2018
6
ISSN 1433-2620 > B 43362 >> 22. Jahrgang >>> www.digitalproduction.com
Publiziert von DETAIL Business Information GmbH

Deutschland € 17,90
Österreich € 19,–
Schweiz sfr 23,–

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

NOVEMBER | DEZEMBER 06:2018

Game Engines
Realtime und Interactive – aber welche taugt für was?

Praxis!
Lighting, ZBrush, Scratch 9, Gaea, Flame und mehr ...

animago 2018!
Alle Gewinner des diesjährigen Awards!

4 194336 217907 06

Der Logon Screen von Scratch 9: Links Zugriff auf sämtliche Settings, rechts der Zugriff auf Logfiles, Lizenzen und die Quicktour und in der Mitte die Projektverwaltung. Das Logo in der oberen Hälfte, dem Viewport, lässt sich im übrigen in den System Settings durch ein eigenes ersetzen.

Scratch, die Neunte

Das Schweizer Taschenmesser der Postproduktion hat einen Facelift bekommen, und obendrein einen recht signifikanten. In diesem Artikel stellen wir das Tool vor und gehen auf die wichtigsten Neuerungen in Version 9 ein. von Matthias Aderhold

Als erstes DI-Tool überhaupt lief Scratch zu 100% auf der GPU – und das schon 2003. Damals war es jedoch nicht mehr als ein (70.000 US-Dollar teurer) 2K DPX-Player. Im Laufe der Jahre kamen dann immer mehr Grading, Editing, und Compositing Tools hinzu, die aus dem Player ein vollwertiges Finishing-System machten. Nun hat Assimilate Version 9 herausgebracht und damit einen signifikanten Schnitt im UI gemacht: weg vom alten „Player vs. Construct"-Konzept, hin zu Tab-basierter Navigation. Damit ist der Einstieg für Neulinge wesentlich einfacher gestaltet, ohne dem Operating Speed Abbruch zu tun.

Auch unter der Haube hat sich eine Menge getan: Version 9 bohrt die Performance in der File-IO auf, unterstützt Matchbox Shader, die das vorhandene Toolset locker um 200 neue Tools erweitern, und Support für 180-Grad-Stereo-VR wurde ebenfalls sowohl für Live-Capture und -streaming als auch für Postproduktions-Workflows hinzugefügt. Doch nicht nur das Finishing Toolset wurde aufgestockt, auch DITs werden Freude am neuen Scratch haben – nicht nur dank verbesserter Transcoding Performance, sondern auch durch dedizierte Tools und Funktionen, die den DIT-Alltag wesentlich erleichtern und eine Menge Klicks sparen.

