

Wintersemester 2024/25

Computergrafik - Aufgaben zur Klausurvorbereitung

Aufgabe 1 (Transformationen)

- a) Nenne die vier Möglichkeiten Rotationen zu repräsentieren, die in der Vorlesung behandelt wurden.
- b) Ordne den folgenden Anwendungsfällen die Repräsentationsmethode zu, die am besten geeignet ist und begründe deine Entscheidung kurz.
- Shader Programmierung

 - CAD / Modellierungssoftware

 - Game Engine UI

 - Game Engine Internal

 - Software zur Simulation von Robotergelenken (Tipp: hier müssen ggf. mehrere Rotationen interpoliert werden)
- c) Gib für Translation, Rotation, und Skalierung jeweils eine Beispielmatrix an für die Nutzung mit homogenen Koordinaten im zweidimensionalen Raum.
- d) Gib eine Matrix an, die einen Vektor um $(2, -3)$ verschiebt und um 180 Grad dreht (gleicher Raum / Dimension wie Aufgabe c)). Markiere welcher Teil der Matrix wofür zuständig ist.

Aufgabe 2 (Lighting & Colors)

a) Es seien folgende, halbtransparente Objekte gegeben, die den gleichen Pixel überdecken:

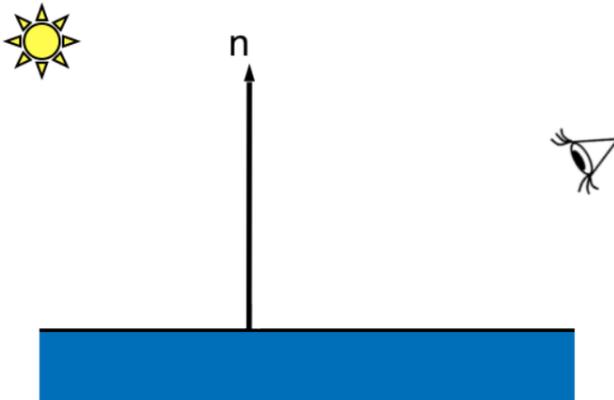
(a) Objekt A mit $C_A = (0, 1, 0, \frac{1}{2})$

(b) Objekt B mit $C_B = (\frac{1}{2}, 1, 0, \frac{1}{2})$

Die Farbe $C_S = (0, 0, 1, 1)$ befindet sich schon im 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) Nenne drei relevante Modelle zur Repräsentation von Farben aus der Vorlesung. Beschreibe kurz, wie jedes Farbmodell eine Farbe codiert. Stelle die Farbe Cyan in allen drei Modellen dar.

c) Benenne und beschreibe kurz die drei Komponenten des Phong-Shading-Modells. 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.



Aufgabe 3 (Render Pipeline)

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

_____	Primitive Operations
_____	Rasterization
_____	Primitive Assembly
_____	Vertex Assembly
_____	Fragment Operations
_____	Vertex Operations

- b) Bei Postprocessing wird das Originalbild aus dem Framebuffer i.A. in eine oder mehrere Textur(en) geschrieben und anschließend weiterverarbeitet. Der folgende Fragmentshader wird benutzt, um zwei Dreiecke auf den kompletten Viewport abzubilden. Welche 2 Postprocessing-Effekte werden durch den folgenden Fragment-Shader realisiert?

```
1 #version 330 core
2
3 uniform sampler2D renderedTexture; // Textur aus dem ersten
4                                     //           Rendering-Pass
5 in vec2 UV;                          // interpolierte UV-Koordinaten
6 out vec3 color;                       // zu berechnender Farbwert
7
8 void main()
9 {
10     color = vec3(1,1,1) -
11             texture( renderedTexture, vec2(UV.x, -UV.y) ).rgb;
12 }
```

Aufgabe 4 (Geometry)

- a) Gib ein Prädikat an (entweder als Formel oder Pseudocode), welches prüft, ob ein Punkt vor oder hinter eine Ebene liegt. Erläutere dieses kurz.
- b) Entwickle mit dem Prädikat aus Teilaufgabe a) ein erweitertes Prädikat, welches prüft, ob ein Punkt in einem gegebenen Tetraeder T liegt. Du kannst dabei annehmen, dass die Normalen der Dreiecke nach außen zeigen. Außerdem kannst du für die Aufgabe das Prädikat aus Teilaufgabe a) mit einem Kürzel angeben.

Aufgabe 5 (Rasterization)

- a) Gegeben sei eine Linie zwischen den Punkten P_0 und P_1 . Skizziere einen Algorithmus (mit Pseudocode) zur inkrementellen Rasterisierung der Linie.
- b) Warum berechnen wir die Linie inkrementell?