Stefan Johansson
stefanj@cs.umu.se

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE

3D Studio—Part 3
Shading and more

Please, read the whole project description before starting the implementation.

Purpose

This is the third and final part of the project. The aims of Part 3 are to add different lighting models, shading techniques, and textures to your 3D object. Especially you will get a good understanding of the different steps in the graphic pipeline. In addition, one purpose of the final part is to practice to present and discuss the design and functionality of the final result in a small group.

Specification

Extend your software with the following (in addition to the tasks in Parts 1 and 2):

- ~~Implement per-pixel shading using Phong (or Blinn-Phong) lighting model in the fragment shader. Remember to only add specular light if diffuse light (lambertian) is greater than zero, `n·l>0`.~~
- If normals are not provided in the OBJ file, they have to be computed.
- ~~Change default material of the object and properties of the light source by a GUI.~~
- Add a texture of your choice to the object. The user should be able to choose in the GUI if the object is shown with a texture or not. Compute the texture coordinates by using the Two-part texture mapping method (not using the texture coordinates in the OBJ file). It does not need to look perfect and it is not mandatory to read the MTL file.

Any extension or alternation of the functionality is encouraged as long as it does not limit the usability or intention of the assignment. Good resources of free textures are https://polyhaven.com and https://3dtextures.me/.

Bonus tasks

The following extensions give bonus points on the first written exam. Bonus points are only given if Parts 1 and 2 are completed and demonstrated in time and the basic functionalities work. The bonus points can only be used to get a higher grade (not to pass) the exam. Any other addition is also encouraged and can render bonus points if discussed with the tutors **before** handing in the project. A maximum of 6 bonus points are given.

One (1) bonus point each:

- Add the possibility to change between different shading models, for example, per-vertex (Gouraud), per-pixel (Phong), and none (wireframe).
- Add a ground floor below your object. The floor can be of any color or have a texture. (A ground floor makes it easier to navigate in the world.)
- Implement Toon shading.
- Add a Sky box.

Stefan Johansson
stefanj@cs.umu.se

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE

- Read and use the texture coordinates (if present) in the OBJ file.
- Read and use the MTL file.

Two (2) bonus points each:

- Several light sources, including one that moves along a parametric curve, e.g., a sequence of segments forming a closed curve.
- Activate smooth movement of the camera using parametric curves (see last page).
- Several objects in the same scene and/or read OBJ files with compound objects (several objects in the same OBJ file).
- Add mirror or shadow effect on the ground floor and/or object.

The user interface

The GUI from Part 2 is now going to be extended with easy adjustments of the parameters for material, textures, and light properties. It should also be possible to show the object with texture.

An updated GUI is available for you to use. Code for this can be found under 'Files/Project Resources' on Canvas, one file for Qt and one for GLFW. The layout of the GUI may be changed or extended as long as the original functionalities are included. A good idea can be to change the default values to something you think is more appropriate. See the *_Readme.pdf file in each package for instructions on how to integrate the code into your existing project.

A note on using texture coordinates

If you plan to use the texture coordinates in the OBJ file, then you have to read this.

In the OBJ file, two vertices can have the same position and normal but different texture coordinates. That happens, for example, in the seam between two textures, where the texture is different depending on which polygon the vertex belongs to. In that case, the “two vertices” (even if they are actually the same vertex) must have different index number. This since the same index is used for both the vertex position and the corresponding texture coordinate.

Oral presentation

The result of the project (parts 1 to 3) must be presented in a smaller group. Notice, your application does not have to meet all the requirements listed under the specification, but be in a state such that the idea is made clear and the object is visible. The group will consist of 4-6 students and the instructor. The presentation must fulfill these requirements:

1. You do your presentation logged in at the CS’s Linux system on a by us provided Computer, or if you are presenting over Zoom on your own computer.
2. The presentation should take around **15 minutes**.
3. You give a **demonstration of the software** from the project that shows the required functionalities, and extra features if present. We focus on the new features in part 3.

Stefan Johansson
stefanj@cs.umu.se

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE

4. Make sure you have **at least two 3D models** (OBJ files) prepared. If you use the texture coordinates, (at least) one should have none (showing that you can apply a texture without pre-computed texture coordinates).
5. You should also prepare a **short presentation** consisting of 2-4 slides. It must contain a **presentation of your system design** (1-2 slides), **which (if any) bonus tasks you have added,** and **your own reflections** of the chosen design and the project (1 slide). Focus on the responsibilities of different components, generality, modularity, etc. You may use UML diagrams or similar so illustrate your system design, but you do not need to.

The presentation is done Wednesday January 13 or Thursday January 14, 2022. You can sign up for a slot in the Canvas Calendar, starting on January 3. If you do not have specific requirements for a specific time, please wait a day or two before signing up. You may also give the presentation over Zoom. If so, you must notify me (Stefan) before the session starts.

At the end of the presentation, we will do a short oral course evaluation.

External Libraries and Licenses

If you use any existing libraries then it is important that you read the license for each library and check in which circumstances you are allowed to use it. An appropriate disclaimer and any necessary license file(s) must be included in your final source bundle.

Instructions

This is an individual assignment. The code should follow good programming practice and if you are using any libraries that are not pre-installed, include them in separate subfolders, for example ./lib and ./include, and set appropriate parameters in the Makefile. No written report is required.

Post your **finished solution** as a zip-file on Canvas no later than **January 14, 2022, 17:00**. The code is **reviewed** and comments on the solution is given on Canvas. *Clean up the code and remove all garbage code, irrelevant comments, etc.*

Timeslots for the **oral presentation are provided January 13 and 14, 2022, in B212** (CS department's conference room Lovelace on 2nd floor, MIT-building) as well as on Zoom. You reach B212 from the entrance on your right when you have entered the MIT-building's main entrance.