Comp480

Programming Assignment #4
Due Mar.-16 (Mon.) (before 11:59pm)

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

Requirements:

(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
   - Load at least two objects, e.g., cow and beethovan models.
2) The scene should be viewable from several viewpoints including at a minimum:
   - A fixed "overview camera" positioned above the scene looking directly down that always includes the whole scene
   - 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
3) Provide an interface for cycling through, or selecting, each of the views with key maps; you can do that by clicking "0", "1", and so on.
4) Select an object or a camera with a mouse; use the selection technique using a back buffer (15 pts)
   - If you click "s", you enable a selection mode.
   - Once you select an object, translate the object with a mouse; the selected object should follow the mouse cursor, if the current transformation is done in the viewing space.
   - If you click the key again (e.g., toggling), you disable the election mode.
   - If we type "b", toggle of showing the back buffer
   - (Note: reading back buffers may not work correctly when you use remote machines.)
5) Provide an interface for moving cameras and objects. (15 pts)
   - For manipulation of cameras, provide pan (p), dolly (d), zoom (o), roll (r), and trackball (t); I know lots of key maps..
   - For manipulation of objects, provide translation and trackball (t) interface in the viewing space; keep all the functions that you implemented for PA2 and PA3
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. (10 pts)
   - Support that you can choose each part and provide constrained motion (e.g., rotation if you type "r") about each joint of your articulated object
7) Print all the key maps and their functions if you click "h" (5 pts)
8) Implement multiple views within one window, like a 3D modeling package (10 pts)
   - 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).

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. Byun).