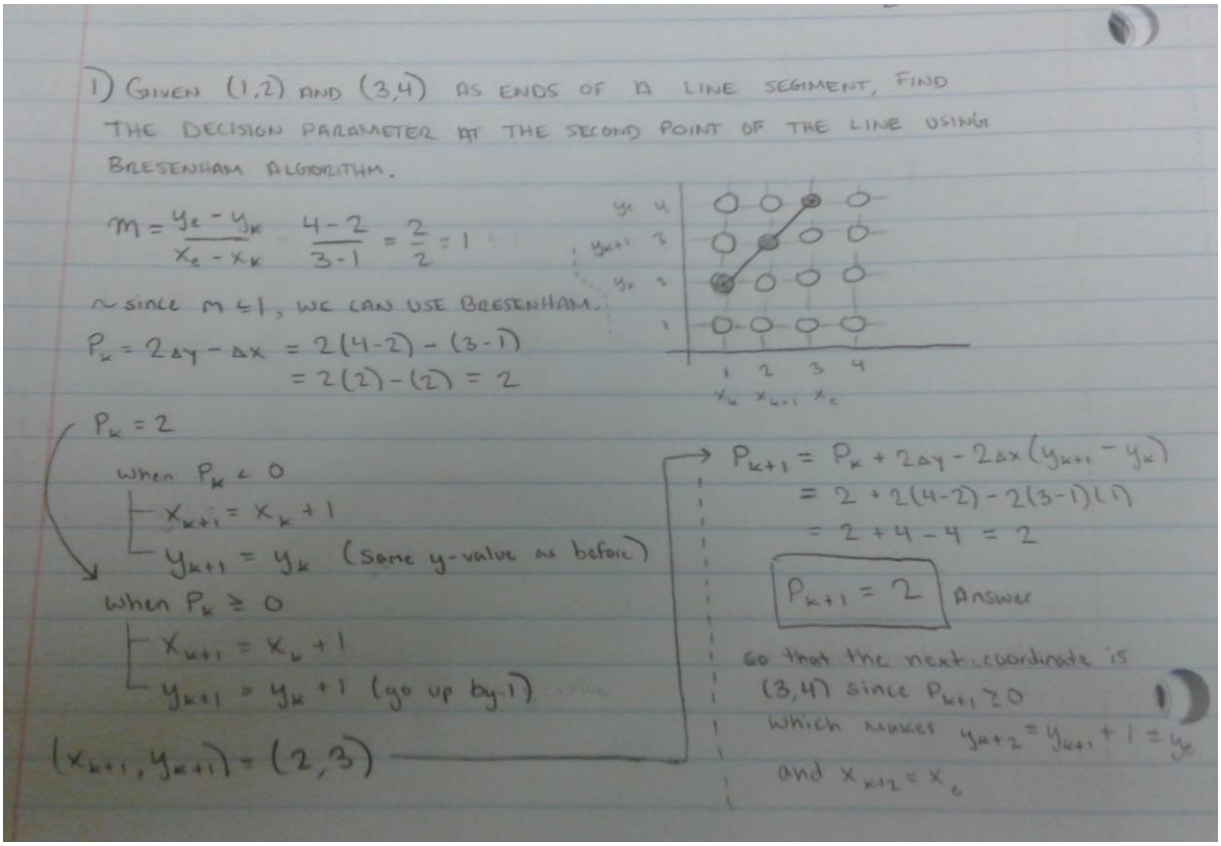
Alexander M. Nicoara
CS 411: Computer Graphics
Professor Gady Agam
Assignment #1: Raster Graphics

Introduction

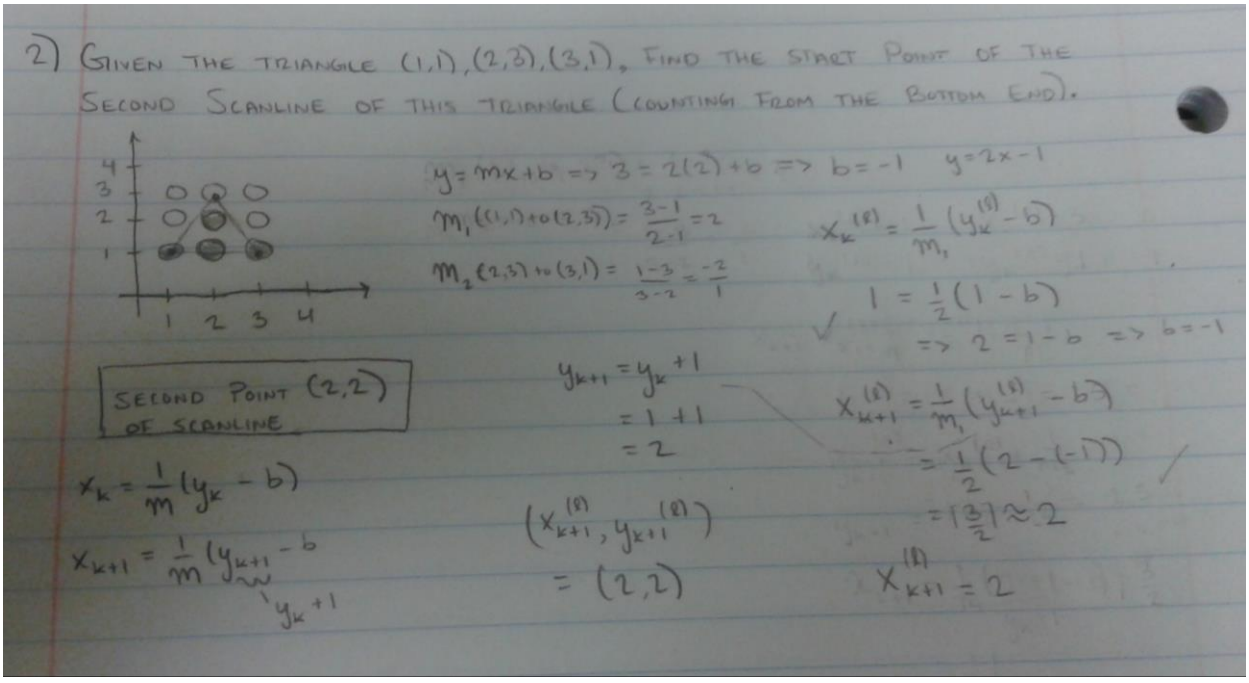
The goal of this assignment was to learn, familiarize ourself with, and apply practices pertaining to raster graphics. The first part of this assignment deals with using algorithms/methods to solve written problems, while the second part is applying them by programming them visually using HTML and JavaScript.

Questions

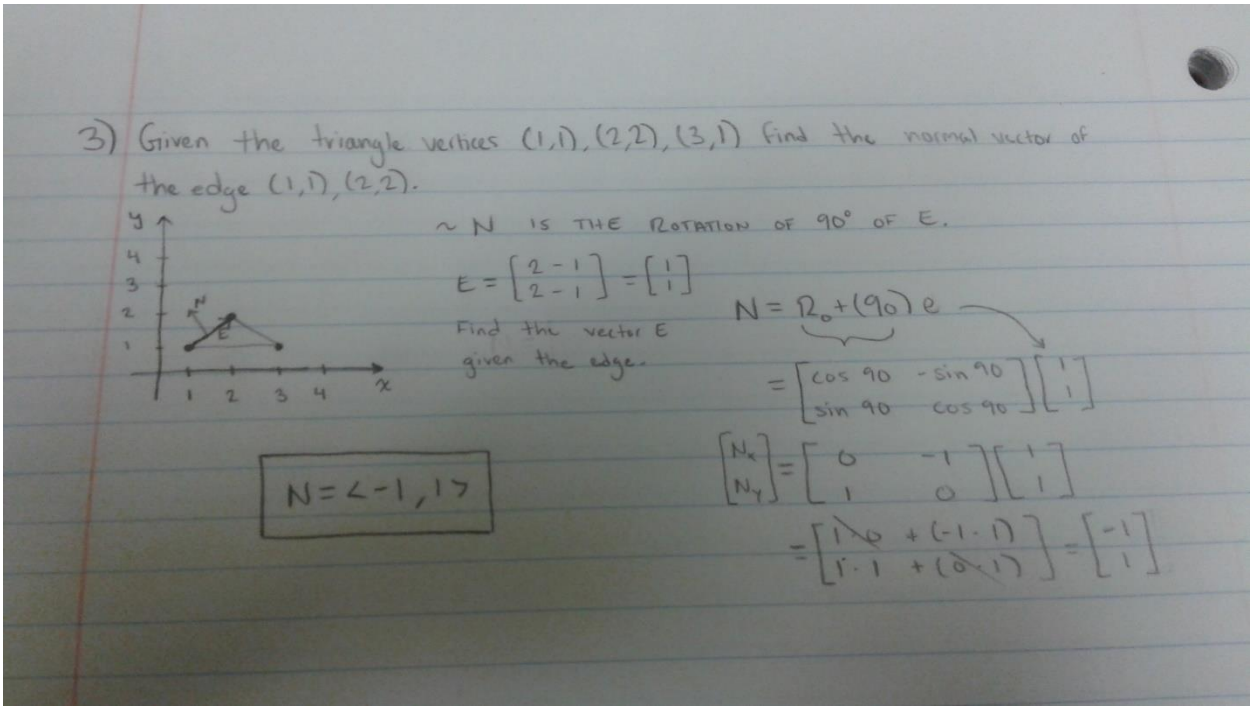
#1) Let (1,2) and (3,4) be the two endpoints of a discrete line segment. Find the value of the decision parameter at the second point of the line (the one after (1,2)) when using the Bresenham line algorithm.



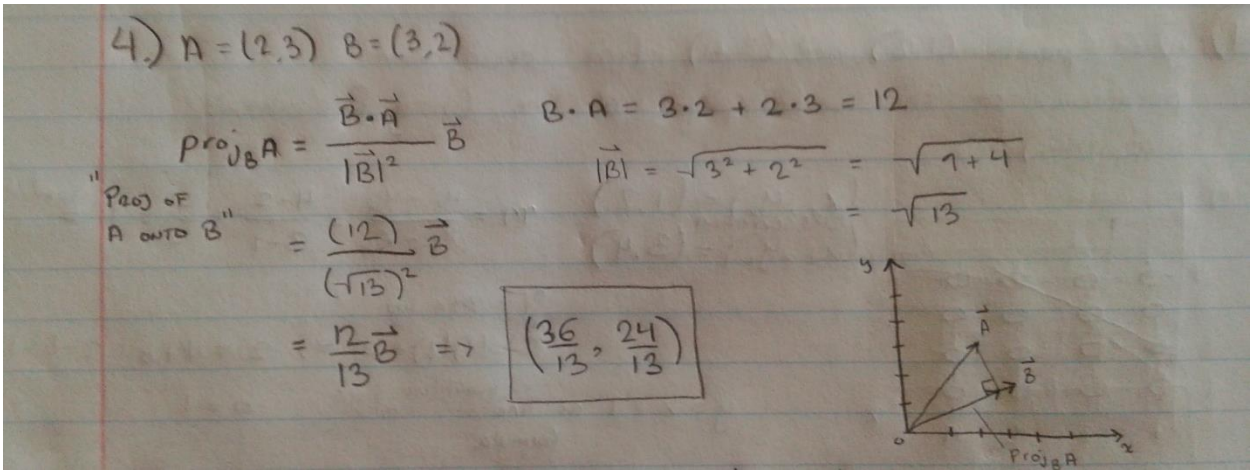
#2) Given the triangle vertices (1,1) (2,3) (3,1) find the start point of the second scanline of this triangle (counting from the bottom end).



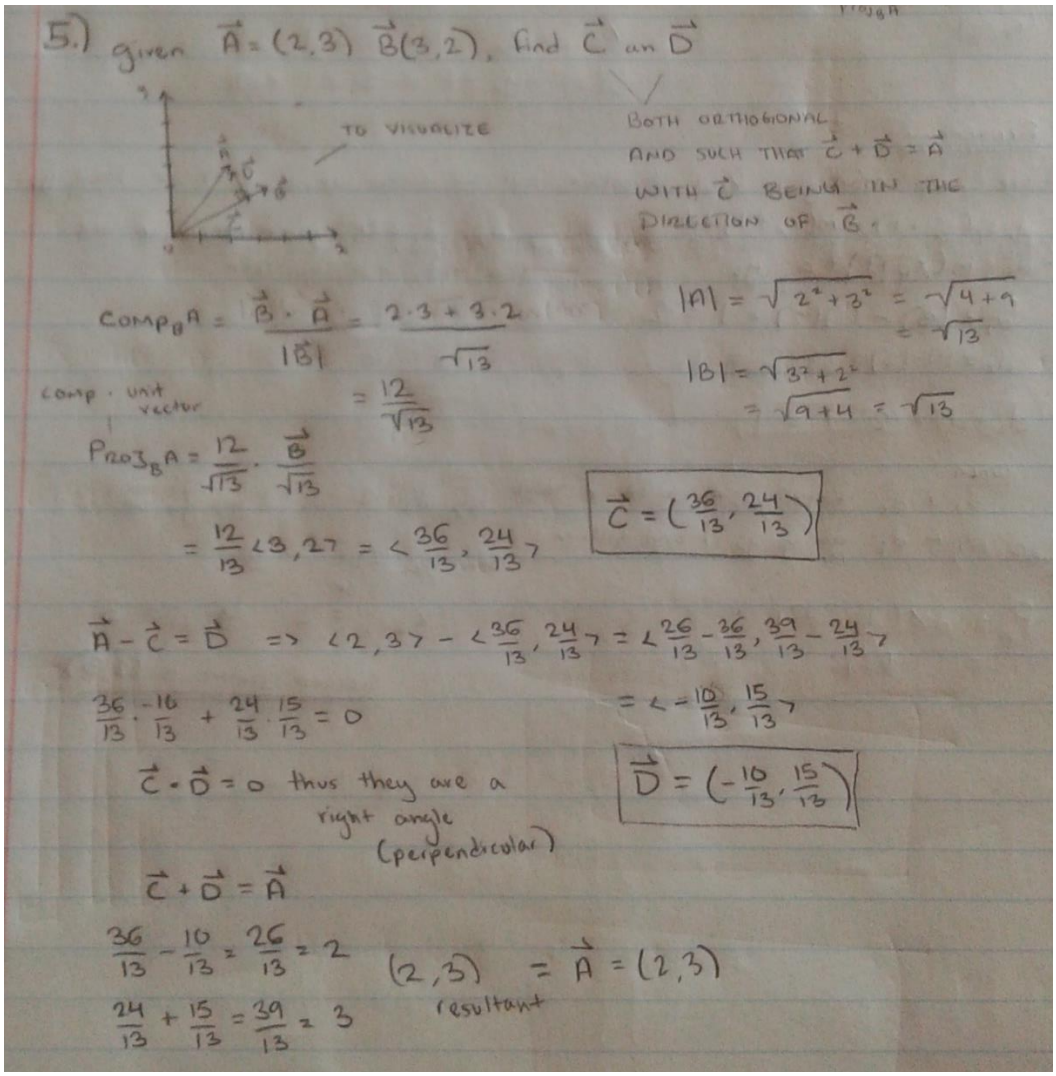
#3) Given the triangle vertices (1,1) (2,2)(3,1) find the normal vector of the edge (1,1)(2,2).



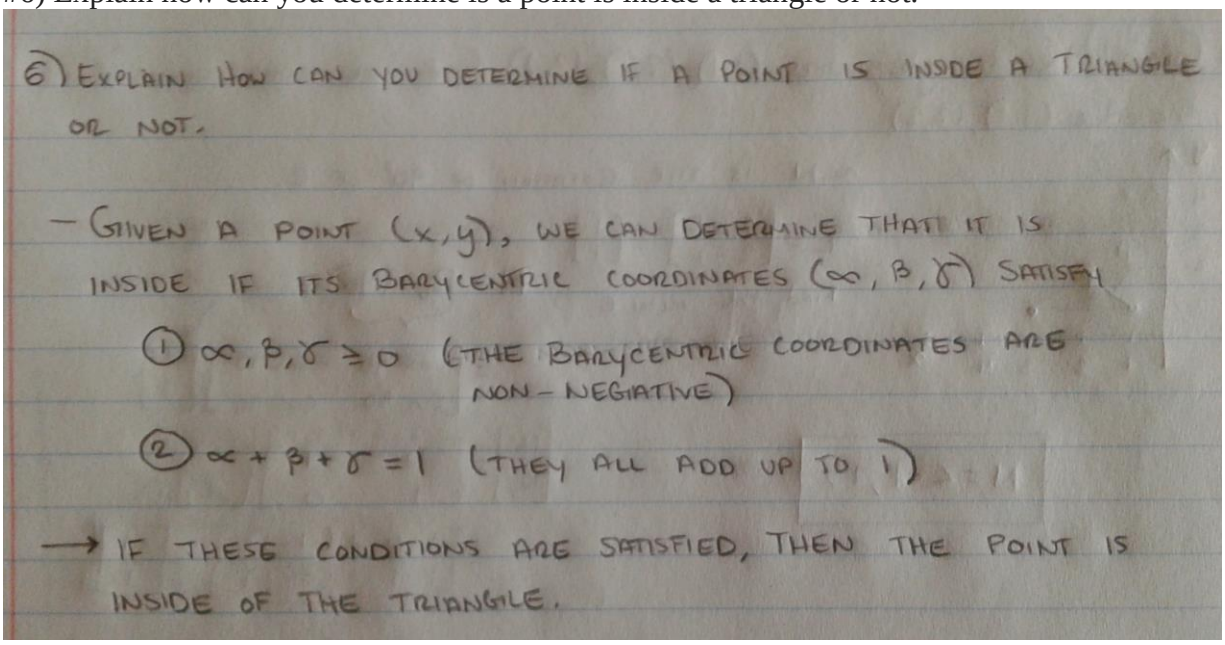
#4) Given the vectors A=(2,3) and B=(3,2) find the projection of the vector A onto the vector B.



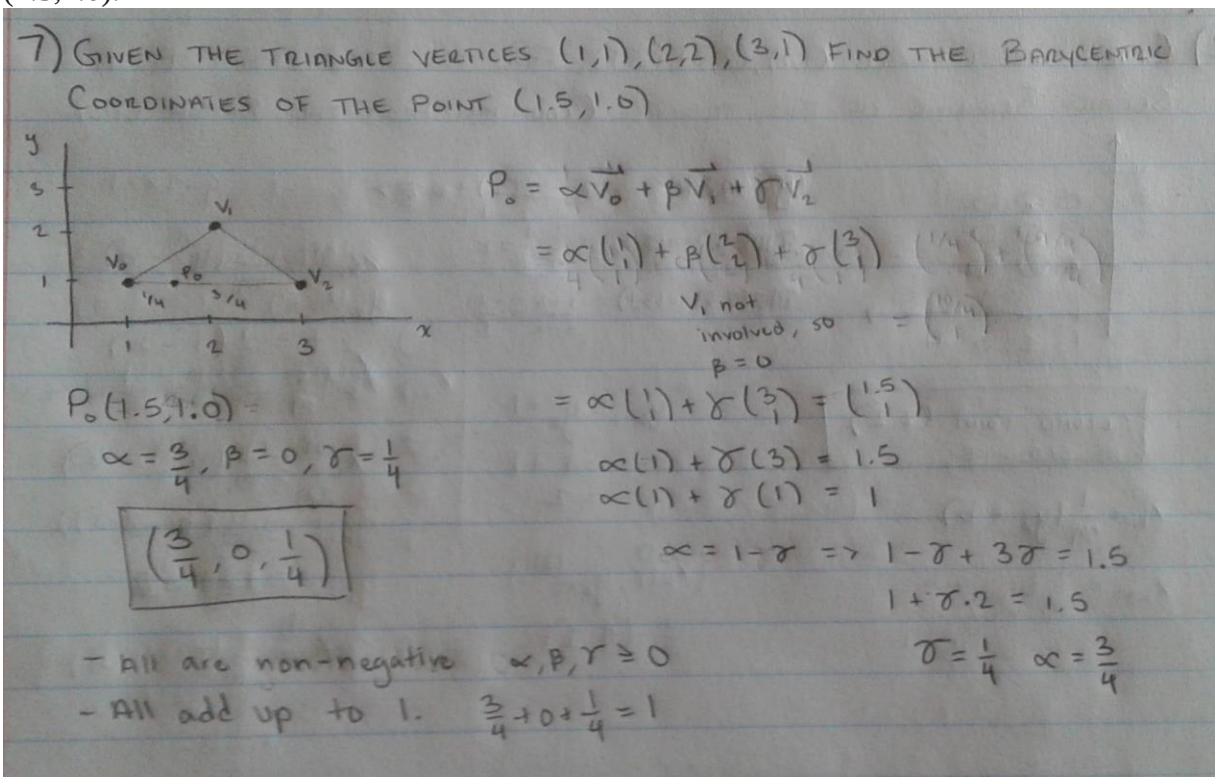
#5) Given the vectors A and B from the previous question, find two perpendicular vectors C and D such that C+D=A and the vector C is in the direction of B.



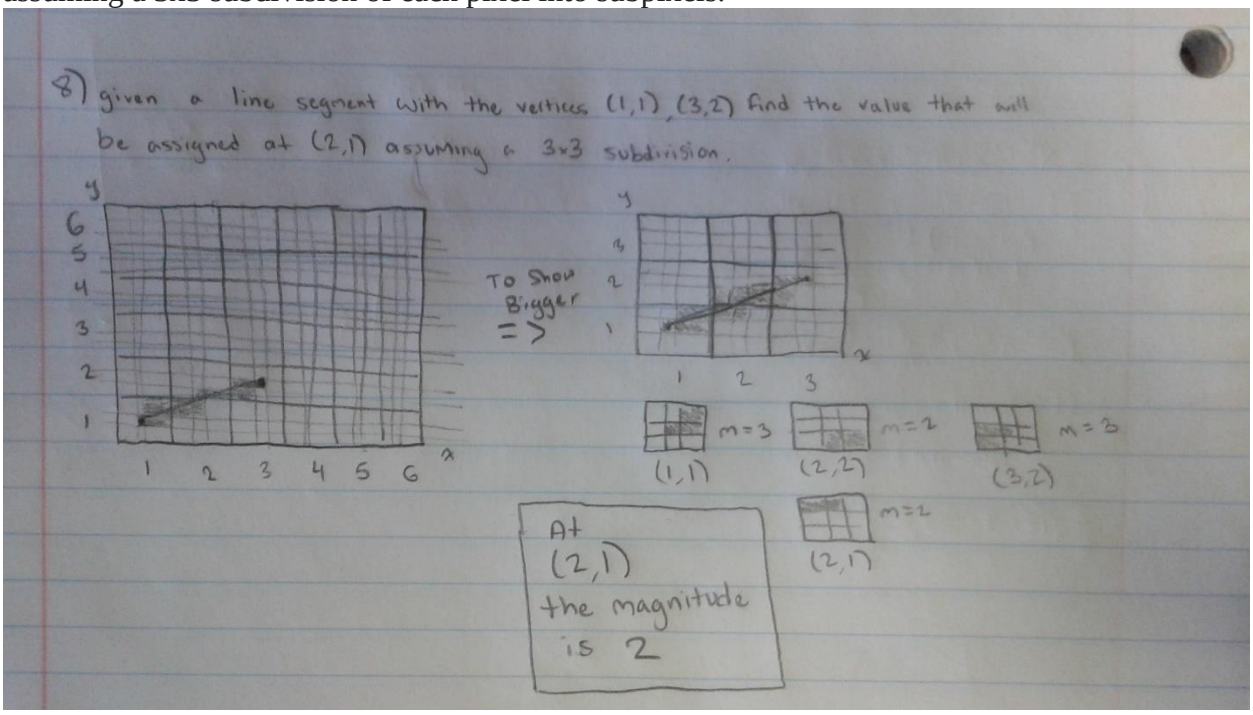
#6) Explain how can you determine is a point is inside a triangle or not.



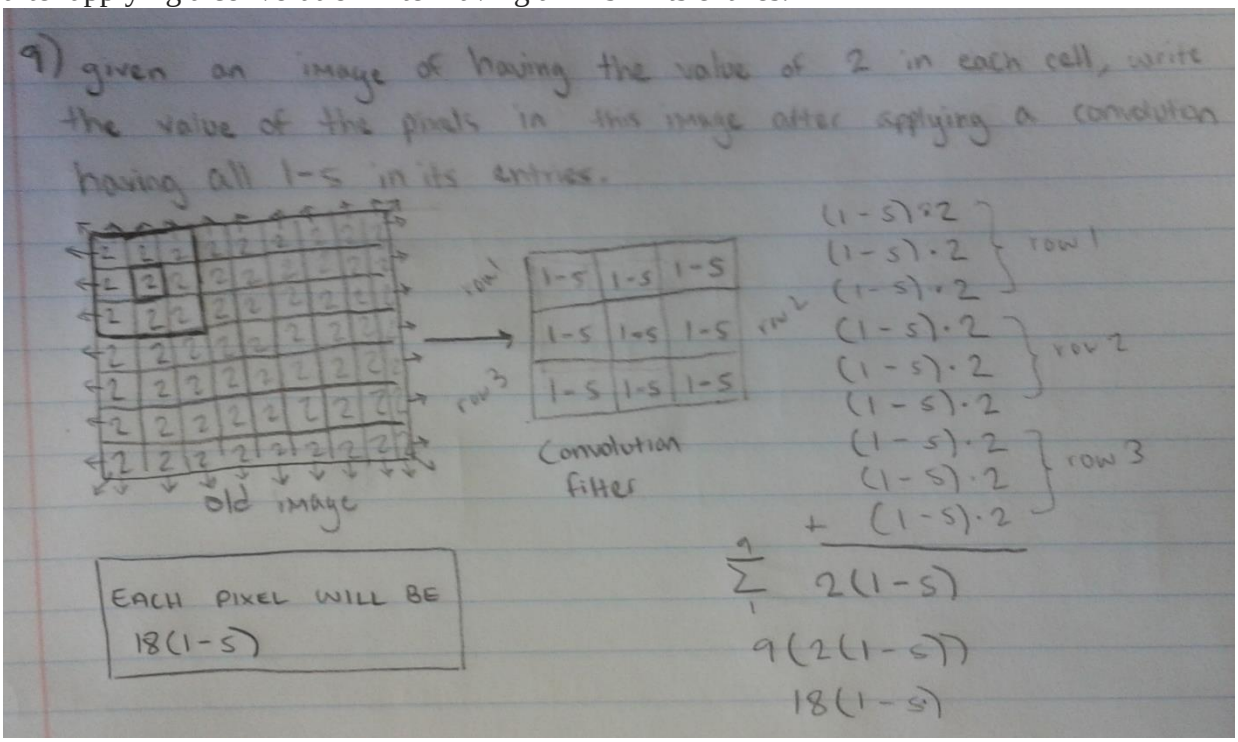
#7) Given the triangle vertices (1,1) (2,2)(3,1) find the barycentric coordinates of the point (1.5,1.0).



#8) Given a line segment with vertices (1,1) (3,2) find the value that will be assigned at (2,1) assuming a 3x3 subdivision of each pixel into subpixels.

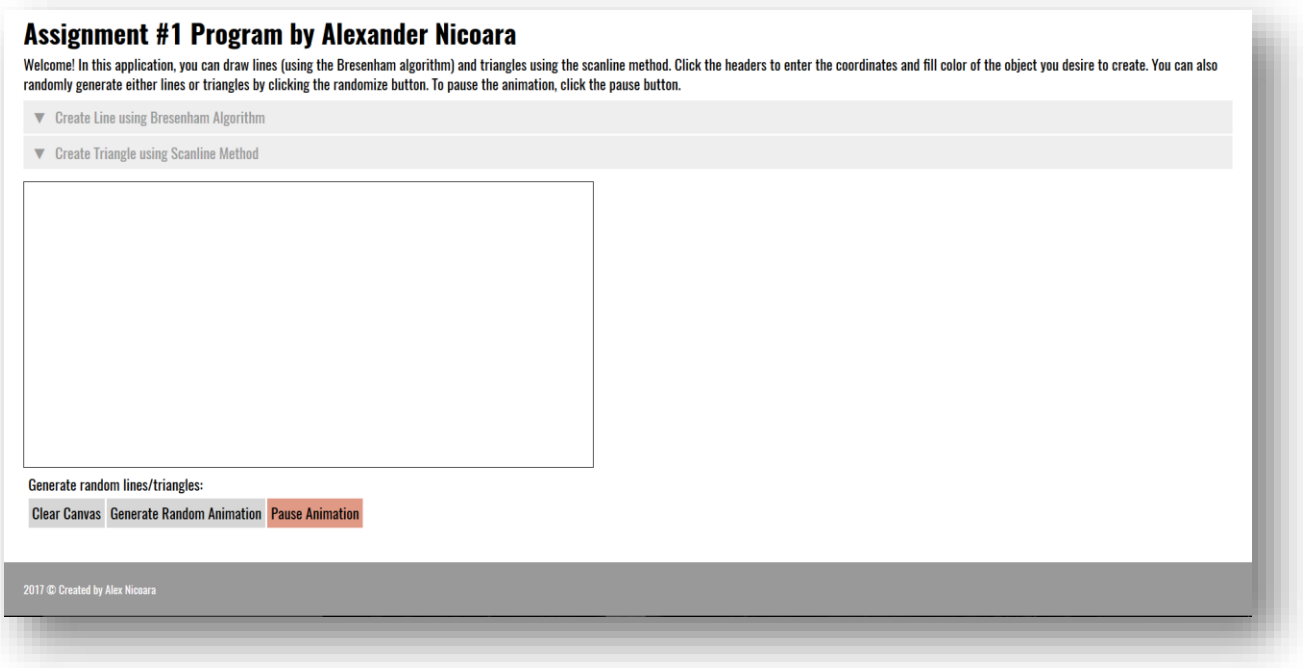


#9) Given an image having the value of 2 in each cell, write the value of the pixels in this image after applying a convolution filter having all 1-s in its entries.



JavaScript Program

The JavaScript program I created for assignment #1 was good way to visualize how the Bresenham algorithm and Triangle scanline fill method work. The picture below shows my application when you first open it:

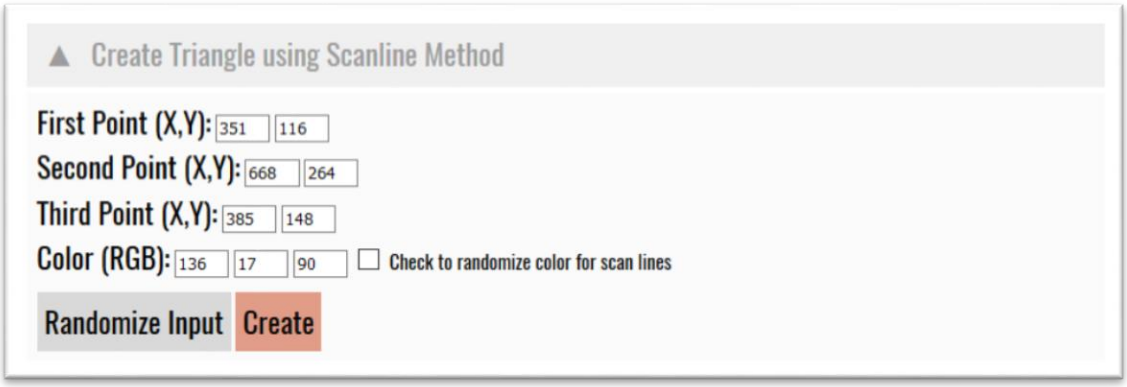


Once the application is opened, you can do 3 things:

1. Draw a line using the Bresenham algorithm, which takes the following parameters listed below (2 coordinates, color rgb):

In my program, you can also randomize input so you can get a feel with how my program performs with arbitrary values.

2. Draw a Triangle using the scan line method, which takes the following parameters (3 coordinates (order doesn't matter), color rgb, switch for random color values for each scanline):



3. Start a random line/triangle generator which spawns either one of these figures every 0.1 seconds (100 milliseconds):

You can also clear/erase the canvas to start from scratch without having to refresh the page. You can also pause the animation.

JavaScript Test Cases

Here are some test cases which verify my programs correctly implement the algorithms and fulfill the requirements of the assignment:

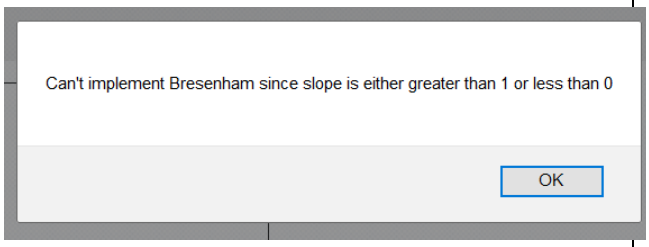
Input	Output
Bresenham Algorithm	
First Point (X,Y): 15 115 Second Point (X,Y): 200 170 Color (RGB): 00 00 00 We expect a black line starting from (15,115) drawn to (200,170) on the 700 x 350 canvas.	

First Point (X,Y): 172 314
Second Point (X,Y): 513 328
Color (RGB): 255 0 0

We expect a red line starting from (172,314) drawn to (513,238) on the 700 x 350 canvas.

First Point (X,Y): 50 100
Second Point (X,Y): 100 600
Color (RGB): 255 0 0

Since the slope of the coordinates entered above is (500/50)=100, the bresenham algorithm can't be implemented since it's greater than 1. We expect an exception to be thrown.



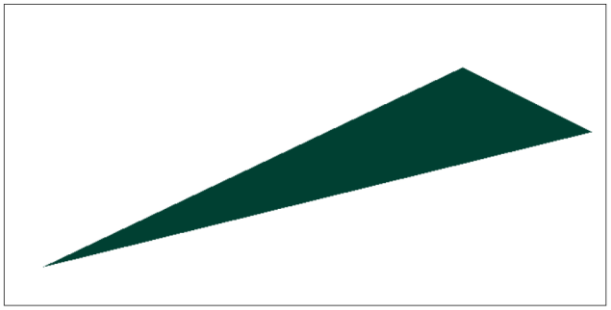
Triangle Scan Line Algorithm

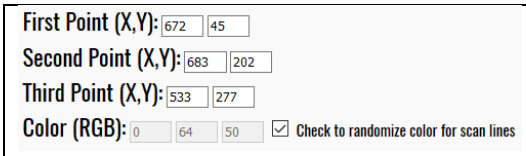
First Point (X,Y): 5 5
Second Point (X,Y): 30 30
Third Point (X,Y): 65 5
Color (RGB): 00 00 00 Check to randomize color

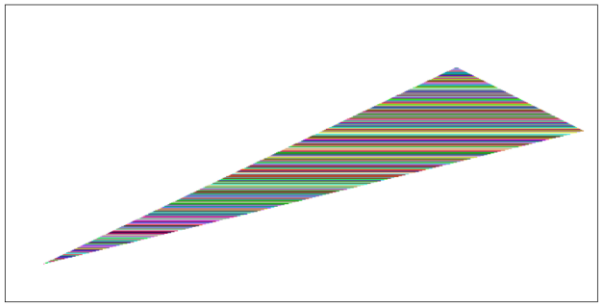
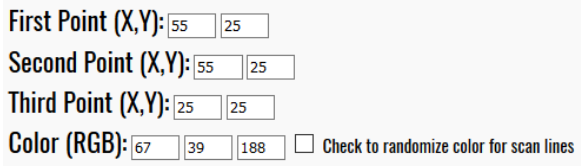
We expect a black flat bottom triangle to be drawn at the coordinates above in the lower left corner of the canvas.

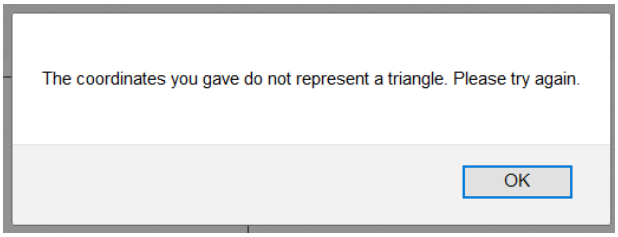
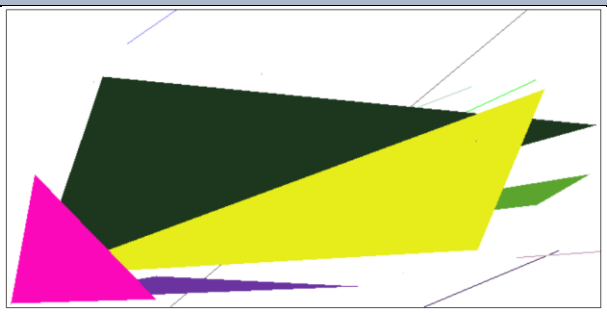
First Point (X,Y): 672 45
Second Point (X,Y): 683 202
Third Point (X,Y): 533 277
Color (RGB): 0 64 50 Check to randomize color for scan lines

I clicked the randomize coordinates button, and it generated 3 new coordinates with a new color to fill the image with. We expect a triangle at the designated points that is the color forest green.



<table>
<tr>
<td> 

Checked off the randomize scan line color checkbox and redrew the same triangle as above. It should give us the same shape with different colors for each scanline.</td>
<td>  </td>
</tr>
<tr>
<td> 

Since all the y-coordinates are then same, we are implementing a line segment instead of a triangle, so this won’t work. We expect to get an error message saying that we need valid triangle coordinates.</td>
<td> 
</td>
</tr>
<tr>
<td colspan="2">Randomizer (Animation)</td>
</tr>
<tr>
<td>When the Generate Random Animation button is clicked, we expected to see either a line or triangle (with random coordinates & color) spawned every 0.1 seconds. When pause is clicked, the animation halts.</td>
<td>  </td>
</tr>
</table>

Assignment Conclusion

Conclusively, this assignment was a challenge at first glance. Even though I am familiar with HTML and JavaScript, I knew very little about the HTML canvas and how to draw/implement pixelation (to show rasterization) on it. Using the MDN website, I learned how the canvas worked and found out how to use JavaScript to implement the rasterization algorithms. From there, I implemented my algorithm using handwritten/calculated test cases which I used the console to help me debug and find out what went wrong. One other issue I ran into along the way was the order of the coordinates and how the algorithm was calculated incorrectly because of the values being in a strange order. For the triangle scan line, I used conditional blocks to eliminate the need to put coordinate values in order so that the algorithm can be correctly calculated. As

for the Bresenham, I didn’t do the same because the points should be in order to show that the slope should be $0 < m < 1$.

Overall, this lab really helped me develop a deeper understanding of the mechanics of the Bresenham and Triangle scan line algorithms. I also really enjoyed programming in HTML/JS and thought it was good programming practice for me (since I want to pursue a web development career).