CP2406

Task 1

Traffic Simulator

James Makarios 14165676
12-8-2021

Table of Contents

Problem Specification 2
User Story 2
Problem Decomposition 4
Vehicle 4
Vehicle Types 4
Road 5
Road Types 5
Road Attributes 7
Road Rules 8
UML Diagram 10
Main Program 10
Vehicle 10
Road 11
Intersection 12
Program 13
Flow 13
Preset Road 14
Output 14
Vehicle 14
Road 15
Intersection 17
GitHub Repositories 20

Problem Specification

The program is a car traffic simulator. It aims to simulate the movement of vehicles such as cars, busses, and motorbikes in the road. Vehicles should not violate any traffic rules and aims to be as efficient as possible.

User Story

User 1

I want a simulator that allows me to create my own roads and intersections, so that I can simulate the traffic in my city.

Priority: High

Features to add:

- Road and intersection creation

User 2

I want a simulator that has a preset of roads and intersections, so I do not need to create one every time I want to simulate.

Priority: Medium

Features to add:

- Preset roads

User 3

I want a simulator that has different types of vehicles, so that I can observe the behaviors of different vehicles.

Priority: High

Features to add:

- Different types of vehicles

User 4

I want a simulator that has a traffic light system, so that I can observe the efficiency of the traffic light in managing traffic.

Priority: Medium

Features to add:

- Traffic light system

User 5
I want to be able to adjust the speed of the simulator, so that I can speed up / slow down the simulator to suit my needs.

Priority: Low

Features to add:

- Road and intersection creation

Problem Decomposition

In this program, there are 2 main objects to be developed. The 2 objects are:

1. Vehicle
2. Road

The simulator will be using continuous cycle to simulate the traffic. In each cycle, vehicles will move certain distance along the road. Additionally, the traffic light will also change if needed. The frequency of the cycle can be determined by the user.

Vehicle

A vehicle is an object that travels along the road. A vehicle will continuously move in different directions until it leaves the road. For this program, there will be 3 types of vehicles:

1. Car
2. Bus
3. Motorbike

Each vehicle has these attributes:

- Length
  Determines the length of the vehicle
- Breadth
  Determines the breadth of the vehicle
- Speed
  Determines how far the vehicle travel in 1 cycle

Vehicle Types

Car

A car is the most common vehicle in this simulator. Its attributes' values are:

- Length: 1 unit
- Breadth: 0.5 unit
- Speed: 1 – 5 units / cycle

Bus

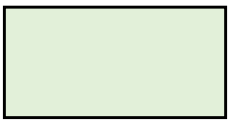
A bus is similar to a car but has longer length and lower maximum speed. Its attributes' values are:

- Length: 3 units
- Breadth: 0.5 unit
- Speed: 1 – 2 units / cycle

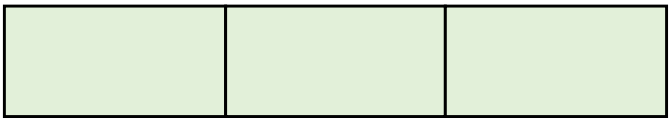
Motorbike

A motorbike is similar to a car but has a shorter length and higher maximum speed. Its attributes' values are:

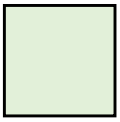
- Length: 0.5 unit
- Breadth: 0.5 unit
- Speed: 1 – 7 units / cycle



Car



Bus



Motorbike

Road

A road is an object where vehicles can travel on. Vehicles will continuously move along different roads throughout the simulation. For this program, there will be 3 types of roads:

1. Straight
2. 4-Way Intersection
3. 3-Way Intersection

A road can have these attributes:

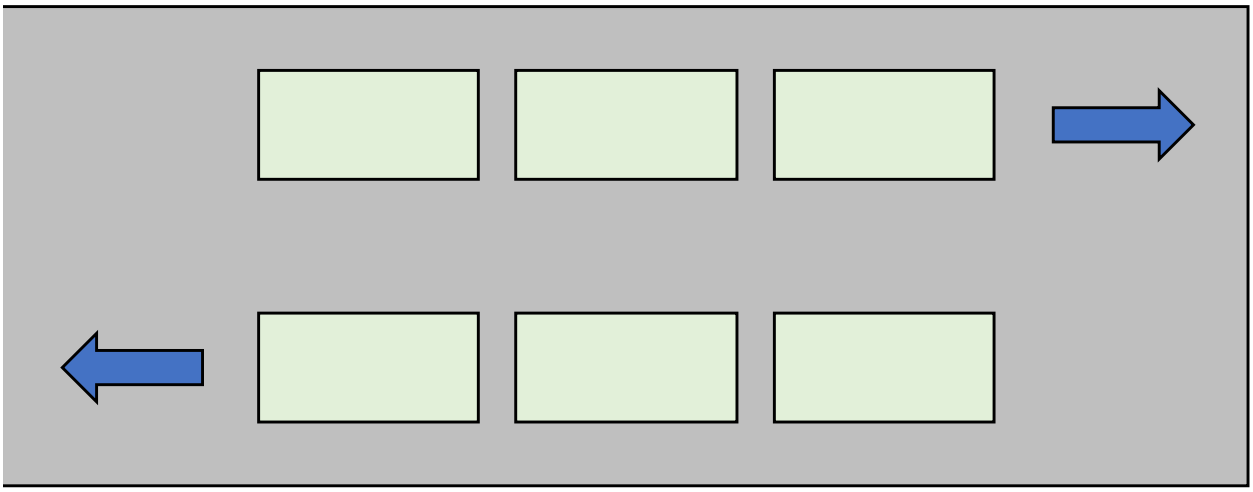
- Lane
- Traffic Light

In each road, there will be 2 lanes, 1 in each direction.

