Übung 7 - Gruppe 142

Visual Computing - Grafikpipeline & Eingabemodalitäten & VR+AR

TECHNISCHE UNIVERSITÄT DARMSTADT

Aufgabe 1: Allgemeine Fragen

TECHNISCHE UNIVERSITÄT DARMSTADT

a) In der Augmented Reality soll die Wahrnehmung des Nutzers verbessert und vergrößert werden. So werden hier zusätzliche Informationen bereitgestellt, wie die Geschwindigkeit in einem Head-up Display oder anstehende Termine am Rand des Sichtfeldes.
   In der Virtual Reality ist die virtuelle Welt das Ziel des Nutzers. Er möchte damit interagieren, und wenn möglich sollte er nicht bemerken sich in einer virtuellen Welt zu befinden.

b) In der Computer Vision geht es darum, gegenstände in Bilder zu erkennen, in der Computergrafik möchte man gegenstände erzeugen. Im Endeffekt kann vision als umgekehrte Grafik betrachtet werden.

c) 3D Objekte werden am Computer durch Polygon Netze mit Material, Beleuchtung in verschiedenen Szenen dargestellt.

Aufgabe 1: Allgemeine Fragen

TECHNISCHE UNIVERSITÄT DARMSTADT

d)
- **Flat Shading**: Aufteilung in Primitive. Die orientierung normale des Primitives ergibt dann die Helligkeit.
- **Gouraud Shading**: Aufteilung in Primitive. Helligkeit wird über die Eckpunkte Normale berechnet und diese Werte entlang der Kanten des Primitives linear interpoliert.
- **Phong Shading**: Wie Gouraud Shading, nur wird die Helligkeit auch auf den Flächen der Primitive interpoliert.

e) Hüllkörper umfassen Primitive, um Schnitt- oder Kollisionstests mit anderen Primitiven zu vereinfachen. Um dies zu gewährleisten, sollten Hüllkörper so einfach wie möglich aufgebaut sein, also Kugeln oder Bounding Boxes. Im 3-dimensionalen werden dann Kugeln oder Würfel verwendet.

Aufgabe 2: Geometrieverarbeitung

TECHNISCHE UNIVERSITÄT DARMSTADT

a) Zuerst wird das Sichtvolumen mithilfe der Model-Transformation an das Koordinatensystem angepasst. Danach werden die Texturen mit dem Painter's-Algorithmus gezeichnet. Dieser zeichnet die am weitesten entfernten Objekte zuerst. Da in unserem Beispiel alle Objekte nur einen z-Wert haben, kann ein Objekt nach dem anderen gezeichnet werden. -> Himmel -> Gras -> Sonne -> Haus -> Auto

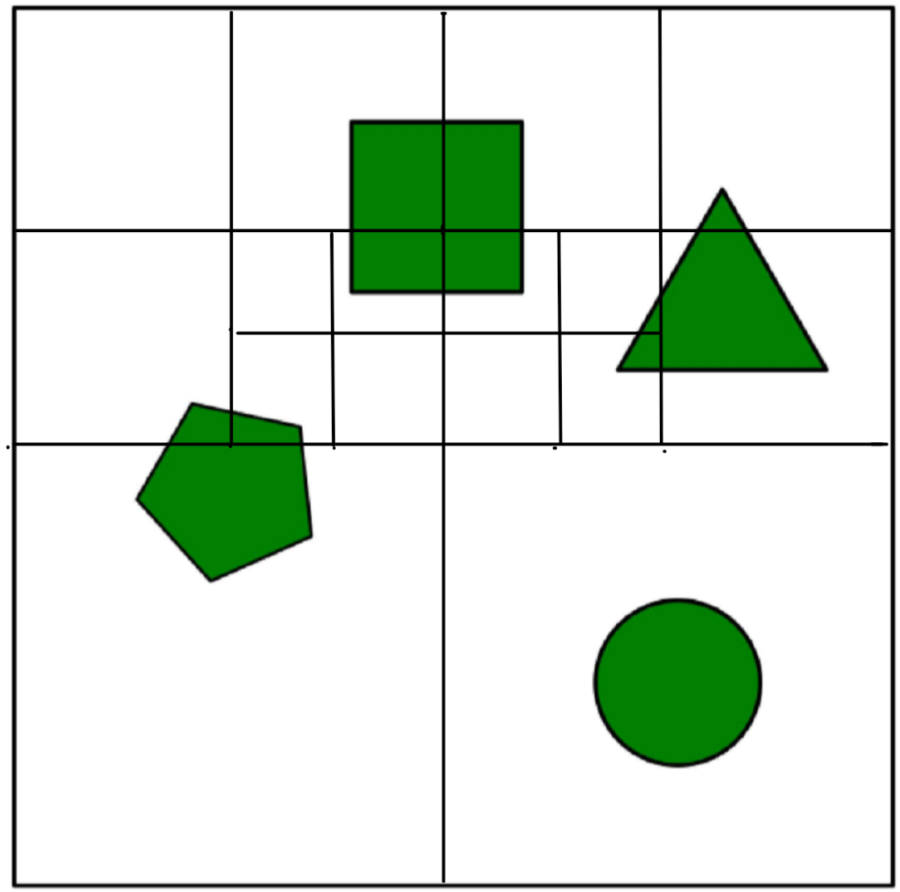
Aufgabe 2: Geometrieverarbeitung

TECHNISCHE UNIVERSITÄT DARMSTADT

b)

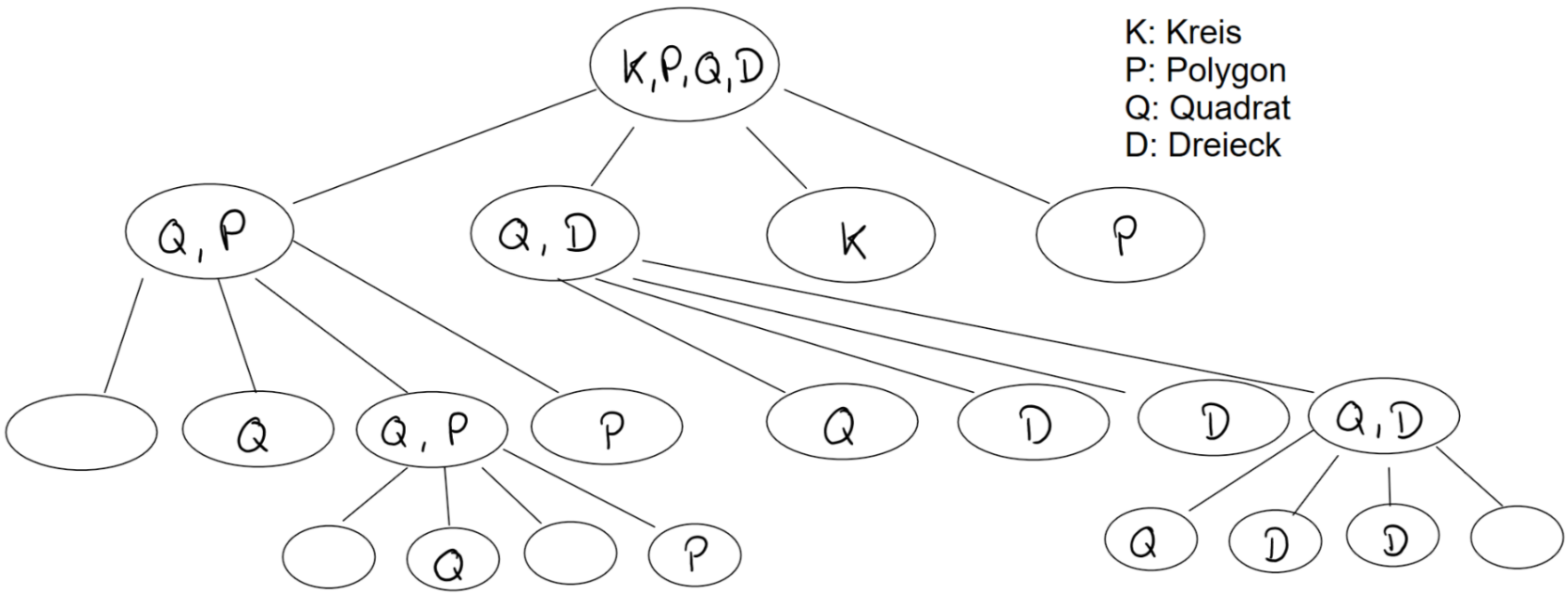
Aufgabe 3: Bäume
Quadtree

TECHNISCHE UNIVERSITÄT DARMSTADT



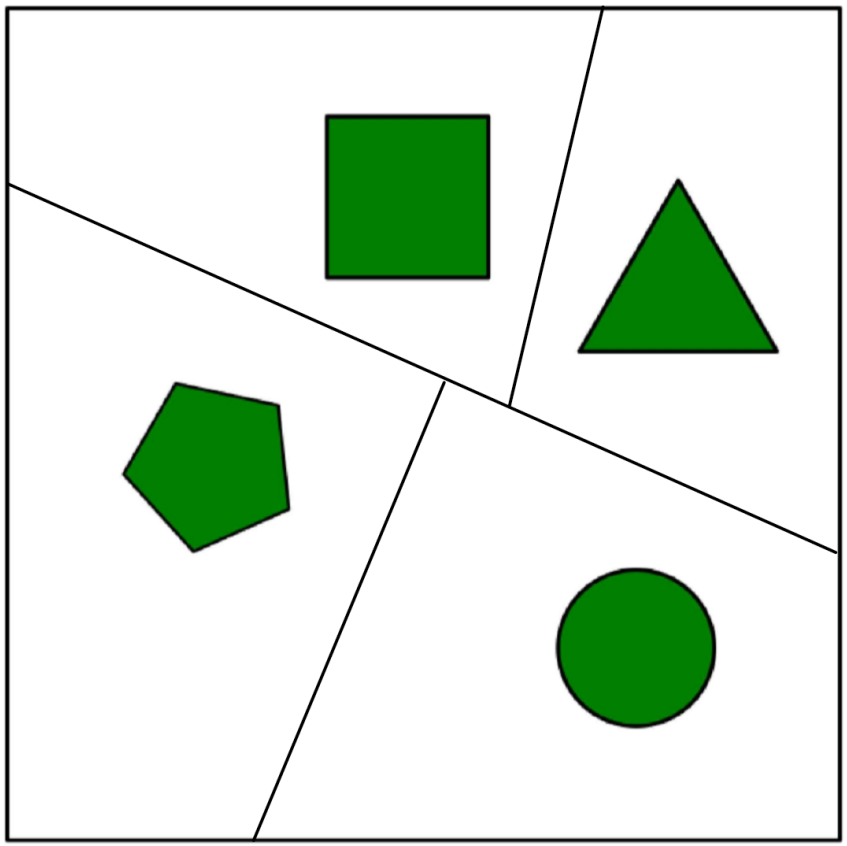
Aufgabe 3: Bäume
Quadtree

TECHNISCHE UNIVERSITÄT DARMSTADT



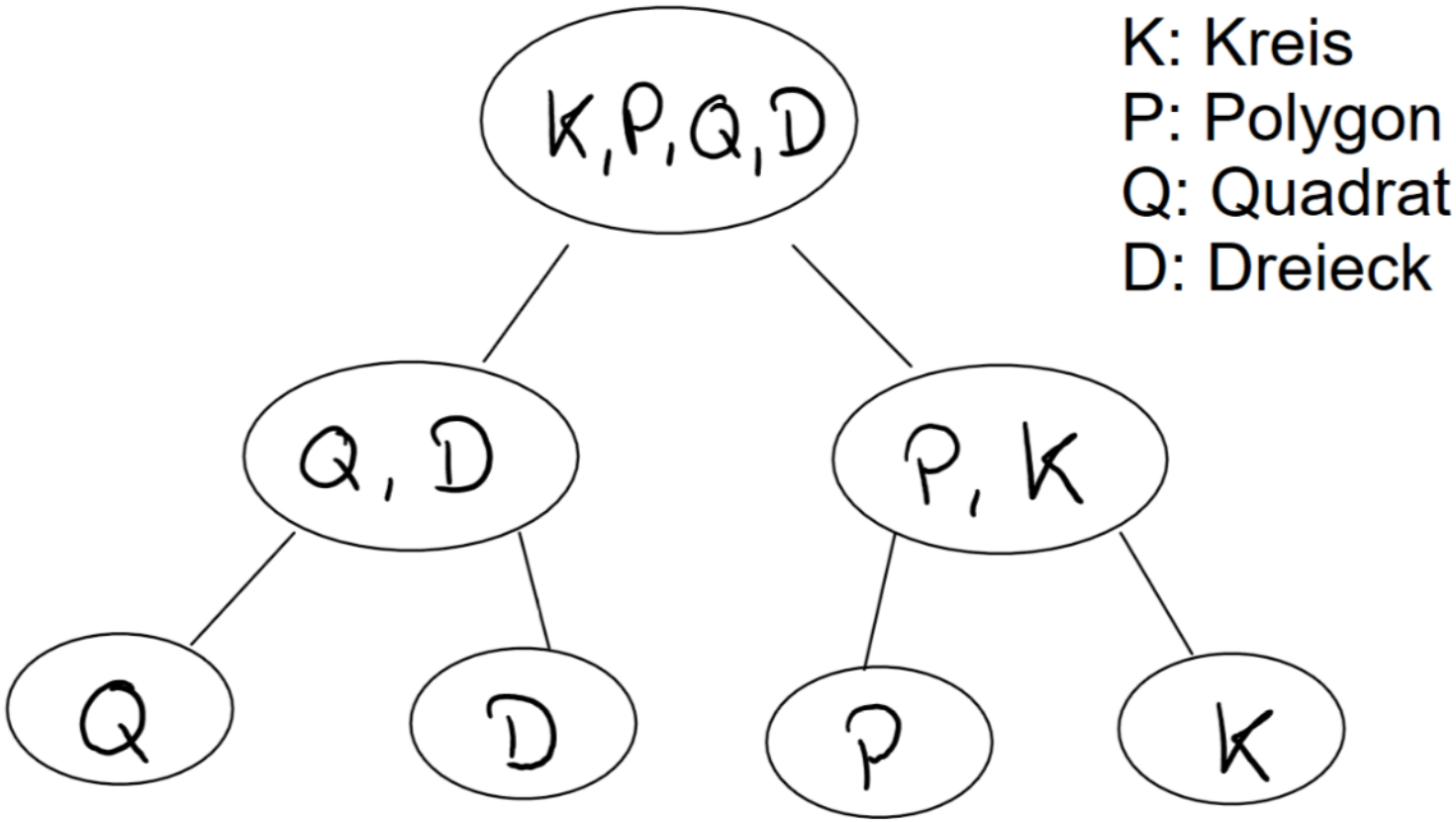
Aufgabe 3: Bäume
BSP

TECHNISCHE UNIVERSITÄT DARMSTADT



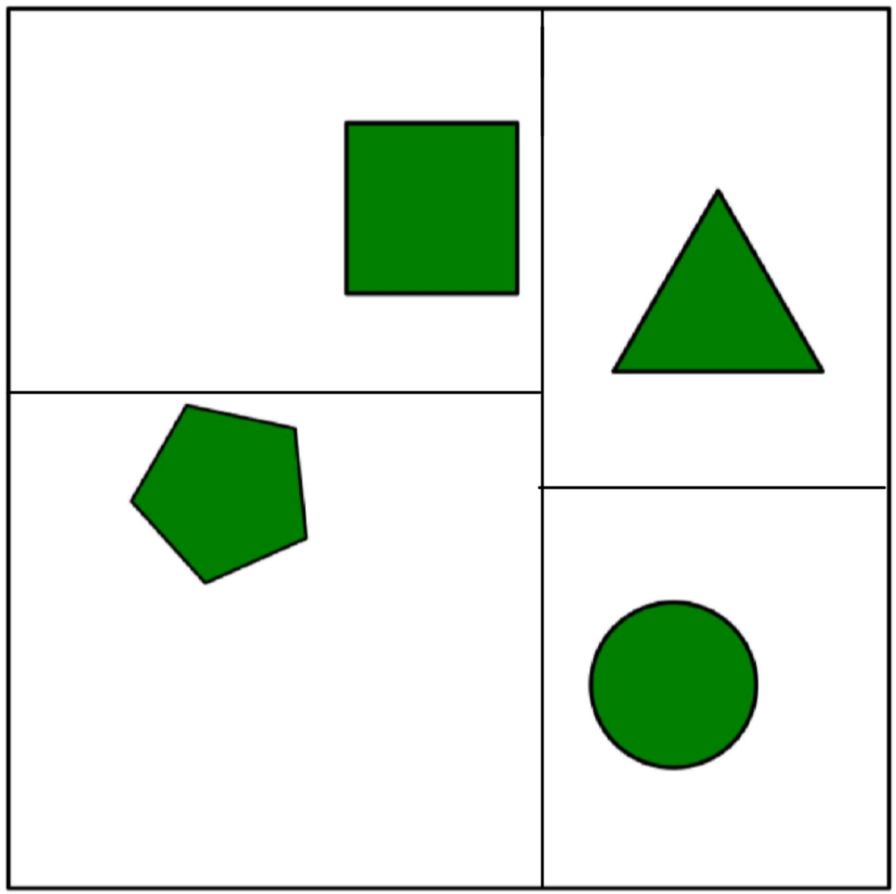
Aufgabe 3: Bäume
BSP

TECHNISCHE UNIVERSITÄT DARMSTADT



