Name: VAIBHAV DNYANESHWAR MOHITE

Class: SE IT A

RollNo: 25045

# ASSIGNMENT NO: 4

**CODE:**

#include<iostream>

#include<GL/glut.h> #include<math.h> using namespace std; int option = 1; struct Point { int x; int y; }; struct Color { float r; float g; float b; };

Color getPixelColor(int x, int y) {

Color color;

glReadPixels(x,y,1,1,GL\_RGB,GL\_FLOAT,&color);

return color;

}

void setPixelColor(int x, int y, Color color) {

glColor3f(color.r, color.g, color.b); glBegin(GL\_POINTS);

glVertex2i(x, y); glEnd(); glFlush(); }

void FloodFill(int x, int y, Color oldColor, Color newColor) { Color color; color = getPixelColor(x, y);

if (color.r == oldColor.r && color.g == oldColor.g && color.b == oldColor.b) {

setPixelColor(x, y, newColor); FloodFill(x + 1, y, oldColor, newColor);

FloodFill(x, y + 1, oldColor, newColor);

FloodFill(x - 1, y, oldColor, newColor);

FloodFill(x, y - 1, oldColor, newColor);

}

return;

}

void Boundaryfill(int x, int y, Color oColor, Color bColor) { Color color; color = getPixelColor(x, y);

if ((color.r != oColor.r || color.g != oColor.g || color.b != oColor.b) && (color.r != bColor.r || color.g != bColor.g || color.b != bColor.b)) {

setPixelColor(x, y, oColor); Boundaryfill(x + 1, y, oColor, bColor);

Boundaryfill(x, y + 1, oColor, bColor);

Boundaryfill(x - 1, y, oColor, bColor);

Boundaryfill(x, y - 1, oColor, bColor);

}

return;

} void display(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP); glVertex2i(250, 250); glVertex2i(250, 300); glVertex2i(300, 300); glVertex2i(300, 250); glEnd(); glFlush();

} void myinit() { glClearColor(0.0, 0.0, 0.0, 1.0); glColor3f(1.0, 1.0, 1.0); glPointSize(1.0); gluOrtho2D(0.0, 650.0, 0.0, 650.0);

}

void MYmouse(int button, int state, int x, int y) {

Color newColor = { 0.0f,1.0f,0.0f };

Color oldColor = { 0.0f,0.0f,0.0f };

Color bColor = { 1.0f,1.0f,1.0f }; Color oColor = { 1.0f,0.0f,0.0f }; if (option == 1) {

FloodFill(x, 650 - y, oldColor, newColor);

} else if (option==2){

Boundaryfill(x, 650 - y, oColor, bColor);

}

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

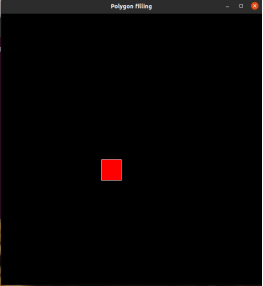
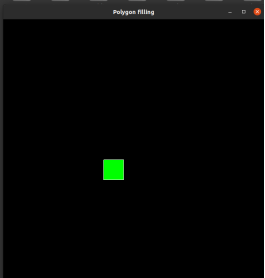
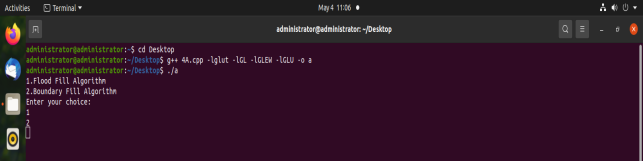
glutInitWindowSize(650, 650); glutInitWindowPosition(0, 0); glutCreateWindow("Polygon filling"); cout <<"1.Flood Fill Algorithm\n"; cout <<"2.Boundary Fill Algorithm\n"; cout <<"Enter your choice:\n"; cin >>option; glutDisplayFunc(display);

glutMouseFunc(MYmouse); myinit(); glutMainLoop();

return 0;

}

**OUTPUT:**



BOUNDARY

FILL

FLOOD

FILL