

Übungen zu Computergrafik

Sommersemester 2016

Blatt 8: Shading

Aufgabe 8.1: Shading Verfahren (15 Punkte)

Stellen Sie sich vor, Sie sollen für ein Dreieck die Lichtberechnung vornehmen. Das Dreieck resultiert in insgesamt 27 Pixeln auf dem Bildschirm. Wie oft muss die Lichtberechnung (berechnet aus der Normalen die Beleuchtungsstärke) unter der Verwendung von *Gouroudshading* bzw. *Phongshading* ausgeführt werden?

Bereiten Sie sich darauf vor, Ihrem Tutor im Testat die Unterschiede zwischen *Flatshading*, *Gouroudshading* und *Phongshading* zu erklären. Gehen Sie auch auf die wesentlichen Vor- und Nachteile der Verfahren bei der Benutzung für verschiedene Geometrien (Würfel, Kugel, Raumschiff Enterprise) ein.

Musterlösung:

Siehe Skript.

Aufgabe 8.2: Flatshading Implementation (40 Punkte)

In dieser Aufgabe sollen Sie die Lichtberechnung aus der Vorlesung (mit ambientem, diffusen und spekularem Licht) implementieren.

Ihre Szene soll von einem Punktlicht mit den Koordinaten $(0, 5, 2, 1)$, einer Grundintensität I_e von 10 und Abschwächungsfaktoren $C_1 = 1$, $C_2 = 0.05$ beleuchtet werden. Schaffen Sie in der Klasse `PointLight` eine geeignete Datenstruktur für die Repräsentation einer Punktlichtquelle. Die Lichtquelle muss im Controller erstellt und dem `Renderer` übergeben werden.

Die `DrawableObjects` wurden um die Möglichkeit, die in der Vorlesung genannten Oberflächen-Eigenschaften vorhalten zu können, erweitert. Diese Eigenschaften benötigen Sie für die Lichtberechnung. Wir gehen von einem vereinfachten Flat-Shading aus und färben Objekte nur in Graustufen ein. Sie können davon ausgehen, dass sowohl die diffuse als auch die spekulare Objektfarbe weiß (also 1.0) ist. Als spekularen Exponenten O_e wählen Sie standardmäßig 3.0.

Vervollständigen Sie nun die Methode `calculateBrightnessAtPoint`, die für einen Eckpunkt eines Ihrer Dreiecke mit Hilfe der neuen Beleuchtungsangaben eine Farbe (i.e., einen Grauwert) berechnet. Gehen Sie dabei davon aus, dass die Intensität der ambienten Beleuchtung I_a 0.4 beträgt und die Intensität I_e der Punktlichtquelle für die diffuse Reflexion mit der Entfernung abnimmt, d.h. $I_e = \cos(L, N) \cdot \frac{I_0}{C_1 + C_2 * r}$, wobei r der euklidische Abstand des Eckpunktes zur Lichtquelle ist. Da im Term für die spekulare Beleuchtung der Cosinus bereits berücksichtigt wird, können Sie hier annehmen, dass $I_e = \frac{I_0}{C_1 + C_2 * r}$.

Der `Renderer` wurde bereits angepasst, sodass das Mittel der für die Vertices berechneten Farben auf der ganzen Fläche aufgetragen wird. Machen Sie sich mit diesem Code (in `render`) vertraut.

Musterlösung:

Siehe Anhang.

Aufgabe 8.3: Normal-Mapping (15 Punkte)

Machen Sie sich kurz Gedanken über die folgenden Problemstellungen und skizzieren Sie Ihre Überlegungen, sodass Sie diese Ideen im Testat schnell Ihrem Tutor vorstellen können. Notieren Sie auch Probleme, die Ihnen aufgefallen sind. *Hinweis:* Sie müssen in dieser Aufgabe keine fertigen Algorithmen entwickeln, sondern lediglich grob mögliche Lösungen entwerfen.

- Ihnen wurde eine Heightmap gegeben (vgl. Folie 25 aus der Vorlesung zu Texturen). Wie können Sie daraus eine Normalmap generieren? Sie sollen also jedem Punkt auf der Textur eine Normale zuordnen, die aus den gegebenen Höheninformationen berechnet werden muss.
- Nehmen Sie nun an, dass Sie die generierte Normalmap vorliegen haben, in der an jedem Texel (Pixel in der Textur) das Triple (x, y, z) der Normale gespeichert ist. Ihnen fällt auf, dass die z -Komponente immer positiv ist, da die Normale nie in das Objekt hineinzeigen kann. Versuchen Sie diese Besonderheit und die Eigenschaft, dass Normalen immer die Länge 1 haben, auszunutzen, um die Normalen effizienter zu speichern (z. B. mit zwei float-Werten (m, n) , statt mit dreien). Überlegen Sie sich, wie man zwischen (m, n) und (x, y, z) umwandeln kann.
- Nun müssen Sie die Lichtberechnung für einen Pixel (auf dem Bildschirm) durchführen. Ihnen wurde die Normale aus der Normalmap (x_n, y_n, z_n) sowie die Normale (x, y, z) der existierenden Geometrie gegeben. (x, y, z) ist bereits in WC transformiert worden; (x_n, y_n, z_n) liegt aber noch in dem Texturkoordinatensystem vor. Überlegen Sie sich grob, wie Sie die Normale aus der Normalmap nun in WC überführen können (denken Sie an weitere eventuell benötigte Angaben).

Aufgabe 8.4: Fragen (30 Punkte)

Beantworten Sie Ihrem Tutor Fragen zu den Inhalten der Vorlesung.