Karlsruher Institut für Technologie

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

Nachklausur Computergrafik
SS 2024

7.8.2024

Name	
Matrikelnummer	

Beachten Sie:

- Die Klausur umfasst 25 Seiten (13 Blätter) mit 11 Aufgaben.
- Es sind **nur** Geodreiecke, Lineale und Zirkel als 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	14	11	10	11	14	27	16	6	38	17	16	**180**

Note

Aufgabe 1: Farbe und Wahrnehmung (14 Punkte)

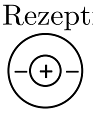
a) Vervollständigen Sie die Sätze, indem Sie den entsprechenden mathematischen Operator angeben! **(2 Punkte)**

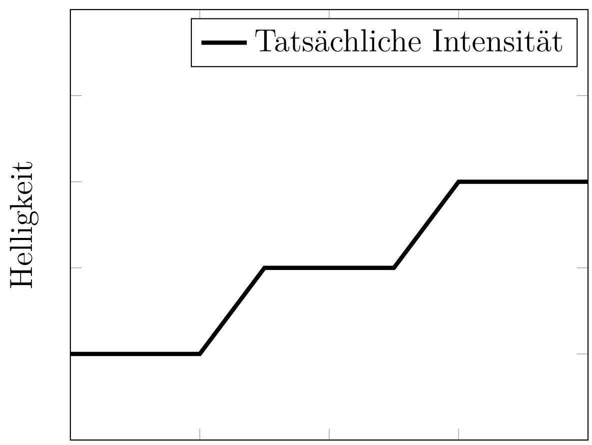
   i) Bei additiver Farbmischung werden die Spektren ________.

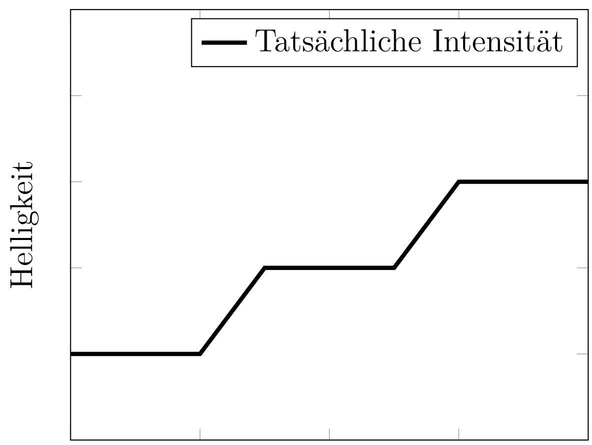
   ii) Bei subtraktiver Farbmischung werden die Spektren ________.

b) Erklären Sie mithilfe der Just Noticeable Difference (JND), warum sich eine potenzfunktionsbasierte Quantisierung besser für die Speicherung von Bildern gegenüber einer linearen Quantisierung eignet! **(6 Punkte)**

c) Zeichnen Sie die wahrgenommene Helligkeit mit rezeptiven Feldern des Typs ON für den angegebenen Intensitätsverlauf in die Grafik ein! Zur Korrektur können Sie die zweite Zeichnung verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnung bewertet werden soll!* **(3 Punkte)**

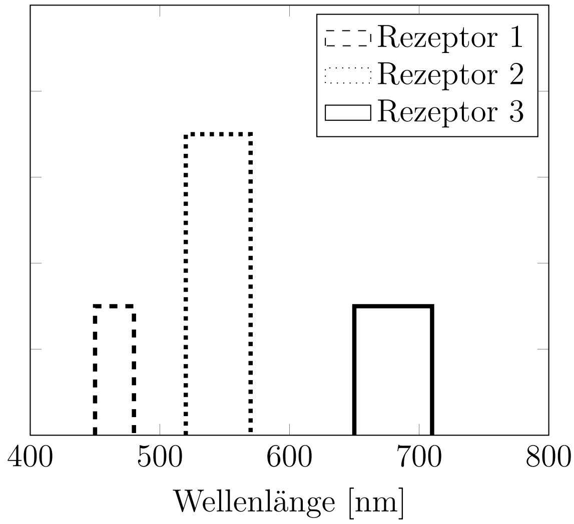
Hinweis: Rezeptives Felder des Typs ON: 



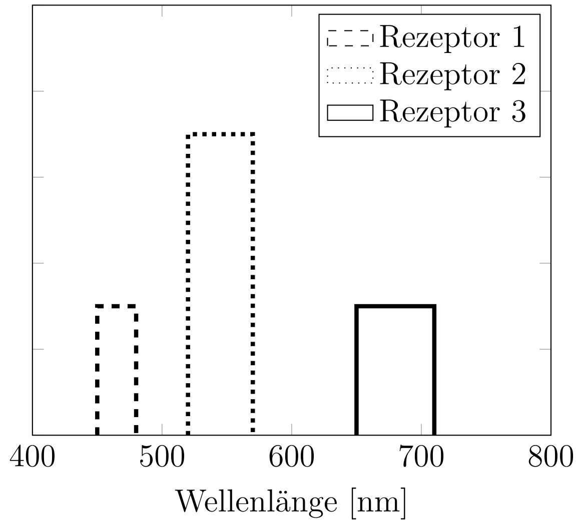


(Ersatzgrafik)

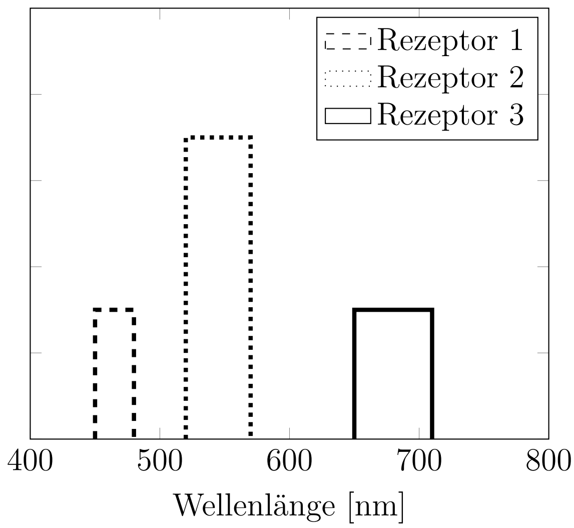
d) Gegeben sei ein imaginärer Beobachter, der drei Rezeptoren mit den in der Grafik gezeigten spektralen Antworten besitzt. Skizzieren Sie zwei verschiedene Spektren, die für den Beobachter den gleichen Farbeindruck auslösen! Zeichnen Sie jeweils ein Spektrum in eine Zeichnung. Zur Korrektur können Sie die unteren Zeichnungen verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnungen bewertet werden sollen!* **(3 Punkte)**



