University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: MASTER'S IN COMPUTER SCIENCE	Spring semester, 2013 Open access
Writer: ABENA KWANTWIWAA **OCRAN**	..... (Writer's signature)
Faculty supervisor: PROFESSOR REGGIE **DAVIDRAJUH**	
Title of thesis: MODELLING OF URBAN TRAFFIC NETWORK OF SIGNALIZED INTERSECTIONS	
Credits (ECTS): 30	
Key words:	Pages: 92 + enclosure: CD Stavanger, 14th June 2013 Date/year

Front page for master thesis
Faculty of Science and Technology
Decision made by the Dean October 30th 2009

ABSTRACT

This report presents how traffic network of signalized intersection in a chosen urban area called Tema is synchronized. Using a modular approach, two different types of traffic intersection commonly found in an urban area were modelled i.e. a simple intersection and a complex intersection. A direct road, even though not an intersection, was also included in the modelling because it's commonly found in an urban area plus it connects any two intersections. Each of these scenarios was modelled using a Petri Net modelling tool called GPenSIM. In real life, every traffic network has a control system, in order to emulate this real-world environment; a GUI was developed in JAVA with methods which are called in GPenSIM to enable user's control each of the traffic models using buttons. The GUI also shows a graphical representation of a traffic light for all three scenarios. Outputs from the simulation are displayed on the GUI through change in colours of the lights. Results show that Petri Net is a simple and easy tool to use in modelling real world environment. It also shows that the traffic network for any given area can be simulated in real-time making it easier and possible to test and analyse any given intersection before a traffic light system is built or already existing traffic light system can be improved upon to optimize traffic control.

Table of Contents

ACKNOWLEDGEMENT

I will like to use this opportunity to first thank God for all the grace and strength He has given me to come this far. I will also like to thank my supervisor Professor Reggie Davidrajuh for all his help, advice and encouragement. I thank Joshua Tetteh for designing the background image for me and finally I say thank you to everybody who has helped or encouraged me throughout my Master program here in Norway.

Chapter One

INTRODUCTION

This paper is a final year thesis report for Computer Science, faculty of Electrical Engineering at the University of Stavanger.

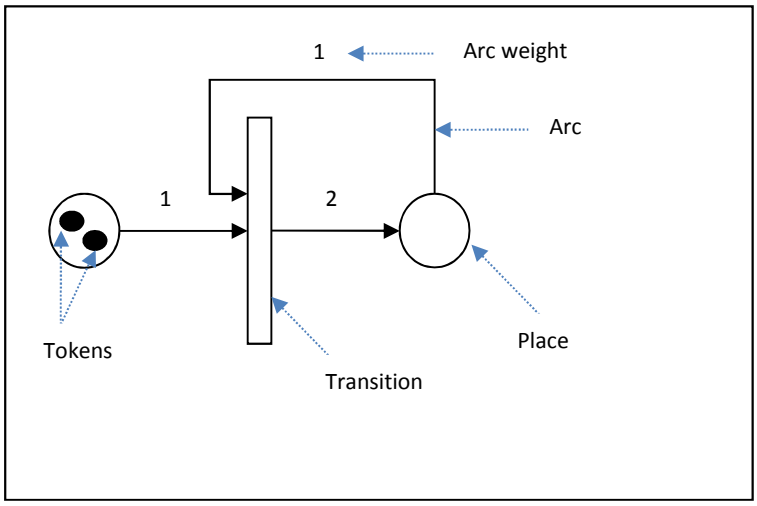
Background

Traffic Light system

According to [1], a traffic light is a visual signal (a system of coloured lights) for controlling traffic. It was first known to be used in 1912. An electric lamp that usually has a red, a green, and a yellow light and that is used to control traffic is called a traffic light [2]. [3]Defines traffic light as a set of automatically operated colored lights, typically red, amber and green for controlling traffic at road junction crosswalks.

Petri Net

It is a tool used for the study of systems through modelling, simulation and performance evaluation, of discrete-event dynamic systems. Analysis of the Petri net reveals important information about the structure and dynamic behaviour of the modelled systems [4]. It consists of places (passive elements), transitions (active elements) and tokens (elements which are subject to change and can be found only in places) [5]. Arcs connect places and transitions. Elements of a Petri net are shown in the figure below.



GPenSIM

GPenSIM (General purpose Petri net Simulation) is a software tool developed in MATLAB, to allow users to module and simulate events. According to [6],the reason for developing a new simulator are for flexibility, extensibility and ease of use. Most people do not like developing in mathematics or with mathematical software. Abstracting these mathematical details and providing graphical user interface for users makes it easy to use. GPenSIM was written to allow seamless integration with the other toolboxes that are also available in the MATLAB environment[6].

Modelling and Simulation in GPenSIM

Developing a Petri net model in GPenSIM consists of two steps. In the first step one has to define the static Petri net graph in one or more Petri net Definition Files (PDF).It is identifying basic elements of the Petri net and connecting them with arcs. The second step is assigning the dynamics of a Petri net in a Main Simulation File (MSF).Initial markings are assigned to places and firing times to transitions. When these files are created, the MSF file can be run to simulate the model [6].

Problem Definition

Tema is an urban area found in Greater Accra, Ghana. According to [7], it has an urban population of 392,044. Greater Accra region has a population growth rate of 0.5%, hence the area is getting chocked with individuals by the day. Increase in population yields in increase in transportation demands. Due to lack of good public transportation, people are compelled to find their own means of transportation by buying cars, motorcycles, bicycles etc.

An average working family in Tema has a minimum of 2 cars. This leads to increase in the number of vehicles on the roads. Poor planning and development in road and transport makes it difficult to expand the roads to accommodate these increasing numbers. A normal journey that takes about 30 to 45minutes can go up to hours due to non-existence of traffic lights or dysfunctional ones. Drivers take illegal routes to join main roads causing users who have the right of way to be stuck in traffic for hours. This cause's major traffic jams on main roads. The issue of road expansion is a whole topic on its own which is not the main focus here. The focus here is how to control the proper and direct flow of these heavy traffic jams.

There are no proper road systems designs for pedestrians or cyclist therefore they all use road without any fear of vehicle passing. It has gone so bad that people run off the road only when it's about to rain and care less about going off the road if it's a vehicle approaching. Traffic lights and pedestrian crossings at intersections have to be synchronized to stop pedestrians and cyclists from crossing roads at anytime and anywhere.

Works done on traffic light simulation normally focuses on just one intersection. Modelling and simulation of different types of intersections is another area we will like to explore and analyse.

An overview of the problems defined is shown in table 1. Images were downloaded from Google image.

Images	source
	http://www.google.no/imgres?q=traffic+in+tema&sa=X&rlz=1C2OPRA_enNO523NO524&biw=1517&bih=741&tbm=isch&tbnid=MYILYcYJLz7S1M:&imgrefurl=http://lekmagh.org/about-lekma/231/road-network-and-conditions/&docid=ndzWFNIJk7ELcM&imgurl=http://lekma.lakesidegh.com/wp-content/uploads/2010/08/roads1.jpg&w=3264&h=2448&ei=EnO2UZDuG83Z4QT_goD4Ag&zoom=1&ved=1t:3588,r:19,s:0,i:137&iact=rc&dur=1698&page=2&tbnh=178&tbnw=251&start=19&ndsp=25&tx=184&ty=45
	http://photos.myjoyonline.com/pages/photos/932/38736.php
	http://farm3.staticflickr.com/2179/1653991204_3e3e37e66e_o.jpg

	http://img.modernghana.com/images/content/gbluhs_tema_assembly.jpg
	http://elderdillonhales.blogspot.no/2012/07/last-p-day-in-ghana-mtc.html

Table 1: Images of problems defined

Adaptation

The opportunity given me as an international student to study Discrete Simulation and Performance Analysis course in Norway inspired me to work on this project to help optimize the traffic control system in my home country Ghana. Not only can it be used in Ghana but also in any country with traffic network.

This work was adopted from a traffic light synchronization project designed by Tormod and Piyush [8]. It is also based on stochastic timing(firing time and delays associated with Petri net transitions are assumed to be random variables) [5],resource sharing and real-time simulation of traffic signal control studied in Discrete Simulation and Performance Analysis course [9].

Chapter Two

MODEL AND DESIGN

In this project, a simple model and a complex model of traffic intersection in Tema is designed. In a real world environment, between any two intersecting road is a direct road linking them therefore a simple road with a pedestrian crossing is also included in this simulation.

The simple intersection module consists of a three-lamp system with a four-way intersection. Traffic flow is from North to South, South to North, West to East and East to West. Cars are not allowed to make turns at the intersection and there are no pedestrian crossing.

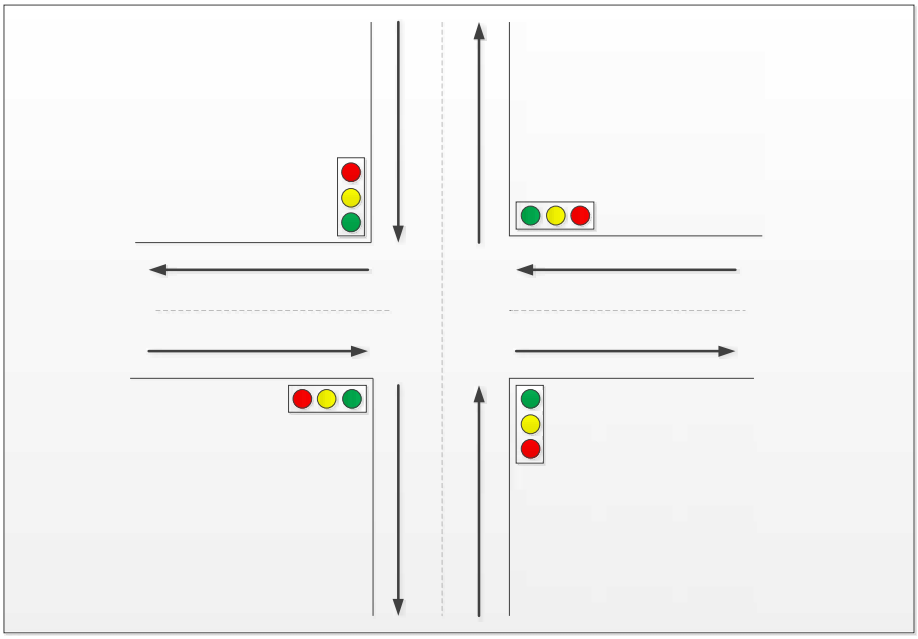


Figure 1: An overview of a Simple Intersection in Tema

The complicated module consists of a five-lamp system with a four-way intersection and a two-lamp system for pedestrian crossing at each intersection. Traffic flow is from North to South, South to North, West to East, East to West, North to West, North to East, South to West, South to East, West to North, West to South, East to North and East to South.

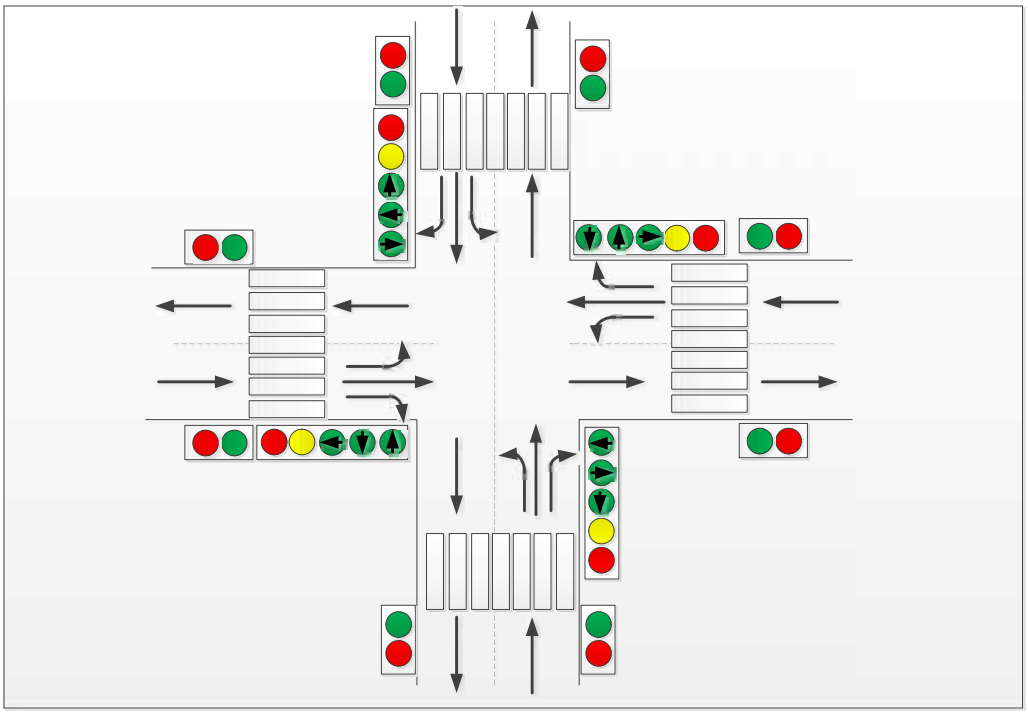


Figure 2: An overview of complex intersection in Tema

The direct road module consists of a three-lamp system with a one-way road and a two-lamp system for pedestrian crossing. Traffic flow is from North to South and South to North.

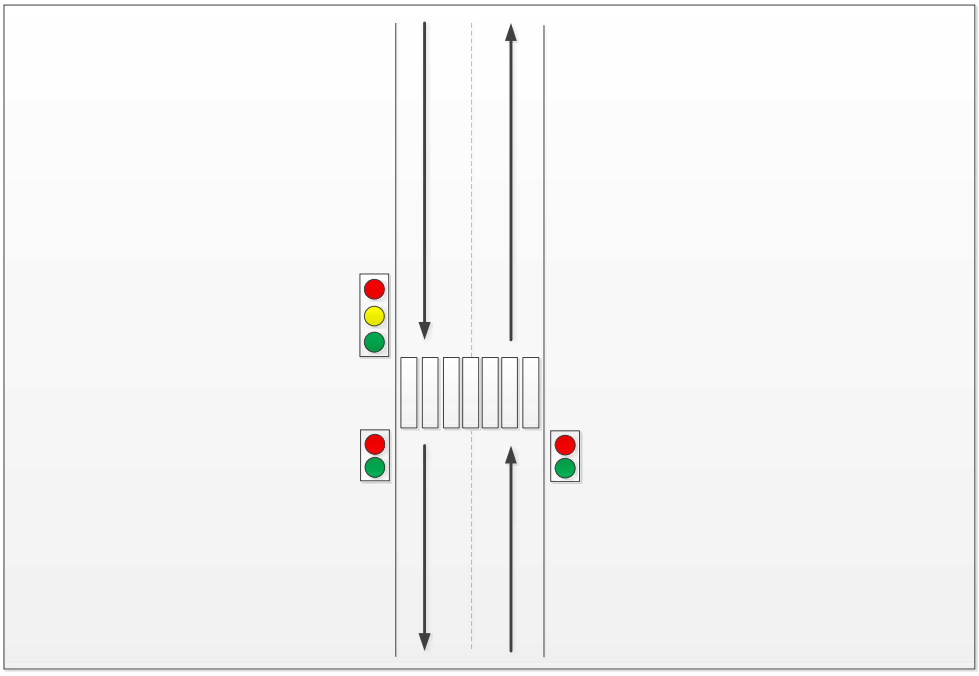


Figure 3: An overview of a direct road with pedestrian crossing in Tema

Traffic Control

All three modules have a normal cycle and an emergency cycle. According to [9] a normal cycle's lamp transition through red -> red & yellow -> green -> yellow ->red and an emergency cycle waits for a normal cycle to complete then the lamp blinks yellow. To resume a normal cycle, the start button must be pressed. A pedestrian cycle waits for a normal cycle to complete and the lamp turns red, after a while, the normal cycle resumes [9]. The models with pedestrian cycle have their control at the traffic crossing post.

Changes in the light are controlled by the firing of transitions. According to [10], firing times in real life systems are stochastic hence Poisson random function is used for stochastic firing of transitions in this simulation. Firing of transitions for the normal cycle is based on fixed time control whereas pedestrian and emergency cycles are based on dynamic time control whereby a button has to be pressed to activate them.

Each module is designed with its own control system. No external hardware is used in the simulation but a user interface is designed with buttons to control the different cycles of traffic system and display output of the simulation in a graphical mode.

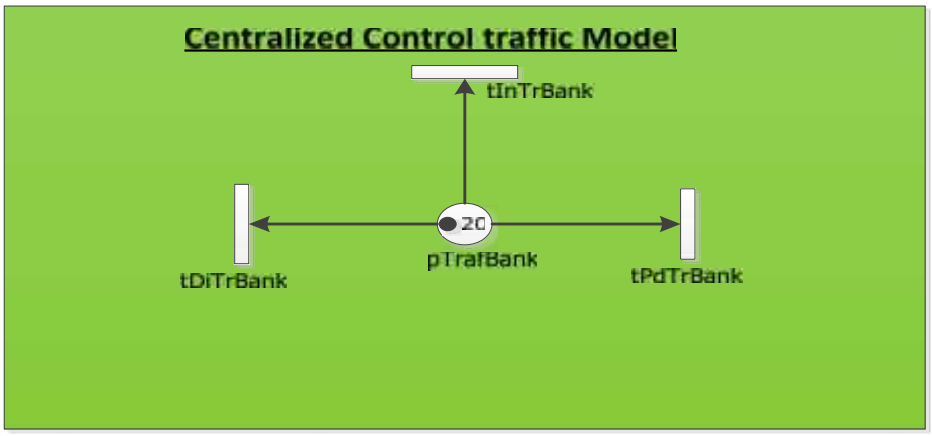
Module Control

A modular (many PDF's, one PDF for each module) approach [10] is used in the simulation.

Traffic Connection Module

It connects and controls all the modules. It serves as the token bank. When a button is pressed on the user interface, tokens are fired from this control to the appropriate module for the cycle to run.

Figure 4: Petri net model for traffic connection control



Simple Intersection Module

This module changes the state of the light in a bi-directional mode. North and South lights transition through the states together whereas West and East transition together.

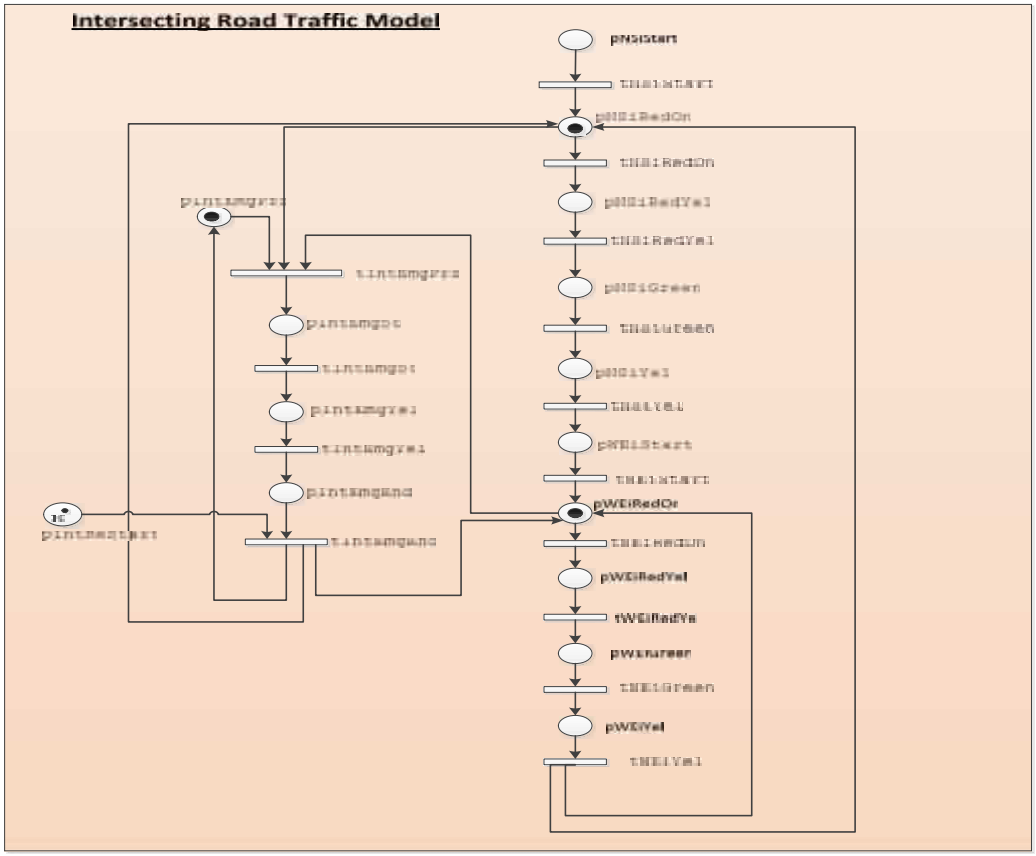
In the normal cycle,

- North and South light transition through the different colors completes the cycle then hands control to the West and East light.
- Whilst a pair of lights goes through a transition of lights, the other pair remains in the red state.

In the emergency cycle,

- Normal cycle completes and control is handed over to the emergency cycle.
- All four sets of lights change to the yellow state.
- The lights remain in this state till the start button is pressed to resume a normal cycle.

