

DAYANANDA SAGAR UNIVERSITY



A MINI PROJECT REPORT

ON

“TRAFFIC SIMULATION”

SUBMITTED TO THE 6th SEMESTER

COMPUTER GRAPHICS

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted

by

| | |
|----------------------|---------------|
| GURUDEV H Y | (ENG17CS0079) |
| HARIHARAN B | (ENG17CS0083) |
| KUSHAL N | (ENG17CS0115) |
| MOHAMMED ZAHID PASHA | (ENG17CS0129) |

VI Semester, 2020

Under the supervision of

Prof. Ranjitha G, Assistant Professor

1. ABSTRACT:

As we know, the traffic signals direct the flow of traffic with the exemption of signals with turning arrows which should be compulsory in accord with each other. Traffic signals help us to drive our vehicles in a safer manner lowering the risk of accidents, if properly followed.

Same method we implemented in our project which was used in real life .We have used this traffic signal in a cross road to avoid mishappening. When there is a green signal it means the vehicle can move or run on the road but if this green signal turns off and the red signal is turned on then it indicates the driver to stop his vehicle and at the same time signal allows some other side vehicle to cross the road. But if the vehicle is in the middle of a cross or vehicle has crossed the signal then there is no meaning of red light for the driver. When we press left mouse click then red light will glow and all vehicles are stopped and when we press and hold right mouse click then yellow light will glow and all vehicles ready to go .When we release the right mouse click then green light will glow and then the vehicle starts moving.

2. PROBLEM STATEMENT

To develop a motion of a car in response to traffic lights. We have three lights in the signal Red, Green and Yellow. Each light has some meaning . Red means stop , Green means go, Yellow means ready to go.

3. INTRODUCTION

- Traffic flow phenomenon consists of a wide range of complex activities, embracing vehicle arrivals, speed of travel, lane discipline, overtaking and crossing logic, gap acceptance, acceleration and deceleration etc..
- Different road characteristics will be govern by vehicle performance, characteristics and behaviour of road user
- The understanding of traffic flow can be obtained by observing how the vehicles move into the traffic stream by mathematical models, or analytical methods
- The science of this study has been developed with this approach.
 - The analytical approach requires that field data be collected on a large scale. i.e if a proper understanding of a complex heterogeneous traffic on Indian Road is to be achieved, real life observations need to be collected
- Time consuming and expensive

In this Project we are considering the basic Traffic simulation rules to control the vehicles passing through the road by using the clicks on the mouse.

4. LITERATURE SURVEY

Vehicle Traffic Simulation by Norman Badler badler@seas.upenn.edu Univ. of Pennsylvania Philadelphia, PA CIS 400/401,2010

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.639.1511&rep=rep1&type=pdf>

Researching on Simulation of Traffic Scene by OpenGL by Shuang Zhang, Huan Liu, Shixiong Zhang, 2010

<https://ieeexplore.ieee.org/document/5577929>

5. METHODOLOGY

5.1 ALGORITHM:

- Line drawing algorithm for traffic light pole and body of the car.
- Circle drawing algorithm for wheels of the car and lights of the traffic pole.
- Flood fill algorithm for colouring.

5.2 LANGUAGE:

C++ using OpenGL API for graphics

5.3 TOOLS USED: CodeBlocks

- It can create native Win32 executables, either console or GUI, as well as DLLs and static libraries.
- The application is platform independent.
- C++ was chosen to code the project as it is the most commonly used language in conjunction with OpenGL.

6. CODE:

```
#include<windows.h>
#include<stdio.h>
#include<GL/glut.h>

void bus();
void road();
void signal();
void car();
void car2();
void mydisplay();
void display();
void frontsreen();
void drawstring();
void setFont();
void myMouse();
void update();
void control();
void helpscreen();
GLint a=300,b=-300,flag=0,traffic_regulator=1,control_keyl,control_keyr;
GLfloat red=0,blue=1,green=.3;

GLfloat p=0,q=0,r=0;
//-----
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 helpscreen()

```

```

{
    setFont(GLUT_BITMAP_TIMES_ROMAN_24);
    glClearColor(0,0,0,0);/*background for cover page*/
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0,1,0);
    drawstring(550.0,700.0,0.0,"TIPS");
    glColor3f(1,0,0);
    drawstring(650.0,700.0,0.0,"AND");
    glColor3f(0,0,1);
    drawstring(750.0,700.0,0.0,"TRICKS");
    glColor3f(0.5,0.1,0.2);
    drawstring(350.0,640.0,0.0,"Stop the traffic (Red Light
    CLICK");
    glColor3f(0.5,0.1,0.3);
    drawstring(350.0,540.0,0.0,"Yellow Signal
    BUTTON (HOLD ON)");
    glColor3f(0.5,0.1,0.4);
    drawstring(350.0,440.0,0.0,"Green Signal
    BUTTON (RELEASE)");
    glColor3f(0.4,0.1,0.5);
    drawstring(350.0,340.0,0.0,"Allow vehicles to MOVE left to right
    PRESS 'L'");
    glColor3f(0.5,0.1,0.6);
    drawstring(350.0,240.0,0.0,"Allow vehicles to MOVE right to left
    PRESS 'R'");
    glColor3f(0.5,0.1,0.7);
    drawstring(350.0,140.0,0.0,"Speed up the vehicles
    PRESS 'S'");
    glColor3f(0.5,0.1,0.8);
    drawstring(350.0,90.0,0.0,"Help
    PRESS 'H'");
    glColor3f(0.5,0.1,0.9);
    drawstring(350.0,40.0,0.0,"Escape
    PRESS 'ENTER'");
    glFlush();
}

```

```

void control()
{
    if(control_keyl!='l'||control_keyr!='r')
    {
        if(control_keyl=='l')
        a=a+6;
        if(control_keyr=='r')
        b=b-6;
    }
}

```

```

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;
}

}

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();

}

void mydisplay(void)
{

```

```

glClear(GL_COLOR_BUFFER_BIT);
if(flag==0)
    frontscreen ();
if(flag==1)
    helpscreen();
if(flag==2)
    display();
glutSwapBuffers();
}

```

```

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

```

```

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();
}

```

```

void road()

```

```

{
glPushMatrix();
glScaled(40.0,40.0,0.0);
glColor3f(0.1,0.1,0.1);
glBegin(GL_POLYGON);
//straight road
glVertex2f(0,5);
glVertex2f(40,5);
glVertex2f(40,10);
glVertex2f(0,10);
glEnd();
//green edge
glBegin(GL_POLYGON);
glColor3f(0.1,0.2,0.1);
glVertex2f(0,5);
glVertex2f(40,5);
glVertex2f(40,4);
glVertex2f(0,4);
glEnd();
//cross road
glColor3f(0.1,0.1,0.1);
glBegin(GL_POLYGON);
glVertex2f(10,10);
glVertex2f(15,10);
glVertex2f(0,40);
glVertex2f(4,40);
glEnd();
glPopMatrix();
}

```

```

void signal()
{
glPushMatrix();
glScaled(40.0,40.0,0.0);
//stand
glColor3f(0.1,0.2,0.1);
glBegin(GL_POLYGON);
glVertex2f(15,7);
glVertex2f(15,8);

```

```
glVertex2f(18,8);
glVertex2f(18,7);
glEnd();
//pole
glBegin(GL_POLYGON);
glVertex2f(16,7);
glVertex2f(17,8);
glVertex2f(17,15);
glVertex2f(16,15);
glEnd();
//board
glBegin(GL_POLYGON);
glVertex2f(15.5,15);
glVertex2f(17.5,15);
glVertex2f(17.5,10);
glVertex2f(15.5,10);
glEnd();
//red
glColor3f(p,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(16,14.5);
glVertex2f(17,14.5);
glVertex2f(17,14);
glVertex2f(16,14);
glEnd();
//yellow
glColor3f(q,q,0.0);
glBegin(GL_POLYGON);
glVertex2f(16,13.5);
glVertex2f(17,13.5);
glVertex2f(17,13);
glVertex2f(16,13);
glEnd();
//green
glColor3f(0.0,r,0.0);
glBegin(GL_POLYGON);
glVertex2f(16,12.5);
glVertex2f(17,12.5);
glVertex2f(17,12);
```

```
glVertex2f(16,12);  
glEnd();  
glPopMatrix();  
}
```

```
void bus()  
{  
    glPushMatrix();  
    glTranslated(a,50.0,0.0);  
    glScaled(40.0,40.0,0.0);  
    glColor3f(0.5,0.0,0.0);  
    //bus out line  
    glBegin(GL_POLYGON);  
    glVertex2f(25,8);  
    glVertex2f(25,9.5);  
    glVertex2f(26,11);  
    glVertex2f(32,11);  
    glVertex2f(32,8);  
    glEnd();  
    //window frame  
    glColor3f(0,0.1,1);  
    glBegin(GL_POLYGON);  
    glVertex2f(26.1,9.5);  
    glVertex2f(26.1,10.5);  
    glVertex2f(31.8,10.5);  
    glVertex2f(31.8,9.5);  
    glEnd();  
    //Doors  
    glColor3f(0,0.8,1);  
    glBegin(GL_POLYGON);  
    glVertex2f(26.2,9);  
    glVertex2f(26.2,10.4);  
    glVertex2f(27.7,10.4);  
    glVertex2f(27.7,9);  
    glEnd();
```

```
glColor3f(1,1,1);  
glBegin(GL_POLYGON);
```

```
glVertex2f(27,8.4);
glVertex2f(27,10.4);
glVertex2f(27.7,10.4);
glVertex2f(27.7,8.4);
glEnd();
//small windows
glColor3f(0,1,1);
glBegin(GL_POLYGON);
glVertex2f(27.8,9.6);
glVertex2f(27.8,10.4);
glVertex2f(29,10.4);
glVertex2f(29,9.6);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(29.1,9.6);
glVertex2f(29.1,10.4);
glVertex2f(30.2,10.4);
glVertex2f(30.2,9.6);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(30.3,9.6);
glVertex2f(30.3,10.4);
glVertex2f(31.7,10.4);
glVertex2f(31.7,9.6);
glEnd();

//driver window
glColor3f(0,0.8,1);
glBegin(GL_POLYGON);
glVertex2f(25,9.5);
glVertex2f(26,11);
glVertex2f(26,9.5);
glEnd();
glPopMatrix();
//tyre
glPushMatrix();//front tyre
glTranslated(a+970,320,0.0);
glScaled(20.0,20.0,0.0);
```

```
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(3.0,2.5);
glVertex2f(3.0,2.6);
glVertex2f(3.15,3.1);
glVertex2f(3.2,3.2);
glVertex2f(3.3,3.35);
glVertex2f(3.4,3.4);
glVertex2f(3.5,3.45);
glVertex2f(3.6,3.55);
glVertex2f(3.7,3.6);
glVertex2f(3.8,3.63);
glVertex2f(4.0,3.65);
glVertex2f(4.2,3.7);
glVertex2f(4.4,3.7);
glVertex2f(4.6,3.65);
glVertex2f(4.8,3.55);
glVertex2f(5.0,3.45);
glVertex2f(5.1,3.4);
glVertex2f(5.2,3.25);
glVertex2f(5.3,3.2);
glVertex2f(5.4,3.0);
glVertex2f(5.5,2.5);
```

```
glVertex2f(5.45,2.15);
glVertex2f(5.4,1.9);
glVertex2f(5.35,1.8);
glVertex2f(5.2,1.6);
glVertex2f(5.0,1.5);
glVertex2f(4.9,1.4);
glVertex2f(4.7,1.3);
glVertex2f(4.6,1.27);
glVertex2f(4.4,1.25);
glVertex2f(4.0,1.25);
glVertex2f(3.9,1.3);
glVertex2f(3.75,1.35);
glVertex2f(3.6,1.4);
glVertex2f(3.45,1.55);
glVertex2f(3.3,1.7);
```

```
glVertex2f(3.2,1.8);  
glVertex2f(3.1,2.2);  
glEnd();  
glPopMatrix();
```

```
glPushMatrix();//back tyre  
glTranslated(a+1140,320,0.0);  
glScaled(20.0,20.0,0.0);  
glColor3f(0.0,0.0,0.0);  
glBegin(GL_POLYGON);  
glVertex2f(3.0,2.5);  
glVertex2f(3.0,2.6);  
glVertex2f(3.15,3.1);  
glVertex2f(3.2,3.2);  
glVertex2f(3.3,3.35);  
glVertex2f(3.4,3.4);  
glVertex2f(3.5,3.45);  
glVertex2f(3.6,3.55);  
glVertex2f(3.7,3.6);  
glVertex2f(3.8,3.63);  
glVertex2f(4.0,3.65);  
glVertex2f(4.2,3.7);  
glVertex2f(4.4,3.7);  
glVertex2f(4.6,3.65);  
glVertex2f(4.8,3.55);  
glVertex2f(5.0,3.45);  
glVertex2f(5.1,3.4);  
glVertex2f(5.2,3.25);  
glVertex2f(5.3,3.2);  
glVertex2f(5.4,3.0);  
glVertex2f(5.5,2.5);
```

```
glVertex2f(5.45,2.15);  
glVertex2f(5.4,1.9);  
glVertex2f(5.35,1.8);  
glVertex2f(5.2,1.6);  
glVertex2f(5.0,1.5);  
glVertex2f(4.9,1.4);  
glVertex2f(4.7,1.3);
```



```
glVertex2f(4.6,1.27);
glVertex2f(4.4,1.25);
glVertex2f(4.0,1.25);
glVertex2f(3.9,1.3);
glVertex2f(3.75,1.35);
glVertex2f(3.6,1.4);
glVertex2f(3.45,1.55);
glVertex2f(3.3,1.7);
glVertex2f(3.2,1.8);
glVertex2f(3.1,2.2);
glEnd();
glPopMatrix();
}
```

```
void car()
{
glPushMatrix(); //making color for outer line
glTranslated(b,190.0,0.0);
glScaled(20.0,20.0,0.0);
glColor3f(1.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(2.5,2.5);
glVertex2f(3.0,3.5);
glVertex2f(3.5,3.75);
glVertex2f(4.0,4.0);
glVertex2f(4.5,4.0);
glVertex2f(5.0,3.75);
glVertex2f(5.5,3.5);
glVertex2f(5.75,3.0);
glVertex2f(6.0,2.5);
glVertex2f(16.5,2.5);
glVertex2f(16.75,3.0);
glVertex2f(17.0,3.5);
glVertex2f(17.5,3.75);
glVertex2f(18.0,4.0);
glVertex2f(18.5,4.0);
glVertex2f(19.0,3.75);
glVertex2f(19.5,3.5);
glVertex2f(19.75,3.0);
```

```
glVertex2f(20.0,2.5);
glVertex2f(21.0,2.5);
glVertex2f(21.0,4.0);
glVertex2f(21.5,4.0);
glVertex2f(21.0,4.5);
glVertex2f(20.0,5.0);
glVertex2f(15.0,5.0);
glVertex2f(14.0,5.5);
glVertex2f(13.0,6.0);
glVertex2f(12.0,6.5);
glVertex2f(11.0,7.0);
glVertex2f(6.0,7.0);
glVertex2f(5.0,6.5);
glVertex2f(4.5,6.25);
glVertex2f(4.25,6.0);
glVertex2f(4.0,5.75);
glVertex2f(3.5,5.5);
glVertex2f(3.0,5.5);
glVertex2f(1.9,5.45);
glVertex2f(1.8,5.4);
glVertex2f(1.7,5.35);
glVertex2f(1.6,5.3);
glVertex2f(1.5,5.25);
glVertex2f(1.4,5.15);
glVertex2f(1.3,5.0);
glVertex2f(1.2,4.85);
glVertex2f(1.1,4.7);
glVertex2f(1.0,4.3);
glVertex2f(1.0,3.2);
glVertex2f(1.1,3.05);
glVertex2f(1.2,2.9);
glVertex2f(1.3,2.9);
glVertex2f(1.4,2.75);
glVertex2f(1.5,2.65);
glVertex2f(1.6,2.6);
glVertex2f(1.7,2.55);
glVertex2f(1.8,2.5);
glVertex2f(1.9,2.45);
glVertex2f(2.0,2.5);
```

```
glEnd();
```

```
glColor3f(1.0,1.0,1.0); //color for outer window  
glBegin(GL_POLYGON);  
glVertex2f(5.0,5.0);  
glVertex2f(14.0,5.0);  
glVertex2f(11.5,6.5);  
glVertex2f(10.5,6.75);  
glVertex2f(7.0,6.75);  
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //making outer line for car  
glBegin(GL_LINE_LOOP);  
glVertex2f(2.5,2.5);  
glVertex2f(3.0,3.5);  
glVertex2f(3.5,3.75);  
glVertex2f(4.0,4.0);  
glVertex2f(4.5,4.0);  
glVertex2f(5.0,3.75);  
glVertex2f(5.5,3.5);  
glVertex2f(5.75,3.0);  
glVertex2f(6.0,2.5);  
glVertex2f(16.5,2.5);  
glVertex2f(16.75,3.0);  
glVertex2f(17.0,3.5);  
glVertex2f(17.5,3.75);  
glVertex2f(18.0,4.0);  
glVertex2f(18.5,4.0);  
glVertex2f(19.0,3.75);  
glVertex2f(19.5,3.5);  
glVertex2f(19.75,3.0);  
glVertex2f(20.0,2.5);  
glVertex2f(21.0,2.5);  
glVertex2f(21.0,4.0);  
glVertex2f(21.5,4.0);  
glVertex2f(21.0,4.5);  
glVertex2f(20.0,5.0);  
glVertex2f(15.0,5.0);  
glVertex2f(14.0,5.5);
```

```
glVertex2f(13.0,6.0);
glVertex2f(12.0,6.5);
glVertex2f(11.0,7.0);
glVertex2f(6.0,7.0);
glVertex2f(5.0,6.5);
glVertex2f(4.5,6.25);
glVertex2f(4.25,6.0);
glVertex2f(4.0,5.75);
glVertex2f(3.5,5.5);
glVertex2f(3.0,5.5);
glVertex2f(1.9,5.45);
glVertex2f(1.8,5.4);
glVertex2f(1.7,5.35);
glVertex2f(1.6,5.3);
glVertex2f(1.5,5.25);
glVertex2f(1.4,5.15);
glVertex2f(1.3,5.0);
glVertex2f(1.2,4.85);
glVertex2f(1.1,4.7);
glVertex2f(1.0,4.3);
glVertex2f(1.0,3.2);
glVertex2f(1.1,3.05);
glVertex2f(1.2,2.9);
glVertex2f(1.3,2.9);
glVertex2f(1.4,2.75);
glVertex2f(1.5,2.65);
glVertex2f(1.6,2.6);
glVertex2f(1.7,2.55);
glVertex2f(1.8,2.5);
glVertex2f(1.9,2.45);
glVertex2f(2.0,2.5);
glEnd();
```

```
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINE_LOOP); //outer line for design a car
glVertex2f(8.0,3.0);
glVertex2f(16.0,3.0);
glVertex2f(16.5,3.5);
glVertex2f(17.0,4.0);
```

```
glVertex2f(16.5,4.25);
glVertex2f(16.0,4.5);
glVertex2f(15.0,4.5);
glVertex2f(15.0,5.0);
glVertex2f(14.0,5.0);
glVertex2f(11.5,6.5);
glVertex2f(10.5,6.75);
glVertex2f(7.0,6.75);
glVertex2f(5.0,5.0);
glVertex2f(7.0,5.0);
glVertex2f(6.5,4.5);
glEnd();
```

```
glBegin(GL_LINES); //connecting outer line
glVertex2d(7.0,5.0);
glVertex2d(15.0,5.0);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
glVertex2d(15.0,4.0);
glVertex2d(17.0,4.0);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
glVertex2d(15.0,3.5);
glVertex2d(16.5,3.5);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
glVertex2d(15.0,5.0);
glVertex2d(14.0,3.0);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
```

```
glVertex2d(12.0,5.0);  
glVertex2d(12.0,6.2);  
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line  
glBegin(GL_LINES);  
glVertex2d(7.0,5.0);  
glVertex2d(7.0,6.7);  
glEnd();
```

```
glBegin(GL_POLYGON); //drawing a back tyre  
glVertex2f(3.0,2.5);  
glVertex2f(3.0,2.6);  
glVertex2f(3.15,3.1);  
glVertex2f(3.2,3.2);  
glVertex2f(3.3,3.35);  
glVertex2f(3.4,3.4);  
glVertex2f(3.5,3.45);  
glVertex2f(3.6,3.55);  
glVertex2f(3.7,3.6);  
glVertex2f(3.8,3.63);  
glVertex2f(4.0,3.65);  
glVertex2f(4.2,3.7);  
glVertex2f(4.4,3.7);  
glVertex2f(4.6,3.65);  
glVertex2f(4.8,3.55);  
glVertex2f(5.0,3.45);  
glVertex2f(5.1,3.4);  
glVertex2f(5.2,3.25);  
glVertex2f(5.3,3.2);  
glVertex2f(5.4,3.0);  
glVertex2f(5.5,2.5);
```

```
glVertex2f(5.45,2.15);  
glVertex2f(5.4,1.9);  
glVertex2f(5.35,1.8);  
glVertex2f(5.2,1.6);  
glVertex2f(5.0,1.5);  
glVertex2f(4.9,1.4);
```

```
glVertex2f(4.7,1.3);  
glVertex2f(4.6,1.27);  
glVertex2f(4.4,1.25);  
glVertex2f(4.0,1.25);  
glVertex2f(3.9,1.3);  
glVertex2f(3.75,1.35);  
glVertex2f(3.6,1.4);  
glVertex2f(3.45,1.55);  
glVertex2f(3.3,1.7);  
glVertex2f(3.2,1.8);  
glVertex2f(3.1,2.2);  
glEnd();
```

```
glBegin(GL_POLYGON); //drawing front tyre  
glVertex2f(17.0,2.5);  
glVertex2f(17.0,2.6);  
glVertex2f(17.15,3.1);  
glVertex2f(17.2,3.2);  
glVertex2f(17.3,3.35);  
glVertex2f(17.4,3.4);  
glVertex2f(17.5,3.45);  
glVertex2f(17.6,3.55);  
glVertex2f(17.7,3.6);  
glVertex2f(17.8,3.63);  
glVertex2f(18.0,3.65);  
glVertex2f(18.2,3.7);  
glVertex2f(18.4,3.7);  
glVertex2f(18.6,3.65);  
glVertex2f(18.8,3.55);  
glVertex2f(19.0,3.45);  
glVertex2f(19.1,3.4);  
glVertex2f(19.2,3.25);  
glVertex2f(19.3,3.2);  
glVertex2f(19.4,3.0);
```

```
glVertex2f(19.5,2.5);  
glVertex2f(19.45,2.15);  
glVertex2f(19.4,1.9);
```

```

glVertex2f(19.35,1.8);
glVertex2f(19.2,1.6);
glVertex2f(19.0,1.5);
glVertex2f(18.9,1.4);
glVertex2f(18.7,1.3);
glVertex2f(18.6,1.27);
glVertex2f(18.4,1.25);
glVertex2f(18.0,1.25);
glVertex2f(17.9,1.3);
glVertex2f(17.75,1.35);
glVertex2f(17.6,1.4);
glVertex2f(17.45,1.55);
glVertex2f(17.3,1.7);
glVertex2f(17.2,1.8);
glVertex2f(17.1,2.2);
glEnd();
glPopMatrix();
}
void car2()
{
glPushMatrix(); //making color for outer line
glTranslated(b-2000,190.0,0.0);
glScaled(20.0,20.0,0.0);
glColor3f(1.0,1.0,0.4);
glBegin(GL_POLYGON);
glVertex2f(2.5,2.5);
glVertex2f(3.0,3.5);
glVertex2f(3.5,3.75);
glVertex2f(4.0,4.0);
glVertex2f(4.5,4.0);
glVertex2f(5.0,3.75);
glVertex2f(5.5,3.5);
glVertex2f(5.75,3.0);
glVertex2f(6.0,2.5);
glVertex2f(16.5,2.5);
glVertex2f(16.75,3.0);
glVertex2f(17.0,3.5);
glVertex2f(17.5,3.75);
glVertex2f(18.0,4.0);

