Bresenham's Circle Drawing Algorithm

David Oniani
onianidavid@gmail.com

November 14, 2021

1 Derivation

Let the radius of the circle centered at $(0, 0)$ be given and equal r. Any point (x, y) on the circle satisfies the following:

$$x^2 + y^2 = r^2 \tag{1}$$

Consider the upper half of the first quadrant. The next point could then be placed either to the east $(x+1, y)$ or southeast $(x + 1, y - 1)$. The errors for both scenarios are:

$$e(x + 1, y) = (x + 1)^2 + y^2 - r^2 \tag{2}$$

$$e(x + 1, y - 1) = (x + 1)^2 + (y - 1)^2 - r^2 \tag{3}$$

The decision parameter d representing the total error is defined as follows:

$$d = e(x + 1, y) + e(x + 1, y - 1) = 2(x + 1)^2 + y^2 + (y - 1)^2 - 2r^2 \tag{4}$$

Since the x coordinate always increases, the next point is some $(x + 1, y_n)$ and we get:

$$d_n = e(x + 2, y_n) + e(x + 2, y_n - 1) = 2(x + 2)^2 + {y_n}^2 + (y_n - 1)^2 - 2r^2 \tag{5}$$

The difference is equal to the following:

$$d_n - d = 2(2x + 3) + \left({y_n}^2 - y^2\right) + \left((y_n - 1)^2 - (y - 1)^2\right) \tag{6}$$

Finally, we have:

$$d_n - d = 2(2x + 3) \qquad \Longrightarrow d_n = d + 4x + 6 \qquad \text{if } d \leq 0 \ (y_n = y) \tag{7}$$

$$d_n - d = 2(2x + 3) - 4y + 4 \qquad \Longrightarrow d_n = d + 4(x - y) + 10 \qquad \text{if } d > 0 \ (y_n = y - 1) \tag{8}$$

$$d_0 = 2(0 + 1)^2 + r^2 + (r - 1)^2 - 2r^2 \qquad \Longrightarrow d_0 = 3 - 2r \qquad \text{at initial point } (0, r) \tag{9}$$

2 Implementation

- Time Complexity: $O(r)$ where r is the radius of the circle
- Space Complexity: $O(1)$

```python
import matplotlib.pyplot as plt


def bresenham(radius: float) -> None:
    """Draws a circle using Bresenham's Circle Drawing Algorithm."""

    # Initial parameters
    x: int = 0
    y: float = radius
    d: float = 3 - 2 * radius

    # Consider the upper half of the first quadrant
    while x <= y:
        # Put eight points via eight-way symmetry
        plt.scatter([x, x, -x, -x, y, y, -y, -y], [y, -y, y, -y, x, -x, x, -x])

        # Make decisions and update parameters
        if d <= 0:
            d += 4 * x + 6
        else:
            d += 4 * (x - y) + 10
            y -= 1
        x += 1

    # Show the drawing
    plt.title("Circle Drawing Using Bresenham's Circle Drawing Algorithm")
    plt.show()
```

A Python Implementation of Bresenham's Circle Drawing Algorithm