

Aufgabe 14: Quader-Quader-Transformation (5 Punkte)

Entwickeln Sie eine Transformation, die einen achsenparallelen Quader, gegeben durch die untere linke Ecke (x_1, y_1, z_1) und die obere rechte Ecke (x_2, y_2, z_2) , transformiert in einen zweiten achsenparallelen Quader, gegeben durch die untere linke Ecke (x_3, y_3, z_3) und die obere rechte Ecke (x_4, y_4, z_4) . Geben Sie die 4×4 -Transformationsmatrix an.

Aufgabe 15: Würfelrotation (5 Punkte)

Entwickeln Sie eine Transformation, die einen Einheitswürfel mit den Eckkoordinaten $(0,0,0)$ und $(1,1,1)$ um f Grad an der Diagonalen als Rotationsachse rotiert. Die Rotationsrichtung sei gegen den Uhrzeigersinn, wenn von der Ecke $(1,1,1)$ entlang der Diagonalen zum Ursprung geblickt wird. Geben Sie die 4×4 -Transformationsmatrix an.

Aufgabe 16: Rotation um eine beliebige Achse (10 Punkte)

Entwickeln Sie eine Transformation $M(U, f)$, die eine Rotation an einer beliebigen Rotationsachse $U = (u_x, u_y, u_z)$ um f Grad beschreibt. Hinweis: Transformieren Sie dazu die Rotationsachse U auf die z -Achse, rotieren sie an der z -Achse um f Grad und transformieren Sie anschließend von der z -Achse zurück auf die Rotationsachse U . Geben Sie die 4×4 -Transformationsmatrix an.

Aufgabe 17: Spiegelung an einer beliebigen Ebene (10 Punkte)

Entwickeln Sie eine Transformationsfolge, die ein geometrisches Objekt an einer beliebigen Ebene im Raum (Spiegelebene) reflektiert. Die Ebene sei gegeben durch einen Punkt P_E in dieser Ebene und den Ebenen-Normalenvektor N_E . Geben Sie die Reihenfolge und die einzelnen 4×4 -Transformationsmatrizen an.

Aufgabe 18: Geometrische Transformationen in OpenGL (15 + 5 Punkte)

Modellieren und rendern sie ein Modell eines Tretrollers. Erstellen Sie ein hierarchisches, konzeptionelles Modell der Geometrie eines Tretrollers und legen Sie darin die Transformationen von den lokalen Koordinatensystemen zum Weltkoordinatensystem fest. Per Konvention soll das Trittbrett in der xz -Ebene liegen, die Vertikale verläuft entlang der y -Achse. Das Tretrollermodell soll die wesentlichen Bauteile computergraphisch, aber zugleich vereinfacht visualisieren.

Verwenden Sie zum Aufbau des Tretrollermodells die Primitive, die mit Hilfe von Methoden des Programmrahmens bereitgestellt werden:

- `drawBox()` zeichnet einen Quader mit Seitenlänge 1. Der Quader ist im Ursprung zentriert.
- `drawSphere()` zeichnet eine Kugel mit dem Radius 1. Die Kugel ist im Ursprung zentriert.
- `drawCylinder(bool caps)` zeichnet einen Zylinder mit dem Radius 0.5, den Deckelflächen in Höhe 0 und 1 und der z -Achse als Mittelachse. `caps` gibt an, ob der Zylinder mit oder ohne Deckelflächen gezeichnet wird.



Abbildung 1 2-Rad-Tretroller und 3-Rad-Tretroller (links). Modell eines vereinfachten "Scooters" (rechts).

- *drawTorus(float innerRadius, float outerRadius)* zeichnet einen Torus mit dem inneren Radius *innerRadius* und dem äußeren Radius *outerRadius*. Der Torus liegt in der *xz*-Ebene und ist im Ursprung zentriert.

Zur Positionierung, Orientierung und Skalierung der Primitive können Sie auf die Transformationsfunktionen von OpenGL *glTranslate*, *glRotate* und *glScale* zurückgreifen. Gegebenenfalls können weitere Primitive von Ihnen implementiert und verwendet werden.

Das Programm soll folgende Benutzereingaben verstehen: Mit der Taste ‚a‘ können die Koordinatenachsen eingeblendet und mit der Leertaste kann die Szene um die *y*-Achse rotiert werden. Diese Funktionen sind im Programmrahmen bereits realisiert. Beispiele für den geometrischen Aufbau von Tretrollern finden Sie in Abbildung 1.

Für die korrekte Bearbeitung dieses Aufgabenteils bekommen Sie 15 Punkte.

Hinweis: Ein hierarchisches Modell vereinfacht die Modellierung. Zum Beispiel können Sie eine Methode *drawWheel* implementieren, die ein einzelnes Rad zentriert im Ursprung zeichnet, das dann durch Transformation entsprechend im Tretrollergesamtkoordinatensystem positioniert wird.

Zusatzaufgabe: Animieren Sie in Ihrer Implementierung von Aufgabe 18 die Räder und den Lenker (5 Punkte).

Aufgaben 14 bis 17 sind einzeln abzugeben, die Aufgabe 18 kann dagegen zu zweit bearbeitet werden. Senden Sie Ihre Lösungen bitte bis zum Donnerstag, den 21.6.2001, an cgi2001@hpi.uni-potsdam.de. Alternativ können die Aufgaben 14 bis 17 auch handschriftlich bearbeitet und am HPI abgegeben werden.