Figure 5: Petri net model for simple intersection in Tema



Complex Intersection Module

This module also changes the state of the light in a bi-directional mode but in a different cycle.

In the normal cycle,

- North and South lights in straight direction transition through the different colors, hands control to the West and East lights in straight direction.
- West and East lights in straight direction transition through the different colors, hands control to North and South in left direction.
- North and South lights in left direction transition through the different colors, hands control to the West and East lights in left direction.
- West and East lights in left direction transition through the different colors, hands control to North and South in straight direction.
- North and South lights in straight direction transition through the different colors, hands control to the West and East lights in straight direction.

- West and East lights in straight direction transition through the different colors, hands control to North and South in right direction.
- North and South lights in right direction transition through the different colors, hands control to the West and East lights in right direction.
- West and East lights in right direction transition through the different colors, hands control to North and South in straight direction.
- The cycle repeats itself.
- All the pedestrian lights remain in the red state.

In the emergency cycle,

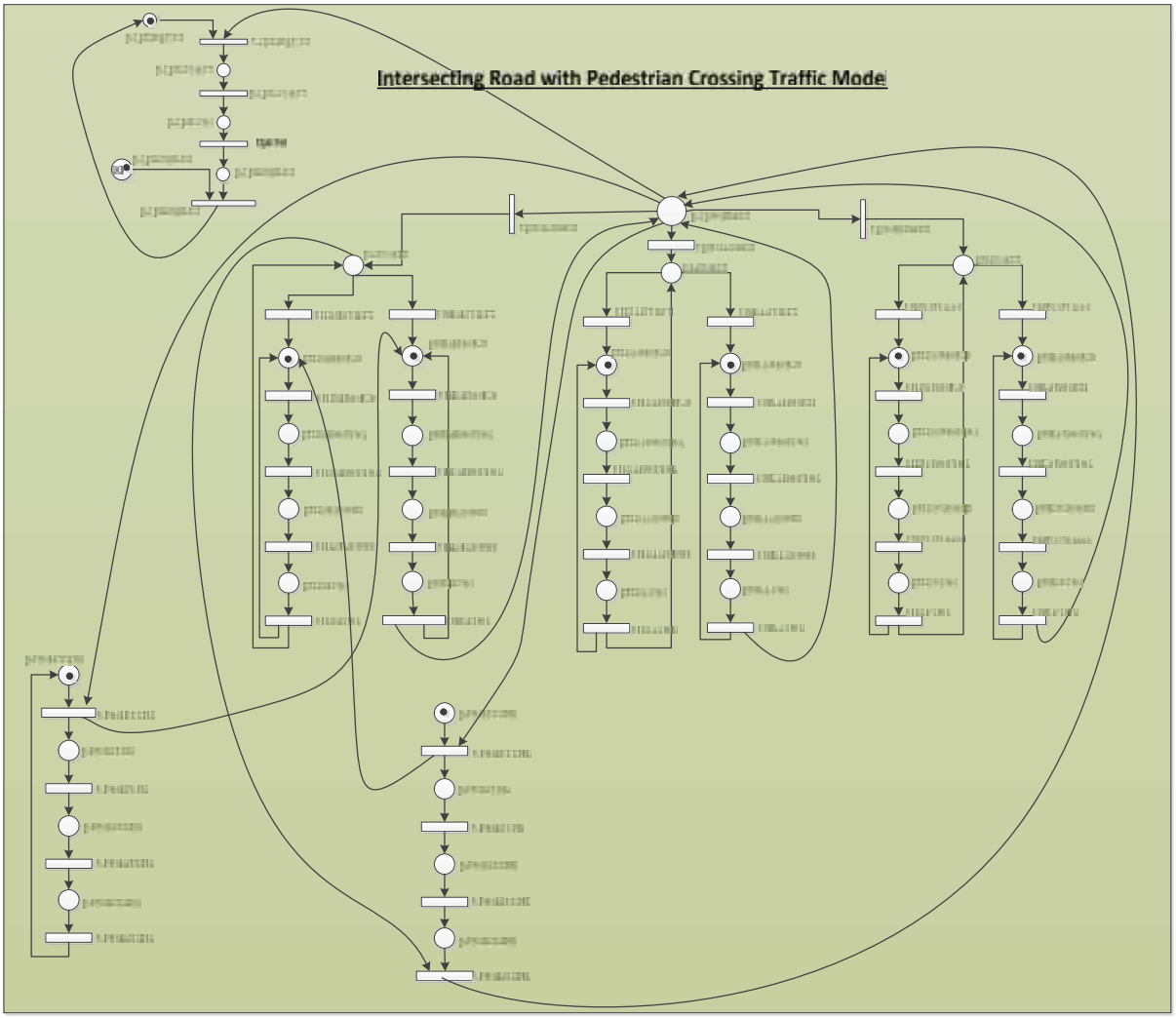
- Normal cycle completes and hands control to the emergency cycle.
- All four sets of lights change to the yellow state.
- Pedestrian lights show black because they only have red and green state.
- The lights remain in this state till the start button is pressed to resume a normal cycle.

In the Pedestrian cycle,

- Normal cycle in one direction completes and hands control to the pedestrian cycle i.e. If North and South in straight direction is running it hands control to West and East lights in the straight direction to complete the normal cycle in straight direction then hands control to pedestrian cycle.
- Traffic lights in North and South direction turn red if pedestrian button is pressed in either the North or South crossing.
- Traffic lights in the opposite straight direction i.e. in the West and East straight direction transition through the colors to allow traffic to flow in that direction till pedestrian cycle completes.
- Traffic lights in West and East turn red if pedestrian button is pressed in either the West or East crossing.

- Traffic lights in the opposite straight direction i.e. in the North and South straight direction transition through the colors to allow traffic flow till pedestrian cycle completes.
- The pair of Pedestrian lights shows green at the crossing at which it was pressed.
- Each pair of Pedestrian lights shows red for all the crossings where button was not pressed.
- The cycle resumes the normal state after pedestrian cycle completes.

Figure 6: Petri net model for complex intersection in Tema



Direct road Module

In this module, there is only one traffic light with a 3-lamp system and a pair 2-lamp system for pedestrian lights.

In the normal cycle,

- The lights transition through the colors till another cycle is pressed.
- Pedestrian lights remain in the red state.

In the emergency cycle,

- The normal cycle completes and hands control to the emergency cycle.
- The traffic lights change to a yellow state.
- The pedestrian lights change to a black state.
- The lights remain in this state till the start button is pressed to resume a normal cycle.

In the pedestrian cycle,

- The normal cycle completes and hands control to the pedestrian cycle.
- Traffic lights changes to a red state
- Pedestrian lights changes to a green state.
- Normal cycle resumes after pedestrian cycle completes.
- Pedestrian lights change back to red.

Figure 7: Petri net model for direct road in Tema

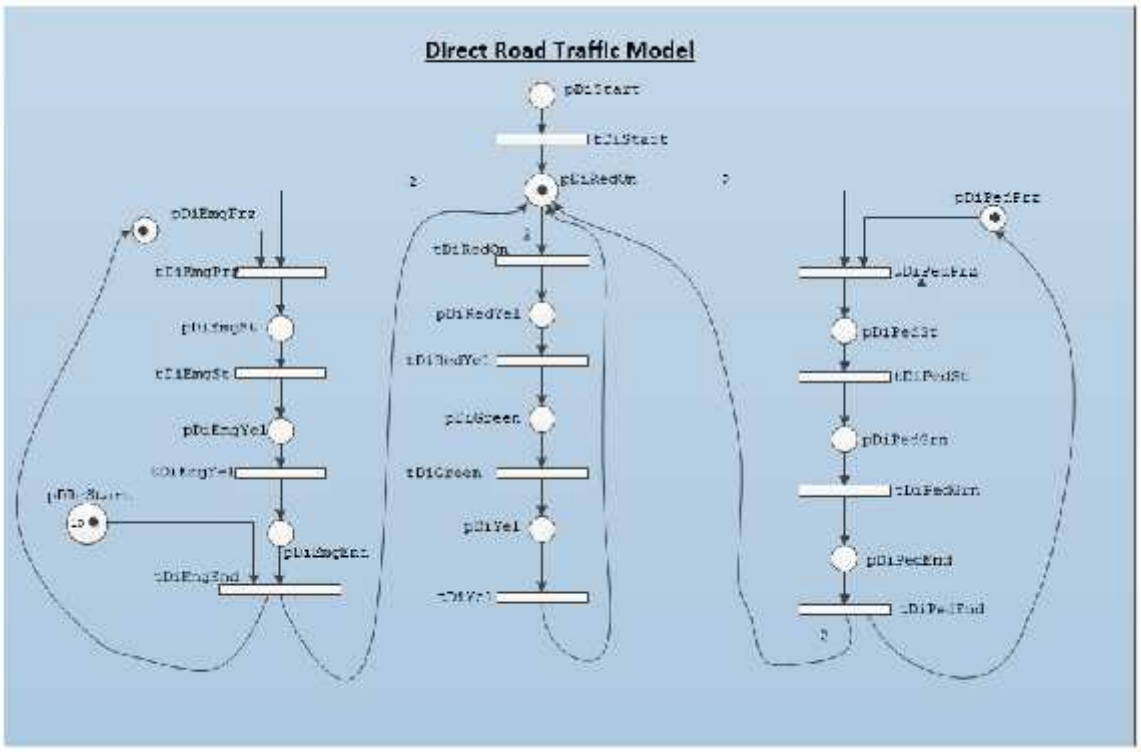
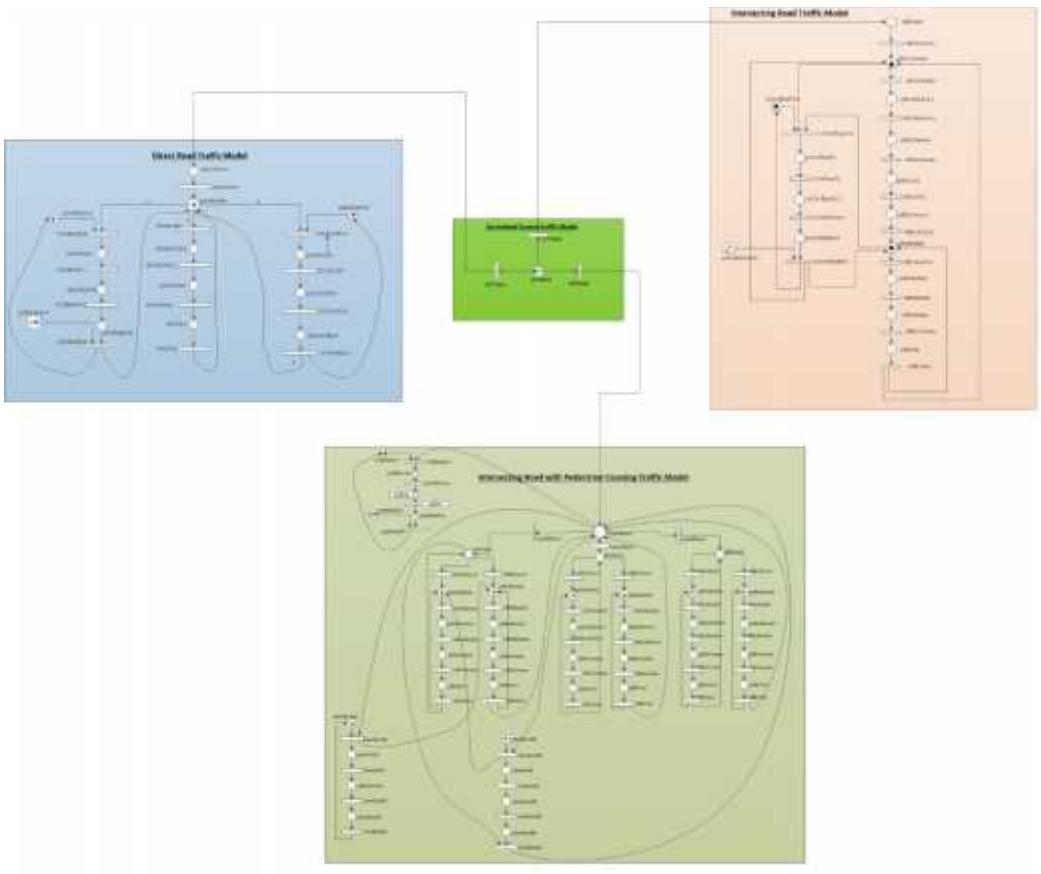


Figure 8: Petri net model of all 3 modules connecting to the traffic connection module



Technique

Ordinary P/T Petri net was used in the simulation. The complex intersection module used semafor [11] method of resource sharing to control the flow of traffic in a particular direction.

Colored Petri net is another technique for simulating DEDs but it was not used here because each module has its own set of places and transition to represent the lights[12]. Running the various modules does not depend on a particular type or kind of token present at the token bank. Once a module completes its cycle, a token is released from the token bank to start new cycle [13].

According to [14], marked graph are used for performance analysis and modelling of large DEDS. The reason this model was not used was because a marked graph is a Petri net which has exactly one input transition and one output transition [14]. In this design, some of the output transition is more than one which violates this rule.

Chapter Three

IMPLEMENTATION

Implementation of Modules

This project is implemented based on the module description defined in section 2. Transitions are assigned stochastic firing times. Each module defines its own set of places, transitions and arcs connecting them. A common PRE file was created to define conditions to be met before individual transitions can fire. Actions that need to occur after a transition has fired are defined in the common POST file. Within the POST file, values of Semafor are changed.

Snippet 1: Firing times of transition

```
dynamic.firing_times =
{'tDiStart',poissrnd(1,1,1),'tDiRedOn',poissrnd(2,1,1),'tDiYel',poissrnd(5,1,1),'tDiGreen',poissrnd(4,1,1),'tDiRedYel',poissrnd(3,1,1),...
'tDiPedPrz',poissrnd(4,1,1),'tDiEmgYel',poissrnd(2,1,1),'tDiPedGrn',poissrnd(7,1,1),'tDiPedEnd',poissrnd(10,1,1),... %direct firing times
   'tNSiRedOn',poissrnd(2,1,1),'tNSiYel',poissrnd(3,1,1),
'tNSiGreen',poissrnd(4,1,1),'tNSiRedYel',poissrnd(3,1,1),...
   'tWEiRedOn',poissrnd(3,1,1),'tWEiYel',poissrnd(4,1,1),
'tWEiGreen',poissrnd(5,1,1),'tWEiRedYel',poissrnd(4,1,1),... %intersection firing times
```

Snippet 2: Defining places, transitions and connecting arcs

```
function [png]=trafficConn_def()
png.PN_name = 'Connection for all traffic types Petri Net definition';
png.set_of_Ps ={'pTrafBank'};
png.set_of_Ts = {'tDiTrBank','tInTrBank','tPdTrBank'};
png.set_of_As = {'pTrafBank','tDiTrBank',1,'tDiTrBank','pDiStart',1,...
   'pTrafBank','tInTrBank',1,'tInTrBank','pNSiStart',1,...
   'pTrafBank','tPdTrBank',1,'tPdTrBank','pIpedBank',1};
```

Snippet 3: Common PRE file defining conditions for transition to fire, example tDiTrBank

```
function[fire,transition]=COMMON_PRE(transition)
global direct;global trafGlobal;global intersection;global global_info;
```

```
if strcmp(transition.name,'tDiTrBank'),
 if strcmp(direct.dirRun,'true'),
    if eq(direct.flag,0),
       direct.flag=1; fire=1;trafGlobal.window.setDirectionRed(444);
     else
       fire=0;
    end
 else
    fire=0;
 end
return;
```

Snippet 4: Common POST file defining actions to occur after a transition has fired

```
function [] = COMMON_POST(transition)
global direct;
global intersection;
global global_info;


counter=0;
if strcmp(transition.name,'tDiYel'),
   direct.flag=0;
   return;
elseif strcmp(transition.name,'tWEiYel'),
    trafGlobal.window.setDirectionRed(2);    intersection.flag=0;
   return;
```

A GUI was developed in JAVA to graphically display the output results of the simulation. The .jar file was imported into the MSF file to enable GPenSIM call its functions. The GUI has methods which GPenSIM calls to check if a button is pressed and methods for updating the GUI. Changes in the lights on the GUI are controlled within the PRE and POST files and it's triggered by the firing transition.

Snippet 5: Methods called in the GUI by GPenSIM to check if a button is pressed

```
public int getGlobalValOne()
   {
```

```
    return roadGlobals.g1Int;
  }

  public String getGlobalDirectStart()
  {
   return String.valueOf(roadGlobals.clickDirStart);
  }

   public String getGlobalDirectRun()
  {
   return String.valueOf(roadGlobals.dirRun);
  }
```

Snippet 6: One of the methods called to update the GUI

```
public void setDirectionYellow(int d)
 {
    switch(d)
 {
    case 1:N2light.setYellow();S2light.setYellow();
       break;
    case 2:W2light.setYellow();E2light.setYellow();
       break;
    case 11:tNlight.setYellow();tSlight.setYellow();
       break;
    case 22:tWlight.setYellow();tElight.setYellow();
       break;
    case 111:light.setYellow();
       break;
 }
}
```

Snippet 7: GPenSIM calls a function in JAVA which changes the colour of the lights within a PRE file

```
trafGlobal.window.setDirectionRed(1);
trafGlobal.window.setDirectionRed(2);
```

A .m file was created to receive feedback from the GUI. In this file, the simulation runs in a loop. When a button is clicked, a global variable in JAVA is updated. Because .jar file is imported into GPenSIM, the main functions are exposed therefore are called within each loop to detect if an action is performed on a button based on the value of the JAVA global value. This file calls methods defined in snippet 5.

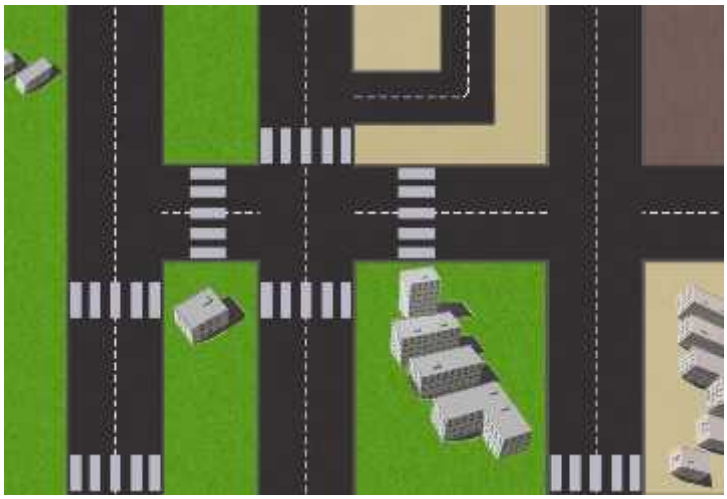
Snippet 8: GPenSIM detects changes in the GUI and gives control to the particular module

```
function [traffic_inputs]=get_Traffic_Inputs()
global direct;
global vv;
import roadPackage.*;
vv=roadPackage.roadFxnClass();s=vv.getGlobalValOne;
while(s==3)
   vv=roadPackage.roadFxnClass();
   s=vv.getGlobalValOne;
end
Dir=vv.getGlobalDirectStart();
direct.dirRun=vv.getGlobalDirectRun();
if strcmp(direct.dirRun,'true'),
   if strcmp(Dir,'true'),
      direct.Direct='on';
   else
      direct.Direct='off';
   end;
end
```

Implementation of GUI

The graphical user interface background image was designed using Adobe fireworks. Using java tools and shapes in NetBeans IDE 7.3, the different types of lights and buttons were placed at each corner of an intersection. Icon used for pedestrian crossing was downloaded from icon archive. (http://www.iconarchive.com/show/phuzion-icons-by-kyo-tux/Sign-Public-icon.html)

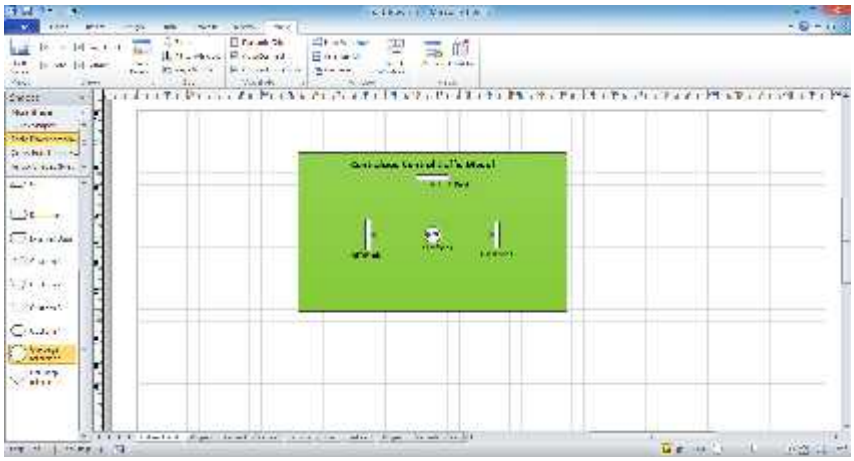
Figure 9: GUI background without lights



Implementation of module overview and Petri Net model

The overview images for the different traffic modules were designed in VISIO 2010. It was also used in building their Petri Net models using basic flow chart shapes.

