Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Technik & Architektur

Computer Graphics

Zwischenprüfung HS 14

Teile: Grafik + Projektive Geometrie

Thomas Koller

Name: Mosberger Vorname: Dominik

(Bitte mit Druckbuchstaben schreiben)

Unterschrift: D. Mosberger

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
6.5	2	8	0	4	3	4	1	1	29.5	[illegible]

Hochschule Luzern
Technik & Architektur

Aufgabe 1: Farbe und Farbsysteme (14P)

6.5

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

Name der Farbe	RGB	CMY	CMYK	HSV
blau	0,0,1	1,1,0	1,1,0,0	180,1,1
grau	0.5,0.5,0.5	0.5,0.5,0.5	(0,0,0,0.5)	x,0.5,0.5
cyan	0,1,1	(1,0,0)	1,0,0,0	240,1,1

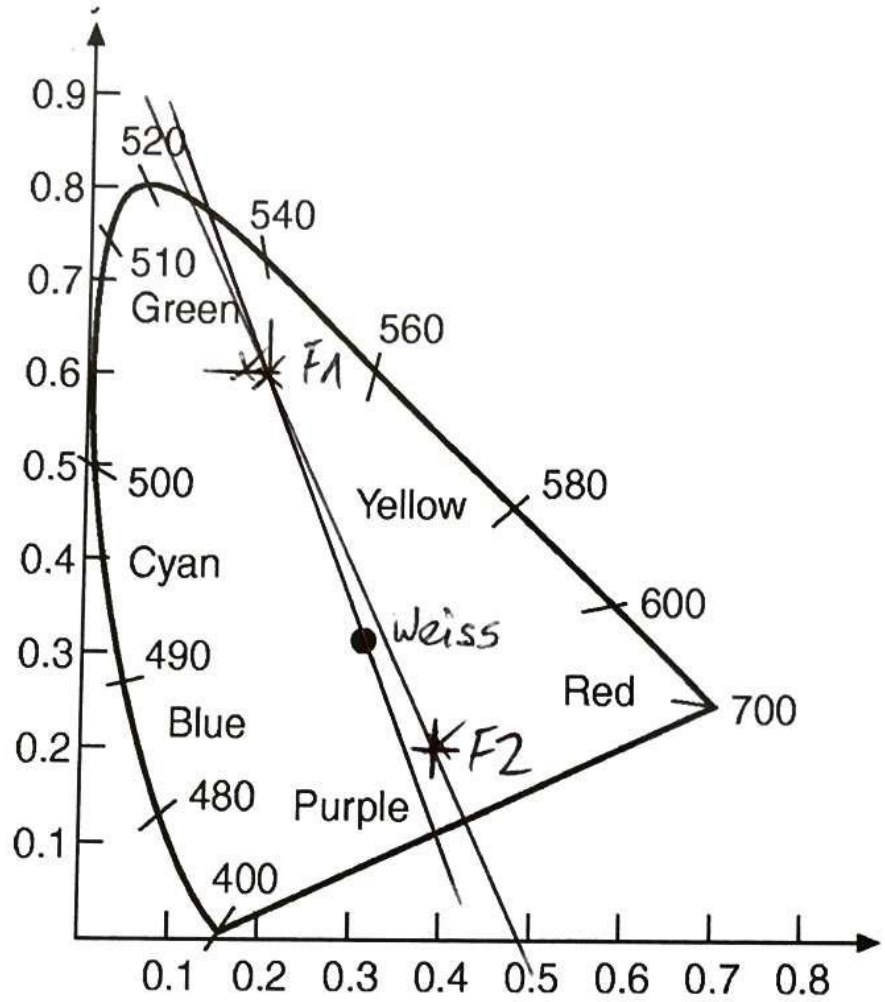
4.5

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)?

