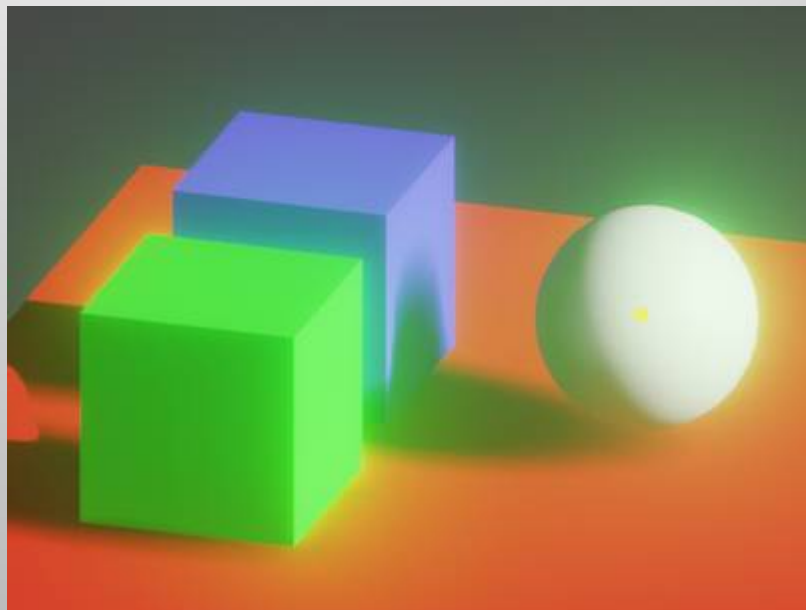


Blender Handbuch 2.8

Kapitel 5: Beleuchtung



Henricus

Version September 2019

Das Blender-Programm wird ständig weiterentwickelt. Daher ist es nicht vermeidbar, dass einige Funktionsbeschreibungen in diesem Text mit der Zeit überholt sind. In der nächsten Auflage werden die Erläuterungen dann wieder auf den neuesten Stand gebracht.

Inhalt

5	Beleuchtung	2
5.1	Beleuchtung bei Eevee	3
5.1.1	Lampenarten	4
	Die Hauptmerkmale der Lampenarten.....	4
	Widgets	5
	Besonderheiten von Spot	5
	Besonderheiten von Area	6
	IES Texture	6
5.1.2	Nebel (volumetrisches Licht) in Eevee.....	8
5.1.3	Light Probe	9
	Reflection Cubemap.....	9
	Reflection Plane	11
	Indirekte Beleuchtung (Irradiance Volume).....	11
5.1.4	Fluoreszenz (Bloom)	12
5.1.5	Addon Tri-lighting	13
5.2	Besonderheiten der Beleuchtung im Cycles Renderer	15
5.2.1	Indirekte Beleuchtung.....	15
	Lichtpfade.....	15
	Selbstleuchtende Materialien.....	17
	Mist bei Cycles	18

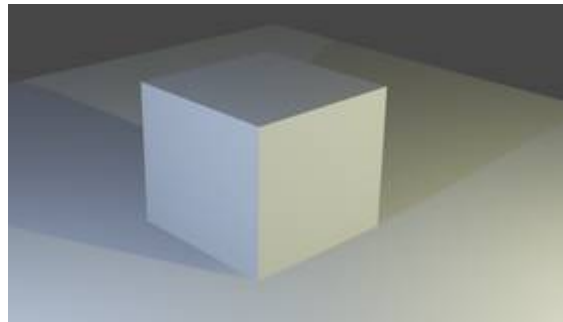
5 Beleuchtung

Die Ausleuchtung einer Szene ist (neben vielen anderen Faktoren) entscheidend für deren Gesamteindruck. Für die Programmierung sind die Wege, die Lichtstrahlen nehmen, eine große Herausforderung und es gibt viele verschiedene Arten zu deren Berechnung. In einigen Hinsichten unterscheiden sich dabei *Eevee* und *Cycles Render* grundlegend, wobei nicht immer zu entscheiden ist, welcher Renderer die anspruchsvollere Lösung bereitstellt.


Licht setzt eine Lichtquelle voraus. Die uns aus der Realität vertrauten Lichtquellen sind *Lampen*, die sich im Raum lokalisieren lassen. Ich behalte diese Bezeichnung bei, obwohl bei Blender 2.8 eine Änderung der Bezeichnung stattgefunden hat: Was früher *Lamp* hieß, wird jetzt als *Light* bezeichnet.

Zunächst setzen wir uns also mit den verschiedenen Lampen in Blender auseinander. Dabei gelten bei Eevee (Kap. 5.1) und Cycles (Kap. 5.2) teilweise unterschiedliche Regeln.

- Selektierte Lampen können mit [shift] + [D] dupliziert werden. Wenn man das Duplikat dann verschiebt, kann man die Szene aufhellen. Wenn man dann noch die Lichtfarbe verändert, wird die Szene schon sehr viel interessanter.
- Im folgenden Beispiel werden Würfel und Tisch (plane) von zwei *Point*-Lampen beleuchtet, einer mit leicht gelbem (wärmeren) und einer mit leicht blauem (kälteren) Licht. Beide Lampen werfen Schatten; im linken Teil des Tisches dominiert das blaue Licht, im rechten Teil das gelbe.



5.1 Beleuchtung bei Eevee

Alle *Lampen* werden bei Blender 2.8 als *Light* bezeichnet. Die Startszene mit dem Würfel enthält gewöhnlich bereits ein *Light*. Wenn du dieses mit RMT selektierst, erscheint im Properties-Panel bei den Object Data ein neues Icon. 

- Das Untermenü *Preview* zeigt, wie sich das Licht dieser Lampe im Raum verteilen würde.
- Im Untermenü *Light*
 - kann der Typ des Light verändert werden: *Point*, *Sun*, *Spot* und *Area*. Zunächst behandeln wir hier *Point*; zu den anderen Typen kommen wir später;
 - im Farbfeld kann die Lichtfarbe eingestellt werden;
 - Mit *Power*: wird die Lichtintensität in Watt eingestellt;
 - *Specular*: Hierbei handelt es sich um eine Multiplikator zur Steuerung der Lichtintensität. Werte ungleich 1.0 führen zu nicht-photorealistischen Resultaten. Verändere diesen Wert daher nur zu künstlerischen Zwecken.
 - *Radius*: bezieht sich auf die Größe der Lichtaustrittsfäche. Ein kleiner Radius erzeugt schärfere Schatten.
 - *Custom Distance*: Wenn Haken gesetzt, wird die vom Nutzer definierte Abschwächung des Lichts anstelle einer globalen Schwelle verwendet.
 - *Distance*: Abstand, nach dem der Einfluss des Lichts 0.0 beträgt.
- Wenn im Untermenü *Shadow* **kein** Haken gesetzt ist, haben die beleuchteten Gegenstände keinen Schatten.



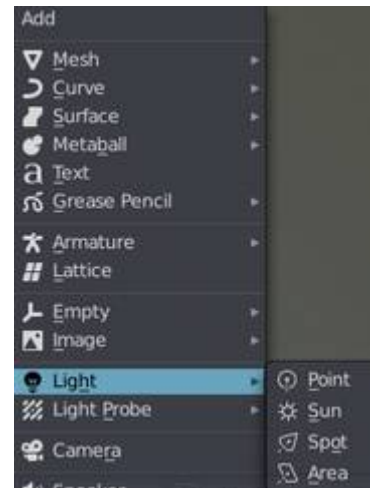
- *Clip Start*: Abstand vom leuchtenden Objekt (Lampe oder Gegenstand) ab dem der Schatten beginnt. Irgendein Objekt vor diesem Abstand wirft keinen Schatten. (*Clip End* steht nur für Sonnenlicht zur Verfügung.)
- *Softness*: Größe des Filters, der sich auf den Schattenwurf auswirkt. Je größer der Filter umso unschärfer der Schatten.
- *Bias*: Hiermit werden unerwünschte Effekte reduziert, bei denen Objekte auf sich selbst Schatten werfen, z.B. wenn sich der Arm eines Characters zwischen Körper und Lichtquelle befindet.
- *Exponent*: Hiermit wird bei Anwendung von ESM (Exponential Shadow Mapping) das Auftreten von Lichtlecks (unbeabsichtigtes Auftreten von Lichteinfall) vermindert.
- *Bleed Bias*: Hiermit wird bei Anwendung von VSM (Variance Shadow Mapping) das Auftreten von Lichtlecks (unbeabsichtigtes Auftreten von Lichteinfall) vermindert.

5.1.1 Lampenarten

Du kannst über [shift] + [A] jederzeit eine neue Lampe einfügen, indem du auf *Light* und dann auf *Point*, *Sun*, *Spot* oder *Area* klickst. Im Outliner erscheint die Lampe dann mit dem Namen der ausgewählten Lampenart.

Eine zweite Möglichkeit besteht über das Menü *Add*, das du in der Kopfzeile der *3D Viewport* sichtbar machen kannst. Nach Klick auf *Add* öffnet sich dort eine Auswahl an Objekten und unter *Light* findest du dieselben Lampentypen wie beim Ausruf über [shift] + [A].

Aber Achtung: In den *Properties* kannst du (wenn eine Lampe selektiert ist) im Untermenü *Light* auch nachträglich jeder Lampe einen anderen Lampentyp zuweisen, indem du auf eine der Schaltflächen klickst.



Das kann leicht zu Irritationen führen, denn der Lampenname im *Outliner* wird dadurch *nicht* verändert.

Die Hauptmerkmale der Lampenarten

Die Hauptmerkmale, um dies hier geht, sind:

Gerichtet: Hat das Licht eine bestimmte Richtung, aus der die Lichtstrahlen kommen?

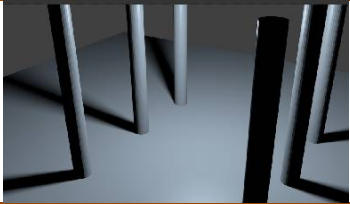
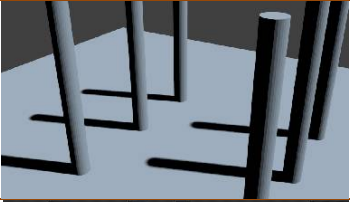

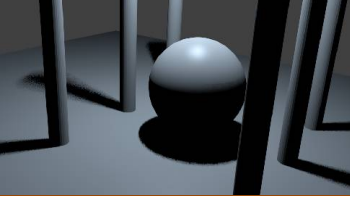
Falloff: Hängt die Beleuchtungsstärke von der Entfernung der Lichtquelle ab?

Schatten: Wirft das Licht Schatten und – wenn ja – in welche Richtung?

Beleuchtung: In welcher Weise wird die Szene beleuchtet?

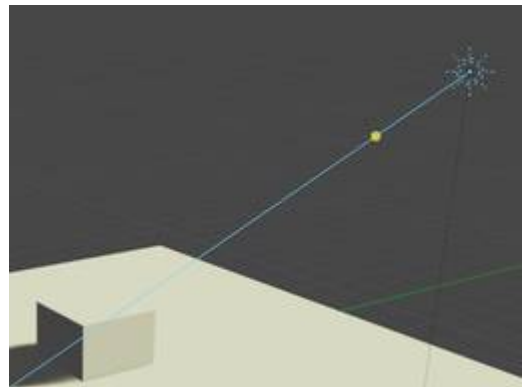
In den folgenden Beispielen werden sechs Stäbe und teilweise eine Kugel verwendet, wobei sich die Lampe etwa in der Mitte der Szene befindet.

Die folgende Tabelle enthält die wichtigsten Angaben zu den einzelnen Lampenarten.

Lampenart	gerichtet	Falloff	Schatten	Beleuchtung	Wirkung
Point	Nein	ja	zeigen von der Lampe weg	Wie eine Glühbirne	
Sun	ja	nein (Die Sonne ist überall gleich hell.)	parallel (Sonne ist unendlich entfernt)	überall gleich	
Spot	ja	ja	radial	eng abgegrenzt; Größe und Form sind einstellbar	
Area	ja	Gamma-Wert	radial	bei zu großer Distance grell	

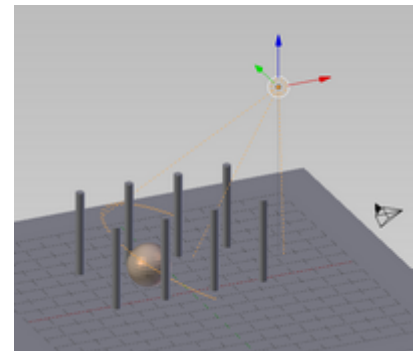
Widgets

Abgesehen von der Lampenart *Point* können die Lampen in ihrer Richtung von sog. *Widgets* (=Einerichtungen) gesteuert werden. Dabei erscheint auf dem Richtungsstrahl der Lampe ein gelber Punkt, den so mit der LMT anfassen und bewegen kannst. Du wirst feststellen, dass sich die Schatten entsprechend verändern.



Besonderheiten von Spot

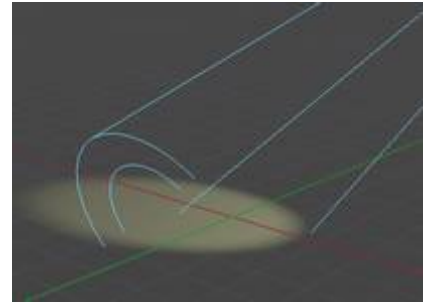
- *Spot* ist ein gerichtetes Licht: Das Licht fällt von der Lichtquelle aus nur in eine Richtung; alle was hinter der Lichtquelle liegt, bleibt dunkel. Der Lichtkegel wird angezeigt und kann nach Drücken von Taste [R] geschwenkt werden.
- Der Abstand zwischen Lichtquelle und dem Ende des Kegels in nebenstehender Anzeige kann im Feld **Distance** (unter **Falloff**) verändert werden.



- Größe und Form der beleuchteten Fläche können – wie mit den *Widgets* – auch im Untermenü *Spot Shape* eingestellt werden:

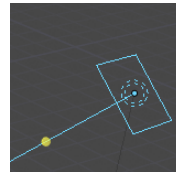


- Die Größe wird im Feld *Size* festgelegt. Die eingegebene Gradzahl meint, wie weit der Lichtkegel gespreizt ist.
- Der Wert für *Blend* gibt an, wie hart (oder weich) der Übergang zwischen beleuchtetem und unbeleuchtetem Bereich ist: niedriger Wert = hart; hoher Wert = weich.
- Den *Blend*-Wert kann man sich auch im Viewport ansehen; dort erscheint der Übergang dann als zweiter Kreis. Im rechten Bild ist *Blend* = 0.6 eingestellt.



Besonderheiten von Area

- *Area* ist ein Flächenlicht (Licht-Panel), dessen Leuchtfäche im Feld *Shape* als Quadrat (*Square*), Rechteck (*Rectangle*), Kreis (*Disc*) oder Ellipse dargestellt werden kann.



- Durch die *Size* wird festgelegt, wie weit sich die *Power* auf die Lichtfläche verteilt. Ist die *Size* groß (z.B. 10 x 10), erscheint die Beleuchtung daher (bei gleicher *Power*) schwächer, d.h. dunkler.



IES Texture

In Kap. 4.2.4 (S. 31) wurde diese Textur bereits beschrieben. Zur Vermeidung des Nachschlagens wird der Text hier wiederholt.

IES beschreibt die Form, wie eine Lichtquelle ihre Lichtintensität verteilt. Am ehesten eignet sich diese Textur für *Point Lights*.

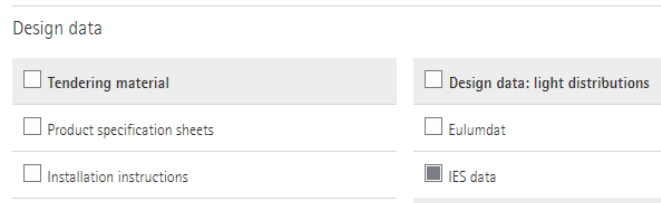
Vor dem Einsatz des Nodes benötigst du geeignete IES-Dateien. Diese gibt es gratis im Internet, z.B. bei *Lithonia Lighting*:

- Rufe auf: <https://lithonia.acuitybrands.com/>
- Klicke in der Kopfzeile auf Resources → Photometric Downloads
- Dort findest du eine Übersicht zu den verschiedenen Bereichen von IES. Du kannst die Dateien einzeln oder in Gruppen (als *.zip-Dateien) herunterladen.

