

Übungen zu Computergraphik I

– Übungsblatt 4 –

Lehrstuhl für Computergraphik
und Multimediasysteme

Andreas Görlitz, Jan Mußmann

Abgabe: Bis spätestens **6. November 2018**, 10 Uhr
Besprechung: **Dienstag 13. November und Mittwoch 14. November 2018**
Gesamtpunktzahl nach Übungsblatt 4: 21 von 65

Hinweis: Die Programmieraufgaben müssen per E-Mail an **Jan Mußmann** eingereicht werden. Geben Sie dabei bitte immer Ihren **Namen**, Ihre **Matrikelnummer**, sowie Ihre **Übungsgruppe (Di. / Mi.)** an. Geben Sie nur die von Ihnen **geänderten Dateien** ab (`GLWidget.cpp`, `vs.glsl` und `fs.glsl`).

Aufgabe 1 OpenGL Theorie (2 Punkte)

- 1.1 Erklären Sie was ein **Vertex Array Object** (VAO) ist. Erklären Sie insbesondere den Zusammenhang bzw. Unterschied zwischen VAO und **Buffer** (genauer: **Vertex Buffer Object** (VBO)).
- 1.2 Ein **Shader-Programm** besteht aus einem **Vertex-Shader** und einem **Fragment-Shader**. Erklären Sie in welcher Reihenfolge diese Shader aufgerufen werden und auf welchen Daten (Punkte / Pixel / ...) diese Shader jeweils arbeiten.

Aufgabe 2 OpenGL Praxis (4 Punkte) Nehmen Sie als Ausgangsbasis das auf unserer Webseite bereitgestellte Programmgerüst `ueb04.zip`. Um das Projekt zu kompilieren, folgen Sie der Anleitung auf der CG1 Übungsseite.

- 2.1 Die Methode `initializeBuffers()` ist im aktuellen Programmgerüst unvollständig, bzw. wird lediglich das **Vertex Array Object** (VAO) erstellt (durch `glGenVertexArrays()`). Erweitern Sie die Methode `initializeBuffers()` so dass sie, wie in der Vorlesung (Folie 3.20) besprochen, die Positionsdaten aus `geometry` auf die Grafikkarte überträgt.
Hinweis: Wenn Sie alles korrekt gemacht haben, sollten Sie ein fliegendes Dreieck auf Ihrem Monitor sehen.
- 2.2 Der Parameter `color` im Vertex-Shader (Datei `vs.glsl`) wird aktuell nicht ausgewertet. Stattdessen wird im Fragment-Shader (Datei `fs.glsl`) die Farbe auf `(0, 0, 0, 1)` gesetzt.
Ändern Sie sowohl den Vertex-Shader als auch den Fragment-Shader so dass die Farbe auf `color` gesetzt wird (d.h. geben Sie den Wert von `color` vom Vertex-Shader weiter an den Fragment-Shader und setzen sie den Wert von `fcolor` entsprechend).
- 2.3 Definieren Sie das in der Zeichnung dargestellte sternenförmige Objekt mit der Längeneinheit d , die der Funktion `setupStar()` als Parameter übergeben wird (vgl. Abb. 1).

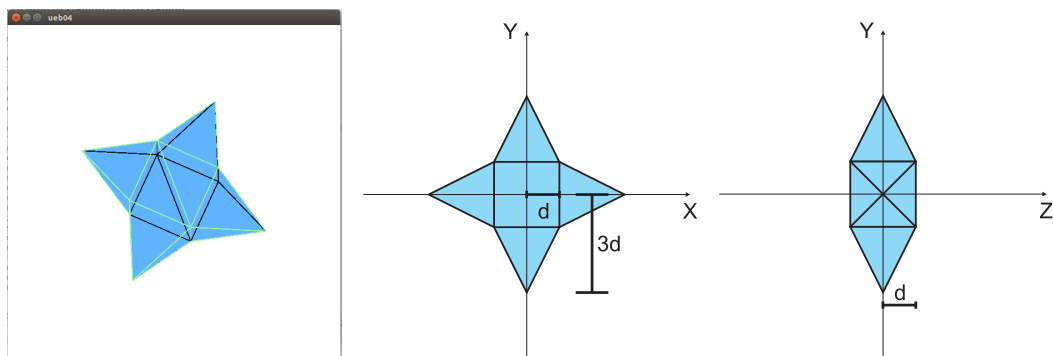


Abbildung 1: *Sternenförmiges Objekt*