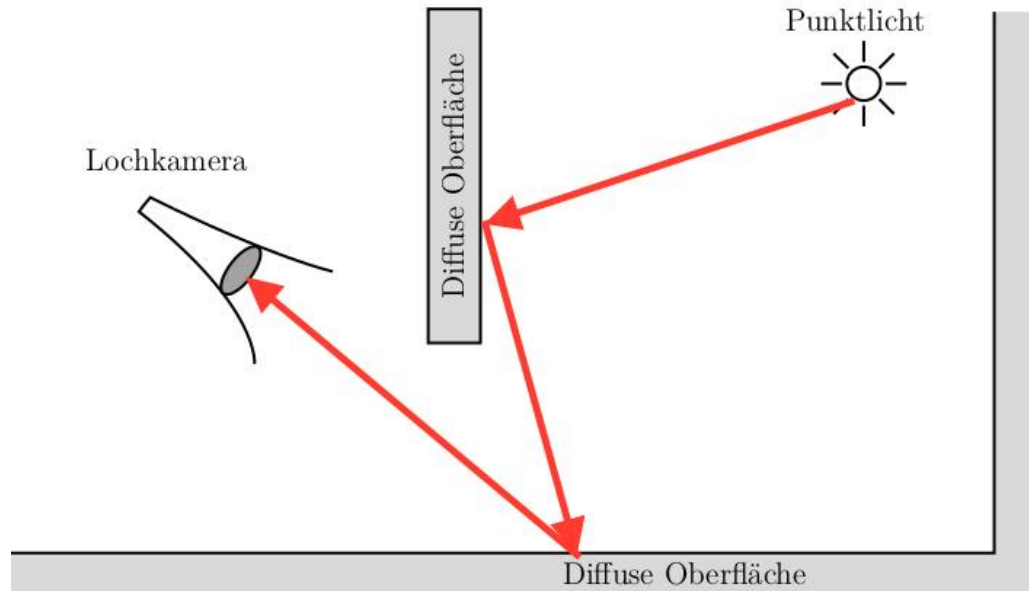


## Aufgabe 1: Raytracing

### Teilaufgabe 1a

Zeichnen Sie in der folgenden Szene mit Liniensegmenten einen möglichst einfachen Lichtpfad von der Kamera zur Punktlichtquelle ein, der zwar in der geometrischen Optik auftreten kann, jedoch nicht mit Whitted-Style Raytracing erzeugt werden kann!



### Teilaufgabe 1b

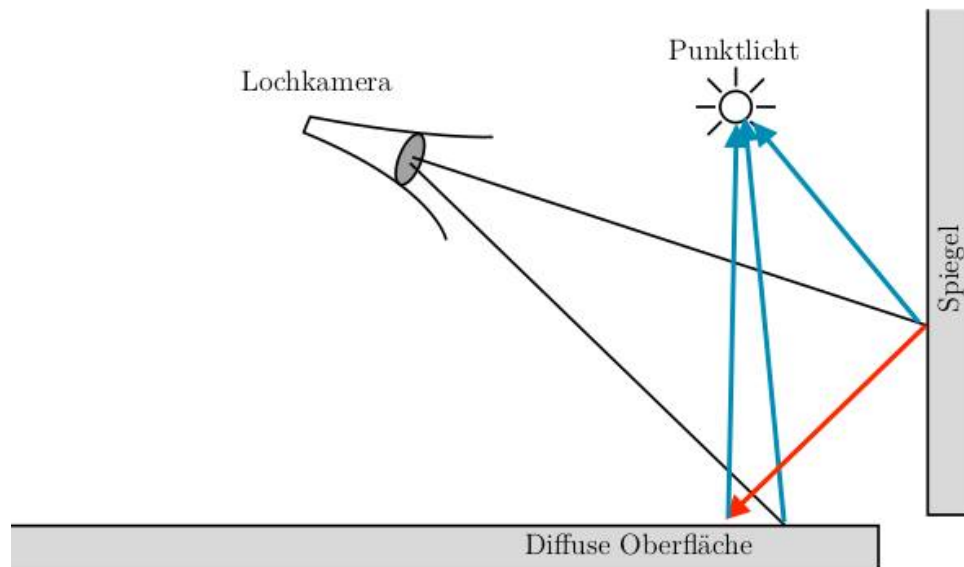
Begründen Sie kurz, warum der Pfad nicht mit Whitted-Style Raytracing erzeugt werden kann!

