

Möglichkeiten der 3D-Charakteranimation auf Webseiten

Diplomarbeit

zur Erlangung
des akademischen Grades

Diplom - Informatiker

an der

Fachhochschule für Technik und Wirtschaft Berlin

**Fachbereich Wirtschaftswissenschaften II
Studiengang Angewandte Informatik
Schwerpunkt Multimedia**

1. Betreuer: Prof.Dr.-Ing. Thomas Jung
2. Betreuer: Herr Jürgen Muscheller

Eingereicht von

Volker Heisterberg

14.5.2001

Inhaltsverzeichnis

1. Vorwort.....	5	9.5 Wild Tangent´s Webdriver.....	33
2. Einige Begriffsdefinitionen und Erläuterung von 3D-Prinzipien.....	6	10. Tabellarische Gegenüberstellung.....	35
3. Die Geschichte von Web3D.....	9	11. Fazit der Gegenüberstellung.....	39
3.1 Von VRML zu Web3D.....	9	11.1 Objektvisualisierung.....	39
3.2 Die Anfänge der Webisodes.....	10	11.2 Charakteranimation.....	39
3.3 Eigene Webisode - Überlegungen.....	10	12. Standardisierungsinitiativen.....	40
4. Verwendete 3D-Programme.....	11	12.1 Die H-Anim Workgroup.....	40
4.1 3ds max 4 und Charakter Studio.....	11	12.2 MPEG 4.....	41
4.2 Maya.....	13	13. Implementierungsvorgang: Learning Tai-Chi with E-sha.....	42
5. Der Charakter E-sha.....	14	13.1 Anforderungsanalyse und Technologieauswahl.....	42
5.1 Wer ist E-sha?.....	14	13.2 Konzeptionsphase.....	43
5.2 Nurbs to Polygon.....	15	13.2.1 Tai-Chi E-sha.....	43
5.3 Generierung der Texturkoordinaten.....	15	13.2.2 Freie Kamera oder nicht?.....	43
5.4 Die Textur.....	15	13.3 Drehbuch, MoCap und Audioaufnahmen.....	44
6. Grobe Unterteilung der verschiedenen Technologien.....	16	13.3.2 Motion-Capture für Tai-Chi.....	44
7. Fake3D.....	17	13.3.2 Datengewinnung für die Dialogsequenzen.....	44
7.1 Flash.....	17	13.4 Nachbearbeitung der MoCap - Daten und damit verbundene Probleme.....	46
7.1.1 Vorstellung von Flash.....	17	13.5 Binding mit Charakter Studio.....	47
7.1.2 2D in Flash?.....	17	13.6 Generierung der einzelnen Phoneme für Lip-Sync.....	48
7.1.3 Von 3D nach 2D.....	17	13.7 Pulse Creator.....	49
7.1.4 Illustrate in der Praxis.....	17	13.8 Export von 3ds max nach Pulse.....	50
7.1.5 Fazit.....	18	13.9 Lipsync in Pulse.....	51
7.2 Virtual Friends.....	19	13.10 Problemstellungen in Pulse3D und Lösungswege.....	52
8. Systeme zur Objektvisualisierung.....	20	13.10.1 E-sha langsam einblenden.....	52
8.1 Viewpoint´s Experience Technology.....	20	13.10.2 Interaktion durch Kopfklicken.....	52
8.1.1 Die Geschichte von VET.....	20	13.10.3 Interaktion mit dem Schwert.....	53
8.1.2 Die Technik hinter VET.....	20	13.10.4 Die Kameraproblematik.....	53
8.1.3 Zukunftsausblicke.....	21	13.10.5 Wechselnde Idle-Behavioerrs.....	54
Cult3D (Cycore).....	22	13.11 Entwurf des Webinterfaces.....	55
8.2.1 Technologievorstellung.....	22	13.11.1 Interfacegestaltung.....	55
8.2.2 Cult3D in der Praxis.....	22	13.11.2 Multimediakomponente.....	55
8.2.3 3DmeNow.....	23	13.11.3 Probleme mit Netscape.....	55
8.3 Open-SPX (Interzart).....	24	13.12 Pulse <-> Browser Kommunikation.....	56
8.4 Hyper Space (Kaon Interactive).....	25	13.13 Projektfazit.....	56
9. Systeme mit Möglichkeiten zur Charakteranimation.....	26	14. Vorstellung der bekanntesten Avatare....	57
9.1 Shout3D.....	26	15. Eigene Überlegungen zu Einsatzmöglichkeiten.....	60
9.1.1 Technologievorstellung.....	26	16. Intelligenz für Avatare.....	61
9.1.2 Shout3D in der Praxis.....	27	16.1 Kiwilogic.....	61
9.2 Pulse3D.....	28		
9.3 B3D (Brilliant Digital).....	29		
9.3.1 Technologievorstellung.....	29		
9.3.2 B3D in der Praxis.....	30		
9.4 Virtools 2.0.....	31		

16.2 Artificial Life.....	61
16.3 Novomind.....	61
17. Zukunftsausblicke.....	61
17.1 Intels Internet-3D-Technologien und Shockwave3D.....	62
17.2 Der Geforce III-Chip.....	64
Anhang A: Fachbegriffserklärungen.....	65
Anhang B: Literaturliste.....	68

Inhalt der CD:

Webesha V0.5 Diplom Edition

Systemvoraussetzungen:

Internet Explorer 5
ab Pentium II 300
3D-Beschleuniger optional
Netzzugang für Plug-In Download
Administrationsrechte für Plug-In Installation

Zum Start bitte index.html öffnen.

1.Vorwort

Als ich während meines Studiums mit **Java3D** in Berührung kam, war ich sofort begeistert von der Möglichkeit, 3D-Modelle, die ich in *alias/wavefronts Maya* erstellt hatte, über das **OBJ**-Format in eigene Applikationen einzubinden. Zwar waren kaum Möglichkeiten zu anspruchsvolleren Animationen vorhanden, doch in mir keimte die Idee, in Maya erstellte Animationen auf Webseiten einzusetzen.

In Robert Polevois **Webreference**¹, einer webbasierten 3D-Kolumne, fand ich Bestätigung dafür, dass meine Überlegungen in die richtige Richtung führten: Polevoi zeichnete ein Bild des ‚neuen‘ 3D-Artists [rp 1]: Im Gegensatz zur Marktsättigung im High-End-Bereich sieht er die Zukunft viel mehr in interaktiven Applikationen, sei es nun für das Web oder den Spielmarkt, der sich durch Hochleistungskonsolen wie *Sony's Playstation II*² oder *Microsoft's X-Box*³ enorm vergrößern soll. Vom Artist selber erwartet er, dass er sich diesen neuen Herausforderungen stellt und das nötige Know-how aneignet, um seine Position im Markt der Zukunft zu sichern. Als aufmerksamer Beobachter des Bereichs **Web3D** ließ er es sich nicht nehmen, alle ihm bekannten neuen Technologien in seiner Artikelreihe kurz vorzustellen. Seine Begeisterung vor allem für **Shout3D** führte dazu, dass er mittlerweile selber bei *Eyematic*⁴, einer Firma, die durch den Erwerb von *ShoutInteractive* in den Besitz dieser Technologie gelangte, beschäftigt ist und bereits ein Buch [Sho1] über Shout3D veröffentlicht hat.

Nachdem ich mein Diplomthema bereits angemeldet hatte, erschien die Ausgabe 3/2000 der *digitalproduction*⁵ mit Neuigkeiten von der **Siggraph** [dp 00a] und siehe da: Web3D hatte sich zum Hauptthema der Messe gemauert [ct 1] und die Zahl der Neuvorstellungen von Softwarelösungen zur Verwendung von 3D-Modellen im Web übertraf meine Erwartungen bei weitem.

Für meine Diplomarbeit hatte ich zwei Aufgaben gewählt: Die erste war, einen Vergleich der verschiedenen Web3D-Technologien anzustellen, der als Entscheidungsgrundlage für Entwicklungen in diesem Bereich dienen kann.

Die zweite war, eine geeignete Lösung für Echtzeit-Charakteranimation auf einer Webseite zu finden und eine Beispielimplementierung vorzunehmen.

Dazu bot sich mir die Gelegenheit, bei *ID-TV*⁶, einer 100%igen Tochter der *I-D Media AG*⁷, ein Web3D-Projekt, das von Charakter-Animation Gebrauch machte, zu planen und durchzuführen. In diesem kam eine bestehende Figur aus der Avatar-Familie von *I-D Media*, die Halbasiatin **E-sha**, zum Einsatz.

¹ <http://www.webreference.com/3d/>

² <http://www.playstation.com/>

³ <http://www.x-box.com/>

⁴ <http://www.eyematic.com/>

⁵ <http://www.digitalproduction.de/>

⁶ <http://www.id-tv.de/>

⁷ <http://www.id-media.com/>

2. Einige Begriffsdefinitionen und Erläuterung von 3D-Prinzipien

Um den Einstieg in die komplexe 3D-Welt zu erleichtern, erscheint es mir sinnvoll, zunächst einige Grundprinzipien zu erläutern. Bei 2D-Grafiken genügen 2 Koordinaten, um einen Punkt genau zu spezifizieren. Bei räumlichen, dreidimensionalen Grafiken wird jeder Punkt durch 3 Koordinaten spezifiziert. Deshalb ist es nötig, um die Lage eines Punkt genau zu beschreiben, mit mehreren Ansichten zu arbeiten: Wenn ich z.B. sowohl eine Frontal- als auch eine Seitenansicht des gleichen virtuellen Raumes verwende, kann ich eine Position genau bestimmen. Um festzulegen, welche Achse nun eigentlich nach oben und welche in die Tiefe zeigt, verwenden 3D-Welten ein Weltkoordinatensystem, dessen Mittelpunkt in der Mitte des Raumes liegt. Einzelne 3D-Objekte werden zur Darstellung aus vielen Dreiecken zusammengesetzt.

Um 3D-Objekte zu erstellen gibt es eine ganze Reihe von Programmen: **3ds max** von *discreet*¹, **Maya** von *alias/wavefront*², **Softimage**³ von *Avid*⁴, **Lightwave** von *Newtek*⁵ und **Cinema4D** von *Maxon*⁶ sind wohl die bekanntesten.

Die Erstellung von 3D-Objekten wird als **Modelling** bezeichnet. Dazu werden verschiedene Ansätze verwendet:

Das **polygonale Modelling** ist wohl das intuitivste: Durch die Verbindung mehrerer Punkte zu Vielecken werden Flächen beschrieben. Intern werden diese Flächen so in Dreiecke zerlegt, dass alle Eckpunkte in einer Ebene liegen, wobei die Oberflächennormale die Richtung der Fläche vorgibt. Ein aus mehreren Polygonen zusammengesetztes Gitternetz wird als **Mesh** bezeichnet. Die Eigenschaft von Polygonen liegt darin, dass zwei Punkte (**Vertices**) immer durch eine Kante (**Edge**) verbunden werden. Deshalb wird die Darstellung einer

Mesh immer eckig sein, wenn man nur nahe genug an das Objekt herangeht.

Um Freiformflächen zu erstellen, verwendet man häufig **NURBS (Non-Uniform Rational B-Splines)**. Bei diesem Geometrietyt liegen die beschreibenden Punkte nicht unmittelbar auf der Oberfläche, sondern nur in deren Nähe. Um NURBS darstellen zu können, werden diese auch in Dreiecke unterteilt. Diesen Vorgang bezeichnet man als **Tesselation**. Diese ist allerdings dynamisch. Je näher der Betrachter an das Objekt herangeht um so feiner wird die Dreiecksunterteilung. Für den Betrachter entsteht so der Eindruck absolut weicher Konturen.

Ein in den letzten Jahren immer mehr in Mode gekommener Ansatz sind die **Subdivide Surfaces**. Bei diesem Verfahren dient eine einfache polygonale Struktur zur Beschreibung einer Freiformfläche.

Für Film- und TV-Produktionen erstellte 3D-Objekte werden zunächst **gerendert**. Darunter versteht man die Berechnung eine photographischen Abbildung der 3D-Szenerie. Je nach verwendetem Renderer können verschiedene lang dauernde Berechnungsverfahren wie **Raytracing**, die Verfolgung der einzelnen Lichtstrahlen, oder **Radiosity**, ein Verfahren, das den Lichtwurf der einzelnen Objekte berücksichtigt, zum Einsatz kommen [rad 1] [rad 2]. Um dieses darzustellen, sind zwar keine 3D-Operationen mehr nötig, dafür geht aber auch jegliche Interaktivität verloren. Bis auf *Pixars*⁷ **Renderman** tessellieren alle mir bekannten Renderer NURBS-Objekte zur Berechnung.

¹ <http://www.discreet.com/>

² <http://www.aliaswavefront.com/>

³ <http://www.softimage.com/>

⁴ <http://www.avid.com/>

⁵ <http://www.newtek.com/>

⁶ <http://www.maxon.de>

⁷ <http://www.pixar.com>

Um interaktive 3D-Grafiken darzustellen ist immer eine **Engine** nötig, welche die zur Veränderung der Darstellung nötigen Berechnungen in Echtzeit vornehmen kann. Solche Engines findet man z.B. in 3D-Spielen. Wie ich in dieser Diplomarbeit beleuchten möchte, gibt es aber auch eine Reihe von Möglichkeiten, 3D-Darstellungen auf Webseiten zu verwenden.

Die Darstellungsqualität eines aufwendigen Renderings kann dabei natürlich noch nicht erreicht werden. Damit auch weniger leistungsfähige Grafikkarten oder CPUs die benötigten Transformationen in einer für den Betrachter als flüssig erscheinenden Geschwindigkeit berechnen können, ist die Zahl der Dreiecke möglichst gering zu halten.

Da das Echtzeit-**Shading**, die Berechnung von Schattierungseffekten, sehr rechenintensiv ist, verwenden echtzeitfähige Engines häufig sogenannte **Lightmaps**: Dabei dient die Darstellung einer beleuchteten Kugel als Referenz für die Beleuchtung der Szenerie. Wenn man z.B. ein Objekt dreht erscheint dieses dynamisch schattiert (**geshadet**). Da die Lichtsituation festgelegt ist, hält sich der Berechnungsaufwand in Grenzen.

Um auch primitivere Modelle detaillierter erscheinen zu lassen, werden die Objekte häufig mit aufwendigen **Texturen** versehen. Damit die Engine weiß, welcher Texturpunkt (**Texel**) an welchem Punkt auf die Fläche zu setzen ist, wird ein 2D-Koordinatensystem über die zu texturierenden Flächen gezogen, die sogenannte **UV-Map**. Diese UV-Koordinaten sind jetzt nur noch den einzelnen **Texeln** zuzuordnen.

Bewegungen steifer Objekte sind am einfachsten durch das Festlegen von Schlüsselbildern (**Keyframes**) über die Zeit zu animieren.

In der Realität verformt sich aber z.B. ein Ball wenn er auf den Boden knallt, ein steifer Ball erscheint unnatürlich. Um so etwas darstellen zu können, müssen die Animationsdaten nicht nur für das Gesamtobjekt sondern für jeden einzelnen Punkt berechnet und ausgewertet werden. Man spricht von **Vertex Animation**.

Ein Schwerpunkt dieser Arbeit stellt die **Charakteranimation** dar. 3D-Charaktere werden in der Regel über ein mit den Geometriedaten verbundenes Skelett animiert, das aus mehreren Knochen (**Bones**) besteht, die durch Gelenke (**Joints**) verbunden sind. Bei der **forward kinematik** ist es nötig, einen Joint nach dem anderen zu rotieren. **Inverse kinematik** ermöglicht es, den Endjoint einer definierten Kette an seine neue Position zu schieben. Die restlichen Joints der Kette folgen dann automatisch der Bewegung.

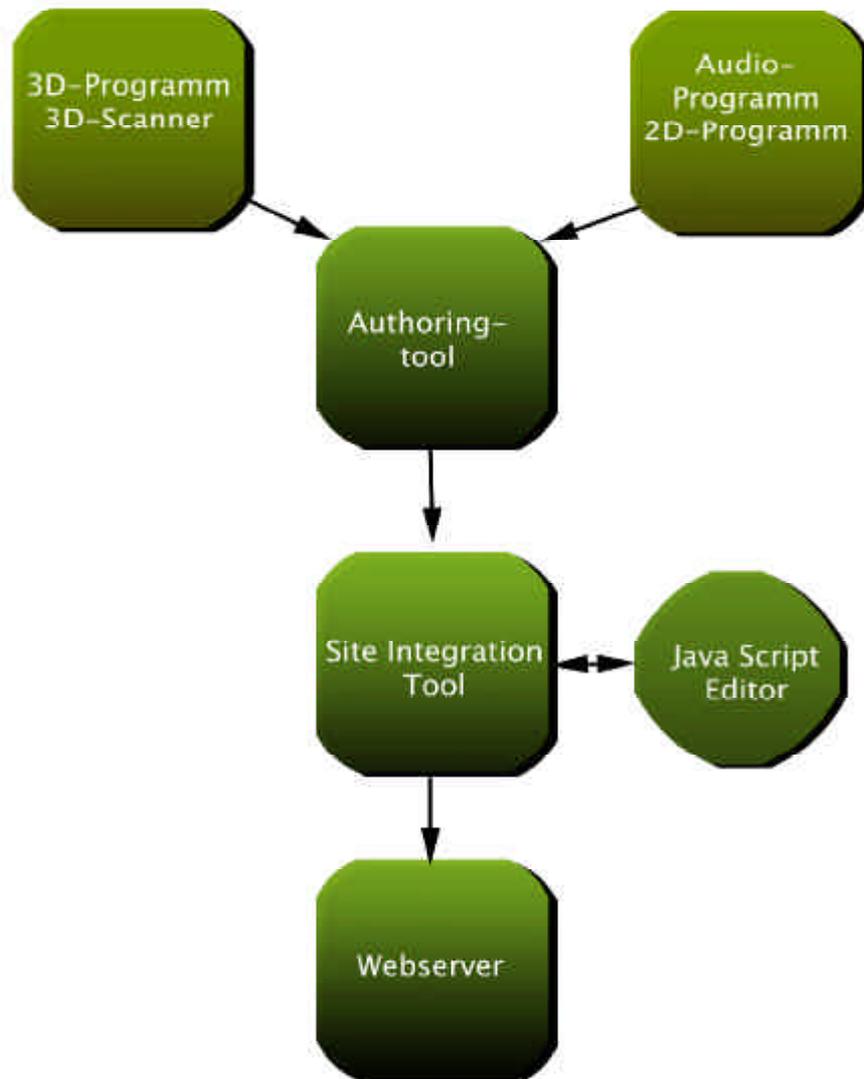
Bei einem häufig als **rigid binding** bezeichneten Verfahren wird jeder Punkt der Geometrie genau einem **Bone** zugeordnet. Dieses eignet sich vorzüglich für das Animieren menschlicher Skelette. Organische Charaktere erscheinen jedoch unnatürlich steif und neigen dazu, an den Gelenken einzuknicken.

Die Lösung bietet **smooth binding**, das in 3ds max als **deformable binding** bezeichnet wird. Bei diesem Verfahren können Punkte von mehreren Gelenken beeinflusst werden. Außerdem kann der Einfluss der einzelnen Gelenke unterschiedlich stark gewichtet werden. Leider sind diese Begriffe nicht klar abgesteckt. So erlaubt z.B. **character Studio**, ein Plugin zur Charakteranimation für 3ds max, auch beim rigid binding Punkte durch mehrere Gelenke beeinflussen zu lassen, allerdings nur wenn man diese manuell zuordnet. Man sollte also bei der Einarbeitung in neue 3D-Tools immer hinterfragen, was genau sich hinter welchem Bindingtyp verbirgt.

Der Kern der Diplomarbeit sind Möglichkeiten zur Darstellung von 3D-Objekten auf Webseiten. Alle von mir betrachteten Lösungen benötigen einen ähnlichen Produktionsablauf: zunächst ist ein Modell in einem **3D-Programm** durch Modelling oder mit Hilfe eines **3D-Scanners** zu erstellen und echtzeitgerecht zu vereinfachen. Wenn es sich dabei um einen Charakter handelt, ist dieser außerdem mit einer animierbaren Skelettstruktur zu verbinden. Danach wird das Modell in ein lösungsspezifisches **Authoringtool** übertragen, wo es mit anderen Medientypen verknüpft und die einzelnen Komponenten mit Interaktionsmöglichkeiten versehen werden.

Als Ergebnis erhält man ein in Webseiten einbettbares Format. Diese **Integration** in das umgebende HTML-Dokument der **Website** stellt den nächsten Schritt dar. Ab und zu ist dieses direkt aus dem Authoringtool möglich, manchmal wird aber auch ein weiteres Programm mit dieser Funktionalität benötigt.

Bei komplexeren Applikationen ist es weiterhin nötig, die Kommunikation zwischen 3D-Inhalt und umgebender Webseite per **JavaScript** zu implementieren. Zu guter Letzt erfolgt die Übertragung auf den **Server**, der den erzeugten Inhalt von nun an für potentielle Interessenten zum Abruf bereit hält.



3. Die Geschichte von Web3D

3.1 Von VRML zu Web3D

Die Geschichte von **VRML (Virtual Reality Modelling Language)** ist eng mit der von *SGI*, dem Vorreiter und jahrelangem Marktführer für 3D-Hochleistungsrechner, verbunden. *SGI* hatte ein ASCII-Format zur Beschreibung interaktiver 3D-Welten namens **OpenInventor** entwickelt. Ein Teil dieses Standards, erweitert um Netzwerkfunktionalitäten, bildete die Basis für die 1995 veröffentlichte **VRML-Spezifikation**¹. Plattformunabhängigkeit, Erweiterbarkeit und die Gewährleistung, auch bei niedrigen Bandbreiten zu funktionieren, waren die ersten Anforderungen.

Der Sockel des Grafikkonzerns begann zu dieser Zeit schon zu bröckeln. So war nicht weiter verwunderlich, dass *SGI* versuchte, sich mit der *Cosmo*-Produktlinie einen neuen Markt zu erschließen: Der **Cosmo Player** war über Jahre hinweg der beste kostenlos abgegebene VRML-Browser. Mit **Cosmo Worlds** war es möglich, VRML-Welten visuell zu entwerfen und mit **Cosmo PlugFX** hatte man eine Lösung zur Erstellung kleiner animierter VRML-Banner im Programm. Das zur *Siggraph 96* in New Orleans vorgestellte **VRML2.0**² erweiterte den Standard vor allem um Funktionen zur Interaktivität. Außerdem wurde auf diesem Event das VRML-Consortium ins Leben gerufen. Im Dezember 1997 ging aus VRML2.0 der ISO-Standard **VRML97**³ hervor.

Doch dann ließ im Juli 1998 eine Hiobsbotschaft die VRML-Gemeinde erzittern: *Cosmo Software* stand zum Verkauf [rp 6]. Da sich kein Interessent finden ließ, (*Sony* war im Gespräch aber ließ das Geschäft dann doch platzen) beschloss *SGI* *zunächst* die Entwicklungsabteilung einzustampfen.

Schließlich erwarb *Platinum Technologies* die *Cosmo* – Produktlinie [rp 7]. Diese Softwarefirma hatte kurze Zeit zuvor bereits *Intervista*, den Hersteller des schlechteren, in Amerika mit Windows95 gebundenen VRML-Browser **World View**, übernommen. *Platinum*, die in der Computergrafikwelt noch ein völlig unbeschriebenes Blatt waren, sahen Einsatzmöglichkeiten für VRML eher im Bereich der Business-Visualisierungen [rp 8].

Das letzte VRML-Symposium fand 1999 in Paderborn statt: Dort verkündete *Platinum Technologies*, die mittlerweile vom Systemsoftwareriesen *Computer Associates*⁴ übernommen worden waren, dass die *Cosmo* Abteilung aufgelöst werden sollte. Nun machte sich zunächst große Ratlosigkeit breit und VRML galt als gestorben. Doch wie ein Phoenix aus der Asche präsentierte das in **Web3D-Konsortium** umbenannte VRML-Konsortium zur *Siggraph 99* den ersten **Web3D-Roundup**⁴, einen nicht ganz ernstzunehmenden Wettbewerb, bei dem das Publikum mittels Ratseln und Tröten aktuelle VRML-Projekte der Branchenführer bewertet. Die Namensänderung in Web3D-Konsortium kam daher, dass man sich von nun an nicht mehr nur auf VRML sondern auf sämtliche webfähige 3D-Technologien konzentrieren wollte. *Blaxxun*⁵ und *ShoutInteractive*⁶ war es gelungen, die Kernelemente des VRML-Browser in einem nur 40K großen **Java-Applet** zu verpacken, so dass nun jeder Java-fähige Browser ohne anfallenden **Plugin**-Download einfache VRML-Inhalte abspielen kann. Vor allem *ShoutInteractive* konnte durch die Präsentation ihrer 3D-Modenschau⁷ für die Modekette *Macy's* neue Maßstäbe setzen. Diese auch heute noch beeindruckende Präsentation ist als der Pionier überzeugender Echtzeit-Charakteranimation auf Webseiten zu betrachten [rp 9].

¹ <http://www.vrml.org/VRML1.0/vrml10c.html>

² <http://www.vrml.org/VRML2.0/FINAL/>

³ <http://www.vrml.org/Specifications/>

⁴ <http://www.computerassociates.com/>

⁵ <http://www.blaxxun.de/>

⁷ <http://www.exciteextreme.com/fashion/>

⁶ <http://www.shoutinteractive.com/>

In letzter Zeit hat sich die VRML – Entwicklung zunehmend auf die Entwicklung des **X3D** – Standards verlagert. Dieser ist abwärtskompatibel zu VRML97 und soll durch die Verwendung der **XML**-Syntax die Integration in Systeme der Zukunft gewährleisten. Das X steht dabei auch für eXtensible: die Standardversion von X3D beinhaltet nur die Core-module. Speziellere Nodes können bei Bedarf nachgeladen werden. Auch Hardwarebeschleunigung via **OpenGL** oder **Direct3D** soll unterstützt werden. Extensions für den schlanken Standard sind z.B. das bereits in *Blaxxuns Contact* – VRML – Browser verwendete Verfahren zur Darstellung von NURBS – Objekten, bei dem der Tessellierungsgrad (Unterteilung des Objektes in von der Grafikkarte darstellbare Dreiecke) entweder frei gewählt oder dynamisch an die vorhandene Hardware angepasst werden kann, **Bumpmaps** und specular Highlights (glänzende Oberflächen). [his 1]

3.2 Die Anfänge der Webisodes

Im Frühjahr 1999 wurde ich zum ersten Mal auf die in San Francisco ansässige Firma *Protozoa*¹ aufmerksam: diese produzierte, damals noch mit VRML, kleine animierte 3D-Kurzgeschichten ähnlich den täglichen Zeitungs-Comicstrips im Bannerformat, in erster Linie um die populäre Comicfigur Dilbert² und seinen Hund Dogbert. Die Idee war clever: populäre Cartoonfiguren wie z.B. die Peanuts, Garfield oder eben Dilbert sind sehr leicht aus primitiven Geometrien zu basteln und stellen bei der Webdistribution den größten Teil des Downloads dar, die Animationsdaten sind dagegen bei solch simplen Figuren so gering, dass man sie nach und nach übertragen (streamen) kann und so komplexere visuell ansprechende Geschichten möglich werden, die so gut wie keine Downloadzeit benötigen. Für diese Strips waren natürlich keine Navigationsmöglichkeiten nötig, es wurde mit einer festen Kamera gearbeitet.

¹ <http://www.protozoa.com>

² <http://www.unitedmedia.com/comics/dilbert/>

³ <http://www.dotcomix.com>

Nach und nach etablierte sich für diese Art von 3D-Filmchen der Begriff **Webisodes**.

Später mauserte sich die mittlerweile in *DotComix*³ umbenannte Firma zum größten Anbieter für Online-Animationen, hatte jedoch in letzter Zeit mit erheblichen finanziellen Schwierigkeiten zu kämpfen. Wer das Programm von *DotComix* in den letzten Jahren verfolgt hat wird sich darüber nicht wundern: Eigenkreationen rund um Figuren des öffentlichen Lebens wie z.B. ‚Gates of Hell‘, das Bill Gates als Höllenbewohner zeigt, oder der ‚Virtual Bill‘, ein unterbelichtetes Machoabbild von Präsident Clinton, wiederholen sich und lassen die klassische 3-Akt-Struktur [may1], die den meisten guten Geschichten zu Grunde liegt, vermissen. Ganz anders bei den Dilbert-Filmchen, die direkt an die Comicstrips von Scott Adams angelehnt zu sein scheinen: Im ersten Akt wird die Situation eingeführt, im zweiten passiert etwas und das führt dann im dritten Akt zu einer Pointe.

3.3 Eigene Webisode-Überlegungen

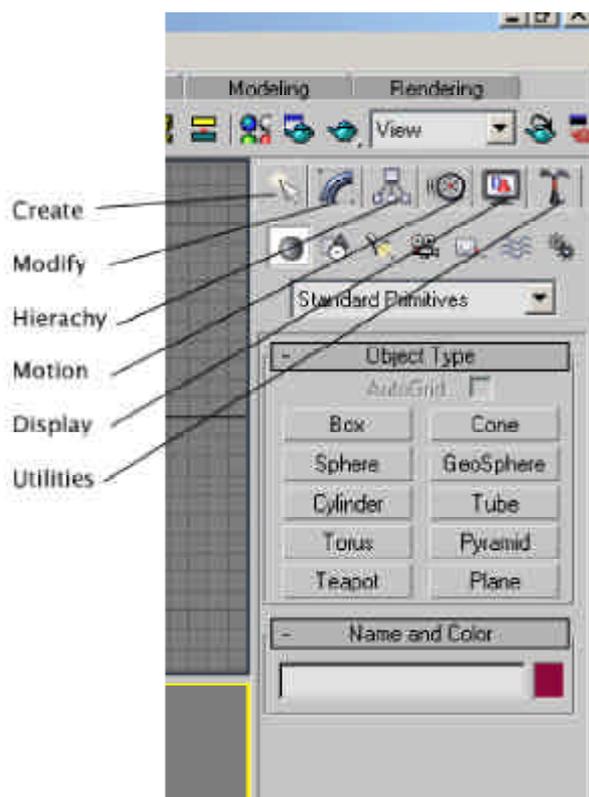
Die Strategie, sich als Portal für solche Animationen etablieren zu wollen, halte ich für unglücklich. Ich denke, dass man sich von dem Bild des klassischen Zeitungsstrips nicht zu weit entfernen sollte. Viele Surfer lesen z.B. morgens erst mal online die neuesten Nachrichten. Während sie noch mit dem Lesen beschäftigt sind, könnte eine Webisode im Hintergrund geladen werden. Wenn der Leser dann am Ende des Screens angekommen ist, muss er, um den Clip zu sehen, nur noch auf den Play-Button drücken, die lästige Wartezeit entfällt. Daher halte ich es für sinnvoller, sich als Content-Lieferant für solche Seiten zu etablieren: durch simples Anpassen der Audiofiles an die jeweilige Landessprache wäre es möglich, an einem Strip gleich mehrfach zu verdienen, ohne auf hohe Besucherzahlen der eigenen Seite angewiesen zu sein.

4. Verwendete 3D-Programme

4.1 3ds max 4 und character studio

Bei meiner Forschungs- und Entwicklungsarbeit war es nötig, *discreets*¹ 3D-Animationssoftware **3ds max** zu verwenden, da dieses System mit den meisten Lösungen für Web3D-Animation zusammenarbeitet.

Die Bedienung von 3ds max erfolgt weitgehend über ein Karteikartensystem, das an den rechten Bildschirmrand anschließt.



Die Karteikarten von 3ds max und deren Funktion

Die **Create**-Kartei enthält Tools zur Erstellung von Objekten und z.B. eines Biped, wenn character studio verwendet wird.

Auf der **Modify**-Palette findet man die Parameter des ausgewählten Objektes und den sogenannten Modifier Stack. Veränderungen an dem Modell sind in diesem hierarchisch übereinander angeordnet, so dass man immer noch Änderungen an vorherigen Zuständen vornehmen kann, die dann auch für später

erfolgte Modifikationen berücksichtigt werden. So ein Modifikator kann zum Beispiel **Edit Mesh** sein, der den Zugriff auf die einzelnen Komponenten einer Mesh ermöglicht. In 3ds max sind das die Koordinatenpunkte (**vertices**), Kanten (**edges**) und Dreiecksflächen (**faces**). Seit Version 4 ist das direkte Arbeiten mit Vierecken möglich. Diese werden bei 3ds max als **Polygone** bezeichnet, obwohl dieser Begriff normalerweise Vielecke mit beliebiger Eckenanzahl beschreibt. Zusätzlich kann eine mesh aus mehreren einzelnen **Elementen** bestehen.

Einstellungen zur Animation der aktuellen Auswahl findet man unter **Motion** und auf der Display-Palette läßt sich die Sichtbarkeit einzelner Objekte an- und ausschalten.

In den **Utilities** findet man Tools wie den **Polygon Counter** und die meisten **Plug-Ins**, z.B. den Pulse Exporter, wenn dieser installiert ist.

Durch einen Rechtsklick auf die oben links angezeigte Bezeichnung der aktuellen Ansicht, kann man zwischen verschiedenen Darstellungsmodi wie **Shaded with highlights**, **Wireframe** oder **Facets** umschalten.

Die Animation erfolgt durch das Festlegen von Schlüsselbildern, sogenannten **Keyframes**. Im **Trackview**, der für alle animierbaren Parameter der Szenenobjekte Spuren enthält, sind die Keyframes über einer Zeitleiste aufgelistet und können darin verschoben, geändert oder durch Hinzufügen eines **Note-Tracks** mit Kommentaren versehen werden.



Das Trackview - Fenster von 3ds max

¹ <http://www.discreet.com>

Wenn eine Extremvariante von einem Objekt, meistens eines Kopfes, erstellt wird, besteht die Möglichkeit, das ursprüngliche Objekt sanft in diese Variante übergehen zu lassen. Dieses nennt man auch **morphen**. Dazu ist dem Modifier Stack des Ausgangsobjektes ein **Morpher** hinzuzufügen und die Extremausprägung als **Morphtarget** zu definieren. Die Gewichtung des Überblendens kann nun von 0 bis 100 % in den Modifikatoroptionen gewählt und animiert werden.

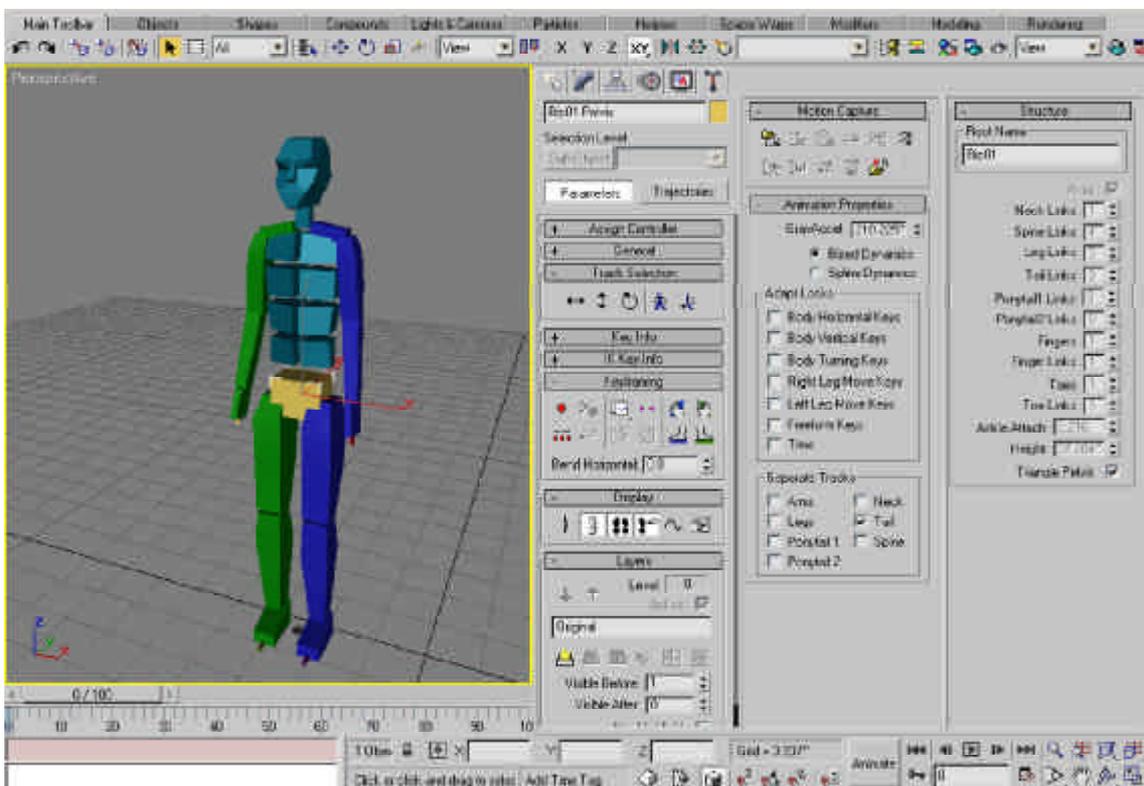
character studio, ein von *discreet* selber vertriebenes **Plug-In** für 3ds max, liegt zur Zeit in der Version 3 vor. Es erleichtert die Erstellung animierter Charaktere ungemein. Ein **Biped** ist eine zweibeinige Skelettstruktur, die zur Animation eines Charakters verwendet werden kann. Durch die Erzeugung von **Footsteps** lassen sich Bewegungen sehr schnell erstellen.

Darüber hinaus beinhaltet **character studio** mit dem **Motion Flow** eine Möglichkeit zur

nichtlinearen Charakteranimation und kann seine Bewegungen aus MotionCapture – Formaten wie **bvh** (biovision) oder **csm** erhalten. Ein weiteres Tool, das **character studio** mitbringt, ist **physics**. Mit diesem Modifikator kann man die zu animierende Mesh an das Biped-Skelett binden. Dieses kann entweder mit **smooth** oder **deformable binding** geschehen. Sämtliche Bewegungen der Biped-Gelenke werden nun auf die zugeordnete Mesh übertragen.

Für die Beine des Bipedes werden standardmäßig keine Keyframes im Trackview dargestellt. Dieses muß erst explizit in den Animation Properties der zugehörigen Motion-Palette angewählt werden. Hier kann auch unter Struktur festgelegt werden, aus wie vielen Gelenken das Biped bestehen soll.

Da ich während des Implementierungsprozesses bei *ID-TV* sehr viel mit **character studio** zu tun hatte werde ich im Bericht darüber noch näher auf Eigenheiten dieses Tools eingehen.



Ein Standardbiped mit seinen komplexen Animationsoptionen auf der Motion-Palette

4.2 Maya



1998 brachte die SGI-Tochter *alias/wavefront*¹ die Animationssoftware **Maya** auf den Markt. Diese vereinte von Anfang an ein komplexes **Partikelsystem**, **polygonale** und **Nurbs-Modelling** Tools, nahezu unbegrenzte Animationsmöglichkeiten und den Renderer in einem Softwarepaket und erfreute sich auf Grund seiner schnellen und gut durchdachten Bedienbarkeit, der hohen Anpaßbarkeit an eigene Vorlieben und der Skriptsprache MEL bald einer außerordentlichen Beliebtheit. Da bisher außer *WildTangent* kein Hersteller Exportplugins zur Übernahme der in Maya erstellten Charakter-Animationen bereitstellen konnte, beschränke ich mich an dieser Stelle auf die im Zusammenspiel mit diesem Exporter wesentlichen Funktionen.

Standardmäßig sind komplexe Tools zum Skelettbau integriert. Um ein Skelett mit der Geometrie zu verbinden, können Maya-Nutzer zwischen **rigid** und **smooth binding** wählen. Beim rigid binding ist es möglich, mittels einer **Lattice Box** Muskelverformungen zu simulieren. Beim smooth binding kann die Gewichtung (**weight**) der einzelnen Bones auch direkt als Grauwert auf die Geometrie gemalt werden.

Durch **Character-Nodes**, die mehrere Attribute des Charakters zusammenfassen, wird die Erstellung separater Animationen für die einzelnen Charakter-Elemente erleichtert. Mit der Version 3.0 wurde ein nichtlinearer Animationseditor namens **Trax** eingeführt. In diesem ist es möglich, verschiedene Animationsclips beliebig aneinander zu reihen bzw. zu überblenden.



Die separaten Animationsclips für den Körper und die Beine können in Mayas nichtlinearem Animationseditor Trax unabhängig voneinander verschoben, skaliert und überblendet werden.

¹ <http://www.aliaswavefront.com>

5. Der Charakter E-sha



5.1 Wer ist E-sha?

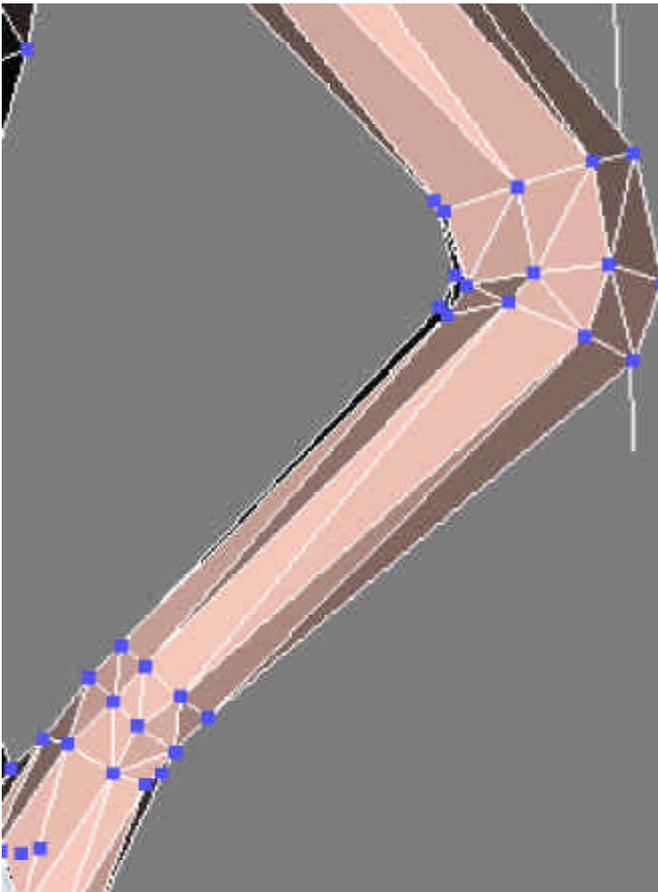
Ihren ersten großen Auftritt sollte die in Maya modellierte **E-sha** im Jahr 2000 als Interpretin des *BMG* – Songs ‚**Dreamgirl**‘ haben. Leider begann die Produktion bevor die Verträge unterzeichnet worden waren und bis das komplett 3D-gereimte Video fertig war hatten die Mitarbeiter der *Bertelsmann Music Group* bemerkt, dass sich der zugehörige Track nicht gut verkaufen lassen würde.

Für *ID TV* ergab sich natürlich eine denkbar ungünstige Situation: das Video war fertig, doch ohne die Rechte an der zu Grunde liegenden Musik, die nun nie erscheinen sollte, konnte es nicht veröffentlicht werden.

Mir kam dieser traurige Umstand zugute, denn so hatte ich die Möglichkeit, den bestehenden Charakter für meine Webentwicklung zu recyceln.

5.2 Nurbs to Polygon

Die ursprüngliche Version der auf 2500 Polygone reduzierten E-sha wurde von Christophe Desse modelliert, wobei er das ursprüngliche NURBS-Modell als Referenz verwendete. In Maya hätte er zwar auch das Nurbs-Modell automatisch in ein polygonales Modell umwandeln können, beim Low-Poly Modelling ist es jedoch äußerst wichtig, sparsam mit Polygonen umzugehen und mehrere Polygone wirklich nur dort einzusetzen, wo sie für Deformationen nötig sind, also z.B. am Ellenbogen oder den Knien [totb 1]. Ungünstige Dreiecksflächen entdeckt man am besten im *flat*-geschadeten Facets-Modus und kann diese durch ein Drehen der Kanten (**Turn Edges**) korrigieren [totb 2].



An Gelenken ist eine höhere Polygonzahl nötig

5.3 Generierung der Texturkoordinaten

Die Texturkoordinaten (**UV-Map**) wurden in 3ds max generiert, indem sämtliche Flächen zur Kamera ausgerichtet wurden und danach eine Projektion stattfand. Durch die Verwendung einer Kopie des ursprünglichen Modells als Morphtarget konnte man aus den planaren Flächen wieder den Originalzustand erzeugen. Diese Vorgehensweise hat zwar den Vorteil, dass es zu keinen Texturverzerrungen kommt, die durch Projektionen auf nicht parallele Flächen entstehen. Allerdings ist es sehr lästig, alle Flächen zur Kamera auszurichten. Im Nachhinein habe ich festgestellt, dass es einen einfacheren Weg gibt, um die gleichen Ergebnisse zu erzielen, von dem ich bei E-shas Periskopschwert Gebrauch machte: Maya kann automatisch eine nichtverzerrte UV-Map generieren. Wenn man das in Maya texturierte Modell als **OBJ** exportiert und dann mit dem kostenlosen **OBJ-Import Plug-In** von Harald Blab¹ in 3ds max einlädt, können diese Texturkoordinaten übernommen werden.

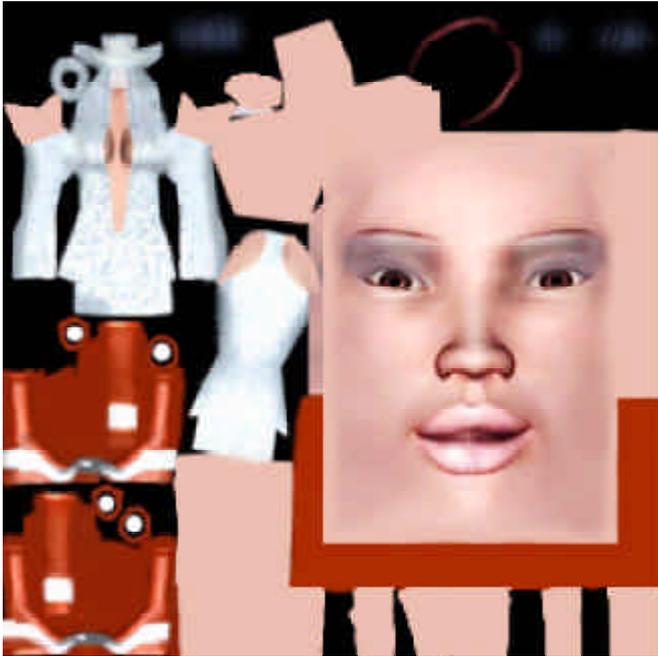
5.4 Die Textur

Ein mit dem ebenfalls kostenlosen **Plug-In Textporter**² exportierter Screenshot der ausgerichteten Flächen konnte nun in Adobes **Photoshop** mit der entsprechenden Textur bemalt werden. Bei älteren Gameengines, wie z.B. der von **Doom**³, verbesserte man häufig die Darstellung, indem Schattierungseffekte direkt auf die Textur gezeichnet wurden. Da **Pulse3D**, die Technologie, die ich in meiner Beispielimplementierung verwendet habe, über eine eigene Shading-Engine verfügt (Lightmaps), war dieses ungeeignet. Einfarbige Bereiche produzierten bei Hose und Schultern die besseren Ergebnisse, obwohl Sie im nicht schattierten Modus flächig und langweilig aussehen.

¹ <http://www.habware.at>

² <http://www.cuneytozdas.com/software/max/textporter/>

³ <http://www.idsoftware.com/archives/doomarc.html>



Diese Textur wurde mittels UV-Map verwendet

Die Texturierungskünste von **Chris Tuhnig**, der für mich die Ausgangstextur anfertigte, konnten jedoch vor allem den Haaren, dem Gesicht und dem Ausschnitt die nötigen Details verleihen. Die genaue Zuordnung der UV-Map und nötige Texturkorrekturen habe ich selber vorgenommen. Um die Texturgröße insgesamt auf 256x256 Pixel reduzieren zu können, ohne dem Gesicht die nötige Qualität zu rauben, habe ich die Texturkoordinaten der einfarbigen Bereiche auf kleinere Bereiche zusammengezogen (runterskaliert) und den Gesichtsbereich aus einer höher aufgelösten Version übernommen. Durch die Reduzierung der Texturgröße auf ein Viertel konnte ich E-sha für den zunächst geplanten Webeinsatz optimieren.

6. Grobe Unterteilung der verschiedenen Technologien

Für den Betrachter macht es visuell kaum einen Unterschied, ob er nun einen echten 3D-Charakter vor sich hat, ein gerendertes 2D-Abbild von diesem oder bei Charakteren im Cartoon-Look eine vektorielle 2D-Figur. Diese Tatsache machten sich Webentwickler in den letzten Jahren zunehmend zu Nutze, um virtuelle Charaktere auf Webseiten agieren zu lassen. Natürlich ist klar, dass bei dieser Variante jegliche Interaktionsmöglichkeiten verloren gehen. Sie eignet sich nur zum Einsatz für vordefinierte Animationen. Technologien aus diesem Bereich möchte ich unter dem Begriff **Fake3D** zusammenfassen.

Wenn der Einsatz von 3D auf Webseiten in erster Linie dem Zweck der **Objektivsualisierung** dient, sind Importmöglichkeiten für Charaktere unnötig. Dafür sollte die Darstellungsqualität der Realität schon sehr nahe kommen. Geeignete Lösungen betrachte ich ebenfalls gesondert.

Zu guter Letzt kommen wir dann zu Lösungen, mit denen Online-3D-Filmchen, komplexere Anwendungen oder gar Spiele zu produzieren sind und die direkt im Browser ablaufen können. Bedingung für die Einordnung in diese Kategorie ist die Möglichkeit, **Charakteranimationen** direkt aus 3D-Paketen zu übernehmen.

7. Fake3D

7.1 Flash

7.1.1 Vorstellung von Flash

Mit dem Programm **Flash** von *Macromedia*¹, das mittlerweile in der Version 5.0 vorliegt, ist es durch die Verwendung vektorieller Bilddaten möglich, 2D-Animationen von äußerst geringer Dateigröße zu erstellen. Kein Wunder, dass sich dieser Standard in Windeseile auf so gut wie alle Plattformen verbreiten konnte, nicht zuletzt da das äußerst kleine Plugin (311K/Windowsversion) mittlerweile gleich mit den großen Browsern mitgeliefert wird. Macromedia behauptet, dass heutzutage 90% der Browser über ein installiertes Flash-Plugin verfügen.

7.1.2 2D in Flash?

Mit Flash wäre es auch möglich, Interaktivität vorzugaukeln, indem vorgeladene 2D-Bilddaten einfach ausgetauscht werden. Dieses Prinzip erinnert ein wenig an ein Daumenkino, hat aber den Vorteil, dass es problemlos sowohl vorwärts als auch rückwärts ausgeführt werden kann. Wenn ich z.B. eine Rotation von einem Objekt gerendert habe, könnte ich mit diesen Einzelbildern leicht eine Drehung simulieren. Die einzige Vorgehensweise, um die Bilddaten zu reduzieren, wäre eine Einzelbildkompression durch *JPEG*, *GIF* oder ähnliche Formate.

7.1.3 Von 3D nach 2D

Im Herbst 1999 kursierte das Intro zur Webpage der französischen Firma *Melondezign*² und versetzte meinen Bekanntenkreis in Erstaunen: An und für sich waren die, mit für Frankreich zur damaligen Zeit so typischen Houseklängen unterlegten, Animationen stilisierter Frauenkörper zwar nichts bahnbrechend Neues,

aber es erschien uns unmöglich, dass diese perspektivisch schwierigen Animationen von Zeichners Hand entstanden sein sollten.



Ausschnitte aus dem 99er - Melondezign -Intro

Dass auch das Team von *Melondezign* nur mit Wasser kocht, wurde uns klar, als wir erfuhren, dass für 3ds max eine Reihe von *Plugins* erhältlich sind, mit deren Hilfe man ohne größere Probleme vektorielle Flash-Daten aus 3D-Objekten generieren lassen kann. Zu diesen zählen **Vecta3D**³ von *Ideaworks3D*, **Swift3D** von *electric rain*⁴ und **Illustrate** von David Gould⁵.

Während *Vecta3D* und *Swift3D* eigenständige Systeme darstellen und nur als weiteres Feature den Export aus 3ds unterstützen, ist *Illustrate* 5.1 ein reines *Plug-In*, das aber neben dem Export in Vektorformate auch als Cartoon-Shader fungiert und bald auch für *Maya* erhältlich sein soll.

7.1.4 Illustrate in der Praxis

Da die vektoriiellen Daten aus den Geometrieinformationen generiert werden, können Texturen zur Farbgebung beim Export leider nicht verwendet werden. Nur die Materialfarbe findet sich, eventuell mit Konturen versehen, in der exportierten Flash-Datei wieder. Um mehrfarbige Objekte zu erstellen, ist es nötig, den einzelnen Dreiecksflächen der Mesh Farbwerte zuzuweisen.

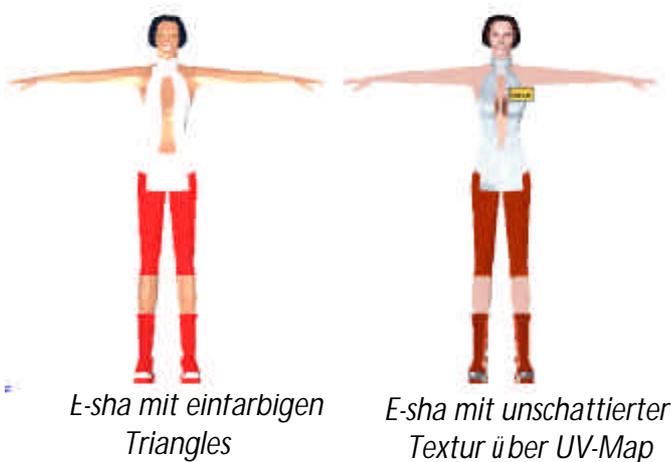
¹<http://www.macromedia.com>

²<http://www.melondezign.com>

³<http://www.vecta3d.com>

⁴<http://www.erain.com>

⁵<http://www.davidgould.com>



bessere Ergebnisse erzielen können. Allerdings ist die bloße Outline-Animation mit 62KB immer noch konkurrenzlos klein.

7.1.5 Fazit

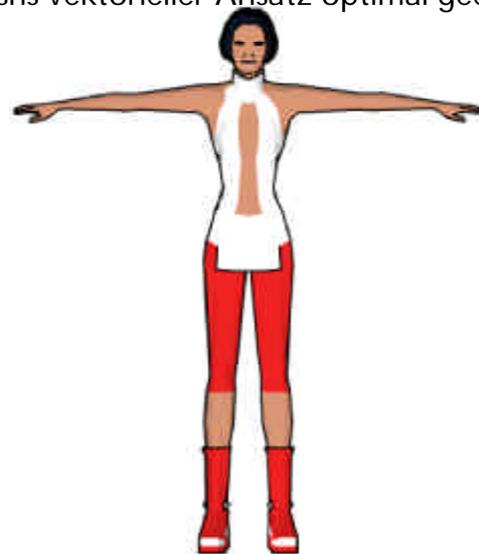
Aus diesen Ergebnissen wird deutlich, dass der Größenvorteil vektorieller Daten mit mehreren Farbflächen erstaunlicherweise nur in der verlustfreien Skalierbarkeit zu liegen scheint. Sinnvoll ist ihre Verwendung hingegen, wenn Kontur- oder Silhouettenanimationen sowie Bilder mit äußerst wenig verschiedenen Farbflächen zum Einsatz kommen sollen: Dafür ist Flashs vektorieller Ansatz optimal geeignet.

Bei doppelten Kanten zeichnet Illustrate eine Linie. Um diesen Effekt dort, wo es nicht gewünscht ist, zu vermeiden ist es ganz besonders wichtig, dafür zu sorgen, dass nicht mehrere Punkte übereinander liegen. Dieses geschieht am einfachsten mit der Funktion **„Weld Selected“** unter Angabe eines geeigneten Toleranzwertes, so dass auch wirklich nur übereinanderliegende Punkte erfasst werden.

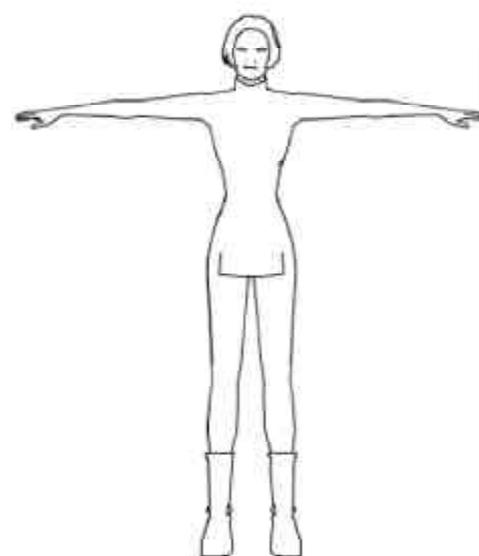
Im Test traten häufig Renderingfehler auf. Zum Beispiel sind immer wieder Flächen falsch coloriert. Dieses Problem lässt sich aber zum Glück in Flash selber, durch das Füllen der betreffenden Fläche mit einem anderen Farbwert, korrigieren. In den seltensten Fällen kommt man ohne Finetuning in Flash aus.

Die vektorielle Version meiner E-sha im mehrfarbigen Cartoon-Look war als Einzelbild 17KB groß. Aus den gleichen Bilddaten ließ sich aber auch durch Reduzierung der Farbpalette auf 32 ein nur 9KB großes GIF erzeugen. Wenn man auf farbige Flächen verzichtet und nur die Konturen dargestellt werden, ist das vektorielle Ergebnis mit 2KB sehr klein.

Ähnliche Ergebnisse lieferten auch Versuche mit einer Animation von 50 Frames: Die Flash-Variante im swf-Format war 681KB groß, mit 50 GIFs a 9KB würde man mit 450KB auch hier das



Vektorielle E-sha im Cartoon-Style (17kb)



Vektorielle E-sha nur als Outlines (2kb)

7.2 Virtual Friends



Die Virtual Friends der Basic-Version

Die Firma *vista new media*¹ aus Köln setzt mit ihrer Software **Virtual Friends**² auf eine Kombination von vorgeordneten Sequenzen mit einer baumähnlichen Dialogstruktur, bei der die Entscheidung für den einen oder anderen Dialogverlauf auf Grund von festzulegenden Signalwörtern erfolgt. Diese **MicroBrain** genannte Intelligenz kann selber mit Hilfe des **Java2** voraussetzenden Virtual Editors definiert werden. Die fertigen Charakter laufen dann auf der Webseite in einem plattformunabhängigen **Java1.1 - Applet**.

Einzelne Animationssequenzen werden schon bevor sie zum Einsatz kommen vorausgeladen. Außer Texte von sich geben und Animationen abspielen, kann das System auch auf Webseiten verweisen. Bei Erwerb der Professional-Version kann auch eine Text-to-Speech-Engine zur Stimmengenerierung eingesetzt werden.

Die Software Virtual Friends gibt es mittlerweile in 6 Ausbaustufen. In den Einsteigervarianten können nur mitgelieferte Charaktere verwendet werden. Dagegen lassen sich in der teureren Developersversion auch eigene Charaktere und Animationssequenzen verwenden.

Außerdem bietet *vista* eine Livevariante an. Mit dieser kann die Charaktersteuerung durch einen Operator über die Tastatur eines zweiten PCs erfolgen. Sie bietet sich für den Einsatz virtueller Charaktere auf Messen oder im Fernsehen an.

Virtual Friends ist, so weit mir bekannt ist, die einzige Komplettlösung für intelligente Avatare auf Webseiten, die keine Programmierkenntnisse erfordert. Wer das System einmal selber ausprobieren möchte, kann dazu eine kostenlose Lightversion mit dem Charakter Henry von der VirtualFriends-Homepage² herunterladen.

¹<http://www.vista.de>

²<http://www.virtualfriends.de>

8. Systeme zur Objektvisualisierung

8.1 Viewpoint´s Experience Technology

8.1.1 Die Geschichte von VET

Anfang 2000 überraschte die Firma *MetaCreations* mit der Ankündigung, ihre gesamte grafische Produktlinie, zu der Klassiker wie **Bryce**, **Poser** oder **Kai´s PowerTools** gehörten, bis auf das relativ junge Produkt **Canoma**¹ abstoßen zu wollen. Canoma ist ein Programm zur Generierung von texturierten Objekten aus mehreren Photoansichten ein und desselben Objektes. Die Geometrie ist hierfür so in die einzelnen Photos einzupassen, dass sie den Maßen des wiederzugebenden Objektes entspricht. Canoma übernimmt dann das perspektivische Entzerren der für die sichtbaren Flächen als Texturen zu verwendenden Bereiche und kann diese dann auf Wunsch auch auf die verdeckten Seiten kopieren.

Die Firma hatte beschlossen, sich von nun an ausschließlich auf die Vermarktung ihres gemeinsam mit *Intel* entwickelten Web3D-Formates **MetaStream** zu konzentrieren. Den Haupteinsatzbereich für dieses sah sie im e-commerce, und da war ein Tool zur einfachen Generierung photorealistischer Objekte natürlich hilfreich. Dennoch wurde dieses Anfang 2000 überraschend an *Adobe*² abgetreten, die dafür versprochen, MetaStream-Exporter in ihre Grafikprodukte zu integrieren [mcw1]. Seit dem ist keine neue Version von Canoma mehr auf den Markt gekommen.

Computer Associates, die durch den Erwerb von *Platinum Technologies* zum Nachlassverwalter der Cosmo-Produkte geworden waren, beteiligte sich zu 20 % an der aus *MetaCreations* neu hervorgegangenen *MetaStream.com Corporation*. Dadurch enttäuschten sie die VRML-Entwicklergemeinschaft, die auf eine Freigabe des CosmoWorlds-Sourcecodes gehofft hatte: Rob Polevoi vermutete damals in seinem Webreference-Beitrag über MetaStream, dass es nun wohl nicht mehr im Interesse von *Computer Associates* lag, die VRML-Konkurrenz durch eine Unterstüt-

zung des offenen Web3D-Standards zu stärken [rp 2].

MetaStream sah sich aber nicht nur als Anbieter der Technologie, sondern auch der gewünschten 3D-Objekte. So braucht es nicht zu wundern, dass es im Oktober 2000 zum Zusammenschluß mit *Viewpoint*,² dem führenden Anbieter professioneller 3D-Modelle, zur **Viewpoint Corporation** kam. In Folge dieses Zusammenschlusses ging aus *MetaStream 2.0* **Viewpoint´s Experience Technology (VET)** hervor.

8.1.2 Die Technik hinter VET

Der Zusammenschluß hatte weitreichende Veränderungen zur Folge: Man wollte weg vom ‚rechteckigen 3D-Browser innerhalb des Browsers‘ und statt dessen zu einem vollwertigen, integrierbaren Medientyp werden. Dieses gelang durch ein Verfahren namens **Draw-Anywhere**. Das neue Plugin verfügt nun über einen transparenten Hintergrund, kann also ohne Probleme in Webseiten eingebettet werden. Da ein Format zur Produktvisualisierung auch bei niedriger Datenmenge hochaufgelöste Objekte liefern sollte, kommt die **Vertex History** zum Einsatz: Je nach Anforderungen kann die Mesh entweder durch das Hinzufügen weiterer Vertices höher aufgelöst oder durch die Reduzierung der Vertex-Anzahl für niedrige Bandbreiten optimiert werden. VET-Inhalte können direkt über **JavaScript** angesprochen werden. Dieses Prinzip finden wir auch bei konkurrierenden Technologien sehr häufig. Eine weitere Neuerung ist die Trennung von Beschreibung und Inhalt: Die Szenenbeschreibung ist gemäß der **XML**-Syntax in einer **MTX**-Datei gespeichert. Die Geometriedaten werden aus einer **MTS**-Datei referenziert. Da diese **XML**-Daten häufig sehr groß geraten, kann man sie noch komprimieren. Auf die entstandenen **MTZ**-Daten stößt man in der Praxis am häufigsten.

¹ <http://www.canoma.com>

² http://www.viewpoint.com/company/g1b_detail.jsp?id=13

Die Verwendung von *XML* hat den Vorteil, dass Szenenbeschreibungen auf Grund von Benutzervorgaben dynamisch generiert werden können. Außerdem können vorhandene *XML*-Tools zur Entwicklung verwendet werden. Für spätere Erweiterungen des Standards sind alte Versionen durch automatisiertes Textparsing um neue Einträge ergänzbar. Geplant ist, dass die einzelnen **Viewpoint Media Assets** (Szenekomponenten) nicht mehr nur aus Images und 3D-Daten, sondern aus allen erdenklichen Medientypen bestehen können.

Die Minimalversion des daher auch in ‚**Viewpoint Media Player**‘ umbenannten Browser-*Plug-Ins* ist heute 340 KB groß. Wenn allerdings der Inhalt weitere Features verlangt, müssen diese automatisch nachgerüstet werden. Die volle Installation umfasst 1,37 MB.

Für nichtkommerzielle Zwecke ist es möglich, eine kostenlose Lizenz (**Broadcast Key**) zu erhalten. Dafür fallen bei kommerziellen Projekten Gebühren zwischen \$2.500 und durchschnittlich \$100.000 an. Nach oben setzt man offenbar bewusst keine Grenze. Wer auf den Broadcast Key verzichten möchte, muss sich leider mit einer von einem Viewpoint-Schriftzug überlagerten 3D-Ansicht zufrieden geben.

Um selber VET – Inhalte zu produzieren, genügen neben einem Programm zur Modellerstellung zwei kostenlose Tools: Der **Viewpoint Scene Builder** und das Tool **Mtx2HTML**.

Die 3D-Modelle können entweder aus Programmen wie **Truespace**¹, **Solidworks**² oder Poser direkt exportiert oder mit **Polytrans**³

aus so ziemlich allen erdenklichen Formaten umgewandelt werden. Direkt importieren kann man neben dem *OBJ*-Format auch *ASE*-Files (Ascii Scene Export) aus 3ds max. Über *ASE* können auch simple Animationen (keine Vertexanimationen) übernommen werden. NURBS-Modelle wandelt Viewpoint automatisch in polygonale Daten um.

Weitere Features, welche die beeindruckende Darstellungsqualität der VET-Modelle begründen, sind der projizierte Schatten, die Vortäuschung poröser Oberflächen (*Bump Mapping*), die Kantenglättung (**Anti-Aliasing**) und mehrere Texturebenen. Bilddaten können mit Hilfe von **TrixelINT**, einer auf dem *Wavelet*-Verfahren basierenden Kompression, geschrumpft werden. Polygone können, müssen aber nicht beidseitig sein. Zur Darstellung werden Lightmaps verwendet.

Animationen können entweder über Buttons mit hinterlegten *JavaScript*-Aufrufen oder über sogenannte **Widgets**, simple Objekte, die später vom Betrachter nur als mit einer Funktionalität verbundene Regionen wahrzunehmen sind, ausgelöst (getriggert) werden.

8.1.3 Zukunftsausblicke

Für die neueste Version der von *Curious Labs*⁴ weiterentwickelten Poser-Reihe sind Exportmöglichkeiten für animierte Charaktere ins Viewpoint-Format angekündigt. Beim Betrachten der Beispiele des Herstellers fällt auf, dass man beim ersten Mal zwar etwas länger auf die Inhalte warten muss, die Erweiterung des Funktionsumfangs des Players jedoch völlig vor dem Benutzer verborgen im Hintergrund geschieht.⁵

¹ <http://www.caligari.com/>

² <http://www.solidworks.com/>

³ <http://www.okino.com/>

⁴ <http://www.curiouslabs.com>

⁵ <http://www.curiouslabs.com/products/proPack/viewpoint/html/viewpointExamples.html>

8.2 Cult 3D (Cycore)

8.2.1 Technologievorstellung

Die schwedische Firma *Cycore*¹, die sich zuvor durch *Plug-Ins* für *Adobes* AfterEffects einen Namen machen konnte, kam bereits Anfang 1999 mit ihrem **Cult3D**-Format auf den Markt. Wie bei Viewpoint sieht man auch bei Cycore den Einsatzbereich im e-commerce, allerdings sich selber nur als Technologie- und nicht als Contentprovider. Die Lizenzpreise von Cult3D sind seit Juni abhängig von der Besucherzahl der Seite. Wenn bis zu 10.000 Besucher pro Jahr gezählt werden, kommt man mit \$1.500 noch recht günstig davon. Diese Gebühr kann sich aber bis zu \$15.500 steigern. Somit entspricht das Preismodell jetzt in etwa dem von Viewpoint. Um eine Produktlinie als 3D-Objekte offline zu vertreiben fällt eine jährliche Gebühr von \$500 an. Dieser Betrag muß auch für Hochschullizenzen und von Entwicklern bezahlt werden. Letztere dürfen dann aber mit Eigenkreationen auf ihrer Homepage werben und erhalten darüber hinaus beim Weiterverkauf von kommerziellen Lizenzen an ihre Kunden Tantiemen von Cycore. Der Hobbymodeller hat auch hier die Möglichkeit, eine kostenlose Lizenz zu erhalten. Das Viewer genannte Plugin ist nicht nur für so gut wie alle Plattformen erhältlich (MacOS, Linux, Windows) sondern durch Verwendung von ActiveX auch in *Macromedia* Director, *Adobe* Acrobat- oder *Microsoft* Officedokumente einbindbar. Leider fällt für jedes Zielsystem (Internet-Explorer, Netscape Communicator, MS-Office ...) ein separater Download an, der mit einer Größe von jeweils mindestens 1,2 MB im Vergleich zu denen der Konkurrenz relativ groß ist. Für die nahe Zukunft sind Cult3D-Viewer für PDAs und sogar Handys in Planung. Der Intel Pentium 4 Prozessor soll die Darstellung zusätzlich beschleunigen [dp 01a].

Intuitive Exportplugins sind sowohl für 3ds max als auch für Maya und **Strata3D**² erhältlich.

¹ <http://www.cycore.com/>

² <http://www.strata3d.com/>

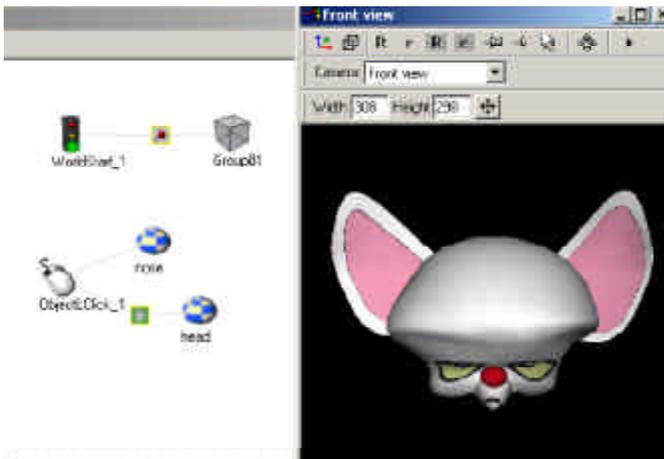
8.2.2 Cult3D in der Praxis

Da Cult3D nur einseitige Polygone unterstützt, ist es vor dem Export aus Maya nötig ‚two sided lighting‘ auszuschalten und ‚back face culling‘ zu aktivieren. Außerdem werden keine Animationsdaten exportiert. Auf die automatische Nurbs-Tessellierung beim Export sollte man lieber verzichten und Mayas Funktion zur Umwandlung in Polygone verwenden. Lichter werden nicht mit exportiert, Cult3D-Objekte verwenden immer die gleiche Standardbeleuchtung.

Der exportierte Cult3D-Szenengraph ist unterteilt in Header, Background, Materials, Nodes und Textures. Das mit den aktuellen Einstellungen zu erwartende Ergebnis ist bereits in einem Viewer zu begutachten. Der Header enthält Informationen über Polygonanzahl sowie den Urheber der Datei. Dem Background kann entweder ein Farbwert oder ein Hintergrundbild zugewiesen werden und unter den Nodes verstecken sich die exportierten Bestandteile des Modells in ihrer ursprünglichen Hierarchie und Bezeichnung. Wer nun direkt die einzelnen 3D-Komponenten anwählt, erhält die Möglichkeit, die Polygonzahl zu reduzieren. Die dazu verwendeten Algorithmen scheinen sehr durchdacht zu sein, denn nach der Reduzierung der Polygonzahl um 50% nahm ich auf Anhieb keinen sichtbaren Unterschied wahr. Alle Änderungen müssen explizit durch einen Klick auf ‚Anwenden‘ bestätigt werden, was ziemlich lästig ist. Dafür können übereinanderliegende Vertices beim Export automatisch erkannt und zusammengeschweißt werden. 3ds max - Anwender können außerdem auswählen, von welchen Komponenten auch Vertexanimationen mitexportiert werden sollen. Die Texturkompression läßt sich ebenfalls separat regeln. Zur Auswahl stehen die verlustfreie 24 Bit Variante, die Verwendung einer 8 Bit Palette oder wieder einmal *Wavelet*. Auch die Größe der Texturen (2^x) kann man probeweise reduzieren bis die Qualität merklich schlechter wird und sich dann für den besten Kompromiss entscheiden.

Um auch komplexere Interaktivität zu erlauben, ohne Programmierkenntnisse vorauszusetzen, verwendet der kostenlose CultDesigner eine visuelle Form des ‚Zusammensteckens‘ von Interaktionsmöglichkeiten [rp 4] [rp 5]. Dieses gelingt nach kurzer Einarbeitungszeit dank der umfangreichen Dokumentation recht gut. Im unten abgebildeten Beispiel wird der Geometriegruppe Group1 unmittelbar nach dem Laden (WorldStart_1 repräsentiert durch eine grüne Ampel) eine Arcball – Funktionalität zugewiesen die bewirkt, dass das Objekt gedreht oder verschoben werden kann.

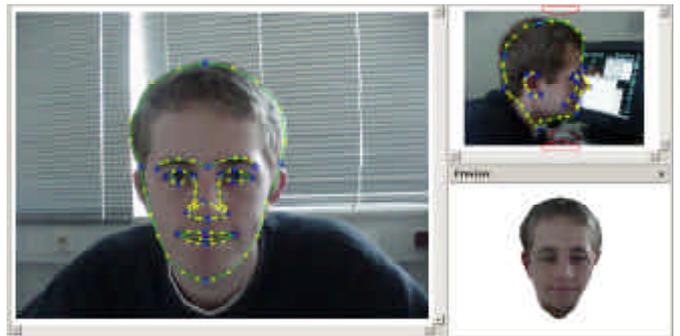
Zusätzlich besteht die Möglichkeit durch einen Klick (ObjectLClick_1) auf die Nase (nose) mit der linken Mousetaste eine in 3ds max vorbereitete Vertexanimation abzuspielen. Brain (so heißt der modellierte Charakter) macht einen tiefen Atemzug.



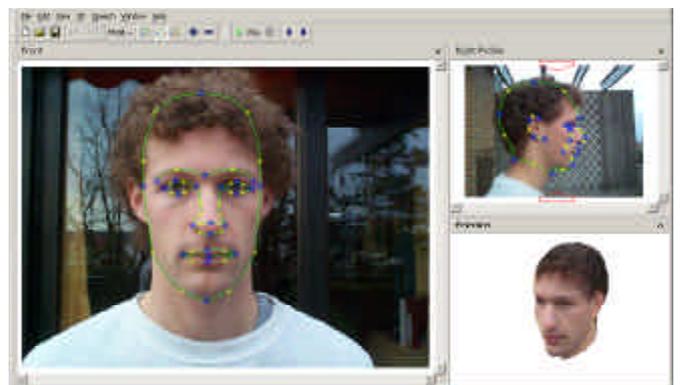
Cult3D verfügt auch über ein eigenes **Partikelsystem**, mit dem z.B. echtzeitfähige Flammen sehr schnell erzeug- und animierbar sind. Wer schon über **Java**-Kenntnisse verfügt, freut sich über das mitgelieferte **API**, das es ermöglicht, Cult3D nach Belieben zu erweitern. Auch eigene Klassen können dann ähnlich den **Java Beans** von weniger versierten Entwicklern visuell im Designer eingesetzt werden. [3dg 1]

8.2.3 3DMeNow

Mit der Software 3DMeNow von *biovirtual*¹ ist es möglich, aus zwei Kopfphotos, einer Frontal- und einer Seitenansicht, in kürzester Zeit durch Anpassen der Kopfkonturen eine 3D-Repräsentation der photographierten Person zu erzeugen. Wenn man diese nun noch mit einem Soundfile versieht, hat man dank automatischer Lippensynchronisierung und dem Export ins Cult3D-Format einen überzeugenden Moderator für seine Homepage. In der kostenlosen Lightversion ist leider die Dauer der Audiofiles auf 7 Sekunden beschränkt. Für \$49 erhält man die Vollversion, aus der sich die Köpfe auch in zahlreiche professionelle 3D-Formate exportieren lassen. Die so generierten Köpfe sind aber keinesfalls mit 3D modellierten Charakteren zu vergleichen, da sie ihre Details eigentlich nur über die Textur erhalten.



Natürlich eignet sich 3DMeNow eher für Personen mit kurzen, glatten Haaren ...



als mit langen oder lockigen. Die Geometrie für die Zähne ist immer die gleiche.

¹ <http://www.biovirtual.com>

8.3 Open-SPX (Interzart)

Die *Interzart AG*¹ aus Hameln machte zunächst durch die Präsentation der Containerbewohner von Big Brother als 3D-Modelle auf der offiziellen Homepage auf sich aufmerksam.

Die hauseigene Lösung für 3D-Objekte, das unter der **GNU Public Licence (GPL)** veröffentlichte Format Open-SPX, hatte man schon 1998 entwickelt². Im Jahr 2000 erwarb man den 3D-Scanner-Hersteller *Dimension 3D-Systems*³. Daher ist verständlich, dass man zur Gewinnung der Modelle vor allem den Einsatz eines Scanners empfiehlt. Alternativ ist aber auch ein kostenloses 3ds max Plugin erhältlich, allerdings vorerst (Stand:2.4.01) nur für die Version 3. Beim Export aus 3ds max ist zu beachten, dass vor dem Export sämtliche Transformationen auf die Standardwerte zurückgesetzt (Utilities>ResetXForm) und die verwendeten Texturen manuell in den gleichen Ordner wie das spx-file kopiert werden müssen.

Animationen können nur im Authoring-Tool **Marlin** hinzugefügt werden. Dieses ist allerdings bisher nur in Verbindung mit dem Kauf eines 3D-Scanners beziehungsweise mit der Software **3D-Snapper** für 99 DM erhältlich. Der 3D-Snapper erstellt photorealistische 3D-Modelle aus 12 Photos des zu generierenden Objektes. Der Benutzer muss nur das Objekt auf jedem Bild ähnlich wie in Photoshop freistellen und Kontrollpunkte auf einer zu verwendenden Unterlage kennzeichnen, den Rest übernimmt das Programm. Im Test ist es mir jedoch nicht gelungen, brauchbare Ergebnisse zu erlangen.

Bisher nur angekündigt ist das Tool **Face Factory**, das eine ähnliche Funktionalität wie 3DmeNow bieten soll.

Zur Darstellung kann sowohl ein **Plug-In** als auch ein **Applet** verwendet werden.

Optisch bleibt Open-SPX durch störende Treppeneffekte an den Kanten (Aliasing) hinter der Konkurrenz zurück. Auch Lichter werden nicht unterstützt, die jeweilige Lichtsituation wird nur über die Farbe der Textur wiedergegeben. Interaktion wird durch Actionmaps auf der Textur ermöglicht.

Das GPL-Format läßt sich gemäß eigenen Anforderungen nach Belieben erweitern, C++ Kenntnisse vorausgesetzt. Da für die Veröffentlichung von Modellen keine Gebühren anfallen, lassen sich auch kommerzielle Lösungen mit Open-SPX ohne Lizenzkosten realisieren.



Käufer des 3D-Scanbooks erhalten das Authoring-Tool Marlin dazu

¹ <http://www.interzart.de>

² <http://www.interzart.de/company/historie.html>

³ <http://www.dimension3d.de/>

8.4 Hyper Space (*Kaon Interactive*)

Mit einer komplett in **Java** realisierten Lösung versucht die junge Firma *Kaon Interactive*¹ ebenfalls im e-commerce Sektor Fuß zu fassen. Da man sich selber auf die Produktion photorealistischer 3D-Modelle konzentrieren möchte, darf jeder Entwickler ohne anfallende Lizenzgebühren nicht-photorealistische Modelle aus 3ds max mit dem **Hyper Space** genannten Viewer-**Applet** veröffentlichen.

Zur Darstellung der Modelle ist kein Download nötig, nur 40K fallen für die **Java-Engine** an. Außerdem ist das dokumentierte **API** erhältlich, so dass auch die Integration des 3D-Formates in eigene Applikationen ermöglicht wird. Als Bedingung gilt jedoch, dass das Kaon – Logo für mindestens 2 Sekunden eingeblendet werden muss. Die dargestellten Modelle werden bei *Kaons* Lösung beim Bewegen reduziert, um die Performance zu verbessern. Sobald sie sich wieder im Ruhezustand befinden, werden sie aber nach und nach geglättet, bis wieder eine extrem gute Darstellungsqualität erreicht ist [rp 3]. Objekte können zwar nur **Gouraud**-schattiert dargestellt werden. Dieses Manko wird jedoch durch Anti-Aliasing und den dynamischen Austausch von Texturen abhängig von der Kameradistanz wieder wett gemacht.

Um Interaktionsmöglichkeiten zu schaffen, werden versteckte Kugeln an die 3D-Objekte geklebt. Diese **Hotspots** können dann beim Überfahren mit der Maus Bilder darstellen oder auf Mouseclick zu anderen URLs verweisen.

Simple Animationsmöglichkeiten sind auch vorhanden, lassen sich aber nur in **Activate!**, dem kostenlosen Authoringtool, erstellen und nicht schon aus 3ds max übernehmen.

Da dieses ebenfalls komplett in **Java** gehalten ist, wird man nicht wie bei den meisten Konkurrenzformaten auf Windows eingeschränkt. Ein neuer und speziell für den Interneteinkäufer gedachter Ansatz ist der ins Interface integrierte Maßstab, mit dem man die exakten Maße der Modelle nehmen kann. Im Developerbereich der *Kaon-Homepage* finden sich neben Einsteigertutorials auch Beispiele für fortgeschrittene Anwender.

Zwar findet man noch keine Hyper Space – Seiten im Netz, aber *Kaon* hat vor kurzem einen Deal mit *Tweeter*², der größten Elektronikette Amerikas, ausgehandelt und wird deshalb schon bald präsenter sein.



Die Hyperspace-Darstellung des Videorekorders in Bewegung ...



... und geglättet nach kurzer Ruhepause.

¹<http://www.kaon.com>

²<http://www.tweeter.com>

9. Systeme mit Möglichkeiten zur Charakteranimation

9.1 Shout3D

9.1.1 Technologievorstellung

Auch wenn *Eyematic*¹ mit **Shout3D** einen eigenen Standard schuf, merkt man ihnen das jahrelange Engagement für offene Web3D-Standards immer noch deutlich an. Das Format **s3d** besteht aus den klassischen VRML-Elementen, die als Nodes bezeichnet werden, erweitert um eigene. Dies hat den Vorteil, dass man sehr viele verschiedene 3D-Pakete zum Export von **VRML-Daten** nach Shout3D verwenden kann. Für anspruchsvollere Inhalte, und da die VRML-Files sonst häufig von Hand nachzubessern sind, sollte man jedoch lieber 3ds max verwenden. Seit Shout3D im Januar 2001 zur Version 2.0 herangereift ist, bietet der 3ds max Exporter allen erdenklichen Komfort. Sowohl der Export von Vertexanimationen als auch von humanoiden Charakteren mit Biped wird voll unterstützt. Shout3D ist ebenfalls eine pure **Java1.1** – Lösung und demzufolge auf allen **Java**-Plattformen einsetzbar [3dg 3]. Eingesetzt wird Shout3D z.B. vom Onlinehändler *Amazon*², der mittlerweile die angebotenen Spielwaren als Shout-Visualisierungen bereit hält.

