**Assignment 4**

/\*

Name: Yash Gawandare

CRN:- 24113064

Roll no.:- 4064

Batch :- S4\*/

#include <GL/glut.h> #include <iostream> #define w 640

#define h 480

using namespace std;

float boundaryColor[3] = {1, 0, 0}, interiorColor[3] = {1, 1, 1}, fillColor[3] = {0, 0, 0}, readpixel[3]; bool fillalgo = true;

void setpixel(int x, int y) { glColor3fv(fillColor); glBegin(GL\_POINTS); glVertex2f(x, y); glEnd();

glFlush();

}

void getpixel(int x, int y, float \*color) { glReadPixels(x, y, 1, 1, GL\_RGB, GL\_FLOAT, color);

}

//Boundary Fill Algorithm void boundaryFill(int x, int y) {

if (x >= w || x < 0 || y >= h || y < 0) return;

getpixel(x, y, readpixel);

if ((readpixel[0] != boundaryColor[0] || readpixel[1] != boundaryColor[1] || readpixel[2] != boundaryColor[2]) &&

(readpixel[0] != fillColor[0] || readpixel[1] != fillColor[1] || readpixel[2] != fillColor[2])) { setpixel(x, y);

boundaryFill(x + 1, y); boundaryFill(x - 1, y); boundaryFill(x, y + 1); boundaryFill(x, y - 1);

}

}

//Flood fill algorithm

void floodfill(int x, int y) {

if (x >= w || x < 0 || y >= h || y < 0) return;

getpixel(x, y, readpixel);

if (readpixel[0] == interiorColor[0] && readpixel[1] == interiorColor[1] && readpixel[2] == interiorColor[2]) {

setpixel(x, y); floodfill(x + 1, y); floodfill(x - 1, y); floodfill(x, y + 1); floodfill(x, y - 1);

}

}

void drawHollowRectangle(int x1, int y1, int x2, int y2) { glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_LINE\_LOOP); glVertex2f(x1, y1); glVertex2f(x2, y1); glVertex2f(x2, y2); glVertex2f(x1, y2); glEnd();

glFlush();

}

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

if (state == GLUT\_DOWN && button == GLUT\_LEFT\_BUTTON) { int X = x;

int Y = h- y;

if (fillalgo) { floodfill(X, Y);

} else {

boundaryFill(X, Y);

}

}

}

void menu(int index) { if (index == 1) {

fillalgo = true;

} else if (index == 2) { fillalgo = false;

} else if (index == 3) { exit(0);

}

glClear(GL\_COLOR\_BUFFER\_BIT);

drawHollowRectangle(100, 100, 250, 250); glFlush();

}

void display() { glClear(GL\_COLOR\_BUFFER\_BIT); drawHollowRectangle(100, 100, 250, 250); glFlush();

}

void init() {

glClearColor(1.0, 1.0, 1.0, 1.0);

glColor3f(1.0, 0.0, 0.0); glMatrixMode(GL\_PROJECTION); glLoadIdentity();

gluOrtho2D(0, w, 0, h);

}

int main(int argc, char \*\*argv) { glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); glutInitWindowSize(w, h); glutInitWindowPosition(100, 150); glutCreateWindow("Seed Fill Algorithm");

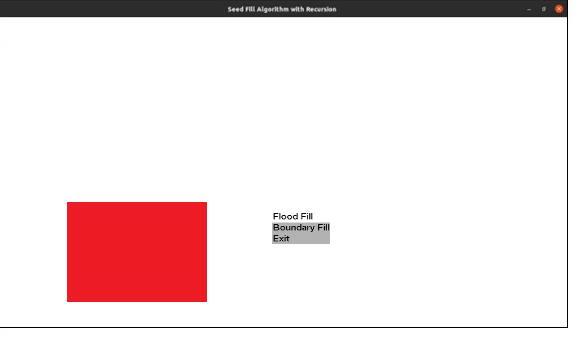
glutCreateMenu(menu); glutAddMenuEntry("Flood Fill", 1);

glutAddMenuEntry("Boundary Fill", 2);

glutAddMenuEntry("Exit", 3); glutAttachMenu(GLUT\_RIGHT\_BUTTON);

init(); glutDisplayFunc(display); glutMouseFunc(mouseClick); glutMainLoop();

return 0;

****

****