Karlsruher Institut für Technologie

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

Nachklausur Computergrafik — WS 2014/15

10. April 2015

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 21 Seiten (11 Blätter) mit 11 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.

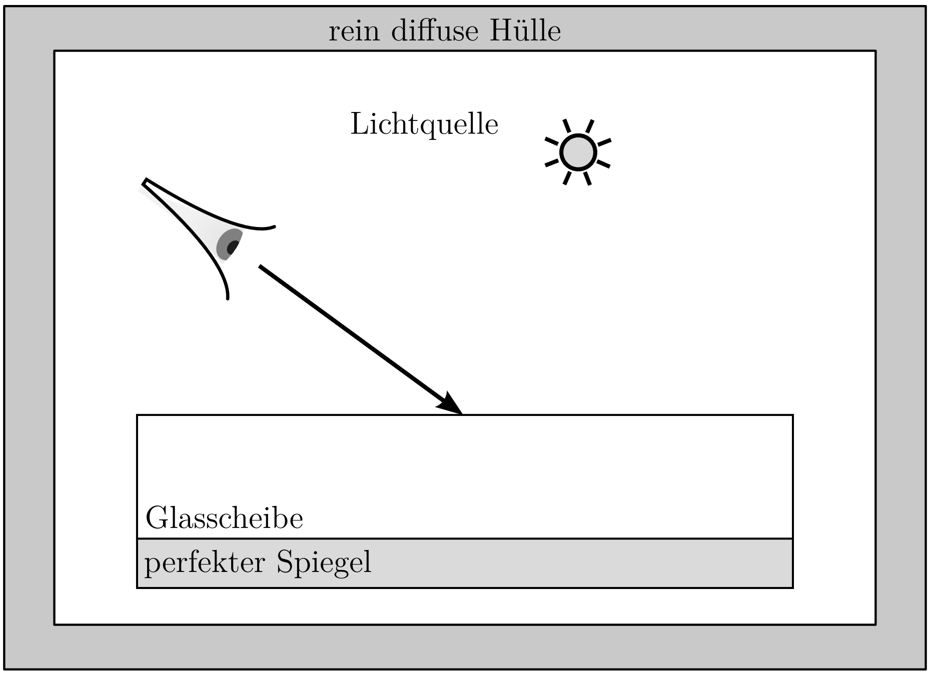
Aufgabe	1	2	3	4	5	6	7	8	9	10	11	**Gesamt**
Erreichte Punkte												
Erreichbare Punkte	8	9	15	6	10	18	7	12	14	9	12	**120**

Note

Aufgabe 1: Raytracing (8 Punkte)

a) In einem Whitted-Style Raytracer wurde der Schnittpunkt des Primärstrahls mit dem nahsten Primitiv bestimmt. Nennen Sie stichpunktartig die Schritte, die nötig sind, um die Farbe des Pixels zu bestimmen! **(3 Punkte)**

b) Gegeben ist eine Szene mit einer Glasscheibe über einem Spiegel und einer Lichtquelle in einem Raum mit rein diffusen Wänden. Der Spiegel *reflektiert* sämtliches Licht, die Glasscheibe *transmittiert* sämtliches Licht. Der Brechungsindex der Glasscheibe ist größer als der des umgebenden Mediums. Zeichnen Sie für den eingezeichneten Primärstrahl alle Sekundärstrahlen ein, die beim Whitted-Style-Raytracing erzeugt werden. Zeichnen Sie keine Schattenstrahlen ein, die der Raytracer vermeiden kann, weil sie unnötig sind! **(5 Punkte)**



Name: ____________________ Matrikelnummer: ________________

Aufgabe 2: Farben und Spektren (9 Punkte)

