



Szenengraphen

Codruța Cosma

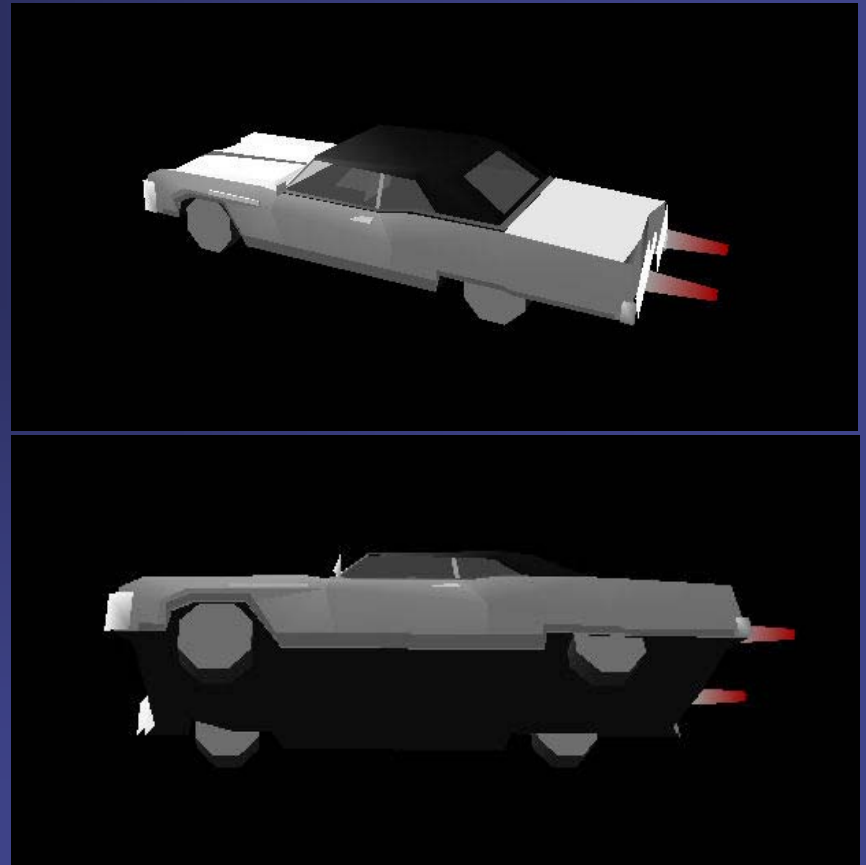
Universität Ulm
Sommersemester 2005

Übersicht

- Einführung
- VRML
- OpenSceneGraph
- Java3D vs. VRML
- OpenGL vs. Java3D und VRML
- Zusammenfassung

Was sind Szenengraphen?

- Datenstruktur für 3D-Szenen
- aus graphentheoretischer Sicht ein Baum
- hierarchische Anordnung der Objekte
- Koordinatenarten



Allgemeines

- Virtuelle Welt als komplettes 3D-Model
- Farben, Texturen, Lichteffekte
- Schatten

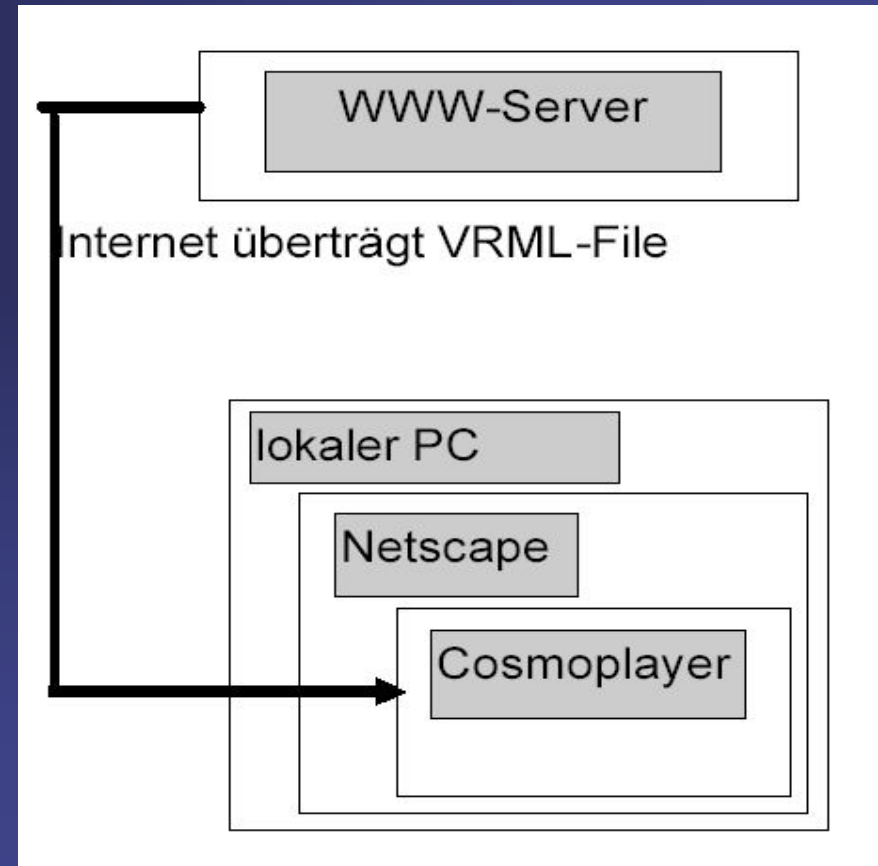


Allgemeines

- Navigation mit Maus und Tastatur
- Stereosound
- Stereovision und natürliches „Umsehen“ für die Zukunft in Planung

Graphiksprache VRML

- Virtual Reality Modelling Language
- Modellierung von 3D Objekten und Szenen im Internet
- VRML-Datei über Internet auf PC übertragen und hier interpretiert



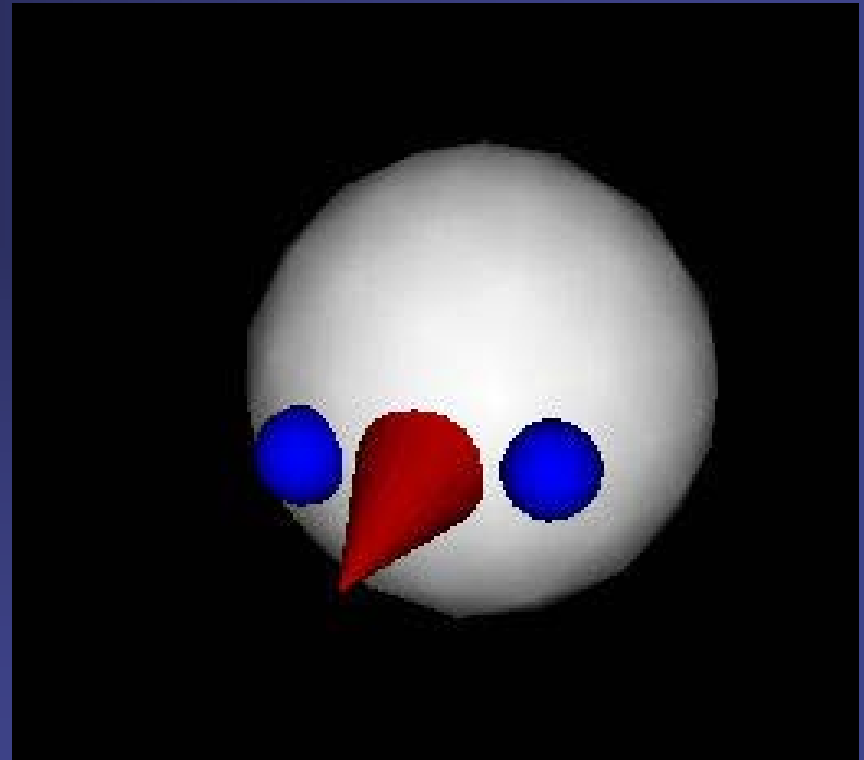
Graphiksprache VRML (2)

- Objekterstellung baut auf einfache geometrische Formen sog. Knoten auf
- Positionierung hierarchisch und mithilfe von Koordinaten
- Transformationsknoten

Graphiksprache VRML (3)

■ Einfaches Beispiel:

- # VRML V2.0 utf8
- # Schneemann mit blauen Augen und roter Nase
- # Kopf
- Shape {
- geometry Sphere { radius 1 }
- appearance Appearance {
- material Material { diffuseColor 1 1 1 }
- }
- }
- # linkes Auge (30° von der Nase)
- Transform {
- translation .5 0 .867
- children Shape {
- geometry Sphere { radius .2 }
- appearance Appearance {
- material Material {diffuseColor 0 0 1 }
- }
- }
- }
- # rechtes Auge (30° von der Nase) dto.

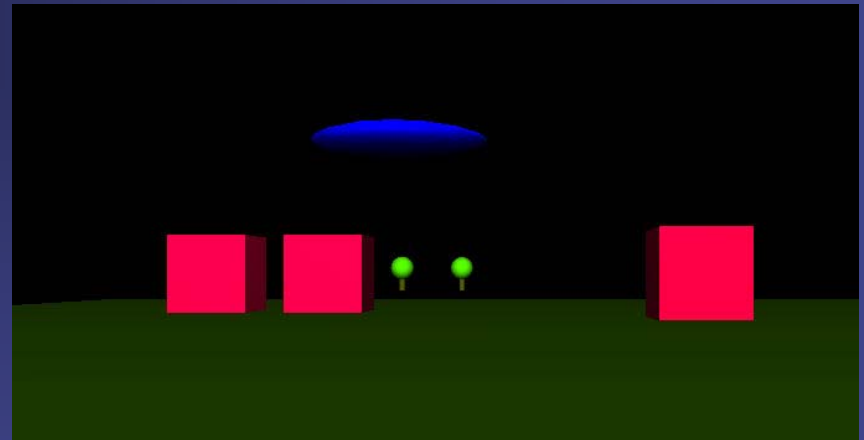


Graphiksprache VRML (4)

- Farben, Materialien, Texturen und Reflexion möglich
- Unterschiedliche Belechtungsknoten
- Laden einer Szene von URL
- Interaktivität durch Sensoren: TouchSensor, ProximitySensor u.a. -> event-orientiert

Graphiksprache VRML (5)

- Animation durch Interpolatoren: z.B. PositionInterpolator
- TimeSensor steuert Interpolatoren
- Extrusion möglich
- Ankerknoten



Graphiksprache VRML (6)

- Navigation im Bild mit Maus und Tastatur
- Objekte drehen, bewegen (walking, flying, pointing, looking)
- Erkennung von Kollisionen limitiert
- verschiedene Viewpoints

OpenSceneGraph

- open-source Plattform zur Erstellung von hochperformanten Grafikapplikationen
 - objektorientiertes System
 - baut auf OpenGL auf
 - stellt zahlreiche Plug-ins und Applikationen zur Verfügung zur Erstellung und Bearbeitung von Szenengraphen
-

OpenSceneGraph(2)

- Anwendung der Basis Physik mit Kollisionenbehandlung und rigide Körper
- Beleuchtung: per-vertex, anisotrop
- Texturen, Schatten, Skinning, Rendering
- Grafik-Import

OpenSceneGraph(3)

- Herausfilterung von Verdeckungen
- Animation: Key-Frame Animation, skelettartige Animation
- Special Effects: Spiegel, Billboarding
- 2D-, 3D- und Streaming-Sound

