

WP Einführung in die Computergrafik

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



Aufgabenblatt 2 - Dreiecksnetze

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

a) Datenstruktur

Schreiben Sie eine Klasse `TriangleMesh`, zur Repräsentation von Dreiecksnetzen. Ein Dreiecksnetz besteht aus einer Liste von Vertices und einer Liste von Dreiecken. Vertices und Dreiecke sollten Sie jeweils durch eine eigene Klasse repräsentieren, weil zum aktuellen Zeitpunkt noch nicht feststeht, welche Informationen mit jedem Vertex oder Dreieck abgelegt werden müssen. In diesem Aufgabenblatt hat ein Vertex zunächst eine 3D-Position und eine Normale. Ein Dreieck hat die Indizes (Ganzzahlwerte ≥ 0) der drei Eckpunkte in der Vertex-Liste und ebenfalls eine Normale. Mehrere Dreiecke können sich also den gleichen Vertex teilen. `TriangleMesh` verwaltet dann je eine Liste von Vertices und eine Liste von Dreiecken.

b) Normalen

Erweitern Sie die Klassen `TriangleMesh` um die Möglichkeit, Normalen für die Dreiecke und für die Vertices zu berechnen. Berechnen Sie immer zuerst die Dreiecksnormalen (aus den Eckpunkten) und dann die Vertexnormalen (gemittelt aus den anliegenden Dreiecken). Speichern Sie die berechneten Normalen mit in der Datenstruktur ab. Bei der Berechnung der Normalen können Sie vereinfachend davon ausgehen, dass es sich bei den repräsentierten Dreiecksnetzen um geschlossene Netze handelt. Jedes Dreieck hat also genau drei Nachbardreiecke. Dies müssen Sie dann bei selbst erzeugten `TriangleMesh`-Objekten natürlich selber sicherstellen.

c) Darstellung

Entwickeln Sie jetzt einen zusätzlichen Szenengraphen-Knoten (siehe vorheriges Aufgabenblatt) zur Darstellung von Dreiecksnetzen. Damit sollen sich Dreiecksnetze mit OpenGL darstellen lassen. Achten Sie bei der Darstellung darauf, dass Sie die Dreiecksnormalen (nicht die Vertexnormalen) verwenden. Wenn Sie also die drei Eckpunkte eines Dreiecks an OpenGL übergeben, dann verwenden Sie für alle drei die gleiche (Dreiecks-) Normale.

[Optional] Verwenden Sie Displaylisten oder (noch besser) Vertex-Arrays für die Übergabe der Vertices und Dreiecke zu Darstellung an OpenGL (Informationen z.B. hier für Displaylisten: http://www.songho.ca/opengl/gl_displaylist.html und hier für Vertex-Arrays: http://www.songho.ca/opengl/gl_vertexarray.html)

Testen Sie die Funktionalität mit mindestens zwei Test-Dreiecksnetzen, bestehend aus mehreren Dreiecken. Sie können z.B. die Körper aus dem vorherigen Aufgabenblatt (Würfel, Tetraeder) oder weitere platonische Körper verwenden.



Quelle:
http://de.wikipedia.org/wiki/Platonischer_K%C3%B6rper