



## 3D Studio

### The Project

#### Objectives and introduction

During the course the assignments call for you to write a limited 3D studio software. This should make you familiar with 3D transformations, lighting, and shading, and how these are implemented in OpenGL. The project is divided into three parts and has the following requirements.

- This is an individual project.
- The project must be written in C++. You will be using Qt as OpenGL window manager.
- It is allowed to use any external libraries.
- The complete project (including all three parts) must be presented and demonstrated in the end of the course. More info about this later on in the course.

If the first two parts are **completed and demonstrated** no later than the due date, bonus points are available for the third part. Bonus points can be used to get a higher grade on the first written exam, but not to pass. The demonstration of the first two parts will be in the CS department's computer labs MA416 and MA426.

#### Use of 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.

#### Tips and guides

The reference pages for all OpenGL functions and the useful Quick Reference Card can be found at

<https://www.opengl.org/sdk/docs/>.

For each command and function the supported version of OpenGL is listed at the corresponding reference page. For this course version 3.3 is sufficient, however you may use up to version 4.3 depending on the support of your graphics card (note that the reference pages are for OpenGL 4.5+, so some functions cannot be used).

The glsl-files follows the specification of the *OpenGL Shader Language* (GLSL). If you have problems with data types or any compilation errors, please take a look at the following links:

[https://www.khronos.org/opengl/wiki/OpenGL\\_Shading\\_Language](https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language)  
(Introduction)

[https://www.khronos.org/opengl/wiki/Data\\_Type\\_\(GLSL\)](https://www.khronos.org/opengl/wiki/Data_Type_(GLSL))  
(GLSL data types)

(The GLSL specification and Quick Reference Guide are also included in OpenGL's reference pages)

The reference pages for Qt can be found at <http://doc.qt.io/qt-5/reference-overview.html>. The installed version of Qt in the computer labs is 5.7.1. To compile the code you must use Qt's *qmake* which generates a Makefile (<http://doc.qt.io/qt-5/qmake-manual.html>). Qmake only need to be run once (as long as you do not edit the project file), after that you use the make file as usual to compile. Alternatively and to simplify the coding you may use Qt Creator (<http://doc.qt.io/qtcreator/index.html>). In that case, do not forget to copy the shader-files to the build directory or change the path to them in `GeometryRender::initialize()`.

An alternative to the build-in math functions in Qt is *OpenGL Mathematics*, GLM, which is a C++ header library based on the GLSL specification that can be of great help (<http://glm.g-truc.net/0.9.8/api>).

### **OpenGL on your own computer**

If you like to use your own computer you must consider the following

- Always test your code in good time before the deadline on the computers in MA416 (MA426)! You may not get the same results.
- Make sure your graphics card supports at least OpenGL 3.3 and GLSL 3.30.
- OpenGL does not normally work over an X-tunnel. It must be run locally.
- If you use a virtualization environment like Oracle VirtualBox, make sure that it supports OpenGL 3.3.
- Linux is preferable, but Windows and Mac should work fine.
- Usually a default graphics driver is installed by default, but it may not support hardware acceleration. Install the right graphics driver. On Linux this can be nothing but tricky. See if you find instructions and follow it. Also install mesa-utils. You can verify that it works with 'glxinfo | grep OpenGL'. It should confirm the versions supported. You can also try 'glxgears'.
- Install the required libraries (qt, etc.) and type 'make'. Pray to your god(s).