**Assignment 3**

/\*

Name: Yash Gawandare

CRN:- 24113064

Roll no.:- 4064

Batch :- S4\*/

#include <iostream> #include <GL/glut.h> #include <math.h> using namespace std; int wl = 1400;

int wh = 900;

void myInit(void) { glClearColor(1.0, 1.0, 1.0, 1.0);

glColor3i(0, 0, 0); glPointSize(2.0);

gluOrtho2D(-wl, wl, -wh, wh);

}

int sign(int val){ if(val<0){

return -1;

}else if(val==0){ return 0;

}else{

return 1;

}

}

void BAlgorihtm(int x1,int y1,int x2,int y2){

//validation for checking source and destination points if (x1 == x2 && y1 == y2) {

cout << "Error: Source and Destination points are the same! Line cannot be drawn." << endl; return;

}

int x=x1; int y=y1;

int Dx=abs(x2-x1); int Dy=abs(y2-y1); int s1=sign(x2-x1); int s2=sign(y2-y1); int interchange;

if(Dy>Dx){

int temp=Dx;

Dx=Dy;

Dy=temp; interchange=1;

}else{

interchange=0;

}

int e=2\*Dy-Dx; glBegin(GL\_POINTS);

// glVertex2i(x,y);

for(int i=1;i<=Dx;i++){ glVertex2i(x, y);

while(e>=0){ if(interchange==1){

x=x+s1;

}else{

y=y+s2;

}

e=e-2\*Dx;

}

if(interchange==1){ y=y+s2;

}else{

x=x+s1;

}

e=e+2\*Dy;

}

glEnd(); glFlush();

}

//function to plot axis void plotAxis(){

glBegin(GL\_POINTS); for(int i=-wl; i<wl; i++){

glVertex2i(i, 0);

}

for(int i=-wh; i<wh; i++){ glVertex2i(0, i);

}

glEnd(); glFlush();

}

// Bresenham's Circle Algorithm void circleAlgo(int r, int h, int k) {

//validation if radius is less than 1

if(r<1){

cout<<"Error: Invalid radius\n";

}

int d = 3 - (2 \* r); int x = 0;

int y = r;

glBegin(GL\_POINTS); while (x <= y) {

if (d <= 0) {

d = d + 4 \* x + 6;

} else {

d = d + 4 \* (x - y) + 10; y--;

} x++;

// Plot all 8 symmetrical points glVertex2i(h + x, k + y); glVertex2i(h - x, k + y); glVertex2i(h + x, k - y); glVertex2i(h - x, k - y);

glVertex2i(h + y, k + x); glVertex2i(h - y, k + x); glVertex2i(h + y, k - x); glVertex2i(h - y, k - x);

}

glEnd(); glFlush();

}

void olympicRing() { glPointSize(5.0);

glColor3f(0.0, 0.129, 0.584); // Olympic blue

circleAlgo(100, 100, 300);

glColor3f(1.0, 0.8, 0.0); // Olympic yellow

circleAlgo(100, 300, 300);

glColor3f(0.0, 0.0, 0.0); // Black

circleAlgo(100, 500, 300);

glColor3f(0.0, 0.6, 0.2); // Olympic green

circleAlgo(100, 200, 200);

glColor3f(0.906, 0.082, 0.129); // Olympic red

circleAlgo(100, 400, 200);

}

void drawShape() { int x = 500;

int y = -400;

// Draw outer circle circleAlgo(200, x, y);

glBegin(GL\_LINES);

// First line of triangle BAlgorihtm(500, -200,327, -500);

// Second line of triangle BAlgorihtm(327, -500, 673, -500);

// Third line of triangle BAlgorihtm(673, -500, 500, -200);

glEnd();

// Draw inner circle circleAlgo(80, x, y);

}

void display() { glClear(GL\_COLOR\_BUFFER\_BIT); circleAlgo(-100,0,0);

glFlush();

}

void drawSpiral() { float angle = 0.0; float radius = 5.0;

float angleIncrement = 0.1; // Smaller for smoother curves float radiusIncrement = 1.5; // Adjusts spacing

glBegin(GL\_LINE\_STRIP); while (radius < 300) {

int x = -radius \* cos(angle) - 700; // Move further left int y = radius \* sin(angle) + 300; // Move upward

glVertex2i(x, y);

angle += angleIncrement; radius += radiusIncrement;

}

glEnd(); glFlush();

}

void menu(int c) {

// glClear(GL\_COLOR\_BUFFER\_BIT);

if (c == 1) { // simple circle glColor3i(0, 0, 0);

circleAlgo(100, 0, 0);

} else if (c == 2) { olympicRing();

} else if (c == 3) { glColor3i(0, 0, 0);

//concentric circle

circleAlgo(200, -300, -300); // Outer circle

circleAlgo(100, -300, -300); // Inner circle

}else if(c==4){ glColor3i(0, 0, 0); drawSpiral();

}

else if(c==5){

plotAxis(); //to plot the axis

}else if(c==6){

drawShape(); // to draw the shape

}

glFlush();

}

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

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); glutInitWindowSize(wl, wh); glutInitWindowPosition(100, 50); glutCreateWindow("Bresenham's Circle Algorithm");

glutDisplayFunc(display); myInit();

glutCreateMenu(menu); glutAddMenuEntry("Simple circle", 1);

glutAddMenuEntry("Olympic ring", 2);

glutAddMenuEntry("Concentric circle", 3);

glutAddMenuEntry("Spiral", 4);

glutAddMenuEntry("Draw Axis", 5);

glutAddMenuEntry("Draw shape", 6); glutAttachMenu(GLUT\_RIGHT\_BUTTON);

glutMainLoop(); return 0;

}





