

Comp770 – Spring 2007

Programming Assignment #1

Due 2/14/07 (before 11:59:59pm)

Objective: Use OpenGL to render, manipulate, and animate 3-D 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 a OBJ model file. You will implement a number of different viewpoints, and provide an interface to position and move the objects within the scene. Every object must be able to move with respect to various selectable coordinate systems, including each model's local coordinate frame and a "world coordinate frame". Finally, 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 code provided in Lecture 6 as a starting point.

Minimal requirements (worth 80%):

- 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)
 - 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 have a picturesque starting point.
- 3) Provide an interface for cycling through, or selecting, each of the views.
- 4) Provide an interface for moving cameras and objects. Provide a track-ball interface for manipulating objects. In addition, provide simple translations, rotations, and uniform scales about
 - The local frame of each object
 - The world coordinate frame
- 5) Provide some indication of the current frame for which the current transformation is being specified (a wireframe depiction is fine).
- 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

Extras:

- 1) (5 pts) Allow for the introduction of arbitrary transformation frames about which any objects can be manipulated (rotated and transformed).
- 2) (5 pts) Implement a Maya-like manipulator for translation. Dragging an axis of the manipulator should constrain motion to along the axis. Dragging the origin pans w.r.t. eye space.
- 3) (5 pts) Add a "fly-to" interface to each free camera. When the user clicks on a point the camera should fly to to a close-up view of that point.
- 4) (5 pts) Animate your objects with a short sequence to "look at" some specified frame
- 5) (5 pts) Animate your objects to move about the scene in an interesting way.
- 6) (5 pts) Automate one of your free cameras. For instance, you might make your sky camera always look at one of your objects, staying a fixed distance away from it.
- 7) (5 pts) 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).
- 8) (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 the instructor along with instructions for running your code.