X3D Graphics for Web Authors

Chapter 5

Appearance, Material, and Textures

Things are not always as they appear.

web|3D CONSORTIUM

<X3D>

Contents

Chapter Overview and Concepts

X3D Nodes and Examples

MultiTextures and Additional Resources

Chapter Summary and Suggested Exercises

References

Chapter Overview

Overview: Appearance, Material, and Textures

Appearance affects associated geometry, containing the following fields

Visual surface properties that interact with lights

- Material and TwoSidedMaterial
- LineProperties and FillProperties

Texture nodes wrap images onto geometry

- ImageTexture, MovieTexture, PixelTexture and MultiTexture
- TextureTransform, TextureCoordinate and TextureCoordinateGenerator

Concepts

Motivation

Appearance, material and texture nodes are intended to allow authors to make 3D objects look similar to objects in the real world

- This goal is always a worthy challenge

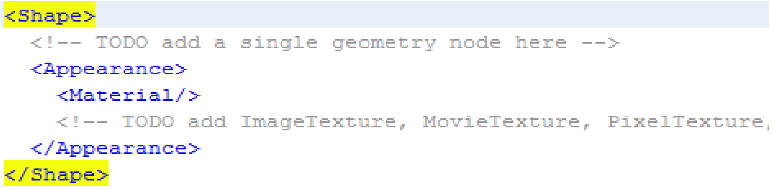
Lighting is an important factor in appearance, because 3D objects reflect their virtual light

- Appearance and lighting are computational
- In this chapter we assume white light available, usually from default NavigationInfo headlight
- Lighting and environment covered in Chapter 11

Parent-child constraints

Each Shape node can contain

- Single geometry node
- Single Appearance node



```
<Shape>
  <!-- TODO add a single geometry node here -->
  <Appearance>
    <Material/>
    <!-- TODO add ImageTexture, MovieTexture, PixelTexture,
  </Appearance>
</Shape>
```

Each Appearance node can contain

- A single Material (or TwoSidedMaterial) node
- FillProperties, LineProperties, TextureProperties
- A single Texture node (image, pixel or movie)

Each Texture node can contain

- Single TextureTransform or TextureTransformGenerator node

Common functionality

Node repetition can be efficiently accomplished via *DEF* and *USE*

- Remember, first *DEF* must precede any *USE* copies
- Simplifies application of consistent coloring to multiple pieces of geometry which are either similar or parts of the same larger object

Consistent, more efficient, easier to globally change all instances at once

- Which is further important when changing styles or applying accessibility techniques throughout

X3D Nodes and Examples

Appearance node

Each Shape contains a single geometry node along with a corresponding Appearance node

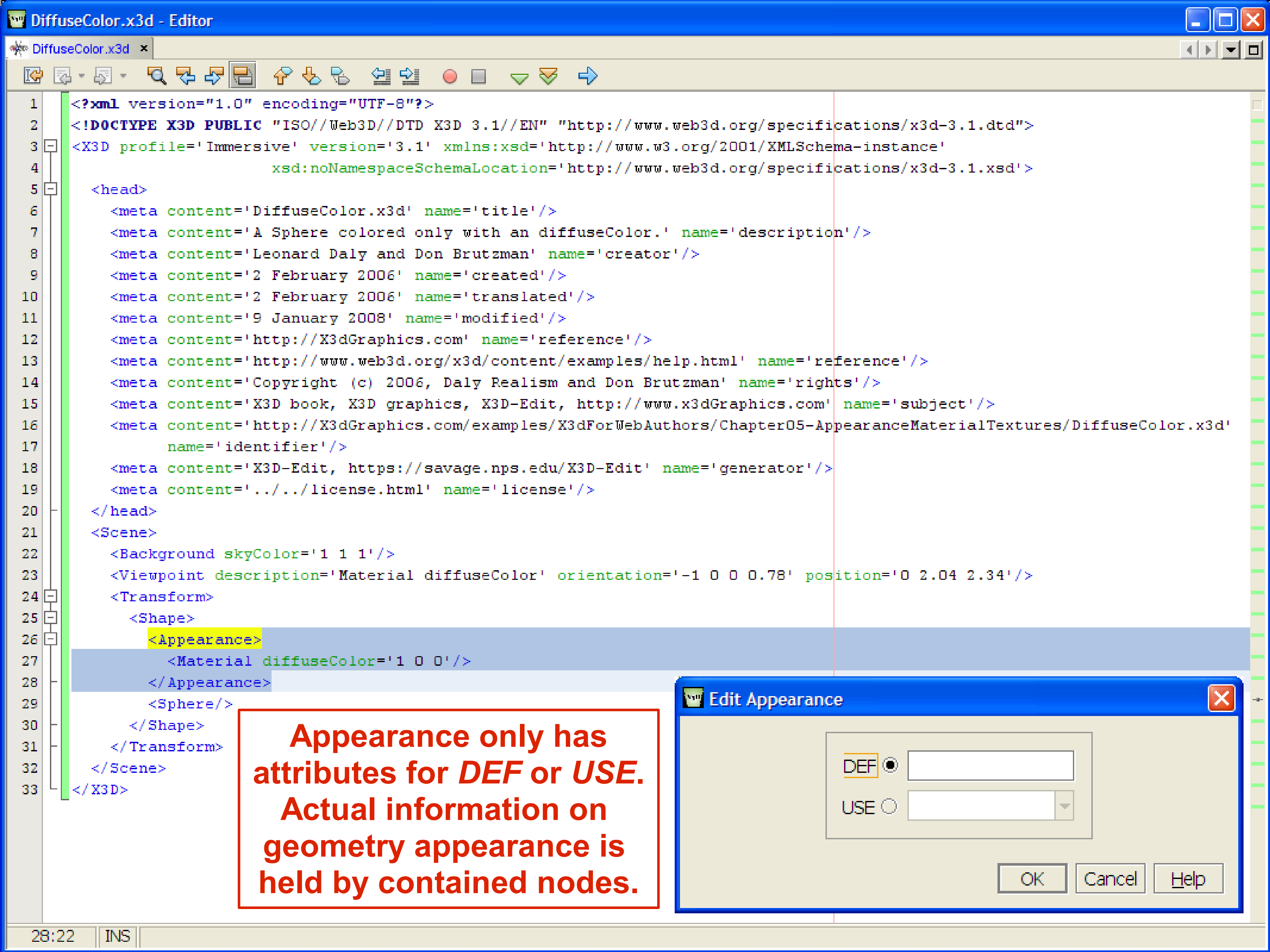
Appearance is a container which may include

- A single Material (or TwoSidedMaterial) node
- Fill/Line/Texture Properties, single Texture node

This close association makes assignment of rendering properties to geometry unambiguous

- Repetition of values for visual consistency is easily accomplished with *DEF/USE* of Appearance, Material, Texture node, etc.
- Clear naming helps, for example

```
<Appearance USE='FoggyGlassAppearance'/>
```



```
<?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='DiffuseColor.x3d' name='title'/>
    <meta content='A Sphere colored only with an diffuseColor.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='9 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 (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/Chapter05-AppearanceMaterialTextures/DiffuseColor.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='Material diffuseColor' orientation='-1 0 0 0.78' position='0 2.04 2.34'/>
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Appearance only has attributes for *DEF* or *USE*. Actual information on geometry appearance is held by contained nodes.

Edit Appearance

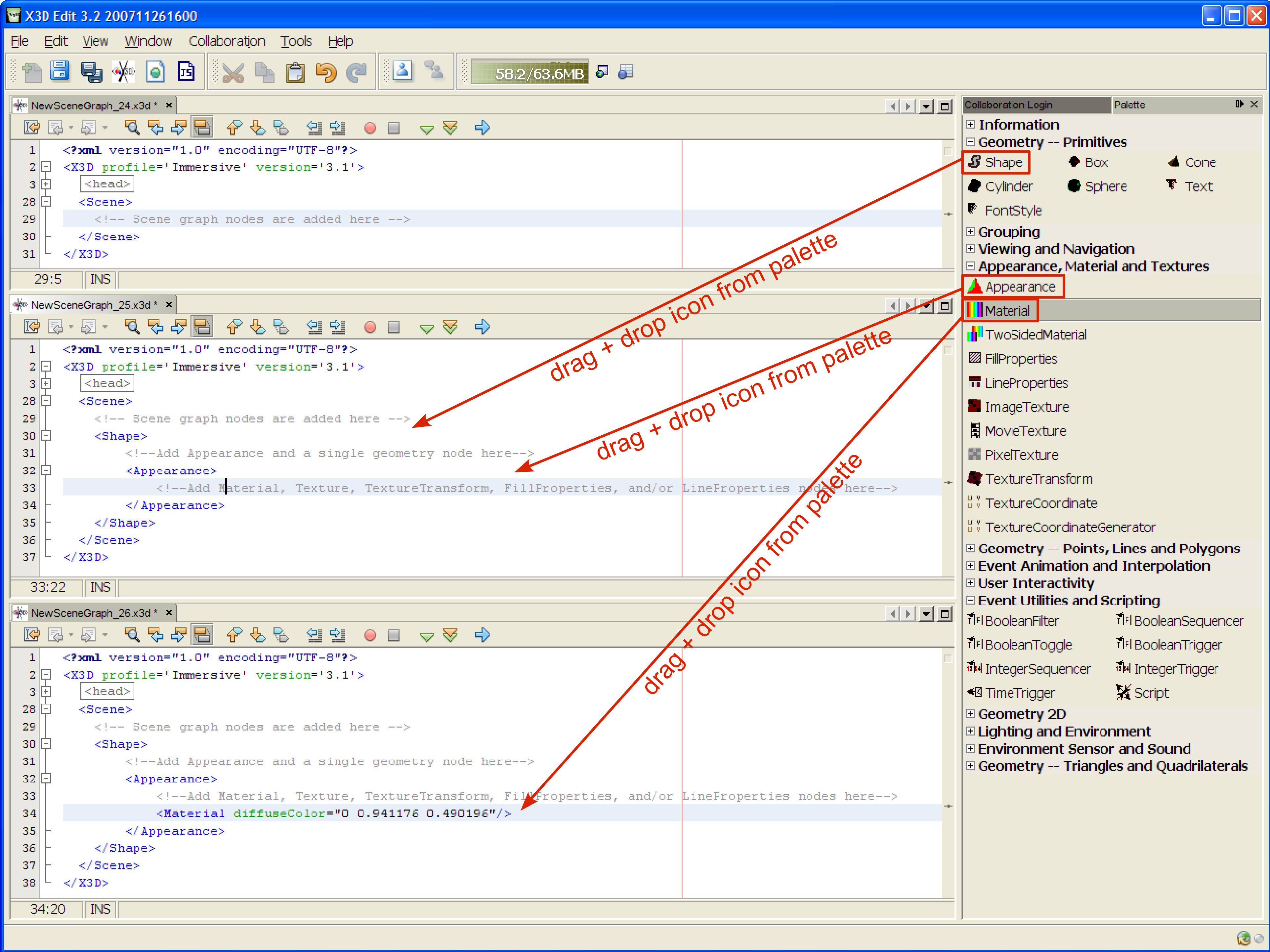
DEF ●

USE ○

Palette simplifies addition of new nodes

Use X3D-Edit palette to pick the node of interest:

- Palette groups match chapter structure, and can be reordered by dragging with mouse
- Upon dragging a new node element into scene, corresponding node editor pops up
- After checking attribute values with node editor, select OK to confirm the new node
- Default attribute values omitted in XML for clarity
- Erroneous node placement in scene graph, or invalid attribute values, cause a validation error
- Accept or reject validation errors as appropriate, then continue with text editing if desired



NewSceneGraph_24.x3d *

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' version='3.1'>
  <head>
  <Scene>
    <!-- Scene graph nodes are added here -->
  </Scene>
</X3D>
```