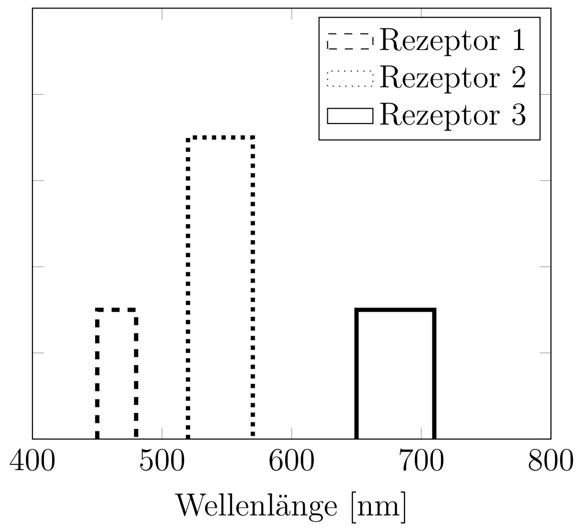
(Spektrum 1)



(Spektrum 2)

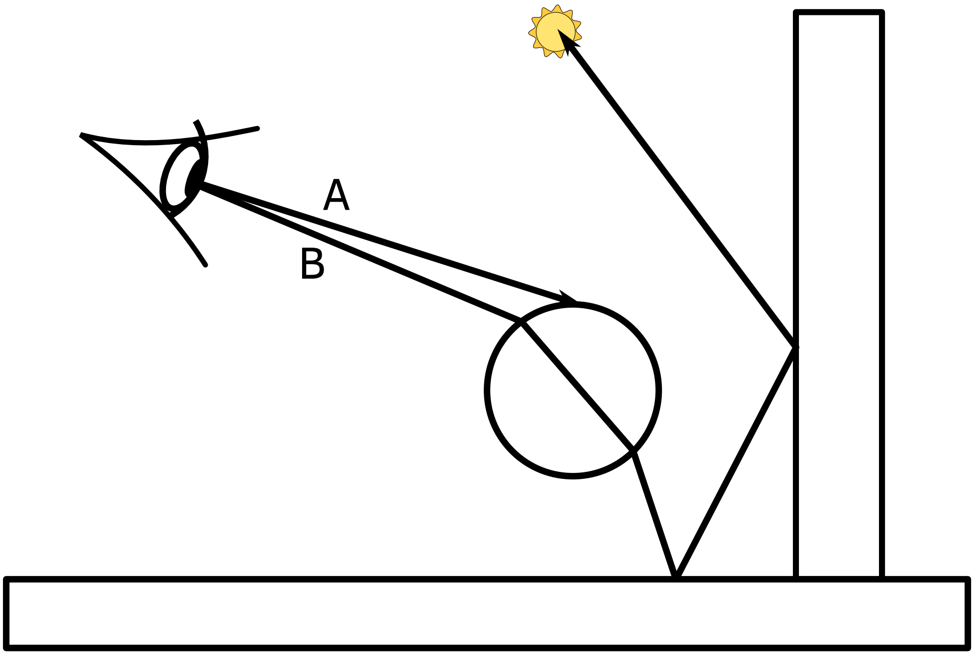


(Ersatzgrafik)



(Ersatzgrafik)

☐ **Aufgabe 2: Whitted-Style Raytracing (11 Punkte)**



Die obige Illustration enthält einen vollständigen Lichttransportpfad B und einen angefangenen Pfad A. Die Kugel sei aus Glas und auf ihrer Oberfläche ist ein Glanzlicht zu sehen ($k_d = 0, k_s = 1, k_r = 1, k_t = 1$, je für diffus, spekular, reflektierend, transmittierend). Wand und Boden sind diffus ($k_d > 0, k_r = k_s = k_t = 0$). Das umgebende Medium hat die Brechzahl $\eta = 1$. Die Szene wird von einer Punktlichtquelle von oben beleuchtet und soll mittels Whitted-style Raytracing dargestellt werden.

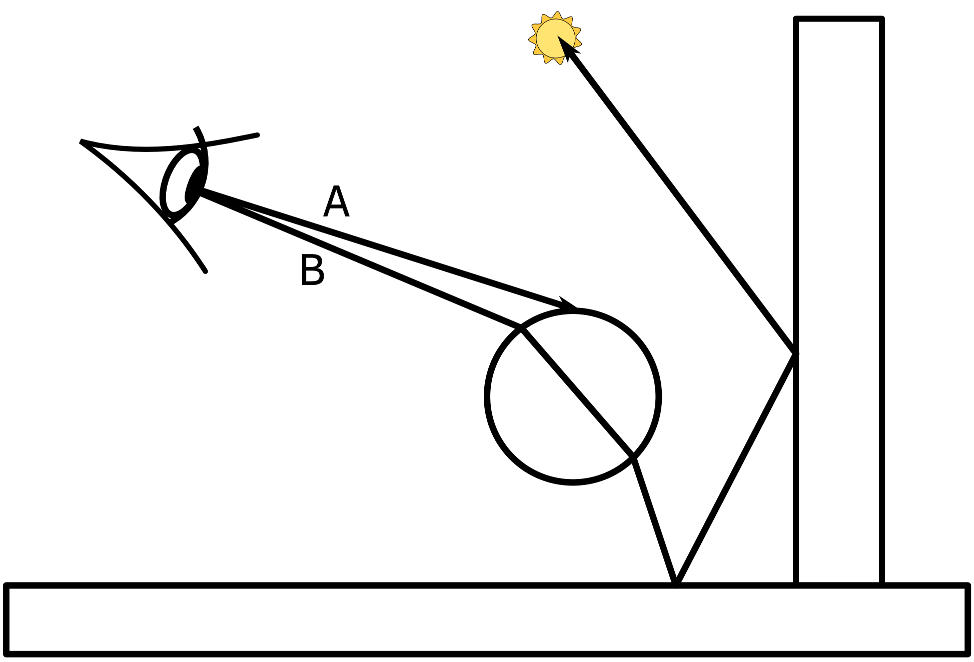
☐ a) Vervollständigen Sie den Pfad A in der Zeichnung, indem Sie maximal zwei Rekursionsschritte durchführen! Zeichnen Sie dabei die Schattenstrahlen als gestrichelte Linien! Zur Korrektur können Sie die zweite Zeichnung am Ende der Aufgabe verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnung bewertet werden soll!* **(5 Punkte)**

☐ b) Kann der Pfad B mit Whitted-style Raytracing erzeugt werden? Falls ja, mit welchen Schritten? Falls nein, warum nicht? **(3 Punkte)**

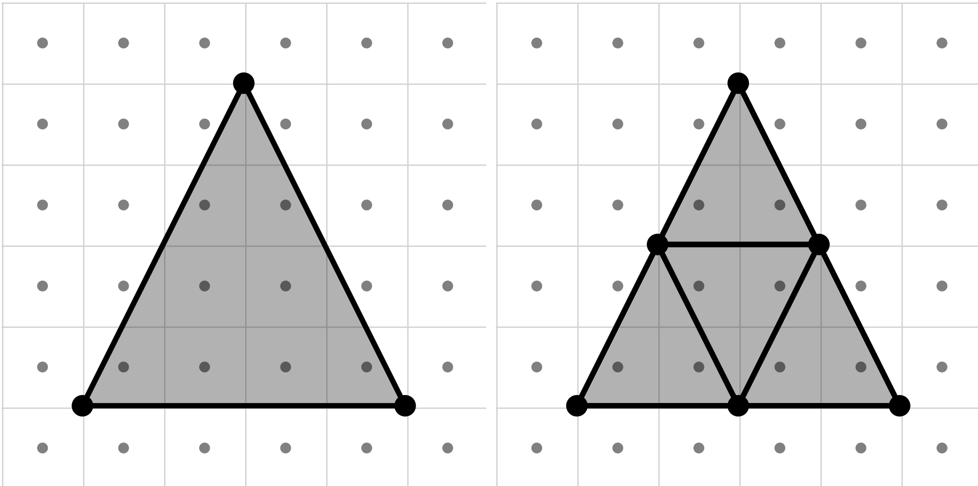
Matrikelnummer: ____________________

c) Benennen Sie ein prominentes visuelles Merkmal der Lichtausbreitung in natürlichen Szenen, welches nicht mit Whitted-style Raytracing abgebildet werden kann, sondern nur mit Distributed Raytracing oder Pathtracing! Begründen Sie! **(3 Punkte)**

Ersatzskizze:



Aufgabe 3: Shading und Beleuchtungsmodell (10 Punkte)



(a) ein Dreieck

(b) ein unterteiltes Dreieck

Abbildung 1

a) Welche Komponente des Phong-Beleuchtungsmodells ist unabhängig von der Lichtrichtung und was ist ihr Zweck beim Schattieren? **(2 Punkte)**

b) Abbildung 1(a) zeigt ein rasterisiertes Dreieck, das einen Teil eines 4×4 Pixel-großen Bereichs bedeckt. In Abbildung 1(b) wurde das Dreieck vor dem Rasterisieren unterteilt. Wie oft wird das Beleuchtungsmodell bei Gouraud beziehungsweise Phong Shading jeweils ausgewertet? Begründen Sie kurz! **(4 Punkte)**

	Abbildung 1(a)	Abbildung 1(b)
Gouraud		
Phong		

Begründung:

c) Eine flache, rein-diffuse Oberfläche wird durch eine direktionale Lichtquelle beleuchtet. Führen Gouraud und Phong Shading zu identischen Resultaten? Begründen Sie!
(4 Punkte)

Aufgabe 4: Räumliche Datenstrukturen (11 Punkte)

a) Ein Hüllkörper ist nur als solcher geeignet, wenn er die Geometrie vollständig umschließt. Erklären Sie, welches Problem beim Raytracing auftreten kann, falls ein Hüllkörper die Geometrie nur zum Teil umschließt! **(3 Punkte)**

b) Gegeben sei die Hüllkörper-Hierarchie (Bounding Volume Hierarchy, BVH) aus Abbildung 2. Bestimmen Sie für die beiden eingezeichneten Strahlen die Reihenfolge der bei der Traversierung besuchten Knoten und auf Schnitt getesteten Dreiecke! Ziel der Traversierung ist das schnelle Finden des nahsten Schnittpunkts. **(8 Punkte)**

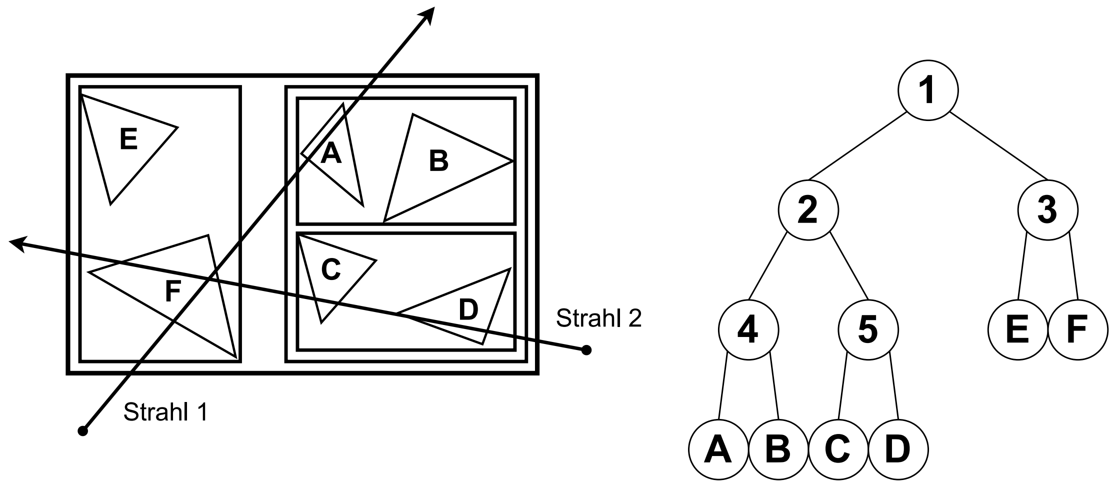


Abbildung 2: Links: Sechs Dreiecke mit Hüllkörperhierarchie. Rechts: Baum-Repräsentation der BVH.

