X3D Graphics for Web Authors

Chapter 3

Grouping Nodes

A Working Group is a technical committee that researches and proposes solutions to specific technical problems relating to X3D.

Web3D Consortium

Contents

Chapter Overview and Concepts

X3D Nodes and Examples

Additional Resources

Chapter Summary and Suggested Exercises

References

Chapter Overview

Overview: grouping nodes

Concepts: DEF/USE, units of measure, coordinate systems, right-hand rule, and bounding boxes

Grouping nodes organize objects in an X3D world

- Group, StaticGroup collect related nodes together
- Transform controls position, orientation and scale
- Inline loads other X3D scenes
- LOD (level of detail) provides different levels of geometry quality according to the user's viewpoint
- Switch can be animated to select different children, one (or none) at a time

Other grouping nodes are covered in Chapter 4

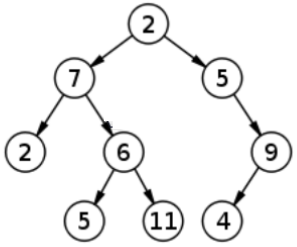
- Anchor, Billboard, Collision

Concepts

Tree terminology 1

Tree (graph theory) (data structures)

- a graph in which any two vertices are connected by exactly one simple path
- any connected graph without cycles is a tree

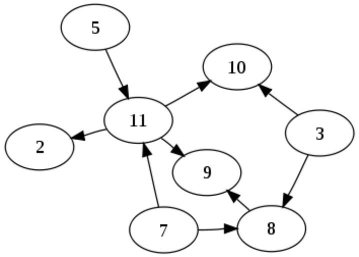


Node

- A node is a vertex in a graph
- Nodes are connected by edges or arcs

Directed acyclic graph (DAG)

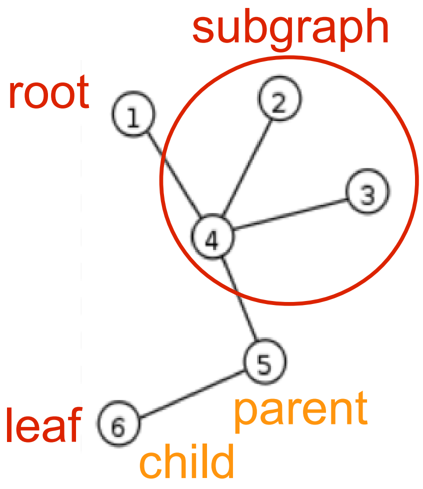
- Arcs connecting nodes have direction
- directed graph with no directed cycles



Tree terminology 2

Parent-child relationship, in a rooted tree

- The *parent* node of a given vertex is the vertex connected to it on the path to the root
- Every vertex except the root has a unique parent
- A *child* node of a vertex v is a vertex of which v is the parent
- A *leaf* node is a vertex without children



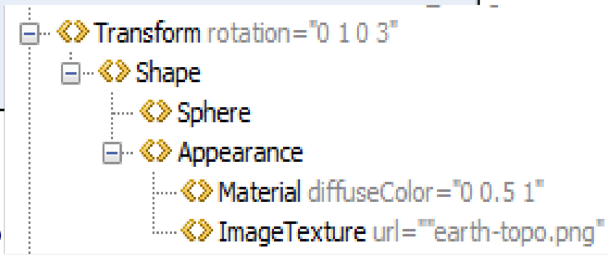
Subgraph is a subset of a graph; is also a graph

Intermediate (branching) or internal node

- A non-leaf vertex (nodes 4 or 5 in example)

XML and X3D correspondence

Opening element	`<Shape>`
Singleton element, attribute="value"	`<Sphere radius="10.0" solid="true"/>`
Opening element	`<Appearance>`
Singleton element, attribute='value'	`<ImageTexture url='earth-topo.png'/>`
Closing element	`</Appearance>`
Closing element	`</Shape>`



```
Transform rotation="0 1 0 3"
  Shape
    Sphere
    Appearance
      Material diffuseColor="0 0.5 1"
      ImageTexture url='"earth-topo.png"'
```

Elements correspond to X3D nodes

Attributes correspond to X3D simple-type fields

Parent-child relationships define *containerField*

Validatable XML using X3D DTD, schema

Grouping rationale

X3D scenes are directed acyclic graphs, made up of subgraphs with intermediate & leaf nodes

Grouping nodes help provide sensible structure

- Functionally related nodes collected together
- Grouping nodes can contain other grouping nodes, i.e. graphs of subgraphs
- Establish common or separate coordinate systems
- Make it easy to label nodes or subgraphs with DEF, then reference copies of those nodes (or grouped collections of nodes) with USE

DEF and USE

DEF names provide a label for any node

- Including child nodes making up that subgraph
- Equivalent to ID type in XML: must be unique within an X3D scene, no duplicate node labels
- Provides target for routing events
- Multiple DEFs: legal in X3D, illegal in XML, harmful

USE labels reference a DEF node

- Spelling is case sensitive, must be identical

DEF label must precede USE reference in scene

- Enables faster performance by single-pass loading
- Not detected by XML validation but still required

DEF naming

Names are important!

- Identifiers describe purpose and functionality
- Strongly influences how you think about a thing
- Provides explanatory documentation
- Must start with a letter, can't use hyphens

Naming convention: CamelCaseNaming

- capitalize each individual word
- avoid abbreviations, since none are consistent and they don't help international readers
- strive for clarity, be brief but complete

Units of measurement

Linear measurements in meters

- 1 m = 39.3”

Angular measurements in radians

- 2 pi = 360 degrees

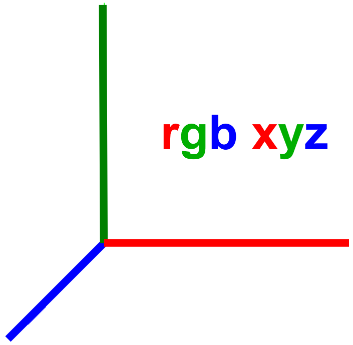
Time measured in seconds

- Starting 1 January 1970 at 00:00:00 midnight
- Sometimes referred to as “Unix time”

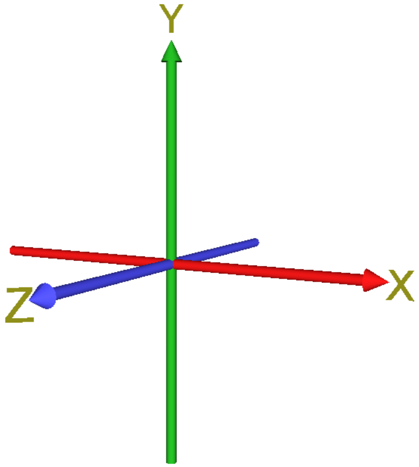
Colors

- RGB red-green-blue floating points ranging [0..1]
- Contrast with HTML use of integers [0..255]

Coordinate systems



Right hand rule for X Y Z order

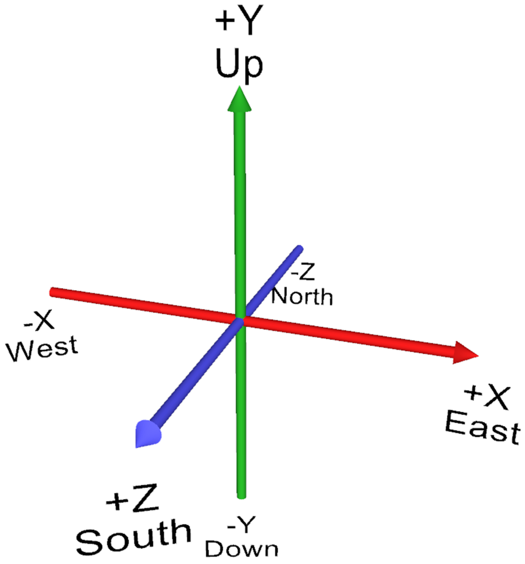


Y axis is up

Correspondence: East, Up, South

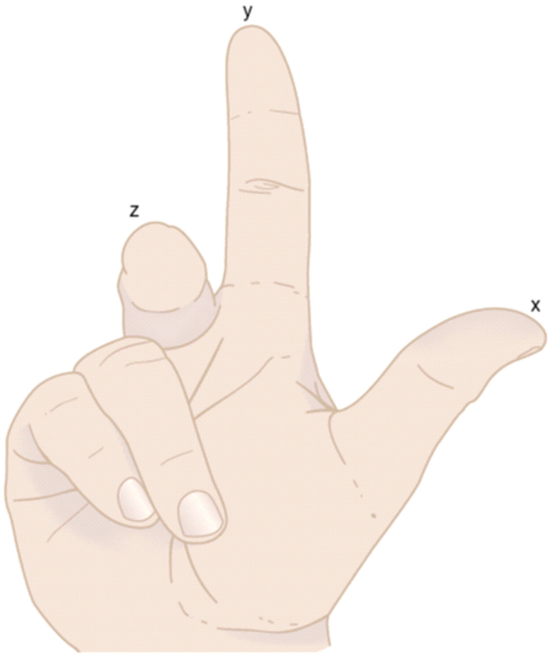
Accept no substitutes!

- or at least realign them ☺

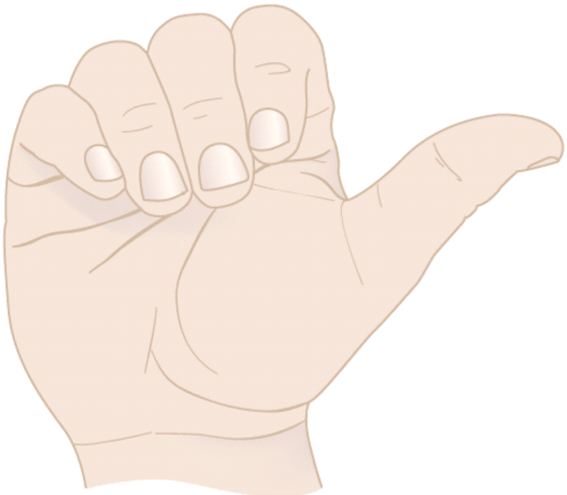


Right hand rules!

First three fingers of right hand must align with the X Y Z axes, in that order



Right hand rule also provides direction of positive rotation about an axis



Bounding boxes

Provides a hint to browsers about object size

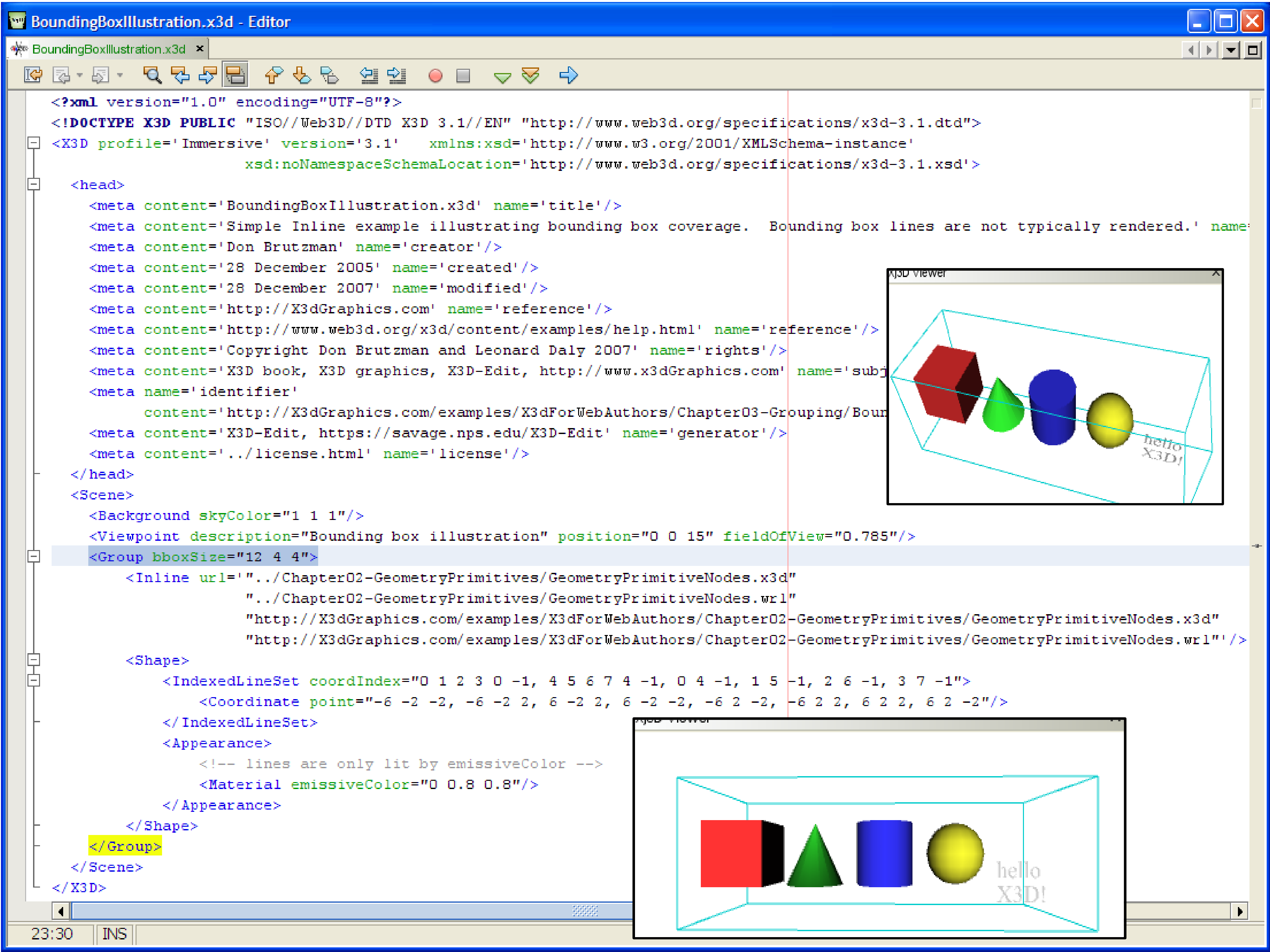
- Does not affect how an object is rendered (drawn) if it is actually larger than the bounding box
- Are never drawn themselves
- Defined by *bboxSize* and *bboxCenter*

Goal is to reduce computational complexity

- browser avoids calculating impossible collisions
- Size accumulates while proceeding up scene graph

Bounding boxes can be ignored by authors

- some authoring tools can provide them if needed



BoundingBoxIllustration.x3d - Editor
BoundingBoxIllustration.x3d
Xj3D Viewer
hello
X3D!
hello
X3D!
23:30
INS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1'   xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='BoundingBoxIllustration.x3d' name='title'/>
    <meta content='Simple Inline example illustrating bounding box coverage.  Bounding box lines are not typically rendered.' name=
    <meta content='Don Brutzman' name='creator'/>
    <meta content='28 December 2005' name='created'/>
    <meta content='28 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subj
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Boun
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor="1 1 1"/>
    <Viewpoint description="Bounding box illustration" position="0 0 15" fieldOfView="0.785"/>
    <Group bboxSize="12 4 4">
        <Inline url='"../Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d"
                     "../Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.wrl"
                     "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d"
                     "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.wrl"'/>
        <Shape>
            <IndexedLineSet coordIndex="0 1 2 3 0 -1, 4 5 6 7 4 -1, 0 4 -1, 1 5 -1, 2 6 -1, 3 7 -1">
                <Coordinate point="-6 -2 -2, -6 -2 2, 6 -2 2, 6 -2 -2, -6 2 -2, -6 2 2, 6 2 2, 6 2 -2"/>
            </IndexedLineSet>
            <Appearance>
                <!-- lines are only lit by emissiveColor -->
                <Material emissiveColor="0 0.8 0.8"/>
            </Appearance>
        </Shape>
    </Group>
  </Scene>
</X3D>
```

X3D Nodes and Examples

Group node

Collects nodes together with related purpose

- Often close to each other spatially

Can make USE copies if a DEF is provided

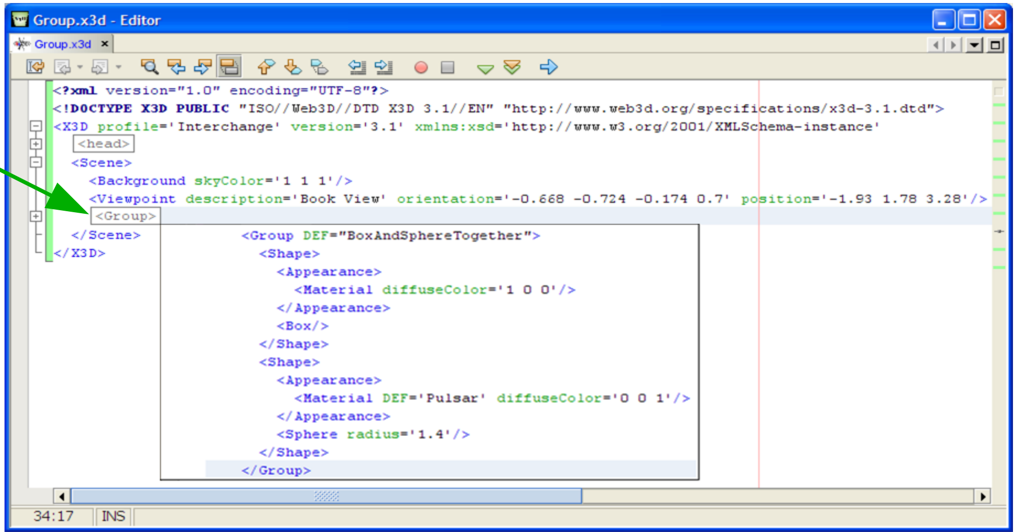
- Example: 4 identical tires in a car model

Simplify editing

- X3D-Edit: collapse node using +,- ticks in margin

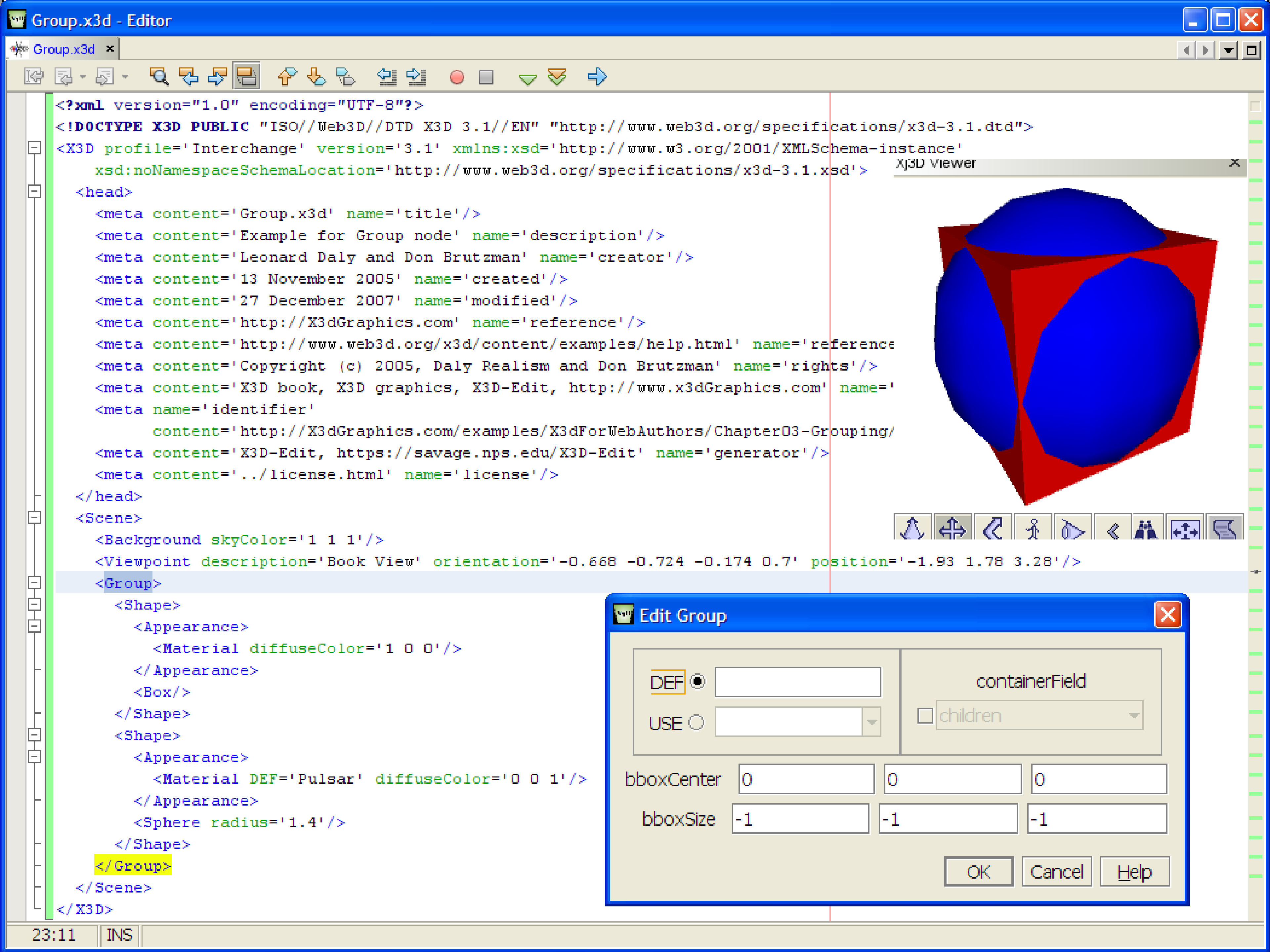
X3D-Edit trick

- Mouse-over to show hidden contents
- Helps to organize your work
- Copy by reference



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
  <head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <Group>
  </Scene>
</X3D>
```

```
<Group DEF="BoxAndSphereTogether">
  <Shape>
    <Appearance>
      <Material diffuseColor='1 0 0'/>
    </Appearance>
    <Box/>
  </Shape>
  <Shape>
    <Appearance>
      <Material DEF='Pulsar' diffuseColor='0 0 1'/>
    </Appearance>
    <Sphere radius='1.4'/>
  </Shape>
</Group>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='Group.x3d' name='title'/>
    <meta content='Example for Group node' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='13 November 2005' name='created'/>
    <meta content='27 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <Group>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
      <Shape>
        <Appearance>
          <Material DEF='Pulsar' diffuseColor='0 0 1'/>
        </Appearance>
        <Sphere radius='1.4'/>
      </Shape>
    </Group>
  </Scene>
</X3D>
```

Edit Group

DEF ●

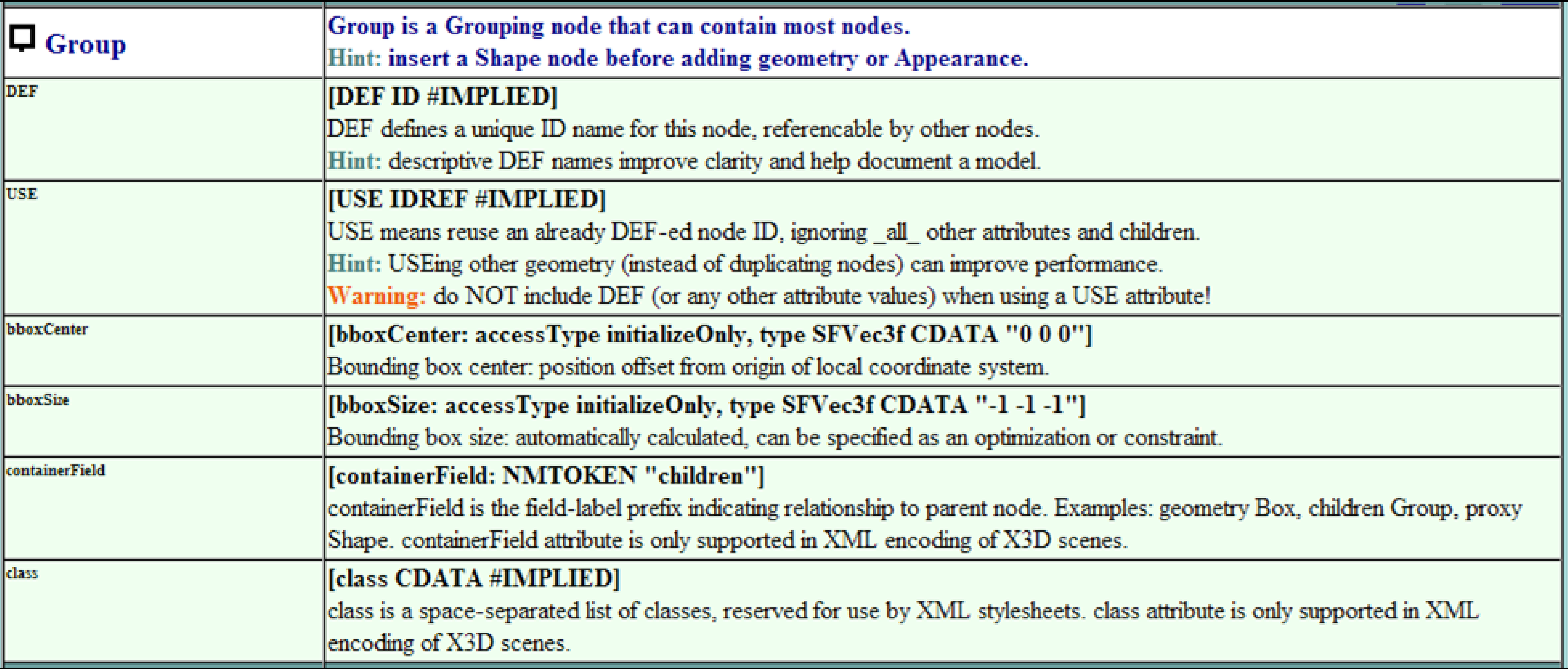
USE ○

containerField

☐ children

bboxCenter 0 0 0

bboxSize -1 -1 -1



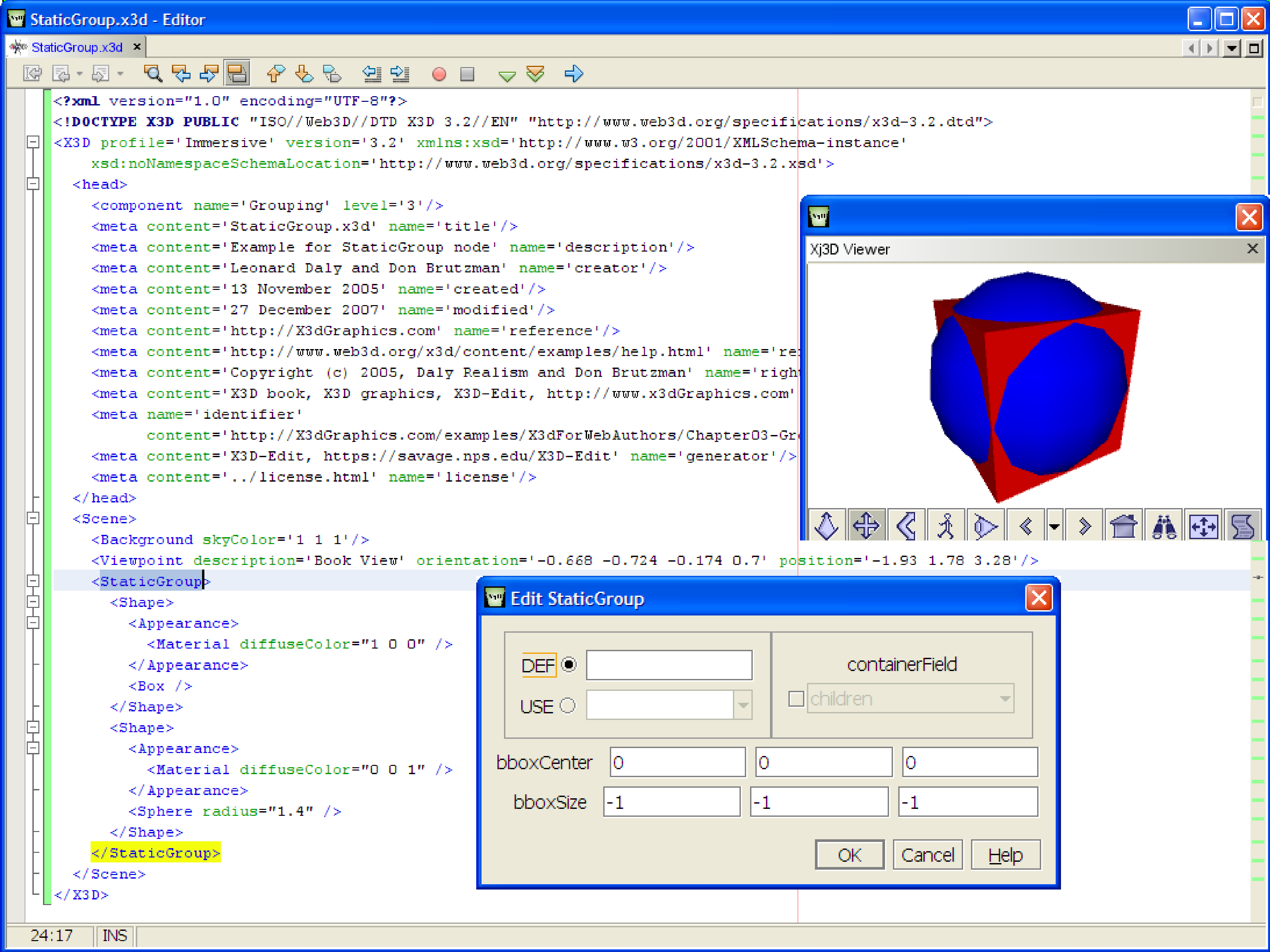
Group	**Group is a Grouping node that can contain most nodes.** **Hint: insert a Shape node before adding geometry or Appearance.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

StaticGroup node

Identical to Group, except that children are not allowed to change or be animated

Rarely used

Allows X3D browsers to simplify underlying data representations and optimize performance, if possible



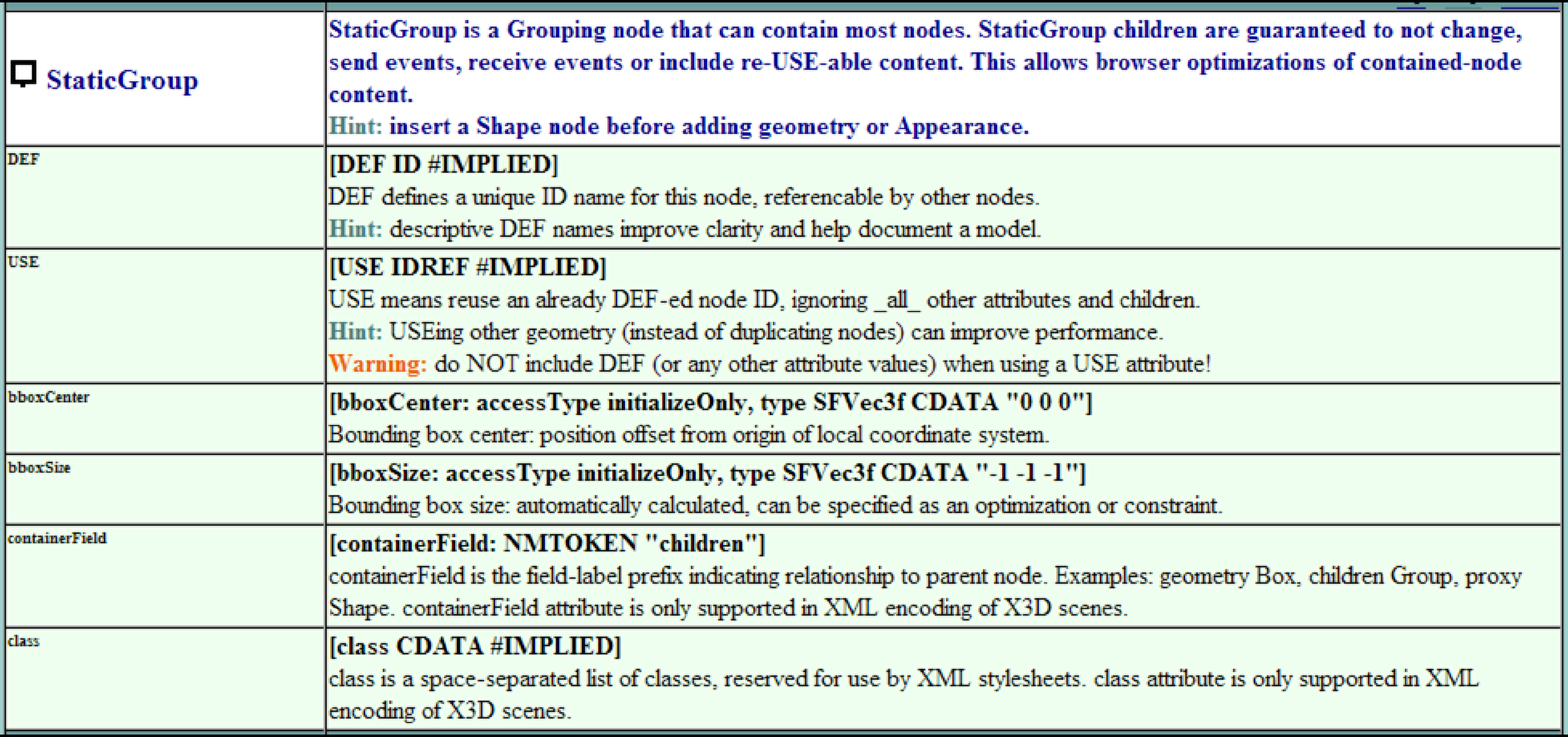
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.2.xsd'>
  <head>
    <component name='Grouping' level='3'/>
    <meta content='StaticGroup.x3d' name='title'/>
    <meta content='Example for StaticGroup node' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='13 November 2005' name='created'/>
    <meta content='27 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='re
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='righ
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com'
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Gr
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <StaticGroup>
      <Shape>
        <Appearance>
          <Material diffuseColor="1 0 0" />
        </Appearance>
        <Box />
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor="0 0 1" />
        </Appearance>
        <Sphere radius="1.4" />
      </Shape>
    </StaticGroup>
  </Scene>
</X3D>
```

Edit StaticGroup

DEF ● [ ]
USE ○ [ ]

containerField
☐ children

bboxCenter: 0 | 0 | 0
bboxSize: -1 | -1 | -1



StaticGroup	StaticGroup is a Grouping node that can contain most nodes. StaticGroup children are guaranteed to not change, send events, receive events or include re-USE-able content. This allows browser optimizations of contained-node content. Hint: insert a Shape node before adding geometry or Appearance.
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

Transform node

Grouping node that defines a coordinate system for its children

Root of X3D scene graph is always at (0 0 0)

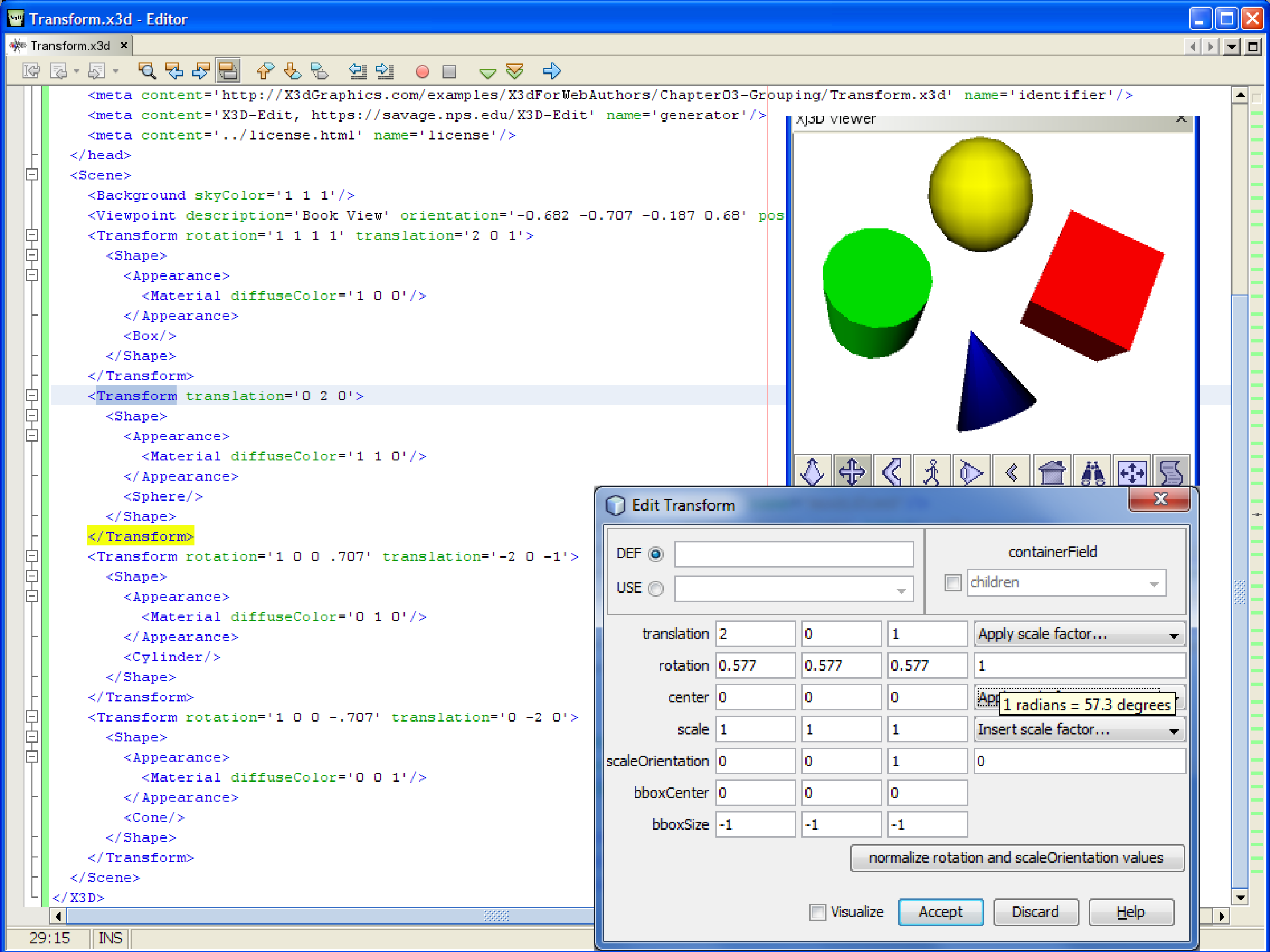
Transform nodes can

- Translate local origin linearly to another coordinate
- Rotate about any axis
- Scale size, uniformly or separately along x y z axes

Group and Transform are among most commonly used nodes

Transform fields

- *translation:* x y z movement in meters from origin of local coordinate system
- *rotation:* [axis x y z]-angle rotation about origin of local coordinate system, using units of radians
- *scale:* x y z (potentially nonuniform) factor for change in object scale to make it larger or smaller
- *center:* origin offset prior to applying rotation
- *scaleOrientation:* [axis x y z]-angle rotation to apply prior to scaling
- *bboxCenter, bboxSize:* x y z bounding box information (if any is provided by author, optional)



Transform.x3d - Editor

Transform.x3d

```
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Transform.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.682 -0.707 -0.187 0.68' pos
    <Transform rotation='1 1 1 1' translation='2 0 1'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
    </Transform>
    <Transform translation='0 2 0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 1 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
    <Transform rotation='1 0 0 .707' translation='-2 0 -1'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 1 0'/>
        </Appearance>
        <Cylinder/>
      </Shape>
    </Transform>
    <Transform rotation='1 0 0 -.707' translation='0 -2 0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 0 1'/>
        </Appearance>
        <Cone/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Edit Transform

