## CP2406 – Traffic Simulator Working Document – Rhys Selwyn 13182099

### Context:

A customer has asked for a traffic simulator to be designed in java. Due to the nature of the project we are only presenting a basic prototype of the system as a demonstration of the company’s ability before undertaking the full project. The basic functionality will include two roads, one car and a traffic light and will simulate movement between.

### Technical Architecture:

There are multiple parts of designing the basic functionality which leads to the need of break down each of the classes and interfaces used to describe the functionality for the user. The classes used will be public as to allow for ease of coding when creating objects inside of objects such as vehicle object inside road object etc. All the classes below have been tested using unit test classes and run correctly and the main interface is also working as intended. Each class is broken down below:

### Vehicle Class:

The vehicle class aims to represent an object of a car, motorbike or bus from the specifications given. It is used as a parent class so that each different vehicle may inherit its methods and variables. The vehicle class will be passed a car length which will determine the length of all the vehicles involved in the simulation.

Constructor:

* initialX -records where the initial back of the car is in the x plane
* initiallY -records where the initial back of the car is in y plane
* direction -records the current direction to allow for changing of the direction and therefore causing the car to drive in a different way
* currentRoad – records the road id that it is currently on, can be edited to allow for changing of road

Methods:

* initiateFrontofVehicle – this method will change the variable currentX1 or currentY1 to the x or y position of the front of the vehicle depending on the direction of travel
* drive – this method will only currently provide functionality in the East direction and will move by the set speed given by the vehicle

Motorbike Subclass:

* Constructor same as vehicle class
* Length = car length/2
* Width = motorbike length/2
* Speed = 2
* Can change speed using set motorbike function

Bus Subclass:

* Constructor same as vehicle class
* Length = car length\*3
* Width = bus length/2
* Speed =1
* Can change speed using set bus speed function

Car Subclass:

* Constructor same as vehicle class
* Length = car length
* Width = car length/2
* Speed =2
* Can change speed using set car speed function

### Road Class:

Constructor:

* roadLength – records the length of a road object given which is to be between 6 and 15 times the length of a car
* initialX1 – records where the road is positioned x plane
* initialY1 – records where the road is positioned y plane
* String orientation – gives off either hortizontal or vertical road which allows for the creation of a road in a certain orientation by making the finish point + roadlength on either x or y axis
* Identifier – records the identifier of the road object so it can be accessed for methods to change roads
* touchingRoads – records the identifier of the next road object which touches the currentRoad so that a vehicle can change roads if needed

Methods:

* checkRoadEnd – obtains the orientation of the road and then finds where the road end will be in the x or y plane
* initiateVehiclesFrontpoint – intiates a vehicles front point in the array list vehicles
* add car , add bus, add motorbike,add trafficlight – these functions allow for traffic lights or vehicles to be spawned on the road and then added to the vehicles or traffic lights arraylist

### Traffic Light Class:

Constructor:

* default light colour = ‘green’
* xpos – records where the light will be placed in x plane
* y pos records where the light will be place yplane

Methods:

* operates – obtains random double value from 0-1 and then compares it to a rate of change and if below or equal to this value it will change the lights colour

### Main Interface:

Initialiser block:

* Run a non static initialiser block to allow for roads to be spawned and then added to a road array list, due to their only being on main object created it won’t overwrite the roads data each time if they are changed

Methods:

Currently there are no methods used within the main class and just a large block of code however in future implementation a simulator window will run while simulateRoads = true and then the system will keep running until the size of each roads vehicle list is equal to zero. The multiple for loops in the main statement will be turned into a drive function separate to the main calling in future implementation.