

Visualisierung der verschiedenen Boolean-Typen, die MeshFusion bereitstellt. Die Farben entsprechen der Zuweisung im MeshFusion 3D Tree.

MeshFusion in Modo 901!

Anfänglich als Standalone-Software für beliebige Produkte entwickelt, liegt das Hauptaugenmerk der Entwickler derzeit auf der Implementierung in Modo. Mit welcher Leidenschaft dies geschieht, sieht man daran, dass aus dem ursprünglichen Plug-in innerhalb relativ kurzer Zeit ein vollwertiger Part von Modo geworden ist. Man kann sich bei gewissen Problemen nicht mehr vorstellen, sie ohne MeshFusion lösen zu müssen. Wo Licht ist, ist natürlich auch immer Schatten, und so gibt es im Bereich der Performance und einiger topologischer Aspekte durchaus noch Entwicklungspotenzial. von André Löscher

MeshFusion basiert im Kern auf GroBoto, eine von Braid Art Labs (Zwei-Mann-Team!) entwickelte Technologie für Echtzeit-3D-Modeling und automatische Mesh-Generierung. Hierbei basiert die Geometrie-Erstellung nicht auf der sonst aus dem Hard Surface Modeling bekannten Bearbeitung von Polygonen, Edges und Vertices, sondern auf der Verknüpfung von Geometrie mittels sogenannter boolescher Operatoren. Diese Verknüpfungen basieren auf boolescher Algebra und enthalten die Operationen UND (Konjunktion, in Modo „Intersect“), ODER (Disjunktion, in Modo „Union“), NICHT (Negation, in Modo „Subtract“) und XOR (ausschließendes ODER). Diese Technologie wurde in den Anfängen als Plug-in vertrieben und definierte das Arbeiten mit Booleans komplett neu (mehr zu GroBoto lesen Sie in DP 04:2013).

Boolesche Funktionen – Modeling neu definiert

Der Sinn von Booleans ergibt sich daraus, dass gewisse Modeling-Prozesse in einem

Hard Surface Modeler nur sehr umständlich oder durch geschickte Tweaks umzusetzen sind. Als Beispiel stelle man sich den Aufwand vor, ein sauberes Loch in eine Kugeloberfläche zu modellieren. Möglich ja, aber aufwendig. Solche Modifikationen lassen sich durch MeshFusion mit einem Klick umsetzen. Für alle Leser die auch CAD-basiert arbeiten: Aus den booleschen Operatoren ergibt sich nach wie vor kein Volumen der Objekte.

Die Funktionen werden ausschließlich auf die Hüllengeometrie angewandt. Nichtsdestotrotz können durch MeshFusion erzeugte Objekte nach dem Finalisierungsprozess durch Export in ein Volumenobjekt ausgegeben werden. Die Integrität des Meshs ist so gut, dass über Power-SubD-Nurbs (Modo-Plug-in) ein Export in Parasolid, Iges, Step, Sat oder Rhino möglich ist. Die Modo-eigene Implementierung der booleschen Operatoren lässt darauf schließen, dass ein Tool in der Art und Weise nicht zielführend sein konnte – hohe Anforderungen an die Ausgangsgeometrie, destruktives Verfahren, (in den allermeisten Fällen) Zerstörung der geo-

metrischen Konsistenz. Und dieses Problem hatte nicht nur Modo, sondern die meisten in 3D-Programmen erzeugten Ergebnisse, die eine nachträgliche Weiterbearbeitung im besten Fall schwierig machten.

Booleans in Modo 901

Bis auf die geometrische Konsistenz, die unter MeshFusion nur noch in extremen Fällen nicht ganz sauber ist, kann das Tool in der heutigen Form in allen anderen Belangen klar punkten. Die Anforderungen an das Ausgangs-Mesh sind bis auf die Nicht-Benutzung von N-Gons (Flächen in der Topologie mit mehr als vier Eckpunkten) akzeptabel. Es basiert auf einem Live-Editing-Verfahren, das nicht nur die Modifikation der einzelnen Komponenten zu jedem Zeitpunkt des Entwicklungsprozesses zulässt, sondern auch deren boolesche Operation und die Schnittkanten zu jeder Zeit editierbar hält.

Dieses wird im Folgenden in den einzelnen Abschnitten detailliert erklärt. Die Plug-in-Version in Modo 801 war schon recht umfangreich, aber es war eben „nur“ ein

Plug-in. Somit war gefühlt ein MeshFusion-Objekt während der Arbeit mit den Pop-ups und der unterschiedlichen GUI nicht wirklich ein Modo-eigenes Objekt. Hier fühlt sich 901 ganz anders an.

MeshFusion-Basics

Um MeshFusion verwenden zu können, muss die Geometrie bestimmte Anforderungen erfüllen. Dieser Umstand macht es sinnvoll, sauber und konsistent zu arbeiten. Idealerweise sollte das Mesh nur aus Quads, also geometrischen Flächen mit vier Eckpunkten bestehen. Diese haben neben der ressourcenschonenden Arbeitsweise in MeshFusion auch noch den Vorteil, dass sich die meisten Modeling Tools besser anwenden lassen und dass das Ergebnis „sauber“ ist.

Die Verwendung von Triangles ist möglich, wird aber direkt bei Erstellung eines Fusion-Objekts mit einer Warnmeldung quittiert, dass hierfür eine aufwendigere Topologie benötigt wird. Nebenbei wird man auch, wenn es um die Integrität der Ausgabe geht, schnell merken, dass man auf Triangles nach Möglichkeit verzichten sollte.

Die Verwendung von Triangles ist für das Erzeugen von Oberflächen (außer bei Planes, die nicht im SubD vorliegen) immer ein Problem. Abgesehen davon, dass meistens der Polycount stark ansteigt, bringen die Quads auch bei Kanten, Ecken und Rundungen immer wieder Probleme mit sich. Spätestens wenn man in den SubD-Modus wechselt, wird das auch ohne Wireframe für das Auge sichtbar (vorderer linker Zylinder). Die Kanten sind unförmig und haben keinen

sauberen Verlauf. Auch N-Gons erzeugen gerade bei reflektierenden Materialien im SubD Probleme (rechter vorderer Zylinder), da die Oberfläche immer ein unsauberes Bild ergibt (obere Fläche).

