

2020

ISSN 1433-2620 > 24. Jahrgang >> www.digitalproduction.com

Publiziert von Pixeltown GmbH

Deutschland € 17,90

Österreich € 19,-

Schweiz sfr 23,-

2

DIGITAL
PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPR

MÄRZ | APRIL 02:2020



Projekte

Känguru-Chroniken, Spides,
Woodville und mehr ...

Hardware

MacBook Pro, Live-Grading,
BMD Keyboard, Sigma fp

und vieles mehr

Flame, ZBrush 2020, Quad
Remesher, Substance Tools



4 194336 217907 02



Der Weltraumdock von Tobias Günther ist im Rahmen der Blender Summer School 2019 entstanden und kann von <https://blender3dschool.de/eevee-pause-am-weltraumdock/> heruntergeladen werden. Für die Reflexionen auf den Drohnen kommen Cubemaps zum Einsatz.

Extremely Extendable Virtual Environment Engine (Eevee)

Einer der großen Hingucker von Blender 2.8x ist definitiv die echtzeitfähige OpenGL-Render-Engine Eevee. Sie hat den schon arg in die Jahre gekommenen Blender Internal Renderer beerbt, welcher sich nach vielen treuen Jahren aus Blender verabschiedet hat. Eevee basiert wie Cycles auf dem PBR-Workflow mit Node-Interface und dient als Basis für den neuen Blender-Workflow, bei dem man zwischen Eevee und Cycles hin- und herschalten kann. Damit wird es auch möglich, lange in Echtzeit zu arbeiten und erst beim letzten Feinschliff auf Cycles zu wechseln. Oder auf das letzte Quäntchen Realismus zu verzichten und ein Projekt komplett in Eevee zu realisieren.

von Gottfried Hofmann

Mit Eevee setzt Blender die Tradition konsequent fort, Namen auszusuchen, die bei der Internetrecherche mit den Bezeichnungen von anderen Dingen kollidieren. Wer schon etwas länger dabei ist, kennt wahrscheinlich noch das Problem, dass bei Google-Suchen nach Blender-Themen gerne Ergebnisse von Küchengerät-Herstellern präsentiert werden. Das Gleiche

galt in der Anfangszeit der mit Blender mitgelieferten Path-Tracing-Engine Cycles für Fahrräder. Jetzt also Eevee. Was herauskommt, wenn man das Wort bei Google eingibt, überlassen wir hier den Lesern.

Vielleicht ist Eevee ja auch ein Akronym. So wirklich festgelegt ist das noch nicht. Auf einer Präsentationsfolie erschien mal Extremely Extendable Virtual Environment Engine (Eevee), aber es könnte auch ein Elegantly Engineered Visualizer of Extreme Efficiency oder der Especially Efficient Visual Evaluation 'Elper sein. Oder darauf anspielen, wie die Zukunft als Freelancer oder Inhaber eines Studios aussehen mag, wenn man auf Eevee wechselt: Eagerly Expecting Vast Earnings Exclamationmark (Frederik Steinmetz hat sich darüber in einem Lighting Talk auf der Blender Conference ausgelassen: https://youtu.be/o4BTL_tLzcs?t=3222). Ganz egal was hinter der Bezeichnung stecken mag, Eevee ist ein guter Kompagnon zu Cycles.

Eevee or Cycles – what's the difference?

Die extrem gut aussehenden Eevee-Demos, welche auch diesen Artikel zieren, haben viele Nutzer neugierig gemacht, und so manch einer fragt sich vielleicht, warum man noch stundenlang mit Cycles rendern sollte, wenn doch Eevee mit ein paar Ab-

strichen das Ergebnis in Echtzeit ausspuckt. Aber Cycles hat weiterhin seinen Platz im Werkzeugkasten von Blender.

Eevee ist ein Rasterisierer

Grundsätzlich erreicht Eevee die Qualität aktueller Physically-Based-Rendering-fähiger Spiele-Engines wie Unreal oder Unity. Auch die Technologie dahinter ist ähnlich, nämlich die gute alte Rasterisierung, auf die Grafikkarten noch immer zugeschnitten sind und die daher extrem performant berechnet werden kann. Dabei wird die sichtbare Geometrie auf eine Ebene, das sogenannte Raster, projiziert. Im Anschluss wird für jeden Pixel berechnet, von welchen Lichtquellen er wie viel Licht abbekommt.

Es kommen natürlich noch einige moderne Extras hinzu wie zusätzliche Berechnungen im Screen Space, aber die Basis ist die Rasterisierung, woraus sich auch die Limitierungen von Eevee ergeben, auf die wir noch zu sprechen kommen werden.

