City Traffic Simulator

Requirements Analysis Document

Team Name
Chavez, Bryan
Majid, Mohammad
Nguyen, Brian
Vuong, Nguyen Huy

Introduction

Purpose

Mayor Mann has an extreme crisis in his hands as its up to him to successfully find a system that will organize his city for the better. Organize the city by increasing the functionality of city traffic lights and stop signs in order to maximize the number of vehicles to a particular destination and minimize the travel time. The following document details the design and structure of the City Traffic Simulator.

Scope

This application is a simulation of the appropriate amount of traffic lights and stop signs for the city of Pacopolis. The mayor of the city is looking to maximize the flow of traffic from point A to point B while minimizing the travel time. Being able to provide an effective simulator in order complete the task Mayor Mann has given us.

Objectives and Success Criteria

The objective is to maximize flow of traffic from start to destination and minimizing the time of travel. Also, successfully placing the appropriate amount of traffic lights and stop signs in the areas needed.

Definitions, Acronyms, and Abbreviations

OOP[1] : Object-oriented programming

GUI[2] : Graphical user interface.

UML[3] : Unified Modelling Language

RID: Requirement Identifier.

RAD : Requirement Analysis Document

CSV : Comma-seperated values

References

City map of Pacopolis: http://pacman.wikia.com/wiki/Maze

Overview

The overview of this project is mainly organizing enough traffic lights and stop signs to be processed correctly and set in the city of Pacopolis. The city of Pacopolis' goal is achieving the necessary help in setting the traffic lights and stop signs to help minimize the amount of time needed for each vehicle to get to their specific destination. This system will eventually minimize accidents and damages done in the city too eventually becoming a functional city that will help portray a correct way of driving getting to the needed destination with less time and hopefully no accidents.

Current System

The current system will be built and run

Nonfunctional Requirements

N1. Change logs and git branches will be used for keep track of different versions of the simulation.

N2. A GUI application will be created to represent the traffic simulations.

N3. The application will be runnable on any device.

N4. The application can send alerts if there might be any conflicts going on.

N5. The application will be user interactive.

System Models

Use Case Model

**On Page 5-12

Structural Model

The structure of this simulation well have a organized map. The map will include several vehicles and the vehicles will be tested to find the least amount of time needed to go to a specific destination. These cars will abide by the traffic rules and abide by the stop sign to stop at a specific amount of time in the preset of traffic lights or stop signs.

Behavioral Model

When the car interacts with the traffic light, depending on what the traffic light is set too the vehicle will stop for a certain amount of time. Also when the vehicle object is interacting with the stop sign object, the vehicle will abide by the rules presented in the program by stopping for a certain amount of time.

User Interface: Navigational Paths and Screen Mockups

User will be able to interact with the GUI application by adding / removing cars.

Additional application functionality will added more over time.

Proposed System

Overview

This application uses object-oriented programming. The program utilizes a few main objects which model traffic lights, stop signs, vehicles and the city map.

The Map class is the GUI class which has inherited all the methods of a GUI application. The Map class will simulates the city and movement of vehicles and traffic lights.

The Vehicle class will be the class to control Vehicle movement and functionality. It will make use of different type of data structures and algorithm to maximize a vehicle performances.

Functional Requirements

F1. The system will make use of different types of algorithms and data structures to maximize program efficiency, performances and response time.

F2. User establishes start and end destination

F3. Car will travel 15 mph for the first ⅛ mile and 30mph thereafter until it stops..

F3. When user selects start, simulation begins.

F4. Car will stop after it reaches destination.

F5. Car stops at a stop sign 3 steps

F6. Car stops at traffic light when red fo 3 steps

F7. Timer starts and stop per user

F8. Map resets per user

F9. Green light for particular direction and no cross traffic than light stay green.

F10. Red light and car from opposite direction comes with no cross-traffic than light turns green.

F11. Traffic in both directions then sensor-based traffic light is timer-based.

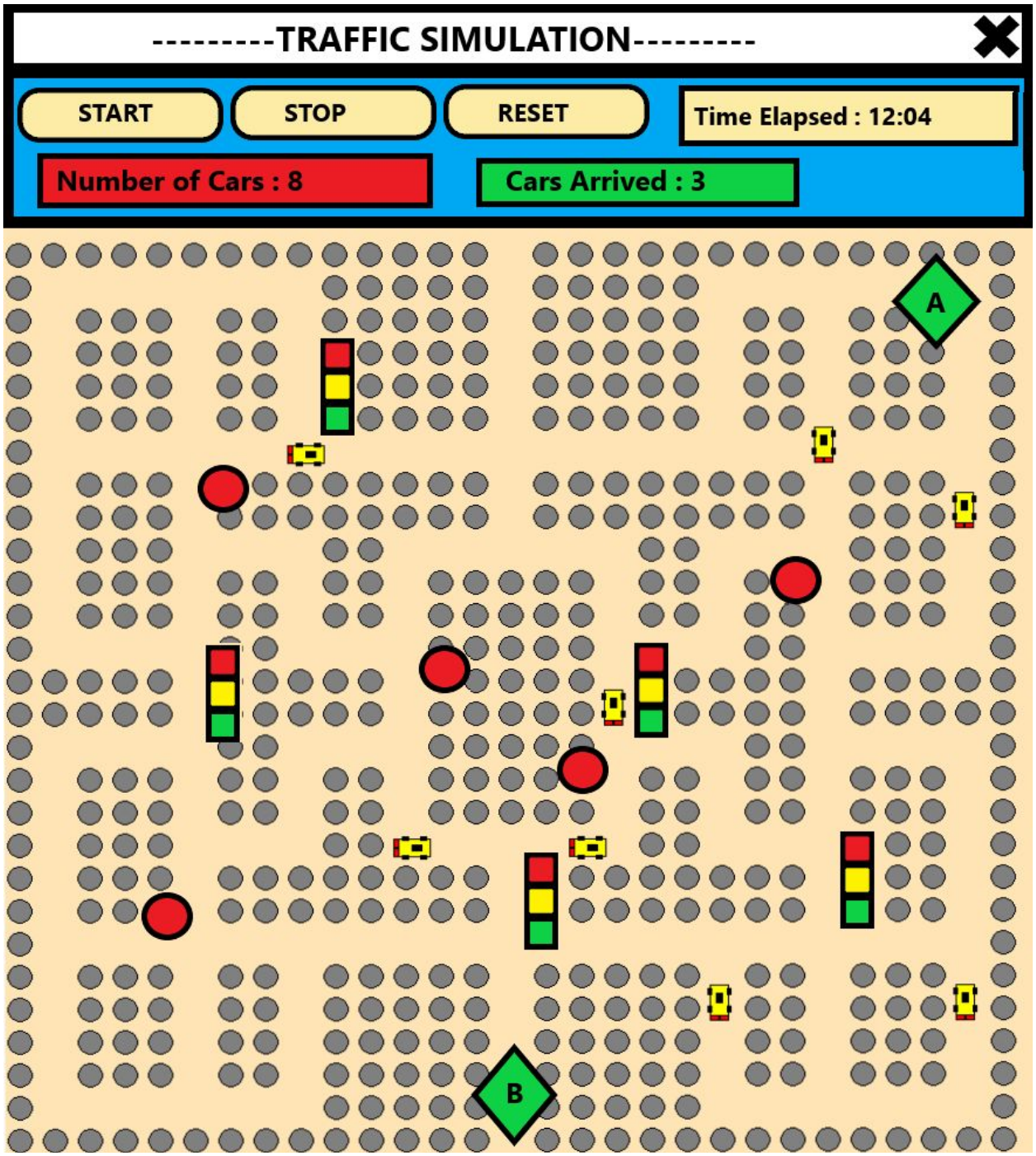
GLOSSARY

[1] : a programming paradigm that uses Object (data structures which have files and methods) which interact together within a program