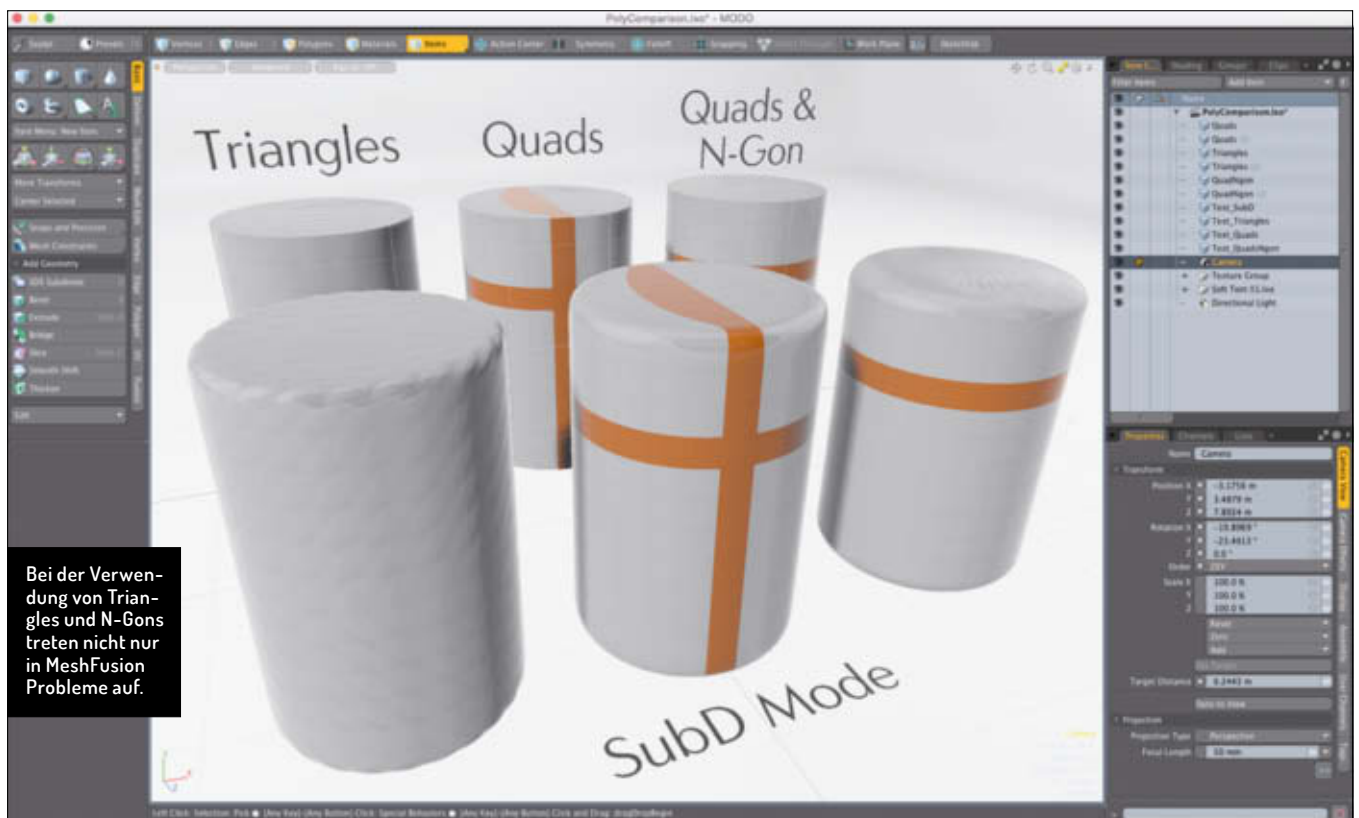
Für die weitere Bearbeitung sind sowohl Triangles als auch N-Gons eine Barriere. Orange dargestellt sind saubere Loops im Mesh, in die man neue Slices einbringen könnte. In einem rein aus Quads aufgebauten Mesh kann man vielleicht nicht immer einen linearen Cut machen (ja, nach Geometrie), aber ein Slicing ist jederzeit möglich, da ich bei vier Vertices aus jeder Richtung einen Cut setzen kann. Bei Triangles und N-Gons lässt sich kein Loop generieren, da sie geometrisch gesehen keinen sauberen Cut erlauben. Modo ermöglicht zwar, mit dem Edge Slice Tool auch Triangles und N-Gons beliebig zu schneiden, aber die mächtige Pattern-Selection, die eine Selektion basierend auf einem bestimmten Muster logisch und automatisch fortführen kann, ist in diesem Fall nicht einsetzbar.

Was zu einem sofortigen Abbruch der „Fusionierung“ führt, sind N-Gons, also Polygone mit mehr als vier Vertices. Möchte man in Modo seine Geometrie daraufhin überprüfen, ob N-Gons vorhanden sind, kann dies bei selektiertem Mesh über den Tab „Lists“ geschehen. Im Bereich „Statistics“ erweitert man den Eintrag „Polygons“ und „By Vertex“ und hat dann eine gezielte Übersicht auf Mesh Parts mit „x“-Vertices. Quads wären hier unter der Rubrik „4“ zu finden, Triangles entsprechend unter „3“ und N-Gons unter „>4“. Somit sollte hier in der Spalte „Num“ eine Null stehen.

Mit MeshFusion wird in den Assets eine sehr gute Bibliothek an Elementen (Qbics) mitgeliefert, die weitaus umfangreicher ist als die Auswahl an den sogenannten Primitives. Aus der Menge der beigefügten Objekte lässt sich für eine Vielzahl von Grundelementen eine Basis auswählen, die dann die optimale Mesh-Geometrie aufweist (Quads). Der Unterschied besteht nur in einem speziellen Item-Namen (Initial „Q_“) und einer abweichenden Darstellung im 3D Viewport (grünlich schimmernde Oberfläche), die aber im Shader Tree angepasst werden kann.

Das durch MeshFusion erzeugte Objekt wird dann in einem finalen Prozess in eine normale, Mesh-basierte Oberflächengeometrie umgewandelt. Dieser Schritt lässt sich zu jeder Zeit vornehmen, auch innerhalb des Entwicklungsprozesses, um zum Beispiel die Integrität der Ausgabe zu prüfen oder einen Export zur Weiterbearbeitung in ein anderes Programm zu testen.

Neben dem eigentlichen Mesh kann in diesem Prozess gleichzeitig eine UV-Map erzeugt werden. Dies ist gerade bei komplexen Strukturen ein wichtiger und arbeitssparender Schritt, da mit MeshFusion sehr natürliche Strukturen bis hin zu organischen Formen möglich sind. Müsste man das Ergebnis im Nachhinein noch unwrappen, würde unter Umständen noch einmal der gleiche Zeitaufwand wie für das Modellieren an sich anfallen, da Unwrapping bei komplexen Formen teilweise ein Glücksspiel ist. So aber kann die UV-Map automatisch durch MeshFusion generiert werden und einem sofortigen Texturieren des Objekts steht nichts mehr im Wege.



Bei der Verwendung von Triangles und N-Gons treten nicht nur in MeshFusion Probleme auf.

New Fusion Item

Ein Fusion-Objekt ist im Grunde genommen nichts anderes als ein Konstrukt sich durchdringender und/oder verschmelzender Einzelteile, für die Modo formbare Schnitt-

kanten, sogenannte Strips erstellt. Aber, so simpel es klingt, bevor man ein MeshFusion-Objekt angeht, besteht die Kunst darin, sich über die Bedeutung und Form der Einzelteile klar zu werden. Die Herangehensweise kann (muss aber nicht) eine ganz andere

sein, als man es basierend auf den Kenntnissen aus dem Hard Surface Modeling erwarten würde.

