

World & Raytracer

Übung 3 // DOC

B. Bleckmann, A. Rezai, V. Risch

Hit-Object

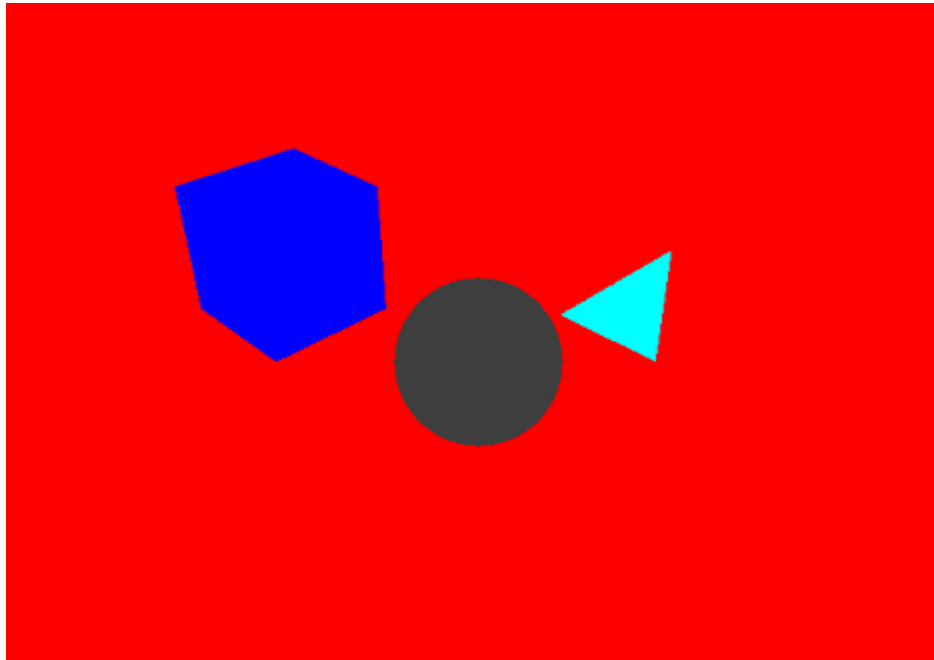
Das Hit Object wurde erfolgreich um eine zusätzliche Normale (intersectionNormal) erweitert.

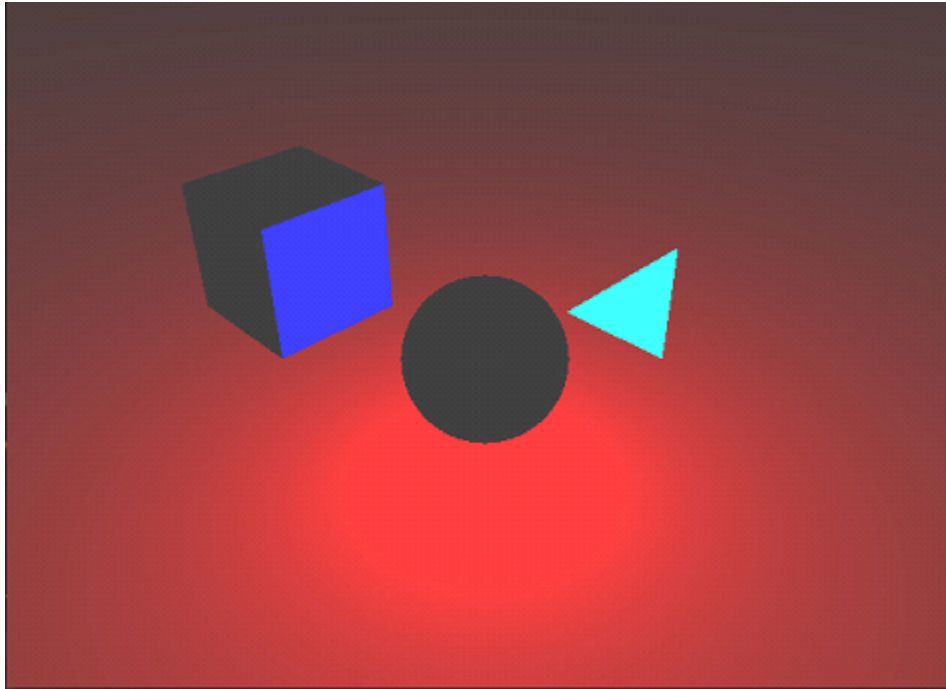
Normalen der Geometrien

Ebene und AxisAlignedBox besitzen bereits die zu verwendenden Normalen. Kugel und Dreieck müssen berechnet werden.

