

Blender 3D  
Summer  
School

# Blender kennenlernen

Ausgabe 2022



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>Einleitung &amp; Grundlagen</b>	<b>4</b>
Für wen ist dieses Ebook	4
Was ist 3D	4
Es werde Licht	4
In Blender läuft das Licht rückwärts	5
Wir sezieren ein 3D-Objekt	6
Der Kamera Standpunkt	7
Kleine Materialkunde	8
<b>Blender</b>	<b>9</b>
Der erste Start	9
Grobe Orientierung	9
Einstellungen	10
Workspaces	10
Fenster-Typen	12
Gekachelte Oberfläche	13
3D-View	15
<b>Formen erschaffen und verändern</b>	<b>17</b>
Primitive Objekte einfügen	17
Objekteigenschaften ändern	18
Der Röntgen-Blick	19
Edit-Mode	20
<b>Packpapier einfärben</b>	<b>21</b>
Materialien	21
Shader & Nodes	22
Flächen unterschiedlich einfärben	24
Wohin das Packpapier aufgeklebt wird	25
Ein UV-Layout anlegen	26
<b>Licht &amp; Ausgabe</b>	<b>28</b>
Beleuchtung	28
Render-Kamera	29
Rendern mit Cycles	30
<b>Glossar</b>	<b>31</b>
<b>Rettungs-Tastenkürzel</b>	<b>33</b>

Dieses EBook ist eine Produktion der Blender3dSchool.de, 2022.

Inhalte & Abbildungen können beliebig verwendet werden unter Angabe folgenden Urhebernachweises (Creative-Commons CCBy-Lizenz):

Blender kennenlernen, Blender3dSchool.de, EBook, 2022.



# 1. Einleitung & Grundlagen

## 1.1. Für wen ist dieses Ebook

Das Ebook wird verfasst von DozentInnen der Blender-3D-School, die dem Newbee-Track zugeordnet sind. Es soll generell 3D-EinsteigerInnen und speziell Blender-EinsteigerInnen unterstützen in die 3D-Welt einzutauchen und das 3D-Programm Blender zu erlernen.

Während das sehr umfangreiche und ausführliche [Blender-Handbuch](#) zu allen und jeden Buttons und Funktionen Auskunft geben kann und neben der Geschichte von Blender auch Hinweise zur Installation gibt, ist dieses Ebook eher ein Leitfaden zu den aller wichtigsten Punkten, um möglichst schnell ein Überblick über die Arbeit mit Blender zu erhalten.

## 1.2. Was ist 3D

Die Welt des 3D-Crafting (zu deutsch der Erschaffung von 3D-Werken) ist groß. Die meisten Menschen kennen 3D-Erzeugnisse aus Computer- und Konsolenspielen oder aus dem Kino und verbinden mit 3D die Erschaffung neuer bzw. alternativer Welten. Die Arbeitsschritte die notwendig sind, um ein solches Spiel oder einen solchen Film herzustellen werden wir im Rahmen dieses EBooks kurz ankratzen, aber wir sollten nicht verschweigen, dass für die Fähigkeit einen Kino-Blockbuster oder ein AAA-Game zu erzeugen eine solide Ausbildung und oder viele Jahre des motivierten Selbststudiums und der praktischen Arbeit notwendig sind. Außerdem wird ein ganzes Team benötigt, organisiert in Departments, die jeweils eigenen Berufsbildern innerhalb der 3D-Welt entsprechen.

Darüberhinaus umfaßt der 3D-Bereich auch die Herstellung physikalisch realer Objekte im 3D-Druck und das Digitalisieren physikalischer Objekte mit 3D-Scannern. Nützliche 3D-Anwendungen sind überaus zahlreich, z.B.

- Die schon erwähnten Produkte der Unterhaltungsbranche
- Prothesen- und Ersatzteilherstellung mit 3D-Druck
- Medizinische Therapien in VR
- Konservierung und Archivierung von Kulturgütern
- Wissenschaftliche Simulation und Visualisierung
- Entwurf, Konstruktion und Visualisierung von Produkten

Die Software Blender kann für nahezu alle dieser Bereiche eingesetzt werden, so wie ein großes schweizer Taschenmesser des 3D-Crafting. Es gibt natürlich auch noch viele andere 3D-Softwares, die auf die jeweiligen Aufgaben spezialisierter ausgerichtet sind.

## 1.3. Es werde Licht

Menschen und Tiere haben Sinnesorgane um die Umgebung wahrzunehmen. Das Auge des Menschen besitzt jeweils eine Retina im hintenliegenden Augapfel welche Lichtstrahlen in elektrische Impulse umwandelt und sie an das Gehirn weiterleitet. Bevor die Lichtstrahlen aber durch die Linse in das Auge treffen, werden diese eventuell an Objektoberflächen reflektiert. Durch die Materialbeschaffung der Objekt-Oberflächen werden den auftreffenden Lichtstrahlen bestimmte Farbfrequenzen entzogen, so dass

die auf die Retina auftreffenden Lichtstrahlen so Informationen der Materialoberfläche (Farbe und Helligkeit) an des Betrachter's Auge weitergeben. Gespeist werden die Lichtstrahlen von einer Lichtquelle. Die größte Lichtquelle in unserem Sonnensystem ist der Stern im Zentrum genannt Sonne, welche Licht in einem großen Farbspektrum aussendet. Die unterschiedlichen Farben des von der Sonne ausgesendeten Lichts können z.B. gut in einem Regenbogen betrachtet werden. Das Sonnenlicht erscheint uns aber meist weiß, da sich die Farben bei Überlagerung zu weiß addieren. Zusammengefasst ist der zeitliche Ablauf wie folgt:

1. Die Lichtquelle (z.B. die Sonne) sendet Lichtstrahlen aus
2. Die Lichtstrahlen werden von Objekten reflektiert und entziehen ihnen dabei bestimmte Farbanteile des Lichts
3. Die Lichtstrahlen fallen in das Auge - die Kamera des Menschen - und treffen letztendlich auf die Retina, an der sie die Materialinformationen der besuchten Objekte "übergeben"

## 1.4. In Blender läuft das Licht rückwärts

In Blender (und auch allen anderen mir bekannten 3D-Anwendungen) läuft dabei das Licht genau in umgekehrter Reihenfolge:

1. Die Kamera schießt Strahlen (englisch Rays) durch kleine Öffnungen (die Bildschirmpixel) in die virtuelle Welt (siehe Abb. 1)
2. Die Strahlen treffen auf virtuelle Objektoberflächen, welche eine bestimmte Helligkeit und Farbe besitzen
3. Genau in dieser Farbe und Helligkeit wird nun der jeweilige Ursprungspixel auf dem Bildschirm eingefärbt

Der Prozess des Aussendens und Reflektieren von Strahlen und Einfärbens von Pixeln auf einem 2-dimensionalen Bildschirm wird **Rendern** genannt. Das Einfärben als Teil des Renderprozess wird **Shaden / Shading** genannt.

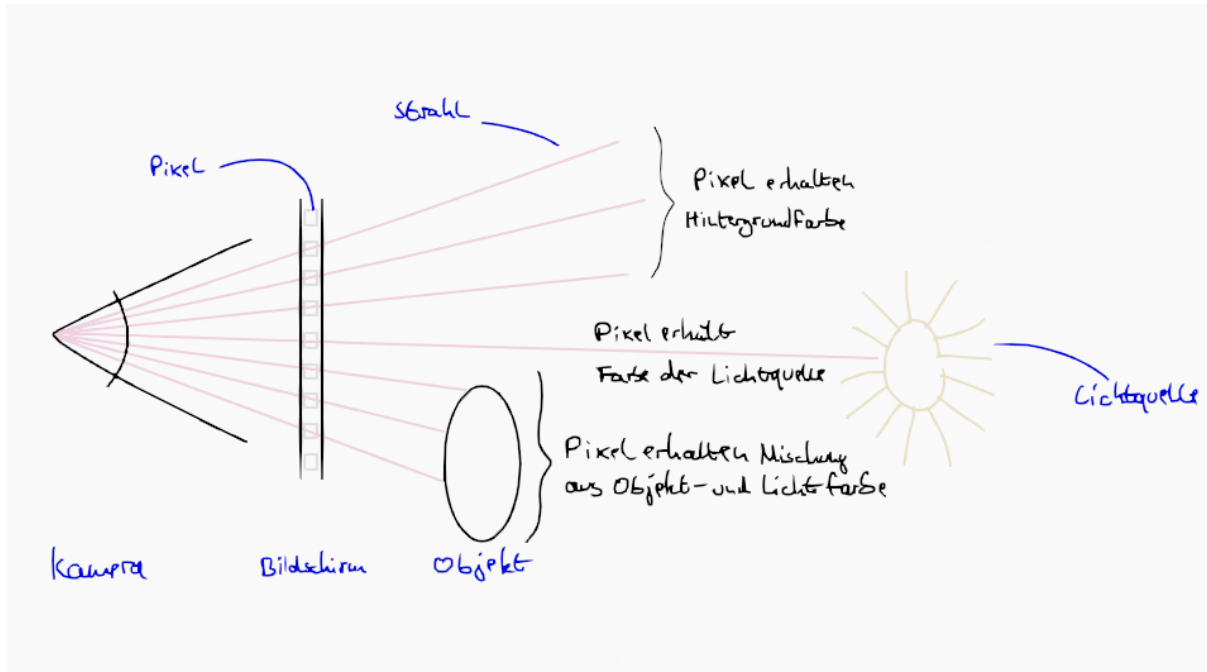


Abb. 1: Der Laufweg von Strahlen beim Rendern

Der Grund für die Invertierung des Lichtwegs liegt in der Beschränkung der Grafikkarten-Hardware. Diese kann es sich schlicht und einfach nicht leisten alle Lichtstrahlen, die von einer Lichtquelle ausgesandt werden, zu verfolgen, denn viele dieser Lichtstrahlen werden niemals auf eine Kamera treffen und dementsprechend von niemanden gesehen. Indem der Spieß umgedreht wird, muss sich die Grafikkarte daher nur mit den Lichtstrahlen beschäftigen, welche definitiv auch "gesehen" werden.

## 1.5. Wir sezieren ein 3D-Objekt

Objekte der realen Welt bestehen aus Molekülen und Moleküle bestehen aus Atomen, die wiederum aus Elektronen und Protonen. Selbst ein kleiner Sandkorn besteht immer noch aus sehr sehr vielen Atomen (es wird von circa 42 Trillionen Silizium- und Sauerstoff-Atomen ausgegangen). Diese vielen Atome können mit unserer 3D-Hardware zur Zeit nicht abgebildet werden, weswegen wir etwas "schummeln" müssen. Virtuelle Objekte (egal ob Sandkörner oder Sandstein-Schlösser) setzen wir wie folgt zusammen: Zuerst wird das Objekt mit einem Drahtgitter (engl. **Wireframe** oder **Mesh**) in Form gebracht (siehe Abb. 2). Hierdurch erhalten virtuelle Objekte auch ein Volumen (wenn sie nicht gerade dünne Tischdecken sein sollen). Dieses Drahtgitter wird dann mit Packpapier überzogen, welches nach Belieben angemalt werden kann. Hierdurch erhält das Objekt neben der Form nun auch eine Oberfläche mit Helligkeit und Farbe. Die Summe aller Packpapiereigenschaften, u.a. die Farbe und Helligkeit, wird als **Material** zusammengefasst.

Die Grafikkarte muss sich nun nur die Drahtgitter-Punkte merken und welche **Punkte** (engl. Vertex) zu welcher **Kante** (engl. Edge) und welche Kante mit welchem Packpapier überzogen wird, so dass eine **Fläche** (engl. Face) entsteht. Diese Kanten-überspannenden Packpapier-Flächen werden auch **Polygone** genannt. Sind genau 3 Punkte an der Fläche beteiligt, spricht man von Dreieck (engl. Triangle oder kurz **Tri**), bei 4 Punkten nennt man die Fläche auch Quadrat (engl. Quadrat, oder kurz **Quad**), bei mehr als 4 Punkten wird in Blender der Begriff **NGon** genutzt.