Würfel: blau + rot = ~~purpur~~ purple

Hintergrund: gelb + rot = orange

0



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 auf der eingezeichneten Verbindungslinie von F_1 und F_2

2

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

Alle auf der Verbindungslinie zwischen F_1 und Weiss.

0

Aufgabe 2: **Dithering (8P)**

a) Weshalb wird Dithering verwendet? (2P)

n-mal

Dadurch entsteht ein ~~doppelt~~ so grosses Bild

-> Tinte / Toner / Druckerschwärze sparen

1/2

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)

10	120	255	231
1	148	191	178

$$\begin{bmatrix} 6 & 7 & 8 \\ 1 & 0 & 2 \\ 4 & 3 & 5 \end{bmatrix}$$

?

1/2

Aufgabe 3: Mittelpunktschema (8P)

a) Sie möchten eine Linie mit dem Mittelpunktschema berechnen die vom Punkt P0 = (0, 2) zum Punkt P1 = (7, 7) führt. Welche Pixel werden gezeichnet? Wie ist der Wert der Entscheidungsvariablen d bei jedem Pixel? (8P)

~~$F(0,2) = 0 \cdot 0 + 6 \cdot 7 + 9$~~

$P_0 \times P_1 = \begin{matrix} 0 & 7 \\ 2 & 7 \end{matrix} = 0 \cdot 7 + 2 \cdot 7 = 14$

$Dx = 7 - 0 = 7$

$Dy = 7 - 2 = 5$

$DE = 2 \cdot Dy = 10$

$DNE = 2 \cdot (Dy - Dx) = -4$

$d = 2 \cdot Dy - Dx = 3$

$y = y_0 = 2$ ✓

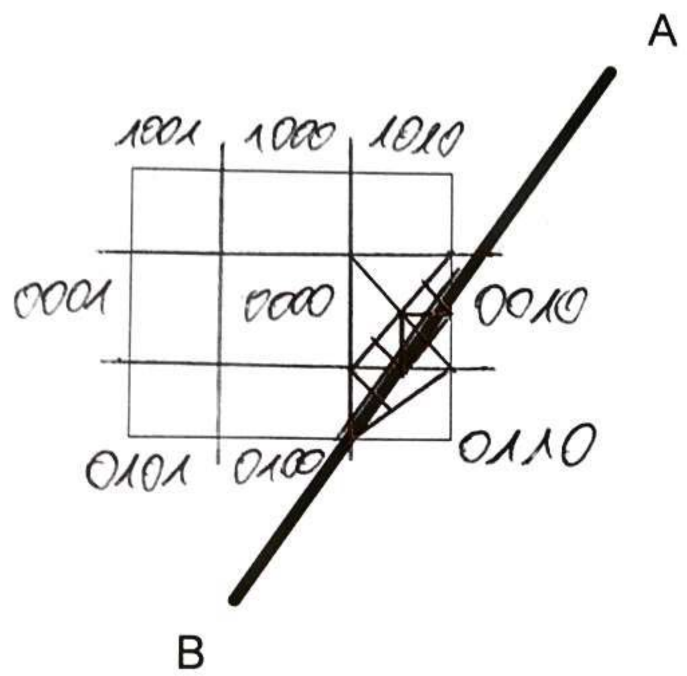
Pixel:	
0,2	x = 1, d = −1, y = 3
1,3	x = 2, d = 9, y = 3
2,3	x = 3, d = 5, y = 4
3,4	x = 4, d = 1, y = 5
4,5	x = 5, d = −3, y = 6
5,6	x = 6, d = 7, y = 6
6,6	x = 7, d = 3, y = 7
7,7 ✓	

