Karlsruher Institut für Technologie

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

Hauptklausur Computergrafik
WS 2020/2021

9. März 2021

Name	
Matrikelnummer	

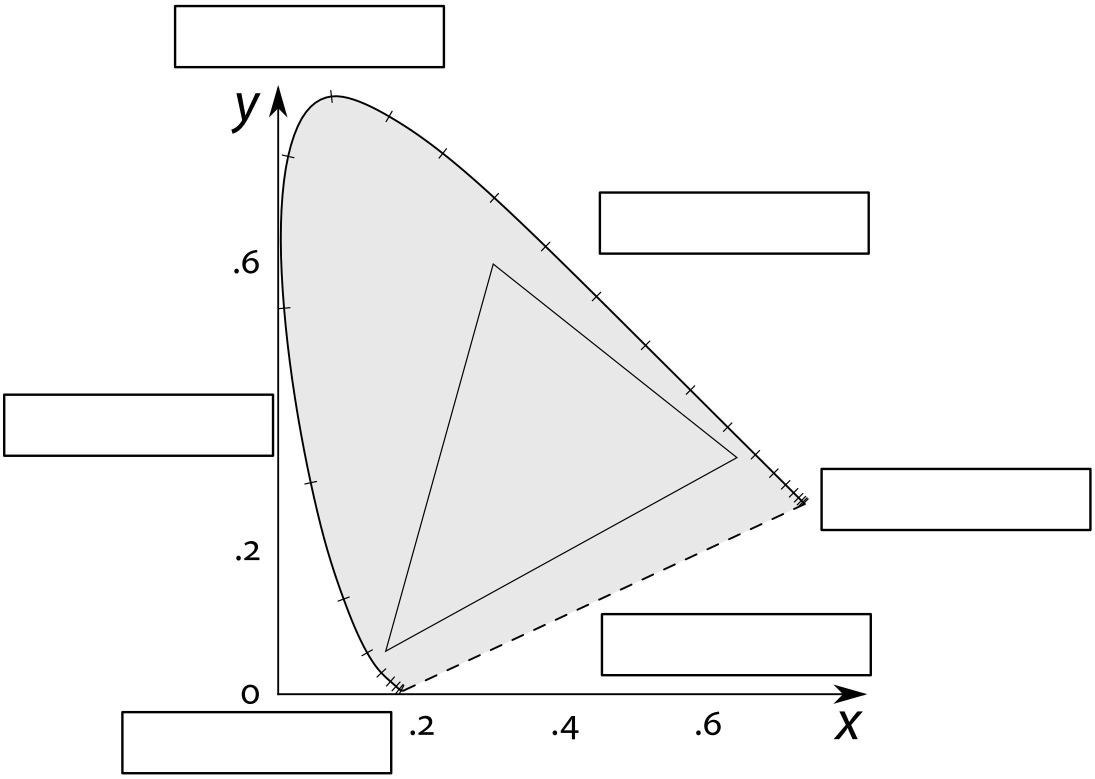
Beachten Sie:

- Die Klausur umfasst 24 Seiten (12 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.
- Streichen Sie nicht zu bewertende Lösungen durch.
- Wir akzeptieren auch englische Antworten.

Aufgabe	1	2	3	4	5	6	7	8	9	10	11	**Gesamt**
Erreichte Punkte												
Erreichbare Punkte	11	14	17	22	11	18	16	13	30	12	16	**180**

Note

Aufgabe 1: Farbe und Perzeption (11 Punkte)



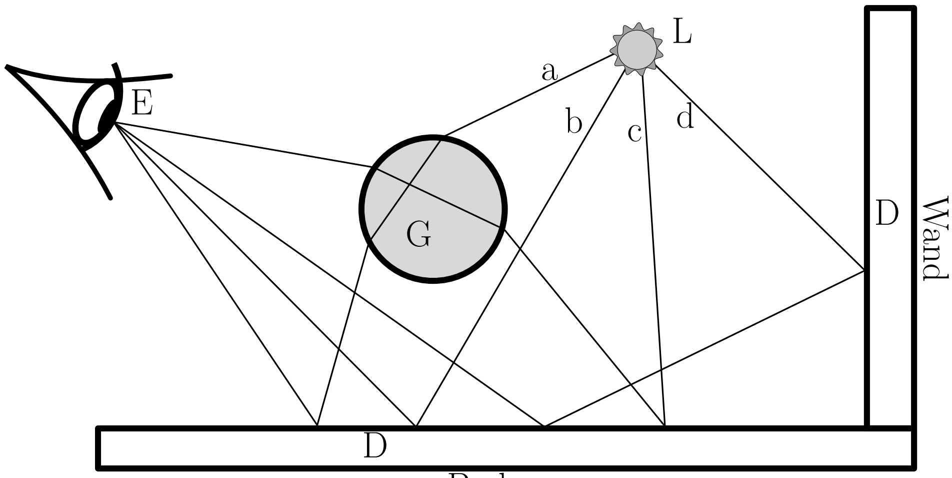
a) Das oben stehende CIE-Chromatizitätsdiagramm enthält sechs leere Kästen. Tragen Sie dort die entsprechenden Farben ein! **(3 Punkte)**

b) Die Achsen sind mit x und y beschriftet. Was bedeuten diese Symbole hier und wie berechnet man ihren Wert für einen gegebenen spektralen Stimulus $s(\lambda)$? **(6 Punkte)**

c) Die *Just Noticeable Difference* (JND) ist als gerade noch wahrnehmbarer Unterschied der Luminanz definiert. Wie kann sie mathematisch beschrieben werden und welche Eigenschaften hat sie? **(2 Punkte)**