Road Types

Straight

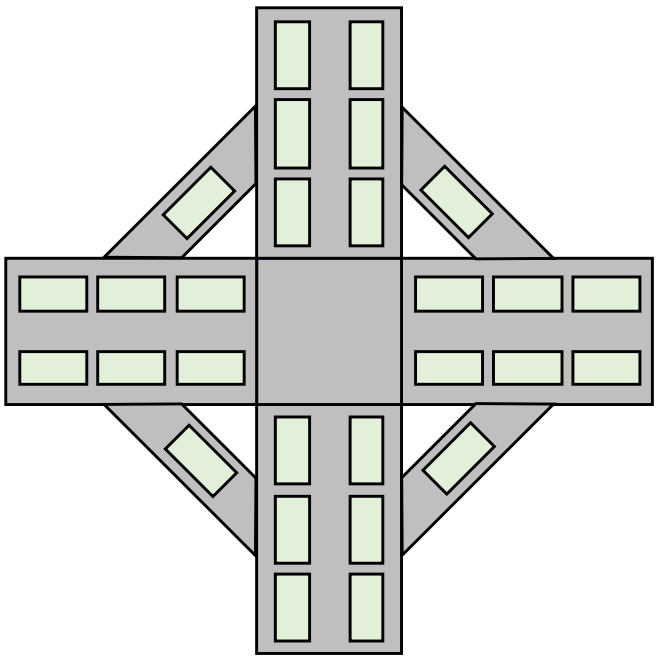
Straight roads are roads that allow vehicles to travel a specified direction. Vehicles are unable to turn direction while travelling in a straight road. The length of straight road can vary, with the minimum being 3 units, and the maximum being 15 units. There will be no traffic light in a pure straight road as there will be no change in direction from the vehicles.



Straight Road with a Length of 3 units

4-Way Intersection

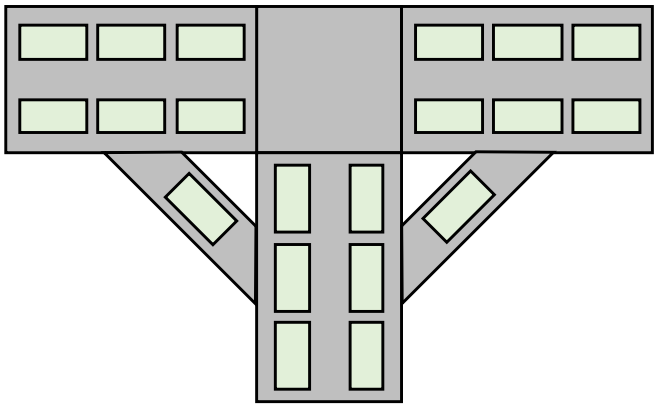
4-way intersection are intersection that allow vehicles to change their direction of travel. There are 4 directions that the vehicle will travel, North, East, South, and West. In each direction, there will be a straight road with length of 3 units. At the intersection, there will be a traffic light in each direction.



4-Way Intersection

3-Way Intersection

3-way intersection are intersection that allow vehicles to change their direction of travel. Unlike 4-way intersection, there will be only 3 directions that the vehicle can turn.



3-Way Intersection

Road Attributes

Lane

A lane is an object that allows vehicle to travel in 1 direction at a time. A lane can be divided into segments based on the length of a road. A segment is 1 unit in length, and can occupy either:

- 1 car
- 1/3 bus
- Up to 2 motorbikes

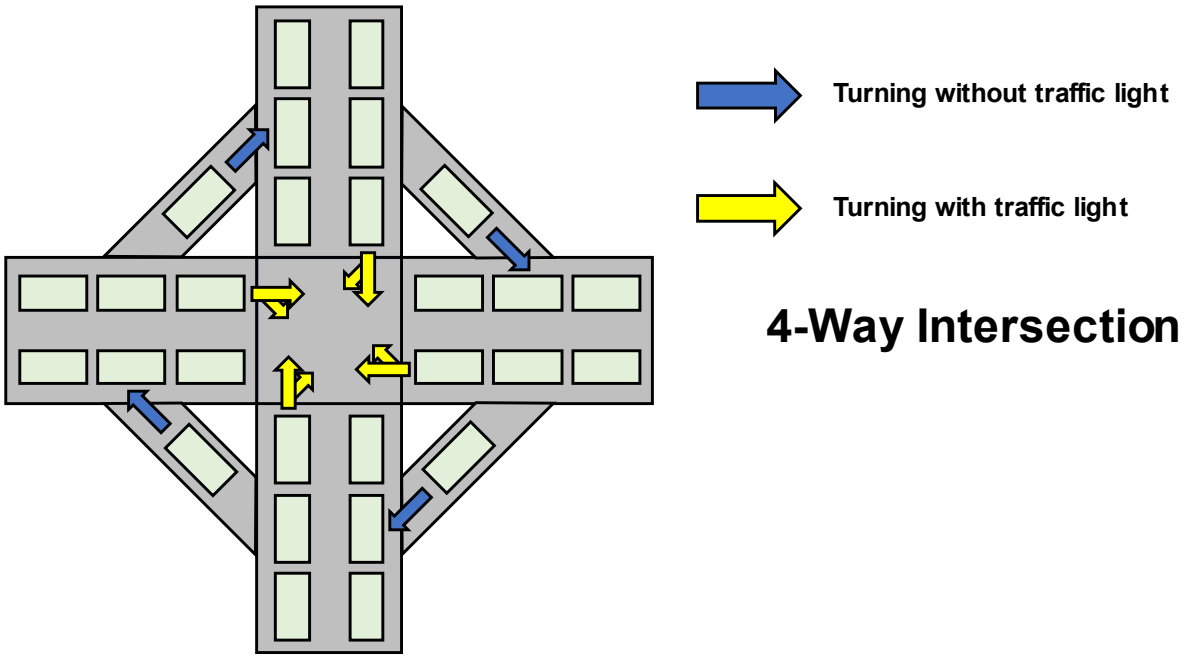
Cars and busses are not allowed to occupy an empty segment that are only occupied by 1 motorbike.

Traffic Light

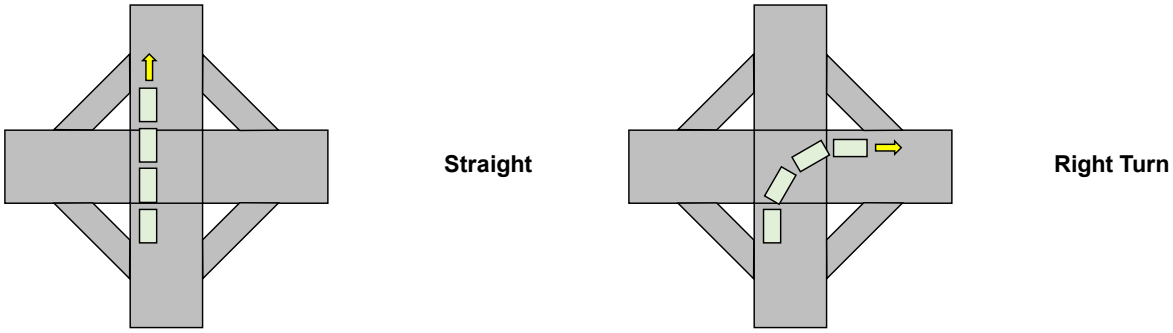
A traffic light is an object that controls the movement of vehicles in the road. There are 2 states that a traffic light can be in, `GO` and `STOP`. In `GO` state, vehicles are allowed to move freely to the direction. In `STOP` state, vehicles must stop all their movement.