Wer auf höhere Performance Wert legt, kann schon heute den im Betastadium befindlichen **OpenGL**-Beschleuniger testen.

Die Lizenzgebühren für Shout3D sind relativ niedrig. Auch für professionelle Anwendungen fallen nur \$100 an. Mittels des kostenlosen Shout3D-Wizards lassen sich jedoch nur simple Applikationen zusammenklicken. Zur Erstellung komplexerer Anwendung muß man das mitgelieferte **Java-API** verwenden, für Nichtprogrammierer ein ziemlicher Einarbeitungsaufwand. Wer schon ein wenig in **Java** bewandert ist, sollte mit dem Shout3D-**API** dank der mitgelieferten Dokumentation eigentlich ziemlich schnell zu Rande kommen. Wichtig ist bei der Entwick-

lung, dass die zur Darstellung verwendete Klasse von Shout3DApplet abzuleiten ist.

Wenn man den Shout3D-Wizard explizit zur Optimierung des Web-Archives auffordert, prüft dieser, von welchen Shout3D-Klassen unsere Applikation überhaupt Gebrauch macht. Nur diese werden dann auch für die Webapplikation mit ausgegeben.

Shout3D verfügt über zwei Varianten des Anti-Aliasing: Bei progressivem Anti-Aliasing erfolgt eine Darstellungsglättung sobald die Kamera nicht mehr bewegt wird, bei normalem Anti-Aliasing werden die Kanten zwischen 3D-Modell und Hintergrund bei jedem Neuzeichnen geglättet. Damit 3ds-Modelle auch in Shout über weiche Schattierungsverläufe an den Kanten und verschieden texturierte Flächen verfügen können, wurde als neuer Geometrietyp **Multimeshes** eingeführt. Von dieser Bezeichnung sollte man sich jedoch nicht in die Irre führen lassen: mit Intels MultResolution-Meshes hat dieser Geometrietyp nichts zu tun. Die zweite Neuheit ist die Unterstützung von NURBS (NonUniformRationalB-Spline). Leider lassen sich diese bisher nur aus dem VRML97-Editor **Spazz3D**³ exportieren.

PostRenderEffects ermöglichen es, die beschriebenen Shout3D-Szenen vor der Darstellung zu bearbeiten. Dadurch werden nicht nur Effekte wie Nebel, das Invertieren der Farbwerte oder wellenartige Verzerrungen der Darstellung sondern auch Rollovereffekte bzw. mit Funktionalitäten hinterlegte 3D-Elemente möglich: Der sogenannte Picker schießt von der aktuellen Mausposition einen Strahl durch die 3D-Szene. Wenn dieser auf ein bestimmtes Objekt trifft, kann darauf eine Reaktion wie z.B. die Einblendung eines Informationstextes erfolgen.

¹<http://www.eyematic.com>

²<http://www.amazon.com>

³<http://www.spazz3d.com>

Leider konnten Darstellungsqualität und Exportmöglichkeiten im Oktober 2000 noch nicht mit Pulse3D konkurrieren, ansonsten hätte ich mich sicher schon auf Grund der grenzenlosen Erweiterbarkeit für eine Shout3D – Beispielimplementierung entschlossen. Wenn man sich überlegt, dass Shout3D von *Eyematic* als Proposal für den X3D-Standard eingereicht worden ist, könnte das langfristig doch noch einen Sieg des offiziellen Web-Standards mit sich bringen.

Auch für die Zukunft hat *Eyematic* große Pläne: Neben einem 3DMeNow-ähnlichen Tool zur raschen Erstellung photorealistischer Köpfe arbeitet man momentan an einem System zur Revolutionierung der Facial Animation: Glaubt man den Versprechungen, wird es schon sehr bald möglich sein, Sequenzen für die Gesichtsanimation in Echtzeit aus Videodaten zu generieren. Dieses wäre genau das Tool, das momentan noch schmerzlich vermisst wird, um Avataren auf Webseiten wirklich zum Durchbruch zu verhelfen.

9.1.2 Shout3D in der Praxis

Die Behauptungen bezüglich des Biped-Exporters wollte ich dann aber doch nicht ohne eigene Testresultate stehen lassen. Erste Versuche mit Character Studio 3.0 (CS 3) und 3ds max 4.0 bescherten mir zunächst nur jede Menge Abstürze. Erst durch einen Hinweis der Entwickler auf Probleme mit CS 3 im Webboard der Shout3D-Homepage versuchte ich noch einmal mit der Vorgängerversion 3D Studio Max 3.11, CS 2.2 und einem Standardbiped mein Glück und siehe da, die Darstellung des Charakters entsprach nahezu 1:1 der im 3ds-Viewport. Bipedes mit deformable binding werden genauso unterstützt wie bulges (Muskelausdehnungen) oder tendons (Sehnen). Dieses ist weitaus komfortabler als Lösungen von Konkurrenzformaten wie z.B. Pulse3D, die nur ‚rigid binding‘ erlauben.

Wenn der Keyframeexport nicht präzise genug ausfällt, weil z.B. keine inverse Kinematik unterstützt wird, kann man alternativ Samples der einzelnen Frames exportieren. Erstaunlicherweise fällt die Länge der Animation im komprimierten Einsatzformat so gut wie nicht ins Gewicht. Eine 200 Frames lange E-sha Animation war mit 382 KB nur um 4KB kleiner als die Variante mit 1000 Frames. Um eine Textur zu verwenden, ist diese entweder in den gleichen Ordner wie das s3z – File zu kopieren oder deren Adresse in der s3d – Datei vor dem Import in den Wizard mittels eines Texteditors anzupassen. Außerdem ist mir aufgefallen, dass in der vorliegenden Version Texturdarstellung und **OpenGL**-Unterstützung nur funktionieren, wenn man die ausgegebene HTML-Datei direkt über den Browser und nicht aus dem Wizard heraus öffnet. Dieses ist zwar verwunderlich, stellt aber kein größeres Problem dar. Von der Beta-Version der **OpenGL**-Unterstützung würde ich abraten, da viele Effekte wie z.B. das **Bump Mapping** noch nicht unterstützt werden und im Test Darstellungsfehler auftraten.



Meine E-sha im neuen Glanz von Shout3D 2.0

9.2 Pulse3D

1997 begann die zuvor durch Spiele wie Iron Helix oder Bad Mojo bekannt gewordene Firma *Pulse Interactive* die Entwicklung ihrer eigenen Web3D-Technologie, **Pulse3D**¹. Mit dieser ist es der in San Francisco ansässigen Firma gelungen, zum Marktführer für webbasierte Charakter Animation zu werden. Zu den Kunden zählen mittlerweile so große Namen wie *The Jim Henson Company*, *Warner Brothers* oder *Mattel*.

Die zur Contenterstellung nötigen Tools, den **Pulse Creator** und das 3ds-Plugin **Pulse Producer**, kann sich jeder kostenlos von der Homepage herunterladen. Da das komplette Paket mit sämtlichen Tutorials mittlerweile auf 48 MB angeschwollen ist, kann man sich die Entwicklungssoftware mitsamt einem kostenlosen NTSC-Tutorialvideo auch nach Hause schicken lassen.

Lizenzgebühren für Pulse fallen erst an, wenn wirklich Content veröffentlicht wird: für \$2.500/Jahr kann man 5 Webseiten mit 3D-Inhalten bestücken.



Im April 2001 werden die Besucher der Pulse-Homepage von Kermit dem Frosch begrüßt

Das Rennen um das kleinste PlugIn geht bisher klar an Pulse: nur 370 KB groß ist der Player für Mac und die Windows-Familie. Die Pulse-Inhalte können aber auch in **Real-Media Streams** eingebettet oder vom **Quicktime-Player** wiedergegeben werden.

Wer sich an die Pulse3D-Entwicklung macht, sollte ein wenig Zeit mitbringen: die Möglichkeiten des Creators sind so umfangreich, dass das Tool den Einsteiger zunächst überfordert. Um die Entwicklungsarbeit zu erleichtern, bietet *Pulse Interactive* kostenlosen Developer-Support, der in der Regel innerhalb von 24 Stunden erfolgt.

Herausragend ist bei Pulse zunächst die durch dynamische Lightmaps sehr gute Darstellung, die auch dynamische Lichteffekte, glänzende Oberflächen, Transparenzen, Anti-Aliasing und **Bumpmaps** unterstützt. Geometriedaten können außer aus 3ds max auch über die verbreiteten 3D-Formate DXF und **OBJ** importiert werden. Der Skelettbau lässt sich, falls das verwendete 3D-Paket diese Funktionalität nicht bieten sollte, direkt im Creator vornehmen. Zur Animation der Skelette kann inverse Kinematik eingesetzt werden, und der direkte Import von bvh-MotionCapture-Daten ist ebenfalls möglich. Transparenzen lassen sich animieren, Lippen-synchronisation kann beim Authoring automatisiert werden und über die Skriptsprache **PulseScript** lassen sich eigene Funktionalitäten implementieren.

Seit der Version 4.3 kann Texturkompression durch das leistungsfähige **Wavelet**-Verfahren erfolgen. Zur Übertragung lassen sich Animationsdaten und MP3-Audiofiles zu streambaren Paketen zusammenfassen.

Über Praxiserfahrungen mit Pulse3D werde ich ausführlich in dem Abschnitt über meine Beispielimplementierung berichten.

¹ <http://www.pulse3d.com>

9.3 B3D (Brilliant Digital)

9.3.1 Technologievorstellung

*Brilliant Digital*¹ hat das hauseigene Web3D-Format **B3D** auf streambare, webbasierte Filme zugeschnitten. Setzte man zuerst noch auf die Wiedergabe durch den eigenen Digital Projector, sind mittlerweile auch Möglichkeiten zur Integration der Inhalte in Webseiten vorhanden.

Die Funktionen des Projectors sind in drei Kategorien unterteilt: Unter Channel hat man die Möglichkeit, aus allen im Cache gespeicherten Filmen auszuwählen und unter Settings lässt sich die Hardware-3D-Beschleunigung an- und auszuschalten. Der Autor kann zusätzlich auswählen, durch welche Effekte die Inhalte während des Abspielens verfremdbar sein sollen: durch das Aktivieren von Blow Up z.B. beginnen diese zu pulsieren, Disco Lights sollten selbsterklärend sein, in der Jellyworld wabert alles hin und her und als Wireframe wird die zu Grunde liegende Struktur der Geometrie offenbart. Ich hoffe, dass es dem Autor überlassen ist, das im unteren Bereich des Players angebrachte Werbebannerfenster zu verwenden oder auch nicht. Bei der Wiedergabe von Inhalten von *Brilliant Digital* selber fordert dieses unangenehm oft neue Anzeigen an. Der Player kann nicht nur b3d-Elemente darstellen, sondern auch reguläre Webseiten. Somit ist die Integration aller Webformate möglich.

An *Brilliant Digital*s Beispiel für den Einsatz zum Online-Shopping sollte man wohl noch ein wenig nachbessern: Der Benutzer muss über die Cursortasten oder ein eingeblendetes Interface selber durch einen Shop navigieren, was mehr schlecht als recht gelingt. Durch Überfahren mit der Maus kann er die Preise der jeweiligen Produkte erfahren. Diese Informationen sollen direkt einer Backend-Preis-Datenbank entstammen können.

Bei seinen Authoringtools verfolgt B3D eine andere Strategie als die Konkurrenz: Ohne zu Bezahlen kann man zwar ein wenig mit den Tools experimentieren, die Funktionalität ist aber doch so eingeschränkt, dass man nicht wirklich weit kommt. Die Installation ist nur durch den **b3d-Installer** direkt von der Homepage möglich. Ärgerlich wird das vor allem, wenn während des 13 MB umfassenden Downloads des Authoringtools **B3D Studio 2.0** die Leitung unterbrochen wird. In diesem Fall helfen nur viel Geduld und ein Neuanfang.

B3D-Projekte sind durch eine Ordnerstruktur gegliedert, wobei eine Trennung zwischen Geometrien und Animationsdaten erfolgt. Die im **S3D**-Format vorliegenden Geometrien werden aus dem streambaren B3D-Format nur referenziert. Somit hat der Betrachter nur den Download des S3D-Files abzuwarten, bevor der Film beginnen kann. Mit Shout3D hat dieses übrigens nur die Formatbezeichnung gemeinsam.

Möglichkeiten für eigene Erweiterungen sind nicht gegeben. Da es sich bei dem verwendeten Player um eine **ActiveX**-Komponente handelt, kann dieser auch in Worddokumente oder PowerPoint-Präsentationen eingebettet werden. Gestreamt werden Soundfiles und Animationsdaten.

Da im Unterschied zu einer Flashanimation Bild und Ton synchronisiert werden können, wurde B3D in den letzten Monaten zunehmend für webbasierte 3D-Musikvideos verwendet.

¹ <http://www.brilliantdigital.com/>

9.3.2 B3D in der Praxis

Die Oberfläche des B3D Studios erinnert an einen nichtlinearen Animationseditor. Beim Export aus 3ds sind die einzelnen Animationspuren durch die Angabe von Keyframes zu kennzeichnen und mit einer Bezeichnung zu versehen. Die automatische Lippensynchronisation konnte ich leider nicht testen, da ich mit der Trial-Version leider keine eigenen Charaktere ins S3D-Format exportieren konnte.

Als **Phonem** bezeichnet man die kleinsten unterscheidbaren Lauteinheiten der menschlichen Sprache. Für diese sind Varianten des Kopfes zu erstellen, die über einen Morphmodifikator als Targets zu definieren sind. Vor dem Export sind für diese Morphtargets jeweils Keyframes zu setzen und dann mit der entsprechenden Bezeichnung zu exportieren. Bei den minimalen Einstellungen zur Lippenanimation setzt B3D Phoneme für neutral, B/M/P, Th, N/NG und A/H voraus.

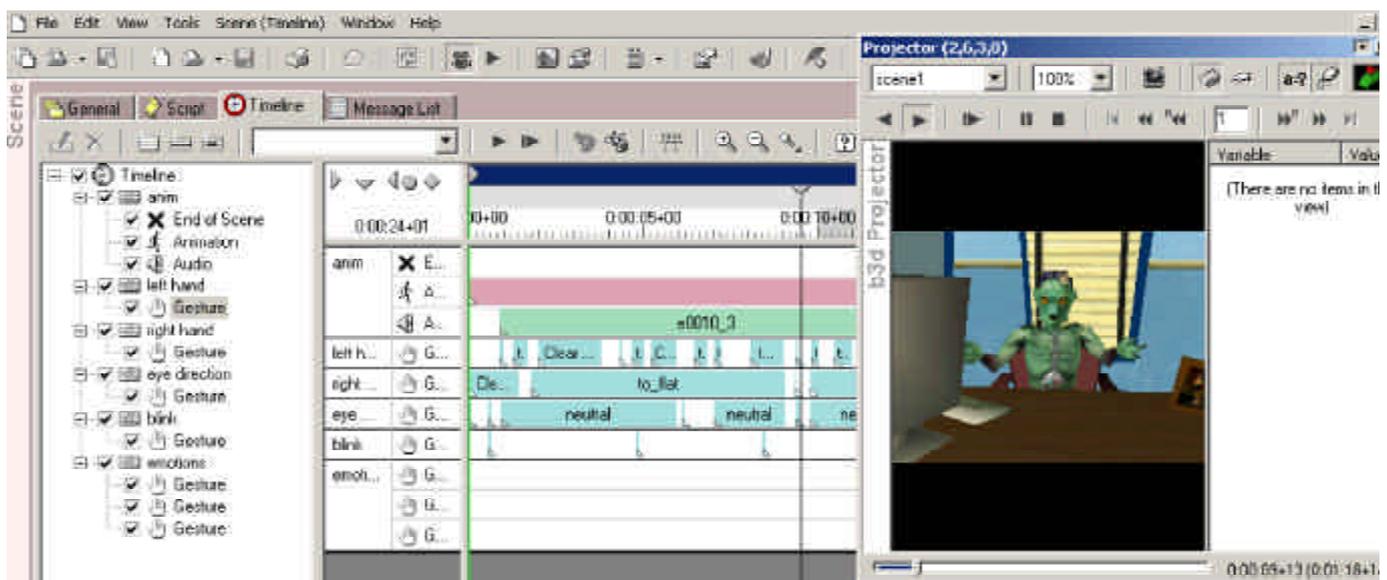
Bei der Verwendung von Biped sollte man die einzelnen Glieder zunächst in herkömmliche Meshobjekte umwandeln. Danach kann der

Charakter über den eigenen **B3DSkin – Modifier** an diese Skelettstruktur gebunden werden.

Leider kann nur eine Kamera pro Max-File exportiert werden. Ein Perspektivenwechsel wird so nur über eine animierte Kamera möglich.

Zur Beleuchtung werden nur gerichtete und Omni-Lichter unterstützt, dann allerdings mit Schattenwurf.

Der Vorteil von B3D liegt darin, dass nach Erwerb der Softwarelizenzen beliebig viele Inhalte angefertigt werden können. Leider beschränken sich die Möglichkeiten bisher weitgehend auf Webisodes. Zukünftige Versionen sollen auch über Skripting – Möglichkeiten verfügen und somit B3D auch für Spiele und interaktive Applikationen interessant machen.



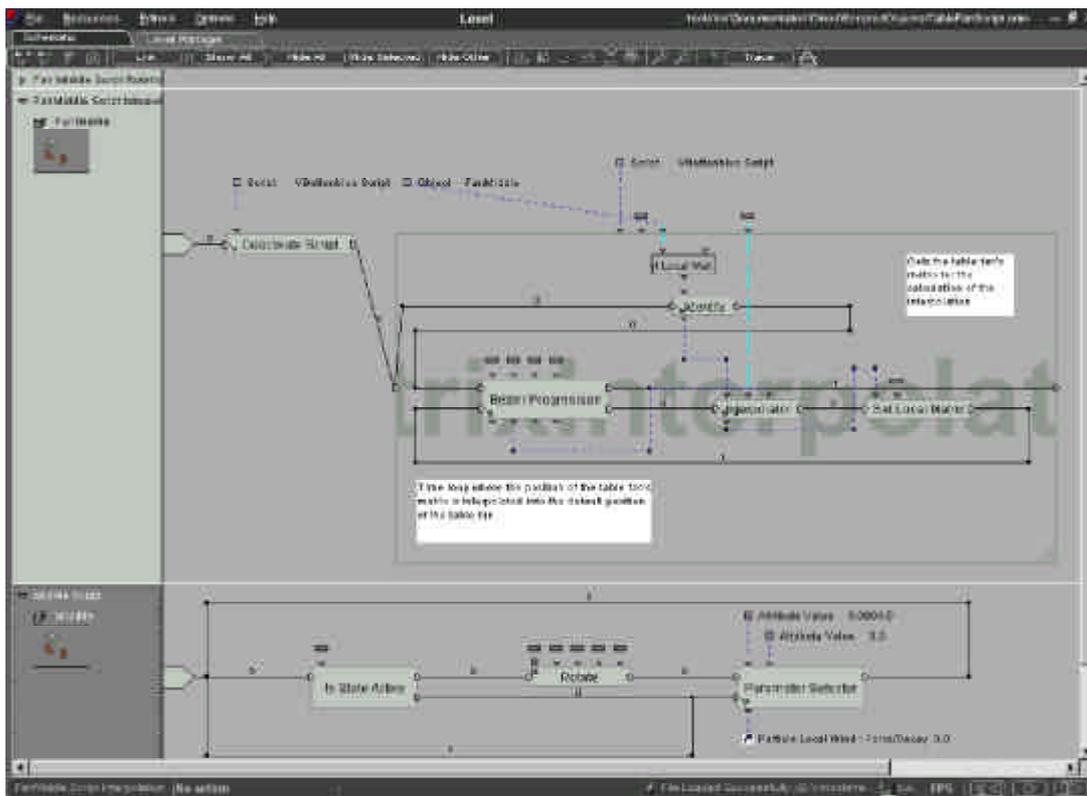
Anhand des Intromovies werden ein paar Grundprinzipien von B3D Studio erläutert

9.4 Virtools 2.0

Unter dem Namen **Nemo** war **Virtools**² das erste Entwicklungspaket, dem der Ruf anhaftete, das Potential zu besitzen, so etwas wie der **Macromedia** Director² für 3D-Spiele zu werden [3dl 1] [dp 99a]. Lange Zeit jedoch war Rätselraten um die Pioniersoftware angesagt: Der Link zu nemo.com führte ins Leere, und so vermutete auch ich zunächst, dass es der Software nicht gelungen wäre, sich zu etablieren. Dass dem nicht so ist, unterstrichen **Virtools**, die Entwickler von Nemo, auf einer großangelegten Promotion-Tour Anfang 2001 und ließen schon erste Blicke auf **Virtools 2.0** erhaschen. Den Namen Nemo hatte sich leider eine französische Multimediaagentur nur wenige Tage vor der Warenzeichenanmeldung sichern lassen. Die neue Version Virtools 2.0 läuft deshalb noch unter dem Codenamen **„Next Generation“**.

Schon mit Nemo 1.0 ließen sich 3D-Applikationen sowohl für Webseiten, mittels eines **Webplayer** genannten Plugins, als auch zur Integration in **Macromedias** Director sowie als eigenständige Applikationen produzieren.

Einzelne Objekte können nahtlos aus 3ds max übernommen und mit verschiedenen Animationen als Behaviors versehen werden. Mittels eines an Legosteine erinnernden Steckverfahrens, kann man aus sogenannten Building Blocks komplexe Interaktionsschemen erstellen. Der Druck auf eine Taste kann z.B. die ‚nach vorne laufen‘ – Animation auslösen. Dabei entsprach die Darstellungsqualität schon in Nemo 1.0 in etwa der **Halflife**³-Engine (Egoshooter von *Valve/Sierra*⁴).



Das Stecksystem der Behavior Blocks in Virtools

¹ <http://www.virttools.com/>

² <http://www.macromedia.com/>

³ <http://www.half-life.net/>

⁴ <http://www.sierra.com/>



Screenshot einer mit *Virtools 2.0* entwickelten Demo für ein kommendes **X-Box** Game auf einer *GeForce II*

Mit der günstigen **Creation** – Version konnte man schon für 2.000 DM eigene Entwicklungen machen. Um aber auf das **API** und die Sourcen Zugriff zu haben und die Nemo-Engine in komplexere Applikationen integrieren zu können, musste man schon tiefer in die Tasche greifen und die **Dev**–Version für 10.000 DM erwerben. Diese Trennung wurde beim Generationswechsel aufgehoben. Von nun an gibt es nur noch die Dev-Version, deren Käufer auch eine Schulung erhalten sollen, um das erworbene Produkt richtig einsetzen zu können. Bei größeren Projekten hat man auch die Möglichkeit, Unterstützung durch Virtools-Mitarbeiter zu erhalten. Erste Demos sahen schon sehr vielversprechend aus. Kein Wunder, bietet doch Virtools 2.0 sowohl **OpenGL** als auch **Direct3D** – Support inklusive der Unterstützung der Geometriebeschleunigung durch moderne *nvidia*¹-Grafikkarten.

Neben dem bewährten 3ds max Exporter wird es mit der neuen Version definitiv ähnliche Tools auch für Lightwave und Maya geben. Aus diesen und anderen Paketen war der Export bisher nur über das **Game Exchange Format** möglich.

Mediendateien wie Sound- und Videofiles werden bei Virtools außerhalb der eigentlichen Applikation gespeichert und sind daher problemlos austauschbar. Natürlich können diese auch einer Datenbank entnommen werden. Für das Audiostreaming verwendet man das mp3-Format. Neu hinzu kommt im Laufe des Jahres der Virtools Integrator, ein Programm zum Erstellen serverseitiger Applikationen, die dann auch Multi-User-Games ermöglichen sollen. Das Multiplayergame **Beyond3000**², das *Paraworld*³ mit Virtools für *Pro7*⁴ realisiert hat, verwendet dazu noch eigene **DLLs**. [e.m 1]

¹ <http://www.nvidia.com>

² <http://www.beyond3000.de>

³ <http://www.paraworld.de>

⁴ <http://www.pro7.de>

9.5 WildTangent's Webdriver



Im Juli 1998 wurde die Firma *WildTangent*¹ von Alex St. John, der zuvor für *Microsoft* das **DirectX-API** entwickelt hatte, gegründet. Ihr Ziel war es zunächst, in Kooperation mit *Microsoft* das Software Development Kit **Chromeeffects** weiterzuentwickeln, das es erlaubt, über **XML**-Tags z.B. Webseiten um **DirectX**-Funktionalitäten zu bereichern. Leider wurde dieses Projekt im November 1998 wieder beendet. Enttäuscht darüber beschloss das *WildTangent*-Team, sich von *Microsoft* zu trennen um eine eigene Lösung für interaktive Webinhalte zu entwickeln.

Die erste Betaversion des **Webdrivers** wurde Ende Mai 1999 für Entwickler veröffentlicht. Die Version 1.0 war am 22. September des selben Jahres erhältlich, durfte aber noch nicht für kommerzielle Projekte verwendet werden. Zum Release der Version 2.0 im März 2001 hat man sich endlich auf eine Lizenzpolitik festgelegt, die die Trennung von Online und Stand-Alone Applikationen beinhaltet. Onlineapplikationen (OA) sind in drei Gruppen aufgeteilt:

Unter kleinen OAs versteht man bei WildTangent z.B. Object Viewer oder Logovisualisierungen. Hierfür fallen Lizenzgebühren von \$1000/Produkt und Seite an.

Unter eine mittlere OA würde z.B. meine Tai-Chi E-sha fallen, der Preis liegt bei \$5000/Produkt und Seite. \$20.000 sind für komplexe Onlinewelten oder -spiele zu entrichten.

Werden die Stand-Alone Applikationen regulär vertrieben, gehen 2% der Bruttoeinnahmen an die Game-Engine Entwickler, mindestens jedoch \$10.000. Für Demoversionen, die dem Endnutzer kostenlos zur Verfügung gestellt werden sollen, fallen einmalig \$1.000 an und bei nichtkommerziellen Produkten verzichtet man ganz auf Lizenzgebühren.

Mit der Version 2.0 konnte WildTangent zusätzlich zum bestehenden 3D Studio MAX 3.1 - Exporter mit Betaversionen zum Export aus 3ds max 4 und Maya aufwarten. Der Maya-Exporter integriert sich nahtlos in die Oberfläche und unterstützt auch den Export von Clips des nichtlinearen Animationseditors **Trax**. Um mit weniger mehr erreichen zu können, ist **Smooth Bind** der einzige unterstützte Deformer. Der Export smooth gebundener Charaktere funktionierte im Test auf Anhieb und reibungslos.

Das neue SDK ist nicht mehr an eine Programmiersprache gebunden, **Java** oder **JavaScript** können genauso Verwendung finden wie C++, Delphi oder VisualBasic. Da die Entwicklung mit WildTangent den Entwicklern die hardwarenahe Programmierung erspart, ist zu erwarten, dass Spieleentwicklungen mit diesem SDK in einem Bruchteil des bisherigen Zeitbedarfs erfolgen können. Der interessierte Entwickler findet im Developer - Bereich der WildTangent Homepage jede Menge Tutorials für den schnellen Einstieg.

¹ <http://www.wildtangent.com>

Joysticks und ähnliche Spielecontroller können problemlos verwendet werden. Die Engine beherrscht auch aus dem Browser die Darstellung im Vollbildmodus, und Möglichkeiten für Multi-User-Onlinespiele sind ebenso vorhanden wie Kollisionserkennung sowohl auf **Bounding Box** als auch auf Polygonebene.

Dass der WebDriver in erster Linie für Spiele konzipiert ist, hindert den Entwickler nicht daran, auch andere Applikationen mit diesem SDK anzufertigen: WildTangent bietet darüber hinaus auch Screensaver, Visualisierungsplugins für die bekanntesten MP3-Player und einen Viewer für streambare Landkarten an. Von WildTangent selber entwickelte Spiele finanzieren sich häufig durch Werbefenster im unteren Teil des Bildschirms.

Dass in Zukunft noch sehr viel von WildTangent zu erwarten ist, zeichnet sich auch dadurch ab, dass seit November 2000 der Kult-3D-Artist **Paul Steed**, zuvor Artdirector bei *ID Software*¹, den Entwicklern von **Quake**, zum Kernteam gehört.