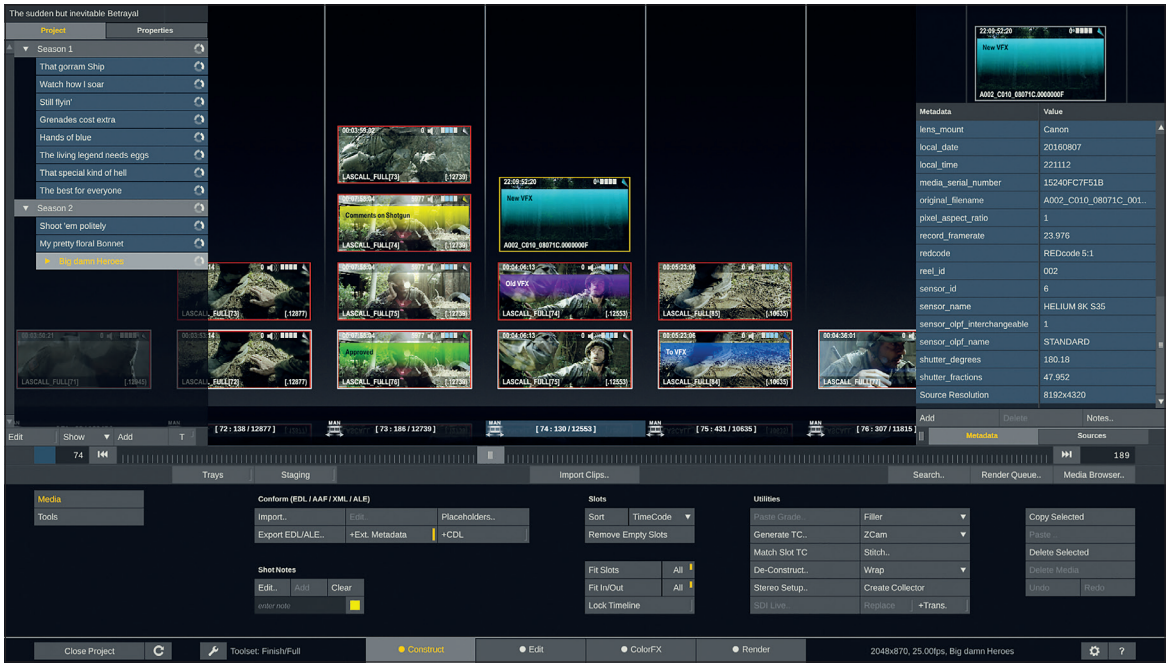
Der Grundriss des Assimilate Scratch

Im Grunde ist Scratch recht einfach aufgebaut: Es gibt einen Logon Screen, von wo aus man Projekte erzeugen und Zugriff auf diverse Settings hat: System-Settings – hier wird u.a. der SDI-Output konfiguriert. Scratch unterstützt jegliche Video-IO von Aja, Blackmagic und Bluefish 444. Aber auch Grading Panels können hier aktiviert (deren Mapping im Übrigen 100% User-defined ist) und sogenannte Custom Commands erstellt werden. Custom Commands sind vom User definierte Buttons, die an bestimmten Stellen im UI platziert werden können (z.B. im Rechtsklickmenü) und Funktionen ausführen können, wie etwa ein Plug-in inkl. vordefiniertem Preset anwenden oder ein selbst geschriebenes Script ausführen. Scratch 9 liefert sogar eigene Scripts mit, die Clips an Nuke oder After Effects schicken und den finalen Render auch wieder zurückholen können – jeweils nur mit einem Klick.

Öffnet man ein Projekt, findet man nun im neuen V9-Release vier Tabs vor: Construct, Edit, ColorFX und Render. Außerdem findet sich am unteren Bildschirmrand ein Toolset-Dropdown, mit dem man sein Toolset je nach Anwendung beschneiden (und damit vereinfachen) kann. Im folgenden schauen wir uns selbstverständlich das gesamte Toolset an. Zurück zu den Tabs:

Construct – in diesem Tab wird Footage importiert, Timelines conformt und Assets gemanagt

Etwa in der Mitte des UIs findet man den Button „Import Clips..". Klickt man denselben, findet man sich in einem File Browser



Das Construct-Tab: Hier finden Import, Conform und Asset Management statt. Dabei werden Clips in Slots angeordnet – Versionen eines Clips werden vertikal aufeinandergestapelt. Farblich markierte Nodes helfen bei der Zuordnung.

wieder, welcher den User einzelne Clips, aber auch ganze Ordner importieren lässt. An unterstützten Formaten mangelt es nicht: Von Arri Raw, RED Raw, Sony Raw über die gängigen Quicktime- und .mxf-Formate inklusive aller denkbaren Codecs bis hin zu CinemaDNG, Canon Raw, Phantom Raw und sogar Exoten wie Kinefinity Raw ist alles vorhanden. Da Scratch nativ in Full Range RGB arbeitet, wurde in V9 für YUV-basierte Formate die Wandlung nach RGB auf die GPU verlagert – die ist nicht nur schneller, sondern geschieht auch noch in 32 Bit Floating Point und ist damit akkurater und erhält wesentlich mehr Bildinformation als die gängige Wandlung auf der CPU. Zurück ins Construct-Tab – lädt man sich einige Clips in die aktuelle Timeline, werden diese sortiert in Slots in einer Art Storyboard-Ansicht dargestellt – damit fällt das Sortieren von Clips und Versionen derselben leichter. Das Slot-basierte Konzept der Scratch Timeline ist überhaupt eine großartige Entwicklung, denn es ist dreidimensional: Auf der X-Achse liegen die Clips – einer nach dem anderen. Auf der Y-Achse eines jeden Slots kann man Versionen ablegen. Das können nicht nur verschiedene Versionen eines Grades oder VFX-Shots sein, sondern auch verschiedene Takes von ein und demselben Shot einer Szene. Wichtig zu wissen: Am Ende wird immer der zuunterst liegende Clip heraus gerendert. Auf die Z-Achse kommt man mit einem Klick auf „Staging". Auf den ersten Blick hat sich nicht viel verändert. Man hat immer noch den jeweiligen Clip in dem Slot, in dem er sitzt, vor Augen. Jedoch ist dieser Slot nun eben nicht vertikal, sondern in der Tiefe zu betrachten. Hier werden jetzt alle Elemente abgelegt, die z.B. für einen Composite gebraucht werden, wie etwa Alphas oder Texturen. Diese sind ja weder Versionen des aktuellen Clips, noch sind es eigenständige Clips im Timeline-Kontext. Auf diese Art bleibt die Timeline schön sauber.

