

# VRML - Authoring Tools

VRML-Seminar 1998/99  
Universität Mannheim - Lehrstuhl PI 4

von Nicolai Scheele

# **1. Überblick - Welche Typen von Authoring Tools gibt es?**

Seit einiger Zeit gibt es eine breite Palette verschiedener Authoring Tools, die dem Benutzer auf unterschiedliche Art und Weise beim Erstellen seiner VRML-Welten unterstützen. Grundsätzlich kann man dabei folgende Grundtypen unterscheiden:

- **Scene-Editoren:** Diese Editoren bilden den größten Anteil der Authoring Tools. Mit Hilfe verschiedener Aspekte und Bearbeitungstools lassen sich visuell dreidimensionale Szenen umsetzen, die dann im VRML-Dateiformat gespeichert werden können.
- **Script-Editoren:** Sie unterstützen lediglich die textbasierte Programmierung der VRML-Szenarien.
- **Construction Sets:** Dies sind spezielle Editoren, die eine Reihe vorkonstruierter Elemente aus einem bestimmten Bereich zur Verfügung stellen. Der Anwender kann beliebig viele dieser Elemente nach bestimmten Regeln zusammensetzen.
- **Sonstige Tools:** Neben diversen Editoren gibt es auch noch eine Reihe weiterer Hilfsmittel, die den Anwender beim Fehlersuchen, Optimieren und Konvertieren seiner Dateien helfen sollen.

Es gibt inzwischen eine fast unüberschaubare Menge an Tools und Editoren sowohl aus dem kommerziellen als auch aus dem Free- und Sharewarebereich. Selbstverständlich kommt es dadurch zu erheblichen Unterschieden in Funktionsumfang und Qualität bei den Programmen; in den folgenden Abschnitten werde ich deshalb nur einen Querschnitt der üblicheren Funktionen zu den einzelnen Authoring Tool Typen angeben.

## **2. Scene-Editoren**

Scene-Editoren erlauben dem Benutzer, mittels rein grafischer Tools komplexe dreidimensionale Szenen und Welten aufzubauen und detailliert auszustatten. Je nach Editor stehen dabei drei oder mehr verschiedene Ansichten zur Verfügung: von vorne, von der Seite und von oben; teilweise noch isometrisch und perspektivisch. Diese Ansichten bilden

einen virtuellen Raum, in dem einfache und komplexe Objekte erstellt, verschoben, kombiniert und auf beliebige Weise verändert werden können. All dies geschieht entweder per Drag&Drop innerhalb der grafischen Darstellung oder mittels Eingabe der genauen Daten, falls eine genaue Plazierung nötig ist.

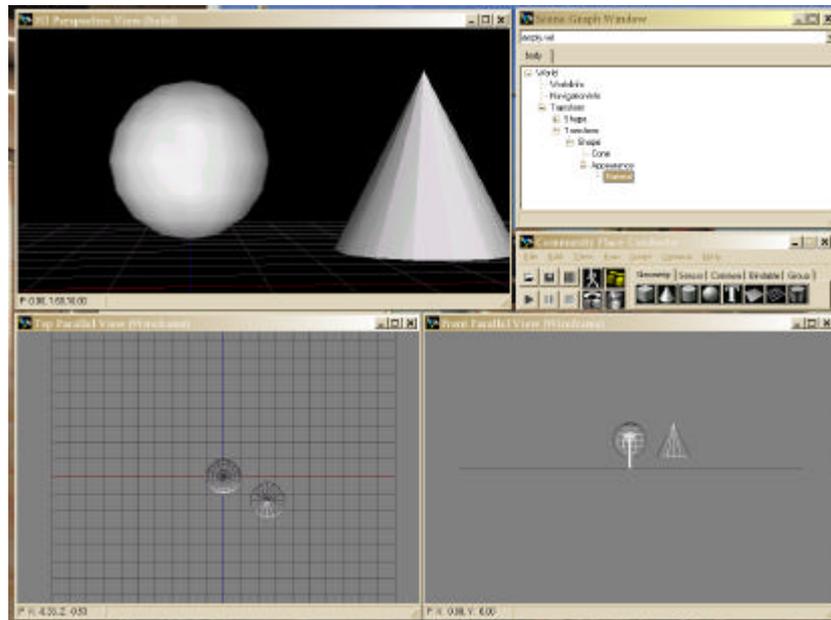


Abb. 1: Sony's Community Place Conductor

Wichtig ist auf jeden Fall, daß der Benutzer keinen Kontakt mit dem eigentlichen VRML-Dateiformat hat, sondern sich ganz auf die Gestaltung der Welt konzentrieren kann. Mehr oder weniger detaillierte VRML-Kenntnisse helfen zwar in vielen Fällen bei der Arbeit mit Scene-Editoren, doch sind sie nicht unbedingt nötig, um ansprechende virtuelle Welten zu erstellen.

## 2.1. Warum Scene-Editoren?

Der Hauptgrund für die Verwendung von Scene-Editoren liegt im VRML-Dateiformat selbst. Dies hat mehrere Ursachen:

1. VRML-Code wird durch die vielen Verschachtelungen sehr schnell unübersichtlich. Dies behindert natürlich die Erstellung komplexer Welten, vor allem aber wird die Fehlersuche und Wartung erheblich erschwert.
2. Primitives, die durch Angabe vieler Eckpunkte bestimmt werden, gleichen im VRML-Code langen Zahlenkolonnen, die weder vernünftig berechnet noch nachträglich modifiziert werden können.

3. Ähnliches gilt für Animationen, da hier für eine unbestimmte Anzahl von Keyframes viele Daten der betroffenen Nodes angegeben werden müssen (Position, Skalierung, Rotation...)

Des Weiteren soll auch Programmier-Laien die Möglichkeit gegeben werden, VRML-Szenen zu erstellen, ohne sich mühselig in das Dateiformat einarbeiten zu müssen, das, wie bereits erwähnt, ohnehin nur bedingt zur Programmierung geeignet ist.

## 2.2. Umgang mit Scene-Editoren

Anhand des Shareware-Editors "Spazz3D" wird im Folgenden grob der Arbeitsablauf in Scene-Editoren demonstriert. Startet man Spazz3D erscheint zunächst folgendes Bild:

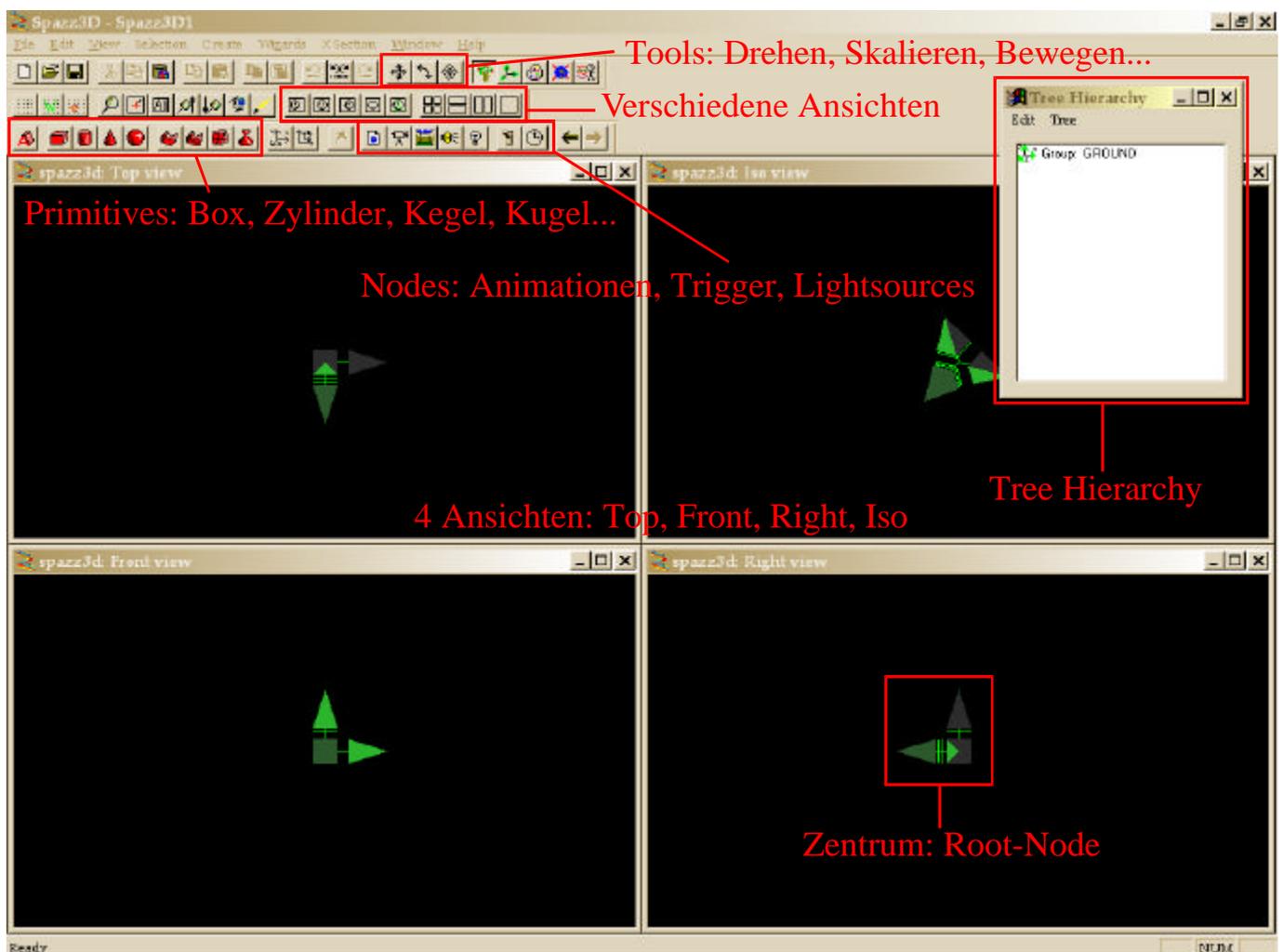


Abb. 2: Spazz3D nach dem Start

Nun kann man in der Toolbar die Schaltfläche eines Primitiv-objektes anwählen; fährt man mit dem Cursor dann über die noch schwarzen Ansichtsfenster, kann man den gewünschten Körper mit Hilfe eines Gitterrahmens in dem Szenario bewegen. Ein

einfacher Klick genügt um ihn fest zu plazieren - er nimmt nun eine solide Form an. Solange das Primitiv noch markiert ist (hellblaue Umrahmung), kann man es mit den Hilfsmitteln in der oberen Toolbarleiste weiter verschieben, rotieren oder skalieren. Ein Doppelklick auf das Objekt öffnet ein weiteres Fenster, in dem Koordinaten und Achsenrotationen genau angegeben werden können, aber auch Textur und Material bestimmt wird.

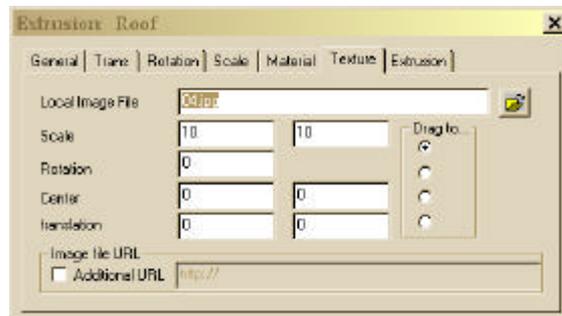


Abb. 3: Eigenschaftfenster für Primitives

Ebenso einfach lassen sich Groupnodes in die Szene einbinden und als Basis für alle folgenden Objekte definieren. Natürlich können auch nachträglich die Nodes per Drag&Drop im Tree Hierarchy-Fenster (Abb.2 rechts oben) innerhalb des Szenenbaums verschoben werden; relative Koordinaten werden bei Wechsel des zu Grunde liegenden Groupnodes automatisch berechnet.

Sind genügend Objekte vorhanden, können weitere wichtige Nodes eingefügt werden: Anchors, Lightsourcen, Viewpoints, Backdrops, sowie Proximity- und Touch-sensors. Ebenso einfach wie die Primitives lassen auch sie sich in der Szene und im Szenenbaum plazieren und per Doppelklick genau definieren.



Abb. 4: Routendefinition bei einem Touchsensor

Trigger können alles auslösen, was per VRML (ohne externe Java/-script Programme) möglich ist: Licht ein- und ausschalten, Audio-effekte abspielen, Animationen starten...

Für die Animationen ist - wie in den meisten Scene-Editoren - ein Key-frame-Editor eingebaut, der per Doppelklick auf ein Animationsobjekt gestartet wird. Hier muß zunächst angegeben werden, wieviele Keyframes benötigt werden und an welchen Zeitpunkten sie erreicht werden sollen. Des weiteren wird die Laufzeit der Animation benötigt, ob sie einmal oder ständig laufen soll und welche Nodes betroffen sind. Nun müssen für jeden Keyframe die markierten Objekte in die gewünschte Position gebracht und gegebenenfalls Rotation und Skalierung angepaßt werden. Späteres Nachbearbeiten ist jederzeit möglich, auch kann man die Animation in den Ansichtsfenstern provisorisch ablaufen lassen.

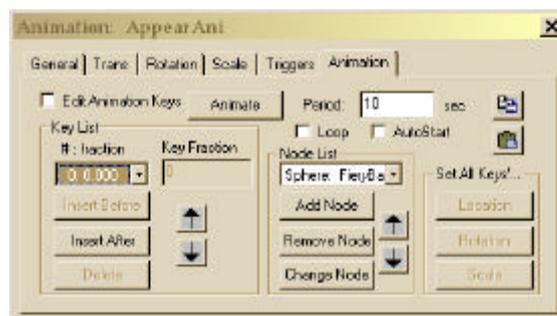


Abb. 5: Key-frame-Editor

Das bisherige Ergebnis kann man sich entweder mit einem eingebauten, recht primitiven Browser anzeigen lassen oder gleich in einem "richtigen" Browser wie dem CosmoPlayer. Ist man mit dem Ergebnis zufrieden, kann man es nach VRML exportieren. Dabei stehen einige Optionen zur Verfügung, die hauptsächlich das Aussehen und die Lesbarkeit des erzeugten Codes modifizieren.

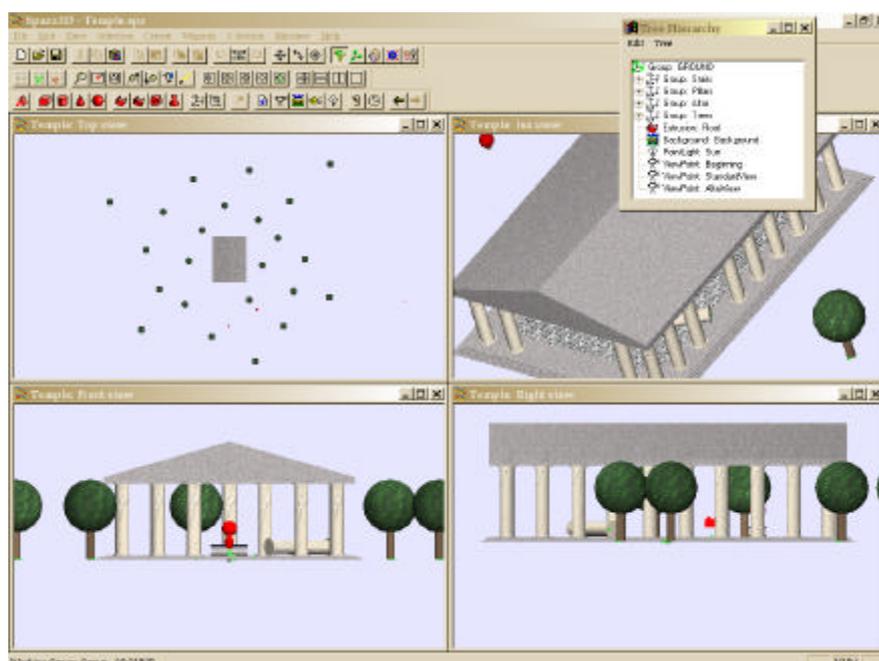


Abb. 6: Eine fertige Szene in Spaz3D

### 2.3. Funktionalität der Scene-Editoren

Jeder Editor sollte - neben einer brauchbaren grafischen Darstellung - eine gewisse Grundfunktionalität besitzen, die zur stressfreien Weltenkreation zwingend notwendig ist. Zu diesen grundsätzlichen Funktionen gehören:

1. Platzierung von Primitives. Jeder Editor muß die Basisprimitives unterstützen: Kubus, Kugel, Zylinder, Kegel, Text. Weiterhin sollten brauchbare Gestaltungsmöglichkeiten für Extrusions und Elevation Grids existieren, wie z.B. Rotationskörper, Prismen, extrudierte Schrift und unebene Flächen.
2. Die erstellten Primitives müssen innerhalb des Raumes verschoben, skaliert und rotiert werden können.
3. Primitives müssen Texturen zugeordnet werden können, die, im Idealfall, im Editorfenster angezeigt oder zumindest angedeutet werden. Gute Editoren bieten neben einer größeren Anzahl vordefinierter Farben auch einen Katalog mit Bitmap-texturen.
4. Trigger, Kamerapositionen, Lichtquellen, Anchors, etc. müssen genauso einfach in der Szene untergebracht werden können, wie Primitives (Drag&Drop).
5. Einfache Routes (z.B. Trigger startet bei Mausklick eine Animation) sollten mit wenigen Mausklicks erstellt werden können.
6. Die meisten Editoren stellen auch einen Animationseditor zur Verfügung, mit dem einfache Animationen erstellt und betrachtet werden können. Es wird vom Benutzer lediglich angegeben, wie ein oder mehrere Objekte zu verschiedenen Zeiten innerhalb der Animation platziert werden und aussehen sollen (Keyframes); den Rest berechnet das Programm bzw. der VRML-Browser.
7. Schließlich muß dem Benutzer die Möglichkeit gegeben werden, seine Objekte zu gruppieren und manuell im Szenenbaum zu platzieren.

Diese Funktionen gewährleisten prinzipiell die Erstellung beliebig komplexer Welten. Es gibt jedoch noch eine Reihe vieler zusätzlicher Tools, die dem Benutzer des jeweiligen Editors wesentlich mehr Komfort oder zusätzliche Gestaltungsmöglichkeiten bieten.

So bieten einige Editoren die komfortable Einbindung von Java/JavaScript-Nodes, meist verbunden mit eingebauten Java-Editoren, die Templates für verschiedene Anwendungsgebiete in diesen Sprachen bieten (z.B. Modifikation von Node A in

Abhängigkeit von Node B). Die so erstellten Programme können dann einfach eingebaut werden; Routes erstellt der Editor automatisch.

Auch die automatisierte Erstellung komplexer Animationen (kreisförmig, mit Beschleunigung...) wird von wenigen (teureren) Editoren ermöglicht. Die nötigen Java-classes werden automatisch erstellt.

Zuletzt sollte noch erwähnt werden, daß insbesondere der Umgang mit Extrusions und Elevation Grids ein maßgeblicher Faktor für die Qualität und Brauchbarkeit eines Editors ist: Je mehr verschiedene Tools zur Erstellung dieser Primitives vorhanden und je einfacher sie zu bedienen sind, um so leichter ist es für den Benutzer, realistischere Welten zu erstellen. Teure Programme wie Cosmo Worlds verfügen darüber hinaus über eine umfangreiche Bibliothek bereits vorgefertigter Polyeder.

## **2.4. Was fehlt noch?**

Obwohl die derzeit angebotenen Editoren durchaus ausgereift und benutzerfreundlich sind, gibt es dennoch einige Features, die man zum Teil schmerzlich vermißt - selbst bei den sehr teuren Programmpaketen. Dazu gehören:

- ... die Automatisierung geometrischer Berechnungen. Zum Beispiel unterstützt kaum ein Editor den Benutzer bei der Plazierung eines Zylinders, dessen Endpunkte angegeben werden sollen; langwieriges Drehen und Skalieren oder komplizierte externe Berechnungen sind die Folge.
- ...wirklich leistungsfähige Modelling-Tools komplexer Figuren (Drehkörper, Prismen...)
- ...die Emulation höherer geometrischer Körper durch VRML-Primitives. Vorstellbar sind z.B. Torus, Pyramiden mit verschiedenen Grundfiguren oder die aus dem Raytracing bekannten CSG-Objekte (Schnittmenge, Kombination, Differenzen)

Ein weiteres, sehr großes Problem entsteht durch die massive Überladung mit Funktionen und Details, die bei einigen Programmen insbesondere im gehobenen Preisbereich schnell abschreckend auf den noch ungeübten Benutzer wirken kann. Nur die wenigsten Editoren können gleich nach dem Installieren bedient werden - zum Teil recht lange

Einarbeitungszeiten sind nötig, bevor brauchbare Ergebnisse erzielt werden können. Verbesserte Online-Hilfen, tiefgreifende Tutorials und unterschiedliche Menüstrukturen für Anfänger, Fortgeschrittene und Experten könnten helfen, diese Mißstände zu beseitigen.

### **3. Script-Editoren**

Script-Editoren müssen nicht zwangsläufig VRML-spezifisch sein; einige bekanntere Programmier-Editoren unterstützen inzwischen die VRML-Programmierung auf Textbasis durch Features wie Syntax-Highlighting, Auto-Indentation und kontextsensitive Hilfe. Es gibt aber auch einige speziell auf das VRML-Format ausgerichtete Editoren (z.B. Sitepad), die noch zusätzliche Hilfen wie Errorchecking oder Standard-Templates zum Einfügen bieten.

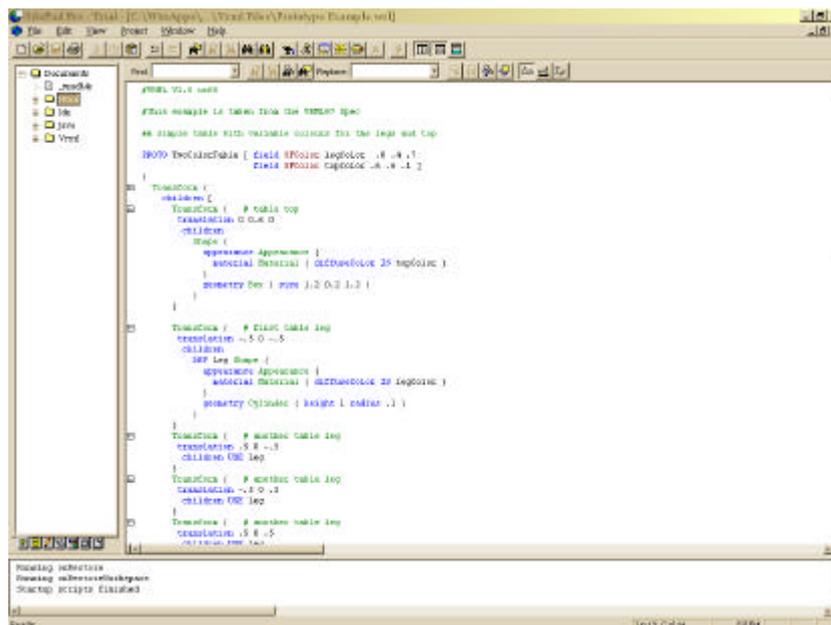


Abb. 7: SitePad Pro

Grundsätzlich sind Script-Editoren hauptsächlich zur Nachbearbeitung der von den Scene-Editoren erstellten VRML-Dateien sinnvoll. So können Werte bereinigt, Routen erstellt, Java-Scripte eingebunden oder sonstige Modifikationen am Source vorgenommen werden.

## **4. Construction Sets**

Ähnlich wie bei den Scene-Editoren ist der Benutzer in der Lage, mittels grafischer Tools virtuelle Welten zu erzeugen. Anders jedoch ist die Art und Weise, wie dies geschieht. Während Scene-Editoren dem Anwender mehr oder minder einfach die Einbindung aller verfügbaren VRML-Nodes gestattet und ihm die Gestaltung der Welt weitgehend offen läßt, konzentrieren sich Construction Sets auf einen bestimmten Bereich. Innerhalb dieses Bereiches werden eine größere Anzahl von Elementen vorgegeben, die dann - bestimmten Regeln folgend - nach dem Baukastenprinzip zusammengesetzt werden können. Dies ermöglicht die extrem einfache - vielleicht sogar automatische - Erstellung äußerst komplexer Szenarien.

Welten, die mit Construction Sets zusammengestellt werden, sehen prinzipiell sehr ähnlich aus; sonderlich viel Platz für individuelle Gestaltung wird nicht gestattet. Dennoch hat diese Art von Editoren sinnvolle Anwendungsmöglichkeiten, z.B. die Darstellung technischer Vorgänge (Produktionsabläufe in der Fabrik), Architektur ("Begehen" eines Hauses, bevor es gebaut wird), Innenausstattung, Landschaftsplanung u.s.w. Hier steht hauptsächlich die Visualisierung im Vordergrund, um Entscheidungen oder Planung zu vereinfachen.

Eine weitere Anwendungsmöglichkeit hat sich in letzter Zeit erst eröffnet: Immer mehr sog. "virtual Chat Rooms" werden angeboten, in denen sich eingeloggte User als "Avatare" in einer vordefinierten "Arena" bewegen und mit anderen reden können. Für diese Arenas gibt es bereits eine Vielzahl von Construction Kits, die verschiedene Häuser, Autos, Brunnen und andere Objekte zur Ausstattung anbieten. Die Erstellung der Avatar Arenas geschieht so in wenigen Minuten per Drag&Drop.

## **5. Sonstige Tools**

Neben den bisher besprochenen, hauptsächlich zur Erstellung und Anpassung der VRML-Welten geeigneten Authoring Tools gibt es noch einige andere Utilities, die den Benutzer bei bestimmten Aufgaben, die speziell den VRML-Code betreffen, unterstützen.

Ein besonders herausragendes Programm in dieser Kategorie ist das Shareware-Tool "Chisel" (Nachfolger von "Vorlon"). Seine Funktionalität teilt sich in mehrere Schritte auf, die der Benutzer Schritt für Schritt oder auch selektiv einzeln ausführen kann. Zunächst wird der VRML-Code auf Fehler und Unstimmigkeiten geprüft, dabei wird auch auf mögliche Schwierigkeiten mit einem der wichtigeren Browsern hingewiesen. Neben der rein syntaktischen Prüfung erkennt Chisel viele logische Fehler und löst diese z.T. automatisch. Im Anschluß daran kann der Code optimiert werden: Hier sucht Chisel nach überflüssigen (sprich: nicht sichtbaren oder ungenutzten) Nodes und entfernt sie; ebenso kann die Anzahl der Eckpunkte bei Extrusions nach bestimmten Schemata reduziert werden. Darüber hinaus wird der Code in eine bestimmte Struktur gebracht, die optimale Aufbau- und Displaygeschwindigkeit im Browser verspricht. Schließlich wird der transformierte Code in gut lesbarem Format gespeichert.

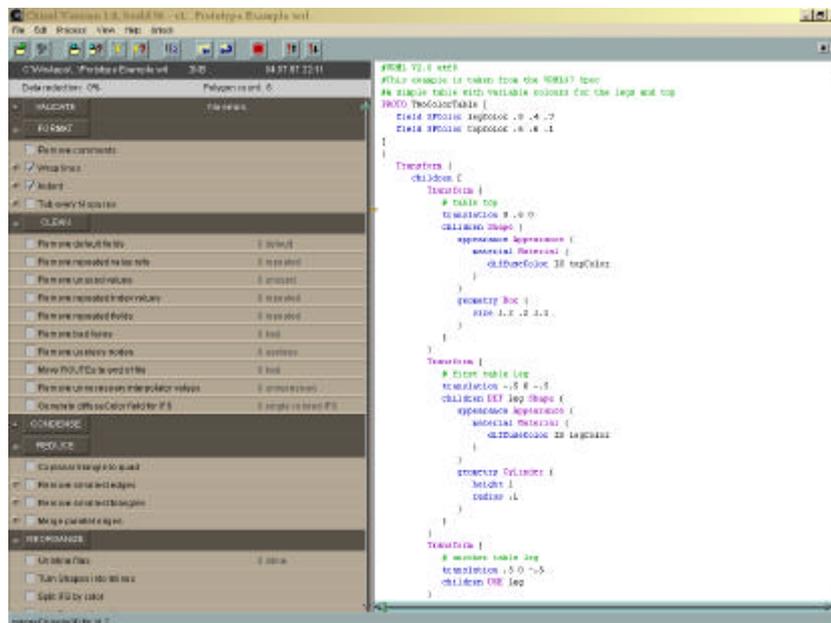


Abb. 8: Chisel v1.0

Weitere, zu bestimmten Anlässen sehr nützliche Tools sind Crosscompiler, die Dateien anderer 3D/2D-Formate analysieren und in VRML-Code umwandeln. So können bereits angefertigte Szenen aus anderen Programmen wie 3D Studio, Simply 3D oder auch POV-Ray für VRML vorbereitet und dann mit wenig Nachbereitung fertiggestellt werden - zumindest ist dies die Intention dieser Programme; die Ergebnisse die geliefert wurden, waren zumeist völlig unbrauchbar. Die Umwandlung von 2D-Formaten wie WMF oder TrueType-Fonts war schon wesentlich produktiver, da hier durchaus sinnvolle Objekte hergestellt werden können, die sich sehr einfach in neue Szenen einbauen lassen.

## **6. Zusammenfassung**

Es gibt bereits eine große Auswahl sehr nützlicher Authoring Tools, die den Benutzer nicht nur bei der Erstellung seiner Welten, sondern auch beim Vor- und Nachbereiten maßgeblich unterstützen. Da viele Tools aus dem Free- oder Sharewarebereich stammen und für die meisten kommerziellen Programme kostenlose Evaluation-versionen verfügbar sind, kann sich der Anwender je nach Kenntnisstand, gewünschten Funktionsumfang und Budget ein für ihn passenden Editor oder unterstützendes Hilfsmittel aussuchen. So steht VRML quasi für jeden als Gestaltungsmöglichkeit seiner Webseiten offen.

Andererseits sind VRML-Authoring Tools noch ein recht neuer Sektor in der Softwarebranche und ständig kommen neue Produkte oder neue Versionen auf den Markt, so daß bereits in naher Zukunft mit wesentlich mächtigeren und einfacher zu bedienenden Tools zu rechnen ist. Zum einen aus diesem Grund, aber auch zum Einarbeiten ist eines der vielen Shareware-Programme vielleicht die bessere Wahl.