

2020

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

Publiziert von Pixeltown GmbH

2

DIGITAL  
PRODUCTION

Deutschland € 17,90

Österreich € 19,-

Schweiz sfr 23,-

# DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPR

MÄRZ | APRIL 02:2020



## Projekte

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

## Hardware

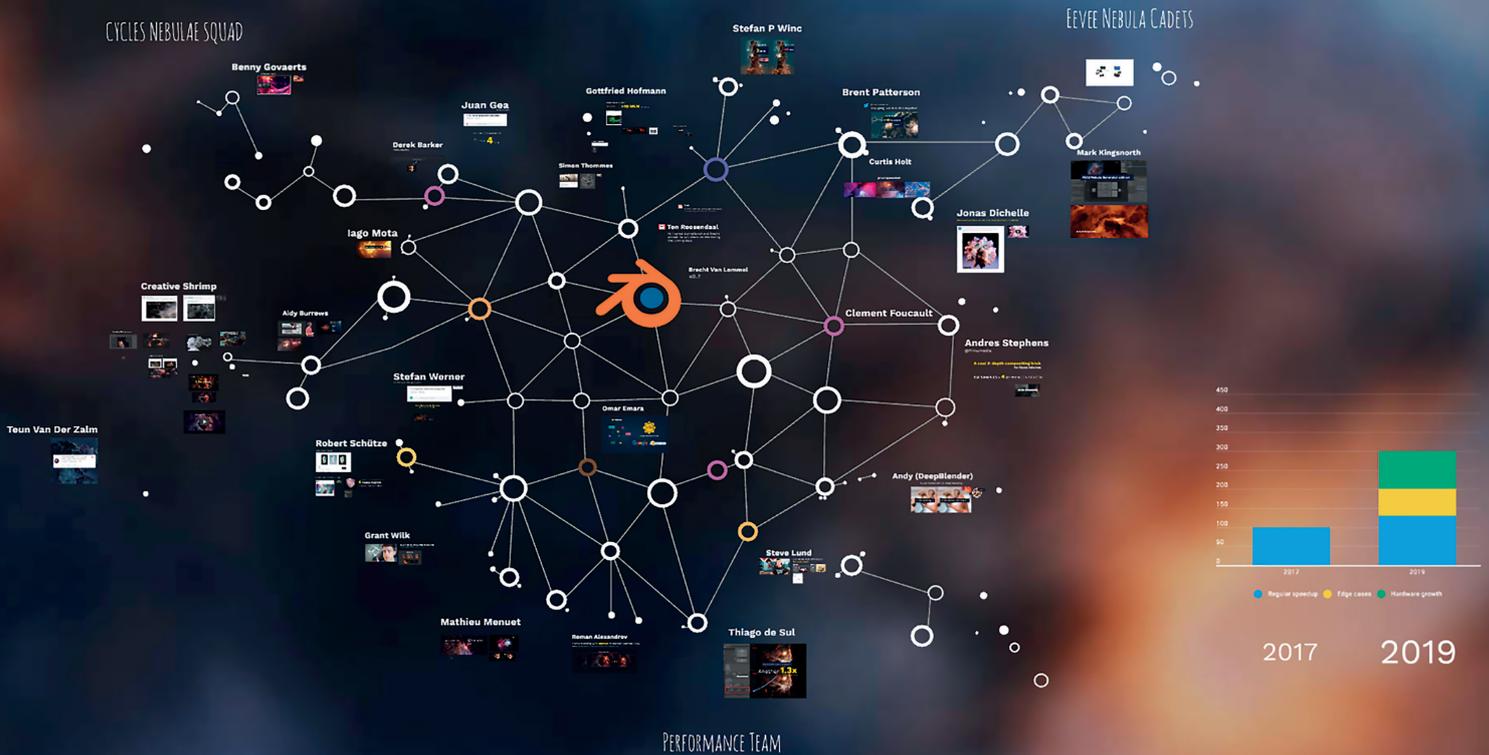
MacBook Pro, Live-Grading,  
BMD Keyboard, Sigma fp

## und vieles mehr

Flame, ZBrush 2020, Quad  
Remesher, Substance Tools



# #b3d community



Eine Karte des Teils der Blender Community, der direkt oder indirekt an dem Nebelprojekt beteiligt war.

## The Power of the Blender Community

Gleb Alexandrov ist in der Blender Community kein unbeschriebenes Blatt. Der YouTube-Kanal seiner Tutorial-Seite Creative Shrimp hat mehr als 100.000 Abonnenten, und auch auf Facebook und Twitter ist er präsent. Irgendwann stellte er auf Twitter die Frage, ob jemand in Blender den „Krakatoa Look“ für Weltraumnebel erreicht hätte. **von Gottfried Hofmann**

Ich war der Meinung, dass ich die Antwort darauf hätte, da ich selbst schon Experimente in die Richtung unternommen hatte und mit Point-Density-Texturen in Cycles das meiner Meinung nach perfekt dafür geeignete Werkzeug parat hätte. Ich stellte also den Kontakt her, zeigte ihm meine Experimente, er zeigte mir seine, und nach einigem Hin und Her hatten wir ein passables Ergebnis und nebenbei noch ein Skript, das in Blender einen Bullet-Time-Effekt mit Bewegungsunschärfe ermöglicht. Aber

es ging noch viel mehr vor sich. Mehr und mehr Mitglieder aus der Blender Community begannen, selbst Weltraumnebel zu erstellen, BLEND-Dateien auszutauschen und die Latte Stück für Stück höher zu legen.

### Blender Conference 2019

Ein paar Monate später hatte ich nicht nur Gleb, sondern auch diversen anderen Nutzern via Twitter bei Nebelschwierigkeiten ausgeholfen und sogar zwei kleine Patches für Blender geschrieben, von denen einer die Limitierung der maximal möglichen Partikel aufhebt und der andere ein spezifisches Problem des Nebelrenderns mit Eevee löst. Das ganze Ausmaß der Nebelmanie wurde mir aber erst bewusst, als Gleb auf der Blender Conference (und kurz darauf auf dem animago) in einer Präsentation darlegte, was in den Monaten davor alles passiert war. Und zwar eine ganze Menge.

### Go Procedural

Gleb hatte beschlossen, dass eine Lösung, die nur auf prozedurale Texturen setzt und

damit den Aufwand einer Simulation umgeht, der bessere Ansatz wäre. Denn Simulationen und ihre Caches sind in Blender immer noch ein Garant für Instabilität. Da boten sich das von Simon Thommes erstellte Procedural Noise Pack an. Es ist als CCO über Gumroad erhältlich (<https://gumroad.com/1/NOISE-P>) und erweitert Eevee und Cycles um eine Reihe prozeduraler Texturen. Diese erweisen sich als sehr nützlich für die Erzeugung von Nebulae, die sich direkt rendern lassen, ohne vorher eine Simulation laufen zu lassen.

### Small and Farmable

Dadurch wurde auch die Auswahl an Renderfarmen größer, denn viele Renderfarmen können keine Blender-Simulationen backen und erwarten den gesamten Cache als Upload. Gerade bei Community-Renderfarmen wie Sheep it wird das aber zum Problem, da damit zu jedem einzelnen Rechner Gigabytes an Daten verschickt werden müssten. Daher wurde dort z.B. die maximale Projektgröße auf 500 Mbyte beschränkt, viel zu wenig für eine detaillierte Simulation.