Eine weitere Quelle ist die Firma *ERCO*:

- Rufe auf: <https://www.erco.com/de/>
- Klicke in der Kopfzeile auf Produkte → Alle Produkte (oder eine Auswahl)
- Es werden dir eine Fülle von Lampen gezeigt; wähle z.B. Wandleuchten → Starpoint

- Jetzt musst du weit herunterscrollen, bis du zu der Liste kommst, wo die einzelnen Artikel aufgeführt sind. Klicke dort auf eine Artikelnummer.
- Wieder musst du weit nach unten scrollen, bis du auf eine Übersicht mit dem Titel *Design Data* stößt. Dort schließlich findest du *IES data* und kannst diese downloaden.



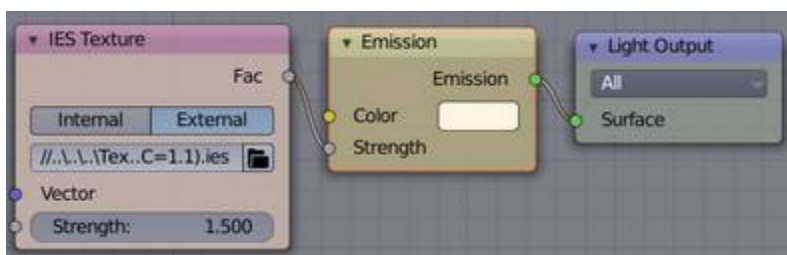
Wenn du eine Reihe von IES-Dateien heruntergeladen hast, brauchst du noch einen Viewer, um die verschiedenen Lichtformen schnell beurteilen zu können die sich hinter diesen Dateien verbergen:

- Rufe auf: <http://photometricviewer.com/>
- Download die neueste Version (>3.4)
- Rufe den IES Viewer auf und lade eine IES-Datei. Im Fenster unten links erscheint ein Schema, das die Beleuchtungsform zeigt.
- Wenn du auf das Glühbirnen Icon oben links klickst, öffnet sich ein Fenster mit der Beleuchtungsform:



Zurück zu Blender:

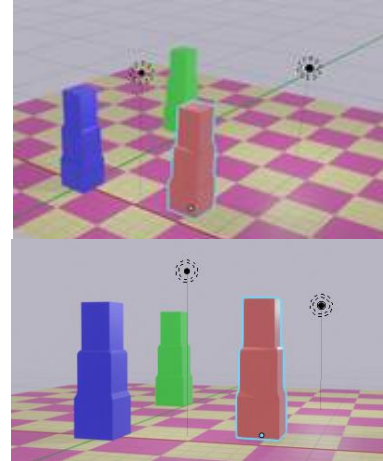
- Rufe eine Ebene auf und drehe sie in die Senkrechtposition.
- Platziere davor eine Point-Lampe.
- Rufe den zugehörigen Node auf. Die Lichtverteilung sieht bisher so aus:
- Füge jetzt einen *IES Texture*-Node hinzu und verbinde dessen Ausgang mit dem Eingang *Strength* des *Emission*-Nodes.
- Klicke im *IES* Node auf *External* und öffne eine der zuvor heruntergeladenen *IES*-Datei. Der Unterschied ist deutlich:



5.1.2 Nebel (volumetrisches Licht) in Eevee

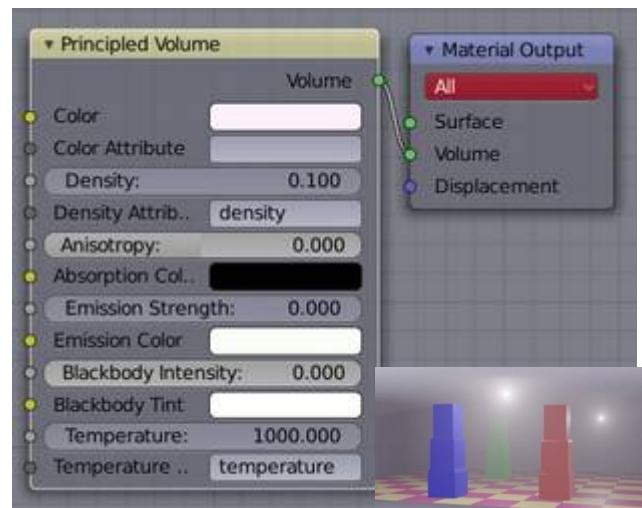
Volumetrisches Licht bezieht sich auf die Ausbreitung des Lichts in einem Körper, der nicht auf seiner Oberfläche eingefärbt wird, sondern im Inneren. Die Wirkung besteht in der Erzeugung von Nebel.

Wir gehen zunächst von dieser Szene aus: Auf einem Schachbrettmuster stehen drei Meshes; zwei Point-Lampen besorgen die Beleuchtung. (Bild oben: Schrägaufsicht; Bild unten: Kamerasicht)



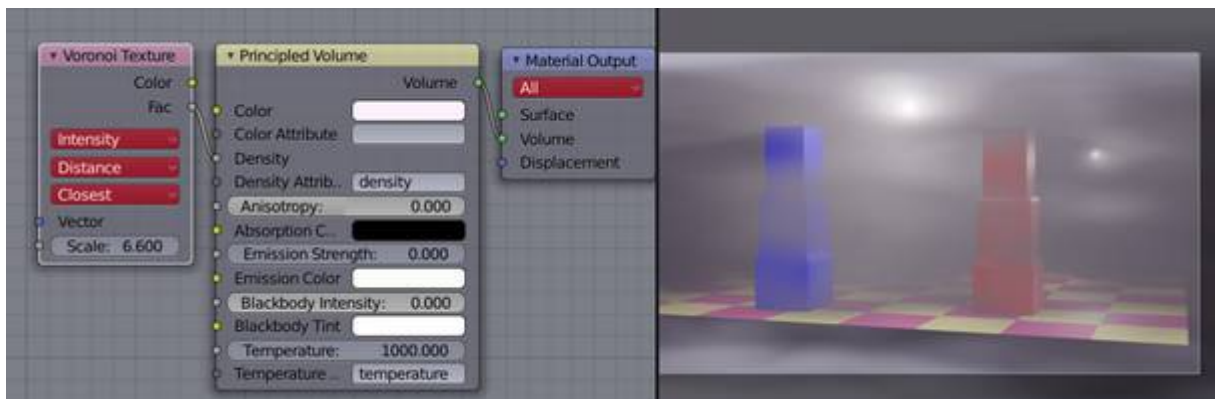
Erzeugung des Nebels an einem Beispiel

- Erzeuge einen Cube, der alle Objekte einschließt.
- Nimm in den Properties (Render) folgende Einstellungen vor:
 - Sampling:
 - Render = 300
 - Viewport = 200
 - Volumetrics:
 - Tile Size = 2px (höhere Auflösung)
 - Bloom:
 - Haken setzen
 - Radius = 1.4
- Selektiere den Cube und rufe ein neues Material auf.
 - Lösche im Shader Editor den Principled Shader.
 - Füge stattdessen den Principled Volume Shader ein.
 - Setze Density = 0.10
 - Verbinde den Volume-Ausgang mit dem Volume-Eingang des Material Output Shaders.
 - Im Color-Feld kannst du die Farbe des Nebels ändern.



Variante 1: Wolken

- Füge einen Voronoi Texture Node ein und verbinde dessen Ausgang Fac mit dem Eingang Density. Das Ergebnis lässt den Nebel wie Wolken aussehen:



- Probiere andere Texturen aus!

- Lösche die drei Point-Lampen und füge stattdessen eine Spotlight-Lampe ein:
 - Power = 1000W
 - Radius = 0.25m
 - Size = 50°
 - Blend = 1.0
- Setze die *Background Color* in *World* völlig schwarz.

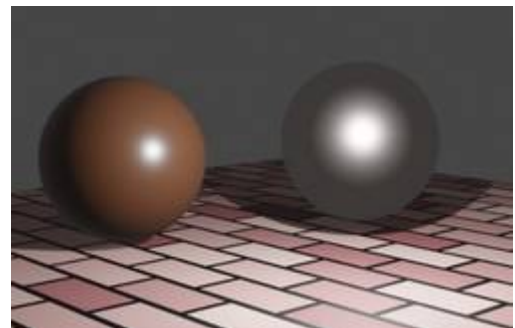
5.1.3 Light Probe

Reflection Cubemap und *Reflection Plane* sind Funktionen zur Erzeugung von Reflektionen.

