**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)-**  **[Not needed]** |
| **Code-**  #include <iostream>  #include<GL/gl.h>  #include <GL/glut.h>  using namespace std;  float movement1=-1.5;  float movement2=1.5;  void square1()  {  glColor3d(0,0,255);  glLoadIdentity();  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef(movement1, 0.0f, 0.0f);  glBegin(GL\_QUADS);  glVertex2f(0.1f, 0.0f);  glVertex2f(0.3f, 0.0f);  glVertex2f(0.3f, 0.2f);  glVertex2f(0.1f, 0.2);  glEnd();  glPopMatrix();  }  void square2()  {  glColor3d(1,0,0);  glLoadIdentity();  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef(movement2, 0.0f, 0.0f);  glBegin(GL\_QUADS);  glVertex2f(0.1f, 0.0f);  glVertex2f(0.3f, 0.0f);  glVertex2f(0.3f, 0.2f);  glVertex2f(0.1f, 0.2);  glEnd();  glPopMatrix();  }  void update1(int value)  {  movement1 += .01;  if(movement1 > 1.5)  {  movement1 = -1.5;  }  glutPostRedisplay();  glutTimerFunc(10, update1, 0);  }  void update2(int value)  {  movement2 -= .01;  if(movement2 < -1.5)  {  movement2 = 1.5;  }  glutPostRedisplay();  glutTimerFunc(10, update2, 0);  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  square1();  square2();  glutSwapBuffers();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutCreateWindow("Question 1");  glutDisplayFunc(display);  gluOrtho2D(-2,2,-2,2);  glutTimerFunc(10,update1,0);  glutTimerFunc(10,update2,0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-2**  Design a car which will have rotating wheels. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include <windows.h>**  **#include <math.h>**  **using namespace std;**  **float rotation1 = 0.1f;**  **float rotation2 = 0.1f;**  **void Car()**  **{**  **glColor3ub(0,0,255);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.4,0.5f);**  **glVertex2f(-0.6f,0.1f);**  **glVertex2f(0.2f,0.1f);**  **glVertex2f(0.0f,0.5f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.9f,0.15f);**  **glVertex2f(-0.9f,-0.1f);**  **glVertex2f(0.5f,-0.1f);**  **glVertex2f(0.5f,0.15f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.9f,-0.1f);**  **glVertex2f(-0.9f,-0.15f);**  **glVertex2f(-0.7f,-0.15f);**  **glVertex2f(-0.7f,-0.1f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(0.3f,-0.1f);**  **glVertex2f(0.3f,-0.15f);**  **glVertex2f(0.5f,-0.15f);**  **glVertex2f(0.5f,-0.1f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.9f,0.1f);**  **glVertex2f(-0.9f,0.15f);**  **glVertex2f(-0.5f,0.15f);**  **glVertex2f(-0.5f,0.1f);**  **glEnd();**  **}**  **void FrontTier()**  **{**  **glPushMatrix();**  **glTranslatef(-0.6f,-0.2f,0.0f);**  **glRotatef(rotation1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.15;**  **float A=(i\*2\*pi)/200;**  **float r=0.09;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.15;**  **float A=(i\*2\*pi)/200;**  **float r=0.08;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.15;**  **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();**  **}**  **void BackTier()**  **{**  **glPushMatrix();**  **glTranslatef(0.2f,-0.2f,0.0f);**  **glRotatef(rotation2, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.15;**  **float A=(i\*2\*pi)/200;**  **float r=0.09;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.15;**  **float A=(i\*2\*pi)/200;**  **float r=0.08;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.15;**  **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();**  **}**  **void Display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **Car();**  **FrontTier();**  **BackTier();**  **glutSwapBuffers();**  **}**  **void update1(int value)**  **{**  **rotation1+=1.0f;**  **if(rotation1 > 360.0)**  **{**  **rotation1-=360;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update1, 0);**  **}**  **void update2(int value)**  **{**  **rotation2+=1.0f;**  **if(rotation2 > 360.0)**  **{**  **rotation2-=360;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update2, 0);**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(400,400);**  **glutCreateWindow("Question no 2");**  **glutDisplayFunc(Display);**  **glutTimerFunc(10, update1, 0);**  **glutTimerFunc(10,update2,0);**  **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)-**  **[Not needed]** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include <windows.h>**  **#include <math.h>**  **using namespace std;**  **float movement = -1.5;**  **float rotaion1 = 0.0f;**  **float rotaion2 = 0.0f;**  **void Car()**  **{**  **glColor3ub(0,0,255);**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.4,0.5f);**  **glVertex2f(-0.6f,0.1f);**  **glVertex2f(0.2f,0.1f);**  **glVertex2f(0.0f,0.5f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.9f,0.15f);**  **glVertex2f(-0.9f,-0.1f);**  **glVertex2f(0.5f,-0.1f);**  **glVertex2f(0.5f,0.15f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.9f,-0.1f);**  **glVertex2f(-0.9f,-0.15f);**  **glVertex2f(-0.7f,-0.15f);**  **glVertex2f(-0.7f,-0.1f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(0.3f,-0.1f);**  **glVertex2f(0.3f,-0.15f);**  **glVertex2f(0.5f,-0.15f);**  **glVertex2f(0.5f,-0.1f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.9f,0.1f);**  **glVertex2f(-0.9f,0.15f);**  **glVertex2f(-0.5f,0.15f);**  **glVertex2f(-0.5f,0.1f);**  **glEnd();**  **}**  **void FrontTier()**  **{**  **glPushMatrix();**  **glTranslatef(-0.6f,-0.2f,0.0f);**  **glRotatef(rotaion1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.09;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.08;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **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();**  **}**  **void BackTier()**  **{**  **glPushMatrix();**  **glTranslatef(0.2f,-0.2f,0.0f);**  **glRotatef(rotaion2, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.09;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.08;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **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();**  **}**  **void Display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(movement,0.0f,0.0f);**  **Car();**  **FrontTier();**  **BackTier();**  **glPopMatrix();**  **glutSwapBuffers();**  **}**  **void edit1(int value)**  **{**  **rotaion1+=1.0f;**  **if(rotaion1 > 360.0)**  **{**  **rotaion1-=360;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, edit1, 0);**  **}**  **void edit2(int value)**  **{**  **rotaion2+=1.0f;**  **if(rotaion2 > 360.0)**  **{**  **rotaion2-=360;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, edit2, 0);**  **}**  **void movingtier1(int value)**  **{**  **movement += .01;**  **if(movement > 1.5)**  **{**  **movement = -1.5;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, movingtier1, 0);**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(400,400);**  **glutCreateWindow("Question no 3");**  **glutDisplayFunc(Display);**  **glutTimerFunc(20, edit1, 0);**  **glutTimerFunc(20,edit2,0);**  **glutTimerFunc(20,movingtier1,0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Design a windmill with rotating blades |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include <windows.h>**  **#include <math.h>**  **using namespace std;**  **float rotation1 = 0.0f;**  **float rotation2 = 0.0f;**  **void Fan()**  **{**  **glPushMatrix();**  **glTranslatef(0.0f,0.0f,0.0f);**  **glRotatef(rotation1, 0.0f, 0.0f,1.0f);**  **glBegin(GL\_LINES);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **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();**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2f(-0.02,0.4);**  **glVertex2f(-0.02,-0.4);**  **glVertex2f(0.03,-0.4);**  **glVertex2f(0.03,0.4);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2f(-0.4,0.02);**  **glVertex2f(-0.4,-0.02);**  **glVertex2f(0.4,-0.03);**  **glVertex2f(0.4,0.03);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(0,0,0);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=0.05;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x,y);**  **}**  **glEnd();**  **glPopMatrix();**  **}**  **void Display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **glBegin(GL\_QUADS);**  **glColor3ub(139,0,0);**  **glVertex2f(-0.1,0.25);**  **glVertex2f(-0.2,-0.5);**  **glVertex2f(0.2,-0.5);**  **glVertex2f(0.1,0.25);**  **glEnd();**  **Fan();**  **glutSwapBuffers();**  **}**  **void movement1(int value)**  **{**  **rotation1+=1.0f;**  **if(rotation1 > 360.0)**  **{**  **rotation1-=360;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, movement1, 0);**  **}**  **void Movement2(int value) {**  **rotation2+=1.0f;**  **if(rotation2 > 360.0)**  **{**  **rotation2-=360;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, Movement2, 0);**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(400, 400);**  **glutCreateWindow("Question no 4");**  **glutDisplayFunc(Display);**  **glutTimerFunc(20, movement1, 0);**  **glutTimerFunc(20,Movement2,0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |