Grundlagen der Computergrafik, Blatt 7

Lukas Baur, 3131138
Felix Bühler, 2973410
Marco Hildenbrand, 3137242

12. Dezember 2017

Aufgabe 1

1) Translation und Skalierung

$$M_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$

2) Translation und Rotation

$$M_2 = \begin{bmatrix} cos(-30) & -sin(-30) & 0 \\ sin(-30) & cos(-30) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{3} & \frac{1}{2} & 2\sqrt{3} \\ -\frac{1}{2} & \frac{\sqrt{2}}{3} & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

3) Scherung

$$M_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4) Rotation

$$M_4 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} cos(-60) & -sin(-60) & 0 \\ sin(-60) & cos(-60) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} & 1-\sqrt{3} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & 1+\sqrt{3} \\ 0 & 0 & 1 \end{bmatrix}$$

Aufgabe 5

a)

Die Translation und Rotation werden nacheinander ausgeführt, wobei die Rotation zuerst ausgeführt wird da Transformationen bei Hintereinander Ausführung von rechts nach Links ausgeführt werden.

b)

Der Z-Buffer wird zur Speicherung der Z-Werte jedes einzelnen Pixels verwendet. Dazu wird zu jedem Pixel abgespeichert, in welcher Tiefe (von der Kamera der Szene ausgemessen) sich das nächste Objekt (bzw. deren Oberfläche) befindet. Die Idee ist, dass man so auf einfache Art und Weise zwei Objekte mit dem Z-Buffer vereinigen kann, da nur derjenige Teil (pro Pixel) zu sehen ist, der sich näher an der Kamera befindet. Problematik beim Umgang mit parallelen Berechnungen: Beim Rendern auf der GPU braucht der Z-Buffer einen großen Teil des Speichers sowie der Datenübertragungsrate.

c)

Beim Clipping gibt es mehrere Ansätze mit welchem dieses implementiert werden kann. Die erste Möglichkeit ist Clipping on-the-fly was so viel bedeutet wie, dass während der Rasterisierung eine Laufvariable mitläuft welche dafür sorgt, dass nichts außerhalb des Bildschirms gezeichnet wird. Dann gibt es noch das analytische Clipping bei dem analysiert wird wo es zu Überschneidungen kommt mit dem Bildschirmrand und dann mit verschiedenen Methoden behoben wird.

Wenn man Clipping bei Wireframe anwendet kann es zu Mehrdeutigkeiten kommen, das heißt da Clipping alles was das Bild verlässt abschneidet könnten mehrere Gitter übereinander liegen. Dasselbe gilt für die ausgefüllten Dreiecke dort kann es auftreten das beim abschneiden nun 1 Dreieck vollkommen von einem anderen verdeckt wird.