Programming Assignment 1: Scan Conversion

Due Date: 23rd ugust 2013

• Introduction

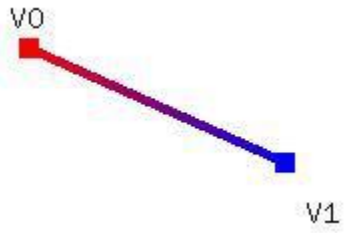
In this assignment you will build an application which is able to rasterize both line segments and triangles. You are required to do the drawing yourself and NOT use openGL primitives!

Your program should receive as input a file containing list of lines and triangles to be drawn in the given format (see attachment). Each vertex of a triangle/line is allowed to have a different color specified. Therefore you must also implement color interpolation across a line/triangle. The rendered output should be displayed to a window. You may assume that vertices lie within the viewport (no clipping is required).

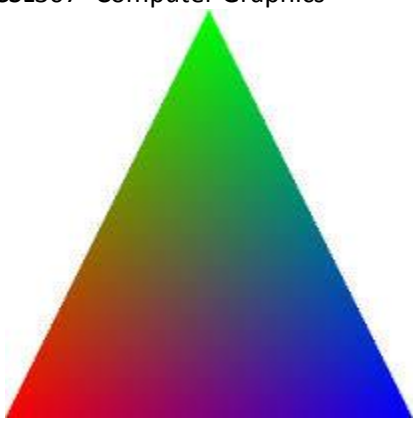
• Assignment

Your task is to:

1. Implement line segment rasterization using the Bresenham algorithm
    a. You should do the drawing yourself and NOT use openGL primitives!
    b. Make sure that it works for **ALL** lines (Bresenhan algorithm discussed in class only works for certain lines!!)
    c. The lines drawn should be 1-pixel thick.
    d. Once your line drawing code is correctly implemented, incorporate color interpolation across the line using the parametric equation of a line segment. Example: Line with end points having colors Red and Blue.



2. Implement Triangle Rasterization
    a. Read Section 3.5 of textbook for more details.
    b. Once your triangle drawing code is correctly implemented, incorporate color interpolation using barycentric coordinates. A very simple explanation of barycentric interporlation is given at http://classes.soe.ucsc.edu/cmps160/Fall10/resources/barycentricInterpolation.pdf



- **Submit**

1. Complete source code as a tar-gzipped archive.
    a. Include only source code but **NO** executables.
    b. Also include your test input file called "***input.txt***"
    c. When unzipped, it should create a directory with your ID. Example: **P2008CS1001** (case sensitive!!!)
2. It should include a **makefile** for compiling your code. It should compile without any errors and produce the executable for testing.
    a. ***Negative marks for any problems/errors***
    b. Executable should be called "rasterize"
    c. Command to test your program with input: ***rasterize Input.txt***
3. The code must be reasonably commented and written in an easily understandable manner.
    a. ***Negative marks for illegible code!!***
4. Include a README file to convey any details.
5. Submit/Upload it to moodle.

- **Grading**

1. Correct implementation line segment rasterization with single color: 15 points
2. Correct implementation of color interpolation for lines: 15 points
3. Correct implementation triangle rasterization with single color: 15 points
4. Correct implementation of color interpolation for triangles: 15 points
5. Correct output for your test file: 20 points
6. Correct output for instructor's test files: 20 points
7. Total = 100
8. Late Submission: 20% Penalty per extra day
9. **Can discuss ideas, but STRICTLY no copying/sharing or source code.**