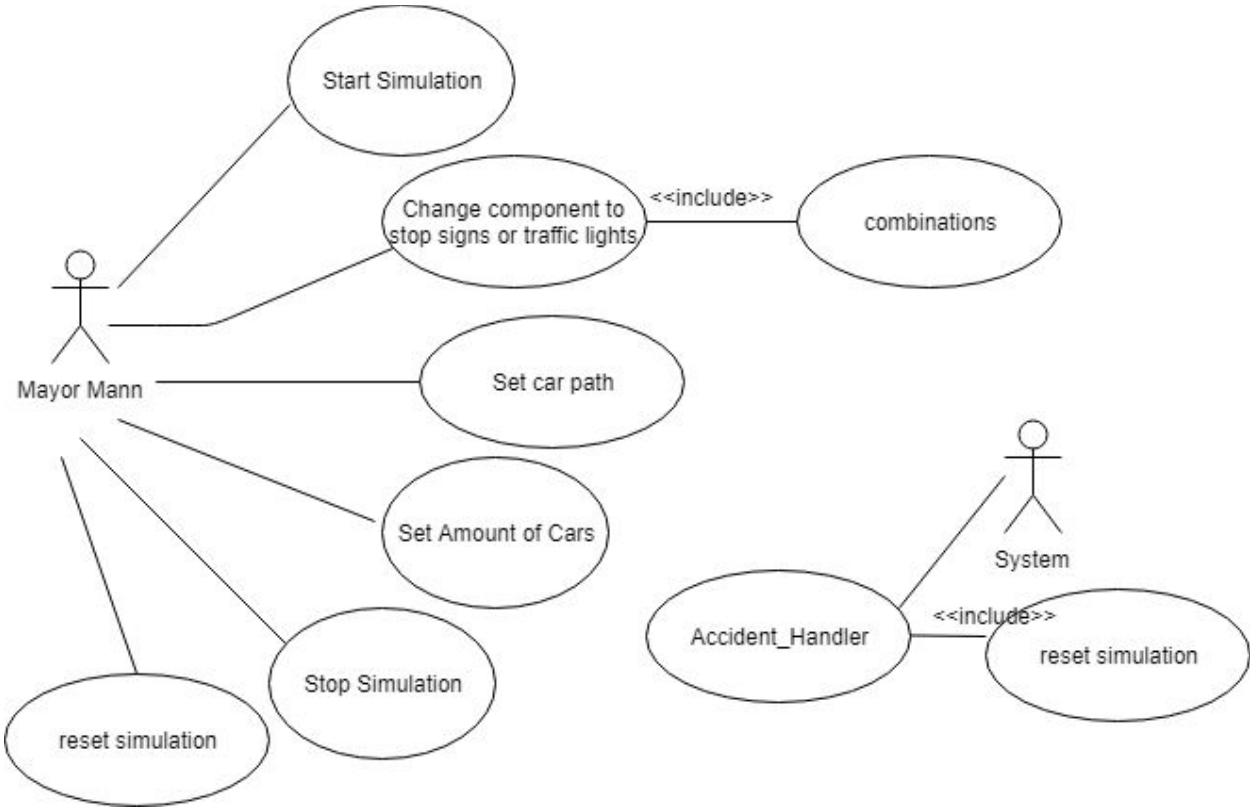
[2]: a graphical interface where graphical objects are created from interaction with the device.

[3]: A way of visualizing the structure of the system through appropriate use of diagram.

