

## VRML-Kurs

### Teil 1

#### *VRML:*

Virtual Reality Markup Language

→ Virtual Reality Modelling Language

Beschreibungssprache für VR-Szenen

(kein Softwaresystem)

Umsetzung durch *VRML-Browser*

VRML-Szenen in ASCII-Dateien abgelegt

Standardisierung: Dieselbe Szene durch verschiedene Browser darstellbar

#### VRML-Umfang:

3D-Grafik

+ Dynamik (Animation)

+ Interaktion

+ Klang

+ Internetfähigkeit (VRML-Browser als Plugin von Webbrowsern)

Versionen: VRML 1.0, 2.0, 97

#### Features:

Geometriebeschreibungen	Grundprimitive (Quader, Zylinder, Kegel, Kugel) polygonale Objekte elevation grids (Terrain-Gitter) extrudierte Objekte
Materialbeschreibungen	Farbe Schattierungen
Beleuchtungsbeschreibungen	

Texturbeschreibungen	
Textbeschreibungen	
Transformationen	Translationen Rotationen Skalierungen allgemeine Transformationen
Animationen	Zeit Interpolation
Hintergrund und Nebel	
Interaktionsbeschreibungen	
Multimediabeschreibungen	Video 3D-Klang
Strukturbeschreibungen	Gruppierung Prototypen
Detaillierungsbeschreibungen	LOD = level of detail
Kamerabeschreibungen	Viewpoints
Hypermediabeschreibungen	Anchors
Beschreibung von Programmierlogik	Java Javascript VRMLscript

- VRML kann mit immersiven Technologien umgesetzt werden
- Einbindung von Programmen beliebiger Komplexität
- Programme können in VRML-Szenen eingreifen  
(Anwendungen z.B.: Mehrbenutzersysteme, Datenbank-Anbindung, Simulationen)

## Geschichte

- Erste Ideen zu virtuellen Welten im WWW von Mark Pesce und Tony Parisi, 1994
- im Rahmen der ersten internationalen Konferenz über das WWW (Mai 1994 bei CERN in Genf) wird eine Sitzung zu einer geplanten "Virtual Reality Markup Language" abgehalten (wichtige Rolle: Tim Berners-Lee, der Entwickler von HTML); das Akronym "VRML" wird geprägt

- Oktober 1994: VRML 1.0 wird von Tony Parisi und Gavin Bell präsentiert, basiert nach Entscheidung durch Internet-Abstimmung (Mailingliste) auf Open Inventor von SGI. Beginn der Unterstützung durch SGI, Netscape und Microsoft
- VRML Architecture Group (VAG; 8 technische Experten) wird im Anschluss an die SIGGRAPH'95 gegründet (August 1995)
- Anfang 1996 Aufruf zu Vorschlägen für VRML 2.0 durch die VAG
- August 1996: VRML 2.0-Spezifikation wird auf der SIGGRAPH'96 vorgestellt, nach offener Internet-Abstimmung auf Moving Worlds von SGI basierend; Gründung des VRML-Konsortiums (Vertreter von Firmen, Forschungseinrichtungen und Universitäten)
- Anfang 1997: Beginn der ISO-Standardisierung mit der Erarbeitung von VRML 97
- Ende 1997: VRML 97 wird standardisiert als ISO/IEC DIS 14772-1
- Zukunft: x3d, ISO in 2002 angestrebt

### VRML-Browser

Cosmo Player      Download: <http://www.cai.com/cosmo/>  
 World View        (dito)  
 Open Worlds  
 ....

Seite mit Links zu verschiedenen Download-Sites:  
<http://www.web3d.org/vrml/vrml.htm>

Cosmo Player 2.1 für Win95/NT: Plugin zu Netscape 3.01 und höher, MS Internet Explorer 4.x und höher.

Jeder Browser stellt Navigationshilfen für den 3D-Raum zur Verfügung  
 Anordnung und Bezeichnungsweisen browserspezifisch, aber ähnliche Grundfunktionen.

Hier für Cosmo Player:

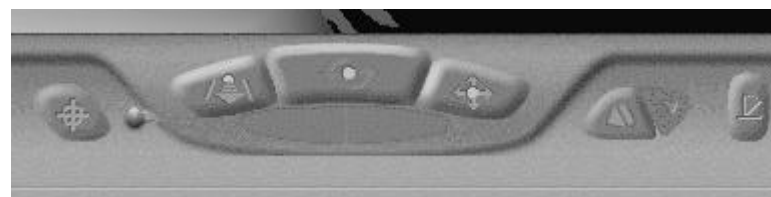
## 6 Haupt-Navigationsmodi

Gehen	Go	Bewegungen in der horizontalen Ebene
Neigen	Tilt	Änderung des Blickwinkels
Gleiten	Slide	Bewegungen in der vertikalen Ebene
Rotieren	Rotate	lässt die Szene rotieren
Schwenken	Pan	Eigenbewegung um eine Szene, die im Blickfeld bleibt
Zoomen	Zoom	Änderung des Abstands von der Szene

Die eigentliche Navigation erfolgt mit der Maus. Die Form des Mausursors zeigt an, welcher Navigationsmodus aktiv ist. Zwischen der oberen und der unteren Dreiergruppe wird mit einem virtuellen Hebel umgeschaltet ("Steuerelemente ändern"; auch durch Hotkeys (Tastatur) möglich).



*Schaltfläche mit Funktionen der ersten Gruppe*

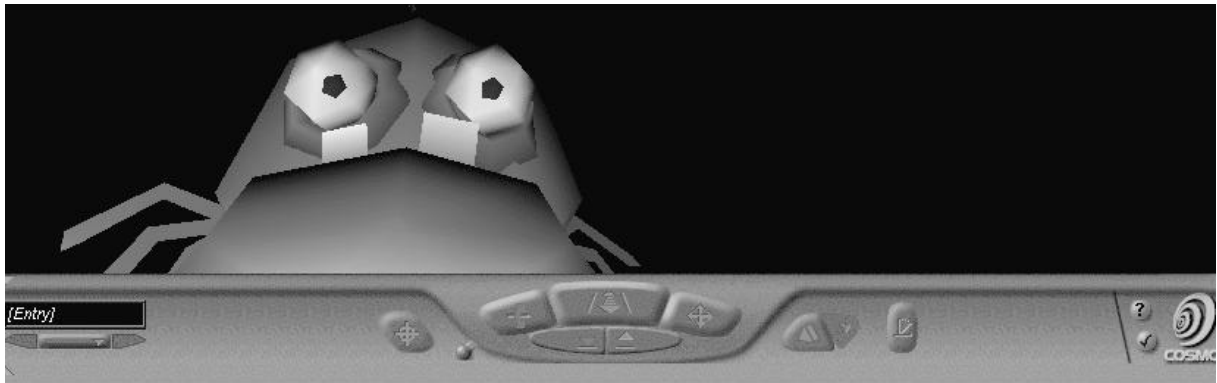


*Schaltfläche mit Funktionen der zweiten Gruppe*

Zusätzliche Sonderfunktionen:

Gravitation	Gravity	in hügeliger Landschaft am Boden bleiben (man folgt den Höhen und Tiefen der Oberfläche)
Treiben	Float	erlaubt, zu fliegen

Suchen	Seek	durch Anklicken eines Objekts bewegt man sich an dieses heran
Bewegung rückgängig machen	Undo move	
Bewegung wiederherstellen	Redo move	
Richten	Straighten	man wird in eine Orientierung senkrecht zur Horizontalebene gebracht
Viewpoint		man kann Blickpunkte speichern und mit Namen versehen
Preferences		Aktivierung des Voreinstellungs-Menüs
COSMO		Hyperlink zur Webseite von Cosmo
Help		Online-Hilfe (HTML)
Warnleuchten		werden aktiv bei Fehlern, <b><i>bei Anklicken erscheint die Fehlermeldung</i></b>



## VRML-Dateien

Endung .wrl ("world")

ASCII-Datei (genauer: ab V. 2.0 UTF-8 Zeichensatz gem. ISO 10646-1:1993)

Aufbau einer VRML-Datei:

- Header (Version und Zeichensatzangabe, obligatorisch)
- Zeilenkommentare
- VRML-*Knoten*
- innerhalb der Knoten-Spezifikationen: *Felder* (= festgelegte Attributierungen von Knoten, denen Werte zugewiesen werden)
- PROTO-Statements
- ROUTE-Statements

Zuordnung von Feldangaben zu Knoten mit geschweiften Klammern { ... }

Gruppierung von Knoten durch Gruppen-Knoten, "Kinder" in eckigen Klammern [ ... ]

Beispiel einer VRML 1.0-Datei:

```
#VRML V1.0 ascii
# created by W. K. 2001
# Huette

# zeichne den Rumpf
Cylinder
{
  height 2
  radius 2
}
# Verschiebung des Koordinatenursprungs
Translation
{
  translation 0 2 0
}
# zeichne das Dach
Cone
{
  bottomRadius 2.5
}
```

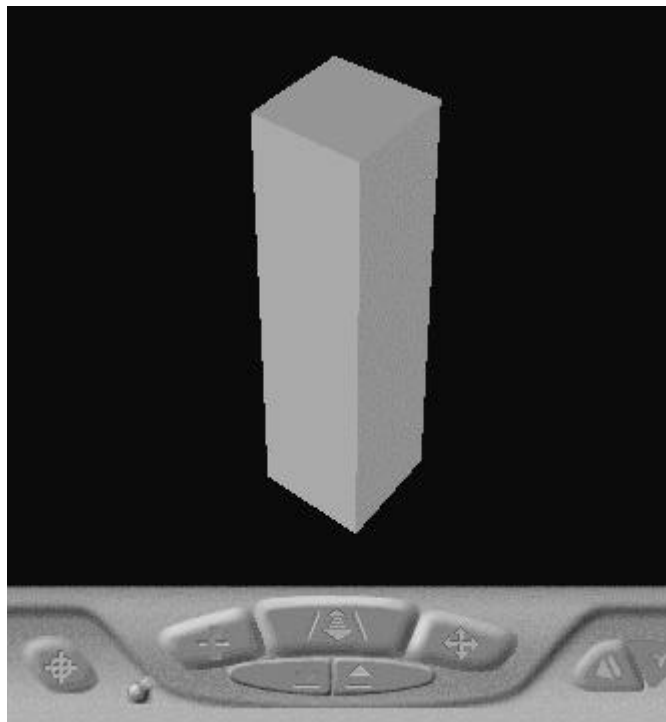
Ergebnis:



Beispiel einer VRML 2.0 - Datei:

```
#VRML V2.0 utf8
Shape
{
  geometry Box
  {
    size 1 4 1
  }
  appearance Appearance
  {
    material Material
    {
      diffuseColor 0.5 0.5 0.5
    }
  }
}
```

Ergebnis:



Header für VRML 97:  
#VRML V97 utf8

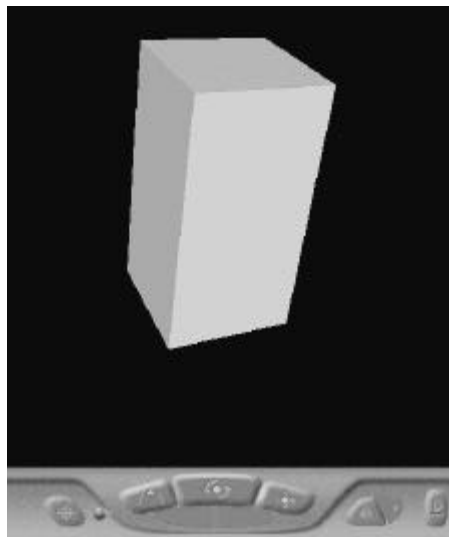


Felder, für die keine Spezifikation angegeben wird, werden automatisch mit Default-Werten besetzt. Z.B. ist `Cylinder { }` ein Zylinder mit Höhe 1 und Radius 1.

Die Primitiv-Objekte (Würfel, Zylinder, Kegel, Kugel) werden standardmäßig um den Nullpunkt zentriert.

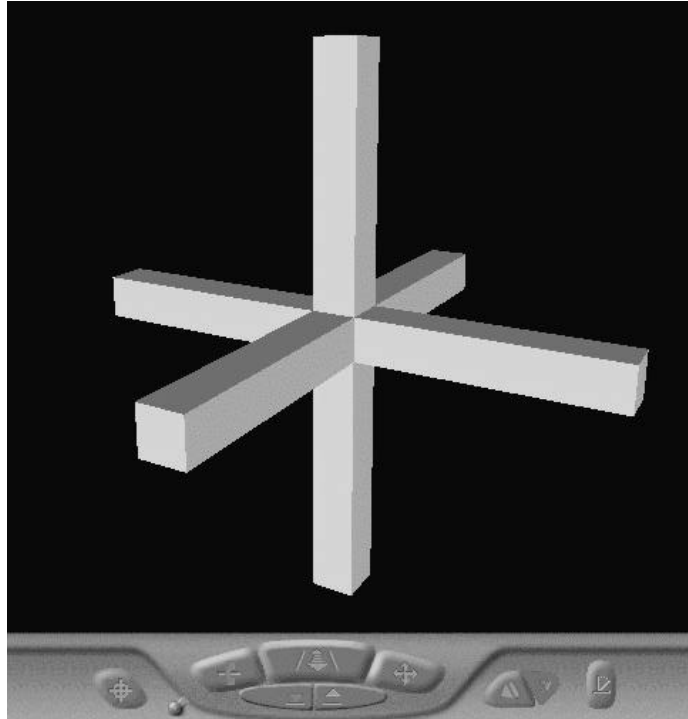
```
#VRML V1.0 ascii
Cube
  {
    height 4
  }
```

⇒ Säule, die sich 2 Einheiten nach oben und unten und je 0,5 Einheiten nach rechts, links, vorn und hinten erstreckt



Überlagerung von 3 verschieden dimensionierten Quadern zu einem "Koordinatenkreuz":

```
#VRML V1.0 ascii
Cube
  { width 25 }
Cube
  { height 25 }
Cube
  { depth 25 }
```



Das `parts`-Feld dient dem Selektieren von Teilmengen der Primitivobjekte:

```
#VRML V1.0 ascii
Cylinder
{
  radius 3
  height 0.7
  parts (SIDES)
}
```



```

#VRML V1.0 ascii
Cylinder
  {
    radius 3
    height 0.7
    parts (TOP)
  }

```



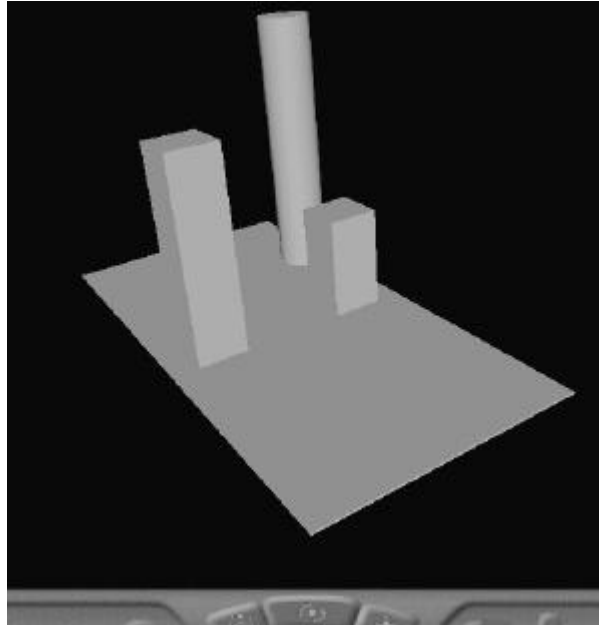
Transformationsknoten transformieren das Koordinatensystem für folgende Konstruktionen (vgl. PostScript):

```

#VRML V1.0 ascii
Cube
  { height 0.1 width 10 depth 20 }
Translation
  { translation -3 4 0 }
Cube
  { height 8 }
Translation
  { translation 6 -2 0 }
Cube
  { height 4 }
Translation
  { translation 0 4 -3 }
Cylinder
  { height 12 }

```

Ergebnis:



Felder der bisher eingeführten geometrischen Primitiv-Knoten (VRML 1.0):

**Cube**

```
{  
  width  
  height  
  depth  
}
```

**Sphere**

```
{  
  radius  
}
```

**Cylinder**

```
{  
  height  
  radius  
  parts # default: ALL, alternativ: SIDES, TOP, BOTTOM  
        # auch möglich: ( TOP | BOTTOM )  
}
```

```

Cone
{
  height
  bottomRadius
  parts # ALL, SIDES, BOTTOM
}

```

## Textausgabe in VRML

```

#VRML V1.0 ascii
AsciiText
{
  string "Text"
}

```

Ergebnis:



Felder des **AsciiText**-Knotens:

```

AsciiText
{
  string # mehrere Zeilen: [ "Zeile 1", "Zeile 2", "Zeile 3" ]
  justification # default: LEFT, alt.: RIGHT, CENTER
  width # 0 bedeutet: default-Weite benutzen
  spacing # (default: 1) für Zeilenabstände
}

```

## FontStyle-Knoten:

```
FontStyle
{
  family      # SERIF | SANS | TYPEWRITER
  style       # NONE | BOLD | ITALIC
  size        # Höhe in Längeneinheiten von VRML,
              # default: 10, darf auch negativ sein
}
```