Cycles nutzt Path Tracing

Cycles hingegen nutzt Path Tracing zum Rendering. Diese Form der Bildberechnung wird dann eingesetzt, wenn höchsten Ansprüchen an Qualität und Fotorealismus genügt werden muss wie zum Beispiel bei Kinofil-

men oder wenn einfach zu viel Geometrie in der Szene vorhanden ist. Dabei wird von der Kamera aus für jeden Pixel eine Anzahl an Strahlen, die sogenannten Samples, in die Szene geschickt. Diese springen so lange herum, bis sie eine Lichtquelle treffen. Der Weg des Strahls wird dann zurückverfolgt und es wird berechnet, welchen Anteil er an der Beleuchtung des Pixels hat.

Sichtbarkeit

Damit ist bereits einer der Kernunterschiede zwischen Cycles und Eevee angesprochen. Eevee kann prinzipiell nur darstellen, was die Kamera direkt sieht. Bereits das Einschalten einfacher Transparenz erhöht nicht nur die Renderzeit, sondern man hat dann auch schon die Qual der Wahl, welche Methode man einsetzen will. Aber auch Schatten, indirektes Licht, Refraktionen und schon einfache Reflexionen sind bei Eevee Limitierungen unterlegen, deren Reduktion zusätzliche Arbeit bedeuten kann.

Spiegeln, Spiegeln ...

Spiegelungen sind hier ein gutes Beispiel. Standardmäßig spiegelt sich in Eevee in den

Objekten nur die Umgebung. Will man jetzt auch andere Objekte aus der Szene in den Spiegelungen sehen, kann man zunächst Screen Space Reflections einschalten. Doch in diesen erscheint nur, was auch direkt für die Kamera sichtbar ist. Die Rückseite von Objekten kann in einer solchen Spiegelung zum Beispiel gar nicht erscheinen.

Zur Probe

Die Antwort auf das Problem sind Light Probes. Handelt es sich bei dem spiegelnden Objekt um eine ebene Fläche, sollte man eine Reflection Plane einsetzen. Diese rendert ein weiteres Bild, welches dann als Reflexion auf Flächen gelegt wird, die mit ihrer Normalen in die gleiche Richtung zeigen wie die Probe und die sich innerhalb der Distance befinden. Dadurch werden halbwegs akkurate Reflexionen möglich, wenn auch nur auf planen Flächen.

Für sonstige Objekte muss man dann schon zur Reflection Cubemap greifen. Weil bei dieser gleich sechs Bilder gerendert werden müssen, macht sie das nicht für jeden Frame, sondern sie wird einmalig gebacken. Sprich wenn es um sich bewegende Objekte geht, hilft diese leider nicht. Denn ein Auto-

