CSO351: Computer Graphics

Lab Assignment 2.Circle.(b):
Circle Generation using Polynomial Equation

Objective:

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

Algorithm:

- **Step 1:** Get the inputs of coordinates of center (h, k) and length of radius r of the circle.
- **Step 2:** Set the initial variables as:

```
x = 0
i = step size
x_end = r/(2^(1/2))
```

- **Step 3:** Test to determine whether the entire circle has been scan-converted. If $x > x_{end}$ then stop.
- **Step 4:** Compute $y = (r^2 - x^2)^{1/2}$ and round it down.
- **Step 5:** Plot the eight points found by symmetry concerning the center (h, k) at the current (x, y) coordinates.

```
Plot (x + h, y +k)        Plot (-x + h, -y + k)
Plot (y + h, x + k)       Plot (-y + h, -x + k)
Plot (-y + h, x + k)      Plot (y + h, -x + k)
Plot (-x + h, y + k)      Plot (x + h, -y + k)
```

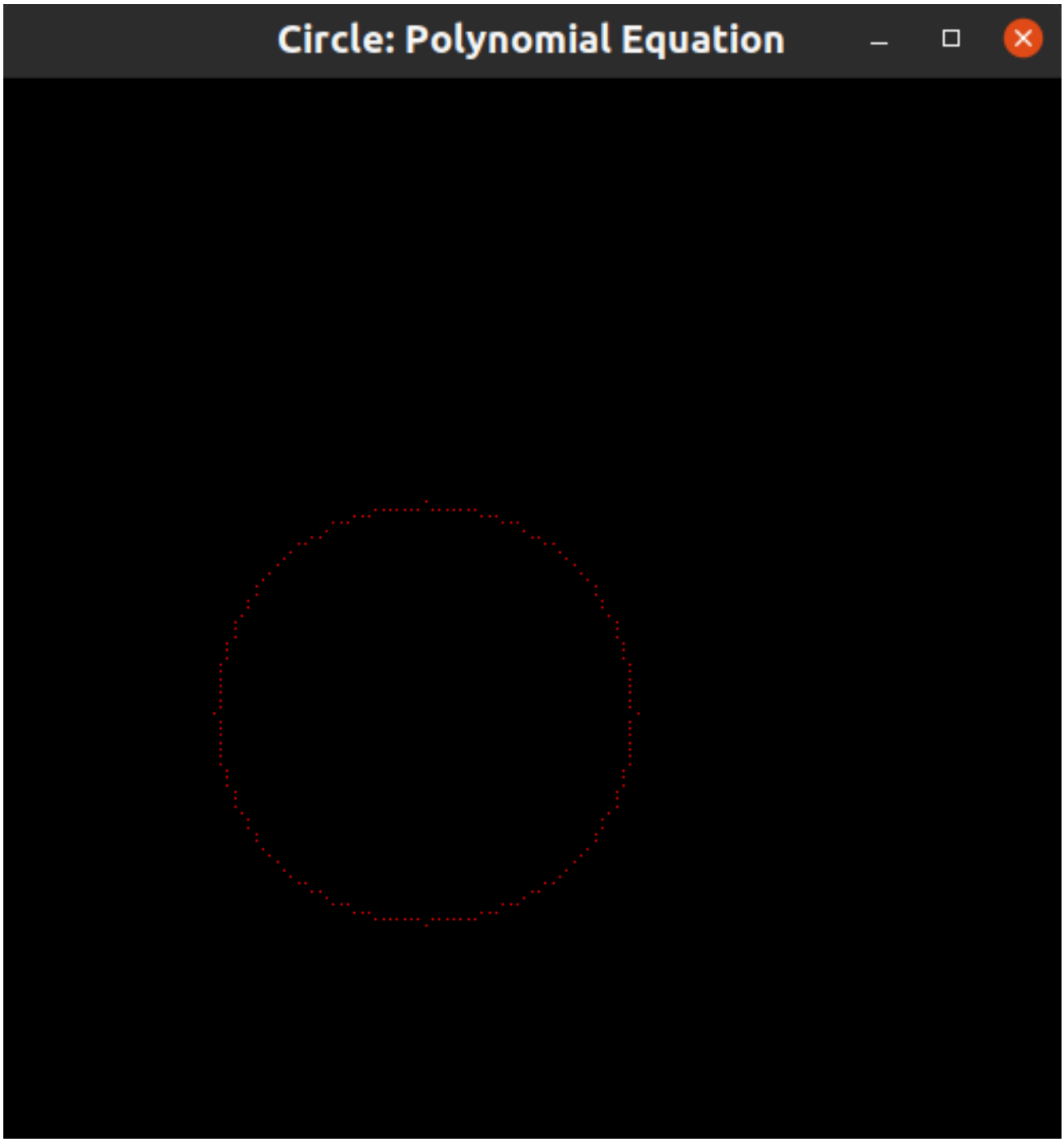
- **Step 6:** Increment x = x + i
- **Step 7:** Go to step 3.

Result:

Input:

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

Output:



Conclusion:

- The resulting circle has large gaps.
- The calculations are not very efficient with square root operations.

Appendix: Code

```
#include <stdio.h>
#include <iostream>
#include <GL/glut.h>
#include <cmath>

using namespace std;
int h, k, r;

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

void plot(int x, int y)
{
     glBegin(GL_POINTS);
     glVertex2i(x + h, y + k);
     glEnd();
}

void circle()
{
     int x = 0;
     double y = r;
     double x2 = r / sqrt(2);
     while (x <= x2)
     {
          y = sqrt(r * r - x * x);
          plot(x, floor(y));
          plot(x, -floor(y));
          plot(-x, floor(y));
          plot(-x, -floor(y));
          plot(floor(y), x);
          plot(-floor(y), x);
          plot(floor(y), -x);
          plot(-floor(y), -x);
          x += 1;
     }
}

void display(void)
{
```

```
	glClear(GL_COLOR_BUFFER_BIT);
	glColor3f(1.0, 0.0, 0.0);
	circle();
	glFlush();
}

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