**Lab Taks-5**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* 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 \_move = 0.0f;**  **float \_move1 = 0.0f;**  **void drawScene()**  **{**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(0, 0, 1);**  **glLoadIdentity(); //Reset the drawing perspective**  **glMatrixMode(GL\_MODELVIEW);**  **// First box**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(1.1f, 0.0f);**  **glVertex2f(0.5f, 0.0f);**  **glVertex2f(0.5f, 0.2f);**  **glVertex2f(1.1f, 0.2);**  **glEnd();**  **glPopMatrix();**  **// Second box**  **glPushMatrix();**  **glTranslatef(\_move1, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.5f, -0.2f);**  **glVertex2f(-1.1f, -0.2f);**  **glVertex2f(-1.1f, 0.0f);**  **glVertex2f(-0.5f, 0.0f);**  **glEnd();**  **glPopMatrix();**  **glutSwapBuffers();**  **}**  **void update(int value)**  **{**  **// First box**  **\_move -= 0.02;**  **if (\_move < -1.5)**  **{**  **\_move = 1.0;**  **}**  **// Second box**  **\_move1 += 0.02;**  **if (\_move1 > 1.5)**  **{**  **\_move1 = -1.0;**  **}**  **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)-** |

|  |
| --- |
| **Question-2**  Design a car which will have rotating wheels. |
| **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;**  **void wheel()**  **{**  **//one**  **glLoadIdentity(); //Reset the drawing perspective**  **glOrtho(-3,3,-3,3,-3,3);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(-0.75,0.51,0);**  **glRotatef(\_angle1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(160, 159, 160);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y );**  **}**  **glEnd();**  **glPopMatrix();**  **//two**  **glLoadIdentity(); //Reset the drawing perspective**  **glOrtho(-3,3,-3,3,-3,3);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(0.74,0.5,0);**  **glRotatef(\_angle1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(160, 159, 160);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y );**  **}**  **glEnd();**  **glPopMatrix();**  **}**  **void car()**  **{**  **glLoadIdentity(); //Reset the drawing perspective**  **glOrtho(-3,3,-3,3,-3,3);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **//body**  **glBegin(GL\_POLYGON);**  **glColor3ub(146, 11, 82);**  **glVertex2f(-1.4,0.5);**  **glVertex2f(-1.4,1);**  **glVertex2f(1.4,1);**  **glVertex2f(1.4,0.5);**  **glEnd();**  **//head**  **glBegin(GL\_POLYGON);**  **glColor3ub(146, 11, 82);**  **glVertex2f(-1,1);**  **glVertex2f(-0.5,1.5);**  **glVertex2f(0.5,1.5);**  **glVertex2f(1,1);**  **glEnd();**  **//one window**  **glBegin(GL\_POLYGON);**  **glColor3ub(13, 98, 135);**  **glVertex2f(-0.8,1);**  **glVertex2f(-0.4,1.4);**  **glVertex2f(-0.09,1.4);**  **glVertex2f(-0.09,1);**  **glEnd();**  **//two window**  **glBegin(GL\_POLYGON);**  **glColor3ub(13, 98, 135);**  **glVertex2f(0.09,1);**  **glVertex2f(0.09,1.4);**  **glVertex2f(0.4,1.4);**  **glVertex2f(0.8,1);**  **glEnd();**  **//one light**  **glBegin(GL\_POLYGON);**  **glColor3ub(232, 97, 20);**  **glVertex2f(-1.4,0.9);**  **glVertex2f(-1.4,1);**  **glVertex2f(-1.25,1);**  **glVertex2f(-1.25,0.9);**  **glEnd();**  **//two light**  **glBegin(GL\_POLYGON);**  **glColor3ub(232, 97, 20);**  **glVertex2f(1.25,0.9);**  **glVertex2f(1.25,1);**  **glVertex2f(1.4,1);**  **glVertex2f(1.4,0.9);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(68, 67, 67);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-0.75,y+0.51 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(68, 67, 67);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+0.74,y+0.5 );**  **}**  **glEnd();**  **glPopMatrix();**  **}**  **void drawScene()**  **{**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **car();**  **wheel();**  **glutSwapBuffers();**  **}**  **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**  **}**  **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, update1, 0); //Add a timer**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **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;**  **void wheel()**  **{**  **//one**  **glLoadIdentity(); //Reset the drawing perspective**  **glOrtho(-3,3,-3,3,-3,3);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **glTranslatef(-0.75,0.51,0);**  **glRotatef(\_angle1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(160, 159, 160);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y );**  **}**  **glEnd();**  **glPopMatrix();**  **//two**  **glLoadIdentity(); //Reset the drawing perspective**  **glOrtho(-3,3,-3,3,-3,3);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **glTranslatef(0.74,0.5,0);**  **glRotatef(\_angle1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(160, 159, 160);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y );**  **}**  **glEnd();**  **glPopMatrix();**  **}**  **void car()**  **{**  **glLoadIdentity(); //Reset the drawing perspective**  **glOrtho(-3,3,-3,3,-3,3);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **//body**  **glBegin(GL\_POLYGON);**  **glColor3ub(146, 11, 82);**  **glVertex2f(-1.4,0.5);**  **glVertex2f(-1.4,1);**  **glVertex2f(1.4,1);**  **glVertex2f(1.4,0.5);**  **glEnd();**  **//head**  **glBegin(GL\_POLYGON);**  **glColor3ub(146, 11, 82);**  **glVertex2f(-1,1);**  **glVertex2f(-0.5,1.5);**  **glVertex2f(0.5,1.5);**  **glVertex2f(1,1);**  **glEnd();**  **//one window**  **glBegin(GL\_POLYGON);**  **glColor3ub(13, 98, 135);**  **glVertex2f(-0.8,1);**  **glVertex2f(-0.4,1.4);**  **glVertex2f(-0.09,1.4);**  **glVertex2f(-0.09,1);**  **glEnd();**  **//two window**  **glBegin(GL\_POLYGON);**  **glColor3ub(13, 98, 135);**  **glVertex2f(0.09,1);**  **glVertex2f(0.09,1.4);**  **glVertex2f(0.4,1.4);**  **glVertex2f(0.8,1);**  **glEnd();**  **//one light**  **glBegin(GL\_POLYGON);**  **glColor3ub(232, 97, 20);**  **glVertex2f(-1.4,0.9);**  **glVertex2f(-1.4,1);**  **glVertex2f(-1.25,1);**  **glVertex2f(-1.25,0.9);**  **glEnd();**  **//two light**  **glBegin(GL\_POLYGON);**  **glColor3ub(232, 97, 20);**  **glVertex2f(1.25,0.9);**  **glVertex2f(1.25,1);**  **glVertex2f(1.4,1);**  **glVertex2f(1.4,0.9);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(68, 67, 67);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x-0.75,y+0.51 );**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(68, 67, 67);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.25;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+0.74,y+0.5 );**  **}**  **glEnd();**  **glPopMatrix();**  **}**  **void drawScene()**  **{**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **car();**  **wheel();**  **glutSwapBuffers();**  **}**  **void update(int value)**  **{**  **\_move += .02;**  **if(\_move > 1.3)**  **{**  **\_move = -1.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **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**  **}**  **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**  **glutTimerFunc(20, update1, 0); //Add a timer**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Design a windmill with rotating blades |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include <windows.h>**  **#include<math.h>**  **using namespace std;**  **float \_angle1 = 0.0f;**  **void blade()**  **{**  **glColor3ub(248, 205, 113);**  **glLoadIdentity();//Reset the drawing perspective**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glRotatef(\_angle1,0,0,5);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.02,0.068);**  **glVertex2f(-0.4,0.8);**  **glVertex2f(-0.2,1);**  **glVertex2f(0.02,0.068);**  **glEnd();**  **glPopMatrix();**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glRotatef(\_angle1,0,0,5);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-0.8,-0.4);**  **glVertex2f(-1,-0.2);**  **glVertex2f(-0.06,0.04);**  **glVertex2f(-0.07,0);**  **glEnd();**  **glPopMatrix();**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glRotatef(\_angle1,0,0,5);**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.068,-0.02);**  **glVertex2f(0.068,0.02);**  **glVertex2f(0.8,-0.2);**  **glVertex2f(1,-0.4);**  **glEnd();**  **glPopMatrix();**  **glLoadIdentity(); //Reset the drawing perspective**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glRotatef(\_angle1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(235, 217, 179);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.07;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y );**  **}**  **glEnd();**  **glPopMatrix();**  **glBegin(GL\_POLYGON);**  **glColor3ub(121, 116, 107);**  **glVertex2f(-0.2,-1.2);**  **glVertex2f(-0.05,-0.05);**  **glVertex2f(0.05,-0.05);**  **glVertex2f(0.2,-1.2);**  **glEnd();**  **}**  **void scenery()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(191, 116, 14);**  **glVertex2f(-1.2,-1.4);**  **glVertex2f(-1.2,-0.6);**  **glVertex2f(1.2,-0.7);**  **glVertex2f(1.2,-1.4);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(86, 155, 21);**  **glVertex2f(-1.2,-0.6);**  **glVertex2f(-1.2,0.2);**  **glVertex2f(1.2,0.1);**  **glVertex2f(1.2,-0.7);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(35, 137, 198);**  **glVertex2f(-1.2,0.2);**  **glVertex2f(-1.2,1);**  **glVertex2f(1.2,1);**  **glVertex2f(1.2,0.1);**  **glEnd();**  **}**  **void drawScene()**  **{**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **scenery();**  **blade();**  **glutSwapBuffers();**  **}**  **void update(int value)**  **{**  **\_angle1+=2.0f;**  **if(\_angle1 > 360.0)**  **{**  **\_angle1-=360;**  **}**  **glutPostRedisplay(); //Notify GLUT that the display has change**  **glutTimerFunc(20, update, 0); //Notify GLUT to call update again in 25 milliseconds**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Transformation");**  **glutDisplayFunc(drawScene);**  **glutTimerFunc(20, update, 0); //Add a timer**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |