**Lab Taks-6**

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 include resources for all the section in the table

|  |
| --- |
| **Question-**  Develop an animation that will change the background color of the window after 20ms. Use at least two different colors. |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **float move\_1 = 0.0f;**  **float move\_2 = 0.0f;**  **void display();**  **void top\_box()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **//glTranslatef(move\_1, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 20.0f);**  **glVertex2f(10.0f, 20.0f);**  **glVertex2f(10.0f, 30.0f);**  **glVertex2f(0.0f, 30.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_1(int value) {**  **move\_1 += 0.49999; //assume as moving speed**  **if(move\_1>50) //boundary for the positive x axis for last part of the box**  **{**  **move\_1 = -61;//reappear the box**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_1, 0);**  **}**  **void bottom\_box()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **//glTranslatef(move\_2, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(-10.0f, 0.0f);**  **glVertex2f(-10.0f, 10.0f);**  **glVertex2f(0.0f, 10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_2(int value) {**  **move\_2 -= 0.49999; //assume as moving speed**  **if(move\_2 < -50) //boundary for the negative x axis for last part of the box**  **{**  **move\_2 = 61;//reappear the box**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_2, 0);**  **}**  **void displayback(int x)//again start from white**  **{**  **glutDisplayFunc(display);**  **}**  **void display\_2()**  **{**  **glClearColor(0.0f, 1.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **top\_box();**  **bottom\_box();**  **glutTimerFunc(500,displayback,0);**  **glutSwapBuffers();**  **}**  **void Scene\_2(int x)//third call //green**  **{**  **glutDisplayFunc(display\_2);**  **}**  **void display\_1()**  **{**  **glClearColor(1.0f, 0.0f, 0.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **top\_box();**  **bottom\_box();**  **glutTimerFunc(500,Scene\_2,0);**  **glutSwapBuffers();**  **}**  **void Scene\_1(int x) //second call //red**  **{**  **glutDisplayFunc(display\_1);**  **}**  **void display() {**  **glClearColor(1.0f, 1.0f,1.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **top\_box();**  **bottom\_box();**  **glutTimerFunc(500,Scene\_1,0);**  **glutSwapBuffers();**  **}**  **void Scene()//first call //white**  **{**  **glutDisplayFunc(display);**  **}**  **void identity\_axis()**  **{**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-50,50,-50,50);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Background Color Change");**  **glutDisplayFunc(Scene);**  **identity\_axis();**  **glutTimerFunc(20, update\_1, 0); //Add a timer**  **//glutTimerFunc(20, update\_2, 0); //Add a timer**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop an animation that will call four objects separately, each after 20 ms. |
| **Code-**  **#include<cstdio>**  **#include <GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **void display();**  **void scene();**  **GLfloat position = 0.0f;**  **GLfloat position\_1 = 0.0f;**  **GLfloat position\_2 = 0.0f;**  **GLfloat position\_3 = 0.0f;**  **GLfloat speed = 0.1f;**  **void update\_3(int value) {**  **if(position\_3 >10)**  **{**  **position\_3 = -4.0f;**  **}**  **position\_3 += speed;**  **glutPostRedisplay();**  **glutTimerFunc(20,update\_3,0);**  **}**  **void update\_2(int value) {**  **if(position\_2 >10)**  **{**  **position\_2 = -4.0f;**  **}**  **position\_2 += speed;**  **glutPostRedisplay();**  **glutTimerFunc(20,update\_2,0);**  **}**  **void update\_1(int value) {**  **if(position\_1 <-8)**  **{**  **position\_1 = 2.0f;**  **}**  **position\_1 -= speed;**  **glutPostRedisplay();**  **glutTimerFunc(20,update\_1,0);**  **}**  **void update(int value) {**  **if(position<-8)**  **{**  **position = 2.0f;**  **}**  **position -= speed;**  **glutPostRedisplay();**  **glutTimerFunc(20,update,0);**  **}**  **void displayback(int val)**  **{**  **glutDisplayFunc(display);**  **}**  **void display\_3()**  **{**  **//4th box**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glPushMatrix();**  **//glTranslatef(position\_3,0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glColor3f(1.0f, 1.0f, 0.0f);**  **glVertex2f(6.0f, 