Prof. G. Zachmann
J. Boeckers
N. M. Jadid
A. Mühlenbrock

Universität Bremen
Fachbereich 3
Arbeitsgruppe CGVR
25. Oktober 2024

Wintersemester 2024/25

Übungen zu Computergrafik - Blatt 2

Abgabe am 03.11.2024, 23:59 Uhr

Aufgabe 1 (Vektoren und Matrizen (C++), *5 Punkte*)

Dies ist die erste C++ Programmieraufgabe. Lade Dir dazu das VectorsAndMatrices-Framework von der CGVR-Webseite herunter und stelle sicher, dass Du einen C++ Compiler, CMake sowie eine IDE installiert hast. Mehr dazu erfährst Du im Tutorium.

Das Ziel dieser ersten Aufgabe ist, eine Vektor- und Matrix-Klasse zu implementieren, damit wir in zukünftigen Aufgabenblättern einfach mit *Punkten*, *Vektoren* und *Matrizen* rechnen können. Wir verwenden dabei für *Punkte* und *Vektoren* eine gemeinsame C++ Klasse, die im Framework den Namen `Vec4f` besitzt. Für Matrizen verwenden wir die Klasse `Mat4f`.

Hintergrund, Erklärung und Hinweise

Damit wir Vektoren und Punkte voneinander unterscheiden können, besitzt die `Vec4f`-Klasse anstatt der üblichen drei Komponenten (x, y, z) die vier Komponenten (x, y, z, w). Dabei soll die vierte Komponente w für einen Punkt vorerst immer den Wert $w = 1$ und für einen Vektor den Wert $w = 0$ speichern, d.h.:

- ein Punkt **p** enthält immer die vier Komponenten $(p_x, p_y, p_z, 1)$
- ein Vektor **v** enthält immer die vier Komponenten $(v_x, v_y, v_z, 0)$

Für Transformationen im dreidimensionalen Raum – z.B. Rotation oder Skalierung – hast Du vermutlich bereits 3 x 3 Matrizen kennengelernt, die mit einem Punkt oder Vektor multipliziert werden können. Da bei der Multiplikation von Matrix und Vektoren allerdings die Spaltenanzahl der Matrix gleich der Zahl der Komponenten des Vektors sein muss, und unsere `Vec4f`-Klasse vier Komponenten besitzt, speichert die `Mat4f`-Klasse ebenfalls als 4 x 4 Matrix. Aus einer 3 x 3 Matrix kann man wie folgt eine entsprechende 4 x 4 Matrix erstellen:

$$\begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \widehat{=} \begin{bmatrix} m_{00} & m_{01} & m_{02} & 0 \\ m_{10} & m_{11} & m_{12} & 0 \\ m_{20} & m_{21} & m_{22} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Warum es in der Computergrafik so sinnvoll und auch üblich ist, vier Komponenten für dreidimensionale Vektoren und Punkte zu verwenden, ebenso wie eine 4 x 4 Matrix anstelle eine 3 x 3 Matrix zu nutzen, erfährst Du später in der Vorlesung im Abschnitt „Homogene Koordinaten“.

Achtung: Wir verwenden zur einfachen Kompatibilität mit OpenGL in den Aufgabenblättern immer die *Column-Major-Order*, d.h. wir speichern die Komponenten Spaltenweise in ein Array:

`float data[16] =` $\{m_{00}, m_{10}, m_{20}, m_{30}, m_{01}, m_{11}, ..., m_{33}\}$

Framework

Das Framework enthält folgende Dateien:

- `main.cpp`: Diese Klasse enthält die *main*-Funktion, was der Startpunkt unserer Applikation darstellt. Sie erstellt ein Fenster und initialisiert OpenGL und enthält die Main-Loop. An dieser Klasse sollen keine Änderungen vorgenommen werden.
- `simple_gui.h`: Diese Datei enthält genau eine Funktion, um die GUI im Fenster mithilfe der Library ImGui zu rendern, die Du bei Programmstart siehst. An dieser Klasse sollen keine Änderungen vorgenommen werden.
- `Vec4f.h` und `Vec4f.cpp`: Diese Klasse repräsentiert wahlweise einen 3D-Vektor oder 3D-Punkt.
- `Mat4f.h` und `Mat4f.cpp`: Diese Klasse repräsentiert eine 4 x 4 Matrix, die in späteren Aufgabenblättern zur Transformation von 3D-Punkten und 3D-Vektoren verwendet wird.

Aufgabenstellung

Deine Aufgabe ist, die Implementierung der Source-Dateien `Vec4f.cpp` und der `Mat4f.cpp` entsprechend der Dokumentation in den zugehörigen Header-Dateien `Vec4f.h` und der `Mat4f.h` zu vervollständigen. Beachte alle Todo's in beiden CPP-Dateien.

Aufgabe 2 (Vektoren und Matrizen, *0,5+0,5 Punkte*)

a) Wie verhält sich das Ergebnis der Multiplikation $M * \mathbf{v}$ einer 4 x 4 Matrix M mit einem Vektor $\mathbf{v}$, wenn die vierte Spalte der Matrix die Werte $(x, y, z, 1)$ mit $x \neq 0$, $y \neq 0$ und $z \neq 0$ enthält und $v_w = 0$ ist? (max. 2 Sätze!)

b) Wie verhält es sich in (a), wenn $v_w = 1$ ist? (max. 2 Sätze!)

Hinweis: Probiere es im Programm aus, welches Du in Aufgabenstellung 1 entwickelt hast.

Aufgabe 3 (Kreuzprodukt, *2 Punkte*)

Zeige, dass das Keuzprodukt $\mathbf{v} \times \mathbf{w}$ orthogonal zu $\mathbf{v}$ ist.

Aufgabe 4 (Dreiecksungleichung, *2 Punkte*)

Zeige, dass die Dreiecksungleichung gilt: $\|A + B\| \leq \|A\| + \|B\|$
Tipp: Verwende die Schwarz'sche Ungleichung aus der Vorlesung und beginne mit dem Term $\|A+B\|^2$.
.