

2017

ISSN 1433-2620 > B 43362 >> 21. Jahrgang >>> www.digitalproduction.com

Published by **A TEC**

Deutschland € 17,70

Österreich € 19,-

Schweiz sfr 23,-

1

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

JANUAR | FEBRUAR 01:2017



CG-Characters!

Im Fokus: Helden-Design und Statisten-Creation

Maya Bifrost

Neue Wassersimulation – bis zum letzten Schluck

Neue Tools

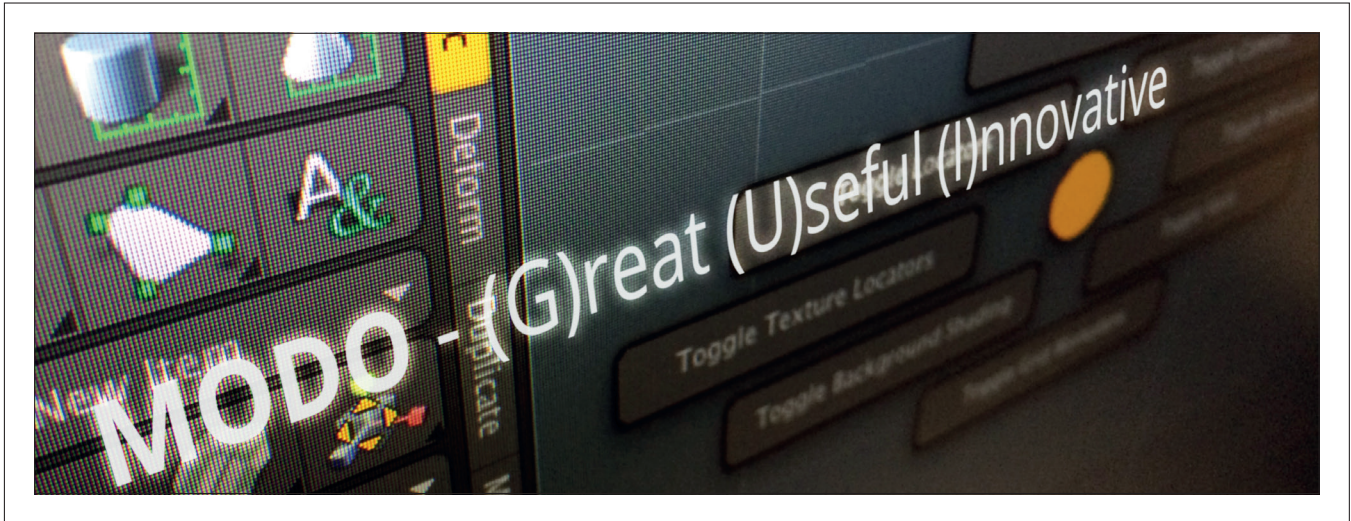
ZBrushCore, Trapcode, Blender 2.78 & mehr



4 194336 217709



01 >



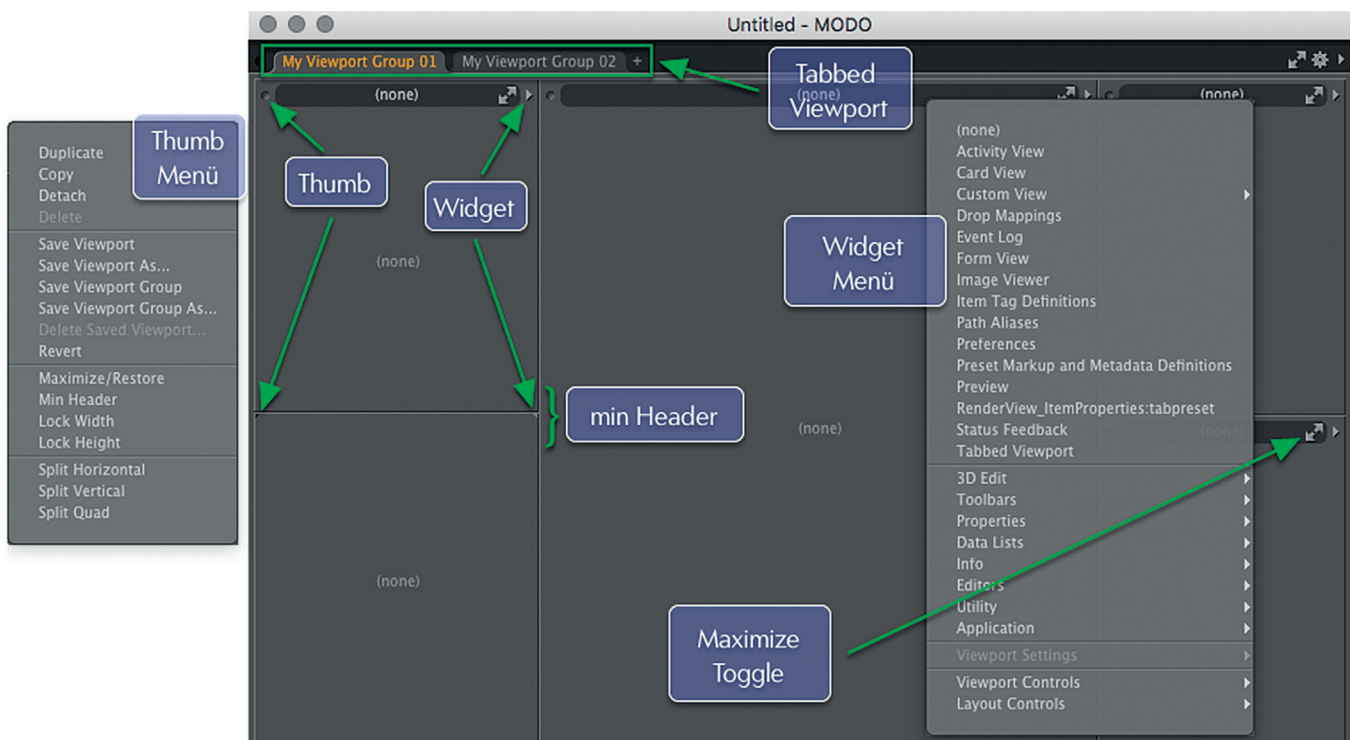
UI, Modo!

Für ein effizientes Arbeiten mit einer Applikation stellt sich immer die gleiche Frage: Wie weit ist es sinnvoll, sie zu individualisieren? Eine Anpassung an die eigenen Gepflogenheiten optimiert mit Sicherheit den Workflow – aber was ist bei der Arbeit mit Kollegen, wie sieht es bei dem nächste Release aus, und was ist, wenn man mal ein Tutorial machen möchte und die eigene Oberfläche so gar nichts mehr mit dem Original zu tun hat? von André Löscher

Fakt ist, dass die Flexibilität, die Modo bietet, nicht dazu verleiten lassen sollte, einfach alles auf den Kopf zu stellen. Mannigfaltige Möglichkeiten bedeuten nicht automatisch mehr Effizienz, denn das Layout-System und die Tool-Pipeline in Modo sind gut durchdacht und funktionieren (vorausgesetzt, man lässt sich darauf ein) sehr effektiv.

Grundlegend hat Modo diesen Gedanken schon jeher in den Layouts umgesetzt (via Ctrl+Tab mit dem Layout-Switcher), die je nach Aufgabenbereich des UI (inklusive Viewport, Tastebord-Layout und Commands) organisieren und von nicht benötigten Elementen befreien. Diese lassen sich aber trotzdem nach den eigenen Anforder-

ungen anpassen und auch speichern, so dass die eigene Vorstellung des Modo Layouts hier permanent hinterlegt werden kann. Überzeugen kann Modos GUI weiterhin, da sie sich (in den allermeisten Fällen) in allen Bereichen auf die gleichen Konventionen stützt, sodass einmal Gelerntes immer wieder verwendet werden kann.

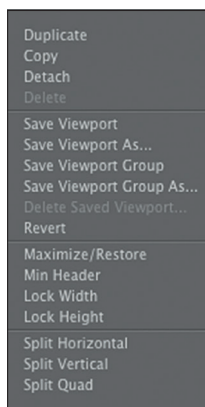


Zu empfehlen ist hierbei auf jeden Fall, in der Modo-Hilfe unter The Modo Interface\Customizing your Layout das kleine Tutorial anzuschauen, in dem der Model View von Grund auf erstellt wird. <http://bit.ly/ModoCustLay>

Auf dem Bau – der eigene Viewport

Um das Interface und die Funktionalität erweitern zu können, sollte man sich zuerst mit den nötigen Schaltern vertraut machen. In Modo wird der grundlegende Aufbau eines Views durch Fenster definiert, die nach Belieben mit Elementen gefüllt werden.

Hierbei können bestehende Fenster verwendet oder erweitert werden, oder man füllt ein komplett leeres Fenster (Layout > Windows > New Window) mit Inhalt. Für die Anpassung der Fenster ist einerseits der kleine Knopf (Thumb) in der oberen linken Ecke jedes Fensters zuständig, andererseits kann man auch das kleine nach rechts zeigende Dreieck (Widget) bzw. in der minimierten Version (Min Header) nur noch eine kleine dunkelgraue Ecke links und eine hellgraue Ecke rechts verwenden. Mit RMB auf den



