

Wavefront OBJ Format

Johannes Diemke

Übung im Modul OpenGL mit Java
Wintersemester 2010/2011

Motivation

- 3D-Modelle bestehen i. d. R. aus Tausenden von Dreiecken
 - ▶ Händische Definition im Quellcode nahezu unmöglich
 - ▶ Spezielle 3D Modellierungssoftware notwendig
- Im Bereich der Computerspiele ist 3ds Max beliebt
 - ▶ Blizzard Entertainment
 - ▶ Crytek
 - ▶ Electronic Arts
 - ▶ Remedy Entertainment
 - ▶ ...
- Blender stellt eine Open Source Alternative dar

Wavefront OBJ Dateien

- Wavefront's Advanced Visualizer Format
- Definieren Geometrie und dessen Eigenschaften
 - ▶ Polygonale Objekte
 - ▶ Freiform-Objekte (Kurven und Flächen)
 - ▶ Material inkl. Texturen
- Animation wird nicht unterstützt
- Existiert als ASCII Format (.obj) und binäres Format (.mod)
- Von nahezu allen Programmen unterstützt
- Mittlerweile aber obsolet → COLLADA
- Vollständige Spezifikation:
<http://local.wasp.uwa.edu.au/~pbourke/dataformats/obj/>

Das ASCII Dateiformat (.obj)

- Einfache ASCII Textdatei
 - ▶ Menschenlesbar
 - ▶ Theoretisch im Texteditor editierbar
- Enthält Anweisungen zur Definition von
 - ▶ Vertices
 - ▶ Texturkoordinaten
 - ▶ Normalen
 - ▶ Faces
 - ▶ Material-Gruppen
 - ▶ Polygon-Gruppen
 - ▶ Benannte Objekte
 - ▶ Smoothing Groups
 - ▶ Freiform-Objekte

#-Anweisung

- Zeilen die mit einem Rautenzeichen beginnen sind Kommentare

```
# a comment line
```

v-Anweisung

- Definiert einen Vertex
- Als Argumente folgen die x -, y - und z -Komponente
- Werden fortlaufend, beginnend bei eins, durchnummeriert

```
v x y z
```

vt-Anweisung

- Definiert eine Texturkoordinate
- Als Argumente folgen die u -, v - und optionale w -Komponente
- Werden fortlaufend, beginnend bei eins, durchnummeriert

```
vt u v [w]
```

vn-Anweisung

- Definiert eine Normale
- Als Argumente folgen die x -, y - und z -Komponente
- Werden fortlaufend, beginnend bei eins, durchnummeriert

vn x y z

f-Anweisung

- Definiert Polygon über eine Liste von n -Tupeln aus Indizes auf
 - ▶ Vertices
 - ▶ Texturkoordinaten (optional)
 - ▶ Normalen (optional)
- Vertices werden standardmäßig CCW angegeben
- Definition durch Angabe einer Liste von Vertices

f v1 v2 v3 v4 ...

f-Anweisung (Forts.)

- Definition durch Angabe einer Liste von 2-Tupeln bestehend aus einem Vertex und einer Texturkoordinate

f v1/vt1 v2/vt2 v3/vt3 ...

- Definition durch Angabe einer Liste von 3-Tupeln bestehend aus einem Vertex, einer Texturkoordinate und einer Normale

f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3 ...

- Texturkoordinaten können auch weggelassen werden

f v1//vn1 v2//vn2 v3//vn3 ...

g-Anweisung

- Dient der Gruppierung von Polygonen
- Alle auf eine Gruppen-Definition folgenden Polygone sind Teil der Gruppe

```
g group_name1 [group_name2] ...
```

mtllib-Anweisung

- Gibt die zu verwendenden externen MTL-Dateien (.mtl) an
- Material Libraries definieren durch usemtl nutzbare Materialien

```
mtllib filename1 [filename2] ...
```

usemtl-Anweisung

- Bestimmt das für nachfolgende Polygone zu nutzende Material
- Material wird in externer MTL-Datei (.mtl) definiert

```
usemtl name
```

s-Anweisung

- Definiert die zu verwendene Smoothing-Gruppe
- Dient der Gruppierung von Polygonen zur interpolierten Normalenberechnung
- Wird nur bei Abwesenheit von Vertex-Normalen berücksichtigt
- Ist das Argument `off` oder `0` wird keine Smoothing-Gruppe verwendet

```
s number|off|0
```

Wavefront MTL Dateien

- Material Library
- Einfache ASCII Textdatei (.mtl)
- Definiert aus OBJ-Dateien referenzierbare Materialien
 - ▶ Materialfarben
 - ▶ Transparenz
 - ▶ Texturen
- Berücksichtigt keine neueren Entwicklungen
- Aktuelle Software ergänzt das ursprüngliche Format
 - ▶ Specular Maps
 - ▶ Parallax Maps
- Vollständige Spezifikation:
<http://local.wasp.uwa.edu.au/~pbourke/dataformats/mtl/>