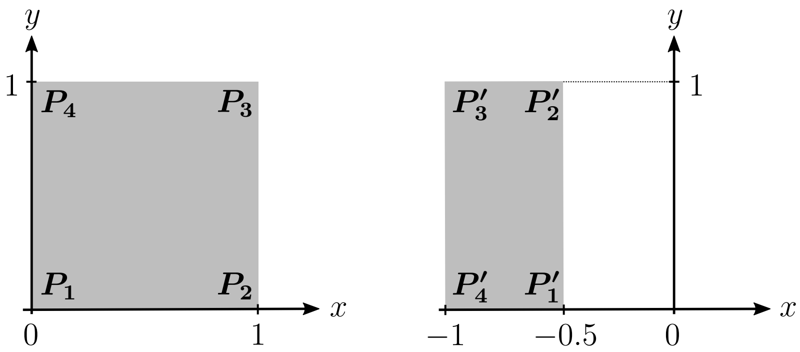
a) Nennen Sie drei Farbräume mit je einem typischen Einsatzgebiet! **(3 Punkte)**

b) Nennen Sie einen Grund für die Durchführung von Gamma-Korrektur! **(3 Punkte)**

c) Was ist Metamerie und warum ist sie für die Farbdarstellung von Monitoren wichtig? **(3 Punkte)**

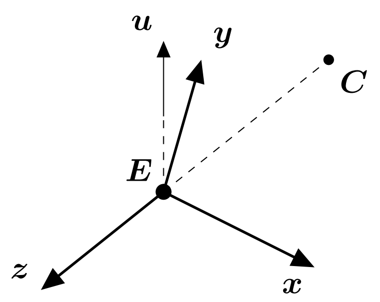
Aufgabe 3: Transformationen (15 Punkte)

a) Gegeben ist ein Rechteck in 2D mit den Eckpunkten $\boldsymbol{P_1}, \ldots, \boldsymbol{P_4}$. Die folgende Abbildung zeigt das Rechteck vor (links) und nach einer affinen Transformation (rechts). Geben Sie eine 2D-homogene Matrix $M \in \mathbb{R}^{3\times 3}$ an, welche die Abbildung beschreibt. **(6 Punkte)**



Name: ______________________ Matrikelnummer: ______________________

b) Eine virtuelle Kamera soll so platziert werden, dass sich ihr Ursprung beim Augpunkt $\boldsymbol{E}$ befindet, sie auf den Punkt $\boldsymbol{C}$ ausgerichtet ist, und dabei der Winkel zwischen der Y-Achse der Kamera und des Up-Vektors $\boldsymbol{u}$ mit $\|\boldsymbol{u}\| = 1$ minimal ist.



Geben Sie an, wie die orthonormalen Achsen $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ des Kamerakoordinatensystems in Weltkoordinaten berechnet werden! **(6 Punkte)**

c) Geben Sie nun die *homogene* Transformationsmatrix V an, die von Welt- in Kamerakoordinaten transformiert! **(3 Punkte)**

Aufgabe 4: Phong-Beleuchtungsmodell (6 Punkte)

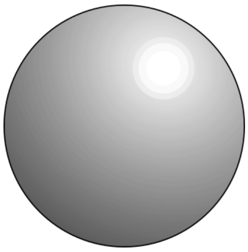
Gegeben sei die Oberflächennormale $\boldsymbol{n}$, die Lichtrichtung $\boldsymbol{l}$, die Blickrichtung $\boldsymbol{v}$ und der Phong-Exponent e. Die Richtungsvektoren $\boldsymbol{l}$ und $\boldsymbol{v}$ zeigen vom Oberflächenpunkt zur Lichtquelle bzw. zum Betrachter.

a) Wie berechnet man die reflektierte Lichtrichtung $\boldsymbol{r_l}$? **(2 Punkte)**

b) Wie lautet die Formel für die spekulare Komponente im Phong-Beleuchtungsmodell? **(2 Punkte)**

Name: ____________________ Matrikelnummer: ____________________

c) Die folgende Abbildung zeigt eine Kugel, die mit Phong-Exponent $e = 10$ schattiert wurde.