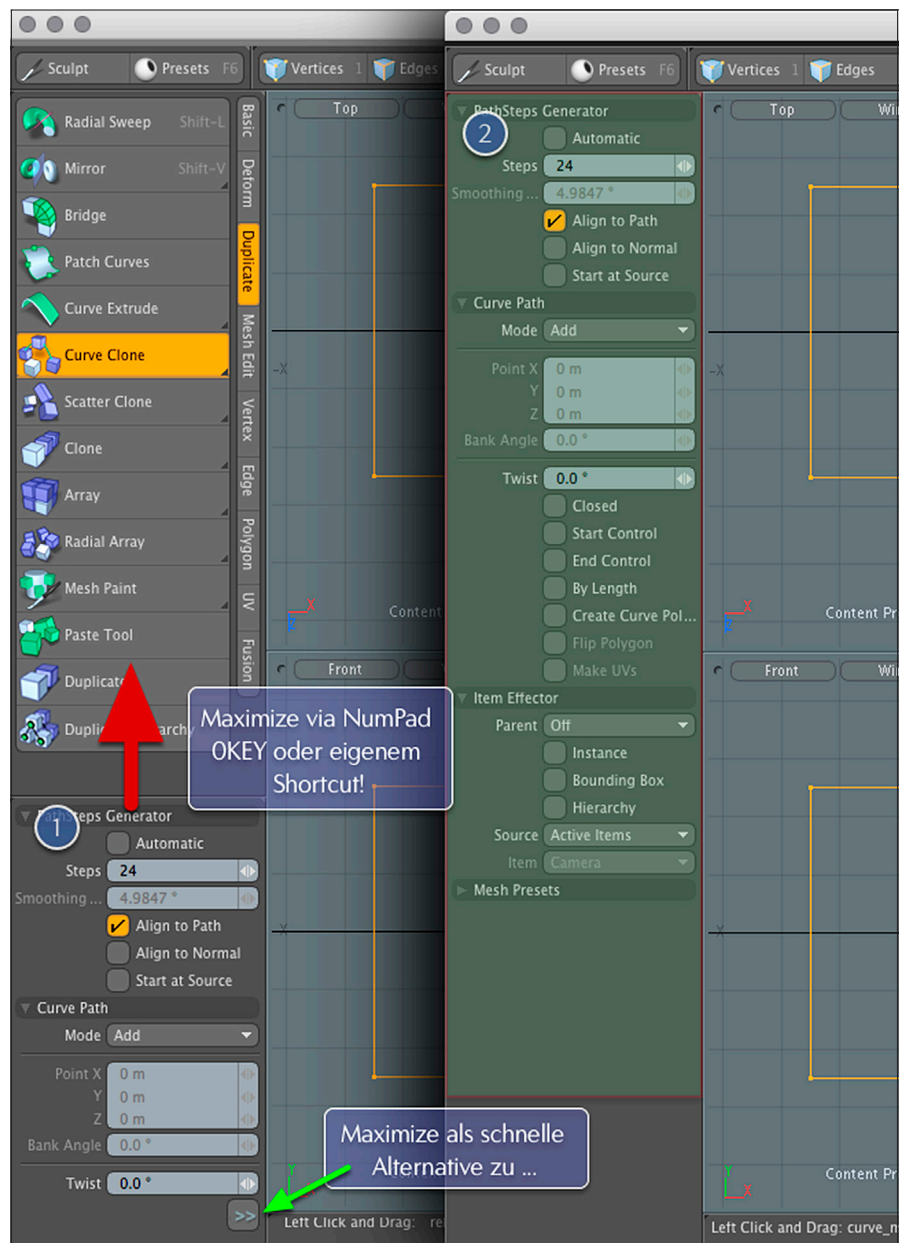
Thumb erhält man die Optionen zur Steuerung der Darstellung des Fensters. Hier kann das Fenster dupliziert, abgekoppelt oder gelöscht werden, man kann das gesamte Layout des Viewports oder der Gruppen speichern, und Fenster können horizontal und vertikal geteilt werden (schneller:

Ctrl+vertical/horizontal Drag auf Thumb um vertikal/horizontal zu teilen).

Die eher unscheinbare Option „Min Header“ schafft bei der Vielzahl von Fenstern erstaunlich viel Platz durch einfaches Ausblenden des Kopfbereichs. Mit LMB und Drag kann die Viewport Group an eine andere Position des Layouts verschoben werden.

Mögliche Ziele werden hier während der Bewegung durch orange Linien markiert. Das Widget bietet einem alle möglichen Inhalte für das gewählte Fenster. Zur besseren Übersicht sind diese (teilweise) in Kategorien eingeteilt. Zu beachten ist hierbei, dass einige Einträge wiederum Gruppen enthalten, sodass bei Auswahl der entsprechende Fensterbereich unter Umständen mit mehr als einem neuen Fenster gefüllt wird.

Die Basis eines leeren Fensters (im Widget: none) ist eine Viewport Group (im Widget: falls nicht im Root, dann unter Application), die wiederum mit beliebigen Inhalten gefüllt werden kann. Der einfachste Weg hierzu ist die Auswahl eines vordefinierten Inhalts aus dem Widget-Menü. Sollte dies einmal kein passgenaues Ergebnis liefern, kann zum Beispiel via Form View und dann erneut via Widget > Viewport



Das sinnvolle Eingliedern in Viewport Groups vergrößert mithilfe des Maximize/Restore Toggles den Viewport enorm.

Settings mehr oder weniger auf jegliche Tabs, Buttons, Pop-ups etc. (sogar der installierten Plug-ins) zurückgegriffen werden, um einen gänzlich individuellen Inhalt zu kreieren. Hierbei ist die schiere Masse an Auswahlmöglichkeiten wahrscheinlich die größte Barriere.

Um das Fenster-Management nun gedanklich abzuschließen, kann man via Tabbed Viewport auf der ersten Hierarchieebene beliebig viele eigene Layouts erstellen, sodass keine Wünsche offen bleiben dürften.

