CSO351: Computer Graphics

Lab Assignment 2.Circle.(c):
Circle Generation using Breshenham's Algorithm

Objective:

Write a program in C/C++ for implementation of cirlce generation using Breshenham's algorithm.

Algorithm:

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

- **Step 2:** Initialize the starting point coordinates (X_0, Y_0) as:

```
x0 = 0
y0 = 0
```

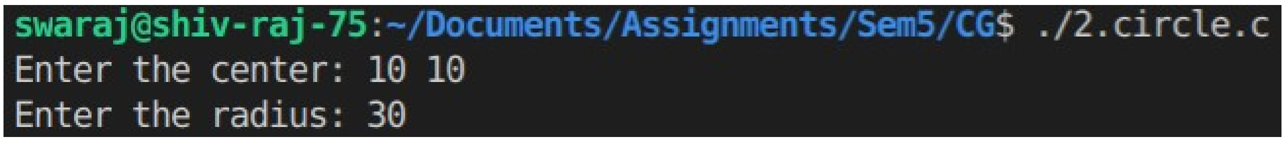
- **Step 3:** Set decision parameter, d = 3 – (2 * r).

- **Step 4:** Repeat the following until x <= y and draw the circle using 8 symmetry points shifting the center of the obtained circle to the given center coordinates:

```
if (d < 0)
{
    x = x + 1;
    d = d + (4*x) + 6;
}
else
{
    x = x + 1;
    y = y - 1;
    d = d + 4*(x - y) + 10;
}
```

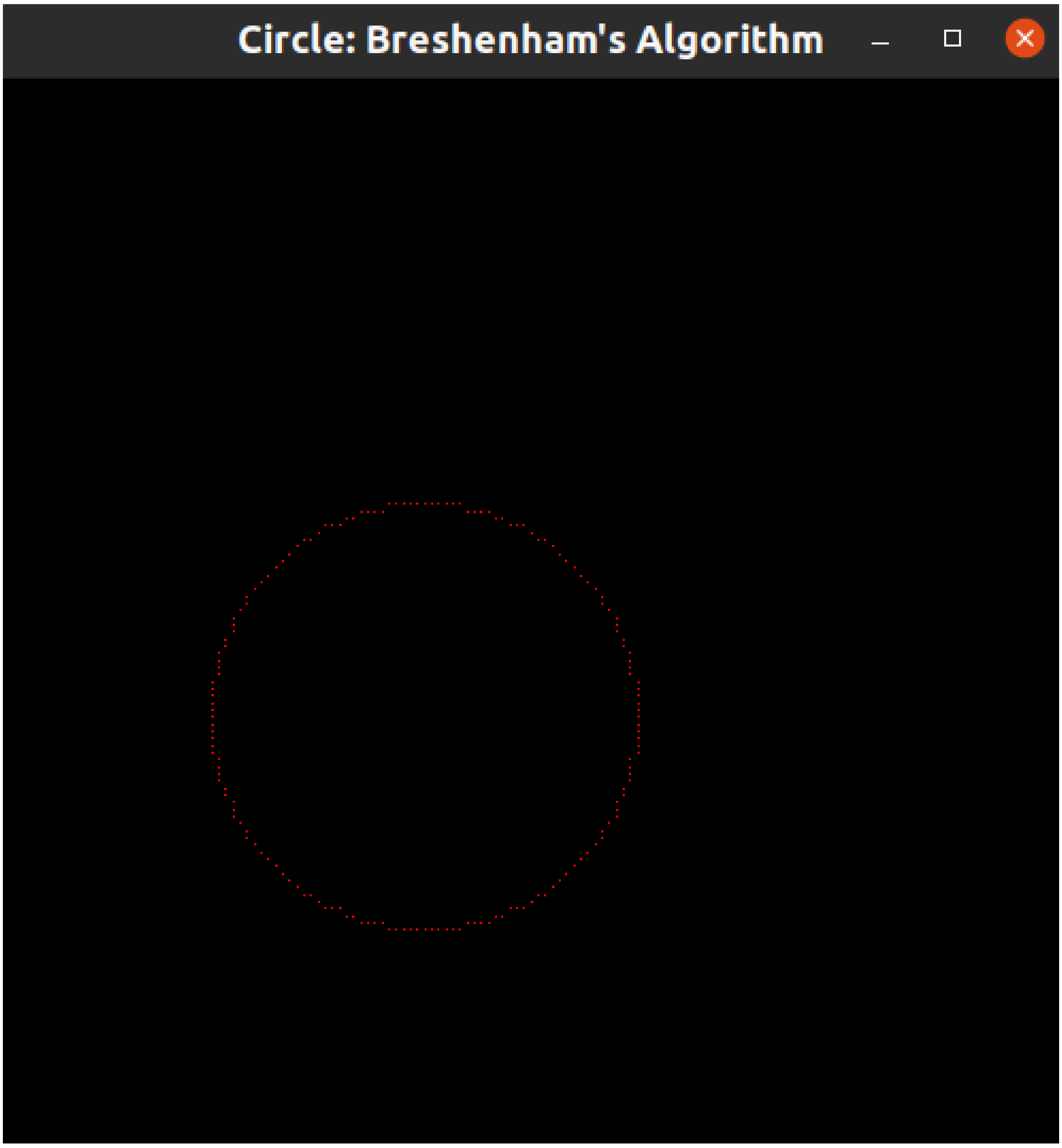
Result:

Input:



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

Output:



Conclusion:

- It only involves addition, subtraction and multiplication
- There is no need of squares, square roots and trigonometric functions.
- While generating points, there is a problem of accuracy.
- This algorithm is not suitable for complex and high graphic images.

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 xi, int yi)
{
	plot(h + xi, k + yi);
	plot(h + xi, k - yi);
	plot(h - xi, k + yi);
	plot(h - xi, k - yi);
	plot(h + yi, k + xi);
	plot(h + yi, k - xi);
	plot(h - yi, k + xi);
	plot(h - yi, k - xi);
}

void circle(int h, int k, int r)
{
	int xi = 0, yi = r, d = 3 - 2 * r;
	while (xi <= yi)
	{
		draw(h, k, xi, yi);
		if (d < 0)
			d += 4 * xi + 6;
		else
		{
			d += 4 * (xi - yi) + 10;
			yi--;
		}
		xi++;
	}
}
```

```
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: Breshenham's Algorithm");
	glutDisplayFunc(display);
	init();
	glutMainLoop();
	return 0;
}
```