

2022

ISSN 1433-2620 > 26. Jahrgang >> www.digitalproduction.com

Publiziert von BuschGlatz Germany GmbH

Deutschland € 17,90

Österreich € 19,-

Schweiz sfr 23,-

2

DIGITAL
PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

MÄRZ | APRIL 02:2022



Assets

Schnelle Hilfe mit Dropbox
Replay und das element

Projekte

Neoshin, Matrix 4, Marmalade,
Parasol und Kurzgesagt

Tools

Blender 3, Boris FX, Krita,
ZBrush und Colour Tools

Praxis

Digitale Menschen,
Video LUT und Vegas 19

Der Splashscreen von Blender 3.0 zeigt eine Szene aus „Sprite Fright“, dem neuen Kurzfilm der Blender Animation Studios in Amsterdam.



Blender 3.0 mit CyclesX, Asset Manager und Geometry Nodes

Mit Blender 3.0 beginnt ein neuer Release-Zyklus, weshalb wieder einige größere Änderungen aufgenommen wurden. Zu den Highlights gehören der lang ersehnte Asset Manager und das Rewrite der Path Tracing Render Engine Cycles als CyclesX, mit der RTX-Karten von Nvidia endlich so richtig ausgereizt werden. Außerdem funktioniert sie wieder auf AMD-Karten der neuesten Generation. Die Geometry Nodes haben ebenfalls ein riesiges Update erhalten. Sie verhalten sich jetzt endlich ähnlich wie sonstige Nodesysteme in Blender und können mit mehr als 100 neuen Nodes deutlich mehr Funktionalität abbilden. Aber das ist noch nicht alles, denn Neuigkeiten gibt es an fast allen Ecken und Enden.

von Gottfried Hofmann

Es ist jetzt schon etwas mehr als 10 Jahre her, da war ich Teil der Demo-Crew am Blender-Stand auf der FMX in Stuttgart. Gleich der erste Interessierte, dem ich die damals neuesten Features in Blender 2.50 präsentierte, fragte, wo sich denn der Asset Manager befinde. Ich zeigte den Link- und Append-Workflow von Blender und versicherte, dass ein Asset Manager unter den Entwicklern nicht nur diskutiert werde, sondern bereits in aktiver Entwicklung sei. Es könne also nicht mehr lange dauern. Etwas später im gleichen Jahr besuchte ich zum ersten Mal den Blenderday, die deutschsprachige Konferenz für Blender-User. Dort durfte ich eine Entwicklerin kennenlernen, die mir einen funktionstüchtigen Prototypen eines Asset Managers zeigte. Ich war voller Zuversicht.

Wie schon erwähnt, ist besagter erster Blick auf den Asset Manager jetzt schon etwas mehr als zehn Jahre her. Seither wurde er sicher schon mehrfach überarbeitet. Und als fertig kann er dennoch nicht bezeichnet werden. Aber er ist auch nicht mehr experimentell, sondern lässt sich direkt aktivieren und hat auch gleich sein eigenes Editorfenster bekommen.

Selbst ist der Blender-User

Wenn man ihn zum ersten Mal öffnet, ist der Asset Manager noch leer. Er muss zunächst mit Assets befüllt werden. Dafür gibt es die Möglichkeit, Materialien, Objekte und Welten als Asset zu speichern. Wünschenswert wäre an dieser Stelle auch die Unterstützung von Collections gewesen, worauf wir aber wahrscheinlich noch mindestens bis Blender 3.1 warten müssen. Als Asset kann ein Datenblock z.B. per Rechtsklick im Outliner markiert werden. Dann taucht er in der Assetliste der Blender-Datei auf. Dort kann man es mit einem eigenen Previewbild, einer Beschreibung und Autoreninformationen versehen sowie verschlagworten.

Vom Konzept her Altbewährtes

Wenn man das Asset immer zur Verfügung haben möchte, muss die Blender-Datei, in der es sich befindet, im User-Library-Ordner gespeichert werden. Die Assets aus allen in diesem Ordner befindlichen Blender-Dateien werden im Browser unter „User Library“ angezeigt. Auf diese Art hat man auf Dateisystemebene die Kontrolle über seine Assets und kann den Ordner ohne Probleme in zukünftige Versionen von Blender übernehmen oder z.B. über ein Netzlaufwerk oder gar Dropbox mit Kollegen teilen. Einer gemeinsam genutzten Asset-Bibliothek steht dadurch nichts im Wege.

Neues unter der Haube bricht Kompatibilität

Unter der Haube wurden daher einige Änderungen durchgeführt, die die Vorwärtskompatibilität einschränken. Beim Speichern von komprimierten Dateien wird jetzt der Standard-Algorithmus statt des bisherigen Gzip verwendet, was in manchen Fällen die Zeit zum Speichern auf ein Zwanzigstel, zum Laden auf ein Drittel reduziert. Zudem werden dadurch leicht kleinere Dateien erzeugt. Eine in 3.0 gespeicherte komprimierte Datei lässt sich dadurch aber nicht mehr in niedrigeren Versionen von Blender öffnen. Zur Python-Umgebung, die Blender mit jeder Installation mitbringt, wurde das Standard-Paket hinzugefügt, damit man komprimierte Dateien direkt manipulieren kann.

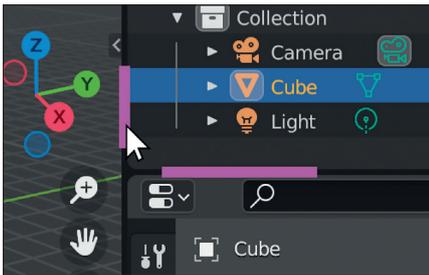
Eine weitere Änderung, die die Kompatibilität zu älteren Versionen einschränkt, ist eine Überarbeitung der Library Overrides. Modifier und Constraints sollten jetzt in lokalen Overrides wie gewünscht funktionieren. Dafür verschwinden sie, wenn man die Dateien mit älteren Versionen von Blender öffnet.

Proxies sollen im Laufe der 3.xer-Serie von Blender nach und nach verschwinden. In Blender 3.0 bedeutet das, dass keine Möglichkeiten mehr geboten werden, Proxies aus verlinkten Datenblöcken zu erzeugen. Vorhandene Proxies werden automatisch zu Library Overrides konvertiert. Man kann dieses Verhalten aber im Experimental-Bereich der Benutzereinstellungen ausschalten.

