

Test-Framework für Shader



Ausarbeitung für die Lehrveranstaltung
Visuelle Effekte

Kaveh Yousefi

Dokumenteninformation

Version:	1.0
Datum:	10.07.2015

Historie

Version	Datum	Bearbeitung
1.0	10.07.2015	Erste Version; Datei angelegt.

Inhalt

- Projektziel
- Klärung von Grundbegriffen
- Vorstellung umgesetzter Shaders
- Beschreibung des Frameworks
- Aussicht und kritische Würdigung

Projektziel

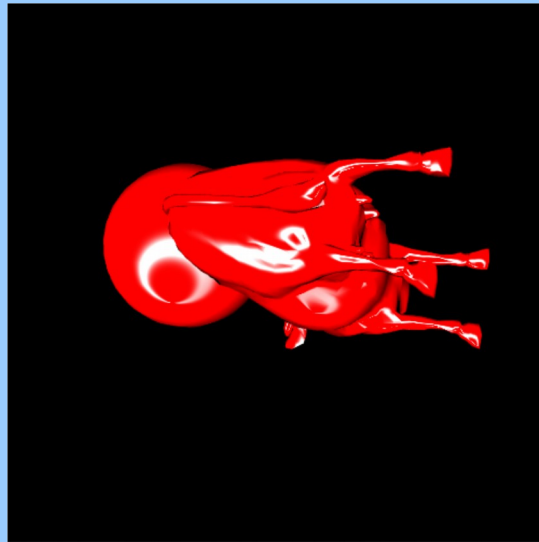
- Ziel ist die Umsetzung eines Test-Frameworks.
 - Dieses basiert auf einer Kombination von
 - HTML5
 - JavaScript
 - WebGL.
- Es enthält Implementierungen verschiedener Shading-Modelle*.
- Dies befähigt zum
 - Experimentieren mit unterschiedlichen Modellen
 - Vergleichen der Shader.

*) Anmerkung: Ein Shading-Model ist in der Regel eine Umformulierung und Vereinfachung des zu Grunde liegenden BRDF (*Bidirection Reflectance Distribution Function*).
Vgl. [dempski2005advanced], S. 90.

Grundbegriffe & Prinzipien

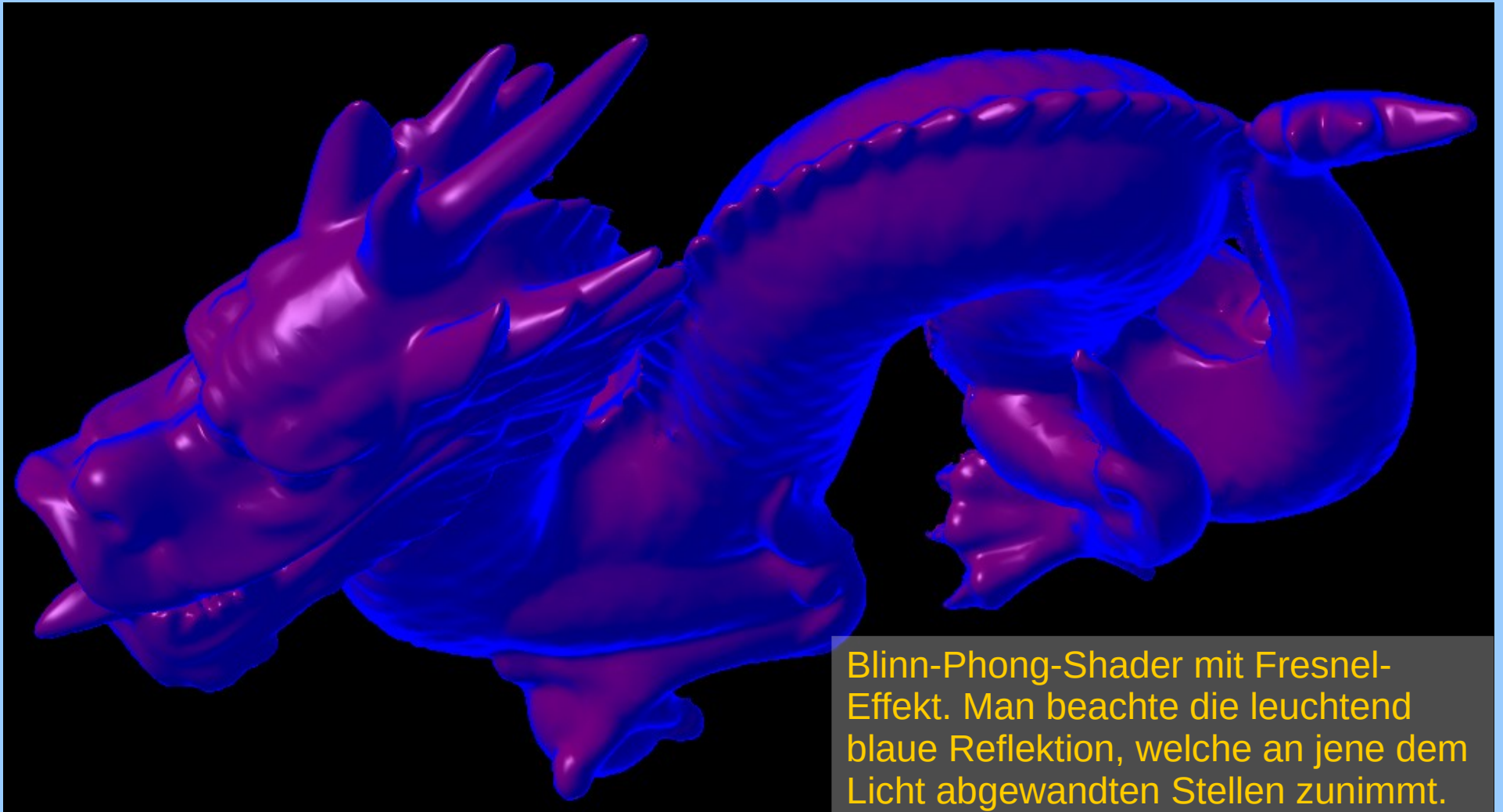
- Anisotropie:
 - Im Kontext der Beleuchtungsrechnung: Reflektive Eigenschaft eines Materials ist abhängig vom Winkel des Betrachters um die Oberflächennormale.
⇒ Aussehen ist abhängig vom Blickwinkel.
 - Gegensatz: Isotropie.
- Fresnel-Effekt:
 - Anteil am reflektiertem bzw. gebrochenem Licht ist abhängig vom Betrachtungswinkel.
 - Beispiel: Aus bestimmten Winkeln scheint Wasser weniger „durchsichtig“.

Beispiel - Anisotropie



Beispiel für
Anisotropie: Die Form
der Lichtflächen auf
den Modellen ändert
sich bei Modifikation
des Blickwinkels.

Beispiel – Fresnel-Effekt



Blinn-Phong-Shader mit Fresnel-Effekt. Man beachte die leuchtend blaue Reflektion, welche an jene dem Licht abgewandten Stellen zunimmt.

Vorgestellte Modelle

- Blinn-Phong
- Cook-Torrance
- Minnaert
- Oren-Nayar
- Phong
- Strauss
- Ward (anisotropic)



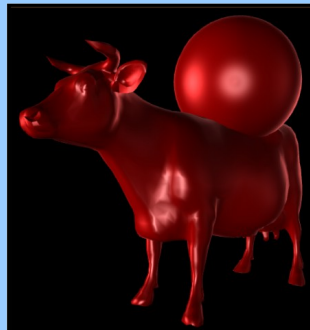
Blinn-Phong



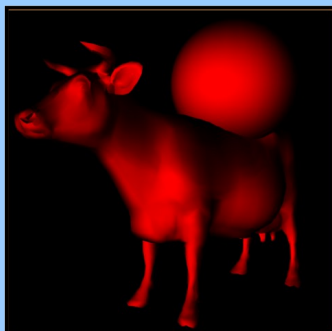
Phong



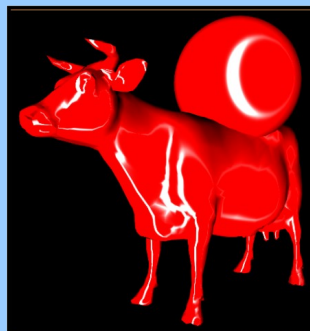
Cook-Torrance



Strauss



Minnaert



Ward
(anisotropic)



Oren-Nayar

Blinn-Phong



Blinn-Phong

- Abgeleitet aus dem Phong-Beleuchtungsmodell.
- Vereinfachung der Gleichung für bessere Effizienz.
- Oftmals realistischere Ergebnisse.
- Einsatzgebiet:
 - Allgemeine Materialien.
 - Insbesondere aber Plastik.

Blinn-Phong - Parameter

Parameter	Beschreibung	Anmerkung
ambient color	Umgebungsbezogene Materialfarbe.	
diffuse color	Diffuse Materialfarbe.	
emissive color	Eigenleuchten als Materialfarbe.	
specular color	Spekulare Materialfarbe.	
shininess	Definiert die Stärke des Glänzens der Oberfläche.	