Road Rules

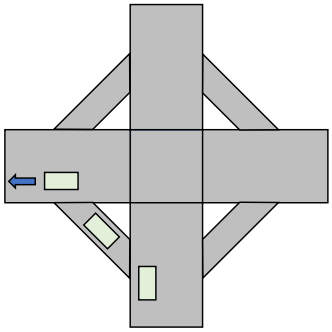
Intersection



At the end of an intersection, a vehicle can only move straight or turn right. It must wait for the corresponding traffic light to turn be at `GO` state before it can move. A vehicle will need to travel 2 units before reaching the next road.



For left turns, vehicle can turn a segment before the end of the intersection. There is no traffic light for left turn, and vehicle can turn at any time as long as it is empty. A vehicle will need to travel 1 unit before reaching the next road.



Left Turn

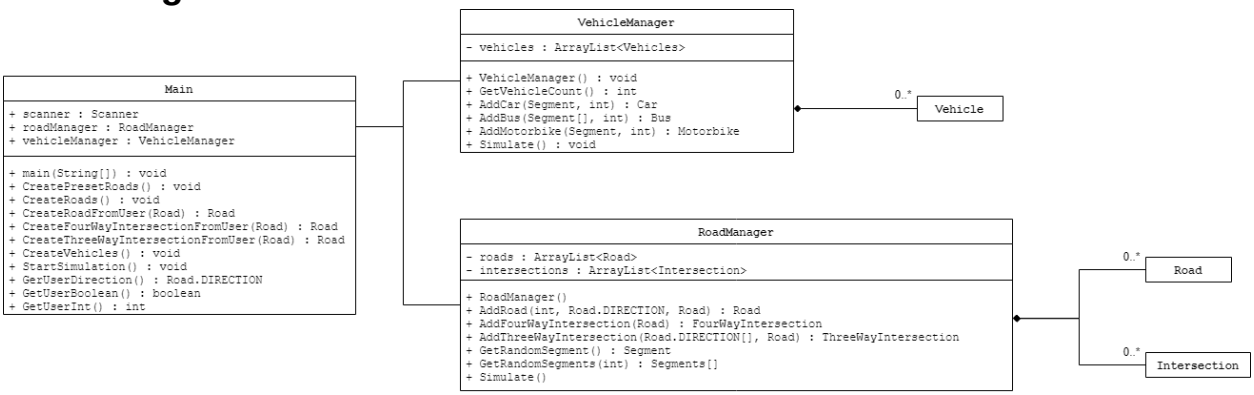
Traffic Light in Intersection

For traffic light in intersection, it will have two directions, straight and right. The traffic light will synchronize its timings with traffic lights that are directly opposite of it. The cycle of the traffic light will be:

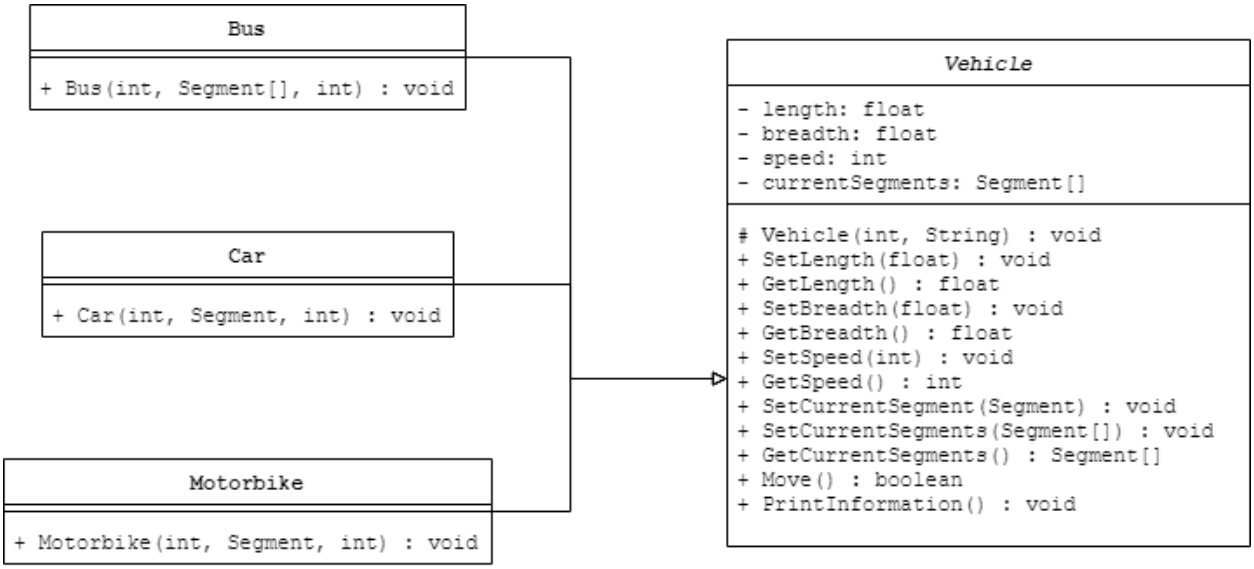
Cycle	**Straight**	**Right**
1	STOP	STOP
2	GO	STOP
3	STOP	GO

