

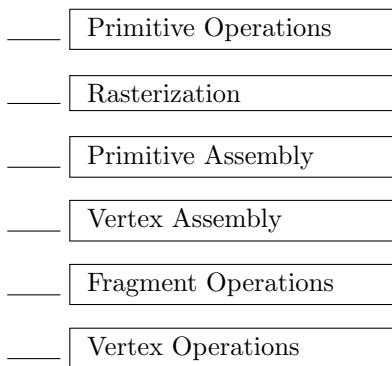
Probeklausur- Computergrafik

Wintersemester 2025/26

27. 01. 2026, 10¹⁵ – 11¹⁵ (60 min)

Aufgabe 1 (Render Pipeline)

- a) Gegeben seien folgende Schritte der Renderpipeline in nicht korrekter Reihenfolge. Notieren Sie neben den Boxen die Reihenfolge, in der diese Schritte innerhalb Pipeline durchgelaufen werden.



- b) Geben Sie zu jedem Schritt eine Operation an, die dort ausgeführt wird.

Aufgabe 2 (Transformationen & Geometrie)

- a) Nenne die vier Möglichkeiten Rotationen zu repräsentieren, die in der Vorlesung behandelt wurden.
- b) Gib eine Matrix an, die einen Punkt um 90 Grad dreht und danach um $(2, -3)$ verschiebt (gesucht wird eine Matrix zur Nutzung mit homogenen Koordinaten im zweidimensionalen Raum). Markiere welcher Teil der Matrix wofür zuständig ist.
- c) Nachfolgend sind drei 2×2 Matrizen und ein Vektor für den 2D-Raum definiert.

$$A = \begin{pmatrix} 1 & 0.5 \\ -0.2 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 1.5 & 0 \\ 0 & 1.5 \end{pmatrix} \quad \vec{v} = \begin{pmatrix} -2 \\ 6 \end{pmatrix}$$

Die Matrizen A, B und C stellen drei Transformationen dar, jedoch sollen nur zwei davon verwendet werden. Wie müssen der Vektor \vec{v} mit den Matrizen multipliziert werden, um zuerst die Skalierung, und dann die Rotation auszuführen? Notieren Sie zuerst die Formel für den neuen Vektor \vec{v}' , dann rechnen Sie die expliziten Werte aus.

- d) Gib ein Prädikat an (entweder als Formel oder Pseudocode), welches prüft, ob ein Punkt P vor einer Ebene liegt, die aus den Punkten A, B und C und der Normalen n besteht. Erläutere dieses kurz.

Aufgabe 3 (Shader)

Gegeben sind folgender Vertex- und Fragment-Shader:

Vertex Shader

```
#version 330 core
layout(location = 0) in vec3 position;
layout(location = 1) in vec3 normal;

uniform mat4 MVP; // model-view-projection matrix
uniform mat4 M;   // model-to-world matrix

out vec3 N;
out vec3 P;

void main() {
    P = vec3(M * vec4(position, 1.0));
    N = mat3(M) * normal;
    gl_Position = MVP * vec4(position, 1.0);
}
```

Source-Code 1: Vertex Shader

Fragment Shader

```
#version 330 core
in vec3 N;
in vec3 P;

uniform vec3 lightPos;

out vec4 fragColor;

void main() {
    vec3 L = normalize(lightPos - P);
    float diff = max(dot(normalize(N), L), 0.0);
    fragColor = vec4(vec3(diff), 1.0);
}
```

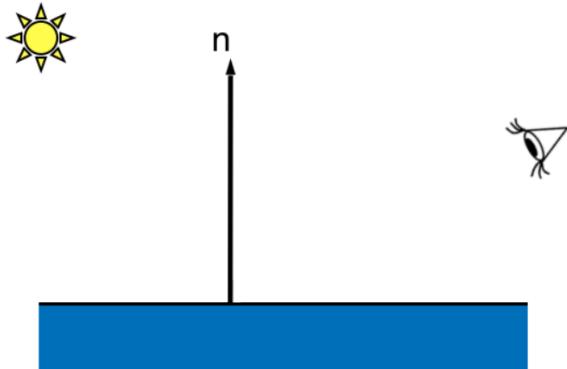
Source-Code 2: Fragment Shader

Kreuzen Sie für jede Aussage an, ob sie **wahr** oder **falsch** ist.

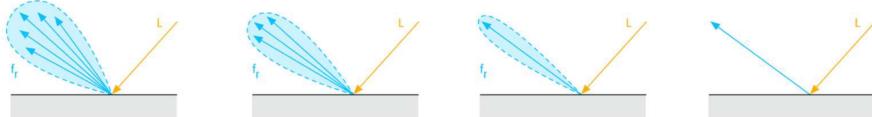
	wahr	falsch
a) Die Beleuchtung wird im World Space berechnet.	<input type="checkbox"/>	<input type="checkbox"/>
b) Die Normalen werden korrekt transformiert.	<input type="checkbox"/>	<input type="checkbox"/>
c) Nicht-uniformes Skalieren in M kann zu Beleuchtungsfehlern führen.	<input type="checkbox"/>	<input type="checkbox"/>
d) MVP könnte durch projection * view * M ersetzt werden.	<input type="checkbox"/>	<input type="checkbox"/>
e) Die Lichtquelle ist eine Punktlichtquelle.	<input type="checkbox"/>	<input type="checkbox"/>

Aufgabe 4 (Lighting)

- Nennen Sie drei unterschiedliche Arten von primären Lichtquellen.
- Benenne und beschreibe kurz die drei Komponenten des Phong-Shading-Models. Zeichne die relevanten drei Vektoren und zwei Winkel in die Abbildung ein. Erläutere noch kurz, für welchen der Terme diese relevant sind und wie sie diesen beeinflussen.

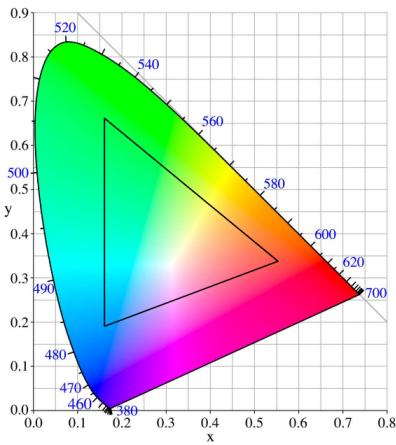


- Gegeben sind bildhaft dargestellt die folgenden Normal Distribution Functions (NDF) einer Microfacet BRDF Funktion. Darunter sind Sphären, welche ein metallisches Material mit einer der vier BRDF Funktionen haben. Ordnen Sie die jeder NDF die Sphäre mit dem entsprechenden Material zu. Welchen Effekt modelliert die NDF dementsprechend?



Aufgabe 5 (Farben)

- a) Es seien folgende, halbtransparente Objekte gegeben, die den gleichen Pixel überdecken:
- Objekt A mit $C_A = (0, 1, 0, \frac{1}{4})$
 - Objekt B mit $C_B = (\frac{1}{2}, 1, 0, \frac{1}{2})$
- Die Farbe $C_S = (0, 0, 1, 1)$ befindet sich schon auf dem Color Buffer. Nun wird zuerst Objekt A gerendert, und danach Objekt B. Berechnen Sie die Farbe C_R , die durch Alpha-Blending aus den Farben C_A , C_B , und C_S entsteht.
- b) Zu welchem Zweck wurden die Farbmodelle RGB, CMY(K) und HLS eingeführt? In welchen (technischen oder Anwendungs-) Bereichen werden sie verwendet?
- c) Gegeben sei die Farbe $C = (0.5, 0.3, 0.2)$ im RGB Farbraum. Geben Sie die dieselbe Farbe im CMY Farbraum an.
- d) Beschreiben Sie kurz was ein Color-Gamut ist, wie z.B. in der Abbildung dargestellt. Nennen Sie zwei Gründe, warum Color-Gamuts in der Computergrafik relevant sind.

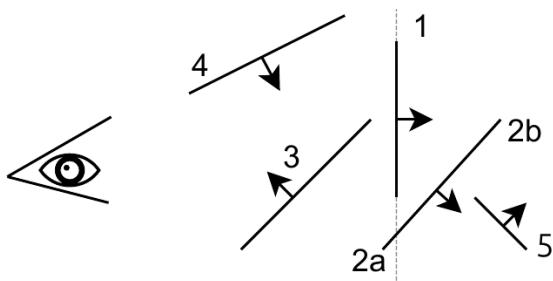


Aufgabe 6 (Visibility)

a) Konstruieren Sie den BSP-Tree für die gegebenen Polygone unter Beachtung der folgenden Regeln:

- Als Trennpolygon (Knoten) ist immer das Polygon mit der kleinsten numerischen Ziffer zu wählen.
- Falls ein Polygon durch ein Trennpolygon geschnitten wird, ist es entsprechend zu splitten (z. B. in 2a und 2b).
- Die an den Polygonen eingezeichneten Pfeile geben die Normalenrichtung der Polygone an.
- Der linke Teilbaum eines Knotens enthält die hinter dem Trennpolygon liegenden Polygone, der rechte Teilbaum die vorderen Polygone.

Zeichnen Sie den vollständigen BSP-Tree.



b) Geben Sie für den in Teil (a) konstruierten BSP-Tree die Reihenfolge an, in der die Polygone vom gegebenen Viewpoint aus gerendert werden, sodass eine korrekte Darstellung gemäß dem Painter's Algorithmus entsteht.

- Die Ausgabe soll als Liste der Polygonbezeichnungen erfolgen.
- Die Reihenfolge ist von hinten nach vorne anzugeben.