

Assignment 07: Light Models

In this assignment you have to implement the three basic light models in the GLSL file `shaders/PhongShader.frag`.

In particular, you should modify the six functions from line 42 to line 74. For each light model, one function computes the light ray, and another the color intensity. Some parameters of the model are in external variables, as described in the corresponding comments.

All functions receive two parameters: `vec3 pos` contains the position of the point on the surface for which the parameters of the light need to be computed, and `int i` defines the index of the considered light to retrieve its features.

`vec3 direct_light_dir(vec3 pos, int i)`
returns the direction of a Direct light. Its direction is contained in global variable `gubo.lightDir[i]`.

`vec3 direct_light_color(vec3 pos, int i)`
computes the color of a Direct light. Its color value is contained in global variable `gubo.lightColor[i].rgb`.

`vec3 point_light_dir(vec3 pos, int i)`
returns the direction of a Point light. The position of the light is contained in global variable `gubo.lightPos[i]`.

`vec3 point_light_color(vec3 pos, int i)`
computes the color of a Point light. The basic color of the light is defined in `gubo.lightColor[i].rgb`. Its scaling factor g can be found in `gubo.lightColor[i].a`. The decay power β is constant, and it is fixed to 2.0. The position of the light is contained in `gubo.lightPos[i]`.

`vec3 spot_light_dir(vec3 pos, int i)`
returns the direction of a Spot light. The position of the light is contained in global variable `gubo.lightPos[i]`, while its direction in `gubo.lightDir[i]`. The spot light extends a point light by confining its effect within the volume defined by an inner and an outer cones.

`vec3 spot_light_color(vec3 pos, int i)`
computes the color of a Spot light. The basic color of the light is defined in `gubo.lightColor[i].rgb`. Its scaling factor g can be found in `gubo.lightColor[i].a`. The decay power β is constant, and it is fixed to 2.0. The position of the light is contained in `gubo.lightPos[i]`. The direction of the light is defined in `gubo.lightDir[i]`. The cosine of half of the inner angle is specified in `gubo.cosIn`, while the cosine of half of the outer angle is given in `gubo.cosOut`. The last two parameters are identical for all the spot lights in the scene and source-dependent.

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

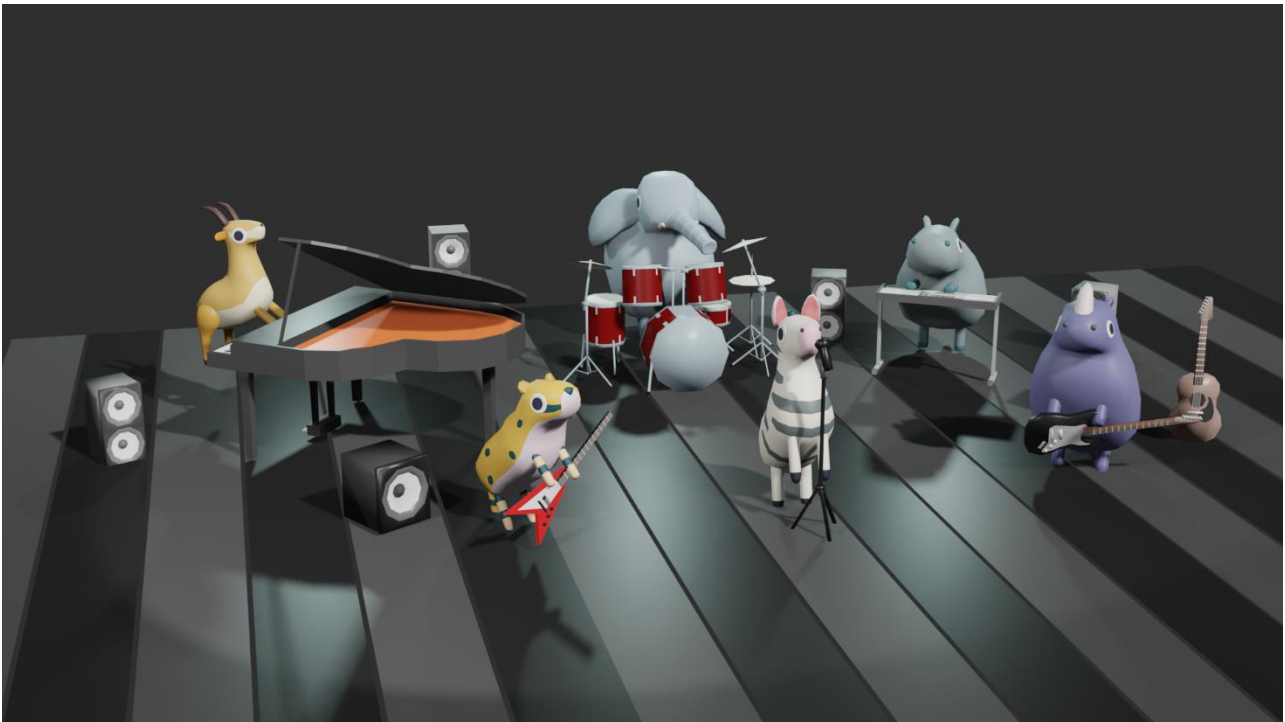
Starting from the section “Data Types and Variables”, at around 1/3 of the page. Please ignore what is presented before since it refers to a very old version of OpenGL which uses concepts that are now deprecated and not valid for Vulkan.

Shaders should be compiled using the following names:

Source	->	Binary
PhongShader.frag		PhongFrag.spv
PhongShader.vert		PhongVert.spv
TextShader.frag		TextFrag.spv
TextShader.vert		TextVert.spv

The scene contain a total of five light sources: a directiYou should be able to see an animal band, similar to the following¹:

¹ **Waring:** this is a view of the scene created in Blender. The correct solution of the assignment will be very similar, but not identical. For example, it will lack shadows, and it will use only an approximation of the indirect lighting.



You can move the view using either the keyboard, the mouse or a game pad, using the controls listed below. Moreover, keys from 1 to 3 will turn on and off the lights of a given type: this can help you in focusing on a specific light source at a time. When you are satisfied, press of the SPACE key to save the screenshots of your results in files **A07_1 .png** to **A07_4 .png**. Please check that their content matches your window, as such files will be an important part of the final delivery of this assignment.

