



AUFGABE 1: Traversierung eines Szenengraphen / Affine Transformationen

Auf Basis des C skelett-Codes (verfügbar unter <http://www.cg.tu-berlin.de/cg1.html>) 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) Der Aufbau eines sinnvollen Szenengraphen (Siehe robot.c und robot.h). Dabei ist es nicht erforderlich, den Roboter aus der mitgelieferten Binärdatei perfekt abzubilden, jedoch sollte der konstruierte Szenengraph (der Baum) mindestens die Tiefe Zwei besitzen. (1 Punkt)
- b) Die Traversierung des Szenengraphen (Siehe scenegraph.c und scenegraph.h). (1 Punkt)
- c) Die Transformation der einzelnen Szenengraph-Knoten (Siehe scenegraph.c und scenegraph.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 Szenengraph-Knoten 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 rotieren. (1 Punkt)
- e) Jeder Knoten des Szenengraphen soll per rechtsklick-Menü anwählbar und rotierbar sein (siehe glutcallbacks.c). (1 Punkt)
- f) Anhand eines beliebigen Knotens im Szenengraph soll das Phänomen des Gimbal-Lock erläutert werden. (1 Punkt)

AUFGABE 2: Theoriefragen

1. Für welche Kombinationen zweier Transformationen (Rotation, isotrope/anisotrope Skalierung, Translation) ist die Reihenfolge der Anwendung irrelevant? Es gibt 10 mögliche Kombinationen. (2 Punkte)
2. Sie sollen einen Vektor \mathbf{p}_1 durch eine Folge von Rotationen um die x-, y- und/oder z-Achse in einen Vektor \mathbf{p}_2 überführen.
 - a. Welche Bedingung müssen \mathbf{p}_1 und \mathbf{p}_2 erfüllen, damit das überhaupt möglich ist? (0,5 Punkte)
 - b. Beschreiben Sie einen Algorithmus, der die notwendigen Operationen (Achsen und Winkel) bestimmt. (1 Punkt)
 - c. Wieviele Rotationen sind bei geschicktem Vorgehen maximal notwendig? (0,5 Punkte)
3. Diskutieren Sie Vor- und Nachteile von gerichteten, azyklischen Graphen im Vergleich zu Bäumen als Szenengraphen. (1 Punkt)
4. In der Programmieraufgabe wird *double buffering* verwendet (siehe Funktion display() in glutcallbacks.c). Was passiert beim *double buffering* und wieso wird es in dieser Aufgabe verwendet? (1 Punkt)