YADAVRAO TASGAONKAR COLLEGE OF ENGINEERING

University of Mumbai

DEPARTMENT OF COMPUTER ENGINEERING

A

MINI PROJECT REPORT

ON

Graphical Representation of Traffic signal

Submitted by:

Rohan Uday Khambekar

Under the guidance of

1. Mr. Mahesh Bagali

Lecturer
Dept of CE

2. Mr. Abhijit Patil

Asst. Professors
Dept of CE

University of Mumbai

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the mini project work entitled **"Traffic Simulation"** carried out by Mr. Rohan Uday Khambekar in partial fulfillment of the requirements of Computer Graphics and Visualization Laboratory prescribed by the University of Mumbai, Mumbai, for the IV Semester B.E (Computer Science & Engineering) Degree course during the year 2011- 2012.

Project Guides:

1. Mr. Mahesh Bagali

2. Mr. Abhijit Patil

Head of the Department

Mr. Prasad Tambekar

Examiners:

Internal:

External:

Table of Contents

ACKNOWLEDGEMENTS

We are glad that we could finish our mini project on Simulation of Traffic signals as a part of Computer graphics and visualization laboratory.

We would like to thank our HOD Prof. Prasad Tambekar (Dept. of CE) for allowing us to work on the project. We express our gratitude towards Mr. Abhijit Patil (Asst. Professor Dept. of CE) and Mr. Mahesh Bagali (Lecturer Dept. of CE) for guiding us throughout the completion of the project. They have also helped at every stage of the project, telling what is expect and taught us how to make it done.

We are grateful to the college for keeping labs open whenever required and providing us the Systems and Required softwares.

Our parents have supported and encouraged us all the way here we forget not to remember them. And last but not the least all the lab assistants for helping us in print outs, software installations etc. We thank them whole heartedly.

ABSTRACT

The simulation of traffic involves the modeling of a complex system that is open, shows emergent phenomena, and non-linear relationships. This project outlines the ideas involved in traffic simulation and reports on the results of a traffic simulator designed to model congestion amongst urban and suburban roads. Overall this project successfully shows a relationship between the amount of congestion present in a traffic network and the mean speed of the vehicles in the network, and the effects of various different intersection controllers on traffic flow. It is discovered that the best method for maintaining high mean speeds as traffic flows through intersections is to use a mixture of intersection controllers on connected roads to vary the flow of traffic on each road.

In this project I present the simulation of the traffic signal by making use of the built in functions in the header files including glut, studio etc. for fragmentation, Rasterization, polygon filling, and Animation.

For the interactive working of the program like stopping and directing the vehicles the mouse and keyboard functions are been used.

Chapter 1

INTRODUCTION

1.1 Overview of the project

Vehicle Traffic Simulations are designed to accurately model the flow of real world vehicle traffic in a graph-based network and model the development of congestion. They form a complex system that is open. Shows emergent phenomena, non-linear relationships and can contain feedback loops. An accurate and realistic traffic simulation could aide in finding solutions for the easing of traffic congestion that in turn could aide and improve traffic flows in real world.

Traffic lights, which may also be known as stoplights, traffic signals, signal lights are signaling devices positioned at road intersections, pedestrian crossings and other locations to control competing flows of traffic. Traffic lights alternate the right of way of road users by displaying lights of a standard color (red, yellow, green).

In the typical sequence of colored lights:

1. Illumination of the green light allows traffic to proceed in the direction denoted,

2. Illumination of the yellow light denoting, if safe to do so, prepare to stop short of the intersection,

3. Illumination of the red signal prohibits any traffic from proceeding.

1.2 Problem Statement

Problem was regarding the depth information of the signal pole, vehicle and road in 2D implementation it is been solved using order in associated display function calls.

Controlling the vehicles by lightening the red, yellow and green light is done with interactive functions including mouse and keyboard functions.

