Code journal: polar coordinates also draw lines

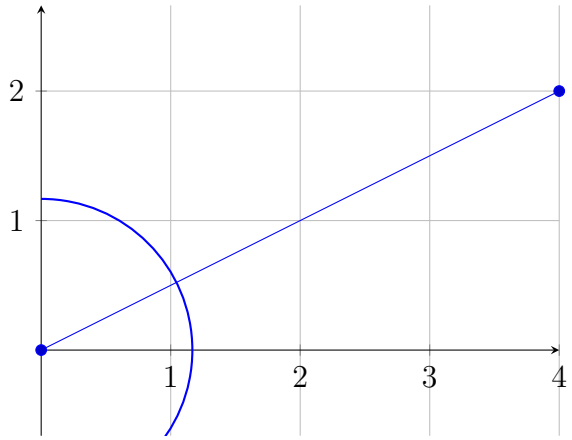
22.06.2020
Mariana Ávalos Arce

1 Line between two points

If I know that polar coordinates are defined as:

$$\begin{aligned} x &= rcos(t) \\ y &= rsin(t) \end{aligned} \tag{1}$$

Then I remembered also that a line is a series of points, we the **radius** is growing but the **angle** is constant. Therefore, for drawing a line I needed to know these two things: radius and an angle between two given points, let's say (0, 0) and (4, 2).



Then I said to myself: let's take the angle between two points with `angle = arctan(y/x)` as if there was an imaginary Pythagorean triangle between them. For the radius, I figured that the distance of the vector that unites these two points will do the job. Then, my `for` loop would simply go from 0 to that calculated distance, if the count starts from the **first point**. The idea would be the following:

```
dist = int(distance(p1, p2))
for r in range(dist):
    x = p1[0] + r * math.cos(rads)
    y = p1[1] + r * math.sin(rads)
    img[y, x] = color
```

I also noticed that when the line's x values are ordered there is no problem with the angle. However, when the x values of first point are larger than the second's, I needed to add to the angle the value of π, probably because of the fact that computers show the angle that is closer to zero in every case. The output when I finished the program was:

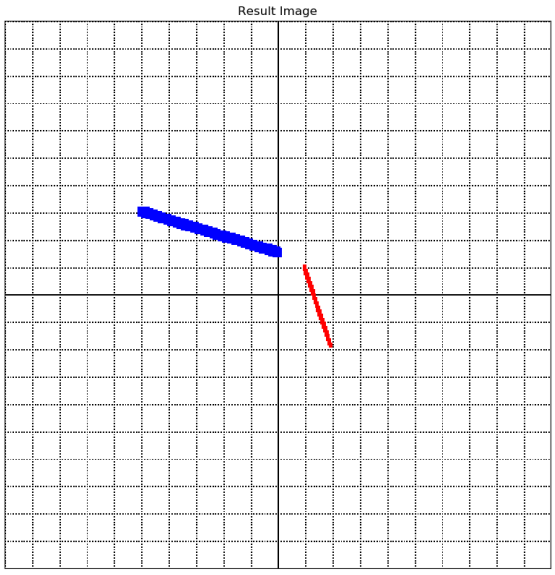


Figure 1: Drawing two lines with the program

For the thickness, I basically did a squared-window loop that went from `-thickness/2` to `thickness/2` around each pixel that was drawn. I had fun.