Karlsruher Institut für Technologie

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

Klausur Computergraphik — WS 2010/11

Mittwoch, 20. April 2011

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 **10 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	6	8	4	5	8	6	6	9	8	**60**

Note

Name: ______________________________ Matrikelnummer: ____________

Aufgabe 1: Wahrnehmung, Farbe und Rasterbilder (6 Punkte)

a) Was versteht man unter *Metamerie* beim Farbsehen des Menschen? (1 Punkt)

b) Erklären Sie, was man unter *Schwarzkörperstrahlung* und *Farbtemperatur* versteht! (2 Punkte)

c) Gegeben sind folgende Filterkernel:

$$A = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline\end{array} \quad B = \begin{array}{|c|c|c|}\hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline\end{array} \quad C = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline\end{array} \quad D = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline\end{array}$$

Beschreiben Sie jeweils knapp in Worten, was die einzelnen Filter A, B, C und D bei einer diskreten Faltung mit einem Graustufenbild bewirken. (2 Punkte)

d) Was versteht man unter einem *normalisierten* Filterkernel? Welche globale Eigenschaft eines Bildes ändert sich, wenn ein Filterkernel nicht normalisiert ist? (1 Punkt)

Name: ________________________________ Matrikelnummer: ____________

Aufgabe 2: Prozedurale Modelle (8 Punkte)

a) Was sind Turbulenzfunktionen und wie können Sie aus Noise-Funktionen gebildet werden? Nennen Sie zwei Beispiele, wofür sie in der Computergraphik eingesetzt werden können! (2 Punkte)

b) Es sind folgende drei D0L-Systeme (deterministische, kontext-freie Lindenmayer-Systeme) $G_i = (V, \Sigma, S_i, P_i)$ mit $V = \Sigma = \{F, f, +, -, [,]\}$ für $i \in \{1, 2, 3\}$ definiert:

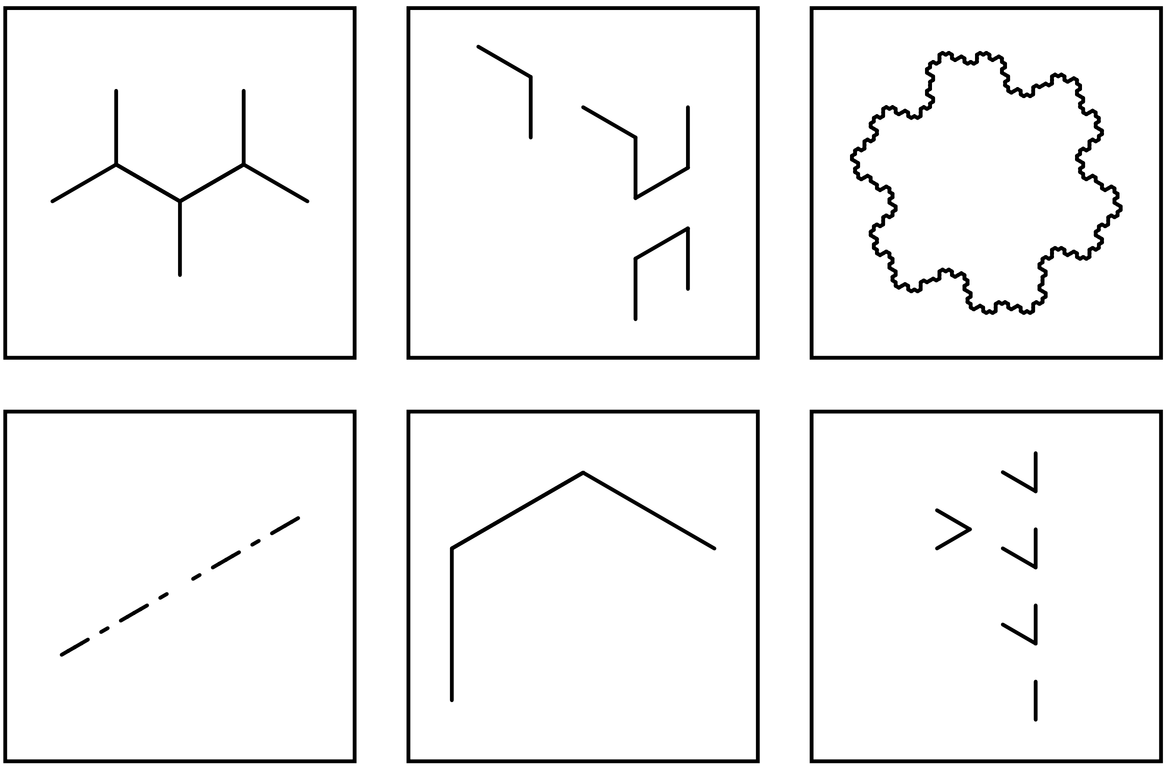
1) $S_1 = F$; $P_1 = \{F \mapsto F[+F][-F]\}$
2) $S_2 = F$; $P_2 = \{F \mapsto F[+F]\}$
3) $S_3 = F$; $P_3 = \{F \mapsto F[f - F]fF\}$

Führen Sie für *jedes* der L-Systeme *jeweils zwei Ersetzungsschritte*, ausgehend vom Startsymbol, durch und geben Sie die erzeugten Worte an. (3 Punkte)

c) Die nachfolgenden sechs Bilder zeigen Turtle-Grafiken ($\delta = 60$), von denen drei mit den L-Systemen aus Teilaufgabe b) erstellt wurden. Die Interpretation der Symbole ist dabei wie folgt:

- F: Vorwärtsbewegung *mit* Zeichnen einer Strecke
- f: Vorwärtsbewegung *ohne* Zeichnen eines Strecke
- $+$ und $-$: Rechts-/Linksdrehung um den Winkel δ
- [und]: Push (Sichern) und Pop (Wiederherstellen) des Zustands auf dem Stack

Ordnen Sie für $i \in \{1, 2, 3\}$ die L-Systeme G_i den richtigen Grafiken durch entsprechende Beschriftung zu. (3 Punkte)

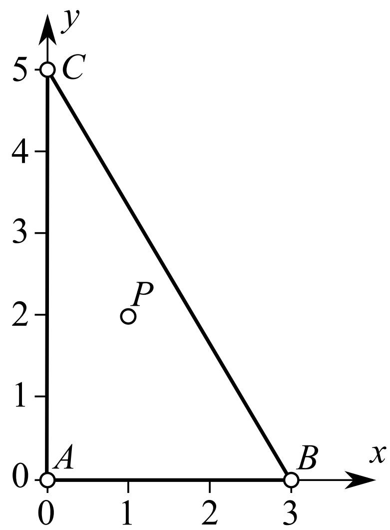


Name: ______________________________ Matrikelnummer: ______________

Aufgabe 3: Supersampling und Baryzentrische Koordinaten (4 Punkte)

a) Erklären Sie adaptives Supersampling und stochastisches Supersampling mit Stratifikation und die Unterschiede dazwischen! (2 Punkte)

b) Gegeben ist das Dreieck ABC mit den Eckpunkten A = (0, 0), B = (3, 0), C = (0, 5). Berechnen Sie für den Punkt P = (1, 2) die baryzentrischen Koordinaten λ_A, λ_B, λ_C bezüglich der Eckpunkte A, B, C des Dreiecks. (2 Punkte)



Name: ______________________________ Matrikelnummer: ____________

Aufgabe 4: Texturen (5 Punkte)

a) Wie wird aus einer Textur eine Mip-Map-Pyramide erstellt? Wie hoch ist der zusätzliche Speicherbedarf? (1,5 Punkte)

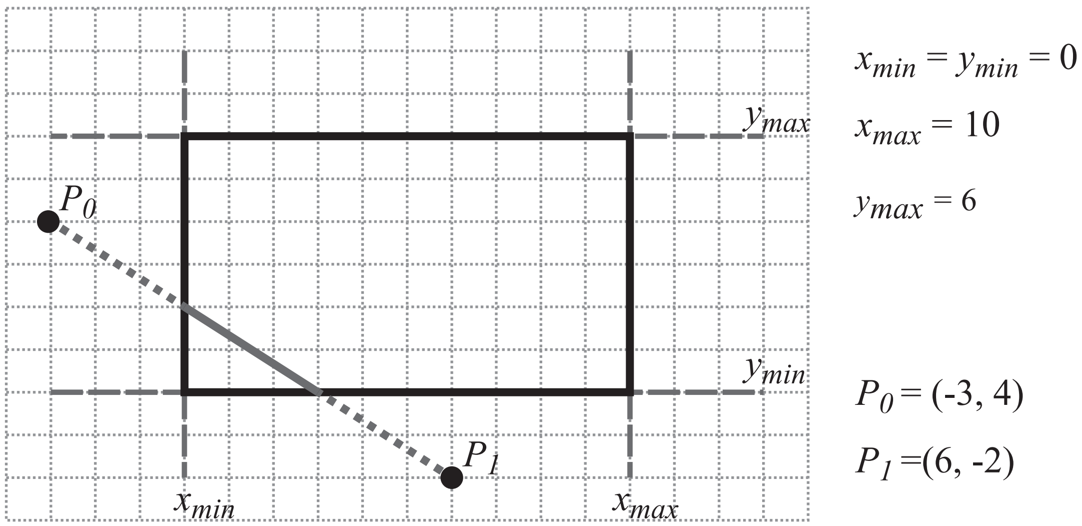
b) Welche Probleme bei der Texturfilterung *im Fall der Verkleinerung* ("Texture Minification") löst Mip-Mapping, welche nicht? (1,5 Punkte)

c) Wofür verwendet man Environment Mapping? Was speichert eine Environment Map und welche vereinfachenden Annahmen werden bei der Anwendung getroffen? (2 Punkte)

Name: ______________________________ Matrikelnummer: ____________

Aufgabe 5: α-Clipping (8 Punkte)

Verwenden Sie den α-Clipping-Algorithmus, um die Strecke $P = \overline{P_0 P_1}$ am Viewport begrenzt durch x_{min}, x_{max}, y_{min}, y_{max} abzuschneiden (siehe Abbildung):



a) Berechnen Sie die *Window Edge Coordinates* (WEC) für beide Endpunkte P_0 und P_1 bezüglich des Viewports $(x_{min}, x_{max}, y_{min}, y_{max})$. (2 Punkte)

$\mathrm{WEC}_{x_{min}}(P_0) =$ $\qquad$ $\mathrm{WEC}_{x_{min}}(P_1) =$

$\mathrm{WEC}_{x_{max}}(P_0) =$ $\qquad$ $\mathrm{WEC}_{x_{max}}(P_1) =$

$\mathrm{WEC}_{y_{min}}(P_0) =$ $\qquad$ $\mathrm{WEC}_{y_{min}}(P_1) =$

$\mathrm{WEC}_{y_{max}}(P_0) =$ $\qquad$ $\mathrm{WEC}_{y_{max}}(P_1) =$

b) Wenn Sie alleine die WEC betrachten, welche Kanten des Viewports werden dann potenziell geschnitten? Begründen Sie Ihre Antwort mit Hilfe der Outcodes! (2 Punkte)

c) Führen Sie den α-Clipping-Algorithmus *nur* für die Kanten durch, die potenziell geschnitten werden, und bestimmen Sie den Teil der Strecke P, der innerhalb des Viewports liegt. Beschreiben Sie jeden Schritt des Algorithmus, bis Sie das Endergebnis erhalten. (4 Punkte)

Name: ______________________________ Matrikelnummer: ____________

Aufgabe 6: Rasterisierung (6 Punkte)

a) Erklären Sie den Unterschied zwischen *Gouraud-* und *Phong-Shading* bei der Schattierung eines Dreiecks mit einem Rasterisierungsverfahren! (1 Punkt)

b) Was versteht man unter einem *Accumulation-Buffer*, wie Sie ihn von OpenGL kennen? Nennen Sie zwei Beispiele für Effekte, die sich damit erreichen lassen! (2 Punkte)

c) Bewerten Sie folgende Aussagen, indem Sie *wahr* oder *falsch* ankreuzen: (3 Punkte)

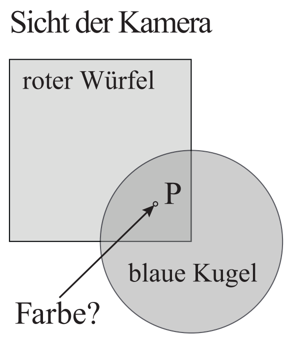
Aussage	**Wahr**	**Falsch**
T-Vertices können bei Phong-Shading Artefakte verursachen.	☐	☐
Z-Fighting kann durch die Repräsentation der Tiefenwerte mit beschränkter Genauigkeit im Tiefenpuffer entstehen.	☐	☐
Je feiner eine Oberfläche tesselliert wird, umso geringer werden die Unterschiede zwischen Gouraud- und Phong-Shading.	☐	☐
Bei der Rasterisierung ist eine perspektivisch korrekte Abbildung der Textur aufwendiger als eine affine, da pro Pixel eine zusätzliche Division benötigt wird.	☐	☐
Der Scissor-Test dient dazu, durchsichtige Teile einer Oberfläche gemäß einer Textur wegzuschneiden.	☐	☐
Screendoor-Transparency stellt transparente Objekte mittels Blending dar.	☐	☐

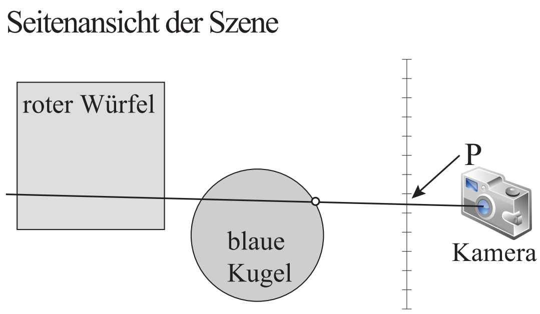
Name: ______________________________ Matrikelnummer: ______________

Name: ______________________________ Matrikelnummer: ____________

Aufgabe 7: OpenGL (6 Punkte)

a) Gegeben sei folgende Szene, die aus einem roten Würfel und einer blauen Kugel besteht:





Aus Sicht der Kamera befindet sich die Kugel *vor* dem Würfel. Die Farbe der blauen Kugel wird mit dem Befehl `glColor4f(0.0, 0.0, 1.0, 0.5);` gesetzt. Die Farbe des roten Würfels mit `glColor4f(1.0, 0.0, 0.0, 0.5);`. Das bedeutet auch, dass beide Objekte einen Alpha-Wert von 0.5 haben und somit semi-transparent sind.

Folgende OpenGL-Zustände sind vor dem Zeichnen gesetzt und der Hintergrund und Z-Buffer wurden wie folgt gelöscht:

```
glDepthFunc(GL_LEQUAL);
glCullFace(GL_FRONT);
glEnable(GL_BLEND);
glBlendEquation(GL_FUNC_ADD);
glDisable(GL_LIGHTING);

glColor4f(0.0, 0.0, 0.0, 0.0);
glClearDepth(1.0);
glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
```

Berechnen Sie für die folgenden zwei Fälle die Farbe und den Alpha-Wert des Pixels P, in der sich die beiden Objekte überschneiden (siehe Skizze). Dabei werden *zuerst* die OpenGL-Befehle für das Zeichnen der *blauen Kugel* im Vordergrund aufgerufen, *danach* der *rote Würfel* im Hintergrund gezeichnet. (3 Punkte)

1)
```
glEnable(GL_DEPTH_TEST);
glBlendFunc(GL_ONE_MINUS_SRC_ALPHA, GL_SRC_ALPHA);
```

2)
```
glDisable(GL_DEPTH_TEST);
glBlendFunc(GL_ONE, GL_SRC_ALPHA);
```

Name: ________________________________ Matrikelnummer: ____________

b) Bringen Sie die folgende Operationen in die richtige Reihenfolge, wie sie in der Fixed-Function-Pipeline von OpenGL ausgeführt werden, indem Sie die Ziffern 1 bis 5 in die Spalte “Reihenfolge” eintragen: (2 Punkte)

Reihenfolge	*Operation*
	Model-View-Transformation anwenden
	Tiefentest
	Texturierung
	Projektionstransformation anwenden
	Beleuchtungsberechnung

c) An welcher Stelle innerhalb der Reihenfolge, die Sie in Teilaufgabe b) angegeben haben, wird das Clipping durchgeführt? Begründen Sie Ihre Antwort. (1 Punkt)

Name: ______________________________ Matrikelnummer: ____________

Aufgabe 8: Vertex- und Fragment-Shader (9 Punkte)

a) Sie sollen die Funktionalität der Fixed-Function-Pipeline von OpenGL exakt mittels Vertex- und Fragment-Shader nachbilden. Kreuzen Sie an, welche der folgenden Operationen Sie im Vertex-Shader, im Fragment-Shader oder in keinem davon implementieren müssen: (3 Punkte)

Operation	**Vertex-Shader**	**Fragment-Shader**	**Weder noch**
Model-View-Transformation anwenden	☐	☐	☐
Tiefentest	☐	☐	☐
Texturierung	☐	☐	☐
Projektionstransformation anwenden	☐	☐	☐
Beleuchtungsberechnung	☐	☐	☐
Clipping	☐	☐	☐

b) Erläutern Sie die Parametertypen `attribute`, `uniform` und `varying` der OpenGL Shading Language und grenzen Sie diese voneinander ab! (3 Punkte)

Name: ______________________________ Matrikelnummer: ____________

c) Ergänzen Sie die folgenden Vertex- und Fragment-Shader so, dass eine einfache diffuse Beleuchtungsberechnung *pro Pixel* durchgeführt wird. Sie können die Farbe der Lichtquelle als (1.0, 1.0, 1.0, 1.0) annehmen und Abschwächung der Intensität mit der Distanz vernachlässigen. Ihnen stehen dazu die OpenGL-Vertex-Attribute `gl_Vertex`, `gl_Normal`, und außerdem die Konstante `gl_LightSource[0].position` (enthält die Position der Lichtquelle) und die Matrizen `gl_ModelViewMatrix` und `gl_NormalMatrix` zur Verfügung. (3 Punkte)

Vertex Shader

```
void main() {

    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;

}
```

Fragment Shader

```
void main() {

    gl_FragColor =
}
```

Name: ______________________________________ Matrikelnummer: ______________

Aufgabe 9: Bézier-Kurven (8 Punkte)

a) Gegeben sind die quadratischen Bernstein-Polynome $B_i^2(u) = \binom{2}{i} u^i (1-u)^{2-i}$:

$$\begin{aligned} B_0^2(u) &= 1 - 2u + u^2 \\ B_1^2(u) &= 2u - 2u^2 \\ B_2^2(u) &= u^2 \end{aligned}$$

Die quadratische Bézier-Kurve $F(u) = \sum_{i=0}^{2} B_i^2(u)\mathbf{b_i}$ soll durch einen Basiswechsel bezüglich der Monombasis ausgedrückt werden, d.h. in der Form $F_{monom}(u) = \sum_{i=0}^{2} \mathbf{a_i} u^i$ dargestellt werden. Geben Sie an, wie Sie die Koeffizienten $\mathbf{a_0}$, $\mathbf{a_1}$ und $\mathbf{a_2}$ in Abhängigkeit von $\mathbf{b_0}$, $\mathbf{b_1}$ und $\mathbf{b_2}$ bestimmen! (3 Punkte)

b) Was versteht man unter affiner Invarianz der Bézier-Repräsentation? (1 Punkt)

c) Gegeben sei eine kubische Bézier-Kurve $F(u) = \sum_{i=0}^{3} B_i^3(u)\mathbf{b_i}$ mit $\mathbf{b_0} = (0{,}0)$, $\mathbf{b_1} = (1{,}2)$, $\mathbf{b_2} = (3{,}1)$, $\mathbf{b_3} = (4{,}1)$ und ein weiterer Punkt $\mathbf{c_3} = (7{,}0)$. $B_i^3(u)$ bezeichnet das i-te Bernstein-Polynom vom Grad 3. Zeichnen Sie das Kontrollpolygon für eine zweite kubische Bézier-Kurve $G(v) = \sum_{i=0}^{3} B_i^3(v)\mathbf{c_i}$, die im Punkt $\mathbf{c_3}$ endet und C^2-stetig an $F(u)$ an der Stelle $u = 1$ anschließt! Geben Sie an, wie Sie die Kontrollpunkte $\mathbf{c_i}$ bestimmen! Skizzieren Sie $G(v)$ (nicht exakt auswerten oder zeichnen)! (4 Punkte)