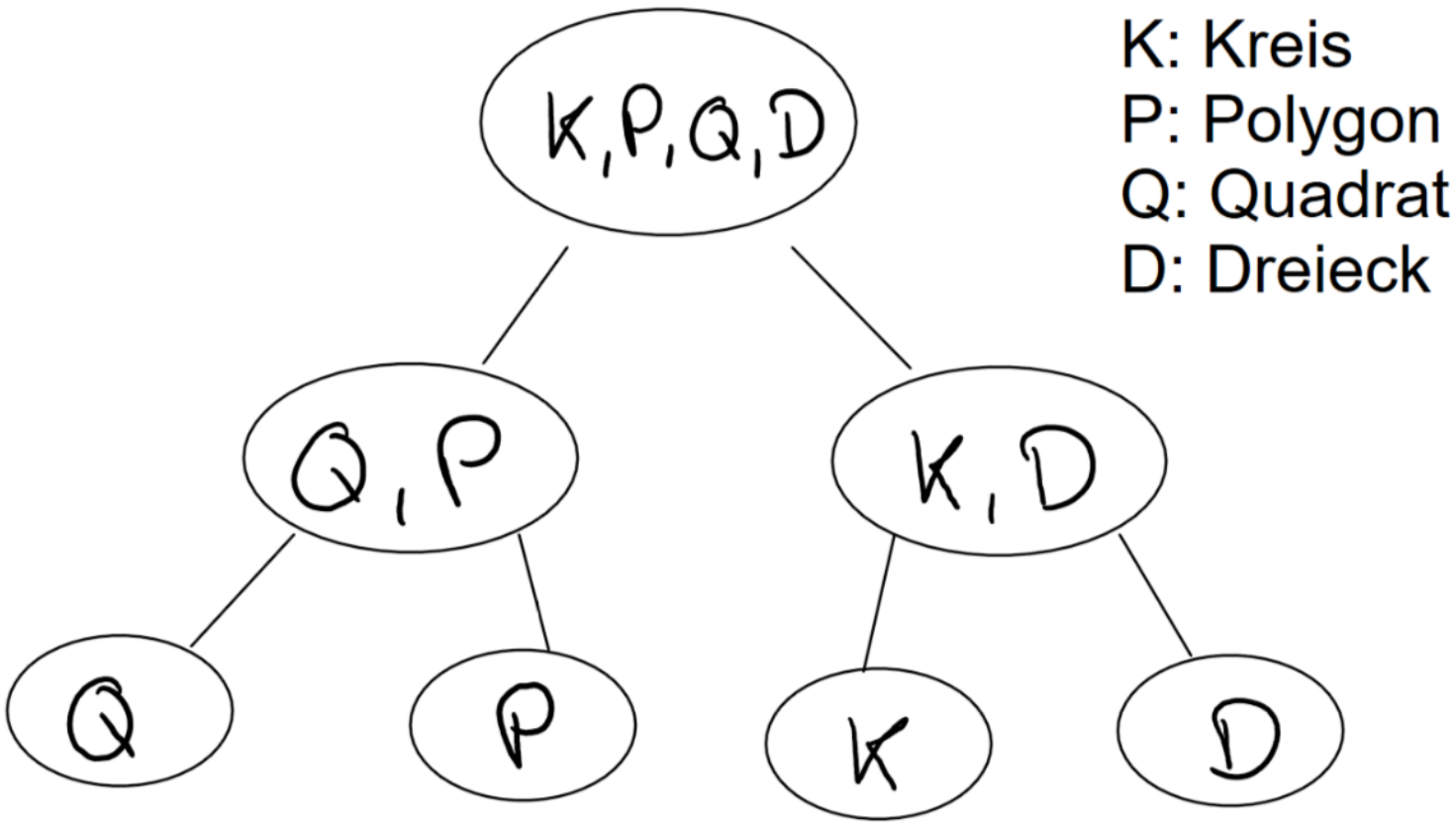
Aufgabe 3: Bäume
k-d

TECHNISCHE UNIVERSITÄT DARMSTADT



Aufgabe 3: Bäume
k-d

TECHNISCHE UNIVERSITÄT DARMSTADT



Aufgabe 4: Rasterisierung

TECHNISCHE UNIVERSITÄT DARMSTADT

Die Anwendung des Bresenham-Algorithmusliefert die Werte

i	Fehler	x_i	y_i	neuer_Fehler
0	3.5	2	2	3.5
1	3.5 - 4 = -0.5	3	3	-0.5 + 7 = 6.5
2	6.5 - 4 = 2.5	4	3	2.5
3	2.5 - 4 = -1.5	5	4	-1.5 + 7 = 5.5