29:5 INS

NewSceneGraph_25.x3d *

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' version='3.1'>
  <head>
  <Scene>
    <!-- Scene graph nodes are added here -->
    <Shape>
        <!--Add Appearance and a single geometry node here-->
        <Appearance>
            <!--Add Material, Texture, TextureTransform, FillProperties, and/or LineProperties nodes here-->
        </Appearance>
    </Shape>
  </Scene>
</X3D>
```

33:22 INS

NewSceneGraph_26.x3d *

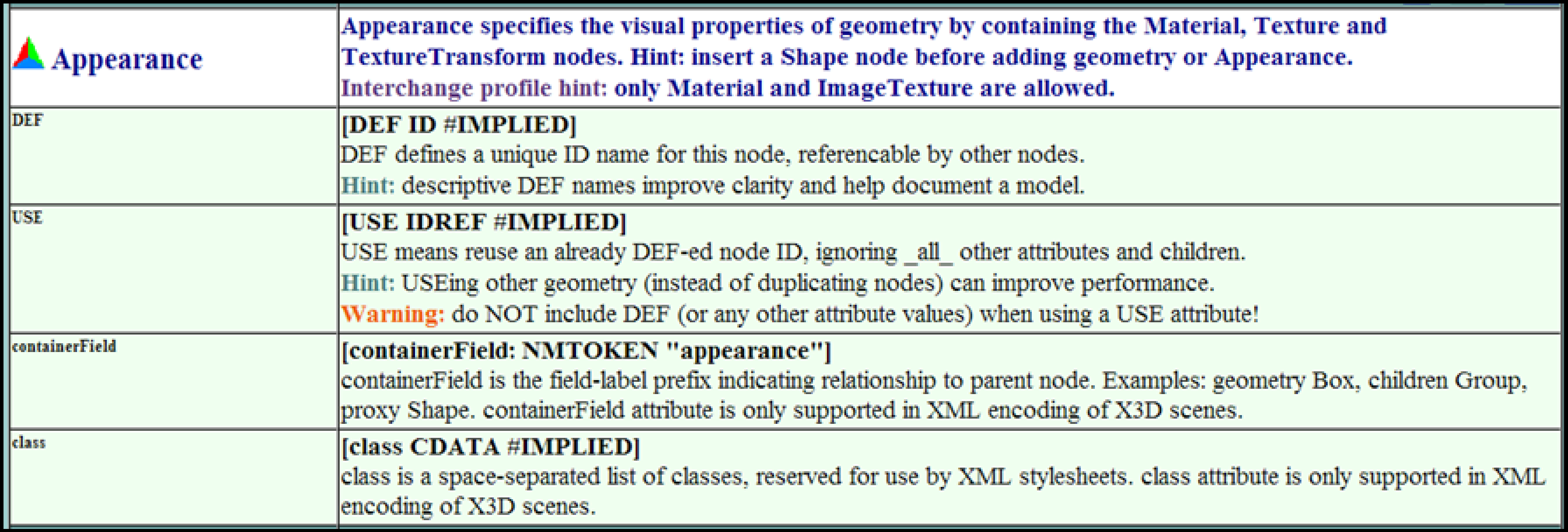
```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' version='3.1'>
  <head>
  <Scene>
    <!-- Scene graph nodes are added here -->
    <Shape>
        <!--Add Appearance and a single geometry node here-->
        <Appearance>
            <!--Add Material, Texture, TextureTransform, FillProperties, and/or LineProperties nodes here-->
            <Material diffuseColor="0 0.941176 0.490196"/>
        </Appearance>
    </Shape>
  </Scene>
</X3D>
```

34:20 INS

drag + drop icon from palette

drag + drop icon from palette

drag + drop icon from palette



Appearance	**Appearance specifies the visual properties of geometry by containing the Material, Texture and TextureTransform nodes. Hint: insert a Shape node before adding geometry or Appearance.** **Interchange profile hint: only Material and ImageTexture are allowed.**
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!
containerField	**[containerField: NMTOKEN "appearance"]** 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.

Material node

Material controls how most geometry is colored, whether it is transparent or glowing, etc.

Surface visual properties are applied equally across all polygons making up a shape

Material properties define how geometry visually interacts with light sources in the scene

- Lighting and Environment is covered in Chapter 11
- Rendering results also depend on view perspective

Material is an important node to master

Reading X3D Specification node signatures

Actual X3D Specification entries are as follows:

- SFFloat [in,out] *ambientIntensity* 0.2 [0,1]
- SFColor [in,out] *diffuseColor* 0.8 0.8 0.8 [0,1]
- SFColor [in,out] *emissiveColor* 0 0 0 [0,1]
- SFFloat [in,out] *shininess* 0.2 [0,1]
- SFColor [in,out] s*pecularColor* 0 0 0 [0,1]
- SFFloat [in,out] *transparency* 0 [0,1]

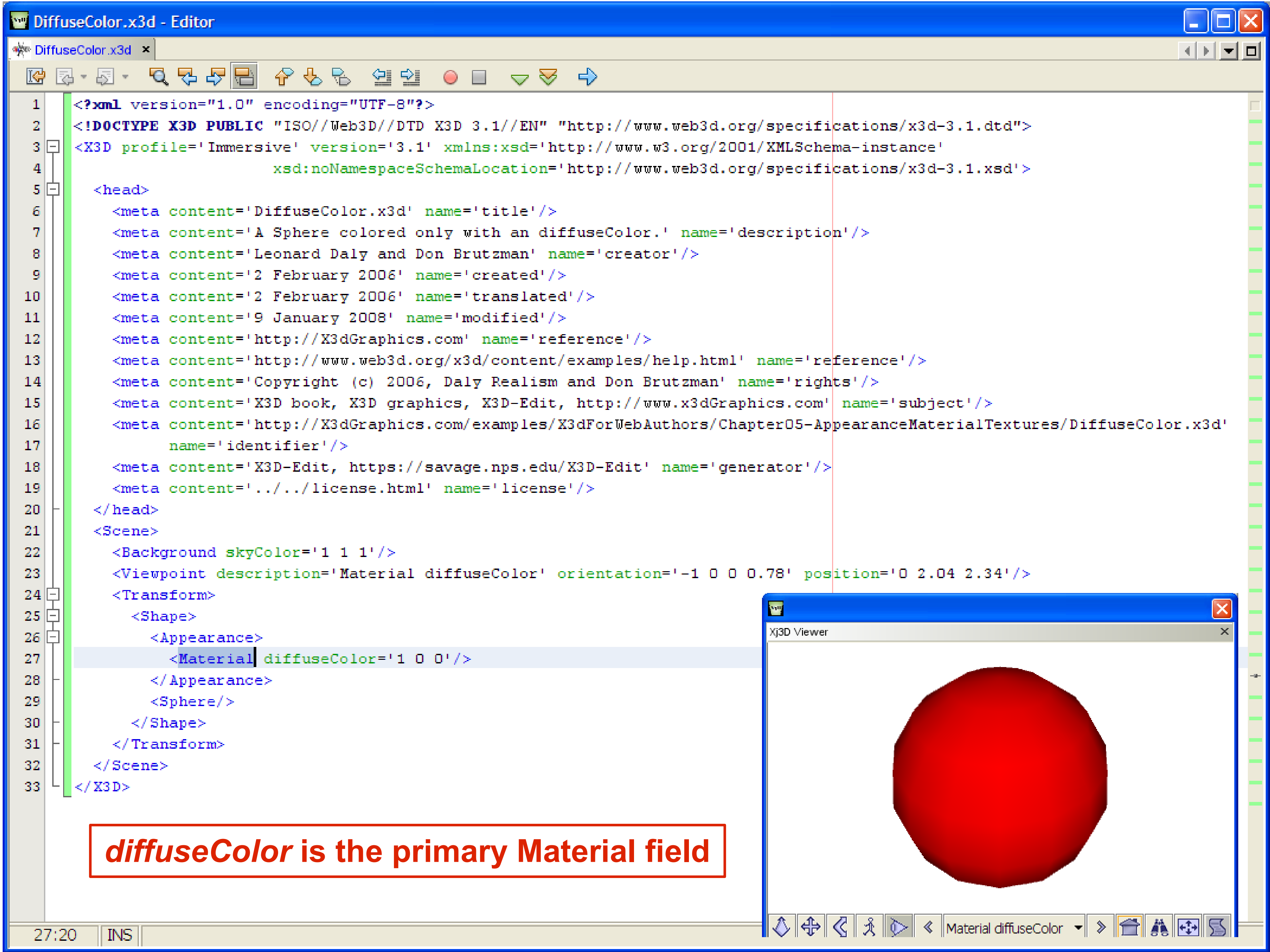
These field signatures are interpreted as follows:

- SFColor and SFFloat are field types
- [in,out] is accessType (i.e. "in, out" = inputOutput)
- default value is followed by [min,max] inclusive

Material fields

Color, transparency and shininess fields together make up Material properties. Examples follow.

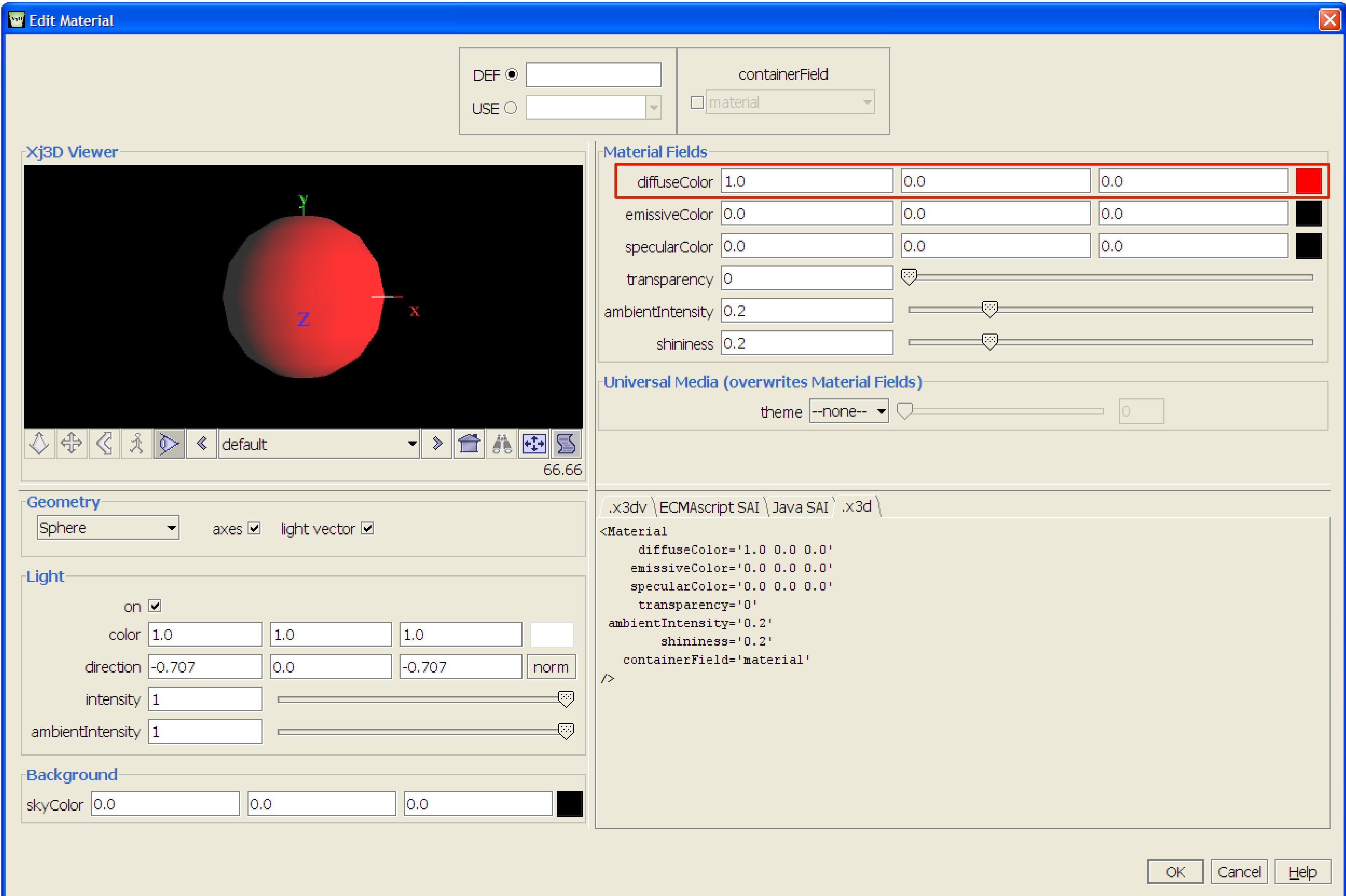
- *diffuseColor* reflects all X3D light sources, depending on viewing angles towards each light
- *ambientIntensity* is reflection multiplication factor
- *emissiveColor* is glowing component, normally off, independent of reflected light
- *specularColor* governs reflection highlights
- *shininess* controls specular intensity
- *transparency* is ability to see through an object: 1 is completely transparent, 0 is opaque



```
<?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='DiffuseColor.x3d' name='title'/>
    <meta content='A Sphere colored only with an diffuseColor.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='9 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 (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/Chapter05-AppearanceMaterialTextures/DiffuseColor.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='Material diffuseColor' orientation='-1 0 0 0.78' position='0 2.04 2.34'/>
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

***diffuseColor* is the primary Material field**

Material editor: *diffuseColor*



DEF ●

USE ○

containerField

□ material

Material Fields

diffuseColor	1.0	0.0	0.0	🟥
emissiveColor	0.0	0.0	0.0	⬛
specularColor	0.0	0.0	0.0	⬛
transparency	0			
ambientIntensity	0.2			
shininess	0.2			

Universal Media (overwrites Material Fields)

theme --none-- 0

Geometry

Sphere axes ☑ light vector ☑

Light

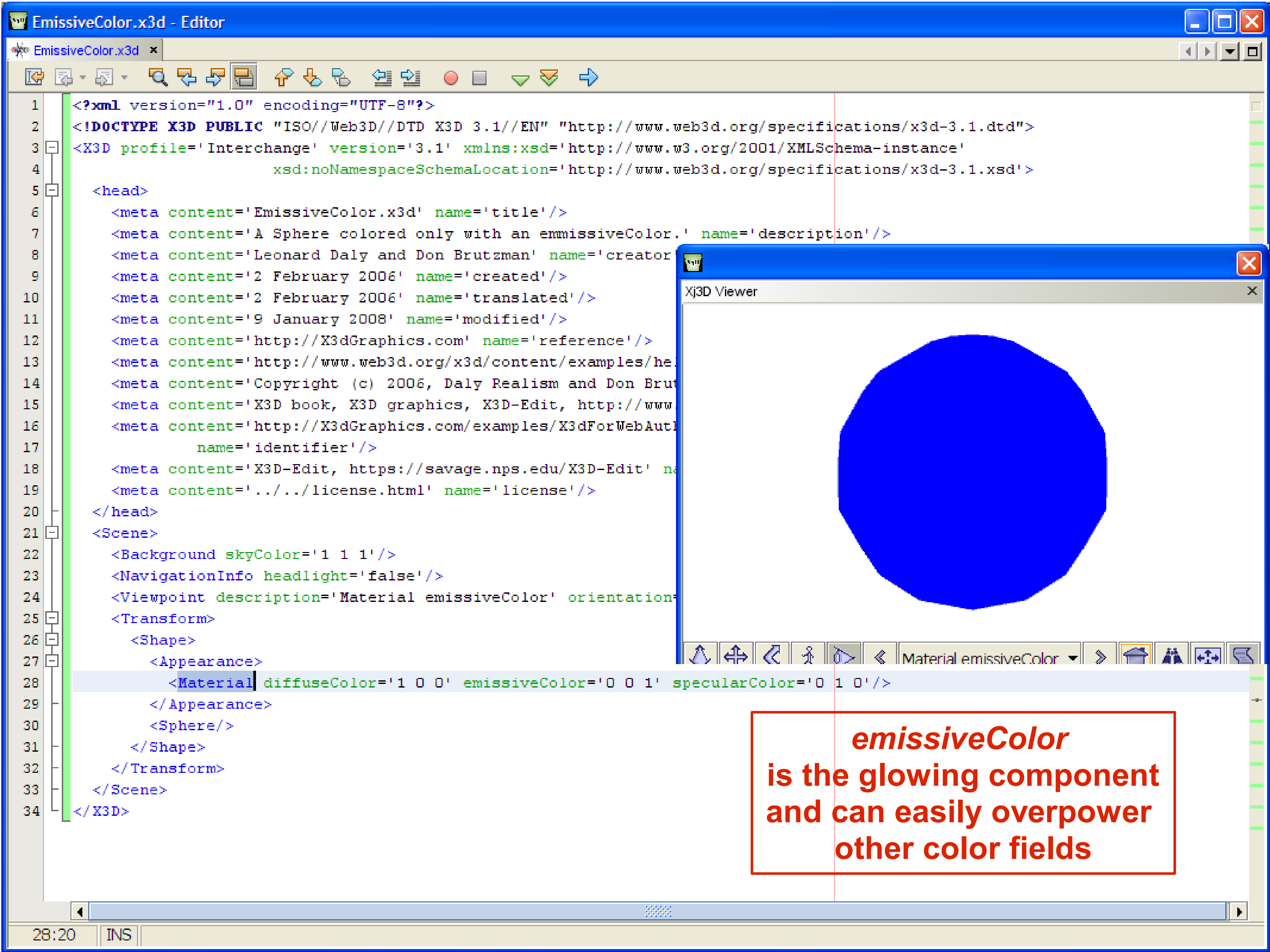
on ☑				
color	1.0	1.0	1.0	
direction	-0.707	0.0	-0.707	norm
intensity	1			
ambientIntensity	1			

