

## 1 Qt5

- Fenstererzeugung und Maus-/Tastatursteuerung
- <http://doc.qt.io/>

## 2 OpenGL

- Punktbasierte Geometriebeschreibung. Aus Punkten werden geometrische Primitive zusammengesetzt: Punkte, Linien, Dreiecke.
- Vertex-Attribute (Position, Farbe, Normale, Texturkoordinaten) definieren das Erscheinungsbild der gezeichneten Primitive.
- OpenGL ist eine Zustandsmaschine
- Shader-Program (GLSL) beschreiben, wie aus Vertex-Attributen und **uniform**-Variablen Bilder berechnet werden
- **GLSL**
  - Vertex-Shader: Transformationen der Vertices: Rotation, Skalierung, Translation, Projektionen.
  - Fragment-Shader: Berechnung der Farb- und Tiefenwerte für jedes Pixel, auf das Primitive abgebildet werden.
  - Integrierte Datentypen und Operationen: Vektoren (**vec3**, **vec4** mit **dot**, **cross**, **reflect**), Matrizen (**mat3**, **mat4** mit **scale**, **rotate**, **transpose**).
  - OpenGL: Übergabe von Vertex-Array-Objekten an Shader. GLSL: Im Vertex-Shader als **in**-Werte.
  - Kommunikation zwischen Vertex- und Fragment-Shader über **in-out**-Variablen mit oder ohne Interpolation (**flat**, **smooth**, **noperspective**).
  - Konstante Werte (von Vertex-Stream unabhängig) werden als **uniform**-Werte im Shader definiert.
- Nur Core-Funktionalität wird unterstützt, kein Compatibility-Mode
- <https://www.opengl.org/sdk/docs/man/docbook4/>

## 3 GLM - OpenGL Mathematics

- Bringt GLSL-Matrix-/Vektor-Datentypen und -Operationen auf die CPU
- <http://glm.g-truc.net/>

## 4 CG

Einige Utility-Funktionen z.B. zum Laden von Shader-Codes werden von uns zur Verfügung gestellt und sind im Namespace **CG** zusammengefasst