

# Szenegraphen & Modellierung

## Szenegraphen – Warum?

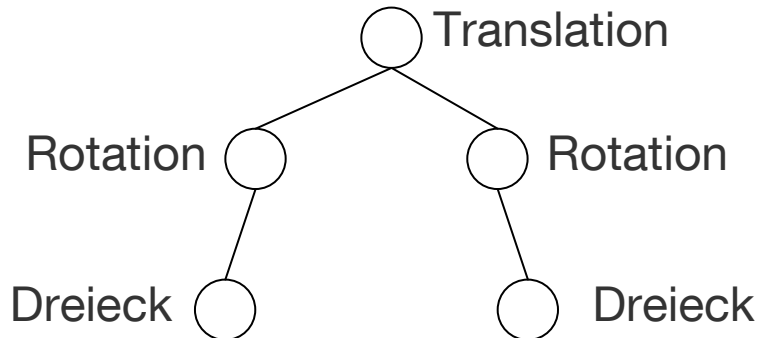
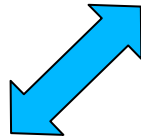
- Beispielszene:
  - Zwei Dreiecke im Raum
  - Beide um (1,2,1) verschoben
  - Jedes einzelne gedreht
- Kleine Szene – viel Text
  - Große Szenen?
  - Unübersichtliches Verhältnis der Objekte
  - Laden/Speichern/Verändern

```
glTranslatef(1.0, 2.0, 1.0);  
glPushMatrix();  
glRotatef(1.0, 0.3, 0.4);  
glBegin(GL_TRIANGLE);  
    glVertexf(1.0, 1.0, 1.0);  
    glVertexf(1.0, 2.0, 1.0);  
    glVertexf(1.0, 1.0, 2.0);  
glEnd();  
glPopMatrix();  
glRotatef(1.0, 0.2, 1.0);  
glBegin(GL_TRIANGLE);  
    glVertexf(1.0, 1.0, 1.0);  
    glVertexf(1.0, 2.0, 1.0);  
    glVertexf(1.0, 1.0, 2.0);  
glEnd();
```

 Szenegraphen

# Szenegraphen

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



```
glTranslatef(1.0, 2.0, 1.0);  
glPushMatrix();  
glRotatef(1.0, 0.3, 0.4);  
glBegin(GL_TRIANGLE);  
    glVertexf(1.0, 1.0, 1.0);  
    glVertexf(1.0, 2.0, 1.0);  
    glVertexf(1.0, 1.0, 2.0);  
glEnd();  
glPopMatrix();  
glRotatef(1.0, 0.2, 1.0);  
glBegin(GL_TRIANGLE);  
    glVertexf(1.0, 1.0, 1.0);  
    glVertexf(1.0, 2.0, 1.0);  
    glVertexf(1.0, 1.0, 2.0);  
glEnd();
```

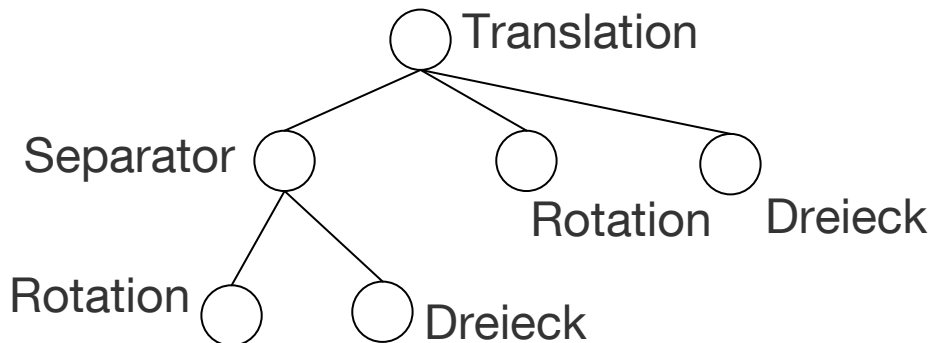
## Verschiedene Szenegraphen in C++

- OpenInventor
  - SGI Referenzimplementierung, nicht ganz aktuell
  - Coind3D
  - VGS
- OpenSceneGraph
- OpenSG

## Open Inventor Beispiel

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

```
glTranslatef(1.0, 2.0, 1.0);  
glPushMatrix();  
glRotatef(1.0, 0.3, 0.4);  
glBegin(GL_TRIANGLE);  
    glVertexf(1.0, 1.0, 1.0);  
    glVertexf(1.0, 2.0, 1.0);  
    glVertexf(1.0, 1.0, 2.0);  
glEnd();  
glPopMatrix();  
glRotatef(1.0, 0.2, 1.0);  
glBegin(GL_TRIANGLE);  
    glVertexf(1.0, 1.0, 1.0);  
    glVertexf(1.0, 2.0, 1.0);  
    glVertexf(1.0, 1.0, 2.0);  
glEnd();
```



## Was muss ich machen?

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

```
// Initialisierung
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);
m_ivRoot->addChild(new SoDirectionalLight);
m_ivRoot->addChild(myMaterial);
m_ivRoot->addChild(new SoCone);

...

// neu Zeichnen
m_ivRender->apply(m_ivRoot);
```

## Nodes:

- Simple Shapes
- Complex Shapes: Triangle Strip Set
- Property Nodes: Material, Light
- Transformation
- Manipulator
- Camera



Transform



Light



Appearance



Manipulator



Metric/Topology



Node Kit



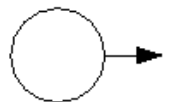
Property (Misc.)



SoSelection



Shape



Callback

## Ein Objekt ändern

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

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

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

## Tipps

- Niemals ein OpenInventor Objekt löschen (delete)
- Dateiformate VRML und IV sind *sehr* ähnlich
- Szenegraphen können mit dem Coindesigner gebaut und betrachtet werden



# Referenzen

- **The Inventor Mentor**