

Wintersemester 2011/12

Übungen zu Computergraphik I - Blatt 7

Abgabe am 21. 12. 2011

Aufgabe 1 (Depth-Complexity und Overdraw, 2+2 Punkte)

- Erklären Sie den Unterschied zwischen den Begriffen *Depth-Complexity* und *Overdraw*
- Geben Sie einen Algorithmus zum Rendern einer beliebigen Szene aus Dreiecken an, der einen Overdraw von 1 für alle Pixel erreichen.

Aufgabe 2 (Z-Buffer: Wireframe-Darstellung mit Hidden Line Removal, 4 Punkte)

In OpenGL kann man sehr einfach Polygone entweder *solid* oder *wireframe* rendern. (Der Unterschied liegt allein im Zustand, in den OpenGL versetzt wird; im ersten Fall sagt man

```
glBegin(GL_POLYGON); glVertex(...); ...glEnd();
```

im zweiten Fall sagt man

```
glBegin(GL_LINE_LOOP); glVertex(...); ...glEnd(); )
```

Geben Sie ein Verfahren an, mit dem man mit Hilfe des Z-Buffers eine Szene in *wireframe* mit verdeckten Linien (*hidden line removal*) rendern kann. (Pseudo-Code genügt.) Es sollte in der Lage sein, eine Szene so zu rendern wie auf der rechten Seite in Folie 2 des Kapitels "Visibility Computations".

Tip: Ihr Algorithmus wird wahrscheinlich ein 2-Pass-Rendering-Verfahren sein.

Aufgabe 3 (Rendering mit BSP-Trees, 2 Punkte)

Angenommen, für eine gegebene Szene mit n Polygonen ist ein BSP schon erstellt worden. Wir nehmen außerdem an, dass bei diesem Aufbau nur wenige Splits vorgekommen sind, so dass der BSP $O(n)$ viele Fragmente enthält. Wir gehen weiterhin davon aus, dass das Rendering eines Polygons/Fragments die Zeit $O(1)$ benötigt.

Welche Laufzeit hat das *back-to-front* Rendering der kompletten Szene mittels BSP? Bitte begründen Sie Ihre Antwort.

Aufgabe 4 (Clipping, 3 Punkte)

Gegeben ist ein konvexes Polygon mit n Ecken, welches als Clip-Windows dient. Daran soll ein beliebiges Dreieck geclippt werden. Wie viele Eckpunkte hat das resultierende geclippte Polygon minimal/maximal? Begründen Sie Ihre Antwort ausführlich.

Aufgabe 5 (Line-Clipping, 2 Punkte)

Wo wird im Cyrus-Beck-Algorithmus die Konvexität benötigt? Skizzieren Sie ein konkretes Beispiel, bei dem der Algorithmus versagt.

Aufgabe 6 (Transformation eines Vierecks, 3 Punkte)

Gegeben sei ein konvexes Viereck A, B, C, D (nicht notwendigerweise ein Parallelogramm!) im 2D. Dieses soll durch eine Kette von Transformationen in das Einheitsquadrat transformiert werden.

- a) Überlegen Sie sich eine Kette von Transformationen, die ein Rechteck P, Q, R, S in das Einheitsquadrat überführt.
- b) Geben Sie eine Kette von Transformationen an, die ein Parallelogramm A', B', C', D' in das Einheitsquadrat überführt.