Wenn *Microsofts* Windows-Systeme die einzige Zielplattform darstellen, was ja bei Spielen nicht unüblich ist, lässt sich mit dem WildTangent-SDK und etwas OOP-Know-how schnell eine Multimedia-Applikation nach eigenen Vorstellungen realisieren, wobei dem Entwickler kaum Grenzen gesetzt sind.



WildTangent's Snowboard-Game finanziert sich durch Werbefenster

¹ <http://www.idsoftware.com>

10. Tabellarische Gegenüberstellung

Stand: 05/2001	Systeme zur Objektvisualisierung			
	Viewpoint Experience Technologies	Cycore Cult3D	Interzart Open-SPX	Kaon Hyperspace
Lizenzmodell und Kosten	<p>Ab \$2.500, durchschnittlich \$10.000 je nach Einsatzgebiet</p> <p>kostenlos für nichtkommerzielle Zwecke</p>	<p>Staffelung nach Besucherzahlen:</p> <p>von \$1.500 / Jahr bei < 10.000 Besuchern bis \$15.500 / Jahr</p> <p>\$500 für eine Produktlinie, die offline vertrieben wird, für Entwicklungspartner sowie Hochschullizenzen</p> <p>umsonst für Hobbyentwickler</p>	<p>Kosten fallen nur beim Erwerb eines 3D-Scanners und der mitgelieferten Software an</p> <p>Bei Veröffentlichung keine Lizenzgebühren</p>	<p>Kostenlos solange man keine photorealistischen Objekte darstellt.</p> <p>Photorealistische Objekte dürfen nur von Kaon selber produziert werden</p> <p>Kosten: ~ \$600 / Modell \$200-300 / Photographie</p>
Plug-In	Viewpoint Media Player, Minimalversion: 341 KB Vollversion: 1,405 KB	Cult3D Viewer IE Version: 1,2 MB NC Version: 1,17 MB	SPX-Plugin (770 KB) Java-Applet (60 KB) ermöglicht Lösungen ohne Plug-In	Java-Applet (40 KB) ermöglicht Lösungen ohne Plug-In
Erweiterbarkeit	Für Viewpoint selber sehr leicht, SDK kann angefordert werden	Kostenloses Java- API	Ja, da Format unter GNU Public Licence	Kostenloses Java- API
Hardwaresupport	Noch nicht, OpenGL / Direct3D - Support angekündigt für Ende 2001	Bisher nur für nvidia-Grafikkarten	Noch nicht	Nein
Unterstützte Plattformen	Windows 9x/NT/2000 MacOS (Beta)	Windows 9x/NT/2000 MacOS, Linux 2.0/ Red Hat	Windows 9x/2000 MacOS Dank OpenSource-Plugin leicht portierbar	Alle Java -Plattformen
Animationsmöglichkeiten	Bisher nur Objekttransformation, Texturanimation, Kameraanimation	Objekttransformation, Vertexanimation, Partikelsystem , Texturanimation über Hotspots, Kameraanimation	Kameraanimation, steife Objektanimation	Kameraanimation, steife Objektanimation
Unterstützte Medientypen	Vektorformate geglättete Texte Grafikformate mit Alphakanälen	Sound (Wav)	Sound Images Video	Sound
Interaktionsmöglichkeiten	Über versteckte Geometrien (Widgets): Animationen triggern, Rollovereffekte	Über Mouseevents auf Objekten oder Hotspots: URLs öffnen, Messages an die Hostapplikation schicken, Conditions prüfen, Animationen starten, Sound abspielen ...	Actionmaps auf der Textur können mit Funktionen versehen werden	Hotspots auf der Hintergrundgrafik oder den 3D-Modellen können Roll-overtexte anzeigen oder durch Mouseclick auf Webseiten verweisen

Stand: 05/2001	Systeme zur Objektvisualisierung II			
	Viewpoint Experience Technologies	Cycore Cult3D	Interzart Open-SPX	Kaon Hyperspace
Darstellungsverfahren	Lightmaps Unterstützung für Bumpmaps und Glanzeffekte (specular highlights) dynamisch projizierte Schattenebene Anti-Aliasing	Feste Lichtquelle Bumpmaps Reflectionmaps Anti-Aliasing Specular Highlights	Lichtsituation wird nur durch Textur wiedergegeben Reflectionmaps Opacity	Diffuse feste Lichtquelle, da Lichtsituation schon über die Textur wiedergegeben wird Progressives Anti-Aliasing
Verbreitung	Häufig	Häufig	Noch selten	Noch sehr selten
Exportmöglichkeiten aus 3D-Programmen	3ds max, Poser, truespace, polytrans, Solidworks und mehr via OBJ	3ds max, Maya	3ds max	3ds max
Streaming	Objekte werden progressiv aufgebaut	nicht unterstützt	Objekte werden progressiv aufgebaut	Objekte werden progressiv aufgebaut
Entwicklersupport	Ausführliche Anleitungen als PDF-Files	Communitysite: www.worldof3d.com Onlineschulung	Anfragen per e-mail, Tutorials für das SDK unter http://scanware.dimension-3d.com/	Jede Menge Tutorials auf der Kaon-Homepage
Webbeispiele	http://bpgprod.sel.sony.com/bpcccontent/bpchome/3danimations.99999.BPC.html	http://www.cnn.com/2001/TECH/space/04/10/mars.odyssey/index.html http://www.tour-eiffel.com/ http://www.ericsson.com/consumers/spg.jsp?page=frames	http://www.extravaganza.de/Shop/indexa.htm http://www.motorola.de/mobitel/public/produkte/datenblaetter/00159/datenblatt_td.shtml	http://www.kaon.com/hsgallery/index.html

Stand: 05/2001	Systeme mit Möglichkeiten zur Charakteranimation				
	Shout 3D 2.0	Pulse3D	B3D	Virtools 2.0	WildTangent
Lizenzmodell und Kosten	Professional \$100 Educational \$80 („Educational only“ beim Ladevorgang) Trial Free (Shout3D - Banner zieht sich durch den Inhalt)	1-5 Seiten \$2.500 / Jahr	B3d Studio 2.0 \$995 B3d Exporter für 3DS Max \$195	Virtools Dev - 10.000 DM + Lizenzgebühren	Für Logos oder Objektvisualisierungen \$1.000 / Produkt und Seite Mittlere Webprojekte \$5.000 / Produkt und Seite Onlinespiele oder begehbare Welten \$20.000
Plug-In	<i>Java-Applet</i> für Lösungen ohne <i>Plug-In</i> Optionales <i>OpenGL</i> -beschleunigtes <i>Plug-In</i> im Betatest (138 K)	Pulse3D Player (370 KB) Content integrierbar in Quicktime und RealVideo	Digital Player (~700KB) kann nur über die b3d-homepage installiert werden	Webplayer für Version 2.0 < 1 MB für Version 1.0 1,8 MB	Webdriver (<1 MB) wird automatisch installiert, wenn Content dargestellt werden soll
Erweiterbarkeit	Unbegrenzt dank <i>Java-API</i>	Nach Erlernung der Pulse-eigenen Scriptsprache PulseScript	Nein	SDK für eigene Behaviorblocks und Sources in Dev enthalten	Kostenloses SDK
Hardwaresupport	<i>Applet</i> -> nein <i>Plug-In</i> -> <i>OpenGL</i>	<i>Direct3D</i> (Windows) <i>OpenGL</i> (MacOS)	<i>Direct3D</i> <i>OpenGL</i> angekündigt	<i>OpenGL</i> / <i>Direct3D</i>	<i>Direct3D</i>
Unterstützte Plattformen	Alle <i>Java</i> -Plattformen	Windows 9x/2000 NT ohne Hardwarebeschleunigung MacOS	Windows 9x/2000 NT bisher ohne Hardwarebeschleunigung	Windows 9x/NT/2000 MacOS geplant für Sommer 2001	Windows 9x/2000 NT ohne Hardwarebeschleunigung
Animationsmöglichkeiten	Biped-Charakteranimationen (rigid + deformable) MorphTargets Kollisionserkennung Vertexanimationen	Automatisches LipSync beim Authoring, dynamische Lichteffekte, Vertexanimationen, Morphtargets	Automatisches LipSync, Animationsübernahme aus 3ds max, Lichtanimation	<i>Partikelsystem</i> , Biped-Charakteranimationen, Vertexanimation, Dynamics mit Havok (optionale Engine)	Smooth skinned characters, Licht- und Kameraanimationen, Vertexanimationen
Unterstützte Medientypen	Sound (Au + Wav)	Sound (Wav) Kompression via MP3 oder GSM (default)	Sound	Sound, Video	Sound (Wav + Midi)
Interaktionsmöglichkeiten	3D-Objekte mit Funktionalitäten (Picker), Keyboard	Durch Objektclicks oder erfüllte Bedingungen (Conditions): Auslösen von Animationen, Verweisen auf Webseiten ...	Story Branching, Hotspots	Nahezu unbegrenzt durch eigenes Behaviorsystem	Unbegrenzt durch eigenes SDK

Stand: 05/2001	Systeme mit Möglichkeiten zur Charakteranimation II				
	Shout 3D 2.0	Pulse3D	B3D	Virtools 2.0	WildTangent
Darstellungsverfahren	Headlight, directional lights, <i>Bumpmaps</i> , Specular Maps, Cubic Environment Maps Opacity Progressive Anti-Aliasing	Dynamische Lightmaps Reflection Mapping Opacity Anti-Aliasing	Unterstützung von directional / omni lights Übernahme von Texturfarbe Reflection Mapping Opacity	Dynamische Lightmaps ähnlich aktuellen Gameengines	BSP (Binary Space Partitioning) - Szenen Dynamische Lightmaps Point, directional und parallel point lights mit dynamischem Schatten ähnlich aktuellen Gameengines
Verbreitung	Häufig	Sehr häufig	Viele eigene Webisodes, wenig von anderen Anbietern	Im Web eher selten	Bisher vor allem eigene Produktionen
Exportmöglichkeiten aus 3D-Programmen	3ds max Spazz3D andere via VRML	3ds max, Maya angekündigt andere via <i>OBJ</i> , DXF	3ds max Exporter (kostenpflichtig) Maya Exporter in Vorbereitung	3ds max, andere via <i>Game Exchange Format</i>	3ds max maya
Streaming	Texturen werden nachgeladen	MP3s Animationsdaten	Audio- und Animationsdaten	Audiodaten	Modells, Texturen, Audiofiles
Entwicklersupport	Messageboard auf www.shout3d.com	Knowledge Base Kostenloser schneller Entwicklersupport per E-Mail	Tutorials auf der Homepage	Seite zum Austausch für Entwickler: http://www.theswapmeet.com	Tutorials sind im kompletten Entwicklungspaket enthalten, mehr unter: http://www.wildtangent.com/developer/
Webbeispiele	http://www.excitextreme.com/fashion http://www.shout3d.com/scripts/loadpage.pl?user_id=id&file=gal_tootsies.htm	http://www.pulse3d.com/explore/gallery.asp	http://www.brilliantdigital.com/movies/default.asp http://www.brilliantdigital.com/music/default.asp	http://www.virttools.com/demos/index.asp	http://www.wildtangent.com/candy/

11. Fazit der Gegenüberstellung

11.1 Objektvisualisierung

Im Kampf um das bessere Web3D-Format zur Objektdarstellung liefern sich *Cycore* und *Viewpoint* ein Kopf an Kopf Rennen. Wenn die Darstellungsqualität im Vordergrund stehen soll, würde ich **VET** auf Grund der Streambarkeit, dem projizierten Schatten und der Integrierbarkeit direkt in die Webseite den Vorzug gewähren. Wenn allerdings komplexere Interaktionsmöglichkeiten gewünscht sind, eigene Erweiterungen benötigt werden, Maya-Modelle verwendet werden oder das *Partikelsystem* zum Einsatz kommen soll, siegt **Cult3D**.

Interzart's **Open-SPX** lohnt sich für Firmen vor allem beim Erwerb eines 3D-Scanners von *Dimension3D-Systems*. So können schnell selber 3D-Modelle von den eigenen Produkten generiert und veröffentlicht werden, ohne dass laufende Kosten entstehen. Erweitern lässt sich die Funktionalität obendrein. Wer schon eine 3ds max - Lizenz besitzt, hat mit **OpenSPX** als einzigem Format im Test die Möglichkeit, seine photorealistischen Modelle auch für kommerzielle Projekte ohne anfallende Lizenzkosten im Web einzusetzen.

Kaon's **Hyperspace** ist eine kostenlose Alternative für nicht-realistische Modelle. Wenn man sich photorealistische Modelle von *Kaon* direkt generieren lässt, kommt man preisgünstiger zum Ziel als bei *Viewpoint*, muss dafür aber auch eine weniger beeindruckende Darstellung in Kauf nehmen.

11.2 Charakteranimation

Für den *Java*-Tüftler ist **Shout3D** eine tolle Sache. Er kann die Engine nach Belieben in eigenen Applikationen verwenden und die Lizenzgebühren sind niedrig. Außerdem ist der 3ds max - Exporter erstaunlich leistungsfähig und auch perfekt für Charaktere zu verwenden. Für Nicht-Programmierer bietet sich diese Lösung aber höchstens zur Objektvisualisierung an.

Pulse3D ist nicht zu Unrecht das verbreitetste Format für webbasierte Charakteranimation. Eine tolle Sache ist, dass so ein komplexes Tool wie der Pulse Creator kostenlos erhältlich ist und so in Ruhe getestet werden kann, bevor man sich für diese Lösung entscheidet. Wenn der Maya-Exporter erhältlich ist, wird sich der Status von Pulse sicher noch weiter festigen können. Für Spiele und komplexere Applikationen sind andere Lösungen jedoch empfehlenswerter, weil Pulse z.B. keine Joysticks unterstützt.

B3D ist wohl die einfachste Möglichkeit, eigene 3D-Filme ohne viel Entwicklungsarbeit direkt aus 3ds max ins Netz zu bringen. Für Spiele und Applikationen ist es dagegen wegen fehlender Erweiterungsmöglichkeiten ungeeignet.

Die Entscheidung zwischen **Virtools** und **WildTangent** fällt wohl je nachdem ob man lieber visuell oder im Sourcecode arbeitet. Unerfahreneren Programmierern wird der Einstieg durch die BehaviorBlocks von Virtools erleichtert. WildTangent hat den Vorteil, dass man zunächst ohne anfallende Kosten entwickeln und erst mit dem fertigen Produkt auf Distributionssuche gehen kann. Bei Virtools muss man leider die Katze im Sack kaufen. Für nichtkommerzielle Spieleprojekte ist WildTangent ein sehr mächtiges Tool, mit dem es auch Einsteigern ermöglicht wird, schnell zu professionellen Ergebnissen zu kommen.

12.2 MPEG 4

Während die älteren Standards der 1988 gegründeten *Motion Pictures Expert Group*, **MPEG 1** und das für DVDs verwendete **MPEG 2**, ‚bloß‘ Videocodierungsverfahren standardisierten, ist **MPEG 4** eher als interaktiver Multimediasstandard zu betrachten: Neben den Möglichkeiten, auch bei sehr geringen Datenraten den qualitativen Transfer von Videodaten zu ermöglichen, dient er zur interaktiven Verbindung von WWW, animierten Grafiken, Videodaten und 3D-Objekten. Die audiovisuellen Inhalte sind als Medienobjekte vorhanden, eine separat übertragene Szenenbeschreibung regelt die baumähnlichen Beziehungen dieser zueinander.

BIFS (Binary Format for Scenes), das Format zur Beschreibung der 3D-Elemente in MPEG 4, ist zwar an VRML angelehnt, jedoch kein ASCII- sondern ein binäres Format und daher kleiner. Da MPEG 4 gerade auch bei drahtlosen Übertragungen zum Einsatz kommen soll, war es nötig, das Format sehr robust zu gestalten. Durch die Angabe von **Quality of Service** - Informationen ist MPEG 4 skalierbar, die Urheberrechte werden geschützt und ein Verfahren zur Synchronität sorgt dafür, dass auch immer der richtige Ton an der richtigen Stelle des Videobildes abgespielt wird. Dieses wird möglich, da jedes Media Object über einen eigenen Timecode verfügt. Die Möglichkeiten zur Interaktivität umfassen sowohl die lokale Manipulation der MPEG 4 - Szene als auch den Datenaustausch mit dem Server über einen Rückkanal.

Schon die erste Version von MPEG 4, die Ende 1998 erschien, bot die Möglichkeit, zusätzlich zu einem Audiostream Informationen in einem internationalen System über auftretende Phänomene und Parameter zur Gesichtsanimation zu übermitteln. Damit diese Informationen auch verwendet werden können, ist genau festgelegt, wie ein computergeneriertes Gesicht zu definieren ist.

Die im Jahr 2000 erschienene Version 2 wurde um ähnliche Möglichkeiten für die Körperanimation (Body Animation) erweitert. Man spricht von **Advanced BIFS**. Die Texturkompression kann nun mittels des *Wavelet*-Verfahrens erfolgen, dieses liefert sehr viel bessere Ergebnisse als *JPEG*.

Leider lassen sich noch keine Anwendungsbeispiele dazu im Web finden, auch wenn gerade in Verbindung mit dem neuen Mobilfunkstandard *UMTS* die Anwendungsmöglichkeiten auf der Hand liegen. Wie wäre es z.B. wenn mir ein kleiner Charakter meine frisch eingetroffene Kurzmitteilung auf dem Handydisplay vorlesen könnte? Oder wenn Kopf und Stimme vielleicht noch auf den Absender abgestimmt wären?

MPEG 4 ist sehr umfassend und wird daher zunächst nur teilweise realisiert. So findet sich z.B. ein Teil der Algorithmen zur Videokompression in Microsofts ASF (Advanced Streaming Format) wieder. Ob es aber die immerhin schon 3 Jahre alte 3D-Technologie BIFS noch schaffen wird, bei der momentanen Konkurrenz auf diesem Gebiet aus ihrem Schattendasein heraus zu treten, bleibt abzuwarten. [dp 01b]

13. Implementierungsvorgang: Learning Tai-Chi with E-sha

13.1 Anforderungsanalyse und Technologieauswahl

Da die Firma *ID-Media* in erster Linie durch ihre Highend-3D-Avatare bekannt geworden ist erhält sie häufig Anfragen für Webavatare. Bisher konnte man jedoch bis auf vorgerenderte Flash-Sequenzen oder animated GIFs nach dem Daumenkinoprinzip keine Erfahrungen in diesem Bereich sammeln. Aus diesem Grund beschloss die Abteilung Research & Development in der Zeit zwischen Aufträgen ein Beispielprojekt zu Demonstrationszwecken und um Erfahrungen zu sammeln zu realisieren. Da sich diese Aufgabe mit meinem Diplomthema deckte konnte ich die Entwicklungsarbeit übernehmen.

Die Projektplanung begann im Oktober 2000. Anforderungen an das zu realisierende Projekt waren:

- Demonstration verschiedener Interaktionsmöglichkeiten
- Sinnvoller Einsatz von Motion Capture
- Lippensynchrone Gesichtsanimation
- Simulierte Nutzerkommunikation, da Avatarintelligenz beim professionellen Einsatz durch ein separates System gewährleistet werden würde
- Low-Polygon-Character, um auch ohne 3D-Beschleuniger ablauffähig zu sein
- Einsatz der Präsentation nicht im Web, sondern nur zur Vorführung bei der Kundenwerbung
- Integration des 3D-Inhalts in eine Webseitenumgebung
- Charakter soll Nutzer auf Webseiten verweisen können

Sehr schnell fiel die Entscheidung, Pulse3D zu verwenden. Entscheidungsgründe dafür waren:

- MotionCapture-Daten konnten verwendet werden
- Optionale Hardwarebeschleunigung
- gute dynamische Darstellungsqualität durch Lightmaps
- keine Anschaffungskosten

- Lizenzgebühren fallen nur bei der Veröffentlichung im Netz an
- bestehende Anwendungen mit Pulse3D zur webbasierten Charakteranimation machten den solidesten Eindruck
- Kommunikation zwischen Webinterface und 3D-Inhalt über *JavaScript* möglich

Alternative Technologien wurden aus verschiedenen Gründen abgelehnt:

- **Fake3D**-Lösungen unterstützen keine Änderung der Perspektive
- **Cult3D**, **VET**, **Open-SPX** und **Hyper-space** eignen sich nicht für animierte Charaktere
- **Shout3D** besaß in der Version 1.0 noch nicht den heutigen Funktionsumfang und wurde wegen der fehlenden Hardwarebeschleunigung abgelehnt.
- **B3D** bot damals noch keine Möglichkeiten zur Integration in die Webseite, ist nicht erweiterbar und läßt sich noch nicht über *JavaScript* ansprechen
- **Virtools** waren zum damaligen Zeitpunkt noch als Nemo in der Versenkung verschwunden
- **WildTangent** durfte damals zwar schon ausprobiert, aber noch nicht lizenziert werden
- **VRML** wurde auf Grund der nicht bekannten Exportmöglichkeiten für Charakteranimationen und der schlechteren Darstellungsqualität verworfen

Weitere Wünsche von Kunden, die auch Pulse3D nicht erfüllen konnte und auf deren Realisierung verzichtet werden mußte, waren:

- Automatische Lippenanimation für austauschbare Audiofiles
- Individuelle Anpassbarkeit des Avatars durch den Benutzer

13.2 Konzeptionsphase

13.2.1 Tai-Chi E-sha

Man hatte sich zwar darauf geeinigt, ein Demoprojekt zu erstellen, hatte aber noch kein brauchbares Konzept dafür gefunden. Für mich war jedoch klar, dass ein inhaltlicher Aufhänger gefunden werden musste, um das ganze nicht zu einer bloßen Technologiedemonstration verkommen zu lassen und den nicht vorhandenen Inhalt durch Banalitäten wie ‚Hallo, ich bin E-sha und wohne auf Webseiten. Am liebsten höre ich HipHop. Welche Musik hörst du gerne?‘ ersetzen zu müssen.

Meine Idee eines Tai-Chi Kurses wurde von den Kollegen akzeptiert:

Sie passte zur Halbasiatin E-sha, gab mir die Möglichkeit, die Dialoge mit Informationen über diesen alten Meditationssport zu füllen und audiovisuell einen asiatisch angehauchten Gesamteindruck der Seite anzustreben.

Die Verwendung von MotionCapture machte Sinn, da die Erstellung anmutiger Tai-Chi Bewegungen auch erfahrenen Animatoren sehr viel Zeit abverlangen würde.

Verschiedene Interaktionsmöglichkeiten wurden demonstriert, indem neben der Bedienung über die Buttons des Interfaces die dritte Tai-Chi Sequenz durch einen Klick auf das im Performance-Modus in der Szene befindliche, rotierende Periskopschwert gestartet werden kann. Zusätzlich sollte die Dialogsimulation per Mausklick auf E-shas Kopf ausgelöst werden.

Da es sich auch in diesen Dialogen um Tai-Chi dreht, machte es Sinn, dass E-sha zum Abschluss eines Gesprächs Browserfenster mit Seiten, denen man weiterführende Informationen entnehmen kann, öffnet.

13.2.2 Freie Kamera oder nicht?

Lange diskutiert wurde, ob eine Änderung der Ansicht durch den Betrachter sinnvoll sei. Meiner Meinung nach war dieses zwingend nötig, schon alleine um den Vorteil von Echtzeit-3D zu nutzen. Außerdem reizte mich die Vorstellung, die selbe Sequenz bei jedem erneuten Abspielen aus einer neuen Perspektive betrachten zu können. Die Kollegen von *ID-TV* hatten weniger Vertrauen in die Kameraführung des Benutzers. Ihnen war es wichtig, den vermittelten Eindruck möglichst genau planen zu können.

Es galt also, einen möglichst guten Kompromiss zu finden zwischen einer verstellbaren Kamera und einer Garantie, dass E-sha sich immer innerhalb des sichtbaren Bereichs aufhält.

13.3 Drehbuch, MoCap & Audioaufnahmen

13.3.1 Motion-Capture für Tai-Chi

Bernd Bartel, der im Rahmen des Hochschulsports der *Fachhochschule für Technik und Wirtschaft* (FHTW) einen Tai-Chi-Kurs anbietet, erklärte sich bereit nach Babelsberg zu kommen und sich von mir erfassen (**capturen**) zu lassen. Eigentlich wollte ich, dass die Bewegungsdaten für E-sha von einer Frau stammen, um durch weiblich anmutende Bewegungen ihre Attraktivität zu unterstreichen. Der erfahrene MotionCapture-Artist Emiliano Nanfaro und ein genaues Analysieren und Vergleichen männlicher und weiblicher Tai-Chi Performer konnten meine Bedenken jedoch zerstreuen: Da bei diesem Sport die Bewegungen ohnehin im angespannten Zustand und nach festen Schemen erfolgen macht das Geschlecht des Performers kaum einen Unterschied. Aus seiner langjährigen Erfahrung wußte Emiliano Nanfaro jedoch, dass die Gesprächssequenzen



Bernd Bartel während des Capture-Vorgangs

sehr viel überzeugender wirken würden, wenn sie von einem weiblichen Performer gespielt werden. Aus diesem Grund setzte ich Bernd nur für die Tai-Chi Sequenzen ein.

Bei ID-TV kommt ein optisches Motion Capture System der Firma *Motion Analysis*¹ zum Einsatz: der Akteur muß sich in eine Art Neoprenanzug zwängen, bei dem an allen Gelenken reflektierende Styroporkugeln, sogenannte **Marker**, angebracht sind. Er agiert nun in der Mitte von 10 kreisförmig angeordneten Kameras. Das Zentrum ist durch ein Kreuz gekennzeichnet. Diese speziellen Kameras können bis zu 240 Signale pro Sekunde aufzeichnen. Bevor man mit dem Capturen der Bewegungsdaten beginnen kann, ist es nötig, das System zu initialisieren. Dazu hat der Akteur in die **T-Pose** zu gehen (ausgestreckte Arme) und der Operator definiert in **Eva**, der mitgelieferten Software zum Tracken der Marker, einmal die hierarchische Zusammengehörigkeit der einzelnen Marker. Da ja die Abstände zwischen diesen konstant bleiben, kann der Computer die Marker weitgehend automatisch verfolgen. Bei Unklarheiten muss der Operator korrigierend eingreifen und versuchen, die Hierarchie wieder herzustellen. Um die genaue Position eines Markers bestimmen zu können, benötigt man zwei Kameraansichten. Die 10 Kameras sind nötig, da die Marker in manchen Situationen vom Körper des Akteurs verdeckt werden. Natürlich waren für die dritte Tai-Chi Sequenz, in der ein Periskopschwert verwendet werden sollte, auch an diesem Marker anzubringen.

13.3.2 Datengewinnung für die Dialogsequenzen

Für die übrigen Sequenzen war es sinnvoll, zunächst mit dem Ton anzufangen, da er zur Orientierung für den Performer während des Capturens verwendet werden sollte. Nachdem ich mir die Texte ausgedacht hatte, fand ich zum Glück mit **Sasha Pereras** aus London die perfekte Stimme für E-sha.

¹ <http://www.motionanalysis.com/>

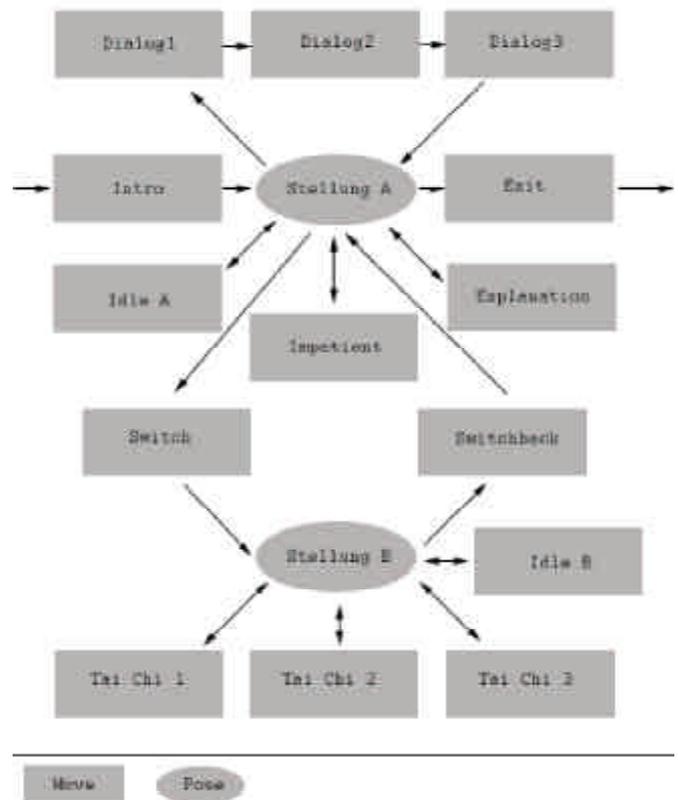
Als Performerin für die Dialogszenen konnte ich **Gloria Buddrig**, eine der Grafikerinnen von *I-D Media*, gewinnen. Als Referenz während des Capturens wurden ihr die Audiofiles vorgespielt. Außerdem hatte sie Ausdrücke der Texte, zu denen ich Regieanweisungen notiert hatte, erhalten. Bei manchen Sequenzen war mir besonders wichtig, dass Bewegungen synchron zu bestimmten Wörtern erfolgen sollten. Da ich während des Capturens mit vor Ort war und zu entscheiden hatte, wann eine Bewegung zu meiner Zufriedenheit erfolgt war, konnte ich meine Vorstellung auch selber erläutern.



