Sequence Diagram and Class Diagram

**Group-B**

**Ishant Upadhyay  
Francky Ngabo  
Hamid Reza Heydari  
Raditya Surya Pratama  
Hasan Lapa  
Ji Nan**

Table of Contents

[Introduction 2](#_Toc414363566)

[Sequence Diagram 3](#_Toc414363567)

[1. Add Crossing 4](#_Toc414363568)

[2. Remove Crossing 5](#_Toc414363569)

[3. Adjust number of cars 6](#_Toc414363570)

[4. Adjust Green Time 7](#_Toc414363571)

[5. Checking lane availability 8](#_Toc414363572)

[6. Move pedestrians 9](#_Toc414363573)

[7. Load file 10](#_Toc414363574)

[8. Save file 11](#_Toc414363575)

[9. Start application 12](#_Toc414363576)

[10. Stop application 13](#_Toc414363577)

[11. Pause application 14](#_Toc414363578)

[12. Rotate map 15](#_Toc414363579)

[Class Diagram 16](#_Toc414363580)

[General : 16](#_Toc414363581)

[Detailed: 16](#_Toc414363582)

[Class Description 17](#_Toc414363583)

[1. Car 17](#_Toc414363584)

[2. Crossing 17](#_Toc414363585)

[3. TrafficControl 17](#_Toc414363586)

[4. Pedestrian 18](#_Toc414363587)

[5. TrafficLight 18](#_Toc414363588)

[6. WithPedestrian 18](#_Toc414363589)

[7. WithoutPedestrian 18](#_Toc414363590)

[8. Simulation 19](#_Toc414363591)

[9. Setting 19](#_Toc414363592)

[10. Lane 19](#_Toc414363593)

# Introduction

This document contains the structure of sequence diagram and class diagram. In order to know how the application it is very important to know the sequence diagram and its class diagram. Sequence and class diagram help programmer to develop the application with ease.

# Sequence Diagram

The list of sequence diagram:

1. Add Crossing
2. Remove Crossing
3. Adjust number of cars
4. Adjust Green Time
5. Checking lane availability
6. Move pedestrians
7. Load file
8. Save file
9. Start application
10. Stop application
11. Pause application
12. Rotate Map