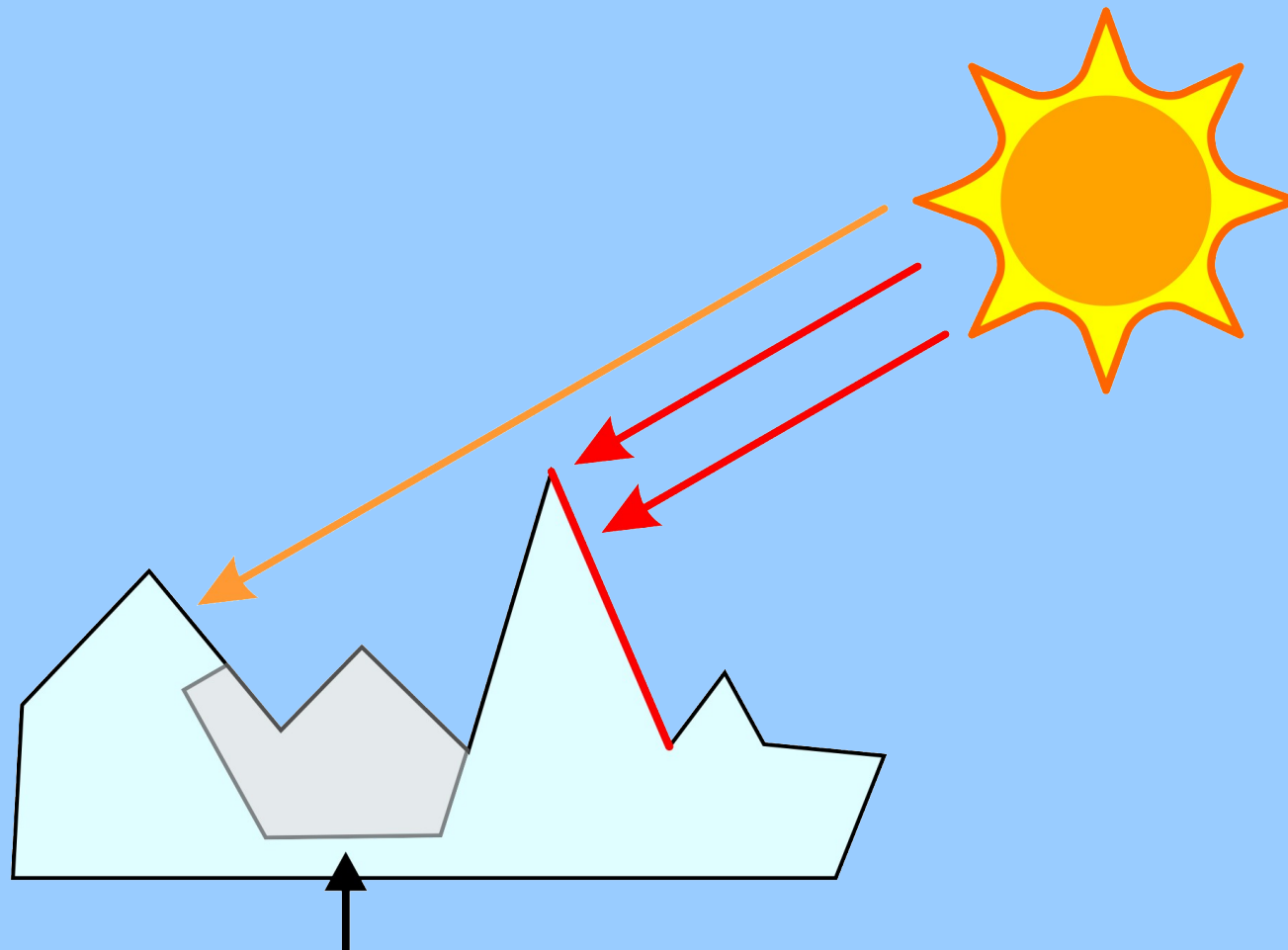
Cook-Torrance



Cook-Torrance

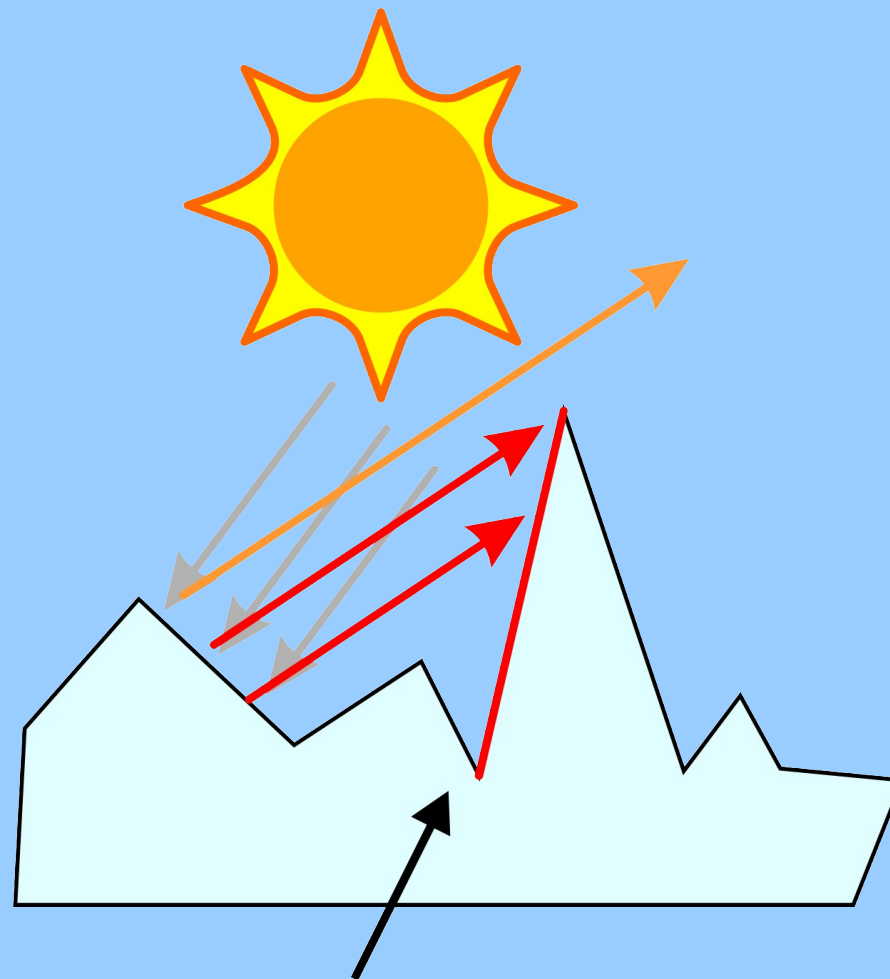
- Betrachtet Oberfläche als Anhäufung winziger Mikrofacetten, welche Licht in verschiedene Richtungen reflektieren.
- Berücksichtigt *shadowing* und *masking*:
 - *shadowing*: Mikrofacetten können verhindern, dass andere Mikrofacetten Licht erhalten.
 - *masking*: Mikrofacetten können verhindern, dass andere Mikrofacetten Licht auswerfen.
 - Beide Effekte beeinflussen diffuse und spekulare Reflektion.
- Steuerung erfolgt über drei Terme:
 - geometrischer Term
 - Fresnel-Term
 - Rauheitsterm.

Cook-Torrance - shadowing



Mikrofacette auf der rechten Seite
blockiert einfallendes Licht und
schattiert damit jene links von ihr.

Cook-Torrance - masking



Mikrofacette auf der rechten Seite
blockiert das reflektierte Licht jener
links von ihr und **maskiert** sie damit.

Cook-Torrance

- Geometrischer Term:
 - Berücksichtigt *shadowing* und *masking*.
- Fresnel-Term:
 - Steuert Menge des reflektierten Lichts.
 - Bestimmt Reflektion von Mikrofacetten und damit wie metallisch die Oberfläche erscheint.
 - Ersetzt spekulares Licht anderer Modelle \Rightarrow realistischer.
- Rauheitsterm:
 - Rauheit basiert auf Verteilung der Steigung der Mikrofacetten.
 - Wird über eine Verteilungsfunktion berechnet.
- Einsatzgebiet:
 - Metalle.

