

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

SEPTEMBER | OKTOBER 05:2017



Pipeline

Houdini, Fabric Engine und Scripting in neuem Licht

Förderung

Filmidee? Check! Auf in die Förderung – wir zeigen wie.

Tools & Tricks!

C4D R19, Arnold, Resolve 14, ZBrush 4 R8 & mehr



Hinter den Kulissen der CityEngine

Einer der Vorteile der FMX: Man trifft die Leute, die hinter den Kulissen die Software aufbauen – und anlässlich seines Vortrags haben wir uns mit Simon Hägler unterhalten, einem der Entwickler der CityEngine. von Bela Beier

Gerade eben wurde CityEngine 2017.0 veröffentlicht und mit ihr neue Features. Grund genug, bei den Machern nachzufragen, was uns so erwartet und wo es hingeht. Gefragt haben wir bei Simon Hägler, Software-Developer bei Esri R&D Zurich mit Schwerpunkt prozedurales Modeling – und seit Procedural-Inc.-Zeiten mit im CityEngine-Team.

DP: Was wird deiner Meinung nach die User an der nächsten Version von CityEngine am meisten freuen?

Simon Hägler: Ich denke, am meisten freut es, dass wir das UI vereinfacht und aufgeräumt haben – flacher, weniger Clutter und visuell moderner. Man glaubt nicht, was Interface-Verbesserungen im Alltag an Effizienz bedeuten. Zusätzlich unterstützt CityEngine nun auch Szenarien, mit denen man verschiedene Varianten der gleichen Szene bequem verwalten kann. Zusammen mit den modernisierten Dashboards ist das ein wertvolles Tool für Stadtplaner.

Die spannendste Neuerung sind für mich die „Local Edits“. Das bedeutet, man kann interaktiv im Viewport an einzelnen prozeduralen Objekten visuell arbeiten. Wir haben nun also eine Lösung für das altbekannte Problem, dass prozedurale Modelle nicht direkt editierbar sind.

Ein Beispiel: Wir generieren prozedural – wie gehabt – eine Fensterhöhe und können jetzt per Mausklick einzelne Fenster anpassen. Wir haben dazu das Feature „Handles“ aus der Vorversion aufgebohrt. Jetzt kann man einfacher in Details eingreifen und alles



modellieren, wie es einem gefällt.

DP: Wie weit kann ein Artist eingreifen?

Simon Hägler: So weit wie man will. Je nach Pipeline – die Handles können in den prozeduralen Regeln für beliebige Bemaßungen und Parameter freigegeben werden, zum Beispiel vom TD – ist nun jeder Parameter für sich freischaltbar und vom Artist mit einem Klick bearbeitungsfertig. Je nach Projekt und Szenengröße macht es nicht immer Sinn, alles lokal editierbar zu machen, das muss man selber entscheiden. Theoretisch kann man alles mit lokalen Edits versehen. Gerade bei Architekturvisualisierungen sind individuelle Eingriffsmöglichkeiten schon sehr hilfreich.

DP: Man könnte sich also auch komplett vom Prozeduralen verabschieden?

Simon Hägler: Das hängt vom Projekt selber ab, aber ich denke, wenn man bei 50.000 Gebäuden an jedem Local Edits einbaut, hat man die CityEngine schon ein bisschen rückwärts bedient (lacht). Wenn wir das bei unseren Usern öfters sehen, müssen wir natürlich überlegen, wie wir es intelligenter machen können.

Auf der anderen Seite ist es durchaus normal, dass Leute nicht-prozedural mit der CityEngine arbeiten – im Sketch-Up-Style,

sozusagen. Dazu kommt, dass viele User die CityEngine verwenden, um Satellitenbilder zu tracen, um die Umrisse und Größen von Gebäuden nachzuzeichnen.

DP: In der neuen Version sind jetzt ja auch kontextsensitive Parameter. Wie hat man sich das vorzustellen?

Simon Hägler: Diese Parameter sind enorm mächtig. Wir haben in CGA – unserer Architektur-Programmiersprache – weitere Operationen eingebaut. Ein neues Feature sind z.B. „Labels“ die man jeder Komponente eines Gebäudes vergeben kann. Damit lassen sich dann in CGA Abfragen formulieren, sodass neue Gebäudeteile nur noch bedingt erzeugt werden.

