Karlsruher Institut für Technologie

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

Nachklausur Computergrafik
SS 2022

8.8.2022

Name	
Matrikelnummer	

Beachten Sie:

- Die Klausur umfasst 27 Seiten (14 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	13	9	14	28	11	17	17	12	33	9	17	**180**

Note

Aufgabe 1: Farbe und Perzeption (13 Punkte)

a) Wie nennt man den Effekt, dass ein Farbeindruck im Allgemeinen durch verschiedene Spektren hervorgerufen werden kann? **(1 Punkt)**

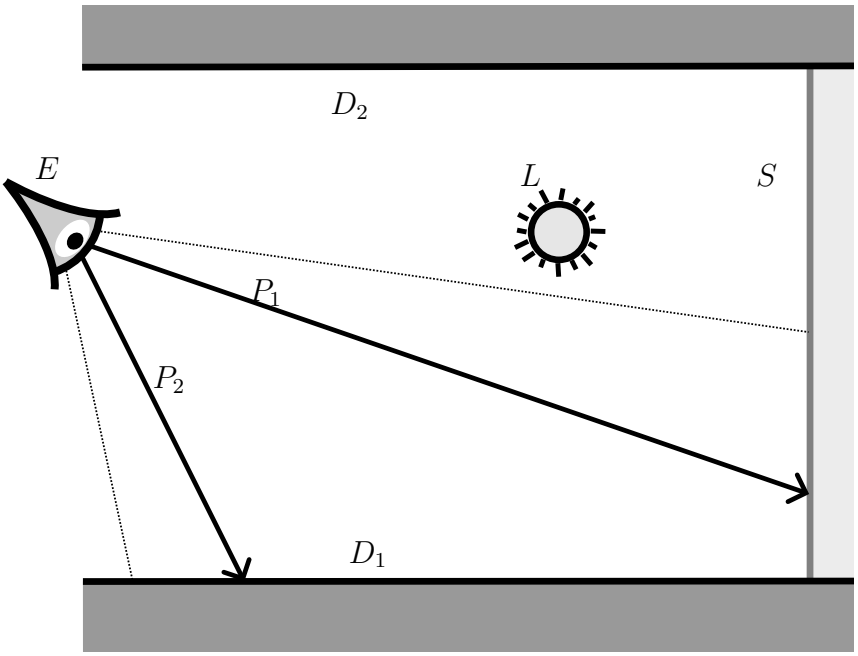
b) Sind Cyan, Magenta und Gelb übliche Primärfarben für additive oder für subtraktive Farbmischung? Warum wären sie als Primärfarben für die andere Farbmischung schlecht geeignet? **(4 Punkte)**

c) Geben Sie die Formel(n) zur Berechnung der Tristimulus-Werte aus einem Spektrum $A(\lambda)$ an und benennen Sie die Funktionen! **(4 Punkte)**

d) Berechnen Sie die Chromatizität einer Farbe, die als (X, Y, Z) im XYZ-Farbraum gegeben ist! Wie können sich Farben unterscheiden, die dieselbe Chromatizität haben? **(4 Punkte)**

Matrikelnummer: _____

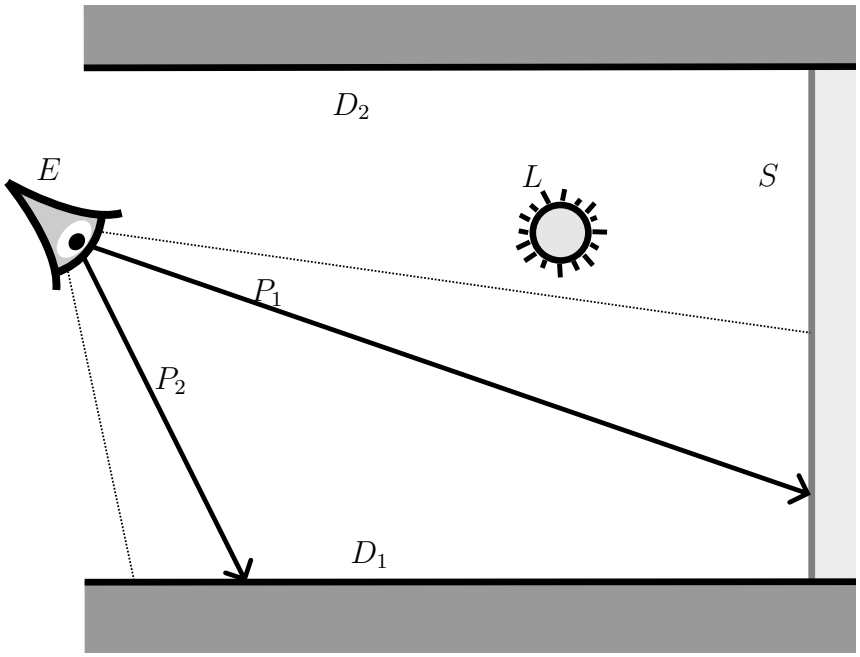
Aufgabe 2: Whitted-style Raytracing (9 Punkte)



In obiger Skizze ist die Kamera E und eine Punktlichtquelle L eingezeichnet. Weiterhin gibt es als Geometrie einen diffusen Boden D_1 und eine diffuse Decke D_2, sowie einen Spiegel S. Diese Szene soll nun mit Whitted-style Raytracing dargestellt werden.

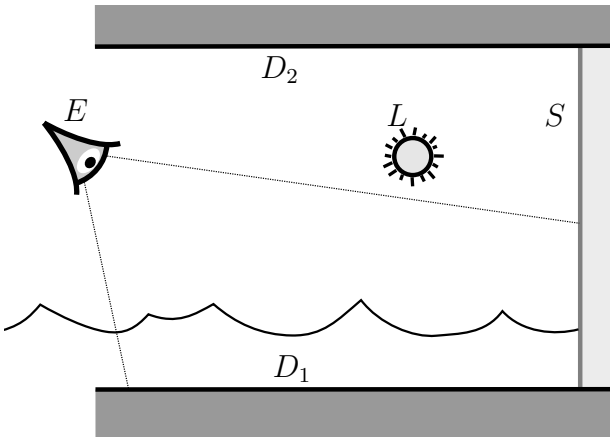
a) Auch in der Skizze eingezeichnet sind zwei Pfadsegmente P_1 und P_2. Vervollständigen Sie diese Pfade mittels Whitted-style Raytracing, indem Sie die weiteren erzeugten Strahlen einzeichnen!

Zur Korrektur können Sie die zweite Zeichnung verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnung bewertet werden soll!* **(3 Punkte)**



b) Die Kamera E sieht nur den Boden und den Spiegel im Bild, nicht die Decke. Die gestrichelten Linien deuten das Sichtfeld an. Die Szene wird einmal mit roter Decke und einmal mit blauer Decke gerendert. Wie unterscheiden sich die Bilder? Begründen Sie! **(3 Punkte)**

c) Der Boden wird unter Wasser gesetzt. Kamera und Licht bleiben über Wasser (siehe Skizze). Die Wasseroberfläche wird als semi-transparente Fläche modelliert. Wie muss man beim Schattenstrahl vorgehen, damit der Boden nicht schwarz dargestellt wird? Warum ist das Ergebnis immer noch anders als bei Distributed Raytracing oder Path Tracing? **(3 Punkte)**



Matrikelnummer: _____

Aufgabe 3: Phong-Beleuchtungsmodell (14 Punkte)

a) Geben Sie die Formel für den spekularen Term des Phong-Beleuchtungsmodells an! Beschreiben sie, wie ein Glanzlicht vom Phong-Exponenten abhängt und begründen Sie anhand der Formel! **(4 Punkte)**

b) Bei der Auswertung des Phong-Beleuchtungsmodells für glatte Schattierung wird eine Normale benötigt. Beschreiben Sie kurz, wie eine Vertex-Normale aus den angrenzenden Dreiecken in einem Dreiecksnetz berechnet werden kann! **(3 Punkte)**

c) Für ein Dreiecksnetz soll das Phong-Beleuchtungsmodell zusammen mit Gouraud Shading ausgewertet werden. Beschreiben Sie, wie die Schattierung der Dreiecksflächen berechnet wird! **(3 Punkte)**

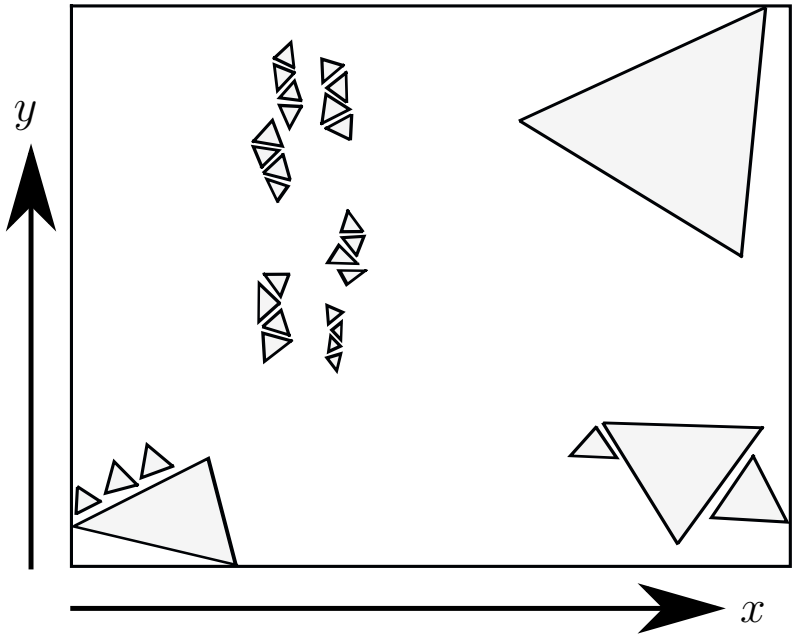
d) Durch eine Punktlichquelle in der Szene soll auf dem Dreiecksnetz aus Aufgabe 3c) ein Glanzlicht zu erkennen sein. Allerdings erscheint das Glanzlicht schwach und unnatürlich. Nennen sie einen möglichen Grund hierfür und erklären Sie zwei verschiedene Ansätze, um die Darstellung zu verbessern! Die Materialparameter sollen hierbei nicht verändert werden. **(4 Punkte)**

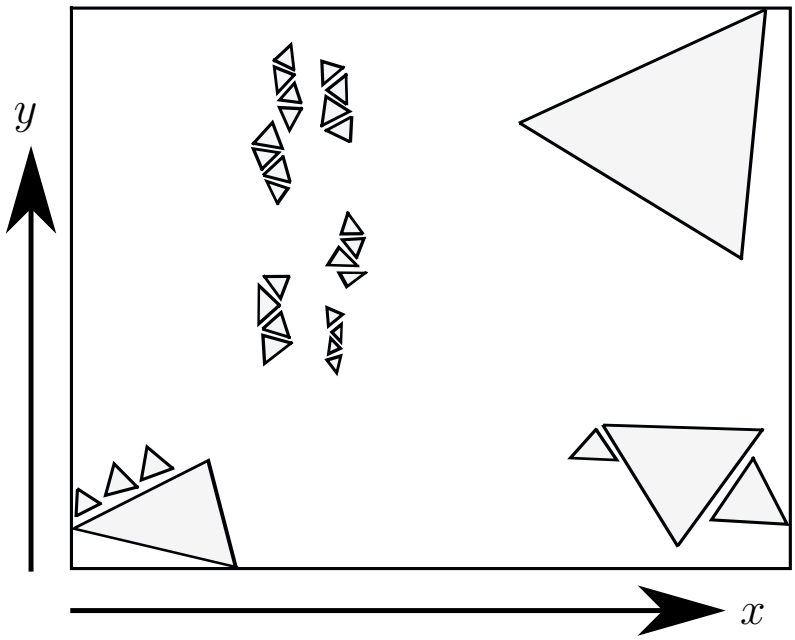
Matrikelnummer: _____

Aufgabe 4: Räumliche Datenstrukturen (28 Punkte)

a) Gegeben ist die folgende 2D-Szene mit 32 Primitiven und dem Hüllrechteck der Szene. Konstruieren Sie dafür eine Hüllkörperhierarchie (BVH) mit folgenden Kriterien: **(8 Punkte)**

- Unterteilen Sie abwechselnd senkrecht zur x- und zur y-Achse, beginnend mit der x-Achse.
- Verwenden Sie als Hüllkörper achsenorientierte Boxen (AABBs).
- Unterteilen Sie, wie Sie es von der Surface Area Heuristic erwarten würden.
- Unterteilen Sie nach zwei Schritten nur Knoten mit mehr als 8 Primitiven.
- Bei Fehlern können Sie die zweite Skizze nutzen. *Kennzeichnen Sie dann eindeutig, welche Skizze bewertet werden soll!*





b) Wir testen einen Strahl mit Ursprung **e** und Richtung **d** im Zweidimensionalen auf einen Schnitt mit einem achsenorientierten Hüllquader mit Begrenzungen $X_{1,2}$ und $Y_{1,2}$. Dafür werden die Strahlparameter t_{near} und t_{far} (siehe Skizze) wie in der Vorlesung berechnet:

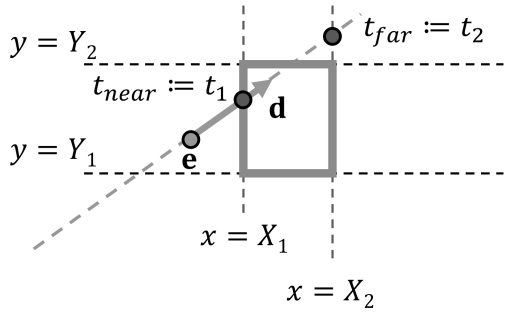
$t_{near} = -\infty$
$t_{far} = \infty$

Für jede Dimension, hier gezeigt für X:

$t_1 \leftarrow (X_1 - e_x)/d_x$
$t_2 \leftarrow (X_2 - e_x)/d_x$

if $t_1 > t_2$: SWAP(t_1, t_2)

$t_{near} \leftarrow \max(t_1, t_{near})$
$t_{far} \leftarrow \min(t_2, t_{far})$



☐ i) Wie können Sie anhand von t_{near} und t_{far} erkennen, ... **(6 Punkte)**

... dass die Box vom Strahl geschnitten wird, aber komplett hinter dem Strahlursprung liegt?	
... dass die Box vom Strahl geschnitten wird und der Strahlursprung in der Box liegt?	
... dass die Box vom Strahl überhaupt nicht geschnitten wird?	
... dass die Box vom Strahl in Strahlrichtung geschnitten wird, und der Strahl außerhalb der Box startet?	

☐ ii) Geben Sie für den letzen Fall aus i) an, wie der Punkt p berechnet wird, an dem der Strahl in die Box eintritt! **(2 Punkte)**

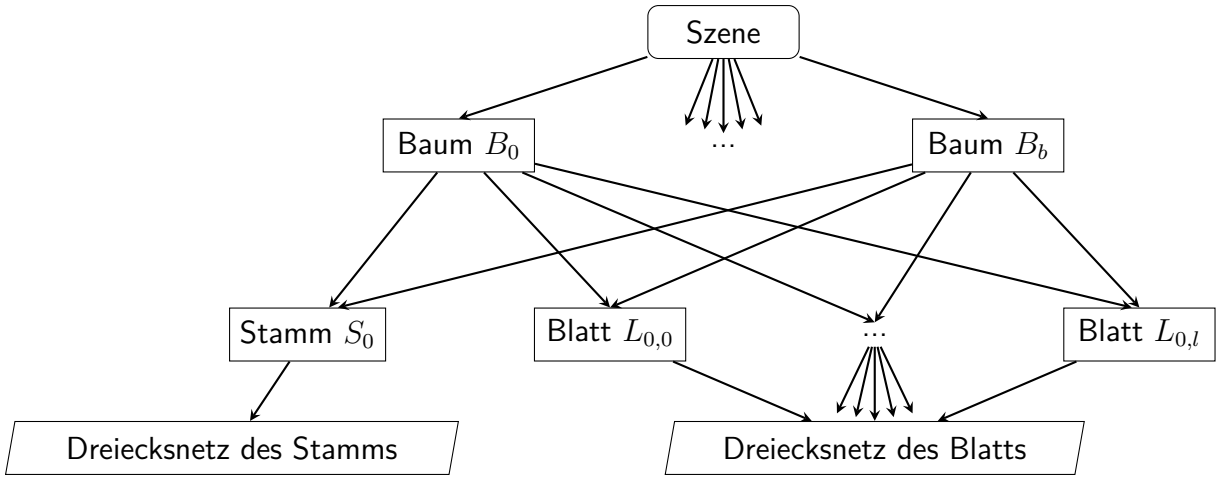
$p =$

c) Vergleichen Sie das Vorgehen bei den Beschleunigungsstrukturen BVH und kD-Baum, wenn bei der Traversierung ein Schnittpunkt des Strahls mit einem Dreieck gefunden wurde, der im aktuellen Teilraum bzw. Hüllkörper liegt: Handelt es sich dabei immer um den gesuchten nahsten Schnittpunkt? Begründen Sie Ihre Antwort! Beschreiben Sie, wie gegebenenfalls die Traversierung fortgesetzt werden muss! **(12 Punkte)**

i) kD-Baum:

ii) BVH:

Aufgabe 5: Transformationen (11 Punkte)



Der abgebildete Szenengraph repräsentiert einen Wald mit b Instanzen eines Baumes. Jede Baum-Instanz besteht aus einem Stamm und l Instanzen eines Blatts. Der Szenengraph speichert außerdem Transformationsmatrizen:

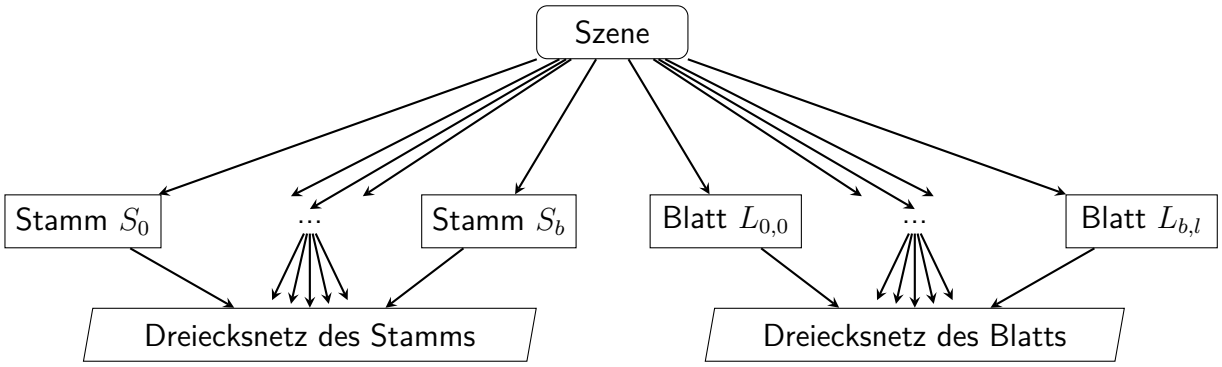
- B_i ist die Transformationsmatrix des i-ten Baums aus Baum- in Weltkoordinaten, und
- $L_{i,j}$ ist die Transformationsmatrix des j-ten Blatts am i-ten Baum aus Blatt- in Baumkoordinaten.

a) Gegeben ist ein Vertex $v_{i,j}$ und eine Normale $n_{i,j}$ in lokalen Koordinaten des j-ten Blatts am i-ten Baum. Geben Sie die Transformationsmatrizen an, die den Vertex und die Normale in Weltkoordinaten transformieren! **(5 Punkte)**

b) Wie viele Matrizen sind in diesem Szenengraph gespeichert? **(3 Punkte)**

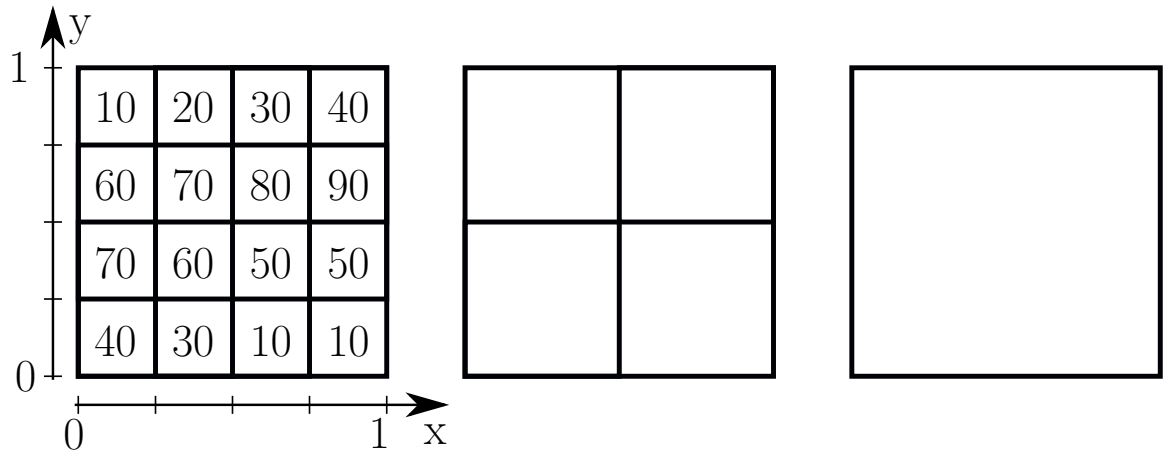
Matrikelnummer: _____

c) Bei Raytracing auf Grafikhardware können Instanzen keine weiteren Instanzen aufrufen und der Szenengraph muss deshalb wie in der Abbildung unten umstrukturiert werden. Wie viele Matrizen sind in diesem umstrukturierten Szenengraph gespeichert? **(3 Punkte)**



Aufgabe 6: Texturen und Baryzentrische Koordinaten (17 Punkte)

a) Gegeben ist die folgende 4×4-Textur:

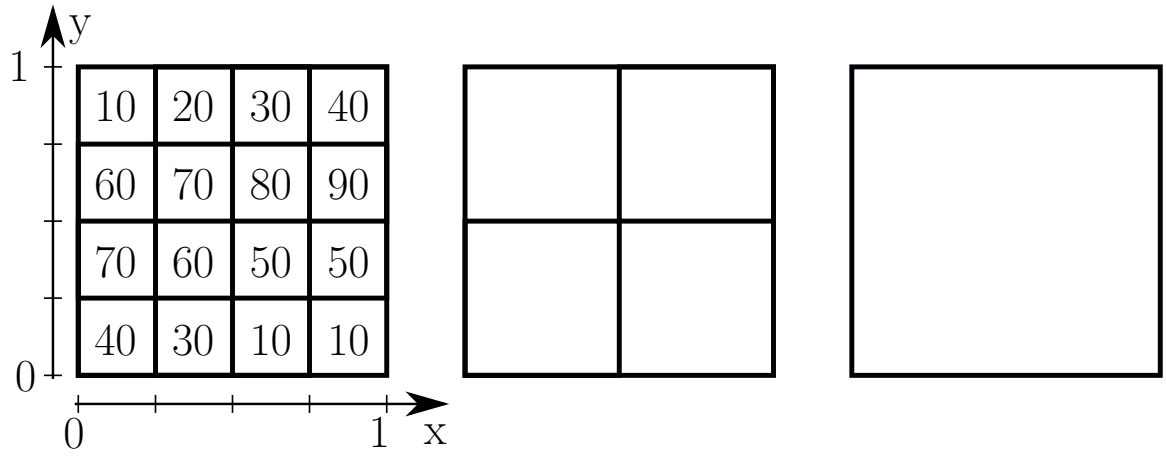


i) Geben Sie den ausgelesenen Texturwert für die Texturkoordinaten $(1.625, 1.625)$ an, wenn der Clamping- bzw. Repeat-Modus beim Zugriff im Nearest Neighbor Modus verwendet wird! **(4 Punkte)**

Clamp:

Repeat:

ii) Vervollständigen Sie in der Abbildung die Mipmap-Stufen der Textur! Zur Korrektur können Sie die zweite Abbildung verwenden. *Kennzeichnen Sie dann eindeutig, welche Lösung bewertet werden soll!* **(5 Punkte)**



Matrikelnummer: _____

b) Gegeben sind die Punkte $p, x, y, z \in \mathbb{R}^3$. Die Punkte x, y und z spannen ein Dreieck mit Flächeninhalt größer 0 auf. Schreiben Sie einen Algorithmus in Pseudocode, der `True` zurück gibt, falls p innerhalb des Dreiecks $\triangle(x, y, z)$ liegt, und sonst `False`! Die Hilfsfunktion **`float area`**`(vec3 a, vec3 b, vec3 c)` gibt den Flächeninhalt des Dreiecks $\triangle(a, b, c)$ zurück. Für Punkte p, für die kein Aufruf von `area` nötig ist, soll vorzeitig `False` zurückgegeben werden. Sie dürfen in Ihrem Pseudocode auch übliche mathematische Operatoren, sowie Funktionen, wie Sie sie aus GLSL kennen, verwenden. **(8 Punkte)**

```
float area(vec3 a, vec3 b, vec3 c);

bool triangleTest(vec3 p, vec3 x, vec3 y, vec3 z) {




}
```

Aufgabe 7: OpenGL-Pipeline (17 Punkte)

a) Diese Aufgabe behandelt die OpenGL-Pipeline.

i) Bringen Sie die folgenden Pipeline-Stufen in die richtige Reihenfolge und kennzeichnen Sie die programmierbaren Stufen! **(5 Punkte)**

(B) Blending, (C) Clipping, (F) Fragment Shader, (V) Vertex Shader,
(G) Geometry Shader

	Stufe	Programmierbar?
1		
2		
3		
4		
5		

ii) Begründen Sie, warum Stufe 1 und 2 vor Stufe 3 durchgeführt werden müssen! **(3 Punkte)**

iii) Begründen Sie, warum Stufe 3 vor Stufe 4 durchgeführt werden muss! **(3 Punkte)**

Matrikelnummer: _____

b) Nennen Sie für jeden der folgenden Anwendungsfälle, ob dieser jeweils entweder im **Vertex-** oder im **Geometry-Shader** sinnvoll durchgeführt werden kann und begründen Sie Ihre Antworten!

i) Eine große Anzahl an Instanzen eines Dreiecksnetzes wird gezeichnet. **(3 Punkte)** ☐

ii) Ein Partikelsystem generiert eine Million Rauchpartikel, die mit folgendem OpenGL-Befehl gezeichnet werden:

```
glDrawArrays(GL_POINTS, 0, 1000000);
```

Für jeden der Punkte soll ein kameraorientiertes Viereck an dessen Position generiert werden, auf das später im Fragment Shader eine Textur gezeichnet wird. **(3 Punkte)** ☐

Aufgabe 8: Texture-based Volume Rendering mit Blending (12 Punkte)

Ziel dieser Aufgabe ist die volumetrische Darstellung eines medizinischen Datensatzes durch Alpha Blending von Schichtbildern. Wie in Abbildung 1 gezeigt, wird dazu jedes Schichtbild entlang der Z-Achse versetzt als Textur auf ein Rechteck gezeichnet.

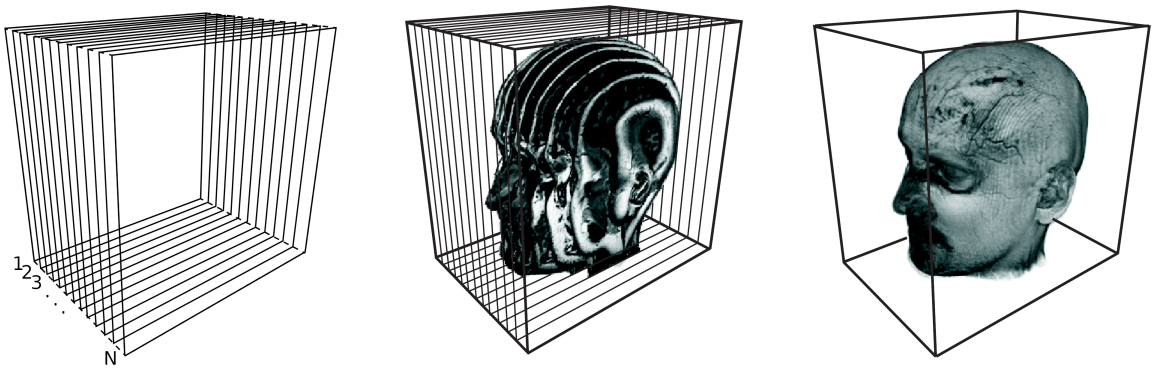


Abbildung 1: Darstellung eines Volumens durch einen Stapel von 2D Texturen. Links, Geometrie der Szene. Mitte, Ergebnis nach Texturierung der Rechtecke. Rechts, Ergebnis bei deutlicher Erhöhung der Zahl der Schichtbilder und Verwendung von Alpha Blending.

