

Proseminar Computer Graphics Summer Semester 2015

Programming Assignment 1

Hand-out: March 22nd, 2015

Hand-in: April 24th, 2014, 11:59pm

Topics

- General OpenGL and GLSL programming
- OpenGL program structure
- Setup of rendering model
- ModelView transformations
- Basic animation



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 first assignment is the setup of a simple 3D scene with basic objects representing a merry-go-round. The geometrical model should be hierarchical, comprising at least two levels of animation (i.e. rotation of base platform and further animation of individual objects). Thus, add random, but plausible animations to the objects. Appropriate code should be developed following the shader-based OpenGL programming paradigm. The implementation can start based on the provided initial code examples in the Proseminar. The main target is to get acquainted with basic elements of OpenGL programming and to implement the theoretical concepts taught in the lecture.

Tasks

1. Setup the hierarchical geometrical model. You have some freedom in how to design your model of the merry-go-round. Basic 2D/3D objects should be used as decorative elements. The merry-go-round should have at least two levels.
2. Add animation to the merry-go-round. The modeling of the underlying physics is not required. The animation can be random, but should appear plausible.

Note that advanced lighting computations are not required for this assignment; these will be a topic of a later tasks. But we will see that these enhance the visual realism.

Submission of your solution is due on April 24th, 2015 at 11:59pm. Please submit all developed source code (i.e. no executables) in a ZIP file via OLAT (access will be made available soon).

Development should be carried out in teams of 2-3 students. Please get in touch if you are unable to find a team. 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 15 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 if 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!