Übungsblatt 4

Willkommen zur vierten Übung zur Vorlesung *Generative Computergrafik*. Dieses Blatt behandelt Dreiecksnetze und Shading.

Aufgabe 1. 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 Punktindizes (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, ein Viereck in der Form

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

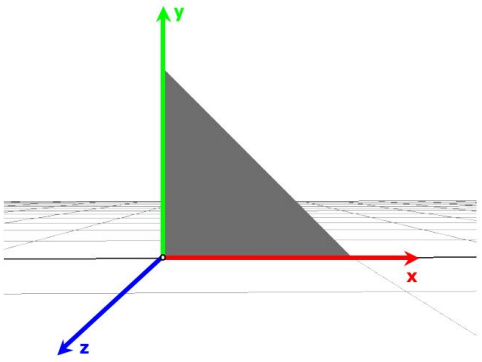
Sind keine Texturkoordinaten vorhanden, werden zwei aufeinanderfolgende Slashes geschrieben (//). Sind weder Texturkoordinaten noch Normalenvektoren vorhanden, werden die Punktindizes alleine, ganz ohne Slashes geschrieben.

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.

Schreiben Sie ein Python Programm, dass eine `obj`-Datei einliest und die folgenden Informationen ausgibt:

1. Datei enthält Texturkoordinaten (Ja/Nein)
2. Datei enthält Normalen (Ja/Nein)
3. Maximale Anzahl Ecken pro Polygon (3, 4, ...)
4. Anzahl Vertices
5. Anzahl Faces
6. Anzahl Edges

Weiterhin soll Ihr Programm *Vertex-Normalen* berechnen, falls die **obj**-Datei keine solchen enthält und anschließend als `ply`-Datei exportieren, bei der alle Flächen Dreiecke sind. Nachfolgend finden Sie ein einfaches Beispiel für eine `ply`-Datei. Eine vollständige Beschreibung des Fileformats finden Sie unter `http://paulbourke.net/dataformats/ply/`.

```
ply
format ascii 1.0
comment This file represents a triangle (© U. Schwanecke 2018)
element vertex 3
property float32 x
property float32 y
property float32 z
property float32 nx
property float32 ny
property float32 nz
element face 1
property list uint8 int32 vertex_indices
end_header
0 0 0 0 0 1
2 0 0 0 0 1
1 1 0 0 0 1
3 0 1 2
```

Mit Hilfe des Open Source Programms *Meshlab* (`https://www.meshlab.net`) können Sie `obj`, `ply` und viele andere Dateiformate für geometrische Modell einlesen und darstellen.