2023
1

ISSN 1433-2620 > 27. Jahrgang >> www.digitalproduction.com
Publiziert von Busch Glatz Germany GmbH

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

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

JANUAR | FEBRUAR 01:2023

Star Trek
Unendliche Weiten & Virtual Production

Projekte
HFF, Witcher, Black Adam und Spongebob

Tools
CTA, Womb3D, Road Generator und Adobe

Tests
4D People, XMG, Topaz Video AI und mehr

Nuke Tools Vol.2 – User Interface

Haben wir in der letzten Ausgabe noch unseren Node Tree mit ein paar passenden Tools aufgehübscht, wollen wir in dieser Ausgabe gucken, welche Tools uns helfen können, die UI von Nuke ein bisschen mehr unseren Bedürfnissen anzupassen.

von Christoph Zapletal

To-Do-List

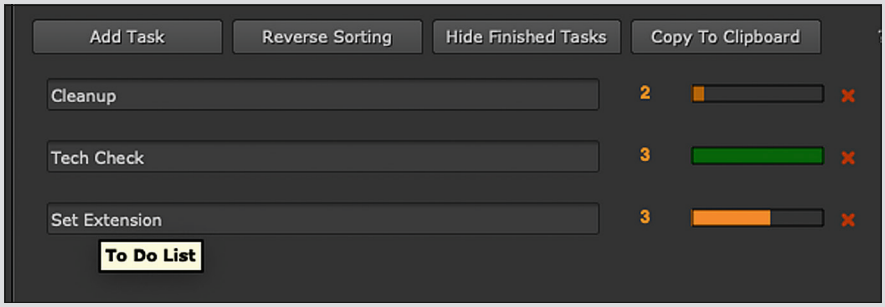
Zugegeben, wer Shotgrid, FTrack oder ein anderes Pipeline-Tool nutzt, wird mit der To-Do-List von Frank Rueter nicht so viel anfangen können. Wer aber nicht alle Nase lang in der E-Mail des Producers oder Supervisors nach dem nächsten Task suchen möchte, für den kann dieses kleine Tool eine echte Erleichterung sein. Das Python-Script lässt sich als eigene Pane in Nuke öffnen und erlaubt einem, sobald man sein Script das erste Mal gespeichert hat, eine kleine To-Do-Liste anzulegen.

Dabei kann man beliebig viele Tasks anlegen, sie priorisieren und einen Status von „waiting" zu „in progress" bis hin zu „finished" vergeben werden. Die Liste wird schlauerweise als separates XML File abgelegt und lässt sich auch sortieren und in die Zwischenablage kopieren.

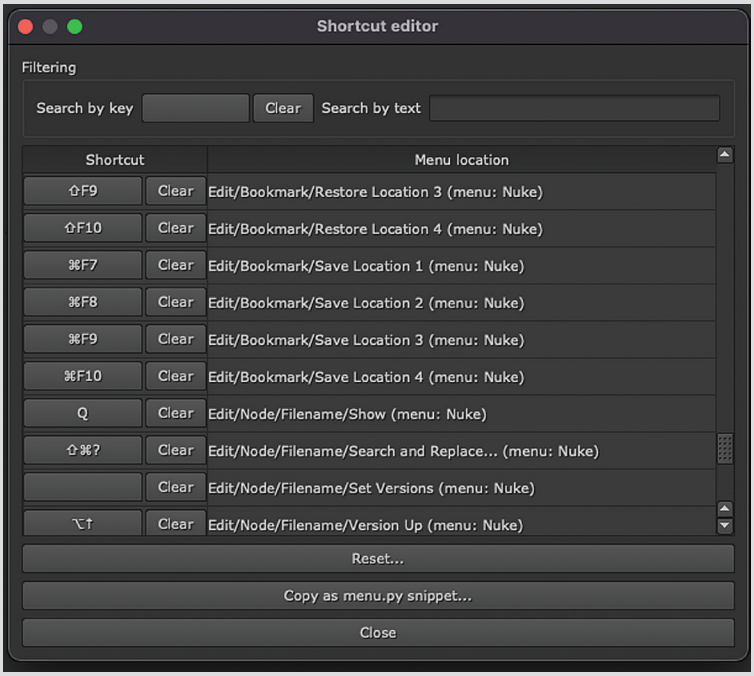
▷ **is.gd/nuke_todo**

Shortcut Editor

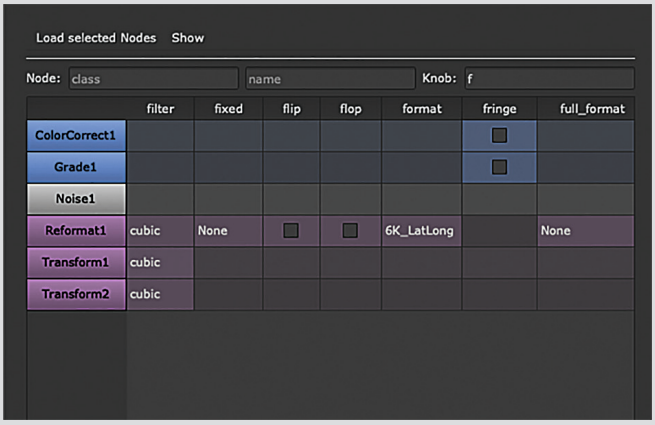
Es ist schon zum Haare raufen: Da hat man eine Software im – sagen wir es diplomatisch – oberen Preissegment, eine Software, die der Quasi-Standard für VFX-Bearbeitung ist und dann hat diese Software nicht mal einen Shortcut Editor. Die einzige Möglichkeit, in einem „Vanilla"-Nuke einen Shortcut zu definieren oder zu modifizieren, ist über Python. Machbar, aber alles andere als intuitiv. Ben



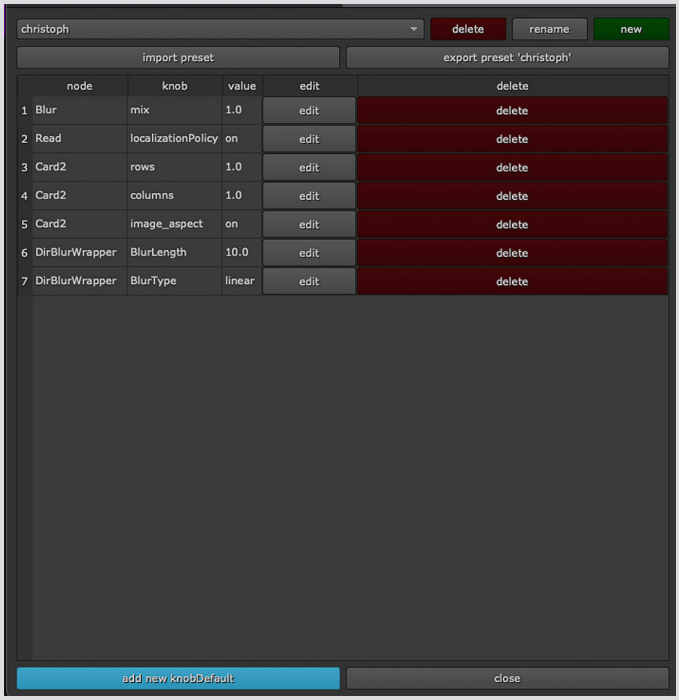
Für alle Listenfans – die ToDoList von Frank Rueter



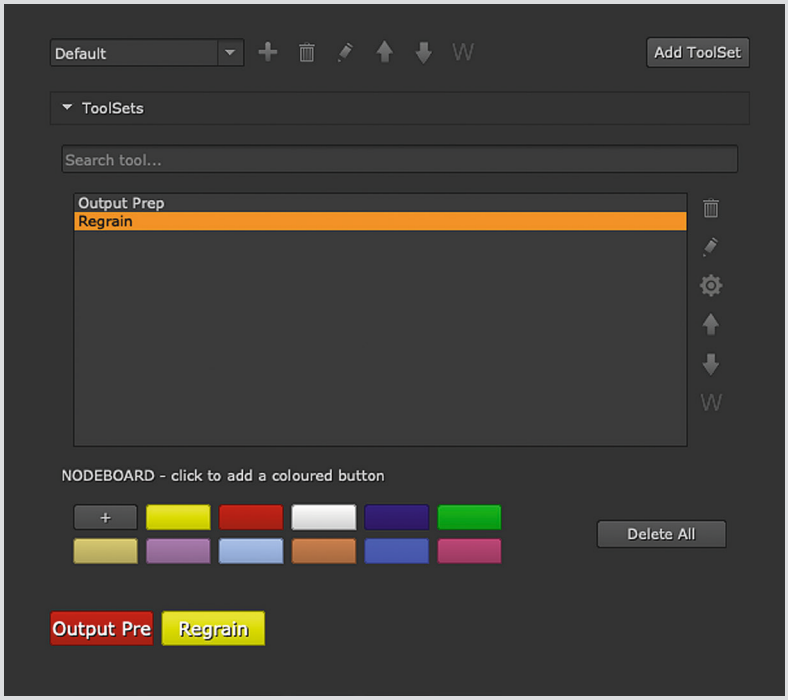
Sollte eigentlich zur Standardausstattung gehören – Der Shortcut Editor.



Und noch etwas für Listenfans – das Node Table von Mitja Müller-Jend und Marcel Goldschen-Ohm



Cragl'sTools: Neue Defaults für alle...



Nodeboard: „Toolsets for the Win!"

Dickson hat hier zum Glück Abhilfe geschaffen. Hat man sein Python Script einmal installiert, findet man im Menu Bar unter „Edit" den Eintrag „Edit Keyboard Shortcuts" und damit endlich den lang vermissten Editor. Und der lässt keine Wünsche offen. So lassen sich auch später installierten Plug-ins oder Gizmos Shortcuts zuweisen und man kann nach Shortcuts oder aber auch nach Funktionen suchen. Darüber hinaus gibt es Warnhinweise bei Konflikten zu bereits bestehenden Shortcuts. Und wer sich am Ende doch noch an Python rantrauen will, kann seinen neu kreierten Shortcut als Python Code kopieren – und so jemandem zur Verfügung stellen, der den Shortcut Editor nicht benutzt.

▷ **is.gd/nuke_shortcut_edit**

Node Table

Das Node Table lässt sich am besten als Listenansicht für den Node Tree umschreiben. Ist das Script einmal installiert, lassen sich beliebig viele Nodes aus dem Node Tree in die Node Table laden. Und dort? Dort werden all Parameter der Node, die sich sonst im Properties-Panel angezeigt werden, in einer Listenansicht dargestellt. Um den Überblick zu behalten, kann nicht nur nach der Art der Node, dem Namen, sondern auch vor allem nach speziellen Knobs gesucht werden. So kann man schnell mehreren Nodes, dieselbe Resolution, oder dasselbe Label verpassen, vielleicht aber auch einfach nur mehrere Nodes auf einen Schlag „disablen". Dies ist besonders praktisch, wenn die Nodes, die man anpassen oder abgleichen möchte, quer über das eigene Script verteilt sind. Was besonders gut gelöst ist: Hat man mehrere, unterschiedliche Nodes markiert, von denen einzelne einen bestimmten Parameter nicht unterstützen, werden diese Felder ausgegraut. Außerdem wird jede Node in der Liste mit der Ihr zugehörigen Farbe unterlegt, was die Navigation um einiges erleichtert.

▷ **is.gd/nuke_nodetable**

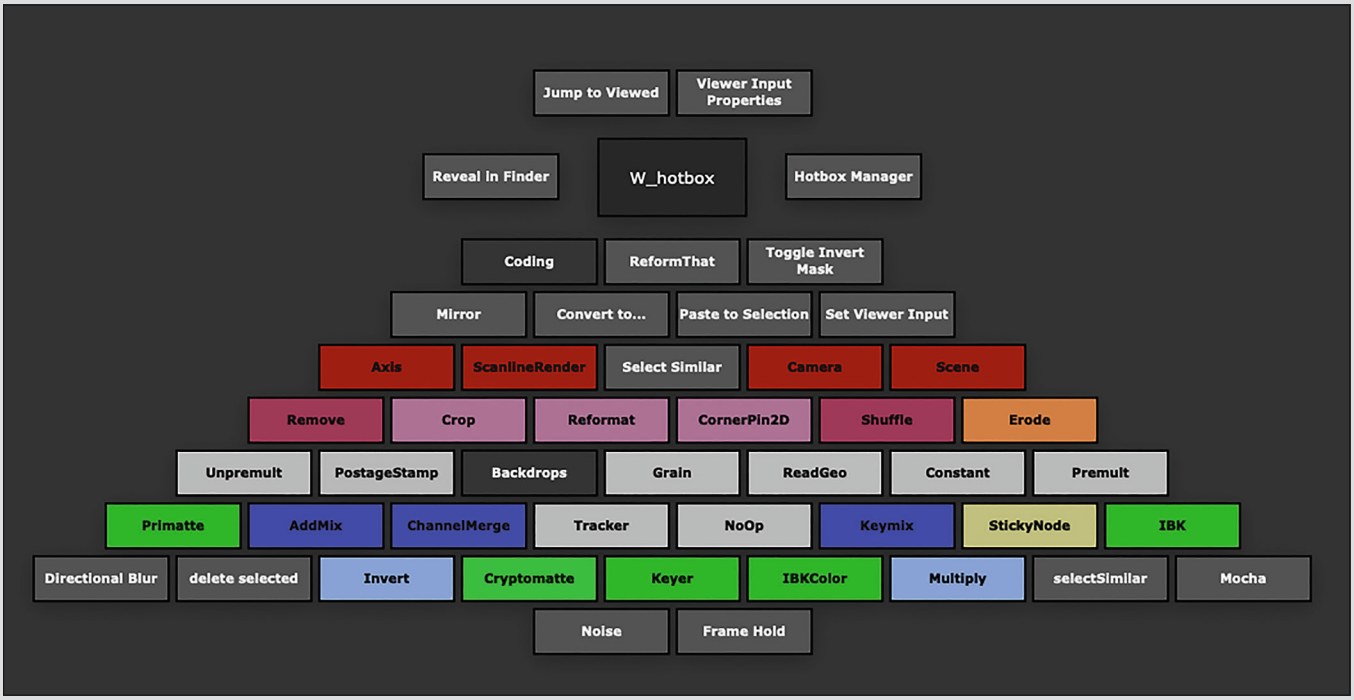
Defaults

Der durch seine Cragl-Tools in der Nuke-Community bekannte Simon Jokuschies hat sich mit Defaults eines Problems angenommen, welches jeder Artist sicherlich kennt: Die Default-Values, die sich die Entwickler bei Foundry irgendwann mal ausgedacht haben, sind einfach nie der Default-Wert, den man eigentlich braucht. Nehmen wir ein ganz einfaches Beispiel: Standardmäßig wird jede Card, die man in Nuke generiert, mit 8 Columns und 8 Rows generiert, obwohl man für klassische Projection Setups diese Unterteilung meist gar nicht braucht und die Geometrie nur unnötig komplex macht. Wenn man ehrlich ist, stellt man solche unnötig hohen Werte aber so gut wie nie auf ein vernünftiges Maß zurück. Hier greift „Default". In jedem numerischen Feld erschient nun beim Rechtsklick eine neue Option „set as new knob default". Einmal bestätigt, wird bei jeder neuen Node nun der zuvor eingestellte Wert als neuer Ausgangswert gesetzt. Und was ist mit nicht-numerischen Eingaben, wie Drop-Downs? Nun, diese können über die Option „show knob list" editiert werden. Ob man hier nun den Masken-Input einer Grade Node oder den Blur Type des Directional Blurs ändert, immer wiederkehrende Klicks kann man sich von nun sparen. Bevor die neuen Defaults nun aber die Kolleg:innen, mit denen man sich die Workstation teilt, in den Wahnsinn treiben, kommt „Default" mit mehreren Profilen daher, so dass ein jeder sich Nuke so einstellen kann, wie er mag. Und wenn die neuen Defaults einem doch nicht so zusagen, kann man sie auch ganz einfach wieder löschen.

▷ **is.gd/nuke_default_knobs**

W_Hotbox

Das wohl mächtigste Tool in dieser Liste dürfte die W_Hotbox von Wouter Gilsing sein. Maya-User kennen das Prinzip bereits: Per Hotkey erscheint rund um den Cursor ein radial angelegtes Menü, in dem sich Nodes generieren, aber auch Python Code ausgeführt und Parameter verändert werden können. Der Hotkey zum Aufrufen der Hotbox ist standardmäßig der Tilde-Key, kann aber nach der Installation in den Nuke-Preferences geändert werden. Die Hotbox



W_Hotbox: Radial-Menüs müssen nicht rund sein – sind aber meistens praktisch.

ist in drei Teile unterteilt: Unterhalb des Cursors findet man die Funktionen, die immer angeboten werden, egal, ob eine Node ausgewählt ist oder nicht. Dies ist also der prädestinierte Platz für Buttons, die eine neue Node erstellen sollen.

Links und Rechts des Cursors findet man dann Shortcuts zum Finder/Explorer und in den Hotbox Manager, der gleich noch wichtig werden wird. Oberhalb des Cursors befindet sich dann der kontextsensitive Teil der Hotbox. Hat man eine (oder mehrere) Nodes ausgewählt, erscheinen hier Optionen, die einem den Weg in das Properties Panel ersparen können. In der Grade Node kann man zum Beispiel den Clamp toggeln oder schnell den zu korrigierenden Channel wählen. In einer Read Node kann man die Project Frame Range an die Länge des Clips angleichen oder schnell auf eine andere Version wechseln. Das Geheimnis hinter all diesen Shortcuts heißt mal wieder: Python. Und hier kommt jetzt der Hotbox Manager ins Spiel, denn hier kann man neue Buttons für die Hotbox anlegen und ihnen Funktionen zuweisen.

Oben Links im Hotbox Manager hat man nun die Wahl zwischen „All", „Multiple" und „Single". „All" meint damit den Bereich der Hotbox unterhalb des Cursors, also zum Beispiel die Buttons für neue Nodes. Über das Plus-Icon kann man hier nun einen neuen Button anlegen und ihm einen Namen geben. Soweit so gut, doch leider ist der Button so noch ohne Funktion. Hier braucht es dann schon ein bisschen Python Code in der rechten Seite des Hotbox Managers. Wollen wir zum Beispiel einen neuen Button dazu bringen, die „Noise"-Node aufzurufen, wäre der entsprechende Code-Schnipsel:

```
nuke.createNode('Noise')
```

Der nächste große Bereich ist „Single", in dem die Funktionen der Hotbox oberhalb unseres Cursors untergebracht sind. Hier ändert sich das Procedere ein kleines bisschen, da wir ja zuerst definieren müssen, für welche Nodes diese Funktionen zur Verfügung stehen. Hier kann man eine Class anlegen, die den gleichen Namen wie die Node Class in Nuke hat (abrufbar über den Shortcut „I" bei selektierter Node). Ist dieses eingerichtet, kann man wie vorher Buttons erstellen und sie mit Python Code versehen. Je nachdem, welche Funtkionen man aufrufen möchte, wird es hier zunehmend komplexer. „Multiple schließlich sind Funktionen, die nur zur Verfügung stehen soll, wenn mehrere Nodes unterschiedlicher Classes gleichzeitig ausgewählt sind. Hier wird man ohne weitergehende Python-Kenntnisse kaum glücklich werden. Glücklicherweise liefert Wouter Gilsing eine ganze Library an Funktionen und Buttons mit. Allein schon mit dieser Library „Out-of-the-Box" ist die Hotbox ein wirklicher Timesaver. Aber die Library kann auch als guter Spickzettel dienen, um eigene Buttons zu basteln. Und ehe man sich versieht und ohne das man es eigentlich wollte, hat man so schon etwas Python gelernt ... Wenn einen das nicht abschreckt und man die nötige Frustrationstoleranz für ein bisschen Trial and Error hat, kann man sich die Hotbox hier zu einem richtigen Schweizer Taschenmesser für Nuke machen.

▷ **is.gd/nuke_hotbox**

Nodeboard

Wer gerne und viel mit Toolsets in Nuke arbeitet, wird das Nodeboard lieben. Nach der Installation erscheint das von Filip Suska geschriebene Tool als eigenes Panel. Hier können Toolsets in verschiedenen Gruppen, zum Beispiel projektbasiert abgelegt und auch ohne größeren Pipeline-Backbone mit Kollegen geteilt werden. Dabei wird einfach in den Nuke Preferences eine gemeinsame Save-Location festgelegt, auf die jeder Zugriff hat. Zwei Features machen das Nodeboard allerdings noch sehr viel mächtiger als die hauseigenen Toolsets von Nuke. Zum einen können ganze Toolset-Gruppen als auch individuelle Toolsets direkt als Button an die W_Hotbox gesendet werden, zum anderen können Toolsets mit einem Klick auf das Zahnrad Live geändert und aktualisiert werden, ohne das alte Toolsets gelöscht oder umbenannt werden müssen.

▷ **is.gd/nuke_nodeboard** **›ei**