Phase Zero - General Report (QA)

Group Members: Fatemeh Azami, Arman Eghlimi, Ali Behrad

April 15, 2023

This part will provide some info and suggestions about major parts of the project that should be focused on. As the application will be a food delivery system , it's not that hard to conclude that food ordering process and the delivery process are two main parts (and also challenges) of the app development. so as stated we will talk about **Routing** and **Ordering Food**.

In the previous article we discussed about the food ordering process. in this article the routing process will be covered.

Routing: **Building a graph model of urban traffic map includes 4 steps:**

Data collection : Possessing info about public transportation systems, the traffic peak hours, places where experience traffic jams etc can be useful to build a traffic map of a city. also you can use pre-built maps using APIs in the app.

Graph construction : By the data provided at the previous section, we are now able to construct a graph that models the city. a suitable graph may contain info about the paths, details about the buildings, time estimation system based on the average speed of vehicles, directions of certain paths and so on.

Routing algorithms : A delivery system without an optimized searching algorithm really sucks! Fastest route finding algorithms, algorithms for finding the routes with the least traffic jams and etc can be used.

Setting up : Now that most of the desired info provided, it's time to assemble things and build our urban model which is mobilized with some optimized algorithms and models.

Saving all these info can be done using mathematical relations and tools such as matrices. also we can store precise geographical location of places in something like a graph-database.

To implement all these we need some classes to store info about the path, starting and ending points and info about the paths will be crossed, time required etc. to calculate the shortest path between the starting and ending points we can use two algorithms called A* and Dijkstra. however there are other routing algorithms out there but we discuss about these ones.

1) A* Algorithm: the A-Star algorithm is used to find the shortest path between starting and ending nodes. providing the number of nodes and edges between them as well as the length of the edges (also called the weight) and heuristic, the A* algorithm finds the shortest path from the specified start node to all the other nodes. the principle behind the A* algorithm is to iteratively look at the node with the currently smallest priority(which is the shortest distance from the start plus the heuristic to the goal) and update all not yes visited neighbors if the path to it via the current node is shorter. This is similar to Dijkstra algorithm, with the difference of that the lowest priority node visited first rather than the shortest path one. Dijkstra uses the distance as the priority, whereas A* uses the distance plus the heuristic.
Heuristic that is a straight line distance :
1. Initialize the distance to the starting node as 0 and the distances to all other nodes as infinite.
2. Initialize the priority to the starting node as the straight-line distance to the goal and the priorities of all other nodes as infinite
3. Set all nodes to ”unvisited”
4. While we haven’t visited all nodes and haven’t found the goal node :

1. Find the node with currently lowest priority (for the first pass, this will be the source node itself).

2. If it’s the goal node, return its distance.

3. For all nodes next to it that we haven’t visited yet, check if the currently smallest distance to the neighbor is bigger than if we were to go via the current node.

4. If it is, update the smallest distance of that neighbor to be the distance from the source to the current node plus the distance from the current node to that neighbor, and update its priority to be the distance plus its straight-line distance to the goal node.

2) Dijkstra Algorithm: Dijkstra algorithm is one of the prominent algorithms to find the shortest path from the source node to the destination node. it uses the greedy approach to find the shortest path. how it works is it starts from the starting point and ignores the longer distances while doing an update:

1. All nodes should be marked as unvisited.

2. All nodes must be Initialized with the ”infinite” (a big number) distance. The starting node must be Initialized with zero.

3. Mark starting node as the current node.

4. From the current node, analyze all of its neighbors that are not visited yet, and compute their distances by adding the weight of the edge, which establishes the connection between the current node and neighbor node to the current distance of the current node.

5. Now, compare the recently computed distance with the distance allotted to the neighboring node, and treat it as the current distance of the neighboring node.

6. After that, the surrounding neighbors of the current node, which has not been visited, are considered, and the current nodes are marked as visited.

7. When the ending node is marked as visited, then the algorithm has done its job.

8. Pick the unvisited node which has been allotted the minimum distance and treat it as the new current node. After that, start again from step4.

3) Heuristic Algorithm: A heuristic is a technique that is used to solve a problem faster than the classic methods. These techniques are used to find the approximate solution of a problem when classical methods do not. Heuristics are said to be the problem-solving techniques that result in practical and quick solutions. Heuristics are strategies that are derived from past experience with similar problems. Heuristics use practical methods and shortcuts used to produce the solutions that may or may not be optimal, but those solutions are sufficient in a given limited time frame. There are various types of heuristics, including the availability heuristic, affect heuristic and representative heuristic. Each heuristic type plays a role in decision-making. Let’s discuss about the Availability heuristic, affect heuristic, and Representative heuristic.

Advantages of Stated Algorithms:

1. *A* Algorithm*: A* search is the most commonly known form of best-

first search. It uses the heuristic function h(n) and cost to reach the node n from the start state g(n). It has combined features of UCS and greedy best-first search, by which it solve the problem efficiently.

2. *Availability heuristic*: Availability heuristic is said to be the judgment that people make regarding the likelihood of an event based on information that quickly comes into mind.

3. *Representative heuristic*: It occurs when we evaluate an event's probability on the basis of its similarity with another event.

4. *Affect heuristic*: It is based on the negative and positive feelings that are linked with a certain stimulus. It includes quick feelings that are based on past beliefs. Its theory is one's emotional response to a stimulus that can affect the decisions taken by an individual.

5. *Limitations*: because this algorithm is based on the information about the past, it may not be useful at present time since decision makings change fast. also its hard to change the solution once you used these types of solutions frequently.

Github Project Repo
Trello Board
Author : Ali Behrad