Karlsruher Institut für Technologie

Institut für Betriebs- und Dialogsysteme
Lehrstuhl für Computergrafik
Prof. Dr.-Ing. Carsten Dachsbacher

Nachklausur Klausur Computergrafik WS 2011/12

Dienstag, 13. MÃďrz 2012

Kleben Sie hier **nach Bearbeitung der Klausur** den Aufkleber hin.

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 Blätter mit 9 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 die betreffende Box ganz aus:
  Falsche Antworten führen zu Punktabzug.
  Jede Multiple-Choice-Aufgabe wird mit mindestens 0 Punkten bewertet.
- Kleben Sie **nach Bearbeitung der Klausur** den Aufkleber mit Ihrem Namen und Ihrer Matrikelnummer auf dieses Deckblatt.

Aufgabe	1	2	3	4	5	6	7	8	9	**Gesamt**
Erreichte Punkte										
Mögliche Punkte	10	12	11	15	19	13	13	16	11	**120**

Note

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 1: Wahrnehmung und Farbräume (10 Punkte)

a) Eine Grafikkarte ist an ein Anzeigegerät mit einem Gamma-Wert von 2.0 angeschlossen und muss eine entsprechende Gamma-Korrektur durchführen. *Berechnen Sie den Intensitätswert*, den die Grafikkarte an das Anzeigegerät senden muss, um eine Ausgabe mit der *Hälfte* der Maximalintensität zu erreichen. (Der Wertebereich der Koeffizienten reicht von 0 bis zur Maximalintensität 1.0.) **(3 Punkte)**

b) Sie haben ein Bild im RGB-Farbraum gegeben und wollen den Helligkeitskontrast erhöhen. In welchen der in der Vorlesung vorgestellten Farbräume wandeln Sie es um, um diese Kontrasterhöhung möglichst einfach durchführen zu können? Welche Berechnung(en) führen Sie dazu auf den Koeffizienten dieses Farbraums aus? **(3 Punkte)**

Name: ______________________ Matrikelnummer: ______________________

c) Bewerten Sie die folgenden Aussagen, indem Sie *wahr* oder *falsch* ankreuzen. **(4 Punkte)**

Aussage	**Wahr**	**Falsch**
Um den Farbeindruck für einen Menschen eindeutig zu beschreiben, genügt ein Farbmodell mit 3 Koeffizienten.	☐	☐
Durch diese 3 Koeffizienten ist dann das Spektrum ebenso eindeutig festgelegt.	☐	☐
Der RGB-Einheitswürfel enthält alle sichtbaren Farben.	☐	☐
Der RGB-Einheitswürfel enthält Farben, die sich im CIE XYZ-Farbmodell nicht darstellen lassen.	☐	☐

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 2: Raytracing und prozedurale Modelle (12 Punkte)

a) In einem Whitted-Style Raytracer wurde ein Schnittpunkt eines Primärstrahls mit einem Primitiv gefunden. Nennen Sie **stichpunktartig** alle Schritte, die nötig sind, um die Farbe des zugehörigen Pixels zu bestimmen. **(4 Punkte)**

b) Was ist der konzeptuelle Unterschied zwischen Raymarching und Raycasting bzw. Raytracing? Nennen Sie einen Anwendungsfall, bei dem Sie Raymarching verwenden würden, und begründen Sie, warum Sie für diesen Fall Raytracing *nicht* sinnvoll einsetzen können. **(4 Punkte)**

Name: ______________________ Matrikelnummer: ______________________

c) Bewerten Sie die folgenden Aussagen, indem Sie *wahr* oder *falsch* ankreuzen. **(4 Punkte)**

Aussage	**Wahr**	**Falsch**
Prozedurale Modelle erlauben eine kompakte Beschreibung von Texturen oder Objekten, aber die Resultate sind oft nur schwer zu kontrollieren.	☐	☐
Eine ideale Noise-Funktion sollte weder bandbegrenzt sein noch räumliche Korrelationen aufweisen, um erkennbare Periodizitäten und monotone Strukturen zu vermeiden.	☐	☐
Um Turbulenz-Texturen zu erstellen, werden immer höhere Oktaven von Noise-Texturen aufsummiert und dabei immer stärker gewichtet, damit die Resultate so weit wie möglich gegen Weißes Rauschen konvergieren.	☐	☐
Constructive Solid Geometry ist eine Technik zum Modellieren von Festkörpern, bei der Objekte durch boolesche Operatoren kombiniert werden.	☐	☐

