Interactive Graphics

Homework 1

Online April 10th, 2023
Deadline: **Sunday April 31st 2023 11.59pm 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. You can, however, read all the documentation you want (including the WebGL and GLSL official documents on https://www.khronos.org/), but not use the code included.

To complete the assignment you need to use GitHub Classroom, start by creating your own repository in the GitHub Classroom of the course by clicking on this link https://classroom.github.com/a/qtUfD5fv, please select your name from the list (taken from Google Classroom)**.** If you are not registered in Google Classroom please do so by going to https://classroom.google.com/c/NTkyMTg2NzIwNzQ3?cjc=x544cbf and email the teacher so that he can enter you in GitHub Classroom. The assignment material includes this PDF file and two directories, Homework1 (containing homework1.html and homework1.js) and Common (containing MV.js and initShaders.js). You need only to modify the two files (homework1.html and homework1.js), add the texture 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 simple model of a chair of 20 to 40 (maximum) vertices. Each vertex should have associated a normal (3 or 4 coordinates. Explain in the document how you chose the normal coordinates. Model the chair in its own coordinates.
2. Choose an origin (different from the origin of the chair and include the rotation of the chair around the origin and along all three axes. Control with buttons/menus the axis of rotation, direction, speed, and start/stop.
3. 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 initial parameters so that the object is clearly visible and the object is completely contained in the viewing volume. By changing the parameters you should be able to obtain situations where the object is partly or completely outside of the view volume.
4. Add a spotlight, and place it outside of the view volume. Assign to it all the necessary parameters (your choice) including an attenuation factor. Add a button that turns the light on and off and sliders to control the angle of the opening and the direction of the spotlight.
5. Assign to the object a material with the relevant properties (your choice).
6. Implement both per-vertex and per-fragment shading models. Use a slider (0 to 100) to combine the two computed colors.
7. Implement a quantization mechanism, activated by a button. If quantization is active, let the user select 8 different colors and all fragments can only be assigned one of these 8 colors. More precisely, compute the color of each fragment as in point 6 and the replace the final color with the closest color of the 8 chosen ones.

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

The solution should be delivered on the GitHub Classroom repository. Do not post solutions or code on Google Classroom. Use Google Classroom only for questions and clarifications. Do not ask for clarifications or comments by email, use only Google Classroom