02561 COMPUTER GRAPHICS DTU COMPUTE

Worksheet 4: Lighting and (forward) shading

Reading	RTR: Sections 5.1-5.3 (and, optionally, Section 9.3)
Purpose	The purpose of this set of exercises is to get acquainted with lighting and shading in WebGPU and WGSL. We compute the local illumination of an object based on ambient, diffuse and specular material properties, and a light source.
Part 1	Draw a sphere in perspective view. • Start from Part 2 of Worksheet 3, which draws three wireframe cubes in perspective view. Simplify such that you draw just one cube in the image center and switch to drawing triangles instead of wireframe. • Draw a unit sphere instead of a unit cube using Loop-subdivision of a tetrahedron. • Insert two buttons: one which increments the subdivision level and one which decrements the subdivision level.
Part 2	Use depth buffer and back face culling to remove hidden surfaces. • Draw the vertex positions as colors ($\boldsymbol{c} = 0.5 \cdot \boldsymbol{p} + 0.5$). • Use the depth buffer to ensure that you are looking at the nearest part of the surface of the sphere. • Enable back face culling to improve efficiency. If your result is different from the previous step, you should flip the winding order of your triangles.
Part 3	Use Gouraud shading (with true normals) to draw a diffuse sphere lit by a distant, white, directional light with direction $(0, 0, -1)$. • Obtain the surface normal in the vertex shader. • Think of the color of the sphere as its diffuse reflection coefficient k_d. Introduce a distant light with direction $\boldsymbol{l}_e = (0, 0, -1)$ and light emission $L_e = (1, 1, 1)$, no distance attenuation. Compute the diffusely reflected light in the vertex shader and set the vertex color to this result (note that $\vec{\omega}_i = \boldsymbol{l} = -\boldsymbol{l}_e$). • Let the camera orbit the sphere over time.
Part 4	Implement the full Phong reflection model in the vertex shader. Select a diffuse and a specular color for the sphere and create (scaling factor) sliders for material parameters (k_d, k_s, s) and light parameters (L_e, L_a). Here, s is the shininess (Phong exponent). Use the "input" event instead of the onchange function if you prefer immediate response while using a slider (instead of only getting response when the slider is released). Use white for the light and just a single slider for each parameter. In addition, assume $L_i = L_e$ (no shadows) and $k_a = k_d$ in the model.

Part 5	Use Phong shading by moving your implementation of the Phong reflection model to the fragment shader and varying positions and normals across triangles instead of colors. Remember to re-normalize direction vectors that are varying and therefore linearly interpolated across a triangle.
Part 6	Answer the following questions: a) What is the difference between flat shading, Gouraud shading, and Phong shading? How would you implement flat shading? Based on your results in Parts 4 and 5, is Gouraud or Phong shading the best method for simulating highlights? b) What is the difference between a directional light and a point light? c) Does the eye position influence the shading of an object in any way? d) What is the effect of setting the specular term to $(0, 0, 0)$? e) What is the effect of increasing the shininess exponent (α)? f) In what coordinate space did you compute the lighting?