Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Alina Hang, Dominikus Baur

Computergrafik 1
Sommersemester 2011
Übungsblatt 3

Übungsblatt 3: WebGL 101

Abgabe:
Dieses Übungsblatt ist einzeln oder in einer Gruppe zu lösen (wir empfehlen allerdings es allein zu bearbeiten). Die Lösung ist bis **Montag**, den **30. Mai 2011, 12:00 Uhr s.t.** über UniWorx (http://www.pst.ifi.lmu.de/uniworx) abzugeben.
Benennen Sie die Dateien nach dem Schema <Übungsblatt>-<Aufgabe>.<extension>. Packen Sie alle Dateien in eine ZIP-Datei und laden Sie diese bei UniWorx hoch.

Inhalt:
Ziel dieses Übungsblattes ist, sich einen ersten Eindruck von WebGL zu verschaffen und einige elementare Konzepte zu erarbeiten. Wundern Sie sich nicht dass dieses Übungsblatt eine Menge Tutorials enthält – ohne die Grundlagen lässt sich noch nicht viel Interessantes machen.

Hintergrund:
In diesem Abschnitt finden Sie in jedem Übungsblatt Hilfsmaterialien, Anleitungen und andere frei verfügbare Informationsquellen die Ihnen bei der Bearbeitung helfen können.
https://www.khronos.org/registry/webgl/specs/1.0/ (Offizielle Spezifikation der Khronos Gruppe)
http://learningwebgl.com/blog/?page_id=1217 (Mutter aller WebGL-Tutorialseiten – arbeiten Sie bitte für dieses Übungsblatt die Tutorials 0 und 1 durch)
http://www.peter-strohm.de/webgl/index.php (Sehr gut erklärtes deutsches Tutorial zu WebGL)
http://wiki.delphigl.com/index.php/Tutorial_WebGL (WebGL Tutorial für OpenGL Programmierer)
Da WebGL eine brandneue Technologie ist beachten Sie bitte bei jedem Tutorial das Sie finden das Veröffentlichungsdatum – eventuell sind Dinge schon überholt!

Aufgabe 1: Einstieg WebGL

Arbeiten Sie das Codegerüst durch, das für Aufgabe 1 auf der Homepage bereitgestellt wird. Es basiert auf Giles Thomas' WebGL Tutorial Nummer 1 (http://learningwebgl.com/blog/?p=28).

a) Lesen Sie Giles' Erklärungen zum Code um ihn zu verstehen. Die Erklärungen zu Shadern sind im Augenblick noch nicht wichtig, da das Thema gesondert in einem späteren Übungsblatt behandelt wird.
b) WebGL basiert auf dem stark optimierten OpenGL ES (der Mobilvariante von OpenGL) und enthält daher keinen *immediate mode*, bei dem Zeichenbefehle direkt hintereinander ausgeführt werden. Mit welchen Befehlen würden Sie das weiße Viereck in diesem immediate mode zeichnen (mehr Infos: http://wiki.delphigl.com/index.php/glBegin)? Wieso sind Vertex Buffer Objects schneller als der immediate mode?
c) Schreiben Sie den Code so um, dass statt des Vierecks ein Kreis gezeichnet wird (gefüllt oder nicht bleibt Ihnen überlassen).

Ludwig–Maximilians–Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Alina Hang, Dominikus Baur

Aufgabe 2: Transformationen

In dieser Aufgabe sollen Sie mit Kamerapositionen und –ausrichtungen experimentieren. Benutzen Sie dazu das Codegerüst das auf der Homepage zur Verfügung gestellt wird (Anmerkung: Es ist völlig in Ordnung für Matrix-/Vektoroperationen eine Bibliothek wie glMatrix zu verwenden. Manuelles Berechnen der Matrizen ist aber natürlich auch eine tolle Übung ;).

a) Die Kameraposition soll anpassbar sein: Fügen Sie dazu unterhalb des Canvas drei Textfelder für die Koordinaten hinzu. Sobald der Benutzer diese ändert soll die Kamera in der Szene entsprechend verschoben werden, dabei aber weiterhin auf den Würfel im Ursprung gerichtet sein.
   Tipp: Nutzen Sie die Funktion *lookAt* in der glMatrix Bibliothek!
b) Erstellen Sie ein zweites Set von Textfeldern für eine zweite Kameraposition. Fügen Sie einen Schieberegler zwischen diesen und den Textfeldern aus Aufgabenteil a) hinzu. Wenn der Benutzer den Schieberegler zieht soll sich die Kamera auf einen Punkt zwischen den durch die Textfelder bestimmten Endpunkten bewegen (z.B.: Schieberegler ist genau in der Mitte: Kamera liegt auf Punkt direkt in der Mitte zwischen den Endpositionen). Achten Sie wiederum auf die Ausrichtung der Kamera auf den Ursprung.
c) Fügen Sie zu guter Letzt noch eine Checkbox ein, die die Projektion von perspektivisch auf orthographisch wechselt.
   Tipp: glMatrix enthält eine Methode um eine orthographische Projektionsmatrix zu berechnen!

Aufgabe 3 (*): Spaß mit Primitiven

Erstellen Sie basierend auf Ihrer Lösung für Aufgabe 2 (oder dem Codegerüst dafür) eine möglichst interessante 3D-Szene. Diese darf mit beliebig vielen Primitiven gefüllt sein. Sie können auch mit Animationen der Kamera arbeiten um einen 3D-Film zu erzeugen. Seien Sie kreativ ☺

Hinweis: Diese Aufgabe ist eine Stern (*) - Aufgabe, d.h. wir präsentieren die beste Lösung auf der Vorlesungshomepage!