

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

References

Chapter Overview

Overview: grouping nodes

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

Tree

Directed acyclic graph (DAG)

Node

Parent-child relationship

Subgraph

Leaf node

Intermediate or internal node

XML and X3D correspondence

Opening element

Singleton element, `attribute="value"`

Opening element

Singleton element, `attribute='value'`

Closing element

Closing element

Review

```
<Shape>
  <Sphere radius="10.0" solid="true"/>
  <Appearance>
    <ImageTexture url='earth-topo.png'/>
  </Appearance>
</Shape>
```

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
- Provides target for routing events

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!

- 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

- $1 \text{ m} = 39.3''$

Angular measurements in radians

- $2 \pi = 360 \text{ degrees}$

Time measured in seconds

- Starting 1 January 1970

Coordinate systems

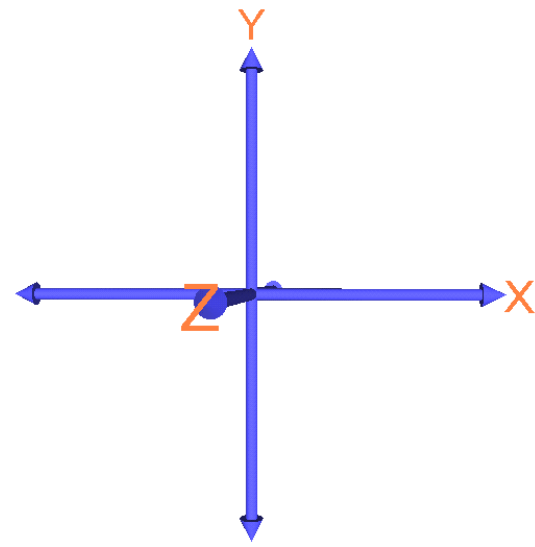
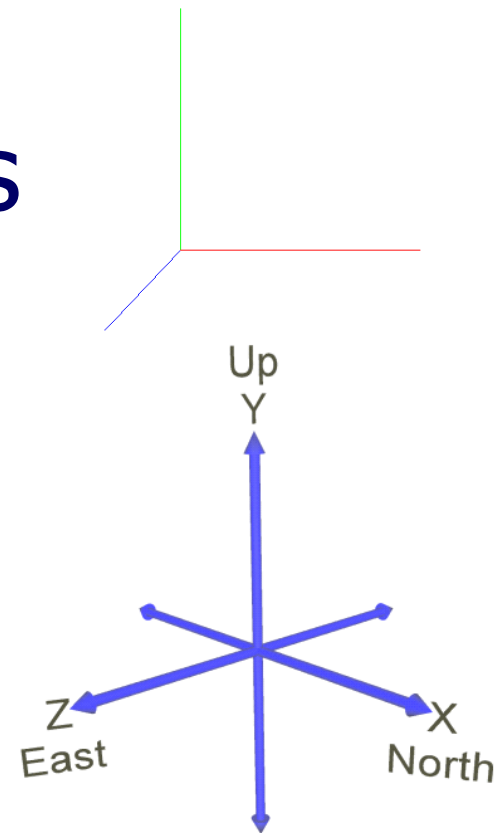
Right hand rule for X Y Z order

Y axis is up

Correspondence: North, Up, East

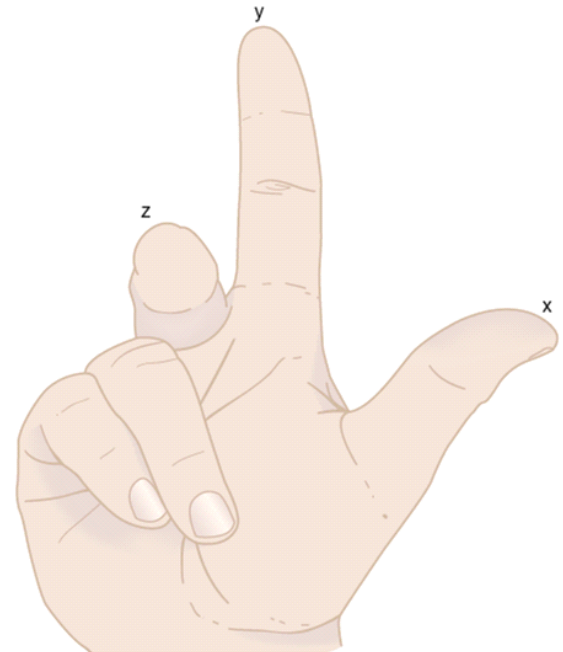
Accept no substitutes!

- or at least realign them ☺

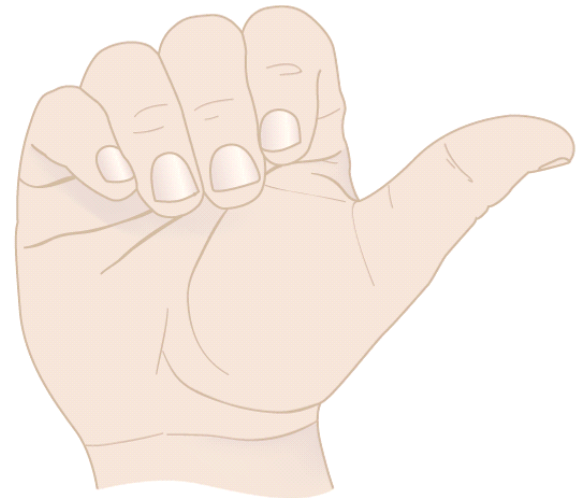


Right hand rules!

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

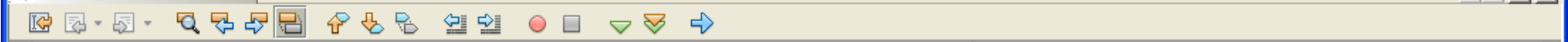
- Does not affect how an object is rendered (drawn) if it is actually larger than the bounding box
- Are never drawn themselves
- Defined by *bboxSize* and *bboxCenter*

Goal is to reduce computational complexity

- browser avoids calculating impossible collisions

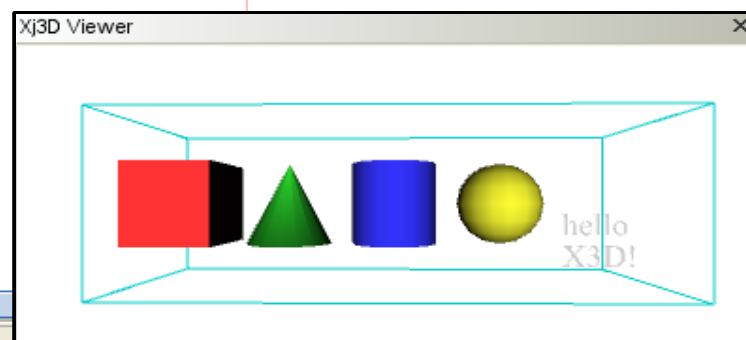
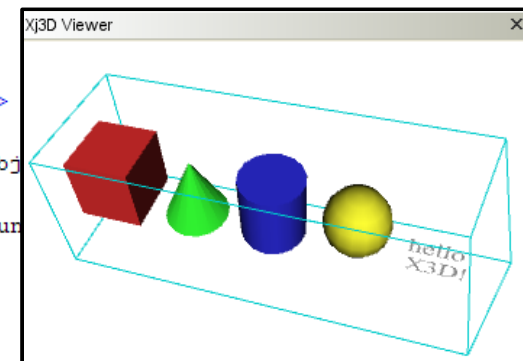
Bounding boxes can be ignored by authors

