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

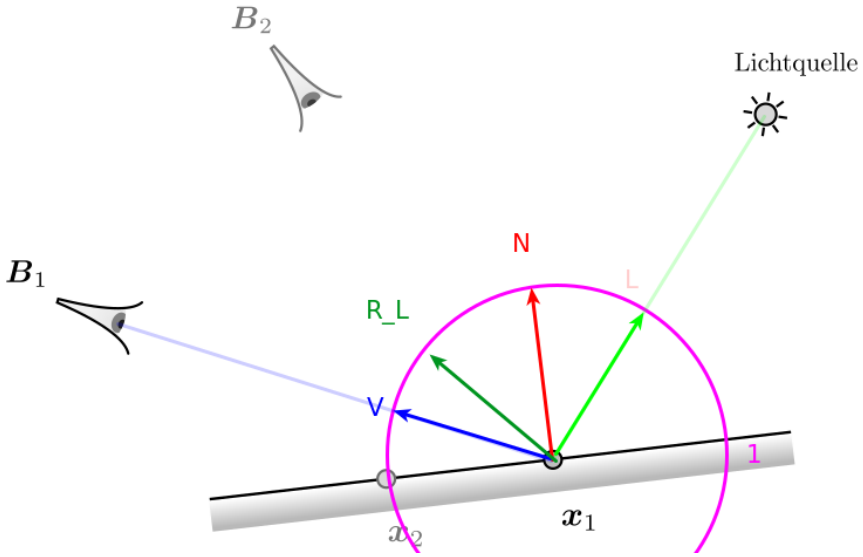
Wie nennt man das physikalische Gesetz oder Prinzip, welches den Zusammenhang zwischen Einfallswinkel und Reflektivität beschreibt?

Die **Fresnelsche Formeln** beschreiben die Beziehung zwischen Transmission und Reflexion in Abhängigkeit des Winkels.

Aufgabe 2: Beleuchtung, Licht und Wahrnehmung

Teilaufgabe 2a

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



Die 4 Vektoren sind:

- View-Vektor V
- Normale N,
- Licht-Vektor L und
- Reflektionsvektor R_L

$$I = \underbrace{k_a \cdot I_L}_{\text{ambient}} + \underbrace{k_d \cdot I_L \cdot (N \cdot L)}_{\text{diffus}} + \underbrace{k_s \cdot I_L (R_L \cdot V)^n}_{\text{spekular}}$$

Teilaufgabe 2b

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

(i) *... der Punkt x_2 statt x_1 betrachtet wird?*
L, R_L, V: Spekular und diffus

(ii) *... die Szene aus der Position B_2 statt B_1 betrachtet wird?*
V: Spekular (Glanzlichter sind abhängig vom Winkel zwischen V und R_L)

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?

Spekulare 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 zwischem 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 $= 0$.

Vektor	Punkt	Richtung	Kartesische Koordinaten
$(1, 2, 3, 1)$	☑	☐	$(1, 2, 3)$
$(1, 2, 3, 0.1)$	☑	☐	$(10, 20, 30)$
$(1, 2, 3, 0)$	☐	☑	$(1, 2, 3)$

Teilaufgabe 3b

Korrekt sind:

1. dreht die y-Achse in Richtung z-Achse
2. Scherung um Faktor a
3. ist teilverhältnistreu; erhält die Parallelität von Linien

Aufgabe 4: Texturen und Texture-Mapping

Teilaufgabe 4a

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?

Mip-Mapping ist eine Vorverarbeitung der Textur, um das **Aliasing**-Problem bei **Minification** zu behandeln.

Es werden kleiner skalierte, vorverarbeitete Versionen der Textur erstellt (Stufe i: um Faktor 2^i pro Achse kleiner).

Teilaufgabe 4b

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?

Was: Eine Environment-Map ist eine Textur zur Darstellung der Umgebung.

Anwendung: Durch eine Environment-Map kann die Reflektion/Beleuchtung eines Objekts bestimmt werden, ohne aufwendiges Ray-Tracing zu betreiben.

Konkret: Spiegelungen auf „flüssigem“ Terminator (vgl. Folien)

Annahme: Die Umgebung ist weit genug entfernt, sodass die Position keine Rolle spielt. Es wird nur die Blickrichtung verwendet. Man bestimmt für Cube-Maps also die Richtung (die Fläche des Würfels) und greift nur entsprechend dieser auf die Textur der betreffenden Würfelfläche zu.

Aufgabe 5: Räumliche Datenstrukturen

Teilaufgabe 5a

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

Teilaufgabe 5b

- Am wenigsten aufwendig (1): Gitter
- (2): kd-Baum mit Objekt Median
- (3): kd-Baum mit Surface Area Heuristic

Aufgabe 6: Prozedurale Modellierung

Teilaufgabe 6a

Nennen Sie drei Vorteile (Stichpunkte) von prozeduralen Texturen!

