Computer Graphics (UCS505)

Project on

Road Safety on Highway

Submitted By

Aditi Nirwan	102003380
Jahnvi Gangwar	102003372
Mukul Singhal	102003370

3CO15

B.E. Third Year – COE

Submitted To:

Dr. Rajkumar Tekchandani

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147001

Table of Contents

Introduction to the Project

Our team's OpenGL project aims to provide an enjoyable and interactive learning experience by simulating the movement of cars and buses on a highway, while also displaying road safety rules on the screen. This involves creating accurate vehicle models and using animation to simulate their movement on the road network. Our menu-driven project showcases two different simulations of highway traffic, one during the day and the other at night. To create a visually captivating and authentic simulation, the project employs various computer graphics techniques, including lighting, shading, texturing, and animation.

This project is an excellent example of how computer graphics and animation can be used to create an engaging and interactive learning experience. The simulation of highway traffic with road safety rules and regulations presented on the screen provides an excellent opportunity for viewers to learn about the importance of safe driving while having fun at the same time. The realistic vehicle models and dynamic lighting and shading effects make the simulation more visually appealing and engaging, adding to the overall learning experience.

Computer Graphics concepts Used

In our project, various computer graphics concepts are employed. The key concepts used in this project are:

- **2D Graphics Rendering**: In this project, 2D graphics rendering is used to create a graphical representation of the cars, bus, roads as well as the sun.
- **OpenGL:** OpenGL (Open Graphics Library) is a cross-platform graphics API used for rendering 2D and 3D graphics. In this project, OpenGL functions are employed to make the bus and car viewable.
- **Coordinate System and Transformations:** The car traffic light system project utilizes a Cartesian coordinate system to position the traffic lights on the screen. The traffic lights are represented by their (x, y) coordinates and their state is updated by modifying their coordinates based on the current state of the system. OpenGL transformation functions such as glMatrixMode, glLoadIdentity, and glOrtho are employed to establish the coordinate system and guarantee accurate display of the traffic lights on the screen.
- **Animation and Game Loop:** The game loop is an essential component of the project, responsible for updating the game state and rendering the graphics. The game loop is implemented using OpenGL's timer function, glutTimerFunc, which ensures the smooth and consistent movement of the snake and other game objects. The vehicle's animation is achieved by updating its position and redrawing the game objects at regular intervals, creating the illusion of continuous motion.

User-Defined Functions

In this project, several user-defined functions are implemented to handle various aspects, The key user-defined functions used in this project are:

- **void myinit():** This function is responsible for initializing the matrix mode to projections. The function glLoadIdentity(), loads the identity matrix into the projection matrix. This resets the projection matrix to its default state, so that any subsequent transformations will start from scratch. The function gluOrtho2D() specifies how the 3D scene will be mapped onto the 2D screen, based on the specified range of x and y values. This is a typical setup for a 2D drawing environment, where the x-axis and y-axis represent the width and height of the screen, respectively.

- **void mydisplay():** This function clears the colour buffer bit ,it calls the display function and swaps front and back buffers.

- **void display():** In this function we specify the glutTimerFunc() where we have our update function. After that we call our animated object functions.

- **void car() and void car2() :** These functions are drawing the car's body and windows by specifying a series of 2D points using the glVertex2f function within glBegin and glEnd functions, and then colouring the shapes using the glColor3f function. The glPushMatrix function is used to save the current matrix on a stack so that it can be restored later with glPopMatrix. The function first translates the car horizontally by b units, moves it down by 220 units along the y-axis and scales it up by a factor of 20 in both the x and y directions using glTranslated and glScaled. The glColor3f function is then used to set the color for the inner and outer parts of the car. The glBegin function is used to indicate the start of a new shape to be drawn, and glEnd is used to indicate the end of that shape.

- **void bus():** These functions are drawing the bus's body ,windows and tyres by specifying a series of 2D points using the glVertex2f function within glBegin and glEnd functions, and then coloring the shapes using the glColor3f function. The glPushMatrix function is used to save the current matrix on a stack so that it can be restored later with glPopMatrix. The function first translates the car horizontally by b units, moves it down by 220 units along the y-axis and scales it up by a factor of

40 in both the x and y directions using glTranslated and glScaled. The glColor3f function is then used to set the color for the inner and outer parts of the bus. The glBegin function is used to indicate the start of a new shape to be drawn, and glEnd is used to indicate the end of that shape.

