LINE DRAWING ALGORITHM

Text of the problem: Two pixels are displayed on a computer screen. Write a drawing algorithm for the line segment that joins these pixels on the screen. You can also solve similar problems, such as drawing a circle, when you know its center and radius.

Overview:

The basic "line drawing" algorithm used in computer graphics is Bresenham’s Algorithm. This algorithm was developed to draw lines on digital plotters, but has found wide-spread usage in computer graphics. The algorithm is fast – it can be implemented with integer calculations only – and very simple to describe.

Conventions:

- The bottom-left is (0,0) such that pixel coordinates increase in the right and up directions (e.g. that the pixel at (7,5) is directly above the pixel at (7,4)).
- The pixel centers have integer coordinates.

There are three cases that we need to take into consideration:

1. The angle between the line and the Ox axis is bigger than 45 degrees
2. The angle between the line and the Ox axis is 45 degrees
3. The angle between the line and the Ox axis is smaller than 45 degrees

When the angle is more than 45 degrees, y is always incremented, and x is not always incremented.

When the angle is 45 degrees, both x and y coordinates are incremented.

When the angle is less than 45 degrees, x is always incremented, and y is not always incremented.

We can find this out by dividing $\Delta y$ by $\Delta x$, (where $\Delta y$ is the difference between the y coordinates of the 2 pixels, $y_2-y_1$). If $m = \Delta y/\Delta x = 1$, the slope is 45 degrees. If $m = \Delta y/\Delta x > 1$, the slope is more than 45 degrees. If $m = \Delta y/\Delta x < 1$, the slope is less than 45 degrees.

Case 1: The slope is less than 45 degrees.
In this case, the $x$ values will be incremented at each step. We have to decide whether the $y$ value is incremented or not.

What we need to do now is finding out whether $d_1$ is bigger than $d_2$ or not.

The equation of the actual line is $y = m \cdot (x(k) + 1) + c$, where $c = y(1) - \Delta y/\Delta x \cdot x(1)$ (for simplicity)

Now, we replace $y$ with $m \cdot (x(k) + 1) + c$ in the $d_1$ and $d_2$ relations.

$d_1 = m \cdot (x(k) + 1) + c - y(k)$ and $d_2 = y(k) + 1 - m \cdot (x(k) + 1) - c$.

$d_1 - d_2 = [m \cdot (x(k) + 1) + c - y(k)] - [y(k) + 1 - m \cdot (x(k) + 1) - c]$

$d_1 - d_2 = 2m \cdot (x(k) + 1) - 2y(k) + 2c - 1$
However, \( m = \Delta y / \Delta x \), which is a float value, but we need integer numbers. To get integers, we are multiplying the relation with \( \Delta x \). The colour red means that those values are constant.

\[
\Delta x \cdot (d1 - d2) = \Delta x \left[ 2 \cdot \Delta y / \Delta x \cdot (x(k) + 1) - 2y(k) + 2c - 1 \right]
\]

You might think this expression is equal to:

\[
\Delta x \cdot (d1 - d2) = 2 \cdot \Delta y \cdot x(k) - 2 \cdot \Delta x \cdot y(k) + 2 \cdot \Delta y + 2 \cdot \Delta x \cdot c - \Delta x
\]

Let \( P(k) = \Delta x \cdot (d1 - d2) = 2 \cdot \Delta y \cdot x(k) - 2 \cdot \Delta x \cdot y(k) + 2 \cdot \Delta y + 2 \cdot \Delta x \cdot c - \Delta x \)

Because they are constant, they have no effect in our decision, so we can skip them. This way:

\[
P(k) = 2 \cdot \Delta y \cdot x(k) - 2 \cdot \Delta x \cdot y(k)
\]

If \( P(k) = 2 \cdot \Delta y \cdot x(k) - 2 \cdot \Delta x \cdot y(k) \), that means that \( P(k + 1) = P(next) = 2\Delta y \cdot x(next) - 2 \cdot \Delta x \cdot y(next) \).

To find out which pixel to colour, we need to subtract \( P(k) \) from \( P(next) \).

