

# Blender Dokumentation/ Druckversion/ Material

Aus Wikibooks

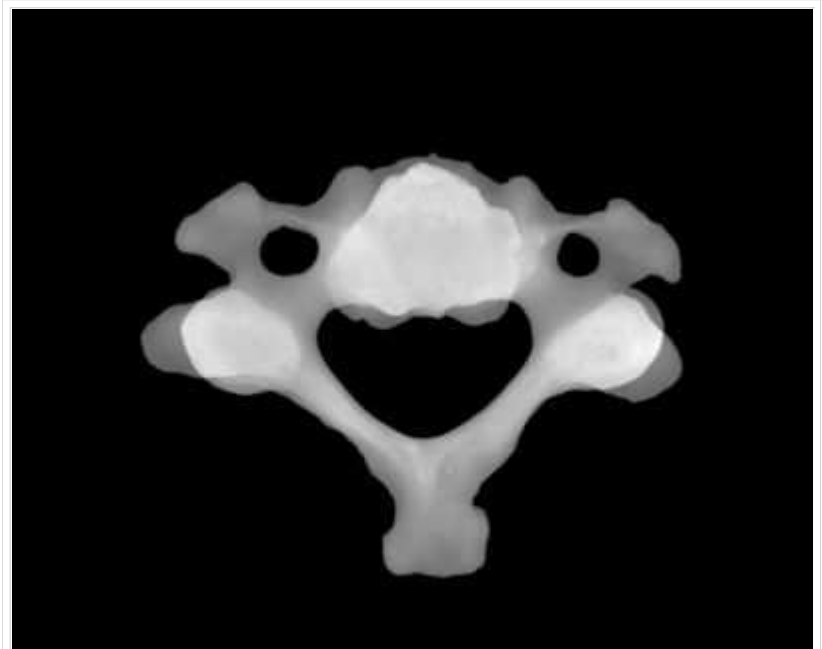
< Blender Dokumentation

## Materialien und Texturen

Diese Seite bezieht sich auf **Blender v2.42a**

Fünf Dinge bestimmen das Aussehen einer Oberfläche in Blender:

1. Die Struktur der Oberfläche, d.h. bei Meshobjekten das Mesh selbst.
2. Das grundlegende Material der Oberfläche, insbesondere seine Farbe, die Transparenz (Durchsichtigkeit), die Menge des zurückgeworfenen Lichtes sowie die Art und Weise, wie die Lichtreflexionen berechnet werden. In Blender wird dies als Material bezeichnet.
3. Auf das Material aufgebrachte Grafiken und Filter. Diese können nicht nur die Farbe, sondern fast alle Materialeinstellungen beeinflussen. In Blender wird dies als Textur bezeichnet.
4. Die Beleuchtung.
5. Effekte, z.B. Glow (Glühen), Halos und insbesondere Ramp-Shader.



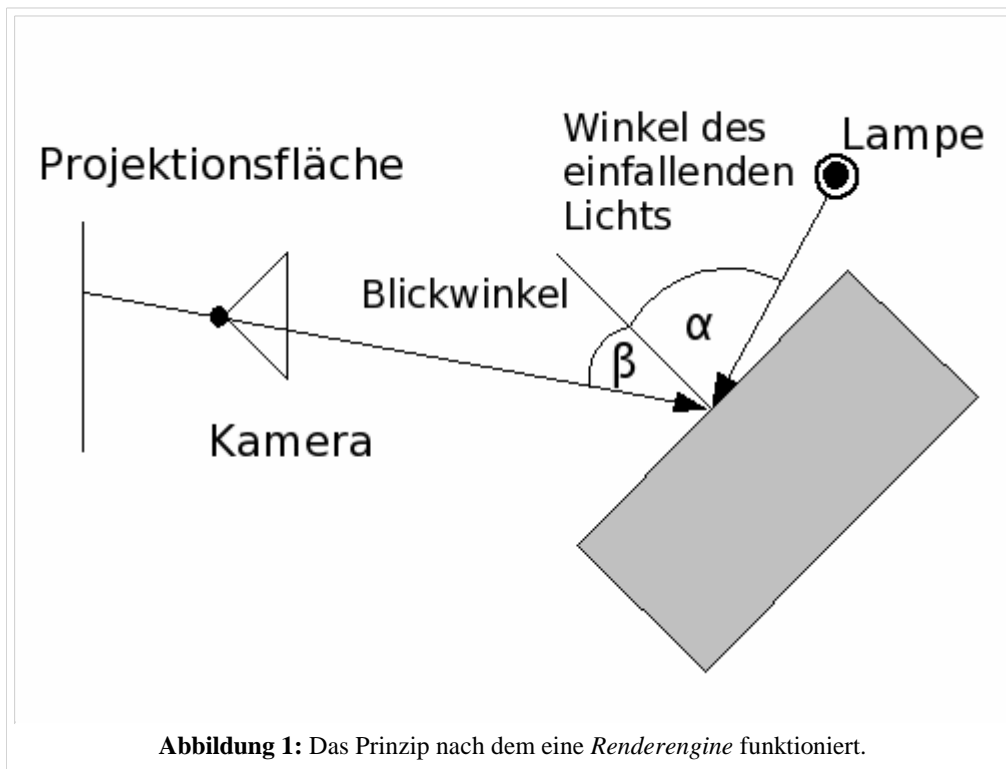
**Abbildung 1:** Neu in Version 2.42: *Transmissivity* berücksichtigt die Dicke von Objekten. So entstand dieses virtuelle Röntgenbild eines Halswirbelknochens.

Es ist zunächst nicht ganz einfach zu verstehen, wie man ein bestimmtes Aussehen erreichen kann, zumal der gleiche Effekt auf unterschiedlichen Ebenen mit völlig unterschiedlichen Einstellungen erzielt werden kann. Im Abschnitt Materialeinstellungen - Links finden Sie eine Reihe von Links zu Materialsammlungen. Beginnen Sie mit diesen, und variieren Sie sie. So werden Sie am schnellsten befriedigende Ergebnisse erzielen.

Bevor Sie mit Materialien wirksam umgehen können, müssen Sie verstehen, wie simuliertes Licht und Oberflächen in der *Renderengine* von Blender interagieren und wie die Materialeinstellungen diese gegenseitige Beeinflussung kontrollieren. Je besser Sie die Engine verstehen, um so mehr können Sie aus ihr herausholen.

Das gerenderte Bild, das Sie mit Blender erstellen, ist eine Projektion der Szene auf eine imaginäre Fläche, der so genannten *Projektionsfläche* (viewing plane). Die Projektionsfläche entspricht dem Film in einer normalen Kamera, oder den Stäbchen und Zäpfchen des menschlichem Auges, außer, dass sie kein echtes, sondern nur simuliertes Licht empfangen kann.

Um ein Bild einer Szene zu rendern, müssen wir zuerst feststellen, welches Licht der Szene welchen Punkt der *Projektionsfläche* erreicht. Der einfachste Weg diese Frage zu beantworten ist es, eine gerade Linie (den simulierten Lichtstrahl) zurück durch den Punkt auf der *Projektionsfläche* und den Blickpunkt (der Position der Kamera) zu verfolgen, bis sie eine renderfähige Oberfläche in der Szene trifft. Daraus können wir dann ermitteln, welches Licht diesen Punkt treffen würde. Die Oberflächeneigenschaften und der Winkel des einfallenden Lichtes sagen uns, wie viel des Lichtes entlang des einfallenden Blickwinkels reflektiert würde (**Abbildung 1**).



Es gibt zwei verschiedene Arten von Reflexionen, die auf einer Oberfläche entstehen können, wenn sie von einem Lichtstrahl getroffen wird: diffuse Reflexion und gerichtete Reflexion (Glanzlichter). Diffuse Reflexion und gerichtete Reflexion unterscheiden sich eigentlich nur durch die jeweilige Beziehung zwischen dem Winkel des einfallenden Lichtes und dem Winkel des reflektierten Lichtes zueinander.

## Links

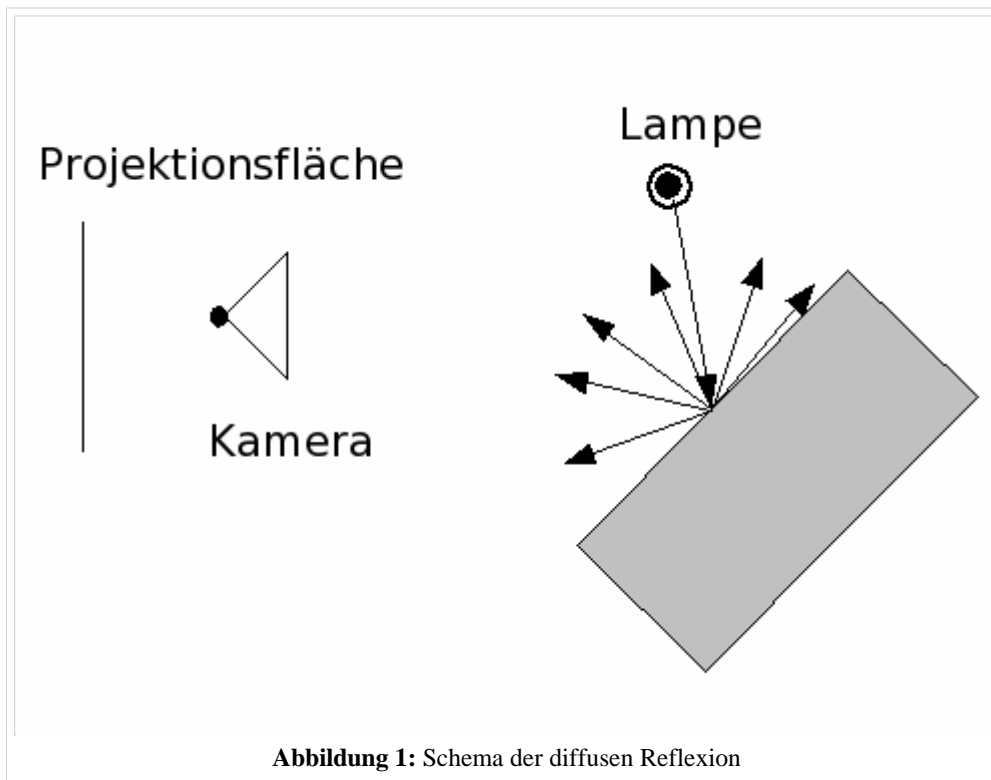
Die englischsprachige Vorlage dieses Textes (<http://www.blender.org/modules/documentation/html/c4206.html>)  
 Erklärung des Fotorezeptors (<http://de.wikipedia.org/wiki/Photorezeptor>)  
 Photorealistic Texturing for Dummies (<http://leigh.cgcommunity.com/tutorials.htm>)

## Diffuse Reflexion

*Diese Seite bezieht sich auf  
 Blender v2.42*

Licht, das auf eine rauhe Oberfläche trifft und durch diffuse Reflexion zurückgeworfen wird, wird zerstreut, d.h. in alle Richtungen mehr oder weniger gleichmäßig ("isotropisch") verteilt (**Abbildung 1**). Das bedeutet, dass die Kamera unabhängig vom Blickwinkel immer die selbe Menge Licht empfängt.

Diffuses Licht ist im einfachsten Fall also unabhängig vom Standpunkt der Kamera. Die Menge des - eine Oberfläche treffenden - Lichtes hängt vom Einfallswinkel des Lichtes ab, also von der Position der Lichtquelle. Wenn das meiste des auftreffenden Lichtes diffus reflektiert wird, bekommt die Oberfläche ein mattes Aussehen.



Eine perfekt gleichmäßige Reflexion des Lichtes kommt z.B. bei rauhem Plastik vor, bei vielen Materialien wird das Licht nicht in alle Richtungen gleich emittiert. Als extremes Gegenbeispiel kann gebürstetes Metall dienen, an einer solchen Oberfläche wird das Licht sehr stark in eine bestimmte Richtung reflektiert.

Blender besitzt fünf verschiedene Berechnungsmethoden zur Berechnung der diffusen Reflexion, die entsprechend unterschiedliche Oberflächeneigenschaften widerspiegeln. Darüberhinaus gibt es noch eine Option *Tangent*, die sich sowohl auf diffuse als auch auf spekuläre Reflexionen bezieht. Diese verschiedenen Berechnungsmethoden werden als "Shader" bezeichnet.

Die Shader haben verschiedene Parameter. Allen gemeinsam sind die Parameter *Ref* [Reflektivität], der die relative Menge (0 bis 1) des zurückgeworfenen Lichtes angibt, und die Farbe *Col* der Oberfläche.



Benutzen Sie in der Regel *Oren-Nayar* als Shader für das diffuse Licht, und *Blinn* als Shader für die Glanzlichter.

Die implementierten Shader sind:

- *Lambert* - Das war Blenders einziger Diffusionsshader bis zur Version 2.27. Deshalb beziehen sich alle alten Tutorien auf ihn und alle Bilder, die vor der Version 2.28 gerendert wurden, benutzten diesen *Shader*. Dieser *Shader* besitzt nur die Standardparameter.
- *Oren-Nayar* - Dieser *Shader* wurde erstmals in Blender 2.28 vorgestellt. Mit ihm nähert sich die Diffusion etwas mehr der Realität an, da dieser *Shader* neben den beiden Standardparametern (*Col* und *Spec*) noch einen dritten benutzt, der die Rauigkeiten einer Fläche festlegt (*Rough*). Dadurch wird das Licht etwas weicher verteilt. Benützen Sie zunächst kleine *Rough* Werte, unter 0.5.
- *Toon* - Dieser *Shader* wurde ebenfalls erstmals in Blender 2.28 vorgestellt. Er ist nicht dafür gedacht, die Realität nachzuahmen, sondern ein zeichentrickähnliches Material zu erstellen. Mit klaren Schattengrenzen und gleichmäßig beleuchteten und im Schatten liegenden Bereichen. Trotzdem ist er relativ simpel und benötigt zwei weitere Parameter, die die Größe der beleuchteten Bereiche (*Size*) und die Schärfe der Schattengrenzen (*Smooth*) definieren. Der *Toon* Shader ist sehr variabel.
- *Minnaert* - Diesen Shader gibt es erst seit der Version 2.37. Er entspricht dem *Lambert* Shader, besitzt aber noch einen weiteren Parameter *Dark*. Größere *Dark* Werte als 1 dunkeln Flächen direkt in Aufsicht ab, wodurch das Material ein samtiges Aussehen erhält. Um die gleiche Gesamthelligkeit zu erreichen, müssen Sie den *Ref* Wert entsprechend erhöhen. *Dark* Werte kleiner als 1 hellen Flächen direkt in Aufsicht auf, wodurch

das Material metallischer erscheint.

- *Fresnel* - In Version 2.42 eingeführt. Flächen die vom Licht streifend getroffen werden, werden durch den Fresnel Shader besonders aufgehellt. Die Parameter *Fresnel* und *Fac* entsprechen denen bei Raytracing Spiegelungen.
- *Tangent* - Option seit Version 2.42. Dies ist ein extrem anisotroper Shader, d.h. das Licht wird stark in eine bevorzugte Richtung reflektiert, und nicht in alle Richtungen gleichmäßig. Die Richtung der Lichtemission wird durch die UV-Koordinaten festgelegt, d.h. Sie müssen die Objekte zunächst mit UV-Koordinaten versehen - auch wenn Sie keine Textur zuweisen wollen! Typische Oberflächen sind gebürstetes Metall oder eine CD. Das Ergebnis ist von der UV-Map und stark vom Blickwinkel abhängig.

