

Präsentationen in 3D

(Großer Beleg)

Name: Jan Wojdziak

**Ergebnis einer Studienarbeit an der
Technischen Universität Dresden**

Inhaltsverzeichnis

INHALTSVERZEICHNIS	2
AUFGABENSTELLUNG	4
I EINFÜHRUNG.....	5
1.1 Motivation.....	5
1.2 Gliederung der Arbeit	5
II 3D TECHNIKEN	7
1 Einführung.....	7
2 OpenGL.....	7
2.1 Das Programmiermodell.....	8
2.2 Schnittstellen zum System	8
3 Open Inventor	8
3.1 Szenendatenbank	9
4 Java3D.....	9
4.1 Szenengraph.....	9
4.2 Koordinatensystem.....	9
5 VRML	10
5.1 Geschichtlicher Abriss	10
5.2 Dateiaufbau und Szenengraph.....	10
5.3 Animation und Interaktion.....	11
5.4 Schwächen von VRML	12
6 X3D.....	13
6.1 Profile.....	13
6.2 Architektur	14
7 Vergleich von VRML und X3D	16
III VERGLEICH BESTEHENDER SOFTWARE	17
1 Software zur Präsentationserstellung.....	17
1.1 Cult3D	17
1.2 EON Studio.....	18
1.3 Macromedia Director	20
1.4 RawKee – X3D Plug-in für Alias Maya	22
2 Beurteilung bestehender Erstellungssoftware	24
3 Software zur Präsentationsdarbietung.....	25
3.1 Cult 3D Viewer	25
3.2 EON Viewer.....	26
3.3 Shockwave-Player	26
3.4 VRML/X3D-Player.....	26
4 Beurteilung bestehender Darbietungssoftware	27

IV RAWKEE - OPEN SOURCE X3D PLUGIN (IM DETAIL)	30
1.1 Maya	30
1.2 RawKee.....	30
V MAYA PROGRAMMIERUNG	34
1.1 Einführung.....	34
1.2 Aufbau des RawKee Plug-ins.....	35
1.3 X3D Hintergrund	36
1.4 X3D Text	40
1.5 X3D Switch.....	43
1.6 X3D Scalar-Interpolator	44
VI TUTORIUM	45
VII ZUSAMMENFASSUNG UND AUSBLICK	50
GLOSSAR	51
ANHANG	54
A Technische Voraussetzungen	54
B Codeentwicklung	55
C Dateiübersicht	56
D Literatur und Internetquellen	60
Ansprechpartner	63

Aufgabenstellung

Um Vorlesungen und die für diese unterstützend bereitgestellten Vorlesungsmaterialien zu verbessern, sollen in dieser Belegarbeit Werkzeuge zur Erstellung von 3D-Präsentationen analysiert und verglichen werden. Anhand eines Aufgabenkatalogs sollen bestehende Softwaresysteme untersucht und beurteilt werden. Dabei soll der Vergleich unter dem Gesichtspunkt der Verwendung im Vorlesungskontext durchgeführt werden. Sollte keines der derzeit existierenden Werkzeuge den Ansprüchen, die an dieses gestellt werden erfüllen, so ist ein neues Präsentationsinstrument zu erstellen oder ein bereits existierendes zu erweitern. Zur Veranschaulichung sollte mit diesem Softwaretool eine 3D-Präsentation erstellt werden, um die Funktionen des Werkzeugs und seiner Bedienbarkeit zu verdeutlichen. Bei der Beurteilung und Entwicklung des Präsentationswerkzeugs ist auf folgende 3 Schwerpunkte besonderes Augenmerk zu richten:

Erstellung der Präsentation

Der erste Schritt bezüglich der Präsentation einer 3D-Szene ist die Erstellung dieser. Dabei sollte dem Nutzer ein Werkzeug an die Hand gegeben werden, mit dem er nach einer angemessenen Einarbeitungszeit 3D-Welten kreieren kann. Die angestrebte Zielgruppe sind Nutzer ohne gesondertes Wissen in den Bereichen Computergrafik und Programmierung. Sie sollten die Anwendung ohne Aneignung von Hintergrundwissen nutzen können. Es ist daher eine graphische Benutzeroberfläche für den Anwender bereitzustellen. Die Entwicklungsumgebung sollte der Komplexität der Anwendung angemessen und möglichst intuitiv bedienbar sein. Mithilfe des Werkzeuges sollen 3D-Präsentationen erstellt werden können. Bausteine des Werkzeugs sollen dabei grafische Primitive oder einfache dreidimensionale Objekte im Raum sein. Des Weiteren sollte dem Entwickler die Möglichkeit gegeben werden Interaktionsschnittstellen sowie Animationen in die Szene zu integrieren. Diese Aktionen sollten während der Präsentationsdarbietung mithilfe von Tastatur beziehungsweise Maus ausführbar sein. Da die Präsentationen in universitären Vorlesungen verwendet werden sollen, ist darauf zu achten, dass bei der Erstellung einer 3D-Szene ein sequenzieller Ablauf mehrerer Teilszenen eingearbeitet werden kann. Die Präsentation sollte also nicht ausschließlich aus einer Gesamtszene, in der mit Objekten eine Interaktion ausgeführt wird bestehen, sondern, ähnlich einer Folienpräsentation verschiedene Schritte eines Zyklus oder einer Abhandlung dargestellt werden können. Festgelegte Perspektiven innerhalb des dreidimensionalen Raumes sollten durch den Nutzer definierbar sein, um die Navigation während der Präsentation zu erleichtern. Des Weiteren sollte das Werkzeug die Möglichkeit bieten, häufig verwendete Komponenten die zur Veranschaulichung von Vorlesungsthematiken dienen, zu erstellen. Zu diesen Elementen gehören beispielsweise Hinweispeile, das Einbinden von Sound sowie die Darstellung von erläuterndem Text. Durch einen begleitenden Wortlaut soll eine Verbindung zwischen 2D und 3D Szenenkomponenten für Vorlesungspräsentationen geschaffen werden. Ein Schwerpunkt sollte darauf gelegt werden, eine Präsentation in möglichst kurzer, beziehungsweise angemessener Zeit erstellen zu können.

Speicherung der Präsentation

Nach dem Erstellungsprozess ist die Sicherung der 3D-Szene wesentlich. Zum einen bietet dies die Möglichkeit der Separation von Entwurfs- und Präsentationswerkzeug und zum anderen einen Vorteil bezüglich der Portierung und Wiederverwendbarkeit. Zum Speichern der Präsentation sollte ein freies Dateiformat verwendet werden. Es besteht jedoch ebenso die Möglichkeit ein eigenes Dateiformat zu erarbeiten. Es sollte bei der Entwicklung darauf geachtet werden nur eine Datei pro Präsentation beim Speichervorgang zu erzeugen.

Wiedergabe der Präsentation

Zur Präsentation der 3D-Szene ist eine Abspielsoftware bereitzustellen. Die Anwendung kann entweder in Form eines Plug-in für ein bestehendes Präsentationswerkzeug oder als eigenständige Abspielsoftware konzipiert sein. Auch die Integration in einen Webbrowser ist denkbar. Man sollte beim Abspielen der Präsentation die Möglichkeit haben, sich mittels Tastatur oder Maus im 3D-Raum zu bewegen und zwischen den in der Entwicklung definierten festen Standpunkten zu wechseln. Die gesamte Präsentation sollte innerhalb dieser Anwendung abgespielt und über eine möglichst einfache Bedienung interaktiv gesteuert werden können.

I Einführung

1.1 Motivation

In den letzten Jahrzehnten hat sich das Gesicht des Unterrichts in regelmäßigen Abständen grundlegend geändert. Fundamental beruhte zu jeder Zeit der Lehrbetrieb auf den didaktischen Modellen und damit auf den theoretischen Grundlagen des Lehrens und Lernens. Aufgrund lehrpsychologischer Erkenntnisse ist das Erscheinungsbild des Unterrichts in einem fortwährenden Prozess. Vom behavioristischen Ansatz über den Kognitivismus bis zum in den 90er Jahren entwickelten konstruktivistischen Ansatz wurden die Lerntheorien weiterentwickelt, um den Erfolg und somit den Erkenntnisgewinn und die Fähigkeiten des Lernenden zu maximieren. Dabei sind die unterschiedlichen didaktischen Grundsätze nicht nur im zeitlich aufeinander folgenden Verlauf, sondern ebenso kontextabhängig zu betrachten. Verschiedene Lehrinhalte fordern jeweils angepasste Methoden um das zu vermittelnde Wissen weiterzugeben. So müssen zum Beispiel Gesichtspunkte wie Wahrnehmung und Gedächtnis des Lernenden genauso wie Taxonomie und Formulierung als Gegenstand, abhängig von den Lehr- und Lernzielen, in die Überlegungen involviert werden.

Ein weiterer Aspekt, der die Entwicklung des Lehrens beeinflusst hat, ist der des technologischen Fortschritts. So haben Computer und das Internet den Unterricht an Schulen und Universitäten grundlegend verändert. Durch den konsequenten Einsatz der heute verfügbaren Techniken zur Informationsvermittlung ergeben sich neue potente Lösungen zur Visualisierung und Interaktion. Dieser Vorgang ist derzeit keineswegs abgeschlossen, sondern befindet sich in einem Stadium in dem, um die Effektivität des Lernens zu verbessern, eine Vielzahl didaktischer Varianten möglich und miteinander kombinierbar sind. Der Einsatz von Multimedia in seiner worteigensten Bedeutung ist eine zentrale Komponente. Der Einsatz verschiedener Medien verstärkt, wenn diese angemessen gehandhabt wird, den Lernerfolg. Im visuellen Bereich beschränkten sich die Hilfsmittel zur Wissensvermittlung auf den zweidimensionalen beziehungsweise pseudo-dreidimensionalen Raum. Heutzutage werden 3D-Präsentationen bereits erfolgreich bei Produktexpositionen, virtuellen Begehungen und weiteren Szenarien verwendet. Im universitären Bereich kommen jedoch derzeit hauptsächlich Tageslichtprojektoren und Folienpräsentationen via Leinwand zum Einsatz. Die Visualisierung von Lehrinhalten mittels einer 2D-Präsentation kann für manche Problembereiche ausreichend sein. Gleichwohl verlangen ausgesuchte Bereiche eine dreidimensionale Präsentation, um den Lernprozess zu optimieren. Die Veranschaulichung einer Thematik oder einer Konstruktion im Raum ermöglicht dem Lernenden eine einfachere Wissensaneignung und ein besseres Verständnis. Durch eine freie Beweglichkeit in diesem, kann ein Prozess oder Gegenstand im Raum von allen Seiten gleichermaßen betrachtet werden. Dadurch ist es dem Lernenden möglich das Objekt in seiner Ganzheit zu erfassen. Darüber hinaus können durch eine freie Navigation im Raum dargestellte Objekte sowohl in ihrer Gesamtheit als auch detaillierte Teilbereiche dieser betrachtet werden.

Um die Lerneffizienz weiter zu steigern sollte es möglich sein Animationen in dieser Präsentation ablaufen zu lassen und, um den Unterschied zu einer Videowiedergabe zu benennen, den Lehrenden interaktiv in den Ablauf der präsentierten Vorgänge eingreifen zu lassen. Dadurch ist es ihm möglich auf eventuelle Schwerpunkte oder Schlüsselszenen zeitkonkret einzugehen.

Jedoch sollten Vorteile einer 2D- Folienpräsentation nicht unbetrachtet bleiben. Die Wichtigkeit von Text bei der Wissensvermittlung ist unumstritten. Sie ist für den Lernenden unverzichtbar. Darum ist es erforderlich, diese Bausteine zu einem Werkzeug der Unterrichtsgestaltung zu verbinden.

1.2 Gliederung der Arbeit

Die Belegarbeit beginnt im Kapitel 2 mit einer Betrachtung verschiedener 3D-Techniken. Darin werden die Grundprinzipien von OpenGL, Open Inventor, Java3D, VRML und X3D vorgestellt und voneinander abgegrenzt. Darüber hinaus wird auf die Unterschiede sowie Gemeinsamkeiten bei der Umsetzung der Techniken in einer 3D-Präsentation eingegangen.

Das dritte Kapitel stellt Werkzeuge zur Präsentationserstellung vor. Es werden vier verschiedene Instrumente betrachtet und analysiert. In einer Beurteilung wird geprüft inwieweit sich die Werkzeuge für die Umsetzung der in der Aufgabenstellung geforderten Kriterien eignen. Es handelt sich bei der untersuchten Software um das Autorenwerkzeug Macromedia Director, die Werkzeuge EON Studio und Cult3D sowie das Maya Plug-in RawKee.

Jede erstellte Präsentation, unabhängig vom Erstellungswerkzeug, benötigt eine Plattform zur Visualisierung. Abspieldateien stellen diese dar und werden ebenfalls im dritten Kapitel vorgestellt und verglichen. Des Weiteren werden die bestehenden Player bezüglich ihrer Handhabung beurteilt.

Die Ergebnisse der Untersuchungen aus Kapitel 3 und der Vergleich mit den an die Software gestellten Kriterien führen zur Entscheidung das RawKee Plug-in zu erweitern und den geforderten Charakteristiken anzupassen. Das Plug-in wird im vierten Kapitel detaillierter betrachtet, ausgewertet und die Erweiterungsmöglichkeiten aufgezeigt.

Die Umsetzungen der Plug-in Erweiterungen sind in Kapitel 5 beschrieben. Dabei wird sowohl auf die Maya-Programmierung in Form von MEL-Skripten als auch die Umsetzungen mittels der C++ API eingegangen. Des Weiteren wird die Visualisierung der Erweiterungen in der Software Alias Maya dargestellt.

Das sechste Kapitel beschreibt ein Tutorium zum weiterentwickelten Maya Plug-in. Es wird der gesamte Erstellungsprozess einer 3D-Präsentation, vom Ausführen der Software bis zum Export und der Visualisierung der erzeugten 3D-Szene, erläutert.

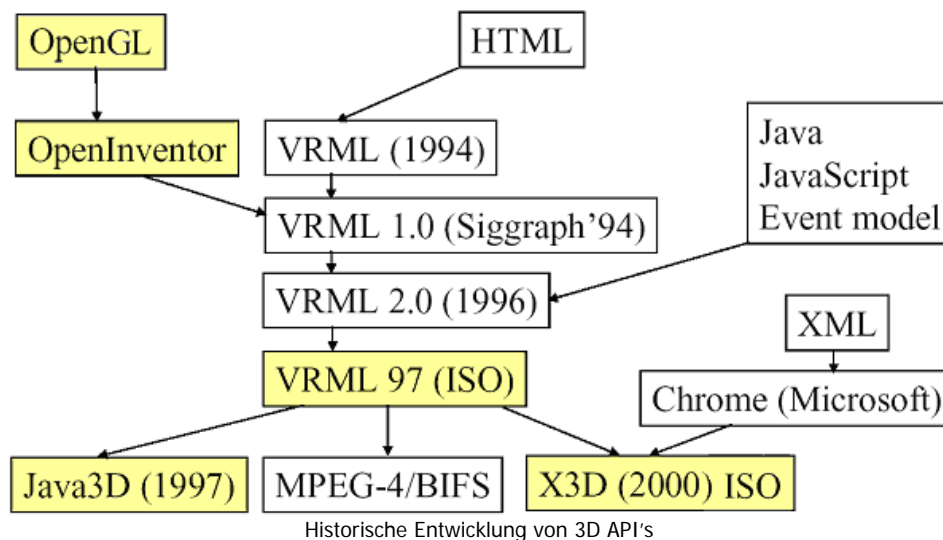
Das letzte Kapitel bildet den Abschluss dieser Arbeit und gibt eine Zusammenfassung dieser Studienarbeit sowie einen Ausblick auf zukünftige Projekte.

II 3D Techniken

1 Einführung

3D Techniken werden zur Erzeugung und Animation von Objekten in 3D Szenen genutzt. Heutige Anwendungen erweitern Textobjekte, Grafiken und Primitive auf die 3. Dimension, integrieren Audio, Video sowie Animationen und ergänzen diese durch Navigations- und Interaktionsmöglichkeiten. Sie finden ihren Einsatz in wissenschaftlichen Visualisierungen sowie im Bereich der virtuellen Realität. Das Spektrum der Nutzer von 3D-Präsentationen ist auf diesem Gebiet sehr vielfältig. Aufgrund dessen sind Anwendungen in naturwissenschaftlichen Bereichen wie der Medizin oder der Mathematik ebenso wie in technischen Sektoren denkbar. Bei Simulationen, Produkt- oder Konzeptvisualisierung kommen 3D Techniken ebenfalls zum Einsatz. Dabei bilden die Grafiksprachen OpenGL, Java3D, VRML und dessen Nachfolger X3D frei verfügbare „Low Level“ Standards und stellen die Grundlage für viele 3D- aber auch 2D-Anwendungen dar.

Im Folgenden ist eine Übersicht über die Entwicklung der 3D API's (Application Programming Interface) dargestellt. In den nächsten Abschnitten wird auf die verschiedenen Techniken, die in der Grafik gelb unterlegt sind, näher eingegangen.



2 OpenGL

OpenGL (Open Graphics Library) ist eine Spezifikation für ein plattform- und programmiersprachen-unabhängige Schnittstelle zur Entwicklung von 2D- und 3D-Computergrafiken. Der OpenGL Standard beschreibt etwa 250 Befehle, die eine Darstellung komplexer 3D-Szenen in Echtzeit erlauben.

OpenGL entstand ursprünglich aus dem von Silicon Graphics (SGI) entwickelten IRIS GL. Im so genannten Fahrenheit-Projekt versuchten Microsoft und SGI ihre 3D-Standards zu vereinheitlichen. Das Projekt wurde jedoch aufgrund finanzieller Schwierigkeiten auf Seiten von SGI abgebrochen.

Der OpenGL-Standard wird vom OpenGL ARB (Architecture Review Board) festgelegt. Das ARB existiert seit 1992 und besteht aus einer Reihe von Firmen. Stimmberechtigte Mitglieder sind die Firmen 3DLabs, Apple, ATI, Dell, IBM, Intel, NVidia, SGI und Sun (Stand Nov.2004). Microsoft hat das ARB im März 2003 verlassen.

Neue Funktionen in OpenGL werden meist zuerst als herstellereigenspezifische Erweiterungen eingeführt und gehen dann den Weg über herstellerübergreifende und ARB Erweiterungen zur Kernfunktionalität. Dies erlaubt es neue Möglichkeiten der Grafikhardware zu nutzen und dennoch OpenGL abstrakt zu halten.

Aufgrund seiner Plattformunabhängigkeit ist OpenGL im professionellen Bereich als 3D-Standard nach wie vor führend. Im Bereich der Computerspiele wurde es jedoch in den letzten Jahren zunehmend von Microsofts Direct3D verdrängt. Nach sieben Vorgängerversionen ist OpenGL 2.1 die aktuelle Version des Standards.

2.1 Das Programmiermodell

OpenGL ist eine standardisierte Grafikkbibliothek. Sie ist keine „High Level“ Objektbeschreibungssprache, sondern stellt ausschließlich Punkte, Linien und konvexe Polygone räumlich dar. Dem Programmierer wird eine kleine Sammlung geometrischer Primitive, Punkte, Zeilen, Polygone, Bilder und Bitübersichten zur Verfügung gestellt. Die Befehle steuern, wie diese Nachrichten in den Bildspeicher übertragen werden. OpenGL ist dabei als state machine (Zustandsmaschine) implementiert. Dies bedeutet, dass der Zustand der Render Engine über Schaltfunktionen eingestellt wird und für alle weiteren Funktionsaufrufe erhalten bleibt. Dieser Status bleibt unverändert, solange keine weiteren Zustandsänderungen auftreten und damit die Parameter die eine Darstellung von gerenderten Objekten beeinflussen können, vorliegt. Auf diese Weise werden beispielsweise nicht für jeden Vertex die gewünschte Farbe mitgeteilt, sondern diese wird einmalig gesetzt, woraufhin alle folgenden Vertices in dieser Farbe dargestellt werden. Auf dieselbe Weise ist es möglich globale Lichtquellen an- oder auszuschalten und viele weitere Zustände zu konfigurieren.

Der Grund für dieses Design begründet sich in der aufwändigen Reorganisation der Rendering Pipeline bei fast jeder Änderung des Zeichenmodus. Daher wird die Neuordnung vermieden so lange es sinnvoll möglich ist. Auch wäre es für den Programmierer ermüdend, Dutzende von Parametern immer wieder neu anzugeben. Oft können viele tausend Vertices bearbeitet werden bevor wieder ein Zustand geändert werden muss, während manche Zustände sogar gänzlich unverändert bleiben. Beispielsweise bleiben Lichtquellen meistens für alle Objekte einer Szene konform. Viele Zustände werden zumindest für die Dauer des Renderns eines kompletten Objekts beibehalten. Zum Beispiel wird ein Auto komplett um einen bestimmten Vektor verschoben und nicht in seine Einzelteile zerlegt und diese einzeln verschoben. Auch in Direct3D verfolgt man dieses zustandsbasierte Konzept.

2.2 Schnittstellen zum System

Da es sich bei OpenGL um eine reine Grafikkbibliothek handelt, sind nur die Grafikkommandos genormt. Die Verwaltung von Fenstern, Puffern (wie etwa dem Z-Buffer) und Renderkontexten, um mit mehreren Anwendungen gleichzeitig die Schnittstelle nutzen zu können, wird dem Windowsystem überlassen. Diese Funktionalitäten müssen mit Hilfe dafür vorgesehener, betriebssystemabhängiger Bibliotheken zur Verfügung gestellt werden.

Es existieren verschiedene Bibliotheken die OpenGL mit dem architektonisch darunter liegenden Betriebssystem verbinden. Mit der Zusatzbibliothek GLU (OpenGL Utility Library) besteht die Möglichkeit der Darstellung komplexer geometrischer Objekte und ist aufgrund dessen, fester Bestandteil jeder OpenGL Implementierung und zugleich Grundlage für viele Anwendungen. GLX, WGL und AGL sind betriebssystemabhängige Bibliotheken die OpenGL mit dem Betriebssystem verknüpfen. GLUT baut auf diesen Bibliotheken auf und bietet eine plattformunabhängige API für Ein- und Ausgabe sowie die Erstellung von Renderkontexten.

Aufgrund des beträchtlichen Funktionsumfangs von OpenGL bauen viele „High Level“ APIs wie Open Inventor auf dieser Technologie auf. Darüber hinaus postieren sich Applikationen wie Alias Maya und 3D Studio Max ebenso wie Computerspiele gleichsam auf der Grafikspezifikation.

3 Open Inventor

Open Inventor ist eine objektorientierte C++-Grafikkbibliothek zur Erstellung von 3D-Grafiken unter Verwendung von OpenGL. Es besteht aus einer Sammlung von Objekten und Methoden zur Erzeugung interaktiver 3D-Grafikapplikationen. Es ist ein Werkzeug welches über abstrahierende Funktionen verfügt, die über den elementaren Grafikbefehlen angelegt sind. Da Open Inventor auf Open GL basiert, ist es auf verschiedenen Plattformen verfügbar.

Es bietet ein Sortiment von Bausteinen, aus denen man ganze Applikationen zusammensetzen kann. Diese Bausteine bedienen sich schneller Hardwareunterstützung und sind sehr systemnah programmiert. Obwohl sie sehr umfangreich sein können, bringt ihre Verwendung nur minimalen Programmieraufwand mit sich.

Open Inventor enthält vordefinierte Objekte wie unter anderem Kugeln, Zylinder und Facesets. Des Weiteren ist es relativ einfach den Katalog um eigene Körper zu erweitern. Die Komponenten einer Applikation werden in einem Szenengraph abgelegt und dann über OpenGL gerendert.

3.1 Szenendatenbank

Open Inventor arbeitet auf der Basis eines Szenengraphen. Der Graph wird in dieser Programmbibliothek Szenendatenbank genannt und ist eine dynamische Repräsentation der darzustellenden Szene als Graph. Er ist im Normalfall gerichtet und azyklisch. Objekte werden darin als Knoten (Nodes) dargestellt und gespeichert. Die Knoten eines Inventorgraphen setzen sich aus verschiedenen Klassen zusammen. Es wird zwischen geometrischen Objekten, den Eigenschaften dieser Objekte und Operationen auf ihnen unterschieden. Eine weitere Knotenart definiert Aktionen. Diese Aktivitäten dienen verschiedenen Zwecken: zum Rendern, zum Auswählen von Objekten, zur Berechnung von Bounding Boxes, zur Verarbeitung von Events und zur Dateiausgabe. Des Weiteren existieren Objekte zur vereinfachten Erzeugung von Animationen. Der Bereich der Interaktion beinhaltet Klassen für Events und so genannte "Smart Nodes". Eventklassen dienen beispielsweise zur Verwaltung von Maus-Button-Events oder reagieren auf bestimmte Tastenbetätigungen. "Smart Nodes" enthalten spezielle Intelligenz, die sie zur Ausführung von Funktionen benötigen. So kann zum Beispiel ein Knoten zur Selektion von Objekten über Mausevents das Picking überwachen und die selektierten Objekte farblich hervorgehoben darstellen. Node Kits stellen sinnvoll zusammengefasste Mengen von Knoten dar, die den Aufbau von komplexen und dabei konsistenten Szenen ermöglichen. Kombinierbare Knotenarten werden zu Subgraphen zusammengefügt. Wird ein Node Kit in die Szene eingearbeitet, werden beispielsweise automatisch ein Knoten für ein geometrisches Objekt, ein Materialknoten und eine Transformation gleichzeitig eingefügt. Auf Applikationsebene beinhaltet Open Inventor noch einige Hilfsprogramme, wie zum Beispiel Editoren für Farbe, Material, Lichtquellen, sowie verschiedene Viewer, die spezielle Interaktionen des Benutzers mit der Präsentation ermöglichen.

4 Java3D

Java3D ist eine Bibliothek von Java-Klassen zur Erzeugung, Manipulation und Darstellung dreidimensionaler Grafiken innerhalb von Java-Applikationen und -Applets. Mit der Java3D-API können, in ein Java-Programm eingebettet, dreidimensionale Objekte modelliert, gerendert sowie das Verhalten und die Ansicht gesteuert werden.

4.1 Szenengraph

Java3D wird seit 1997 von Sun entwickelt. Die Version 1.0 erschien im Dezember 1998 und die aktuelle Version 1.4 ist seit März 2006 veröffentlicht. Als ein wichtiges Leistungsmerkmal der aktuellen Fassung ist die Möglichkeit der Shader-Programmierung für aktuelle Grafikhardware hervorzuheben.