**1.1 If** \( P(k) < 0 \), \( P(k + 1) = P(k) + 2 \cdot \Delta y \)

**1.2 If** \( P(k) >= 0 \), \( P(k + 1) = P(k) + 2 \cdot \Delta y - \Delta x \)

**Algorithm:**

Algorithm Bres \((x1, x2, y1, y2)\)

{  
  \( x = x1; \)
  \( y = y1; \)
  \( dx = x2 - x1; \)
  \( dy = y2 - y1; \)
  \( P = 2dy \cdot dx; \)
  while(x \( \leq x2)\)
  {
    putpixel\( (x, y)\);
    \( x++; \)
    if\( P < 0 \) \)
    \( P = P + 2 \cdot dy; \)
  else
  {
    \( P = P + 2 \cdot dy - 2 \cdot dx; \)
    \( y++; \)
  
}
Example:
\[
x_1=1 \\
y_1=1 \\
x_2=8 \\
y_2=5
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>-3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
Case 2: The slope is greater than 45 degrees

This case is identical to the first one. It is just a matter of replacing the x and y variables (if in 1st case, \( P = 2 \cdot \Delta y - \Delta x \), in the 2nd case, \( P = 2 \cdot \Delta x - \Delta y \))

Example:

\[
\begin{align*}
x_1 &= 1 \\
y_1 &= 1 \\
x_2 &= 3 \\
y_2 &= 6
\end{align*}
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>-3</td>
</tr>
</tbody>
</table>
\[
\begin{array}{ccc}
3 & 4 & 1 \\
3 & 5 & -5 \\
3 & 6 & -1 \\
\end{array}
\]
Case 3: The slope is 45 degrees

Both x and y values are incremented every time, so we can use either of the other 2 cases, because P will always have the same value.
Example:

\[ x_1=1 \]
\[ y_1=1 \]
\[ x_2=4 \]
\[ y_2=4 \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
Circle drawing algorithm
Overview:

We will use Bresenham’s circle drawing algorithm. We will divide the circle in four quadrants, each of 90 degrees and further in 8 octants, each of 45 degrees. The algorithm is almost the same for all of the sectors.

Conventions:

Let initial x be $x(i)$ and initial y be $y(i)$.

First quadrant - top right;
Second quadrant - top left;
Third quadrant - bottom left;
Fourth quadrant - bottom right;
The line is drawn clockwise.

If a point is inside the circle, its distance from the circle will be considered negative and if it is outside, its distance from the circle will be considered positive.

Analysis:

We will have a closer look at the first quadrant, which is the top right one.
In this case, x will always be incremented, but we can’t tell the same thing about y. So, we have to decide whether to increment y or not.

We will have to choose between the \((x+1, y)\) pixel and the \((x+1, y-1)\) pixel. We can do that by finding the distances between each of the two pixels and the circle, which are \(d_1\) and \(d_2\).

\[d(x, y) = x^2 + y^2 - R^2\] . This is a parameter that we use to find the sign for the next incrementation.

We know that \(d_1\) is the distance between the \((x(k)+1, y(k))\) pixel and the circle and that \(d_2\) is the distance between the \((x(k)+1, y(k)-1)\) pixel and the circle so we can write them as:

\[d_1 = (x(k) + 1)^2 + y(k)^2 - R^2\]
\[d_2 = (x(k) + 1)^2 + (y(k) - 1)^2 - R^2\]

We now need to decide which pixel is closer to the circle and for that we will need a decision variable, which we will name \(P(k)\).

\[P(k) = d_1 + d_2\]

\[P(k) = 2(x(k) + 1)^2 + y(k)^2 + (y(k) - 1)^2 - 2R^2\]

\[P(k+1) = 2(x(k) + 2)^2 + y(k+1)^2 + (y(k + 1) - 1)^2 - 2R^2\]

We need to write \(P(k+1)\) in relation to \(P(k)\).

\[P(k+1) = P(k) + 2(2 \cdot x(k) + 3) + (y(k + 1) + y(k)) \cdot (y(k + 1) - y(k)) + (y(k) - 2 + y(k) - 1) \cdot (y(k) - 2 - y(k) + 1)).\]

\(d_1\) will be considered positive, as the \((x(k) + 1, y(k))\) pixel is outside the circle and \(d_2\) negative, because the \((x(k) + 1, y(k) - 1)\) pixel is inside the circle.

\[P(k) = d_1 + d_2\]

So, if \(P(k) <= 0\), it means that \(d_2\) is greater than \(d_1\), and the next pixel will be \((x(k) + 1, y(k) - 1)\).

\[x(k + 1) = x(k) + 1\] and \(y(k + 1) = y(k)\).

We now need to replace these values.

That gives: \[P(k+1) = P(k) + 2(2x(k) + 3) = P(k) + 4 \cdot x(k) + 6.\]
If \( P(k) > 0 \), it means that \( d1 \) is greater than \( d2 \), and the next pixel will be \((x(k) + 1, y(k))\).

\[
x(k + 1) = x(k) + 1; \quad y(k + 1) = y(k) - 1.
\]

We now need to replace these values.

That gives: \( P(k + 1) = P(k) + 4(x(k) - y(k)) + 10 \).

The first pixel has the coordinates \((0, R)\), \( \Rightarrow P(0) = 3 - 2R \).

**Algorithm:**

\[
\begin{align*}
x1 &= 0; \\
y1 &= r; \\
P(0) &= 3 - 2*r;
\end{align*}
\]

\[
\text{while}(x>=y)
\]
\[
\{ \\
\quad \text{if} \ (P(k) <= 0) \\
\quad \quad \{ \\
\quad \quad \quad x(k) + 1 = x(k) + 1 \\
\quad \quad \quad y(k) + 1 = y(k) \\
\quad \quad \quad P(k) + 1 = P(k) + 4 \cdot x(k) + 1 + 6 \\
\quad \quad \} \\
\quad \} \\
\]

\[
\text{if} \ (P(k) > 0) \\
\quad \{ \\
\quad \quad x(k) + 1 = x(k) + 1 \\
\quad \quad y(k) + 1 = y(k) - 1 \\
\quad \quad P(k) + 1 = P(k) + 4(x(k) + 1 - y(k) + 1) + 10 \\
\quad \} \\
\]

**Example:**

Let's take a circle with radius 4 and the centre at \((0,0)\).

This is the table for the first octant:

<table>
<thead>
<tr>
<th>x(i)</th>
<th>y(i)</th>
<th>P(k)</th>
</tr>
</thead>
</table>

17
Now, we will calculate the coordinates of the second octant by swapping the x and y coordinates.

<table>
<thead>
<tr>
<th>First octant</th>
<th>Second octant</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 4)</td>
<td>(3, 3)</td>
</tr>
<tr>
<td>(1, 4)</td>
<td>(3, 2)</td>
</tr>
</tbody>
</table>
So, the first quadrant (90 degrees), will be:

<table>
<thead>
<tr>
<th>First quadrant</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 4)</td>
<td></td>
</tr>
<tr>
<td>(1, 4)</td>
<td></td>
</tr>
<tr>
<td>(2, 3)</td>
<td></td>
</tr>
<tr>
<td>(3, 3)</td>
<td></td>
</tr>
<tr>
<td>(3, 2)</td>
<td></td>
</tr>
<tr>
<td>(4, 1)</td>
<td></td>
</tr>
<tr>
<td>(4, 0)</td>
<td></td>
</tr>
</tbody>
</table>
Now, we can calculate all of the circle’s points, using the first quadrant:

<table>
<thead>
<tr>
<th>Quadrant 1 (x, y)</th>
<th>Quadrant 2 (-x, y)</th>
<th>Quadrant 3 (-x, -y)</th>
<th>Quadrant 4 (x, -y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 4)</td>
<td>(0, 4)</td>
<td>(0, -4)</td>
<td>(0, -4)</td>
</tr>
<tr>
<td>(1, 4)</td>
<td>(-1, 4)</td>
<td>(-1, -4)</td>
<td>(1, -4)</td>
</tr>
<tr>
<td>(2, 3)</td>
<td>(-2, 3)</td>
<td>(-2, -3)</td>
<td>(2, -3)</td>
</tr>
<tr>
<td>(3, 3)</td>
<td>(-3, 3)</td>
<td>(-3, -3)</td>
<td>(3, -3)</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>(-3, 2)</td>
<td>(-3, -2)</td>
<td>(3, -2)</td>
</tr>
<tr>
<td>(4, 1)</td>
<td>(-4, 1)</td>
<td>(-4, -1)</td>
<td>(4, -1)</td>
</tr>
<tr>
<td>(4, 0)</td>
<td>(-4, 0)</td>
<td>(-4, -0)</td>
<td>(4, 0)</td>
</tr>
</tbody>
</table>