Strahl 1: { }

Strahl 2: { }

Matrikelnummer: ______________________

Aufgabe 5: Transformationen (14 Punkte)

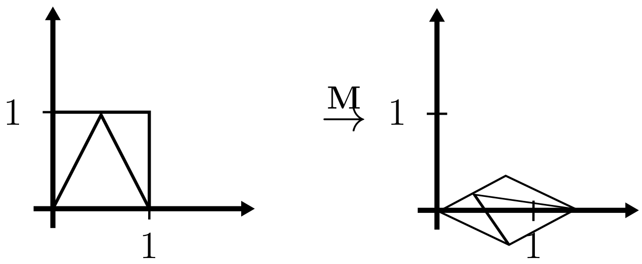
Gegeben sind die affinen Transformationen **A** bis **F**:

$$\mathbf{A} = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

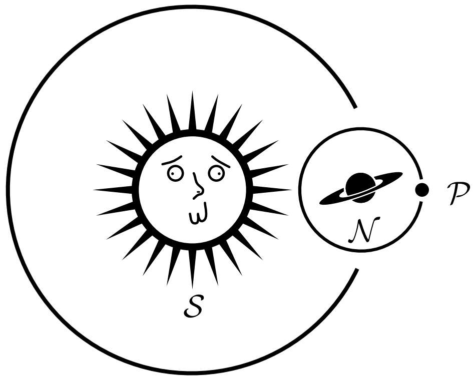
$$\mathbf{E} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{F} = \begin{pmatrix} \pi & -\pi & 0 \\ \pi & \pi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

a) Geben Sie an, wie die affine Transformation **M** aus den Matrizen **A** bis **F** gebildet werden kann, die das Objekt wie in der folgenden Abbildung transformiert! **(6 Punkte)**



$\mathbf{M} = \boxed{} \times \boxed{} \times \boxed{}$

b) Für ein Sci-Fi-Computerspiel sollen die Bewegungen der Himmelskörper *Sol* $\mathcal{S}$, *Neptun* $\mathcal{N}$ und *Proteus* $\mathcal{P}$ modelliert werden. $\mathcal{P}$ liegt in einer Umlaufbahn um $\mathcal{N}$, $\mathcal{N}$ selbst liegt in einer Umlaufbahn um $\mathcal{S}$:



i) Geben Sie die Transformationsmatrix $\mathbf{P}$ für $\mathcal{P}$ in Abhängigkeit von

- der Translation $\mathbf{T}_{\mathcal{P}}$ von $\mathcal{P}$ in die Umlaufbahn von $\mathcal{N}$ und der Rotation $\mathbf{R}_{\mathcal{P}}$ von $\mathcal{P}$ um $\mathcal{N}$ und
- der Translation $\mathbf{T}_{\mathcal{N}}$ von $\mathcal{N}$ in die Umlaufbahn von $\mathcal{S}$ und der Rotation $\mathbf{R}_{\mathcal{N}}$ von $\mathcal{N}$ um $\mathcal{S}$

an, so dass sich $\mathcal{P}$ an der korrekten Position relativ zu $\mathcal{S}$ befindet! **(6 Punkte)**

$$\mathbf{P} = \square \times \square \times \square \times \square$$

ii) Durch ein schwarzes Loch wird das gesamte *Sol*-System einer seltsamen Transformation $\mathbf{L}$ unterzogen:

$$\mathbf{L} = \begin{pmatrix} \pi & 0 & 0 \\ 0 & \frac{1+\sqrt{5}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Zeigen die Normalen des *Sol*-Systems nach Transformation mit $\mathbf{L}$ und Normalisierung in diesem Fall in die richtige Richtung? Wie können Normalen im Allgemeinen richtig transformiert werden? **(2 Punkte)**

Matrikelnummer: ____________________

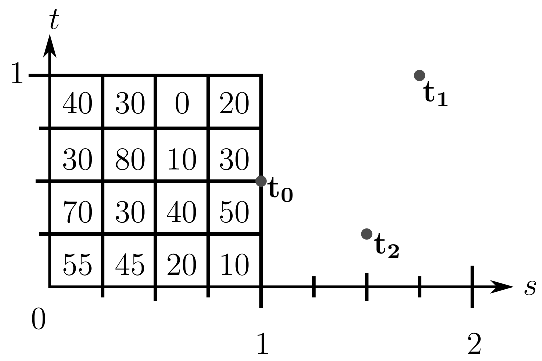
Aufgabe 6: Texturen und Baryzentrische Koordinaten (27 Punkte)

a) Nennen Sie drei mögliche Parametrisierungen von Environment Maps, die Sie aus der Vorlesung kennen! Geben Sie jeweils an, wie viele 2D-Texturen zur Speicherung benötigt werden und ob die Verzerrungen der Parametrisierung gering oder hoch sind! **(6 Punkte)**

Parametrisierung	Anzahl Texturen	Verzerrung (gering oder hoch)

b) Nennen Sie zwei Rendering-Techniken, um bei der Darstellung zusätzliche Oberflächendetails zu erzeugen und erklären Sie den Unterschied zwischen diesen! **(5 Punkte)**

c) Gegeben sei das Dreieck $\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}$ mit Texturkoordinaten $\mathbf{t_i} = (s_i, t_i),\ i \in \{0, 1, 2\}$ aus Abbildung 3. Berechnen Sie die Texturkoordinaten (s_x, t_x) für den Punkt $\mathbf{x}$, sowie den Wert bei Nearest Neighbor-Interpolation, wenn der Texture Wrapping-Modus `GL_MIRRORED_REPEAT` verwendet wird! **(8 Punkte)**



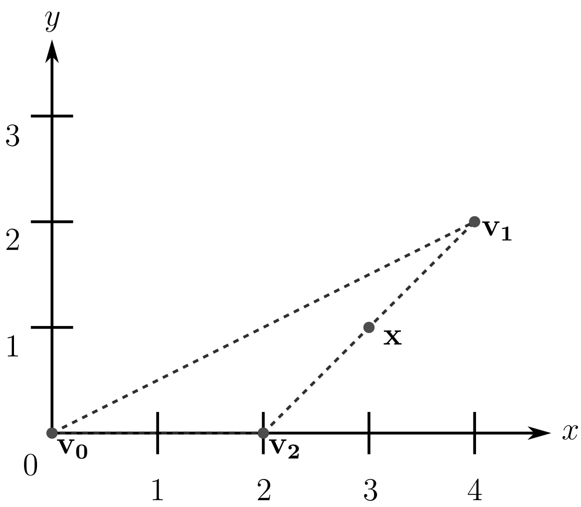
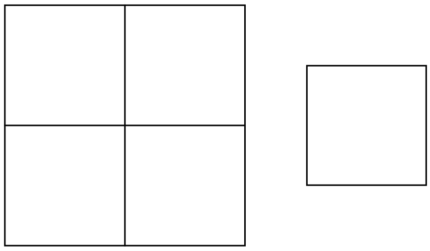
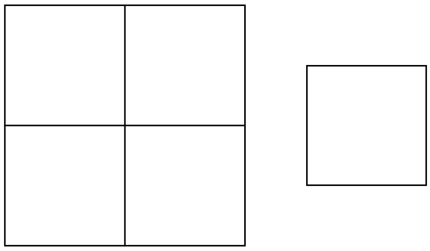


Abbildung 3

d) Um Artefakte bei Verkleinerung (Minification) zu reduzieren, werden Texturen mithilfe von Mip Maps vorgefiltert. Berechnen Sie die Mip Map-Stufen der Textur aus Abbildung 3 und tragen Sie sie in der folgenden Zeichnung ein! Zur Korrektur können Sie die zweite Zeichnung verwenden. *Kennzeichnen Sie eindeutig, welche Zeichnung bewertet werden soll!* **(4 Punkte)**





e) Es findet nun ein Texturzugriff mit trilinearer Interpolation mit berechneter Mip Map-Stufe $n = 0.8$ statt. Auf welche vorberechneten Mip Maps wird zugegriffen und wie werden sie gewichtet? **(4 Punkte)**

Aufgabe 7: OpenGL Pipeline (16 Punkte)

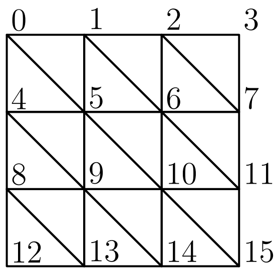
a) Geben Sie die Reihenfolge folgender Stufen der OpenGL-Pipeline an! **(3 Punkte)**
Hinweis: Die Nummern der Stufen reichen aus.

1) Rasterisierung
2) Fragment Shader
3) Vertex Shader
4) Geometry Shader
5) Fragment Operationen

b) Der Vertex Shader gibt homogene Koordinaten nach der Projektionstransformation aus. Nennen Sie zwei spätere, nicht-programmierbare Schritte in der Grafik-Pipeline, die ohne homogene Koordinaten *nicht* korrekt möglich sind! Begründen Sie stichpunktartig! **(4 Punkte)**

Matrikelnummer: ____________________

c) In dieser Aufgabe geht es um die Rasterisierung eines Dreiecksnetzes mit 4×4 Vertices wie unten abgebildet.



Wie viele Aufrufe N des Vertex Shaders werden jeweils *mindestens* erzeugt, wenn das abgebildete Dreiecksnetz mit den folgenden Aufrufen gezeichnet wird? **(9 Punkte)**

Für diese Aufgabe gilt:

- Die Angabe Ihrer Rechnung reicht aus, das Endergebnis ist nicht notwendig.
- Bei `GL_TRIANGLE_STRIP` wird ein neuer Dreiecksstreifen begonnen, indem zwei zusätzliche Dreiecke ohne Flächeninhalt übergeben werden, deren Rasterisierung dann keine Fragmente erzeugt.

`glDrawArrays(GL_TRIANGLES, ...)`:

$N =$

`glDrawArrays(GL_TRIANGLE_STRIP, ...)`:

$N =$

`glDrawElements(GL_TRIANGLES, ...)`:

$N =$

Aufgabe 8: OpenGL Blending (6 Punkte)

Im Folgenden werden verschiedene Situationen beschrieben, für die Blending verwendet wird. Geben Sie jeweils an, welche OpenGL-Konfiguration notwendig ist, um die gewünschte Ausgabe zu erzeugen

Hinweis: Es genügt, jeweils die Zahlen der Konstanten einzutragen.

Mögliche Parameter:
(1) `GL_ZERO`, (2) `GL_ONE`,
(3) `GL_SRC_COLOR`, (4) `GL_ONE_MINUS_SRC_COLOR`, (5) `GL_DST_COLOR`,
(6) `GL_ONE_MINUS_DST_COLOR`, (7) `GL_SRC_ALPHA`,
(8) `GL_ONE_MINUS_SRC_ALPHA`, (9) `GL_DST_ALPHA`,
(0) `GL_ONE_MINUS_DST_ALPHA`.

Mögliche Funktionen:
(1) `GL_FUNC_ADD`, (2) `GL_FUNC_SUBTRACT`,
(3) `GL_FUNC_REVERSE_SUBTRACT`, (4) `GL_FUNC_MIN`, (5) `GL_FUNC_MAX`.

a) Im Framebuffer liegt bereits die sichtbare opake Geometrie vor. Nun soll semitransparente Geometrie mit einem Alpha-Wert darüber gezeichnet werden. **(3 Punkte)**

```
glBlendFunc(                          ,                          );

glBlendEquation (                                                );
```

b) Eine Szene soll mit mehreren Lichtquellen beleuchtet werden. Die Szene wurde bereits mit der Beleuchtung der ersten Lichtquelle gezeichnet. Nun wird die Szene mehrfach gezeichnet, um jeweils die Beleuchtung einer weiteren Lichtquelle zu akkumulieren. **(3 Punkte)**

```
glBlendFunc(                          ,                          );

glBlendEquation (                                                );
```

Aufgabe 9: OpenGL Shader (38 Punkte)

In dieser Aufgabe geht es um das Rendering eines reflektierenden Metallobjekts mittels OpenGL GLSL-Shader. Bei der Schattierung soll Normal Mapping verwendet und durch eine Environment Map beleuchtet werden.

a) Zunächst soll das Objekt wie beschrieben dargestellt werden.

   i) Verwenden Sie Phong oder Gouraud Shading? Begründen Sie kurz! **(2 Punkte)**

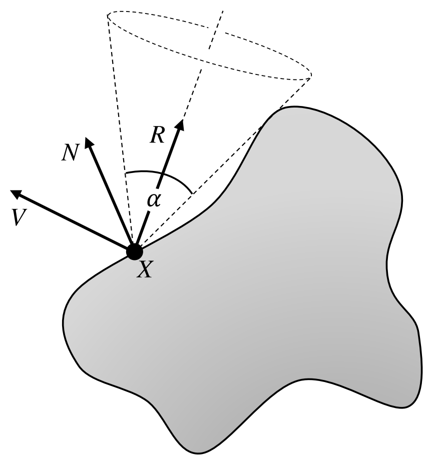
   ii) Die Rauheit der Metalloberfläche soll über der Oberfläche variieren. Welche Texturierungstechnik ermöglicht das? Was bedeutet dies für die Beleuchtung durch die Environment Map? **(3 Punkte)**

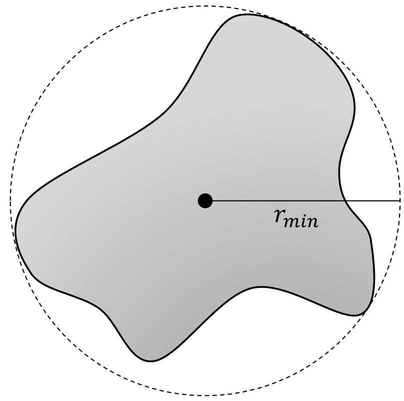
   iii) Ist es möglich die geometrischen Vertex-Normalen während des Rendering-Vorgangs in einem Geometry Shader zu berechnen? Begründen Sie! **(3 Punkte)**

b) Bei Environment Mapping kann Selbstverschattung nicht ohne Weiteres berücksichtigt werden und soll nun durch einen Rendering-Trick hinzugefügt werden. Hierzu steht das Metallobjekt zusätzlich als Distanzfunktion zur Verfügung, die bereits implementiert ist:

```
float DF( vec3 x ) { ... }
```

Die Distanzfunktion `DF(.)` ist nicht vorzeichenbehaftet, d.h. sie gibt Werte größergleich 0 zurück. Zudem ist bekannt, dass das Objekt durch eine Hüllkugel mit dem Radius r_{min} umschlossen werden kann.





Im Fragment Shader sollen Sie nun für einen Oberflächenpunkt folgende Schritte durchführen:

- Berechnen Sie die Reflexionsrichtung R für den Zugriff auf die Environment Map. Hierzu ist der Oberflächenpunkt X, die Normale N und die Betrachterrichtung V in Weltkoordinaten gegeben.
- Mittels Sphere Tracing sollen Sie den Reflexionsstrahl R (beschrieben durch $r(t) = X + t \cdot R$) auf Schnitt testen. Überlegen Sie sich geeignete Abbruchkriterien für das Sphere Tracing, die *nicht* auf einer festen Schrittzahl basieren!
- Um eine weiche (nicht-binäre) Verschattung darzustellen, bestimmen Sie *approximativ* den maximalen Öffnungswinkel α eines Kegels zentriert um den Reflexionsstrahl mit der Spitze in X (siehe Abbildung). Für die approximative Berechnung verwenden Sie keine zusätzlichen Auswertungen von `DF(.)`!

☐ Achten Sie auf mögliche numerische Probleme und erläutern Sie stichpunktartig (unterhalb des Shader) wie Sie vorgehen! **(30 Punkte)**

```
#define M_PI 3.14159265358979323846264338327950

uniform float r_min = ...

void float DF( vec3 x ) { ... }

void main() { ...

vec3 X, N, V;

// Berechne X, N, V (Sie müssen hier nichts bearbeiten!)
...

float alpha = M_PI;        // Öffnungswinkel des Kegels
bool  hit = false;         // true, falls R das Objekt schneidet
bool  done = false;        // true, wenn Sphere Tracing beendet werden kann

// Berechnen Sie hier die Reflexionsrichtung:

vec3 R =


// Sphere Tracing: aktuelle Position

vec3 p =


while ( !hit && !done )
{





}

// Shading mit R, Textur-Lookup, alpha etc. Sie müssen hier nichts bearbeiten!
...
} // Ende main()
```

Begründung/Erklärung: an welchen Stellen im GLSL-Programmcode haben Sie auf numerische Probleme geachtet?

Aufgabe 10: Prozedurale Modellierung (17 Punkte)

a) In dieser Aufgabe geht es um Lattice Value-Rauschfunktion. Diese nutzen eine deterministische Gitter-Funktion `float g(int s, int t)`, die Werte aus $[0, 1)$ für ganzzahlige Gitterkoordinaten liefert.

i) Welche Anforderung an Rauschfunktionen wird in der folgenden Implementierung verletzt? Begründen Sie Ihre Antwort stichpunktartig! **(3 Punkte)**

```
float bad_noise(float x, float y) {
    int x = int(floor(x));
    int y = int(floor(y));
    return g(x, y);
}
```

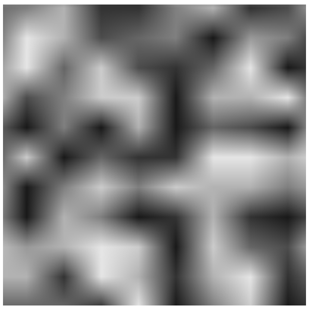
ii) Definieren Sie eine verbesserte Rauschfunktion in GLSL oder Pseudocode, die die Anforderung erfüllt! **(8 Punkte)**

Hinweise: `float mix(float min, float max, float a)` interpoliert für `a` $\in [0, 1]$ zwischen `min` und `max`. `float floor(float v)` rundet `v` ab und `float fract(float v)` liefert den Dezimalteil von `v`.

```
float noise(float x, float y){

}
```

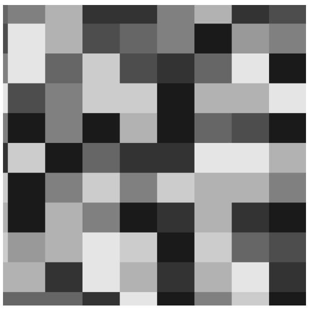
b) Geben Sie für jede der folgenden vier Varianten an, *welche* Eigenschaft im Vergleich zur der oben gegebenen und dargestellten Rauschfunktion `noise2(x, y)` geändert wurde und *wie* sie geändert wurde! **(6 Punkte)**



Originale Rauschfunktion `noise2`

Geänderte Eigenschaft *und* Änderung Variante 1
Antwort:

Geänderte Eigenschaft *und* Änderung Variante 2
Antwort:



Geänderte Eigenschaft *und* Änderung Variante 3
Antwort:

Geänderte Eigenschaft *und* Änderung Variante 4
Antwort:

Aufgabe 11: Bézier-Kurven (16 Punkte)

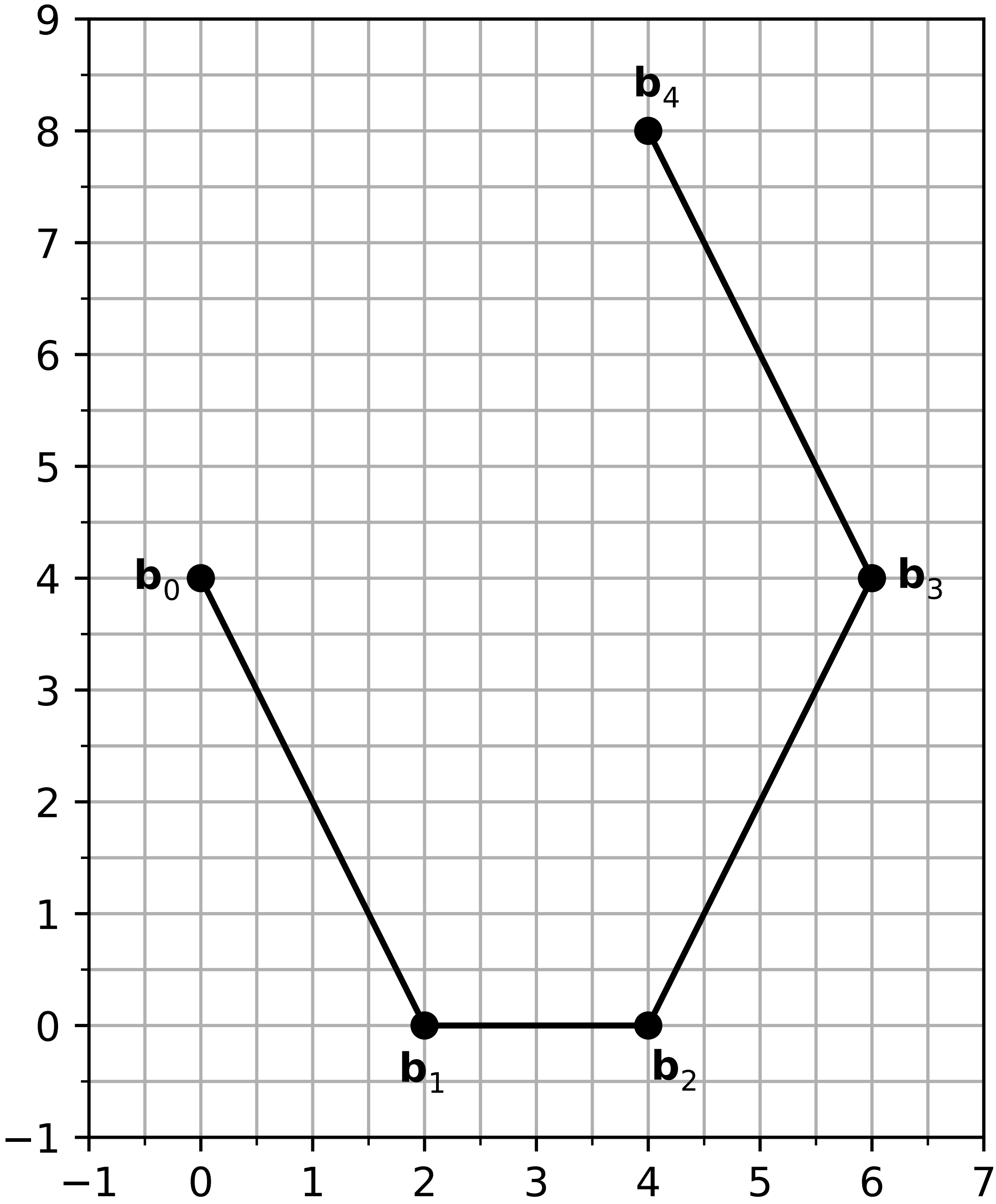
a) Gegeben sei die Bézier-Kurve $F(u) = \sum_{i=0}^{4} \mathbf{b}_i B_i^4(u)$ vom Grad 4 mit $u \in [0,1]$ und

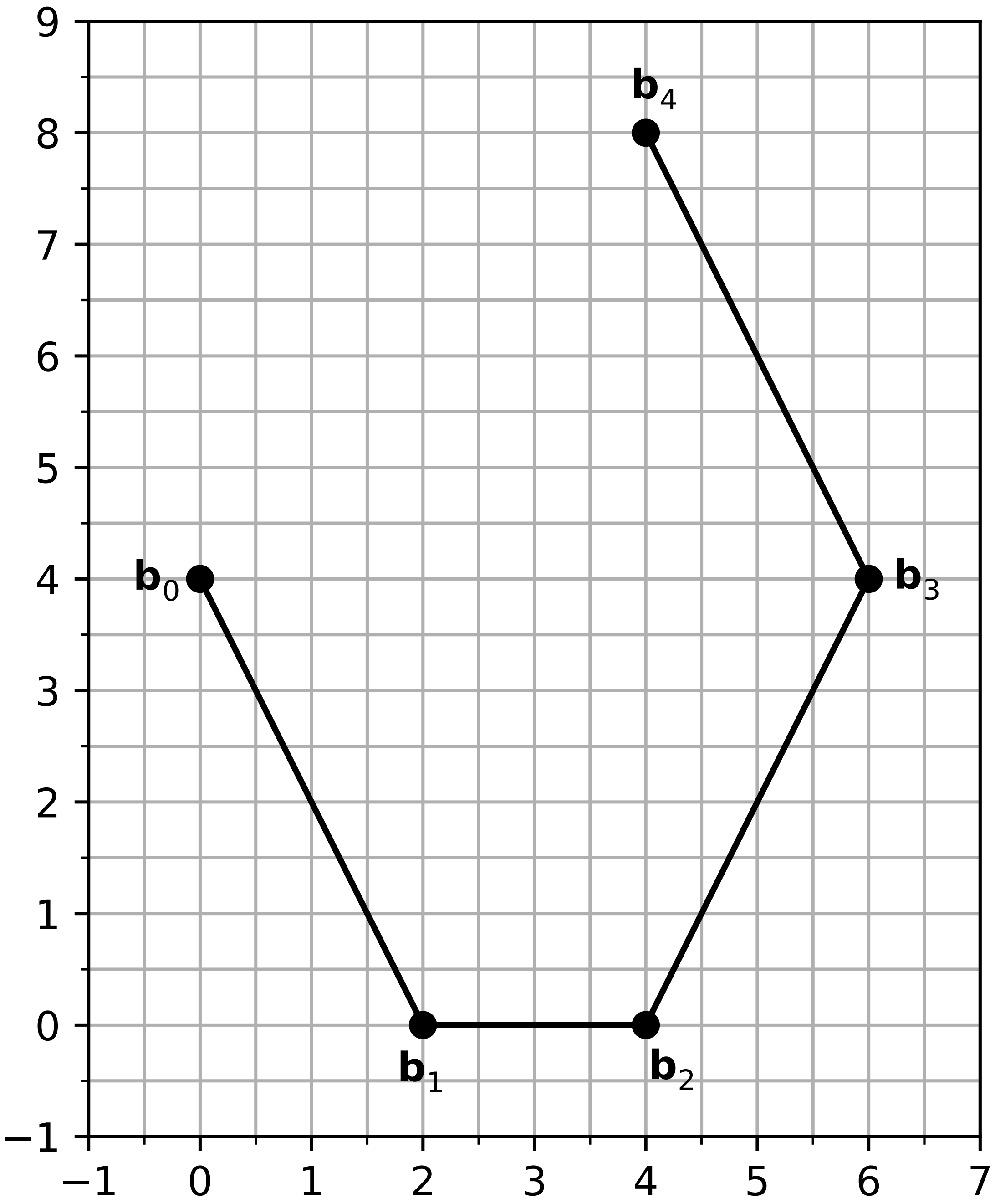
$$\mathbf{b}_0 = (0,4), \qquad \mathbf{b}_1 = (2,0), \qquad \mathbf{b}_2 = (4,0), \qquad \mathbf{b}_3 = (6,4), \qquad \mathbf{b}_4 = (4,8).$$

