

[Das Blender-Buch](#)

3D-Grafik und Animation mit Blender

Bearbeitet von
Carsten Wartmann

5., aktualisierte Auflage 2014. Buch. 426 S. Kartoniert

ISBN 978 3 86490 051 8

Format (B x L): 18,5 x 24,5 cm

[Weitere Fachgebiete > EDV, Informatik](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](#) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

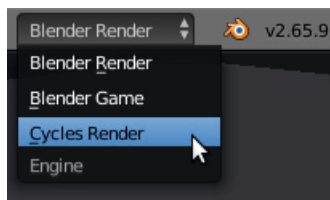
8 Cycles

Cycles ist die seit Blender 2.61 verfügbare physikbasierte Renderengine. Hauptentwickler ist Brecht van Lommel, der schon lange die interne Renderengine in Blender betreut. Nachdem Blender Intern mit seinem 80er-Jahre-Design immer mehr Schwächen zeigte und Erweiterungen nicht mehr einfach zu integrieren waren, entschloss sich Brecht van Lommel 2011 [CYCLES] dazu, eine neue Renderengine zu schreiben. Grundlage waren etliche Artikel über moderne Rendertechniken und so entstand ein solides Grundgerüst, auf dem dann die zukünftige Renderengine von Blender stehen soll. Cycles soll den Blender Intern Renderer in Zukunft einmal komplett ablösen, bis dahin ist es aber sicher noch eine Weile hin und so behält auch BI weiterhin seine Berechtigung.

Cycles ist physikbasiert, was die Engine in einer Nische zwischen physikalisch korrekten und auf totalen Realismus ausgerichteten sowie komplett programmierbaren Renderern positioniert. Die Physikbasiertheit ermöglicht eine intuitive Benutzung, um realistische Renderings zu erstellen, gleichzeitig sind aber auch Tricks und Cheats möglich, um einen speziellen Look oder eine Renderzeitverkürzung zu erzielen. Ziel ist es, eine Engine für die Produktion von Filmen für kleine bis mittelständische Studios und Künstler zu schaffen.

Ein weiteres Designziel ist es, eine interaktive Renderengine zu entwickeln. Wird etwas in der Szene geändert (Beleuchtung, Materialien etc.), so wird die Szene im 3D View sofort neu gerendert, wobei erst ein grobes, aber komplettes Bild dargestellt und das dann laufend verfeinert wird. Um dies zu erreichen, wird auch die Leistung moderner Grafikkarten genutzt, um zu rendern.

Die Shader, also Programmteile, die das Aussehen berechnen (Beleuchtung, Material, Texturen), sind Node-basiert und reihen sich so in die Entwicklung ein, die sowieso schon mit den Node-basierten Materialien und dem Compositor (sowie in Zukunft einem Node-basierten Partikelsystem) in Blender Einzug gehalten hat. Diese Nodes ermöglichen eine hohe Flexibilität und komplexe Shader, sind aber trotzdem leichter zu überblicken als ein GUI-basierter Ansatz wie bei den Blender-Intern-Materialien.

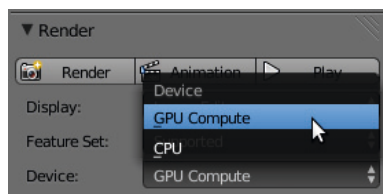


Cycles wird inzwischen standardmäßig als Add-on mit Blender verteilt und ist auch aktiviert, so dass ein Umschalten auf Cycles Render im Header von Blender reicht, um Cycles zu benutzen. Dies zeigt auch ein weiteres Designziel von Cycles: Die Schnittstellen zu externen Render-Engines sollen verbessert werden, so dass Blender-Anwender in Zukunft die Wahl zwischen möglichst vielen Engines haben, die trotzdem alle perfekt in Blender integriert sind.

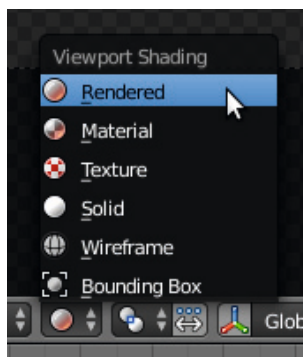
8.1 Übung: Schnellstart in Cycles



Laden Sie die Szene `Cycles/Cycles_Schnellstart.blend` von meiner Seite. Sie besteht wie üblich aus einem Affenkopf auf einer Grundfläche. Schalten Sie nun die Renderengine auf Cycles Render um. Eventuell haben Sie jetzt schon aus dem Augenwinkel eine leichte Veränderung der Darstellung im 3D View und im Scene Context bemerkt, hier hat sich das Render-Panel verändert und es gibt nun das Menü `Feature Set`. Im Prinzip könnten Sie hier schon die Darstellungsmethode im 3D View auf `Rendered` umschalten und der Affenkopf würde in einer ambient beleuchteten Umgebung berechnet werden.