Cook-Torrance - Parameter

Parameter	Beschreibung	Anmerkung
c	Rauheitsfaktor für die Blinn-Verteilungsfunktion (Rauheitsterm).	Nicht notwendig für die Beckmann-Verteilungsfunktion.
m	Rauheitsfaktor für den Rauheitsterm.	
R0	Faktor für den Einsatz im Fresnel-Term.	Wird zumeist als Teil der Rauheitsfaktoren angesehen.

Minnaert



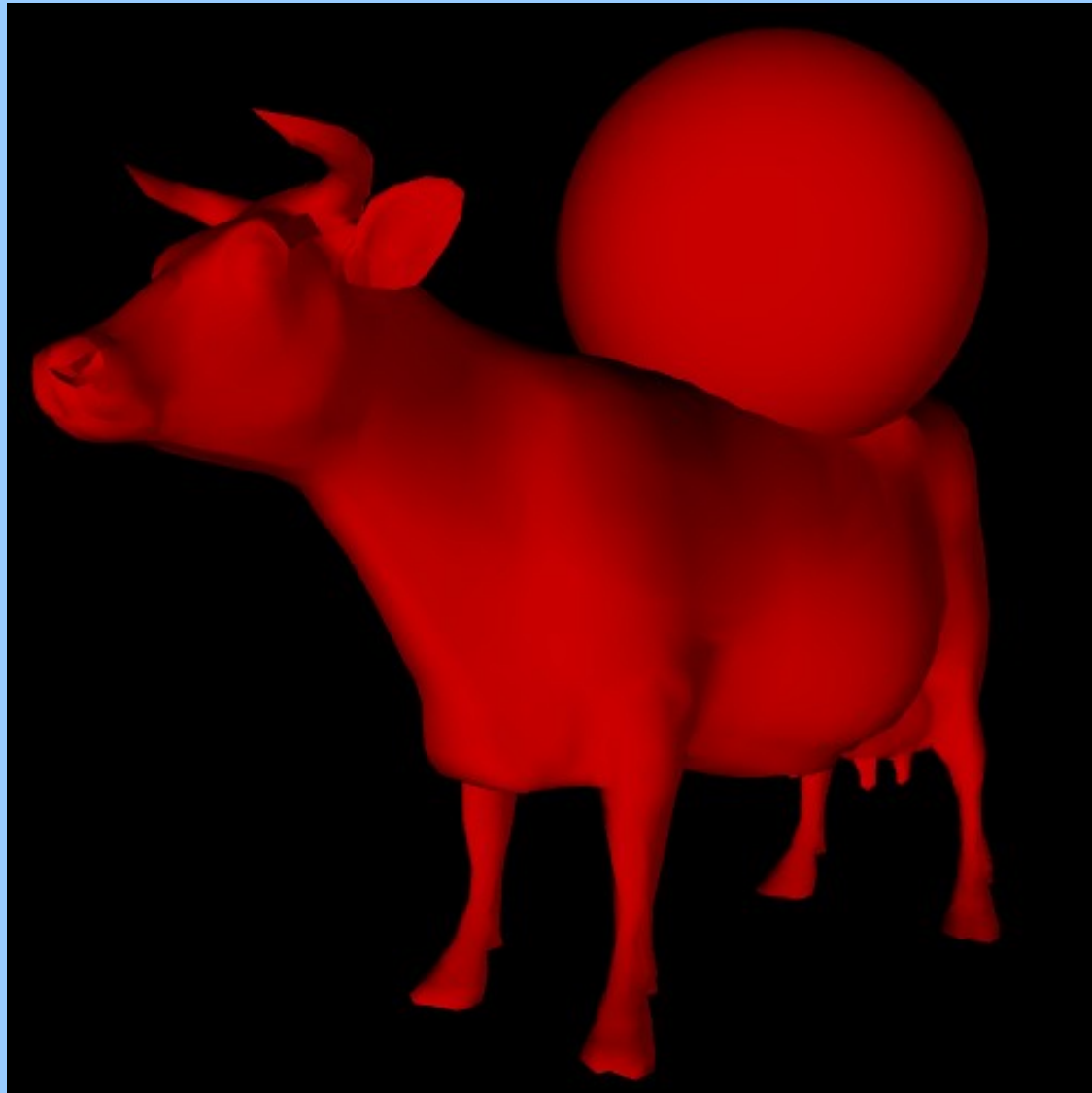
Minnaert

- Diffuses Modell.
- Eigentlich für andere Zwecke entwickelt.
- Verwendet einen Rauheitsfaktor.
- Charakteristisch: dunkle Streifen entlang Objektkanten.
- Einsatzgebiet:
 - Darstellung von Stoffen, insbesondere **Seide**.

Minnaert - Parameter

Parameter	Beschreibung	Anmerkung
diffuse color (D)	Diffuse Oberflächenfarbe.	
roughness (m)	Rauheitsfaktor.	

Oren-Nayar



Oren-Nayar

- Ein rein diffus-reflektives Modell (kein spekularer Anteil).
- Erweiterung des Lambert-Modells.
- Einbringung von Rauheit (*roughness*):
 - Oberfläche wird als Anhäufung von Mikrofacetten betrachtet.
 - Rauheit basiert auf Normalverteilung der Facetten: Hohe Standardabweichung \Rightarrow größere Unterschiede in ihren Richtungen.
- In der Regel „flachere“ Darstellung als bei Lambert-Modell.
- Einsatzgebiet:
 - **Raue** Oberflächen.
 - Beispielsweise Mondoberfläche, Ton, Stoffe.

Oren-Nayar - Parameter

Parameter	Beschreibung	Anmerkung
diffuse color (D)	Diffuse Oberflächenfarbe.	
roughness (σ)	Rauheitsfaktor.	

