

Hauptklausur Computergrafik WS 2016/17

8. März 2017

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 20 Seiten (10 Blätter) mit 10 Aufgaben.
- Es sind **keine Hilfsmittel** zugelassen.
- Sie haben **90 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.

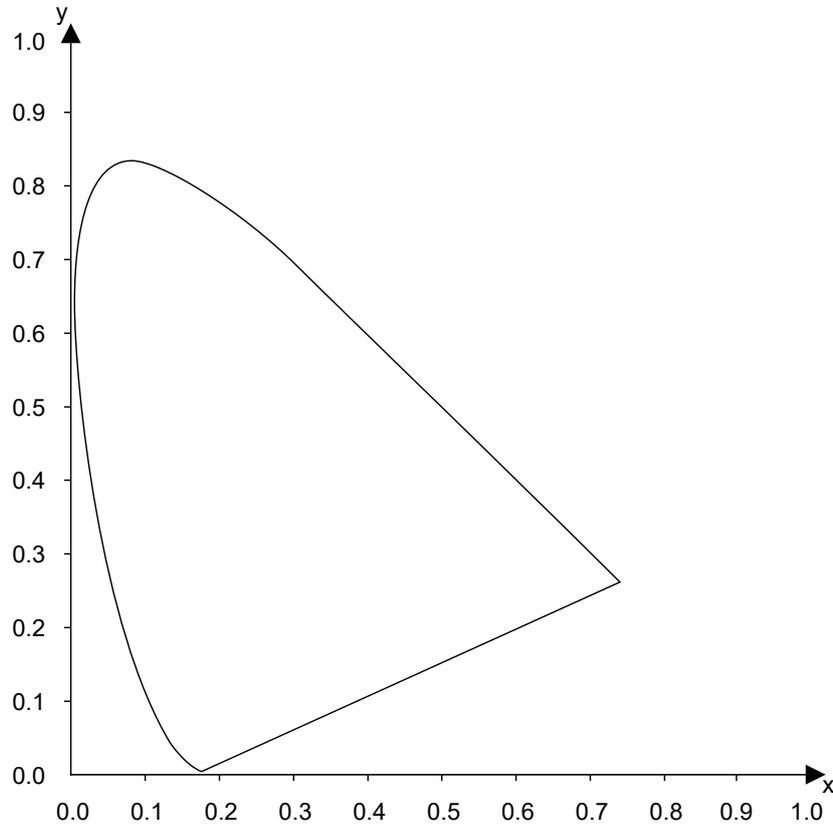
Aufgabe	1	2	3	4	5	6	7	8	9	10	Gesamt
Erreichte Punkte											
Erreichbare Punkte	16	12	31	16	17	20	18	12	23	15	180

Note



Aufgabe 1: Farben (16 Punkte)

In dieser Abbildung ist das CIE Chromatizitätsdiagramm dargestellt, auf das sich die ersten beiden Teilaufgaben beziehen:



a) Zeichnen Sie die Kurve, auf der sich die Spektralfarben befinden, deutlich in das Diagramm ein. Beschriften Sie die Kurve mit *a*. (3 Punkte)



b) Zeichnen Sie einen möglichen, realistischen RGB-Gerätegamut in das Diagramm ein und beschriften Sie dessen Primärfarben mit R für rot, G für grün und B für blau. (3 Punkte)

Name: _____

Matrikelnummer: _____

- c) Bei der Einstellung der Gamma-Korrektur für Monitore verwendet man oft ein Muster das zur Hälfte aus je komplett schwarzen und komplett weißen Zeilen besteht und vergleicht es mit einem Grauwertfeld (siehe Abbildung unten). Wenn bei einem Grauwert $g \in [0, 1]$ die Helligkeitseindrücke des Feldes und des Musters identisch sind, welchen Gamma-Wert kann man dann für den Monitor annehmen? **(6 Punkte)**