UML Diagram

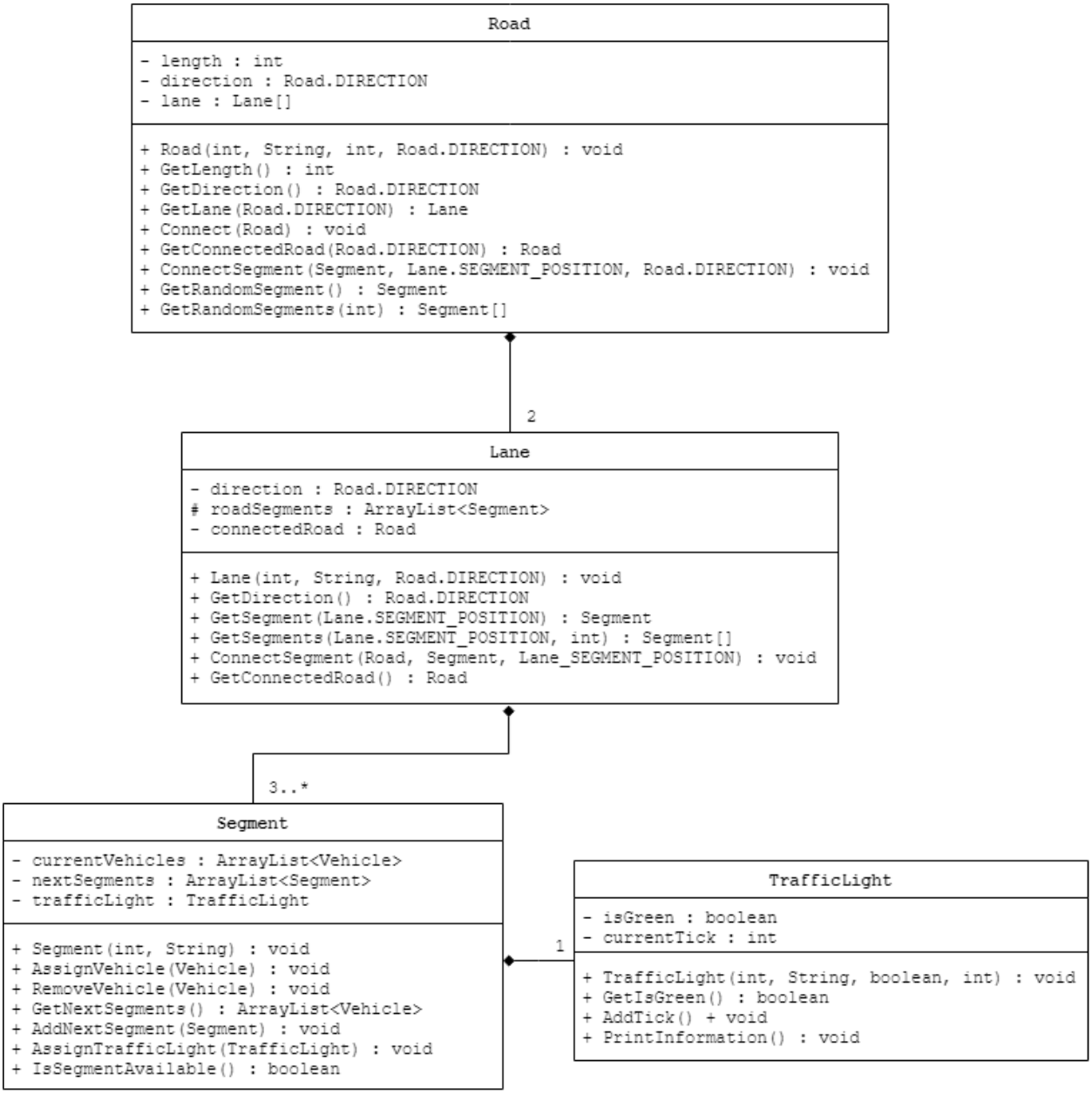
Main Program



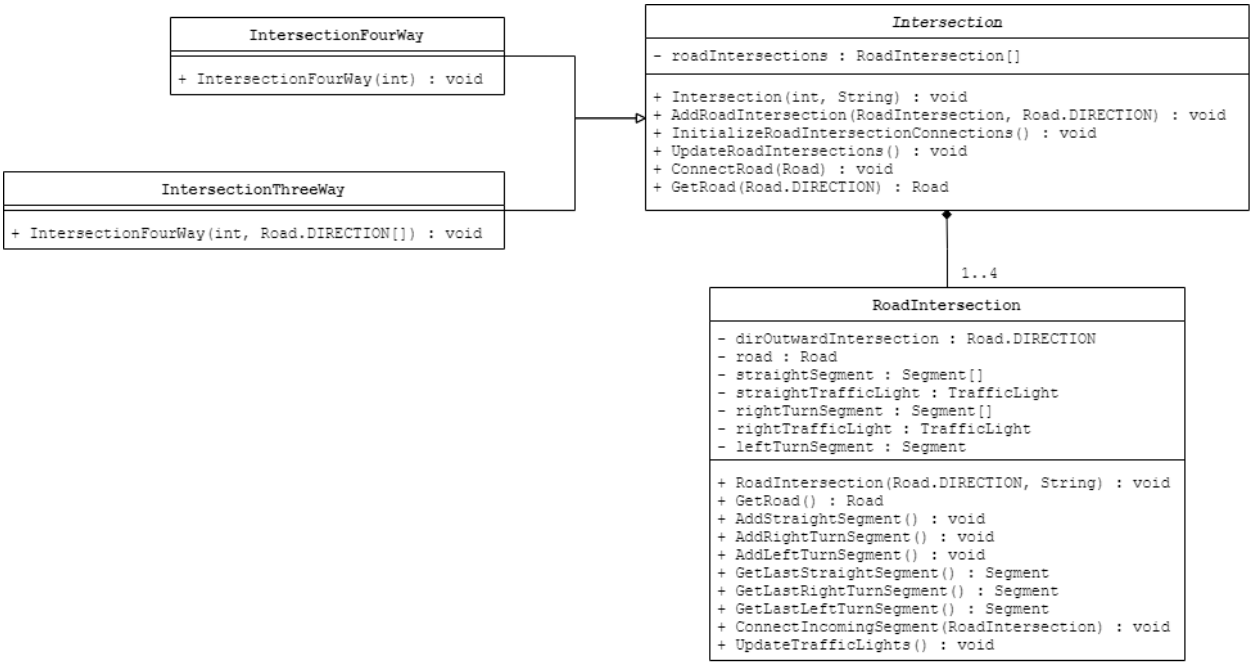
Vehicle



Road


Road
- length : int
- direction : Road.DIRECTION
- lane : Lane[]
+ Road(int, String, int, Road.DIRECTION) : void
+ GetLength() : int
+ GetDirection() : Road.DIRECTION
+ GetLane(Road.DIRECTION) : Lane
+ Connect(Road) : void
+ GetConnectedRoad(Road.DIRECTION) : Road
+ ConnectSegment(Segment, Lane.SEGMENT_POSITION, Road.DIRECTION) : void
+ GetRandomSegment() : Segment
+ GetRandomSegments(int) : Segment[]
2
Lane
- direction : Road.DIRECTION
roadSegments : ArrayList
- connectedRoad : Road
+ Lane(int, String, Road.DIRECTION) : void
+ GetDirection() : Road.DIRECTION
+ GetSegment(Lane.SEGMENT_POSITION) : Segment
+ GetSegments(Lane.SEGMENT_POSITION, int) : Segment[]
+ ConnectSegment(Road, Segment, Lane_SEGMENT_POSITION) : void
+ GetConnectedRoad() : Road
3..*
Segment
- currentVehicles : ArrayList<Vehicle>
- nextSegments : ArrayList
- trafficLight : TrafficLight
+ Segment(int, String) : void
+ AssignVehicle(Vehicle) : void
+ RemoveVehicle(Vehicle) : void
+ GetNextSegments() : ArrayList<Vehicle>
+ AddNextSegment(Segment) : void
+ AssignTrafficLight(TrafficLight) : void
+ IsSegmentAvailable() : boolean
1
TrafficLight
- isGreen : boolean
- currentTick : int
+ TrafficLight(int, String, boolean, int) : void
+ GetIsGreen() : boolean
+ AddTick() + void
+ PrintInformation() : void