Java 3D organisiert die Funktionalität der zugrunde liegenden OpenGL bzw. DirectX Schnittstelle in ein leichter verständliches objektorientiertes Programmkonzept auf Basis eines Szenengraphen. In diesem wird der logische Aufbau der darzustellenden Objekte auf eine gleichartig aufgebaute, baumähnliche Struktur abgebildet, die im Wesentlichen aus Definitionen von Transformationen und Geometriedaten besteht. Die so strukturierte Sicht der Szene erlaubt eine komfortable Handhabung der Objekte. Auch die Integration einer Soundkulisse in den Szenengraphen ist möglich.

Die Bibliothek beinhaltet Objekte und Methoden zur Präsentation von 3D-Szenen und nutzt für deren Darstellung ausschließlich vorimplementierte und an das jeweilige Betriebssystem angepasste Funktionsbibliotheken. Dabei wird zur Darstellung aller erzeugten Objekte die im System eingesetzte Videohardware verwendet. Potentiell ist damit die erreichbare Renderinggeschwindigkeit ähnlich hoch wie bei direkter Programmierung mit C und OpenGL oder Direct3D. Daher besteht die Möglichkeit, leistungsfähige 3D-Szenarien zu entwickeln, welche auf einer grafischen Bedienoberfläche visualisiert werden können.

Ein direkter Zugriff auf OpenGL- bzw. Direct3D-Funktionen ist von Java 3D nicht vorgesehen. Leistungsmerkmale dieser Schnittstellen, die von Java 3D nicht gekapselt werden, können damit von Applikationen auf Basis dieser Technologie nicht verwendet werden. Aufgrund der stetigen Weiterentwicklung der Videohardware hat dies wiederholt dazu geführt, dass neu unterstützte Leistungsmerkmale in Java3D nur verzögert oder gar nicht verfügbar sind.

4.2 Koordinatensystem

Die Orientierung der Raumachsen in Java 3D bildet ein rechtshändiges Koordinatensystem. In der Nomenklatur von Java 3D wird der so aufgespannte Raum als Virtuelles Universum bezeichnet. Lediglich

dieses Koordinatensystem ist vorgesehen, es gibt keine Möglichkeit den Achsen eine andere Orientierung zuzuordnen.

Die so genannten Locales-Objekte erreichen die Umsetzung eines sehr präzisen Koordinatensystems. Durch die Koordinaten ist es möglich, von atomarer Ebene bis zum Makrokosmos alle Objekte in einem Koordinatensystem berechnen zu lassen. Das Virtuelle Universum kann dabei mehrere Locales verwalten, welches verschiedene Lokalitäten in dem virtuellen Universum modelliert und zueinander in Beziehung setzt. Aufgrund dessen wird die Nutzung von Java3D für viele wissenschaftliche Anwendungen, beispielsweise im Bereich der Astronomie, vereinfacht.

5 VRML

VRML bedeutet Virtual Reality Modeling Language und ist eine Beschreibungssprache für 3D-Szenen inklusive Geometriedarstellungen, Ausleuchtung von Szenen, Animationen und Interaktionsmöglichkeiten. Sie ist plattformunabhängig und durch Browser Plug-ins oder eigenständige Abspielsoftware interpretierbar.

VRML wurde ursprünglich als 3D-Standard für das Internet entwickelt. Die meisten 3D-Modellierungswerkzeuge ermöglichen den Im- und Export von VRML-Dateien, wodurch sich das Dateiformat auch als ein Austauschformat von 3D-Modellen etabliert hat. Eine VRML-Darstellung (zum Beispiel innerhalb eines Webbrowsers oder einer Standalone Anwendung) wird vom Computer des Betrachters in Echtzeit generiert. Das bedeutet, dass der Computer jedes einzelne Bild aus den vorhandenen Geometriedaten, sowie dem Verhalten und den Bewegungen der Szenerie ständig neu berechnet. Aus diesem Grund scheiden fotorealistische Darstellungen mit rechenaufwendigen Raytracing- oder Radiosity-Verfahren aus. Es werden auch beim Benutzen von vordefinierten Betrachterpositionen (Viewpoints), beim Wechsel zwischen diesen Punkten und bei Kamerafahrten keine fertigen Bilder aus Filmsequenzen abgespielt.

Komplexe VRML-Szenen stellten früher daher unter Umständen hohe Anforderungen an die Hardware. Wie schnell, beziehungsweise wie flüssig die Bewegungen in einer Szene erfolgen, hängt vom Prozessor(takt) und vor allem von der Grafikkarte des wiedergebenden Computers ab.

5.1 Geschichtlicher Abriss

Der Grundstein für VRML wurde von SGI 1992 mit dem Open Inventor ASCII Format gelegt. Auf der W3C Tagung 1994 in Genf wurde erstmals der Begriff VRML geprägt. Noch im selben Jahr, zur 2. W3C Tagung, erfolgte die Einigung auf die Spezifikation VRML 1.0. Ebenso vollzog sich die Gründung der VRML Architecture Group, die sich 1998 zum Web3D Konsortium wandeln sollte. VRML unterstützte bis zu diesem Zeitpunkt nur statische Szenen und so war, aufgrund gestiegener Anforderungen und erweiterter Möglichkeiten, der daraus resultierende Schritt die Ausarbeitung von VRML2.0 im Jahr 1996. Durch diese Weiterentwicklung war die Erstellung von dynamischen und interaktiven 3D-Welten möglich. Gleichermaßen werden seither Skalierbarkeit und Skripting unterstützt. 1997 wurde der Progress zu VRML97 weiterentwickelt und als ISO/IEC14772-1 verabschiedet.

5.2 Dateiaufbau und Szenengraph

VRML-Dateien bestehen aus einem Vorspann (Header) und einem Rumpf, in dem Knoten (Nodes) definiert sind, die als Informationseinheiten der Szene fungieren. In der Headerzeile stehen Auskünfte zum Dokumentenformat sowie zur verwendeten Version der Sprache. Diese Daten fungieren als Dateidentifikatoren für die Abspielsoftware. Darauf folgende Knoten dienen der Beschreibung der Objekte und deren Eigenschaften in der 3D-Welt. Verschiedene, in VRML definierte Knoten, beschreiben unter anderem Gestaltseigenschaften, Farbattribute, Lichtquellen, Viewpoints, Sensoren und Interpolatoren. Knoten bilden folglich die Bausteine eines VRML Knotenbaumes und setzen sich im Allgemeinen aus einem Knotentyp und den zu diesem gehörenden Attributfeldern zusammen. Diese Felder und die in ihnen festgelegten Werte definieren innerhalb geschweifter Klammern die Knotenattribute.

Im Folgenden sind die wichtigsten Knotentypen tabellarisch aufgelistet und werden in den darauf folgenden Abschnitten näher beschrieben:

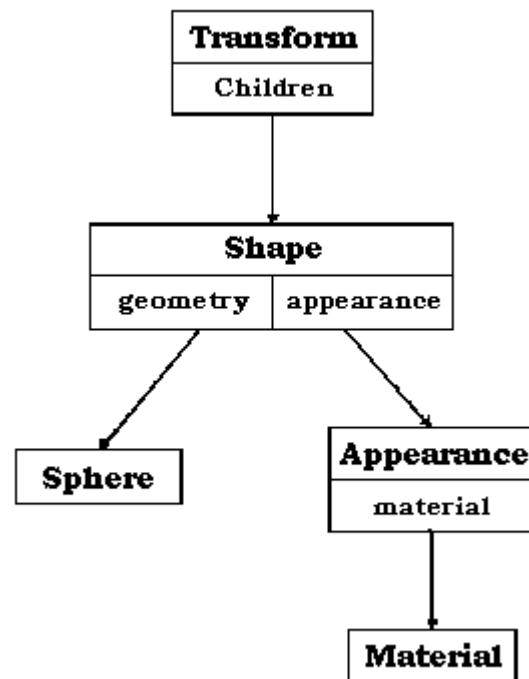
Einträge	Beschreibung
Gestalt und Geometrie	Box, Cone, Coordinate, Cylinder, ElevationGrid, Extrusion, IndexedFaceSet, IndexedLineSet, Normal, Pointset, Shape, Spere, Text
Erscheinungsbild	Appearance, Color, FontStyle, ImageTexture, Material, MovieTexture, PixelTexture, TextureCoordinate, TextureTransform
Gruppierung	Anchor, Billboard, Collision, Group, Inline, LOD, Switch, Transform
Umgebung	AudioClip, Background, DirectionalLight, Fog, PointLight, Sound, SpotLight
Navigation	NavigationInfo, Viewpoint
Animation (durch Interpolatoren)	ColorInterpolator, CoordinateInterpolator, NormalInterpolator, OrientationInterpolator, PositionInterpolator, ScalarInterpolator, TimeSensor
Interaktion (durch Sensoren)	CylinderSensor, PlaneSensor, ProximitySensor, SphereSensor, TouchSensor, VisibilitySensor
Sonstiges	Script, WorldInfo

Die nachfolgende Grafik veranschaulicht schematisch den Aufbau eines Knotenbaums beziehungsweise Szenengraphen und den dazugehörigen VRML-Quelltext einer Kugel. Diese wird durch ihre Gestalt sowie das Oberflächenmaterial beschrieben und durch einen Gruppierungsknoten zusammengefasst.

```

Transform {
  translation 3 0 0
  children [
    Shape {
      geometry Sphere {
        radius 2
      }
      appearance Appearance {
        material Material {
          diffuseColor 1 0 0
        }
      }
    }
  ]
}

```



5.3 Animation und Interaktion

Um Animationen und Interaktionen in einer VRML-Szene zu realisieren, müssen Knoten und deren Eigenschaften verknüpft werden können. Dadurch kann beispielsweise ein Tastendruck die Oberflächenbeschaffenheit einer Kugel verändern. Verbindungen zwischen den Knoten einer Szene werden durch so genannte Routes gebildet, über die ein Austausch von Nachrichten (Events) ermöglicht wird. Events werden typischerweise durch Sensoren generiert und ausgelöst. Dabei können Knoten Nachrichten sowohl senden (EventOut) als auch empfangen (EventIn). Die Routes dienen damit als Verbindungsglied der Kommunikationspartner beim Nachrichtenaustausch. Routes haben in einem VRML Dokument die folgende Form:

```
Route NodeName.event.OutName to NodeName event.InName
```

Events bestehen aus zwei Komponenten, Veränderung und Zeitpunkt, die eine eindeutige Auslösung der Aktion ermöglichen. Somit kann widerspruchsfrei festgelegt werden welche Veränderungen, bei dem zu

einer klar definierten Zeit auftretenden Ereignis, ausgelöst werden. Events können zum einen durch ein Ereignis im Ablauf der Szene oder direkt durch eine Interaktion mit dem Nutzer ausgelöst werden. Dafür werden Sensoren Ereignisse zugeordnet, die bei der Wechselbeziehung zwischen Anwender und Software ausgeführt werden. Der Benutzer kann beispielsweise eine Animation über Time-Sensor Knoten zeitgesteuert starten und stoppen oder direkt über einen Touch-Sensor auf den Time-Sensor einwirken und den Verlauf manuell beeinflussen. Der aktive Einfluss des Benutzers beschränkt sich jedoch auf relativ einfache Abläufe. Durch das Einbinden von Skripten ist es jedoch möglich auch weitaus komplexere Aufgaben zu definieren. Unterstützte Sprachen sind VRML-Script, JavaScript und Java, die über den Script-Knoten in ein VRML-Dokument eingebunden werden können.

Das folgende Codebeispiel verdeutlicht die VRML-Eventstruktur. Ein Tastendruck auf die Maus lässt einen dargestellten Würfel transparent erscheinen. Dabei löst der definierte Touch-Sensor eine Animation aus, die den Transparentwert des Würfels über einen Zeitraum, definiert mittels Time-Sensor, hinweg verändert.

```
#VRML V2.0 utf8
Transform {
  children [
    Shape {
      appearance Appearance{
        material DEF material1 Material
        {
          diffuseColor 0.8 0.8 0.8
        }
      }
      geometry Box{
        size 2 2 2
      }
    }, DEF ausloeser TouchSensor {}
  ]
}
DEF globalClock TimeSensor {
  cycleInterval 5
}
ROUTE ausloeser.touchTime TO globalClock.startTime
DEF engine1 ScalarInterpolator {
  key [ 0.000 1.000 ]
  keyValue [ 0.000 , 1.000 ]
  ROUTE globalClock.fraction_changed TO engine1.set_fraction
  ROUTE engine1.value_changed TO material1.set_transparency
}
```

5.4 Schwächen von VRML

VRML hat in seiner bisherigen Geschichte nicht die erwartete große Verbreitung gefunden. Dies ist auf verschiedene Punkte zurückzuführen:

Das VRML-Format ist nicht streamingfähig und führt daher zu langen Ladezeiten auf Seiten des Anwenders. Ein weiterer Kritikpunkt ist die oft schlechte visuelle Qualität der Darstellung im Browser. Die Browser können eine 3D-Szene darüber hinaus nur mit Hilfe eines Plug-ins anzeigen, welches zuvor aus dem Internet heruntergeladen werden muss. Dies wirkt ebenfalls einer raschen Verbreitung entgegen.

Eine VRML-Datei erzeugt monolithisch große Welten auf der Basis von ASCII wobei für die Verbreitung im Internet ein modularer Komponentenansatz sowie ein Binärformat Vorteile bieten würden. Die VRML-Syntax ist nicht objektorientiert und die VRML-Skriptsprache verfügt über keine standardisierten APIs. Der Effekt der Kapselung wird noch durch die fehlende Interoperabilität mit anderen Mediennetzstandards verstärkt.

Komplexe Programmstrukturen sind mit VRML allein nicht zu erzeugen. Die Gesamtheit der Kritikpunkte führt zu Akzeptanzschwierigkeiten aufseiten des Nutzers.

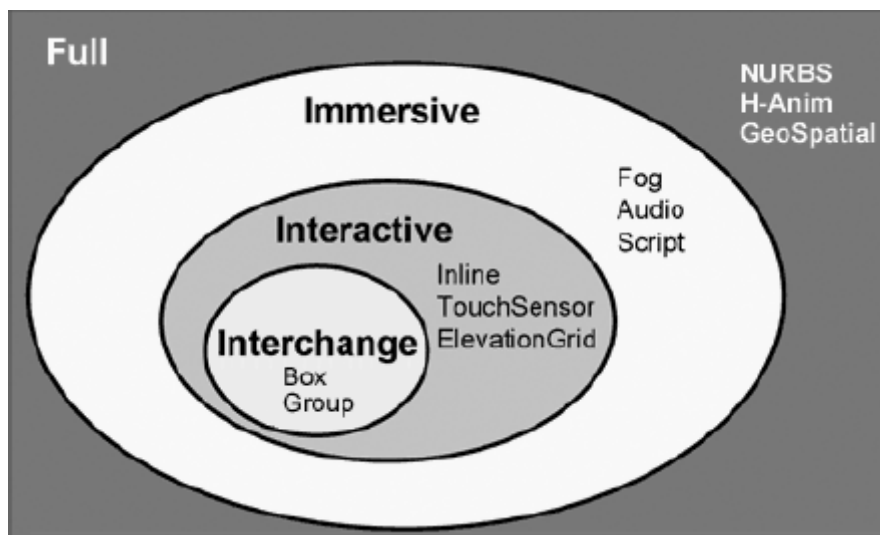
6 X3D

Extensible 3D, kurz X3D, ist eine 3D-Modellierungssprache, die in ihrer Syntax entweder auf XML (XML-Encoding, Dateiendung .x3d) oder auf VRML (Classic-Encoding, Dateiendung .x3dv) aufbauen kann. X3D bildet den offiziellen Nachfolger des VRML-Standards mit dem Ziel dessen Schwächen und Nachteile zu beseitigen. Der Markt stellt die Anforderung die Interoperabilität mit anderen Netzstandards herzustellen und damit verbunden eine engere Medienintegration zu ermöglichen. Der komponentenbasierte Ansatz des Dateiformates macht X3D flexibler und leistungsfähiger. Das Format ist weiterhin in der Entwicklung durch das Web3D Konsortiums und soll zu VRML kompatibel bleiben. Seit 2004 ist X3D als offener ISO Standard für interaktive 3D Darstellungen spezifiziert.

Das X3D Format soll seine Zielsetzungen in vielen Bereichen wie Wirtschaft und Bildung verwirklichen. Präsentationen dieser 3D Technik werden bei E-Commerce Produkten und Technikpräsentationen eingesetzt. Weiterhin findet sie Verwendung in visuellen Simulationen und Trainingsanwendungen. Weitere mögliche Einsatzgebiete sind unter anderem Spiele und Unterhaltungssoftware sowie in den Bereichen Architektur und Navigation.

6.1 Profile

Das X3D Format ist in verschiedene Profile untergliedert. Diese bauen aufeinander auf wobei der Nachfolger das gesamte Funktionsspektrum des Vorgängerprofils in sich vereinigt. Die abgebildete Grafik veranschaulicht die X3D Profile.



X3D unterstützt das immersive Profil

Das grundlegende Profil wird „Interchange“ genannt. Es unterstützt den Austausch von Geometriedaten, Texturen und simpler Beleuchtung sowie einfachen Animationen. Nicht unterstützt werden dagegen Interaktionen, weder innerhalb der Komponenten der Anwendung noch durch externe Einflüsse wie den Anwender. Darauf aufbauend bildet das „Interactive“ Profil die nächste Entwicklungsstufe. Dabei wird durch die Integration von Sensoren Interaktionsmöglichkeiten mit der 3D-Welt bereitgestellt. Des Weiteren wurden die Beleuchtungsmöglichkeiten verbessert. Beim derzeitigen Entwicklungsstand ist X3D dem „Immersive“ Profil zuzuordnen. Es bietet eine vollständige Unterstützung für 3D Grafik und Interaktion. Darüber hinaus sind Audiounterstützung, Kollisionserkennung und Nebeleffekte realisiert. Ebenso können über Skriptknoten Java-Code und Javascripte in eine X3D-Datei integriert werden. Damit hat das

„Immersive“ Profil den Funktionsumfang von VRML97. Die letzte Stufe der Entwicklung bildet das „Full“ Profil. Es vereinigt alle Profile und unterstützt neben der bisherigen Polygon- ebenso die NURBS-Darstellung. Humanoid-Animation und Geo-Spatial sind weitere Komponenten die in dieses Profil integriert sind.

6.2 Architektur

Grundkonzept: Szenengraph

X3D ist XML basiert und übernimmt dadurch die strukturierte Beschreibung der Extensible Markup Language, die sich ideal für hierarchische Strukturen wie einen Szenengraphen eignet. Dieser, als gerichtete azyklische Baumstruktur, besteht aus Blatt- und Gruppenknoten, wobei letztere beliebige Knoten zusammenfassen können. Blattknoten können zudem in grafische und nicht grafische Knoten unterteilt werden, die abermals gliederbar sind. Geometrie-Knoten, mit denen sich einfache geometrische Objekte, wie zum Beispiel eine Kugel, definieren lassen, sind Teil der grafischen Knoten. Durch sie werden die Objekte einer Szene definiert. Hinzukommen verschiedene weitere Knotentypen zur Definition von Beleuchtung oder Kamerasicht. Durch Interpolator-Knoten können Animationen in die Präsentation integriert werden. Um Interaktionen zu ermöglichen erfordert es die Integration von Sensorknoten. Allen diesen Knoten können in so genannten Feldern definierte Eigenschaften zugewiesen werden. Dabei legt eine X3D-DTD die Default-Werte der Knoteneigenschaften fest. Diese Werte können durch die Definition von Events verändert werden und dadurch wiederum Animationen erzeugt werden.

Event Architektur

Die Animationen und Interaktionen einer X3D-Szene bauen auf dem „event passing mechanism“ auf. Dabei definiert jeder involvierte Knoten Namen und Typen von Ereignissen die er generieren oder empfangen kann. Routes sind dabei die Ereignispfade die zwischen Sender und Empfänger definiert werden. Es wird dabei, nach der Definition des Knotentyps, in diesem Fall Route, ein Attribut des Senderknotens mit einem des Empfängerknotens verbunden. Folgendes Beispiel soll dies verdeutlichen.

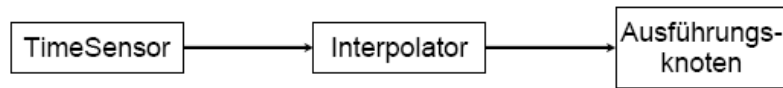
```
<ROUTE fromNode='<NODE>' fromField='<FIELD>' toNode='<NODE>' toField='<FIELD>' />
```

Zum Auslösen von Ereignissen werden Sensoren verwendet. Sensoren gehören zur Gruppe der Blattknoten. Sie bilden als Nutzerinteraktionsprimitive nur das auslösende Moment und müssen mit Routes verknüpft werden um Effekte zu erzielen. Für Animationen bildet der Time-Sensor die Basis. Er generiert Interrupts und ermöglicht aufgrund dieser Eigenschaft zeitgenaue Animationsabläufe. Gesteuert werden können diese Animation mittels diversen weiteren Sensoren wie dem Touch- oder dem Proximity-Sensor. Diese Sensoren erzeugen Events, wenn die ihnen übergeordnete Gruppe aktiviert wird. Wenn beispielsweise ein Touch-Sensor unter einer Kugel gruppiert ist, kann in der 3D-Szene durch anklicken der Kugel das Event ausgelöst werden.

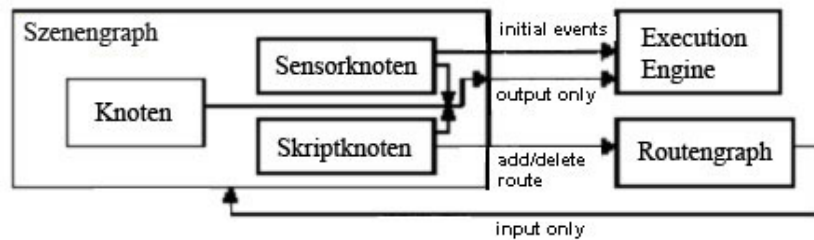
Skripte und Interpolatoren

Mit Skripten ist es möglich beliebiges Verhalten innerhalb einer X3D-Szene zu definieren. Es kann mittels Java Klassen eingebettet oder als Java- bzw. ECMA-Skript spezifiziert werden. Mithilfe der Skripte können einzelne Felder von Knoten gelesen oder geschrieben werden. Auf diese Art und Weise ist es möglich die Grenzen des X3D Formats zu erweitern. Die Skripte wiederum bilden Skriptknoten die zwischen Ereignis-Generatoren und Empfängern platziert werden können.

Interpolatoren gehören zur Gruppe der Blattknoten. Sie sind build-in Skripte, die Interpolationsberechnungen für einfache Animationen übernehmen. Diese werden meistens mit Time- und Event-Sensoren sowie weiteren Knoten kombiniert. Interpolatoren besitzen zwei charakteristische Felder: Key und KeyValue. Das erste Feld ist eine (sortierte) Menge von Schlüsseln, und das Zweite ist eine Menge von gleich vielen (oder evtl. einem Vielfachen von) Schlüsselwerten. Über ein inEvent wird ein Skalar an den Interpolator gesendet. Dieser prüft in welchem Schlüsselintervall sich dieser Wert befindet. Nun wird das korrespondierende Schlüsselwertintervall gesucht und der passende Wert durch lineare Interpolation zwischen den Intervallgrenzen bestimmt. Der berechnete Wert wird dann durch ein outEvent gesendet. Die folgende Grafik veranschaulicht dies.



Der wesentliche Nachteil eines Interpolators ist die Erzeugung ausschließlich linearer Interpolationen. Nichtlineare Verläufe, z.B. eine beschleunigte Bewegung, sind somit nicht exakt modellierbar. Dadurch können unschöne Unstetigkeitsstellen in der Animation auftreten. Dieser Nachteil kann auf zwei Arten kompensiert werden. Zum einen können durch Angabe genügend vieler Schlüssel und Schlüsselwerte nichtlineare Verläufe approximiert werden und zum anderen können externe Programme (z.B. JavaScript, Java) eingebunden werden, die eine genauere Berechnung durch imperative Sprachkonstrukte erlauben.



Zusammenwirken der Komponenten in einer X3D-Szene

Das folgende Codebeispiel zeigt eine X3D-Szene. Der Würfel ist mit einem Touch-Sensor verbunden, der ihn beim Anklicken in Bewegung versetzt. Die Animation ist mittels eines Positions-Interpolators in Verbindung mit einem Time-Sensor realisiert. Die verschiedenen Komponenten sind über Routes miteinander verbunden.

```

<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive" version="3.0">
  <head>
  </head>
  <Scene>
    <Transform DEF='W'>
      <Shape DEF='Wuerfel'>
        <Appearance DEF='initialShadingGroup'>
          <Material DEF='lam' ambientIntensity='0' diffuseColor='0 0.4 0' />
        </Appearance>
        <Box DEF='x3dBox1' />
      </Shape>
      <TouchSensor DEF='TS' />
    </Transform>
    <TimeSensor DEF='tis' cycleInterval='5' startTime='0' />
    <PositionInterpolator DEF='PI'
      key='0 0.5 1'
      keyValue='-10 0 -20 10 0 -20' />
    <ROUTE fromNode='TS' fromField='touchTime' toNode='tis'
      toField='set_startTime' />
    <ROUTE fromNode='tis' fromField='fraction_changed' toNode='PI'
      toField='set_fraction' />
  </Scene>
</X3D>
  
```

```

    <ROUTE fromNode='PI' fromField='value_changed' toNode='W'
    toField='translation' />
  </Scene>
</X3D>

```

7 Vergleich von VRML und X3D

Bei VRML und X3D handelt es sich um Beschreibungssprachen. Gemeinsamkeiten der beiden Sprachen liegen im Bereich des strukturellen Aufbaus. Sowohl VRML als auch X3D arbeiten mit einem Szenengraphen. Des Weiteren wurden eine Reihe von Knotenbezeichnungen von VRML in X3D übernommen und weitergeführt. Durch die Weiterentwicklung von X3D sind neue Knotenelemente in der Spezifikation hinzugekommen, die nicht von VRML unterstützt werden. Weiterhin existiert eine Reihe von Unterschieden, die bereits in den zurückliegenden Abschnitten beschrieben wurden und hier noch einmal tabellarisch zusammengefasst sind.