Name: ______________________ Matrikelnummer: ______________________

Aufgabe 3: Transformationen (11 Punkte)

In der Vorlesung haben Sie Transformationen, wie sie in der Computergrafik verwendet werden, kennengelernt.

a) Gegeben seien die folgenden orthonormalen Vektoren im $\mathbb{R}_3$:

$$\mathbf{u} = \left(-\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}\right)^T, \mathbf{v} = \left(-\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}\right)^T, \mathbf{w} = \left(\sqrt{\frac{2}{3}}, 0, \frac{1}{\sqrt{3}}\right)^T$$

Eine 3×3 Transformationsmatrix $\mathbf{M}$ transformiert die Koordinaten eines Punktes in kartesischen Koordinaten in das Koordinatensystem, das von $\mathbf{u}, \mathbf{v}$ und $\mathbf{w}$ aufgespannt wird.

- Um welchen Typ Transformation handelt es sich dabei? **(1 Punkt)**

- Nennen Sie die effizienteste Methode, die Inverse dieser Transformation zu finden. **(1 Punkt)**

b) Nennen Sie zwei Gründe für die Verwendung hierarchischer Modellierung. Welche Datenstruktur haben Sie in der Vorlesung kennengelernt, die dabei hilft, Transformationen beim Traversieren des Szenengraphen zu verwalten? **(3 Punkte)**

Name: ____________________ Matrikelnummer: ____________________

c) Betrachten Sie folgende Transformationsmatrizen in **homogenen 2D-Koordinaten**:

$$\mathbf{A} = \begin{pmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} & 0 \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2.5 \\ 0 & 0 & 1 \end{pmatrix}$$

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

Die Abbildung $\mathbf{p}'$ eines Punktes $\mathbf{p} \in \mathbb{R}^2$, mit $\mathbf{p} = (p_x, p_y, p_w)^T$, erhält man dabei durch $\mathbf{p}' = \mathbf{M}\mathbf{p}$.

Nennen Sie jeweils, um welche Transformation es sich handelt und wie deren Parameter sind (z.B. „Rotation in/gegen den Uhrzeigersinn um 123°.“). **(6 Punkte)**

Hinweis: Winkel sind im Bogenmaß angegeben.

A	
B	
C	
D	

Name: ____________________________ Matrikelnummer: __________________

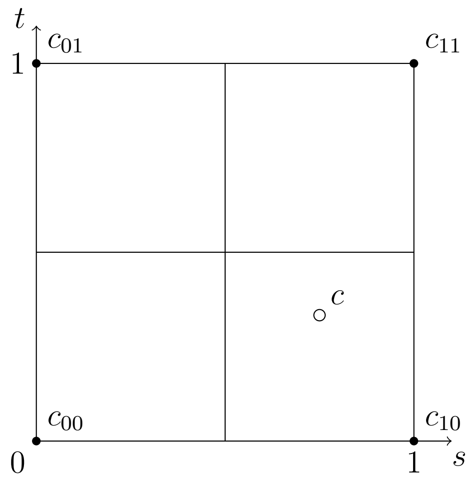
Aufgabe 4: Texturen (15 Punkte)

a) Was ist eine Environment-Map? Wofür wird sie eingesetzt? Welche Annahmen trifft man dabei? **(5 Punkte)**

b) Welches Problem beim Texture-Mapping löst Mip-Mapping? Erklären Sie *kurz* die Idee! **(6 Punkte)**

c) Im Folgenden ist eine 2×2 Graustufentextur mit Texelwerten $c_{00}, c_{10}, c_{01}, c_{11}$ gegeben. Die Texelwerte sind an den Ecken der Textur, definiert im Bereich $[0;1]^2$, gespeichert.

Geben Sie den Wert c an, der sich bei Auslesen der Textur mittels bilinearer Interpolation an der Stelle $(s,t) = \left(\frac{3}{4}, \frac{1}{3}\right)$ ergibt. **(4 Punkte)**

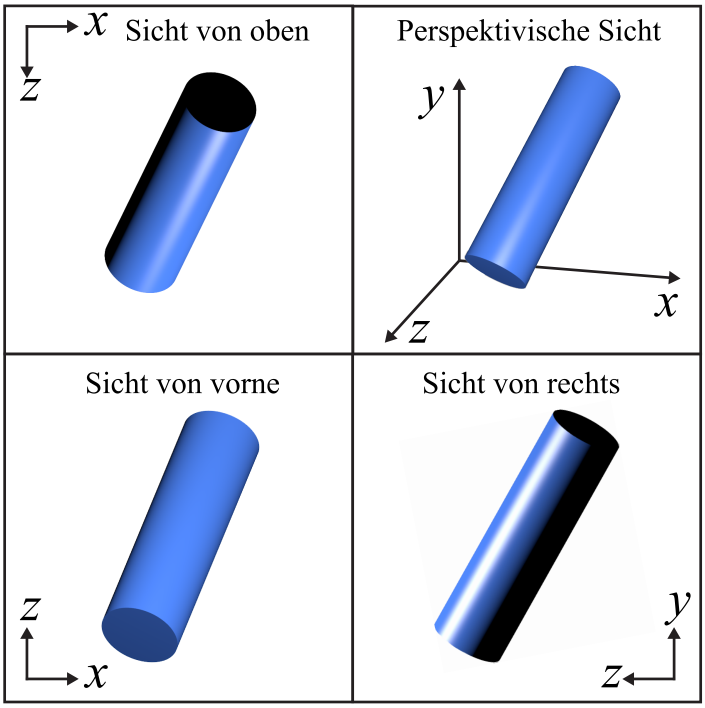


$c =$

Name: ______________________ Matrikelnummer: ______________

Aufgabe 5: Hierarchische Datenstrukturen (19 Punkte)

a) Nennen Sie vier Arten von Hüllkörpern *(bounding volumes)*, die Sie in der Vorlesung kennengelernt haben, und sortieren Sie sie danach, wie eng sie den in der Darstellung gegebenen Zylinder umschließen können. **(4 Punkte)**



eng umschliessend
weit umschliessend

b) In dieser Aufgabe betrachten wir einen Octree, eine Bounding-Volume-Hierarchie (BVH), und einen kD-Baum.

Alle drei Datenstrukturen sind so implementiert, dass sie Primitive nur in *Blattknoten* speichern. Primitive, die in mehreren Blattknoten gespeichert werden müssten, werden so geteilt, dass jeder Teil nur in einem Blatt gespeichert werden muss. An jedem Blattknoten werden *alle* dort gespeicherten Objekte getestet und der nächste Schnittpunkt zurückgegeben.

Bewerten Sie den Wahrheitsgehalt der Aussage, wenn sie auf die jeweilige Datenstruktur bezogen wird, indem Sie *wahr* oder *falsch* **ankreuzen**, und **begründen** Sie kurz Ihre Entscheidung. **(9 Punkte)**

Aussage:	„Der *erste* während der Traversierung gefundene Schnittpunkt ist zugleich der *nächste* Schnittpunkt zum Ursprung des Strahls."
Octree ☐ wahr ☐ falsch	
BVH ☐ wahr ☐ falsch	
kD-Baum ☐ wahr ☐ falsch	

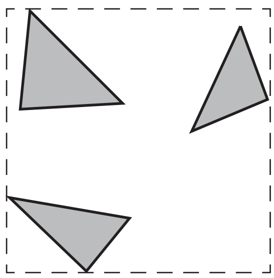
c) Konstruieren Sie für die Dreiecke in dem Kasten einen zweidimensionalen kD-Baum, wobei das *Objektmittel* (object median) *entlang der Achse der größten Ausdehnung* als Unterteilungsheuristik zu verwenden ist. Jedes Blatt des kD-Baums darf höchstens ein Dreieck enthalten.

Zeichnen Sie die Unterteilungslinie für jeden inneren Knoten des Baumes und **nummerieren Sie sie entsprechend der Tiefe** des Knotens im kD-Baum, wie im Beispiel am Ende der Aufgabe gezeigt ist. **(6 Punkte)**

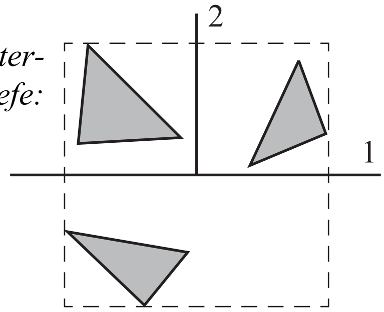
Beispiel

Hinweis: Das folgende Beispiel zeigt nur, wie Sie *Unterteilungen einzeichnen* und *nummerieren* sollen. Es zeigt *nicht*, wie unterteilt werden soll.

Ausgangszustand:



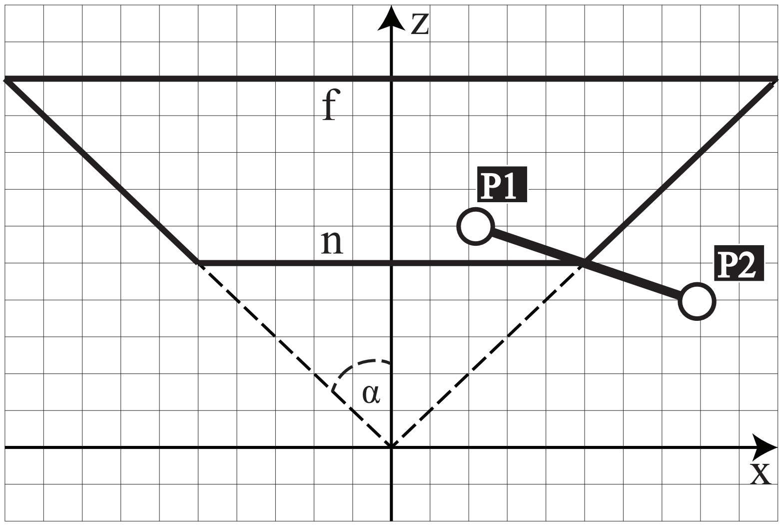
Gezeichnete Unterteilungen mit Tiefe:



Name: ______________________ Matrikelnummer: ______________________

Aufgabe 6: Projektive Transformationen und Clipping (13 Punkte)

Gegeben sei eine zweidimensionale perspektivische Projektion. Die zu rasterisierende Szene sei bereits in Kamerakoordinaten transformiert, so dass sich die Kamera im Ursprung befindet und in Richtung der *positiven* Z-Achse zeigt.



Die Sichtpyramide ist so skaliert, dass ihre Eckpunkte bei $\begin{pmatrix} \pm n \\ n \end{pmatrix}$ bzw. $\begin{pmatrix} \pm f \\ f \end{pmatrix}$ liegen, also der Öffnungswinkel $\alpha = 45°$ ist. Die 2D-Projektionsmatrix $\mathbf{M}_{proj}$ der Kamera ist:

$$\mathbf{M}_{proj} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & -20 \\ 0 & 1 & 0 \end{pmatrix}$$

Die kartesischen Koordinaten $\begin{pmatrix} x \\ z \end{pmatrix}$ der Eingabepunkte sind:

$$P_1 = \begin{pmatrix} 2 \\ 6 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 8 \\ 4 \end{pmatrix}$$

Sie sollen nun die Projektionstransformation auf die Strecke $\overline{P_1 P_2}$ anwenden und diese dann gegen die zweidimensionale Sichtpyramide clippen.

a) Transformieren Sie die Vertizes P_1, P_2 in homogene Clip-Koordinaten, und geben Sie die homogenen Koordinaten der transformierten Punkte *vor* der Dehomogenisierung an. **(3 Punkte)**

$P_{1h} =$ $\qquad\qquad$ $P_{2h} =$

b) Als nächstes müssen Sie die „window edge coordinates" der projizierten Punkte in Clip-Koordinaten berechnen. Welche Kanten müssen potentiell auf Schnitt mit der Strecke überprüft werden? **(4 Punkte)**

$WEC_{x=w}(P_1) =$ $WEC_{x=w}(P_2) =$

$WEC_{x=-w}(P_1) =$ $WEC_{x=-w}(P_2) =$

$WEC_{z=w}(P_1) =$ $WEC_{z=w}(P_2) =$

$WEC_{z=-w}(P_1) =$ $WEC_{z=-w}(P_2) =$

c) Wenden Sie nun α-Clipping für die Strecke $\overline{P_1 P_2}$ an und geben Sie den Lösungsweg an! Geben Sie ebenfalls die Endpunkte P_1' und P_2' der Teilstrecke in der Sichtpyramide in normalisierten Device-Koordinaten an. **(6 Punkte)**

Hinweis: Verdeutlichen Sie sich die Lage des Punktes, den Sie nach dem ersten α-Clipping-Schritt erhalten!

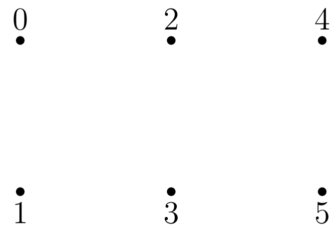
$P_1' =$ $P_2' =$

Name: ______________________ Matrikelnummer: ______________________

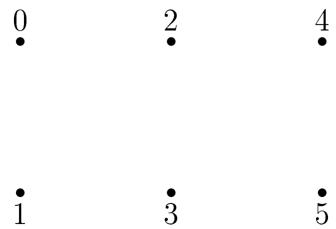
Aufgabe 7: OpenGL und Rasterisierung (13 Punkte)

a) Im Folgenden soll ein Dreiecksnetz mit der jeweils angegebenen Topologie unter Verwendung eines Indexbuffers gezeichnet werden. Zeichnen Sie die Kanten der entstehenden Dreiecke ein, wenn die Vertizes $0, 1, 2, 3, 4, 5$ wie in der Abbildung dargestellt liegen. **(4 Punkte)**

   I. *Triangle Strips* mit Indexbuffer $:= \{0, 1, 2, 3, 4, 5\}$



   II. *Triangle List* mit Indexbuffer $:= \{1, 3, 2, 3, 5, 4\}$



b) Es soll eine Szene aus opaken (vollständig undurchsichtigen) und semi-transparenten Dreiecksnetzen *effizient* gezeichnet werden. Es soll ein *nicht-kommutativer* Blending-Operator verwendet und ein möglichst korrektes Resultat erzielt werden.

Bringen Sie die folgenden Programmteile in die **richtige Reihenfolge**, indem Sie sie in aufsteigender Reihenfolge in der linken Spalte der Tabelle nummerieren: der erste Schritt wird mit „1" nummeriert, der zweite Schritt mit „2", usw.

Streichen Sie Schritte, die nicht unbedingt nötig sind. **(9 Punkte)**

Hinweis: Es gibt mehrere korrekte Lösungsabfolgen.

	Aktiviere das Schreiben in den Tiefenpuffer: `glDepthMask(GL_TRUE)`
	Deaktiviere das Schreiben in den Tiefenpuffer: `glDepthMask(GL_FALSE)`
	Aktiviere den Tiefentest: `glEnable(GL_DEPTH_TEST)`
	Deaktiviere den Tiefentest: `glDisable(GL_DEPTH_TEST)`
	Zeichne alle transparenten Dreiecksnetze in beliebiger Reihenfolge
	Sortiere alle transparenten Dreiecksnetze anhand ihrer Bounding-Boxen und zeichne sie in der Reihenfolge von hinten nach vorne.
	Zeichne alle opaken Dreiecksnetze in beliebiger Reihenfolge
	Sortiere alle opaken Dreiecksnetze anhand ihrer Bounding-Boxen und zeichne sie in der Reihenfolge von hinten nach vorne.
	Lösche den Farb- und Tiefenpuffer: `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)`
	Aktiviere Blending: `glEnable(GL_BLEND)`
	Deaktiviere Blending: `glDisable(GL_BLEND)`

Name: ____________________ Matrikelnummer: ____________________

Aufgabe 8: OpenGL-Shading-Language (16 Punkte)

Vervollständigen Sie die OpenGL-Shading-Language (GLSL) Shader-Programme auf den nächsten Seiten, sodass die nachfolgend deklarierte Funktion `computeLighting()` für ein Rendering mit *Phong Shading* eingesetzt wird.

```
float computeLighting(in vec3 position, // Oberflaechenpunkt
                      in vec3 normal,   // Normale
                      in vec3 lightPos, // Lichtposition
                      in float kd,      // k_d, Materialkoeffizient
                      in float I);      // Intensitaet der Lichtquelle I_L
```

Sie brauchen also *nicht* selbst die Beleuchtungsberechnung implementieren, rufen Sie stattdessen `computeLighting()` mit den richtigen Parametern auf.