8

Gezeichnete Punkte

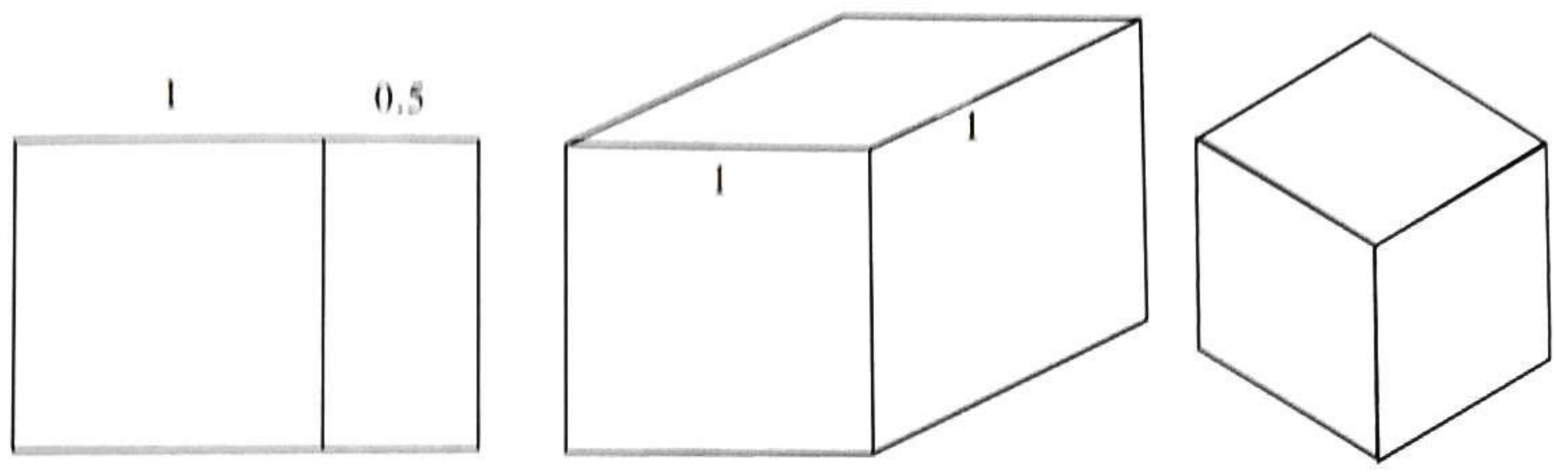
Aufgabe 4: **Kappung (8)** = Clipping !

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).



Gezeichnet werden Bereiche, die nicht von $\overline{AB}$ berührt wurden

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



a) b) c)

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

a) orthographisch

b) kavaliersprojektion

c) isometrisch

4

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

$$\begin{bmatrix} \frac{2}{x_R - x_L} & 0 & 0 & -\frac{x_R + x_L}{x_R - x_L} \\ 0 & \frac{2}{y_O - y_U} & 0 & -\frac{y_O + y_U}{y_O - y_U} \\ 0 & 0 & -\frac{2}{F-N} & -\frac{F+N}{F-N} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

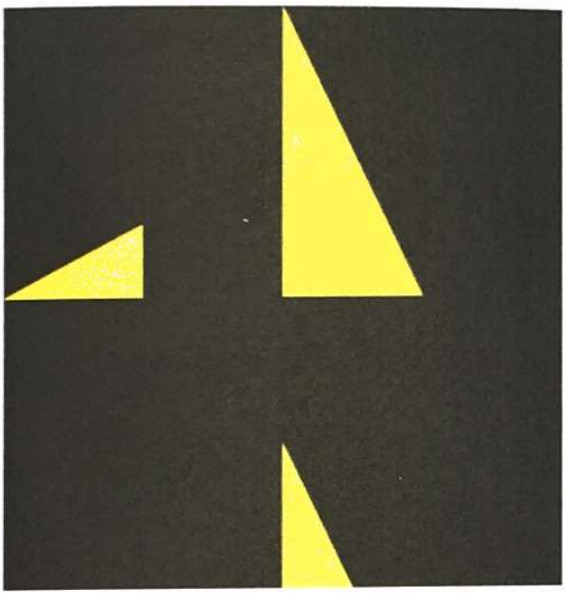
c) Wie werden sich durchdringende Dreiecke im Tiefensortiertungsverfahren (Painter's Algorithm) und im z-Buffer Algorithmus behandelt (4P)?

Painter: zuletzt gezeichnete Polygone überdecken frühere

z-Buffer: Beim Zeichnen jedes Pixels wird geprüft, ob der neue Pixel näher (z-Wert) liegt

Aufgabe 6: WebGL (15P)

Das WebGL Programm auf der folgenden Seite sollte dieses Bild berechnen. (Die Funktionen initShaders und createGLContext 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, 1, 0, 1);

durch 1 ersetzen = weiss

3

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

Die ~~Vertices~~ Indices müssen noch definiert werden.

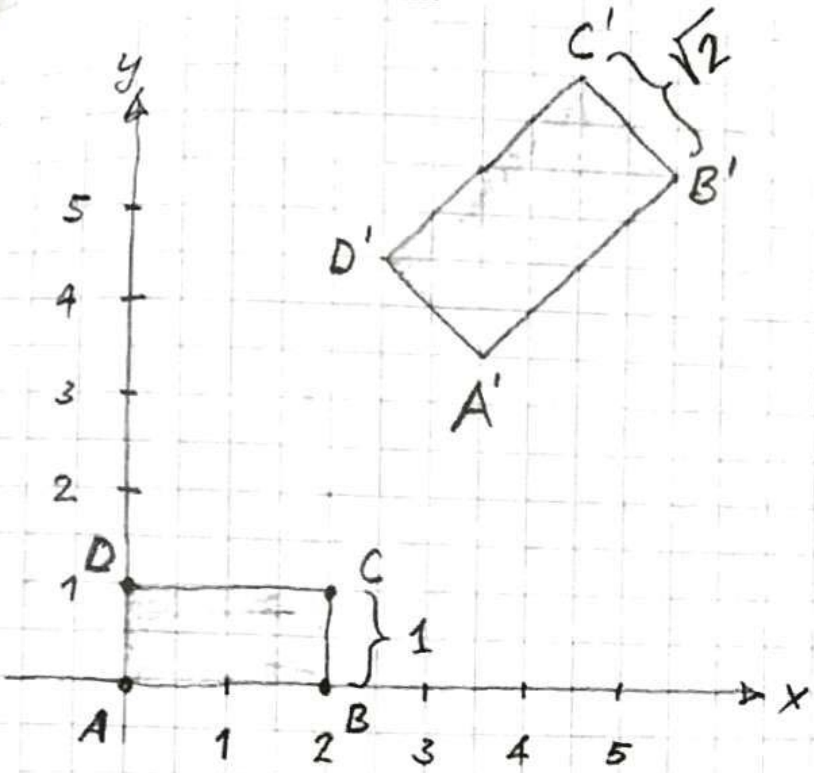
gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indices);

gl.drawElements(GL.TRIANGLE, 3, GL.FLOAT, 0);

```
var canvas;
var gl;
var shaderProgram;
var aVertexPositionId;
var uModelViewMatrixId;
var bufferObject;

var VSHADER_SOURCE =
"attribute vec2 aVertexPosition;" +
"uniform mat4 uModelViewMatrix;" +
"void main() {" +
"    vec4 position = vec4(aVertexPosition, 0.0, 1.0);" +
"    gl_Position = uModelViewMatrix * position;" +
"}";

var FSHADER_SOURCE =
"precision mediump float;"+
"void main() {"+
"      gl_FragColor = vec4(1.0, 1.0, 0.0, 1.0);"+
"}";

function startup() {
    canvas = document.getElementById("gameCanvas");
    gl = createGLContext(canvas);
    gl.clearColor(0.2, 0.2, 0.2, 1.0);
    initShaders();
    setupAttributes();
    defineObject();
    draw();
}

function initShaders() {
    //...
}

function createGLContext(canvas) {
    //...
}

function setupAttributes() {
    aVertexPositionId = gl.getAttribLocation(shaderProgram, "aVertexPosition");
    uModelViewMatrixId = gl.getUniformLocation(shaderProgram, "uModelViewMatrix");
}

function defineObject() {
    var vertices = [
        0,0,
        1,0,
        0,2,
    ];
    bufferObject = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferObject);
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);
}
function draw() {
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.vertexAttribPointer(aVertexPositionId, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(aVertexPositionId);

    var matrix = mat4.create();
    var orthoMatrix = mat4.create();
    mat4.ortho(orthoMatrix, -100, 100, -100, 100, 0.0, 1.0);
    mat4.scale(matrix, orthoMatrix, [50, 50, 1]);

    gl.uniformMatrix4fv(uModelViewMatrixId, false, matrix);
    gl.drawArrays(gl.TRIANGLE_STRIP,0,3);

    // TODO:


}
```

