

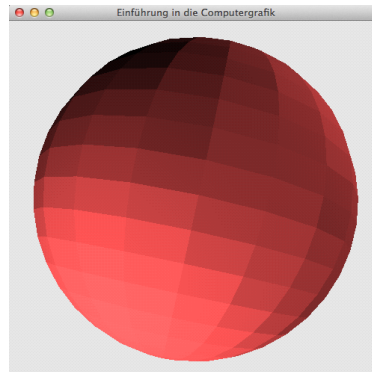
WP Einführung in die Computergrafik

WS 2013/2014, Hochschule für Angewandte Wissenschaften (HAW), Hamburg
Prof. Dr. Philipp Jenke, Lutz Behnke



Aufgabenblatt 2 - Netze

In diesem Aufgabenblatt entwickeln Sie eine Datenstruktur zur Repräsentierung und Darstellung von Dreiecksnetzen.



Darstellung eines Oberflächen-Dreiecksnetzes.

ITriangleMesh	
•	<code>addTriangle(Triangle):void</code>
•	<code>addVertex(Point3d):int</code>
•	<code>getNumberOfTriangles():int</code>
•	<code>getNumberOfVertices():int</code>
•	<code>getTriangle(int):Triangle</code>
•	<code>getVertex(int):Point3d</code>
•	<code>clear():void</code>

Interface *ITriangleMesh*.

Aufgabe 1: Datenstruktur

Vorgegeben ist das Interface *ITriangleMesh*. Schreiben Sie eine Klasse *TriangleMesh*, die das Interface implementiert. In der Klasse müssen Knoten (Vertices) und Dreiecke (Triangles) verwaltet werden. Ein Vertex hat 3 Koordinaten. Ein Dreieck besteht aus drei Indizes und einer Oberflächennormale. Für die Repräsentation der Dreiecke wird im Interface die Klasse *Triangle* verwendet, die Sie auch in Ihrer Klasse *TriangleMesh* verwenden sollen.

Aufgabe 2: Dreiecksnormalen

Bei der Beleuchtungsrechnung spielt die Oberflächennormale eine zentrale Rolle. Zunächst gehen wir davon aus, dass jedes Dreieck eine Normale hat. Diese muss senkrecht auf der Dreiecksfläche stehen und die Länge 1 haben. Wichtig ist außerdem die Orientierung der Normalen - sie muss "nach außen" zeigen. Erweitern Sie die Klasse *Triangle* um eine Methode

```
public void computeNormal(Point3d pA, Point3d pB, Point3d pC);
```

die die Normale des Dreiecks aus den drei Eckpunkten (übergeben als Parameter) berechnet und abspeichert.

Tip: Kreuzprodukt.

Aufgabe 3: Darstellung

Um Dreiecksnetze mit Java3D darstellen zu können, müssen Sie als ein geeigneter Knoten in den Szenengraph eingehängt werden. Schreiben Sie eine Klasse *MeshShapeFactory* mit einer statischen Methode

```
public static Shape3D createMeshShape(TriangleMesh mesh);
```

die für ein *TriangleMesh*-Objekt einen Szenengraph-Knoten vom Typ *Shape3D* erzeugt. Gehen Sie folgendermaßen vor:

- Anlegen eines Objekts vom Typ *TriangleArray*
 - ein *TriangleArray* ist eine Liste von Punkten, je drei aufeinander folgende Punkte beschreiben ein Dreieck
 - für jeden Punkt kann man die *Coordinates* und die *Normal* setzen - vorausgesetzt man hat das Array zu Beginn entsprechend initialisiert
 - wir verwenden für alle Knoten, die zu einem Dreieck gehören jeweils die gleiche Normale - die Dreiecksnormale
- Verpacken des *TriangleArrays* in ein *Shape3D*-Objekt
- Setzen einer Farbe/*Appearance*

Hinweise: Zum Testen Ihrer Funktionalität können Sie die Klasse *TriangleMeshFactory* verwenden. Damit können Sie Beispielnetze erzeugen.