Die folgende Tabelle zeigt wesentliche Unterschiede zwischen beiden Sprachen

	VRML	X3D
Eindeutigkeit	<ul style="list-style-type: none"> - Unterschiedliche Browser können eine Szene different darstellen - Browser Plug-in und Anwendungen, haben die komplette Spezifikation implementiert 	<ul style="list-style-type: none"> - grundlegender X3D Viewer beinhaltet das Immersive – Profil - Eindeutigkeit gewährleistet - Spezifikationsneuerungen werden von Abspielsoftware nur teilweise unterstützt
Komplexität	<ul style="list-style-type: none"> - hohe Komplexität - kein binäres Dateiformat - Erstellung monolithischer Welten 	<ul style="list-style-type: none"> - binäres Dateiformat wird zur Verfügung gestellt - modularer Komponentenansatz
Programmierbarkeit	<ul style="list-style-type: none"> - Keine objektorientierte Programmierung - Keine standardisierte API 	<ul style="list-style-type: none"> - Erweiterbarkeit und Objektorientierung - Interoperabilität mit Standard-API Modellen
Features	<ul style="list-style-type: none"> - keine zusätzlichen Features 	<ul style="list-style-type: none"> - Streaming - Ausführen von DirectX 9 Funktionen - Auswertung von Tastatureingaben

III Vergleich bestehender Software

1 Software zur Präsentationserstellung

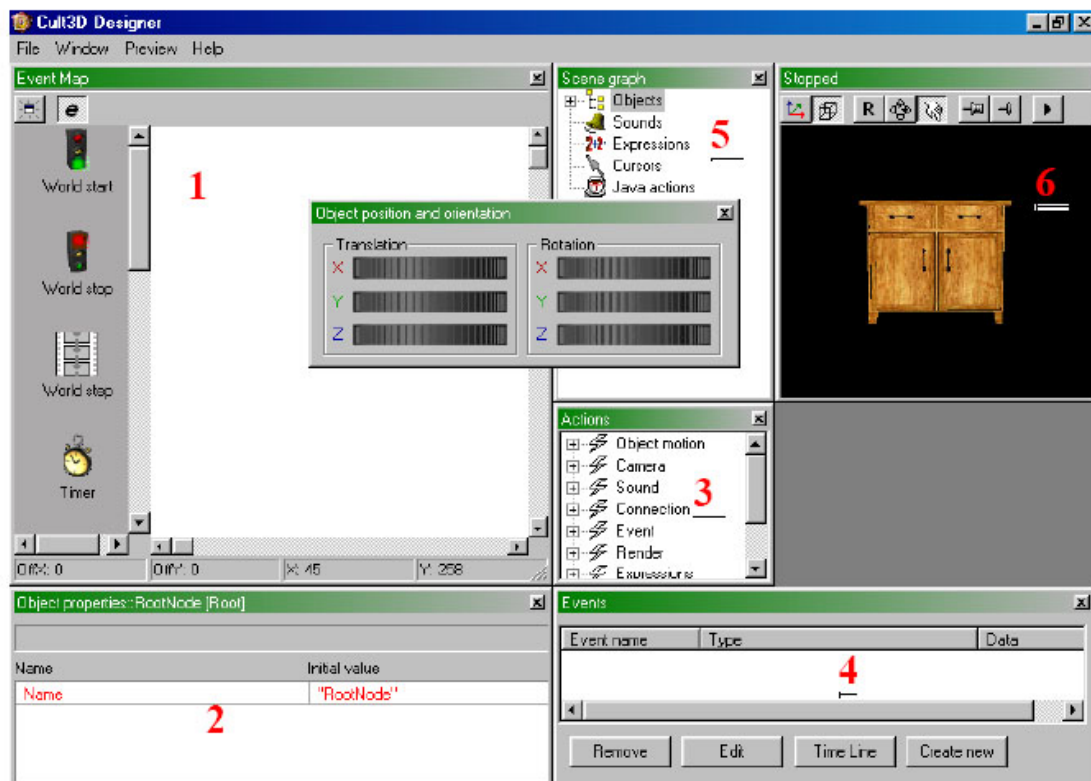
1.1 Cult3D

Cult3D ist eine Software zur Erzeugung interaktiver 3D-Szenen. Mit diesem Werkzeug ist es möglich Objekten Funktionen zuzuweisen und Interaktionen mit diesen zu definieren. Das Cult3D-Softwarepaket besteht aus drei Komponenten: Das Cult3D Exporter Plug-in, der Cult3D Designer und der Cult3D Player bilden zusammen das Werkzeugpaket zur Erstellung interaktiver Präsentationen.

Die Erzeugung der 3D-Szene muss in einer separaten Modellierungssoftware wie Alias Maya oder 3D Studio Max durchgeführt werden. Nach Beendigung der Modellierung, Texturierung und Animationserstellung erfolgt der Export mittels des Exporter Plug-ins. Es stehen für die Modellierungswerkzeuge von Seiten der Cult3D Entwickler spezielle Exporter zur Verfügung. Jedoch stellen sie derzeit nur für 3D Studio Max und ältere Maya Versionen einen Exporter zur Disposition. Beim Export einer Szene wird eine c3d-Datei erzeugt die im Cult3D Designer importiert werden kann. Der Designer bildet den Kern von Cult3D. Mit diesem Werkzeug werden die 3D-Objekte mit Interaktionselementen verknüpft. Zum Visualisieren einer erstellten Szene wird der von Cult3D bereitgestellte Cult3D Player benötigt. Der Designer erzeugt beim Export co-Dateien. Diese Dateien sind komprimiert und optimiert um die Dateigröße zu reduzieren und können im Cult3D Player angezeigt werden.

Cult3D Designer

Die Oberfläche des Cult3D Designers setzt sich aus verschiedenen Fenstern innerhalb der Bedienoberfläche zusammen. Die erstellte und importierte Szene kann während des Bearbeitungsprozesses um Interaktionen und Animationen erweitert werden. Darüber hinaus können Partikelsysteme, Reflexionen, Sound und weiterer Effekte der Szene hinzugefügt werden. Die folgende Abbildung zeigt die Bedienelemente des Designers.



Die Bedienoberfläche des Cult3D Designers

Über das Fenster der „Event Map“ (1) werden Events definiert, um Aktionen auszulösen. Events können sowohl Maus- als auch Tastatureingaben sein. Die Eigenschaften und Zusammenhänge von auslösenden

Events und den darauf folgenden Aktionen werden in Eventgraphen innerhalb dieser Ansicht visualisiert. Es können Interaktionen zum Auslösen von Rotation und Translationen der Objekte definiert werden, aber auch das Abspielen von in der Modellierungssoftware erstellten Animationssequenzen und deren Reihenfolge innerhalb der 3D-Szene vereinbart werden. Über eine Verknüpfung von Eventgraphen ist es dem Nutzer möglich eine Kette von auslösenden Events zu erstellen und damit einen sequenziellen Ablauf von Teilszenen zu definieren. Im Fenster „Events“ (4) werden bestehende Ereignisdefinitionen aufgelistet und verwaltet und können gegebenenfalls bearbeitet beziehungsweise entfernt werden. Das Fenster „Scene graph“(5) zeigt eine hierarchische Anordnung der Objekte der Szene sowie weitere Komponenten wie beispielsweise Kameras und Lichtquellen. Die Objekte des Szenengraphen sind die aus der Modellierungssoftware exportierten Elemente die mittels der Event Map mit Interaktionen versehen werden können. Die Baumstruktur kann im Szenengraphen aufgebrochen werden um Abhängigkeiten zwischen den Komponenten der Szene zu ändern und neu zu strukturieren. Durch das Vorschauenfenster(6) hat der Nutzer jederzeit die Möglichkeit die erstellte Szene zu sehen und die Präsentation zu starten um gegebenenfalls definierte Interaktionen oder Animationen zu testen. Das Fenster zeigt nach Aktivierung immer den aktuellen Entwicklungsstand der Szene an. Der Cult3D Designer stellt vorgefertigte Aktionen im „Actions“ Fenster(3) zur Verfügung. Mit diesen können die gesamte Szene sowie einzelne Objekte manipuliert werden. Dazu werden eine Aktion mit einem Event und die Daten mit der Aktion, die von den Daten durchgeführt werden sollen, verbunden. Im „Object Properties“ Fenster(2) werden gespeicherte Daten, die von der Präsentation verwendet werden, angezeigt. Jedes Objekt der Szene besitzt einen Satz ihm zugehöriger Eigenschaften. Eine Eigenschaft setzt sich aus dem Tripel Name, Initial-Wert und dem Datentyp zusammen. Jedes Objekt hat bei der Initialisierung eine festgelegte Menge von Eigenschaften, die durch den Nutzer erweitert werden kann. Im Cult3D Designer ist es mittels der verschiedenen Bedienelemente und ohne weiteren Programmieraufwand möglich, eine 3D-Welt mit Interaktionen zu versehen und zu einer Präsentation zu komplettieren.

Lizenzvereinbarung

Die Software die unter der Adresse <http://www.cult3d.com/download/download.asp> herunterzuladen ist, unterliegt Lizenzvereinbarungen. Diese legen fest, dass eine Software nur auf einem Computer gleichzeitig installiert sein darf und zusätzlich eine Kopie in digitaler Form zur persönlichen Sicherung der Software freigestellt ist. Die mithilfe der Software erstellten Präsentationen dürfen nur betriebsintern oder privat gezeigt werden. Bei einer Nutzung außerhalb dieser Rahmenbedingungen ist eine Lizenzgebühr an die Entwicklerfirma zu entrichten.

Das Abändern und Rekompilieren sowie das Weitergeben und Vermieten der Software ist nicht möglich.

Vor und Nachteile

Der Cult3D Exporter und der Designer bieten im Zusammenwirken die Möglichkeit der 3D-Präsentationserstellung. Dabei fokussiert sich das Hauptaugenmerk von Cult3D auf die Interaktionsintegration. Das Modellieren und Texturieren sowie die Erstellung von festen Animationen wird nicht durch diese Werkzeuge realisiert, sondern erfolgt durch Modellierungssoftware wie Maya oder 3D Studio Max. Dies hat zur Folge, dass die Erstellung einer 3D-Szene nur mittels zwei verschiedener Werkzeugen durchgeführt werden kann. Der Exporter bildet dabei das Bindeglied zwischen Aufbau und Interaktionsintegration. Im Designer von Cult3D stehen dem Nutzer viele Komponenten zur Definition von Aktionen zur Verfügung. Ebenso ist die Übersichtlichkeit über die gesamte Szene durch verschiedene frei skalierbare Fenster jederzeit gegeben. Durch das Vorschauenfenster innerhalb der Anwendungen ist es möglich, jederzeit den aktuellen Status abzufragen und zu kontrollieren.

Zur Definition von Events steht mit der Event Map ein Werkzeug zur Verfügung, mit dem der Nutzer anschaulich und nachvollziehbar Aktionen erstellen kann. Die Übersichtlichkeit kann bei der Erarbeitung komplexerer Eventgraphen jedoch verloren gehen.

Durch die Lizenzbestimmungen ist es, ohne Entrichtung von Lizenzgebühren, nicht möglich die Software kostenfrei zur Erstellung und Publikation von 3D-Präsentationen zu verwenden.

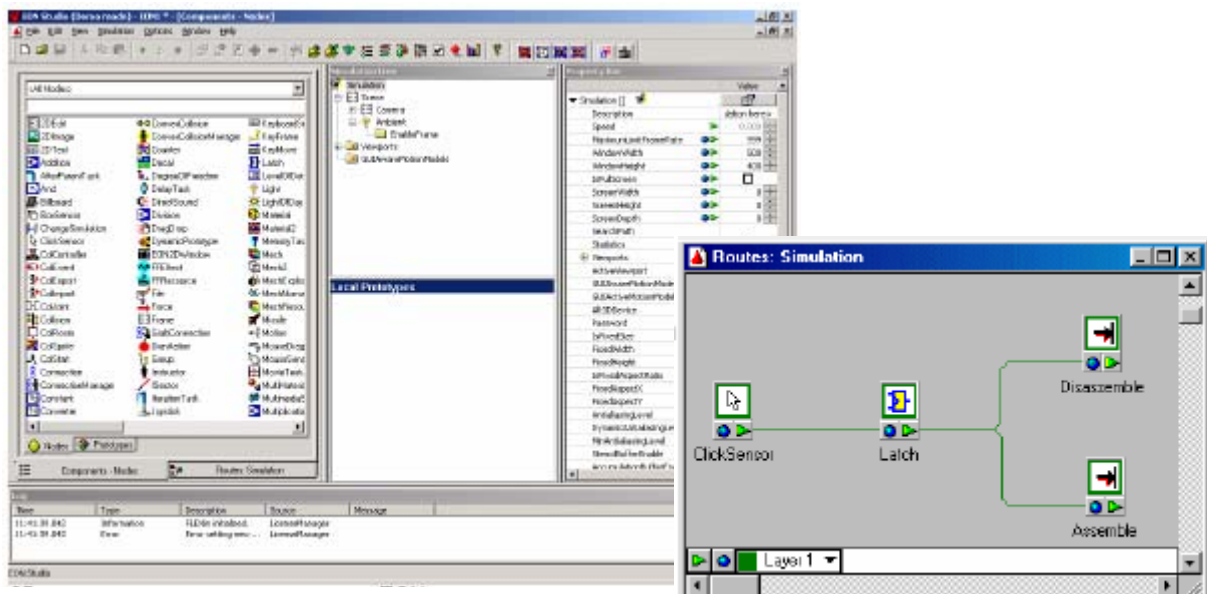
1.2 EON Studio

EON Studio ist ein Autorenwerkzeug zur Entwicklung von interaktiven 3D-multimedia Echtzeitapplikationen auf der Basis des Windows Betriebssystems. Das Werkzeug basiert auf dem VRML Standard. Der Entwickler, EON Reality, legt bei der Entwicklung der Software seine Schwerpunkte auf Erstellung

von interaktiven 3D-Welten und Produktpräsentationen. Darüber hinaus kommt EON Studio bei der Entwicklung von Trainingsszenarios und virtuellen Simulationen zum Einsatz.

Der erste Schritt bei der Erstellung einer 3D-Szene mit EON Studio ist der Import von 3D Geometrien aus Modellierungswerkzeugen wie Alias Maya oder 3D Studio Max. EON Studio kann aber auch Daten aus CAD Programmen weiterverarbeiten. Nach dem Import der Rohdaten erfolgt das Hinzufügen von Objektverhalten. Dies kann zum einen über die grafische Benutzeroberfläche des Autorenwerkzeugs erfolgen zum anderen können Verhalten mittels JavaScript oder VBScript definiert werden. C++ Elemente können mit dem EON SDK ebenfalls in die Präsentation integriert werden. Nach Fertigstellung der 3D-Szene mit allen Animationen und Interaktionen kann die Szene nach dem Export aus dem Werkzeug mit dem ebenfalls von EON Reality entwickelten EON Viewer visualisiert und im Netz zur Verfügung gestellt werden. Beim Export von 3D-Szenen aus dem EON Studio werden edz-Dateien erzeugt.

Die EON Studio Oberfläche besteht aus diversen Fenstern, die verschiedene Ansichten der 3D-Szene darstellen. Das Fenster des „Simulation Tree“ ist eine grafische Repräsentation des Szenengraphen der 3D-Präsentation. Der Simulation-Knoten bildet den Wurzelknoten der gesamten Präsentation und der Scene-Knoten den des Szenengraphen. Dieser kann im Simulation Tree per Drag' n' Drop situationspezifisch angepasst werden. Weiterhin werden Knoten für Kameras, Beleuchtung und Animationen angezeigt. Neben dem „Simulation Tree“ ist das „Component Window“ ein weiteres zentrales Fenster des EON Studios. Es beinhaltet Komponenten um die 3D-Szene mit Interaktionen zu versehen und um sie zu erweitern und auszubauen. Das „Component Window“ unterscheidet dabei drei Knotentypen. Die Agent-Nodes, als erste Kategorie, dienen der Initialisierung von Operationen. Neben den Agent-Nodes sind Animation-Nodes und Sensor-Nodes über das „Component Window“ auswählbar. Die Ansicht des Routes Window dient der Definition von Routes. Mittels dieser wird Objektverhalten definiert. Dies geschieht über die Verknüpfung von Ausgangsfeldern des Startknotens mit den Eingangsfeldern des Endknotens der Verbindung. Die Routeerstellung wird über einen Graphen visualisiert. Es besteht die Möglichkeit verschiedene Ebenen zu definieren, um eine bessere Übersicht der bestehenden Verknüpfungen zu gewährleisten. Neben diesen Hauptfenstern sind für den Nutzer weitere Werkzeuge zur Szenenbearbeitung verfügbar. So können Knotenattribute für Animationen und Interaktionen gesetzt werden oder für einzelne Knoten die Gesamtheit aller Verbindungen angezeigt werden. Durch die Verknüpfung von Einzelabläufen kann eine Abfolge von Aktionen definiert werden. Dadurch ist es dem Anwender möglich einen Szenenablauf innerhalb einer Präsentation zu erstellen.



EON Studio Oberfläche und Routes Window

Lizenzbestimmung

EON Studio als eine Komponente eines großen Produktkataloges der Firma EON Reality ist nicht frei von Lizenzgebühren. Es besteht lediglich die Möglichkeit eine Testvariante der neuen Version für zwei Wochen unentgeltlich zu nutzen. Der Download erfolgt nach erfolgreicher Registrierung bei EON Reality. Eine weitere Variante die Software zu verwenden ergibt sich durch den Download einer Vor-

gängerversion des Produktes über einen nicht begrenzten Zeitraum. Der Funktionsumfang dieser Freeware ist eingeschränkt, wodurch ein Speichern von erstellten Projekten und Szenen nicht möglich ist. Um eine, mit EON Studio erstellte Szene abspeichern zu können benötigt man einen Aktivierungscode. Dieser ist erst nach dem Kauf der Software erhältlich. Das Produkt ist unter der Adresse <http://www.eonreality.com/developer/> erhältlich.

Neben den eingeschränkten Nutzungsrechten sind auch die Weitergabe und die Nutzung für nicht private Zwecke untersagt.

Vor und Nachteile

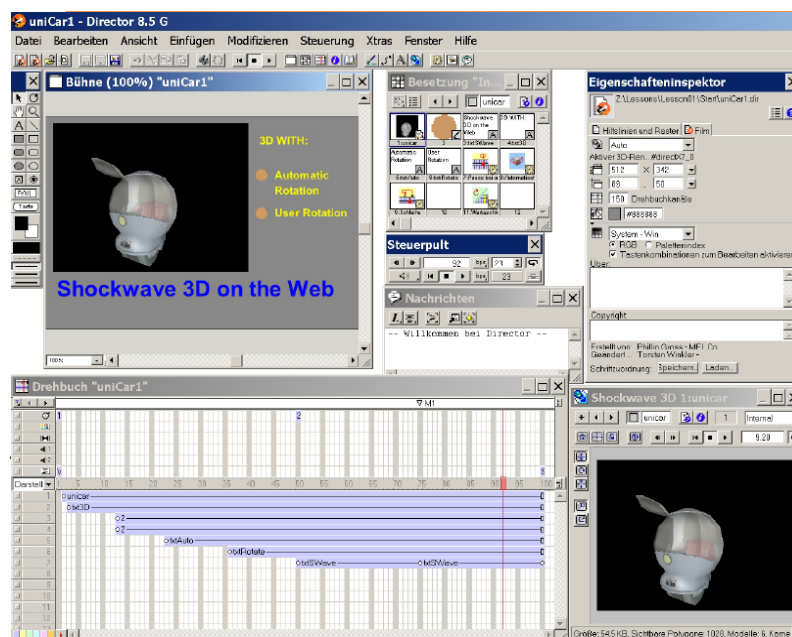
EON Reality stellt ein umfassendes Softwarepaket für den Nutzer bereit. Neben dem EON Studio stehen noch weitere Werkzeuge für den Import und Export von Dateien sowie die Bereitstellung von Daten zur Verfügung. Mit dem EON Studio ist eine Erstellung von 3D-Präsentationen leicht und umfassend möglich. Es bietet eine Fülle von Erstellungsmöglichkeiten um eine 3D-Szene mit Interaktionen und Features wie die Erzeugung von Walkthrough-Szenarien oder komplexen Lichtberechnungen zu versehen. Die vielfältigen Importmöglichkeiten im EON Studio bietet dem Nutzer die Möglichkeit interaktive 3D-Szenen aus vielen verschiedenen Rohdaten zu verwenden. Dadurch können Daten aus ArchiCAD genauso verwendet werden wie Szenen aus Maya oder Studio Max. Aus dieser Vielseitigkeit folgt aber eben auch die Verwendung von mehreren Softwarekomponenten zur Erzeugung der Präsentation. Weiterhin ist es nicht möglich die Software frei und ohne finanziellen Aufwand zu nutzen.

1.3 Macromedia Director

Macromedia Director (kurz: Director), ist ein Autorensystem zur Erstellung komplexer, interaktiver, multimedialer Internet-, DVD-, CD-ROM- und Kiosk-Anwendungen. Es verfügt über die spezifische objektorientierte Programmiersprache Lingo, unterstützt JavaScript und integriert eine Vielzahl verfügbarer Medienstandards. Mit Director ist es möglich Programme für Windows und Mac OS zu erzeugen. Mit dem Shockwave Plug-in können Director-Anwendungen über das Internet gestreamt und in Webbrowsern abgespielt werden. Darüber hinaus ermöglicht Macromedia Director seit der Version 8.5 3D-Objekte zu importieren sowie zu erstellen und diese in Echtzeit zu steuern.

Konzepte

Macromedia Director dient der Erstellung dynamischer Multimediaanwendungen. Folgende Abbildung zeigt beispielhaft die Oberfläche des Werkzeugs während des Erstellungsprozesses einer 3D-Szene.



Oberfläche von Macromedia Director 8.5

Anhand einer Zeitachse wird der Ablauf der Szene organisiert. Director spricht bei einer Anwendung von einem Film. Dieser wird auf der Bühne, dem Vorschaufenster, des Werkzeugs, präsentiert. Die Grundelemente des Films sind die Darsteller. Darsteller können Text, Grafiken, Bilder oder Sounds sowie 3D-Objekte sein. Diese können als Instanzen, so genannte Sprites, der Darsteller auf der Bühne platziert werden und sind damit in der Präsentation sichtbar. Um die korrekte Anordnung der Sprites auf der Bühne zu gewährleisten, erhalten diese zusätzliche Eigenschaften wie Position oder Farbe. Ein weiteres Sprite-Attribut ist die Puppet Eigenschaft. Sie gibt Auskunft darüber, ob der Sprite über ein Lingo-Skript gesteuert wird.

Über die Bedienoberfläche des Drehbuchs, in der Oberfläche der Anwendung, wird das zeitliche Zusammenspiel verschiedener Medienobjekte organisiert. Das Drehbuch selbst ist dafür in bis zu 1000 nummerierte Kanäle unterteilt. Durch diese können Sprites zeitlich organisiert werden. Zusätzliche Spezialeffektkanäle steuern Tempo, Sounds und Übergänge der Anwendung.

Skriptsprache Lingo

Lingo ist die Programmiersprache des Autorenwerkzeugs. Die Lingo-Programme werden vom Laufzeitsystem interpretiert. Mittels Lingo können Ereignishandler definiert werden. Diese bestimmen ein Verhalten das einem Objekt, und damit einem Frame oder Sprite, zugeordnet werden kann. Das Lingo Skript wird dabei als Verhaltensskript bezeichnet. Ein Ereignishandler kann sich aber auch auf den gesamten Film beziehen und wird dann als Filmskript betitelt. Handler werden bei einer Aktion wie beispielsweise einem Mausklick ausgelöst und daraufhin ausgeführt. Die Grundsätzliche Form eines Lingo-Skript veranschaulicht die folgende Code-Zeile:

```
on EREIGNIS; BEFEHL(E) ...; end
```

Lingo besitzt als Programmiersprache objektorientierte Ansätze und kann dank stetiger Weiterentwicklung mittlerweile als objektorientierte Sprache bezeichnet werden. Seit Director MX2004 wird auch JavaScript (ECMA-Script) als Sprache unterstützt. Lingo und Java Skripte können im selben Programm oder Skript verwendet werden. Durch JavaScript lassen sich im Vergleich zu Lingo einige Befehlsfolgen eleganter schreiben. Darüber hinaus sind durch JavaScript einige neue Funktionen hinzugekommen.

Director3D

Zur Erzeugung von 3D-Szenen arbeitet Director mit 3D Darstellern. Diese enthalten eine komplette 3D Welt. Ein Sprite eines solchen Darstellers stellt dabei eine bestimmte Kamerasicht der 3D Welt dar, bestehend aus Objekten der Typen: Modell, Modellressource, Shader, Textur, Bewegung, Licht, Kamera und Gruppe. Primitive als einfache Modellressourcen sind in Director vordefiniert und können in die 3D-Szene integriert werden. Zur Erstellung einer komplexeren Szene bedarf es der Hilfe einer 3D Modellierungssoftware über deren Export die Daten in Director importiert und zur Verfügung gestellt werden. Die Bestandteile eines 3D-Darstellers sind in Nodes organisiert und in Parent-Child Hierarchien strukturiert. Diese sind beim Import einer 3D-Szene bereits festgelegt, können aber in Director mittels Lingo-Skripte neu definiert werden. Die Definition von Interaktionen und Animationen der Objekte erfolgt in Macromedia Director hauptsächlich über Lingo-Skripte. Dabei wird bei einem zu definierenden Verhalten zwischen Auslöser und Aktion unterschieden. Die Aktionen lassen sich folgenden Typen zuordnen: Lokale Aktionen sind einem bestimmten Sprite zugeordnet und legitimieren nur Auslöser von diesem. Öffentliche Aktionen können dagegen von einem beliebigen Sprite ausgelöst werden. Der dritte Typ sind unabhängige Aktionen die keinen Auslöser benötigen. Das von diesen Typen ausgelöste Verhalten wird durch Lingo modelliert. Vorgefertigte Lingo-Skripte bilden eine Verhaltensbibliothek auf die der Nutzer zurückgreifen kann um den Programmieraufwand zu minimieren. Zu lokalen Aktionen können Verhalten wie Modellrotation oder Transformation sowie Wiedergabe von Animationen gehören. Öffentliche Aktionen lösen Verhalten wie Kamerabewegungen aus und automatische Aktionen führen beispielsweise zu permanenter Drehung eines Objekts. Auslöser einer Aktion können ein Tastendruck oder Mausklick des Nutzers sowie interne Vorgänge und Abhängigkeiten sein.

Lizenzvereinbarung

Die Firma MacroMind wurde 1984 von Marc Canter, Jay Fenton und Mark Pierce gegründet. Diese entwickelte für den Apple Macintosh die Animationssoftware VideoWorks, welche zwei Versionen später in Director umbenannt wurde. Durch den Zusammenschluss mit der Firma Authorware entstand Macromedia.

Unter dem Dach von Macromedia wurde Director bis zum Dezember 2005 kontinuierlich weiterentwickelt. Ab diesem Zeitpunkt wurde Macromedia von Adobe Systems komplett übernommen. Director erscheint auch dort im Produktportfolio.

Um Macromedia Director nutzen zu können bedarf es einer Lizenz, die käuflich erworben werden muss. Macromedia bietet separat eine Lizenz für allgemein- und berufsbildende Schulen an. Der Preis der Lizenz ist von der Anzahl der Nutzer abhängig. Auf der Internetseite der Entwicklerfirma wird eine 30-Tage Shareware angeboten. Diese Version umfasst den gesamten Funktionsumfang des Produktes, ist aber ausschließlich für private Zwecke bestimmt. Die Freischaltung der Shareware erfolgt über einen Aktivierungsschlüssel der nach dem Erhalt der Lizenz zur Verfügung gestellt wird.

Vor- und Nachteile

Macromedia Director findet als Werkzeug für Multimediaanwendungen weite Verbreitung. Durch Director3D bietet Macromedia die Möglichkeit 3D-Präsentationen zu erstellen. Die integrierten 3D-Primitive reichen jedoch nicht zur Erstellung einer umfangreicheren Präsentation aus. Aufgrund dessen müssen die 3D-Objekte und Animationen aus einer separaten Modellierungssoftware heraus importiert werden. Dies hat ein Arbeiten mit 2 Werkzeugen, die synchronisiert werden müssen, zur Folge. Ein erzeugter 3D-Darsteller bildet einen einzelnen Sprite im Drehbuch und schränkt damit die Möglichkeiten die sich durch das Arbeiten mit dem Drehbuch hinsichtlich der zeitlichen Abfolge der Szenenkomponenten ergibt, weitgehend ein.

Das Hinzufügen von Interaktionen hat das Programmieren von Skripten zur Folge. Diesen Nachteil kann die von Director bereitgestellte Aktionen- und Verhaltensbibliothek nur teilweise kompensieren. Der Nutzer muss bei der Erstellung einer 3D-Präsentation in der Lage sein Lingo- oder JavaScript zu programmieren. Macromedia Director ist kein Open-Source Projekt und es bedarf den Erwerb entsprechender Lizenzen für die Nutzung des Werkzeugs. Dies hat Einschränkungen beziehungsweise Verbote hinsichtlich Weitergabe und Abänderungen des Programmcodes zur Folge und natürlich hinsichtlich der Veröffentlichung von 3D-Präsentationen.

1.4 RawKee – X3D Plug-in für Alias Maya

RawKee ist der Name eines Open Source X3D Plug-ins für Maya. Es ist sowohl für Maya Complete als auch für Maya Unlimited verfügbar. Es wird vom Archaeology Technologies Laboratory (ATL) an der North Dakota State University entwickelt. Der Projektstart erfolgte im November 2003 und die Software ist seit März 2006 in der Version 1.0 verfügbar. Die Entwicklung begann mit dem Ziel ein Werkzeug zu erstellen, mit dem archäologische und anthropologische Inhalte einfach zugänglich und für jedermann verfügbar sind. Durch die Umsetzung dieser Software wurde es möglich Explorationen und Untersuchungen an archäologischen Fundstücken nicht mehr direkt, sondern mittels eines digitalen Abbildes durchzuführen.

Das Plug-in kann in die 3D-Modellierungssoftware Maya eingebunden werden und den gesamten Prozess der Erstellung von 3D-Präsentationen verwirklichen. Es baut dafür auf den bestehenden Funktionen und Möglichkeiten von Maya auf und die zur Erzeugung der Präsentation fehlenden Komponenten werden mittels des Plug-ins eingebunden.

Installation

RawKee wird in den Maya Versionen 4.5 – 5.0 für Windows 2000 und Windows XP unterstützt.

Bei den nachfolgenden Maya Distributionen, derzeit 6.0, 6.5 und 7.0 wird das Plug-in neben Windows 2000 und XP gleichfalls unter Mac OSX und Linux bedient.

Um RawKee nutzen zu können muss das Plug-in zuvor von der Internetseite https://sourceforge.net/project/showfiles.php?group_id=116002 heruntergeladen und installiert werden. Auf dieser Internetseite besteht die Möglichkeit für die verschiedenen Maya Versionen und Betriebssysteme die passende RawKee Version zu wählen. Darüber hinaus stehen noch verschiedene Installationsvarianten zur Verfügung. Man kann das Plug-in mittels einer Installationsroutine oder durch manuelles Erzeugen der entsprechenden Verzeichnisstruktur in der Maya Software verfügbar machen. Die folgenden Erläuterungen beschränken sich auf die Installation mittels Routine und dies im Speziellen unter Windows. Bei den aktuellen RawKee Versionen ist bei der Installation lediglich auf die Korrektheit des Installationspfades zu achten. Dieser sollte mit der Maya Installation übereinstimmen. Bei älteren Versionen ist die Vorgehensweise bei der Installation different. Die Installation sollte in einen Ordner erfolgen, der sich im gleichen Pfad wie die Maya-Installation befindet. Bei diesem Vorgang wird ein

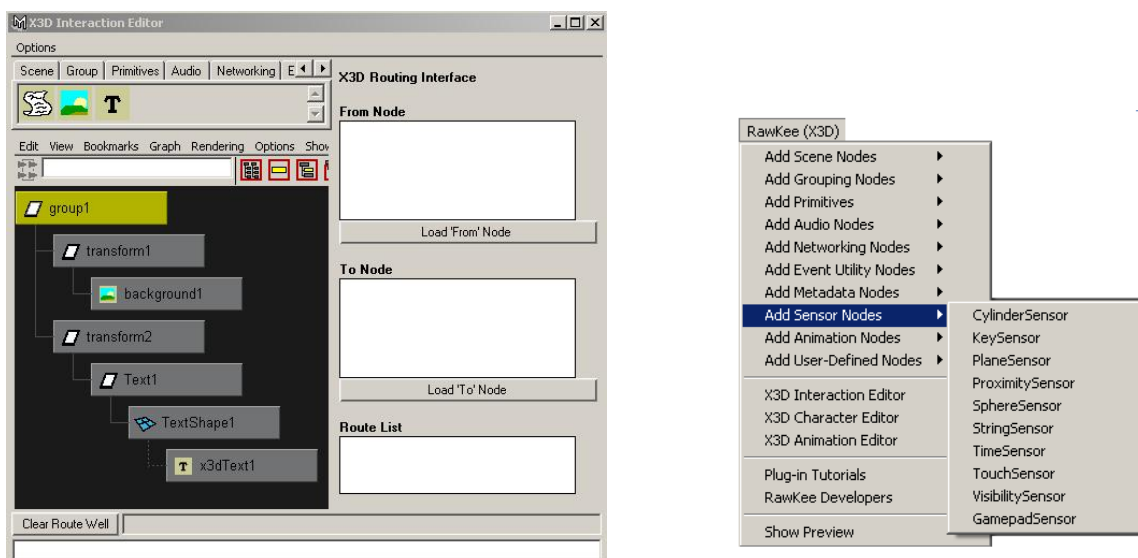
separater RawKee Ordner erstellt. Zusätzlich muss die bei der Maya-Installation erzeugte Datei maya.env durch die gleichnamige Datei aus dem RawKee Ordner ersetzt werden. Dadurch wird die Bereitstellung der RawKee Daten realisiert.

Daraufhin ist das Plug-in in Maya verfügbar. Es kann durch den Plug-in Manager in Maya eingebunden werden. Die Einkopplung erfolgt unter dem Menüpunkt Window > Settings/Preferences > Plug-in Manager... Im Menüfenster kann durch auswählen der Datei x3d.dll das Plug-in integriert werden. Ebenso ist es möglich eine automatische Initialisierung während des Startvorgangs der Software zu definieren. Über den soeben beschriebenen Weg kann das Plug-in auch wieder entkoppelt werden.

Funktionen und Funktionsweise

Der in der Menüleiste von Maya erschienene Eintrag des RawKee Plug-ins beinhaltet 5 Untergliederungen. Die erste beherbergt die „Addition Area“ in dem durch Auswahl der Unterpunkte Knoten im Dependency Graph (Szenengraph), und damit ebenfalls auf der Bühne, erzeugt werden können. Die einzelnen Menüeinträge sind nach Kategorien geordnet. Der zweite Unterbereich ist die „Editor Area“. Diese ist unterteilt in Interaction-Editor, Animation-Editor und Character Editor. Der bedeutendste Eintrag ist hierbei der des Interaction-Editors. In ihm können ebenso alle verfügbaren Knoten erzeugt werden und er bietet darüber hinaus die Darstellung der aktuellen Szene als Graph. Weiterhin erfolgt in ihm die Definition der Routes. Über diesen Editor ist es dem Anwender möglich alle Komponenten die das Plug-in unterstützt direkt anzusprechen und zu verwenden. Der Interaction-Editor ermöglicht es damit eine komplette 3D-Szene zu erstellen und diese mit Animationen und Interaktionen zu versehen. Die Animationserstellung kann ebenso über den Animations-Editor erfolgen. In ihm können Translations-, Rotations- und Skalierungsanimationen definiert werden. Der Character-Editor ermöglicht Charaktererstellung und Animation für den HAnim-Export. Die unter dem Menüpunkt „Utility-Commands“ befindlichen Funktionen dienen der Aufbereitung der Szene für den Export in das VRML- oder X3D-Format sowie die Definition von Multi-Texturen. Im darunter liegenden Bereich des RawKee-Menüs stehen dem Nutzer Hilfen zur Verfügung. Über diese Einträge ist es möglich auf Plug-in Tutorials zuzugreifen oder die Entwickler zu kontaktieren. Der letzte Bereich realisiert eine Vorschauansicht der Präsentation. Beim Aufruf des Eintrags „Show Preview“ wird in einem Browserfenster die aktuelle Szene dargestellt.

Die in Maya erstellte Szene kann als Gesamtheit exportiert werden. Ebenso steht es dem Benutzer frei nur die von ihm selektierten Teilbereiche einer Szene dem Export zu übergeben. RawKee unterstützt mit Classic, XML und Binary drei verschiedene Exportvarianten. Neben X3D wird auch der Export von VRML97-Dokumenten unterstützt. Während des Exportvorgangs wird dieser über das Output-Window von Maya protokolliert. Die nachfolgenden Abbildungen zeigen Ausschnitte aus der Plug-in Oberfläche.



Interaktions-Editor und Menü des RawKee Plug-in

Der erste Schritt bei der Entwicklung einer 3D-Präsentation mittels Maya ist die Erstellung der Objekte in der Szene. Dieser Teilschritt kann sowohl mit den Maya-internen als auch mit denen im RawKee Plug-in

bereitgestellten Funktionen erfolgen. Daraufhin können die Animationen hinzugefügt werden. Diese werden auf Basis der Keyframe Animation in Maya erstellt. Die vom Nutzer erzeugten Animationen werden dann mit Hilfe von Interpolatoren und Time-Sensoren für den Export in die X3D-Szene erweitert. Der vorletzte Schritt ist das Hinzufügen von Interaktionselementen in die Szene. Für einen fehlerfreien Ablauf einer Präsentation sind die Objekte, die Animationen und die Interaktionskomponenten zu verbinden. Dazu müssen als letzter Schritt Routes definiert werden. Diese legen die Event-Architektur der 3D-Szene fest.

Lizenzvereinbarung

Das RawKee Plug-in unterliegt der GNU Lesser General Public Licence (LGPL). Diese Lizenz erlaubt es die Software für einen beliebigen Zweck zu verwenden, sie zu vervielfältigen und weiterzugeben. Es ist ebenso möglich die Software zu ändern und geänderte Versionen weiterzugeben. Die veränderten Versionen müssen jedoch ebenfalls nach der LPGL lizenziert werden. Programme, die eine LPGL lizenzierte Software verwenden, dürfen diese nur extern in Form von zum Beispiel Bibliotheken verwenden.

Vor- und Nachteile

Dieses Maya Plug-in bietet dem Nutzer die Möglichkeit den gesamten Erstellungsprozess einer 3D-Präsentation in Maya durchzuführen. Es können Objekte erzeugt und animiert und nutzer- sowie systemgesteuerte Aktionen definiert werden. Eine Präsentation kann aus mehreren Teilszenen bestehen die mittels Mausinteraktion gesteuert werden. Es ist jedoch noch nicht möglich die Abfolge von Aktionen mittels freiwählbarer Tastatureingaben zu definieren. Dies ist auf den derzeit nicht abgeschlossenen Entwicklungsstand des X3D-Formats sowie der X3D-Player zurückzuführen. RawKee bindet sich sehr gut in die bestehende Maya Umgebung ein und bietet unter zu Hilfenahme des Dependency Graphs eine Übersicht über den strukturellen Aufbau der Szene. Die bereitgestellten Editoren ermöglichen einfache Erstellungen von Animationen und Routes. Bei der Erstellung von Objekten ist jedoch die Nutzung von NURBS nur bedingt möglich. Alle NURBS Flächen müssen vor dem Export in Polygone konvertiert werden, da das X3D-Format NURBS nicht unterstützt. Die Übersichtlichkeit im Dependency Graph bei komplexeren Szenen kann bei schlechter Wartung schnell verloren gehen. Aufgrund dessen ist eine disziplinierte Szenenerstellung unumgänglich.

Der gesamte Prozess von der Erstellung bis zur Präsentation ist, bis auf eine Ausnahme, durch Open Source Projekte realisierbar. Die Ausnahme besteht in der zugrunde liegenden Software. Die 3D-Modellierungssoftware Maya unterliegt Lizenzbedingungen und ist nicht frei verfügbar. Das RawKee Plug-in ist durch die LPGL für jeden Nutzer frei verwend- und erweiterbar. Der Export aus Maya erfolgt in das offene X3D Format und kann mit freier X3D Abspielsoftware wie dem Xj3D-Player visualisiert werden. Die in Maya erstellte Szene kann in einem X3D Browser anders dargestellt werden als dies in der internen Maya Szenenvorschau der Fall ist. Aufgrund dessen ist die Vorschau mittels RawKee Plug-in unumgänglich. Das RawKee Plug-in ist damit ein kompaktes Werkzeug zur Präsentationserstellung wobei die Leistungsfähigkeit durch das X3D-Format und die Unübersichtlichkeit begrenzt ist.

2 Beurteilung bestehender Erstellungsoftware

Diese Beurteilung berücksichtigt die Aufgabenstellung. Daraus resultieren spezielle Betrachtungsweisen auf die in diesem Dokument vorgestellten Werkzeuge zur Präsentationserstellung. Die Schlussfolgerungen orientieren sich an dem Ziel der Erstellung einer 3D-Präsentation als Hilfsmittel zu einer Vorlesung im universitären Bereich. In dieser Beurteilung werden ausschließlich die vorgestellten Erstellungswerkzeuge betrachtet.

Die mit den Anwendungen erstellten 3D-Szenen sollten, um einen Ablauf einer Situation oder eines Vorgangs beschreiben zu können, aus mehreren Teilszenen bestehen. Es sollten Animationen und Interaktionen in die Szenerie integriert werden können. Voraussetzung für die Arbeit mit den Werkzeugen ist ebenso eine einfache Bedienbarkeit der Software ohne Programmieraufwand sowie eine effizient nutzbare Bedienoberfläche. Eine 3D-Präsentation sollte in möglichst kurzer Zeit erzeugt werden können.

Mit allen untersuchten Werkzeugen ist die Erstellung einer 3D-Szene, unter Einbeziehung der beschriebenen Kriterien, mit Einschränkungen möglich. Jedoch sind unterschiedliche Ansätze und Aktivitäten notwendig um diese zu realisieren.

Die Werkzeuge Cult3D, EON Studio und Macromedia Director beschränken sich in ihrem Wirkungskreis weitestgehend auf den Teilbereich der Interaktionsintegration in die Szene. Jede dieser Anwendungen verlangt den Import von Daten. Die 3D-Objekte und fertigen Animationen müssen aus Modellierungsprogrammen wie Alias Maya oder 3D Studio Max importiert werden, um diese dann mittels des Werkzeugs zu einer Präsentation zu komplettieren. Der Vorteil des Nutzers liegt dabei in der Entscheidungsfreiheit über das Modellierungswerkzeug. Der Anwender hat die Möglichkeit Software zur Erzeugung zu nehmen, die auf den jeweiligen Fall der Szenenerstellung zugeschnitten ist. Aufgrund dessen ist eine detaillierte Vorausplanung unumgänglich. Das Hinzufügen von Objekten während der Einarbeitung von Interaktionen hat den erneuten Import der Rohdaten zur Folge. Dieser Nachteil besteht beim RawKee Plug-in nicht. Durch die direkte Integration in die Modellierungssoftware kann der Modellierungsprozess, die Animations- und Interaktionserstellung quasiparallel stattfinden. Durch die nahe Bindung an die Modellierungssoftware ist Unterstützung anderer Modellierungswerkzeuge jedoch nicht möglich. Macromedia Director bietet ebenso wie das RawKee Plug-in die Möglichkeit 3D-Objekte im Werkzeug zu erstellen und Interaktionen hinzuzufügen. Die Erstellung einer Präsentation hat aber im Macromedia Director Programmieraufwand zur Folge. Trotz der bestehenden Bibliothek ist das Schreiben eigener Skripte oder das modifizieren Bestehender unumgänglich.

Übersichtlichkeit und einfache Bedienbarkeit ist für ein Werkzeug von zentraler Bedeutung. Bei Cult3D und EON Studio ist dies gegeben. Durch verschiedene Fenster hat der Anwender zu jedem Zeitpunkt den Überblick über den aktuellen Entwicklungsstand. Die Übersichtlichkeit leidet bei Cult3D etwas, insbesondere bei größeren Projekten, unter der Visualisierung der Komponentenbeziehungen mittels Graphen. EON Studio überzeugt mit seiner Vielseitigkeit und der großen Anzahl an Optionen die dem Nutzer zur Verfügung gestellt werden. Director wurde für die Entwicklung von Multimediaanwendungen im Internet konzipiert. Director3D wurde in die Software als zusätzliches Modul integriert. Die Oberfläche ist daher nicht für diese Art von Präsentationen entworfen. Dies führt zu einem eher schwierigen Arbeiten mit diesem Werkzeug. Die Arbeit mit dem RawKee Plug-in setzt Grundkenntnisse mit Maya voraus. Die Oberfläche ist der von Maya angepasst und auch die Visualisierungen sind ähnlich. An manchen Stellen ist die Bedienung jedoch nicht intuitiv und bedarf einer Einarbeitung. Die Erstellung einer Präsentation mit Cult3D, EON Studio und RawKee ist mit einem angemessenen Zeitaufwand möglich. Der Erstellungsaufwand in Director ist aufgrund der teils fehlenden Übersichtlichkeit und des Programmieraufwandes höher.

RawKee ist im Gegensatz zu allen anderen Werkzeugen ein Open Source Projekt und ist damit kostenlos verfügbar. Es besteht, aus der Lizenzvereinbarung resultierend, die Möglichkeit die Software den gegebenen Ansprüchen anzupassen und das Werkzeug damit für den Nutzer zu optimieren. Dies ist aufgrund der Lizenzbestimmungen der anderen Werkzeuge nicht möglich.

Zum Abspielen der Präsentationen bedarf es bei Cult3D, EON Studio und Director spezieller von den Softwareentwicklern bereitgestellten Wiedergabeprogrammen beziehungsweise Browser Plug-ins. Aufgrund der speziellen Dateiformate dieser Werkzeuge werden diese expliziten Abspielsoftware vorausgesetzt. RawKee Exporte basieren auf dem offenen X3D-Format. Derzeit gibt es eine Vielzahl von Playern die dieses Format unterstützen. Einige von ihnen sind Open Source Projekte und damit kostenlos verfügb- und nutzbar.

Die Entscheidung fiel zugunsten des RawKee Plug-ins aus. Die Erstellung mit nur einem Werkzeug und der Open Source Status geben dabei den Ausschlag. Das Plug-in unterliegt einer permanenten Weiterentwicklung und soll im Verlauf dieses Beleges den Anforderungen angepasst werden. Im Kapitel IV wird auf Einzelheiten des Plug-ins eingegangen und Kritikpunkte bezüglich des Werkzeuges analysiert.

3 Software zur Präsentationsdarbietung

3.1 Cult 3D Viewer

Der Cult3D Viewer ermöglicht die Darbietung von, im Cult3D Designer erstellten, 3D-Präsentationen. Die Abspielsoftware steht für alle gängigen Plattformen ohne Kostenaufwand zur Verfügung. Durch den Cult3D Viewer können 3D-Szenen im Internet präsentiert werden. Darüber hinaus stehen Plug-ins für Microsoft Office und Adobe Acrobat zur Verfügung. Dadurch können Präsentationen direkt in Dokumente integriert und in diesen präsentiert werden.

3.2 EON Viewer

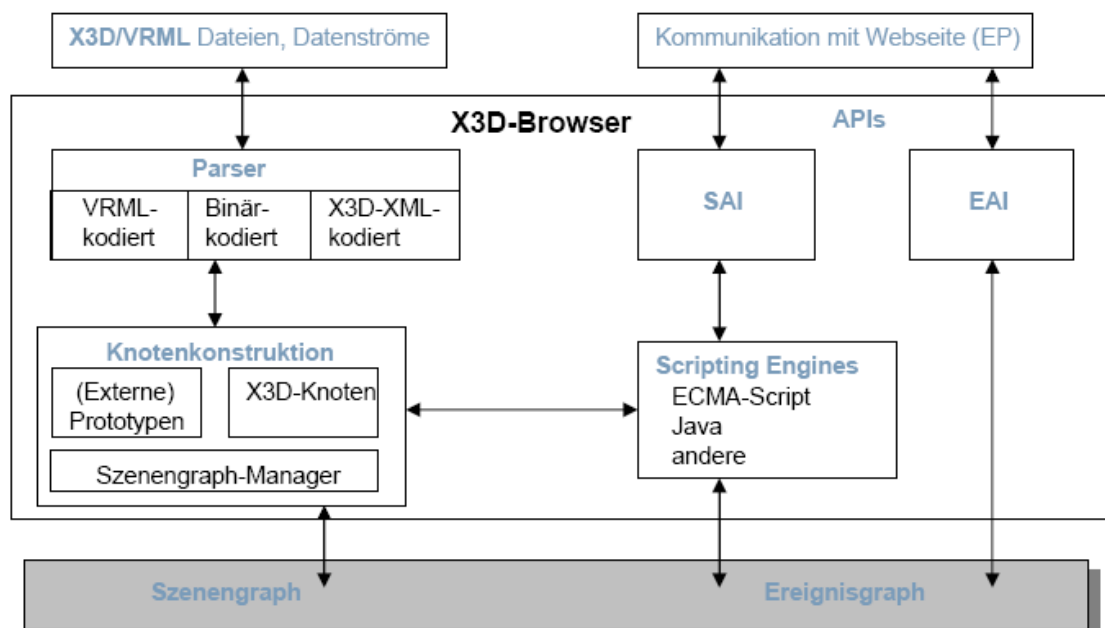
Durch den EON Viewer ist die Visualisierung von EON Präsentationen möglich. Die Software ist kostenfrei erhältlich. Der Player verwendet EonX, ein ActiveX-Control, zur Darstellung der Präsentation. Im EON Viewer kann mit der 3D-Szene interagiert werden. Der Player kann Echtzeit-Reflexionen, Schatten und environment mapping darstellen. Neben den gängigen Anwendungsszenarien werden auch fly-through Szenaren umgesetzt. Der EonX Plug-in zur Darstellung von Applikationen im Web-Browser ist für Internet Explorer und Netscape verfügbar. Eine Standalone Anwendung wird ebenfalls angeboten.

3.3 Shockwave-Player

Shockwave bezeichnet eine Software-Technologie des Herstellers Adobe für die Darstellung interaktiver Inhalte im Internet. Directors Shockwave Dateien haben die Dateiendung .dcr. Multimediale, interaktive Shockwave-Applikationen lassen sich mit diesem Plug-in direkt im Browser abspielen. Eine Director Anwendung kann auch als ActiveX Control zum Beispiel in eine Power Point Präsentation integriert werden. Das Plug-in für Shockwave kann kostenfrei bei Adobe heruntergeladen werden und lässt sich in zahlreichen Browsern installieren. Es steht jedoch kein Plug-in für Linux zur Verfügung.

3.4 VRML/X3D-Player

Eine Vielzahl von VRML-Playern ist derzeit verfügbar. Einige von ihnen unterstützen neben VRML dessen Nachfolger X3D. Die Software ist jedoch nur teilweise unentgeltlich. Mit dem Xj3D-Player beispielsweise, steht dem Nutzer ein Open Source Projekt zur freien Verfügung. Die verschiedenen Player unterscheiden sich hauptsächlich in ihren Visualisierungsmöglichkeiten und in der Navigation. Die grundlegende Funktionsweise ist jedoch gleich. Der Browser nimmt den Datenstrom der Datei auf und parst diesen, um eine Abbildung der Knotenkonstruktion zu erstellen und die Hierarchien zwischen den Elementen zu analysieren. Aus den gewonnenen Informationen kann der Szenengraph erstellt werden. Mit einer Scripting Engine werden die ECMA- beziehungsweise Javaskripte ausgewertet. Skript Knoten erweitern die Darstellungs- und Interaktionsmöglichkeiten des X3D-Formats. Zusammen mit den analysierten Knoten der Szene kann der Ereignisgraph der Präsentation erstellt werden. Über das Script Authoring Interface (SAI) werden dem Browser die in den Skript Knoten befindlichen und mittels Interpreter ausgewerteten Programmteile zugeführt. Externe Skripte werden über das External Authoring Interface (EAI) eingebunden. Somit stehen dem Browser der Szenengraph und der Ereignisgraph zur Verfügung und können in diesem umgesetzt werden. Das folgende Schema verdeutlicht dies.



4 Beurteilung bestehender Darbietungssoftware

Der Vergleich der untersuchten Abspielsoftware richtet sich nach den betrachteten Erstellungswerkzeugen. Cult3D, EON Reality und Macromedia Director bieten ihre eigens entwickelten Player zum Download an. Diese Viewer stellen nur die im Erstellungsprozess des jeweiligen Werkzeuges entstandenen Dateiformate dar. Deshalb ist der Vergleich zwischen diesen Softwareprodukten schwierig. Bei allen drei Playern werden die erstellten Komponenten visualisiert und korrekt dargestellt. Die Darstellungsmöglichkeiten der Abspielsoftware umfassen dabei den gesamten Funktionsumfang der Erstellungsoftware. Maya bietet bei der Erstellung von 3D-Szenen eine Fülle von Möglichkeiten an. Nur ein Teil der Funktionen können beim Export in das X3D-Format übernommen und vom Player dargestellt werden. Der Nutzer muss sich dessen bewusst sein, um keine unnötigen Einstellungen vorzunehmen, die beim Export der Datei nicht übernommen werden. Ein weiteres Problem stellt das Parsen der Präsentationen im Player dar. Der Akzeptanzgrad der X3D-Dateien beim Laden in die unterschiedlichen Player-Software ist dabei different.

Im Allgemeinen unterscheiden sich die X3D und VRML Player untereinander in ihrer Handhabung. Einige bieten eine offensichtliche Menüleiste an. Bei dieser Abspielsoftware ist eine unkomplizierte Navigation umgesetzt. Andere Abspielsoftware bieten eine versteckte Navigation die durch Pop-up Menüs realisiert wird. Dies führt zu einem größeren Darstellungsbereich für die 3D-Szene, wirkt jedoch der Benutzerfreundlichkeit entgegen.