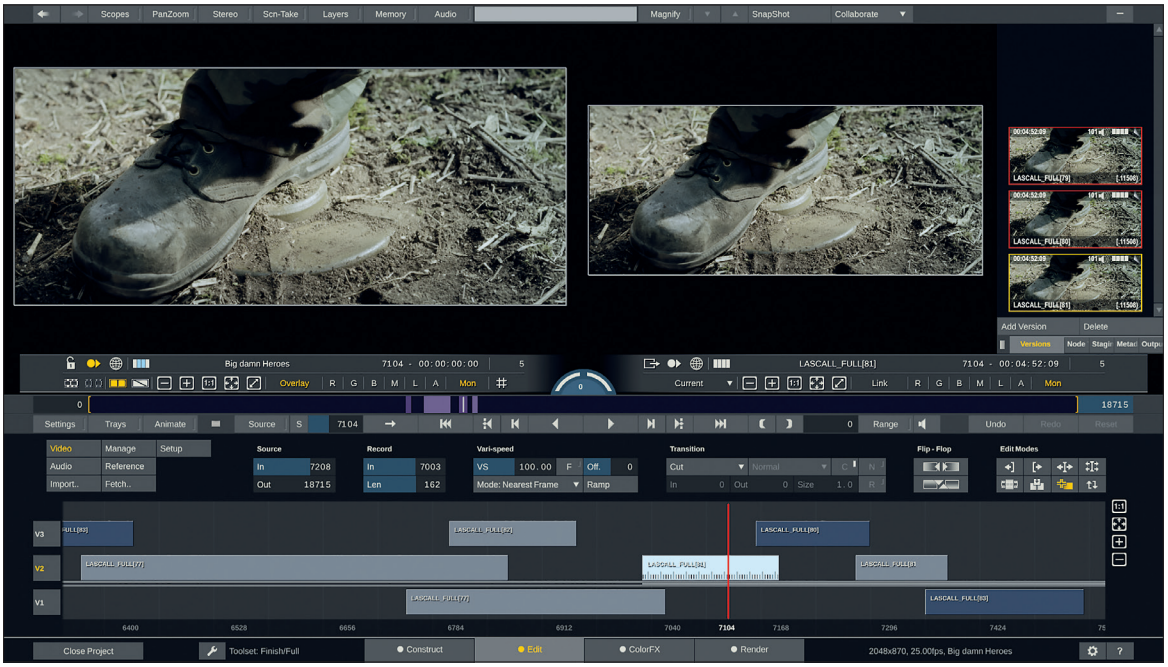
Scratch verfügt über sogenannte Swipe Areas – das heißt, man stößt mit dem Cursor an den Bildschirmrand und ein Menü poppt auf. Im Construct-Tab gibt es zwei davon: Links befindet sich der Project Tree (Hotkey „Q"), mit welchem man Timelines erstellen und auch in Gruppen anordnen kann. Rechts findet man den Metadata Stack (Hotkey „W"), der einem immer die Metadaten des aktuell angewählten Clips anzeigt. Scratch liest eine ungeheure Menge an Metadaten aus – sei es RED Raw, DNx .mxf oder simples Quicktime ProRes – man findet hier mehr Metadaten als in den meisten anderen Tools. Das schöne an Scratch ist, dass man jede Form von Metadaten im Anschluss für Burn-ins oder Export Filenames nutzen oder auch als .ale exportieren kann. Der Metadata Stack hat am unteren Ende noch ein Sources-Tab. Wenn der momentan angewählte Clip ein Composite ist, der aus mehreren Source Clips oder Compositing-Elementen zusammengesetzt ist, findet man hier eine schnelle Auflistung aller im Comp verwendeten Sources. Die Construct-Oberfläche macht Scratch zu einem ausgezeichneten Tool für Review und Asset Management – vor allem da es sich via XML-Scripting gut mit z.B. Shotgun, Ftrack und dergleichen verständigen kann.