Background

skyColor 0.0 0.0 0.0 ⬛

.x3dv | ECMAScript SAI | Java SAI | .x3d

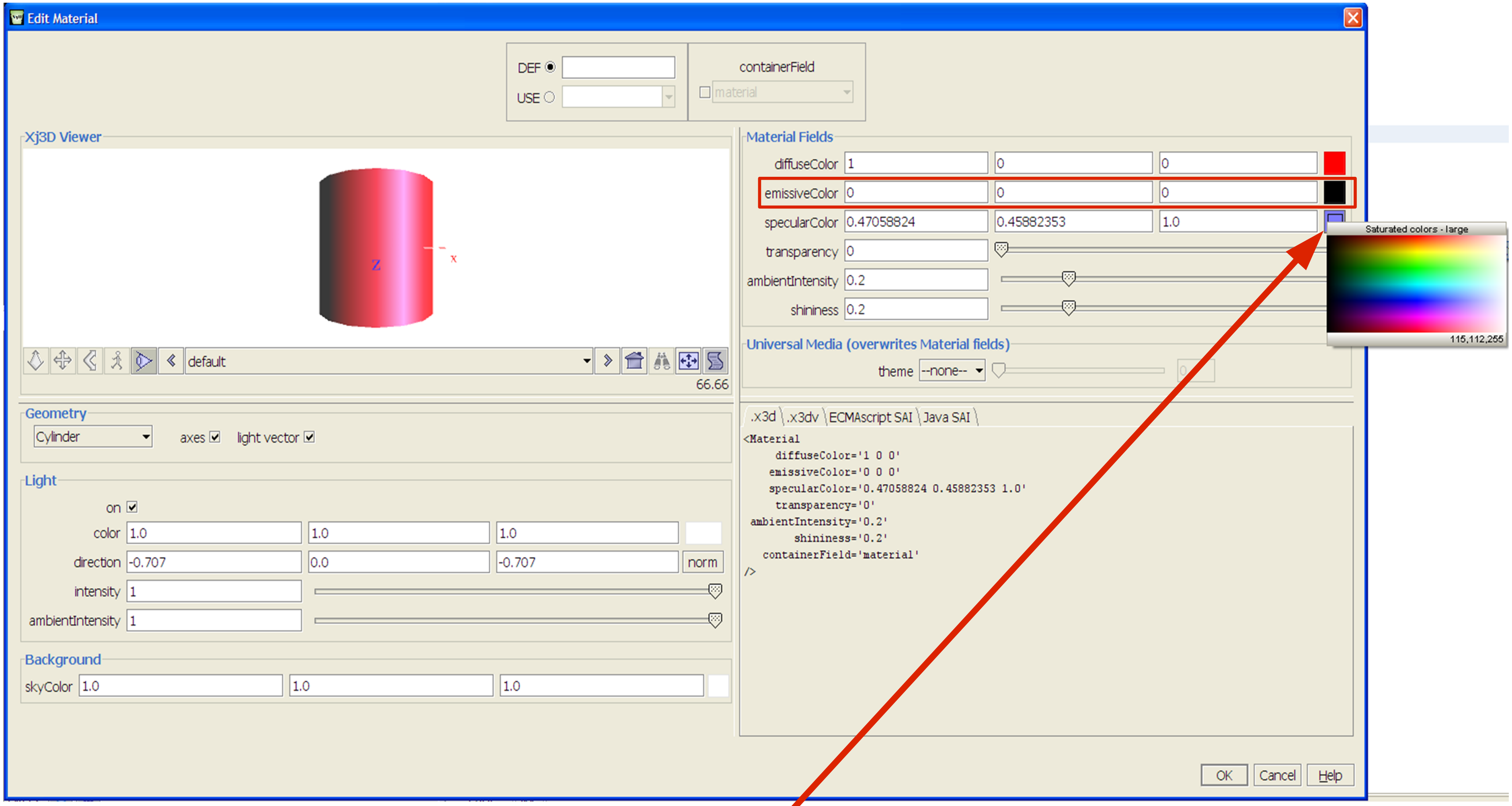
```
<Material
     diffuseColor='1.0 0.0 0.0'
    emissiveColor='0.0 0.0 0.0'
    specularColor='0.0 0.0 0.0'
     transparency='0'
 ambientIntensity='0.2'
        shininess='0.2'
   containerField='material'
/>
```



```
<?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='EmissiveColor.x3d' name='title'/>
    <meta content='A Sphere colored only with an emmissiveColor.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='9 January 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/he
    <meta content='Copyright (c) 2006, Daly Realism and Don Brut
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www
    <meta content='http://X3dGraphics.com/examples/X3dForWebAut
         name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' na
    <meta content='../../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo headlight='false'/>
    <Viewpoint description='Material emissiveColor' orientation=
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 1 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

emissiveColor
is the glowing component and can easily overpower other color fields

Material editor color selector



Edit Material

DEF ●

USE ○

containerField ☐ material

Xj3D Viewer

Geometry

Cylinder axes ☑ light vector ☑

Light

on ☑

color	1.0	1.0	1.0
direction	-0.707	0.0	-0.707
intensity	1		
ambientIntensity	1		

Background

skyColor	1.0	1.0	1.0

Material Fields

diffuseColor	1	0	0
emissiveColor	0	0	0
specularColor	0.47058824	0.45882353	1.0
transparency	0		
ambientIntensity	0.2		
shininess	0.2		

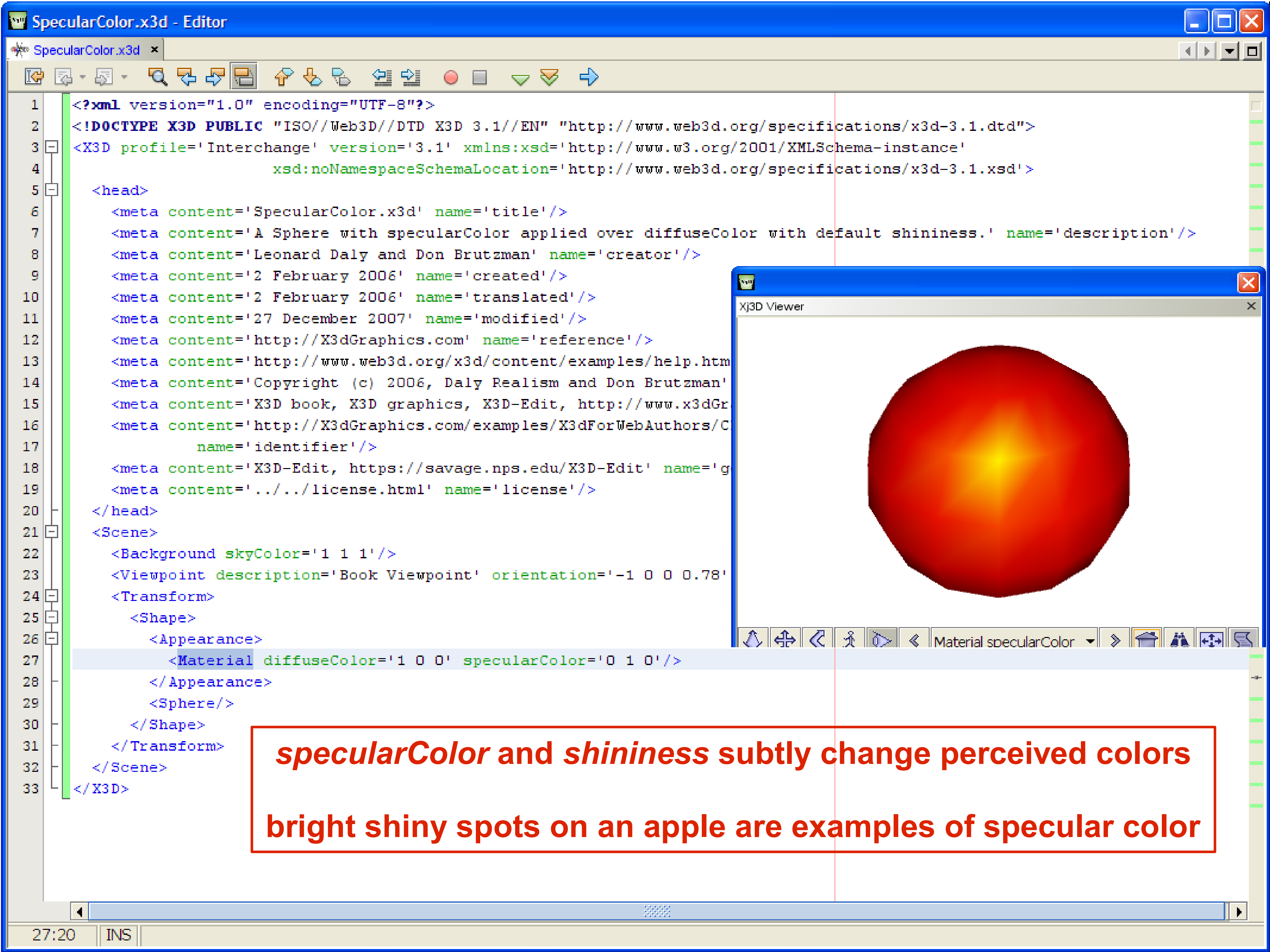
Universal Media (overwrites Material fields)

theme --none--

.x3d .x3dv ECMAscript SAI Java SAI

