Karlsruher Institut für Technologie

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

Hauptklausur Computergrafik
WS 2019/2020

09. März 2020

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 18 Seiten (9 Blätter) mit 11 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	**Gesamt**
Erreichte Punkte												
Erreichbare Punkte	19	12	12	29	12	15	16	16	30	9	10	**180**

Note

Aufgabe 1: Farbe und Perzeption (19 Punkte)

a) Geben Sie die Koordinaten (x, y) im Chromatizitätsdiagramm einer Farbe $\mathbf{C} = (X, Y, Z)$ im XYZ-Farbraum an! **(3 Punkte)**

$x =$

$y =$

b) Was ist der Gamut eines Ausgabegerätes? **(3 Punkte)**

c) Was sind Spektralfarben? Warum können sie nicht auf einem RGB-Monitor dargestellt werden? **(4 Punkte)**

d) Welche Eigenschaft der menschlichen Wahrnehmung wird beim Dithering ausgenutzt? **(4 Punkte)**

e) Ein Monitor mit Gamma-Wert γ erzeugt für jeden Pixel $(i, j) \in [0, 3839] \times [0, 2159]$ aus dem Wert x_{ij} im Framebuffer die Intensität I_{ij}. Wie können Sie trotz unbekanntem γ eine Fläche darstellen, die beim Betrachter den Eindruck einer Intensität von $I = 0.5 I_{max}$ hervorruft? Wozu ist das nützlich? **(5 Punkte)**

Matrikelnummer: _____

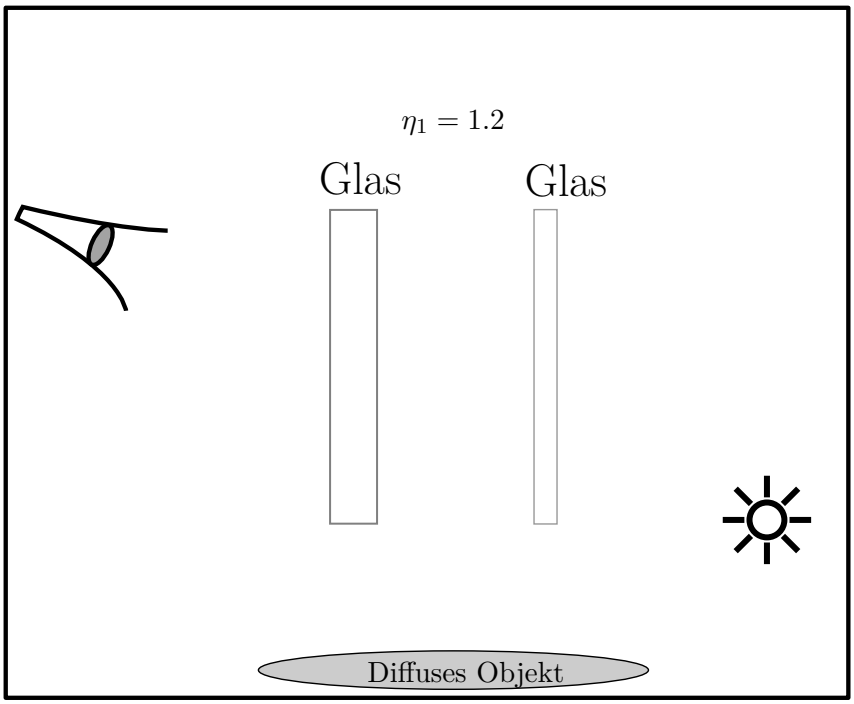
Aufgabe 2: Whitted-Style Raytracing (12 Punkte)

Im Folgenden soll mittels Whitted-Style Raytracing und Phong-Beleuchtungsmodell ein Bild berechnet werden. Dabei befinden sich zwei Glasscheiben mit Brechungsindex $\eta_1 = 1.2$ über einem diffusen Objekt mit $k_d > 0, k_r = 0$ (siehe Abbildung). Außerhalb des Glases ist der Brechungsindex $\eta_0 = 1$. Die Glasscheiben haben eine Reflektivität $k_r > 0$ und Transmissivität $k_t > 0$. Der ambiente, diffuse und spekulare Koeffizient der Glasscheibe ist jeweils null ($k_a = 0, k_d = 0, k_s = 0$).

a) Erzeugen Sie zwei Strahlen ausgehend von der Kamera und verfolgen Sie diese, sodass:

- Für Einen die minimale Anzahl an Sekundärstrahlen verfolgt wird, obwohl ein Objekt der Szene getroffen wird!
- Für den Anderen eine unendliche Anzahl an Sekundärstrahlen verfolgt werden müsste! (Zeichnen Sie bis einschließlich Rekursionstiefe 3)

