Experiment no : 01

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>
using namespace std;
class Shape {
protected:
 int x, y;
public:
 Shape(int x = 0, int y = 0) : x(x), y(y) {}
 virtual void draw() = 0;
};
class Line : public Shape {
private:
 int x1, y1, x2, y2;
public:
 Line(int x1, int y1, int x2, int y2) : Shape((x1 + x2) / 2, (y1 + y2) / 2),
x1(x1), y1(y1), x2(x2), y2(y2) {}
 void draw() {
 int dx = x2 - x1;
```

```
 int dy = y2 - y1;
 int steps = dx > dy ? dx : dy;
 float xInc = dx / (float)steps;
 float yInc = dy / (float)steps;
 float x = x1;
 float y = y1;
 for (int i = 0; i <= steps; i++) {
 putpixel(x, y, 14);
 x += xInc;
 y += yInc;
 }
 }
};
class Circle : public Shape {
private:
 float radius;
public:
 Circle(int x, int y, float radius) : Shape(x, y), radius(radius) {}
 void draw() {
 float x = 0, y = radius;
 float d = 3 - 2 * radius;
 display(x, y);
 while (y >= x) {
 x++;
 if (d > 0) {
 y--;
 d = d + 4 * (x - y) + 10;
 } else {
 d = d + 4 * x + 6;
 }
```

```
display(x, y);
}
}
void display(int x, int y) {
putpixel(this->x + x, this->y + y, 3);
putpixel(this->x - x, this->y + y, 3);
putpixel(this->x + x, this->y - y, 3);
putpixel(this->x - x, this->y - y, 3);
putpixel(this->x + y, this->y + x, 3);
putpixel(this->x - y, this->y + x, 3);
putpixel(this->x + y, this->y - x, 3);
putpixel(this->x - y, this->y - x, 3);
}
};
int main() {
int gd = DETECT, gm;
initgraph(&gd, &gm, NULL);
float cx, cy, radius;
  cout << "Enter the center coordinates (x y) for the circumscribed circle: "<<endl;
  cin >> cx >> cy;
  cout << "Enter the radius for the circumscribed circle: ";
  cin >> radius;
  Circle circumscribedCircle(cx, cy, radius);

  cout << "Enter the center coordinates (x y) for the inscribed circle: "<<endl;
  cin >> cx >> cy;
  cout << "Enter the radius for the inscribed circle: ";
  cin >> radius;
  Circle inscribedCircle(cx, cy, radius);
```

```
    int x1, y1, x2, y2;
    cout << "Enter the coordinates (x1 y1 x2 y2) for the first line: "<<endl;
    cin >> x1 >> y1 >> x2 >> y2;
    Line line1(x1, y1, x2, y2);

    cout << "Enter the coordinates (x1 y1 x2 y2) for the second line: "<<endl;
    cin >> x1 >> y1 >> x2 >> y2;
    Line line2(x1, y1, x2, y2);

    cout << "Enter the coordinates (x1 y1 x2 y2) for the third line: "<<endl;
    cin >> x1 >> y1 >> x2 >> y2;
    Line line3(x1, y1, x2, y2)

 circumscribedCircle.draw();
 inscribedCircle.draw();
 line1.draw();
 line2.draw();
 line3.draw();
 getch();
 closegraph();
 return 0;
}
```

Output :

```
d-comp-pl-ii-15@dcompplii15-OptiPlex-3070: ~/Downloads
d-comp-pl-ii-15@dcompplii15-OptiPlex-3070:~/Downloads$ g++ Pattern.cpp -o p -lgraph
d-comp-pl-ii-15@dcompplii15-OptiPlex-3070:~/Downloads$ ./p
Enter the center coordinates (x y) for the circumscribed circle:
[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.
150
180
Enter the radius for the circumscribed circle: 57
Enter the center coordinates (x y) for the inscribed circle:
150
180
Enter the radius for the inscribed circle: 28.5
Enter the coordinates (x1 y1 x2 y2) for the first line:
102
150
198
150
Enter the coordinates (x1 y1 x2 y2) for the second line:
102
150
150
236
Enter the coordinates (x1 y1 x2 y2) for the third line:
150
236
198
150
```