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.

Part 2

1. **TAs will check your assignment (AL2 lab slot)**
2. **Deadline: 17/9/13**

We want to create a *block & sphere world* in which all objects are made up of cubical blocks and spheres. In this part, you will create the objects in the block & sphere world from a basic definition of block and spheres, using geometric modeling methods as discussed in the class. Do the following.

1. Represent a block and a sphere in its own (local) coordinate system (simple vertex list/ center and radius representation will do).
2. Implement a set of library procedures for translation, rotation, scaling and shearing.
3. Call these library functions to construct a scene of the block & sphere world having the following objects. **Note that these objects are created by applying transformations on the basic definitions of Q2 (use your imagination for making the objects with blocks & spheres).**
    a. A house
    b. A tree
    c. A road
    d. One character (human/non-human)
4. Represent the above scene using either of the procedures from part 1?
5. Learn OpenGL functions for geometric transformations. Construct the same scene using OpenGL functions (not your library functions) for comparison.

Notes:
1. **You can use OpenGL functions for rendering the scene in Q3 (e.g. illumination, projection etc.).**
2. **The objects can be simple approximation rather than the exact shape. They need not be perfect.**

Part 3

1. **TAs will check your assignment (AL2 lab slot)**
2. **Submission deadline: 22/10/13**

The block and sphere world you have created in the previous assignment needs to be colored. Subsequently, it needs to be project on a view plane. You will do these based on the concepts discussed in the class. Implement the following.

1. Implement a set of library procedures for performing the following tasks.

   a. Calculating intensity at a point using simple lighting model.
   b. Interpolation of intensities at pixel locations using Gouraud shading.
   c. Transferring world co-ordinate scene to eye-coordinate.
   d. Perspective projection (simple one-point perspective).
   e. Window to viewport transformation.
2. Using the procedures you defined in Q1, perform the said tasks for your scene.
3. Learn OpenGL functions for the above tasks. Construct the same scene using OpenGL functions (not your library functions) for comparison.

Notes:

1. **You may ignore canonical view volume transformation.**
2. **For Q2 above, you should define your scene and view in a way such that clipping and hidden surface removals are not important. One way to do that is to use simple objects with non-overlapping surfaces and a properly defined view position. Note that you can avoid clipping but cannot avoid HSR, only can minimize it. For this assignment, distortions due to HSR will be considered (you need not worry about HSR).**
3. **You can use OpenGL functions for rendering the scene in Q2 (e.g. scan conversion).**