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

Aussage	Wahr	Falsch
Bei der subtraktiven Farbmischung werden Farben durch Multiplikation ihrer Spektren gemischt.	<input type="checkbox"/>	<input type="checkbox"/>
Die subjektiv empfundene Stärke eines Sinneseindrucks ist proportional zur Intensität des physikalischen Reizes.	<input type="checkbox"/>	<input type="checkbox"/>
Zur Beschreibung des Farbeindrucks von additiven Mischfarben sind nur die Tristimuluswerte und der Farbraum der Ausgangsfarben erforderlich.	<input type="checkbox"/>	<input type="checkbox"/>
Von einem Tristimuluswert kann eindeutig auf das zugrundeliegende Spektrum geschlossen werden.	<input type="checkbox"/>	<input type="checkbox"/>

Aufgabe 2: Phong-Beleuchtungsmodell (12 Punkte)

Das Phong-Beleuchtungsmodell ist, wie in der Vorlesung, gegeben durch

$$I = k_a \cdot I_L + k_d \cdot I_L \cdot (\mathbf{N} \cdot \mathbf{L}) + k_s \cdot I_L \cdot (\mathbf{R}_L \cdot \mathbf{V})^n.$$

Sie können sämtliche Vektoren als normiert annehmen.

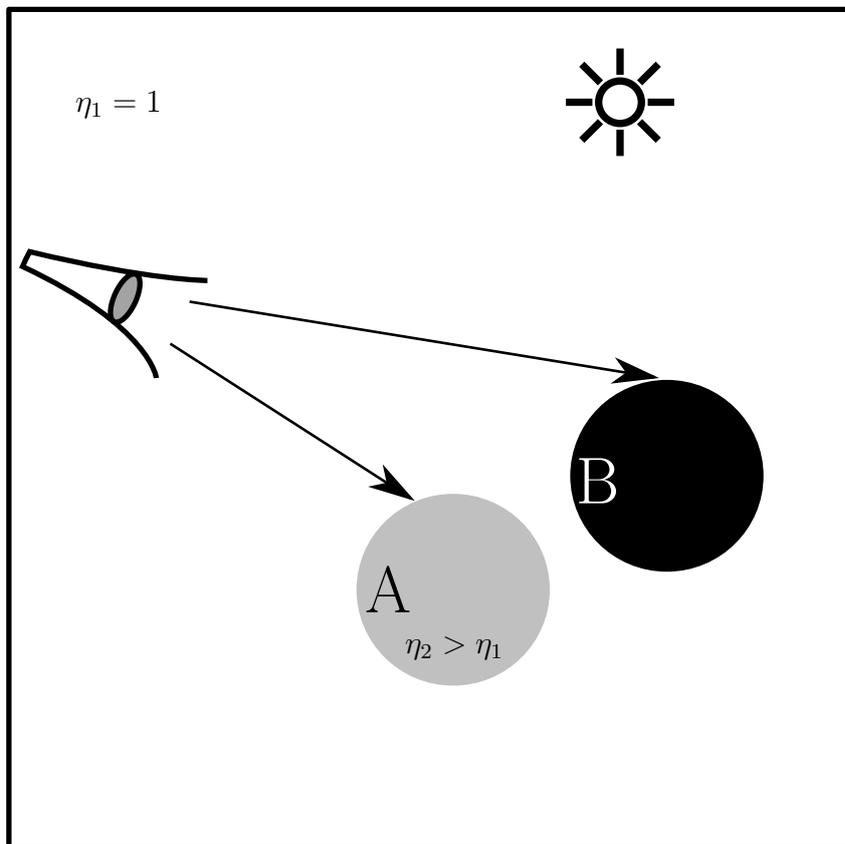
a) Werden Glanzlichter beliebig hell, wenn der Phong-Exponent gegen unendlich geht?
(3 Punkte)

b) Ist es möglich spekulare Glanzlichter zu sehen, wenn sich die Lichtquelle hinter der Kamera befindet ($\mathbf{V} = \mathbf{L}$)? Geben Sie ein Beispiel an, oder begründen Sie Ihre Antwort!
(3 Punkte)

c) Wie kann das Phong-Beleuchtungsmodell effizient mit der Beleuchtung aus Environment Maps verwendet werden? Was muss hierzu vorberechnet werden und wie wird auf diese Daten zugegriffen? **(6 Punkte)**

Aufgabe 3: Whitted-style Raytracing (31 Punkte)

- a) Die Abbildung zeigt eine Szene, die aus einer rein transmittierenden Kugel **A** mit $k_t > 0$, einer rein spiegelnden Kugel **B** mit $k_r > 0$ und einer Punktlichtquelle in einem diffusen Raum besteht. Für beide Kugeln gilt $k_a = k_d = k_s = 0$. Zeichnen Sie in die Abbildung ein, wie ein Whitted-style Raytracer die beiden Sichtstrahlen weiterverfolgt. Dabei sollen nur die Sekundärstrahlen erzeugt werden, die notwendig sind! Kennzeichnen Sie die Pfadsegmente als Primärstrahlen (**P**), Schattenstrahlen (**S**), Reflexionsstrahlen (**R**) oder Transmissionsstrahlen (**T**). (6 Punkte)



- b) Zur Implementierung eines Whitted-style Raytracers stehen Ihnen die im folgenden Programmcode definierten Strukturen und Funktionen zur Verfügung. Der Raytracer soll dabei keine unnötigen Berechnungen ausführen. Die Rekursion soll abgebrochen werden, sobald der Beitrag eines Sekundärstrahls kleiner als ϵ ist. Transmission wird nicht behandelt und es gibt nur eine Lichtquelle!



Vervollständigen Sie den Algorithmus auf der nächsten Seite! **(25 Punkte)**

```
struct Mat      // Reflexionskoeffizienten
{
    vec3 k_a; // ambient
    vec3 k_d; // diffus
    vec3 k_s; // spekulär
    float k_r; // perfekte Spiegelung
};

struct Isect   // Speichert Attribute an einem Schnittpunkt
{
    Mat m; // Material
    vec3 P; // Position
    vec3 N; // normierte Oberflächennormale
    bool found; // True falls Schnittpunkt zwischen 'start' und
                // 'end' gefunden
};

struct Light   // Punktlichtquelle
{
    vec3 P; // Position
    vec3 I; // Intensität
};

Light l; // Punktlichtquelle

// Lokale Beleuchtungsberechnung
vec3 shade(Isect i, Ray incoming, Light l);

// Erzeugt ausgehend von Punkt P einen an der
// Normale N reflektierten Strahl
Ray reflect(Ray incoming, vec3 N, vec3 P);

// Erzeugt Strahl von 'start' zu 'end'
Ray ray(vec3 start, vec3 end);

// schneidet Szenengeometrie mit Strahlsegment
// zwischen 'start' und 'end'
Isect intersect(Ray ray);
```