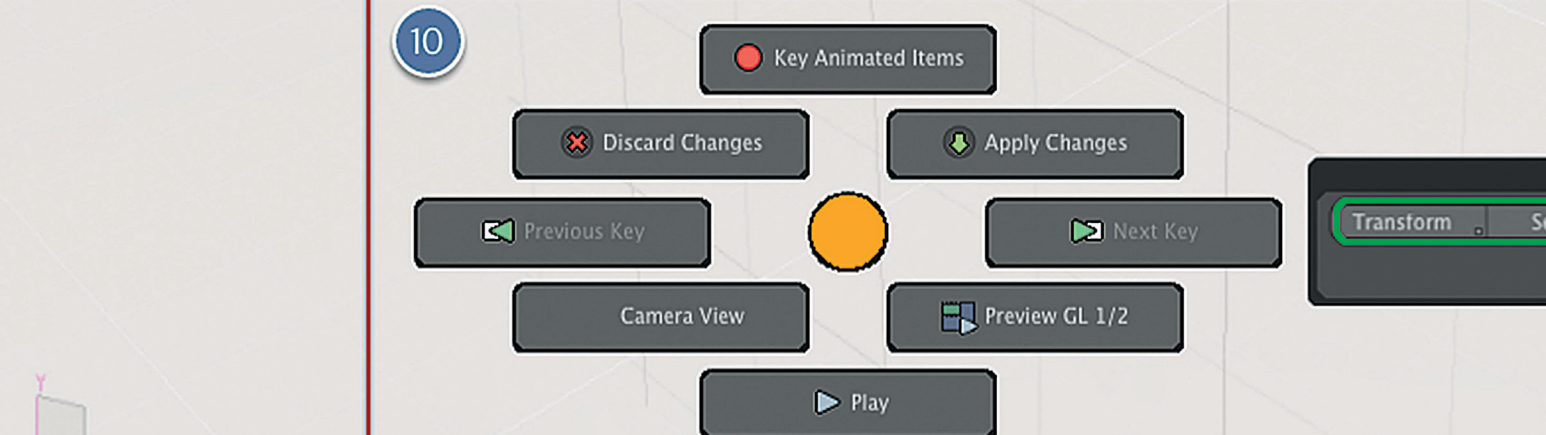
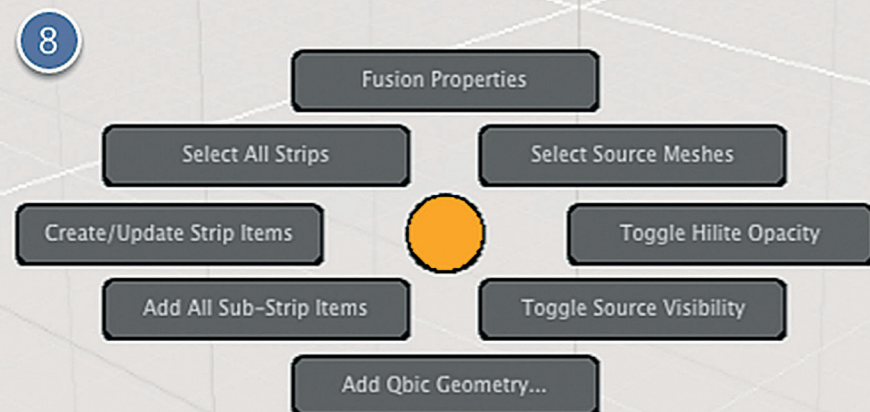
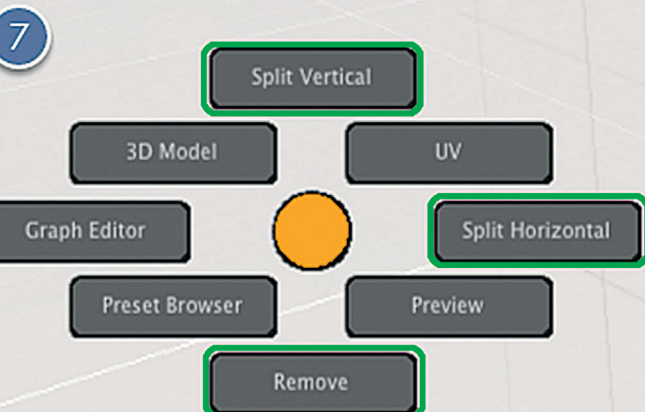
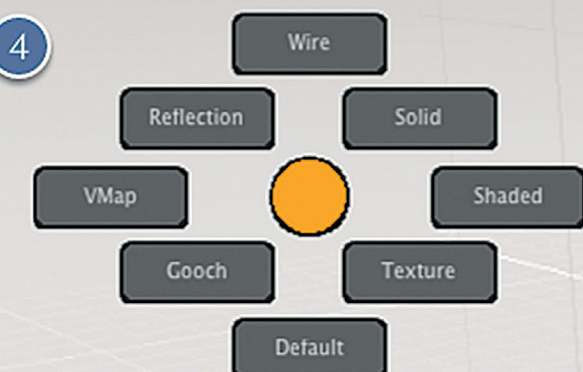
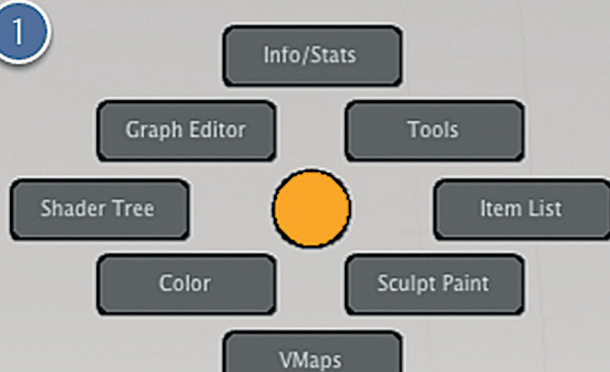
Interaktion – Arbeiten zwischen den Fenstern

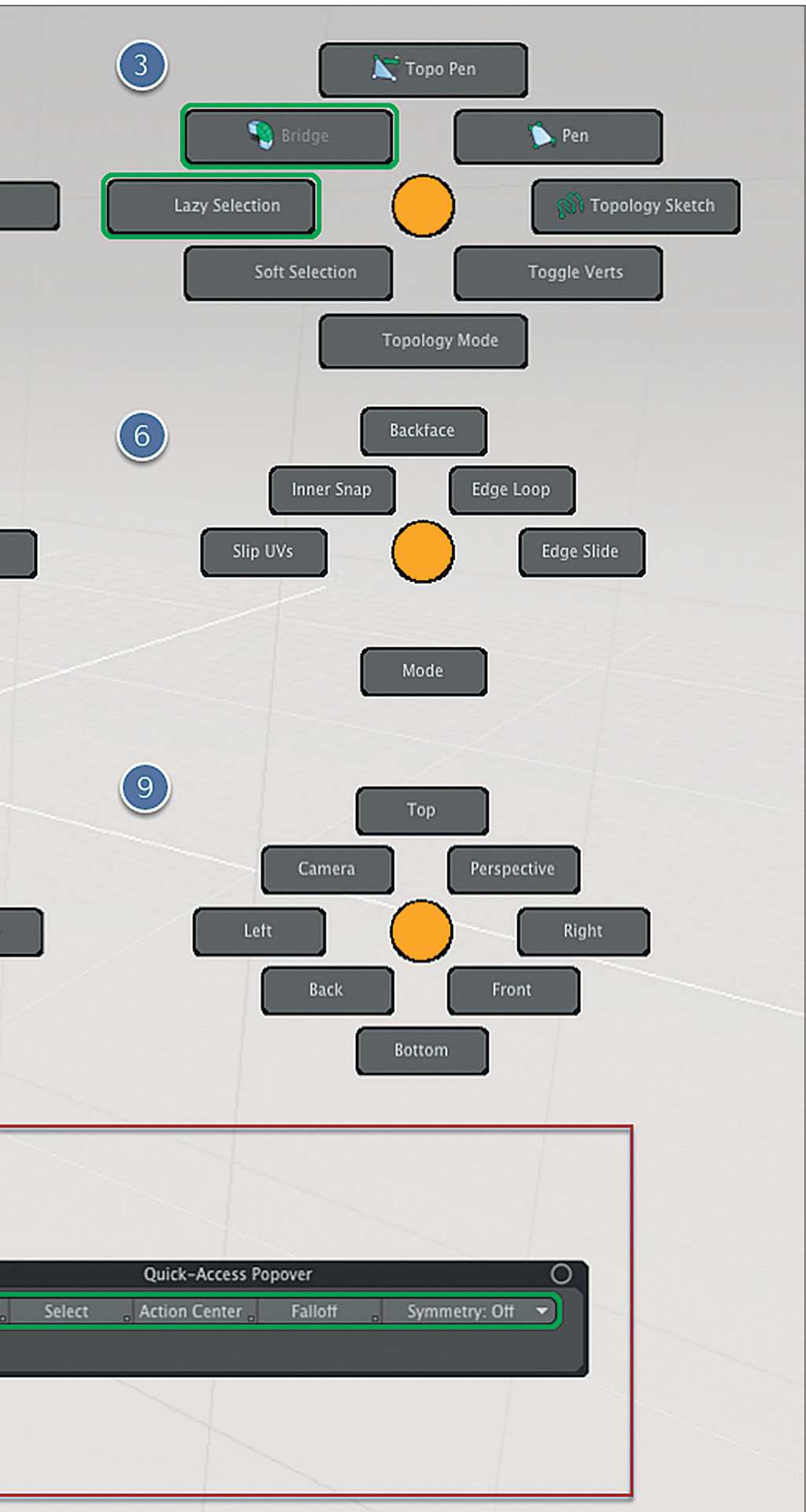
Hat man nun den optimalen Viewport gefunden, stellt sich die Frage nach dem effizienten Arbeiten mit den Elementen. Hierbei ist zuerst eine aus den Optionen bekannte

Funktion elementar wichtig: Maximize/Restore. Trotz der Separation einzelner Funktionen in die Layouts ist der Viewport je nach Monitorgröße umfangreich gefüllt und der Raum zum Arbeiten oder zur Wertänderung wird unter Umständen stark begrenzt.

Dies wird häufig sichtbar durch den kleinen Doppelpfeil (>>), der am unteren Rand eines Views auf Inhalt verweist, der keinen Platz mehr findet. Via „Maximize“ lässt sich jeder View in Modo extrem schnell auf die maximale Größe bringen, noch schneller geht es mit dem Shortcut: „O“-Taste mit dem Cursor über dem gewünschten View. So lässt sich in den meisten Fällen viel einfacher mit zu wenig Platz umgehen, als per Klick auf den Doppelpfeil oder das Verschieben von Viewport-Kanten.

