

MEL-Scripting für die **CG-Produktion**

Anhand der letzten Kurzfilm-Produktion "I am Light" von sechs Studenten der Mediadesign Hochschule in Berlin wird in diesem Artikel das Vereinfachen einer 3D-Produktion in Maya durch das Scripts in MEL verdeutlicht. Die Bilder mit den Scripts können Sie sich in der DP-Augmented Reality App anschauen, oder im Webinhalt unter digitalproduction.com herunterladen.

von Luigi Pisciotta

aya Embedded Language, besser bekannt als MEL, ist die Scripting Sprache von Autodesk Mava. MEL versucht sich an der Programmiersprache PHP zu orientieren und ist im Aufbau wie auch in der Syntax der Programmiersprache C sehr ähnlich. Maya läuft und besteht u.a. aus mehreren tausend MEL-Scripts, die das ausführen, was wir gerade in Maya brauchen: von ganz einfachen Scripts (wobei man nicht zwingend Programmierkenntnisse braucht, um diese nachzubauen) bis zu komplexen Scripts, die uns stets behilflich sind, ohne dass wir diese bewusst als Scripts wahrnehmen. Diese Scripts kann man nicht nur durch "Ausführen" über den Script Editor laufen lassen, sondern man kann diese in Maya integrieren und durch GUIs (Graphical User Interface) aufrufen und

ausführen. Man kann sozusagen durch MEL-Scripts, Mayas Funktionen erweitern und verbessern. Seit Maya 8.5 wurde auch die Programmiersprache Python in dieser Software integriert.

Warum Scripten?

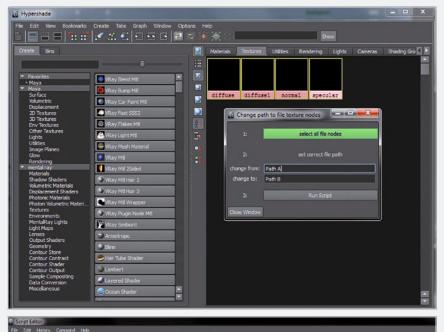
Die Vorteile von Scripts sind, dass alle Abläufe die man in Maya durchführen kann, über MEL ausgeführt und automatisiert werden können. In einer 3D-Produktion - sei es ein privates oder ein professionelles Projekt - gibt es sehr viele Schritte, die bearbeitet werden müssen. Häufig müssen ganz bestimmte Abläufe ständig wiederholt werden. In solchen Situationen ist es immer hilfreich, ein Script zu erstellen das diesen Ablauf automatisiert. Durch Scripting kann man Maya dazu bringen, alle repetitiven oder langwierigen Arbeiten zu erledigen oder Abläufe einer Produktion zu beschleunigen. Scripts werden ganz besonders beim Rigging benutzt, aber auch für die anderen Abschnitte einer Produktionspipeline sind sie sehr hilfreich, wie z.B. Szenenaufbau, Lighting, Rendering u.s.w.

Zeitersparnis

Ein gutes MEL-Script ist immer eines, dass dem Artist Arbeit abnimmt oder vereinfacht. Solche Scripts entstehen meistens erst nach der Feststellung einer Problematik oder eines repetitiven Ablaufs.

Zu Beginn scheint Scripting keine so große Zeitersparnis zu bringen. Denn anstatt sofort

02 WWW.DIGITALPRODUCTION.COM



an der Aufgabe zu arbeiten, schreibt man zunächst ein Script um, dieses Aufgabe später mit nur einem Klick zu erledigen? Das klingt auf den ersten Blick nicht wirklich produktiv, sondern eher nach einer "Nerd-Challenge". Doch Scripting ist wie ein Langzeit-Investment: es verkürzt zum Schluss die Produktionszeiten mehr, als es den anfangs scheint. In einer CG- oder VFX-Produktion ist Scripting nicht mehr wegzudenken.

MEL und Pipelining

In der Produktion "I am Light" war am Anfang nicht sicher, ob diese mit Mental Ray oder mit V-Ray durchgeführt werden würde. Solange sich die Produktion beim Modeling oder sogar im fortgeschrittenen Stadium der Animation befindet, ist es kein Problem. Zum Problem wird es, wenn man mit dem Shading und Lighting beginnt, oder wenn man Rendern will.

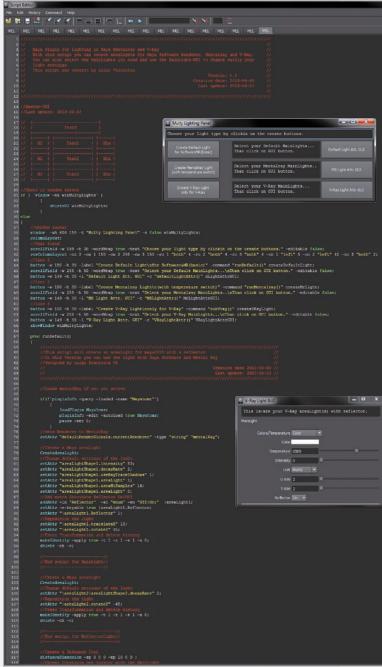
Die Gründe, weshalb es nicht klar war mit welchen Renderer gearbeitet werden sollte, lagen darin, dass V-Ray nicht bis zu dem Zeitpunkt in der Renderfarm integriert war und das Team bis dato wenig bis keine Erfahrung mit V-Ray für Maya hatte, aber trotzdem unbedingt diese eine letzte Produktion im Verlauf des Studiums, mit V-Ray durchführen wollte. Welche Möglichkeiten hat man in solche Situationen: man will etwas Neues ausprobieren und einsetzten, aber hat im Hinterkopf immer die Goldene Regel der Produktion: "Never change a running system".

Es gab also die Möglichkeit, in Mental Ray die Produktion anzufangen und kurz vor dem

MEL und Pipelining

```
Established Figure - commond "defaultions demands" reclaiments of the common of the co
```

```
Fig. 10. No. 1
```



SCIENCE & EDUCATION

Lighting mit MEL

Rendering alle Shader, Lichter und Rendersettings für V-Ray zu ändern, sobald V-Ray in die Farm integriert ist. Das hätte bedeutet, alles doppelt zu machen und die Deadline nicht einhalten zu können. Gleichzeitig wäre die Wahrscheinlichkeit höher gewesen, mehr Fehler einzubauen, die das Rendering über die Farm fast unmöglich gemacht hätten (z.B.: für den einen Renderer unbekannte Nodes vom zweiten Renderer oder zugewiesene Shader, die nicht für den Renderer zu verarbeiten sind). Nicht zu vergessen: das Lighting. Alle Lichter hätten neu gesetzt werden müssen.

Die zweite Möglichkeit bestand darin, V-Ray auf den eigenen Maschinen zu benutzen und eventuell kurz vor dem Rendering zu merken, dass wir für diese Produktion nicht V-Ray in die Farm integrieren könnten, oder evtl.

über die Farm Probleme bekommen können und aus Zeitmangel nicht beheben können. Die Gefahr dabei wäre, dass die Qualität nicht ausreichend ist oder dass die Renderzeiten sind zu hoch o.ä. und dann im letzten Augenblick auf Mental Ray zurückweichen um dann das Shading, das Lighting, und das Rendering neu zumachen.

Warum also nicht die zweite Möglichkeit auswählen und dabei trotzdem eine Ausweichmöglichkeit auf Mental Ray offen halten, ohne mehr Zeitaufwand oder das Risiko die Fehlerquote zu erhöhen?

Die Idee, einen solches Script zu schreiben war eine Schlussfolgerung einer schon früher bemerkten Problematik von V-Ray 2.0 für Maya: keine Prävisualisierung der Einflüsse der Lichter im Viewport – man muss ständig den IPR-Renderer laufen lassen (und der ist nicht gerade das stabilste Tool in Maya, zumindest nicht in der Educational Version). Mit einem sehr einfachen MEL-Script wurde es wieder hell im Viewport von Maya.

