance,
        i. User input: Ability to mark points on map or input address using text box.
        ii. Output: Shortest route as well as elevation considered route are displayed.
        iii. Reset: Set application to default configurations.
4. Extensibility:
    a. The use of **Strategy design pattern** makes it easy to extend the implementation to more algorithms.
    b. The geographic region can be extended to wider regions by changing the radius considered.
    c. The use of **Observer design pattern** renders the implementation efficient for concurrent access by multiple users.
5. Testability:
    a. The use of MVC design pattern ensures that the application is easily testable.
    b. Use of unittest ensures automation in test-cases.