```



```
glVertex2f(18.5,4.0);
glVertex2f(19.0,3.75);
glVertex2f(19.5,3.5);
glVertex2f(19.75,3.0);
glVertex2f(20.0,2.5);
glVertex2f(21.0,2.5);
glVertex2f(21.0,4.0);
glVertex2f(21.5,4.0);
glVertex2f(21.0,4.5);
glVertex2f(20.0,5.0);
glVertex2f(15.0,5.0);
glVertex2f(14.0,5.5);
glVertex2f(13.0,6.0);
glVertex2f(12.0,6.5);
glVertex2f(11.0,7.0);
glVertex2f(6.0,7.0);
glVertex2f(5.0,6.5);
glVertex2f(4.5,6.25);
glVertex2f(4.25,6.0);
glVertex2f(4.0,5.75);
glVertex2f(3.5,5.5);
glVertex2f(3.0,5.5);
glVertex2f(1.9,5.45);
glVertex2f(1.8,5.4);
glVertex2f(1.7,5.35);
glVertex2f(1.6,5.3);
glVertex2f(1.5,5.25);
glVertex2f(1.4,5.15);
glVertex2f(1.3,5.0);
glVertex2f(1.2,4.85);
glVertex2f(1.1,4.7);
glVertex2f(1.0,4.3);
glVertex2f(1.0,3.2);
glVertex2f(1.1,3.05);
glVertex2f(1.2,2.9);
glVertex2f(1.3,2.9);
glVertex2f(1.4,2.75);
glVertex2f(1.5,2.65);
glVertex2f(1.6,2.6);
```

```
glVertex2f(1.7,2.55);  
glVertex2f(1.8,2.5);  
glVertex2f(1.9,2.45);  
glVertex2f(2.0,2.5);  
glEnd();
```

```
glColor3f(1.0,1.0,1.0); //color for outer window  
glBegin(GL_POLYGON);  
glVertex2f(5.0,5.0);  
glVertex2f(14.0,5.0);  
glVertex2f(11.5,6.5);  
glVertex2f(10.5,6.75);  
glVertex2f(7.0,6.75);  
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //making outer line for car  
glBegin(GL_LINE_LOOP);  
glVertex2f(2.5,2.5);  
glVertex2f(3.0,3.5);  
glVertex2f(3.5,3.75);  
glVertex2f(4.0,4.0);  
glVertex2f(4.5,4.0);  
glVertex2f(5.0,3.75);  
glVertex2f(5.5,3.5);  
glVertex2f(5.75,3.0);  
glVertex2f(6.0,2.5);  
glVertex2f(16.5,2.5);  
glVertex2f(16.75,3.0);  
glVertex2f(17.0,3.5);  
glVertex2f(17.5,3.75);  
glVertex2f(18.0,4.0);  
glVertex2f(18.5,4.0);  
glVertex2f(19.0,3.75);  
glVertex2f(19.5,3.5);  
glVertex2f(19.75,3.0);  
glVertex2f(20.0,2.5);  
glVertex2f(21.0,2.5);  
glVertex2f(21.0,4.0);  
glVertex2f(21.5,4.0);
```

```
glVertex2f(21.0,4.5);
glVertex2f(20.0,5.0);
glVertex2f(15.0,5.0);
glVertex2f(14.0,5.5);
glVertex2f(13.0,6.0);
glVertex2f(12.0,6.5);
glVertex2f(11.0,7.0);
glVertex2f(6.0,7.0);
glVertex2f(5.0,6.5);
glVertex2f(4.5,6.25);
glVertex2f(4.25,6.0);
glVertex2f(4.0,5.75);
glVertex2f(3.5,5.5);
glVertex2f(3.0,5.5);
glVertex2f(1.9,5.45);
glVertex2f(1.8,5.4);
glVertex2f(1.7,5.35);
glVertex2f(1.6,5.3);
glVertex2f(1.5,5.25);
glVertex2f(1.4,5.15);
glVertex2f(1.3,5.0);
glVertex2f(1.2,4.85);
glVertex2f(1.1,4.7);
glVertex2f(1.0,4.3);
glVertex2f(1.0,3.2);
glVertex2f(1.1,3.05);
glVertex2f(1.2,2.9);
glVertex2f(1.3,2.9);
glVertex2f(1.4,2.75);
glVertex2f(1.5,2.65);
glVertex2f(1.6,2.6);
glVertex2f(1.7,2.55);
glVertex2f(1.8,2.5);
glVertex2f(1.9,2.45);
glVertex2f(2.0,2.5);
glEnd();

glColor3f(0.0,0.0,0.0);
glBegin(GL_LINE_LOOP); //outer line for design a car
```

```
glVertex2f(8.0,3.0);
glVertex2f(16.0,3.0);
glVertex2f(16.5,3.5);
glVertex2f(17.0,4.0);
glVertex2f(16.5,4.25);
glVertex2f(16.0,4.5);
