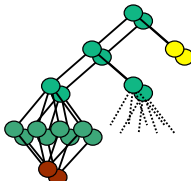





Virtuelle Realität

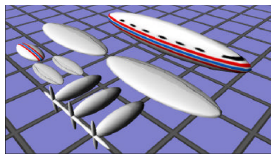
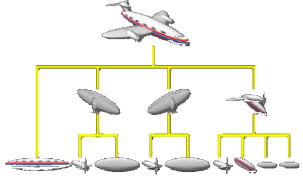
Scenegraphs (Game Engines)



G. Zachmann
Clausthal University, Germany
cg.in.tu-clausthal.de

Motivation

- **Immediate mode vs. retained mode:**
 - Immediate mode = OpenGL / Direc3D = Applikation schickt Polygone / State-Befehle an die Grafik → flexibler
 - Retained mode = Scenegraph = App. legt vordefinierte Datenstrukturen an, die Pgone und States speichern → bequemer und evtl. effizienter
- **Flach vs. hierarchische Datenstrukturen:**


 - *Code re-use* und *Know-how re-use!*
 - **Descriptive, not imperative** (cv. C vs. Prolog)
 - Thinking objects ... not rendering processes

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 2

Struktur eines Szenegraphen

- Gerichteter, azyklischer Graph, i.A. ein echter Baum
- Heterogene Knoten
- Beispiel:

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 3

Semantik

- Semantik der Knoten:
 - Wurzel = "Universum"
 - Blätter = "*content*" (Geometrie, Sound, ...)
 - Innere Knoten = Gruppierung, State(-änderungen), und nicht-geometrische Funktionalität (z.B. Transf.)
- Gruppierung: nach welchen Kriterien bleibt der Applikation überlassen:
 - Geometrische Nähe? (Scenegraph induziert BV-Hierarchie!)
 - Nach Material? (state changes kosten Performance!)
 - Nach log. Bedeutung? (alle Wasserleitungen, alle Kabel, alle Sitze, ...)
- Semantik der Kanten = Vererbung des "State"
 - Transformation
 - Material
 - Lichtquellen

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 4

Knotenarten

- 2 Hierarchien: Szenegraph-Hierarchie + Klassenhierarchie
- Die Mächtigkeit und Flexibilität eines Szenengraphen hängt von der Menge der zur Verfügung stehenden Knotenklassen ab!
- Etliche Klassen sind nicht Teil des Szenengraphen, aber doch Teil der Szene

```

graph TD
    Knoten --> InnereKnoten
    Knoten --> Blaetter
    InnereKnoten --> LOD
    InnereKnoten --> Transform
    InnereKnoten --> Group
    Blaetter --> Geometrie
    Blaetter --> Partikelsystem
    Blaetter --> Sound
    NichtKnoten --> Material
    NichtKnoten --> Textur
    NichtKnoten --> Lichtquellen
  
```

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 5

Transformationssemantik

- Alle Scenegraphs behandeln diese Semantik gleich
- Transformationsknoten = neues lokales Koordinatensystem
- ```

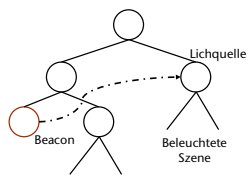
pushMatrix();
multMatrix(...);
traverse sub-tree;
popMatrix();

```

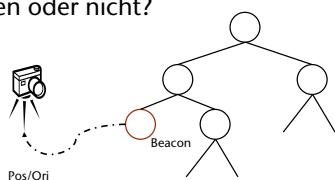
G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 6

## Issues

- **Lichtquellen:**
  - Teil des Szenengraphen (meistens)
  - Semantik (OpenSG):
    - beleuchtet Teilbaum darunter
    - Pos./Ori kommt von **Beacon**
  - Je nach Art (directional, point, spot) wird verschiedener Anteil der Transformation verwendet



- **Kamera: Knoten im Szenengraphen oder nicht? (gibt beide Varianten)**
  - Ja: Kamera ist ein Knotentyp
  - Nein: Beacon-Konzept



G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 7

## Material

- Property eines Knotens
- Vererbung: top-down
  - Pfad von Wurzel zu Blatt muß mindestens 1 Material haben
  - Folge:
    - Jedes Blatt wird mit eindeutig definiertem Material gerendert
    - Dieses läßt sich leicht bestimmen
- Schlechte Idee (Inventor): Vererbung left-to-right!