Die folgenden Bilder sollen einen ersten Eindruck von den Shadern vermitteln, und wurden alle mit einem *CookTorr* Specular Shader von 0.1 gerendert. Um überzeugende Materialien zu erstellen, ist allerdings erheblich mehr Aufwand nötig, als nur einen Shader auszuwählen. Die Bildfehler stammen aus der Umwandlung in das GIF-Format, das hier auf Wikibooks leider als einziges Animationsformat zur Verfügung steht.

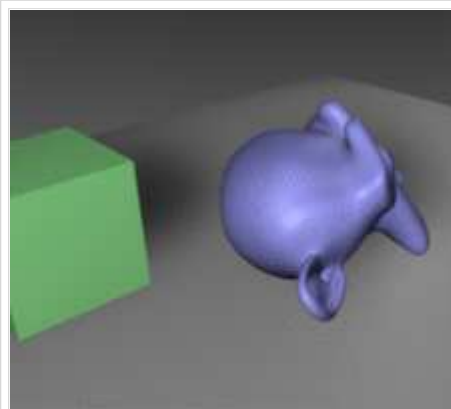


Abbildung 1: Lambert-Shader. *Ref=0.8*

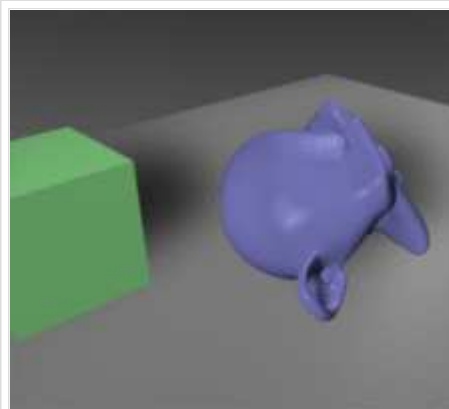


Abbildung 2: Oren-Nayar-Shader. *Ref=0.8*

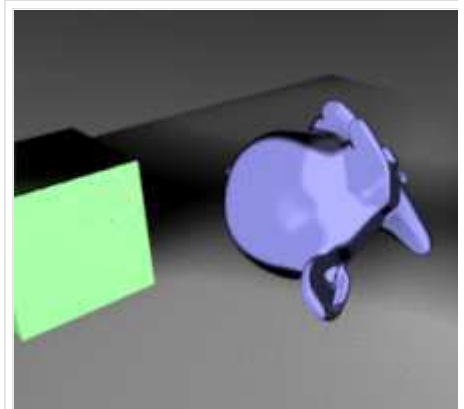


Abbildung 3: Toon-Shader. *Ref=0.8*,  
*Smooth=0.1*

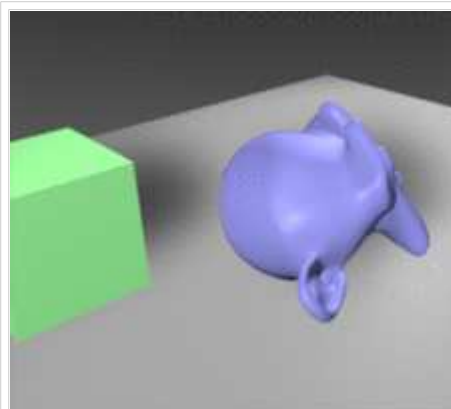


Abbildung 4: Minnaert-Shader, *Ref=0.8*

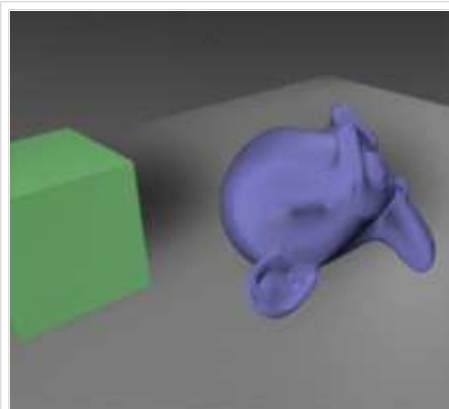


Abbildung 5: Minnaert-Shader, *Ref=1.0*

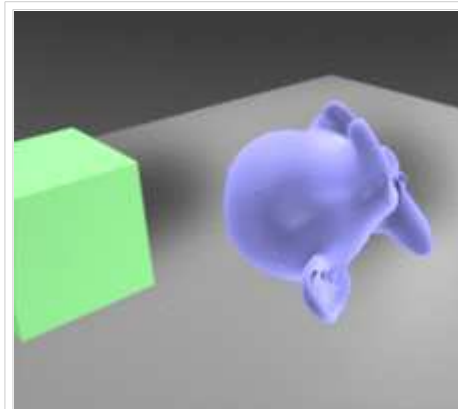
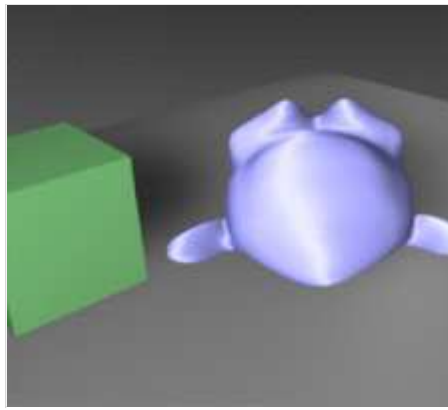


Abbildung 6: Fresnel-Shader, *Ref=0.8*,  
*Fac=1.25*



**Abbildung 7:** Tangent-Option für Lambert Ebene und der Würfel sind ohne Tangent gerendert.

## Links

Die englischsprachige Vorlage dieses Textes

([http://download.blender.org/documentation/html/c4531.html#materials\\_diffusion](http://download.blender.org/documentation/html/c4531.html#materials_diffusion))

Ein Lambertscher Strahler (diffuse Reflexion) ([http://de.wikipedia.org/wiki/Lambertsches\\_Gesetz](http://de.wikipedia.org/wiki/Lambertsches_Gesetz))

Erklärung des Begriffes *Isotrop* (<http://de.wikipedia.org/wiki/Isotrop>)

Von „[http://de.wikibooks.org/wiki/Blender\\_Dokumentation/\\_Druckversion/\\_Material](http://de.wikibooks.org/wiki/Blender_Dokumentation/_Druckversion/_Material)“

---

- Diese Seite wurde zuletzt am 12. Oktober 2006 um 21:01 Uhr geändert.
- Inhalt ist verfügbar unter der GNU Free Documentation License.
- Datenschutz
- Über Wikibooks
- Haftungsausschluss

# Blender Dokumentation/ Druckversion/ Material2

Aus Wikibooks

< Blender Dokumentation

## Glanzlichter

Diese Seite bezieht sich auf **Blender v2.42**



**Abbildung 1:** Ein Material das spiegelt und Glanzlichter besitzt.



**Abbildung 2:** Hier spiegelt das Material immer noch, die Glanzlichter sind aus.

Glanzlichter (*Specular Light*) sind etwas, was es in Wirklichkeit so nicht gibt. Aber sie lösen ein großes Problem. **Mit Glanzlichtern werden die Spiegelungen der Lampen auf Oberflächen simuliert.**

Das ist aus zwei Gründen notwendig:

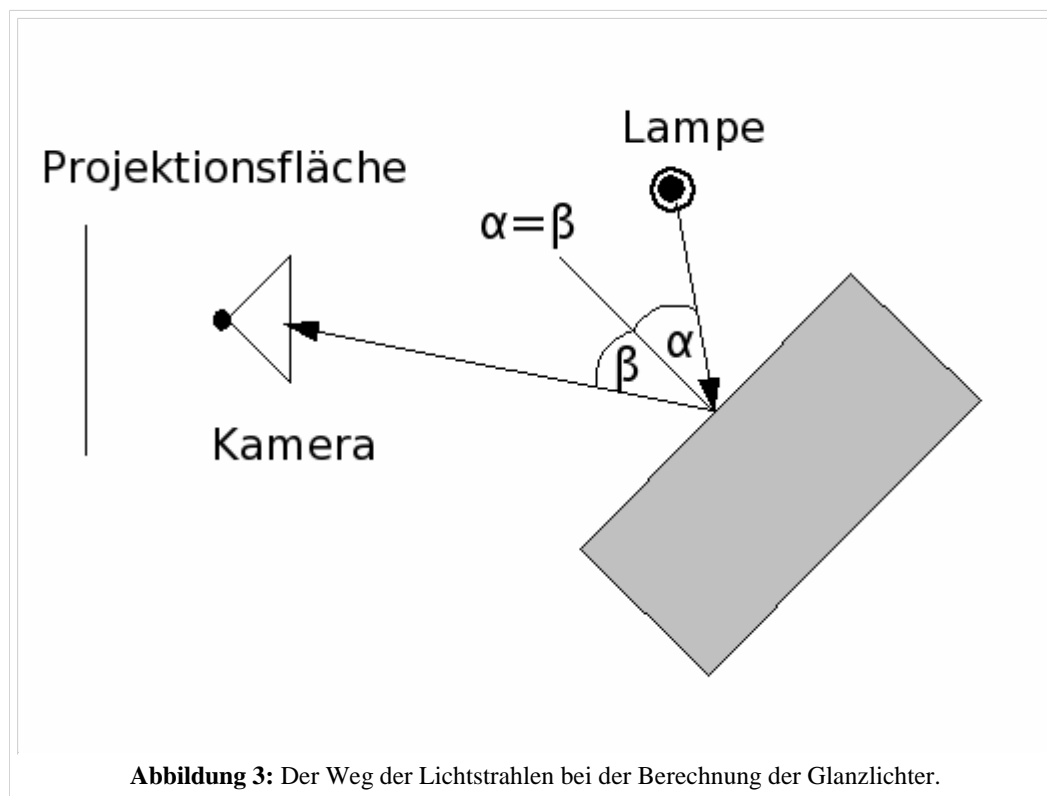
1. In Blender ist eine Lampe unsichtbar. Richten Sie die Kamera auf die Lampe, können Sie sie nicht sehen. Sie müssten eine Lampe modellieren, und ihr das richtige Material, eventuell mit entsprechendem *Emit* Wert zuweisen.

2. Oberflächen spiegeln ihre Umgebung nicht wider. Sie müssen *RayMirror* oder *Environmentmaps* benutzen, damit Oberflächen spiegeln. Dann wird das Material aber insgesamt spiegelnd, was häufig nicht erwünscht ist, außerdem werden die Renderzeiten stark verlängert.

Daher benützt man *Specular Shader*, um diese Reflexionen vorzutäuschen.

Anders als bei diffuser Reflexion sind Glanzlichter vom Blickwinkel abhängig, da es sich um die Simulation von Spiegelungen handelt. Nach dem Reflexionsgesetz [1] (<http://de.wikipedia.org/wiki/Reflexionsgesetz>) wird Licht, das eine spiegelnde Oberfläche trifft, unter dem selben Winkel reflektiert (Einfallswinkel = Ausfallswinkel). Das macht den Blickwinkel sehr wichtig. Entweder sieht man die Spiegelung, da das Licht in die Kamera gespiegelt wird, oder man sieht die Spiegelung nicht.

*Specularity* ist bei erstaunlich vielen Materialien zu beobachten, auch wenn wir diese sicher nicht als spiegelnd bezeichnen würden. Stoffe zeigen beispielsweise häufig recht große *Spec* Werte, genauso Plastik oder Holzoberflächen. In der Regel ist aber die Härte (*Hardness*) der Glanzlichter klein.



In der Realität werden diffuse Lichtstreuung und Glanzlichter durch genau den selben Prozess erzeugt. Diffuse Lichtstreuung dominiert auf Oberflächen, die winzige Unebenheiten aufweisen. Das Licht wird in viele verschiedene Richtungen von jedem Punkt der Oberfläche mit kleinen Winkeländerungen reflektiert (unter Umständen in Abhängigkeit der Wellenlänge, aber das kann man in Blender noch nicht simulieren).

Glanzlichter auf der anderen Seite dominieren auf glatten Oberflächen. Das heisst, dass die zerstreuten Strahlen eines jeden Punktes der Oberfläche fast alle in die selbe Richtung reflektiert werden, statt, wie bei der Diffusion in alle Richtungen. Wobei das eigentlich nur eine Frage des Maßstabs ist. Wenn die Unebenheiten der Oberfläche kleiner als die Wellenlänge des einfallenden Lichtes sind, erscheint sie flach und fungiert als Spiegel.

Da Glanzlichter nur ein Fake eines ziemlich komplexen physikalischen Vorganges sind, sollte man nicht allzu viele Erwartungen an ihre Realitätsnähe stellen. Erfreulicherweise ist das aber nicht so schlimm, da es nicht auffällt. Zum Beispiel sind Glanzlichter in Wirklichkeit nicht nur rund, sondern entsprechen der gespiegelten Form der Lichtquelle (des Fensters, der Neonröhre, ...). Da die Sonne und Glühbirnen aber rund sind, passt das runde Glanzlicht für viele Beleuchtungssituationen. Echte Glanzlichter ändern ihre Stärke außerdem bei transparenten Gegenständen unter sich ändernden Blickwinkeln, zeigen also auch den "Fresnel-Effekt".

Wie die Diffusion, haben auch die Glanzlichter eine ganze Reihe von Implementierungen oder "Shadern". Wieder teilt jede dieser Implementierungen zwei Standardparameter: die Glanzfarbe und die Stärke des Glanzes, in einem

Bereich von 0 bis 2. Dies erlaubt es einem, dem Glanz mehr Energie zuzuweisen, als eigentlich einfällt. So hat ein Material mindestens zwei verschiedene Farben, die von der Diffusion und die vom Glanz. Die Glanzfarbe steht standardmässig auf Weiß, was allerdings geändert werden kann, um interessante Effekte zu erhalten.

Die fünf Glanzshader sind:

- *CookTorr* - Das war Blenders einziger Glanzshader bis zur Version 2.27. Ausserdem war es bis zu dieser Version nicht möglich den diffusen Shader und den Glanzshader getrennt von einander einzustellen, es gab nur eine einzige Materialimplementation. Neben den zwei Standardparametern benutzt dieser Shader noch einen dritten, bezeichnet mit *hardness* (Härte), mit welchem sich der Durchmesser der Spitzlicher einstellen lässt. Je tiefer die *hardness* Einstellung, um so breiter die Lichter.
- *Phong* - Das ist ein anderer mathematischer Algorithmus zur Berechnung von Glanzlichtern. Er unterscheidet sich nicht sehr stark vom CookTorr und wird von den selben Parametern gesteuert.
- *Blinn* - Dies ist - eher - ein realitätsnaher *Shader*, der gut zu dem diffusen Oren-Nayar *Shader* passt. Er wirkt echter wegen seines vierten Parameters, dem *index of refraction* (IOR zu deutsch: Brechungsindex) zusätzlich zu den drei oben erwähnten. Dieser Parameter wird nicht für die Berechnung der Strahlenbrechung benutzt (hierfür wird ein Raytracer benötigt), sondern um die Intensität und Ausdehnung von Glanzlichtern mittels des Snell Gesetzes zu berechnen. Härte- und Glanz-Parameter geben hier einen zusätzlichen Grad von Freiheit.

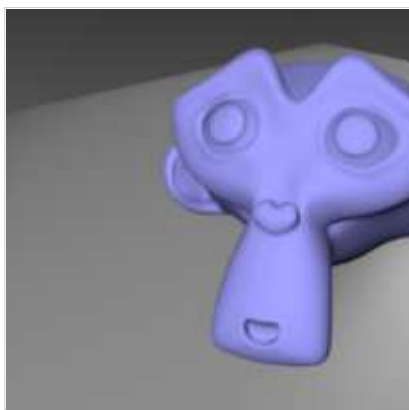
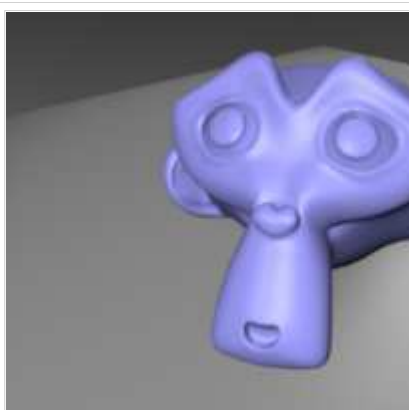
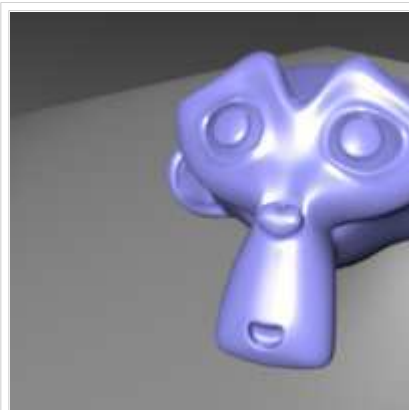


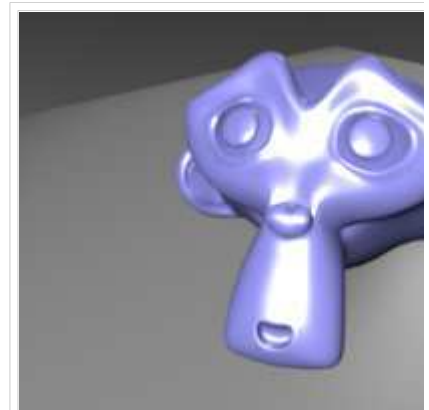
Abbildung 4: *Spec*: 0.0. Keine C



*Spec*: 0.5, *Hard*: 50, *Refr*: 2.0. Kleine sehr weiche Glanzlichter. Die *Hardne* großflächig.

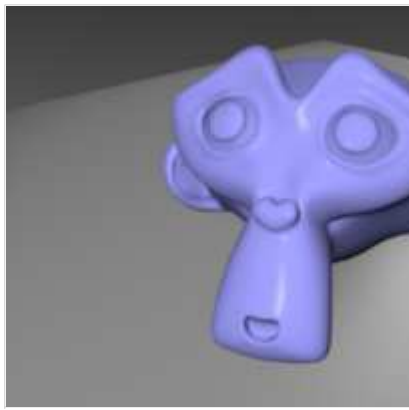


*Spec*: 0.5, *Hard*: 50, *Refr*: 4.0. Kle großflächige Glanzlicht

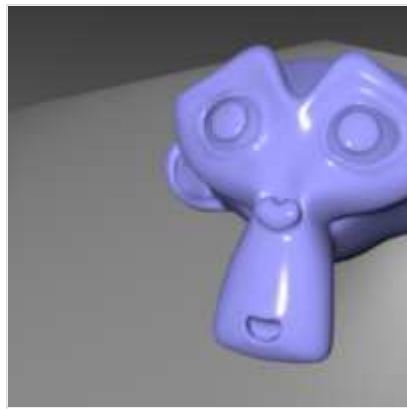


*Spec*: 0.5, *Hard*: 50, *Refr*: 8.0. Große scharfrandige Glanzlichter mit ein Intensität.

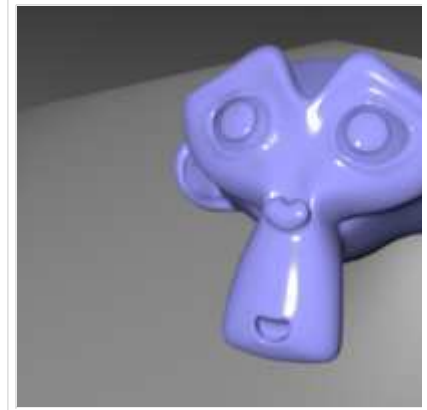




*Spec: 0.5, Hard: 260, Refr:*



*Spec: 0.5, Hard: 260, Refr: 4.0. Große (spitze) Glanzlichter.*



*Spec: 0.5, Hard: 260, Refr: 8.0. Große Refr Parameter: Kleine, scharf begrenzte Glanzlichter.*

- *Toon* - Dieser *Shader* ist das Gegenstück zum diffusen mit selbem Namen. Er wurde entworfen um scharfe, gleichmäßige Glanzlichter, wie im Zeichentrick üblich, zu produzieren. Er hat keinen Härteparameter, dafür aber einen für *Size* (Größe) und einen für *Smooth* (Feinheit), mit denen sich die Ausdehnung und die Schärfe der Glanzlichter einstellen lässt.
- *WardIso* - "Ward isotropic gaussian specular shader", besonders für Plastik geeignet.

Die Implementierung in Blender trennt diffuse Reflexion und Glanzlichter. Daher kann sehr einfach kontrolliert werden, wieviel des einfallenden Lichtes diffus reflektiert, als Glanzlicht reflektiert oder absorbiert wird.



### Farbe

Die Materialfarbe ist nur ein Element während des Renderns. Die letztendliche Farbe setzt sich aus der Material- und Lichtfarbe zusammen. Die Spec-Farbe ist bei Nicht-Metallen weiß, d.h. Nicht-Metalle ändern die Lichtfarbe bei gerichteter Reflexion nicht.

Metalle dagegen ändern die Lichtfarbe bei gerichteter Reflexion, das gilt für die *Spec* Farbe wie auch für *RayMirror*.



## Links

Die englischsprachige Vorlage dieses Textes (<http://www.blender.org/modules/documentation/html/x4251.html>)

## Materialeinstellungen

*Diese Seite bezieht sich auf  
Blender v2.42*

In diesem Abschnitt betrachten wir die verschiedenen Materialeinstellungen in Blender, und was diese bewirken.

Wählen Sie zunächst ein Objekt aus. Durch drücken von **F5** oder klicken auf den Button , wechseln Sie in den *Shading* Kontext. Das *Material Buttons* Fenster erscheint. Solange das Objekt kein Material besitzt, ist das Fenster sehr übersichtlich. Fügen Sie ein neues Material mit dem Button *Add New* hinzu (**Abbildung 1**). Sie können selbstverständlich auch ein bereits vorhandenes Material auswählen .



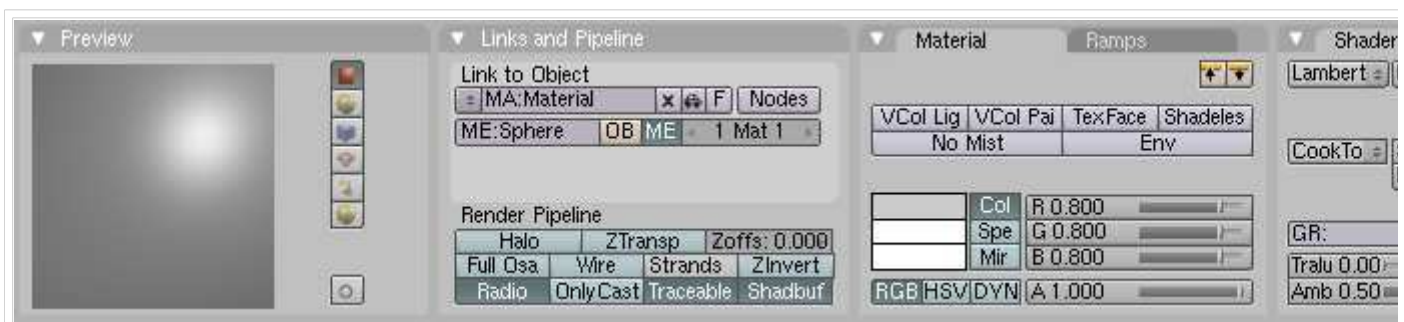
**Abbildung 1:** Ein neues Material hinzufügen

Normalerweise werden Materialien mit dem *Mesh (ME)* verknüpft, in diesem Fall mit dem Mesh "Sphere" (*ME:Sphere*). Sie können Sie aber auch mit dem Objekt selbst verknüpfen, wenn Sie den *OB* Button drücken. Das ist dann notwendig, wenn zwei Objekte das gleiche Mesh teilen, z.B. weil Sie sie mit **Alt-D** kopiert haben, aber die Meshes unterschiedliche Materialien tragen sollen.

Im folgenden gehen wir davon aus, dass Sie die Materialien mit dem Mesh verknüpfen (Normalfall).

*0 Mat 0* zeigt an, dass Sie zur Zeit noch kein Material mit dem Mesh verknüpft haben (erste Zahl). Mit mehreren Materialien pro Mesh beschäftigen wir uns im Abschnitt *Multimaterial*.

Nachdem Sie ein Material hinzugefügt haben, befinden sich in den *Material Buttons 7* Panel und jede Menge Auswahlmöglichkeiten. Von links nach rechts sind das *Preview*, *Links and Pipeline*, *Material*, *Ramps* (siehe Abschnitt *Ramp Shader*), *Shaders*, *Mirror Transp* und *Texture* (siehe Abschnitt *Texturen*) (**Abbildung 2**, der *Texture* Panel ist nicht gezeigt).



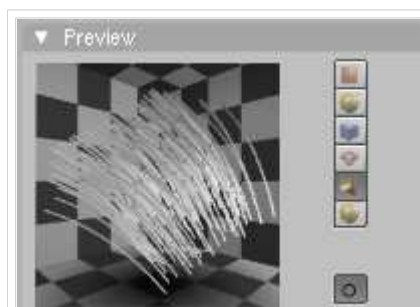
**Abbildung 2:** Übersicht über die *Material Buttons*, der *Texture* Panel ist nicht gezeigt.

## Preview

Das *Preview* Panel zeigt die Material Vorschau. Es handelt sich dabei um ein echtes Rendering, kann sich also bei sehr komplexen Materialien etwas verzögern. Wählen Sie dann die Ebene als Vorschauobjekt aus, oder minimieren das *Preview* Panel. In der Voreinstellung zeigt das *Preview* Panel eine Ebene senkrecht von oben (nur XY-Koordinaten), Sie können aber auch Kugel, Würfel, Suzanne, statische Partikel oder eine Kugel vor einem Hintergrund einstellen (**Abbildung 3**).



**Abbildung 3a:** *Preview* Panel mit Suzanne.



**Abbildung 3b:** *Preview* Panel mit statischen Partikeln (*Strand*).

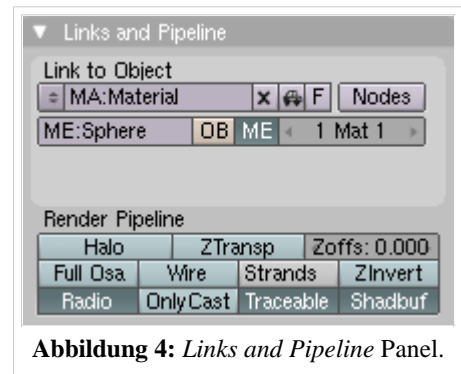


**Abbildung 3c:** *Preview* Panel Kugel mit Himmel.

Der unterste Button stellt das *Antialiasing* für den *Preview* an oder ab.

## Links and Pipeline


- **MA:** Setzen Sie hier in der obersten Zeile den Namen des Materials (**Shift-LMT** in das Feld).
- **X:** löscht das Material. Wenn das Material mehrfach genutzt wird (also für mehrere Objekte oder Meshes), steht dort die Anzahl der Nutzer. Dann können Sie durch Klicken auf diesen Button auch *Single User* Kopien anfertigen.
- **"Auto:"** erstellt einen möglichst beschreibenden Namen für das Material (z.B. in Abhängigkeit der Materialfarbe).
- **F:** erzeugt einen *Fake* Benutzer für das Material. Dann wird das Material auch gespeichert, wenn es sonst von keinem Objekt mehr benutzt wird. Normalerweise wird ein Material nicht gespeichert, wenn es nicht mehr benutzt wird. Einen eigentlichen Löschvorgang für Materialien gibt es nicht.
- **Nodes:** Macht das Material zu einem *Material-Node-Tree*. Material-Nodes dienen dazu, hochkomplexe Materialien und bestimmte Effekte zu erzeugen. So können Sie mehrere vollständige Materialien übereinanderlegen. Sie werden sie in der Regel nur für klar bestimmte Zwecke benutzen (Siehe: Material Nodes).



Im Abschnitt *Render Pipeline* finden Sie die Einstellungen, die man auch für einen Node-Tree nur im Ganzen einstellen kann. Für alle normalen Materialien werden Sie die Einstellungen hier voreingestellt lassen, bei durchsichtigen Materialien müssen Sie unter Umständen *ZTransp* anstellen.

- **Halo:** Rendert das Material als Halo. Siehe Abschnitt Halos und Lens Flares.
- **ZTransp:** Sie sehen hinter dem Objekt liegende, andere Objekte. Ohne *ZTransp* ist das Material zwar unter Umständen durchsichtig, Sie sehen dahinterliegende Objekte aber trotzdem nicht. Stattdessen sehen Sie den Hintergrund. *ZTransp* funktioniert ohne Raytracing, alternativ können Sie *RayTransp* auf dem *Mirror Transp* Panel einsetzen, das allerdings nur mit Raytracing.
- **Zoffs:** Der Abstand des Objektes zur Kamera wird künstlich verkürzt. Das ist z.B. nützlich, wenn zwei Oberflächen fast den gleichen Abstand zur Kamera haben, und die eine Oberfläche vor der anderen liegen soll.
- **Full OSA:** Vollständiges Antialiasing für Glanzlichter, Shading und Texturen. Bei Raytracing wird das sowieso benutzt, ob es hier angeschaltet ist oder nicht. Es ändert nicht den *OSA* Level, es verhindert nur, dass durch bestimmte Zeitoptimierungen nicht alle Samples berechnet werden. Siehe Release Notes zu Version 2.34, Rendering Engine ([http://www.blender3d.org/cms/Rendering\\_engine.320.0.html](http://www.blender3d.org/cms/Rendering_engine.320.0.html)).
- **Wire:** Rendert das Objekt als Drahtgittermodell.
- **Strands:** Optionen zum Rendern von statischen Partikeln (Haare, Fell) (siehe den Abschnitt Partikel Haare).
- **ZInvert:** Der ZBuffer wird invertiert. Das dreht Vorne und Hinten um.
- **Radio:** Wenn das Objekt in die Radiosity Berechnung mit einbezogen werden soll.
- **OnlyCast:** Das Objekt wird nicht gerendert und nicht gespiegelt, wirft aber Schatten.
- **Traceable:** Das Objekt wird beim Raytracing nicht berücksichtigt, wenn diese Option ausgestellt ist. Das bedeutet unter anderem:
  - Das Objekt wirft keine Raytracing Schatten mehr.
  - Es wird nicht mehr durch *Ray Mirror* gespiegelt.
  - Durch ein *RayTransp* transparentes Objekt ist es nicht zu sehen.
- **Shadbuf:** Das Objekt wirft keine Schatten bei *Buffer-Shadow*. Siehe auch die Abschnitte Raytracing Schatten und Buffer Schatten.