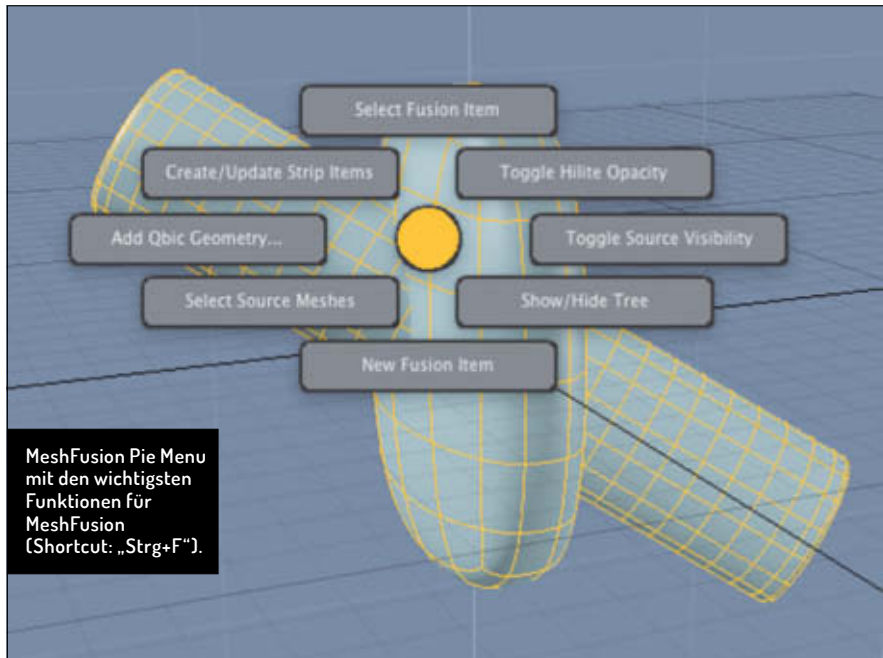
Vor dem Erstellen ist es von Bedeutung, in welcher Reihenfolge die Elemente selektiert werden. Bei „Union of Meshs“ und „Intersect Meshs“ liegt es in der Natur der Dinge, dass die Reihenfolge nicht relevant ist. Bei einer Subtraktion schon. Hier kann entweder erst das Grundobjekt gewählt werden und alle nachfolgend selektierten Objekte werden abgezogen („Substract Others from 1st Selected“), oder das erste Objekt wird von allen nachträglich selektierten subtrahiert („Substract 1st Selected from Others“).

Sind nun zwei oder mehr saubere Mesh Items vorhanden, gibt es mehrere Möglichkeiten, ein Fusion-Objekt zu erstellen. Die erste und offensichtlichste Methode ist die Nutzung des Buttons „New Fusion“ im „Fusion“-Tab der „Modeling Toolbox“. Eine zweite und sehr viel schnellere Möglichkeit besteht in der Nutzung des kontextsensitiven Fusion Pie Menu, das mit dem Shortcut „Strg+F“ zu erreichen ist. Das sich daraufhin öffnende Optionsfenster ist das gleiche, das auch über das Menü zu erreichen ist.

Zuerst sollte man dem Fusion Item, wie auch jedem anderen Objekt in der Item List, einen Namen geben. Je mehr Objekte eine Szene umfasst, umso sinnvoller ist dieser kleine Arbeitsaufwand am Anfang. Auch macht es Sinn für die Übersicht der Szene, die einem Fusion Item zugeordneten Meshs in einer Gruppe zu sammeln. Hierzu alle Objekte in der Item List markieren und mit „Strg+G“ eine Gruppe erstellen. Dies hat keinen Einfluss auf die Funktionalität in Modo (also keine Angst vor Double Transformation!), es macht das Handling und die Selektion aber weitaus einfacher. Nun kann man zwischen den zwei grundlegenden Darstellungsformen von MeshFusion wählen: „3D Tree Fusion“ und „Schematic Fusion“. Obwohl ein 3D Tree Fusion am Anfang bestimmt „cooler“ wirkt, ist für mich auf jeden Fall die Schematic-Version die erste Wahl, aber dazu später mehr im Kapitel „Schematic Fusion“.

Um die Funktionsweise von MeshFusion besser verstehen zu können, kann man sein erstes Fusion auf jeden Fall als 3D Tree starten, da es das Grundprinzip sehr gut veranschaulicht. Es ist eine visuelle Reproduktion der Grundbausteine des fertigen Objekts und zeigt die Verknüpfung und Gruppierung der Elemente. Mit diesem Setting zu starten hat weiterhin den Vorteil, dass ein 3D Tree Fusion nachträglich in ein Schematic Fusion umgewandelt werden kann, dies umgekehrt aber nicht möglich ist.

Die nächste Checkbox für „Absolute Strip Width“ ist neu und ermöglicht das Erstellen von Schnittkanten aller Teile mit einer fixen



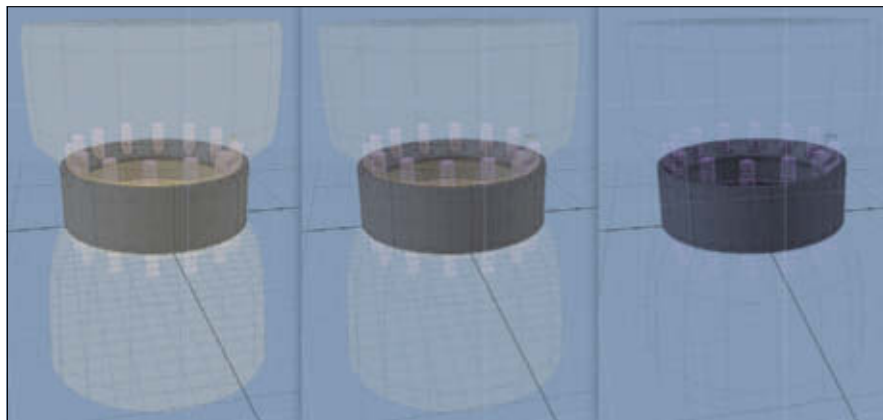
MeshFusion Pie Menu mit den wichtigsten Funktionen für MeshFusion (Shortcut: „Strg+F“).



Über die Einstellung „Absolute Strip Width“ lassen sich Strips erzeugen, die eine parallele Geometrie basierend auf einem gegebenen Wert in Millimetern aufweisen.

Topologie verläuft parallel.

Topologie verläuft konisch.



Die Source Meshs werden als Ghosts schemenhaft dargestellt. „Cycle Mesh Opacity“ ermöglicht es, die Transparenz der Ghosts anzupassen. In komplexen Szenen ein wichtiges Tool

Dimensionen in Modo

Da MeshFusion sozusagen dazu anhält, intensiver in Richtung Konstruktion und CAD zu denken, haben natürlich auch Faktoren wie Größen und Längen eine weitaus größere Relevanz als bei normalem Modeling für ein Raytracing. Seit 801 gibt es eine schnelle und einfache Möglichkeit, ein Objekt sofort bei der Selektion mit den Dimensionen zu versehen. Dazu wird das Element einfach in der Item List selektiert und dann in den „Properties“ unter „Display“ die Option „Add Dimensions“ gewählt. Weitere Möglichkeiten für das Arbeiten mit Dimensionen finden sich unter „View“. Das „Ruler Tool“ ist ein dreidimensionales Lineal, mit dem sich Distanzen zwischen beliebigen Komponenten messen lassen. Mit dem „Protractor Tool“ lassen sich sehr genau Winkel bestimmen. Auch hier können über zwei Handles beliebige Komponenten oder Positionen im Raum selektiert werden. Zuletzt findet sich dort noch das „Dimensions Tool“, das auf den ersten Blick im Item Mode den gleichen Effekt hat wie die Funktion „Add Dimensions“. In den Component Modes lässt sich mit dem „Dimensions Tool“ aber sogar die Dimension eines Polygons oder einer Edge anzeigen. Für alle diese Tools kann für das genaue Platzieren die Funktion „Snapping“ aktiviert werden, die es einem extrem erleichtert, die Handles an zum Beispiel Vertices oder dem Grid auszurichten. Es kann als Snapping Target sogar der Center einer Edge oder eines Polygons definiert werden.