Und hier bekommen jetzt auch die Viewport Groups eine entscheidende Rol-



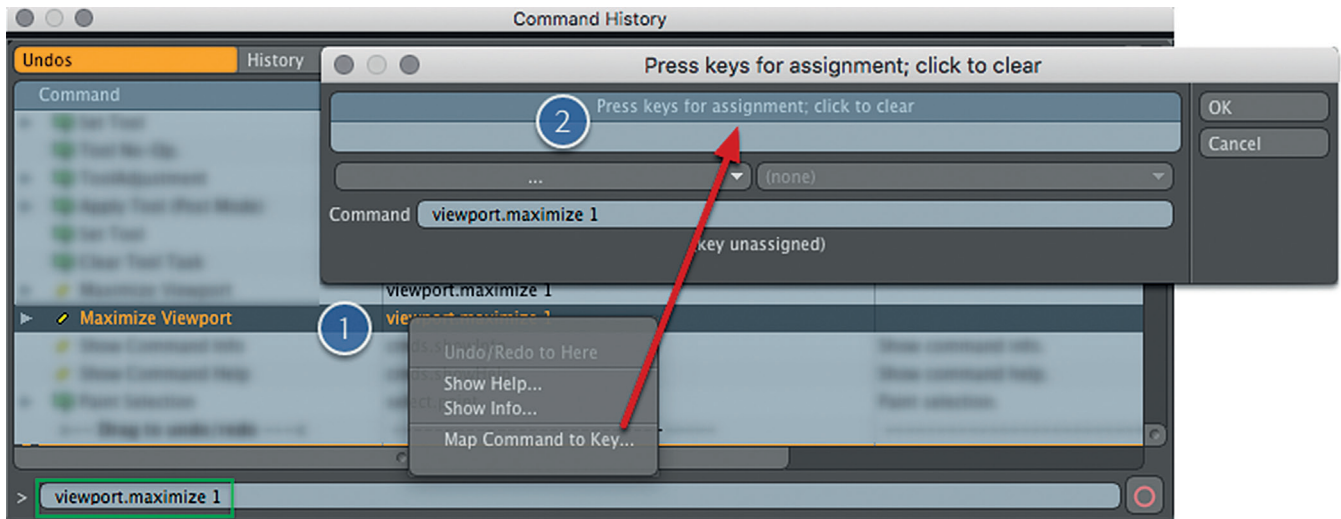


le. Würde „Maximize“ ein Fenster immer auf die volle Größe maximieren, wäre ein sinnvolles Arbeiten nicht mehr möglich. Somit reicht „Maximize“ immer nur bis auf die Hierarchie-Ebene der Viewport Group. Soll heißen: Es ist sinnvoll, den Model-Bereich und den Datenbereich jeweils übergreifend in eine Viewport Group zu setzen, damit das Maximieren eines Datenfeldes nicht in den Model-Bereich eingreift und umgekehrt.

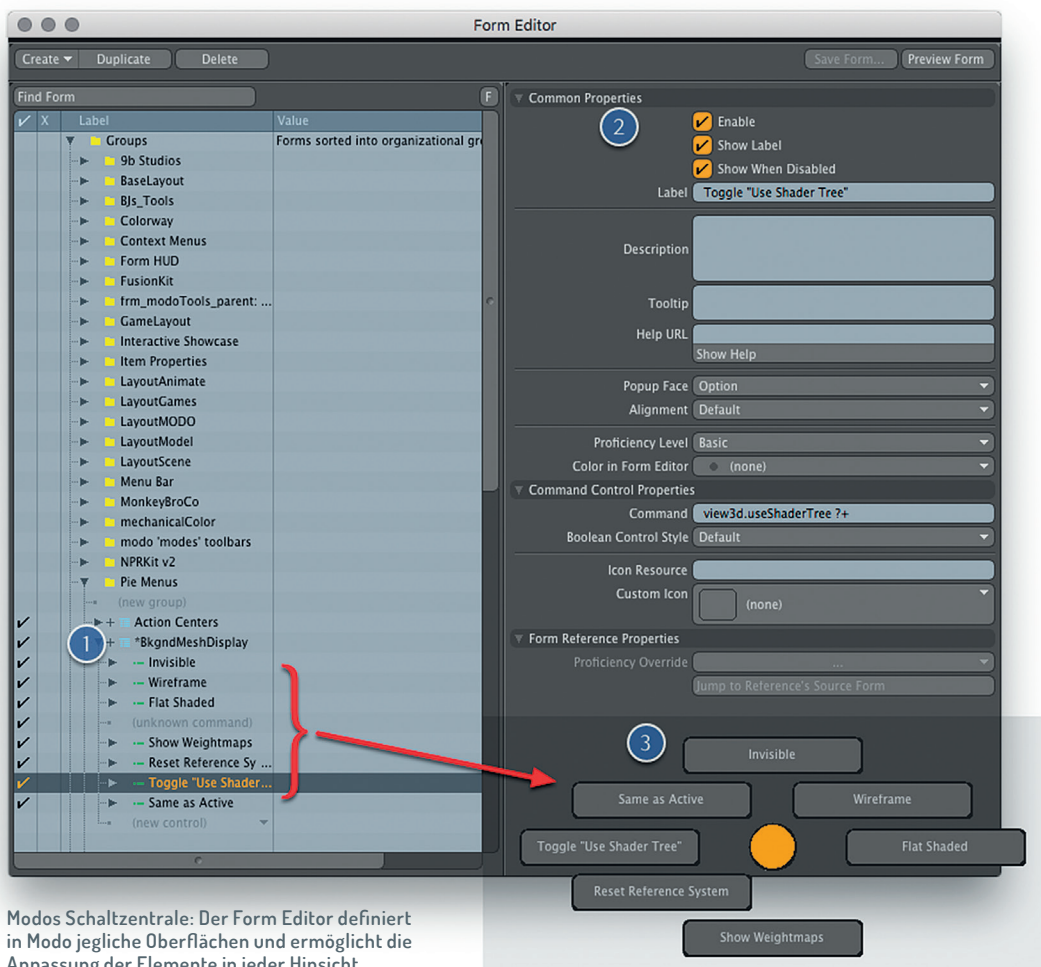
Achtung: Bei normalem Header signalisiert der gelbe Pfeil ein maximiertes Fenster, hat das Fenster aber einen minimierten Header, weist visuell nichts darauf hin, dass unter Umständen mehrere Fenster durch dieses verdeckt werden. Gerade wenn man anfängt, die Maximize-Funktion häufiger zu nutzen, wird man also das ein oder andere Mal ein verdecktes Fenster suchen. Da der Shortcut das lästige Klicken unterbindet, haben Laptop User hier mit der Standardbelegung „O“-Taste ein Problem, da die „O“ des Num-Pads benötigt wird. Dies ist ein günstiger Augenblick, um einen weiteren wichtigen Teil des Modo UI einzuführen: den Input Editor, der direkt unter System zu finden ist.