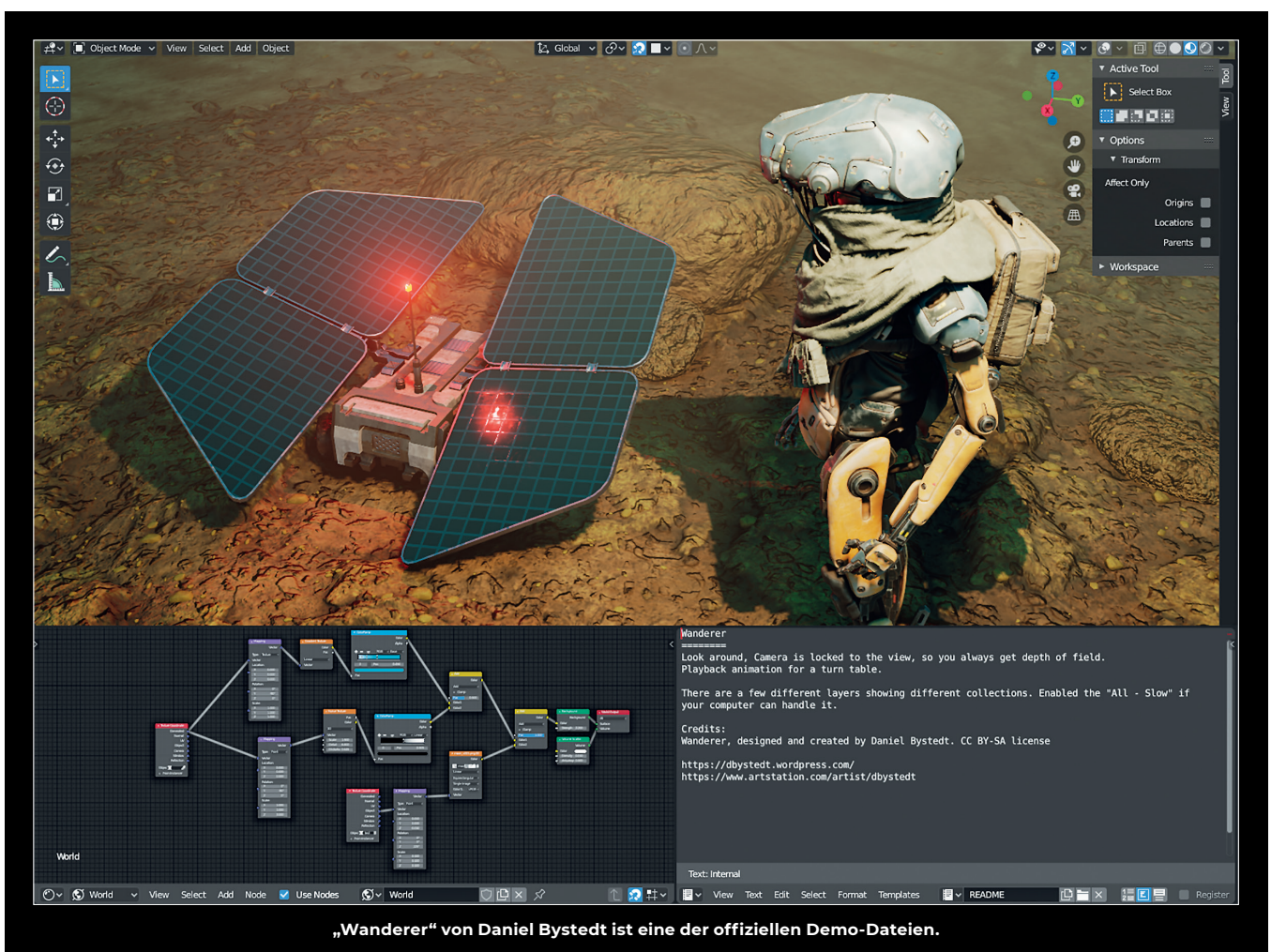
Bake-Feature ist noch nicht implementiert. Analog verhält es sich mit der Irradiance Probe für indirektes Licht.

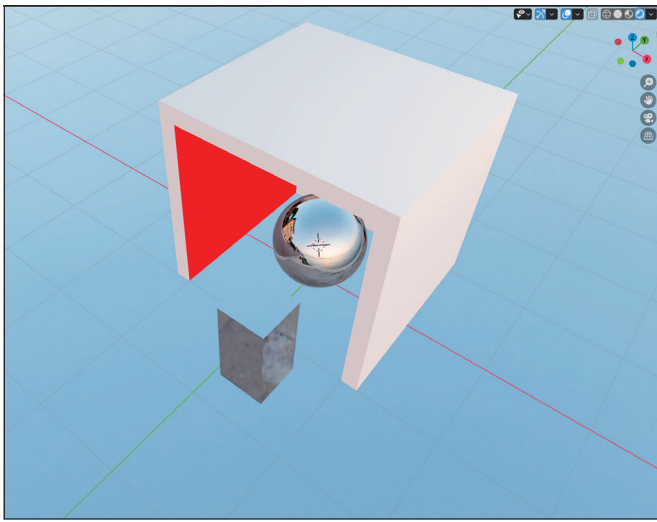
Manchmal hat Cycles die Nase vorn

Wir haben eine Bilderserie mit verschiedenen Möglichkeiten erstellt, in Eevee Reflexionen darzustellen. Die Szene mit den spiegelnden Objekten auf spiegelndem Boden und einer diffusen Cornell Box ist bewusst so gewählt, dass sie mit Eevee schwierig zu realisieren ist. Screen Space Reflections, Reflection Planes und Reflection Cubemaps wollen sinnvoll eingesetzt und kombiniert werden. Und selbst dann hat Cycles in dieser Szene die Nase vorn, wie die Ground Truth zeigt. Möglicherweise könnte sich dies aber ändern, sobald Raytraced Reflections Einzug in Eevee halten.

