



Aufgabenblatt 3: Halbkanten-Datenstruktur

In diesem Aufgabenblatt entwickeln Sie eine Halbkanten-Datenstruktur zur Repräsentierung und Darstellung von Dreiecksnetzen. Es müssen Tests vorhanden sein, die sicherstellen, dass die Halbkanten-Datenstruktur korrekt aufgebaut wurde.

Aufgabe 3.1: Datenstruktur

Schwerpunkte: Umsetzen der Halbkanten-Datenstruktur

Aufgabe: Vorgegeben ist weiter das Interface `ITriangleMesh`. Schreiben Sie eine Klasse `HalfEdgeTriangleMesh`, die wieder das Interface implementiert. Setzen Sie die Datenstruktur mit dem Halbkanten-Ansatz um. Dazu finden Sie im Package `computergraphics.datastructures.halfedge` bereits die notwendigen Klassen `HalfEdge`, `HalfEdgeVertex` und `HalfEdgeTriangle`. In der Klasse `HalfEdgeTriangleMesh` müssen Halbkanten, Facetten und Vertices jeweils als Listen abgelegt werden. Die Halbkantenstruktur wird durch die im Interface vorgegebenen Methoden aufgebaut. `addVertex()` fügt einen neuen Vertex hinzu. `addTriangle()` fügt ein Dreieck zwischen drei Vertices hinzu. Dazu sind die Indices der drei Vertices als Parameter übergeben. Sie müssen damit folgendes machen:

- Erzeugen einer neuen Facette und drei Halbkanten.
- Setzen aller Eigenschaften der neuen Objekte (außer der Referenzen auf gegenüberliegende Halbkanten, das geschieht im folgenden Schritt).

Nachdem alle Vertices und alle Facetten eingefügt wurden, müssen noch zwei weitere Informationen gesetzt werden:

- Jedem Vertex muss eine ausgehende Halbkante zugewiesen werden.
- Zu jeder Halbkante muss die gegenüberliegende Halbkante gesetzt werden. Finden Sie dazu für jede Halbkante vom Vertex `v1` zum Vertex `v2` die passende gegenüberliegende Halbkante von `v2` nach `v1`. Den Zielvertex einer Halbkante erhalten Sie durch `<halfedge>.getNext().getStartVertex()`.

Setzen Sie dies in einer Hilfsmethode um. Diese Methode können Sie aufrufen, nachdem Vertices und Facetten eingefügt wurden.

Aufgabe 3.2: Darstellung

Schwerpunkte: Darstellung der Halbkanten-Datenstruktur

Aufgabe: Erstellen Sie einen weiteren Szenengraph-Knoten, der ein Dreiecksnetz in Halbkanten-Repräsentation darstellen kann.

Aufgabe 3.3: Normalen

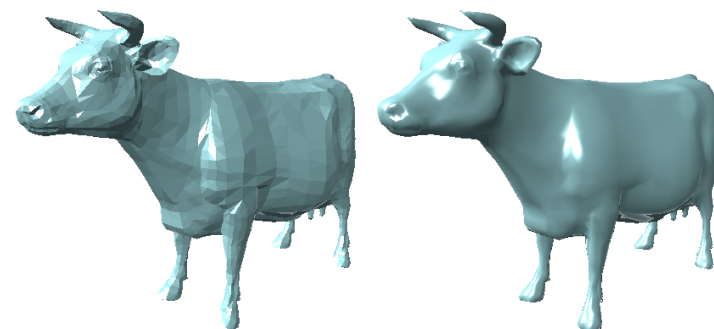


Abbildung 1: Oberflächendarstellung mit Dreiecksnormalen (links) und Vertexnormalen

Schwerpunkte: Normalen berechnen

Aufgabe: Berechnen Sie wie im vorherigen Aufgabenblatt die Dreiecksnormalen. Berechnen Sie daraus - unter Verwendung der Halbkanten-Datenstruktur - Vertexnormalen (also in konstanter

Berechnungskomplexität pro Vertex). Über den Szenengraph-Knoten muss es möglich sein, Dreiecksnetze entweder mit Beleuchtung per Dreiecksnormalen oder mit Beleuchtung per Vertexnormalen darzustellen.

Aufgabe 3.4: Rand

Schwerpunkte: Rand bestimmen

Aufgabe: Finden Sie anhand der Halbkanten-Darstellung heraus, welche Halbkanten den Rand eines Dreiecksnetzes darstellen. Zeichnen Sie diese explizit bei der Darstellung (z.B. als Linien). In der Datei assets/meshes/hemisphere.obj finden Sie ein Beispiel-Mesh mit Rand.