Homework 2
Topic: Lighting and Shading with GLSL

This is a programming assignment. At this stage, we assume that you have already read the preliminary reading material for Homework 2 (HW2_PR.pdf) and seen the preview html that shows a similar end result that you should obtain when you complete this homework.

Students are required to use Javascript (Three.js) for this assignment. You will be writing your code in the **index.html** file of the starter project that is provided to you.

Please document all your answers, plots, and insights in a **single pdf** file containing all requested results (read Submission section below before you prepare the pdf file!). Finally, please submit a .zip/.7z file with all source code and the pdf you have prepared (**<Student ID>_HW<number>.zip**, e.g. 118033991234_HW1.zip) to **Piazza**.

Introduction

In this task we will be implementing two sets of shaders for our objects. One set (vertex + fragment) will be for the Gouraud shading (per-vertex lighting) and the second will be for the Phong shading (per-fragment lighting). For both cases implement the Phong's lighting model (ambient term + diffuse/Lambertian term + Phong's specular term).

You will need the following normalized vectors in your shaders:

- v – the direction towards the viewer
- n – the surface normal
- l – the direction towards the light source
- r – the reflection of the light incident vector, see GLSL *reflect()*

The vector r can alternatively be the reflection of the viewer incident vector. The angle between the reflected viewer incident and l and the reflected light incident and v will be of the same size. Because cosine is symmetric, the specular term in the lighting model will be of the same value.
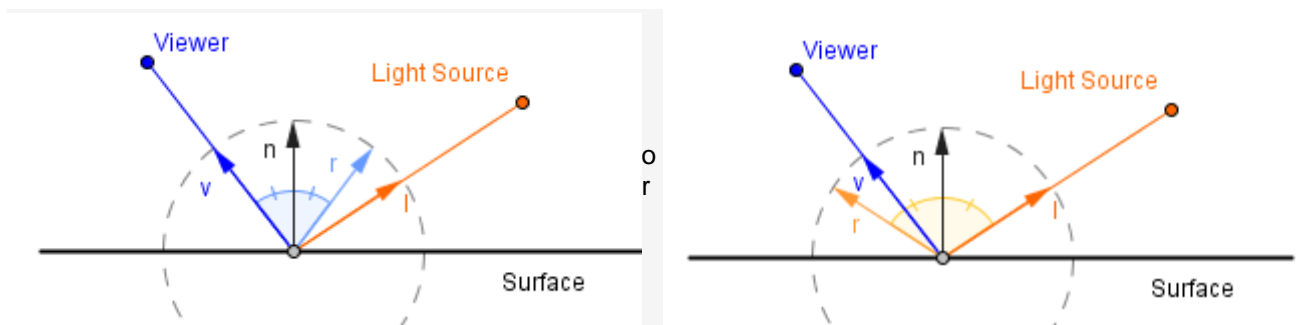
For the diffuse term you need to use the cosine of the angle between l and n

So overall you need to use the dot product $l \cdot n$ for the diffuse term and $v \cdot r$ for the specular.
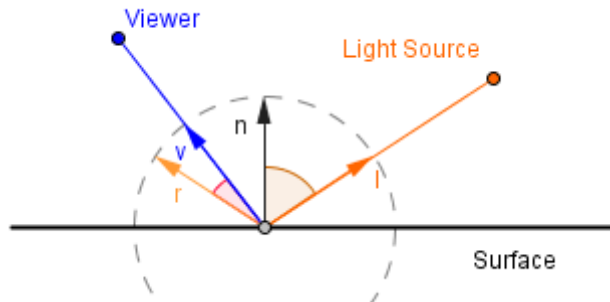
In order to see more clearly the difference between the Gouraud and Phong shading, let us make the body of the object to be a very crudely approximated sphere.

Tasks

The starter code will initially not work. First add the missing uniform variables to the shader, then proceed with the shader code.

You may want to change some parts of the chopper drawing functions to reflect your own chopper.

Do not use the THREE.MeshPhongMaterial or other presets.

Implement the aforementioned shaders to get a result similar to the images below. When implementing the shaders, remember that the calculations will be done in the camera space. This means that the viewer position is actually the origin.
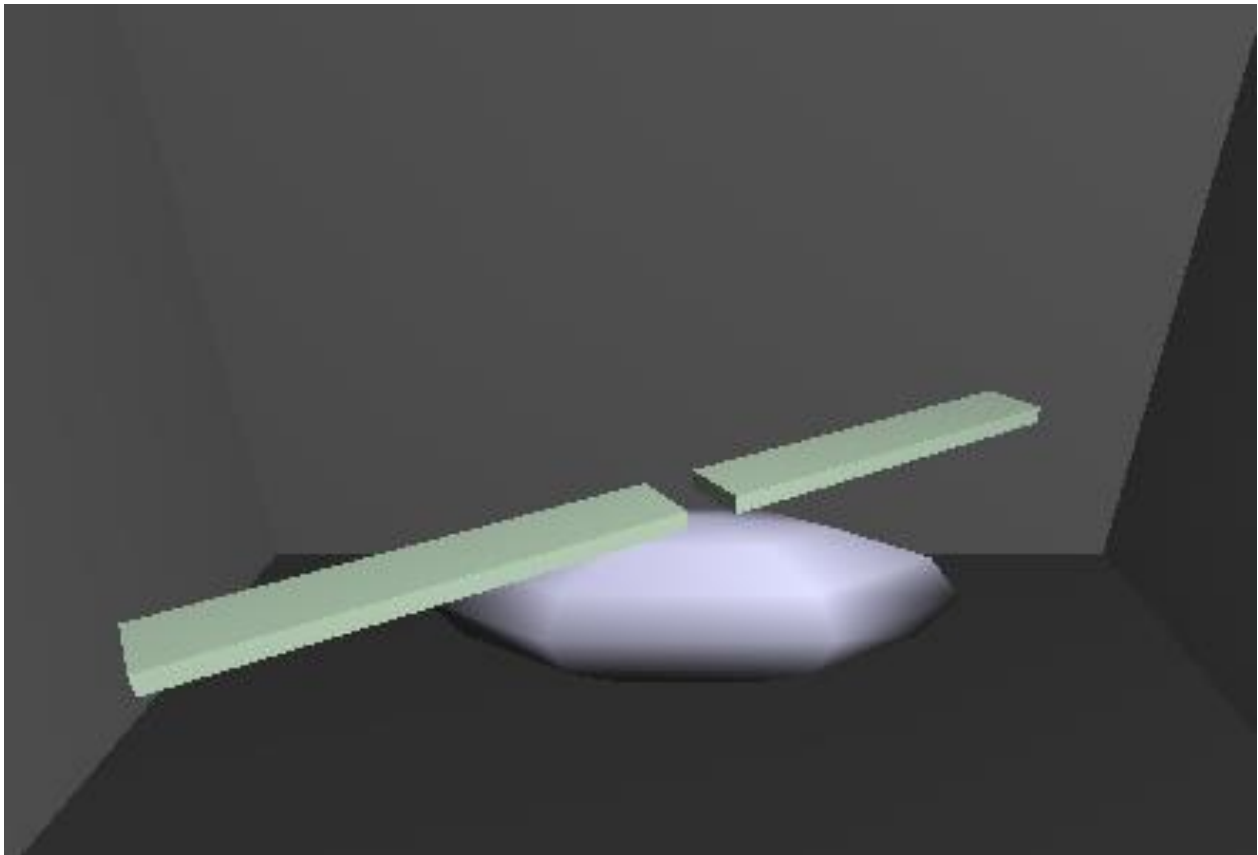
Task 1 of 3: Implement Gouraud Shading

Figure 1: Gouraud Shading (End result should look like this)
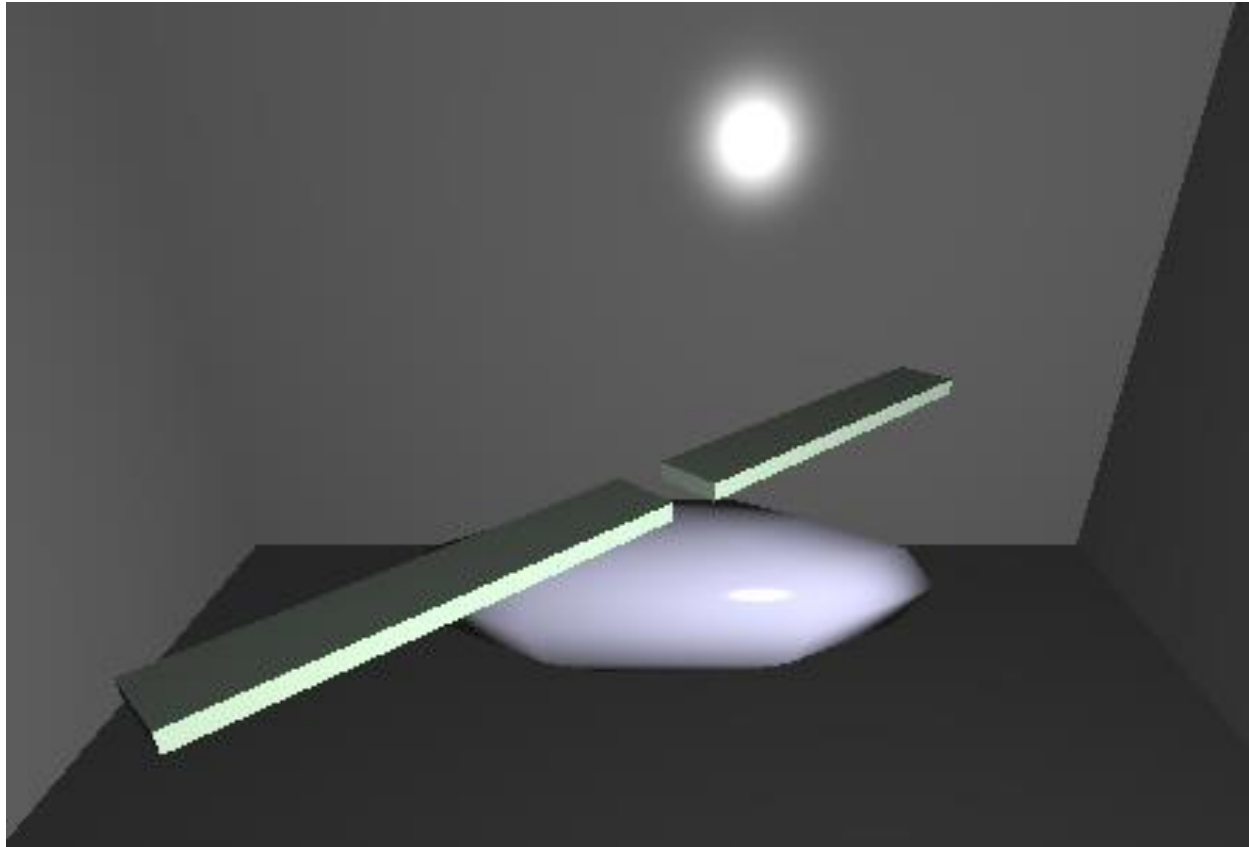
Task 2 of 3: Implement Phong Shading

Figure 2: Phong Shading (End result should look like this)

Task 3 of 3:

Describe in the task submission why Gouraud shading is missing a specular highlight on the back wall?

Submission

1. Add screenshots to the pdf of tasks 1 to 2.
2. Add your answer for task 3 in pdf.

Questions?

First, ~~Google~~ Baidu it! It's a good habit to use the internet to answer your question. For 99% of all question, the answer is easier found online than asking us. Also make sure to check out the helper resources we have added on the course webpage. If you can't figure it out this way, post your question on Piazza (preferably public question rather than a private one, so that everyone can benefit from the given answer) or send us an e-mail.