Aufgabe 2: Whitted-style Raytracing (14 Punkte)

Im folgenden Diagramm sind vier Lichttransportpfade a, b, c und d eingezeichnet. Das Auge ist als E gekennzeichnet, rein-diffuse Objekte als D, die rein-spekulare Glaskugel (mit $k_a = k_d = k_s = 0, k_t > 0, k_r > 0$) als G und die Punktlichtquelle als L.



a) Geben Sie für jeden der Transportpfade an, ob dieser durch Whitted-style Raytracing erzeugt werden kann oder nicht, und begründen Sie Ihre Antworten! **(8 Punkte)**

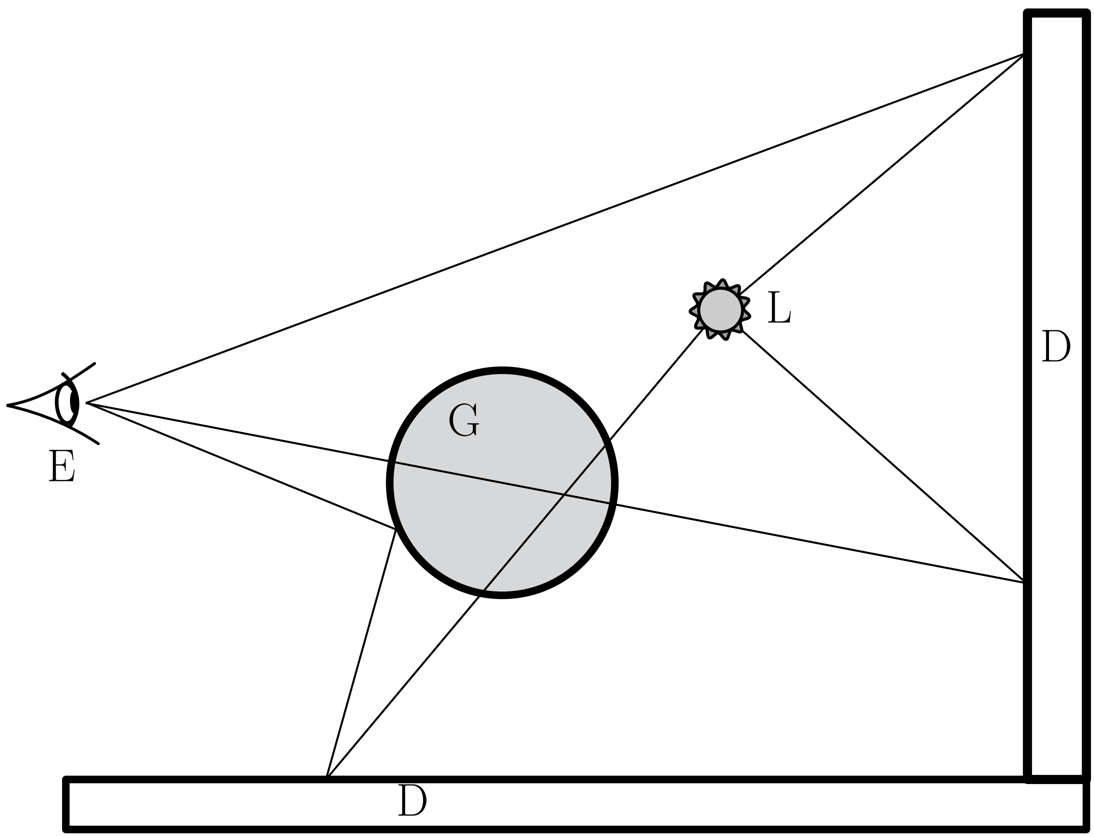
a:

b:

c:

d:

b) Zeichnen Sie im folgenden Bild für alle Strahlsegmente ein, ob es sich jeweils um einen Primär- (P), Transmissions- (T), Reflexions- (R) oder Schattenstrahl (S) handelt! Notieren Sie dazu die Buchstaben P, T, R oder S an dem jeweiligen Segment! **(6 Punkte)**



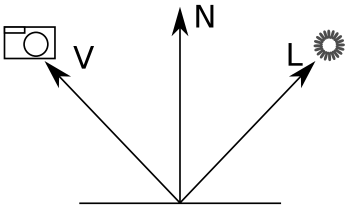
Aufgabe 3: Phong-Beleuchtungsmodell (17 Punkte)

a) In der Vorlesung haben Sie das Phong-Beleuchtungsmodell kennengelernt. Geben Sie den Term des Modells an, der die Glanzlichter kontrolliert, und benennen Sie dessen einzelne Komponenten kurz! **(7 Punkte)**

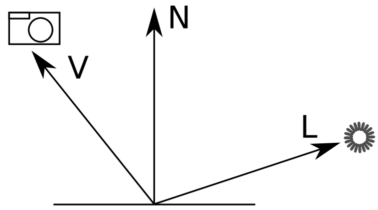
b) Beschreiben Sie kurz (ein Satz genügt) den Unterschied zwischen Gouraud- und Phong-Shading! Wann ist bei der Bildsynthese kein sichtbarer Unterschied zwischen diesen Verfahren wahrnehmbar? **(6 Punkte)**

c) Ordnen Sie die Konfigurationen (1), (2), (3) und (4) absteigend nach der Stärke der resultierenden Beleuchtung! Die Oberflächenparameter sind $k_a = 0, k_s = k_d = 1/2$. **(4 Punkte)**