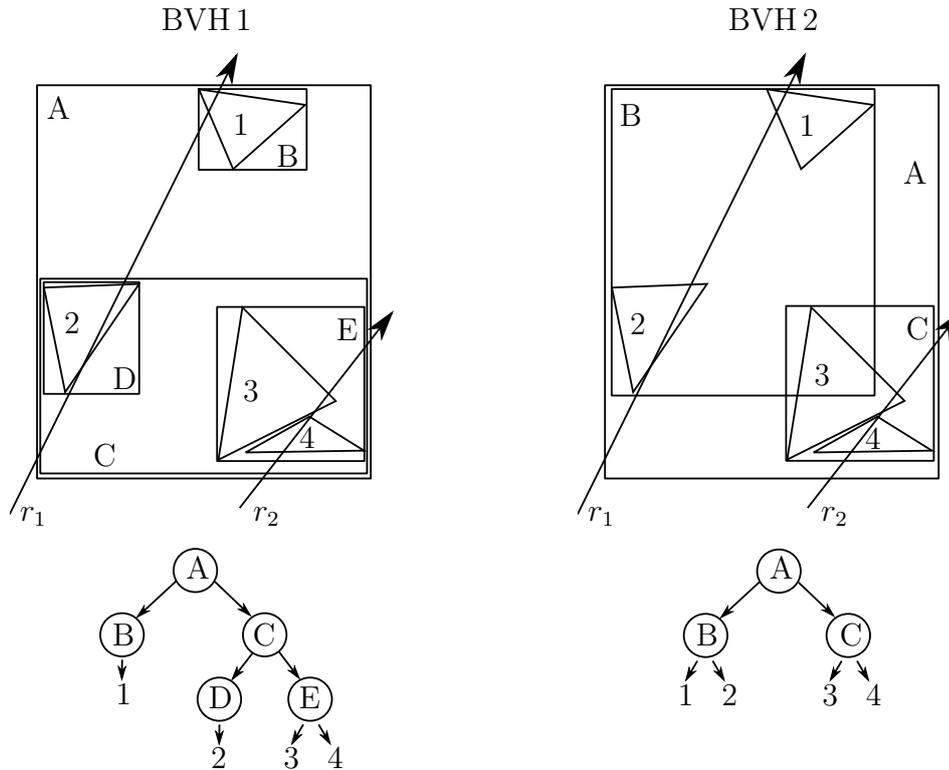
Name: _____

Matrikelnummer: _____

```
vec3 trace(Ray ray, float w = 1.0f)
{
    vec3 color = vec3(0);
    Isect i = intersect(ray);
```

```
    return color;
}
```

Aufgabe 4: Räumliche Datenstrukturen (16 Punkte)



In obiger Abbildung sind zwei unterschiedliche Hüllkörperhierarchien (BVH) für dieselbe Szene gegeben. Die Unterteilungsebene wurde jeweils orthogonal zur Achse mit der größten Ausdehnung gewählt. Der Aufbau wurde abgebrochen, sobald sich maximal zwei Dreiecke in einem Knoten befinden. Die zu den Knoten gehörigen achsenparallelen Hüllquader (AABB) sind in der Abbildung mit Buchstaben gekennzeichnet.

a) Ordnen Sie zu, welche Aufteilung der Szene bei der Konstruktion der BVH verwendet wurde: (3 Punkte)

Spatial median	
Object median	

Name: _____

Matrikelnummer: _____

- b) Die Hüllkörperhierarchie wird für die beiden Strahlen r_1 und r_2 traversiert bis der nahste Schnittpunkt gefunden wurde. Geben Sie für jeden Strahl an, welche Hüllkörper *traversiert* werden, d.h. in den Knoten des Baums abgestiegen wird, und welche Dreiecke auf Schnitt getestet werden, wenn BVH 1 bzw. BVH 2 verwendet wird.

Geben Sie Ihre Lösung als Folge von Buchstaben (für Hüllkörper) und Zahlen (für Dreiecke) in Reihenfolge der Traversierung an! Die Lösung für BVH 1 und r_1 ist bereits vorgegeben. **(9 Punkte)**