Gloria Buddrig in der T-Pose zur Initialisierung

Professionelle Tips zum Capturen für interaktive Anwendungen fand ich in [gam 1].

Zur Veranschaulichung der benötigten Bewegungen habe ich dieses Schema generiert:



In jedem Fall soll E-shas Auftritt mit dem *Intro* beginnen, das sie in die **Stellung A** führt. Alle weiteren von diesem Punkt aus möglichen Animationen müssen also aus dieser Stellung erfolgen. Wenn keine Nutzeraktion erfolgt soll sie zunächst ab und zu unruhig hin und her wippen (*Idle A*) bevor sie dann irgendwann zu quengeln beginnt (*Impatient*). Durch die Switch-Bewegung streckt E-sha ihre Arme aus und befindet sich nun in **Stellung B**, die ja auch schon Bernd als Ausgangspose für seine Tai-Chi-Bewegungen verwendet hatte. Diese Gemeinsamkeit war nötig, um später einen sauberen Übergang zwischen Dialog- und Tai-Chi-Bewegungen erzielen zu können.

¹ <http://www.motionanalysis.com/>

13.4 Nachbearbeitung der MoCap – Daten und damit verbundene Probleme

Nach unserer Capturesession hatten wir zunächst nur die Markerpunkte, sogenannte **Trackerdaten**. EVA, das MotionAnalysis-Capture-Tool, arbeitet mit diesen Daten im **TRC** Format¹. Wenn nun die einzelnen zusammengehörenden Punkte mit Knochen verbunden werden entsteht ein Skelettformat, in unserem Fall **HTR** (Hierarchical Translation-Rotation)². Um diese Daten auf ein Biped-Skelett übertragen zu können, müssen sie zunächst in **csm** (character studio motion) oder **bvh** (Biovision) – Daten³ umgewandelt werden, die von character studio importiert werden können. Eigentlich gilt das HTR-Format als robuster, auch wenn bvh vor allem im semiprofessionellen Bereich verbreiteter ist.

Nun trat aber ein Problem auf: Das sogar durch ein Dongle geschützte MotionAnalysis-Plugin für 3ds max zur Umwandlung von HTR-Daten nach CSM schneidet die Nachkommastellen einfach ab, wodurch natürlich eine extrem unruhige Animation entsteht. Leider waren wir im Dezember noch völlig auf dem Holzweg, was die Ursache dieses Ruckeln angeht. Das MoCap-Team von *ID-TV*, Emiliano Nannfaro und Tom Holzinger, gab sich alle Mühe, die Daten in *kaydaras Filmbox*⁴, einem Tool zur Bearbeitung und Konvertierung von MoCap-Files, zu glätten, die Ergebnisse in 3ds max, mit denen ich weiterarbeiten mußte, blieben wegen des Exportfehlers dennoch dürrtig.

Schließlich beschloss ich bis Weihnachten 2000 wenigstens eine Sequenz von Hand zu korrigieren, um dem Drängen nach einer vertretbaren Vorabpräsentation nachkommen zu können. Dabei merkte ich schnell, dass dieses nicht der für das Gesamtprojekt zu verfolgende Weg sein konnte:

Der Zeitbedarf war einfach zu groß. Wenn z.B. ein Ruckeln in der Hand zu sehen war, konnte dieses ja in 3 Armgelenken ausgelöst werden. Zusätzlich erschwert wurde die Situation dadurch, dass 3ds max keine Animationskurven von Rotationen anzeigen und bearbeiten lassen kann. Zum Glück kam mir Anfang Januar eine Idee: Ich gab Emiliano, der die einzelnen Sequenzen in Filmbox ineinander blendete, ein Standard-bvh-Skelett zum Import. Auf dieses wurden dann die Bewegungsdaten übertragen (geplottet) und siehe da, die so gewonnenen bvh-Daten ließen sich problemlos in saubere csm-Daten umwandeln.

Um csm--Daten auf ein Biped zu übertragen ist es zunächst nötig, ein **Marker Name File** (init.mnm) mit der Festlegung der Namen und Anzahl der einzelnen Körperglieder zu öffnen. Danach weiß character studio welche Bewegungsdaten welchem Biped-Teil zuzuordnen sind. Beim Import hat man außerdem die Möglichkeit, die Daten reduzieren oder **Footsteps** aus ihnen extrahieren zu lassen. Wenn der Einsatz des Projektes im Web angestrebt worden wäre hätte es auf jeden Fall Sinn gemacht, diese Datenreduzierung vorzunehmen. Da ich jedoch in der Experimentierphase mit dieser Funktion noch die ohnehin durch den Umwandlungsfehler verfälschten Daten verwendete, die reduziert noch unruhiger wurden, gab ich meine in diese Richtung gehenden Bemühungen auf.

Ab und zu war es noch nötig, Anfang oder Ende der Animationen manuell zu korrigieren, damit die Positionen übereinstimmten, ohne dass ein Schlittern der Figur entstand. Dazu habe ich einfach die letzten Keyframes gelöscht und dann noch ein, zwei Schritte manuell animiert.

¹ <http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/TRC.html>

² <http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/HTR.html>

³ <http://www.biovision.com/bvh.html>

⁴ <http://www.kaydara.com>

13.5 Binding mit character studio

Da ich nun verwendbare Daten vorliegen hatte konnte ich zum nächsten Schritt kommen: es galt, das Biped mit der zugehörigen Mesh zu verbinden, um diese durch die animierte Skelettstruktur verformen zu lassen. Dieser Prozess wird gemeinhin als **Binding** bezeichnet.

character studio lässt zwischen **rigid** und **deformable** Binding auswählen, wobei die Länge der einzelne Gelenke (**Links**) und ein festzulegender Umgebungsbereich (**Envelope**) bestimmen, welcher Vertex an welches Gelenk gebunden werden soll.

Das deformable Binding liefert nicht nur die schöneren Ergebnisse sondern geht auch schneller. Sehnen (**tendons**) und Muskelverformungen (**bulges**) sind nur im Zusammenspiel mit diesem Binding-Typ zu verwenden. Leider unterstützt der Pulse Exporter das deformable binding (noch) nicht, daher muss das weitaus unkomfortablere rigid binding zum Einsatz kommen.

Beim rigid binding muss man explizit sagen, dass die Darstellung im Viewport auch gemäß diesem Verfahren erfolgen soll, andernfalls erwartet den Animator eine unerfreuliche Überraschung wenn er seine in 3ds max noch wunderschöne Bewegung das erste mal im exportierten Pulsefile zu sehen bekommt. Zum Glück kann auch beim rigid binding der Einfluss mehrerer Links gewichtet werden, dazu sind die zu bearbeitenden Vertices zunächst zu **sperrern**. Den gesperrten Zustand erkennt man daran, dass die einzelnen Punkte von kleinen Quadraten umgeben werden. Nun kann man die Gewichtung der verfügbaren Links manuell vornehmen, sollte aber darauf achten dass die bearbeiteten Vertices vor dem Abspeichern gesperrt sind. 3ds max nimmt ansonsten beim nächsten Öffnen die Zuordnung erneut über Envelopes und Links vor, die aufwendige manuelle Gewichtung geht verloren! Auch das ansonsten unterstützte Abspeichern der neuen Zuordnung über das **phy**-Format ist leider nicht

wie sonst möglich. Somit müssen manuelle Änderungen jedes Mal, wenn ein neues Binding nötig wird, erneut vorgenommen werden.

Das Binding erfolgt im sogenannten **Figure Mode**. In diesen kann das Biped jederzeit wieder zurückversetzt werden. Die Figurmesh ist nun mit einem **Physique**-Modifizier zu versehen und an das Becken des im Figure-Mode befindlichen Bipedes, das den Anfang der Charakter-Hierarchie bildet, zu binden. Dazu wird die Funktion **Attach to Node** benutzt.

Eine sehr praktische Funktion ist ‚**Copy Posture**‘, das die aktuelle Skelettpose sichert. Später kann diese mittels ‚**Paste Posture**‘ wieder auf das Skelett kopiert werden. Davon habe ich Gebrauch gemacht, um Berndts beziehungsweise Glorias Bewegungsdaten zu synchronisieren: vor dem Einladen der MoCap-Daten habe ich einfach die Initpose aus dem Figure Mode gesichert, so konnte ich etwaige Verzerrungen ausgleichen.

Häufig stimmen die Maße des gecaptureten Performers nicht mit denen des Modells überein. Für diesen Fall gibt es zum Glück Tools, die nur einmal zu verwenden sind und nur zur Verfügung stehen, wenn ungefilterte MoCap-Daten geladen wurden: Durch das Wechseln in den **Talent Figure Mode** kann man das Biped nach belieben skalieren. Wenn man durch ein zweites Anklicken des Buttons in den normalen Modus zurückwechselt, werden die Veränderungen gegenüber den ursprünglichen Daten berechnet. Diese lassen sich nun in einem Figur Structure File (**.fig**) sichern. Zusätzlich sollte man noch **Adjust Talent Pose** ausführen, damit die veränderten Maße auch die Markerdaten direkt beeinflussen. Auch diese Ergebnisse lassen sich speichern und zwar in einem Pose Adjustment File (**.cal**). Zukünftig kann man direkt beim Importvorgang diese Dateien verwenden, um immer die gleiche Änderung des Bipedes gegenüber den Maßen des Performers zu erzielen.

13.6 Generierung der einzelnen Phoneme für Lip-Sync

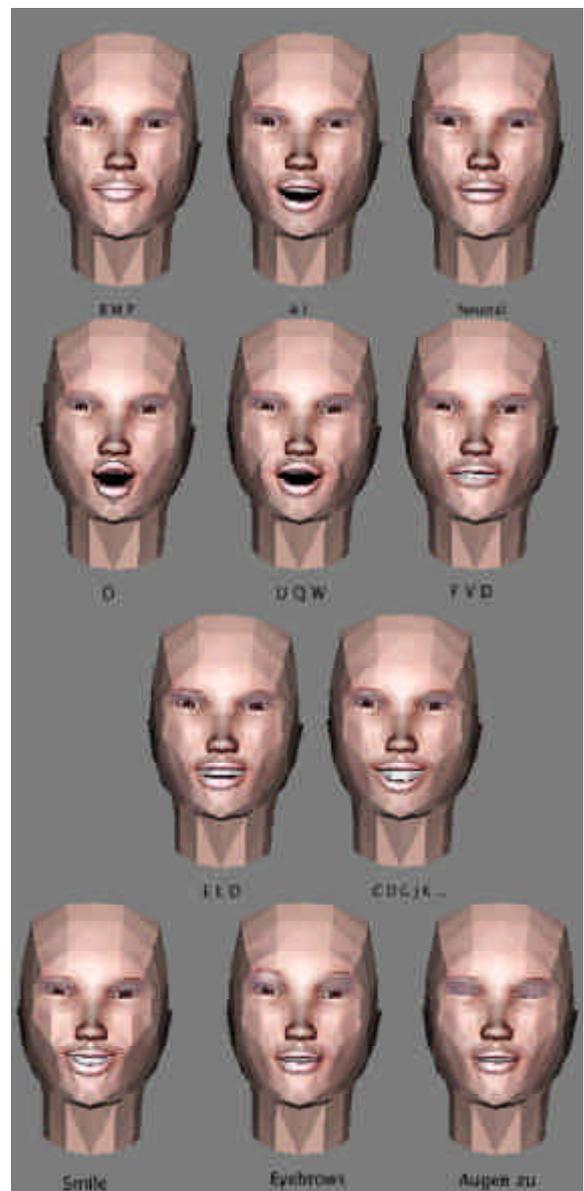
Um möglichst realistische und synchrone Lippenanimation in den Dialogen zu gewährleisten, ist es nötig, verschiedene Extremposen des Mundes zu modellieren. Da in Pulse3D immer nur eine Animation pro Charakter zur gleichen Zeit abgespielt werden kann, war es ohnehin nötig, den Kopf als separates Objekt zu verwenden. Ich habe diesen, der ein separates Element war, also mit der **detach**-Funktion vom Restkörper getrennt und mehrmals dupliziert. Nach **Preston Blair**, einem der Staranimatoren bei *Disney*, aus dessen Feder z.B. die animierten Nilpferde bei *Fantasia* stammen, unterscheidet man grob 9 Phoneme¹:

A, I
E
O
U
C, D, G, K, N, R, S, Th, Y, Z
F, V, D (weich)
L, D (hart)
B, M, P
W, Q

Allerdings sollte man beachten, dass es sich bei diesen Vorgaben um die englische Aussprache handelt. Das ‚Eh‘ wird also z.B. wie unser ‚Ilieh‘ ausgesprochen. Da E-sha´s Texte ohnehin in Englisch gehalten wurden, musste ich aber nicht umdenken. Da sich der Mund für L vom E eigentlich nur durch die Zunge unterscheidet und dieses Detail beim verwendeten Low-Poly-Modell ausgespart worden war, konnte ich E & LD zu einem Phonem zusammenfassen, auch U und WQ unterscheiden sich nicht so stark dass eine Trennung nötig gewesen wäre. Statt dessen beschloß ich noch ein paar emotionale Ausdrücke in die Liste aufzunehmen: ein Lächeln, hochgezogene Augenbrauen und geschlossene Augenlider.

Auf den ursprünglichen Kopf ist nun ein Morphmodifikator anzuwenden, wobei die ver-

schiedenen Extremformen als Morphtargets zu definieren sind. Wichtig ist, dass die Anzahl der Vertices zwischen den einzelnen Köpfen immer gleich zu bleiben hat. Mit den Reglern des Morph-Modifikators ist es nun möglich, die verschiedenen Extremposen unterschiedlich stark zu gewichten. So kann z.B. ein fünfzigprozentiges F mit einem 40 % Lächeln und halb gesenkten Augenlidern kombiniert werden. All das bietet allerdings auch der Pulse Creator. Es reicht also, wenn in 3ds max vor dem Export die Morphtargets als solche eingerichtet worden sind.

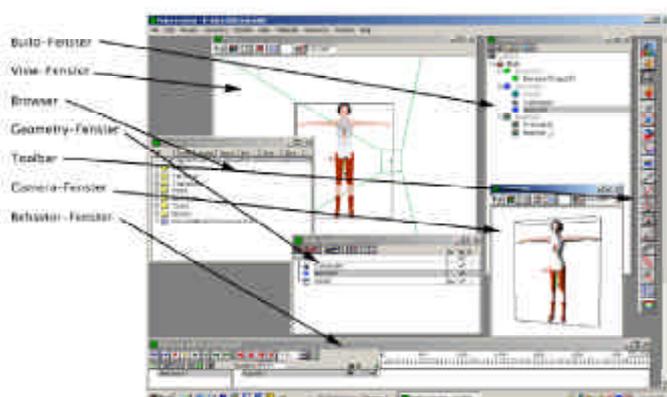


¹ http://www.geocities.com/~gcmartin/mouth_shapes.html

13.7 Pulse Creator

Im folgenden Abschnitt möchte ich ein wenig auf die Eigenheiten des Pulse Creators eingehen, um dann später bei der Beschreibung meiner Lösungen für das Beispielprojekt nur noch auf die Funktionalität eingehen zu müssen.

Der Neueinsteiger in das Pulse3D-Authoring-tool **Pulse Creator** sollte ein wenig Zeit einplanen, um sich an die Multi-Window-Strategie zu gewöhnen. Unmittelbar nach dem Öffnen einer exportierten Datei erscheint zunächst nichts außer dem **Build**-Fenster, das sämtliche enthaltenen Komponenten auflistet. Erst wenn man diese explizit anklickt öffnet sich das **Geometry**-Fenster mit einer zugehörigen Ansicht (**View**-Fenster), in der nur die Objekte dargestellt werden, die auch schon im Geometry-Fenster aufgelistet werden. Um sämtliche Komponenten sehen zu können, ist es daher zunächst nötig, diese im Build-Fenster anzuklicken.

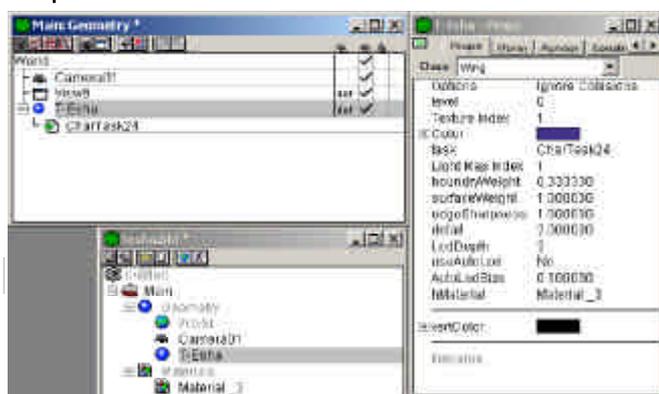


Die Vielzahl der Fenster des Pulse-Creators verwirrt den Neueinsteiger

Desweiteren ist es wichtig zu bedenken, dass für die einzelnen Kameras bzw. Ansichten separate Darstellungseinstellungen verwendet werden. Ausschlaggebend für die exportierten Einstellungen ist aber nur der Blick durch die **Kamera**. Um diesen zu testen sind zwei Klicks nötig: einen auf den Eintrag im Build-Fenster, um die Kamera im Geometry-Fenster anzuzeigen und noch einen auf diesen neuen Eintrag im Geometry-Fenster, um durch die Kamera zu gucken.

Im **Buildfenster** wird immer nur die Spitze komplexerer Hierarchien angezeigt. Zugriff auf tiefer liegende Elemente erhält man nur im Geometry-fenster. Im Behavior-Fenster findet man sämtliche Animationen des angewählten Objektes und hat die Möglichkeit, neue zu erstellen. Außer klassischen Keyframes können auch Skript-Keys Bestandteil eines Behaviors sein.

Geometrieobjekte unterscheiden sich in Pulse3D durch ihre Klassenzugehörigkeit. Diese kann durch eine Drop-Down Liste innerhalb der Objekteigenschaften geändert werden. Objekte der **Poly**klasse können z.B. ins glattere **Wing**-Format umkonvertiert werden, Objekte, die der **Charakter**klasse angehören, können Behaviors (Animationen) abspielen und sind per Maus drehbar. In der Praxis ist es häufig nötig, Verhaltensweisen mehrerer Klassen zu kombinieren. Dieses erfolgt durch die baumartige Anordnung von Skripts, die einfach aus dem Browser in das Geometry-Fenster oder das Taskfeld der Properties zu ziehen sind.



Auch ein Wing-Objekt kann durch Zuweisung eines CharTask Charakter-Eigenschaften besitzen

Ein Material kann auch aus mehreren Lightmaps bestehen. Den damit versehenen Objekten kann dann über das Attribut **Light Map Index** explizit die jeweils zu verwendende Lightmap des Materials zugeordnet werden. Durch das Ändern dieses Attributes ist es möglich, dynamisch zwischen zwei verschiedenen Darstellungsarten dieses Objekts zu wechseln.

13.8 Export von 3ds max nach Pulse

Im folgenden Abschnitt möchte ich kurz beschreiben, wie Daten aus 3ds max in den Pulse Creator gelangen. Ich denke, dass daraus ersichtlich wird, wie ich dieses bei meinem Tai-Chi Projekt vorgenommen habe.

Zum Export der 3ds max - Animationen nach Pulse3D gibt es den kostenlosen **Pulse Producer**, den man nach der Installation unter den Utilities finden kann. In der mittlerweile vorliegenden Version 4.4 kommt dieser endlich auch mit europäischen Systemeinstellungen klar. Bei früheren Versionen gab es Probleme, da beim Export die Nachkommastellen gemäß der deutschen Norm mit Kommas und nicht durch einen Punkt abgetrennt wurden: Die so entstandenen Files konnte Pulse nicht verarbeiten. Bei *ID-TV* musste ich als Pulse-Operator deshalb mit amerikanischen Systemeinstellungen arbeiten.

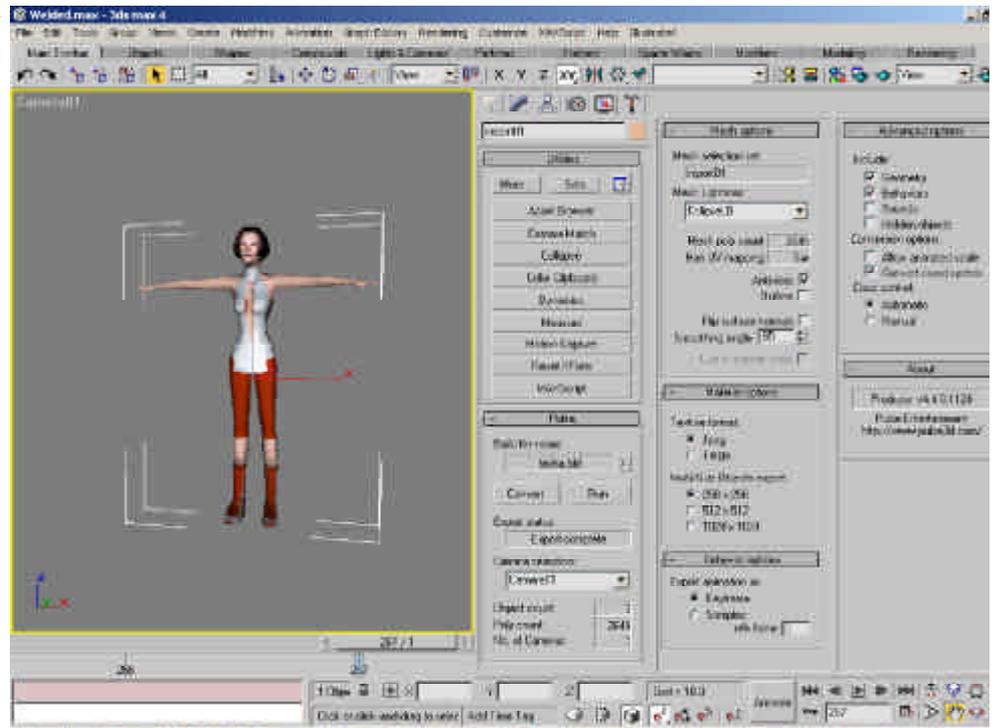
Wichtig ist, dass vor dem Export eine Kamera in der Szene vorhanden sein muss. Wenn mehrere existieren ist vor dem Export die zu verwendende aus einer Drop-Down Liste auszuwählen. Außerdem ist pro Objekt zur Darstellung eine passende Lightmap anzugeben. Der Benutzer kann jetzt schon kennzeichnen, welche Objekte später von Pulse geglättet oder mit einer Kontur versehen werden sollen. Diese Eigenschaften und die verwendete Lightmap lassen sich aber natürlich auch später noch ändern.

Um Animationen nach Pulse zu exportieren sind im Trackview von 3ds max Kommentare (**Notes**) am Anfang und gewünschten Ende zu erstellen, die ähnlich der HTML-Syntax aufgebaut sein müssen:

„Rotate“ z.B. leitet des Behavior Rotate ein, „/Rotate“ schließt es ab. So ist es möglich, mehrere Pulse-Animationen aus einer 3ds-Szene zu generieren und diese auch schon zu benennen.

Animationen können entweder als **Keyframes** oder als **Samples** exportiert werden. Bei Keyframes werden nur die für Veränderungen entscheidenden Animationsdaten angegeben, bei Samples alle Gelenkpositionen in festzulegenden zeitlichen Abständen. Letzteres ist zu verwenden wenn nicht-lineare Animationsverläufe verwendet wurden und die Animation 1:1 von Pulse übernommen werden soll, da die Interpolation zwischen den einzelnen Keyframes beim Export immer linear erfolgt. Erst im Creator kann der lineare Animationsverlauf in Kurven (**F-Curves**) umgewandelt werden.

Wenn ein Objekt mit einem **Morph-Modifikator** versehen ist, stehen die **Targets** automatisch auch in Pulse zur Verfügung und können wie z.B. in meinem Fall zur Synchronisation von Lippenbewegungen verwendet werden.



Die Optionen des Pulse Producers

13.9 Lipsync in Pulse

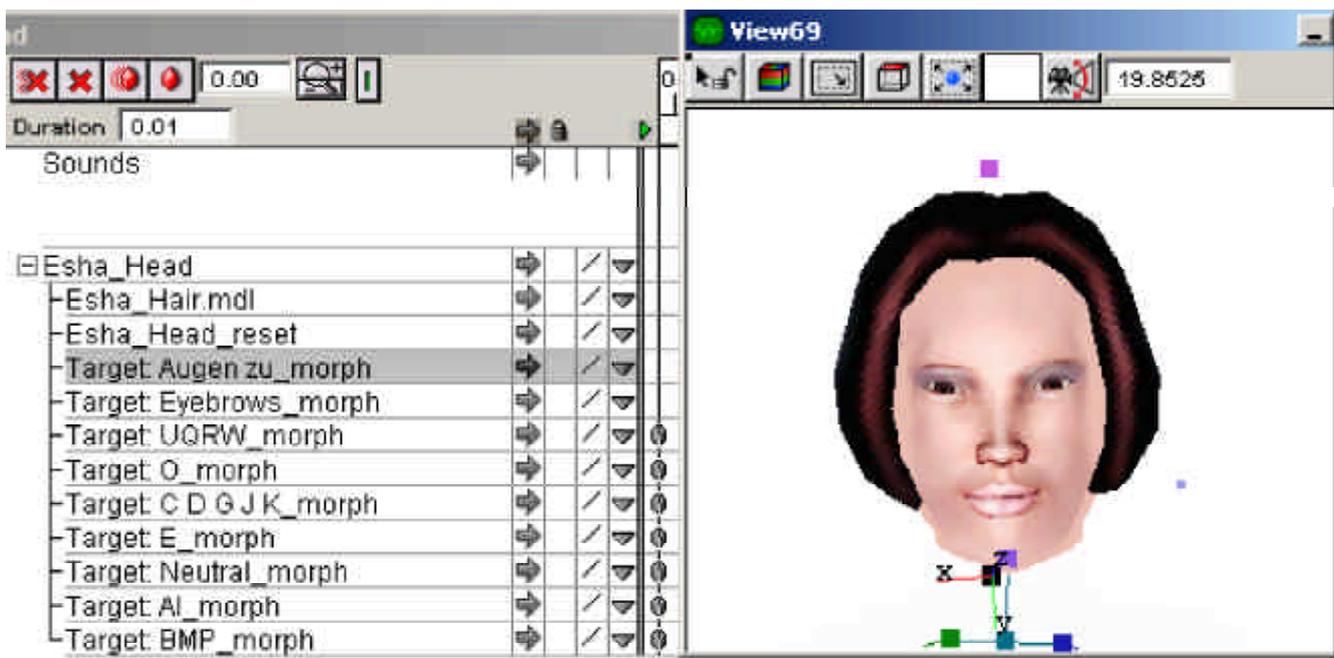
Pulse besitzt zum Glück die Fähigkeit, lip-pensynchrone Animationen automatisch zu generieren. Dieses erfordert jedoch ein wenig Vorarbeit durch den Animator. E-sha's Kopf ist in 3ds mit einem Morphmodifikator versehen worden, der die verschiedenen Phoneme als Zielobjekte verwendet. Im Behavior-Editor des Kopfes kann man sämtliche Morphs unterhalb des Objektes ausmachen. Nach dem Anwählen eines Morphs ist es nun möglich, diesen unterschiedlich stark wirken zu lassen:

Durch Drücken der **1** setzt man den Morph schrittweise bis auf 0 % zurück, und mit der **2** kann man ihn bis auf 100% anheben. Die Taste **3** kann benutzt werden, um ihn auch unter 0% ins negative zu steigern, und die **4** ermöglicht die Intensivierung über 100% hinaus.

