Prof. G. Zachmann
D. Mohr

Universität Bremen
Fachbereich 3
Arbeitsgruppe CGVR
29. Oktober 2013

Wintersemester 2013/14

Übungen zu Computergraphik - Blatt 3

Abgabe am 05. 11. 2013

Aufgabe 1 (Mathematische Grundlagen, *4 Punkte*)

Gegeben sei eine Ebene E der Form $\mathbf{n}\mathbf{x} + c = 0$ und ein Punkt P. Der Einfachheit halber können Sie annehmen, dass die Normale $\mathbf{n}$ normiert ist. Bestimmen Sie den Punkt Q, der durch Spiegelung von P an E hervorgeht.

Aufgabe 2 (Geometrische Prädikate, *4 Punkte*)

Beschreiben Sie ein geometrisches Prädikat, um zu testen, ob ein Punkt $P \in \mathbb{R}^3$ innerhalb eines gegebenen Dreieckes A, B, C im $\mathbb{R}^3$ liegt. Verwenden Sie keine Projektion oder Ähnliches, sondern ausschließlich das Skalar- und das Kreuzprodukt.

Tip: eine (von mehreren) Möglichkeiten ist, den Umlaufsinn der Teildreiecke, die mit 2 Eckpunkten des Ursprungsdreiecks und dem Punkt P gebildet werden, zu überprüfen.

Aufgabe 3 (Rasterisierung, *3 Punkte*)

Gegeben sei eine Gerade die durch die Punkte $P_1 = (2; 5)$ und $P_2 = (8; 2)$ verläuft. Berechnen Sie die Rasterung der Geraden mittels des Midpoint-Algorithmus. Geben Sie alle Zwischenschritte Ihrer Rechnung mit an. Zeichnen Sie das Ergebnis in das in Abb. 1 angegebene Raster.

Aufgabe 4 (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 2 illustriert einen Rekursionsschritt, und Abbildung 3 zeigt ein konkretes Beispiel. Die folgenden Instanzvariablen sind zu beachten:

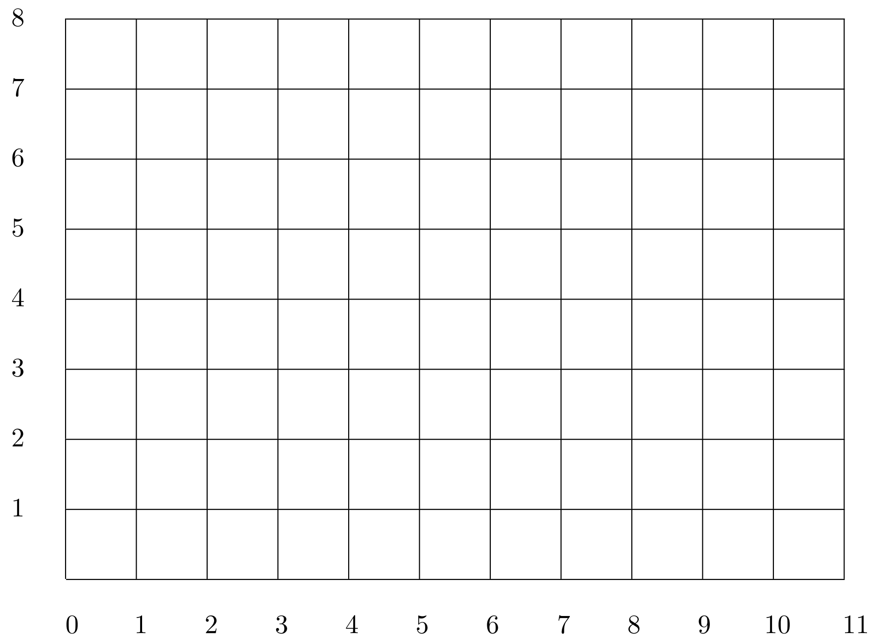


Abbildung 1: Pixelraster

- `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 Vatertetraeders).
  Tip: Interpolieren Sie linear zwischen dem Dreiecksmittelpunkt P_m und den Eckpunkten $P_{\{0,1,2\}}$. Abbildung 3 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()`.

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 liegen und rot, wenn er auf der anderen Seite liegt. Schneidet der Tetraflake die Ebene, so wird er gelb eingefärbt.

Die Farbwerte werden über `glColor3f()` gesetzt. Die Ebene wird durch einen Punkt in der Ebene, `m_plane_point`, und zwei orthogonale Vektoren `m_plane_vectors` definiert.

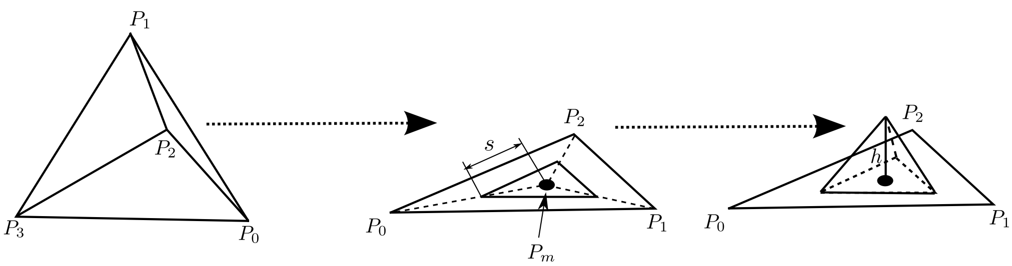
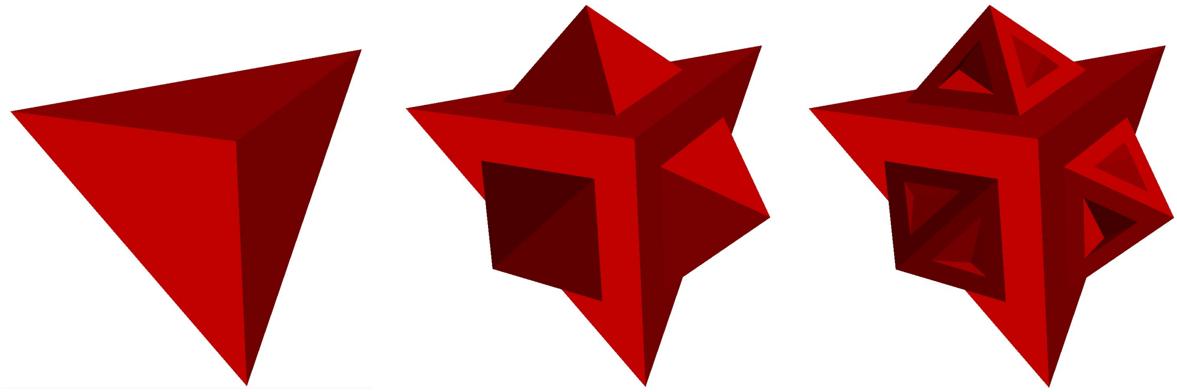


Abbildung 2:



Rekursionstiefe 0 Rekursionstiefe 1 Rekursionstiefe 2

Abbildung 3: Beispiel wie das Ergebnis aus Teil b) aussehen soll