**Lab Taks-5**

Submission Guidelines-

* Rename the file with your serial number only
* Must submit within the announced time.
* Must include resources for all the section in the table

|  |
| --- |
| **Question-1**  Create an animation using two box that will move in the opposite direction. |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **float \_move1 = 0.0f;**  **float \_move = 0.0f;**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity(); //Reset the drawing perspective**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.5f, 0.1f);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(-0.4f, 0.0f);**  **glVertex2f(-0.4f, 0.1f);**  **glEnd();**  **glPopMatrix();**  **glColor3d(0,1,0);**  **glLoadIdentity(); //Reset the drawing perspective**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move1, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.6f, 0.1f);**  **glVertex2f(0.7f, 0.1f);**  **glVertex2f(0.7f, 0.0f);**  **glVertex2f(0.6f, 0.0f);**  **glEnd();**  **glPopMatrix();**  **glutSwapBuffers();**  **}**  **void update(int value) {**  **\_move += .02;**  **if(\_move > 1.0)**  **{**  **\_move = 0.0;**  **}**  **\_move1 -= .02;**  **if(\_move1 < -1.0)**  **{**  **\_move1 = 0.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Transformation");**  **glutDisplayFunc(drawScene);**  **gluOrtho2D(-2,2,-2,2);**  **glutTimerFunc(20, update, 0); //Add a timer**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question-2**  Design a car which will have rotating wheels. |
| **Graph Plot (Picture)-A drawing of a car  Description automatically generated** |
| **Code-**  **#include <iostream>**  **#include <GL/gl.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **using namespace std;**  **float \_angle1 = 0.0f;**  **// Function to draw a filled circle**  **void circle(float radius, float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POINTS);**  **for (int i = 0; i < 200; i++) {**  **glColor3f(r, g, b);**  **float pi = 3.1416;**  **float A = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(A);**  **float y = radius \* sin(A);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **// Function to draw a car**  **void car() {**  **glColor3f(1.0, 0.0, 0.0);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.5f, 1.0f);**  **glVertex2f(1.5f, 1.0f);**  **glVertex2f(2.0f, -0.5f);**  **glVertex2f(-2.0f, -0.5f);**  **glEnd();**  **glColor3f(1.0, 1.0, 0.0);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.6f, 2.0f);**  **glVertex2f(1.01f, 2.0f);**  **glVertex2f(0.83f, 1.0f);**  **glVertex2f(-.7f, 1.0f);**  **glEnd();**  **// Draw wheels**  **}**  **void wheel (){**  **glPushMatrix();**  **glTranslatef(-1.0f, -0.7f, 0.0f);**  **glRotatef(\_angle1, 0.0f, 0.0f, 1.0f);**  **circle(0.5, 0.0, 0.0, 1.0, 1.0, 0.0);**  **glPopMatrix();**  **glPushMatrix();**  **glTranslatef(1.0f, -0.7f, 0.0f);**  **glRotatef(\_angle1, 0.0f, 0.0f, 1.0f);**  **circle(0.5, 0.0, 0.0, 1.0, 1.0, 0.0);**  **glPopMatrix();**  **}**  **void update1(int value) {**  **\_angle1+=2.0f;**  **if(\_angle1 > 360.0)**  **{**  **\_angle1-=360;**  **}**  **glutPostRedisplay(); //Notify GLUT that the display has changed**  **glutTimerFunc(20, update1, 0); //Notify GLUT to call update again in 25 milliseconds**  **}**  **// Display function**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **car();**  **wheel();**  **glutSwapBuffers(); // Swap the buffers to display the content**  **}**  **// Main function**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Rotating Car");**  **gluOrtho2D(-6, 6, -6, 6);**  **glutTimerFunc(20, update1, 0);**  **glutDisplayFunc(drawScene);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-A computer screen shot of a red and yellow object  Description automatically generated** |