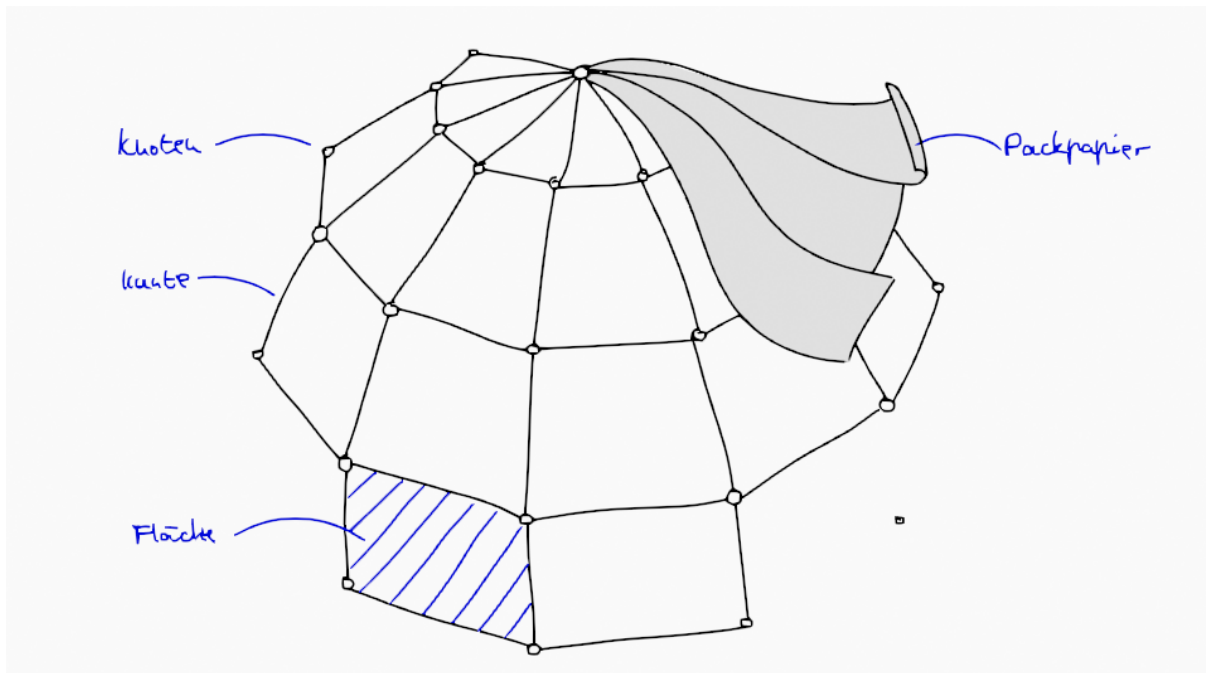


Abb. 2: Das Drahtgitter-Modell mit draufgeklebtem Packpapier

Um nun für das Rendering die Pixel auf dem Bildschirm richtig einzufärben, braucht man "nur" schauen auf welches Polygon ein bestimmter Strahl trifft und welches Packpapier / Material diesem Polygon zugeordnet wurde.

## 1.6. Der Kamera Standpunkt

Alle virtuellen Objekte, sowie die Kamera, durch welche diese Objekte betrachtet werden, sind Teil der gleichen 3D-**Szene**. Nur die Objekte, welche für die Kamera sichtbar sind, werden auf dem **Render-Bild** oder Video erscheinen. Durch Verschieben & Rotieren der Kamera kann der Blickwinkel und die Blickposition verändert werden, so dass weitere Objekte in das Blickfeld der Kamera geraten, während andere das Blickfeld verlassen.

Statt die Kamera zu bewegen, können sich aber genauso auch die Objekte bewegen und damit das Kamera-Blickfeld verlassen oder betreten (siehe Abb. 3).

Jede Kamera hat neben dem Öffnungswinkel aber auch eine minimale und eine maximale Objektentfernung in der sich zu rendernde Objekte befinden dürfen. Der Grund für die maximale Entfernung liegt wieder an den knappen Hardware-Ressourcen. Die Begrenzung der Sichtweite kann man sich wie einen Nebelschleier vorstellen, der weit entfernte Objekte verbirgt. Was verborgen ist, bewahrt die Grafikkarte (**GPU**) vor zuviel Arbeit.

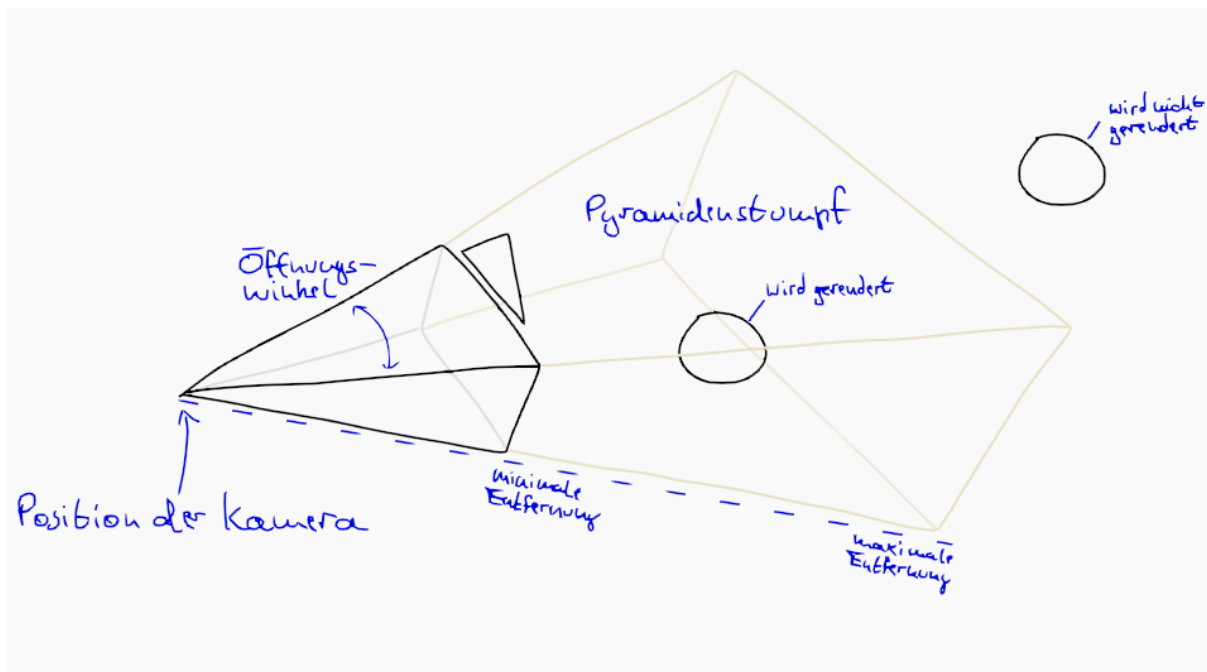


Abb. 3 Nur Objekte im Blickfeld und innerhalb der minimalen und maximalen Entfernung der Kamera werden gerendert, dieser Bereich wird auch Pyramidenstumpf genannt.

## 1.7. Kleine Materialkunde

Echte, physisch-anfassbare Objekte sind aus einem bestimmten Stoff gemacht. Wir nennen diesen Stoff in der 3D-Produktion "**Material**" und versuchen meist die "virtuellen" 3D-Materialien so gut es geht den Natur-Stoff-Vorbildern anzugleichen (manchmal will man sich aber auch bewusst absetzen, z.B. bei Cartoon-Shadern). Damit wir die echten Stoffe gut nachahmen können, ist ein Verständnis förderlich, welche unterschiedlichen Charakteristika einen echten Stoff ausmacht.



### 1. Farbe & Helligkeit

Farben sind das stärkste visuelle Identitätsmittel, um einen Stoff oder ein Objekt wiederzuerkennen und spielen im Leben aller Menschen / Tiere eine große Rolle (Angst = rot, Entspannung = grün, Helligkeit = Licht = Futter etc.). Die Farbe Orange bekam gar ihren Namen von einer bekannten Frucht namens Orange, die sehr markant gefärbt ist. Farben lassen sich meistens sehr gut virtuell nachahmen, da der Farbstandard "sRGB" der heutigen Monitore relativ groß ist (bestimmte Blau- oder Grüntöne können allerdings nur auf speziellen Monitoren dargestellt werden).

### 2. Oberflächenbeschaffenheit, Rauheit & Reflektion

Die Mikro- und Nanostruktur einer Oberfläche hat einen großen Einfluss auf das reflektierte Licht. Die bläulich schimmernden Schmetterlingsflügel sind nicht etwa mit blauen Pigmenten ausgestattet, sondern haben eine speziell-gerasterte Struktur aus denen nur der blaue Lichtanteil der Sonnenlichts entkommt. Bei perfekt glatten Oberflächen spiegelt sich die Umgebung, je rauher aber eine Oberfläche ist, desto mehr wird das reflektierte Licht gestreut und das Spiegelbild verschwimmt oder ist überhaupt nicht sichtbar. Wirkt eine Oberfläche schwarz, wird ein Großteil des ankommenden Lichts durch die Oberfläche absorbiert.

### 3. Lichtdurchlässigkeit

Durch Glas und Wasser kann man hindurchschauen, hier wird das Licht lediglich am Materialübergang (Luft zu Wasser und Wasser zu Luft) gebrochen. Andere Stoffe sind undurchsichtig, lassen aber trotzdem Licht durchscheinen (z.B. Haut, gut zu beobachten bei den Ohren des Menschen) und wieder andere Stoffe sind extrem Licht-schluckend, z.B. dunkle, dicke, schwere Vorhänge, die zugezogen ein Zimmer vollständig abdunkeln können.

### 4. Leiter & Nicht-Leiter

Stoffe, die keinen Strom leiten (z.B. Gummi oder Porzellan) und als Isolatoren Verwendung finden, reflektieren das Licht verschwommener und dumpfer als Metall-Stoffe, die Strom leiten können. Bei hoch polierten Metallen hingegen reflektiert sich die Umgebung und besonders auch Lichtquellen nahezu perfekt auf der Oberfläche. Die hellsten Punkte auf reflektierenden Oberflächen werden auch **Glanzpunkte** genannt und sind sehr wichtig für die korrekte Wirkung eines Materials, da sich oft nur im Glanzpunkt die Feinstruktur der Oberfläche erkennen lässt (da wir die virtuellen Materialien nicht anfassen können, ist die Oberflächenbeschaffenheit oft nur durch den Glanzpunkt wahrnehmbar).

Die hier genannten Charakteristika stellen nur einen kleiner Ausschnitt dar mit denen die vielen Stoffe beschrieben werden könnten, die es auf unserem Planeten gibt. Für die virtuelle Materialerstellung bilden sie jedoch schon eine solide Basis (siehe Abb. 16 Kapitel 4.2).

## 2. Blender

### 2.1. Der erste Start

Beim ersten Start von Blender werden im “Splash-Screen” ein paar Einstellungen abgefragt wie die Sprache, die Maustaste zum Selektieren (Standard: links), was die Leertaste machen soll (Standard: die Animation abspielen) und welches Farb-Theme für die Oberfläche genutzt werden soll (Standard: dark). Alle diese Einstellungen können jederzeit in (Editieren | Einstellungen , engl. Edit | Preferences) verändert werden.

### 2.2. Grobe Orientierung

Nach dem Schließen des Splash-Screens sieht man folgende Aufbau (siehe Abb. 4). Neben einer oberen Menüleiste (1) befindet sich im Hauptteil des Fensters die 3D-Ansicht (2). Rechts-oben befindet sich eine Übersicht der Szenen-Objekte (3), rechts-unten ein Dialog mit vielen Einstellungen (4) und unten eine Timeline (deutsch Zeitleiste, 5).

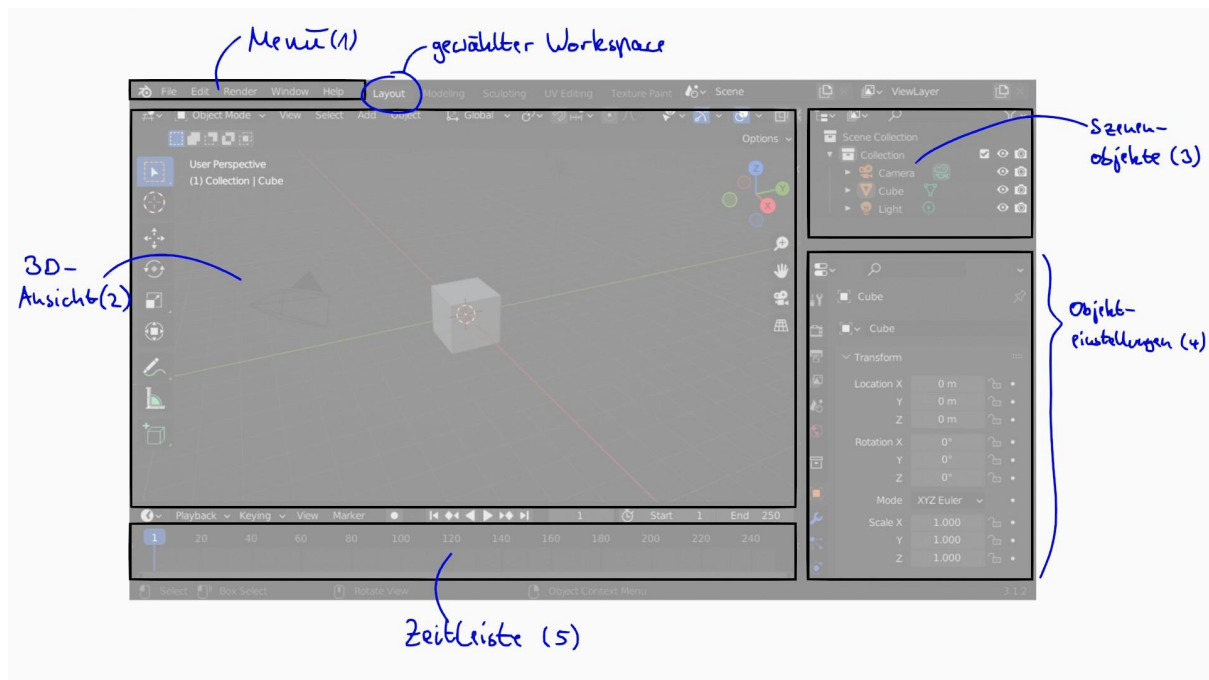


Abb. 4. Die Oberfläche von Blender im Workspace “Layout”.

### 2.3. Einstellungen

Im Edit-Menü (erreichbar über das Menü, siehe Abb. 4 Menü (1)) können die Programm-Einstellungen verändert werden. Der wohl häufigste Grund in dieses Menü zu gehen ist das Einschalten bzw. Installieren von Plugins im Menü **Add-Ons** mit denen man Blender für spezielle Arbeitsziele anpassen kann.

Außerdem sehr häufig: Über das Menü “System” muss die zu verwendende GPU explizit für den Cycles-Renderer freigeschaltet werden, sonst wird Cycles im CPU-Modus betrieben (siehe Kapitel 5.3).

User, die mit Touchpad arbeiten oder keine Maus mit mittlerer Maustaste besitzen, finden im Input-Menü auch eine Checkbox zur Emulierung des mittleren Mouse-Buttons (funktioniert dann mit ALT+Linke Maustaste).

## 2.4. Workspaces

Blender ist eine sehr vielseitige Applikation und bietet für viele 3D-Berufe spezialisierte Funktionen an, die in so genannten **Workspaces**, auch Window-Templates genannt, gruppiert sind. Die am oberen Rand befindlichen Workspaces enthalten z.B. auch einen Workspace zu Modeling und Shading in denen spezifische Funktionen für die Objektformung oder die Materialanlage (Packpapier-Gesaltung) zur Verfügung stehen.



Abb. 5: Screenshot der oberen Workspace-Zeile. Per Default ist der Layout-Workspace aktiviert, welcher genutzt wird, um eine 3D-Szene einzurichten (engl. to layout)

## 2.5. Fenster-Typen

Die Fenster von Blender sind die eigentlichen 3D-Arbeitsorte. Jedem Fenster ist immer ein bestimmter Fenster-Typ zugeordnet. Mit dem Typ-Schalter kann man den aktuellen Fenster-Typ jederzeit verändern (siehe Abb. 6). Damit bräuchte man theoretisch die Workspaces nicht, da man durch Umschalten der Fenster-Typen stets zwischen Fenster-Typen für Modellierung, Materialerstellung usw. wechseln kann. Da ein Workspace aber mehrere Fenster enthält die alle für einen konkreten 3D-Job gemeinsam gebraucht werden, ist das Verwenden der Workspaces auf die Dauer sehr viel angenehmer (das Fenster-Typ Umschalten entfällt dann).

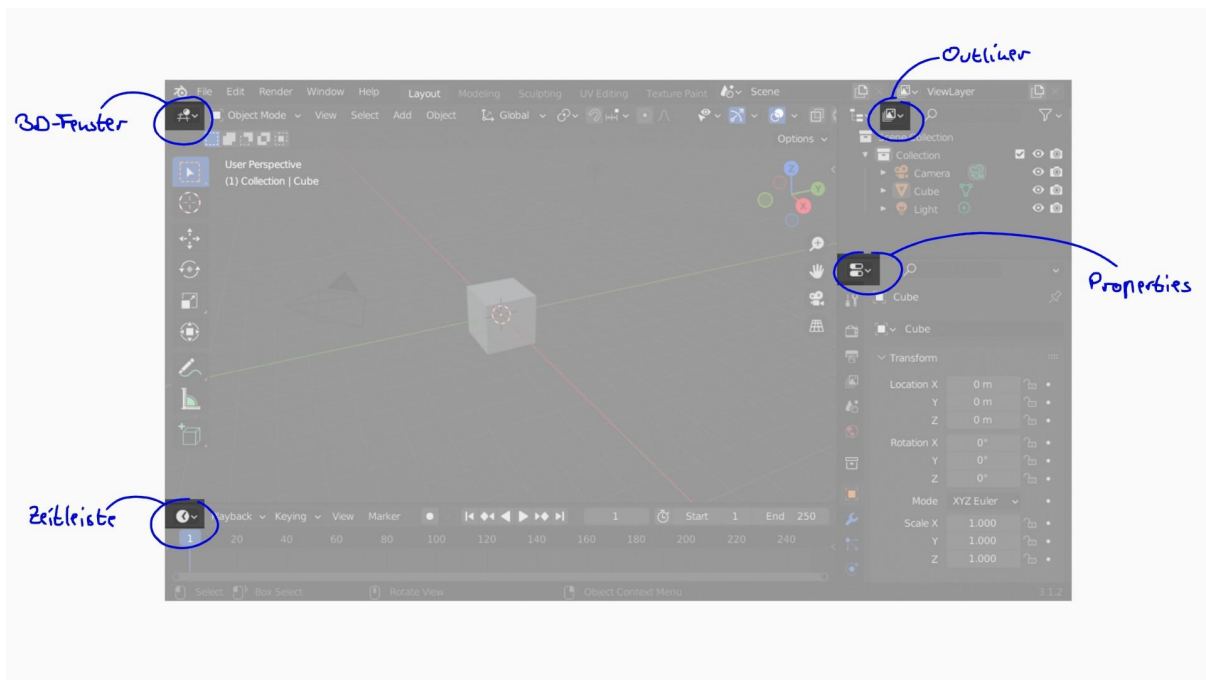








Abb. 6: Mit den Fenster-Typ-Schaltern kann jedes Fenster in einen anderen Typ umgeschaltet werden (die Workspaces enthalten Fenster mit aufgabenbezogenen, voreingestellten Fenster-Typen).

Folgende Fenster-Typen sind besonders relevant:

Tabelle 1: wichtige Fenstertypen

Icon	Typ	3D-Job	Beschreibung
	3D-View	Modelling, Vorschau	Das Arbeitspferd von Blender. Objekte und Meshes können hier bearbeitet werden. Viele Workspaces nutzen den 3D-View auch als Vorschau.
	Properties	alle	Dialoge mit Einstellungen zum aktivierten Objekt
	Outliner	Layout	Übersicht aller Objekte in der 3D-Szene
	Shader Editor	Materialien erstellen	Lichtreaktion der Materialien einstellen und Texturen zuordnen, PBR-Shader erstellen
	Image Editor	Materialien erstellen	Bilder betrachten zur Nutzung als Textur im Shader
	UV Editor	Materialien erstellen	Die Verbindung zwischen Mesh-Drahtgitter und Packpapier-Überzug definieren



## 2.6. Gekachelte Oberfläche

Die Fenster lassen sich in Blender beliebig oft unterteilen und auch umgekehrt wieder vereinigen, um eine für die 3D-Arbeit des individuellen Users optimale Anordnung von Fenstern zu ermöglichen (oft muss man mehrere Fenster nebeneinander legen, um den Bearbeitungseffekt zu sehen). Jede Fenster-Unterteilung (engl. Split) erzeugt ein weiteres Fenster in den Grenzen des alten Fensters. Diese Art der Fenster-Anordnung / Unterteilung ist ziemlich einzigartig und deshalb für viele User erstmal ungewöhnlich und verwirrend.

Durch ein Rechtsklick auf eine Fenster-Grenze erscheint die Möglichkeit das Fenster zu teilen (siehe Abb. 7). Wird Split gewählt kann mit der Maus das neue Fenster auf die benötigte Größe aufgezoogen werden.

Die Option Fenster zusammenzufügen ist nur an einer gemeinsamen Grenze möglich, an der zuvor ein altes Fenster geteilt wurde. Wird Join gewählt kann man mit der Maus entweder in das eine oder andere Fenster hineinziehen, welche dann durch das andere Fenster vollständig übernommen wird.

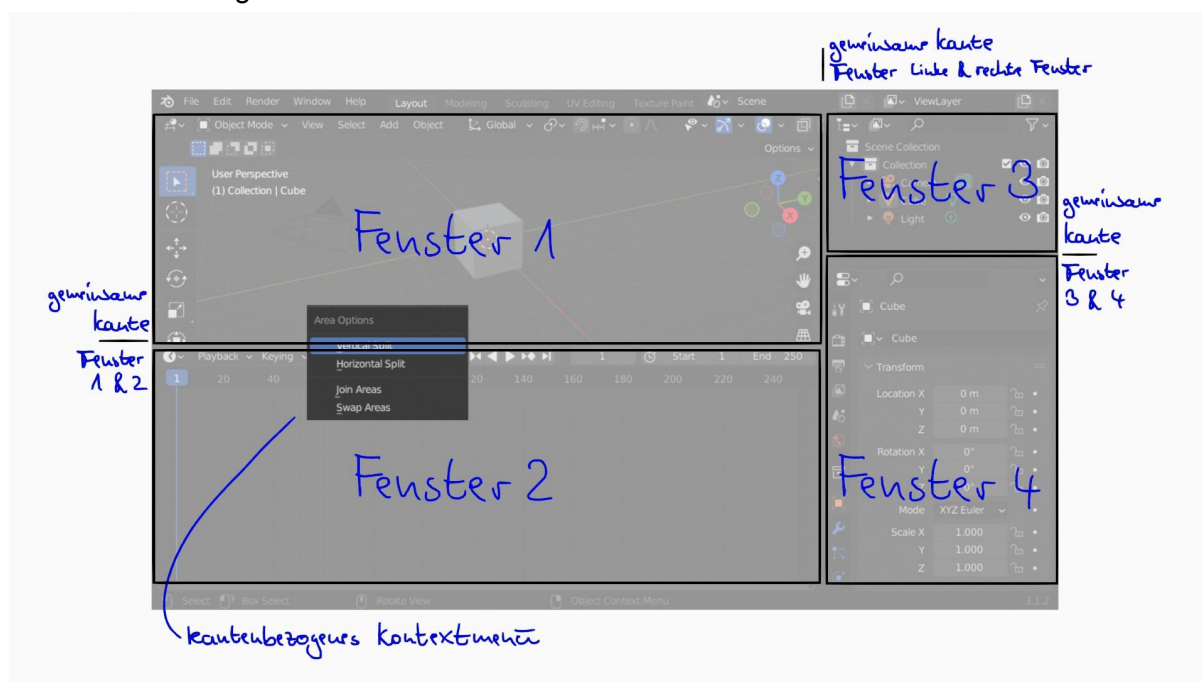


Abb. 7: Nach Rechtsklick einer gemeinsamen Kante erscheint ein Kontextmenü, mit dem Fenster geteilt oder vereint werden.

## 2.7. 3D-View

Die 3D-View ist der wichtigste Fenster-Typ und wird für viele 3D-Jobs benötigt. Nach dem Start sieht man im 3D-View folgende Inhalte (siehe Abb. 8): Ein Koordinaten-System (1) mit der X- und Y-Achse (die nach oben gerichtete Z-Achse kann bei Bedarf auch eingeschaltet werden). Am Treffpunkt aller Achsen befindet sich der globale Nullpunkt (2). Rechts befinden sich Schnellwahlschalter zur Änderung der Arbeitsansicht (3). Oben in der 3D-View befinden sich Einstellungen um die Darstellungsart und Detailfülle anzupassen (4). Links befinden sich Tools, um 3D-Objekte in der Szene zu erschaffen und zu verändern (5).