## Material

- Mit den Buttons CopyToBuffer/PasteFromBuffer  können Sie ein Material inklusive der gesamten Textureinstellungen in den Zwischenspeicher kopieren und aus dem Zwischenspeicher wieder einfügen.

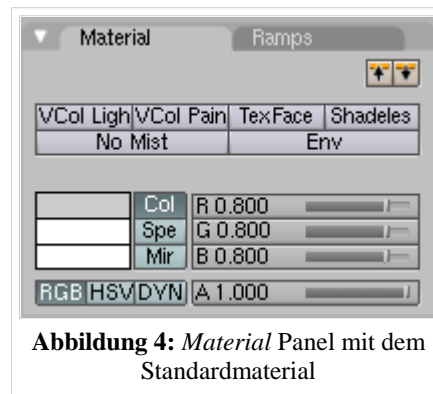


Abbildung 4: Material Panel mit dem Standardmaterial

#### Eigenschaften auf mehrere Objekte gleichzeitig kopieren:

Um die Materialeigenschaften gleichzeitig auf mehrere Objekte zu kopieren, werden alle Objekte selektiert, die das Zielmaterial erhalten sollen. Als Letztes wird zusätzlich das Objekt selektiert, das das Zielmaterial schon besitzt. Nach drücken von **Strg-L** öffnet sich der *Make Links* Dialog, in dem *Materials* ausgewählt wird. Damit erhalten alle selektierten Objekte das Material des zuletzt gewählten Objektes.

- *VCol Light*: Mit Vertexpaint können Sie den Vertices Farbe zuweisen. Diese kann als Lichtquelle benutzt werden. Wichtig für das Radiosity Modelling, da es für die Lösung angeschaltet sein muss.
- *VCol Paint*: Die Grundfarbe wird durch die Vertexfarbe ersetzt.
- *TexFace*: UV gemappte Texturen werden gerendert. Diese Option wird nicht empfohlen, benutzen Sie besser die Option auf dem Map Input Panel.
- *Shadeless*: Das Material reagiert nicht auf Licht oder Schatten. Nützlich z.B. für Projektionsflächen von Videos, damit diese ihre ursprüngliche Helligkeit behalten.
- *No Mist*: Das Material ist auch im Nebel gut zu sehen.
- *Env*: Wie *Alpha 0*, *ZTransp* aus. Der Welthintergrund ist zu sehen, aber nicht die Objekte hinter dem transparenten Objekt.

Als vielleicht wichtigste Einstellung wird hier die Farbe des Materials gesetzt. Dabei hat jedes Material drei unterschiedliche Farben:

- Die Grundfarbe oder *Diffuse* Farbe (*Col* Button). Das ist die Farbe, die vom *Diffuse* Shader benutzt wird.
- Die Glanzlichtfarbe (*Spec*). Diese Farbe wird vom *Specular* Shader benutzt.
- Die Spiegelfarbe (*Mir*). Diese Farbe wird insbesondere beim Environment Mapping und bei RayMirror benutzt.

Dabei können Sie zwischen dem *RGB* und dem *HSV* Farbschema wechseln.

Der *DYN* Button wird benutzt um bestimmte dynamische Eigenschaften des Objektes festzulegen, die z.B. in der Game-Engine benutzt werden. Das geht inhaltlich über den vorgesehenen Rahmen dieser Dokumentation hinaus, und soll hier nicht weiter besprochen werden.

- *A*: Alpha-Wert. Der Alpha-Wert ist die Sichtbarkeit/Lichtundurchlässigkeit des Objektes,  $A = 1$  heist also total lichtundurchlässig = vollständig sichtbar. Stellen Sie nur *Alpha* auf Null, scheint der Welthintergrund durch. Sie müssen zusätzlich *ZTransp* oder *RayTransp* (im *Ray Mirror* Panel) anstellen, damit das Material wirklich durchsichtig wird (siehe Raytracing Transparenz und Solide und hohle Glasgegenstände).  
Alternativ können Sie auch *Fresnel-Transparenz* einsetzen.

## Ramps

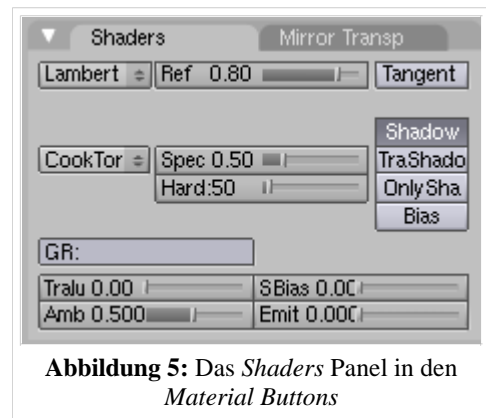
*Ramps* werden für besondere Materialeffekte, insbesondere Cell-Shader (Cartoon-Stil), Autolacke, Haut usw. eingesetzt. Das *Ramps* Panel wird auf der Seite Ramp Shader erläutert.

## Shaders

Benutzen Sie als Shader für normale Materialien die Kombination *Oren-Nayar* und *Blinn*.

Im *Shaders* Panel können Sie die verwendeten *Diffuse* und *Specular* Shader auswählen (**Abbildung 5**).

- *Ref*: Dieser Regler ist für alle *Diffuse* Shader von Bedeutung. Er gibt die Menge an diffus zurückgeworfenem Licht an. Das ist im wesentlichen die Helligkeit des Materials.
- *Spec*: Der *Spec* Wert gibt entsprechend die Menge an zurückgeworfenem Glanzlicht an. Die meisten Materialien haben kleine *Spec* Werte.
- *Hard*: Die Größe des *Spec* Bereiches. Je kleiner die *Hardness*, desto größer der *Spec* Bereich.



**Abbildung 5:** Das *Shaders* Panel in den *Material Buttons*

Die Shader *Toon* und *WardIso* haben keinen *Hard* Parameter.

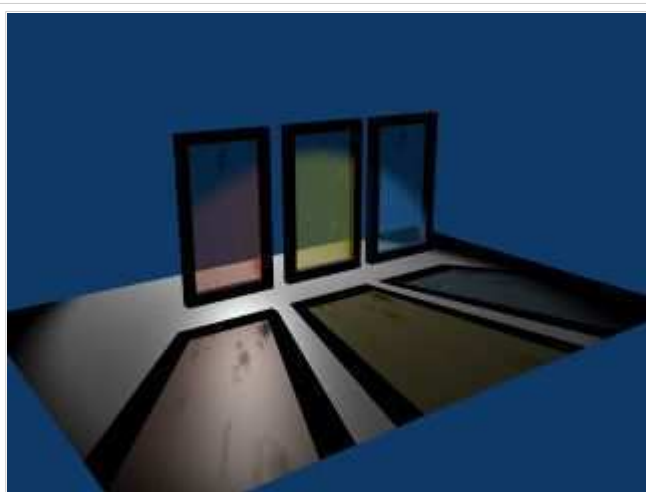
Physikalisch gesehen sollte  $Ref + Spec$  immer kleiner als 1 sein, sonst wirft das Objekt mehr Licht zurück als es empfängt, aber wir machen hier Computergrafik und keine Physiksimulation.

Je nach gewähltem Shader gibt es unter Umständen noch andere Regler. Beispiele, Bedeutung und Einsatz der verschiedenen Shader wurden bereits ausführlich in den Abschnitten *Diffuse Reflexion* und *Glanzlicher* besprochen.

- *Tangent*: Anisotroper Shader. Gut für den Effekt von gebürstetem Metall oder CDs. Das Objekt muss UV-Koordinaten besitzen! Siehe die Releasenotes zu Version 2.42 ([http://www.blender.org/cms/Material\\_Features.764.0.html](http://www.blender.org/cms/Material_Features.764.0.html)) für Beispiele.
- *Shadow*: Ist diese Option aus, empfängt das Objekt keine Schatten.
- *TraShadow*: Eine häufig übersehene Option und beliebte Fehlerquelle. Der Schatten auch von transparenten Objekten ist schwarz, wenn nicht für das **schattenempfangende** Objekt *TraShadow* angeschaltet ist.
- *Only Shadow*: Es wird nur der auf das Objekt fallende Schatten berechnet.

Zum Thema Schatten siehe auch die Abschnitte *Raytracing Schatten* und *Buffer Schatten*.

- *Bias*: Verhindert bestimmte Fehler bei der Schattenberechnung an der Schattengrenze. Siehe Release Notes zu v2.34, Misc Improvements ([http://www.blender3d.org/cms/Misc\\_improvements.355.0.html](http://www.blender3d.org/cms/Misc_improvements.355.0.html)) . Sie müssen *Bias* ausstellen, um *SBias* zu benutzen.
- *SBias*: Manuelle Einstellung des Bias, um Schattenfehler zu verringern. Wenn es trotz der Aktivierung des *Bias* Buttons zu Schattenfehlern kommt, müssen Sie *Bias* ausstellen und *SBias* einstellen.
- *GR*: Das Objekt wird nur von Lampen in der Gruppe *GR* beleuchtet.
- *Translucency*: Das Objekt leuchtet selbst, wenn es von hinten angestrahlt wird. Das ist ein Effekt, der besonders gut bei Milchglasscheiben oder staubigem Glas zu beobachten ist. Dünnes Papier oder Stoff zeigt ebenfalls *Translucency* (**Abbildungen 6 und 7**).



**Abbildung 6:** Glas mit *Translucency*



**Abbildung 7:** Glas ohne *Translucency*

- *Amb* (*Ambient*): Der Ambient Parameter regelt

1. Die Menge an Ambient Light die ein Material empfängt.
  2. Die Menge an Licht, die bei Ambient Occlusion vom Material empfangen wird.
  3. Die Menge an Licht, die beim Radiosity rendern vom Material empfangen wird.
- *Emit*: Das Objekt wird selbstleuchtend. Es beleuchtet aber keine anderen Objekte, außer bei der Radiosity Berechnung.

## Mirror Transp(arency)

Hier werden die Einstellungen vorgenommen, die mit Raytracing Spiegelungen (blau umrandet) und Raytracing Transparenz (braun umrandet) zusammenhängen. Sie finden die Erläuterungen hierzu auf den Seiten Raytracing Spiegelungen (*Ray Mirror*) und Raytracing Transparenz (*Ray Transp*). Raytracing Spiegelungen sind eine einfache, aber rechenintensive Möglichkeit realistische Spiegelungen der Umgebung zu erzeugen. Raytracing Transparenz ermöglicht transparente Objekte, die das Licht realistisch brechen.

Die gelb umrandeten Optionen funktionieren auch ohne Raytracing.

- *Fresnel*: Dies ist die zweite Möglichkeit (neben dem *Alpha* Wert) Objekte durchsichtig zu machen. *Fresnel* berücksichtigt dabei den Blickwinkel. Physikalisch am "richtigsten" ist ein *Fresnel* Wert von 5. Schauen Sie senkrecht auf ein Objekt, ist es maximal durchsichtig. Je schräger man auf ein Objekt blickt, desto weniger durchsichtig wird es. Ein Beispiel finden Sie unter Blickwinkel abhängige Texturen, die genaue mathematische Erläuterung unter Raytracing Spiegelungen. *Fresnel* funktioniert nur bei aktiviertem *ZTransp* oder *Ray Transp*.
- *Fac*: Die Stärke des *Fresnel* Effektes. *Fac* muss größer als 1 sein, sollte in der Regel aber nur wenig über 1 liegen.

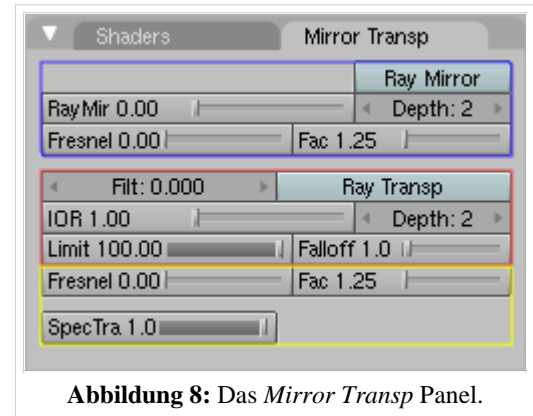


Abbildung 8: Das *Mirror Transp* Panel.

Zur Demonstration der verschiedenen Transparenzeinstellungen einige Bilder (alle ohne Schatten).

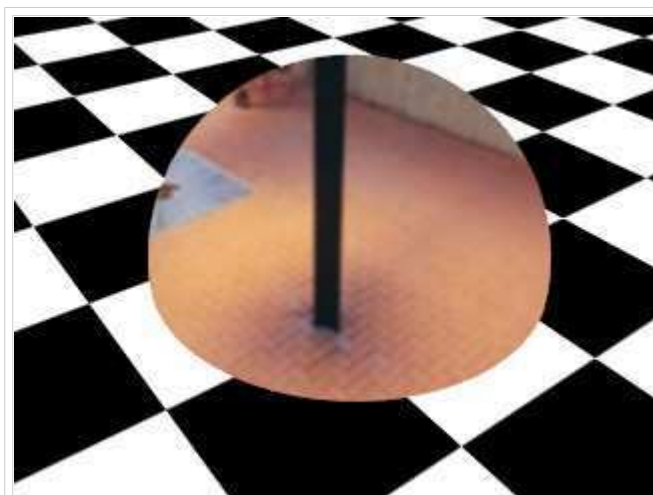


Abbildung 9: *Alpha* ist null, aber *ZTransp* ist aus. Der Welthintergrund wird sichtbar. Die karierte Ebene schneidet die Halbkugel zwar, ist aber trotzdem nicht sichtbar.

