Inf-1100, Assignment 2

Submission: 23:59 September 23rd, 2022

Overview

In this assignment, you will try to formulate an algorithm (or two) to draw a line on a computer screen. We encourage you to discuss the assignment with other students, and to collaborate with another person in order to implement two algorithms. Note that this assignment is **not** mandatory.

Description

An image shown on a computer screen is a collection of colored dots/squares (pixels). The computer screen itself is organized as a two-dimensional coordinate system with a number of pixels in the horizontal (x) and vertical (y) direction. The more pixels a screen can handle, the more detailed can be displayed in an image. For example, a high definition screen (HD) has 1920 pixels in the x direction and 1080 pixels in the y direction. In contrast, a screen only capable of showing DVD quality images will have 720 pixels in the x direction, and 576 in the y direction.

This assignment is divided into four steps for each algorithm:

1. Given the start and end coordinates of a **horizontal** line, formulate an algorithm to compute the coordinates of the pixels along the line. Example coordinates: (10,10), (100,10) or (200,100), (100,100).

2. Given the start and end coordinates of a **vertical** line, formulate an algorithm to compute the coordinates of the pixels along the line. Example coordinates: (100,10), (100,200) or (200,50), (200,250).

3. Given the start and end coordinates of a line with a **45-degree slope**, formulate an algorithm to compute the coordinates of the pixels along the line. Example coordinates: (0,0), (100,100) or (100,100), (200,0).

4. Given the start and end coordinates of a line, formulate an algorithm to compute the coordinates of the pixels along the line.

We encourage you to try to implement the algorithms in C code. To this end we provide you with code to open a window on the computer screen and code to set a pixel in the window. In the provided code, the window is set up with 1024 pixels horizontally (the x-axis) and 768 pixels vertically (the y-axis). The point 0,0 is in the upper left corner of the window. 0, 767 is in the lower left corner, 1023,0 is in the upper right corner, and 1023,767 is in the lower right corner. Each pixel is represented as a 32-bit value encoding the color of the pixel (in RBG format, see RGB).

To open a window on the screen we use a library called Simple Directmedia Layer (SDL). More information on SDL can be found here. Do not worry too much about SDL for this assignment. All the SDL "magic" is handled by the code provided by us.

C solution

If you decide to implement your algorithm, your starting point is the following set of files:

- drawline.h - Specifies the interface of the drawline functions. Do not modify this file.
- a_drawline.c - An empty stub for the first drawline function (complete this).
- b_drawline.c - An empty stub for the second drawline function (complete this).
- Program.c – A program for opening a window and drawing some lines using the two drawline functions (expand this to test your algorithms).
- Makefile - A Makefile for compiling the code.

We've bundled these files in a zip file. The zip file is in the same folder as this document. Use this zip file as a starting point, as it also includes the necessary SDL files.

Submission

This assignment is not mandatory. However, we highly recommend you complete it. You will receive feedback on your code if you choose to submit your solution. No report is required, and you will not receive feedback on any submitted report.

Hint

A good starting point for finding different line drawing algorithms is https://en.wikipedia.org/wiki/Line_drawing_algorithm.