CP2406 ASSESSMENT TASK 1 WORKING DOCUMENT

# Problem Specification:

The problem is to create a working traffic simulator. The simulator will simulate roads, traffic lights, and vehicles (cars, buses and motorbikes).

# Problem Decomposition:

## Class Design:

Main will act as the simulator and will create each object and call their methods when appropriate. It will also create a new instance of the Frame class and use it to display the GUI. The member fields needed are; timer, road, road2, road3, roads, car, trafficLight. These member fields will be private as they will only be used by Main. The methods of this class are; Main(), animate(), paintComponent(), main(). These will be public.

Frame will act as the view model for the GUI. There will be no member fields for this class, but there will be two methods; Frame() and paint(). These will be public since they will be used in Main.

The Shape class will be abstract and be used as a parent class. The member fields needed are; x, y, color. These will be private as they will only be accessed within its own class and its child classes. The methods are; Shape() and paintComponent(). These will be public since they will be accessed outside of itself.

Objects of the Road class have the role of having a car drive over them and having a traffic light at either the start of the road or the end of the road. The member fields needed are; height, width, is\_horizontal, x, y, color. These member fields will be private as they are not being accessed outside of the class. The methods of this object that are needed are; Road(), setDimensions(), getX(), getY(), getHeight(), getWidth(), getOrientation(), paintComponent(). These methods are to be public since they will be accessed in the Main class. The Road class will extend the Shape class.

Objects of the TrafficLight class have the role of being next to a road and changing its state. The member fields need are; state, change\_rate, width, height, x, y, color. These will be private as they won’t be accessed outside of itself. The methods needed are; TrafficLight(x, y), paintComponent(g), setState(), getX(), getY(), getWidth(), getState(), getColor(). These will be public as they are used outside of itself.

The Vehicle class will be abstract and used as a parent class. The member fields needed are x, y, xDir and yDir. These will be private since the only classes using them will be itself and its child classes. The methods needed will be Vehicle(), move(), paintComponent() and update(). These will be public since they will be accessed outside of itself and its child classes.

Objects of the Car class have the role of driving on a road and through a traffic light when the traffic light is green but stopping when it is red. The member fields needed are; x, y, xDir, yDir, width, height, color, is\_horizontal. These will be private since they won’t be used outside of itself. The methods needed are; Car(), move(), setxDir(), setyDir(), getWidth(), getHeight(), getPosX(), getPosY(), getColor(), getOrientation(), get\_direction(), update(), resetCar(), paintComponent(). These methods are to be public since they will be used outside of itself.

## Method Design:

The method signatures for Main will be; Main(), animate(), paintComponent(Graphics), main().  
 - Main will be used to initialize the Main class.  
 - animate() will be used to animate the car moving along the road, as well as changing the trafficLight’s state.  
 - paintComponent(Graphics) will be used to paint each object.  
 - main() will be used to initialize a new instance of Frame.

The method signatures for Frame will be; Frame() and paintComponent().  
 - Frame() is used to initialize Frame and set up the GUI.  
 - paintComponent(Graphics) is used to paint everything.

The method signatures for Shape will be Shape(int, int) and paintComponent(Graphics).  
 - Shape(int, int) is used to initialize Shape and its child classes.  
 - paintComponent(Graphics) is used to paint itself.

The method signatures for Road will be Road(int, int, Boolean), setDimensions(Boolean), getX(), getY(), getHeight(), getWidth(), getOrientation(), paintComponent(Graphics).  
 - Road(int, int, Boolean) is used to initialize a road object by setting its variables.  
 - setDimensions(Boolean) is used to display the road as be horizontal or vertical.  
 - getX() is used to return the current x value.  
 - getY() is used to return the current y value.  
 - getHeight() is used to return the height of the road.  
 - getWidth() is used to return the width of the road.  
 - getOrientation() is used to return whether or not the road is horizontal.  
 - paintComponent(Graphics) is used to draw the road based on its color and its orientation.

The method signatures for TrafficLight will be TrafficLight(int, int), paintComponent(Graphics), setState(), getX(), getY(), getWidth(), getState(), getColor().  
 - TrafficLight(int, int) is used to initialize an object of the TrafficLight class by setting all the variables.  
 - paintComponent(Graphics) is used to draw the trafficLight based on its color and dimensions.  
 - setState() is used to set the state of the trafficLight based by getting a random number and comparing it to the change\_rate. Then the color will change if the random number is greater than the change rate.  
 - getX() is used to return the current x value.  
 - getY() is used to return the current y value.  
 - getWidth() is used to return the width of the trafficLight.  
 - getState() is used to return the current state of the trafficLight.  
 - getColor() is used to return the color of the trafficLight based on the state.

The method signatures of Vehicle will be Vehicle(int, int), move(), paintComponent(Graphics) and update(TrafficLight, Boolean, Road[]).  
 - Vehicle(int, int) will be used to set the default value to each member field.  
 - move() will set the x and y values based on the xDir and yDir variables.  
 - paintComponent(Graphics) is used to draw itself.  
 - update(TrafficLight, Boolean, Road[]) is used to update the car based on the TrafficLight, its orientation and the current road.

The method signatures for Car will be Car(int, int, int, int), move(), setxDir(int), setyDir(int), getWidth(), getHeight(), getPosX(), getPosY(), getColor(), getOrientation(), get\_direction(), update(TrafficLight, Boolean, Road[]), resetCar(), paintComponent(Graphics).  
- Car(int, int, int, int) is used to initialize all the member fields with default values.  
- move() will set the x and y values based on the xDir and yDir variables.  
- setxDir(int) is used to set the xDir based on the int that is passed in.  
- setyDir(int) is used to set the yDir based on the int that is passed in.  
- getPosX() is used to return the current x value.  
- getPosY() is used to return the current y value.  
- getHeight() is used to return the height of the car.  
- getWidth() is used to return the width of the car.  
- getColor() is used to return the color of the car.  
- getOrientation() is used to return whether or not the car is horizontal.  
- get\_direction() is used to return a random direction.  
- update(TrafficLight, Boolean, Road[]) is used to update the car based on the TrafficLight’s x position, the TrafficLight’s state, whether or not the car is horizontal and the variables of the road, and the direction it wants to travel based on the output from get\_direction().  
- resetCar() is used to reset the car’s position when it reaches the end.  
- paintComponent(Graphics) is used to draw the car based on its color and its orientation.