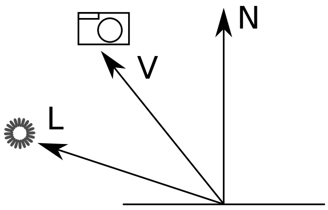
(1)



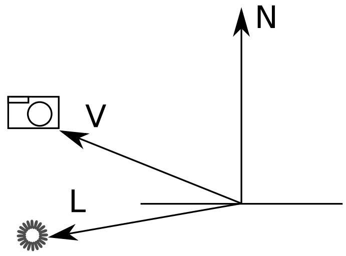
(2)



(3)

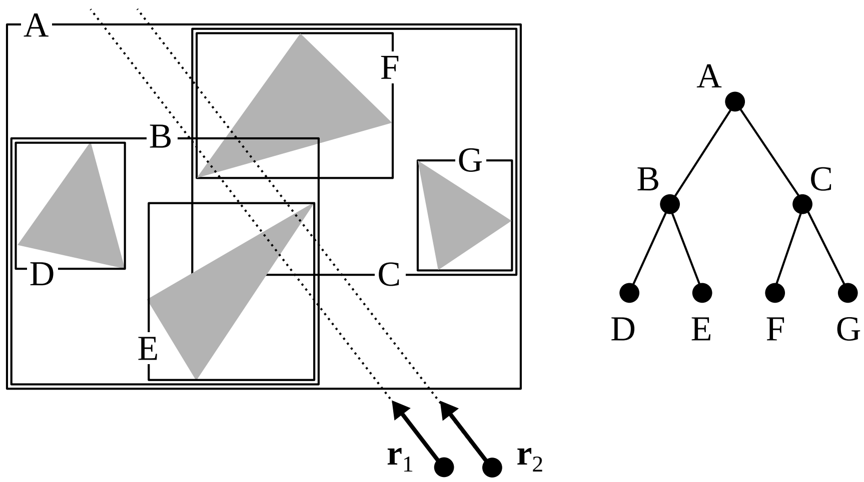


(4)



Aufgabe 4: Räumliche Datenstrukturen (22 Punkte)

In dieser Aufgabe werden Bounding Volume-Hierarchien (BVH) mit achsenausgerichteten Bounding Boxen (AABB) betrachtet. Eine 2D-Szene mit der zugehörigen BVH ist unten abgebildet. Links sind vier Dreiecke und die zugehörigen AABBs zu sehen, rechts die Baumstruktur.



a) Für die oben eingezeichneten Strahlen $\mathbf{r}_1$ und $\mathbf{r}_2$ soll jeweils der nächstgelegene Schnittpunkt bestimmt werden. Bei der Traversierung werden dabei näher gelegene Knoten zuerst verarbeitet. Geben Sie die Reihenfolge an, in der jeweils Knoten des BVH traversiert werden! Nutzen Sie dabei die Bezeichner A bis G! **(5 Punkte)**

$\mathbf{r}_1$:

$\mathbf{r}_2$:

b) Beantworten Sie die folgenden Fragen kurz! **(6 Punkte)**

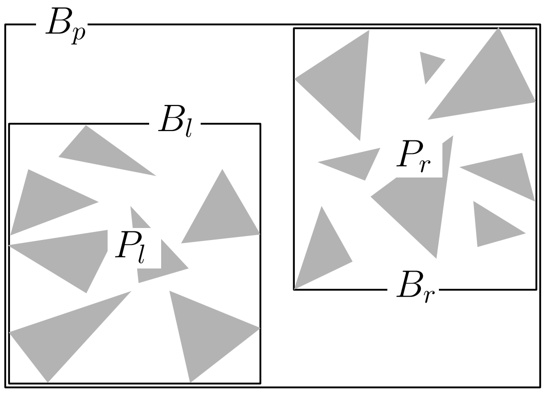
1) Welches Problem kann bei Verwendung von AABBs häufiger auftreten, wenn die Szene lang gestreckte Dreiecke enthält?

2) Welche Art von Hüllkörper ist in diesem Fall besser geeignet?

3) Nennen Sie einen Nachteil dieser Hüllkörper aus 2)!

Matrikelnummer: ____________________

Nun betrachten wir den top-down-Aufbau einer binären BVH. Wir nutzen dabei die *Surface Area Heuristic* (SAH), um zu entscheiden ob und wie ein Knoten in zwei Kindknoten unterteilt wird. Die folgende Abbildung zeigt eine mögliche Unterteilung einer 2D-Szene:



Wir nutzen die folgenden Bezeichnungen:

- Die AABB B_p umschließt die AABBs B_l und B_r der aufgeteilten Primitivgruppen P_l und P_r.
- B_l enthält die Primitive P_l und B_r enthält die Primitive P_r.
- Die Anzahlen der Primitive sind $|P_l|$ und $|P_r|$.
- Der Oberflächeninhalt der AABBs ist $SA(B_p)$, $SA(B_l)$, $SA(B_r)$.
- Der Oberflächeninhalt der Dreiecke ist $SA(P_b), SA(P_l), SA(P_r)$.
- Die Kosten zur Traversierung eines inneren Knotens betragen C_T.
- Die Kosten für einen Schnittest mit einem Primitiv betragen C_i.

c) Geben Sie die Formel zur Berechnung der Kosten für die oben gezeigte Unterteilung gemäß SAH an! Alternativ wird auch eine **präzise** Beschreibung der Heuristik in Worten als Antwort akzeptiert! **(7 Punkte)**

$C =$

e) Wir betrachten die vier unten gennanten Beschleunigungsstrukturen. Geben Sie jeweils ohne Begründung die Nummern der folgenden Eigenschaften an, die zutreffen!

