Name : Shrikant Gavhale

Div A, Batch B

Roll no 26

**Flood fill circle algorithm**

**Code :**

#include <windows.h>

#include <bits/stdc++.h>

#include <math.h>

#include <gl/glut.h>

struct Point {

GLint x;

GLint y;

};

struct Color {

GLfloat r;

GLfloat g;

GLfloat b;

};

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(320, 240, oldColor, newColor);

}

void draw\_circle(Point pC, GLfloat radius) {

GLfloat step = 1/radius;

GLfloat x, y;

for(GLfloat theta = 0; theta <= 360; theta += step) {

x = pC.x + (radius \* cos(theta));

y = pC.y + (radius \* sin(theta));

glVertex2i(x, y);

}

}

void display(void) {

Point pt = {320, 240};

GLfloat radius = 50;

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

draw\_circle(pt, radius);

glEnd();

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(200, 200);

glutCreateWindow("Flood Fill Circle");

init();

glutDisplayFunc(display);

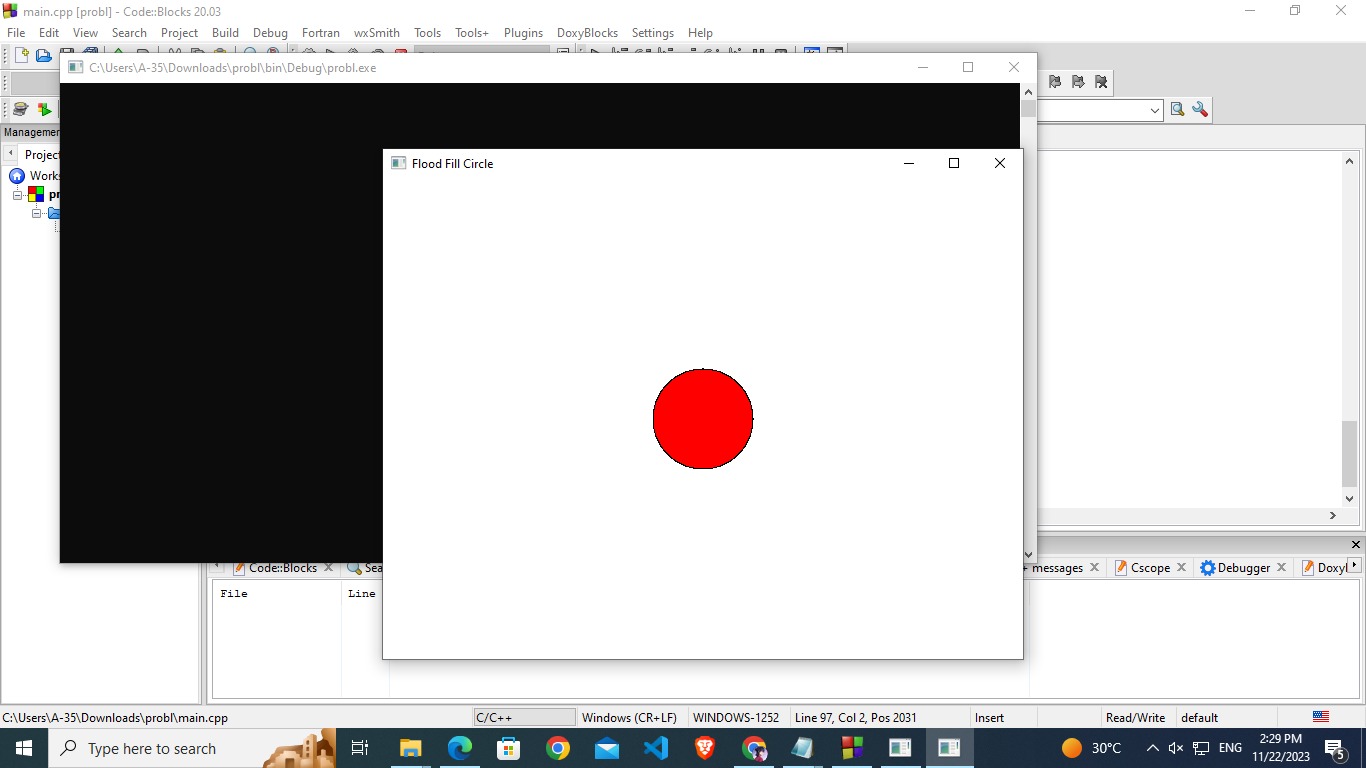
glutMouseFunc(onMouseClick);

glutMainLoop();

return 0;

}

**Output :**

****