



Computergrafik (SS 2018)

Übung 4

fällig Montag **14. Mai**, 14:20

- Geben Sie bei jeder Abgabe alle Ihre Matrikelnummern auf jedem Blatt an.
- Verspätet eingereichte Abgaben können nicht gewertet werden.

Aufgabe 1 (Theorie: Lighting & Shading)

- (a) (4P) Werten Sie das Phong-Beleuchtungsmodell am Punkt $(0, 0, -3)$ mit der Normalen $(0, 0, 1)$ aus. Verwenden Sie dabei *nicht* die Blinn-Approximation. Die folgenden Parameter sind gegeben:
- Projektionszentrum an der Stelle $(0, 0, 0)$
 - Punktlichtquelle 1 an der Stelle $(2, 0, -1)$ mit Farbe $(1, 1, 1)$
 - Punktlichtquelle 2 an der Stelle $(0, 2, -4)$ mit Farbe $(0, 1, 1)$
 - $C_a = \text{diag}(0.2, 0.2, 0.2)$
 - $C_d = \text{diag}(0.5, 0.5, 0)$
 - $C_s = \text{diag}(1, 1, 1)$
 - Shininess $s = 2$
- (b) (2P) Werten Sie das Phong-Beleuchtungsmodell für die gleiche Situation, allerdings nun *mit* Blinn-Approximation und Shininess $s = 8$, aus.
- (c) (2P) Gegeben ist das Dreieck mit den Vertices $(0, 0, -2)$, $(2, 0, -2)$, $(0, 2, -2)$ und den Vertex-Normalen $(0, 0, 1)$, $(1, 0, 0)$, $(0, 1, 0)$. Die Beleuchtung des Punktes $(1, 1, -2)$, den das Dreieck enthält, soll mittels Phong Shading berechnet werden. Für welchen Punkt und welche Normale oder welche Punkte und welche Normalen muss dazu das Phong-Beleuchtungsmodell ausgewertet werden?

Hinweis: Wundern Sie sich nicht, wenn sich bei diesen Aufgaben "unschöne" Werte ergeben; es reicht dann aus, einige Nachkommastellen anzugeben.



Aufgabe 2 (Praxis: Phong Lighting + Flat Shading)

Schauen Sie sich den Code für Übungsblatt 4 an und öffnen Sie ihn in Ihrem Browser.

- (a) (12P) Ergänzen Sie die Funktion `lightTriangles` um Code, der das Phong-Beleuchtungsmodell mit Blinn-Approximation auswertet. Verwenden Sie Flat Shading, mit einer Auswertung des Beleuchtungsmodells pro Dreieck (ausgewertet am Schwerpunkt des Dreiecks).

Es ist eine Punktlichtquelle vorgesehen. Die Position dieser Punktlichtquelle (`lightPos`) können Sie mit den Slidern variieren. Die Farbe des Lichts der Lichtquelle ist als `lightColor` gegeben.

Das Projektionszentrum befindet sich am Punkt `camPos`.

Die ambient-Koeffizienten C_a erhalten Sie in dieser Aufgabe, indem Sie `triColor` mit `cAmbient` multiplizieren. Analoges gilt für die diffuse- und specular-Koeffizienten. Der Shininess-Parameter liegt als `shine` vor. Die Parameter des Beleuchtungsmodells können Sie mit den vier unteren Slidern variieren.

Speichern Sie das Ergebnis der Phong-Beleuchtung in die Variable `phongColor`.

Hinweis: Beachten Sie die Methoden `subtract`, `normalize`, `add`, `cross`, `dot` von `Vector3D` bzw. `Point3D`; diese können hier nützlich sein.

Hinweis: Wir haben C_a , etc., als Matrix kennengelernt. Da alle Nichtdiagonaleinträge hier 0 sind, lässt es sich auch als Vektor darstellen. Für die Multiplikation mit einem anderen Vektor ist dann die komponentenweise Vektormultiplikation nötig; dazu steht `Vector3D.mulVec` zur Verfügung.

Hinweis: Sie können die Größe des Canvas mit der Konstanten `size` anpassen, um ggf. das Rendering zu beschleunigen.

Aufgabe 3 (Bonus: Phong Shading)

Ersetzen Sie das Flat Shading aus Aufgabe 2 durch Phong Shading, d.h. das pro-Dreieck-Lighting durch pro-Pixel-Lighting. Als Vertexnormale n für Vertex p können Sie (vor der Modelviewtransformation!) $n = p$ annehmen (aufgrund der speziellen Situation, dass wir es hier mit einer Einheitskugel um den Ursprung zu tun haben.)