Karlsruher Institut für Technologie

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

Nachklausur Computergrafik — WS 2013/14

09. April 2014

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.
- 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	12	13	7	9	7	16	9	14	4	8	11	10	**120**

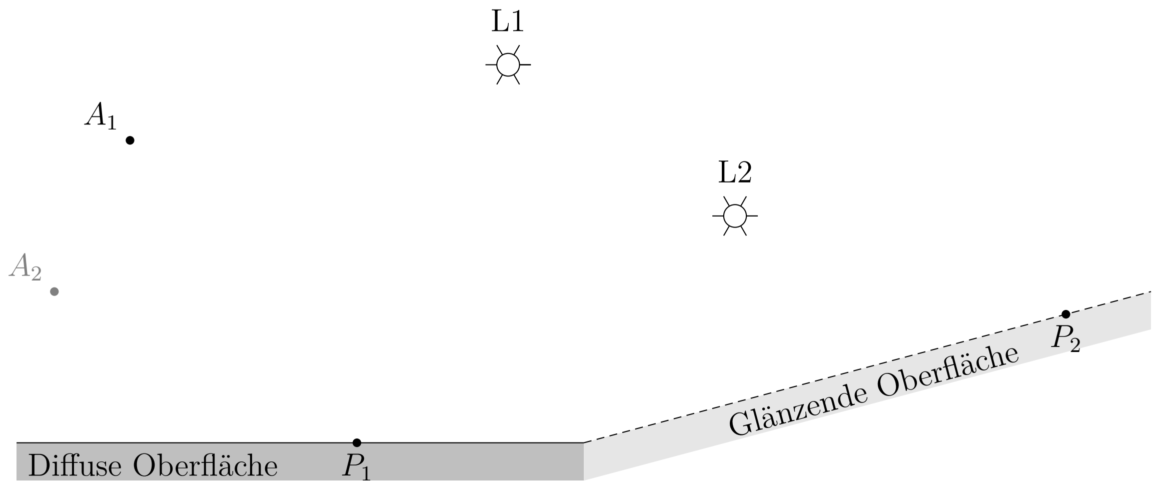
Note

Aufgabe 1: Das Phong-Beleuchtungsmodell (12 Punkte)

In der Abbildung unten ist eine Szene gegeben, die durch Raytracing dargestellt werden soll. Darin befinden sich zwei Augpunkte (A_1 und A_2), zwei Oberflächen, die eine davon diffus und die andere glänzend, sowie zwei Punktlichter (L_1 und L_2), die *beide eine Intensität von 1* haben.

Für die Berechnung von Oberflächenfarben soll das Phong-Beleuchtungsmodell verwendet werden. Der Materialparameter k_a ist für beide Oberflächen 0. Bei der *diffusen Oberfläche* ist $k_d > 0$ und $k_s = 0$. Bei der *glänzenden Oberfläche* sind $k_d > 0$ und $k_s > 0$. Sie müssen für beide Lichter *keine* entfernungsabhängige Abschwächung berücksichtigen.

Hinweis:
Gehen Sie für die folgenden Aufgaben davon aus, dass die Szene aus dem **Augpunkt $\mathbf{A_1}$** betrachtet wird, solange die Aufgabe den Augpunkt nicht explizit angibt.



a) Zeichnen Sie für den Punkt P_1 auf der diffusen Fläche alle Vektoren aus Komponenten des Beleuchtungsmodells ein, die zur Farbe des Punktes tatsächlich beitragen! Benennen Sie diese Vektoren! **(2 Punkte)**

b) Geben Sie mit Hilfe der benannten Vektoren die Formel an, mit der die Farbe der Oberfläche im Punkt P_1 bestimmt werden kann! **Unterstreichen** Sie dabei die Vektoren, die normiert sein *müssen*! **(2 Punkte)**

c) Welche der beiden Lichtquellen trägt mehr zur Intensität der Oberfläche an P_1 bei? Begründen Sie Ihr Antwort! **(2 Punkte)**

d) Wie verändert sich die Intensität von P_1, wenn er statt von A_1 nun von A_2 betrachtet wird? Begründen Sie Ihre Antwort *kurz*! **(2 Punkte)**

e) Es soll nun die Farbe des Punktes P_2 bestimmt werden. Dieser befindet sich auf der glänzenden Oberfläche. Zeichnen Sie für P_2 die Vektoren ein, die zur Bestimmung seiner Farbe benötigt werden! Beschränken Sie sich dabei auf die Vektoren für L_1. **(2 Punkte)**

f) Wie verändert sich die Intensität von P_2, wenn er statt von A_1 von A_2 betrachtet wird? Begründen Sie Ihre Antwort *kurz*! **(2 Punkte)**