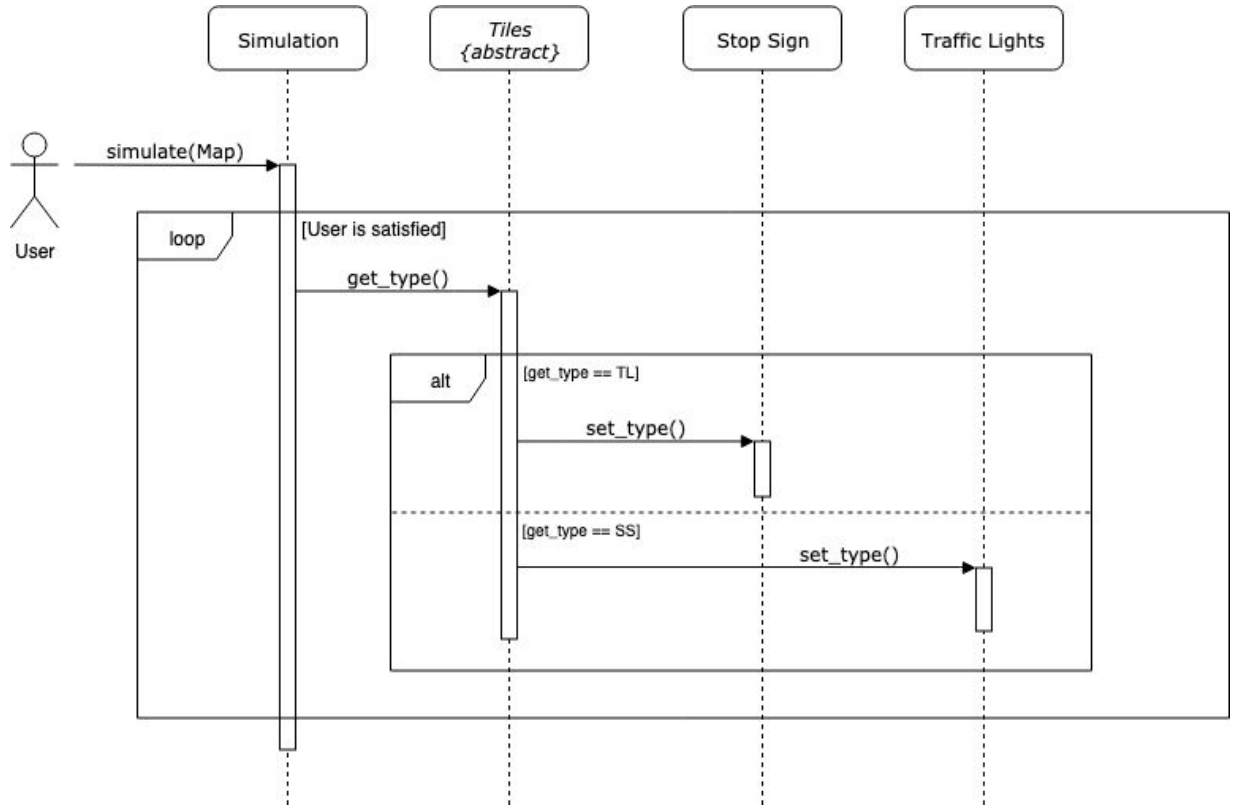
NEW WIREFRAME W/ MOCK UP:



Use case Diagram:



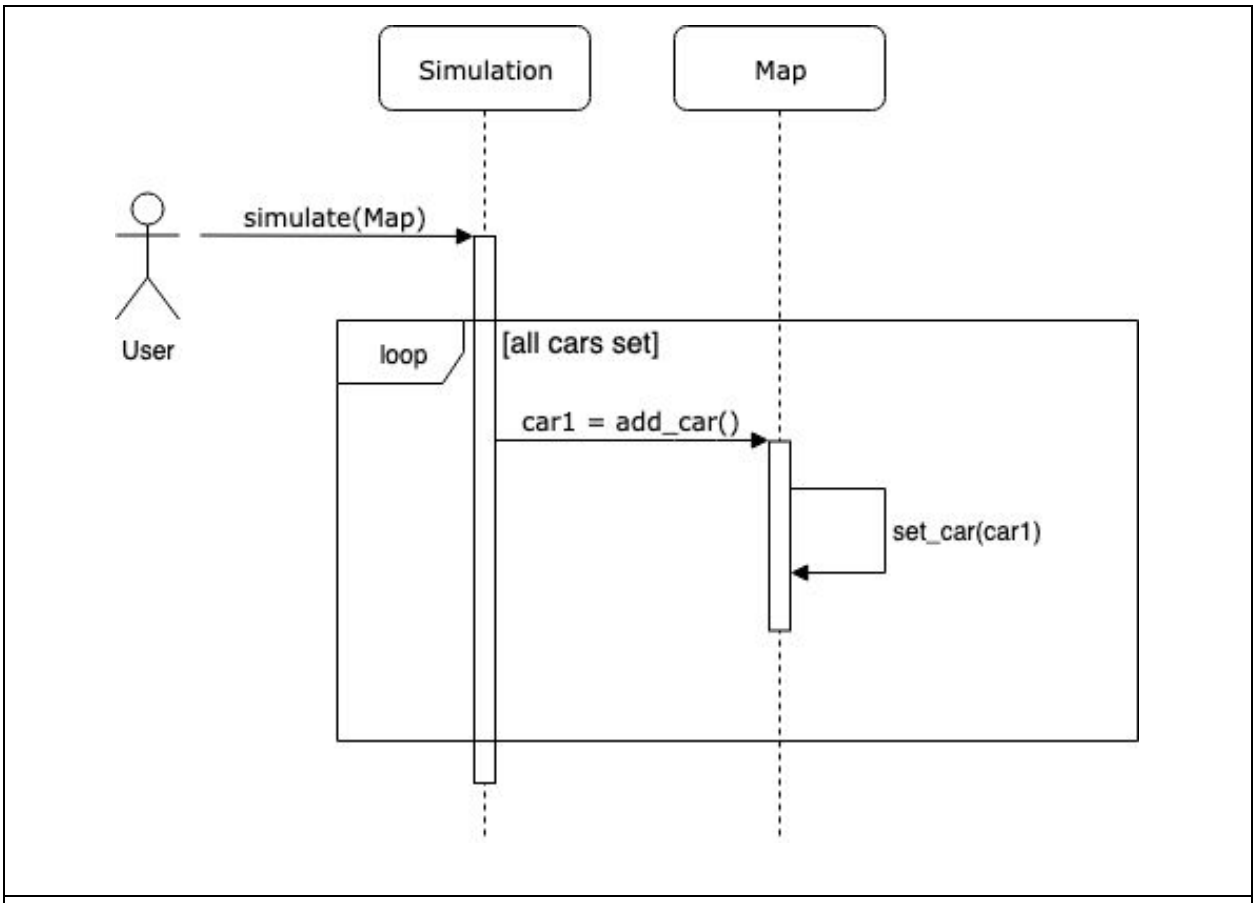
Sequence Diagrams:



Use Case 1

Actor: User

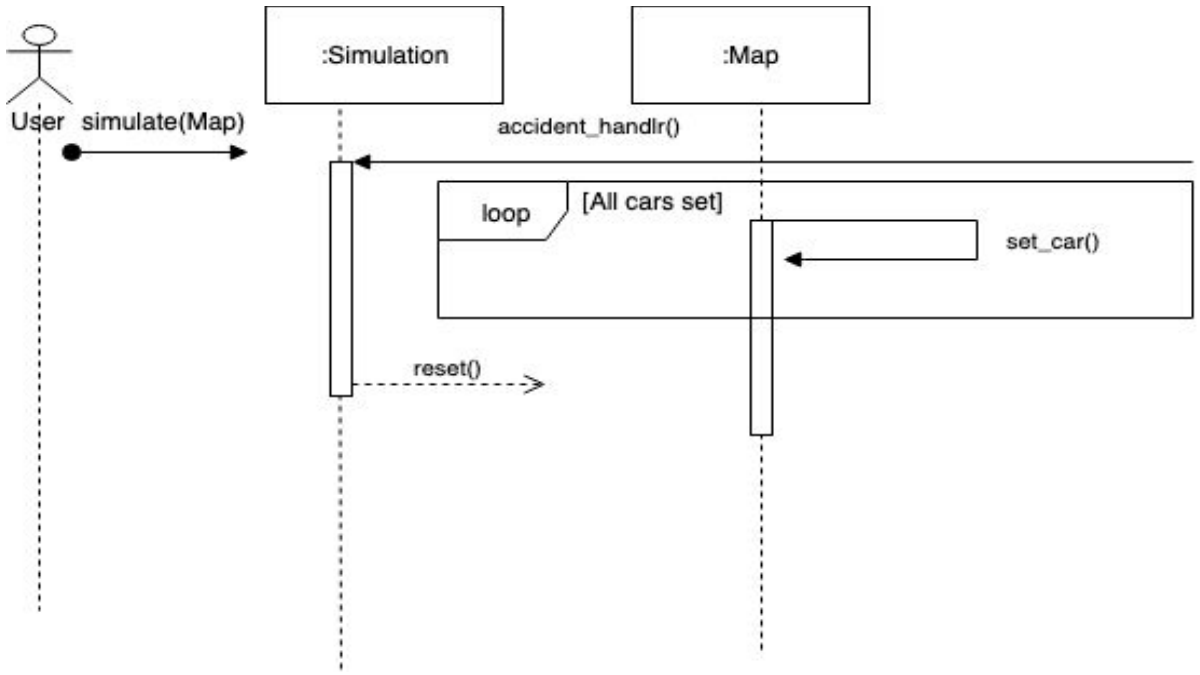
Basic Flow: The mayor will use a gui application that will give him the ability to represent traffic lights and stop signs to quest for the best combination of lights and signs from the desired starting point to ending.