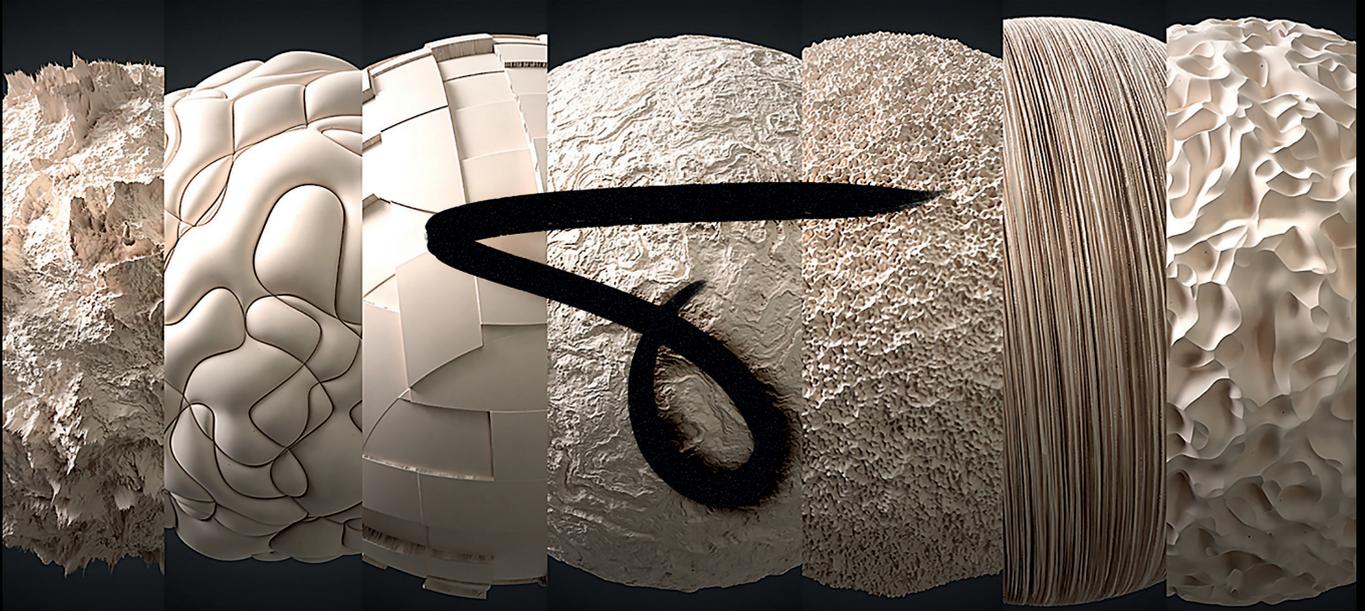
**Nebel von Simon Thommes – Dieses Nebelrendering wurde von Simon Thommes erstellt. Von Simon stammen auch die zusätzlichen prozeduralen Texturen, die im Artikel erwähnt werden.**



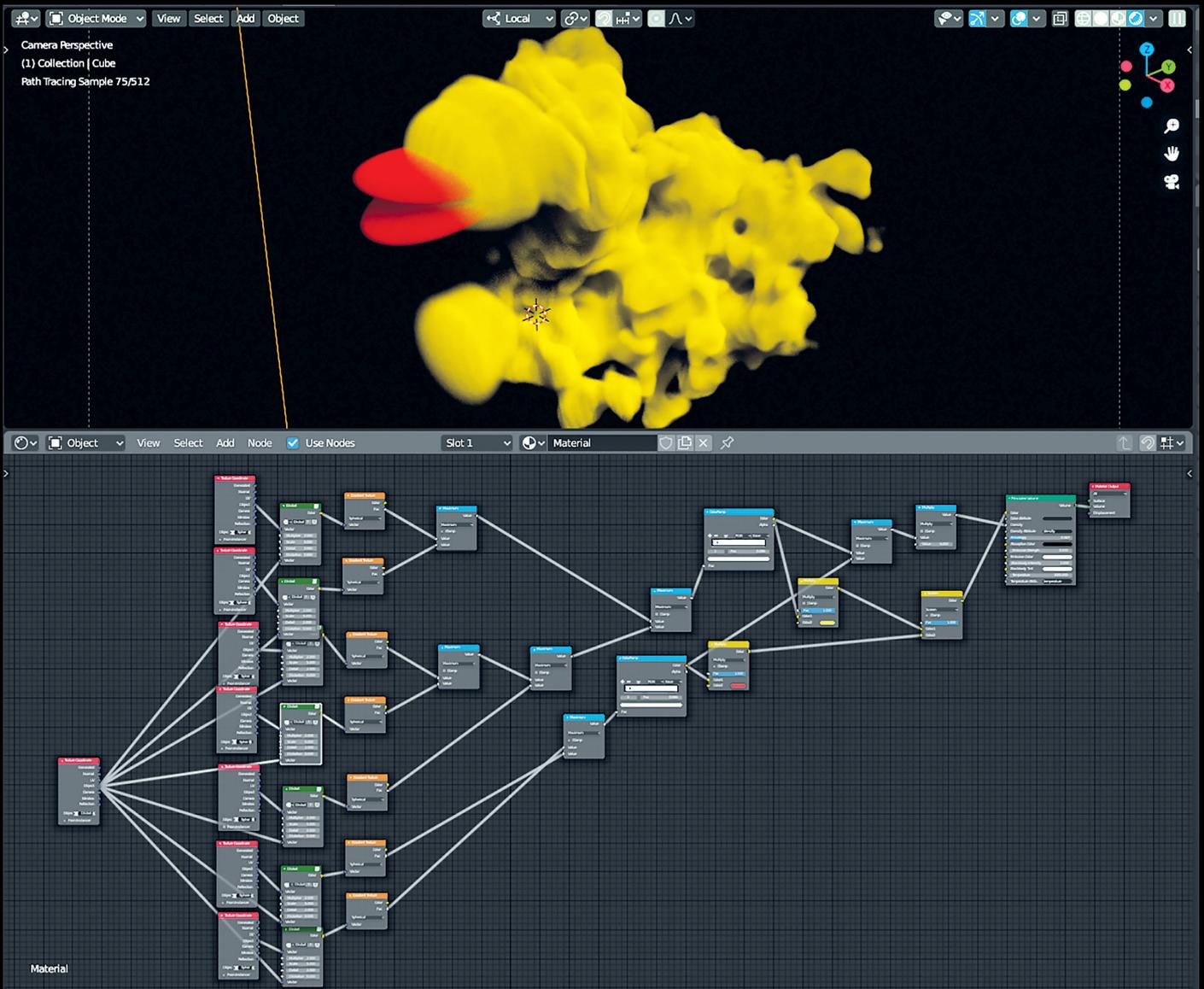
**Nebel von Gottfried Hofmann – Ich habe auch ein Beispiel beigetragen. Der Nebel basiert auf verzerrten Point-Density-Texturen, die von einem Partikelsystem gespeist werden, das wiederum von einer Rauchsimulation verwirbelt wurde.**



