Proseminar Computer Graphics
Summer Semester 2015

Programming Assignment 3
Hand-out: May 15th, 2015
Hand-in: June 5th, 2015, 11:59pm

Topics

- Color representations
- Light sources
- Lighting/reflection models
- Shading approaches

Outline

The overall goal of the programming project during the semester is to create a shader-based OpenGL program showing an animated merry-go-round. This target will be realized in four steps, each representing a programming assignment. In each step the complexity of the scene will be further improved, until a final full 3D animated model with several effects is realized.

The general idea of the third assignment is to include lighting and shading in the rendering to enhance the visual realism of the previously implemented animated scene. Key extensions are light sources, color handling, computation of reflection, and triangle shading. Appropriate code should be developed following the shader-based OpenGL programming paradigm. The implementation should extend the program developed in the initial two assignments of the Proseminar. A main target is to get acquainted with programming vertex and/or fragment shaders to achieve improved object appearance.

Tasks

1. Add light sources to your rendering scene. There should be at least two positional light sources, but you may add more. The emitted light of one positional light source should be a chromatic color. Also, one positional light source should be animated with the merry-go-round. Depending on the light source you may want to represent it geometrically (e.g. by a sphere).

2. Provide functionality that permits a user to interactively adjust parameters of emitted light. A user should be able to change hue and value of a light (you can assume fully saturated colors). Note that the scene lights will only take effect in combination with the lighting/reflection models to be implemented in the third task.

3. Implement either Gouraud or Phong shading in combination with the Phong reflection/lighting model. Assume different reflective properties of the objects on your merry-go-round (i.e. by varying ambient, diffuse, specular terms). Provide functionality to selectively turn on and off the ambient, diffuse, and specular components of the rendering.

Submission of your solution is due on June 5th, 2015 at 11:59pm. Please submit all developed source code (i.e. no executables) in a ZIP file via OLAT.

Development should be carried out in teams of 2-3 students – please continue in the same teams or get in touch if any changes are required. Make sure to acknowledge all team members in the code and submission (please name the ZIP file with all surnames). Only one submission per team is required. Provide adequate information on program usage.

Finally, please respect the academic honor code. Each team should work on its own on the solution of the assignment. Any attempts at plagiarism or collusion will lead to 0 marks and further scrutiny. Please get in touch in case of questions or problems with regard to the assignment.

In total there are 20 marks achievable in this assignment.

Implementation Remarks

Your solution should be developed using OpenGL 3.3 and GLSL 3.3. You can check available drivers and versions on Linux with the command `glxinfo`. All required libraries are installed on the machines in the ZID computing labs, e.g. RR14, RR15, RR19, RR20.

While you are free to develop in any environment, the final submission should compile and run in Linux. As reference, the machines in RR15 will be considered. Make sure to provide a Makefile with your submission, so that after unpacking your solution can be compiled directly in your top-level directory via `make`.

Test early enough in case you develop in a different hardware environment. If you intend to work on your own setup, it is preferable to employ vendor-specific (e.g. Nvidia, AMD) driver implementations (instead of using e.g. Mesa OpenGL). Also note that in addition to OpenGL, the packages freeglut and GLEW are used in the examples. For the solution a shader-based programming approach should be targeted. Follow a proper coding style and provide appropriate comments in the code.

Resources

- Lecture and Proseminar slides as well as code and information is available via http://igs.uibk.ac.at/ (→ Teaching)
- OpenGL homepage
  http://www.opengl.org
- OpenGL 3.3 reference pages
  http://www.opengl.org/sdk/docs/man3
- Legacy information
  http://www.opengl.org/wiki/Legacy_OpenGL
- OpenGL Tutorial
  http://www.opengl-tutorial.org
- GLUT implementation freeglut
  http://freeglut.sourceforge.net/
- OpenGL Extension Wrangler GLEW
  http://glew.sourceforge.net/
- D. Shreiner, G. Sellers, J. Kessenich, B. Licea-Kane, "OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3" Addison-Wesley, 8th edition, 978-0321773036, 2013.

Note: Be mindful of employed OpenGL and GLSL versions!