Breite (Millimeter). Ist diese Option deaktiviert, wird die Stärke der Schnittkante anhand des Größenverhältnisses und Lagewinkels der zusammengeführten Flächen der Grundobjekte erstellt. Dies kann gerade bei gerundeten Oberflächen eine ungewünschte Varianz in der Breite der Schnittkanten erzeugen. Aktiviert man die Option, wird eine einheitliche Stärke der Schnittkanten berechnet, sodass die Kanten gerade für maschinell gefräste oder in ähnlicher Weise gefertigte Stücke weitaus authentischer wirken. Diese Option lässt sich auch nachträglich unter den Fusion „Properties“, Tab „Fusion“ in den „Global Strip Settings“ ändern.

Über das Optionsfeld „Show Tree“ kann die Anzeige der Miniaturausgabe der MeshFusion-Hierarchie aktiviert/respektive deaktiviert werden und mit „New Schematic Workspace“ definiert man, ob das Fusion-Objekt in der Schematic View („Layout > Layouts > Setup“ respektive „Schematic Fusion“) in den Standard-Workspace übernommen oder ob ein neuer, eigener Workspace erstellt wird. Wird dies aktiviert und ein Schematic Fusion erstellt, trägt der Workspace den gleichen Namen wie das Fusion Item. Mit einem Klick auf eine der vier „Create New Fusion Item“-Optionen wird nun das eigentliche Fusion-

Objekt erzeugt. Direkt nach der Generierung kann der Tree mittels der Handles des Move Tools bewegt werden. Im 3D Viewport erscheint nun je nach Form und boolescher Operation das entsprechende Fusion-Objekt und an die Stelle der Source Meshs treten schemenhaft die ursprünglichen Meshs als „Ghost“-Objekte.

Qbics – Formen-Quell für Fusion

Basierend auf der am Anfang beschriebenen Anforderung, dass die für MeshFusion verwendeten Meshs gewisse Voraussetzungen erfüllen sollten, haben sich die normalen Primitives einerseits von ihrer Topologie als ungünstig erwiesen (zum Beispiel ist die Standard-Sphere an den Polen aus Triangles aufgebaut) und andererseits stellt ein Designer an Grundformen natürlich andere Anforderungen, als nur eine geometrische Basis zu haben.

Somit bieten die Qbics Designern die Möglichkeit, den Einstieg in die Entwicklung ihrer Idee durch die Verwendung dieser Bibliothek zu vereinfachen. Die Qbics sind mit derzeit 51 Elementen eine extrem umfangreiche Erweiterung der Grundformen in Modo, denn ihr unterschiedliches Aussehen gegenüber Primitives erhalten sie nur durch ein vordefiniertes Material. Löscht man dieses (siehe auch Kapitel „Sauber machen – Fusion-Optimierung“ in der nächsten Ausgabe DP 07:2015) oder ersetzt es durch ein Standardmaterial, stellt man fest, dass man hier ein ganz normales Mesh vor sich hat. Nur bei der Größe sollte man immer ein Auge auf die Dimensionen haben, die teilweise deutlich von den 1 x 1 m der Primitives abweichen.

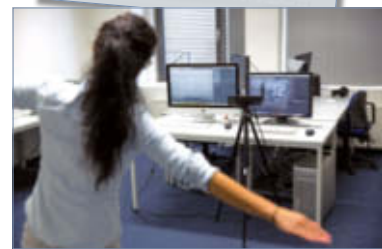
Da es im Gegensatz zu den Standard-Primitives keine Möglichkeit gibt, die Qbics basierend auf einer Größenvorgabe zu erstellen oder „aufzuziehen“, sei hier noch einmal der Hinweis auf eine versteckte, aber nützliche Funktion gegeben, die dieses auch nach dem Einfügen des Qbics ermöglicht. In der Modeling Toolbox unter „Snaps and Precision“ findet sich der Command „Absolute Scaling“. Dies ist definitiv ein Geheimtipp, wenn man häufig mit exakten Maßen für bestimmte Objekte arbeiten muss. Als Referenzwert kann das Maß jeder Achse genommen werden. Für die Transformation lässt sich ein „Uniform Scale“ (alle Achsen werden basierend auf dem Skalierungsfaktor für die Referenzachse angepasst), ein „Explicit Scale“ (alle Achsen werden exakt auf den angegebenen Wert skaliert) oder – die wahrscheinlich mächtigste Option – ein „Reference Scale“ ausführen. Das Letztere agiert auch in den Component Modes „Edges“ und „Polygons“. Selektiert man ein oder mehrere Edges/Polys und gibt für sie entsprechende



Mocap Zuhause!

Sabrina Langenberger und Uli Plank haben sich im Low-Cost-Motion-Capture-Segment umgeschaut – und Kobold für gut befunden. Jetzt exerzieren wir einmal den kompletten Workflow durch.

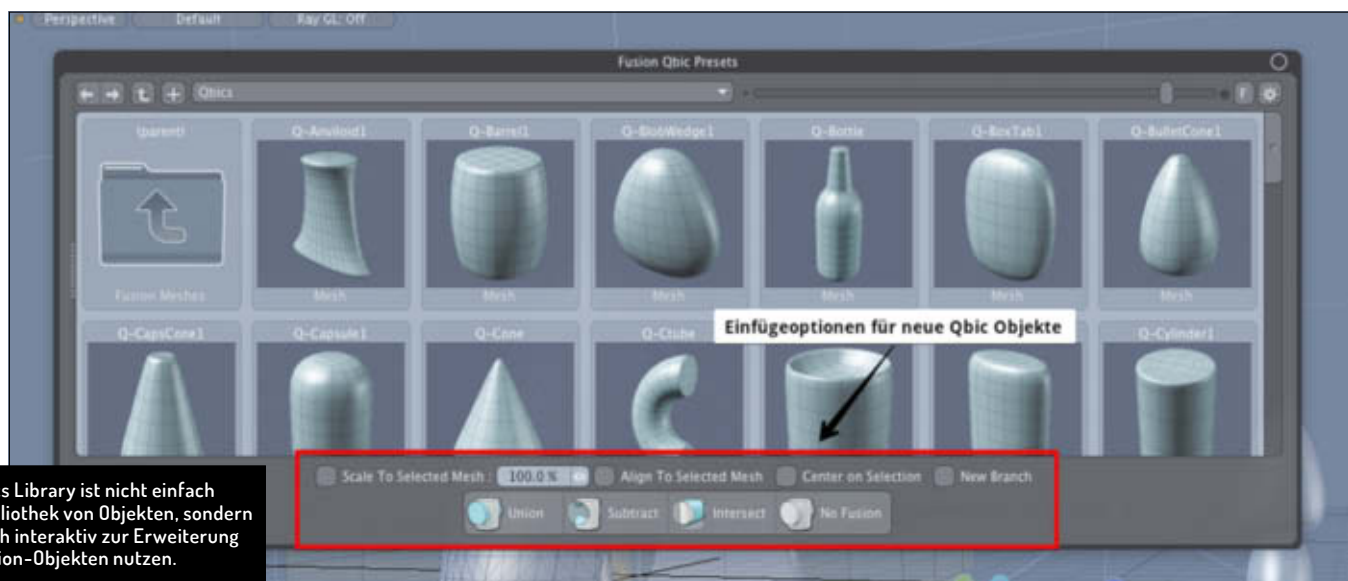
PDF zum Download unter www.digitalproduction.com



CREATIVE MEDIA EDUCATION

Diploma-, Bachelor- und Masterabschlüsse in der Medienbranche

www.sae.edu



Die Qbics Library ist nicht einfach eine Bibliothek von Objekten, sondern lässt sich interaktiv zur Erweiterung von Fusion-Objekten nutzen.