```
<Material
    diffuseColor='1 0 0'
   emissiveColor='0 0 0'
   specularColor='0.47058824 0.45882353 1.0'
    transparency='0'
ambientIntensity='0.2'
       shininess='0.2'
  containerField='material'
/>
```

Click colored box to select a color

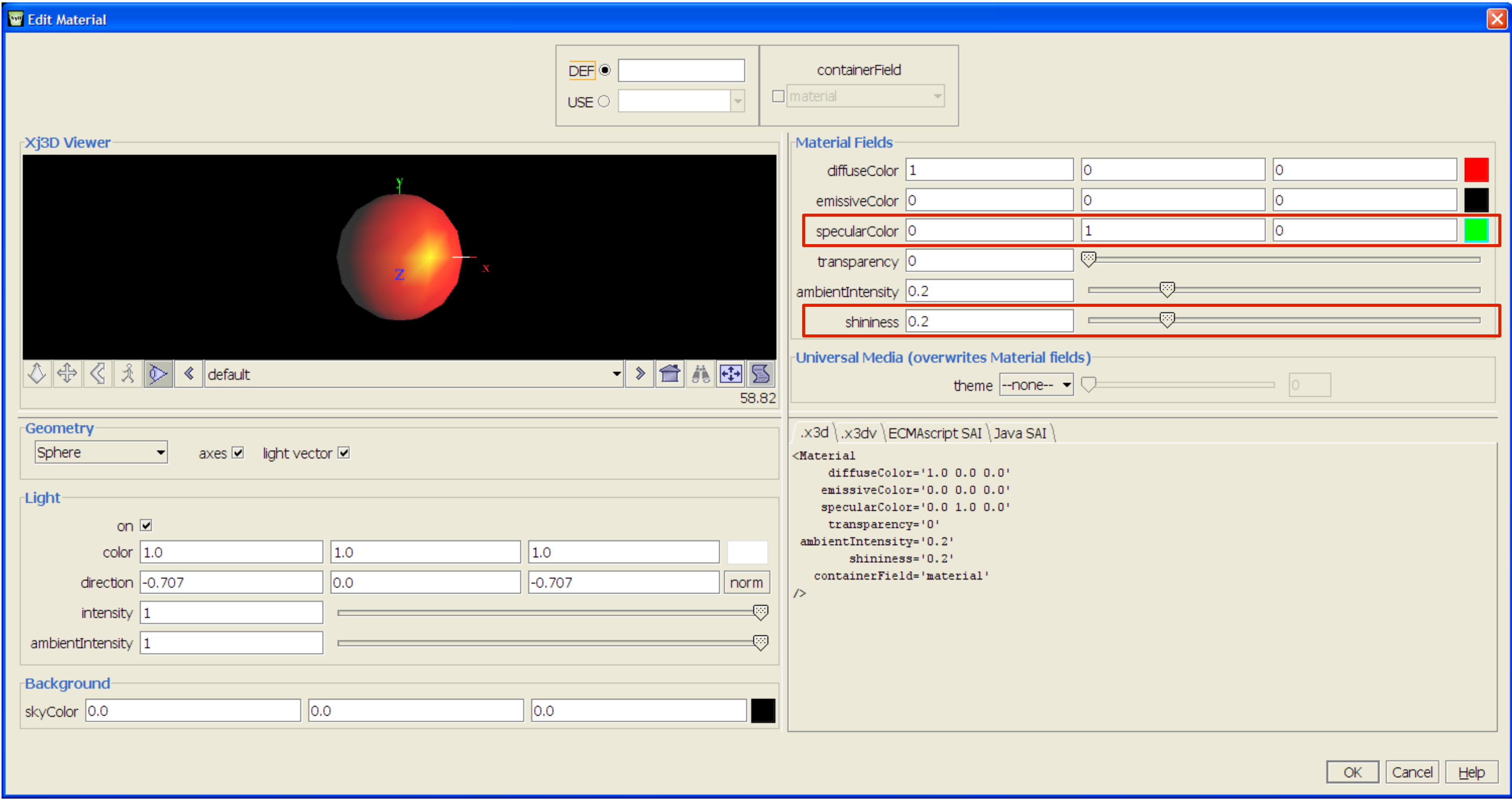


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='SpecularColor.x3d' name='title'/>
    <meta content='A Sphere with specularColor applied over diffuseColor with default shininess.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='27 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.htm
    <meta content='Copyright (c) 2006, Daly Realism and Don Brutzman'
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGr
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/C
         name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='g
    <meta content='../../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book Viewpoint' orientation='-1 0 0 0.78'
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0' specularColor='0 1 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

specularColor **and** ***shininess*** **subtly change perceived colors**

bright shiny spots on an apple are examples of specular color

Material editor *specularColor, shininess*



DEF ●

USE ○

containerField

☐ material

Xj3D Viewer

default

58.82

Material Fields

diffuseColor	1	0	0	🟥
emissiveColor	0	0	0	⬛
specularColor	0	1	0	🟩
transparency	0			
ambientIntensity	0.2			
shininess	0.2			

Universal Media (overwrites Material fields)

theme --none-- 0

Geometry

Sphere axes ☑ light vector ☑

Light

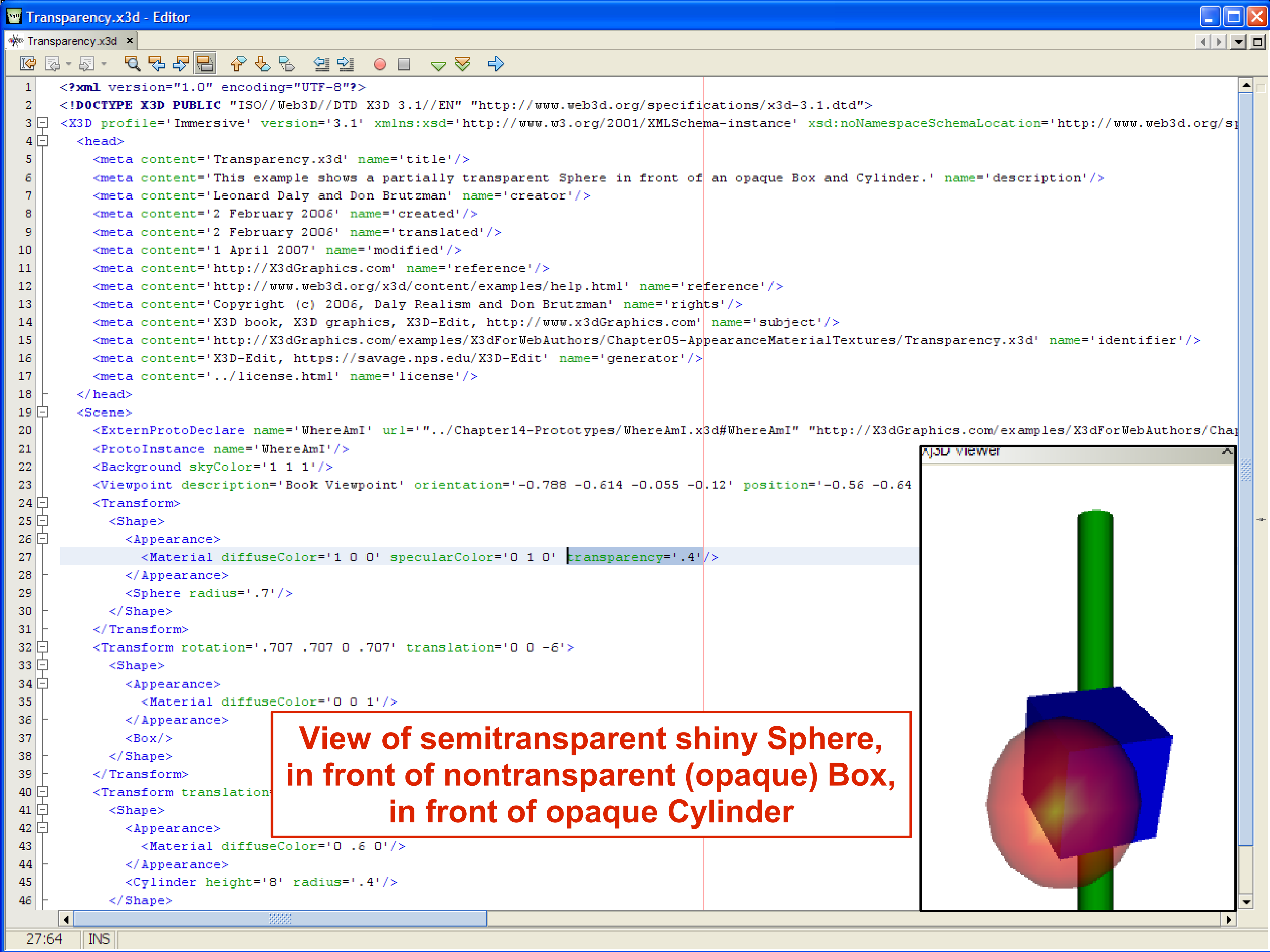
on	☑			
color	1.0	1.0	1.0	
direction	-0.707	0.0	-0.707	norm
intensity	1			
ambientIntensity	1			

Background

skyColor	0.0	0.0	0.0

.x3d .x3dv ECMAScript SAI Java SAI

```
<Material
      diffuseColor='1.0 0.0 0.0'
     emissiveColor='0.0 0.0 0.0'
     specularColor='0.0 1.0 0.0'
      transparency='0'
 ambientIntensity='0.2'
         shininess='0.2'
    containerField='material'
/>
```