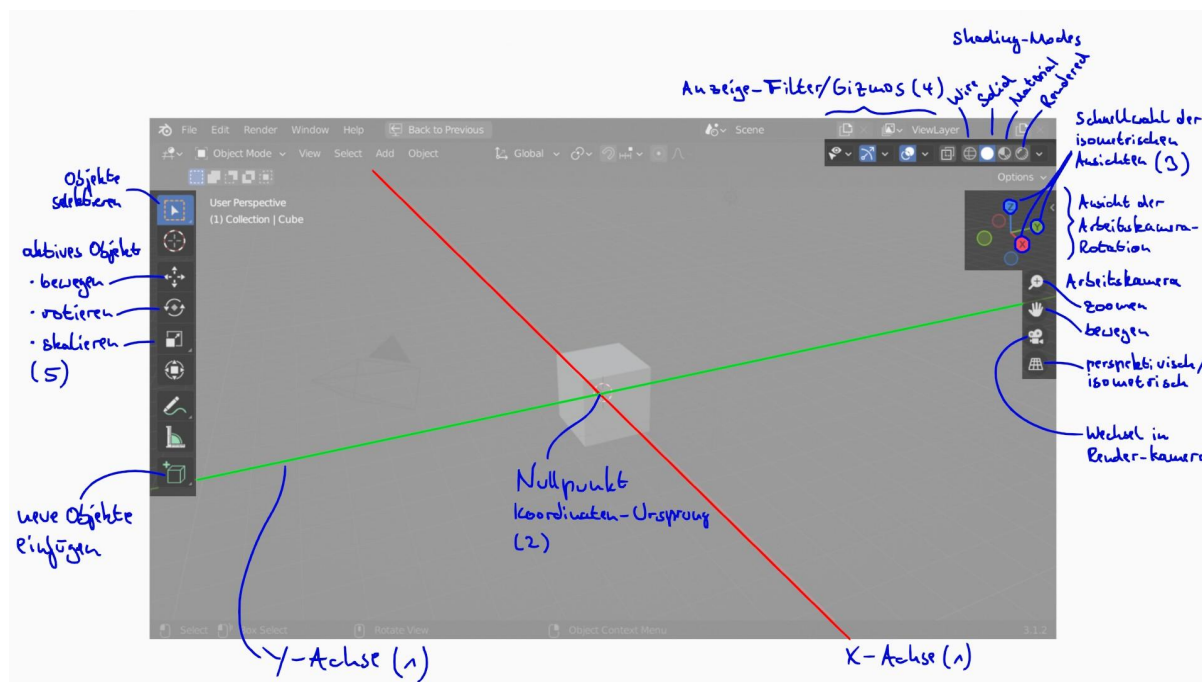


Abb. 8: der Fenster-Typ 3D-View

Die 3D-View ermöglicht primär in die 3D-Szene hineinzuschauen mit der Viewport-Kamera (auch Arbeitskamera genannt). Mit der Maus kann diese Viewport-Kamera von Richtung und Position verändert werden:

Maustaste / Keyboard-Taste	Wirkung
Maus bewegen bei gedrückter mittlerer Maustaste	Die Sicht rotieren
Maus bewegen bei gedrückter Shift-Taste	Die Sicht seitlich bewegen (engl. strafen)
Das Mousrad (mittlere Taste) scrollen	In die Szenen hinein- oder hinauszoomen

Ganz schnell verliert man mal beim Steuern der Viewport-Kamera die Orientierung und sieht die Objekte nicht mehr. Mit der **Home-Taste** des Keyboards (der Mauszeiger muss sich innerhalb des 3D-View Fensters befinden) wird die Sicht wieder so eingestellt, dass sich alle Objekte der 3D-Szene im Sichtbereich der Viewport-Kamera befinden.

Obwohl Blender neu installiert / gestartet wurde, befinden sich immer schon 3 Objekte in der 3D-Szene, die man nicht selbst eingefügt hat (siehe Abb. 9). Dies ist ein 2x2 Meter großer Default-Cube (1) und eine Lichtquelle (2). Obwohl man mit der Viewport-Kamera in die 3D-Szene schauen kann, hat diese keinen Effekt auf den Bildausschnitt des letztlich gerenderten Bildes. Hierfür ist ausschließlich die Platzierung und Ausrichtung der Render-Kamera (3) verantwortlich, die sich auch als 3D-Objekt bewegen und rotieren lässt.

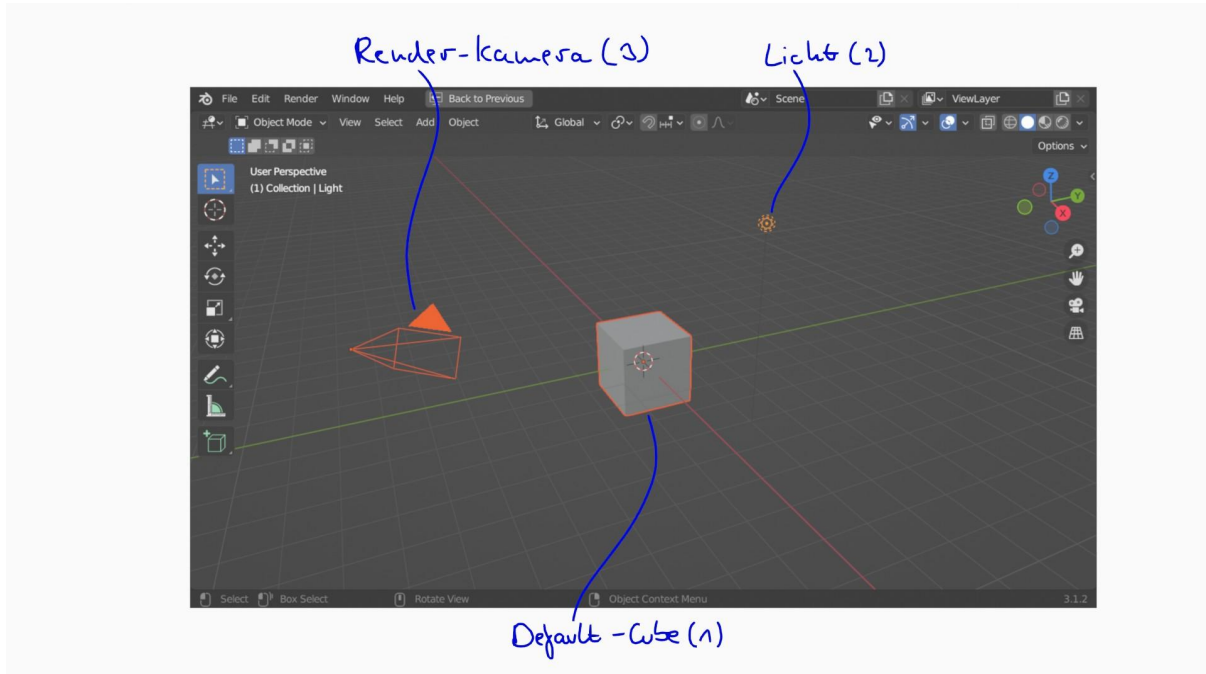


Abb.9: Initiale Objekte im 3D-View

### 3. Formen erschaffen und verändern

#### 3.1. Primitive Objekte einfügen

Im 3D-View-Fenster können mit dem Add-Button (Abb. 8 links unten) neue Primitiv-Shapes (Kugel, Kubus, Kegel, Donut) in die 3D-Szene gezogen werden. Hierbei wird mit dem Start des Click-Mouse-Ziehens mitgeteilt, wo das Objekt hin soll (2), die Länge des gedrückt-Ziehens bestimmt, wie groß die Grundfläche sein soll (3) und nach Loslassen der Maustaste wird letztendlich noch bestimmt, wie hoch das Objekt werden soll (4). Hierbei können Objekte entweder auf den X-Y-Boden gesetzt, oder auf schon bestehende Objekte draufgesetzt werden. Bei letzterer Variante wird das neue Objekte eventuell so gedreht, dass es flach ("plan") mit dem schon bestehende Objekte verklebt wird (sie werden nicht wirklich verklebt und können jederzeit wieder auseinander genommen werden).

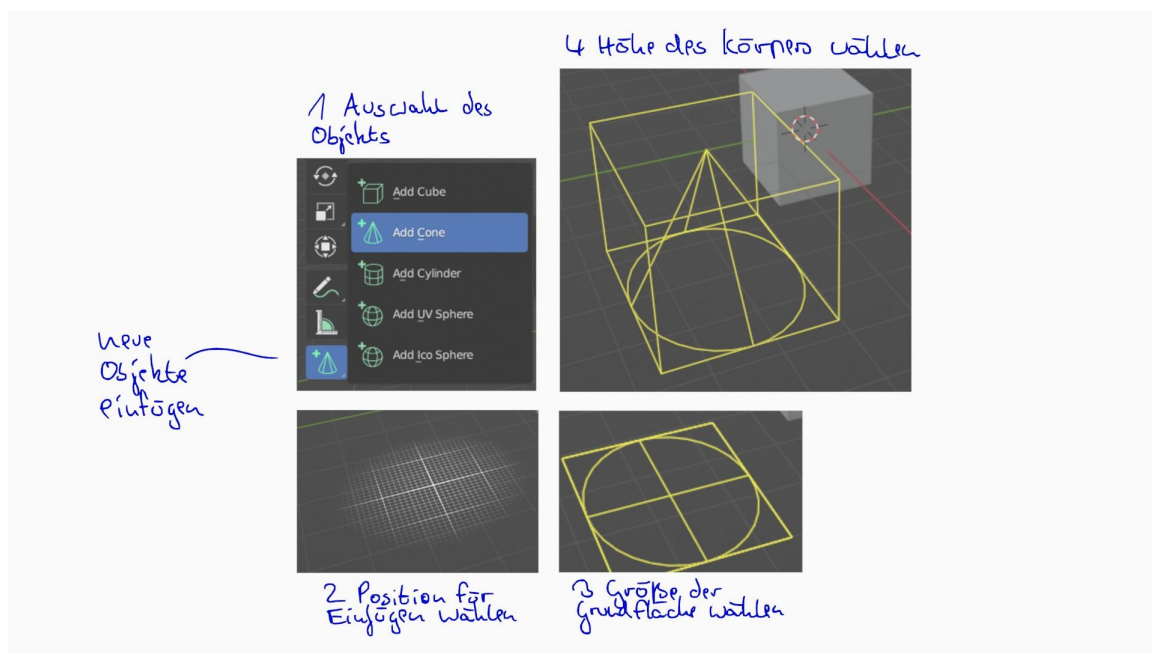


Abb. 10: ein neues Primitiv-Shape der 3D-Szene hinzufügen

## 3.2. Objekteigenschaften ändern

Die Platzierung mit der Maus geht relativ schnell, ist naturgemäß aber ungenau. Mit einem Blick in das Eigenschafts-Fenster links (engl. Property, siehe Abb. 11) können die Objekteigenschaften als numerische Zahlen abgelesen werden. Ein Objekt wird in der 3D-Szene beschrieben durch die drei **Transformations**-Eigenschaften a) die Position (engl. Location), b) die Rotation und c) die Skalierung. Pro Objekteigenschaft sind alle 3 Dimensionen x,y,z einstellbar und mit einem Klick in das jeweilige Feld editierbar. Wird das Editieren mit der Eingabetaste (engl. Enter) abgeschlossen, wird die neue Position / Rotation / Skalierung sofort in der 3D-View sichtbar (insofern das Objekt von der Viewport-Kamera immer noch eingefangen wird).

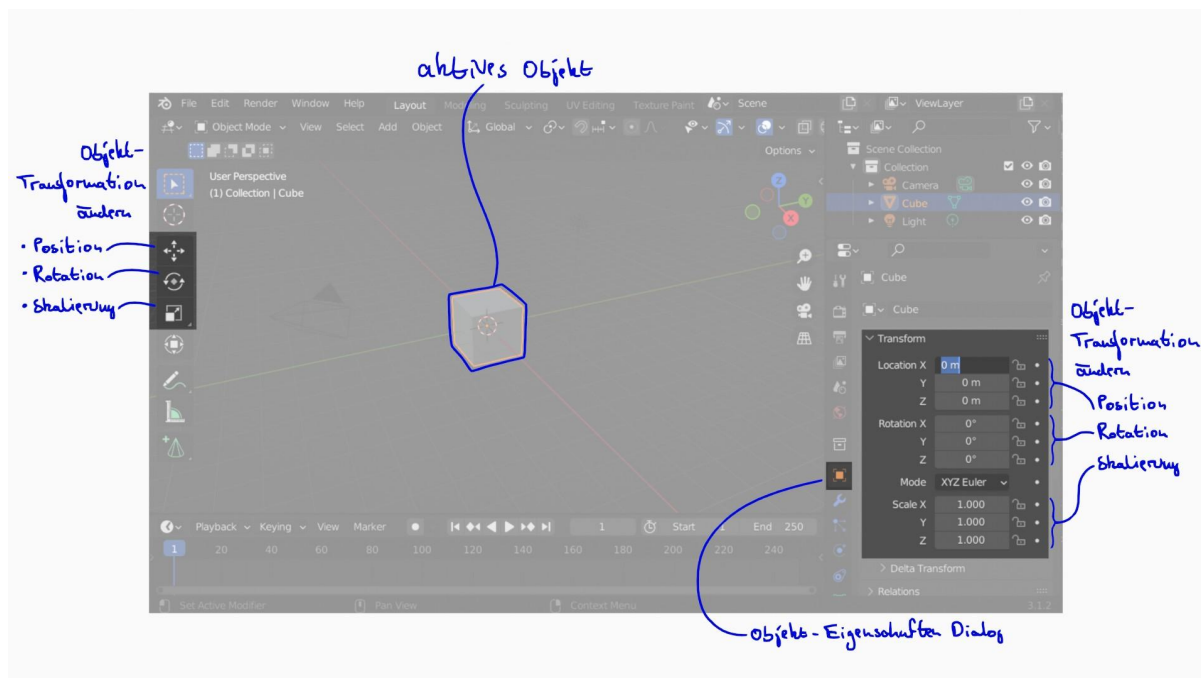


Abb. 11: die Objekteigenschaften in der genauen Beschreibung

Jedes Objekt besitzt seine eigenen 3 Transformations-Eigenschaften, selbst das Licht und die Render-Kamera.



### 3.3. Der Röntgen-Blick

Befinden sich in der 3D-Szene sehr viele Objekte, kann es sein, dass man in der Viewport-Kamera nicht mehr alle genau sehen kann (siehe Abb. 12). Spätestens dann kann der Röntgen-Blick weiterhelfen, der als Wireframe-Shading im Viewport-Shading-Menü eingestellt werden kann. Dieser macht die Packpapier-Überzüge der Drahtgitter-Modelle durchsichtig, so dass man auch durch andere Objekte verdeckte Körper sehen kann.