Praktisch sieht das zum Beispiel so aus: Als Balkon eines Gebäudes frage ich ab, ob ich einen Teil des gegenüberliegenden Gebäudes berühre, das mit dem Label „Wand“ versehen ist. Wenn ich „Wand“ berühre, entsteht kein Balkon. Wenn ich ein Objekt mit dem Label „Wäscheleine“ berühre, kann ich den Balkon bauen. Sozusagen ein selektiver Test auf Überlappung und Berührung, jeweils innerhalb und außerhalb des Gebäudes.

Dazu kommt dann noch die Schwester-Operation „Minimum Distance“, die bei Bedarf auch ohne Namen arbeitet und abfragt, wie weit es denn – um im Beispiel zu bleiben – bis zur Wand ist. Das unterstützt auch Teilbereiche: Bei 10 Metern Distanz gibt es einen Balkon, 5 bis 10 Meter gibt es einen kleineren Austritt, und unter 5 Metern bleibt es eine glatte Wand.

Versionierung 2.0

„Scenarios“ sind ein neues Feature, welches verschiedene Versionen – also multiple Designs – innerhalb einer Szene erlaubt. Diese lassen sich side-by-side vergleichen und editieren – und die neuen Dashboards können bereits damit umgehen. Somit können verschiedene Varianten direkt getestet werden und auf die Pipeline-Bereitschaft geprüft beziehungsweise zur Abnahme bereit gemacht werden. Bis zu 64 Varianten sind möglich und sollten auch den unentschlossensten Art Director zufriedenstellen. Zwischen allen Szenarien können Objekte per Rechtsklick verschoben oder kopiert werden – auch als Einzelkämpfer hat man so die Möglichkeit, große Szenen übersichtlich zu gestalten und erst zuletzt final zusammzusetzen. Die Szenarien sind auch kompatibel mit dem 3VR-Exporter.

Local Edits

Wie im Interview angesprochen schlägt die neue CE-Version die Brücke zwischen prozeduralem und klassischem Modeling – jedes Objekt in jeder Szene kann direkt angefasst werden. Somit wird der Feinschliff – zum Beispiel unterschiedlich große Fenster, ein paar Balkone unterschiedlicher Größe, verschiedene Etagen-Gestaltungen und Ähnliches – direkt in CityEngine angewendet, ohne die prozedurale Grundlage der Szene anzufassen. Und wenn es mit einem durchgegangen ist, gibt es einen Reset für alle lokalen Edits innerhalb einer Szene.



CGA Neighborhood Queries

Die interne Scriptsprache der CityEngine – CGA – hat einige interessante Operatoren verpasst bekommen – die sogenannten „Context Operations“. Objekte werden mit Labels versehen, die dann miteinander interagieren können. Mögliche Operationen sind zum Beispiel Distanzmessungen – wie weit ist dieses Objekt vom nächsten mit dem Label entfernt. Das klingt erstmal trivial, aber wer prozedural Tankstellen in seiner Stadt platzieren will, oder Parks, kann hier schon einiges an Städtebaulogik leisten, und das nur mit der Funktion „minimumDistance0“. Dies ist nur ein Beispiel, viele andere kommen natürlich noch hinzu. Dazu können auch Objekte innerhalb von Boxen, Vertices und vieles weitere so logisch verknüpft werden.

Maß halten

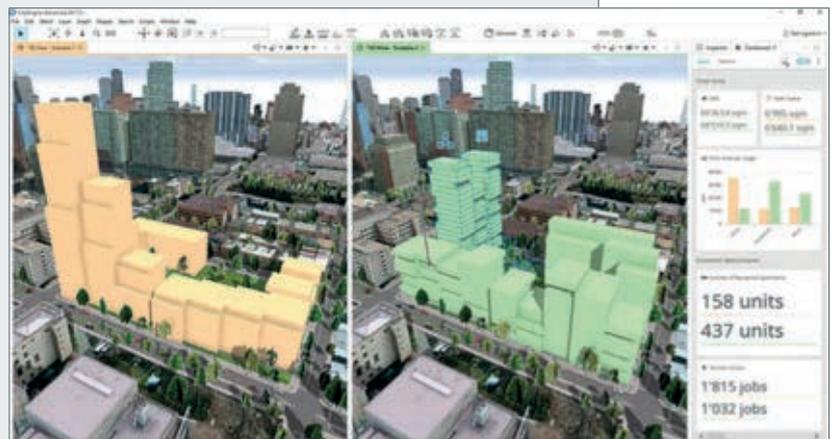
Ganz frisch ist ein digitales Maßband – Distanzen, Höhen, Winkel zwischen Gegenständen sowie Flächen können nun sowohl absolut als auch in Relation zueinander präzise gemessen werden. Dabei werden einfach zwei oder mehr Punkte in der Szene für



