CSO351: Computer Graphics

Lab Assignment 2.Circle.(a): Circle Generation using Parametric Equation

Objective:

Write a program in C/C++ for implementation of cirlce generation using parametric equation.

Algorithm:

- **Step 1:** Get the inputs of coordinates of center (h, k) and length of radius r of the circle.

- **Step 2:** Initialize an angle theta to 0.

- **Step 3:** The following parametric equations generate the (x, y) coordinates of a point on the circle given an angle theta:

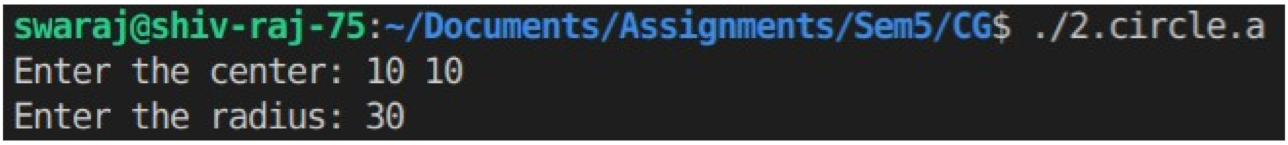
```
x = h + r cos(theta)
y = k + r sin(theta)
plot (x, y)
```

- **Step 4:** While theta >= 45°, there's an increment to theta each time round the loop. It draws straight line segments between these successive points on the circle. The circle is thus drawn as a series of straight lines. If the increment is small enough, the result looks like a circle.

- **Step 5:** Continue plotting obtained x and y on incrementing theta by 1 till it reaches 45. The points of other octants can be obtained using symmetry.

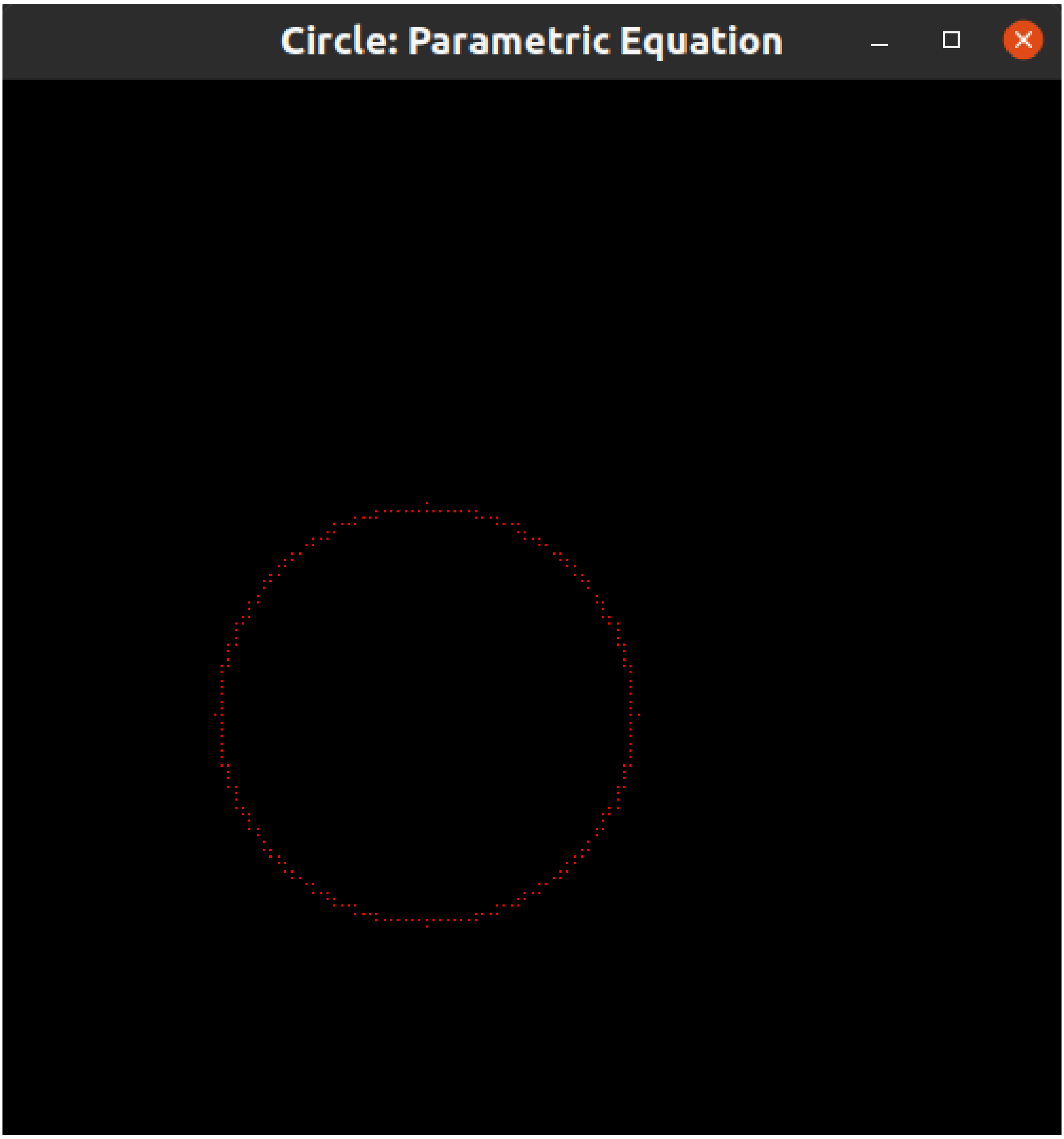
Result:

Input:



```
swaraj@shiv-raj-75:~/Documents/Assignments/Sem5/CG$ ./2.circle.a
Enter the center: 10 10
Enter the radius: 30
```

Output:



Conclusion:

- This concludes that the algorithm is very slow because the trigonometric functions are slow.
- Obtained circle is not at all smooth as seen in the generated plot.

Appendix: Code

```
#include <GL/glut.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int h, k, r;

void plot(GLint x, GLint y)
{
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

void draw(int h, int k, int x, int y)
{
    plot(h + x, k + y);
    plot(h + x, k - y);
    plot(h - x, k + y);
    plot(h - x, k - y);
    plot(h + y, k + x);
    plot(h + y, k - x);
    plot(h - y, k + x);
    plot(h - y, k - x);
}

void circle(int h, int k, int r)
{
    for (int i = 0; i <= 45 * r; i++)
    {
        float theta = (i / (180.0 * r)) * acos(-1);
        int x = r * cos(theta), y = r * sin(theta);
        draw(h, k, x, y);
    }
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    circle(h, k, r);
    glFlush();
}
```

```
void init(void)
{
	glClearColor(0.0, 0.0, 0.0, 0.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(-50, 100, -50, 100);
}

int main(int argc, char **argv)
{
	printf ("Enter the center: ");
	scanf("%d %d", &h, &k);
	printf ("Enter the radius: ");
	scanf("%d", &r);
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
	glutInitWindowSize(500, 500);
	glutInitWindowPosition(0, 0);
	glutCreateWindow("Circle: Parametric Equation");
	glutDisplayFunc(display);
	init();
	glutMainLoop();
	return 0;
}
```