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

Universität Bremen
Fachbereich 3
Arbeitsgruppe CGVR
15. November 2024

Wintersemester 2024/25

Übungen zu Computergrafik - Blatt 5

Abgabe am 24.11.2024, 23:59 Uhr

In diesem Übungsblatt beschäftigen wir uns mit Transformationen. Lade Dir dazu das Transformation-Framework herunter.

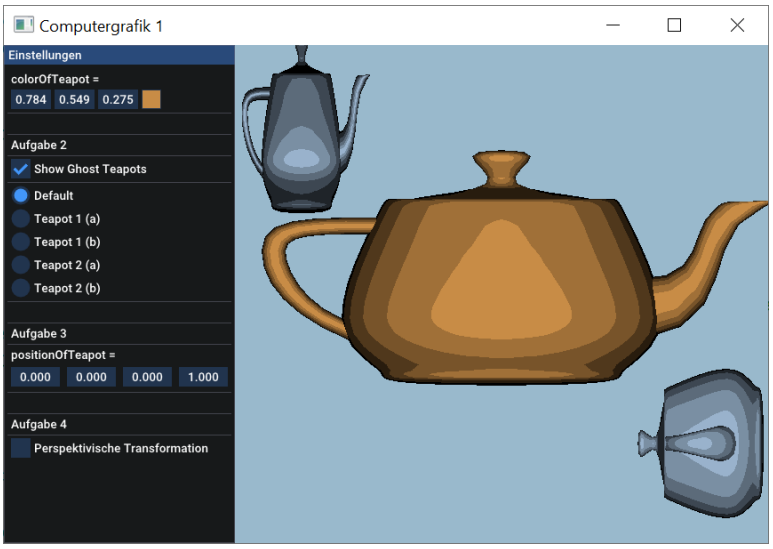


Abbildung 1: Das Fenster des Transformation-Framework

Aufgabe 1 (Matrixfunktionen erstellen (C++), *3 Punkte*)

Deine Aufgabe ist es, Funktionen zur Erzeugung von verschiedenen Transformationsmatrizen in der `Mat4f`-Klasse zu implementieren.

(a) Implementiere die *translation*-Funktion, sodass sie eine Verschiebung eines Punktes um die gegebenen `x`-, `y`- und `z`-Werte durchführt.

```
static Mat4f translation(const float x, const float y, const float z);
```

(b) Implementiere die *rotationX*- und *rotationY*-Funktionen, sodass sie Rotationen um die X- bzw. Y-Achse des gegebenen Winkels durchführen (mit Radianten als Eingabe).

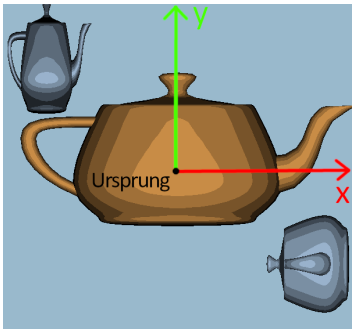
```
static Mat4f rotationX(const float angle);
static Mat4f rotationY(const float angle);
```

(c) Implementiere die *scale*-Funktion, sodass die zurückgegebene Matrix Skalierungen entsprechend der für die Achsen gegebenen Werte durchführt.

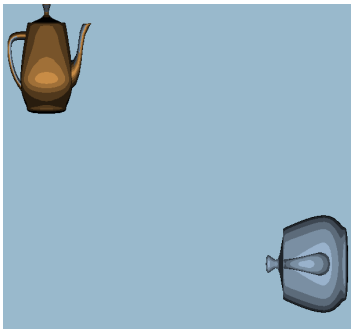
```
static Mat4f scale(const float x, const float y, const float z);
```

Aufgabe 2 (Verkettung von Transformationen (C++), *4 Punkte*)

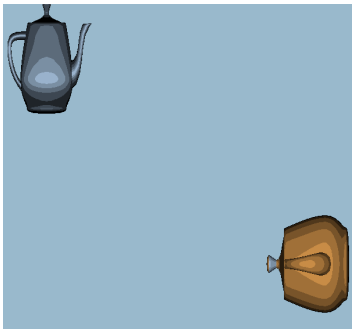
Wenn Du in der GUI die Option *'Zeige Ghost-Teapots'* aktivierst, werden dir zwei weitere Teapots gezeigt, frei im Raum liegen. Darüber hinaus bietet die GUI fünf verschiedene Transformations-Modi, nämlich *Default, Teapot 1 (a), Teapot 1 (b), Teapot 2 (a), Teapot 2 (b)*.



(a) Default



(b) Lösung: Teapot 1 (a,b)



(c) Lösung: Teapot 2 (a,b)

Im Programmcode in der *main.cpp* findest Du einen Bereich ab Zeile 216, der je nach gewählten Modus unterschiedliche Transformationsmatrizen der *model*-Matrix zuweißt. Nutze ausschließlich die Verkettung der von dir implementierten Transformationsmatrizen, sodass der gelbe Teapot möglichst genau in der Pose der blauen Ghost-Teapots liegt. Eine Verkettung kannst Du mithilfe des Multiplikationsoperators durchführen.