```
<?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/s
  <head>
    <meta content='Transparency.x3d' name='title'/>
    <meta content='This example shows a partially transparent Sphere in front of an opaque Box and Cylinder.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='1 April 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/Chapter05-AppearanceMaterialTextures/Transparency.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <ExternProtoDeclare name='WhereAmI' url='"../Chapter14-Prototypes/WhereAmI.x3d#WhereAmI" "http://X3dGraphics.com/examples/X3dForWebAuthors/Chap
    <ProtoInstance name='WhereAmI'/>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book Viewpoint' orientation='-0.788 -0.614 -0.055 -0.12' position='-0.56 -0.64
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0' specularColor='0 1 0' transparency='.4'/>
        </Appearance>
        <Sphere radius='.7'/>
      </Shape>
    </Transform>
    <Transform rotation='.707 .707 0 .707' translation='0 0 -6'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 0 1'/>
        </Appearance>
        <Box/>
      </Shape>
    </Transform>
    <Transform translation
      <Shape>
        <Appearance>
          <Material diffuseColor='0 .6 0'/>
        </Appearance>
        <Cylinder height='8' radius='.4'/>
      </Shape>
```

View of semitransparent shiny Sphere, in front of nontransparent (opaque) Box, in front of opaque Cylinder

Universal Media materials library

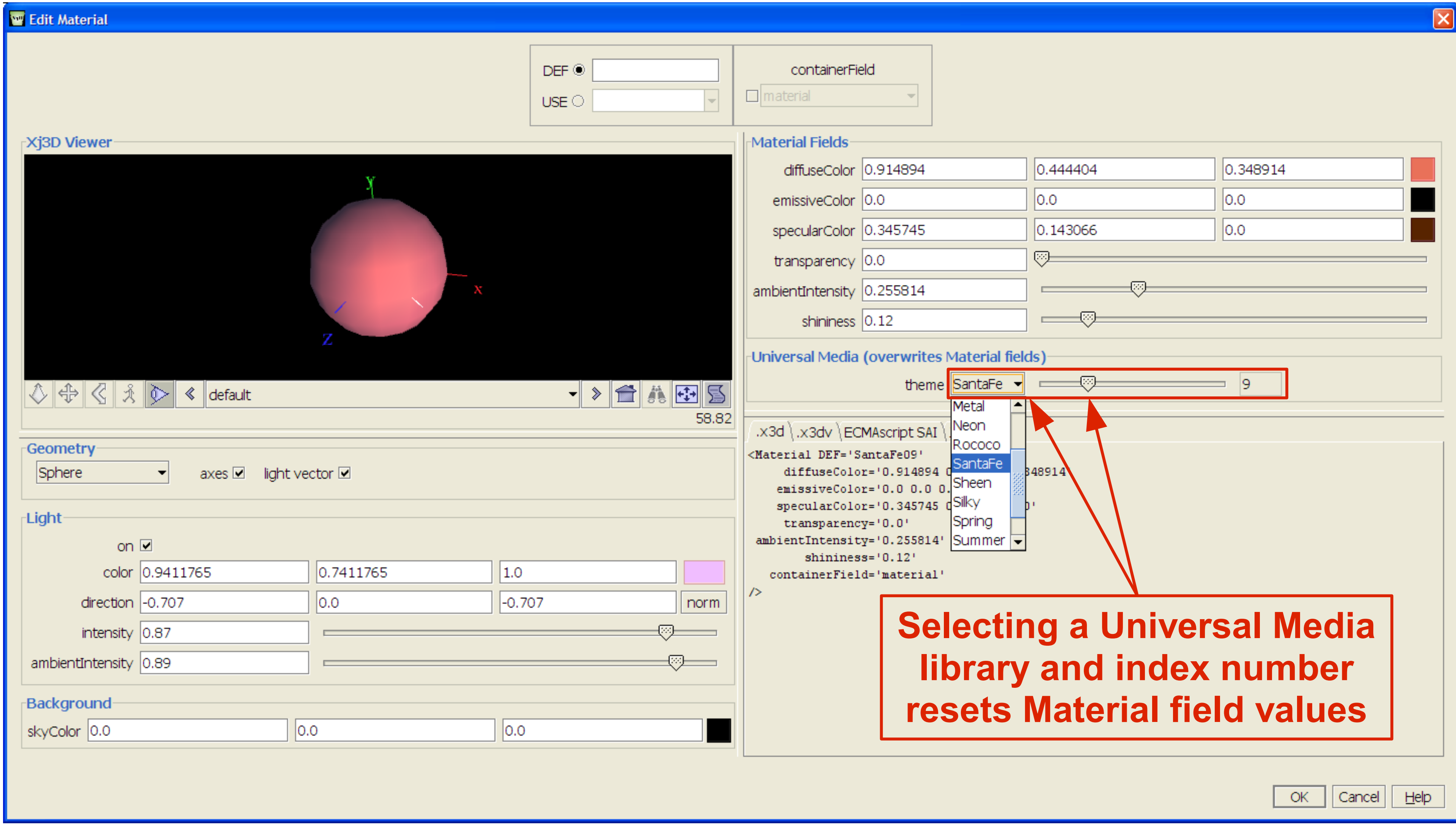
The Universal Media materials were originally created by SGI as part of OpenInventor in the 1990s as a convenience to authors

Each set of materials is grouped for visual compatibility and aesthetic appeal

Now converted and available for X3D use

- David Rousseau converted to VRML97
- Aaron Walsh created VRML Universal Media archive
- Don Brutzman translated into X3D as prototypes, cut/paste field values, also embedded in X3D-Edit
- http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials

Selecting a Universal Material value



Edit Material

DEF ●
USE ○

containerField
□ material

Xj3D Viewer

Geometry

Sphere   axes ☑   light vector ☑

Light

on ☑

color	0.9411765	0.7411765	1.0	🟪
direction	-0.707	0.0	-0.707	norm
intensity	0.87			
ambientIntensity	0.89			

Background

skyColor	0.0	0.0	0.0	⬛

Material Fields

diffuseColor	0.914894	0.444404	0.348914	🟥
emissiveColor	0.0	0.0	0.0	⬛
specularColor	0.345745	0.143066	0.0	🟫
transparency	0.0			
ambientIntensity	0.255814			
shininess	0.12			

Universal Media (overwrites Material fields)

theme SantaFe   9

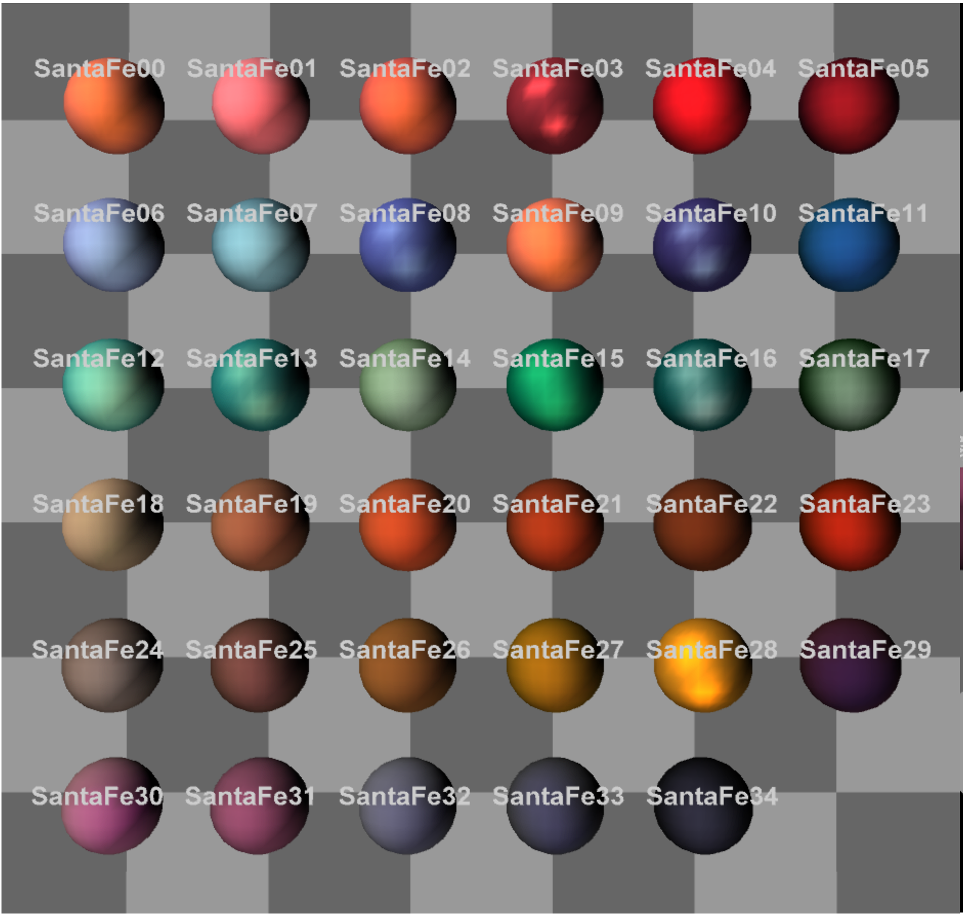
```
<Material DEF='SantaFe09'
     diffuseColor='0.914894 0          348914'
    emissiveColor='0.0 0.0 0.
    specularColor='0.345745 0          0'
     transparency='0.0'
 ambientIntensity='0.255814'
        shininess='0.12'
   containerField='material'
/>
```

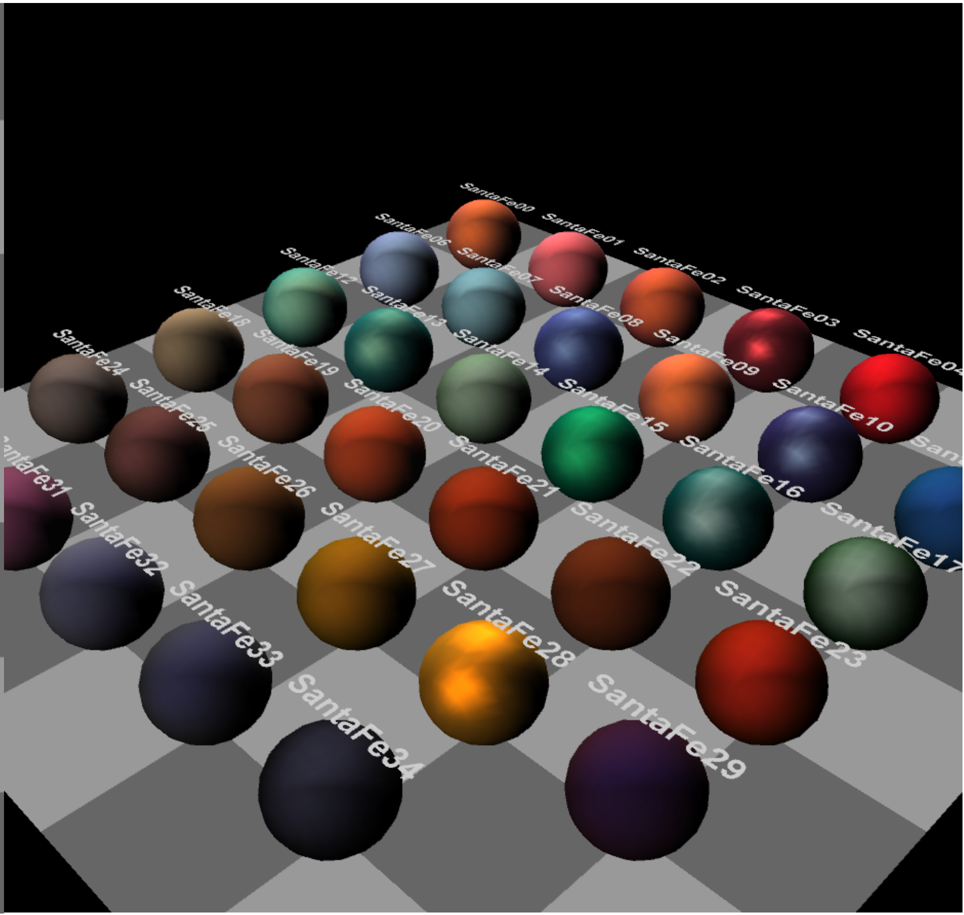
Selecting a Universal Media library and index number resets Material field values

Universal Media Material libraries include

ArtDeco, Autumn, Glass, Metal, Neon, Rococo, SantaFe, Sheen, Silky, Spring, Summer, Tropical, Winter

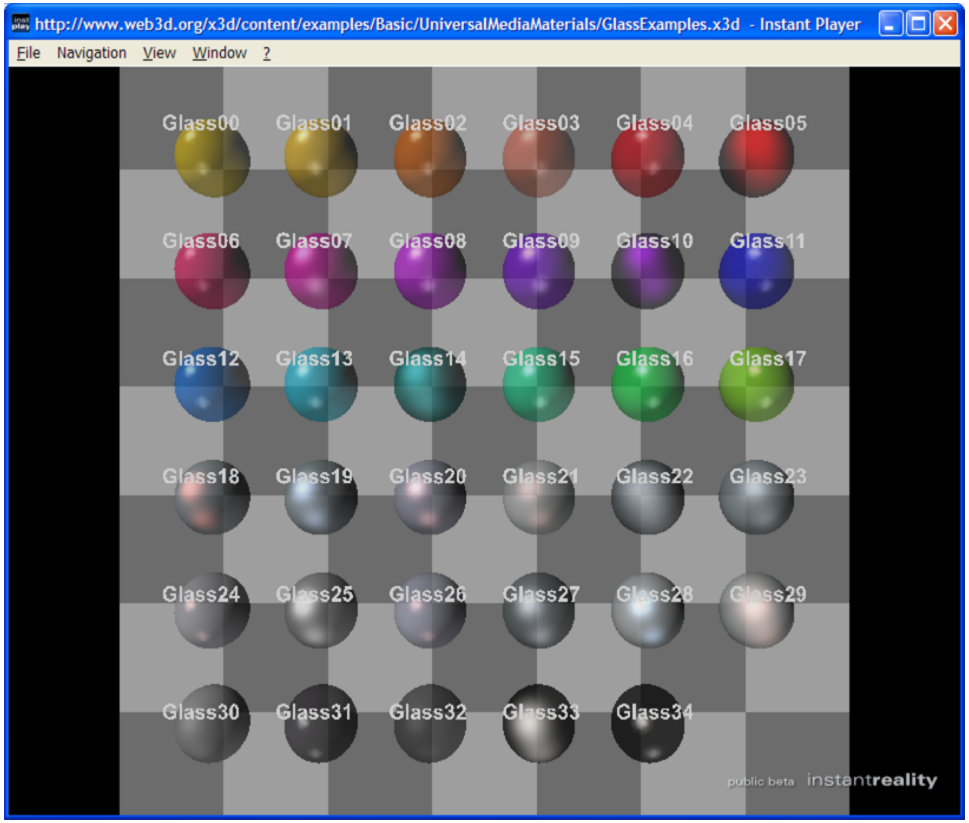
http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials





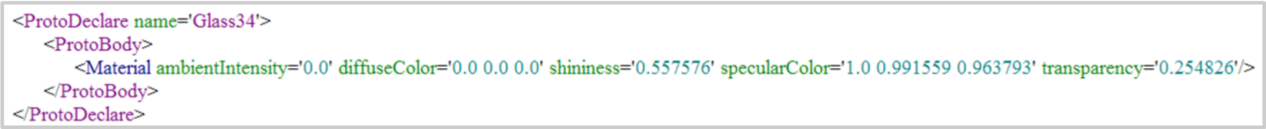
Universal Media Material library online

http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials

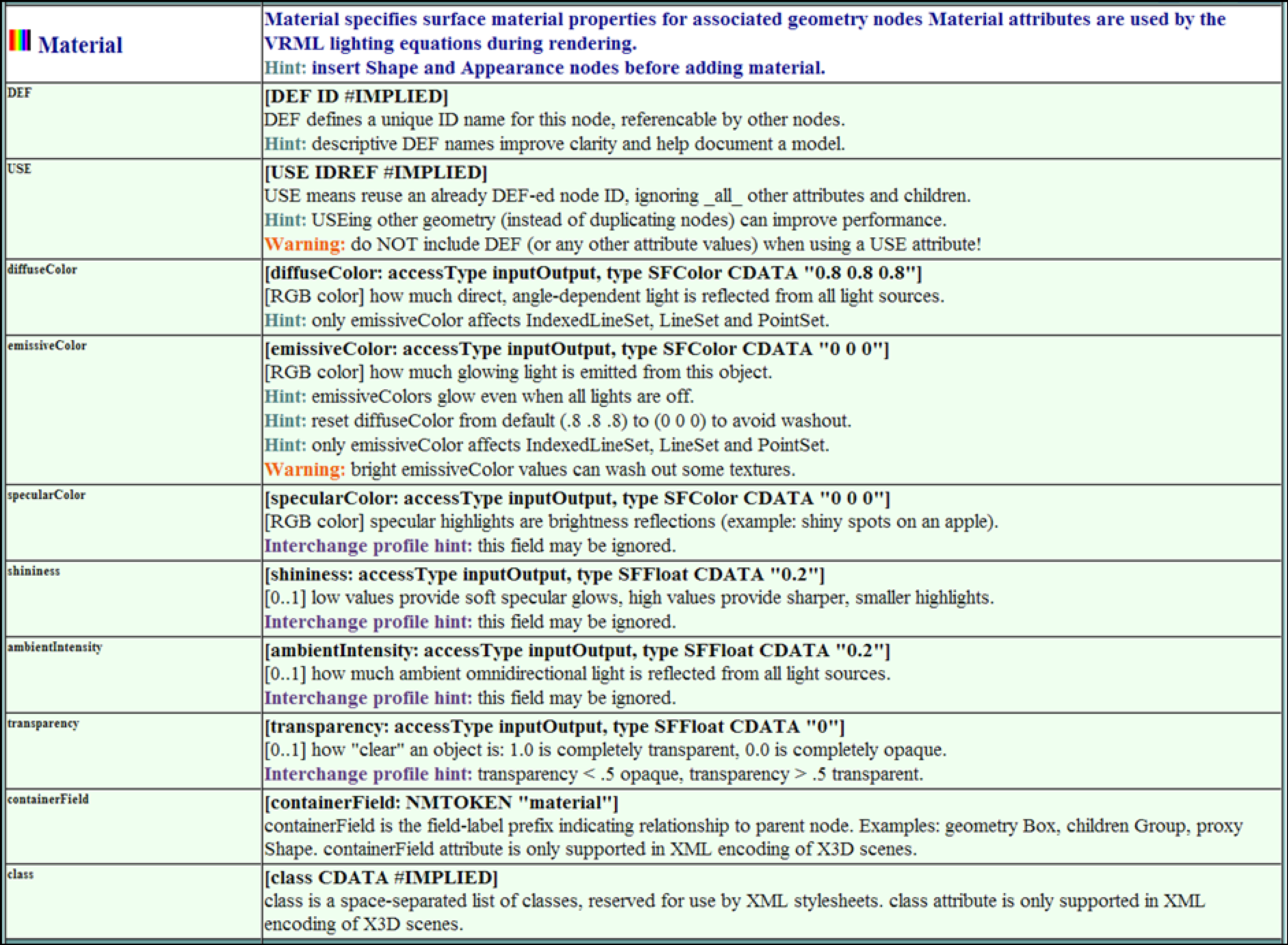


- Example scenes good for browsing
- Click sphere for closeup view
- Click text for X3D material source
- Click grid to restore full view

http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassExamples.x3d