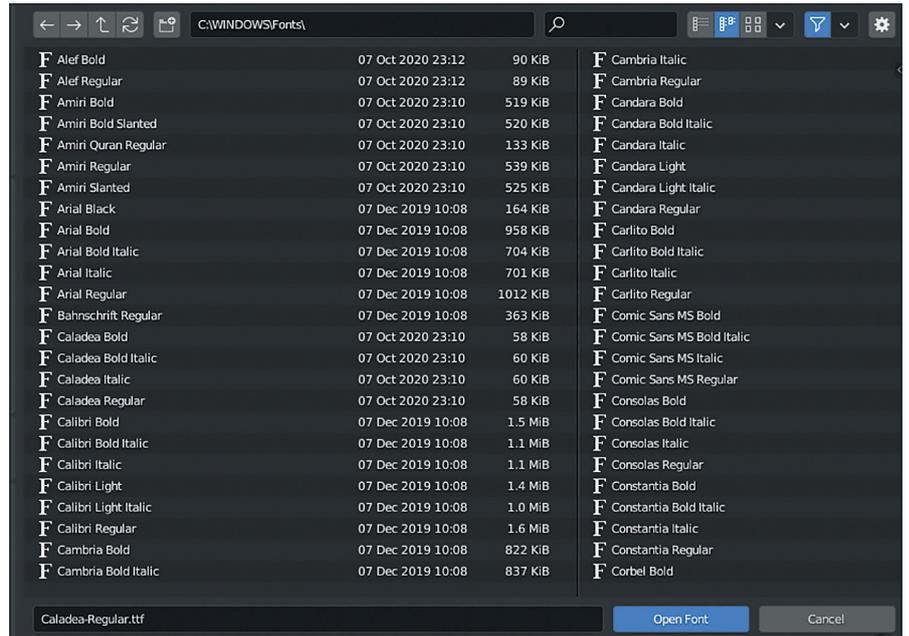
Die Namen von Vertexgruppen werden jetzt nicht mehr in Objekten, sondern direkt im Mesh gespeichert, wo sich auch die tatsächlichen Vertexgruppen befinden. Diese kleine und sinnvolle Änderung hat aber auch Auswirkungen auf die Rückwärtskompatibilität. Wenn man z.B. ein Mesh aus einer Datei verlinkt, die in Blender 2.93 erstellt wurde, und das Objekt lokal macht,



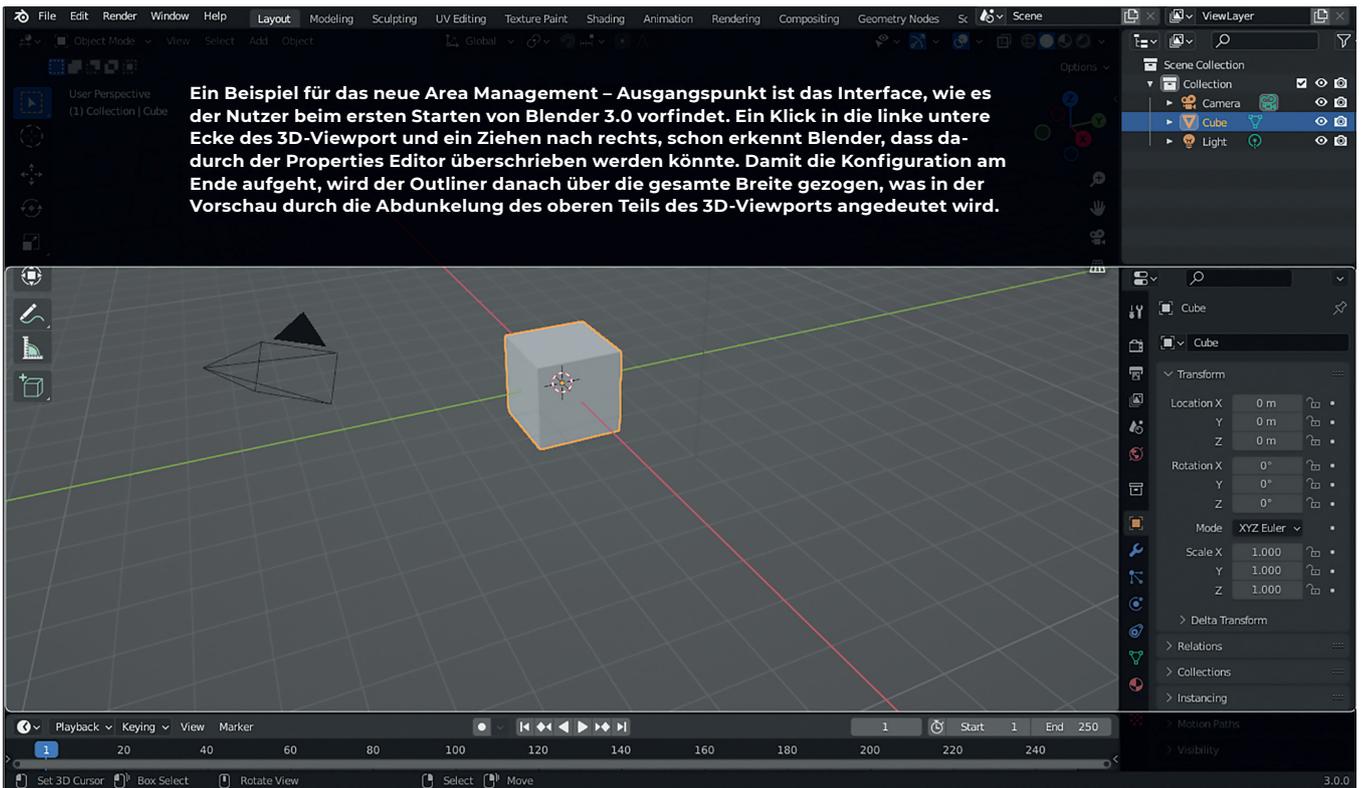
Der Editor für Custom Properties ist mächtiger und übersichtlicher geworden.



Grenze: Die Hitbox für Änderungen an den Editorkacheln wurde breiter gemacht.



Bei Schriftarten wird jetzt der Name der Schriftart angezeigt anstatt wie bisher der Dateiname.



das Mesh aber weiter verlinkt bleibt, kann Blender die Vertexgruppen des Meshes nicht mehr lesen. Es ist auch nicht mehr möglich, zwei unterschiedliche Listen an Namen für die Vertexgruppen in zwei unterschiedlichen Objekten zu haben, die sich das gleiche Mesh teilen. Die Referenzierung unterschiedlicher Bones über die Namen von Vertexgruppen bei Objekten, die sich das gleiche Mesh teilen, funktioniert ebenfalls nicht mehr.

UI-Änderungen

Auch das Interface hat für Version 3.0 einen Refresh erhalten. So wurde der Dialog zum Erzeugen von Custom Properties komplett überarbeitet. Schriftarten werden jetzt mit ihrem tatsächlichen Namen dargestellt statt wie bisher über ihren Dateinamen. Das Standardthema hat einen Refresh erhalten, wodurch 3.0 leicht anders aussieht

als die Vorgänger aus der 2.9er-Serie. Blender war bisher dafür bekannt, dass man aus Versehen Änderungen an der Benutzeroberfläche wie z.B. bei der Fensterkonfiguration machen konnte, die dann schwer wieder rückgängig zu machen waren. Letzteres sollte in Blender 3.0 nicht mehr der Fall sein, denn das Area Management ist jetzt deutlich einfacher, da für ein Verschmelzen die einzelnen Kacheln nicht mehr exakt

aneinanderpassen müssen. Damit ist es jetzt viel einfacher möglich, ein Editorfenster zu vergrößern auf Kosten von umliegenden Editoren. Zusätzlich gibt es jetzt einen neuen Operator, der ein Editorfenster bzw. ein Area direkt schließt, und der Bereich zwischen zwei Editorfenstern bzw. -kacheln, auf das man zum Ändern klicken kann, wurde breiter gemacht, damit man zum Bearbeiten der Kacheln mit der Maus keine Mikrobewegungen mehr durchführen muss.

Hinweis bei schreibgeschützten Dateien

Wenn man Blender schließt und die aktuelle Datei verändert wurde, fragt Blender nach, ob man sie nicht vor dem Schließen speichern möchte. Wenn die Datei aber schreibgeschützt oder anderweitig blockiert ist, wurde Blender dann einfach geschlossen, ohne die Änderungen zu speichern. Das ist ein Problem, das besonders beim verteilten Arbeiten mit Versionskontrollsystemen leicht zu Datenverlusten führen konnte. In Blender 3.0 wird der Nutzer in einem solchen Fall daher mit einer Fehlermeldung darauf hingewiesen, dass die Datei nicht geschrieben werden kann, und Blender bleibt geöffnet.

Das Widget bleibt

Wenn man in den Local View wechselt, beschränken sich die angezeigten Statistiken wie z.B. die Anzahl der Vertices, Edges und Faces jetzt auf die sich im Local View befindlichen Objekte. Wenn man im Edit Mode einen modalen Operator wie z.B. das Extrudierwerkzeug einsetzt, bleiben die Navigationswidgets jetzt sichtbar. Dadurch kann man sie zum Abschätzen von Positionen einsetzen, da man so jederzeit sehen kann, wohin gerade die World-Achsen zeigen. Der Grund, warum die Widgets bisher ausgeblendet wurden, war, dass bei modalen Operatoren keine gleichzeitige Navigation im Viewport möglich ist. Man wollte also gerade bei Neulingen Verwirrungen vermeiden. Da die

Widgets aber in einem solchen Fall sowieso nicht erreicht werden können, hat man sich dazu entschieden, sie dauerhaft anzuzeigen.

Neuerungen für Windows-Nutzer

Wenn man Blender unter Windows startet, blinkt nicht mehr schnell das Konsolenfenster auf, und wenn man in Windows rendert, erscheint das Fenster nicht mehr hinter dem Hauptfenster von Blender, sondern davor. Die Platzierung von Blender-Fenstern auf verschiedenen Monitoren wurde unter Windows ebenfalls verbessert, und man kann die aktuelle Blender-Installation in den Benutzereinstellungen jetzt mit einem Klick mit Blender-Dateien verknüpfen.

Vieleorts hübschere Thumbnails

Bei der Erzeugung von Vorschau Bildern wurde an vielen Stellen optimiert. So werden Thumbnails für Objekte, wie man sie z.B. im Asset Manager vorfindet, jetzt aus einem leicht schrägen Kamerawinkel erzeugt, damit man die Form besser erkennen kann. Starke Verzerrungen, wie sie bisher z.B. bei Köpfen vorkommen konnten, sollten damit der Vergangenheit angehören.

Thumbnails im Dateibrowser werden jetzt mit einer höheren Qualität skaliert, wodurch Aliasing- bzw. Treppchenartefakte der Vergangenheit angehören sollten. Die Thumbnails von Blenderdateien werden jetzt mit einer höheren Auflösung gerendert, und der User kann auswählen, ob beim Speichern ein Screenshot erzeugt werden soll oder ob das Thumbnail ein Render aus der aktiven Kameraansicht mit dem aktuell ausgewählten Viewport Shading werden soll.

Für Dateien, die auf einem Cloudspeicherdienst wie Microsoft OneDrive liegen und dort als offline markiert sind, wird keine Vorschau mehr generiert, damit die Anzeige nicht durch langwieriges Herunterladen von Daten blockiert wird. Bei der Materialvorschau wurden die UVs

der Kugel so angepasst, dass sie am Äquator quadratisch sind und an den Polen keine Zacken mehr haben.

