1 The Modelling-Viewing Pipeline

1. Do we need to have a modelling-viewing pipeline? Can we, instead just form the requisite perspective projection matrix and multiply it with all objects to form the image? Answer keeping in mind that the process has to be implemented and is not purely based on mathematical/theoretical considerations.

2. Why do we do Projection Normalization in our Modelling-Viewing Pipeline?

3. Where should clipping be done in the modelling-viewing pipeline and why?

4. How can we get the following in OpenGL (answer in terms of a general sequence of operations, exact syntax of commands in not necessary):
   (a) Axonometric Projections
   (b) Oblique Projections
   (c) 2 and 3 point perspective projections

5. Why is the Z (or depth) coordinate carried through the Modelling-Viewing pipeline when we only need the X and Y coordinates to locate pixels on the final image?

6. What kind of transformation does the Z coordinate undergo from the WCS to VCS to CCS to NDCS to DCS? Answer in terms of changes in depths that you can resolve successfully under such a sequence of transformations?

7. What are Euler angles? How can we define a viewing transformation in terms of Euler angles?
2 Visibility

1. What is the effect of using the pseudodepth on the accuracy of depth resolution in the Z-Buffer algorithm?

2. What effect does the choice of polygon that becomes the root of the BSP tree have on the use of BSP trees for resolving visibility?

3. A 3D world consists of number of objects. A BSP tree, $T$, is constructed for this world. If the entire world, i.e., all the objects in the world undergo a rotation by $\theta$ degrees about the vertical axis passing through the centroid of the world, what is the effect on the BSP tree? If $T'$ is the BSP tree computed for the rotated world, what is the relation between $T$ and $T'$. 