Use Case 2

Actor: User

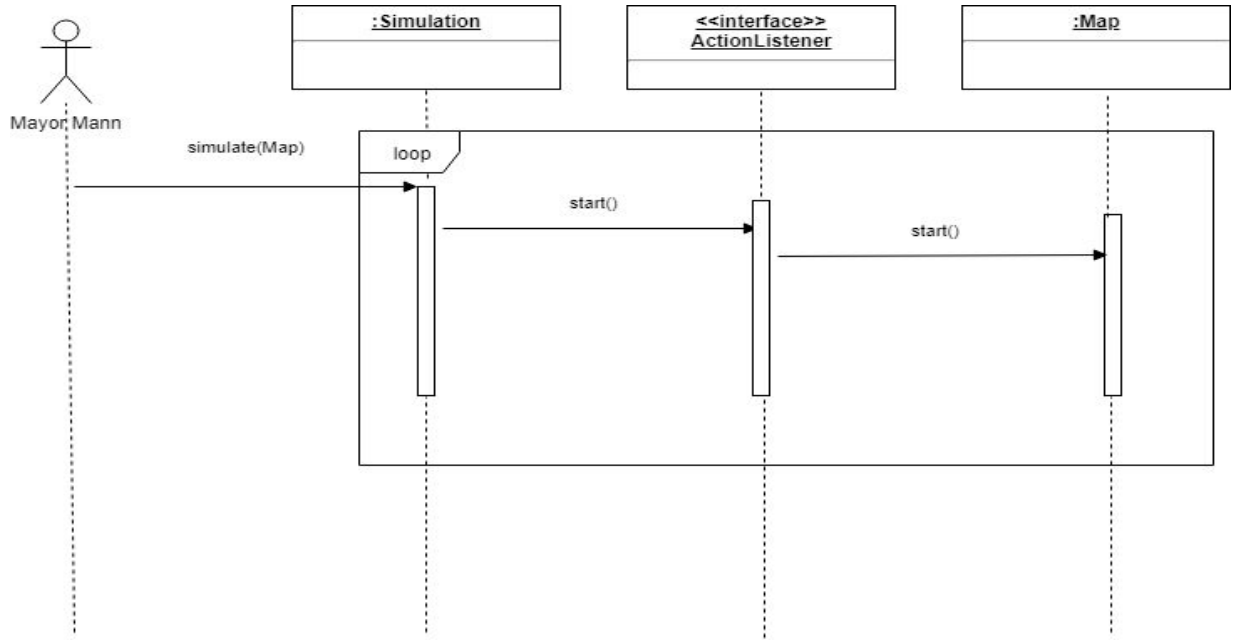
Basic Flow: Setting the appropriate amount of cars.



