CS 462: Computer Graphics Lab

In the assignments, you will design a small graphics application by implementing a set of procedures and algorithms that correspond to the different stages of the pipeline. The application allows the user to *define* object(s), which are then displayed in a *specified* way. **You should create a library of the procedures you are implementing, which can be called to implement graphics applications.**

The implementation will be done in stages, in sync with the progress in the theory course. Along with the implementation of the basic algorithms, you will also be required to learn the corresponding OpenGL functions and use those as specified in the assignments.

Instructions:

a) Form two-member groups (inform me in case of any difficulties/confusion).
b) Any copying is strictly prohibited.

Part 1

1. TAs will check your assignment (AL2 lab slot)
2. Date of submission: 3/9/13

At the very beginning of the graphics pipeline, we need to represent/model 3D objects. We have discussed about different object representation techniques in the class. In part 1 of the assignment, you are expected to do the following.

1. Learn about different object representation techniques supported by OpenGL. Construct a scene with **atleast 5** different objects represented using **atleast 3** different representation techniques (I leave it to your imagination to design the scene).
2. Create a library of procedures to represent objects. Your library should have the following functions.
    a. A function to represent objects using vertex list.
    b. An octree-based interior representation (you may define your own policy for representing non-uniformity).
3. Draw TWO scenes with TWO objects each. The objects are same in both the scenes. In one scene, the objects are represented using vertex list. In the other, they are represented using octrees (with non-uniform voxels). You may use OpenGL functions to render the scene (for color, projection etc.).
4. Write a procedure to create triangular meshes from a vertex-list definition. **Assume convex polygonal surfaces only**. Show the objects in the 1st scene of Q3 (i.e., defined with vertex list) in the mesh form. You may use OpenGL functions to render the scene (for color, projection etc.).

Note: you should create the scene in Q3 above by calling the library procedures you defined. The mesh form in Q4 must also use the library procedures you defined.