CS770/870 Fall 2008 Introduction to Scene Graphs

Related material in Hill and Kelley: None

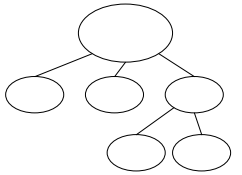
Preview

- Problem: How do we represent complicated scenes efficiently & flexibly?
- Solution: Hierarchical scene graph

The Problem

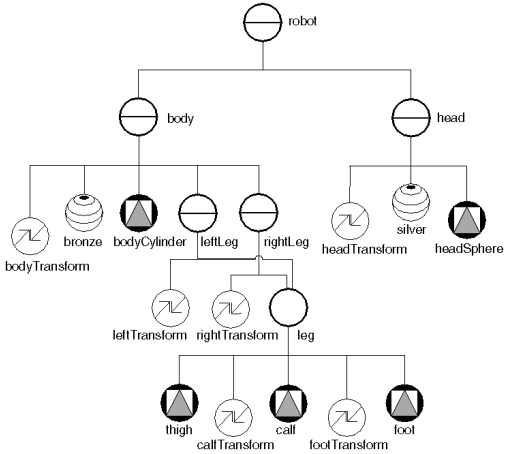
- How do we represent complicated scenes efficiently & flexibly?
  - Many 3D objects
    - composed of other objects
    - arbitrary reference frames between subobjects
    - Some subobjects share rendering state (OpenGL)
  - Down the road… want more efficient ways to
    - Render complex scenes
    - Select objects
    - Detect collisions between objects, etc.

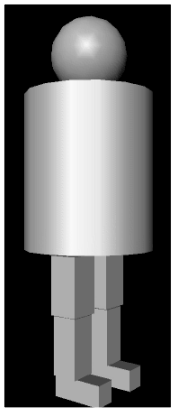
A Solution: Scene Graphs



- Hierarchy of Nodes
- To begin, let nodes represent
  - 3D geometry (vertices, lines, faces)
  - Transformation matrices
  - Rendering state (lighting, color, etc.)
  - Groupings of other nodes
- Rendering becomes a depth-first traversal

