

Computergrafik 1

Lab 6

Aufgabe 1 Diffuse und ambiente Beleuchtungsberechnung

Die Farbintensität bei reiner diffusen Beleuchtungsberechnung ist proportional zu

$$f_{\text{diffuse}} = \max\left(\frac{\vec{n}^T}{\|\vec{n}\|} \cdot \frac{\vec{l}}{\|\vec{l}\|}, 0\right),$$

wobei \vec{n} die Normale des zu beleuchtenden Punktes ist. Weiter ist \vec{l} der Vektor von dem zu beleuchtenden Punkt \vec{p} zur Lichtquellenposition \vec{L} :

$$\vec{l} = \vec{L} - \vec{p}.$$

Wir führen Beleuchtungsberechnung im Camera-Space aus. Deshalb müssen Normale \vec{n} und \vec{p} in den Camera-Space transformieren. In Lab 5 haben Sie bereits die Normale in den Camera-Space transformiert. Nun transformieren wir \vec{p} in den Camera-Space.

Die Model-View Matrix M sollte bei Ihnen wie folgt berechnet werden:

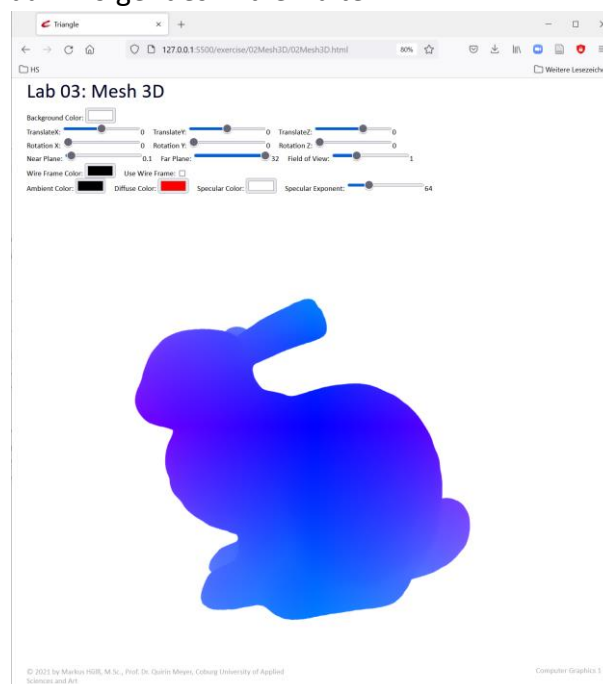
$$M = T(\vec{t}) \cdot R_x(\alpha_x) \cdot R_y(\alpha_y) \cdot R_z(\alpha_z)$$

Sie transformiert die Positionen aus dem Object-Space in den Camera-Space.

- (a) Übergeben Sie die Model-View Matrix an den Vertex Shader mittels einer uniformen Variablen. Nennen Sie diese `u_mv`!

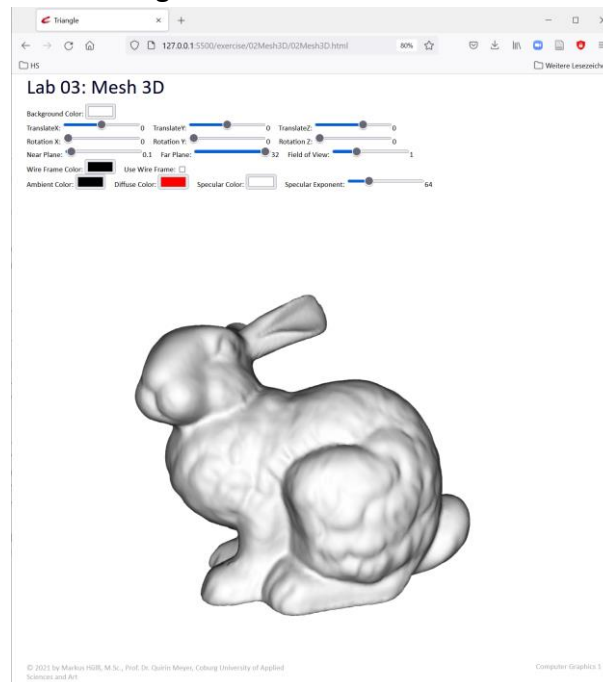
Hinweis: Dies funktioniert ähnlich wie bei `u_mvInvT` in Lab 5.

