

Bresenham's Circle Drawing Algorithm

David Oniani
onianidavid@gmail.com

October 30, 2022

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:

$$x^2 + y^2 = r^2 \quad (1)$$

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:

$$e(x+1, y) = (x+1)^2 + y^2 - r^2 \quad (2)$$

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

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

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

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

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

The difference is equal to the following:

$$d_n - d = 2(2x+3) + (y_n^2 - y^2) + ((y_n-1)^2 - (y-1)^2) \quad (6)$$

Finally, we have:

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

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

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

2 Implementation

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

```
1 import matplotlib.pyplot as plt
2
3
4 def bresenham(radius: float) -> None:
5     """Draws a circle using Bresenham's Circle Drawing Algorithm."""
6
7     # Initial parameters
8     x = 0
9     y = radius
10    d = 3 - 2 * radius
11
12    # Consider the upper half of the first quadrant
13    while x <= y:
14        # Put eight points via eight-way symmetry
15        plt.scatter([x, x, -x, -x, y, y, -y, -y], [y, -y, y, -y, x, -x, x, -x])
16
17        # Make decisions and update parameters
18        if d <= 0:
19            d += 4 * x + 6
20        else:
21            d += 4 * (x - y) + 10
22            y -= 1
23        x += 1
24
25    # Show the drawing
26    plt.title("Circle Drawing Using Bresenham's Circle Drawing Algorithm")
27    plt.show()
```

A Python Implementation of Bresenham's Circle Drawing Algorithm.