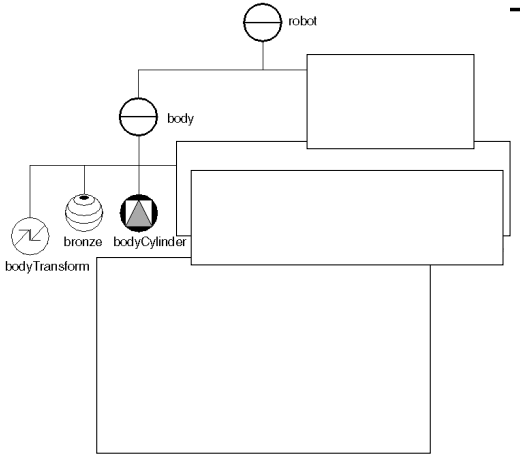
A Simple Scene Graph

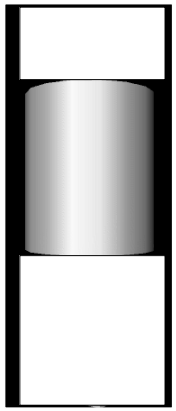




Figures 3-10 and 3-11 from *The Inventor Mentor*

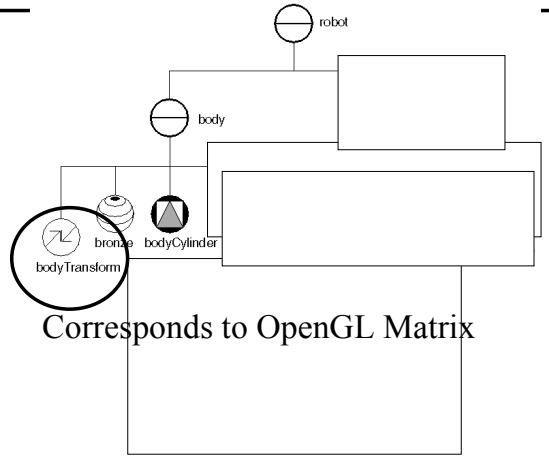
A Simple Scene Graph



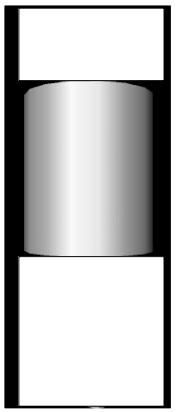


Figures 3-10 and 3-11 from *The Inventor Mentor*

A Simple Scene Graph

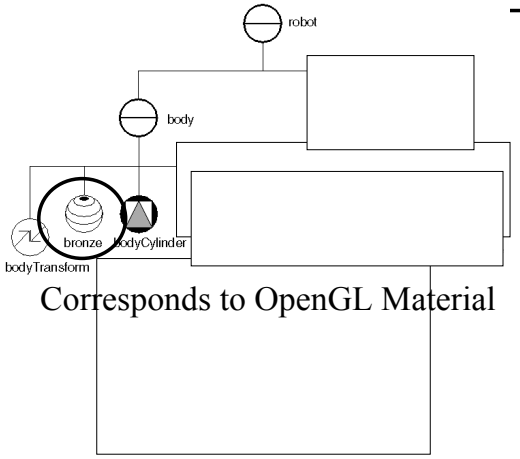


Corresponds to OpenGL Matrix

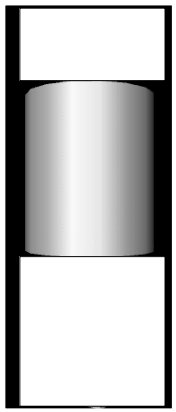


Figures 3-10 and 3-11 from *The Inventor Mentor*

A Simple Scene Graph

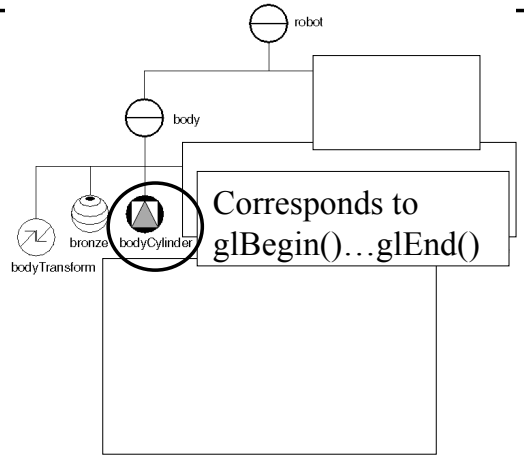


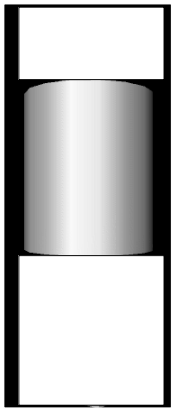
Corresponds to OpenGL Material



Figures 3-10 and 3-11 from *The Inventor Mentor*

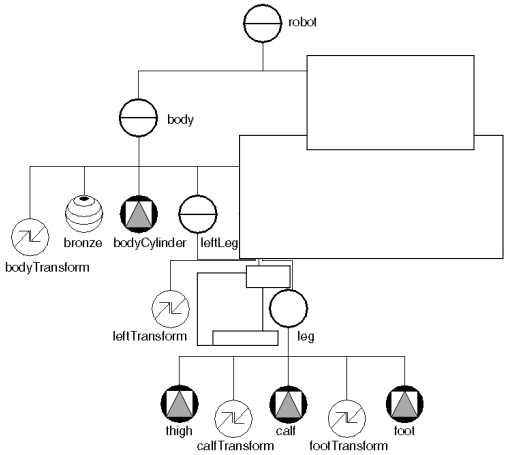
A Simple Scene Graph

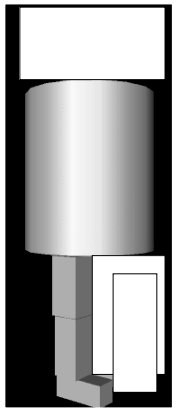




Figures 3-10 and 3-11 from *The Inventor Mentor*

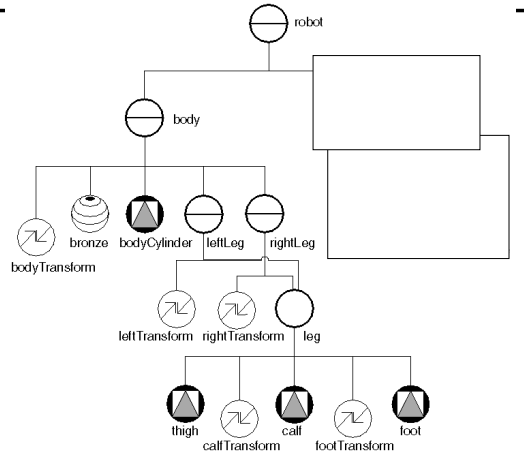
A Simple Scene Graph

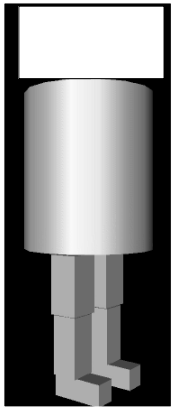




Figures 3-10 and 3-11 from *The Inventor Mentor*

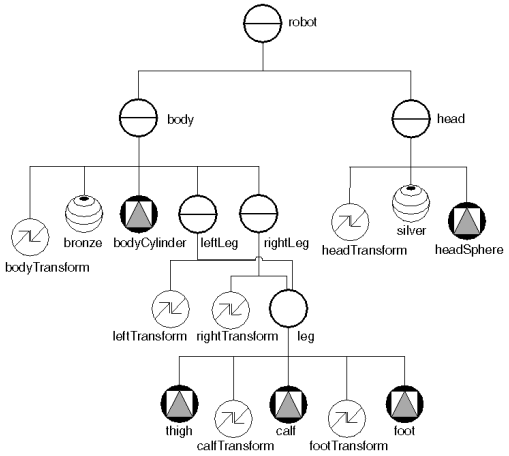
A Simple Scene Graph

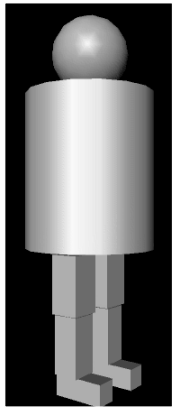




Figures 3-10 and 3-11 from *The Inventor Mentor*

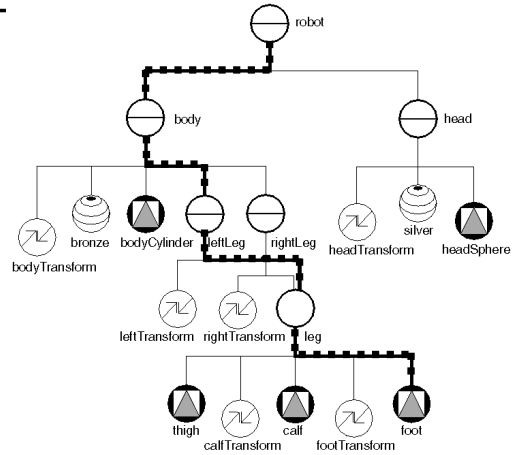
A Simple Scene Graph

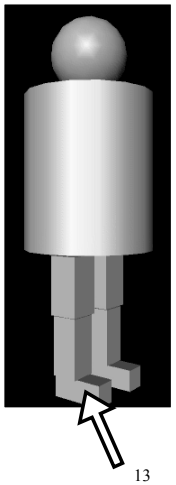




Figures 3-10 and 3-11 from *The Inventor Mentor*