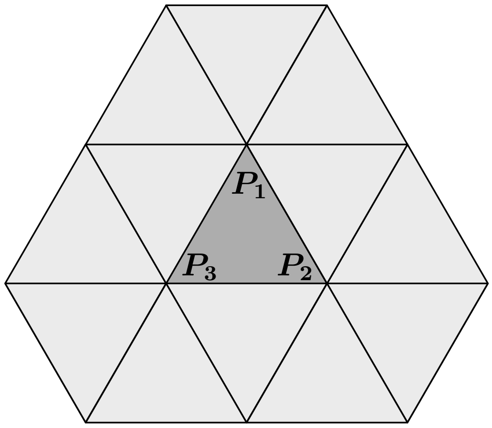


i) Wie verändert sich das Glanzlicht, wenn e größer wird? **(1 Punkt)**

ii) Wie verändert sich das Glanzlicht, wenn die Kugel um eine beliebige Achse rotiert? **(1 Punkt)**

Aufgabe 5: Dreiecke und Schattierung (10 Punkte)

Gegeben sei ein Dreiecksnetz, bei dem nur die Koordinaten der Vertizes bekannt sind.



a) Wie kann die Normale $\boldsymbol{n}$ des mittleren Dreiecks aus den Punkten $\boldsymbol{P_i}$ bestimmt werden? **(3 Punkte)**

b) Um das Dreiecksnetz mit Phong-Shading glatt zu schattieren, benötigen Sie Normalen an den Vertizes. Erläutern Sie stichpunktartig das Verfahren, das Sie in der Vorlesung hierfür kennen gelernt haben!

(3 Punkte)

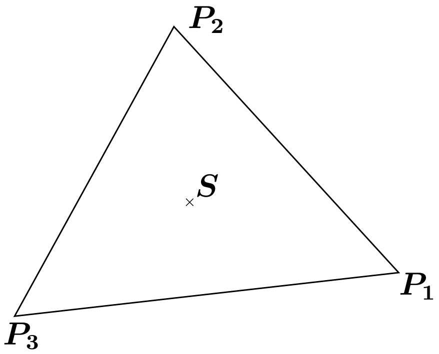
Name: ______________________ Matrikelnummer: ______________________

c) Was ist der Unterschied zwischen Gouraud- und Phong-Shading? **(2 Punkte)**

d) Nennen Sie je *einen* Vorteil und Nachteil von Gouraud-Shading! **(2 Punkte)**

Aufgabe 6: Texturen und Texturfilterung (18 Punkte)

In einem Raytracer sollen texturierte Dreiecke dargestellt werden. Gegeben sei ein Dreieck mit Eckpunkten $\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3} \in \mathbb{R}^3$ und ein Schnittpunkt $\boldsymbol{S} \in \mathbb{R}^3$. Für die Eckpunkte sind die Texturkoordinaten $T_1, T_2, T_3 \in \mathbb{R}^2$ bekannt. Das Dreieck hat den Flächeninhalt $A_\Delta(\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3})$.



a) Geben Sie die Formeln zur Bestimmung der baryzentrischen Koordinaten $\lambda_1, \lambda_2, \lambda_3$ an! **(3 Punkte)**

b) Wie berechnet man die Texturkoordinate T_S für den Schnittpunkt $\boldsymbol{S}$? Geben Sie die Formeln an! **(1 Punkt)**

Name: ______________________ Matrikelnummer: ______________________

c) Die Textur wird nun mit “Nearest Neighbor Filtering” bei T_S ausgelesen. Dabei können Artefakte entstehen. Welche zwei Fälle unterscheidet man? Nennen Sie jeweils ein Beispiel für diese Artefakte! **(3 Punkte)**

d) Welche texturbezogenen Lösungen bietet OpenGL an, um die Artefakte aus Aufgabe c) zu reduzieren? Erklären Sie stichpunktartig, wie sie funktionieren! **(4 Punkte)**