a) In welcher Reihenfolge müssen die Schichtbilder gezeichnet werden? Warum ist das Zeichnen in einer bestimmten Reihenfolge für korrekte Darstellung notwendig? **(3 Punkte)**

b) Konfigurieren Sie die Alpha Blending-Operation für vormultipliziertes Alpha Blending (also für den Fall, dass die Farbwerte in den Texturen vormultipliziert sind)! **(3 Punkte)**

```
glEnable(GL_BLEND)
glBlendEquation(                    )
glBlendFunc(                    ,                    )
```

Matrikelnummer: _____

c) Nun werden zusätzlich zu den semitransparenten Schichtbildern auch opake geometrische Primitive, wie beispielsweise die VR-Controller, mittels OpenGL gezeichnet. Warum muss opake Geometrie für korrekte Tiefendarstellung nicht sortiert werden? **(3 Punkte)** ☐

d) Müssen die opaken geometrischen Primitive vor oder nach den semitransparenten Schichtbildern gezeichnet werden? Begründen Sie! **(3 Punkte)** ☐

Aufgabe 9: GLSL-Shader (33 Punkte)

Hinweis 1: Die Teilaufgaben können unabhängig voneinander bearbeitet werden.
Hinweis 2: Sie dürfen in jeder Teilaufgabe Funktionen aus vorherigen Teilaufgaben verwenden.

In dieser Aufgabe vervollständigen Sie Raycasting mittels Sphere Tracing in einem Fragment Shader so, dass die Schattierung mit Ambient Occlusion und vorgefilterten Environment Maps durchgeführt wird.

Der Ablauf im Shader ist wie folgt:

```
uniform vec3 cam_pos;
out vec4 frag_color;

void main(){
    vec3 d = getFragmentDir();
    d = normalize(d);
    vec3 p = sphereTrace(cam_pos, d);
    if (p.x == FLOAT_MAX) frag_color = vec3(0.0);
    else frag_color = ambientOcclusion(p) + environment(p, d);
}
```

- Der Fragment Shader wird für jeden Pixel einmal ausgeführt.
- **vec3** `getFragmentDir()` gibt die Strahlrichtung in Weltkoordinaten zurück, in die der Kamerastrahl verfolgt wird.

Die Geometrie wird durch eine vorzeichenbehaftete Distanzfunktion beschrieben. In den Teilaufgaben können im Shader folgende Hilfsfunktionen verwendet werden:

- **float** `distfield(`**vec3** `p)` wertet die Distanzfunktion an der Stelle `p` aus.
- **vec3** `gradient(`**vec3** `p)` wertet den Gradienten der Distanzfunktion an `p` aus.
- **vec3** `random_dir()` erzeugt eine zufällige Richtung r mit $||r|| = 1$ und $r.z \geq 0$.

Matrikelnummer: _____

a) Bestimmen Sie in `sphereTrace(o, d)` mittels Sphere Tracing den nächsten Schnittpunkt entlang des Strahls mit Ursprung `o` und normalisierter Richtung `d`! Beenden Sie die Suche, wenn die Distanz kleiner als `epsilon` ist! Falls Sie nach 1000 Schritten keinen Schnittpunkt gefunden haben, gehen Sie davon aus, dass kein Schnitt existiert und es wird `FLOAT_MAX` zurückgegeben! **(9 Punkte)**

```
float epsilon = 0.001;

vec3 sphereTrace(vec3 o, vec3 d) {

    for(int step = 0; step < 1000; step++) {









    }

    return vec3(FLOAT_MAX); // kein Schnittpunkt
}
```

b) Geben Sie in `normal(vec3 p)` die Normale so zurück, dass sie direkt in der Beleuchtungsberechnung für den Oberflächenpunkt `p` verwendet werden kann! **(2 Punkte)**

```
vec3 normal(vec3 p) {









}
```

c) Berechnen Sie Ambient Occlusion an einem Schnittpunkt `p`, indem Sie `N` zufällige Strahlen oberhalb der Oberfläche des Schnittpunkts verfolgen. Bestimmen Sie die Anzahl $\texttt{M} \leq \texttt{N}$ der Strahlen, die keine andere Fläche schneiden! **(12 Punkte)**

Gehen Sie dazu wie folgt vor:

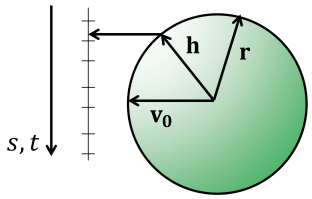
- Berechnen Sie ein orthonormales Koordinatensystem am Schnittpunkt `p` mit der Normale bei `p` als z-Achse des Koordinatensystems. Der Vektor `u` ist nicht parallel zur z-Achse und kann zur Bestimmung der Transformationsmatrix verwendet werden.
- Transformieren Sie `N` zufällige Richtungen von `random_dir` in Weltkoordinaten.
- Testen Sie, ob die transformierte Richtung keine andere Fläche mehr schneidet!

```
vec3 ambientOcclusion(vec3 p) {
    const int N = 100;

    vec3 x, y, z;

    z = normal(p);

    vec3 u = nonParallel(z);

    // Bestimmen Sie Basisvektoren x und y
    // für ein orthonormales Koordinatensystem aus x, y und z

    x =



    y =



    float M = 0;
    for(i = 0; i < N; i++) {
        vec3 d = randomDir(); // d.z >= 0

        // Transformieren Sie d in Weltkoordinaten wi
        vec3 wi;
```

```
        // Testen Sie, ob ein Strahl entlang wi keinen Schnitt mehr hat.
        // Falls ja, erhöhen Sie den Zähler M. Vermeiden Sie Selbstverschattung.









    }

    return vec3(M / float(N)); // Ambient Occlusion-Wert
}
```

d) In dieser Aufgabe sollen Sie die Beleuchtung durch Environment Maps auswerten.