Kugel:

Kleinere Fehler korrigiert die evtl. zu Problemen führten (Punkt vor Strich, Rundungsfehler), Führten mitunter zu diesen Ergebnissen / Fehlern:





Dreieck:

zweiten Constructor implementiert, in einem wird Normale für komplettes Triangle berechnet, der andere bekommt 3 Normalen übergeben. Beim Dreieck kam es wie erwartet zu keinerlei Fehlern in den Berechnungen, da wir lediglich den Konstruktor mit der berechneten Normalen nutzen. Jene Formeln halfen zur Berechnung:

The cross product of two sides of the triangle equals the surface normal. So, if $V = P_2 - P_1$ and $W = P_3 - P_1$, and N is the surface normal, then:

$$N_x = (V_y * W_z) - (V_z * W_y)$$

$$N_y = (V_z * W_x) - (V_x * W_z)$$

$$N_z = (V_x * W_y) - (V_y * W_x)$$

If A is the new vector whose components add up to 1, then:

$$A_x = N_x / (|N_x| + |N_y| + |N_z|)$$

$$A_y = N_y / (|N_x| + |N_y| + |N_z|)$$

$$A_z = N_z / (|N_x| + |N_y| + |N_z|)$$

My sources:

http://freespace.virgin.net/hugo.elias/routines/r_cross.htm

[http://en.wikipedia.org/wiki/Normal_\(geometry\)](http://en.wikipedia.org/wiki/Normal_(geometry))

Let $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$ and $P_3 = (x_3, y_3, z_3)$. The normal vector to the triangle with these three points as its vertices is then given by the cross product $n = (P_2 - P_1) \times (P_3 - P_1)$. In matrix form, we then see that

$$n = \text{det} \begin{pmatrix} i & j & k \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{pmatrix}$$

$$= \begin{pmatrix} (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1) \\ (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1) \\ (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \end{pmatrix}$$

If you need that the sum of the coefficients of \hat{n} equals 1, then set α equal to the sum of the coefficients of n and then let $\hat{n} = \frac{1}{\alpha}n$. Obviously, if $\alpha = 0$ then you will never be able to satisfy your condition as any scalar multiple of n will have the same zero-sum of coefficients.

World & Raytracer

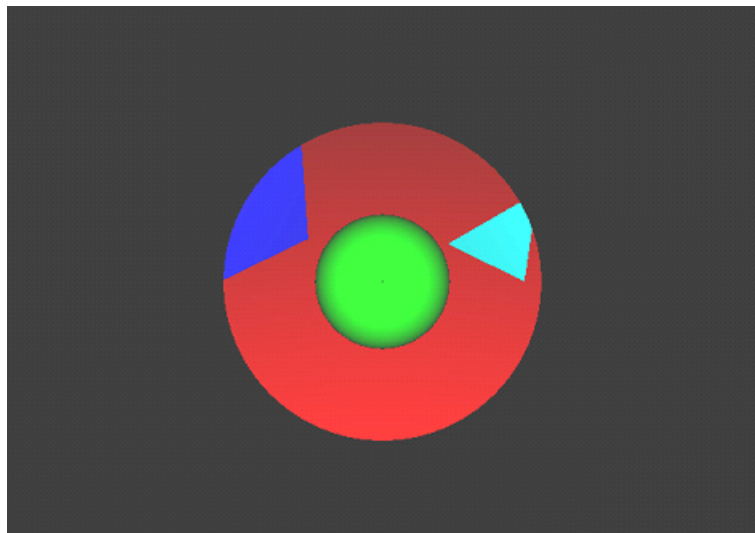
Die Änderungen an World-Klasse, sowie Raytracer konnten recht zügig und ohne größere Probleme implementiert werden.

Licht

Implementieren der Licht-Klassen verlief weitestgehend ohne Probleme anhand der UML-Klassen-Diagramme. Einige Missverständnisse und kleinere Fehler in den Berechnungen konnten schnell behoben werden anhand der Demoszene. Fehler bei Spotlight-Berechnung führte zu längerer Fehlersuche (einige Stunden)

Material

Implementieren der Material-Klassen verlief weitestgehend ohne Probleme. ColorFor Methoden teilweise etwas aufwändiger bei Lambert und Phong. Probleme bei Lambert konnten relativ schnell behoben werden. Durch testen des Lambert in Demoszene 8 stießen wir z.B. auf diesen Fehler in der Berechnung (AmbienteLight Farbe wurde nicht korrekt mit Light Farbe der Szene verrechnet)



Phong zog sich auf einige stunden hin, auf Grund eines Fehlers in der reflectOn Methode -> führte dazu, dass specular Licht "invertiert" angezeigt wurde