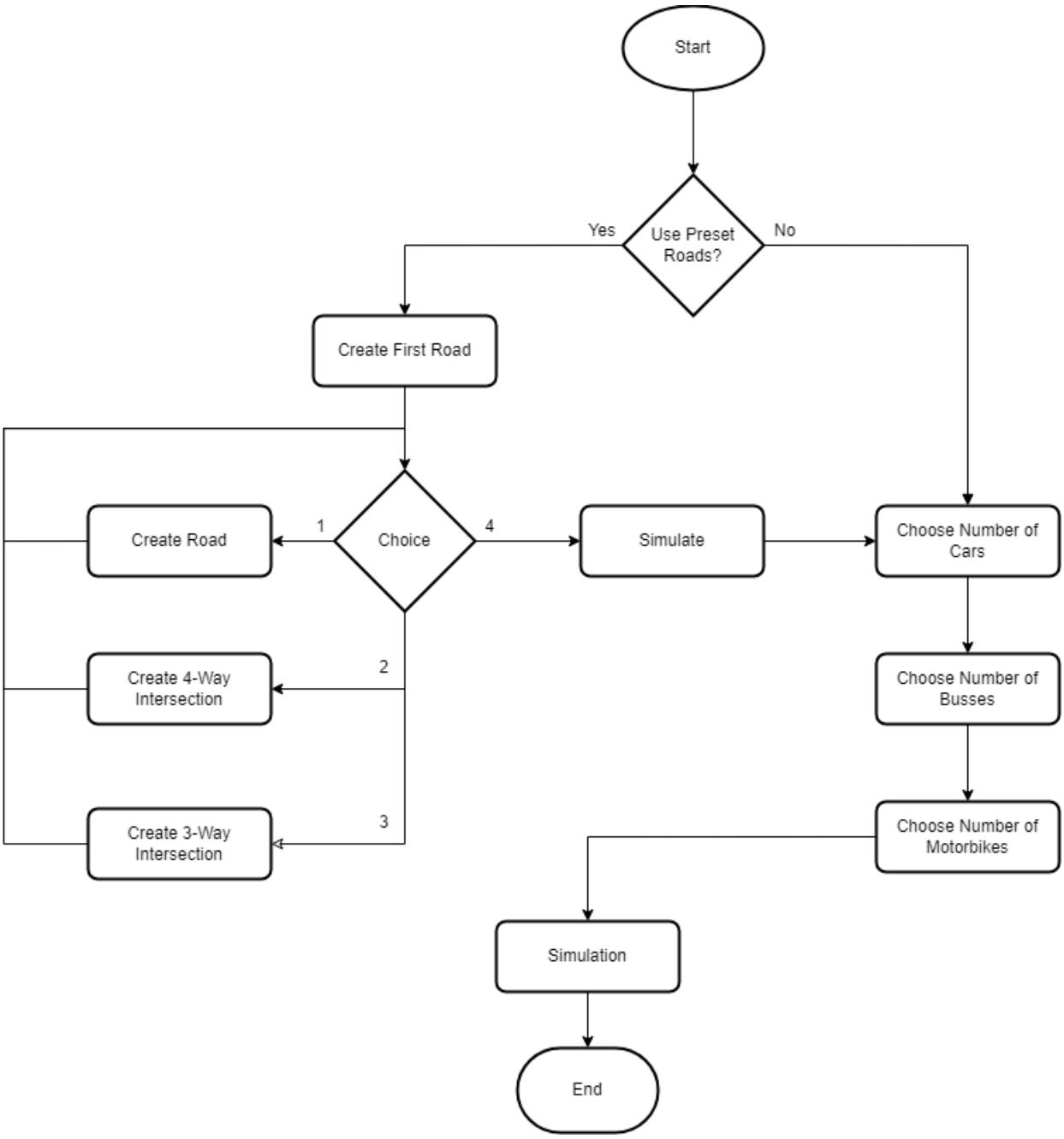
Intersection



Program

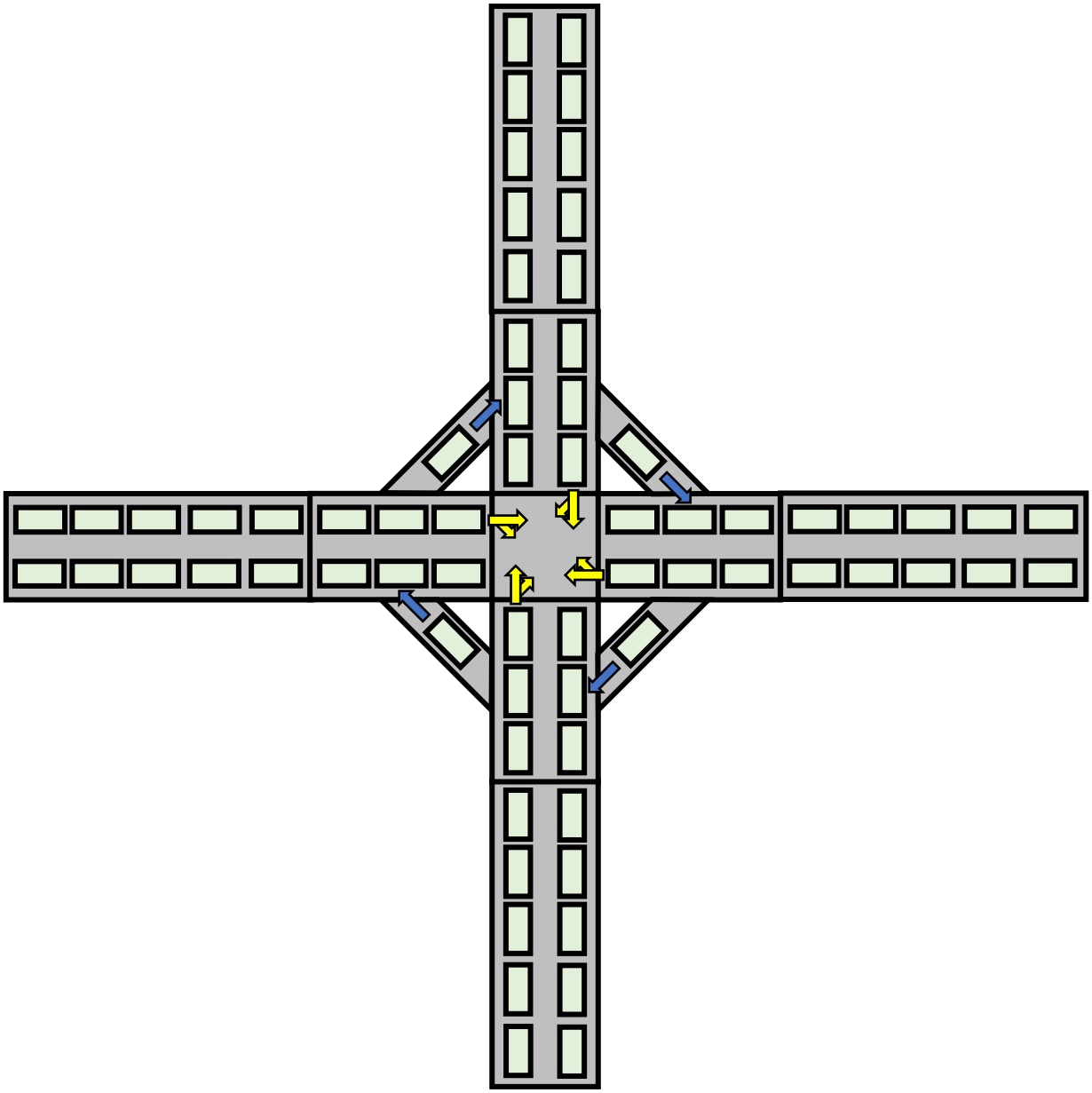
Flow

The flowchart of the program is:



Preset Road

The preset road will look like:



Output

Vehicle

A sample output of a vehicle will be:

`Car_0`

`Car` indicates the type of vehicle.

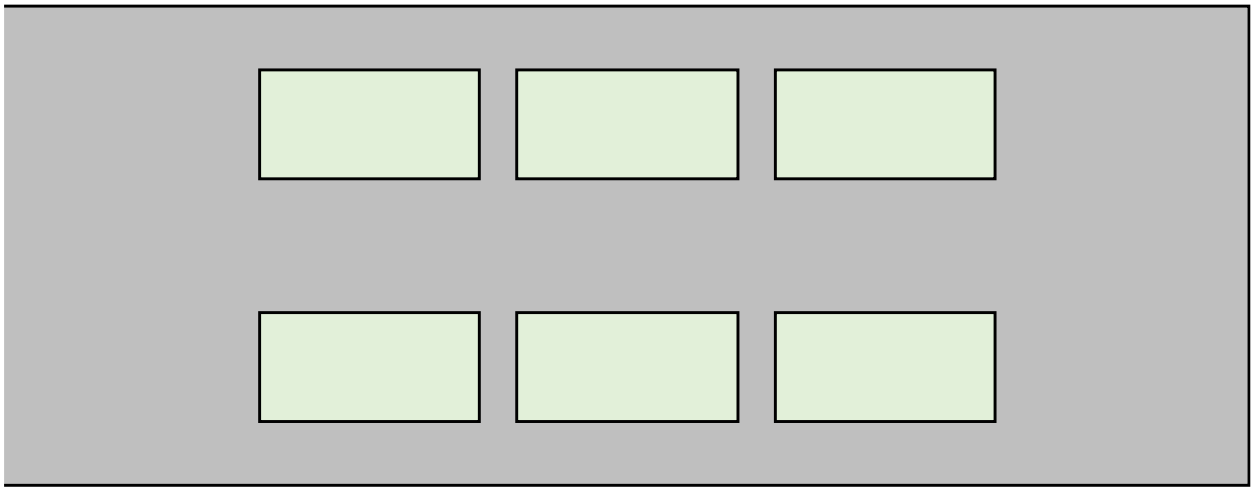
`_0` indicates the index of the vehicle.

