

Teil 3

Gizmo-Integrierung in Nukes UI

Im dritten und letzten Teil der Gizmo-Workshop-Reihe zeigt Simon Jokuschies, wie Sie die beiden selbst erstellten „HighpassSharpen“ und „TheGlow“ Gizmos in die Benutzeroberfläche von Nuke integrieren und eine sinnvolle Ordnerstruktur anlegen.

von Simon Jokuschies

Ein paar Zeilen Quelltext in Python reichen aus, um die erstellten Gizmos dauerhaft in Nukes Benutzeroberfläche zu integrieren. Für die individuelle Konfiguration von Nuke steht die sogenannte „Nuke Home Directory“ bereit. Dies ist ein Ordner, der bei der Installation von Nuke automatisch angelegt wird. In diesem können eigene Elemente, wie beispielsweise Gizmos, Python-Skripte und vieles mehr, abgelegt werden. Dadurch lässt sich Nuke grundlegend verändern und erweitern. Je nach verwendetem Betriebssystem variiert der Pfad zur „Nuke Home Directory“ wie folgt:

```
Linux: /home/username/.nuke
Mac: /Users/username/.nuke
Windows: C:\Users\username\.nuke
```

Ordner, die mit einem „.“ beginnen sind unter Linux und Mac unsichtbar, daher wird die „Nuke Home Directory“ für diejenigen Nutzer im System höchstwahrscheinlich nicht angezeigt. Um diesen Ordner zu erreichen, muss der Ordnerpfad entweder über die Kommandozeile manuell eingegeben oder es müssen per Terminal-Befehl alle unsichtbaren Dateien angezeigt werden.

Beim Starten von Nuke werden die Dateien, die sich in der „Nuke Home Directory“ befinden, geladen – dies geschieht jedoch nicht automatisch. Per Python-Befehl muss die Anweisung dazu gegeben werden.

Hierfür kommen die „init.py“ und die „menu.py“ ins Spiel; dabei handelt es sich, wie aus der Dateiendung „.py“ erkennbar ist,

um Python-Dateien. Alle Befehle innerhalb der „init.py“ sind Anweisungen, die ausgeführt werden, bevor Nukes Benutzeroberfläche geladen ist. Die „init.py“ eignet sich beispielsweise zum Setzen eigener Plug-in-Pfade oder zum Festlegen von Standard-Knob-Werten beim Erstellen neuer Nodes. Die Befehle in der „menu.py“ sind Anweisungen, die ausgeführt werden, nachdem Nukes Benutzeroberfläche geladen wurde.

Diese Datei kann beispielsweise für die Erstellung eigener Menüeinträge und dem Hinzufügen grafischer Benutzeroberflächen dienen. Beide Dateien werden von Nuke automatisch geladen; im Folgenden werden sie dafür benutzt, um ein eigenes Node-Menü zu realisieren.

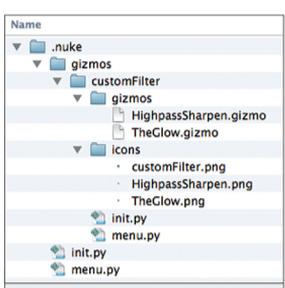


Abb. 28. Strukturiert – Die „Nuke Home Directory“ enthält nun eine geordnete Gizmo-Bibliothek.

Ordnerstruktur erstellen

Es ist möglich, die beiden erstellten Gizmos direkt in der „Nuke Home Directory“ an oberster Ebene zu kopieren und zu laden – jedoch entsteht hierdurch sehr schnell eine unüberschaubare Datensammlung. Damit die „Nuke Home Directory“ übersichtlich bleibt, sollte man sich eine sinnvolle Ordnerstruktur überlegen, mit der sich die eigenen Elemente sortiert ablegen lassen. Hierfür wird in der „Nuke Home Directory“ zunächst ein Ordner mit dem Namen „Gizmos“ erstellt – in diesem werden später alle eigenen Gizmos gespeichert.

Weil sich in Nuke mehrere eigene Node-Menüs kreieren lassen, wird zur besseren Übersichtlichkeit pro Node-Menü ein Unterordner erstellt. Da beide erstellten Gizmos in die Kategorie „Filter“ fallen, wird ein eigenes Node-Menü mit dem Namen „customFilter“ erstellt. Der Unterordner erhält den gleichen Namen „Filter“.

In Nukes Benutzeroberfläche können Node-Menüs und Gizmos Icons tragen, aus diesem Grund werden die Gizmos und Icons getrennt voneinander gespeichert. In dem „customFilter“-Ordner werden daher mit „Gizmos“ und „Icons“ zwei Unterordner erstellt. Als Icons können für das „customFilter“-Menü und die beiden Gizmos individuelle .png-Bilddateien mit einer festen Größe von 32 x 32 Pixeln erstellt werden, diese legen Sie daraufhin in den „Icons“-Ordner ab.

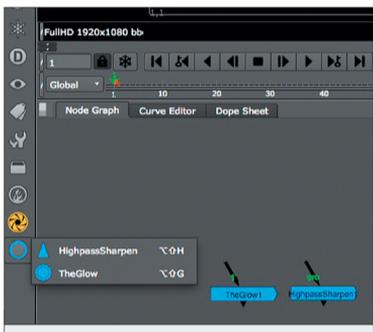


Abb. 29. Individualisiert – Nukes Node-Menü enthält nun ein eigenes Node-Menü „customFilter“ mit den beiden erstellten Gizmos.

Die beiden exportierten Gizmos „HighpassSharpen“ und „TheGlow“ kopieren Sie danach in den „Gizmos“-Ordner. Beim Starten von Nuke werden diese Elemente jedoch noch nicht geladen, denn was noch fehlt, ist der Befehl an Nuke, diese zu laden. Zunächst wird hierfür in der „init.py“, die sich in der „Nuke Home Directory“ befindet, eine Zeile hinzugefügt. Falls diese Datei noch nicht existiert, kann sie manuell mit einem Texteditor erstellt und als „init.py“ in der „Nuke Home Directory“ auf oberster Ebene abgespeichert werden. Die „init.py“ erhält folgende Zeile:

```
nuke.pluginAddPath("gizmos/customFilter")
```

Hiermit wird der von der „init.py“ aus relative gelegene Pfad „gizmos/customFilter“ beim Systemstart hinzugefügt und alle in diesem Ordner enthaltenen Befehle können daraufhin beim Systemstart ausgeführt werden. Mit dem Hinzufügen des Pfads zum „PluginPath“ lädt Nuke die Gizmos jedoch noch nicht. Im Ordner „customFilter“ wird nun eine weitere „init.py“-Datei erstellt, diese trägt die beiden Zeilen:

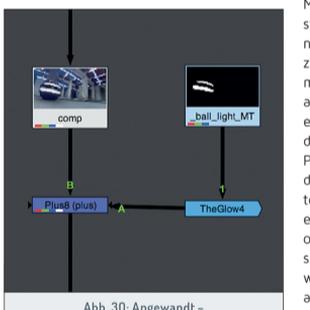


Abb. 30. Angewandt – Der „TheGlow“ Gizmo in der Praxis

```
nuke.pluginAddPath("gizmos")
nuke.pluginAddPath("icons")
```

Mit diesen Zeilen werden dem „PluginPath“ zusätzlich die Unterordner „Gizmos“ und „Icons“ aus dem Ordner „customFilter“ hinzugefügt – nun werden die Ordnerinhalte beim Starten von Nuke auch geladen. Dies lässt sich mit einem Befehl in Python überprüfen. Alle Ordner, die in Nukes „PluginPath“ enthalten sind, lassen sich anzeigen. Hierfür dient folgende Zeile, die in Nukes Scripteditor eingegeben werden kann:

```
nuke.pluginPath()
```

Nachdem die Unterordner „Gizmos“ und „Icons“ nun erfolgreich zu Nukes „PluginPath“ hinzugefügt worden sind, muss nun abschließend der Befehl zum Einbinden der Gizmos in die Benutzeroberfläche erfolgen. Dies wird in der eigens erstellten Python-Datei „menu.py“ in dem Unterordner „customFilter“ realisiert. Da Python auf Zeilenumbrüche achtet, ist jeder mit einem Umbruchsymbol (↵) markiert – das Layout verhindert lange Zeilen. Sie trägt folgende Zeilen:

```
custom_menu = nuke.menu("Nodes").
addMenu("customFilter",
icon="customFilter.png")
custom_menu.
addCommand("HighpassSharpen", "nuke.
createNode("HighpassSharpen"),
"Shift+Alt+H", icon="HighpassSharpen.png")
custom_menu.addCommand("TheGlow", "nuke.
createNode("TheGlow"), "Shift+Alt+G",
icon="TheGlow.png")
```

In der ersten Zeile wird das eigene Node-Menü erstellt und in der Variablen „custom_menu“ gespeichert, um dieses generierte Objekt im Folgenden adressierbar zu machen. Mit dem Ausdruck „nuke.menu("Nodes")“ wird Nukes Node-Menü aufgerufen. Der Befehl „addMenu()“ erstellt ein neues Node-Menü. Der Methode werden zwei Parameter hinzugefügt: Der erste Parameter legt den Namen des Menüs, in diesem Fall „customFilter“, fest. Der zweite Parameter „icon="customFilter.png“ ist ein sogenanntes „Keyword Argument“, das optional festgelegt werden kann. Mit diesem wird dem Node-Menü ein Icon zugewiesen. Nuke sucht daraufhin beim Starten alle Ordner seiner „PluginPath“-Liste nach der Datei „customFilter.png“ ab und wird im manuell hinzugefügten Ordner „Icons“

BPM BROADCAST & PROFESSIONAL MEDIA GMBH
www.bpm-media.de
Obenhauptstr. 15,
22335 Hamburg //
Tel. 040/557624-0
// eMail info@bpm-media.de
VIDEO & NLE HARD- UND SOFTWARE

d3d dreidesign
3D Visualisierung | Animation | Architekturpräsentation
www.dreidesign.com

CADNETWORK GMBH
www.CADnetwork.de
Premium. Performance. Innovation.
Emil-Hoffmann-
Straße 55-59,
D - 50995 Köln-Rodenkirchen,
Tel.: 02236/99379-10, Fax: 02236/99379-29,
eMail: info@cadnetwork.de
WORKSTATION & RENDERLÖSUNGEN

BRANDNEW ENTERTAINMENT GMBH
www.brandnew-entertainment.de
Tel.: 0208 - 377 11 190
Zum Aquarium 6a
45047 Oberhausen
eMail: mail@brandnew-entertainment.de
Kontakt: Björn Kleiber
FULL SERVICE FILM- & POSTPRODUCTION

FORMVOLLENDUNG
In Form gebracht – an modernsten Schnittplätzen
MAGENTA TV
Fernsehproduktionsgesellschaft
Kaiser-Friedrich-Ring 66
65105 Wiesbaden
0611 / 93 20 96 0
www.magenta-tv.de

DP Abo verschenken!
digitalproduction.com/geschenkaboo

DP BUSINESS CORNER
Ihr Unternehmen oder Ihr Produkt kurz & auffällig platziert
Verfügbare Anzeigengrößen:
• 53mm Breite x 35mm Höhe
• 53mm Breite x 70mm Höhe
Jetzt buchen!
Kontakt: sales@digitalproduction.com

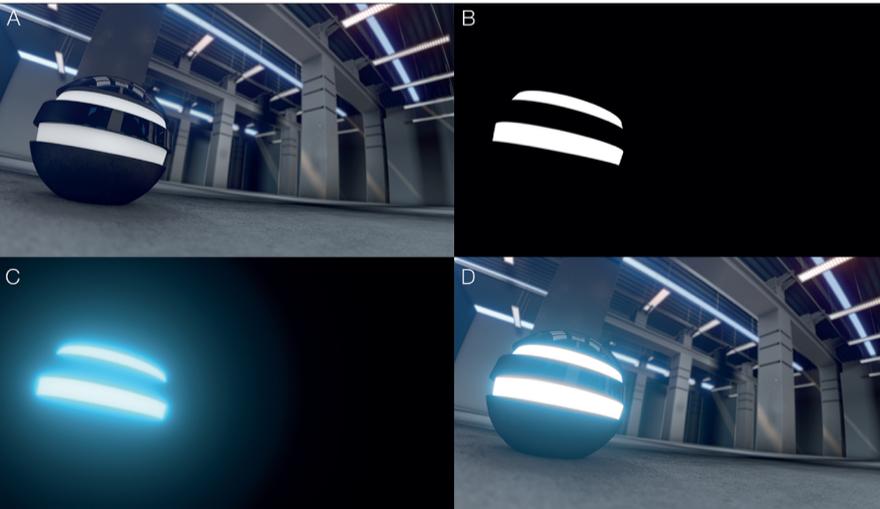


Abb. 31: Angewandt – Das Bild zu einem bestimmten Zeitpunkt in der Comp (A). Die separate Matte (B) erhält den Glow-Effekt (C). Dieser Effekt lässt sich über einen Merge Node mit dem Blending-Modus „Plus“ hinzufügen (D).

des Unterordners „customFilter“ fündig. Die zweite und dritte Zeile fügt die Gizmos zum eigenen Node-Menü hinzu. Hierfür wird das „custom_menu“-Objekt aufgerufen und mit dem Befehl „addCommand“ ein neuer Eintrag hinzugefügt. Dabei werden vier Parameter übermittelt: Der erste Parameter „HighpassSharpen“ entspricht dem Namen, so wie er in der Benutzeroberfläche erscheint. Der zweite Parameter stellt den Befehl dar, der ausgeführt wird, sobald dieser Eintrag im Node-Menü angeklickt wird. Hierbei wird mit „nuke.createNode()“ ein Node, in diesem Fall ein Gizmo, im Node Graph erstellt. Dieser Funktionsaufruf erwartet als Parameter die Node-Klasse, von welcher ein Objekt kreiert werden soll – in diesem Beispiel stellt die Node-Klasse den Namen des Gizmos, also „HighpassSharpen“, dar. Hierbei durchsucht das Programm beim Aufruf des Befehls alle Ordner seiner „PluginPath“-Liste und wird im manuell hinzugefügten Unterordner „Gizmos“ des Ordners „customFilter“ fündig. Der dritte Parameter ist optional und stellt mit „Shift+Alt+H“ ein Tastaturkürzel zum Anlegen des Gizmos dar. Der vierte, ebenfalls optionale Parameter legt analog zum Erstellen von „custom_menu“ ein Icon in der Benutzeroberfläche fest. Die „Nuke Home Directory“ enthält nun neben den bereits existierenden Elementen die in Abbildung 28 angezeigten Objekte. Beim Neustart von Nuke wird die im Ordner „customFilter“ enthaltene „menu.py“ automatisch gelesen

und ausgewertet und somit das selbst erstellte Node-Menü mit seinen Gizmos geladen (Abb. 29). Da der Gizmo-Ordner in der „Nuke Home Directory“ strukturiert aufgesetzt wurde, lassen sich nun auf einfache Art neue Gizmos hinzufügen. Hierzu muss lediglich der Gizmo in den Unterordner „Gizmos“ und optional ein Icon in den Unterordner „Icons“ kopiert werden. In der „menu.py“ kann daraufhin eine weitere Zeile zum Einbinden in die Benutzeroberfläche, analog zu den letzten beiden Zeilen, hinzugefügt werden. Um ein weiteres Node-Menü anzulegen, lässt sich der „customFilter“-Ordner kopieren und entsprechend umbenennen. Daraufhin können die Unterordner „Gizmos“ und „Icons“ gelehrt und mit eigenen Elementen gefüllt werden. Danach fügen Sie in der „init.py“ der „Nuke Home Directory“ den neuen Ordner für das Node-Menü zum „PluginPath“ hinzu. Abschließend lässt sich nun die „menu.py“ des neuen Node-Menüs mit den neuen Werten entsprechend ändern. Auf diese Weise entsteht eine strukturierte Gizmo-Bibliothek. Das „TheGlow“ Gizmo wurde für das Titelbild der Workshop-Reihe wie folgt angewandt: Der helle Teil der futuristischen Kugel, der leuchten soll, erhielt im 3D-Rendering eine Matte. Auf dieses Bild wurde der „TheGlow“-Node angewandt. Das Ergebnis habe ich daraufhin mit einem Merge Node im Blending-Modus „Plus“ über das Originalbild gelegt (Abb. 30). Die Abbildung 31 zeigt die einzelnen Schritte hierfür.

Gizmos in Gruppen konvertieren

Gizmos bieten eine praktische Art, gruppierte Nodes zu exportieren und mit anderen Nutzern zu teilen. Ein wichtiger Aspekt sollte jedoch beachtet werden: Sobald in einem Nuke-Skript ein Gizmo verwendet wird, installiert sich das eigene Computer nicht, welches auf dem eigenen Computer nicht installiert ist, erzeugt Nuke eine Fehlermeldung.

Diese Fehlermeldung tritt allerdings nicht auf, sofern es sich um einen Group Node handelt. Es ist von daher generell sinnvoll, alle Gizmos in einem Nuke-Skript stets über die Schaltfläche „Copy to Group“ im Tab „Node“ in Gruppen zu konvertieren. Hierbei bleiben alle Funktionalitäten und sogar die eingestellten Knob-Werte der Gizmos erhalten. Die Gizmos kreieren lediglich ein Abbild ihrer Funktionen und erlangen ihren Ausgangszustand als Group Nodes zurück. > mf

Simon Jokuschies (www.leafpictures.de) studierte Informationstechnologie und Gestaltung an der Fachhochschule Lübeck sowie ein Auslandssemester Digital Media an der Auckland University of Technology in Neuseeland. Er arbeitet bei der nhb Video GmbH im Bereich der Postproduktion als Compositor/Compositing TD. Seine Schwerpunkte liegen in den Bereichen Compositing, 3D und Programmierung. Zudem ist er Gründer von Gragl – VFX Tools (www.cragl.com) sowie von pixilati, dem Onlinetzwerk für Mediendesigner und Kreative (www.pixilati.de).