

Szenegraphen: OpenInventor

Tobias Sielhorst

27 November 2006

chair for computer aided medical procedures

department of computer science | technische universität münchen

Szenegraphen – Warum?

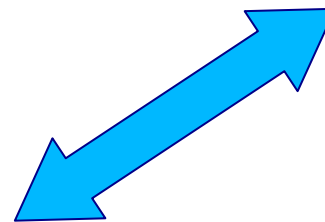
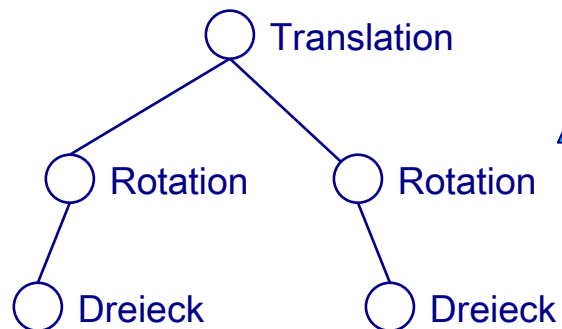
- Beispielszene:
 - Zwei Dreiecke im Raum
 - Beide um (1,2,1) verschoben
 - Jedes einzelne gedreht
- Kleine Szene – viel Text
 - Große Szenen?
 - Unübersichtliches Verhältnis der Objekte
 - Laden/Speichern/Verändern?

```
glTranslatef(1,2,1);
glPushMatrix();
glRotatef(1,0.3,0.4);
glBegin(GL_TRIANGLES);
    glVertexf(1,1,1);
    glVertexf(1,2,1);
    glVertexf(1,1,2);
glEnd();
glPopMatrix();
glRotatef(1,0.2,1);
glBegin(GL_TRIANGLES);
    glVertexf(1,1,1);
    glVertexf(1,2,1);
    glVertexf(1,1,2);
glEnd();
```

→ Szenegraphen

Szenegraphen

- Jeder Stapel (state machine) lässt sich in eine Baumdarstellung umwandeln
- Eine Traversierung (Tiefensuche) bringt den Stapel wieder hervor



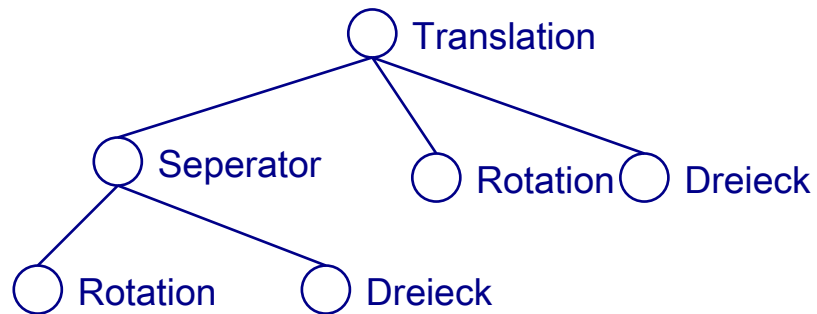
```

glTranslatef(1,2,1);
glPushMatrix();
glRotatef(1,0.3,0.4);
glBegin(GL_TRIANGLES);
    glVertexf(1,1,1);
    glVertexf(1,2,1);
    glVertexf(1,1,2);
glEnd();
glPopMatrix();
glRotatef(1,0.2,1);
glBegin(GL_TRIANGLES);
    glVertexf(1,1,1);
    glVertexf(1,2,1);
    glVertexf(1,1,2);
glEnd();
    
```

Verschiedene Szenegraphen für C++

- OpenInventor
 - SGI (LGPL) Referenzimplementierung, nicht ganz aktuell
<http://oss.sgi.com/projects/inventor/>
 - Coin3D (GPL oder PEL) <http://www.coin3d.org>
 - TGS (PEL)
- Direct3D based on DirectDraw, not on OpenGL
- OpenSceneGraph <http://www.openscenegraph.org> (different open source licences)
- OpenSG <http://www.opensg.org> (LGPL)

