

COS 350 – Computer Graphics Scene Modeling

ASSIGNED: October 3, 2008

DUE: October 20, 2008

SUBMISSION: Create a zip file of all of the source code, compiled code, etc and submit through the course moodle pages.

WEB PAGE: Capture at least 3 images from the output of your submitted assignment and create a web page including the images and an explanation of your work. Make a directory in your public_html/350 directory named Scene. Store your web page for this assignment as index.html in this directory.

In this assignment, a scene model will be built and displayed using 3 display windows and a control window. The display windows present front, side, and perspective views. The control window contains GLUI elements to manipulate the viewing parameters.

Requirements

- the scene will contain multiple objects – the objects themselves can be simple
- each object will have material properties assigned to it
- multiple lights should be present in the scene
- the scene is stored in a tree structure so that the display function is a tree traversal
- the scene is viewed from 3 directions with 1 window open for each
 - front view, side view, perspective view
- scene manipulation controls (using the GLUI library) to support
 - change camera position in perspective window
 - zoom in/out in front and side windows
 - pan “sideways” in front and side windows

Comments

The use of GUI manipulators with a tree traversal approach to drawing a scene raises some interesting issues. Where in the code are the changes from the interface asserted?

Global manipulators that alter the entire scene such as rotating the scene around an axis would be included at the beginning of the drawing process in the same manner for each window. This type of motion is not really very useful since multiple views are provided.

Local manipulators specific to a given window are more useful since they alter the viewing parameters for that window. Perspective windows would typically support some form of rotation while front/side/top windows would more naturally include zooming and panning. These changes could either alter the camera position (good for zoom) or be applied as a transformation after the camera position and parameters are defined by gluLookAt. Panning operations to appear natural should either translate the whole scene in one direction or change both camera position and where the camera is pointed.

Local manipulators specific to a given object are more difficult. For manipulation of a single object (or selected group of objects), only the object is changed. Since the object is stored as one or more nodes in a tree, the object node (or parent node for non-primitive object) must be changed by modifying the transformation node stored in the tree. This type of change is more related to modification of the scene rather than changes in viewing parameters. Therefore, this type of change is more naturally included in the scene building program.

Resources

Useful example code and links to GLUT reference material and example code are provided on the course moodle page.

- figuretr.c – example code from text – draws robot figure using tree traversal
- multiFigure.c – example from text (does not use tree traversal) modified to include front view, side view, and perspective view windows
- gluiFigure.cpp – multiFigure.c with a GLUT rotation control added to manipulate the perspective view window

Extra Credit – 20 points maximum

Support some type of manipulation of individual objects and, thus, manipulation of nodes in the scene tree. For example, a flag to indicate if a given object is to be displayed could be added to the node along with a method to change the visibility of each object. Another example would allow movement of an individual object. Such changes will be visible in all windows. Manipulation must be based on a GLUT interface.

Grade Template

materials, lights, program basics	20
tree traversal scene drawing	30
3 viewing windows, each with proper behavior	20
GUI interface design and implementation	30