1) Basiert auf Raumunterteilung.
2) Bei der Traversierung kann ein Primitiv mehrfach betrachtet werden.
3) Mailboxing kann genutzt werden, um überflüssige Schnitttests zu vermeiden.
4) Ungleich verteilte Geometrie kann effizient gehandhabt werden. **(4 Punkte)**

BVH:

kD-Baum:

reguläres Gitter:

Octree:

Matrikelnummer: ____________________

Aufgabe 5: Transformationen (11 Punkte)

Wir betrachten den Punkt $\mathbf{p} := (x, y, z, w)^T \in \mathbb{R}^4$ im 3D-Raum, der durch homogene Koordinaten gegeben ist.

a) Geben Sie eine Matrix $\mathbf{R_z}(\phi) \in \mathbb{R}^{4\times4}$ an, sodass $\mathbf{R_z}(\phi)\mathbf{p}$ im Winkel $\phi \in \mathbb{R}$ um die z-Achse rotiert ist! **(2 Punkte)**

$\mathbf{R_z}(\phi) =$

b) Geben Sie eine Matrix $\mathbf{T} \in \mathbb{R}^{4\times4}$ an, sodass $\mathbf{Tp}$ um 4 Längeneinheiten entlang der x-Achse und um -3 Längeneinheiten entlang der z-Achse verschoben ist! **(2 Punkte)**

$\mathbf{T} =$

c) Gegeben sind die Punkte mit den folgenden homogenen Koordinaten:

$$\mathbf{q_0} := \begin{pmatrix} 3 \\ 5 \\ 1 \\ 1 \end{pmatrix} \in \mathbb{R}^4, \qquad \mathbf{q_1} := \begin{pmatrix} 6 \\ 10 \\ 2 \\ 1 \end{pmatrix} \in \mathbb{R}^4, \qquad \mathbf{q_2} := \begin{pmatrix} 6 \\ 10 \\ 2 \\ 2 \end{pmatrix} \in \mathbb{R}^4.$$

Wie viele unterschiedliche Punkte im projektiven Raum enthält die Menge $\{\mathbf{q_0}, \mathbf{q_1}, \mathbf{q_2}\}$? Begründen Sie Ihre Antwort! **(3 Punkte)**

d) Gegeben ist die Matrix

$$\mathbf{M} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \in \mathbb{R}^4.$$

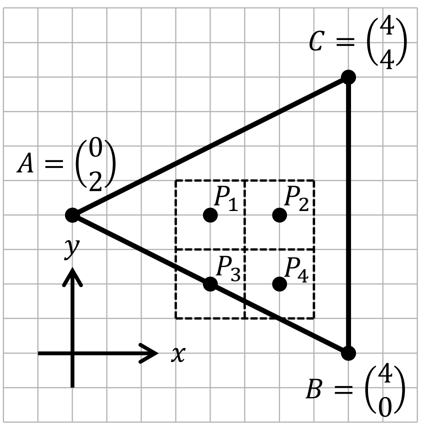
Berechnen Sie die Koordinaten des Punktes $(x', y', z', w')^T := \mathbf{Mx}$! Beschreiben Sie in Worten, welche Transformation **M** darstellt! Welche Eigenschaft haben alle Punkte nach dieser Transformation gemein? **(4 Punkte)**

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} =$$

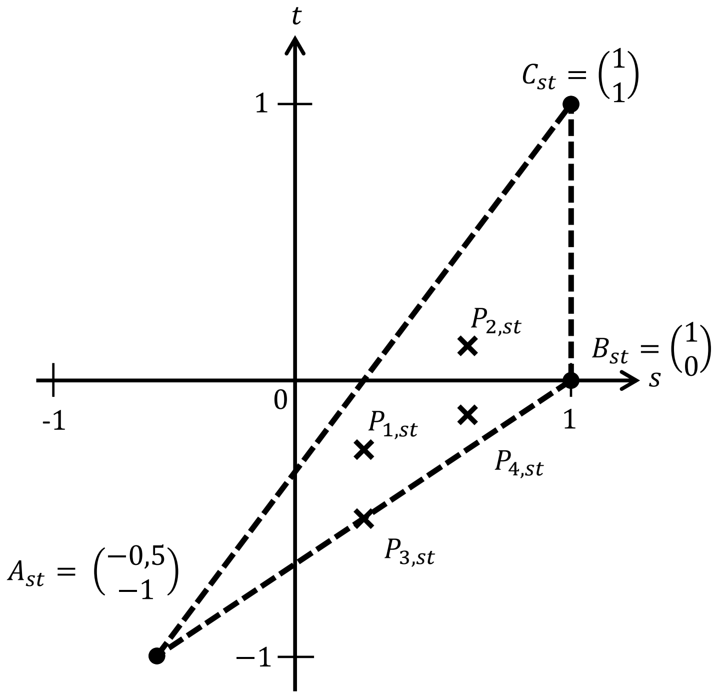
Matrikelnummer: ______________________

Aufgabe 6: Texturen und Baryzentrische Koordinaten (18 Punkte)

Betrachten Sie in der linken Abbildung das Dreieck mit den Eckpunkten $A = (0,2), B = (4,0), C = (4,4)$, sowie die Koordinate des Pixelmittelpunkts $P_1 = (2,2)$. Zusätzlich sind drei benachbarte Pixelmittelpunkte eingezeichnet. Die rechte Abbildung zeigt die zugehörigen Texturkoordinaten A_{st}, B_{st}, C_{st} und $P_{1,st}, P_{2,st}, P_{3,st}, P_{4,st}$ im Texturraum.