Chapter 2

REQUIREMENTS SPECIFICATION

2.1 Hardware Requirements:

- Model: Dell Inspiron N4010
- Processor: Intel® Core™ i3 CPU M 370 @ 2.40GHz
- Installed memory (RAM): 3.00GB (2.86 GB usable)
- System type: 64-bit capable
- Graphics: Intel® HD Graphics
- Total size of Hard disk: 300GB

2.2 Software Requirements:

- Operating System: Microsoft windows 7 ultimate
- Integrated Development Environment: Microsoft Visual Studio 2010
- Application Program Interface: OpenGL
- Libraries: GL (OpenGL Basic Functions)

  GLU (OpenGL Utility Library)

  GLUT (Open GL Utility Tool Kit)

Chapter 3

LITERATURE SURVEY

3.1 Existing model

Vehicle Traffic Simulations are designed to accurately model the flow of real world vehicle traffic in a graph-based network and model the development of congestion. They form a complex system that is open, shows emergent phenomena, non-linear relationships and can contain feedback loops. An accurate and realistic traffic simulation could aide in finding solutions for the easing of traffic congestion that in turn could aide and improve traffic flows in the real world.

There are two key components involved in the simulation of vehicle traffic. The first is the underlying road network consisting of roads, lanes and intersections each with their related individual and combined properties. The second is the individual agents acting within the network. Ultimately the agents within the system are bound by the rules of the local area of the network they are operating in, however they are also affected by the actions of the other agents around them.

The ultimate measure and goal of any accurate traffic simulation would be the development of a tool that could identify congestion problems during the design and development phase of new road construction and therefore allowing practical solutions to be found at this point. While it is not expected that this project will fully develop such a tool it is hoped that the ideas discussed herein could aide in the future development of such a model. All of this makes vehicle traffic simulation an attractive and challenging research area.