Whitted-Style sendet nur von Kamera Strahlen aus und verfolgt Schatten-, Transmission und Reflektionsstrahlen als Sekundärstrahlen. Da aber von der Kamera aus verfolgt wird, kann (in diesem Fall) kein Strahl die Rückseite (aus Sicht der Kamera) der diffusen Fläche treffen  $\Rightarrow$  die indirekte Beleuchtung die durch diffuses Abstrahlen von der Rückseite der diffusen Fläche erfolgt, kann mit Whitted-Style Raytracing nicht verfolgt werden.

### Teilaufgabe 1c

In der folgenden Abbildung sind zwei Primärstrahlen gegeben. Führen Sie mit diesen zeichnerisch Whitted-Style Raytracing durch! Kennzeichnen Sie Reflexionsstrahlen mit

„R“, Transmissionsstrahlen mit „T“ und Schattenstrahlen mit „S“!



### Teilaufgabe 1d

1. **Ray generation:** Erzeuge Sichtstrahlen durch jeden Pixel.
2. **Ray intersection:** Schnittberechnung; also: Finde Objekt welches den Strahl schneidet und am nächsten zur Kamera ist.
3. **Shading:** Schattierung / Beleuchtungsberechnung.

## Aufgabe 2: Farben

### Teilaufgabe 2a

Wie nennt man die Funktionen, mit denen man Tristimulus-Werte zu einem gegebenen Spektrum berechnen kann?

Color Matching Funktionen

### Teilaufgabe 2b

- Es gibt eine lineare Abbildung zwischen den Farbräumen XYZ und xyY.
- Falsch, da bei der Umrechnung durch  $X + Y + Z$  dividiert wird ( $x = \frac{X}{X+Y+Z}$ ,  $y = \frac{Y}{X+Y+Z}$ )

- *Es gibt eine lineare Abbildung zwischen den Farbräumen RGB und XYZ.*
- Korrekt, der Farbraum wurde konstruiert, um eine möglichst einfache Umrechnung zu ermöglichen.
- *Die subjektiv empfundene Stärke von Sinneseindrücken ist proportional zur Intensität des physikalischen Reizes.*
- Falsch, es besteht ein logarithmischer Zusammenhang.

### Teilaufgabe 2c

*Welche Information beinhalten die  $x$ - und  $y$ -Komponenten einer Farbdarstellung im CIE- $xyY$ -Farbraum zusammengefasst?*

Sie beschreiben die Farbe (Chromatizität) unabhängig von der Intensität.

### Teilaufgabe 2d

*Ein RGB-Eingabebild wird in den CIE- $xyY$ -Farbraum transformiert. Um Speicher zu sparen werden die  $x$ - und  $y$ -Komponenten in geringerer Auflösung gespeichert, während die Auflösung der  $Y$ -Komponente beibehalten wird.*

*Warum ist dieses Vorgehen im Hinblick auf einen menschlichen Betrachter deutlich besser, als die Auflösung der RGB-Komponenten zu reduzieren?*

Die Kontrastsensitivität für Chrominanz (Farbe) ist vor allem im hochfrequenten Bereich deutlich geringer als für Luminanz, daher kann man den Chrominanz-Anteil ( $xy$ -Werte) in geringerer Auflösung speichern (was dem Weglassen hoher Frequenzen entspricht), ohne dass mit den Augen ein großer Unterschied erkennbar ist.

## Aufgabe 3: Homogene Koordinaten

*Beim Raytracing soll ein in Weltkoordinaten definierter Strahl mit einem Objekt in dessen lokalen Koordinatensystem geschnitten werden. Die affine Transformation von lokalen in Weltkoordinaten ist durch die homogene Transformationsmatrix  $M \in \mathbb{R}^{4 \times 4}$  gegeben.*

### Teilaufgabe 3a

*Mit welcher Matrix kann der Strahl von Weltkoordinaten in lokale Koordinaten transformiert werden?*

Mit der Matrix  $M^{-1}$ , da diese die inverse Transformation von Punkten aus *lokalen* in *Weltkoordinaten* darstellt.

### Teilaufgabe 3b

Der Strahl  $\mathbf{R}(t) = \mathbf{E} + t\mathbf{d}$ , ( $\mathbf{E}, \mathbf{d} \in \mathbb{R}^3$ ) soll mit der Matrix aus der Teilaufgabe a) in das lokale Koordinatensystem transformiert werden. Geben Sie dazu den transformierten Punkt  $\mathbf{E}'$  und die transformierte Richtung  $\mathbf{d}'$  an!

Zur Vereinfachung geht man zu erweiterten Koordinaten  $\begin{pmatrix} \mathbf{E} \\ 1 \end{pmatrix}$  und  $\begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$  über (1 für Punkte, 0 für Richtungen, da diese Differenzen zweier Punkte darstellen).

Damit erhält man  $\begin{pmatrix} \mathbf{E}' \\ 1 \end{pmatrix} = M^{-1} \begin{pmatrix} \mathbf{E} \\ 1 \end{pmatrix}$  und  $\begin{pmatrix} \mathbf{d}' \\ 0 \end{pmatrix} = M^{-1} \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$

### Teilaufgabe 3c

Gegeben sei nun

$$M = \begin{pmatrix} 0 & 0 & -1 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Berechnen Sie für diese Werte die Matrix aus Aufgabe a)!

Invertiere die Matrix:

$$\left( \begin{array}{cccc|cccc} 0 & 0 & -1 & 5 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right) \longrightarrow \dots \longrightarrow \left( \begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right)$$

Allgemein gilt für eine Matrix der Form  $M = \begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix}$ :  $M^{-1} = \begin{pmatrix} A^{-1} & -A^{-1}t \\ 0 & 1 \end{pmatrix}$

## Aufgabe 4: Transformationen

### Teilaufgabe 4a

Gegeben sei folgendes Liniensegment  $(\mathbf{p}_0, \mathbf{p}_1)$  im Clip-Space:

$$\mathbf{p}_0 = \begin{pmatrix} -1 \\ 1 \\ 5 \\ 10 \end{pmatrix}, \quad \mathbf{p}_1 = \begin{pmatrix} 0.5 \\ -0.5 \\ 15 \\ 5 \end{pmatrix}$$

#### Teilaufgabe 4a (i)

Wie lauten die Normalized Device Coordinates der Punkte  $\mathbf{p}_0$  und  $\mathbf{p}_1$ ?

TODO

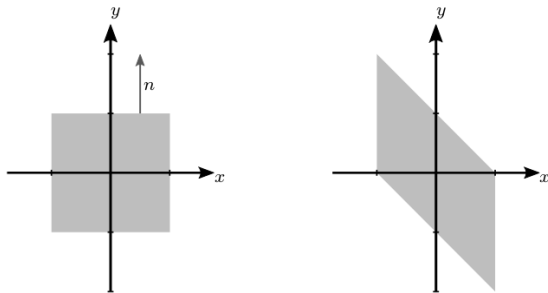
#### Teilaufgabe 4a (ii)

Warum sollte man dieses Liniensegment vor der Normalisierungstransformation (Dehomogenisierung) clippen?

TODO

### Teilaufgabe 4b

Die folgende Abbildung zeigt ein Quadrat in 2D vor (links) und nach einer Scherung (rechts).



i) Geben Sie die Transformationsmatrix  $M \in \mathbb{R}^{2 \times 2}$  an, die die Scherung beschreibt!

Die Scherung hat zur Folge, dass beim Erhöhen der x-Koordinate um eine Einheit die y-Koordinate um eine Einheit verringert wird. Damit hat die Matrix die Form

$$M = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$$

- ii) Wie berechnet man allgemein die Matrix  $N \in \mathbb{R}^{2 \times 2}$  an, die zur Transformation der Normalen verwendet werden muss? Bestimmen Sie diese Matrix!

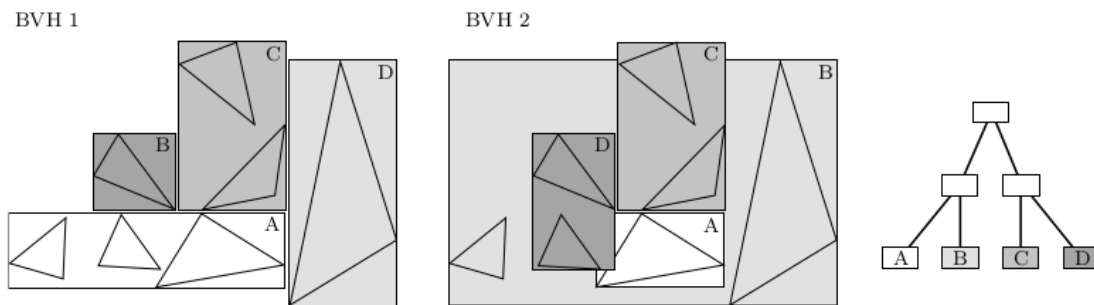
$$N = (M^{-1})^\top = \left( \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}^{-1} \right)^\top = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^\top = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

- iii) Berechnen Sie mithilfe von  $N$  die transformierte Normale  $\mathbf{n}'$  zur in der Abbildung eingezeichneten Normalen  $\mathbf{n}$ !

$$\mathbf{n}' = \frac{N\mathbf{n}}{\|N\mathbf{n}\|} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

## Aufgabe 5: Beschleunigungsstrukturen und Hüllkörper

### Teilaufgabe 5a



Welche der beiden Hierarchien kann von einem Raytracer effizienter durchlaufen werden? Begründen Sie dies in Stichpunkten!

TODO

## Teilaufgabe 5b

#	Aussage	Wahr	Falsch	Kommentar
1	Ein kD-Baum kann für $N$ Primitive den Aufwand für Strahlschnitte von $O(N)$ auf $O(\log(N))$ reduzieren.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	average case
2	Die Surface Area Heuristic sorgt dafür, dass bei den Kindknoten gleich viele Primitive zugeteilt werden.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	minimale Kosten
3	Für die Suche der Trennebene (Split Plane) mit Objektmittel (Object Median) ist immer eine vollständige Sortierung der Primitive notwendig.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sortiert werden die Objektmittelpunkte
4	Teilen eines Knotens am Objektmittel (Object Median) führt stets zu den effizientesten Hüllkörperhierarchien.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SAH
5	Die Surface Area Heuristic macht die Annahme, dass Strahlen stets außerhalb des zu unterteilen Hüllkörpers starten.	TODO	TODO	
6	Es gibt Szenen, in denen ein kD-Baum keinen Vorteil bringt.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identische Dreiecke

## Teilaufgabe 5c

TODO: Bitte kontrollieren!

#	Aussage	AABB	OBB	Kugel
1	Der Hüllkörper kann einen beliebig orientierten Würfel optimal, also ohne freien Raum umschliessen.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Es gibt orthonormale Transformationen des eingeschlossenen Objektes, die das Volumen des optimalen Hüllkörpers verändern.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Es gibt affine Transformationen des eingeschlossenen Objektes, die das Volumen des Hüllkörpers verändern. <sup>1</sup>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Gegeben sind zwei Objektmengen und zu jeder Menge ihr optimaler Hüllkörper. Der Aufwand, den optimalen Hüllkörper für alle Objekte zu bestimmen, ist unabhängig von der Anzahl der Objekte.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

<sup>1</sup>Skalierung. Allerdings gibt es nur für AABBs affine Transformationen, welche  $\frac{V_{\text{Hüllkörper}} - V_{\text{Objekt}}}{V_{\text{Hüllkörper}}}$  verändert.

## Aufgabe 6: Texturen

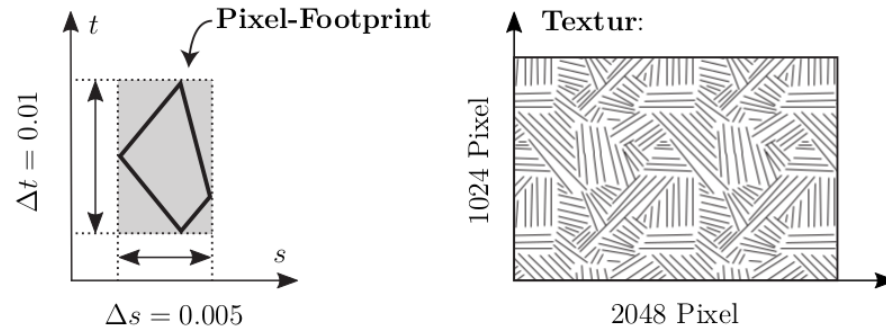
### Teilaufgabe 6a

#	Aussage	Wahr	Falsch	Kommentar
I1	Das Filtern von Texturen wird üblicherweise bei der Minification und der Magnification angewandt, um Aliasing-Artefakte zu vermeiden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nur Magnification
I2	Das Filtern von Texturen benötigt bei Verfahren ohne Vorfilterung mehr Zugriffe auf den Texturspeicher als einfaches Auslesen.	TODO	TODO	
II1	Mipmapping bekämpft das Aliasing-Problem bei der Unterabtastung von Texturen.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
II2	Mipmapping berücksichtigt anisotrope Footprints.	TODO	TODO	
III1	Bei Einsatz von Summed-Area-Tables kann man mit nur vier Texturzugriffen die Summe über eine kreisförmige Fläche berechnen.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
III2	Bei Einsatz von Summed-Area-Tables benötigt man im Allgemeinen mehr Bits zur Repräsentation der enthaltenen Werte als bei gewöhnlichen Texturen mit dem gleichen Wertebereich.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	



## Teilaufgabe 6b

Eine Textur wird mit einem trilinearen Filter abgetastet. Die folgende Abbildung zeigt den Pixel-Footprint im  $(s, t)$ -Texturraum ( $s, t \in [0, 1]$ ) und rechts daneben die Textur:



Welche Mipmap-Stufen werden für die trilineare Interpolation verwendet, wenn die Textur eine Auflösung von  $2048 \times 1024$  hat? (Mipmap-Stufe 0 hat die größte Auflösung). Begründen Sie Ihre Antwort!

Texturgröße:

$$0.005 \cdot 2048 = 10.24 \quad (1)$$

$$0.01 \cdot 1024 = 10.24 \quad (2)$$

$$\Rightarrow 10.24 \cdot 10.24 = 104.8576 \quad (3)$$

Stufen:

1.  $1024 \times 512$
2.  $512 \times 256$
3.  $256 \times 128$
4.  $128 \times 64$
5.  $64 \times 32$
6.  $32 \times 16$
7.  $16 \times 8 = 128$
8.  $8 \times 4 = 32$

Es werden also die Stufen 7 und 8 verwendet.

## Aufgabe 7: Beleuchtung

### Teilaufgabe 7a

$\mathbf{P} = \lambda_1 \mathbf{P}_1 + \lambda_2 \mathbf{P}_2 + \lambda_3 \mathbf{P}_3$ , wobei  $\lambda_1 + \lambda_2 + \lambda_3 = 1$  und  $\lambda_1, \lambda_2, \lambda_3 \geq 0$

### Teilaufgabe 7b

- Berechne  $I_i = k_d I_L(\mathbf{n}_i \cdot \text{normalize}(\mathbf{L} - \mathbf{P}_i))$  für  $i = 1, 2, 3$
- Interpoliere  $I = \lambda_1 I_1 + \lambda_2 I_2 + \lambda_3 I_3$
- Bemerkung:  $\text{normalize}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$

### Teilaufgabe 7c

- Interpoliere  $\mathbf{n} = \text{normalize}(\lambda_1 \mathbf{n}_1 + \lambda_2 \mathbf{n}_2 + \lambda_3 \mathbf{n}_3)$
- Berechne  $I = k_d I_L(\mathbf{n} \cdot \text{normalize}(\mathbf{L} - \mathbf{P}))$

### Teilaufgabe 7d

In den Eckpunkten. Allgemein gilt im  $i$ -ten Eckpunkt:  $\lambda_i = 1$  sowie  $\lambda_j = 0$  ( $\forall j \neq i$ ). Somit sind die Berechnungen in diesen Punkten äquivalent, wie man leicht aus den obigen Formeln entnehmen kann.

### Teilaufgabe 7e

Im rechten Fall wird mehr Licht zum betrachter reflektiert.

### Teilaufgabe 7f

Fresnel-Effekt

## Aufgabe 8: Partikeleffekte und OpenGL-Blending

### Teilaufgabe 8a

*Zunächst werden nur Feuerpartikel gerendert. Konfigurieren Sie OpenGL so, dass die Feuerpartikel Licht in Richtung des Betrachters emittieren!*

---

```
1 glBlendEquation(GL_FUNC_ADD);
2 glBlendFunc(sfactor=GL_ONE, dfactor=GL_ONE);
```

---

### Teilaufgabe 8b

*Warum ist der Tiefentest aktiviert? Warum ist Schreiben in den Tiefenpuffer deaktiviert?*

TODO

### Teilaufgabe 8c

*Zunächst werden nur Feuerpartikel gerendert. Konfigurieren Sie OpenGL so, dass die Feuerpartikel Licht in Richtung des Betrachters emittieren!*

---

```
1 glBlendEquation(GL_FUNC_ADD);
2 glBlendFunc(sfactor=GL_ONE, dfactor=GL_ONE);
```

---

## Teilaufgabe 8d

*Ihnen fällt auf, dass sich Rauch und Feuer nicht korrekt zusammenfügen, wenn Sie erst Feuerpartikel wie in Aufgabe a) und dann Rauchpartikel wie in Aufgabe c) rendern. Welche Schritte müssen Sie vornehmen, um das Problem zu beheben?*

Den Renderablauf verändern:

- ☐ ☒ TODO Zuerst die Rauchpartikel zeichnen, dann die Feuerpartikel.
- ☐ ☒ TODO `glDepthFunc(GL_GREATER)`: Zeichenreihenfolge umkehren.
- ☐ ☒ TODO Rauchpartikel front-to-back sortieren.
- ☐ ☒ TODO Feuerpartikel und Rauchpartikel zusammen sortieren und zeichnen.

OpenGL Blending-Konfigurationen:

- ☐ ☒ TODO `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`
- ☐ ☒ TODO `glBlendFunc(GL_FUNC_ADD, GL_ONE)`
- ☐ ☒ TODO `glBlendFunc(GL_ONE_MINUS_DST_ALPHA, GL_SRC_ALPHA)`
- ☐ ☒ TODO `glBlendFunc(GL_ONE, GL_ONE_MINUS_SRC_ALPHA)`

Den Fragment-Shader verändern:

- ☐ ☒ TODO Alpha-Ausgabewert von Feuerpartikeln auf (1 - Texel-Alpha) setzen.
- ☐ ☒ TODO Alpha-Ausgabewert von Feuerpartikeln auf 1 setzen.
- ☐ ☒ TODO Alpha-Ausgabewert von Feuerpartikeln auf 0 setzen.
- ☐ ☒ TODO Ausgabewerte unverändert beibehalten.