**Splash Nebula – Der Nebel-Render von Gleb Alexandrov, der die Basis für den Splash Screen der Beta-Version von Blender 2.82 bildet.**



Rauschen von Simon Thommes – Beispiele für die Möglichkeiten des Procedural Noise Packs von Simon Thommes. Die Node-Gruppen sind als CC0 lizenziert und über Gumroad erhältlich (<https://gumroad.com/l/NOISE-P>).



Duck Distortion – Frühe R&D-Datei, die aufzeigt, dass prozedurale Verzerrung der UVW-Koordinaten eines volumetrischen Objekts in Blender möglich ist. Lediglich mit der Rauchsimulation ist es noch nicht möglich.



**Gasriese – Die prozeduralen Nebel lassen sich auch für Renderings von Planeten einsetzen.**

Mit rein prozeduralen Texturen hingegen blieb die Datenmenge im Rahmen, und damit war der Weg frei, die Community-Renderfarm Sheep it einzusetzen. Diese funktioniert prinzipiell so, dass man seinen eigenen Computer zum Rendern zur Verfügung stellen kann. Für jeden gerenderten Job erhält man Punkte. Diese kann man einsetzen, um selbst auf der Farm rendern zu können, was Sheep it zu einem typischen Communityprojekt macht.

## Fraktale

Auch Fraktale wurden ausprobiert, um kosmische Nebel darzustellen. Diese fantastische Form der Mathematik hat Kinogänger aus aller Welt schon in Filmen aus dem Marvel Cinematic Universe wie „Ant-Man“, „Guardians of the Galaxy“ und „Dr. Strange“ bezaubert. Hier halfen Arbeiten von Jonas Dichelle und Robert Schütze weiter, die zeigten, dass sich Fraktale sogar in Eevee halbwegs performant berechnen lassen. Allerdings waren sie nicht wirklich art-directable. Also zurück zu herkömmlichen prozeduralen Texturen.

