



AUFGABE 1: Projektive Transformationen / Kanonisches Sichtvolumen

Nate Robins' OpenGL Projektions-Tutorial, verfügbar unter http://www.cg.tu-berlin.de/cg1_07.html, ist im Rahmen dieser Übung um eine Visualisierung des kanonischen Sichtvolumens (clip-space view) zu erweitern (im zip-Archiv ist hierzu eine Binärdatei als Beispiel enthalten). Die Aufgaben im Einzelnen sind:

- a) Das bestehende Tutorial ist um ein zusätzliches GLUT sub-Fenster zu erweitern, siehe `glutCreateSubWindow` in der Datei `projection.c` und, als Vorlage, die Binärdatei `cg1_ex2.exe`. (1 Punkt)
- b) Analog zu den world-space und screen-space Views ist eine clip-space View zu implementieren, in der das kanonische Sichtvolumen und das projektiv transformierte Modell dargestellt werden. Hierzu müssen die callback Funktionen `clip_display`, `clip_reshape` und `clip_menu` erstellt und registriert werden. (2 Punkte) **Hinweise:** a) die projektive Transformation in OpenGL invertiert die z-Achse. b) die aktuellen Projektions- und Modelview-Matrizen des screen-space Views sind in den Arrays `modelview` und `projection` gespeichert (siehe `projection.c`). c) Das kanonische Sichtvolumen ist in OpenGL ein Würfel mit den Extremwerten $(-1,-1,-1)$ und $(1,1,1)$.
- c) Die clip-space View soll durch Ziehen der Maus um den Ursprung rotieren. (1 Punkt)
- d) Per Knopfdruck (und/oder dem Rechtsklick-Menü) sollen sechs zusätzliche OpenGL clip Planes ein- und ausgeschaltet werden können. Diese sollen ein Clipping des kanonischen Sichtvolumens simulieren. (1 Punkt).
- e) **Bonuspunkt:** Die Beleuchtung des Modells in der clip-space View wird durch die projektive Transformation nicht korrekt dargestellt, da die Normalenvektoren ebenfalls projektiv transformiert werden. Um dies zumindest teilweise zu beheben soll `glEnable(GL_NORMALIZE)` verwendet werden. Wie könnte eine korrekte Lösung dieses Problems aussehen? (1 Punkt)

AUFGABE 2: Theoriefragen

1. Wieso wird in OpenGL das Clipping am kanonischen Sichtvolumen vorgenommen? (1 Punkt)
2. Wie kann man mittels Outcodes überprüfen, ob eine Gerade in 3D ein ebenes, konvexes Polygon schneidet? (0,5 Punkte)
3. Entwickeln Sie den Outcode für das kanonische Sichtvolumen. Welche Primitive können im 3D-Fall behandelt werden? (1 Punkt)
4. Erläutern Sie die Dualität von Ebenen und Punkten im projektiven 3-Raum. (1 Punkt)
5. Was ist die Komplexität von Clipping eines einfachen (nicht notwendigerweise konvexen) Polygons an einem konvexen Polygon? Auf welches „Primitiv“ bezieht sich hier die Komplexitätsangabe? (1 Punkt)
6. Warum wird Backface-Culling am besten im kanonischen Sichtvolumen durchgeführt? (0.5 Punkte)