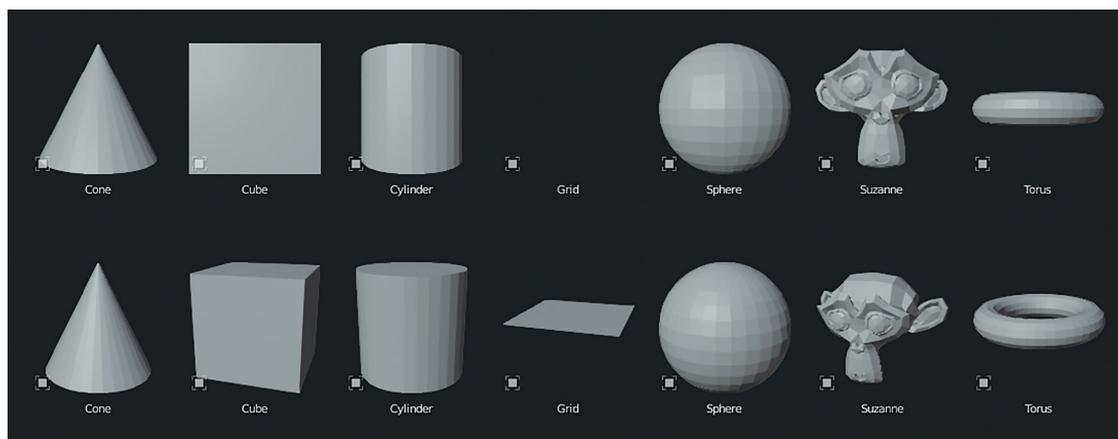
CyclesX ist stark mit RTX

Eines der Highlights von Blender 3.0 ist mit Sicherheit die Integration von CyclesX. Dabei handelt es sich um eine großangelegte Überarbeitung der Path Tracing Render Engine Cycles, die in Blender mitgeliefert wird. Hierfür wurde die Kommunikation von Cycles mit modernen Grafikkarten verbessert. Mit RTX-fähigen Grafikkarten und dem OptiX-Backend kann man sich auf einen 2- bis 8-fachen Performanceboost freuen. Dafür müssen die Grafikkartentreiber mindestens die Version 470 tragen. Das Kompilieren der Renderkernel zur Laufzeit wurde ebenfalls optimiert und sollte in typischen Fällen jetzt in unter einer Sekunde erledigt sein. OptiX und Denoising können jetzt auch für Baking eingesetzt werden.

Kein OpenCL mehr in Cycles, dafür HIP

Zur Unterstützung von AMD-Grafikkarten wurde in Cycles bisher OpenCL genutzt. Leider gab es immer wieder Treiberprobleme, und die Entwicklung des Standards war schon länger zum Stillstand gekommen. Daher hat man sich in Blender 3.0 entschlossen, das OpenCL-Backend komplett aus Cycles zu entfernen. Unter Mac OS bedeutet das momentan, dass die meisten User nur noch mit der CPU rendern können. Es gibt aber Pläne, GPU-Unterstützung für Mac OS mittels Metal in Version 3.1 zurückzuholen.

Windows-Nutzer:innen mit aktuellen AMD-Grafikkarten können sich hingegen jetzt schon darauf freuen, endlich genauso gut rendern zu können wie mit der Konkurrenz von Nvidia, denn es gibt jetzt ein HIP-Backend. Das funktioniert unter Windows mit Karten der RDNA- und RDNA2-Generation. Dazu gehören die Serien Radeon RX 5000 und RX 6000 sowie die Radeon



Durch eine Änderung des Kamerawinkels lassen sich die Objekte in ihren Thumbnails jetzt besser ausmachen. So werden Grid und Plane, die keine Ausdehnung in die dritte Dimension haben, überhaupt erst sichtbar und der Kopf von Suzanne erscheint nicht mehr verzerrt.

Pro W6000. Als Treiber werden Radeon Pro 21.Q4 oder Adrenalin 21.12.1 vorausgesetzt. Man arbeitet bereits an der Unterstützung von älteren Kartengenerationen und hofft, diese zusammen mit Linux-Support in Blender 3.1 unterzubringen.

Über das HIP-Backend können Besitzer von unterstützten AMD-Karten jetzt die gleichen Features auf der GPU nutzen wie bei Nvidia und gleichzeitig schneller rendern als früher mit OpenCL. Zudem müssen zur Laufzeit keine Kernel kompiliert werden, wodurch der Render sofort startet.

Endlich kein BVH-Neubau bei Animationen mehr

Wenn man eine Animation rendert, kann jetzt die BVH-Beschleunigungsstruktur zwischen verschiedenen gerenderten Frames wiederverwendet werden. Damit entfällt ein weiterer Grund für lange Renderzeiten, denn dies kann bei komplexer Geometrie durchaus länger dauern als das Rendern an sich. Ebenfalls aufgehoben werden kompilierte Shader und Texturen, die sich im GPU-Speicher befinden. Dank neuer Algorithmen für Scheduling und Anzeige geht die Arbeit im Viewport bei eingeschaltetem Render Preview deutlich flüssiger von der Hand als bisher. Beide Neuerungen helfen auch, wenn nur eine CPU zum Rendern verfügbar ist.

Keine Renderkacheln mehr

In Blender 2.93 wurde das Bild beim Rendern standardmäßig in kleinere Elemente, sogenannte Kacheln oder Tiles, unterteilt. Diese wurden nach und nach berechnet, bis das Bild fertig war. Eine der Möglichkeiten, auf die Performance Einfluss zu nehmen, war das Auswählen der richtigen Größe für die Tiles.

In Blender 3.0 wird standardmäßig progressiv gerendert, die Performance wird dabei automatisch im Hintergrund optimiert. Dabei ist das gesamte Bild sichtbar und klar Stück für Stück auf. Dadurch sieht man auch gleich eine Art Gesamtvorschau für das fertige Bild. Der Nachteil ist aber, dass dadurch deutlich mehr Arbeitsspeicher benötigt wird als bei kleinen Tiles, besonders wenn man sehr viele Render Passes eingeschaltet hat. Daher werden Bilder, die größer sind als 2.048x2.048 standardmäßig doch in Kacheln unterteilt. Wenn der Arbeitsspeicher für eine Szene nicht ausreicht, kann eine Reduktion der Tile Size Abhilfe schaffen.

Ganz perfekt ist die automatische Performance-Optimierung allerdings immer noch nicht. Daher gibt es immer noch Sonderfälle, in denen eine kleinere Tile Size hilfreich sein kann. Wenn ein Objekt zum Beispiel deutlich länger zum Rendern benötigt als der Rest der



Shadow-Catcher-Demoszene mit Boden als Shadow Catcher mit dem Principled BSDF in Standardeinstellungen im Compositor über den Hintergrund gelegt. Das vom Objekt abgestrahlte indirekte Licht gibt dem Schatten eine leicht bläuliche Färbung.

Quelle: The Cycles Encyclopedia



Demoszene des Shadow Catcher auf Metall mit Boden als Shadow Catcher mit dem Principled BSDF in Standardeinstellungen im Compositor über den Hintergrund gelegt. Der Boden hat Shadow Catcher aktiviert und ein Material mit 100% Metallic und einer Roughness von 0,2. Die klare Spiegelung zeigt, dass der Shadow Catcher jetzt selbst für Wasser- und Glasoberflächen geeignet ist.

Quelle: The Cycles Encyclopedia

Szene, was z.B. bei haarigen Charakteren passieren kann, besteht die Möglichkeit, mit einer kleineren Kachelgröße die Grafikkarte besser auszulasten.