Road

A sample output of a road will be:

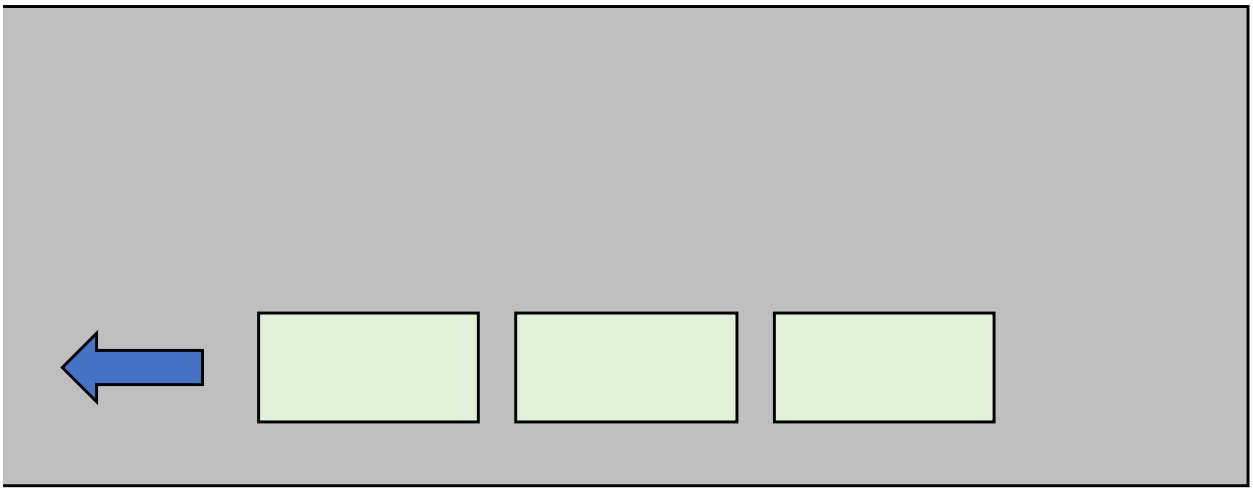
`Car_0 on Road_2-WEST_Lane-Segment_2`

`Road_2` indicates a road with index 2.



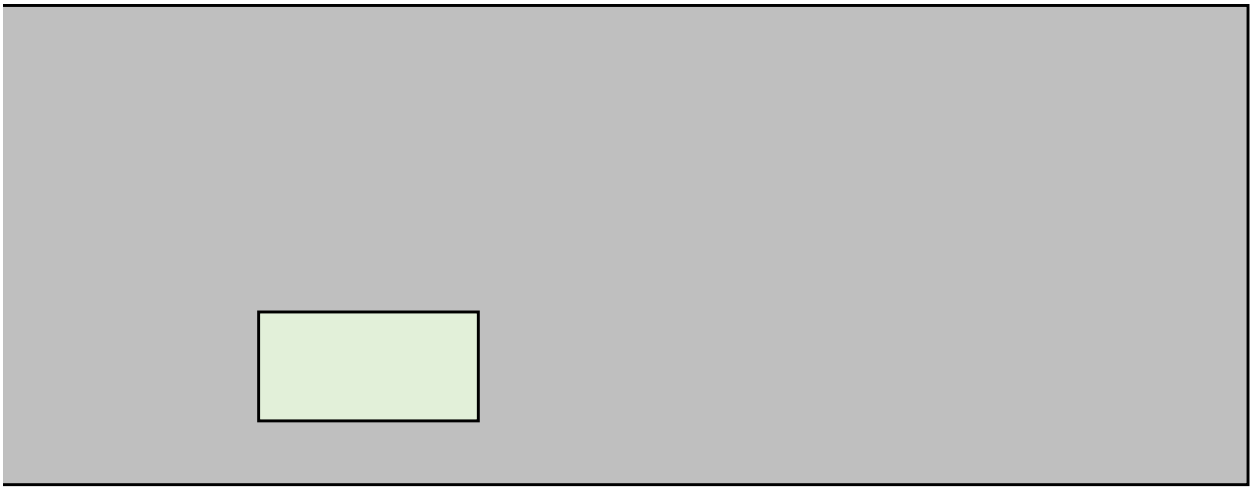
Road_2

`WEST_Lane` indicates that the vehicle is on the lane that is going west.



WEST_Lane

`Segment_2` indicates that the vehicle is on the segment with index 2 of the current lane.



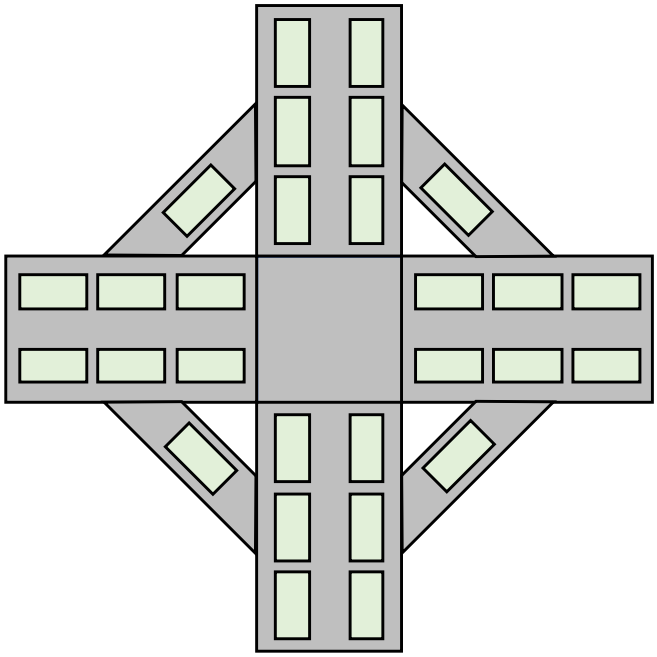
Segment_2

Intersection

A sample output of an intersection will be:

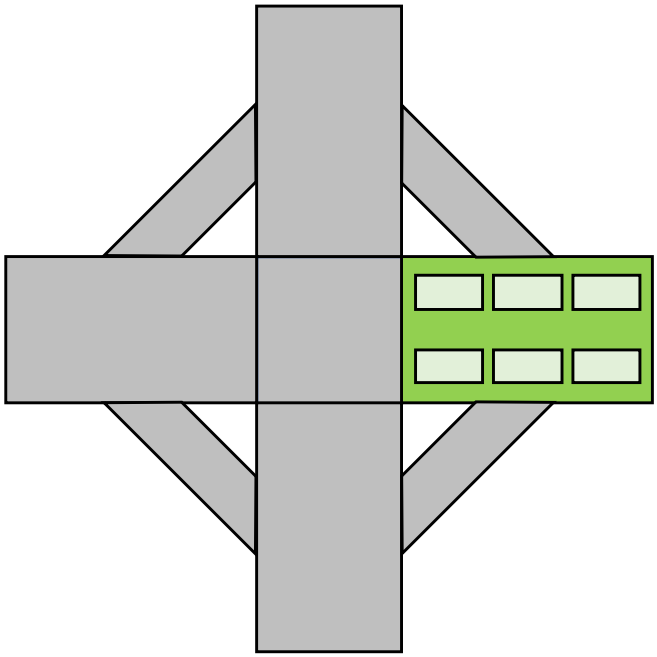
`Car_0 on 4WayIntersection_0-EAST_Road_1-WEST_Lane-Segment_0`

`4WayIntersection_0` indicates that the vehicle is on a 4-Way Intersection with index 0.