Reflection Cubemap

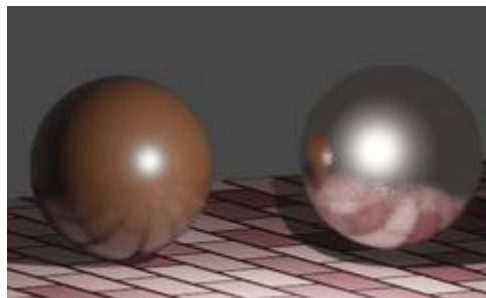
Du solltest die Rendered-Ansicht verwenden. Zu Veranschaulichung werden hier eine *Plane* mit Brick Texture zwei *UV Spheres* verwendet. Die Beleuchtung geht in einem *Point Light* aus.

Die linke Kugel hat ein non-metallic Material, die rechte ein metallic Material.



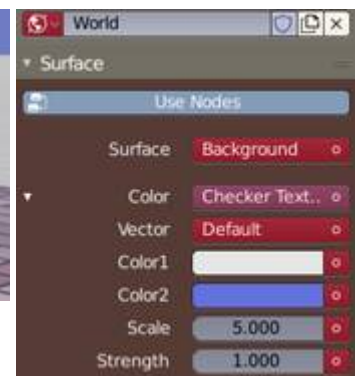
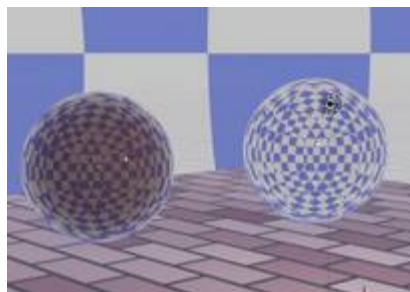
- Zu erkennen ist, dass die Brick-Textur von beiden Kugeln nicht auf der Unterseite gespiegelt wird.

- Eine Möglichkeit diese Spiegelung zu erzeugen, besteht in der Aktivierung der Funktion *Screen Space Reflections*, die du im Render-Menü rechts findest. Das Ergebnis sieht dann so aus. Aber das nur nebenbei.



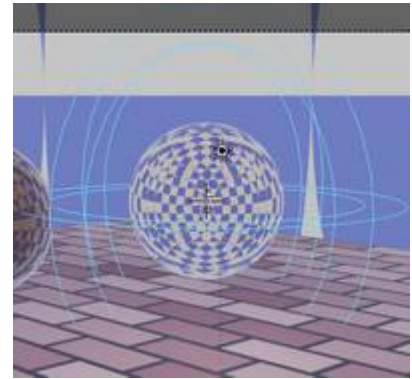
Deaktiviere die Funktion *Screen Space Reflections* bitte wieder.

- Wenn du dem *Background* z.B. eine *Checker*Textur gibst, kannst du erkennen, dass die Kugeln nicht nur die Lampe, sondern auch den Hintergrund spiegeln.



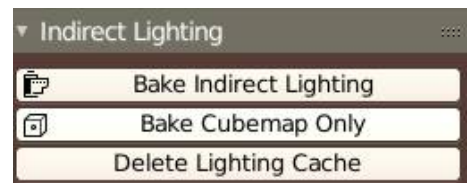
- Gehe so vor, um der rechten Kugel eine *Reflection Cube Map* hinzuzufügen:

- Selektiere die Kugel mit RMT. [shift] + [S] → Cursor to selected.
- [shift] + [A] → Light Probe → Reflection Cube Map



- Jetzt muss die Szene gebacken werden:

- Öffne im Render-Menü das Untermenü *Indirect Lighting*. Klick auf *Bake Cubemap Only*.
- Im Ergebnis zeigen sich schwarze Kreise auf der Kugel. Diese entstehen, wenn die Clipping-Punkte der Cubemap innerhalb des Objekt sind.



- Zur Beseitigung dieses Fehlers:

- Mache die Clipping-Grenzen sichtbar, indem du in den *Object Data* der *Reflection Cubemap* das Untermenü *Viewport Display* öffnest und dort einen Haken hinter *Clipping* setzt.

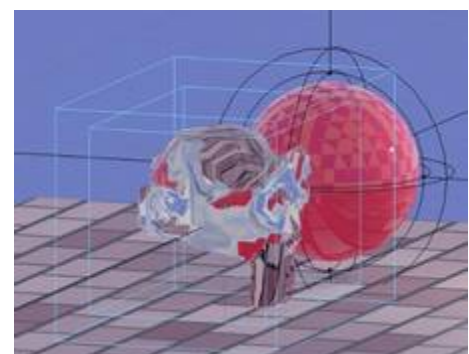
- In dem Untermenü Probe kannst du den *Clipping Start* einstellen. Dieser sollte **außerhalb der Kugel**, aber **oberhalb der Bodenfläche** liegen.



- Du kannst in die Cubemap auch die Linke Kugel einbeziehen, indem du sie skalierst. Einfacher ist es, wenn du die Cubemap duplizierst und dann dieses Duplikat auf die linke Kugel ziehst.

- Glas-Material

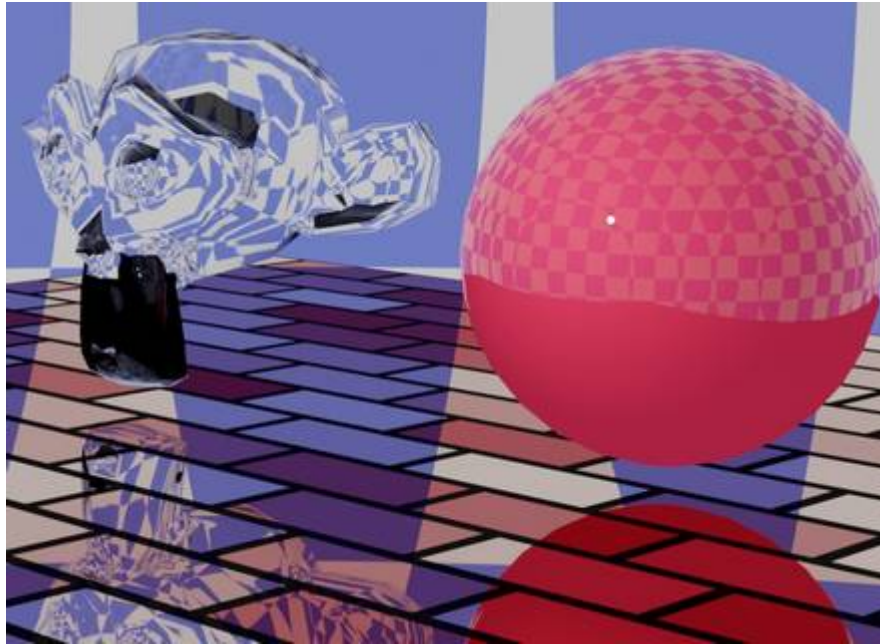
- Erzeuge den *Monkey Suzan* anstelle der linken Kugel und ordne ihm Glas-Material zu.
- Selektiere Suzan mit RMT. [shift] + [S] → Cursor to selected.
- [shift] + [A] → Light Probe → Reflection Cube Map
- Ändere im Untermenü *Probe* im Feld *Type* von *Sphere* in *Box*.
- Färbe zur besseren Sichtbarkeit die Kugel rot ein und wechsle zu non-metallic Material.
- Im Glas-Material von Suzan spiegeln sich jetzt sowohl die Kugel als auch Boden um Hintergrund.



Reflection Plane

Die Reflection Plane ist für flache Objekte geeignet. Hier soll sie auf den Fußboden angewendet werden.

- Selektiere den Boden mit RMT. [shift] + [S] → Cursor to selected.
- [shift] + [A] → Light Probe → Reflection Plane
- Skaliere die Plane etwas größer als den Boden und weise ihr Glas-Material zu.
- Bake; jetzt reflektiert der Boden sowohl Monkey als auch die rote Kugel.

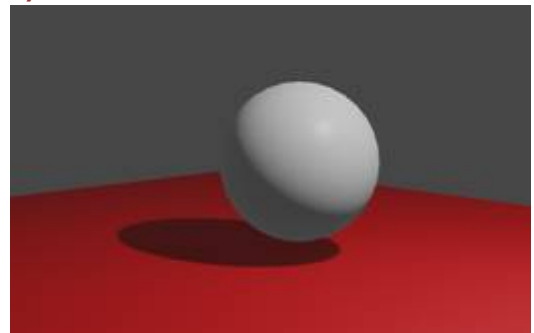


<https://www.youtube.com/watch?v=JzilEjipws>

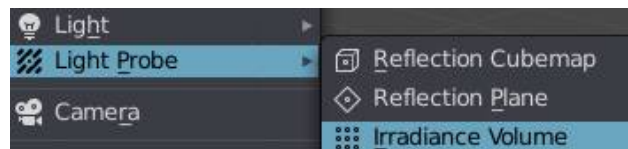
Indirekte Beleuchtung (Irradiance Volume)

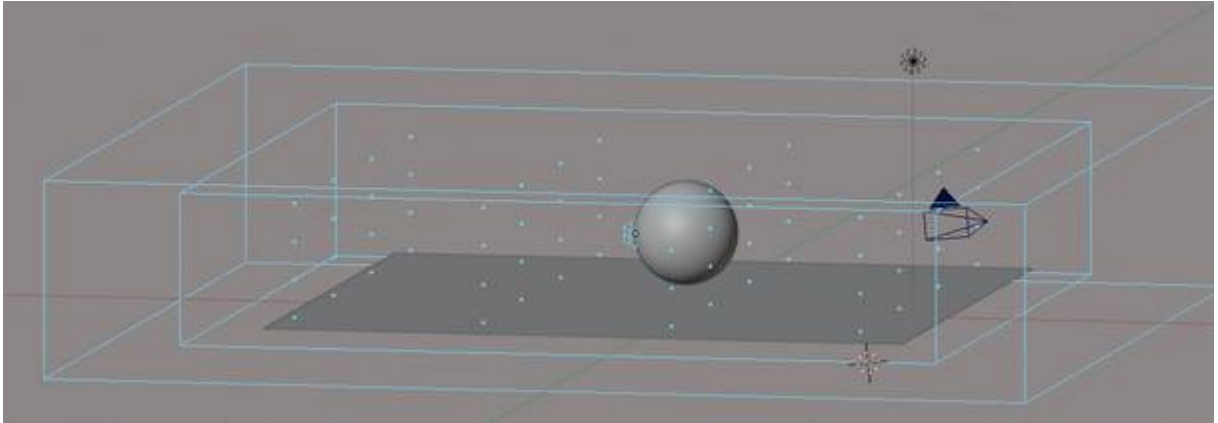
Zu Veranschaulichung wird hier eine rote *Plane* mit einer weißen *UV Sphere* verwendet. Die Beleuchtung geht in einem *Point Light* aus.

Allerdings findet hier zunächst keine indirekte Beleuchtung statt: Von der roten Fläche wird kein Licht zurückgestrahlt; die Unterseite der Kugel hat dieselbe Farbe wie der Hintergrund. (Wenn du auf Cycles umschaltest, wirst du sehen, dass unter diesem Renderer die Unterseite der Kugel rot angestrahlt wird.)



- Um in Eevee indirekte Beleuchtung zu ermöglichen, brauchst du ein weiteres Objekt: [shift] + [A] → Light Probe → Irradiance Volume.
- Skaliere das neue Objekt in der Weise, dass die Plane und die Kugel beide eingeschlossen sind.





- Öffne jetzt in den Properties das Render-Menü und dort das Untermenü *Indirect Lighting*; klicke dort auf den Button *Bake Indirect Lighting*. Jetzt haben wir eine indirekte Beleuchtung: Die Kugel wird auf der Unterseite rot angestrahlt.
- Wenn du weitere Objekte mit verschiedenen Farben in die Nähe der Kugel stellst, werden auch diese indirekte Beleuchtung erzeugen (immer *Bake* aktivieren!).
- Die Punkte innerhalb des IrradianceVolume-Objekts kennzeichnen die Auflösung.  Wenn du auf den zugehörigen Button  klickst, öffnet sich das Menü:
- Die Auflösung kann variiert werden, indem in den drei Feldern *Resolution X, Y und Z* die Werte verändert werden. Im Einzelfall kann dies Auswirkungen auf die Form der indirekten Beleuchtung haben.



5.1.4 Fluoreszenz (Bloom)

Du kannst bewirken, dass die Gegenstände einer Szene zu leuchten beginnen, wenn sie angestrahlt werden. Die englische Bezeichnung *Bloom* lässt sich am besten mit *Fluoreszenz* übersetzen.

- Setze im Render Menü (Properties) einen Haken vor das Untermenü *Bloom* und öffne es. Die einzelnen Parameter bedeuten:

Threshold: Schwelle (Helligkeit der Pixel), ab der ein Bloom-Effekt erzeugt werden soll

Knee: Sanftheit des Übergangs zwischen der Wirkung unter- und oberhalb der Schwelle

Radius: Weite der Auswirkung des Effekts

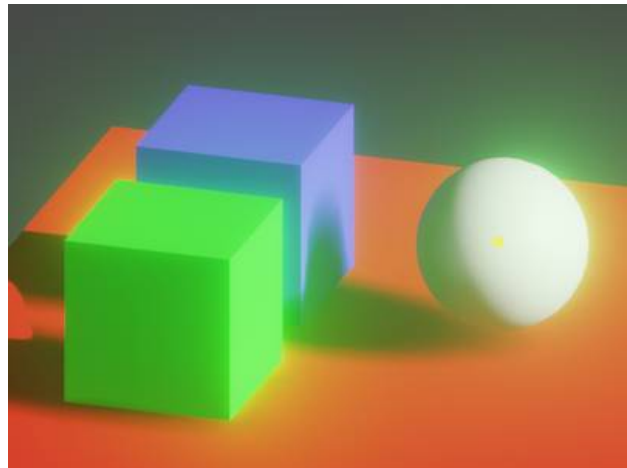
Color: Farbe des Effekts

Intensity: Effektstärke (Beim default-Wert von 0.05 ist kein Effekt zu bemerken.)

Clamp: Maximalintensität eines Bloom-Pixels; bei 0.0 ist diese Funktion ausgeschaltet.



Für das Beispielbild wurden die oben gezeigten Einstellungen verwendet:



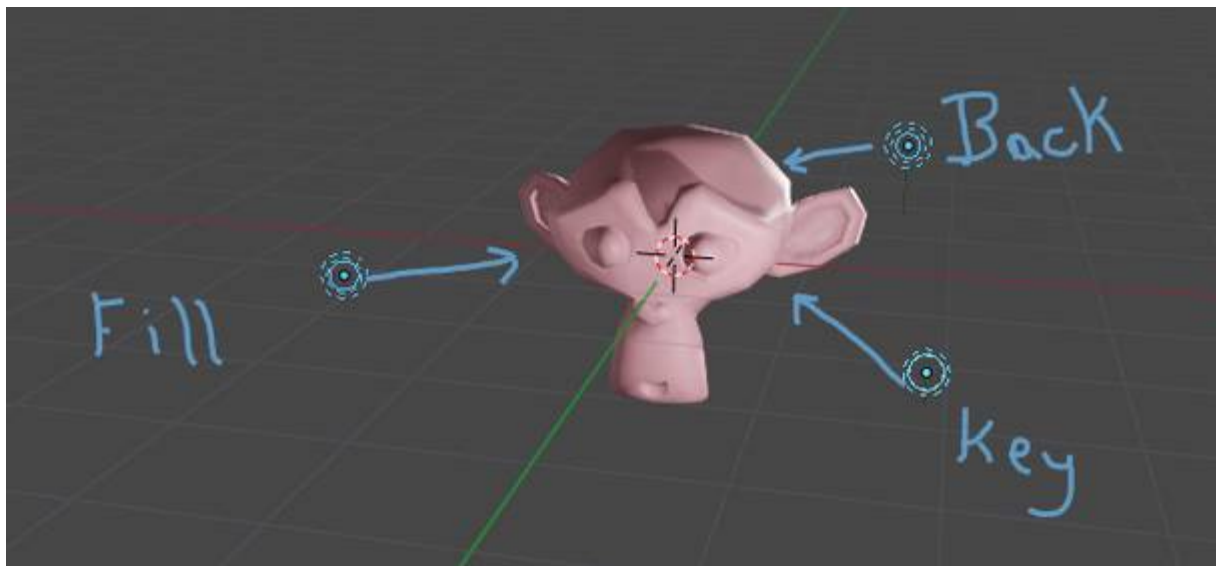
5.1.5 Addon Tri-lighting

Das Prinzip

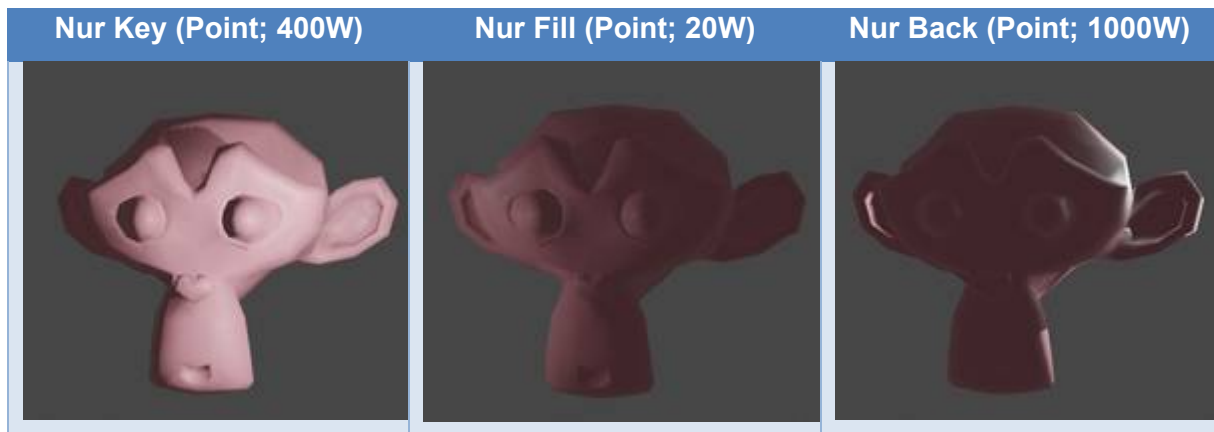
Das Three-point Lighting System ist bei Fotos und Film die gängige Methode zur Ausleuchtung von Szenen.

Die Szene wird mit drei Lampen ausgeleuchtet:

1. Key Light: Das ist die Hauptlichtquelle, von dieser erhält das Objekt das meiste Licht.
2. Fill Light: Hiermit werden Schattenpartien aufgehellt; es ist daher weitaus schwächer als das Key Light. Hiermit wird auch der Kontrast weniger scharf.
3. Back Light (auch Rim Light): Hiermit wird das Objekt von hinten angeleuchtet und erhält so einen leichten Lichtsaum.



Die Wirkung der einzelnen Lampen wird erkennbar, wenn jeweils nur eine Lampe aktiv ist:

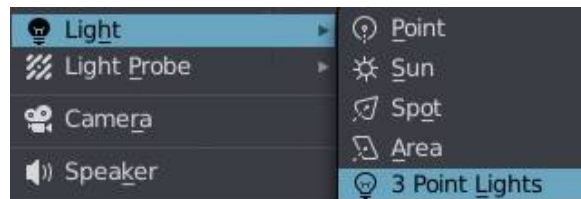


Das Addon Tri-Lighting

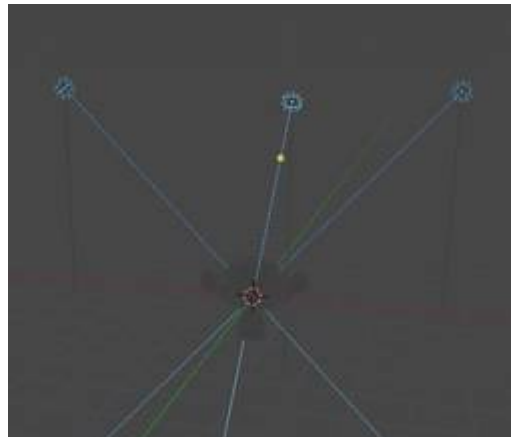
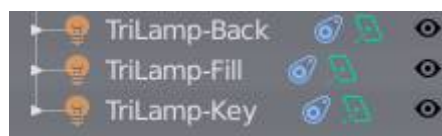
- Installation
 - Öffne die Preferences: Edit → Preferences
 - Klick auf Add-ons
 - Klick in der Liste der Addon-Kategorien auf *Lighting*.
 - Setze einen Haken vor *Lighting: Tri-lighting*

- Aufruf:

- Selektiere das zu beleuchtende Objekt.
- [shift] + [A] → Light → 3 Points Light



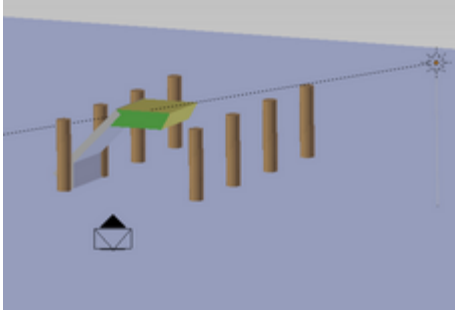
- Das Ergebnis: Es werden drei Area-Lampen installiert. Die Lampen sind allerdings weder im Hinblick auf die Power (1.5W – 3.0W) noch auf die Positionierung unmittelbar einsatzfähig. Da ist dann immer noch eine Menge Eigenarbeit erforderlich.
- Im Outliner sind die Lampen übrigens konventionsgemäß benannt:



5.2 Besonderheiten der Beleuchtung im Cycles Renderer

5.2.1 Indirekte Beleuchtung

Nehmen wir als Beispiel nebenstehende Szene: Da gibt es gerade und schräg stehende Flächen, ein paar Pfähle, eine Lichtquelle (sun) und die Kamera.



- Die Berechnungsmethode bei *Cycles* zeigt sich am auffälligsten an der Unterseite des „Dachs“: Das Licht, das auf den Boden fällt, wird zurückgeworfen und beleuchtet daher diese Fläche, so dass die gelbe Farbe sichtbar wird.
- Des Weiteren fallen auch diese gelben Lichtstrahlen wieder auf den Boden und erzeugen dort einen gelben Schimmer.
- Ähnlich kannst du beobachten, dass die braune Farbe des Pfahls links vorne bei *Cycles* auf die Seite der Dachkonstruktion abstrahlt.
- Mit dem *Cycles Renderer* wird also auch das indirekte Licht berechnet.

Lichtpfade

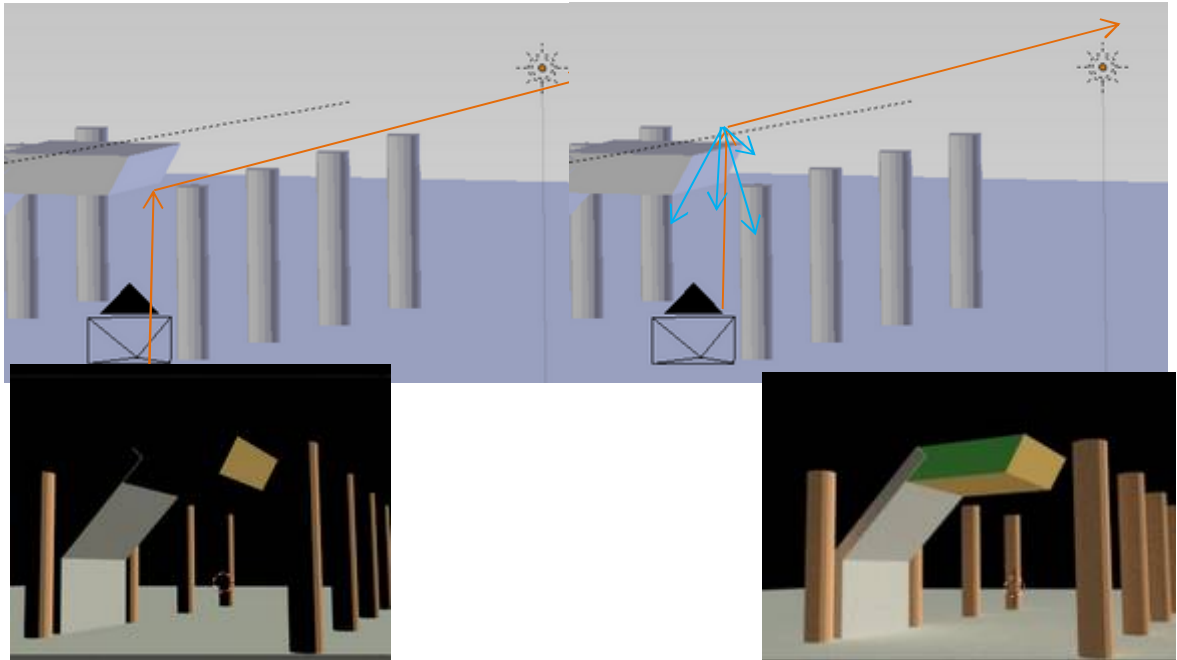
Die Wege, die dieses indirekte Licht nimmt, werden als *Light Paths* bezeichnet. Dies ist auch der Name eines Untermenüs beim Renderer.