Zeichnen Sie die rekursiv verfolgten Strahlen und beschriften Sie diese mit **P** für Primärstrahlen, **S** für Schattenstrahlen, **T** für Transmissionsstrahlen und **R** für Reflektionsstrahlen! **(8 Punkte)**



b) Nennen Sie die zwei Möglichkeiten aus der Vorlesung, um Strahlenbäume mit unendlicher Tiefe beim Whitted-Style Raytracing zu verhindern! **(4 Punkte)**

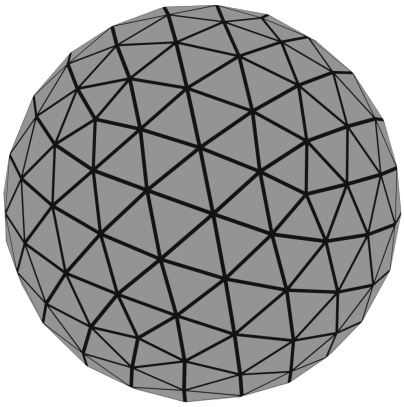
Aufgabe 3: Phong-Beleuchtungsmodell und Phong-Shading (12 Punkte)

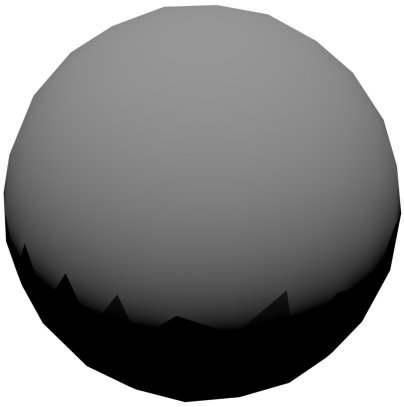
In dieser Aufgabe wird das Phong-Modell verwendet, um die Beleuchtung von Oberflächen durch Lichtquellen zu berechnen.

a) Wie ändert sich die Beleuchtung einer glänzenden Oberfläche, wenn der Phong-Exponent *erhöht* wird? **(2 Punkte)**

b) Für eine Intensität I_L sowie Koeffizienten k_a, k_d und k_s hat das Phong-Modell ein Maximum $I_{\max}$. Geben Sie $I_{\max}$ und die dazugehörige Konfiguration der Vektoren **L**, **V** und **N** an! **(6 Punkte)**

c) Zur approximativen Darstellung einer Kugel soll ein Dreiecksnetz (links) verwendet werden, dessen Beleuchtung mit Whitted-Style Raytracing, Phong-Shading und dem Phong-Beleuchtungsmodell berechnet wird (rechts).





Warum treten trotz Phong-Shading noch Unstetigkeiten in der Schattierung auf? **(4 Punkte)**

Matrikelnummer: _____

Aufgabe 4: Räumliche Datenstrukturen (29 Punkte)

Gegeben sei ein Strahl $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ mit $\mathbf{o}, \mathbf{d} \in \mathbb{R}^3$ und $\|\mathbf{d}\| = 1$, sowie ein achsenorientierter Hüllquader (AABB) mit Begrenzungspunkten $\mathbf{p} = (p_x, p_y, p_z)$ und $\mathbf{q} = (q_x, q_y, q_z)$, wobei $p_i < q_i, i \in \{x, y, z\}$. Der Strahl soll nun mit den Begrenzungsebenen der AABB geschnitten werden, um zu bestimmen, ob ein Schnittpunkt mit der AABB besteht.

a) Seien $d_x, d_y, d_z > 0$.

i) Wie können Sie effizient den Strahlparameter t_{p_x} bestimmen, der die Entfernung von $\mathbf{o}$ zum Schnittpunkt mit der durch p_x festgelegten yz-Ebene beschreibt? **(4 Punkte)**

$t_{p_x} =$

ii) Seien $t_{p_y}, t_{p_z}, t_{q_x}, t_{q_y}$ und t_{q_z} analog bestimmt. Durch eine Fallunterscheidung bestimmt man beim Strahlschnitt mit einer AABB, ob tatsächlich ein Schnittpunkt existiert. Geben Sie für die folgenden Fälle jeweils die Bedingung zu deren Feststellung an, sowie gegebenenfalls den gefragten Schnittpunkt! Sie dürfen bei Fall 2 und 3 davon ausgehen, dass die vorherigen Bedingungen nicht erfüllt sind.