Phong



Phong

- Eines der ältesten Modelle.
- Bedeutsam auf Grund von Einbringung der spekularen Reflektion, berechnet über den Reflektionsvektor.
- Physikalisch nicht sehr plausibel.
- Jedoch leicht zu implementieren.
- Einsatzgebiet:
 - Allgemeine Materialien.
 - Insbesondere aber Plastik.

Phong - Parameter

Parameter	Beschreibung	Anmerkung
ambient color	Umgebungsbezogene Materialfarbe.	
diffuse color	Diffuse Materialfarbe.	
emissive color	Eigenleuchten als Materialfarbe.	
specular color	Spekulare Materialfarbe.	
shininess	Definiert die Stärke des Glänzens der Oberfläche.	

Strauss



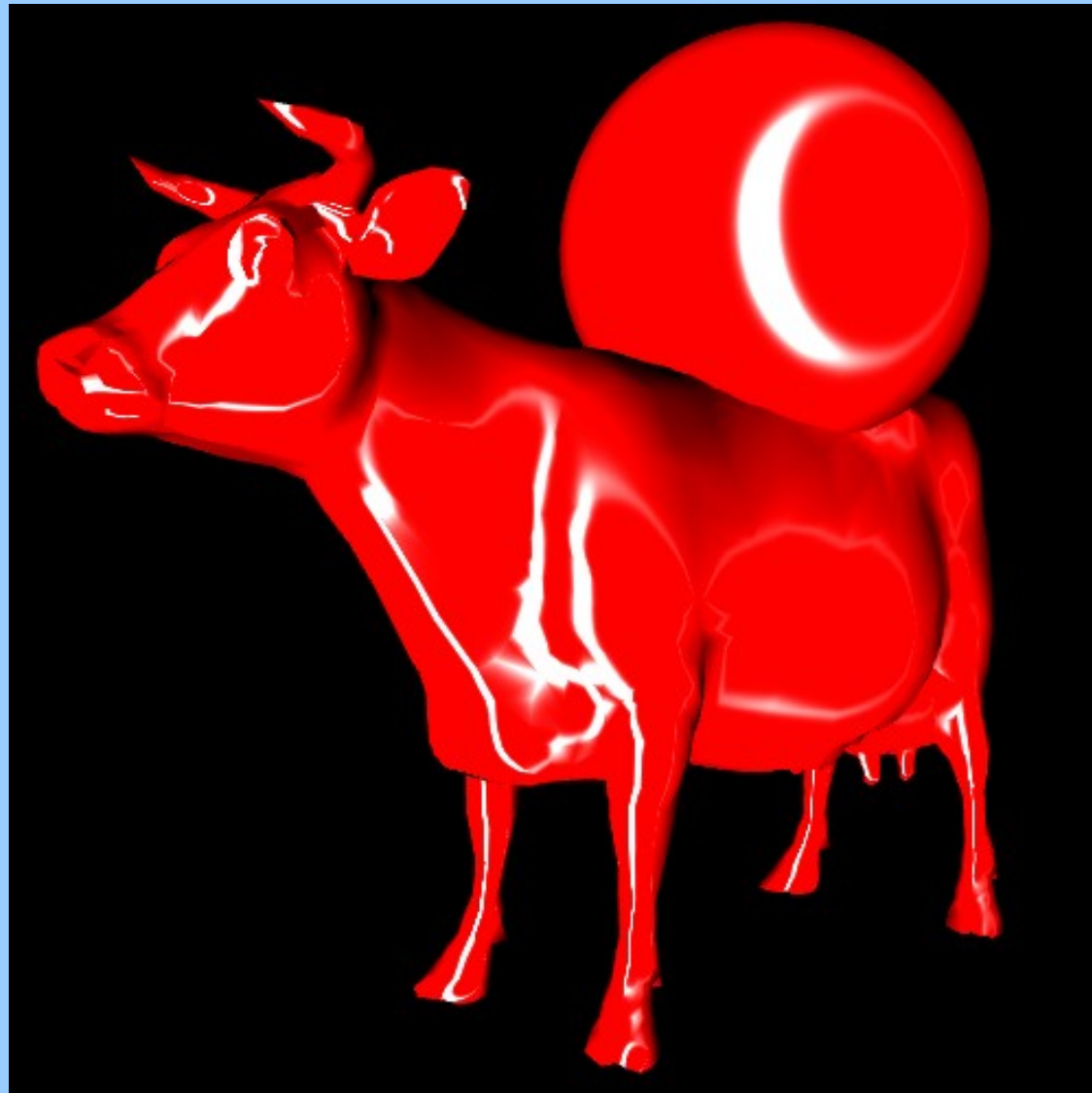
Strauss

- Hauptziele:
 - Einfache Handhabung von Parametern.
 - Gleichzeitig große Bandbreite an Materialien darstellbar.
- Kombiniert existente Beleuchtungsmodelle.
- Einsatzgebiet:
 - Verschiedenste Materialien.
 - Insbesondere geeignet für Metalle.
 - Jedoch auch Plastik gut darstellbar.

Strauss - Parameter

Parameter	Beschreibung	Anmerkung
metalness m	Wie metallisch ist das Material?	Wertebereich: [0, 1].
smoothness s	Wie glatt ist das Material?	Wertebereich: [0, 1].
transparency t	Simuliert die Wirkung von Licht (Energie).	Nicht identisch mit Alpha im RGBA-Farbmodell, sollte aber gleichen Wert besitzen. Wertebereich: [0, 1]
surface color c	(Diffuse) RGB-Oberflächenfarbe.	Einzige Farbe im Modell.
index of refraction n	Brechungsindex	Nicht im Shader verwandt, nur für Ray-Tracing oder Global Illumination.
fresnel constant K_f	Beeinflusst den Fresnel-Effekt.	Eigentlich kein Parameter, sondern Konstante = 1,12.
shadow constant K_s	Beeinflusst die Simulation von Schattierungen auf der Oberfläche.	Eigentlich kein Parameter, sondern Konstante = 1,01.
off-specular peak k	Spekularer Zusatzwert für sehr raue Oberflächen.	In der Regel konstant auf 0,1 gesetzt.