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!* **(8 Punkte)**

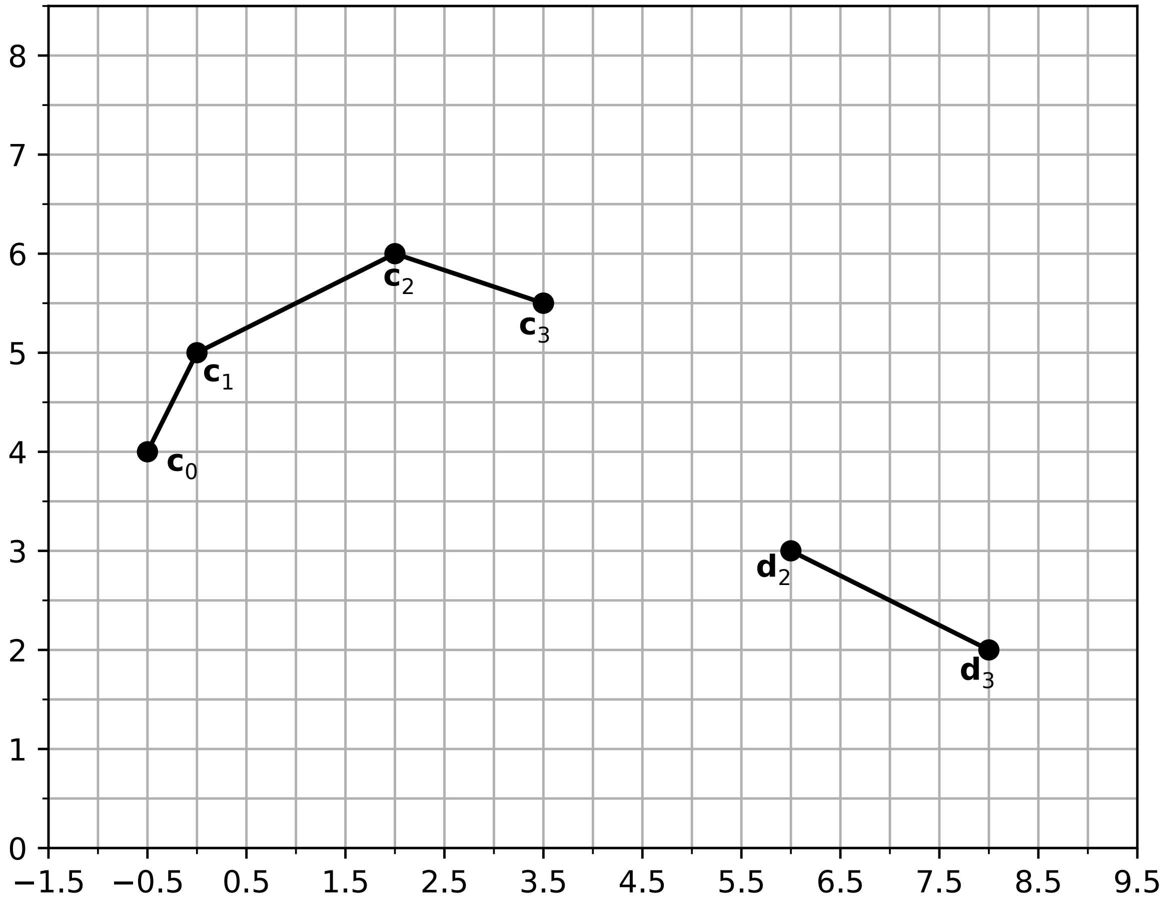




9
8
7
6
5
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
7
$\mathbf{b}_0$
$\mathbf{b}_1$
$\mathbf{b}_2$
$\mathbf{b}_3$
$\mathbf{b}_4$

Matrikelnummer: ____________________

b) Gegeben seien die Bézier-Kurven $G(u) = \sum_{i=0}^{3} \mathbf{c}_i B_i^3(u)$ und $H(u) = \sum_{i=0}^{3} \mathbf{d}_i B_i^3(u)$ vom Grad 3 mit $u \in [0, 1]$. Bestimmen Sie die Punkte d_0 und d_1 so, dass sich zwischen G und H ein C^2-stetiger Übergang ergibt! Begründen Sie mit dem Wissen aus der Vorlesung, dass die von Ihnen gewählten Punkte d_0 und d_1 in einem C^2-stetigem Übergang zwischen G und H resultieren! **(8 Punkte)**



$\mathbf{d}_0 =$ ______________ $\mathbf{d}_1 =$ ______________

Begründung C^2-Stetigkeit:

C^0 stetig, weil ...

C^1 stetig, weil ...

C^2 stetig, weil ...