

Computer Graphics

Zwischenprüfung HS 14

Teile: Grafik + Projektive Geometrie

Thomas Koller

Name: Schuler Vorname: Stefan


(Bitte mit Druckbuchstaben schreiben)

Unterschrift: Stefan Schuler

Rahmenbedingungen:

1. Prüfungszeit: Max. 120 Minuten
2. Schreiben Sie Ihren Namen und Vornamen mit Druckbuchstaben oben auf dieses Blatt. Mit der Unterschrift bezeugen Sie, dass Sie diesen Prüfungsteil persönlich und nur mit erlaubten Hilfsmitteln bearbeitet haben. Blätter ohne diese Angaben werden nicht bewertet.
3. Es handelt sich um eine schriftliche Prüfung mit Benützung von Unterlagen auf Papier oder in elektronischer Form auf dem Computer. Das Internet darf nicht benutzt werden.
4. Sollte eine Aufgabenstellung Unklarheiten aufweisen, können Sie sich an eine Aufsichtsperson wenden.
5. Schreiben Sie möglichst verständlich und gut leserlich. Missverständliche Lösungen werden nicht berücksichtigt.
6. Benutzen Sie den Freiraum unter den Aufgaben für Ihre Lösung.

Für die Korrektur (nicht ausfüllen!)

C1	C2	C3	C4	C5	C6	P1	P2	P3	Punkte	Visum
8	2	3	7	4	4	4	0	0	32	

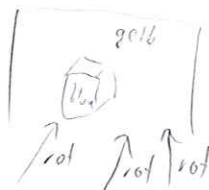
Aufgabe 1: Farbe und Farbsysteme (14P)

a) Vervollständigen sie die folgende Tabelle (6P):

Name der Farbe	RGB	CMY	CMYK	HSV
blau	0, 0, 255	1, 1, 0	0,33; 0,33; 0; 0	240°, 100%, 100%
grau	157, 157, 157	0,38, 0,38, 0,38	(0,0,0,0.5)	0°, 0%, 62%
schwarz	0, 0, 0	(1,1,1)	100%, 100%, 100%, 0%	180°, 100%, 100%

$$\begin{matrix} C \\ M \\ Y \end{matrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \rightarrow \begin{matrix} C \\ M \\ Y \\ K \end{matrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0,33 \\ 0,33 \\ 0,33 \end{pmatrix} \rightarrow \begin{pmatrix} 0,67 \\ 0,67 \\ 0,67 \end{pmatrix}$$

b) Sie platzieren einen blauen Würfel auf vor einem gelben Hintergrund und beleuchten beides mit einer roten Lichtquelle. In welcher Farbe erscheint der Würfel und in welcher Farbe erscheint der Hintergrund (4P)?

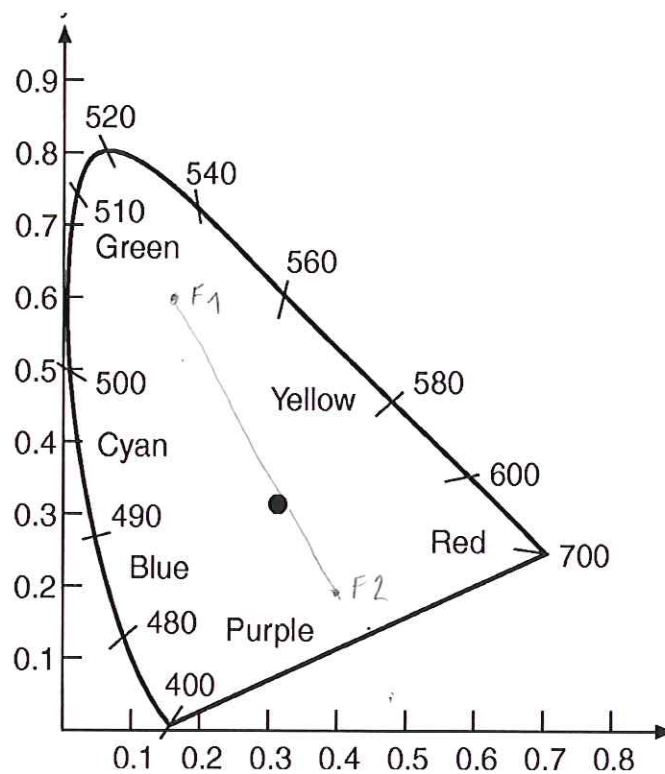


Hintergrund: orange
Würfel: grün



blau reflektiert kein rot → schwarz
gelb reflektiert rot → rot

Der gelbe Hintergrund wirkt Rot, da er Rot reflektiert (orange)



Die Farben F1 und F2 seien in der CIE-Normfarbtafel durch die Koordinaten $F1 = (0.2, 0.6)$ und $F2 = (0.4, 0.2)$ gegeben.

c) Welche weiteren Farben können durch diese 2 Farben gemischt werden (2P) ?

alle Farben auf der Verbindungslinie F_1 und F_2

2

d) Was sind die (ungefähren) Koordinaten der Komplementärfarbe von F1 (2P) ?

0.4, 0.1

F_2

2

F_1 Komp \rightarrow durch weiss \rightarrow ca. $(0.4, 0.1)$

Aufgabe 2: Dithering (8P)

a) Weshalb wird Dithering verwendet? (2P)

Weil Drucker im allgemeinen nur schwarz und weiss darstellen können, aber keine Graustufen

Mittels Dithering werden Farben mit der Auflösung simuliert.

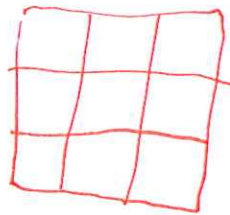
z.B. Haben Drucker eine grössere Auflösung, können aber wenige Farbstufen darstellen als Bildschirme.

b) Sie möchten das folgende Bild auf einem schwarz-Weiss Drucker darstellen. Verwenden Sie eine geeignete 3x3 Dithermatrix und berechnen Sie die gezeichneten Pixel, das ursprüngliche Bild besitzt 256 Intensitätsstufen von 0-255. (6P)