```
<ProtoDeclare name='Glass34'>
    <ProtoBody>
        <Material ambientIntensity='0.0' diffuseColor='0.0 0.0 0.0' shininess='0.557576' specularColor='1.0 0.991559 0.963793' transparency='0.254826'/>
    </ProtoBody>
</ProtoDeclare>
```



Material	**Material specifies surface material properties for associated geometry nodes Material attributes are used by the VRML lighting equations during rendering.** **Hint: insert Shape and Appearance nodes before adding material.**
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!
diffuseColor	**[diffuseColor: accessType inputOutput, type SFColor CDATA "0.8 0.8 0.8"]** [RGB color] how much direct, angle-dependent light is reflected from all light sources. **Hint:** only emissiveColor affects IndexedLineSet, LineSet and PointSet.
emissiveColor	**[emissiveColor: accessType inputOutput, type SFColor CDATA "0 0 0"]** [RGB color] how much glowing light is emitted from this object. **Hint:** emissiveColors glow even when all lights are off. **Hint:** reset diffuseColor from default (.8 .8 .8) to (0 0 0) to avoid washout. **Hint:** only emissiveColor affects IndexedLineSet, LineSet and PointSet. **Warning:** bright emissiveColor values can wash out some textures.
specularColor	**[specularColor: accessType inputOutput, type SFColor CDATA "0 0 0"]** [RGB color] specular highlights are brightness reflections (example: shiny spots on an apple). **Interchange profile hint:** this field may be ignored.
shininess	**[shininess: accessType inputOutput, type SFFloat CDATA "0.2"]** [0..1] low values provide soft specular glows, high values provide sharper, smaller highlights. **Interchange profile hint:** this field may be ignored.
ambientIntensity	**[ambientIntensity: accessType inputOutput, type SFFloat CDATA "0.2"]** [0..1] how much ambient omnidirectional light is reflected from all light sources. **Interchange profile hint:** this field may be ignored.
transparency	**[transparency: accessType inputOutput, type SFFloat CDATA "0"]** [0..1] how "clear" an object is: 1.0 is completely transparent, 0.0 is completely opaque. **Interchange profile hint:** transparency < .5 opaque, transparency > .5 transparent.
containerField	**[containerField: NMTOKEN "material"]** 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.

TwoSidedMaterial node

TwoSidedMaterial fields are identical to Material, with the addition of the following new fields:

- *backAmbientIntensity, backShininess, backTransparency*
- *backDiffuseColor, backEmissiveColor, backSpecularColor*

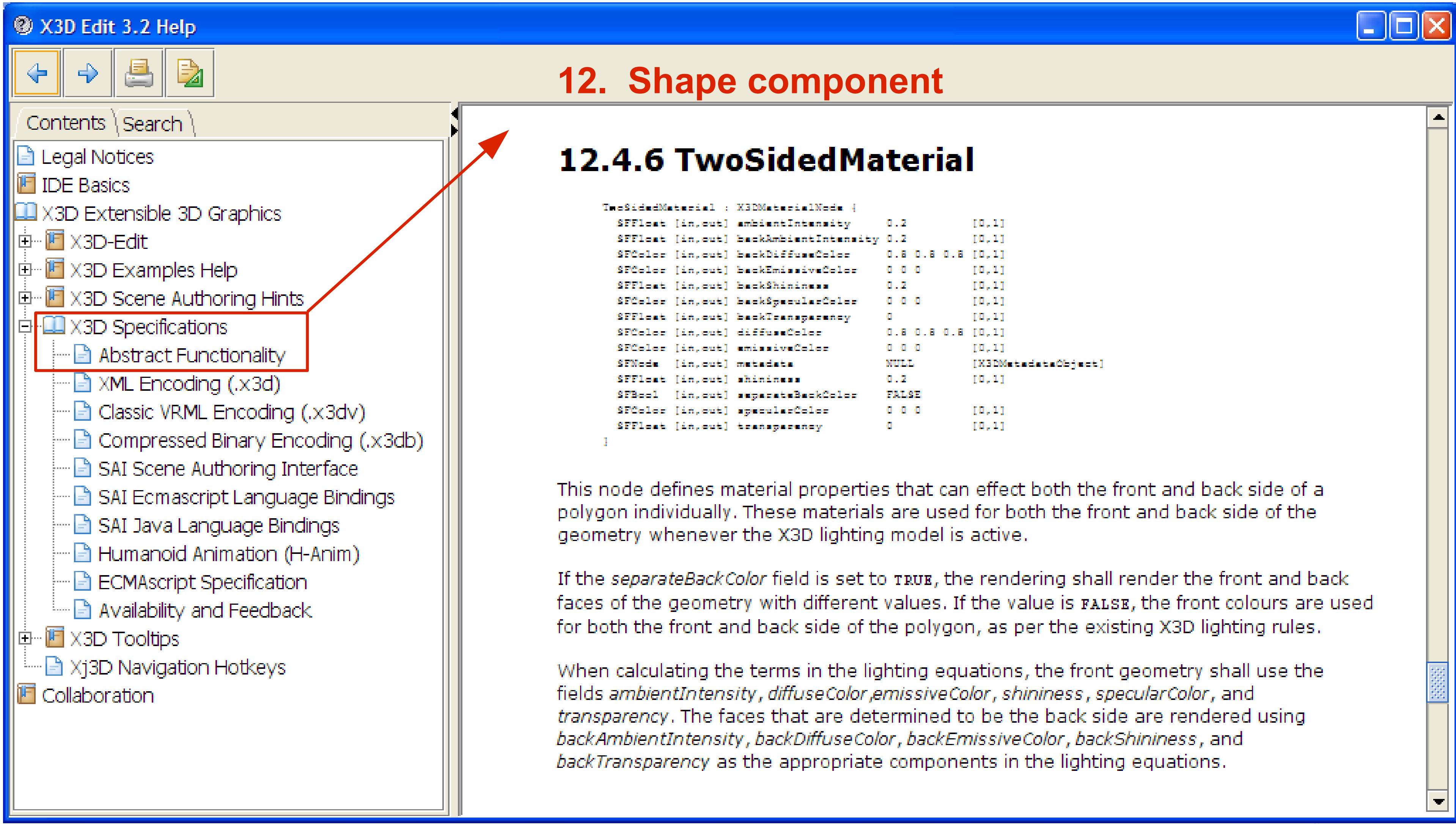
The 'back' fields determine how the 'backsides' of polygons are drawn

- Such as insides of primitive geometry
- Corresponding geometry must have *solid*='false'

Hint: *separateBackColor*='true' to enable back

Hint: include `<component name='Shape' level='4'/>`

TwoSidedMaterial specification help entry in X3D-Edit


X3D Edit 3.2 Help
Contents
Search
Legal Notices
IDE Basics
X3D Extensible 3D Graphics
X3D-Edit
X3D Examples Help
X3D Scene Authoring Hints
X3D Specifications
Abstract Functionality
XML Encoding (.x3d)
Classic VRML Encoding (.x3dv)
Compressed Binary Encoding (.x3db)
SAI Scene Authoring Interface
SAI Ecmascript Language Bindings
SAI Java Language Bindings
Humanoid Animation (H-Anim)
ECMAScript Specification
Availability and Feedback
X3D Tooltips
Xj3D Navigation Hotkeys
Collaboration

12. Shape component

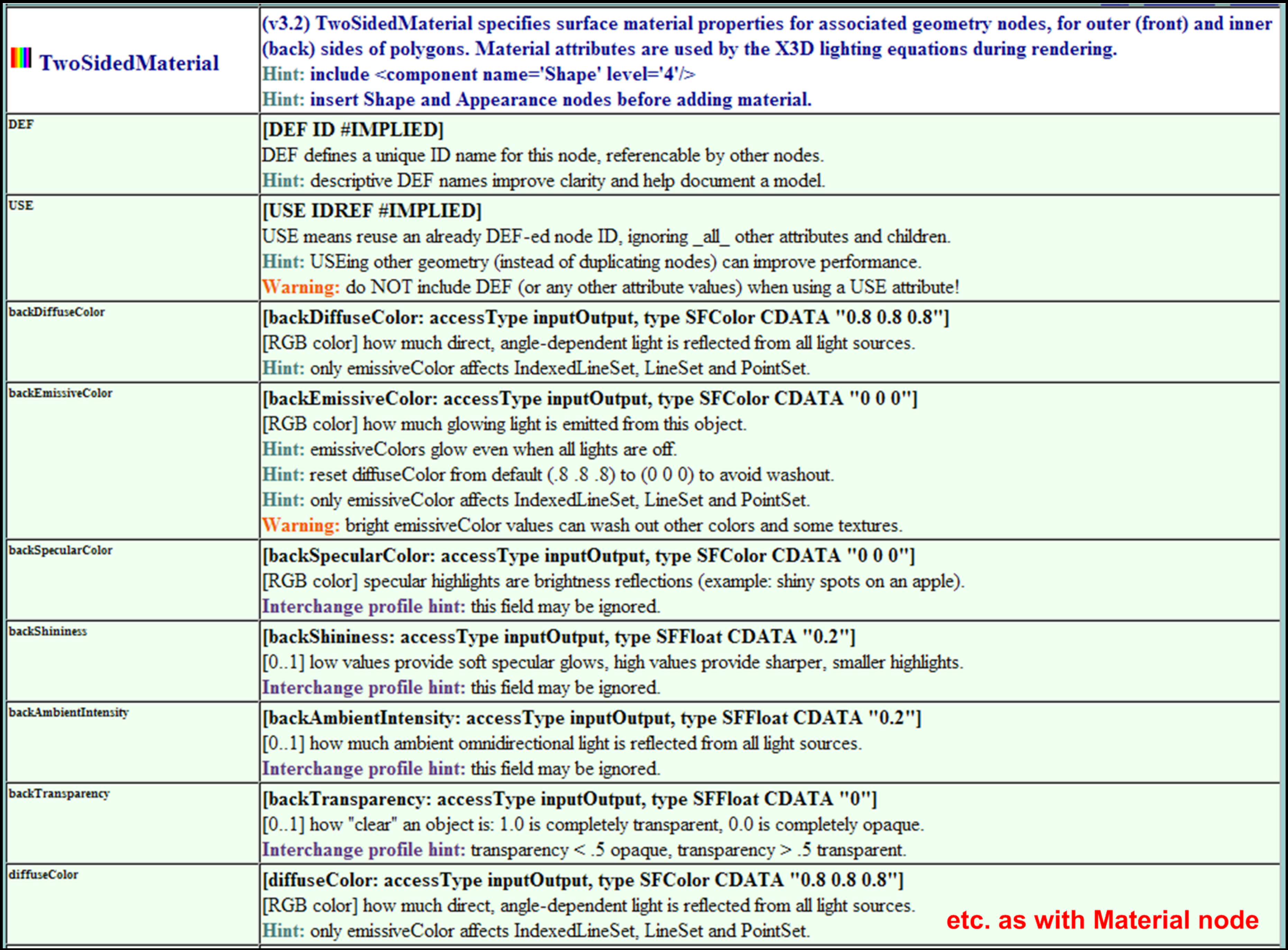
12.4.6 TwoSidedMaterial

```
TwoSidedMaterial : X3DMaterialNode {
  SFFloat [in,out] ambientIntensity     0.2         [0,1]
  SFFloat [in,out] backAmbientIntensity 0.2         [0,1]
  SFColor [in,out] backDiffuseColor     0.8 0.8 0.8 [0,1]
  SFColor [in,out] backEmissiveColor    0 0 0       [0,1]
  SFFloat [in,out] backShininess        0.2         [0,1]
  SFColor [in,out] backSpecularColor    0 0 0       [0,1]
  SFFloat [in,out] backTransparency     0           [0,1]
  SFColor [in,out] diffuseColor         0.8 0.8 0.8 [0,1]
  SFColor [in,out] emissiveColor        0 0 0       [0,1]
  SFNode  [in,out] metadata             NULL        [X3DMetadataObject]
  SFFloat [in,out] shininess            0.2         [0,1]
  SFBool  [in,out] separateBackColor    FALSE
  SFColor [in,out] specularColor        0 0 0       [0,1]
  SFFloat [in,out] transparency         0           [0,1]
}
```

This node defines material properties that can effect both the front and back side of a polygon individually. These materials are used for both the front and back side of the geometry whenever the X3D lighting model is active.

If the *separateBackColor* field is set to `TRUE`, the rendering shall render the front and back faces of the geometry with different values. If the value is `FALSE`, the front colours are used for both the front and back side of the polygon, as per the existing X3D lighting rules.

When calculating the terms in the lighting equations, the front geometry shall use the fields *ambientIntensity*, *diffuseColor*,*emissiveColor*, *shininess*, *specularColor*, and *transparency*. The faces that are determined to be the back side are rendered using *backAmbientIntensity*, *backDiffuseColor*, *backEmissiveColor*, *backShininess*, and *backTransparency* as the appropriate components in the lighting equations.



TwoSidedMaterial	(v3.2) TwoSidedMaterial specifies surface material properties for associated geometry nodes, for outer (front) and inner (back) sides of polygons. Material attributes are used by the X3D lighting equations during rendering. Hint: include <component name='Shape' level='4'/> Hint: insert Shape and Appearance nodes before adding material.
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!
backDiffuseColor	[backDiffuseColor: accessType inputOutput, type SFColor CDATA "0.8 0.8 0.8"] [RGB color] how much direct, angle-dependent light is reflected from all light sources. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet.
backEmissiveColor	[backEmissiveColor: accessType inputOutput, type SFColor CDATA "0 0 0"] [RGB color] how much glowing light is emitted from this object. Hint: emissiveColors glow even when all lights are off. Hint: reset diffuseColor from default (.8 .8 .8) to (0 0 0) to avoid washout. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet. Warning: bright emissiveColor values can wash out other colors and some textures.
backSpecularColor	[backSpecularColor: accessType inputOutput, type SFColor CDATA "0 0 0"] [RGB color] specular highlights are brightness reflections (example: shiny spots on an apple). Interchange profile hint: this field may be ignored.
backShininess	[backShininess: accessType inputOutput, type SFFloat CDATA "0.2"] [0..1] low values provide soft specular glows, high values provide sharper, smaller highlights. Interchange profile hint: this field may be ignored.
backAmbientIntensity	[backAmbientIntensity: accessType inputOutput, type SFFloat CDATA "0.2"] [0..1] how much ambient omnidirectional light is reflected from all light sources. Interchange profile hint: this field may be ignored.
backTransparency	[backTransparency: accessType inputOutput, type SFFloat CDATA "0"] [0..1] how "clear" an object is: 1.0 is completely transparent, 0.0 is completely opaque. Interchange profile hint: transparency < .5 opaque, transparency > .5 transparent.
diffuseColor	[diffuseColor: accessType inputOutput, type SFColor CDATA "0.8 0.8 0.8"] [RGB color] how much direct, angle-dependent light is reflected from all light sources. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet.

etc. as with Material node

FillProperties node

FillProperties specifies additional characteristics that can be applied to the material shading of geometry nodes

- Adds to basic effects of peer Material and texture

FillProperties is a new X3D node not in VRML97

- If backwards compatibility needed and FillProperties effects are critical, consider an additional secondary technique to also backup this functionality

Hint: include `<component name='Shape' level='3'/>`

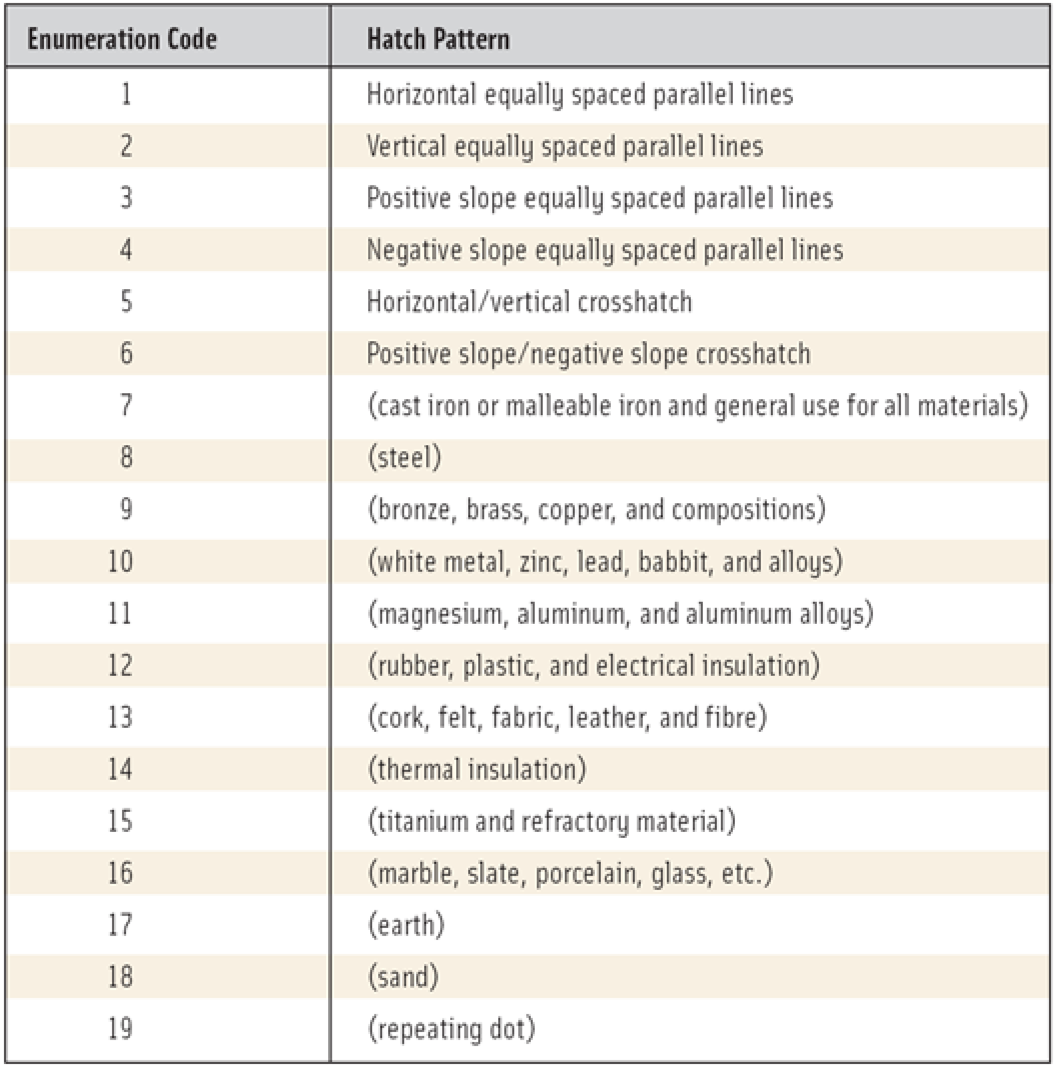
Note: hatch effects are not affected by lighting

FillProperties fields

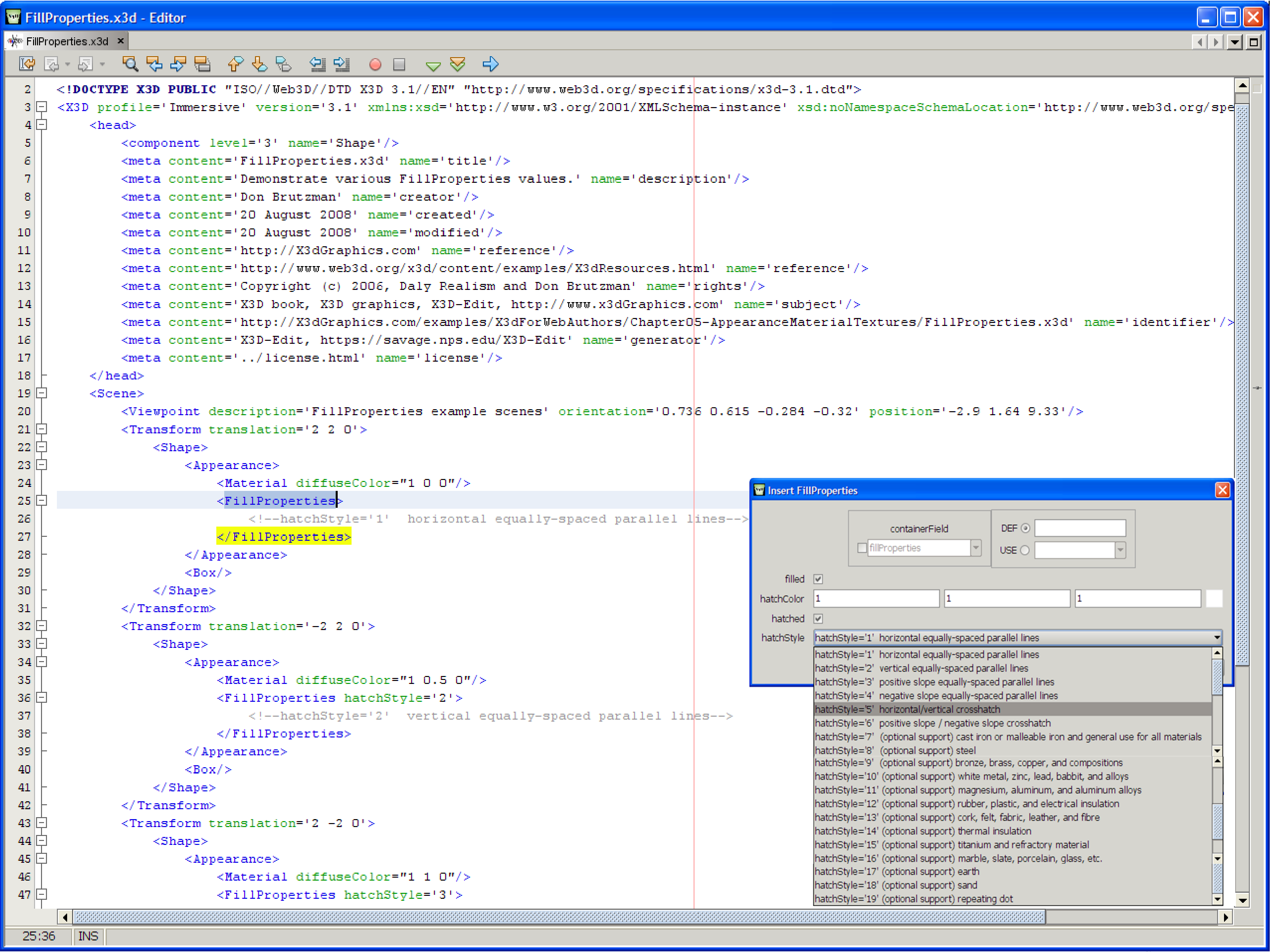
- *filled* is a boolean (true or false) field to indicate whether the material properties are filled in. Setting *filled*='false' can be useful to highlight hatching effects.
- *hatched* is another SFBool single-field boolean that turns hatching effects on or off. Hatching can be a helpful user-interaction technique to indicate selection or objects of interest.
- *hatchColor* is the color applied to hatching effects over the material surface. Be sure to use a color that distinguishes hatching from *diffuseColor*.
- *hatchStyle* codes follow on the next slide

FillProperties hatchStyle codes

(parentheses indicate optional support)



Enumeration Code	Hatch Pattern
1	Horizontal equally spaced parallel lines
2	Vertical equally spaced parallel lines
3	Positive slope equally spaced parallel lines
4	Negative slope equally spaced parallel lines
5	Horizontal/vertical crosshatch
6	Positive slope/negative slope crosshatch
7	(cast iron or malleable iron and general use for all materials)
8	(steel)
9	(bronze, brass, copper, and compositions)
10	(white metal, zinc, lead, babbit, and alloys)
11	(magnesium, aluminum, and aluminum alloys)
12	(rubber, plastic, and electrical insulation)
13	(cork, felt, fabric, leather, and fibre)
14	(thermal insulation)
15	(titanium and refractory material)
16	(marble, slate, porcelain, glass, etc.)
17	(earth)
18	(sand)
19	(repeating dot)



FillProperties.x3d - Editor

FillProperties.x3d

