Homework #1: Line Drawing

Due Date: 11:59pm Monday 30 September 2013

Write a C++ program that implements the Midpoint line drawing algorithm. The program should read in a set of polylines as on stdin (standard input), and outputs on stdout a PPM image file containing the rasterized lines.

Input

The input to the program consists of the following format:

```
polyline
     0.0 0.0
     1.0 0.0
     1.0 1.0
     0.0 1.0
polyline
     3.0 -3.5
     7.0 5.0
```

where the program should start at the first point and then draw lines to the next point, then the following point and so on.

The program takes in the following command line arguments:

```
draw2d [xmin] [xmax] [ymin] [ymax] [xres] [yres]
```

where:

- xmin, xmax, ymin, ymax: the range of x and y values
- xres, yres: the size of the output image

Output

The output is a PPM image file, which has the following format:

```
P3
[xres] [yres]
[max intensity]
```

```
[r0] [g0] [b0]
[r1] [g1] [b1]
….
```

where (r0, g0, b0) is the color of the top-left pixel, and pixels go from the top-left towards the bottom right.

For example:

```
P3
2 2
255

255 255 255
0 0 0
0 0 0
255 255 255
```

represents a 2x2 image (with 4 pixels) where the top-left pixel is white, the top-right is black, the bottom-left is black, and the bottom-right is white.

To run on a file lines.txt and view the output:

```
draw2d -1.5 1.5 -1.5 1.5 200 200 < lines.txt | display -
```

Program Flow

- To convert from the input point coordinates to pixels, you have to make use of the extent of the raster i.e. [xmin, xmax, ymin, ymax] and of the size of the raster i.e. [xres, yres]. For example, Given a point (x, y), to find the pixel where this point lies, we need to perform something like this:

$$i = \left\lfloor \frac{x - xmin}{xmax - xmin} \times (xres - 0.5) \right\rfloor$$

  where i is the horizontal dimension of the pixel, and the same to find j. These coordinates are then fed into the function below.

- You need to write a function that implements the Midpoint Algorithm for line drawing. This function will take as input the start and end *pixels* of the line and should determine which pixels

belong to the line.

- To keep record of the pixels that are lit i.e. that belong to any of the lines, you need to maintain some kind of a *raster*, or 2D array, with dimensions $xres \times yres$ that contain all the pixels. Then, a reference of this object should be passed to the line drawing function to record when a pixel should be lit or not.
- Finally, after drawing all the lines required by the input, you should iterate on the array of pixels in the raster in row-major order and output the pixels in the format required by the PPM file.

Instructions

- All code should be implemented in C++ under Linux.
- Please submit your homework in one zip file named as follows: *HW##.FirstName.LastName.zip*, so for example if your name is Mohamed Aly and this is homework #1, then the file name should be *HW01.Mohamed.Aly.zip*.
- Please include all your code and sample output in the zip file, with a README file to explain what you did. Failure to follow these instructions will cause deductions from your grade.
- You are allowed to discuss the problems among yourselves. However, **copying** any part of the code will result a grade of **ZERO**. No exceptions.

Acknowledgment

This homework is adapted from CS 171 at Caltech.