Figure 10: A screen shot of VISIO 2010 interface



Chapter Four

TESTING, ANALYSIS AND RESULTS

User Manual for running simulation

Testing is done through the GUI designed in JAVA. All three scenarios are shown in the GUI, each with its own control system. There is a start and emergency button below the module's label but the pedestrian buttons are place besides the pedestrian lights. The 3/5-set lights are for cars and the 2-set lights are for pedestrians.

Figure 11: GUI showing the control buttons and lights

To run a module, click on the start button placed below the scenario label. This button is deactivated once pressed and reactivated when emergency button is pressed. Click on either left or right button place by the 2-set lights to run the pedestrian cycle. Emergency button found below the start button runs the simulation through the emergency cycle.

Figure 12: Normal cycle for direct road

Figure 13: Emergency cycle for direct road

Figure 14: Pedestrian cycle for direct road

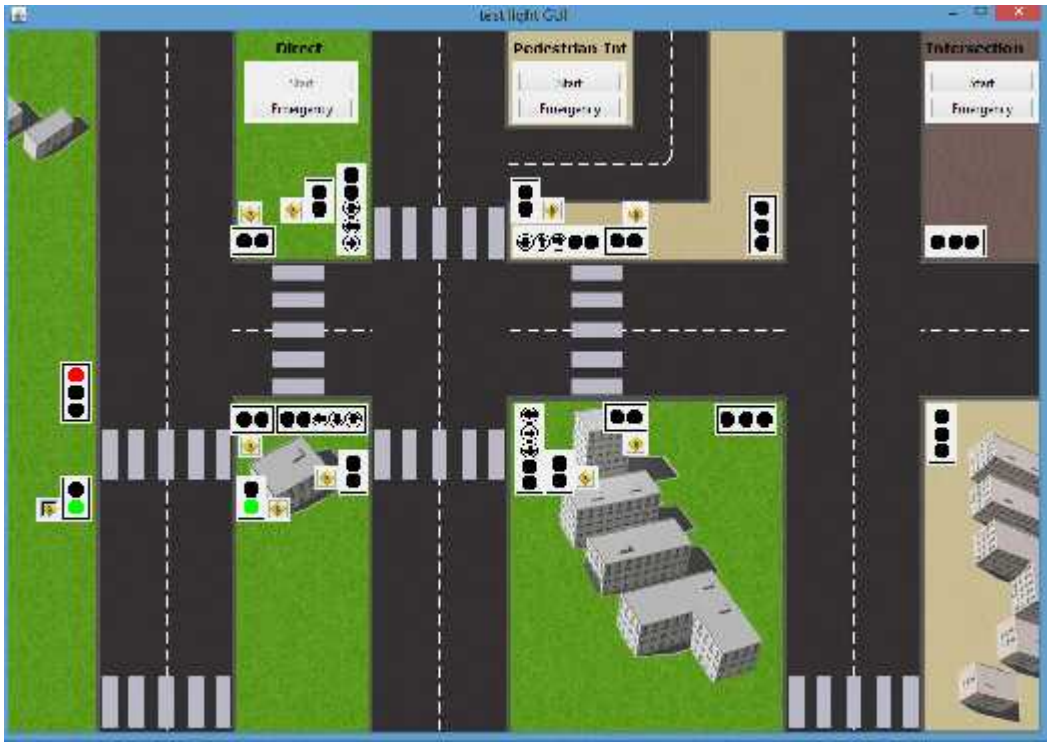


Figure 15 : Normal cycle for simple intersection

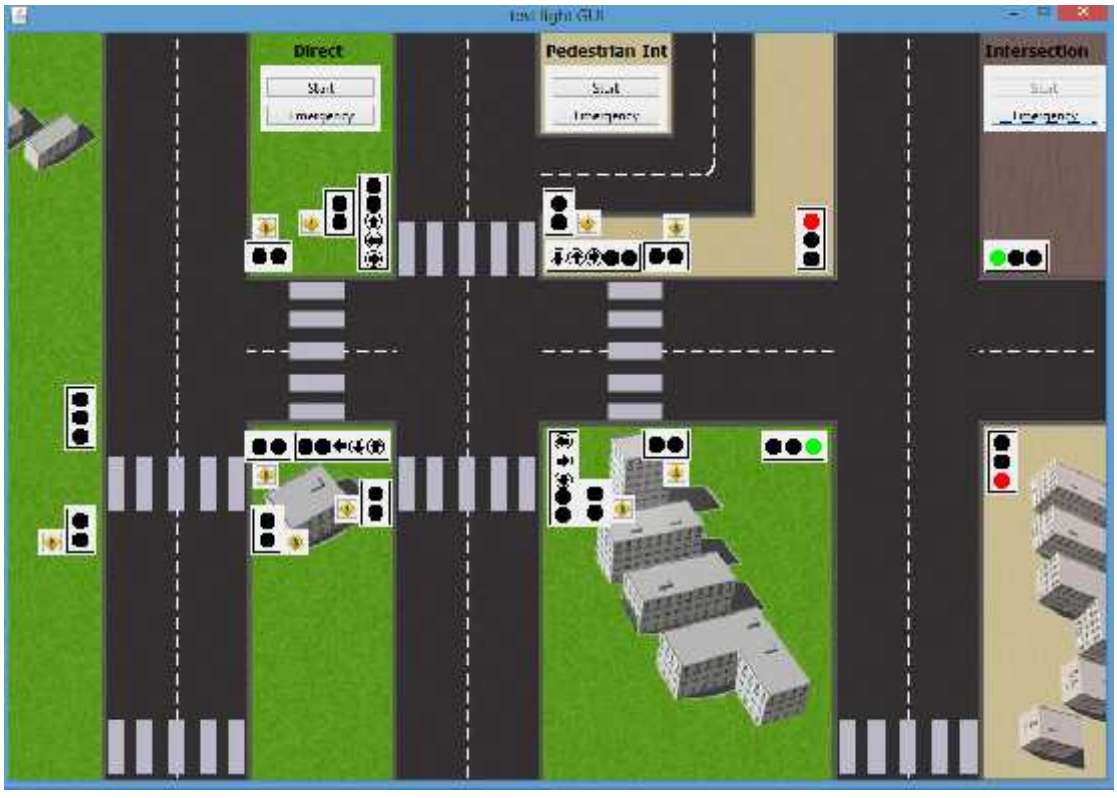


Figure 16: Emergency cycle for simple intersection

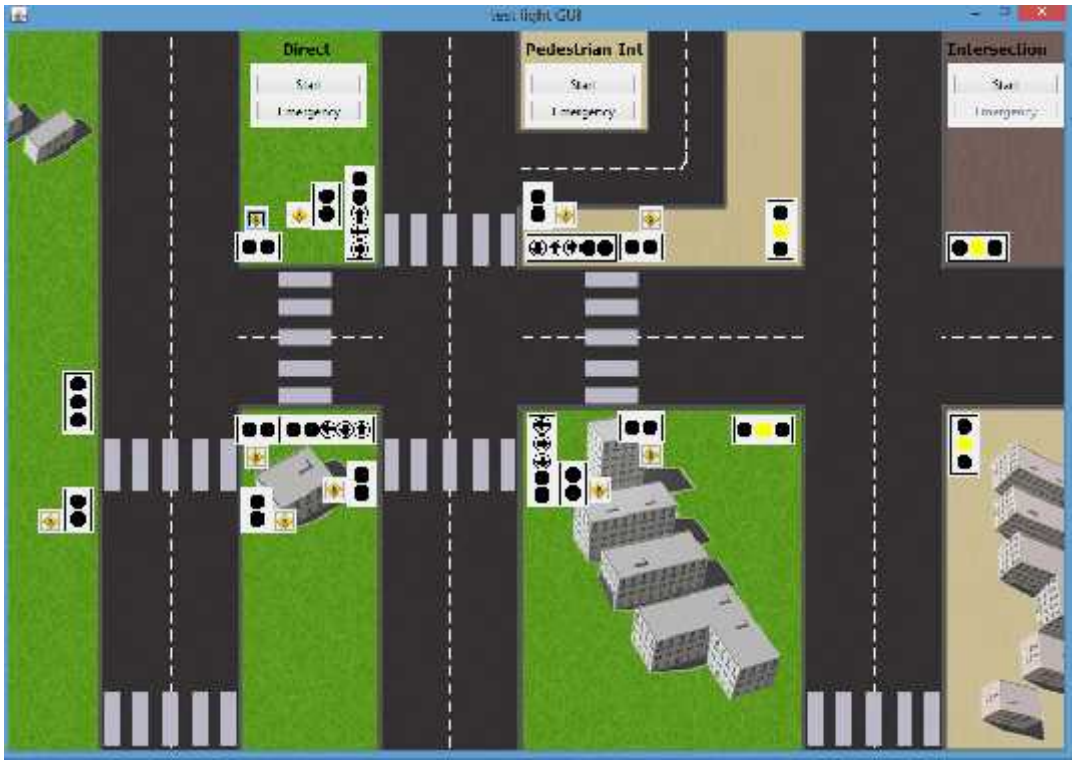


Figure 17: Normal cycle for complex intersection

Figure 18: Emergency cycle for complex intersection

Figure 19: Pedestrian cycle in the North direction for complex intersection

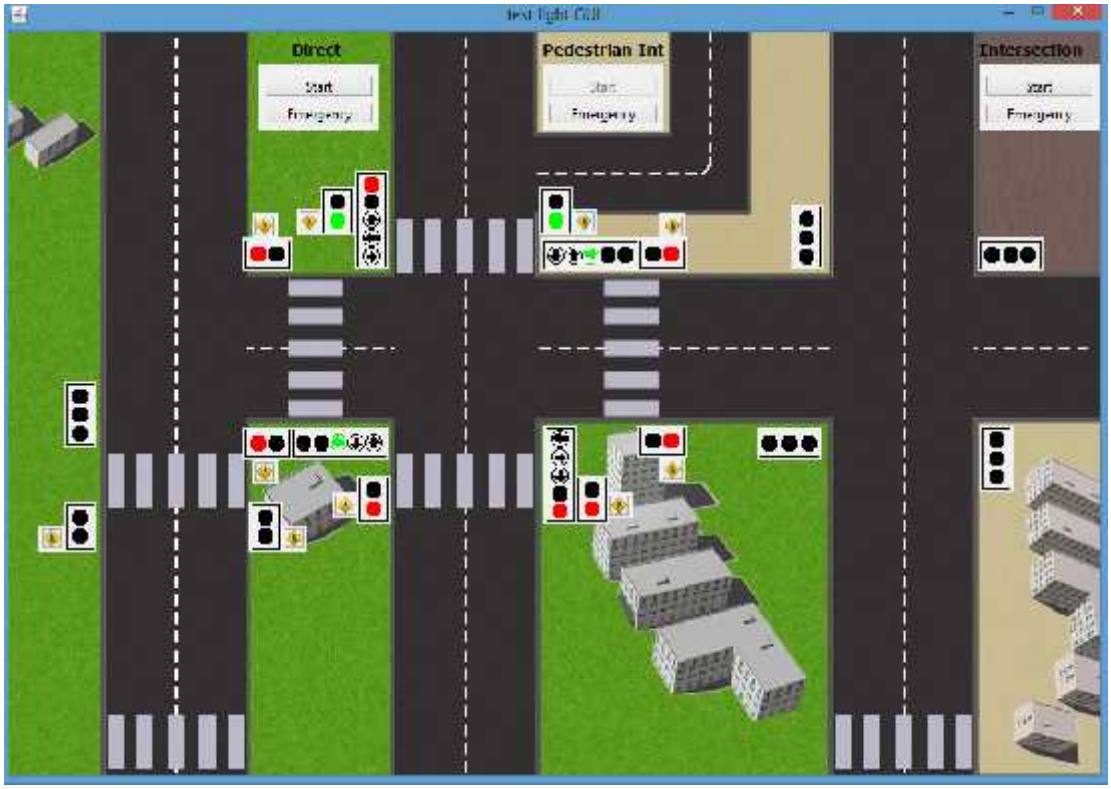


Figure 20: Pedestrian cycle in the South for complex intersection

Figure 21: Pedestrian cycle in the East direction for complex intersection

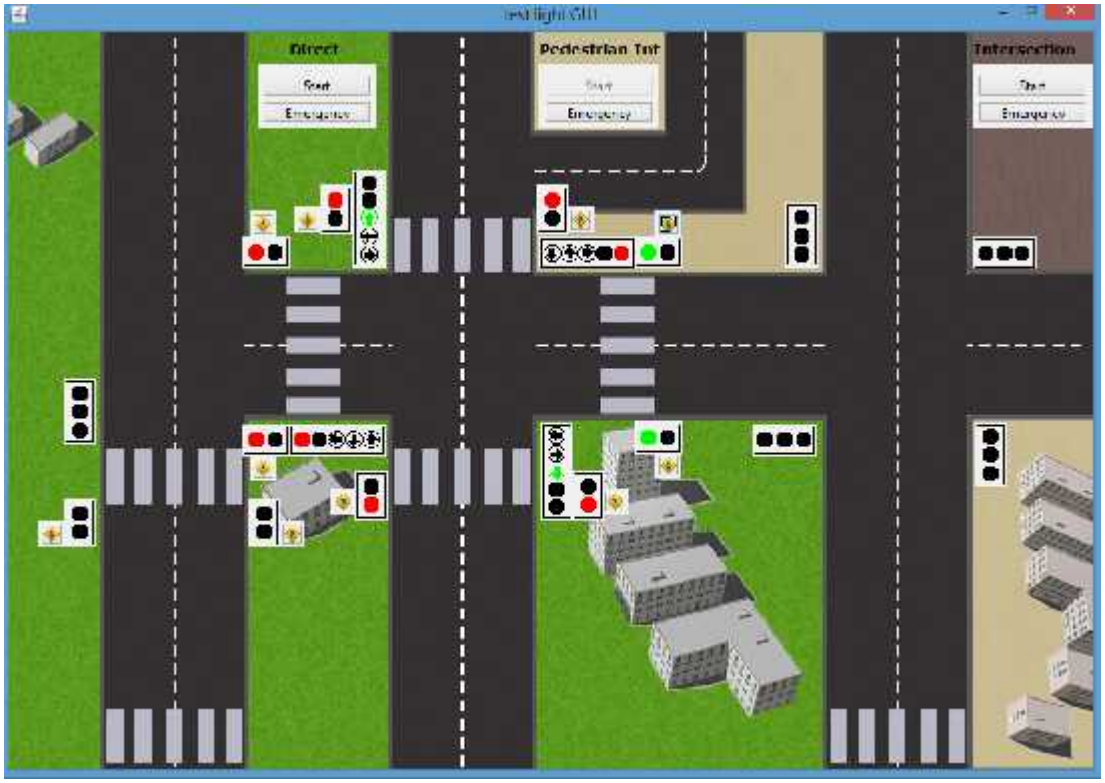


Figure 22: Pedestrian cycle in the West direction for complex intersection

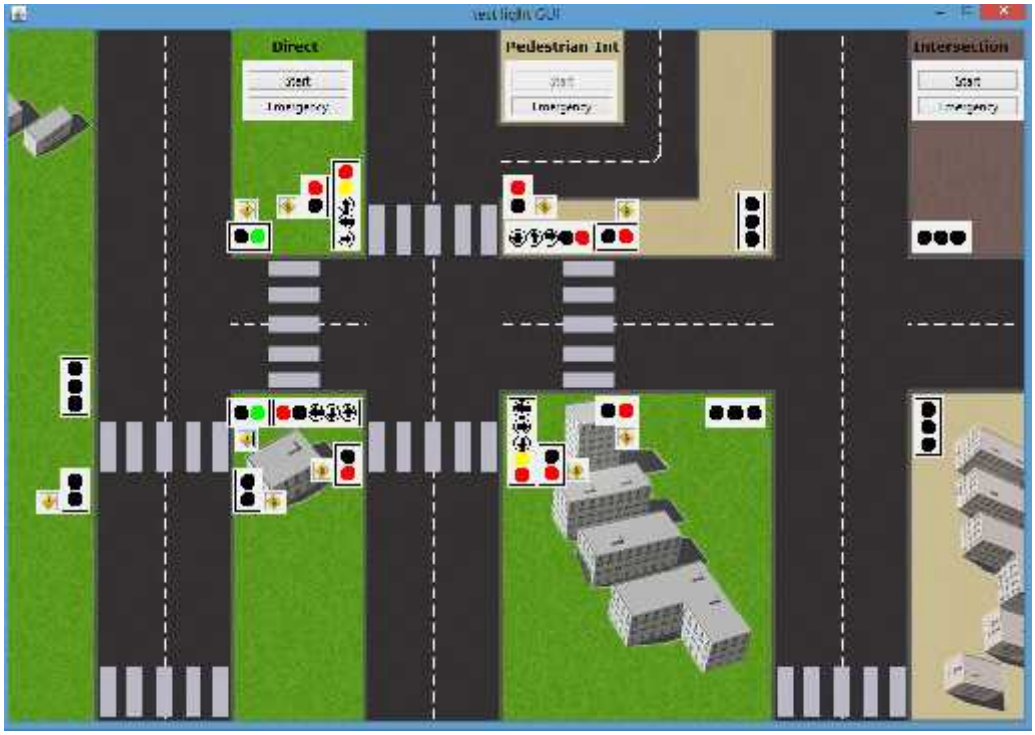


Figure 23: Normal cycle for all 3 modules

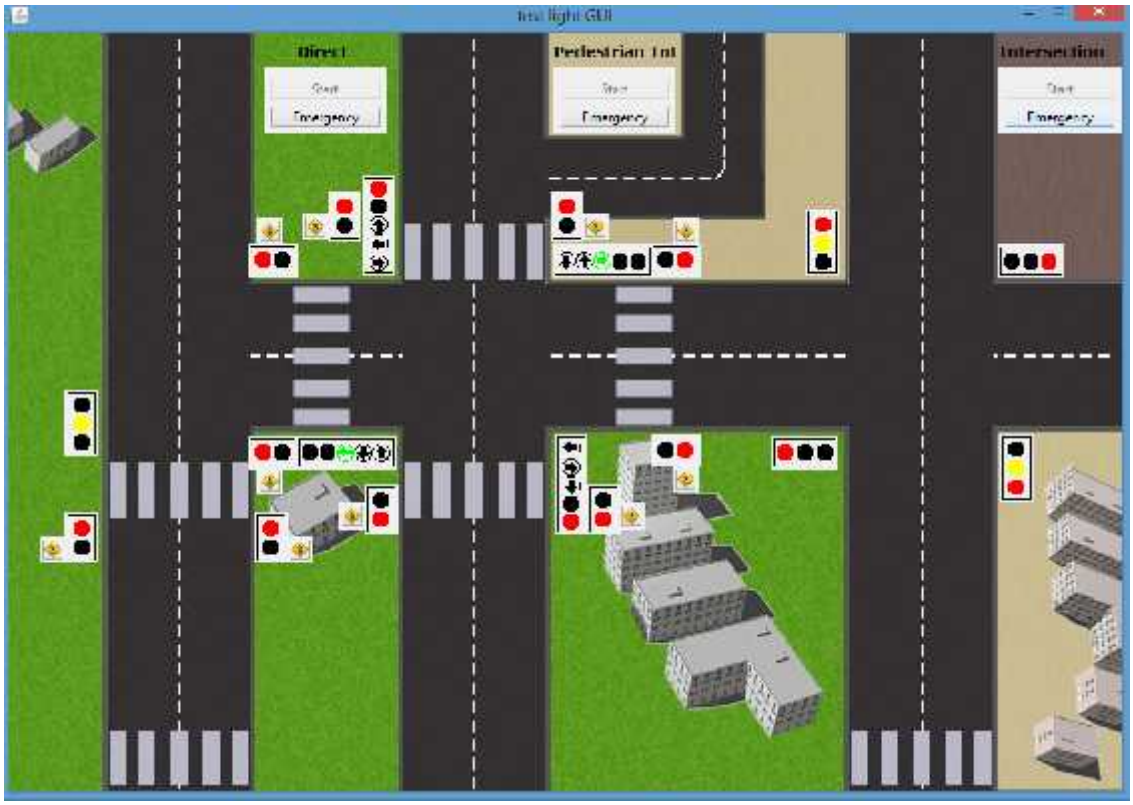


Figure 24: simulation time and toke frequency in direct road model

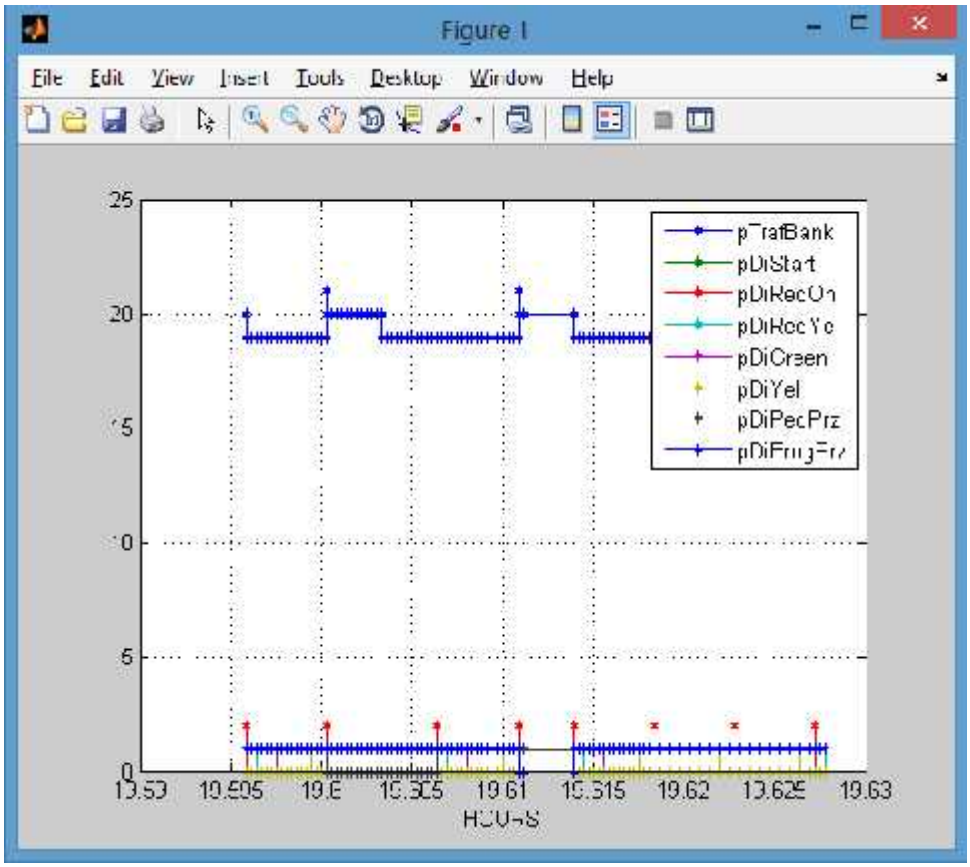


Table 2: Legend definition for figure 24

place name	meaning
pTrafBank	traffic bank
pDiStart	direct start
pDiRedOn	direct red on
pDiRedYel	direct red and yellow
pDiGreen	direct green
pDiYel	direct yellow
pDiPedPrz	direct pedestrian pressed
pIntEmgPrz	direct emergency pressed