- **void background() and void sun()** : The function, background(), is used to draw a background for a graphical scene. The function uses OpenGL commands to draw two polygons: one for the sky and one for the grass field. The second function, sun(), is used to draw a yellow sun in the scene. The function also uses OpenGL commands to draw a polygon, but this time with a loop to create a circle shape.
- **myKeyboard():** It is a user-defined function that can be used with the GLUT framework in order to handle keyboard input events. It is a call back function that is called by GLUT whenever the user presses a key on the keyboard. In our simulation, if you press ‘D’ key on the keyboard, you can view the highway in the Day Mode and on pressing ‘N’ key on the keyboard, you can view the highway in the Night Mode (default mode).

Code

```
#include<stdio.h>
#include<GLUT/glut.h>
#include<cmath>
#include<string.h>

void background();
void sun();
void bus();
void road();
void car();
void signal();
void mydisplay();
void display();
void update();
void myKeyboard();

GLint a=300,b=-300;
GLfloat cx = 5,cy=15,radius=1;
GLfloat p=0,q=0,r=0;

int count = 300,flag=1;

int rule_counter=0,rules_length=5;
char* rules[5] = {"Drive slower, live longer.","Do not use phone while
driving.","Speed thrills but kills!.","Better Late than Never!","Do not mix
drinking and driving."};


void output(int x, int y, char *string,void *font)
{
    int len, i;
    flag?glColor3f(1, 1, 1):glColor3f(0,0,0);
    glRasterPos2f(x,y);
    len=(int) strlen(string);
    for (i = 0; i < len; i++)
    {
        glutBitmapCharacter(font,string[i]);
    }
}


void mydisplay(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    display();
    glutSwapBuffers();
}

void myKeyboard( unsigned char key, int x, int y )
{
    switch(key)
    {
        case 13: {rule_counter = (rule_counter+1)%rules_length;break;}
        case 'd':
        case 'D':flag=0;break;
        case 'N':
        case 'n':flag=1; break;
```

```
        default:break;
    }
}

void update(int value)
{
    a=a-6;
    b=b+6;
    count--;

    if(!count){
        a=300;
        b=-300;
        count=400;
    }

    glutPostRedisplay();
}

void display(void)
{
    glutTimerFunc(50,update,0);
    glClear(GL_COLOR_BUFFER_BIT);
    background();
    output(1200,700,"Project Name: Highway",GLUT_BITMAP_TIMES_ROMAN_10);
    output(1200,680,"Made By: ",GLUT_BITMAP_TIMES_ROMAN_10);
    output(1200,660,"Jahnvi Gangwar 102003372",GLUT_BITMAP_TIMES_ROMAN_10);
    output(1200,640,"Aditi Nirwan  102003380",GLUT_BITMAP_TIMES_ROMAN_10);
    output(1200,620,"Mukul Singhal 102003370",GLUT_BITMAP_TIMES_ROMAN_10);
    output(70,100,"INSTRUCTIONS: ",GLUT_BITMAP_TIMES_ROMAN_24);
    output(100,70,"Press D for Day View",GLUT_BITMAP_TIMES_ROMAN_24);
    output(100,45,"Press N for Night View",GLUT_BITMAP_TIMES_ROMAN_24);
    output(400,70,"Press Enter for new rules",GLUT_BITMAP_TIMES_ROMAN_24);
    output(900,640,"NH-8",GLUT_BITMAP_TIMES_ROMAN_24);
    sun();
    road();
    bus();
    signal();
    output(510, 550, rules[rule_counter], GLUT_BITMAP_TIMES_ROMAN_24);
    car();
    glFlush();
}

void background(){
    glPushMatrix();
    glScaled(40.0,40.0,0.0);

    !flag?glColor3f(0.529, 0.808, 0.922):glColor3f(0,0,0.5);; // sky
    glBegin(GL_POLYGON);
    glVertex2f(0,10);
    glVertex2f(0,20);
    glVertex2f(40,20);
    glVertex2f(40,10);
    glEnd();

    glColor3f(0.133, 0.545, 0.133); //grass field
    glBegin(GL_POLYGON);
    glVertex2f(0,10);
    glVertex2f(0,0);
    glVertex2f(40,0);
```

```
    glVertex2f(40,10);
    glEnd();

    glPopMatrix();
}


void sun(){
    glPushMatrix();
    glScaled(40.0,40.0,0.0);
    glBegin(GL_POLYGON);
    !flag?glColor3f(1,1,0):glColor3f(1,1,1);
    for (int i = 0; i < 50; i++)
    {
        float theta = i * 2.0f * 3.14159f / 50;
        float x = radius * cos(theta) + cx;
        float y = radius * sin(theta) + cy;
        glVertex2f(x, y);
    }
    glEnd();
    glPopMatrix();
}

void signal()
{
    glPushMatrix();
    glScaled(40.0,40.0,0.0);

    glColor3f(0.545, 0.271, 0.075); //stand
    glBegin(GL_POLYGON);
    glVertex2f(15,7);
    glVertex2f(15,8);
    glVertex2f(18,8);
    glVertex2f(18,7);
    glEnd();

    glBegin(GL_POLYGON); //pole
    glVertex2f(16,7);
    glVertex2f(17,8);
    glVertex2f(17,15);
    glVertex2f(16,15);
    glEnd();

    glBegin(GL_POLYGON); //board
    glVertex2f(11.5,15);
    glVertex2f(22,15);
    glVertex2f(22,10);
    glVertex2f(11.5,10);
    glEnd();
    glPopMatrix();
}

void road()
{
    glPushMatrix();
    glScaled(40.0,40.0,0.0);
    glBegin(GL_POLYGON); //road
    glColor3f(0.1,0.1,0.1);
    glVertex2f(0,5);
    glVertex2f(40,5);
```