Neuer Workflow mit Cycles

Da das adaptive Sampling verbessert wurde, haben die Entwickler auch die Voreinstellungen geändert. Die Anzahl der Samples ist sowohl für den Viewport als auch für den finalen Render deutlich höher als in 2.93, dafür ist jeweils der Noise Threshold schon gesetzt. Die Idee ist, dass sich über den Noise Threshold heuristisch entscheiden lässt, wann ein Pixel ausreichend Strahlen bzw. Samples erhalten hat, sprich wie stark oder wenig stark das Restrauschen ist. Alternativ kann man auch eine maximale Renderzeit pro Frame einstellen oder ganz klassisch die Anzahl an Samples pro Pixel festlegen.

Die neue Einstellung „Scrambling Distance“ erlaubt eine gewisse Korrelation zwischen Pixeln. Dadurch kann Rendern auf der GPU beschleunigt werden. Dies kann automatisch geschehen, was auch die von den Entwicklern empfohlene Einstellung ist.

Besseres Denoising

OpenImageDenoise von Intel wurde auf Version 1.4 upgedatet. Damit sollen Details besser erhalten werden, und über die

neue Prefilter-Option können Hilfs-Passes wie Albedo und Normal schon im Vorfeld entrauscht werden. Für Volume Rendering wurden diese beiden Passes ebenfalls verbessert, wodurch Dunst und Nebel in der Szene besser erhalten bleiben sollten.

Neuer Shadow Catcher

Der Shadow Catcher in Cycles wurde ebenfalls überarbeitet und ist jetzt mächtiger als je zuvor. Es handelt sich dabei um ein Objekt, das selbst transparent ist, aber Licht und Schatten aus der Umgebung speichern kann. Bisher funktionierte das nur mit Schatten, doch in Blender 3.0 kann der Shadow Catcher jetzt auch indirektes Licht einfangen und es gibt die Möglichkeit, Lichtquellen selektiv von ihrer Wirkung auszunehmen. Um alle Features des neuen Shadow Catchers nutzen zu können, muss der entsprechende Render Pass aktiviert und im Compositing mit dem Hintergrund multipliziert werden. Nur dann wird das indirekte Licht mitgenommen. In der Viewport-Vorschau zeigt der Shadow Catcher lediglich die Schatten an.

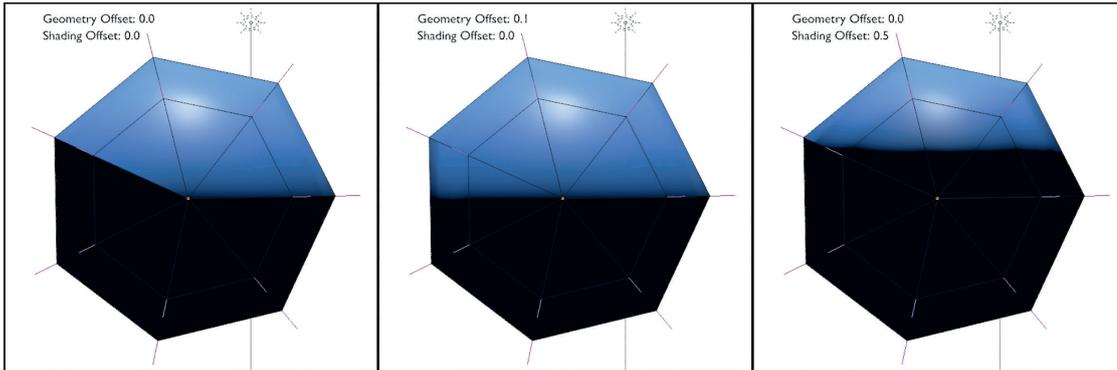
Nie wieder Shadow-Terminator-Artefakte

Bei Low-Poly-Objekten mit Smooth Shading zeigen Path Tracing Engines wie Cycles Artefakte beim Übergang zwischen Licht



Demoszene für Shadow Terminator Geometry Offset in Low-Poly-Variante. An der Grenze zwischen Licht und Schatten zeigen sich im linken Bild die typischen Shadow-Terminator-Artefakte. Bereits ein winziger Geometry Offset reicht, um das Problem zu beheben (Mitte). Noch höhere Werte helfen nicht weiter (rechts).

Quelle: The Cycles Encyclopedia



Shadow Terminator Geometry Offset vs. Shading Offset Low-Poly Icosphere mit Shade Smooth und einer Sonne als Lichtquelle, die senkrecht von oben scheint. Im linken Bild ist deutlich ein Shadow-Terminator-Artefakt sichtbar. In der Mitte wurde der Geometry Offset genutzt, um das Artefakt verschwinden zu lassen, rechts der Shading Offset. Da der Shading Offset das Artefakt kaschiert, indem die Grenze zwischen Licht und Schatten verschoben wird, wird dadurch auch der Schattenbereich größer.

Quelle: The Cycles Encyclopedia

und Schatten, dem sogenannten Shadow Terminator. Diese entstehen dadurch, dass die Objekte zwar aus sehr großen Polygonen bestehen, aber durch das Shading eigentlich rund dargestellt werden sollen. An den Polygongrenzen kann diese Scharade nicht aufrecht erhalten werden, und eine harte Kante entsteht. Bisher konnte man in Cycles mit dem Shadow Terminator Offset dagegenwirken, indem die Schattengrenze in Richtung der Lichtquelle verschoben wurde. Dadurch wurde aber auch der Schatten verändert, weshalb eine alternative Methode hermusste. Der neue Geometry Offset verschiebt den Shading-Punkt ein wenig in Richtung der Normalen und kann damit das Problem lösen, ohne den Bereich im Schatten zu vergrößern. Die bisherige Methode steht weiterhin zur Verfügung, wurde aber umbenannt in Shading Offset.

Große Änderungen beim Subsurface Scattering

Beim Subsurface Scattering wurden die Methoden „Cubic“ und „Gaussian“ entfernt, es verbleiben „Christensen-Burley“ und „Random Walk“. Die letztere Methode arbeitet nach physikalischen Prinzipien der Lichtausbreitung in Objekten und wurde mit den neuen Parametern „Anisotropy“ und „IOR“ aufgeböhrt. Für menschliche Haut wird ein Anisotropy-Wert von ca. 0,8 empfohlen. Die zusätzlichen Parameter und die gewählten Farben haben dabei Einfluss auf die Scattering-Distanz. Wem das zu komplex ist, der kann weiterhin die bisherige Methode ein-

setzen, die in „Random Walk (Fixed Radius)“ umbenannt wurde.