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 8

## Sharing von Geometrie

- Problem: große Szenen mit viel identischer Geometrie
- Idee: DAG (statt Baum)
  - Problem: Zeiger/Namen von Knoten sind **nicht mehr eindeutig!**
- Lösung: trenne Struktur von Inhalt
  - Baum besteht nur noch aus **einer** Sorte Knoten
  - Knoten bekommen **spezielle** Eigenschaften / Inhalt durch **Attachments / Properties**
  - Vorteile
    - alles wird share-bar
    - Viele Szenengraphen zur selben Szene möglich
    - Ein Knoten kann viele Attachments (= Eigenschaften) bekommen

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 9

## Multi-threading / Thread-safety

- Performer: App / Cull / Draw – Modell
- Idee: mehrere Szenengraphen
- Problem: Speicheraufwand
- Lösung:
  - "Aspects" und *Copy-on-Write* der Attachments
  - Jeder Thread "sieht" einen eigenen Aspect
  - Problem: einfacher Zugriff über Pointers  
`geom->vertex[0]`  
geht nicht mehr
  - Lösung:
    - Smart Pointers
    - Pro Klasse eine Pointerklasse. Bsp.:  
`geomptr = Geometry::create(...);`  
`geomptr->vertex[0] ...`

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 10

▪ Synchronisation: Changelists

▪ Distributed Rendering:

- Wunsch: Rendern auf einem Cluster
- Problem: Änderung des Szenengraphen propagieren
- Lösung: Changelists übertragen
  - Enthalten IDs von geänderten Knoten / Properties
  - Werden bei Update übertragen

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 11

Erweiterbarkeit

- Wunsch:
  - Neue Knoten als SOs/DLLs
  - Soll auch der Loader verstehen können (ohne Neu-Compilieren)
  - Alle Traversierungen sollen funktionieren
- Lösung:
  - Design Patterns (Factory, Visitor, ...)

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 12

## Anwendungskriterien für Szenengraphen

- Wann soll man Scenegraphs verwenden:
  - Komplexe Szenen: viele verschiedene Materialien, viel Geometrie, oft ist nur ein Teil zu sehen, komplexe Transformationshierarchien
  - Relativ statische Geometrie
  - Spezifische Features, die ein Scenegraph bietet (Partikel, Clustering, ...)
- Wann man einen Scenegraph **nicht** verwenden soll:
  - Einfache Szenen (ein Objekt in der Mitte)
  - Hochgradig dynamische Geometrie

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 13

## Einige (ehem.) populäre Scenegraphs

- SGI Performer (<http://www.sgi.com/software/performer/>)
- Java3D (<http://java.sun.com/products/java-media/3D/>)
- Inventor/Coin (<http://oss.sgi.com/projects/inventor/> , <http://www.coin3d.org/>)
- VRML & X3D
- OpenSG (<http://www.opensg.org/>) !
- Open Scene Graph
- Viele andere (siehe [www.sf.net](http://www.sf.net) , "Game Engines List", ...)

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 14

## Geschichte

- Zusammen mit der OO Programming
- Davor eher "flache" Strukturen (GKS, Starbase, Phigs)
- Inzwischen nur noch open-source Scenegraphs und Game-Engines

The chart shows the active periods of various 3D rendering technologies from 1992 to 2005. Red lightning bolts indicate the end of a technology's active period.

| Technology    | Start Year | End Year |
|---------------|------------|----------|
| Open Inventor | 1992       | 1995     |
| Performer     | 1992       | 1997     |
| Y             | 1994       | 1997     |
| Cosmo3d       | 1995       | 1997     |
| Optimizer     | 1996       | 1999     |
| DirectModel   | 1997       | 1999     |
| OpenGL++      | 1997       | 2004     |
| Java3d        | 1998       | 2004     |
| Fahrenheit    | 1998       | 2000     |
| OpenSG        | 1998       | 2005     |

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 15

## Exkurs: Memory-Layout für schnelles Rendering

- Häufiges Problem: elegante Strukturierung der Daten (aus Sicht des Software-Engineerings) ist ungünstig für schnelles Rendering
- Terminologie: "Array of Structs (AoS)" vs. "Struct of Arrays (SoA)"
- Zur Illustration: Beispiel Molekül-Visualisierung
  - Sauberes Software-Engineering würde folgende Klasse enthalten

```

class Atom
{
public:
 Atom(uint atom_number, Vec3 position, ...);
private:
 Vec3 position_m;
 uint atom_number_m;
 Atom bonds[max_num_bonds];
 ...
};

```

G. Zachmann Virtuelle Realität und Simulation – WS 11/12 Scenegraphs 16



- Damit wäre dann eine Molekül folgendermaßen zu definieren:

```
class Molekule
{
public:
 Molekule(const std::vector<Atom> & atoms);
private:
 std::vector<Atom> atoms;
 ...
};
```

- Memory Layout = Array of Structs:

|     |     |       |     |     |       |     |     |       |  |
|-----|-----|-------|-----|-----|-------|-----|-----|-------|--|
| pos | num | bonds | pos | num | bonds | pos | num | bonds |  |
|-----|-----|-------|-----|-----|-------|-----|-----|-------|--|

- Problem dabei: Speicher-Transfer ist sehr langsam
- Alternative: Struct of Arrays

