Console-based Coding Challenge

(Assessment Task 2) – Project Software Development Process

Jack Mooney - 13230559

# Problem Specification

To design and develop a GUI based traffic simulator that requires a set of vehicles (car, bus, motorbike), that moves in multiple directions on multiple roads and obeys the stop and go commands of multiple traffic lights. The program will be executed within a Main class and utilize a statistics class to keep track of the results of the simulation.

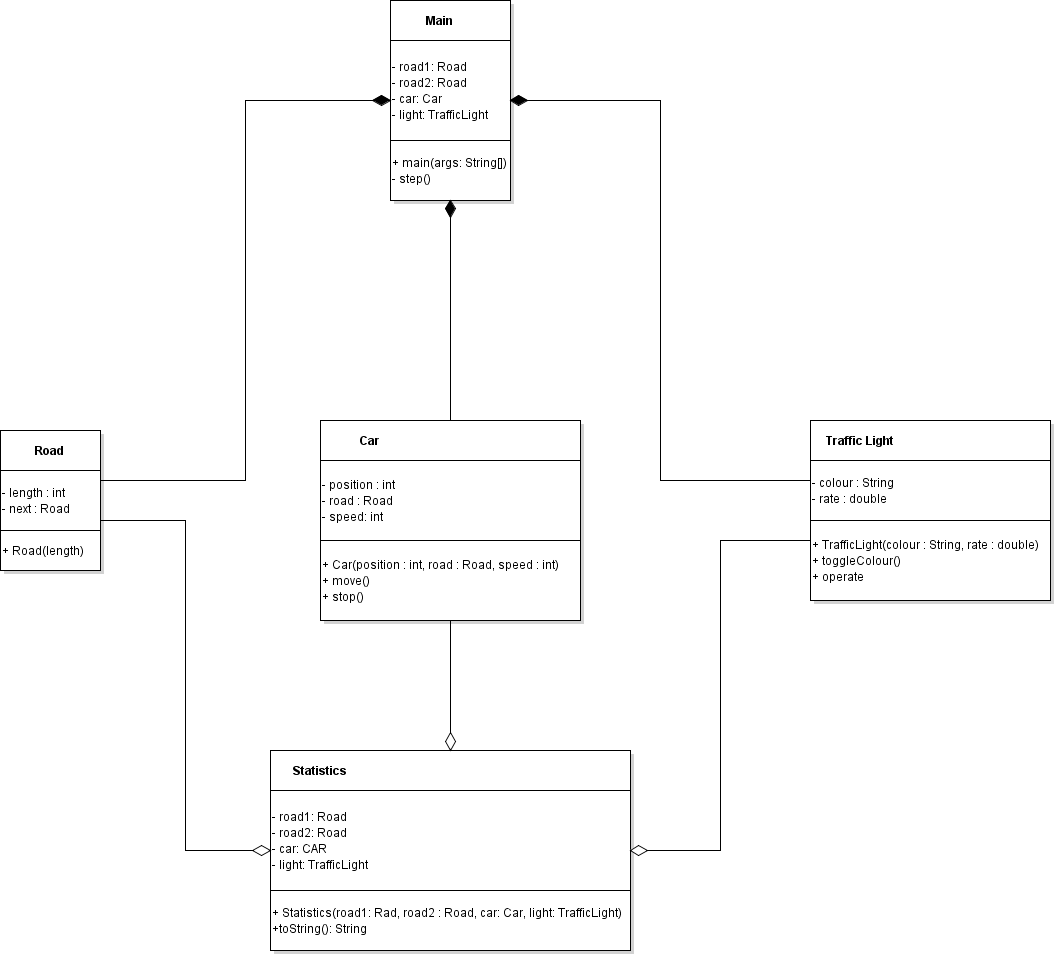
This program will be used during JCU open day in 2020. It will behave like a simplified traffic simulator.

# Problem Decomposition

This program will be constructed using five classes, in which comprises of five objects.

1. Main
2. Statistics
3. Road
4. Car
5. TrafficLight

# Class Design

* UML Diagram

## Main

This class is responsible for starting the execution of the program. It will create the instances of each object, define the values and provide the procedure for the simulation.

|  |  |
| --- | --- |
| **Attributes** | **Behaviors** |
| Defines the class objects: road1, road2, car & light | Main will execute the simulation.  Road1 & Road2 will have 8 segments and will connect to the next road.  The Car will start on the first road (road1), at the first position (0), at a maximum speed of 20kmph.  The TrafficLight initial state is green and has a timed cycle rate of 0.03 to turn red.  The statistics passes road1, road2, car and light objects and prints the current conditions of these objects.  The method will call the statistical formula object (stats) and perform the 100 step simulation with an incremental time delay of 1 second.  Step() entails the conditions for when the car will go or stop based upon the state of the traffic light. |

## Statistics

This class defines the instructions for the print statement by returning the values of the simulation into a readable string format.

|  |  |
| --- | --- |
| **Attributes** | **Behaviors** |
| Defines unique instances of: road1, road2, car & light | Statistics creates unique instances of objects to and passes them to a string builder to display the print information.  toString() is the method used to be able to convert the values into a string format for the print statement. |

## Road

This class is responsible for constructing the road/boundaries that the vehicle object will travel on. Each Road object stores a set integer value.

|  |  |
| --- | --- |
| **Attributes** | **Behaviors** |
| The length of the road is defined as an integer with a set value (8).  The road(1) will also be connected to the next road(2) | Set the length of the road to a defined integer value. |

## Car

This class defines the conditions of the vehicle. This class stores the value of the car’s position on the road, the current road the car is travelling on and the speed at which the car is travelling.

|  |  |
| --- | --- |
| **Attributes** | **Behaviors** |
| Defines what segment the car is positioned at in relation to the segmental length of the road.  Defines what road the car is on.  Defines the speed of the car. | Starts the car on the first position of road1 travelling at 20kmph  Move()  Moves the car to the next position on the road. If the current position of the car is at the last position on the current road, then the current position is reset to 0. If the current road reaches the next connected road segment then this function moves the car to the next segment.  Stop()  Sets the cars speed to 0kmph. |

## TrafficLight

Defines the condition that changes the behavior of the vehicle to go or stop. The traffic light is positioned on the last segment of the first road.

|  |  |
| --- | --- |
| **Attributes** | **Behaviors** |
| Has two primary colors: Green & Red.  Defines a rate for the colour time cycle. | Get and set the Colour and cycle rate of the light.  toggleColour() Changes the state of the traffic light.  Operate()  Makes the traffic light change if the random value is less than the value of the cycle rate. |