Aufgabe 1

Zu Abbildung 1:

Verschieben des Ursprungs zum Punkt p mit Translationsmatrix:

Skalieren der Achsen auf die Hälfte mit Skalierungsmatrix:

Kombination der beiden Matrizen ergibt:

Hat der der Kreis seinen Mittelpunkt im Ursprung, dann wird er derart verschoben, dass sein Mittelpunkt der Punkt p ist, der Radius des Kreises wird halbiert. Ist der Mittelpunkt p, dann wird er um p verschoben.

Zu Abbildung 2:

Translationsmatrix:

Skalierungsmatrix:

Rotationsmatrix:

Daraus ergibt sich diese Transformationsmatrix:

Ein Kreis mit dem Mittelpunkt im Ursprung wird um p verschoben und entlang seiner horizontalen Achse halbiert, es entsteht eine Ellipse, welche außerdem um -45° rotiert wird. Ist der Mittelpunkt p, dann wird der Kreis um p verschoben.

Aufgabe 2

TODO

Aufgabe 3

Eine Translationsmatrix hat stets die Determinante 1 und ist somit invertierbar. Eine Rotationsmatrix hat ebenfalls stets die Determinante 1 und ist damit auch invertierbar. Die Determinante von M ist somit ebenfalls 1, damit ist auch die Gesamtabbildung invertierbar.

Für eine Translationsmatrix gilt, dass ihre Inverse der Translationsmatrix für den entgegengerichteten Translationsvektor entspricht:

Für eine Rotationsmatrix gilt, dass sie orthogonal ist, deshalb lässt sich die Inverse durch das Transponieren ermitteln.

Aufgabe 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation 1 \ Operation 2 | Rotation | Isotrope Skalierung | Anisotrope Skalierung | Translation |
| Rotation | Entfällt | (nicht) kommutativ | (nicht) kommutativ | (nicht) kommutativ |
| Isotrope Skalierung | (nicht) kommutativ | Entfällt | Entfällt | (nicht) kommutativ |
| Anisotrope Skalierung | (nicht) kommutativ | Entfällt | Entfällt | (nicht) kommutativ |
| Translation | (nicht) kommutativ | (nicht) kommutativ | (nicht) kommutativ | Entfällt |

Aufgabe 5

Bei der doppelten Pufferung sind, das Bild, welches angezeigt wird, und das Bild, welches momentan berechnet wird, zwei verschiedene Bilder. Wenn das zu berechnende Bild bereit ist werden die Bilder vertauscht.

Ohne doppelte Pufferung würde für beide Bilder der gleiche Speicherbereich verwendet werden, das alte Bild, welches angezeigt wird, kann durch das neu berechnete Bild teilweise überlagert werden, dies wird als Tearing bezeichnet und vom Benutzer meist als sehr störend wahrgenommen, weshalb man Techniken wie die doppelte Pufferung einsetzt.

Würde in der Programmieraufgabe keine doppelte Pufferung eingesetzt werden, so würde