Haben Sie allerdings eine schnelle Grafikkarte (aktuell wird nur CUDA auf Nvidia benutzt, Unterstützung für AMDs OpenCL ist in der Entwicklung, scheitert allerdings noch an einigen Bugs in OpenCL), so können Sie diese auch für die Berechnung nutzen, dazu sollten aber die aktuellsten Grafikkartentreiber installiert sein. Die Einstellungen finden Sie in den User Preferences (`Strg` - `Alt` `U`) im System-Tab unten links. Wenn Sie hier das `Compute Device` auf `CUDA` ändern können, sind die richtigen Treiber und eine kompatible Grafikkarte installiert und die Berechnung in Cycles wird je nach Karte um einiges schneller. Sollten Sie mehrere Grafikkarten im System haben, kann noch die Karte ausgewählt werden, die Rendern soll. Ist diese Hardware-Auswahl erfolgt, kann auch im Scene-Kontext `GPU Compute` gewählt werden.



Nun können Sie die Darstellungsmethode im 3D View auf `Rendered` umschalten und ein erstes Bild von Cycles gerendert bekommen. Was hier wie ein Rendering aussieht, ist aber in Wirklichkeit immer noch unser 3D View. Versuchen Sie doch einmal den View mit gehaltener `Q`-Taste und Mausebewegungen zu drehen, Cycles wird nun je nach Rechenleistung Ihres Computers die Ansicht immer neu berechnen. Sobald Sie mit der Transformation des Views aufhören, verfeinert Cycles das Bild immer weiter. Dass es sich um einen 3D View handelt, merken Sie auch daran, dass Sie weiterhin per `Q` Objekte selektieren können, unten links im View wird dann auch der Name des Objekts angezeigt.

Selektieren Sie nun bitte den Affenkopf und wechseln in den Material-Kontext. Mit Klick auf den `New`-Button erstellen Sie hier ein neues Mate-

rial mit dem Diffuse BSDF-Shader. Dieser Shader definiert ein diffus Licht streuendes Material. Ändern Sie doch einmal die Farbe des Materials durch Klick auf das Farbfeld neben Color: und Cycles wird das Objekt direkt neu rendern. Probieren Sie auch einmal andere Shader, einige sind allerdings nur in Kombination mit aufwändigeren Materialien sinnvoll.

Im Moment ist die Beleuchtung noch wie an einem sehr wolkigen Tag. Wechseln Sie in den World Context und erstellen mit **New** eine neue Welt. Mittels **Strength**: und der Farbe können Sie nun eine grundsätzliche Lichtstimmung einstellen. Als Nächstes können Sie eine Lichtquelle mit  -  → Lamp hinzufügen. Aktuell arbeiten die Lampentypen Point, Spot, Sun und Area so, wie man es erwarten würde, die Hemi Lamp wird aktuell noch nicht unterstützt und wie eine Lampe vom Typ Sun berechnet. Im nächsten Abschnitt werden die einzelnen Lampentypen ausführlich behandelt.

