# Experiment no : 03

Name : Atharva B. Iparkar Date : 03/09/2024

Roll no : S211045

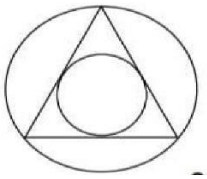
Class : S.E.

Div : A

Batch : A-2

## Problem Statement :

Write a C++ program to draw the following pattern. Use DDA line and Bresenham's circle algorithm. Apply the concept of encapsulation.



## Code :

```
#include <iostream>

#include <graphics.h>

#include <stdlib.h>

using namespace std;

class DCircle {

private:

int x0, y0;

public:

DCircle() : x0(0), y0(0) {}

void setOffset(int xx, int yy) {

x0 = xx;
```

```
y0 = yy;
}
void drawCircle(int x1, int y1, int r) {
float d = 3 - 2 * r;
int x = 0, y = r;
do {
putpixel(x1 + x0 + x, y0 + y - y1, 15);
putpixel(x1 + x0 + y, y0 + x - y1, 15);
putpixel(x1 + x0 + y, y0 - x - y1, 15);
putpixel(x1 + x0 + x, y0 - y - y1, 15);
putpixel(x1 + x0 - x, y0 - y - y1, 15);
putpixel(x1 + x0 - y, y0 - x - y1, 15);
putpixel(x1 + x0 - y, y0 + x - y1, 15);
putpixel(x1 + x0 - x, y0 + y - y1, 15);
if (d <= 0) {
d = d + 4 * x + 6;
} else {
d = d + 4 * (x - y) + 10;
y--;
}
x++;
} while (x < y);
```

```
}
};
class Point {
protected:
int xco, yco, color;
public:
Point() : xco(0), yco(0), color(15) {}
void setCoordinates(int x, int y) {
xco = x;
yco = y;
}
void setColor(int c) {
color = c;
}
void draw() {
putpixel(xco, yco, color);
}
};
class DLine : public Point {
private:
int x2, y2;
public:
```

```
DLine() : Point(), x2(0), y2(0) {}

void setLine(int x, int y, int xx, int yy) {

Point::setCoordinates(x, y);

x2 = xx;

y2 = yy;

}

void drawLine(int color) {

float dx = abs(x2 - xco);

float dy = abs(y2 - yco);

float length = (dx >= dy) ? dx : dy;

float xInc = (x2 - xco) / length;

float yInc = (y2 - yco) / length;

float x = xco + 0.5, y = yco + 0.5;

for (int i = 1; i <= length; i++) {

Point::setCoordinates(x, y);

Point::setColor(color);

Point::draw();

x += xInc;

y += yInc;

}

Point::setCoordinates(x, y);

Point::draw();
```

```
}

};

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, NULL);

int x, y, r, x1, x2, y1, y2, xmax, ymax, xmid, ymid, n;

DCircle circle;

cout << "Enter coordinates of the center of the circle:\n";

cout << "Enter the value of x: ";

cin >> x;

cout << "Enter the value of y: ";

cin >> y;

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

cin >> r;

xmax = getmaxx();

ymax = getmaxy();

xmid = xmax / 2;

ymid = ymax / 2;

setcolor(1);

circle.setOffset(xmid, ymid);

line(xmid, 0, xmid, ymax);

line(0, ymid, xmax, ymid);
```

```
setcolor(15);

circle.drawCircle(x, y, r);

cout << "Enter total number of lines: ";

cin >> n;

DLine line;

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

cout << "Enter coordinates of point x1: ";

cin >> x1;

cout << "Enter coordinates of point y1: ";

cin >> y1;

cout << "Enter coordinates of point x2: ";

cin >> x2;

cout << "Enter coordinates of point y2: ";

cin >> y2;

line.setLine(x1 + xmid, ymid - y1, x2 + xmid, ymid - y2);

line.drawLine(15);

}

cout << "Enter coordinates of the center of the circle:\n";

cout << "Enter the value of x: ";

cin >> x;

cout << "Enter the value of y: ";

cin >> y;
```

```
cout << "Enter the radius of the circle: ";

cin >> r;

setcolor(5);

circle.drawCircle(x, y, r);

getch();

delay(200);

closegraph();

return 0;

}
```

## Output :

```
d_comp_pl_ii_11@d-comp-pl-ii-11:~/SE_A2_S211045_Atharva$ g++ Pattern.cpp -o p -
lgraph

d_comp_pl_ii_11@d-comp-pl-ii-11:~/SE_A2_S211045_Atharva$ ./p
Enter coordinates of the center of the circle:
Enter the value of x: [xcb] Unknown sequence number while processing queue
[xcb] Most likely this is a multi-threaded client and XInitThreads has not been called
[xcb] Aborting, sorry about that.
p: ../../src/xcb_io.c:260: poll_for_event: Assertion `!xcb_xlib_threads_sequence_lost' failed.
100
Enter the value of y: 70
Enter the radius of the circle: 30
Enter total number of lines: 3
Enter coordinates of point x1: 40
Enter coordinates of point y1: 40
Enter coordinates of point x2: 100
Enter coordinates of point y2: 124
Enter coordinates of point x1: 40
Enter coordinates of point y1: 40
Enter coordinates of point x2: 160
Enter coordinates of point y2: 40
Enter coordinates of point x1: 160
Enter coordinates of point y1: 40
Enter coordinates of point x2: 100
```

Enter coordinates of point y2: 124
Enter coordinates of the center of the circle:
Enter the value of x: 100
Enter the value of y: 62
Enter the radius of the circle: 60