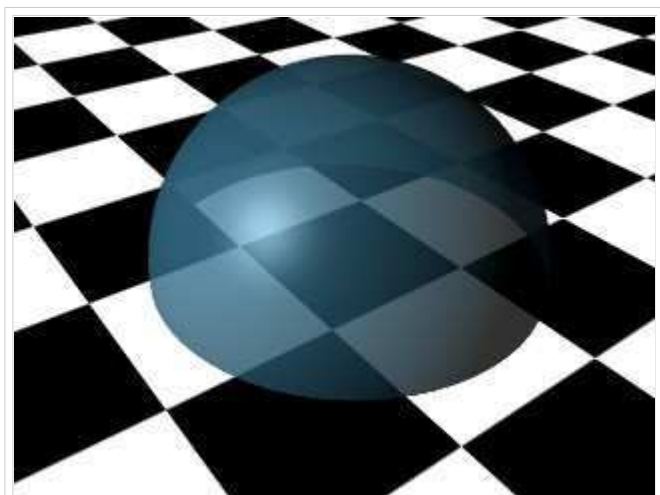
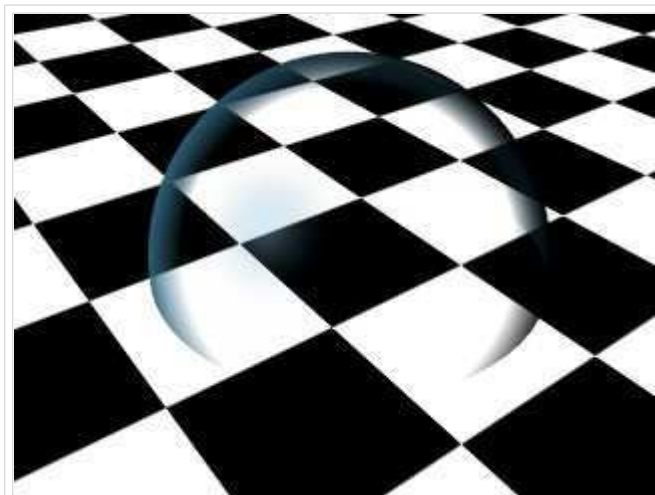
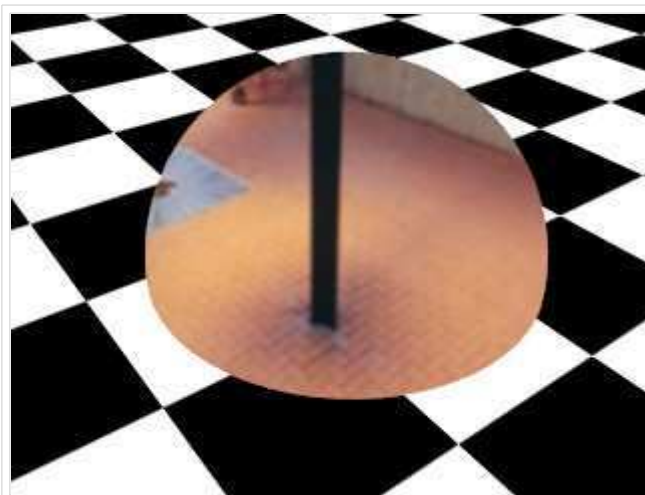


Abbildung 10: *Alpha* und *ZTransp*. *Alpha* musste ich höher setzen, damit man die Halbkugel überhaupt sieht. *ZTransp* ist jetzt angeschaltet. Das hintere Objekt (die Ebene) wird sichtbar.

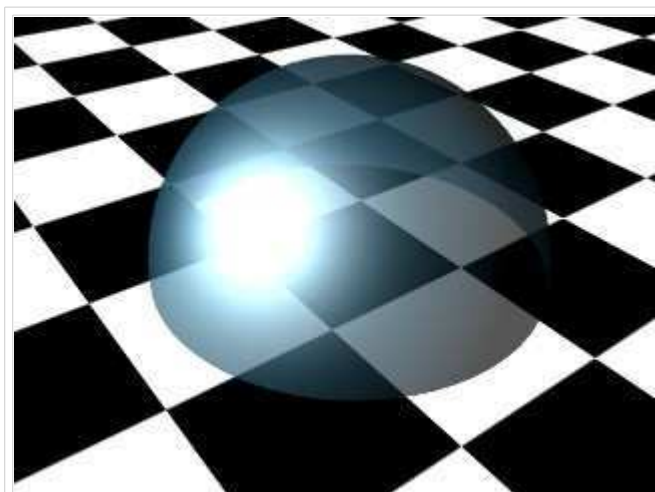


**Abbildung 11:** *Fresnel* und *ZTransp*. *Fresnel* ist 5, *Fac* ist 1.13. *Alpha* ist 1.

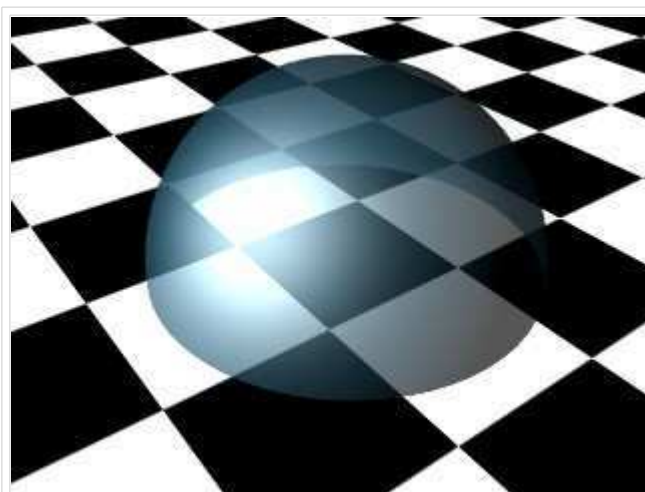


**Abbildung 12:** *Env* an, *ZTransp* aus. Das ist identisch zu *Alpha* 0, *ZTransp* aus.

- *SpecTra*: Gibt an, wie undurchsichtig die Glanzlichter sein sollen. Ist *SpecTra* 1, ist das Glanzlicht undurchsichtig. Das ist physikalisch auch ziemlich richtig, da ein Glanzlicht eine Spiegelung einer Lichtquelle ist. Transparente Objekte spiegeln an den Stellen, an denen sie undurchsichtig sind, sie sind durchsichtig an den Stellen, an denen sie nicht spiegeln. Ob der Wert von 1 stimmt hängt davon ab, wie groß das Glanzlicht ist und wie weit die Lichtquelle entfernt ist. Aber zum Glück lassen wir uns bei den Glanzlichtern leicht täuschen, so dass das Bild nicht immer physikalisch korrekt sein muss.



**Abbildung 13:** *SpecTra* ist 1. Die Stelle mit dem Glanzlicht ist undurchsichtig.



**Abbildung 14:** *SpecTra* ist 0. Die Stelle mit dem Glanzlicht ist durchsichtig.

## Links

Die Kombinationsmöglichkeiten der Materialeinstellungen sind fast unendlich. Es gibt aber eine ganze Reihe von Materialsammlungen für Blender. Einige sind hier aufgelistet. Teilweise sind die angewendeten Techniken allerdings erst verständlich, wenn Sie den gesamten Abschnitt über Material und Texturen gelesen haben. Im Abschnitt Append wird erklärt, wie Sie diese Materialien für Ihre eigenen Objekte benutzen.

Wenn Sie selbst besonders gelungene Beispiele kennen, verlinken Sie diese bitte hier.

Blender Materials Database ([http://www.geocities.com/pollythesheep/matlib\\_index.html](http://www.geocities.com/pollythesheep/matlib_index.html))

Zur Zeit 83 Materialien. Metall, Stein, Holz, Glas, Plastik, Planeten

Release Notes zu prozeduralen Texturen ([http://www.blender3d.org/cms/Procedurals\\_Gallery.279.0.html](http://www.blender3d.org/cms/Procedurals_Gallery.279.0.html))

Sehr überzeugende Metalle (Gold, Silber, Kupfer, Bronze), Leder, Beton, Orangenschale, polierter Stein, ...

Skin und Sketch Shader (<http://www.luceri.it/DXCOL/downloads/blender%20material.htm>)  
Simulation von SubSurface Scattering mit Hilfe von Ramp Shadern. Außerdem Simulation von Handzeichnungen.

Sonix' Blender 2.34 onwards car material library  
(<http://www.free-web-space.biz/sonix/Cars/Blender234CarMaterialLibraryR1.html>)  
Lack, Glas, Chrom, Reifen ...

Die englischsprachige Vorlage dieses Textes (<http://www.blender.org/modules/documentation/html/x4288.html>)

Von „[http://de.wikibooks.org/wiki/Blender\\_Dokumentation/\\_Druckversion/\\_Material2](http://de.wikibooks.org/wiki/Blender_Dokumentation/_Druckversion/_Material2)“

---

- Diese Seite wurde zuletzt am 12. Oktober 2006 um 21:08 Uhr geändert.
- Inhalt ist verfügbar unter der GNU Free Documentation License.
- Datenschutz
- Über Wikibooks
- Haftungsausschluss



# Blender Dokumentation/ Druckversion/ Material3

Aus Wikibooks

< Blender Dokumentation

## Ramp Shader

bezieht sich auf Blender v2.37a

Sehr viele Materialien - wie Haut oder Metall - zeigen Farbvariationen in Abhängigkeit der Lichtmenge die die Oberfläche trifft, oder der Richtung des Lichteinfalls. Diese Effekte lassen sich mit Hilfe von *Ramp Shadern* erreichen. Das Besondere an den *Ramp Shadern* und der Schlüssel zu ihrem Verständnis sind die *Input Methoden*!

Mit Hilfe der *Ramp* Optionen in Blender kann ein Farbverlauf (inklusive Transparenz) für das Material definiert werden. Dann wird ausgewählt, von welchem Parameter der Verlauf abhängen soll, und wie er mit der Materialfarbe und den Texturen gemischt werden soll.

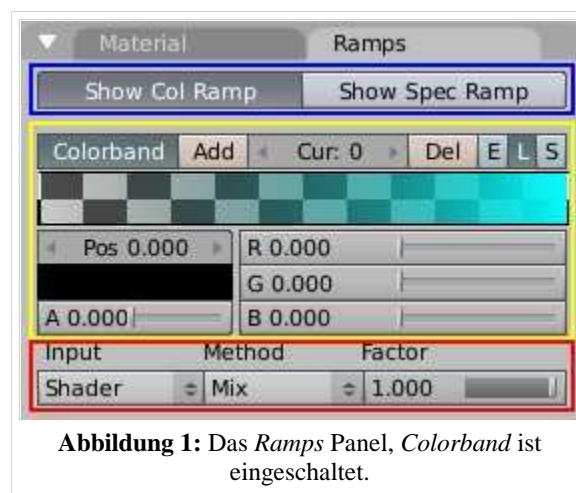
Da die Texturen vor dem Shading aufgebracht werden, können die Farbverläufe die Texturen vollständig ersetzen, oder mit ihnen verrechnet werden. Das geschieht zum einen über Alpha-Werte der Textur selbst, über die Blending Modes und über den Blendfaktor. *Ramp Shader* sind also eine zusätzliche Möglichkeit, das Material des Objektes zu beeinflussen. Alle anderen Materialoptionen bleiben erhalten!

*Ramp Shader* sind ein wirklich tolles Feature, mit dem Sie Materialien sehr viel realistischer gestalten, oder besondere Effekte erzielen können (z.B. Cartoon). Darüberhinaus beeinflusst es die Renderzeit nur sehr wenig.

## Anschalten

Wechseln Sie mit **F5** in die *Material buttons* und öffnen das *Ramps* Panel. Das sieht noch sehr leer aus. Klicken Sie auf den Button *Colorband*. Dadurch werden die *Ramp Shader* eingeschaltet, und eine Reihe von weiteren Optionen und Buttons wird sichtbar (**Abbildung 1**).

Die Auswirkungen der *Ramp Shader* können Sie sehr gut im *Preview* Fenster und ebenfalls im 3D-Fenster beurteilen. Wechseln Sie dazu am besten in die Kameraansicht (**Num-0**) und schalten mit **Shift-Z** in die *Shaded* Ansicht.



**Abbildung 1:** Das *Ramps* Panel, *Colorband* ist eingeschaltet.

## Farbverlaufseditor [Colorband]

Das Colorband ist Blenders Farbverlaufs-Editor (**Abbildung 1**, der gelb umrandete Teil). Jeder festgelegte Wert - genauer Wertetupel, aber wir bleiben hier bei der Bezeichnung Wert - im Farbverlauf wird durch einen senkrechten Strich symbolisiert. Der aktive Strich wird dicker dargestellt.

Jeder Strich auf dem Farbband kann an jeden Ort verschoben werden (es können damit auch die Positionen gewechselt werden), jeder Alpha- und Farbwert kann ihm zugewiesen werden. Blender interpoliert die Werte von einem Strich zum nächsten. Der Interpolationstyp wird durch die Buttons E (quadratisch), L (linear) und S (B-Spline) ausgewählt. Die Interpolation ist nicht im Colorband sichtbar, wohl aber in der Vorschau und im 3D-Fenster.

Die Striche (Werte) werden mit dem *Cur*: Button oder mit **LMT** ausgewählt. *Add* und *Del* fügen Werte hinzu bzw. löschen diese, dabei beginnt die Zählung ganz links bei 0. Es sind max. 15 Werte (0-14) möglich. Den Wert mit der Nummer "0" können Sie nicht löschen, aber natürlich verschieben. Neue Werte werden immer in der Mitte (bei 0.5) hinzugefügt.

Im Feld *Pos* wird die Position des aktuellen Wertes auf dem Band angezeigt, darunter dessen Farbe und Alphawert. Der Alphawert bestimmt nicht den Alphawert des Materials, macht also nicht einen Gegenstand durchsichtiger, sondern bestimmt die Stärke des *Ramp Shader* Effektes.

Die senkrechten Striche können mit "Drag and Drop" auf dem Band, oder durch Eingabe des numerischen Wertes in das *Pos* Feld positioniert werden.

## Input

Für den *Input* stehen vier Möglichkeiten zur Verfügung:

- *Shader*: Die Richtung des Lichteinfalles bestimmt die Farbe. Dabei ist die Position ganz rechts im *Colorband* senkrecht (90°) auftreffendes Licht, die Position in der Mitte waagrecht (streifend) einfallendes Licht, die Position ganz links wäre dann -90°. Um diese Farbe sehen zu können, müssen Sie mehrere Lampen in der Szene haben. Die Helligkeit wird natürlich weiterhin von der Menge des Lichtes und dem *Ref*- bzw. *Spec* Wert des Materials festgelegt.
- *Energy*: Sowohl die Richtung als auch die Lichtenergie wird berücksichtigt. Je dunkler die Lampe ist, desto mehr wird der linke Teil des Farbverlaufs berücksichtigt, je heller, desto stärker der rechte Teil. Die Lichtmenge ändert also die Farbe des Materials.
- *Normal*: Die Flächennormale (relativ zur Kamera) wird benutzt. Ganz rechts im Verlauf entspricht der Richtung senkrecht zur Kamera, ganz links im Verlauf von der Kamera weg. Damit werden Sie durch die Kamera die Farbe ganz links im Verlauf niemals sehen können. Dieser Effekt ist auch mit Texturen möglich (Abschnitt Map Input), wurde aber der Vollständigkeit halber hier hinzugefügt. Es wird vielfach empfohlen, statt diesem *Ramp Shader* die entsprechende Textur zu benutzen, nichts desto trotz findet man viele komplexere Materialien mit genau dieser Einstellung.
- *Result*: Die Gesamthelligkeit aller Lampen (die anderen Optionen arbeiten pro Lampe). Je dunkler die Lampenhelligkeit, desto stärker wird der linke Wert im Farbverlauf berücksichtigt, je heller, desto mehr der rechte Wert. Die Helligkeit hängt nicht mehr von den Lampen, sondern nur von den Farbeinstellungen ab. Damit kann man "Cartoon" Shader, oder farbige Schatten erzeugen.