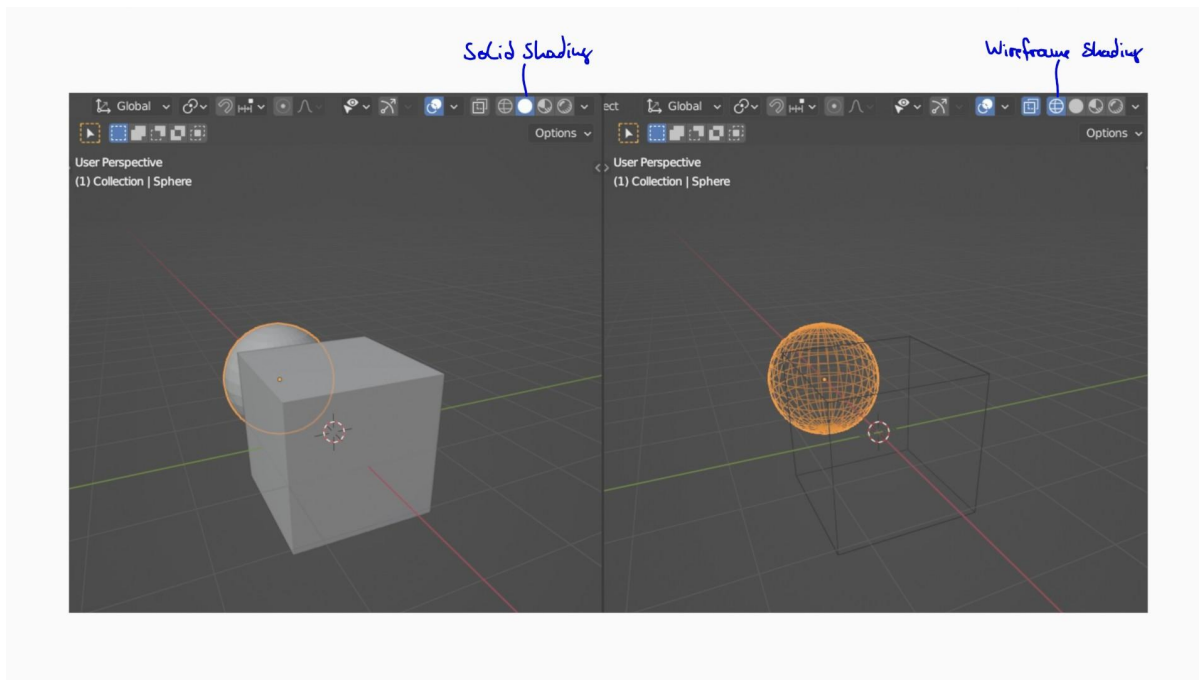


Abb. 12: Der Kubus verdeckt im Solid-Mode die Kugel, während sie im Wireframe-Mode sichtbar ist

Auf das vorherige Solid-Shading (die Packpapiere sind nicht durchsichtig) kann jederzeit wieder zurückgestellt werden.

### 3.4. Edit-Mode

Der Standard-Bearbeitungsmodus von Objekten heisst **Objekt-Mode** und ermöglicht das Verändern der Objekteigenschaften wie Position, Rotation und Skalierung. Da 3D-Objekte aus Knoten, Kanten und Flächen bestehen, muss man diese auch ändern können. Dafür ist der **Edit-Mode** gedacht (siehe Abb. 13). Durch die Auswahl welche der so genannten geometrischen Primitive (nicht zu verwechseln mit Primitiv-Shapes) editiert werden sollen, wird der Select-Mode genutzt. Im Knoten-Select-Modus (Vertex-Select-Mode) können nun einzelne Knoten selektiert und genau so wie bei Objekten im Objekt-Modus verschoben werden. Kanten und Flächen können rotiert und skaliert werden.

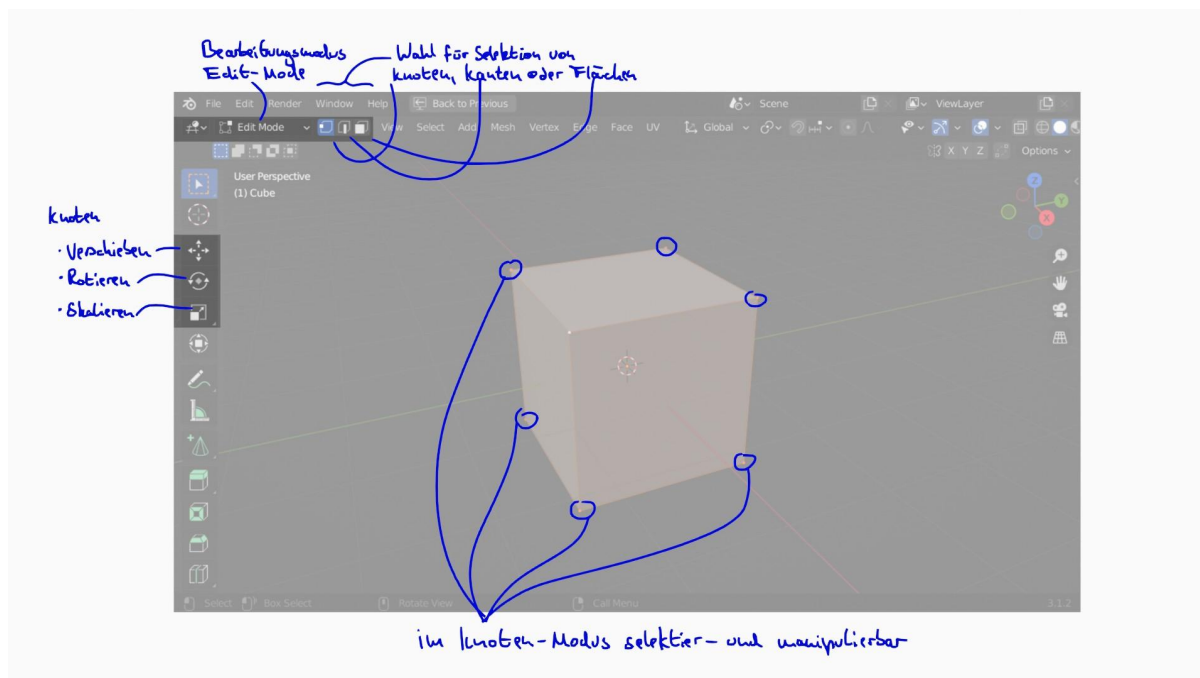


Abb. 13: Mit dem Edit-Mode können Knoten, Kanten und Flächen eines aktiven Objekts bearbeitet werden.

## 4. Packpapier einfärben

### 4.1. Materialien

Materialien sind neben der Form, die ein Objekt hat, am wichtigsten für die Wieder-Erkennung bzw. Wahrnehmung des Objekts. Den Material-Slots eines Objekts (siehe Abb. 14) kann ein schon im Blender-File existierendes Material zugewiesen oder ein neues Material erstellt werden. Die Wiederbenutzung von einmal erstellten Materialien ist sehr sinnvoll, da dann alle z.B. Rostig-Eisen Oberflächen gleich rostig und damit konsistent wirken.

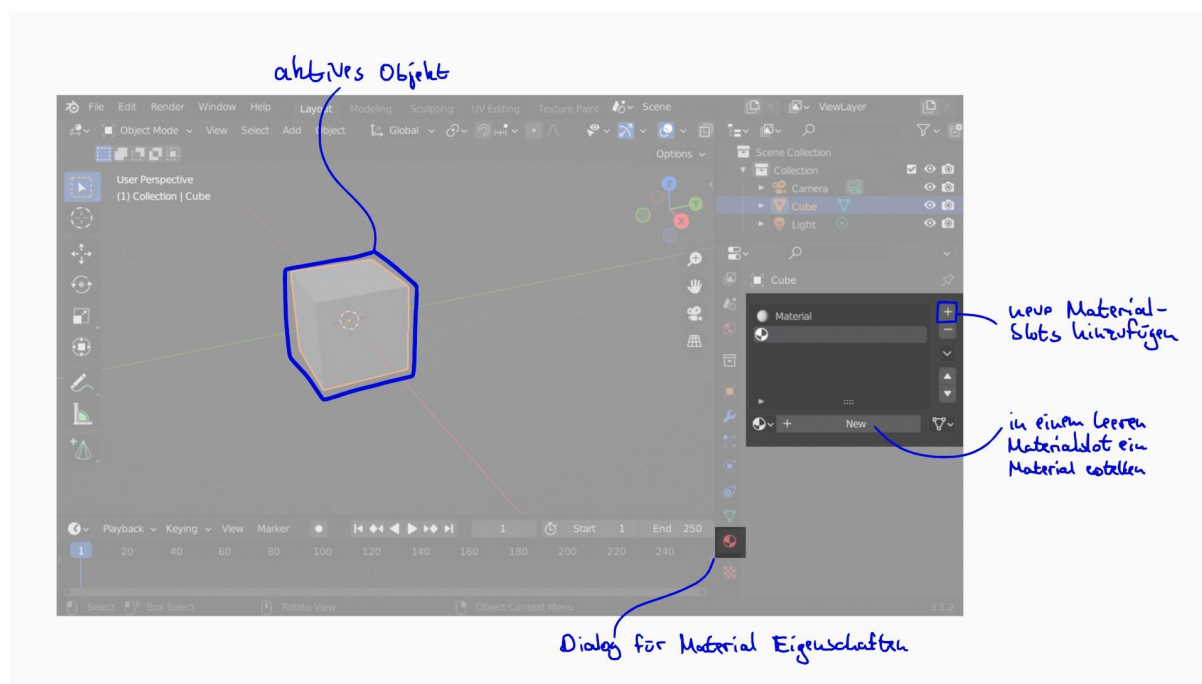


Abb. 14: Jedes Objekt kann einen oder mehrere Material-Slots mit Materialien haben.

Ist auf dem Objekt nur ein Material-Slot mit Material vorhanden, wird dieses Material für alle Flächen verwendet. Man kann aber auch mehrere Slots mit unterschiedlichen Materialien anlegen und diese im Edit-Mode spezifischen Flächen zuweisen (siehe Abb. 15).

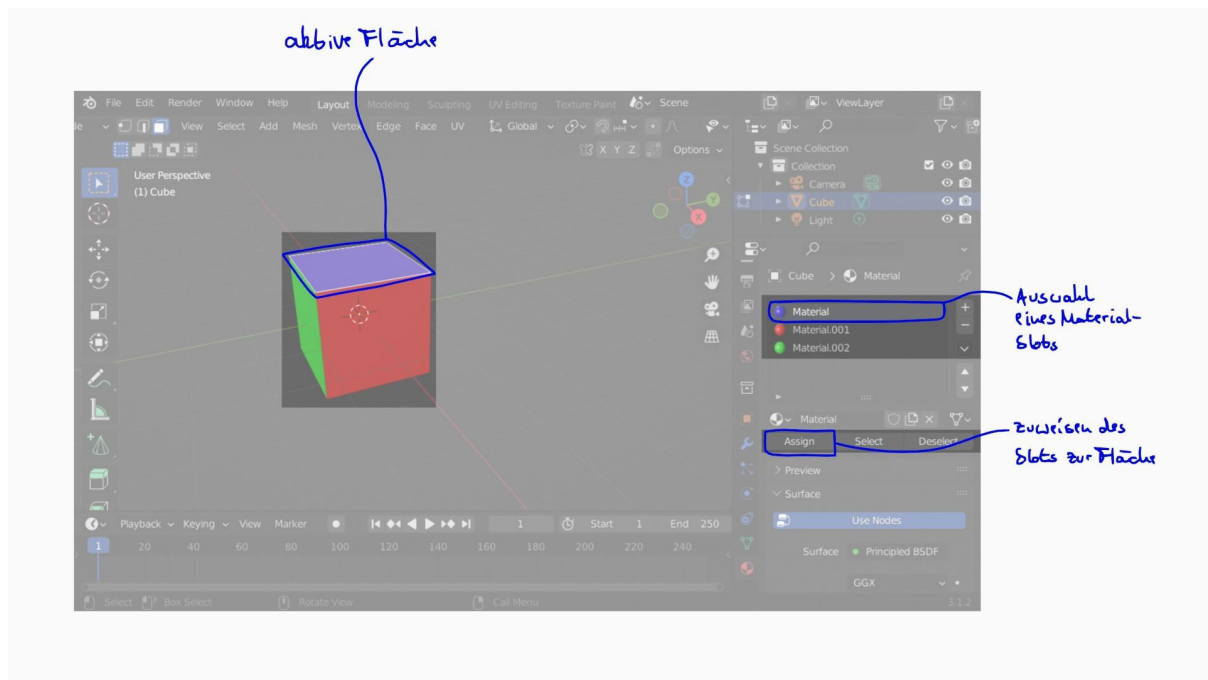


Abb. 15: Die Zuweisung eines Materialslots zu einer bestimmten Fläche.