6.0f);**  **glVertex2f(8.0f, 6.0f);**  **glVertex2f(8.0f, 8.0f);**  **glVertex2f(6.0f, 8.0f);**  **glEnd();**  **glPopMatrix();**  **glutTimerFunc(1500,displayback,0);**  **glFlush();**  **}**  **void scene\_3(int val) {**  **glutDisplayFunc(display\_3);**  **}**  **void display\_2()**  **{**  **//3rd box**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glPushMatrix();**  **//glTranslatef(position\_2,0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glColor3f(0.0f, 0.0f, 1.0f);**  **glVertex2f(2.0f, 6.0f);**  **glVertex2f(4.0f, 6.0f);**  **glVertex2f(4.0f, 8.0f);**  **glVertex2f(2.0f, 8.0f);**  **glEnd();**  **glPopMatrix();**  **glutTimerFunc(1500,scene\_3,0);**  **glFlush();**  **}**  **void scene\_2(int val)**  **{**  **glutDisplayFunc(display\_2);**  **}**  **void display\_1()**  **{**  **//2nd box**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glPushMatrix();**  **//glTranslatef(position\_1,0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glColor3f(0.0f, 1.0f, 0.0f);**  **glVertex2f(6.0f, 2.0f);**  **glVertex2f(8.0f, 2.0f);**  **glVertex2f(8.0f, 4.0f);**  **glVertex2f(6.0f, 4.0f);**  **glEnd();**  **glPopMatrix();**  **glutTimerFunc(1500,scene\_2,0);**  **glFlush();**  **}**  **void scene\_1(int val)**  **{**  **glutDisplayFunc(display\_1);**  **}**  **void display()**  **{**  **//1st box**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glPushMatrix();**  **//glTranslatef(position,0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glColor3f(1.0f, 0.0f, 0.0f);**  **glVertex2f(2.0f, 2.0f);**  **glVertex2f(4.0f, 2.0f);**  **glVertex2f(4.0f, 4.0f);**  **glVertex2f(2.0f, 4.0f);**  **glEnd();**  **glPopMatrix();**  **glutTimerFunc(1500,scene\_1,0);**  **glFlush();**  **}**  **void scene()**  **{**  **glutDisplayFunc(display);**  **}**  **void identity\_axis()**  **{**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f);**  **glLoadIdentity();**  **gluOrtho2D(0,10,0,10);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitWindowSize(320, 320);**  **glutInitWindowPosition(50, 50);**  **glutCreateWindow("Four Object in Four Display Calling");**  **glutDisplayFunc(scene);**  **identity\_axis();**  **glutTimerFunc(20, update, 0);**  **//glutTimerFunc(20, update\_1, 0);**  **//glutTimerFunc(20, update\_2, 0);**  **//glutTimerFunc(20, update\_3, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |
| **Question-**  Develop a code that will have four different objects (keep it simple). The objects will move to the left, right, up and down in a loop. |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **float move\_1 = 0.0f;**  **void top\_box()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(move\_1, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 20.0f);**  **glVertex2f(10.0f, 20.0f);**  **glVertex2f(10.0f, 30.0f);**  **glVertex2f(0.0f, 30.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_1(int value) {**  **move\_1 += 0.49999; //assume as moving speed**  **if(move\_1>50) //boundary for the positive x axis for last part of the box**  **{**  **move\_1 = -61;//reappear the box**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_1, 0);**  **}**  **float move\_2 = 0.0f;**  **void bottom\_box()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(move\_2, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(-10.0f, 0.0f);**  **glVertex2f(-10.0f, 10.0f);**  **glVertex2f(0.0f, 10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_2(int value) {**  **move\_2 -= 0.49999; //assume as moving speed**  **if(move\_2 < -50) //boundary for the negative x axis for last part of the box**  **{**  **move\_2 = 61;//reappear the box**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_2, 0);**  **}**  **float move\_3 = 0.0f;**  **void left\_box()**  **{**  **glColor3d(1,0,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(0.0f, move\_3, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-40.0f, 0.0f);**  **glVertex2f(-30.0f, 0.0f);**  **glVertex2f(-30.0f, 10.0f);**  **glVertex2f(-40.0f, 10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_3(int value) {**  **move\_3 += 0.49999; //assume as moving speed**  **if(move\_3 > 50) //boundary for the negative x axis for last part of the box**  **{**  **move\_3 = -60;//reappear the box**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_3, 0);**  **}**  **float move\_4 = 0.0f;**  **void right\_box()**  **{**  **glColor3d(1,0,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(0.0f, move\_4, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(30.0f, 0.0f);**  **glVertex2f(40.0f, 0.0f);**  **glVertex2f(40.0f, -10.0f);**  **glVertex2f(30.0f, -10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_4(int value) {**  **move\_4 -= 0.49999; //assume as moving speed**  **if(move\_4 <-50) //boundary for the negative x axis for last part of the box**  **{**  **move\_4 = 60;//reappear the box**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_4, 0);**  **}**  **void identity\_axis()**  **{**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-50,50,-50,50);**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **top\_box();**  **bottom\_box();**  **left\_box();**  **right\_box();**  **glutSwapBuffers();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Box Move Animation");**  **glutDisplayFunc(display);**  **identity\_axis();**  **glutTimerFunc(20, update\_1, 0); //Add a timer**  **glutTimerFunc(20, update\_2, 0); //Add a timer**  **glutTimerFunc(20, update\_3, 0); //Add a timer**  **glutTimerFunc(20, update\_4, 0); //Add a timer**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop a code that will have four different objects (keep it simple). Four different keys will be dedicated each objects. The objects will move to the left, right, up and down in a loop as the keys are pressed individually. |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **float speed = 0.49999;**  **float move\_1 = 0.0f;**  **void top\_box()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(move\_1, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 20.0f);**  **glVertex2f(10.0f, 20.0f);**  **glVertex2f(10.0f, 30.0f);**  **glVertex2f(0.0f, 30.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_1(int value) {**  **if(move\_1>50) //boundary for the positive x axis for last part of the box**  **{**  **move\_1 = -61;//reappear the box**  **}**  **else if (move\_1<-61)**  **{**  **move\_1=50;**  **}**  **move\_1 += speed; //assume as moving speed**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_1, 0);**  **}**  **float move\_2 = 0.0f;**  **void bottom\_box()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(move\_2, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 0.0f);**  **glVertex2f(-10.0f, 0.0f);**  **glVertex2f(-10.0f, 10.0f);**  **glVertex2f(0.0f, 10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_2(int value) {**  **if(move\_2 < -50) //boundary for the negative x axis for last part of the box**  **{**  **move\_2 = 61;//reappear the box**  **}**  **else if(move\_2>61)**  **{**  **move\_2=-50;**  **}**  **move\_2 -= speed; //assume as moving speed**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_2, 0);**  **}**  **float move\_3 = 0.0f;**  **void left\_box()**  **{**  **glColor3d(1,0,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(0.0f, move\_3, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(-40.0f, 0.0f);**  **glVertex2f(-30.0f, 0.0f);**  **glVertex2f(-30.0f, 10.0f);**  **glVertex2f(-40.0f, 10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_3(int value) {**  **if(move\_3 > 50) //boundary for the negative x axis for last part of the box**  **{**  **move\_3 = -60;//reappear the box**  **}**  **else if(move\_3<-60)**  **{**  **move\_3=50;**  **}**  **move\_3 += speed; //assume as moving speed**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_3, 0);**  **}**  **float move\_4 = 0.0f;**  **void right\_box()**  **{**  **glColor3d(1,0,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(0.0f, move\_4, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(30.0f, 0.0f);**  **glVertex2f(40.0f, 0.0f);**  **glVertex2f(40.0f, -10.0f);**  **glVertex2f(30.0f, -10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_4(int value) {**  **if(move\_4 <-50) //boundary for the negative x axis for last part of the box**  **{**  **move\_4 = 60;//reappear the box**  **}**  **else if(move\_4>60)**  **{**  **move\_4=-50;**  **}**  **move\_4 -=speed; //assume as moving speed**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_4, 0);**  **}**  **void handleKeypress(unsigned char key, int x, int y)**  **{**  **switch (key) {**  **case 'r'://right**  **glutTimerFunc(20, update\_1, 0);**  **glutPostRedisplay();**  **break;**  **case 'l': //left**  **glutTimerFunc(20, update\_2, 0);**  **glutPostRedisplay();**  **break;**  **case 'u': //up**  **glutTimerFunc(20, update\_3, 0);**  **glutPostRedisplay();**  **break;**  **case 'd': //down**  **glutTimerFunc(20, update\_4, 0);**  **glutPostRedisplay();**  **break;**  **case 's': //stop**  **speed=0.0;**  **glutPostRedisplay();**  **break;**  **case 'a': //again start**  **speed=0.49999;**  **glutPostRedisplay();**  **break;**  **}**  **}**  **void identity\_axis()**  **{**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-50,50,-50,50);**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **top\_box();**  **bottom\_box();**  **left\_box();**  **right\_box();**  **glutSwapBuffers();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Keyboard Interaction");**  **glutDisplayFunc(display);**  **identity\_axis();**  **glutKeyboardFunc(handleKeypress);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop a code that will have four different objects (keep it simple). Two of the objects will move to the right as the right click is made on the mouse and two of the objects will move to the left as the left key is pressed on the mouse. |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **float move\_right = 0.0f;**  **float speed =0.49999;**  **void top\_box\_1()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(move\_right, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 25.0f);**  **glVertex2f(10.0f, 25.0f);**  **glVertex2f(10.0f, 35.0f);**  **glVertex2f(0.0f, 35.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void top\_box\_2()**  **{**  **glColor3d(1,1,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(move\_right, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.0f, 10.0f);**  **glVertex2f(10.0f, 10.0f);**  **glVertex2f(10.0f, 20.0f);**  **glVertex2f(0.0f, 20.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_right(int value) {**  **if(move\_right>50) //boundary for the positive x axis for last part of the box**  **{**  **move\_right = -60;//reappear the box**  **}**  **else if (move\_right<-60) //boundary for the positive x axis for last part of the box**  **{**  **move\_right = 50;//reappear the box**  **}**  **move\_right += speed; //assume as moving speed**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_right, 0);**  **}**  **float move\_left = 0.0f;**  **void bottom\_box\_1()**  **{**  **glColor3d(1,0,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef( move\_left,0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(20.0f, -20.0f);**  **glVertex2f(30.0f, -20.0f);**  **glVertex2f(30.0f,-10.0f);**  **glVertex2f(20.0f, -10.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void bottom\_box\_2()**  **{**  **glColor3d(1,0,0);**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef( move\_left,0.0f,0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(20.0f, -35.0f);**  **glVertex2f(30.0f, -35.0f);**  **glVertex2f(30.0f, -25.0f);**  **glVertex2f(20.0f, -25.0);**  **glEnd();**  **glPopMatrix();**  **}**  **void update\_left(int value) {**  **if(move\_left < -80) //boundary for the negative x axis for last part of the box**  **{**  **move\_left = 30;//reappear the box**  **}**  **else if (move\_left >30)**  **{**  **move\_left = -80;//reappear the box**  **}**  **move\_left -= speed; //assume as moving speed**  **glutPostRedisplay();**  **glutTimerFunc(20, update\_left, 0);**  **}**  **void handleMouse(int button, int state, int x, int y) {**  **if (button == GLUT\_LEFT\_BUTTON)**  **{ glutTimerFunc(20, update\_left, 0);}**  **if (button == GLUT\_RIGHT\_BUTTON)**  **{glutTimerFunc(20, update\_right, 0); }**  **glutPostRedisplay();}**  **void handleKeypress(unsigned char key, int x, int y) {**  **switch (key) {**  **case 's': //stop**  **speed = 0.0f;**  **break;**  **case 'r': //restart**  **speed = 0.49999f;**  **break;**  **glutPostRedisplay();**  **}}**  **void identity\_axis()**  **{**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-50,50,-50,50);**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **top\_box\_1();**  **top\_box\_2();**  **bottom\_box\_1();**  **bottom\_box\_2();**  **glutSwapBuffers();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Mouse Interaction");**  **glutDisplayFunc(display);**  **identity\_axis();**  **glutKeyboardFunc(handleKeypress);**  **glutMouseFunc(handleMouse);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |