Institut für Visualisierung und Datenanalyse
Lehrstuhl für Computergrafik
Prof. Dr.-Ing. Carsten Dachsbacher

Karlsruher Institut für Technologie

Nachklausur Computergrafik
SS 2019

09. September 2019

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 24 Seiten (12 Blätter) mit 12 Aufgaben.
- Es sind **keine Hilfsmittel** zugelassen.
- Sie haben **90 Minuten** Bearbeitungszeit.
- Schreiben Sie Ihre Matrikelnummer oben auf jedes bearbeitete Aufgabenblatt.
- Schreiben Sie Ihre Lösungen auf die Aufgabenblätter. Bei Bedarf können Sie weiteres Papier anfordern.
- Wir akzeptieren auch englische Antworten.

Aufgabe	1	2	3	4	5	6	7	8	9	10	11	12	**Gesamt**
Erreichte Punkte													
Erreichbare Punkte	11	13	21	16	14	17	16	12	8	31	11	10	**180**

Note

Aufgabe 1: Farbe und Perzeption (11 Punkte)

a) Welche Funktionen benötigen Sie, um die Tristimulus-Werte eines Farbraums zu einem Spektrum $s(\lambda)$ zu berechnen? Wie werden daraus die Tristimulus-Werte berechnet? **(3 Punkte)**

b) Ein Farbeindruck kann nicht eindeutig in ein Spektrum umgewandelt werden. Begründen Sie warum und benennen Sie dieses Phänomen! **(3 Punkte)**

c) Wie können Sie sich die Trennung von Luminanz und Chrominanz bei der Bildkompression zunutze machen? Begründen Sie, warum dies eine sinnvolle Strategie ist! **(3 Punkte)**

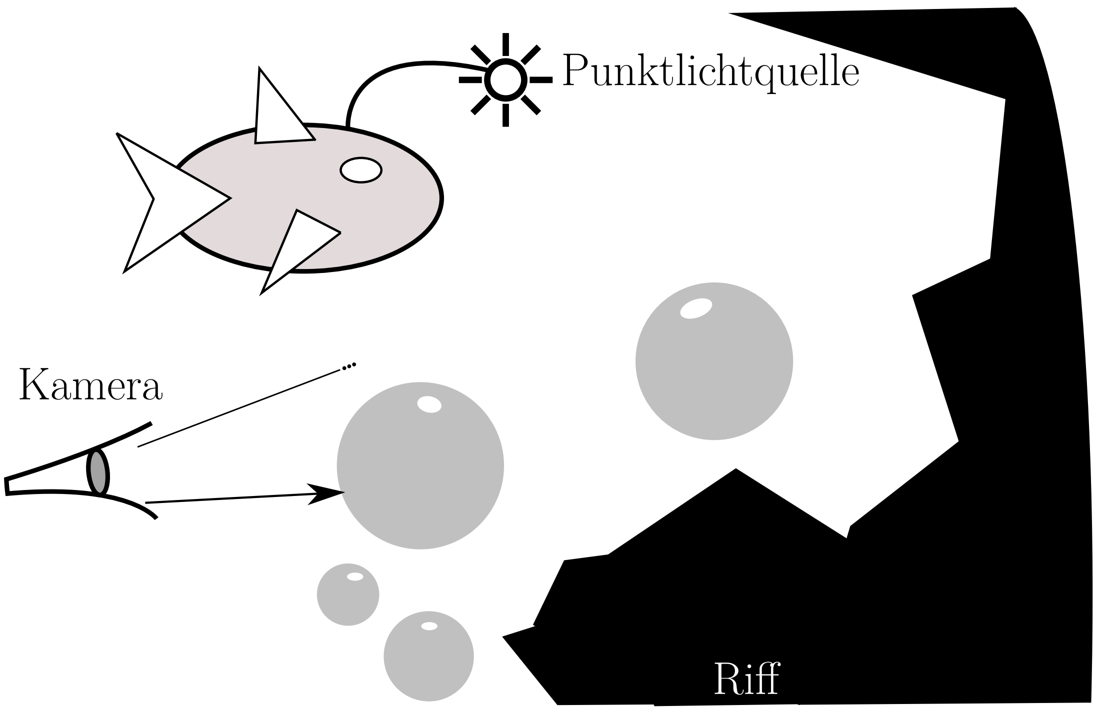
d) Wie ändert sich der wahrgenommene Kontrast zwischen zwei Pixeln, wenn ihre Leuchtdichten um den gleichen Wert erhöht werden? Begründen Sie Ihre Antwort! **(2 Punkte)**

Aufgabe 2: Ray Tracing (13 Punkte)

Unter Wasser beleuchtet ein Armflosser-Fisch mit seiner biolumineszierenden Angel seine Umgebung und aufsteigende Blasen. Das Wasser hat Brechungsindex $\eta_1 = 1.2$, das Gas in den Blasen Brechungsindex $\eta_2 = 1$.

Die Blasen haben Reflexionskoeffizienten $k_r = 0$ und auch die ambienten, diffusen und spekularen Koeffizienten sind 0. Das Riff und der Fisch weisen nur diffuse Reflexion auf.

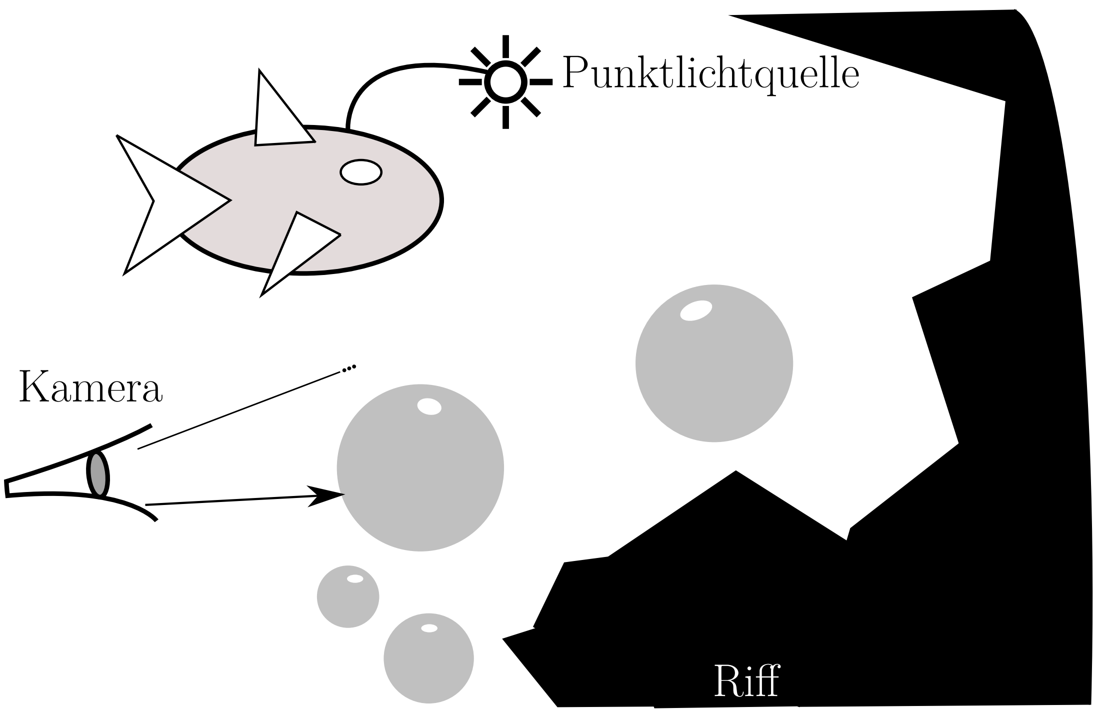
a) Setzen Sie die beiden eingezeichneten Primärstrahlen wie beim Whitted-style Ray Tracing fort! Zeichnen Sie nur die Strahlen ein, die der Raytracer erzeugen muss, und bezeichnen Sie alle Strahlen mit P für Primär-, T für Transmissions-, R für Reflexions- und S für Schattenstrahlen! Wenn Sie einen Fehler korrigieren müssen, nutzen Sie dafür die leere Vorlage auf der nächsten Seite und kennzeichnen Sie dies! **(6 Punkte)**



b) Bis zu welcher Rekursionstiefe muss der Raytracer Strahlen verfolgen, bis durch die Blase(n) das Riff beleuchtet erscheint? Der Primärstrahl entspricht Rekursionstiefe 0. **(2 Punkte)**

Matrikelnummer: ______________________

c) Warum ist die Rekursionstiefe in dieser Szene beschränkt? Welche Änderung würde die Beschränkung aufheben? Nennen Sie eine Möglichkeit, die Sie in der Vorlesung kennengelernt haben, um das Terminieren des Raytracers sicherzustellen! **(5 Punkte)**

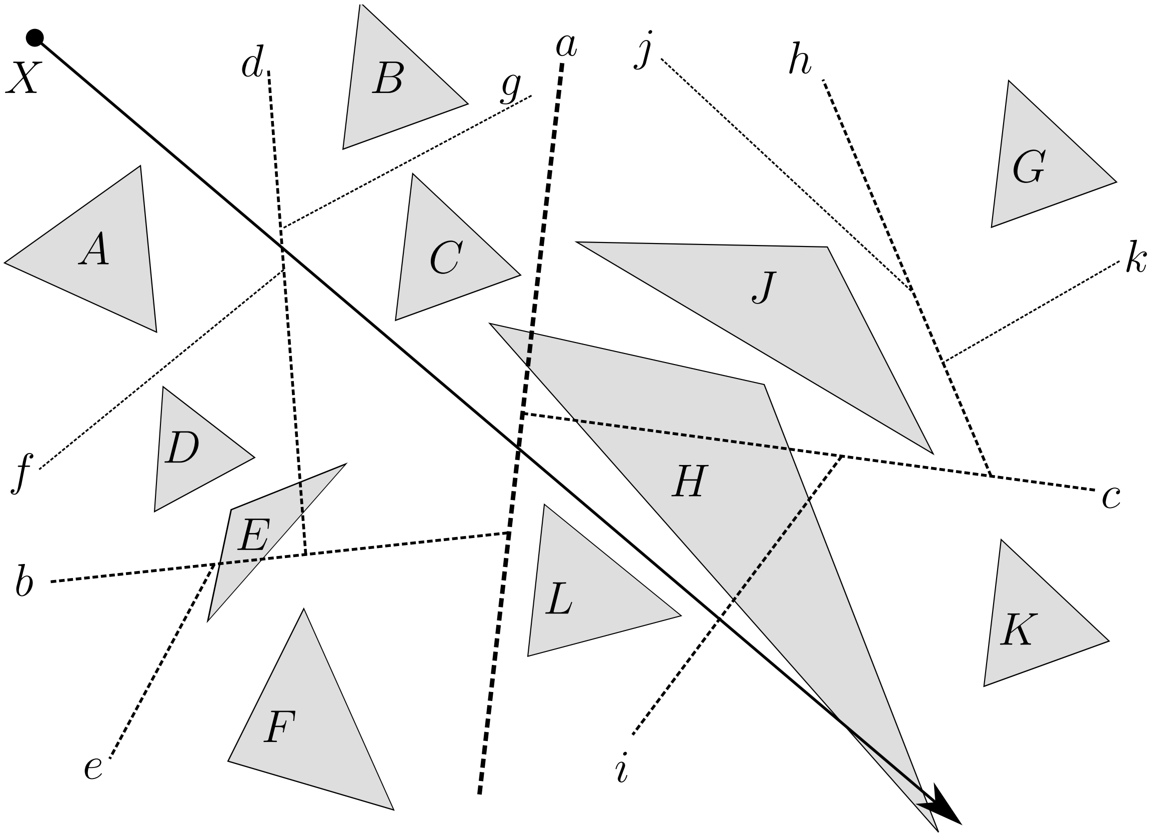


Aufgabe 3: Beschleunigungsstrukturen (21 Punkte)

a) Nennen Sie zwei Hüllkörperformen (Bounding Volumes) und für jede einen Vorteil gegenüber der anderen! **(4 Punkte)**

b) Beim Aufbau einer hierarchischen Hüllkörperstruktur kann es sich lohnen, die Menge von Primitiven eines Knotens nicht weiter zu unterteilen.
Geben Sie die Formel der aus der Vorlesung bekannten Kostenheuristik an und wie man mit ihrer Hilfe entscheidet, ob die Unterteilung eine Beschleunigung von Schnitttests erwarten lässt! Geben Sie für jede Komponente der Heuristik deren Bedeutung an! **(8 Punkte)**

c) In der Abbildung sehen Sie eine Menge von Primitiven, die mit einer hierarchischen Beschleunigungsstruktur unterteilt wurde.

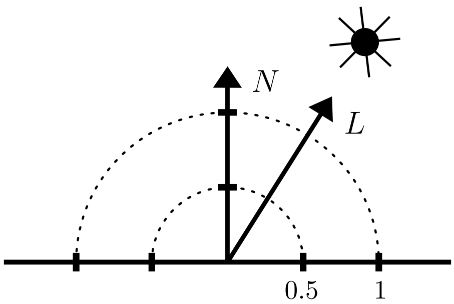


i) Wie heißt diese Beschleunigungsstruktur? Woran erkennen Sie sie? **(2 Punkte)**

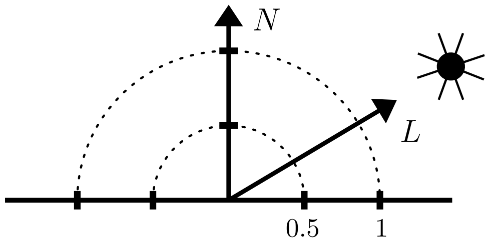
ii) Traversieren Sie die Beschleunigungsstruktur für den eingezeichneten Strahl, um den nächsten Schnittpunkt zu X zu finden! Geben Sie an, in welcher Reihenfolge Schnitttests mit den Splitebenen $a, b, c, \ldots$ und Primitiven $A, B, C, \ldots$ durchgeführt werden, bis der Schnittpunkt mit H feststeht! Es wird kein Mailboxing verwendet. Falls mehrere Reihenfolgen möglich sind, geben Sie eine beliebige davon an! **(7 Punkte)**

Aufgabe 4: Beleuchtungsberechnung (16 Punkte)

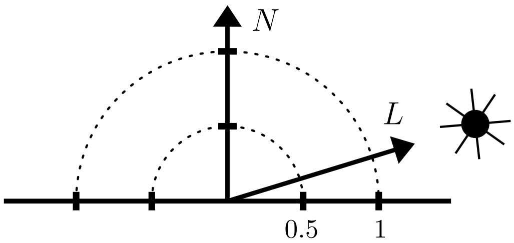
a) Skizzieren Sie das Ergebnis des Phong-Beleuchtungsmodells jeweils für die verschiedenen Konfigurationen! k_a, k_d und k_s sind dabei die Koeffizienten der ambienten, diffusen und spekularen Komponenten, n ist der Phong-Exponent. Wenn Sie einen Fehler korrigieren müssen, nutzen Sie dafür die beiden leeren Vorlagen und kennzeichnen Sie dies! **(8 Punkte)**



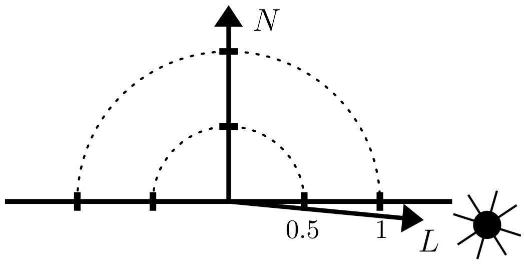
i) $k_a = 0, k_d = 1, k_s = 0, n = 10$



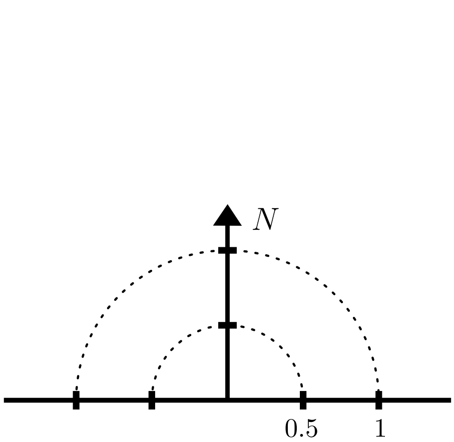
ii) $k_a = 0.6, k_d = 0, k_s = 0.4, n = 1000$

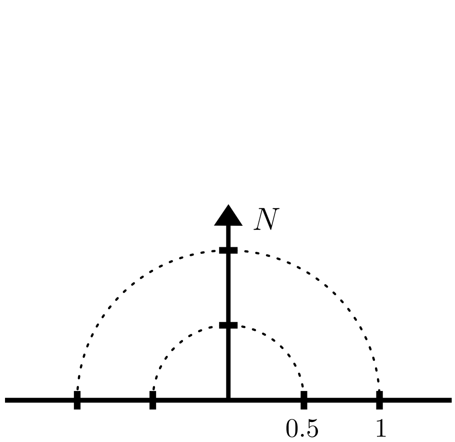


iii) $k_a = 0, k_d = 1, k_s = 0, n = 5$



iv) $k_a = 0.2, k_d = 0.3, k_s = 0.5, n = 20$





b) Zur glatteren Darstellung von Dreicksnetzen wird üblicherweise eine interpolierte Normale für die Beleuchtungsberechnung verwendet. Dazu werden an den Eckpunkten gespeicherte Normalen (Vertex-Normalen) mit baryzentrischen Koordinaten linear komponentenweise interpoliert.

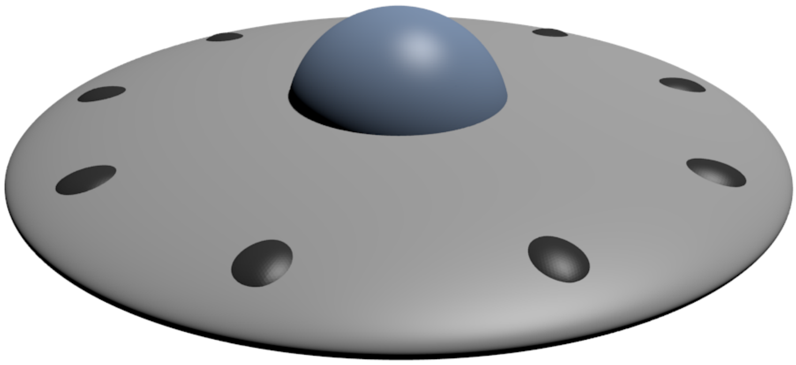
i) Wie heißt dieses Verfahren? **(1 Punkt)**

ii) Was ist bei der Interpolation zu beachten? **(1 Punkt)**

iii) Wie können mit diesem Verfahren gleichzeitig scharfe und glatte Kanten dargestellt werden? **(3 Punkte)**

iv) Wie können für ein gegebenes Dreiecksnetz Vertex-Normalen für glatte Schattierung berechnet werden? **(3 Punkte)**

Aufgabe 5: Transformationen (14 Punkte)



Ein Bild der oben gezeigten Szene soll mit Hilfe von Raytracing berechnet werden. Das Modell besteht aus 11 transformierten (verschobenen und skalierten) Einheitskugeln. Die Einheitskugel ist definiert durch $\sqrt{x^2 + y^2 + z^2} = 1$.

a) Geben Sie eine homogene Transformationsmatrix $S \in \mathbb{R}^{4\times 4}$ an, die eine anisotrope Skalierung um die Faktoren $s_x,\ s_y,\ s_z$ in x-, y-, und z-Richtung beschreibt! **(2 Punkte)**

b) Geben Sie eine homogene Transformationsmatrix $T \in \mathbb{R}^{4\times 4}$ an, die eine Verschiebung um $t_x,\ t_y,\ t_z$ in x-, y-, und z-Richtung beschreibt! **(2 Punkte)**

Matrikelnummer: ______________________

c) Geben Sie eine Transformationsmatrix $M \in \mathbb{R}^{4\times 4}$ an, die eine Einheitskugel in einen Ellipsoid mit Mittelpunkt $\begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$ und Radien $g_x,\ g_y,\ g_z$ transformiert! **(2 Punkte)**

d) Zur einfacheren Schnittpunktberechnung kann ein Strahl $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ in das lokale Koordinatensystem der mit M transformierten Einheitskugel transformiert werden. Der transformierte Strahl sei $\mathbf{r}'(t') = \mathbf{o}' + t' \cdot \mathbf{d}'$.

i) Geben Sie an, wie $\mathbf{o}'$ und $\mathbf{d}'$ aus $\mathbf{o}$ und $\mathbf{d}$ berechnet werden! Wie ist jeweils die homogene Komponente von $\mathbf{o}$ und $\mathbf{d}$ zu wählen? **(4 Punkte)**

ii) Gegeben sei eine Funktion zur Schnittberechnung des Strahls $\mathbf{r}'(t')$ mit einer Einheitskugel. Wie bestimmen Sie damit den Schnittpunkt von $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ mit dem Ellipsoid und den dazugehörigen Strahlparameter t? **(4 Punkte)**

Aufgabe 6: Texturen und Baryzentrische Koordinaten (17 Punkte)

Gegeben seien die Punkte $\mathbf{P}_1 = (x_1, y_1, z_1)^T, \mathbf{P}_2 = (\dots)^T, \mathbf{P}_3 = (\dots)^T$ eines Dreiecks, sowie ein Vektor von baryzentrischen Koordinaten $\Lambda = (\lambda_1, \lambda_2, \lambda_3)$.

a) Stellen Sie die Matrix $\mathbf{M}$ auf, mit welcher der zu $\boldsymbol{\Lambda}$ gehörige Punkt auf dem Dreieck als $\mathbf{P} = \mathbf{M}\boldsymbol{\Lambda}$ dargestellt werden kann! **(3 Punkte)**

b) Gegeben sind Vektoren $\mathbf{T}$ und $\mathbf{B}$, sodass $\lambda_2 = \mathbf{T} \cdot (\mathbf{P} - \mathbf{P}_1)$ und $\lambda_3 = \mathbf{B} \cdot (\mathbf{P} - \mathbf{P}_1)$. Wie lässt sich λ_1 für den Punkt $\mathbf{P}$ bestimmen, ohne $\mathbf{P}_2$ und $\mathbf{P}_3$ zu verwenden? **(2 Punkte)**

c) Die *baryzentrischen Koordinaten* λ_2' und λ_3' eines Punktes $\mathbf{P}'$ ergeben sich aus einer Verschiebung von λ_2 und λ_3 von $\mathbf{P}$ um $\Delta\lambda_2$ und $\Delta\lambda_3$. Bestimmen Sie $\mathbf{P}'$! **(6 Punkte)**

d) Für welche wichtige Texture Mapping-Technik betrachtet man üblicherweise die Texturkoordinaten von Nachbarpixeln? Was wird aus deren Differenzen bestimmt? **(2 Punkte)**

e) Warum lassen sich Aliasing-Artefakte durch *Vorfilterung* von Texturen vermeiden und welches Theorem kann dann erfüllt werden? **(4 Punkte)**

Aufgabe 7: OpenGL Pipeline (16 Punkte)

Diese Aufgabe behandelt die OpenGL-Grafik-Pipeline.

a) Erläutern Sie kurz die Idee des *Double Buffering*! **(3 Punkte)**

b) In welchem Koordinatensystem findet das Clipping von Grafikprimitiven statt? Nennen Sie zwei Gründe, warum das Clipping in genau diesem Koordinatensystem stattfindet! **(4 Punkte)**

c) Nennen Sie die drei programmierbaren Shader in der OpenGL-Grafik-Pipeline, deren Programmierung in der Vorlesung besprochen wurde (also ohne Hardware-Tesselierung) und ordnen Sie diese anhand ihrer Reihenfolge in der Pipeline. Nennen Sie zu jedem Shader die hauptsächliche Funktion, die dieser erfüllen *muss*! **(6 Punkte)**

1) Name:

   Funktion:

2) Name:

   Funktion:

3) Name:

Funktion:

d) Nennen Sie drei Verarbeitungsschritte, die nicht frei programmierbar sind und die nach dem letzten Shader zur Anwendung kommen können! **(3 Punkte)**

Aufgabe 8: Blending (12 Punkte)

Mit Blending können vom Fragment-Shader ausgegebene Farbwerte (source) mit bereits im Framebuffer stehenden (destination) kombiniert werden. OpenGL wird nun für das Zeichnen semitransparenter Objekte konfiguriert:

```
glEnable(GL_BLEND);
glBlendEquation(GL_FUNC_ADD);
glBlendFunc(GL_SRC_COLOR, GL_ONE_MINUS_SRC_COLOR);
```

a) Geben Sie an, wie Blau- und Alpha-Komponente B bzw. A der Ergebnisfarbe aus der Source-Farbe $c_s = (R_s, G_s, B_s, A_s)$ und Destination-Farbe $c_d = (R_d, G_d, B_d, A_d)$ berechnet werden! **(3 Punkte)**

b) Zeigen Sie, dass diese Blending-Konfiguration nicht kommutativ ist! **(5 Punkte)**

c) Welche Besonderheit für das Zeichnen von Primitiven folgt aus Aufgabenteil b)? **(1 Punkt)**

Matrikelnummer: ____________________

d) Geben Sie eine kommutative Blending-Konfiguration an! Nennen Sie ein Beispiel für deren Verwendung! **(3 Punkte)**

```
glEnable(GL_BLEND);
```

Aufgabe 9: Stenciltest (8 Punkte)

In dieser Aufgabe soll ein Objekt mit Umrandung gezeichnet werden. Dazu benutzen Sie Maskierung mit dem Stencilbuffer. Geben Sie dazu die Konfiguration des Stenciltests in den Aufgabenteilen a) und b) an. Das Objekt wird dann in zwei Durchgängen wie folgt gezeichnet:

1. Konfiguriere den Stenciltest wie in Teilaufgabe a).
2. Zeichne das Objekt.
3. Konfiguriere den Stenciltest wie in Teilaufgabe b).
4. Skaliere das Objekt uniform etwas größer und zeichne es erneut. Dabei wird ein Fragment-Shader verwendet, der die Farbe der Umrandung ausgibt.

Zur Erinnerung:

```
glStencilFunc( stencilfunc, refstencil, mask )
glStencilOp( fail, zfail, zpass )
```

Mögliche Stencil-Funktionen sind: `GL_ALWAYS`, `GL_NEVER`, `GL_EQUAL`, `GL_NOT_EQUAL`, `GL_LESS`.

Die möglichen Stencil-Ops wählen Sie aus: `GL_KEEP`, `GL_REPLACE`, `GL_INCR`, `GL_DECR`.

a) Geben Sie die Stenciltest-Konfiguration für den ersten Durchgang an! **(4 Punkte)**

```
glStencilFunc(              ,              ,              );
glStencilOp  (              ,              ,              );
```

b) Geben Sie die Stenciltest-Konfiguration für den zweiten Durchgang an! **(4 Punkte)**

```
glStencilFunc(              ,              ,              );
glStencilOp  (              ,              ,              );
```

Aufgabe 10: Phong Shading und Spiegelung mit Fresnel-Effekt (31 Punkte)

Sie sollen in dieser Aufgabe GLSL-Shader implementieren, die ein Objekt mit Phong Shading und zusätzlicher perfekter Spiegelung darstellen. In einem ersten Rendering-Durchgang wird das Objekt mit Schattierung aufgrund einer Lichtquelle gezeichnet. Im zweiten Durchgang wird die perfekte Spiegelung auf der Oberfläche mit einer Cube Environment Map dargestellt. Die finale Farbe soll mittels Blending aus den Ausgaben der beiden Rendering-Durchgänge kombiniert werden, sodass die Stärke der Reflexion mit dem Fresnel-Term f multipliziert wird, und die Schattierung mit dem Faktor $(1 - f)$ multipliziert wird.

Die Schattierung können Sie mit der Funktion `computeShading(...)` für einen Oberflächenpunkt an der Position **P**, mit Normale **N**, Richtung zur Lichtquelle **L** und Reflexionsvektor **R** berechnen. Alle Vektoren werden in Weltkoordinaten angegeben. **R** ist die Reflexion der Blickrichtung an der Oberfläche und kann mit der GLSL-Funktion `reflect(...)` berechnet werden. **R** wird ebenfalls für den Zugriff auf die Environment Map verwendet.

Den Fresnel-Term berechnet die Funktion `fresnel(in vec3 N, in vec3 R)` für eine Normale **N** und einen Reflexionsvektor **R**.

Beide Rendering-Durchgänge teilen sich einen Vertex Shader und verwenden jeweils einen eigenen Fragment Shader.

a) Ergänzen Sie zunächst die OpenGL Render-States so, dass die Oberfläche korrekt in beiden Rendering-Durchgängen rasterisiert wird und das Blending funktioniert! **(4 Punkte)**

Durchgang 1:

```
glEnable( GL_DEPTH_TEST );
glDepthFunc( GL_LESS );
glDisable( GL_BLEND );
renderObject();
```

Durchgang 2:

```
glEnable( GL_DEPTH_TEST );
glDepthFunc( [          ] );
glEnable( GL_BLEND );
glBlendFunc( [          ], [          ] );
renderObject();
```

b) Ergänzen Sie nun die GLSL-Shader! Übergeben Sie nicht mehr Daten in Aus-/Eingabe-Variablen als notwendig! **(27 Punkte)** (je 9 Punkte pro Shader)

Hilfsfunktionen:

```
// berechnet den Fresnel-Term f mit 0<=f<=1
float fresnel( in vec3 N, in vec3 R ) { ... }
// wertet das Schattierungmodell aus und gibt einen RGB-Vektor zurück
vec3 computeShading( in vec3 P, in vec3 N, in vec3 L, in vec3 R ) { ... }
```

Vertex Shader:

```
uniform mat4 matrixMVP,  // Model-View-Projection Matrix
    matrixMW,    // Transformation von Vertices aus Modell- in Weltkoordinaten
    matrixNrml; // Transformation von Normalen aus Modell- in Weltkoordinaten
uniform vec3 lightPos;   // Lichtquellenposition in Weltkoordinaten
uniform vec3 camPos;     // Kameraposition in Weltkoordinaten

in vec4 in_position;
in vec3 in_normal;

// definieren Sie hier NUR DIE NOTWENDIGEN Ausgabe-Variablen



void main() {
    gl_Position = matrixMVP * in_position;

    // berechnen Sie hier die NOTWENDIGEN Ausgabewerte



}
```

Matrikelnummer: ____________________

Hilfsfunktionen (Wiederholung der Deklarationen auf der vorigen Seite):

```
// berechnet den Fresnel-Term f mit 0<=f<=1
float fresnel(in vec3 N, in vec3 R ) { ... }
// wertet das Schattierungmodell aus und gibt einen RGB-Vektor zurück
vec3 computeShading( in vec3 P, in vec3 N, in vec3 L, in vec3 R ) { ... }
```

Fragment Shader (Durchgang 1):

```
// definieren Sie hier NUR DIE NOTWENDIGEN Eingabe-Variablen
```

```
out vec4 out_color;

void main() {

}
```

Hilfsfunktionen (Wiederholung der Deklarationen auf der vorigen Seite):

```
// berechnet den Fresnel-Term f mit 0<=f<=1
float fresnel(in vec3 N, in vec3 R ) { ... }
// wertet das Schattierungmodell aus und gibt einen RGB-Vektor zurück
vec3 computeShading( in vec3 P, in vec3 N, in vec3 L, in vec3 R ) { ... }
```

Fragment Shader (Durchgang 2):

```
// Cube Environment Map
uniform samplerCube samCube;

// definieren Sie hier NUR DIE NOTWENDIGEN Eingabe-Variablen

out vec4 out_color;

void main() {

}
```

Matrikelnummer: ____________________

Aufgabe 11: Prozedurale Modellierung (11 Punkte)

Feuer lässt sich prozedural mit Hilfe von Rausch- und Turbulenzfunktionen modellieren.

a) Geben Sie drei Vorteile der prozeduralen Modellierung *zur Laufzeit* gegenüber händisch modellierten und gespeicherten Texturen an! **(3 Punkte)**

b) Nennen Sie drei Eigenschaften, die eine Rauschfunktion erfüllen muss! **(3 Punkte)**

c) Geben Sie den Ausdruck einer Turbulenzfunktion an! **(3 Punkte)**

d) Wie ändert sich der Berechnungsaufwand der Turbulenzfunktion, wenn doppelt so hohe Frequenzen erzeugt werden sollen? **(2 Punkte)**

Aufgabe 12: Bézier-Kurven (10 Punkte)

Gegeben sind die folgenden Bézier-Kurven $a(u) = \sum_{i=0}^{3} a_i B_i(u)$ und $b(u) = \sum_{i=0}^{3} b_i B_i(u)$ (siehe auch Skizze):

$$a_0 = \begin{pmatrix} 2 \\ 6 \end{pmatrix},\ a_1 = \begin{pmatrix} 5 \\ 3 \end{pmatrix},\ a_2 = \begin{pmatrix} 12 \\ 1 \end{pmatrix},\ a_3 = \begin{pmatrix} 14 \\ 6 \end{pmatrix}.$$

$$b_0 = \begin{pmatrix} \cdots \\ \cdots \end{pmatrix},\ b_1 = \begin{pmatrix} \cdots \\ \cdots \end{pmatrix},\ b_2 = \begin{pmatrix} 26 \\ 1 \end{pmatrix},\ b_3 = \begin{pmatrix} 28 \\ 6 \end{pmatrix}.$$

a) Bestimmen Sie b_0 und b_1 so, dass in a_3 ein C^1-stetiger Übergang von $a(u)$ nach $b(u)$ entsteht! **(4 Punkte)**.

b) Werten Sie die Kurve b für $u = 0.5$ grafisch mit dem de Casteljau-Algorithmus aus! Skizzieren Sie anschließend b mithilfe des ausgewerteten Punkts! **(6 Punkte)**.