

## Zu den Tutorials

Diese Tutorials soll Ihnen dabei helfen, die Grundlagen von OpenGL 3.3 unter C++ zu verstehen. Gleichzeitig wird das in der LVA benutzte Framework vorgestellt – damit ist hier im Wesentlichen eine Sammlung von Open Source Libraries gemeint, welche so wichtige Funktionalitäten bereitstellen wie das Erzeugen eines OpenGL Context, Matrizen- und Vektorrechnung und das Laden von 3D Assets.

In einigen Schritten wird gezeigt werden, wie eine – extrem einfache – Beispielgeometrie gezeichnet werden kann.

- Schritt 1: Ein Fenster erzeugen & OpenGL initialisieren.
- Schritt 2: GLSL Shader zu einem ausführbaren GLSL Programm linken und das grundlegende OpenGL Setup.
- Schritt 3: Geometrie definieren & auf die GPU laden.
- Schritt 4: Die Verknüpfung von GLSL Programmobjekt und Geometrie
- Schritt 5: Die Verknüpfung von GLSL Programmobjekt und Geometrie, mit Vertex Array Objects.
- Schritt 6: Kameramatrizen mit GLM

Zu jedem Schritt gibt es ein ausführbares Code Sample. Ab Schritt 4 wird sogar tatsächlich was gerendert ;-). Sehen Sie sich bitte jeweils die *main.cpp* an, sowie ab Schritt 2 auch Vertex- und Fragment Shader. Der C++ Code enthält noch einige Zusatzkommentare.

Die Tutorials gehen davon aus, dass Sie in der CG1 den Code, also die C# Version desselben, schon kennengelernt haben – und wenn Sie den Code damals vielleicht auch nicht zu 100% verstanden haben, so sollten Ihnen das Gezeigte zumindest vertraut sein.

Der gezeigte Code stellt das benötigte Minimum dar, um ein 3D Objekt mit modernem OpenGL zu rendern. Er stellt somit die Grundlage für die erste Hausübung dar. Im Tutorium zu unserer LVA wird Manuel Dobusch im Wesentlichen dasselbe mit Ihnen durchgehen, nur mit etwas komplexerer Geometrie, Sie können ihm zu den Tutorials ruhig Fragen stellen.

Der Code ist zum einfacheren Verständnis als *Codewurst* präsentiert, in anderen Worten: es gibt kaum Hilfsklassen. Bezüglich der Sprache wurde keine Rücksicht auf ihren C++ Wissensstand genommen (ich weiß zum Beispiel nicht, wann exakt Sie in der C++ Einführung das erste Mal den `std::vector` – das Äquivalent zur C# `List` – kennenlernen), in der Annahme, dass Sie die meisten Konstrukte intuitiv verstehen werden. Sollte es Unklarheiten zum C++ Teil geben, melden Sie sich ruhig.