#include <stdlib.h>

#include <iostream>

using namespace std;

#include <GL/glut.h>

float w = 640, h = 480, sizef = 0, r = 1, b = 1, g = 1;

float x1, y5, x2, y2, radi;

void init() {

gluOrtho2D(0, w, 0, h);

}

void dashed(float x1, float y5, float x2, float y2) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(1, 0x00FF);

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2f(x1, y5);

glVertex2f(x2, y2);

glEnd();

glDisable(GL\_LINE\_STIPPLE);

glFlush();

}

void dashed\_dot(float x1, float y5, float x2, float y2) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(1, 0x1C47);

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2f(x1, y5);

glVertex2f(x2, y2);

glEnd();

glDisable(GL\_LINE\_STIPPLE);

glFlush();

}

void solid\_line(float x1, float y5, float x2, float y2) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2f(x1, y5);

glVertex2f(x2, y2);

glEnd();

glFlush();

}

void circle\_line(float r) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0f, 1.0f, 1.0f); // Set color to white

glBegin(GL\_LINE\_LOOP);

for (int i = 0; i < 100; ++i) {

float theta = 2.0f \* 3.1415926f \* float(i) / float(100);

float x = r \* cosf(theta) + w / 2;

float y = r \* sinf(theta) + h / 2;

glVertex2f(x, y);

}

glEnd();

}

void circle\_dots(float r) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0f, 1.0f, 1.0f); // Set color to white

glBegin(GL\_POINTS);

for (int i = 0; i < 100; ++i) {

float theta = 2.0f \* 3.1415926f \* float(i) / float(100);

float x = r \* cosf(theta) + w / 2;

float y = r \* sinf(theta) + h / 2;

glVertex2f(x, y);

}

glEnd();

glFlush();

}

void circle\_strokes(float r) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0f, 1.0f, 1.0f); // Set color to white

glBegin(GL\_LINES);

for (int i = 0; i <= 36; i++)

{

float angle = 3.14 / 12 \* i;

glVertex2f(r \* cos(angle) + w / 2, r \* sin(angle) + w / 2);

glVertex2f(40 \* cos(angle) + w / 2, 40 \* sin(angle) + w / 2);

}

glEnd();

glFlush();

}

void circle\_quads(float r) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0f, 1.0f, 1.0f); // Set color to white

int num\_segments = 100;

float angle\_increment = 2.0f \* 3.1415926f / num\_segments;

glBegin(GL\_QUADS);

for (int i = 0; i < num\_segments; ++i) {

float theta1 = angle\_increment \* i;

float theta2 = angle\_increment \* (i + 1);

float x1 = r \* cosf(theta1) + w / 2;

float y1 = r \* sinf(theta1) + h / 2;

float x2 = r \* cosf(theta2) + w / 2;

float y2 = r \* sinf(theta2) + h / 2;

glVertex2f(w / 2, h / 2);

glVertex2f(x1, y1);

glVertex2f(x2, y2); // Second vertex

glVertex2f(w / 2, h / 2); // Center again to close the quad

}

glEnd();

}

void disp() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glEnd();

glFlush();

}

void d\_menu(int op) {

switch (op) {

case 1:

cout << "Enter the radius for the circle: ";

cin >> radi;

circle\_line(radi);

break;

case 2:

cout << "Enter the radius for the circle: ";

cin >> radi;

circle\_dots(radi);

break;

case 3:

cout << "Enter the radius for the circle: ";

cin >> radi;

circle\_strokes(radi);

break;

case 4:

cout << "Enter the radius for the circle: ";

cin >> radi;

circle\_quads(radi);

break;

case 5:

cout << "Enter x1, y1, x2, y2: ";

cin >> x1 >> y5 >> x2 >> y2;

dashed(x1, y5, x2, y2);

break;

case 6:

cout << "Enter x1, y1, x2, y2: ";

cin >> x1 >> y5 >> x2 >> y2;

dashed\_dot(x1, y5, x2, y2);

break;

case 7:

cout << "Enter x1, y1, x2, y2: ";

cin >> x1 >> y5 >> x2 >> y2;

solid\_line(x1, y5, x2, y2);

break;

}

glutPostRedisplay();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(300, 150);

glutCreateWindow("Menu");

init();

int lineMenu = glutCreateMenu(d\_menu);

glutAddMenuEntry("Dashed Line", 5);

glutAddMenuEntry("Dash and dot Line", 6);

glutAddMenuEntry("Solid Line", 7);

int circleMenu = glutCreateMenu(d\_menu);

glutAddMenuEntry("Using line", 1);

glutAddMenuEntry("Using dots", 2);

glutAddMenuEntry("stroked circle", 3);

glutAddMenuEntry("using quads", 4);

glutCreateMenu(d\_menu);

glutAddSubMenu("Line", lineMenu);

glutAddSubMenu("circle", circleMenu);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

glutDisplayFunc(disp);

glutMainLoop();

return 0;

}