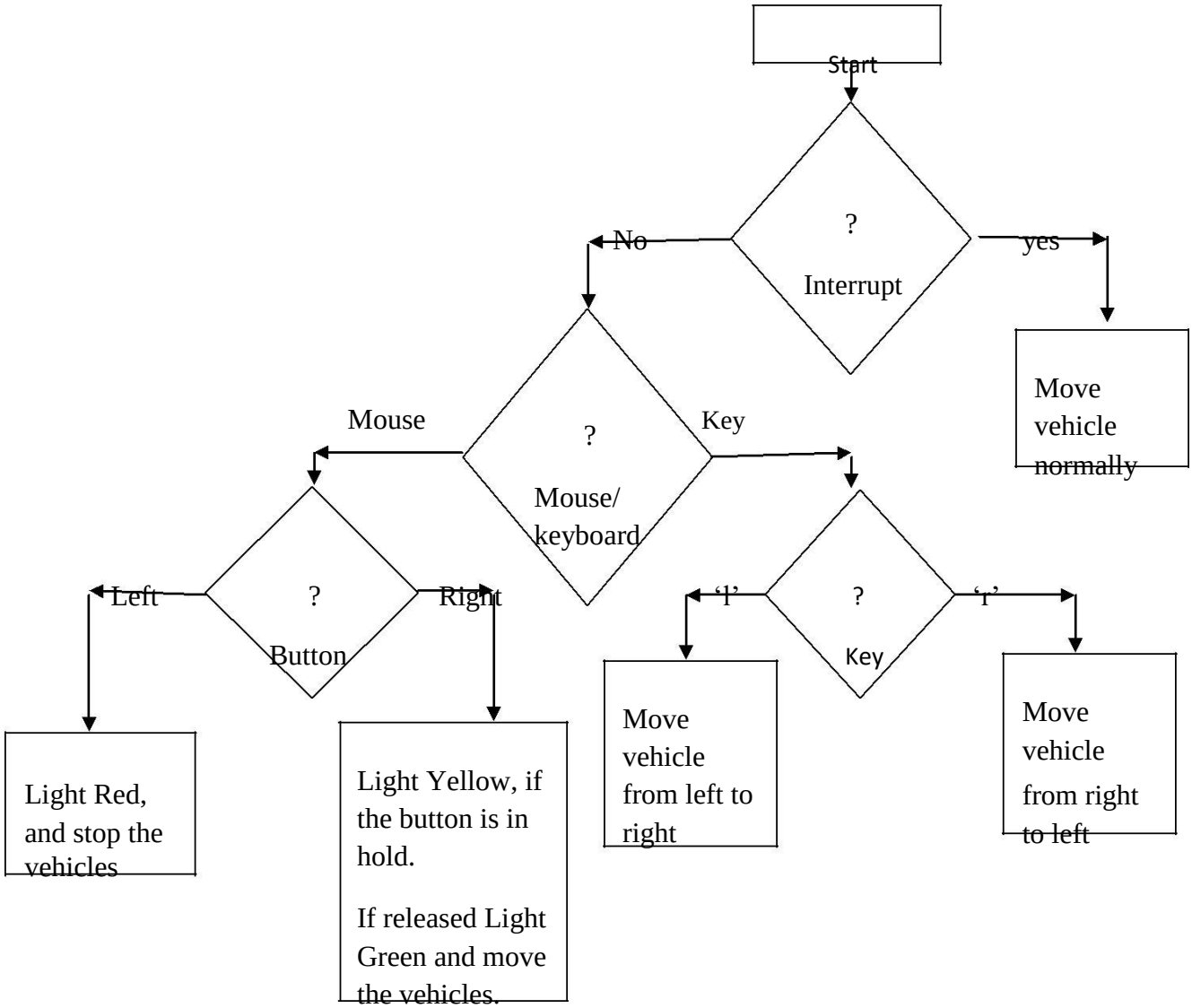
3.2 Proposed Model

Making a simulator containing three different scenarios, the first is a grid of intersections controlled by traffic lights which have a random duration. The cars owing through the intersection always continue moving in the same direction of travel so they never turn left or right into other roads. The cars have a small random variable to create differences in speed and bunch cars behind slower ones, however there is no passing.

The second and third scenarios involve the handling of ambulances, the first of these switches the traffic light to green on the road that the ambulance is travelling on when it arrives and the final scenario switches the light ahead of the ambulances arrival so it does not need to slow down. In both situations cars also allow the ambulance to pass.

Chapter 4

SYSTEM DESGIN



Chapter 5

IMPLEMENTATION

5.1 Main function

This main function is capable of handling the arguments given in the argument list at the command prompt as we have used variable ‘argc’ for total number of arguments and ‘argv’ for the array of argument list.

Main function initializes the Display Mode, Window Size and position. Then it invokes the display function within glutDisplayFunc() as the call back function.

```
void main(int argc, char* argv[])
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
        glutInitWindowSize(1346,728);
        glutInitWindowPosition(0,0);
        glutCreateWindow("Traffic signal");
        glutDisplayFunc(mydisplay);

 /*call back functions*/
        glutKeyboardFunc(myKeyboard);
```

```
        glutMouseFunc(myMouse);
        myinit();
        glutMainLoop();
}
```

5.2 Keyboard function

This Interactive function is invoked at the main program within glutKeyboardFunc() and it is repeatedly called during the execution of the program and hence handles the keyboard interrupts from the users.

When a key is been pressed the ASCII code of the key and the position of the screen pointer at the time of interruption will be sent to the function.

```
void myKeyboard( unsigned char key, int x, int y )
{
switch(key)
{
    case 13:
            if(flag==1)
            {
                flag=2;
                mydisplay();
            }
            if(flag==0) //Ascii of 'enter' key is 13
```

```
                {
                        flag=1;
                        mydisplay();
                } break;
        case 'l': control_keyl=key;
          p=0;q=0;r=1;break; case
             'r': control_keyr=key;
               p=0;q=0;r=1;break;
        case 's': mydisplay();break;
        case 'h': flag=1;mydisplay();
        break; default: break;
    }
}
```

5.3 Mouse function

This Interactive function is invoked at the main program within glutMoseFunc() and it is repeatedly called during the execution of the program and hence handles the Mouse interrupts from the users.

When a button is been pressed the identity of the button and the state of the button and the position of the screen pointer at the time of interruption will be sent to the mouse function.

```
void myMouse(int button,int state,int x,int y)

    {
        if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
        {
        traffic_regulator=0;
        p=1;q=0;r=0;}
       if(button==GLUT_RIGHT_BUTTON && state==GLUT_DOWN)
        {
        traffic_regulator=0;
        p=0;q=1;r=0;
        }
        if(button==GLUT_RIGHT_BUTTON && state==GLUT_UP)
        {
        traffic_regulator=1;
        p=0;q=0;r=1;
        }
        glutPostRedisplay();
}
```

5.4 Displaying text on the screen

The setFont() function is used to set the type of the font we are using. The drawstring() function takes the position of the string to be displayed on the screen in X Y Z coordinates and the string to display.

```
void *currentfont;
void setFont(void *font)
{
        currentfont=font;
}

void drawstring(float x,float y,float z,char *string)
{
        char *c;
        glRasterPos3f(x,y,z);

        for(c=string;*c!='\0';c++)
        { glColor3f(0.0,0.0,0.0);
                glutBitmapCharacter(currentfont,*c);
        }
}

void frontscreen(void)
```

```
{
setFont(GLUT_BITMAP_TIMES_ROMAN_24);
glClearColor(0.15,0.1,0.01,0);/*background for cover
page*/ glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1,0,0);
drawstring(450.0,700.0,0.0,"BEARYS INSTITUTE OF TECHNOLOGY
"); glFLush();
}
```

5.5 Display functions

We have used three display functions in this program the first one that is the myDisplay function is called in the main program using glutDisplayFunc() call back function. The mydisplay function controls the displaying of the front screen, help screen or the display function.