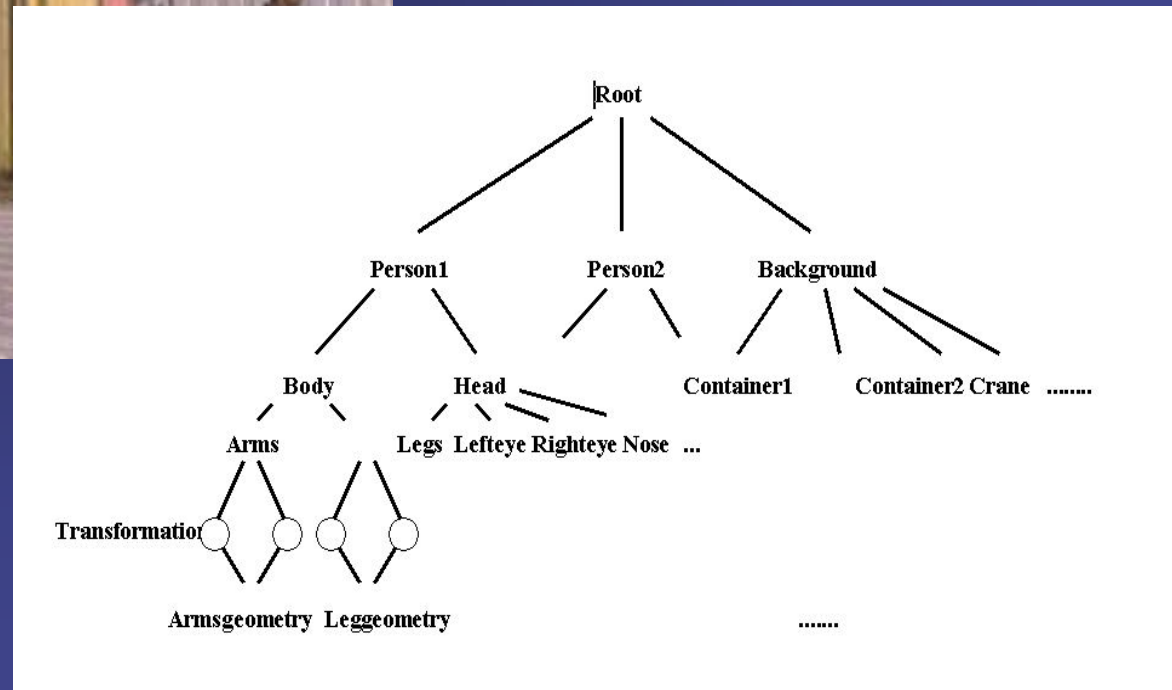


OpenSceneGraph(4)

- dient zur Erstellung von Flugsimulatoren, virtuelle Welten, Spiele oder wissenschaftliche Visualisierung



Beispiel Szenengraph-Baum



Nasa's blue marble project

Nasa's Blue Marble



Java3D vs. VRML

- Erweiterung von Java die es erlaubt 3D-Welten zu programmieren
- VRML -> einfache Inhalte
- Java3D -> komplexe Inhalte und Applikationen

Java3D vs. VRML (2)

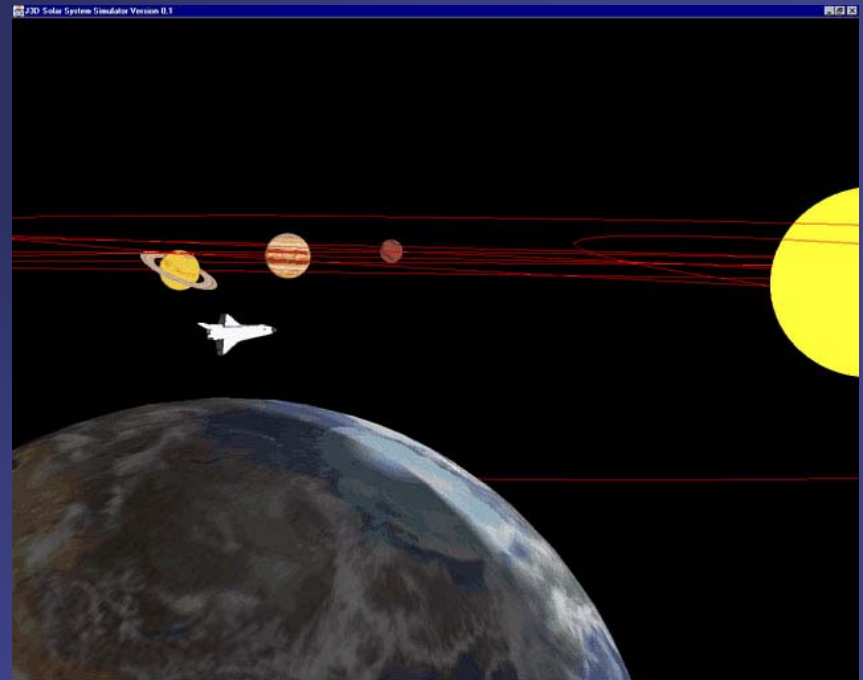
- baut ebenfalls auf Hierarchien von Knoten auf
- gleiche Konzepte für Formen, Materialien, Transformationen und Beleuchtung
- komplexer Support für Geometrie, Texturen, Transparenz

Java3D vs. VRML (3)

- bessere Realisierung von Beleuchtung und Nebel
- Sound-Playback und -Nachklang effizienter
- bessere Qualität für Viewing und Rendering