- Die Überschrift *Max Bounces* bezieht sich auf die Anzahl, mit der ein Lichtstrahl von einer Oberfläche abprallt, auf die er zuvor getroffen ist. Wenn du dort den *Total*-Wert auf Null stellst, wird nur die erste Oberfläche berücksichtigt, auf die ein Lichtstrahl trifft. Wenn dort also ein Wert von 12 eingestellt ist, werden 12 *Bounces* („Abpraller“ des Lichtstrahls) berechnet.



- Normalerweise geht man davon aus, dass die Lichtstrahlen von einer Lichtquelle ausgehen, auf eine Oberfläche treffen, dort abprallen und dann auf die Kamera (oder das Auge) treffen. *Cycles* geht bei der Berechnung einen anderen Weg, nämlich von der Kamera aus auf ein Pixel (einer Oberfläche) und „sucht“ dann von dort aus nach einer Lichtquelle, die dieses Pixel beleuchtet.

- Bei einem Wert *Total* = 0 wird nur nach dieser einen Lichtquelle gesucht (Bild links). Wenn hingegen der Total-Wert höher gesetzt wird, werden von diesem Punkt ausgehend weitere Pfade untersucht und geprüft, ob noch weitere Lichtquellen bei der Berechnung berücksichtigt werden müssen. Wenn dabei weitere beleuchtete Flächen „entdeckt“ werden, die Licht abstrahlen, werden deren Farbe und Lichtintensität ebenfalls berücksichtigt (Bild rechts).



- Der (default-)Wert *Total* = 12 dürfte für die meisten Zwecke reichen, zumal *Cycles* die Bounces ohnehin nach einer bestimmten Anzahl abbricht, wenn mehr Bounces voraussichtlich zu keiner Verbesserung mehr führen würden.
- Bei den Werten für *Diffuse*, *Glossy* und *Transmission* nützt es nichts, wenn diese größer als der Total-Wert sind.
 - ✚ *Diffuse* bezieht sich auf diffuse Oberflächen.
 - ✚ *Glossy* bezieht sich auf spiegelnde Oberflächen. Wenn es nur darum geht, spiegelndes Material darzustellen, reichen 4 Bounces aus. Bei anderen Objekten (z.B. Schmuck) können höhere Werte angebracht sein.
 - ✚ *Transmission* bezieht sich auf durchsichtige Oberflächen. Der eingetragene Wert meint dann Anzahl z.B. von Gläsern, durch die ein Lichtstrahl hindurchgeht. Wenn du also z.B. 8 Glasscheiben hintereinander setzt, aber *Transmission* = 7 setzt, kannst du durch die 8. Scheibe nichts mehr sehen. – Wenn du durch wirklich viele Gläser hindurchsehen willst, sollten folgende Einstellungen vorgenommen werden:

<i>Total</i>	= 128
<i>Diffuse</i>	= 5
<i>Glossy</i>	= 5
<i>Transmission</i>	= 128

So ist die Durchsicht gewährleistet, ohne dass für **Diffuse-** oder **Glossy-**Bounces unnötig Rechenkapazität gebunden wird.

Selbstleuchtende Materialien

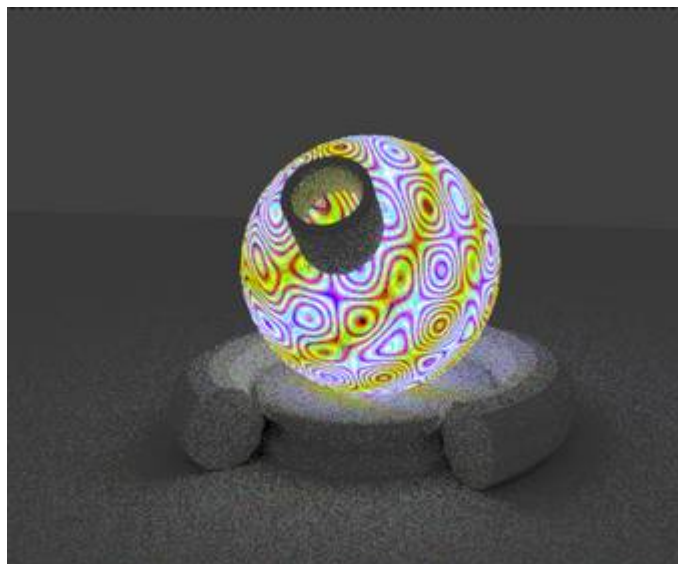
- Auch unter dem *Eevee* ist es möglich, einzelnen Materialien die Eigenschaft *Emit* zu verleihen. Diese Materialien leuchteten dann zwar selbst, allerdings fällt dieses Licht nicht auf andere Objekte, diese bleiben also unbeleuchtet.
- Das ist unter dem Cycles Renderer anders: Jedem beliebigen Objekt kann dort Leuchtkraft gegeben werden. In unserem Beispielbild wurde der Kugel das Material *Emission* zugewiesen. Sie leuchtet jetzt nicht nur selbst, sondern beleuchtet auch die übrigen Gegenstände.
- Es ist also unter Cycles nicht unbedingt erforderlich, eigens Lampen zu installieren.



17

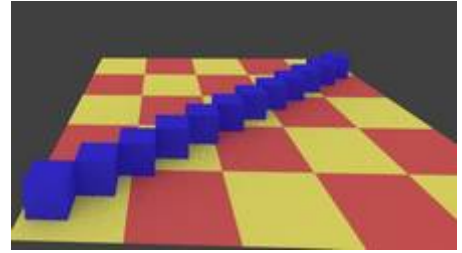
Beispiel Texturen

- Es ist darüber hinaus möglich, auf die Leuchtoobjekte Texturen zu projizieren. Hier wurde *Magic Texture* eingesetzt.



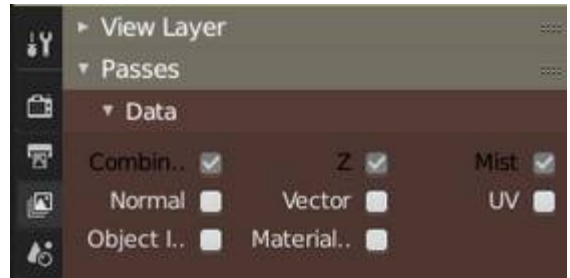
Mist bei Cycles

Als Beispiel verwenden wir ein Array blauer Würfel, diesmal auf einem rot-gelben Schachbrett-Untergrund.



18

- Zunächst musst du in den *Properties* das Panel *View Layers* aufrufen. Dort musst du im Unter-Panel *Passes* ein Häkchen nach *Mist* setzen.



