

# Aufgabenblatt 1

## Ziel

Dieses Praktikum dient zur Einarbeitung in die einzelnen Komponenten von WebGL 1.0 (OpenGL ES 2.0 / GLSL ES 1.00). Sie erarbeiten sich Grundlagen der objektorientierten Programmierung mit JavaScript und der OpenGL Rendering Pipeline. Sie beschäftigen sich zudem mit den Grundprinzipien des Grafikstandards und erstellen erste 2D/3D Objekte mittels OpenGL Grafik-Primitiven und Shadern.

## Spielregeln

Das Praktikum gilt als bestanden, wenn alle Aufgaben dieses Arbeitsblattes erfolgreich bearbeitet und vorgestellt wurden. Bereiten Sie Ihre Abgabe gut vor. Ergänzen Sie jede Aufgabe mit handschriftlich Skizzen und Erklärungen, die sie am Ende präsentieren können.

## Literatur

- WebGL Dokumentation: <https://developer.mozilla.org/en-US/docs/Web/WebGL>
- WebGL Tutorial: <https://webglfundamentals.org/>
- Nischwitz, Fischer, Socher, Haberäcker: Computergrafik und Bildverarbeitung (Band I: Computergrafik).<sup>1</sup>
- glmatrix Dokumentation: <http://glmatrix.net>

## Benötigte Ausstattung

- Ein aktueller Internetbrowser mit Entwicklertools (vorzugsweise Firefox)
- Ein beliebiger Texteditor (z.B. Notepad++, Visual Studio Code, Vim)
- Download im Ilias: *HelloTriangle.zip* und *HelloKeyboard.zip*

---

<sup>1</sup>Link zum eBook über <https://www.fh-muenster.de/bibliothek/katalog/>

## Aufgabe P1 - Einstieg in die Shader Programmierung

**Hinweis:** Testen Sie zunächst, ob Ihr System den technischen Anforderungen des Praktikums entspricht und rufen Sie folgende Seite auf: <http://webglsamples.org/dynamic-cubemap/dynamic-cubemap.html>. Zumeist reicht ein Update der Grafiktreiber oder unter Unix/Linux die Aktivierung proprietärer Grafiktreiber. Es wird empfohlen die Entwicklertools des Firefox zu nutzen. Hier können verschiedene Werkzeuge über die Werkzeugeinstellungen ein- und ausgeschaltet werden.

Starten Sie das Projekt *HelloTriangle*.

Analysieren Sie den Source-Code. An welcher Stelle des Codes werden Fragment- und Vertex-Shader eingebunden? Wozu dienen Sie jeweils? Und auf welchen Daten arbeiten Sie?

Ändern Sie die Farbe des Dreiecks im Fragment-Shader, indem Sie die **Vertex-Koordinaten als Farbwerte** interpretieren. Wie lassen sich die Pixel-Farbwerte im Fragment-Shader aus den Vertex-Koordinaten im Vertex-Shader ermitteln? Wie behandeln Sie Koordinaten, die außerhalb des Farbbereiches liegen?

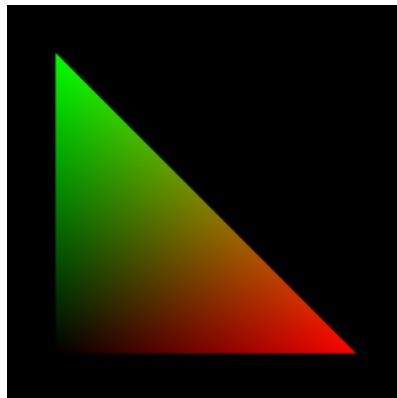


Abbildung 1: Zielbild.

## Aufgabe P2 - Grafische Grundobjekte

Zeichnen Sie zunächst ein Viereck (neue Klasse), welches sich aus zwei Dreiecken zusammensetzt.

Zeichnen Sie im zweiten Schritt einen aus Vierecken zusammengesetzten Quader (neue Klasse) mit gleich langen Seiten. Achten Sie darauf, dass die Frontseiten des Quaders nach außen zeigen.

Skizzieren Sie den Quader auf Papier. Fügen Sie ihrem Code entsprechende Kommentare hinzu, so dass sich Ihr Ergebnis auf der Skizze nachvollziehen lässt.

## Aufgabe P3 - Objekt Inspizieren

Fügen Sie in der Szene eine Tastatursteuerung ein (*HelloKeyboard.zip*). Rotieren Sie dabei die Welt, so dass das Objekt aus allen Positionen betrachtet werden kann.

Zur Berechnung der Rotationsmatrix nutzen Sie `glmMatrix` und die darin enthaltenen Methoden. Sorgen Sie dafür, dass die Matrix an den entsprechenden Shader weitergegeben wird, um Auswirkung auf die Szene zu haben.