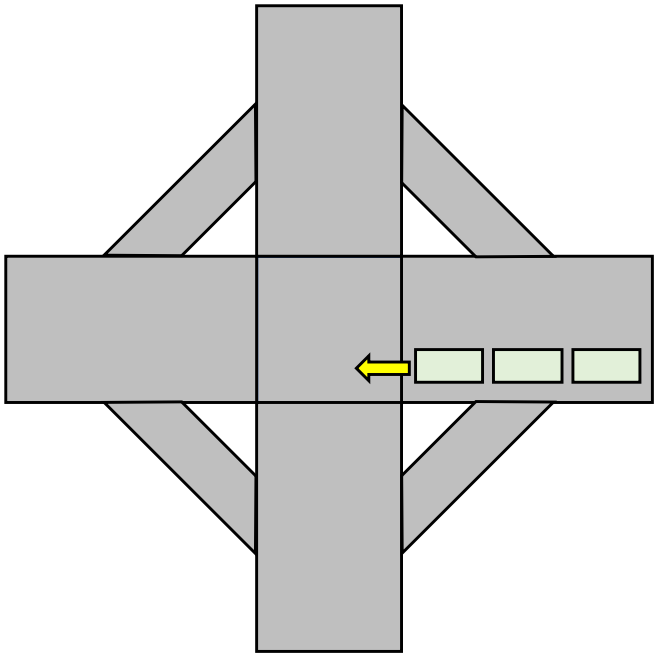
4WayIntersection_0

`EAST_Road_1` indicates that the vehicle is on the east side of the intersection.



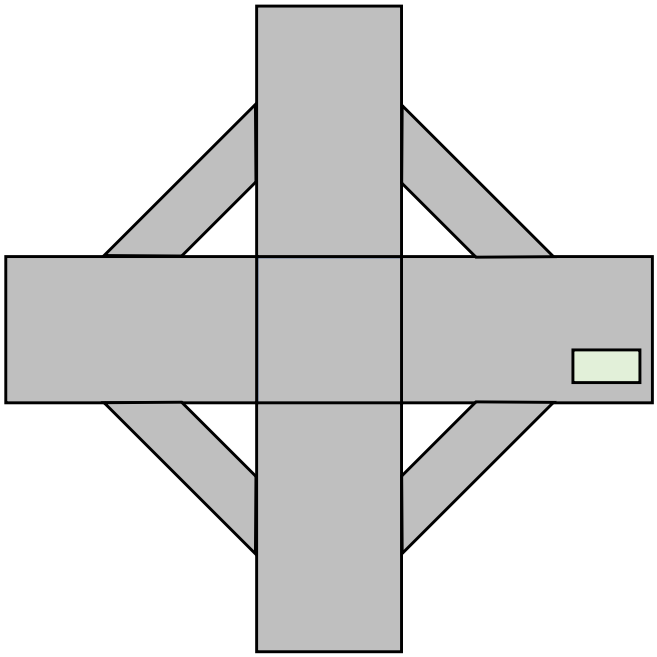
EAST_Road_1

`WEST_Lane` indicates that the vehicle is on the lane that is going west.

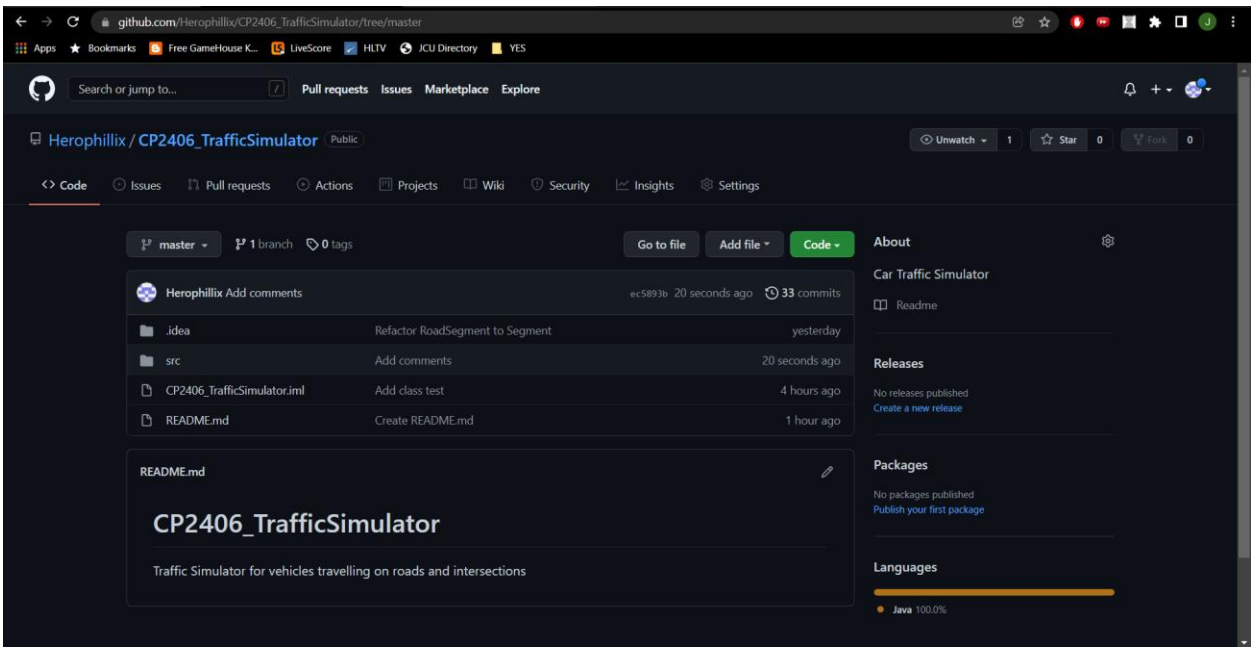


WEST_Lane

`Segment_0` indicates that the vehicle is on the segment with index 0 of the current lane.



`Segment_0`

GitHub Repositories

Link: https://github.com/Herophillix/CP2406_TrafficSimulator