Tipp: Sie dürfen Skizzen anfertigen, diese werden nicht bewertet. Die Variablen t_{near} und t_{far} können Ihnen helfen:

$$t_{\text{near}} = \max(t_{p_x}, t_{p_y}, t_{p_z}),$$
$$t_{\text{far}} = \min(t_{q_x}, t_{q_y}, t_{q_z}).$$

Fall 1: Der Strahl schneidet die Box nicht: **(3 Punkte)**

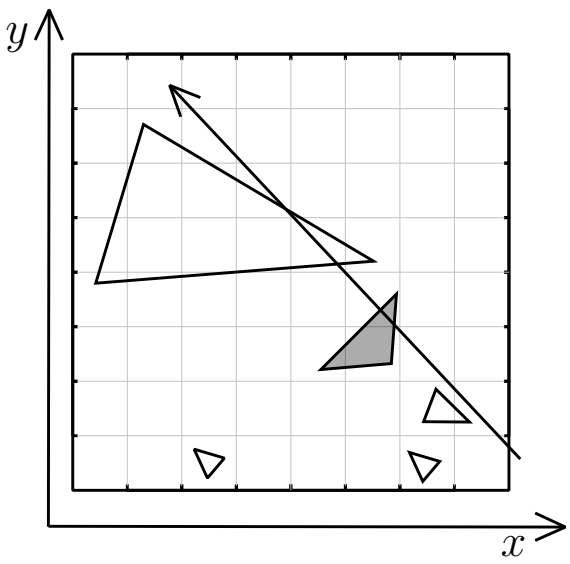
Fall 2: Der Strahl schneidet die Box nur hinter dem Strahlursprung (also nicht in Strahlrichtung): **(4 Punkte)**

Fall 3: Der Strahlursprung liegt außerhalb der Box und der Strahl schneidet die Box in Strahlrichtung: **(6 Punkte)**

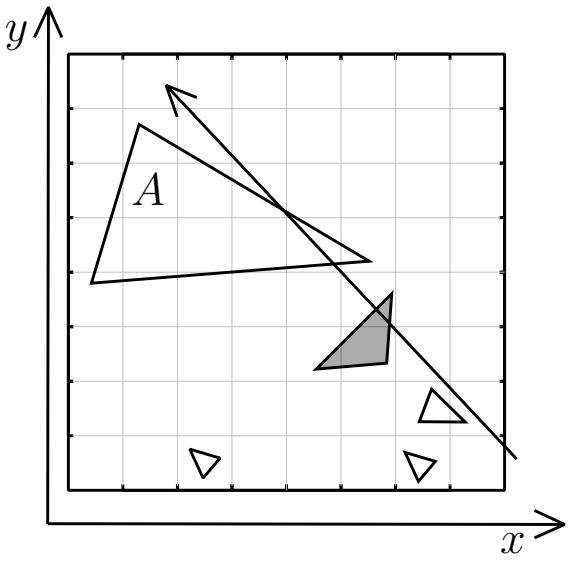
Bedingung:

Schnittpunkt:

b) Gegeben sei die abgebildete 2D-Szene. Sie sollen nun zwei räumliche Datenstrukturen konstruieren und in die Skizze einzeichnen. Falls Sie sich verzeichnen, dürfen Sie die Skizzen auf der folgenden Seite nutzen. Streichen Sie nicht zu bewertende Skizzen durch!



Quadtree



BVH

i) Konstruieren Sie für die linke Szene einen Quadtree! Unterteilen Sie dafür die Szene so lange, bis jeder Knoten höchstens ein Kind hat! **(3 Punkte)**

ii) Konstruieren Sie für die rechte Szene eine BVH über AABBs! Unterteilen Sie dafür die Szene, bis jeder Knoten höchstens ein Kind hat! Führen Sie Object Median Splits beginnend mit der y-Achse abwechselnd entlang der beiden Achsen durch! Falls eine ungerade Anzahl Dreiecke unterteilt wird, erhält die Menge näher am Ursprung ein Dreieck mehr. **(4 Punkte)**

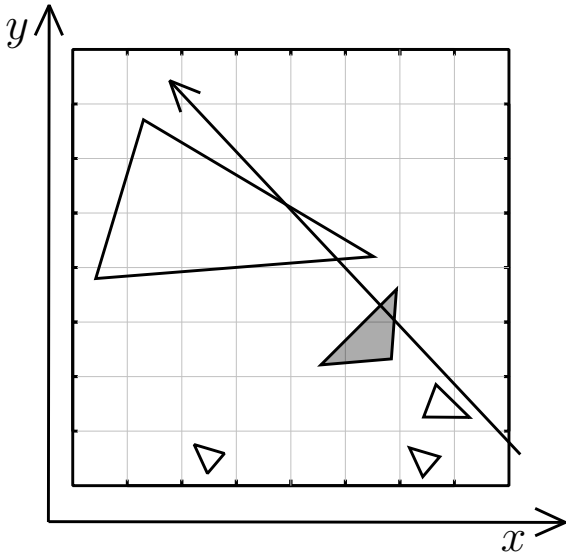
iii) Nun soll Ihre BVH für den eingezeichneten Strahl traversiert werden. Mit wie vielen Dreiecken muss der Strahl geschnitten werden, bevor der Schnittpunkt mit dem grauen Dreieck als nächster Schnittpunkt feststeht? Warum muss dafür kein Schnitttest mit Dreieck A durchgeführt werden? **(5 Punkte)**

Anzahl Schnitttests mit Dreiecken:

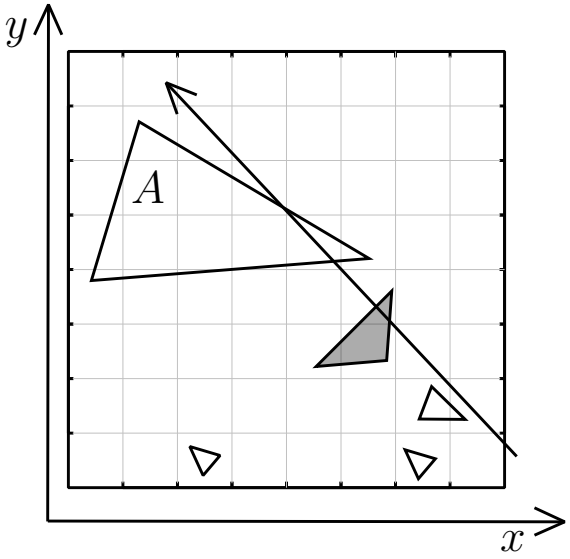
Kein Schnittest mit Dreieck A, weil

Matrikelnummer: _____

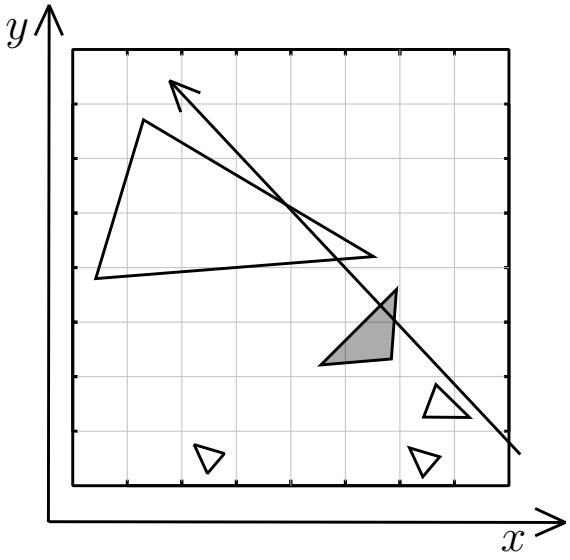
Ersatzskizzen für b):



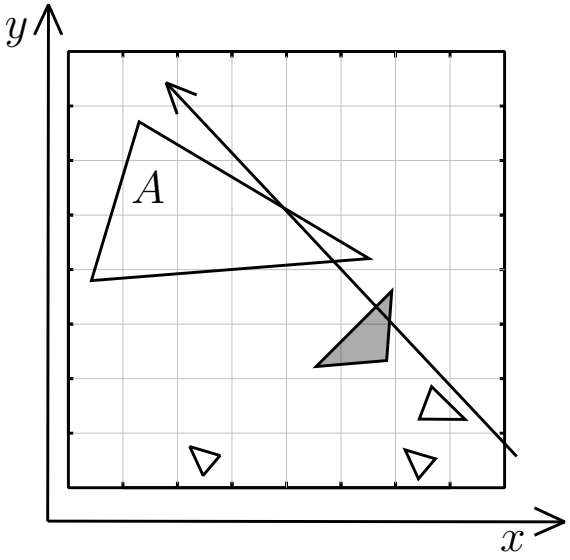
Quadtree



BVH



Quadtree



BVH

Aufgabe 5: Transformationen (12 Punkte)

a) Wie verändert das Transponieren einer nicht-uniformen Skalierungsmatrix die beschriebene Transformation? **(2 Punkte)**

b) Wie verändert das Transponieren einer Rotationsmatrix die beschriebene Transformation? **(2 Punkte)**

