STL MODEL VIEWER DOCUMENTATION

1 Objectives

The objectives of this program is to interactively view a 3D 'model' allowing adaptive view option (i.e. point cloud representations, triangulated mesh or node volumes). The purpose is to aid in the development of volume based model recreation algorithm inspired by camera modelling and computer vision, by providing a means of interactively viewing the algorithm output.

2 Program Workflow

The following is a graphical representation of the program components (classes and functions) along with their communication paths.

3 Program Compoenents

This section outlines the programs components.

3.1 User Interface

The user interface will be done with the 'dear imgui' library as to allow cross platform compatibility. The users main control is the mesh drawing type (how the model is displayed) & the view angle (where the camera is displayed in the world).

3.1.1 Mesh drawer

Options:

1. Point cloud (purely indices)
2. Triangulated Mesh (default drawing style)
3. Node volumes (will have to develop)

3.1.2 Camera model control (display viewer)

This will use keyboard and mouse input controls to modify view positions. Current command options:

1. click and drag (x,y,z) translation.
2. CTRL, click and drag, (x,y,z) rotation about a point (will have to work out how to define this).
3. Scroll Up - translate into the screen (i.e. zoom in)
4. Scroll Down - translate out of the screen (i.e. zoom out)
5. CTRL, SHIFT, click and drag - select a region of nodes.

3.2 The Model

The most important section of the program

3.2.1 Reading from STL

3.2.2 Writing to STL

Basic

3.2.3 Reading from computer vision algorithm

This section will outline the binary file structure for the Computer model, how it's read in and its individual components.

3.2.4 Saving as computer vision algorithm / STL

This section will outline the options and trade off when saving the model as either a self defined model or an STL file

3.3 Meshing class

Will outline the different implementations of meshing

3.4 Shading class

Will outline the different implementations of Shading

3.5 Display

Will outline the display implementations