Eine Übersicht der Abspielsoftware für X3D- und VRML-Formate ist in den folgenden Tabellen aufgezeigt. Die erste Tabelle zeigt den Softaretyp und die Kompatibilitäten zwischen Abspielsoftware und Betriebssystem. Die zweite Tabelle bietet einen Überblick über die wichtigsten X3D-Player und deren Eigenschaften.

Software	Version	Softwaretyp	Betriebssystem			Browser		X3D
			Windows	Linux	Macintosh	Internet Explorer	Mozilla Firefox	
Cortona	4.2	Plug-in	X		X	X	X	
Octaga Player	2.0	Plug-in, Stand alone	X	X		X	X	X
Cosmo Player	2.1.1	Plug-in	X			X	X	
Flux Player	1.2	Plug-in, Stand alone, Toolkit	X			X	X	X
blaxxun Contact	5.3	Plug-in	X			X		
BS Contact	6.2	Plug-in	X			X	X	X
Free WRL	1.18	Plug-in, Stand alone		X	X		X	X
Open VRML	0.16	Plug-in, Stand alone		X	X		X	
XJ3D	1.0	Stand alone, Toolkit	X	X	X			X
Carina	20060520	Stand alone	X	X	X			X
Demotride	2.01	Stand alone	X					
Orbisnap	0.2	Stand alone	X	X	X			
VRML view	2.0b	Stand alone	X	X				
BS Contact J	6	Applet	X	X	X	X	X	X
Blaxxun3D	2.30	Applet	X	X	X	X	X	

	XJ3D 1.0	Octaga Player 2.0	Flux Player 1.2	BS Contact 6.2
Version	1.0	2.0.0.2	1.2	6.2
Bildformate	PNG, GIF, JPG, BMP, DIB, ICO, CUR	PNG, GIF, JPG, BMP, DIB, ICO, CUR	PNG, GIF, JPG, BMP, DIB, ICO, CUR	PNG, GIF, JPG, BMP, DIB, ICO, CUR
Soundformate	WAV,Midi,MP3,RealAudio,WMV	WAV,Midi,MP3,RealAudio,WMV	WAV,Midi,MP3,RealAudio,WMV	WAV,Midi,MP3,RealAudio,WMV
Videoformate	MPEG-1,RealVideo,WMV,AVI,GIF	MPEG-1,RealVideo,WMV,AVI,GIF	MPEG-1,RealVideo,WMV,AVI,GIF	MPEG-1,RealVideo,WMV,AVI,GIF
Navigationsformen	<ul style="list-style-type: none"> - Umschaltung zwischen Viewpoints über Funktionsleiste erfolgt schnell und einfach inkl. Beschriftung des Viewpoints - verschiedene Navigationsmöglichkeiten (Walk, Zoom,...) - kein Navigieren zwischen zuvor erzeugten freien Sichten möglich - keine Kamerafahrt beim Wechsel zwischen den Viewpoints - keine Horizontalasurichtung möglich 	<ul style="list-style-type: none"> - Navigationsgeschwindigkeit einstellbar - verschiedene Navigationsmöglichkeiten (Walk, Zoom,...) - Viewpoint Navigation ist umständlich - Kamerafahrt zwischen den Viewpoints vorhanden - verschiedene Sichtmodi (Gitter, Punkte,...) einstellbar - keine Horizontalasurichtung möglich 	<ul style="list-style-type: none"> - Kamerafahrt zwischen Viewpoints vorhanden - wenig Navigationsvarianten - Horizontalasurichtung möglich - Navigieren zwischen freien Sichten möglich 	<ul style="list-style-type: none"> - umständliche Menüführung bei Viewpoint-Navigation - Kamerafahrt zwischen Viewpoints vorhanden - Navigationseinstellung schlecht konfigurierbar - verschiedene Sichtmodi (Gitter Punkte,...) einstellbar - keine Horizontalasurichtung möglich
Beleuchtung	<ul style="list-style-type: none"> - gute, helle Beleuchtung - interne Beleuchtung im Player nicht abschaltbar 	<ul style="list-style-type: none"> - Extra Beleuchtung zuschaltbar - schlechte Licht – Schatten Übergänge 	<ul style="list-style-type: none"> - Gute helle Beleuchtung - interne Beleuchtung zuschaltbar 	<ul style="list-style-type: none"> - interne Beleuchtung zuschaltbar - schlechte Licht – Schatten Übergänge
Kollisionsdetektion	<ul style="list-style-type: none"> - im Walk-Modus vorhanden 	<ul style="list-style-type: none"> - im Walk-Modus vorhanden und abschaltbar 	<ul style="list-style-type: none"> - im Walk-Modus vorhanden 	<ul style="list-style-type: none"> - in verschiedenen Navigationsformen vorhanden und abschaltbar
Gravitation	<ul style="list-style-type: none"> - im Walk-Modus vorhanden 	<ul style="list-style-type: none"> - im Walk-Modus vorhanden und abschaltbar 	<ul style="list-style-type: none"> - im Walk-Modus vorhanden 	<ul style="list-style-type: none"> - in verschiedenen Navigationsformen vorhanden und abschaltbar
Lizenz	<ul style="list-style-type: none"> - Open Source 	<ul style="list-style-type: none"> - Kommerziell und als Testversion verfügbar 	<ul style="list-style-type: none"> - Kommerziell aber für private Nutzung frei verfügbar 	<ul style="list-style-type: none"> - Kommerziell und als Testversion verfügbar
Einstellungen	<ul style="list-style-type: none"> - Keine Einstellungsoptionen im Player vorhanden 	<ul style="list-style-type: none"> - Konfiguration des Players möglich 	<ul style="list-style-type: none"> - Nur wenige Einstellmöglichkeiten 	<ul style="list-style-type: none"> - Konfiguration des Players möglich

IV RawKee - Open Source X3D Plugin (im Detail)

1.1 Maya

Die Software Maya ist eine, von der Firma Alias (in der Zwischenzeit von Autodesk übernommen) entwickelte, sehr verbreitete 3D Visualisierungs- und Animationssoftware. Der Anwendungsbereich liegt primär in der Film- und Fernsehindustrie und bei der Erstellung von Grafiken für Computer- und Videospiele. Daneben wird Maya auch in anderen Bereichen wie der industriellen Fertigung, Architekturvisualisierung und in Entwicklung und Forschung verwendet. Maya ist eines der bekanntesten und meistgenutzten Softwareprodukte im Bereich 3D-Modellierung, Computeranimation und Rendering. Zur Entwicklung einer 3D-Szene steht dem Nutzer eine grafische Benutzeroberfläche zur Verfügung. Die Menüleiste von Maya passt sich in ihren Menüpunkten in Abhängigkeit vom Menüset an. Diese Abhängigkeiten helfen dem Nutzer für den jeweiligen Entwicklungsschritt die notwendigen Funktionen verfügbar zu haben. Durch sie werden Modellierung, Animation und Rendering voneinander getrennt. In jedem der Menüsets bietet Maya einen großen Umfang an Funktionen an, die eine Erstellung der Szene ermöglichen.

Der Grundbaustein von jeder Szenenentwicklung in Maya bildet der Dependency Graph. Dies ist ein Szenengraph der die gesamten Informationen der 3D Welt zusammenfasst. Die Szene setzt sich dabei aus voneinander unabhängigen Knoten zusammen. Jeder Knoten besitzt Attribute über die eine Kommunikation zwischen den verschiedenen Knoten der Szene realisiert wird. Sie können dafür Informationen über die Input-Connection aufnehmen, gegebenenfalls weiterverarbeiten und über die Output-Connections an andere Knoten weitergeben.

Eine der herausragenden Funktionalitäten der Maya Software ist die Offenheit des Grundmoduls bezüglich Erweiterungen von Drittanbietern. Es ist möglich die Software selbst an sehr individuelle Anforderungen einer bestimmten Zielgruppe anzupassen. Die aus dieser Architektur entstehenden Möglichkeiten bilden den Ausgangspunkt der RawKee Plug-in Entwicklung.

1.2 RawKee

Das RawKee Plug-in ist derzeit in der Version 1.0.3 verfügbar. Nach dem Einbinden steht das Plug-in in der Maya-Umgebung bereit. Die Modellierung und die Animation der Szene erfolgt mittels der in der Software zur Verfügung gestellten Funktionen. Knotenelemente, die beim Export einer Szene in das X3D-Format benötigt werden, können ebenfalls über den Interaction-Editor oder die Menüpunkte des Plug-ins hinzugefügt und über den Dependency Graph strukturiert werden. Nach der Fertigstellung wird die Szene exportiert und die Bestandteile der Szene dabei in einen X3D oder VRML –Szenengraph transformiert. Die zur Verfügung stehenden X3D-Knotentypen und deren Attribute sind durch das Web3D Konsortium festgelegt. RawKee unterstützt die wichtigsten dieser Knotenelemente. Die disponiblen Knotentypen sind im Folgenden tabellarisch, nach der Menüstruktur des Plug-ins, aufgelistet.

Menüpunkt	X3D Knoten	Beschreibung
Add Scene Nodes	Viewpoint	Definiert Position und Blickrichtung in die Szene
	Navigation-Info	beschreibt Betrachtungsmodell und physikalischen Eigenschaften des Nutzeravatars.
	World-Info	Speichert Hintergrundinformationen zur 3D Welt
	Directional Light	Erzeugung eines parallelen Lichteinfall zur Beleuchtung der Szene
	Spot-Light	Erzeugung einer kegelförmig ausstrahlenden Lichtquelle
	Point Light	Erzeugung einer Punktlichtquelle zur Szenenbeleuchtung
	Background	Definition von Hintergrundfarben und eines Umgebungswürfels
Add Grouping Nodes	Transform	Gruppenknoten zur Aufnahme und Gruppierung von Knoten
	Group	Gruppierungsknoten
	Switch	Gruppierungsknoten, der immer nur einen Kindknoten ausführt
	Collision	Ermittelt Kamera-Objekt-Kollisionen durch Viewpoint und Navigationinfo (Avatar-Size)

Menüpunkt	X3D Knoten	Beschreibung
	LoD	nutzt den Abstand Kamera-Objekt, um zwischen als Kinderknoten enthaltenen Detaillierungsstufen umzuschalten
Add Primitives	Box	Geometrie-Knoten zur Darstellung eines Würfels
	Cone	Geometrie-Knoten zur Darstellung eines Kegels
	Cylinder	Geometrie-Knoten zur Darstellung eines Zylinders
	Sphere	Geometrie-Knoten zur Darstellung einer Kugel
	Indexed Face Set	Geometrie-Knoten mit untergeordneten Eigenschaftsknoten
	Text	Geometrie-Knoten zur Darstellung von 2D-Text
Add Audio Nodes	Audio Clip	Stellt Audio Clip, der von einem (Sound)Knoten verwendet werden kann, bereit
	Sound	Referenziert Audioclip zur Soundwiedergabe
Add Networking Nodes	Anchor	Gruppenknoten, der durch Aktivierung den Inhalt der referenzierten URL lädt
	Inline	Gruppierungsknoten, der Knoten via URL laden kann
	Load Sensor	Generiert Events wenn Knoten der Watch-List geladen werden
Add Event Utility Nodes	Boolean Filter	Filtert boolesche Werte und Events und wertet sie aus
	Boolean Sequencer	erzeugt periodisch diskrete boolesche Werte, die zu anderen boolesche Attributen geroutet werden können
	Boolean Toggle	Negiert boolesche Werte und gibt diese aus
	Boolean Trigger	Konvertiert eingehendes Zeitsignal in boolesche Werte
	Integer Sequencer	erzeugt periodisch diskrete Integer-Werte, die zu anderen Integer-Attributen geroutet werden können
	Integer Trigger	Konvertiert boolesche Werte und Zeitwerte in Integerwerte
	Time Trigger	Konvertiert boolean (true)-events in time-events
Add Metadata Nodes	Metadata Double	Stellt Informationen über Elternknoten dar- inkl. Doublewerte
	Metadata Float	Stellt Informationen über Elternknoten dar- inkl. Floatwerte
	Metadata Integer	Stellt Informationen über Elternknoten dar- inkl. Integerwerte
	Metadata Set	Container für Metadatenknoten
	Metadata String	Stellt Informationen über Elternknoten dar- inkl. Stringwerte
Add Sensor Nodes	Cylinder Sensor	Cylinder-Sensor konvertiert Zeigerbewegungen (z.B. einer Maus) in Rotationsangaben durch Einsatz eines an der lokalen Y-Achse ausgerichteten, unsichtbaren Zylinders.
	Key Sensor	Generiert Ereignisse bei Tastendruck
	Plane Sensor	Konvertiert Ereignisse in 2D-Bewegungen auf der X-Y-Ebene
	Proximity Sensor	Generiert Ereignisse wenn der Benutzer eine Mausbewegung über einen definierten Bereich ausführt
	Sphere Sensor	Konvertiert Zeigerbewegungen in Rotationsangaben
	String Sensor	Generiert Ereignisse nach den Tastatureingaben des Nutzers
	Time Sensor	Generiert Ereignisse in Abhängigkeit von der Zeit
	Touch Sensor	Generiert Ereignis wenn Benutzer auf eine Geometrie klickt
	Visibility Sensor	Erkennt wenn ein definierter Bereich durch Navigation des Nutzers sichtbar wird
	Gamepad Sensor	Reagiert auf Eingaben durch ein Gamepad
Animation Nodes	Color Interpolator	Generiert Farbbereich, dessen Werte zu einem Color-Attribut geroutet werden können
	Coordinate Interpolator	Interpoliert Bewegungen anhand einer Liste von Werten
	Normal Interpolator	Erzeugt Mengen von Normalvektoren auf der Oberfläche einer Einheitskugel

	Orientation Interpolator	Interpolation einer Rotationsbewegung
	Position Interpolator	Interpolation von Wertetripeln zur Transformation von Knoten
	Scalar Interpolator	Interpolation mittels eines Wertes
Add User-Defined Nodes	Script	Erstellung von ECMA- oder Java-Skripten

Die nachfolgende Tabelle vervollständigt die Menüstruktur des Plug-ins in Maya, beschreibt aber Funktionen und keine konstruierbaren X3D-Knotenelemente

Menüpunkt	Unterpunkt	Beschreibung
X3D Character Editor		Erzeugung von Charakteranimationen mittels Time-Sensoren zur Animationserstellung mittels Joints
X3D Animation Editor		Bedienoberfläche zur Animationserzeugung mittels Time-Sensoren
Utility Commands	Set All MultiTexture Modes: Default	Zurücksetzen der multitexture-export Eigenschaften
	Set All MultiTexture Modes: Replace	Ersetzen einer Textur innerhalb eines Multitexture-Knotens
	Set All MultiTexture Modes: Modulate	Zusammenführung der einzelnen Texturen zu einer Multitextur
	Set All MultiTexture Modes: Add	Hinzufügen von Texturen zu einem MultiTexture-Knoten
	Make All Textures PixelTextures	Export der Texturen als Pixeltexture (Array von Pixel-Werten)
	Make All Textures ImageTextures	Export der Texturen als Image-Texturen (Image-Texturen bildet eine 2D-Bilddatei auf ein geometrisches Objekt ab)
	Export Scene Prep	Optimierung der Szene bezüglich der Darstellbarkeit und Dateigröße für den Export
Plug-in Tutotrials		Internetverknüpfung zu RawKee-Tutorien
RawKee Developers		Internetverknüpfung zu RawKee-Entwicklern
Show Preview		Erzeugung einer Szenenvorschau

Das RawKee Plug-in nutzt bei der Szenendarstellung den Dependency Graph von Maya. Sowohl Maya- als auch X3D-Szenen sind durch einen Szenengraphen darstellbar. Aufgrund dessen wird die Szene während des Erstellungsvorgangs über Maya-Knoten veranschaulicht. Die Struktur ist damit schon während der Erstellung ersichtlich und die Attribute, die sowohl X3D- als auch Maya Knoten besitzen, können durch den von Maya bereitgestellten Attribut-Editor angezeigt und bearbeitet werden. Die Visualisierung des Graphen wird im Interaction-Editor des Plug-ins umgesetzt. Die Erzeugung aller X3D Knotenelemente einer 3D-Präsentation ist in ihm möglich. Die Erzeugung und Manipulation ist mit den in Maya zur Verfügung stehenden Grundfunktionen jedoch benutzerfreundlicher. Zur Erstellung einer 3D-Präsentation für eine Vorlesung, nach den in der Einleitung erarbeiteten Ansprüchen an ein solches Werkzeug, fehlen in der ursprünglichen Software Erzeugungsmöglichkeiten. X3D unterstützt die Darstellung von Szenenhintergründen durch Farben sowie Farbverläufen. Das Integrieren von Bildern als environment cube ist ebenfalls definiert, aber in RawKee nicht umgesetzt. Die Integration von Text in die Szene ist mit dem Plug-in gleichermaßen nicht realisiert. Text ist jedoch unverzichtbar bei einer Vorlesungspräsentation. Ein weiterer noch nicht im Plug-in umgesetzter Aspekt ist die Zusammensetzung einer 3D-Präsentation aus mehreren Teilszenen. Die Möglichkeit ein separates ECMA-Skript, zum Erreichen der Zielstellung zu schreiben, steht dem Nutzer zur Verfügung, jedoch sollte dieser Programmieraufwand automatisiert ablaufen, um dem Nutzer den Erstellungsprozess zu erleichtern. Ein letzter Kritikpunkt ist die Ein- und Ausblendung von Objekten während der Präsentation. Dies ist im Plug-in derzeit nicht möglich.

Folgende Punkte sind daher dem Plug-in während der Erstellung dieser Belegarbeit hinzugefügt worden:

- Integration von Hintergrund mit Farbverläufen in eine 3D-Präsentation
- Erzeugung eines environment cube
- Erstellung und Positionierung von 2D Text
- Automatische Ausrichtung des Textes zum Betrachter
- Konfigurationsmöglichkeiten bezüglich des Erscheinungsbildes von 2D-Text
- Automatische Skriptgenerierung zur Navigation zwischen Teilszenen der Präsentation
- Möglichkeit der Erstellung von Transparenzanimationen in der Szene

Die Entwicklung der Erweiterungen und deren Nutzung werden im nächsten Kapitel beschrieben.

V Maya Programmierung

1.1 Einführung

Kaum eine Branche ist so dynamisch wie die der Computergrafik. Maya als Software zur 3D-Szenenerstellung ist mit an der Spitze dieser Entwicklung. Dadurch unterliegt die Software einem ununterbrochenen Aktualisierungsprozess. Dennoch ist es für die Entwickler nicht möglich alle Bedürfnisse der Nutzer zu befriedigen. Aufgrund dessen ist Maya als offene Architektur konzipiert. Benutzer der Software können das Werkzeug an ihre individuellen Anforderungen anpassen. Dies beinhaltet die Abänderung der Bedienoberfläche genauso wie das Hinzufügen neuer Charakteristiken. Diese Wandlungsfähigkeit beruht auf dem zugrunde liegenden Framework, bestehend aus 3D-Komponenten wie zum Beispiel Objekten, Animationen oder Dynamics. Diese können individuell angepasst und erweitert werden. Die Entwicklung von Maya-Erweiterungen kann mittels zwei Programmiersprachen erfolgen: MEL oder C++. Welche der beiden Sprachen bei der Erweiterung von Maya verwendet werden sollten, hängt von externen Faktoren ab.

MEL

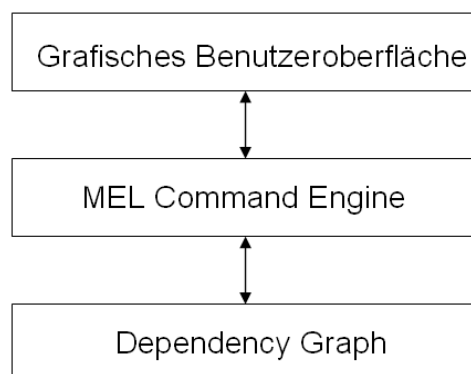
MEL ist ein Akronym für Maya Embedded Language. Diese Programmiersprache wurde speziell für Maya entwickelt und ist plattformunabhängig. MEL ist eine Interpretersprache und kann damit ohne vorherige Kompilierung unverzüglich ausgeführt werden. Dadurch ist es möglich MEL-Skripte in Maya zu schreiben und zu testen. Auf externe Werkzeuge kann dabei verzichtet werden. Der Nachteil bei der Ausführung von Mel-Skripten besteht in der Performance. Sie werden langsamer ausgeführt als vergleichbare C++ Programme. Dies ist auf die Laufzeitinterpretation des Codes zurückzuführen. Mel Skripte müssen zum Zeitpunkt der Ausführung interpretiert und in Maschinen-Code konvertiert werden. Diese Schritte sind bei bereits kompilierten C++ Programmen unnötig.

C++

Maya kann Programme, die in der Standard C++ Programmiersprache geschrieben sind, ausführen. Dadurch ist die Erstellung von Maya-Plug-ins, die nahtlos mit dem Maya-Framework zusammenarbeiten, möglich. Die Schnittstelle bildet eine aus Klassenbibliotheken bestehende C++ API. Zur Erarbeitung eines Plug-ins wird ein C++ Programm geschrieben, welches die grundlegenden Maya Komponenten verwendet und diese ausbaut.

Die Entscheidung für oder gegen eine der beiden Optionen wird durch den zu erstellenden Gegenstand beeinflusst. Der Funktionsumfang der Programmiersprachen ist nicht deckungsgleich. Einige Features können nur mit MEL realisiert werden und andere wiederum nur mittels C++. Die Entscheidungsfindung beruht damit auf der zu erstellenden Erweiterung. Zur Lösung eines gegebenen Problems ist die Kombination beider Techniken ebenso möglich. Das RawKee Plug-in bedient sich dieser Kombination von MEL- und C++ Programmteilen.

Das Maya-System lässt sich auf das nebenstehende Schema vereinfachen. Der Nutzer der Software interagiert mit der Maya Bedienoberfläche. Darüber können Funktionen wie das Bewegen von Objekten oder das Ändern von Parametern ermöglicht werden. Bei der Benutzung der Oberfläche übergibt Maya automatisch entsprechende MEL Anweisung, die von der MEL Command Engine interpretiert und ausgeführt werden. Diese Anweisungen werden auf dem Dependency Graph, der die Struktur der Szene beinhaltet, umgesetzt. Der Dependency Graph definiert nicht nur die Daten der Szene er definiert ebenso den Ablauf dieser. Das RawKee Plug-in arbeitet auf der Basis dieser Architektur und konvertiert beim Export einer Szene in das X3D- oder VRML Format den Dependency Graph in einen Szenengraph des Export-Formats.



1.2 Aufbau des RawKee Plug-ins

Allgemein

Das RawKee Plug-in besteht sowohl aus MEL-Skripten als auch aus C++ programmierten Softwareteilen. Nach der Installation des Plug-ins sind die Mel-Skripte in der Ordnerstruktur der Maya-Installation unter /Scripts/Others abgelegt. Die Skripte können mit einem beliebigen Texteditor betrachtet und modifiziert werden. Der C++ Source Code ist kompiliert und in die Datei x3d.mll transformiert im Plug-ins Ordner der Maya Installation abgelegt. Maya stellt für die Erstellung einer *.mll Datei einen Plug-in Wizard für das Microsoft Visual Studio bereit. Damit dieser Wizard verfügbar ist, muss jedoch die Maya-Komponente „Development Kit“ bei der Installationsroutine von Maya installiert worden sein. Wenn dies der Fall ist, befindet sich das Plug-in im Pfad devkit/pluginwizard des Maya-Ordners. Mittels dieses Plug-in Wizards kann ein C++ API Projekt für Maya im Visual Studio erzeugt werden. Der Erstellungsvorgang, also die Bereitstellung der von Maya benötigten mll-Datei, des Visual Studio Projektes wird daraufhin automatisiert durchgeführt. Dabei werden für den Erstellungsvorgang notwendige Konfigurationen vorgenommen. Der Source Code wird kompiliert und die Bibliotheksdatei x3d.mll wird automatisch erzeugt. Damit werden die vom Nutzer zu konfigurierenden Projekteinstellungen auf ein Minimum reduziert.

MEL-Skripte

Durch das Einbinden des Plug-ins in Maya wird die dll-Datei geladen. Während des Ausführungsvorgangs der x3d.mll werden alle RawKee MEL-Skripte ausgeführt. Die derzeit 137 Skript-Dateien dienen der Visualisierung des RawKee Plug-ins innerhalb der Maya-Oberfläche. Sie visualisieren die Menüeinträge sowie die Darstellung der verschiedenen Editoren. Die Oberfläche für die Exporteigenschaften wird ebenfalls in MEL-Skripten realisiert. Die Skripte können in 3 Kategorien unterteilt werden. Dateien die mit dem Akronym AE für Attribut Editor beginnen, stellen Funktionen bereit, die das Layout und die für das jeweilige X3D-Knotenelement definierbaren Attribute anzeigt. Diese, im Attribut-Editor definierten Werte und Einstellungen, werden in der aktuellen Szene übernommen und die Folgerungen aus den Attributen in Maya angezeigt. Ebenso werden sie beim Export in die X3D-Datei übergeben. Die vom Web3D Konsortium grundlegend definierten Knotenattribute werden in Dateien mit dem Namen x3d_Knotenname.mel beschrieben. Der genaue Aufbau der Dateien wird in den Abschnitten 1.3 bis 1.6 dieses Kapitels näher beschrieben. Der dritten Kategorie sind nachfolgende Dateien zuzuordnen:

- x3d.mel
- x3d_node_creation_proc.mel
- x3d_scenegraph_ui_tree.mel
- x3d_ie_menus.mel
- x3dUtilCommands.mel
- x3d_exporter_procedures.mel

Die Datei x3d.mel erzeugt das RawKee Menü mit allen Einträgen sowie das Attributfenster für die Export-Eigenschaften. Weiterhin werden die Vorschaufunktion und weitere grundlegende Funktionen in ihr definiert. Die Einträge des RawKee Menüs verweisen auf die Datei x3d_node_creation_proc.mel. In dieser werden für alle X3D-Knotenelemente die entsprechenden Knoten im Dependency Graph erzeugt. Durch diese Abstraktion ist es möglich den Szenengraph in Maya zu visualisieren. Die Funktionen der Datei definieren darüber hinaus Eigenschaften der Knoten bezüglich ihrer Eltern-Kind Beziehungen mit anderen Knoten. Dadurch können Teilgraphen erstellt werden die aus mehr als einem Knoten bestehen. Der Szenengraph einer 3D-Präsentation wird im Interaction-Editor des Plug-ins dargestellt. Dieser wird in der Datei x3d_scenegraph_ui_tree.mel beschrieben. Die Datei beinhaltet des Weiteren Funktionen zur Menüführung mit der eine Erzeugung von Szene-Knoten innerhalb des Editors möglich ist. Der Aufbau des Routing Fensters innerhalb des Interaction-Editors wird ebenfalls festgelegt. Damit bei der Erstellung von Routes das Definieren von falschen Verknüpfungen verhindert wird, legt die Datei x3d_ie_menus.mel fest, welche Attribute des Ausgangsknotens mit welchen Attributen anderer Knoten verbunden werden können. Dabei wird unterschieden welche Attribute als Output eines Knotens und welche als Input beschrieben werden können. Die auswählbaren Attribute basieren auf den in den MEL-Skripten festgehaltenen Knotenattributen. X3dUtilCommands.mel definiert den Animation- sowie den

Character-Editor. Es wird dabei die Erstellung von Interpolatoren und Time-Sensoren beschrieben, ebenso wie die Verknüpfung der Komponenten durch Routes. Die Erstellung von Animationen im Animation-Editor zieht eine Automatisierung des Ablaufs von Teilschritten nach sich, die gleichfalls manuell im Interaction-Editor vom Nutzer vorgenommen werden können. Zusätzlich werden Funktionen zur Optimierung der Szene beschrieben. Dies beinhaltet unter anderem das Löschen unnötiger Knotenelemente. Die Datei `x3d_exporter_procedures` enthält Funktionen die den Exportvorgang vorbereiten. Die Bereitstellung von für den Export notwendigen Daten steht dabei im Vordergrund.

