



1113

Klausurprüfung Computergrafik 2 WS 2011/2012, Gruppe 1

(von der/dem Studierenden auszufüllen)

Name:	
Matrikelnummer:	

(vom Dozenten auszufüllen)

Aufgabe	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Punkte																	

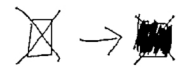
(vom Dozenten auszufüllen)

Aufgabe	18	19	20	21	22	23								Summe	Note
Punkte	4	0	2	3										6	1,0

u.g.

Bitte beachten Sie unbedingt folgende Hinweise:

- Prüfen Sie die Vollständigkeit der Unterlagen:
 - 10 Blätter (inkl. Deckblatt) mit insgesamt 23 Aufgaben.
- Füllen Sie das Deckblatt aus, indem Sie Ihren Namen und Ihre Matrikelnummer in die dafür vorgesehenen Felder schreiben.
- Sollte eines Ihrer Blätter vom Deckblatt getrennt werden, versehen Sie es unbedingt mit Ihrem Namen und Ihrer Matrikelnummer. Wenn Blätter nicht zugeordnet werden können, werden fehlende Blätter mit 0 Punkten bewertet.
- Für Multiple-Choice-Aufgaben können jeweils keine, eine oder mehrere korrekte Antworten aufgeführt sein.
- Tragen Sie bei Aufgaben mit mehreren auszufüllenden Feldern (_____) in *jedes* Feld eine Antwort ein.
- Außer einem Stift sind keine Hilfsmittel zugelassen.
- Sie können die Rückseiten der Blätter für Nebenrechnungen nutzen.
- Falls Sie ein Kästchen versehentlich angekreuzt haben, schwärzen Sie vollständig, um es als „nicht angekreuzt“ zu markieren.





1. Skalarprodukt

5 Punkte 2,5 P

Wie ist das Skalarprodukt zweier Vektoren $S=(x,y,z)^T$ und $S'=(x',y',z')^T$ in einem kartesischen Koordinatensystem definiert?

$(x-x')(y-y')(z-z')$

$x \cdot x' + y \cdot y' + z \cdot z'$

$|S| |S'| \cos(\angle S, S')$

2. Kreuzprodukt

5 Punkte

Was sind Eigenschaften des Kreuzprodukts $R = S \times T$ zweier Vektoren S und T ?

1 P

R ist orthogonal zu S

R ist orthogonal zu T

$|R|$ ist die Fläche des von S und T aufgespannten Dreiecks

$|R|$ ist die Fläche des von S und T aufgespannten Parallelogramms

(S, T, R) ist ein rechtshändiges Koordinatensystem

(S, T, R) ist ein linkshändiges Koordinatensystem

3. Vertex-Normale

5 Punkte 4 P

Welche der folgenden Aussagen über 3D-Normalenvektoren sind richtig?

Eine Normale steht senkrecht auf der zu modellierenden Oberfläche.

Die Summe der Komponenten eines Normalenvektors ist immer 1.

Für die Beleuchtungsberechnung geschlossener Oberflächen wird in der Regel die nach außen zeigende Normale verwendet.

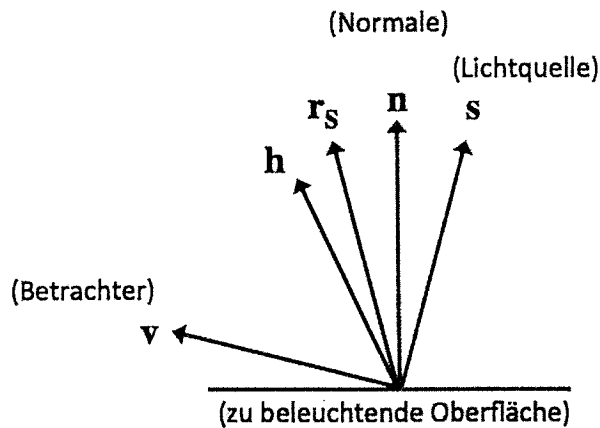


4. Beleuchtungsmodelle nach Phong und Blinn-Phong

5 Punkte

OP

Das folgende Diagramm zeigt schematisch die Vektoren, die bei der lokalen Beleuchtungsberechnung nach Phong und Blinn-Phong eine Rolle spielen.



r_s : Spiegelung von s an n

h : Winkelhalbierende zwischen s und v

Die Beleuchtungsmodelle bestehen jeweils aus einem ambienten, einem diffusen und einem spekularen Term. Welcher Winkel spielt jeweils eine Rolle? Ordnen Sie im folgenden jedem genannten Term die entsprechenden beiden Vektoren zu, die den für den Term relevanten Winkel definieren. Falls für einen Term keiner der Vektoren eine Rolle spielt, tragen Sie einen Strich („—“) ein.

Ambienter Term:

Winkel zwischen

s und v

Diffuser Term:

Winkel zwischen

s und n

Spekularer Term (Phong):

Winkel zwischen

n und r_s

Spekularer Term (Blinn-Phong):

Winkel zwischen

h und n

5. Refraktion

5 Punkte

OP

Welche der folgenden Aussagen sind richtig?

Der Brechungsindex n eines Mediums beschreibt seine optische Dichte im Vergleich zu der von Glas.

Die Schlick-Approximation der Fresnel-Gleichung beschreibt das Verhältnis von Brechungswinkel zu Einfallswinkel.

Die Fresnel-Gleichungen beschreiben das Verhältnis von reflektierter und transmittierter Lichtmenge.



6. Raytracing – Schattenstrahlen

5 Punkte ✓

Welche Aussagen über Schattenstrahlen beim Raytracing sind falsch?

- Schattenstrahlen werden benötigt, um zu testen, ob eine Lichtquelle vor oder hinter der zu beleuchtenden Oberfläche liegt.
- Um weiche Schattenkanten bezüglich einer Punktlichtquelle zu erhalten, müssen mehrere Schattenstrahlen gesendet werden.
- Schattenstrahlen stehen senkrecht auf der zu beleuchtenden Oberfläche.

7. Raytracing – Schnittpunkte und Verdeckung

5 Punkte ✓

Ihr Raytracer sendet einen Primärstrahl aus der virtuellen Lochkamera durch einen Pixel auf der virtuellen Bildebene in die Szene und berechnet eine Menge $\{ t_i \}$ von Schnittpunkten mit Objekten in der Szene. Welcher der zurückgelieferten Schnittpunkte ist derjenige, der die Pixelfarbe bestimmt?

- Der Schnittpunkt mit dem kleinsten positiven t_i
- Der Schnittpunkt mit dem größten negativen t_i
- Der kleinste Schnittpunkt mit $-\varepsilon \leq t_i \leq \varepsilon$ (wobei ε ein kleiner Toleranzwert ist)

8. Transformationsmatrizen

5 Punkte ✓

Welche Typen von 3D-Transformationen können nicht mit einer 3x3-Matrix dargestellt werden?

- Rotation
- Translation
- Skalierung
- Scherung



9. Homogene Koordinaten

5 Punkte 4P

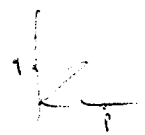
Punkte und Richtungsvektoren können mittels homogener Koordinaten mit vier Komponenten $(x, y, z, w)^T$ dargestellt werden. Welche der folgenden Aussagen sind falsch?

- Bei einem Punkt ist $w=0$.
- Bei einem Richtungsvektor ist $w=1$.
- ? Die Subtraktion von zwei Punkten ist nicht erlaubt.
- Die Addition von zwei Richtungsvektoren ergibt wieder eine Richtung.
- ✗ Die Addition von zwei Punkten ergibt wieder einen Punkt.
- Die Addition eines Punktes mit einer Richtung ergibt einen Punkt.
- Bei der Multiplikation $V' = M V$ einer Rotations- oder Skalierungsmatrix M mit einem homogenen Vektor V bleibt die w -Koordinate gleich: von $V'_w = V_w$.

10. Transformation von Vektoren

5 Punkte 3P

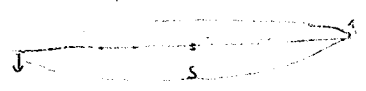
Gegeben sei eine 4×4 Model-View-Matrix M , die z.B. von Modellkoordinaten in Kamerakoordinaten transformiert. Dabei sind Rotation, Translation und beliebige Skalierungen (außer mit 0) in allen Dimensionen erlaubt. Welche der folgenden Aussagen sind richtig?

- Gegeben seien zwei Punkte p und q sowie der Richtungsvektor $v = q - p$. Dann gilt $Mv = M(q - p)$. 
- Bei der Transformation mit M bleiben Winkel zwischen Richtungsvektoren erhalten.
- ✗ Normalen dürfen nicht mit M transformiert werden, da nicht sichergestellt ist, dass sie aus Sicht der Kamera noch senkrecht auf der jeweiligen Oberfläche stehen.

11. Zusammengesetzte Transformationen

5 Punkte 0P

Eine Form bestehe aus einer Menge von Vertices mit Positionen X_i , die durch einen Linienzug verbunden werden. Wie kann ich die Form um ihren Schwerpunkt S rotieren?

- ✗ Jeden Vertex X_i zuerst rotieren und dann um S verschieben 
- ✗ Jeden Vertex um $-S$ verschieben, dann rotieren, dann um $+S$ verschieben
- ? Den Vertex, der am nächsten an S ist, in den Nullpunkt verschieben, und dann alle Vertices rotieren.



12. Klassische vs. programmierbare OpenGL-Grafikpipeline

5 Punkte

ØP

Sie haben in der Vorlesung die Unterschiede zwischen der klassischen Fixed-Function-Pipeline (FFP) und der programmierbaren Pipeline (PP) kennengelernt. Welche der folgenden Aussagen sind richtig?

- Die PP erlaubt es, verbesserte Rasterisierungs-Algorithmen zu implementieren.
- Die PP erlaubt flexibleren Umgang mit dem Inhalt von Texturen und der Kombination mehrerer Texturen.
- Phong-Beleuchtung ist in der FFP bereits eingebaut.
- Die FFP unterstützt nur Vertex- und Fragment-Shader; die PP bietet des weiteren noch Geometry Shader und Tessellation Shader.

im Folien

13. Die Pipeline-Stufen von OpenGL ES und WebGL

5 Punkte

ØP

Im folgenden sind die Pipeline-Stufen von OpenGL ES 2.0 / WebGL sowie einige nicht vorhandene Stufen aufgelistet. Versehen Sie nur die „echten“ Stufen, die tatsächlich in OpenGL ES 2.0 / WebGL realisiert sind, mit von 1 an aufsteigenden Nummern, um sie in die richtige Reihenfolge zu bringen. Markieren Sie die Stufen, die nicht zur WebGL-Pipeline gehören, mit einem Strich („-“).

- | | |
|---------------------------|------------------------------------|
| <u>1</u> Rasterisierung | <u>3</u> Primitive Assembly |
| <u>2</u> Beleuchtung | <u>5</u> Model-View Transformation |
| <u>-</u> Vertex Shading | <u>-</u> Compute Shading |
| <u>4</u> Vertex Assembly | <u>6</u> Clipping |
| <u>7</u> Texturing | <u>-</u> Framebuffer-Operationen |
| <u>8</u> Fragment-Shading | |

in Folien

Vert. Ass → Vert. Sha → Prim Ass → Clipp → Rasteri → Pre Fragment →
Framebuffer



14. Datenfluss in der programmierbaren Pipeline

5 Punkte

0P

In der Vorlesung haben Sie die drei Shader-Variablen-Typen **attribute**, **varying** und **uniform** kennengelernt. Bitte setzen Sie in die nachfolgenden Aussagen jeweils den Typ ein, auf den die Aussage zutrifft. Sollte kein Typ zutreffen, tragen Sie einen Strich („-“) ein.

Variablen vom Typ varying dienen der Ausgabe aus dem Vertex-Shader.

Eine Variable vom Typ uniform enthält während der gesamten Ausführung eines drawArrays()-Befehls einen konstanten Wert.

Daten, die für jedes Fragment durch Interpolation aus den umgebenden Vertex-Daten berechnet werden, stehen in varying - Variablen.

Texturen werden als uniform - Variablen repräsentiert.

Um Daten zu definieren, die sich für jeden Vertex eines Modells potentiell ändern, verwendet man eine attribute - Variable.

Texturkoordinaten werden zumeist als varying - Variablen repräsentiert.

15. Fragmente und Fragment-Shader

5 Punkte

4P

Welche der folgenden Aussagen sind richtig?

- Einem Pixel im Framebuffer kann über ein Fragment eine Farbe zugewiesen werden.
- Mehrere Fragmente können auf dasselbe Pixel abgebildet werden.
- Im Fragment-Shader-Code kann auf die Vertex-Attribute der Vertices zugegriffen werden, die das Dreieck bilden, in welchem das Fragment liegt.

16. WebGL Grundaufbau und Einbettung

5 Punkte

0P

Welche der folgenden Aussagen sind richtig?

- WebGL ist über das `<frustum>`-Element in HTML 5 eingebettet.
- Eine WebGL-Applikation muss stets einen Vertex- und einen Fragment-Shader spezifizieren.
- WebGL ermöglicht es, unabhängig in mehrere Bereiche in derselben HTML-Seite zu rendern.
- Zu einem HTML-`<canvas>`-Objekt muß ein WebGL-Kontext-Objekt erzeugt werden, um mittels WebGL in dieses `<canvas>`-Objekt zu zeichnen.



17. drawArrays() und geometrische Primitive

5 Punkte

2P

WebGL unterstützt beim Aufruf von `drawArrays()` verschiedene geometrische Primitiven wie z.B. `TRIANGLES` oder `TRIANGLE_FAN`. Bitte tragen Sie in die folgenden Sätze die richtigen Zahlen ein:

Ein Aufruf von `drawArrays()` mit Typ `TRIANGLES` und 24 Vertices führt zur Erzeugung von 8 Dreiecken.

Ein Aufruf von `drawArrays()` mit Typ `TRIANGLE_FAN` und 24 Vertices führt zur Erzeugung von 22 Dreiecken.

Um einen Würfel zu malen, müssen 12 Dreiecke erzeugt werden.

Um ein Rechteck zu malen, muss `drawArrays()` mit einem Array von 6 Vertices aufgerufen werden.

18. WebGL Zustandsverwaltung

5 Punkte

4P

Welche der folgenden Aussagen über WebGL sind richtig?

WebGL stellt einen Szenengraphen zu Verfügung, der die hierarchische Modellierung zusammengesetzter Transformationen erlaubt.

Nach einem `drawArrays()`-Befehl muss die Applikation die Werte aller Variablen, die der Shader verwendet, entsprechend setzen.

Nach einem `drawArrays()`-Befehl verlieren die Shader-Variablen ihre Gültigkeit und müssen neu gesetzt werden.

Ein `drawArrays()`-Befehl verlässt sich darauf, dass alle Shader-Variablen bereits vor seinem Aufruf von der Applikation initialisiert wurden.





19. Vertex vs. Fragment Shader

5 Punkte

0P

Betrachten Sie die folgenden Codezeilen: in welchem Shader-Typ dürfen diese Zeilen jeweils erscheinen? Markieren Sie eine Zeile mit „V“, wenn sie in einem Vertex-Shader gültig ist, mit „F“, wenn sie in einem Fragment-Shader gültig ist, mit „VF“ oder „FV“ wenn in beiden, und mit „—“ wenn in keinem der beiden. (Annahme: die Variablen M und p wurden zuvor entsprechend deklariert).

(VF) uniform sampler2D dayTex;

(VF) uniform mat4 modelViewMatrix;

~~V~~ 0 gl_Position = M * p;

vif0 attribute vec3 normal;

~~V~~ gl_FragColor = vec4(1, 0, 0, 1);

(VF) varying vec3 color;

20. Vertex-Beleuchtung vs. Fragment-Beleuchtung

5 Punkte

2P

Was sind die Unterschiede zwischen Vertex-Beleuchtung und Fragment-Beleuchtung nach dem Phong-Modell?

Bei der Vertex-Beleuchtung erhält jedes Fragment innerhalb eines Dreiecks die gleiche Farbe.

Bei der Vertex-Beleuchtung wird die Phong-Funktion für jeden Vertex einzeln berechnet.

Bei der Fragment-Beleuchtung wird der Farbwert durch Interpolation aus den umgebenden Vertices interpoliert.

Bei der Fragment-Beleuchtung wird die Normalenrichtung für jedes Fragment durch Interpolation berechnet.



21. Texturfilterung

5 Punkte

3 P

Welche der folgenden Aussagen sind richtig ?

- Bei Texturvergrößerung (Magnification) ist ein Pixel größer als die Projektion eines Texels auf das Pixel.
- Bei Texturverkleinerung (Minification) ist ein auf das Pixel projizierte Texel kleiner als das Pixel.
- WebGL unterstützt für Textur-Vergrößerung (Magnification) u.a. die Modi *Nearest Neighbour*, *Linear Interpolation* und *Mip-Map*.
- Durch eine Mip-Map wird die Darstellung der Textur schärfer.
- Das „MIP“ in Mip-Map steht für *Maximum Intensity Projection* und berechnet den maximalen Luminanz-Wert derjenigen Texel, die auf ein Pixel projiziert werden.

22. Shadow Mapping und Shadow Volumes

5 Punkte

4 P

Welche der folgenden Aussagen sind richtig?

- Bei Shadow Mapping werden für jedes Blocker-Dreieck mehrere Schatten-Polygone unter Verwendung des Stencil Buffers gezeichnet.
- Eine Shadow Map ist eine Tiefenkarte der Szene aus Sicht der Lichtquelle.
- Wenn sich eine Lichtquelle oder ein geometrisches Objekt in der Szene bewegen, muss auch die Shadow Map neu berechnet werden.
- Für eine statische Szene müssen Shadow Maps nur einmal berechnet werden und können für jede zu erzeugende Ansicht wiederverwendet werden.

23. Stencil Buffer

5 Punkte

0 P

Welche Art von Daten wird in einem Stencil Buffer gespeichert?

- Tiefenwerte für Schatten
- Transparenz-Werte
- Bilddaten für Multi-Pass-Verfahren
- Floating-Point-Ergebnisse
- Pixelweise Zähler
- Sprite-Daten für Stenciling