Ward (anisotropic)



Ward (anisotropic)

- Empirisches Modell: Basiert auf Beobachtungen.
 - Hieraus wurden Gleichungen erstellt.
- Physikalisch recht plausibel.
- Ward-Modell existiert in zwei Varianten:
 - isotropisch
 - anisotropisch.
- Anisotropie durch zwei orthogonale Rauheitskoeffizienten erreicht (x- und y-Richtung).
- Einsatz:
 - Metalle, insbesondere gebürstete.

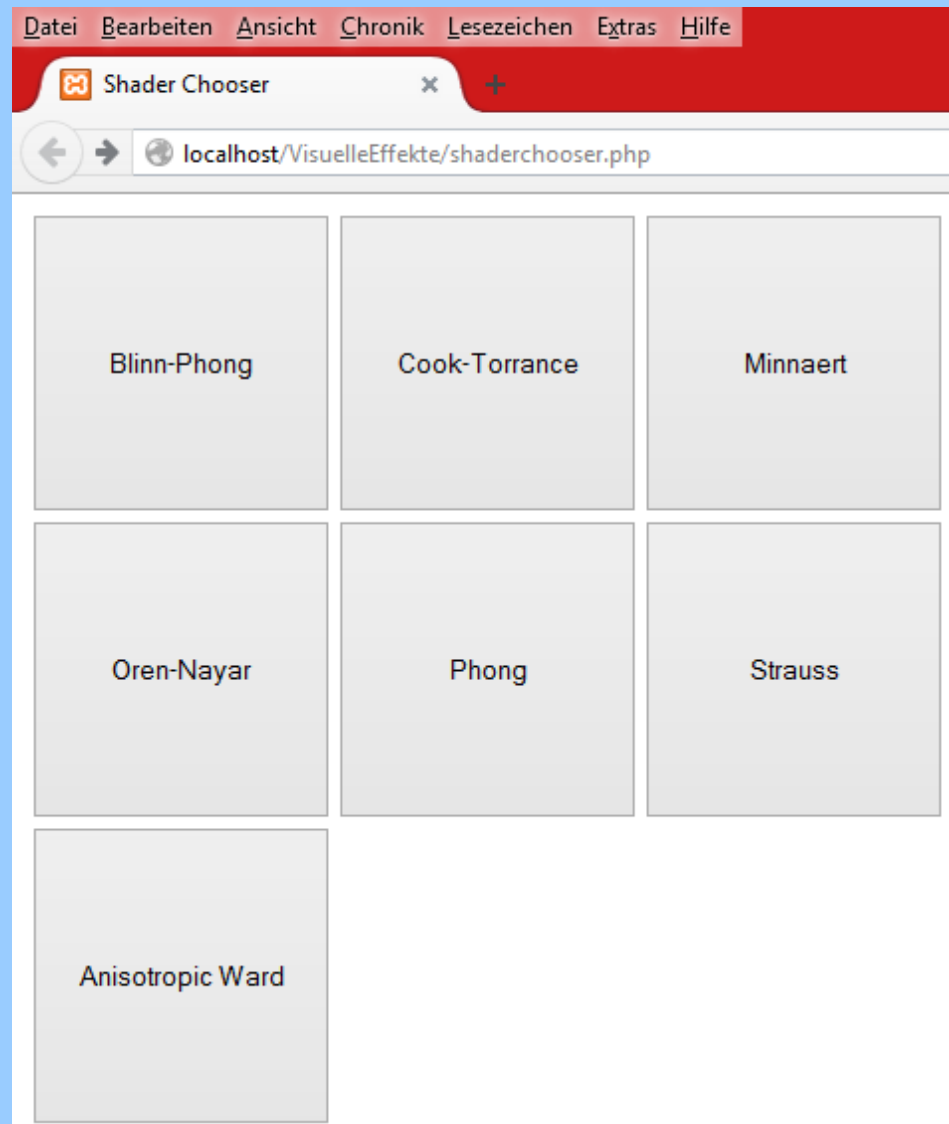
Ward - Parameter

Parameter	Beschreibung	Anmerkung
direction	Richtung des Materials als 3D-Vektor.	
roughness.x	Rauheitsfaktor entlang der X-Achse.	Sorgt für Anisotropie.
roughness.y	Rauheitsfaktor entlang der Y-Achse.	Sorgt für Anisotropie.

Framework – Prinzip (1/3)

- Verschiedene Shading-Modelle werden angeboten.
 - Jeder Vertex- bzw. Fragment-Shader residiert in einer eigenen Datei und wird zurzeit mittels PHP eingelesen und der Quellcode in HTML eingebettet.
- Die Auswahl des Shader-Modells erfolgt in einem eigenen HTML-Formular, dem „Shader Chooser“.
 - Die Auswahl wird an das entsprechende PHP-Script versandt und dort ausgewertet.
 - Daraufhin wird, wie oben beschrieben, die dynamische Einsetzung von GLSL-Code durchgeführt.

Framework – Prinzip (2/3)



Das „Shader Chooser“-Formular mit den zurzeit angebotenen Shader-Modellen. Jedes Modell kann über Betätigen der entsprechenden Schaltfläche geladen werden.