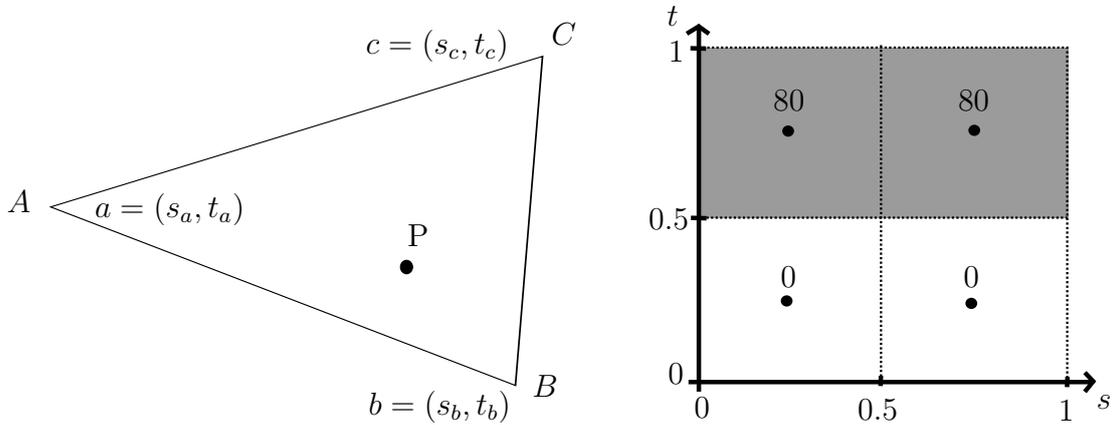
Strahl	BVH 1	BVH 2
r_1	ACD2	
r_2		

- c) Nennen Sie je einen Vorteil der beiden Hüllkörper-Hierarchien BVH 1 und BVH 2 *in dieser Szene*. **(4 Punkte)**



Aufgabe 5: Texturen (17 Punkte)

Gegeben ist ein texturiertes Dreieck mit den Eckpunkten A, B, C . Den Eckpunkten sind die dargestellten Texturkoordinaten zugewiesen. Weiterhin ist auf der rechten Seite die zu verwendende 2×2 Grauwerttextur abgebildet.



- a) Für den Punkt P seien die baryzentrischen Koordinaten λ_A und λ_B bezüglich A und B gegeben. Geben Sie eine Formel zur Berechnung der Texturkoordinate (s, t) am Punkt P an. **(3 Punkte)**

- b) Die Texturkoordinate am Punkt P sei nun gegeben durch $(s, t) = (\frac{1}{2}, \frac{5}{8})$. Berechnen Sie die beiden Grauwerte $c_{NN}(P)$ und $c_{BL}(P)$, die sich bei Auslesen der Textur mit *Nearest-Neighbor* und durch *bilineare Interpolation* ergeben. **(4 Punkte)**

Name: _____

Matrikelnummer: _____

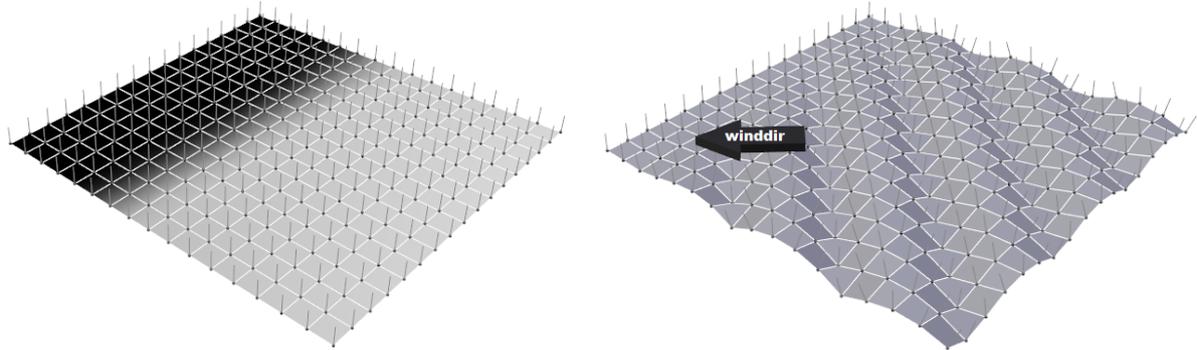
- c) Erklären Sie kurz wie bei Filterung mittels Mip-Mapping eine Mip-Map Stufe n bestimmt wird! **(3 Punkte)**

- d) Was muss beim Berechnen von Mip-Maps von Texturatlantent berücksichtigt werden? **(3 Punkte)**

- e) Nennen Sie eine Parametrisierung für Environment Maps die eine Aufnahme mit einem Fotoapparat ermöglicht und einen Nachteil dieser Parametrisierung! **(4 Punkte)**

Aufgabe 6: OpenGL-Shader (20 Punkte)

Sie erhalten die Aufgabe, eine Windanimation für Stoffe und Flaggen prozedural mittels OpenGL-Shader umzusetzen. Hierzu wird Geometrie mit einem Vertex-Attribut `in_flex`, das die Beweglichkeit des Oberflächenpunkts definiert, gezeichnet.