The second display function that is display() calls the objects like road(),car() etc. According to the order that we have written hence it indirectly handles the depth information.

```
void mydisplay(void)
{
glClear(GL_COLOR_BUFFER_BIT);
if(flag==0)
frontscreen
(); if(flag==1)
        helpscreen();
```

```
if(flag==2)
display();
glutSwapBuffers();
}
glutPostRedisplay();
}

void display(void)
{
if(traffic_regulator)
glutTimerFunc(50,update,0);
glClear(GL_COLOR_BUFFER_BIT);
glClearColor(red,green,blue,0);/*back ground for sky*/
road();
bus();
signal();
car();
car2();
glFlush();
}
```

5.6. Update functions

Update function is used for the transformation of the objects this function is invoked in display function inside glutTimerFunc(50,update,0). This function calls the update every 50 mille seconds hence the variable ‘a’, ’b’ are all modified then when we translate the object with this points we will see the animation effect.

```
void update(int value)
{
a=a-6;
b=b+6;
control();
/*making day to night*/
if(blue!=0&&green!=0)
{blue-=.004;green-=.004;
}
```

Chapter 6

TESTING

6.1 Test Results

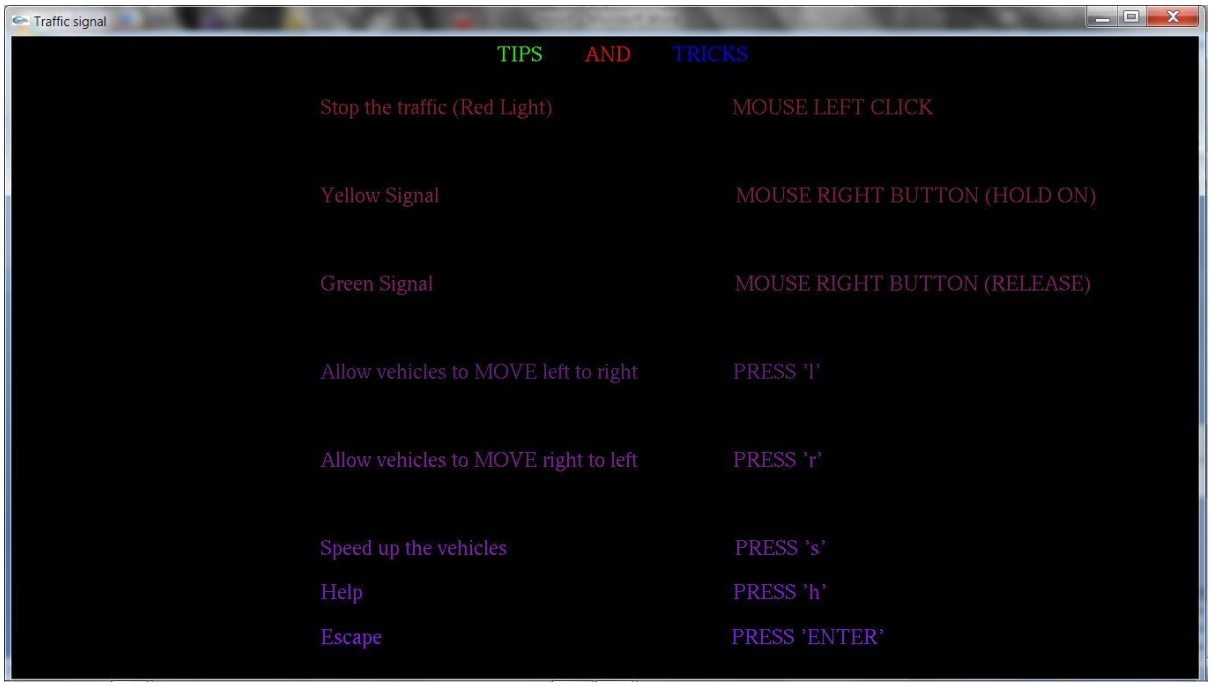
case	State	Expected	Obtained	Results
Case 1	Screen-1	Show the project details in specified font and location on the screen	Same as expected	Pass
Case-2	Screen-1	Move to screen2 that is help screen when Enter key is pressed.	Same as expected	Pass
Case-3	Screen-2	Move to screen3 that is traffic signal when Enter key is pressed.	Same as expected	Pass
Case-4	Screen-3	Move to screen2 when the key 'h' is pressed.	Same as expected	Pass
Case-5	Screen-3	Stop the flow of the traffic and light red when left button of the mouse is pressed	Same as expected	Pass
Case-6	Screen-3	Off the red light and light the yellow when right button of the mouse is on hold.	Same as expected	Pass
Case-7	Screen-3	Off the yellow light and light the green; simultaneously allow the traffic to flow when right button of the mouse is released.	Same as expected	Pass
Case-8	Screen-3	Allow only right side vehicles to flow when the key 'r' is pressed.	Same as expected	Pass
Case-9	Screen-3	Allow only left side vehicles to flow when the key 'l' is pressed.	Same as expected	Pass
Case-10	Screen-3	Increase the speed of the vehicles successively every time the key 's' is pressed.	Same as expected	Pass

