



Einführung in VRML

- VRML heisst Virtual Reality Modeling Language, ein Teil von “Web3D”/”X3D”.
- VRML ist einfach ein 3D Datenaustausch-Format: Es besitzt Sprachelemente für die heute in 3D-Applikationen üblichen Elemente wie Objekte, Licht, Texturen...
- VRML ist ähnlich zu HTML: Dadurch kann man es einfach im Internet als Dokument transportieren und in einem geeigneten “Browser” anzeigen lassen.



Mehr zu VRML

- VRML liefert die Technologie, drei Dimensionen, zwei Dimensionen, Text und Multimedia in einem kohärenten Modell unterzubringen.
- Dadurch entstehen in Verbindung mit Skriptsprachen und dem Internet völlig neue Anwendungsmöglichkeiten.
- VRML ist die Basis des “Cyberspace” und von virtuellen on-line Communities.
- VRML kann per se keine echten Multiuser-Simulationen oder -Welten mit Interaktion darstellen (wie z.B. Rollenspiele etc.).



Noch mehr VRML

- Auf dem Client wird mit einem Plug-In (z.B. Blaxxun oder cortona) das VRML Format interpretiert und das “Rendering” besorgt, d.h. die 3D-Darstellung.
- VRML basiert auf dem von SGI erfundenen Dateiformat “Open Inventor”, aber besitzt leider kein API (Programmier-Schnittstelle).
- VRML ist Basis von MPEG4 und Java3D.



Vor-Geschichte von VRML

- 1992: SGI veröffentlicht das Iris Inventor 3D Toolkit, geschrieben in C++. Hatte bereits viele Eigenschaften des heutigen VRML.
- 1994: Mark Pesce und Tony Parisi entwickeln einen frühen Prototypen eines 3D-Browsers für das WWW. Sie riefen Interessenten auf, Vorschläge für eine formale Spezifikation von 3D im WWW einzureichen.



Versionen von VRML

- Die erste Version (VRML 1.0) lässt einige Schlüssel-Eigenschaften vermissen:
 - Keine Animation,
 - Keine Interaktion,
 - Kein “Behavior”, also Verhalten von Objekten bei Aktionen.
- 1996: In Kyoto beschliesst die ISO, VRML 2.0 als Standard zu veröffentlichen.



VRML und andere Formate

- VRML Files können Referenzen zu Files in vielen anderen Formaten enthalten:
 - JPEG, PNG, GIF,
 - MPEG,
 - WAV, MIDI,
 - Java, JavaScript.
- Dies sind alles unabhängige Standards, die im WWW weit verbreitet sind.



VRML + HTML + Scripting/Java

- VRML in HTML einbetten:
Dies geht mit Hilfe des `<EMBED>` oder `<OBJECT>` HTML Tags. Nicht jeder Browser muss das unterstützen.
- Script-Code in VRML eingebettet:
Standard-Feature von VRML 2.0; benutzt einen so genannten “Script Node”.
- Java Applet kommuniziert mit VRML
Browser: Erweiterung von VRML 2.0, genannt “EAI” oder “External Authoring Interface”.



VRML/X3D + JAVA

- Java Klassen erzeugen und manipulieren VRML nodes: Entwicklungs-Libraries.
- In Verbindung mit X3D kann man Xj3D benutzen, um X3D zu erzeugen bzw. per SAI zu manipulieren.
- SAI: Scene Access Interface (=EAI)



Überblick über VRML (1)

- “Scene Graph” Struktur:
Eine VRML Datei ist hierarchisch aufgebaut. Objekte werden “Nodes” (Knoten) genannt.
- Nodes:
Es gibt ca. 54 verschiedene **Typen von Nodes**, z.B. für geometrische Grundformen, Erscheinungs-Eigenschaften, Klänge usw. Nodes speichern ihre Daten in Felder, von denen es 20 verschiedene Typen gibt.



Überblick über VRML (2)

- **Event Architektur:**
VRML 2.0 kennt einen “Event” oder Nachrichten-Mechanismus, über den Nodes in einem Scene Graphen miteinander kommunizieren können. Jeder Knotentyp definiert individuelle Events und Event-Typen, die Knoten diese Typs senden und empfangen können. Route-Statements definieren die Verbindungen zwischen Event-Quellen und -Senken.
- **Sensoren:**
Grund-Elemente für die User-Interaktion und Animation. Animationen werden über “TimeSensors” gesteuert, die Events generieren und damit die Animation steuern.



Überblick über VRML (3)

- **Script Nodes:**
Können zwischen Event-Quellen und -Senken geschaltet werden, um das Verhalten z.B. einer Animation zu beeinflussen.
- **Interpolator Nodes:**
Werden benutzt, um (getriggert durch Events eines TimeNodes) Objekte auf bestimmten Bahnen zu bewegen.
- **Inline Node / EXTERNPROTO:**
Einbindung anderer VRML Welten (irgendwo aus dem WWW) als “Subroutine”.



Los gehts mit VRML

- Buch: VRML 97 bei Addison Wesley (1998)
- Tutorial im Web:
<http://web3d.vapourtech.com/tutorials/vrml97/index.html>
- File-Struktur eines .wrl Files:
 - Header: `#VRML V2.0 utf8`
 - Scene Graph: Besteht aus Nodes, die hierarchisch angeordnet die Geometrie beschreiben, plus Nodes für Event-Erzeugung und Routing.
 - Prototypes: Dadurch kann man Nodes-Typen erweitern.
 - Event Routing.



Beispiel-VRML-File

```
#VRML V2.0 utf8

Transform {
  translation 2 0 3
  children [
    Shape {
      geometry Sphere { radius 1.5 }
    } # Ende von Shape
  ] # Ende von children
} # Ende von Transform
```



Erklärung zum Beispiel

- # kennzeichnet Kommentare.
- Transform { } ist ein Gruppenknoten und schliesst einen Scene Graphen ein. Hinter der schliessenden Klammer kann man zu besserer Lesbarkeit auf das zugehörige Schlüsselwort hinweisen.
- Translation positioniert den Gruppenknoten im Raum. Ausprobieren!
- Children[] ist ein Feld, das die Kindknoten des Gruppenknotens sammelt, hier nur ein einziges: Eine Kugel mit Radius 1.5



Absolute und relative Position

- Wenn man **mehrere “transform{” Knoten nacheinander** aufführt, die jeweils ein anderes “translation” Statement enthalten, wird jeder dieser Knoten absolut positioniert.
- Man kann auch mehrere solche **Knoten ineinander verschachteln**, indem man nach dem “shape{” Statement den nächsten “transform{” Knoten einfügt.
- Merke: Das ist nicht dasselbe, der **Kindknoten erbt die Position des Elternknotens**, und “translation” ist relativ dazu!

Beispiel relative Translation

```
#VRML V2.0 utf8

Transform { translation 2 0 3

  children [

    Shape { geometry Sphere { radius 1.5 } } # Shape

    Transform {
      translation 1 0 1
      children [
        Shape {
          geometry Cone { bottomRadius 1 height 2}
        } # Shape Cone
      ] # children Nr. 2
    } # Transform Nr. 2

  ] # children
} # Transform
```

Drehen von Objekten

- Zum Drehen benutzt man das Feld “rotation” des Knotens “transform{”.
- rotation nimmt Parameter vom Typ SFRotation: `yaw pitch roll value`
- Das heisst: Drehung um die X-, Y- und Z-Achse. Und der Drehwinkel im Bogenmass.
- Beispiel: `1 0 0 3.1415`
bedeutet Drehung um X-Achse, etwa 180°

Farbe ins Spiel bringen

- Im Kindknoten Shape ist ein Feld “appearance” zu setzen.
- Ein weiterer Knoten “material” kennt Farben:

```
Shape {  
    appearance Appearance {  
        material Material  
            { diffuseColor 1 0 0 } # Rot  
    }  
    geometry Sphere {}  
}
```



Syntax in VRML-Files

- VRML unterscheidet Groß- und Kleinschreibung: **Sphere != sphere**
- Kommentare beginnen mit #
Alles **hinter #** in der Zeile ist Kommentar
- Whitespace sind *Space, Tab, Komma, Linefeed, Carriage Return*: Sie werden ausserhalb von Strings **ignoriert**.
- Namen dürfen keine Sonderzeichen oder Whitespace enthalten.



Vergleich mit X3D

- Die Syntax ist vereinfacht und basiert auf XML (X3D ist eine XML Anwendung)
- Es gibt die Möglichkeit, Applikations-Daten in der X3D Welt unterzubringen, die der Viewer nicht kennt und ignoriert
- X3D und Java sind enger miteinander verzahnt als VRML und Java (Xj3D, SAI)
- X3D kann komprimiert und verschlüsselt werden, die Viewer sollten das können



Vor-Übungen zur Belegarbeit

- Erstellen Sie jetzt selbst einfache Welten, die aus den Grundgeometrien “Sphere”, “Cone”, “Box” und “Cylinder” bestehen.
- Experimentieren Sie mit den “transform” und “rotate” Feldern.
- Beobachten Sie, was bei absoluter und relativer Positionierung geschieht.
- Viel Spass!