

**Proseminar Computer Graphics
Summer Semester 2015**

Programming Assignment 2

Hand-out: April 17th, 2015

Hand-in: May 15th, 2015, 11:59pm

Topics

- OpenGL programming
- Camera handling
- User interaction



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 second assignment is the setup of a basic interactive camera, to be controlled by a user. The camera should have two modes – an automatic mode orbiting around the model, as well as a user controlled mode. Appropriate code should be developed following the shader-based OpenGL programming paradigm. The implementation should extend the program developed in the first assignment of the Proseminar. The main target is to get acquainted with elements of camera handling and user interaction.

Tasks

1. Further improve the geometrical model of the merry-go-round. You may include more complex models loaded via the provided OBJ loader. Add a fixed background structure (e.g. walls and floor) to provide a static reference for the various rotations (note that the rendering will become more realistic in the next assignment when lighting will be implemented).
2. For the first camera mode, add a reasonable automatic camera path. The camera should move around the scene in an engaging fashion, partly controllable by some user input, such as speed of animation, inserting pauses, switching direction.
3. The second camera mode comprises full user control. Implement an interactive camera that can be controlled via mouse and keyboard by the user. Consider various options, e.g. having a fixed focus on the scene center or a fully free camera motion.

Submission of your solution is due on May 15th, 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 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 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 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!