Project Title

TRAFFIC SIMULATION PROJECT

Student / Team Information

Team Leader *Name: Dhruv Bhatt* *University roll no.:2418455* *Student ID: 240211470* *Email:dhruvbhatt6396@gmail.com*	
Name: Dibyanshu Negi *University roll no.: 2418461* *Student ID: 240111185* *Email: negid0925@gmail.com*	
Name: Kavyansh Dhasmana *University roll no.: 2418590* *Student ID: 240211251* *Email: kavyanshdhasmana99.9@gmail.com*	

PROPOSAL DESCRIPTION

Motivation

Traffic congestion is one of the most common and serious issues faced in modern cities due to the rapid increase in the number of vehicles. Efficient management of traffic flow and finding the shortest route between two points is essential to reduce travel time, fuel consumption, and overall pollution. The motivation behind this project is to simulate and study how cars move across a city road network and how congestion can be represented using fundamental data structures. In this project, the city roads are modeled as a graph, cars are represented as queues, and the shortest path is found using the Breadth-First Search (BFS) algorithm. This not only provides a simplified view of real-world traffic management but also helps in applying classroom concepts like ***graph traversal, queue operations, and algorithms*** *to solve practical problems. The project further motivates learning because it can be extended to real applications such as GPS navigation, intelligent traffic systems, and smart city planning. Thus, the project bridges the gap between academic learning and real-world problem-solving, giving us both practical exposure and deeper understanding of data structures and algorithms.*

Project Goals and Milestones

Project Goals

1. ***Modeling the City as a Graph***
   - *Represent junctions as nodes and roads as edges.*
2. ***Pathfinding using BFS***
   - *Implement Breadth-First Search to find shortest path between two points.*
3. ***Car Movement Simulation***
   - *Use queues (FIFO) to represent cars on roads.*
4. ***Traffic Congestion Handling***
   - *Detect queue overflow as traffic jam.*
5. ***Visualization & Output***
   - *Display shortest path and car movement.*
   - *(Optional) Graphical visualization using Python libraries.*

Project Approach

1. Problem Understanding

- *Increasing vehicles lead to traffic congestion.*
- *To manage traffic, shortest path routing and flow simulation are required.*
- *Goal: Represent city as a* ***graph****, cars as* ***queues****, and apply* ***BFS*** *for shortest path.*

---

2. System Design

- ***Graph Representation****: Junctions → Nodes, Roads → Edges.*
- ***Car Representation****: Each car assigned a source & destination.*
- ***Queue Concept****: Each road acts as a queue (FIFO order).*
- ***Algorithm Choice****: BFS used since all roads are assumed equal weight (unweighted graph).*

---

3. Implementation Steps

1. ***Graph Module***
    - *Represent city using adjacency list.*
2. ***Pathfinding Module***
    - *Use BFS to find shortest path between source & destination.*
3. ***Queue Module***
    - *Cars move along roads using enqueue/dequeue operations.*
    - *Queue overflow = traffic jam.*
4. ***Traffic Simulation***
    - *Cars are placed on graph and move step-by-step.*
    - *Multiple cars tested simultaneously.*
5. ***Optional Features (if extended)***
    - *Traffic signals (Red/Green).*
    - *Alternate routing when congestion occurs.*
    - *Visualization using matplotlib / networkx.*

---

4. Testing & Validation

- *Test with single car from source to destination.*
- *Test with multiple cars to check queue behavior.*
- *Test congestion by reducing queue size.*
- *Validate BFS always finds the shortest path.*

---

6. Outcome

- *Working simulation of traffic flow.*
- *Cars successfully move via shortest path.*
- *Congestion and queues properly handled.*
- *Visualization (if added) for better understanding.*

System Architecture (High Level Diagram)

1. Input Layer

- *User enters city map (graph of junctions and roads).*
- *User enters car details (car ID, source, destination).*

2. Processing Layer

- *Graph Module: stores city as adjacency list.*
- *BFS Module: calculates shortest path between source & destination.*
- *Queue Module: simulates cars on each road (FIFO).*
- *Traffic Control Module (optional): handles signals and congestion.*

3. Output Layer

- *Displays shortest path chosen by each car.*
- *Shows cars entering and leaving queues (traffic flow)*

Project Outcome / Deliverables

1. Shortest Path Calculation

- *BFS algorithm successfully finds the shortest path between any two junctions in the city.*

2. Traffic Flow Simulation

- *Cars are represented as queues on roads.*
- *Movement of cars (enqueue/dequeue) shows real-time traffic flow.*

3. Traffic Congestion Modeling

- *Queue overflow on roads represents traffic jams.*
- *Cars wait in queue until road is clear.*

4. Basic Traffic Management

- *If signals are implemented, cars stop at red lights and move on green.*

5. Visualization (if added)

- *Graphical display of city map using Python libraries (networkx, matplotlib).*
- *Visualization of car movement and traffic build-up.*

Assumptions

1. Graph Representation

- *City roads are modeled as a graph (junctions = nodes, roads = edges).*
- *All roads are bidirectional unless otherwise specified.*

2. Car Representation

- *Each car is represented as a unique ID or number.*
- *Cars move one step (one edge) at a time.*
- *Cars follow the First-In-First-Out (FIFO) rule on each road (queue).*

3. Path Finding

- *All cars use BFS to calculate the shortest path (minimum number of junctions).*
- *Travel time on each road is assumed to be equal (unweighted graph).*

4. Traffic Flow

- *Only one lane per road is considered.*
- *Queue overflow indicates traffic congestion or jam.*

- *Cars instantly move from one road to another once space is available (no overtaking).*

5. Traffic Signals (if implemented)

- *Signals operate in fixed time cycles (Red/Green).*
- *Cars stop at red signals and proceed on green.*

6. Simplifications

- *No accidents, breakdowns, or real-world disturbances considered.*
- *Roads are assumed to have uniform length and capacity.*
- *All cars move at the same average speed.*

References

https://docs.python.org/3/library/collections.html
https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/
https://www.geeksforgeeks.org/queue-data-structure/