Figure 25: Simulation time and token frequency in simple intersection

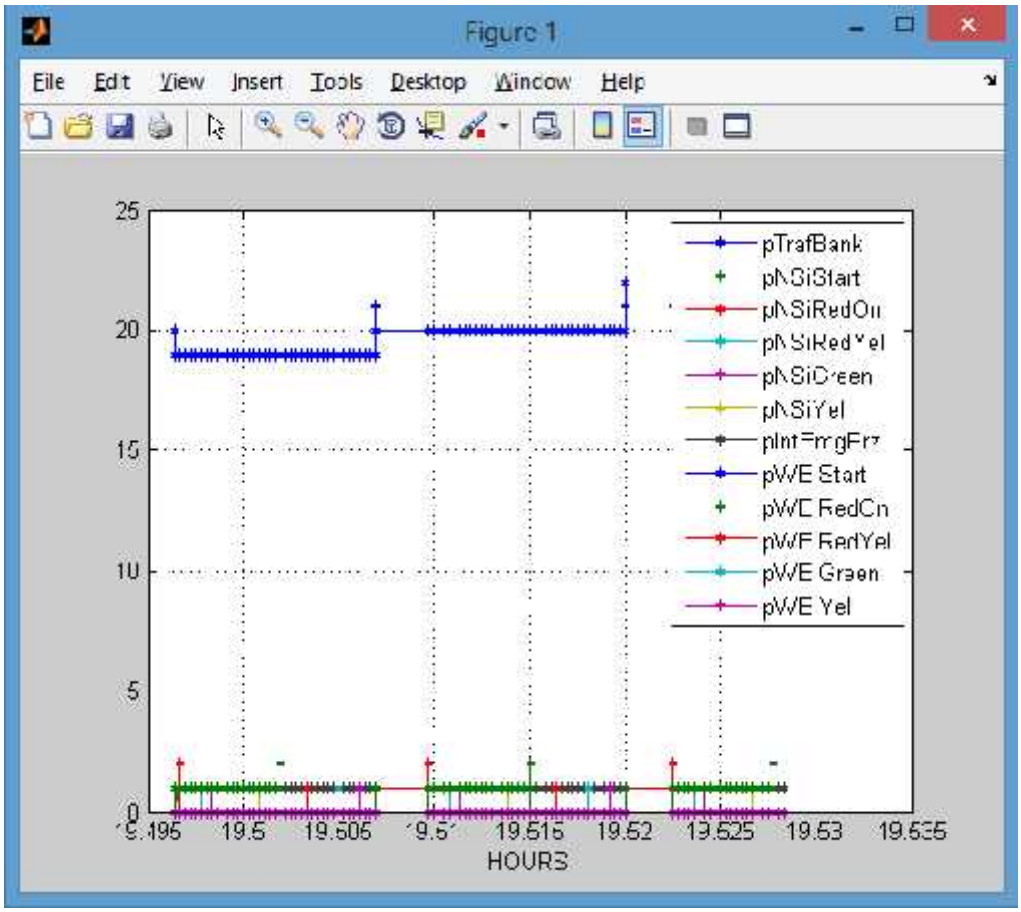


Table 3: Legend definition for figure 25

place name	represents
pTrafBank	Traffic bank
pNSiStart	Intersection NS start
pNSiRedOn	Intersection NS red on
pNSiRedYel	intersection NS red and yellow
pNSiGreen	intersection NS green
pNSiYel	intersection NS yellow
pIntEmgPrz	Intersection emergency pressed
pWEiStart	intersection WE start
pWEiRedOn	intersection WE red on
pWEiRedYel	intersection WE red and yellow
pWEiGreen	intersection WE green
pWEiYel	intersection WE yellow

Figure 26: Simulation time and token frequency in complex intersection

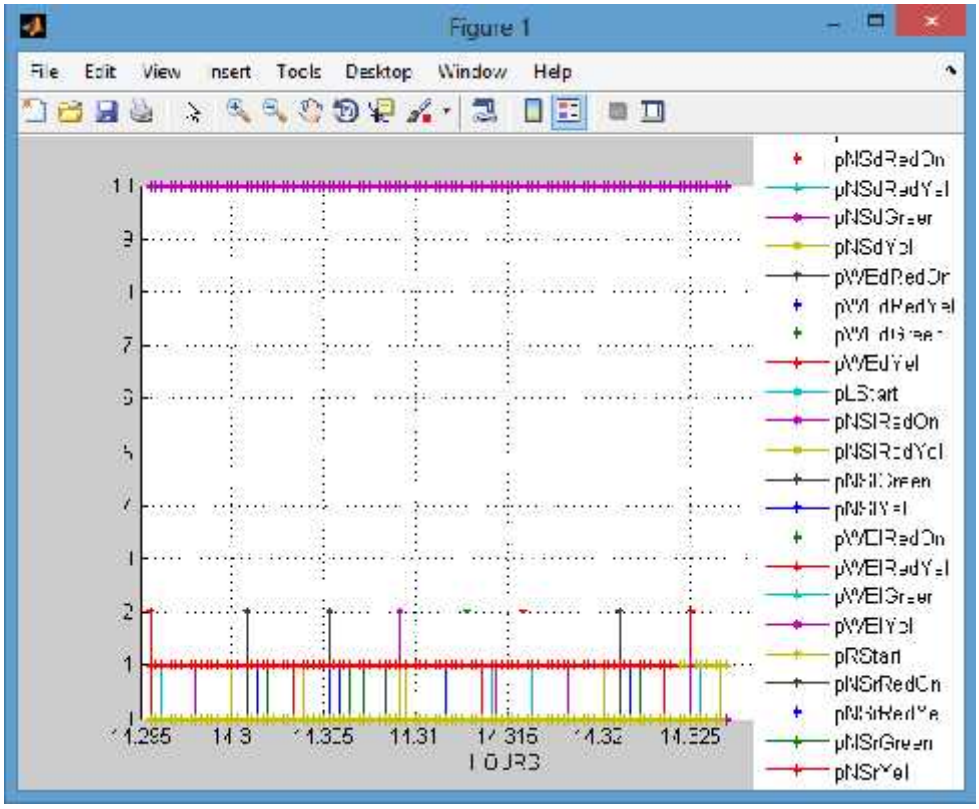


Table 4: Definition of legends in figure 26

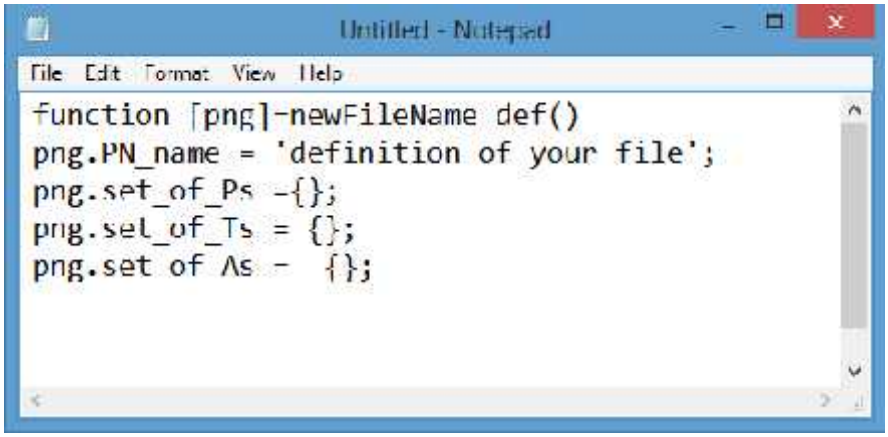
place name	meaning	place name	meaning
pIpedBank	Intersection with pedestrian token Bank	pNSrRedYel	turn right in NS red and yellow
pDStart	Direct direction start	pNSrGreen	turn right in NS green
pNSdRedOn	Direct North South red on	pNSrYel	turn right in NS yellow
pNSdRedYel	direct NS red yellow	pWErRedOn	turn right in WE red on
pNSdGreen	direct NS green	pWErRedYel	turn right in WE red and yellow
pNSdYel	direct NS yellow	pWErGreen	turn right in WE green
pWEdRedOn	direct WE red on	pWErYel	turn right in WE yellow
pWEdRedYel	direct WE red and yellow	pIpEmgPrz	intersection with pedestrian emergency pressed
pWEdGreen	direct WE green	pIpEStart	intersection with pedestrian emergency start

pWEdYel	direct WE yellow	pIpEYel	intersection with pedestrian emergency yellow
pLStart	turn left start	pIpEmgEnd	Intersection with pedestrian emergency end
pNSlRedOn	turn left in NS red on	pERestart	emergency restart
pNSlRedYel	turn left in NS red and yellow	pPedPrzNS	pedestrian pressed in NS
pNSlGreen	turn left in NS green	pPedStNS	pedestrian start in NS
pNSlYel	turn left in NS yellow	pPedGrnNS	pedestrian green in NS
pWElRedOn	turn left in WE red on	pPedEndNS	pedestrian end in NS direction
pWElRedYel	turn left in WE red and yellow	pPedPrzWE	pedestrian pressed in WE
pWElGreen	turn left in WE green	pPedStWE	pedestrian start in WE
pWElYel	turn left in WE yellow	pPedGrnWE	pedestrian green in WE
pRStart	turn right start	pPedEndWE	pedestrian end in WE direction
pNSrRedOn	turn right in NS red on		

User manual for adding new modules

A modular approach was used in this work therefore it's possible to add new modules. This section explains the steps needed to make additions.

- Define a new PDF file using this template. In this file define all your places, transitions and arc weight between them. A sample code is found in *directRoad_def.m*



- Edit *trafficConn_def.m* file, add the name of the transition to the existing list of *png.set_of_Ts = {'newTransitionname'}*.This transition serves as the link between the traffic control and your PDF file. Define the arc weight as shown in lines 5-7 of the same file. This project has three modules and each module connection is defined on a new line.

- In the MSF file, *light_pn.m* add the name of your PDF file to *png=petrinetgraph({'newPDFname'})* list. Define all places that have initial markings in *dynamic.initial_markings = {'placeA',number_of_tokens}.* Define firing time of transitions in *dynamic.firing_times = {'transitionA',firing_time}*

If you are still not sure, you can use any of the 4 defined PDF files as a guide:

1. trafficConn_def.m
2. directRoad_def.m
3. IntersectingRoad_def.m
4. Ped_IntersectingRoad_def.m

The MSF is: *light_pn.m*

Chapter Five

DISCUSSION

From this project it has been shown that control of traffic synchronization can be implemented using Petri Net. Not only can it be used to model one intersection but many. The results prove that it will be easy for road and highways authorities to adopt this method to simulate synchronization for any given area before roll-out. This work was done using a modular approach therefore making it possible to add or remove modules.

Though this work was based on the project work done by Tormod and Piyush [8], emphasis was based on different intersections with other road complexities and how they connect to each other. In their model, they implemented only a simple traffic intersection so a complex one was implemented in this project. They did not implement pedestrian crossing at any of their intersections. The whole model was also managed by one control. According to [9], emergency and normal cycles are controlled remotely whereas pedestrian cycles are controlled at signal post that was why Start and Emergency buttons were placed differently from the pedestrian buttons. In this project each pedestrian crossing had two control buttons, one on each side of the road. Pedestrian crossings normally have two lights i.e. red and green, there are no yellow lights but in [8],it was implemented with a yellow light.

A major limitation in this work is the logic handling the traffic flow in the complex intersection when a pedestrian button is pressed. I believe it can be improved on with further work so that traffic can easily flow in different directions and not only in the straight direction when pedestrian buttons are pressed.

Adding of new modules is possible but displaying results visually is impossible on the GUI because GUI is limited to the defined modules in this work only. Unless a new GUI is designed to display the new overall design. A further work in a dynamic user interface where users can easily create their own should be possible.

Consideration was given only to the lights and control but other major factors like rush hour, number of vehicles moving in a particular direction etc were not considered. Analysis on these factors can make a better simulation. Results from the analysis of these factors can be used to control the time at which the lights should change example, if from the analysis it is

observed that at 17:00hrs, there are a lot of vehicles in the north-south direction, then firing times of transition for the green light can be increased so that more cars can move in that direction.

Chapter Six

CONCLUSION

Petri Net is an efficient and easy tool to use in modelling and simulation of real world scenarios. It allows resource sharing, scheduling and analysis of a simulation in real-time. GPenSIM, software for modelling in Petri Net is compatible with other languages therefore makes it easy to study and analyse results graphically. From the work above, it proves that traffic network within an urban area can easily be simulated and the results can be used to solve existing issues with traffic network. With this modular approach, new intersections can always be added and already existing ones can be upgraded or downgraded.

REFERENCES

1. Merriam-Webster. *Traffic light*. 2013 [cited 2013 April,2013]; Available from: http://www.merriam-webster.com/dictionary/traffic%20light.
2. Merriam-Webster. *Traffic light*. 2013 [cited 2013 May 2013]; Available from: http://www.learnersdictionary.com/search/traffic%20light.
3. Google. *traffic light*. [cited 2013 May 2013]; Available from: https://www.google.no/#sugexp=cgrwrth&gs_rn=17&gs_ri=psy-ab&tok=im58qMMBrGZahCwEceZJuQ&suggest=p&pq=traffic%20light&cp=17&gs_id=s&xhr=t&q=traffic+light+definition&es_nrs=true&pf=p&sclient=psy-ab&oq=traffic+light+def&gs_l=&pbx=1&bav=on.2,or.r_cp.r_qf.&bvm=bv.47810305,d.bGE&fp=385327d7913fc473&biw=1517&bih=693.
4. Peterson, J.L., *Petri Net Theory and the Modelling of Systems*. 1981, Prentice Hall PTR Upper Saddle River, NJ, USA: Prentice-Hall. 288.
5. Hruz, B. and M.C. Zhou, *Modeling and Control of Discrete-event Dynamic Systems*. 2007, London: Springer.
6. Davidrajuh, R. and I. Molnar, *Designing a new tool for modelling ans simulation of discrete-event systems.* Issues in Information Systems, 2009. **10**(2).
7. Service, G.S., *Population by Region, District, Age Groups and sex, 2010*, 2010.
8. Lea, T.E. and P. Duggal, *Traffic Light Synchronization*, 2012, University of Stavager.
9. Davidrajuh, R., *Traffic Signal Control*, D.S.a.P.A. University of Stavanger, Editor 2012. p. 1-18.
10. Davidrajuh, R., *A Tool for Modeling and Simulation of Discrete-Event Systems*. GPenSIM. 2010.
11. Davidrajuh, R., *Simple Examples with GPenSIM*, D.S.a.P.A. University of Stavanger, Editor 2012. p. 2-6.
12. Davidrajuh, R., *CPN Example: Circular Train*, D.S.a.P.A. University of Stavanger, Editor 2012. p. 2-20.
13. Davidrajuh, R., *Coloring Tokens in GPenSIM*, D.S.a.P.A. University of Stavanger, Editor 2012. p. 2.
14. Davidrajuh, R., *Marked Graph*, D.S.a.P.A. University of Stavanger, Editor 2012. p. 2,6.

