

Wintersemester 2011/12

## Übungen zu Computergraphik I - Blatt 4

Abgabe am 30. 11. 2011

Auf der Webseite zur Vorlesung liegt ein kleines Rahmenprogramm für die folgende Aufgabe zum Download bereit. Die unten erwähnten Funktionen finden sie in der Klasse `GLWidget`.

### Aufgabe 1 (OpenGL/Qt, 3+4+5 Punkte)

a) Implementieren Sie die Methode `drawTetrahedron()`. Der Methode werden vier Parameter vom Typ `Vector3` übergeben, die die Eckpunkte des Tetraeders bestimmen.

Berechnen Sie auch die Flächennormalen und geben Sie diese mit `glNormal3f()` an.

Hinweise:

- Die Normalen müssen vor den Vertices angegeben werden.
- Die Normalen zeigen immer vom Objekt weg, also nach außen.
- Die Normale müssen auf die Länge 1 normalisiert sein.

b) In dieser Teilaufgabe sollen Sie ein etwas komplexeres Objekt (der Einfachheit halber *Tetraflake* genannt) konstruieren. Als Basis nehmen Sie den Tetraeder. Auf jede Seitenfläche des Tetraeders wird ein weiterer Tetraeder (Kindtetraeder) aufgesetzt, dabei sollen die Grundflächen des Vater- und Kindtetraeders mittig zentriert sein. Diesen Vorgang kann man rekursiv fortsetzen. Abbildung 1 illustriert einen Rekursionsschritt, und Abbildung 2 zeigt ein konkretes Beispiel. Die folgenden Instanzvariablen sind zu beachten:

- `m_flakeRecursionDepth` bestimmt die Rekursionstiefe .
- `m_flakeTetraChildSize`  $\in [0, 1]$  bestimmt die Größe der Grundfläche des Kindtetraeders (diese liegt somit zwischen 0 und der Grundfläche des Vatern tetraeders).  
Tip: Interpolieren Sie linear zwischen dem Dreiecksmittelpunkt  $P_m$  und den Eckpunkten  $P_{\{0,1,2\}}$ .  
Abbildung 2 zeigt ein Beispiel mit dem Interpolationswert  $s \approx 0.5$ .
- `m_flakeTetraHeight`  $\in [0, 1]$  bestimmt die Höhe  $h$  (damit liegt die Höhe zwischen 0 und  $\|(P_1 - P_0) \times (P_2 - P_0)\|$ ).

Implementieren Sie die oben erklärte Funktionalität in der Funktion `drawTetraFlake()`.

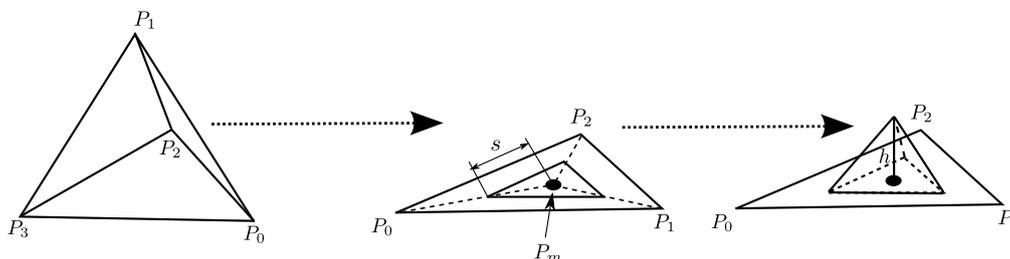


Abbildung 1:



Abbildung 2: Beispiel wie das Ergebnis aus Teil b) aussehen soll

- c) Im Framework steht eine Szene zur Verfügung, die eine Ebene und einige Tetraflakes enthält. Modifizieren Sie die Methode `drawTetraFlake()` so, dass ein Vertex grün gezeichnet wird, wenn er auf einer Seite der Ebene liegt und rot, wenn er auf der anderen Seite liegt. Die Farbwerte werden über `glColor3f()` gesetzt. Die Ebene wird durch ein Punkt in der Ebene, `m_plane_point`, und zwei orthogonale Vektoren `m_plane_vectors` definiert.