Ein Script für das Lighting mit V-Ray im Maya Viewport

Dieses Script, dass für die V-Ray Version 3.0 obsolet werden wird (aber für die 2.0 Version sehr hilfreich ist), erstellt ein V-Ray rectangleLight und ein Maya areaLight. Alle Werte des V-Ray-Lichts sind mit dem areaLight verknüpft: Farbe, Intensität, Größe, u.s.w. Um keine doppelten Lichter zu rendern, stellt das Script die Eigenschaften der Lichter so ein, dass nur der eine im Rendering sichtbar ist

04 WWW.DIGITALPRODUCTION.COM

```
proc runVray()

{

//This script will creace an arealight visible in viewport for mays2013 with a reflector

//In this Version you can use the light with Mays V-Ray

//Scripted by Luigi Pisciotta VI

// Creation date 2013-04-10 //

// Last update: 2013-04-13 //

// Last update: 2013-04-13 //

// automatically load V-Ray plug-in on script load if it's not loaded already

if (!('pluginInfo -q -l vrayformaya'))

{

loadPlugin vrayformaya;

pause -sec 5;

}

//Create a Distance Tool

distanceDimension -sp 0 0 0 -ep 10 0 0;

//Create V-Ray Rectangle Light

shadingNode -saLight VRayLightRectShape;

//Dange default stratumes of the light

setAttr "VRayLightRectShapel.invisible" 1;

setAttr "VRayLightRectShapel.invisible" 1;

setAttr "VRayLightRectShapel.subdive" 16;

// May series abstracted Reflectors on/Off

addAttr -ln Teelector" -at "enum" -en "Off:On:" |VRayLightRectl;

setAttr "VRayLightRectl.retates" 1;

// Reposition the light

setAttr "VRayLightRectl.retates" 45;

// Load Script for V-Ray MainLight//

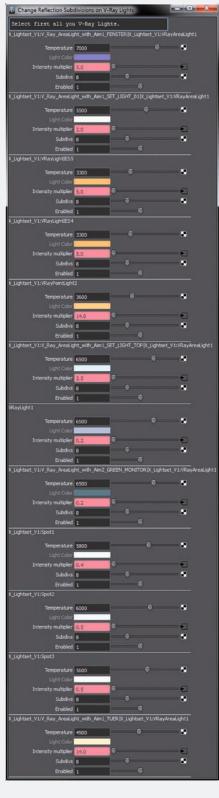
// Create V-Ray Rectangle Light

shadingNode -saLight VRayLightRectShape;

// Create V-Ray Rectangle Light

shadingNode -saLigh
```

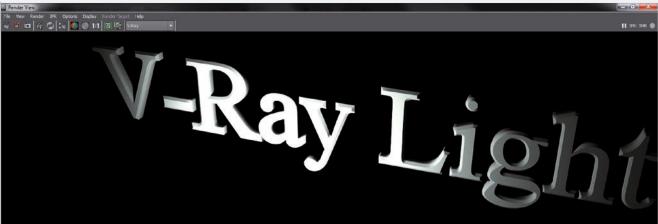
makeIdentity -apply true -t 1 -r 1 -s 1



und der andere nur im Viewport. Durch eine Maya Node werden die Intensitätswerte des V-Ray Lichts in die Intensitätswerte des area-Lights konvertiert, damit man im Viewport genau das sieht, was auch später im gerenderten Bild an Licht-Stimmung zu sehen ist. Dieses Script war die Basis für viele späteren Erweiterungen für die Produktion "I am Light".

Da man z.B. mit der Taste "t" in Maya eine Art Locator bei dem areaLight bekommt und





Lighting mit MEL

dieser sehr nützlich ist, dies aber beim rectangleLight nicht vorhanden ist, wurde im Script auch ein Locator erstellt, mit dem man das rectangleLight auf das gewünschte Objekt zielen lassen kann.Der Switch von V-Ray auf Mental Ray wurde eingebaut, um zusätzliche Mental Ray Nodes zu erstellen und zu verknüpfen, damit das Endresultat im Rendering mit Mental Ray dem entsprach, wie das Ergebnis mit V-Ray.

Ein zweites rectangleLight wurde als Reflektor/Fill-Light eingebaut (natürlich auch mit zusätzlichem areaLight). Über den Distance Tool und einer Set Range Node wurde der Abstand des Key-Lights (also des Hauptlichtes) festgestellt - und dementsprechend wurde die Intensität des Reflektor-Lichtes eingestellt. Der Key-Light und der Reflektor wurden miteinander so verknüpft, dass beide über einen Locator das Objekt anvisieren und

beim Bewegen des Key-Lights der Reflektor folgt, aber immer noch selbständig nachjustiert werden kann.

Die Fertigstellung des Basis Scripts mit Beta-Testerfolgte an nur einem Tag. Innerhalb einer Woche entstanden alle anderen Erweiterungen. Eine Szene zu beleuchten brauchte nun, durch das Verzichten vom IPR-Renderer, nur die Hälfte der Zeit und erzeugte keine Abstürze von Maya. Der Regisseur konnte gleich im Viewport Anweisungen geben und deren Veränderungen sehen, ohne auf Renderings warten zu müssen. Dadurch wurden auch die Zeiten, die man für eine Abnahme oder Veränderung braucht, verkürzt.

Lighting mit MEL

Gerade was Änderungswünsche beim Lighting angeht, wenn die Prävisualisierung jetzt durch den vorherigen MEL Script so schnell geht, sollte man auch eine Methode finden diese zu ändern, ohne jedes Licht einzeln selektieren zu müssen um es im Attribute Editor zu steuern.

Mit MEL kann man nicht nur schreiben was Maya erstellen und verknüpfen soll, sondern auch bestimmte Objekte suchen lassen, diese selektieren, filtern, usw. - alles was man normalerweise per Hand macht, kann über ein Script laufen. Somit wurde zum Beschleunigen des Lightings in der Produktion "I am Light" ein Script geschrieben, der genau das macht: Beim Ausführen des Scripts wird in der Szene nach Lichtern gesucht, und nur die V-Ray Lights werden selektiert. Eine GUI, die auch durch das MEL-Script erstellt wird, erscheint und enthält alle wichtigen Attribute von jedem selektierten Licht. Somit ist man in der Lage, schnell die Intensität und Farbe/Temperatur oder die Größe jedes Lichtes über nur ein Fenster zu verändern. Somit erhält man durch das Script für das Lighting im Viewport mit V-Ray sofort ein Feedback, was bedeutet, dass man dem Regisseur direkt die Änderungen zeigen und das Lighting der Szene abnehmen lassen kann.

Nach eineinhalb Tagen war dieser Script produktionstauglich. Die Zeitersparnis war unglaublich. Um eine Szene gut und schön zu beleuchten, so dass diese auch in einer animierten Szene funktioniert, hätte man in dieser Produktion drei bis fünf Tage gebraucht – also fast eine Woche pro Szene mit Änderungen und Abnahme. Nur anhand dieser zwei Scripts konnte man eine Szene pro Tag final abnehmen lassen und im Falle der Fälle in weniger als fünf Minuten alle Lichter auf Mental Ray umstellen.

Shader Switch

Um die Pipeline so aufzubauen, dass man eine komplette Szene von V-Ray auf Mental Ray ein-

fach per Knopfdruck umstellen kann, gehören nicht nur Scripts für die Umstellung der Lichter, sondern auch die Shader und die Rendereinstellungen müssen geändert werden.

Die Shader neu zu erstellen ist die langwierigste Arbeit. Die Nodes neu zu verknüpfen, ist der einfachste Part davon, doch auch sehr mühselig, wenn man 30 bis 40 Shader mit jeweils zwei bis drei Nodes verknüpfen muss. Mit einen sehr einfachen kurzen MEL Script kann man die Verknüpfungen von einem Shader zum anderen umleiten. Man braucht lediglich in der richtigen Reihenfolge die gewünschten Shader und Nodes zu selektieren und das Script auszuführen.

Es müssen aber auch noch die ganzen Einstellungen des Materials geändert werden. Da kann zwar ein Script etwas helfen, aber nichtsdestotrotz bleibt es eine Aufgabe, indem man visuelle Vergleiche machen muss, da V-Ray Materials anders sind und anders auf Licht reagieren wie Mental Ray Shader. Für so eine Situation kann man lediglich ein paar Werte über einen Script übertragen, wenn es schnell gehen muss. In diesem speziellen Fall stößt man allgemein an die Grenze des Scriptings sobald es um eine Aufgabe geht, die mit visuellem Empfinden zu tun hat. Ein "Mach-es-schöner-Button" ist leider noch nicht über Scripting möglich.