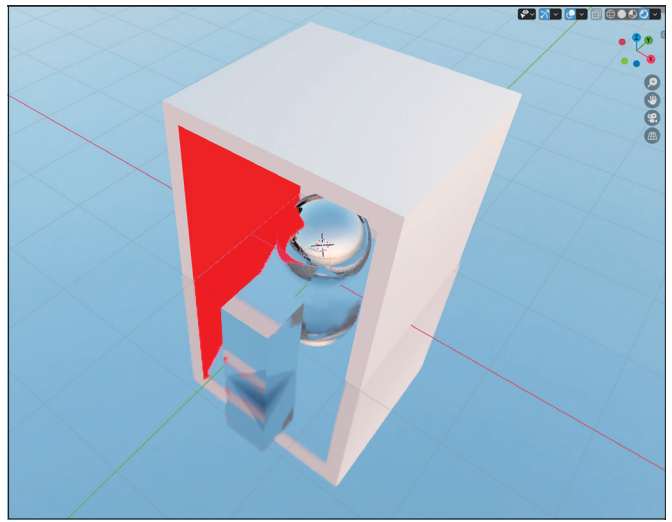
Materialvorschau

Wo Eevee jetzt schon glänzt, ist die Vorschau von Materialien. Solange keine Refraktion oder mehrfache Spiegelungen für die Evaluation eines Shader-Setups notwendig sind, zeigt Eevee hier Resultate, die Cycles sehr

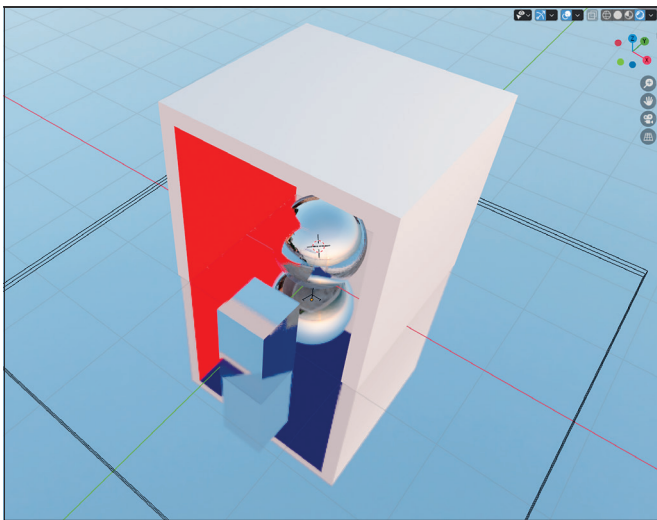




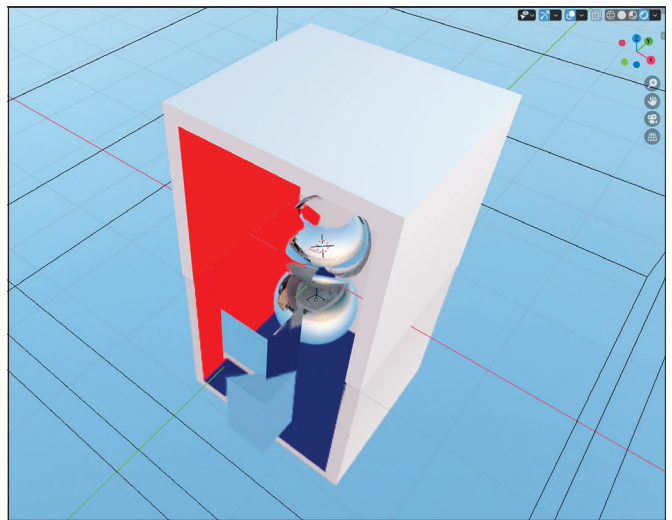
Default – Eine Cornell Box mit blauer Decke auf einer spiegelnden Ebene, in der eine spiegelnde Kugel und neben der ein spiegelnder Quader stehen. Die Einstellungen entsprechen den Standardeinstellungen von Eevee.



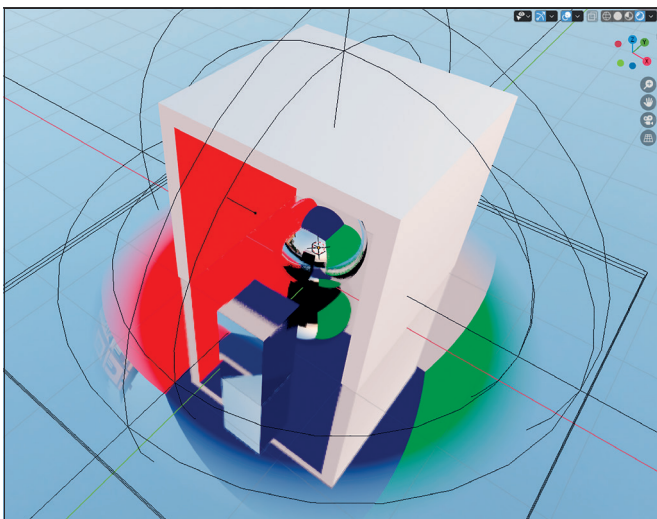
Screen Space Reflections – Die gleiche Szene mit Screen Space Reflections eingeschaltet. Auf den ersten Blick sieht das Ergebnis passabel aus, aber in der Kugel spiegelt sich die Umgebung, wo eigentlich die Innenwände der Box sein sollten. Auf dem spiegelnden Boden fehlt das Blau der Innenseite der Decke der Cornell Box.



