

Assignment 17: Ambient lighting

In this assignment you have to complete both the Vertex and the Fragment Shader contained in files `shaders/BRDFShader.vert` and `shaders/BRDFShader.frag`, approximate the rendering equation for three different materials and lighting conditions. The files must be compiled into `shaders/BRDFVert.spv` and `shaders/BRDFFrag.spv` as seen in *Assignment 10*.

First, the correct location ids for input and output should be defined both in *Vertex* and *Fragment Shaders*:

Vertex Shader			
in		out	
var	location	var	location
<code>inPosition</code>	0	<code>fragPos</code>	0
<code>inNormal</code>	1	<code>fragNorm</code>	1
<code>inTexCoord</code>	2	<code>fragTexCoord</code>	2

Fragment Shader			
in		out	
var	location	var	location
<code>fragPos</code>	0	<code>outColor</code>	0
<code>fragNorm</code>	1		
<code>fragTexCoord</code>	2		

Then, for each scenario, function ending with `Casex_BRDF()` should be implemented to return a `vec3` vector containing the RGB color computed by corresponding the model. Each function receives the following parameters:

- `vec3 N` : the direction of the normal vector (unitary vector)
- `vec3 V` : the direction of the view vector (unitary vector)
- `vec3 Ca` : ambient color of the surface

The functions receive also other parameters relevant to the particular model. Refer to the comments in the code for their definition. Lights parameters are instead defined inside a Uniform Block called `gubo`. The first two models are lit by a single directional light, while the last uses four directional lights. Each model uses a different ambient light model. In particular:

- `gubo.lightDir0` -> a `vec3` containing the direction of the first or only light (for spot and directional lights).
- `gubo.lightColor0` -> a `vec3` containing the basic color of the light first or only light.
- `gubo.lightDir1, gubo.lightDir2, gubo.lightDir3` -> three `vec3` containing the direction of the second, third and fourth lights (for model 3).
- `gubo.lightColor1, gubo.lightColor2, gubo.lightColor3` -> a `vec3` containing the basic color of the second, third and fourth light.
- `gubo.AmbColor` -> a `vec3` containing the basic ambient color. For model 2, it corresponds to the bottom color, and for model 3 to the constant component for Spherical Harmonics lighting.

- `gubo.TopColor` -> a `vec3` containing the top ambient color for model 2, and to the difference related to the `y` component of the normal direction for Spherical Harmonics lighting in model 3.
- `gubo.DzColor`, `gubo.DxColor` -> two `vec3` containing the difference related to the `z` and `x` components of the normal direction for Spherical Harmonics lighting in model 3.
- 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 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