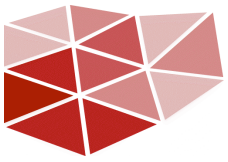


CG1 Übungsblatt 1

25.Oktober 2011

Abgabe: Donnerstag, 15. November 2011

AUFGABE 1: **Traversierung eines Szenengraphen / Affine Transformationen**

Auf Basis des C++ Skelett-Codes (http://www.cg.tu-berlin.de/menue/teaching/computer_graphics_1) ist die Konstruktion und Traversierung eines Szenengraphen inklusive der erforderlichen affinen Transformationen zu implementieren. Im existierenden Code ist lediglich der Torso eines Roboters vorhanden. Ein möglicher Lösungsvorschlag ist als Binärdatei im zip Archiv beigefügt (cg1_ex1.exe für win32 und cg1_ex1 für linux). Die Aufgaben im Einzelnen sind:

a) Erstellen Sie einen Szenengraphen welcher zur Repräsentation eines Roboters geeignet ist (siehe `robot.cpp` und `robot.h`). Es ist nicht erforderlich, den Roboter aus der mitgelieferten Binärdatei perfekt abzubilden, jedoch sollte der konstruierte Szenengraph mindestens die Tiefe Zwei besitzen. (1 Punkt)
b) Implementieren Sie die Traversierung des Szenengraphen (siehe `scenegraph.cpp` und `scenegraph.h`). (1 Punkt)
c) Implementieren sie die Transformation der Knoten im Szenengraphen (siehe `node.cpp` und `node.h`). Dabei ist zu beachten, dass nicht alle Knoten ihren Rotationsmittelpunkt im lokalen Zentrum/Schwerpunkt haben müssen. (1 Punkt)
d) An den jeweiligen Rotationsmittelpunkten der Knoten im Szenengraphen sollen die lokalen Koordinatenachsen x, y und z in Rot, Grün und Blau, sowie eine Kugel (`glutWiredSphere()`) gezeichnet werden. Diese sollen mit dem jeweiligen Knoten im Szenengraph rotieren. (1 Punkt).
e) Implementieren Sie eine Reset-Funktionalität, die alle Rotationen im Szenengraph auf Null zurücksetzt. Diese Funktion soll sowohl durch die rechte Maustaste in einem Menü als auch durch Tippen von 'r' auf der Tastatur anwählbar sein (siehe `context.cpp` und `context.h`). (1 Punkt)
f) Demonstrieren Sie anhand eines beliebigen Knotens im Szenengraph das Phänomen des Gimbal-Lock, und erläutern sie dessen Ursachen. (1 Punkt)

Zusatzpunkte (freiwillig):

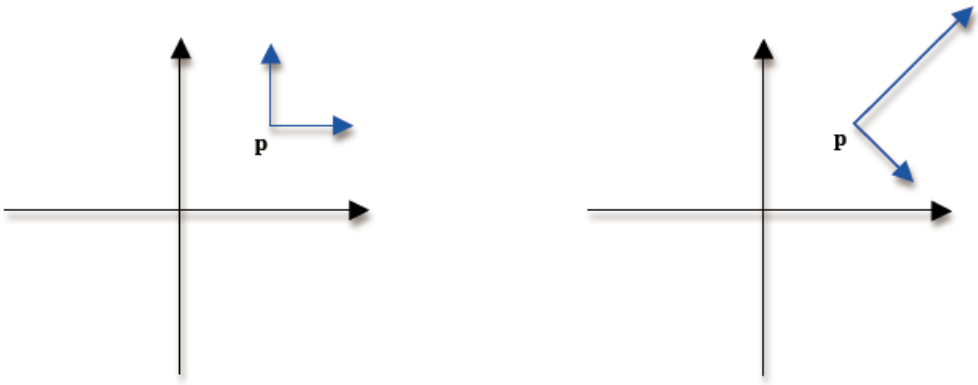
g) Um die Benutzerinteraktion zu verbessern, implementieren Sie einen Virtual Trackball, der intuitive und robuste Rotation eines jeden Knotens per Maus ermöglicht (wie in Vorlesung und Übung besprochen). (1 Punkt)
h) Ermöglichen Sie die Auswahl von Gliedmaßen per Maus. Greifen Sie dabei nach Möglichkeit nicht auf deprecated OpenGL-Funktionalität zurück. (1 Punkt)

AUFGABE 2: **Theoriefragen**

a) Beschreiben Sie die folgende Transformation des lokalen Koordinatensystems an **p** = (px,py) (blau) mit elementaren 3x3 Transformationsmatrizen (1 Punkt). Welchen Effekt hat die Transformation auf einen Kreis, der am Ursprung zentriert ist? Welchen Effekt hat die Transformation auf einen Kreis, der an **p** zentriert ist? (1 Punkt)

Technische Universität Berlin
Computer Graphics

Prof. Dr. Marc Alexa,
Katrin Lang, Olivier Rouiller, Ronald Richter



b) Leiten Sie schrittweise eine Formel her, die einen Vektor **v** um eine beliebige Achse, gegeben durch den Einheitsvektor **k**, mit dem Winkel **θ** rotiert. Dabei sollen ausschliesslich Vektoroperationen, jedoch *keine* Matrixoperationen, verwendet werden. Folgende Schritte sind dazu notwendig:
   1. Bestimmen Sie die Projektion $\mathbf{v}_k$ des Vektor **v** auf Vektor **k**. (0,5 Punkte)
   2. Bestimmen Sie zwei orthogonale Basisvektoren $\mathbf{v}_1$ und $\mathbf{v}_2$, die die Fläche senkrecht zu Vektor aufspannen. (0,5 Punkte)
   3. Bestimmen Sie die um den Winkel **θ** rotierten Vektoren $\mathbf{v}_1'$, $\mathbf{v}_2'$ und $\mathbf{v}_k'$. (0,5 Punkte)
   4. Fassen Sie alle Teilschritte zu einer Formel in Abhängigkeit von **v**, **k** und **θ** zusammen, die den rotierten Vektor **w** bestimmt. (0,5 Punkte)

c) Gegeben ist die homogene Transformationsmatrix **M** als Produkt einer Translationsmatrix **T** und einer Rotationsmatrix **R**, **M** = **TR**. Bestimmen Sie die Inverse der Transformationsmatrix ($\mathbf{M}^{-1}$) unter Berücksichtigung bzw. Ausnutzung der Eigenschaften der gegebenen Matrizen. (1 Punkt)

d) Für welche Kombinationen zweier Transformationen (Rotation, isotrope/anisotrope Skalierung, Translation) ist die Reihenfolge der Anwendung irrelevant? Es gibt 10 mögliche Kombinationen. Erstellen Sie eine Tabelle.
(2 Punkte)

e) In der Programmieraufgabe wird double buffering verwendet (siehe Funktion `display()` in `context.c`). Was passiert beim double buffering und wieso wird es in dieser Aufgabe verwendet?
(1 Punkt)