//boundry fill

#include <GL/freeglut.h>

#include <GL/gl.h>

// Function to set pixel color at given coordinates

void setPixel(int x, int y, float color[3]) {

glBegin(GL\_POINTS);

glColor3fv(color);

glVertex2i(x, y);

glEnd();

glFlush();

}

// Function to get pixel color at given coordinates

void getPixel(int x, int y, float color[3]) {

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

}

// Boundary Fill Algorithm

void boundaryFill(int x, int y, float fillColor[3], float boundaryColor[3]) {

float interiorColor[3];

getPixel(x, y, interiorColor);

// Check if current pixel is boundary color or already filled with fill color

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

(interiorColor[0] != fillColor[0] || interiorColor[1] != fillColor[1] || interiorColor[2] != fillColor[2])) {

setPixel(x, y, fillColor); // Fill current pixel with fill color

// Recursively fill adjacent pixels

boundaryFill(x + 1, y, fillColor, boundaryColor);

boundaryFill(x - 1, y, fillColor, boundaryColor);

boundaryFill(x, y + 1, fillColor, boundaryColor);

boundaryFill(x, y - 1, fillColor, boundaryColor);

}

}

// Render function

void render() {

glClearColor(1, 1, 1, 1); // Set clear color to white

glClear(GL\_COLOR\_BUFFER\_BIT);

// Draw boundary polygon

glColor3f(0, 0, 0); // Set boundary color to black

glBegin(GL\_LINE\_LOOP);

glVertex2i(100, 100);

glVertex2i(200, 100);

glVertex2i(200, 200);

glVertex2i(100, 200);

glEnd();

glColor3f(0, 0, 0); // Set boundary color to black

glBegin(GL\_LINE\_LOOP);

glVertex2i(150, 150);

glVertex2i(250, 150);

glVertex2i(250, 250);

glVertex2i(150, 250);

glEnd();

// Perform boundary fill starting from seed point (150, 150) with fill color red

float fillColor[3] = {1, 0, 0}; // Fill color: Red

float boundaryColor[3] = {0, 0, 0}; // Boundary color: Black

boundaryFill(125, 125, fillColor, boundaryColor);

float filColor[3] = {0, 1, 0}; // Fill color: Red

float boundarColor[3] = {0, 0, 0}; // Boundary color: Black

boundaryFill(175, 175, filColor, boundarColor);

glFlush();

}

// Main function

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(400, 400);

glutCreateWindow("Boundary Fill Algorithm");

gluOrtho2D(0, 400, 0, 400);

glutDisplayFunc(render);

glutMainLoop();

return 0;

}