Die Eigenschaften der Materialien werden im Shader-Editor definiert. Da es sehr viele Realwelt-Materialien gibt (siehe Kapitel Kleine Materialkunde) und Materialien sehr unterschiedlich auf das Licht reagieren, benötigt man für den Nachbau in Blender ein sehr leistungsfähiges System, was die Flexibilität einer Programmiersprache hat. Hierfür werden Shader-Nodes genutzt, die eine Art graphische Programmiersprache repräsentieren.

## 4.2. Shader & Nodes

Beim Einstellen der Eigenschaften eines Materials setzt Blender auf das "Physically-Based-Rending" (PBR). Beim PBR werden Materialeigenschaften anhand von physikalischen Messgrößen aus Schichten zusammengesetzt bzw. die Shader sind schichtweise aufgebaut, um das Verhalten von Echt-Materialien möglichst gut nachzuahmen. Das PBR-Konzept wurde gemäß einer Definition von Disney umgesetzt, nachdem sich ein Shader aus schon erwähnten Schichten verschiedener Material-Effekte (Grundfarbe, Lichtdurchlässigkeit, Glanz etc.) aufbauen lässt. Die wichtigsten Schichten sind:

Effekt- Name	Deutsche Übersetzung	Wirkung
0 Base-Color (auch Albedo)	Grundfarbe	Einfärbung der Oberfläche ohne Einfluss richtungsabhängiger Lichtquellen

genannt)		
1 Metallic	Metall-Anteil	Wirkung des Oberfläche als Metall oder als Nicht-Metall; hat starke Auswirkungen auf Reflektionen
2 Specular	Glanz-Anteil	Bestimmt die Stärke der positionsunabhängigen Reflektion
3 Roughness	Rauheit	Wie Rau ist die Oberfläche; je weniger rau, desto stärker spiegelt sich die Umgebung
4 Trans- mission	Lichtdurch- lässigkeit	Wie stark Lichtdurchlässig ist die Oberfläche (das Licht wird beim Durchgang durch das Material verändert)
5 Emission	Leuchtkraft	Wie stark sendet die Oberfläche Licht aus
6 Alpha	Transparenz	Wie transparent ist die Oberfläche (das Licht wird beim Durchgang nicht verändert)

Im [Blender-Manual](#) gibt es zu den jeweiligen Einstellungen obiger Material-Effekte anschauliche Übersichten.

Der PBR-Shader, der in Blender Principled-BSDF-Shader heisst, enthält für alle oben genannten Effekte Regler oder Textur-Eingaben und wirkt zugegebenermaßen ein wenig unübersichtlich. Man hat sich aber bewusst dafür entschieden alle Effekte in einem prinzipiellen-Basis-Shader einzubauen, damit es für die Artists leichter ist Materialien zu definieren und mit anderen auszutauschen. Vor dem BSDF-Shader hatte jeder Artist seine eigenen mega-komplexen Shader-Trees (Verknüpfungen von vielen einzelnen Nodes) gebaut, welche man dann nur sehr schwer zwischen unterschiedlichen 3D-Softwares austauschen konnte. Der Austausch ist nun deutlich einfacher möglich. Egal ob Blender, Unity, Unreal, Renderman, Playcanvas etc. die meisten 3D-Tools kennen das PBR-Konzept und man kann sicher sein, dass die Shader mit gleichen Effekt-Einstellungen auch (nahezu) gleich in den verschiedenen Tools aussehen.



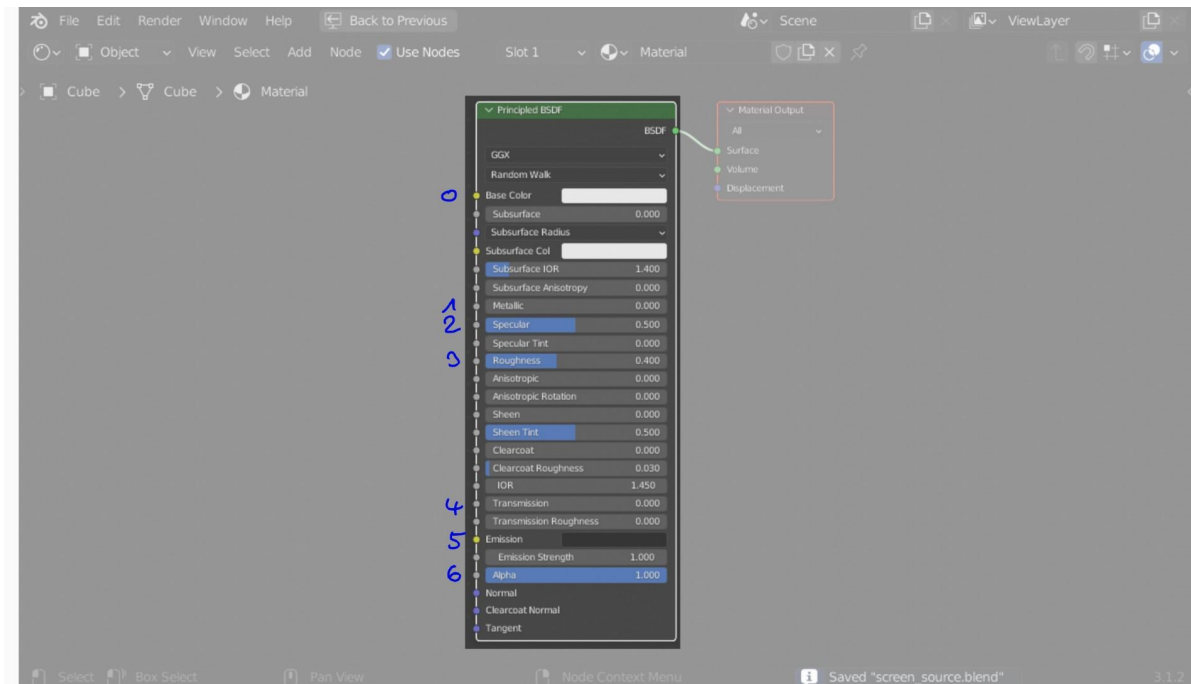
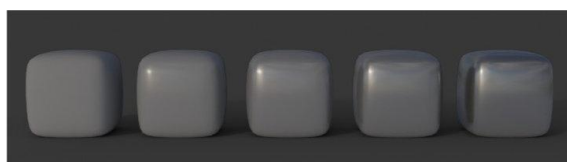


Abb. 16 Der Principled-BSDF-Shader mit den vielen Schicht-Einstellungen



Metallic (1)



Transmission (4)



Specular (2)



Emission (5)



Roughness (3)



Alpha (6)

Abb. 17 Verschiedene Einstellungen der Effekt-Schichten (geringster Wert links, höchster Wert rechts).

### 4.3. Flächen unterschiedlich einfärben

Im Principled-BSDF-Shader läßt sich für jede der unterschiedlichen Effekt-Schichten eine individuelle Farbe bzw. Helligkeit einstellen Diese gilt dann aber für alle Flächen eines Objekts, egal wo diese liegen. In vielen Fällen soll aber kein einheitlicher Farb- / Helligkeitswert verwendet werden. Nach dem Vorbild der Natur gibt es keine zwei Flächenstücke die alle exakt den gleichen Farbwert haben, kleine Unebenheiten und Staub machen jedes Flächenstück einzigartig. Möchte man daher einer Effekt-Schicht größere Variationen von Farb- / Helligkeitswerten zuweisen, kommt das unterschiedlich bemalte Packpapier zum Einsatz. Im Shading-Workspace gibt es den Fenster-Typ “Shader-Editor”. In diesen können aus einem Datei-Explorer vorhandene Texturen mit Drag`n-Drop reingezogen und dann mit den Input-Pins der Effekt-Schichten verbunden werden (siehe Abb. 18).

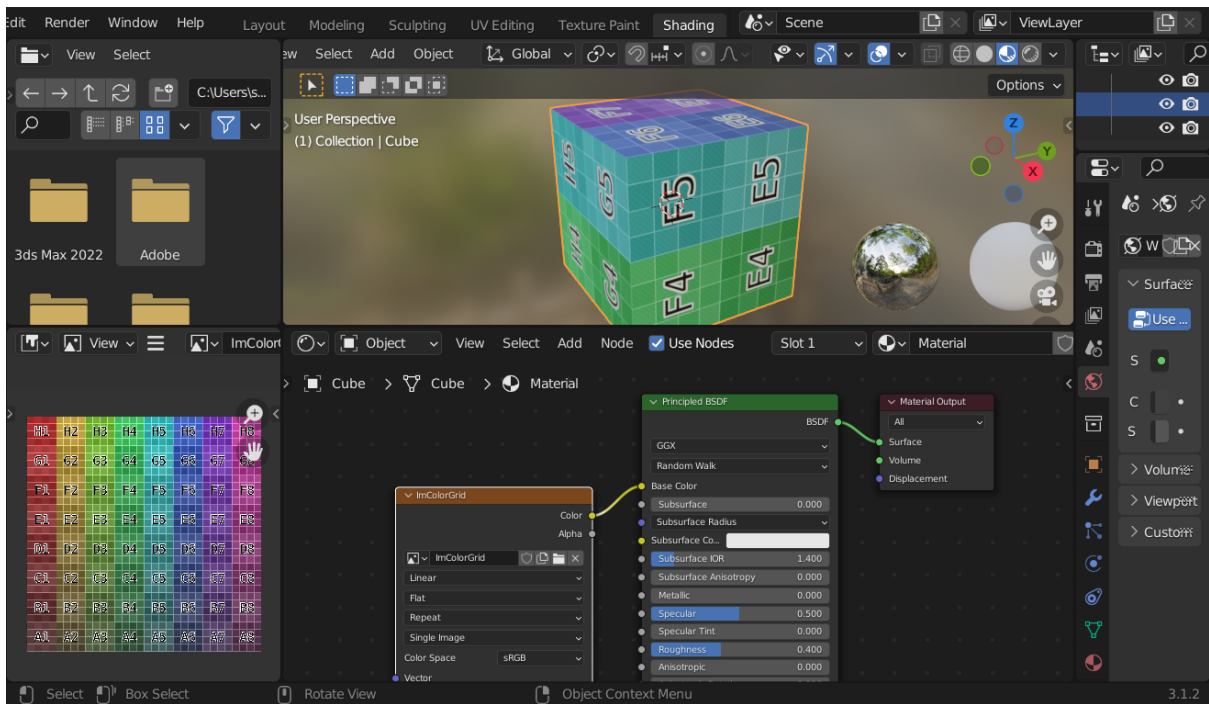


Abb. 18 Das Material des Default-Cube erhielt eine Textur für die Base-Color.

## 4.4. Wohin das Packpapier aufgeklebt wird

Beim Bekleben von Kartons mit bunten Packpapier macht man sich (wenn man es ordentlich machen will) Gedanken, an welche Stelle des Kartons welcher Teil des Packpapier geklebt wird. Meist will man verhindern, dass kleine Fehler oder Risse im Packpapier die besonders gut sichtbare Oberseite bedecken. Die Zuordnung - von in der flachen Ebene ausgerolltem - Packpapier auf die Oberfläche eines Karton-Objekts (oder jedes anderen beliebig komplex geformten Objekts) nennt man UV-Layout. Layout steht einfach für Zuordnung und "UV" steht für die horizontalen ("U") und vertikalen ("V") Koordinatenachsen des Packpapiers, wenn es flach auf dem Boden liegt. Eine bestimmte Position des flachen Packpapiers ist durch eine konkrete UV-Position beschreibbar (ähnlich wie eine Position auf der Erdoberfläche mit einem Längen- und Breitengrad beschreibbar ist). Nun kann für jeden Knoten eines Meshes (die Position des Knoten auf der Mesh-Oberfläche ist ja bereits über xyz-Koordinaten beschreibbar), genau eine UV-Position zugeordnet werden um damit eine feste Zuordnung zwischen flachen Packpapier und der Oberfläche des 3D-Objekts zu definieren (siehe Abb. 19).