e) Was versteht man unter trilinearer Texturfilterung? **(2 Punkte)**

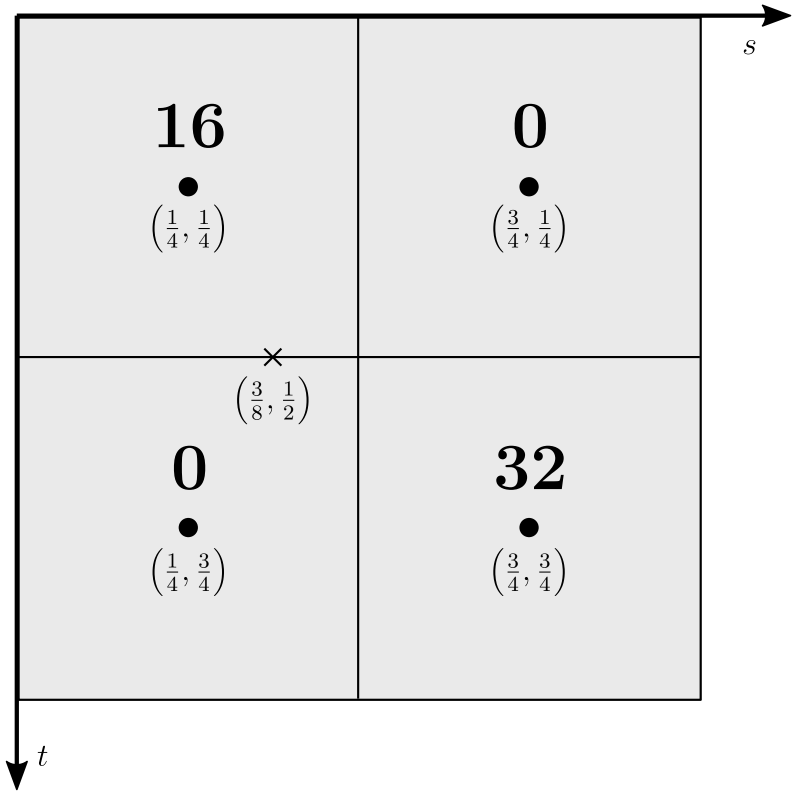
f) In OpenGL sind die folgenden Parameter für die Texturierung gesetzt:

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
```

Berechnen Sie den Wert, der aus der unten abgebildeten Textur an der Stelle

$$(s,t) = \left(\frac{3}{8}, \frac{1}{2}\right)$$

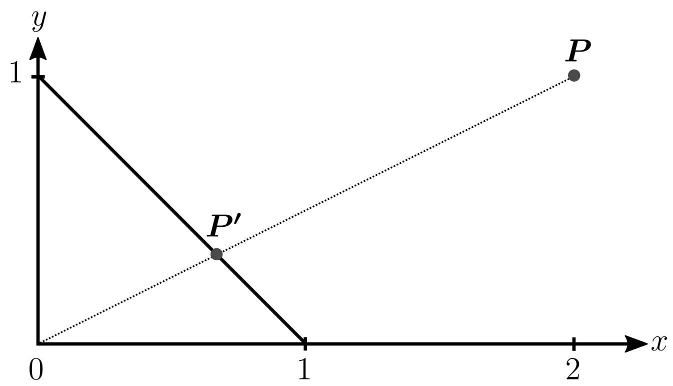
ausgelesen wird! Wie nennt man dieses Berechnungsschema? **(5 Punkte)**



Name: ______________________ Matrikelnummer: ______________________

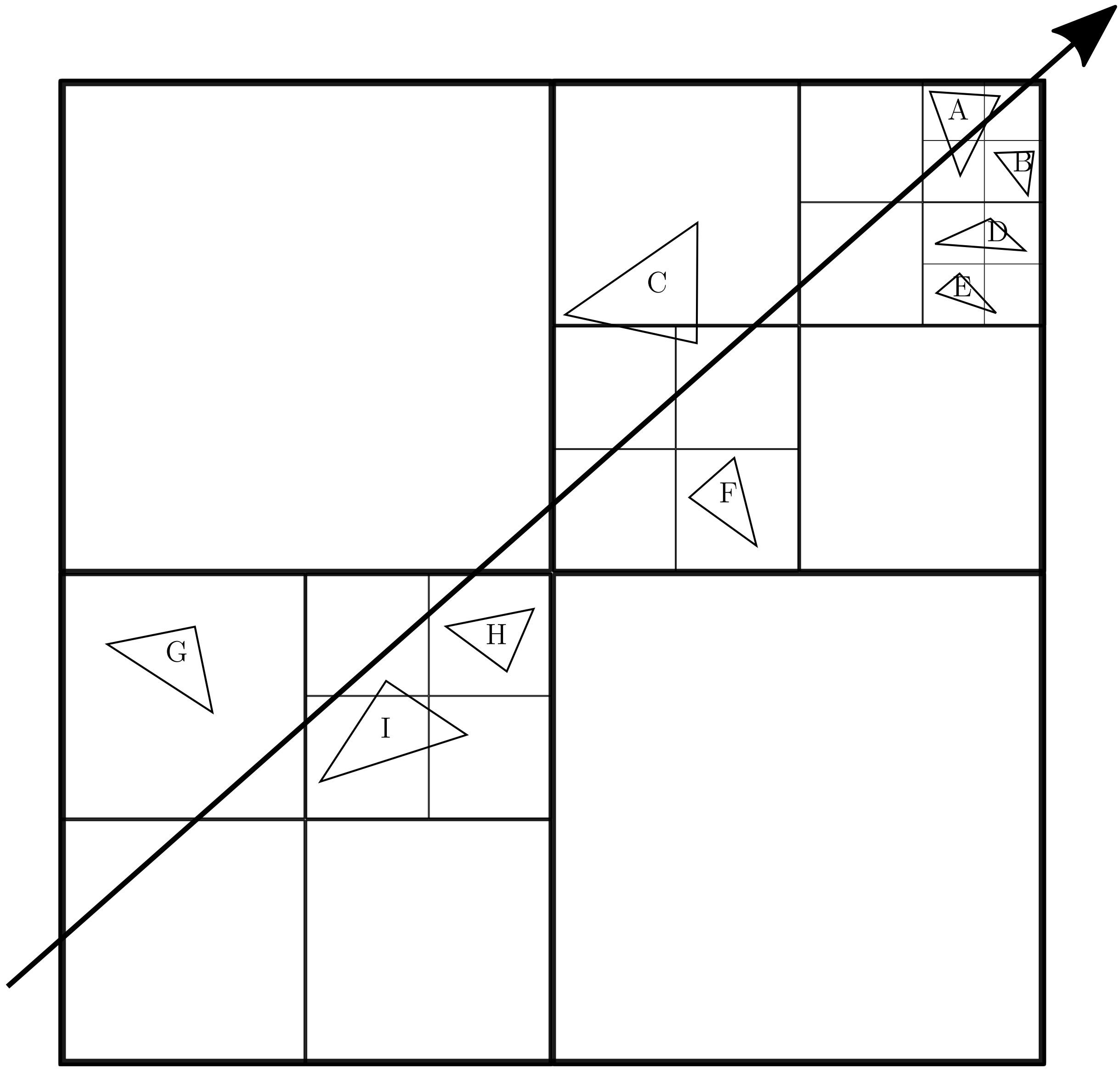
Aufgabe 7: Projektionen (7 Punkte)

a) Die folgende Abbildung zeigt eine Projektion, die jeden Punkt $\boldsymbol{P} \in [0, \infty]^2$ auf den Unterraum $\{\boldsymbol{P} \in [0, \infty]^2 : \boldsymbol{P_x} + \boldsymbol{P_y} = 1\}$ projiziert. Das Projektionszentrum ist dabei der Ursprung $(0, 0)$. Geben Sie eine 2D-homogene Matrix $M \in \mathbb{R}^{3\times 3}$ an, die diese Abbildung beschreibt und transformieren Sie beispielhaft den Punkt $\boldsymbol{P} = (2, 1)$. **(7 Punkte)**



Aufgabe 8: Beschleunigungsstrukturen (12 Punkte)

a) Der abgebildete Quadtree wird für den eingezeichneten Strahl traversiert, um den nahsten Schnittpunkt mit einem Dreieck zu bestimmen. Geben Sie sämtliche Schnitttests zwischen Dreiecken und dem Strahl an, die durchgeführt werden, wenn *kein* Mailboxing verwendet wird! **(4 Punkte)**



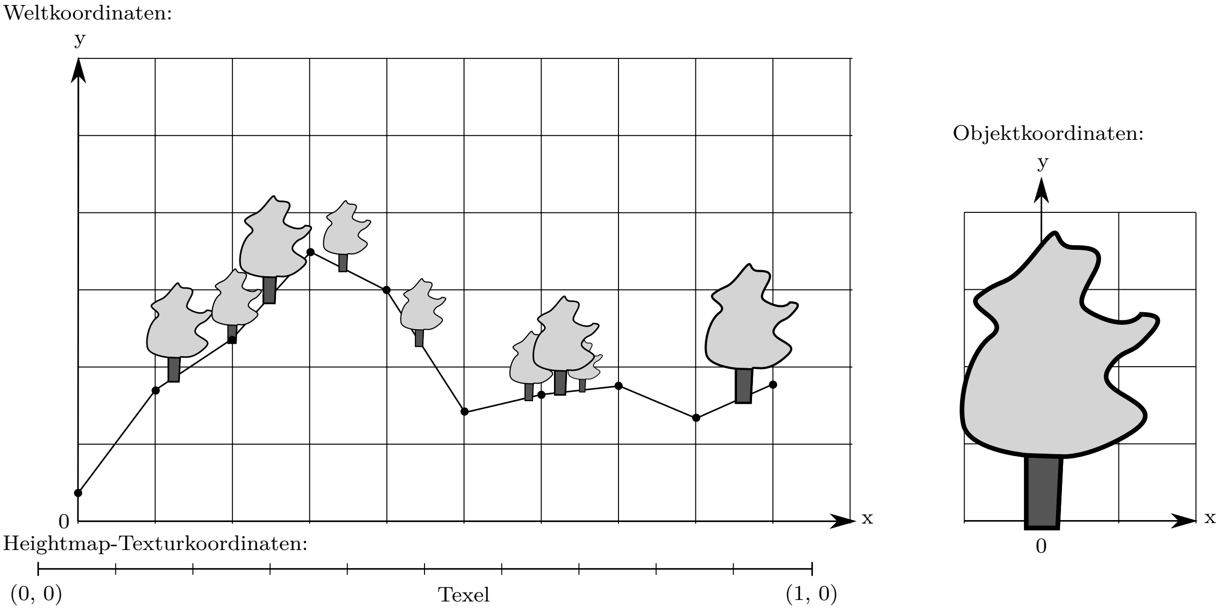
Name: ______________________ Matrikelnummer: ______________________

b) Bewerten Sie die folgenden Aussagen, indem Sie *Wahr* oder *Falsch* ankreuzen. **(8 Punkte)** ☐

Aussage	**Wahr**	**Falsch**
Der Baum einer Hüllkörperhierarchie ist immer balanciert.	☐	☐
Der Speicherbedarf für ein reguläres Gitter ist unabhängig von der Anzahl der Primitive.	☐	☐
Ein kD-Baum hat immer achsenparallele Split-Ebenen.	☐	☐