0..255

10	120	255	231
1	148	191	178

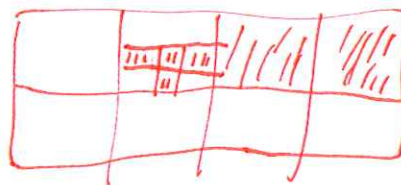
3x3 Matrizen



→ 10 Stufen

Quantisiertes Bild: 0..9

0	4	9	9
0	5	7	6



$$\frac{x}{25.6}$$

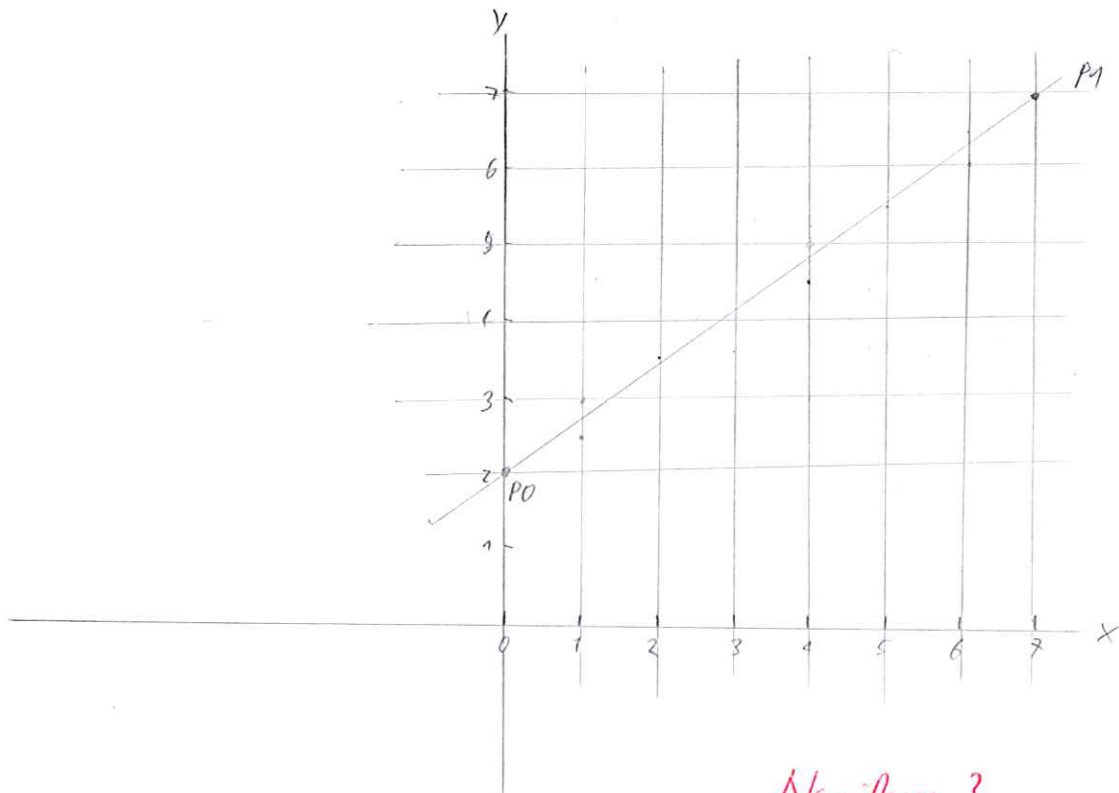
und nicht 25.5, da $\frac{255}{25.5} = 10$ und 10 ist nicht im Range

$$0..25 \rightarrow 0$$

$$26..51 \rightarrow 1$$

Aufgabe 3: Mittelpunktschema (8P)

- a) Sie möchten eine Linie mit dem Mittelpunktschema berechnen die vom Punkt $P_0 = (0, 2)$ zum Punkt $P_1 = (7, 7)$ führt. Welche Pixel werden gezeichnet? Wie ist der Wert der Entscheidungsvariablen d bei jedem Pixel? (8P)



Algorithmus?

3

$$F(x, y) = a \cdot x + b \cdot y + c = 0$$

$$a = \Delta y \cdot (y_1 - y_0) \quad b = -\Delta x \quad c = \Delta x \cdot y_0$$

$$d_{i+1} = F(M) = F\left(x_i + 1, y_i + \frac{1}{2}\right) = \Delta y \left(x_i + 1\right) - \Delta x \left(y_i + \frac{1}{2}\right) + \Delta x \cdot y_0$$