Yafray in Version 0.0.8 ignoriert die Option *Result*, die anderen drei Optionen funktionieren, wenn auch nicht ganz so, wie in Blender.

## Method / Factor

Die *Method* bestimmt, wie der *Ramp Shader* mit den Materialien und Texturen verrechnet wird. Das entspricht den Ebenenmodi aus Bildbearbeitungsprogrammen. *Mix* bedeutet Ersetzung. Eine genauere Erklärung der Modi finden Sie im Abschnitt Blending Modes.

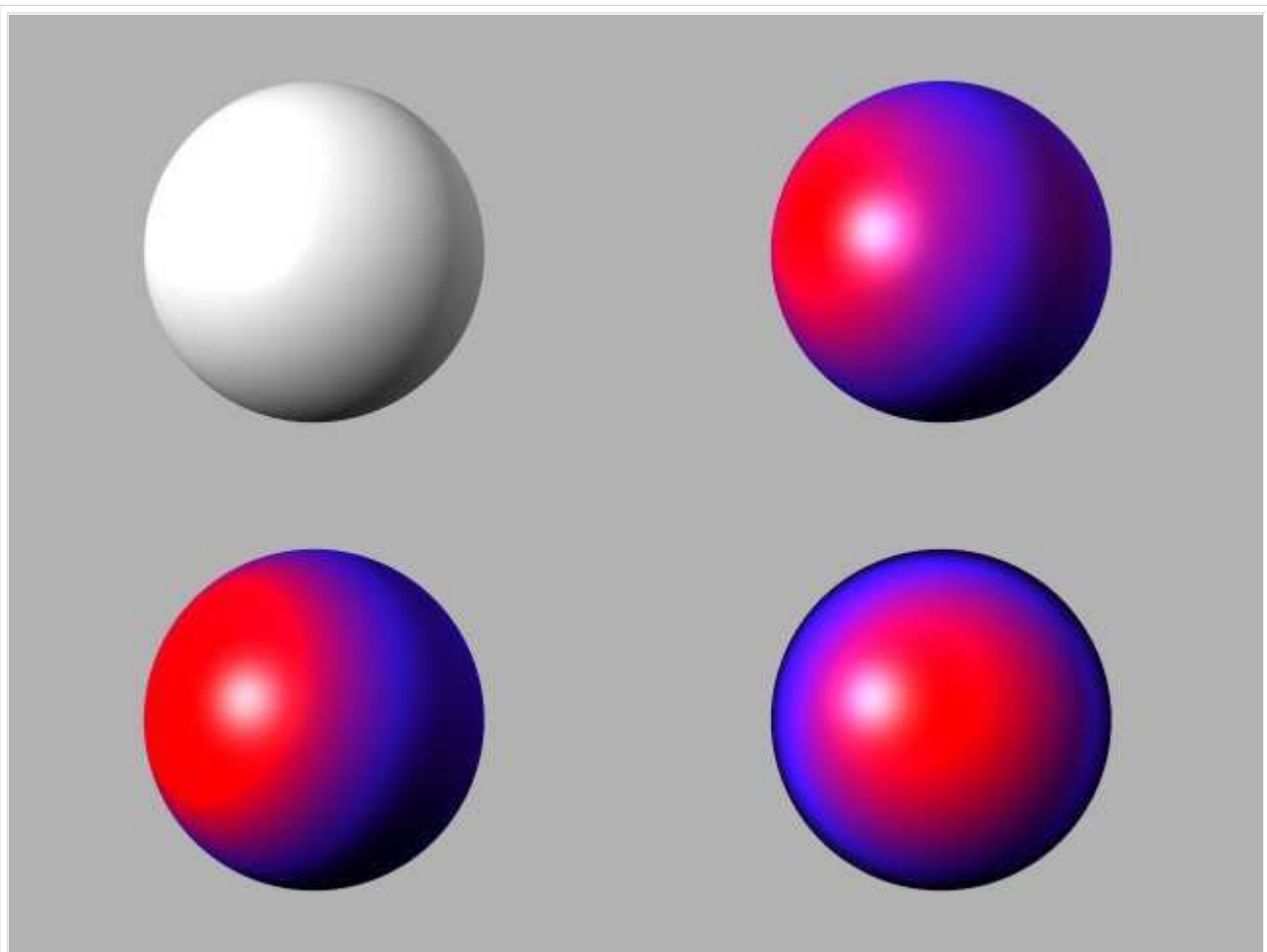
Der *Factor* gibt an, wie stark der Effekt der *Ramp Shader* im Verhältnis zu dem Material und den Texturen sein soll. Ein *Factor* von 1 im *Mix* Modus zeigt nur den *Ramp Shader*.

## Diffuse

*Ramp Shader* können den Diffuse (*Ref*) und/oder den Specular Shader (*Spec*) beeinflussen. Wir werden hier nur verschiedene Einstellungen für *Diffuse* vorstellen.



**Abbildung 2:** Ramp Shader Einstellungen für die Beispiele in Abbildung 3.



**Abbildung 3:** Von links nach rechts und oben nach unten: Ausgangskugel, Shader, Energy, Normal

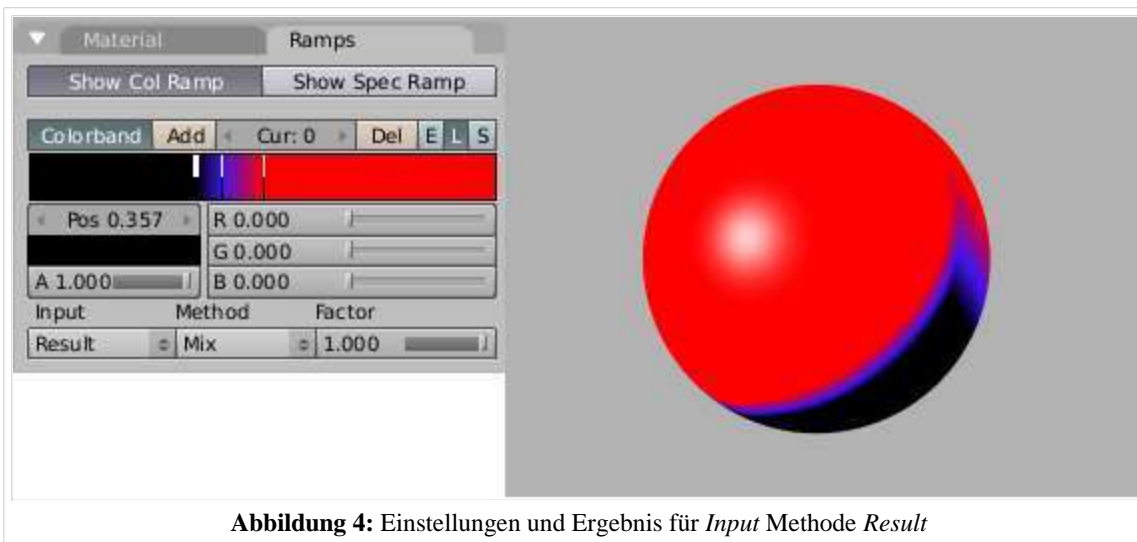


Abbildung 4: Einstellungen und Ergebnis für *Input Methode Result*

## Links

Release Notes zu v2.34 (Ramp Shaders) ([http://www.blender3d.org/cms/Ramp\\_Shaders.348.0.html](http://www.blender3d.org/cms/Ramp_Shaders.348.0.html))  
 Videotutorial zu Ramp Shaders und Colorband (engl.)  
 ([http://www.blender3d.org/cms/Model\\_Material\\_Light.397.0.html](http://www.blender3d.org/cms/Model_Material_Light.397.0.html))  
 Skin- und Toonshader mit Ramps (<http://www.luceri.it/>)

## Raytracing-Spiegelungen

bezieht sich auf Blender v2.37a

Raytracing Spiegelungen sind eine Methode, um in Blender realistisch spiegelndes Material zu erstellen. Wie jede Raytracing Methode erfordern sie das Aktivieren der Option *Ray* in den Render Buttons (**F10**). Und genauso wie bei den anderen Raytracing Berechnungen kann die Renderzeit unter Umständen sehr lang werden.

Die verschiedenen Einstellungsmöglichkeiten finden Sie in den *Material Buttons* im Panel *Mirror Transp* (Abbildung 1).

- *Ray Mirror*: stellt die Raytracing Spiegelungen an oder ab.
- *RayMir*: gerichtete Reflektivität (Spiegelung) des Materials. Hier gilt: Einfallswinkel gleich Ausfallswinkel. *RayMir*=1 bedeutet ideal spiegelnd, *RayMir*=0 entsprechend überhaupt nicht spiegelnd.
- *Depth*: Wie oft ein gespiegelter Lichtstrahl wieder gespiegelt wird (erfordert mehrere spiegelnde Oberflächen). Haben sie nur einen geraden Spiegel, funktioniert auch *Depth*=0. Ein großer *Depth* Wert treibt bei mehreren spiegelnden Objekten die Renderzeit sehr in die Höhe, ein *Depth* Wert von 2 oder 3 genügt in der Regel.
- *Fresnel*: Blickwinkelabhängigkeit der Reflektion. *Fresnel*=0 bedeutet keine Blickwinkelabhängigkeit. Bei größeren *Fresnel* Werten ist die Spiegelung bei senkrechter Betrachtung kleiner, bei schräger Betrachtung größer. Eine genauere Besprechung des *Fresnel* Wertes folgt weiter unten.
- *Fac*: Einflussfaktor des *Fresnel* Wertes. *Fac*=1 schaltet *Fresnel* aus. Wenn *Fresnel* und *Fac* auf 5 stehen, spiegelt das Material überhaupt nicht (aus keiner Richtung).

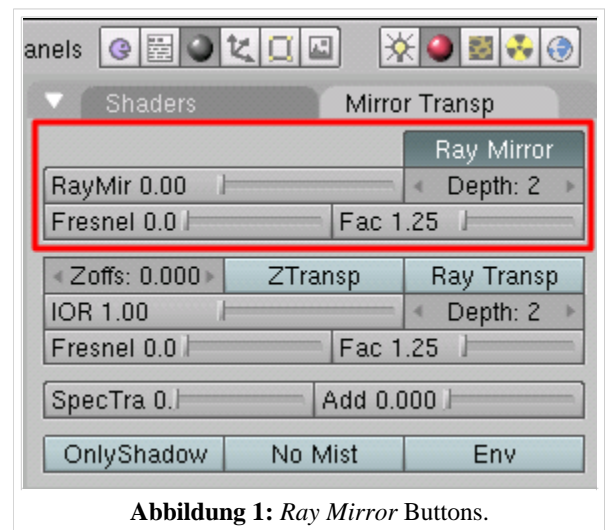


Abbildung 1: *Ray Mirror* Buttons.



Die Farbe des reflektierten Lichts wird über die *Mir* Einstellungen im *Material* Panel eingestellt.

Wird *Traceable* deaktiviert, wird das Material nicht in einem Spiegel reflektiert, reflektiert aber selbst (Vampir Effekt ;-)).

## Fresnel

Wenn man auf eine Wasseroberfläche schaut, bemerkt man dass mit zunehmendem Blickwinkel - also je schräger man schaut - das Wasser undurchsichtig wird. Stattdessen reflektiert das Wasser nur noch. Blickt man von oben senkrecht in das Wasser hinein, so reflektiert es nur sehr wenig, ist dafür aber sehr durchsichtig. Das nennt man den *Fresnel* Effekt (gesprochen "Fre-nel"). Der *Fresnel* Effekt ist für Metalle sehr klein und kann praktisch vernachlässigt werden (sofern die Metalle ganz sauber sind). Glas und Wasser dagegen zeigen einen sehr starken *Fresnel* Effekt.

Der *Fresnel* Wert ist daher für transparente Materialien von besonderer Bedeutung, er spielt aber auch eine Rolle für nur teilweise transparentes Material, wie z.B. Plastik. Außerdem wird man viele Materialien (z.B. Autolack) in Blender so behandeln, als bestünden sie nur aus einem einheitlichen Material, gerade Lack aber besteht aus mehreren Schichten mit sehr unterschiedlichen Eigenschaften. Die Physik dahinter exakt zu simulieren ist mit vertretbarer Renderzeit vermutlich nicht möglich. Letztlich wird es also immer auf den zu erzielenden Effekt ankommen, welche Einstellungen von *Ray Mirror*, *Fresnel* und *Fac* man wählt.

## Spiegel

Metalloberflächen zeigen praktisch keine Abhängigkeit der Reflexion von der Blickrichtung, der *Fresnel/Fac*-Wert für ideale Spiegel ist demnach 0.0/1.0. Da Spiegel an der Vorderseite aber eine Glasscheibe haben, oder die Metalloberfläche dreckig sein kann, kann man auch kleine Werte (0.1/1.1) wählen, z.B. mit einer Wolkentextur für die Spiegeloberfläche.

Bei Aluminium ist es z.B. so, dass das Reflexionsvermögen für unpolarisiertes Licht erst ab einem Betrachtungswinkel von 70° etwa um 5% abfällt, um dann ab 85° wieder anzusteigen. Bei fast 90° Einfall wird das Licht dann zu fast 100% reflektiert. Man wird diesen Effekt also ignorieren können.

## Glas

Der reflektierte Intensitätsanteil beträgt für den Übergang von Luft zu Glas bei senkrechter Betrachtung etwa 4%. Bis zu einem Winkel von 50° bleibt die Reflexion für unpolarisiertes Licht auch etwa so. Nun ändert sich die Winkelabhängigkeit aber dramatisch.

In ungefährender Parabelform steigt die Reflexion auf 100% bei einem Betrachtungswinkel von 90° (siehe unten).

## Wasser

Der reflektierte Intensitätsanteil beträgt für den Übergang von Wasser zu Glas bei senkrechter Betrachtung etwa 2%.

Am "physikalischsten" ist die Einstellung beider *Fresnel*-Werte auf fünf, insbesondere sollten sie gleich groß sein.



Wenn nichts zum Spiegeln da ist, können Sie *Ray Mirror* einstellen wie sie wollen. Sorgen Sie also für eine interessante Umgebung!