Eine große Menge an Funktionalität versteckt sich im Media Browser – eine tabellarische Auflistung aller Clips im Projekt, mit der man wunderbar Batch-Operationen ausführen kann. Zum Beispiel kann man Batch Renaming anhand von Metadaten vornehmen – das ist besonders hilfreich bei Dailies Workflows, wo man z.B. die Reel-ID eines jeden Clips mit dem Source Filename desselben ersetzen möchte, bevor man Avid DNxHD .mxf rausrechnet. In dem Fall kann man einfach die jeweilige Spalte anwählen und z.B. mit #sname (source name) oder einer Kombination aus mehreren Hashcodes füllen.

Des Weiteren kann man alle Metadaten verändern, neue hinzufügen, aber auch LUTs, CDLs, Audiodateien und dergleichen batch-importieren (und anhand von Timecode, Filename oder anderen Metadaten auf die jeweiligen Clips matchen). Seit Version 9 verfügt Scratch über einen intelligenten LTC-(Linear oder Longitudinal Timecode) Detection-Algorithmus, der auch besonders im Dailies-Bereich hilfreich ist, wenn mit nicht TC-fähigen DSLR-Kameras gefilmt wird.

Edit

Wenig überraschend kann man hier den conformten Schnitt anpassen. In der unteren Bildhälfte wird die Timeline angezeigt, in der oberen kann man das aktuelle Bild unterhalb des Cursors begutachten. Zugegebenermaßen stiften die 8 (!) Editmodi im Scratch Editor mehr Verwirrung, als dass sie die Arbeit erleichtern. Audio-Editing steht ebenfalls noch auf der Todo-Liste von Assimilate. Dennoch: Die überwiegende Mehrheit an Filmen und Videos wird immer noch auf dedizierten Schnittsystemen geschnitten, und Assimilate hat auch nicht vor, das in nächster Zukunft zu ändern. Viel mehr wurde bei Scratch 9 Wert darauf gelegt, das Conforming zu verbessern und mehr Effekte und Animationen aus den gängigen .xml- und .aaf-Formaten zu übernehmen. Das Edit-Menü besteht im Wesentlichen aus vier verschiedenen Modu-



Im Edit-Tab wird (wenig überraschend) geschnitten! Dabei hat man aber Zugriff auf die Scopes, den Version Stack, um ggf. Shots auszutauschen, und Framing Tools. Alles andere ist dem ColorFX-Tab überlassen.

len: Video (In/Out Timecodes, Varispeed, Transitions und die Edit-Modi), Manage (Slot- und Track-Management), Setup (Framing im Timeline-Kontext), Audio (Audiofiles per Clip oder auf die gesamte Timeline laden) und Reference – hier kann man eine Schnittreferenz via Drag-and-drop mit der Timeline assoziieren, um dann den Schnitt während des Conforms in Dual View (Hotkey „D") oder Split View (Hotkey „S") zu überprüfen. In der Mitte des UIs finden sich eine Mini-Timeline sowie Playback und Viewport Controls. Wichtig (wenngleich etwas versteckt) sind auf der linken Seite die Buttons für Settings, Trays und Animation. In den Settings finden sich Playback-Optionen, Color Management, Guides wie etwa Action und Title Safe, aber auch LUT-Export-Settings und Panel Mappings.

Die Trays unterteilen sich in zwei Tabs: Project und Gallery. Während im Project-Tab ausschließlich Clip und Frame References gespeichert werden, die tatsächlich auf bestehende Clips in einer Timeline linken, werden im Gallery-Tab pure Grades gespeichert, die sich im Userfolder auf Finder-/Explorer-Ebene wiederfinden und so auch zu anderen Systemen mitgenommen werden können. Hinter dem Animation-Button befindet sich ein Spline-Editor, mit dem man jegliche Animation feintunen kann. Am oberen Bildschirmrand findet man Tools, die sich als Floating Windows bequem auf dem UI verschieben lassen – darunter sind Scopes, ein Scene-/Take-Updater-Tool und ein Audio-Mixer, mit dem sich der Ton millisekundengenau anpassen lässt. Die wichtigsten Hotkeys für den Viewport sind Space und Alt. Hält man Space gedrückt, kann man das Bild im Viewport frei verschieben – mit Alt lässt es sich skalieren.