## Add Crossing



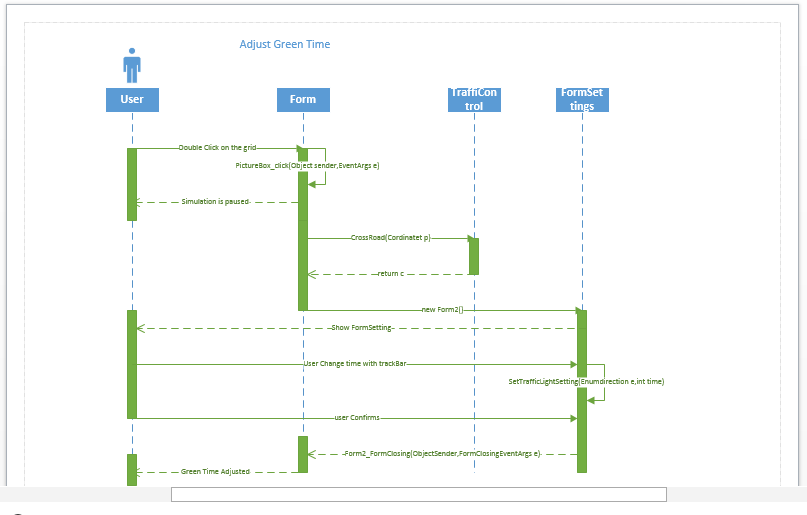
## Remove Crossing



## Adjust number of cars



## Adjust Green Time



## Checking lane availability



## Move pedestrians



## Load file



## Save file



## Start application



## Stop application



## Pause application

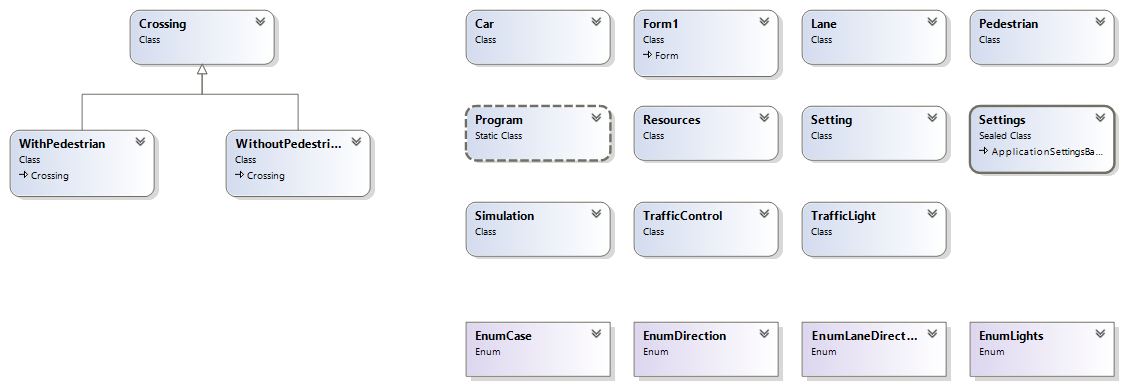


## Rotate map

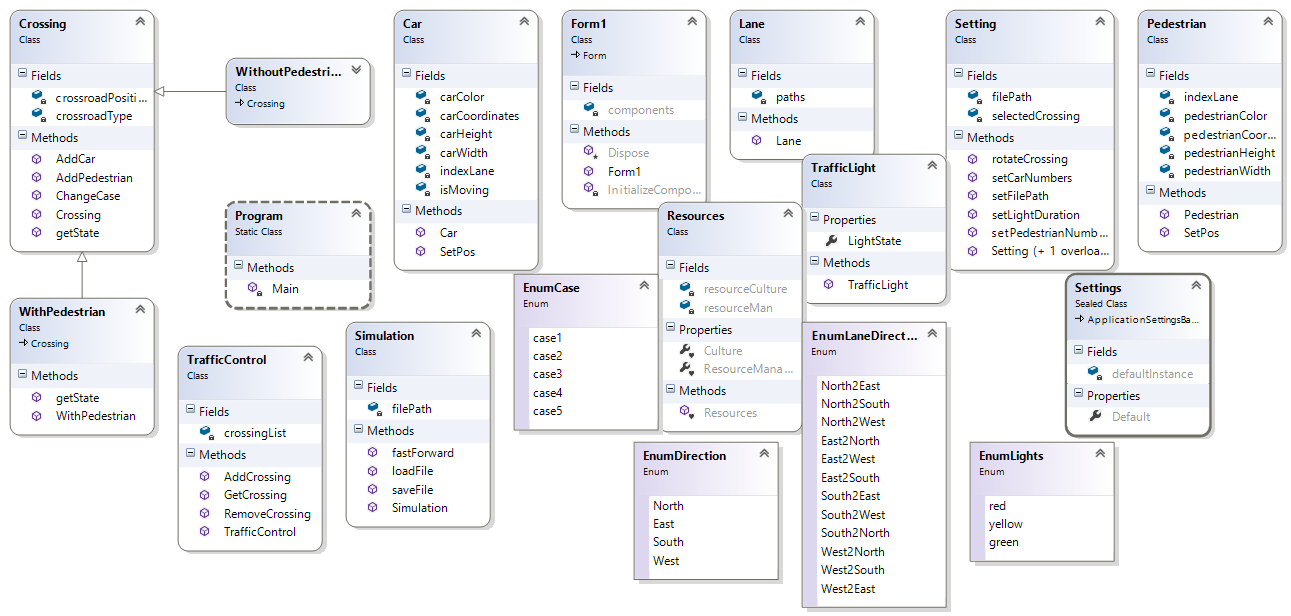


# Class Diagram

## General :



## Detailed:



# Class Description

## Car

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| Car() | Constructor of the class |  |
| setPos(Point loc) | Set new car position | New position of the car |

## Crossing

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| Crossing(int type, int row, int col) | Constructor of the class | 1. Type: type of the crossing 2. Row: row number of the crossing on the grid 3. Col: col number of the crossing on the grid |
| AddCar(EnumLaneDirection direction, Car c) | Add new car object to crossing selected lane | 1. Direction: crossing direction to add car 2. C: car object |
| AddPedestrian(Pedestrian p, int laneIndex) | Add new pedestrian object to crossing selected lane | 1. P: pedestrian object 2. laneIndex: pedestrian lane where pedestrian should be added |
| ChangeCase() | Change case |  |
| TrafficLight[] getState() | Return array of traffic light object states of the current crossing |  |

## TrafficControl

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| TrafficControl() | Constructor of the class |  |
| AddCrossing(int type, int row, int col) | Add new crossing to crossing Lists | 1. Type: crossing type 2. Row: row index position of crossing at the grid 3. Col: col index positon of crossing at the grid |
| RemoveCrossing(int row, int col) | Removes crossing object at defined row and column | 1. Row: row index position of crossing at the grid 2. Col: col index positon of crossing at the grid |
| Crossing GetCrossing(int row, int col) | Return crossing object at defined row and column | 1. Row: row index position of crossing at the grid 2. Col: col index positon of crossing at the grid |

## Pedestrian

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| Pedestrian() | Constructor of the class |  |
| setPos(Point loc) | Set new pedestrian position | New position of the car |

## TrafficLight

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| TrafficLight() | Constructor of the class |  |
| CheckState() | Return the state of traffic light object |  |

## WithPedestrian

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| WIthPedestrian(int type, int row, int col) : base(type, row, col) | Constructor of the class | 1. Type: crossing type 2. Row: row index position of crossing at the grid 3. Col: col index positon of crossing at the grid |
| Override TrafficLight[] getState() | Return array of traffic light object states of the current crossing |  |

## WithoutPedestrian

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| WIthPedestrian(int type, int row, int col) : base(type, row, col) | Constructor of the class | 1. Type: crossing type 2. Row: row index position of crossing at the grid 3. Col: col index positon of crossing at the grid |
| Override TrafficLight[] getState() | Return array of traffic light object states of the current crossing |  |

## Simulation

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| Simulation() | Constructor of the class |  |
| saveFile(string path) | Save the simulation object into binary file | Path: location to store the saved file |
| Simulation loadFile(string path) | Load the simulation object from a binary file at certain location | Path: location of the stored file |
| fastForward() | Fast forward the speed of simulation |  |

## Setting

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| Setting() | Constructor of the class |  |
| Setting(Crossing C) | Constructor of the class | C: selected crossing |
| SetFilePath(string path) | Set default saving file path | Path: location to store the saved file |
| RotateCrossing() | Load the simulation object from a binary file at certain location |  |
| setCarsNumber(string op) | Increase or decrease number of cars object | Op: operand ( increase or decrease) |
| setPedestriansNumber(string op) | Increase or decrease number of pedestrian object | Op: operand ( increase or decrease) |
| setLightDuration(int time) | Set the duration of traffic light | Time: desired time |

## Lane

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Parameter** |
| Lane(Point[] paths) | Constructor of the class | Paths: array of path point of the lane |