

COSC 3P91 – Assignment 2 – 7023609

AMEYA CHINDARKAR, Brock University, Canada

In this assignment, I developed a time-stepped traffic simulation game utilizing a console-based interface for player interaction. The primary objective was to extend the design formulated in Assignment 1 into a functioning simulation, guided by a provided UML Class Diagram to ensure a unified starting point for all participants. The simulation game models an urban traffic network as a graph where vehicles, including cars, trucks, and buses, navigate through road segments and intersections, simulating real-world traffic scenarios. The system is designed to be simple, and divided into 6 different packages, each handling an aspect of the game.

1 VEHICLE PACKAGE

The Vehicle package is a collection of classes and interfaces designed to model the various aspects of vehicles within the simulation environment. This package serves as the backbone for creating and managing different types of vehicles, such as cars, trucks, and buses, each with their unique attributes and behaviors.

1.1 Vehicle abstract class

This is the core class from which all specific vehicle types inherit. It defines common properties such as color, weight, size, maximum speed, movement status, reputation, and damage status. The class also provides the basic interface for vehicle movement through the move method, which must be implemented by subclasses.

1.2 Vehicle classes

These classes extend the abstract Vehicle class, adding specific characteristics and behaviors for different types of vehicles:

Car: Models a standard car, potentially with additional features like the number of doors.

Truck: Models a standard truck, potentially with additional features such as cargo space.

Bus: Models a standard bus, potentially with additional features like the number of passengers.

1.3 DamageStatus Class

Tracks and manages the damage status of a vehicle, including its current condition and history of damage received and inflicted. This affects the vehicle's performance and may influence player decisions.

1.4 Reputation Class

Manages the reputation of a vehicle or its driver within the simulation. This could influence interactions with other vehicles and affect gameplay, such as determining the outcome of challenges or gambles.

Author's address: Ameya Chindarkar, Brock University, 1812 Sir Isaac Brock Way, St. Catharines, ON, L2S 3A1, Canada.

2 MAP PACKAGE

The Map package is a crucial component designed to represent and manage the game's environment, specifically the layout and structure of the urban traffic network. This package encompasses a variety of classes and interfaces that together model the physical space within which vehicles move, including roads, intersections and lanes.

2.1 Intersection Class

Models the intersections within the traffic network, where road segments meet or cross. Intersections manage the flow of vehicles from different directions and may include features like traffic signals or signs to direct movement.

2.2 RoadSegment Class

Represents individual road segments within the traffic network. Each road segment may contain multiple lanes and connects two intersections, defining the pathways vehicles can travel.

2.3 Lane Class

Details the lanes within a road segment, allowing for more granular control of vehicle positioning and movement. Lanes support the simulation of lane-specific behaviors, such as lane changes or dedicated turn lanes.

2.4 TrafficElement Class

An abstract base class or interface from which specific map elements like RoadSegment and Intersection inherit. It provides common properties or methods that are shared across different types of traffic elements.

2.5 TrafficNetwork Class

Serves as the central class for managing the entire map or traffic network. It typically includes collections of road segments and intersections, forming a graph-like structure that vehicles navigate.

2.6 Point Class

A utility class representing a point in two-dimensional space, used to define the positions of various map elements or vehicle locations within the traffic network.

3 MOVEMENTSTATUS PACKAGE

The MovementStatus package is designed to encapsulate all aspects related to the movement and orientation of entities within the simulation environment. This package defines how vehicles navigate the traffic network.

3.1 Direction Enumeration

Defines the directions or more specific orientation options that a vehicle can assume within the simulation. : NORTH, SOUTH, EAST, WEST

3.2 MovementDecision Enumeration

Enumerates the possible decisions a vehicle can make regarding its next movement. This includes actions like CHANGE-LANE-LEFT, CHANGE-LANE-RIGHT, TURN-LEFT,

TURN-RIGHT, and KEEP-STRAIGHT, reflecting the choices available to vehicles at intersections or along road segments.

3.3 MovementStatus Class

This class represents the current movement state of a vehicle, including its position, speed, and direction. It serves as a comprehensive snapshot of a vehicle's dynamic status at any given time.

3.4 Position Class

Models the precise location of a vehicle within the traffic network.

4 GAME PACKAGE

The Game package serves as the central hub for coordinating the various components of the simulation, including vehicle management, traffic network configuration, and player interactions. This package orchestrates the game's logic, rules, and progression, as well as the main method.

4.1 ChallengeHandler Class

Manages scenarios where players can engage in challenges or take risks, such as navigating through busy intersections or overtaking vehicles. This component is responsible for determining the outcomes of such actions based on player decisions, vehicle states, and random chance.

4.2 GameEngine Class

The core class responsible for managing the game loop, updating the state of the simulation at each timestep, and handling the interactions between different components of the game, such as vehicles, the traffic network, and players. This class implements the MovementControl interface.

4.3 MovementControl Interface

Defines the contract for updating and managing vehicle movements within the simulation. Implementations of this interface process player decisions, enforce traffic rules, and update vehicle positions accordingly.

5 GUI PACKAGE

The GUI package focuses on providing a console-based user interface for interacting with the simulation. The package contains two primary components: the ConsoleInterface class and the UserInterface interface.

5.1 UserInterface Interface

This interface defines the contract for user interaction within the simulation. It outlines essential methods that any user interface, whether console-based or graphical, should implement to facilitate communication between the game and its players.

5.2 ConsoleInterface Class

An implementation of the UserInterface that utilizes the console (standard input and output) for all user interactions. This class provides a straightforward way to interact with the game through text commands and messages. It directly supports the methods outlined in

the `UserInterface` interface, ensuring that messages, prompts, and errors are appropriately handled and displayed in the console.

6 PLAYER PACKAGE

The `Player` package in a traffic simulation game is designed to encapsulate all aspects related to the players participating in the simulation, making decisions based on predefined algorithms.

6.1 Player Class

The central class within the `Player` package, representing an entity controlling a vehicle within the traffic simulation. This class is responsible for managing player-specific data, such as the vehicle they control and their current status within the game (e.g., location, score, reputation).