## Eevee

Die meisten Beiträge aus der Community kamen definitiv beim Thema Eevee. So viele, dass es müßig wäre, die Namen aller Beteiligten aufzuschreiben. Wichtige Impulse kamen von Brent Patterson, der wahrscheinlich die meisten Tests erstellt hat. Andreas Stephens hat eine Möglichkeit entwickelt, mehrere Schichten von volumetrischen Objekten im Compositor zusammenzufügen, um in Eevee immer nur einen Ausschnitt zu rendern. Das klingt jetzt aufwendig, ist aber eine Möglichkeit, die Beschränkungen von Eevee bei der Qualität zu umgehen. Mark

Kingsnorth hat sogar einen Generator für Nebel in Eevee als Add-on entwickelt.

## Cycles optimiert

Die Ergebnisse in Eevee konnten aber allesamt nicht mit der Qualität von Cycles mithalten. Letztere Engine hatte aber das Problem, dass sie extrem lange zum Berechnen der Weltraumnebel brauchte. Volumetrisches Rendering ist noch nicht die Stärke von Cycles. Daher begann nun die Optimierungsphase, wobei auch wieder Unterstützung aus der Community kam. Neben zahlreichen Tipps zum allgemeinen Beschleunigen des Renderprozesses und Tests von noch nicht ganz fertigen Denoising-Implementierungen teilte Theory Studios ihren Blender-Build, der einige Optimierungen rund um volumetrisches Rendering mit Cycles enthält. Die Entwickler von E-Cycles, einer kommerziellen Version der Cycles Render Engine mit zahlreichen Optimierungen, fügten sogar eine eigene Nebula-Checkbox hinzu, um die Berechnung von volumetrischen Nebeln in E-Cycles nochmals zu beschleunigen, was je nach Szene bis zu 16-mal schnelleres Rendering erlaubte.

## Optical Flow

Um die Renderzeiten weiter reduzieren zu können, wurde das Seitenverhältnis der Renderings angepasst. Das Cinemascope-Format mit 2,35:1 rendert ca. ein Drittel schneller als 16:9. Aber auch das war noch nicht genug. Um nicht jeden einzelnen Frame rendern zu müssen, erstellte Roman Alexandrov, der Bruder von Gleb, ein Tool in OpenCV, das mittels Optical Flow Zwischenbilder generierte, womit nur noch jeder zweite oder dritte Frame gerendert werden musste.

## Lehrstück

Die Reise von Gleb durch die Blender Community auf der Suche nach dem perfekten Weltraumnebel kann man als Lehrstück dafür sehen, wie man nicht nur Hilfe bei den eigenen Projekten bekommen, sondern damit sogar zur Entwicklung von Blender beitragen kann. Sicherlich wird nicht jeder bei jedem Problem gleich eine eigens angepasste Render Engine erhalten. Wenn man aber offen an die Community herantritt und keine Angst hat, neben Renderings auch die BLEND-Dateien und sonstige Quellen zu teilen, kann man durchaus etwas zurückbekommen. Mehr dazu im Interview.



**Gleb Alexandrov is a 3D artist, tutorial maker of Creative Shrimp and a coffee maniac. <https://creativeshrimp.com> <http://glebalexandrov.com>**

**DP:** In "Space VFX Elements" you had to cheat nebulae by using planes and billboards. Then you decided that this had to change and turned towards the community to find the best solution to improve how nebulae can be created in Blender. That started an avalanche.



**Beleuchtung – Die Ausleuchtung ist bei Weltraumnebeln ein entscheidender Faktor.**



**Beleuchtung – Grundsätzlich ist es so, dass beim Rendern von Nebeln in Cycles die Stärke des Rauschens eng an die Anzahl der Lichtquellen gekoppelt ist. Grundsätzlich gilt dabei: je weniger, desto besser. Bei diesem Bild konnte die Reduktion von 9 auf 2 Lichtquellen das Rauschen um den Faktor 4,5 verbessern.**



**Gleb Alexandrov:** We've been wanting to play with the same toys as other 3D folks who use 3ds Max with Krakatoa render engine, or Houdini and so on, creating jaw-dropping animations. Volumetric nebulae by Teun Van der Zalm for example. It was painful to stick to 2D planes after you've seen something like that. Real deal.

