**Lab Report. 04**

**Subject: Computer Graphics Lab**

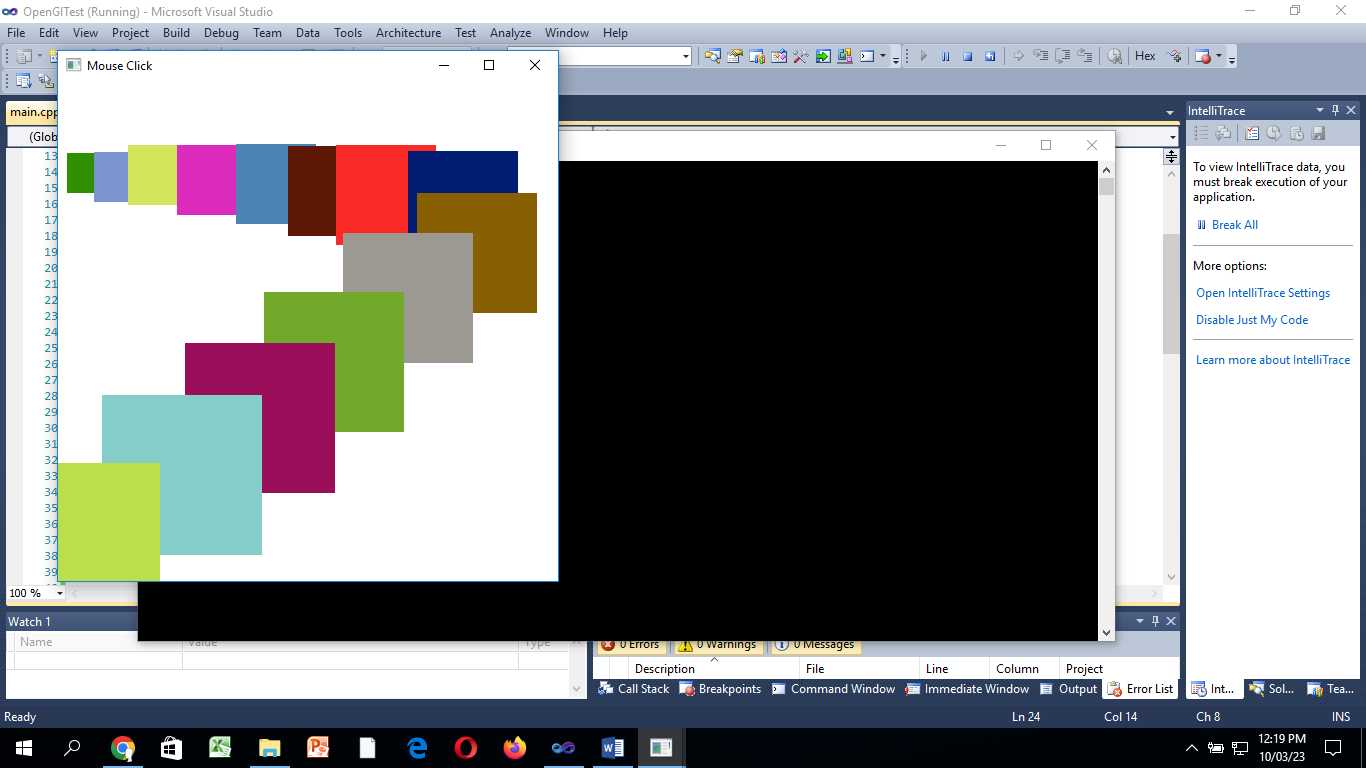
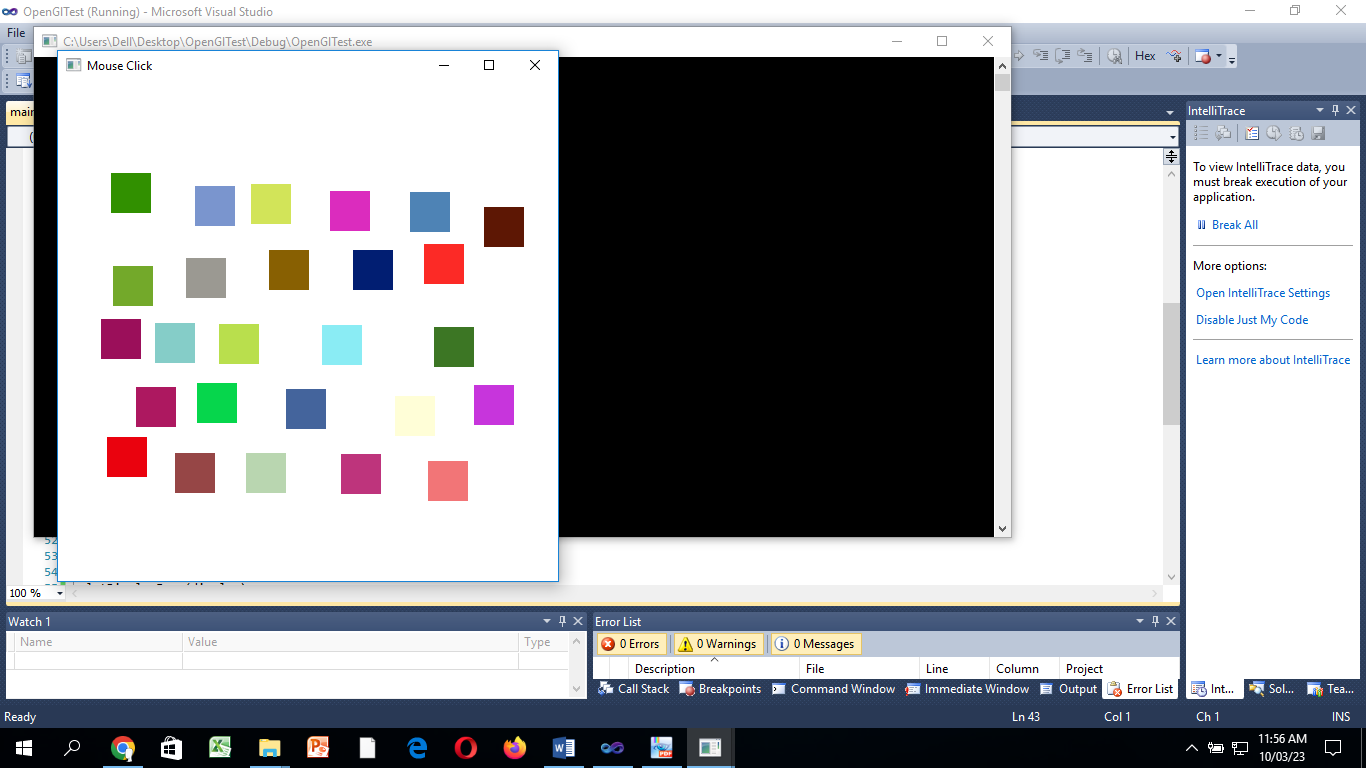
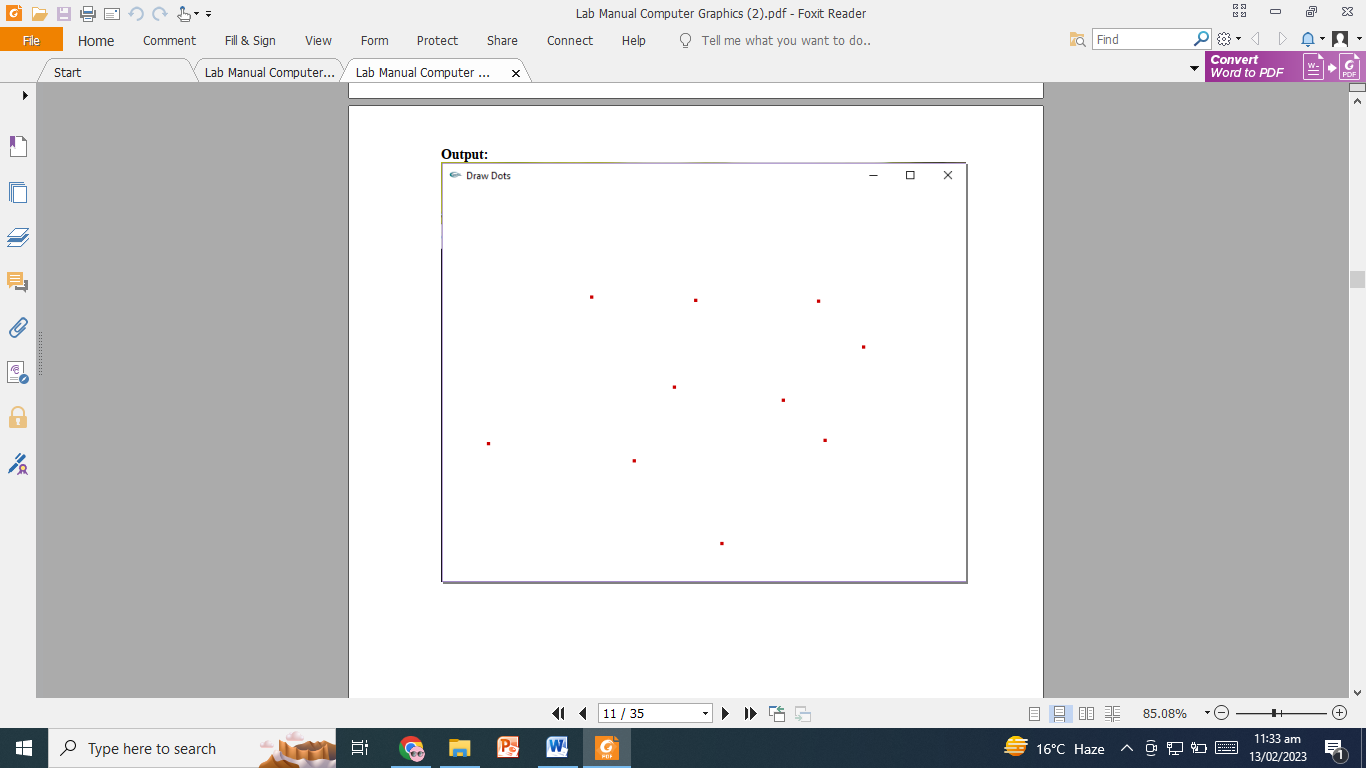


Submitted to: Dr. Junaid

Submitted by: Nayab Ashraf (RC-299)

**Date of Submission: 02 October, 2024**

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



**Task 01:**

**Update the code so that it camputes all the data points create using mouse (fig a).**

#include<stdio.h>

#include<iostream>

#include<GL\glut.h>

#include<math.h>

using namespace std;

void myInit()

{

glClearColor(0.0, 0.0, 0.0, 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

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void MyMouseFunc(int button, int state, int mousex, int mousey)

{

if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

glColor3f(0.8f, 0.0f, 0.0f);

glPointSize(4.0);

glBegin(GL\_POINTS);

glVertex2i(mousex, 500 - mousey);

glEnd();

glFlush();

}

else if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN)

{

exit(0);

}

}

void display()

{

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(500, 500);

glutInitWindowPosition(50, 50);

glutCreateWindow("Mouse Click");

myInit();

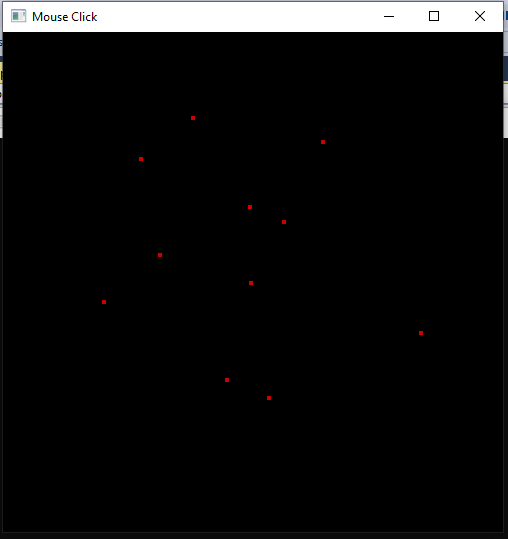
glutDisplayFunc(display);

glutMouseFunc(MyMouseFunc);

glutMainLoop();

return 0;

}



**Task 02:**

**Use rand() function to draw each dot with a different color (fig b).**

#include<stdio.h>

#include<iostream>

#include<GL\glut.h>

#include<math.h>

using namespace std;

void myInit()

{

glClearColor(0.0, 0.0, 0.0, 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

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void setRandomColor()

{

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

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

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

glColor3f(r, g, b);

}

void MyMouseFunc(int button, int state, int mousex, int mousey)

{

if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

glColor3f(0.8f, 0.0f, 0.0f);

setRandomColor();

glPointSize(10.0);

glBegin(GL\_POINTS);

glVertex2i(mousex, 500 - mousey);

glEnd();

glFlush();

}

else if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN)

{

exit(0);

}

}

void display()

{

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(500, 500);

glutInitWindowPosition(50, 50);

glutCreateWindow("Mouse Click");

myInit();

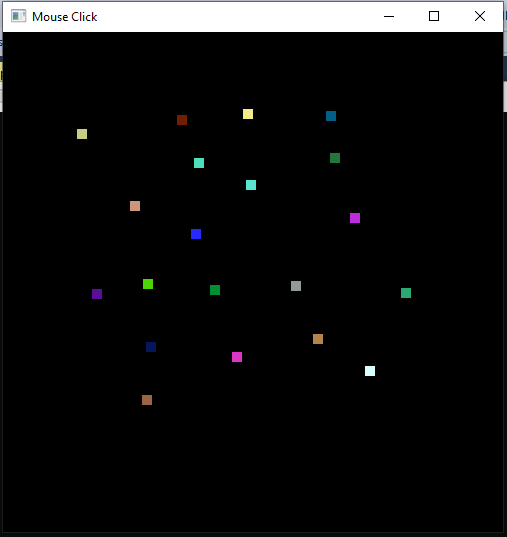
glutDisplayFunc(display);

glutMouseFunc(MyMouseFunc);

glutMainLoop();

return 0;

}



**Task 03:**

**Each mouse left-click increases the size of dot by 10 (fig c).**

#include<stdio.h>

#include<iostream>

#include<GL\glut.h>

#include<math.h>

using namespace std;

float dotSize = 10;

void myInit()

{

glClearColor(0.0, 0.0, 0.0, 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

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void setRandomColor()

{

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

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

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

glColor3f(r, g, b);

}

void MyMouseFunc(int button, int state, int mousex, int mousey)

{

if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

glColor3f(0.8f, 0.0f, 0.0f);

setRandomColor();

glPointSize(dotSize);

glBegin(GL\_POINTS);

glVertex2i(mousex, 500 - mousey);

glEnd();

dotSize += 10.0;

glFlush();

}

else if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN)

{

exit(0);

}

}

void display()

{

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(500, 500);

glutInitWindowPosition(50, 50);

glutCreateWindow("Mouse Click");

myInit();

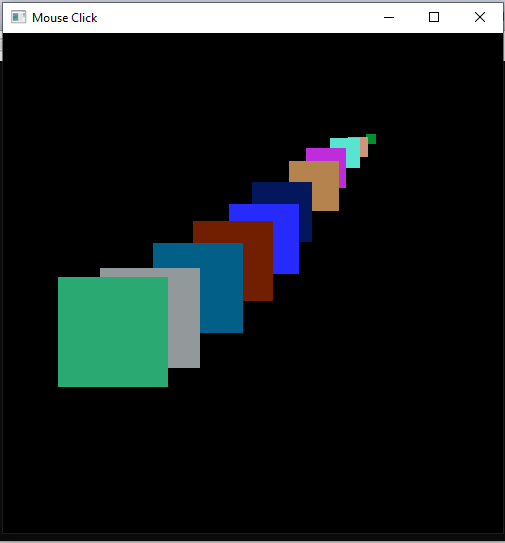
glutDisplayFunc(display);

glutMouseFunc(MyMouseFunc);

glutMainLoop();

return 0;

}

****

**Task 04:**

**Right-click decreases the size of dot by 10.**

#include<stdio.h>

#include<iostream>

#include<GL\glut.h>

#include<math.h>

using namespace std;

float dotSize = 40.0;

void myInit()

{

glClearColor(0.0, 0.0, 0.0, 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

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void setRandomColor()

{

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

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

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

glColor3f(r, g, b);

}

void MyMouseFunc(int button, int state, int mousex, int mousey)

{

if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

glColor3f(0.8f, 0.0f, 0.0f);

setRandomColor();

glPointSize(dotSize);

glBegin(GL\_POINTS);

glVertex2i(mousex, 500 - mousey);

glEnd();

glFlush();

dotSize += 10.0;

}

else if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN)

{

setRandomColor();

glPointSize(dotSize);

glBegin(GL\_POINTS);

glVertex2i(mousex, 500 - mousey);

glEnd();

dotSize -= 10.0;

glFlush();

}

}

void display()

{

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE);

glutInitWindowSize(500, 500);

glutInitWindowPosition(50, 50);

glutCreateWindow("Mouse Click");

myInit();

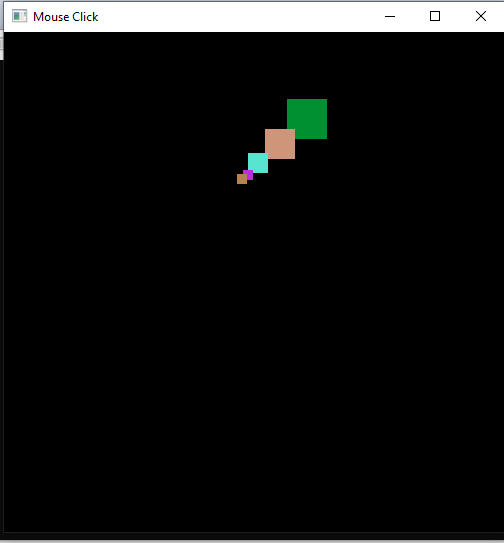
glutDisplayFunc(display);

glutMouseFunc(MyMouseFunc);

glutMainLoop();

return 0;

}

****