Pfad-Chaos

Um eine schnellere Verbindung zum Server zu erreichen, wurde dieser an der Mediadesign Hochschule in Berlin, in Projekt-Server und Render-Server geteilt. Somit konnte auf einer 2Gbit-Leitung an den Szenen gearbeitet werden und auf der anderen 2Gbit-Leitung gerendert werden. Doch obwohl man in Maya ein Projekt setten kann, um alle Ordner mit Referenden, Textur- oder Cachepfade bestimmen kann, werden diese öfters als absolute Pfade geschrieben - und beim Verschieben des Projektordners muss man alle Verknüpfungen neu machen.

Da wir vor dem Rendering auf einen Server gearbeitet haben und für das Rendering selber auf einem anderen arbeiten mussten, wurde ein Script geschrieben, der nach dem Verschieben des Projektordners alle Pfade ändert. Somit wurde sichergestellt, dass alles, was an einem Pfad gebunden ist, von Maya gefunden und auch gerendert wird.

Rendering Einstellungen übertragen

Auch wenn es im Renderer verschiedene Einstellungsmöglichkeiten gibt, benutzt man in einer Produktion meistens fast dieselben Einstellungen. Je nach Szene kann man Werte verkleinern oder vergrößern, um die beste Qualität in der kürzesten Renderzeit zu erzeu-

gen. Einige Renderer können sogar Einstellungen speichern, und man lädt diese lediglich in die neuen Szene hinein. Im Laufe der Zeit kann man Rendereinstellungen, die man am häufigsten anwendet oder die man als Basis öfters benutzt, über einen MEL Script aufrufen. In den ersten Zeilen des Scripts sollte man zur Sicherheit immer den jeweiligen Renderer laden. Danach sieht das Script fast wie eine Liste an Eigenschaften aus. Somit hat man immer die besten Rendereinstellungen und die Render Passes für jeden benutzten Renderer immer per Knopfdruck bei sich und braucht lediglich ein paar Werte zu ändern. Dies spart meistens bis zu einer halben Stunde Arbeit.

Schneller durch Maya – Einen Shader bauen durch MEL

Bei der Produktion "I am Light", aber auch bei vielen anderen Produktionen, haben die Shader fast immer dieselbe Basisstruktur: eine Color Map (auch über eine Average Node in den Color-Attribut verknüpft), eine Normal Map, Displacement oder Bump Map und eine Specular oder Reflection Map. Natürlich baut man für die Hauptcharaktere komplexere Shader, aber was die Props und das Environment angeht, ist die Herstellung von solchen Shadern eigentlich immer gleich - man wählt eine Shader Node, man verknüpft ein paar Texturen und ändert die Werte. Wenn man viele von diesen Shadern erstellen muss, dann kann es mühselig sein, immer wieder dieselben Prozesse durchzuführen, und man hat zum Schluss wenig Zeit sich richtig auf die Einstellungen des Shaders selbst zu konzentrieren oder etwas zu experimentieren.

Mit einem sehr einfachen kurzen MEL Script kann mit einem Klick nicht nur ein V-Ray Standard Material erstellt werden, sondern auch gleich mehrere File Nodes. Was noch zu tun bleibt, ist lediglich die richtigen Dateien zu verlinken. Die Zeitersparnis liegt bei ca. fünf Minuten pro Shader.

Jetzt hat man z.B. bei einem zugewiesenen Shader mit drei Files, insgesamt sechs Nodes: die Shading Group, der Shader, die drei File Nodes und eine Place2dTextur Node. All diese Nodes müssten ietzt einzeln benannt werden. damit die Szene auch ordentlich bleibt. Durch ein zweites Script selektiert man einfach das schon benannte 3D-Modell, auf das man den Shader zugewiesen hat, Selektiert alle Shadingnodes und führt das Script aus. Das Script liest den Namen des Modells und überträgt es nicht nur an den Nodes, sondern ergänzt diesen mit der richtigen Bezeichnung. Bei einem Shader z.B. für einen Tisch, wird der Shader mit diesem Script "Table_Mtl" genannt, die Color Map "Color_Table", die Normal Map "Normal_Table", bis alle Nodes unbenannt sind.

DP



Szenen aufbauen über Scripts

In jeder Produktion modelliert man Objekte, erstellt Shader, rigged und animiert in einzelnen Dateien. Man erstellt dann eine neue Maya Datei, benennt sie z.B. "Shot 02" und lädt alles, was in dieser Szene gebraucht wird als Proxy, Referenz - oder importiert diese direkt.