Er ist die Schaltzentrale zur kompletten Individualisierung der Tastatureingaben unter Modo – ob das Ändern eines bestehenden Shortcuts, das Hinzufügen eines neuen, das Ausführen eines Makros und weitere, mächtige Funktionen, all dies geschieht hier. Um nun also für den Anfang einfach die Maximize-Operation auf einen neuen Shortcut zu legen, kann man den entsprechenden Befehl – wenn bekannt – entweder direkt im Input Editor eingeben, oder man lässt Modo den Befehl anzeigen und weist dann den Shortcut zu.

- Liste der wichtigsten Pie Menus**
- 1 Ctrl+3 | Schnelles Öffnen diverser Pop-over
 - 2 Ctrl+1 | GL Viewport Toggles
 - 3 Ctrl+4 | Topology Tools
 - 4 Ctrl+2 | GL Viewport Style
 - 5 Ctrl+6 | Background Mesh Display, teilweise individualisiert via Form Editor
 - 6 Ctrl+5 | Topology Pen, nur aktiv wenn Topology Pen aktiv ist
 - 7 Ctrl+Shift+Space (eigene Definition) | GL Viewport Restore Space, via Shortcut auf deutscher Tastatur nicht verfügbar
 - 8 Ctrl+F | Mesh Fusion Tools
 - 9 Ctrl+Space | GL Viewport Typs
 - 10 Alt+Space | kontextsensitive Schnellauswahl (Animation/3D Model View)



Über die Command History ist das Mapping eines neuen Shortcuts sehr einfach.



Modos Schaltzentrale: Der Form Editor definiert in Modo jegliche Oberflächen und ermöglicht die Anpassung der Elemente in jeder Hinsicht.

Eintrag Key-Mappings. Es empfiehlt sich, die Option „Append to File“ zu aktivieren, da so die Datei immer erweitert werden kann und ansonsten manchmal Mappings verlorengehen.

Und ein Tipp für Mac-User: Auf diese Weise lassen sich auch Shortcuts korrigieren, die durch das deutsche Tastaturlayout ansonsten nicht zugänglich sind, wie zum Beispiel Invert Selection (die öffnende eckige Klammer kann ganz einfach auf zum Beispiel Alt+5 gemappt werden) oder Connected (auf Alt+6).

Mit globalen Shortcuts lassen sich hervorragend layoutübergreifende Commands ausführen, doch wird man damit nicht alle Operationen abbilden können und wollen. So ist es manchmal extrem hilfreich, auch im Model-Layout auf die Sculpt-

Hierzu verwendet man idealerweise die Command History, die über „F5“ aufgerufen werden kann. Sie speichert jegliche Aktionen in Modo und kann auch verwendet werden, um schnell in der Undo-Liste zu einem bestimmten Punkt zurückzuspringen. Mit geöffneter History führt man jetzt einmal die Aktion aus, die man neu belegen oder ändern möchte. Diese wird dann direkt in der Command History angezeigt. Mit RMB kann man neben Hilfe zu dem Thema und Infos zu dem Command auch direkt ein neues Map-

ping zuweisen. Möchte man den Shortcut auf einen bestimmten Bereich, einen View oder ein Layout begrenzen, kann dies in dem Menü mit dem Button „...“ definiert werden. Im Gegensatz zu dem Viewport-Layout lässt sich das Mapping sogar einzeln als Config-File abspeichern und jederzeit auf eine andere Version von Modo übertragen.

Um ein Layout abzuspeichern, wählt man File > Config Export und vergibt einen beliebigen Namen. In dem dann folgenden Dialogfeld selektiert man unter Fragment den

Tools zuzugreifen, schnell einmal aus dem Setup ein „Align to Work Plane“ zu nutzen oder im Animation-Layout schnell Zugriff auf die Item-List zu erhalten.

Hierfür hat Modo zwei zusätzliche UI-Erweiterungen parat: Palettes und Pie Menus. Die Palettes sind im Grunde Auskopplungen aus dem bestehenden GUI als Floating Windows, sodass man jederzeit (und permanent) Zugriff auf entsprechende Funktionen erhalten kann. Eine umfangreiche Auswahl findet sich unter Layout > Palettes, wobei auch hier

bei häufigerer Verwendung eine Kopplung an einen Shortcut sinnvoll ist. Standardmäßig lassen sich via F2 die Model Tool Bar, via F3 die Sculpt and Paint Tool Bar und F4 die Setup Tool Bar aktivieren, welche man sehr häufig auch in anderen Layouts benötigt.

Die wohl mächtigste UI-Komponente sind die Pie Menus, die ein visuelles Button-Menü an der Cursor-Position erzeugen und die Auswahl einer Option durch einfaches Hovern und Lösen des Shortcuts ohne weitere Aktion (z.B. Klick) zulassen.

Um das unter No.7 aufgeführte Pie Menu, welches schnellen Zugriff auf das View-Management bietet, zu erhalten, muss über System > Input Editor ein eigener Shortcut mit dem Command „attr.formPopover {90553987532:sheet}“ definiert werden. Wie unter 5. zu sehen ist, können die Pie Menus auch angepasst beziehungsweise komplett neu erstellt werden.

Aufgebohrt – Modos Grundfeste

Nicht nur die Pie Menus, sondern die gesamte Modo-Oberfläche ist modular aufgebaut. So wie die Fenster individuell zusammenstellbar sind, können auch die Inhalte angepasst und individualisiert werden. Unter System > Form Editor findet sich hier das Tool, welches Zugriff auf Modos gesamtes Basissystem bietet.