glVertex2f(15.0,4.5);
glVertex2f(15.0,5.0);
glVertex2f(14.0,5.0);
glVertex2f(11.5,6.5);
glVertex2f(10.5,6.75);
glVertex2f(7.0,6.75);
glVertex2f(5.0,5.0);
glVertex2f(7.0,5.0);
glVertex2f(6.5,4.5);
glEnd();
```

```
glBegin(GL_LINES); //connecting outer line
glVertex2d(7.0,5.0);
glVertex2d(15.0,5.0);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
glVertex2d(15.0,4.0);
glVertex2d(17.0,4.0);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
glVertex2d(15.0,3.5);
glVertex2d(16.5,3.5);
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line
glBegin(GL_LINES);
glVertex2d(15.0,5.0);
glVertex2d(14.0,3.0);
```

```
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line  
glBegin(GL_LINES);  
glVertex2d(12.0,5.0);  
glVertex2d(12.0,6.2);  
glEnd();
```

```
glColor3f(0.0,0.0,0.0); //connecting outer line  
glBegin(GL_LINES);  
glVertex2d(7.0,5.0);  
glVertex2d(7.0,6.7);  
glEnd();
```

```
glBegin(GL_POLYGON); //drawing a back tyre  
glVertex2f(3.0,2.5);  
glVertex2f(3.0,2.6);  
glVertex2f(3.15,3.1);  
glVertex2f(3.2,3.2);  
glVertex2f(3.3,3.35);  
glVertex2f(3.4,3.4);  
glVertex2f(3.5,3.45);  
glVertex2f(3.6,3.55);  
glVertex2f(3.7,3.6);  
glVertex2f(3.8,3.63);  
glVertex2f(4.0,3.65);  
glVertex2f(4.2,3.7);  
glVertex2f(4.4,3.7);  
glVertex2f(4.6,3.65);  
glVertex2f(4.8,3.55);  
glVertex2f(5.0,3.45);  
glVertex2f(5.1,3.4);  
glVertex2f(5.2,3.25);  
glVertex2f(5.3,3.2);  
glVertex2f(5.4,3.0);  
glVertex2f(5.5,2.5);
```

```
glVertex2f(5.45,2.15);  
glVertex2f(5.4,1.9);
```

```
glVertex2f(5.35,1.8);
glVertex2f(5.2,1.6);
glVertex2f(5.0,1.5);
glVertex2f(4.9,1.4);
glVertex2f(4.7,1.3);
glVertex2f(4.6,1.27);
glVertex2f(4.4,1.25);
glVertex2f(4.0,1.25);
glVertex2f(3.9,1.3);
glVertex2f(3.75,1.35);
glVertex2f(3.6,1.4);
glVertex2f(3.45,1.55);
glVertex2f(3.3,1.7);
glVertex2f(3.2,1.8);
glVertex2f(3.1,2.2);
glEnd();
```

```
glBegin(GL_POLYGON); //drawing front tyre
glVertex2f(17.0,2.5);
glVertex2f(17.0,2.6);
glVertex2f(17.15,3.1);
glVertex2f(17.2,3.2);
glVertex2f(17.3,3.35);
glVertex2f(17.4,3.4);
glVertex2f(17.5,3.45);
glVertex2f(17.6,3.55);
glVertex2f(17.7,3.6);
glVertex2f(17.8,3.63);
glVertex2f(18.0,3.65);
glVertex2f(18.2,3.7);
glVertex2f(18.4,3.7);
glVertex2f(18.6,3.65);
glVertex2f(18.8,3.55);
glVertex2f(19.0,3.45);
glVertex2f(19.1,3.4);
glVertex2f(19.2,3.25);
glVertex2f(19.3,3.2);
glVertex2f(19.4,3.0);
```

```
glVertex2f(19.5,2.5);
glVertex2f(19.45,2.15);
glVertex2f(19.4,1.9);
glVertex2f(19.35,1.8);
glVertex2f(19.2,1.6);
glVertex2f(19.0,1.5);
glVertex2f(18.9,1.4);
glVertex2f(18.7,1.3);
glVertex2f(18.6,1.27);
glVertex2f(18.4,1.25);
glVertex2f(18.0,1.25);
glVertex2f(17.9,1.3);
glVertex2f(17.75,1.35);
glVertex2f(17.6,1.4);
glVertex2f(17.45,1.55);
glVertex2f(17.3,1.7);
glVertex2f(17.2,1.8);
glVertex2f(17.1,2.2);
glEnd();
glPopMatrix();
}
```

```
void myinit()
{
```