- (b) Transformieren Sie die eingehende Position nun mittels der Matrix `u_mv` in den Camera-Space und leiten Sie diese an den Fragment-Shader weiter. Nennen Sie die Variable `fs_position`. Geben Sie im Fragment-Shader den Absolutbetrag von `fs_position` als Farbe aus. Sie sollten dann folgendes Bild erhalten:

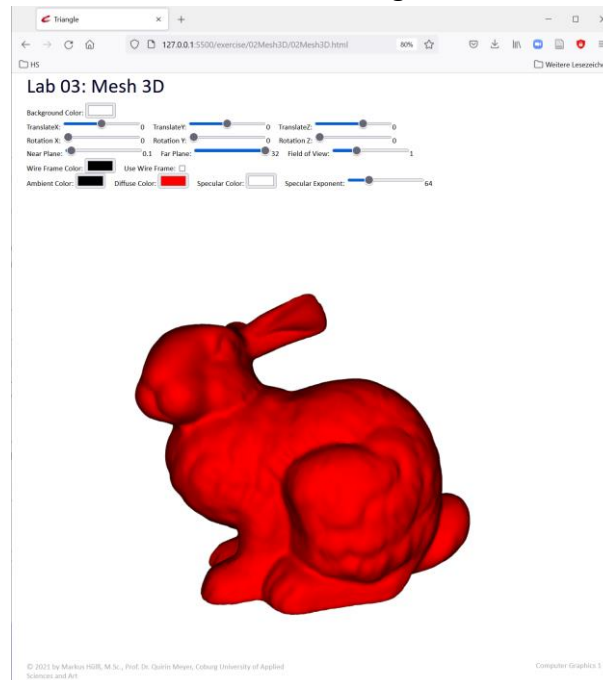


- (c) Berechnen Sie im Fragment-Shader den Vektor \vec{l} . Nehmen Sie als Position der Lichtquelle $\vec{L} = \vec{0}$, also die Kameraposition, an.

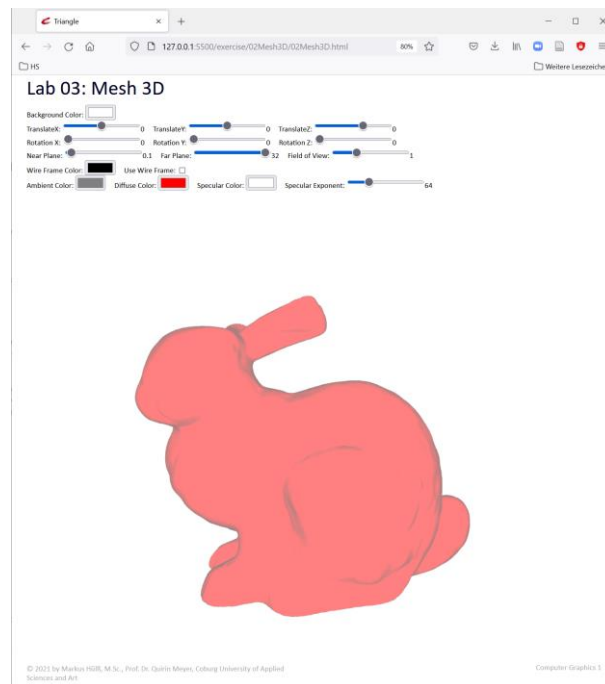
- (d) Berechnen Sie f_{diffuse} . Gewichten Sie anschließend f_{diffuse} mit der Farbe Weiß und geben Sie diese aus. Sie sollten dann folgendes Bild erhalten:



- (e) Im User Interface kann man die diffuse Farbe konfigurieren. Übergeben Sie die diffuse Farbe (sie steht in der draw Methode bereits in `diffuseR`, `diffuseG` und `diffuseB` zur Verfügung) als uniforme Variable an den Fragment-Shader und gewichten Sie f_{Diffuse} mit dieser Farbe anstelle von Weiß. Sie sollten dann folgendes Bild erhalten:



- (f) Geben Sie nun auch die ambiente Farbe an den Fragment Shader und berücksichtigen Sie diese in der Beleuchtungsberechnung. Wie das geht, finden Sie in den Folien. Wenn Sie den ambienten Anteil im User Interface auf den RGB Vektor (128, 128, 128) stellen, erhalten Sie folgendes Bild:

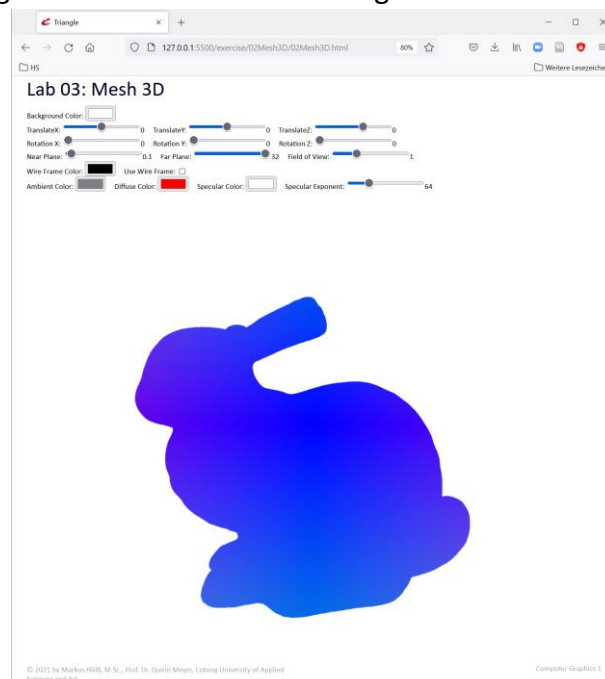


Aufgabe 2 Spekulare Beleuchtungsberechnung

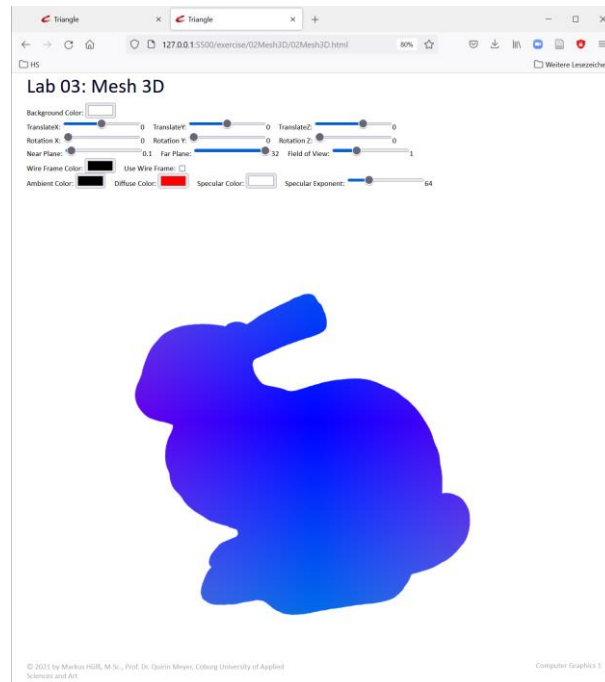
Für den spekularen Anteil nach Blinn-Phong benötigen wir den View-Vektor \vec{v} und zwar im Camera-Space. Das ist der Vektor von dem zu beleuchtenden Punkt \vec{p} zur Position der Kamera \vec{C} , d.h.:

$$\vec{v} = \vec{p} - \vec{C}.$$

- (a) Berechnen Sie den Vektor \vec{v} im Fragment-Shader, normalisieren Sie diesen und geben Sie dessen Absolutbetrag als Farbe aus. Sie sollten folgendes Bild erhalten:



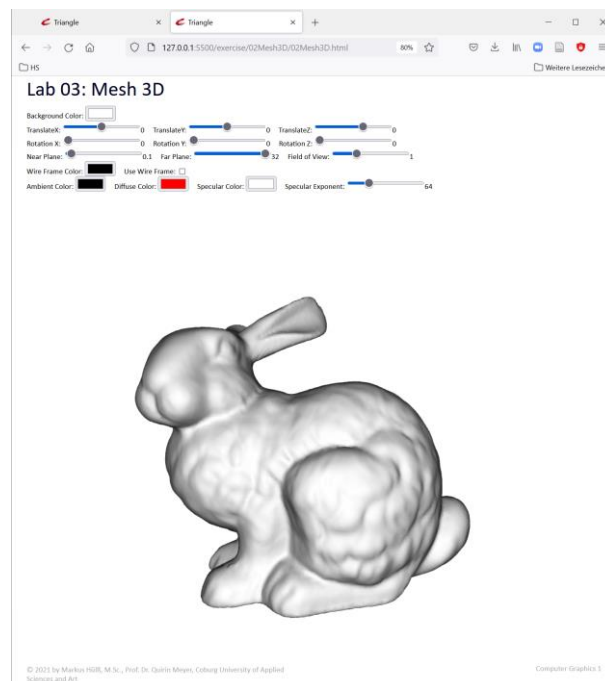
- (b) Bestimmen Sie den normalisierten Halfway-Vektor \vec{h} aus \vec{l} und \vec{v} . In den Folien finden Sie, wie das funktioniert. Wenn Sie den Absolutbetrag der Farbe ausgeben erhalten Sie folgendes Bild:



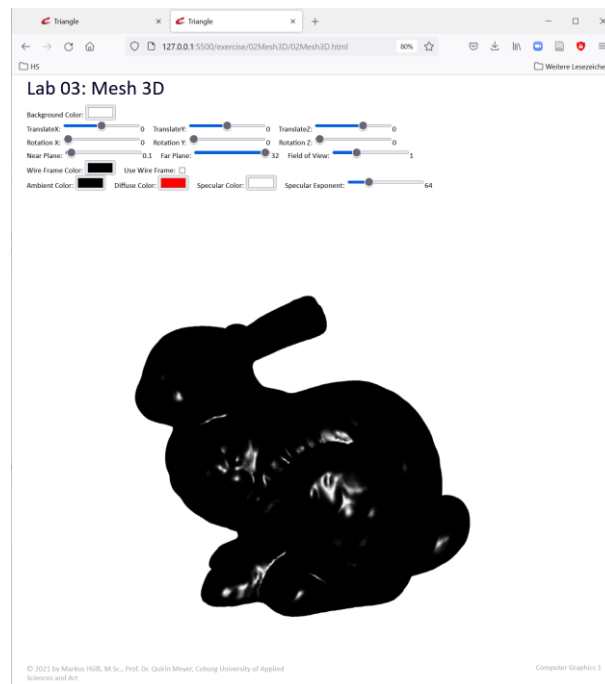
(c) Wie Sie aus der Vorlesung wissen, ist der spekulare Anteil proportional zu

$$f_{\text{Specular}} = \max\left(\frac{\vec{n}^T}{\|\vec{n}\|} \cdot \frac{\vec{h}}{\|\vec{h}\|}, 0\right).$$

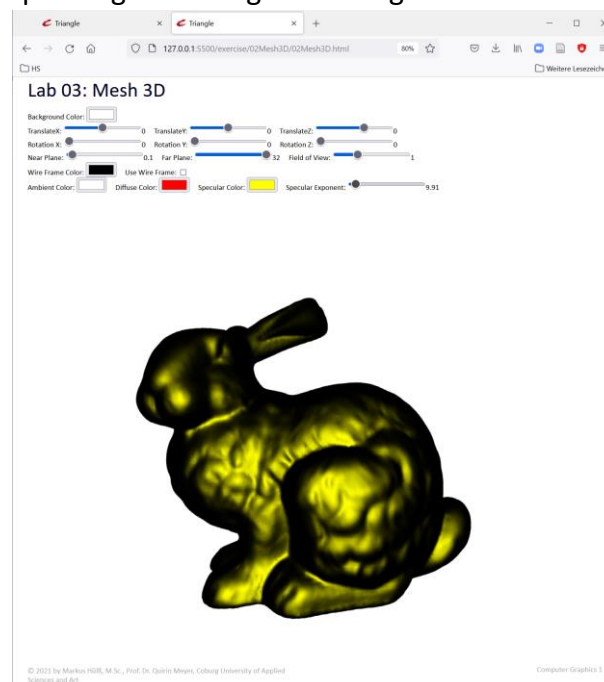
Zum Testen können Sie f_{Specular} mit der Farbe Weiß gewichten und ausgeben. Dann sollten folgendes Bild erhalten:



(d) Um die Größe des Glanzlichtes zu verändern wird f_{Specular}^s gerechnet. Gewichten Sie im Fragment-Shader $f_{\text{Specular}}^{128}$ mit der Farbe Weiß. Recherchieren Sie, wie die Potenzfunktion in GLSL aufgerufen wird. Dann erhalten Sie folgendes Bild:



(e) Übergeben Sie den Exponenten und die spekulare Farbe mittels uniformer Variablen an den Shadern. In den Variablen `specularR`, `specularG`, `specularB` und `specularExponent`, die bereits in der `draw` Methode ausgelesen werden finden, Sie die geeigneten Werte. Sie können dann zum Beispiel folgende Ausgabe erzeugen



(f) Kombinieren Sie nun ambientes, diffuses und sepkulares Licht und geben Sie die Farbe aus:

