**Lab Report. 05**

**Subject: Computer Graphics Lab**

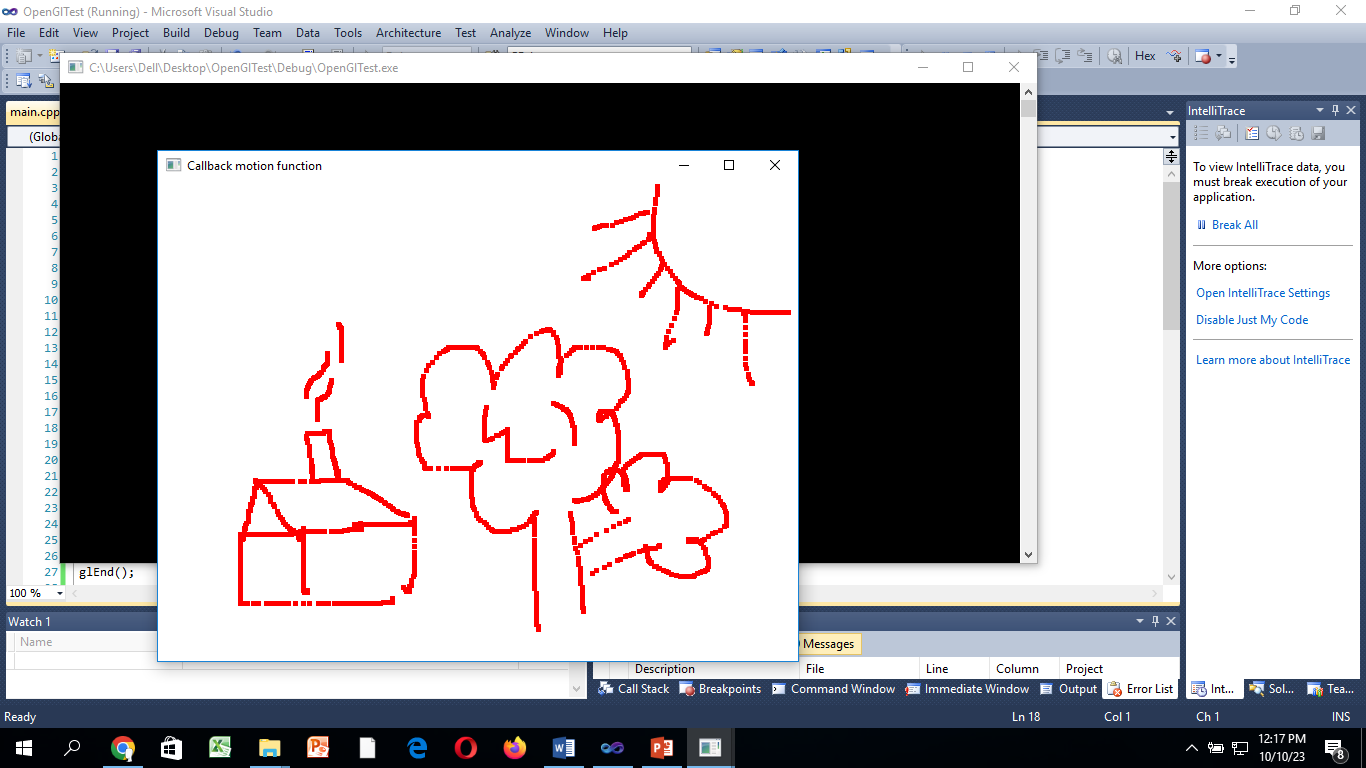
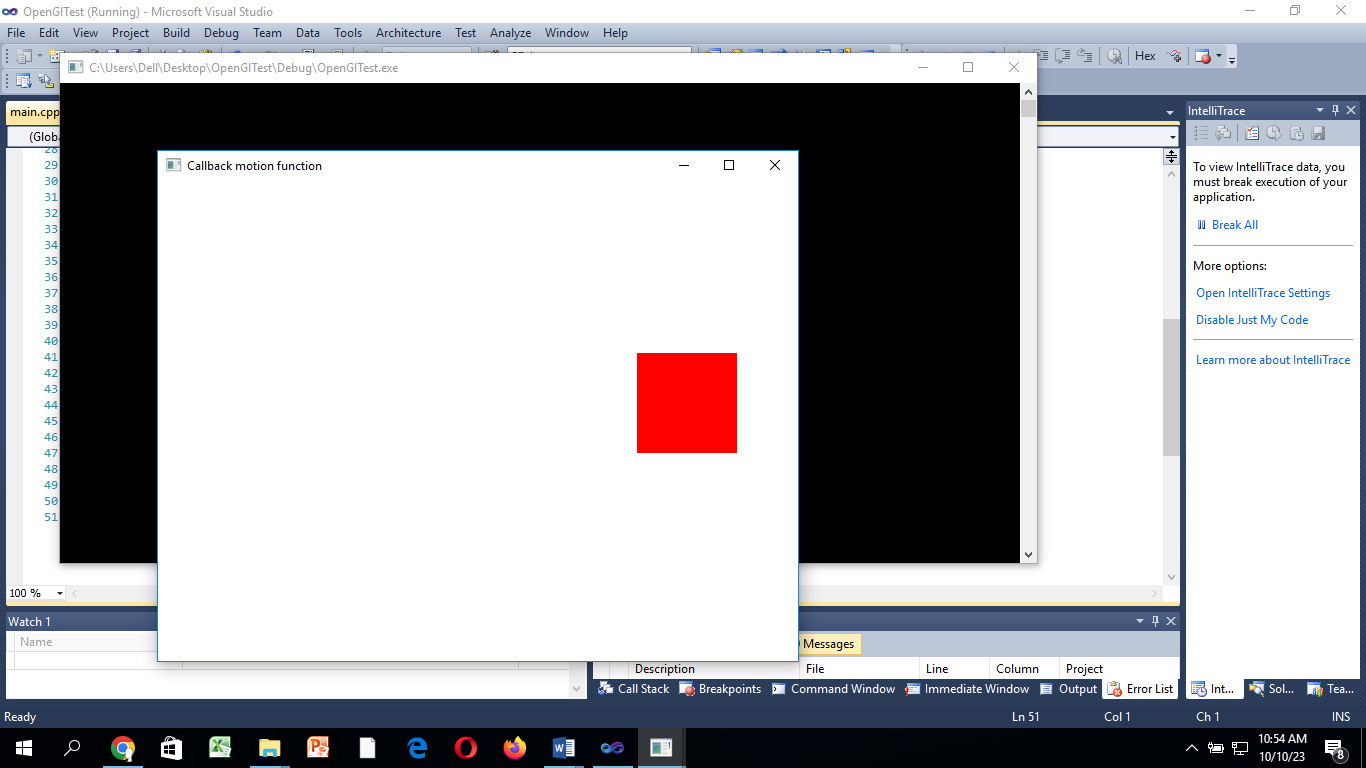
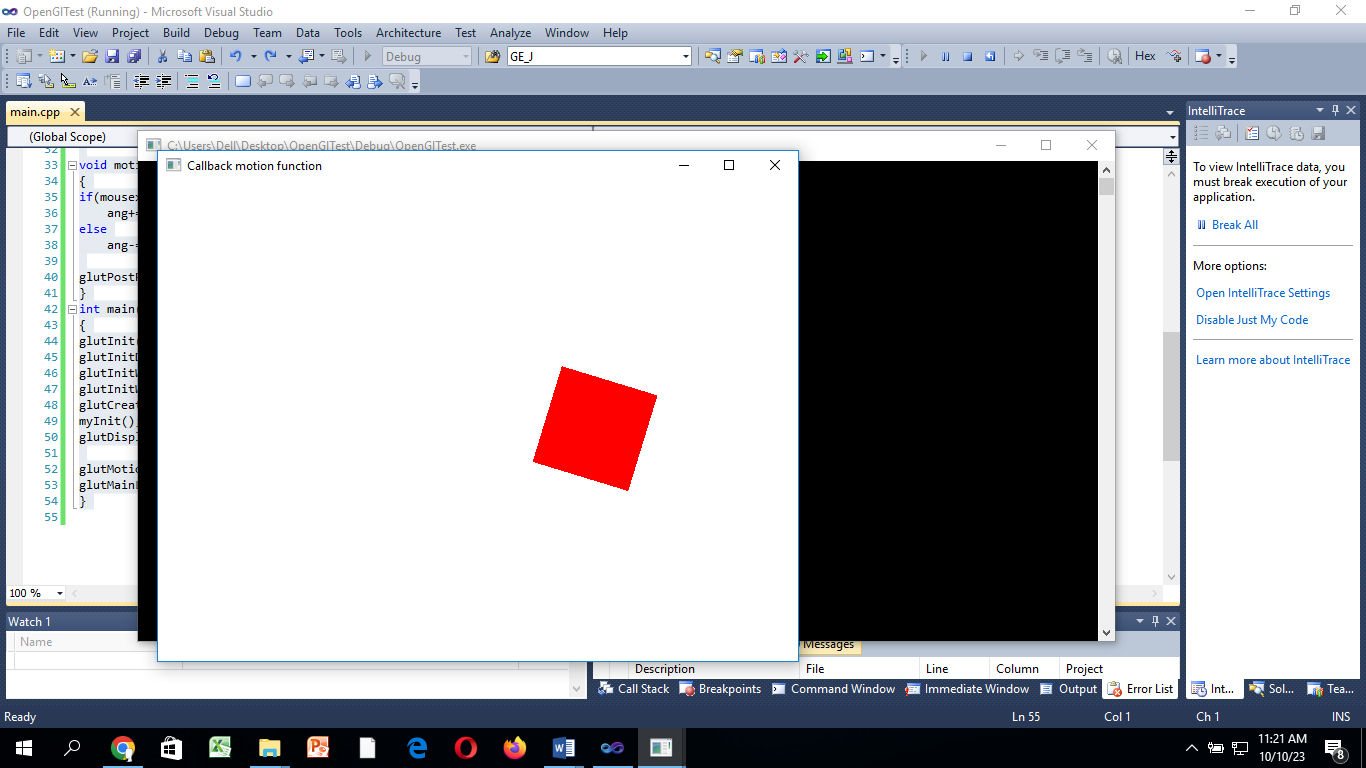


Submitted to: Dr. Junaid

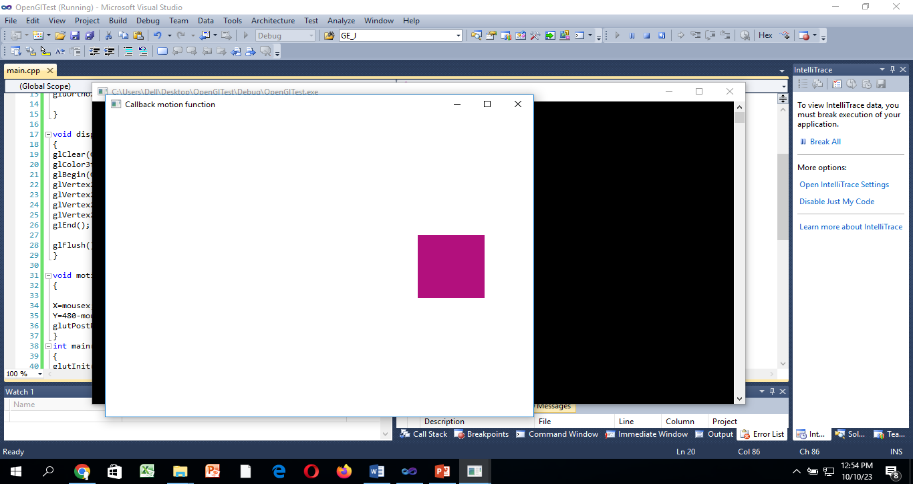
Submitted by: Nayab Ashraf (RC-299)

**Date of Submission: 09 October, 2024**

National University of Modern Languages, Department of Computer Science, Rawalpindi Campus.



**Fig a**  **Fig b** **Fig c**



**Fig d**

**Task 01:**

**Implement Quiz-2 of theory.**

#include<stdio.h>

#include<iostream>

#include<GL\glut.h>

#include<math.h>

using namespace std;

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 1.0);//Defines the color of the background

glMatrixMode(GL\_MODELVIEW);//takes either GL\_MODELVIEW or GL\_PROJECTION as input

glLoadIdentity();//Loads the identity matrix into the chosen matrix i.e. resets the MODEL\_VIEW matrix to identity

gluOrtho2D(0.0, 500, 0.0, 500);//Describes a unit system Right, Left, Bottom, Top

}

void displayCircle()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

GLfloat centerx, centery, radius, n;

GLfloat const PI = 3.14;

//1st circle

centerx = 100; centery = 100; radius = 50;

glColor3f(1.0f, 0.0f, 0.0f);

n = 360;

glPointSize(2);

glBegin(GL\_POINTS);

for (GLfloat i = 0; i <n; i++)

{

float theta = i\*PI / 180.0;

glVertex2f(radius\*cosf(theta) + centerx, radius\*sinf(theta) + centery);

}

glEnd();

//2nd circle

centerx = 205; centery = 100; radius = 50;

glColor3f(1.0f, 1.0f, 0.0f);

n = 360;

glPointSize(2);

glBegin(GL\_POINTS);

for (GLfloat i = 0; i <n; i++)

{

float theta = i\*PI / 180.0;

glVertex2f(radius\*cosf(theta) + centerx, radius\*sinf(theta) + centery);

}

glEnd();

//3rd circle

centerx = 150; centery = 190; radius = 50;

glColor3f(0.0f, 1.0f, 0.0f);

n = 360;

glPointSize(2);

glBegin(GL\_POINTS);

for (GLfloat i = 0; i <n; i++)

{

float theta = i\*PI / 180.0;

glVertex2f(radius\*cosf(theta) + centerx, radius\*sinf(theta) + centery);

}

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(500, 500);

glutInitWindowPosition(50, 50);

glutCreateWindow("To draw 3 circles");

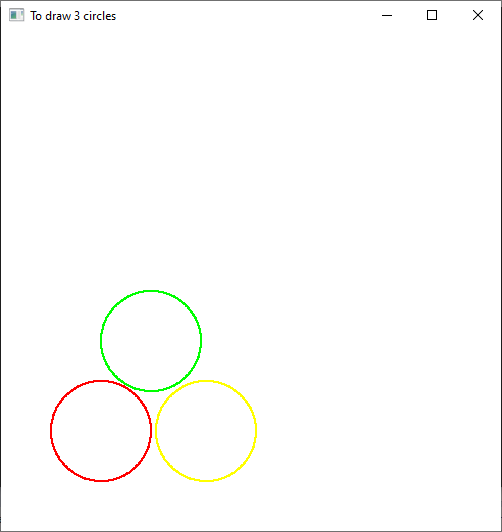
myInit();

glutDisplayFunc(displayCircle);

glutMainLoop();

return 0;

}

****

**Task 02:**

**Update the code so that it can make complete scene (fig a).**

#include<Windows.h>

#include<GL\glut.h>

#include<math.h>

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(1.0f, 0.0f, 0.0f);

glPointSize(5.0);

glMatrixMode(GL\_MODELVIEW);

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void display()

{

glFlush();

}

void motionfunction(int mousex, int mousey)

{

glBegin(GL\_POINTS);

glVertex2i(mousex, 480 - mousey);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(640, 480);

glutInitWindowPosition(150, 150);

glutCreateWindow("Callback motion function");

myInit();

glutDisplayFunc(display);

glutMotionFunc(motionfunction);

glutMainLoop();

}

****

****

**Task 03:**

**Implement a program that construct square of 100x100 at the mouse position(fig b).**

#include<Windows.h>

#include<GL\glut.h>

#include<math.h>

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(1.0f, 0.0f, 0.0f);

glPointSize(5.0);

glMatrixMode(GL\_MODELVIEW);

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void display()

{

glFlush();

}

void motionfunction(int mousex, int mousey)

{

glBegin(GL\_POLYGON);

glVertex2i(mousex , 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey + 100);

glVertex2i(mousex, 480 - mousey + 100);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(640, 480);

glutInitWindowPosition(150, 150);

glutCreateWindow("Callback motion function");

myInit();

glutDisplayFunc(display);

glutMotionFunc(motionfunction);

glutMainLoop();

}

****

#include<Windows.h>

#include<GL\glut.h>

#include<math.h>

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(1.0f, 0.0f, 0.0f);

glPointSize(5.0);

glMatrixMode(GL\_MODELVIEW);

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void display()

{

glFlush();

}

void motionfunction(int mousex, int mousey)

{

glBegin(GL\_TRIANGLES);

glVertex2i(mousex , 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey);

glVertex2i(mousex + 50, 480 - mousey + 100);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(640, 480);

glutInitWindowPosition(150, 150);

glutCreateWindow("Callback motion function");

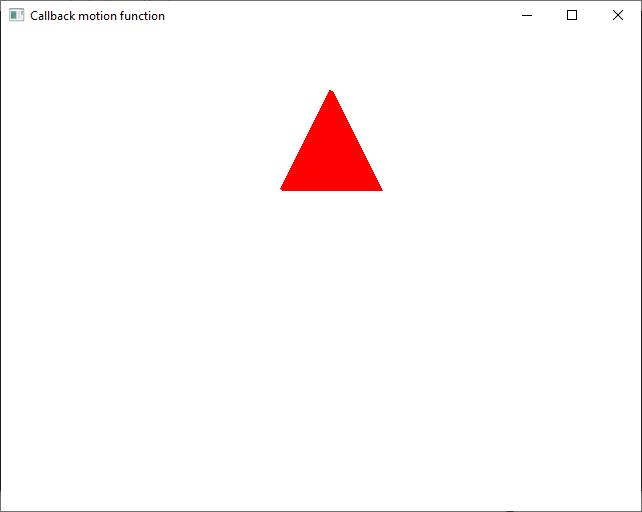
myInit();

glutDisplayFunc(display);

glutMotionFunc(motionfunction);

glutMainLoop();

}

****

**Task 04:**

**Impement a program that rotates the square using glRotatef(ang, X,Y,Z) (fig c)**

#include<Windows.h>

#include<GL\glut.h>

#include<math.h>

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(1.0f, 0.0f, 0.0f);

glPointSize(5.0);

glMatrixMode(GL\_MODELVIEW);

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void display()

{

glFlush();

}

void motionfunction(int mousex, int mousey)

{

glRotatef(45, 0, 0, 1);

glBegin(GL\_POLYGON);

glVertex2i(mousex, 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey + 100);

glVertex2i(mousex, 480 - mousey + 100);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(640, 480);

glutInitWindowPosition(150, 150);

glutCreateWindow("Callback motion function");

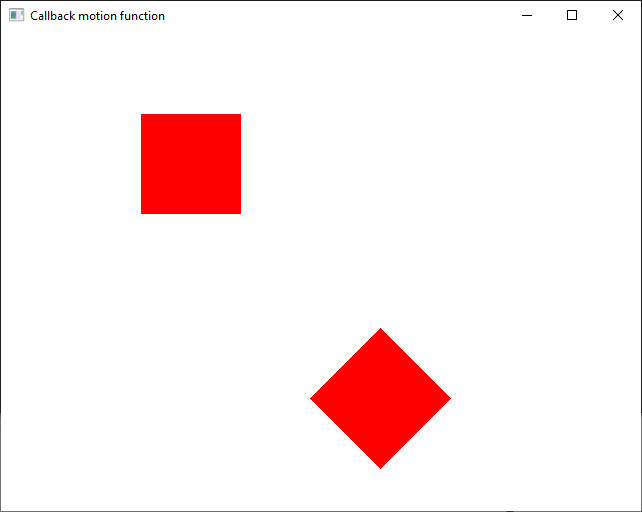
myInit();

glutDisplayFunc(display);

glutMotionFunc(motionfunction);

glutMainLoop();

}

****

**Task 05:**

**Change the color of square randomly.**

#include<Windows.h>

#include<GL\glut.h>

#include<math.h>

void myInit()

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(1.0f, 0.0f, 0.0f);

glPointSize(5.0);

glMatrixMode(GL\_MODELVIEW);

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void display()

{

glFlush();

}

void motionfunction(int mousex, int mousey)

{

float r = (float)rand() / RAND\_MAX;

float g = (float)rand() / RAND\_MAX;

float b = (float)rand() / RAND\_MAX;

glColor3f(r, g, b);

glRotatef(45, 0, 0, 1);

glBegin(GL\_POLYGON);

glVertex2i(mousex, 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey);

glVertex2i(mousex + 100, 480 - mousey + 100);

glVertex2i(mousex, 480 - mousey + 100);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(640, 480);

glutInitWindowPosition(150, 150);

glutCreateWindow("Callback motion function");

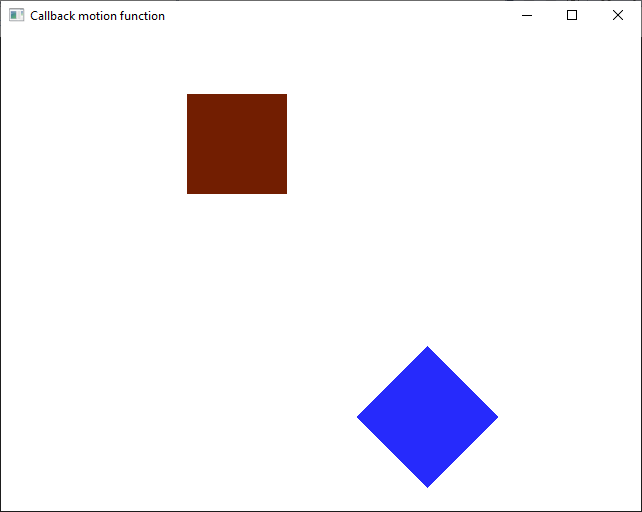
myInit();

glutDisplayFunc(display);

glutMotionFunc(motionfunction);

glutMainLoop();

}

****