

4050/6050

Programming Assignment 2: 3D Viewer

This project is to develop a graphics program to view 3D models. This will be accomplished by importing a file for a 3D model in the OBJ file format (sample below). Sample model files are provided but the program should work on general files of the specified type.

Starting from the code given for Assn 1, your new program will read and import a 3D model. Set up a simple viewing with an orthonormal camera, centered at the origin. Center and normalize the size of the model as it is brought in. That is, average the vertex positions and set the average at the origin. Then, scale to fit in your window size, by vertex max and min values. Also set the model far enough away from the screen to allow viewing and transformations. (50pts)

Next, the system will perform the following: (points as specified)

Transforms: (20pts) The user indicates their intention to transform the model by selecting one of the following keys: (T) for Translation, (R) for Rotation, (E) for scale. For each transform, the "A" "W" "S" "D" keys will be used to modify the transform relative to the view. For translation, AWS D is Left, Up, Down, Right respectively. For rotation, AD rotates the model about the vertical axis and WS rotates about the horizontal. For scale, AD or WS is used to make the model bigger or smaller. Speed of the transform should be tuned based on the performance in 110 lab machines.

Perspective: (20pts) Add perspective view transformation toggled through "V" button. To get full points for this, you need to select parameters that will work with your choice of normalization and depth of the model as described above.

Note, basic viewing can be done simply by dropping the third dimension for each vertex and drawing only the values for the x,y (simple orthographic projection.) Note, you will need to adapt your line rendering from Assn 1 to view the model in "wireframe".

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 Assn 1. Finally, to receive full credit you will also need to have your code organized and well-documented (10 pts).

Optionally, the user can toggle between wireframe and rendered color ("Z-buffer" algorithm) with button "Z" (for Z-buffer/wire toggle) (20pts EC). The color of each polygon can be selected at random to allow a visual separation between neighboring polygons.

You will turn in the assignment using the standard "hand-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 OBJ file format is very simple: [obj file description](#)

You can ignore everything but the Vertex and Face information in the model file. Also, you can consult references but should write your own parser for this file format.

And here are a couple simple files to start from: [sample file1](#) [sample file2](#)