

## Hauptklausur Computergrafik

WS 2014/15

11. März 2015

Kleben Sie hier  
**vor Bearbeitung  
der Klausur** den  
Aufkleber auf.

### Beachten Sie:

- Trennen Sie vorsichtig die dreistellige Nummer von Ihrem Aufkleber ab. Sie sollten sie gut aufheben, um später Ihre Note zu erfahren.
- Die Klausur umfasst 21 Seiten (11 Blätter) mit 12 Aufgaben.
- Es sind **keine Hilfsmittel** zugelassen.
- Vor Beginn der Klausur haben Sie 5 Minuten Zeit zum *Lesen* der Aufgabenstellungen. Danach haben Sie **60 Minuten** Bearbeitungszeit.
- Schreiben Sie Ihren Namen und Ihre Matrikelnummer oben auf jedes bearbeitete Aufgabenblatt.
- Schreiben Sie Ihre Lösungen auf die Aufgabenblätter. Bei Bedarf können Sie weiteres Papier anfordern.
- Wenn Sie bei einer Multiple-Choice-Frage eine falsche Antwort angekreuzt haben und diesen Fehler korrigieren möchten, füllen Sie das betreffende Kästchen ganz aus:



- Falsche Kreuze bei Wahr-Falsch Multiple-Choice-Aufgaben führen zu Punktabzug. Jede Teilaufgabe wird mit mindestens 0 Punkten bewertet.

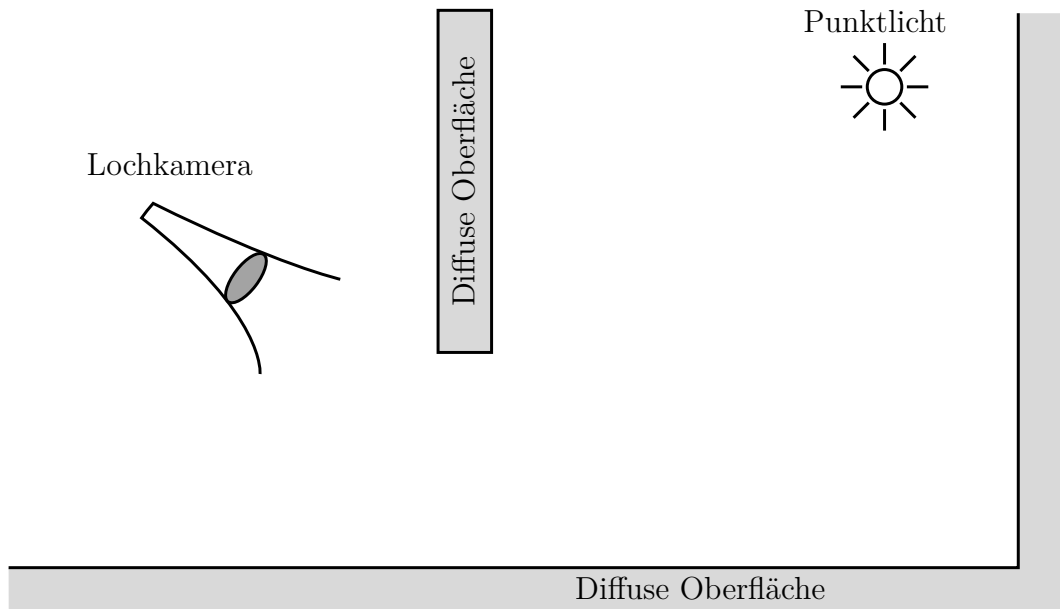
Aufgabe	1	2	3	4	5	6	7	8	9	10	11	12	Gesamt
Erreichte Punkte													
Erreichbare Punkte	11	8	8	10	13	14	13	12	6	4	8	13	120

Note



### Aufgabe 1: Raytracing (11 Punkte)

- a) 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! (2 Punkte)

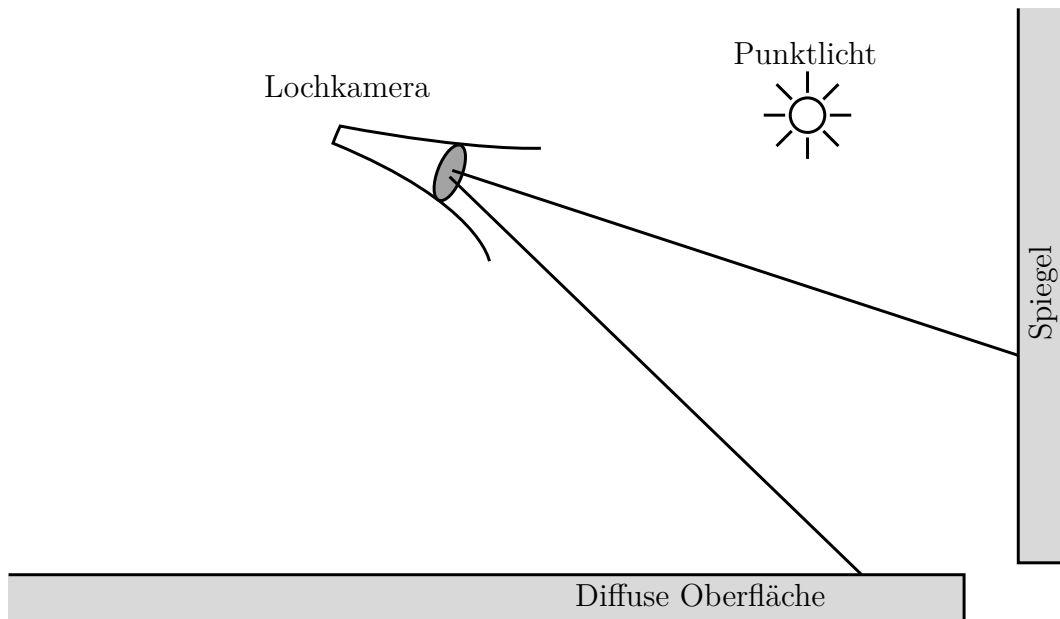


- b) Begründen Sie *kurz*, warum der Pfad nicht mit Whitted-Style Raytracing erzeugt werden kann! (2 Punkte)

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

- c) 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“! (4 Punkte)

☐

- d) In welche drei Schritte lässt sich das prinzipielle Vorgehen bei jedem Raytracing-Programm unterteilen? (3 Punkte)

☐

**Aufgabe 2: Farben (8 Punkte)**☐

- a) Wie nennt man die Funktionen, mit denen man Tristimulus-Werte zu einem gegebenen Spektrum berechnen kann? **(1 Punkt)**

☐

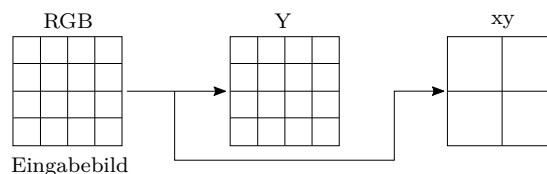
- b) Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen! **(3 Punkte)**

Aussage	Wahr	Falsch
Es gibt eine lineare Abbildung zwischen den Farbräumen XYZ und xyY.	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt eine lineare Abbildung zwischen den Farbräumen RGB und XYZ.	<input type="checkbox"/>	<input type="checkbox"/>
Die subjektiv empfundene Stärke von Sinneseindrücken ist proportional zur Intensität des physikalischen Reizes.	<input type="checkbox"/>	<input type="checkbox"/>

☐

- c) Welche Information beinhalten die x- und y-Komponenten einer Farbdarstellung im CIE-xyY-Farbraum zusammengekommen? **(1 Punkt)**

- d) 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? **(3 Punkte)**

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

### Aufgabe 3: Homogene Koordinaten (8 Punkte)



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.

- a) Mit welcher Matrix kann der Strahl von Weltkoordinaten in lokale Koordinaten transformiert werden? (1 Punkt)



- b) 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! (3 Punkte)



- c) 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)! (4 Punkte)





#### Aufgabe 4: Transformationen (10 Punkte)

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

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



i) Wie lauten die Normalized Device Coordinates der Punkte  $\mathbf{p}_0$  und  $\mathbf{p}_1$ ? (2 Punkte)

$$\mathbf{p}_0^{NDC} = \begin{pmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{pmatrix}, \quad \mathbf{p}_1^{NDC} = \begin{pmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{pmatrix}$$

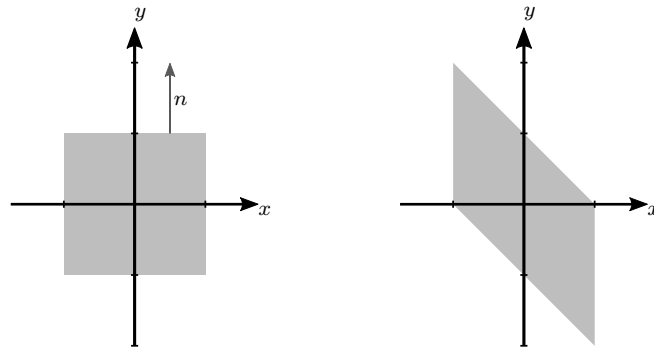


ii) Warum sollte man dieses Liniensegment vor der Normalisierungstransformation (Dehomogenisierung) clippen? (2 Punkte)

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

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



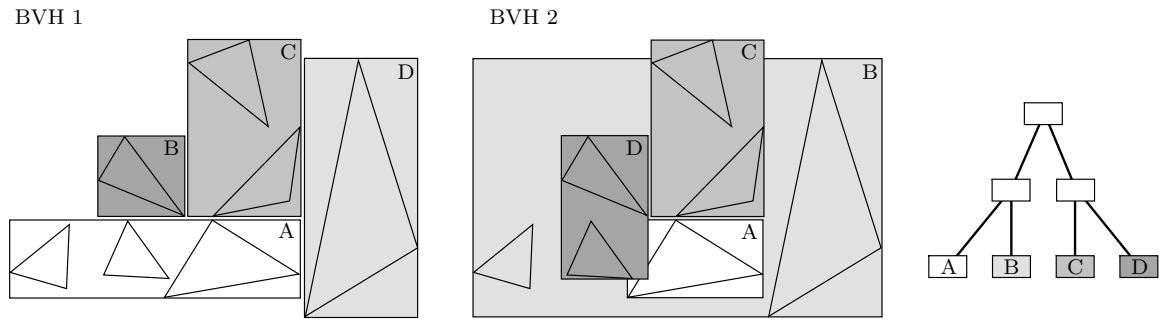
- i) Geben Sie die Transformationsmatrix  $M \in \mathbb{R}^{2 \times 2}$  an, die die Scherung beschreibt! **(2 Punkte)**

- 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! **(3 Punkte)**


- iii) Berechnen Sie mithilfe von  $N$  die transformierte Normale  $n'$  zur in der Abbildung eingezeichneten Normalen  $n$ ! **(1 Punkt)**

### Aufgabe 5: Beschleunigungsstrukturen und Hüllkörper (13 Punkte)

- a) Die folgende Abbildung zeigt die Blattknoten zweier Hüllkörperhierarchien (BVH). Beide Hierarchien beinhalten dieselben Dreiecke und besitzen je vier Blattknoten A, B, C, D. Die Topologie beider BVH ist gleich und als Baum rechts dargestellt.



Welche der beiden Hierarchien kann von einem Raytracer effizienter durchlaufen werden? Begründen Sie dies in *Stichpunkten!* (3 Punkte)

BVH 1 BVH 2 ☐



Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

- b) Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen! (6 Punkte) ☐

Aussage	Wahr	Falsch
Ein kD-Baum kann für $N$ Primitive den Aufwand für Strahlschnitte von $O(N)$ auf $O(\log(N))$ reduzieren.	<input type="checkbox"/>	<input type="checkbox"/>
Die Surface Area Heuristic sorgt dafür, dass beiden Kindknoten gleich viele Primitive zugeteilt werden.	<input type="checkbox"/>	<input type="checkbox"/>
Für die Suche der Trennebene (Split Plane) mit Objektmittel (Object Median) ist <i>immer</i> eine vollständige Sortierung der Primitive notwendig.	<input type="checkbox"/>	<input type="checkbox"/>
Teilen eines Knotens am Objektmittel (Object Median) führt stets zu den effizientesten Hüllkörperhierarchien.	<input type="checkbox"/>	<input type="checkbox"/>
Die Surface Area Heuristic macht die Annahme, dass Strahlen stets außerhalb des zu unterteilenden Hüllkörpers starten.	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt Szenen, in denen ein kD-Baum keinen Vorteil bringt.	<input type="checkbox"/>	<input type="checkbox"/>

- c) Kreuzen Sie jeweils die passenden Kästchen an! Sie erhalten für jede vollständig richtige Zeile einen Punkt. (4 Punkte) ☐

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

**Aufgabe 6: Texturen (14 Punkte)**

- a) Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen! In jedem Block können eine, zwei oder keine Aussage wahr sein. Pro korrekt angekreuztem Block gibt es zwei Punkte. **(6 Punkte)**

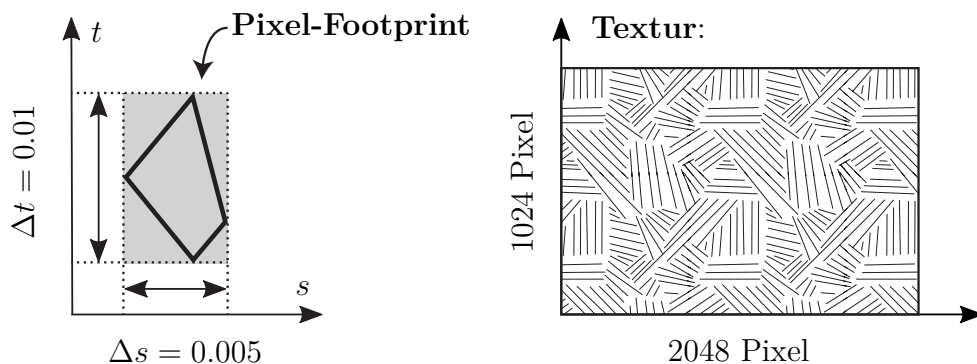
☐

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

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

- b) 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:



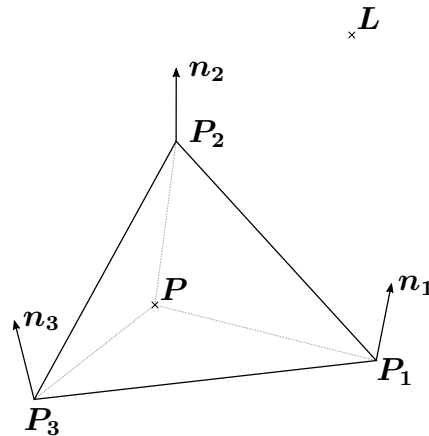
Welche Mipmap-Stufen werden für die trilineare Interpolation verwendet, wenn die Textur eine Auflösung von  $2048 \times 1024$  hat? (Mipmap-Stufe 0 hat die größte Auflösung). Begründen Sie Ihre Antwort! (8 Punkte)





### Aufgabe 7: Beleuchtung (13 Punkte)

Für einen Punkt  $\mathbf{P}$  auf einem Dreieck sind die baryzentrischen Koordinaten  $\lambda_i$ , sowie pro Vertex des Dreiecks seine Position  $\mathbf{P}_i$  und Normale  $\mathbf{n}_i$  gegeben. Das Material des Dreiecks ist rein diffus und hat die Reflektanz  $k_d$ . Die Lichtquelle befindet sich an der Position  $\mathbf{L}$ . Den distanzabhängigen Intensitätsabfall der Lichtquelle können Sie ignorieren.



a) Geben Sie an, wie  $\mathbf{P}$  aus den Punkten  $\mathbf{P}_i$  berechnet wird! (1 Punkt)



b) Geben Sie die Vorschrift für die Berechnung der Beleuchtung mittels *Gouraud-Shading* am Punkt  $\mathbf{P}$  an! (4 Punkte)

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

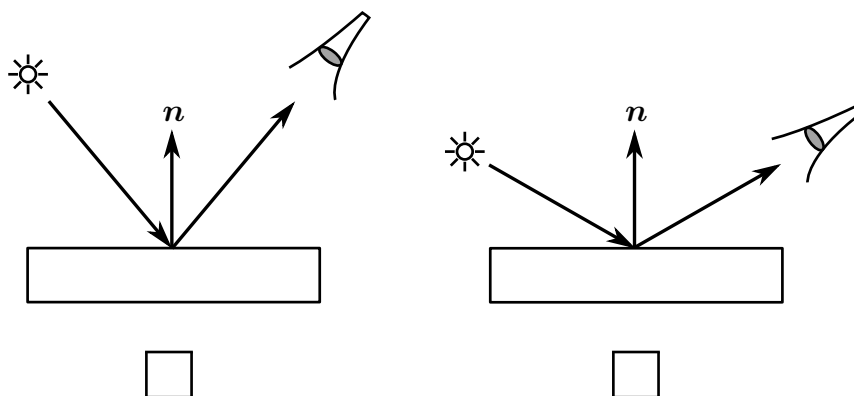
- c) Geben Sie die Vorschrift für die Berechnung der Beleuchtung mittels *Phong-Shading* am Punkt ***P*** an! (4 Punkte)

☐

- d) An welchen Punkten berechnen beide Algorithmen immer ein identisches Ergebnis? (2 Punkte)

☐

- e) Die folgende Abbildung zeigt Reflexionen an einer Glasscheibe. Kreuzen Sie an, in welcher Situation ein größerer Anteil des Lichtes zum Betrachter reflektiert wird! (1 Punkt)

☐

- f) Wie nennt man den physikalischen Effekt aus Aufgabe e)? (1 Punkt)

☐



### Aufgabe 8: Partikeleffekte und OpenGL-Blending (12 Punkte)

In dieser Aufgabe sollen Feuer und Rauch mithilfe von Partikeln dargestellt werden, die als texturierte Quadrate senkrecht zur Blickrichtung gezeichnet werden. Die Textur wird mit einem Fragment-Shader ausgelesen.

Die Partikeltextur liegt **vormultipliziert** vor. Das heißt, dass die Farbkomponenten jedes Texels bereits mit dem jeweiligen Alpha-Wert multipliziert wurden.

Die OpenGL-Pipeline ist bereits mit folgendem Zustand konfiguriert:

```
glEnable(GL_BLEND);  
glEnable(GL_DEPTH_TEST);  
glDepthMask(GL_FALSE);  
glDepthFunc(GL_LESS);
```

Zur Konfiguration des Blendings stehen Ihnen folgende Konstanten zur Verfügung:

<input type="checkbox"/> 0	GL_ZERO	<input type="checkbox"/> 4	GL_DST_ALPHA
<input type="checkbox"/> 1	GL_ONE	<input type="checkbox"/> 5	GL_ONE_MINUS_DST_ALPHA
<input type="checkbox"/> 2	GL_SRC_ALPHA	<input type="checkbox"/> 6	GL_FUNC_ADD
<input type="checkbox"/> 3	GL_ONE_MINUS_SRC_ALPHA	<input type="checkbox"/> 7	GL_FUNC_SUBTRACT



- a) Zunächst werden nur Feuerpartikel gerendert. Konfigurieren Sie OpenGL so, dass die Feuerpartikel Licht in Richtung des Betrachters emittieren! (2 Punkte)

```
glBlendEquation( ☐ );  
glBlendFunc(sfactor = ☐ , dfactor = ☐ );
```



- b) Warum ist der Tiefentest aktiviert? Warum ist Schreiben in den Tiefenpuffer deaktiviert? (2 Punkte)



- c) Statt Feuerpartikel werden nun Rauchpartikel gerendert, die Licht absorbieren. Die Partikel sind hierzu bereits von hinten nach vorne sortiert. Ändern Sie das Blending entsprechend! (2 Punkte)

```
glBlendFunc(sfactor = ☐ , dfactor = ☐ );
```

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

- d) 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? **(6 Punkte)**

☐

Kreuzen Sie jeweils die passenden Kästchen an! Sie erhalten für jeden vollständig richtigen Block 1.5 Punkte.

**Den Renderablauf verändern:**

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

**OpenGL Blending-Konfigurationen:**

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

**Den Fragment-Shader verändern:**

- ☐ Alpha-Ausgabewert von Feuerpartikeln auf  $(1 - \text{Texel-Alpha})$  setzen.
- ☐ Alpha-Ausgabewert von Feuerpartikeln auf 1 setzen.
- ☐ Alpha-Ausgabewert von Feuerpartikeln auf 0 setzen.
- ☐ Ausgabewerte unverändert beibehalten.

**Vertex-Attribute hinzufügen:**

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



### Aufgabe 9: OpenGL (6 Punkte)



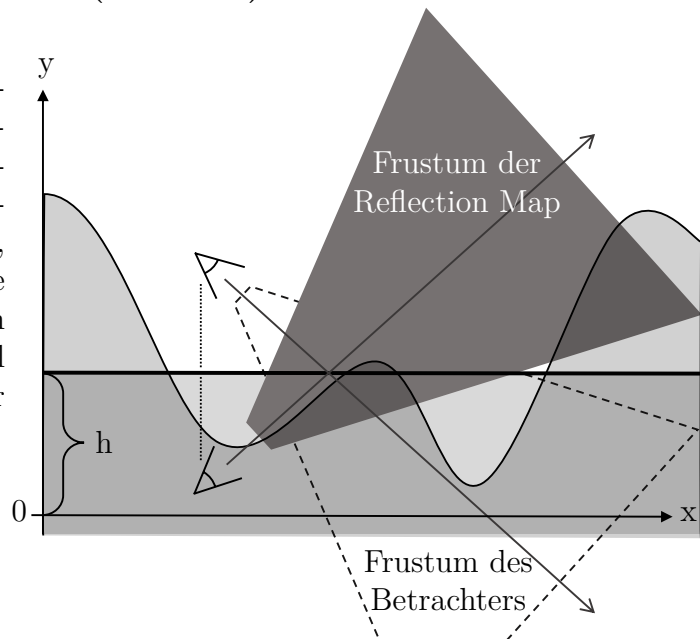
Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen! (6 Punkte)

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



**Aufgabe 10: Reflexionen in OpenGL (4 Punkte)**

Es soll eine Wasseroberfläche mit planarer Reflexion gezeichnet werden. Dazu wird eine Reflection Map gerendert, wie in der nebenstehenden Abbildung dargestellt. Um zu vermeiden, dass in der Reflexion sichtbare Teile der Szene von nicht in der Reflexion sichtbaren Teilen verdeckt werden, soll im Fragment-Shader Clipping an der Wasserebene durchgeführt werden.



Vervollständigen Sie den folgenden Fragment-Shader für das Rendern der Reflection Map! (4 Punkte)



```
...
uniform float h; // Y-Koordinate der Wasserebene im World Space

in vec3 P;       // World-Space-Position des aktuellen Fragments
...

void main(void)
{
    // Implementieren Sie hier das Clipping:

    // Es folgt das spezifische Shading des jeweiligen Objekts
    ...
}
```



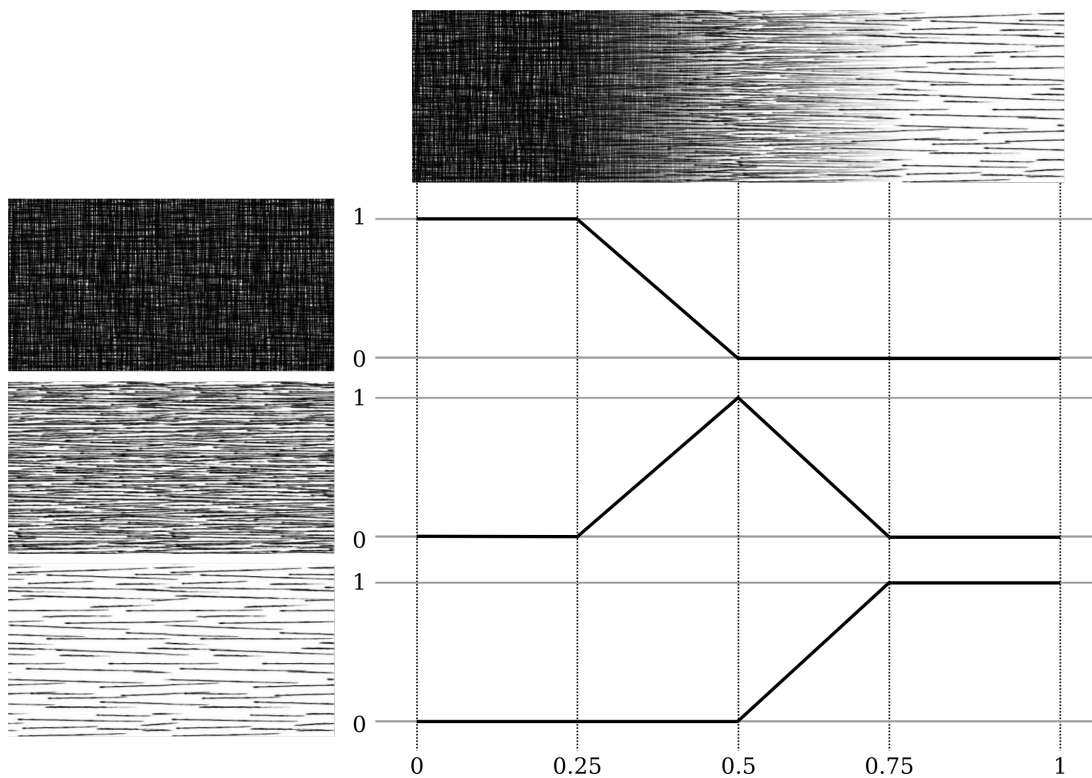
### Aufgabe 11: GLSL-Hatching (8 Punkte)



- a) Geben Sie eine Formel zur linearen Interpolation zwischen den Werten  $y_1 = f(x_1)$  und  $y_2 = f(x_2)$  in Abhängigkeit von  $x \in [x_1, x_2]$ ,  $x_2 > x_1$  an! **(2 Punkte)**

- b) Nun sollen Sie eine einfache Form von Schraffur (engl. *hatching*) im Fragment-Shader implementieren. Dabei werden pro Fragment mehrere Eingabetexturen gemischt. Die Interpolationsgewichte hängen von der Helligkeit der Oberfläche ab.

Die Eingabetexturen, Interpolationsgewichte und die Ausgabertextur sind im folgenden Bild dargestellt:



Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Vervollständigen Sie die Funktion `get_hatched_color()`! (6 Punkte)



```
...
uniform sampler2D dunkel; // dunkle Textur
uniform sampler2D mittel; // mittlere Textur
uniform sampler2D hell;   // helle Textur

in vec2 tex_coord; // Texturkoordinate
...

vec4 get_hatched_color(float h) // Helligkeit h liegt in [0, 1].
{
    vec4 t_d = texture(dunkel, tex_coord);
    vec4 t_m = texture(mittel, tex_coord);
    vec4 t_h = texture(hell,   tex_coord);

    }
}
```



### Aufgabe 12: Bézierkurven (13 Punkte)



a) Was versteht man unter affiner Invarianz bei Bézierkurven? (2 Punkte)

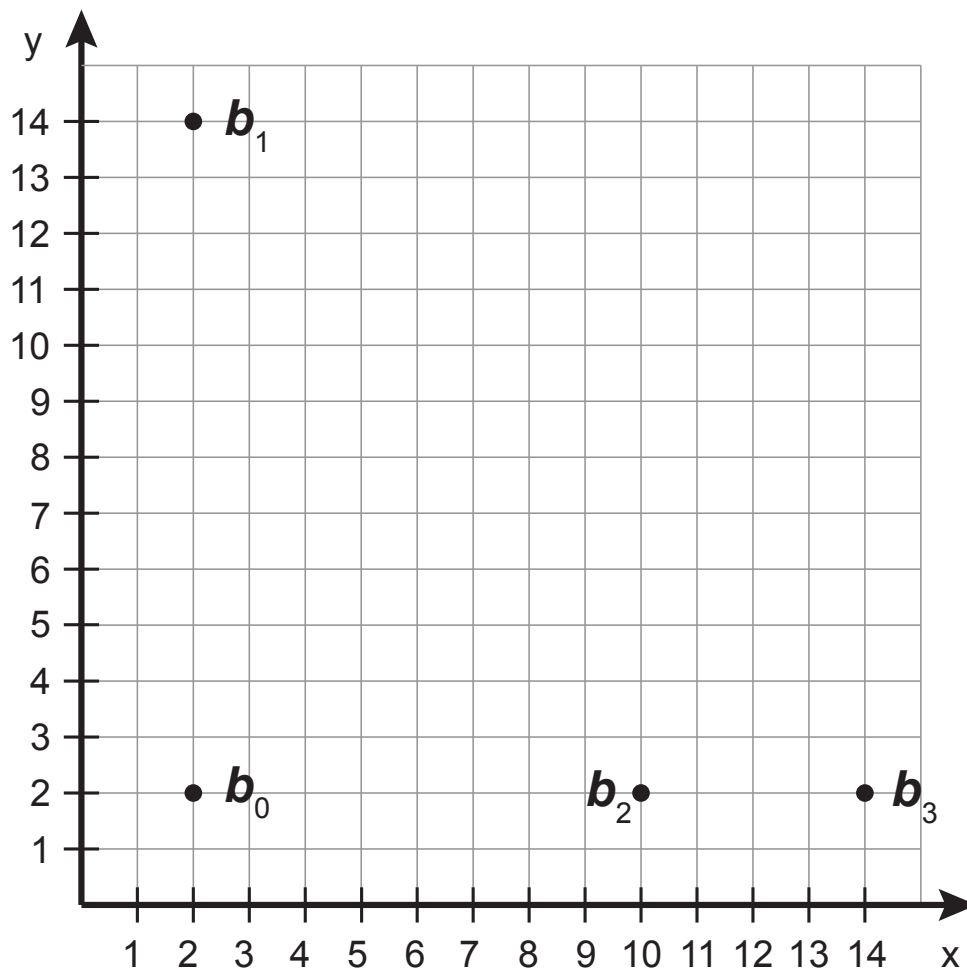
b) Gegeben sei die Bézierkurve  $\mathbf{b}(u) = \sum_{i=0}^3 \mathbf{b}_i B_i^3(u)$  mit den kubischen Bernsteinpolynomen  $B_i^3(u)$  und den Kontrollpunkten  $\mathbf{b}_i$ .



I) Werten Sie die Bézierkurve zeichnerisch an der Stelle  $u = \frac{1}{2}$  mit dem de Casteljau-Algorithmus aus! Zeichnen Sie alle Punkte der Konstruktion, inklusive dem endgültigen Kurvenpunkt, ein! (4 Punkte)



II) Skizzieren Sie den Verlauf der Bézierkurve! (2 Punkte)

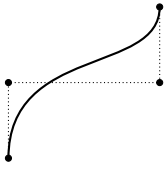


Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

- c) 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! **(5 Punkte)**

☐

Kurve	Ja	Nein	Begründung
	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	