

Übungsblatt 8: Meshes

Abgabe:

Dieses Übungsblatt ist einzeln oder in einer Gruppe zu lösen (wir empfehlen allerdings es allein zu bearbeiten). Die Lösung ist bis **Montag, den 11. Juli 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:

Bisherige Übungsblätter verwendeten als Objekte meist simple Kisten oder 2D-Objekte. Dieses Übungsblatt behandelt den Umgang mit Polygon-Meshes, also komplexeren Objekten die eventuell auch aus externen Objekt-Dateien geladen werden.

Hintergrund:

In diesem Abschnitt finden Sie in jedem Übungsblatt Hilfsmaterialien, Anleitungen und andere frei verfügbare Informationsquellen die Ihnen bei der Bearbeitung helfen können.

https://developer.mozilla.org/en/WebGL/Lighting_in_WebGL

<http://learningwebgl.com/blog/?p=1523> (Tutorial 13)

http://wiki.delphigl.com/index.php/Tutorial_WebGL (WebGL Tutorial für OpenGL Programmierer)

Hinweis:

Mit dem Update auf Firefox 5 wurde die Anzeige von Cross-Domain Texturen gesperrt. Möglichkeiten, wie Sie die Aufgaben dennoch bearbeiten können, finden Sie unter:

http://www.die-informatiker.net/post/Computergrafik_SS11/Texturen_in_Firefox_5/105392#105392

Aufgabe 1: Per-Fragment Lighting

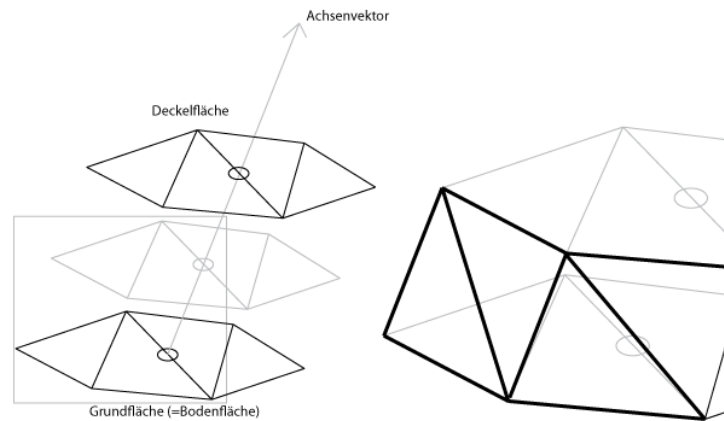
Die Beleuchtung im letzten Blatt wurde auf einer Per-Vertex-Basis durchgeführt, was die Ergebnisse gerade bei simplen Modellen nicht sonderlich interessant macht. In dieser Aufgabe soll die Beleuchtung auf Fragment-Basis durchgeführt werden.

- Lesen Sie das WebGL Tutorial 13: <http://learningwebgl.com/blog/?p=1523>
- Nutzen Sie den Beispielcode von Aufgabe 7-2 (oder Ihre eigene Lösung) und ändern Sie das Beleuchtungsmodell auf Per-Fragment statt Per-Vertex-Lighting.

Aufgabe 2: Objektgenerierung

Komplexe Objekte können nicht nur manuell oder in einem 3D-Editor erstellt werden, sondern lassen sich auch algorithmisch erzeugen. In dieser Aufgabe sollen Sie ein Extrude-Objekt herstellen: Die Idee dabei ist einen Grundkörper zu nehmen und diesen entlang einer Achse zu führen um einen dreidimensionalen Körper zu kreieren.

- a) Nutzen Sie das Codegerüst für diese Aufgabe und erweitern Sie die Methode *createExtrudeObject*. Diese Methode nimmt vier Parameter: Ein Array von 2D-Punkten für die Grundform, einen 3D-Vektor als Richtung der Achse, eine Länge und ein *wgl_obj* Objekt. Im Codegerüst ist ein Sechseck als Grundform vorgegeben und wird noch nicht extrudiert zurückgegeben. Achten Sie außerdem darauf, dass Ihr Code mit beliebigen Grundformen und Achsen funktioniert.
- b) Gehen Sie wie folgt vor:
- Berechnen Sie den Mittelpunkt der Grundform wo die Achse ansetzt.
 - Füllen Sie den Vertexbuffer mit den Punkten der Grundform (Bodenfläche).
 - Rücken Sie Stück für Stück entlang der Achse vor und fügen Sie Dreiecke ein um die Außenhaut des Objekts zu erzeugen (s. rechte Seite der Skizze).
 - Sobald Sie die gewünschte Länge erreicht haben fügen Sie wieder die Punkte der Grundform (natürlich an die richtige Position verschoben) in den Vertexbuffer ein um das Objekt abzuschließen (Deckelfläche).
 - Erzeugen Sie abschließend einen passenden Indexbuffer.



Aufgabe 3: Laden von JSON-3D-Objekten

3D-Editoren wie 3DS Max, Blender oder Maya erlauben das Erstellen beliebig komplexer Objekte für 3D-Anwendungen. Um mit diesen Modellen in WebGL zu arbeiten müssen sie zuerst importiert werden.

- Lesen Sie Tutorial 14 (<http://learningwebgl.com/blog/?p=1658>). Konzentrieren Sie sich dabei auf den JSON-Import Teil.
- Nehmen Sie das Codegerüst für diese Aufgabe von der Homepage und importieren Sie die Teekanne.