i) Die Environment Maps sind als Sphere Map parametrisiert (siehe Skizze rechts). Der Texel in der Texturmitte entspricht der Richtung $\mathbf{v_0} = (0, 0, 1)$. Wandeln Sie in `texCoord(`**`vec3`**` d)` die übergebene Richtung in Texturkoordinaten für Sphere Maps um!



(5 Punkte)

```
vec2 texCoord(vec3 d) {
    // Richtung d -> Texturkoordinaten für Sphere Map




}
```

Platz für Notizen:

ii) Gegeben sind die Environment Map `envmap`, die vorgefilterte Environment Map `envmapGlossy` für glänzend-imperfekte Reflexionen, sowie die vorgefilterte Environment Map `envmapDiffuse` für diffuse Beleuchtung. Berechnen Sie in `environment` die Textkurkoordinaten für die Zugriffe auf die drei Texturen und setzen Sie diese in die `texture`-Aufrufe ein, um die Gesamtbeleuchtung auszuwerten! **(5 Punkte)**

```
uniform sampler2D envmap;
uniform sampler2D envmapGlossy;
uniform sampler2D envmapDiffuse;

// p: Oberflächenpunkt
// d: normalisierte eingehende Betrachtungsrichtung
vec3 environment(vec3 p, vec3 d) {
    vec3 n = normal(p);
    vec3 r = reflect(d, n);

    vec3 specular = vec3(texture(envmap,                    ));

    vec3 glossy = vec3(texture(envmapGlossy,                ));

    vec3 diffuse = vec3(texture(envmapDiffuse,              ));

    return specular + glossy + diffuse;
}
```

Aufgabe 10: Prozedurale Modellierung (9 Punkte)

Wir erzeugen eine Graustufen-Textur durch eine Turbulenzfunktion, basierend auf einer Noise-Funktion n:

$$f(\mathbf{x}, K) = \sum_{k=1}^{K} \frac{1}{2^k}\, n(2^k\, \mathbf{x}).$$

a) Welche einfache Möglichkeit gibt es, eine solche Turbulenzfunktion zu filtern, um Aliasing zu vermeiden? **(3 Punkte)**

b) Welche Eigenschaft muss die Noise-Funktion haben, damit sich dabei die Gesamthelligkeit der Textur nicht verändert? **(3 Punkte)**

c) Bei der einfachen Variante wird es zu sichtbaren Sprüngen kommen. Wie könnten diese kaschiert werden? **(3 Punkte)**

Matrikelnummer: _____

Aufgabe 11: Bézier-Kurven (17 Punkte)

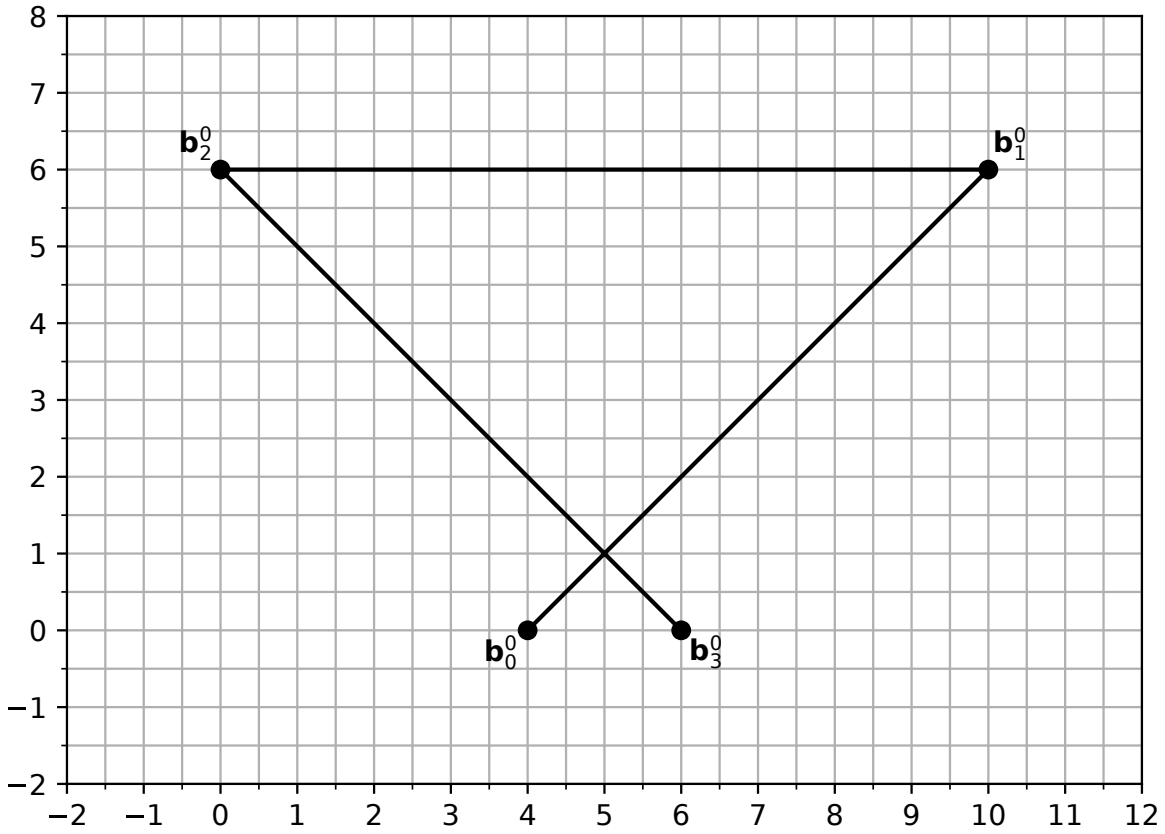
a) Gegeben sei die Bézier-Kurve $F(u) = \sum_{i=0}^{3} \mathbf{b}_i B_i^3(u)$ vom Grad 3 mit $u \in [0, 1]$ und