Vertex-Attribute hinzufügen:

- ☐ ☒ TODO zur feingranularen Anpassung der OpenGL Blending-Konfiguration,
- ☐ ☒ TODO zur Unterscheidung von Feuer- und Rauchpartikeln,
- ☐ ☒ TODO zur Einstellung der Zeichenebene.

## Aufgabe 9: OpenGL

#	Aussage	Wahr	Falsch
1	Die Dehomogenisierung wird in der OpenGL-Pipeline nach dem Fragment-Shader durchgeführt.	TODO	TODO
2	Die Pipeline-Stufe „Primitive Assembly“ ist programmierbar.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Clipping am Sichtvolumen wird von der OpenGL-Pipeline vor dem Fragment-Shader durchgeführt.	TODO	TODO
4	Shader-Programme können den Zustand des OpenGL-Zustandsautomaten (State Machine) verändern.	TODO	TODO
5	Die Beleuchtungsberechnung muss im Fragment-Shader immer in Kamera-Koordinaten durchgeführt werden.	TODO	TODO
6	Texturen können ausschließlich im Fragment-Shader gelesen werden.	TODO	TODO

## Aufgabe 10: Reflexionen in OpenGL

---

```
1 // ...
2 uniform float h; // Y-Koordinate der Wasserebene im World Space
3 in vec3 P; // World-Space-Position des aktuellen Fragments
4 // ...
5
6 void main(void)
7 {
8     // Implementieren Sie hier das Clipping:
9     if (P.y < h) {
10         discard;
11     }
12
13     // Es folgt das spezifische Shading des jeweiligen Objekts
14     // ...
15 }
```

---

## Aufgabe 11: GLSL-Hatching

### Teilaufgabe 11a

$$\left(1 - \frac{x - x_1}{x_2 - x_1}\right) y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

oder

$$y_1 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

## Teilaufgabe 11b

---

```
1 uniform sampler2D dunkel; // dunkle Textur
2 uniform sampler2D mittel; // mittlere Textur
3 uniform sampler2D hell; // helle Textur
4 in vec2 tex_coord; // Texturkoordinate
5
6 vec4 get_hatched_color(float h) // Helligkeit h liegt in [0, 1].
7 {
8     vec4 t_d = texture(dunkel, tex_coord);
9     vec4 t_m = texture(mittel, tex_coord);
10    vec4 t_h = texture(hell, tex_coord);
11
12    float weight_d = clamp(1. - 4. * (h - .25), 0., 1.);
13    float weight_m = clamp(1 - 4. * abs(h - .5), 0., 1.);
14    float weight_h = clamp(4. * (h - .5), 0., 1.);
15
16    return weight_d * t_d + weight_m * t_m + weight_h * t_h;
17 }
```

---

## Aufgabe 12: Bézierkurven

### Teilaufgabe 12a

Was versteht man unter affiner Invarianz bei Bézierkurven?

Sei  $F(u) = \sum_{i=0}^n \mathbf{b}_i B_i^n$  eine Bézierkurve und  $\varphi(x) = Ax + t$  eine affine Funktion. Dann gilt:

$$\varphi(F(u)) = \sum_{i=0}^n B_i^n \varphi(\mathbf{b}_i)$$

### Teilaufgabe 12b



### Teilaufgabe 12c

Geben Sie an, ob es sich bei den folgenden Kurven mit gegebenem Kontrollpolygon um Bézierkurven handelt! Begründen Sie jeweils kurz Ihre Antwort, falls es sich nicht um eine Bézierkurve handelt!

1. Nein, da die Kurve nicht symmetrisch ist. Insbesondere müsste sie durch die Mitte des zweiten und dritten Kontrollpunktes gehen.
2. Ja
3. Nein, da die Kurve nicht innerhalb der konvexen Hülle der Kontrollpunkte ist.