Use Case 3

Actor: User

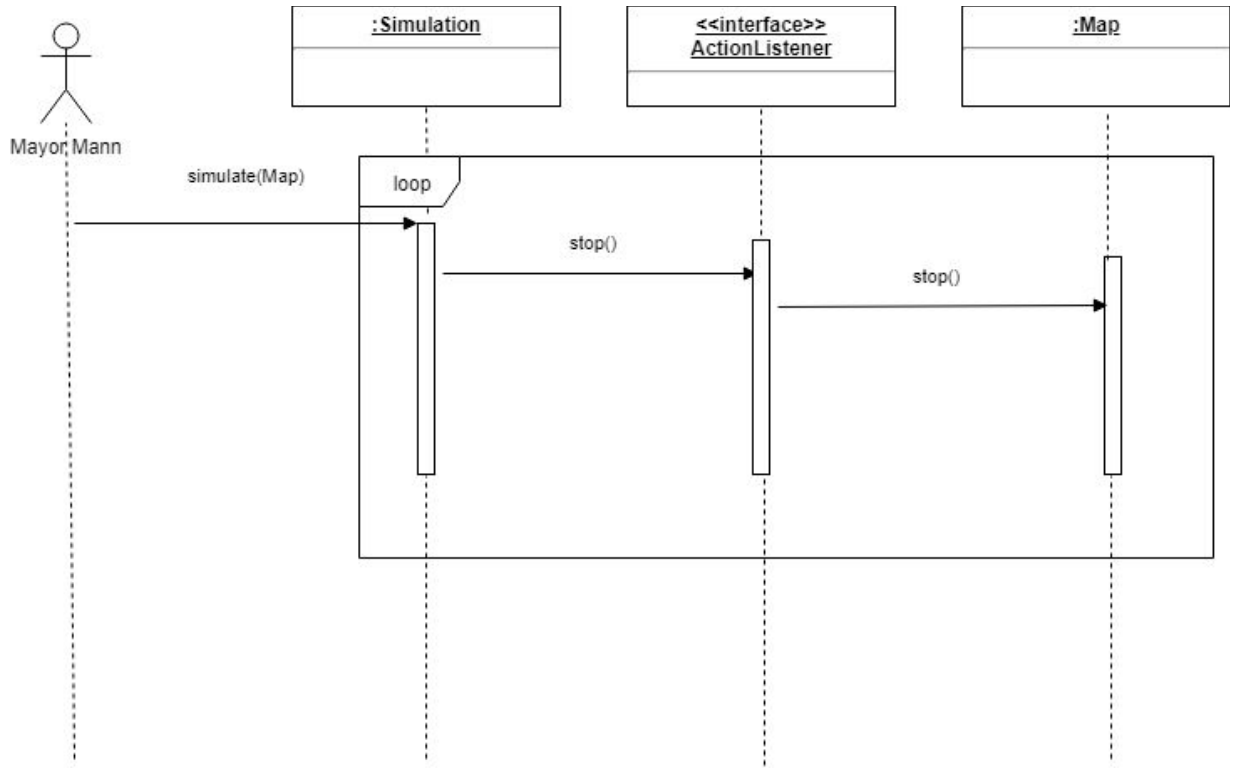
Basic Flow: While mayor Mann runs simulation and collision is created the simulator resets the simulator for the actor to use once again.



Use Case 4

Actor: Mayor Mann

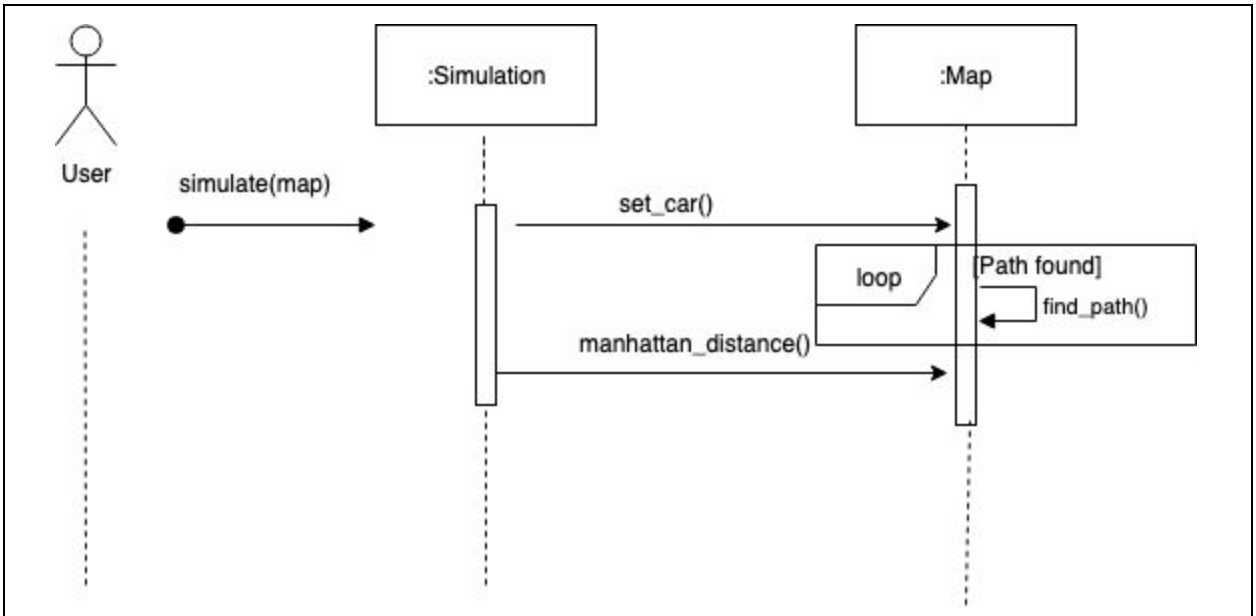
Basic Flow: The user starts simulation.