die Distanzmessung markiert. Laser-Linien zur Distanzmessung werden eingeblendet, um optisch zu kontrollieren, ob man auch die richtigen Sachen misst. Wenn man hier falsch geklickt hat, sind alle Messpunkte im Nachhinein noch anpassbar.

Neues Interface

CityEngine 2017.0 wurde grafisch komplett überarbeitet – und kann nun auch mit DPI-starken Displays und Fonts bei hohen Auflösungen umgehen – außerdem ist sie optisch aufgeräumter. Dazu kommt ein Dashboard, in dem diverse Charts zur Beschreibung der aktuellen Szene zur Verfügung stehen – auch in verschiedenen Szenarien. So sind Größen, Zeiten, Polygonanzahl und vieles mehr immer im Blick. Und angesichts der breiter werdenden Monitore wird so die gesamte Bildfläche effizienter verwendet.

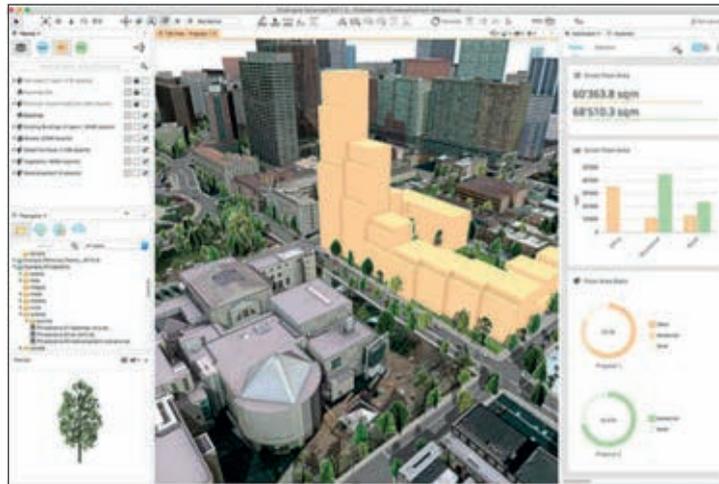


Weitere Features

Soviel zu den wichtigen Neuerungen – des Weiteren sind nun Red Hat Enterprise 7 und CentOS 7 offiziell supportet, das Interface lässt sich in neuen Sprachen darstellen und, was viele freuen wird, die Basic-License unterstützt nun Collada und KML. Dazu kommen Anpassungen beim Viewport und an vielen weiteren Stellen.

Fazit

Die 2017er-Version ist eine definitive Weiterentwicklung – das neue Interface wirkt aufgeräumt und logisch, die Szenarien sind – bei entsprechender Gestaltung des Projekts – für die Render-Vorbereitung ein echter Segen, und Local Edits geben im Alltag den letzten Schliff für das Environment.



DP: Diese Labels lassen sich auch auf das neue Zaun-Tool anwenden und die Vegetation innerhalb und außerhalb der Gärten?

Simon Hägler: Genau. Hier wäre ein Beispiel: Distanz zur Mauer weniger als 2 Meter ergibt maximal Gras, bis 5 Meter Sträucher, und erst weiter weg darf es Bäume geben. Das hat einen zweiten Aspekt, nämlich die Geschwindigkeit. Unsere User wollen – wie jeder Artist – immer schneller arbeiten und mehr Möglichkeiten innerhalb der Library haben.

Das Zaun-Tool war in der Programmierung sehr spannend – fast alles Regelmäßige kann ein Zaun sein, zum Beispiel Pylone und Hütchen auf der Straße oder Maschendrahtzaun, Baumreihen an Felderkanten oder Straßenmarkierungen oder Leitpfosten. Alles ist parametrisch und nur eine Regel. Und zusammen mit den Labels und kontextsensitiven Parametern nimmt das wirklich Schwung auf.



