***Working Document***

The client needs a traffic simulator; a way to simulate real life city road/car traffic. The program will be tailored to meet these needs, through the use of console, where it would allow user input, notably in creating the setting for the simulation itself. However, this version of the program is only able to simulate a single car moving along a straight road with a single traffic light.

This problem can be separated into class objects for the program and simulation that interact with each other and the input, including:

Car

The Car class object defines the average road vehicle. Could be a normal car, a bus, a motorcycle, a truck, or the like.

Contains the following attributes:

+ id (string): *the car class object’s own unique identifier*

+ length (string): *the physical space the vehicle occupies parallel to the road.*

+ breadth (integer): *the physical space the vehicle occupies perpendicular to the road.*

+ speed (integer): *the car’s travel speed relative to the simulation*

+ position (integer): *where the car is, relative to the road*

+ road (road): *the road that the car is on*

The car will be able to move along the road at the road’s speed limit through the use of the move() method, interacting with whatever is in the path (such as traffic lights and, later, other cars. By interaction, this mostly means the car may stop moving in response to a red light or, similarly, another car that is stationary). The car will also be in control of how it is represented graphically with its draw() method.

Road

The Road class object defines a single lane of road

Contains the following attributes:

+ id (string): *the road class object’s own unique identifier*

+ length (integer): *the amount of “road” that the cars can travel on*

+ width (integer): *the amount of “lanes” for cars to stay side by side*

+ speed\_limit (integer): *the maximum speed cars are to have obligated by law*

+ start\_position (integer[]): *the starting co-ordinate point of the road*

+ end\_position (integer[]): *the end co-ordinate point of the road*

+ orientation (VERTICAL or HORIZONTAL): *which way the road is facing*

+ cars (Array of Cars): *list of Car class objects placed on the road*

+ lights (Array of TrafficLights): *list of TrafficLight class objects placed on the road*

+ roads (Array of Roads): *list of Road class objects connected to this one*

The length of the road will be variable depending on user input. Roads will interact with other roads by being connected to them, have the ability for multiple connections, creating a very basic intersection with multiple options for the car to take. Cars will move along the road from the starting position (1) to the end position, that depends on the road’s length. Traffic lights can be placed at only the end position of the road. The road will also be in control of how it is represented graphically with its draw() method, depending on its orientation.

TrafficLight

The TrafficLight class object defines the simple red/green (no yellow) traffic light

Contains the following attributes:

+ id (string): *the TrafficLight class object’s own unique identifier*

+ state (boolean): *whether the light is green (1, True, on) or red (0, False, off)*

+ position (integer): *where the traffic light is, relative to the road*

+ road (road): *the road that the traffic light is on*

The traffic light will operate, using the lights() method. Randomly changing from green to red. This operation will be based on pseudo-random numbers generated by the program. The light will be placed on a road only at its final position, the end of the road, and will interact with vehicles that are also at that position. If the light is red the vehicles will stop and not move to the next road. If the light is green the vehicles will continue past and move to the next road. it will also be able to control how it is drawn graphically with the draw() method.

Simulator

The Simulator class generates the graphical display of the simulation on the screen

+ cars\_spawning (integer): *total numbers of cars to be spawned in the simulation*

+ cars\_spawned (integer): *how many cars have been spawned so far.*

+ cars\_gone (integer): *cars that have been removed due to reaching the end of the simulation*

+ cycles (integer): *the number of simulation cycles between spawns*

+ speed (integer): *how fast the simulation goes*

This class will load the location of road and traffic light objects from the editor. Cars will be created for the simulation here and the speed of the simulation is also set here depending on user input form dialog boxes in main. The simulation will be run within a timer.

Editor

This class allows the user to edit the graphical simulation, such as placing or removing roads

Main

The program will also have a/the Main() class, which is where the whole simulation takes place, including implementations of the class objects