- Erst wenn du das getan hast, erscheint im *World*-Panel ein neues Unter-Panel mit der Bezeichnung *Mist Pass*:



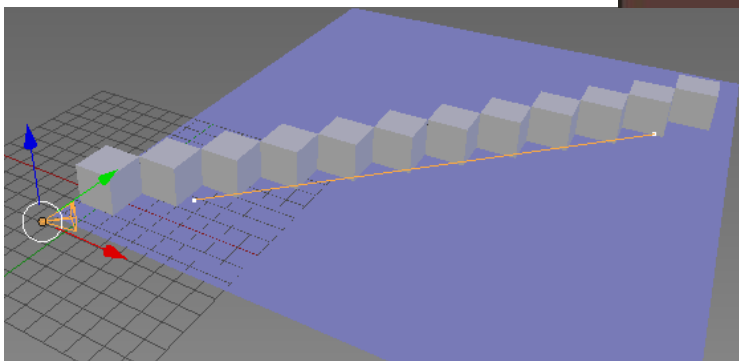
- Die Eintragungen in den Feldern haben dieselbe Bedeutung wie bei BI:

Start: Der eingetragene Wert bestimmt, ab welcher Entfernung (in Blender-Einheiten) der Nebel beginnt. Bei einem hohen Wert kann man anfangs noch gut sehen, erst dann beginnen die Umrisse zu verschwimmen. Bei 0.0 ist bereits der erste Würfel unscharf.

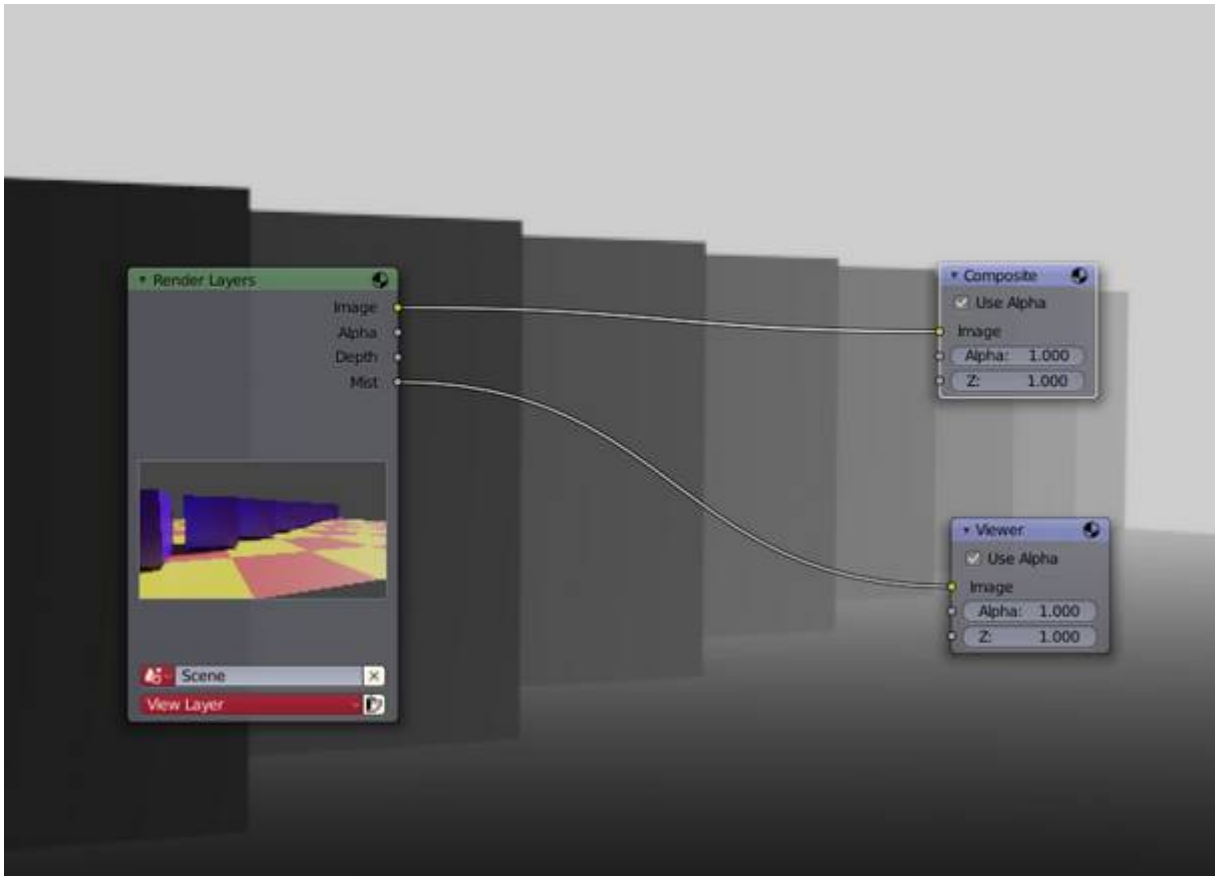
Depth: Ab welcher Entfernung wird der Nebel immer dichter?

Falloff: Berechnungsart zum Stärkerwerden des Nebels. Es stehen zur Verfügung: *Quadratic*, *Linear* und *Inverse Quadratic*.

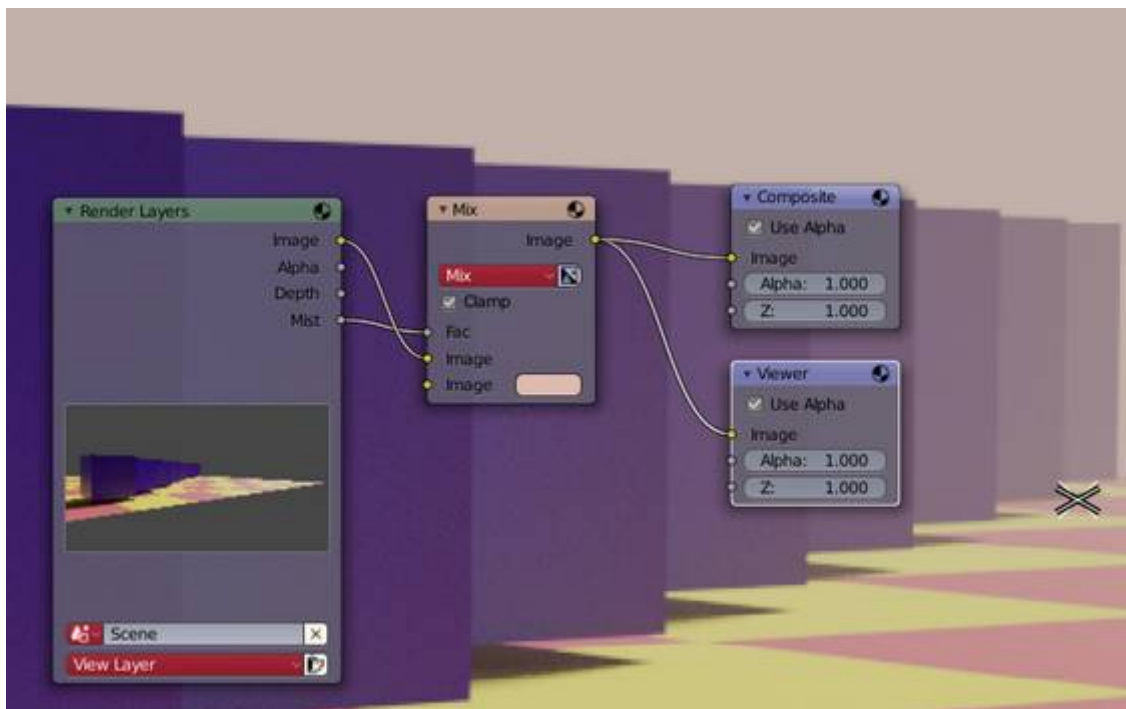
- Wenn du die Kamera selektierst und dann im Unterpanel *Viewport Display* der **Kamera** das Kästchen *Mist* mit einem Häkchen versiehst, kann du die eingestellten Entfernungen sichtbar machen:



- Wenn du jetzt [F12] drückst, um zu rendern, ist von Nebel noch nichts zu sehen. In der Compositor-Ansicht kannst du aber sehen, dass der Node des gerenderten Bildes einen zusätzlichen Ausgang hat: *Mist*. Wenn du vom *Mist*-Ausgang des Nodes *Render Layers* eine Verbindung zum *Viewer*-Node ziehst, kannst du im *Backdrop* schon mal den Nebel-Effekt sehen – allerdings zunächst nur schwarz/weiß.



- Der einfachste Weg, nun Farbe in das Bild zu bekommen ist der, einen Mix-Node zwischenschalten:



- Über das Farbfeld kannst du sogar die Farbe des Nebels verändern: