

Sommersemester 2011

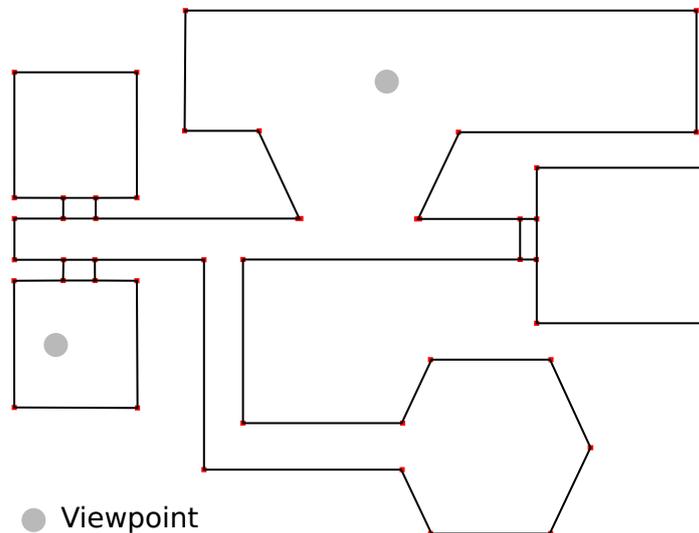
Übungen zu Geometrische Datenstrukturen für die Computergraphik - Blatt 4

Abgabe am 29. 06. 2011

ACHTUNG: Abgabe in der Vorlesung am Mittwoch den 29. 06. 2011!

Aufgabe 1 (Erzeugung eines BSP-Baumes, 2+3 Punkte)

Gegeben ist die folgenden Szene (2D).



- Erstellen Sie einen BSP-Tree (entweder Autopartition, oder allg. BSP).
- BSPs können verwendet werden, um eine Szene mit korrekter Verdeckung zu rendern, ohne einen z-Buffer zu verwenden. Wählen Sie zwei Standorte für einen Betrachter (Viewpoints) und geben Sie an, in welcher Reihenfolge der Baum durchlaufen bzw. die Strecken gezeichnet werden.

Aufgabe 2 (BSP-Tree, 4+2 Punkte)

- Wie hoch kann ein BSP-Tree, der aus 7 disjunkten (d.h. nicht schneidenden oder berührenden) nicht collinearen Linien-Segmente aufgebaut wird, minimal/maximal sein. Geben Sie je ein Beispiel an. Zeichnen Sie für jeden Fall die Szene und den BSP-Baum auf.
- Welche minimale/maximale Höhe hat ein BSP-Baum mit n disjunkten nicht collinearen Linien-Segmente?

Aufgabe 3 (BSP-Tree, 4 Punkte)

Gegeben ist ein BSP-Baum T und eine Menge von Objekten. Des Weiteren sei der Viewpoint p_{view} gegeben, welcher oberhalb der Splitting-Ebene des root-Knotens von T liegt. Beweisen Sie, dass der PaintersAlgorithm korrekt ist. Zeigen Sie hierfür, dass wenn (ein Teil von) Objekt A vor (Teilen von) Objekt B scan-konvertiert (can-convert) wird, so kann Objekt A nicht vor Objekt B liegen.

```
PaintersAlgorithm( $T$ ,  $p_{view}$ )
  Let  $v$  be the root of  $T$ 
  if  $v$  is a leaf
    Scan-convert the object fragments in  $S(v)$ 
  else if  $p_{view} \in h_v^+$ 
    PaintersAlgorithm( $T^-$ ,  $p_{view}$ )
    Scan-convert the object fragments in  $S(v)$ 
    PaintersAlgorithm( $T^+$ ,  $p_{view}$ )
  else if  $p_{view} \in h_v^-$ 
    PaintersAlgorithm( $T^+$ ,  $p_{view}$ )
    Scan-convert the object fragments in  $S(v)$ 
    PaintersAlgorithm( $T^-$ ,  $p_{view}$ )
  else ( $*p_{view} \in h_v*$ )
    PaintersAlgorithm( $T^+$ ,  $p_{view}$ )
    PaintersAlgorithm( $T^-$ ,  $p_{view}$ )
```

PaintersAlgorithm