Objektraum (x, y)

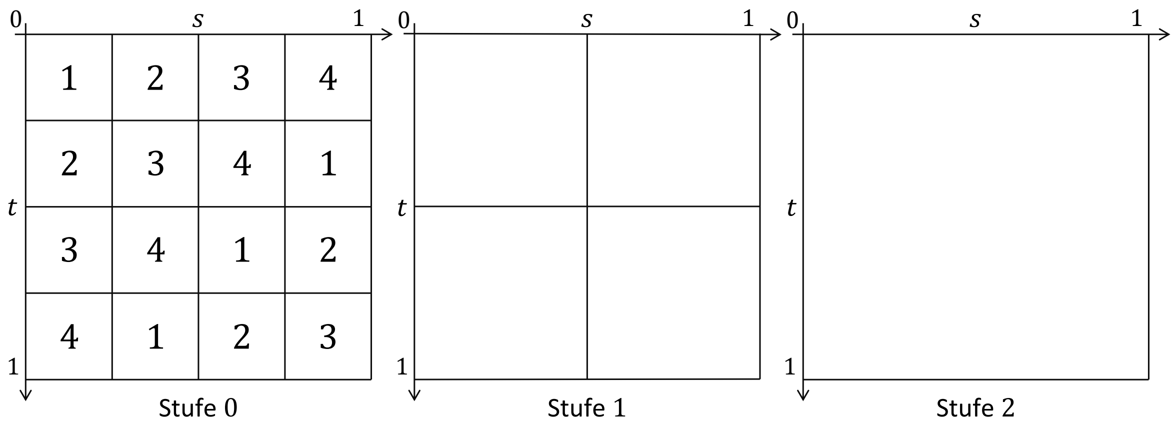


Texturraum (s, t)

a) Was ist die interpolierte Texturkoordinate $P_{1,st}$ für den Punkt $P_1 = (2,2)$? Geben Sie auch den Rechenweg an! **(8 Punkte)**

b) Die weiteren Texturkoordinaten seien: $P_{2,st} = (\frac{5}{8}, \frac{1}{8})$, $P_{3,st} = (\frac{2}{8}, -\frac{4}{8})$ und $P_{4,st} = (\frac{5}{8}, -\frac{1}{8})$. Bestimmen Sie näherungsweise über die Texturkoordinaten der Nachbarpixel die Maße des isotropen Footprints für Mip-Mapping für den Pixel um P_1! **(4 Punkte)**

c) Vervollständigen Sie die drei Mip-Map-Stufen der Textur in der unteren Abbildung! Geben Sie an, welche der Mip-Map-Stufen für den Footprint aus b) ausgewählt wird! Geben Sie außerdem an, welcher Wert bei „Nearest-Neighbor-Filtering“ mit dem Wrapping-Mode `GL_REPEAT` für den Pixel an p_1 ausgelesen wird und begründen Sie dies! **(6 Punkte)**



Stufe:

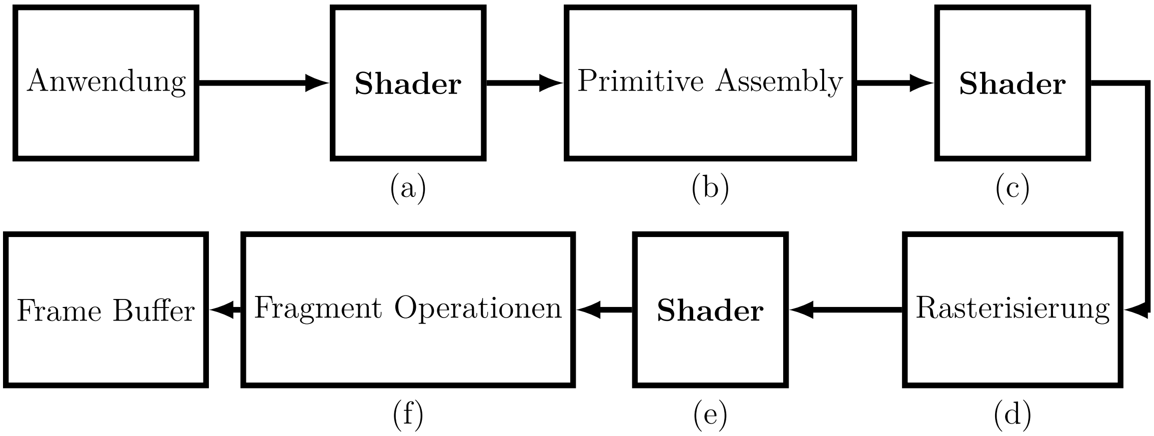
Ausgelesener Wert:

Hinweis: GL_REPEAT = Wiederholung der Textur ohne Spiegelung

Matrikelnummer: ______________________

Aufgabe 7: OpenGL-Pipeline (16 Punkte)

Gegeben sei der folgende Ausschnitt der OpenGL-Pipeline. Die Tessellierungsfunktionalität wird in dieser Aufgabe nicht betrachtet.



a) Nennen Sie den jeweiligen Namen der Shader-Stufen und geben Sie **eine** typische Aufgabe an, die dort ausgeführt wird! **(6 Punkte)**

(a)

(c)

