





3D-Visualisierung von bewegten Objekten

Vortrag in Anwendungen I



Agenda

- **Motivation**
- VRML 
- Animation
- Logik in der 3D-Welt
- X3D 
- VRML- und X3D-Viewer
- Editoren
- Mein Beitrag zum Ferienclub



Motivation

allgemein

- Sachen echt 3D zu betrachten ist wesentlich schöner als nur 2D
 - Architektur-Modelle
 - medizinische Anwendungen

persönlich

- mit 3D-Anwendungen beschäftigen



Übersicht 3D-Produkte

standardisiert:



- VRML / X3D
- java3D

proprietär:

- Anfy 3D12
- Shout 3D13
- Cult 3D14
- viewpoint



Agenda

- Motivation
- VRML 
- Animation
- Logik in der 3D-Welt
- X3D 
- VRML- und X3D-Viewer
- Editoren
- Mein Beitrag zum Ferienclub



Geschichte

3D im Web

- 1994 - 1995: VRML 1.0
- 1995 - 1996 : VRML 2.0
- 1997: VRML 2.0 wird als VRML97 ISO-Standard
- 1997: Gründung des VRML-Konsortiums
- 1999: Umbenennung in Web3D-Konsortium
- 2002: X3D



VRML - Beispiel

- Einführungs-Demo



VRML - Beispiel

- Beispiel: Würfel



VRML - Beispiel

- Beispiel: Würfel

```
#VRML V2.0 utf8
Shape
{
  geometry Box { }
  appearance Appearance
  {
    material Material
    {
      diffuseColor 1.0 0.0 0.0
    }
  }
}
```



VRML - Struktur

- Alles zusammen im Szenegraph: Verschachtelte Knoten
- Knoten entspr. Objekte
- Datenfelder entspr. Attributen



VRML - Wiederverwendung

- Knoten können Namen bekommen, um später zu referenzieren (DEF name)
- weiterhin kann man sie später erneut instantiieren (USE name)
- Prototypen definierbar
 - Attribute später veränderbar
 - getrennt in Interface und Implementierung
 - auch in anderen Dateien definierbar
- mit `Inline` kann man andere Dateien einbinden



VRML - geometrische Knoten

- sichtbarer Teil der 3D-Welt
- werden im Knoten Shape angeordnet
- die „einfachen“ geometrischen Knoten
 - Box: Quader
 - Cone: Kegel
 - Cylinder
 - Sphere: Kugel
 - Text
 - ...



VRML - geometrische Knoten

- die „komplizierten“ geometrischen Knoten
 - ...
 - ElevationGrid
 - PointSet
 - IndexedLineSet
 - IndexedFaceSet
 - Extrusion



VRML - Aussehen

Aussehen der geometrischen Knoten:

Unterknoten Appearance

- Material
- Texturen
 - ImageTexture
 - MovieTexture
 - PixelTexture
- TextureTransform



VRML - Aussehen

Aussehen: Unterknoten `material`

- `Farben`
 - `diffuseColor` (Eigenfarbe)
 - `emissiveColor` (Eigenleuchten)
 - `specularColor` (Farbe von Reflexionen)
- `shininess` (Reflexionsgrad)
- `transparency` (Grad der Transparenz)
- `ambientIntensity` (Leuchtkraft)



VRML - Hintergrund

Knoten BackGround

- Boden und Himmel
- jeweils Farben und Farbübergänge möglich
 - besonders am Himmel interessant



VRML - Lichtquellen

verschiedene Lichtquellen möglich:

- DirectionalLight
- PointLight
- SpotLight



VRML - Beobachter

Beobachter:

- avatarSize (Größe des Beobachter)
- headlight (Helmlicht)
- speed
- type
- visibilityLimit



VRML - Viewpoints

```
Viewpoint
{
    position    0    0    10
    orientation  0    0    1    0
    jump FALSE
    description "von oben"
}
```



VRML - Hello World



```
#VRML V2.0 utf8
Group {
  children [
    Viewpoint {
      description "hello, world!"
      orientation 0 1 0 1.57
      position 6 -1 0
    }
    NavigationInfo {
      type [ "EXAMINE" "ANY" ]
    }
    Shape { # Weltkugel
      geometry Sphere {
      }
      appearance Appearance {
        texture ImageTexture {
          url [ "earth-topo.png" ]
        }
      }
    } # Shape
  ]
  # ... Fortsetzung
```

```
# ... Fortsetzung

Transform {
  rotation 0 1 0 1.57
  translation 0 -2 1.25
  children [
    Shape {
      geometry Text {
        string [ "Hello" "world!" ]
      }
      appearance Appearance {
        material Material {
          diffuseColor 0.1 0.5 1
        }
      }
    } # Shape
  ] # children
} # Transform
] # children
} # Group
```



Agenda

- Motivation
- VRML 
- **Animation**
- Logik in der 3D-Welt
- X3D 
- VRML- und X3D-Viewer
- Editoren
- Mein Beitrag zum Ferienclub



Animation - Beispiel

- Beispiel: bewegter Würfel



Animation

Animation besteht aus folgenden Komponenten:

- Sensor (auslösen von und reagieren auf Ereignisse)
- Routen = Verbindungen zwischen Objekten
- Interpolator (berechnet Zwischenwerte)
- (geometrisches) Objekt



Animation

Üblicherweise veränderte Felder:

- translation
- rotation
- scale
- Kindsknoten hinzufügen oder entfernen



Animation - Sensoren

- TimeSensor: Zeitgeber
- TouchSensor: Berührungen mit Mauszeiger
- VisibilitySensor: Sichtbarkeit eines Objektes
- ProximitySensor: Betrachters betritt Bereich
- PlaneSensor: Verschiebung eines Objektes in xy
- CylinderSensor: Objekt um die y-Achse drehen
- SphereSensor: Rollen eines Objektes



Animation - Ereignisse

Ereignisse

- werden bei Veränderung eines Werts ausgelöst
→ „Methodenaufruf“
- werden durch bestimmte Felder repräsentiert



Animation - Routen

- Ereignisse werden entlang der Routen verschickt
- von einem Datenfeld eines Objektes zu einem Datenfeld eines Objektes

```
ROUTE Uhr.fraction_changed TO  
      Interpolator.set_fraction
```



Animation - Interpolatoren

Interpolatoren berechnen Zwischenwerte

- PositionInterpolator: (x, y, z)
- CoordinationInterpolator: (für komplexe Flächen)
- NormalInterpolator: (über Normalenvektoren)
- OrientationInterpolator (Achse mit Drehung)
- ColorInterpolator: (RGB-Werte)
- ScalarInterpolator: (einzelne Werte)



Animation - Interpolatoren

- Zwischenwerte werden aufgrund gegebener Anfangsintervalle berechnet
- Beispiel für Transparenz:

```
DEF Name ScalarInterpolator {  
  key [0.0 0.9 0.0]  
  keyValue [  
    0.0, # undurchsichtig  
    1.0, # durchsichtig  
    0.0  # undurchsichtig  
  ]  
}
```



Animation - Beispiel

- Beispiel: bewegter Würfel

```
#VRML V2.0 utf8
# Animation

Background { skyColor 1 1 1 }

DEF Wuerfel Transform
{
  children
  [
    Shape
    {
      geometry Box {}
      appearance Appearance
      {
        material Material {}
      }
    }
  ]
}
# Fortsetzung ...
```

```
# ... Fortsetzung



DEF Uhr TimeSensor {
  cycleInterval 5.0
  loop TRUE
}

DEF Interp PositionInterpolator {
  key [ 0.0 0.9 1.0 ]
  keyValue
  [
    0.0 -2.0 0.0,
    0.0  2.0 0.0,
    0.0 -2.0 0.0
  ]
}

ROUTE Uhr.fraction_changed
      TO Interp.set_fraction
ROUTE Interp.value_changed
      TO Wuerfel.set_translation
```



Agenda

- Motivation
- VRML 
- Animation
- **Logik in der 3D-Welt**
- X3D 
- VRML- und X3D-Viewer
- Editoren
- Mein Beitrag zum Ferienclub



Logik in VRML

- bisherige Knoten etc. reichen für komplexe Animationen nicht aus
→ Logik nötig



Logik in VRML

- bisherige Knoten etc. reichen für komplexe Animationen nicht aus
→ Logik nötig
- Programmiersprachen
 - JavaScript
 - Java



Logik - Schnittstelle

Schnittstelle zwischen VRML und Prog-Sprache

- Knoten vom Typ Script

```
DEF Name Script
{
    field      SFVec3f    position [1 2 3]
    eventIn    SFFloat    setHoehe
    eventOut   SFInt32    hoeheChanged
    url "klasse.class"
}
```



Logik - Java

Packages, die der Viewer zur Verfügung stellt:

- `vrml.*`
- `vrml.field.*`
- `vrml.node.*`



Logik - Java

Implementierung erbt von der Klasse `vrml.Script`.

- dort beliebige Funktionalität

Ereignis-Verarbeitung



- `processEvent(Event)`

Zugriff auf definierte Ereignisse (Felder entsprechend)

- `getEventOut(String name)`
- `getEventIn(String name)`



Agenda

- Motivation
- VRML 
- Animation
- Logik in der 3D-Welt
- **X3D** 
- VRML- und X3D-Viewer
- Editoren
- Mein Beitrag zum Ferienclub



X3D

- Nachfolger von VRML
- Umwandlung VRML \leftrightarrow X3D möglich
 - automatisch
 - quasi ohne Verlust



X3D

- Würfel-Beispiel:
VRML

```
#VRML V2.0 utf8
```

```
Shape {  
  geometry Box { }  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
    }  
  }  
}
```



X3D

- Würfel-Beispiel:
VRML

```
#VRML V2.0 utf8
```

```
Shape {  
  geometry Box { }  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
    }  
  }  
}
```

X3D

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"  
  "http://www.web3d.org/specifications/x3d-3.0.dtd">  
<X3D profile="Full">  
  <Scene>  
    <Shape>  
      <Box />  
      <Appearance>  
        <Material  
          diffuseColor="1.0 0.0 0.0"/>  
      </Appearance>  
    </Shape>  
  </Scene>  
</X3D>
```




X3D

- Struktur von X3D im Prinzip gleich
 - VRML-Knoten werden XML-Elemente
 - Datenfelder werden XML-Attribute
- ein Vorteil von XML: Einfacher lesbar, da immer klar ist, welche End-Klammer zu welchem Anfang gehört



X3D - Komponenten

- X3D ist modular aufgebaut
- für VRML gibt es keinen Browser, der die gesamte Spezifikation erfüllt.
- spezialisierte Browser möglich, die nur einen Teil, den aber optimal implementieren
- X3D ist erweiterbar (XML-typisch)





X3D - Profile

Profile setzen sich aus Komponenten zusammen

- Core Profile
 - ganz minimal
- Interchange Profile
 - beschränkten Lichteinsatz, keine Interaktion
- Interactive Profile
 - nur eingeschränkte Interaktion erlaubt
- MPEG-4 interactive Profile: MPEG-4 Standard
- Immersive Profile: volle VRML Funktionalität
- Full Profile
 - Volle Implementation der X3D Spezifikation



Agenda

- Motivation
- VRML 
- Animation
- Logik in der 3D-Welt
- X3D 
- **VRML- und X3D-Viewer**
- **Editoren**
- Mein Beitrag zum Ferienclub



VRML- und X3D-Viewer

- Browser-Plugin von Cortona (VRML)
 - für private Nutzung kostenlos
- Browser-Plugin von www.bitmanagement.de (beides)
 - zum Testen kostenlose Version
- Xj3D vom Web3D-Konsortium (X3D)

Für PocketPCs und Handhelds

- Pocket Cortona



VRML- und X3D-Editoren

- Dune
 - Einfacher grafischer Editor zum Bearbeiten von VRML-Dateien; nach 5 Minuten intuitiv bedienbar
 - X3D-Export möglich; Java-Bibliotheken werden benötigt
- Blender3d
 - sieht sehr professionell aus
 - Probleme mit der Anzeige
 - vermutlich überdimensioniert

(Links finden sich bei den Quellangaben)





VRML- und X3D-Editoren

- mjbWorld
 - sehr technisch orientiert
- X3D-Edit
 - Einfacher Editor vom Web3D-Konsortium
 - Ließ sich bei mir nicht starten



Agenda

- Motivation
- VRML 
- Animation
- Logik in der 3D-Welt
- X3D 
- VRML- und X3D-Viewer
- Editoren
- **Mein Beitrag zum Ferienclub**



Mein Beitrag

3D-Visualisierung des Ferienclubs

Anzeige von

- Gelände
- Attraktionen / Sportstätten / ...
- Aufenthaltsort von Personen
 - Wo sind / waren unsere Kinder
 - Wo sind meine Freunde gerade?



Mein Beitrag

Aufenthalt von Personen

- Anfang: statisch
 - aktuell
 - zu einem bestimmten Zeitpunkt
- Ziel: „Film“
 - Realtime
 - wo war eine Person über einen gewissen Zeitraum?



Mein Beitrag

Anzeigen von Personen

- Person realistisch darstellen
 - Körper, evtl. in den Farben der jeweiligen Kleidung
 - Versuch: Gesicht anzeigen (mit Foto)



Ich benötige

- Wer ist und war wo?
 - Feststellung der Positionen der einzelnen Gäste
 - Daten der RFID-Tags
- Farben der Kleidung
- Fotos der Gesichter



Ich benötige

- Positionen der Personen
 - müssen ständig aktuell gehalten werden
 - ständiges Abfragen nötig
 - viel Datenverkehr
- evtl. Einsatz von Agenten
 - Abfragen der Daten der RFIDs
 - nur Veränderungen an die Anwendung schicken



Ich benötige

- sonstige Daten der Personen
 - Gesichter
 - Kleidung
- ändern sich üblicherweise nicht
 - Abfrage über Webservice



Quellen

- <http://www.vrml2.de/>, 2005-05-26
- <http://www.debacher.de/vrml/>, 2005-05-27
- <http://www.web3d.org/>, 2005-05-26
- <http://www-lehre.inf.uos.de/~okrone/DIP/node35.html>, 2005-06-02
- XML-basierte 3D-Szenebeschreibungssprache - Praxistauglichkeit des VRML-Nachfolgers X3D (Dilpomarbeit von Stefan Rother an der HAW, 2003)
- <http://www.hoepnet.de/index.php?id=35&action=detail&sid=21&cat=22>, 2005-06-02
- <http://www.oreilly.de/catalog/perlmodger/manpage/vrmlpm.htm>, 2005-06-03
- <http://www.foruma.de/vrml-tutorial/index.htm>, 2005-06-03
- <http://www.uni-essen.de/hrz/beratung/hrzblatt/hrz150/vrml.html>, 2005-06-03
- <http://www.csv.ica.uni-stuttgart.de/vrml/howto.html>, 2005-06-06
- http://www.gisuser.com.au/GU/content/2001/GU47/gu47_feature/gu47_feature_3.html, 2005-06-07
- <http://www.google.de/search?q=vrml>, immer :-)

Multimedia-Labor der HAW

- <http://mmlab.haw-hamburg.de/>



Quellen

Viewer

- Cortona: Browser-Plugin für VRML-Dateien:
<http://www.parallelgraphics.com/products/cortona/>, 2005-05-26
- Cortona: Viewer für Pocket PC und Handhelp PC
<http://www.parallelgraphics.com/products/cortonace/>, 2005-06-02
- Browser-Plugin für VRML und X3D: <http://www.bitmanagement.de/>, 2005-05-28
- Xj3D: <http://www.web3d.org/x3d/applications/xj3d/>, 2005-06-03

Editoren:

- <http://www.csv.ica.uni-stuttgart.de/homes/js/linuxtag/editoren.html>, 2005-06-07
- Blender3D: <http://www.blender3d.org>, 2005-05-28
 - Dune: http://www.csv.ica.uni-stuttgart.de/vrml/dune/index_white_dune.html, 2005-05-28
 - X3D-Edit: <http://www.web3d.org/x3d/content/README.X3D-Edit.html>, 2005-05-28
 - mjbWorld: <http://www.euclideanspace.com/mjbWorld/index.htm>, 2005-06-02
 - Liste mit vielen Programmen:
<http://www.csv.ica.uni-stuttgart.de/homes/js/linuxtag/editoren.html>, 2005-06-07



Fragen ??



Danke für die
Aufmerksamkeit