#include <math.h>

#include <GL/glut.h>

struct Point {

GLint x;

GLint y;

};

struct Color {

GLfloat r;

GLfloat g;

GLfloat b;

};

void draw\_dda(Point p1, Point p2) {

GLfloat dx = p2.x - p1.x;

GLfloat dy = p2.y - p1.y;

GLfloat x1 = p1.x;

GLfloat y1 = p1.y;

GLfloat step = 0;

if(abs(dx) > abs(dy)) {

step = abs(dx);

} else {

step = abs(dy);}

GLfloat xInc = dx/step;

GLfloat yInc = dy/step;

for(float i = 1; i <= step; i++) {

glVertex2i(x1, y1);

x1 += xInc;

y1 += yInc; } }

void init() {

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(0.0, 0.0, 0.0);

glPointSize(1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0, 640, 0, 480); }

Color getPixelColor(GLint x, GLint y) {

Color color;

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

return color; }

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

glColor3f(color.r, color.g, color.b);

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

glFlush(); }

void floodFill(GLint x, GLint 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 onMouseClick(int button, int state, int x, int y) {

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

Color oldColor = {1.0f, 1.0f, 1.0f};

floodFill(101, 199, oldColor, newColor); }

void display(void) {

Point p1 = {100, 100}, // bottom-right

p2 = {200, 100}, // bottom-left

p3 = {200, 200}, // top-right

p4 = {100, 200}; // top-left

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

draw\_dda(p1, p2);

draw\_dda(p2, p3);

draw\_dda(p3, p4);

draw\_dda(p4, p1); glEnd();

glFlush(); }

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(200, 200);

glutCreateWindow("Open GL"); init();

glutDisplayFunc(display);

glutMouseFunc(onMouseClick);

glutMainLoop(); return 0; }