(e)

b) Ein großes Dreiecksnetz soll kompakt in OpenGL-Puffern gespeichert und effizient mit einem einzigen OpenGL-Aufruf `glDrawElements` gezeichnet werden. Wie nennt sich diese Repräsentation von Dreiecksnetzen und welche Puffer verwenden Sie hierzu? **(3 Punkte)**

c) Warum muss bei Rasterisierung zwingend Clipping durchgeführt werden? In welchem Koordinatensystem findet Clipping am effizientesten statt? **(4 Punkte)**

d) Falls in den Shadern nur die Transformationmatrix von Modell- in Clip-Koordinaten (Model-View-Projection) zur Verfügung steht, kann eine Beleuchtungsberechnung nur in einem Koordinatensystem korrekt durchgeführt werden. Welches ist das und warum? **(3 Punkte)**

Matrikelnummer: ____________________

Aufgabe 8: Alpha-Blending (13 Punkte)

Eine Szene mit opaken und semitransparenten Dreiecken soll mittels Alpha-Blending gezeichnet werden.

a) Geben Sie an, wie die OpenGL-Pipeline konfiguriert werden muss, sodass Tiefenwerte korrekt geschrieben und benutzt werden, um opake und semitransparente Dreiecke zu zeichnen. *Nennen* oder *beschreiben* Sie die zwei OpenGL-Kommandos die vor jeder Zeichenoperation nötig sind! **(4 Punkte)**

1) Vor dem Zeichnen der opaken Dreiecke:

2) Vor dem Zeichnen der semitransparenten Dreiecke:

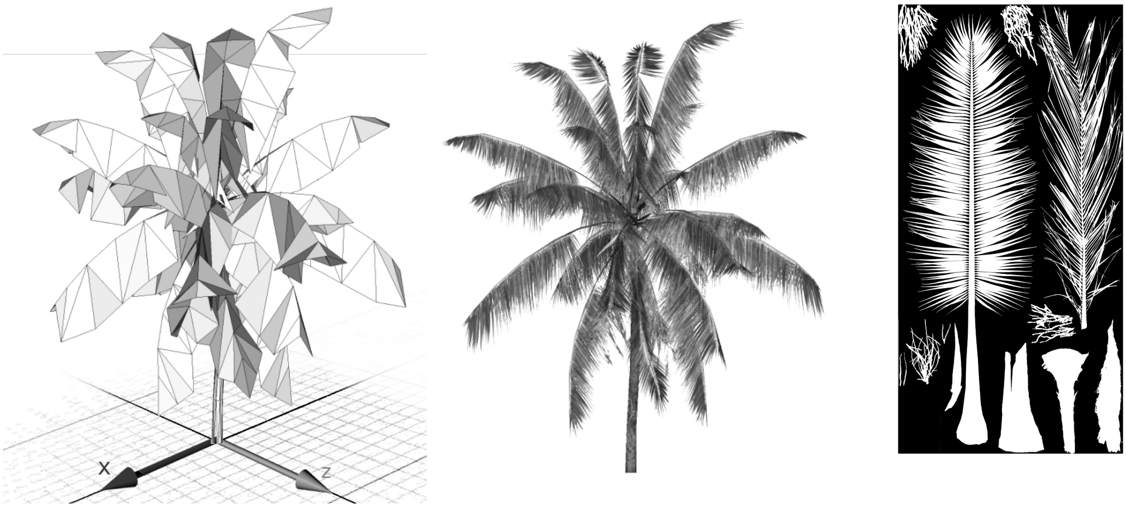
b) Für korrektes Alpha-Blending müssen die semitransparenten Dreiecke vor dem Zeichnen sortiert werden. Begründen Sie kurz, warum dies notwendig ist, und geben Sie das Sortierkriterium und die -Reihenfolge an! **(3 Punkte)**

c) Verwenden Sie folgende Befehle in der richtigen Reihenfolge, sodass die opaken und semitransparenten Dreiecke korrekt gezeichnet werden! Nicht alle Befehle müssen dabei verwendet werden! Verwenden Sie die Kürzel (O, T, 1, ...) anstatt die Befehle auszuschreiben! **(6 Punkte)**

(O) `<draw_opaque>`	**(3)** `glEnable(GL_BLEND)`
(T) `<draw_sorted_transparent>`	**(4)** `glDisable(GL_BLEND)`
(1) `glBlending(GL_TRUE)`	**(5)** `glBlendEquation(GL_FUNC_ADD)`
(2) `glBlending(GL_FALSE)`	**(6)** `glBlendEquation(GL_FUNC_SUB)`
(7) `glBlendFunc(GL_SRC_ALPHA, GL_DST_ALPHA)`	
(8) `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`	
(9) `glBlendFunc(GL_ONE, GL_SRC_ALPHA)`	
(10) `glBlendFunc(GL_SRC_COLOR, GL_ONE_MINUS_SRC_COLOR)`	

Aufgabe 9: GLSL: Instanziierte Vegetation (30 Punkte)

In dieser Aufgabe sollen Sie die OpenGL-Shader für ein instanziiertes Rendering von Palmen schreiben. Das untere Ende des Palmenstammes liegt im Ursprung des Modellkoordinatensystems. Die Palmentextur enthält einen Alpha-Kanal zur Festlegung transparenter Teile der Oberfläche:



Geometrie der Palme (links), texturierte Palme (mitte) und Alpha-Kanal der Textur (rechts). Die Alphawerte sind entweder völlig transparent oder völlig opak.