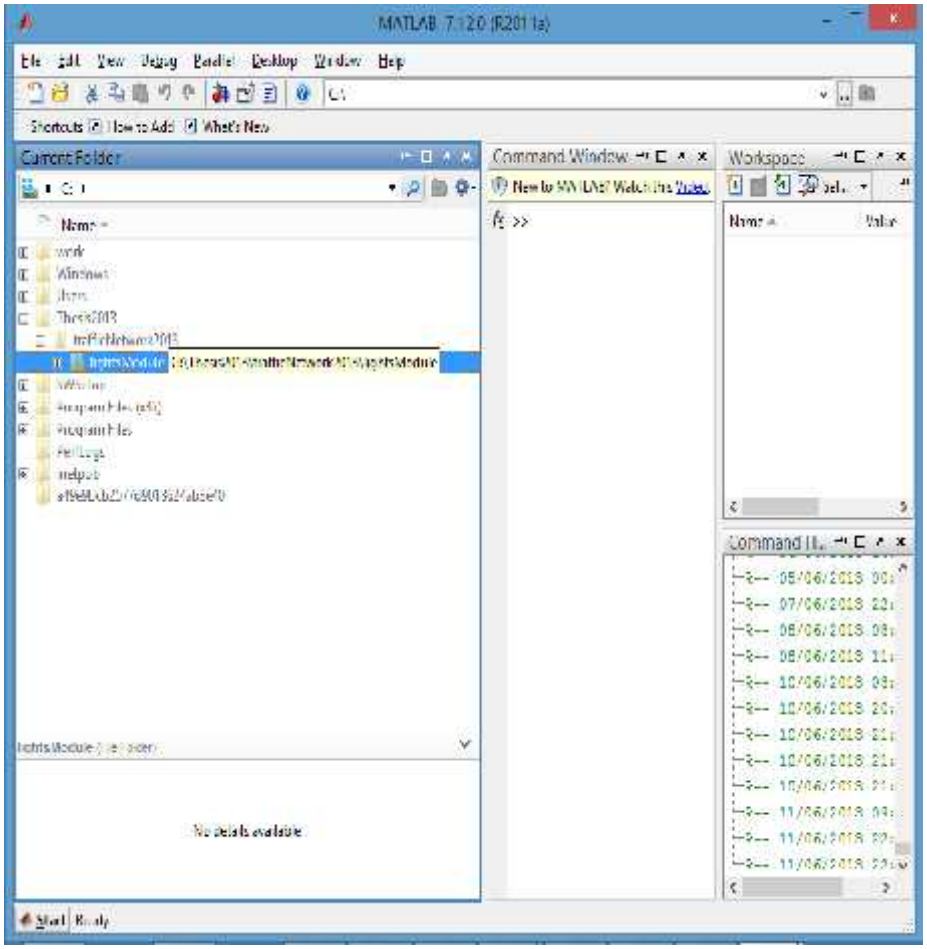
APPENDIX

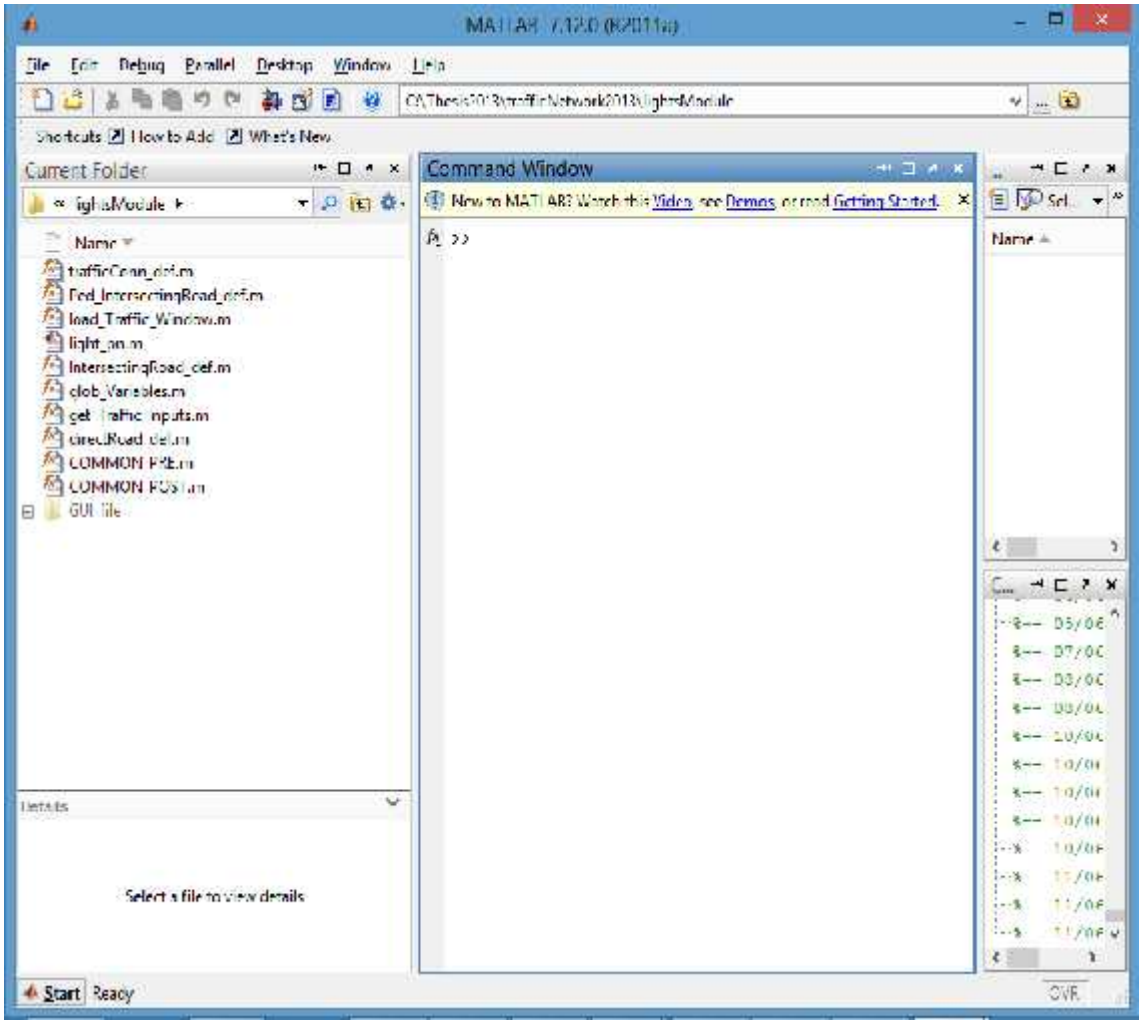
System Environment Requirements

- GPenSIM v6
- MATLAB (R2011a), 64-bit
- Java jre-6

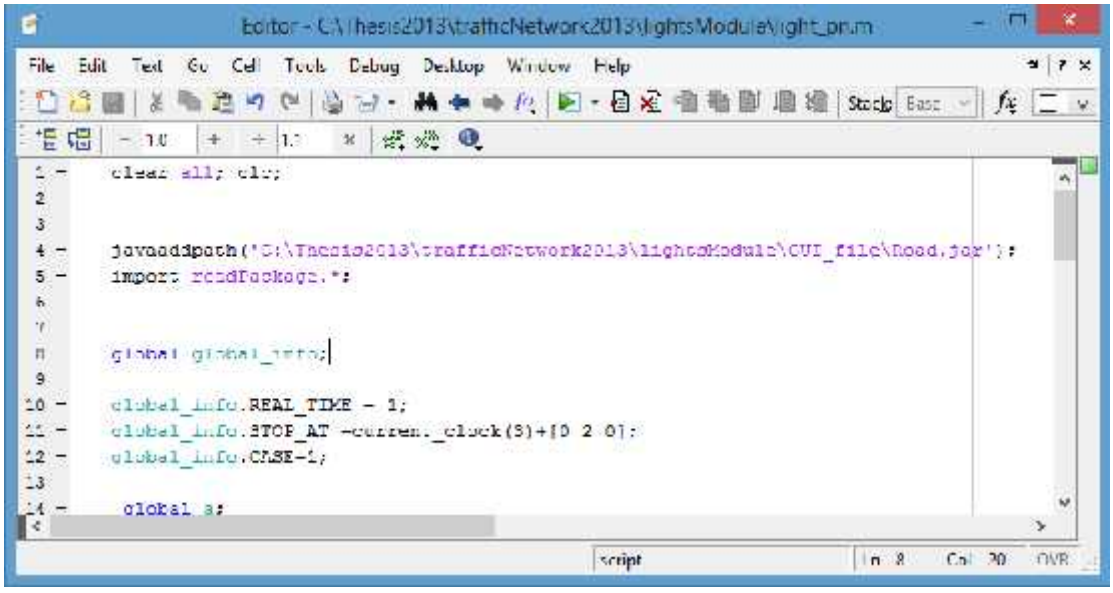
Installation Manual

1. Extract Thesis2013.rar file to any location of your choice.
2. Start MATLAB
3. Set path to GPenSIM as explained in [6]
4. In Current Folder pane of MATLAB, browse to the location of unzipped file



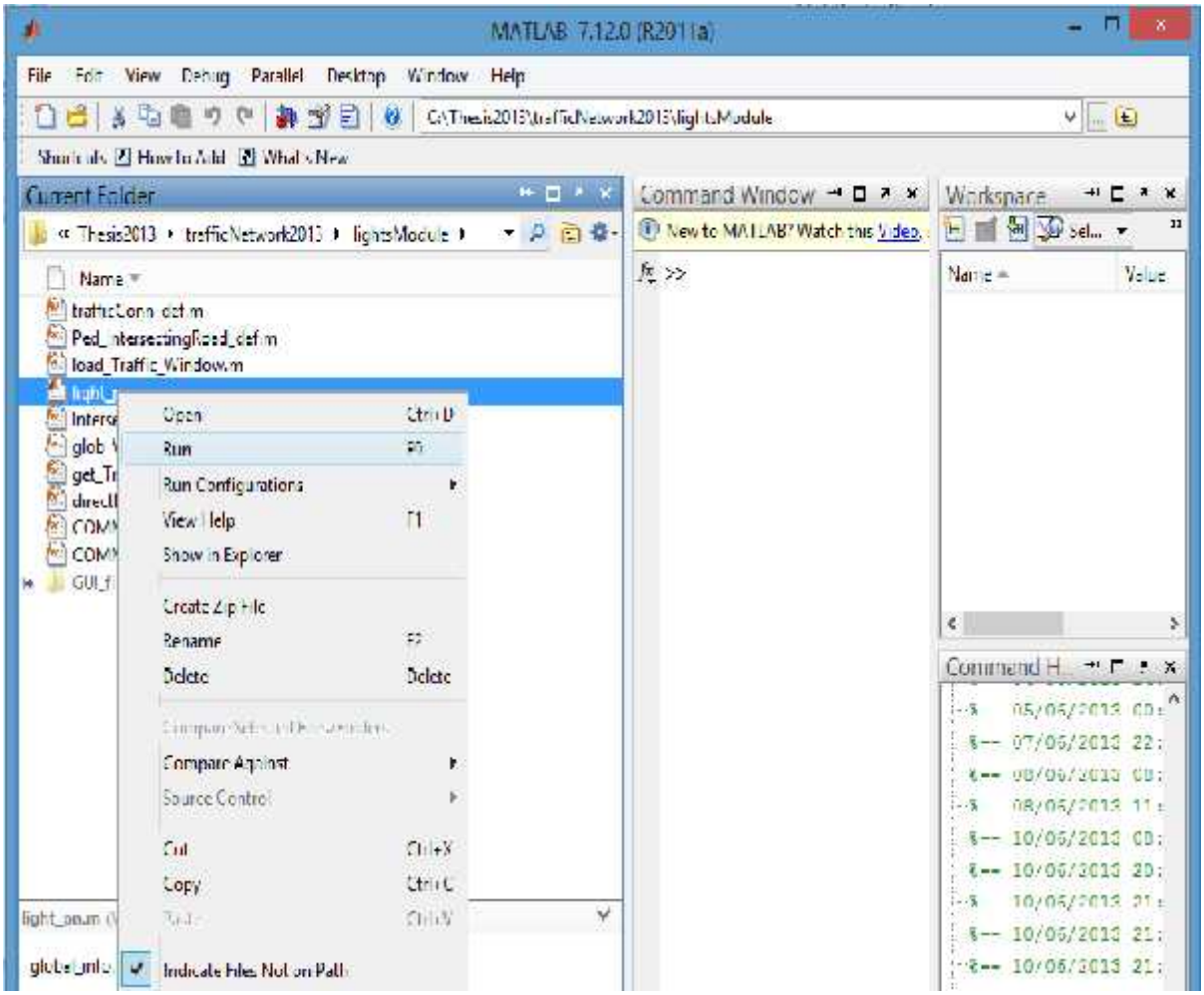


The path of the GUI .jar file is added in the MSF file. The .jar file is found in the folder named GUI_file found below the COMMON_POST file. To add a path, just open light_pn.m file in notepad, then change"C:" to the location where the unzipped Thesis2013 folder is.



User Manual

1. Right click on the MSF file name light_pn.m file and click run.



2. The GUI interface shows
3. Click on any of the start buttons to run the simulation

Code

GPenSIM code

Source code 1: light_pn.m

```
clear all; clc;

javaaddpath('C:\Thesis2013\trafficNetwork2013\lightsModule\GUI_file\Road.jar');
import roadPackage.*;

global global_info;
 global_info.REAL_TIME = 1;
global_info.STOP_AT =current_clock(3)+[0 2 0];
global_info.CASE=1;
 global a;
 global trafGlobal;
 glob_Variables();

trafGlobal=load_Traffic_Window();
a=get_Traffic_Inputs();

png = petrinetgraph({'directRoad_def','trafficConn_def','IntersectingRoad_def','Ped_IntersectingRoad_def'});

dynamic.initial_markings ={'pTrafBank',20,'pDiRedOn',1,'pDiPedPrz',1,'pDiEmgPrz',1,'pDReStart',10,...
   'pNSiRedOn',1,'pWEiRedOn',1,'pIntEmgPrz',1,'pIntReStart',10,...
   'pNSdRedOn',1,'pWEdRedOn',1,'pNSlRedOn',1,'pWElRedOn',1,'pNSrRedOn',1,'pWErRedOn',1,...
   'pIpEmgPrz',1,'pERestart',10,'pPedPrzNS',1,'pPedPrzWE',1};

dynamic.firing_times                                                                                    =
{'tDiStart',poissrnd(1,1,1),'tDiRedOn',poissrnd(2,1,1),'tDiYel',poissrnd(5,1,1),'tDiGreen',poissrnd(4,1,1),'tDiRedYel',poissrnd(
3,1,1),...
   'tDiPedPrz',poissrnd(4,1,1),'tDiEmgYel',poissrnd(2,1,1),'tDiPedGrn',poissrnd(7,1,1),'tDiPedEnd',poissrnd(10,1,1),...
%direct firing times
   'tNSiRedOn',poissrnd(2,1,1),'tNSiYel',poissrnd(3,1,1),'tNSiGreen',poissrnd(4,1,1),'tNSiRedYel',poissrnd(3,1,1),...
   'tWEiRedOn',poissrnd(3,1,1),'tWEiYel',poissrnd(4,1,1),'tWEiGreen',poissrnd(5,1,1),'tWEiRedYel',poissrnd(4,1,1),...
%intersection firing times
   'tNSdRedOn',poissrnd(2,1,1),'tNSdYel',poissrnd(3,1,1),'tNSdGreen',poissrnd(4,1,1),'tNSdRedYel',poissrnd(3,1,1),...

'tWEdRedOn',poissrnd(3,1,1),'tWEdYel',poissrnd(4,1,1),'tWEdGreen',poissrnd(5,1,1),'tWEdRedYel',poissrnd(4,1,1),...%Direc
t ped intersection firing time
   'tNSlRedOn',poissrnd(2,1,1),'tNSlYel',poissrnd(3,1,1),'tNSlGreen',poissrnd(4,1,1),'tNSlRedYel',poissrnd(3,1,1),...
   'tWElRedOn',poissrnd(3,1,1),'tWElYel',poissrnd(4,1,1),'tWElGreen',poissrnd(5,1,1),'tWElRedYel',poissrnd(4,1,1),...    %Left
```

```
ped intersection firing time
  'tNSrRedOn',poissrnd(2,1,1),'tNSrYel',poissrnd(3,1,1),'tNSrGreen',poissrnd(4,1,1),'tNSrRedYel',poissrnd(3,1,1),...
  'tWErRedOn',poissrnd(3,1,1),'tWErYel',poissrnd(4,1,1),'tWErGreen',poissrnd(5,1,1),'tWErRedYel',poissrnd(4,1,1),...%Right ped intersection firing times
  'tIpEmgPrz',poissrnd(2,1,1),'tIpEYel',poissrnd(5,1,1),...%firing time for emergency cycle of intersection with pedestrian crossing
  'tPedPrzNS',poissrnd(3,1,1),'tPedGrnNS',poissrnd(6,1,1),'tPedEndNS',poissrnd(10,1,1),...
  'tPedPrzWE',poissrnd(3,1,1),'tPedGrnWE',poissrnd(6,1,1),'tPedEndWE',poissrnd(10,1,1)};

sim = gpensim(png, dynamic);
print_statespace(sim);
```

Source code 2: trafficConn_def.m

```
function [png]=trafficConn_def()
png.PN_name = 'Connection for all traffic types Petri Net definition';
png.set_of_Ps ={'pTrafBank'};
png.set_of_Ts = {'tDiTrBank','tInTrBank','tPdTrBank'};
png.set_of_As =  {'pTrafBank','tDiTrBank',1,'tDiTrBank','pDiStart',1,... %arc definition of transition linking traffic control and directRoad_def
  'pTrafBank','tInTrBank',1,'tInTrBank','pNSiStart',1,...%arc definition of transition linking traffic control and IntersectingRoad_def
  'pTrafBank','tPdTrBank',1,'tPdTrBank','pIpedBank',1};%arc definition of transition linking traffic control and Ped_IntersectingRoad_def
```

Source code 3: directRoad_def.m

```
function [png]=directRoad_def()
png.PN_name = 'Direct road petri net definition file';
png.set_of_Ps = {'pDiStart','pDiRedOn','pDiRedYel','pDiGreen','pDiYel',...
  'pDiEmgPrz','pDiEmgSt','pDiEmgYel','pDiEmgEnd','pDReStart',...
  'pDiPedPrz','pDiPedSt','pDiPedGrn','pDiPedEnd'};
png.set_of_Ts = {'tDiStart','tDiRedOn','tDiRedYel','tDiGreen','tDiYel',...
  'tDiEmgPrz','tDiEmgSt','tDiEmgYel','tDiEmgEnd',...
  'tDiPedPrz','tDiPedSt','tDiPedGrn','tDiPedEnd'};
png.set_of_As = {'pDiStart','tDiStart',1,'pDReStart','tDiEmgEnd',1, 'pTrafBank','tDiPedEnd',1,...
  'tDiStart','pDiRedOn',1,'pDiRedOn','tDiRedOn',2,'tDiRedOn','pDiRedYel',1,...
  'pDiRedYel','tDiRedYel',1,'tDiRedYel','pDiGreen',1,'pDiGreen','tDiGreen',1,...
  'tDiGreen','pDiYel',1,'pDiYel','tDiYel',1,'tDiYel','pDiRedOn',1,'tDiYel','pTrafBank',2,...
  'pDiRedOn','tDiEmgPrz',2,'pDiEmgPrz','tDiEmgPrz',1,'tDiEmgPrz','pDiEmgSt',1,...
```

```
'pDiEmgSt','tDiEmgSt',1,'tDiEmgSt','pDiEmgYel',1,'pDiEmgYel','tDiEmgYel',1,...
'tDiEmgYel','pDiEmgEnd',1,'pDiEmgEnd','tDiEmgEnd',1,'tDiEmgEnd','pDiRedOn',1,'tDiEmgEnd','pDiEmgPrz',1,...
'pDiRedOn','tDiPedPrz',2,'pDiPedPrz','tDiPedPrz',1,'tDiPedPrz','pDiPedSt',1,...
'pDiPedSt','tDiPedSt',1,'tDiPedSt','pDiPedGrn',1,'pDiPedGrn','tDiPedGrn',1,...
'tDiPedGrn','pDiPedEnd',1,'pDiPedEnd','tDiPedEnd',1,'tDiPedEnd','pDiRedOn',2,'tDiPedEnd','pDiPedPrz',1};
```

Source code 4: IntersectingRoad_def.m

```
function [png]=IntersectingRoad_def()
png.PN_name = 'Dntersecting road petri net definition file';
png.set_of_Ps = {'pNSiStart','pNSiRedOn','pNSiRedYel','pNSiGreen','pNSiYel','pIntEmgPrz',...
   'pIntEmgSt','pIntEmgYel','pIntEmgEnd','pIntReStart','pWEiStart','pWEiRedOn','pWEiRedYel',...
   'pWEiGreen','pWEiYel'};
png.set_of_Ts = {'tNSiStart','tNSiRedOn','tNSiRedYel','tNSiGreen','tNSiYel','tIntEmgPrz',...
   'tIntEmgSt','tIntEmgYel','tIntEmgEnd','tWEiStart','tWEiRedOn','tWEiRedYel','tWEiGreen','tWEiYel'};
png.set_of_As = {'pNSiStart','tNSiStart',1,'tNSiStart','pNSiRedOn',1,'pNSiRedOn','tNSiRedOn',2,...
   'tNSiRedOn','pNSiRedYel',1,'pNSiRedYel','tNSiRedYel',1,'tNSiRedYel','pNSiGreen',1,...
   'pNSiGreen','tNSiGreen',1,'tNSiGreen','pNSiYel',1,'pNSiYel','tNSiYel',1,'tNSiYel','pWEiStart',1,...
   'pWEiStart','tWEiStart',1,'tWEiStart','pWEiRedOn',1,'pWEiRedOn','tWEiRedOn',2,'tWEiRedOn','pWEiRedYel',1,...
   'pWEiRedYel','tWEiRedYel',1,'tWEiRedYel','pWEiGreen',1,'pWEiGreen','tWEiGreen',1,'tWEiGreen','pWEiYel',1,...
   'pWEiYel','tWEiYel',1,'tWEiYel','pNSiRedOn',1,'tWEiYel','pWEiRedOn',1,'tWEiYel','pTrafBank',2,...
   'pNSiRedOn','tIntEmgPrz',1,'pWEiRedOn','tIntEmgPrz',1,'pIntEmgPrz','tIntEmgPrz',1,'tIntEmgPrz','pIntEmgSt',1,...
   'pIntEmgSt','tIntEmgSt',1,'tIntEmgSt','pIntEmgYel',1,'pIntEmgYel','tIntEmgYel',1,'tIntEmgYel','pIntEmgEnd',1,...
   'pIntEmgEnd','tIntEmgEnd',1,'pIntReStart','tIntEmgEnd',1,'tIntEmgEnd','pIntEmgPrz',1,'tIntEmgEnd','pNSiRedOn',1,...
   'tIntEmgEnd','pWEiRedOn',1,};
```