$$x_0 + 1, y_0 + \frac{1}{2} = (1, 2.5) \rightarrow (1, 3)$$

$$x_1 + 1, y_1 + \frac{1}{2} = (2, 3.5) \rightarrow (2, 4)$$

$$x_2 + 1, y_2 + \frac{1}{2} = (3, 4.5) \rightarrow (3, 5)$$

$$x_3 + 1, y_3 + \frac{1}{2} = (4, 5.5) \rightarrow (4, 6)$$

$$x_4 + 1, y_4 + \frac{1}{2} = (5, 6.5) \rightarrow (5, 7)$$

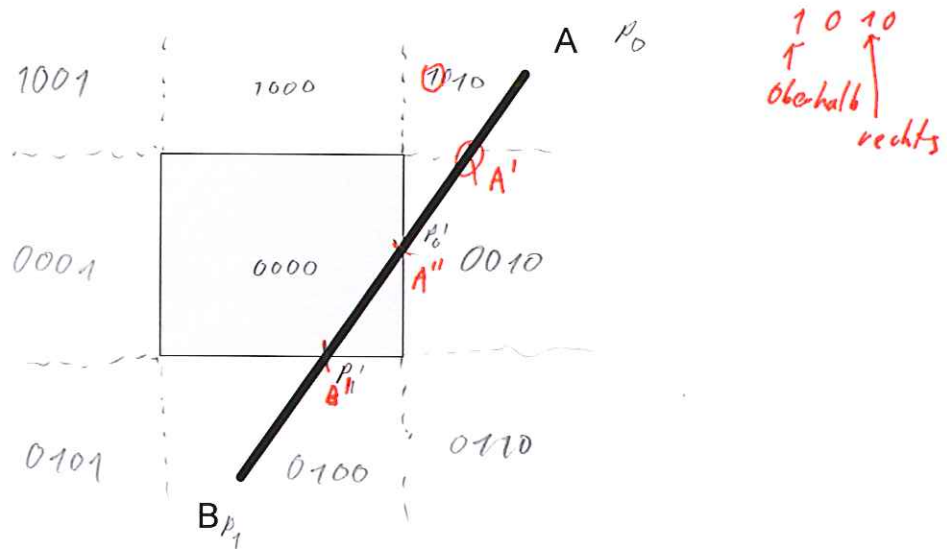
$$x_5 + 1, y_5 + \frac{1}{2} = (6, 7.5) \rightarrow (6, 8)$$

$$x_6 + 1, y_6 + \frac{1}{2} = (7, 8.5) \rightarrow (7, 9)$$

(✓)

Aufgabe 4: Kappung (8)

- a) Die Line von A nach B soll auf den Bereich innerhalb des Rechtecks gekappt werden. Führen Sie die einzelnen Schritte nach der Methode von Cohen-Sutherland aus (8P).



1. Schritt: $code(P_0) = 1010$
 $code(P_1) = 0100$
 AND 0000 → schneiden

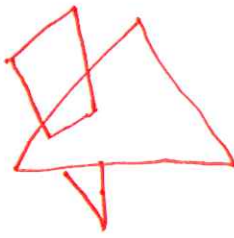
2. Schritt
 Schnittpunkt mit Kante 0010 (rechts)
 → neue Gerade $P_0' P_1$

$code(P_0') = 0000$
 $code(P_1) = 0100$
 AND 0000 ✓

3. Schritt
 Schnittpunkt mit Kante 0100 (unten)
 → P_1'
 Gerade $P_0' \rightarrow P_1'$ ✓

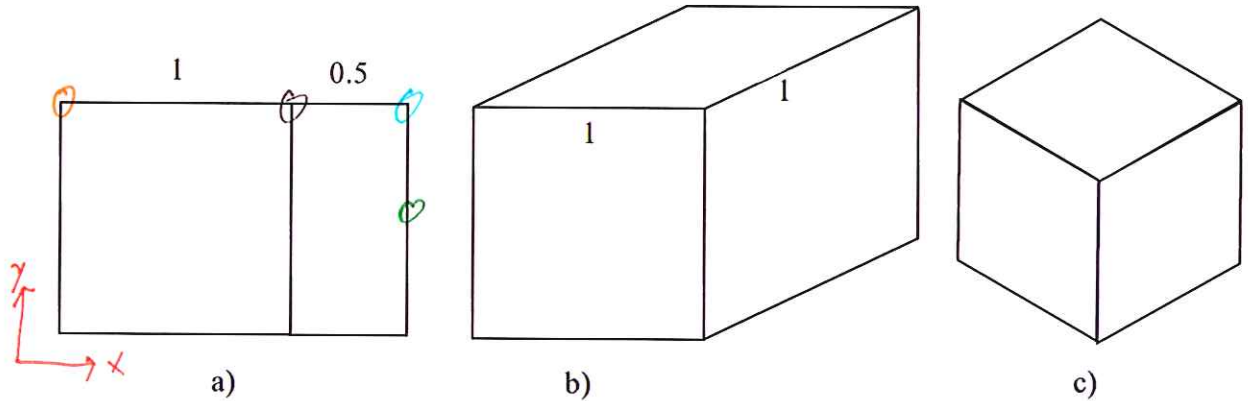
7

c)



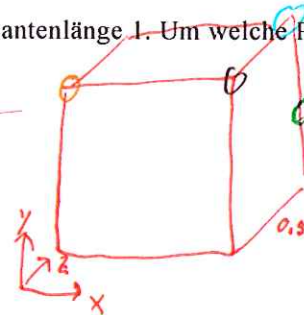
aufteilen

Aufgabe 5: Projektionen und 3D Darstellung (12P)



- a) Die Abbildungen zeigen einen Würfel mit Kantenlänge 1. Um welche Projektionen handelt es sich (6P)?