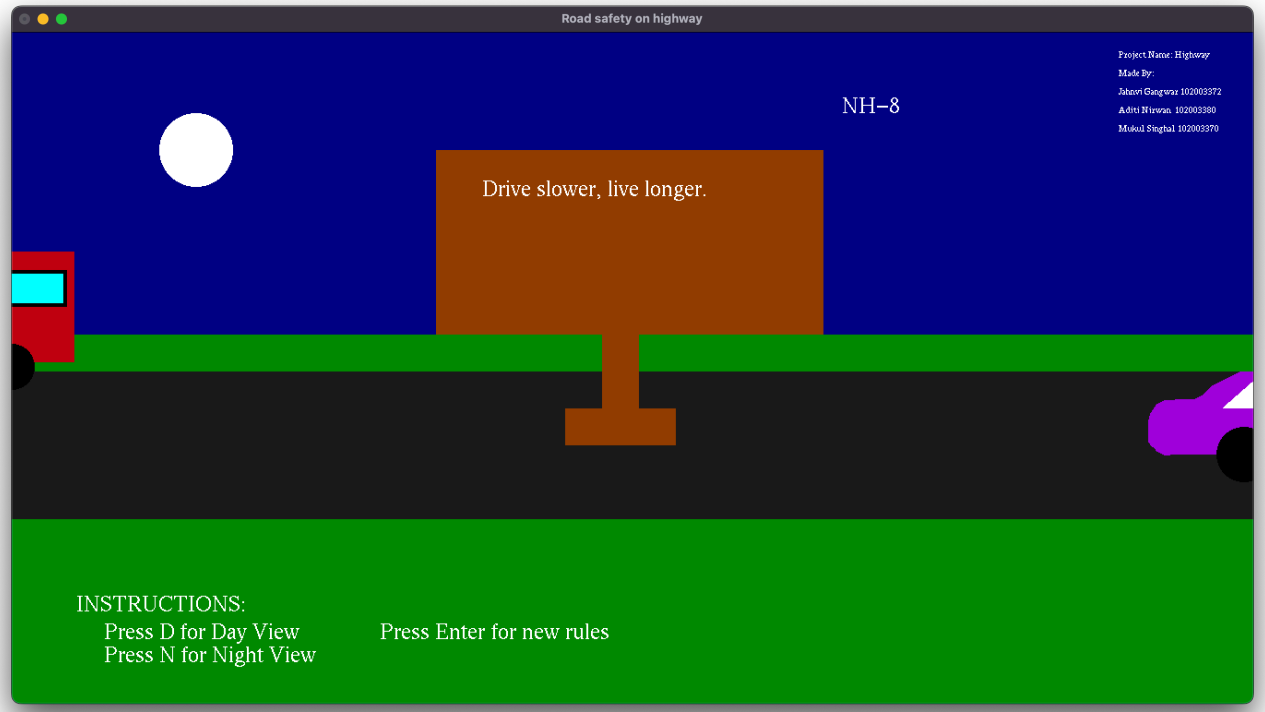
```
    glVertex2f(40,9);
    glVertex2f(0,9);
    glEnd();
    glPopMatrix();
}

void bus () {
    glPushMatrix ();
    glTranslated (a, 50.0, 0.0);
    glScaled (40.0, 40.0, 0.0);
    glColor3f (0.698, 0.133, 0.133);    //bus body - fill
    glBegin (GL_POLYGON);
    glVertex2f (25, 8);
    glVertex2f (25, 9.5);
    glVertex2f (26, 11);
    glVertex2f (32, 11);
    glVertex2f (32, 8);
    glEnd ();
    glColor3f (1.000, 0.894, 0.882);    //Doors
    glBegin (GL_POLYGON);
    glVertex2f (27, 8.4);
    glVertex2f (27, 10.4);
    glVertex2f (27.7, 10.4);
    glVertex2f (27.7, 8.4);
    glEnd ();
    glColor3f (0, 0, 0);        //window-frame
    glBegin (GL_POLYGON);
    glVertex2f (27.7, 9.5);
    glVertex2f (27.7, 10.5);
    glVertex2f (31.8, 10.5);
    glVertex2f (31.8, 9.5);
    glEnd ();
    glColor3f(0,1,1); // back windows
    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();
    glColor3f (0, 1, 1);        // front window
    glBegin (GL_POLYGON);
    glVertex2f (25, 9.5);
    glVertex2f (26, 11);
    glVertex2f (26, 9.5);
    glEnd ();
    glPopMatrix ();
    glPushMatrix (); //front tyre
    glTranslated (a + 970, 320, 0.0);
```

```
    glScaled (10.0, 10.0, 0.0);
    glBegin (GL_POLYGON);
    glColor3f (0, 0, 0);
    for (int i = 0; i < 50; i++){
        float theta = i * 2.0f * 3.14159f / 50;
        float x = 2.5 * cos (theta) + 7.2;
        float y = 2.5 * sin (theta) + 4.5;
        glVertex2f (x, y);
    }
    glEnd ();
    glPopMatrix ();
    glPushMatrix ();//back tyre
    glTranslated (a + 1140, 320, 0.0);
    glScaled (10.0, 10.0, 0.0);
    glBegin (GL_POLYGON);
    glColor3f (0, 0, 0);
    for (int i = 0; i < 50; i++){