Aufgabe 2: Raytracing (13 Punkte)

Beantworten Sie die folgenden Fragen zum Thema Raytracing:

a) Was versteht man unter „uniformem Supersampling“?
Welches Problem wird damit verringert? **(4 Punkte)**

b) Nennen Sie zwei Kriterien, die Sie in der Vorlesung kennengelernt haben, um die Rekursion bei Whitted-Style Raytracing abzubrechen! **(3 Punkte)**

c) Was ist der Unterschied zwischen Distributed Raytracing und Whitted-Style Raytracing? Welchen Lichttransport kann man durch Distributed Raytracing berechnen, den Whitted-Style Raytracing nicht erfassen kann? **(2 Punkte)**

Name: ______________________ Matrikelnummer: ______________

d) Nennen Sie kurz und stichpunktartig die zwei Schritte, die zur Berechnung von Vertex-Normalen bei einem Dreiecksnetz notwendig sind! Gehen Sie dabei davon aus, dass nur die Vertex-Positionen und die Topologie des Netzes gegeben sind! **(4 Punkte)**

Aufgabe 3: Farben und Farbwahrnehmung (7 Punkte)

a) Nehmen Sie eine Kamera an, die mit Sensoren Licht im Bereich [380nm, 780nm] erfassen kann. Die *Empfindlichkeitskurve* eines Sensors ist gegeben durch die Funktion

$$E(\lambda) : [380\text{nm}, 780\text{nm}] \mapsto \mathbb{R}$$

I) Wie berechnet man die Sensorantwort a für ein Spektrum $S(\lambda)$? **(2 Punkte)**

II) Gegeben sind nun zwei Spektren, $S_1(\lambda)$ und $S_2(\lambda)$, sowie ihre Sensorantworten a_1 und a_2. Welcher mathematische Zusammenhang muss zwischen den Sensorantworten bestehen, damit S_1 und S_2 im Bezug auf die gegebene Kamera Metamere sind? **(1 Punkt)**

b) Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen. **(4 Punkte)**

Aussage	**Wahr**	**Falsch**
Das HSV-Farbmodell trennt Farbton von Helligkeit.	☐	☐
Der Farbeindruck einer additiv gemischten Farbe hängt nicht vom Farbeindruck der Ausgangsfarben ab.	☐	☐
Farbige Flächen werden unabhängig von ihrer Umgebung vom menschlichen Auge immer gleich wahrgenommen.	☐	☐
Der Machsche Bandeffekt ist vor allem bei Phong-Shading ein Problem.	☐	☐

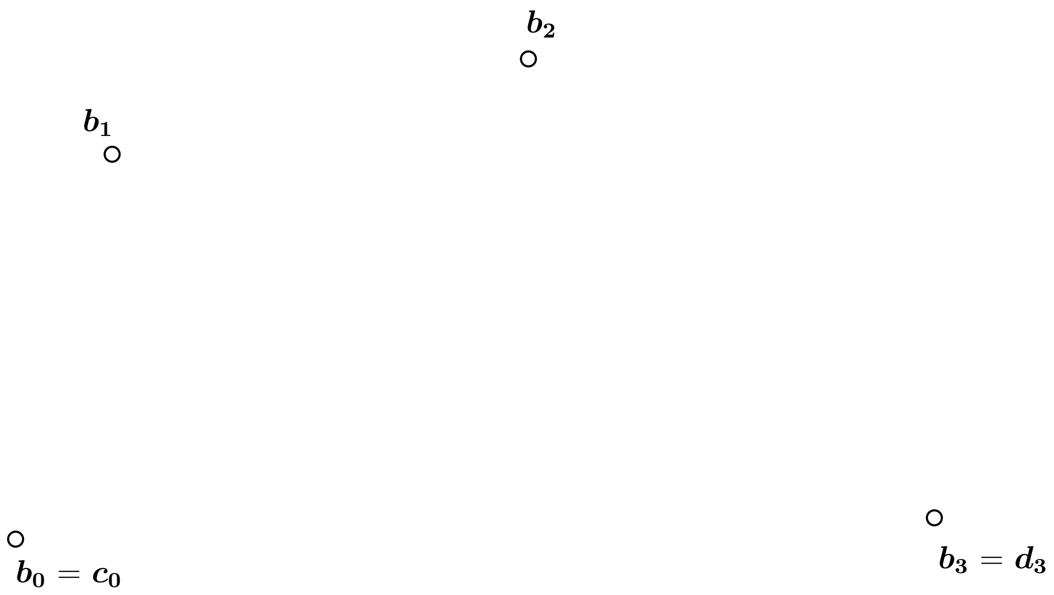
Name: ____________________ Matrikelnummer: ____________________

