

Aufgabe 23: Bernstein-Polynome (20 Punkte)

Die Bernstein-Polynome vom Grad $m > 0$ sind definiert für $0 \leq i \leq m$ als:

$$B_{i,m}(t) = \binom{m}{i} \cdot t^i \cdot (1-t)^{m-i}.$$

Zeigen Sie folgende Eigenschaften:

(a) Maximum: $B_{i,m}(t)$ besitzt bei $t = \frac{i}{m}$ sein Maximum

(b) Summation: $\sum_{i=0}^m i \cdot B_{i,m}(t) = m \cdot t$

(c) Symmetrie: $B_{i,m}(t) = B_{m-i,m}(1-t)$

(d) Differentiation:

$$\frac{d}{dt} B_{i,m}(t) = m \cdot (B_{i-1,m-1}(t) - B_{i,m-1}(t)) \quad \text{für } 1 \leq i \leq m-1$$

$$\frac{d}{dt} B_{0,m}(t) = -m \cdot (1-t)^{m-1} = -m \cdot B_{0,m-1}(t)$$

$$\frac{d}{dt} B_{m,m}(t) = m \cdot t^{m-1} = m \cdot B_{m-1,m-1}(t)$$

Aufgabe 24: Rekursive Berechnung der Bézier-Kurvenpunkte (10 Punkte)

Zeigen Sie, wenn P_1, P_2, \dots, P_{m+1} die Kontrollpunkte einer Bézier-Kurve $Q(t)$ vom Grad m sind, kurz $Q_{[1,m+1]}(t)$, dann gilt:

$$Q_{[1,m+1]}(t) = (1-t) \cdot Q_{[1,m]}(t) + t \cdot Q_{[2,m+1]}(t).$$

Aufgabe 25: Hierarchische Szenenmodellierung (20 Punkte)

Modellieren Sie mit den im Programmrahmen bereitgestellten Szenengraphkomponenten einen Roboter. Der Roboter soll aus Kopf (mit Sensoren), Rumpf (mit Armen und Fingern) sowie einem Bein mit einer Standfläche bestehen. Die Roboterteile sollen hierarchisch und konzeptionell im Szenengraphen abgebildet sein. Für eine kompakte Spezifikation des Roboter-Modells nutzen Sie die Möglichkeit, Teilgraphen des Szenengraphen mehrfach referenzieren zu können (*shared subgraphs*). Als geometrisches Primitiv verwenden Sie die Box, die durch ihre beiden Extrempunkte spezifiziert wird.

In Abbildung 1 ist die Software-Architektur der Szenengraph-Bibliothek dargestellt, deren Implementierung Sie im Programmrahmen finden. Vervollständigen Sie die Implementierung

der Klassen `Box`, `Rotation` und `Scaling`. Machen Sie sich mit dem Interface und der Implementierung der anderen Klassen vertraut.

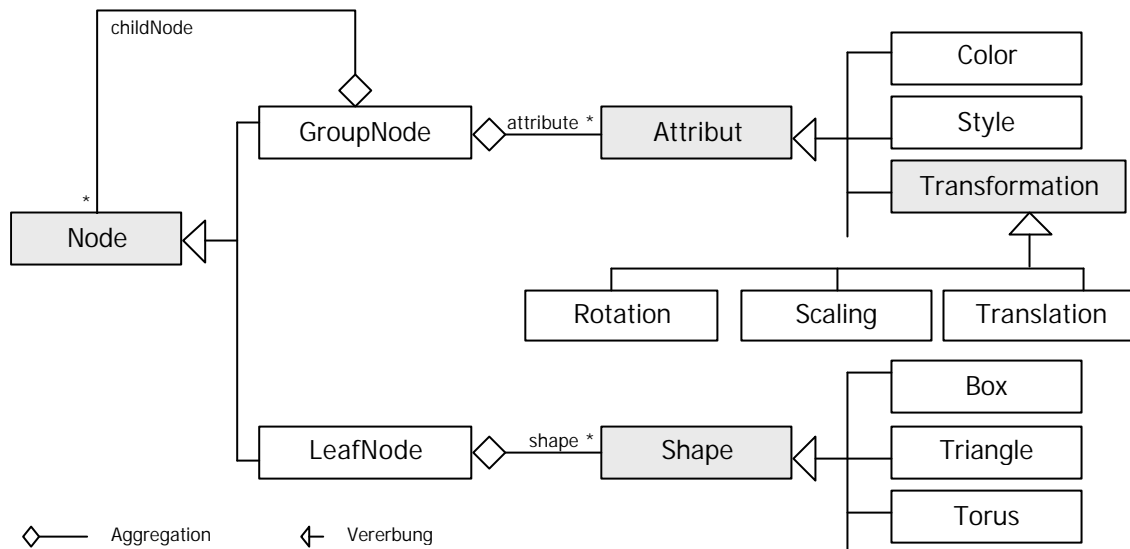


Abbildung 1 Klassendiagramm der zur Verfügung stehenden Szenengraph-Bibliothek

Aufgabe 26: Kameraausrichtung mit LookAt (20 Punkte)

Entwickeln Sie eine Klasse `LookAt`, die zur Steuerung der Kameraausrichtung verwendet werden soll. Die Klasse `LookAt` ist von der Klasse `Transform` abgeleitet und besitzt die Attribute `from`, `to` und `up`.

- Programmieren Sie mit Hilfe der Klasse `LookAt` eine Kameraanimation: Die Kamera soll sich auf einem von Ihnen vorgegebenen Pfad um ein Objekt, z.B. den Roboter aus Aufgabe 25, bewegen. Hängen Sie dazu ein Objekt der Klasse `LookAt` in den Szenengraphen und verwenden Sie zur zeitlichen Steuerung des Ablaufs die `onTimer` Methode aus dem Programmrahmen.
- Erweitern Sie die Kameraanimation dadurch, dass der Kamerapfad durch eine kubische Bézier-Kurve festgelegt werden kann. Die vier Kontrollpunkte der Bézier-Kurve sollen frei gesetzt werden können. Die Bézier-Kurvenpunkte sollen mit dem de-Casteljau-Algorithmus berechnet werden.

Aufgaben 23 und 24 sind einzeln abzugeben, die Aufgaben 25 und 26 können dagegen zu zweit bearbeitet werden. Senden Sie Ihre Lösungen bitte bis zum Montag, den 16.7.2001, **12:00**, an cgi2001@hpi.uni-potsdam.de. Alternativ können die Aufgaben 23 und 24 auch handschriftlich bearbeitet und am HPI abgegeben werden.

Die Klausur findet am Freitag, 27. Juli 2001, 12 – 15 Uhr im Haus 10 (SR+HS) in Golm statt.