

## Aufgabe 1: Raytracing

### Teilaufgabe 1a

*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!*

- (1) Primärstrahlen (2) Reflektionsstrahlen (rekursiv) (3) Transmissionsstrahlen (rekursiv)  
(4) Schattenstrahlen

### Teilaufgabe 1b

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

In Bild 2, da dort der Winkel des Strahls auf die Oberfläche flacher ist.

### Teilaufgabe 1c

*Wie nennt man das physikalische Gesetz oder Prinzip, welches den Zusammenhang zwischen Einfallswinkel und Reflektivität beschreibt?*

Snelliussches Brechungsgesetz. Es lautet

$$n_1 \cdot \sin(\theta_1) = n_2 \cdot \sin(\theta_2)$$

wobei die Winkel von der Oberflächennormale aus gemessen werden.  $n_1, n_2$  sind Materialkonstanten.



Vektor	Punkt	Richtung	Kartesische Koordinaten
$(1, 2, 3, 1)$	<input checked="" type="checkbox"/>	<input type="checkbox"/>	$(1, 2, 3)$
$(1, 2, 3, 0.1)$	<input checked="" type="checkbox"/>	<input type="checkbox"/>	$(1, 2, 30)$
$(1, 2, 3, 0)$	<input type="checkbox"/>	<input checked="" type="checkbox"/>	$(1, 2, 3)$

### Teilaufgabe 2c

*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?*

Diffuse Komponente

Ein großes  $n$  führt dazu, dass **Glanzlichter** kleiner, aber intensiver werden. Die Reflektion wird „perfekter“.

### Teilaufgabe 2d

#	Aussage	Wahr	Falsch
1	Zu drei gewählten Primärfarben gibt es immer Spektralfarben, die durch die Kombination dieser drei Farben nicht realisierbar sind.	x	
2	Menschen können geringe Helligkeitsunterschiede im Bereich niedriger Lichtintensität besser wahrnehmen als im Bereich hoher Lichtintensität.	x	
3	Es gibt keinen linearen Zusammenhang zwischen dem CIE-XYZ- und dem RGB-Modell.		x
4	Gammakorrektur ist nur bei Röhrenmonitoren notwendig.		x

## Aufgabe 3: Transformationen

### Teilaufgabe 3a

*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.*

Ein Punkt hat als letzte Komponente einen Wert  $\neq 0$ , eine Richtung hat dort  $= 0$ .

### Teilaufgabe 3b

Korrekt sind:

1. dreht die x-Achse in Richtung y-Achse

2. Scherung um Faktor  $a$
3. erhält die Parallelität von Linien (bei den anderen beiden ist die Scherung ein Gegenbeispiel)

## **Aufgabe 4**

### **Teilaufgabe 4a**

TODO

### **Teilaufgabe 4b**

TODO

## **Aufgabe 5**

### **Teilaufgabe 5a**

TODO

### **Teilaufgabe 5a**

TODO

## **Aufgabe 6**

### **Teilaufgabe 6a**

TODO

### **Teilaufgabe 6b**

TODO

## Teilaufgabe 6c

---

```
1  in vec3 A; // Ursprung des Strahls.
2  in vec3 D; // Die normalisierte Richtung des Strahls.
3  in float tMax; // Abbruchkriterium: maximale Suchdistanz.
4  uniform float epsilon; // Toleranz
5
6  // Distanzfunktion. Liefert den Abstand von x zur nächsten Fläche.
7  float DF( vec3 x ) { ... }
8
9  // Implementieren Sie Sphere Tracing in dieser Funktion.
10 bool sphereTrace( out vec3 pos, out int steps ) {
11     pos = A;
12     steps = 0;
13     float t = 0.;
14     while (t < tMax) {
15         float d = DF(pos);
16         pos += d * D;
17         if (abs(d) < epsilon) {
18             return true;
19         }
20     }
21     return false;
22 }
```

---

## Aufgabe 7

### Teilaufgabe 7a

TODO

### Teilaufgabe 7b

TODO

## Aufgabe 8

### Teilaufgabe 8a

TODO

### Teilaufgabe 8b

TODO

### Teilaufgabe 8c

TODO

## Aufgabe 9

---

```
1  in vec4 p; // Position des Vertex in Objektkoordinaten.
2  uniform float t; // Aktueller Zeitpunkt.
3  uniform float t1; // Die Zeitpunkte der drei Keyframes.
4  uniform float t2;
5  uniform float t3;
6  uniform mat4 M1; // Die drei Transformationsmatrizen (Objekt->Welt).
7  uniform mat4 M2;
8  uniform mat4 M3;
9  uniform mat4 VP; // Die View-Projection-Matrix.
10
11 void main() {
12     vec4 pWorld;
13     if (t < t2) {
14         pWorld = mix(M1 * p, M2 * p, (t - t1) / (t2 - t1));
15     } else {
16         pWorld = mix(M2 * p, M3 * p, (t - t2) / (t3 - t2));
17     }
18
19     gl_Position = VP * pWorld;
20 }
```

---

## Aufgabe 10

### Teilaufgabe 10a

---

```
1 void renderScene() {  
2     // Setup vor dem Löschen von Frame- und Tiefenpuffer  
3     glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );  
4     // Zeichnen der Szene ab hier  
5  
6     //TODO  
7 }
```

---

### Teilaufgabe 10b

TODO

### Teilaufgabe 10c

TODO

## Aufgabe 11: Bézierkurven

### Teilaufgabe 11a

TODO

### Teilaufgabe 11b

TODO