Es gibt mehrere Lösungswege – deklarieren Sie alle notwendigen `in`, `out` oder `uniform` Variablen, die Sie für Ihre Lösung benötigen. Die Semantik der Eingabevariablen können Sie den Kommentaren im Shader Code entnehmen.

Die Shader-Programme befinden sich auf den nächsten Seiten.

a) **Vertex-Shader (10 Punkte)**

Name: ____________________ Matrikelnummer: ____________________

```
#version 140
uniform mat4 matMV;     // Modelviewmatrix zur Transformation von
                        // Objekt- nach Kamerakoordinaten
uniform mat4 matP;      // Projektionsmatrix
uniform mat3 matN;      // Normalenmatrix

uniform vec3 I;         // Intensitaet der Lichtquelle I_L
uniform vec3 lightPos;  // Lichtposition in Kamerakoordinaten

in vec3 in_v;           // enthaelt Vertex in Objektkoordinaten
in vec3 in_n;           // enthaelt Normalenvektor mit Laenge 1
in float kd;            // Materialkoeffizient k_d
```

```
void main(void)
{

    gl_Position =
}
```

Name: ____________________ Matrikelnummer: ____________________

b) **Fragment-Shader (6 Punkte)**

```
#version 140
```

```
out vec4 out_color; // Ausgabefarbe bestimmt unter Verwendung
                    // von computeLighting()

void main(void)
{
```

```
    out_color =
}
```

Name: ______________________ Matrikelnummer: ______________

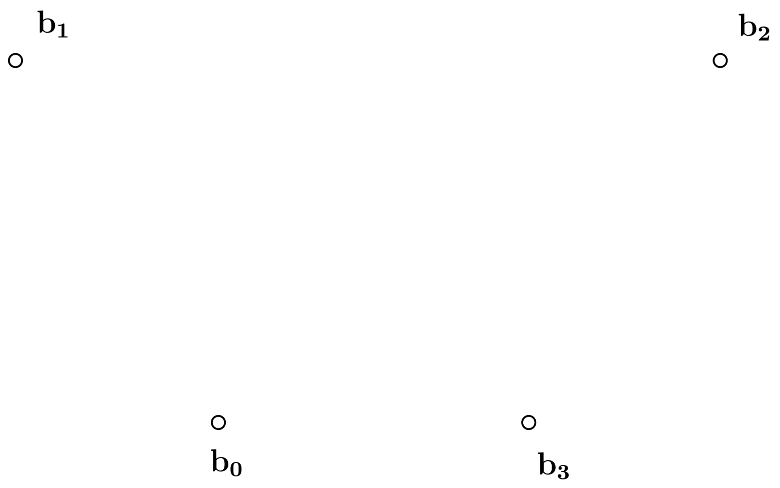
Aufgabe 9: Bézierkurven und Béziersplines (11 Punkte)

a) Nennen Sie drei wichtige Eigenschaften der Bézierkurven, die Sie in der Vorlesung kennengelernt haben. **(3 Punkte)**

b) Gegeben sei die Bézierkurve $\mathbf{b}(u) = \sum_{i=0}^{3} \mathbf{b}_i B_i^n(u)$ mit Kontrollpunkten $\mathbf{b}_i$. **(4 Punkte)**

I) Werten Sie die Bézierkurve zeichnerisch an der Stelle $u = \frac{1}{3}$ mit dem **de-Casteljau-Algorithmus** aus. Markieren Sie den **Punkt** $\mathbf{b}(\frac{1}{3})$. Eine Skizze genügt!

II) Skizzieren Sie ebenfalls den **Kurvenverlauf** der Bezierkurve.



c) Gegeben sei der folgende kubische Bézierspline. Der Punkt $\mathbf{b_3} = \mathbf{c_0}$ werde auf die neue Position $\mathbf{b'_3} = \mathbf{c'_0}$ verschoben. Markieren oder benennen Sie den Teil des Splines, der sich dadurch verändert. **(4 Punkte)**