Die Applikation zeichnet die Geometrie mittels Instanziierung mit folgendem Aufruf:

```
glDrawArraysInstanced(Glenum mode, Glint first,
                      Glsizei count, GLsizei primcount)
```

Ergänzen Sie die folgenden Shader (Vertex, Geometry, Fragment) so, dass sie folgende Aufgaben umsetzen: **(30 Punkte)**

- Alle Dreiecke von Modell-Instanzen, deren Koordinatenursprung (Anfang des Stammes) hinter der Kamera liegt oder einen Abstand zur Kamera von mehr als 1000 Längeneinheiten hat, sollen komplett verworfen werden.
- Nutzen Sie den Alpha-Wert der Textur, damit transparente Teile der Palme den Wert des Tiefenpuffers nicht verändern.
- Diffuse Schattierung wird anhand der Normalen in Weltkoordinaten mit Phong-Shading mit der vorgegebenen Funktion `shade(vec3 reflectance, vec3 normal)` berechnet. Sie müssen `shade(...)` nicht selbst implementieren.

Hinweis: Unterschiedliche Lösungen sind möglich!

a) Vervollständigen Sie den Vertex-Shader so, dass `gl_Position` berechnet wird! Definieren und implementieren Sie die weiteren Ausgaben des Shaders! Die Modell-Transformation `M` für die aktuelle Instanz wird von der Funktion `readTransformation(...)`, die Sie nicht selbst implementieren müssen, ausgelesen und als Matrix übergeben.

```
// shared uniforms
uniform mat4 V, P;          // View- und Projection-Matrix
uniform sampler2D M_tex;    // Textur mit Modell-Transformationen der Instanzen

in vec3 position;    // Vertexkoordinate in Modellkoordinate
in vec3 normal;      // Vertexnormale in Modellkoordinate
in vec2 uv;          // Texturkoordinate

out vec2 tex_uv_g;
// Definieren Sie hier Ihre Shader-Ausgaben
out







mat4 readTransformation(int instance_id, sampler2D texture) { ... }
vec3 shade(vec3 reflectance, vec3 normal) {...}

void main()
{
    // Die Modell-Transformationsmatrix der aktuellen Instanz
    mat4 M = readTransformation(gl_InstanceID, M_tex);

    // Setze Sie Ihre Shaderausgaben und gl_Position
    tex_uv_g = uv;

    gl_Position =

}
```

b) Vervollständigen Sie den Geometry-Shader! Denken Sie daran, Ihre Ausgaben und `gl_Position` pro Vertex zu setzen!

```
layout (triangles) in;
layout (triangle_strip, max_vertices = 3)

// shared uniforms
uniform mat4 V, P;          // View- und Projection-Matrix
uniform sampler2D M_tex;    // Textur mit Modell-Transformationen der Instanzen

in vec2 tex_uv_g[];
// Definieren Sie hier Ihre Eingaben (in ...) und Ausgaben (out ...)









out vec2 tex_uv;

mat4 readTransformation(int instance_id, sampler2D texture) { ... }
vec3 shade(vec3 reflectance, vec3 normal) {...}

void main()
{










    // Setzen Sie gl_Position und die Shaderausgaben pro Vertex
    for(int i = 0; i < 3; i++)
    {
        gl_Position = //...










        tex_uv = tex_uv_g[i];
        EmitVertex();
    }
    EndPrimitive();
}
```

c) Vervollständigen Sie den Fragment-Shader! Die Textur mit RGB-Farbe und Alpha-Wert wird bereits ausgelesen.

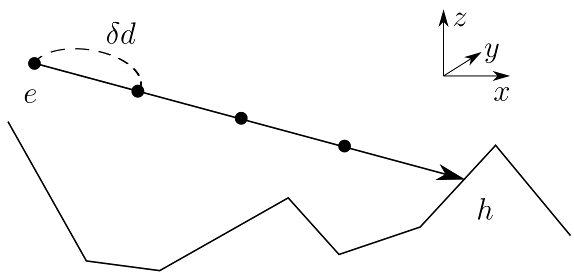
```
// shared uniforms
uniform mat4 V, P;          // View- und Projection-Matrix
uniform sampler2D M_tex;   // Textur mit Modell-Transformationen der Instanzen
uniform sampler2D palme;   // RGBA Palmen-Textur

in vec2 tex_uv;
// Definiere Eingaben (in ...)







out vec4 out_color;

mat4 readTransformation(int instance_id, sampler2D texture) { ... }
vec3 shade(vec3 reflectance, vec3 normal) {...}

void main()
{
    // Reflektanz (Farbe) und Alpha
    vec4 R = texture(palme, tex_uv);















    out_color =









}
```

Aufgabe 10: Prozedurale Modellierung und Ray Marching (12 Punkte)

a) Wie ist eine Turbulenzfunktion basierend auf einer Rauschfunktion $n(\boldsymbol{x})$ definiert? Beschreiben Sie stichpunktartig die vorkommenden Terme! **(4 Punkte)**

Wir betrachten nun eine spezielle Turbulenzfunktion $t(\boldsymbol{x})$, mit $\boldsymbol{x} = (x, y)$ in $\mathbb{R}^2$, mit der ein Höhenfeld $h(\boldsymbol{x}) = (x, y, t(\boldsymbol{x}))$ definiert ist: $t(\boldsymbol{x}) = n(\boldsymbol{x}) - |n(2\boldsymbol{x})| - |n(3\boldsymbol{x})|$.