Aufgabe 4: Bézier-Kurven (9 Punkte)

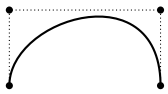
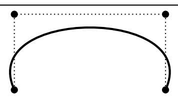
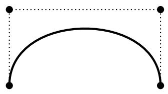
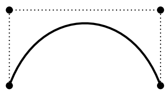
a) Gegeben sei die Bézier-Kurve $\boldsymbol{b}(u) = \sum_{i=0}^{3} \boldsymbol{b}_i B_i^3(u)$ mit den Kontrollpunkten $\boldsymbol{b}_i$, wobei $u \in [0, 1]$ und B_i^3 das i-te Bernstein-Polynom vom Grad 3 ist.

I) Werten Sie die Bézier-Kurve zeichnerisch mit dem **de-Casteljau-Algorithmus** an der Stelle $u = \frac{1}{3}$ aus! **Markieren** Sie den Punkt $\boldsymbol{b}(\frac{1}{3})$! **(3 Punkte)**

II) Bei Unterteilung der Bézier-Kurve an der Stelle $u = \frac{1}{3}$ ergeben sich zwei Teilkurven. Diese werden jeweils wieder von einem Kontrollpolygon beschrieben. Zeichnen Sie die Kontrollpunkte $\boldsymbol{c_0}$ bis $\boldsymbol{c_3}$ und $\boldsymbol{d_0}$ bis $\boldsymbol{d_3}$ der beiden neuen Kontrollpolygone ein! **(2 Punkte)**



b) Geben Sie an, ob es sich bei den folgenden Kurven mit gegebenem Kontrollpolygon um Bézier-Kurven handelt. Begründen Sie jeweils kurz Ihre Antwort, falls es sich *nicht* um eine Bézier-Kurve handelt! **(4 Punkte)**

Kurve	**Ja**	**Nein**	**Begründung**
	☐	☐	
	☐	☐	
	☐	☐	
	☐	☐	

Name: ______________________ Matrikelnummer: ______________

Aufgabe 5: Transformationen (7 Punkte)

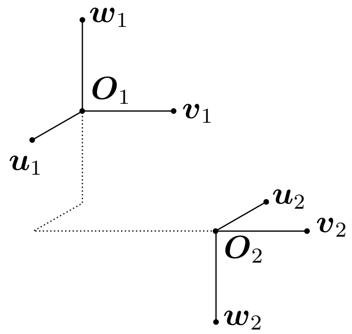
Gegeben sind zwei Koordinatensysteme mit den Ursprüngen

$$\boldsymbol{O}_1 = \begin{pmatrix} -2 \\ 1 \\ -1 \end{pmatrix} \quad \text{und} \quad \boldsymbol{O}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

und den orthonormalen Basisvektoren

$$\boldsymbol{u}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \boldsymbol{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \boldsymbol{w}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\boldsymbol{u}_2 = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad \boldsymbol{v}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \boldsymbol{w}_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}.$$



Gesucht ist die *homogene* Matrix $M \in \mathbb{R}^{4\times 4}$, die Koordinaten eines Punktes $\boldsymbol{P}_1$, gegeben im System $S_1 = (\boldsymbol{O}_1, \boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1)$, in das System $S_2 = (\boldsymbol{O}_2, \boldsymbol{u}_2, \boldsymbol{v}_2, \boldsymbol{w}_2)$ transformiert:

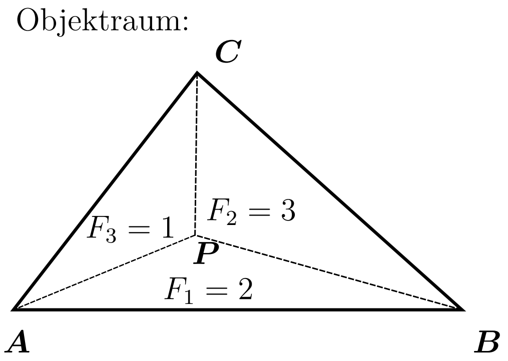
$$\boldsymbol{P}_2 = M \cdot \boldsymbol{P}_1.$$

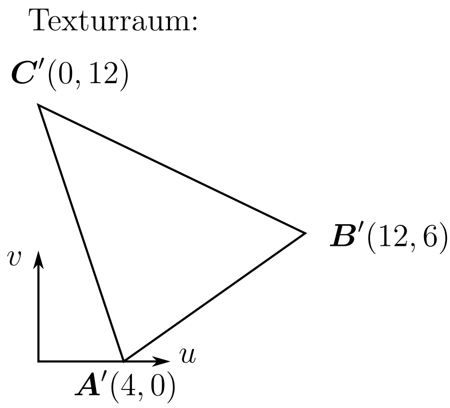
Geben Sie die endgültige Transformationsmatrix M sowie den Rechenweg und die dabei benötigten Matrizen an! Beschreiben Sie stichpunktartig, welche Transformation diese Matrizen beschreiben! Matrixmultiplikationen (sofern vorhanden) müssen Sie nicht ausrechnen. **(7 Punkte)**

Aufgabe 6: Texturierung (16 Punkte)

a) Gegeben ist ein Dreieck mit den Eckpunkten $\boldsymbol{A}$, $\boldsymbol{B}$ und $\boldsymbol{C}$ (siehe Abbildung). Den Eckpunkten sind die zweidimensionalen Texturkoordinaten $\boldsymbol{A'}$, $\boldsymbol{B'}$ und $\boldsymbol{C'}$ zugewiesen. Das Dreieck wird im Punkt $\boldsymbol{P}$ von einem Strahl geschnitten. F_1, F_2 und F_3 bezeichnen jeweils die Flächeninhalte der eingezeichneten Teildreiecke.

Berechnen Sie die baryzentrischen Koordinaten λ_A, λ_B und λ_C von $\boldsymbol{P}$ und daraus die Texturkoordinaten $\boldsymbol{P'} = (u, v)$ des Punktes $\boldsymbol{P}$! Geben Sie jeweils Ihren Rechenweg an! **(5 Punkte)**





$\lambda_A =$

$\lambda_B =$

$\lambda_C =$

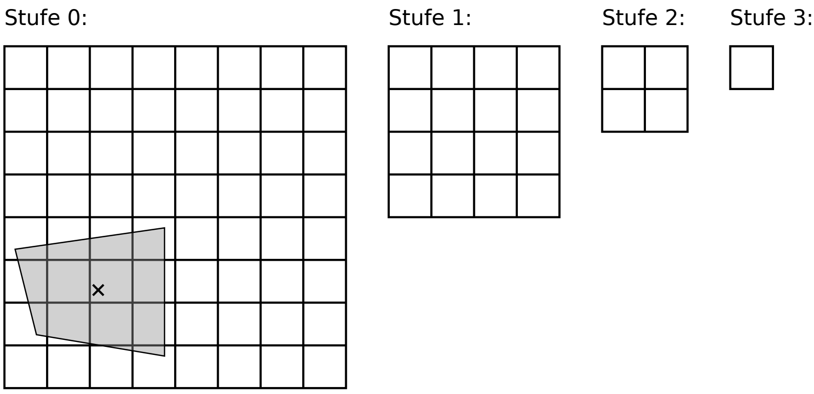
$u =$

$v =$

Name: ____________________ Matrikelnummer: ____________________

b) Gegeben ist eine Textur mit 8×8 Texeln und 4 Mipmap-Stufen sowie ein Pixel-Footprint und dessen Mittelpunkt (siehe Abbildung). Kreuzen Sie in der Mipmap-Pyramide die Texel an, die für eine *trilineare Interpolation* zur Bestimmung des Farbwertes verwendet werden! Begründen Sie *kurz* Ihre Wahl der Mipmap-Stufen! **(5 Punkte)**