        float theta = i * 2.0f * 3.14159f / 50;
        float x = 2.5 * cos (theta) + 7.2;
        float y = 2.5 * sin (theta) + 4.5;
        glVertex2f (x, y);
    }
    glEnd ();
    glPopMatrix ();
}
void car () {
    glPushMatrix ();        //making color for outer line
    glTranslated (b, 220.0, 0.0);
    glScaled (20.0, 20.0, 0.0);
    glColor3f (0.580, 0.000, 0.827);
    glBegin (GL_POLYGON);        //inner car - fill
    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);    //window color
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 ();

glBegin (GL_POLYGON);
glColor3f (0, 0, 0);
for (int i = 0; i < 50; i++)
{
    float theta = i * 2.0f * 3.14159f / 50;
    float x = 1.5 * cos (theta) + 6.2;
    float y = 1.5 * sin (theta) + 2.5;
    glVertex2f (x, y);
}
glEnd ();

glBegin (GL_POLYGON);
glColor3f (0, 0, 0);
for (int i = 0; i < 50; i++)
{
    float theta = i * 2.0f * 3.14159f / 50;
    float x = 1.5 * cos (theta) + 16.2;
    float y = 1.5 * sin (theta) + 2.5;
    glVertex2f (x, y);
}
glEnd ();
```

```
    glPopMatrix ();
}
void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glPointSize(1.0);
    gluOrtho2D(0.0,1346.0,0.0,728.0);
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(1346,728);
    glutInitWindowPosition(0,0);
    glutCreateWindow("Road safety on highway");
    glutDisplayFunc(mydisplay);
    glutKeyboardFunc(myKeyboard);
    myinit();
    glutMainLoop();
    return 0;
}
```

Output ScreenShots




Road safety on highway
Project Name: Highway
Made By:
Jahnvi Gangwar 102003372
Aditi Nirwan 102003380
Mukul Singhal 102003370
NH–8
Do not use phone while driving.
INSTRUCTIONS:
Press D for Day View
Press N for Night View
Press Enter for new rules