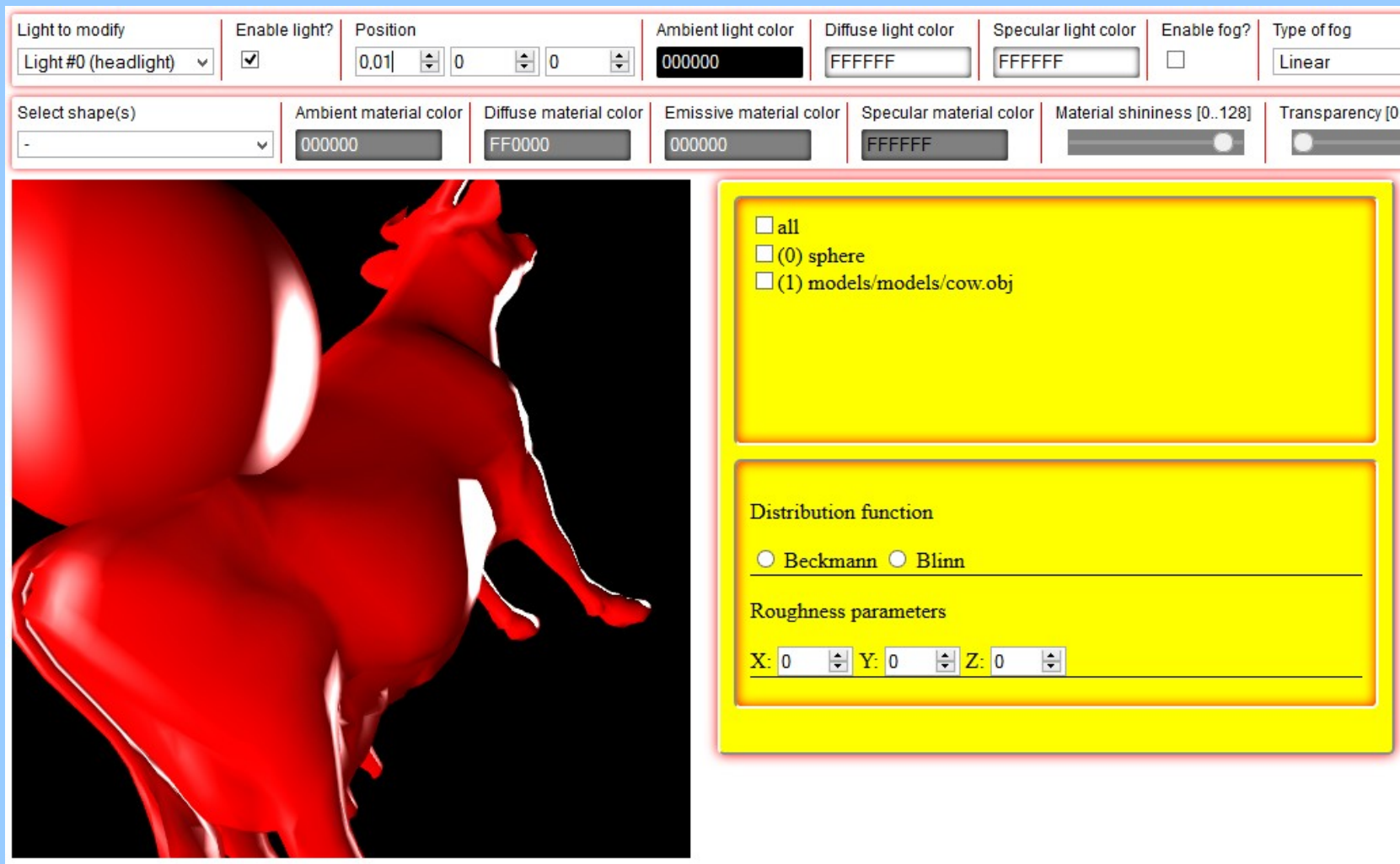
Framework – Prinzip (3/3)

- 3D-Objekte befinden sich in der Szene.
 - Die Objekte können über Menüstruktur ausgewählt werden: Entweder einzeln oder alle gleichzeitig.
- ⇒ Simulation eines flachen Szenegraphen (eigentlich eine „Szeneliste“).
- Die Steuerung innerhalb der Szene ist über Computermouse oder Tastatur möglich.

Framework – Konfiguration (1/2)

- Dynamische HTML-Bedienelemente existieren.
- Je nach Shader sind unterschiedliche Parameter notwendig.
 - Adäquate Steuerelemente werden bereitgestellt.
 - Diese ermöglichen die Einstellung der Parameter.
 - Bsp.: Farbe, Rauheit.
- Übergreifende Einstellungen sind daneben möglich.
 - Bsp.: Transparenz.
- Weitere Szene-Elemente sind unabhängig davon konfigurierbar:
 - Licht, Nebel

Framework – Konfiguration (2/2)

