**Lab Taks-5**

|  |
| --- |
| **Question-**  Create an animation using two box that will move in the opposite direction. You must change enlarge the x and y axis. |
| **Graph Plot (Picture)-** |
| **Code-**  **#include<cstdio>**  **#include <GL/gl.h>**  **#include <GL/glut.h>**  **GLfloat position1 = 0.0f;**  **GLfloat position2 = 0.0f;**  **GLfloat speed = 0.2f;**  **void box1() //id box1**  **{**  **glBegin(GL\_QUADS);**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(-0.2f, -0.2f);**  **glVertex2f( 0.2f, -0.2f);**  **glVertex2f( 0.2f, 0.2f);**  **glVertex2f(-0.2f, 0.2f);**  **glEnd();**  **}**  **void box2()**  **{**  **glBegin(GL\_QUADS); //id box2**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(-0.2f, 0.4f);**  **glVertex2f(0.2f, 0.4f);**  **glVertex2f(0.2f, 0.8f);**  **glVertex2f(-0.2f, 0.8f);**  **glEnd();**  **}**  **void update(int value) {**  **if(position1 <-3.0)**  **position1 = 2.0f;**  **position1 -= speed;**  **if(position2 >3.0)**  **position2 = -2.0f;**  **position2 += speed;**  **glutPostRedisplay();**  **glutTimerFunc(100, update, 0);**  **}**  **void init() {**  **gluOrtho2D(-2,2,-2,2);**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glPushMatrix();**  **glTranslatef(position1,0.0f, 0.0f);**  **box1();**  **glPopMatrix();**  **glPushMatrix();**  **glTranslatef(position2,0.0f, 0.0f);**  **box2();**  **glPopMatrix();**  **glFlush();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitWindowSize(480, 320);**  **glutInitWindowPosition(50, 50);**  **glutCreateWindow("Two Box Translation Animation");**  **glutDisplayFunc(display);**  **init();**  **glutTimerFunc(100, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Design a car |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **void carwindowfront()**  **{**  **glBegin(GL\_POLYGON); //id windowfront**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(-0.09f, 0.39f);**  **glVertex2f(-0.19f, 0.21f);**  **glVertex2f(-0.01f, 0.21f);**  **glVertex2f(-0.01f, 0.391f);**  **glEnd();**  **}**  **void carwindowback()**  **{**  **glBegin(GL\_POLYGON); //id 11windowback**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(0.01f, 0.39f);**  **glVertex2f(0.19f, 0.39f);**  **glVertex2f(0.29f, 0.21f);**  **glVertex2f(0.01f, 0.211f);**  **glEnd();**  **}**  **void carbody()**  **{**  **glBegin(GL\_POLYGON); //id 11window**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(-0.2f, 0.2f);**  **glVertex2f(-0.4f, 0.2f);**  **glVertex2f(-0.4f, 0.0f);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(0.4f, 0.2f);**  **glVertex2f(0.3f, 0.2f);**  **glVertex2f(0.2f, 0.4f);**  **glVertex2f(-0.1f, 0.4f);**  **glEnd();**  **}**  **void carwheel()**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(1.0,0,1.0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.10;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **}**  **void car()**  **{**  **carbody();**  **carwindowfront();**  **carwindowback();**  **glTranslatef(-0.2,0.0f, 0.0f);**  **carwheel();**  **glTranslatef(0.4,0.0f, 0.0f);**  **carwheel();**  **}**  **void init() {**  **gluOrtho2D(-2,2,-2,2);**  **}**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **car();**  **glPopMatrix();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(480, 300); // Set the window's initial width & height**  **glutInitWindowPosition(50, 50);**  **glutCreateWindow("Running Car"); // Create a window with the given title**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **init(); // Enter the event-processing loop**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Create a simple traffic signal scenario using your car code from the previous question and the concept of rotation and scaling. There will be two cars moving in opposite directions. |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **GLfloat position1 = 0.0f;**  **GLfloat speed = 0.1f;**  **void carwindowfront()**  **{**  **glBegin(GL\_POLYGON); //id windowfront**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(-0.09f, 0.39f);**  **glVertex2f(-0.19f, 0.21f);**  **glVertex2f(-0.01f, 0.21f);**  **glVertex2f(-0.01f, 0.391f);**  **glEnd();**  **}**  **void carwindowback()**  **{**  **glBegin(GL\_POLYGON); //id 11windowback**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(0.01f, 0.39f);**  **glVertex2f(0.19f, 0.39f);**  **glVertex2f(0.29f, 0.21f);**  **glVertex2f(0.01f, 0.211f);**  **glEnd();**  **}**  **void carbody()**  **{**  **glBegin(GL\_POLYGON); //id 11window**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(-0.2f, 0.2f);**  **glVertex2f(-0.4f, 0.2f);**  **glVertex2f(-0.4f, 0.0f);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(0.4f, 0.2f);**  **glVertex2f(0.3f, 0.2f);**  **glVertex2f(0.2f, 0.4f);**  **glVertex2f(-0.1f, 0.4f);**  **glEnd();**  **}**  **void carwheel()**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3f(1.0,0,1.0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.10;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **}**  **void car()**  **{**  **carbody();**  **carwindowfront();**  **carwindowback();**  **glTranslatef(-0.2,0.0f, 0.0f);**  **carwheel();**  **glTranslatef(0.4,0.0f, 0.0f);**  **carwheel();**  **}**  **void update(int value) {**  **if(position1 <-3.0)**  **position1 = 2.0f;**  **position1 -= speed;**  **glutPostRedisplay();**  **glutTimerFunc(100, update, 0);**  **}**  **void init() {**  **gluOrtho2D(-2,2,-2,2);**  **}**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **//glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(position1,0.0f, 0.0f);**  **car();**  **glPopMatrix();**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(480, 300); // Set the window's initial width & height**  **glutInitWindowPosition(50, 50);**  **glutCreateWindow("Running Car"); // Create a window with the given title**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **init(); // Enter the event-processing loop**  **glutTimerFunc(100, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |