

CMSI 371-01

COMPUTER GRAPHICS

Spring 2015

Assignment 0430a

This assignment seeks to wrap up scene rendering so that we can move on to dynamic/interactive behavior.

Outcomes

This assignment will affect your proficiency measures for outcomes *1b*, *1c*, *2a–2c*, *3a*, *3d*, and *4a–4f*.

Proficiencies of + can now be applied to all outcomes in this assignment.

Not for Submission

If you have been following along with the Angel textbook, at this point, we have pretty much covered Chapters 1, 2, 4, and 5, with the exception of Sections 4.14, 5.10, and 5.11. Non-graphics-specific interaction material in Chapter 3 was studied in CMSI 370; the computer graphics content shouldn't be too hard to assimilate if you understand the rest of the material.

The lighting model that we are using, plus a few more details, can be seen in greater detail in Angel Chapter 6 and the orange book Chapter 9. And of course, the C or C++ code in the orange book must be superseded by or adapted into JavaScript and WebGL.

For Submission

For the following tasks, keep building on *homework/pipeline* on your git repository. Do rename files, however, to better reflect what you have going now (yes, that means no *hello-webgl.htmls* etc. anymore—you're way past "hello" now!).

The New Normal

Add support for normal vectors to your shape objects. You may use any technique for generating them, including (correctly) using the functions given to you, writing code of your own, and manually specifying them. You want to do this because you will then set up...

Lights!

Implement a lighting model for your scene. At a minimum, you should use the model shown in class and detailed in the reading. You can go be-

yond that if you wish (e.g., the UberLight model described in Chapter 12 of the orange book).

In terms of design, you have the option of integrating this model into a general construct such as a Scene object, or you may add the code directly to the main JavaScript sequence. This flexibility is provided because the raw functionality alone entails sufficient work. But if you can get past that *and* make your lighting code reusable, then great!

Camera!

Implement the camera transformation matrix and integrate it into your scene. Again, you may abstract this out or just make it part of the main sequence. Note that, at this point, you should have full scene layout and navigation capabilities.

Action...Almost

Ultimately it may be too restrictive to strictly partition the remaining 3D scene work to just lighting and camera or just animation and interaction. However, acknowledging the amount of work involved, we separate animation/interaction work into the next assignment. Chances are that you will want to work on both concurrently—after all, what fun will it be to have a camera but not be able to fly around like a superhero? ;-)

Commit and push your work to your git repository under *homework/pipeline*.