- a) ~~Perspektivisch~~ Kabinett
b) Kavalier
c) isometrisch



4

- b) Wie lautet die 4x4 Projektionsmatrix für die Projektion in Abbildung a) (2P)?

$$\begin{aligned} x &\rightarrow x + 0,5z \\ y &\rightarrow y \\ z &\rightarrow 0 \end{aligned} \quad \begin{pmatrix} 1 & 0 & 0,5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix}$$

- c) Wie werden sich durchdringende Dreiecke im Tiefensortierungsverfahren (Painter's Algorithm) und im z-Buffer Algorithmus behandelt (4P)?

Tiefensortieren

- Painter's: 1. Sortiere Polygone nach minimaler z-Koordinate
2. Falls sich Polygone in z überlagern, betrachte zusätzliche Bedingungen
3. Falls keine der Bedingungen erfüllt ist, vertausche die Polygone und überprüfe die Bedingung nochmals

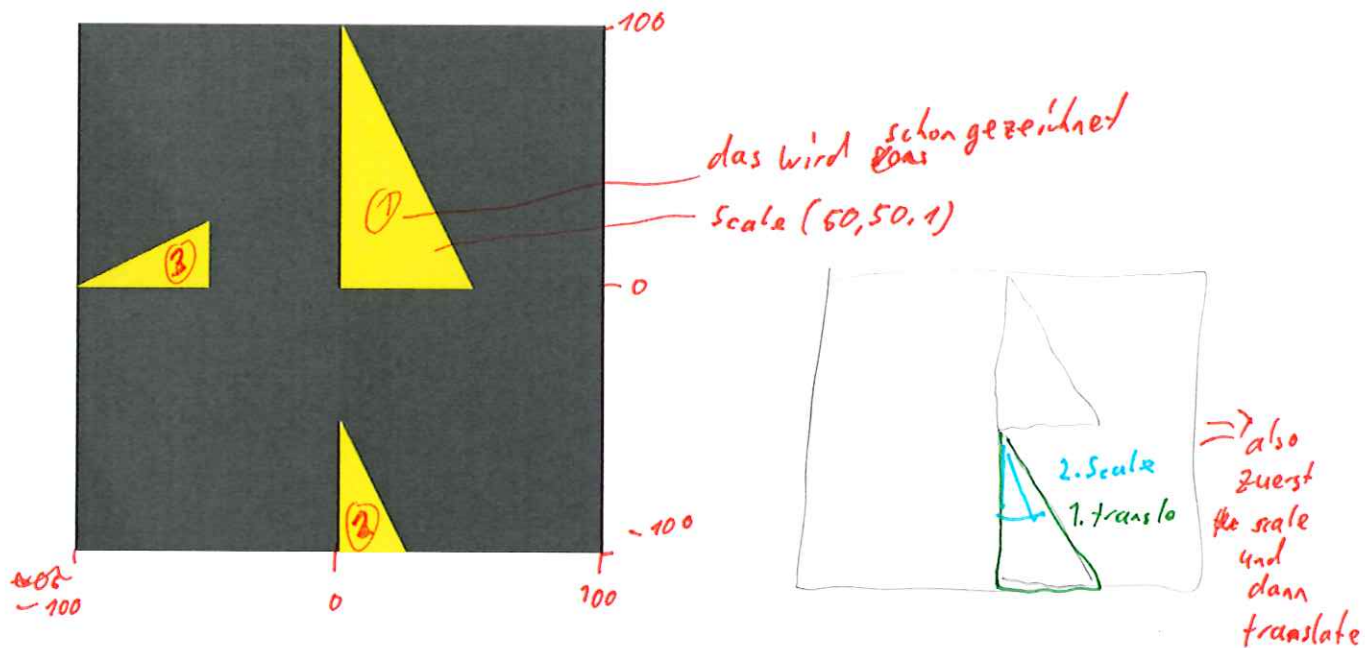
z-Buffer: Pro Pixel wird außer der Farbe auch die Tiefe (z-Wert) gespeichert.
Beim Zeichnen jedes Pixels wird geprüft ob der neue Pixel näher liegt
Wie man z-Buffer auf diese Situation anwendet

