Modelling-Viewing Pipeline : Viewing

CS475 / CS675: Computer Graphics - Assignment 2

Deadline: 25/9/2015, 23:59:59

1 Introduction

This is the second part of a two part assignment on exploring the modelling-viewing pipeline. This part deals with viewing.

2 To do

1. Simulate the modelling-viewing pipeline in OpenGL. The models that have to used must be the models you created in assignment 1.

2. Create a scene file called myscene.scn. This will be a ASCII text file in the following format:

```
#Model file details followed by modelling transformations
<Name of first model .raw file>
<Sx Sy Sz>
<Rx Ry Rz>
<Tx Ty Tz>

<Name of second model .raw file>
<Sx Sy Sz>
<Rx Ry Rz>
<Tx Ty Tz>

<Name of third model .raw file>
<Sx Sy Sz>
<Rx Ry Rz>
<Tx Ty Tz>

#VCS Setup
#In World coordinates specify
<Ex Ey Ez>
<Lx Ly Lz>
```

```
<Ux Uy Uz>

#Frustum Size
#In View coodinates speficy, wrt the eye
<L, R, T, B>
<N, F>
```

3. The first section in the scene file above will give the names of the three files containing the model that you made earlier. After each filename give the scaling, rotation (angles in degrees)and translation parameters that situate this model in the WCS. Note the the line starting with a "#" is a comment and need not be present in your scene file.

4. The next section gives, in WCS, coordinates of the Eye and the LookAt point, and the unit Up Vector. The section after this gives, in VCS, the size of the image on the near plane with L, R, T, B being the left, right, top and bottom dimensions measured from the eye on the near plane. N and F give the distance to the near and far planes from the eye - they should be positive numbers.

5. To start with, parse the scene file and load all the models. Draw the frustum in WCS with lines, draw a eye as a point and draw all the models after applying all the modelling transformations to them. The part of the frustum between the near and far planes must be drawn in cyan, the projectors from the near plane to the eye in magenta, the eye in red, and the objects in their respective colours.

6. Each model must be in a separate VAO, and the frustum must be in another VAO. You can use as many VBOs inside the VAO as you want.

7. You will define keys 1, 2, 3 and 4 to move from the WCS to VCS (1), from the VCS to CCS (2), from CCS to NDCS (3), from NDCS to DCS (4). For each one of these you have to construct the matrix manually, as explained in class. You can use glm::mat4 to store the matrix and glm::vec4 for storing the vertices, GLM or GLSL can be used for multiplying matrices with vertices to proceed through various stages of the pipeline. No other GLM function is to be used as a part of the simulation. You can also use GLM functions for dot products, cross products and normalizing vectors.

8. The simulation is to be visualized under a second orthographic projection - glm can be used to set this up in the usual manner and use of glm::ortho and glm::lookat is allowed for this, and only this.

9. You can have the keys that allowed you to translate and rotate the entire scene in the previous assignment, still active if you want to visualize the simulation better, from other view points.

10. A bonus of upto 20 marks will be awarded if clipping is implemented in 3D in CCS, or in 2D in DCS. A clipping algortihm of your choice can be implemented. The bonus marks will be awarded only if the rest of the assignment is complete, so please complete the regular assignment before attempting the bonus.

3 Marking

The assignment will be marked as follows:

1. Implementing the scene file parsing correctly : 10 marks
2. Visualizing all the models and frustum in WCS correctly, including multiple VAOs: 30+10 marks
3. WCS to VCS: 20 marks
4. VCS to CCS: 20 marks
5. CCS to NDCS: 20 marks
6. NDCS to DCS: 20 marks
7. HTML Report : 10 marks
8. Bonus - Clipping in 2D or 3D: 20 marks
9. Total : 140+20 marks

Also, note:

- Late submission will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., zero marks if the assignment is more than three days late after the due date.)
- It is **ok** to submit a partially done assignment if you cannot complete it. You will get partial credits.
- Please do not plagiarise source code.

4 To Submit

1. A Tar-Gzipped archive of the complete source code and the created .raw models. It should compile using a Makefile on any Ubuntu system.
2. Include a README file in the .tgz archive containing a link to a html report page on the assignment that should contain some details about what you implemented and images of the results that you generated.
3. Submission will be via the submission page.