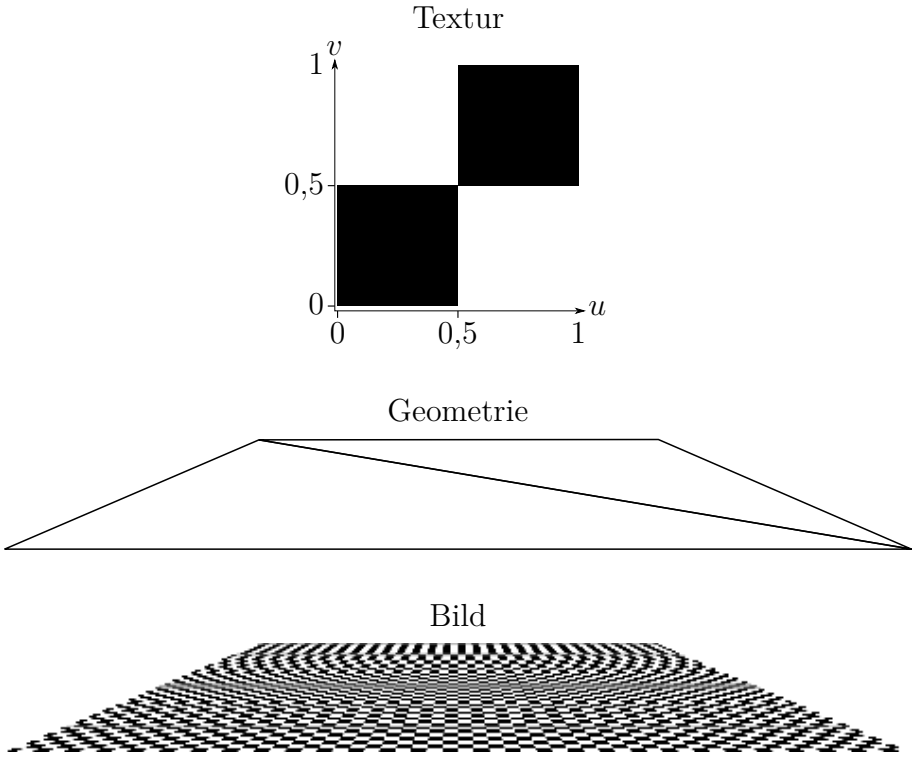
c) Nennen Sie eine affine Transformation im dreidimensionalen Raum, die durch eine 3×3-Matrix beschrieben werden kann! Nennen Sie außerdem eine affine Transformation im dreidimensionalen Raum, die *nicht* durch eine 3×3-Matrix beschrieben werden kann! **(2 Punkte)**

d) Eine Kamera befindet sich im Ursprung und blickt in Richtung der x-Achse mit Up-Vektor $(0, 0, 1)^T$. Nun wird die Kamera an Position $(4, 5, 6)^T$ verschoben und so rotiert, dass sie in Richtung der y-Achse blickt, aber ihren Up-Vektor beibehält. Gesucht ist die View-Transformation V, die einen Punkt in homogenen Koordinaten von Weltkoordinaten in Kamerakoordinaten transformiert. Wie und aus welchen grundlegenden Transformationen können Sie V berechnen? Geben Sie dafür auch die konkreten Matrizen dieser grundlegenden Transformationen an! Sie müssen die Matrix V selbst nicht ausrechnen. **(6 Punkte)**

V =

Matrikelnummer: _____

Aufgabe 6: Texturen & Baryzentrische Koordinaten (15 Punkte)



Die oben gezeigte Schachbrett-Textur soll auf der dargestellten Fläche (bestehend aus zwei Dreiecken) angezeigt werden. Darunter ist das Resultat abgebildet.

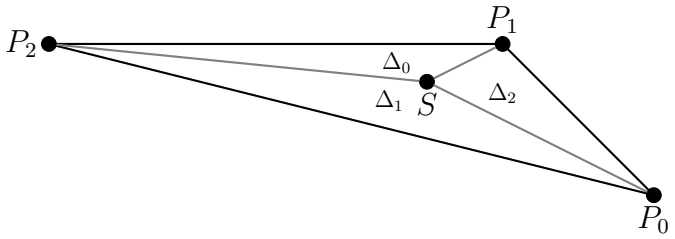
a) Wodurch entstehen die Artefakte, die insbesondere in der hinteren Hälfte der Oberfläche zu sehen sind? **(2 Punkte)**

b) Zur verbesserten Darstellung soll Mip-Mapping eingesetzt werden. Berechnen Sie die Mipmap-Stufen der Textur in der folgenden Abbildung: **(2 Punkte)**

1	1	0	0
1	1	0	0
0	0	1	1
0	0	1	1

c) Beschreiben Sie kurz, auf welche Mipmap-Stufen bei der trilinearen Interpolation zugegriffen wird! Was geschieht mit den Farbwerten daraus? **(4 Punkte)**

d) Für einen Punkt S im hinteren Dreieck seien die Flächeninhalte der Teildreiecke Δ_0, Δ_1 und Δ_2 gegeben (siehe Abbildung). Geben Sie eine Formel an, um die $i$te Komponente der baryzentrischen Koordinaten λ_i von S zu berechnen! **(2 Punkte)**



$\lambda_i =$

e) Die Texturkoordinaten $T(P)$ an den Eckpunkten des hinteren Dreiecks seien:

$$T(P_0) = \begin{pmatrix} 28 \\ 0 \end{pmatrix}, T(P_1) = \begin{pmatrix} 28 \\ 28 \end{pmatrix}, T(P_2) = \begin{pmatrix} 0 \\ 28 \end{pmatrix}$$

Berechnen Sie die interpolierten Texturkoordinaten im Punkt S mit den folgenden baryzentrischen Koordinaten! Geben Sie sowohl das vollständig ausgewertete Endergebnis sowie den Rechenweg an! **(3 Punkte)**

$$\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \frac{1}{14} \begin{pmatrix} 4 \\ 5 \\ 5 \end{pmatrix}$$

$T(S) =$

f) Nennen Sie zwei Möglichkeiten, um Texturkoordinaten außerhalb des Intervalls $[0, 1]^2$ zu behandeln! **(2 Punkte)**

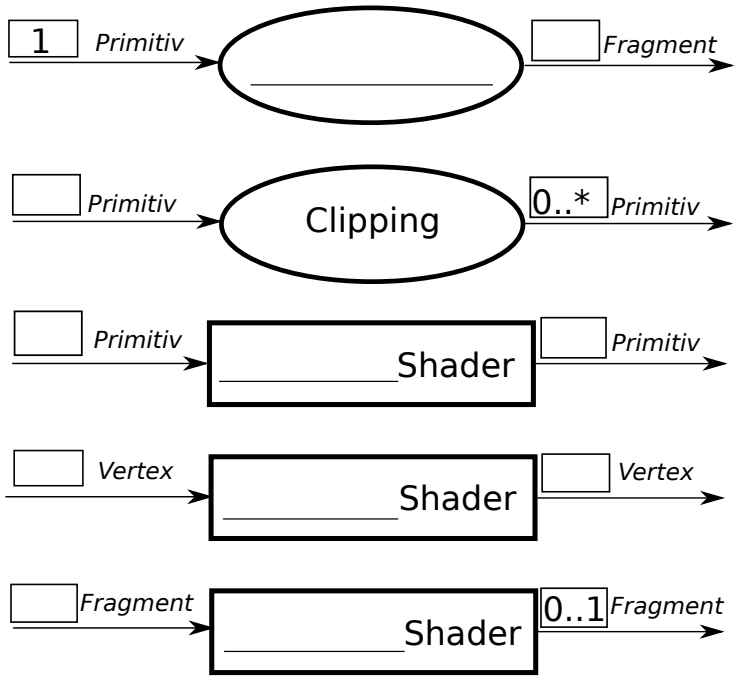
Matrikelnummer: _____

Aufgabe 7: OpenGL Pipeline (16 Punkte)

a) In der Abbildung sind in ungeordneter Reihenfolge einzelne Stufen der OpenGL 3 Grafik-Pipeline zur Rasterisierung von Dreiecken dargestellt.

- Tragen Sie die fehlenden drei Shader und Pipeline-Stufen ein!
- Vervollständigen Sie die Anzahl der Ein- und Ausgaben!
- Bringen Sie die Pipeline-Stufen in die korrekte Reihenfolge und geben Sie jeweils an, ob eine Stufe optional ist oder nicht!

(10 Punkte)



Reihenfolge *(1, 2, 3, 4, 5)*	Optional? *(Ja / Nein)*
	Ja

b) Durch welche Operation in der Pipeline wird die korrekte Verdeckung von unsortierten Dreiecken gewährleistet? An welcher Stelle der Pipeline findet diese Operation statt? **(3 Punkte)**

c) An welcher Stelle in der Pipeline findet die perspektivische Projektion statt? Begründen Sie, warum die Projektion nicht früher oder später durchgeführt werden kann! **(3 Punkte)**

Aufgabe 8: Blending (16 Punkte)

Sie haben in der Vorlesung Alpha-Blending zur Darstellung semi-transparenter Objekte kennengelernt.

a) Verwenden Sie folgende OpenGL-Befehle, um opake und semi-transparente Geometrie mit Alpha-Blending darzustellen! Konfigurieren Sie die korrekten Zustände für Tiefentest und Blending vollständig vor jedem Draw-Aufruf! Sie dürfen Befehle mehrmals verwenden. Nicht alle Befehle müssen verwendet werden. Verwenden Sie die Numerierung anstatt die Befehle auszuschreiben! **(8 Punkte)**

(0) `<draw_opaque>`	**(6)** `glEnable(GL_BLEND)`
(1) `<draw_transparent>`	**(7)** `glDisable(GL_BLEND)`
(2) `glEnable(GL_DEPTH_TEST)`	**(8)** `glBlendEquation(GL_FUNC_ADD)`
(3) `glDisable(GL_DEPTH_TEST)`	**(9)** `glBlendEquation(GL_FUNC_SUBTRACT)`
(4) `glDepthMask(GL_TRUE)`	**(10)** `glBlendEquation(GL_MIN)`
(5) `glDepthMask(GL_FALSE)`	**(11)** `glBlendEquation(GL_MAX)`
(12) `glBlendFunc(GL_ONE, GL_ZERO)`	
(13) `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`	
(14) `glBlendFunc(GL_ONE, GL_ONE)`	
(15) `glBlendFunc(GL_SRC_COLOR, GL_ONE_MINUS_SRC_COLOR)`	

b) Geben Sie die Berechnungsvorschrift für die resultierende Farbe im Framebuffer an, falls Alpha-Blending verwendet wird! Im Framebuffer steht bereits die Farbe $c_d = (R_d, G_d, B_d, A_d)$, der Fragmentshader gibt die Farbe $c_s = (R_s, G_s, B_s, A_s)$ aus. **(2 Punkte)**

Matrikelnummer: _____

c) Ist Alpha-Blending kommutativ? Begründen Sie oder geben Sie ein Gegenbeispiel für die Farben c_d und c_s an, sodass bei Vertauschen die resultierende Farbe im Framebuffer nicht die gleiche ist! Zeigen Sie die Korrektheit Ihres Gegenbeispiels durch Rechnung! **(6 Punkte)**

Aufgabe 9: GLSL Shader: Imperfekte Spiegelungen (30 Punkte)

In dieser Aufgabe sollen imperfekte Spiegelungen auf rauen Oberflächen approximiert werden. Zuerst werden Reflexionsstrahlen einer perfekten Spiegelung in einem Fragment Shader verfolgt, und anschließend das einfallende Licht weichgezeichnet.

a) Vervollständigen Sie den folgenden Fragment Shader, der für jeden Pixel des Bildes ausgeführt wird und das einfallende Licht einer perfekten Spiegelung berechnet! Verwenden Sie dazu die vorgegebene Funktion `raytrace_incident_light`! Die Koordinaten `P` und Normalen `N` der Schnittpunkte der Primärstrahlen sind in Weltkoordinaten in Texturen gespeichert, die im Fragment Shader bereits ausgelesen werden. **(10 Punkte)**

```
// Berechnung des einfallenden Lichts (RGB) für den Strahl O + t*D, t>=0
vec3 raytrace_incident_light(vec3 O, vec3 D);

uniform sampler2D positions, normals;
in vec2 texcoord;

in  vec3 V_interpolated;    // Vektor zur Kamera (Weltkoordinaten)
out vec4 r_in;              // Rückgabe für einfallendes Licht

const float ray_eps = 1e-7; // Epsilon zur Vermeidung von Selbstverschattung

void main() {
    vec3 P = texture(positions, texcoord).xyz;
    vec3 N = normalize(texture(normals, texcoord).xyz);









}
```

b) Das einfallende Licht sei nun für jeden Pixel in der Textur `tex_r_in` (in Bildschirmauflösung) gespeichert. Diese Farbwerte werden weichgezeichnet, um den Eindruck einer Reflexion an einer rauen Oberfläche zu simulieren. Dazu wird eine quadratische Nachbarschaft von $-\mathtt{S}$ bis $+\mathtt{S}$ Pixeln um jeden bearbeiteten Pixel betrachtet.

Die Filtergewichte w_i eines Nachbarpixels i hängen von dessen Normale $\mathtt{N}_i$ und der dazugehörigen Reflexionsrichtung $\mathtt{R}_i$, sowie `N` bzw. `R` des bearbeiteten Pixels ab: $w_i = (\langle \mathtt{N}, \mathtt{N}_i \rangle_+ \langle \mathtt{R}, \mathtt{R}_i \rangle_+)^{\mathtt{phong\_exp}}$. Dabei bezeichnet $\langle \cdot, \cdot \rangle_+$ das Skalarprodukt begrenzt auf den Wertebereich $[0, 1]$. Approximieren Sie V_i für die Nachbarpixel jeweils mit V!

Vervollständigen Sie den Fragment Shader so, dass er über alle Pixel der Nachbarschaft iteriert, deren Gewichte ausrechnet und das gewichtete Resultat ausgibt! **(20 Punkte)**

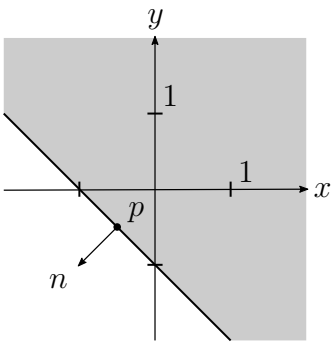
```
uniform sampler2D tex_r_in; // einfallendes Licht aus (a)
uniform sampler2D normals;  // Normale der Oberflächen
uniform vec2 pixelWidth;    // = 1 / Textur- bzw. Bildschirmauflösung

in  vec2 texcoord;          // Texturkoordinate des bearbeiteten Pixels
in  vec3 V_interpolated;    // Vektor zur Kamera (Weltkoordinaten)

out vec4 res;               // weichgezeichnetes Resultat

const int S = 3;            // Nachbarschaftsgroesse

void main() {
    vec3 V = ... // wie in Aufgabe a) berechnet
    vec3 N = normalize(texture(normals, texcoord).xyz);
    float phong_exp = texture(normals, texcoord).a;
    vec3 R = ... // wie in Aufgabe a) berechnet

    res = vec4(0.0);









}
```

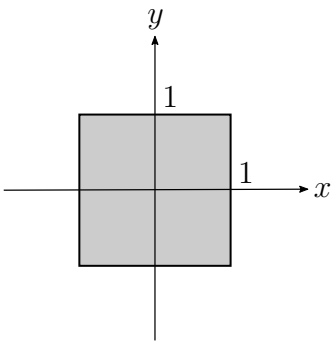
Aufgabe 10: Prozedurale Modellierung (9 Punkte)

a) Beim Zugriff auf prozedurale Texturen gibt es die Möglichkeit, diese entweder zur Laufzeit auszuwerten (implizit) oder durch Zugriff auf eine vorab generierte Textur (explizit). Wägen Sie diese beiden Methoden gegeneinander ab, indem Sie *jeweils* zwei *Nachteile* nennen! (Hinweis: Vorteile einer Methode dürfen als Nachteile der anderen verwendet werden.) **(4 Punkte)**

Nachteile Implizite Methode	Nachteile Explizite Methode

b) Gegeben sei die Distanzfunktion $\mathrm{D}_p(x)$ eines zweidimensionalen Halbraums, abgegrenzt durch eine Ebene mit Stützvektor p und Normale $n = p/\|p\|$. Die linke Abbildung zeigt als Beispiel die Halbräume der Instanz $\mathrm{D}_{(-0.5,-0.5)}(x)$. Nutzen Sie $\mathrm{D}_p(x)$, um die Distanzfunktion eines im Ursprung zentrierten Quadrates $\mathrm{D}_{\mathrm{quad}}(x)$ mit Seitenlänge 2 zu definieren! (Siehe rechte Abbildung) **(5 Punkte)**





$\mathrm{D}_{\mathrm{quad}}(x) =$

Matrikelnummer: _____

Aufgabe 11: Bézier-Kurven (10 Punkte)

a) Die beiden kubischen Bézier-Kurven $b(u) = \sum_{i=0}^{3} b_i B_i^3(u)$ und $c(u) = \sum_{i=0}^{3} c_i B_i^3(u)$ treffen sich am Punkt b_3 und c_0. Bestimmen Sie die Stetigkeit der daraus resultierenden Kurve und begründen Sie kurz Ihre Antwort! **(4 Punkte)**

$$b_0 = \begin{pmatrix} 14 \\ 2 \end{pmatrix},\ b_1 = \begin{pmatrix} 12 \\ 1 \end{pmatrix},\ b_2 = \begin{pmatrix} 10 \\ 1 \end{pmatrix},\ b_3 = \begin{pmatrix} 8 \\ 2 \end{pmatrix}.$$

$$c_0 = \begin{pmatrix} 8 \\ 2 \end{pmatrix},\ c_1 = \begin{pmatrix} 6 \\ 4 \end{pmatrix},\ c_2 = \begin{pmatrix} 4 \\ 4 \end{pmatrix},\ c_3 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

b) Die folgenden Kontrollpunkte definieren eine kubische Bézier-Kurve $b(u) = \sum_{i=0}^{3} b_i B_i^3(u)$:

$$b_0 = \begin{pmatrix} 1 \\ 3 \end{pmatrix},\ b_1 = \begin{pmatrix} 5 \\ 9 \end{pmatrix},\ b_2 = \begin{pmatrix} 9 \\ 3 \end{pmatrix},\ b_3 = \begin{pmatrix} 13 \\ 1 \end{pmatrix}.$$

Werten Sie die Kurve für $u = 0.5$ grafisch mit Hilfe des de Casteljau-Algorithmus aus! Skizzieren Sie die Schritte der Auswertung sowie die resultierende Bézier-Kurve! **(6 Punkte)**.