DP: Welche Pipeline-Entwicklung hat dich in der letzten Zeit überrascht?

Simon Hägler: Dass manche alten Formate wiederkommen. Wir hatten in letzter Zeit einige Anfragen zum Thema .stl-Export, speziell für den 3D-Druck. Drucken ist ein spannendes Thema für Architekten, die sich vom klassischen Modellbau verabschieden und direkt auf einem 3D-Drucker ausdrucken – es ist schneller, stabiler und leichter zu bearbeiten. Und mit der Unterteilung des großen Modells in logische Einheiten im Drucker sind Iterationen schnell umgesetzt.

Doch die Standardsprache der 3D-Druckerprogramme ist STL (STL = STereoLithography oder auch Standard Tessellation Language), was eigentlich schon völlig überholt ist – dachten wir zumindest. Es kommt eben alles wieder.

DP: Aber wie halten Architekten Modelle – gerade wenn es größer wird – umsetzbar und ausdrückbar? Gerade das Innenleben ist ja im Modellbau oft wichtig?

Simon Hägler: Das ist prozedural problemlos möglich. CityEngine ist nicht limitiert auf Gebäude-Außen-Volumen – auch Innenräume werden so bestückt, zum Beispiel für Personalplanung und Visualisierung. Wir planen so zum Beispiel auch unsere eigenen Räume.

»Es ist immer am besten, die Daten so spät wie möglich zu generieren.«

Simon Hägler
Software Developer, Esri R&D Zürich

Und technisch repetitive Strukturen lassen sich wunderbar erzeugen – ich hatte vor einiger Zeit mal die Golden Gate Bridge prozedural erzeugt. Im Prinzip alle menschengemachten Objekte lassen sich erschaffen – und wie wir wissen, da fällt vieles drunter.

DP: Wenn man sich mit der neuen Version in CityEngine einarbeiten will, welche Ressourcen kannst du da empfehlen? Und an welchen Stellen im Netz kann man sich mit 3D-Daten eindecken, um seine eigenen Städte zu bauen?

Simon Hägler: Wir haben ein eigenes Forum, das GeoNet (geonet.esri.com), wo sich alles sammelt, was man für den Einstieg braucht. Zusätzlich gibt es auch viele Tutorials und Beispiele mit fertigen Szenen und Regeln.

CityEngine-Regeln weitergeben (und konsumieren) kann man auch via unserer Cloud-Plattform „ArcGIS Online“. Mit einem Klick lassen sich CityEngine-Regeln direkt aus CityEngine als „Rule Packages“ teilen (da sind dann alle referenzierten Assets direkt drin). Das kann aber

u.U. kostenpflichtig sein. Wer kein ArcGIS-Online-Konto hat, kann die Rule Packages auch als Dateien teilen.

DP: Und wenn du jetzt aus deiner Sicht auf die Projekte mit der CityEngine schaust, worauf bist du stolz?

Simon Hägler: Zuerst fällt mir „Zoomania“ ein, wo die Stadt – in CityEngine gemacht – ein so wesentlicher Bestandteil der Welt ist. Dann fällt mir spontan noch Double Negative ein, die einen Renderman-Export aus der CityEngine für „Man of Steel“ benutzt haben. Hauptsächlich weil Renderman-Dateien hier an einer Stelle zum Einsatz kamen, wo sie üblicherweise nicht hingehören – das ist ja eigentlich ein Transfer-Format ganz kurz vor dem Renderer. Aber damals hatten wir ja den Alembic Exporter noch nicht. Als Entwickler ziehe ich vor den Studio Pipeline TDs/Devs immer wieder den Hut.

Abseits vom Projektbeispiel: Als Entwickler finde ich es immer toll, wenn die Leute mit unserer API arbeiten (github.com/esri/esri-cityengine-sdk) – also unsere Geometrie-Generierung in ein anderes Tool einbetten, egal wo, von Houdini bis Blender.