4	5.5 - 4 = 1.5	6	4	1.5
5	1.5 - 4 = -2.5	7	5	-2.5 + 7 = 4.5
6	4.5 - 4 = 0.5	8	5	0.5
7	0.5 - 4 = -3.5	9	6	-3.5 + 7 = 3.5

. $\vec{x}_{Start} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ und $\vec{x}_{Ziel} = \begin{bmatrix} 9 \\ 6 \end{bmatrix}$.

Mit $dx = x_{end} - x_{start}$, $dy = y_{end} - y_{start}$ folgt für $1 \leq i \leq x_{end}$ gilt

$$Fehler_i = neuer\_Fehler_{i-1} - dy, \; neuer\_Fehler_i = \begin{cases} Fehler_i + dx & \text{,wenn } Fehler_i < 0 \\ Fehler_i & \text{,sonst} \end{cases},$$

$$x_i = x_{i-1} \text{ und } y_i = \begin{cases} y_{i-1} + 1 & \text{,wenn } Fehler_i < 0 \\ y_{i-1} & \text{,sonst} \end{cases}.$$

Sowie für $i = 0$ gelten die Startwerte, sowie $Fehler_0 = 0.5 \cdot dx$.

Aufgabe 4: Rasterisierung

TECHNISCHE UNIVERSITÄT DARMSTADT

damit folgt für die Grafik