

Hauptklausur Computergrafik

WS 2012/13

12. März 2013

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 19 Seiten (9 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 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	5	12	6	10	9	16	12	15	13	12	10	120

Note

Aufgabe 1: Raytracing (5 Punkte)

- a) Raytracing nach Whitted, wie Sie es in der Vorlesung kennengelernt haben, folgt den Gesetzen der geometrischen Optik. Ergänzen Sie die folgende Liste um die *3 weiteren* Strahltypen, die bei diesem Raytracing-Verfahren vorkommen! (3 Punkte)

1) Kamera- oder Primärstrahl

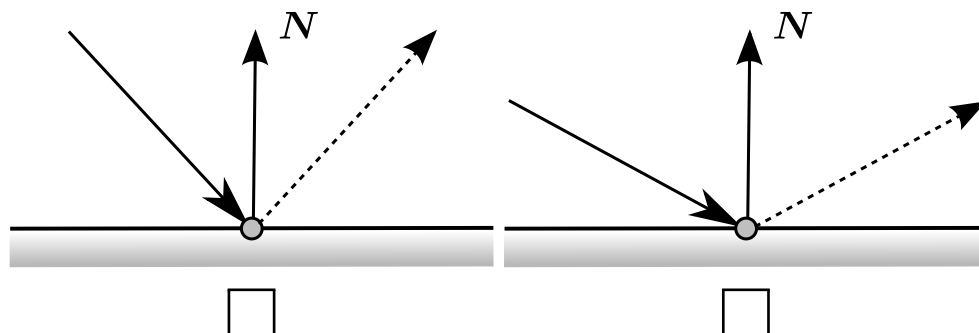
2)

3)

4)

- b) Die folgenden Skizzen zeigen zwei Lichtstrahlen mit unterschiedlichem Einfallswinkel die an einer *spekularen Glasoberfläche* reflektiert werden (der Vektor N ist die Oberflächennormale).

Kreuzen Sie den Fall an, in dem ein größerer Teil des einfallenden Lichts reflektiert wird! (1 Punkt)



- c) Wie nennt man das physikalische Gesetz oder Prinzip, welches den Zusammenhang zwischen Einfallswinkel und Reflektivität beschreibt? (1 Punkt)

Gesetz oder Prinzip	
Brewstersches Gesetz	<input type="checkbox"/>
Snellsches Gesetz	<input type="checkbox"/>
Fresnelsche Formeln	<input type="checkbox"/>
Fermatsches Prinzip	<input type="checkbox"/>