So it was a no-brainer to explore this direction, real volumetric stuff, massive particle simulations. That being said, nothing like that has been made in Blender, if you browse through the Youtube "3D nebulae" videos, Artstation posts, you won't find anything like that (with some rare exceptions like the art of Stefan Payne-Wardeenaar on #b3d Twitter. That's pretty close to a high-budget look we were aiming for (<https://twitter.com/StefanPWinc>).

So yeah, we started to talk to people who may have had some insights into volume rendering in Cycles, particle systems, point density shader and other things that we thought could be used to recreate the Krakatoa nebulae look. It was such an interesting ping-pong, we learned a lot over a short timeframe. In parallel, we were testing our own stuff – and also trying to reverse engineer the techniques of Teun Van der Zalm (hit a brick wall). Bit by bit, with the help of the Blender community, we were making progress. When we were in the middle of preparing the talk for Blender Conference 2019, it became clear that we were able to render something that is miles ahead of our previous attempts. By the way, the title of the talk was "3D Nebulae and the Power of the Blender Community", it's like self-fulfilling prophecy, you know.

**DP: Did you expect that amount of help and feedback from the community?**

**Gleb Alexandrov:** No way we could expect such feedback and contribution from the community. It seemed almost like the universe was aligning itself towards our vision. Or maybe we just picked the right wave at the right moment. Because many things have happened since we released the original Space VFX course (with 2D planes for nebulae).

Not only the code advancements, although it's obviously important, but also

procedural techniques shared on Twitter by various #b3d members, some interesting developments like rendering the Mandelbulb fractal in Eevee or experimenting with distorting UV coordinates of the 3D noise textures.

It's incredible how even a small slice of the communication happening within the CGI community, based around open-source software in this case, reveals that so much is going on. So many people are working on the similar things at the same time, without even knowing each other. And once you start connecting the dots, it's just incredible how much we can achieve in this smart-crowd manner. The open-source way of research & development, so to speak.

Well, connecting the dots is not always easy, you have to know what you're searching for, with whom to talk and so on, and you should be willing to share your files, ideas, work-in-progress images and more. The name of the game is transparency here. We were perfectly comfortable to share our nebulae BLENDS (and get even better BLENDS in return).

**DP: Your engagement helped Blender become better at volume rendering and is an example that non-coders can also make an impact in the development of Blender. How did that come about?**

**Gleb Alexandrov:** I think that most of innovation happens precisely in the non-coder slice of the community. What people do with software and what they want from software, these are two main things which drive the development of open-source software. Even if it's an implicit demand which manifests itself in the form of videos, tweets, posts and so on. I bet that if you post a nebula animation in the style of the "Star Trek Into Darkness" end credits, you'll see ripples all across the community, inspiring new cosmic endeavours and also, in turn, shifting priorities in Blender development. Someone will write a patch to speed up particles or add 5D noise or whatever, or simply take your BLEND and take it to the next level.

Even our not-so-great w.i.p. animations sparked a lot of interest and lead to some innovation. All thanks to people who are

willing to collaborate and participate. Donate a bit of their computers' processing power on Sheep it to help render or give access to the university computers for a few hours, or share a custom solution for rendering fractals – or writing a script for freezing time.

**DP: What's your favourite channel to ask for help?**

**Gleb Alexandrov:** In a broad sense, googling stuff helps. The go-to channel is Twitter, I think. Maybe I'm not aware of other platforms, but for us Twitter is the best in terms of asking #b3d for help and sharing our progress. It's a world-wide cocktail party, as Gary Vaynerchuk put it. You can poke people, spark conversations in such an effortless way that it makes Twitter priceless.

**DP: What's your advice for someone who is looking for help from the Blender community and wants to get the same level of feedback you did?**

**Gleb Alexandrov:** Start by generating some content that is cool and share it. Make the first step e.g. give something, then ask for help. Share some tip that is useful. It's all about collaboration, so be willing to share. Non zero sum game means everyone wins so there's no point in holding back.

**DP: It seems like the splash for Blender 2.82 is a nebula ...**

**Gleb Alexandrov:** Haha, it is nebula indeed! At least for the beta version. Pablo Vazquez thought that it would be cool to have something from outer space this time. Thanks so much, Pablo! Hopefully we'll keep pushing frontiers together. >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.Blender-Diplom.com](http://www.Blender-Diplom.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.