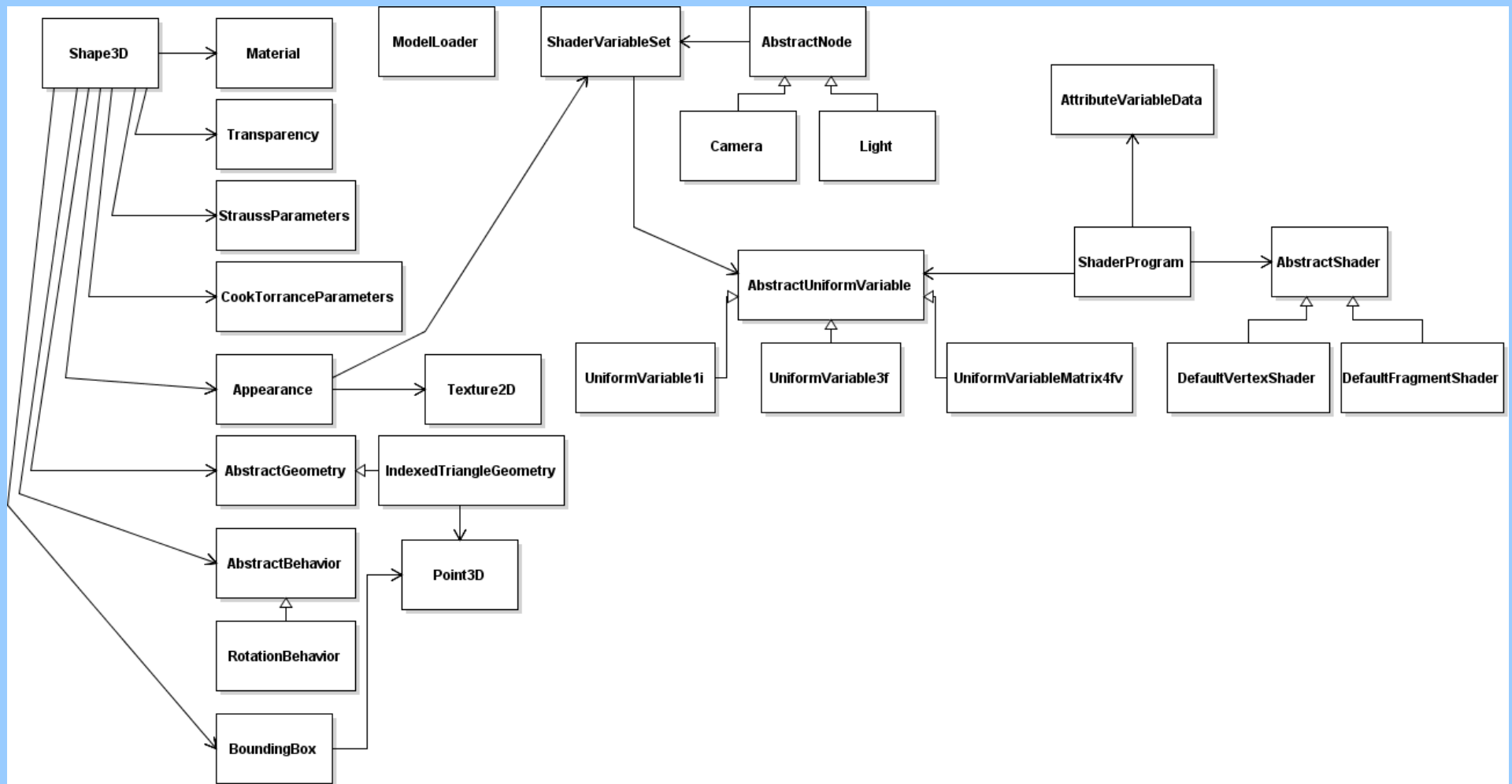


Beispiel für angestrebte dynamische Konfiguration: Nur die Parameter des aktuell geladenen Cook-Torrance-Shaders werden in der gelben Tafel angezeigt. Darüber ist der Szenegraph sichtbar.

Framework – Architektur (1/2)

- Die Architektur ist lose an die Spezifikation der Java-Bibliothek *Java 3D* angelehnt [[sowizral1997java](#)].
- Daher findet eine Unterteilung in eine größere Anzahl an Typen („Klassen“) mit wohl definierten Verantwortlichkeiten statt.
- Primäre Ziele:
 - Erweiterbarkeit
 - Bspw. durch neue Modellparameter für das Material.
 - Bspw. durch „abstrakte“ Oberklasse für Geometrien.
 - Lesbarkeit
 - Bspw. Modularisierung über separate JavaScript-Dateien.
 - Bspw. pro *uniform*-Variabletyp eigener JavaScript-Typ.

Framework – Architektur (2/2)



Teilausschnitt aus dem Klassendiagramm des aktuellen Systemzustands.

Obligatorische Erweiterungen

- Einige Punkte bedürfen noch der Implementierung bzw. Vervollständigung:
 - Maus-basierte Kameranavigation.
 - Beachtung der Texturen.
 - Fresnel-Effekte.
 - Komplettierung der dynamischen Bedienelemente zur Shader-Konfiguration.
 - Transformation selektierter 3D-Objekte (Translation, Rotation, Skalierung, Scherung).

Optionale Erweiterungen

- Sofern genügend Zeit vorhanden ist, sind folgende Erweiterungen denkbar:
 - Weitere Shading-Modelle:
 - Ashikhmin-Shirley Anisotropic
 - Lafortune
 - Schlick
 - Shader pro 3D-Objekt setzen.
 - Dynamisches Hinzufügen von 3D-Objekten.

Ausblick

- Potenzielle zukünftige Weiterentwicklungen, welche wahrscheinlich nicht mehr in das Projekt Einzug finden werden:
 - Implementierung einer interaktiven Objektauswahl („picking“).
 - Schattenberechnung.
 - Effizienzberechnung (Performance einzelner Shader messen und ausgeben).

Zu klärende Fragen

- Shader originalgetreu abbilden?
 - Zurzeit enthalten viele Shader vom Modell abweichende Komponenten, um mehr Möglichkeiten zu bieten.
 - Bsp.: Oren-Nayar-Shader, eigentlich rein diffus, besitzt auch spekularen, ambienten und emissiven Term wie ein Phong-Shader.
- XAMPP als Grundvoraussetzung inadäquat?
 - Zurzeit eingesetzt, um
 - vom Benutzer ausgewählten Shader mittels HTML-Formular auszuwerten
 - und Shader-Dateien dynamisch in HTML einzusetzen.
 - Wirkt möglicherweise als Hürde für Interessenten.

Literaturverzeichnis

[3drender2001], Jeremy Birn, Fresnel Effect,
<http://www.3drender.com/glossary/fresneleffect.htm>

[dempski2005advanced] Dempski, Kelly and Viale, Emmanuel,
Advanced lighting and materials with shaders, 2005

[engel2008programming] Wolfgang F. Engel, Jack Hoxley, Ralf Kornmann, Niko Suni, Jason Zink, Programming Vertex, Geometry, and Pixel Shaders, 2008

[kornmann2011d3dbook] Ralf Kornmann, D3DBook, 2011,
http://content.gpwiki.org/D3DBook:Table_Of_Contents

[sowizral1997java] Kevin Sowizral, Kevin Rushforth, Henry Sowizral, The Java 3D API Specification, 1997