# Programming Assignment 1: Z-Buffer Rendering - due midnight 11/13

This project is to develop a simple Z-buffer renderer to view 3D models. This will be accomplished by importing a file for a 3D model derived of triangle polygons in a specified file format (below). Sample model files are provided but the progam will work on general files of the specified type.

The program will use Lab 3 as a starter and generalize it to view the model in the file as well as allow basic manipulation as specified below. Use the Z-buffer render algorithm as detailed in class (Lecture 6) to render only the nearest polygon on the scene by storing the depth value in a custom depth data structure. The color of each triangle can be specified at random to allow a visual separation between neighboring triangles.

To be clear, your program cannot use GL functions for polygons or lines (e.g. GL_line) but instead, must draw the image one pixel at a time as was done in lab.

Note, projection can be done simply by dropping the third dimension for each vertex and drawing only the values for the x,y. (FYI, this is a simple orthographic projection.)

Next, the viewer must be able to rotate the model by spinning it about a vertical axis at the center of the model. You will determine this axis from the model file (e.g. average all vertices) To accomplish the rotation, the original 3D vertices must be rotated and then the Z-buffer renderng should be applied to the transformed points.

Finally, to receive full credit you will also need to have your code

organized and well-documented, including describing carefully how to control the interaction (e.g. rotation control).

For extra credit, there are two additions to be added. First, you can receive some easy extra credit points by allowing the user to also draw a wireframe of the model by switching modes (wireframe and filled-in) Next, you can also get points for adding the ability to pan (equivalent to translating the model left-right and up-down) and zoom (equivalent in this case to performing a uniform scale on the model so that is grows larger and smaller.) The latter (pan/zoom) is all or nothing. And you should consider what is a good choice for inputs for each part of the functionality (rotate, change mode, etc.) and communicated that clearly in the program.

Point Breakdown:
Load model 20
Render 35
Rotate 35
Code comments/organization 10

Extra credit:
Zoom/pan 10
Wireframe 5

You will turn in the assignment using the standard "turn-in" before midnight on the due date. Late assignments are accepted up to one week late with penalties of 5 pts off per day, and counting the weekend as a single day.

# Model file specs and samples

The file format is very simple: 1) X number of points; 2) Y number of

triangles; 3) a list of X points in Cartesian space; 4) a list of Y triangles specified as 3 indices from the list of X points, with the first index as 0. Have a look at the sampletri.txt in the set of samples in zip file to see how an easy model appears.