Assignment #01

Point Value: 90 points

Fall 2016

Due Date: Fri., Sep. 30, 11:59pm

Prof. James O'Brien

This assignment gets you started doing basic shading calculations. It is the basis for more complex shading models and you will use it again in later assignments.

1. **This assignment should be done in pairs or alone. You may share ideas with other groups, but you may not share code.**

2. **You may develop on Unix, Linux, OS X, or Windows. The platform you use will be the one used to grade assignments. Keep in mind that there are slight variations due to OS versions, different libraries, and other factors, so you should verify that your code runs on the instructional machines appropriate for you platform choice.**

3. **We will be using the submit software for submission of this assignment. Instructions for using the submission software are on the class website. You should include a README file that at the minimum contains the following information:**

    *   Your (and your partner's) name

    *   The platform your code runs on

    *   The location of your source code (i.e. indicate who in your group has done the submission, and on what platform). *Only one of the people in your group should submit the actual code. The other person should only submit the README file.*

    All files needed to compile your code should appear in the submitted directory. It is your responsibility to make sure that they will compile and run properly.

    *   Windows: The grader should be able to recompile your program by simply opening the project and rebuilding it from scratch.

    *   Unix and OS X: The grader should be able to recompile your program simply by typing "make".

4. **Do not wait until the last minute to start this assignment. This assignment should be easy, but it may not go as smoothly as you would like. Assume something unexpected will happen and give yourself time to deal with it.**

    Check the discussion group regularly for updates on the assignment or other clarification. We will assume that anything posted there is henceforth known to all.

5. **Once you have your assignment working, you should also update your class web page to include an "Assignment 01" link to a page with some images generated by your code. If you like, you may write code to output images directly from your software, use a library like lib-JPEG, or you can use screen shots. *Make sure that your web page images demonstrate all the features that you have implemented in your code.***

Assignment #01

Point Value: 90 points

Fall 2016

Due Date: Fri., Sep. 30, 11:59pm

Prof. James O'Brien

---

6. **Program Specification**

   For this assignment, you will write a program that:

   - Opens a window that contains an OpenGL rendering area.

   - Displays a circular shape that will be shaded using the Phong Illumination Model for point and directional lights.

   - The circular shape should occupy most of the window. If the window is resized it should update the display so that the shape still occupies most of the window and is still round. (In this context "most" would mean that the diameter of the circle should be 90% of the smaller of the window's height or width.)

   - When either ESC key or Q is pressed the program should exit.

   You program will take a series of command line options:

   - *-ka r g b*

     This is the ambient color coefficients of the sphere material. The parameters r g b are numbers between 0 and 1 inclusive.

   - *-kd r g b*

     This is the diffuse color coefficients of the sphere material. The parameters r g b are numbers between 0 and 1 inclusive.

   - *-ks r g b*

     This is the specular color coefficients of the sphere material. The parameters r g b are numbers between 0 and 1 inclusive.

   - *-spu pu*

     This is the power coefficient on the specular term in the u direction for an anisotropic material. It is a number between 0 and max_float.

   - *-spv pv*

     This is the power coefficient on the specular term in the v direction for an anisotropic material. It is a number between 0 and max_float.

   - *-sp p*

     This is the power coefficient on the specular term for an isotropic material. It is a number between 0 and max_float. (*i.e.* the same as setting *pu* and *pv* the the same value.)

   - -pl x y z r g b

     This adds a point light to the scene. The x y z values are the location of the light. The r g b values are it's color. Note that the x y z values are relative to the sphere. That is, the center of the sphere is at the origin and the radius of the sphere defines one unit of length. The Y direction is UP, the X direction is to the right on the screen, and the Z direction is "in your face." The r g b value are between 0 and max_float, NOT between 0 and 1 (that is, the r g b values encode the brightness of the light).

Assignment #01

Point Value: 90 points

Due Date: Fri., Sep. 30, 11:59pm

CS 184/284a: Foundations of Computer Graphics   page 3 of 4

Fall 2016

Prof. James O'Brien

- -dl x y z r g b

    This adds a directional light to the scene. The x y z values are the direction that the light points in. The r g b values are it's color. See -pl for coordinate system notes.

There may be up to 5 point lights and 5 directional lights (10 total) in a scene. Image r g b values of 1.0 should be mapped to a display values of 255.

All command line arguments are optional. The default values should be a black sphere with no lights. If material parameters are multiply specified, the last value should be used.

The coordinate system you use in your program should have a unit sphere at the origin and the viewer looking down the *Z-axis*. The *X-axis* points to the right of the screen, the *Y-axis* points up, and the viewer is located infinitely far away in the positive *Z* direction. You should assume that the vector from eye to the surface will always be *-Z*. If this description seems confusing to you, please see the instructor or one of the TAs *before* the assignment is due.

7. **Optional features that you can implement for extra credit are:**

    - Writing the image to a image file instead of to the screen. (Default would still be to the screen, but an optional command line argument would cause no OpenGL window to be open and file output instead.) The image format must be something commonly supported by most image viewers, for example JPEG, PNG, TIFF, or PPM. (Specified with -o filename)

    - Some sort of "toon" shading. (Specified as you think best.)

    - Other shapes besides spheres. (Specified as you think best.)

    - Multiple spheres. (Each specified with -s x y z r .)

    - Ashikhmin-Shirley materials. (Specified with -asm .)

    All features should be clearly documented in your README file. They should be implemented through additional command line options. The behavior of the arguments specified as required for the assignment should not be changed. (In other words your enhanced program must be fully backwards compatible with the original specification.) Your README file should explain what your optional features are and provide example command invocations that produce nice results. *Some images showing these results should also be posted on your web page for the assignment.*

8. **The class resource page has example code for you to start with. You may use this code if you like, or you can start from scratch. If you chose to use the example code then you are responsible for figuring out how to use it. If there is a bug in that code, please bring it to the attention of the TAs immediately. Please be advised that even if there is a bug in their code do not assume that you will be allowed to turn in the assignment late.**

Assignment #01

Point Value: 90 points

Due Date: Fri., Sep. 30, 11:59pm

9. This assignment will be graded in part by looking at your web page to see that you have posted examples demonstrating all required features and by running your code to test it on a specific set of test inputs. At the very _minimum_, your web page should have:

    - _An image showing diffuse only shading from a single light source_

    - _An image showing specular only shading from a single directional light source_

    - _An image showing specular only shading from a single point light source_

    - _An image showing combined specular and diffuse shading_

    - _An image showing anisotropic specularity specularity_

    - _An image with multiple light sources_

    Each image should have a caption explaining what is being shown and stating what command-line invocation was used to generate each image.

    If you work in a group, you should all link to the same web page and the web page should list your group members.

    Grading will include points for aesthetics and creativity as demonstrated on your web page.

10. _Be aware that putting images on your website that were not generated by your code is considered cheating and will be treated as such._

11. Questions should be posted to Piazza or emailed to cs184.

12. The following will be helpful:

    - If p is a point on the sphere and c is the center of the sphere, the normal is **n**=normalize(p - c)

    - If r is the radius, then r^2 = (p-c).(p-c)

    - The parametric v direction is v=normalize(y-n($n.y$))

    - The parametric u direction is u=normalize(cross(v,n))