

# WP Einführung in die Computergrafik

SS 2014, Hochschule für Angewandte Wissenschaften (HAW), Hamburg  
Prof. Dr. Philipp Jenke, Lutz Behnke



## Aufgabenblatt 5 - Texturen

In diesem Aufgabenblatt erweitern Sie Ihre bestehenden Datenstrukturen und den Szenengraphen um die Möglichkeit, Texturen bei der Darstellung zu verwenden.



Darstellung von texturierten Oberflächen-Dreiecksnetzen (Quadrat, Kugel, Würfel).

### Änderungshistorie

23.04.2014      Zusätzliche Informationen unter b) zur Material-Datei.

#### a) Erweiterung der Datenstrukturen

Für die Verwendung von Texturen müssen Sie Texturkoordinaten repräsentieren. Die Texturkoordinaten sollen zusammen mit den Dreiecken abgelegt werden. Im `TriangleMesh` soll dazu eine Liste von Texturkoordinaten vorhanden sein (genau wie die Liste der Vertices). Für ein Dreieck wird pro Ecke ein Index in diese Liste gespeichert.

#### b) Einlesen von Texturkoordinaten

Wenn Sie ein Dreiecksnetz manuell aufbauen, dann können Sie natürlich auch die Texturkoordinaten von Hand eingeben. Es soll aber auch die Möglichkeit geben, die Texturkoordinaten aus einer Wavefront OBJ-Datei einzulesen. Erweitern Sie dazu Ihren Importer. Texturkoordinaten haben den Op-Code `vt`. Zunächst müssen Sie alle Einträge für Texturkoordinaten in einer Liste zwischenspeichern. Zusammen mit den Dreiecken (Op-Code `f`) werden dann auch die Indizes des Texturkoordinaten eingelesen (siehe Aufgabenblatt 3) und dann in der `TriangleMesh`-Datenstruktur gespeichert. Das OBJ-Format bietet auch die Möglichkeit, eine Textur als Material abzuspeichern. In der Materialdatei (Endung `.mtl`) wird eine Textur mit `map_Kd <Name-des-Texturbildes>` abgelegt. Den Namen der zugehörigen Materialdatei finden Sie in der OBJ-Datei unter dem Op-Code `mtllib`. In der OBJ-Datei finden Sie außerdem eine Zeile `usemtl <Name-des-Materials>`. Damit wird angegeben, dass das Material für alle folgenden Facetten in der Datei gilt. Theoretisch könnten in einer OBJ-Datei mehrere Materialien verwaltet werden. Wir beschränken und hier aber auf den Fall, dass nur ein Material in einer OBJ-Datei vorkommt. Erweitern Sie Ihren Importer, um diese Information zu lesen und im `TriangleMesh` abzuspeichern.

#### c) Erzeugen einer OpenGL-Textur

Texturen sind im Prinzip einfach Bilder. Sie müssen OpenGL aber natürlich bekannt gemacht werden. Manche OpenGL-Treiber verlangen, dass Texturen als Länge und Breite immer Potenzen von 2 ( $2^n$  Pixel) verwenden. Setzen Sie, um sicher zu gehen, nur Texturen ein, die diese Bedingung erfüllen. Jogl bietet für die Repräsentation einer Textur die Klasse `Texture`. Eine Instanz der Klasse erzeugen Sie mit `TextureIO.newTexture()` aus einer Bilddatei.

#### d) Verwenden der Textur

Zum Verwenden einer Textur müssen Sie zunächst die Texturkoordinaten an OpenGL übertragen. Das machen Sie genauso wie das Setzen der anderen Eigenschaften der Vertices (Normalen, Farben) mit einem entsprechenden OpenGL-Aufruf unmittelbar vor dem `glVertex`-Aufruf. Texturkoordinaten übergeben Sie mit `glTexCoord2f`.

Dann müssen Sie festlegen, dass Sie überhaupt Texturen verwenden wollen. Dies geschieht mit `gl.glEnable(GL.GL_TEXTURE_2D);`

Damit OpenGL weiß, welche Textur verwendet werden soll, müssen Sie diese aktivieren. Dies geschieht mit `glBindTexture`.

### **e) Weitere Textureinstellungen**

Es gibt eine ganze Reihe weiterer Einstellungsmöglichkeiten für die Konfiguration. Recherchieren Sie, was die folgenden bedeuten und welche Werte Sie setzen können:

- `glTexParameteri( GL.GL_TEXTURE_2D, GL.GL_TEXTURE_WRAP_S, ...`
- `glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, ...`
- `glTexEnvf( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, ...`

### **Hinweise**

Für das Verwenden von Texturen in OpenGL gibt es viele Tutorials im Internet. Um die notwendigen Befehle nachzuschlagen, bietet sich immer das "Red Book" an [1]. Dieses Buch finden Sie auch als Online-Version unter <http://www.glprogramming.com/red/>.

### **Referenzen**

- [1] Dave Shreiner, Graham Sellers, John Kessenich, and Bill Licea-Kane: The OpenGL® Programming Guide, 8th Edition