Maße an, wird der gesamte Rest proportional skaliert. Weiß man also die Größe eines Bereichs, kann man hiermit das ganze Objekt basierend auf diesen Werten anpassen.

In den „Fusion Qbic Presets“ gibt es seit 901 eine eher unscheinbare, aber sehr praktische Erweiterung. Während in der Plug-in-Version unter 801 die Elemente ähnlich der normalen Item-Bibliothek durch Doppelklick im Nullpunkt des Welt-Koordinatensystems eingefügt wurden und manuell dem Fusion hinzugefügt werden mussten, kann man jetzt über die entsprechenden Optionsflächen und Buttons sofort definieren, mit welcher booleschen Funktion und an welcher

Position der Geometrie und Hierarchie das neue Objekt eingefügt werden soll. Voraussetzung hierfür ist, dass das Ziel-Mesh vor(!) dem Öffnen der Bibliothek selektiert wird. Meiner Meinung nach sind alleine solche Detailverbesserungen ein Segen und machen die Integration von MeshFusion in Modo noch sinnvoller.

Fusion-Struktur in 3D

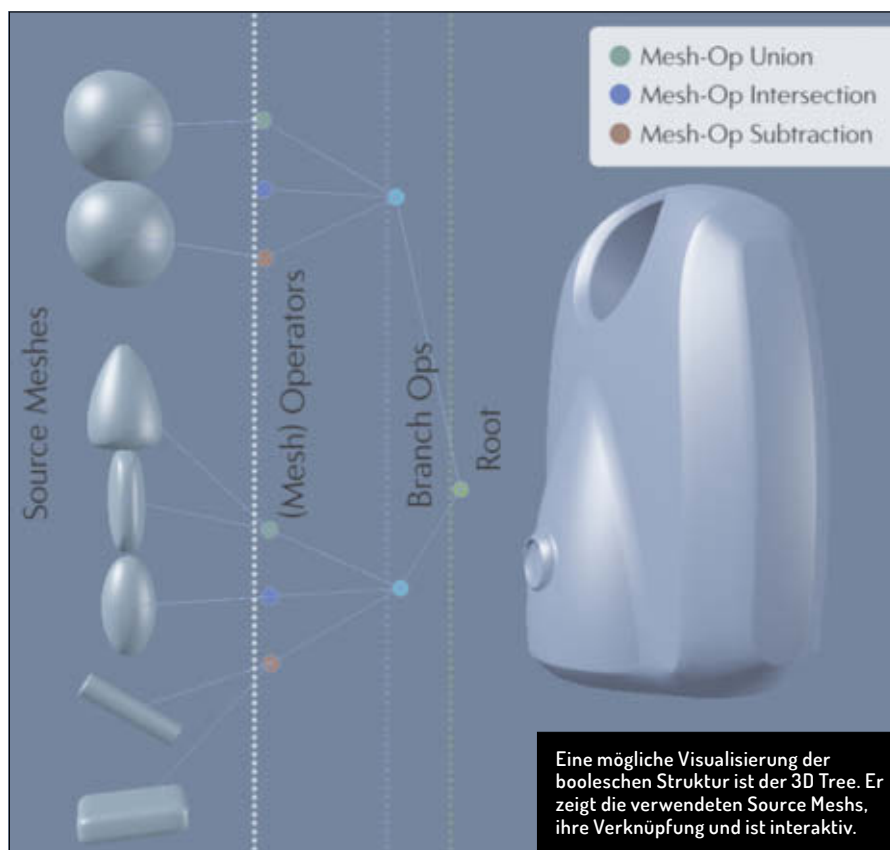
Kleinere Kompositionen, die schnell von der Hand gehen sollen, können einfach als 3D Tree direkt im Viewport erstellt werden. Man hat nur ein Fenster zu kontrollieren und die

Organisation der Elemente lässt sich direkt über den Tree vornehmen. Der Tree befindet sich immer an einer fixen Position neben dem Fusion Item. Die einzelnen Knoten unterscheiden sich farblich und geben so ihre Funktion an. Wenn man den Baum vom Startpunkt („Root“, einzelner grünlicher Knoten) aus betrachtet, dann gibt es immer einen vierstufigen Hierarchieaufbau.

Der Root steht für das Fusion-Objekt selbst und kombiniert alle „Zweige“ (Branch) mit einer booleschen Vereinigung („Union“). Diese Einstellung ist statisch und kann nicht geändert werden. Dieses führt bei einigen Projekten dazu, dass man sich im Vornherein gut überlegen muss, welche Grundelemente man in welche Branches mit welchen booleschen Operationen steckt. Spätestens hier wird aber klar, dass MeshFusion durch die nicht destruktive Umsetzung und das Live-Editing die Modo-internen booleschen Operatoren weit hinter sich gelassen hat. Der Root ist auch der Knoten, der für das Verschieben des Trees im Raum genutzt werden kann.

Als Nächstes folgen die Branch-Operatoren (kurz: Branch-Ops). Beim Erstellen eines neuen MeshFusion-Objekts wird immer genau ein vom Root ausgehender Zweig („Branch“) generiert. Im Verlauf einer Entwicklung können dann weitere Branches hinzukommen, die allesamt eine boolesche Schnittmenge („Intersect“) bilden. Der Sinn der Zweige wird bei komplexeren Projekten sehr schnell klar. Es ist eine Art Gruppierungsmöglichkeit, um bestimmte Teilaspekte des Modells voneinander zu separieren und dann in das Grundobjekt einzurechnen. Es verhindert damit, dass einige boolesche Operatoren bestimmte Teile der Komposition beeinflussen, obwohl sie diese überschneiden.

Die nun folgenden Mesh-Operatoren (kurz: Mesh-Ops) sind die Schlüsselemente für die aktive Zuweisung der booleschen



Eine mögliche Visualisierung der booleschen Struktur ist der 3D Tree. Er zeigt die verwendeten Source Meshes, ihre Verknüpfung und ist interaktiv.

Funktionen, die letztendlich die neue Geometrie aus den Grundobjekten erzeugen. Sie sind durch ihre Farbe gekennzeichnet und es kann damit ein „Union“ (grüner Punkt, Eselsbrücke: „Addition“ ist immer grün), ein „Intersect“ (blauer Punkt, Eselsbrücke: steht „in der Mitte“) oder ein „Subtract“ (roter Punkt, Eselsbrücke: „Subtraktion“ ist immer rot) von Geometrie erstellt werden. Für die Zuweisung der Mesh-Op ist es extrem sinnvoll, dass die Knoten im 3D Tree nicht nur zur Visualisierung der Funktion dienen, sondern auch interaktiv eingesetzt werden können. Aber dazu mehr im Abschnitt „Fusion Interactive“.

Die letzte Ebene stellt die ursprünglich verknüpften Objekte dar, sodass man neben dem Ergebnis der booleschen Funktionen auch noch die Quellstrukturen, die Source Meshs sehen kann. Dies ist sehr sinnvoll, da man zwar auch in der 3D View die Konturen der Source Meshs einblenden kann (die Ghosts), dieses bei komplexen Konstruktionen aber schnell unübersichtlich wird. Weiterhin vereinfacht es die Selektion einzelner Elemente stark, da innenliegende Strukturen andernfalls nur schwer erreichbar sind. Beim Hinzufügen, Verändern oder Bewegen kann es manchmal vorkommen, dass Mesh-Elemente nicht an den Enden der Ops-Knoten, sondern direkt auf den Knoten oder in irgendeiner Weise versetzt sind. Hierbei reicht es meistens, den Baum einmal über „Tree Visibility“ ein- und auszublenden (auch über „Strg+F“ > „Show/Hide Tree“) oder das Mesh noch einmal neu zu verknüpfen. Sollte man einmal in einem Component Mode die Geometrie vom Center des Items wegbewegt haben, kann über „More Fusion Controls“ > „Center to Bounds“ eine Neupositionierung des Objekts erfolgen.

