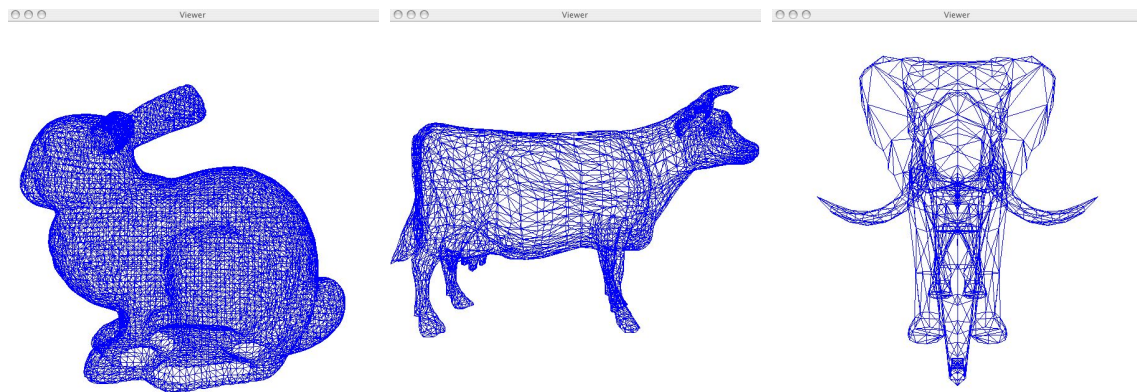


Übungsblatt 7

Aufgabe 1. Erweitern Sie Ihr OpenGL-Programm aus Aufgabe 4 von Übungsblatt 6 so, dass es ein Dreiecks-Netz, welches als Indexed Faceset gespeichert ist, einlesen und darstellen kann. Das Dreiecks-Netz liegt im Objekt-Fileformat **obj** (siehe Rückseite) vor. Eine nähere Beschreibung dieses Formats finden Sie auf der Rückseite. Unter dem Link <http://www.mi.hs-rm.de/~schwan/Vorlesungen/GenCG/> finden Sie vier Modelle (Stanford-Bunny, Cow, Elephant und Squirrel) im **obj**-Fileformat. Analog zur Aufgabe 4 von Übungsblatt 6 soll Ihr Programm beim Aufruf von `python oglViewer.py object.obj` die Daten aus der Datei `object.obj` einlesen und das zugehörige Modell in einem GLUT-Fenster darstellen.

Geben Sie Ihre Lösung bis spätestens zum 02.06.2013 um 23.59 Uhr über das *read.MI* ab. *Spätere Abgaben können nicht mehr berücksichtigt werden.*



Ihr **obj**-Viewer soll (mindestens) die folgenden Features besitzen:

- (2 Punkte) *Einlesen eines obj-Files und Darstellung dessen Inhalts als Polygonnetz mit Hilfe von VBOs.*
- (3 Punkte) *Navigation mit Hilfe der Maus*
 - Bei gedrückter *linker Maustaste* soll das Modell rotiert werden können.
 - Bei gedrückter *mittlerer Maustaste* soll an das Modell heran- bzw. weggezoomt werden können.
 - Bei gedrückter *rechter Maustaste* soll das Modell verschoben werden können.
- (3 Punkte) *Umschalten zwischen Orthogonal- und Zentral- Projektion.* In beiden Fällen soll die Kamera geeignet angepasst werden, wenn die Fenstergröße geändert wird.
- (2 Punkte) *Wechseln der Hintergrund- und Objekt-Farbe.* Als Hintergrund- und Objekt-Farbe sollen, unabhängig voneinander, mindestens die Farben *Schwarz, Weiß, Rot, Grün, Blau* und *Gelb* eingestellt werden können.

Das Objekt-Fileformat **obj** ist ein einfaches ASCII Fileformat, um die Geometrie und andere Eigenschaften (Farben, Textur, ...) von Objekten zu beschreiben. Das **obj**-Fileformat unterstützt sowohl polygonale Objekte (definiert durch Punkte, Linien und Polygone) als auch Freiform-Objekte (definiert durch Kurven und Flächen). Unter <http://paulbourke.net/dataformats/> finden Sie eine komplette Beschreibung des Formats. In einem **obj**-File werden

- *Objekt-Punkte (vertices)* durch ein vorangestelltes **v**
- *Texturkoordinaten* durch ein vorangestelltes **vt**
- *Vertex-Normalen* durch ein vorangestelltes **vn**

gekennzeichnet.

Die einzelnen Polygone (*faces*) werden durch ein vorangestelltes **f**, gefolgt von den Punktindices (für **v**, **vt** und **vn** jeweils durch **/** getrennt) gekennzeichnet. Ein Dreieck wird also in der Form

```
f v/vt/vn v/vt/vn v/vt/vn
```

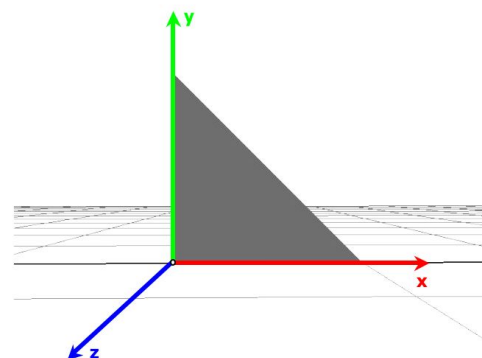
dargestellt. Sind keine Texturkoordinaten vorhanden, werden zwei aufeinanderfolgende Slashes geschrieben (**//**). Sind weder Texturkoordinaten noch Normalenvektoren vorhanden, werden die Punktindices alleine, ganz ohne Slashes geschrieben.

Die folgende *Beispiel-Datei* definiert ein Dreieck. Hierzu werden zunächst die drei Eckpunkte (0.0, 0.0, 0.0), (1.0, 0.0, 0.0), (0.0, 1.0, 0.0) definiert. Anschließend folgt die Definition der Normalenvektoren (0.0, 0.0, 1.0), (0.0, 0.0, 1.0), (0.0, 0.0, 1.0). Die letzte mit **f** beginnende Zeile definiert schließlich ein Dreieck mit den Eckpunkten (1, 2, 3), wobei Punkt 1 die Normale 1, Punkt 2 die Normale 2 und Punkt 3 die Normale 3 zugewiesen bekommt. Texturkoordinaten sind keine vorhanden.

```
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0

vn 0.0 0.0 1.0
vn 0.0 0.0 1.0
vn 0.0 0.0 1.0

f 1//1 2//2 3//3
```



Eine **obj** Datei kann mehrere Vertex-, Vertexnormalen- und Face-Blöcke enthalten. Die **Nummerierung der Vertices und Vertexnormalen ist fortlaufend** und beginnt nicht etwa für jeden Block wieder bei 1.