Program Exit Assessment

Sabrina Tochkov

Math for Computer Science

Hand-In #4 (Corrected)

DDA

Step 1

$$\Delta x = 14 - 9 = 5$$

$$\Delta y = 22 - 18 = 4$$

Step 2

$$M = \frac{\Delta y}{\Delta x} = \frac{4}{5} = 0.8$$

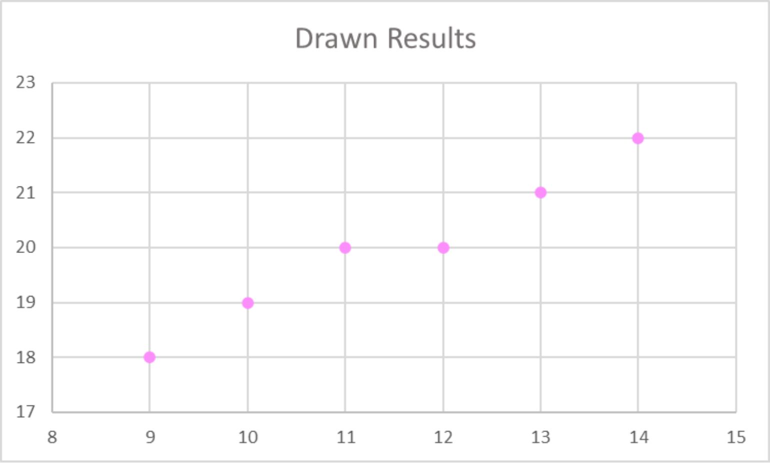
Step 3

$$since\ M = 0.8 < 1$$

$$x_{p+1} = 1 + x_p$$

$$y_{p+1} = M + y_p$$

n	x_n	y_n	graphed point
0	9	18	(9, 18)
1	10	18.8	(10, 19)
2	11	19.6	(11, 20)
3	12	20.4	(12, 20)
4	13	21.2	(13, 21)
5	14	22	(14, 22)



Simple Bresenham Line Drawing Algorithm

Step 1

$$\Delta x = 14 - 9 = 5$$
$$\Delta y = 22 - 18 = 4$$

Step 2

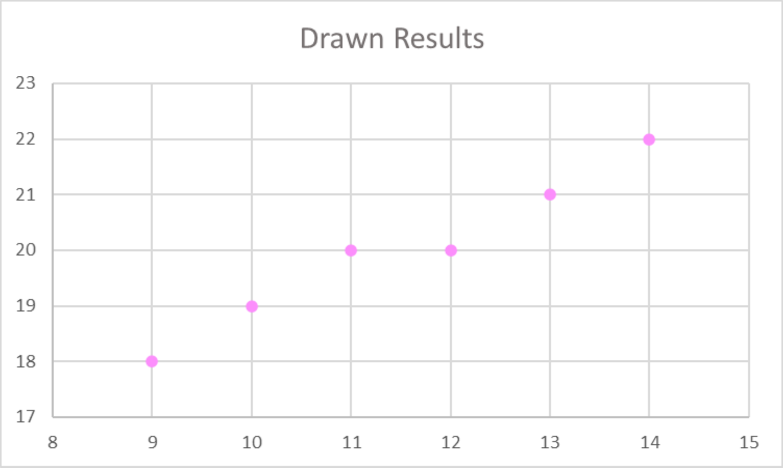
$$P_k = 2\Delta y - \Delta x = 2(4) - (5) = 3$$

Step 3

$$since\ P_k = 3 > 0$$
$$x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k + 1$$

$$when\ P_k < 0$$
$$x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k$$

n	P_k	x_n	y_n	graphed point
0	3	9	18	(9, 18)
1	1	10	19	(10, 19)
2	-1	11	20	(11, 20)
3	7	12	20	(12, 20)
4	5	13	21	(13, 21)
5	3	14	22	(14, 22)



Advantages:

- The x_{k+1} and y_{k+1} formulas are simpler than the DDA ones, because they only involve integer arithmetic.
- It is faster than the DDA algorithm, since there is no rounding necessary! There are no floating-point calculations.
- Points drawn are more accurate than in the DDA algorithm (in general).

Disadvantages:

- Does not prevent anti-aliasing (so jagged lines will appear instead of smoother ones when we zoom in on images).
- It can only handle *simple* line drawings (hence the name), nothing too complex.
- This algorithm cannot handle downsizing (making images smaller), nor create a compression artifact (visibly reducing the image quality to save space).

Thought Process

Solving this problem was crystal clear. The algorithms and steps were well-defined, which allowed me to start working on the hand-in right away. Some additional research was necessary in order to identify the differences between the DDA and Simple Bresenham Line Drawing Algorithms. The GateVidyalay website (and a few others) helped me better understand the pros and cons of each. The procedure itself felt fairly familiar, since I have all the necessary foundational precepts to work through it. For example, calculating slopes has already become second nature to me as a high school math tutor. I made use of Excel's scatterplot graphing (which I used for the previous hand-in) to display the drawn results of each line drawing algorithm. In my opinion, this was the most intuitive problem to solve out of the 4 we've done this semester.

Computer Science Field: Graphics Developer / Graphics Programmer

A **Graphics Developer** is responsible for creating software for visual objects and works in many different work settings, like effects studios, video game companies, and even robotics companies. According to Christa Terry[1], editor for the educational website *Noodle*, Graphics Developers design applications, games and videos by using a variety of computer graphics skills: rendering, lighting, shading, shadowing, etc.

At the root of digitally rendering an image are **algorithms** that determine which pixels of the screen will light up and which ones won't. These roots go for miles… all the way back to the simplest line-drawing algorithms like the DDA and the Simple Bresenham Line Drawing Algorithm. To generate these complex and photorealistic 2D and 3D images, Graphics Developers must use programs that are built upon the foundation of line-drawing algorithms. Working as a Graphics Developer involves a lot of high-level math, like Linear Algebra and Calculus. Becoming a worker in this field requires an understanding of calculating coordinates of figures, images and objects in 2D/3D space. It also requires understanding how to optimize graphical features and image quality. Over the years, these simple line-drawing algorithms were improved and refined to accomplish this optimization.

For example, in order to display an image, Xiaolin Wu created a line-drawing algorithm which is better than the Bresenham one because it generates anti-aliased lines, according to GeeksForGeeks[2]. These kinds of algorithms make up many softwares like Adobe Flash and Unity3D, which are used by Graphics Programmers.

Therefore, a good grasp of these building blocks is necessary to be part of the field of Graphics Development.

[1] TERRY, Christa. *How to Become a Graphics Developer and Bring Game Concepts to Life.* URL: https://www.noodle.com/articles/how-to-become-a-graphics-developer-and-bring-game-concepts-to-life#:~:text=A%20graphics%20 developer%20

[2] *Anti-aliased Line | Xiaolin Wu's algorithm.* URL : https://www.geeksforgeeks.org/anti-aliased-line-xiaolin-wus-algorithm/