Mittels Raymarching wird einem Strahl von $\boldsymbol{e}$ in Schritten $\delta\boldsymbol{d}$ gefolgt. Gesucht ist nun der erste Schnittpunkt mit dem Höhenfeld, für den gilt: Die z-Koordinate von $e + t\delta\boldsymbol{d}$ ist kleiner als die von h an der entsprechenden Stelle.



b) Die Auswertung von $h(\boldsymbol{x})$ ist teuer. Wie kann der Schnitttest möglichst effizient erfolgen, ohne dass die Auswertung der Rauschfunktion $n(\boldsymbol{x})$ selbst optimiert wird? **(4 Punkte)**

c) Wie kann die Normale an einem Schnittpunkt $\boldsymbol{s}$ berechnet werden? Sie können die analytische oder eine näherungsweise Berechnung angeben oder diese in Worten beschreiben. **(4 Punkte)**

Matrikelnummer: ____________________

Aufgabe 11: Bézier-Kurven (16 Punkte)

a) Werten Sie die kubische Beziér-Kurve $\mathbf{F}_1(u)$ mit den Kontrollpunkten

$$\mathbf{b}_0^0 = (8, 9) \qquad \mathbf{b}_1^0 = (4, -7) \qquad \mathbf{b}_2^0 = (0, -7) \qquad \mathbf{b}_3^0 = (-4, 9)$$

rechnerisch an der Stelle $u = \frac{1}{4}$ mit dem de Casteljau-Algorithmus aus und tragen Sie die Zwischenergebnisse in die Felder der Pyramide ein! **(8 Punkte)**

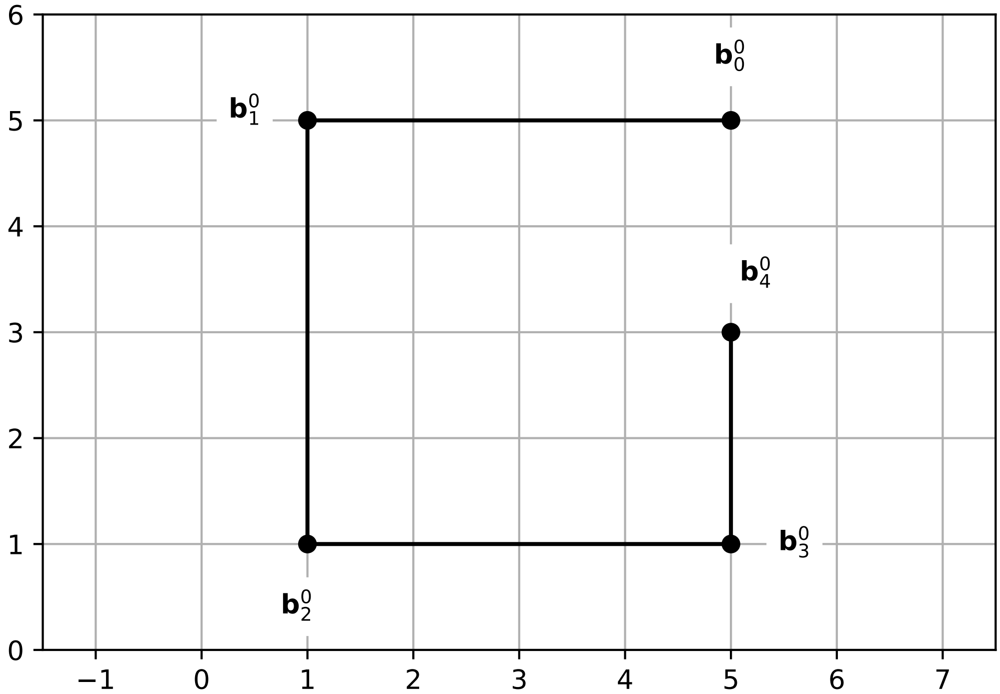
<table>
<tr><td colspan="3"></td><td colspan="2">$\mathbf{b}_0^3 =$</td><td colspan="3"></td></tr>
<tr><td colspan="2"></td><td colspan="2">$\mathbf{b}_0^2 =$</td><td colspan="2">$\mathbf{b}_1^2 =$</td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2">$\mathbf{b}_0^1 =$</td><td colspan="2">$\mathbf{b}_1^1 =$</td><td colspan="2">$\mathbf{b}_2^1 =$</td><td></td></tr>
<tr><td colspan="2">$\mathbf{b}_0^0 = (8, 9)$</td><td colspan="2">$\mathbf{b}_1^0 = (4, -7)$</td><td colspan="2">$\mathbf{b}_2^0 = (0, -7)$</td><td colspan="2">$\mathbf{b}_3^0 = (-4, 9)$</td></tr>
</table>

b) Geben Sie an, wie Sie aus den Zwischenergebnissen in Aufgabe a) die exakte Tangentenrichtung am Punkt $\mathbf{F}_1(\frac{1}{4})$ berechnen können! **(4 Punkte)**

c) Wir betrachten die quartische Beziér-Kurve $\mathbf{F}_2(u) = \sum_{i=0}^{4} \mathbf{b}_i^0 B_i^4(u)$ mit den folgenden Kontrollpunkten:

$$\mathbf{b}_0^0 = (5,5) \qquad \mathbf{b}_1^0 = (1,5) \qquad \mathbf{b}_2^0 = (1,1) \qquad \mathbf{b}_3^0 = (5,1) \qquad \mathbf{b}_4^0 = (5,3)$$

Werten Sie die Kurve $\mathbf{F}_2(u)$ für $u = \frac{1}{2}$ grafisch mit Hilfe des de Casteljau-Algorithmus aus! Skizzieren Sie die Schritte der Auswertung, zeichnen Sie den Punkt ein und nutzen Sie diese, um die komplette Bézier-Kurve zu skizzieren! **(4 Punkte)**



Alternativskizze. Streichen Sie nicht zu bewertende Skizzen durch!