Um zum Beispiel das modifizierte Pie Menu aus No.5 zu erstellen, öffnet man den Form Editor, klickt einmal auf Find Form und öffnet via Ctrl+6 das Pie Menu. Dieses wird jetzt in der Objekt-Hierarchie angezeigt und kann beliebig angepasst und erweitert werden. Auf die gleiche Art und Weise können so jegliche Elemente und Objekte von Modos UI verändert werden.

Ganz unschlagbare Informationen

Da auch die Tools in Modo eigentlich ein Sammelsurium von Grundfunktionen sind und über die Tool-Pipeline genauso individualisiert und neu erstellt werden können wie Elemente im UI, ist die Grenze hier fließend und könnte bestimmt noch einen weiteren Artikel füllen. Ein paar themenübergreifende Tipps aber noch zum Schluss, die noch einen Workflow-Boost bringen.

- ▷ Sowohl die Item-List als auch der Shader-Tree sind zentrale Elemente in Modo und werden durch den hierarchischen Aufbau teilweise sehr lang (vertikal). Die häufig benötigte Drag-and-drop-Funktionalität wird zur Qual, wenn die Elemente weit auseinander liegen und man immer den unteren Rand treffen muss, damit die Liste weiter scrollt. Lösung: Man erzeugt sich in einem anderen Tabbed Viewport eine zweite Item List/Shader Tree. So kann man in einer Liste die Quell-Position, in der anderen die Ziel-Position einstellen und zieht das Item/Light/Material etc. einfach von einer Liste in die andere.
- ▷ Alle Werteingabefelder in Modo sind kalkulationsfähig. Somit können Berechnungen (Multiplizieren *, Dividieren /, Addieren +, Subtrahieren -) direkt in den Feldern als mathematische Gleichung eingegeben werden. In Kombination mit Ctrl kann dies auch sofort für Wertgruppen relativ übernommen werden. Um zum Beispiel die Auflösung zu verdoppeln, die Frame Width „*2“ und mit Ctrl+Enter bestätigen. Die Height wird automatisch angepasst.
- ▷ Ist ein Tool aktiv und somit eine Werteingabe fokussiert, kann durch Sliden im Viewport der aktive Wert verändert werden. Beim Bevel-Tool kann hierdurch zum Beispiel die Eingabe für Value interaktiv angepasst werden. Eine weitere Möglichkeit ist, während eines aktiven Tools „K“ zu drücken, um direkt unter dem Cursor ein Channel Haul mit den aktiven Werteingaben zu erhalten.
- ▷ Alle Buttons mit einem kleinen dunklen Dreieck in der rechten unteren Ecke enthalten ein Pop-up mit weiteren Optionen. Normalerweise wird dieses durch LMB und halten aktiviert. Etwas schneller geht es mit RMB.
- ▷ Unter Preferences > Display > Tool Handles > Draw Style können die Tool Handles (dauerhaft) von Basic auf Advanced gestellt werden und bieten dann in vielen Situationen zusätzliche Funktionalitäten direkt im Viewport.
- ▷ Mit Ctrl+D können die Werte eines Tools wieder auf Default gesetzt werden.
- ▷ Viele Views lassen sich in bis zu zwei Stufen minimieren und verändern dabei

ihre Darstellung (zum Beispiel Item-List, Toolbox).

- ▷ Alle Farbeingabefelder bieten eine erweiterte Funktionalität bei Verwendung von Zusatztasten: Shift+Drag = Helligkeit, Ctrl+Drag = Färbung, Alt+Drag = Sättigung.
- ▷ Um bei einem Shortcut einen Toggle zu erzeugen, setzt man hinter den Command ein „?+“.

Fazit

Ich arbeite selbst als Dozent an einer Hochschule und bin immer darauf bedacht, mich nicht zu weit von den Default-Settings eines Programms zu entfernen. Doch die Möglichkeiten, die Modo im Bereich der Individualisierung bietet, gehen einfach weit über ein einfaches Schickmachen hinaus. Auf diese Weise lassen sich teilweise Arbeitsprozesse extrem beschleunigen, sodass man sich mit dem Layout-System, dem Form und dem Input Editor auf jeden Fall einmal auseinandersetzen sollte. Die Möglichkeit, diese eigenen Definitionen dann auch in jede neue Version zu überführen, machen eine Anpassung umso sinnvoller. Somit hat sich bei meinen Kollegen und Studenten häufig die Situation ergeben, dass meine Anpassungen eher das Interesse geweckt haben, als Frust darüber, dass es nicht ›das Original‹ ist. Mein Tipp – einfach mal antesten! > ei



André ist ausgebildeter Mediengestalter (B.Sc.) und arbeitet als Freiberufler mit 10 Jahren Berufserfahrung und als Lehrer für Schule und Wirtschaft. Sein Portfolio umfasst Projekte aus Grafikdesign, 3D, CGI und Anwendungsentwicklung. Dazu arbeitet er als Dozent und Berater für Unternehmen und öffentliche Einrichtungen, und als externer Dozent an der FH Bielefeld, Fachbereich Gestaltung.

www.revid.de

Links

- Ein Tutorial zum Viewport
- ▷ <http://bit.ly/ModoCustLay>
- Das Docset MODO 10 für Dash
- ▷ bit.ly/modo10docset
- Windows Alternative zu Dash: Velocity
- ▷ velocity.silverlakesoftware.com

Anzeige

maconcept.

maconcept. ist The Foundry Distributor / Reseller für D-A-CH und weitere europäische Länder. Sie sind Händler? Dann freuen wir uns auf ihre Anfragen. Sie sind User? Fragen Sie uns nach einem aktuellen Angebot!

**THE
FOUNDRY.**
AUTHORIZED MODO PARTNER