Shader bauen durch MEL

//Create AverageNode by Luigi Pisciotta VI 05.02.2013

//This script creates an average node and connects the textures to it

//Select 2 textures and run the script

string \$select[] = 'ls -sl';

//Create AverageNode

shadingNode -asUtility plusMinusAverage;

//Connects the textures to the AverageNode

connectAttr -force (\$select[0] + ".outColor") plusMinusAverage1.input3D[0];

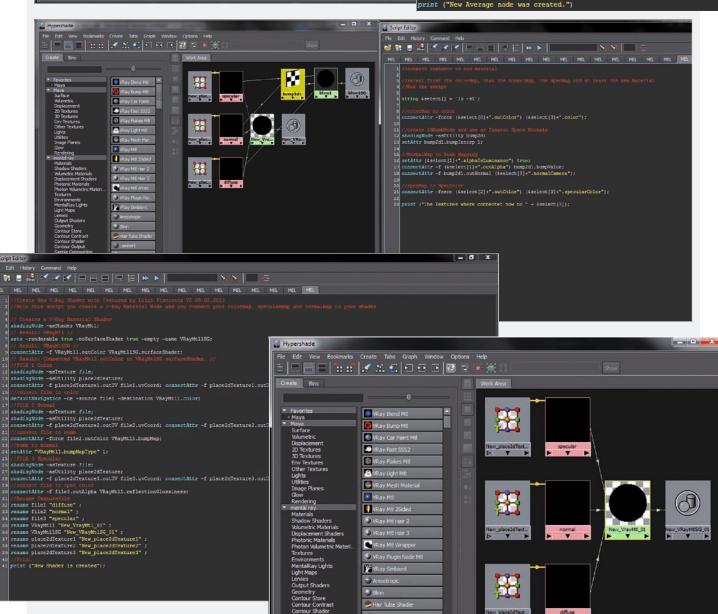
connectAttr -force (\$select[1] + ".outColor") plusMinusAverage1.input3D[1];

//Renalming

rename plusMinusAverage1 "Average_01";

//End

print ("New Average node was created.")



8 WWW.DIGITALPRODUCTION.COM

Das kann manchmal auch länger dauern, wenn man viele Props hat, da man jede Datei einzeln hinein laden muss.

Durch ein Script kann man Maya befehlen, in bestimmte Ordner nach beliebigen Dateien nachzuschauen und diese in der Szene hineinzuladen. Wenn ein Artist jedoch unsauber gearbeitet hat und das finale Objekt, bestehend aus mehreren Polygon-Objekten, nicht in einer Obergruppe zusammengefasst hat, bekommt man beim hineinladen solcher Dateien im Outliner eine lange Liste an Objekten, die nicht nur für Unordnung sorgen, sondern später beim evtl. Lightlinking mühselige Arbeit erschaffen.

In diesem Script wird deswegen jede Datei, die in Maya geladen wird gruppiert und so benannt wie der Dateiname selber – genau wie beim manuellen Referenzieren oder Importieren externer Dateien. Diese Szenen gehen erstmals an den Animator, der den oder die Charaktere und evtl. Gegenstände in dieser Szene animieren muss. Der Animator sollte nicht unbedingt beim Versuch, einen Controler zu selektieren, ein Objekt der Szene selektieren und diesen aus Versehen verschieben oder keyen.

Um so etwas zu vermeiden, erstellt man normalerweise Display-Layers in denen man Objekte zuweist, um diese schnell von sichtbar auf unsichtbar stellen zu können (falls nicht gebraucht) und als Referenz oder Template anzeigen zu lassen, damit man nicht aus Versehen diese selektieren und verschieben kann. Daher wird in diesem Script für jede importierte Datei ein Display-Layer erstellt und so benannt wie die Gruppe selber. Nun braucht man lediglich die Objekte in der Szene richtig zu positionieren. Dies könnte man auch scripten, wenn man von Anfang an weiß, wo diese im 3D-Raum hin müssen.