```
<!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/spe
    <head>
        <component level='3' name='Shape'/>
        <meta content='FillProperties.x3d' name='title'/>
        <meta content='Demonstrate various FillProperties values.' name='description'/>
        <meta content='Don Brutzman' name='creator'/>
        <meta content='20 August 2008' name='created'/>
        <meta content='20 August 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 (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/Chapter05-AppearanceMaterialTextures/FillProperties.x3d' name='identifier'/>
        <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
        <meta content='../license.html' name='license'/>
    </head>
    <Scene>
        <Viewpoint description='FillProperties example scenes' orientation='0.736 0.615 -0.284 -0.32' position='-2.9 1.64 9.33'/>
        <Transform translation='2 2 0'>
            <Shape>
                <Appearance>
                    <Material diffuseColor="1 0 0"/>
                    <FillProperties>
                        <!--hatchStyle='1'  horizontal equally-spaced parallel lines-->
                    </FillProperties>
                </Appearance>
                <Box/>
            </Shape>
        </Transform>
        <Transform translation='-2 2 0'>
            <Shape>
                <Appearance>
                    <Material diffuseColor="1 0.5 0"/>
                    <FillProperties hatchStyle='2'>
                        <!--hatchStyle='2'  vertical equally-spaced parallel lines-->
                    </FillProperties>
                </Appearance>
                <Box/>
            </Shape>
        </Transform>
        <Transform translation='2 -2 0'>
            <Shape>
                <Appearance>
                    <Material diffuseColor="1 1 0"/>
                    <FillProperties hatchStyle='3'>
```

Insert FillProperties

containerField □ fillProperties

DEF ●

USE ○

filled ☑

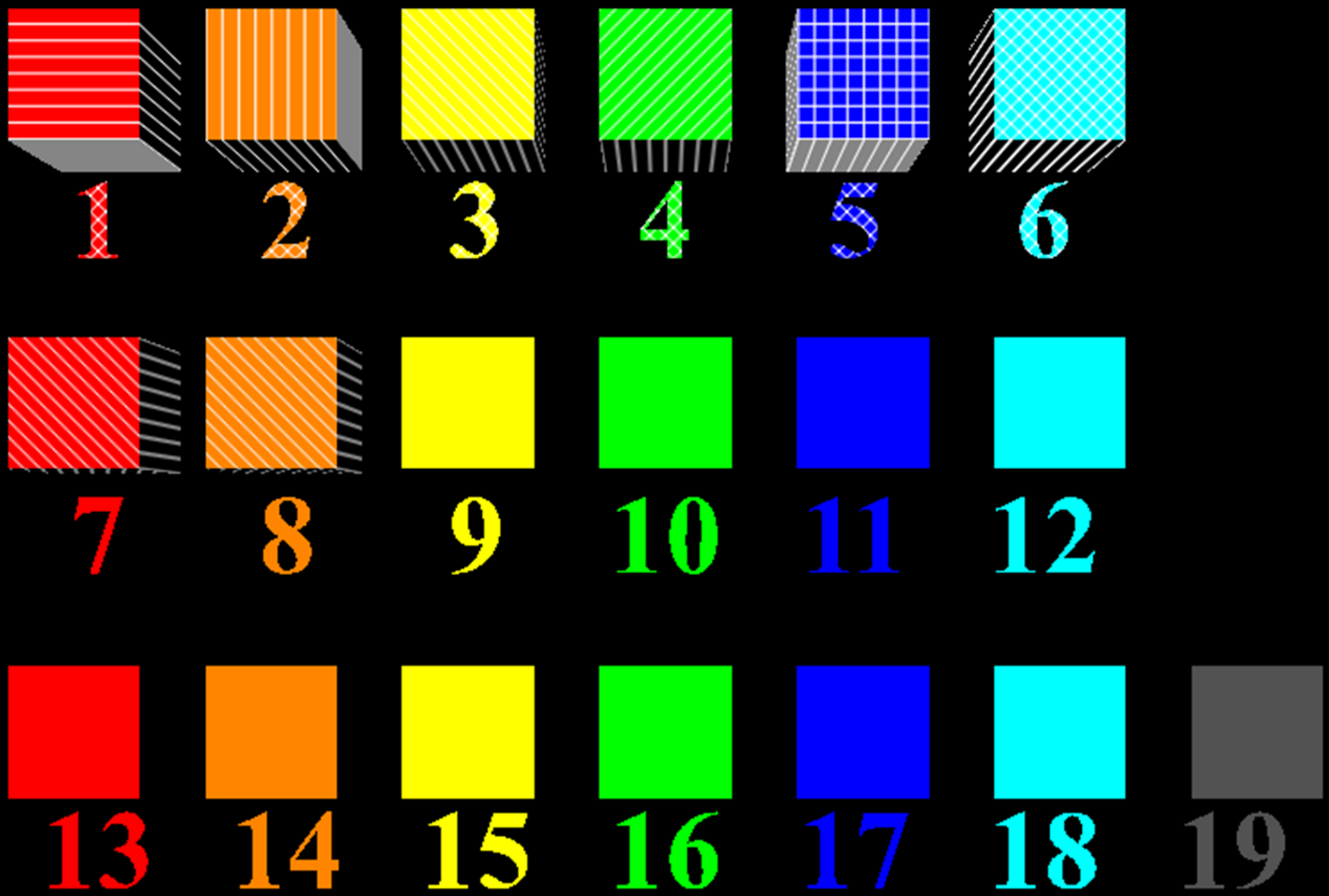
hatchColor 1 1 1

hatched ☑

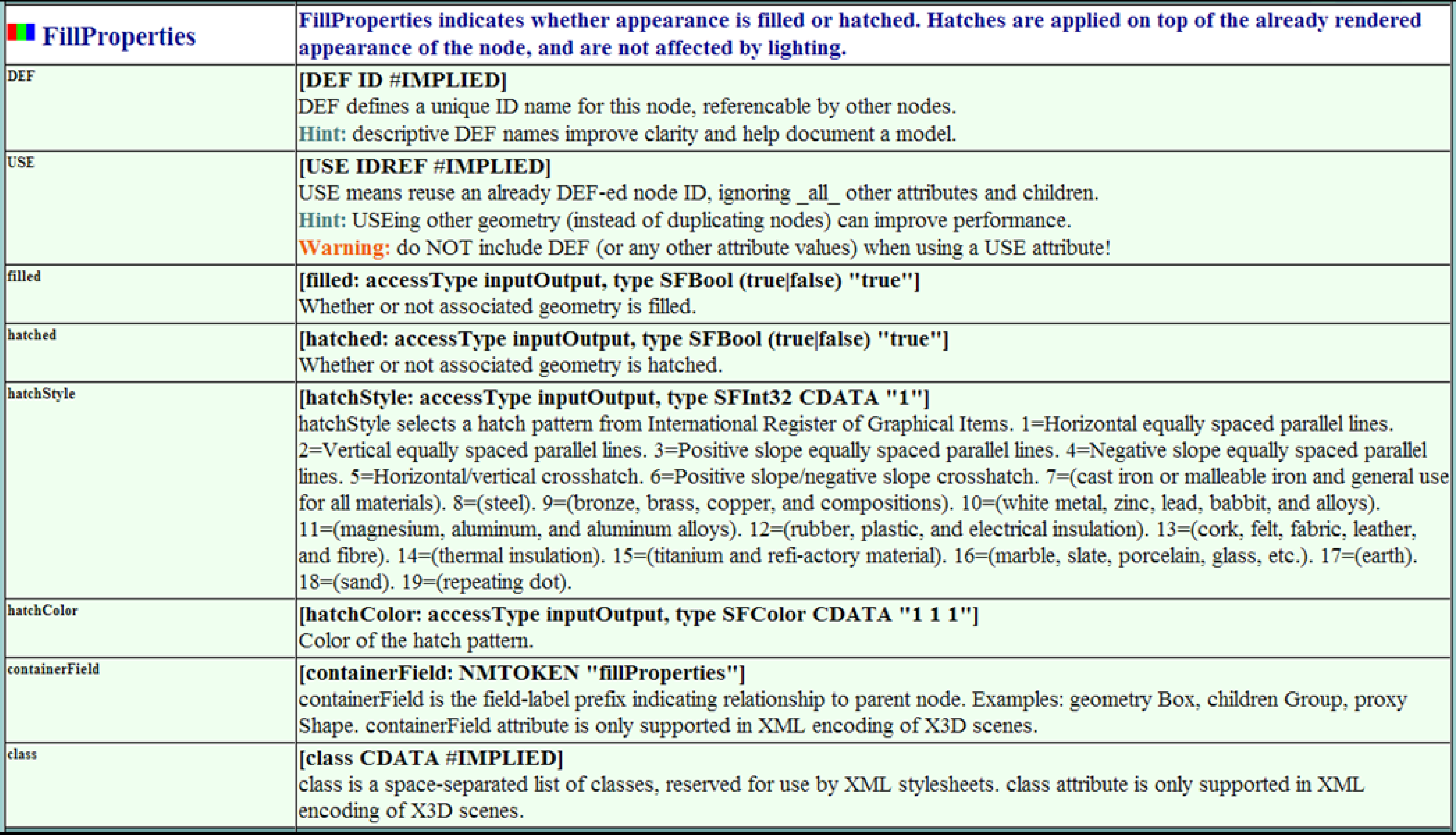
hatchStyle hatchStyle='1' horizontal equally-spaced parallel lines

- hatchStyle='1' horizontal equally-spaced parallel lines
- hatchStyle='2' vertical equally-spaced parallel lines
- hatchStyle='3' positive slope equally-spaced parallel lines
- hatchStyle='4' negative slope equally-spaced parallel lines
- hatchStyle='5' horizontal/vertical crosshatch
- hatchStyle='6' positive slope / negative slope crosshatch
- hatchStyle='7' (optional support) cast iron or malleable iron and general use for all materials
- hatchStyle='8' (optional support) steel
- hatchStyle='9' (optional support) bronze, brass, copper, and compositions
- hatchStyle='10' (optional support) white metal, zinc, lead, babbit, and alloys
- hatchStyle='11' (optional support) magnesium, aluminum, and aluminum alloys
- hatchStyle='12' (optional support) rubber, plastic, and electrical insulation
- hatchStyle='13' (optional support) cork, felt, fabric, leather, and fibre
- hatchStyle='14' (optional support) thermal insulation
- hatchStyle='15' (optional support) titanium and refractory material
- hatchStyle='16' (optional support) marble, slate, porcelain, glass, etc.
- hatchStyle='17' (optional support) earth
- hatchStyle='18' (optional support) sand
- hatchStyle='19' (optional support) repeating dot

25:36 INS



Octaga™



FillProperties	**FillProperties indicates whether appearance is filled or hatched. Hatches are applied on top of the already rendered appearance of the node, and are not affected by lighting.**
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!
filled	**[filled: accessType inputOutput, type SFBool (true|false) "true"]** Whether or not associated geometry is filled.
hatched	**[hatched: accessType inputOutput, type SFBool (true|false) "true"]** Whether or not associated geometry is hatched.
hatchStyle	**[hatchStyle: accessType inputOutput, type SFInt32 CDATA "1"]** hatchStyle selects a hatch pattern from International Register of Graphical Items. 1=Horizontal equally spaced parallel lines. 2=Vertical equally spaced parallel lines. 3=Positive slope equally spaced parallel lines. 4=Negative slope equally spaced parallel lines. 5=Horizontal/vertical crosshatch. 6=Positive slope/negative slope crosshatch. 7=(cast iron or malleable iron and general use for all materials). 8=(steel). 9=(bronze, brass, copper, and compositions). 10=(white metal, zinc, lead, babbit, and alloys). 11=(magnesium, aluminum, and aluminum alloys). 12=(rubber, plastic, and electrical insulation). 13=(cork, felt, fabric, leather, and fibre). 14=(thermal insulation). 15=(titanium and refi-actory material). 16=(marble, slate, porcelain, glass, etc.). 17=(earth). 18=(sand). 19=(repeating dot).
hatchColor	**[hatchColor: accessType inputOutput, type SFColor CDATA "1 1 1"]** Color of the hatch pattern.
containerField	**[containerField: NMTOKEN "fillProperties"]** 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.

LineProperties node

LineProperties specifies additional characteristics that can be applied to the material shading of geometry nodes

- Adds to basic effects of peer Material and texture
- Also applies to geometry edges, not just lines

LineProperties is a new X3D node not in VRML97

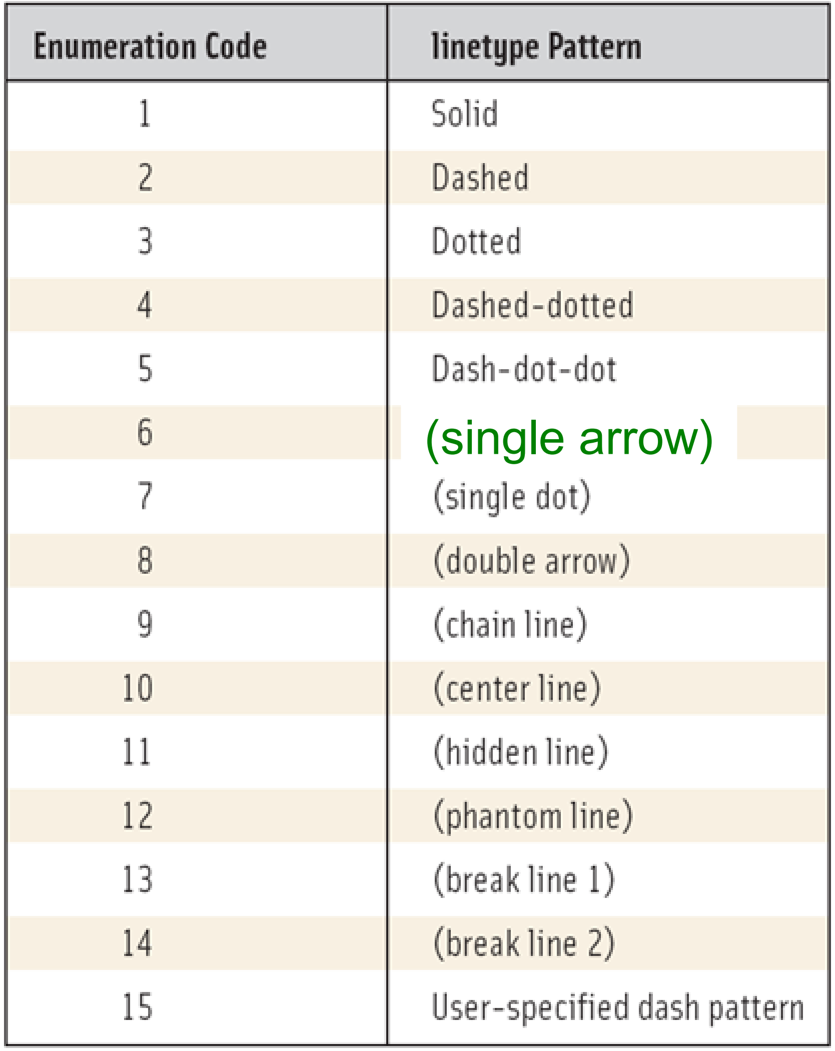
- If backwards compatibility needed and FillProperties effects are critical, consider an additional secondary technique to also backup this functionality
- Hint: include `<component name='Shape' level='2'/>`

LineProperties fields

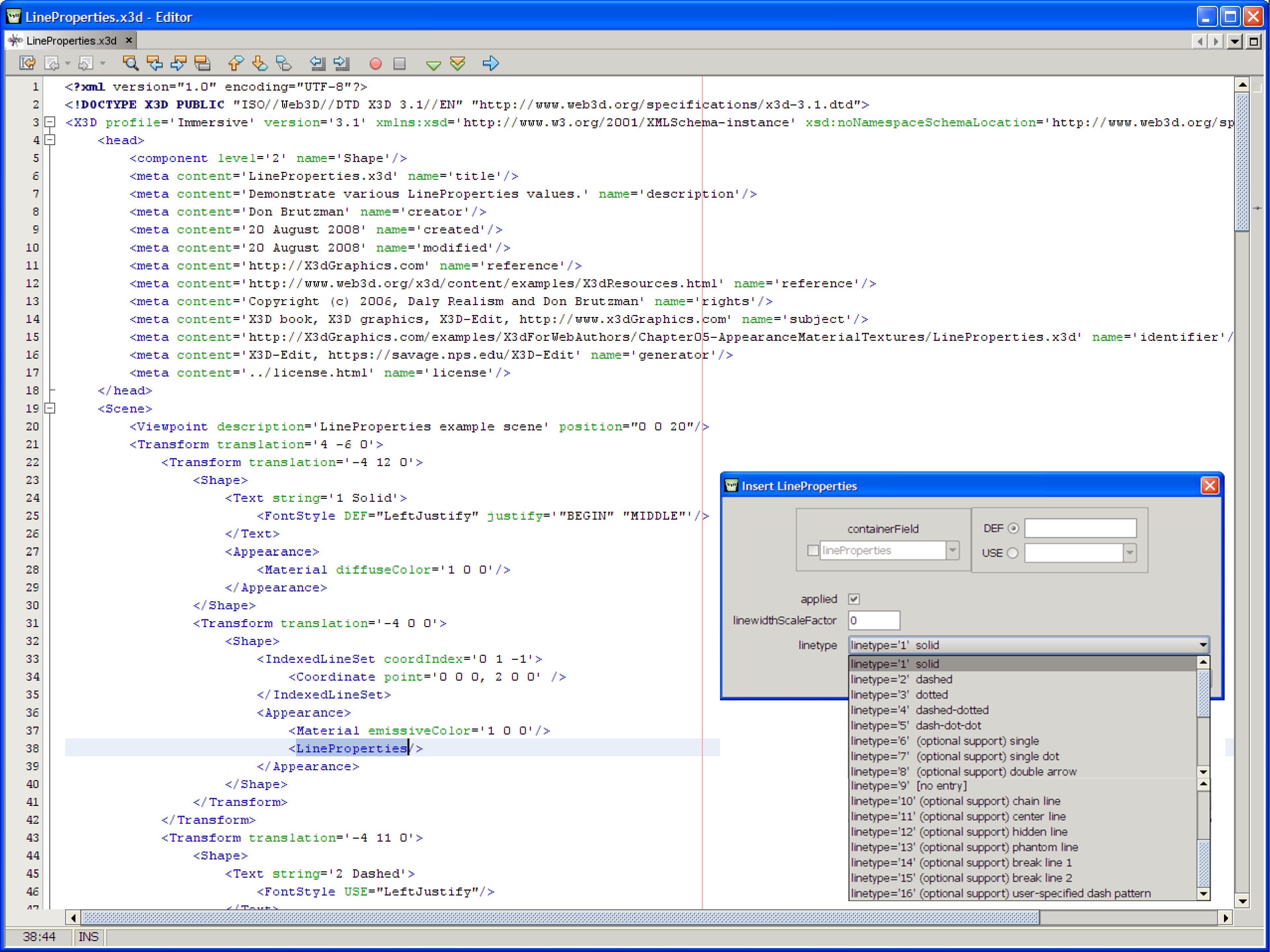
- *applied* is an SFBool field to turn the line property effects on or off, which can be set up as a helpful user-interaction technique
- *linewidthScaleFactor* (note irregular capitalization) provides a multiplicative factor to scale the nominal X3D-browser line width
- *linetype* (note irregular capitalization) selects a line pattern, with allowed values listed on following slide

LineProperties *linetype* values

(parentheses indicate optional support)



Enumeration Code	linetype Pattern
1	Solid
2	Dashed
3	Dotted
4	Dashed-dotted
5	Dash-dot-dot
6	(single arrow)
7	(single dot)
8	(double arrow)
9	(chain line)
10	(center line)
11	(hidden line)
12	(phantom line)
13	(break line 1)
14	(break line 2)
15	User-specified dash pattern



