Lights and Materials

COMP 4302/6909 — Assignment 4

Due: March 13, 2024 (Wednesday) 11:59 PM

Assignment Objective:

To start using the WebGL light model and see how the light model works together with the material properties of the illuminated models.

Your Task:

In this assignment you are asked to write a program where you will be able to add illumination effects to your scene using WebGL. Alternatively, you may produce a Three.js, Unity, or an Unreal demo that performs the same job as the WebGL assignment.

There are three basic shading styles: flat shading, where each polygon is rendered with a unique color according to the normal of the surface that reflects the light; smooth (a.k.a. Gouraud) shading, where the shading at each of the corners or vertices of the polygon is computed and then the hardware produces interpolated shading across the surface of the polygon; the last model is Phong shading, which uses the Phong illumination model discussed in class and applies it to every single fragment. Gouraud shading is a shading style which can only be achieved when using smooth shading. For Gouraud shading to be successfully implemented, the system needs to compute per-vertex normals, normally produced as an interpolation from the surface normals. These and other shading techniques are implemented using pixel shaders, which you might want to try for your final project.

Take your previous assignment and add the WebGL code necessary to enable lighting, including planar shadows. After that, set up the material properties so that each object has a distinct color and appears properly shaded.

To illustrate the light position, insert keyboard input to setup the sources in three different positions: 'T' will place the light source on top of the viewer's camera, 'L' on the left, and 'R' on the right. (use a different color for each). Use these same keys to turn light on and off.

To demonstrate shading modes, you will be asked to add to your scene an additional mode that can be used to illustrate the two shading styles (flat shading and smooth shading). To switch between shading styles, use 'F' for flat shading, 'G' for smooth shading, and 'H' for Phong shading. Remember that for correct lighting to take place, it is essential to turn on the light source(s) and set up proper surface normals for all the models surfaces. Otherwise, the models might be dark (non-lit) or inconsistently illuminated and might look anywhere from somewhat wrong to completely invisible.

Getting Started:

Download the source code available for the course examples, particularly Chapter 6 from the link below:

http://interactivecomputergraphics.com/8E/Code/

You should be able to analyze and test all these programs using an HTML5 & WebGL compatible web browser and a computer with hardware-supported graphics rendering.

Grading:

Your program will be tested and graded using a standard WebGL platform. The grade will be based on your program's functionality (whether it works under different settings or not), as well as the efficiency of your implementation. The weights for different components are as follows:

1. Properly setting up the surface normals for the objects being shown on the scene. (20%)
2. Properly setting up different material properties for the objects being shown on the scene. (10%)
3. Properly showing two different light sources at fixed locations and with distinctive geometry that are used to illuminate the objects on the scene and can be turned on and off independently. (20%)
4. Properly setting up the lights above with an ambient light source component that is used to illustrate the objects on the scene and which can be turned from totally dark to intense brightness using a dial or a slider (from 0-total darkness to 100-total brightness, with a default of 30). (10%)
5. Properly showing one moving point light source that is used to illuminate some of the objects on the scene and illustrate the moving shadows projected on a ground plane. The moving point light should be shown as a 3D model (like a lightbulb has a shape) so that it is easy to see where the point light is located and should move up and down the Y axis. (10%)
6. Have a light source attached to the viewer during navigation (like a head lamp), setting up a spot light that can be turned on and off and can be controlled with the mouse. (10%)
7. Show properly the three shading modes: flat, Gouraud, and Phong shading, using the keyboard mapping described above. (10%)
8. Provide a basic interface to make use of the different light sources mentioned above. Make sure to insert buttons (5%) and keys (5%) to switch on and off each of the light sources (10% total).
9. Video competition: Produce a video that is meant to be distributed to the public, where you explain how to achieve one of the features of your assignment in a tutorial style. The winners of the competition (up to three participants will be chosen) will earn 15% extra points, will be featured in class, and will earn bragging rights. A source of inspiration can be found in the Guerrilla CG videos: https://www.youtube.com/channel/UC5fpWfCQ95VFghlkH1RG70w
   Feel free to post your submission directly on a free-access video platform, so that it can be more easily shown.

Explain your modes of operation in the bottom of the HTML file or in a README.txt file so the TA understands how to properly operate and evaluate your program.

What and How to Hand in:

You are handing in the source of your program, as well as any data files required for running your program. Your source code must contain sufficient internal documentation to facilitate grading. This includes your name, student number, a brief description of what the program does, which items you claim to have completed and a listing of known bugs, if any, at the top of the file. Send in your source program(s) through the Direct2Learn Dropbox as a single .zip file. No late submission is allowed.

If you choose to develop in Unity or Unreal, you will need to submit a 3-minute video of your project and the link to the source code of your project. The video should demonstrate your submission has the same functionality as requested above.

Assignment Submission Declarations:

- To avoid any confusion, in this assignment you will want to explain to the TA very clearly what options you are submitting and what is the total grade you are aiming for. You will also need to provide the link to the project you have completed if your project was created using Unity or Unreal. You can document this in the cover page of your assignment.
- After your text responses, provide a separate page titled "References & Sources", documenting all external sources, including Web resources and AI programs or assistants (such as Chat GPT). Even if there were no external sources used by you, the declaration must be part of the assignment, and in such a case, it must contain at least this statement "No external resources except those provided by the instructor were used to produce this assignment."

- Failure to provide any of these declarations above will result in an invalid submission for the part where the documentation is missing. For example, if the whole submission is in a Game Engine and the Github or google drive link to the submission is not provided, then the whole submission will get 50% of what a properly documented submission would get (ordinary grade * 0.5).

Verifying your submission:

Once you have placed your assignment into the Dropbox, click on the link that indicates our submission has been done, download it, open it, revise the contents, test that the program you have submitted actually is the one intended (not the source files, for example), and if it does not correspond to what you want the TA to mark, you must resubmit again, until you are satisfied, the TA will only mark the last submission with the timestamp that is prior to the indicated deadline.

Link to Github or google drive repository for Game Engine submissions. In the case that you are submitting a Unity or Unreal project, it is not practical to have the project sent through D2L, as this could require a couple gigabytes of storage. In these cases, you are asked to provide instead a link to the Github or google drive repository where each part of the submission is found. To reduce the chances of having your work plagiarized by someone else, your links must be private and give only access to yourself and the following two github accounts (instructor & TA): omerpas & AiurNanzatov or, for access to a google repository: omeruvia@mun.ca & aznanzatov@mun.ca

Late submissions are allowed. For each day of a late submission 15% of the grade will be removed. You may submit your assignment up to 6 days late (in which case, the assignment grade would be regular grade minus 90%).

Documenting and referencing your sources:

To avoid plagiarism issues, insert a disclaimer under a section called “Sources & References” explaining whether you adapted code or used information from any source other than the textbook’s author. Any source that is external to the course must be documented. All the source code submitted must be understood by yourself. You must be able to explain all elements of the code if this is deemed necessary by the markers or instructor after submission.