Rendering aufsetzen

Wie schon erwähnt, kann man mit MEL Eigenschaften vom Rendering übernehmen. Doch es bleibt trotzdem mühselig die ganzen Passes zu aktivieren und einzustellen. Auch das ist mit MEL lösbar.

Was ist aber mit komplexeren Passes wie z.B die Multimatte in V-Ray? Um diese korrekt benutzen zu können, müssen die Objekte oder Materialien, die später als separate Alphamasken gebraucht werden, erstmals eine Object-ID oder eine Material-ID bekommen. In der Kurzfilm-Produktion "I am Light" wurden dem Compositing fast 50 Alphamasken geliefert – mehr als 15 Multimatte-Passes. Das bedeutet, dass man bei fast 50 Objekten und Materialien eine Object-ID oder Material-ID erstellen muss und eine fortlaufende Nummer vergeben muss.

Durch einen sehr einfaches Script braucht man nur die Objekte zu wählen, die eine ID-

3D & ANIMATION

nen erweitert und so umgeschrieben, dass die gewünschten Objekte immer die selbe ID bekommen. Somit hält man vorsorglich auch Ordnung im Compositing, und kann dort auch evtl. über einen Python Script im gleichen Compositing-Baum nur die Readnodes ersetzen.

Resultat

DIGITAL ART

In der Produktion "I am Light" wurden mehr als 20 MEL Scripts geschrieben: für gezielte Selektion, Umbenennung, Shading und ganz besonders für den Szenenaufbau, das Pipelining, Lighting und Rendering. Ohne diese Scripts hätte die Produktion dieses Projektes sicherlich drei bis fünf Wochen länger gedauert. Um so viel Zeit durch diese



```
Script Editor
    Edit History Command Help
setAttr "defaultRenderGlobals.enableDefaultLight" 0;
       setMayaSoftwareFrameExt(3 , 0);
setAttr "defaultRenderGlobals.extensionPadding" 4;
       setAttr "miDefaultOptions.scanline" 0;
       setAttr "miDefaultFramebuffer.datatype" 4;
       setAttr "defaultResolution.width" 1920;
       setAttr "defaultResolution.height" 1080;
       setAttr "defaultResolution.deviceAspectRatio" 1.777;
       setAttr "defaultResolution.pixelAspect" 1;
      setAttr "perspShape.displayResolution" 1;
setAttr "perspShape.displayGateMaskColor"
setAttr "perspShape.displaySafeAction" 1;
       setAttr "perspShape.displaySafeTitle" 1;
```

Rendering

Nummer benötigen. Das Script erstellt mehrere Multimatte-Passes, erzeugt den V-Ray Extra Attribut Objekt-ID oder Material-ID (je nach Bedarf) und öffnet eine GUI mit den aufgelisteten Objekten, Gruppen oder Materialien mit dem default Wert "0".

Mit dieser GUI kann man jedem Objekt eine ID-Nummer vergeben. Dass man die Nummern per Hand vergeben muss, liegt daran, dass man evtl. einige Objekte oder Materialien in einer bestimmten Multimatte zusammen haben will. In den Multimatten von "I am Light" sollten z.B. Augen, Mund

und Zunge des Jungen in einer Multimatte und Kopf, Fingernägel und Hände in einer anderen

Da das Script Multimatte-Passes erstellt mit fortlaufenden ID-Nummern muss man daher für z.B. die zweite Multimatte den gewünschten Objekten die Werte "4", "5" und "6" vergeben. Dass hilft auch dem Compositer schneller zu zu arbeiten, weil er genau weiß, in welcher Multimatte welche Masken sind. Aber die Arbeit jedem Objekt eine ID-Nummer zu vergeben, macht man auch nur das erste Mal. Das Script wurde für die restlichen SzeScripts zu sparen, musste man sich nicht nur mehrere Wochen mit dem Herstellen dieser Scripts befassen, sondern auch sehr sauber und sorgfältig arbeiten, damit diese immer einsetzbar sind. Es wurden z.B. Scripts geschrieben die nur für diese bestimmte Produktion bestimmt waren und auch nur in dieser funktionieren. Andere wurden so verbessert, dass sie für andere Hochschul-Produktionen jetzt im Einsatz sind und sich somit in einige Arbeitsprozesse in Maya auch bei anderen Kollegen und Kommilitonen etabliert haben.