Was noch alles rausgeflogen ist

Neben den beiden Subsurface-Scattering-Algorithmen wurden noch weitere Features aus Cycles entfernt. Dazu gehört das Sampling-Muster „Correlated Multi Jitter“. Es verbleiben nur „Sobol“ und „Progressive Multi Jitter“, Letzteres ist für Adaptive Sampling Pflicht. Der NLM-Denoiser wurde ebenfalls entfernt, die Entwickler empfehlen stattdessen OpenImageDenois, das ebenfalls auf der CPU funktioniert, aber deutlich bessere Ergebnisse erzielt.

Branched Path Tracing wurde ebenfalls komplett aus Cycles entfernt. Die Entwickler empfehlen stattdessen, über Adaptive Sampling automatisch dort mehr Samples einzusetzen, wo noch viel Rauschen im Bild vorhanden ist. Was allerdings noch nicht zurückgekommen ist, ist Many Light Sampling, das eine der Stärken von Branched Path Tracing war. Dabei werden Shadow Rays nicht nur zu einer, sondern gleich zu mehreren Lichtquellen verschickt. Aufgrund seiner Nützlichkeit sollte dieses Feature in Zukunft separat wieder auftauchen.

Was in Cycles noch alles dazugekommen ist

Cycles hat auch eine Reihe kleinerer Features erhalten. So können Lichtobjekte wie Point Lights, Spot Lights und Area Lights jetzt auch direkt für die Kamera sichtbar gemacht

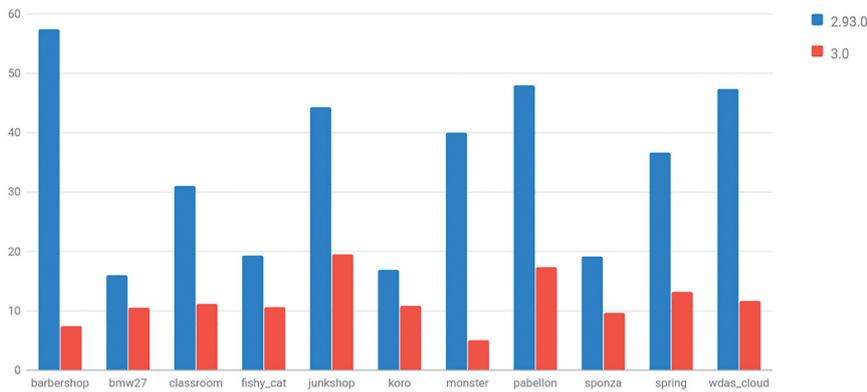
werden. Bei Szenen mit volumetrischen Shadern ist deren Darstellung jetzt auf der CPU und auf der GPU identisch. Ein neuer Render Pass erlaubt die Ausgabe der Objektpositionen im 3D-Raum, und Light Clamping soll jetzt nicht verrauschtes Licht besser erhalten, anstatt dessen Stärke einfach hart zu deckeln. Wer seine eigenen Shader mit der Open Shading Language schreibt, kann den so entstandenen Nodes jetzt Metadaten mitgeben, damit die Slider, Checkboxes und Dropdowns eigene Namen erhalten.

Eevee

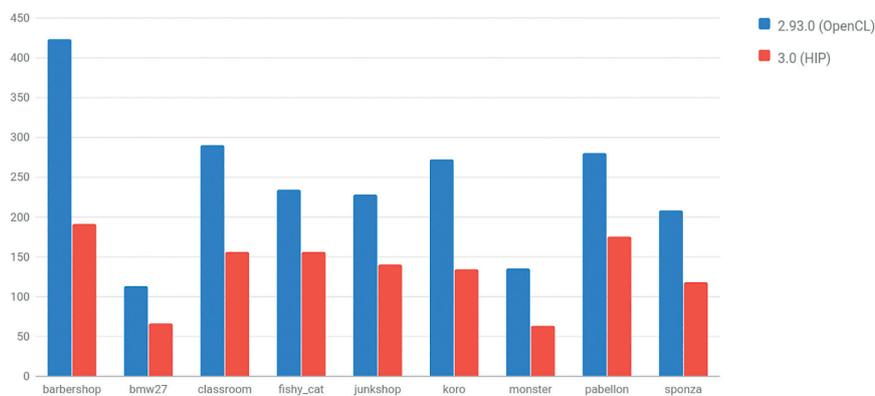
Der Attribute Node kann jetzt auch in Eevee Custom Attributes auslesen, wie sie z.B. von Geometry Nodes generiert werden können, und der Wavelength Node, der bisher nur in Cycles zur Verfügung stand, funktioniert jetzt auch in Eevee. Intern verwendet er eine Color Map ähnlich der Blackbody Node.

Fields statt Attribute in den Geometry Nodes

Mit den Geometry Nodes in Blender können Objekte, Meshes und mehr prozedural generiert und bearbeitet werden. Die erste Implementierung war zwar bereits mächtig, von der Anwendung her aber für Blender-User ungewohnt. Denn anstatt, wie in Blender üblich, Daten über die Verbindungen



OptiX Benchmark: Unterschied in der Renderzeit zwischen Blender 2.93 (blau) und 3.0 (rot) bei unterschiedlichen Benchmark-Dateien mit einer Nvidia Quadro RTX A6000 und OptiX, durchgeführt von Brecht van Lommel



AMD HIP Benchmark: Unterschied in der Renderzeit zwischen Blender 2.93 mit OpenCL (blau) und 3.0 mit HIP (rot) bei unterschiedlichen Benchmark-Dateien mit einer AMD Radeon RX 5500 XT, durchgeführt von Brecht van Lommel

zwischen Nodes zu transportieren, wurden diese in speziellen Attributen gespeichert, die an anderer Stelle wieder aufgerufen und verändert werden konnten. Während diese Art zu Arbeiten für Programmierer an der Tagesordnung steht, ist man von den Node-Systemen in Blender doch eher eine visuelle Repräsentation des Datenflusses gewohnt.

Mit den neuen Fields in Geometry Nodes ist genau dieses Problem angegangen worden. Daten fließen jetzt über herkömmliche Nodeverbindungen. Funktionen, die auf diesen Daten arbeiten, werden über gestrichelte Linien visualisiert. Alternativ kann man sich



OpenImageDenoise-1.4-Demoszene mit SSS-Shader auf dem Blobfish mit 8 Samples pro Pixel gerendert und mit OpenImageDenoise mit allen Passes entrauscht. Links ohne Prefilter, rechts mit. Der Prefilter reduziert hauptsächlich Artefakte, die an den Kanten von Objekten auftreten. Besonders sichtbar ist dies am Bilderrahmen.

Quelle: The Cycles Encyclopedia



OpenImageDenoise-1.4-Demoszene mit SSS-Shader auf dem Blobfish mit 8 Samples pro Pixel gerendert und mit OpenImageDenoise mit allen Passes und eingeschaltetem Prefilter entrauscht. Selbst die über eine Bump Map erzeugten Strukturen auf der Wand werden erhalten.

Quelle: The Cycles Encyclopedia



Unter der Oberfläche: Die drei in Blender 3.0 angebotenen Methoden für Sub-surface Scattering im Vergleich. Bei Random Walk bleiben feine Details etwas besser erhalten.

Quelle: The Cycles Encyclopedia

Fields wie Daten vorstellen, die bei jedem Durchgang einen anderen Wert annehmen können. Die Evaluierung funktioniert dabei so ähnlich wie beim Shader Graph in Eevee und Cycles, wodurch man sich schnell an das System gewöhnt. Netter Nebeneffekt: Die Nodes für prozedurale Texturen konnten direkt aus dem Shader Editor portiert werden, man kann jetzt Noise, White Noise, Gradient, Voronoi, Checker, Magic, Wave und Musgrave in Geometry Nodes exakt gleich verwenden wie in Eevee und Cycles. Im Modifier-System war das bisher nicht möglich, da dort nur auf das Legacy-Textursystem aus Zeiten von Blender Internal zugegriffen werden konnte.

Attribute können natürlich weiter erzeugt und verwendet werden, sie dienen jetzt aber mehr der Kommunikation mit anderen Teilen von Blender, z.B. um sie beim Rendern abzurufen.

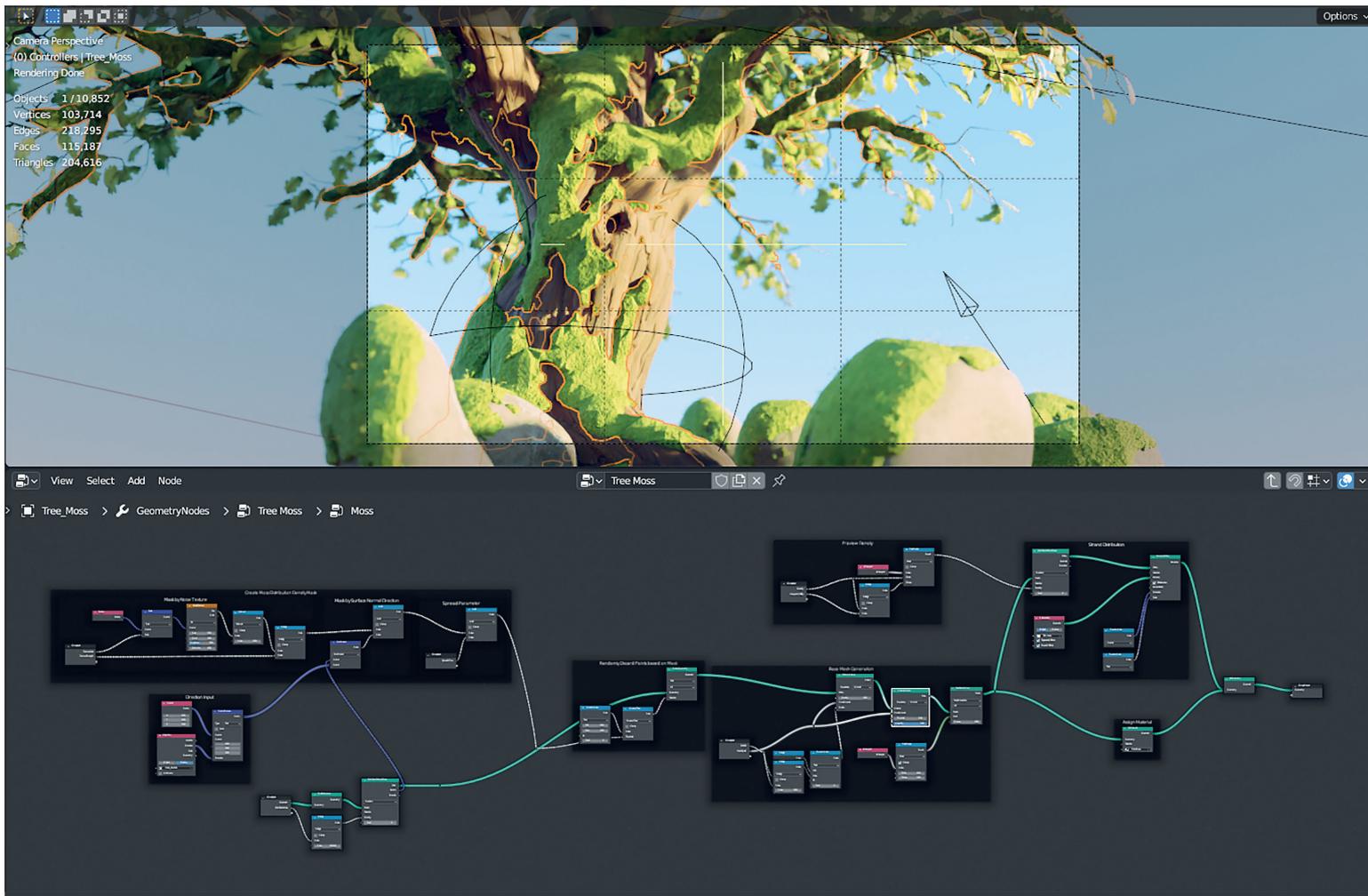
Über 100 neue Geometry Nodes

Auch bei den Funktionen in Blender, die von Geometry Nodes abgedeckt werden, hat sich einiges getan. Mehr als 100 Nodes sind dazugekommen. Zu den Highlights zählen sicher die zahlreichen Nodes, mit denen sich Kurven bearbeiten lassen. Von der Spline Resolution bis zum Tilt, egal welches Attribut einer Kurve man auch bearbeiten möch-

te, es gibt den dazu passenden Node. Hinzu kommt eine Auswahl an Curve Primitives wie Line, Star, Spiral oder Circle.

Text per Geometry Nodes bearbeiten

Auf den Curve Nodes bauen die neuen Text Nodes auf, deren Herzstück der Node „String to Curves“ ist. Diese erzeugt Instanzen der einzelnen Buchstaben in Form von Kurven, was eine weitere Neuerung für die Geometry Nodes darstellt. Denn diese können jetzt deutlich besser mit Instanzen umgehen. Bis Blender 2.93 wurden die meisten Instanzen relativ schnell implizit zu echter Geometrie



Moos und Blätter mit Geometry Nodes – ein Demo von Simon Thommes zum prozeduralen Verteilen von Moos und Blättern auf unterschiedlichen Objekten

Lizenz: CC-BY



Controller im VR-Modus – das „Race Spaceship“ von Alessandro Chiffi im VR-Modus mit visualisierten Controllern

konvertiert, was weder der Performance noch dem Speicherverbrauch zugute kam. In Blender 3.0 wird jetzt versucht, Instanzen erst möglichst spät in der Evaluation des Node Graphs zu realisieren. Dadurch ist der Text to Curve Node in vielen Fällen performanter als das gewöhnliche Textwerkzeug in Blender, da jeder Buchstabe nur einmal konvertiert werden muss und bei jedem weiteren Vorkommen im Text zu einer Instanz gewandelt wird.

Materialien zur Laufzeit ersetzen

Das Management von Materialien kann jetzt ebenfalls in Geometry Nodes durchgeführt werden. Das hat den Vorteil, dass Material Slots endlich relativ einfach zur Laufzeit ausgetauscht werden können. Der Geometry Nodes Modifier wird auch auf Volume Objects unterstützt.

Änderungen am Node Editor

Warnungen zu Problemen im Geometry Nodes Tree werden jetzt auch im Modifier angezeigt, damit man sie weniger leicht übersieht. Verbindungen zwischen Nodes erhalten jetzt die Farbe der Sockets, mit denen sie verbunden sind, und Nodeverbindungen, die über die Anzeigegrenze des Node Editor hinausreichen, werden etwas abgedunkelt, damit besonders lange Verbindungen nicht zu visueller Unordnung führen. Wenn man einen Node oder eine Verbindung an die Grenzen des Editorfensters zieht, folgt die Editoransicht automatisch, damit man nicht mehr die Verbindung ablegen und neu erzeugen muss, wenn man an den Rand des Editors gelangt. Zusätzlich wurde bei den Geometry Nodes an vielen Stellen an der Performance geschraubt.

VR Scene Inspection mit Controller-Support

Über das mitgelieferte Add-on „VR Scene Inspection“ kann man schon länger Blender-Szenen direkt in VR betrachten. Bisher war man dabei aber auf festgelegte Punkte und deren nähere Umgebung beschränkt. In Blender 3.0 wurde Controllerunterstützung nachgereicht. Damit lassen sich die Controller in VR nicht nur visualisieren, sondern auch für typische VR-Navigation per Teleport einsetzen. Die bisher für die Navigation eingesetzten Landmarks – das sind vom User vordefinierte Aussichtspunkte im 3D-Raum, zwischen denen während einer VR-Session hin- und hergeschaltet werden kann – können jetzt mittels beliebiger Objekte definiert werden. Zuvor war dies Kameraobjekten vorbehalten. Wenn es die VR-Brillen hergeben, wird das VR-Rendering jetzt in einer höheren Farbtiefe durchgeführt, und es werden Varjo Headsets mit Quad-View und Augentracking unterstützt (Varjo VR-3, XR-3).

Videoschnitt in Blender jetzt endlich erwachsen

Auch der in Blender integrierte Videoschnitteditor hat ein großes Update erhalten. Erstmals ist es nun möglich, im Vorschaufenster mittels Widgets die Position, Rotation und Skalierung eines Clips zu bearbeiten. Strips können mit Color Tags eingefärbt werden und zeigen ihren Inhalt über Thumbnails direkt im Sequence Editor an. Zudem lassen sich jetzt bis zu 128 Strips übereinander platzieren; bisher waren nur 32 auf einmal möglich. Das Layout des Speed Effect Strips, mit dem man die Geschwindigkeit eines Videos oder eine Bildsequenz bearbeiten kann, wurde überarbeitet, das Snapping

verbessert und die Funktion, alle Strips am aktuellen Frame auszuwählen, wurde zurückgebracht. Daneben wurde wie an so vielen anderen Stellen in Blender 3.0 auch beim Video Sequence Editor die Performance verbessert. Insgesamt fühlt sich Blender jetzt fast wie ein vollwertiges Videoschnittprogramm an, wenn man davon absieht, dass die Bedienung auf Blender-Nutzer:innen und nicht professionelle Cutter zugeschnitten ist.

Weitere Performance-Schübe

An der Performance-Schraube wurde auch an vielen weiteren Stellen gedreht. Große Projekte mit vielen Verlinkungen aus der gleichen Bibliothek laden jetzt deutlich schneller. Die Anzeige von Text im Interface wurde ebenfalls optimiert, und beim Verschieben von Nodes sollten deren Inhalte weniger häufig hin- und herspringen. Beim Bearbeiten von sehr großen Meshes funktionieren manche Operationen jetzt zwei- bis dreimal so schnell, und insgesamt sollten sich Änderungen an solcherlei großen Meshes jetzt deutlich flüssiger anfühlen.

Fazit

Blender 3.0 bringt so viele Performance-Vorteile und lang ersehnte Features, dass man eigentlich nicht zu Version 2.93 zurück möchte, auch wenn diese als Long-Term-Stable-Version (LTS) natürlich viel stabiler arbeitet. Man sollte allerdings beachten, dass es einige Änderungen gibt, die die Kompatibilität mit den Vorgängerversionen einschränkt und in seltenen Fällen ein lieb gewordenes Feature fehlt. Wer Skripte für Blender entwickelt, sollte besondere Vorsicht walten lassen, denn die umfangreichen Änderungen haben sich auch in der Python-API niedergeschlagen. > ei



Gottfried Hofmann ist Diplom-Informatiker und bietet seit mehreren Jahren professionellen Support sowie Schulungen für die freie 3D-Software Blender an. Als freischaffender Autor schreibt er für Fach- und Computerzeitschriften. Er hat zahlreiche Blender-Tutorials verfasst, u.a. für CG Tuts+ und CG Cookie. Weiterhin betreibt er die Webseite www.BlenderDiplom.com, auf der Blender-Tutorials in deutscher und englischer Sprache zur Verfügung stehen und Schulungen gebucht werden können, und hilft bei der Organisation von BlenderDay und Blender Summer School in Mannheim.