8.2 Cycles Camera

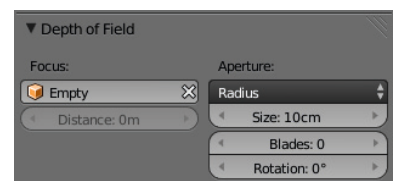
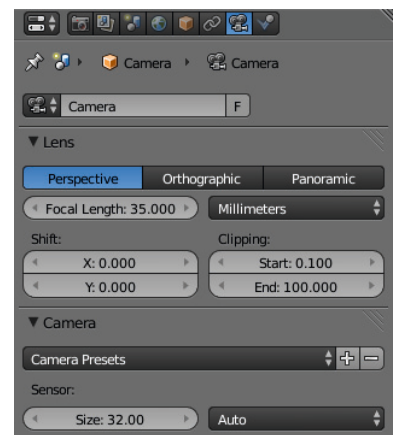
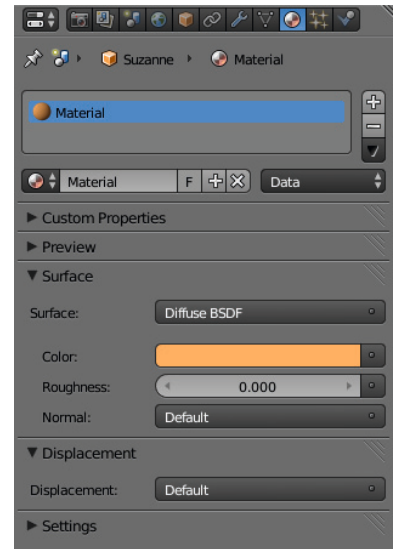
Cycles unterstützt wie auch Blender Intern perspektivische (normale Kamera), orthografische und Panorama-Kameras. Orthografische Kameras werden gerne zum Berechnen von Texturen und eher grafischen Layouts verwendet, Panoramakameras berechnen Panoramen, die die ganze Welt komplett abbilden können.

Die normale Kamera reagiert wie eine Fotokamera in der realen Welt, der wichtigste Parameter ist die Brennweite **Focal Length**:, die wie bei Objektiven üblich in Millimeter (oder abgeleiteten Einheiten) angegeben wird. Da die Berechnung an den optischen Gesetzen ausgerichtet ist, gibt es aber noch eine weitere wichtige Größe im **Camera**-Panel, die Sensorgröße, was entweder der Größe des Aufnahmechips in Digitalkameras oder der belichteten Filmgröße entspricht. Zusammen mit der Brennweite ergeben sich dann die Eigenschaften des Systems aus Objektiv und Kamera – wichtig für die Abbildungsgröße und Schärfentiefe (auch Tiefenschärfe genannt). Diese an der realen Welt ausgerichtete Betrachtungsweise findet sich auch im **Camera Presets**-Menü wieder, wo Sie einige Daten von realen Kameras finden. Dies zusammen mit dem Bildseitenverhältnis im Render Context ermöglicht es, einfacher 3D-Szenen mit Realaufnahmen zu kombinieren.

Mittels **Shift**: können Sie nach Festlegen einer Perspektive die Kamera-sicht noch verschieben (und so das Objekt positionieren), ohne stürzende Linien zu produzieren; dies ist bei Architektur-Renderings sehr beliebt.

Schärfentiefe – Depth of Field (DOF)

Die Cycles-Kameras berechnen sehr realistische Schärfentiefeeffekte. Der wichtigste Parameter ist hier der Schärfepunkt (**Focus**:), den man als Entfernung (**Distance**:) in Blendereinheiten (softwarespezifische Einheit) oder




metrischen Einheiten vom Kameraursprung aus angeben kann. Praktischer ist allerdings ein Hilfsobjekt (z. B. ein Empty oder auch den Hauptakteur) anzugeben, woraufhin der Schärfepunkt immer auf diesem Objekt liegt. Durch Animation des Hilfsobjekts oder den **Distance**-Wert kann auch eine manuelle oder automatische (Autofokuspumpen) Schärfeeinstellung animiert werden.

Der zweite, wesentlich die Schärfentiefe beeinflussende Faktor ist die Blendenöffnung (**Aperture**), die entweder direkt oder als Blendenstufe (**F/Stop**) eingegeben werden kann. Dabei gilt wie bei echten Kameras: Je kleiner die Blendenstufe, desto größer die Blendenöffnung, desto kleiner ist der Schärfentiefebereich. Dabei ist auch zu beachten, dass je größer die Brennweite ist, desto kleiner der Schärfentiefebereich ausfällt.

Blades: gibt die Anzahl der Lamellen für die Blende an, Werte unter 3 führen zu kreisförmigen Halos um Glanzlichter, Werte darüber entsprechend zu Dreiecken, Vierecken etc. Übliche Anzahlen der Lamellen von Irisblenden sind 9, was zu einem Nonagon, also einem Neuneck führt.

8.3 Cycles-Lampentypen

Neben Licht vom Welthintergrund und Licht ausendenden Meshes (Emission Shader), sind Lampen eine weitere und einfach zu handhabende Lichtquelle für Ihre Szenen. Lichtquellen sind wie im BI auch in Cycles nicht direkt in dem berechneten Bild zu sehen und werden im Modelling Prozess genau wie die Lichter in Blender Intern Renderer (Kapitel 7) bewegt und animiert.