Ein kD-Baum braucht spezielle Vorkehrungen, um redundante Schnitttests mit demselben Dreieck auszuschliessen.	☐	☐
Ein Verfahren zur Erzeugung eines kD-Baumes erzeugt auch gültige BSP-Bäume.	☐	☐
Reguläre uniforme Gitter leiden nicht unter dem Teapot-in-a-Stadium Problem.	☐	☐
Die Komplexität der Bestimmung eines Schnittpunktes in einem BSP-Baum mit n Primitiven liegt im Optimalfall in $\mathcal{O}(\log n)$.	☐	☐
Das Traversieren einer Hüllkörperhierarchie kann abgebrochen werden sobald ein Schnittpunkt gefunden wurde.	☐	☐

Aufgabe 9: Instancing (GLSL) (14 Punkte)

Die Abbildung unten zeigt ein Höhenfeld im Weltkoordinatensystem. Die Vertizes des Höhenfelds liegen auf einem regelmäßigen Gitter auf den ganzzahligen Koordinaten der xz-Ebene, ihre y-Koordinate entspricht den Höhen aus einer Höhenkarte `heightMap`. Die Zuordnung von Heightmap-Texturkoordinaten zu Weltpositionen erfolgt wie illustriert, je ein Texelzentrum entspricht einem Vertex.



Ihre Aufgabe ist es, Bäume auf dem Höhenfeld mittels Instancing in einem Vertex-Shader zu zeichnen. Die Platzierung und Skalierung jedes Baums ist in der 1D-Textur `treeInstanceData` gespeichert, auf die mit `gl_InstanceID` zugegriffen wird. Die xy-Komponenten der Texel speichern die Weltposition eines Baums in der xz-Ebene. Die z-Komponente enthält einen Skalierungsfaktor für die Geometrie des Baums.

Das Baummodell wurde so modelliert, dass der Ursprung in Objektkoordinaten dem Punkt entspricht, an dem die jeweilige Bauminstanz auf dem Höhenfeld aufsetzen soll (Bild rechts).

Ergänzen Sie den Vertex-Shader so, dass die Bäume an der jeweiligen Position und entsprechend skaliert dargestellt werden!

Name: ______________________ Matrikelnummer: ______________________

Vervollständigen Sie folgenden Vertex-Shader: **(14 Punkte)**

```
uniform mat4 VP; // View-Projection Matrix

uniform sampler2D heightMap; // Höhenkarte (Rot-Komponente speichert Höhe)
uniform vec2 rcpHMS; // Kehrwerte der heightMap-Auflösung in s- und t-Richtung

// (r,g,b) == (x-Position, z-Position, Skalierung)
uniform sampler1D treeInstanceData;

in vec3 POS; // Vertex-Position in Objektkoordinaten
in vec3 NRM; // Vertex-Normale in Objektkoordinaten

out vec3 wpos; // Vertex-Position in Weltkoordinaten
out vec3 wnrm; // Vertex-Normale in Weltkoordinaten

void main(void)
{
   vec3 data = texelFetch(treeInstanceData, gl_InstanceID, 0);











   gl_Position =

}
```

Aufgabe 10: Normal Mapping in Objektkoordinaten (GLSL) (9 Punkte)

Ein Dreiecksnetz wird mithilfe eines OpenGL Fragment Shaders gezeichnet und soll mit einer für diffuse Beleuchtung vorgefilterten Environment Map (gespeichert in der Cube Map `eMap`) beleuchtet werden.

Weiterhin ist eine Normal Map `nMap` gegeben, welche detaillierte Normalen für die Beleuchtungsberechnung in *Objektkoordinaten* enthält.

Die Texturzugriffe erfolgen mit den Funktionen:

`vec4 texture(samplerCube sampler, vec3 c);` bzw.

`vec4 texture(sampler2D sampler, vec2 c);`.

Ergänzen Sie den Fragment Shader wie folgt:

- Die Normale wird aus der Normal Map ausgelesen und mithilfe der gegebenen Matrizen in Weltkoordinaten transformiert.
- Die Ausgabefarbe wird durch diffuse Reflexion mit dem Reflexionskoeffizienten `kd` und der Environment Map bestimmt.

Name: ______________________ Matrikelnummer: ______________________

Vervollständigen Sie den folgenden Fragment Shader: **(9 Punkte)**

```
uniform samplerCube eMap; // Environment Map mit diffuser Beleuchtung
uniform sampler2D   nMap; // Normal Map mit Normalen in Objektkoordinaten

uniform mat3 matO2W; // Transformationsmatrix von Objekt- zu Weltkoordinaten
uniform mat3 matW2O; // Transformationsmatrix von Welt- zu Objektkoordinaten

uniform vec3 kd; // diffuser Reflexionskoeffizient

in vec2 tc;      // Texturkoordinate des Fragments
out vec3 color; // Ausgabefarbe des Fragments

void main(void)
{






}
```

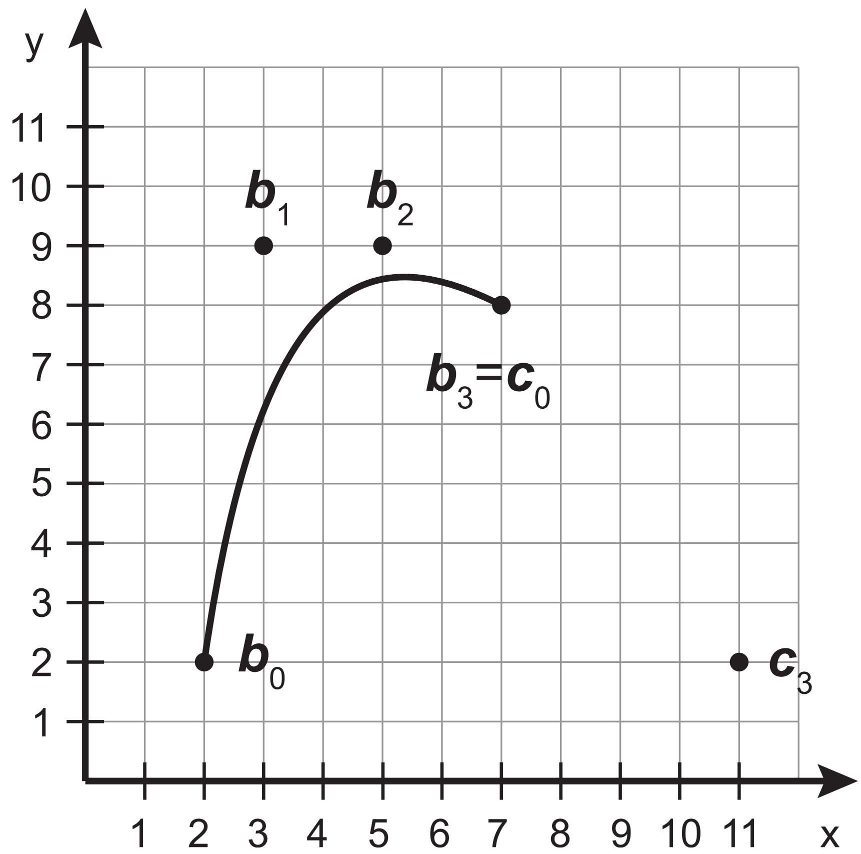
Aufgabe 11: Bézier-Kurven und Bézier-Splines (12 Punkte)

a) Gegeben sei die Bézier-Kurve $\boldsymbol{b}(u) = \sum_{i=0}^{3} \boldsymbol{b}_i B_i^3(u)$ mit den kubischen Bernsteinpolynomen $B_i^3(u)$ und Kontrollpunkten $\boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$.

Eine zweite Bézier-Kurve $\boldsymbol{c}(v) = \sum_{i=0}^{3} \boldsymbol{c}_i B_i^3(v)$ soll C^2-stetig angeschlossen werden. Die Punkte $\boldsymbol{c}_0 = \boldsymbol{b}_3$ und $\boldsymbol{c}_3$ sind bereits vorgegeben.

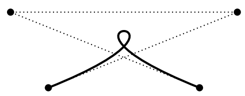
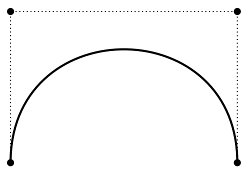
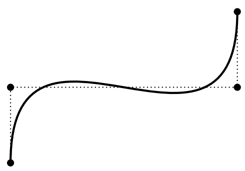
I) Zeichnen Sie die beiden noch fehlenden Punkte $\boldsymbol{c}_1$ und $\boldsymbol{c}_2$ so ein, dass der resultierende Bézier-Spline C^2-stetig im Anschlusspunkt ist. **(5 Punkte)**

II) Skizzieren Sie grob den Verlauf der zweiten Bézier-Kurve $\boldsymbol{c}(v)$! (nicht exakt auswerten oder zeichnen!) **(1 Punkt)**



Name: ____________________ Matrikelnummer: ____________________

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! **(6 Punkte)**

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