# CS-411 – Assignment 5 (5%)

Illumination models

Due by: November 16, 2017

In this assignment you need to extend the surface viewer of assignment 4 to include a movable light source, light source properties, and material properties. Light source properties should include: ambient, diffuse, and specular properties (in RGB) as well as constant, linear, and quadratic attenuation factors. Material properties should include: ambient, diffuse, specular, emission properties, as well as a shininess parameter controlling specular highlights. The position of the light source, the parameters of the light source, and the material properties should be exposed for interactive control through buttons or keyboard keys. Answer the following questions and implement the necessary code. Submit the assignment using the submission instructions of the first assignment. Use the skeleton program of assignment 4. You are not required to use the skeleton program and may modify it as needed.

1. General questions (use JavaScript and/or manual computations):

   (a) Explain how you can make a light source stay at a fixed location or move with the camera.

   (b) Let $(0, 0, 2)$ be a position of a positional light source. Let $n = (0, 0, 1)$ be the normal at vertex $v = (4, 0, 0)$. Assume a white light source with ambient intensity $I_a = 0.2$, diffuse intensity $I_d = 0.3$, and specular intensity $I_s = 0.4$. Assume constant and linear attenuation factors of 1, and a quadratic attenuation factor of 0. Given a material with ambient, diffuse, and specular coefficients, 0f 0.5, 0.6, and 0.7, respectively, compute the intensity at the vertex $v$ due to ambient reflection.

   (c) Using the information in the previous question compute the intensity at vertex $v$ due to diffuse reflection.

   (d) Using the information in the previous question compute the intensity at vertex $v$ due to specular reflection. Assume a shininess coefficient of 1 and that the position of the light source is given by $(8, 0, 2)$.

   (e) Let $t = (v_1, v_2, v_3)$ be a triangle where $v_1 = (1, 2, 3), v_2 = (4, 5, 7), v_3 = (7, 8, 9)$. Compute a unit normal to this triangle assuming a counterclockwise winding order.

   (f) Let $v = (1, 2, 1)$ be a vertex with a normal $n = (0, 1, 0)$. Let a light source be located at $(2, 1, 1)$ and a camera located at $(0, 1, 1)$. Let the RGB Intensity of the light source be $I_d = (1, 2, 3)$, the diffuse reflection coefficient at the vertex be $k_d = 0.9$, and the specular reflection coefficient be $k_s = 0.2$. Compute the light intensity and the vertex due to diffuse and specular reflection (assume that there is no ambient light).

   (g) Describe a method for approximating the dot product between a reflected ray and the direction of viewing with fewer computations (using the half way vector).

   (h) Explain how the efficiency of intensity computations can be increased when assuming a distant viewer and/or a distant light source.

(i) Explain why we do not consider the ambient contribution of refracted light when computing the combined reflected+refracted light intensity at a location.

(j) Given a normal vector $n$, a vector $l$ pointing towards the light source, a vector $v$ pointing towards the viewer, and a vector $t$ indicating the direction of a refracted ray, write a formula for computing intensity due to specular and diffuse refraction (i.e. light transmitted through the surface). Assume that the intensity of the light source is $I$ and that the specular and diffuse refraction coefficients are both 1.

(k) Explain the main difference between Gouraud and Phong shading. Explain which of these models will produce more accurate results and which will be more efficient.

(l) Let $v_1 = (1, 2), v_2 = (4, 2), v_3 = (3, 6)$be the vertices of a 2d triangle after projection. Let the RGB light intensities in these vertices be $I_1 = (1, 1, 1), I_2 = (2, 2, 2), I_3 = (3, 3, 3)$. Compute the interpolated intensity at location $p = (3, 3)$ when using the Gouraud shading algorithm.

(m) Explain the difference between using vertex normals and face normals. Explain how vertex normals can be computed from face normals.

2. Program implementation (use JavaScript and WebGL):

(a) Write a program that performs surface rendering and which allows moving the light source using either keyboard keys, a slider bar, or buttons. Light source computations should be performed in the vertex shader.

(b) Add to the program the ability to control light source parameters by user interaction (assume a light source where the parameters for the different RGB components are identical).

(c) Add to the program the ability to control the light attenuation coefficients by user interaction.

(d) Add to the program the ability to control material properties by user interaction. (assume a material where the parameters for the different RGB components are identical).

(e) Note that the final program should support simultaneous modification of all the parameters specified above.

(f) Use the support code of the previous assignment and follow the submission instructions of the first assignment.