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0,1346.0,0.0,728.0);
}
```

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

```
/*call back functions*/  
glutDisplayFunc(mydisplay);  
glutKeyboardFunc(myKeyboard);  
glutMouseFunc(myMouse);
```

```
myinit();  
glutMainLoop();  
}
```


7. TEST CASES:

| Sl No | Inputs Given | Expected Outputs | Generated Outputs | Result |
|--------------|--------------------------------|---|--|---------------|
| 1 | No Mouse Clicks | No movement of the vehicles and signal is in red colour | No movement of the vehicles and signal is in red colour | Pass |
| 2 | Right click and hold the mouse | No movement of the vehicles and signal should turn yellow | No movement of the vehicles and the signal turned yellow | Pass |
| 3 | Release the Mouse Click | Signal turned Green and the Movements of the vehicles | Signal turned Green and Movements of the vehicles | Pass |
| 4 | Left click from the Mouse | The Signal should again turn into Red and the movements of the vehicles must stop | The Signal turned into Red and the movements of the vehicles was stopped | Pass |

8. OUTPUT/SAMPLE SCREENSHOTS:



Fig-1: Start page that gives information about the team members

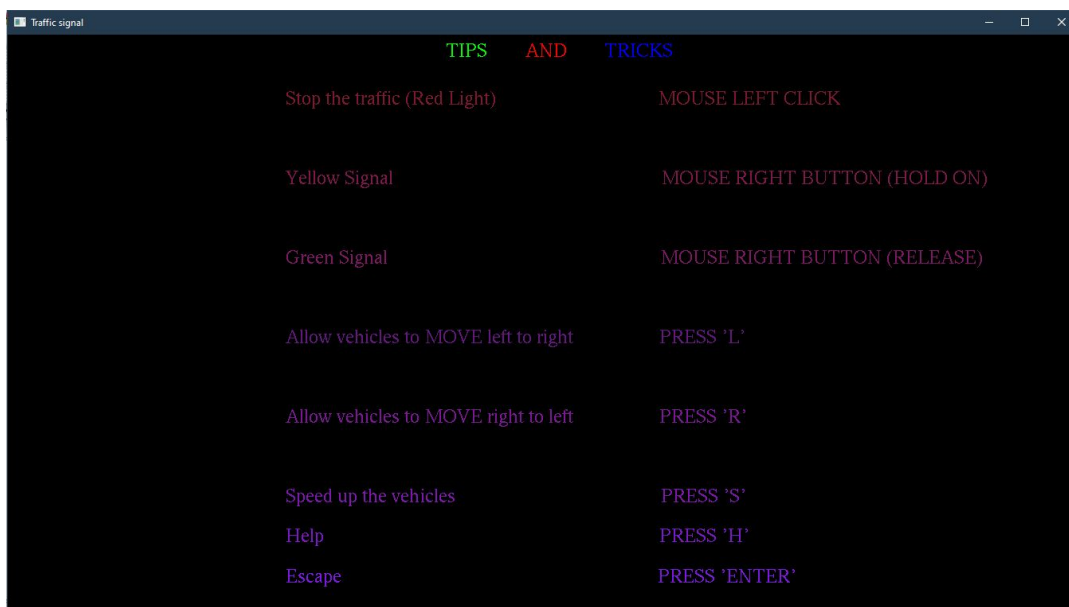


Fig-2: Help menu that explains how to control the traffic

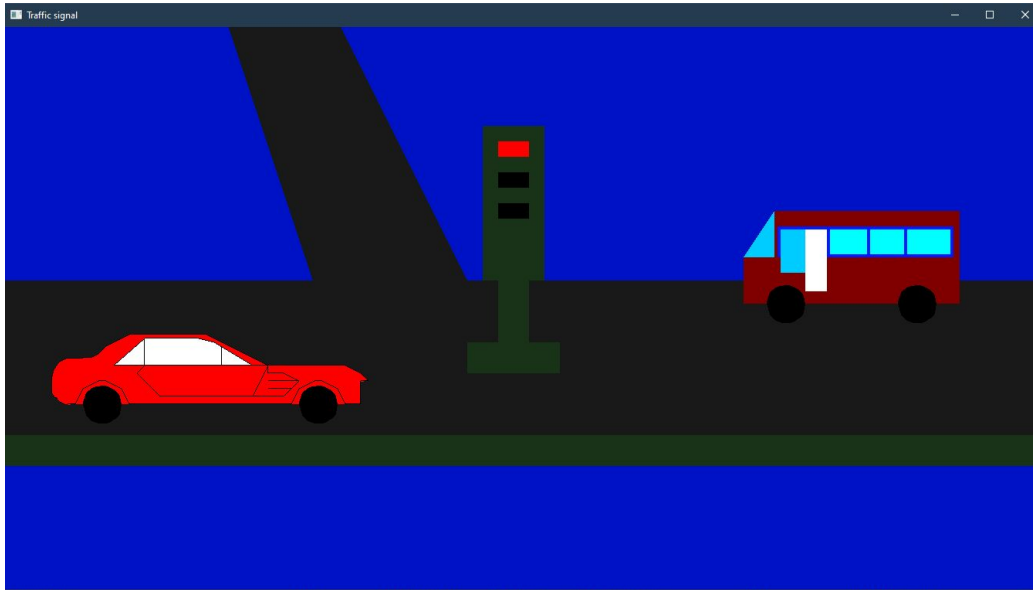


Fig-3:Signal is Red (No vehicles will move)

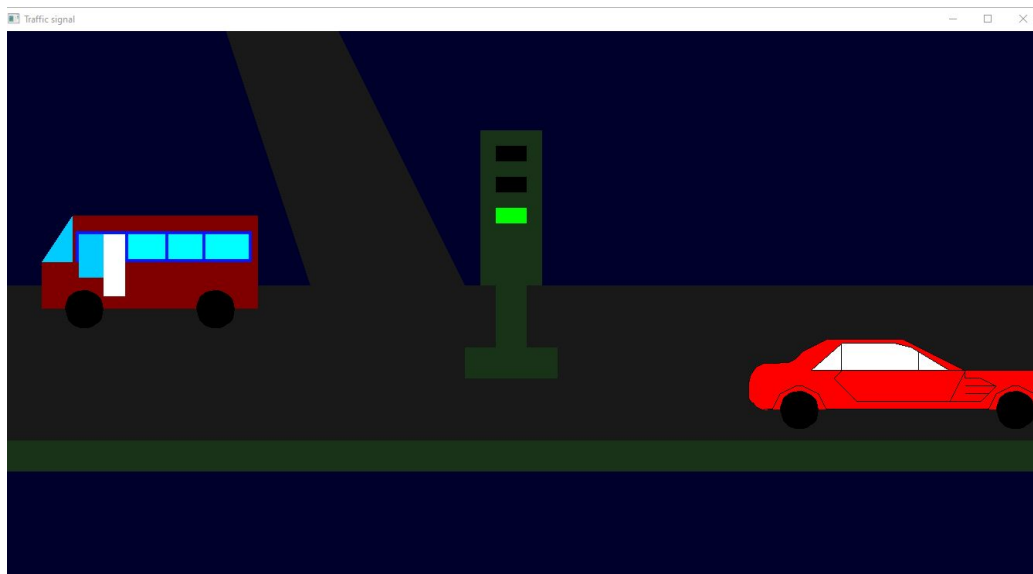


Fig-4: Signal is Green (Vehicles are under motion)

9 . REFERENCES:

[1]https://en.wikipedia.org/wiki/Traffic_simulation

[2]<https://www.openglprojects.in>