DEF ●

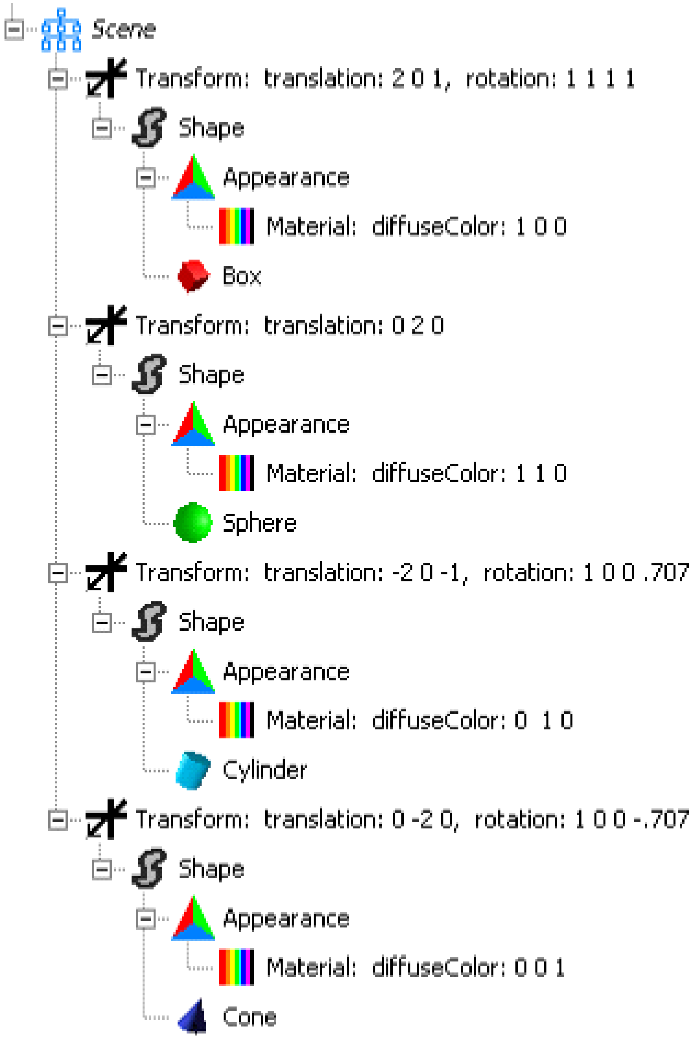
USE ○

containerField □ children

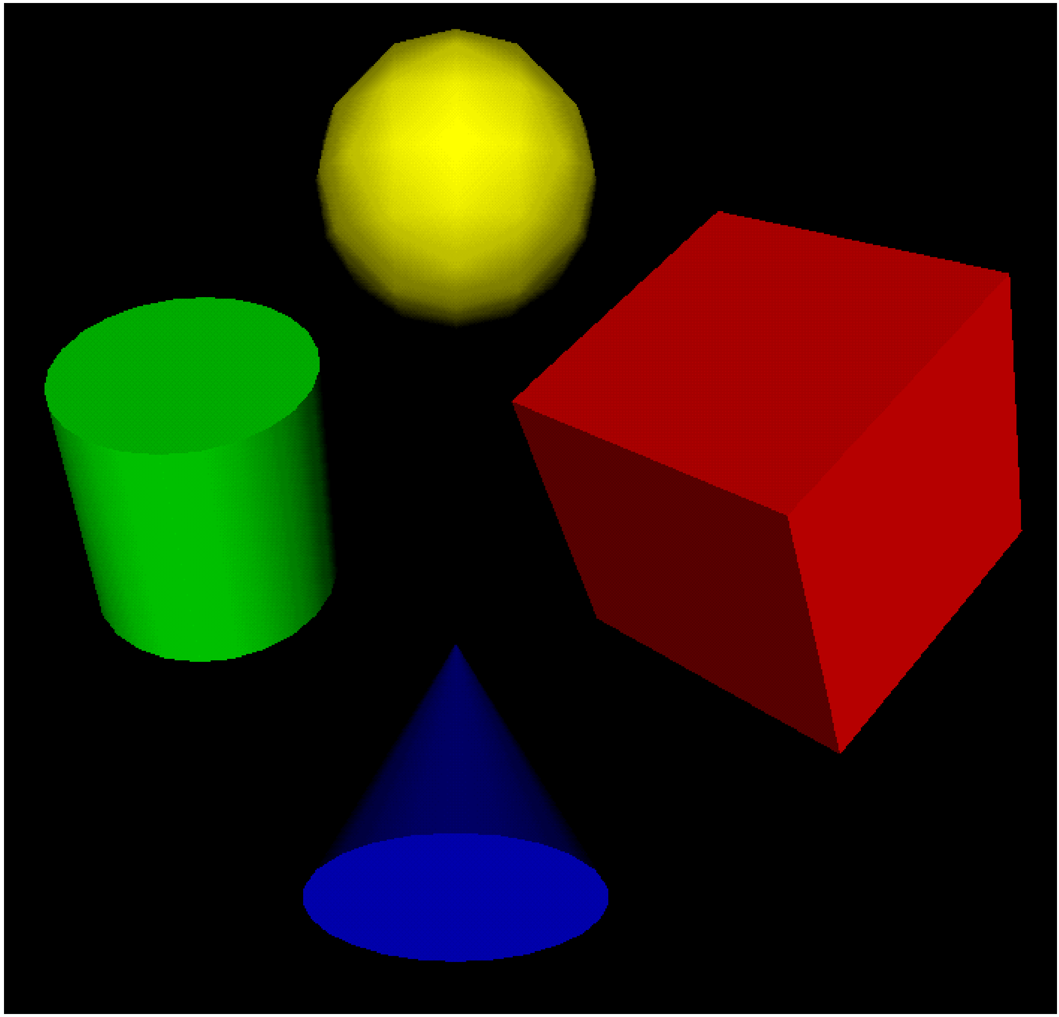
translation	2	0	1	Apply scale factor...
rotation	0.577	0.577	0.577	1
center	0	0	0	App
scale	1	1	1	Insert scale factor...
scaleOrientation	0	0	1	0
bboxCenter	0	0	0	
bboxSize	-1	-1	-1	

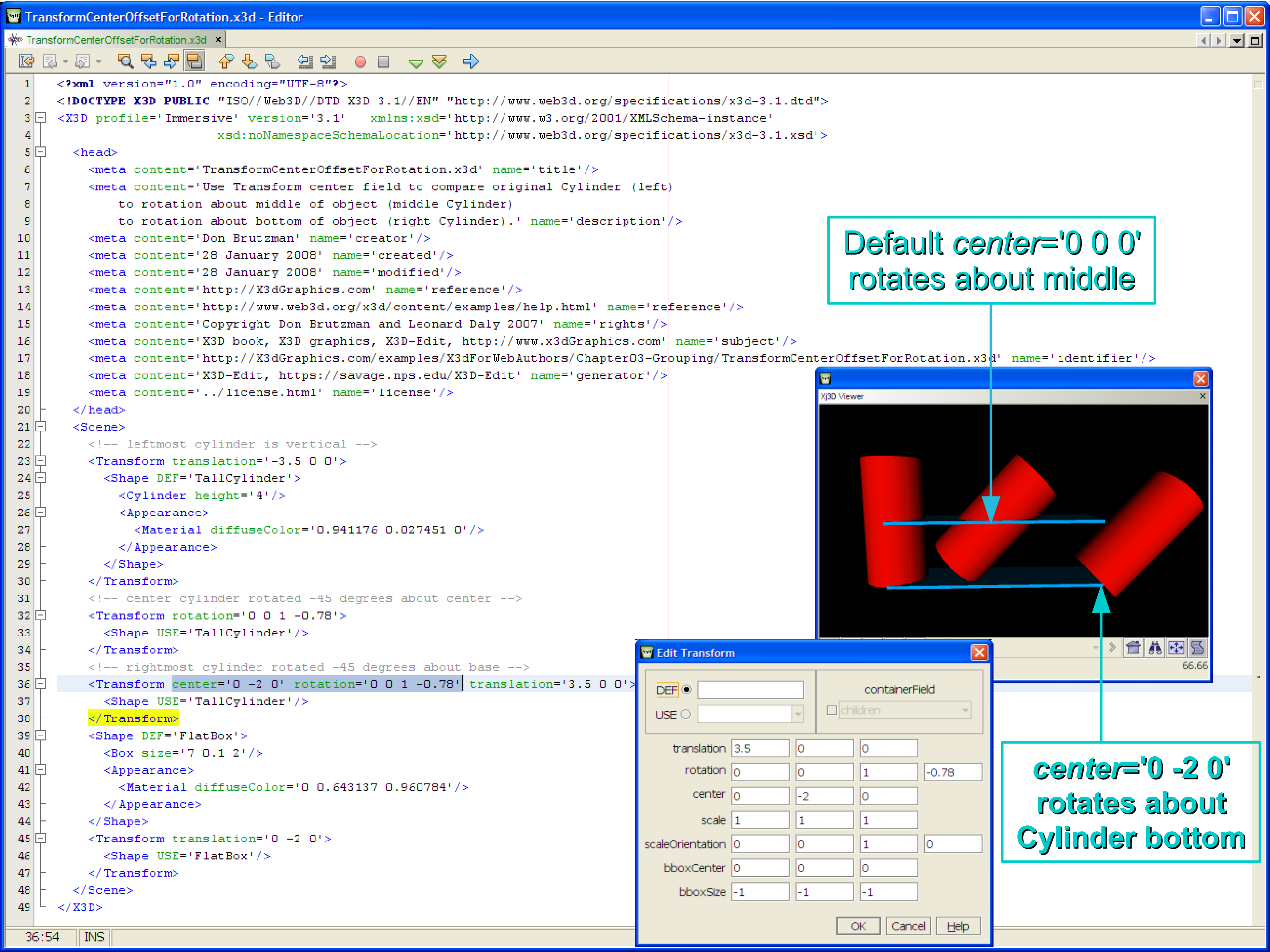
1 radians = 57.3 degrees

Each Transform is a scene subgraph



- Scene
  - Transform: translation: 2 0 1, rotation: 1 1 1 1
    - Shape
      - Appearance
        - Material: diffuseColor: 1 0 0
      - Box
  - Transform: translation: 0 2 0
    - Shape
      - Appearance
        - Material: diffuseColor: 1 1 0
      - Sphere
  - Transform: translation: -2 0 -1, rotation: 1 0 0 .707
    - Shape
      - Appearance
        - Material: diffuseColor: 0 1 0
      - Cylinder
  - Transform: translation: 0 -2 0, rotation: 1 0 0 -.707
    - Shape
      - Appearance
        - Material: diffuseColor: 0 0 1
      - Cone





TransformCenterOffsetForRotation.x3d - Editor
TransformCenterOffsetForRotation.x3d
Xj3D Viewer
66.66
36:54
INS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1'   xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='TransformCenterOffsetForRotation.x3d' name='title'/>
    <meta content='Use Transform center field to compare original Cylinder (left)
        to rotation about middle of object (middle Cylinder)
        to rotation about bottom of object (right Cylinder).' name='description'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='28 January 2008' name='created'/>
    <meta content='28 January 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/TransformCenterOffsetForRotation.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <!-- leftmost cylinder is vertical -->
    <Transform translation='-3.5 0 0'>
      <Shape DEF='TallCylinder'>
        <Cylinder height='4'/>
        <Appearance>
          <Material diffuseColor='0.941176 0.027451 0'/>
        </Appearance>
      </Shape>
    </Transform>
    <!-- center cylinder rotated -45 degrees about center -->
    <Transform rotation='0 0 1 -0.78'>
      <Shape USE='TallCylinder'/>
    </Transform>
    <!-- rightmost cylinder rotated -45 degrees about base -->
    <Transform center='0 -2 0' rotation='0 0 1 -0.78' translation='3.5 0 0'>
      <Shape USE='TallCylinder'/>
    </Transform>
    <Shape DEF='FlatBox'>
      <Box size='7 0.1 2'/>
      <Appearance>
        <Material diffuseColor='0 0.643137 0.960784'/>
      </Appearance>
    </Shape>
    <Transform translation='0 -2 0'>
      <Shape USE='FlatBox'/>
    </Transform>
  </Scene>
</X3D>
```

Default *center*='0 0 0' rotates about middle

Edit Transform

DEF ●
USE ○
containerField ☐ children

translation	3.5	0	0	
rotation	0	0	1	-0.78
center	0	-2	0	
scale	1	1	1	
scaleOrientation	0	0	1	0
bboxCenter	0	0	0	
bboxSize	-1	-1	-1	

OK Cancel Help

center='0 -2 0' rotates about Cylinder bottom

Complex rotations

When in doubt, nest multiple Transform nodes

- Substitute roll, pitch, yaw values in order, e.g.

```
<Transform rotation='0 1 0 yaw'>
    <Transform rotation='0 0 1 pitch'>
        <Transform rotation='1 0 0 roll'>
            <!-- Shape (pointing along X axis, Y up) goes here -->
        </Transform>
    </Transform>
</Transform>
```

This approach helps ensure correctness & clarity

- Without reducing computational performance of viewer rendering, since recalculations are avoided whenever no intermediate changes occur in the composite transformation matrix

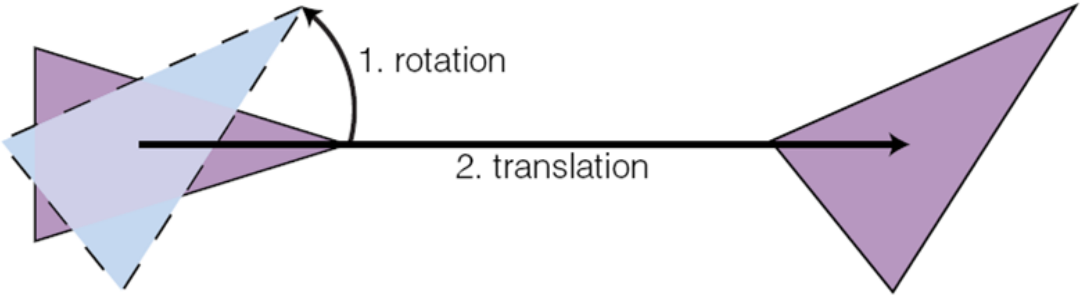
Order of transformation operations

The ordering of transformation operations is important and not symmetric. Algorithm:

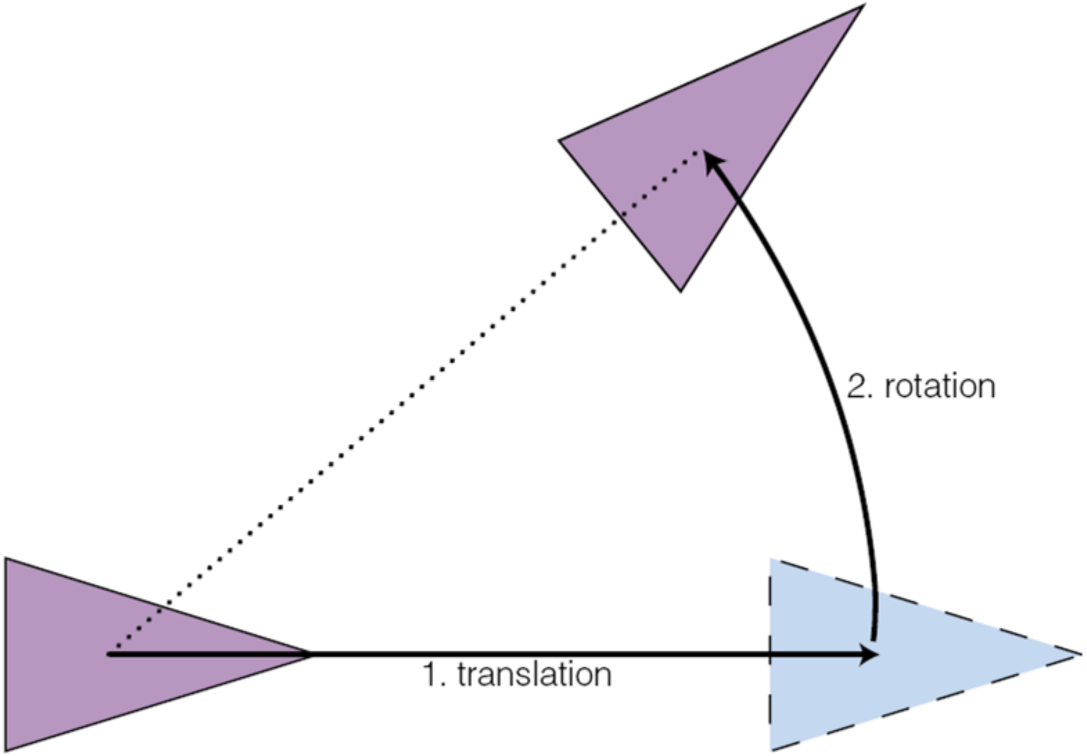
- Apply reverse *center* offset to set up for properly centered scaling and orientation operations
- Apply reverse *scaleOrientation*, then apply *scale* operation, then apply forward *scaleOrientation* to regain initial frame
- Apply *rotation* to final direction, then apply forward *center* offset to regain initial origin
- Apply *translation* to final location of new coordinate frame

Comparing out-of-order operations

Case 1



Case 2



Equivalent transformations

```
Transform {
        center C
        rotation R
        scale S
        scaleOrientation SR
        translation T
        children [...]
}
```

```
Transform {
        translation T
        children Transform {
                translation C
                children Transform {
                        rotation R
                        children Transform {
                                rotation SR
                                children Transform {
                                     scale S
                                     children Transform {
                                          rotation -SR
                                          children Transform {
                                               translation -C
                                               children [...]
                                          }
                                     }
                                }
                        }
                }
        }
}
```

Using matrix transformation notation, where

- **C** (center),
- **SR** (scaleOrientation),
- **T** (translation),
- **R** (rotation), and
- **S** (scale)

are the equivalent transformation matrices, then

- **P'** is transformed child point **P**
- **P' = T · C · R · SR · S · −SR · −C · P**

Matrix operations

Matrix operations are not directly exposed in X3D

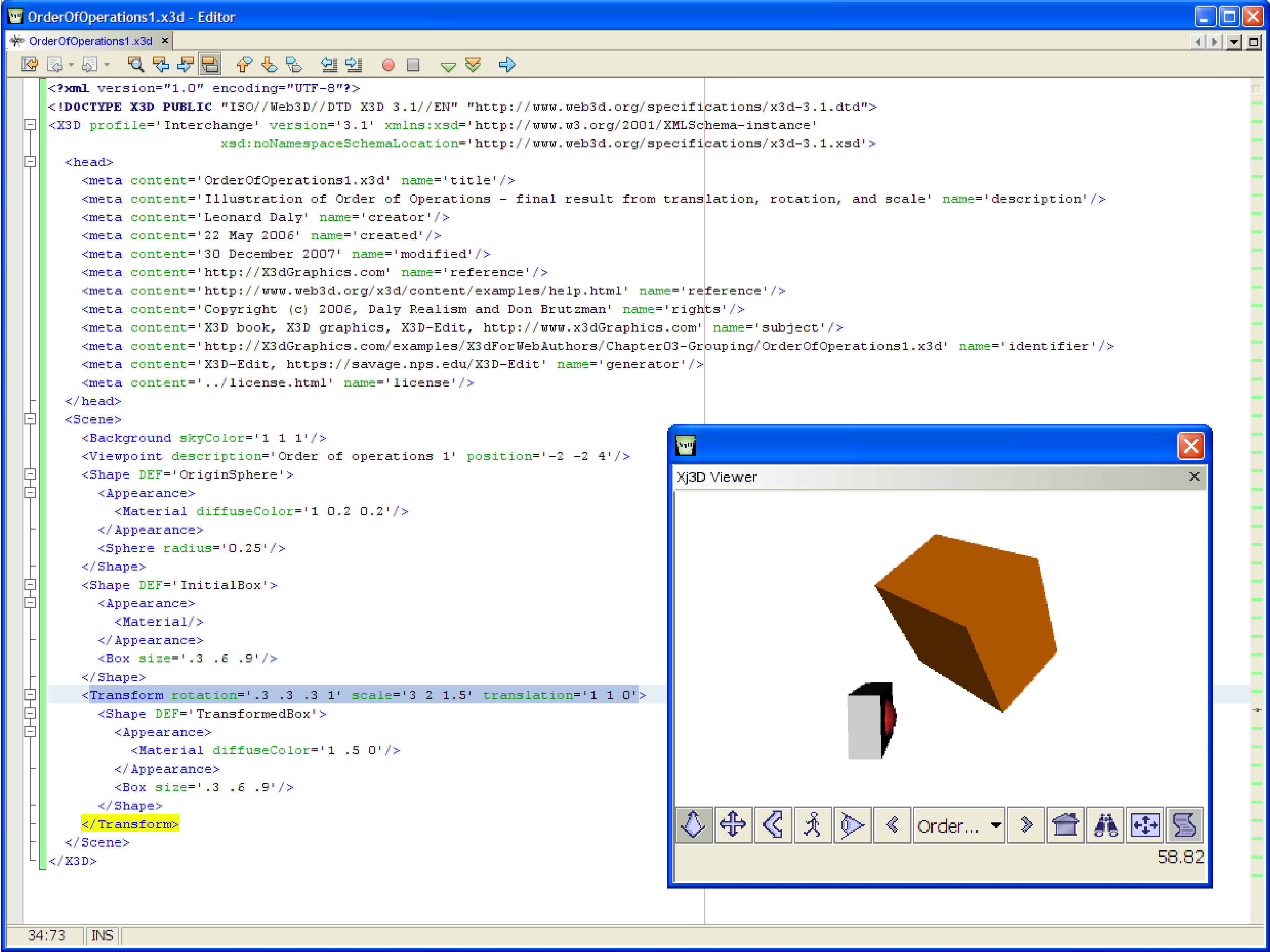
- Unlike most imperative programming interfaces which use 4x4 transformation matrix operations
- Instead Transform nodes provide a regularized way to perform translation, rotation, scaling

Transform includes a specific order of operations

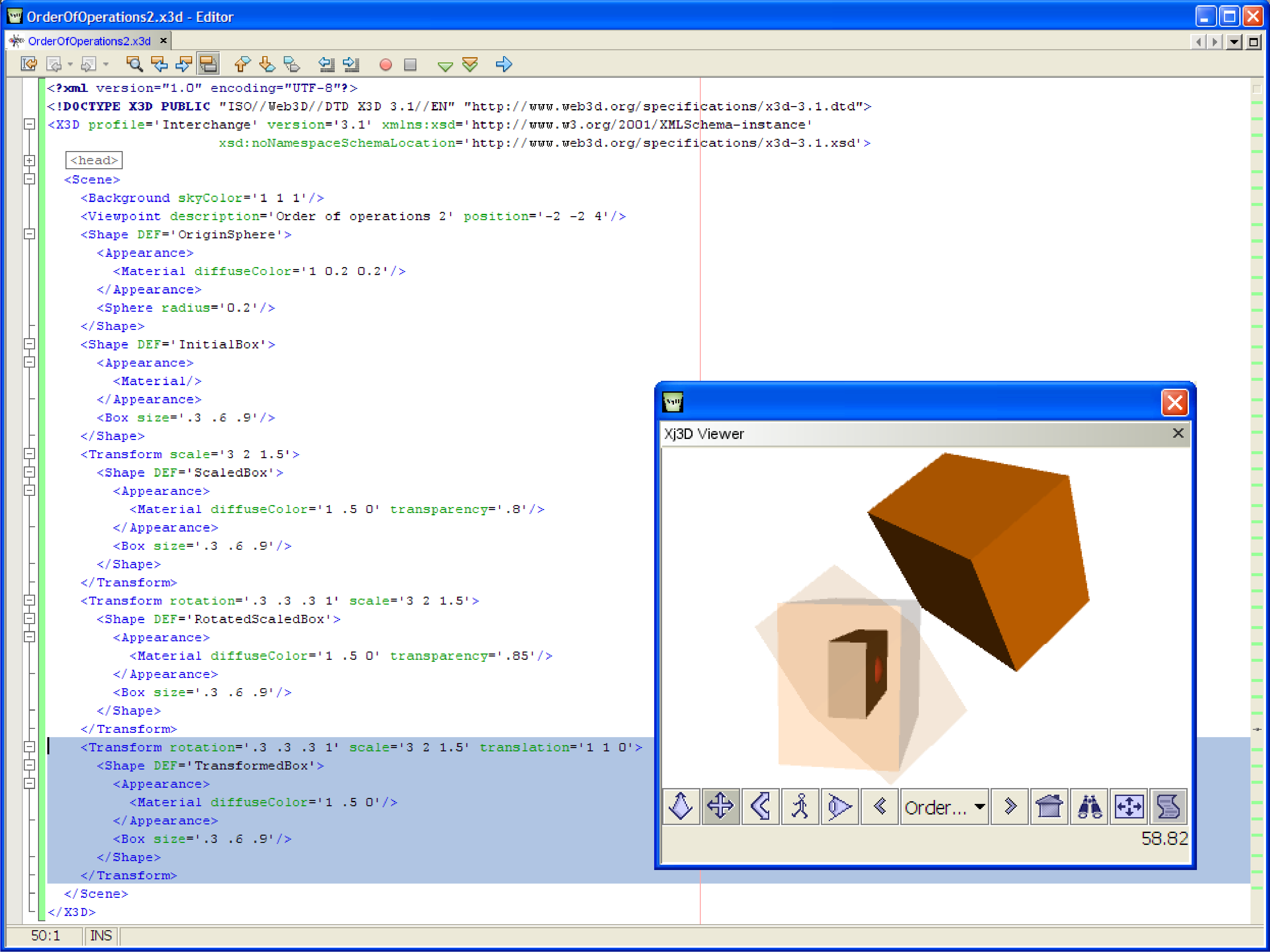
- Illustrated in next slides

Flexible: multiple Transform nodes can be nested

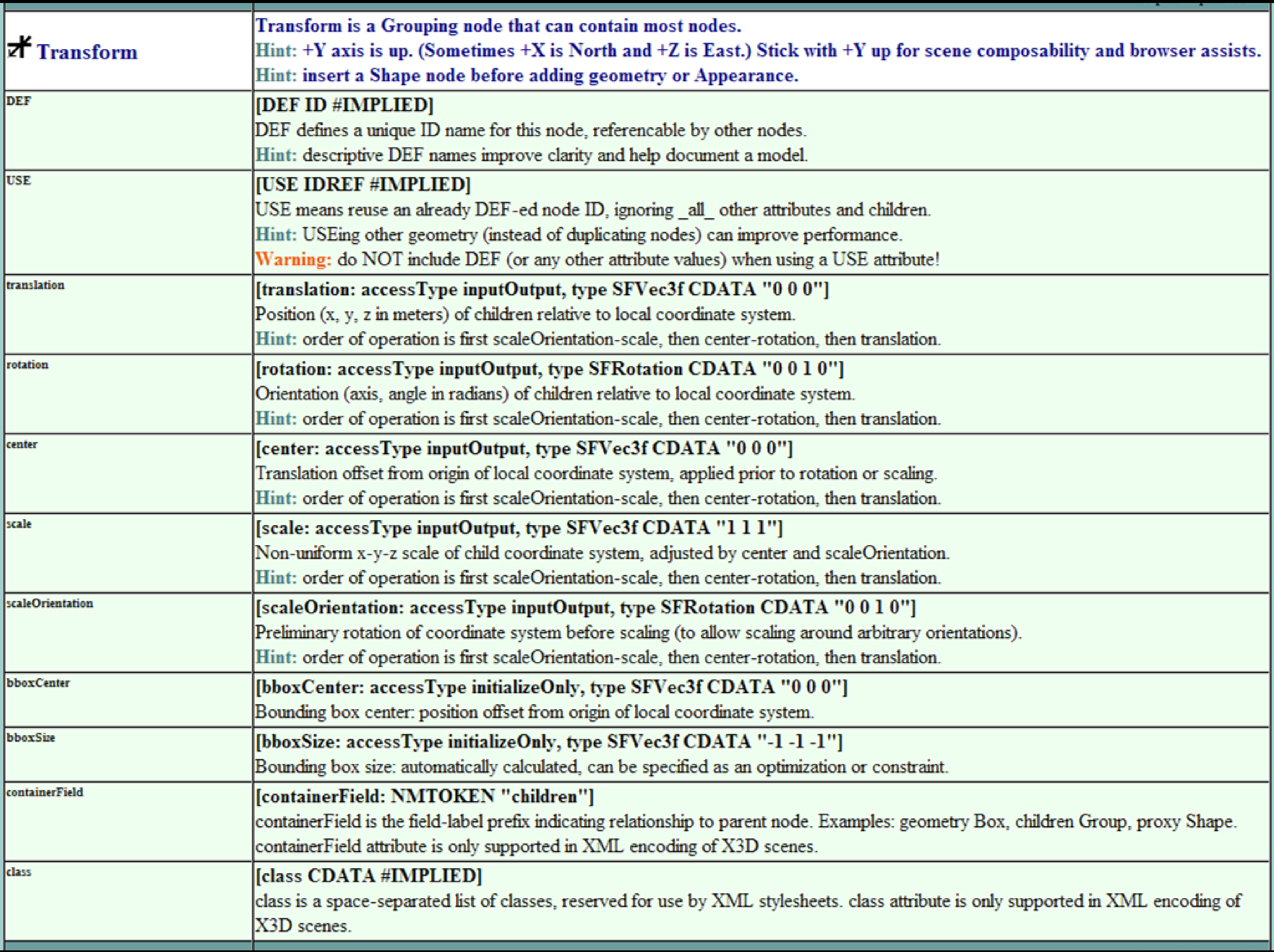
- Each Transform establishes new coordinate frame



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='OrderOfOperations1.x3d' name='title'/>
    <meta content='Illustration of Order of Operations - final result from translation, rotation, and scale' name='description'/>
    <meta content='Leonard Daly' name='creator'/>
    <meta content='22 May 2006' name='created'/>
    <meta content='30 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright (c) 2006, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/OrderOfOperations1.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Order of operations 1' position='-2 -2 4'/>
    <Shape DEF='OriginSphere'>
      <Appearance>
        <Material diffuseColor='1 0.2 0.2'/>
      </Appearance>
      <Sphere radius='0.25'/>
    </Shape>
    <Shape DEF='InitialBox'>
      <Appearance>
        <Material/>
      </Appearance>
      <Box size='.3 .6 .9'/>
    </Shape>
    <Transform rotation='.3 .3 .3 1' scale='3 2 1.5' translation='1 1 0'>
      <Shape DEF='TransformedBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Order of operations 2' position='-2 -2 4'/>
    <Shape DEF='OriginSphere'>
      <Appearance>
        <Material diffuseColor='1 0.2 0.2'/>
      </Appearance>
      <Sphere radius='0.2'/>
    </Shape>
    <Shape DEF='InitialBox'>
      <Appearance>
        <Material/>
      </Appearance>
      <Box size='.3 .6 .9'/>
    </Shape>
    <Transform scale='3 2 1.5'>
      <Shape DEF='ScaledBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0' transparency='.8'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
    <Transform rotation='.3 .3 .3 1' scale='3 2 1.5'>
      <Shape DEF='RotatedScaledBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0' transparency='.85'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
    <Transform rotation='.3 .3 .3 1' scale='3 2 1.5' translation='1 1 0'>
      <Shape DEF='TransformedBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```



Transform	**Transform is a Grouping node that can contain most nodes.** **Hint: +Y axis is up. (Sometimes +X is North and +Z is East.) Stick with +Y up for scene composability and browser assists.** **Hint: insert a Shape node before adding geometry or Appearance.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
translation	**[translation: accessType inputOutput, type SFVec3f CDATA "0 0 0"]** Position (x, y, z in meters) of children relative to local coordinate system. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
rotation	**[rotation: accessType inputOutput, type SFRotation CDATA "0 0 1 0"]** Orientation (axis, angle in radians) of children relative to local coordinate system. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
center	**[center: accessType inputOutput, type SFVec3f CDATA "0 0 0"]** Translation offset from origin of local coordinate system, applied prior to rotation or scaling. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
scale	**[scale: accessType inputOutput, type SFVec3f CDATA "1 1 1"]** Non-uniform x-y-z scale of child coordinate system, adjusted by center and scaleOrientation. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
scaleOrientation	**[scaleOrientation: accessType inputOutput, type SFRotation CDATA "0 0 1 0"]** Preliminary rotation of coordinate system before scaling (to allow scaling around arbitrary orientations). Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

Inline node

Loads another X3D world within current scene

- Supported formats depend on user's X3D browser: XML encoding .x3d, ClassicVRML encoding .x3dv, compressed binary .x3db, possibly VRML97 .wrl
- Load another world into your scene, or vice versa

Inline scene is positioned, rotated and scaled to match the local coordinate frame

- Local reference frame determined by parent Transformation node hierarchy
- User's viewpoint does not change automatically to the loaded Inline scene's default Viewpoint

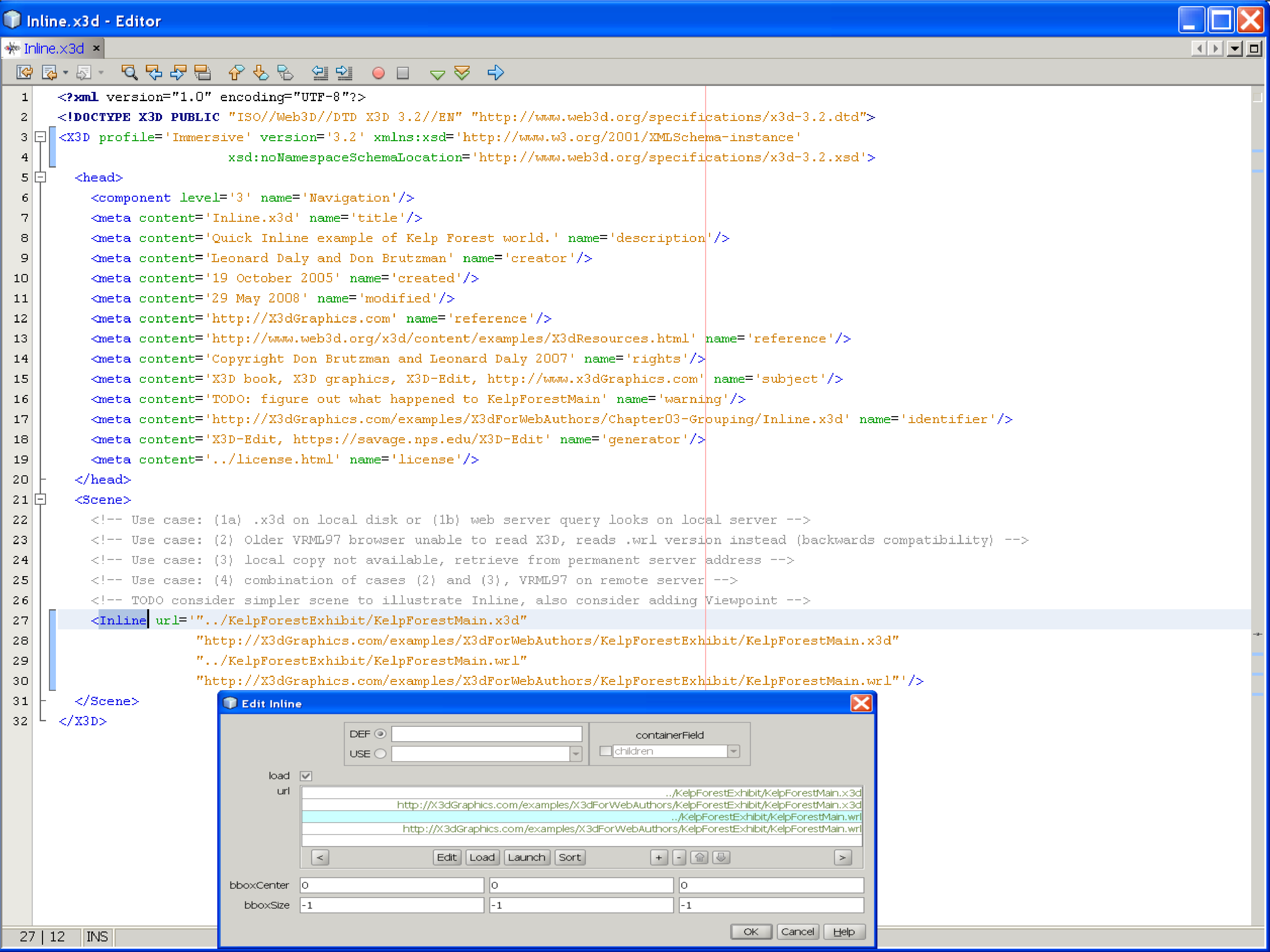
url field

url = uniform resource locator

- Equivalent to universal resource identifier (uri)

url field is a “quoted” string array that can hold multiple equivalent addresses

- Each address should point to same resource
- Each address is retrieved and evaluated, in order, until the desired Inline file is successfully retrieved
- Relative addresses can work on localhost or server
- Absolute addresses provide reliable backup
- Interesting variations possible

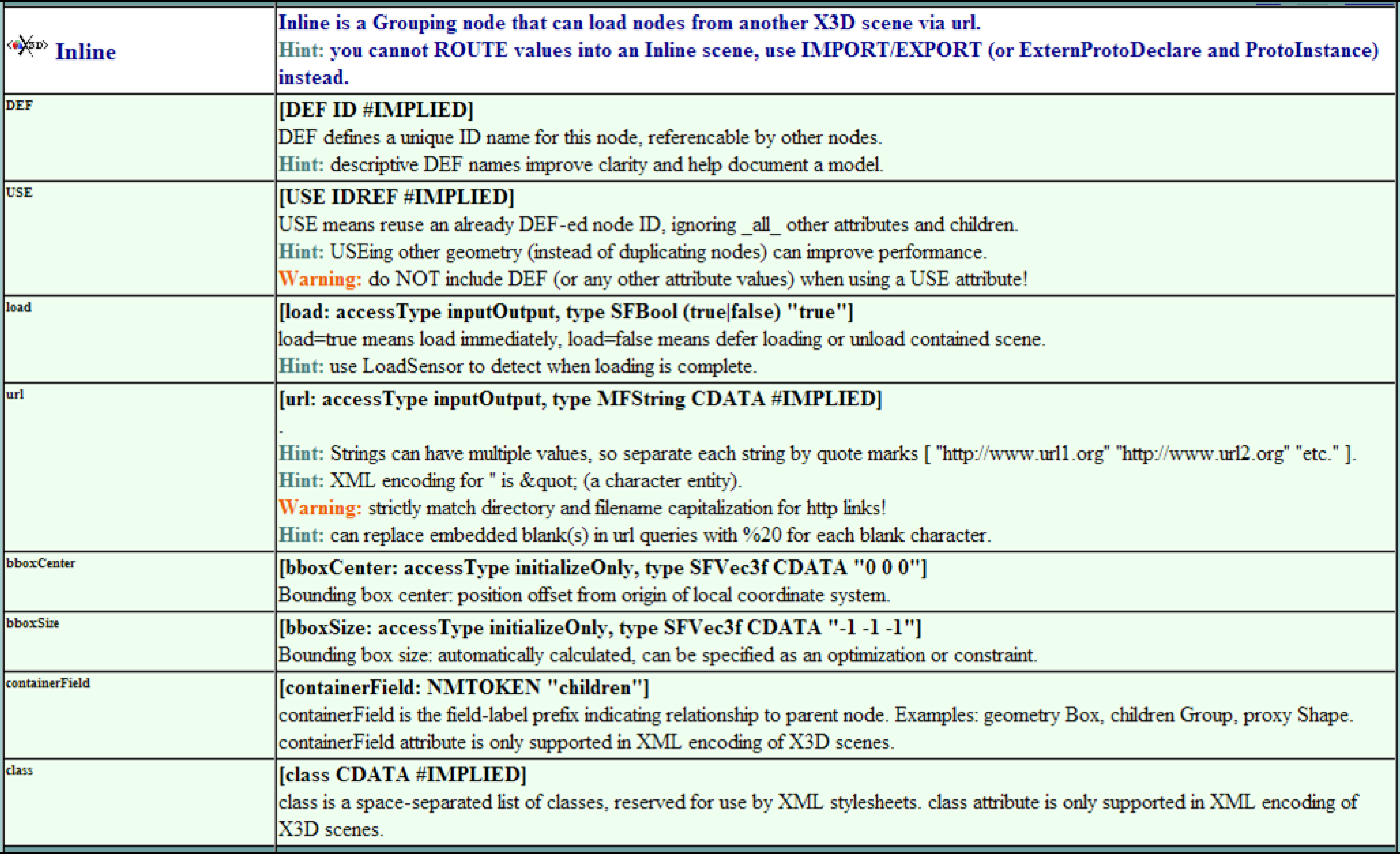


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.2.xsd'>
  <head>
    <component level='3' name='Navigation'/>
    <meta content='Inline.x3d' name='title'/>
    <meta content='Quick Inline example of Kelp Forest world.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='19 October 2005' name='created'/>
    <meta content='29 May 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/X3dResources.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='TODO: figure out what happened to KelpForestMain' name='warning'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Inline.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <!-- Use case: (1a) .x3d on local disk or (1b) web server query looks on local server -->
    <!-- Use case: (2) Older VRML97 browser unable to read X3D, reads .wrl version instead (backwards compatibility) -->
    <!-- Use case: (3) local copy not available, retrieve from permanent server address -->
    <!-- Use case: (4) combination of cases (2) and (3), VRML97 on remote server -->
    <!-- TODO consider simpler scene to illustrate Inline, also consider adding Viewpoint -->
    <Inline url='"../KelpForestExhibit/KelpForestMain.x3d"
             "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d"
             "../KelpForestExhibit/KelpForestMain.wrl"
             "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.wrl"'/>
  </Scene>
</X3D>
```

27 | 12 INS

DEF ●
USE ○
containerField ☐ children
load ☑
url
../KelpForestExhibit/KelpForestMain.x3d
http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d
../KelpForestExhibit/KelpForestMain.wrl
http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.wrl

bboxCenter 0 0 0
bboxSize -1 -1 -1



Inline	**Inline is a Grouping node that can load nodes from another X3D scene via url.** **Hint: you cannot ROUTE values into an Inline scene, use IMPORT/EXPORT (or ExternProtoDeclare and ProtoInstance) instead.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
load	**[load: accessType inputOutput, type SFBool (true|false) "true"]** load=true means load immediately, load=false means defer loading or unload contained scene. Hint: use LoadSensor to detect when loading is complete.
url	**[url: accessType inputOutput, type MFString CDATA #IMPLIED]** . Hint: Strings can have multiple values, so separate each string by quote marks ["http://www.url1.org" "http://www.url2.org" "etc."]. Hint: XML encoding for " is &quot; (a character entity). Warning: strictly match directory and filename capitalization for http links! Hint: can replace embedded blank(s) in url queries with %20 for each blank character.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

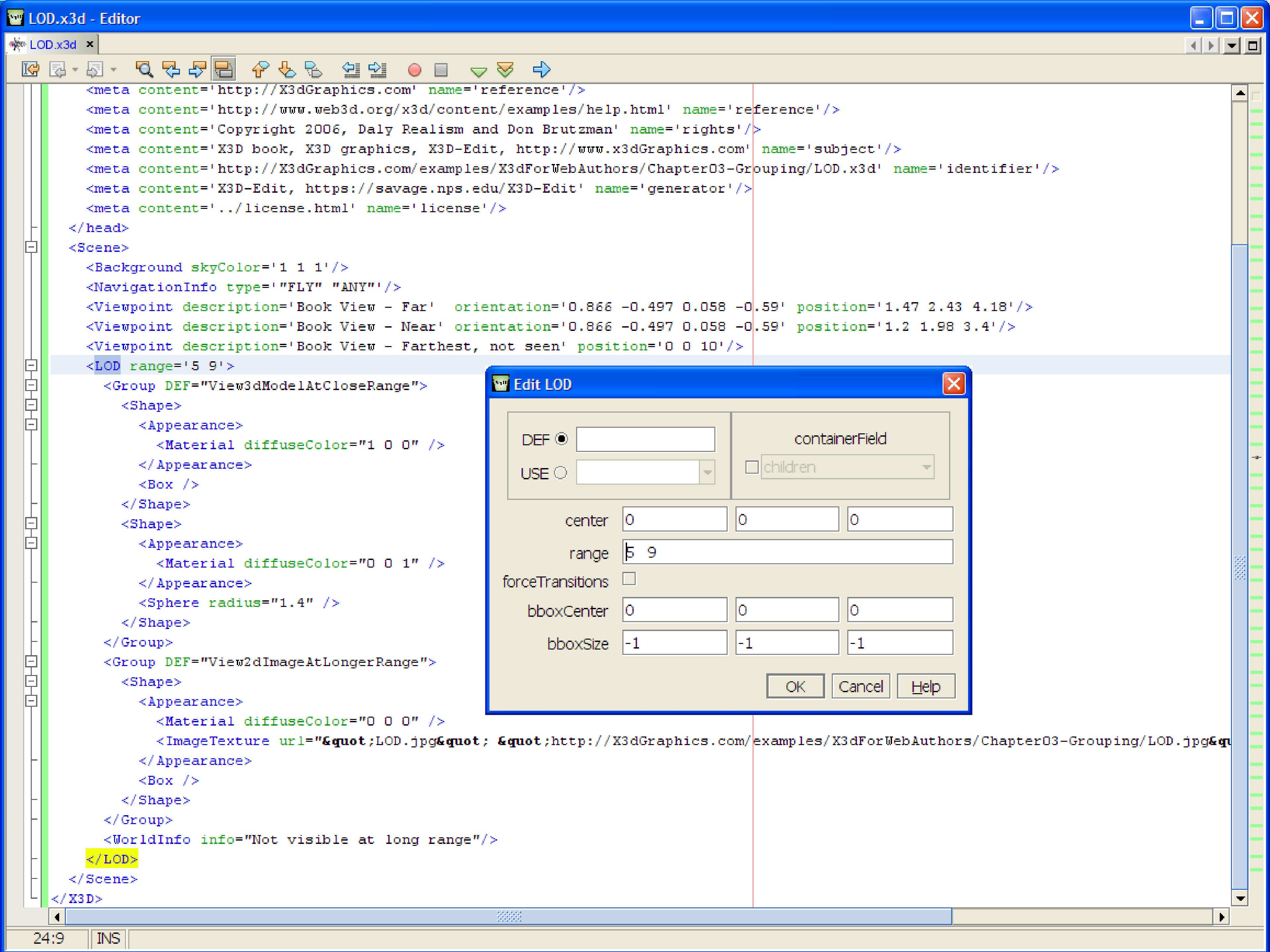
LOD (level of detail) node

LOD holds multiple versions of same geometry

- Also defines array of range values corresponding to transition distance between each version
- Selection of appropriate LOD child is based on distance to user's current view position

LOD improves scene performance by reducing polygonal complexity

- Use high-fidelity geometry at close range, and progressively simpler geometry when farther off
- Range values are hint unless *forceTransitions*='true'

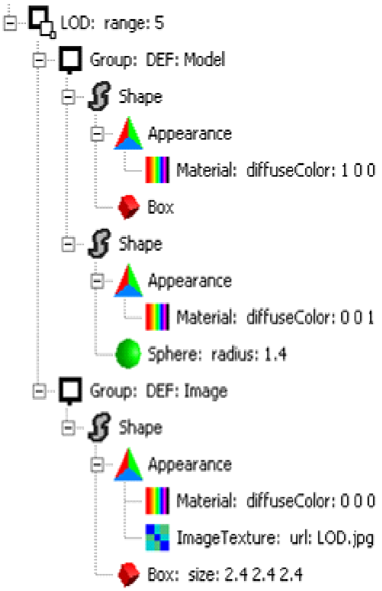


```
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo type='"FLY" "ANY"'/>
    <Viewpoint description='Book View - Far'  orientation='0.866 -0.497 0.058 -0.59' position='1.47 2.43 4.18'/>
    <Viewpoint description='Book View - Near' orientation='0.866 -0.497 0.058 -0.59' position='1.2 1.98 3.4'/>
    <Viewpoint description='Book View - Farthest, not seen' position='0 0 10'/>
    <LOD range='5 9'>
      <Group DEF="View3dModelAtCloseRange">
        <Shape>
          <Appearance>
            <Material diffuseColor="1 0 0" />
          </Appearance>
          <Box />
        </Shape>
        <Shape>
          <Appearance>
            <Material diffuseColor="0 0 1" />
          </Appearance>
          <Sphere radius="1.4" />
        </Shape>
      </Group>
      <Group DEF="View2dImageAtLongerRange">
        <Shape>
          <Appearance>
            <Material diffuseColor="0 0 0" />
            <ImageTexture url="&quot;LOD.jpg&quot; &quot;http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.jpg&qu
          </Appearance>
          <Box />
        </Shape>
      </Group>
      <WorldInfo info="Not visible at long range"/>
    </LOD>
  </Scene>
</X3D>
```

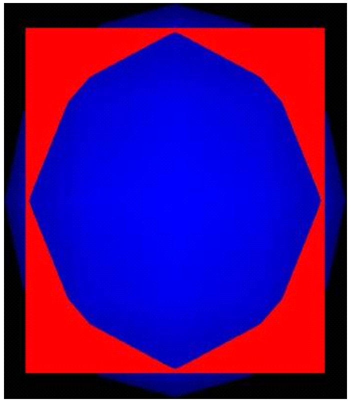
Edit LOD

DEF ●		containerField	
USE ○		☐ children	
center	0	0	0
range	5 9		
forceTransitions	☐		
bboxCenter	0	0	0
bboxSize	-1	-1	-1

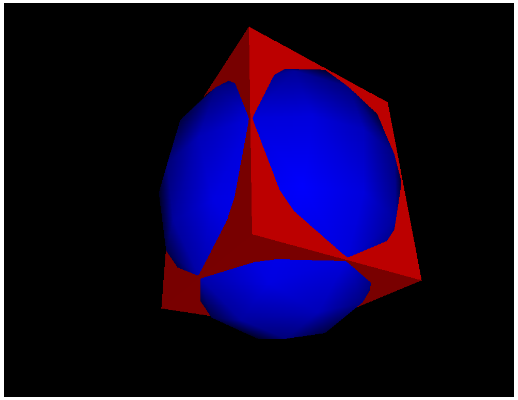
Example LOD range transitions



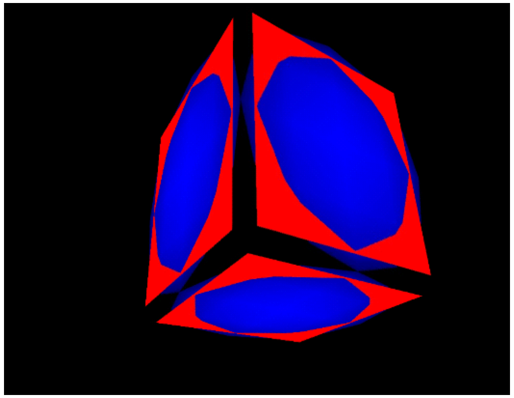
```
LOD: range: 5
  Group: DEF: Model
    Shape
      Appearance
        Material: diffuseColor: 1 0 0
      Box
    Shape
      Appearance
        Material: diffuseColor: 0 0 1
      Sphere: radius: 1.4
  Group: DEF: Image
    Shape
      Appearance
        Material: diffuseColor: 0 0 0
        ImageTexture: url: LOD.jpg
      Box: size: 2.4 2.4 2.4
```



LOD.jpg



Near view



Far view

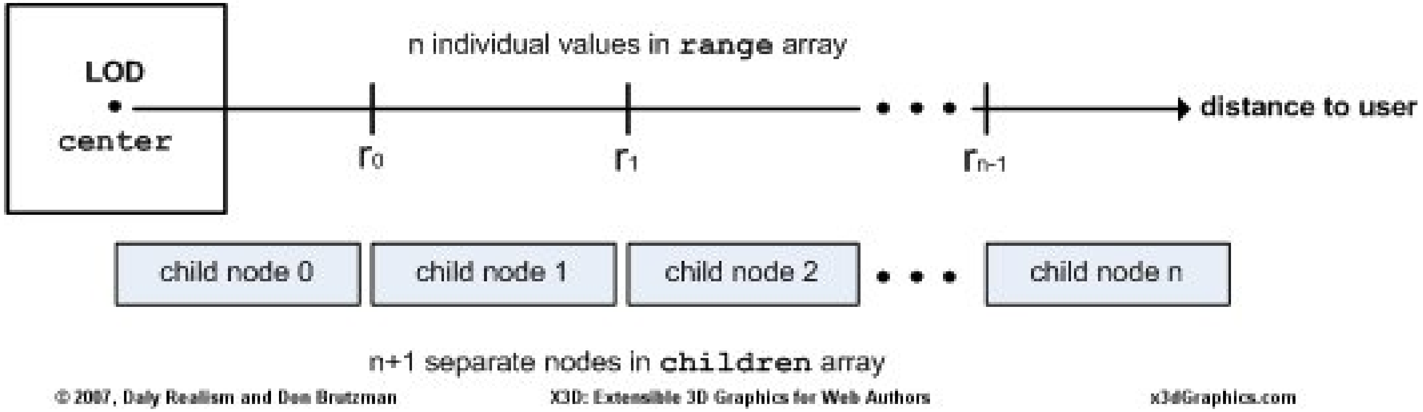
Each child of LOD should represent the same object

- Use Switch to change between different objects

Small difference between X3D and VRML97

- *containerField* name is 'children', not 'level'

LOD range transitions

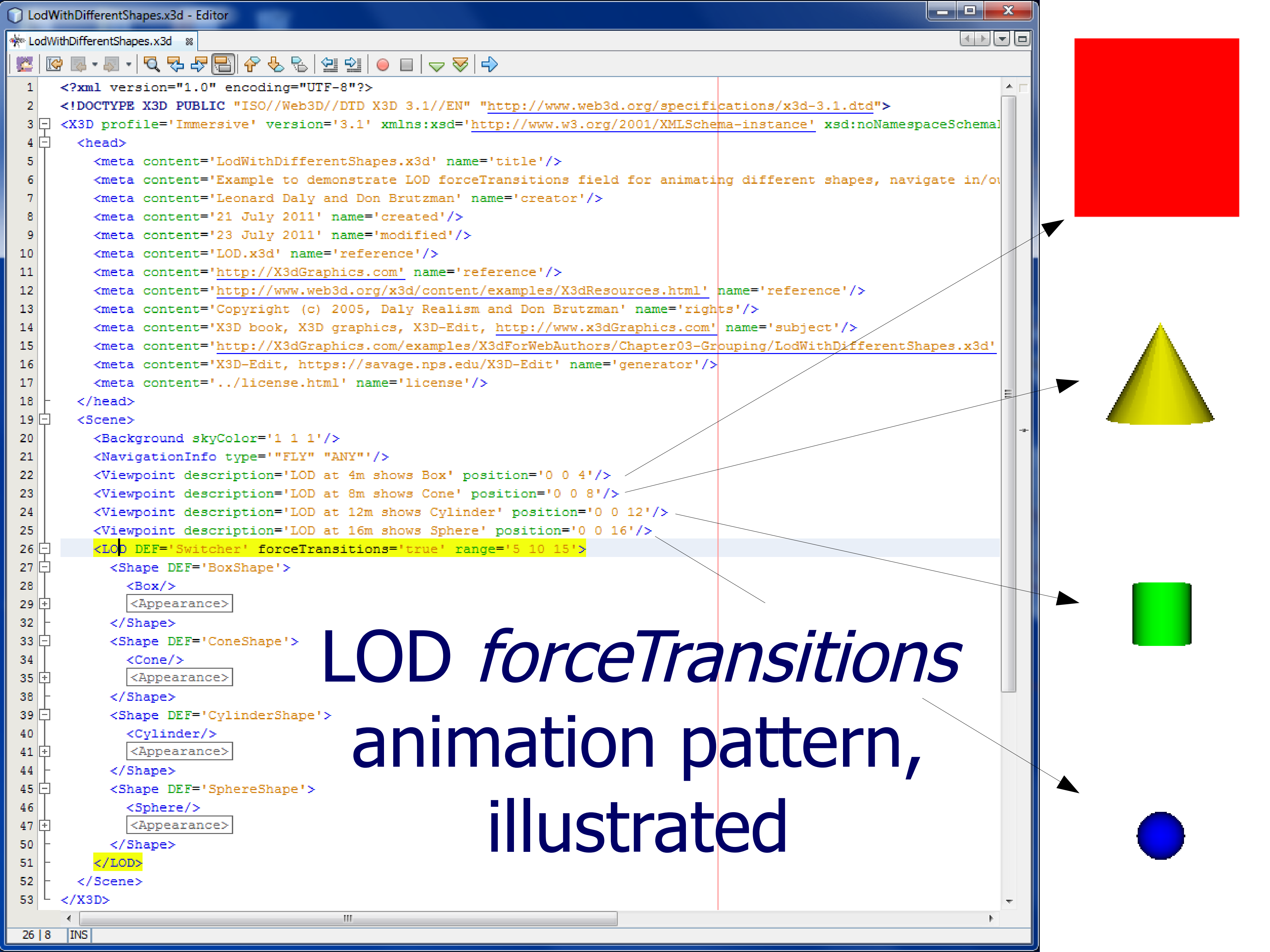

LOD
center
n individual values in range array
distance to user
r0
r1
rn-1
child node 0
child node 1
child node 2
child node n
n+1 separate nodes in children array
© 2007, Daly Realism and Don Brutzman
X3D: Extensible 3D Graphics for Web Authors
x3dGraphics.com

Must have one more child than # of range values

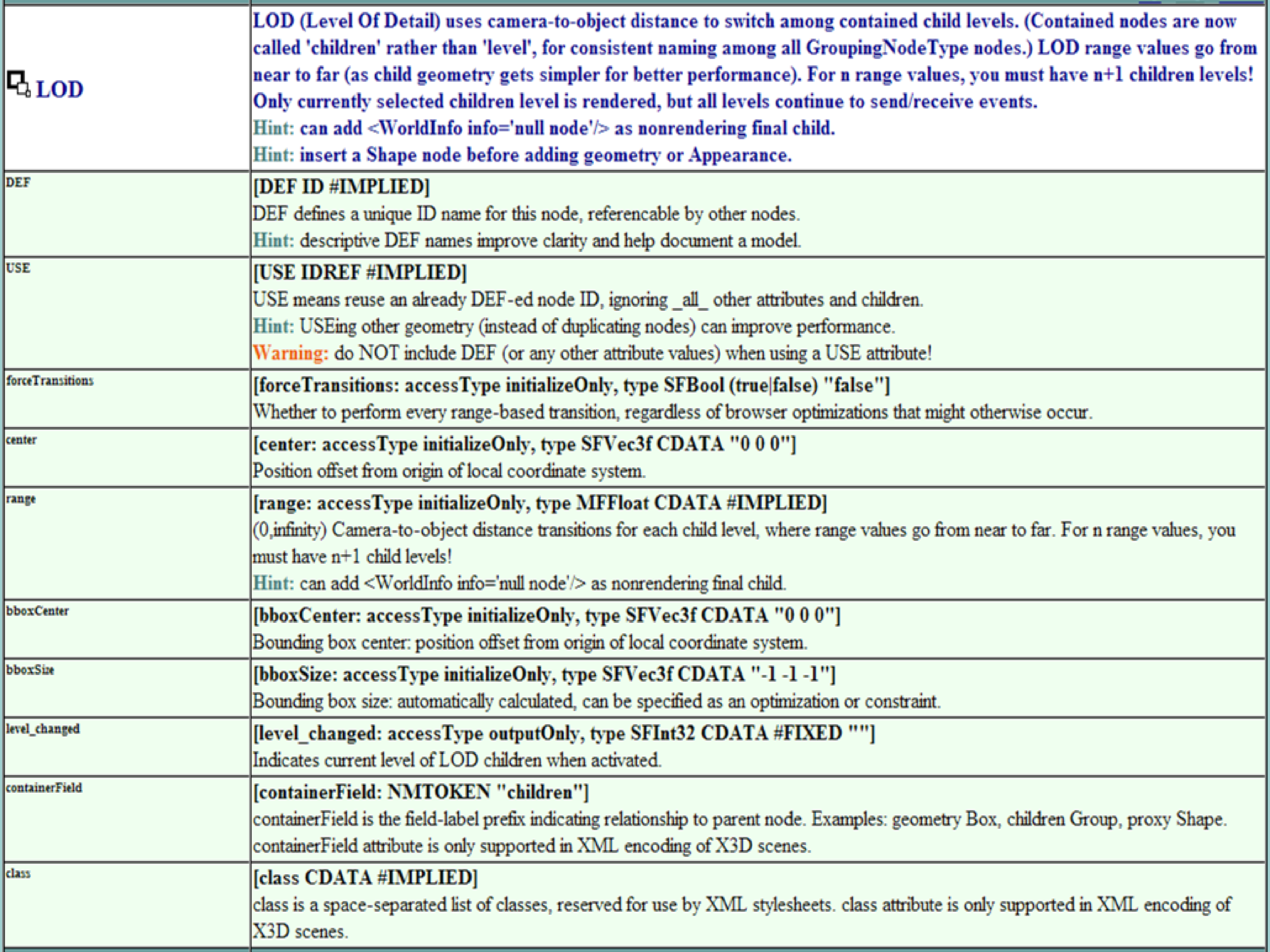
Each value in range array indicates suggested transition point between child versions

- Browser can honor or ignore based on performance
- Use *forceTransitions*='true' for strict view transitions at each of the range values specified

LOD *forceTransitions* animation pattern, illustrated


LodWithDifferentShapes.x3d - Editor
LodWithDifferentShapes.x3d
26 | 8
INS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaL
  <head>
    <meta content='LodWithDifferentShapes.x3d' name='title'/>
    <meta content='Example to demonstrate LOD forceTransitions field for animating different shapes, navigate in/ou
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='21 July 2011' name='created'/>
    <meta content='23 July 2011' name='modified'/>
    <meta content='LOD.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/X3dResources.html' name='reference'/>
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LodWithDifferentShapes.x3d'
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo type='"FLY" "ANY"'/>
    <Viewpoint description='LOD at 4m shows Box' position='0 0 4'/>
    <Viewpoint description='LOD at 8m shows Cone' position='0 0 8'/>
    <Viewpoint description='LOD at 12m shows Cylinder' position='0 0 12'/>
    <Viewpoint description='LOD at 16m shows Sphere' position='0 0 16'/>
    <LOD DEF='Switcher' forceTransitions='true' range='5 10 15'>
      <Shape DEF='BoxShape'>
        <Box/>
        <Appearance>
      </Shape>
      <Shape DEF='ConeShape'>
        <Cone/>
        <Appearance>
      </Shape>
      <Shape DEF='CylinderShape'>
        <Cylinder/>
        <Appearance>
      </Shape>
      <Shape DEF='SphereShape'>
        <Sphere/>
        <Appearance>
      </Shape>
    </LOD>
  </Scene>
</X3D>
```



LOD	LOD (Level Of Detail) uses camera-to-object distance to switch among contained child levels. (Contained nodes are now called 'children' rather than 'level', for consistent naming among all GroupingNodeType nodes.) LOD range values go from near to far (as child geometry gets simpler for better performance). For n range values, you must have n+1 children levels! Only currently selected children level is rendered, but all levels continue to send/receive events. Hint: can add <WorldInfo info='null node'/> as nonrendering final child. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
forceTransitions	[forceTransitions: accessType initializeOnly, type SFBool (true|false) "false"] Whether to perform every range-based transition, regardless of browser optimizations that might otherwise occur.
center	[center: accessType initializeOnly, type SFVec3f CDATA "0 0 0"] Position offset from origin of local coordinate system.
range	[range: accessType initializeOnly, type MFFloat CDATA #IMPLIED] (0,infinity) Camera-to-object distance transitions for each child level, where range values go from near to far. For n range values, you must have n+1 child levels! Hint: can add <WorldInfo info='null node'/> as nonrendering final child.
bboxCenter	[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"] Bounding box center: position offset from origin of local coordinate system.
bboxSize	[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"] Bounding box size: automatically calculated, can be specified as an optimization or constraint.
level_changed	[level_changed: accessType outputOnly, type SFInt32 CDATA #FIXED ""] Indicates current level of LOD children when activated.
containerField	[containerField: NMTOKEN "children"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

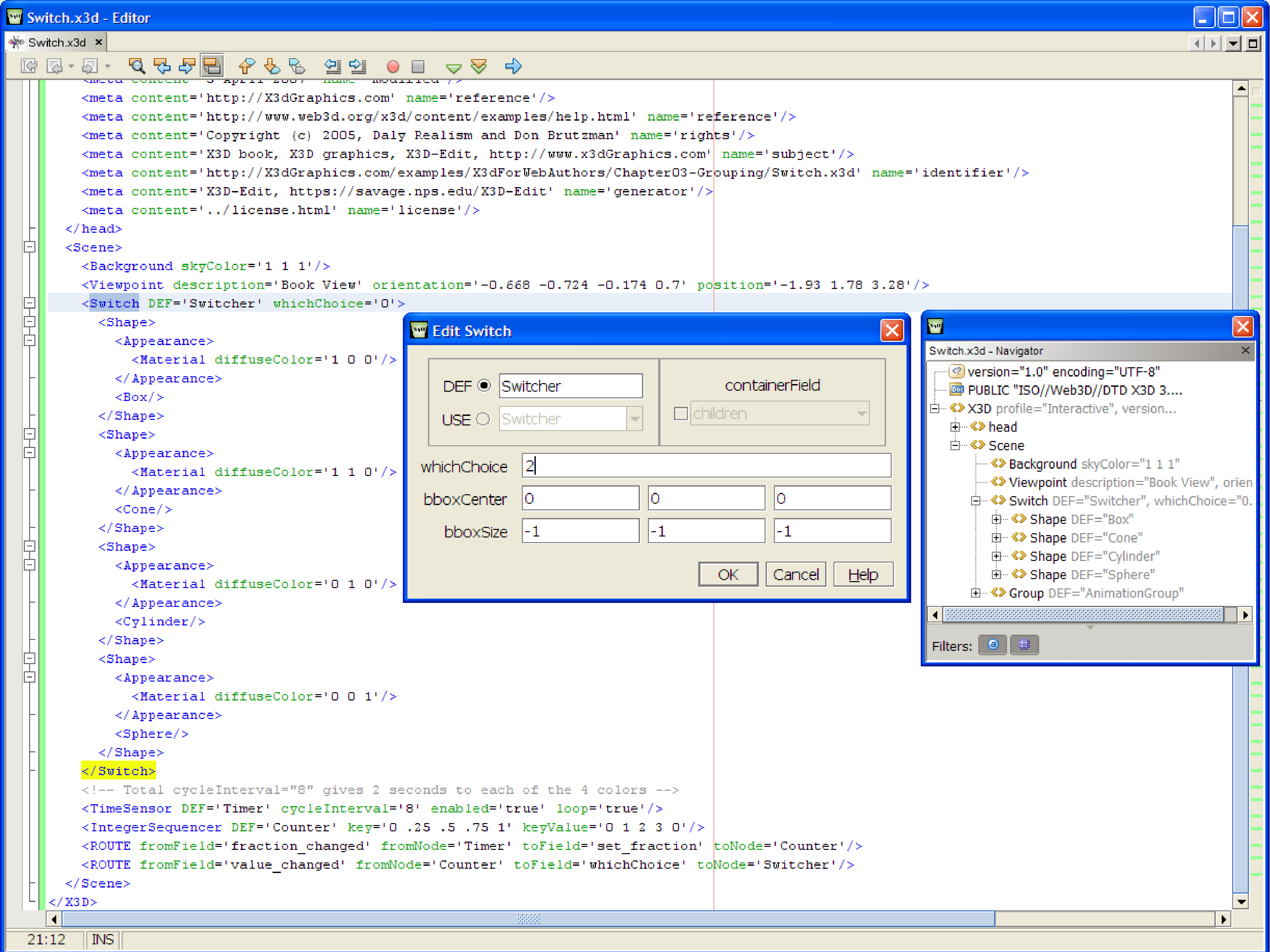
Switch node

Switch selects only one (or none) of its children for rendering

- Initial child index is *whichChoice*='0'
- *whichChoice*='-1' indicates no child is selected

Can manually change values

- Sometimes better to hide geometry rather than to comment out large blocks
  - (which may already have embedded comments)
- Chapter 7 Event Animation describes how to change selections using event animation



Switch.x3d - Editor

Switch.x3d

```
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Switch.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <Switch DEF='Switcher' whichChoice='0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 1 0'/>
        </Appearance>
        <Cone/>
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 1 0'/>
        </Appearance>
        <Cylinder/>
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 0 1'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Switch>
    <!-- Total cycleInterval="8" gives 2 seconds to each of the 4 colors -->
    <TimeSensor DEF='Timer' cycleInterval='8' enabled='true' loop='true'/>
    <IntegerSequencer DEF='Counter' key='0 .25 .5 .75 1' keyValue='0 1 2 3 0'/>
    <ROUTE fromField='fraction_changed' fromNode='Timer' toField='set_fraction' toNode='Counter'/>
    <ROUTE fromField='value_changed' fromNode='Counter' toField='whichChoice' toNode='Switcher'/>
  </Scene>
</X3D>
```

21:12 INS

Edit Switch

DEF ● Switcher

USE ○ Switcher

containerField

□ children

whichChoice 2

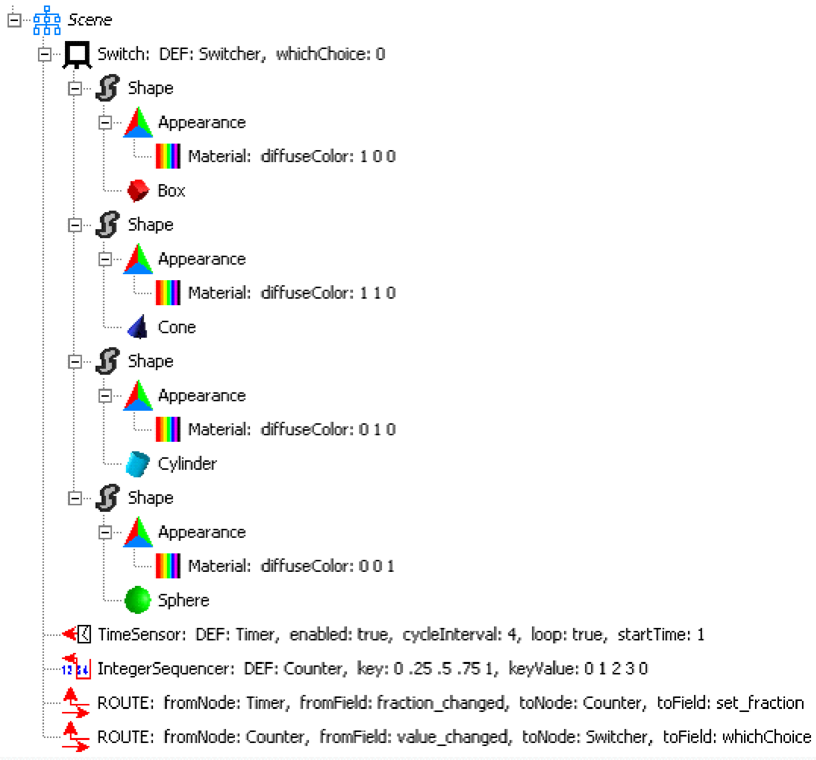
bboxCenter 0 0 0

bboxSize -1 -1 -1

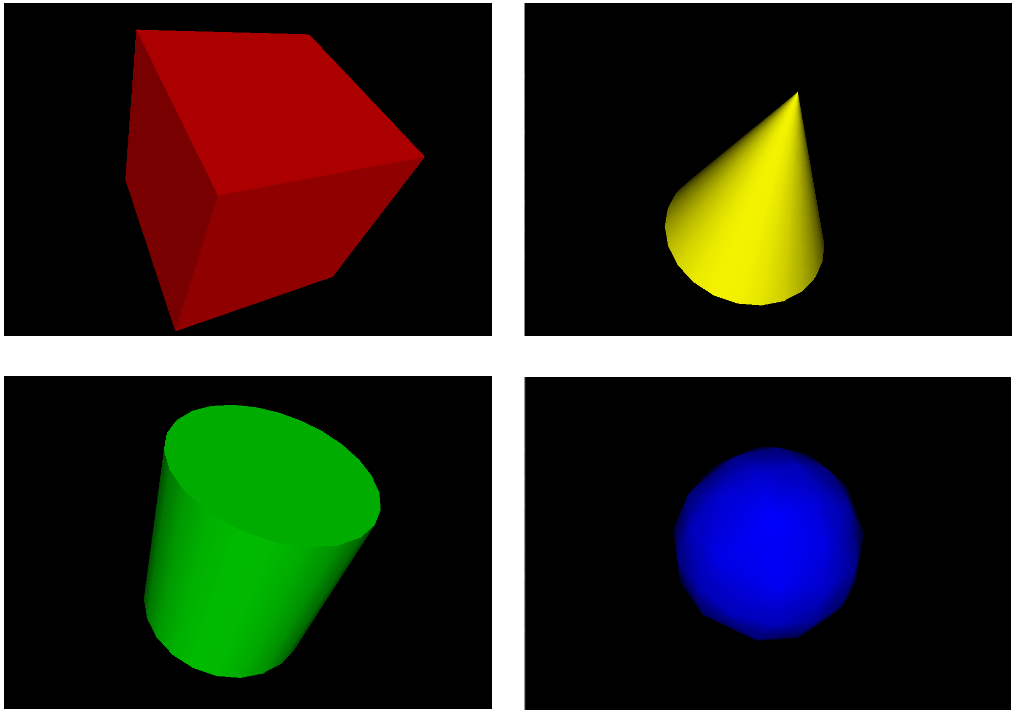
Switch.x3d - Navigator

```
version="1.0" encoding="UTF-8"
PUBLIC "ISO//Web3D//DTD X3D 3....
X3D profile="Interactive", version...
  head
  Scene
    Background skyColor="1 1 1"
    Viewpoint description="Book View", orien
    Switch DEF="Switcher", whichChoice="0.
      Shape DEF="Box"
      Shape DEF="Cone"
      Shape DEF="Cylinder"
      Shape DEF="Sphere"
    Group DEF="AnimationGroup"
```

Switch node example

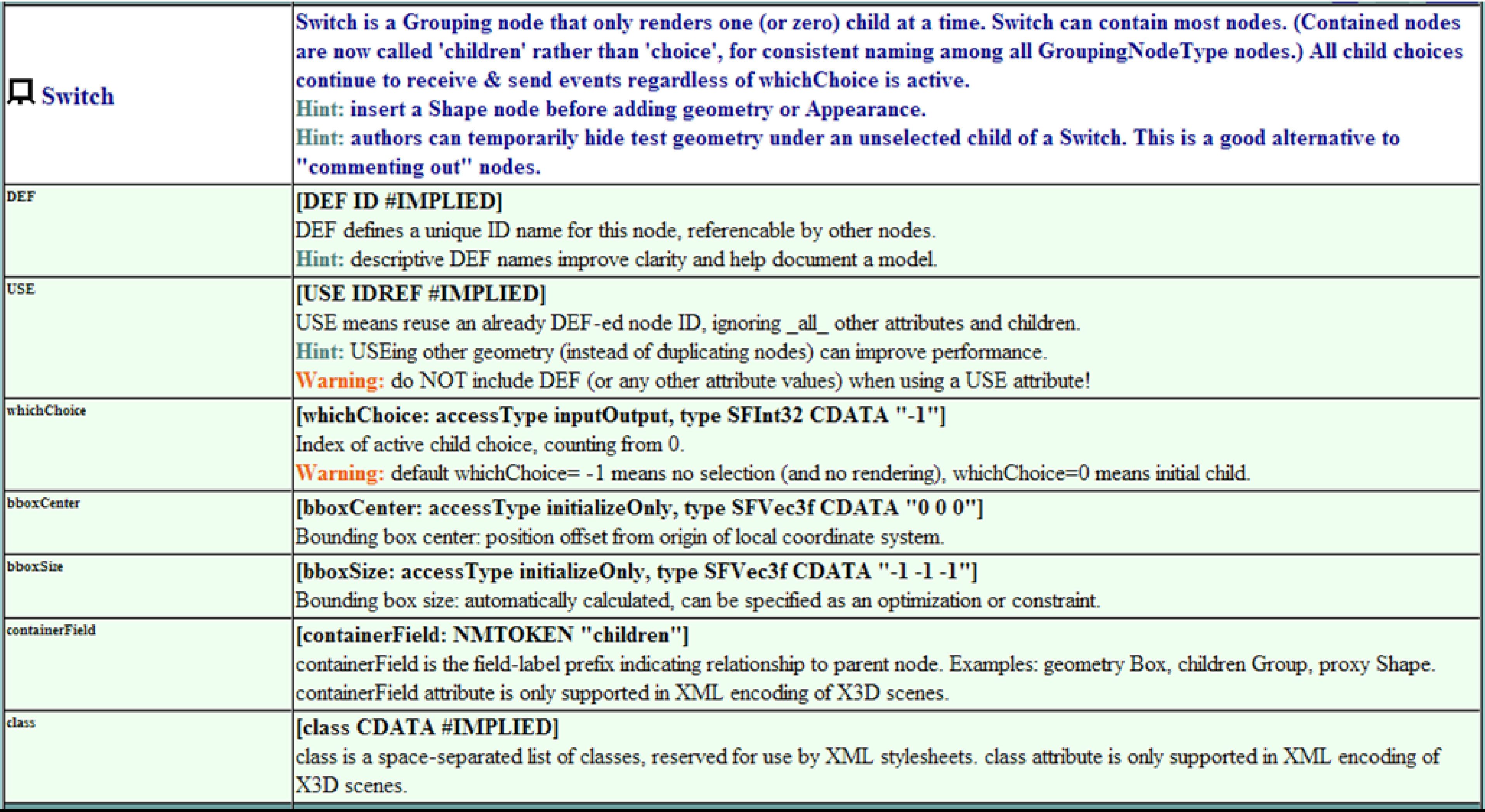


```
Scene
  Switch: DEF: Switcher, whichChoice: 0
    Shape
      Appearance
        Material: diffuseColor: 1 0 0
      Box
    Shape
      Appearance
        Material: diffuseColor: 1 1 0
      Cone
    Shape
      Appearance
        Material: diffuseColor: 0 1 0
      Cylinder
    Shape
      Appearance
        Material: diffuseColor: 0 0 1
      Sphere
  TimeSensor: DEF: Timer, enabled: true, cycleInterval: 4, loop: true, startTime: 1
  IntegerSequencer: DEF: Counter, key: 0 .25 .5 .75 1, keyValue: 0 1 2 3 0
  ROUTE: fromNode: Timer, fromField: fraction_changed, toNode: Counter, toField: set_fraction
  ROUTE: fromNode: Counter, fromField: value_changed, toNode: Switcher, toField: whichChoice
```



Note *whichChoice* starts at index 0, so -1 means none

- Child-node *containerField* ='children', not 'choice'



Switch	**Switch is a Grouping node that only renders one (or zero) child at a time. Switch can contain most nodes. (Contained nodes are now called 'children' rather than 'choice', for consistent naming among all GroupingNodeType nodes.) All child choices continue to receive & send events regardless of whichChoice is active.** **Hint: insert a Shape node before adding geometry or Appearance.** **Hint: authors can temporarily hide test geometry under an unselected child of a Switch. This is a good alternative to "commenting out" nodes.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
whichChoice	**[whichChoice: accessType inputOutput, type SFInt32 CDATA "-1"]** Index of active child choice, counting from 0. Warning: default whichChoice= -1 means no selection (and no rendering), whichChoice=0 means initial child.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

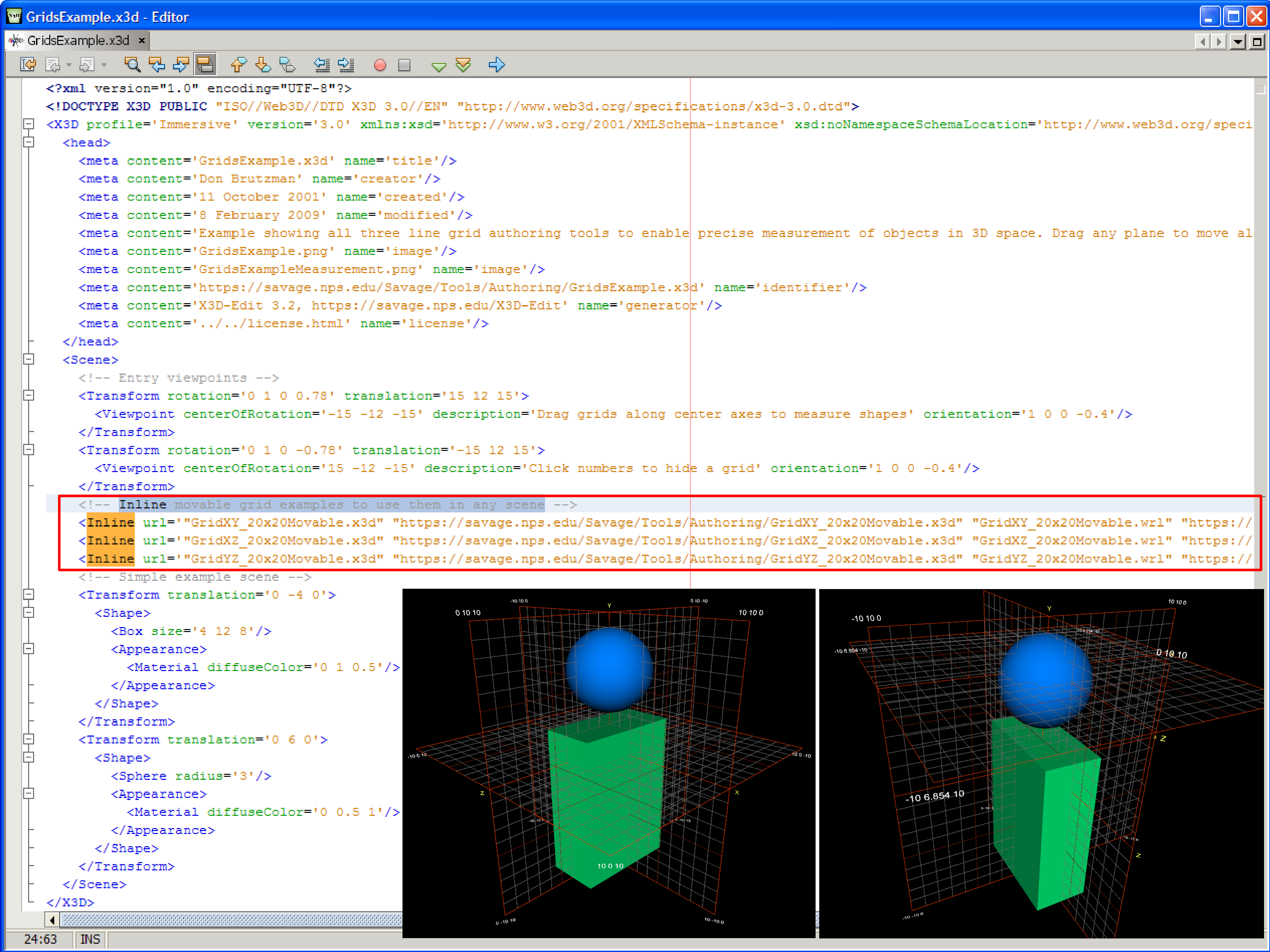
Additional Resources

Savage Tools for Authoring: grids

3D grids for object placement are provided by many 3D authoring environments

Reusable grid examples: Savage authoring tools

- https://savage.nps.edu/Savage/Tools/Authoring
  - Grids Example
  - Grids Example Pixel Texture
  - Grid XY 20x 20 Movable
  - Grid XZ 20x 20 Fixed
  - Grid XZ 20x 20 Movable
  - Grid XZExample
  - Grid XZPrototype
  - Grid YZ 20x 20 Movable



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile='Immersive' version='3.0' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/speci
  <head>
    <meta content='GridsExample.x3d' name='title'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='11 October 2001' name='created'/>
    <meta content='8 February 2009' name='modified'/>
    <meta content='Example showing all three line grid authoring tools to enable precise measurement of objects in 3D space. Drag any plane to move al
    <meta content='GridsExample.png' name='image'/>
    <meta content='GridsExampleMeasurement.png' name='image'/>
    <meta content='https://savage.nps.edu/Savage/Tools/Authoring/GridsExample.x3d' name='identifier'/>
    <meta content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../../license.html' name='license'/>
  </head>
  <Scene>
    <!-- Entry viewpoints -->
    <Transform rotation='0 1 0 0.78' translation='15 12 15'>
      <Viewpoint centerOfRotation='-15 -12 -15' description='Drag grids along center axes to measure shapes' orientation='1 0 0 -0.4'/>
    </Transform>
    <Transform rotation='0 1 0 -0.78' translation='-15 12 15'>
      <Viewpoint centerOfRotation='15 -12 -15' description='Click numbers to hide a grid' orientation='1 0 0 -0.4'/>
    </Transform>
    <!-- Inline movable grid examples to use them in any scene -->
    <Inline url='"GridXY_20x20Movable.x3d" "https://savage.nps.edu/Savage/Tools/Authoring/GridXY_20x20Movable.x3d" "GridXY_20x20Movable.wrl" "https://
    <Inline url='"GridXZ_20x20Movable.x3d" "https://savage.nps.edu/Savage/Tools/Authoring/GridXZ_20x20Movable.x3d" "GridXZ_20x20Movable.wrl" "https://
    <Inline url='"GridYZ_20x20Movable.x3d" "https://savage.nps.edu/Savage/Tools/Authoring/GridYZ_20x20Movable.x3d" "GridYZ_20x20Movable.wrl" "https://
    <!-- Simple example scene -->
    <Transform translation='0 -4 0'>
      <Shape>
        <Box size='4 12 8'/>
        <Appearance>
          <Material diffuseColor='0 1 0.5'/>
        </Appearance>
      </Shape>
    </Transform>
    <Transform translation='0 6 0'>
      <Shape>
        <Sphere radius='3'/>
        <Appearance>
          <Material diffuseColor='0 0.5 1'/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Basic examples: DIS gimbals

Common problem: trying to adapt roll, pitch, yaw angles into single axis-angle SFRotation value

Example solution provided in X3D Basic Examples

- http://www.web3d.org/x3d/content/examples/Basic
- Distributed Interactive Simulation, Gimbals

Dragging circular rings of the gimbals provides SFRotation and DIS (roll pitch yaw) values

- External script Gimbals.js shows conversion math

Chapter Summary

Chapter Summary

Grouping nodes collect and select other nodes

- Concepts: DEF/USE, units of measure, coordinate systems, right-hand rule, and bounding boxes

Grouping nodes are fundamental to well-behaved design of an effective scene graph

- Group, StaticGroup collect children nodes together
- Transform provides translation, rotation, scale
- Inline loads other X3D content
- LOD supports level-of-detail performance gains
- Switch selects one (or none) of children

- Related grouping nodes covered in Chapter 4
  - Anchor, Billboard, Collision

Suggested exercises

Demonstrate the ability to perform translations and rotations by arranging several geometric shapes together

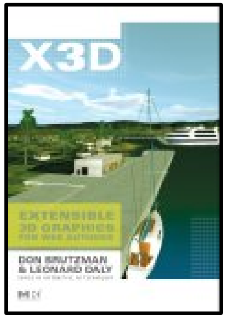
Inline an X3D scene into your own, or vice versa

Create low-fidelity, medium-fidelity, high-fidelity versions of an object, then arrange them within a level-of-detail LOD node

Use a Switch to “hide” unwanted geometry

References

References 1


X3D
EXTENSIBLE 3D GRAPHICS FOR WEB AUTHORS
DON BRUTZMAN & LEONARD DALY

X3D: Extensible 3D Graphics for Web Authors by Don Brutzman and Leonard Daly, Morgan Kaufmann Publishers, April 2007, 468 pages.

- Chapter 3, Grouping Nodes
- http://x3dGraphics.com
- http://x3dgraphics.com/examples/X3dForWebAuthors

X3D Resources

- http://www.web3d.org/x3d/content/examples/X3dResources.html

References 2

X3D-Edit Authoring Tool

- https://savage.nps.edu/X3D-Edit

X3D Scene Authoring Hints

- http://x3dgraphics.com/examples/X3dSceneAuthoringHints.html

X3D Graphics Specification

- http://www.web3d.org/x3d/specifications
- Also available as help pages within X3D-Edit

References 3

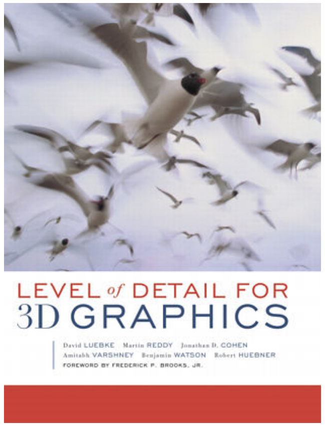
VRML 2.0 Sourcebook by Andrea L. Ames, David R. Nadeau, and John L. Moreland, John Wiley & Sons, 1996.

- http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm
- http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook
- Chapter 05 - Positioning Shapes
- Chapter 06 - Rotating Shapes
- Chapter 07 - Scaling Shapes

References 4

Level of Detail for 3D Graphics, by D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner, Morgan Kaufmann, second edition, 2006.

http://lodbook.com


LEVEL of DETAIL FOR
3D GRAPHICS
David LUEBKE Martin REDDY Jonathan D. COHEN
Amitabh VARSHNEY Benjamin WATSON Robert HUEBNER
FOREWORD BY FREDERICK P. BROOKS, JR.

Contact

Don Brutzman

brutzman@nps.edu

http://faculty.nps.edu/brutzman

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.2149 voice

CGEMS, SIGGRAPH, Eurographics

The Computer Graphics Educational Materials Source(CGEMS) site is designed for educators

- to provide a source of refereed high-quality content
- as a service to the Computer Graphics community
- freely available, directly prepared for classroom use
- http://cgems.inesc.pt

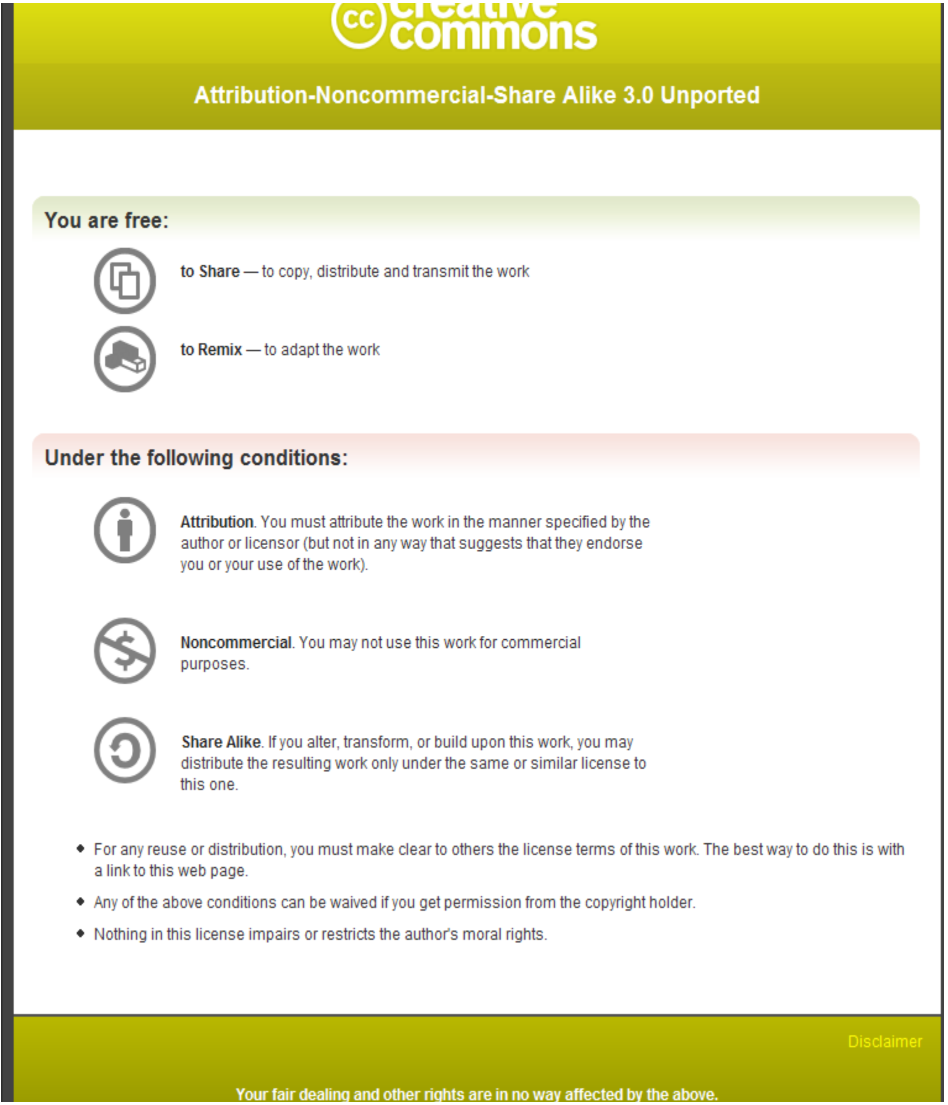
X3D for Web Authors recognized by CGEMS! ☺

- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics

Creative Commons open-source license

http://creativecommons.org/licenses/by-nc-sa/3.0



Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial. You may not use this work for commercial purposes.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

Disclaimer

Your fair dealing and other rights are in no way affected by the above.

Open-source license
for X3D-Edit software and X3D example scenes

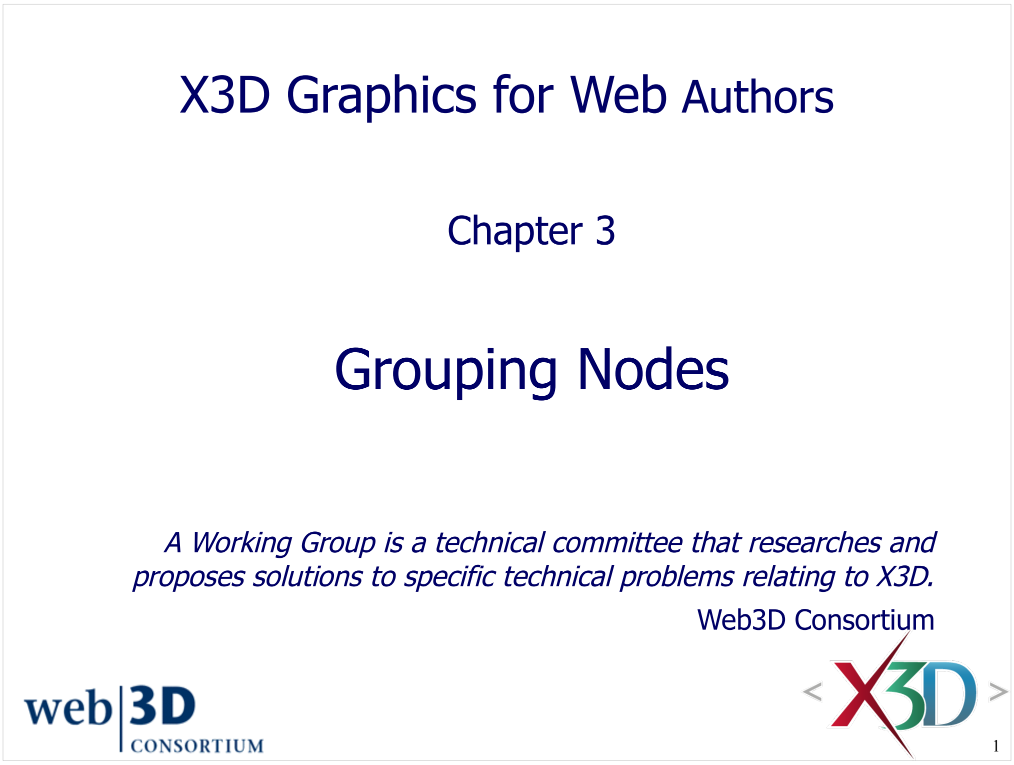
http://www.web3d.org/x3d/content/examples/license.html

Copyright (c) 1995-2013 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the Naval Postgraduate School (NPS) Modeling Virtual Environments and Simulation (MOVES) Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



X3D Graphics for Web Authors

Chapter 3

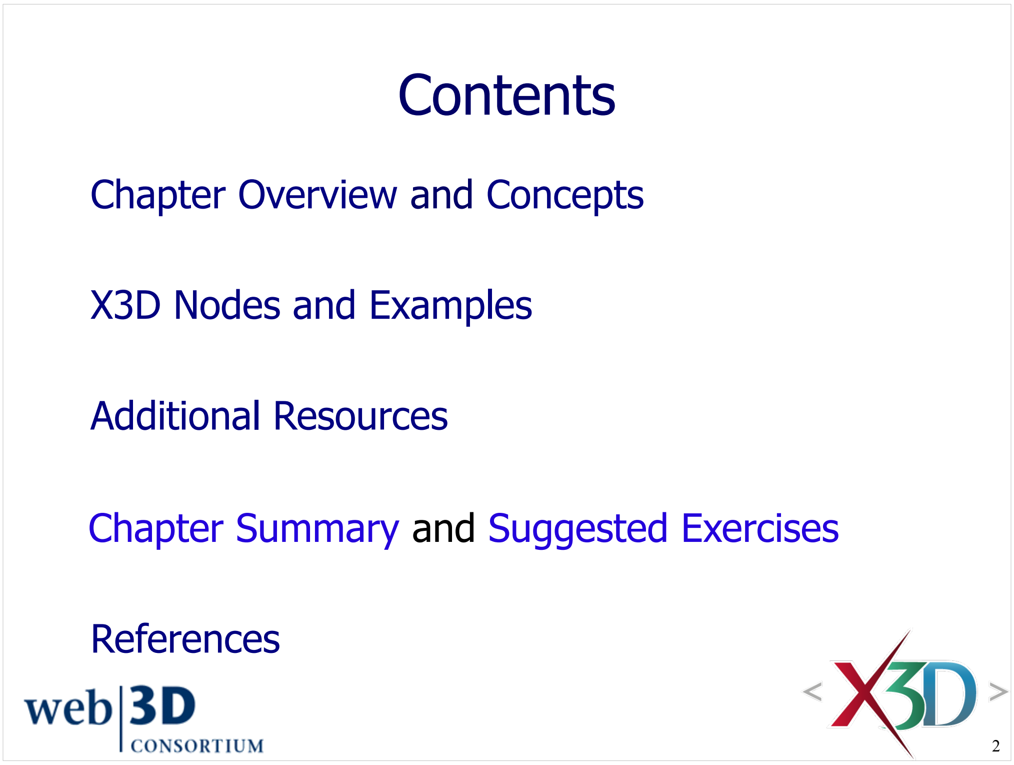
Grouping Nodes

A Working Group is a technical committee that researches and proposes solutions to specific technical problems relating to X3D.

Web3D Consortium

1

The Web3D Consortium (www.web3d.org) actively supports many working groups and an active community of interested users. Scene authors, software developers, industry professionals, and 3D enthusiasts continue to add to the long list of capabilities in X3D. Web3D Consortium membership is open to both organizations and individuals.



Contents

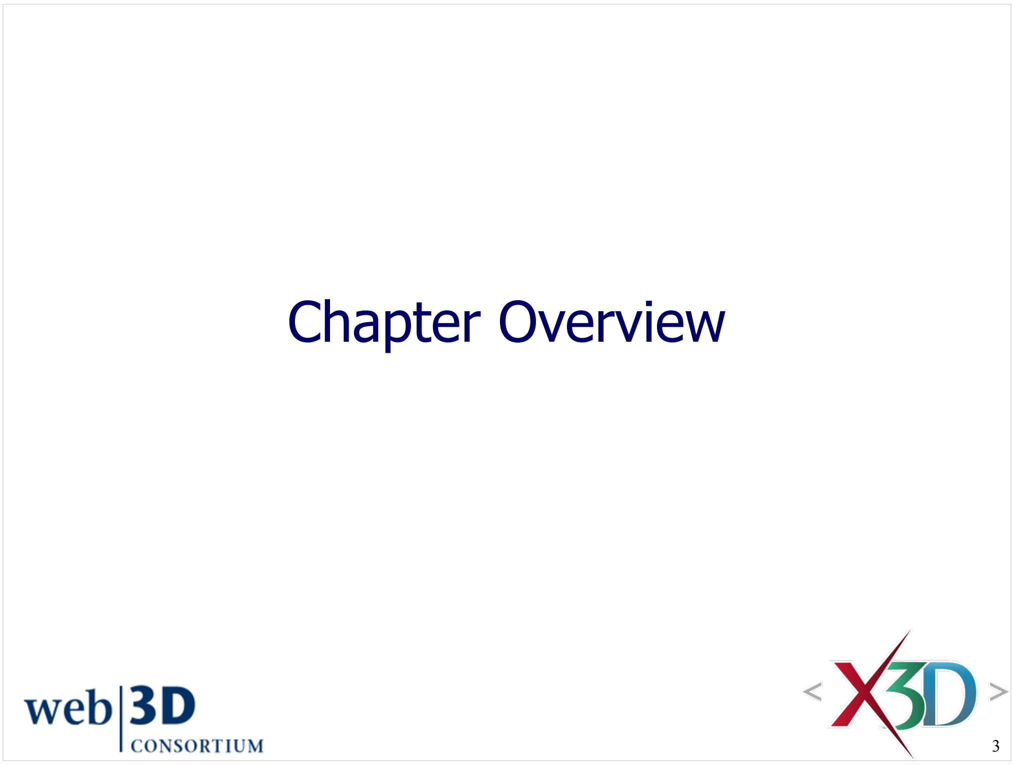
Chapter Overview and Concepts

X3D Nodes and Examples

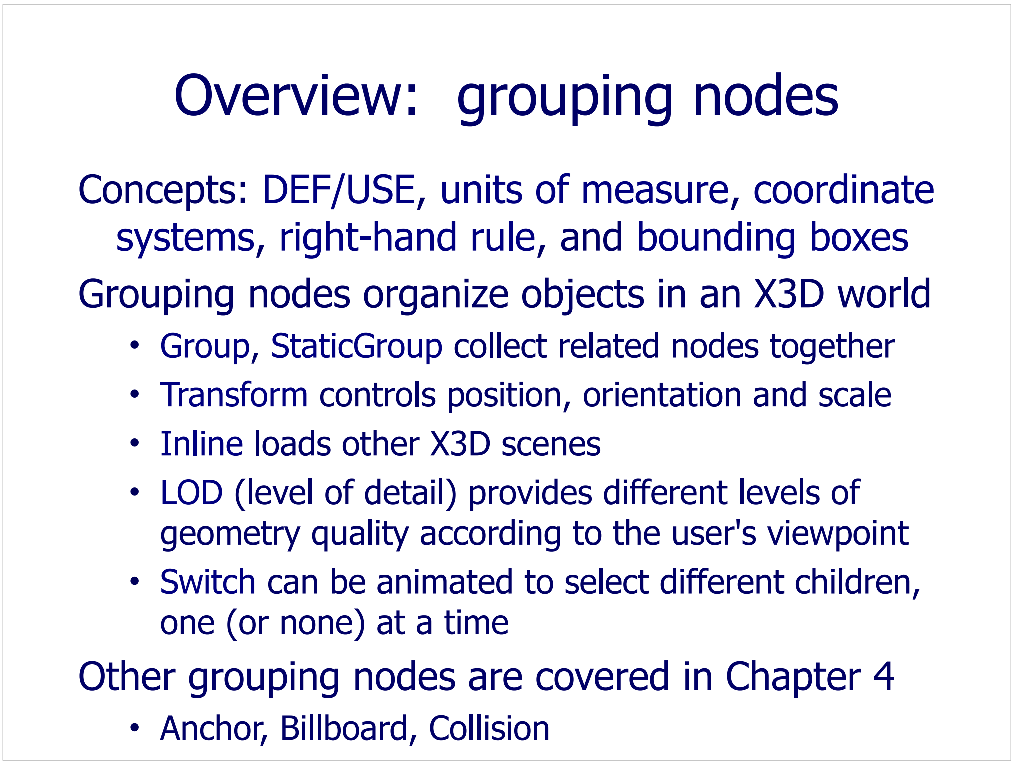
Additional Resources

Chapter Summary and Suggested Exercises

References



Chapter Overview



Overview: grouping nodes

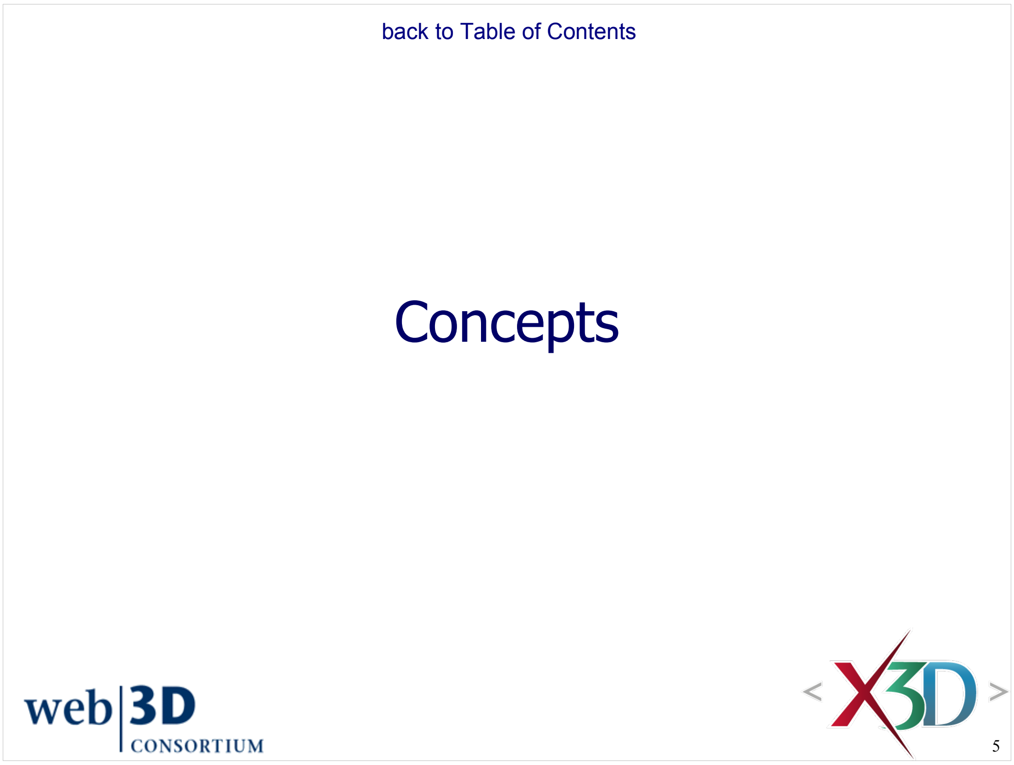
Concepts: DEF/USE, units of measure, coordinate systems, right-hand rule, and bounding boxes

Grouping nodes organize objects in an X3D world

- Group, StaticGroup collect related nodes together
- Transform controls position, orientation and scale
- Inline loads other X3D scenes
- LOD (level of detail) provides different levels of geometry quality according to the user's viewpoint
- Switch can be animated to select different children, one (or none) at a time

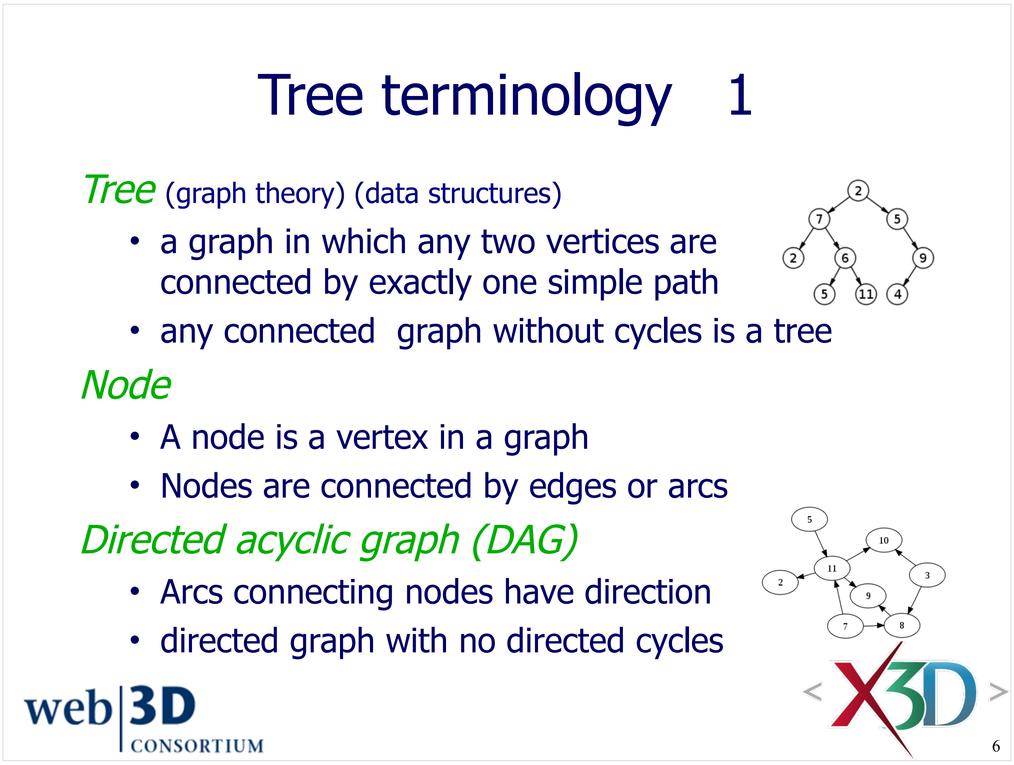
Other grouping nodes are covered in Chapter 4

- Anchor, Billboard, Collision



back to Table of Contents

Concepts



Tree terminology 1

Tree (graph theory) (data structures)

- a graph in which any two vertices are connected by exactly one simple path
- any connected graph without cycles is a tree

Node

- A node is a vertex in a graph
- Nodes are connected by edges or arcs

Directed acyclic graph (DAG)

- Arcs connecting nodes have direction
- directed graph with no directed cycles

2
7
5
2
6
9
5
11
4
5
10
11
3
2
9
7
8

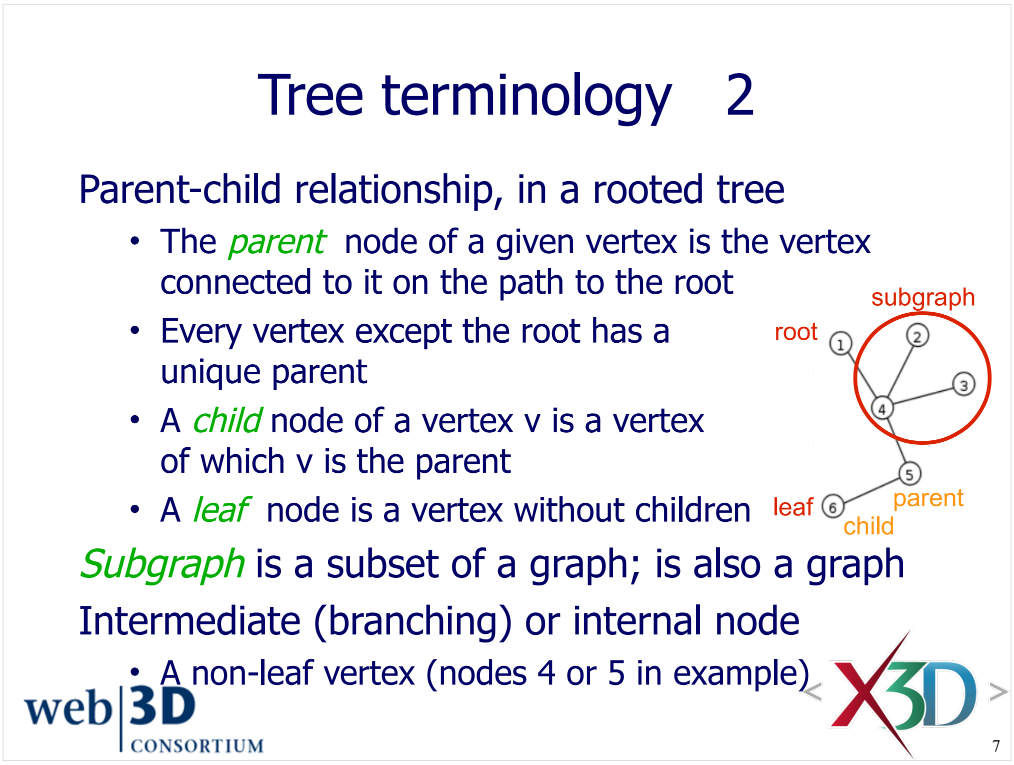
These definitions are good discussion topics. Graph theory is cool!

- Parent-child relationship in a tree determines direction of an arc

Images used with permission from corresponding Wikipedia pages

- http://en.wikipedia.org/wiki/File:Binary_tree.svg
- http://en.wikipedia.org/wiki/File:Directed_acyclic_graph_3.svg

The directed acyclic (no cycles) nature of these graphs is important. A cycle in the graph, when traversed, becomes an infinite loop. Infinite loops are not acceptable for a Web-based presentation language because they can freeze a user's Web browser.



Tree terminology 2

Parent-child relationship, in a rooted tree

- The *parent* node of a given vertex is the vertex connected to it on the path to the root
- Every vertex except the root has a unique parent
- A *child* node of a vertex v is a vertex of which v is the parent
- A *leaf* node is a vertex without children

Subgraph is a subset of a graph; is also a graph

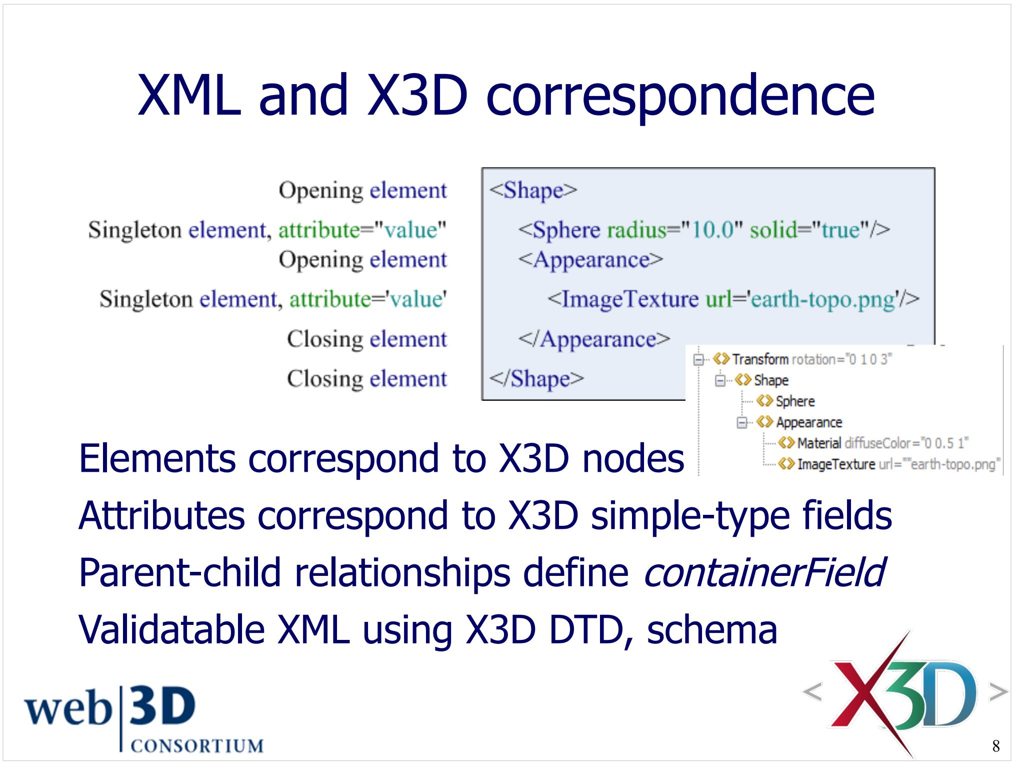
Intermediate (branching) or internal node

- A non-leaf vertex (nodes 4 or 5 in example)

These definitions are good discussion topics. Graph theory is cool!

Image used with permission from corresponding Wikipedia page

- http://en.wikipedia.org/wiki/File:Tree_graph.svg



XML and X3D correspondence

Opening element	`<Shape>`
Singleton element, attribute="value"	`<Sphere radius="10.0" solid="true"/>`
Opening element	`<Appearance>`
Singleton element, attribute='value'	`<ImageTexture url='earth-topo.png'/>`
Closing element	`</Appearance>`
Closing element	`</Shape>`

```
Transform rotation="0 1 0 3"
  Shape
    Sphere
    Appearance
      Material diffuseColor="0 0.5 1"
      ImageTexture url='"earth-topo.png"'
```

Elements correspond to X3D nodes

Attributes correspond to X3D simple-type fields

Parent-child relationships define *containerField*

Validatable XML using X3D DTD, schema

web|3D CONSORTIUM

<X3D>

8

XML documents have a tree structure that is a good match for the X3D scene graph.

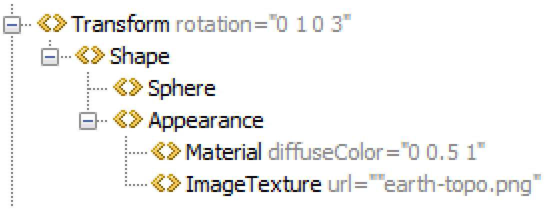
- Graph nodes correspond to X3D node elements.
- Graph arcs correspond to parent-child relationships

Critical benefit: XML well-formed checks and validation detect numerous tricky errors.

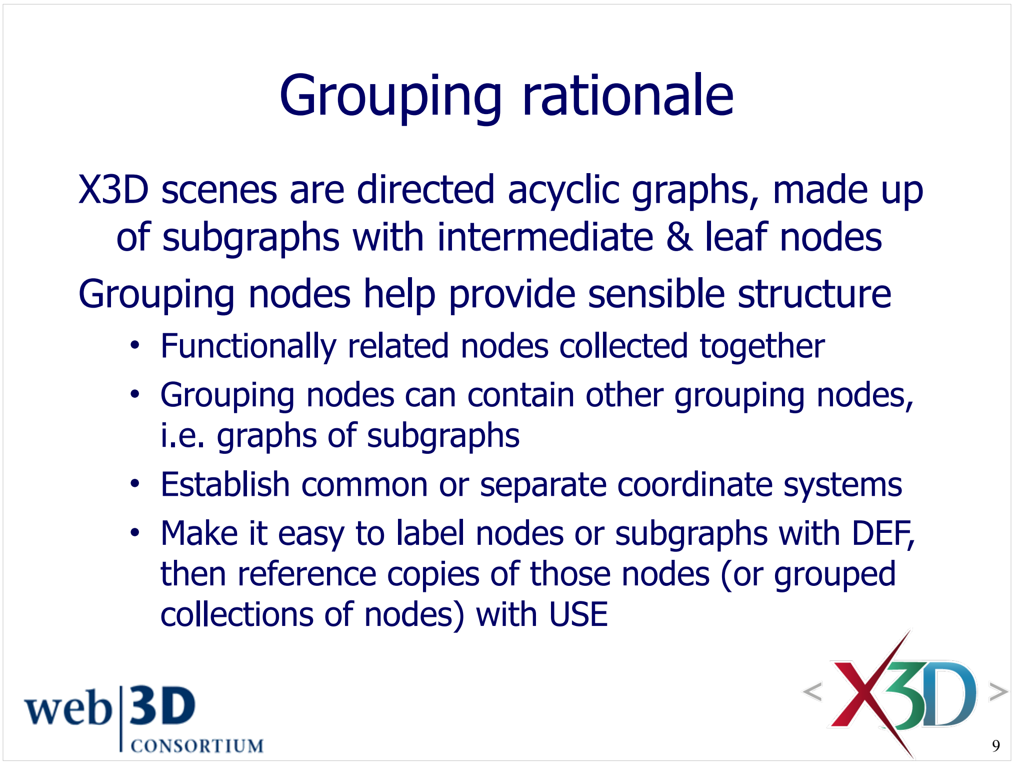
- Draconian parse rule prevents an XML parser from continuing if errors are encountered. This is a good thing, because it forces the author to find and fix critical input problems, rather than having the application somehow trying to fix or recover from incorrect input.
- This approach thus prevents Garbage In Garbage Out (GIGO) syndrome.
- It is better to know that faults occur. The worst error is the unrecognized error.

[This slide is a review from Chapter 1]

Corresponding tree diagram:



```
Transform rotation="0 1 0 3"
  Shape
    Sphere
    Appearance
      Material diffuseColor="0 0.5 1"
      ImageTexture url='"earth-topo.png"'
```



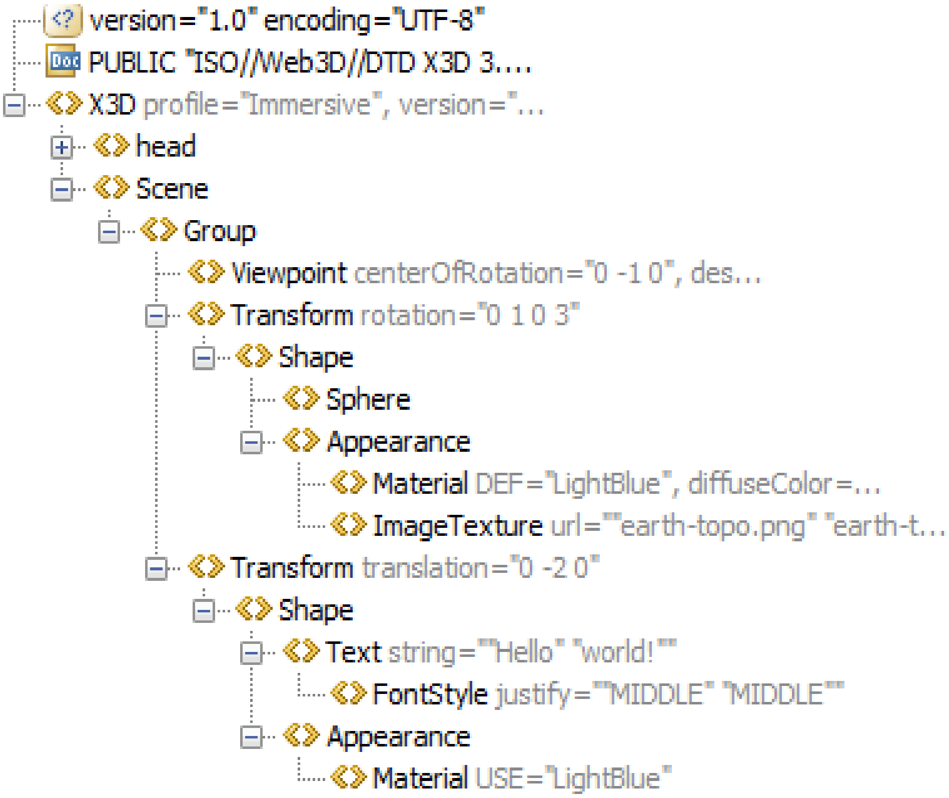
Grouping rationale

X3D scenes are directed acyclic graphs, made up of subgraphs with intermediate & leaf nodes

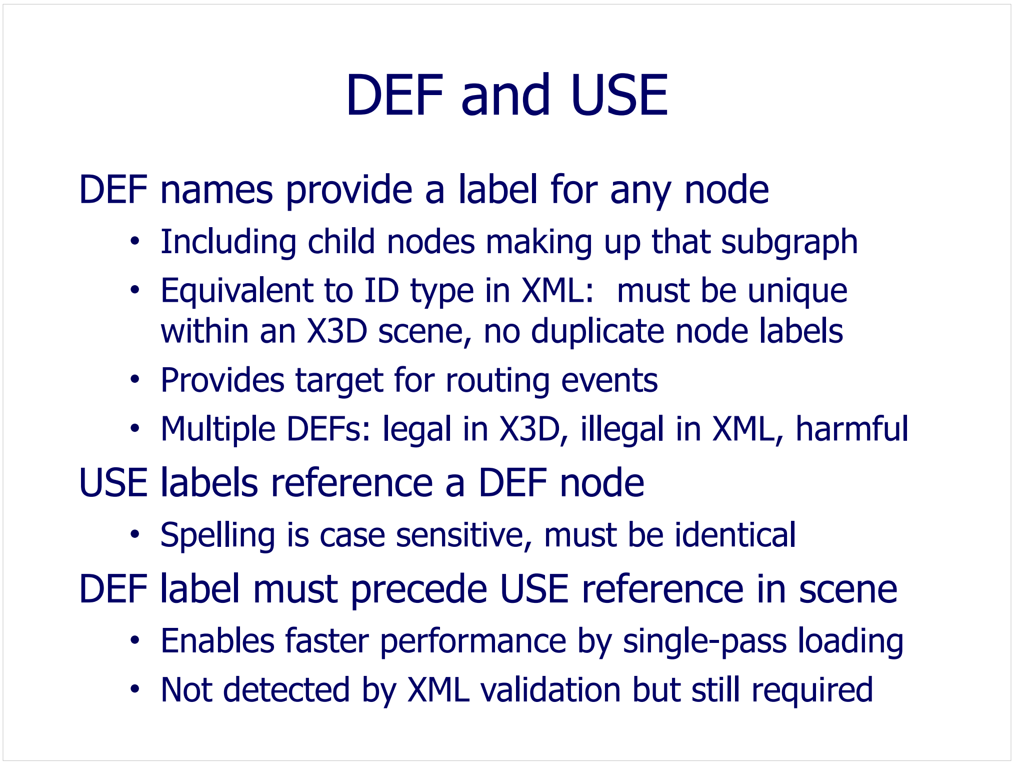
Grouping nodes help provide sensible structure

- Functionally related nodes collected together
- Grouping nodes can contain other grouping nodes, i.e. graphs of subgraphs
- Establish common or separate coordinate systems
- Make it easy to label nodes or subgraphs with DEF, then reference copies of those nodes (or grouped collections of nodes) with USE

DEF and USE provide special performance benefits that do not break the fundamental nature of the scene-graph tree.



```
version="1.0" encoding="UTF-8"
PUBLIC "ISO//Web3D//DTD X3D 3....
X3D profile="Immersive", version="...
    head
    Scene
        Group
            Viewpoint centerOfRotation="0 -1 0", des...
            Transform rotation="0 1 0 3"
                Shape
                    Sphere
                    Appearance
                        Material DEF="LightBlue", diffuseColor=...
                        ImageTexture url=""earth-topo.png" "earth-t...
            Transform translation="0 -2 0"
                Shape
                    Text string=""Hello" "world!""
                        FontStyle justify=""MIDDLE" "MIDDLE""
                    Appearance
                        Material USE="LightBlue"
```



DEF and USE

DEF names provide a label for any node

- Including child nodes making up that subgraph
- Equivalent to ID type in XML: must be unique within an X3D scene, no duplicate node labels
- Provides target for routing events
- Multiple DEFs: legal in X3D, illegal in XML, harmful

USE labels reference a DEF node

- Spelling is case sensitive, must be identical

DEF label must precede USE reference in scene

- Enables faster performance by single-pass loading
- Not detected by XML validation but still required

USE nodes can greatly speed up performance and reduce memory requirements for duplicative scene subgraphs.

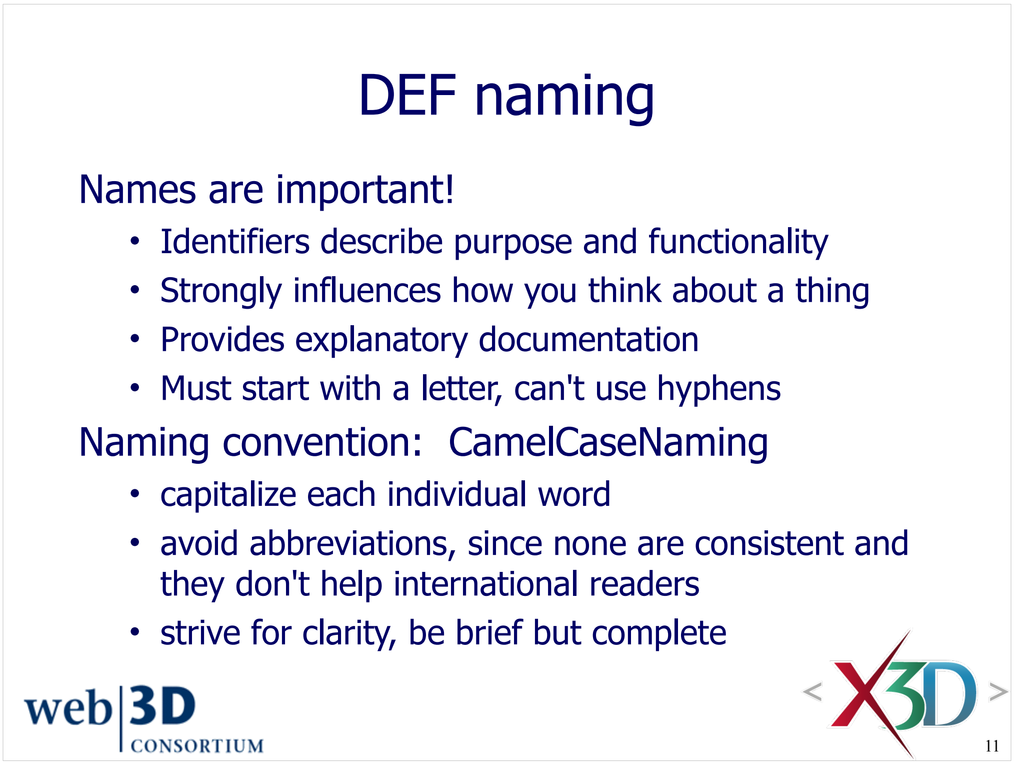
USE nodes also simplify the authoring and modification of scene consistency for commonly reused graphics items such as material values or image textures.

Since DEF defines the original instance of a node, and USE references it, this is an example of copy by reference (rather than copy by value).

- X3D DEF corresponds to XML datatype ID
- X3D USE corresponds to XML datatype IDREF
- Thus XML validation will confirm that DEF/USE names are legally formed and also that a USE name has a corresponding DEF

Software engineering design principle: Don't Repeat Yourself (DRY)

- In other words, define a critical value only once, and then refer to it as needed



DEF naming

Names are important!

- Identifiers describe purpose and functionality
- Strongly influences how you think about a thing
- Provides explanatory documentation
- Must start with a letter, can't use hyphens

Naming convention: CamelCaseNaming

- capitalize each individual word
- avoid abbreviations, since none are consistent and they don't help international readers
- strive for clarity, be brief but complete

web 3D CONSORTIUM

<X3D>

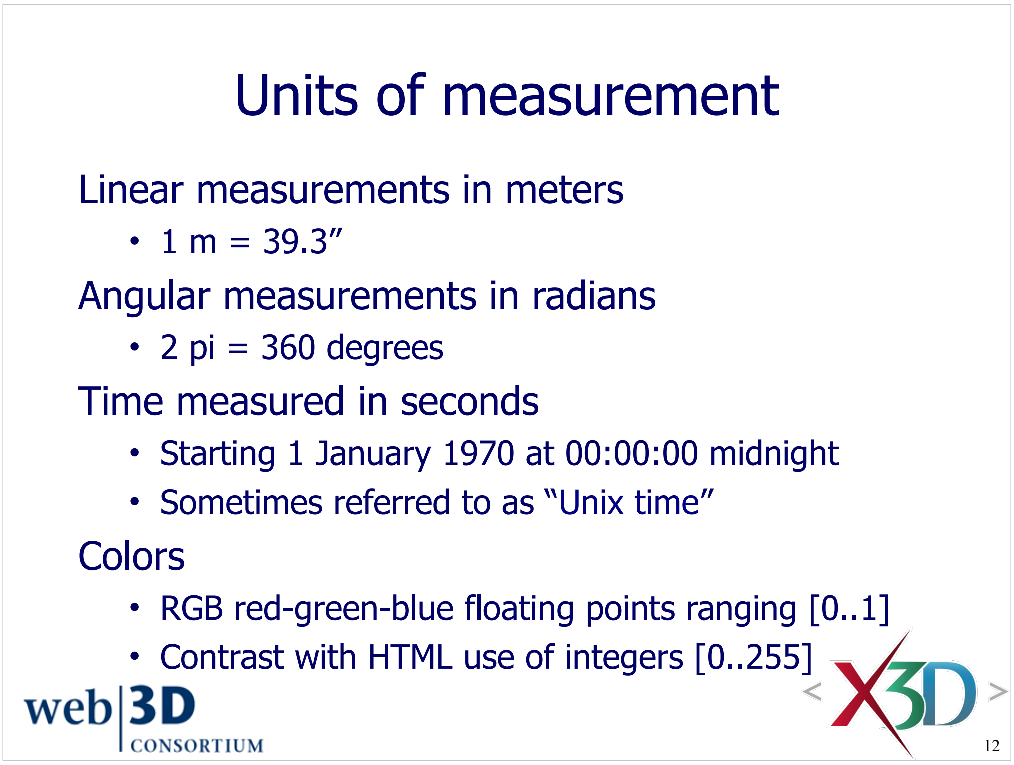
11

Naming metric: can the DEF name be used in a sentence sensibly?

Irony: you know that you have the proper name for something when no one asks about it any more. – Jeff Weekley

The X3D Scene Authoring Hints include guidance on good naming conventions. These hints are available in the X3D-Edit help system, and also online at

http://www.web3d.org/x3d/content/examples/X3dSceneAuthoringHints.html#NamingConventions



Units of measurement

Linear measurements in meters

- 1 m = 39.3”

Angular measurements in radians

- 2 pi = 360 degrees

Time measured in seconds

- Starting 1 January 1970 at 00:00:00 midnight
- Sometimes referred to as “Unix time”

Colors

- RGB red-green-blue floating points ranging [0..1]
- Contrast with HTML use of integers [0..255]

Warning: using degree values rather than radians is a common mistake by new authors learning X3D.

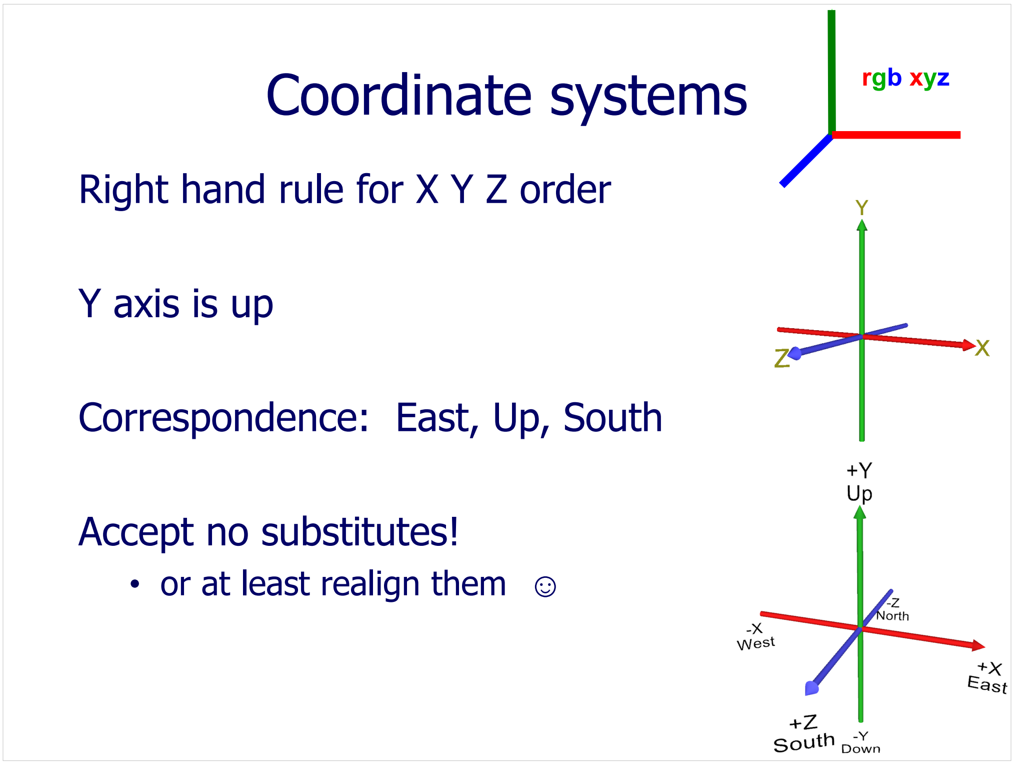
The convention for time units is quite common in computer science and was ultimately inherited from the Unix operating system. Reference: “Unix time.”

Choice of units is strict and consistent throughout: Systeme Internationale (SI), also known as the metric system.

- Potential drawback: sometimes values are in other units of measurement and scaling is required
- Benefit: units are always consistent and implicit, not needing to be defined, thus avoiding a significant potential source of error

TODO: X3D v3.3 is introducing a Units Component to permit the use of different base units for length, angles, mass, speed and acceleration for a given X3D scene.

TODO: it would be helpful to have a time conversion calculator provided in X3D-Edit



Coordinate systems

Right hand rule for X Y Z order

Y axis is up

Correspondence: East, Up, South

Accept no substitutes!

- or at least realign them ☺

See Figures 3.1 and 3.1, page 68, *X3D for Web Authors*

There are a total of eight different Euler angle systems, each with different relative orientations for the X, Y and Z axes.

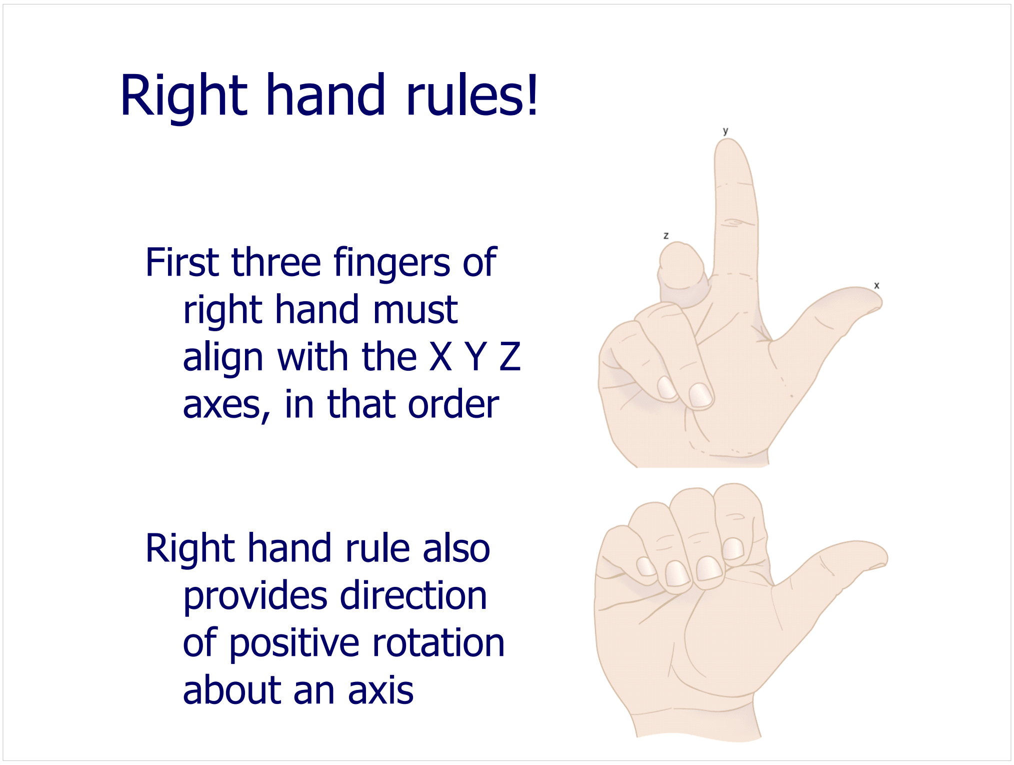
Half of these follow a left-hand rule, rather than a right-hand rule. Occasionally a graphics book comes out that presents mathematical equations using a left-hand rule. Immediately throw such books in the fire so that further pain and suffering is prevented!

The second and third displayed examples are

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/CoordinateAxesNSEW.x3d

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/CoordinateAxes.x3d
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/CoordinateAxesInlineExample.x3d

Ordinarily we ignore correspondences with geographic North, South, East and West, since regular X3D coordinates are single-precision floating point, while the Geospatial nodes use double-precision floating-point values in order to capture latitude and longitude coordinates with sufficient accuracy. The X3D Geospatial Component (and the X3D Earth exemplars) are advanced topics that are not covered in this book.



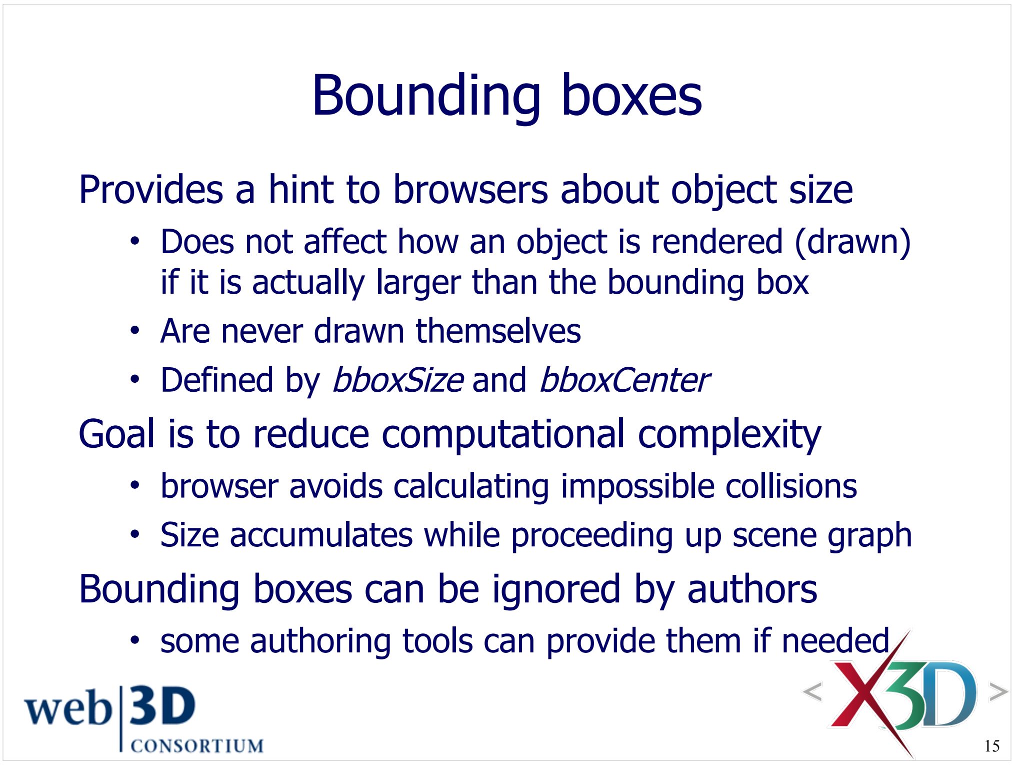
Right hand rules!

- First three fingers of right hand must align with the X Y Z axes, in that order

- Right hand rule also provides direction of positive rotation about an axis

Figures 3.3 and 3.4, pages 69-70, *X3D for Web Authors*

Instructors and students alike should frequently use their right hand to illustrate proper orientation relationships. It is a big help. (Don't worry about onlookers!)



Bounding boxes

Provides a hint to browsers about object size

- Does not affect how an object is rendered (drawn) if it is actually larger than the bounding box
- Are never drawn themselves
- Defined by *bboxSize* and *bboxCenter*

Goal is to reduce computational complexity

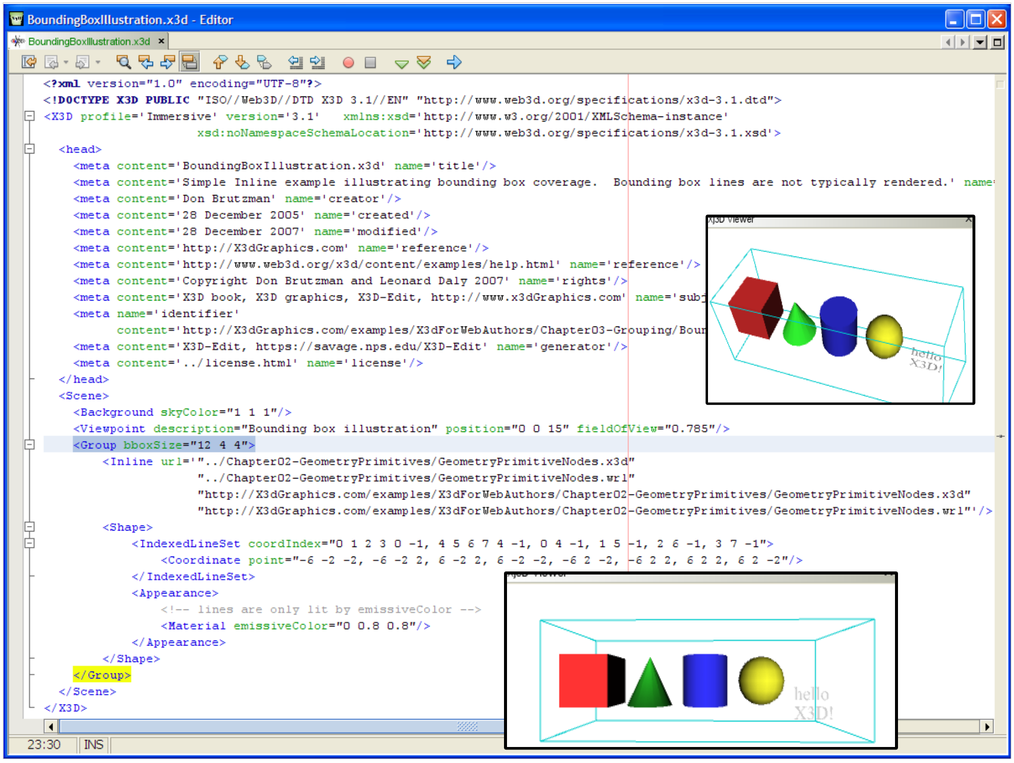
- browser avoids calculating impossible collisions
- Size accumulates while proceeding up scene graph

Bounding boxes can be ignored by authors

- some authoring tools can provide them if needed

Note that bounding boxes are invisible and not displayed.

If defined, bounding box dimensions need to account for all children in the contained scene subgraph.



BoundingBoxIllustration.x3d - Editor

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1'   xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='BoundingBoxIllustration.x3d' name='title'/>
    <meta content='Simple Inline example illustrating bounding box coverage.  Bounding box lines are not typically rendered.' name
    <meta content='Don Brutzman' name='creator'/>
    <meta content='28 December 2005' name='created'/>
    <meta content='28 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subj
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Boun
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor="1 1 1"/>
    <Viewpoint description="Bounding box illustration" position="0 0 15" fieldOfView="0.785"/>
    <Group bboxSize="12 4 4">
      <Inline url='"../Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d"
                   "../Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.wrl"
                   "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d"
                   "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.wrl"'/>
      <Shape>
        <IndexedLineSet coordIndex="0 1 2 3 0 -1, 4 5 6 7 4 -1, 0 4 -1, 1 5 -1, 2 6 -1, 3 7 -1">
          <Coordinate point="-6 -2 -2, -6 -2 2, 6 -2 2, 6 -2 -2, -6 2 -2, -6 2 2, 6 2 2, 6 2 -2"/>
        </IndexedLineSet>
        <Appearance>
          <!-- lines are only lit by emissiveColor -->
          <Material emissiveColor="0 0.8 0.8"/>
        </Appearance>
      </Shape>
    </Group>
  </Scene>
</X3D>
```

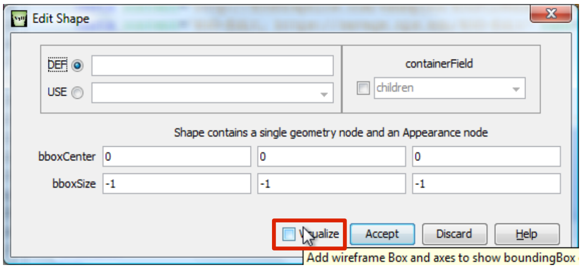
Note that bounding boxes are invisible and not displayed. This wireframe has been explicitly added to this particular scene in order to illustrate bounding box principles. Also note that the bounding values are greater than the minimum possible, and thus a little bit larger than what t tool would compute.

The bounding box illustrated here is somewhat larger than the actual bounding box size.

The IndexedLineSet node is described in Chapter 6 Geometry: Points Lines Polygons.

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/BoundingBoxIllustration.x3d

X3D-Edit now includes visualization options that adds extra geometry to show bounding box dimensions.



Edit Shape

DEF ●

USE ○

containerField: □ children

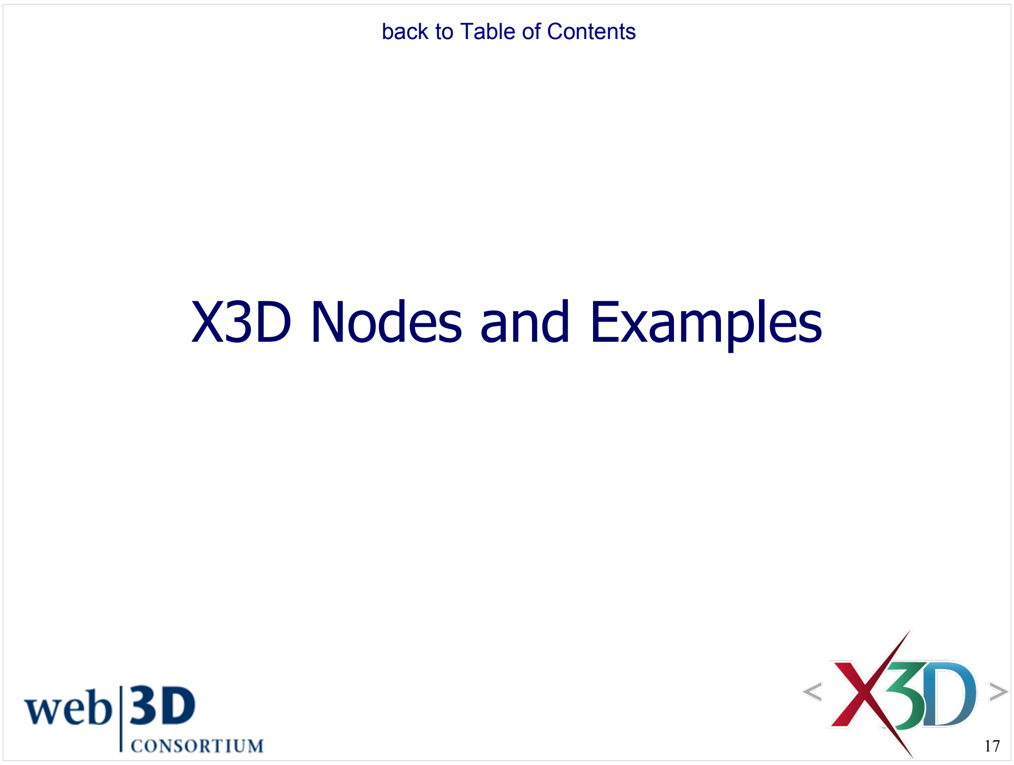
Shape contains a single geometry node and an Appearance node

bboxCenter: 0 | 0 | 0

bboxSize: -1 | -1 | -1

Note that X3D-Edit also enables computation of all bounding box values (using Xj3D) via the menus for:

Conversions => Cad Filter Conversions => Add bounding boxes



back to Table of Contents

X3D Nodes and Examples

Group node

Collects nodes together with related purpose

- Often close to each other spatially

Can make USE copies if a DEF is provided

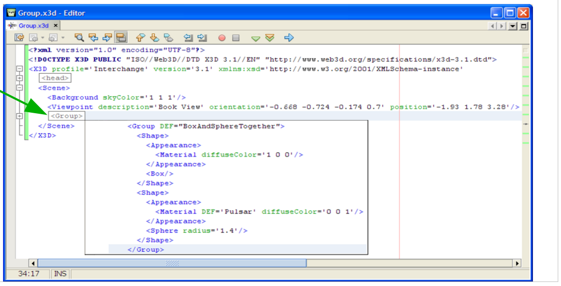
- Example: 4 identical tires in a car model

Simplify editing

- X3D-Edit: collapse node using +,- ticks in margin

X3D-Edit trick

- Mouse-over to show hidden contents
- Helps to organize your work
- Copy by reference

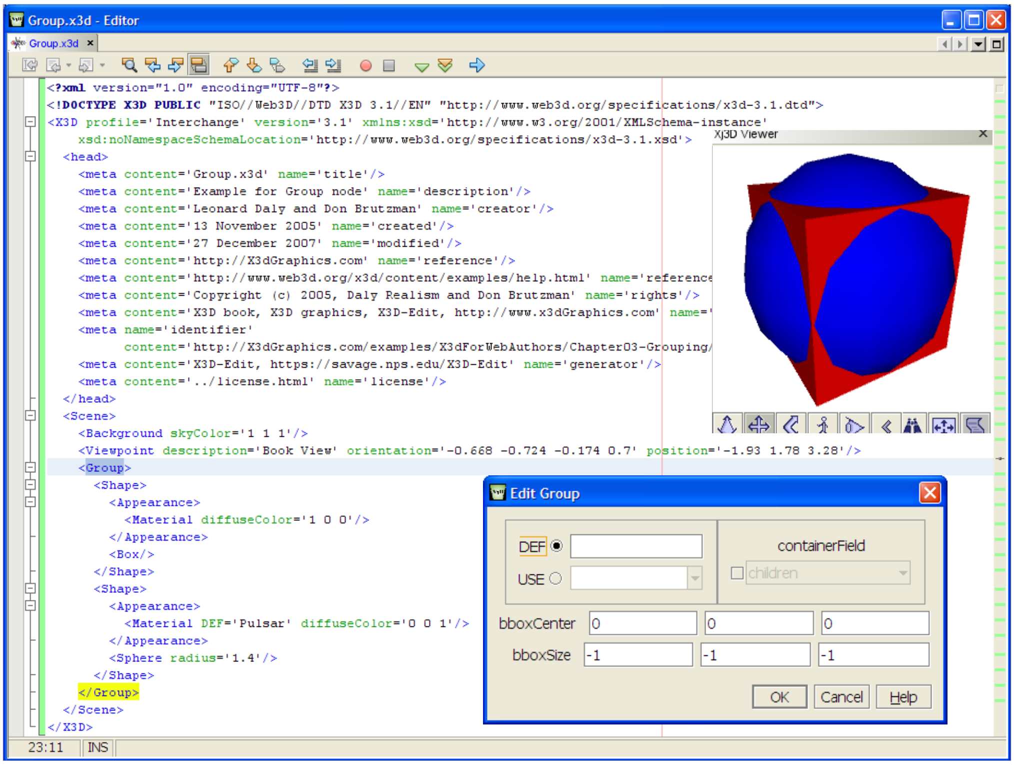


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
  <head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <Group>
  </Scene>
</X3D>
```

```
<Group DEF="BoxAndSphereTogether">
  <Shape>
    <Appearance>
      <Material diffuseColor='1 0 0'/>
    </Appearance>
    <Box/>
  </Shape>
  <Shape>
    <Appearance>
      <Material DEF='Pulsar' diffuseColor='0 0 1'/>
    </Appearance>
    <Sphere radius='1.4'/>
  </Shape>
</Group>
```

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Group.x3d

Hint: DEF and USE are often used with Group nodes that hold repeatable groups of geometry.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='Group.x3d' name='title'/>
    <meta content='Example for Group node' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='13 November 2005' name='created'/>
    <meta content='27 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <Group>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
      <Shape>
        <Appearance>
          <Material DEF='Pulsar' diffuseColor='0 0 1'/>
        </Appearance>
        <Sphere radius='1.4'/>
      </Shape>
    </Group>
  </Scene>
</X3D>
```

Edit Group

DEF ●

USE ○

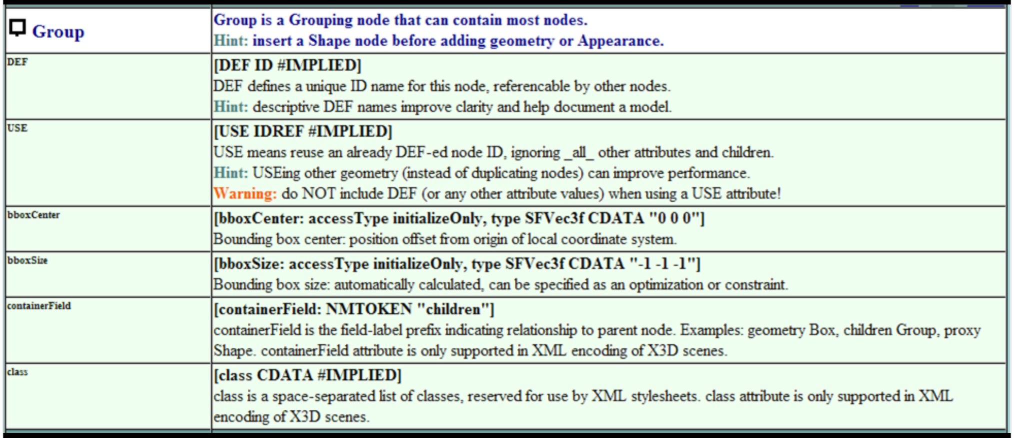
containerField

□ children

bboxCenter	0	0	0
bboxSize	-1	-1	-1

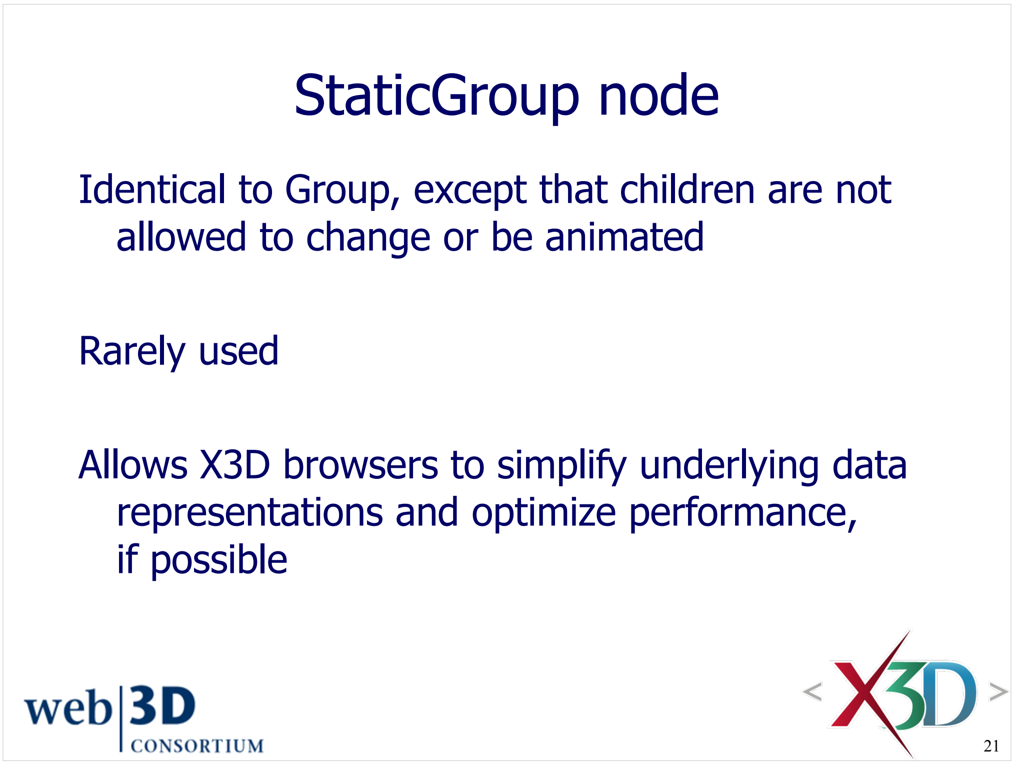
Figure 3.5, page 76, *X3D for Web Authors*

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Group.x3d



◻ Group	**Group is a Grouping node that can contain most nodes.** **Hint: insert a Shape node before adding geometry or Appearance.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

http://www.web3d.org/x3d/content/X3dTooltips.html#Group

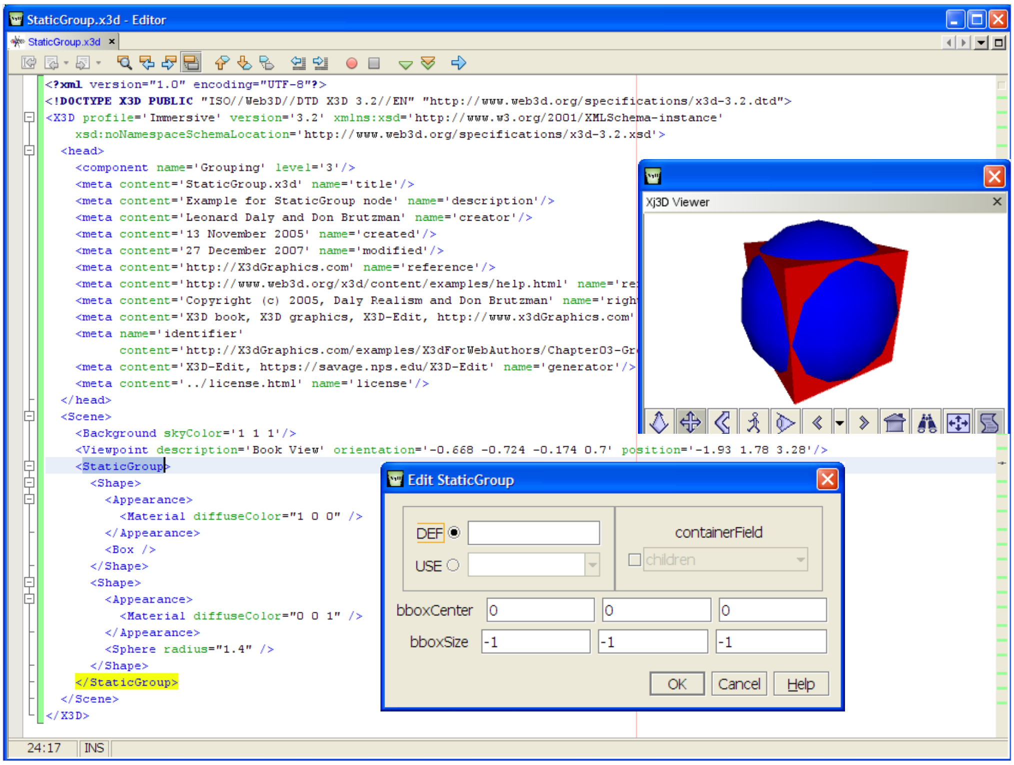


StaticGroup node

Identical to Group, except that children are not allowed to change or be animated

Rarely used

Allows X3D browsers to simplify underlying data representations and optimize performance, if possible



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.2.xsd'>
  <head>
    <component name='Grouping' level='3'/>
    <meta content='StaticGroup.x3d' name='title'/>
    <meta content='Example for StaticGroup node' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='13 November 2005' name='created'/>
    <meta content='27 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='re
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='righ
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com'
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Gr
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <StaticGroup>
      <Shape>
        <Appearance>
          <Material diffuseColor="1 0 0" />
        </Appearance>
        <Box />
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor="0 0 1" />
        </Appearance>
        <Sphere radius="1.4" />
      </Shape>
    </StaticGroup>
  </Scene>
</X3D>
```

Edit StaticGroup

DEF ●

USE ○

containerField ☐ children

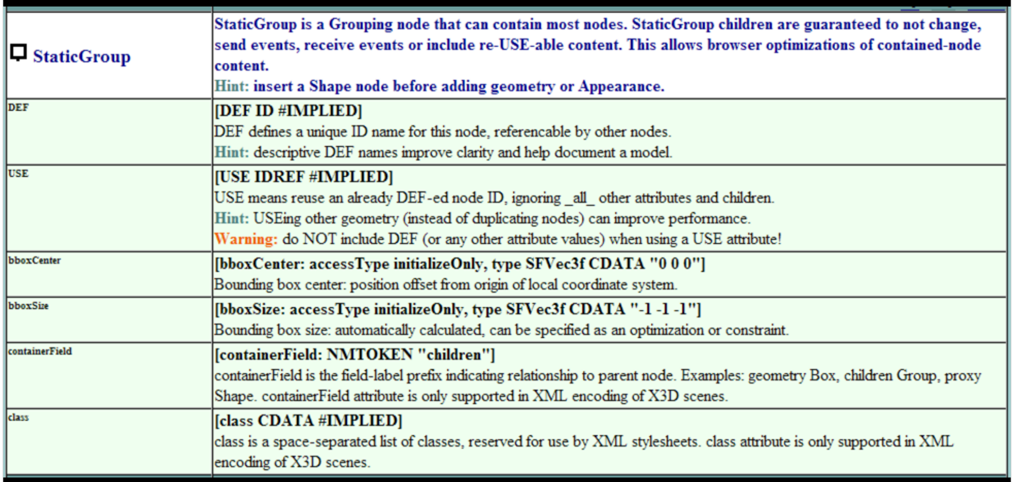
bboxCenter 0 0 0

bboxSize -1 -1 -1

Figure 3.5, page 76, *X3D for Web Authors*

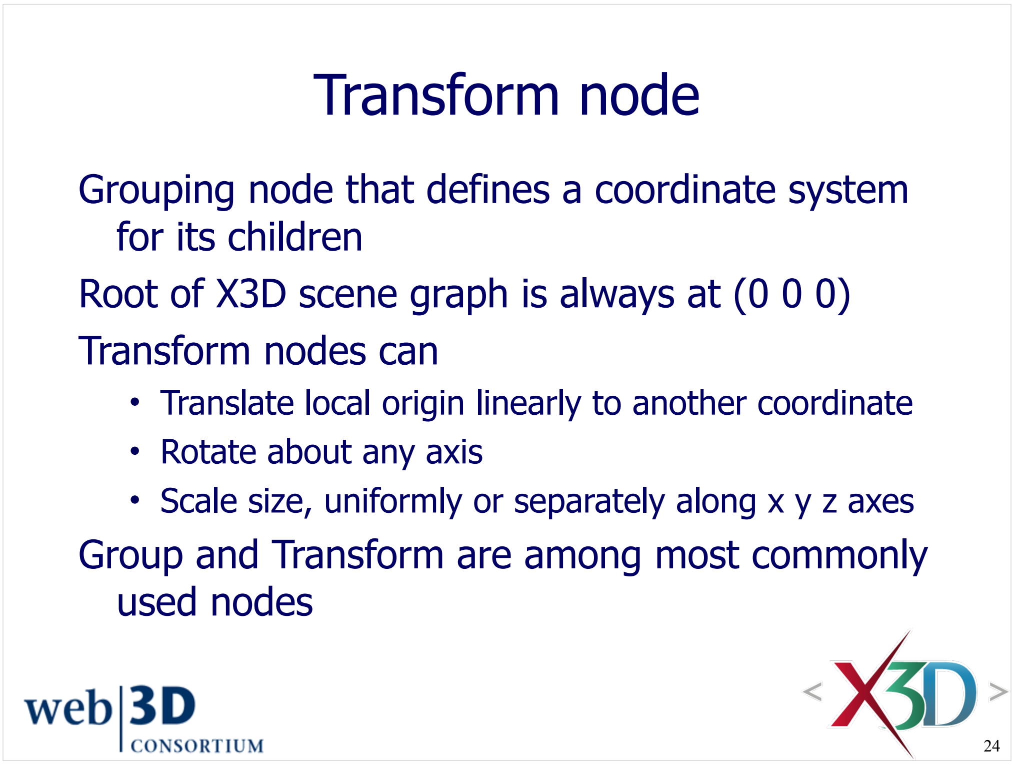
http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/StaticGroup.x3d

Note that a Group might be used identically here instead of a StaticGroup.



StaticGroup	**StaticGroup is a Grouping node that can contain most nodes. StaticGroup children are guaranteed to not change, send events, receive events or include re-USE-able content. This allows browser optimizations of contained-node content.** **Hint: insert a Shape node before adding geometry or Appearance.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. **Hint:** descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. **Hint:** USEing other geometry (instead of duplicating nodes) can improve performance. **Warning:** do NOT include DEF (or any other attribute values) when using a USE attribute!
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

http://www.web3d.org/x3d/content/X3dTooltips.html#StaticGroup



Transform node

Grouping node that defines a coordinate system for its children

Root of X3D scene graph is always at (0 0 0)

Transform nodes can

- Translate local origin linearly to another coordinate
- Rotate about any axis
- Scale size, uniformly or separately along x y z axes

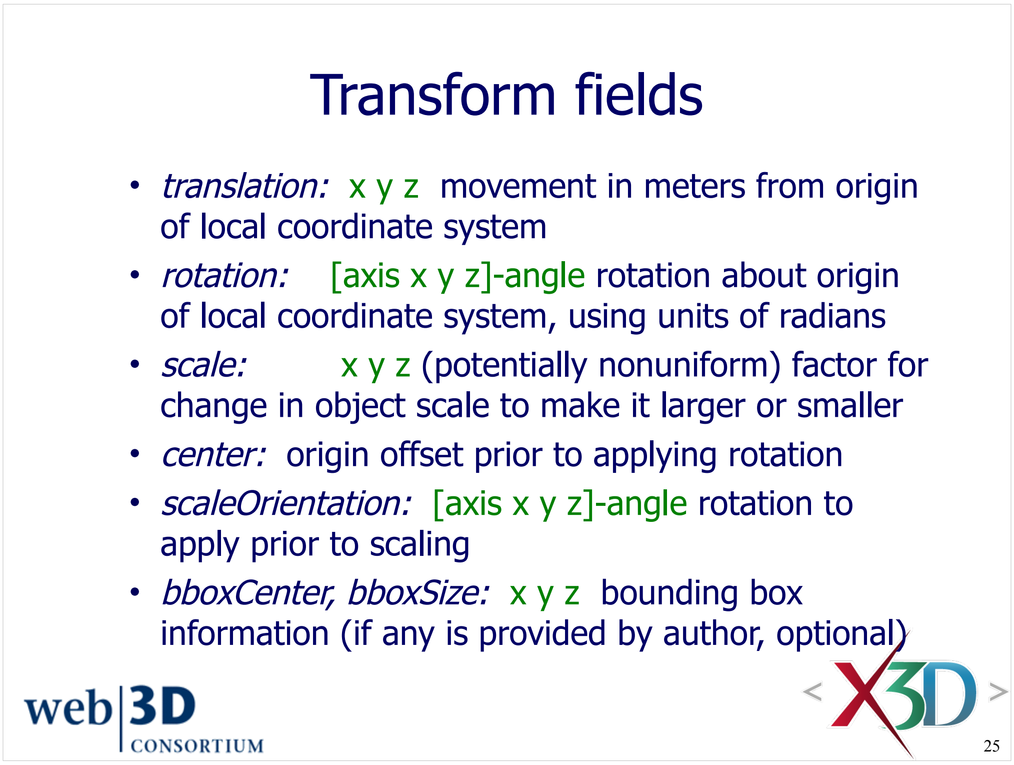
Group and Transform are among most commonly used nodes

web|3D CONSORTIUM

<X3D>

24

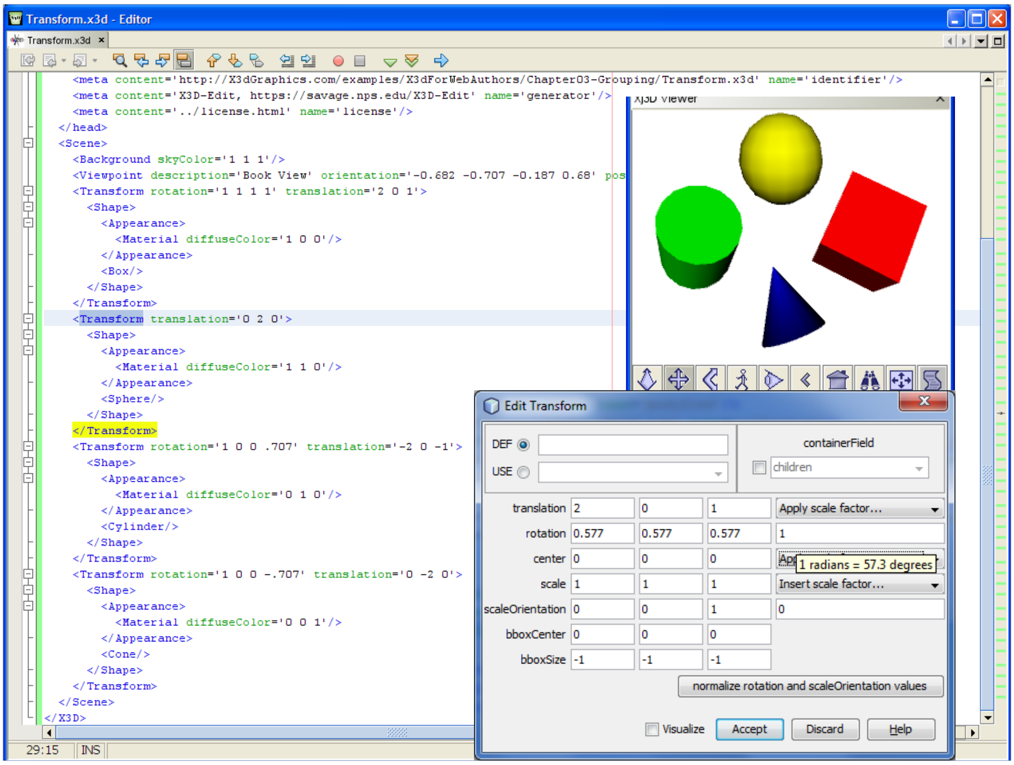
Transform is perhaps the most fundamental of all X3D nodes.



Transform fields

- *translation:* x y z movement in meters from origin of local coordinate system
- *rotation:* [axis x y z]-angle rotation about origin of local coordinate system, using units of radians
- *scale:* x y z (potentially nonuniform) factor for change in object scale to make it larger or smaller
- *center:* origin offset prior to applying rotation
- *scaleOrientation:* [axis x y z]-angle rotation to apply prior to scaling
- *bboxCenter, bboxSize:* x y z bounding box information (if any is provided by author, optional)

TODO picture or animated scene to illustrate rotation?



Transform.x3d - Editor

```
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Transform.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.682 -0.707 -0.187 0.68' pos
    <Transform rotation='1 1 1 1' translation='2 0 1'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
    </Transform>
    <Transform translation='0 2 0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 1 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
    <Transform rotation='1 0 0 .707' translation='-2 0 -1'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 1 0'/>
        </Appearance>
        <Cylinder/>
      </Shape>
    </Transform>
    <Transform rotation='1 0 0 -.707' translation='0 -2 0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 0 1'/>
        </Appearance>
        <Cone/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

29:15 INS

Edit Transform

DEF ● USE ○

containerField □ children

translation	2	0	1	Apply scale factor...
rotation	0.577	0.577	0.577	1
center	0	0	0	App 1 radians = 57.3 degrees
scale	1	1	1	Insert scale factor...
scaleOrientation	0	0	1	0
bboxCenter	0	0	0	
bboxSize	-1	-1	-1	

Figure 3.6, page 79, *X3D for Web Authors*

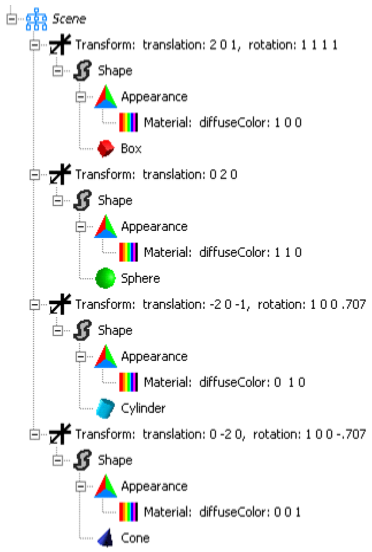
http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Transform.x3d

Note that X3D-Edit includes a tooltip for radian fields that shows angle values in degrees.

X3D-Edit data-entry trick: entered values with a magnitude greater than 6.28 (i.e. 2 Π) prompt the user to confirm if the value is actually degrees instead of radians.

- Can then convert from degrees to correct radian value easily
- Can also support smaller values by adding 360°, e.g. 363° for 3°

Each Transform is a scene subgraph



- Scene
  - Transform: translation: 2 0 1, rotation: 1 1 1 1
    - Shape
      - Appearance
        - Material: diffuseColor: 1 0 0
      - Box
  - Transform: translation: 0 2 0
    - Shape
      - Appearance
        - Material: diffuseColor: 1 1 0
      - Sphere
  - Transform: translation: -2 0 -1, rotation: 1 0 0 .707
    - Shape
      - Appearance
        - Material: diffuseColor: 0 1 0
      - Cylinder
  - Transform: translation: 0 -2 0, rotation: 1 0 0 -.707
    - Shape
      - Appearance
        - Material: diffuseColor: 0 0 1
      - Cone

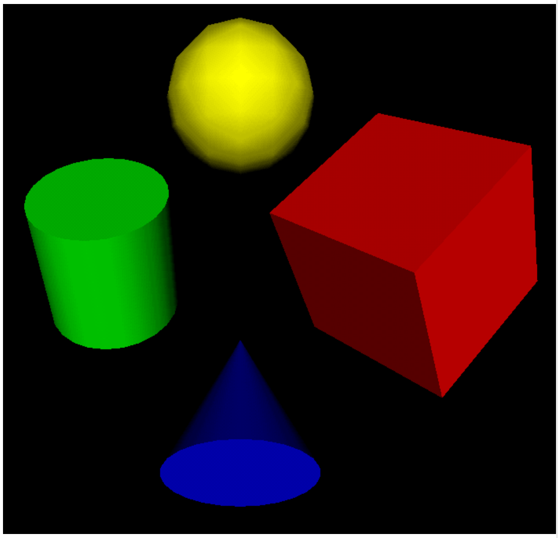
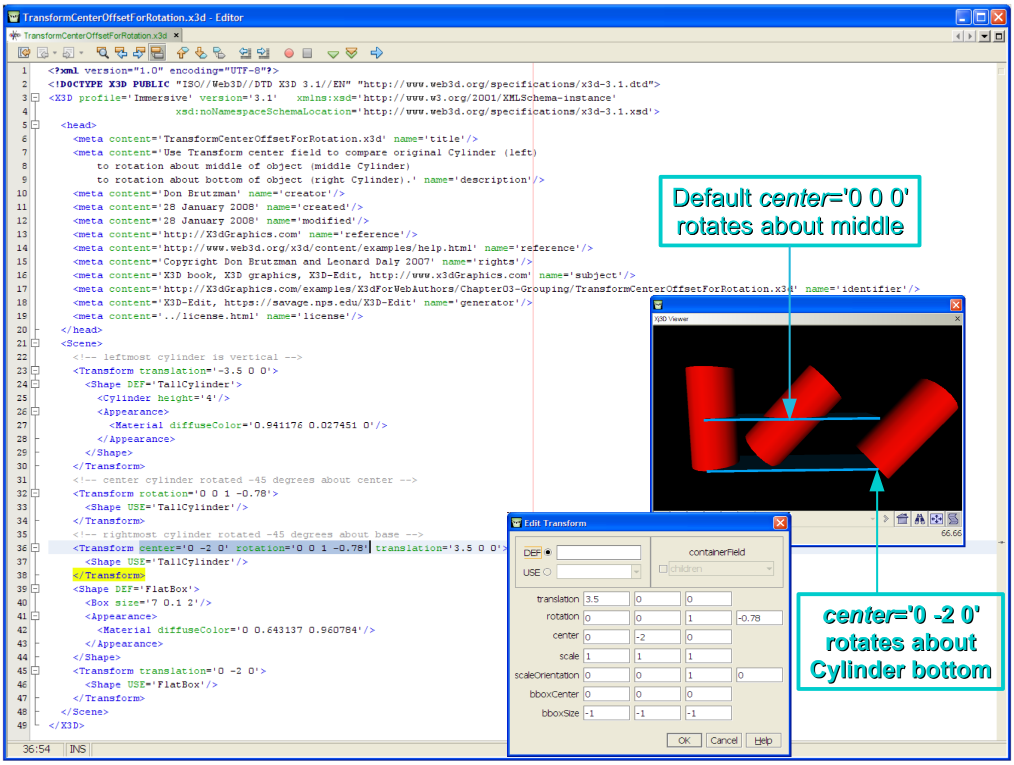


Figure 3.6, page 79, *X3D for Web Authors*

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Transform.x3d



TransformCenterOffsetForRotation.x3d - Editor
TransformCenterOffsetForRotation.x3d
Xj3D Viewer
66.66
36:54
INS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1'   xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='TransformCenterOffsetForRotation.x3d' name='title'/>
    <meta content='Use Transform center field to compare original Cylinder (left)
        to rotation about middle of object (middle Cylinder)
        to rotation about bottom of object (right Cylinder).' name='description'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='28 January 2008' name='created'/>
    <meta content='28 January 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/TransformCenterOffsetForRotation.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <!-- leftmost cylinder is vertical -->
    <Transform translation='-3.5 0 0'>
      <Shape DEF='TallCylinder'>
        <Cylinder height='4'/>
        <Appearance>
          <Material diffuseColor='0.941176 0.027451 0'/>
        </Appearance>
      </Shape>
    </Transform>
    <!-- center cylinder rotated -45 degrees about center -->
    <Transform rotation='0 0 1 -0.78'>
      <Shape USE='TallCylinder'/>
    </Transform>
    <!-- rightmost cylinder rotated -45 degrees about base -->
    <Transform center='0 -2 0' rotation='0 0 1 -0.78' translation='3.5 0 0'>
      <Shape USE='TallCylinder'/>
    </Transform>
    <Shape DEF='FlatBox'>
      <Box size='7 0.1 2'/>
      <Appearance>
        <Material diffuseColor='0 0.643137 0.960784'/>
      </Appearance>
    </Shape>
    <Transform translation='0 -2 0'>
      <Shape USE='FlatBox'/>
    </Transform>
  </Scene>
</X3D>
```

Edit Transform

DEF ●

USE ○

containerField ☐ children

translation	3.5	0	0	
rotation	0	0	1	-0.78
center	0	-2	0	
scale	1	1	1	
scaleOrientation	0	0	1	0
bboxCenter	0	0	0	
bboxSize	-1	-1	-1	

Default *center*='0 0 0' rotates about middle

center='0 -2 0' rotates about Cylinder bottom

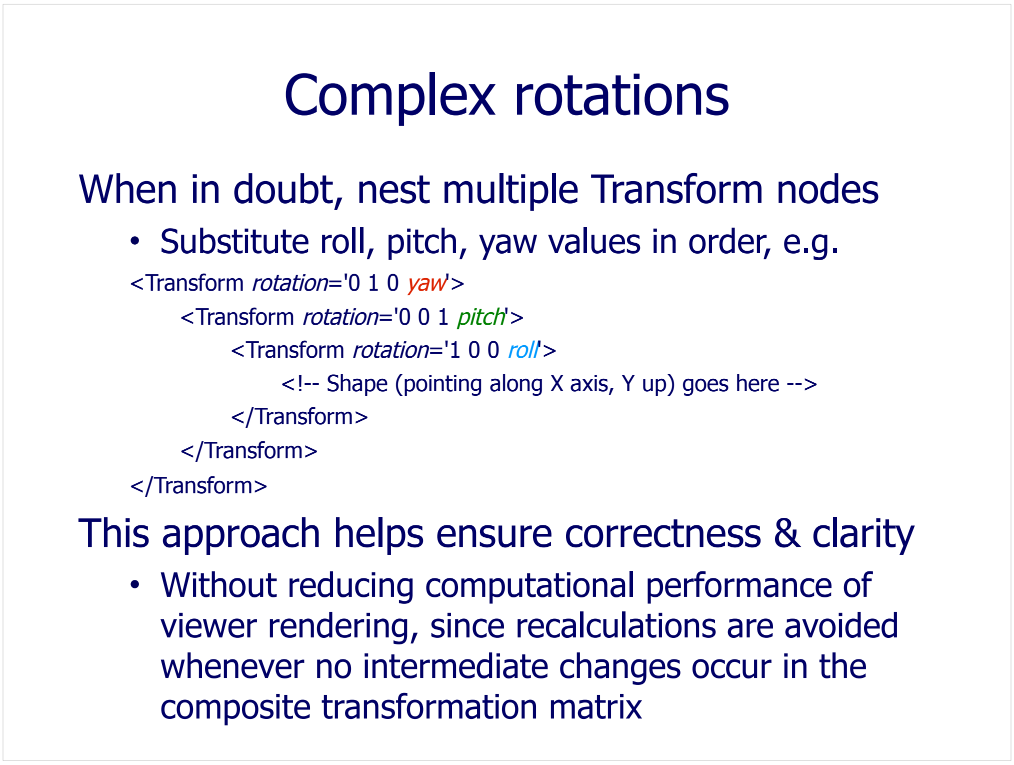
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/TransformCenterOffsetForRotation.x3d

Note the three Cylinders in this example. Each has a height of 4m which aligns with Y axis, ± 2m above/below Y=0.

- Left Cylinder is not rotated
- Middle Cylinder is rotated about its default *center*='0 0 0'
- Right Cylinder is rotated about its bottom, *center*='0 -2 0'

Note that *rotation*='0 0 1 0.78' doesn't have to be too mysterious either.

- The '0 0 1' vector value means rotate about the Z axis, which is coming out of the page
- The '0.78' radians angle can be figured out as well:
  - 360º = 2π ≈ 6.28, then
  - 180º = π   ≈ 3.14
  -   90º = π/2 ≈ 1.57
  -   45º = π/4 ≈ 0.78, which matches the 45º rotation shown in the figure above



Complex rotations

When in doubt, nest multiple Transform nodes

- Substitute roll, pitch, yaw values in order, e.g.

```
<Transform rotation='0 1 0 yaw'>
    <Transform rotation='0 0 1 pitch'>
        <Transform rotation='1 0 0 roll'>
            <!-- Shape (pointing along X axis, Y up) goes here -->
        </Transform>
    </Transform>
</Transform>
```

This approach helps ensure correctness & clarity

- Without reducing computational performance of viewer rendering, since recalculations are avoided whenever no intermediate changes occur in the composite transformation matrix

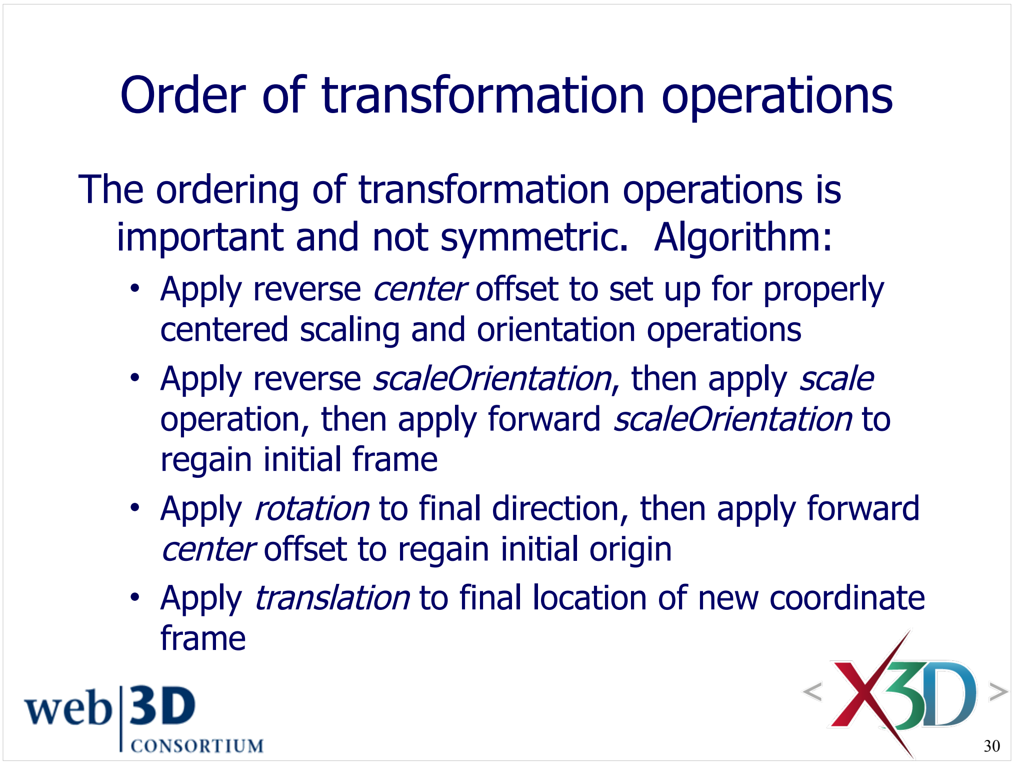
Assuming model shapes are oriented to point along the local X axis, with Y axis up: the following Transform rotations are performed from the innermost to the outermost

- First roll about the nose direction, which is the local X axis
- Next pitch up/down about the local Z axis, which points out right side of model
- Finally yaw to match heading direction about local Y axis

Note that all rotations are following the right-hand rule. Thus you must carefully check positive/negative directions of rotation for the roll/pitch/yaw values you want to apply.

See the DIS gimbals slide under Additional Resources for an advanced example.

TODO: addition of a rotation calculator for X3D-Edit that converts multiple rotations about different axes into a single SFRotation (and vice versa). There used to be the Vapour tools for Windows, though these are no longer publicly available.



Order of transformation operations

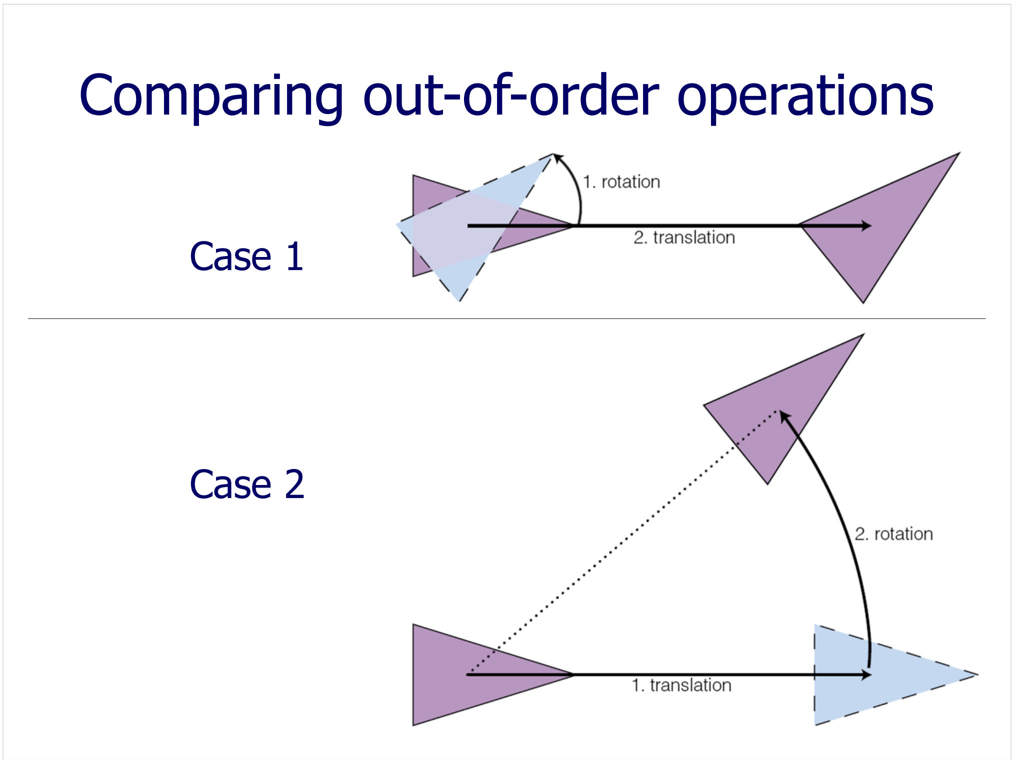
The ordering of transformation operations is important and not symmetric. Algorithm:

- Apply reverse *center* offset to set up for properly centered scaling and orientation operations
- Apply reverse *scaleOrientation*, then apply *scale* operation, then apply forward *scaleOrientation* to regain initial frame
- Apply *rotation* to final direction, then apply forward *center* offset to regain initial origin
- Apply *translation* to final location of new coordinate frame

30

The next slide illustrates these steps.

Lower-level computer graphics interfaces often employ the concept of 4x4 transformation matrices, which are described on the Matrix operations slide.



Comparing out-of-order operations

Case 1

Case 2

Figure 3.7, page 80, *X3D for Web Authors*

Case 1: first rotation, then translation. (Requires one Transform node in X3D)

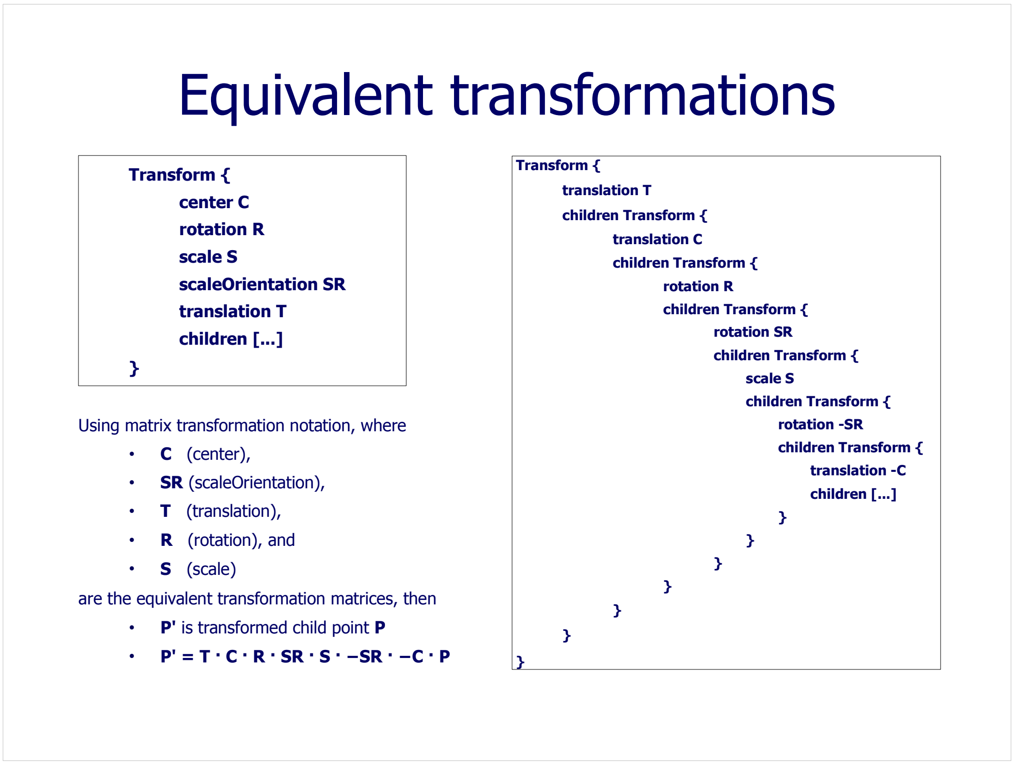
Case 2: first translation, then rotation. (Requires two Transform nodes in X3D)

The intermediate steps (blue triangle) are not displayed when rendering a 3D scene.

Results (the second purple triangle) are not equivalent. Thus the application of transformation steps (scale, rotation, translation) are order dependent.

Case 1 corresponds to the way that a single X3D Transform node works: first rotation, then translation.

Case 2 is also possible, but requires two Transform nodes to apply steps in the order desired.



Equivalent transformations

```
Transform {
        center C
        rotation R
        scale S
        scaleOrientation SR
        translation T
        children [...]
}
```

Using matrix transformation notation, where

- **C** (center),
- **SR** (scaleOrientation),
- **T** (translation),
- **R** (rotation), and
- **S** (scale)

are the equivalent transformation matrices, then

- **P'** is transformed child point **P**
- **P' = T · C · R · SR · S · −SR · −C · P**

```
Transform {
    translation T
    children Transform {
        translation C
        children Transform {
            rotation R
            children Transform {
                rotation SR
                children Transform {
                    scale S
                    children Transform {
                        rotation -SR
                        children Transform {
                            translation -C
                            children [...]
                        }
                    }
                }
            }
        }
    }
}
```

Figure 3.8, page 81, *X3D for Web Authors*

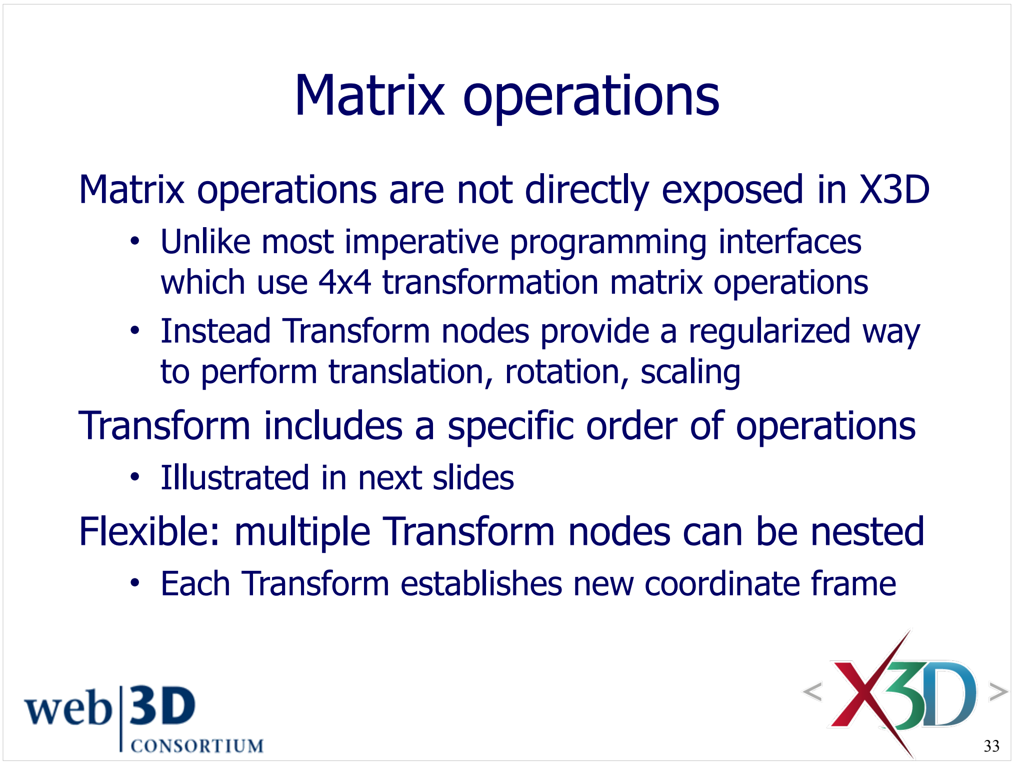
The Transform on the left is equivalent to the set of Transform nodes on the right.

Most 3D graphics programming languages are more complicated than X3D in this respect, requiring the author to carefully apply matrix algebra to transformation matrices.

The way to read the governing matrix equation at the bottom left corner is from right to left. The order of operations is strictly defined for a single Transform node.

Summary: first apply center and scaling operations, then rotation, then translation.

If you really want to perform these operations in a different order than X3D, so that it matches some other matrix-operations source code, then use multiple nested X3D Transform nodes.



Matrix operations

Matrix operations are not directly exposed in X3D

- Unlike most imperative programming interfaces which use 4x4 transformation matrix operations
- Instead Transform nodes provide a regularized way to perform translation, rotation, scaling

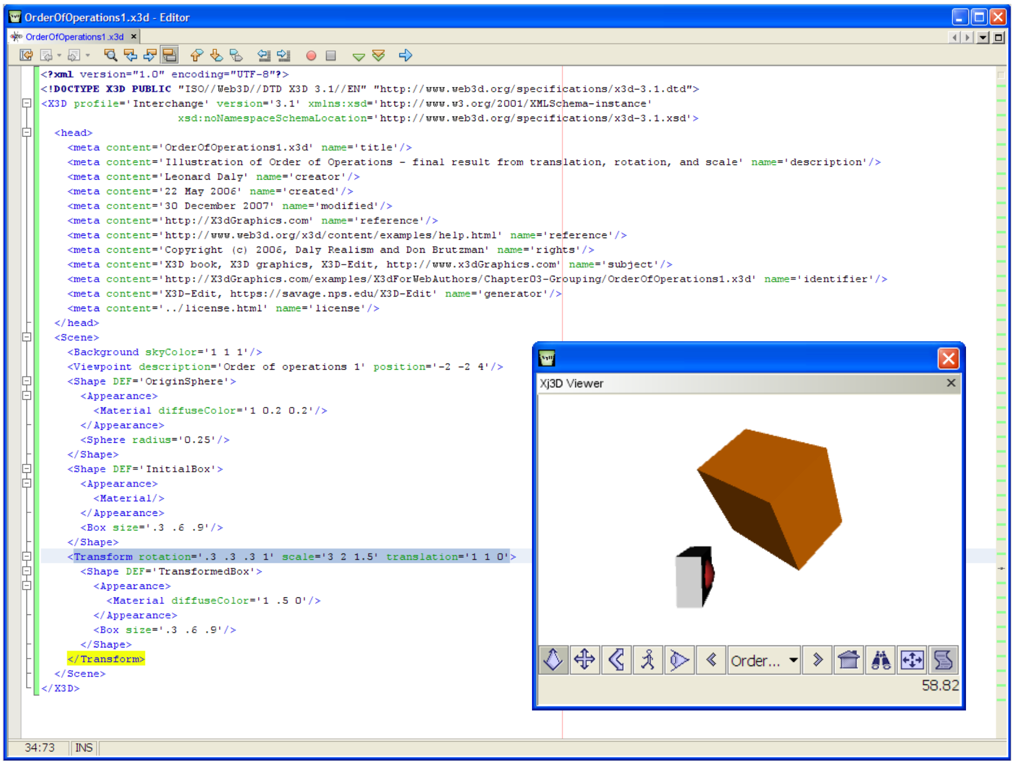
Transform includes a specific order of operations

- Illustrated in next slides

Flexible: multiple Transform nodes can be nested

- Each Transform establishes new coordinate frame

Advanced topic: matrix operations are exposed in the Scene Access Interface (SAI) application programming interface (API) for X3D. Nevertheless these are provided as a programming convenience for classical algorithms and rarely used.

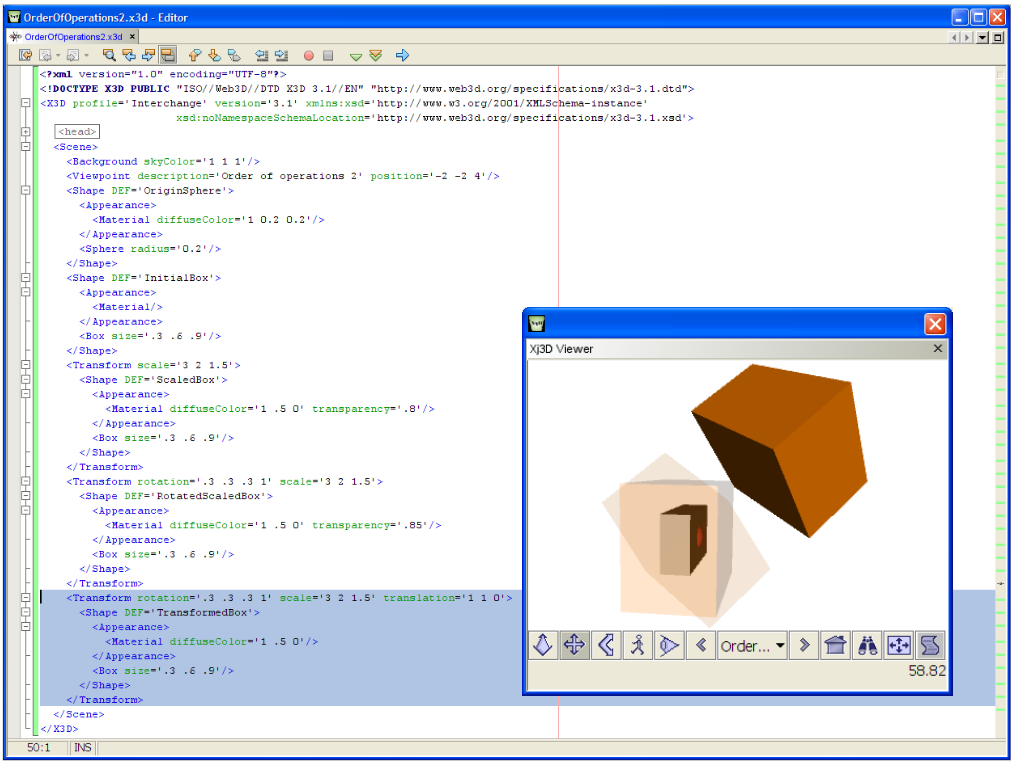


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='OrderOfOperations1.x3d' name='title'/>
    <meta content='Illustration of Order of Operations - final result from translation, rotation, and scale' name='description'/>
    <meta content='Leonard Daly' name='creator'/>
    <meta content='22 May 2006' name='created'/>
    <meta content='30 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright (c) 2006, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/OrderOfOperations1.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Order of operations 1' position='-2 -2 4'/>
    <Shape DEF='OriginSphere'>
      <Appearance>
        <Material diffuseColor='1 0.2 0.2'/>
      </Appearance>
      <Sphere radius='0.25'/>
    </Shape>
    <Shape DEF='InitialBox'>
      <Appearance>
        <Material/>
      </Appearance>
      <Box size='.3 .6 .9'/>
    </Shape>
    <Transform rotation='.3 .3 .3 1' scale='3 2 1.5' translation='1 1 0'>
      <Shape DEF='TransformedBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

The next two scenes illustrate the combined effect of scaling, rotation and translation operations. In this scene, we see all three operations performed at once by a single Transform node. Using a single Transform node is the most common way to perform this task.

The small red sphere shows the origin of the local coordinate reference frame.

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/OrderOfOperations1.x3d



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                         xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Order of operations 2' position='-2 -2 4'/>
    <Shape DEF='OriginSphere'>
      <Appearance>
        <Material diffuseColor='1 0.2 0.2'/>
      </Appearance>
      <Sphere radius='0.2'/>
    </Shape>
    <Shape DEF='InitialBox'>
      <Appearance>
        <Material/>
      </Appearance>
      <Box size='.3 .6 .9'/>
    </Shape>
    <Transform scale='3 2 1.5'>
      <Shape DEF='ScaledBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0' transparency='.8'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
    <Transform rotation='.3 .3 .3 1' scale='3 2 1.5'>
      <Shape DEF='RotatedScaledBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0' transparency='.85'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
    <Transform rotation='.3 .3 .3 1' scale='3 2 1.5' translation='1 1 0'>
      <Shape DEF='TransformedBox'>
        <Appearance>
          <Material diffuseColor='1 .5 0'/>
        </Appearance>
        <Box size='.3 .6 .9'/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

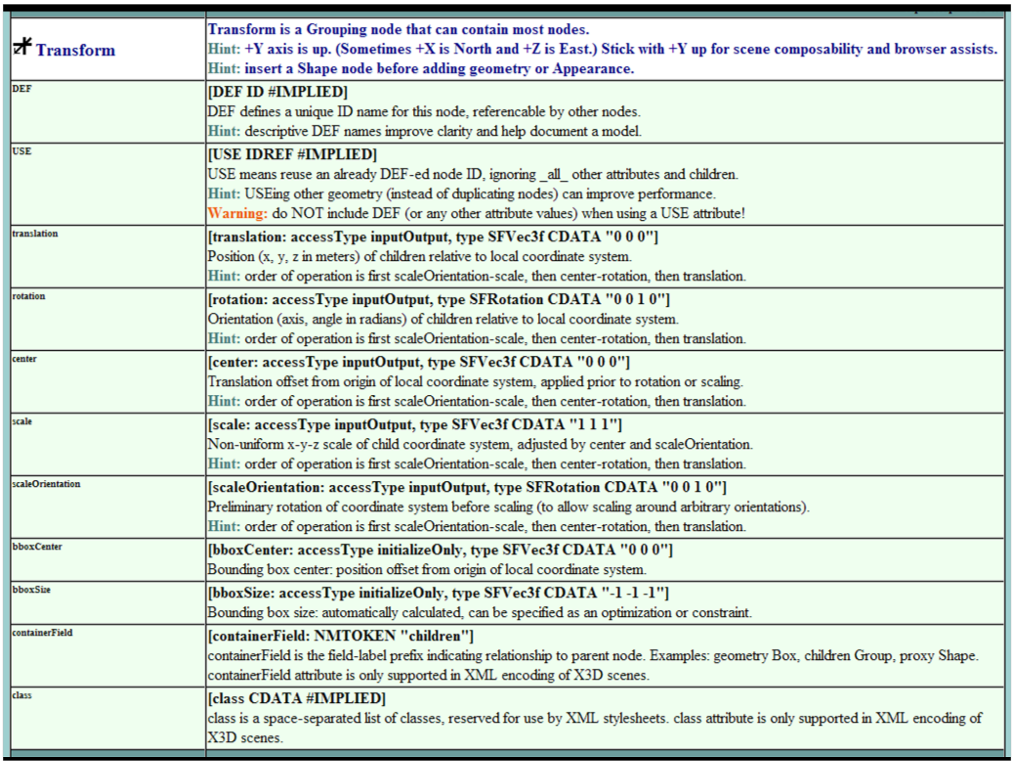
This second scene illustrates the effects of sequentially transforming (scaling, rotating, translating) an object.

(a) shows the original Box
(b) first we scale the object,
(c) then rotate it,
(d) then translate it.

The effect of these three sequential Transform operations matches the combined Transform in the preceding scene.

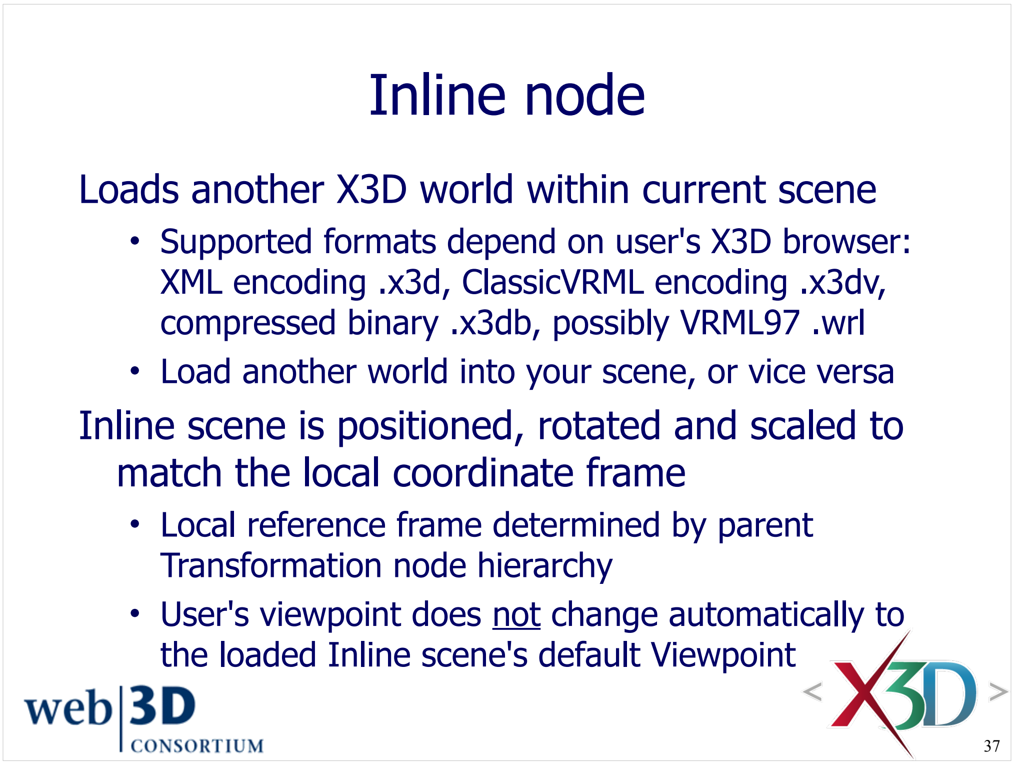
The small red sphere shows the origin of the local coordinate reference frame.

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/OrderOfOperations2.x3d



Transform	Transform is a Grouping node that can contain most nodes. Hint: +Y axis is up. (Sometimes +X is North and +Z is East.) Stick with +Y up for scene composability and browser assists. Hint: insert a Shape node before adding geometry or Appearance.
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
translation	**[translation: accessType inputOutput, type SFVec3f CDATA "0 0 0"]** Position (x, y, z in meters) of children relative to local coordinate system. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
rotation	**[rotation: accessType inputOutput, type SFRotation CDATA "0 0 1 0"]** Orientation (axis, angle in radians) of children relative to local coordinate system. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
center	**[center: accessType inputOutput, type SFVec3f CDATA "0 0 0"]** Translation offset from origin of local coordinate system, applied prior to rotation or scaling. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
scale	**[scale: accessType inputOutput, type SFVec3f CDATA "1 1 1"]** Non-uniform x-y-z scale of child coordinate system, adjusted by center and scaleOrientation. Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
scaleOrientation	**[scaleOrientation: accessType inputOutput, type SFRotation CDATA "0 0 1 0"]** Preliminary rotation of coordinate system before scaling (to allow scaling around arbitrary orientations). Hint: order of operation is first scaleOrientation-scale, then center-rotation, then translation.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

http://www.web3d.org/x3d/content/X3dTooltips.html#Transform



Inline node

Loads another X3D world within current scene

- Supported formats depend on user's X3D browser: XML encoding .x3d, ClassicVRML encoding .x3dv, compressed binary .x3db, possibly VRML97 .wrl
- Load another world into your scene, or vice versa

Inline scene is positioned, rotated and scaled to match the local coordinate frame

- Local reference frame determined by parent Transformation node hierarchy
- User's viewpoint does not change automatically to the loaded Inline scene's default Viewpoint

web3D CONSORTIUM

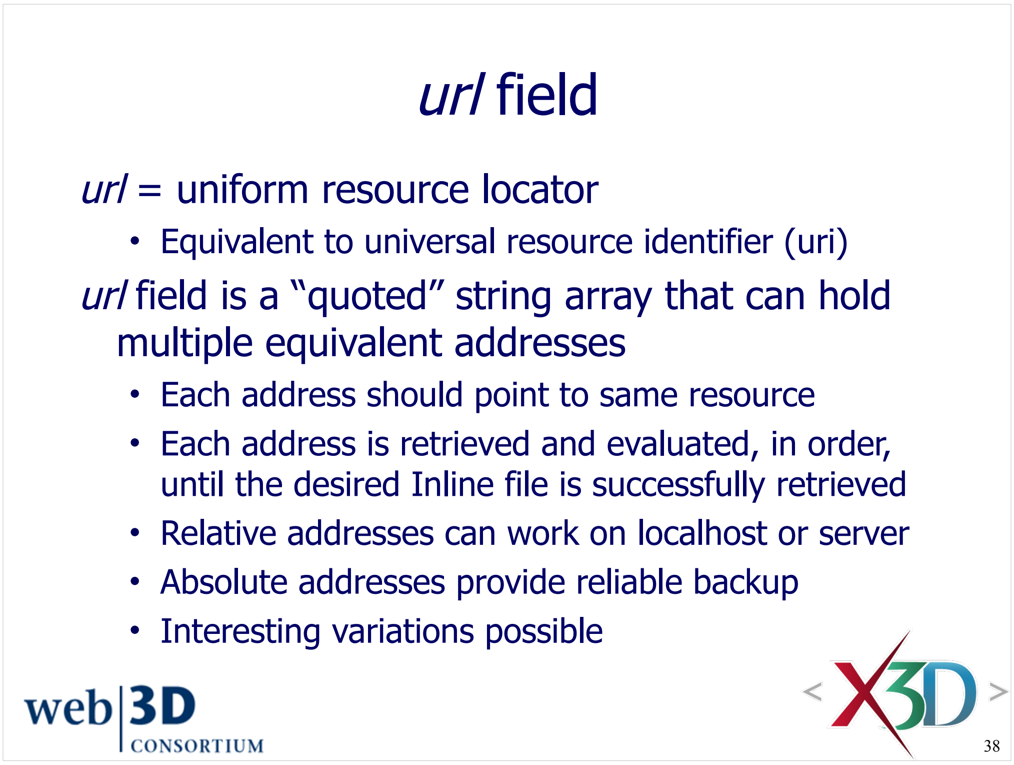
X3D

37

Using Inline nodes is a good way to quickly produce large worlds populated with many models.

The X3D model archives are available for any use, offered by NPS MOVES Institute and the Web3D Consortium under an open-source license.

Note that IMPORT/EXPORT statements allows an author to ROUTE values between the parent scene and the sub-scene contained by an Inline node. Event routing and IMPORT/EXPORT are covered further in Chapter 7 Event Animation and Interpolation, as well as in Chapter 9 Event Utilities and Scripting.



url field

url = uniform resource locator

- Equivalent to universal resource identifier (uri)

url field is a “quoted” string array that can hold multiple equivalent addresses

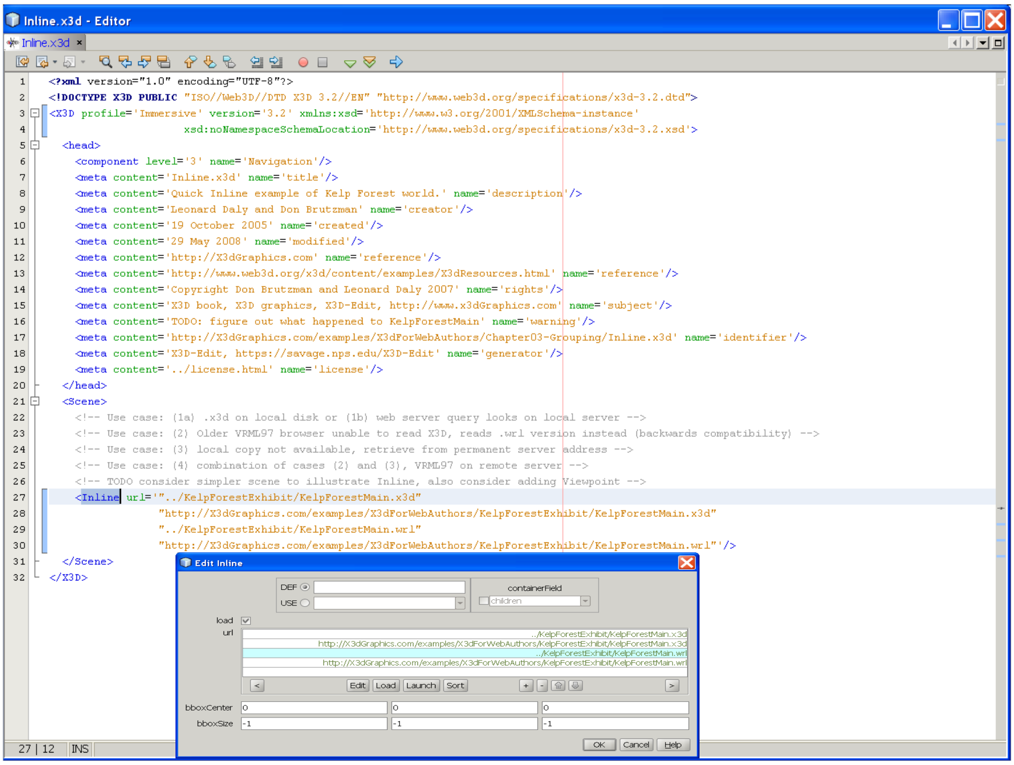
- Each address should point to same resource
- Each address is retrieved and evaluated, in order, until the desired Inline file is successfully retrieved
- Relative addresses can work on localhost or server
- Absolute addresses provide reliable backup
- Interesting variations possible

This approach to *url* -- having an ordered list of legal values – can be considered more robust and more capable that the HTML approach of having just a single *url* value. If the first *url* value fails, the X3D scene keeps on trying the rest until it finds one that works.

The *url* field is also used by a number of other nodes, such as ImageTexture and MovieTexture in Chapter 5.   Functionality for the *url* field is formally defined by X3DUrlObject type, which is defined in the Networking component of the X3D Abstract Specification.

Example variations and benefits provided by url ordering:

- Preferentially pick online (latest greatest) version first, otherwise fall back to local
- Allow scenes to work even when in isolation, for example as email attachment



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.2.xsd'>
  <head>
    <component level='3' name='Navigation'/>
    <meta content='Inline.x3d' name='title'/>
    <meta content='Quick Inline example of Kelp Forest world.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='19 October 2005' name='created'/>
    <meta content='29 May 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/X3dResources.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='TODO: figure out what happened to KelpForestMain' name='warning'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Inline.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <!-- Use case: (1a) .x3d on local disk or (1b) web server query looks on local server -->
    <!-- Use case: (2) Older VRML97 browser unable to read X3D, reads .wrl version instead (backwards compatibility) -->
    <!-- Use case: (3) local copy not available, retrieve from permanent server address -->
    <!-- Use case: (4) combination of cases (2) and (3), VRML97 on remote server -->
    <!-- TODO consider simpler scene to illustrate Inline, also consider adding Viewpoint -->
    <Inline url='"../KelpForestExhibit/KelpForestMain.x3d"
             "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d"
             "../KelpForestExhibit/KelpForestMain.wrl"
             "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.wrl"'/>
  </Scene>
</X3D>
```

Edit Inline

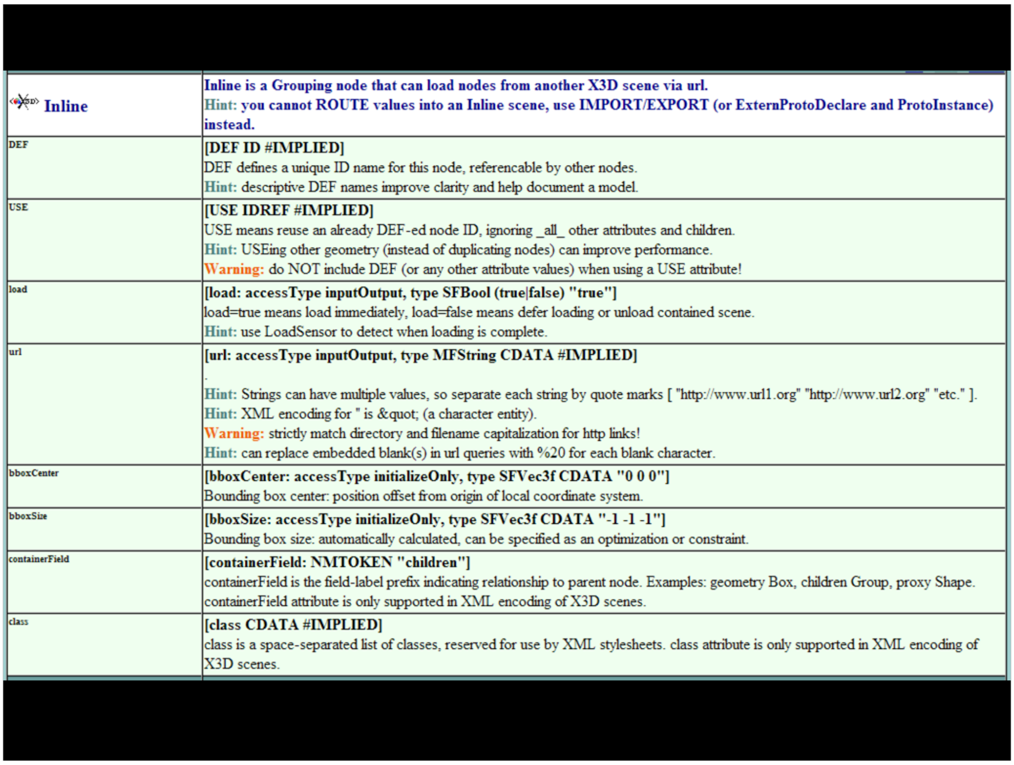
DEF ●
USE ○
containerField ☐ children

load ☑

url
../KelpForestExhibit/KelpForestMain.x3d
http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d
../KelpForestExhibit/KelpForestMain.wrl
http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.wrl

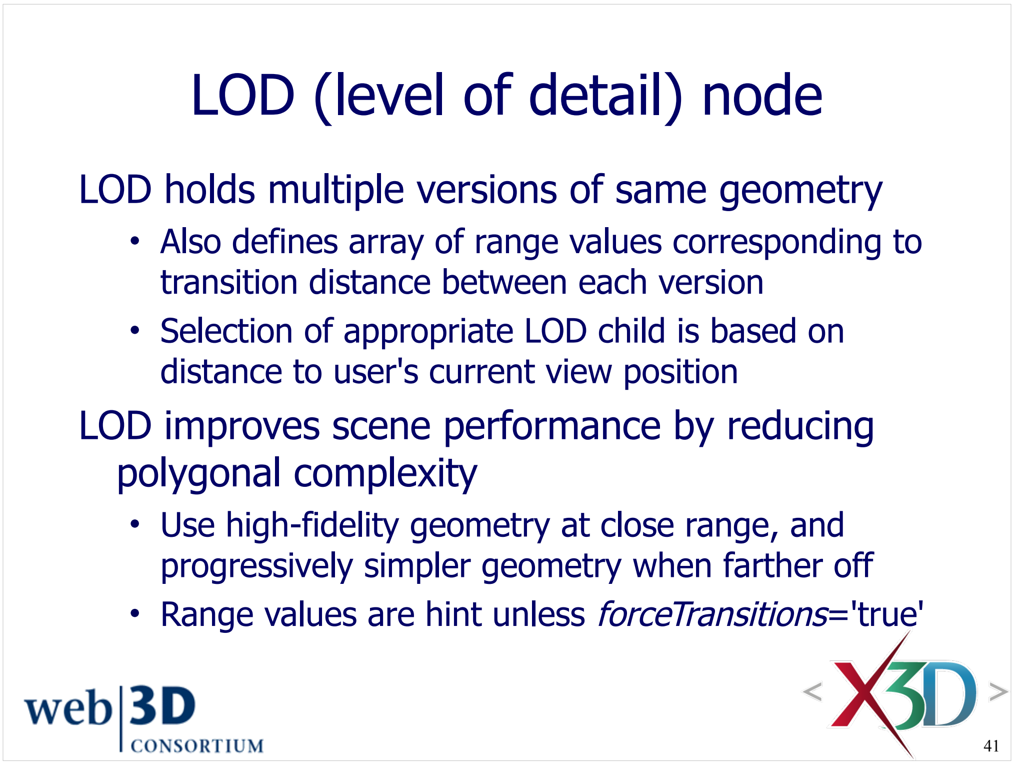
bboxCenter	0	0	0
bboxSize	-1	-1	-1

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Inline.x3d



Inline	**Inline is a Grouping node that can load nodes from another X3D scene via url.** **Hint: you cannot ROUTE values into an Inline scene, use IMPORT/EXPORT (or ExternProtoDeclare and ProtoInstance) instead.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. **Hint:** descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. **Hint:** USEing other geometry (instead of duplicating nodes) can improve performance. **Warning:** do NOT include DEF (or any other attribute values) when using a USE attribute!
load	**[load: accessType inputOutput, type SFBool (true|false) "true"]** load=true means load immediately, load=false means defer loading or unload contained scene. **Hint:** use LoadSensor to detect when loading is complete.
url	**[url: accessType inputOutput, type MFString CDATA #IMPLIED]** . **Hint:** Strings can have multiple values, so separate each string by quote marks ["http://www.url1.org" "http://www.url2.org" "etc."]. **Hint:** XML encoding for " is &quot; (a character entity). **Warning:** strictly match directory and filename capitalization for http links! **Hint:** can replace embedded blank(s) in url queries with %20 for each blank character.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

http://www.web3d.org/x3d/content/X3dTooltips.html#Inline



LOD (level of detail) node

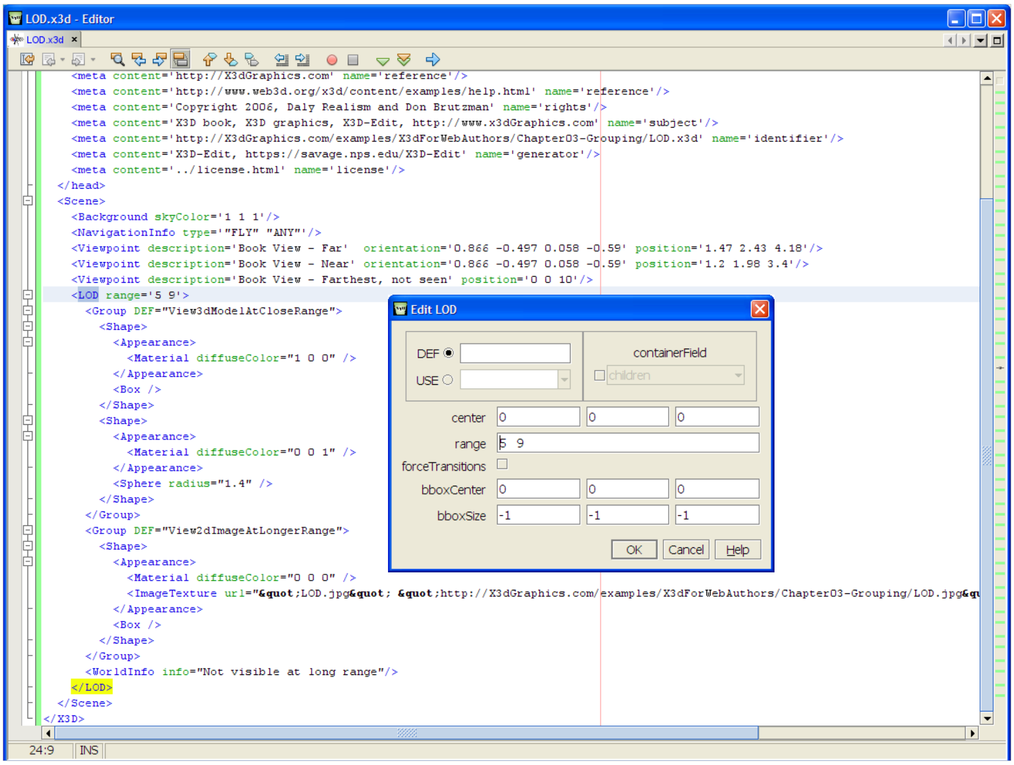
LOD holds multiple versions of same geometry

- Also defines array of range values corresponding to transition distance between each version
- Selection of appropriate LOD child is based on distance to user's current view position

LOD improves scene performance by reducing polygonal complexity

- Use high-fidelity geometry at close range, and progressively simpler geometry when farther off
- Range values are hint unless *forceTransitions*='true'

Typical order for children nodes: high fidelity, medium fidelity, low fidelity, no fidelity.



LOD.x3d - Editor

```
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright 2006, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo type='"FLY" "ANY"'/>
    <Viewpoint description='Book View - Far'  orientation='0.866 -0.497 0.058 -0.59' position='1.47 2.43 4.18'/>
    <Viewpoint description='Book View - Near' orientation='0.866 -0.497 0.058 -0.59' position='1.2 1.98 3.4'/>
    <Viewpoint description='Book View - Farthest, not seen' position='0 0 10'/>
    <LOD range='5 9'>
      <Group DEF="View3dModelAtCloseRange">
        <Shape>
          <Appearance>
            <Material diffuseColor="1 0 0" />
          </Appearance>
          <Box />
        </Shape>
        <Shape>
          <Appearance>
            <Material diffuseColor="0 0 1" />
          </Appearance>
          <Sphere radius="1.4" />
        </Shape>
      </Group>
      <Group DEF="View2dImageAtLongerRange">
        <Shape>
          <Appearance>
            <Material diffuseColor="0 0 0" />
            <ImageTexture url="&quot;LOD.jpg&quot; &quot;http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.jpg&qu
          </Appearance>
          <Box />
        </Shape>
      </Group>
      <WorldInfo info="Not visible at long range"/>
    </LOD>
  </Scene>
</X3D>
```

Edit LOD

DEF ●

USE ○

containerField

□ children

center: 0 | 0 | 0

range: 5 9

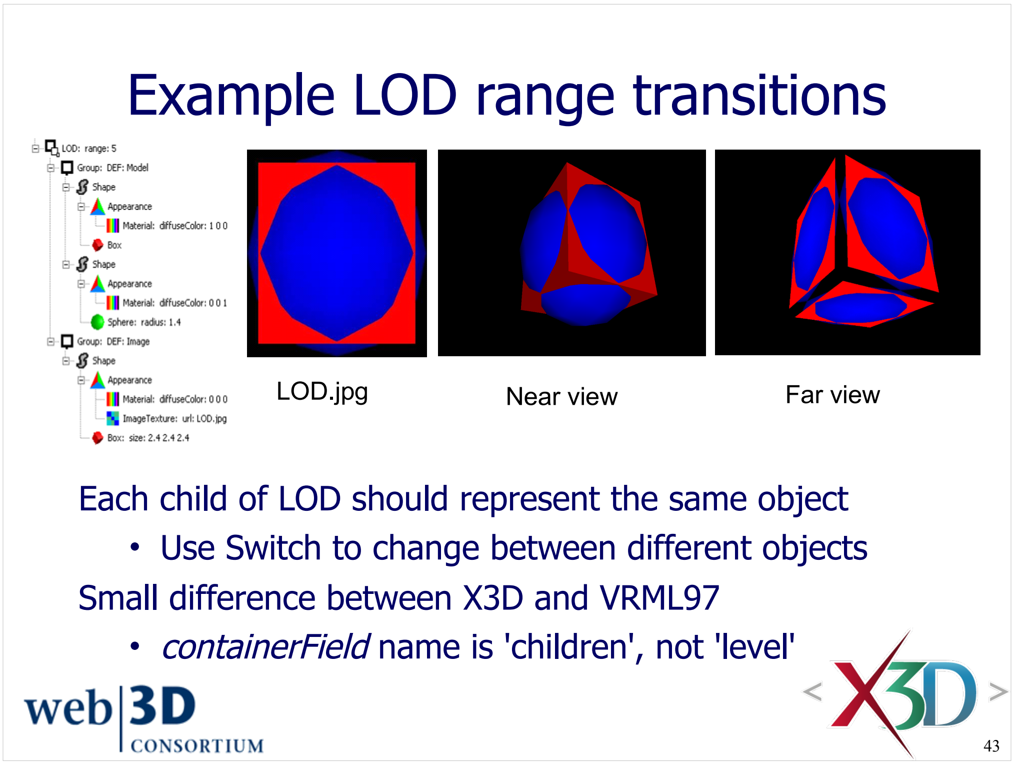
forceTransitions □

bboxCenter: 0 | 0 | 0

bboxSize: -1 | -1 | -1

Animate the scene by zooming your view in and out to watch transitions.
The next slide shows the two different versions of (sphere + box) rendering quality.

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.x3d



Example LOD range transitions

```
LOD: range: 5
  Group: DEF: Model
    Shape
      Appearance
        Material: diffuseColor: 1 0 0
      Box
    Shape
      Appearance
        Material: diffuseColor: 0 0 1
      Sphere: radius: 1.4
  Group: DEF: Image
    Shape
      Appearance
        Material: diffuseColor: 0 0 0
        ImageTexture: url: LOD.jpg
      Box: size: 2.4 2.4 2.4
```

LOD.jpg

Near view

Far view

Each child of LOD should represent the same object

- Use Switch to change between different objects

Small difference between X3D and VRML97

- *containerField* name is 'children', not 'level'

Animate the scene by zooming your view both in and out to watch transitions.

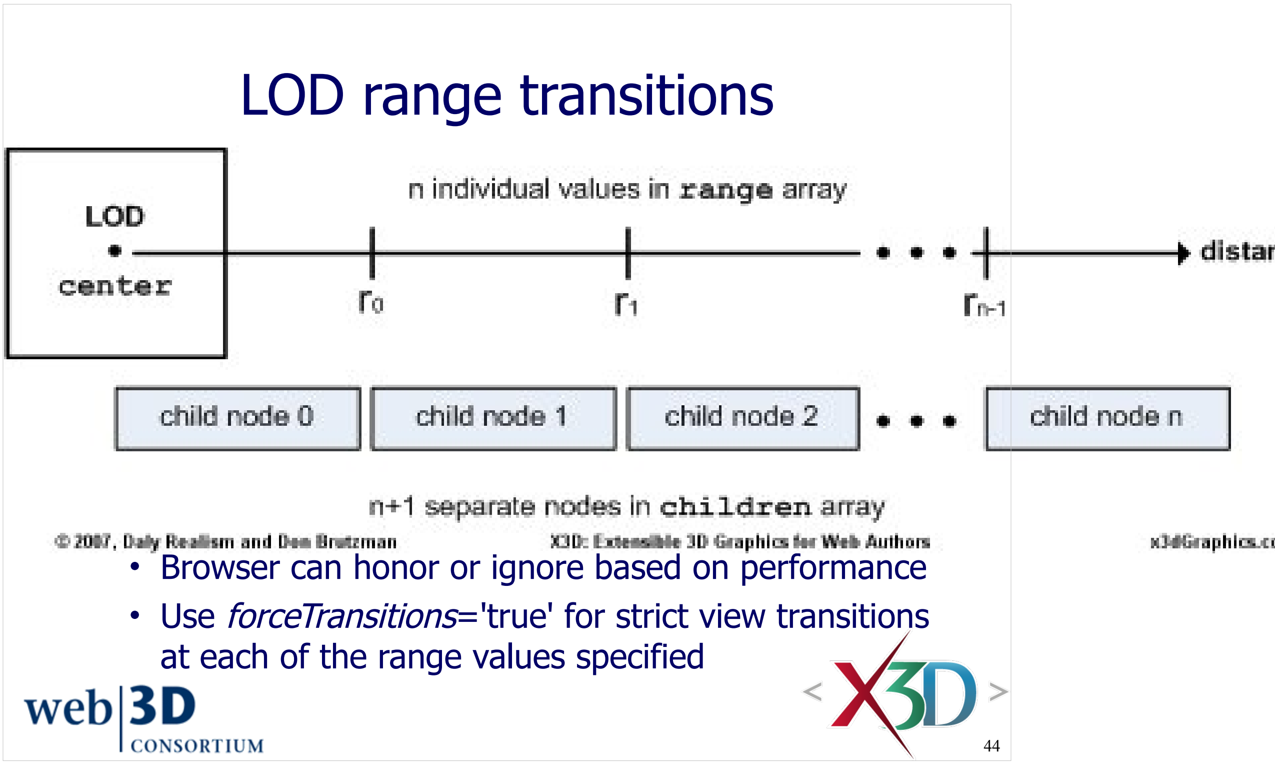
Scene:

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.x3d

ImageTexture:
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LOD.jpg

containerField is the field-name label given to child nodes.

This value was changed in X3D (*containerField*='children') from the original value in VRML97 (*containerField*='level'*)* in order to make the LOD node consistent with other X3DGroupingNode types.



LOD range transitions

LOD
center
n individual values in range array
r_0
r_1
r_{n-1}
distar
child node 0
child node 1
child node 2
child node n
n+1 separate nodes in children array
© 2007, Daly Realism and Don Brutzman
X3D: Extensible 3D Graphics for Web Authors
x3dGraphics.c

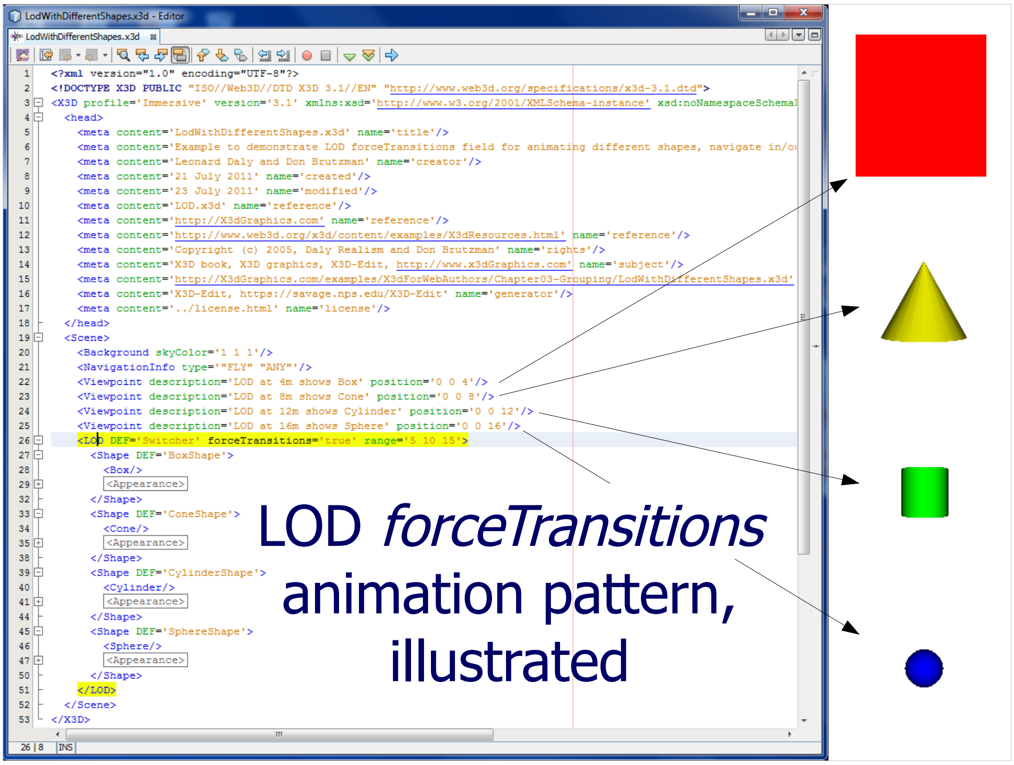
- Browser can honor or ignore based on performance
- Use *forceTransitions*='true' for strict view transitions at each of the range values specified

web 3D CONSORTIUM
<X3D>
44

The *forceTransitions* field is a new feature in X3D version 3.2.

Here is an example of the LOD design pattern:

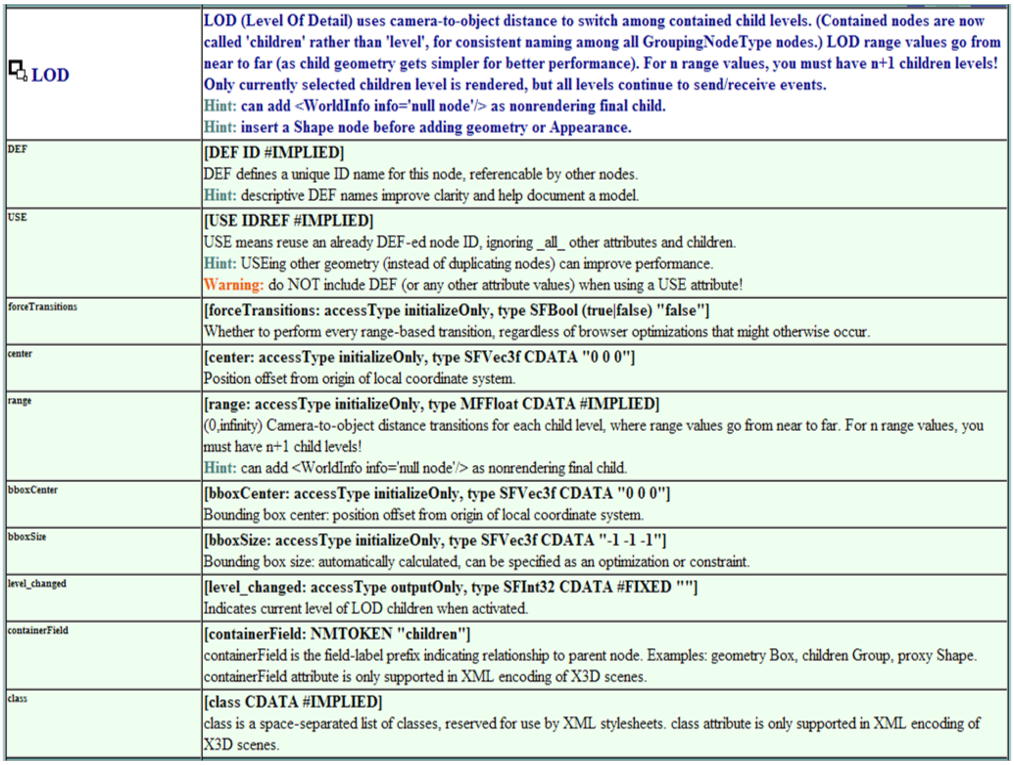
```
<LOD range='100 10000'>
   <!-- close up, less than 100m between camera and object center -->
   <Inline url="MyJeep.x3d"/>
   <!-- medium range, 100-10000m between camera and object center  -->
   <Shape>
      <Box/>
   </Shape>
   <!-- long range: draw nothing when more than 10000m between camera and object center -->
   <Group DEF="Null-DrawNothing"/>
</LOD>
```


LodWithDifferentShapes.x3d - Editor
LodWithDifferentShapes.x3d
26 | 8
INS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaL
  <head>
    <meta content='LodWithDifferentShapes.x3d' name='title'/>
    <meta content='Example to demonstrate LOD forceTransitions field for animating different shapes, navigate in/ou
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='21 July 2011' name='created'/>
    <meta content='23 July 2011' name='modified'/>
    <meta content='LOD.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/X3dResources.html' name='reference'/>
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LodWithDifferentShapes.x3d'
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo type='"FLY" "ANY"'/>
    <Viewpoint description='LOD at 4m shows Box' position='0 0 4'/>
    <Viewpoint description='LOD at 8m shows Cone' position='0 0 8'/>
    <Viewpoint description='LOD at 12m shows Cylinder' position='0 0 12'/>
    <Viewpoint description='LOD at 16m shows Sphere' position='0 0 16'/>
    <LOD DEF='Switcher' forceTransitions='true' range='5 10 15'>
      <Shape DEF='BoxShape'>
        <Box/>
        <Appearance>
      </Shape>
      <Shape DEF='ConeShape'>
        <Cone/>
        <Appearance>
      </Shape>
      <Shape DEF='CylinderShape'>
        <Cylinder/>
        <Appearance>
      </Shape>
      <Shape DEF='SphereShape'>
        <Sphere/>
        <Appearance>
      </Shape>
    </LOD>
  </Scene>
</X3D>
```

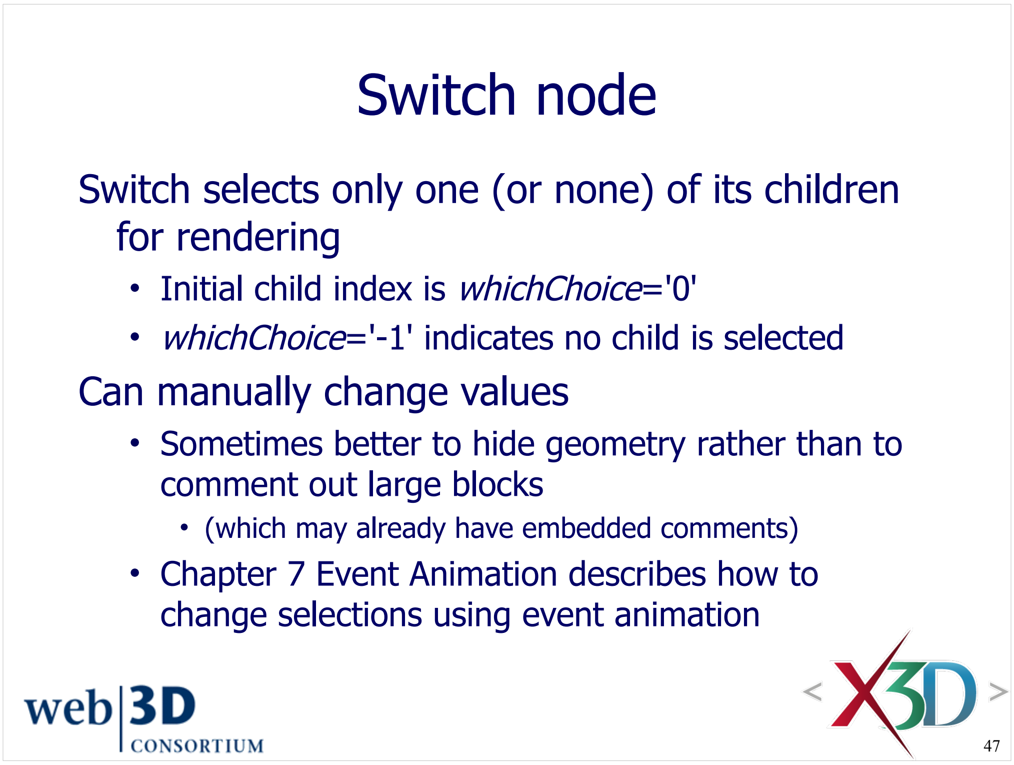
LOD *forceTransitions* animation pattern, illustrated

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/LodWithDifferentShapes.x3d



LOD	**LOD (Level Of Detail) uses camera-to-object distance to switch among contained child levels. (Contained nodes are now called 'children' rather than 'level', for consistent naming among all GroupingNodeType nodes.) LOD range values go from near to far (as child geometry gets simpler for better performance). For n range values, you must have n+1 children levels! Only currently selected children level is rendered, but all levels continue to send/receive events.** Hint: **can add <WorldInfo info='null node'/> as nonrendering final child.** Hint: **insert a Shape node before adding geometry or Appearance.**
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
forceTransitions	**[forceTransitions: accessType initializeOnly, type SFBool (true|false) "false"]** Whether to perform every range-based transition, regardless of browser optimizations that might otherwise occur.
center	**[center: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Position offset from origin of local coordinate system.
range	**[range: accessType initializeOnly, type MFFloat CDATA #IMPLIED]** (0,infinity) Camera-to-object distance transitions for each child level, where range values go from near to far. For n range values, you must have n+1 child levels! Hint: can add <WorldInfo info='null node'/> as nonrendering final child.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
level_changed	**[level_changed: accessType outputOnly, type SFInt32 CDATA #FIXED ""]** Indicates current level of LOD children when activated.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

http://www.web3d.org/x3d/content/X3dTooltips.html#LOD



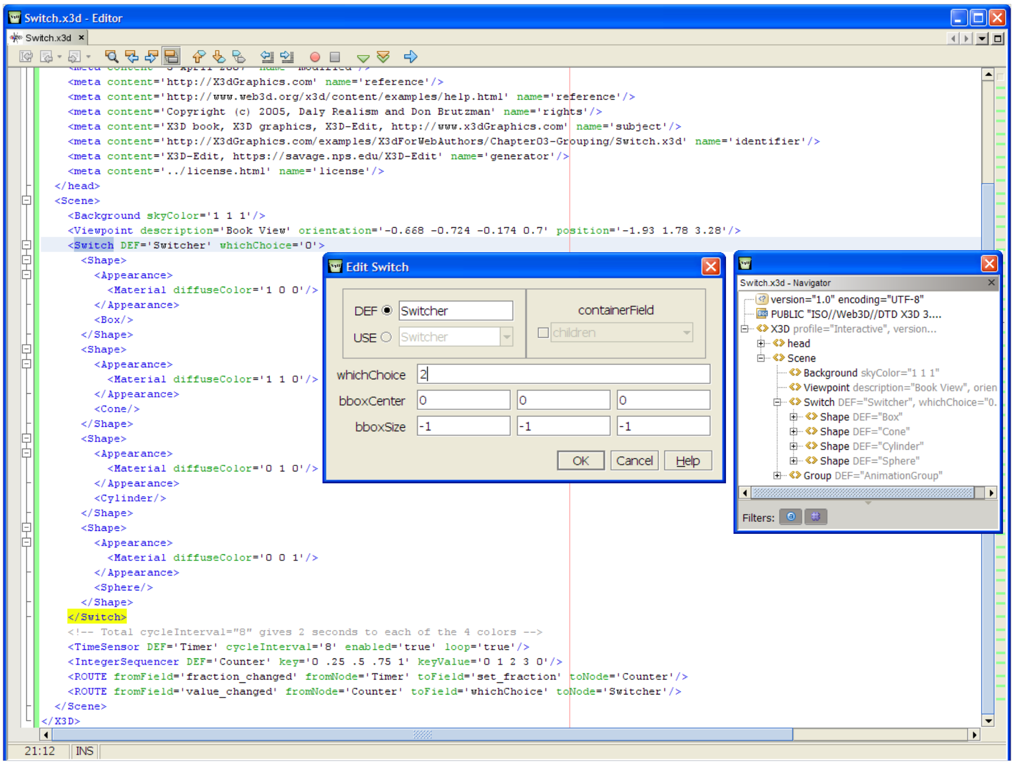
Switch node

Switch selects only one (or none) of its children for rendering

- Initial child index is *whichChoice*='0'
- *whichChoice*='-1' indicates no child is selected

Can manually change values

- Sometimes better to hide geometry rather than to comment out large blocks
  - (which may already have embedded comments)
- Chapter 7 Event Animation describes how to change selections using event animation



```
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright (c) 2005, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Switch.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='-0.668 -0.724 -0.174 0.7' position='-1.93 1.78 3.28'/>
    <Switch DEF='Switcher' whichChoice='0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 1 0'/>
        </Appearance>
        <Cone/>
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 1 0'/>
        </Appearance>
        <Cylinder/>
      </Shape>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 0 1'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Switch>
    <!-- Total cycleInterval="8" gives 2 seconds to each of the 4 colors -->
    <TimeSensor DEF='Timer' cycleInterval='8' enabled='true' loop='true'/>
    <IntegerSequencer DEF='Counter' key='0 .25 .5 .75 1' keyValue='0 1 2 3 0'/>
    <ROUTE fromField='fraction_changed' fromNode='Timer' toField='set_fraction' toNode='Counter'/>
    <ROUTE fromField='value_changed' fromNode='Counter' toField='whichChoice' toNode='Switcher'/>
  </Scene>
</X3D>
```

Edit Switch

DEF ● Switcher
USE ○ Switcher
containerField
□ children
whichChoice: 2
bboxCenter: 0 0 0
bboxSize: -1 -1 -1

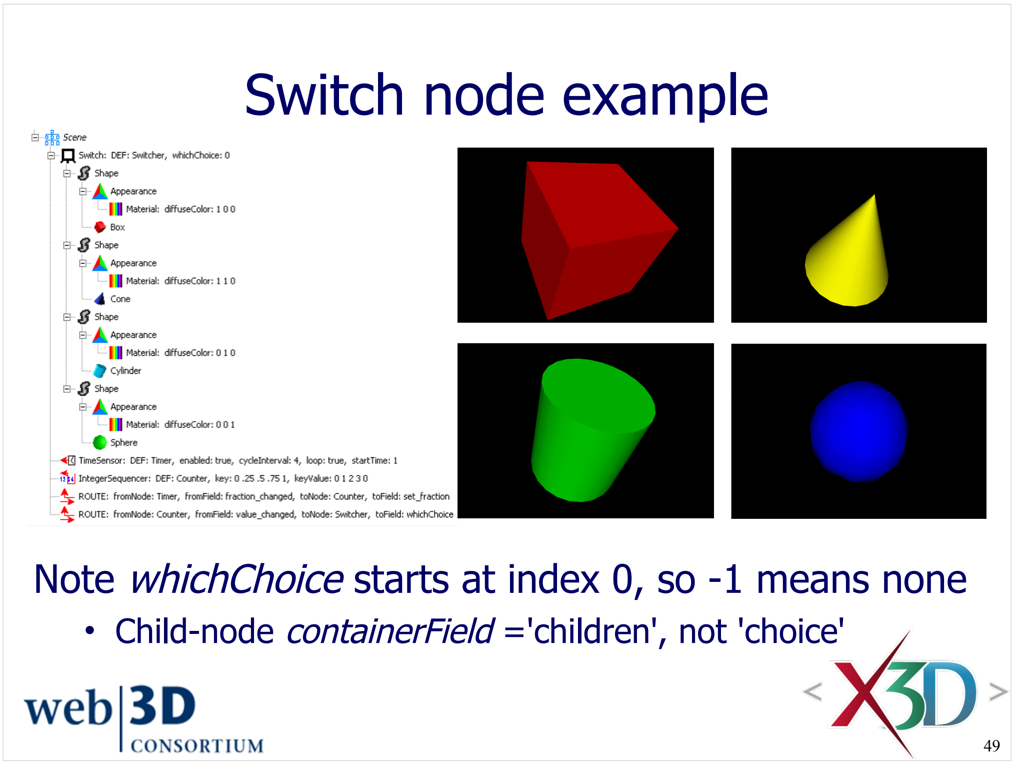
Switch.x3d - Navigator

```
version="1.0" encoding="UTF-8"
PUBLIC "ISO//Web3D//DTD X3D 3....
X3D profile="Interactive", version...
  head
  Scene
    Background skyColor="1 1 1"
    Viewpoint description="Book View", orien
    Switch DEF="Switcher", whichChoice="0.
      Shape DEF="Box"
      Shape DEF="Cone"
      Shape DEF="Cylinder"
      Shape DEF="Sphere"
    Group DEF="AnimationGroup"
```

These scene has an added built-in animation to show the Switch in operation. This allows the scene to automatically illustrate the effects of changing the *whichChoice* field.

Animation is explained in Chapter07 (Event Animation and Interpolation) so don't worry about the nodes that haven't been covered yet. It is a good “look ahead” at some of the capabilities we will be learning.

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Switch.x3d



Switch node example

```
Scene
  Switch:  DEF: Switcher,  whichChoice: 0
    Shape
      Appearance
        Material:  diffuseColor: 1 0 0
      Box
    Shape
      Appearance
        Material:  diffuseColor: 1 1 0
      Cone
    Shape
      Appearance
        Material:  diffuseColor: 0 1 0
      Cylinder
    Shape
      Appearance
        Material:  diffuseColor: 0 0 1
      Sphere
  TimeSensor:  DEF: Timer,  enabled: true,  cycleInterval: 4,  loop: true,  startTime: 1
  IntegerSequencer:  DEF: Counter,  key: 0 .25 .5 .75 1,  keyValue: 0 1 2 3 0
  ROUTE:  fromNode: Timer,  fromField: fraction_changed,  toNode: Counter,  toField: set_fraction
  ROUTE:  fromNode: Counter,  fromField: value_changed,  toNode: Switcher,  toField: whichChoice
```

Note *whichChoice* starts at index 0, so -1 means none

- Child-node *containerField* ='children', not 'choice'

49

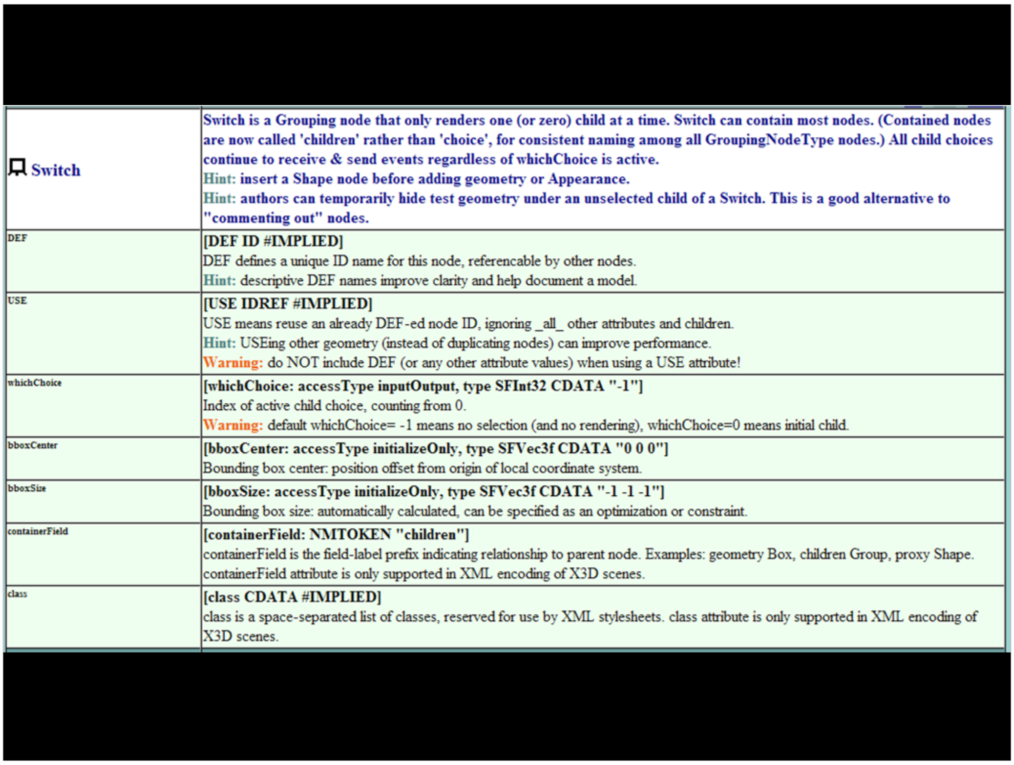
Figure 3.12, page 91, *X3D for Web Authors*

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Switch.x3d

Each of the black-background objects shows the different views that occur when the value of the Switch node's *whichChoice* field is changed.

containerField is the field-name label given to child nodes.

The default *containerField* value for Switch was changed to *containerField*='children' in X3D (from *containerField*='choice' in VRML97) in order to make the Switch node consistent with other X3DGroupingNode types.

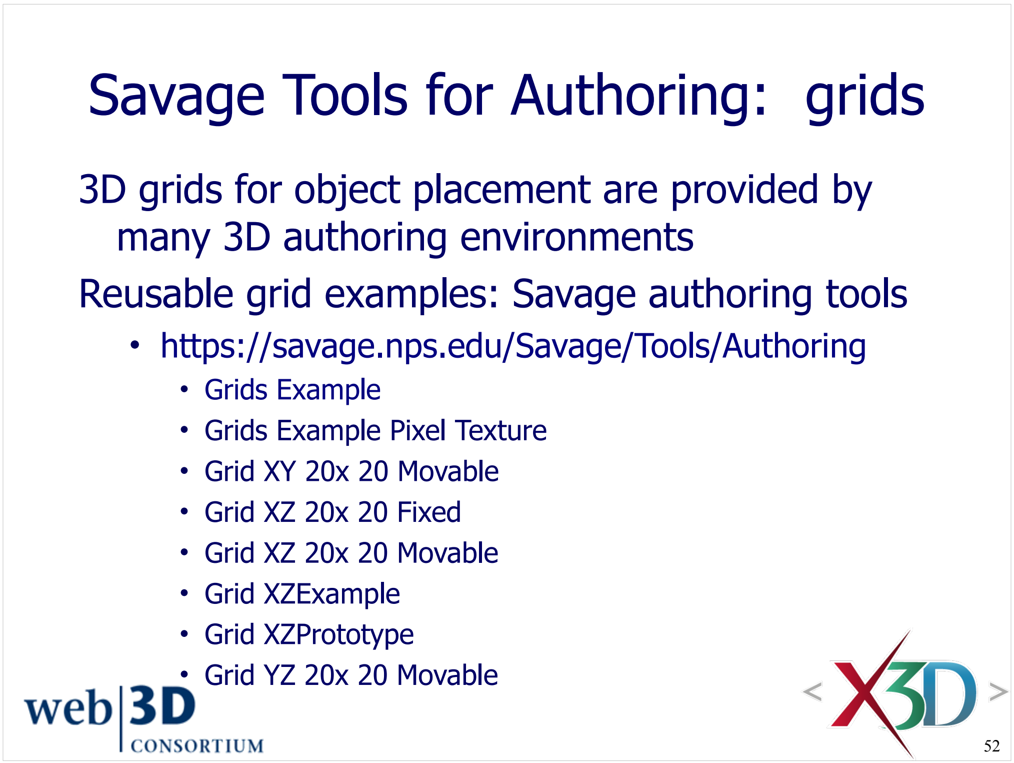


Switch	Switch is a Grouping node that only renders one (or zero) child at a time. Switch can contain most nodes. (Contained nodes are now called 'children' rather than 'choice', for consistent naming among all GroupingNodeType nodes.) All child choices continue to receive & send events regardless of whichChoice is active. Hint: insert a Shape node before adding geometry or Appearance. Hint: authors can temporarily hide test geometry under an unselected child of a Switch. This is a good alternative to "commenting out" nodes.
DEF	**[DEF ID #IMPLIED]** DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	**[USE IDREF #IMPLIED]** USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
whichChoice	**[whichChoice: accessType inputOutput, type SFInt32 CDATA "-1"]** Index of active child choice, counting from 0. Warning: default whichChoice= -1 means no selection (and no rendering), whichChoice=0 means initial child.
bboxCenter	**[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]** Bounding box center: position offset from origin of local coordinate system.
bboxSize	**[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]** Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	**[containerField: NMTOKEN "children"]** containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	**[class CDATA #IMPLIED]** class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

http://www.web3d.org/x3d/content/X3dTooltips.html#Switch

back to Table of Contents

Additional Resources

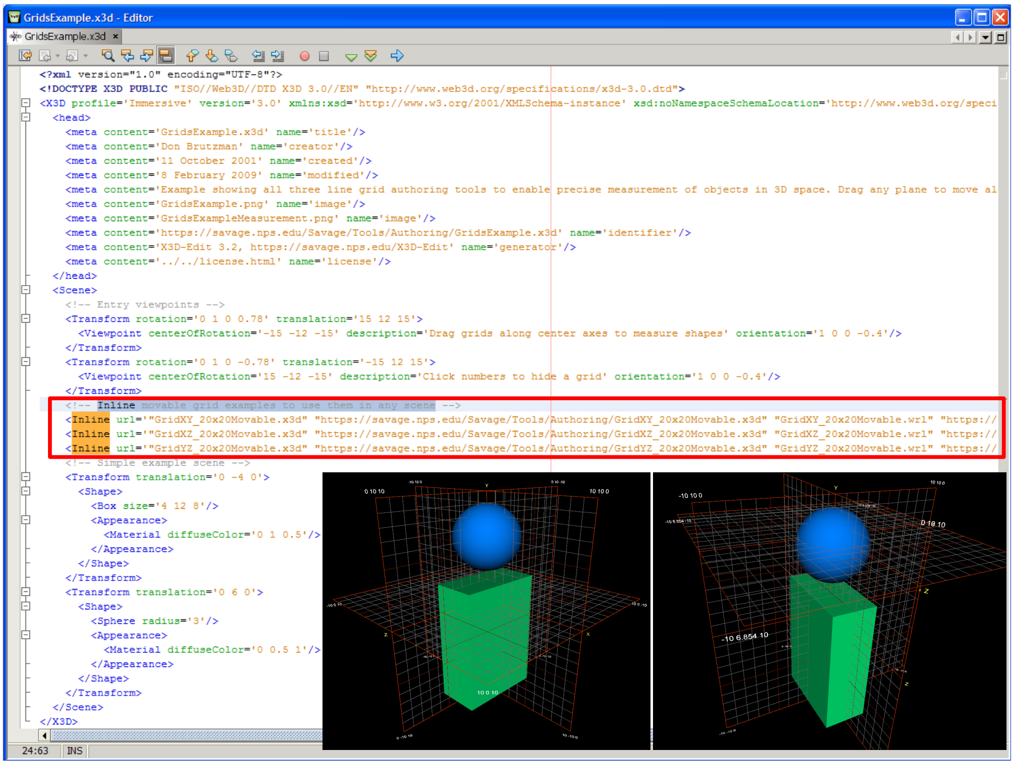


Savage Tools for Authoring: grids

3D grids for object placement are provided by many 3D authoring environments

Reusable grid examples: Savage authoring tools

- https://savage.nps.edu/Savage/Tools/Authoring
  - Grids Example
  - Grids Example Pixel Texture
  - Grid XY 20x 20 Movable
  - Grid XZ 20x 20 Fixed
  - Grid XZ 20x 20 Movable
  - Grid XZExample
  - Grid XZPrototype
  - Grid YZ 20x 20 Movable



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile='Immersive' version='3.0' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/speci
  <head>
    <meta content='GridsExample.x3d' name='title'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='11 October 2001' name='created'/>
    <meta content='8 February 2009' name='modified'/>
    <meta content='Example showing all three line grid authoring tools to enable precise measurement of objects in 3D space. Drag any plane to move al
    <meta content='GridsExample.png' name='image'/>
    <meta content='GridsExampleMeasurement.png' name='image'/>
    <meta content='https://savage.nps.edu/Savage/Tools/Authoring/GridsExample.x3d' name='identifier'/>
    <meta content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../../license.html' name='license'/>
  </head>
  <Scene>
    <!-- Entry viewpoints -->
    <Transform rotation='0 1 0 0.78' translation='15 12 15'>
      <Viewpoint centerOfRotation='-15 -12 -15' description='Drag grids along center axes to measure shapes' orientation='1 0 0 -0.4'/>
    </Transform>
    <Transform rotation='0 1 0 -0.78' translation='-15 12 15'>
      <Viewpoint centerOfRotation='15 -12 -15' description='Click numbers to hide a grid' orientation='1 0 0 -0.4'/>
    </Transform>
    <!-- Inline movable grid examples to use them in any scene -->
    <Inline url='"GridXY_20x20Movable.x3d" "https://savage.nps.edu/Savage/Tools/Authoring/GridXY_20x20Movable.x3d" "GridXY_20x20Movable.wrl" "https://
    <Inline url='"GridXZ_20x20Movable.x3d" "https://savage.nps.edu/Savage/Tools/Authoring/GridXZ_20x20Movable.x3d" "GridXZ_20x20Movable.wrl" "https://
    <Inline url='"GridYZ_20x20Movable.x3d" "https://savage.nps.edu/Savage/Tools/Authoring/GridYZ_20x20Movable.x3d" "GridYZ_20x20Movable.wrl" "https://
    <!-- Simple example scene -->
    <Transform translation='0 -4 0'>
      <Shape>
        <Box size='4 12 8'/>
        <Appearance>
          <Material diffuseColor='0 1 0.5'/>
        </Appearance>
      </Shape>
    </Transform>
    <Transform translation='0 6 0'>
      <Shape>
        <Sphere radius='3'/>
        <Appearance>
          <Material diffuseColor='0 0.5 1'/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

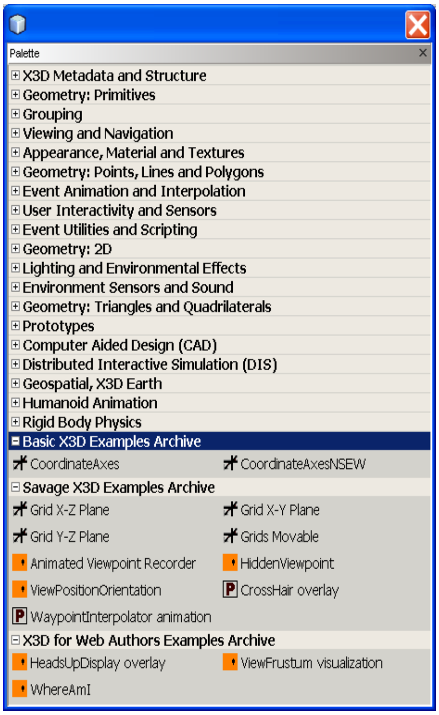
https://savage.nps.edu/Savage/Tools/Authoring/GridsExample.x3d

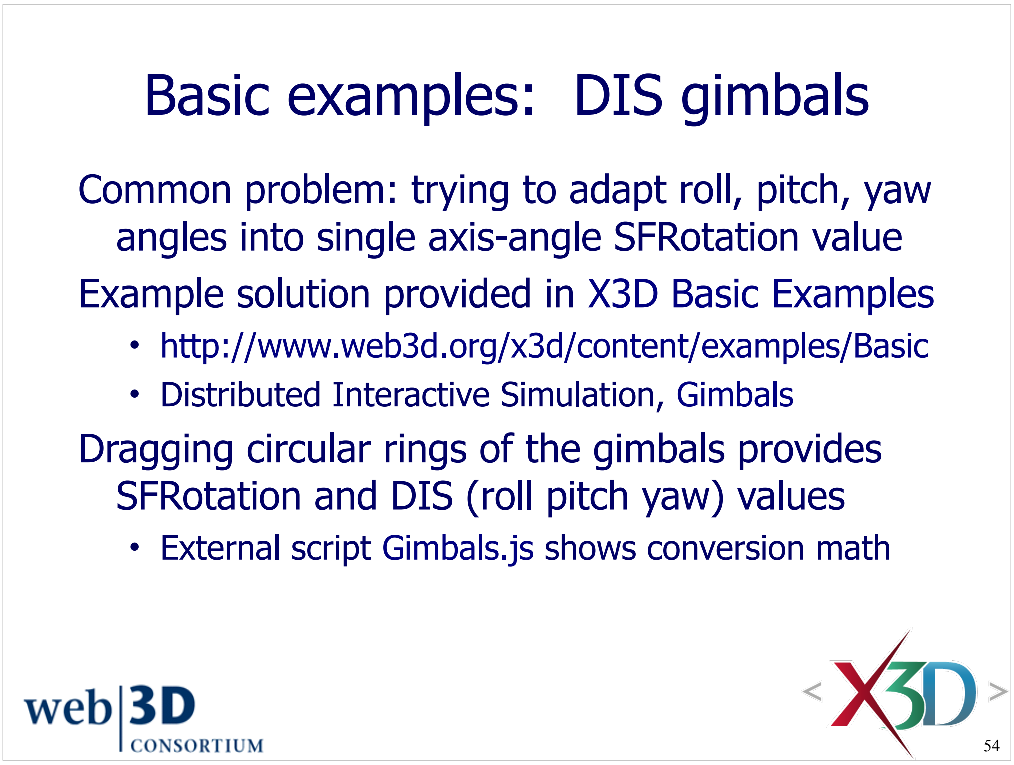
Grids Example shows three grids aligned to XY, XZ, YZ planes respectively.

- The white coordinate labels can be selected to temporarily hide a grid
- The yellow axis labels can be dragged along their perpendicular axis to position each grid along the desired level
- Moving a grid plane updates the white coordinate labels to the new location

This approach lets authors then visually inspect where grid lines intersect geometry in order to get a good measurement of various shape dimensions in any parent scene.

Note that these grids are available in X3D-Edit as built-in authoring tools under the Savage X3D Authoring Tools palette. Simply drag/drop the desired tool icon into a scene for further testing of your content.


Palette
⊞ X3D Metadata and Structure
⊞ Geometry: Primitives
⊞ Grouping
⊞ Viewing and Navigation
⊞ Appearance, Material and Textures
⊞ Geometry: Points, Lines and Polygons
⊞ Event Animation and Interpolation
⊞ User Interactivity and Sensors
⊞ Event Utilities and Scripting
⊞ Geometry: 2D
⊞ Lighting and Environmental Effects
⊞ Environment Sensors and Sound
⊞ Geometry: Triangles and Quadrilaterals
⊞ Prototypes
⊞ Computer Aided Design (CAD)
⊞ Distributed Interactive Simulation (DIS)
⊞ Geospatial, X3D Earth
⊞ Humanoid Animation
⊞ Rigid Body Physics
⊟ Basic X3D Examples Archive
CoordinateAxes
CoordinateAxesNSEW
⊟ Savage X3D Examples Archive
Grid X-Z Plane
Grid X-Y Plane
Grid Y-Z Plane
Grids Movable
Animated Viewpoint Recorder
HiddenViewpoint
ViewPositionOrientation
CrossHair overlay
WaypointInterpolator animation
⊟ X3D for Web Authors Examples Archive
HeadsUpDisplay overlay
ViewFrustum visualization
WhereAmI



Basic examples: DIS gimbals

Common problem: trying to adapt roll, pitch, yaw angles into single axis-angle SFRotation value

Example solution provided in X3D Basic Examples

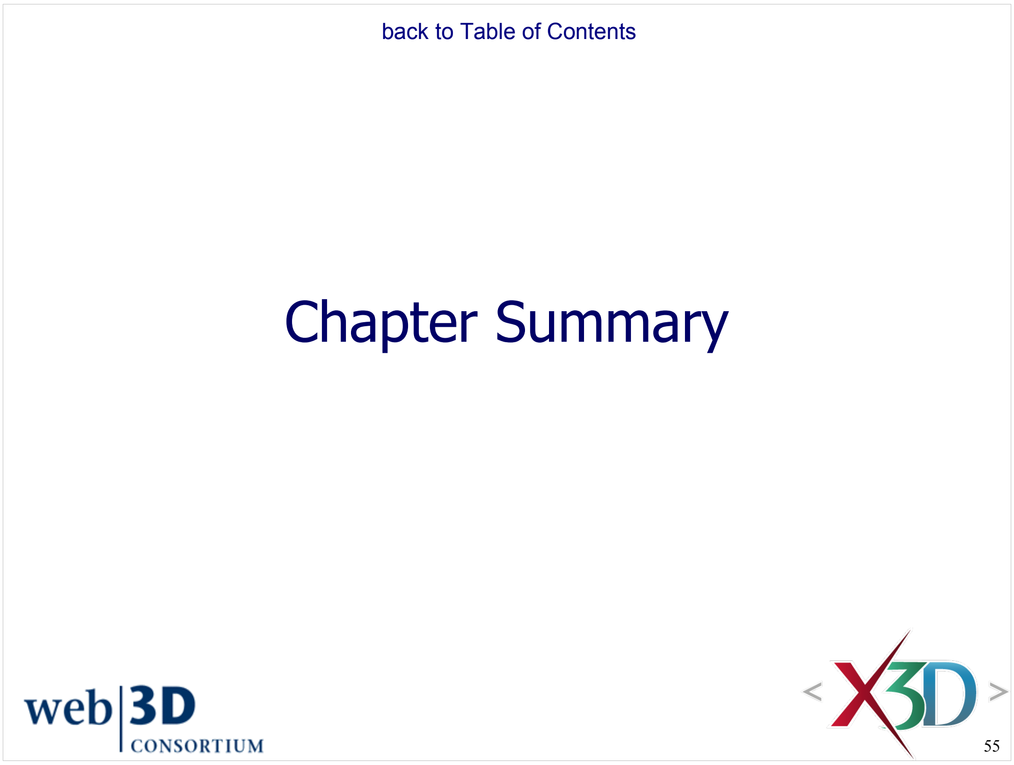
- http://www.web3d.org/x3d/content/examples/Basic
- Distributed Interactive Simulation, Gimbals

Dragging circular rings of the gimbals provides SFRotation and DIS (roll pitch yaw) values

- External script Gimbals.js shows conversion math

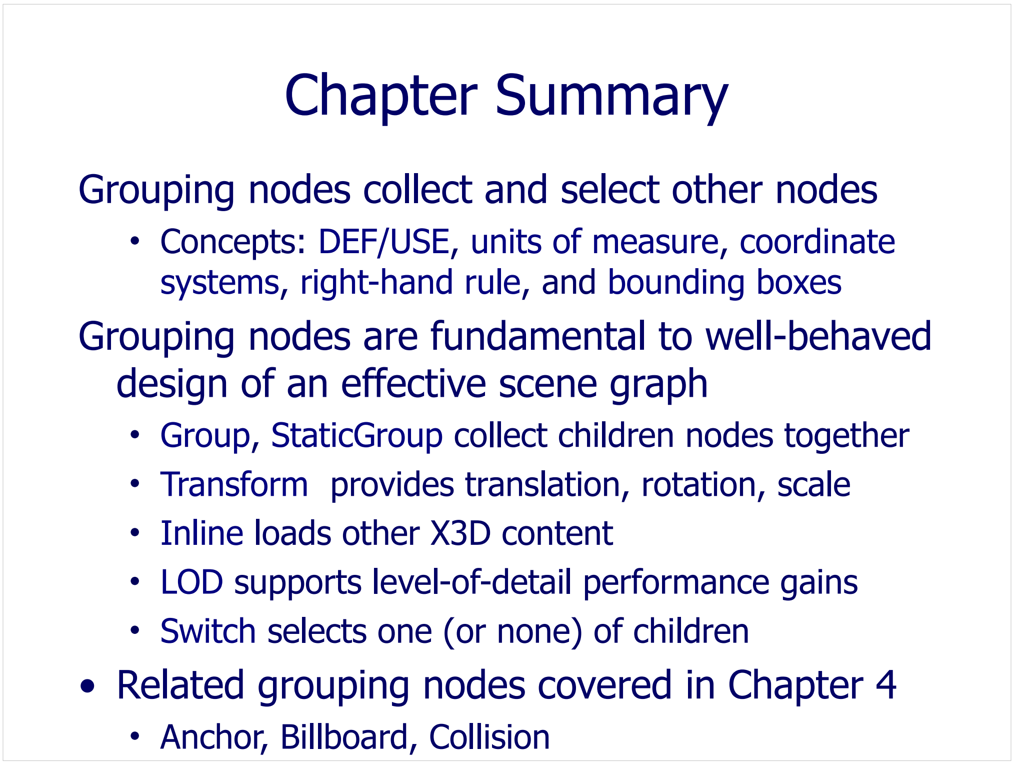
This is a fairly sophisticated script and an advanced technique. Script nodes are covered in Chapter 9, Event Utilities and Scripting.

http://www.web3d.org/x3d/content/examples/Basic/DistributedInteractiveSimulation/Gimbals.x3d



back to Table of Contents

Chapter Summary



Chapter Summary

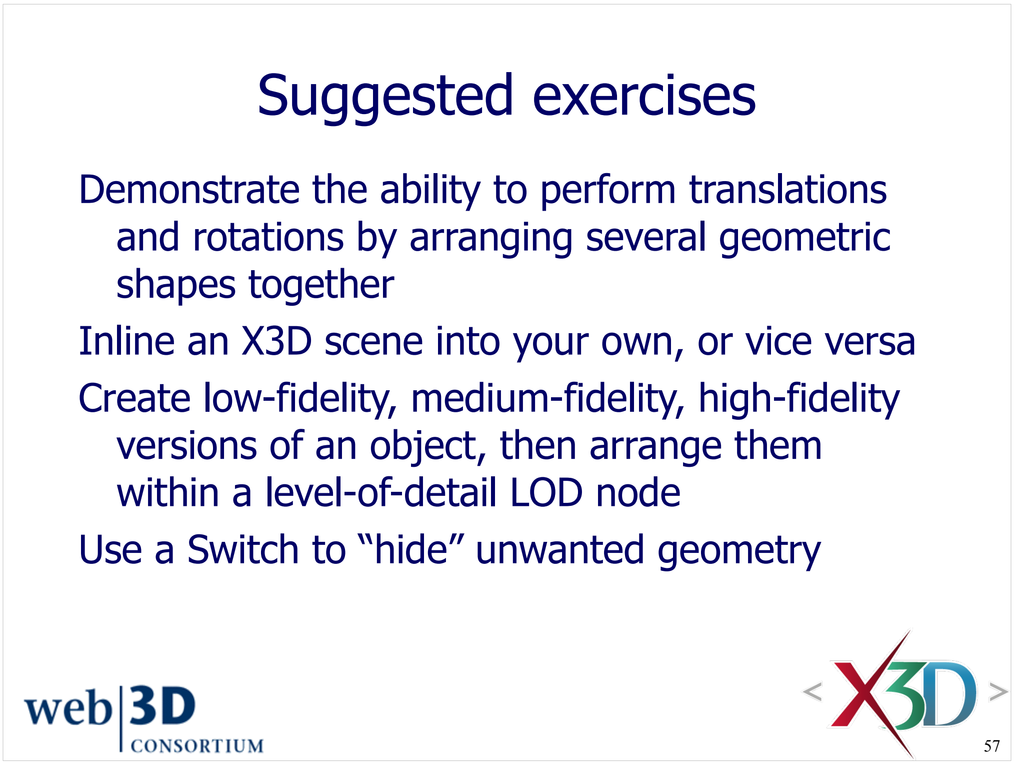
Grouping nodes collect and select other nodes

- Concepts: DEF/USE, units of measure, coordinate systems, right-hand rule, and bounding boxes

Grouping nodes are fundamental to well-behaved design of an effective scene graph

- Group, StaticGroup collect children nodes together
- Transform provides translation, rotation, scale
- Inline loads other X3D content
- LOD supports level-of-detail performance gains
- Switch selects one (or none) of children

- Related grouping nodes covered in Chapter 4
  - Anchor, Billboard, Collision



Suggested exercises

Demonstrate the ability to perform translations and rotations by arranging several geometric shapes together

Inline an X3D scene into your own, or vice versa

Create low-fidelity, medium-fidelity, high-fidelity versions of an object, then arrange them within a level-of-detail LOD node

Use a Switch to “hide” unwanted geometry

Setting <Switch *whichChoice*='-1'> is a good way to hide content under development. This approach is often better than “commenting out” source code, since nested XML comments (i.e. a second XML comment within another XML comment) are not allowed.

References

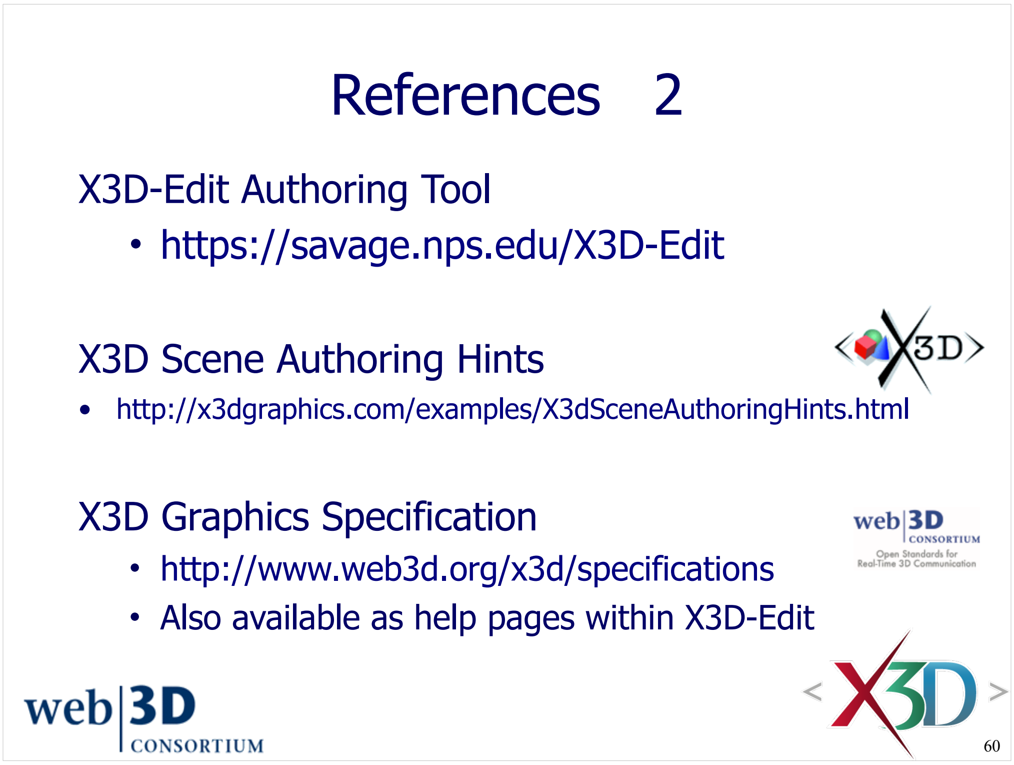
References 1

X3D: Extensible 3D Graphics for Web Authors by Don Brutzman and Leonard Daly, Morgan Kaufmann Publishers, April 2007, 468 pages.

- Chapter 3, Grouping Nodes
- http://x3dGraphics.com
- http://x3dgraphics.com/examples/X3dForWebAuthors

X3D Resources

- http://www.web3d.org/x3d/content/examples/X3dResources.html



References 2

X3D-Edit Authoring Tool

- https://savage.nps.edu/X3D-Edit

X3D Scene Authoring Hints

- http://x3dgraphics.com/examples/X3dSceneAuthoringHints.html

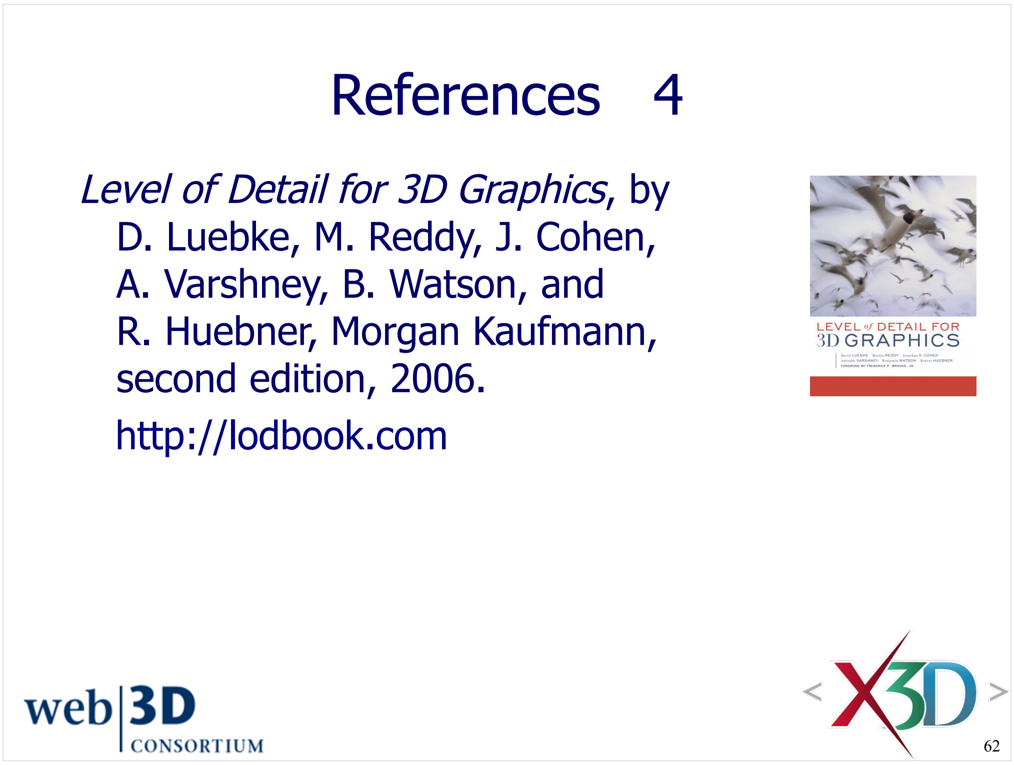
X3D Graphics Specification

- http://www.web3d.org/x3d/specifications
- Also available as help pages within X3D-Edit

References 3

VRML 2.0 Sourcebook by Andrea L. Ames, David R. Nadeau, and John L. Moreland, John Wiley & Sons, 1996.

- http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm
- http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook
- Chapter 05 - Positioning Shapes
- Chapter 06 - Rotating Shapes
- Chapter 07 - Scaling Shapes



References 4

Level of Detail for 3D Graphics, by D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner, Morgan Kaufmann, second edition, 2006.

http://lodbook.com

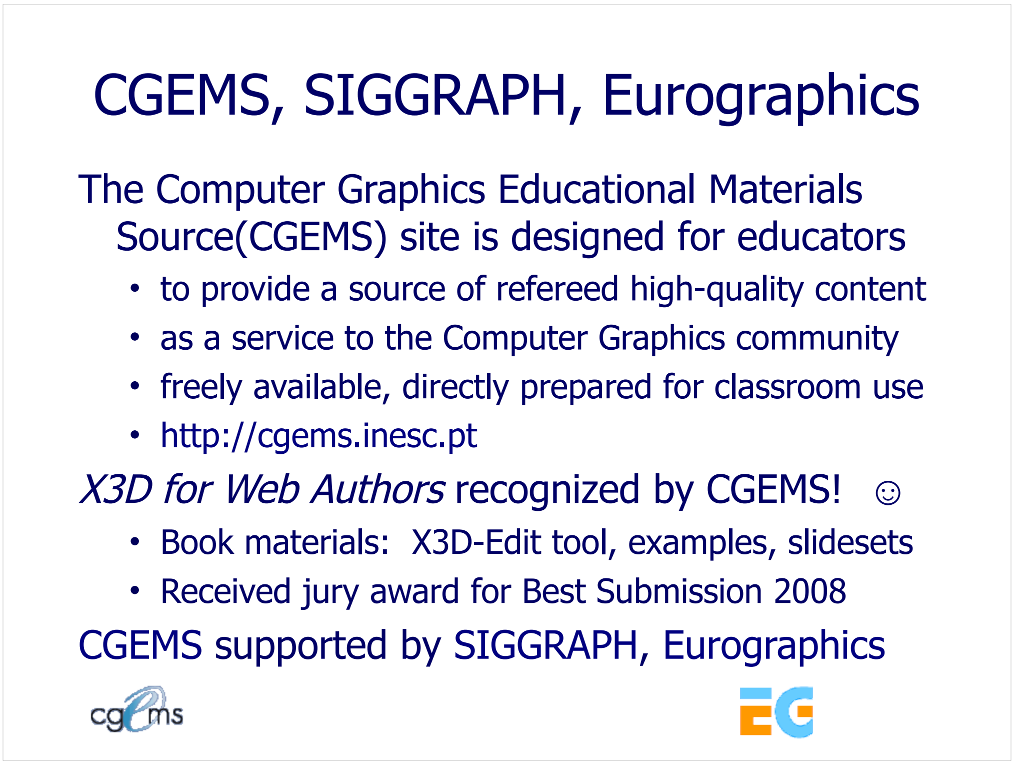
Contact

Don Brutzman

brutzman@nps.edu

http://faculty.nps.edu/brutzman

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.2149 voice



CGEMS, SIGGRAPH, Eurographics

The Computer Graphics Educational Materials Source(CGEMS) site is designed for educators

- to provide a source of refereed high-quality content
- as a service to the Computer Graphics community
- freely available, directly prepared for classroom use
- http://cgems.inesc.pt

X3D for Web Authors recognized by CGEMS! ☺

- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics

From the CGEMS home page:

- http://cgems.inesc.pt

  Welcome to CGEMS - Computer Graphics Educational Materials Source. The CGEMS site is designed for educators to provide a source of refereed high-quality content as a service to the Computer Graphics community as a whole. Materials herein are freely available and directly prepared for your classroom.

List of all published modules:

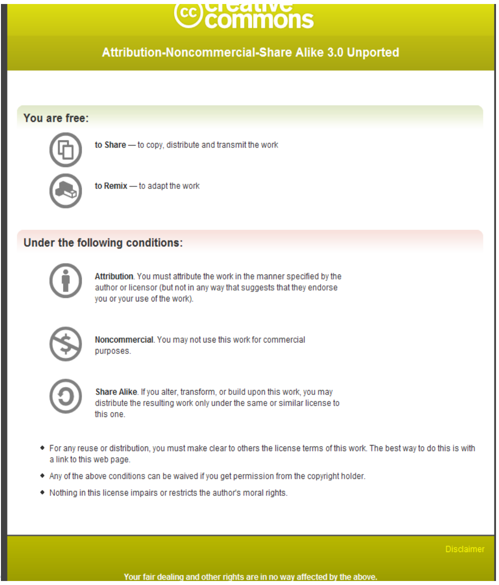
- http://cgems.inesc.pt/authors/ListModules.aspx

CGEMS Editorial Policy:

- http://cgems.inesc.pt/EditorialPolicy.htm

Creative Commons open-source license

http://creativecommons.org/licenses/by-nc-sa/3.0



Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial. You may not use this work for commercial purposes.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

Disclaimer

Your fair dealing and other rights are in no way affected by the above.

65

Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

 * to Share — to copy, distribute and transmit the work

 * to Remix — to adapt the work

Under the following conditions:

 * Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Attribute this work: What does "Attribute this work" mean?

The page you came from contained embedded licensing metadata, including how the creator wishes to be attributed for re-use. You can use the HTML here to cite the work. Doing so will also include metadata on your page so that others can find the original work as well.

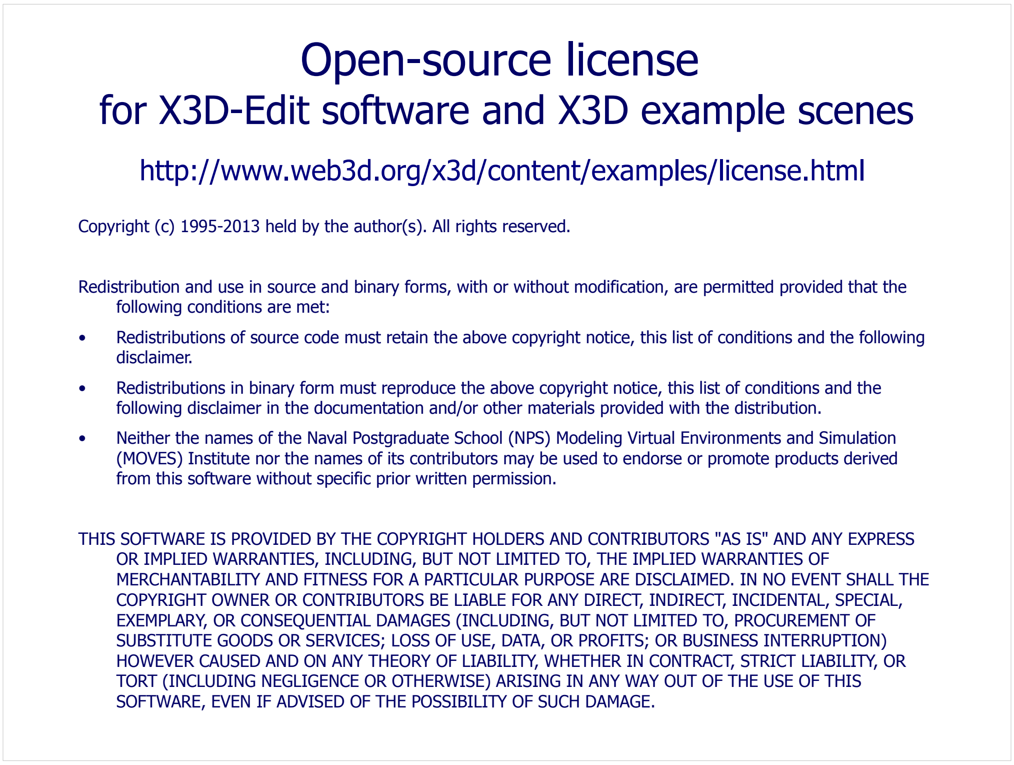
 * Noncommercial. You may not use this work for commercial purposes.

 * Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

 * For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

 * Any of the above conditions can be waived if you get permission from the copyright holder.

 * Nothing in this license impairs or restricts the author's moral rights.



Open-source license
for X3D-Edit software and X3D example scenes

http://www.web3d.org/x3d/content/examples/license.html

Copyright (c) 1995-2013 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the Naval Postgraduate School (NPS) Modeling Virtual Environments and Simulation (MOVES) Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License available at

http://www.web3d.org/x3d/content/examples/license.txt

http://www.web3d.org/x3d/content/examples/license.html

Good references on open source:

Andrew M. St. Laurent, *Understanding Open Source and Free Software Licensing*, O'Reilly Publishing, Sebastopol California, August 2004. http://oreilly.com/catalog/9780596005818/index.html

Herz, J. C., Mark Lucas, John Scott, *Open Technology Development: Roadmap Plan*, Deputy Under Secretary of Defense for Advanced Systems and Concepts, Washington DC, April 2006. http://handle.dtic.mil/100.2/ADA450769