

Assignment 15: Direct Light Models

In this assignment you have to complete the Fragment Shader contained in file `shaders/PhongShader.frag`, to implement *Direct*, *Point* and *Spot* light models. The file must be compiled into `shaders/PhongFrag.spv` as seen in *Assignment 10*. For each light model, function ending with `..._light_dir()` should return a `vec3` vector containing the direction of the light for the considered point on the object, and the function `..._light_color()` should return a `vec3` containing the RGB color of the light, which includes both the hue and the intensity of the considered source. Both functions receives an argument `vec3 pos`, containing the world space position of the considered point on the object. An uniform block called `gubo`, contains all the parameters required to implement the light models. In particular:

- `gubo.lightDir` -> a `vec3` containing the direction of the light (for spot and directional lights).
- `gubo.lightPos` -> a `vec3` containing the position of the light (for spot and point lights).
- `gubo.lightColor` -> a `vec3` containing the basic color of the light.
- `gubo.coneInOutDecayExp.x`-> a `float` component containing the cosine of the outer angle of a spot light.
- `gubo.coneInOutDecayExp.y`-> a `float` component containing the cosine of the inner angle of a spot light.
- `gubo.coneInOutDecayExp.z`-> a `float` component containing the basic distance g for both spot and point lights.
- `gubo.coneInOutDecayExp.w`-> a `float` component containing the denominator exponent β for both spot and point lights: 0 for no decay, 1 for linear decay and 2 for quadratic fading.
- Object `gubo` also contains fields `eyePos` and `selector`, which however are not meaningful for the exercise and are required for other parts of the shader already implemented.

The following GLSL standard procedures can be helpful in solving this exercise:

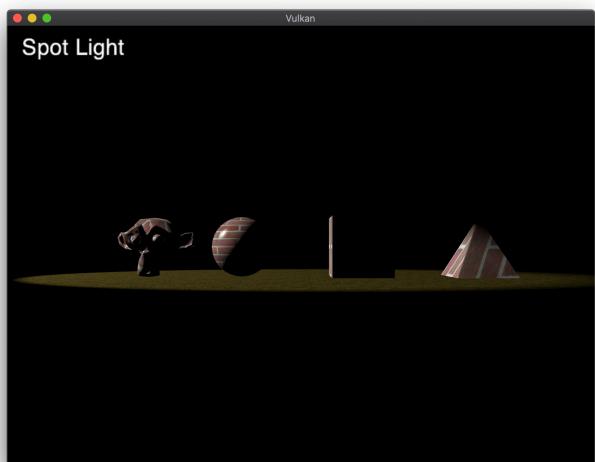
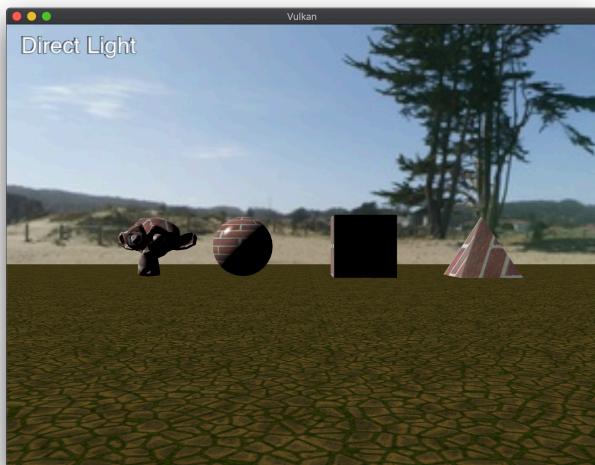
```
normalize()
pow()
dot()
length()
clamp()
max()
min()
```

If you need help about GLSL, you can refer to the following tutorial:

https://cgvr.cs.uni-bremen.de/teaching/cg2_07/literatur/glsl_tutorial/index.html

Starting from the section “Data Types and Variables”, at around 1/3 of the page. Please ignore what is presented before since it refers to a very old version of OpenGL which uses concepts that are now deprecated and not valid for Vulkan.

The expect results should be similar to the following:



Users can move the view using the same keys as in Assignment 0, and remove the texture pressing the **T** key.

ESC – quit the application		SPACE BAR – move to the next light			T: toggle the texture	
Q: roll left	W: forward	E: roll right	R: up		↑: look up	
A: left	S: backward	D: right	F: down	←: look left	↓: look down	→: look right