CIS 441/541: Project #1C
Due April 14, 2021 (which means 6am April 15, 2021)
Worth 6% of your grade

Instructions

1) Download get_triangles.cxx. It has a routine to read triangles from a file. Incorporate its GetTriangles routine in the place of GetTriangles from project 1B.
2) Download the geometry file "proj1c_geometry.vtk".
3) Implement the scanline algorithm for arbitrary triangles and fill up the image buffer with their colors.
4) Note that the output image is 1786x1344. You should change the image size and initialize the buffer to be black (0,0,0). This was done for you in project1b.cxx, so just make sure that code didn't go anywhere.
5) The correct image (GoDucks.png) is posted to the website. Your program should output an image called allTriangles.png.

Note that:
- a file is available online that is helpful for debugging. For each pixel, it says which triangle deposited a color onto that pixel. When differencer tells you a pixel is wrong, you can use this file to narrow down debugging to the exact triangle.

When you are done, submit your code to Canvas.

If your code does not produce exactly the same image, you should expect to get less than half credit. You can confirm that it produces the same image with the difference program and the reference image on the website (GoDucks.png).

Notes (some repeated from 1B's notes):

1) The source data set has 376 triangles that are already "going up" or "going down." One triangle in the source data set has a flat side (i.e., "going right").
2) Some pixels may be outside the screen. Plan for that.
3) Don't forget to use double precision and the floor_441 and ceil_441 functions.
4) Be careful about which quantities you want to be integers and which you want to be double precision.
5) While none of the triangles in the input data set overlap, the handling with floor__441 and ceil__441 effectively make each triangle be "epsilon" bigger. As a result, the picture change slightly based on the order you render the triangles. I.e., if triangles T1 and T2 deposit to the same pixel, then the correct image is dependent on the order you render them. Please follow these conventions: render the triangles in ascending order. I.e., rendering triangle 0, then triangle 1, etc. Also, please allow overwrites. So if T2 is rendered after T1, then T2 should overwrite T1.
6) I just copied the contents of reader.cxx into my code and commented out the old GetTriangles routine. I also forgot to copy the header files and got a compiler

error. (I want a single file for your submission, so copying the contents of reader.cxx is good.)