Chapter 7

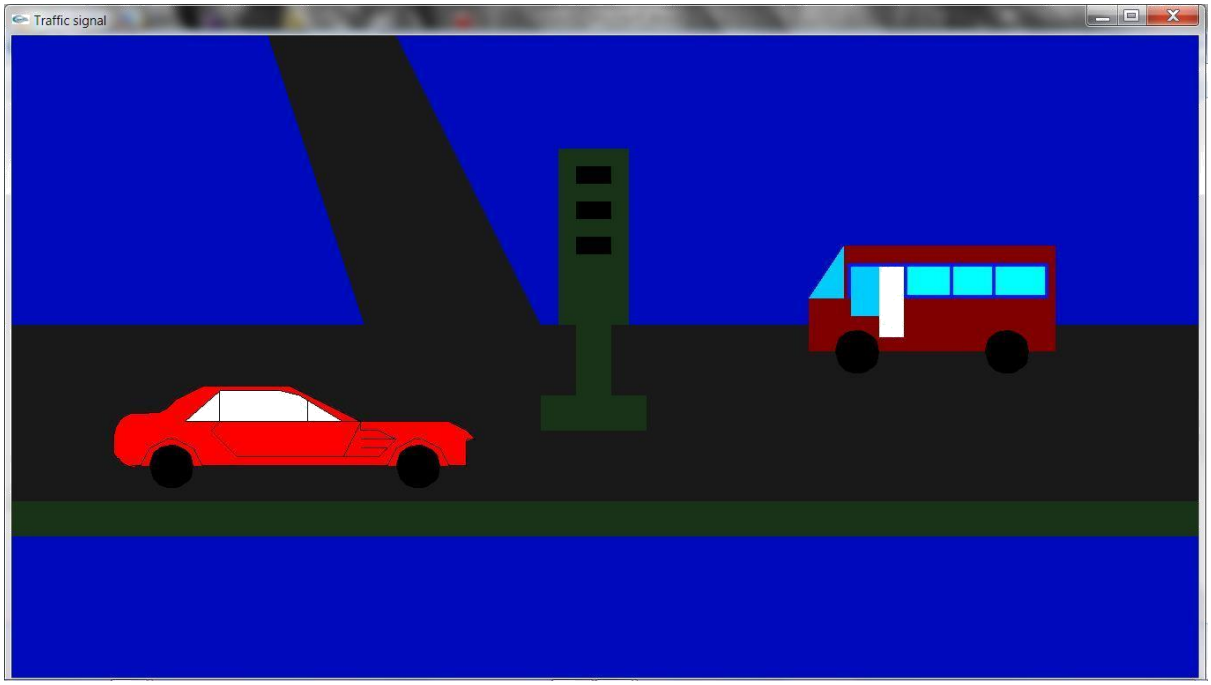
RESULT

6.1 Front screen showing project details

6.2 Second screen giving the user instructions

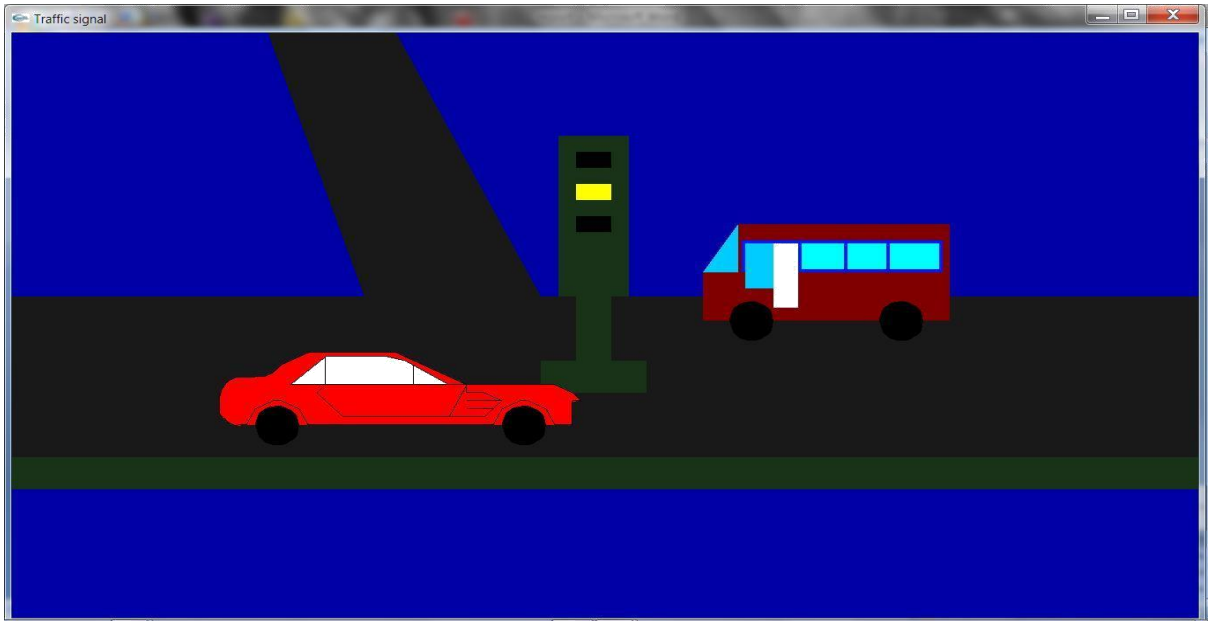


7.3 Third Animated screen showing the road traffic; where the traffic signal is located at the circle or the junction. And vehicles are moving in opposite direction.

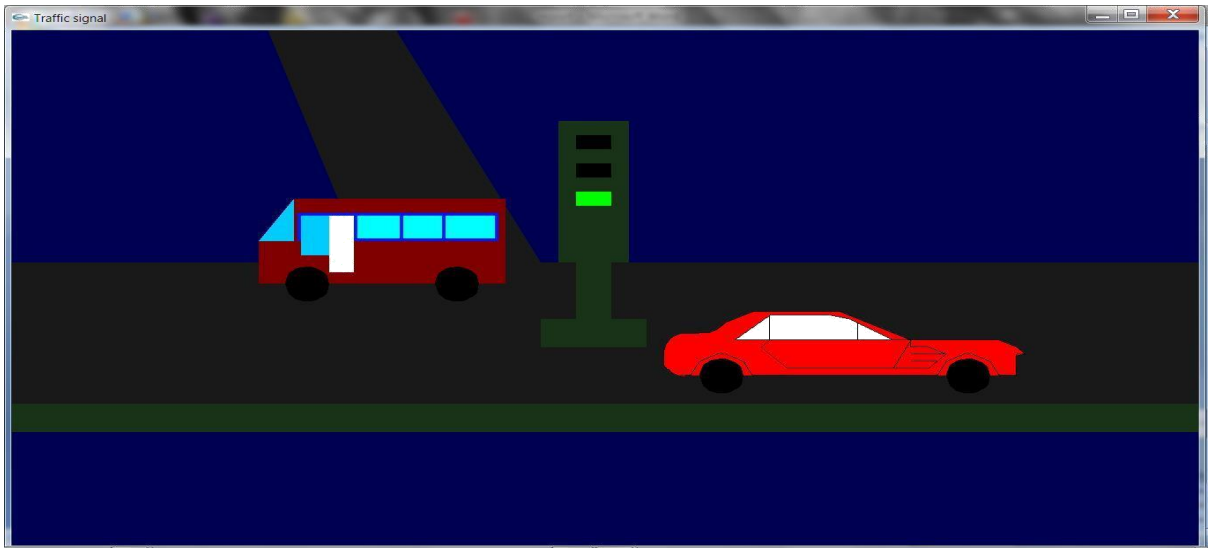


7.4 Mouse left button is pressed hence the traffic is under control.

7.5 Mouse right button is pressed and kept in hold; hence the yellow is lighting to make the drivers prepared for the move.