Lampen werden wie gewohnt per -**A**→Lamp hinzugefügt. Im Lamp-Panel des Lamp-Kontexts kann dann der Lampentyp noch verändert werden. Aktuell arbeiten die Lampentypen Point, Spot, Sun und Area so, wie man erwarten würde, die Hemi Lamp wird aktuell noch nicht unterstützt und wie ein Lampe vom Typ Sun berechnet.

Size:

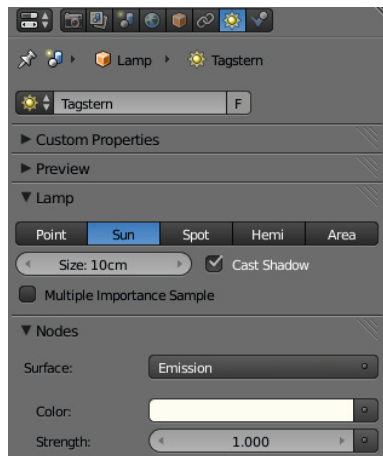
Mit dem **Size**-Parameter wird die Größe der Lampe in Blender oder Szeneneinheiten (Meter, Zentimeter etc.) festgelegt. Je größer die Lampe, desto weicher der Schatten und Shading.

Cast Shadow

Mit diesem Button kann die Schattenberechnung an- und ausgeschaltet werden, was unter Umständen auch die Rechenzeit verkürzen kann.

Color:

Hier kann mit dem Standard-Farbwähler die Farbe der Lichtquelle ausgewählt werden.



Strength:

Die Stärke der Lichtquelle in Watt (W), mit Ausnahme des Typs Sun-Lamp.

Point Lamp

Point Lamps sind Lichtquellen, die ihr Licht gleichmäßig in alle Richtungen abstrahlen. Mit einem Size: von null entsprechen sie einer nur theoretisch möglichen Punktlichtquelle mit absolut scharfen Schatten, mit einem Size:>0 werden Point Lamps zu kugeligen Lichtquellen mit entsprechend weicheren Schatten und Shading.

Spot Lamp

Spot Lamps strahlen das Licht innerhalb eines Kegels in eine definierte Richtung aus. Size: im Lamp-Panel bestimmt auch hier die Weichheit des Schattens und von Shadings. Mit Blend: im Spot Shape-Panel kann unabhängig davon der Rand des Lichtkegels weicher gemacht werden. Size: im Spot Shape-Panel bestimmt den Öffnungswinkel des Lichtkegels in Grad.

Area Lamp

Area Lamps strahlen das Licht von einer rechteckigen Fläche aus nach dem Lambertschen Gesetz ab, was praktisch bedeutet, dass die Lichtstärke kugelförmig gleichmäßig von der Fläche ausgehend verteilt ist.

Sun Lamp

Sun Lamps strahlen paralleles Licht in einer bestimmten, durch die Rotation der Lichtquelle bestimmten Richtung aus. Die Entfernung von der Szene spielt keine Rolle, das Licht wird mit der Entfernung nicht schwächer. Durch diese Eigenschaften ist der Strength:-Parameter nicht mit denen anderer Lichtquellen vergleichbar und muss typischerweise viel kleiner eingestellt werden.

8.4 Das Materialsystem von Cycles

Für Cycles-Materialien gilt im Prinzip das in Abschnitt 7.13 Gesagte. Neu wird es erst ab dem Surface-Panel, wichtigster Parameter ist hier Surface:, mit diesem Menü können verschiedene Shader eingestellt werden. Shader sind Rechenvorschriften, die der Renderengine sagen, wie das Licht mit der Oberfläche des Objekts interagiert, absorbiert, reflektiert und gestreut wird, aber auch ob ein Objekt selbst Licht aussendet. Die in Blender verwendeten Rechenvorschriften nennen sich BSDF, Bidirectional Scattering Distribution Function, auf Deutsch etwa Bidirektionale Streuungsverteilungsfunktion, die angeben, wie Licht auf einer Oberfläche diffus gestreut und gebeugt wird. Erweiterungen dieser Funktion sorgen dafür, dass auch Reflexion, Transmission und Beugung berechnet werden können.

