Comp480

Programming Assignment #1
Due 3/20/08 (before 11:59:59pm)

Objective: Use OpenGL to render, manipulate, and animate 3D models. Your scene should be composed of a ground plane, and at least 3 models. One model must be articulated with at least 3 joints (e.g. a Luxo lamp, a robot, etc.), and at least one model must be read in from an OBJ model file. You will implement a number of different viewpoints, and provide an interface to position and move the objects within the scene. Your program should allow the user to select particular joints in the articulated object and move it, and its children, relative to its joint constraints using a natural "mouse-based" user interface. You can use the provided skeleton codes as a starting point.

Developing environment: Windows OS is recommended

Minimal requirements (worth 80%):

(Note: items shown with violet color is already included in the skeleton codes)

1) Create a simple scene that contains your models, including a ground plane, and other assorted items for visual reference
2) The scene should be viewable from several viewpoints including at a minimum:
    - An "eye point" and "view direction" attached to each object (e.g. The cow's point of view); you need to define eye point and view direction for each object.
    - A fixed "overview camera" positioned above the scene looking directly down that always includes the whole scene
3) At least 2 free cameras that can be moved around to generate arbitrary views of the scene. Each camera should be treated as a coordinate frame and should be visually rendered with a camera model
4) Provide an interface for cycling through, or selecting, each of the views with key maps
5) Provide an interface for moving cameras and objects.
    - For each object, provide translation, rotations, trackball interface in the local modeling space
    - For each camera, provide pan, dolly, zoom, roll, and trackball
6) Draw your articulated object hierarchically: that is, each part will not keep track of its overall position, but rather its position with respect to other parts. As you draw, you should use the GL matrix stack to go from each part either to a child part, or back to the parent part.
7) Provide an intuitive "mouse-based" interface to provide constrained motion about each joint of your articulated object; provide a key-map based selection method to choose each part of the articulated objects.

Extras:

1) (5%) Add a "fly-to" interface to the viewing camera. When the user clicks on a point in the image space, the camera should fly to a close-up view of that point.
2) (5%) Select an object with a mouse; use the selection technique using a back buffer; provide a toggle key map to see the back buffer
    - If you select a key, you enable a selection mode.
    - Once you select an object, you can move around the object with a mouse
    - If you select the key again (e.g., toggling), you disable the election mode.
3) (5%) Animate your objects with a short sequence to "look at" some specified point by a mouse
4) (5%) Automate one of your free cameras. For instance, you might make your sky camera always look at one of your animated objects, staying a fixed distance away from it.
    - Provide a selection mechanism to choose one of free cameras and an object
5) (5%) Implement multiple views within one window, like a 3D modeling package. Use orthographic top, side, and/or front views. One view should be perspective. (This could be useful for composing scenes for your ray tracer in a later assignment).
6) (Variable) Some idea of your own. Be sure to consult with the instructor.

Policies: Everyone must turn in their own assignment. You can collaborate with others, but any work that you turn in should be your own. Turn in your work by emailing an archived and compressed version of it (source and executable) to TA (Mr. NaeJin Kong) and demonstrate your code to the TA for the evaluation.