ColorFX

Hier spielt sich der Großteil der Arbeit in Scratch ab: Grading und Compositing. Linkerhand lässt sich der Layer Stack ins UI swipen, während man rechts den Version Stack herbeizaubern kann. Dieser zeigt zunächst alle Versionen eines Clips im jeweiligen Slot an – diese lassen sich auch anwählen, während ein Clip abspielt, und per Drag-and-drop in den Split oder Dual View bewegen. Doch der Version Stack ist nur einer der fünf in diesem Swipe-Menü: Über die Tabs am unteren Ende kann man in den Node Tree, Staging, den Metadata Stack oder den Output Stack wechseln – zu Letzterem später mehr.

Der Node Tree füllt sich erst, sobald man den aktuellen Shot nestet (was im Grunde jedoch einfach einen neuen Node an den Source Node anfügt) oder andere Compositing-Elemente wie Texture Clips oder (OFX-) Plug-ins hinzufügt. Scratch ist unter der Haube ein Node-basiertes Tool, obwohl die meisten User den Layer Stack auf der linken Seite bevorzugen und damit auf einem einzigen Node arbeiten. Dennoch kann grundsätzlich jeder Node im Node Tree eine unbegrenzte Anzahl an Layern haben und noch wichtiger: Caching in Scratch funktioniert ebenfalls Node-basiert – für jeden Node lässt sich der jeweilige Input in ein frei wählbares Image-Sequence-Format cachen.

Neben den gängigen OFX-Plug-ins kann man ab Version 9 auch Matchbox Shader in Scratch verwenden. Matchbox wurde vor einigen Jahren von Autodesk für Flame entwickelt – User konnten ihre eigenen Shader schreiben und sie mittels des Matchbox Nodes in den Flame Batch laden. Im Laufe der Zeit hat sich so eine kostenlose Kollektion von Shadern auf **https://logik-matchbook.org** zusammengestellt – ganz und gar Community-basiert. Das Schöne an diesen Shadern ist, dass sie größtenteils in Echtzeit laufen und ständig von der Community weiterentwickelt werden. Mit Version 9 können User nun diese Matchbox Shader (die Kollektion ist mittlerweile auf ca. 300 Shader angewachsen) auf ihre Shots und Composites in Scratch packen – ein schönes Add-on für Scratchs Realtime-Compositing-Environment. Gerade bei anspruchsvolleren Comps macht dann die Nutzung des Node Trees Sinn.

Render

In diesem Tab kloppt man nun die getane Arbeit wieder aus Scratch heraus. Auch hier findet man einen Node Tree vor, mit dem man seine Timeline exportieren kann – eine clevere Art, um Output-Versioning zu betreiben: Man kann hier sogenannte Transformer Nodes nutzen, um den Content zu skalieren (z.B. von 4K herunter auf HD), Colorspace Conversions anzuwenden (z.B. von Rec. 709 auf P3) oder auch spezielle Gradings oder LUTs auf einzelnen Branches des Trees anzuwenden.

Besonders für Dailies-Workflows interessant ist auch der Burn-in Node, der es erlaubt, jegliche Metadaten eines Clips, aber auch Letterboxen oder Action/Title Safes ins Bild zu packen. Am Ende eines Branches fügt man dann einen Encoder Node an – das kann ein Avid DNx-Encoder sein, aber auch H.264/H.265, ProRes (ebenfalls auf Windows verfügbar) oder simple Image-Sequence-Exporter. Mit jedem Node kann man via Doppelklick ins ColorFX-Tab wechseln und Grades auf die gesamte Timeline legen (oder das Burn-in konfigurieren). Sobald man mit einem Output Node im ColorFX- oder Edit-Tab

Der Hauptarbeitsbereich in Scratch ist das ColorFX-Tab – FX weil es eben wesentlich mehr, als nur Color kann. Mittels Layer Stack (links) und Node Tree (rechts) lassen sich durchaus anspruchsvolle Composites erstellen – gerade mit den jetzt verfügbaren Matchbox Shadern läuft das meiste immer noch schneller als in anderen, dedizierten Compositing Tools.

ist, wird einem rechts auch der Output Stack anstatt des Version Stacks angezeigt – so ist es auch möglich, zu anderen Output Nodes zu wechseln, ohne erst wieder ins Render Tab zu müssen. Allerdings ist die Stack-Ansicht gerade bei größeren Output Trees wesentlich unintuitiver als der schöne Node Tree.

Aber zurück ins Render-Tab: Anfangen tut alles mit dem Main Output Node – dieser repräsentiert quasi die jeweilige Timeline (der Project Tree ist linkerseits ebenfalls im Render-Tab via Swipe verfügbar). Standardmäßig hat der Main Output Node die Auflösung, Framerate, Aspect etc., die in den Project Settings als Default vorgegeben sind – jedoch kann man sowohl die Project Settings als auch die Settings des Main Output Nodes (und damit der Timeline) jederzeit ändern. Überhaupt: In Scratch kann man eigentlich alles zu jeder Zeit ändern – man ist nie festgenagelt auf ein spezielles Setting. Damit kann man flexibel auf Änderungen in der Pipeline reagieren – etwa wenn sich mitten im Dreh die Kamera verabschiedet und eine andere hergenommen werden muss.

Wenn es an den Output-Filename im Rendering geht, verfügt Scratch über ein ausgeklügeltes System von #-codes, welche man nutzen kann, um den Filename dynamisch anhand von Metadaten zu gestalten. So kann man z.B. einfach #name verwenden, um die gesamte Timeline als einen Clip mit dem Namen des Output Nodes zu exportieren. Fügt man jedoch einen #-Code ein, der von Clip zu Clip unterschiedlich ist, so ist Scratch gezwungen, einzelne Clips zu rendern – z.B. indem man #sname (Source Name) verwendet. Selbstverständlich kann man #-Codes auch kombinieren und so etwa #scene_#take_#sname.#ext anlegen. Auch Slashes sind möglich, um dynamisch Ordner anzulegen – etwa anhand der Reel-ID eines Clips: Dailies/#reelid/#sname.#ext. Alle Output-Filenames lassen sich natürlich als Preset speichern, sodass man sich nicht jedes Mal die Finger wund tippen muss – auch sind alle verfügbaren #-Codes via Dropdown inkl. kurzer Erklärung verfügbar, damit man sie nicht alle auswendig lernen muss. Speichert man sich den Output Tree (oder auch nur einen Branch desselben), werden selbstverständlich alle Filenames in diesem Template ebenfalls gespeichert – und man kann obendrein ein solches Template noch in den Project Settings als Default angeben. Damit wird das gewählte Template inklusive aller Settings und Filenames automatisch an jede neu erstellte Timeline angehängt. Vieles in Scratch lässt sich so weitestgehend automatisieren – wer noch mehr möchte, sei dazu eingeladen, sich in die Untiefen des XML- und Python-Scriptings zu stürzen. Ebenfalls im Render-Tab vorhanden: die Möglichkeit, QC- und HDR-Reports zu erstellen – sowohl für die aktuelle Timeline als auch Group als auch projektweit. Hier kann man auswählen, welche Metadaten im QC-Report enthalten sein sollen, von welchem Frame ein etwaiges Thumbnail gezogen wird und wie das Layout des Reports gestaltet werden soll. Wer sich mit XSLT auskennt, kann auch eigene Layouts hinzufügen.