Use Case 5:

Actor: Mayor Mann

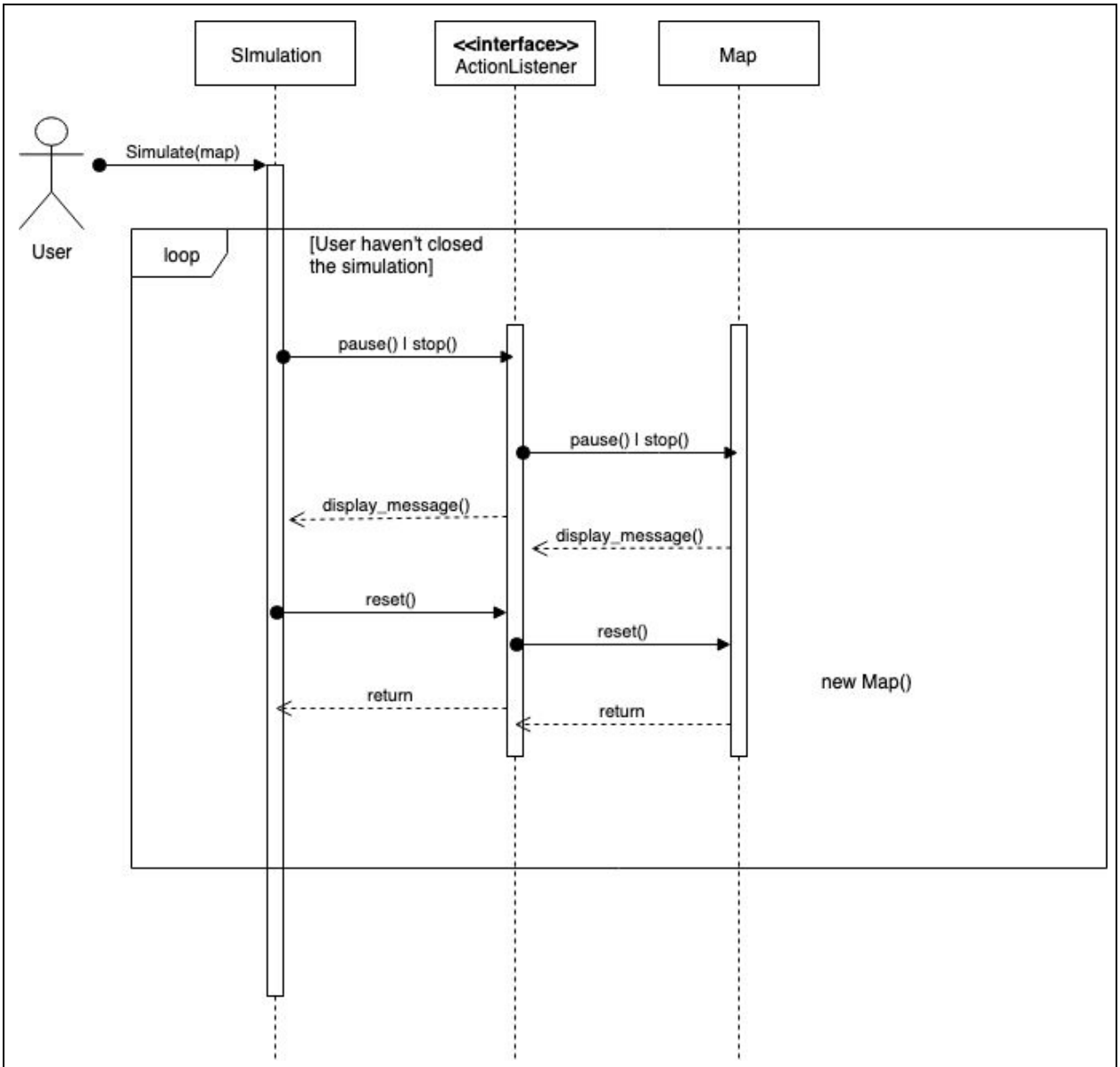
Basic Flow: The user stops simulation than the map freezes with where ever the vehicles are at and it's time.



Use Case 6:

Actor: User

Basic Flow: Set the path by placing where the car will start and use coordinates to set the distance.



Use Case 7:

Actor: User

Basic Flow: Reset simulation anytime the user decides to.