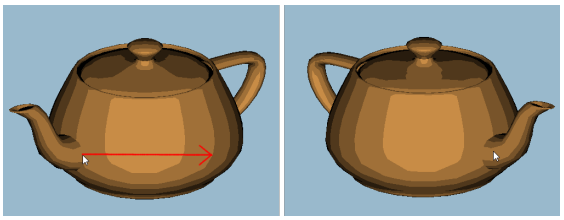
Beachte: Bei (a) und (b) soll für den selben Teapot die Pose mithilfer unterschiedlicher Verkettungen von Transformationen erreicht werden. Wenn Du bei Teapot 1 (a) zuerst eine X-Rotation und dann eine Translation verwendet hast, dann darfst Du dies bei Teapot 1 (b) nicht genauso machen. Aber Du dürftest zuerst eine Translation anwenden, und dann die X-Rotation.

Hinweise: Die Richtung der X- und Y-Achse sind in Abbildung (a) gezeigt, die Z-Achse ist unwichtig. Der Ursprung (0,0,0) liegt in der Mitte. An den Seiten des Bildes ist der Wert einer Achse jeweils -1 oder 1. Du kannst die Konstante `M_PI` *verwenden.*

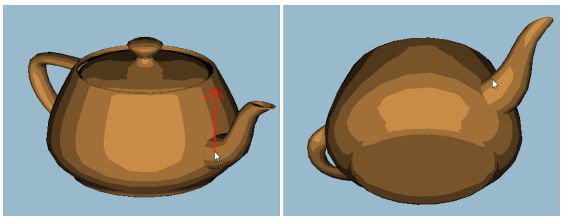
Aufgabe 3 (Objekt mit Maus steuern (C++), *2 Punkte*)

Ziel dieser Aufgabenstellung ist es, das sichtbare 3D-Objekt mithilfe der Maus verschieben und rotieren zu können.

a) Verändere die Default-Definition der `model`-Matrix (`main.cpp`, Zeile 214) so, dass sie eine Rotation um die X- und Y-Achse durchführt. Nutze dazu die von Dir implementierten Rotationsfunktionen und multizipliere die zurückgegebenen Matrizen bei Bedarf. Dabei sollen die Variablen `mouseMovementX` und `mouseMovementY` als Winkel verwendet werden. Erstelle die Rotation so, dass sie sich beim Bewegen der Maus wie folgt verhalten:



(a) Maus mit gedrückter Taste nach rechts ziehen



(b) Maus mit gedrückter Taste nach oben ziehen

Beachte, dass die Rotationsrichtung und Rotationsgeschwindigkeit dabei so gewählt werden soll, dass das Objekt gezogen wird (d.h. der berührte Punkt des Teapots soll sich auf kurze Strecken ganz grob mit der Maus mitbewegen, insbesondere in die selbe Richtung – siehe Tutorium).

Textaufgabe: Ein weiterer expliziter Bestandteil dieser Aufgabe ist es, dass Du die rotationMatrix-Variable im Quellcode kommentierst und dort beschreibst, welche Rotation zuerst angewendet wird und warum Du welche `mouseMovement`-Variable mit welcher Rotationsfunktion verwendet hast.

b) Ergänze zudem eine Verschiebung - also Translation - des Teapots in der Default-Definition der `model`-Matrix. Die Verschiebung soll genauso groß sein wie die `x`-, `y`- und `z`-Werte der vordefinierten Variable `positionOfTeapot`, die auch über die Gui verändert werden kann.

Beachte: Die Rotation soll zuerst angewendet werden und dann die Translation.

Aufgabe 4 (Alles eine Frage der Perspektive (C++), *1 Punkte*)

Bisher sieht der Teapot irgendwie ungewohnt aus, irgendwie merkwürdig verzerrt. Das liegt daran, dass unser Auge es gewohnt ist, 3D-Objekte perspektivisch Wahrzunehmen (vgl. Wikipedia).

Damit 3D-Objekte perspektivisch gerendert werden, müssen wir also eine perspektivische Transformation anwenden, die durch eine `4x4` - Matrix beschrieben werden kann. In der `Mat4f`-Klasse ist bereits eine statische Funktion `perspectiveTransformation` vorgegeben, die Du verwenden kannst. Was genau in dieser Funktion passiert, erfährst Du später in der Vorlesung im Kapitel „Projektion und Perspektive“.

Der einzige Programmieranteil dieser Aufgabe besteht darin, der `projection`-Variable innerhalb der nachfolgenden if-Bedingung der `main.cpp` die zurückgegebene Matrix der o.g. Funktion zuzuweisen:

```
236         // Initially, we simply have an identity matrix for the projection, so that it
237         // does not affect the position when it is multiplied with other matrices:
238         Mat4f projection;
239 
240         if(usePerspectiveTransformation){
241            // *TODO* (Exercise 4): Use projection matrix and describe the visible results.
242         }
```

Die `projection`-Matrix wird weiter unten im Programmcode an den Vertex-Shader übergeben – siehe Dir an, wie diese im Vertex-Shader verwendet wird. Die Variable `usePerspectiveTransformation` kann über die Checkbox in der Gui verändert werden, sodass die perspektivische Transformation per Knopfdruck an- und abgeschaltet werden kann.

Textaufgabe: Ein weiterer Bestandteil dieser Aufgabenstellung ist es, kurz als C++ Kommentar oberhalb der Zuweisung zu beschreiben, wie sich eine Änderung des z-Wertes der Position des Teapots auswirkt, wenn die perspektivische Transformation aktiviert ist und wie, wenn diese deaktiviert ist.