Links ist die unterteilte Basisgeometrie mit Graustufenvisualisierung des Vertex-Attributs `in_flex` abgebildet. Die durch die Animation veränderte Geometrie ist rechts dargestellt.

- a) Die Verschiebung der Vertices im Wind soll dabei proportional zum Attribut `in_flex` und einer Sinus-Welle entlang der Windrichtung `winddir` erfolgen:

$$\text{amplitude} = \text{in_flex} * \sin(\text{freq} * \text{offset}),$$

wobei `freq` die Periodenlänge bestimmt. `offset` wird durch das Skalarprodukt aus Vertexposition und Windrichtung in Objektkoordinaten berechnet. Die Verschiebung wird entlang der Oberflächennormalen angewandt.



Vervollständigen Sie den Vertex Shader auf der nächsten Seite! **(20 Punkte)**

Name: _____

Matrikelnummer: _____

```
uniform mat4 M;           // Model Matrix (Objekt- zu Weltkoordinaten)
uniform mat4 N;           // Inverse Matrix von M
uniform mat4 VP;          // View-Projection Matrix
uniform vec3 winddir;     // Windrichtung in Weltkoordinaten
uniform float freq;       // Frequenz der vom Wind verursachten Wellen

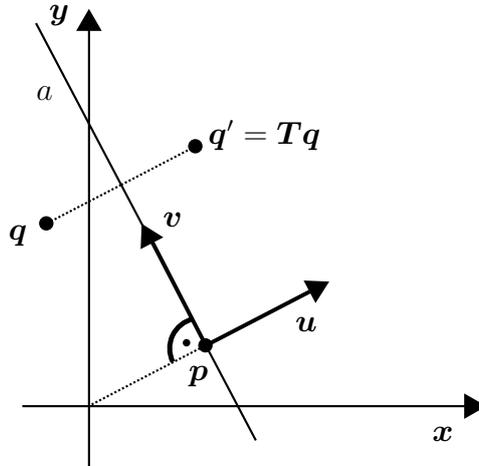
in vec3 in_pos;           // Vertex-Position in Objektkoordinaten
in vec3 in_nrm;           // Vertex-Normale in Objektkoordinaten
in float in_flex;         // Beweglichkeit im Wind (0=fest, 1=voll beweglich)

void main()
{

}

}
```

Aufgabe 7: Transformationen (18 Punkte)



Eine Achse $a \in \mathbb{R}^2$, die den Ursprung nicht schneidet, ist durch den Punkt $\mathbf{p} = (p_x, p_y)$ eindeutig definiert (siehe Abbildung).

a) Bestimmen Sie die Hilfsvektoren \mathbf{u}, \mathbf{v} in Abhängigkeit von \mathbf{p} ! (6 Punkte)

b) Leiten Sie die homogene Transformationsmatrix $\mathbf{T} \in \mathbb{R}^{3 \times 3}$ her, die die Spiegelung eines Punktes $\mathbf{q} \in \mathcal{P}(\mathbb{R}^3)$ an der Achse a beschreibt! Sie müssen die Matrix nicht vereinfachen. (12 Punkte)

Name: _____

Matrikelnummer: _____

Aufgabe 8: OpenGL-Blending (12 Punkte)



- a) Eine Szene mit opaken und semitransparenten Dreiecken soll mit Alpha Blending gezeichnet werden. Dabei wird dem Alpha-Wert eines Fragments eine Opazität zugewiesen, die bei semitransparenten Dreiecken kleiner 1 ist. Weiterhin werden zuerst die opaken und dann die semitransparenten Dreiecke gezeichnet. Nehmen Sie den initialen OpenGL-Zustand als uninitialisiert an. Beantworten Sie die folgenden Fragen indem sie alle richtigen Aussagen ankreuzen. (12 Punkte)



Die opaken Dreiecke werden sortiert	
Von vorne nach hinten	<input type="checkbox"/>
Von hinten nach vorne	<input type="checkbox"/>
Sortieren ist nicht notwendig	<input type="checkbox"/>

Folgende OpenGL Befehle werden vor dem Zeichnen der opaken Dreiecke ausgeführt	
<code>glDepthMask (GL_FALSE) ;</code>	<input type="checkbox"/>
<code>glDepthMask (GL_TRUE) ;</code>	<input type="checkbox"/>
<code>glEnable (GL_DEPTH_TEST) ;</code>	<input type="checkbox"/>
<code>glEnable (GL_BLEND) ;</code>	<input type="checkbox"/>

Die semitransparenten Dreiecke werden sortiert	
Von vorne nach hinten	<input type="checkbox"/>
Von hinten nach vorne	<input type="checkbox"/>
Sortieren ist nicht notwendig	<input type="checkbox"/>

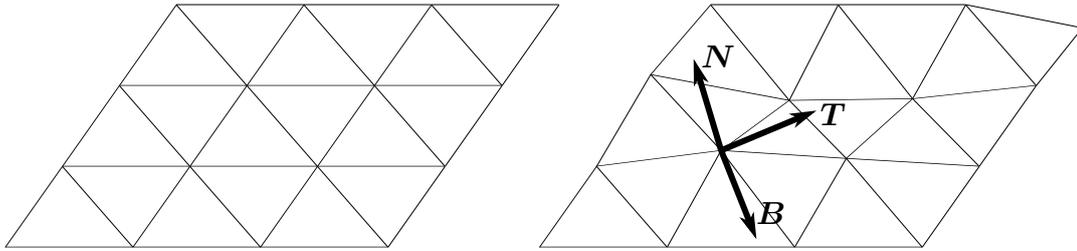
Folgende OpenGL Befehle werden vor dem Zeichnen der semitransparenten Dreiecke ausgeführt	
<code>glDepthMask (GL_FALSE) ;</code>	<input type="checkbox"/>
<code>glDepthMask (GL_TRUE) ;</code>	<input type="checkbox"/>
<code>glDisable (GL_DEPTH_TEST) ;</code>	<input type="checkbox"/>
<code>glEnable (GL_BLEND) ;</code>	<input type="checkbox"/>
<code>glClear (GL_DEPTH_BUFFER_BIT) ;</code>	<input type="checkbox"/>

Zum Verwenden von Alpha-Blending wird folgendes ausgeführt	
<code>glBlendEquation (GL_FUNC_ADD) ;</code>	<input type="checkbox"/>
<code>glBlendFunc (GL_ONE, GL_ONE) ;</code>	<input type="checkbox"/>
<code>glBlendFunc (GL_SRC_ALPHA, GL_DST_ALPHA) ;</code>	<input type="checkbox"/>
<code>glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA) ;</code>	<input type="checkbox"/>

Aufgabe 9: OpenGL-Pipeline (23 Punkte)

Die Darstellung einer Landschaft soll in der programmierbaren OpenGL-Pipeline umgesetzt werden. Das Höhenprofil der Landschaft wird hierzu in Form einer 2D-Textur (Höhenkarte) gespeichert.

- a) Für das Rendering des Terrains wird ein ausreichend vortesselliertes Dreiecksnetz als Basisgeometrie (links) und Displacement Mapping (rechts) verwendet:

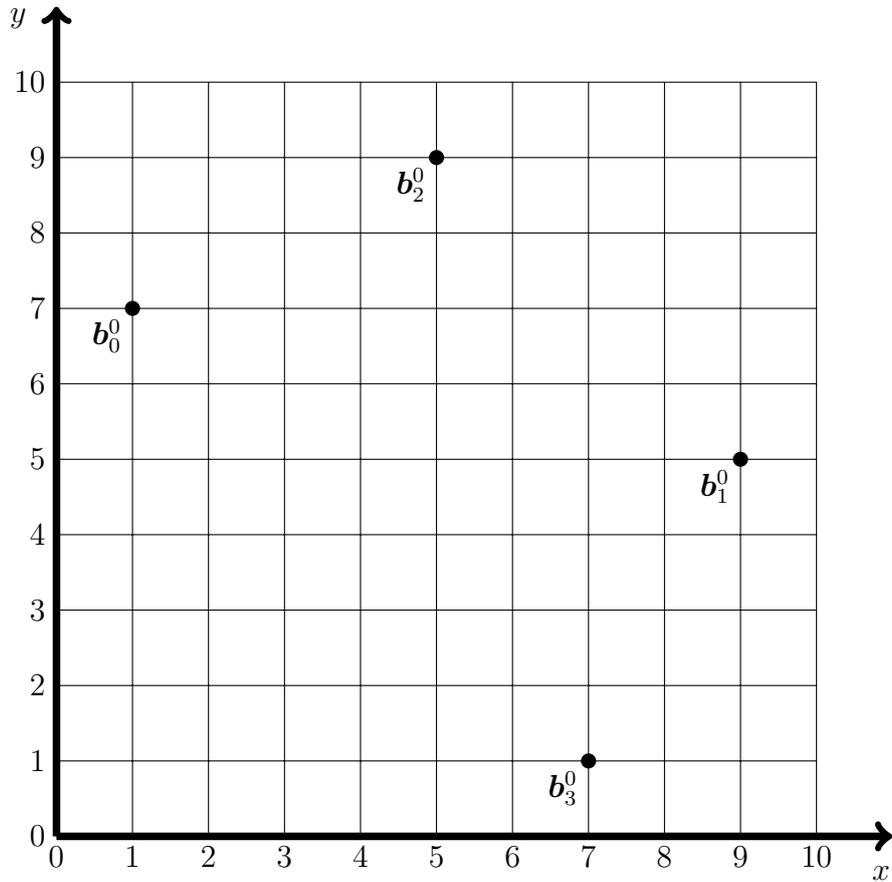


Begründen Sie *kurz* in welchem OpenGL-Shader sich Displacement Mapping in diesem Fall möglichst effizient implementieren lässt? (4 Punkte)



- b) Um den Realismus der Darstellung zu erhöhen soll Nebel, basierend auf der exponentiellen Abschwächung $e^{-\sigma_t d}$, in Abhängigkeit von der Distanz d zwischen Beobachter und Oberflächenpunkt und dem konstanten Parameter σ_t , simuliert werden. In welchem OpenGL-Shader würden Sie die Berechnung durchführen? Begründen Sie Ihre Antwort! (4 Punkte)



Aufgabe 10: Bézierkurven (15 Punkte)

Die vier Kontrollpunkte einer kubischen Bézierkurve seien wie in der Abbildung:

$$\mathbf{b}_0^0 = \begin{pmatrix} 1 \\ 7 \end{pmatrix}, \mathbf{b}_1^0 = \begin{pmatrix} 9 \\ 5 \end{pmatrix}, \mathbf{b}_2^0 = \begin{pmatrix} 5 \\ 9 \end{pmatrix}, \mathbf{b}_3^0 = \begin{pmatrix} 7 \\ 1 \end{pmatrix}.$$

- a) Werten Sie die Kurve für $u = 0.5$ rechnerisch mit dem de Casteljau-Algorithmus aus! **(6 Punkte)**



- b) Skizzieren Sie den Verlauf der Bézierkurve mit dem Wissen über die Eigenschaften von Bézierkurven in obigem Koordinatensystem! **(3 Punkte)**



- c) Gegeben sei eine kubische Bézierkurve mit den Kontrollpunkten \mathbf{b}_i . An diese Kurve soll eine weitere Bézierkurve (Kontrollpunkte \mathbf{c}_i) mit einem C^1 -stetigen Übergang anschließen. Welche Aussage können Sie über deren Kontrollpunkte machen? **(6 Punkte)**