Dreieck 2: - z.B. das 1. Dreieck nach 150 nach unten. Das Dreieck geht dann über die Linie und man sieht nur den kleinen Teil des Dreieck

- mat4.scale(matrix, orthoMatrix, [25, 25, 1]);
mat4.translate(matrix, matrix, [0, -100, 0]);
gl.uniformMatrix4fv(uModelViewMatrixId, false, matrix);

Aufgabe 6: WebGL (15P)

Das WebGL Programm auf der folgenden Seite sollte dieses Bild berechnen. (Die Funktionen initShaders und createContext sind aus Platzgründen nicht aufgeführt, sie können jedoch davon ausgehen, dass sie richtig funktionieren.)



- a) Wieso erscheinen die Dreiecke gelb und nicht weiss? Welcher Befehl auf welcher Zeile ist dafür verantwortlich? (3P)

19

gl.FragColor = vec4(1.0, 1.0, 0.0, 1.0);



3

- b) Ergänzen Sie das Programm ab Zeile 69, sodass alle drei Dreiecke richtig gezeichnet werden. (12P)

// rechts oben Dreieck

mat4.translate(matrix, orthoMatrix, [?]);

// unten Dreieck

mat4.translate(matrix, orthoMatrix, [?]);

// links Dreieck

mat4.translate(matrix, orthoMatrix, [?]);

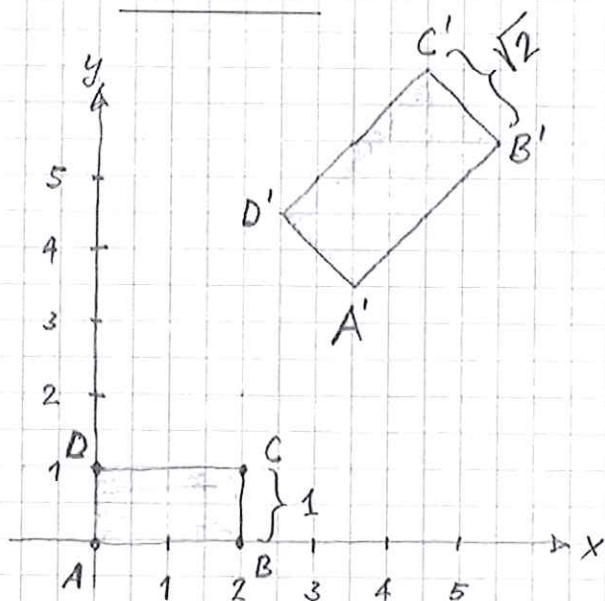
7


```

1 var canvas;
2 var gl;
3 var shaderProgram;
4 var aVertexPositionId;
5 var uModelViewMatrixId;
6 var bufferObject;
7
8 var VSHADER_SOURCE =
9 "attribute vec2 aVertexPosition;" +
10 "uniform mat4 uModelViewMatrix;" +
11 "void main() {" +
12 "    vec4 position = vec4(aVertexPosition, 0.0, 1.0);" +
13 "    gl_Position = uModelViewMatrix * position;" +
14 "}";
15
16 var FSHADER_SOURCE =
17 "precision mediump float;" +
18 "void main() {" +
19 "    gl_FragColor = vec4(1.0, 1.0, 0.0, 1.0);" +
20 "}";
21
22 function startup() {
23     canvas = document.getElementById("gameCanvas");
24     gl = createGLContext(canvas);
25     gl.clearColor(0.2, 0.2, 0.2, 1.0);
26     initShaders();
27     setupAttributes();
28     defineObject();
29     draw();
30 }
31
32 function initShaders() {
33     //...
34 }
35
36 function createGLContext(canvas) {
37     //...
38 }
39
40 function setupAttributes() {
41     aVertexPositionId = gl.getAttribLocation(shaderProgram, "aVertexPosition");
42     uModelViewMatrixId = gl.getUniformLocation(shaderProgram, "uModelViewMatrix");
43 }
44
45 function defineObject() {
46     var vertices = [
47         0,0,
48         1,0,
49         0,2,
50     ];
51     bufferObject = gl.createBuffer();
52     gl.bindBuffer(gl.ARRAY_BUFFER, bufferObject);
53     gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);
54 }
55 function draw() {
56     gl.clear(gl.COLOR_BUFFER_BIT);
57     gl.vertexAttribPointer(aVertexPositionId, 2, gl.FLOAT, false, 0, 0);
58     gl.enableVertexAttribArray(aVertexPositionId);
59
60     var matrix = mat4.create();
61     var orthoMatrix = mat4.create();
62     mat4.ortho(orthoMatrix, -100, 100, -100, 100, 0.0, 1.0);
63     mat4.scale(matrix, orthoMatrix, [50, 50, 1]);
64
65     gl.uniformMatrix4fv(uModelViewMatrixId, false, matrix);
66     gl.drawArrays(gl.TRIANGLE_STRIP, 0, 3);
67
68     // TODO:
69
70 }
71 }
72

