 Computer Graphics – Mastery Assignment 1

# Objective

You will be creating your first primitive types in 2D space. We will be working mostly with points and lines. At this point you must have a strong understanding of how the first lab worked and you will also be expected to already have a few functions written to help with today’s lab. The most important of the functions will be:

* Clearing the back buffer to a certain color
* Drawing a pixel a certain color at a specific position

Today’s proof of concept will be to implement three line algorithms taught in lecture. The three line algorithms are:

* Bresenham
* Midpoint
* Parametric

You will be expected to make these algorithms into function calls because you ***WILL*** be calling them multiple times. It will also make it easier to debug your code ***AND*** assist a lab specialist with helping you in debugging your code. A term you will see repeatedly used **“Positive Slope Line”** refers to the algorithm only working with lines that slope in a positive direction in both x and y. (E.G. The line’s starting x value is lower than the ending x value AND the starting y value is lower than the ending y value).

The position for the Bresenham line is as follows:

* Line Start x0, y100
* Line End x499, y400

# Grading Breakdown

* 25% - Starfield
  + Draw approximately 2000-2500 “stars” for the background
  + Color them white
* 50% - Draw three lines using the three line algorithms
  + Bresenham points are given to you. Color the line red.
  + Midpoint line will be drawn 10 units below the Bresenham. Color the line green.
  + Parametric line will be drawn 10 units below the Midpoint. Interpolate the color from magenta to cyan.
* 75% - Randomized Position
  + Choose any line algorithm
  + Color is dependent on the line algorithm
  + Draw the start and end points as yellow pixels AFTER the line is drawn so we know the algorithm is working correctly.
  + Have a key press to randomize both start and end positions of the line.
  + Your algorithm must be robust enough to handle all octants.
* 100% - Randomized positions with all three line algorithms.
  + Similar to the previous checkpoint, use the next two line algorithms and randomize their start and end positions on button presses as well.
  + Color is dependent on the line algorithm. Stay consistent.
  + Don’t forget to mark the start & end points with yellow pixels.

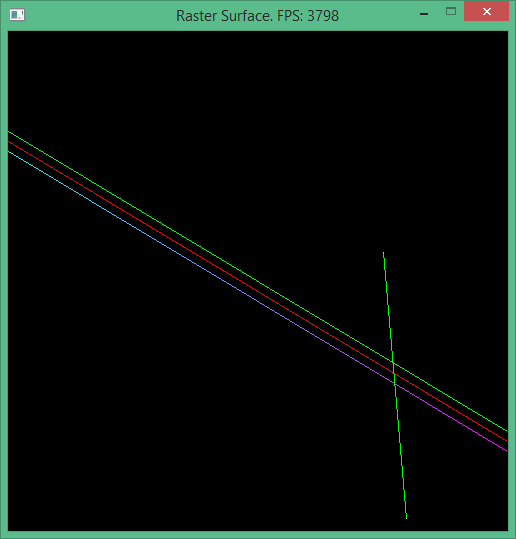
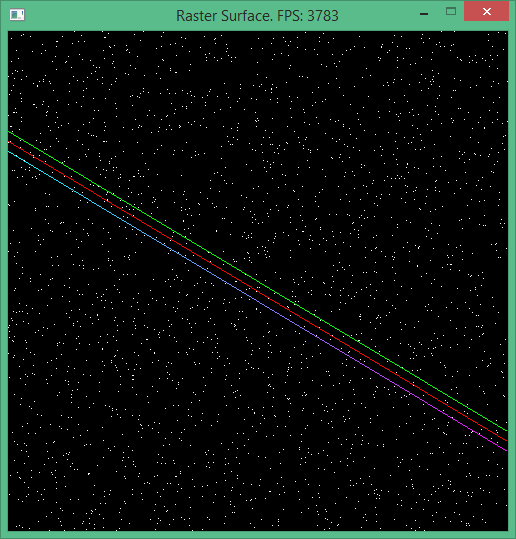
# Additional Challenge

1. Adapt all three line drawing algorithms so there is no more than a ***single loop*** in each.
2. Modify the Bresenham & Midpoint algorithms so they never use any floats, only integers. The book has some very good advice on how to do this!

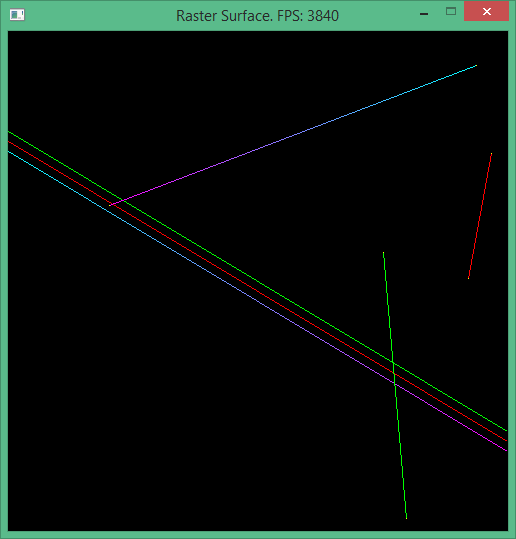
# Setup

# Example Images

50% 75%



100%



# Frequently Asked Questions

**Q**: My midpoint line is not working properly, it says I am above the line when I am actually below it.

**A**: The implicit line equation is based on the gradient, the direction of travel affects what “up” is.