Wavefront OBJ Format

newmtl-Anweisung

- Beginnt eine neue Materialdefinition
- Über den Namen kann das Material später referenziert werden

```
newmtl material_name
```

Ka-Anweisung

- Definiert ambiente Materialfarbe

```
Ka r g b
```

Kd-Anweisung

- Definiert diffuse Materialfarbe

```
Kd r g b
```

Ks-Anweisung

- Definiert spiegelnde Materialfarbe

```
Ks r g b
```

Ns-Anweisung

- Definiert spiegelnden Exponenten s

```
Ns s
```

d-Anweisung und Tr-Anweisung

- Definieren beide die Transparenz

```
d transparency
```

map-Anweisung und bump-Anweisung

- Definiert zu verwendene Texturen

```
map_Ka lenna.tga           # the ambient texture map
map_Kd lenna.tga           # the diffuse texture map
map_Ks lenna.tga           # the specular texture map
map_d lenna_alpha.tga      # the alpha texture map
map_bump lenna_bump.tga    # the bump map
bump lenna_bump.tga        # some implementations use "bump"
```

Wavefront OBJ Format

Beispiel: exd9xl_a.obj

```
# Wavefront OBJ file
# Converted by the DEEP Exploration Deep Exploration 5.5 5.5.3.2642 Release
# Right Hemisphere, LTD
# http://www.righthemisphere.com/
mtllib exd9xl_a.mtl
# object cockpitr
g cockpitr
v -0.74229 1.30720 1.01731
v -0.75097 1.28629 1.01731
v -0.74229 1.30720 1.05808
...
# 1087 verticies
vt 0.05055 0.91097
vt 0.05055 0.95106
vt 0.05453 0.91097
...
```

Beispiel: exd9xl_a.obj (Forts.)

...

```
# 427 texture verticies
```

```
vn 0.96364 0.00865 -0.26707
```

```
vn 0.99695 -0.00076 0.07800
```

```
vn 0.68719 -0.67580 -0.26659
```

...

```
# 1027 normals
```

```
usemtl Material_4
```

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

```
f 4/4/4 2/3/3 3/2/2
```

```
f 2/3/3 4/4/4 5/5/5
```

...

Wavefront OBJ Format

Beispiel: exd9xl_a.mtl

```
# Wavefront material file
# Converted by the DEEP Exploration  Deep Exploration 5.5 5.5.3.2642 Release
# Right Hemisphere, LTD
# http://www.righthemisphere.com/
newmtl Material_4
Ka 1 1 1
Kd 1 1 1
Ks 0.498039 0.498039 0.498039
illum 2
Ns 119.428
map_Kd exdx.png
map_bump
bump
map_opacity
map_d
refl
map_kS
map_kA
map_Ns
```

Wavefront OBJ Format

Beispiel: exd9xl_a.obj



Implementierung eines OBJ Loaders

- Häufig wird nur eine Teilmenge des OBJ Standards implementiert
 - ▶ I. d. R. nur Polygonale Objekte
 - ▶ Material & Texturen
 - ▶ Keine Freiform-Objekte
- Zur Darstellung entsprechende Datenstrukturen aufbauen
 - ▶ Werden Gruppen, Smoothing-Groups oder Materialien benötigt?
 - ▶ Naive Implementierung
 - Für jedes Dreieck Material & Textur setzen
 - Viele State Changes → Performance-Problem
 - ▶ Gruppierung nach verwendetem Material
 - Alle Dreiecke mit gleichem Material in Material-Gruppe
 - Material-Gruppen mit Vertex-Arrays oder VBOs rendern
 - Minimiert die Anzahl der State Changes → Performance-Gewinn
 - ▶ Materialien nutzen teilweise gleiche Texturen → Wiederverwendung

- Dave Shreiner
OpenGL Programming Guide
<http://www.opengl-redbook.com/>
- Richard S. Wright, Benjamin Lipchak und Nicholas Haemel
OpenGL SuperBibel
<http://www.starstonesoftware.com/OpenGL/>
- Randi J. Rost
OpenGL Shading Language
<http://www.3dshaders.com/>
- Tomas Akenine-Möller, Eric Haines und Naty Hoffman
Real-Time Rendering
<http://www.realtimerendering.com/>

- Edward Angel
Interactive Computer Graphics
<http://www.cs.unm.edu/~angel/>
- Gerald Farin und Dianne Hansford
Practical Linear Algebra
<http://www.farinhansford.com/books/pla/>
- Fletcher Dunn und Ian Parberry
3D Math Primer for Graphics and Game Development
www.gamemath.com/