

Aufgabe 19: Perspektivische Division (10 Punkte)

Begründen Sie, warum zur Spezifikation einer Zentralprojektion mit einer Matrix homogene Koordinaten notwendig sind. An welcher Stelle taucht die perspektivische Division auf, welche Art von Transformation verbirgt sich dahinter? Illustrieren Sie letzteres an einer Prinzipskizze.

Aufgabe 20: Perspektivische Projektionstransformationsmatrix (15 Punkte)

a) Berechnen Sie die Orientierungs- und Projektionsmatrix, die nötig ist, um das durch folgende Kameraeinstellungen gegebene Sichtvolumen in das Standard-Sichtvolumen zu überführen:

- Kamerastandpunkt: $P = (0, 0, 0)$
- Blickrichtungsvektor: $Look = (-3, -4, 0)$
- Aufwärtsvektor: $Up = (-4, 3, 0)$
- Öffnungswinkel in der Breite: $q_W = 90 \text{ Grad}$
- Aspect Ratio: 3:2
- Entfernung zur Far-Clipping Plane: $far = 6$
- Entfernung zur Near-Clipping Plane: $near = 1$

b) Gegeben ist folgende Matrix:

$$\begin{pmatrix} -\frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Die Matrix entspricht dem Ergebnis der Multiplikation der Projektionsmatrix mit der Orientierungsmatrix. Berechnen Sie die Position der Kamera, den Blickrichtungsvektor und den Aufwärtsvektor der Kamera, die Öffnungswinkel q_W und q_H der Projektion sowie die Entfernungen zur Near- und Far-Clipping Plane.

Aufgabe 21: Kameraspezifikation (10 Punkte)

Die Kameraposition und Kameraorientierung wird in OpenGL im Allgemeinen mit der GLU-Routine `gluLookAt` definiert. Ein alternatives Kamera-Paradigma positioniert die Kamera auf einer orbitalen Position bezüglich des Koordinatenursprungs. Die Kamera ist dabei immer zum Ursprung ausgerichtet.

In diesem Fall ist die Kameratransformation in Polarkoordinaten spezifiziert. Die Variable *distance* beschreibt den Abstand der Kamera vom Ursprung und definiert den Radius des Orbits. Die Kameraposition wird somit um *distance* Einheiten in die negative *z*-Achse verschoben. Die Variable *azimuth* definiert den Rotationswinkel der Kamera bezüglich der *xy*-Ebene, gemessen von der positiven *x*-Achse. Entsprechend definiert die Variable *elevation* den Rotationswinkel der Kamera bezüglich der *yz*-Ebene, gemessen von der positiven *y*-Achse. Die Variable *twist* definiert eine Rotation der Kamera um ihre Sichtrichtung.

- Illustrieren Sie, wie mit der angegebenen Spezifikation Position, Lage und Ausrichtung der Kamera festgelegt werden.
- Entwickeln Sie eine OpenGL-Routine zur Umsetzung des vorgestellten Kamera-Paradigmas und geben dies in Form eines Source-Code-Fragments wieder.

Hinweis: Die Standardorientierung der Kamera entspricht in OpenGL der Initialisierung des ModelView-Stacks mit der Einheitsmatrix.

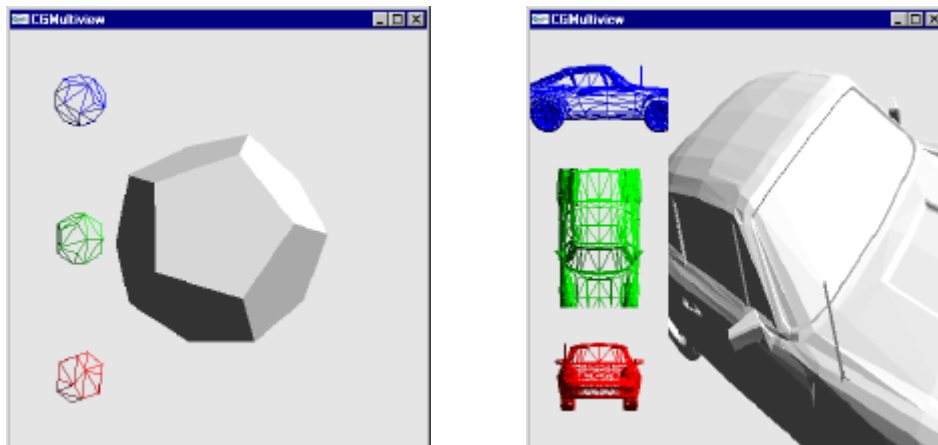
Aufgabe 22: Viewport-Kontrolle in OpenGL (15 Punkte)

Implementieren Sie eine Anwendung, die ein geometrisches Objekt in einer Hauptansicht und in drei Nebenansichten zeigt. Die Hauptansicht soll eine perspektivische Projektion verwenden. Die Nebenansichten sollen die Hauptrisse des geometrischen Objekts zeigen, d.h. sie verwenden orthogonale Projektionen. Alle Ansichten sollen in einem einzigen OpenGL-Fenster integriert werden. Dies geschieht durch Konfiguration des Viewports für jede einzelne Sicht.

Zusätzlich soll folgende Funktionalität auf folgende Tasten implementiert werden:

- „+“ und „-“ zoomen in der Hauptansichtsebene,
- „1“ und „2“ zoomen in der Sichtebene, die parallel zur xy -Ebene liegt,
- „3“ und „4“ zoomen in der Sichtebene, die parallel zur xz -Ebene liegt und
- „5“ und „6“ zoomen in der Sichtebene, die parallel zur yz -Ebene liegt.
- „1“ wechselt den Polygonzeichenmodus der Nebenansichten von gefüllt auf Liniendarstellung und umgekehrt.
- „L“ wechselt den Polygonzeichenmodus der Hauptansicht von gefüllt auf Liniendarstellung und umgekehrt.

Verwenden Sie die in den Übungen besprochenen OpenGL-Befehle `glViewport`, `glOrtho`, `gluPerspective` und `gluLookAt` sowie den bereitgestellten Programmrahmen `CGMultiView`.



Mehrere Ansichten integriert in einem einzigen OpenGL-Fenster.

Aufgaben 19 bis 21 sind einzeln abzugeben, die Aufgabe 22 kann dagegen zu zweit bearbeitet werden. Senden Sie Ihre Lösungen bitte bis zum Montag, den 2.7.2001, **12:00**, an cgi2001@hpi.uni-potsdam.de. Alternativ können die Aufgaben 19 bis 21 auch handschriftlich bearbeitet und am HPI abgegeben werden.