

2019

ISSN 1433-2620 > B 43362 >> 23. Jahrgang >>> www.digitalproduction.com

Publiziert von DETAIL Business Information GmbH

Deutschland € 17,90

Österreich € 19,-

Schweiz sfr. 23,-

2

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

APRIL 02:2019



Plug-ins

Oder soll es länger dauern als unbedingt nötig?

animago!

Monster mit den Gewinnern La Noira & Mackevision

und vieles mehr!

BMD Pocket 4K, Caracter Animator, Flame, Arri ...



4 194336 217907 02

Plug-ins für Autodesk Flame

Die Geschichte von Flame und seinen Plug-ins ist von Anfang an, sagen wir mal, schwierig. Das geht schon bei der Nomenklatur los. Plug-ins hießen bei Flame seit jeher Sparks, damit auch hier Discreets Hang zum Brennbareren – Fire, Smoke, Inferno, Backdraft etc. – Rechnung getragen werden konnte. Doch da fängt es mit den Besonderheiten gerade erst an. Während andere Plattformen entweder eine breite Userbase (After Effects) oder einen gemeinsamen Standard (OFX) hatten, die sie für Plug-in-Developer interessant machten, fristeten die Sparks immer ein gewisses Nischendasein.

von Christoph Zapletal

Dies schlug sich zum einen in der Auswahl nieder, denn nicht jeder Hersteller konnte es rechtfertigen, für ein so kleines Geschäftsfeld die eigenen Plug-ins auf die Spark-Schnittstelle zu adaptieren, zum anderen im Preis, der nicht nur durch die kleine Userbase, sondern auch durch die Hochpreisigkeit von Flame selbst immer signifikant über allen anderen Plattformen lag. All das hat sich bis heute nicht komplett nivelliert, aber die Zeiten, in denen Plug-ins für Flame jenseits der 8.000 Euro kosteten, sind mit dem Öffnen der Software hin zum Mac und der Etablierung eines Mietmodells glücklicherweise vorbei. In diesem Artikel wollen wir zum einen den Blick auf das bestehende Angebot an Plug-ins beziehungsweise Sparks für Flame werfen, zum anderen uns aber auch anschauen, welche anderen Schnittstellen in Flame sich entwi-

ckelt haben und wie sowohl kommerzielle als auch non-kommerzielle Entwickler neue Wege gehen und damit die Workflows in Flame nachhaltig verändern.

Sapphire – Last Man Standing?

Der Quasi-Standard für Flame-Sparks sind seit jeher die Sapphire Plug-ins. Waren sie früher noch das Flaggschiff-Produkt von Genarts, gehören sie mittlerweile zum immer größer werdenden Portfolio von Boris FX. Sie gehörten und gehören zum Must-have einer jeden Postproduktion. Früher gehörten dazu noch die 5D (später Speedsix) Monsters und die Tinder Tools, die aber den Weg alles Endlichen gegangen sind. Während einige wenige der Monster-Plug-ins ihren Weg in das Sapphire-Paket gefunden haben – schließlich hat Genarts die Firma Speedsix damals

gekauft, nur um einige Jahre später selbst von Boris FX gekauft zu werden – sieht es bei Furnace, Tinder und Keylight ein bisschen anders aus. Nachdem The Foundry – bis zu diesem Zeitpunkt ein reiner Plug-in-Hersteller – die Entwicklung und den Vertrieb von Nuke übernommen hatte, entschied Autodesk im Dezember 2008, The Foundry aus dem autorisierten Spark-Entwicklerprogramm zu nehmen. Damit wurde für The Foundry die Hürde, Plug-ins für Flame zu programmieren, um einiges nach oben gelegt, da The Foundry nun keinen Zugang zu Beta-Software und keinen direkten Draht zu den Entwicklern von Autodesk mehr hatte. Im Jahr 2011 endete dann, fast erwartungsgemäß, jedes Engagement von The Foundry im Bereich Autodesk und so war man als Flame Artist um drei großartige Plug-in-Sets ärmer. So bleibt tatsächlich als komplette

Footage Copyright: Finn Jäger





Nur eine kleine Auswahl der verfügbaren Matchbox Shader

Suite nur noch Sapphire von Boris FX. Eine ausführliche Beschreibung der Plug-ins von Sapphire 2019 lesen Sie ab Seite 40.

Die Funktionsweise ist in Flame meist identisch mit ihren Pendants aus After Effects, Nuke und Co., nur eben mit Anpassungen an das Flame UI. Seit einigen Versionen gibt es nun auch GPU-Support für die meisten Sapphire Plug-ins in Flame, und mit entsprechenden Lizenzen können sie auch auf den Burn-Renderfarmen genutzt werden. Anders als auf anderen Plattformen sind die Sapphire Sparks nicht als individuelle Units, sondern nur als Komplettpaket verfügbar. Eine Perpetual License liegt bei 1.695 US-Dollar, während eine jährliche Subscription mit 495 US-Dollar zu Buche schlägt.

Mocha direkt in Flame

Die wohl spannendste Flame-spezifische Neuerung in Sapphire 2019 ist die Integration von Mocha Shape Tracking in Sapphire. Sapphire bietet hier drei unterschiedliche Workflows an, mit denen man auf die Features von Mocha zugreifen kann. Der erste Workflow ist sehr simpel und für das einfache lokale Begrenzen eines einzelnen Plug-ins gedacht. Unter den Cropping-Settings, die man in den Einstellungen jedes einzelnen Sapphire Sparks findet, erhält man Zugriff auf Mocha. Mocha öffnet sich dann in einem separaten Fenster und lädt automatisch den Clip, an dem der Artist in Flame arbeitet. Das UI entspricht hier ziemlich genau dem des allseits bekannten Mocha Pro, nur eben ohne die Pro-Features. Insert- oder Removal-Features sucht man hier also vergeblich. Hier kann man aber mit dem bewährten Planar Tracker von Mocha seine Roto zur Begrenzung des Effekts erstellen, das Pop-up-Window wieder schließen und

zu Flame zurückkehren, wo nun der Effekt durch die in Mocha erstellte Maske begrenzt wurde. Als zweiten Workflow bietet Boris FX nun einen eigenen Mocha Spark an. Rein technisch macht dieser nun nichts anderes, als exakt das gleiche UI in einem Pop-up-Window zu öffnen und den jeweils aktiven Clip zu laden.

Der einzige Unterschied ist, dass das Mocha Spark nun eine Maske bereitstellt, die zum Beispiel im Batch Tree frei mit einem anderen Spark, einem Matchbox Shader oder zum Beispiel in einem Action Composite verwendet werden kann. Während die ersten beiden Workflows dem User einen fertigen Alpha-Kanal zur Verfügung stellen, zählt der dritte Workflow auf den Export von Metadaten. So kann der Artist, genauso wie bei Mocha Pro, sich sowohl Point-Tracker-Daten als auch Roto Shapes als GMask Nodes aus dem Mocha Spark exportieren und dann in den entsprechenden Flame-Modulen laden.

Dies ist durchaus praktisch, vor allem um sich im Workflow größtmögliche Flexibilität zu bewahren. Allerdings ist es bedauerlich, dass der Export von Roto Shapes einzig und allein in den GMask Node von Flame und nicht in den neueren GMask Tracer oder direkt in Action möglich ist. Der GMask Node wurde von Autodesk nun schon vor mehreren Jahren als „Legacy“ deklariert, und alle neuen Entwicklungen und Features fließen alleine in den GMask Tracer. Es wäre wünschenswert, wenn es hier einen besseren und zeitgemäßen Austausch der Daten gäbe. Letzten Endes ist es streitbar, ob der Mocha Spark wirklich ein Spark oder nicht viel mehr ein App Launcher für ein reduziertes Mocha ist. Letzten Endes ist das aber gar nicht so wichtig, viel entscheidender ist, dass es jetzt eine günstige Möglichkeit für Flame

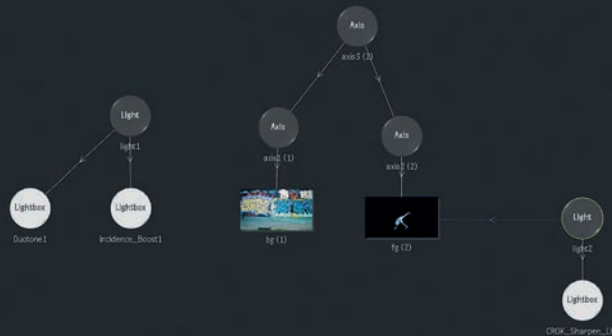
Artists gibt, an den Mocha Planar Tracker zu kommen. Wer nicht unbedingt die Pro-Features von Mocha benötigt und vielleicht eh schon mit den Sapphire Sparks liebäugelt, dem macht Boris FX hier die Entscheidung vielleicht ein bisschen leichter.

Und sonst so?

Photron vertreibt seinen Primatte-Keyer als Spark für Flame. Bei einem aber doch recht saftigen Preis von vier- bis sechstausend Dollar sollte man sich allerdings überlegen, ob das Keying-Toolset von Flame (DP 06:18) nicht vielleicht doch ausreicht. Re:Vision Effects bieten mit dem Reel:Smart Motion Blur und dem Twixtor Timewarper im Gegenzug zwei sehr beliebte Sparks zu durchaus angemessenen Preisen an. Reel:Smart kostet zwischen 145 und 250 US-Dollar, Twixtor ist zwischen 595 und 799 US-Dollar eingepreist, je nachdem ob man eine Node-Locked oder eine Floating-Lizenz benötigt. Auch wenn beide Sparks im Vergleich zu ihren After-Effects-Pendants um einige Funktionen ärmer sind, sind sie immer noch eine wichtige Bereicherung des Flame Toolsets und definitiv einen Blick wert.

Wie die Matchbox geöffnet wurde ...

Würde dieser Artikel sich auf die klassischen Plug-ins, also die Sparks beschränken, wäre er jetzt am Ende und das Fazit entsprechend ernüchternd. Im Vergleich zu anderen Plattformen ist die Auswahl an Plug-ins von Drittanbietern doch recht dürftig. Zum Glück hat auch Autodesk erkannt, dass die klassische Spark-Schnittstelle zu speziell und wartungsintensiv sind und der Flame-Kundenkreis zu klein ist, um andere Plug-in-Hersteller hier



Lightbox lädt zum Experimentieren ein. Wer hat schließlich schon mal mit Noise geleuchtet?

mit an Bord zu bekommen. Um dem entgegenzutreten und auch den Usern selbst eine Schnittstelle zu geben, über die sie ihre eigenen Tools für Flame schreiben können, wurde in Flame 2012 der Matchbox Node für Batch eingeführt. Über diesen Node lassen sich auf GLSL (OpenGL Shader Language) basierte Shader in den Batch Tree integrieren. Hier funktionieren sie augenscheinlich wie die bekannten Sparks, mit zwei wichtigen Unterschieden: Durch die OpenGL-Architektur laufen alle Funktionen automatisch über die Grafikkarte – die GPU-Unterstützung ist also von Haus aus eingebaut. Und der komplette Shader Code wird im Batch Skript mit abgespeichert. Das klingt erst einmal banal, ist aber unheimlich mächtig. Dadurch, dass der komplette Shader mit im Skript abgespeichert ist, müssen anders als bei Sparks frühere Versionen des Shaders nicht auf der Workstation laufen oder Skripte auf aktuelle Versionen konvertiert werden. Darüber hinaus ist es sogar möglich, ein Skript, das Matchbox Shader verwendet, auf einer Workstation zu öffnen, wo die Shader überhaupt nicht installiert sind.

Die Flame-Community brauchte nicht lange, um das Potenzial von Matchbox zu erkennen. Unter logik-matchbook.org finden sich mittlerweile mehrere Hundert Matchbox und Lightbox Shader, die der Community kostenlos zur Verfügung gestellt werden. Und unter flamemark.wixsite.com/postmanvfx bietet Postman VFX auch weit über hundert Shader zum Preis von nur 249 US-Dollar an.

Die Bandbreite der angebotenen Effekte hier abzubilden, würde jeden Rahmen sprengen. Von technischen Tools wie einem Vectorscope, Debändern und Denoisern über Beauty-Shader wie den sehr beliebten CROK_Beauty-Shader von Ivar Beer oder den LS_Wireless von Lewis Saunders bis hin zu prozeduralen Textur-Generatoren ist alles dabei.

Bestätigt durch diesen Erfolg begann Autodesk, Matchbox weiter im System zu etablieren. Zuerst als Effekt, der direkt auf der Timeline angewendet werden konnte –

dieses ist mit Sparks auch schon möglich gewesen. Dann aber konnten Matchbox Shader direkt in Action angewendet werden. Wenn man sie auf die Diffuse-Map anwendet, erreicht man einen ähnlichen Effekt wie im Matchbox Node in Batch oder auf der Timeline. Aber durch die enge Verzahnung in Action können Matchbox Shader nun auch auf Specular oder Normal Maps angewendet werden. Für 3D-Compositing ergeben sich hier völlig neue kreative Möglichkeiten. Mittlerweile hat sich Matchbox so fest in Flame etabliert, dass Autodesk sich entschieden hat, das neue Grading- und Look-Development-Tool von Flame, das Effects Tab bzw. den Image Node, rund um die Action- und Matchbox-Shader-Architektur zu konzipieren. Dementsprechend sind auch zwei der Key Features des letzten Flame Release, der A2 Beauty Shader und das Master Grade Tool, als Matchbox Shader umgesetzt worden.

Eine weitere interessante Weiterentwicklung des Matchbox-Konzeptes sind die Matchbox Camera FX – wie der Name schon vermuten lässt, werden hier Shader nicht auf eine einzelne Textur angewendet, sondern auf die Kamera des Action-3D-Compositors. Dadurch haben die Shader direkten Zugriff auf Informationen wie zum Beispiel Tiefe oder Occlusion, ohne dass man separate Passes generieren muss. Matchbox hat mittlerweile sogar den Sprung auf andere Plattformen geschafft. Sowohl Baselight in der Version 5 als auch Assimilate Scratch in Version 9 unterstützen Matchbox Shader.

Von Matchbox zu Lightbox

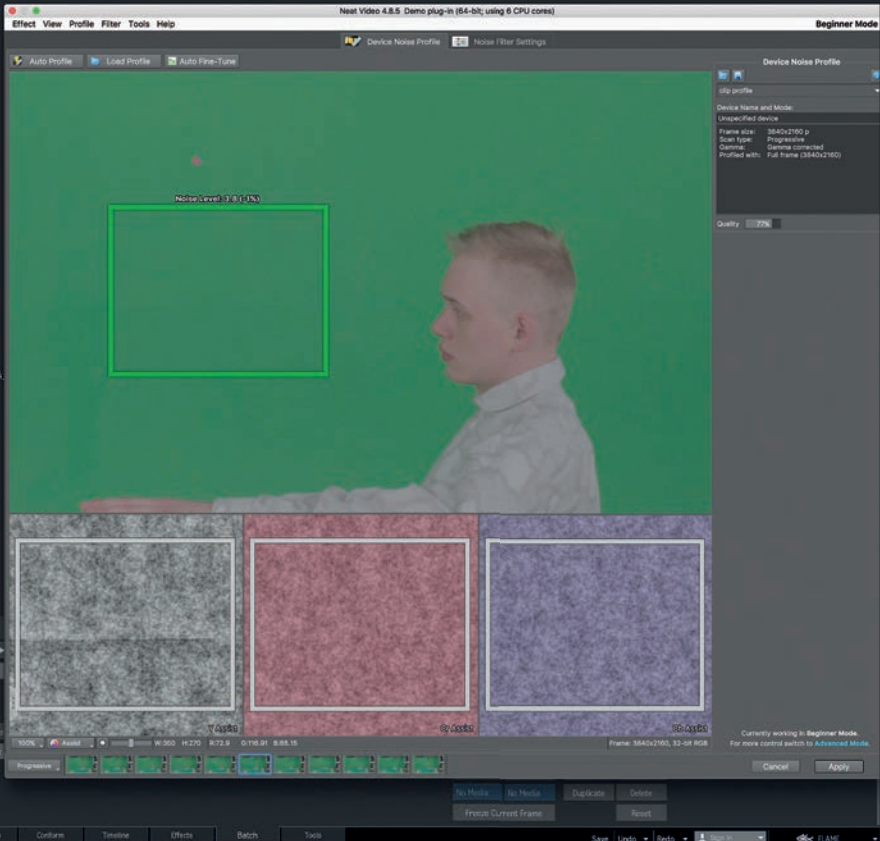
Aber die wohl innovativste Integration der GLSL-Shader in den Action Workflow stellt Lightbox dar. Während Matchbox Shader auf der Textur eines Objekts angewendet werden, also bevor sie im 3D-Raum positioniert werden, werden Lightbox Shader mit einem Licht im Action Composite verbunden und über dieses angewendet. Das heißt, ein Lightbox Shader wird durch das Licht auf die Szene geworfen, und alle Parameter des Lichtes, inklusive Falloff und

Occlusion, werden berücksichtigt. Wenn man schaut, welche Lightbox Shader Autodesk Flame beigefügt hat, wird schnell klar, welche Idee hier bei den Entwicklern vorherrschte: Alle Shader sind ausschließlich Farbkorrektur-Tools, die mithilfe der Lichter angewendet werden können. Doch da auch Lightbox über GLSL skriptbar und somit für die Community offen ist, hat es nicht lange gedauert, bis auch Lightbox Shader jenseits der Farbkorrektur verfügbar wurden. Bokeh-Effekte, Denoiser, Blurs und volumetrische Effekte wie Flammen, Schnee und Eis kann man jetzt durch das Lighting Framework von Action schleudern.

Pybox – jetzt wird gecodet ...

Pybox ist weder ein Plug-in noch ein Shader, und trotzdem kommen wir nicht umhin, über diesen neuen Node für Flame zu sprechen. Pybox ermöglicht direkt von Batch den Zugriff auf Renderer anderer Software. Was heißt das in der Praxis? Nun, Pybox besteht aus zwei Komponenten: einem Node, den der Artist in seinen Flow Graph integriert, und einem Handler, einem in Python geschriebenen Skript, welches definiert, in welche Software und in welches Skript mit welchen Parametern und welchen Variablen Bilddateien geschickt werden und woher Flame das Resultat dann wieder zurückbekommt. Um ein konkretes Beispiel zu nennen: Um einen in Nuke gebauten Unschärfefeffer auf ein Batch Composite in Flame anzuwenden, lädt der Artist einen Pybox Node und den entsprechenden Handler. Innerhalb der Pybox erscheint nun ein UI, das den Artist Werte wie die Unschärfe, das Bokeh, das Blooming, die chromatische Aberration und so weiter einstellen und animieren lässt. Für den Artist sieht so ein Pybox-UI aus wie jeder andere Flame-Node. Unter der Haube schickt Pybox die Daten aber an Nuke, lässt dort die Bilder berechnen und lädt diese zurück in den Batch Viewer. Und Nuke ist hier nur ein Beispiel. Möglich ist dies mit so ziemlich jedem Renderer, der über Python skriptbar ist. Voraussetzung ist

Neat Video direkt in Batch – hiermit dürfte für einige Flame Artists ein lang ersehnter Wunsch in Erfüllung gegangen sein.



natürlich, dass die entsprechende Software auch auf der Flame Workstation installiert ist. Pybox ist sicherlich nichts, was direkt out-of-the-box funktioniert, sondern muss schon vernünftig in die eigene Pipeline integriert sein. Ist dieses aber einmal geschehen, eröffnen sich dem Artist viele neue Möglichkeiten, seinen Workflow zu verändern. Und auch kommerziellen Anbietern steht nun eine weitere Schnittstelle zur Verfügung, über die sie ihr Produkt in Flame integrieren können.

Das erste kommerzielle Tool, welches Pybox für sich nutzt, ist der Baselight Renderer für Flame. Dieser wird als kleines Programm auf der Flame Workstation installiert und kann von nun an über den entsprechenden Handler von Pybox aus gesteuert werden. Die Aufgabe des Baselight Renderers ist es, von Baselight exportierte BLG-Files zu lesen und auf das von Pybox bereitgestellte Material anzuwenden. So können Flame Artists Baselight Grades ohne Roundtripping entweder direkt auf ihre Timeline oder innerhalb ihres Batch Skripts anwenden. In Zeiten, wo Grading und VFX meist parallel laufen und dem Änderungswillen der Kunden keine Grenzen gesetzt werden, kann ein so flexibler Workflow das eine oder andere Timing retten. Die Leistung dieses Workflows ist schon beachtlich. Es wird nicht nur das reine Grading aus Baselight übernommen, auch Masken, Trackings, Keys und Resizes übernimmt der BLG Renderer anstandslos.

Einige Einschränkungen gibt es allerdings noch. So unterstützt Pybox im Moment nur einen Input – wurde in Baselight eine externe Maske oder ein Grain Layer genutzt, muss dieses händisch in Flame nachgebaut werden. Außerdem ist der BLG-Renderer im Moment nur für Linux verfügbar, eine Mac-Version scheint aber in Arbeit zu sein. Was in der Flame-Version des Baselight Renderers leider komplett fehlt, ist die Möglichkeit, das Grading von Flame aus noch zu ändern, wie es bei der Edition-Version des Baselight Renderers für Nuke jetzt schon möglich ist.

Und OFX?

Als Autodesk im Herbst 2018 die derzeit aktuelle Flame-Version (2019.2) präsentierte, dürfte der Traum eines manchen Flame-Artists in Erfüllung gegangen sein, da dort tatsächlich eine OFX-Version von Neat Video unterstützt wurde. Zwar war es schon vorher mit einem Tool namens OFX-Wrapper möglich, Neat Video zu starten, aber das Ganze war eher ein Workaround und somit auch instabil. Und eine Sache fällt deutlich auf: Das entsprechende Plug-in heißt OFX und nicht Neat Video. Insofern scheint Autodesk angefangen zu haben, OFX zu unterstützen. Hier mit Neat Video zu beginnen, scheint nur logisch, war es doch das Tool, das viele Artists am schmerzlichsten vermissten. Es bleibt zu hoffen, dass mit den nächsten Releases sich mehr und mehr OFX-Plug-ins dazugesellen.

Fazit

Wenn man auf die klassischen Sparks guckt, ist das Angebot an Plug-ins für Flame überschaubar.

Doch anstatt an dieser Schnittstelle weiter heruzudoktern, entschied sich Autodesk, mit Matchbox, Lightbox und Pybox neue Wege zu gehen und somit der Flame-Community mehr Möglichkeiten zu geben, selbst Tools beizusteuern – mit durchschlagendem Erfolg.

Matchbox ist aus Flame nicht mehr wegzudenken und hat es schon auf zwei weitere Produkte von Mitbewerbern geschafft. Lightbox bietet eine völlig andere Art und Weise, Effekte anzuwenden, und Pybox lässt Flame auf eine Vielzahl anderer Renderers zugreifen.

Und die beginnende Unterstützung von OFX lässt hoffen, dass Flame mit seinen Plug-ins nicht wieder in eine Sackgasse gerät, wie es bei der ursprünglichen Sparks-Schnittstelle der Fall war. >ei



Christoph Zapletal ist seit mehreren Jahren als freiberuflicher Compositing Artist in Hamburg tätig. Sowohl mit Autodesk Flame als auch The Foundrys Nuke arbeitete er bereits an diversen Werbe- und Spielfilmprojekten mit. Darüber hinaus ist er als Dozent für fxphd und das iSFF (Institut für Schauspiel, Film- und Fernsehberufe) in Berlin tätig. Weitere Infos unter: www.christophzapletal.de