- some authoring tools can provide them if needed



```
<?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='description' />
    <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='subject' />
    <meta name='identifier'
          content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/BoundingBoxIllustration.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"/>
    <Viewpoint description="Bounding box illustration" position="0 0 15" fieldOfView="0.785"/>
    <Group bboxSize="12 4 4">
      <Inline url='../Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d'
            url='../Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.wrl'
            url='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d'
            url='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

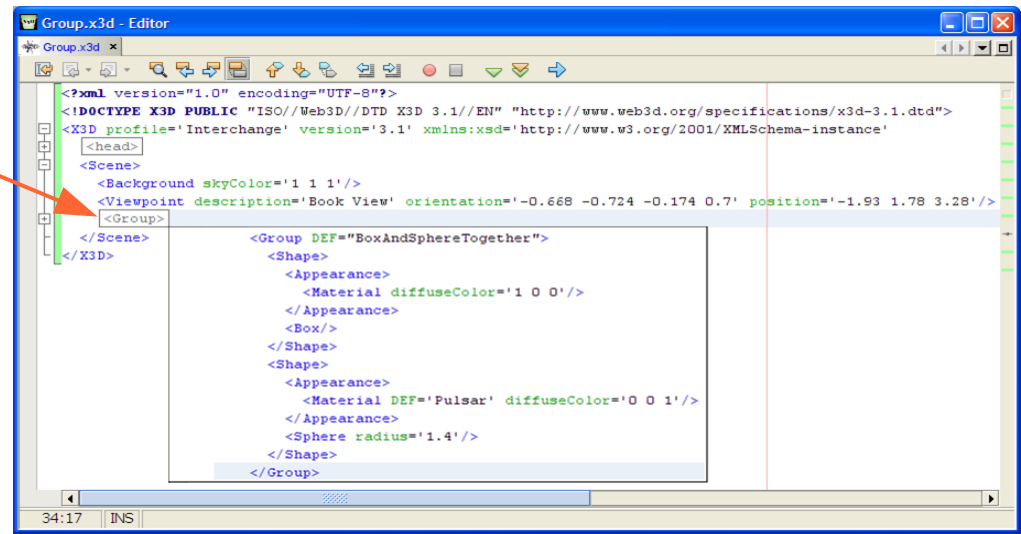
- Often close to each other spatially

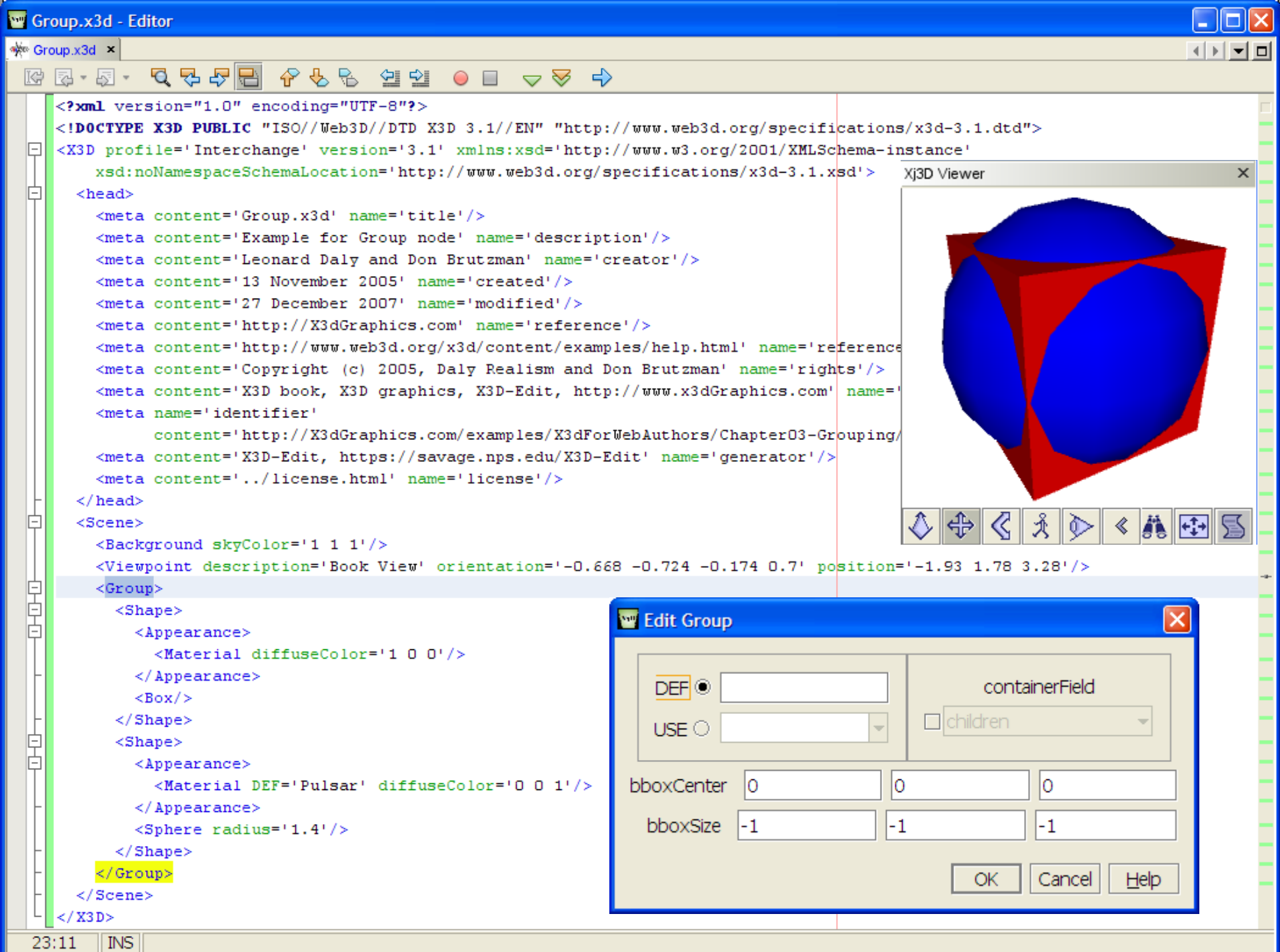
Can make USE copies if a DEF is provided


- Example: 4 identical tires on a car

Simplify editing

- X3D-Edit: collapse node using + - ticks in margin
- mouse-over to show hidden contents





 Group	<p>Group is a Grouping node that can contain most nodes.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>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.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

StaticGroup node

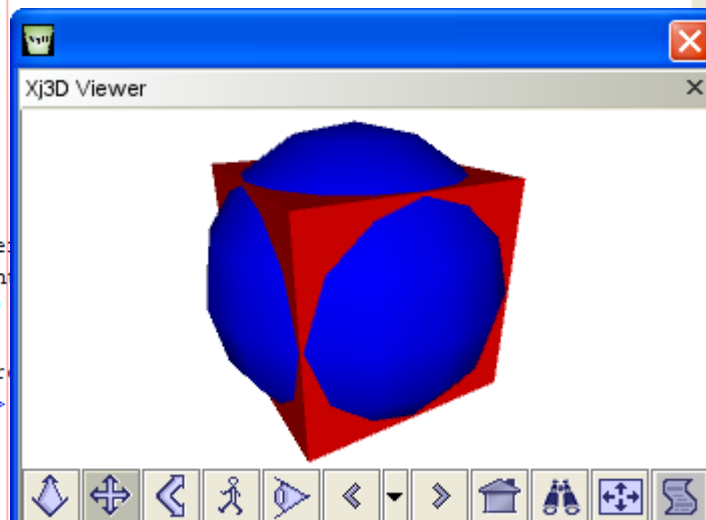
Identical to Group, except that children are not allowed to change or be animated

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='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='rights' />
    <meta name='identifier'
      content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/StaticGroup.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' />
    <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 ☒ containerField

USE ☐ ☐ children

bboxCenter

bboxSize

OK Cancel Help

 StaticGroup	<p>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.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>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.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

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

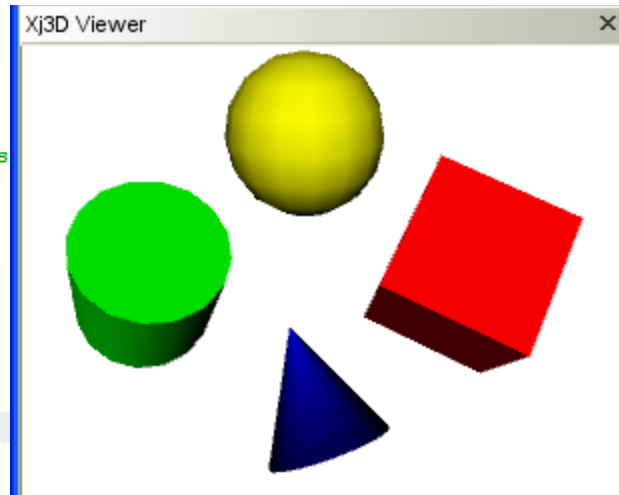
- *translation:* x y z movement in meters from origin of local coordinate system
- *rotation:* [x y z axis]-angle rotation about origin of local coordinate system
- *scale:* x y z (potentially nonuniform) factor for change in object scale to make it larger or smaller
- *center:* origin offset prior to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



```

<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 ☒ containerField

USE ☐ ☐ children

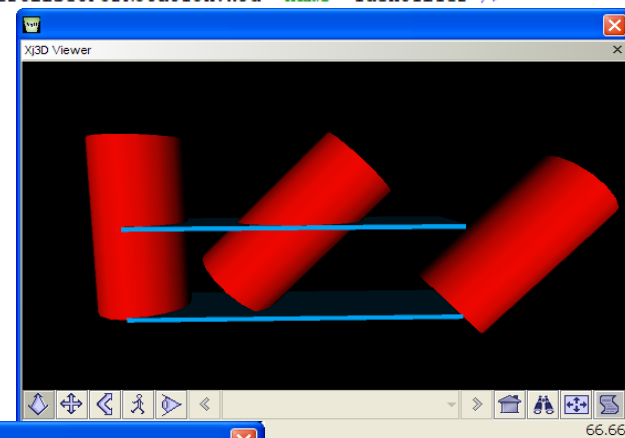
translation	0	2	0
rotation	0	0	1 0
center	0	0	0
scale	1	1	1
scaleOrientation	0	0	1 0
bboxCenter	0	0	0
bboxSize	-1	-1	-1

OK Cancel Help

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3 <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
4     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
5   <head>
6     <meta content='TransformCenterOffsetForRotation.x3d' name='title' />
7     <meta content='Use Transform center field to compare original Cylinder (left)
8       to rotation about middle of object (middle Cylinder)
9       to rotation about bottom of object (right Cylinder).' name='description' />
10    <meta content='Don Brutzman' name='creator' />
11    <meta content='28 January 2008' name='created' />
12    <meta content='28 January 2008' name='modified' />
13    <meta content='http://X3dGraphics.com' name='reference' />
14    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
15    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights' />
16    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
17    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/TransformCenterOffsetForRotation.x3d' name='identifier' />
18    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
19    <meta content='../license.html' name='license' />
20  </head>
21  <Scene>
22    <!-- leftmost cylinder is vertical -->
23    <Transform translation='-3.5 0 0'>
24      <Shape DEF='TallCylinder'>
25        <Cylinder height='4' />
26        <Appearance>
27          <Material diffuseColor='0.941176 0.027451 0' />
28        </Appearance>
29      </Shape>
30    </Transform>
31    <!-- center cylinder rotated -45 degrees about center -->
32    <Transform rotation='0 0 1 -0.78'>
33      <Shape USE='TallCylinder' />
34    </Transform>
35    <!-- rightmost cylinder rotated -45 degrees about base -->
36    <Transform center='0 -2 0' rotation='0 0 1 -0.78' translation='3.5 0 0'>
37      <Shape USE='TallCylinder' />
38    </Transform>
39    <Shape DEF='FlatBox'>
40      <Box size='7 0.1 2' />
41      <Appearance>
42        <Material diffuseColor='0 0.643137 0.960784' />
43      </Appearance>
44    </Shape>
45    <Transform translation='0 -2 0'>
46      <Shape USE='FlatBox' />
47    </Transform>
48  </Scene>
49 </X3D>

```



Edit Transform

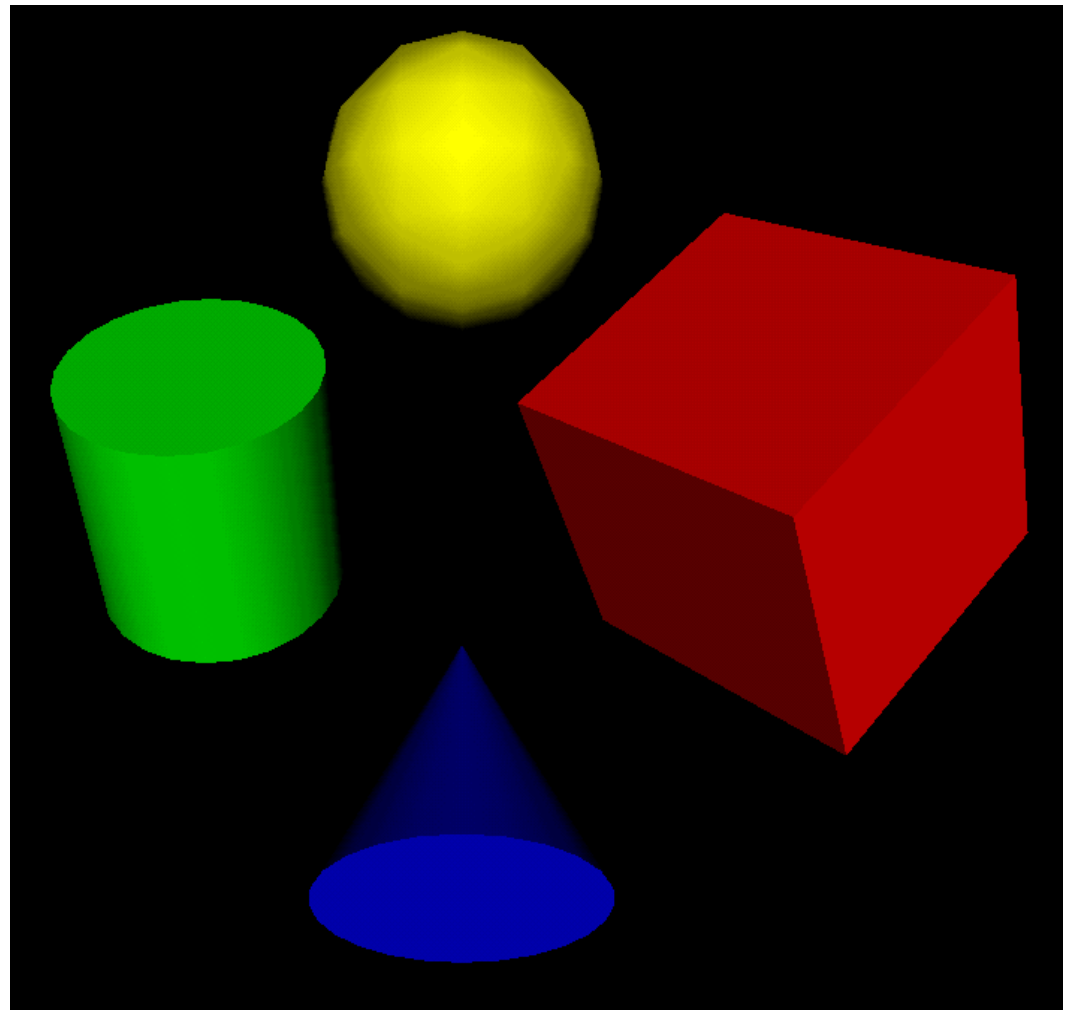
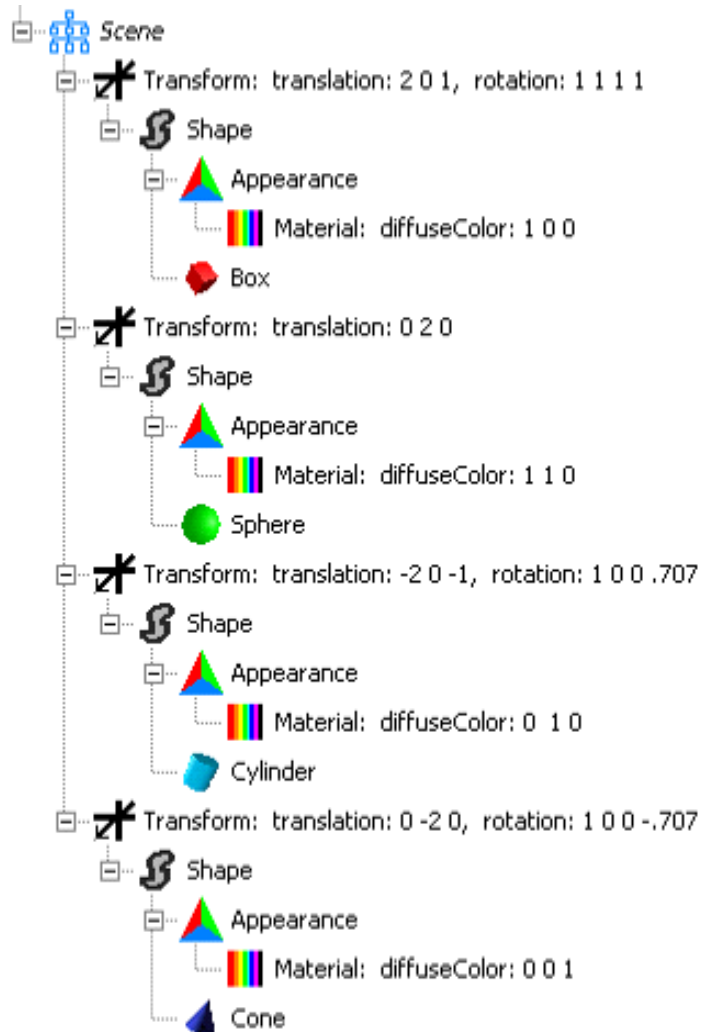
DEF ☐ containerField

USE ☐ ☐ 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

Each Transform is a scene subgraph



Order of transformations

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

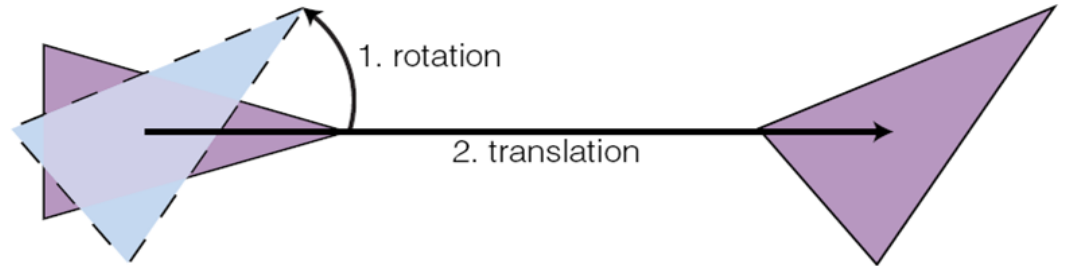
- Apply reverse *center* offset to set up for properly centered scaling and orientation operations
- Apply reverse *scaleOrientation*, then apply *scale* operation, then apply forward *scaleOrientation* to regain initial frame
- Apply *rotation* to final direction, then apply forward *center* offset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

When in doubt, nest multiple Transform nodes

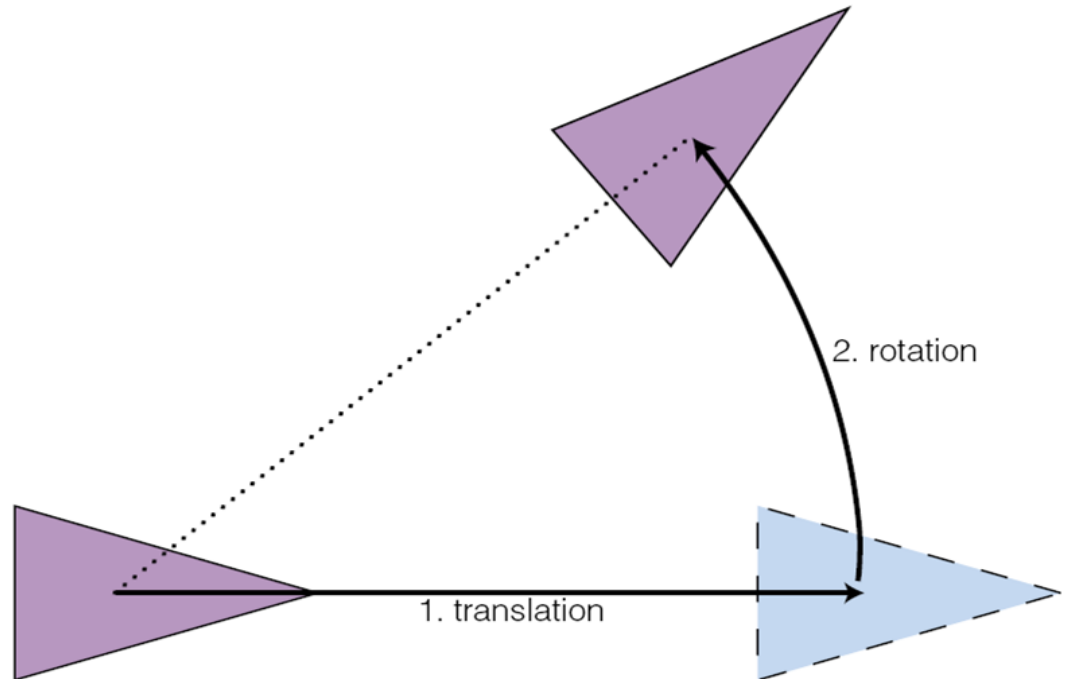
- and perform each step in order individually

Comparing out-of-order operations

Case 1



Case 2



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**
- $\mathbf{P'} = \mathbf{T} \cdot \mathbf{C} \cdot \mathbf{R} \cdot \mathbf{SR} \cdot \mathbf{S} \cdot -\mathbf{SR} \cdot -\mathbf{C} \cdot \mathbf{P}$

```

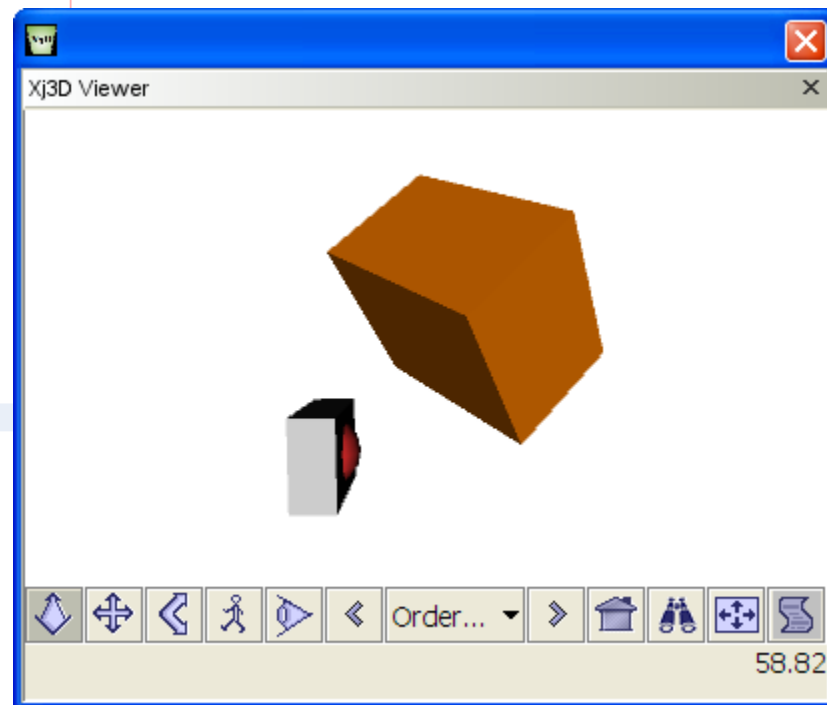
Transform {
    translation T
    children Transform {
        translation C
        children Transform {
            rotation R
            children Transform {
                rotation SR
                children Transform {
                    scale S
                    children Transform {
    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

```

<?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>

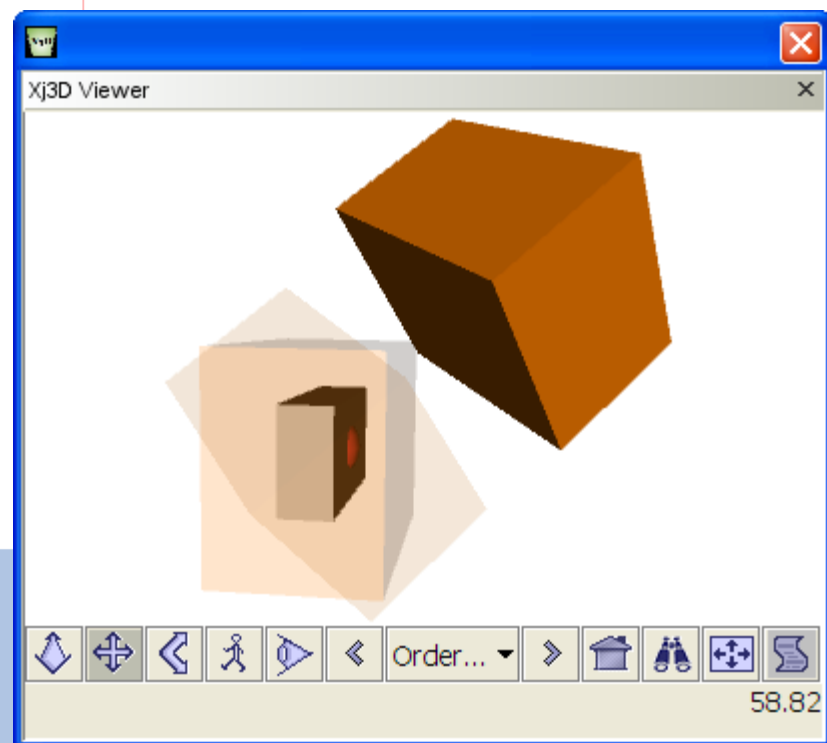
<Box size='.3 .6 .9'/>


</Shape>

</Transform>

</Scene>

</X3D>



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

Inline node

Loads another X3D world within current scene

- Supported formats depend on user's X3D browser
- XML .x3d, ClassicVRML .x3dv,
- Compressed binary .x3db, possibly VRML97 .wrl

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

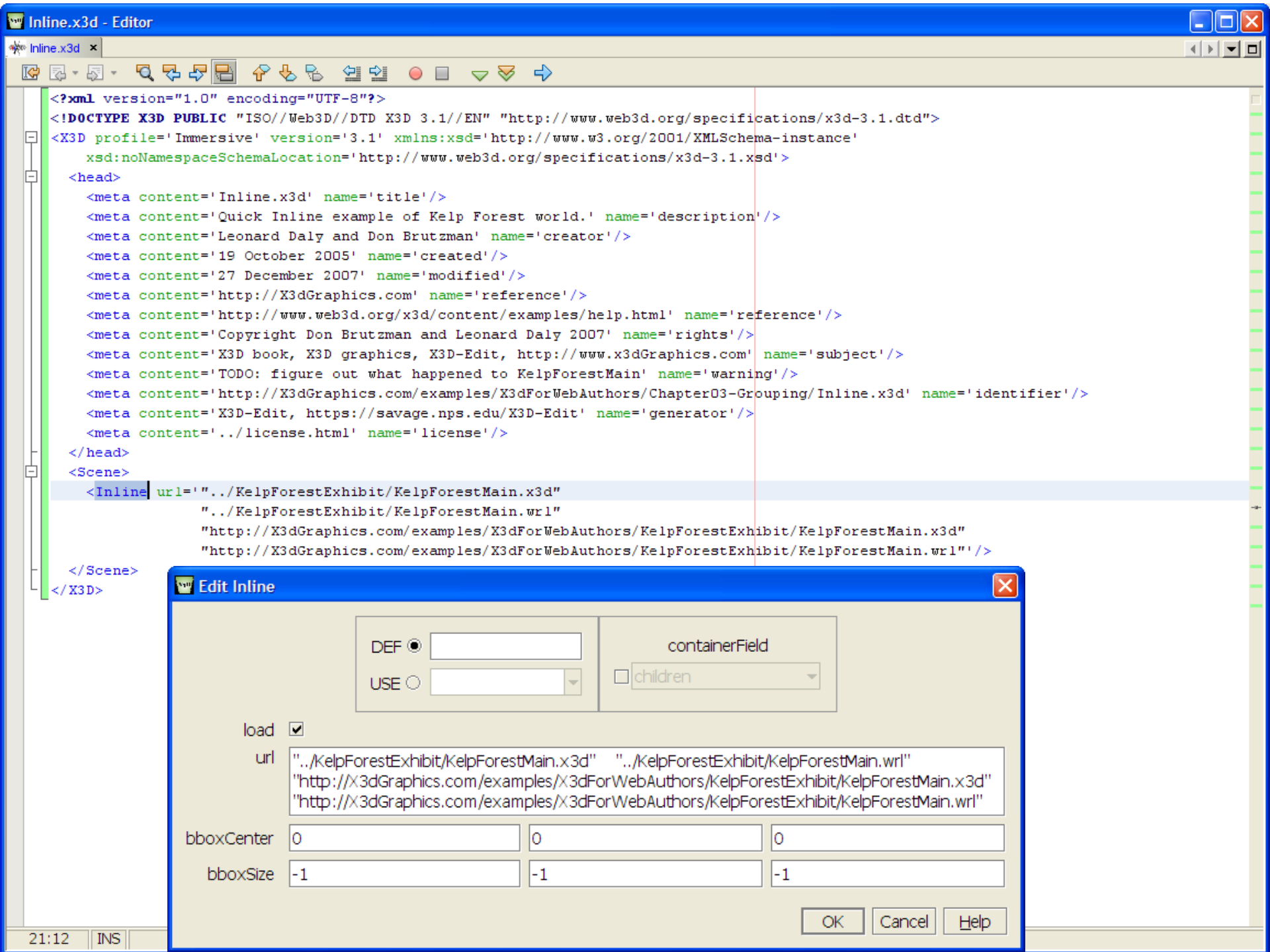
url field


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



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

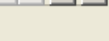
LOD (level of detail) node

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'

[illegible][illegible]

--	--



11



11



11





[illegible]



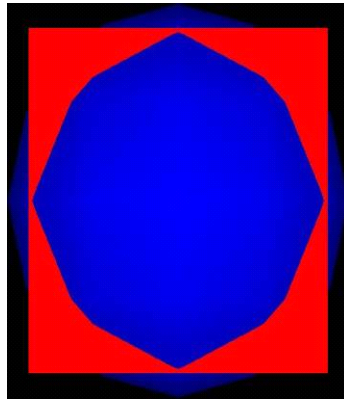
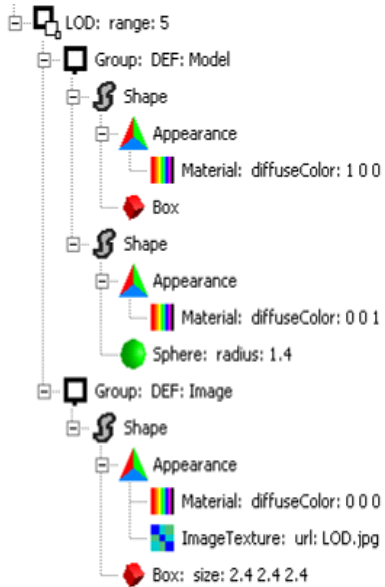
☒ DEF
☐ USE

containerField
☐ children

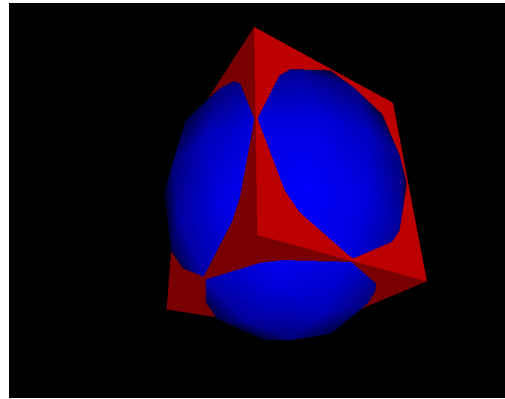
center
 range
 forceTransitions ☐
 bboxCenter
 bboxSize

OK Cancel Help

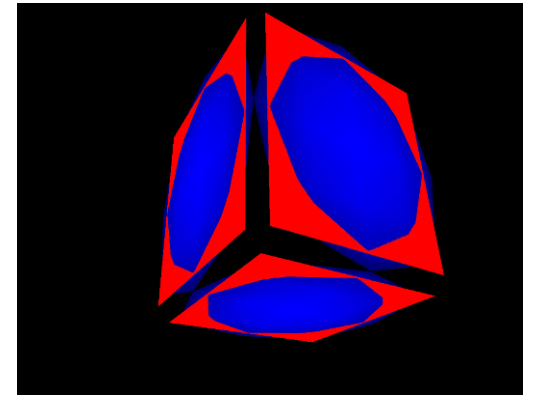
Example LOD range transitions



LOD.jpg



Near view



Far view

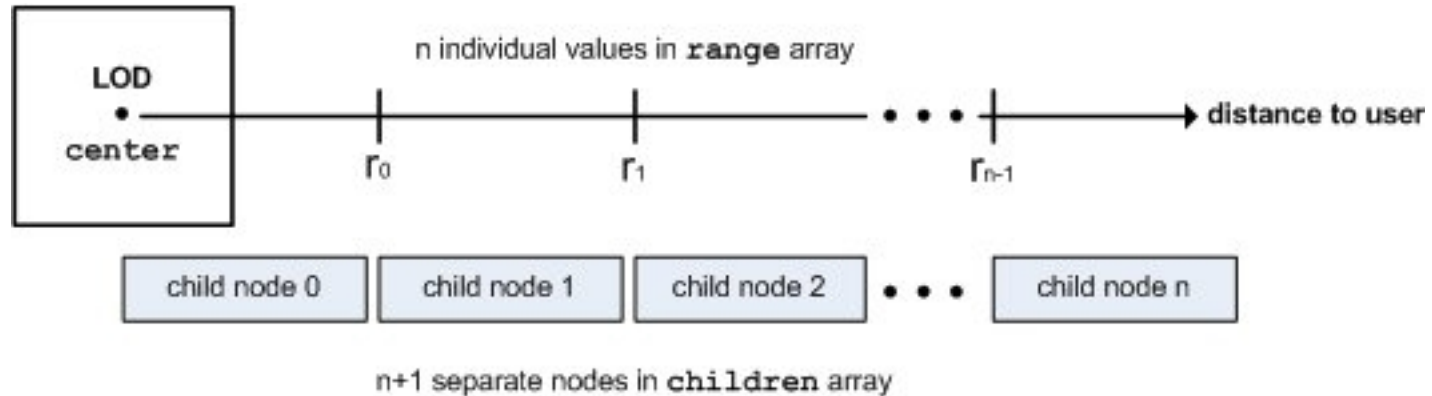
Each child of LOD should represent the same object

- Use Switch to change between different objects

Small difference between X3D and VRML97

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


LOD range transitions



Must have one more child than 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	<p>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.</p> <p>Hint: can add <code><WorldInfo info='null node'/></code> as nonrendering final child.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
forceTransitions	<p>[forceTransitions: accessType initializeOnly, type SFBool (true false) "false"]</p> <p>Whether to perform every range-based transition, regardless of browser optimizations that might otherwise occur.</p>
center	<p>[center: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Position offset from origin of local coordinate system.</p>
range	<p>[range: accessType initializeOnly, type MFFloat CDATA #IMPLIED]</p> <p>(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!</p> <p>Hint: can add <code><WorldInfo info='null node'/></code> as nonrendering final child.</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
level_changed	<p>[level_changed: accessType outputOnly, type SFInt32 CDATA #FIXED ""]</p> <p>Indicates current level of LOD children when activated.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>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.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

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
<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 <input checked="" type="radio"/>	Switcher	containerField	
USE <input type="radio"/>	Switcher	<input type="checkbox"/>	children

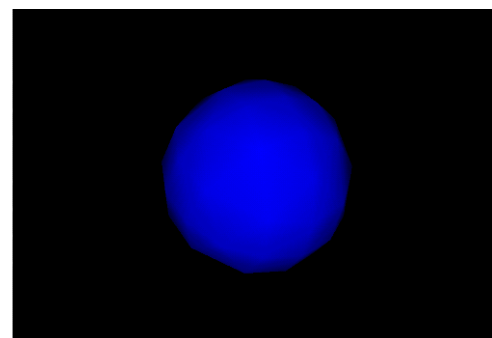
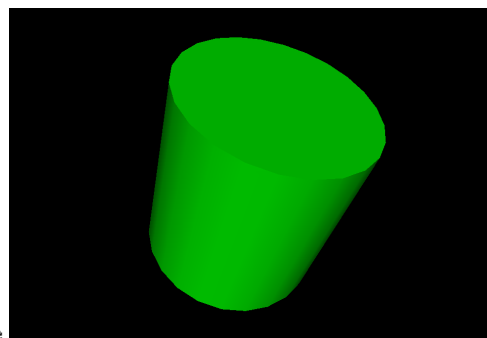
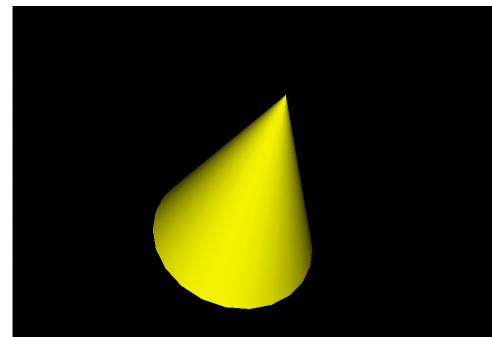
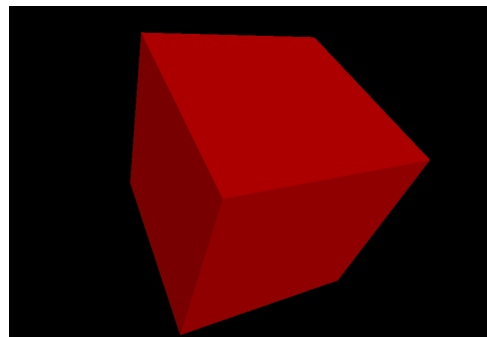
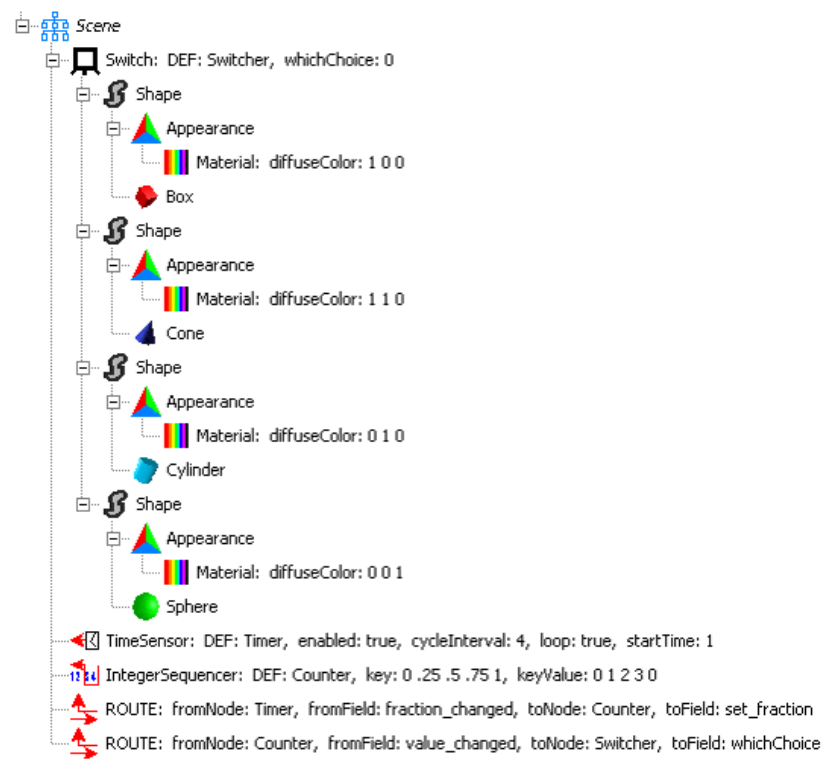
whichChoice

bboxCenter

bboxSize


OK Cancel Help

Switch node example



Note *whichChoice* starts at index 0; -1 means none

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

 Switch	<p>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.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p> <p>Hint: authors can temporarily hide test geometry under an unselected child of a Switch. This is a good alternative to "commenting out" nodes.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
whichChoice	<p>[whichChoice: accessType inputOutput, type SFInt32 CDATA "-1"]</p> <p>Index of active child choice, counting from 0.</p> <p>Warning: default whichChoice= -1 means no selection (and no rendering), whichChoice=0 means initial child.</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>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.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

Additional Resources

Additional Resources

3D grids for object placement

- Provided by many 3D authoring environments
- Grid prototypes in Savage tools for authoring
<https://savage.nps.edu/Savage/Tools/Authoring>

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>

Chapter Summary

Chapter Summary

Grouping nodes collect and select other nodes

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 in Chapter 4
 - Anchor, Billboard, Collision

[back to Table of Contents](#)

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 Examples Help

- <http://www.web3d.org/x3d/content/examples/help.html>

References 2

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

Contact

Don Brutzman

brutzman@nps.edu

<http://web.nps.navy.mil/~brutzman>

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

1.831.656.2149 voice

1.831.656.7599 fax

Open-source license

Copyright (c) 1995-2008 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



Contents

Chapter Overview and Concepts

X3D Nodes and Examples

Additional Resources

Chapter Summary

References



Chapter Overview



Overview: grouping nodes

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

Tree

Directed acyclic graph (DAG)

Node

Parent-child relationship

Subgraph

Leaf node

Intermediate or internal node



TODO: add definitions. These are good discussion topics.

XML and X3D correspondence

	Opening element	<Shape>
Singleton element, attribute="value"	Opening element	<Sphere radius="10.0" solid="true"/>
Singleton element, attribute='value'	Closing element	</Appearance>
	Closing element	</Shape>

Review

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



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

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
- Provides target for routing events

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!

- 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



Test: 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.

The X3D Scene Authoring Hints include guidance on good naming conventions. These 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

- $1 \text{ m} = 39.3''$

Angular measurements in radians

- $2 \pi = 360 \text{ degrees}$

Time measured in seconds

- Starting 1 January 1970



Warning: using degree values rather than radians is a common mistake by new students.

This time convention is quite common and ultimately inherited from the Unix operating system.

Coordinate systems

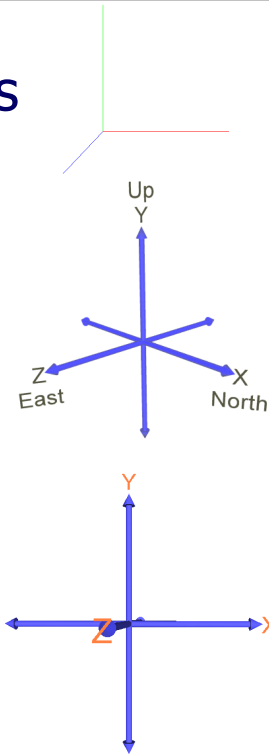
Right hand rule for X Y Z order

Y axis is up

Correspondence: North, Up, East

Accept no substitutes!

- or at least realign them ☺



There are a total of eight different Euler angle systems.

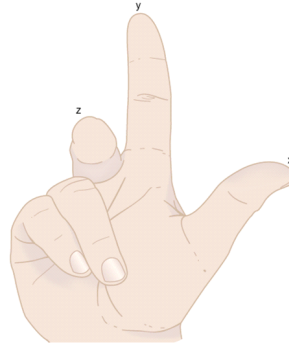
Half of these follow a left-hand rule, rather than a right-hand rule. Occasionally a graphics book comes out that presents mathematical equations using a left-hand rule. Immediately throw such books in the fire so that further pain and suffering is prevented!

The displayed example is

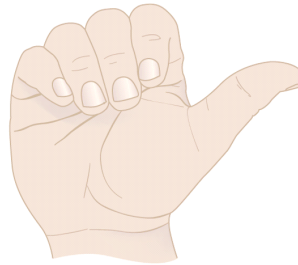
<http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/CoordinateAxesNSEW.x3d>

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



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

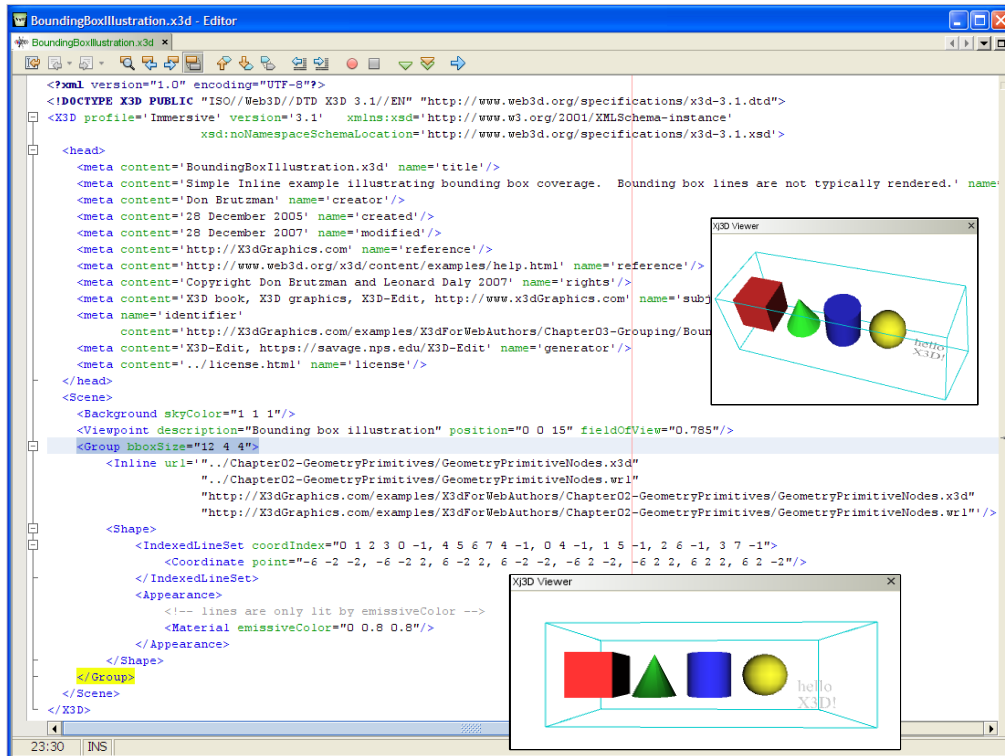
- browser avoids calculating impossible collisions

Bounding boxes can be ignored by authors

- some authoring tools can provide them if needed



Note that bounding boxes are invisible and not displayed.



Note that bounding boxes are invisible and not displayed. This wireframe has been explicitly added to the scene to illustrate bounding box principles.

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

[back to Table of Contents](#)

X3D Nodes and Examples



Group node

Collects nodes together with related purpose

- Often close to each other spatially

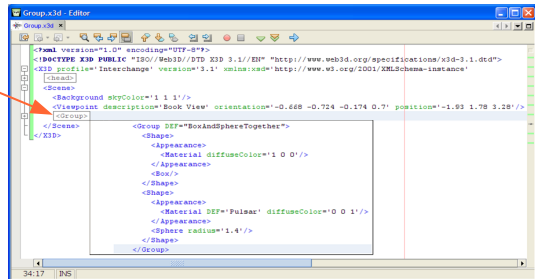
Can make USE copies if a DEF is provided

- Example: 4 identical tires on a car

Simplify editing

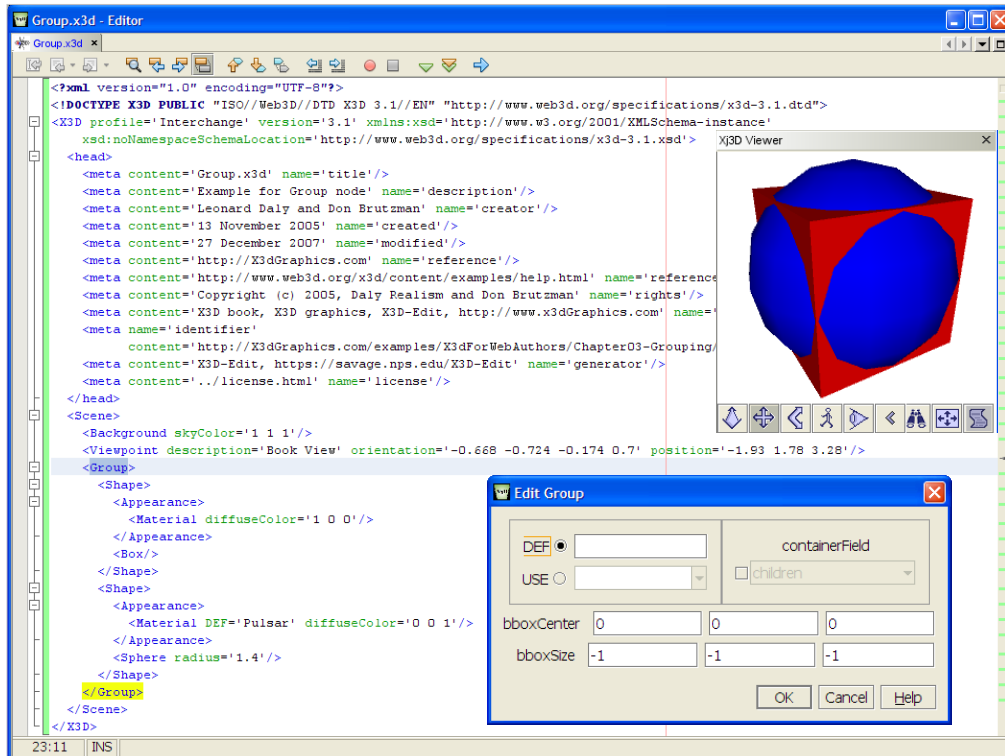
- X3D-Edit: collapse node using + - ticks in margin
- mouse-over to show hidden contents

web|3D
CONSORTIUM




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





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

 Group	<p>Group is a Grouping node that can contain most nodes.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>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.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

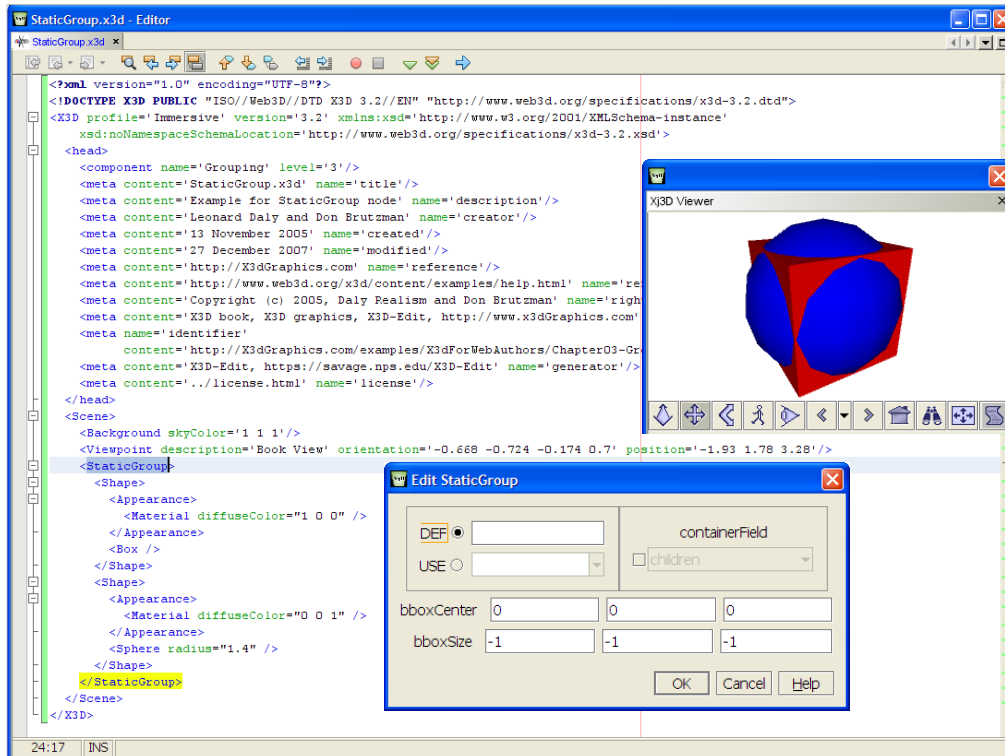
<http://www.web3d.org/x3d/content/X3dTooltips.html#Group>

StaticGroup node

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

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





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

<div> <div></div> <div>StaticGroup</div> </div>	<p>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.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>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.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

<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

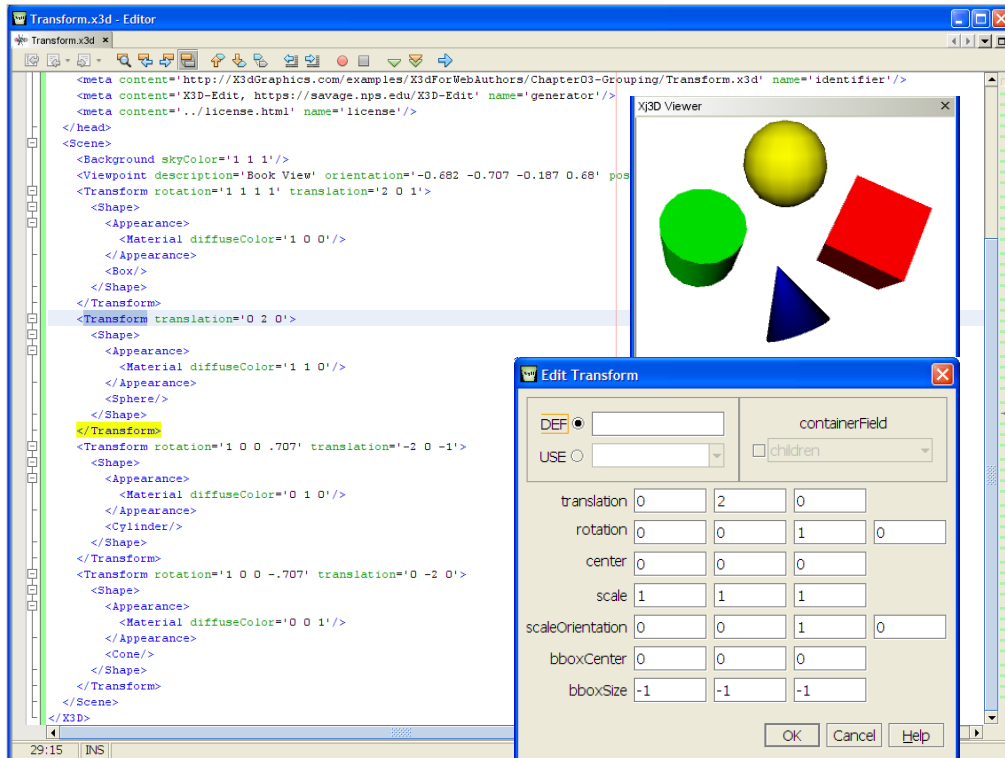
- Translate local origin linearly to another coordinate
- Rotate about any axis
- Scale size, uniformly or separately along x y z axes

Group and Transform are among most commonly used nodes

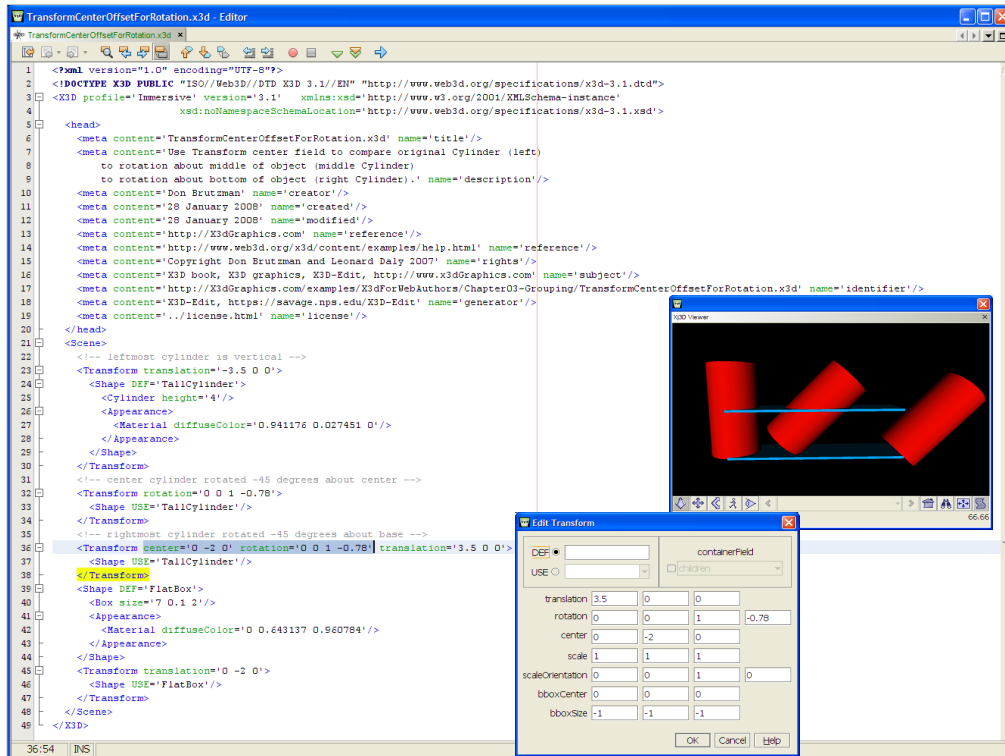


Transform fields

- *translation*: x y z movement in meters from origin of local coordinate system
- *rotation*: [x y z axis]-angle rotation about origin of local coordinate system
- *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*: rotation to apply prior to scaling
- *bboxCenter*, *bboxSize*: bounding box information (if any is provided by author, optional)

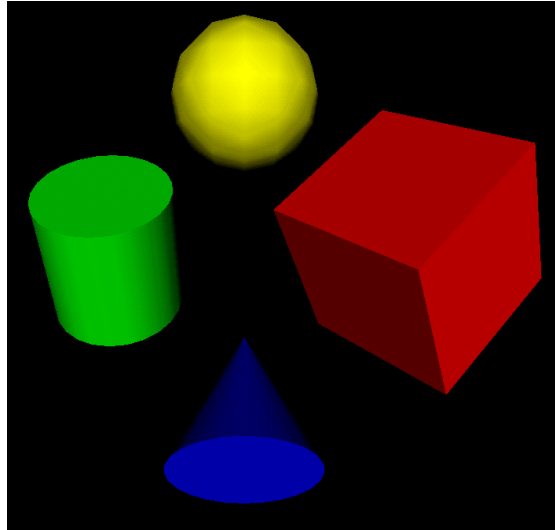
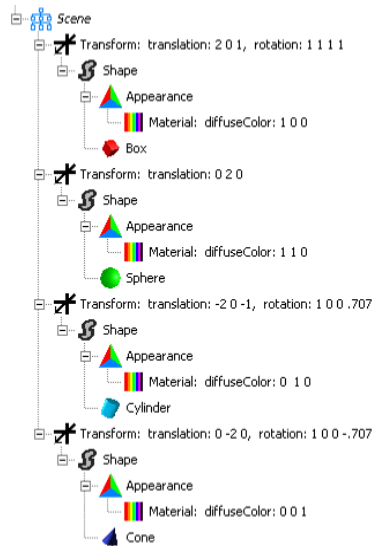


<http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter03-Grouping/Transform.x3d>



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

Each Transform is a scene subgraph



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

Order of transformations

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



The next slide illustrates these steps.

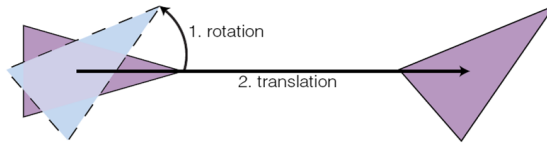
Click to add title

When in doubt, nest multiple Transform nodes

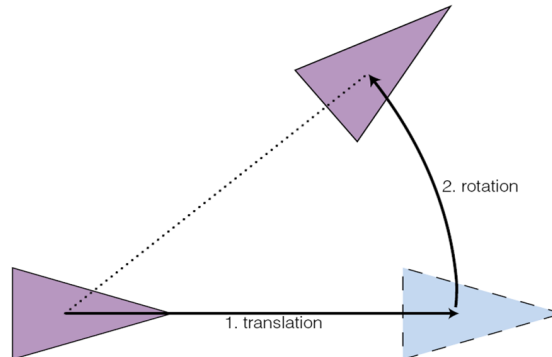
- and perform each step in order individually

Comparing out-of-order operations

Case 1



Case 2



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**
- $P' = T \cdot C \cdot R \cdot SR \cdot S \cdot -SR \cdot -C \cdot P$

```

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 [...]
            }
          }
        }
      }
    }
  }
}

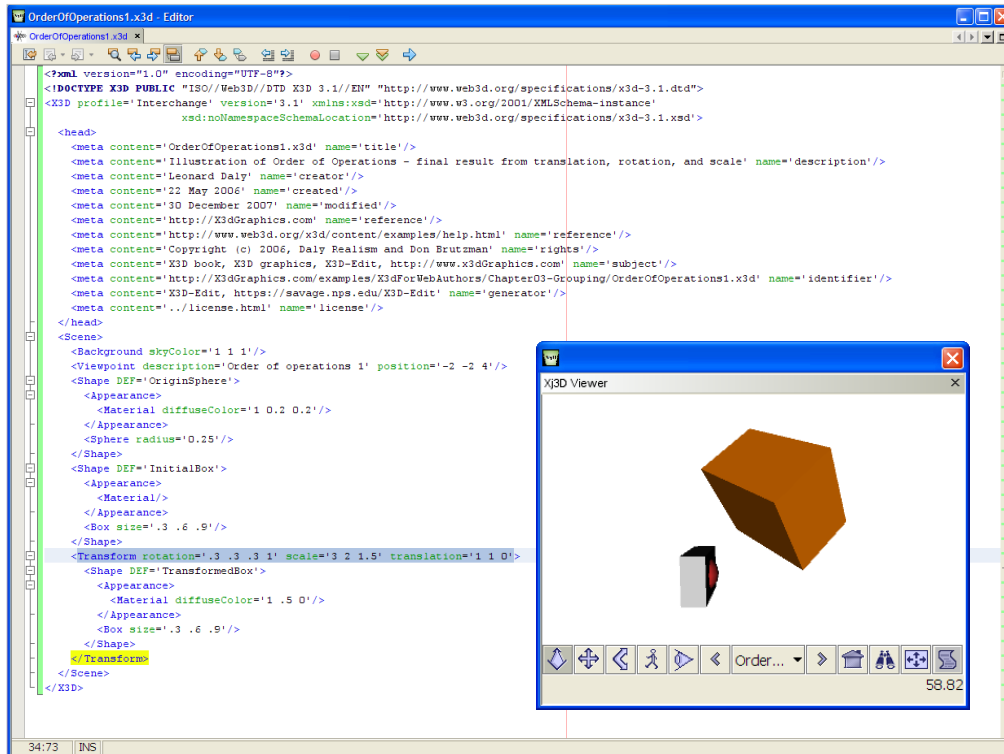
```

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.

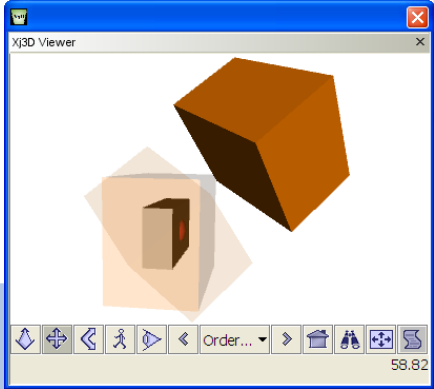
Summarizing: first apply center and scaling operations, then rotation, then translation.



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>



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

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

Chapter03-Grouping

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

<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 .x3d, ClassicVRML .x3dv,
- Compressed binary .x3db, possibly VRML97 .wrl

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

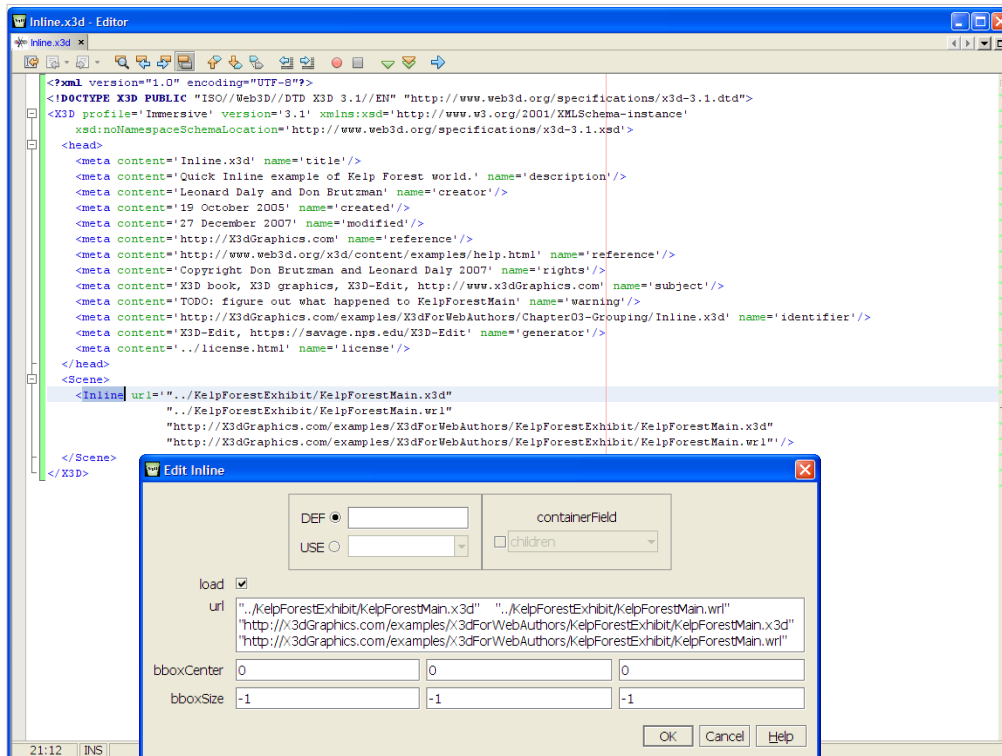
- Equivalent to universal resource identifier (uri)

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




The *url* field is also used by a number of other nodes, such as ImageTexture and MovieTexture in Chapter 5.



An improved url editor is planned for X3D-Edit.

<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 " (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

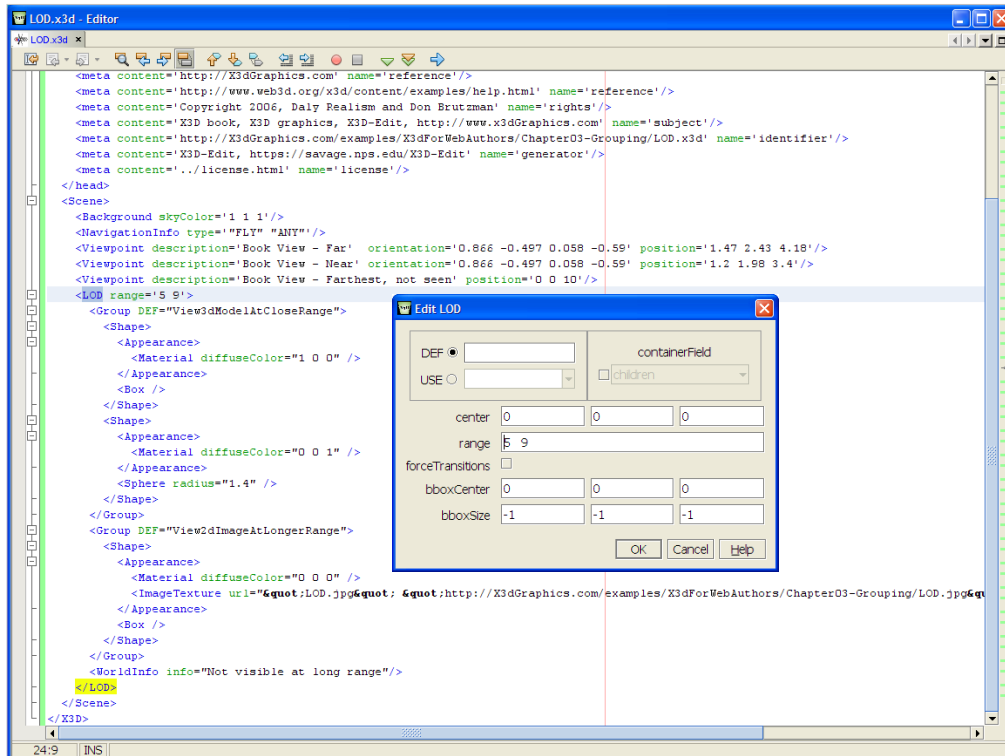
LOD holds multiple versions of same geometry

- 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'



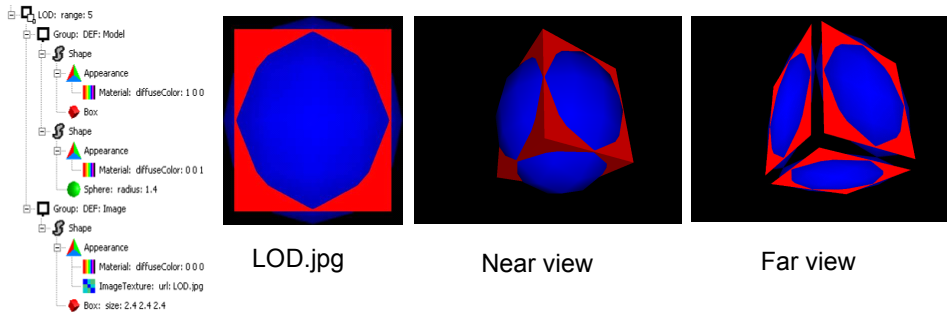


Animate the scene by zooming in and out to watch transitions.

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



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'

web|3D
CONSORTIUM



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

Scene:

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

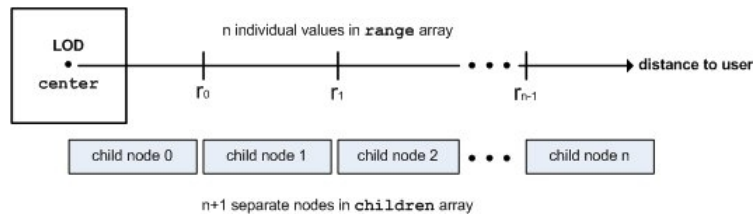
ImageTexture:

<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 from VRML97 in order to make LOD consistent with other X3DGroupingNode types.

LOD range transitions



Must have one more child than 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

web|3D
CONSORTIUM



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

 LOD	<p>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.</p>
DEF	<p>[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.</p>
USE	<p>[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!</p>
forceTransitions	<p>[forceTransitions: accessType initializeOnly, type SFBBool (true false) "false"] Whether to perform every range-based transition, regardless of browser optimizations that might otherwise occur.</p>
center	<p>[center: accessType initializeOnly, type SFVec3f CDATA "0 0 0"] Position offset from origin of local coordinate system.</p>
range	<p>[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.</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"] Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"] Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
level_changed	<p>[level_changed: accessType outputOnly, type SFInt32 CDATA #FIXED ""] Indicates current level of LOD children when activated.</p>
containerField	<p>[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.</p>
class	<p>[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.</p>

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

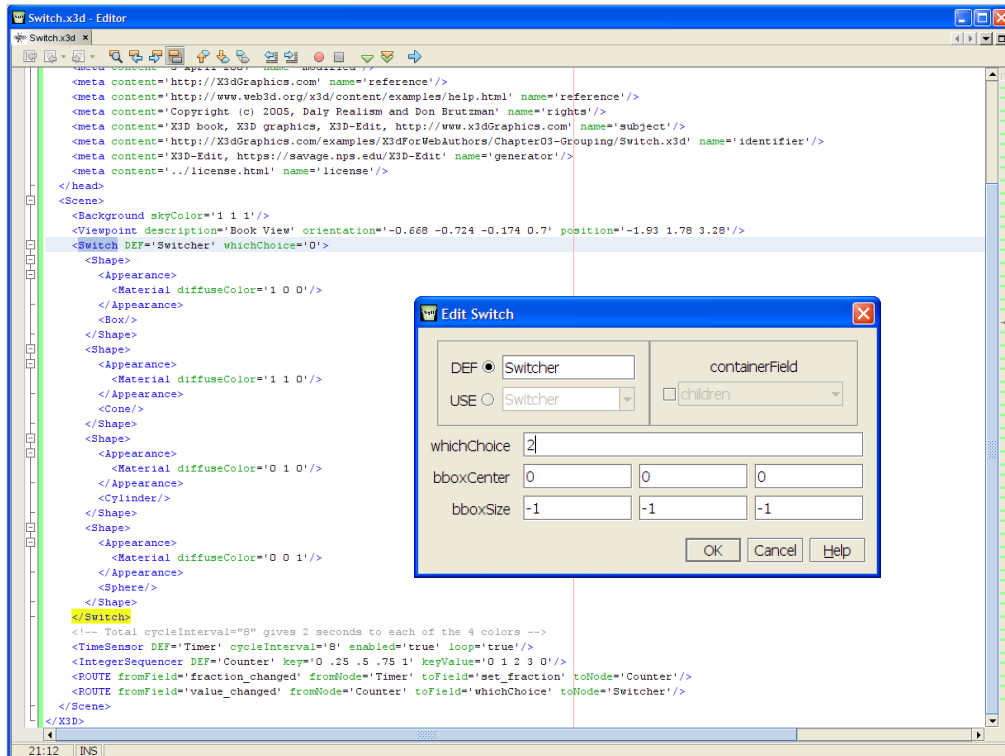
Switch node

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

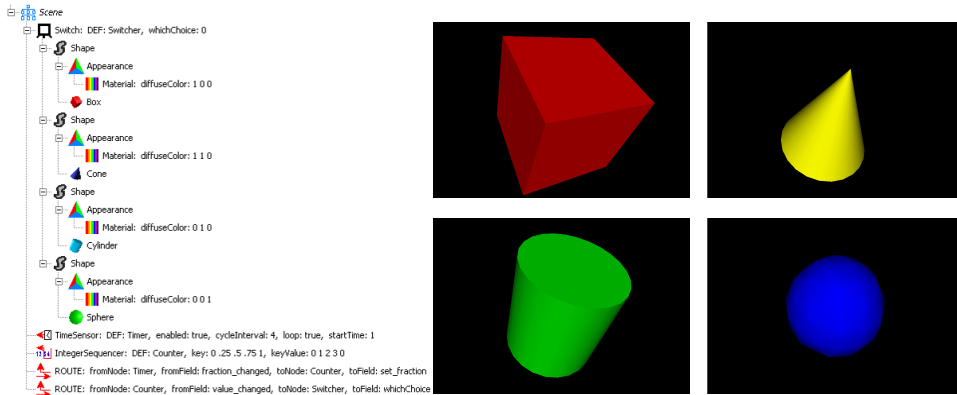


These scene has an added built-in animation to show the Switch in operation.
The Switch node will not change whichChoice otherwise.

Animation is explained in Chapter07-EventAnimation

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

Switch node example



Note *whichChoice* starts at index 0; -1 means none

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

web|3D
CONSORTIUM




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

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

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

This value was changed in X3D from VRML97 in order to make Switch consistent with other X3DGroupingNode types.

 Switch	<p>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.</p> <p>Hint: insert a Shape node before adding geometry or Appearance.</p> <p>Hint: authors can temporarily hide test geometry under an unselected child of a Switch. This is a good alternative to "commenting out" nodes.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p>Hint: descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
whichChoice	<p>[whichChoice: accessType inputOutput, type SFInt32 CDATA "-1"]</p> <p>Index of active child choice, counting from 0.</p> <p>Warning: default whichChoice= -1 means no selection (and no rendering), whichChoice=0 means initial child.</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape.</p> <p>containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>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.</p>

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

[back to Table of Contents](#)

Additional Resources



Additional Resources

3D grids for object placement

- Provided by many 3D authoring environments
- Grid prototypes in Savage tools for authoring
<https://savage.nps.edu/Savage/Tools/Authoring>

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>



Rotation calculators:

Vapour tools for Windows, though no longer publicly available

[back to Table of Contents](#)

Chapter Summary



Chapter Summary

Grouping nodes collect and select other nodes

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 in Chapter 4
 - Anchor, Billboard, Collision



[back to Table of Contents](#)

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 Examples Help

- <http://www.web3d.org/x3d/content/examples/help.html>



References 2

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



Contact

Don Brutzman

brutzman@nps.edu

<http://web.nps.navy.mil/~brutzman>

Code USW/Br, Naval Postgraduate School

Monterey California 93943-5000 USA

1.831.656.2149 voice

1.831.656.7599 fax



Open-source license

Copyright (c) 1995-2008 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.