

## Übung 4

Liebe TeilnehmerInnen der Grafischen Datenverarbeitung,

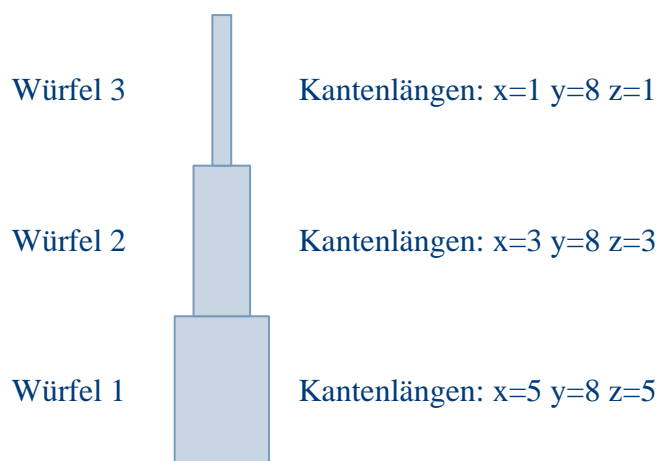
in dieser Übung sollen Sie sich an einfachen Animationen versuchen. Finden Sie zunächst einmal heraus, was eine Animationshierarchie ist. Sie finden eine Erläuterung dazu in den Vorlesungsunterlagen „Transformationen“ ab Folie 40. Wichtig im Zusammenhang mit Animationshierarchien ist, die Bedeutung der Matrizenmultiplikation für Transformationen verstanden zu haben. Das Ziel der Übung ist, einen Roboterarm aus drei Einzelteilen zu bewegen.

Erstellen Sie ein *YoshiX* Projekt. Fügen Sie dem Projekt die Datei „cube\_simple.cpp“ hinzu, die Sie in *Moodle* zusammen mit der Aufgabenstellung finden.

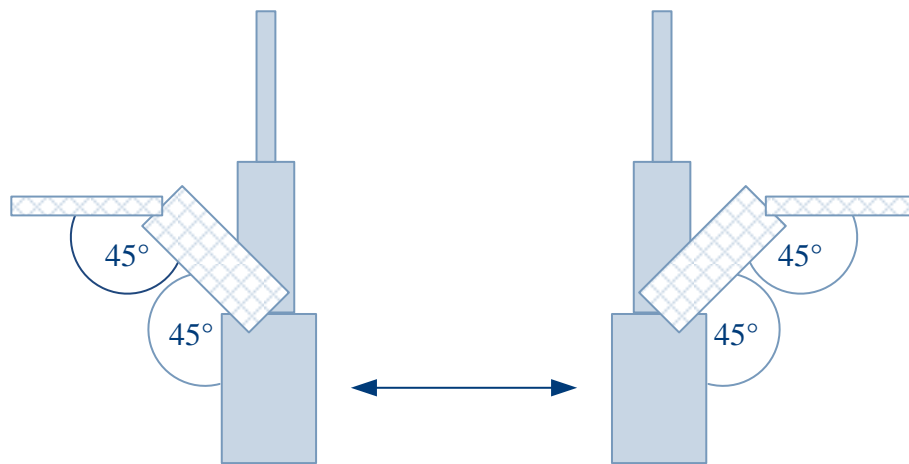
Schauen Sie sich in der Datei die Methode *InternOnFrame* an. Sie sehen an dem Beispiel, wie Sie das gleiche Mesh mit einer Folge von *SetWorldMatrix* + *DrawMesh* Aufrufen mehrmals mit verschiedenen Transformationen zeichnen können.

Legen Sie in der Datei einen *struct SNode* an. Dieser beinhaltet einen Rotationswinkel, eine Position, eine nicht-uniforme Skalierung, eine lokale und eine globale 4x4 Matrix (siehe Folien). Außerdem hat ein Knoten einen Pointer auf einen Kindknoten.

Der Roboterarm soll nur aus transformierten Würfeln mit den unten angegebenen Abmessungen bestehen. Legen Sie 3 Instanzen von *SNode* an und weisen Sie den Würfeln die richtigen Kinder zu. Würfel 1 bildet dabei die Wurzel der Animationshierarchie. Würfel 3 ist ein Blattknoten, weshalb sein Verweis auf einen Kindknoten NULL entspricht.



Der Roboter startet an die Position  $P_1(-4,0,0)$ . Dann dreht er die Arme entsprechend der folgenden Abbildung um die  $z$ -Achse. Die Drehachse verläuft entlang der unteren Mitte des Würfels. In jedem Frame wird maximal um ein Grad gedreht. Anschließend fahren die Arme wieder zurück in die Ausgangsstellung. Nun fährt der Arm an Position  $P_2(4,0,0)$ . Pro Frame wird dabei die Distanz 0.1 zurückgelegt. Wieder werden die Arme gedreht, allerdings in die umgekehrte Richtung. Der Greifarm kehrt danach in die Ausgangsstellung zurück und fährt wieder an die Position  $P_1$ . Der Ablauf wiederholt sich endlos.



Hier noch einige Hinweise:

- Die lokale Matrix von Würfel 1 beschreibt keine Rotation, sondern nur eine Verschiebung.
- Die Verschiebung von Würfel 1 entspricht der Position des Greifarms.
- Die globale Matrix von Würfel 1 entspricht seiner lokalen Matrix.
- Der Würfel 2 ist zu Würfel 1 um den Versatz 4 in y-Richtung verschoben.
- Der Würfel 3 ist zu Würfel 2 um den Versatz 4 in y-Richtung verschoben.
- Der Versatz sollte mit den lokalen Matrizen der Würfel 2 und 3 bestimmt werden.
- Eine Drehung der Würfel um die vorgesehene Achse ist nicht ohne weiteres möglich.
- Die Standardrotationsmatrizen erlauben nur Drehungen um die Hauptachsen.
- Die Drehung muss somit durch eine  $T \cdot R \cdot T^{-1}$  Kombination beschrieben werden.
- Die resultierende Matrix geht ebenfalls in die lokale Matrix der Würfel 2 und 3 ein.

Traversieren Sie in der *InternOnFrame* Methode Ihren Minibaum in einer Schleife. Berechnen Sie für jeden Knoten anhand seiner eingetragenen Werte die lokale Matrix. Multiplizieren Sie die lokale Matrix des Kindknoten mit der globalen Matrix des Elternknoten. Das Resultat ist die globale Matrix des aktuellen Kindknotens. Hat ein Knoten keinen Elternknoten (Wurzel), dann ist die globale Matrix gleich der lokalen Matrix. Setzen Sie die globalen Matrizen mittels der *SetWorldMatrix* Methode. Rufen Sie direkt danach die *DrawMesh* Methode auf.