CSE 333/533 - Monsoon 2022
Assignment 3: Lighting and shading
Due date: 23:59, 7th Oct. 2022

The given code renders a cube in single color and no shading. Note that the camera can be rotated around the central object using mouse. The exercises below ask you to implement simple lighting and shading computations in OpenGL.

1. Switch-on lighting in the given program.
   (a) Replace the cube with the object that you had modelled in Assignment 1. Generate normals for the object.
   Lighting computations (e.g. in the Phong model) require normals along with vertex data. Compute per-vertex normals (remember to normalize the normals - i.e. make sure normals are unit vectors in all your computations). For a parametric surface $f(u, v) : \mathbb{R}^2 \mapsto \mathbb{R}^3$, the normal (normalized) at a point can be calculate as

$$\hat{n} = \frac{\partial f/\partial u \times \partial f/\partial v}{\|\partial f/\partial u \times \partial f/\partial v\|}.$$

   (b) Add a fixed point light source
   In programmable OpenGL, lighting computations are done entirely in the shaders. A point light source is identified by the location of the light and its color. Add these variables to your vertex shader as uniforms and pass-on certain values from your C++ program.
   (c) Use Gouraud shading for diffuse lighting.
   Gouraud shading is per-vertex lighting. Add lighting computations in your vertex shader to perform diffuse shading.

You should note that since the light source is fixed in space, you will be able to see the back side of the lit object. Notice the gradation due to diffuse lighting.
[Functionality: (a) 5 marks (b) 5 marks (c) 10 marks, Code quality and doc: 5 marks, Total: **25 marks**]

2. Use Phong lighting to render the object with specular highlights.
   Modify your vertex shader to include all three lighting components: ambient, diffuse, and specular. Use Phong exponent $\alpha = 32$. Notice that the back side of the object will not be pitch dark since you have added an ambient component.

[Functionality: 10 marks, Code quality and doc: 2.5 marks, Total: **12.5 marks**]

3. Modify your shaders to render the object using Phong shading.
   Use per-fragment Phong shading to generate a better shaded object. Again use the Phong lighting model with all three components. Any per-fragment computation has to be performed in a fragment shader, therefore you need to move shading computations from the vertex shader to the fragment shader.

[Functionality: 10 marks, Code quality and doc: 2.5 marks, Total: **12.5 marks**]

Deliverables (as a single zipped file **Assignment03_<studentID>.zip**) containing:

- C/C++ code (make sure to upload full code and do not include any intermediate object files, delete any other temporary files).
- 2~3 page PDF Report written with with **Latex/MS Word**. Use the acmlarge option (single column) (see sample-acmlarge.tex if writing with Latex). Include screenshots within the report itself (and DO NOT attach separately).

Total marks for this assignment: 50 marks

Bonus *(bonus marks to a maximum of 5 marks will be awarded for the following features. This part is completely optional)*

4. Visualize normals

   In order to visualize normal vectors on the object, you can convert these to colors and assign to vertices of the mesh object. A normalized normal vector $\hat{n} = (n_x, n_y, n_z)$ can be converted into a normalized color value as

$$c = \left(\frac{n_x + 1}{2}, \frac{n_y + 1}{2}, \frac{n_z + 1}{2} \right).$$

   No lighting computations are needed here and these should be computed in the fragment shader for better appearance.(for the same, you may want to pass-on u and v as vertex attributes.)

   [Functionality: 4 marks, Code quality and doc: 1 mark, Total: **5 marks**]

Note: Your code should be written by you and be easy to read. You are NOT permitted to use any code that is not written by you. (Any code provided by the TA can be used with proper credits within your program).