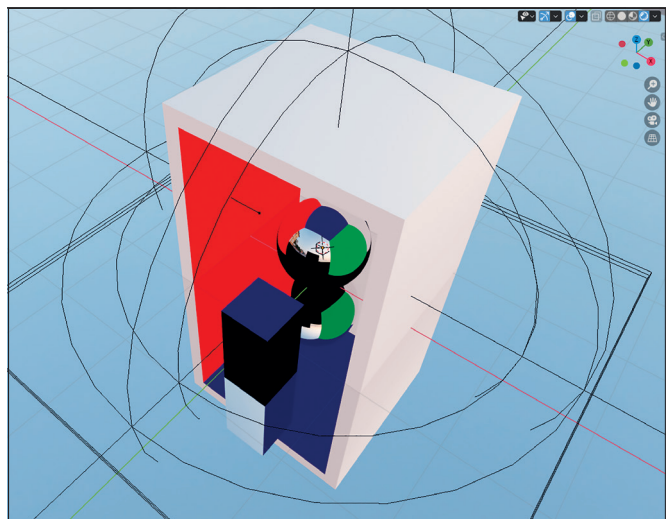
Reflection Plane – Zur Unterstützung der Screen Space Reflections wurde eine Reflection Plane hinzugefügt. Damit spiegelt sich endlich die blaue Decke der Box in der Ebene und damit auch indirekt in der Kugel und dem Quader.



Reflection Plane Distance – Die Distance der Reflection Plane wurde erhöht, um Quader und Kugel komplett einzuschließen. Auf der Oberseite der Kugel erscheint eine Spiegelung der Unterseite der Kugel, da die Reflection Plane von ihrem Ursprung aus rendert. Das gleiche Problem findet sich auf dem Quader, auf dessen Oberseite sich die Kugel so spiegelt, als befände sich die Oberseite auf Höhe der Reflection Plane.



Reflection Plane plus Cubemap – Die Distance der Reflection Plane wurde wieder reduziert und eine Cubemap im Zentrum der Kugel hinzugefügt. Interreflexionen sind mit der Cubemap nicht möglich, daher die schwarzen Bereiche auf der Unterseite der Kugel. Auf der Ebene sind zudem die Reflexionen der Cubemap zu sehen. Das liegt daran, dass Screen Space Reflections in der Szene noch eingeschaltet ist, dann verwendet Eevee die Cubemap statt der Reflection Plane.



Reflection Plane plus Cubemap ohne Screen Space Reflections – Ohne eingeschaltete Screen Space Reflections (SSR) nutzt Eevee für die Ebene die Reflection Plane und für die Kugel und den Quader die Cubemap. Durch das Fehlen von SSR werden die schwarzen Bereiche, bei denen eine Mehrfachspiegelung vorliegt, noch deutlicher.

nahe kommen, nur eben in Echtzeit. Daher ist es auch kein Wunder, dass Eevee intern im Material Preview Mode von Blender 2.8x zum Einsatz kommt, selbst wenn die Render Engine auf Cycles eingestellt ist. (Pro Tipp: Die für Eevee getätigten Einstellungen bleiben dann erhalten, sprich man kann das Material Preview konfigurieren, indem man die Render Engine auf Eevee umschaltet und die Einstellungen ändert.) Besonders erfreulich ist, dass Blender seit Version 2.80 endlich mit einer Auswahl an HDRIs kommt, die man für die Materialvorschau verwenden kann. Man kann also das Aussehen der Materialien in einer realistischen Lichtsituation evaluieren.

Teamplayer

Eevee und Cycles bilden ein richtig gutes Team. Zuerst arbeitet man mit schnellem Feedback in Eevee, bis es darum geht, dem Projekt den letzten Schliff zu verpassen oder das letzte Quäntchen Realismus herauszuholen. An diesem Punkt kann man getrost zu Cycles wechseln.

Wobei Eevee natürlich nicht auf die reine Vorschau beschränkt ist, sondern es durchaus Szenen gibt, die für Eevee prädestiniert sind, wie Motion Graphics, Cartoons oder weitläufige Szenen mit viel diffusem Anteil. Das indirekte Licht kann dabei in Cycles gebacken werden, um den Grad an Realismus nochmals zu erhöhen.

Est. 2014

Ursprünglich hatte der Entwickler Clément Foucault den „PBR Viewport Branch“ ge-

startet, dessen Ziel es war, im Blender Node Editor Shader so bearbeiten und darstellen zu können wie im Editor der Unreal Engine. Das Projekt änderte dann aber seinen Fokus in Richtung Annäherung der Darstellung und der Shader an Cycles, womit es die Aufmerksamkeit der Blender Foundation auf sich

zog. Zwischenzeitlich wurde Clément dank des Blender Development Funds (siehe **DP 06:19**) bei der Foundation angestellt, und es wurden Pläne geschmiedet, wie die neue Engine in Blender integriert werden könnte. Herausgekommen ist Eevee, und die Entwicklung ist noch lange nicht zu Ende.

The person behind Eevee

Clément Foucault is the main developer of the Eevee render engine in Blender.

DP: What's your background and how did you get into Blender development?

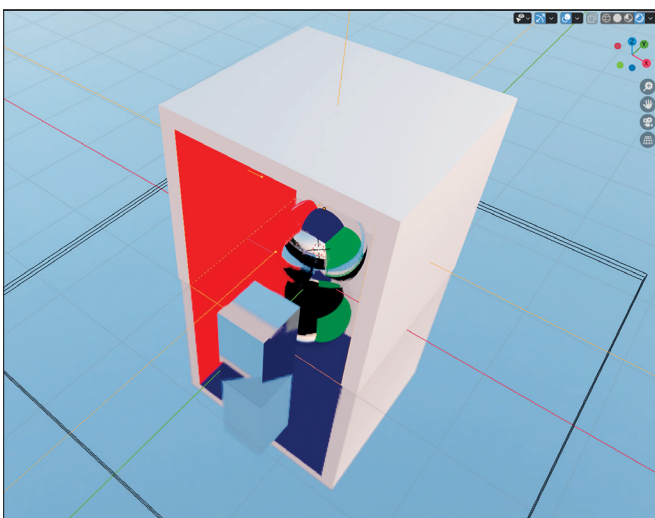
Clément Foucault: I have a degree in 3D animation filmmaking, a pretty generic degree that covers many areas of the animation process. I was already more leaning towards the technical side of production and Blender was already my tool of choice. I then became a freelance 3D artist and did a few gigs while developing the PBR branch in my free time. This project caught the attention of the Blender Institute and they offered me to be part of the 2.8 team and integrate my work in this huge release.

DP: What was the initial spark for your decision to start developing Eevee? Did you use other render engines as reference?

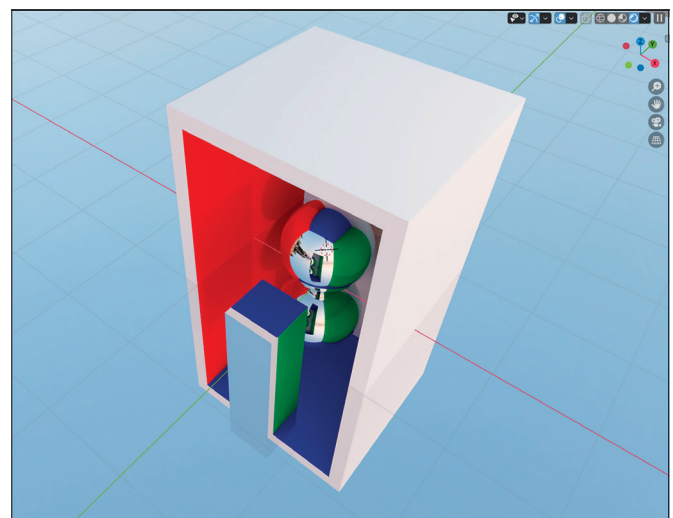
Clément Foucault: The PBR branch is the prequel to Eevee. I had this idea of integrating new realtime rendering techniques during a project where I used Marmoset Toolbag 2 to preview the final look of the assets I was making. Jumping back and forth from Marmoset to Blender was really a pain. I already had done a really low quality PBR for Unreal Engine 3, so one day I decided to give it a go and implement it directly into blender. My main reference during development was Cycles as it was the ground truth.

DP: Did the open movie "Spring" help shape the way we work with Eevee nowadays?

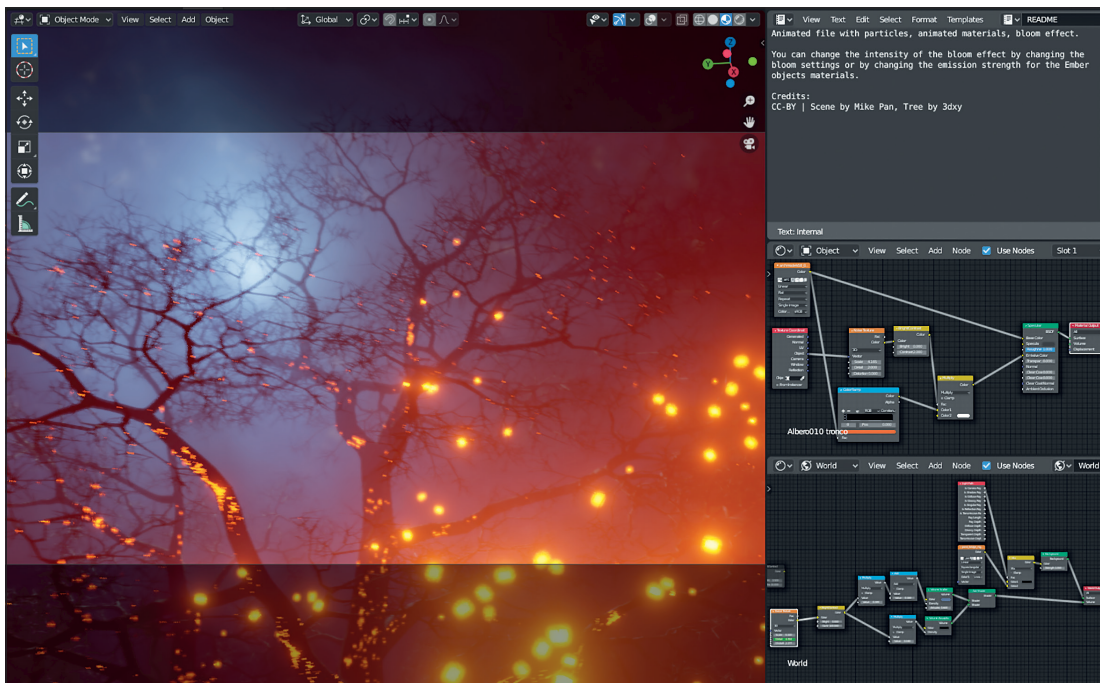
Clément Foucault: "Spring" had to deal with the whole 2.8 instability. It was not meant to directly benefit from Eevee. But the team did use it for texture painting and gave precious feedback about performance and bugs.



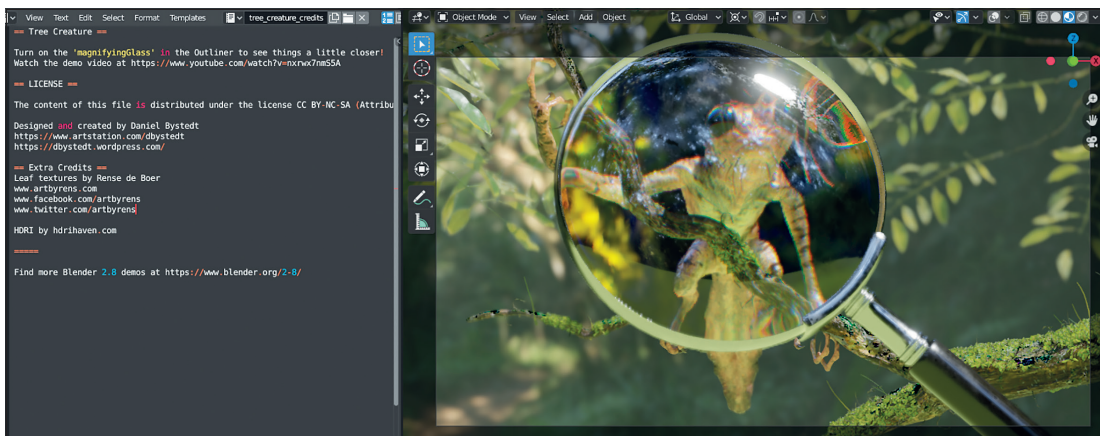
Reflection Plane plus angepasste, kubische Cubemap mit Screen Space Reflections – Die Szene mit eingeschalteten Screen Space Reflections und einer an die Box angepassten, würfelförmigen Cubemap-Probe. Lediglich die spiegelnde Unterseite der Kugel scheint noch ein Problem darzustellen.



Cycles Reflections Ground Truth – Die Szene mit Cycles gerendert. Nicht einmal mehrfache Reflexionen stellen hier ein Problem dar. Auffällig ist der grüne Streifen am Zylinder, der die rechte Innenseite der Cornell Box spiegelt. Diese Spiegelung konnte von keinem der Versuche in Eevee eingefangen werden.



Amber Forest von Mike Pan kombiniert volumetrische Effekte mit einem starken Bloom. Der Baum stammt von 3dxy.



Tree Creature – Ebenfalls von Daniel Bystedt stammt diese Demo, die die Stärken von Eevee voll ausspielt. Es ist sogar ein Vergrößerungsglas in der Szene vorhanden, damit man die Kreatur in Echtzeit näher in Augenschein nehmen kann.

Eevee improved – Was sich seit 2.80 so alles getan hat

Die Schatten wurden zum Beispiel in Blender 2.81 vollständig überarbeitet und lassen sich jetzt einfacher bedienen bei gleichzeitig höherer Qualität und besserer Performance. Auch Kontaktschatten brauchen weniger Tweaking, da sie automatisch die Form der Lichtquelle in die Berechnung einfließen lassen. Lichtquellen funktionieren jetzt auch als Instanzen (z.B. über ein Partikelsystem), und bei Sonnen wird die Clip Distance automatisch berechnet.

Eevee unterstützt jetzt auch den Holdout Shader. Dieser stanz quasi ein transparentes Loch in das Bild und ist mit einer Schablone vergleichbar, ein essenzielles Feature für Compositing. All diese Änderungen zielen darauf ab, die Arbeit mit Eevee zu vereinfachen. Was den Umgang von Eevee mit Transparenz angeht, wurde die Kompatibilität mit Cycles verbessert. Es lassen sich jetzt in beiden Render Engines die gleichen Shader-Kombinationen einsetzen.

Bump Mapping in Eevee wurde ebenfalls angepasst, um genauere Resultate zu erzie-

len, die näher an dem liegen, was Cycles ausgibt. Vor allem mit prozeduralen Texturen funktioniert es jetzt wesentlich besser als zuvor auf Kosten der Performance.

Dual Use

Sowohl in Eevee als auch in Cycles lassen sich eine Reihe neuer und verbesserter Nodes verwenden. Der Vertex Color Node erlaubt einen einfachen Zugriff auf Vertex-Farben und deren Alphakanal. Über den Volume Info Node können analog die Farbe, Dichte von Rauch sowie die Flamme und deren Temperatur bei einer Feuersimulation ausgelesen werden. Bisher war für all das ein Attribute Node nötig, wo man den Namen der Daten, die man wollte, eingeben musste, wie z.B. "density" oder den Namen des gewünschten Vertex Color Layers. Der Object Info Node kann jetzt die dem Objekt zugewiesene Farbe ausgeben.

Noise, Musgrave und Voronoi haben eine vierte Dimension erhalten, wodurch man ihr Aussehen animieren kann. Wenn man auf 4D umschaltet, erscheint dafür ein neuer Slider "W", der sich analog zum Evolution-

Slider in anderen Programmen verhält. Die Voronoi-Textur unterstützt zudem eine Reihe von neuen Mustern.

Wer bisher Math Nodes zum Clamping verwendet hat, kann sich jetzt über einen dedizierten Node freuen, und Map Range erlaubt die Transformation von Wertebereichen. Die Math und Vector Math Nodes zeigen nur noch dann Eingangs-Sockets, wenn diese für die ausgewählte Operation auch genutzt werden können, und es wurden zusätzliche Operationen hinzugefügt. Der Mapping Node hat jetzt Eingänge für Location, Rotation und Scale, die mit anderen Nodes verbunden werden können.

Fazit

Eevee hat Blender einen ganzen Sprung nach vorne gebracht. Die enge Verzahnung mit den Workflows und die Nähe zu Cycles machen Eevee nicht nur zu einem Ersatz für den nicht mehr vorhandenen Blender Internal Renderer sondern auch zu einem integralen Werkzeug bei der täglichen Arbeit mit Blender. Nur Wunder sollte man noch keine erwarten. >ei