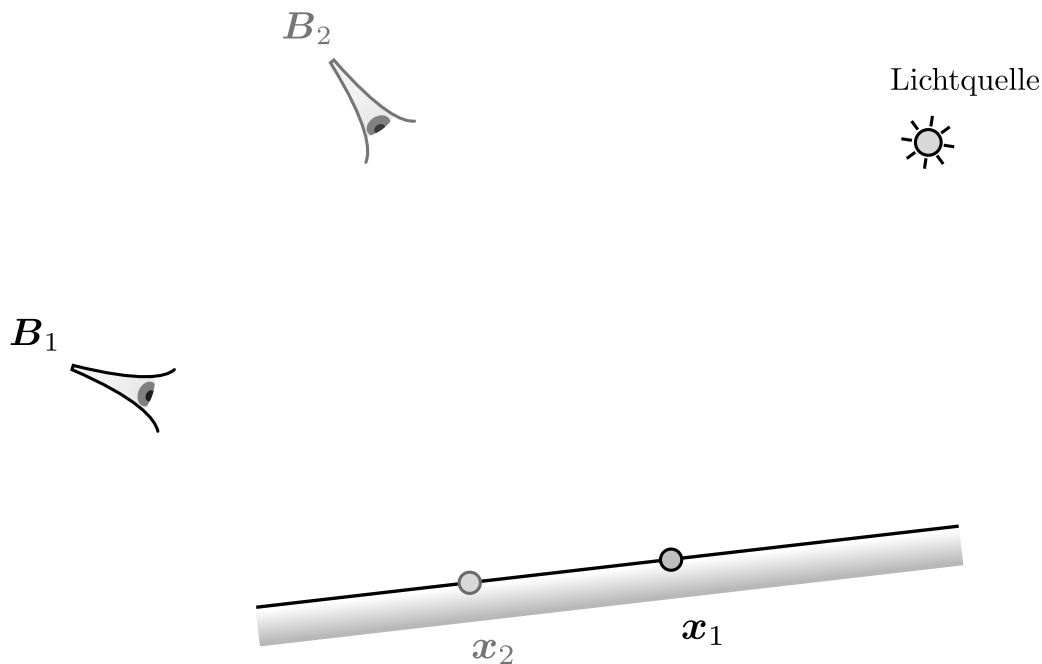
Name: _____

Matrikelnummer: _____

Aufgabe 2: Beleuchtung, Licht und Wahrnehmung (12 Punkte)

In dieser Aufgabe geht es zunächst um das *Phong-Beleuchtungsmodell*, das Sie aus der Vorlesung kennen. Es beinhaltet eine ambiente, diffuse und spekulare Komponente.

- a) Ergänzen Sie die Skizze und zeichnen Sie die 4 Vektoren ein, die im Phong-Beleuchtungsmodell für die Beleuchtungsberechnung benötigt werden! Verwenden Sie für die Skizze die Betrachterposition B_1 und den Oberflächenpunkt x_1 . **(2 Punkte)**



- b) Der Wert welcher Komponente(n) des Phong-Beleuchtungsmodells verändert bzw. verändern sich, wenn in der obigen Situation ... **(3 Punkte)**

i) ...der Punkt x_2 statt x_1 betrachtet wird?

ii) ...die Szene aus der Position B_2 statt B_1 betrachtet wird?

- c) In welcher Komponente taucht der sogenannte Phong-Exponent auf und welchen Einfluss hat er auf die Erscheinung einer Oberfläche? Wie ändert sich das Aussehen, wenn der Phong-Exponent größer gewählt wird? (3 Punkte)

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

Aussage	Wahr	Falsch
Zu drei gewählten Primärfarben gibt es immer Spektralfarben, die durch die Kombination dieser drei Farben <i>nicht</i> realisierbar sind.	<input type="checkbox"/>	<input type="checkbox"/>
Menschen können geringe Helligkeitsunterschiede im Bereich niedriger Lichtintensität <i>besser</i> wahrnehmen als im Bereich hoher Lichtintensität.	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt <i>keinen</i> linearen Zusammenhang zwischen dem CIE-XYZ- und dem RGB-Modell.	<input type="checkbox"/>	<input type="checkbox"/>
Gammakorrektur ist <i>nur</i> bei Röhrenmonitoren notwendig.	<input type="checkbox"/>	<input type="checkbox"/>

Name: _____

Matrikelnummer: _____

Aufgabe 3: Transformationen (6 Punkte)

- a) Gegeben sind Vektoren in homogenen Koordinaten. Kreuzen Sie jeweils an, ob es sich um einen Punkt oder eine Richtung handelt. Geben Sie außerdem die dazugehörigen kartesischen Koordinaten an. **(3 Punkte)**

Vektor	Punkt	Richtung	Kartesische Koordinaten
(1.0, 2.0, 3.0, 1.0)	<input type="checkbox"/>	<input type="checkbox"/>	
(1.0, 2.0, 3.0, 0.1)	<input type="checkbox"/>	<input type="checkbox"/>	
(1.0, 2.0, 3.0, 0.0)	<input type="checkbox"/>	<input type="checkbox"/>	

- b) Kreuzen Sie die richtigen Aussagen an! Es können auch mehrere Aussagen zutreffen. Die gegebenen Matrizen sind *keine* homogenen Matrizen. Die Koordinatensysteme sind rechtshändig. Für jeden vollständig richtigen Block erhalten Sie einen Punkt. **(3 Punkte)**

$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \in \mathbb{R}^{3 \times 3}$	<input type="checkbox"/> dreht die y -Achse in Richtung z -Achse <input type="checkbox"/> dreht die x -Achse in Richtung y -Achse <input type="checkbox"/> ist keine valide Rotationsmatrix
$\begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix} \in \mathbb{R}^{2 \times 2}$ beschreibt eine	<input type="checkbox"/> Scherung um Faktor a <input type="checkbox"/> Skalierung um Faktor a <input type="checkbox"/> Translation um a
Eine affine Abbildung	<input type="checkbox"/> ist teilverhältnistreu <input type="checkbox"/> ist winkelerhaltend <input type="checkbox"/> erhält die Parallelität von Linien

Aufgabe 4: Texturen und Texture-Mapping (10 Punkte)

- a) Was versteht man unter Mip-Mapping? Welches Problem beim Texture Mapping soll damit gelöst werden und wann tritt dieses Problem auf? Wie erzeugt man Mip-Maps? **(6 Punkte)**

- b) Was versteht man unter einer Environment Map? Nennen Sie eine Anwendung von Environment Mapping. Wie wird auf die Environment Map zugegriffen und welche vereinfachende Annahme wird dabei gemacht? **(4 Punkte)**

Name: _____

Matrikelnummer: _____

Aufgabe 5: Räumliche Datenstrukturen (9 Punkte)

- a) In dieser Aufgabe sollen Sie die zutreffenden Aussagen über die räumlichen Datenstrukturen, wie Sie sie in der Vorlesung kennengelernt haben, identifizieren. Kreuzen Sie dazu jeweils die passenden Kästchen an! Sie erhalten für jede vollständig richtige Zeile 2 Punkte. **(6 Punkte)**

Aussage	BVH	Octree	kD-Baum	Gitter
Die Struktur eignet sich gut in Fällen, in denen Primitive gehäuft auftreten und große Leerräume zwischen den Häufungen existieren.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
In einer Hierarchieebene können sich die Zellen der Struktur überlappen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gehen Sie nun davon aus, dass Primitive <i>nicht</i> unterteilt werden und <i>kein</i> Mailboxing verwendet wird. Dann wird jedes Primitiv in jedem Fall höchstens einmal auf einen Schnitt mit dem Strahl getestet.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

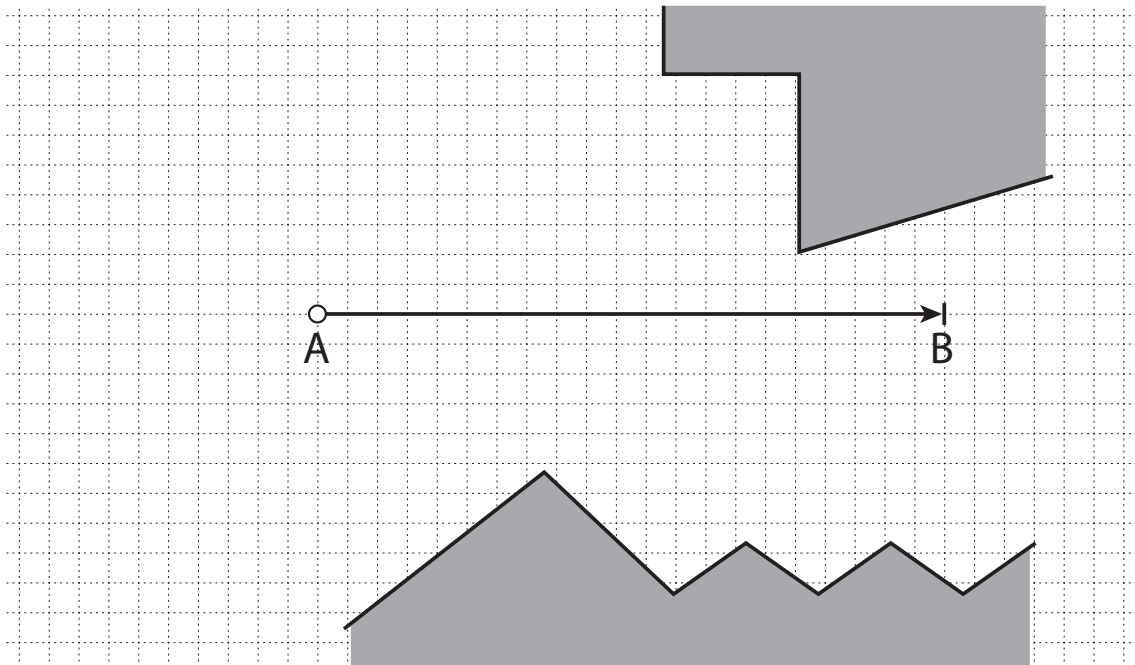
- b) Nummerieren Sie die folgenden 3 Verfahren **aufsteigend** nach dem **Aufwand** für den Aufbau der Datenstrukturen: **(3 Punkte)**

kd-Baum mit Surface Area Heuristic	
Gitter	
kd-Baum mit Object Median	

Aufgabe 6: Prozedurale Modellierung (16 Punkte)

a) Nennen Sie drei Vorteile (Stichpunkte) von prozeduralen Texturen! (3 Punkte)

b) Im folgenden Beispiel sollen Sie *Sphere-Tracing* für Distanzfelder illustrieren. Das Distanzfeld in dieser Szene beschreibt die grauen Körper. Der Strahl, der auf Schnitt getestet werden soll, beginnt am Punkt **A** in Richtung von Punkt **B**, wo er auch endet. Zeichnen Sie die Schritte entlang des Strahls ein! (3 Punkte)



Name: _____

Matrikelnummer: _____

- c) Sie sollen nun Sphere-Tracing in der OpenGL Shading Language programmieren. Dabei soll der nächste Schnittpunkt eines Strahls mit der Szenengeometrie gefunden werden. Als Abbruchkriterium für die Suche dient die Distanz `tMax`. Ein Schnittpunkt ist gefunden, wenn die Distanzfunktion einen kleineren Wert als `epsilon` liefert. In diesem Fall soll die Funktion `sphereTrace` den Wert `true` und den Schnittpunkt `pos` zurückliefern. Die Anzahl der Sphere-Tracing-Schritte wird immer in `steps` zurückgegeben. Ihnen steht die Distanzfunktion `float DF(vec3 x)` zur Verfügung. **(10 Punkte)**

```
in vec3  A;           // Ursprung des Strahls.
in vec3  D;           // Die normalisierte Richtung des Strahls.
in float tMax;        // Abbruchkriterium: maximale Suchdistanz.
uniform float epsilon; // Toleranz

// Distanzfunktion. Liefert den Abstand von x zur nächsten Fläche.
float DF( vec3 x ) { ... }

// Implementieren Sie Sphere Tracing in dieser Funktion.
bool sphereTrace( out vec3 pos, out int steps ) {

}

}
```

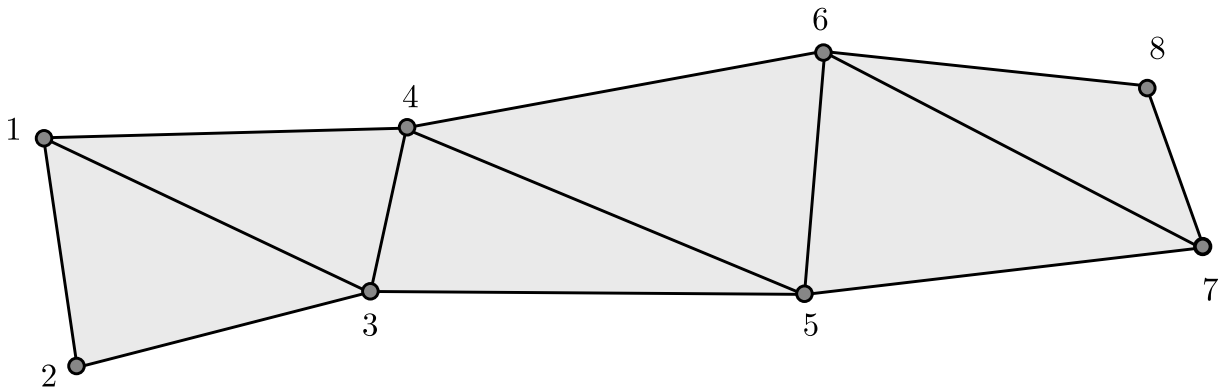
Aufgabe 7: OpenGL Primitive und Shader (12 Punkte)

- a) Das Dreiecksnetz in der Abbildung soll mit verschiedenen Primitivtypen gezeichnet werden. Sie sollen als Lösung Indexlisten angeben; für isolierte Dreiecke z.B. wie folgt:

p1: (1, 2, 3) p2: (1, 3, 4) ...

Eine *richtige Lösung* zeichnet das gesamte Dreiecksnetz und benötigt die *minimale* Anzahl von Primitiven. Wenn es mehrere richtige Lösungen gibt, müssen Sie nur eine angeben.

Back-Face-Culling spielt bei dieser Aufgabe *keine* Rolle. (6 Punkte)



- i) Geben Sie die Indexliste(n) für den Primitivtyp `GL_TRIANGLE_STRIP` an! Wie viele Primitive dieses Typs werden benötigt?
- ii) Geben Sie die Indexliste(n) für den Primitivtyp `GL_TRIANGLE_FAN` an! Wie viele Primitive dieses Type werden benötigt?

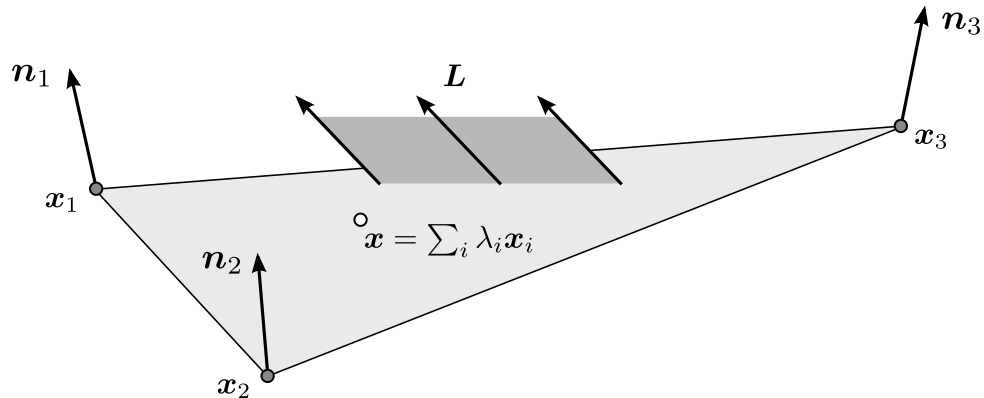
Name: _____

Matrikelnummer: _____

- b) In welchen OpenGL-Shader-Programmen kann man die folgenden Aufgaben durchführen? Kreuzen Sie jeweils die passenden Kästchen an. Sie erhalten für jede vollständig richtige Zeile einen Punkt. **(6 Punkte)**

Aufgabe	Vertex-Shader	Geometry-Shader	Fragment-Shader
Den Primitivtyp von <code>GL_POINT</code> auf <code>GL_TRIANGLE</code> ändern.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Projektionstransformation auf Vertices anwenden und das Ergebnis in <code>gl_Position</code> speichern.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Fragmentfarbe ausgeben.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>uniform</code> -Variablen schreiben.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Das Beleuchtungsmodell auswerten, wenn <i>Phong-Shading</i> verwendet wird.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aus Texturen lesen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Aufgabe 8: Schattierung und Interpolation (15 Punkte)



Die Abbildung zeigt ein Dreieck, das gezeichnet werden soll. Gegebene Größen sind

- Die Eckpunkte x_1 , x_2 und x_3 .
- Die Normalen an den Eckpunkten n_1 , n_2 und n_3 .
- Die Oberflächenfarben an den Eckpunkten c_1 , c_2 und c_3 .
- Die Lichtrichtung L . Die Lichtquelle ist direktional und weiß. Ihre Intensität ist 1.
- Der Punkt x hat die baryzentrischen Koordinaten λ_1 , λ_2 und λ_3 .

Geben Sie in Ihren Lösungen Formeln an!

a) Wie berechnet man die baryzentrische Koordinate λ_3 von Punkt x ? Sie können davon ausgehen, dass x im Dreieck x_1 , x_2 , x_3 liegt. Nutzen Sie die Notation $\Delta(A, B, C)$ für den Absolutwert der Fläche des Dreiecks mit den Eckpunkten A , B und C . **(2 Punkte)**

b) Nehmen Sie an, dass in einem OpenGL-Programm Gouraud-Shading implementiert ist und Sie dies nicht ändern können. Wie können Sie dennoch das Aussehen des Dreiecksnetzes an das Ergebnis von Phong-Shading annähern? **(1 Punkt)**

Name: _____

Matrikelnummer: _____

- c) Bei dieser Aufgabe können Sie $\langle \mathbf{a}, \mathbf{b} \rangle^+ \equiv \max(0, \mathbf{a}^T \mathbf{b})$ benutzen.

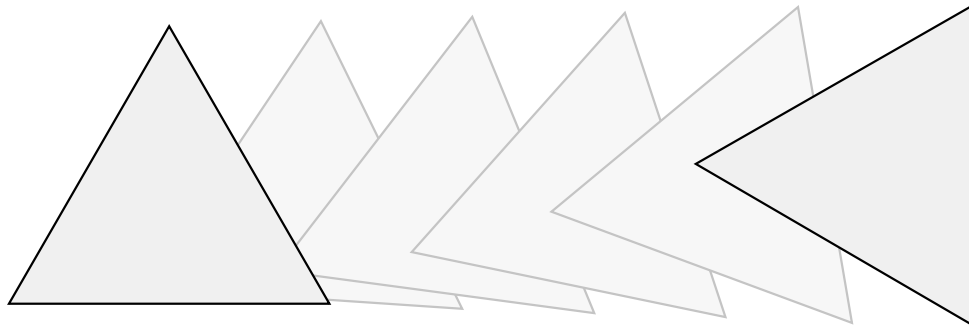
Wie ist die Farbe \mathbf{f} am Punkt $\mathbf{x} = \sum_i \lambda_i \mathbf{x}_i$, die sich durch **ausschließlich diffuse** Reflexion und interpolierter Oberflächenfarbe ergibt, wenn ...

- i) ... Flat-Shading eingesetzt wird? (**4 Punkte**)

- ii) ... Gouraud-Shading eingesetzt wird? (**4 Punkte**)

- iii) ... Phong-Shading eingesetzt wird? (**4 Punkte**)

Aufgabe 9: Keyframing mit der OpenGL Shading Language (13 Punkte)



$$t = t1 \dots\dots\dots t = t2$$

Das sogenannte *Keyframing* ist eine Technik für die Animation von 3D-Modellen. Dabei werden Transformationen zu diskreten Zeitpunkten gespeichert. Während der Wiedergabe wird zwischen den Vertex-Positionen, die sich aus den Transformationen ergeben, interpoliert, um weiche Übergänge zu erreichen.

Sie sollen in dieser Aufgabe Keyframing in einem Vertex-Shader implementieren. Gegeben sind drei Matrizen, $M1$, $M2$ und $M3$, die zu den Zeitpunkten $t1$, $t2$ und $t3$ gelten. Die Matrizen transformieren Vertices aus Objektkoordinaten in Weltkoordinaten. Das Ergebnis dieser Transformationen muss *nicht* dehomogenisiert werden. Der aktuelle Zeitpunkt wird in der Uniform-Variable t übergeben. Sie können annehmen, dass stets die folgenden Ungleichungen gelten:

$$t1 \leq t2 \leq t3$$

$$t1 \leq t \leq t3$$

(Fortsetzung nächste Seite)

Name: _____

Matrikelnummer: _____

Vervollständigen Sie den Shader, um Keyframing zu implementieren. Nutzen Sie einfache lineare Interpolation in *Weltkoordinaten*.

Schreiben Sie den transformierten Vertex in Clip-Koordinaten in `gl_Position`. **(13 Punkte)**

```
in vec4 p;           // Position des Vertex in Objektkoordinaten.
uniform float t;      // Aktueller Zeitpunkt.
uniform float t1;     // Die Zeitpunkte der drei Keyframes.
uniform float t2;
uniform float t3;
uniform mat4 M1;      // Die drei Transformationsmatrizen (Objekt->Welt).
uniform mat4 M2;
uniform mat4 M3;
uniform mat4 VP;      // Die View-Projection-Matrix.

void main() {
```

```
}
```

- a) Eine Szene mit opaken und semitransparenten Dreiecksnetzen soll *möglichst effizient* gezeichnet werden. Dabei soll ein nicht-kommutativer Blending-Operator verwendet und ein *möglichst korrektes Resultat* erzielt werden.

Ergänzen Sie das OpenGL Programm, indem Sie die notwendigen Befehle oder deren Nummer eintragen. Der Tiefentest ist bei Eintritt in diesen Programmteil angeschaltet; die anderen OpenGL-Zustände sind nicht definiert. **(6 Punkte)**

Die folgenden OpenGL-Kommandos und Subroutinen stehen Ihnen zur Verfügung (von denen Sie nicht alle benötigen!):

- (1) `glDepthMask(GL_TRUE);`
- (2) `glDepthMask(GL_FALSE);`
- (3) `glEnable(GL_DEPTH_TEST);`
- (4) `glDisable(GL_DEPTH_TEST);`
- (5) `glEnable(GL_BLEND);`
- (6) `glDisable(GL_BLEND);`
- (7) `drawTrans();` zeichnet alle transparenten Dreiecke in unbestimmter Reihenfolge.
- (8) `drawTransSorted();` sortiert alle transparenten Dreiecksnetze und zeichnet sie in der Reihenfolge von hinten nach vorne.
- (9) `drawOpaque();` zeichnet alle opaken Dreiecke in unbestimmter Reihenfolge.
- (10) `drawOpaqueSorted();` sortiert alle opaken Dreiecksnetze und zeichnet sie in der Reihenfolge von hinten nach vorne.

```
void renderScene() {  
  
    // Setup vor dem Löschen von Frame- und Tiefenpuffer  
  
  
  
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );  
  
    // Zeichnen der Szene ab hier  
  
  
  
}
```


Name: _____

Matrikelnummer: _____

- b) Ein texturiertes 3D-Modell einer Palme (Bild links/mitte) soll gezeichnet werden. Das Bild rechts zeigt den Alpha-Kanal der Textur (schwarz entspricht einem Alpha-Wert von 0, weiß einem Alpha-Wert von 1).



Das 3D-Modell wird nun mit Alpha-Blending (ohne Sortierung der Dreiecke) texturiert mit den folgenden OpenGL-Einstellungen gezeichnet:

```
glEnable( GL_BLEND );  
glBlendEquation( GL_FUNC_ADD );  
glBlendFunc( GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA );  
glEnable( GL_DEPTH_TEST );  
glDepthMask( GL_TRUE );
```

Beschreiben Sie kurz welche Probleme auftreten. **(3 Punkte)**

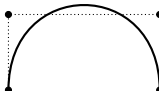
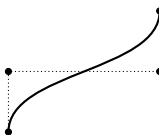
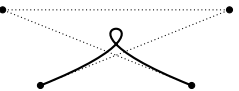
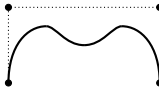
- c) Welche Technik bzw. Funktionalität setzt man in OpenGL üblicherweise ein, um die Probleme aus der vorherigen Aufgabe zu vermeiden? In welchem Typ von OpenGL-Shader kann diese Funktionalität implementiert werden und wie lautet der entsprechende Befehl? **(3 Punkte)**

Aufgabe 11: Bézierkurven (10 Punkte)

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

Aussage	Wahr	Falsch
Bézierkurven interpolieren alle Kontrollpunkte.	<input type="checkbox"/>	<input type="checkbox"/>
Eine Bézierkurve vom Grad $N + 1$ hat N Kontrollpunkte.	<input type="checkbox"/>	<input type="checkbox"/>
Die Bernstein-Polynome bilden eine Basis des \mathbb{R}^3 .	<input type="checkbox"/>	<input type="checkbox"/>
Die Ableitung einer Bézierkurve $dF(u)/du$ ist wieder eine Bézierkurve.	<input type="checkbox"/>	<input type="checkbox"/>

b) Geben Sie an, ob es sich bei den folgenden Kurven mit gegebenem Kontrollpolygon um Bézierkurven handelt. Begründen Sie jeweils kurz Ihre Antwort, falls es sich *nicht* um eine Bézierkurve handelt! (6 Punkte)

Kurve	Ja	Nein	Begründung
	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	