PROJECT 2 – 3D MODELING

CSc 385, Winter 2016
Assigned: Jan 20
Due: Feb 4

Aaron G. Cass
Union College

OBJECTIVES

The primary objective of this project is to learn how to model an interesting scene in 3D, using cube geometry and the transforms we've learned about in class.

STARTER CODE

I'm providing you with starter code that provides a very simple scene with 2 "cubes". The code makes use of a class that handles matrix operations, another that handles coordinate vectors and vector operations, and some structures to manage geometry and a set of *shaders* (we'll discuss shaders in class).

YOUR MISSION(S)

REQUIRED

You are required to make your own scene with cubes transformed in different ways to make something interesting. Add to the existing code to add the new "cubes". Your scene should have at least 5 "cubes", and must involve translations, scaling, and rotations.

You are also required to implement a way to handle simple hierarchical objects (i.e. objects composed of other objects). In your hierarchical objects, for the sub-objects you should store just the *relative* transform from the *parent* object instead of storing the transform from the world frame.

You are also required to add code to the keyboard callback to cycle through the objects in the scene, *selecting* each object in turn, when the user presses the space key. For example, the first object will be selected when you run the program and then pressing the space key once will cause the second object to be selected. Pressing again will cause the third object to be selected, etc... Selection should wrap around to the beginning if the user presses the space key when the last object is selected.

OPTIONS

All of these options are intended to give you experience using GLUT's callback mechanism for user interface implementation – this will be quite useful when we get to more fully interactive systems later in the term.

You must do one of these options. For extra credit, you can do two of them.

Camera Pan: add code to the keyboard callback so that hitting one key (such as the right arrow) pans the camera to the right and hitting another key pans the camera to the left. A pan is a camera rotation.

Camera Tilt: add code to the keyboard callback so that hitting one key (such as the down arrow) tilts the camera down and hitting another key tilts the camera up.

Camera Truck with Mouse: add code to the mouse callback so that dragging the mouse to the right trucks the camera right, while dragging the mouse to the left trucks the camera left.

Object Rotation with Mouse: add code to the mouse callback so that the selected object is rotated based on the mouse movements. A mouse drag to the right would cause the selected object to rotate right (i.e. about the Y axis in the positive direction).

TURN IT IN

To turn in your project, zip the entire directory of code, rename the zip file with your name (e.g. `smith-proj2.zip`), and email it to me. In your email, indicate which of the above options you have done.