Denn es ist immer am besten, die Daten, mit denen man dann arbeiten muss, so spät wie nur irgend möglich zu generieren und sie erst zu dem Zeitpunkt, wo sie gebraucht werden, in die Tools zu laden. Jeder, der warten muss, bis sich Szenen geöffnet haben, wird mir da zustimmen – zumal man oft dann erstmal damit beschäftigt ist, die sichtbaren Objekte so weit zu minimieren, dass man damit arbeiten kann.

Diese Idee des prozeduralen Arbeitens auf die Spitze getrieben – also, irgendwann die Render-Nodes zu sehen, die erst die City-Geometrie erzeugen, wenn wirklich Pixel gebraucht werden – das wäre schon was. Im Preview würde es bedeuten, einen Raytracer zu haben, der erst reagiert, wenn er auf einen aus der Perspektive sicht-

baren CityEngine-Punkt trifft und keinen Moment früher.

DP: Mit den Vorträgen hier auf der FMX und deinen Gesprächen mit Usern: Welche Tools und Formate sollten wir als Branche im Auge behalten, und wo siehst du der Entwicklung mit Spannung entgegen?

Simon Haegler: Zum einen die USD (Universal Scene Description) von Pixar, eine sehr nette Art, Assets und ihr Set-Dressing zu speichern, das Layouting mit diversen Varianten und LODs und ähnlichem. Da sehe ich als CityEngine-Entwickler, was wir da alles an Daten liefern könnten. Wir haben schon erste Experimente gemacht, setzen aber bis jetzt noch auf Alembic – was sich praktischerweise innerhalb von USD benutzen lässt.

Ein weiteres Thema ist der Export in Game Engines via eines soliden .fbx-Workflows. Wir experimentieren mit Unity und Unreal, mit Blick auf VR – um zum Beispiel die Stadt direkt dort anschauen. Oder Unreal und Unity als Rendering zu benutzen, wie wir hier auf der FMX schon an vielen Beispielen gesehen haben.

Was in Zukunft auch noch kommen wird, sind Analytics – im Workflow gesehen zwischen der PreViz und der klassischen Stadtplanung, wo es um sogenannte Line of Sights und Viewsheds geht.

Hier gibt es ja im Architekturbereich diverse Zonen und Regelungen dafür. Ein Beispiel ist, dass hinter der Tower Bridge in London kein Wolkenkratzer gebaut werden darf, weil ansonsten der Stil der Stadt zerstört wird. Und hier entwickeln wir Tools, um genau zu messen, ob gestört wird oder ob es noch innerhalb der Silhouette stattfindet.