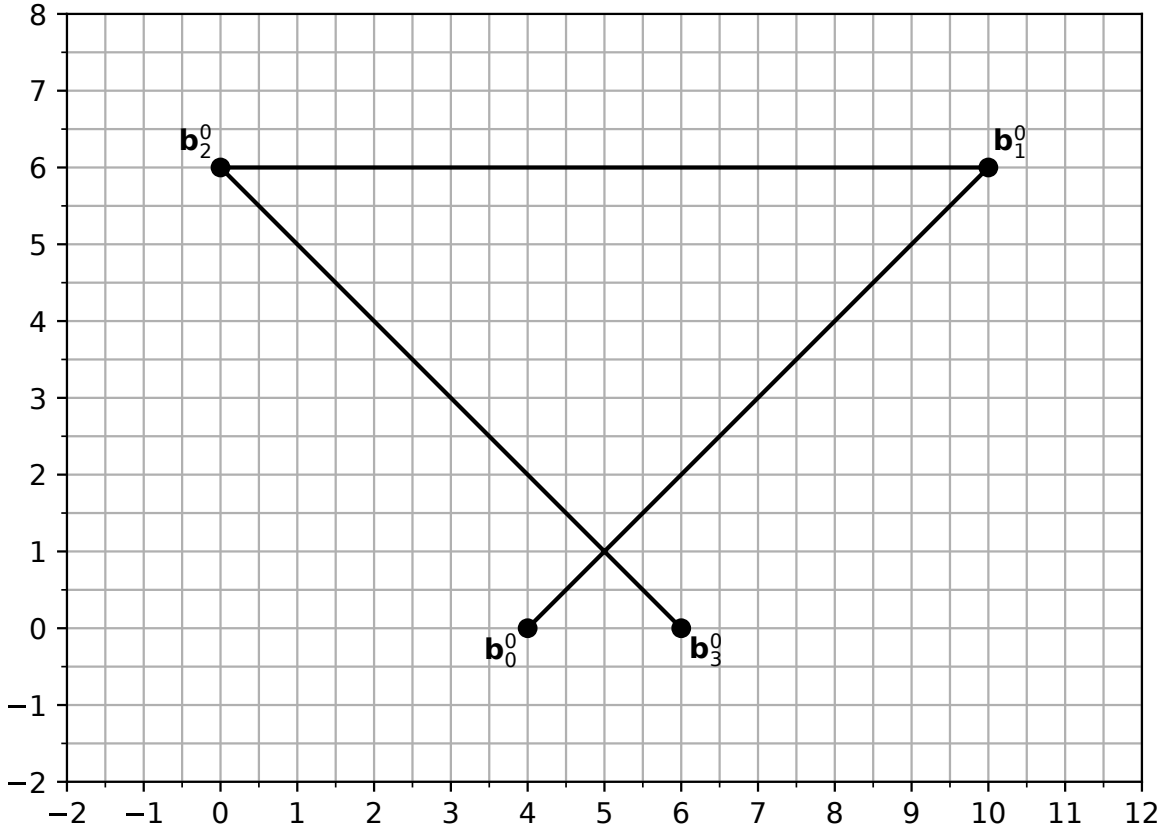
$$\mathbf{b}_0 = (4, 0), \qquad \mathbf{b}_1 = (10, 6), \qquad \mathbf{b}_2 = (0, 6), \qquad \mathbf{b}_3 = (6, 0).$$

Bestimmen Sie den Punkt $F(0.5)$, indem Sie die Kurve *zeichnerisch* einmal in der parametrischen Mitte $u = 0.5$ unterteilen. Machen sie dabei die Kanten und Punkte der Kontrollpolygone der Teilkurven analog zum bereits eingezeichneten Kontrollpolygon von $F(u)$ kenntlich.

Nutzen Sie anschließend die Kontrollpolygone der Teilkurven, um $F(u)$ unter Ausnutzung der Eigenschaften von Bézier-Kurven zu skizzieren!

Zur Korrektur können Sie die zweite Zeichnung verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnung bewertet werden soll!* **(7 Punkte)**




$\mathbf{b}_2^0$
$\mathbf{b}_1^0$
$\mathbf{b}_0^0$
$\mathbf{b}_3^0$

Matrikelnummer: _____

b) Gegeben sei die Bézier-Kurve $G(u) = \sum_{i=0}^{3} \mathbf{c}_i B_i^3(u)$ vom Grad 3 mit $u \in [0, 1]$ und

$$\mathbf{c}_0 = (0, 1), \qquad \mathbf{c}_3 = (1, 0).$$

Bestimmen Sie die Kontrollpunkte $\mathbf{c}_1$ und $\mathbf{c}_2$ so, dass ein symmetrischer Kreisbogen – wie in der Abbildung gezeigt – angenähert wird. Dabei sollen die Tangentenrichtungen bei $\mathbf{c}_0$ und $\mathbf{c}_3$ korrekt sein und die kubische Beziér-Kurve durch den Punkt $\mathbf{p} = (1/\sqrt{2}, 1/\sqrt{2})$ führen. **(10 Punkte)**