## Etwas Physik

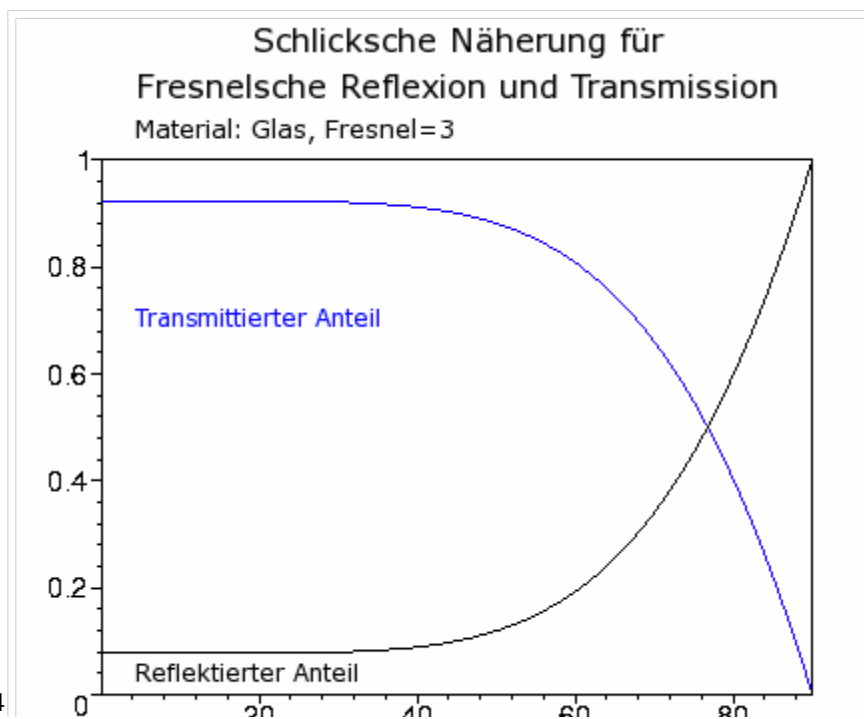
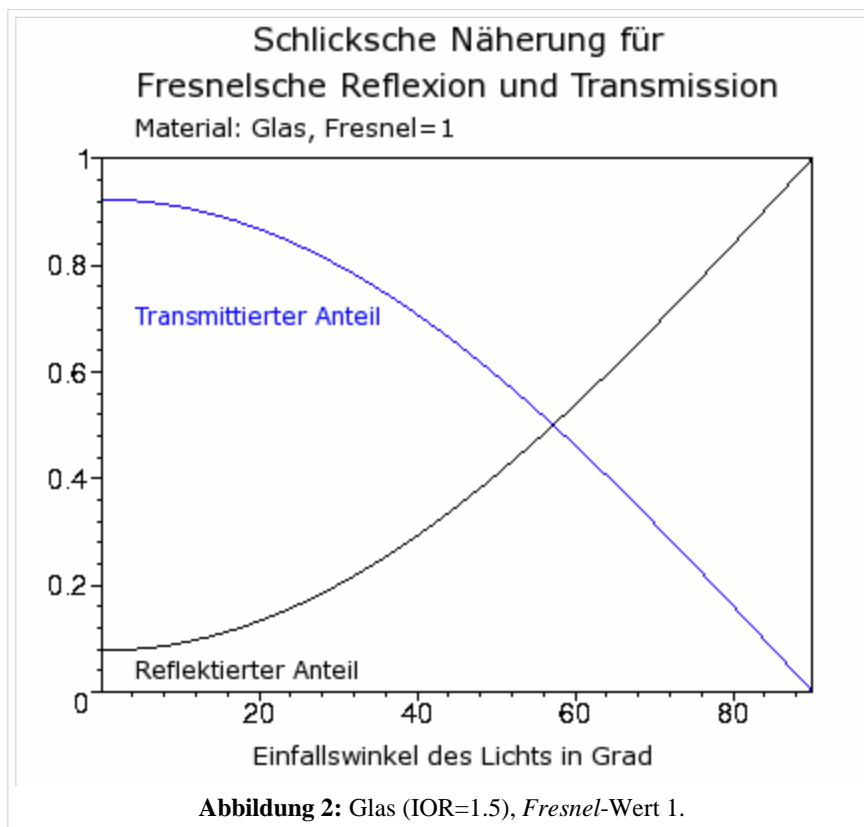
Auf die Grenzfläche zweier Materialien fallendes Licht erfährt prinzipiell drei Effekte.

- **Transmission:** Ein Teil des Lichtes scheint durch ein Material hindurch, dabei wird es gebrochen.
- **Reflexion:** Ein Teil des Lichtes wird reflektiert, dabei gilt das Reflexionsgesetz  
Einfallswinkel=Ausfallswinkel.
- **Absorption:** Ein Teil des Lichtes wird verschluckt.

Die Summe aus Transmission, Reflexion und Absorption muss gleich der einfallenden Lichtmenge sein. Die Summe der relativen Anteile muss also 1 ergeben.

Die Verhältnisse werden leider dadurch verkompliziert, dass das Verhältnis von reflektierter und transmittierter Lichtmenge vom Einfallswinkel der Lichtstrahlen abhängt. Diesem Umstand trägt der *Fresnel* Wert Rechnung. Schaut man senkrecht auf eine Oberfläche reflektiert sie am wenigsten, transparente Materialien transmittieren am besten. Umgekehrt spiegeln bswp. Glasscheiben oder Wasseroberflächen bei schräger Betrachtung sehr stark, der transmittierte Lichtanteil ist entsprechend geringer.

In Blender wird der *Fresnel* Effekt mit der Schlickschen Näherung (Schlick's Approximation) berechnet. Für Glas (IOR=1.5) und Wasser (IOR=1,33) habe ich für verschiedene *Fresnel* Werte jeweils den reflektierten und den transmittierten Anteil dargestellt. Wer sich für die Berechnungsformeln interessiert, sei auf Dielectrics and Distribution in Ray Tracing (<http://www.cs.cornell.edu/Courses/cs465/2004fa/lectures/22advray/22advray.pdf>) verwiesen. Der *Fresnel* Wert fünf für Reflexion und Transmission entspricht dabei der exakten Berechnung am meisten.



### Schlicksche Näherung für Fresnelsche Reflexion und Transmission

Material: Glas, Fresnel=5

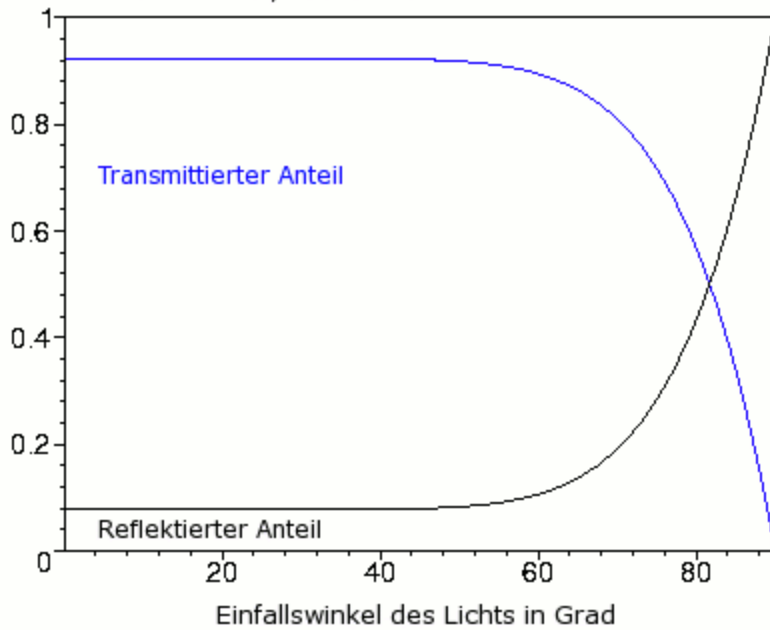


Abbildung 4: Glas (IOR=1.5), *Fresnel*-Wert 5. Dies ist die physikalisch richtigste Darstellung für Fensterglas.

### Schlicksche Näherung für Fresnelsche Reflexion und Transmission

Material: Wasser, Fresnel=1-3-5

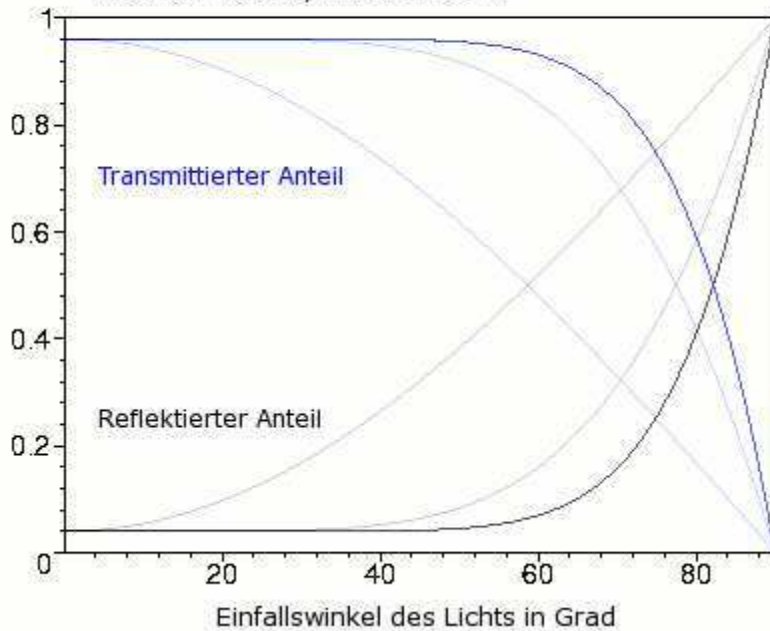


Abbildung 5: Wasser (IOR=1.33), *Fresnel*-Werte 1, 3 und 5 in einem Diagramm aufgetragen. Der Unterschied zum Glas ist verhältnismäßig klein.

## Links

Tutorial mit einem schönen Beispiel zum Fresnel-Effekt (englisch)  
(<http://feblemind.homeip.net/dotclear/index.php/2004/12/26/6-didacticiel-les-reflets---tutorial-reflections>)

Beispieleinstellungen für Chrom. ([http://blendpolis.serverpool.org/f/article\\_view.php?faq=2&fldAuto=14](http://blendpolis.serverpool.org/f/article_view.php?faq=2&fldAuto=14))

Beispieleinstellungen für Yafray (englisch).

(<http://www.elysiun.com/forum/viewtopic.php?t=21972&highlight=ray+mirror+fresnel>)

Dielectrics and Distribution in Ray Tracing

(<http://www.cs.cornell.edu/Courses/cs465/2004fa/lectures/22advray/22advray.pdf>)

Metals and Reflektions ([http://www.neilblevins.com/cg\\_education/metal\\_and\\_refs/metal\\_and\\_refs.htm](http://www.neilblevins.com/cg_education/metal_and_refs/metal_and_refs.htm))

## Raytracing-Transparenz

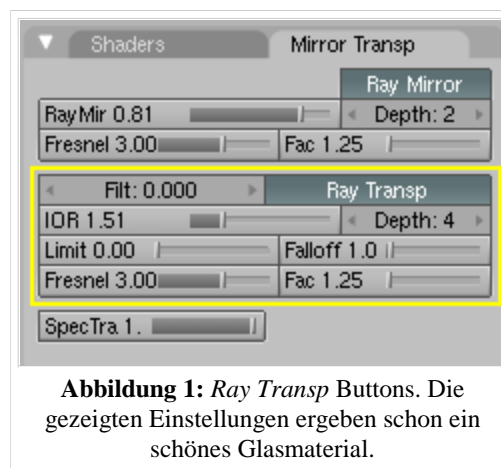
Diese Seite bezieht sich auf  
**Blender v2.42**

Transparenz ist die Materialeigenschaft, die man als "Durchsichtigkeit" bezeichnet. Glas, Kunststoffe, Wasser und viele andere Flüssigkeiten sind transparent. Ein Material in Blender durchsichtig zu machen, ist sehr einfach.



Benutzen Sie für klares Glas einfach die Einstellungen aus **Abbildung 1**. Das Glas üblicherweise glatt ist, erhöhen Sie auf dem *Shaders* Panel die *Specularity* (bis auf 2) und die *Hardness* (auf 260).

1. Setzen Sie *Alpha* auf einen Wert kleiner als 1 (im *Material* Panel der *Material buttons*), oder stellen Sie *Fresnel* auf einen Wert größer 0 und den zugehörigen *Fac* auf einen Wert größer 1 (im *Mirror Transp* Panel der *Material buttons*).
2. Klicken Sie *Ray Transp* im *Mirror Transp* Panel der *Material buttons*.
3. Schalten Sie Raytracing ein (im *Render* Panel der *Szene buttons*).



**Abbildung 1:** *Ray Transp* Buttons. Die gezeigten Einstellungen ergeben schon ein schönes Glasmaterial.

Aber wenn es durchsichtig ist, wieso kann man es dann sehen? Hier beginnt der etwas schwierigere Teil.

Anfangen werden wir mit den verschiedenen Transparenz bezogenen Materialeinstellungen im Panel *Mirror Transp* (**Abbildung 1**). Ich gehe übrigens hier davon aus, dass sie bereits die Seite über Raytracing Spiegelungen gelesen haben.

- *Ray Transp*: Stellt die Raytracing Transparenz an oder ab. Wenn Sie Transparenz ohne Raytracing wollen, müssen Sie dafür *ZTransp* benutzen.
- *Filt*: Transparentes Glas ist vollständig durchsichtig, was auch bedeutet, dass es kein Licht absorbiert. Das ist natürlich unrealistisch. Insbesondere farbiges Glas sollte Licht anderer Farben absorbieren. Das lässt sich mit der *Filt* Einstellung erreichen, auch wenn der *Alpha* Wert des Materials 0 ist! Eine genauere Besprechung folgt unten.
- *IOR*: Index of Refraction = Brechungsindex. Typische Brechungsindizes für verschiedene Materialien finden Sie z.B. bei Wikipedia (der Link ist unten aufgelistet), Wasser hat einen Brechungsindex von 1.3, Fensterglas einen von 1.5. Brechungseffekte funktionieren nur mit Raytracing.
- *Depth*: Die Anzahl an Oberflächen (+1), die ein Lichtstrahl durchdringen kann. *Depth* 0 bedeutet 1 Oberfläche, *Depth* 1 zwei Oberflächen (ein Körper) usw. Für ein Glas benötigt man *Depth* 3, da Vorder- und Rückseite zu durchdringen sind. Zwei hintereinandergestellte Gläser brauchen *Depth* 7.
- *Limit*: Ein Wert über 0 aktiviert *Transmissivity*. Das Licht wird beim Durchgang durch das Objekt abgeschwächt. Der *Limit* Wert begrenzt die Abschwächung, jeder Lichtstrahl der länger ist als *Limit* Blendereinheiten, wird nicht weiter abgeschwächt. Empfohlene Einstellung: 100. Siehe den Abschnitt *Transmissivity*.
- *Falloff*: Exponent der Berechnung der Lichtabschwächung für *Transmissivity*. 1 bedeutet linearer Falloff, 2 quadratischer Falloff, 0.5 entsprechend eine Abhängigkeit von der Quadratwurzel aus dem Lichtweg.
- *Fresnel*: Winkelabhängigkeit der Transparenz. Ein *Fresnel*-Wert von 0 bedeutet keine Winkelabhängigkeit,



und Sie müssen die Transparenz mit dem *Alpha*-Wert des Materials einstellen. Höhere *Fresnel*-Werte bedeuten zunehmende Blickwinkelabhängigkeit. Bei senkrechtem Blick auf die Oberfläche ist das Material transparent, bei schrägem Blick undurchsichtig (Siehe Raytracing Spiegelungen). Die resultierende Transparenz über *Fresnel* und *Alpha* ist gleichwertig. Obwohl *Fresnel* hier ausführlich erwähnt wird, kann es auch mit *ZTransp* eingesetzt werden, benötigt also kein Raytracing!

- *Fac*: Stärke des *Fresnel* Effektes.

## Glas

Zurück zur Ausgangsfrage: Warum kann man durchsichtige Materialien sehen?

1. Sie brechen das Licht. Dies kann Blender mit dem IOR berücksichtigen.
2. Sie spiegeln das Licht - den *Diffuse* Anteil in Blender. Das kann Blender über *Ray Mirror* oder über Environmentmaps berücksichtigen.
3. Sie zeigen Glanzpunkte - der *Spec* Anteil in Blender. Um diese zu variieren kann u.U. eine *Spec-Map* notwendig sein. Glanzpunkte sind Spiegelbilder der Lampen! Auf diese trifft der Fresnel-Effekt genauso zu wie auf alle anderen Reflexionen, Blender berücksichtigt das aber nicht.
4. Sie ändern die Lichtfarbe/Lichtintensität. Dazu benutzt man das *Transparent Filtering* und *Transmissivity*.
5. Sie sind ungleichmäßig dick (Fensterscheiben) oder variieren im Brechungsindex. Dafür eignen sich insbesondere Alpha- und Normal Maps.
6. Sie sind schmutzig. Go wild ;-).



**Abbildung 2:** Ein Glaskopf. Das ist eine Szene aus den Testdateien zur Version 2.37.

Sie können Glas auch ohne Raytracing erzeugen, allerdings verlieren sie dann den IOR, das *Transparent Filtering* und die *Transmissivity*.

## Transparent Filtering (Transparentes Filtern)



**Abbildung 3:** Uh, ist das hässlich. Aber schön? bunt ;-)

Der Filter operiert nur mit dem transparenten Anteil des Materials, egal ob diese Transparenz durch den *Alpha* Wert oder durch die *Fresnel* Einstellung erzeugt wurde. Steht *Filt* auf 0.0 passiert das Licht unverändert das Material. Bei 1.0 wird es vollständig mit der Materialfarbe gefiltert. Um einen Farbfilter zu erzeugen setzen Sie *Alpha* auf 0 und *Filt* auf 1.

## Transmissivity (Lichtabsorption)

*Transmissivity* ist eine Eigenschaft, die man an vielen dicken Glasobjekten oder an Wasser gut beobachten kann. Je dicker das Glasobjekt ist, desto weniger Licht kommt hindurch, da die Durchlässigkeit eben nicht 100% beträgt. Dabei ist das keine Eigenschaft der Oberfläche, sondern des Körpers. Alle anderen Materialeigenschaften betreffen ausschließlich die Oberfläche.

Den Effekt der *Transmissivity* Einstellungen sehen Sie bei einem *Falloff* von 1 einigermaßen deutlich nur unter bestimmten Voraussetzungen:

- Das Glasobjekt ist ziemlich dick.
- Der *Alpha* Wert des Materials ist nicht durchgängig Null, am besten stellen Sie *Alpha* auf 1 und benutzen *Fresnel-Transparenz*.
- Der *Filter* Wert des Materials beträgt ca. 0.5.

Aber Sie können natürlich höhere *Falloff*-Werte benutzen.



**Abbildung 4:** Glas mit und ohne *Transmissivity*. Ist der *Limit* Wert Null, ist das Glas dunkler als der Glaskörper. Ist der *Limit* Wert 1 der Glaskörper dunkler als das Glas. Die Einstellungen finden Sie in *Abbildung 5*.



**Abbildung 5:** Die Einstellungen für das Material in *Abbildung 4*.

**Alpha-Kanal speichern:** Raytracing Transparenz kann als Alpha-Wert in einem entsprechenden Bildformat (z.B. PNG oder TGA) gespeichert werden. Stellen Sie dazu in den Render Buttons im *Format* Panel *RGBA* ein. Klicken Sie im *Render* Panel auf *Premul*, damit der Hintergrund nicht mit gerendert wird.

## Links

Brechzahlen bei Wikipedia (<http://de.wikipedia.org/wiki/Brechungsindex>)

Transparent Filtering in Blender ([http://www.blender3d.org/cms/Transparent\\_Filtering.514.0.html](http://www.blender3d.org/cms/Transparent_Filtering.514.0.html))

## Multimaterial

*bezieht sich auf Blender v2.37a*

Die meisten komplexeren Gegenstände sind aus mehreren Objekten zusammengesetzt. Aber es gibt Situationen (insbesondere z.B. beim SubSurface Modelling) wo ein Objekt aus einem Mesh besteht, aber verschiedene Materialien tragen soll.

## Kurzanleitung

Um einem Mesh verschiedene Materialien zuzuweisen, sind vier Schritte notwendig:

1. Die die Flächen aufspannenden Vertices müssen ausgewählt werden.
2. Nun muss ein neuer *Material index* erzeugt werden. Sozusagen eine Speichermöglichkeit für eine Gruppe von Vertices. *Editing Buttons (F9) -> Link and Materials Panel -> New*
3. Diesem Speicherort werden (im Editmodus) die ausgewählten Vertices zugewiesen (->*Assign*). Man würde wohl in anderen Programmen davon sprechen, dass die Vertex Gruppe dort gespeichert wird. Die Vertexgruppe wird dadurch nicht verändert (warum auch?), behält insbesondere noch ihr Material.
4. Schließlich kann dieser Vertexgruppe ein eigenes Material zugewiesen werden.

Die Schritte 1 und 2 können vertauscht werden.



### Anzahl der Materialien:

Ein kleines Problem: Sie können einem Mesh höchstens 16 verschiedene Materialien zuweisen.


Benötigen Sie mehr Materialien, teilen Sie das Objekt in verschiedene Meshes auf (*Separate* mit der **P**-Taste).

## Ausführliches Beispiel

Diese vier Schritte werden wir an einem ausgearbeiteten Beispiel nachvollziehen.

Wir werden eine Kugel mit zwei verschiedenen Materialien versehen. In Abbildung 1 sind alle Vertices der Kugel ausgewählt, wie Sie im *Material* Panel erkennen können hat das Mesh 1 Material (erste Zahl), und dies ist das gerade ausgewählte (zweite Zahl).

Machen Sie das Material rot, (R=1, G=0, B=0).

Klicken Sie auf das Auto Icon . Der Name des Materials sollte nun "Red" lauten. Schalten Sie in den *Shaded* Modus (Z-Taste), erscheint die Kugel im 3D-Fenster rot. Schalten Sie mit Z wieder zurück in den *Wireframe* Modus.

Wählen Sie die obere Hälfte der Kugelvertices aus. Das geht am schnellsten in der Seitenansicht (Num-1) mit *Blockselect* (B-Taste).

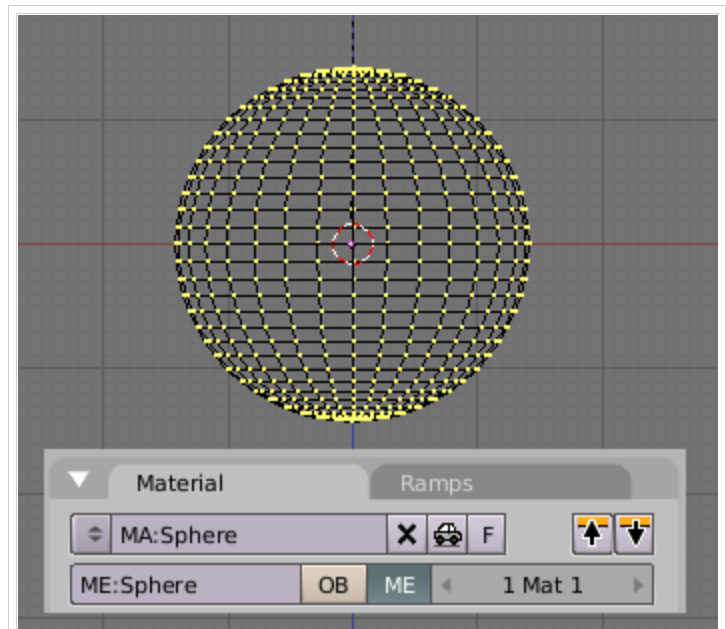


Abbildung 1: Eine Kugel mit einem Material

Wechseln Sie in die *Editing Buttons* (F9). Im *Link and Materials* Panel sind zwei Button Gruppen, wir brauchen hier die rechte Seite unter *Red*, dem Namen des Materials. Hält man den Mauszeiger über den rechten *New* Button, erscheint der Tooltip: *Adds a new Material index*.

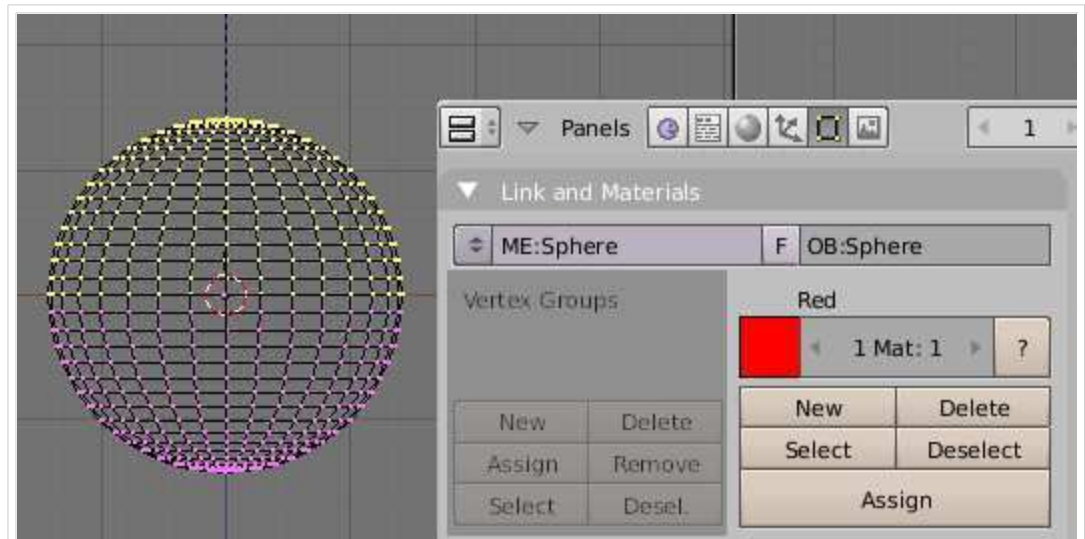


Abbildung 2: Die obere Hälfte der Kugel ist ausgewählt.

Klicken Sie auf *New*, damit erzeugen Sie den neuen *Material index*. Die Angabe im Zahlenfeld sollte jetzt *2 Mat: 2* lauten.

Klicken Sie auf *Assign*. Jetzt haben Sie die Vertices gespeichert. Über die Buttons *Deselect* und *Select* können Sie prüfen, ob Sie alle benötigten Vertices gespeichert haben. Sie können allerdings nicht einzelne Vertices zur Gruppe hinzufügen oder aus ihr entfernen.



Abbildung 3: 2 Materialien, das zweite ist aktiv.





#### Vertex Groups:

Wollen Sie komplizierte Auswahlen erstellen, benutzen Sie die *Vertex Groups* (auf dem gleichen Panel). Die funktionieren fast genauso wie die *Material Indices*, mit *Assign* und *Remove* können Sie Vertices zur Gruppe hinzufügen, oder aus ihr entfernen.

Bis jetzt sehen Sie noch keine Änderung an ihrem Objekt. Beide Vertexgruppen tragen das gleiche Material. Sie müssen der neu erzeugten Gruppe ein eigenes Material zuweisen. Dazu wechseln Sie zurück in die *Shading Buttons* (F5).

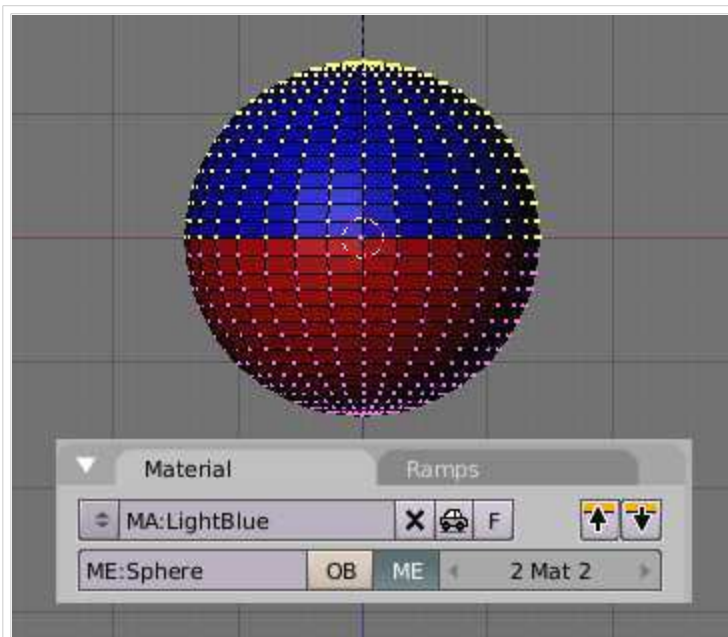
Jetzt haben Sie drei Möglichkeiten.

- Klicken Sie auf die 2 neben dem Materialnamen, erstellen Sie eine *Single user* Kopie.
- Erstellen Sie ein neues Material oder wählen Sie ein vorhandenes aus .
- Kopieren Sie ein Material aus dem Zwischenspeicher .



**Abbildung 4:** Zurück in den *Shading* Buttons. Hier weisen Sie ein neues Material zu.

Wir werden eine *Single user* Kopie erstellen. Ändern Sie die Farbe auf (R=0, G=0, B=1). Klicken Sie auf den Auto Button.



**Abbildung 5:** Unser Ergebnis.

Sollte ihr Bild nicht so aussehen wie in Abbildung 5, haben Sie wahrscheinlich vergessen, den *Assign* Button zu drücken. Das wird gerne vergessen, man ist von anderen Programmen auch daran gewöhnt, dass die Auswahl automatisch einer neu erstellten Gruppe zugewiesen wird. Das ist in Blender nicht der Fall.

## Neue Objekte mit mehreren Materialien

Fügt man mit **Strg-J** zwei Objekte zusammen, so behalten die Vertices ihren *Material index*. Das zusammengefügte Objekt trägt also mehrere Materialien.

Trennt man ein Objekt mit mehreren Material Indizes, so trägt jedes einzelne Objekt alle Indizes des Ursprungsobjektes.

Auch durch Boolesche Operationen erstellte Objekte tragen mehrere Material Indizes, nämlich die der Ausgangsvertices.

Und schließlich trägt das Ergebnis der Radiosity-Modellierung mehrere Material Indizes (was auch das größte Problem ist, da dabei häufig mehr als 16 verschiedene Objekte zusammengefasst werden).

## Links

Die englischsprachige Vorlage dieses Textes (<http://www.blender.org/modules/documentation/htmlII/x4448.html>)  
Flashtutorial (deutsch) ([http://blendpolis.serverpool.org/f/article\\_view.php?fldAuto=24&faq=2](http://blendpolis.serverpool.org/f/article_view.php?fldAuto=24&faq=2))

Von „[http://de.wikibooks.org/wiki/Blender\\_Dokumentation/\\_Druckversion/\\_Material3](http://de.wikibooks.org/wiki/Blender_Dokumentation/_Druckversion/_Material3)“

- 
- Diese Seite wurde zuletzt am 12. Oktober 2006 um 21:08 Uhr geändert.
  - Inhalt ist verfügbar unter der GNU Free Documentation License.
  - Datenschutz
  - Über Wikibooks
  - Haftungsausschluss