Source code 5: Ped_IntersectingRoad_def.m

```
function [png]=Ped_IntersectingRoad_def()
png.PN_name = 'Pedestrian Intersecting road petri net definition file';
png.set_of_Ps = {'pIpedBank','pDStart',...
   'pNSdRedOn','pNSdRedYel','pNSdGreen','pNSdYel','pWEdRedOn','pWEdRedYel','pWEdGreen','pWEdYel',...
   'pLStart','pNSlRedOn','pNSlRedYel','pNSlGreen','pNSlYel','pWElRedOn','pWElRedYel','pWElGreen','pWElYel',...
   'pRStart','pNSrRedOn','pNSrRedYel','pNSrGreen','pNSrYel','pWErRedOn','pWErRedYel','pWErGreen','pWErYel',...
   'pIpEmgPrz','pIpEStart','pIpEYel','pIpEmgEnd','pERestart',...
   'pPedPrzNS','pPedStNS','pPedGrnNS','pPedEndNS','pPedPrzWE','pPedStWE','pPedGrnWE','pPedEndWE'};
png.set_of_Ts = {'tpedDBank','tpedLBank','tpedRBank',...
```

```
'tNSdStart','tNSdRedOn','tNSdRedYel','tNSdGreen','tNSdYel','tWEdStart','tWEdRedOn','tWEdRedYel','tWEdGreen','tWEdYel'
,...
   'tNSlStart','tNSlRedOn','tNSlRedYel','tNSlGreen','tNSlYel','tWElStart','tWElRedOn','tWElRedYel','tWElGreen','tWElYel',...

'tNSrStart','tNSrRedOn','tNSrRedYel','tNSrGreen','tNSrYel','tWErStart','tWErRedOn','tWErRedYel','tWErGreen','tWErYel',...
   'tIpEmgPrz','tIpEStart','tIpEYel','tIpEmgEnd',...
   'tPedPrzNS','tPedStNS','tPedGrnNS','tPedEndNS','tPedPrzWE','tPedStWE','tPedGrnWE','tPedEndWE'};
png.set_of_As = {'pIpedBank','tpedDBank',1,'tpedDBank','pDStart',1,...
   'pDStart','tNSdStart',1,'tNSdStart','pNSdRedOn',1,'pNSdRedOn','tNSdRedOn',2,'tNSdRedOn','pNSdRedYel',1,...
   'pNSdRedYel','tNSdRedYel',1,'tNSdRedYel','pNSdGreen',1,'pNSdGreen','tNSdGreen',1,'tNSdGreen','pNSdYel',1,...
   'pNSdYel','tNSdYel',1,'tNSdYel','pNSdRedOn',1,'tNSdYel','pDStart',1,...
   'pDStart','tWEdStart',1,'tWEdStart','pWEdRedOn',1,'pWEdRedOn','tWEdRedOn',2,'tWEdRedOn','pWEdRedYel',1,...
   'pWEdRedYel','tWEdRedYel',1,'tWEdRedYel','pWEdGreen',1,'pWEdGreen','tWEdGreen',1,'tWEdGreen','pWEdYel',1,...
   'pWEdYel','tWEdYel',1,'tWEdYel','pWEdRedOn',1,'tWEdYel','pIpedBank',1,... %direct traffic model
   'pIpedBank','tpedLBank',1,'tpedLBank','pLStart',1,...
   'pLStart','tNSlStart',1,'tNSlStart','pNSlRedOn',1,'pNSlRedOn','tNSlRedOn',2,'tNSlRedOn','pNSlRedYel',1,...
   'pNSlRedYel','tNSlRedYel',1,'tNSlRedYel','pNSlGreen',1,'pNSlGreen','tNSlGreen',1,'tNSlGreen','pNSlYel',1,...
   'pNSlYel','tNSlYel',1,'tNSlYel','pNSlRedOn',1,'tNSlYel','pLStart',1,...
   'pLStart','tWElStart',1,'tWElStart','pWElRedOn',1,'pWElRedOn','tWElRedOn',2,'tWElRedOn','pWElRedYel',1,...
   'pWElRedYel','tWElRedYel',1,'tWElRedYel','pWElGreen',1,'pWElGreen','tWElGreen',1,'tWElGreen','pWElYel',1,...
   'pWElYel','tWElYel',1,'tWElYel','pWElRedOn',1,'tWElYel','pIpedBank',1,... %left turn traffic model
   'pIpedBank','tpedRBank',1,'tpedRBank','pRStart',1,...
   'pRStart','tNSrStart',1,'tNSrStart','pNSrRedOn',1,'pNSrRedOn','tNSrRedOn',2,'tNSrRedOn','pNSrRedYel',1,...
   'pNSrRedYel','tNSrRedYel',1,'tNSrRedYel','pNSrGreen',1,'pNSrGreen','tNSrGreen',1,'tNSrGreen','pNSrYel',1,...
   'pNSrYel','tNSrYel',1,'tNSrYel','pNSrRedOn',1,'tNSrYel','pRStart',1,...
   'pRStart','tWErStart',1,'tWErStart','pWErRedOn',1,'pWErRedOn','tWErRedOn',2,'tWErRedOn','pWErRedYel',1,...
   'pWErRedYel','tWErRedYel',1,'tWErRedYel','pWErGreen',1,'pWErGreen','tWErGreen',1,'tWErGreen','pWErYel',1,...
   'pWErYel','tWErYel',1,'tWErYel','pWErRedOn',1,'tWErYel','pTrafBank',1,... % right turn traffic model

'pIpedBank','tIpEmgPrz',1,'pIpEmgPrz','tIpEmgPrz',1,'tIpEmgPrz','pIpEStart',1,'pIpEStart','tIpEStart',1,'tIpEStart','pIpEYel',1,...

'pIpEYel','tIpEYel',1,'tIpEYel','pIpEmgEnd',1,'pIpEmgEnd','tIpEmgEnd',1,'pERestart','tIpEmgEnd',1,'tIpEmgEnd','pIpEmgPrz',1,
... % emergecy cycle for intersection pedestrian

'pPedPrzNS','tPedPrzNS',1,'pIpedBank','tPedPrzNS',1,'tPedPrzNS','pWEdRedOn',1,'tPedPrzNS','pPedStNS',1,'pPedStNS','tPe
dStNS',1,...

'tPedStNS','pPedGrnNS',1,'pPedGrnNS','tPedGrnNS',1,'tPedGrnNS','pPedEndNS',1,'pPedEndNS','tPedEndNS',1,'tPedEndNS','
pPedPrzNS',1,...%

'pPedPrzWE','tPedPrzWE',1,'pIpedBank','tPedPrzWE',1,'tPedPrzWE','pNSdRedOn',1,'tPedPrzWE','pPedStWE',1,'pPedStWE','
tPedStWE',1,...
```

```
'tPedStWE','pPedGrnWE',1,'pPedGrnWE','tPedGrnWE',1,'tPedGrnWE','pPedEndWE',1,'pPedEndWE','tPedEndWE',1,'pDStart
','tPedEndWE',1,...
   'tPedEndWE','pPedPrzWE',1,'tPedEndWE','pIpedBank',1};%
```

Source code 6: glob_Variables.m

```
function []=glob_Variables()

global direct;
global intersection;
global intersectionPed;
global global_info;
direct.flag=0;
direct.Pedestrian='off';
direct.Emergency='off';
direct.Restart='off';
direct.Direct='off';
direct.dirRun='false';

intersection.flag=0;
intersection.run='off';
intersection.Emergency='off';
intersection.Restart='off';
intersection.intRun='false';

intersectionPed.flag=0;
intersectionPed.run='off';
intersectionPed.Emergency='off';
intersectionPed.Restart='off';
intersectionPed.PedestrianNS='off';
intersectionPed.PedestrianSN='off';
intersectionPed.PedestrianWE='off';
intersectionPed.PedestrianEW='off';
intersectionPed.pedRun='false';

global_info.semafor=1;
global_info.semaforPRE=1;
global_info.pDSemafor=1;
```

```
global_info.pLSemafor=1;
global_info.pRSemafor=1;
end
```

Source code 7: load_Traffic_Window.m

```
function [traf]=load_Traffic_Window()
 traf.window=roadPackage.roadGUI;
end
```

Source code 8: get_Traffic_Inputs.m

```
function [traffic_inputs]=get_Traffic_Inputs()
global direct;
%global trafGlobal;
global vv;
global intersection;
global intersectionPed;
import roadPackage.*;

vv=roadPackage.roadFxnClass();
s=vv.getGlobalValOne;
while(s==3)
   vv=roadPackage.roadFxnClass();
   s=vv.getGlobalValOne;
end

Emg=vv.getGlobalDirectEmergency();
Dir=vv.getGlobalDirectStart();
Ped=vv.getGlobalDirectPedestrian();
direct.dirRun=vv.getGlobalDirectRun();

if strcmp(direct.dirRun,'true'),
   if strcmp(Emg,'true'),
      direct.Emergency='on';
   else
      direct.Emergency='off';
   end;

   if strcmp(Dir,'true'),
     direct.Direct='on';
   else
```

```
    direct.Direct='off';
  end;

  if strcmp(Ped,'true'),
    direct.Pedestrian='on';
  else
    direct.Pedestrian='off';
  end;
end

IntsxnEmg=vv.getGlobalIntersectionEmergency();
Intsxn=vv.getGlobalIntersectionStart();
intersection.intRun=vv.getGlobalIntRun();

if strcmp(intersection.intRun,'true'),
  if strcmp(IntsxnEmg,'true'),
     intersection.Emergency='on';
  else
     intersection.Emergency='off';
  end;

  if strcmp(Intsxn,'true'),
    intersection.run='on';
  else
    intersection.run='off';
  end;
end

%%
pedEmg=vv.getGlobalIntersectionPdEmergency();
pedStart=vv.getGlobalIntersectionPdStart();
pedPedestrianNS=vv.getGlobalIntersectionPdPedestrianNS();
pedPedestrianWE=vv.getGlobalIntersectionPdPedestrianWE();
pedPedestrianSN=vv.getGlobalIntersectionPdPedestrianSN();
pedPedestrianEW=vv.getGlobalIntersectionPdPedestrianEW();
intersectionPed.pedRun=vv.getGlobalIntersectionPdRun();

if strcmp(intersectionPed.pedRun,'true'),
  if strcmp(pedEmg,'true'),
     intersectionPed.Emergency='on';
  else
     intersectionPed.Emergency='off';
```

```
    end;

    if strcmp(pedStart,'true'),
      intersectionPed.run='on';
    else
      intersectionPed.run='off';
    end;

    if strcmp(pedPedestrianNS,'true'),
      intersectionPed.PedestrianNS='on';
     % global_info.semafor=5;
    else
      intersectionPed.PedestrianNS='off';
    end;
     if strcmp(pedPedestrianWE,'true'),
      intersectionPed.PedestrianWE='on';
     % global_info.semafor=5;
    else
      intersectionPed.PedestrianWE='off';
    end;
     if strcmp(pedPedestrianSN,'true'),
      intersectionPed.PedestrianSN='on';
      %global_info.semafor=5;
    else
      intersectionPed.PedestrianSN='off';
    end;
     if strcmp(pedPedestrianEW,'true'),
      intersectionPed.PedestrianEW='on';
      %global_info.semafor=5;
    else
      intersectionPed.PedestrianEW='off';
    end;
end

traffic_inputs.val=s;%value to show start button has been clicked

end
```

Source code 9: COMMON_PRE.m

```
function[fire,transition]=COMMON_PRE(transition)
 global direct;
```

```
global trafGlobal;
global intersection;
global intersectionPed;
global global_info;
if strcmp(transition.name,'tDiTrBank'),%transition linking direct road with the toke source
 if strcmp(direct.dirRun,'true'),
    if eq(direct.flag,0),
       direct.flag=1;
       fire=1;
       trafGlobal.window.setDirectionRed(111);
       trafGlobal.window.setDirectionRed(222);
     else
       fire=0;
    end
 else
    fire=0;
 end
return;
elseif strcmp(transition.name,'tInTrBank'),%transition linking simple intersecting road with the toke source
 if strcmp(intersection.intRun,'true'),
    if eq(intersection.flag,0),
       intersection.flag=1;fire=1;
       trafGlobal.window.setDirectionRed(1);
       trafGlobal.window.setDirectionRed(2);
     else
       fire=0;
    end
 else
    fire=0;
 end
return;

elseif strcmp(transition.name,'tPdTrBank'),%transition linking complicated intersecting road with the token source
 if strcmp(intersectionPed.pedRun,'true'),
    if eq(intersectionPed.flag,0),
       intersectionPed.flag=1;fire=1;
       trafGlobal.window.setDirectionRed(11);
       trafGlobal.window.setDirectionRed(22);
       trafGlobal.window.setDirectionRed(777);
     else
       fire=0;
    end
```

```
 else
     fire=0;
 end
return;

elseif strcmp(transition.name,'tDiEmgPrz'),
   if strcmp(direct.Emergency,'on'),
      fire=1;
   else
      fire=0;
   end
return;

elseif strcmp(transition.name,'tIntEmgPrz'),
   if strcmp(intersection.Emergency,'on'),
      fire=1;
   else
      fire=0;
   end
return;
elseif strcmp(transition.name,'tIpEmgPrz'),
   if strcmp(intersectionPed.Emergency,'on'),
      fire=1;
   else
      fire=0;
   end
return;

elseif strcmp(transition.name,'tDiEmgEnd'),
   if strcmp(direct.Direct,'on'),
      direct.Pedestrian='off';
      fire=1;
   else
      fire=0;
   end
return;

elseif strcmp(transition.name,'tIntEmgEnd'),
   if strcmp(intersection.run,'on'),
      fire=1;
   else
      fire=0;
```

```
   end
return;
%%
elseif strcmp(transition.name,'tIpEmgEnd'),
   if strcmp(intersectionPed.run,'on'),
     % direct.Pedestrian='off';
      disp('emg yellow');
      fire=1;
   else
      fire=0;
   end
return;

elseif strcmp(transition.name,'tIntEmgYel'),
  trafGlobal.window.setDirectionYellow(1);
  trafGlobal.window.setDirectionYellow(2);
  fire=1;
  return;

 elseif strcmp(transition.name,'tIpEYel'),% intersection emg yellow
  disp('emg yellow');
  trafGlobal.window.setDirectionYellow(11);
  trafGlobal.window.setDirectionYellow(22);
  trafGlobal.window.setDirectionBlack(555);
  fire=1;
  return;

elseif strcmp(transition.name,'tDiEmgYel'),
  trafGlobal.window.setDirectionYellow(111);
  trafGlobal.window.setDirectionBlack(222);
  fire=1;
  return;

elseif strcmp(transition.name,'tWEiStart'),
  trafGlobal.window.setDirectionRed(1);
  fire=1;
  return

elseif strcmp(transition.name,'tDiRedOn'),
  if strcmp(direct.Emergency,'on'),
      fire=0;
  else
```

```
    if strcmp(direct.Pedestrian,'on'),
        fire=0;
    else
        fire=1;
    end
  end
return;

elseif strcmp(transition.name,'pNSiRedOn'),
  if strcmp(intersection.Emergency,'on'),
     fire=0;
  else
     fire=1;
  end
return;

elseif strcmp(transition.name,'pWEiRedOn'),
  if strcmp(intersection.Emergency,'on'),
     fire=0;
  else
     fire=1;
  end
return;

elseif strcmp(transition.name,'tDiPedPrz'),
   if strcmp(direct.Pedestrian,'on'),
      if strcmp(direct.Emergency,'on'),
         fire=0;
      else
         %trafGlobal.window.setDirectionRed(222);
         fire=1;
      end
   else
      fire=0;
   end
return;

elseif strcmp(transition.name,'tDiYel'),
 trafGlobal.window.setDirectionYellow(111);
 fire=1;
return;
```

```
elseif strcmp(transition.name,'tNSiYel'),
 trafGlobal.window.setDirectionYellow(1);
 fire=1;
return;
elseif strcmp(transition.name,'tWEiYel'),
 trafGlobal.window.setDirectionYellow(2);
 fire=1;
return;

elseif (strcmp(transition.name,'tNSdYel')||strcmp(transition.name,'tNSlYel')||strcmp(transition.name,'tNSrYel')),
 trafGlobal.window.setDirectionYellow(11);
 fire=1;
return;

elseif (strcmp(transition.name,'tWEdYel')||strcmp(transition.name,'tWElYel')||strcmp(transition.name,'tWErYel')),
 trafGlobal.window.setDirectionYellow(22);
 fire=1;
return;

elseif strcmp(transition.name,'tDiRedYel'),
 trafGlobal.window.setDirectionRedYellow(111);
  fire=1;
return;

elseif strcmp(transition.name,'tNSiRedYel'),
 trafGlobal.window.setDirectionRedYellow(1);
  fire=1;
return;

elseif strcmp(transition.name,'tWEiRedYel'),
 trafGlobal.window.setDirectionRedYellow(2);
  fire=1;
return;

elseif
(strcmp(transition.name,'tNSdRedYel')||strcmp(transition.name,'tNSlRedYel')||strcmp(transition.name,'tNSrRedYel')),
 trafGlobal.window.setDirectionRedYellow(11);
  fire=1;
return;

elseif
(strcmp(transition.name,'tWEdRedYel')||strcmp(transition.name,'tWElRedYel')||strcmp(transition.name,'tWErRedYel')),
```

```
 trafGlobal.window.setDirectionRedYellow(22);
  fire=1;
return;

elseif strcmp(transition.name,'tDiGreen'),
 trafGlobal.window.setDirectionGreen(111);
  fire=1;
return;

elseif strcmp(transition.name,'tNSiGreen'),
 trafGlobal.window.setDirectionGreen(1);
  fire=1;
return;

elseif strcmp(transition.name,'tWEiGreen'),
 trafGlobal.window.setDirectionGreen(2);
  fire=1;
return;

elseif strcmp(transition.name,'tNSdGreen'),
 trafGlobal.window.setDirectionGreen(11);
  fire=1;
return;

elseif strcmp(transition.name,'tNSlGreen'),
 trafGlobal.window.setDirectionGreenL(11);
  fire=1;
return;

elseif strcmp(transition.name,'tNSrGreen'),
 trafGlobal.window.setDirectionGreenR(11);
  fire=1;
return;

elseif strcmp(transition.name,'tWEdGreen'),
 trafGlobal.window.setDirectionGreen(22);
  fire=1;
return;


elseif strcmp(transition.name,'tWElGreen'),
 trafGlobal.window.setDirectionGreenL(22);
```

```
  fire=1;
return;

elseif strcmp(transition.name,'tWErGreen'),
 trafGlobal.window.setDirectionGreenR(22);
  fire=1;
return;

elseif strcmp(transition.name,'tDiPedSt'),
 trafGlobal.window.setDirectionRed(111);
  fire=1;
return;

elseif strcmp(transition.name,'tDiPedGrn'),
   disp('ped green');
 trafGlobal.window.setDirectionGreen(222);
  fire=1;
return;

elseif strcmp(transition.name,'tPedGrnNS'),
 if strcmp(intersectionPed.PedestrianNS,'on')
    trafGlobal.window.setDirectionGreen(333);
 else
    trafGlobal.window.setDirectionGreen(444);
 end
   fire=1;
return;

elseif strcmp(transition.name,'tPedGrnWE'),
 if strcmp(intersectionPed.PedestrianWE,'on')
    trafGlobal.window.setDirectionGreen(555);
 else
    trafGlobal.window.setDirectionGreen(666);
 end
   fire=1;
return;
%% pre conditions for intersection with pedestrian crossing
elseif strcmp(transition.name,'tpedDBank'),
      if(global_info.semafor==1||global_info.semafor==3),
         if strcmp(intersectionPed.Emergency,'on'),
            fire=0;
         else
```

```
            if
(strcmp(intersectionPed.PedestrianNS,'on')||strcmp(intersectionPed.PedestrianSN,'on')||strcmp(intersectionPed.PedestrianWE,'on')||strcmp(intersectionPed.PedestrianEW,'on')),
                global_info.semaforPRE=global_info.semafor; global_info.semafor=5;fire=0;
            else
                fire=1;
            end;
        end
    else
        fire=0;
    end;
return;
elseif strcmp(transition.name,'tpedLBank'),
    if(global_info.semafor==2),
        if strcmp(intersectionPed.Emergency,'on'),
            fire=0;
        else
            if
(strcmp(intersectionPed.PedestrianNS,'on')||strcmp(intersectionPed.PedestrianSN,'on')||strcmp(intersectionPed.PedestrianWE,'on')||strcmp(intersectionPed.PedestrianEW,'on')),
                global_info.semaforPRE=global_info.semafor;global_info.semafor=5; fire=0;
            else
                fire=1;
            end;
        end
    else
        fire=0;
    end;
  return;
elseif strcmp(transition.name,'tpedRBank'),
    if(global_info.semafor==4),
        if strcmp(intersectionPed.Emergency,'on'),
            fire=0;
        else
            if
(strcmp(intersectionPed.PedestrianNS,'on')||strcmp(intersectionPed.PedestrianSN,'on')||strcmp(intersectionPed.PedestrianWE,'on')||strcmp(intersectionPed.PedestrianEW,'on')),
                global_info.semaforPRE=global_info.semafor; global_info.semafor=5; fire=0;
            else
                fire=1;
            end;
        end
```

```
    else
      fire=0;
    end;
return;
elseif strcmp(transition.name,'tNSdStart'),
    if(global_info.pDSemafor==1),
      fire=1;
    else
      fire=0;
    end;
return;
elseif strcmp(transition.name,'tWEdStart'),
    if(global_info.pDSemafor==2),
      fire=1;
    else
      fire=0;
    end;
return;

elseif strcmp(transition.name,'tNSlStart'),
    if(global_info.pLSemafor==1),
      fire=1;
    else
      fire=0;
    end;
return;
elseif strcmp(transition.name,'tWElStart'),
    if(global_info.pLSemafor==2),
      fire=1;
    else
      fire=0;
    end;
return;

elseif strcmp(transition.name,'tNSrStart'),
    if(global_info.pRSemafor==1),
      fire=1;
    else
      fire=0;
    end;
return;
elseif strcmp(transition.name,'tWErStart'),
```

```
      if(global_info.pRSemafor==2),
        fire=1;
      else
        fire=0;
      end;
return;

elseif strcmp(transition.name,'tPedPrzNS'),
   if (strcmp(intersectionPed.PedestrianNS,'on')|| strcmp(intersectionPed.PedestrianSN,'on')),
       if strcmp(intersectionPed.Emergency,'on'),
         fire=0;
       else
         if(global_info.semafor==5),
            fire=1;
           % global_info.semafor=6;
         else
            fire=0;
         end;
       end;
   else
      fire=0;
   end;
return;

elseif strcmp(transition.name,'tPedPrzWE'),
   if (strcmp(intersectionPed.PedestrianWE,'on')|| strcmp(intersectionPed.PedestrianEW,'on')),
       if strcmp(intersectionPed.Emergency,'on'),
         fire=0;
       else
         if(global_info.semafor==5),
            global_info.pDSemafor=3;
            global_info.semafor=6;
            fire=1;
         else
            fire=0;
         end;
       end;
   else
      fire=0;
   end;
return;
```

```
else
end
fire=1;
```

Source code 10: COMMON_POST.m

```
function [] = COMMON_POST(transition)
 global direct;
global a;
global vv;
global trafGlobal;
global counter;
global pcounter;
global intersection;
global intersectionPed;
global global_info;

counter=0;pcounter=0;

if strcmp(transition.name,'tDiYel'),
   direct.flag=0;
   return;
elseif strcmp(transition.name,'tWEiYel'),
    trafGlobal.window.setDirectionRed(2);
   intersection.flag=0;
   return;
elseif strcmp(transition.name,'tDiEmgEnd'),
    vv.ResetDirectPedestrian();
   direct.flag=0;
   return;

 elseif strcmp(transition.name,'tIpEmgEnd'),
   global_info.semafor=1;
   global_info.pDSemafor=1;
   global_info.pLSemafor=1;
   global_info.pRSemafor=1;
   intersectionPed.flag=0;
   return;
elseif strcmp(transition.name,'tWErYel'),
   trafGlobal.window.setDirectionRed(22);
   disp(global_info.semafor);
```

```
   intersectionPed.flag=0;
   return;
elseif (strcmp(transition.name,'tWEdYel')||strcmp(transition.name,'tWElYel')),
   trafGlobal.window.setDirectionRed(22);
   return;
elseif (strcmp(transition.name,'tNSdYel')||strcmp(transition.name,'tNSlYel')||strcmp(transition.name,'tNSrYel')),
   trafGlobal.window.setDirectionRed(11);
   return;

elseif strcmp(transition.name,'tIntEmgEnd'),
   trafGlobal.window.setDirectionRed(1);
   trafGlobal.window.setDirectionRed(2);
   return;

elseif strcmp(transition.name,'tDiEmgYel'), % in this post file, it will be querying the java code to see if restart in the direct model has been clicked
   while (strcmp(direct.Direct,'off'))
     a=get_Traffic_Inputs();
   end
   return;

elseif strcmp(transition.name,'tIntEmgYel'), % in this post file, it will be querying the java code to see if restart in the intersection model has been clicked
   while (strcmp(intersection.run,'off'))
     a=get_Traffic_Inputs();
   end
   return;

elseif strcmp(transition.name,'tIpEYel'), % in this post file, it will be querying if restart in the intersection Pedestrian model has been clicked
   while (strcmp(intersectionPed.run,'off'))
     a=get_Traffic_Inputs();
   end
   return;
elseif strcmp(transition.name,'tDiPedGrn'),
  while(counter<3)
      counter=counter+1;
  end
  counter=0;
   return;

elseif strcmp(transition.name,'tPedGrnNS'),
```

```
  while(pcounter<4)
      pcounter=pcounter+1;
     % disp(pcounter);
  end
  pcounter=0;
   return;

elseif strcmp(transition.name,'tPedGrnWE'),
  while(pcounter<3)
      pcounter=pcounter+1;
      disp(pcounter);
  end
  pcounter=0;
   return;

elseif strcmp(transition.name,'tDiPedEnd'),
  trafGlobal.window.setDirectionRed(222);
  vv.ResetDirectPedestrian();
   return;

elseif strcmp(transition.name,'tPedEndNS'),
  if strcmp(intersectionPed.PedestrianNS,'on')
    vv.ResetIntersectionPedestrian(1);
    trafGlobal.window.setDirectionRed(333);
  elseif strcmp(intersectionPed.PedestrianSN,'on')
    vv.ResetIntersectionPedestrian(2);
    trafGlobal.window.setDirectionRed(444);
  else
  end
 global_info.semafor=global_info.semaforPRE;
return;

elseif strcmp(transition.name,'tPedEndWE'),
  if strcmp(intersectionPed.PedestrianWE,'on')
    vv.ResetIntersectionPedestrian(3);
    trafGlobal.window.setDirectionRed(555);
  elseif strcmp(intersectionPed.PedestrianEW,'on')
    vv.ResetIntersectionPedestrian(4);
    trafGlobal.window.setDirectionRed(666);
  else
  end
 global_info.semafor=global_info.semaforPRE;
```

```
 global_info.pDSemafor=1;
% intersectionPed.flag=0;
return;

%% post condition for intersection with pedestrian
elseif strcmp(transition.name,'tpedDBank'),
   if (strcmp(intersectionPed.PedestrianNS,'on')||strcmp(intersectionPed.PedestrianSN,'on')),
    % global_info.semafor=5;
   else
     if(global_info.semafor==1)
     global_info.semafor=2;
        elseif(global_info.semafor==3)
     global_info.semafor=4;
       else
     end
   end;
  return;
elseif strcmp(transition.name,'tpedLBank'),
    if (strcmp(intersectionPed.PedestrianNS,'on')||strcmp(intersectionPed.PedestrianSN,'on')),
   else
    global_info.semafor=3;
   end
return;
elseif strcmp(transition.name,'tpedRBank'),
  if (strcmp(intersectionPed.PedestrianNS,'on')||strcmp(intersectionPed.PedestrianSN,'on')),
   else
    global_info.semafor=1;
   end
return;

elseif strcmp(transition.name,'tNSdStart'),
  global_info.pDSemafor=2;
return;
elseif strcmp(transition.name,'tWEdStart'),
  global_info.pDSemafor=1;
return;
elseif strcmp(transition.name,'tNSlStart'),
  global_info.pLSemafor=2;
return;
elseif strcmp(transition.name,'tWElStart'),
  global_info.pLSemafor=1;
return;
```

```
 elseif strcmp(transition.name,'tNSrStart'),
   global_info.pRSemafor=2;
 return;
elseif strcmp(transition.name,'tWErStart'),
   global_info.pRSemafor=1;
 return;

elseif strcmp(transition.name,'tPedPrzNS'),
   global_info.semafor=6;
elseif strcmp(transition.name,'tPedPrzWE'),
   global_info.semafor=6;
return;
else
end
```

GUI Code

Java code 1: roadGUI

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadPackage;

import java.awt.BorderLayout;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JLabel;



public class roadGUI extends JFrame{
   private static final long serialVersionUID=9876543L;
   private roadPanel lPanel;
   private controlPanel cPanel=new controlPanel();

   private roadClass N2light=new roadClass();
   private roadClass S2light=new roadClass(180);
   private roadClass W2light=new roadClass(90);
```

```
private roadClass E2light=new roadClass(270);

private roadClassT tNlight=new roadClassT();
private roadClassT tSlight=new roadClassT(180);
private roadClassT tWlight=new roadClassT(270);
private roadClassT tElight=new roadClassT(90);

private roadClassD light=new roadClassD(180);
private roadClassD leftLight=new roadClassD(01);
private roadClassD rightLight=new roadClassD(01);

private roadClassD tNlightLeft=new roadClassD(01);
private roadClassD tNlightRight=new roadClassD(01);

private roadClassD tSlightLeft=new roadClassD(02);
private roadClassD tSlightRight=new roadClassD(02);

private roadClassD tWlightLeft=new roadClassD(901);
private roadClassD tWlightRight=new roadClassD(901);

private roadClassD tElightLeft=new roadClassD(902);
private roadClassD tElightRight=new roadClassD(902);

JLabel dirLabel=new JLabel("Direct");
JLabel intLabel=new JLabel("Intersection");
JLabel intPedLabel=new JLabel("Pedestrian Int");


public  roadGUI()
{
super("test light GUI");
setBackground();
addTlyts();
setSize(1007,710);
setResizable(false);
setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
setVisible(true);
}

private void setBackground()
{
  lPanel=new roadPanel();setLayout(new BorderLayout());
```

```
  add(lPanel,BorderLayout.CENTER);
  setTrafficLabel();
 }

 private void addTlyts()
 {
    placeLyts();
    cPanel.addControlButtons(lPanel);
    cPanel.addPedestrianControlButtons(lPanel);
 }

 private void setTrafficLabel()
 {
  dirLabel.setFont(new Font("Verdana",Font.BOLD,14));
  dirLabel.setBounds(260, 1, 100, 30);
  lPanel.add(dirLabel);
  intPedLabel.setFont(new Font("Verdana",Font.BOLD,14));
  intPedLabel.setBounds(490, 1, 120, 30);
  lPanel.add(intPedLabel);
  intLabel.setFont(new Font("Verdana",Font.BOLD,14));
  intLabel.setBounds(890, 1, 100, 30);
  lPanel.add(intLabel);
 }


private void placeLyts()
{
  lPanel.add(light);lPanel.add(leftLight);lPanel.add(rightLight);
  lPanel.add(N2light);lPanel.add(S2light);lPanel.add(W2light);lPanel.add(E2light);
  lPanel.add(tNlight);lPanel.add(tSlight);lPanel.add(tWlight);lPanel.add(tElight);
  lPanel.add(tNlightLeft);lPanel.add(tNlightRight);lPanel.add(tSlightLeft);lPanel.add(tSlightRight);
  lPanel.add(tWlightLeft);lPanel.add(tWlightRight);lPanel.add(tElightLeft);lPanel.add(tElightRight);

  N2light.setBounds(717, 158, roadGlobals.horzWidthA, roadGlobals.vertHeightC);
  S2light.setBounds(891, 361, roadGlobals.horzWidthA, roadGlobals.vertHeightC);
  W2light.setBounds(889, 189, roadGlobals.horzWidthC, roadGlobals.vertHeightA);
  E2light.setBounds(686, 362, roadGlobals.horzWidthC, roadGlobals.vertHeightA);

  tNlight.setBounds(318, 128, roadGlobals.horzWidthA, roadGlobals.vertHeightD);
  tSlight.setBounds(491, 361, roadGlobals.horzWidthA, roadGlobals.vertHeightD);
  tWlight.setBounds(259, 362, roadGlobals.horzWidthD, roadGlobals.vertHeightA);
  tElight.setBounds(489, 189, roadGlobals.horzWidthD, roadGlobals.vertHeightA);
```

```
    tNlightLeft.setBounds(287, 142, roadGlobals.horzWidthA, roadGlobals.vertHeightB);
    tNlightRight.setBounds(487,142, roadGlobals.horzWidthA, roadGlobals.vertHeightB);
    tSlightLeft.setBounds(320, 406, roadGlobals.horzWidthA, roadGlobals.vertHeightB);
    tSlightRight.setBounds(520,406, roadGlobals.horzWidthA, roadGlobals.vertHeightB);

    tWlightLeft.setBounds(215, 189, roadGlobals.horzWidthB, roadGlobals.vertHeightA);
    tWlightRight.setBounds(215,362, roadGlobals.horzWidthB, roadGlobals.vertHeightA);
    tElightLeft.setBounds(577, 189, roadGlobals.horzWidthB, roadGlobals.vertHeightA);
    tElightRight.setBounds(577,360, roadGlobals.horzWidthB, roadGlobals.vertHeightA);

    light.setBounds(51, 321, roadGlobals.horzWidthA, roadGlobals.vertHeightC);
    leftLight.setBounds(51, 431, roadGlobals.horzWidthA, roadGlobals.vertHeightB);
    rightLight.setBounds(221, 431, roadGlobals.horzWidthA, roadGlobals.vertHeightB);

}

   public void setDirectionRed(int d)
  {
   switch(d)
   {
      case 1:N2light.setRed();S2light.setRed();
         break;
      case 2:W2light.setRed();E2light.setRed();
         break;
      case 11:tNlight.setRed();tSlight.setRed();
         break;
      case 22:tWlight.setRed();tElight.setRed();
         break;
      case 111:light.setRed();
         break;
     case 222:leftLight.setRed();rightLight.setRed();
         break;
     case 333:tNlightLeft.setRed();tNlightRight.setRed();
         break;
     case 444:tSlightLeft.setRed();tSlightRight.setRed();
         break;
     case 555:tWlightLeft.setRed();tWlightRight.setRed();
         break;
     case 666:tElightLeft.setRed();tElightRight.setRed();
         break;
     case 777:tElightLeft.setRed();tElightRight.setRed();tNlightLeft.setRed();tNlightRight.setRed();
```

```
   tSlightLeft.setRed();tSlightRight.setRed();tWlightLeft.setRed();tWlightRight.setRed();
   tElightLeft.setRed();tElightRight.setRed();
       break;
   }
}

public void setDirectionYellow(int d)
{
    switch(d)
 {
    case 1:N2light.setYellow();S2light.setYellow();
       break;
    case 2:W2light.setYellow();E2light.setYellow();
       break;
    case 11:tNlight.setYellow();tSlight.setYellow();
       break;
    case 22:tWlight.setYellow();tElight.setYellow();
       break;
    case 111:light.setYellow();
       break;
 }
}

 public void setDirectionGreen(int d)
{
  switch(d)
 {
    case 1:N2light.setGreen();S2light.setGreen();
       break;
    case 2:W2light.setGreen();E2light.setGreen();
       break;
    case 11:tNlight.setGreen();tSlight.setGreen();
       break;
    case 22:tWlight.setGreen();tElight.setGreen();
       break;
    case 111:light.setGreen();
       break;
    case 222:leftLight.setGreen();rightLight.setGreen();
       break;
   case 333:tNlightLeft.setGreen();tNlightRight.setGreen();
       break;
   case 444:tSlightLeft.setGreen();tSlightRight.setGreen();
```

```
        break;
    case 555:tWlightLeft.setGreen();tWlightRight.setGreen();
        break;
    case 666:tElightLeft.setGreen();tElightRight.setGreen();
        break;
 }
}

  public void setDirectionGreenL(int d)
{
 switch(d)
 {
    case 11:tNlight.setGreen1();tSlight.setGreen1();
        break;
    case 22:tWlight.setGreen1();tElight.setGreen1();
        break;
 }
}

    public void setDirectionGreenR(int d)
{
 switch(d)
 {
    case 11:tNlight.setGreen2();tSlight.setGreen2();
        break;
    case 22:tWlight.setGreen2();tElight.setGreen2();
        break;

 }
}

  public void setDirectionRedYellow(int d)
{
 switch(d)
 {
    case 1:N2light.setRedandYellow();S2light.setRedandYellow();
        break;
    case 2:W2light.setRedandYellow();E2light.setRedandYellow();
        break;
    case 11:tNlight.setRedandYellow();tSlight.setRedandYellow();
        break;
    case 22:tWlight.setRedandYellow();tElight.setRedandYellow();
```

```
            break;
        case 111:light.setRedandYellow();
            break;
    }
   }

    public void setDirectionBlack(int d)
   {
    switch(d)
   {
        case 1:N2light.setBlack();S2light.setBlack();
            break;
        case 2:W2light.setBlack();E2light.setBlack();
            break;
        case 11:tNlight.setBlack();tSlight.setBlack();
            break;
        case 22:tWlight.setBlack();tElight.setBlack();
            break;
        case 111:light.setBlack();
            break;
        case 222:leftLight.setBlack();rightLight.setBlack();
            break;
        case 444:light.setBlack();leftLight.setBlack();rightLight.setBlack();
            break;
        case
555:tNlightLeft.setBlack();tNlightRight.setBlack();tSlightLeft.setBlack();tSlightRight.setBlack();tWlightLeft.setBlack();
     tWlightRight.setBlack();tElightLeft.setBlack();tElightRight.setBlack();
            break;
    }
   }
}
```

Java code 2: roadPanel

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadPackage;
```

```
import java.awt.Graphics;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JPanel;


public class roadPanel extends JPanel
{
   private static final long serialVersionUID=12899999L;
   private static final String imagePath="Aby.jpg";
   private Image bg;


   public roadPanel()
   {
      ImageIcon bgIcon=new ImageIcon(getClass().getResource(imagePath));
      bg=bgIcon.getImage();
      setLayout(null);
   }

   public void paint(Graphics g)
   {
   g.drawImage(bg,0,0,this);
   setOpaque(false);
   super.paint(g);
   }
}
```

Java code 3: roadGlobals

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadPackage;

public class roadGlobals {
   public static int horzWidthA=30;
   public static int horzWidthB=45;
   public static int horzWidthC=60;
```

```
    public static int horzWidthD=90;

    public static int vertHeightA=30;
    public static int vertHeightB=45;
    public static int vertHeightC=60;
    public static int vertHeightD=90;

    public static int g1Int=3; //this triggers the petrinet model to start running

    public static boolean clickDirStart=false;
    public static boolean clickIntPedStart=false;
    public static boolean clickIntStart=false;

    public static boolean clickDirEmergency=false;
    public static boolean clickIntPedEmergency=false;
    public static boolean clickIntEmergency=false;

    public static boolean clickDirPedestrian=false;
    public static boolean clickIntPdPedestrianNS=false;
    public static boolean clickIntPdPedestrianSN=false;
    public static boolean clickIntPdPedestrianWE=false;
    public static boolean clickIntPdPedestrianEW=false;

    public static boolean dirRun=false;
    public static boolean intRun=false;
    public static boolean intPedRun=false;

}
```

Java code 4: roadFxnClass

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadPackage;

/**
 *
 * @author aby
 */
public class roadFxnClass {
```

```
public int getGlobalValOne()
{
   return roadGlobals.g1Int;
}

public String getGlobalDirectStart()
{
 return String.valueOf(roadGlobals.clickDirStart);
}

 public String getGlobalDirectRun()
{
 return String.valueOf(roadGlobals.dirRun);
}
 public String getGlobalDirectEmergency()
{
 return String.valueOf(roadGlobals.clickDirEmergency);
}

public String getGlobalDirectPedestrian()
{
 return String.valueOf(roadGlobals.clickDirPedestrian);
}

 public String getGlobalIntRun()
{
 return String.valueOf(roadGlobals.intRun);
}

 public String getGlobalIntersectionStart()
{
 return String.valueOf(roadGlobals.clickIntStart);
}

 public String getGlobalIntersectionEmergency()
{
 return String.valueOf(roadGlobals.clickIntEmergency);
}

 public String getGlobalIntersectionPdStart()
{
```

```
  return String.valueOf(roadGlobals.clickIntPedStart);
 }

 public String getGlobalIntersectionPdEmergency()
{
  return String.valueOf(roadGlobals.clickIntPedEmergency);
 }

public String getGlobalIntersectionPdRun()
{
  return String.valueOf(roadGlobals.intPedRun);
 }

 public String getGlobalIntersectionPdPedestrianNS()
{
  return String.valueOf(roadGlobals.clickIntPdPedestrianNS);
 }

  public String getGlobalIntersectionPdPedestrianSN()
{
  return String.valueOf(roadGlobals.clickIntPdPedestrianSN);
 }

  public String getGlobalIntersectionPdPedestrianWE()
{
  return String.valueOf(roadGlobals.clickIntPdPedestrianWE);
 }

  public String getGlobalIntersectionPdPedestrianEW()
{
  return String.valueOf(roadGlobals.clickIntPdPedestrianEW);
 }

//////Resets Pedestrian global value after cycle has completed
  public void ResetDirectPedestrian()
{
   roadGlobals.clickDirPedestrian=false;
}

  public void ResetIntersectionPedestrian(int value)
{
    switch(value)
```

```
   {
      case 1:roadGlobals.clickIntPdPedestrianNS=false;
         break;
      case 2:roadGlobals.clickIntPdPedestrianSN=false;
         break;
      case 3:roadGlobals.clickIntPdPedestrianWE=false;
         break;
      case 4:roadGlobals.clickIntPdPedestrianEW=false;
         break;
    }
  }
}
```

Java code 5: roadClassT

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadPackage;

import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;



public class roadClassT extends JPanel {
   private static final long serialVersionUID=12345678L;

   private Color red;
   private Color yellow;
   private Color green;
   private Color green1;
   private Color green2;
   private Color black;
   private int rotate;



   public roadClassT()
   {
   rotate=0;
   setSize(30,60);
```

```
}

 public roadClassT(int rotate)
{
if(rotate==90||rotate==270)
   {
   this.rotate=rotate;
   setSize(60,30);
   }
else if(rotate==180)
   {
    this.rotate=180;
   setSize(30,60);
   }
else
   {
    this.rotate=0;
   setSize(40,90);
   }
}

public void setBlack()
 {
 red=yellow=green=green1=green2=black;repaint();
 }

 public void setYellow()
 {
 red=Color.BLACK;yellow=Color.YELLOW;green=Color.BLACK;green1=Color.BLACK;green2=Color.BLACK;repaint();
 }

  public void setRed()
 {
 red=Color.RED;yellow=Color.BLACK;green=Color.BLACK;green1=Color.BLACK;green2=Color.BLACK;repaint();
 }

   public void setGreen()
 {
 red=Color.BLACK;yellow=Color.BLACK;green=Color.GREEN;green1=Color.BLACK;green2=Color.BLACK;repaint();
 }
    public void setGreen1()
 {
```

```
 red=Color.BLACK;yellow=Color.BLACK;green=Color.BLACK;green1=Color.GREEN;green2=Color.BLACK;repaint();
 }

  public void setGreen2()
 {
 red=Color.BLACK;yellow=Color.BLACK;green=Color.BLACK;green1=Color.BLACK;green2=Color.GREEN;repaint();
 }
   public void setRedandYellow()
 {
 red=Color.RED;yellow=Color.YELLOW;green=Color.BLACK;green1=Color.BLACK;green2=Color.BLACK;repaint();
 }

protected void paintComponent(Graphics gx)
{
super.paintComponent(gx);

if(rotate==90)//WEST LIGHT
{
   int h[ ] = {10, 15, 15, 18, 12, 6, 10, 10}; int v[ ] = {11, 11,16 ,16, 23, 16 ,16, 11};
   gx.setColor(green2);gx.fillPolygon(h, v, h.length);gx.drawOval(5,8,15,15);
   int h1[ ] = {23, 29, 35, 32, 32, 27, 27, 23}; int v1[ ] = {16, 9, 16, 16, 21, 21 ,16, 16};
   gx.setColor(green1);gx.fillPolygon(h1, v1, h1.length);gx.drawOval(21,8,15,15);
   int h2[ ] = {40,45,45,51,45,45,40,40};int v2[ ] = {12,12,8,14,21,18,18,13};
   gx.setColor(green);gx.fillPolygon(h2, v2, h2.length);gx.drawOval(37,8,15,15);
   gx.setColor(yellow);gx.fillOval(53,8,15,15);
   gx.setColor(red);gx.fillOval(69,8,15,15);
   gx.setColor(Color.BLACK);gx.drawRect(2, 3, 85, 25);//    gx.fillArc(38, 10, 20,30, 50, 80);
}
else  if(rotate==180) //SOUTH LIGHT  done
{
   int h2[ ] = {7,15,15,21,21,15,15,7}; int v2[ ] = {11,5,9,9,14,14,17,11};
   gx.setColor(green2);gx.fillPolygon(h2, v2, h2.length);gx.drawOval(7,4,15,15);
   int h1[ ] = {10,15,15,21,15,15,10,10};int v1[ ] = {27,27,23,29,35,32,32,26};
   gx.setColor(green1);gx.fillPolygon(h1, v1, h1.length);gx.drawOval(7,21,15,15);
   int h[ ] = {13, 18, 18, 21, 15, 9, 13, 13}; int v[ ] = {41, 41,46 ,46, 53, 46 ,46, 41};
   gx.setColor(green);gx.fillPolygon(h, v, h.length);gx.drawOval(7,38,15,15);
   gx.setColor(yellow);gx.fillOval(7,54,15,15);
   gx.setColor(red);gx.fillOval(7,70,15,15);
   gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 85);

}
 else  if(rotate==270) //EAST LIGHT
```

```
  {
    gx.setColor(red);gx.fillOval(4,7,15,15);
    gx.setColor(yellow);gx.fillOval(20,7,15,15);
    int h[ ] = {36,44,44,50,50,44,44,36}; int v[ ] = {14,7,12,12,17,17,20,14};
    gx.setColor(green);gx.fillPolygon(h, v, h.length);gx.drawOval(36,7,15,15);
    int h1[ ] = {58, 63, 63, 66, 60, 54, 58, 58}; int v1[ ] = {10, 10,15 ,15, 21, 15 ,15, 10};
    gx.setColor(green1);gx.fillPolygon(h1, v1, h1.length);gx.drawOval(52,7,15,15);
    int h2[ ] = {70, 76, 82, 79, 79, 74, 74, 70};int v2[ ] ={15, 8, 15, 15, 20, 20 ,15, 15} ;
    gx.setColor(green2);gx.fillPolygon(h2, v2, h2.length); gx.drawOval(68,7,15,15);
    gx.setColor(Color.BLACK);gx.drawRect(2, 2, 85, 25);
  }
  else //NORTH LIGHT  done
  {
    gx.setColor(red);gx.fillOval(7,4,15,15);
    gx.setColor(yellow);gx.fillOval(7,20,15,15);
    int h[ ] = {9, 15, 21, 18, 18, 13, 13, 9}; int v[ ] = {45, 38, 45, 45, 50, 50 ,45, 45};
    gx.setColor(green);gx.fillPolygon(h, v, h.length);gx.drawOval(7, 36, 15, 15);
    int h1[ ] = {7,15,15,21,21,15,15,7}; int v1[ ] = {62,55,59,59,64,64,67,61};
    gx.setColor(green1);gx.fillPolygon(h1, v1, h1.length);gx.drawOval(7, 53, 15, 15);
    int h2[ ] = {10,15,15,21,15,15,10,10};int v2[ ] = {76,76,72,78,84,81,81,76};
    gx.setColor(green2);gx.fillPolygon(h2, v2, h2.length);gx.drawOval(7, 70, 15, 15);
    gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 85);
  }
  }
```

Java code 6: roadClassD

```
package roadPackage;


import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;

public class roadClassD extends JPanel {
  private static final long serialVersionUID=12345678L;


  private Color red;
  private Color yellow;
```

```
private Color green;
private Color black;
private int rotate;

 public roadClassD(int rotate)
{
   this.rotate=rotate;
   setBlack();
}

 public void setBlack()
  {
  red=yellow=green=black;repaint();
 }

 public void setYellow()
 {
 red=Color.BLACK;yellow=Color.YELLOW;green=Color.BLACK;repaint();
 }

  public void setRed()
 {
 red=Color.RED;yellow=Color.BLACK;green=Color.BLACK;repaint();
 }

   public void setGreen()
 {
 red=Color.BLACK;yellow=Color.BLACK;green=Color.GREEN;repaint();
 }

  public void setRedandYellow()
 {
 red=Color.RED;yellow=Color.YELLOW;green=Color.BLACK;repaint();
 }

 public void setAllYellow()
 {
 red=Color.YELLOW;yellow=Color.YELLOW;green=Color.YELLOW;repaint();
 }

protected void paintComponent(Graphics gx)
{
```

```
    super.paintComponent(gx);

    if(rotate==901)
     {
       gx.setColor(red);gx.fillOval(6,7,15,15);
       gx.setColor(green);gx.fillOval(23,7,15,15);
       gx.setColor(Color.BLACK);gx.drawRect(2, 2, 40, 25);
     }
     else if(rotate==902)
    {
       gx.setColor(green);gx.fillOval(6,7,15,15);
       gx.setColor(red);gx.fillOval(23,7,15,15);
       gx.setColor(Color.BLACK);gx.drawRect(2, 2, 40, 25);
    }
    else if(rotate==180)
    {
       gx.setColor(red); gx.fillOval(7,5,15,15);//gx.fillArc(7, 5, 20,30, 50, 80);//
       gx.setColor(yellow);gx.fillOval(7,22,15,15);
       gx.setColor(green);gx.fillOval(7,39,15,15);
       gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 55);
    }
    else if(rotate==01)
    {
       gx.setColor(red);gx.fillOval(7,6,15,15);
       gx.setColor(green);gx.fillOval(7,23,15,15);
       gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 40);
    }
     else if(rotate==02)
    {
       gx.setColor(green);gx.fillOval(7,6,15,15);
       gx.setColor(red);gx.fillOval(7,23,15,15);
       gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 40);
    }
    else
    {}

    }
}
```

Java code 7: roadClass

```
/*
```

```
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadPackage;

import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;



public class roadClass extends JPanel {
   private static final long serialVersionUID=12345678L;

   private Color red;
   private Color yellow;
   private Color green;
   private Color black;
   private int rotate;



   public roadClass()
   {
   rotate=0;
   setSize(30,60);
//   setBlack();
   }

    public roadClass(int rotate)
   {
   if(rotate==90||rotate==270)
      {
      this.rotate=rotate;
      setSize(60,30);
      }
   else if(rotate==180)
      {
       this.rotate=180;
      setSize(30,60);
      }
   else
      {
       this.rotate=0;
```

```
    setSize(40,90);
    }
setBlack();
 }

 public void setBlack()
  {
  red=yellow=green=black;repaint();
 }

 public void setYellow()
 {
 red=Color.BLACK;yellow=Color.YELLOW;green=Color.BLACK;repaint();
 }

  public void setRed()
 {
 red=Color.RED;yellow=Color.BLACK;green=Color.BLACK;repaint();
 }

   public void setGreen()
 {
 red=Color.BLACK;yellow=Color.BLACK;green=Color.GREEN;repaint();
 }

  public void setRedandYellow()
 {
 red=Color.RED;yellow=Color.YELLOW;green=Color.BLACK;repaint();
 }

 protected void paintComponent(Graphics gx)
 {
 super.paintComponent(gx);

 if(rotate==90)
 {
    gx.setColor(green);gx.fillOval(5,8,15,15);
    gx.setColor(yellow);gx.fillOval(22,8,15,15);
    gx.setColor(red);gx.fillOval(38,8,15,15);
    gx.setColor(Color.BLACK);gx.drawRect(2, 3, 55, 25);//    gx.fillArc(38, 10, 20,30, 50, 80);
 }
 else  if(rotate==180)
```

```
   {
      gx.setColor(green); gx.fillOval(7,5,15,15);//gx.fillArc(7, 5, 20,30, 50, 80);//
      gx.setColor(yellow);gx.fillOval(7,22,15,15);
      gx.setColor(red);gx.fillOval(7,39,15,15);
      gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 55);
   }
    else  if(rotate==270)
   {
      gx.setColor(red);gx.fillOval(4,7,15,15);
      gx.setColor(yellow);gx.fillOval(22,7,15,15);
      gx.setColor(green);gx.fillOval(40,7,15,15);
      gx.setColor(Color.BLACK);gx.drawRect(2, 2, 55, 25);
   }
   else
   {
      gx.setColor(red);gx.fillOval(7,6,15,15);
      gx.setColor(yellow);gx.fillOval(7,23,15,15);
      gx.setColor(green);gx.fillOval(7,40,15,15);
      gx.setColor(Color.BLACK);gx.drawRect(2, 2, 25, 55);
   }

   }
}
```

Java code 8: controlPanel

```
package roadPackage;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JPanel;

public class controlPanel extends JPanel
{
    JPanel dirContPanel=new JPanel();JPanel intContPanel=new JPanel();JPanel pedContPanel=new JPanel();
    JPanel dirPedContPanelL=new JPanel(); JPanel dirPedContPanelR=new JPanel();

    JPanel intPedPanelNorthL=new JPanel(); JPanel intPedPanelNorthR=new JPanel();
    JPanel intPedPanelSouthL=new JPanel(); JPanel intPedPanelSouthR=new JPanel();
    JPanel intPedPanelWestL=new JPanel(); JPanel intPedPanelWestR=new JPanel();
```

```
JPanel intPedPanelEastL=new JPanel(); JPanel intPedPanelEastR=new JPanel();



JButton dirStartButton = new JButton("Start");
JButton dirEmgButton = new JButton("Emergency");
JButton dirRestartButton=new JButton("Restart");

ImageIcon icon=new ImageIcon("C:\\work\\Road\\src\\roadPackage\\Sign.gif");

JButton dirPedButtonL=new JButton();
JButton dirPedButtonR=new JButton();

JButton intPedNorthBtnL=new JButton();
JButton intPedNorthBtnR=new JButton();
JButton intPedSouthBtnL=new JButton();
JButton intPedSouthBtnR=new JButton();
JButton intPedWestBtnL=new JButton();
JButton intPedWestBtnR=new JButton();
JButton intPedEastBtnL=new JButton();
JButton intPedEastBtnR=new JButton();

JButton intStartButton = new JButton("Start");
JButton intEmgButton = new JButton("Emergency");

JButton pedStartButton = new JButton("Start");
JButton pedEmgButton = new JButton("Emergency");

public controlPanel()
{
}

public void addControlButtons(JPanel mainPanel)
{
dirContPanel.setLayout(null);
dirContPanel.setBounds(230, 30, 110, 60);// dirContPanel.setBounds(230, 3, 110, 80);
mainPanel.add(dirContPanel);
setStartButton(dirContPanel);    setEmergencyButton(dirContPanel);

pedContPanel.setLayout(null);
pedContPanel.setBounds(490, 30, 110, 60);//(490, 3, 110, 80)
mainPanel.add(pedContPanel);
setInterPedStartButton(pedContPanel,mainPanel);
```

```
setIntPedestrianEmergencyButton(pedContPanel);

intContPanel.setLayout(null);
intContPanel.setBounds(890, 30, 110, 60);//(890, 3, 110, 80)
mainPanel.add(intContPanel);
setIntersectionStartButton(intContPanel,mainPanel);
setIntersectionEmergencyButton(intContPanel);
}

 public void addPedestrianControlButtons(JPanel mainPanel)
{
   dirPedContPanelL.setLayout(null);
   dirPedContPanelL.setBounds(28, 452, 24, 24);
   mainPanel.add(dirPedContPanelL);
   setDirPedestrianButtonL(dirPedContPanelL);

   dirPedContPanelR.setLayout(null);
   dirPedContPanelR.setBounds(250, 452, 24, 24);
   mainPanel.add(dirPedContPanelR);
   setDirPedestrianButtonR(dirPedContPanelR);

   intPedPanelSouthL.setLayout(null);
   intPedPanelSouthL.setBounds(297, 422, 24, 24);
   mainPanel.add(intPedPanelSouthL);
   setIntPedestrianBtnSouthL(intPedPanelSouthL);

   intPedPanelSouthR.setLayout(null);
   intPedPanelSouthR.setBounds(548, 422, 24, 24);
   mainPanel.add(intPedPanelSouthR);
   setIntPedestrianBtnSouthR(intPedPanelSouthR);

   intPedPanelNorthL.setLayout(null);
   intPedPanelNorthL.setBounds(264, 162, 24, 24);
   mainPanel.add(intPedPanelNorthL);
   setIntPedestrianBtnNorthL(intPedPanelNorthL);

   intPedPanelNorthR.setLayout(null);
   intPedPanelNorthR.setBounds(516, 162, 24, 24);
   mainPanel.add(intPedPanelNorthR);
   setIntPedestrianBtnNorthR(intPedPanelNorthR);

   intPedPanelWestL.setLayout(null);
```

```
    intPedPanelWestL.setBounds(223, 166, 24, 24);
    mainPanel.add(intPedPanelWestL);
    setIntPedestrianBtnWestL(intPedPanelWestL);

    intPedPanelWestR.setLayout(null);
    intPedPanelWestR.setBounds(223, 391, 24, 24);
    mainPanel.add(intPedPanelWestR);
    setIntPedestrianBtnWestR(intPedPanelWestR);

    //
    intPedPanelEastL.setLayout(null);
    intPedPanelEastL.setBounds(597, 166, 24, 24);
    mainPanel.add(intPedPanelEastL);
    setIntPedestrianBtnEastL(intPedPanelEastL);

    intPedPanelEastR.setLayout(null);
    intPedPanelEastR.setBounds(597, 389, 24, 24);
    mainPanel.add(intPedPanelEastR);
    setIntPedestrianBtnEastR(intPedPanelEastR);

}


 public void setIntPedestrianBtnEastL(JPanel buttonPanel)
{
buttonPanel.add(intPedEastBtnL);
intPedEastBtnL.setBounds(2, 2, 20, 20);
intPedEastBtnL.setIcon(icon);
intPedEastBtnL.addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent event)
    {
        roadGlobals.clickIntPdPedestrianEW=true;
    }
});
}

 public void setIntPedestrianBtnEastR(JPanel buttonPanel)
{
buttonPanel.add(intPedEastBtnR);
intPedEastBtnR.setBounds(2, 2, 20, 20);
```

```
intPedEastBtnR.setIcon(icon);
intPedEastBtnR.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianEW=true;
   }
});
}

 public void setIntPedestrianBtnWestL(JPanel buttonPanel)
{
buttonPanel.add(intPedWestBtnL);
intPedWestBtnL.setBounds(2, 2, 20, 20);
intPedWestBtnL.setIcon(icon);
intPedWestBtnL.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianWE=true;
   }
});
}

 public void setIntPedestrianBtnWestR(JPanel buttonPanel)
{
buttonPanel.add(intPedWestBtnR);
intPedWestBtnR.setBounds(2, 2, 20, 20);
intPedWestBtnR.setIcon(icon);
intPedWestBtnR.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianWE=true;
   }
});
}

 public void setIntPedestrianBtnNorthL(JPanel buttonPanel)
```

```
{
buttonPanel.add(intPedNorthBtnL);
intPedNorthBtnL.setBounds(2, 2, 20, 20);
intPedNorthBtnL.setIcon(icon);
intPedNorthBtnL.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianNS=true;
   }
});
}

 public void setIntPedestrianBtnNorthR(JPanel buttonPanel)
{
buttonPanel.add(intPedNorthBtnR);
intPedNorthBtnR.setBounds(2, 2, 20, 20);
intPedNorthBtnR.setIcon(icon);
intPedNorthBtnR.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianNS=true;
   }
});
}


public void setIntPedestrianBtnSouthL(JPanel buttonPanel)
{
buttonPanel.add(intPedSouthBtnL);
intPedSouthBtnL.setBounds(2, 2, 20, 20);
intPedSouthBtnL.setIcon(icon);
intPedSouthBtnL.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianSN=true;
   }
```

```
});
}

 public void setIntPedestrianBtnSouthR(JPanel buttonPanel)
{
buttonPanel.add(intPedSouthBtnR);
intPedSouthBtnR.setBounds(2, 2, 20, 20);
intPedSouthBtnR.setIcon(icon);
intPedSouthBtnR.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickIntPdPedestrianSN=true;
   }
});
}



 public void setDirPedestrianButtonR(JPanel buttonPanel)
{
buttonPanel.add(dirPedButtonR);
dirPedButtonR.setBounds(2, 2, 20, 20);
dirPedButtonR.setIcon(icon);
dirPedButtonR.addActionListener(new ActionListener()
{
   @Override
   public void actionPerformed(ActionEvent event)
   {
      roadGlobals.clickDirPedestrian=true;
   }
});
}

  public void setDirPedestrianButtonL(JPanel buttonPanel)
{
buttonPanel.add(dirPedButtonL);
dirPedButtonL.setBounds(2, 2, 20, 20);
dirPedButtonL.setIcon(icon);
dirPedButtonL.addActionListener(new ActionListener()
{
```

```
    @Override
    public void actionPerformed(ActionEvent event)
    {
       roadGlobals.clickDirPedestrian=true;
    }
});
}

public void setStartButton(JPanel buttonPanel)
{
 buttonPanel.add(dirStartButton);
 dirStartButton.setBounds(5,10,100,20);
 dirStartButton.addActionListener(new ActionListener()
       {
          @Override
        public void actionPerformed(ActionEvent event)
        {
          roadGlobals.clickDirStart=true;
          roadGlobals.clickDirEmergency=false;
          roadGlobals.g1Int=1;
          roadGlobals.dirRun=true;
          dirStartButton.setEnabled(false);
          dirEmgButton.setEnabled(true);
         }
       });

}

 public void setInterPedStartButton(JPanel buttonPanel,JPanel mainPanel)
{
 buttonPanel.add(pedStartButton);
 pedStartButton.setBounds(5,10,100,20);//(5,30,100,20)
 pedStartButton.addActionListener(new ActionListener()
       {
          @Override
        public void actionPerformed(ActionEvent event)
        {
           roadGlobals.g1Int=1;
           roadGlobals.clickIntPedStart=true;
           roadGlobals.clickIntPedEmergency=false;
           roadGlobals.intPedRun=true;
           pedStartButton.setEnabled(false);
```

```
            pedEmgButton.setEnabled(true);
         }
        });
}

 public void setIntersectionStartButton(JPanel buttonPanel,JPanel mainPanel)
{
 buttonPanel.add(intStartButton);intStartButton.setBounds(5,10,100,20);
 intStartButton.addActionListener(new ActionListener()
        {
           @Override
         public void actionPerformed(ActionEvent event)
         {
            roadGlobals.g1Int=1;
            roadGlobals.clickIntStart=true;
            roadGlobals.clickIntEmergency=false;
            roadGlobals.intRun=true;
            intStartButton.setEnabled(false);
            intEmgButton.setEnabled(true);
         }
        });
}
  public void setEmergencyButton(JPanel buttonPanel)
{
 buttonPanel.add(dirEmgButton);dirEmgButton.setBounds(5,35,100,20);
 dirEmgButton.addActionListener(new ActionListener()
        {
           @Override
         public void actionPerformed(ActionEvent event)
         {
            roadGlobals.clickDirEmergency=true;
            roadGlobals.clickDirStart=false;
            dirStartButton.setEnabled(true);
            dirEmgButton.setEnabled(false);
         }
        });
}

public void setIntPedestrianEmergencyButton(JPanel buttonPanel)
{
 buttonPanel.add(pedEmgButton);pedEmgButton.setBounds(5,35,100,20);
 pedEmgButton.addActionListener(new ActionListener()
```

```
        {
           @Override
         public void actionPerformed(ActionEvent event)
         {
            roadGlobals.clickIntPedEmergency=true;
            roadGlobals.clickIntPedStart=false;
            pedStartButton.setEnabled(true);
            pedEmgButton.setEnabled(false);
         }
        });
  }

   public void setIntersectionEmergencyButton(JPanel buttonPanel)
  {
   buttonPanel.add(intEmgButton);intEmgButton.setBounds(5,35,100,20);
   intEmgButton.addActionListener(new ActionListener()
        {
           @Override
         public void actionPerformed(ActionEvent event)
         {
            roadGlobals.clickIntEmergency=true;
            roadGlobals.clickIntStart=false;
            intEmgButton.setEnabled(false);
            intStartButton.setEnabled(true);
         }
        });
  }
}
```