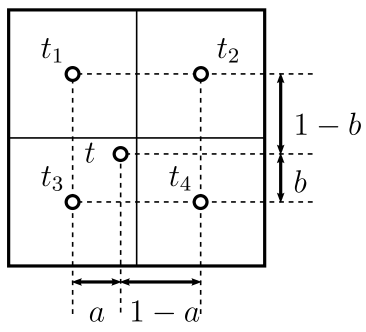
Hinweis:
Wenn Sie Korrekturen vornehmen wollen, zeichnen Sie bitte ein neues Gitter für die entsprechende Stufe und streichen das vorhandene deutlich durch.



Begründung:

c) Welchen Vorteil haben Summed-Area-Tables gegenüber Mipmaps bei der Texturfilterung? **(2 Punkte)**

d) Gegeben ist eine Textur mit vier Texeln. Diese besitzen jeweils in der Mitte die Werte t_1, t_2, t_3 bzw. t_4. Ermitteln Sie den interpolierten Wert an der Stelle t (abhängig von $t_1, \ldots, t_4$), indem Sie das einfachste aus der Vorlesung bekannte C^1-stetige Interpolationsschema für Texturen verwenden! Wie heißt dieses Interpolationsschema? **(4 Punkte)**



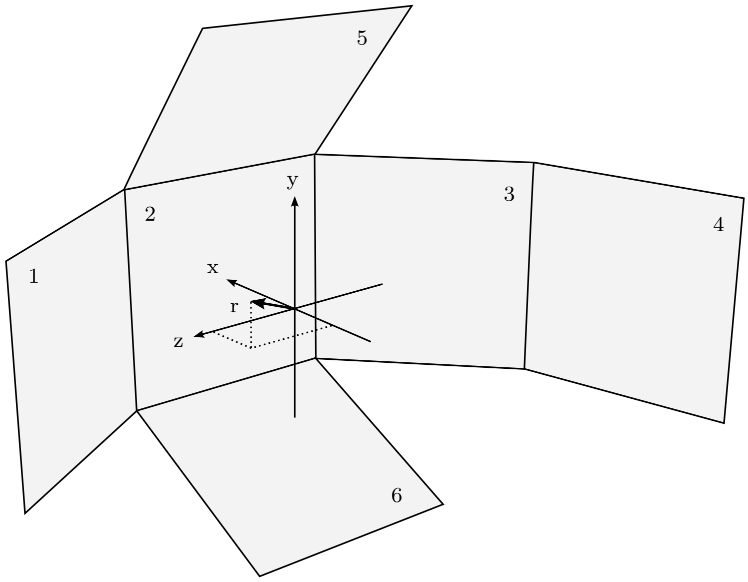
Name des Interpolationsschemas: ....

$t =$

Name: ____________________ Matrikelnummer: ____________________

Aufgabe 7: Cube-Maps und Environment-Mapping (9 Punkte)

a) In der Abbildung ist eine Cube-Map mit lokalem Koordinatensystem zu sehen.



Auf diese Cube-Map soll mit dem Richtungsvektor

$$\boldsymbol{r} = \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix}$$

zugegriffen werden.

I) Wie wird die Cube-Map-Seite bestimmt, auf die zugegriffen wird? Welche ist es für $\boldsymbol{r}$? **(2 Punkte)**

II) Berechnen Sie die Texturkoordinaten des Zugriffs auf der für $\boldsymbol{r}$ ausgewählten Cube-Map-Seite! **(2 Punkte)**

III) Nennen Sie einen Vorteil von Cube-Maps gegenüber Sphere-Maps! **(1 Punkt)**

b) Beantworten Sie die folgenden Fragen zum Thema Environment-Mapping:

I) Was wird in einer Environment-Map gespeichert? **(1 Punkt)**

II) Nennen Sie ein Anwendungsbeispiel für Environment-Maps! **(1 Punkt)**

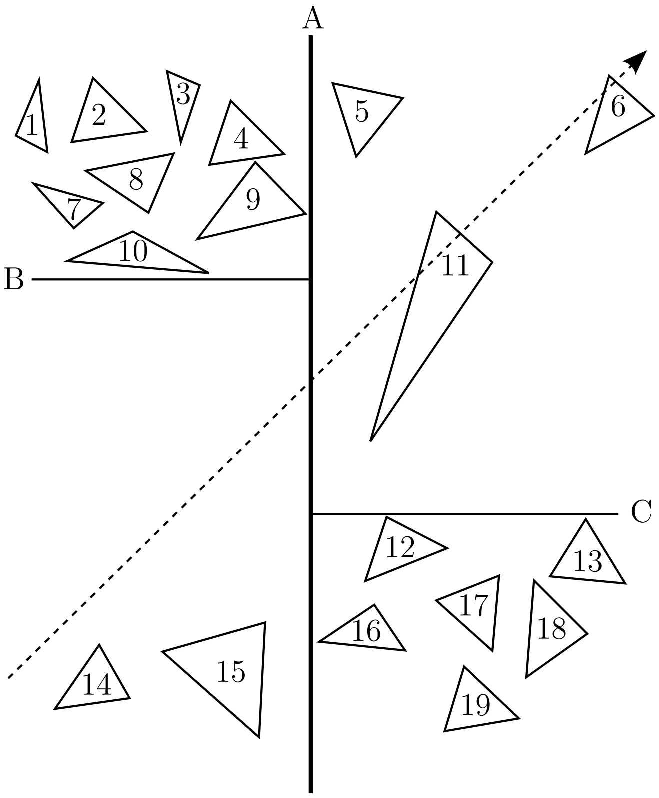
III) Welche grundlegende Annahme wird bei Environment-Mapping gemacht? **(1 Punkt)**

IV) Was bzw. welcher Effekt kann mit vorgefilterten Environment-Maps nicht korrekt dargestellt werden? **(1 Punkt)**

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 8: Hierarchische Datenstrukturen (14 Punkte)

Gegeben sei folgender kD-Baum:



a) Traversieren Sie den abgebildeten kD-Baum, um den Schnittpunkt des Strahls mit dem in Strahlrichtung ersten Dreieck zu ermitteln. Geben Sie hierfür *alle notwendigen Schnitttests* in Reihenfolge der Traversierung an. **(3 Punkte)**

Beispiel für die Angabe der Schnitttests (für einen anderen Strahl in einem anderen Baum):

A, 1, B, 2, C, 3

Reihenfolge der Schnitttests:..

b) Welche der folgenden Aufteilungsstrategien wurde bei der Konstruktion des abgebildeten Baumes verwendet? Begründen Sie jeweils kurz Ihre Antwort, falls es *nicht* die genannte Aufteilungsstrategie gewesen sein kann! **(3 Punkte)**

Aufteilungsstrategie	**Ja**	**Nein**	**Begründung**
räumliches Mittel (Spatial Median)	☐	☐	
Objektmittel (Object Median)	☐	☐	
Kostenfunktion (Surface Area Heuristic)	☐	☐	

c) Nennen Sie jeweils eine Stärke und eine Schwäche der Aufteilung mittels Kostenfunktion (Surface Area Heuristic)! **(4 Punkte)**

Name: ______________________ Matrikelnummer: ______________________

d) Kreuzen Sie jeweils die passenden Kästchen an! Sie erhalten für jede vollständig richtige Zeile 1 Punkt.

Hinweis:
Die Aussagen beziehen sich auf die Datenstrukturen, so wie Sie diese in der Vorlesung kennen gelernt haben. **(4 Punkte)** ☐

Aussage	**Hüllkörperhierarchie (BVH)**	**Oktalbaum (Octree)**	**kD-Baum**	**Gitter**
Der Aufbau-Algorithmus passt die Datenstruktur an die Geometrie an und der Raum wird immer achsenparallel partitioniert.	☐	☐	☐	☐
Die Datenstruktur wird durch einen Binärbaum repräsentiert.	☐	☐	☐	☐
Der Speicherbedarf der Datenstruktur ist abhängig von der Anzahl der Primitive.	☐	☐	☐	☐
Bei der Konstruktion kann die Surface Area Heuristic sinnvoll eingesetzt werden.	☐	☐	☐	☐

Aufgabe 9: Rasterisierung und OpenGL (4 Punkte)

Welche der folgenden Aussagen sind für eine moderne OpenGL-Pipeline wahr?

Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen. **(4 Punkte)**

Aussage	**Wahr**	**Falsch**
Die Präzision des Tiefenpuffers wird verringert, wenn man die Distanz zwischen Near-Plane und Far-Plane vergrößert.	☐	☐
Die OpenGL-Pipeline nutzt den Vertex-Cache beim Zeichnen *ohne* Index-Puffer.	☐	☐
OpenGL-Puffer vom Typ `GL_ELEMENT_ARRAY_BUFFER` werden für das Rendering mit der Shared-Vertex-Repräsentation verwendet.	☐	☐
Ein *Alpha-Test* kann im Fragment-Shader mit dem Befehl `discard` implementiert werden.	☐	☐
Die OpenGL-Funktion `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)` kann verwendet werden, um *additives* Blending zu implementieren.	☐	☐
Shadow-Mapping benötigt den Stencil-Puffer.	☐	☐
Für die Transformation einer Normalen im Vertex-Shader kann immer dieselbe Matrix wie zur Transformation der entsprechenden Vertizes verwendet werden.	☐	☐
Beim Gouraud-Shading werden die im Vertex-Shader berechneten Farben pro Fragment linear interpoliert.	☐	☐

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 10: Tiefenpuffer und Transparenz (8 Punkte)

In dieser Aufgabe soll ein Haus gezeichnet werden. Das Haus besteht aus opaken Wänden und semitransparenten Fenstern.

Die Fenster werden mit einem nicht-kommutativen Blending-Operator gezeichnet. Es ist möglich, durch mehrere Fenster hintereinander hindurchzusehen. Die Szene soll korrekt und so effizient wie möglich mit Hilfe eines Tiefenpuffers gezeichnet werden.

Ihnen stehen die Funktionen

1. **`LöscheTiefenPuffer`**,
2. **`ZeichneFenster`** und
3. **`ZeichneWände`**

zur Verfügung, die jeweils die ihrem Namen entsprechende Funktionalität bereitstellen.

a) In welcher Reihenfolge müssen die drei Funktionen aufgerufen werden, um eine korrekte Darstellung des Hauses zu erhalten? **(2 Punkte)**

b) Welche Sortierung der Geometrie müssen die Funktionen vornehmen? Kreuzen Sie jeweils die passende Spalte an und begründen Sie in jedem Fall Ihre Antwort *knapp*! **(4 Punkte)**

Sortierung	**keine**	**vorne nach hinten**	**hinten nach vorne**	**Begründung**
`ZeichneWände`	☐	☐	☐	
`ZeichneFenster`	☐	☐	☐	

c) Welcher Zustand der OpenGL-Pipeline müssen die Funktionen **`ZeichneWände`** und **`ZeichneFenster`** jeweils herstellen, damit die Darstellung korrekt ist und keine unnötigen bzw. falschen Schreiboperationen in den Tiefenpuffer stattfinden?

☐

Markieren Sie diejenigen Zustände, die aktiv sein müssen! **(2 Punkte)**

		Tiefenvergleich				
	Tiefentest	**`EQUAL`**	**`LESS`**	**`GREATER`**	**Tiefe schreiben**	**Blending**
`ZeichneWände`	☐	☐	☐	☐	☐	☐
`ZeichneFenster`	☐	☐	☐	☐	☐	☐

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 11: Phong-Shading und Phong-Beleuchtungsmodell (11 Punkte)

In dieser Aufgabe sollen Sie Phong-Shading mit dem Phong-Beleuchtungsmodell in der OpenGL Shading Language implementieren.