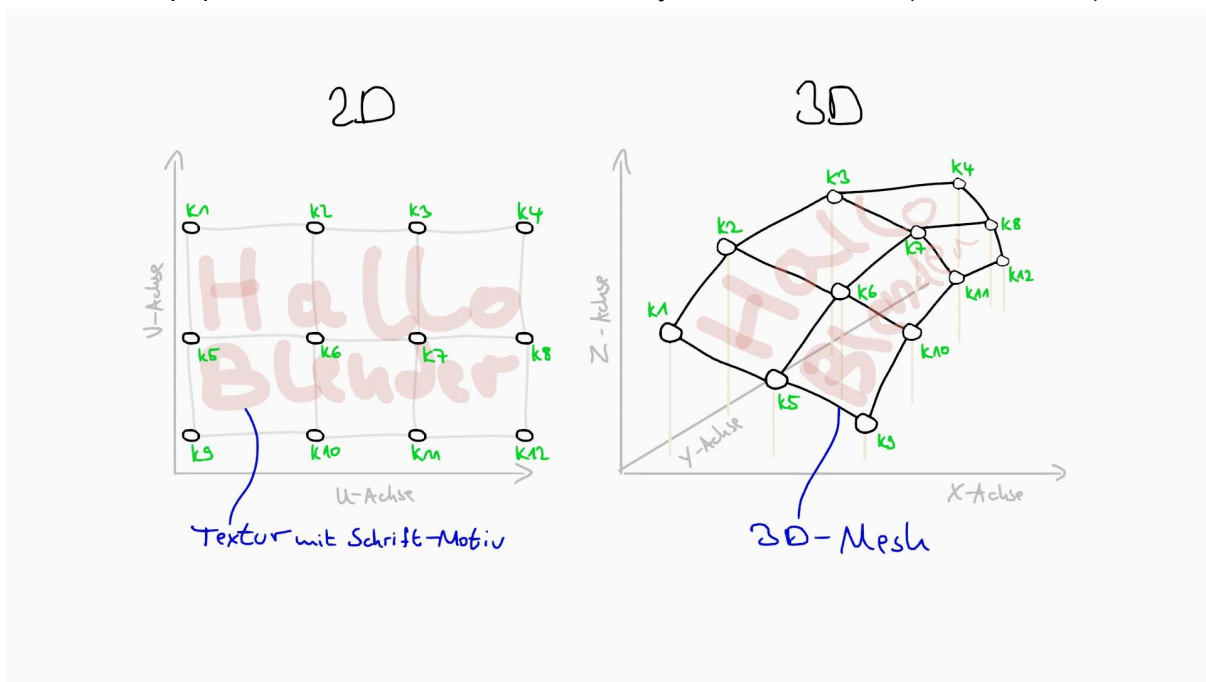


Abb. 19 Jeder Knoten (K1 bis K12) des 3D-Meshes (rechts) hat zwei weitere Koordinaten (U und V), mit denen eine Packpapier-Stelle (links, Textur mit Schrift-Motiv) dem Knoten zugeordnet wird. Damit kann für jede 3D-Fläche (bestehend aus je 4 Knoten) bestimmt werden, welcher Teil der Textur die jeweilige Fläche einfärbt.

## 4.5. Ein UV-Layout anlegen

Für jedes 3D-Objekt, welches mit dem Plus Button (siehe Abb. 10) eingefügt wurde, wird automatisch schon ein UV-Layout angelegt. Im UV-Editing Workspace kann man sich diese Zuordnung von Packpapier-Position zur Objekt-Oberfläche anschauen (beim Wechsel in den UV-Editing Workspace bei aktiviertem Objekt wechselt Blender in den Edit-Mode). Hier (siehe Abb. 20) ist links ein Netz von Knoten auf dem flachen Packpapier zu sehen (Fenster-Typ UV-Layout) und rechts das 3D-Objekt mit aufgeklebten Packpapier (Fenster-Typ 3D-View)

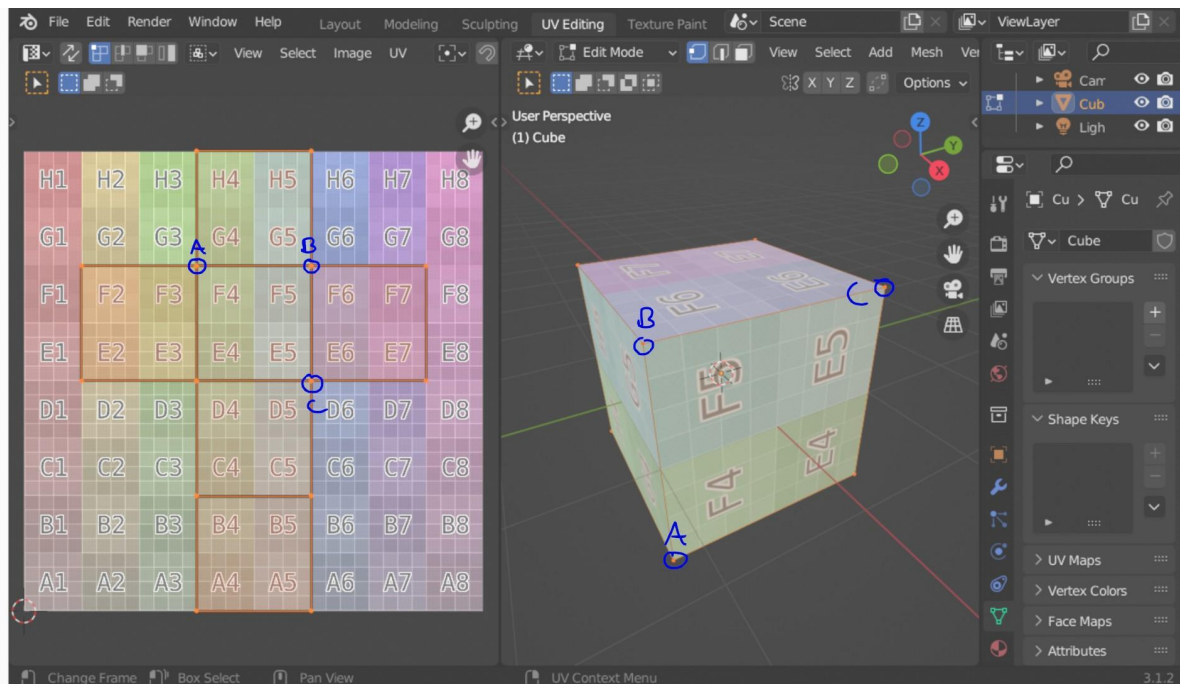


Abb. 20 der UV-Editing-Workspace für den Default-Cube. Für drei Punkte A,B,C wurde die Position auf dem Packpapier (links) und auf der Objektoberfläche (rechts) markiert

Ist bereits eine Textur in Verwendung kann die Zuordnung zwischen Packpapier und Oberfläche des 3D-Objekts sehr schön veranschaulicht werden. Hierfür muss man nur mal im 3D-View in den Shading-Mode "Material Preview" wechseln (hiermit wird das aktuelle, eine Textur verwendete Material dann sichtbar). Im Linken Fenster kann dann die Position des Netzes verschoben, rotiert oder skaliert werden und es wird sofort sichtbar, wie sich die Oberfläche im rechten Fenstern entsprechend mit verändert. Um links die Position des Netzes zu verändern muss man folgende Schritte beachten:

1. Ein Objekt durch Links-Klick aktivieren
2. In den Edit-Mode wechseln (durch Umschalten in den "UV-Editing"-Workspace geschieht das automatisch)
3. Einen, mehrere oder alle Knoten im linken UV-Layout Fenster auswählen
4. Diese beliebig Verschieben / Rotieren / Skalieren mit Hilfe der gleichen Buttons, die schon aus Abb. 11 bekannt sind

Für den Fall, dass ein 3D-Objekt selbst erstellt wurde, muss das UV-Layout nachträglich angelegt werden (außer man weiß, dass man niemals Texturen nutzen wird). Hierfür können im Edit-Mode im Fenster "3D-View" alle Flächen ausgewählt werden, für die man ein (neues) UV-Layout definieren möchte. Danach kann im Menü UV-Layout eine der

Unwrap-Optionen verwendet werden (z.B. Smart UV Unwrap). Hierbei versucht Blender das UV-Layout so anzulegen, dass auf der Oberfläche des 3D-Objekts benachbarte Flächen auch im Packpapier benachbart sind (damit ein Muster im Packpapier auch unterbrechungsfrei auf der Oberfläche des 3D-Objekts erscheint).

# 5. Licht & Ausgabe

## 5.1. Beleuchtung

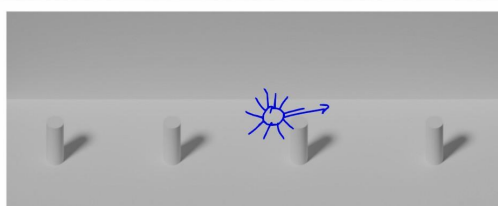
Ohne Lichtquelle gibt es kein Bild. Eine noch so brillant eingerichtete Szene, welche völlig im Dunkeln liegt, kann niemand sehen. Lichtquellen sind deshalb zwingend notwendig. In einem künstlerischen Kontext kann man Lichtquellen als kreatives Ausdrucksmittel verstehen, da der/die BeleuchterIn (die Person, die das Licht einer Szene einrichtet) die Aufmerksamkeit des Betrachters lenken und die Szene auch in eine bestimmte Mood (dt. Gefühlslage) versetzen kann durch Einstellen der Helligkeit und Lichtfärbung.

Blender besitzt Lichter verschiedener Arten:

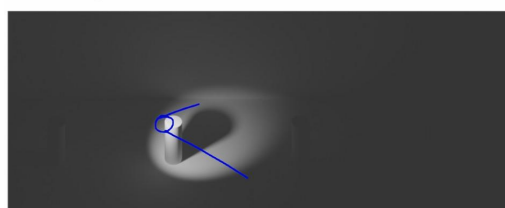
Name	Eigenschaft	Verwendung
Punktlicht (Point Light)	Sendet Licht von einer Position nach allen Richtungen aus	Standard-Licht zur Verwendung als normale Glühlampe
Sonne (Sun)	Positionsloses Licht mit einer Lichtrichtung	Erzeugt parallelen Schattenwurf wie bei der Echt-Sonne
Spotlicht (Spot Light)	Wie Punktlicht, aber sendet Licht von einer Position innerhalb eines Kegels aus	Örtlich begrenzte Beleuchtung mit einstellbarem Öffnungswinkel für lokale Aufhellung mit sanften Übergängen
Flächenlicht (Area Light)	Licht wird über ein großes Areal senkrecht zu einer Fläche ausgesendet	Erzeugt sehr sanfte Schatten, entspricht der Lichtaussendung von Bildschirmen



Point-Light



Sun-Light



Spot-Light



Area-Light

Abb. 21 Lichtwirkung der unterschiedlichen Lampen-Typen

## 5.2. Render-Kamera

Die Render-Kamera kann genauso wie alle anderen 3D-Objekte in der Szene platziert und rotiert werden. Der Teil der Welt, welcher die Kamera einfängt und sich innerhalb des Pyramidenstumpfes befindet (siehe Abb. 3) wird beim Renderprozess berücksichtigt. In einer Szene kann es beliebig viele Kameras geben, aber nur eine kann die aktuelle Render-Kamera der Szene sein (siehe Abb. 22).

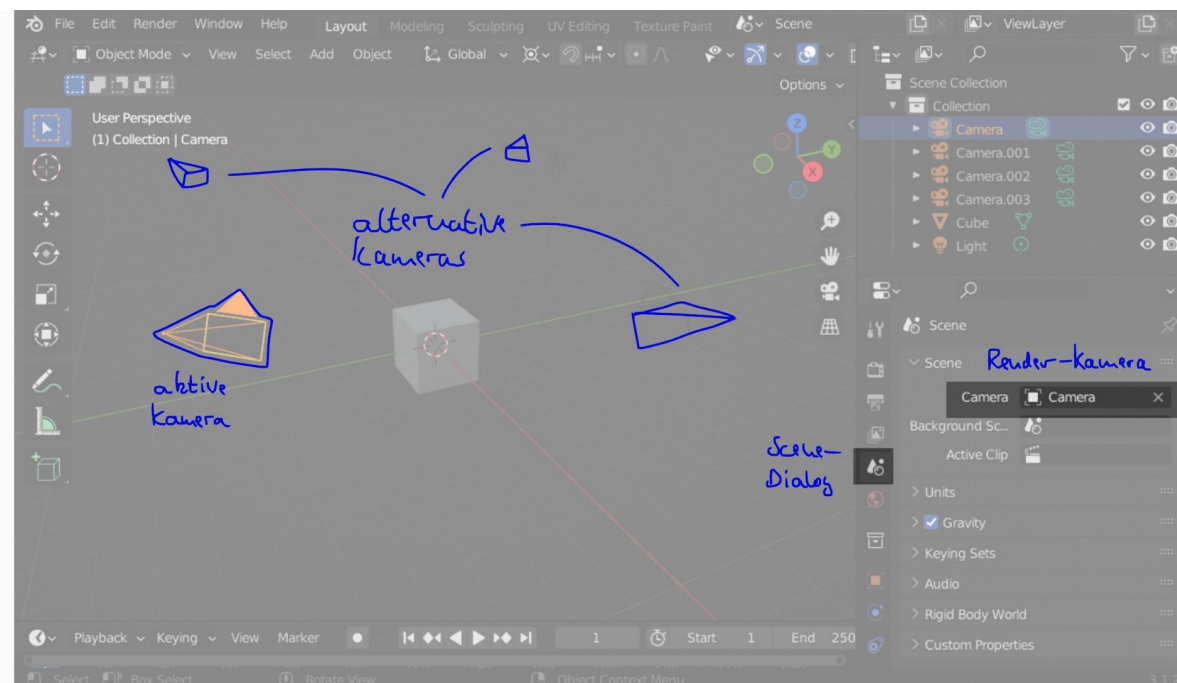


Abb. 22 Im Scene-Dialog wird eine Kamera als Render-Kamera eingestellt.

## 5.3. Rendern mit Cycles

**Cycles** ist ein Renderer, welcher einzelne Lichtstrahlen (engl. Rays) simuliert und diese auf ihrem Lauf von der Kamera bis zur Lichtquelle verfolgt (siehe Abb. 1). Hierbei können die Strahlen auf Objektoberflächen treffen, dort gebrochen, reflektiert und absorbiert werden (was genau passiert, wird über die verwendeten Material-Shader definiert). Cycles besitzt durch diese Strahlennachverfolgung (auch genannt **Raytracing**) eine hohe optische Qualität. Es ist im Vergleich zum anderen Renderer **Eevee** langsamer, ist dafür aber wesentlich unkomplizierter in der Bedienung (man benötigt weniger bis kein technisches Hintergrundwissen, um eine hohe Bildqualität zu erreichen).

Bevor Cycles als Renderer gewählt wird, muss die Grafikkarte für Cycles im Blender Preferences | System - Menü explizit freigeschaltet werden (siehe Abb. 23), da sonst standardmäßig die CPU zum Rendern genutzt wird.



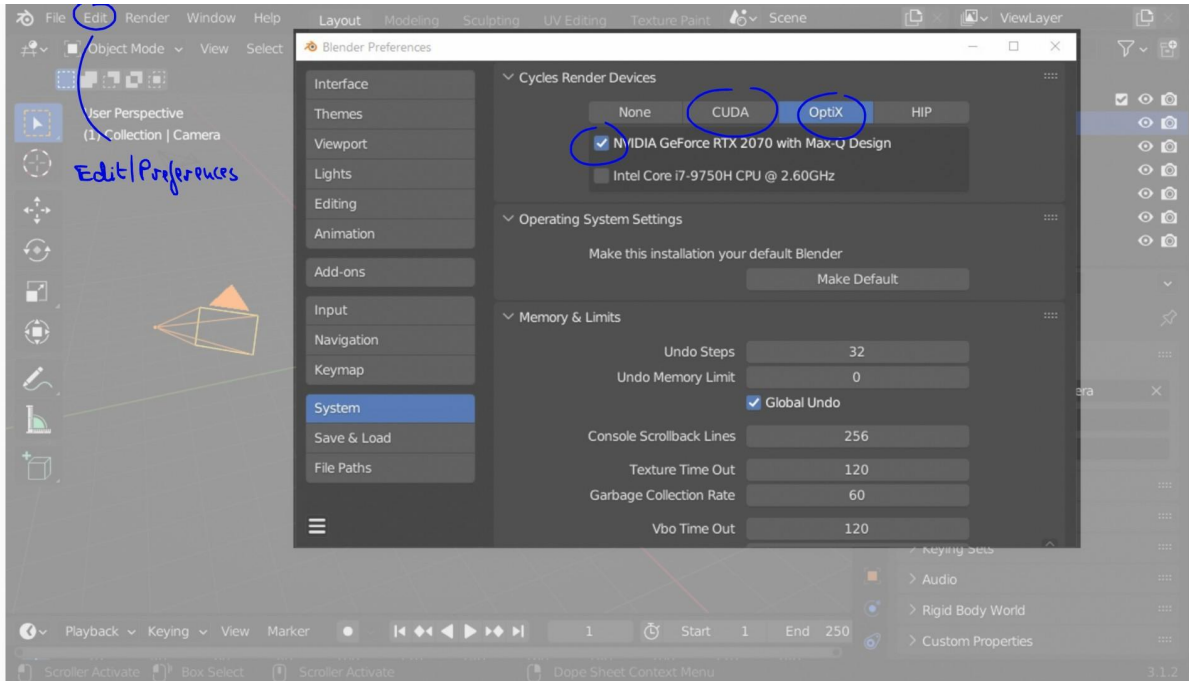


Abb. 23 Anschalten der GPU (Mode CUDA oder bei neueren Grafikkarten auch OptiX) zur Verwendung in Cycles.

Danach kann Cycles als zu verwendender Renderer (auch hier nochmal umstellen auf GPU) explizit im Render-Properties-Dialog ausgewählt werden (siehe Abb. 24). Grundsätzlich kann Cycles auch rein mit der CPU betrieben werden, allerdings ist der Performance-Booster bei modernen Grafikkarten (im Vergleich zur CPU) deutlich feststellbar.

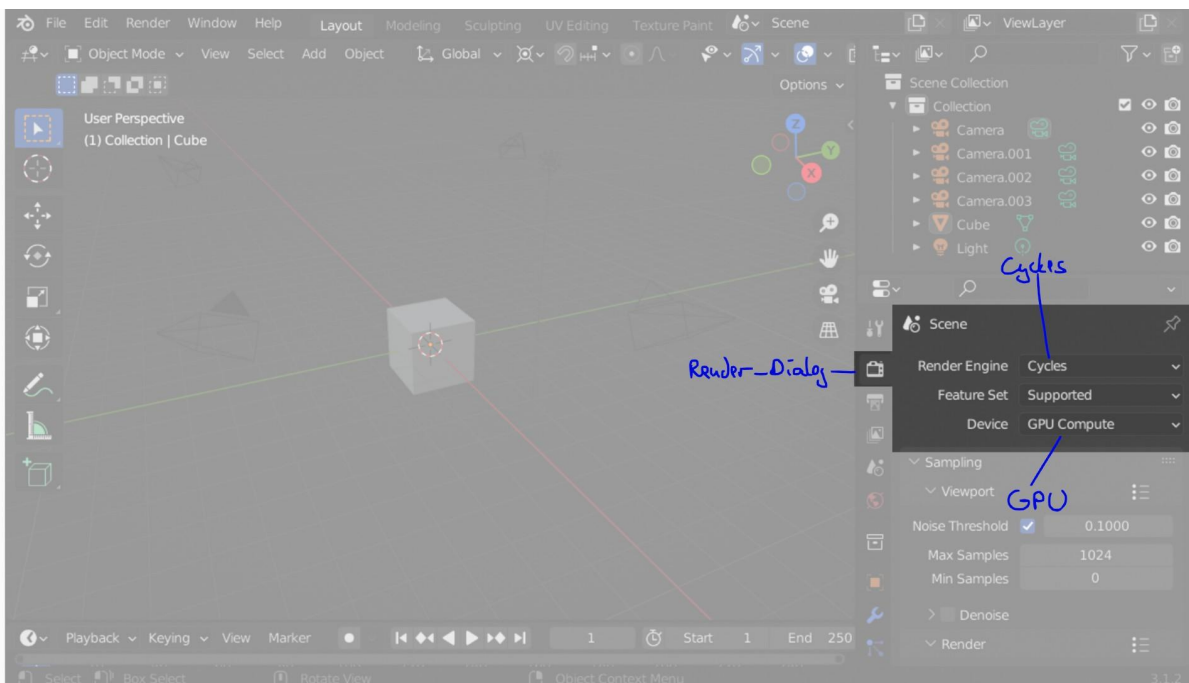


Abb. 24 Cycles wird als Render Engine ausgewählt mit Betrieb über die GPU

Nach Umstellen auf Cycles erscheint die 3D-View Ansicht mit Shading-Mode Rendering etwas verrauscht, was daran liegt, dass nun schon einzelne Lichtstrahlen (Rays) in die

Szene gesendet werden. Durch Druck auf F12 (oder Menü Render | Render Image) wird dann das Rendern ausgelöst und man erhält das Ergebnis des Renderprozesses in einem neuen Fenster, welches man dann mit Image | Save Image abspeichern kann.

## 6. Glossar

Begriff	Erklärung
Add-On	Zusätzliche Funktionen in Blender freischalten
Aktives Objekt oder Fläche	Das mit Links-Click oder in einer Folge über Drücken der Shift-Taste zuletzt selektierte Objekt bzw. Fläche ist aktiv (orange umrandet)
BSDF-Shader	Mathematische Beschreibung wie einfallendes in ausgehendes Licht umgerechnet wird (bidirectional-scattering-density-function); berechnet die Helligkeit & Farbe einer Oberfläche
CPU	Der Prozessor
Cycles	Renderer für nahezu-fotorealistische Ergebnisse mit geringer Vorbereitungs- aber längerer Renderzeit
Edge	deutsch Kante, bestehend aus 2 Kanten
Edit-Mode	Ermöglicht Manipulation von Position, Skalierung und Rotation von Knoten, Kanten und Flächen des dem Objekt zugeordneten Meshs.
Eevee	Sehr schneller Echtzeit-Renderer für künstlich wirkende Szenen; für nahezu-fotorealistische Ergebnisse ist der Vorbereitungsaufwand sehr viel größer als bei Cycles
Face	deutsch Fläche, begrenzt durch Kanten
Flatmap	Eine Textur, die im PBR-Shader als Effektschicht eingesetzt wird
Glanzpunkt	Reflektion der Lichtquelle auf glatten Oberflächen
GPU	Der graphische Prozessor
Material	Beschreibt die sichtbaren Eigenschaften eines Polygons (u.a. Farbe und Helligkeit)
Material-Preview -Shading	Darstellung des zugeordneten Materials mit dazugehörigen Texturen
Mesh	Kanten, Knoten und Flächen eines 3D-Objekts. Jedem 3D-Objekt ist genau ein Mesh zugewiesen. Ein konkretes Mesh kann mehreren 3D-Objekten gleichzeitig zugewiesen werden.
NGon	Polygon mit 5 oder mehr Knoten
Oberfläche	Auf das Gitternetz eines Objekts wird Packpapier aufgeklebt, welches dadurch eine sichtbare Oberfläche erhält
Objekt	Element der 3D-Szene, welches eine Position, Skalierung und Rotation, sowie ein zugewiesenes Mesh besitzt

Objekt-Mode	Ermöglicht Manipulation von Position, Skalierung und Rotation des Objekts
Outliner	Übersicht aller Objekte in der 3D-Szene; Sichtbarkeit, Renderbarkeit und Klickbarkeit kann eingestellt werden
PBR	Physically-Based-Rendering definiert Materialien durch einen geschichteten Aufbau von Effekten
Polygon	Ein zusammengehörender Kantenzug
Primitive Shape	Ein vorgefertigtes Standard-Objekt zum Einfügen (Kugel, Kubus, Donut etc.)
Principled BSDF Shader	Ein Basis-Shader, der alle bekannten Materialien abbilden kann. Eine mathematische Gleichung definiert, in welche Richtungen ein ankommender Lichtstrahl wie stark weitergeleitet wird
Quad	Polygon bestehen aus 4 Knoten
Raytracing	Das Verfolgen von Lichtstrahlen von der Kamera zu Objekt-Flächen, an denen die Strahlen (engl. Rays) i) absorbiert, ii) reflektiert oder iii) gebrochen werden. Bei ii) und iii) werden neue Strahlen erzeugt, die erneut verfolgt werden.
Render-Bild	Ergebnis des Renderings, Abbildung aller Objekte mit der eingestellten Szenen-Beleuchtung
Rendered-Shading	Das Viewport zeigt das Objekt, als würde es gerendert werden
Rendern	Der Prozess des Aussendens & Reflektierens von Strahlen und Einfärbens von Pixeln auf einem Bildschirm
Shading	Einfärben eines Pixels als Teil des Render-Prozesses
Selektiertes Objekt oder Fläche	Entweder das mit Links-Klick ausgewählte Objekt oder Fläche (ist dann auch aktiv) oder innerhalb einer Folge von nacheinander mit gedrückter Shift-Taste ausgewählter Objekte / Flächen (gelb umrandet)
Solid-Shading	Alle Flächen sind undurchsichtig, Objekte werden eventuell verdeckt
Szene	Umgebung mit virtuellen Objekten (engl. Scene)
Textur	Ein Bild / Fotografie, welches als Packpapier um ein Drahtgitter gewickelt wird, wird auch Flat-Map genannt.
Transformations-Eigenschaft	Ein X,Y, oder Z- Zahlenwert der Position / Location, Rotation oder Skalierung
Tri / Triangle	Polygon bestehend aus 3 Knoten
UV-Layout	Zuordnung von Texturbereichen des Packpapiers auf die Oberfläche des Objekts
Vertex	deutsch Knoten

Viewport	3D-View Fenster in das man mit der Arbeitskamera schaut
Wireframe	Drahtgitter aus Kanten und Knoten
Wireframe-Shading	Alle Flächen sind durchsichtig, um verdeckte Objekte zu erkennen
Workspace	Zusammenstellung von Fenstern mit ähnlichem Aufgabenbezug

## 7. Rettungs-Tastenkürzel

Folgende Tastenkürzel helfen bei Problemen oft weiter

Tastenkürzel	Bedingung	Effekt
Home	Der Mauszeiger befindet sich im 3D-View Fenster	Die Viewport-Kamera wird so ausgerichtet, dass wieder alle Objekte in der 3D-Szene gesehen werden
F12	Mindestens eine Render-Kamera muss vorhanden sein	Rendert die Ansicht der Render-Kamera mit dem Licht der 3D-Szene