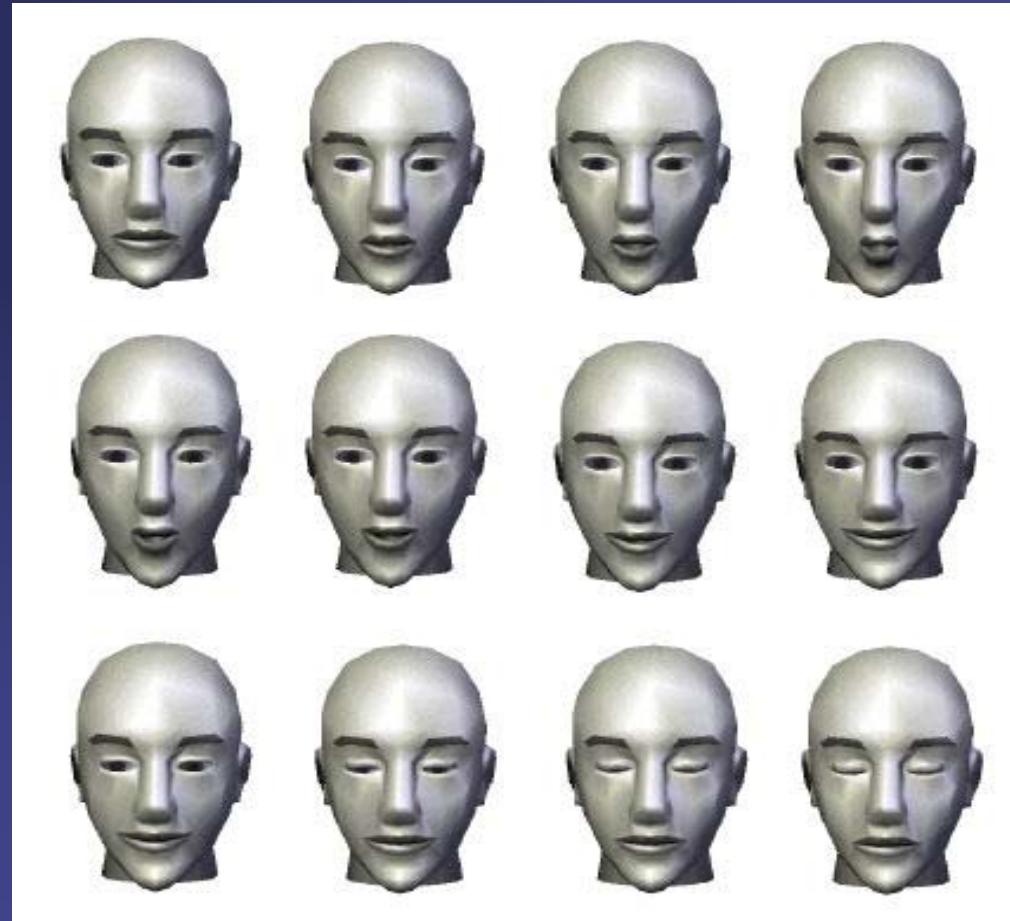
Java3D vs. VRML (4)

- bessere Auflagen für Hintergründe, Kollisionenbehandlung
- neu in Java3D: Clipping und Behaviors
- bessere Interpolatoren -> bessere Interaktivität



Java3D vs. VRML (5)

- Java3D Avatare
- Animation von VRML-Avataren mit Java3D möglich



Java3D vs. VRML (6)

- Wiederverwendung : Java3D (share/link clone) , VRML (def/use, prototypes, inline)
- Java3D -> mehr Interpolatoren (Transparenz-, SwitchInterpolator)
- Java3D -> Scheduler, Aktivierungsfunktion

Java3D vs. VRML (7)

- VRML -> Viewpoints, Extrusion, vordefinierte Formen der primitiven Geometrie (Zylinder, Kugel, Würfel)
- beide -> Zeitgeber, Ankerknoten
- beide -> Hintergrund, Level of Detail (LOD), Billboard

OpenGL vs. Java3D und VRML

- Anti-Aliasing : alle 3, in VRML schlecht
- Schnittstelle zu anderen Applikationen
- Morphing nur Java3D
- Shading in OpenGL und Java3D beeinflussbar
- OpenGL – kein Sound, keine Interpolatoren, keine Wiederverwendung



Zusammenfassung

- unterschiedliche Programmiersprachen für 3D-Applikationen
- ähnlicher Aufbau der Szenengraphen
- unterschiedliche Features mit unterschiedlicher Qualität
- Wahl der Graphiksprache projektorientiert