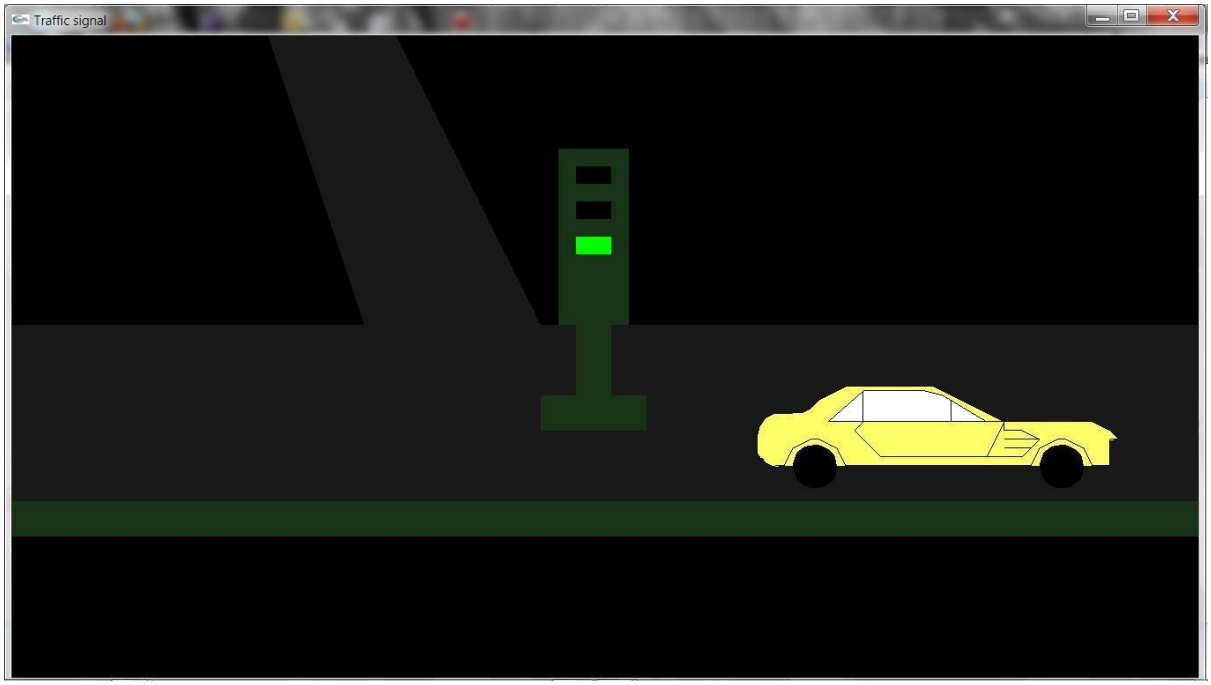


7.6 The character ‘r’ is pressed to make the right side vehicle to move and the Mouse button is released. Now the character ‘l’ is pressed so that the vehicle waiting at the left side of the signal will also move.



7.7 Snap shot showing the vehicle moving with high speed in the traffic free road.

The speed of this vehicle is made higher by pressing the character‘s’.



Chapter 8

CONCLUSION

This article presents the modeling, analysis and implementation of an urban traffic lights system using Timed CP-nets models. Especially, this paper also proposed the module of basic traffic light system model which can assist in designing the extended models. The advantage of the proposed approach is the clear presentation of the system behavior and readiness for implementation. To summarize, this paper has the following contributions.

BIBILIOGRAPHY

[1] Interactive Computer Graphics A Top-Down Approach with OpenGL -Edward Angel, 5th Edition, Addison-Wesley, 2008.

[2] Computer Graphics Using OpenGL – F.S. Hill,Jr. 2nd Edition, Pearson Education, 2001.

[3] Computer Graphics – James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Addison-Wesley 1997.

[4] Computer Graphics - OpenGL Version – Donald Hearn and Pauline Baker, 2nd Edition, Pearson Education, 2003.