Testprüfung CG (Mathe-Teil) SA, 8.11.2014 JFB

1. 2D-Fall



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

(a) Rotation

(b) Translation

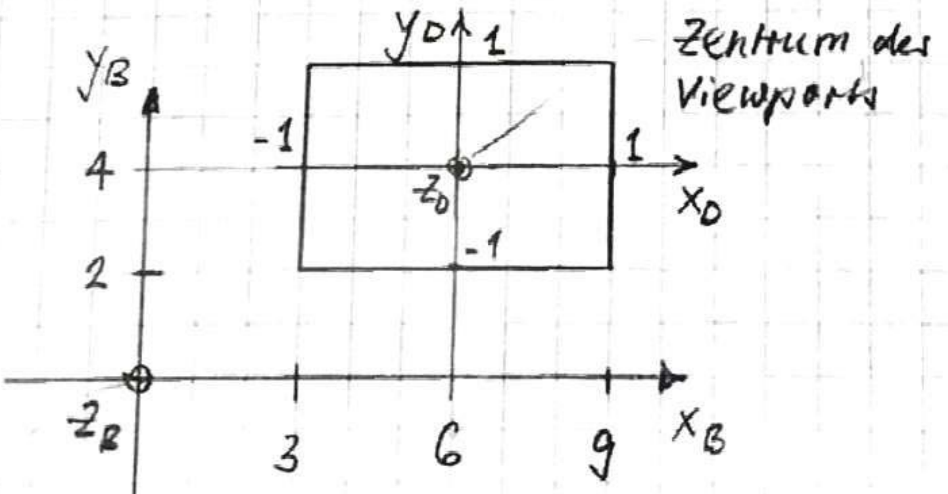
(c) Skalierung

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.

CG Zwischenprüfung Mathe

Dominik Mosberger

1. a) Rotation $= \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = R$ $\quad \to \phi = 45°$

b) Translation $= \vec{t} = \cancel{\ldots} \; 1 + 1' = \binom{0}{0} + \binom{3}{3.5} = (3, 3.5)$

$$T(\vec{t}) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 3.5 \\ 0 & 0 & 1 \end{bmatrix}$$

c) Skalierung $= \vec{s} = \sqrt{2} = \left(\frac{\sqrt{2}}{1}\right) = (\sqrt{2}, \sqrt{2})$

$$S(\vec{s}) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$M = \cancel{\ldots}$

Translation

Die ~~Rotation~~ muss im ~~Nachhinein~~ getätigt werden.

$$M = S(\vec{s}) \cdot R + T(\vec{t}) = \begin{bmatrix} \sqrt{2} \cdot \cos(45) + 1 & 0 + 0 & 0 + 3 \\ 0 & \sqrt{2}\cos(45) & 0 + 3.5 \\ 0 & 0 & 1 + 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 3 \\ 0 & 1 & 3.5 \\ 0 & 0 & 2 \end{bmatrix}$$

4