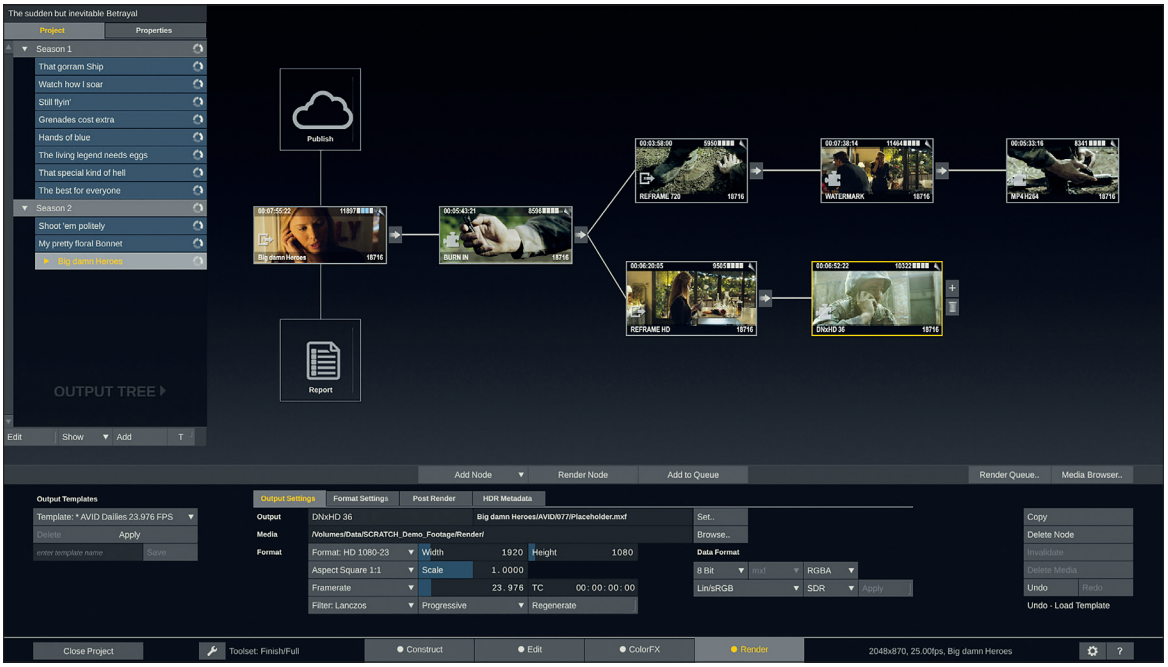
Und das Ganze für wen?

Soweit zur groben Vorstellung von Scratch 9 – aber für wen eignet sich die Software denn nun? Scratch hat als medientechnisches Schweizer Taschenmesser mehr als nur einen Schwerpunkt. Da wäre zunächst der Dailies-Bereich: Scratch ist in der Regel unter den ersten Applikationen, die neue Formate unterstützen – wie etwa das neue Blackmagic Raw oder Canon XF-HEVC, welche beide zur IBC eingeführt wurden. Dabei werden alle gängigen Kameraformate in einem Wahnsinnstempo nach DNxHD, ProRes, H.264 oder H.265 transkodiert – inklusive Metadaten, Reports und Nettigkeiten wie Background Rendering.

Ein anderer Aspekt ist Creative Finishing – Scratch ist wesentlich mehr als nur ein Grading Tool – Realtime Compositing, Matchbox- und OFX-Support sowie Node-basiertes Arbeiten erleichtern es ungemein, schwierige Gradings zu meistern, während der Kunde ungeduldig danebensitzt. Des Weiteren wird Scratch von vielen Studios und VFX-Häusern als Asset Management und Screening Tool genutzt. Schneller .edl/.xml/.aaf-Conform? Kein Problem. Noch schnell CDLs aus Nuke via Batch Import auf alle Shots matchen? Easy. Flott noch VFX-Shots austauschen, Grade anpassen und frische CDLs batch-exportieren? Auch keine Frage. Sobald man dann noch anfängt, das ganze via Scripting an Ftrack, Shotgun und Konsorten zu hängen, hat man eine mehr als würdige Alternative zu Pdplayer und RV.

Und zuletzt wäre da noch Live-VR-Streaming und VR-Postproduktion. Scratch unterstützt nämlich nicht nur sämtliche Headsets als Ausgabegeräte, sondern kann in der VR-Version obendrein noch in einem 360-Grad-Environment graden, painten und compen. Das geschieht, ohne Artefakte zu erzeugen, und, wenn man will, live auf einem HLS- oder direkten Kamera-Stream inklusive live Stitching.

Wer neugierig geworden ist, kann sich online die 30-Tage-Trial herunterladen. Scratch (VR) ist als monatliche Subscription sowie als Jahreslizenz und sogar als permanente Lizenz erhältlich. **› ei**



Render-Tab – der hier verfügbare Node Tree eignet sich hervorragend, um Output-Versioning zu betreiben. Verschiedene Auflösungen, Framerates, Grades, Farbräume, Burn-ins und Ausgabeformate sind schnell angelegt und als Template abgespeichert. Essenziell wichtig ist hier das Hash-Code-basierte Output-Filenaming von Scratch, womit sich Filenames dynamisch anhand von Metadaten generieren lassen.