a) Vervollständigen Sie zunächst den Vertex-Shader! **(5 Punkte)**

```
uniform mat4 matN;   // Normalentransformation (Objekt -> Kamera)
uniform mat4 matM;   // Modelltransformation
uniform mat4 matV;   // Kameratransformation
uniform mat4 matP;   // Projektionstransformation
uniform mat4 matMV;  // Model-View-Matrix
uniform mat4 matMVP; // Model-View-Projection-Matrix

in vec3 P;  // Eingabe-Vertex in Objektkoordinaten
in vec3 n;  // Eingabenormale in Objektkoordinaten

out vec3 P_k; // Vertex-Position in Kamerakoordinaten
out vec3 n_k; // Vertex-Normale in Kamerakoordinaten

void main() {

  P_k =


  n_k =


  gl_Position =

}
```

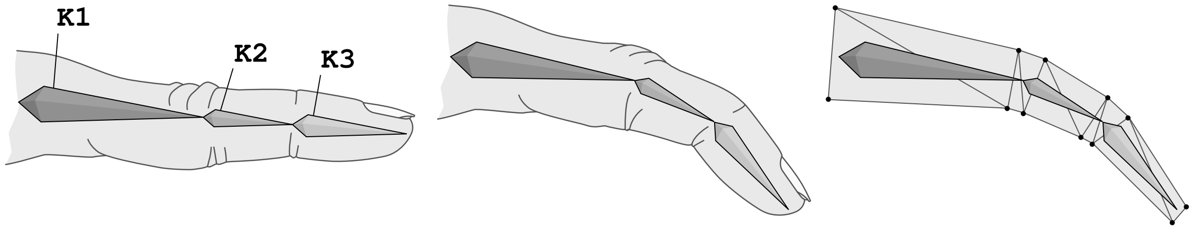
b) Vervollständigen Sie nun den Fragment-Shader! **(6 Punkte)**

```
uniform vec3  L;    // Lichtposition in Kamerakoordinaten

// Materialparameter
uniform vec3  ka, kd, ks;
uniform float pexp; // Phong Exponent

// Intensität der Lichtquelle
uniform vec3 intensity;

in vec3 P_k;
in vec3 n_k;

void main()
{

  vec3 n =


  vec3 v = normalize(                 );


  vec3 l = normalize(                 );


  vec3 r = reflect( -l, n );


  float cosAlpha = max(0.0, dot(v,


  vec3 diffuse = kd * max(0.0, dot(


  vec3 specular = ks *


  gl_FragColor = vec4(intensity*(ka + diffuse + specular), 1.0);
}
```

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 12: Deformation mit Skelettsystemen (10 Punkte)

Die Animation von deformierbaren Körpern wird oft über sogenannte *Skelettsysteme* bewerkstelligt. Dabei werden einzelne Vertizes eines Modells mehreren *Knochen* zugewiesen. Skelettsysteme sind hierarchisch, sodass Kindknochen die Transformation von Elternknochen übernehmen. Die folgende Skizze zeigt einen Finger, der über drei Knochen deformiert wird.



Der Knochen **`K1`** ist die Wurzel der Skeletthierarchie. Sein Kindknochen ist **`K2`**. **`K3`** ist das Kind von **`K2`**.

Es sind drei Transformationsmatrizen gegeben. **`M1`** beschreibt die Lage des gesamten Fingers in Weltkoordinaten. **`M2`** beschreibt die Lage von **`K2`** *relativ zu* **`K1`**. **`M3`** beschreibt die Lage von **`K3`** *relativ zu* **`K2`**.

Für jeden Vertex sind Gewichte **`w[0]`**, **`w[1]`** und **`w[2]`** gegeben. Sie beschreiben, wie viel Einfluss die Transformation des entsprechende Knochens auf den Vertex hat.

Vervollständigen Sie den Vertex-Shader, sodass die Position des Vertex **`P`** wie angegeben gewichtet transformiert wird! In **`gl_Position`** muss dann letztendlich die Position in Clip-Koordinaten geschrieben werden. **(10 Punkte)**

```
in vec3 P;          // Position des Vertex in Objektkoordinaten.
in float w[3];      // Einfluss der Knochen.

uniform mat4 matVP;  // View-Projection-Matrix.
uniform mat4 M1;     // Transformation des Fingers (Objekt -> Welt).
uniform mat4 M2;     // Transformation von K2 relativ zu K1.
uniform mat4 M3;     // Transformation von K3 relativ zu K2.

void main()
{




    gl_Position =
}
```