

**Mawlana Bhashani Science and Technology University**

**Santosh, Tangail -1902**

**Department of Computer Science and Engineering**

**Course Title:** Computer Graphics and Animation Technique Lab

**Lab Report Title:** Draw an object (car) in OpenGL and Animate the car both in forward and backward direction.

**Course Code:** CSE 3206

**LAB REPORT**

|  |  |
| --- | --- |
| **Submitted By:**  Name : Syed Mahir Faisal  ID : CE20005  Year : 3rd Semester : 2nd  Session : 2019-2020  Mawlana Bhashani Science and Technology Unieversity | **Submitted To:**  Lubna Yasmin Pinky  Assistant Professor  Department of CSE  Mawlana Bhashani Science and Technology Unieversity |

Date of Submission:16/10/2023

**Problem Statement:** Draw an object (car) in OpenGL and Animate the car both in forward and backward direction.

\*For backward, rotate the car for real visualization.

**Source Code:**

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

#include <math.h>

GLint b = 300;

float counter = 300.0;

int direction = 1; // 1 for forward, -1 for backward

void initOpenGl()

{

glClearColor(0.5, 0.5, 0.5, 1); // Background Color

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0, 700, 0, 500);

glMatrixMode(GL\_MODELVIEW);

}

void wheel(int x, int y)

{

float th;

glBegin(GL\_POLYGON);

glColor3f(0, 0, 0);

for (int i = 0; i < 360; i++)

{

th = i \* (3.1416 / 180);

glVertex2f(x + 20 \* cos(th), y + 20 \* sin(th));

}

glEnd();

}

void car()

{

glLoadIdentity();

counter = counter + (0.5 \* direction);

glTranslated(counter, 100, 0.0);

glBegin(GL\_POLYGON);

glVertex2f(100, 100);

glVertex2f(100, 160);

glVertex2f(450, 160);

glVertex2f(450, 100);

glBegin(GL\_POLYGON);

glVertex2f(150, 160);

glVertex2f(200, 200);

glVertex2f(400, 200);

glVertex2f(450, 160);

glEnd();

wheel(200, 100);

wheel(380, 100);

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 1.0, 0.0);

car();

glutSwapBuffers();

glFlush();

}

void update(int value)

{

counter += 0.5 \* direction; // Adjust the position based on direction

if (counter >= 600) {

direction = -1; // Change direction when reaching the right edge

}

else if (counter <= 100) {

direction = 1; // Change direction when reaching the left edge

}

glutPostRedisplay();

glutTimerFunc(10, update, 0);

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA | GLUT\_DEPTH);

glutInitWindowSize(700, 500);

glutInitWindowPosition(0, 0);

glutCreateWindow("Car Moving");

initOpenGl();

glutDisplayFunc(display);

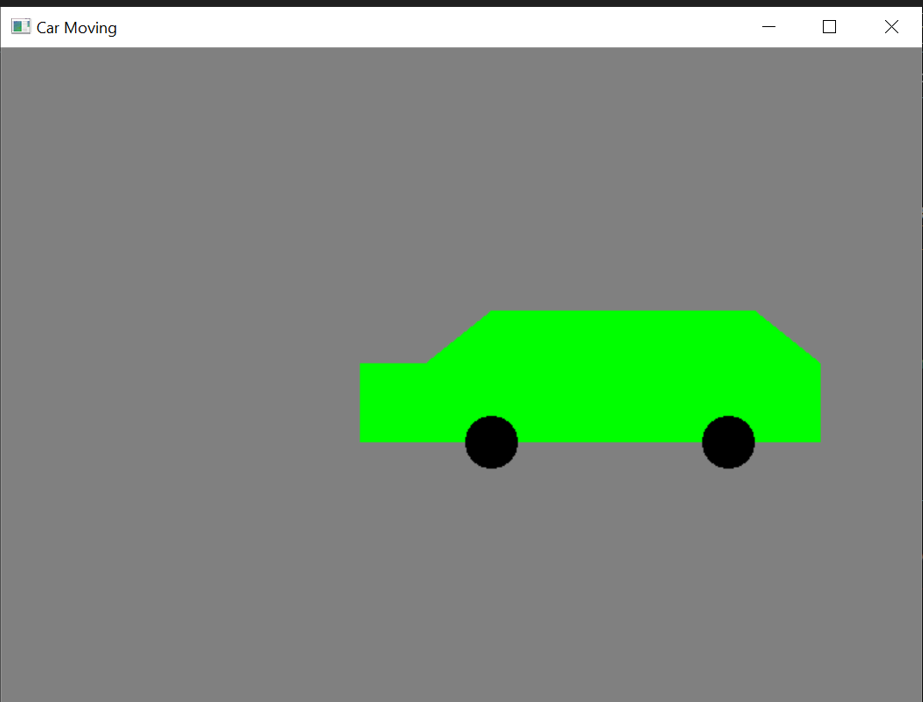
glutTimerFunc(0, update, 0); // Start the update timer

glutMainLoop();

return 0;

}

**Output \*Forward Direction:**

****

**Output \*Backward Direction:**

**A green car on a gray background

Description automatically generated**

**Limitation**: I could not rotate the shape of the car while it is moving backwards.

**Conclusion:**

In conclusion, the provided code is a simple OpenGL program for animating a moving car. It defines a car shape and two wheels, allowing the car to move from left to right on the screen continuously. The car's movement is achieved by adjusting its position using a timer callback function, changing its direction when it reaches the screen's edges. The code sets up an OpenGL window, initializes the necessary parameters, and defines functions for drawing the car and wheels. The "display" function is responsible for rendering the car on the screen, and the "update" function controls the car's movement by updating its position.