Fusion Interactive – Tree Drag-and-drop

Ein unschlagbarer Vorteil des 3D Trees, solange die Struktur nicht zu komplex wird, ist die Interaktivität im 3D Viewport von Modo. Sie lässt sich sowohl für die Zuweisung einer neuen booleschen Operation als auch für das Eingliedern neuer Meshs nutzen. Für das Dragging kann entweder ein Item aus dem 3D Tree, dem Fusion-Objekt selbst oder der Item List verwendet werden. Als Ziel der Aktion kann ein beliebiges Item oder ein Ops-Knoten verwendet werden. Das sich öffnende Menü ist jeweils kontextsensitiv und passt sich der Drop-Aktion und dessen Elementen an.

Ein Dropping auf einen Branch-Ops oder den Root ist nicht möglich (in meinem Build 85499 zwar schon, aber es macht hier keinen Sinn und ist als Bug einzustufen). Wird für die Drag-Aktion ein Mesh genutzt, das noch kein Teil des Fusion-Objekts ist, kann man

in dem folgenden Pop-up definieren, wie das Element in den Fusion Tree eingegliedert werden soll. Wird ein Mesh verwendet, das schon Teil des Fusion-Objekts ist, gibt es mehr Möglichkeiten. Einerseits kann es mit einer neuen booleschen Operation belegt werden. Dies geschieht entweder mit einem Drag direkt auf den gewünschten Knoten oder mit einem Drag auf das gewünschte Mesh und Selektion der Funktion aus dem Kontextmenü. Eine weitere Möglichkeit bei dem Dropping auf ein Mesh ist, das sich ergebende Paar mit einer entsprechenden booleschen Verknüpfung als neuen Branch an den Root zu koppeln. Bezüglich der Branches ist noch zu erwähnen, dass die Drop-Aktionen eines Elements natürlich auch auf Ops-Knoten oder Meshs anderer Branches möglich sind. Wird ein bereits im Fusion-Objekt eingegliedertes Objekt auf seinen eigenen Ops-Knoten gezogen, lässt sich auf einfache Weise das Objekt innerhalb des Trees oder als neues Nicht-Fusion-Objekt duplizieren. Dies ist eine äußerst komfortable und schnelle Methode, um Kopien eines bestehenden Meshs zu erzeugen und diese umgehend innerhalb des Fusion Trees zu verwenden. Weiterhin lässt sich auf diese Weise schnell ein Objekt aus dem Fusion-Objekt ausgliedern („Remove Dropped from Fusion“), ohne es zu löschen.

Schematic Fusion

Wenn nun der 3D Tree eine solch übersichtliche und flexible Darstellung ist, warum braucht es dann noch eine weitere Fusion-Methodik? Dafür gibt es mehrere Gründe. Wie schon am Anfang beschrieben, kann der 3D Tree gewaltige Ausmaße annehmen und wird dann unübersichtlich (bis störend).

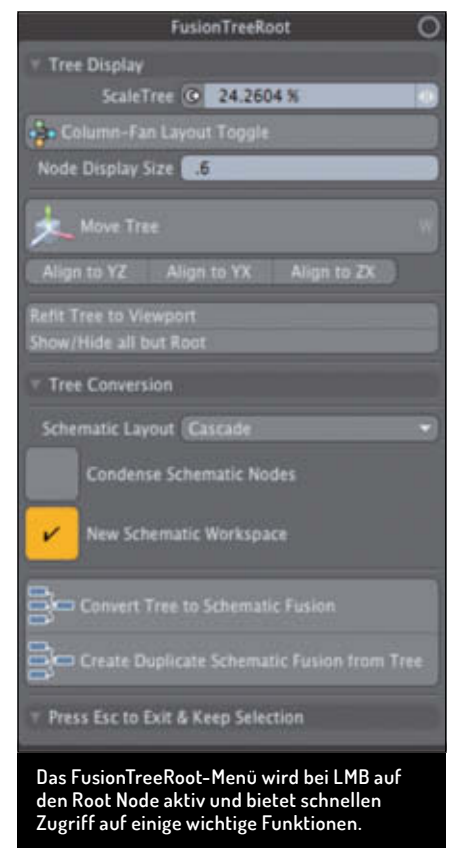
Auch ist die Tatsache, dass er sich an einer fixen Position in der 3D View befindet, nicht unbedingt von Vorteil, da man immer die Ansicht rotieren muss, wenn man Aktionen an ihm vornehmen möchte und er gerade unglücklicherweise hinter dem Objekt im Raum liegt. Wenn es „nur“ diese Kleinigkeiten wären, könnte man natürlich gerne darüber hinwegsehen, aber es gibt einige Szenarien, die es unerlässlich machen, ein Schematic Fusion zu verwenden. Eine sehr offensichtliche Situation, die auch noch im Kapitel „Fusion Flow – Arbeitseffizienz“ (ebenfalls nächste Ausgabe) genauer vorgestellt wird, ist die Verwendung von Instanzen. Versucht man nämlich, diese einem 3D Tree hinzuzufügen, wird man mit einer entsprechenden Meldung darauf hingewiesen, dass dies nur in einem Schematic Fusion möglich ist. Weiterhin lassen sich im Schematic Workspace zum Beispiel gezielt einzelne Parts im Fusion deaktivieren. Manche Fehler lassen sich etwa nur in einem Schematic Fusion

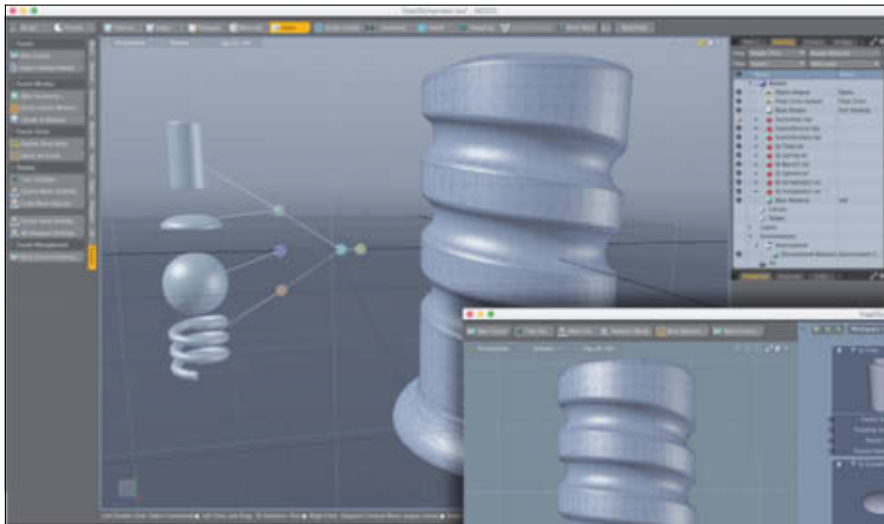
Quick Tree View!

