

WP Einführung in die Computergrafik

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



Aufgabenblatt 1 - Szenengraph

Aufgabe 1: Datenstruktur

Eine sehr wichtige Datenstruktur in der Computergrafik ist der Szenengraph (siehe <http://de.wikipedia.org/wiki/Szenengraph>). Ihre Aufgabe in diesem Aufgabenblatt ist es, einen Szenengraph zu implementieren.

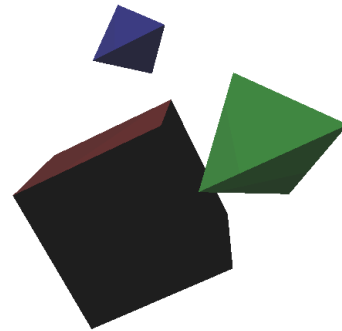


Abbildung 1: Würfel und zwei Tetraeder im Szenengraph.

a) Generischer Knoten

Ein Szenengraph ist aus Knoten aufgebaut. Jeder Knoten kann (muss aber nicht) mehrere Kindknoten haben. Es gibt immer einen obersten Knoten in der Datenstruktur, den Wurzelknoten. Schreiben Sie eine Klasse Node zur Repräsentierung eines Knotens. Die Klasse benötigt Methoden zum Verwalten der Kindknoten. Außerdem soll Node eine Methode

```
public void draw(GL2 gl)
```

bereitstellen. In dieser Methode können OpenGL-Objekte angelegt und dann dargestellt werden. Achten Sie darauf, dass beim Aufruf der Methode `draw()` eines Knotens auch die `draw()`-Methoden der Kindknoten (rekursiv) aufgerufen werden müssen. Die `draw()`-Methode des Wurzelknotens wird in der Methode `drawGLContent()` der vorgegebenen Klasse `Aufgabe1Frame` aufgerufen.

b) Knoten mit Geometrie

Entwerfen Sie mindestens zwei spezialisierte Knoten zur Darstellung von primitiven Körpern. Solche Knoten könnten beispielsweise einen Würfel oder einen Tetraeder repräsentieren. Die Körper sollten im Ursprung zentriert sein und etwa die Ausdehnung 1 haben. Welche konkreten Körper Sie verwenden, ist Ihnen freigestellt. Jeder Körper muss aber aus mehreren Dreiecken bestehen.

c) Transformationsknoten

Bisher liegen alle Körper im Ursprung. Mit Hilfe der Datenstruktur Szenengraph ist es aber elegant möglich, eine Szene aus vielen Körpern aufzubauen, die jeweils an unterschiedlichen Positionen liegen. Dazu verwendet man Transformationsknoten. Transformationsknoten repräsentieren eine Transformation, die dann für alle Kindknoten (und deren Kinder, rekursiv) gilt. Setzen Sie die folgenden Transformationen um:

- Skalierung um einen Faktor
- Translation (Verschiebung) um einen Vektor
- [optional] Rotation um eine beliebige Achse mit einem beliebigen Winkel

OpenGL bringt Funktionalität mit, um die Transformationen umzusetzen, ohne die eigentliche Geometrie zu verändern:

- Skalierung: `gl.glScalef(a,b,c)` // Skalierung in x-Richtung: a, y-Richtung: b, z-Richtung: c
- Translation: `gl.glTranslatef(x,y,z)`

verwendet man eine der OpenGL-Transformationen, dann wird diese gesetzt und nicht automatisch zurückgenommen. Im Szenengraphen soll die Transformation aber nur beim Zeichnen der Kindknoten verwendet werden. Dies kann erzielt werden, indem man sich die alten Transformationen merkt, dann die neuen Transformationen setzt, dann die Kindknoten zeichnet und am Ende die alten Transformationen wieder herstellt:

- Merken der alten Transformationen: `gl.glPushMatrix();`
- Setzen der neuen Transformation: z.B. `gl.glScalef(0.5f, 0.3f, 0.2f);`
- Zeichnen der Kindknoten

- Zurücksetzen der Transformationen: `gl.glPopMatrix();`

d) Verschiedene Farben (Materialien)

Implementieren Sie einen weiteren Typ von Knoten zum Setzen von Oberflächenmaterialien (und damit der Farben). Die notwendigen Code-Fragmente finden Sie in den Vorlesungsfolien. Die sichtbare Farbe setzen Sie über die diffuse Komponente.