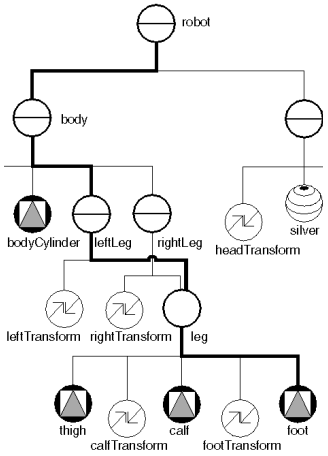
The Notion of a *Path*





Figures 3-12 and 3-11 from *The Inventor Mentor*

What Does a Path Give Us?



- Unique way to address each rendered item
- Way to ID all parts that might be important
  - What was picked?
  - What collided with another object?
  - Can walk up path: foot, left leg, whole robot…

Existing Scene Graphs

- Coin/OpenInventor
  - Early scene graph, closest to OpenGL
- OpenSceneGraph (OSG)
  - Modern scene graph, close to OpenGL
- Ogre3D
  - Modern scene graph with own innovations
  - OpenGL or DirectX, so not close to either
- X3D implementations (VRML successor)

Your Scene Graphs

- Assignment 4: your own simple scene graph
- Now, some base code…
  - Address system
  - GLUI command window
  - How they tie in with your shapes

Review

- Scene graph as an efficient, flexible way to render complex scenes
  - Hierarchy of nodes
  - Rendering: depth-first traversal
  - Path: unique ID for each rendered object
- List of scene graphs
- Base scene graph code

Sources

- Coin3D and general scene graph concepts:
  - Book: *The Inventor Mentor* (Addison-Wesley)
- Open Scene Graph:
  - Home: http://www.openscenegraph.com
  - Books available at http://www.skew-matrix.com/OSGdocs.html
- Ogre3D:
  - Home: http://www.ogre3d.org/
  - Book: *Pro OGRE 3D Programming* (Apress)