An dieser Stelle vielleicht noch einmal der kleine Tipp, dass mit einem Klick auf den Fusion Root Node sich das „FusionTreeRoot“ Menü öffnet (siehe Bild unten). Dieses Menü bietet Zugriff auf die Skalierung des Trees, man kann die Art der Anzeige ändern, den Tree bewegen, seine Anzeige auf eine andere Ebene projizieren, über „Refit Tree to Viewport“ seine Anzeigegröße sehr schnell an einen veränderten Viewport anpassen, und – für dieses Kapitel wichtig – eine Konvertierung in ein Schematic Fusion vornehmen. Ist man sich nicht sicher, ob ein Schematic Fusion der richtige Weg ist, sollte die Option „Create Duplicate Schematic Fusion from Tree“ gewählt werden. Dadurch bleibt der Tree als Kopie bestehen, denn – wie gesagt – gibt es für diese Konvertierung kein Zurück.

lösen (siehe das Thema „Fehler im Fusion-Objekt bei sich durchdringenden Objekten“ im Abschnitt „Was nun – Fusion FAQ“). Auch bietet die Schematic View in komplexeren Konstrukten weitaus mehr Möglichkeiten, die einzelnen Aspekte eines Source Meshs zu nutzen. Diese Möglichkeiten werden im Folgenden beschrieben.

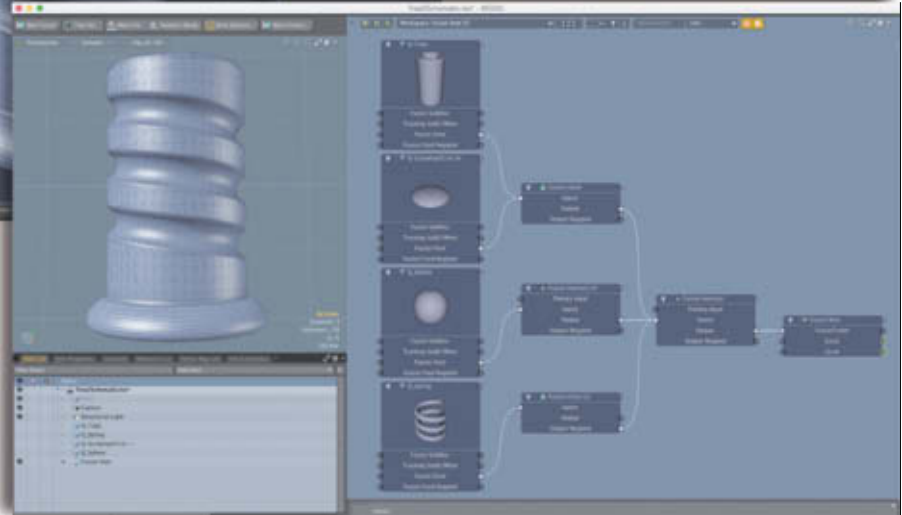
Wenn ein neues Fusion-Objekt als Schematic Fusion erzeugt wird, ist standardmäßig die Option „New Schematic Workspace“ aktiviert. In diesem Fall findet sich im Schematic Layout ein neuer Workspace, der den Namen des Fusion Items trägt. Ist die Option deaktiviert, wird das Fusion-Objekt im





Ein Fusion-Setting als 3D Tree und Schematic haben gleiche Züge, unterscheiden sich aber im Detail!

man die gewünschte Verbindung einfach irgendwo an der Connection-Line und „zieht sie ab“. Auf diesem Wege lassen sich an jeder beliebigen Stelle(!) die Verbindungen trennen und neu anordnen und so der Effekt ganzer Abschnitte auf das Ergebnis sehr schnell aktivieren oder deaktivieren. Dies ist ein Grund, warum ein Node-basiertes System so mächtig und flexibel ist. Denkbar wäre



Standard-Workspace erstellt.

Um das Handling mit Nodes zu verstehen, muss man sich einmal mit ihrem Grundprinzip auseinandergesetzt haben. Während im normalen Workflow Settings über Eingabefelder, Buttons und ähnliche interaktive Eingabefelder gesteuert werden, gibt es bei Node-basiertem Arbeiten entsprechende Knoten-Objekte. Diese haben einen Einfluss auf das Element, dem sie zugeordnet sind, und können seine Eigenschaften verändern, Werte definieren et cetera. Visuell werden die Nodes meistens durch variable kleine Fenster dargestellt, die die entsprechende Funktionalität anzeigen. Alle Nodes haben einen oder mehrere Outputs und unter Umständen einen oder mehrere Inputs, wobei diese optional sind. Ein Knoten kann erst einmal ohne Konnektivität in den Workspace eingebracht, dann nach Bedarf in einen Workflow eingegliedert und auch wieder ausgegliedert werden, ohne ihn dabei zu löschen. In Modo werden Nodes für die unterschiedlichsten Operationen eingesetzt. Sie können mathematische Berechnungen, Forces, Werteabfragen, Settings und etliche weitere Faktoren beeinflussen und können bei Animationen, Physics, Materials, vielen weiteren Aufgabenbereichen und eben auch MeshFusion angewandt werden. Vergleicht man bei einem einfachen Setting die beiden Strukturen, so werden erst einmal einige Parallelen ersichtlich.

Der 3D Tree „liest“ sich ausgehend vom Root, also hier von rechts nach links. Da er eine fixe Position hat, lässt er sich natürlich durch Rotation des Viewports auch ganz einfach drehen. Um die Parallelen aufzuzeigen, ist er aber hier einmal so dargestellt. Da die Mathematik im Hintergrund im Endeffekt die gleiche ist, könnte man in diesem Fall den Tree sogar von den Endpunkten aus

interpretieren, das würde aber der Funktionsweise einer hierarchisch aufgebauten Baumstruktur widersprechen.

Ein Node-basiertes System läuft meistens auf einen Ausgabe-Knoten (es sind auch mehrere „Outputs“ möglich) hin, wobei beliebig viele Knoten und Knoten-Konstrukte ihm einen Input dafür liefern. In Modo ist der Input für alle Knoten auf der linken Seite und auf der rechten Seite der oder die Outputs. Sollte links kein Input vorhanden sein, ist der Knoten selbst der Wert (Konstante) beziehungsweise der Input für den Folgeknoten. Innerhalb eines Knotens sind die In-/Outputs linear zu lesen und können nicht gekreuzt werden. Die Bezeichnung des Channels (in Modo der Name für die entsprechende Eigenschaft) steht dabei in der Mitte.

Die Knoten werden verbunden, indem man aus einem Connector (kleiner Kreis) auf der rechten Seite eines Knotens eine Connection per Drag-and-drop zu einem Connector oder dessen Channel-Namen auf der linken Seite zieht – oder umgekehrt. Relevant für Fusion-Objekte sind nur die runden Konnektoren. Die rautenförmigen stehen für Relationen und werden für ein Fusion-Setting nicht gebraucht.

Wenn man während des Draggings über einen passenden Connector fährt, zeigt einem die Connection-Line durch ihre Farbe, ob die Verbindung etabliert werden kann (grün) oder nicht (rot). Eine Connection kann nur erzeugt werden, wenn die Channels beider Knoten vom gleichen Datentyp sind. Um eine bestehende Connection zu lösen, fasst

auch, dass man sich verschiedene Stränge baut und diese nach Belieben schnell und einfach eingliedern kann.

