Computer Graphics - 2016/2017

Exercise 2 - Part B

Last update: Dec 21, 16:00

In the previous exercise we created an application to intuitively view a 3D model. Now we will shade these models using the Phong lighting model and Gouraud shading.

Detailed description

Light Sources:
There will be two light sources in the scene, starting at locations specified below.

Normal Estimation:
Start by evaluating a normal for each polygon in the mesh (OpenMesh can help you with that). The basic normal mode will use the face normal for each vertex, so when the polygon is drawn, all its vertices have the same normal.

The advanced normal mode (toggle between models by pressing ‘N’) will use the average of face normals for each vertex (OpenMesh can help you with this as well). So when the polygon is drawn each vertex will have its own normal (average of the faces this vertex belongs to).

Shading:
In this exercise you will shade your model using two different methods: The full **Phong** model (where the final pixel colors are computed per pixel from interpolated normals), and its **Gouraud** approximation (where colors are computed at the vertices and then interpolated to get the final pixel colors).

The user will have keyboard control over the shininess exponent of the Phong shading mode (by using the keys ‘-’ and ‘=’).

Important: Your program should support all features that were defined in the previous exercise.

Keyboard Operations:

R / r : Reset to the initial view of the model: rotation, translation and zoom

Q / q : Quit the program. Simply call exit().

W / w : Toggle between wireframe and full mode.

P / p: Toggle between perspective projection (default) and orthographic projection. You should make sure that the orthographic projection is "compatible" with the current perspective view, i.e. it displays the object at roughly the same size, and responds similarly to zooming in and out.

'N'/'n': Switch between normal estimation modes (i.e. per vertex / per face)

'1': Use the full Phong model

'2': Use the Gouraud approximation

'3': Assign pixel colors as defined in the previous exercise.

'=': Increase the shininess coefficient up to 2000.

'-': Decrease the shininess coefficient down to 0.

Mouse operations:

Mouse events should be interpreted as defined in the previous exercise (with the appropriate changes to support the new modes defined here).

Lighting model parameters:

Use the following values as the initial parameters for lighting:

```
lightColor1 = vec3(1.0, 0.9, 0.7); // First light color
lightColor2 = vec3(0.6, 0.6, 1.0); // Second light color
ambientColor = vec3(1.0, 1.0, 1.0); // Ambient light color

light_position1 = vec3(  3.0, 2.0,  1.0); // First light position
light_position2 = vec3( -3.0, 0.0,  1.0); // Second light position

ka = vec3(0.1, 0.1, 0.1); // Ambient coefficient
kd = vec3(0.3, 0.3, 0.3); // Diffuse coefficient
ks = vec3(0.3, 0.3, 0.3); // Specular coefficient
specExp = 200.0; // Default Shininess (can be changed dynamically by the user)
```

Note that the first five values are properties of the environment (light colors and their positions) and the last four values are properties of the object material.

Submission

Include the following in your submission

1. A Readme.txt file, that includes:

- Your id and login
- Your partner's id and login
- A brief description of your implementation, which piece(s) of code you started from and what changes you made to it (/them).

- All files that are required for compilation of your solution with a single ‘make’ command, and the shaders necessary to run it.

Pack all files as a single zip file named by the following pattern: ex2b_<your 9 digits id>_<your_username>_<your partner’s 9 digits id>_<your partner’s 9 digis login>.zip (e.g. ‘`ex2b_123456789_mylogin_987654321_myfriendlogin.zip`’).

School Solution:
You can find the school solution here. Before running the school solution, run “`source ~cg/setpath.csh`“.

Deadline:
You have to submit your solution (via the course’s moodle webpage) no later than Tuesday 27/12 at 23:45. Late submission will result in 2^(N+1) points deduction where N is the number of days between the deadline and your submission (rounded up, the minimum grade is 0, friday and saturday are excluded).
Good luck and have fun!