Nun sind zunächst eigene Behaviors, die nur einen Keyframe lang sind, für die einzelnen Phoneme zu erstellen.

Wenn für sämtliche Phoneme Behaviors vorhanden sind, kann man damit anfangen Pulse auf die Stimme der Figur hin zu ‚trainieren‘. Dazu muß man zunächst ein paar Sprachsequenzen manuell animieren. Man erstellt ein neues Behavior und ergänzt dieses um einen **Sound-** sowie einen **Compound-Track**. Der Compound-Track dient zum Sammeln mehrerer Behaviors. In den Soundtrack lädt man das zu verwendende Soundfile und in den Compound – Track kann man nun die einzelnen Phoneme-Behaviors passend zum Soundfile hineinziehen. Durch Abspielen der Sequenz kann man jederzeit testen wie gelungen die Synchronisation bisher ist.

Sobald man mit dieser manuellen Lippenanimation zufrieden ist, kann man den Creator mittels ‚**Learn Track**‘ auffordern, diese auszuwerten. Er berechnet nun aus den Lauten und den verwendeten Phonemen, welcher Klang welchem Behavior zuzuordnen ist. Dieses muß nun so lange wiederholt werden, bis der Creator genügend Informationen sammeln konnte.

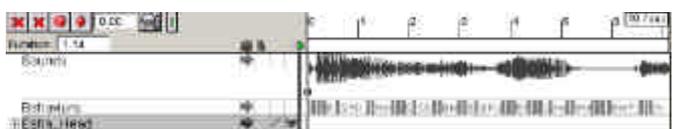


E-shas Kopf mit den zugehörigen Morph-Targets

Um zu prüfen, wie gut der Pulse Creator die automatische Lippensynchronisation schon beherrscht, ist wieder ein neues Behavior zu erstellen, nur dass jetzt der Compound-Track nicht mehr manuell, sondern über ein simples **Do LipSync** gefüllt werden kann. Falls das Ergebnis noch zu wünschen übrig lässt, müssen weitere Gesprächsanimationen von Hand generiert werden. Über Parameter wie ‚Min Energy‘ oder ‚Smoothing‘ kann man das automatische LipSync weiter feintunen.

Emotionale Ausdrücke, wie ein Hochziehen der Augenbrauen oder ein Senken der Augenlider, habe ich in E-shas Fall manuell hinzugefügt. Es wäre aber auch möglich, verschiedene Phoneme für verschiedene Stimmungen zu verwenden: Diese wären dann durch ein Pre- oder Suffix zu kennzeichnen, so dass das Auto-LipSync zwischen Angry_A und Friendly_A unterscheiden kann. Vor der automatischen Lippensynchronisation müßte ich dann je nach E-shas aktueller Stimmung das entsprechende Prefix angeben.

Während der Entwicklungsarbeit kam es häufig vor, dass Keyframes aus dem Compound-Track nicht korrekt wiedergegeben wurden. Dieses Problem ließ sich umgehen, indem der Compound-Track durch **flatten** aufgelöst und in die jeweiligen Animationsspuren eingerechnet wurde.



Gesprächsbehavior mit Phonemen im Compound-Track



... und mittels ‚flatten‘ in einzelne Keyframes umgewandelt

13.10 Problemstellungen in Pulse3D und Lösungswege

Im folgenden Abschnitt wird skizziert, mit welchen Problemen ich während der Entwicklungsphase konfrontiert wurde und wie es mir gelungen ist, diese zufriedenstellend zu lösen.

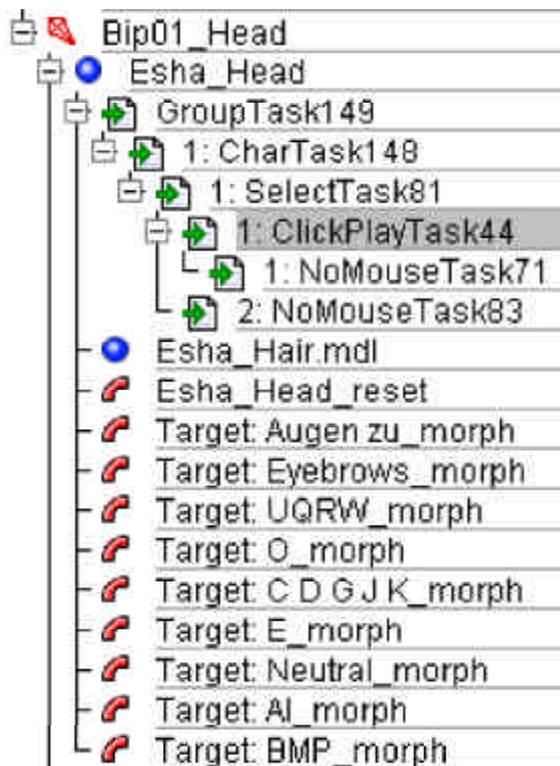
13.10.1 E-sha langsam einblenden

Zu Anfang der Präsentation sollte E-sha sich langsam materialisieren, nachdem der Betrachter auf den Intro-Button gedrückt hat, und dann ihren Einführungstext aufsagen. Die Sichtbarkeit von Objekten lässt sich über einen BeVis - Key ändern. Dieses geschieht abrupt, E-sha sollte aber langsam erscheinen. Um die Transparenz von Objekten zu animieren, bietet es sich an den Alphakanal, der die Transparenz der Textur regelt, über die Zeit zu verändern. Dieses wird von Pulse durch den **BeColor-Key** unterstützt. Da Körper und Kopf zwei separate Objekte sind, die synchron sichtbar werden sollen, starte ich gleich zu Anfang des Intro-Behaviors des Körpers über einen **BeCommand-Key** ein Behavior des Kopfes, das auch diesen langsam von unsichtbar nach sichtbar wechseln läßt.

13.10.2 Interaktion durch Kopfklicken

Durch einen Klick auf E-shas Kopf sollte die Dialogsimulation gestartet werden. Dieses darf aber auch nur genau dann möglich sein, wenn sie sich in der Ausgangsposition befindet. Damit ein Objekt Behaviors abspielen kann, muss es über die Eigenschaften der Character- Klasse verfügen, also entweder selber vom Typ Character sein oder einen Char-Task zugeordnet haben. Das Problem dabei ist, dass Character-Objekte immer auch per Mouse rotierbar sind. Der Kopf sollte aber normalerweise nicht drehbar sein, andererseits aber in der Ausgangsposition interaktive Funktionalitäten bieten. Um dieses Problem zu lösen, habe ich einen **Select-Task** verwendet. Als hierarchische Kinder hat dieser sowohl einen **NoMouseTask**, um die

Mousedrehung zu unterbinden, als auch einen **ClickPlayTask**, um die Dialogsimulation abspielen zu können. Der Wechsel zwischen diesen beiden Varianten muß nun immer, wenn E-sha in die Ausgangsposition oder aus dieser wechselt, über einen **BeCommandKey** mit dem Befehl **select Obj=Esha_Head select=0;** bzw. **select=1;** erfolgen. Damit der Kopf während des Klickens nicht verdreht werden kann, ist unterhalb des ClickPlayTasks wiederum ein NoMouseTask anzubringen.



Die Hierarchie von E-shas Kopf, darunter die exportierten Morphotargets

13.10.3 Interaktion mit dem Schwert

Sobald E-sha in den Performance-Modus wechselt, sollte ein rotierendes Schwert in der Szene auftauchen. Durch einen Klick darauf sollte die dritte Tai Chi Sequenz gestartet werden, in der E-sha das Schwert in der Hand hält. Zur Realisierung habe ich mit zwei Schwertern gearbeitet: Eins hält E-sha immer in der Hand, standardmäßig ist es jedoch versteckt. Das andere befindet sich ebenfalls unsichtbar in der Szene. Ihm ist ein **GroupTask** zugewiesen, durch den es möglich wird, einen Click-

PlayTask und einen MotorTask, der dafür Sorge trägt, dass sich das Schwert pro Sekunde um 90 Grad dreht, zu kombinieren. Sobald E-sha in den Performance-Modus wechselt, wird über einen **BeCommandKey** ein Behavior des Schwerts ausgelöst, das dieses mittels eines direkt in der Hierarchie angebrachten **BeVisKeys** sichtbar macht. Äquivalent funktioniert auch das Ausblenden des Schwertes. Durch den ClickPlay-Task wird das rotierende Schwert aus- und das in E-shas Hand eingblendet. Nachdem die Übung beendet ist wird das wieder rückgängig gemacht.

Das Material des Schwertes besteht aus zwei übereinandergelegten **Lightmaps**: Eine für die diffuse, über die Textur bestimmte Farbe des Griffes, eine für die reflektierende Klinge. Um dieses zu erreichen müssen auch auf der Lightmap Reflektionen vorhanden sein.

13.10.4 Die Kameraproblematik

Wie schon zuvor beschrieben, galt es, eine Lösung zu finden, die sowohl interaktiv ist als auch die Garantie beinhaltet, dass E-sha den sichtbaren Bereich nicht verläßt. Auch Kameras unterscheiden sich in Pulse3D durch ihre Klassenzugehörigkeit.

Zunächst habe ich einen Ansatz mit einer **SmartCam** verfolgt: diese kann zwar um das Objekt, auf das sie ausgerichtet ist, rotiert werden, bewegt sich aber mit dem **Target** (Zielobjekt) mit. Da unsere Umgebung sehr einfarbig gehalten ist, fehlten jedoch die Referenzen, um die Kamerabewegungen richtig zu interpretieren. Das Ergebnis sah es aus als wenn E-sha auf Grund von völlig kaputten MoCap-Daten wackeln würde.

Eine geeignete Lösung fand ich, indem ich eine **DefaultCamera** mit einem Group-Task versah. Dieser bestand aus einem **CharTask**, um es der Kamera zu ermöglichen, Behaviors abzuspielen, und einem **TrackCamTask**.

Dieser bietet die gleiche Funktionalität wie die SmartCam, bewegt sich aber nicht mit, sondern bleibt nur auf das Zielobjekt ausgerichtet. Wenn das Zielobjekt allerdings z.B. E-shas Kopf wäre, würde die Kamera sich durch die Ausrichtung trotzdem mitdrehen. Deshalb habe ich einfach einen unsichtbaren Würfel erstellt, der als Zielobjekt dient.

Im FaceCam - Behavior der Kamera wird dieser nun über einen **BeAttachObj**-Key an den Biped-Kopf angehängt, wobei er sich auch an dessen Position bewegt. Damit sich die Bewegungsruckler nicht auf die Kamera auswirken, wird er aber gleich darauf wieder an die 3D-Szene angehängt, dieses mal allerdings ohne seine Position zu ändern. Äquivalent arbeitet das BodyCam - Behavior. Durch den TrackCamTask besitzt unsere Kamera einen Zoom-Parameter. Verändert wird dieser wieder über BeCommandKeys, die einen Aufruf der Form **Set obj=Camera01 zoom=-1;** enthalten. Um einen weichen Zoom zu erhalten, habe ich eine kleine Animation aus mehreren BeCommandKeys für das Zoomverhalten erstellt:

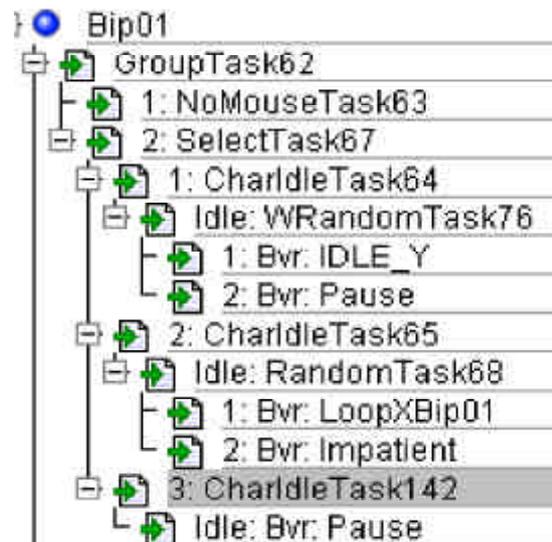
Die Kamera beginnt langsam zu zoomen, beschleunigt dann und bremst zum Ende der Animation wieder ab. Zoombehaviors gibt es natürlich sowohl für das Heran- als auch das Herauszoomen.

13.10.5 Wechselnde Idle-Behaviors

Um überzeugender zu wirken, sollte E-sha, wenn sie gerade nicht damit beschäftigt ist, Behaviors auszuführen, nicht nur steif herumstehen, sondern sich auch mal beim Benutzer darüber beklagen, dass sie nichts zu tun hat. Während E-sha noch unsichtbar ist, darf das allerdings noch nicht passieren.

Zur Lösung des Problems habe ich eine etwas komplexere Hierarchie anlegen müssen. Unmittelbar an das Biped, die Spitze der Hierarchie, habe ich einen GroupTask gehängt, der sich aus einem NoMouseTask, damit E-sha nicht drehbar ist, und einem SelectTask zusammensetzt. Mit diesem SelectTask kann man nun zwischen drei

verschiedenen **CharIdleTasks** auswählen:



Die Taskhierarchie des Bipedes

Einer ist für die Ausgangsstellung, einer für die Performancstellung und einer für die Situationen, in denen keine Idle-Behaviors auftreten sollen, zu verwenden. Der erste und der zweite CharIdleTask besitzen als Kinder je 2 **RandomTasks**, von denen nun nach dem Zufallsprinzip entweder der eine oder der andere abgespielt werden kann, wenn ein untätiger Zustand erreicht ist. Die verwendeten Behaviors enden mit einem Aufruf des Pause-Behaviors über einen BeCommandKey der Form **play obj=Bip01 behave=Pause;**, damit E-sha nicht permanent am quengeln und rumhampeln ist. Dieses Pause-Behavior ist auch das einzige Kind des dritten CharIdleTasks.

Beim Start der Seite ist der dritte CharIdleTask aktiv. Wenn E-sha das erste Mal den Ruhezustand erreicht, wird der erste über einen BeCommandKey selektiert, zum zweiten wird gewechselt, sobald E-sha in den Performancemodus gelangt. Damit die Idle-Behaviors nicht unmittelbar an die vorherigen anschließen, enden ihre Vorgänger auch mit dem Aufruf des Pause-Behaviors. Erst wenn dieses abgespielt wurde, wird wieder ein Idle-Behavior ausgewählt.

13.11 Entwurf des Webinterfaces

13.11.1 Interfacegestaltung

Die Interfacegestaltung wurde mir von **Gloria Buddrik** in Photoshop abgenommen, die es in kürzester Zeit schaffte, mein Konzept perfekt umzusetzen.

Um einzelne Bereiche des Screens dynamisch verändern und mit Funktionalität hinterlegen zu können, ohne mit Ebenen arbeiten zu müssen, habe ich mir die neue **Slice**-Funktion von *Adobe Photoshop 6* zu Nutze gemacht:

Der Screen wird zur Darstellung im Browser in Streifen zerschnitten, die vor dem Betrachter verborgen in einer unsichtbaren Tabelle angeordnet werden. In Adobe's **Image Ready** können für die einzelnen Slices sehr einfach verschiedene Zustände erstellt werden. In meinem Fall war das der Standard-Zustand ‚rollover‘ und der von mir benötigte ‚disabled‘. Das Pulse-Fenster aus der standardmäßig erstellten default.html-Datei habe ich einfach in *Macromedia Dreamweaver 3* geöffnet und via copy-paste in die frei gelassene Mitte der Tabelle der Image Ready Datei kopiert.

13.11.2 Multimediakomponenten

Die für den Dialog benutzten Eingabe-vortäuschungen sind Flash-Animationen auf separaten Ebenen, die auf Mouseclick ihre eigene Ebene und damit sich selbst verstecken und ein Signal zum Abspielen des nächsten Dialogs an den Pulse-Content schicken.

Die in der Präsentation verwendeten Loops stammen von einer Seite¹ mit Copyright-freien Loop und sind größtenteils von **Asim Khan** produziert worden.

13.11.3 Probleme mit Netscape

Wenn der Pulse-Inhalt auf einer separaten Ebene liegt, hat der Netscape Communicator leider Probleme, Events an das integrierte Pulse-Objekt zu schicken. Auch transparente Flash-Hintergründe werden vom Communicator nicht unterstützt. Dieser Schönheitsfehler würde sich beheben lassen, indem ein Screenshot des Interfaces in den Flash-Animationen als Hintergrund verwendet wird, um einen transparenten Schriftzug vorzutäuschen. Dieses habe ich mir aber für eventuelle spätere Versionen aufgehoben.



E-sha und ihr Interface

¹ <http://www.flashkit.com/loops/index.shtml>

13.12 Pulse <-> Browser Kommunikation

Um die Buttons des Interfaces mit der gewünschten Funktionalität zu versehen, war ein wenig *JavaScript*-Programmierung nötig. Auf Grund der Syntax-Highlights habe ich dieses mit **UltraEdit32**¹ gemacht. Den Sourcecode dazu kann man in der verwendeten Datei esha.html einsehen.

Aus Pulse-Behaviors werden über einen **BeMessage**-Key Signale an den umgebenden Browser geschickt. Diese müssen nun innerhalb des Codes der Seite abgefangen und ausgewertet werden. Auf bestimmte Signale erfolgt der Aufruf von Funktionen, z.B. zum deaktivieren eines Buttons. Wenn ein Button deaktiviert wird, ersetze ich ihn durch seinen vorgeladenen Disabled-State. Zusätzlich setze ich eine zugehörige Kontrollvariable auf 1. Durch prüfen auf 0 wird verhindert, dass Pulse-Animationen im Disabled-State ausgelöst werden können.

Um dafür zu sorgen, dass man sich nicht zu nah an E-sha heran oder zu weit weg bewegen kann, habe ich einen Zoomcounter implementiert: Wenn ein bestimmter Mindest- oder Maximalwert erreicht ist wird der jeweilige Button deaktiviert.

Je nachdem, ob sich E-sha im Performance-Modus befindet oder nicht, besitzt der Exit-Button verschiedene Funktionalitäten. Auch hier verwende ich eine Prüfvariable. Ähnlich gehe ich auch vor, um immer zwischen zwei verschiedenen Dialogen zu wechseln.

13.13 Projektfazit

Leider bin ich immer noch ein wenig unzufrieden mit der aktuellen Version der Tai-Chi E-sha:

Zwar gibt es an der Funktionalität kaum noch etwas zu bemängeln, allerdings um so mehr am verwendeten Modell: an diesem sind unbedingt noch einige Vertices zusammenschweißen, um weichere Übergänge zwischen den Polygonen zu erzielen. Außerdem sollte man noch einige Edges um 180 Grad drehen (**turn edges**), um unsauberes Shading zu vermeiden. Auch am Binding ist noch einiges nachzubessern. E-sha verdreht z.B. die rechte Hand völlig unnatürlich, wenn sie in den Performance-Modus geht. Leider war es häufig erst **Dejan Momcilovic**, der Leiter der 3D-Abteilung, der sich die Zeit nahm sich intensiver in das Projekt hineinzudenken und mich auf Schwachpunkte aufmerksam zu machen. Vieles wurde mir aber auch erst durch die intensivere Beschäftigung mit Low-Poly Modelling in 3ds max bewußt, was ja bei diesem Projekt eigentlich nicht meine Aufgabe sein sollte.

Außerdem ärgert mich, dass ich irgendwann aufgehört habe, das Projekt für das Web zu optimieren. Nachdem mir gesagt wurde, dass das Projekt nur von CD gezeigt werden sollte, habe ich natürlich keine Rücksicht mehr auf die Filesize genommen. Außerdem war meine Zeit bis dahin ohnehin schon äußerst knapp bemessen, da ich zu viel mit den defekten MoCap-Daten verschwendet hatte.

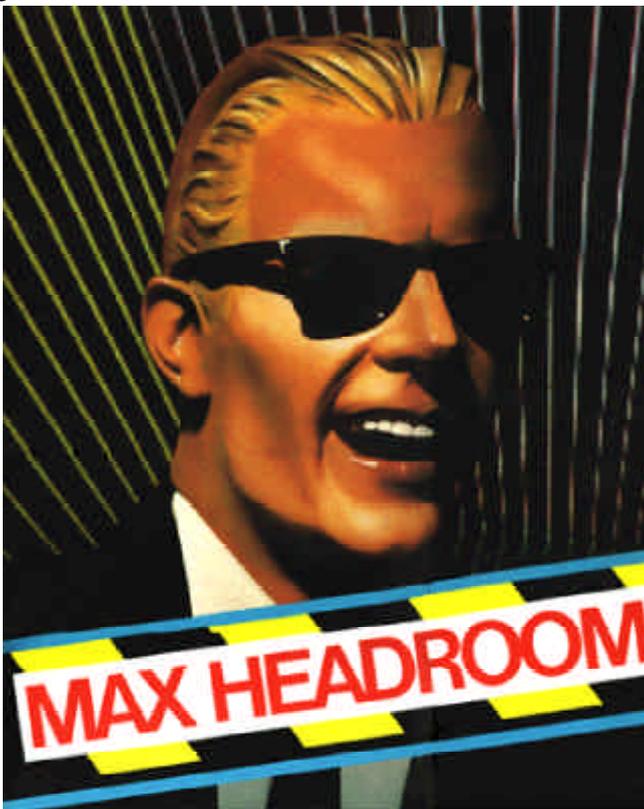
Pulse kann Audiofiles aus dem MP3-Format nur verwenden, wenn diese gestreamt werden sollen. Da dieses nun nicht vorgesehen war, habe ich sämtliche Audiofiles als unkomprimierte wavs integriert, was natürlich erheblich zur Vergrößerung des Projektes beitrug. Auch die MoCap-Files hätten noch optimiert werden

¹ <http://www.ultraedit.com>

müssen, damit diese streambar gewesen wären. Da ich aber erst sehr spät mit ruhigen Bewegungsdaten arbeiten konnte, mußte auch dieses hinten anstehen.

Für den Einsatz von Streaming wäre es nötig gewesen, das Projekt in einzelne Packages zu zerlegen. Dadurch wäre auch ersichtlicher geworden, für welche Komponenten welche Datenmenge anfällt. Dennoch ist es bei einer Präsentation von CD natürlich sinnvoll mit den bestmöglichen Daten zu arbeiten: ein Qualitätsverlust wäre bei der Arbeit mit reduzierten Bewegungsdaten und streambaren MP3s unumgänglich gewesen.

Da die Rechte für E-sha bei *I-D Media* liegen, ist es mir nicht möglich das Projekt selber im Netz zu veröffentlichen, auch wenn ich, solange der Pulse-Banner im Inhalt erhalten bleibt, keine Lizenzgebühren bezahlen müsste. Deshalb hoffe ich, dass ich die Möglichkeit erhalte eine überarbeitete Version für den Webeinsatz, unabhängig von dieser Diplomarbeit, anzufertigen.



14. Vorstellung der bekanntesten Avatare

Die Bezeichnung ‚Avatar‘ entstammt ursprünglich dem Hinduismus. Unter der Vorstellung der Wiedergeburt diente er zur Bezeichnung der vergänglichen menschlichen Hülle. Dieser Begriff wurde 1980 vom US-Militär für Repräsentationen der Benutzer in Simulationsspielen verwendet und gelangte so in die Computerwelt.

Populär wurden Avatare seit 1992 maßgeblich durch **Neil Stephenson's** Cyberpunk-Meilenstein ‚**Snow Crash**‘. Ein Grossteil der Handlung des Buches spielt im ‚Metaverse‘, einer Art 3D-Welt, in der es als größtes Statussymbol betrachtet wird in einem Avatar unterwegs zu sein, der dem natürlichen Erscheinungsbild des Trägers entspricht. Weniger versierte User hingegen müssen sich mit einer ‚Brandy‘ von der Stange zufrieden geben oder teure Avatardesigner mit der Gestaltung ihrer 3D-Repräsentanten beauftragen. Wie sehr dieses Buch auch die VRML-Community der 90er geprägt hat macht z.B. der Name *Blaxxun*, eines der führenden VRML-Anbieter, deutlich: ‚Black Sun‘ heißt in *Snow Crash* das Clubhaus der Hacker im Metaverse.

Die Geschichte der virtuellen Stars begann 1984 mit **Max Headroom**, einer englischen Fernsehserie. Das stotternde Computerwesen wurde damals allerdings noch von einem Schauspieler verkörpert. Der Plastiklook entstand durch eine Gummimaske und per Nachbearbeitung am Amiga.

1996 gelang es *CoreDesign UK* mit dem weiblichen Hauptdarsteller **Lara Croft** ihres **Tombräider**-Spiels, einen ansprechenden 3D-Charakter in Echtzeit darzustellen. Die Erfolgsserie beläuft sich mittlerweile auf fünf Teile, die sich kaum noch voneinander unterscheiden.



Der erste virtuelle Popstar, **Kyoko Date**, entstand ebenfalls 1996, natürlich in Japan, bei der Firma *Horipro* im Rahmen des DK-96 Projekts, wobei DK für DigitalKids steht. 10 Artist waren alleine für das Design von Kyokos Gesicht zuständig. Insgesamt bestand der Charakter aus 40.000 Polygonen. Kyoko Date, zu der auch gleich noch ein erfundenes Persönlichkeitsprofil mitgeliefert wurde, hat mittlerweile mehrere Charthits gelandet, ihren eigenen Fanclub und ist Star zahlreicher Werbespots.

Inzwischen soll sich Bill Gates sämtliche Vermarktungsrechte an dem digitalen Traumgirl gesichert haben [ava 2].

Lustigerweise erschien zeitgleich William Gibsons Buch ‚Idoru‘, das sich ebenfalls mit virtuellen Idolen beschäftigte: der Cybervisionär, der mit seinem Buch ‚Neuromancer‘ schon 1976 viele Technologietrends vorausgeahnt hatte, war von der Realität eingeholt worden.

Für den Sommer 2001 ist ein (leider) real gedrehter Kinofilm angekündigt. England zeigte weitaus weniger Berührungängste zum Entertainmentbereich als z.B. Deutschland: Im Jahr 2000 wurde Lara Croft als Aushängeschild der britischen Wirtschaft verwendet [ava 1].

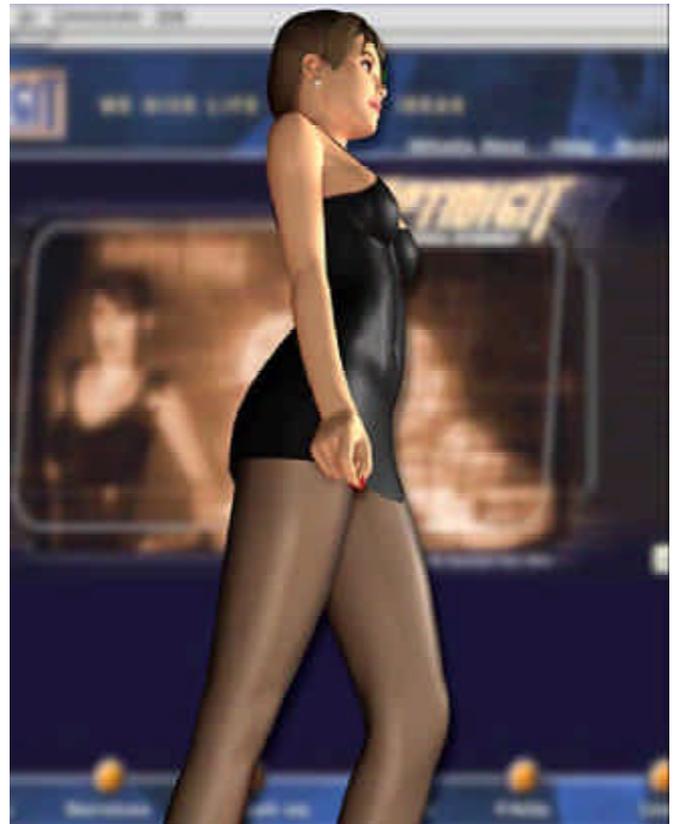
Über die Beziehung der zumeist männlichen Spieler zu ihrem Avatar wurden seitenlange psychologische Abhandlungen verfaßt. Ich selber erinnere mich schmunzelnd an das Zitat eines Opfers der Tombraider-Sucht: „Wenn ich irgendwo abstürze drücke ich schnell auf die ‚Spielstand laden‘-Taste. Ich kann es einfach nicht ertragen, Laras Knochen brechen zu hören.“





Nun war es aber an der Zeit für ein männliches Computerwesen: Am 1. September 1997 wurde E-Cyas erstmals von der ID-Media AG als Promoter des Chatforums Cycosmos, in dem die User, ähnlich wie in alten Rollenspielen, durch 2D-Sprites repräsentiert werden, eingesetzt. Am 3. Januar 2000 kam seine von Alex Christensen (U96) produzierte Single auf den Markt und mittlerweile hat er es sogar zu einem Eintrag im Duden gebracht. Der schon lange versprochene Chat mit dem Meister des Cycosmos lässt aber leider immer noch auf sich warten [ava 3].

Die Vorzüge digitaler Modelle hat auch *Elite*, die weltweit führende Modellagentur, die unter anderem Claudia Schiffer oder Naomi Campbell hervorgebracht hat, entdeckt. Ihre Tochter *Illusion 2K*¹ veranstaltet ähnlich wie die Mutteragentur jeden September einen Modellwettbewerb. Teilnahmeberechtigt sind allerdings nur digitale Kandidatinnen. Die für die Auswahl maßgeblichen Kriterien wurden von Elite-Gründer John Casablancas persönlich festgelegt. Man plant, einmal pro Jahr ein neues Modell zu veröffentlichen. Die erste ihrer Art stammt vom schwedischen 3D-Artist Steven Stahlberg und hört auf den Namen Webbie Tookay [ava 4].



¹ <http://www.illusion2k.com/>

15. Eigene Überlegungen zu Einsatzmöglichkeiten

Echtzeitcharaktere im Internet sind auf jeden Fall eine nette Spielerei, wo jedoch liegen die Einsatzmöglichkeiten?

Eine Idee von mir für den Einsatz von Webcharakteren wäre es, beispielsweise einen Kursus für das Hörgeschädigten-Alphabet auf einer Webseite anzubieten. In einzelnen Lektionen könnten die einzelnen Buchstaben von einem Avatar erläutert und dargestellt werden. Um ein Zertifikat zu erhalten, muß der Benutzer dann einen flüssigen Text aus der Zeichensprache übersetzen.

Die Richtung weiterer Anwendungsfälle habe ich mit E-sha's Tai-Chi Kurs schon aufgezeigt: Denkbar sind komplexere Web-Applikationen beispielsweise für Tanzkurse, Aerobic, Ballsport, Origami oder des Zusammenbauen von Ikea-Möbeln. Der Avatar verliert auch nicht die Geduld, wenn er die gleiche Übung zum zehnten Mal aufführen muß. Durch das freie Wählen der Perspektive kann der Benutzer jedes Mal einem anderen Aspekt der Bewegung seine Aufmerksamkeit schenken.

Immer mehr Computerspiele setzen heutzutage auf 3D-Grafik. Da in diesem Fall die Modelle der Hauptakteure ohnehin vorhanden sind stellt es nur einen geringen Mehraufwand dar, diese auf der Homepage für das neue Produkt werben zu lassen. Eindrucksvoll bewiesen wird das z.B. auf der Homepage zu ‚America McGee's Alice‘¹.

Denkbar wäre auch ein Online-Spiel, das sich am inoffiziellen Berliner Nationalsport ‚Hacky Sack‘ orientiert: Die Aufgabe des Spielers ist es einen kleinen Ball durch Tritte in der Luft zu behalten. Als weiteren Anreiz wären eine Highscore-Liste im Internet sowie eine Multi-Player Funktion wünschenswert. Die Finanze-

rung könnte über eingeblendete Werbebanner erfolgen.

Die sogenannten Webisodes stellen ohnehin schon einen der Hauptanwendungsfälle dar. Wie schon zuvor erwähnt, würde ich jedoch probieren, diese auf vielen unterschiedlichen kommerziellen Portalen einzusetzen.

Wie wäre es wenn man Kleidungsstücke nicht nur drehen und von allen Seiten anschauen könnte, sondern diese sogar, ähnlich einem Blick in den Spiegel, am eigenen Leibe begutachten könnte? Wenn die visuelle Qualität der X3D-Browser in einigen Jahren den heutigen Stand von Pulse & Co. erreicht hat, wird es vielleicht möglich sein durch konsequente Einhaltung des H-Anim Standards die einmal z.B. mittels eines Bodyscanners generierte Repräsentation des eigenen Körpers in Applikationen der verschiedensten Modehäuser zu laden, um bequem vom heimischen Computer aus zu testen, ob das Kleidungsstück den eigenen Typ zur Geltung bringt oder diesem eher widerspricht.

Heutzutage findet man häufig Lösungen, bei denen die Stimmengenerierung über eine Text-to-Speech-Engine erfolgt. Der Einsatz automatisch generierter Stimmen hat natürlich den Vorteil, dass der Charakter z.B. den User bei seinem Vornamen ansprechen kann. Allerdings kann ich mir nicht vorstellen, dass ein Charakter mit einer heutzutage noch viel zu mechanisch klingenden Stimme zum Sympathieträger werden kann. Viel mehr Sinn würde es jedoch machen, jeden Tag neue Textfiles von einem Sprecher aufnehmen zu lassen, die dann im Web verwendet werden können. Die Firma *planb media*² aus Köln arbeitet an einer Lösung für Pulse3D, die die dynamische Integration neuer Audiofiles in den fertigen Pulseinhalt integrieren soll.

¹<http://www.alice.ea.com/>

²<http://www.planb-media.de/>

16. Intelligenz für Avatare

Bis auf die Virtual Friends lassen alle von mir vorgestellten Lösungen für Webavatare den Benutzer nur zwischen verschiedenen vorgegebenen Wegen wählen. Auf individuelle Wünsche können sie nicht reagieren. Ein Avatar ist nur die visuelle Repräsentation des virtuellen Gesprächspartners. Wenn dieser jedoch über Intelligenz verfügt, spricht man von einem **Bot**.

16.1 Kiwilogic¹

Die Lingobots der Hamburger Firma *Kiwilogic* zeichneten bisher zum Beispiel für die Intelligenz hinter der virtuellen Präsidentschaftskandidatin der Agentur *noDNA*², **Jackie Strike**³, verantwortlich. Damit diese auch sprechen kann, verwendet man die von der *noDNA*-Tochter *X-IST*⁴ selber entwickelte Speech-Engine **Vuppet-Voice**.

Erstellt wird ein Bot mit dem **Lingobot Creator**. Der Benutzer muß die möglichen Gesprächsverläufe vorausplanen und kann angeben, auf welche erfüllten Gesprächsbedingungen hin welche Antwort erfolgen soll. Mit einer Demoversion kann jeder Interessent selber ausprobieren, einen Lingobot zum Leben zu erwecken. Lauffähig ist der Creator auf allen Windowssystemen, für die serverseitige Applikation muß Windows NT oder 2000 verwendet werden.

Zwar hat man bei *Kiwilogic* noch keine auf **Callcentern** basierende Lösungen im Angebot, jedoch bestätigte mir ein Mitarbeiter, dass die Entwicklung klar in diese Richtung gehe: erste Beispiele wären in Kürze zu erwarten.



Hinter Jackies Intelligenz steckt Kiwilogic

16.2 Artificial Life⁵

Der **Alife-WebGuide** ist eine komplett in *Java* realisierte Lösung: zur Erstellung der Bot-Intelligenz dient der **Knowledge Editor**. Zusätzlich zu der eigenen Knowledge Base kann ein Bot auch von der Intelligenz seiner Artificial-Life-Geschwister profitieren und Informationen aus einer *Oracle*- oder *JDBC*-Datenbank erhalten. Durch Profilerstellung zu den einzelnen Usern kann sich so ein Artificial Life Bot auch an Inhalte früherer Dialoge erinnern und darauf Bezug nehmen. Die Technologie zur Sprachinhaltserkennung stellt dabei ein Kernelement dar, sie wird als *Alife-SmartEngine* bezeichnet.

16.3 Novomind⁶

Als einziger Anbieter in dieser Übersicht bietet die Hamburger *Novomind AG* mit **IQ** nicht nur ein System für Bot-Intelligenz an, sondern mit **TrueTalk** auch die Möglichkeit direkt mit einem Operator zu chatten. Während die serverseitige Agentensoftware ein System der NT-Familie benötigt unterstützt man clientseitig neben den üblichen Windows-Varianten auch das MacOS.

¹ <http://www.kiwilogic.com>

² <http://www.nodna.com>

³ <http://www.jackiestrike.com>

⁴ <http://www.x-ist.de>

⁵ <http://www.artificiallife.com>

⁶ <http://www.novomind.de>

17. Zukunftsaussblicke

17.1 Intels Internet-3D-Technologien und Shockwave3D

Um 3D im Web zum Durchbruch zu verhelfen, hat der Prozessorhersteller *Intel*¹ eine Reihe von internetfähigen 3D-Technologien entwickelt. Der Gedanke hinter dieser Entwicklung ist einfach: 3D-Objekte sollen dynamisch an die vorhandene Hardware angepasst werden können. Der Artist muss sich so nicht schon beim Modelling über die beim Betrachter vorhandene Systemleistung Gedanken machen, da die erstellten Objekte automatisch auf ein für ausreichende Performance nötiges Maß herunterkaliert werden. Verwendung finden werden diese Technologien vor allem in Macromedias neuem Format **Shockwave3D**: Mit der im Mai 2001 erscheinenden Version von **Macromedias Director 8.5 Shockwave Studio** wird der betagte Director um 3D-Authoring-Funktionen erweitert werden [ct 2]. Das Shockwave-PlugIn muss dafür nur um 500 KB vergrößert werden. Damit auch der Export aus allen größeren 3D-Paketen kein Problem darstellt, haben bereits im Sommer 2000 sämtliche namhaften Hersteller ihre Unterstützung für das kommende Format zugesichert [ct 3]. Maya-Besitzer können das Export-PlugIn schon jetzt (Stand: 8.5.01) anfordern². Für 3D Studio Max3.1 hat sogar *Macromedia* selber einen Exporter entwickelt, der schon verfügbar ist.

Die 5 Technologien hinter von Intel sind³:

Multi Resolution Mesh 2 (MRM)
Subdivision surfaces (SDS)
Skeletal Charakter Animation
Non-Photorealistic Rendering
Particle systems

Durch die Verwendung von **Multi Resolution Meshes** ist es möglich, je nach Bandbreite verschieden hoch aufgelöste Varianten ein und desselben Modells bereit zu halten. Außerdem kann das Modell reduziert werden, um eine bessere Performance zu gewährleisten.

Bei Webanwendungen stößt man häufiger auf das Problem, dass die nötige Rechenpower zwar vorhanden ist, nicht aber die nötige Bandbreite. Zu diesem Zweck macht es Sinn, das Objekt zunächst nur mit dem nötigsten Detaillierungsgrad zu übertragen. Gemäß festgelegter Rundungsoperationen kann dem Objekt dann auf dem Zielrechner zu höherem Detailreichtum verholfen werden. Diese Verfahren wird als **Subdivision Surfaces** bezeichnet.

Bei **Skeletal Charakter Animation** dient eine hierarchische Skelettstruktur zur Animation einer einzelnen Charaktergeometrie. Da die Struktur ebenfalls von character studio übernommen wurde, steht einer Kompatibilität mit dem H-Anim Standard nichts im Wege.

Das **Non-Photorealistic Rendering** ist mit dem Cartoon-Rendering des vorgestellten Illustrate-PlugIn zu vergleichen. Der entscheidende Vorteil gegenüber diesem liegt jedoch in der Echtzeitfähigkeit.

Partikelsysteme stellen schon seit Jahren zur Darstellung natürlicher Effekte (Feuer, Flüssigkeit, ...) ein wichtiges Element der 3D-Animation dar. Der echtzeitfähige Ansatz von Intel ähnelt dem von Cult3D.

Die Libraries, mit denen *Intel* das Web erobern möchte, sind schon seit einer ganzen Weile über *Digimation*¹, einen Distributor für 3D-PlugIns, erhältlich. Verwendung fanden sie bisher u.a. in dem Spiel Dark Project II.

Intel profitiert also gleich mehrfach von den eigenen Entwicklungen: Einerseits fließt durch die Lizenzgebühren Geld in die Kassen, andererseits sind diese Technologien für *Intel*'s aktuelles Flaggschiff, den **Pentium 4**, optimiert, was ihnen einen Vorsprung gegenüber den Mitbewerbern verschaffen könnte.

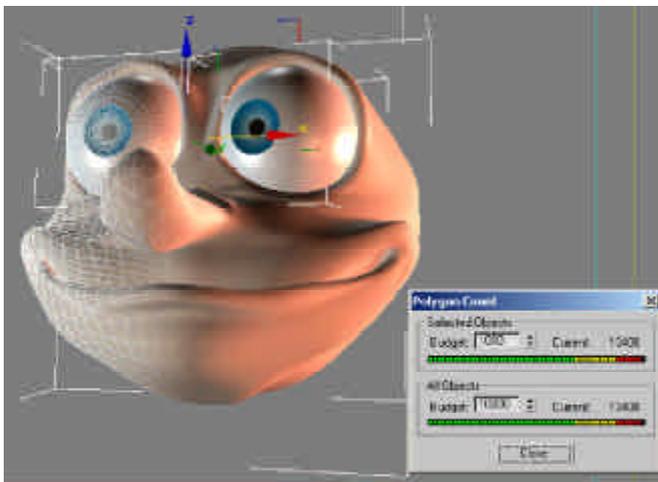
¹ <http://www.intel.com>

² http://www.aliaswavefront.com/en/Community/Special/shockwave/shockwave_m.html#

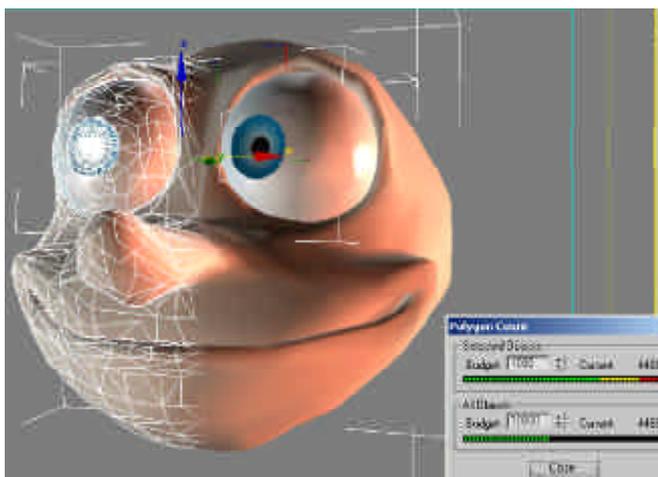
³ <http://developer.intel.com/ial/3dsoftware/>

In 3ds max 4 ist die Multi Resolution Mesh bereits integriert und konnte von mir getestet werden:

Das zu reduzierende Objekt ist zunächst mit einem MultiRes – Modifikator zu versehen. Wenn man nun mit ‚Generate‘ die Erzeugung eines MultiRes-Objektes veranlasst, kann man danach die Darstellung dynamisch reduzieren. In dem abgebildeten Beispiel habe die Auflösung der Augen und des Kopfes für das rechte Bild auf 25 Prozent reduziert. Wie die Screenshots belegen ist der optische Unterschied sehr gering.



Der Originalkopf verfügt noch über 13.408 Polygone



Mit MultiRes auf 4.489 Polygone reduziert

In erster Linie zielt Shockwave3D auf den Online-3D-Game-Bereich ab. Erste Spiele findet man von *Lego*² und *Intel* selber wie z.B. das Anglerspiel *Big Mouth Bass 3D*³.



Screenshot von Intel´s *Big Mouth Bass 3D*

Erste Anwendungen hat die israelische Firma *Oradnet*⁴ in Form von Sportpräsentationen in der Entwicklung: **TOPlay** soll Spielübertragungen in Form von 3D-Grafiken ermöglichen. Den entscheidenden Torschuss kann man aus einer frei gewählten Perspektive verfolgen, wenn man möchte auch aus den Augen eines bestimmten Spielers [ct 3]. *Intel* scheint in diesem Bereich ein sehr großes Potential zu vermuten, so strebt man mit der NBA ein ähnliches Konzept für Basketball an.



Screenshot von *Oradnet*´s *TOPlay*

¹<http://www.digimation.com>

²<http://mindstorms.lego.com/robohunter2/>

³<http://www.shockwave.com/intel/games/>

⁴<http://www.oradnet.com>

17.2 Der Geforce III-Chip

Nach jahrelangem Framerate-Wettrüsten schickt sich *nvidia* an, die 3D-Grafikwelt mit ihrem neuem

GeForce3-Chip zu revolutionieren. So mancher Hardcore-Gamer mag auf Grund erster Benchmarks noch enttäuscht reagiert haben, da die Leistung des neuen *nvidia*-Chips in klassischen **Direct3D**-Spielen wie Unreal Tournament sogar noch unter der des Vorgängermodells Geforce2 GTS liegt. Die Neuerungen verbergen sich hinter dem Begriff **nfinite FX Engine** und läuten ganz nebenbei ein neues Zeitalter von Echtzeit-3D-Grafiken ein.

Genauer betrachtet, besteht die *nfinite FX Engine* eigentlich aus zwei Neuerungen: dem **Vertex Shader** und dem **Pixel Shader**.

Mit Hilfe des Vertex Shaders hat der Programmierer nun die Möglichkeit, kleine Assemblerprogramme für die Scheitelpunkte zu entwickeln, deren Ausführung übernimmt dann der Chip. Einsetzbar ist dieses beispielsweise für **Keyframeanimationen**, da Zwischenschritte in Echtzeit berechnet werden können, **Environmental Effects** wie Nebel oder Rauch, **procedurale Deformationen** beispielsweise für wehende Fahnen, **Morphing**, Bewegungsunschärfe (**Motion Blur**) oder aufwendige Lichtberechnungen, wie sie für doppelseitige Polygone nötig sind [gf 1].

Der Pixelshader erlaubt ähnliche kleine Programme, allerdings für jeden zu zeichnenden Pixel. Durch eigene Schattierungsberechnungen kann so eine echtzeitfähige Darstellung erzielt werden, die am ehesten mit dem **Phong-Shading** zu vergleichen ist.



Der GeForce3 - Chip

Auch der Einsatz von echtem Bump-Mapping wird so in Echtzeit möglich [gf 2].

3ds max 4 unterstützt durch seine Hardwareshader Vertex und Pixel Shader von Haus aus.

Eine weitere Neuerung ist die Unterstützung von **High Definition Surfaces** (HDS). Unter HDS versteht man Flächen, die nicht wie bisher durch die Angabe von zu zeichnenden Dreiecken, sondern durch Splines oder Kontrollpunkte (CVs) beschrieben werden. Der GeForce3-Chip unterstützt die Darstellung von NPatches und sogenannten polynominale Oberflächen.

Informationen über *nvidias OpenGL* und *DirectX8* – Erweiterungen findet man im Developerbereich der *nvidia*-Homepage¹. Dort kann man sich auch den **EffectBrowser** herunterladen, mit dem man eigene Vertexshader und Beispielleffekte, wenn auch langsam, ohne GeForce3 antesten kann.



Einer der in Maya generierten Gegner aus dem kommenden GeForce3-Spiel Doom3

¹ <http://www.nvidia.com>

Anhang A: Fachbegriffserklärungen

API	Application Programmer Interface, eine Sammlung von Klassen, die für eigene Programme verwendet werden kann
Applet	Ein kleines <i>Java</i> -Programm, das in Webseiten eingebettet werden kann und ablauffähig ist, wenn auf dem Zielrechner ein <i>Java</i> -Interpreter, die sogenannte Virtual Machine, vorhanden ist.
ASE	ASCII-Format von 3ds max
Bounding Box	Grobe Quaderstruktur umgibt die Modelle zur Vereinfachung von Kollisionserkennungen
Bump Mapping	Methode, um unebene, poröse Oberflächen vorzutäuschen; wird erreicht, indem die Berechnung der zur Beleuchtungsdarstellung nötigen Oberflächennormale durch ein Graustufenbild (Bumpmap) verändert wird
Direct3D	3D- <i>API</i> aus der <i>DirectX</i> -Familie von Microsoft, ist daher auch nur auf Windows-Systemen und nicht unter NT nutzbar
DirectX	Multimedia- <i>API</i> von Microsoft für Windows-Systeme
DLL	Dynamic Link Libraries bestehen aus Softwaremodulen, die zur Laufzeit nachgeladen werden können
Flat Shading	Shadingverfahren, bei der der Fläche nur ein Farbwert zugewiesen wird
Game Exchange	ASCII-Format von Nichimen, enthält 3D-Objekte, Texturen und Animationen
Gouraud Shading	Shadingverfahren, bei dem nur die Eckpunkte berechnet und die Zwischenräume mit Verläufen versehen werden
Java	Objektorientierte, plattformunabhängige, interpretierte Programmiersprache von Sun Microsystems
Java3D	<i>Java</i> -Aufsatz für <i>OpenGL</i> , leicht aber langsam
JavaScript	Skriptsprache, die direkt in HTML-Dokumente eingebunden werden kann
Java Bean	<i>Java</i> -Komponente, die in einem visuellen Buildertool verwendet werden kann
JDBC	steht für Java DataBase Connectivity, <i>Java-API</i> für den Zugriff auf Datenbanken
JPEG	Sehr verbreitetes Grafikformat der Joint Photographics Expert Group, das zur Kompression eine diskrete Kosinustransformation verwendet
OBJ	Object Files, ursprünglich Format von Wavefront's Advanced Visualizer, unterstützt sowohl polygonale Gitternetze als auch Freiformflächen, sehr verbreitetes ASCII-Austauschformat
Oracle	Verbreitetes Datenbanksystem
OpenGL	freigegebene 3D- <i>API</i> von SGI, Implementierungen für fast alle Plattformen vorhanden
Partikelsysteme	Partikelsysteme sind ein oder mehrere Pixel, die auf Grund von simulierten physikalischen Verhältnissen und Verhaltensweisen animiert werden
Phong Shading	Shadingverfahren, bei dem die Schattierungen für alle vorhandenen Punkte berechnet werden
Playstation II	Spielkonsole von Sony

Plug-In	Ein Plug-In dient zur Erweiterung der Funktionalität des Hostprogramms, z.B. eines Browsers oder eines 3D-Tools
Siggraph	DIE jährliche Computergrafikfachmesse
Texel	Kleinste Einheit einer Textur (Pixel), kann jedoch beim Mapping auch mehrere Pixel umfassen
UMTS	Universal Mobile Telecommunication System, Neuer von NTT Dokomo / Ericcson / Nokia entwickelter Mobilfunkstandard, der Datenraten bis zu 2MBit/Sekunde ermöglicht
Wavelet	Kompressionsverfahren, das durch inhaltsabhängige Kompression sehr viel bessere Ergebnisse als JPEG liefert.
X-Box	Angekündigte Spielkonsole von Microsoft mit einem Grafikchip von nvidia
XML	Metasprache für das Definieren von Dokumenttypen

Anhang B: Literaturliste

- [3dg 1] David Duberman, Cult of Interactivity, 3D Gate, Special Issue Spring 2000, S. 49 - 52
- [3dg 2] David Duberman, MAX ´s Pulse on the Web, 3D Gate, Special Issue Spring 2000, S. 53 - 55
- [3dg 3] David Duberman, No Plug-In Required, 3D Gate, Special Issue Spring 2000, S.57 - 58
- [3dl 1] Werner Lackner, NeMo, 3D Live 4/99, S.25
- [ava 1] Florian Rötzer, Lara Croft als britisches Wirtschaftsmaskottchen , Telepolis, 2.12.1998, <http://www.heise.de/tp/deutsch/inhalt/te/1693/1.html>
- [ava 2] Dave Sieg, A virtual Star, Dave ´s Raves, 16.2.1997, <http://www.davesraves.com/Features/kyoko.html>
- [ava 3] Artur P. Schmidt, E-Cyas, Telepolis, 4.6.1998, <http://www.heise.de/tp/deutsch/inhalt/co/2367/1.html>
- [ava 4] Martin Paetsch, Schöner als Schiffer, Spiegel Netzwelt, 16.7.1999, <http://www.spiegel.de/netzwelt/netzkultur/0,1518,31649,00.html>
- [ct 1] Thomas Korn, Virtuelle Zukunft im Web, c ´t 18/2000, S.52
- [ct 2] Christoph Laue, 3D-Format für Internet Anwendungen, c ´t 17/2000, S.20
- [ct 3] Frank Fremerey, Nicht nur Spielzeug, c ´t 8/2001, S.100
- [dp 99a] Nemo - ein 3D Game Toolkit oder der neue Director für 3D?, digital production 4/99, S.188 - 189
- [dp 00a] Siggraph2000 Webanimation, digital production 3/00, S.158 - 164
- [dp 01a] Cult3D 5.2 von Cystore, digital production 01/01, S. 22
- [dp 01b] MPEG-4 Der neue Multimedia-Standard, digital production 01/01, S.54 - 59
- [e.m 1] Werner Lackner, Jenseits der Realität, e.media magazin 1/2001, S.42 - 44
- [gam 1] Melianthe Kines, Planning and Directing Motion Capture, 19.1.2000, http://www.gamasutra.com/features/20000119/kines_01.htm
- [gf 1] Bernd Haluschka, NVIDIA GeForce3 im Detail, 27.02.2001, <http://www.tecchannel.de/hardware/655/>
- [gf 2] Vince Freemann, Nvidia GeForce III Technology Preview, HardwareCentral, 23.4.2001, <http://www.hardwarecentral.com/hardwarecentral/previews/3046/1/>
- [his 1] <http://www.web3d.org/aboutus/historyspec.htm>
- [may1] Sherri Sheridan, Creating Story Concepts for 3D Animation, S.25-30, Maya 2 Charakter Animation, New Riders
- [mcw1] Dennis Sellers, Adobe to buy Metastream ´s Canoma, 29.3.2000, <http://maccentral.macworld.com/news/0003/29.canoma.shtml>
- [rad 1] <http://www.education.siggraph.org/materials/HyperGraph/radiosity/radiosity.htm>
- [rad 2] Frank G. Walters, An Introduction to Radiosity, 7.2.1995, <http://www.cs.umn.edu/~ryep/classes/cs443/radiosity/radiosity.html>
- [rp 1] Rob Polevoi, The New 3D-Artist, 14.5.2000, <http://webreference.com/3d/lesson88/>
- [rp 2] Rob Polevoi, 3D E-Commerce With MetaStream, 5.1.2000, <http://www.webreference.com/3d/lesson83/>
- [rp 3] Rob Polevoi, Photorealistic Web 3D for E-Commerce, 12.2.2001, <http://webreference.com/3d/lesson88/>
- [rp 4] Rob Polevoi, Cult3D automates interactivity, 14.2.2000 <http://webreference.com/3d/lesson86/>

- [rp 5] Rob Polevoi, More Cult3D interactivity, 28.2.2000
<http://webreference.com/3d/lesson88/>
- [rp 6] Rob Polevoi, Life after Cosmo?, 7.7.1998,
<http://webreference.com/3d/lesson44/>
- [rp 7] Rob Polevoi, VRML – Back from the brink, 15.9.1998,
<http://webreference.com/3d/lesson49/>
- [rp 8] Rob Polevoi, VRML - Platinum Speaks, 28.9. 1998
<http://webreference.com/3d/lesson50/>
- [rp 9] Rob Polevoi, New Era in Web3D, 31.8.1999
<http://webreference.com/3d/lesson74/>
- [sho1] Rob Polevoi, Interactive Web Graphics With Shout3D, Sybex
- [totb 1] Paul Steed, Low Poly Tricks, Tips & Techniques, 1998,
<http://www.loonygames.com/content/1.6/totb/>
- [totb 2] Paul Steed, The Many Faces of Kelly Moncaco, 1999,
<http://www.loonygames.com/content/1.31/totb/>