Für unser Fusion Item bedeutet dies, dass das Gegenstück zu Drag-and-drop im 3D Tree in der Schematic View einfach das Konnectieren an einen anderen Node ist. Im Gegensatz zum Tree kann man für ein Source Mesh aber auch ganz einfach eine Connection trennen, ohne das Item aus der Schematic View zu entfernen. Ein Mesh im Tree kann nur durch ein „Remove“ deaktiviert werden und ist dann komplett aus dem Fusion-Objekt entfernt. Ein weiterer nicht im Tree vorhandener Vorteil ist die Möglichkeit, jedem Mesh weitere Channels bereitzustellen und mit diesen zu arbeiten.

Ein zusätzlicher wichtiger Channel für ein Fusion Mesh ist zum Beispiel „Fusion Feed Negated“. Im normalen Workflow werden die Parts innerhalb eines Fusion-Objekts immer mit ihrem „Fusion Feed“ verbunden. Möchte man Zugriff auf den negierten Fusion Channel bekommen, gibt es dafür zwei Möglichkeiten. Einerseits, da dies ein bei Fusion-Objekten immer wieder benötigter Channel ist, kann man einfach über das Fusion Pie Menu („Strg+F“) gehen, das in diesem Fall aus für die Schematic View angepassten Optionen besteht. Bei selektiertem Item kann dort per „More Schematic Fusion“ > „Add Other Feed Channels“ der negierte Feed Channel hinzugefügt werden. Andererseits, und das gilt für alle zu einem Item-Typ verfügbaren Channels, werden diese bei selektiertem Item im Tab „Channels“

angezeigt und können einfach in den Schematic Workspace gezogen werden ohne die Drop-Aktion direkt auf dem Item ausführen zu müssen (es reicht, das Workspace Fenster zu treffen). Der Channel wird dann wie zuvor sofort im unteren Bereich des Nodes angezeigt.

Alle Fusion Items sind mit ihrem „Fusion Feed“-Output an einen Mesh-Op-Input verbunden. Dabei ist die Anordnung der Nodes optisch zwar ähnlich der des 3D Trees, bei genauerer Betrachtung stellt man aber fest, dass es keinen „Subtract“ Node gibt. Vorhanden sind ein „Fusion Union“, ein „Fusion Intersect“ und ein zweiter „Fusion Union“ Node, der das Gegenstück zum „Subtract“ Node im Tree darstellt. Der Unterschied ist, dass dieser Node nicht mit seinem normalen Output, sondern mit seinem negierten Output Channel an den nachfolgenden „Fusion Intersect“ Node angebinden ist, der im Übrigen die Branch-Op darstellt.

Somit ist die Schematic View bezogen auf die Art der Einbindung der Elemente aussagekräftiger als der 3D Tree, in dem die Operationen ja nur durch die Punkte dargestellt worden sind. Der nachfolgende („Root“) Node ist wie schon beschrieben ein „Fusion Union“ und geht mit seinem Feed direkt in das „Fusion Item“. Diese Grundstruktur ist durch das Erstellen per „New Fusion Item“ in den meisten Fällen ähnlich, kann aber durch neue Nodes, die auch per Fusion Pie Menu eingefügt werden können, beliebig angepasst und erweitert werden.

Bei den vielen verschiedenen Möglichkeiten, die dieses System bietet, kann eine Schematic View ähnlich dem Tree auf eine erstaunliche Größe anwachsen. Im Gegensatz zum Tree gibt es aber hier sehr effiziente Möglichkeiten der Ansicht. Der größte Vorteil ist bestimmt, dass die Schematic View nicht an die Anzeigegröße des Fusion-Objekts im Viewport gekoppelt ist. Somit können beim Arbeiten der 3D Viewport und die Fusion-Struktur unabhängig voneinander betrachtet und bearbeitet werden. In der Schematic View bewegt man sich – verständlicherweise bis auf die nicht vorhandene Möglichkeit zu rotieren – mit den gleichen Gesten wie in einer 2D View im Viewport.

Die Selektion eines Mesh Nodes selektiert auch das Mesh im Viewport. Um die Miniaturansichten der Meshs auch in der Schematics zu haben, kann man diese Option über das kleine Zahnradsymbol in der rechten oberen Ecke via „Display Mesh/Material Previews“ aktivieren.

Eine weitere, nicht gleich offensichtliche Funktion bemerkt man erst, wenn man bei einem größerem Node-Konstrukt weiter heraus zoomt. Bewegt man dann den Cursor über einen Node und wartet einen kleinen Moment, wird dieser Node auf 100 Prozent

gezoomt. Bei aktivem Zooming, kann man dann direkt auf andere Nodes wechseln und sie werden umgehend vergrößert.

Auf diesem Weg kann man auch bei kleiner Ansicht recht schnell die Position eines bestimmten Nodes identifizieren. Um die Ansicht zu optimieren, empfiehlt es sich auch, das Positionieren der Nodes zu nutzen. Einzelne Nodes können einfach durch Drag-and-drop bewegt, mehrere können entweder bei gedrückter LMB durch „Berühren“ selektiert oder mit dem Rectangle Selection Tool via RMB markiert und dann gemeinsam verschoben werden.

Ansonsten können im 3D Viewport natürlich die vorher beschriebenen Drag-and-drop-Aktionen der Meshs weiterhin genutzt werden. Auch das Vorgehen, Meshs komplett aus dem Fusion zu entfernen, wird weiterhin durch Drag-and-drop des Source Meshs auf irgendein anderes Source Mesh und dann die Option „Remove Dropped from Fusion“ vollzogen. Im Schematic Workspace gibt es außer der „Delete“-Funktion keine Möglichkeit, das Mesh nur aus dem Fusion zu entfernen ohne es zu löschen.

Hier noch einmal der Hinweis, dass für alle Drag-and-drop-Aktionen die Source Meshs sichtbar sein müssen! Dies kann weiterhin sehr einfach via Fusion Pie Menu erledigt werden, im Schematic Layout aber mit dem Cursor über dem 3D Viewport. Hier ist das Pie Menu kontextsensitiv und richtet sich nach dem unter dem Cursor befindlichen Arbeitsbereich.

Somit kennen wir nun das Interface und die grundlegende Strategie von MeshFusion im neuen Modo 901 – im zweiten Teil in der nächsten Ausgabe lesen Sie, was es mit Fusion Strips, SubD, Boolean Sculpting, CAD Extend und Fusion UVs auf sich hat. Dazu verrate ich Ihnen Strukturen für effiziente Pipelines, den Ablauf von Fusion2Mesh sowie Tricks zu Optimierung und versteckten Funktionen. Bis dahin! > ei



André ist ein ausgebildeter Mediengestalter (B.Sc.) und arbeitet als Freiberufler mit 10 Jahren Berufserfahrung und 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

The Foundry – MeshFusion für Modo
▷ www.thefoundry.co.uk/products/modo/plugins/meshfusion

Braid Art Labs – GroBoto
▷ www.groboto.com/v3

PRODUCT
INNOVATION
LOUNGE

SOFTWARE, SERVICE & SUPPORT

MAUS SUCHT STARKE TYPEN

07543-609425

Modo 901, Fusion 360,
T-Splines für Rhino. Welcher
Software-Typ passt zu Ihnen?

Wir vereinbaren für Sie ein Date mit unserer neuen Software. Kennenlern-Schulungen, individuelle Beratung ... Oder kommen Sie doch einfach zu unseren ALIAS DAYS. Nächster Termin:

ALIAS DAYS 2015
28.09. / Gaimersheim
www.aliasdays.de

Erfahren Sie mehr über uns und unsere 3D-Software-Angebote unter
www.pilounge.de

Anzeige