ECE4893/8893 **Fall Semester, 2013**

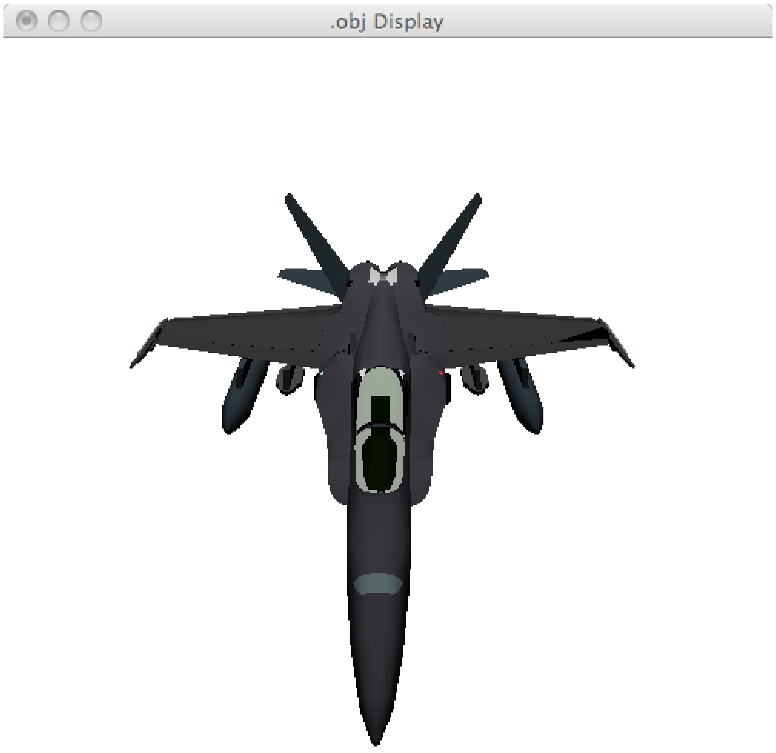
Project 4 – Using OpenGL (Version 2.0)

Assigned: Oct 22, 2013 *Due: Nov 3, 2013, 11:59pm*

Introduction In this assignment we will use the OpenGL graphics API to create and display a three-dimensional image of a jet aircraft. The inputs are two files that describe all of the characteristics of the jet image, including the "faces" (essentially triangles), "vertices" (a single point in 3-D space), "normal vectors" which is a 3-D vector that is normal to a face, and "materials" which describes the light reflectivity of the faces. Further, your program will use keyboard input to control the rotation of the jet, as described below.

The specifics are:

1. Create a class called `Vertex` that has three variables of type `GLFloat` representing the x, y, and z values of the vertex.
2. Create a class called `Face` that has three variables of type `Vertex` representing the three points of the face (triangle).
3. Create a class called `Normal` that has three variables of type `GLFloat` representing the endpoint of the vector normal to a face. Hint, subclass the `Vertex` class to save some typing.
4. Create a class called `Material` that has five variables describing the reflectivity of a face. The five variables are described later in the section discussing the input file formats.
5. Control rotation about the x, y, and z axes by keyboard inputs 'x', 'y', and 'z' respectively. For each axis, if the jet is already rotating, turn the rotation off. If it is not rotating, animate a rotation about the specified axis resulting in one complete rotation every 5 seconds.
6. **Grad Students Only** Control the scaling with two keyboard inputs, 'S' (Larger) and 's' (smaller). The 'S' key increases the scale factor by one percent, and the 's' key decreases the scale factor by 1 percent.
7. **Grad Students Only** Add additional lighting sources (at a location of your choosing) with 'L' (up to 6 total) and remove a lighting source with 'l' (lower case L).
8. The 'q' key quits the program.



Input File Formats There are two input files that describe the jet image.

1. The `jet.obj` describes the "objects". There are a number of fields in this file, described below.
   (a) `mtllib` specifies the name of the materials file, described below.
   (b) `v` specifies a single vertex, with three floating point values specifying x, y, and z respectively.
   (c) `vn` specifies a face "normal" vector, with three floating point values specifying x, y, and z respectively.
   (d) `g` specifies an object name and is not used in this assignment.
   (e) The usemtl specifies the index into the materials array, which is created while reading the materials file (described below).
   (f) `s` specifies a smoothing group, and is not used in this assignment.
   (g) `f` specifies a face (a triangle) of the form $a0/b0/c0$ $a1/b1/c1$ $a2/b2/c2$, where $a0$ is the vertex index of vertex zero. **NOTE. these start from 1, not 0**. $b0$ specifies texture zero which is not used in this assignment. $c0$ is the index of the appropriate normal vector for vertex zero of this face. The next two strings specify the vertex, material and normal for vertices 1 and 2.
   (h) `vt` specifies texture coordinates and is not used in this assignment.

2. The *materials* file species the material for the faces. The format is:
   (a) `newmtl` specifies the index of the next material. Note this starts from zero.
   (b) `Ns` is the *shininess* value.
   (c) `Tr` is the *transparency* value.
   (d) `Ka` is the *ambient lightling* values, in r, g, b.
   (e) `Kd` is the *diffuse lightling* values, in r, g, b.
   (f) `Ke` is the *emissive lightling* values, in r, g, b.

(g) `Ks` is the *specular lighting* values, in r, g, b.

The skeleton `jet.cc` file has an example of how to read either of these files using `ifstream` and the input operator. The skeleton file will also have examples of how to use the material values.

Copying the Project Skeletons

1. Log into `jinx-login.cc` using `ssh` and your prism log-in name.
2. Copy the files from the ECE8893 user account using the following command:

   ```
   /usr/bin/rsync -avu /nethome/ECE8893/OpenGLJet .
   ```

   Be sure to notice the period at the end of the above command.
3. Change your working directory to `OpenGLJet`

   ```
   cd OpenGLJet
   ```
4. Copy the provided `jet-skeleton.cc` to `jet.cc` as follows:

   ```
   cp jet-skeleton.cc jet.cc
   ```
5. Then edit `jet.cc` to create your code for the test cases.

Turning in your project. The system administrator for the jinx cluster has created a script that you are to use to turn in your project. The script is called `riley-turnin` and is found in `/usr/local/bin`, which should be in the search path for everyone. From your **home directory** (not the OpenGLJet subdirectory), enter:

```
riley-turnin OpenGLJet.
```

This automatically copies everything in your `OpenGLJet` directory to a place that we can access (and grade) it.