X3D.mll

Die Funktionen der MEL-Skripte sind mit den C++ Klassen und Funktionen der `x3d.mll` verknüpft. Über die MEL-Skripte werden Daten bereitgestellt auf welche durch den C++ Code zugegriffen wird um den Export der 3D-Präsentation in das X3D/VRML Format durchzuführen.

Die C++ Dateien `x3dKnotenname.cpp` dienen der Definition von X3D Knotenelementen und deren Attributen. Durch den Aufbau der Objekte in den C++ Klassen wird die Struktur des Knotenelements festgelegt. Darüber hinaus sind folgende Dateien zur Umsetzung des RawKee Plug-ins notwendig:

- `webX3DExporter.cpp`
- `web3dExportMethods.cpp`
- `x3dExportOrganizer.cpp`
- `sax3dWriter.cpp`

Die Datei `webX3DExporter.cpp` führt die Initialisierung des Plug-ins in der Maya-Umgebung durch. Die Entkopplung aus der Software ist ebenfalls über diese Datei realisiert. Es ist daher möglich das Plug-in während der Laufzeit zu entkoppeln und wieder einzubinden. Beim Einbinden des Plug-ins werden alle MEL-Skripte ausgeführt und stehen somit in Maya zur Verfügung. Eine weitere Aufgabe der Datei besteht in der Deklaration von Knoten. Damit die X3D Knotenelemente in Maya bereitgestellt werden können, müssen diese erst in Maya registriert werden. Bei deren Initialisierung wird zwischen Gruppenknoten und Blattknoten unterschieden, um eine korrekte Strukturierung der Knoten innerhalb des Dependency Graph und später im exportierten Dokument zu gewährleisten. Darüber hinaus werden in der Datei Deklarationen und Initialisierungen von Startwerten für die Attribute der Knotenelemente durchgeführt.

Über die Datei `web3dExportMethods.cpp` erfolgt die Datenhaltung der Informationseinheiten die exportiert werden sollen. Für jedes Knotenelement mit all den dazugehörigen Unterelementen werden die in Maya definierten Werte verarbeitet und zwischengespeichert. Die Datei `x3dExportOrganizer.cpp` greift auf diesen Datenbestand zu um die Knoten des Dependency Graphs zu manipulieren oder diesen zu exportieren. Dabei werden die für den Export eines Knotenelements nötigen Informationen zusammengetragen. Die Aufarbeitung läuft nach den in den Maya Exporteigenschaften definierten Vorgaben ab. Die Datei bildet die Struktur des Dependency Graph auf die Struktur des Zielformats ab und übergibt diese Informationen an die Datei `sax3dWriter.cpp`. Diese Datei ist vollkommen unabhängig von der Maya API. Durch sie können X3D und VRML Knoten in eine Datei geschrieben werden. Die Funktionen bieten die Möglichkeit Code in XML, VRML und Binary Format zu erzeugen. Der Datei werden schrittweise einzelne Teile des Dependency Graph übergeben und in das Zielformat konvertiert. Das Ergebnis dieses Vorgangs wird in die Zieldatei geschrieben.

1.3 X3D Hintergrund

Das RawKee Plug-in unterstützte den Background-Knoten nach der X3D-Spezifikation bisher nicht. Daher war es mittels des Werkzeuges nicht möglich, während der Erstellung einer 3D-Präsentation in Maya, Hintergrundfarben zu definieren oder Bilder zu integrieren. Der Background-Knoten wurde im Zuge dieser Belegarbeit in den Katalog der von RawKee unterstützten X3D-Knoten aufgenommen. Er simuliert Himmel und Untergrund durch Arrays von Umgebungsfarbwerten. Bilder können unabhängig von der Hintergrundfarbgebung über Pfadangaben in die Präsentation eingebunden werden. Der X3D Backgroundknoten ist nach dem Web3D Konsortium folgendermaßen definiert:

```

Background : X3DBackgroundNode {
    SFBool    [in]    set_bind
    MFFloat   [in,out] groundAngle []    [0,π/2]
    MFColor   [in,out] groundColor []    [0,1]
    MFString  [in,out] backUrl    []    [urn]
    MFString  [in,out] bottomUrl   []    [urn]
    MFString  [in,out] frontUrl    []    [urn]
    MFString  [in,out] leftUrl     []    [urn]
    SFNode    [in,out] metadata    NULL  [X3DMetadataObject]
    MFString  [in,out] rightUrl    []    [urn]
    MFString  [in,out] topUrl      []    [urn]
    MFFloat   [in,out] skyAngle    []    [0,π]
    MFColor   [in,out] skyColor    [0 0 0][0,1]
    SFTime    [out]   bindTime
    SFBool    [out]   isBound
}

```

Die blau unterlegten Elemente der Definition werden vom RawKee Plug-in im Attribut-Editor von Maya unterstützt. Es können somit 2 Farben als Hintergrund definiert werden. Über die Attribute skyColor und groundColor werden diese Farbwerte festgelegt. Es ist bei beiden Attributen die Definition von mehreren Farbwerten optional möglich. Dies führt zu einem Farbverlauf zwischen den Farben. RawKee unterstützt einen Farbverlauf zwischen jeweils 2 Farben für skyColor und für groundColor. Nach der Definition des Background-Knotens ist es nicht notwendig eine groundColor anzugeben, da sich die definierte skyColor auch über den kompletten Hintergrund erstrecken kann. Zur Beschreibung des Farbverlaufes und der Definition der Horizontlinie dienen die Attribute skyAngle und groundAngle. Der Winkel von skyColor liegt zwischen dem Zenit bei 0.0, dem Horizont bei $\pi/2$ beziehungsweise 90 Grad und dem Nadir bei π oder gleichbedeutend mit 180 Grad. Der Winkel von groundAngle liegt zwischen dem Nadir mit 0.0 und dem Horizont bei $\pi/2$. Zusätzlich zu Hintergrundfarbwerten können Bilder eingefügt werden. Das Hinzufügen von bis zu sechs Bildern und damit die Bildung eines environment cube ist möglich. Dafür muss den Attributen ein Verweis auf das zu integrierende Bild deklariert werden. Das folgende Beispiel zeigt einen X3D-Backgroundknoten mit Himmel und Untergrund sowie einem Hintergrundbild. Der Himmel zeigt einen Farbverlauf und erstreckt sich bis zum Horizont.

```

<Background DEF='bg'
    skyAngle='1.57'
    skyColor='0 0 1 0 1 1'
    groundColor='0 0.5 0'
    frontUrl='Bild.jpg' />

```

Zur Manipulation eines Backgroundknotens mittels RawKee Plug-in existieren zwei MEL-Skripte: x3d_background_tables.mel und AEx3dBackgroundTemplate.mel. Zuvor muss der Knoten jedoch im Dependency Graph erzeugt werden. Die Erstellung erfolgt über den Funktionsaufruf im RawKee Menü und dieser ruft wiederum eine Funktion in der Datei x3d_node_creation_proc.mel auf. Die Erstellung des Knotens erfolgt mittels folgender Funktion.

```

global proc createX3DBackgrounds ()
{
    string $selTrans[] = getSelectedGroupingNodes();
    int $selSize = size($selTrans);
    if($selSize > 0)

```

```

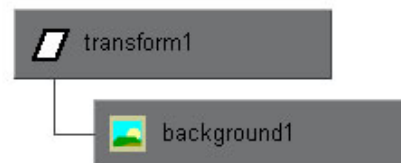
{
    string $newTrans = `createNode "transform" -p $selTrans[0]`;
    string $newBGNode = `createNode
        -n "background#" "x3dBackground"
        -p $newTrans`;

    select $selTrans[0];
}
else
{
    string $newTrans = `createNode "transform"`;
    string $newBGNode = `createNode
        -n "background#" "x3dBackground"
        -p $newTrans`;

    select $newTrans;
}
}

```

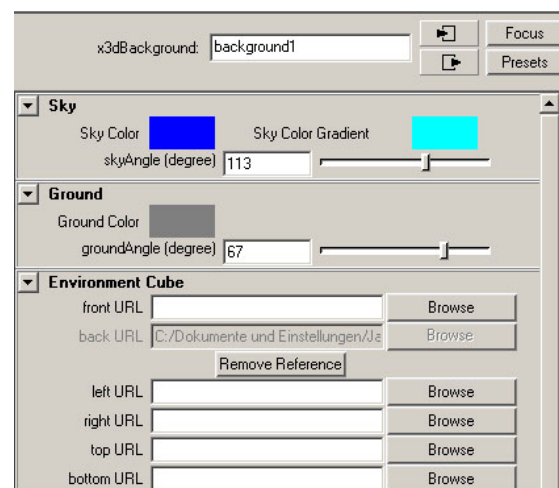
Dabei wird ein Gruppierungsknoten „Transform“ und ein Background-Knoten als dessen Kind-Knoten erzeugt. Eine Nummerierung der Knoten verhindert Verwechslungen bei mehrfacher Erstellung. Die Unterscheidungsanweisung im Quellcode überprüft ob der zu erstellende Knoten in einen bereits bestehenden Knotenbaum eingefügt oder ob dieser neu angelegt wird. Das Ergebnis der Ausführung dieser Prozedur wird im Interaction-Editor, wie im rechten Bild dargestellt, ersichtlich.



In der Datei x3d_background_tables.mel ist die Knotenspezifikation eines X3D Background-Knotens, wie sie auf der vorherigen Seite dargestellt ist, definiert. Durch diese Datei ist festgelegt welche Events erzeugt werden können und welche Events für die Routeerstellung zur Verfügung stehen. Durch die Datei AEx3dBackgroundTemplate.mel wird der Aufbau des Attribut-Editors für diesen Knotentyp festgelegt. Mittels dieser Datei werden auch die Eigenschaften und Werte des Knotens gesetzt und können daraufhin exportiert werden. Die nebenstehende Abbildung zeigt das Erscheinungsbild des Background-Knotens im Attribut-Editor der Maya-Software.

Der Editor ist in drei Bereiche unterteilt. Die erste Sektion ermöglicht die Definition von Farben und den Farbverlauf des Himmels sowie die Einstellung des skyAngle. Darunter befinden sich die Einstellungen für den Untergrund. Die Einstellungsmöglichkeiten des Untergrundes sind äquivalent zu denen des Himmels. Über den dritten Bereich erfolgt die Erstellung des Environment Cube durch Einbindung von Bilddateien.

Zur Erstellung des Editor-Templates in der AEx3dBackgroundTemplate.mel wird eine Vorlage definiert. Diese verweist auf Prozeduren, die das genaue Layout festlegen. Die Komponenten der Schablone verweisen wiederum auf Funktionen, die das Definieren von Attributen ermöglichen. Über diese Funktionen werden die Nutzereinstellungen



festgelegt um sie beim Export ausgeben zu können. Zur Veranschaulichung stellt der folgende Code die Funktionen zum setzen des skyAngle-Attributes dar.

```
global proc AEx3dBackgroundTemplate( string $nodeName )
{
    editorTemplate -beginLayout "Background URL" -collapse false;
        editorTemplate -callCustom
            "AEx3dBackgroundTemplateNew"
            "AEx3dBackgroundTemplateReplace" "skyAngle";
    editorTemplate -endLayout;
}
```

Bei der ersten Erstellung eines Backgroundknotens wird ein Template erzeugt. Bei jedem weiteren Aufruf wird daraufhin dieses nur aktualisiert. Es erfolgt beim erstmaligen Erstellen der Aufruf der Prozedur AEx3dBackgroundTemplateNew. Es wird dabei auf die Prozedur buildSkyAngle verwiesen.

```
global proc AEx3dBackgroundTemplateNew( string $attrName )
{
    string $AENAME;
    string $nodeName[];
    tokenize($attrName, ".", $nodeName);
    setUITemplate -pst attributeEditorPresetsTemplate;

    if($nodeName[1] == "skyAngle"){
        buildSkyAngle($nodeName[0], $attrName, $AENAME);
    }
    setUITemplate -ppt;
}
```

Diese Prozedur definiert die Komponenten und die Anordnung dieser im Attribut-Editor. Darüber hinaus wird der vom Nutzer definierte Wert gespeichert um diesen dem Exportvorgang zur Verfügung zu stellen. Der Wert wird sowohl beim Definieren des skyAngle über den Schieberegler als auch bei direkter Eingabe übergeben.

```
global proc buildSkyAngle(string $nodeName, string $attrName)
{
    string $name = $nodeName+".skyAngle";
    columnLayout skyAngleCol;
        rowLayout -columnWidth2 10 350 -numberOfColumns 2;
            text -l " ";
            floatSliderGrp -label "skyAngle" -field true -min 0.0 -max 3.14
                -cc ("setAttr "+$name+" `floatSliderGrp -q -v newSkyAngle`");
            -dc ("setAttr "+$name+" `floatSliderGrp -q -v newSkyAngle`");
            newSkyAngle;
        setParent ..;
    setParent ..;
}
```

Beim Export des Background-Knotens arbeitet das RawKee Plug-in mittels Mel-Skripten als auch auf Basis von C++ Code.

In der Datei x3dBackground.cpp sind die Attribute des Knotens definiert. Nur in dieser Datei definierter Variablen können beim Export berücksichtigt werden. In der Datei web3dExportMethods.cpp wird eine Array erstellt in das alle Werte des Knotens übernommen werden. Für den skyAngle Wert wird der in Maya definierte Float-Wert dem Array hinzugefügt.

```
MStringArray web3dExportMethods::getBGFieldValues(MFnDependencyNode
depNode, unsigned int subNode)
{
    MStringArray tArray;
    MString tString("0");
    MString tString2 = getSFFloat("skyAngle", depNode);
    if(tString2.operator !=(tString)) tArray.append(tString2);
    else tArray.append("");
    return tArray;
}
```

Auf dieses Array greift die Datei x3dExportOrganizer.cpp zu um den Knoten zu exportieren. In dieser wird der Background-Knoten aufbereitet und mit den Werten aus dem erstellten Array vervollständigt. Als nächster Schritt wird der Knoten zum Export weitergeleitet. Daraufhin wird mittels der Datei sax3dWriter.cpp der Knoten in eine Datei geschrieben. Die Übergabe der Informationen wird durch die Funktion writeNodeField(..) der Datei x3dExportOrganizer.cpp realisiert.

Damit ist der Erstellungsprozess eines X3D Background-Knotens abgeschlossen und zeigt sich in der X3D-Datei folgendermaßen:

```
<?xml version="1.0" encoding="utf-8"?>
<X3D version = '3.2' profile='Immersive'>
  <Scene>
    <Transform DEF='transform1'>
      <Background DEF='background1'
        skyAngle='1.57'
        skyColor='0 0 1 0 1 1'
        groundColor='0 0.5 0'
        frontUrl='Mauer.jpg'
        containerField='children' />
    </Transform> <!-- end of Background -->
  </Scene>
</X3D>
```

1.4 X3D Text

X3D unterstützt die Visualisierung von 2D-Textobjekten. Für eine 3D-Präsentation als Vorlesungshilfsmittel wird diese Komponente benötigt, aber von RawKee noch nicht bereitgestellt. Aufgrund dessen wurde diese dem Plug-in beigelegt. Es ist daher in der erweiterten Form des Maya Plug-ins möglich Text einzugeben, ihn zu konfigurieren und im X3D-Format auszugeben. Nach der Erstellung eines Textknotens kann der Wortlaut über den Attribut-Editor eingegeben und sein Erscheinungsbild verändert werden. Die Veränderungen im Schriftbild werden mit dem X3D-Knoten FontStyle erzielt. Mit ihm kann die Größe, der Stil und die Familie der Textteile des Knotens konfiguriert werden. Bei der Erzeugung einer 2D-Textkomponente kann es dienlich sein, diese immer zum Betrachter hin auszurichten. Der

Billboard-Knoten realisiert dies in einer X3D Szene. Das Text- das Billboard- und das FontStyle-Knotenelement setzten sich nach der X3D-Spezifikation folgendermaßen zusammen:

```

Text : X3DGeometryNode {
  SFNode [in,out] fontStyle NULL [X3FontSyleNode]
  MFFloat [in,out] length [] [0,∞)
  SFFloat [in,out] maxExtent 0.0 [0,∞)
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  MFString [in,out] string []
  SFBool [] solid FALSE
}

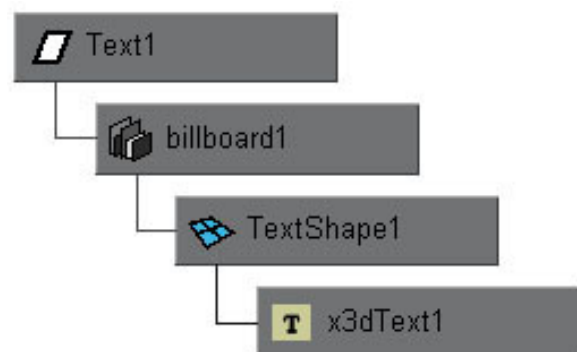
Billboard : X3DGroupingNode {
  MFNode [in] addChilden [X3DChildNode]
  MFNode [in] removeChildren [X3DChildNode]
  SFVec3f [in,out] axisOfRotation 0 1 0 (-∞,∞)
  MFNode [in,out] children [] [X3DChildNode]
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  SFVec3f [] bboxCenter 0 0 0 (-∞,∞)
  SFVec3f [] bboxSize -1 -1 -1 [0,∞) or -1 -1 -1
}

FontStyle : X3DTextStyleNode {
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  MFString [] family "SERIF"
  SFBool [] horizontal TRUE
  MFString [] justify "BEGIN" [BEGIN|END|FIRST|MIDDLE]
  SFString [] language ""
  SFBool [] leftToRight TRUE
  SFFloat [] size 1.0 (0,∞)
  SFFloat [] spacing 1.0 [0,∞)
  SFString [] style "PLAIN" [PLAIN|BOLD|ITALIC|BOLDITALIC]
  SFBool [] topToBottom TRUE
}

```

Das RawKee Plug-in bietet die Möglichkeit alle bedingten Knotenelemente in Maya zu erstellen. Bei der Erzeugung eines Textknotens werden wie in der Abbildung ersichtlich mehrere miteinander verbundene Knoten erzeugt.

Die oberste Ebene bildet ein Text-Knoten. Dieser dient der Gruppierung und zur Strukturierung innerhalb des späteren X3D-Dokuments. Dieser wird auf der Maya Bühne als kubischer Körper dargestellt. Damit ist es möglich den Text an den in der Gesamtszene gewünschten Koordinaten zu positionieren. Den Kind-Knoten des Text-Knotens bildet der Billboard-Knoten. Seiner wiederum ist der Shape-Knoten des Textelementes. Über den Gestalts-

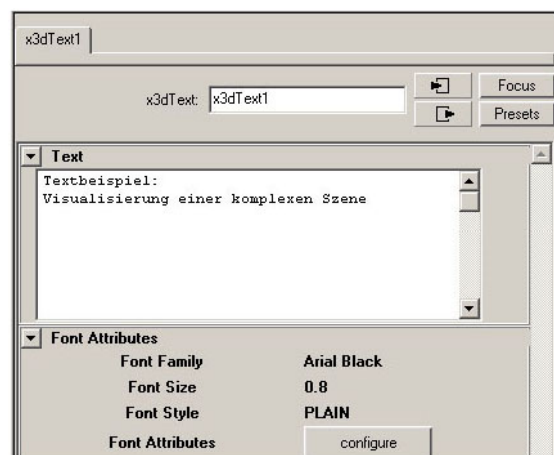


Knoten ist es möglich dem Text eine Farbe oder Textur sowie Transparenzwert und weitere Eigenschaften zuzuweisen. Somit kann jeder Textknoten individuell konfiguriert werden. Den letzten Knoten bildet der x3dTextknoten. Die Attribute dieses Knotens sind sowohl der darzustellende Text selbst als auch dessen Schriftbild. Die Abbildung der vorherigen Seite zeigt die Struktur des Text-Knotens im Dependency Graph. Der entsprechende X3D-Code ist als Vergleich im Folgenden angegeben.

```
<Transform DEF='transform1'>
  <Transform DEF='Text1' translation='0 0 0'>
    <Billboard DEF='billboard1'
      axisOfRotation='0 0 0' containerField='children'>
      <Shape DEF='TextShape1'>
        <Appearance DEF='lambert3SG'>
          <Material DEF='lambert3'
            ambientIntensity='0'
            diffuseColor='0 0 0'
            transparency='0' />
        </Appearance>
        <Text DEF='x3dText1' string='"Text:"
          "- Stichpunkt1"
          "- Stichpunkt2"'>
          <FontStyle DEF='x3d_FontStyle'
            family='Arial'
            size='1'
            style='BOLD' />
        </Text>
      </Shape>
    </Billboard>
  </Transform>
</Transform>
```

Der Billboard-Knoten kann die Darstellungsweise des Textes in der Szene beeinflussen. Im Attribut-Editor des Knotens kann diesbezüglich eine Einstellung über eine Check-Box vorgenommen werden. Durch das Auswählen dieser, ist die Option „Activate Billboard“ gesetzt und der Text wird immer zum Betrachter hin ausgerichtet. Ist die Check-Box deaktiviert wird der Text unverändert zu den Objekten der Szene dargestellt und vollzieht keine Ausgleichsbewegung bei einer Perspektivänderung.

Der Attribut-Editor des x3dText-Knotens in Maya ist nebenstehend abgebildet. Im Attribut-Fenster „Text“ besteht die Möglichkeit einen fortlaufenden Text in die 3D-Szene zu integrieren. Die Visualisierung des Textes in der X3D-Szene wird äquivalent zu der Form im Attribut-Editor dargestellt. Die Zeilenumbrüche befinden sich an den vom Nutzer definierten Stellen. Der Text kann zu einem beliebigen Zeitpunkt während der Szenenerstellung eingefügt und zu einem späteren Zeitpunkt verändert oder erweitert werden. Das Fenster „Font Attributes“ beinhaltet die vom X3D-Format unterstützten



FontStyle-Attribute. Diese Einstellungen müssen nicht vorgenommen werden. Es sind standardmäßig alle Komponenten der Schriftart definiert. Bei der Konfiguration der Schrift wird ein Optionsfenster angezeigt in dem Schriftart, -grad und -schnitt angepasst werden können. Alle Attribute bleiben bis zum Export oder einer Modifikation unverändert.

Die Programmierung des Attribut-Editors und der Export der relevanten Daten vollziehen sich grundlegend äquivalent zum Background-Knoten. Die in der Entwicklung dieses Belegs erstellte Datei `x3d_text_tables.mel` definiert die X3D-Attribute des Text-Knotens sowie deren Eigenschaft bezüglich der Eventstruktur. Das Attribut-Editor Fenster wird mittels der neu erzeugten Datei `AEx3dTextTemplate.mel` dargestellt. Es wird ein Textfenster erzeugt in dem der Text eingegeben werden kann. Bei der Deaktivierung des Fensters wird der Text automatisch einem Array übergeben welches die Daten bis zur nächsten Aktivierung des Textfensters oder dem Export speichert. Bei einer erneuten Aktivierung des Text-Knotens werden die Daten aus diesem Array neu ausgelesen. Die Schriftattribute werden ebenfalls nach durchgeführter Konfigurierung gespeichert und die gewählten Komponenten im Attribut-Editor visualisiert. Der Billboard-Knoten ist in den Dateien `x3d_billboard_tables.mel` und `AEx3d_Billboard.mel` beschrieben. Bei der Template-Erzeugung wird eine Check-Box definiert die jede Veränderung ihres Zustandes vermerkt.

Der Export der Daten wird durch C++ Code realisiert. Dabei baut der Export des Text-Knotens auf bestehenden Implementierungen auf und erweitert diese. Der Export des Transform-Knotens sowie der Text- und Shape-Knoten sind bereits implementiert wurden. Der Billboard-Knoten ist als Gruppierungsknoten definiert und nutzt bereits implementierte Exportfunktionen. Der Blattknoten `x3dText` des Graphen wurde im Zuge der Entwicklung überarbeitet und erweitert. Der Knoten wird in der Funktion `writePrimitive()` in der Datei `x3dExportOrganizer.cpp` aufbereitet und mittels der `saxWriter.cpp` ausgegeben. Der `x3dText`-Knoten ist in Maya als Blattknoten realisiert und kann damit keine weiteren Knoten gruppieren. Aufgrund dessen wurde der, dem in der X3D-Syntax dem Text-Knoten in der Struktur untergeordnete `FontStyle`-Knoten, in den Funktionsumfang des Attribut-Editors des Text-Knotens integriert und wird ebenfalls mit diesem exportiert. Die Datei `x3dExportMethods.cpp` liefert ähnlich dem Ausgabeverfahren des Background-Knotens, die für den Export notwendigen und in Maya definierten Informationen.

1.5 X3D Switch

Damit eine 3D-Präsentation in Teilszenen untergliedert werden kann, bedarf es einer Möglichkeit der Separation der Gesamtpräsentation in verschiedene Szenen. Der X3D-Switch-Knoten nimmt sich dieser Aufgabe an. Er ist ein Gruppierungsknoten der jeweils nur einen seiner Kind-Knoten visualisiert. Der Integer-Wert des `whichChoice` Attributs legt den darzustellenden Kind-Knoten fest. Nur ein aktiver Knoten wird dargestellt und kann Events empfangen oder senden. Ein Switch-Knoten hat folgende Form:

```
Switch : X3DGroupingNode {
    MFNode [in]      addChildren          [X3DChildNode]
    MFNode [in]      removeChildren      [X3DChildNode]
    MFNode [in,out] children              []          [X3DChildNode]
    SFNode [in,out] metadata              NULL       [X3DMetadataObject]
    SFInt32 [in,out] whichChoice          -1         [-1,∞)
    SFVec3f []      bboxCenter            0 0 0     (-∞,∞)
    SFVec3f []      bboxSize              -1 -1 -1  [0,∞) oder -1 -1 -1
}
```

Um einen Szenenablauf in die 3D-Präsentation zu integrieren, müssen die Kind-Knoten des Switch-Elementes nacheinander ausgeführt werden. Dies wird durch das inkrementieren der `whichChoice` Variable umgesetzt. Die Manipulation der Variable wird in Form eines ECMA-Script realisiert. Dieses Skript musste im RawKee Plug-in bisher vom Nutzer geschrieben werden. Dafür wurde ein Skript-Knoten erstellt und diesem bedingte Variablen und Funktionen hinzugefügt. Dieser Vorgang erfolgt nun automatisch. Beim Erstellen eines Switch-Knotens wird zusätzlich ein Skript-Knoten erzeugt und dieser mit notwendigen Variablendeklarationen und dem benötigten Skript versehen. Beide Knoten werden

durch einen Transform-Knoten gruppiert. Das Skript bietet dem Nutzer die Möglichkeit sowohl whichChoice zu inkrementieren als auch zu dekrementieren. Die Anzahl der im Switch-Knoten definierten Teilszenen ist dabei peripher. Das erzeugte Skript hat folgende Form:

```
<Script DEF='switchScript'>
  <field name='stepForward' type='SFTime' accessType='inputOnly' />
  <field name='stepBackward' type='SFTime' accessType='inputOnly' />
  <field name='whichChoice' type='SFInt32' accessType='outputOnly' />
  <field name='x3dSwitch' type='SFNode' accessType='initializeOnly'>
    <x3dSwitch USE='x3dSwitch'></x3dSwitch>
  </field>
  <![CDATA[ecmascript:
    function stepForward() {
      whichChoice++;
      whichChoice %=x3dSwitch.choice.length;
    }
    function stepBackward(){
      whichChoice--;
      whichChoice %=x3dSwitch.choice.length;
    }
  ]]>
</Script>
```

Im Tag „field“ erfolgt die Variablendeklaration. SFTime definiert einen Zeitwert. Dieser speichert den Startzeitpunkt der ausgewählten Szene. Die Variable SFInt32 speichert in diesem Skript den Wert der Variablen whichChoice, die zur Auswahl des Kind-Knotens dient. SFNode initialisiert einen Knoten damit dieser im Skript verwendet und im vorliegenden Fall in den Funktionen aufgegriffen werden kann. Die Funktionen stepForward() und stepBackward() beeinflussen die whichChoice Variable und können mittels Routes und Sensoren die Teilszenendarstellung umsetzen. Das Skript kann darüber hinaus in Maya durch den Nutzer erweitert oder modifiziert werden. Dadurch kann die Navigation den Bedürfnissen des Anwenders angepasst werden. Der Programmieraufwand liegt daraufhin beim Nutzer.

Der Attribut-Editor für den Switch- und Script-Knoten sind in der RawKee-Implementierung bereits integriert. Den MEL-Skripten wurde das automatische Erzeugen von Variablen und deren Initialisierung sowie das selbsttätige Erstellen des Skriptes hinzugefügt. Der Export der Daten erfolgt auf die gleiche Art und Weise wie bei dem Text- oder dem Hintergrundknoten. Die Funktion writeScript() in der Datei x3dExportOrganizer.cpp unterscheidet zwischen einem in Maya erstellten Einzelskript und einem Skript, welches im Zuge der Bildung eines Switch-Knotens erzeugt wurde. In Abhängigkeit der Erzeugungsgeschichte werden die Komponenten mithilfe der Dateien x3dExportOrganizer.cpp und saxWriter.cpp exportiert.

1.6 X3D Scalar-Interpolator

Interpolatoren sind build-in Skripte die bei der Darstellung der 3D-Szene Werte über einem definierten Wertebereich interpolieren. Position-Interpolators erzeugen Translationswerte über einen Zeitraum, wodurch eine Bewegung des Objektes im Raum realisiert werden kann. Die Interpolation erfolgt auf einem Wertetripel das X-, Y und Z-Koordinate repräsentiert. Der Scalar-Interpolator ermittelt einen einzelnen Wert über einen Zeitraum. Mit diesem Interpolator ist es beispielsweise möglich die Transparenz eines Objektes zu manipulieren und dadurch in einer 3D-Präsentation Komponenten ein- oder auszublenden. Der Scalar-Interpolator ist in der bestehenden RawKee Version nicht korrekt implementiert worden. Kleinere Fehler machte, eine einwandfreie Nutzung unmöglich. Die Fehler wurden korrigiert und der Scalar-Interpolator ist nun im vollen Umfang nutzbar.

VI Tutorium

Das Tutorium beschreibt die Erstellung einer 3D-Präsentation mit der Software Alias Maya und dem X3D Plug-in RawKee. Es erläutert die Modellierung einer Szene und die Animation einzelner Komponenten. Es wird eine Aufbauanleitung eines Tisches erstellt, wobei dieser Vorgang aus mehreren Teilschritten besteht. Das Hinzufügen von Navigationskomponenten und Interaktionsobjekten wird ebenfalls erläutert. Der letzte Abschnitt dieses Tutoriums beschreibt den Export der Szene in das X3D-Format und die Visualisierung in einem X3D-Player.

Vorraussetzungen für die Durchführung ist eine korrekt installierte Mayasoftware sowie die Installation des RawKee Plug-ins. Eine Installationsanleitung für das Plug-in ist im Kapitel III unter dem Punkt 1.4 beschrieben. Eine Installation eines X3D-Players ist für die Erstellung der Präsentation nicht zwingend notwendig, unterstützt jedoch den Entwicklungsprozess und dessen Benutzung und wird in diesem Tutorium ebenfalls beschrieben. Eine Übersicht über bestehende Software zur Darstellung von X3D-Szenen ist im Kapitel III unter Punkt 4 dargestellt. Dieses Tutorium ist jedoch keine Einführung in das Modellierungswerkzeug Maya. Der grundlegende Umgang mit dem Werkzeug wird vorausgesetzt. Unter folgenden Adressen sind Einführungstutorien in die Modellierungssoftware zu finden:

- <http://www.highend3d.com/maya/tutorials/>
- <http://www.tutorialized.com/tutorials/Maya/1>
- <http://www.learning-maya.com/modeling.php>

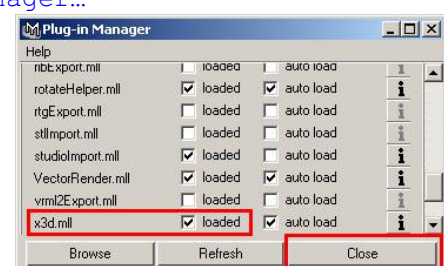


mit RawKee erstellte X3D-Präsentation

Das Tutorium beginnt mit dem Ausführen der Erstellungssoftware unter Windows.

- Start von Alias Maya durch das Ausführen der Datei Maya.exe im Maya-Installationspfad oder im Startmenü
- Einbinden des RawKee Plug-ins über
[Windows > Setting/Preferences > Plug-In Manager...](#)

- Die Checkbox mit der Bezeichnung „loaded“ der Datei x3d.mll aktivieren
 - Durch die Aktivierung der Checkbox „auto load“ wird das Plug-in bei jedem Maya-Start automatisch geladen
 - Das korrekt geladene Plug-in erzeugt den Menüeintrag RawKee (X3D) in der Menüleiste.



- Erzeugen eines Projekt-Ordners:
 - `File > Project > New`
 - Im Projekt-Fenster den Namen und den Pfad des Projektes festlegen und mit dem „Use Defaults“ Knopf bestätigen
 - Das Fenster mit „Accept“ schließen

- Import einer vorgefertigten Szene. Die zu integrierende Szene beinhaltet einen Raum indem im Laufe des Tutoriums der Tisch aufgebaut werden soll. Die zu importierende Datei Zimmer.mb befindet sich auf der dem Beleg beigefügten CD.
 - `File > Import`
 - Datei Zimmer.mb auswählen und importieren

- Beleuchtung der Szene durch ein Directional-Light
 - `RawKee (X3D) > Add Scene Nodes > DirectionalLight`
 - Ausrichtung des Lichts entlang der Y-Achse in
 - Selektieren der Lichtquelle und Aktivierung des Attribut-Editor über `Window > Attribut Editor`
 - Im Attribut-Editor „Intensity“ der Lichtquelle auf 5 einstellen und die Checkbox „Illuminates by Default“ aktivieren

- Erzeugung eines ViewPoints (Kamerasicht in der Szene)
 - `RawKee (X3D) > Add Scene Nodes > ViewPoint`
 - Das auf der Bühne erzeugte Kameraobjekt auswählen und ausrichten
 - Im Attribut Editor unter CameraShape > im Bereich „Extra Attributes“ die Sichtbezeichnung für die Kamera eintragen
 - Dieser Schritt muss wiederholt werden um weitere Kamerasichten zu definieren

- Erstellung eines Hintergrundes mit Himmel inklusive Farbverlauf und Untergrund sowie einem Environment Cube mit einem Landschaftsbild
 - `RawKee (X3D) > Add Scene Nodes > Background`
 - Durch Anklicken des Farbfeldes unter SkyColor wird die Himmelsfarbe auf Blau gesetzt
 - Im nun sichtbaren Farbverlaufsfeld wird hellblau gewählt
 - SkyAngle auf 90 Grad, setzt den Farbverlauf des Himmels vom Zenit bis zum Horizont
 - GroundColor kann auf dieselbe Art und Weise konfiguriert werden
 - Der Environment Cube wird mittels Pfadangaben realisiert
 - Für alle Seiten außer topURL und bottomURL wird das Landschaftsbild Landschaft.jpg über den „Browse“ Button ausgewählt

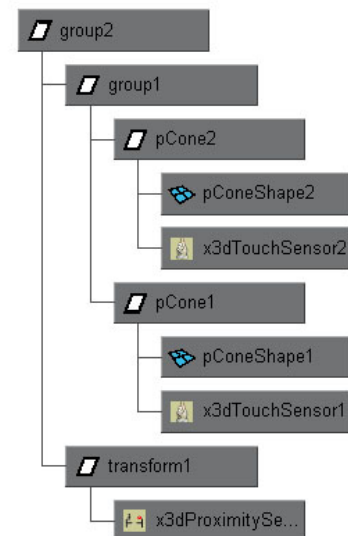
- Erstellen der Navigationsobjekte durch die Erzeugung von 2 Polygonkegel als Navigationswerkzeuge
 - Die beiden Kegel werden erzeugt
 - Der nächste Schritt besteht in der Ausrichtung (Transformieren der Kegel in Z-Richtung) und Gruppierung derer.
 - Erzeugung eines Touch-Sensors in die jeweilige Knotengruppe der beiden Kegel durch `RawKee (X3D) > Add Sensor Nodes > TouchSensor`
 - Erstellung eines Proximity-Sensors mittels `RawKee (X3D) > Add Sensor Nodes > ProximitySensor`
 - Im Attribut Editor des Proximity-Sensors unter „Extra Attributes“ den Wert für „ProxSize“ für alle Koordinaten auf 1000 setzen.
 - Erstellung von Routes zur Verknüpfung von Proximity-Sensor und Navigationspfeilen.

- Beteiligte Knoten müssen in das entsprechende Fenster auf der rechten Seite des Interaction-Editor geladen werden
- Durch Anklicken der geladenen Knoten mit der rechten Maustaste kann das Verknüpfungsattribut gewählt werden.
- Folgende Routes sind zu erstellen:

```
Route x3dProximitySensor1.position_changed
    To group1.set_translation
```

```
Route
x3dProximitySensor1.orientation_changed
    To group1.set_rotation
```

- Gruppierung der beiden Teilbäume zu einem Navigationsgraph
 - Das nebenstehende Bild zeigt die endgültige Baumstruktur
 - Durch den Touch-Sensor können die Kegelobjekte als Interaktionsgegenstand verwendet werden und der Proximity-Sensor ermöglicht die automatische Ausrichtung der Objekte zur Kamera



In den nächsten Schritten dieses Tutoriums wird die Erstellung der einzelnen Teilszenen beschrieben. Diese Szenen können in der Präsentation mit den eben erzeugten Navigationsbuttons durchlaufen werden. Es werden 2 Teilsequenzen erstellt. In der ersten Szene werden die Tischbeine in der Szene eingeblendet. Zusätzlich erscheint Text, der die Bauanleitung textuell darstellt und die präsentierte Animation verdeutlicht

➤ Erzeugung eines Switch-Knotens

```
RawKee (X3D) > Add Grouping Nodes > Switch
```

- Erzeugung von Routes, die das Skript zur Manipulation des Switch-Knotens mit den Navigation-Buttons verbinden

```
Route x3dTouchSensor1.touchTime To switchScript1.stepForward
```

```
Route x3dTouchSensor2.touchTime To switchScript1.stepBackward
```

```
Route switchScript1.whichChoice To x3dSwitch1.set_whichChoice
```

- Unterhalb des Switch-Knotens wird ein Transform-Knoten eingefügt.

```
RawKee (X3D) > Add Grouping Nodes > Transform
```

- Dieser dient als Startknoten wodurch die Animation nicht sofort nach dem Start der Anwendung ausgeführt wird.
- Darunter wird ein weiterer Transform-Knoten eingefügt. Dieser wird später die gesamte erste Szene beinhalten.

➤ Erstellen einer Keyframe-Animation über den Scalar-Interpolator

- Vier Tischbeine werden unter dem Knoten für die erste Szene erstellt und gruppiert sowie auf der Bühne positioniert.
- Erzeugung eines Time-Sensors und eines Scalar-Interpolators ebenfalls als Kind-Knoten des Transform-Knotens der ersten Teilszene

```
RawKee (X3D) > Add Sensor Nodes > TimeSensor
```

```
RawKee (X3D) > Add Animation Nodes > ScalarInterpolator
```

- Im Attribut-Editor des Scalar-Interpolators „Start Frame“ auf 1 und „End Frame“ auf 100 setzen
- „Update Keys“ Button betätigen

- Unter Extra Attributes über dem Skalar mit rechter Maustaste die Animations-Keys setzen
- Einen Key mit dem Skalarwert 1 im Frame 1 und einen weiteren Key mit dem Wert 0 auf Frame 100 setzen
 - Erstellung einer Input- Output-Connection zwischen Time-Sensor und Scalar-Interpolator
- Der Output „SecondLength“ des Interpolators wird mit dem Input „CycleIntervall“ des Sensors verbunden
- Festlegen folgender Routes sind für die Animation sind zwingend notwendig

```
Route x3dTouchSensor1.touchTime To x3dTimeSensor1.set_startTime
```

```
Route x3dTimeSensor1.fraction_changed
    To x3dScalarInterpolator1.set_fraction
```

```
Route x3dScalarInterpolator1.value_changed
    To lambert1.set_transparency
```

- Einbinden eines Textknotens in die erste Teilszene durch Einordnung des Knotens in dieselbe Gruppe wie alle bisherigen Komponenten der Szene

```
RawKee (X3D) > Add Primitives > Text
```

- Im x3dText-Knoten den Begleittext für die Animation einfügen und Schriftattribute festlegen
- Textknoten in der Szene positionieren (Werte: $x = -0.7$, $y = 0.45$, $z = -1$)
- Verknüpfung des Textobjektes mit dem Proximity-Sensor durch Routes, damit sich der Text bei jeder Translation und Rotation dem Anwender zuwendet

```
Route x3dProximitySensor1.position_changed
    To Transform8.set_translation
```

```
Route x3dProximitySensor1.orientation_changed
    To Transform8.set_rotation
```

Im letzten Abschnitt dieses Tutoriums steht die Erstellung der zweiten Teilszene im Mittelpunkt. Dabei wird die Tischplatte erstellt und durch eine Animation vom Boden des Zimmers an die endgültige Position bewegt. Ein Textknoten beschreibt ebenso wie in der ersten Teilszene die Schritte der Animation.

- Erstellung eines Transform-Knotens als Container für die Knotenelemente der zweiten Szene

```
RawKee (X3D) > Add Grouping Nodes > Transform
```

- Modellierung der Tischplatte und der Tischbeine
 - Die Tischbeine müssen erneut erstellt werden, da die Visualisierung der Tischbeine sich nur auf die jeweilige Szene bezieht und aufgrund dessen die Tischbeine aus der Szene 1 nicht mehr visualisiert werden

- Erzeugung der Keyframe-Animation

- Bei dieser Animation soll die Tischplatte vom Boden auf die Tischbeine gelegt werden. Folgende Schritte sind dafür durchzuführen:

- Erzeugung eines Time-Sensors und eines Position-Interpolators ebenfalls als Kind-Knoten des Gruppierungsknotens.

```
RawKee (X3D) > Add Sensor Nodes > TimeSensor
```

```
RawKee (X3D) > Add Animation Nodes > PositionInterpolator
```

- Im Attribut-Editor des Position-Interpolators: „Start Frame“ auf 101 und „End Frame“ 200 setzen
- „Update Keys“ Button betätigen

- Bei Frame 101 liegt die Tischplatte am Boden und ein Key wird gesetzt
- In Frame 200 ist die Tischplatte an der endgültigen Position und ein Key wird gesetzt
 - Erstellung einer Input- Output-Connection zwischen Time-Sensor und Position-Interpolator
- Dazu ist der Output SecondLength des Interpolators mit dem Input CycleIntervall des TimeSensors zu verbinden
 - Erstellung einer Input- Output-Connection zwischen Tischplatte und dem Scalar-Interpolator
- Der Output translate der Tischplatte muss mit dem Input position des Position-Interpolators verbunden werden
- Erstellung der Routes für die Animation:

```
Route x3dTouchSensor1.touchTime To x3dTimeSensor1.set_startTime
```

```
Route x3dTimeSensor2.fraction_changed
      To x3dPositionInterpolator1.set_fraction
```

```
Route x3dPositionInterpolator1.value_changed
      To Tischplatte.set_translation
```

- Einbinden eines Textknotens in die zweite Teilszene durch Gruppierung des Knotens in die aktuelle Gruppe der Szene

```
RawKee (X3D) > Add Primitives > Text
```

- Im x3dText-Knoten Text einfügen und Schriftattribute festlegen
- Textknoten in der Szene positionieren(Werte: x= -0.7, y=0.45, z= -1)
- Verknüpfung des Textobjektes mit dem Proximity-Sensor durch Routes

```
Route x3dProximitySensor1.position_changed
      To transform12.set_translation
```

```
Route x3dProximitySensor1.orientation_changed
      To transform12.set_rotation
```

Damit ist die Erstellung der 3D-Präsentation abgeschlossen. Sie kann daraufhin exportiert und damit in das X3D-Format konvertiert werden. Die Einstellungen für den Export müssen vor diesem festgelegt werden. Die Konfiguration erfolgt in den Exportheigenschaften der Modellierungssoftware. In den Einstellungen ist als Datei-Typ für den Export das RawKee X3D XML Format zu wählen. Weitere Einstellungen können bei diesem Tutorium unverändert bleiben. Daraufhin kann die Szene unter Angabe eines Pfades exportiert und in einer entsprechenden Abspielsoftware visualisiert werden. Durch Ausführen der exportierten Datei wird die standardisierte Wiedergabesoftware des Systems gestartet. Manche der derzeit verfügbaren Player können einige Komponenten der Szene nur ungenügend oder gar nicht darstellen. Die Software Octaga- und FLux-Player stellen die 3D-Präsentation korrekt dar. Die Szene ist im Octaga-Player dabei im vollen Umfang dargestellt und der Flux-Player stellt die Szene bis auf den Hintergrund ebenfalls umfassend dar. Jedoch unterstützt dieser die Unterteilung des Hintergrundes in Himmel und Boden nicht, sondern stellt über den gesamten Hintergrund die Himmelsfarbe dar.

VII Zusammenfassung und Ausblick

Die Zielsetzung dieser Arbeit ist die Entwicklung eines Werkzeugs zur Erstellung von 3D-Präsentationen. Diese sollen im Rahmen von Vorlesungen im universitären Bereich eingesetzt werden können. Die Speicherung einer Präsentation in einer einzelnen Datei soll die Loslösung der 3D-Szene aus dem Werkzeug ermöglichen und eine Darstellung in einer vom Werkzeug unabhängigen Software sicherstellen. Kein bestehendes Werkzeug entspricht im vollen Umfang den Anforderungen die an die Software gestellt werden. Nach der Betrachtung und Auswertung bestehender Erstellungswerkzeuge ist die Weiterentwicklung des RawKee Plug-ins zentrale Komponente der Arbeit gewesen. Das Plug-in bietet die Möglichkeit der Weiterentwicklung und Spezialisierung aufgrund seines Open Source Statuses. Dadurch konnten Komponenten hinzugefügt werden, die für die Umsetzung der Forderungen an die Software notwendig waren. Das Plug-in bietet zusammen mit der Modellierungssoftware Maya die Möglichkeit 3D-Präsentationen zu erstellen und diese zu exportieren. Der Funktionsumfang des Plug-ins deckt die Anforderungen zufrieden stellend. Es bestehen dennoch Optimierungsmöglichkeiten um die Arbeit mit dem RawKee Plug-in effizienter zu gestalten.

Die Funktionsumfänge von Maya und dem Plug-in sind nicht deckungsgleich. Dadurch ergibt sich für den Nutzer die Möglichkeit Einstellungen vorzunehmen, die nicht in das X3D-Format übernommen werden können. Es würde die Arbeit des Nutzers erleichtern nur mit Komponenten arbeiten zu können, die auch in das X3D-Format konvertiert werden. Nach dem Export einer 3D-Präsentation aus dem Erstellungswerkzeug erfolgt die Darstellung der 3D-Szene in einer X3D-Abspielsoftware. Der unterstützte und damit darstellbare Komponentenumfang einer 3D-Szene ist vom jeweiligen X3D-Player abhängig. Dadurch sehen erstellte 3D-Szenen in den Viewern mitunter verschieden aus. Dies erschwert die Arbeit des Nutzers, da dieser auf den Funktionsumfang der Player bereits bei der Erstellung einer Szene Rücksicht nehmen muss.

Die 3D-Präsentation als vorlesungsunterstützendes Hilfsmittel verlangt die Integration von Text in die Szene. Der Text sollte in Form eines HUD's dargestellt werden können. Im X3D-Format ist dies nicht problemlos umsetzbar. Derzeit wird dies mit Hilfe von Textkomponenten realisiert, die an Proximity-Sensoren gekoppelt sind. Damit ist der Text ein Teil der 3D-Szene und eben auch in das Koordinatensystem dieser integriert. Die Darstellung von Text ohne Verdeckung durch andere Körper der Szene ist deshalb schwierig. In einer Veröffentlichung des Web3D-Consortiums im August 2006 wird eine Erweiterung des X3D-Standards bekannt gegeben. In dieser wird die Definition von Layern beschrieben. Durch Layer ist es möglich 2D und 3D-Komponenten in einer Szene zu verbinden, die Komponenten jedoch gekapselt voneinander zu erstellen. Die Erzeugung von HUD's wird dadurch stark vereinfacht. Im Augenblick werden die Komponenten jedoch von keiner Abspielsoftware unterstützt.

Das RawKee Plug-in bietet zusammen mit dem Modellierungswerkzeug Maya und einem X3D-Player die Möglichkeit der Erstellung von 3D-Präsentationen und ermöglicht dadurch komplexe Vorgänge und Strukturen in Vorlesungskontexten zu verdeutlichen.

Die in der Aufgabenstellung formulierten Ansprüche an eine solche Software werden durch das RawKee Plug-in erfüllt. Damit ist es als zusätzliches Hilfsmittel in der Unterrichtsgestaltung einsetzbar.

Glossar

-- A --

ActiveX	Softwarekomponenten-Modell von Microsoft für aktive Inhalte. ActiveX-Komponenten erweitern die COM-Standards von Microsoft.
AGL	Graphikbibliothek, die OpenGL und Mac OS verbinden
Animation	im engeren Sinne jede Technik, mit der einzelbildweise Bewegung im Film geschaffen wird.
API	Schnittstelle, die von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird

-- B --

Behaviorismus	Lernen als Konditionierungsvorgang
Binärformat	Dateiformat, welches nicht mit allgemein verfügbaren Programmen eingesehen werden kann

-- C --

CAD	Computer Aided Design (CAD) bezeichnet eine Art „elektronisches Zeichenbrett“ zur Erstellung meist technischer Zeichnungen
CGL	Graphikbibliothek, die OpenGL und Mac OS verbinden
Compiler	Programm, mit dem symbolische Programmiersprachen in lauffähige Maschinensprachen umgesetzt werden
Computergrafik	Teilbereich der Informatik, der sich mit der künstlichen Erzeugung von Bildern mit Hilfe von Computern beschäftigt

-- D --

Didaktik	Theorie des Unterrichts und ebenso Theorie und Praxis des Lehrens und Lernens.
Direct3D	Programmierschnittstelle (API) von Microsoft für 3D-Computergrafik
DirectX	Sammlung von APIs für Multimediaprogramme auf der Windows-Plattform
DTD	Dokumenttypdefinition ist eine Deklaration in SGML- und XML-Dokumenten, die die Struktur eines solchen Dokumentes festlegt.

-- E --

E-Commerce	virtueller Einkaufsvorgang via Datenfernübertragung
ECMA-Script	JavaScript – objektorientierte Skriptsprache

-- F --

Framework	Softwarearchitektur der Anwendung und stellt für die Entwicklung Bausteine für die Designgrundstruktur zur Verfügung.
-----------	---

-- G --

GLUT	OpenGL-Bibliothek die eine plattformunabhängige API für Ein-/Ausgabe, Erstellen von Rendering-Kontexten und dergleichen bietet
GLX	Schnittstelle die das Interface zwischen dem X Window System und OpenGL bildet
grafische Benutzeroberfläche	ist eine Softwarekomponente, die einem Computerbenutzer die Interaktion mit der Maschine über grafische, metaphorhafte Elemente unter Verwendung eines Zeigegerätes oder Tastatur erlaubt.

-- I --

Interaktion bezeichnet das wechselseitige aufeinander Einwirken von Akteuren und/oder Systemen.

-- J --

Java 3D Bibliothek von Java-Klassen zur Erzeugung, Manipulation und Darstellung dreidimensionaler Grafiken innerhalb von Java-Applikationen und -Applets.

JavaScript objektorientierte Skriptsprache

-- K --

Kognitivismus Lernen als vielschichtiger Prozess der Informationsverarbeitung

Konstruktivismus Lernen, kein passives Aufnehmen und Abspeichern von Informationen, sondern aktiver Prozess der Wissenskonstruktion

-- M --

Maya Alias Maya - 3D Visualisierungs- und Animationssoftware, die primär in der Film- und Fernsehindustrie und bei der Erstellung von Grafiken für Computer- und Videospiele eingesetzt wird.

MEL (Maya Embedded Language) Maya-Interne Steuerungssprache
Multimedia Inhalte und Werke, die aus mehreren der folgenden digitalen Medien bestehen können: Text, Fotografie, Grafik, Animation, Audio und Video.

-- N --

NURBS (Non-Uniform Rational B-Splines) sind mathematisch definierte Kurven oder Flächen

-- O --

OpenGL (Open Graphics Library) Spezifikation für ein plattform- und programmiersprachenunabhängiges API (Application Programming Interface) zur Entwicklung von 3D-Computergrafik.

Open Inventor C++ objektorientierte 3D-Graphik API zur besseren Programmierbarkeit von OpenGL

Open-Source ermöglicht den Einblick in den Quelltext eines Programms, sowie die Erlaubnis diesen Quellcode beliebig weiterzugeben oder zu verändern.

-- P --

Player Abspielsoftware für digitale Medien

Plug-in Bezeichnung für ein Computerprogramm, das in ein anderes Softwareprodukt "eingeklinkt" wird.

Polygon Vieleck - eine geschlossene Figur

-- R --

Rendern Vorgang, aus Rohdaten - durch Anwendung geeigneter Verfahren - neue Daten zu generieren.

-- S --

Standalone Anwendung Anwendung die keinen Webbrowser zur Darstellung der Daten benötigt
Streaming kontinuierliche Übertragung von Daten, das heißt die Übertragung von Datenströmen

Studio Max 3ds Max ist ein 3D Grafik- und Animationsprogramm, welches im Bereich Computerspiele, Comic, Animationen, Film Verwendung findet

-- T --

Toolkit im Software-Bereich eine Sammlung von Programmen

-- V --

Viewer Computerprogramm, mit dem sich Computerdateien anschauen, aber nicht verändern lassen

VRML (Virtual Reality Modeling Language) ist eine Beschreibungssprache für 3D-Szenen, deren Geometrien, Ausleuchtungen, Animationen und Interaktionsmöglichkeiten.

-- W --

Webbrowser spezielle Computerprogramme zum Betrachten von Webseiten im World Wide Web.

WGL Graphikbibliothek, die OpenGL und Windows verbindet

-- X --

X3D (Extensible 3D) 3D-Modellierungssprache, die in der Syntax auf XML aufbaut und mit der sich dreidimensionale virtuelle Welten, Spiele und interaktive Lernanwendungen realisieren lassen.

XML (Extensible Markup Language) Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur, der vom World Wide Web Consortium (W3C) definiert wird.

Anhang

A Technische Voraussetzungen

Im Rahmen der Entwicklung sind die folgenden Software-Versionen verwendet worden:

- Alias Maya 6.0
- Alias Maya 7.0
- Macromedia Director 8.5G
- Macromedia Director MX2004
- Cult3D Designer 5.3
- Eonstudio 4.0 Trail Version
- Octaga Player 2.0.
- Flux Player 1.2
- Flux Player 2.0 Beta Version
- Xj3D 1.0
- BS Contact Stereo 7.0
- BS Contact 6.2
- Java JRE 1.4.2.11
- Microsoft Visual Studio 2003
- XML Spy 2006
- RawKee 1.0 Beta – 1.0.3 (Stand 10.08.2006)
- Subversion 1.3.2
- BitRock InstallBuilder 3.8.0
- WinMerge 2.4.8.0
- Internet Explorer 6.0
- Mozilla Firefox 1.5

B Codeentwicklung

Folgende MEL- und C++ Dateien wurden gegenüber der Version 1.0.3 des RawKee Plug-ins modifiziert.

Dateiname	Codezeilen	Beschreibung
x3d.mel	295,312, 408-409 329-333 411-415 906-914	Erstellung von Menüeinträgen Korrektur an Menüverweisen Definition des Vorschau-Hotkey Vorschaufunktion - showPreview
x3d_scenegraph_ui_tree.mel	245,260	Menüpunkte im Interaction-Editor erstellt
x3d_ie_menus.mel	179-185 505-515	Aufruf Background\Text EventOut Menü Aufruf Background\Text EventIn Menü
x3d_node_creation_procs.mel	231-250 451-455 1213-1285	Erzeugung des Switch-Knotens Erzeugung Underworld-Textknoten Erzeugung von Background-, Textknoten
x3d_exporter_procedures.mel	472-474	Relative Koordinaten des Textknotens bestimmen
AEx3dScriptTemplate.mel	1777-1783 2011-2018 2095-2125 2139-2177	Initialisierung der Skriptvariablen Initialisierung der Skripterstellung Hinzufügen des Skripts Übergeben der Daten
AEx3dTextTemplate.mel	Neu erstellt	Erzeugung des Text Attribut-Editors
x3d_text_tables.mel	Neu erstellt	Erzeugung der Text Routevariablen
AEx3dBackgroundTemplate.mel	Neu erstellt	Erzeugung des Background Attribut-Editors
x3d_background_tables.mel	Neu erstellt	Erzeugung der Background Routevariablen
AEx3dBillboardTemplate.mel	Neu erstellt	Erzeugung des Billboard Attribut-Editors
x3d_billboard_tables.mel	Neu erstellt	Erzeugung der Billboard Routevariablen
x3d_scalarinterpolator_tables.mel	28-67	Überarbeitung der Routevariablen


Dateiname	Codezeilen	Beschreibung
web3dExportMethods.h	53-55 133-136 290 398-400,445-447	Include-Definition Knotendefinition Funktionsdefinition Arraydefinitionen
web3dExportMethods.cpp	139-150,405-416 856-893 4306-4507 6000-6016	Arraydefinition Text und Hintergrund Attribute festlegen Ermitteln der Text und Hintergrundwerte Transparenzfunktion überarbeitet
webX3DExporter.cpp	366-368 384-385,437 1560-1562	Laden der MEL-Skripte für Text und Hintergrund Registrierung des Text- und Hintergrundknotens Entkopplung des Text und Hintergrundknotens
x3dExportOrganizer.h	251	Funktionsdefinition
x3dExportOrganizer.cpp	2819, 2969 2917 4225-4442 4444-4513 5685-5886 6713-6715	Definition des Billboard-Knotens als Gruppenknoten Funktionsaufruf für Hintergrundknoten Einbindung des Textknoten in den Mesh Export Einbindung des Textknoten in den Primitive-Export Einbindung des Switch-Skriptes in den Skript Export Funktion zum Vergleich von Maya und X3D Knoten
x3dText.h	Neu erstellt	Deklarationen des Text Objektes
x3dText.cpp	Neu erstellt	Definition des Maya Objektes Text
x3dBackground.h	Neu erstellt	Deklarationen des Background Objektes
x3dBackground.cpp	Neu erstellt	Definition des Maya Objektes Background
x3dBillboard.h	Neu erstellt	Deklarationen des Billboard Objektes
x3dBillboard.cpp	Neu erstellt	Definition des Maya Objektes Billboard
sax3dWriter.h	86-88	Deklarationen für Export von Text und Hintergrund

C Dateiübersicht

Alle in den folgenden Tabellen aufgeführten Dateien sind MEL-Skripte beziehungsweise C++ Source Codedateien, die für die Erstellung und den Ablauf des RawKee Plug-ins erforderlich sind. Alle Dateien befinden sich auf der, dieser Belegarbeit beiliegenden, CD.



















MEL-Dateien:




















 AEx3dAnchorTemplate.mel	 AEx3dRouteTemplate.mel
 AEx3dBooleanFilterTemplate.mel	 AEx3dScalarInterpolatorTemplate.mel
 AEx3dBooleanSequencerTemplate.mel	 AEx3dScriptTemplate.mel
 AEx3dBooleanToggleTemplate.mel	 AEx3dSoundTemplate.mel
 AEx3dBooleanTriggerTemplate.mel	 AEx3dSphereSensorTemplate.mel
 AEx3dBoxTemplate.mel	 AEx3dSphereTemplate.mel
 AEx3dCollisionTemplate.mel	 AEx3dStringSensorTemplate.mel
 AEx3dColorInterpolatorTemplate.mel	 AEx3dSwitchTemplate.mel
 AEx3dColorRGBATemplate.mel	 AEx3dTextureCoordinateTemplate.mel
 AEx3dColorTemplate.mel	 AEx3dTimeSensorTemplate.mel
 AEx3dConeTemplate.mel	 AEx3dTimeTriggerTemplate.mel
 AEx3dCoordinateInterpolatorTemplate.mel	 AEx3dTouchSensorTemplate.mel
 AEx3dCoordinateTemplate.mel	 AEx3dVisibilitySensorTemplate.mel
 AEx3dCylinderSensorTemplate.mel	 AEx3dWorldInfoTemplate.mel
 AEx3dCylinderTemplate.mel	 x3d.mel
 AEx3dGamepadSensorTemplate.mel	 x3dUtilCommands.mel
 AEx3dGroupTemplate.mel	 x3d_anchor_tables.mel
 AEx3dIndexedFaceSetTemplate.mel	 x3d_appearance_tables.mel
 AEx3dInlineTemplate.mel	 x3d_audioclip_tables.mel
 AEx3dIntegerSequencerTemplate.mel	 x3d_booleanfilter_tables.mel
 AEx3dIntegerTriggerTemplate.mel	 x3d_booleansequencer_tables.mel
 AEx3dKeySensorTemplate.mel	 x3d_booleantoggle_tables.mel
 AEx3dLoadSensorTemplate.mel	 x3d_booleantrigger_tables.mel
 AEx3dMetadataDoubleTemplate.mel	 x3d_box_tables.mel
 AEx3dMetadataFloatTemplate.mel	 x3d_collidablesshape_tables.mel
 AEx3dMetadataIntegerTemplate.mel	 x3d_collision_tables.mel
 AEx3dMetadataSetTemplate.mel	 x3d_collisioncollection_tables.mel
 AEx3dMetadataStringTemplate.mel	 x3d_collisionsensor_tables.mel
 AEx3dNavigationInfoTemplate.mel	 x3d_collisionspace_tables.mel
 AEx3dNormalInterpolatorTemplate.mel	 x3d_color_tables.mel
 AEx3dNormalTemplate.mel	 x3d_colorinterpolator_tables.mel
 AEx3dOrientationInterpolatorTemplate.mel	 x3d_colorrrgba_tables.mel
 AEx3dPlaneSensorTemplate.mel	 x3d_cone_tables.mel

 AEx3dPositionInterpolatorTemplate.mel
 AEx3dProximitySensorTemplate.mel
 x3d_cylinder_tables.mel
 x3d_cylindersensor_tables.mel
 x3d_directionallight_tables.mel
 x3d_exporter_procedures.mel
 x3d_gamepadsensor_tables.mel
 x3d_generic_tables.mel
 x3d_group_tables.mel
 x3d_hanimhumanoid_tables.mel
 x3d_hanimjoint_tables.mel
 x3d_hanimsite_tables.mel
 x3d_ie_menus.mel
 x3d_imagetexture_tables.mel
 x3d_indexedfaceset_tables.mel
 x3d_inline_tables.mel
 x3d_integersequencer_tables.mel
 x3d_integertrigger_tables.mel
 x3d_keysensor_tables.mel
 x3d_loadsensor_tables.mel
 x3d_lod_tables.mel
 x3d_material_tables.mel
 x3d_metadatadouble_tables.mel
 x3d_metadatafloat_tables.mel
 x3d_metadatainteger_tables.mel
 x3d_metadataset_tables.mel
 x3d_metadatastring_tables.mel
 x3d_movietexture_tables.mel
 x3d_multitexture_tables.mel
 x3d_multitexturecoordinate_tables.mel
 x3d_multitexturetransform_tables.mel
 x3d_navigationinfo_tables.mel
 x3d_node_creation_procs.mel
 x3d_coordinate_tables.mel
 x3d_coordinateinterpolator_tables.mel
 x3d_normal_tables.mel
 x3d_normalinterpolator_tables.mel
 x3d_orientationinterpolator_tables.mel
 x3d_pixelttexture_tables.mel
 x3d_planesensor_tables.mel
 x3d_pointlight_tables.mel
 x3d_positioninterpolator_tables.mel
 x3d_proximitysensor_tables.mel
 x3d_rigidbody_tables.mel
 x3d_rigidbodycollection_tables.mel
 x3d_routing_scripts.mel
 x3d_scalarinterpolator_tables.mel
 x3d_scenegraph_ui_tree.mel
 x3d_script_tables.mel
 x3d_shape_tables.mel
 x3d_sound_tables.mel
 x3d_sphere_tables.mel
 x3d_spheresensor_tables.mel
 x3d_spotlight_tables.mel
 x3d_stringsensor_tables.mel
 x3d_switch_tables.mel
 x3d_texturecoordinate_tables.mel
 x3d_texturetransform_tables.mel
 x3d_timesensor_tables.mel
 x3d_timetrigger_tables.mel
 x3d_touchsensor_tables.mel
 x3d_transform_tables.mel
 x3d_viewpoint_tables.mel
 x3d_visibilitysensor_tables.mel
 x3d_worldinfo_tables.mel

C++ Dateien

 cmdWeb3dOutputW01.h
 impl.h
 sax3dWriter.cpp
 sax3dWriter.h
 vrm197FileTranslator.cpp
 vrm197FileTranslator.h
 web3dExportMethods.cpp
 web3dExportMethods.h
 web3dFileTranslator.cpp
 web3dFileTranslator.h
 webX3DExporter.cpp
 webX3DExporter.h
 x3dAnchor.cpp
 x3dAnchor.h
 x3dBooleanFilter.cpp
 x3dBooleanFilter.h
 x3dBooleanSequencer.cpp
 x3dBooleanSequencer.h
 x3dBooleanToggle.cpp
 x3dBooleanToggle.h
 x3dBooleanTrigger.cpp
 x3dBooleanTrigger.h
 x3dBox.cpp
 x3dBox.h
 x3dCollision.cpp
 x3dCollision.h
 x3dColor.cpp
 x3dColor.h
 x3dColorInterpolator.cpp
 x3dColorInterpolator.h
 x3dColorRGBA.cpp
 x3dColorRGBA.h
 x3dCone.cpp
 x3dCone.h
 x3dCoordinate.cpp
 x3dCoordinate.h
 x3dCoordinateInterpolator.cpp
 x3dCoordinateInterpolator.h
 x3dCylinder.cpp
 x3dCylinder.h
 x3dCylinderSensor.cpp
 x3dCylinderSensor.h
 x3dExportOrganizer.cpp
 x3dExportOrganizer.h
 x3dFileTranslator.cpp
 x3dFileTranslator.h
 x3dGamepadSensor.cpp
 x3dGamepadSensor.h
 x3dGroup.cpp
 x3dGroup.h
 x3dIndexedFaceSet.cpp
 x3dIndexedFaceSet.h
 x3dInline.cpp
 x3dInline.h
 x3dIntegerSequencer.cpp
 x3dIntegerSequencer.h
 x3dIntegerTrigger.cpp
 x3dIntegerTrigger.h
 x3dKeySensor.cpp
 x3dKeySensor.h
 x3dLoadSensor.cpp
 x3dLoadSensor.h
 x3dMetadataDouble.cpp
 x3dMetadataDouble.h
 x3dMetadataFloat.cpp
 x3dMetadataFloat.h
 x3dMetadataInteger.cpp
 x3dMetadataInteger.h
 x3dMetadataSet.cpp
 x3dMetadataSet.h
 x3dMetadataString.cpp
 x3dMetadataString.h

 x3dMetadataString.h
 x3dNavigationInfo.cpp
 x3dNavigationInfo.h
 x3dNormal.cpp
 x3dNormal.h
 x3dNormalInterpolator.cpp
 x3dNormalInterpolator.h
 x3dOrientationInterpolator.cpp
 x3dOrientationInterpolator.h
 x3dPlaneSensor.cpp
 x3dPlaneSensor.h
 x3dPositionInterpolator.cpp
 x3dPositionInterpolator.h
 x3dProximitySensor.cpp
 x3dProximitySensor.h
 x3dProximitySensorManip.cpp
 x3dProximitySensorManip.h
 x3dRoute.cpp
 x3dRoute.h
 x3dScalarInterpolator.cpp
 x3dScalarInterpolator.h
 x3dScript.cpp
 x3dScript.h
 x3dSound.cpp
 x3dSound.h

 x3dSphere.cpp
 x3dSphere.h
 x3dSphereSensor.cpp
 x3dSphereSensor.h
 x3dStringSensor.cpp
 x3dStringSensor.h
 x3dSwitch.cpp
 x3dSwitch.h
 x3dTextureCoordinate.cpp
 x3dTextureCoordinate.h
 x3dTimeSensor.cpp
 x3dTimeSensor.h
 x3dTimeTrigger.cpp
 x3dTimeTrigger.h
 x3dTouchSensor.cpp
 x3dTouchSensor.h
 x3dVisibilitySensor.cpp
 x3dVisibilitySensor.h
 x3dWorldInfo.cpp
 x3dWorldInfo.h
 x3dbFileTranslator.cpp
 x3dbFileTranslator.h
 x3dvFileTranslator.cpp
 x3dvFileTranslator.h

D Literatur und Internetquellen

[Einführung]

Virtuelle Lernumgebungen Prof. Dr. Friedrich Institut für Software und Multimediatechnik, TUD
Vorlesung über die Erstellung virtueller Lernumgebungen

[3D-Techniken]

<http://de.wikipedia.org/wiki/Hauptseite>
Definitionen der 3D-Techniken

<http://www1.in.tum.de/twiki/pub/Cleanup/AugmentedRealityWiSe2004/L4-Nov23-04-APIs-3D.pdf>
Einführung in die 3D-API's Open Inventor, VRML, X3D, Java3D

<http://www.ifi.unizh.ch/~noser/COURSES/grLanguages.pdf>
Grundlagen zu OpenGL, VRML, X3D, Java3D

http://www.ikg.uni-hannover.de/lehre/katalog/Geodatenvisuallisierung_II/pdf_files_projekte/3D_Techniken_vort.pdf
Vorstellung und Vergleich von 3D-Techniken

<http://www.kfa-juelich.de/zam/files/docs/vortraege/web3d.pdf>
Präsentationsmöglichkeiten von 3D-Szenen werden aufgezeigt und erläutert

http://www-mmt.inf.tu-dresden.de/Lehre/Wintersemester_99_00/Hauptseminar/details/AusarbeitungX3D_Dachsel.pdf
Vorstellung von VRML und die Weiterentwicklung zu X3D

<http://www-aisg.cis.rl.ac.uk/VVECC/proceed/webgraphics/materials/steed/>
Vorstellung von VRML und die Weiterentwicklung zu X3D

http://www-ra.informatik.uni-tuebingen.de/lehre/ws02/pro_internet_ausarbeitung/proseminar_renner_ws02.pdf
Erläuterungen zur Funktionalität von VRML und X3D und Abgrenzung der Techniken voneinander

[OpenGL]

Computergrafik und OpenGL - Dieter Orlamünder, Wilfried Mascolus - Eine systematische Einführung
Definitionen der 3D-Techniken

<http://web.inf.tu-dresden.de/~wm1/Mascolus/cg.pdf/OpenGL.pdf>
Prinzipien von OpenGL und Einführung in die Programmierung

<http://www.it.fht-esslingen.de/~schmidt/vorlesungen/vr/seminar/ws9899/opengl.html#OPENGL-EINL>
OpenGL Tutorial

[Open Inventor]

<http://www.cg.tuwien.ac.at/studentwork/VRSem96/Inventor/>
Einführung und Funktionsweise von Open Inventor

http://users.informatik.uni-halle.de/~bugdoll/lehre/oiv_vorlesung01.pdf
Einführung und Szenengraph des Graphiktoolkits Open Inventor

<http://www.vis.uni-stuttgart.de/img/sommer/seminar/Vortrag-OpenInventor.pdf>
Übersicht über Open Inventor im Gesamtkontext mit Einschätzung

[Java3D]

<http://www.gdv.informatik.uni-frankfurt.de/lehre/ss2006/Folien/GDV/2006-06-20-Einfuehrung-Java3D.pdf>
Einführung und Funktionsweise von Java3D

<http://graphics.cs.uni-sb.de/Courses/ws9900/cg-seminar/Ausarbeitung/Florian.Obermeier/index.html>
Aufbau, Möglichkeiten und Grenzen von Java3D

[VRML]

http://www.gdv.informatik.uni-frankfurt.de/lehre/ss2001/Uebungsblaetter/GDV_folien_01_VRML_einfuehrung.pdf
Einführung in VRML

<http://www.wicg.informatik.uni-rostock.de/archiv/Semester/SS02/ISemIAM02/vrml/index.html#inter>
Animationserstellung in VRML

http://www.medien.ifi.lmu.de/fileadmin/mimuc/mt_ws0304/material/mt9b.pdf
Erläuterungen zu Interpolatoren Sensoren und Routes sowie Beispielszenen

<http://www.cs.vu.nl/~eliens/documents/vrml/reference/BOOK.HTM>
VRML 97 Referenzbuch

[X3D]

- <http://hclab.uniud.it/let-web3d/x3d-intro.pdf>
Einführung in das X3D-Format
- <http://www.realism.com/x3d/presentations/Web3D-2002/Authoring.ppt>
Vorstellung von X3D-Profilen Knotentypen und Erzeugung von X3D-Welten
- <http://www.ifi.unizh.ch/~noser/COURSES/grLanguages.pdf>
Grundlagen zu OpenGL, VRML, X3D, Java3D
- http://www.mmt.inf.tu-dresden.de/Lehre/Wintersemester_05_06/Vo_MMS/Uebung/Einfuehrung_A4.pdf
X3D Profile sowie Erläuterungen zu Interpolatoren Sensoren und Animationen
- <http://www.cs.kent.edu/~arvind/teaching/courses/distlang-presentations/x3d-group.ppt>
struktureller Aufbau von X3D-Dokumenten
- <http://www.web3d.org/>
Internetseiten des Web3D Konsortiums mit Informationen über X3D und VRML
- http://typo3.cs.uni-paderborn.de/fileadmin/Informatik/AG-Domik/teaching/seminar/3d_grafik_im_web/3D_Formate_im_Web_X3D.pdf
Vorstellung von X3D und die Veränderungen gegenüber VRML

[Cult3D]

- <http://www.cult3d.com>
Homepage von Cult3D / Download der Software sowie erläuternde Texte und Tutorien

[EON Studio]

- <http://www.eonreality.com/developer/>
Downloadseite für Updates und Demos des EON Studios
- <http://www.gup.uni-linz.ac.at/vrcave/VRIC051123-EON.pdf>
Einführung in die EON Studio Oberfläche und Knotentypen
- <http://www.5dt.com/downloads/3rdparty/eonstudio.pdf>
EON Studio Produktbeschreibung
- http://www.eonstudio.de/index_frames.html
Internetpräsenz von EON Reality mit Produktinformationen und Präsentationen
- http://www.e-mediate.nl/nieuws/distribution/eon_reality/eon_whitepaper.pdf
EON Reality White Paper

[Macromedia Director]

- Das Einsteigerseminar Macromedia Director 8.5 Tobias Hauser, Christian Wenz 1.Auflage
Einführung in Lingo, die Funktionsweise und Oberfläche von Macromedia Director
- http://www.mmt.inf.tu-dresden.de/Lehre/Sommersemester_05/Vo_WME/Skript/index.asp?p=vorl_2.pdf
Vorstellung von Werkzeugen zur Erstellung multimedialer Anwendungen
- <http://www.fbe.unsw.edu.au/learning/director/>
Macromedia Director Tutorien
- http://download.macromedia.com/pub/documentation/de/director/mx2004/drmx2004_scripting_ref.pdf
Macromedia Director Skriptreferenz
- <http://www.adobe.com/de/>
Internetpräsenz von Adobe mit Produktinformationen zu Director
- http://asi-www.informatik.uni-hamburg.de/themen/mi/lehre/mi1/skripte/Director/Director%203D_Gim.ppt
Einführung in Director und Lingo

[RawKee]

- <http://rawkee.sourceforge.net/>
Homepage des RawKee Plug-ins mit aktuellen Informationen und Tutorien
- <http://accad.osu.edu/~aprice/courses/694/DG.htm>
Kurzbeschreibung des Dependency Graphs in Maya
- <http://www.web3d.org/x3d/content/X3dTooltipsGerman.html>
Übersicht über die X3D Knotenelemente und deren Attribute
- Maya X3D-Export RawKee Tutorial Tilo Malchow Technische Universität Dresden
Erstellung einer 3D-Präsentation mit RawKee
- http://www.mmt.inf.tu-dresden.de/Lehre/Wintersemester_05_06/Vo_MMS/Uebung/Einfuehrung_A4.pdf
Vorstellung von X3D und RawKee (Knotentypen und Anwendungsszenarien)
- <http://web.inf.tu-dresden.de/~ag-vmw/>
Einführung in Maya sowie Tutorien

<http://web.inf.tu-dresden.de/~ag-vmw/>
Einführung in Maya sowie Tutorien

[VRML/X3D Player]

<http://cic.nist.gov/vrml/vbdetect.html>
Übersicht über die aktuellen VRML und X3D Player und Verknüpfungen zu Downloads

[Maya Programmierung]

Die C++ Programmiersprache Bjarne Stroustrup 4. aktualisierte Auflage

Vorstellung und Beschreibung der Programmiersprache C++

MEL Scripting for Maya Animators Mark R. Wilkins, Chris Kazmier

Beschreibung von MEL und die Erstellung von Skripten

Complete Maya Programming – An Extensive Guide to MEL an C++ API David A. D. Gould

Beschreibung von MEL und die Erstellung von Skripten sowie das Einbinden in Maya

Maya Documentation

Alias Maya Dokumentation

<http://svn.sourceforge.net/viewvc/rawkee/>
Downloadseite der aktuellen RawKee Source Dateien

<http://www.robthebloke.org/index.html>
Tutorien und Codebeispiele zur Maya Programmierung

www.ewertb.com/maya/mel/index.html
Forum für Maya Programmierung

[Glossar]

<http://www.itwissen.info/>
Begriffsdefinitionen

<http://de.wikipedia.org/wiki/Hauptseite>
Begriffsdefinitionen

<http://web.inf.tu-dresden.de/~ag-vmw/>
Einführung in Maya sowie Tutorien

Ansprechpartner

Das vorliegende Produkt entstand im Rahmen Belegarbeit am Lehrstuhl für Computergrafik an der Technischen Universität Dresden im Sommersemester 2006. Dabei ging es um die Erstellung und Visualisierung von 3D-Präsentationen für den Einsatz im Unterricht im universitären Bereich.

Für Fragen und Anregungen wenden Sie sich bitte an Dipl.-Inf Sascha Seewald (sascha.seewald@inf.tu-dresden.de), Betreuer der Belegarbeit, oder Jan Wojdziak (s8085685@mail.inf.tu-dresden.de), Entwickler des Produktes