Das Beispiel in OI



```

glTranslatef(1,2,1);
glPushMatrix();
glRotatef(1,0.3,0.4);
glBegin(GL_TRIANGLES);
    glVertexf(1,1,1);
    glVertexf(1,2,1);
    glVertexf(1,1,2);
glEnd();
glPopMatrix();
glRotatef(1,0.2,1)
glBegin(GL_TRIANGLES);
    glVertexf(1,1,1);
    glVertexf(1,2,1);
    glVertexf(1,1,2);
glEnd();
    
```

- OI behält alle OpenGL Zustände bei, wenn kein Separator eingefügt wurde – der Separator ersetzt also glPush...() und glPop...()

Was muss ich machen?

- OpenInventor Bibliothek ins Projekt einbinden (lib-Datei angeben, .lib-Pfade Datei und .h-Pfade angeben)
- Fenster mit GL-Kontext erzeugen
- Hello Cone:

```
#define COIN_DLL
#include <Inventor/SoDB.h>
#include <Inventor/SbViewportRegion.h>
#include <Inventor/actions/SoGLRenderAction.h>
#include <Inventor/nodes/SoSeparator.h>
#include <Inventor/nodes/SoCone.h>
#include <Inventor/nodes/SoMaterial.h>
#include <Inventor/nodes/SoDirectionalLight.h>

// Intialisierung
SoDB::init();
m_ivRoot = new SoSeparator;
m_ivRoot->ref();
m_ivViewport = new SbViewportRegion(640,480);
m_ivRender = new SoGLRenderAction(*m_ivViewport);
SoMaterial *myMaterial = new SoMaterial;
myMaterial->diffuseColor.setValue(1.0, 0.0, 0.0); // Red
m_ivRoot->addChild(new SoDirectionalLight);
m_ivRoot->addChild(myMaterial);
m_ivRoot->addChild(new SoCone);

// Für jedes Neuzeichnen
m_ivRender->apply(m_ivRoot);
```

Eine Datei laden

```
std::string filename = "../../code/Coin/data/Schraub.iv"
SoSeparator* myObjects;
SoInput mySceneInput;
if (!mySceneInput.openFile(fileName.c_str())) {
    std::cout << "File not found: "
              << fileName.c_str() << std::endl;
    return NULL;
}

myObjects= SoDB::readAll(&mySceneInput);
if ( retSeparator == NULL) {
    std::cout << "Problem reading file: "
              << fileName.c_str() << std::endl;
    return NULL;
}
mySceneInput.closeFile();
```

FAQ

- Welche Objekte gibt es außer dem Kegel?
Liste der verfügbaren Knoten
http://doc.coin3d.org/Coin/group__nodes.html
- Wie ändere ich die Farbe?
Mit einem Materialknoten SoMaterial
- Einfaches zusammenklicken einfacher Szenen mit
CoinDesigner <http://coindesigner.sourceforge.net>
- Komplexe Szenen mit einem komplexen Programm
Blender <http://www.blender.org>
- The Book about OpenInventor “The Inventor Mentor”
ISBN:0201624958
<http://www-evasion.imag.fr/Membres/Francois.Faure/doc/inventorMentor/sji.html>

Ein Objekt ändern

```
SoNode* objectToFind = m_ivRoot->getByName("NameVomObjekt");
SoSeparator* myObject;

if (objectToFind->getTypeId() ==
    SoSeparator::getClassTypeId() )
{
    myObject= (SoSeparator*) objectToFind;
    std::cout << "Das gefundene Objekt ist vom Typ SoSeperator";
}
```

- Mit der Funktion `getByName()` kann man einen Knoten suchen
- Bevor man den den Knoten in die Zielklasse umwandelt, sollte man überprüfen, ob es sich wirklich um die gewünschte Klasse handelt. (Bei OpenInventor funktioniert ein `dynamic_cast` nicht)

Tipps

- Niemals ein OpenInventor Objekt löschen (delete)
- Nicht im Szenegraphen referenzierte Objekte gehen verloren
- Dateiformate VRML und IV sind *sehr* ähnlich