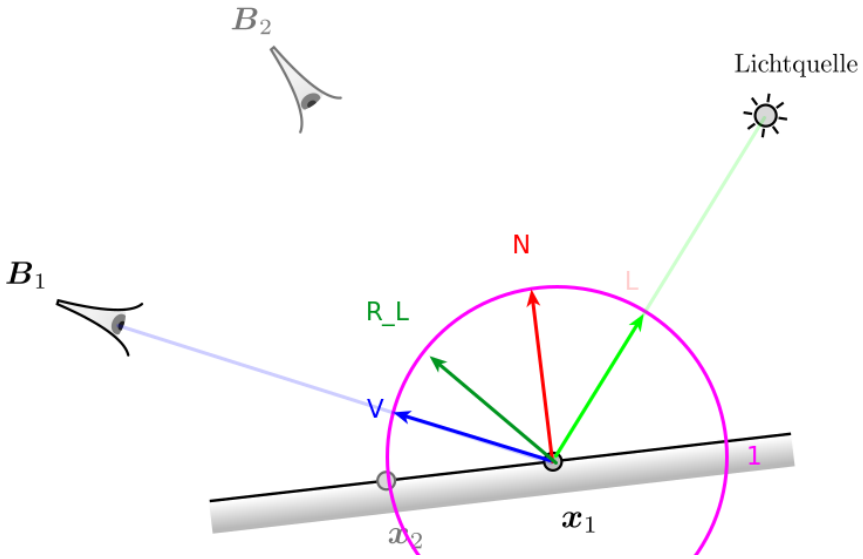
(1) Geringerer Speicheraufwand
(2) Natürliche Strukturen lassen sich leicht beschreiben
(3) Beliebige Vergrößerungen sind möglich (TODO: War das so gewünscht?)

Nachteile:

- Hoher Rechenaufwand

Teilaufgabe 6b

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!*



TODO: Einzeichnen

Teilaufgabe 6c

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.*

spheretracing.frag

```
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 ) {
    pos = A;
    steps = 0;
    float t = 0.;
    while (t < tMax) {
        float d = DF(pos);
        pos += d * D;
        if (abs(d) < epsilon) {
            return true;
        }
    }
    return false;
}
```

Aufgabe 7

Teilaufgabe 7a

i) Indexliste(n) für Primitivtyp `GL_TRIANGLE_STRIP`

p1: (2, 1, 3, 4, 5, 6, 7, 8)

benötigt 1 Primitiv

ii) Indexliste(n) für Primitivtyp `GL_TRIANGLE_FAN`

p1: (3, 2, 1, 4, 5)

p2: (6, 4, 5, 7, 8)

benötigt 2 Primitive

Teilaufgabe 7b

#	Aufgabe	Vertex-Shader	Geometry-Shader	Fragment-Shader
1	Den Primitivtyp von `GL_POINT` auf `GL_TRIANGLE` ändern.		x	
2	Die Projektionstransformation auf Vertizes anwenden und das Ergebnis in `gl_Position` speichern.	x		
3	Die Fragmentfarbe ausgeben.			x
4	`uniform`-Variablen schreiben.			
5	Das Beleuchtungsmodell auswerten, wenn *Phong-Shading* verwendet wird.	x		x
6	Aus Texturen lesen.	x	x	x

Aufgabe 8

Teilaufgabe 8a

$\lambda_3 = \triangle(x, x_1, x_2)$

Teilaufgabe 8b

Dreiecksnetz weiter unterteilen (Tesselieren)

Teilaufgabe 8c

(i) Flat shading

$$n = (x_1 - x_2) \times (x_3 - x_2) \tag{1}$$
$$f = \langle n, L \rangle^+ \cdot (\lambda_1 c_1 + \lambda_2 c_2 + \lambda_3 c_3) \tag{2}$$

(ii) Gouraud-Shading

$$c_i' = c_i \langle n_i, L \rangle^+ \tag{3}$$
$$f = \lambda_1 c_1' + \lambda_2 c_2' + \lambda_3 c_3' \tag{4}$$

(iii) Phong-Shading

$$n = \lambda_1 n_1 + \lambda_2 n_2 + \lambda_3 n_3 \tag{5}$$
$$f = \langle n, L \rangle^+ \cdot (\lambda_1 c_1 + \lambda_2 c_2 + \lambda_3 c_3) \tag{6}$$

Aufgabe 9

keyframing.vert

```
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() {
    vec4 pWorld;
    if (t < t2) {
        pWorld = mix(M1 * p, M2 * p, (t - t1) / (t2 - t1));
    } else {
        pWorld = mix(M2 * p, M3 * p, (t - t2) / (t3 - t2));
    }

    gl_Position = VP * pWorld;
}
```

Aufgabe 10

Teilaufgabe 10a

Kommentar: "glDepthMask (GL_TRUE)" nicht benötigt, da bereits gesetzt.

shader.frag

```
void renderScene() {
    // Setup vor dem Löschen von Frame- und Tiefenpuffer
    // <solution>
    glDepthMask( GL_TRUE ); // (1)
    // </solution>
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    // Zeichnen der Szene ab hier
    // <solution>
    drawOpaque(); // (9)
    glDisable( GL_DEPTH_TEST ); // (4)
    glEnable( GL_BLEND ); // (5)
      drawTransSorted(); // (8)
    glDisable( GL_BLEND ); // (6)
    glDepthMask ( GL_FALSE ); // (2) (ullt nicht benötigt)
    // </solution>
}
```

Teilaufgabe 10b

Da die Dreiecke nicht sortiert wurden, werden überlappende (semi-) transparente Dreiecke u.U. falsch herum überlagert (blended), also die Dreiecke, die hintendran liegen über die, die vornedran sind. Aufgrund der Nicht-Kommutativität des Blendings kommt es so zu einem verfälschten Ergebnis.
TODO: überprüfen, leider kaum Informationen auf den Folien..

Teilaufgabe 10c

TODO

Aufgabe 11: Bézierkurven

Teilaufgabe 11a

- Falsch: Nur die Endpunkte werden interpoliert

- Falsch: Eine Bezierkurve vom Grad N hat $N + 1$ Kontrollpunkte
- Falsch: Sie bilden eine Basis des Polynomraums $\mathbb{R}[X]$
- Richtig, da Bézierkurven eine Basis des Polynomraums sind.

Teilaufgabe 11b

- Nein: Konvexe Hülle der Kontrollpunkte
- Ja
- Ja
- Nein: Variationsreduktion