```

1. 2D-Fall



Das Rechteck ABCD wird durch folgende Transform. in das Rechteck A'B'C'D' überführt.

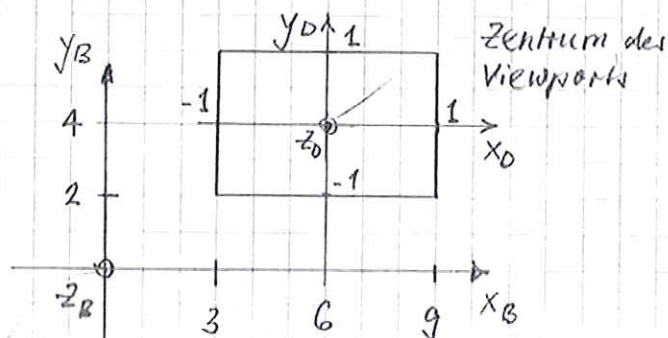
- (a) Rotation
- (b) Translation
- (c) Skalierung

Wie lauten die einzelnen Transformationen und in welcher Reihenfolge müssen sie ausgeführt werden. Wie lautet die Matrix für die gesamte Transformation? Alles in homogenen Koord.!

2. Perspektive

Bilde das Rechteck A'B'C'D' auf die Ebene $z = 5$ ab unter der Annahme, dass das Projektionszentrum in $(0, 0, 10)$ liegt. Wie lautet die entsprechende Matrix in hom. Koord.

3. Viewporttransformation



Ursprung des Bildschirmkoordinatensystems

Geben Sie die Vorschrift an mit welcher die normierten Gerätekoord.

x_D, y_D, z_D in die Bildschirmkoord. x_B, y_B, z_B umgerechnet werden können.

1.

a) Rotation

$$R = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

$$\varphi = 45^\circ$$

$$\begin{pmatrix} \cos(45) & -\sin(45) \\ \sin(45) & \cos(45) \end{pmatrix} =$$

$$= \begin{pmatrix} 0,7 & -0,85 \\ 0,85 & 0,7 \end{pmatrix}$$

$$= \begin{pmatrix} 0,7 & -0,85 \\ 0,85 & 0,7 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1' & 0' & c' & 0' \end{pmatrix}$$

b) Translation

$$T = \begin{pmatrix} 0 & 0 & T_x \\ 0 & 0 & T_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_x = ? \quad T_y = ?$$

$$\sin 45^\circ = \cos 45^\circ$$

! 0

c) Skalierung

$$S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$s_x = ?$$

$$s_y = ?$$

1. Rotation

2. Skalierung

3. Translation

$$((T \cdot S) \cdot R) \cdot v$$

4

0

0