6.837 F99 Asst 4 – Scene Modeling

Last week we learned how to compose modeling primitives into modeled *objects*. Assignment 3 challenged you not to make a single object, but rather to write a *program* to produce a *family* of objects according to some space of parameters. The result will be more than 100 programs which together produce an impressive variety of objects.

This assignment asks you to **manually** invoke your classmates' programs (and your own), with parameters of your own choosing, to produce objects. You will then incorporate these objects into a modeled *scene*. The scene need not be a conventional scene (say, an outdoor landscape, or a house interior); it can be the inside of a machine, or somewhere in outer space, or even an abstract setting (although there should be ample evidence that you have chosen, and situated, objects purposefully, rather than haphazardly).

Modeling Convention: As with your modeled objects, your scenes should **occupy a cube of side one centered at the origin**. How, you ask, if each constituent object is unit size and at the origin? Easy – simply situate all of the unit-size objects in a scene with scalings, translations, and rotations, using units and dimensions of your choice. Then, figure out your scene's bounding box, and add transformation nodes which "unitize" that box in front of a top-level `Separator`. Why employ this convention? Because it's possible (even likely) that someone, someday will take your *whole model* (city, forest, solar system, whatever) and put it on a pedestal in a virtual museum. Make life easy for them (and for us, when we attempt to view your scenes).

The Point: The goals of this assignment are threefold. First, you will learn to work with other people's object families. Some of the creators' intentions may seem strange or less than useful. Some you may not understand. On the other hand, you may exercise their programs, in ways unforeseen to them, to produce extremely interesting objects. Second, you will learn to situate modeled objects, each with its own local coordinate system, in a global coordinate system. Occasionally you may encounter an object which did not respect the modeling "conventions" of Assignment 3. This should lead you to appreciate those who did respect the conventions, and the ease such adherence lends to object placement operations. Third, the assignment should exercise your creative, aesthetic, compositional, and technical skills!

Tools:

- The Asst 3 executables will reside in `/mit/imagery4/6.837/F99/bin/`. **Please do not copy any executables out of this directory**. Rather, use the fully resolved pathname for the program, or add the above directory to your shell path.

- See `/mit/imagery4/6.837/F99/og/` for a gallery of representative `Asst 3` objects.
- Eventually, a gallery of your `Asst 4` scenes will reside online as well, for viewing and enjoyment by the entire class.

To turn in (details TBA by email and online):

- Your modeled inventor scene, `uid_scene_name.iv` (where `uid` is your username, `scene` is a **required keyword** that allows us to identify your scene file, and `name` is some descriptive word for your scene, e.g. `forest, city, snowdome, asteroids`, etc.). **Do not** create any other files of the form `uid_scene_*.iv`, as these will confound our scripts.
- Any other `.iv` files which are the subject of an Inventor `File` include operation, and any files they include, recursively (the point being that we must be able to do `gview uid_scene_name.iv` to parse and view your scene without errors). You should make sure that this works before doing `turnin`, for example by copying only those files you intend to submit to a test directory, and making sure that the command `gview uid_scene_name.iv` works there.
- A short report (as an embedded comment inside the top-level `.iv` file) noting any particular effect(s) you were trying to achieve, any difficulties you overcame along the way, and anything else you would like the staff, or viewers/users of your scene, to know. We will examine this report by invoking `grep "^#" uid_scene_name.iv` (try it!).
- Finally, include a "credits" line for your scene, also as an embedded comment, in your `.iv` file. For example, you might include your name and email address so that any users of this `.iv` file will know where to direct their fan mail (or constructive criticism). Just like movie credits, this line should acknowledge anyone that helped you with your object, or any sources of geometry from which you gained inspiration.

This assignment is **due 5pm Friday, 8 October 1999**.