|  |
| --- |
| **Question-3**  Now move your car of question-2 from left to right in a loop. |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <iostream>**  **#include <GL/gl.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **using namespace std;**  **float \_move = 0.0f;**  **float \_angle1 = 0.0f;**  **// Function to draw a filled circle**  **void circle(float radius, float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POINTS);**  **for (int i = 0; i < 200; i++) {**  **glColor3f(r, g, b);**  **float pi = 3.1416;**  **float A = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(A);**  **float y = radius \* sin(A);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **// Function to draw a car**  **void car() {**  **glColor3f(1.0, 0.0, 0.0);**  **glBegin(GL\_QUADS);**  **glVertex2f(-1.5f, 1.0f);**  **glVertex2f(1.5f, 1.0f);**  **glVertex2f(2.0f, -0.5f);**  **glVertex2f(-2.0f, -0.5f);**  **glEnd();**  **glColor3f(1.0, 1.0, 0.0);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.6f, 2.0f);**  **glVertex2f(1.01f, 2.0f);**  **glVertex2f(0.83f, 1.0f);**  **glVertex2f(-.7f, 1.0f);**  **glEnd();**  **// Draw wheels**  **}**  **void wheel (){**  **glPushMatrix();**  **glTranslatef(-1.0f, -0.7f, 0.0f);**  **glRotatef(\_angle1, 0.0f, 0.0f, 1.0f);**  **circle(0.5, 0.0, 0.0, 1.0, 1.0, 0.0);**  **glPopMatrix();**  **glPushMatrix();**  **glTranslatef(1.0f, -0.7f, 0.0f);**  **glRotatef(\_angle1, 0.0f, 0.0f, 1.0f);**  **circle(0.5, 0.0, 0.0, 1.0, 1.0, 0.0);**  **glPopMatrix();**  **}**  **void update1(int value) {**  **\_angle1+=2.0f;**  **if(\_angle1 > 360.0)**  **{**  **\_angle1-=360;**  **}**  **glutPostRedisplay(); //Notify GLUT that the display has changed**  **glutTimerFunc(20, update1, 0); //Notify GLUT to call update again in 25 milliseconds**  **}**  **void update(int value) {**  **\_move += .2;**  **if(\_move > 7)**  **{**  **\_move = -7.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **// Display function**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **car();**  **wheel();**  **glPopMatrix();**  **glutSwapBuffers(); // Swap the buffers to display the content**  **}**  **// Main function**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Moving Car");**  **gluOrtho2D(-6, 6, -6, 6);**  **glutTimerFunc(20, update1, 0);**  **glutTimerFunc(20, update, 0);**  **glutDisplayFunc(drawScene);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Design a windmill with rotating blades |
| **Graph Plot (Picture)-A graph paper with a drawing of a person's body  Description automatically generated** |
| **Code-#include <iostream>**  **#include <GL/gl.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **using namespace std;**  **float \_angle1 = 0.0f;**  **void windmealbody(){**  **glColor3f(1.0, 0.0, 0.0);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.6f, 0.0f);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(0.4f, -6.0f);**  **glVertex2f(-0.6f, -6.0f);**  **glEnd();**  **}**  **void pakha(){**  **glColor3f(1.0, 1.0, 1.0);**  **glLineWidth(5.0f);**  **glBegin(GL\_LINES);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(0.0f, 4.0f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(3.5f, -1.0f);**  **glEnd();**  **glBegin(GL\_LINES);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(-3.6f, -1.26f);**  **glEnd();**  **}**  **void update1(int value) {**  **\_angle1+=2.0f;**  **if(\_angle1 > 360.0)**  **{**  **\_angle1-=360;**  **}**  **glutPostRedisplay(); //Notify GLUT that the display has changed**  **glutTimerFunc(20, update1, 0); //Notify GLUT to call update again in 25 milliseconds**  **}**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glMatrixMode(GL\_MODELVIEW);**  **windmealbody();**  **glPushMatrix();**  **glRotatef(\_angle1, 0.0f, 0.0f, 1.0f);**  **pakha();**  **glPopMatrix();**  **glutSwapBuffers(); // Swap the buffers to display the content**  **}**  **// Main function**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("windMill");**  **gluOrtho2D(-6, 6, -6, 6);**  **glutTimerFunc(20, update1, 0);**  **glutDisplayFunc(drawScene);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  A computer screen shot of a red and black square  Description automatically generated |