Interactive Graphics

Homework 1

Online April 10th, 2020
Deadline: **Sunday May 3rd, 2020 (23.59, Rome time zone)**

Tasks to do

The homework must be completed alone. Each student should do its own homework and NO CODE SHARING IS ALLOWED. Submissions will be checked for plagiarism and suspicious ones will be rejected and reported. You cannot use code taken from the web, the only code you are allowed to use in your submission is the initial code provided with the assignment and the code of the book stored in Classroom. You can, however, access all the documentation you want (including the WebGL and GLSL official documents on https://www.khronos.org/).

The assignment material includes this PDF file, the publication mentioned in the next page and two directories, Homework1 (containing the files homework1.html and homework1.js) and Common (containing the files MVnew.js and initShaders.js). You need only to modify the two files (homework1.html and homework1.js), add the texture used for point 6 and add a short documentation in PDF format (more details at the end of this file). Please **do not change the names of the files**, you only need to modify their content.

You need to modify the files so to obtain the following effects.

1. Replace the cube with a more complex and irregular geometry of 20 to 30 (maximum) vertices. Each vertex should have associated a normal (3 or 4 coordinates) and a texture coordinate (2 coordinates). Explain how you choose the normal and texture coordinates.
2. Add the viewer position (your choice), a perspective projection (your choice of parameters) and compute the ModelView and Projection matrices in the Javascript application. The viewer position and viewing volume should be controllable with buttons, sliders or menus. Please choose the parameters so that the object is clearly visible.
3. Add two light sources, a spotlight in a finite position and one directional. The parameters of the spotlight should be controllable with buttons, sliders or menus. Assign to each light source all the necessary parameters (your choice).
4. Assign to the object a material with the relevant properties (your choice).
5. Implement a per-fragment shading model based on the shading model described at the end of this document.
6. Add a texture loaded from file (your choice), with the pixel color a combination of the color computed using the lighting model and the texture. Add a button that activates/deactivates the texture.

Describe your solution in a short PDF document (2-3 pages) describing the techniques used, the advantages and disadvantages of the proposed solution and the features of your solution.

How to submit the homework

Don't post solutions or code on Classroom. Use Classroom only for questions and clarifications. Do not ask for clarifications or comments by email, use only Classroom

Shading model to use

The following shading model is a simplified version of the one presented by Lake, A., Marshall, C., Harris, M., & Blackstein, M. This is the version you need to implement:

ALGORITHM Simple Cartoon Shade

1. Calculate the illuminated diffuse color:

$$C_i = a_g \times a_m + a_l \times a_m + d_l \times d_m$$

2. Calculate the shadowed diffuse color:

$$C_s = a_g \times a_m + a_l \times a_m$$

3. Compute the value of:

$$Max\{\overline{L} \cdot \overline{n}, 0\}$$

   If this value is greater or equal than 0.5, assign to the fragment the value $\mathbf{C_i}$, otherwise assign to it the value of **Cs**

Here, $\mathbf{C_i}$ is the vertex color, $\mathbf{a_g}$ is the coefficient of global ambient light, $\mathbf{a_l}$ and $\mathbf{d_l}$ are the ambient and diffuse coefficients of the light source, and $\mathbf{a_m}$ and $\mathbf{d_m}$ are the ambient and diffuse coefficients of the object's material. **L** is the unit vector from the light source to the vertex/fragment, and **n** is the unit vector normal to the surface at the vertex/fragment.

Lake, A., Marshall, C., Harris, M., & Blackstein, M. (2000, June). Stylized rendering techniques for scalable real-time 3d animation. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering* (pp. 13-20). https://dl.acm.org/doi/pdf/10.1145/340916.340918