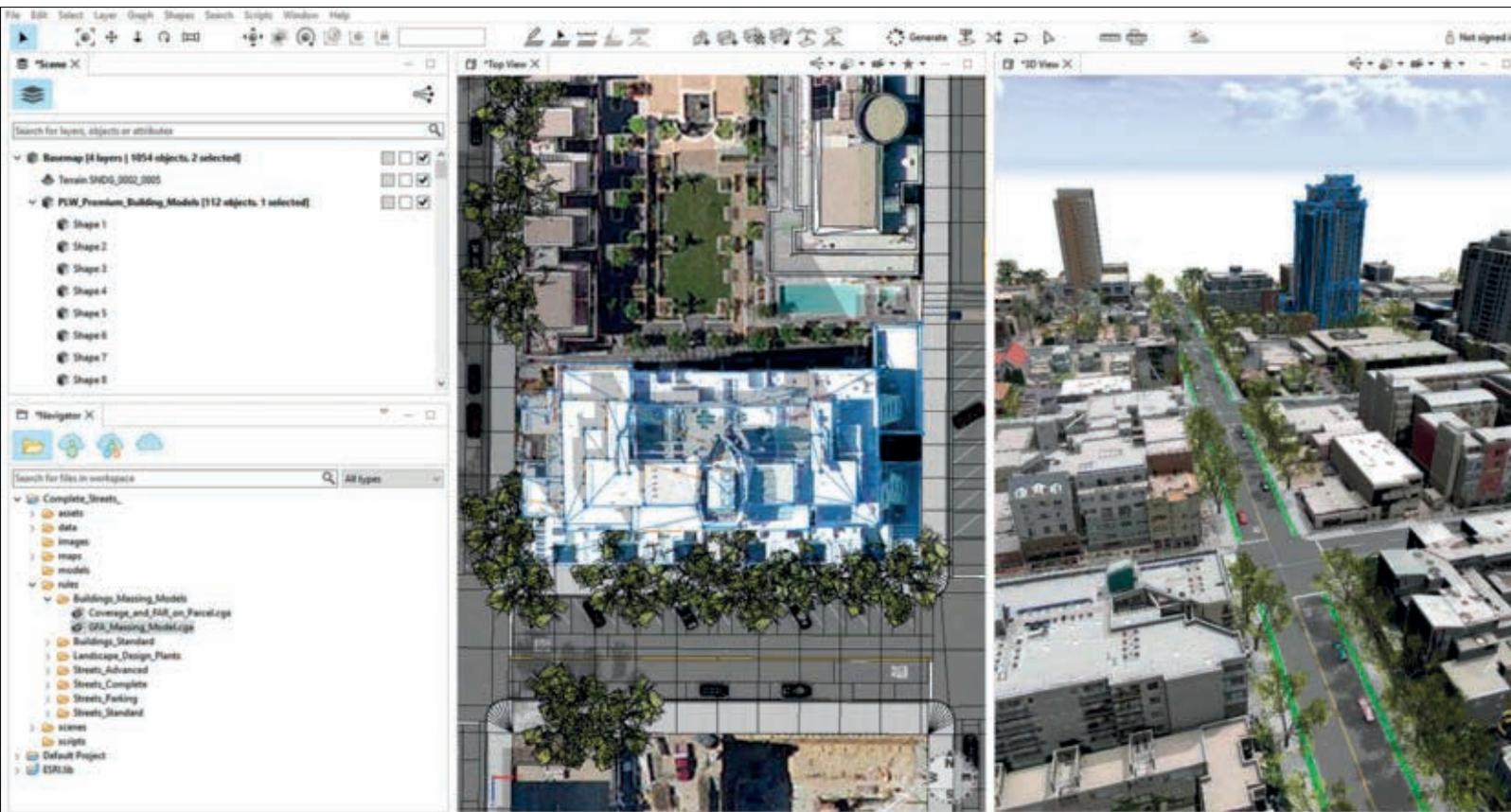
Im ersten Schritt würde man dann – beispielsweise in einer VFX-Produktion, die sich nach Manhattan anfühlen soll – ein Volumen definieren und sobald irgendein angepasstes Gebäude hier auftaucht, wird es rot und die CityEngine weist einen auf den Zonenverstoß hin. Hier wird natürlich der Realismus in den Vordergrund gestellt, nicht unbedingt die Frage: „Verletzte ich gerade irgendeine Stadtplanungsregel?“

Damit knüpfen wir an die allgemeine Produktpalette von Esri an, wo diverse Vorschriften und Regeln der jeweiligen Städte zusammengefasst werden. Da haben wir

große Datenbanken, wo Themen wie Verschattung, spezielle Volumina, Co²-Verbrauch und Ähnliches festgelegt sind.

Wenn wir weiter in die Zukunft schauen – 5 oder 10 Jahre – werden wir über eine ganz andere Datenakquise sprechen. Irgendwann wird es möglich sein, dass sich die Regeln selbst erlernen und aufstellen. Als User füge ich einfach Aerial & Streetview Imagery ein, und mit Deep Learning extrahiert die Software ihre Regeln direkt aus den Bildern und lernt den jeweiligen Architekturstil. Nötig wäre hierfür jedoch, dass CityEngine lernt, die Objekte als solche zu erkennen – zum Beispiel, um Bäume rauszurechnen.

Dann kommt vielleicht ein Stil-Transfer – zum Beispiel, die Eigenschaften einer Straße aus der Züricher Altstadt auf die Stuttgarter Innenstadt zu übertragen. Hier ein Beispiel: Bei der virtuellen Produktion kann man seinen Hintergrund definieren und dann die Stile umschalten und sehen, was einem am besten gefällt – mit einem Klick zwischen Zürich, Venedig, Siena oder San Francisco wechseln. Das geht jetzt zwar auch schon, bedeutet aber immer noch viel Handarbeit und Vorbereitung. >ei



Das neue Interface im Einsatz – einen ausführlichen Workshop von Rainer Duda zum Thema Global Mapping lesen Sie in der nächsten Ausgabe der Digital Production.