# CS-411 – Assignment 1 (5%)

Raster Graphics

Due by: September 26, 2017

1. The purpose of this assignment is to practice raster graphics.
2. Answer the following questions:
   (a) Let (1,2) (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.
   (b) 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).
   (c) Given the triangle vertices (1,1) (2,2)(3,1) find the normal vector of the edge (1,1)(2,2).
   (d) Given the vectors A=(2,3) and B=(3,2) find the projection of the vector A onto the vector B.
   (e) 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.
   (f) Explain how can you determine is a point is inside a triangle or not.
   (g) Given the triangle vertices (1,1) (2,2)(3,1) find the barycentric coordinates of the point (1.5,1.0).
   (h) 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.
   (i) 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.
3. Write a JavaScript program that implements the following:
   (a) Implement the Bresenham straight line algorithm. Make sure that your implementation uses integer addition/subtraction operations only. The function should get as input the starting point (2D in pixels), ending point (2D in pixels), and color (RGB), and should render the line by setting pixels in a target image according to the specified color.
   (b) Implement the scanline triangle filling algorithm for an arbitrary triangle. The algorithm works by computing the starting and ending points of HORIZONTAL scanlines and then filling between these points. The function should get as input three vertices (2D in pixels), and color (RGB), and should render a filled triangle by setting pixels in a target image according to the specified color. Two filling modes should be supported: filling with a uniform color, and filling with a different random color for each scanline (to help verify the correctness of your implementation).
   (c) Write a program that renders random triangles and lines using random colors and locations continuously so that a new triangle or line is rendered every 0.1 seconds.

(d) Have buttons to toggle: line drawing, triangle drawing, fill mode for triangle dearwing (uniform color or different scanline colors), and pausing the animation.

(e) A skeleton program will be provided. You are not required to use the provided skeleton program and may modify it as needed.

### Submission Instructions:

1. Create a folder AS1 in your bitbucket repository and create inside it the following sub-folders: `src`, `doc`, `lib`, and `data`. Organize inside the sub-folders the following:

   (a) Report: prepared as a PDF file. The report should contain answers to questions, a summary of program design issues, description of specific problems you faced and the way in which you solved them, and sample input/output results (text/graphic). The report needs to be sufficiently detailed.

   (b) Source code: all program files and necessary data files.

   (c) Any additional data or libraries you used.

2. Note that we must be able to view your report and execute your program in order to grade it.

3. For programming use JavaScript and WebGL (support code will be provided). Test your program on a recent version of Firefox or Chrome and indicate in the code and the report the browser (name+version) the program was tested on. Include a copy of all external JavaScript files so that the program can run locally. You may use instead C/C++ but note that support code is only provided for JavaScript/WebGL.

4. On or before the due date upload your submission to your bitbucket repository. If you are late, upload the submission when you are ready and send an email to *cs*411@*cs.iit.edu* saying you made a submission. To compute "late days will" we will use the last update date of your repository.

5. Direct all questions/comments regarding the assignment to: *cs*411@*cs.iit.edu*

6. Do not submit a paper copy of your report. You will be contacted by email if some material is missing or if you will need to meet with the TA.