```
<?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/sp
    <head>
        <component level='2' name='Shape'/>
        <meta content='LineProperties.x3d' name='title'/>
        <meta content='Demonstrate various LineProperties values.' name='description'/>
        <meta content='Don Brutzman' name='creator'/>
        <meta content='20 August 2008' name='created'/>
        <meta content='20 August 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 (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/Chapter05-AppearanceMaterialTextures/LineProperties.x3d' name='identifier'/
        <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
        <meta content='../license.html' name='license'/>
    </head>
    <Scene>
        <Viewpoint description='LineProperties example scene' position="0 0 20"/>
        <Transform translation='4 -6 0'>
            <Transform translation='-4 12 0'>
                <Shape>
                    <Text string='1 Solid'>
                        <FontStyle DEF="LeftJustify" justify='"BEGIN" "MIDDLE"'/>
                    </Text>
                    <Appearance>
                        <Material diffuseColor='1 0 0'/>
                    </Appearance>
                </Shape>
                <Transform translation='-4 0 0'>
                    <Shape>
                        <IndexedLineSet coordIndex='0 1 -1'>
                            <Coordinate point='0 0 0, 2 0 0' />
                        </IndexedLineSet>
                        <Appearance>
                            <Material emissiveColor='1 0 0'/>
                            <LineProperties/>
                        </Appearance>
                    </Shape>
                </Transform>
            </Transform>
            <Transform translation='-4 11 0'>
                <Shape>
                    <Text string='2 Dashed'>
                        <FontStyle USE="LeftJustify"/>
                    </Text>
```

Insert LineProperties

containerField ☐ lineProperties

DEF ●

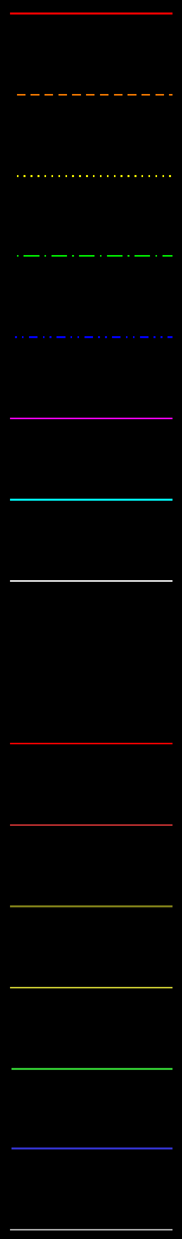
USE ○

applied ☑

linewidthScaleFactor 0

linetype linetype='1' solid

linetype='1' solid
linetype='2' dashed
linetype='3' dotted
linetype='4' dashed-dotted
linetype='5' dash-dot-dot
linetype='6' (optional support) single
linetype='7' (optional support) single dot
linetype='8' (optional support) double arrow
linetype='9' [no entry]
linetype='10' (optional support) chain line
linetype='11' (optional support) center line
linetype='12' (optional support) hidden line
linetype='13' (optional support) phantom line
linetype='14' (optional support) break line 1
linetype='15' (optional support) break line 2
linetype='16' (optional support) user-specified dash pattern



1 Solid

2 Dashed

3 Dotted

4 Dashed-dotted

5 Dash-dot-dot

6 (single arrow)

7 (single dot)

8 (double arrow)

9 [no entry]

10 (chain line)

11 (center line)

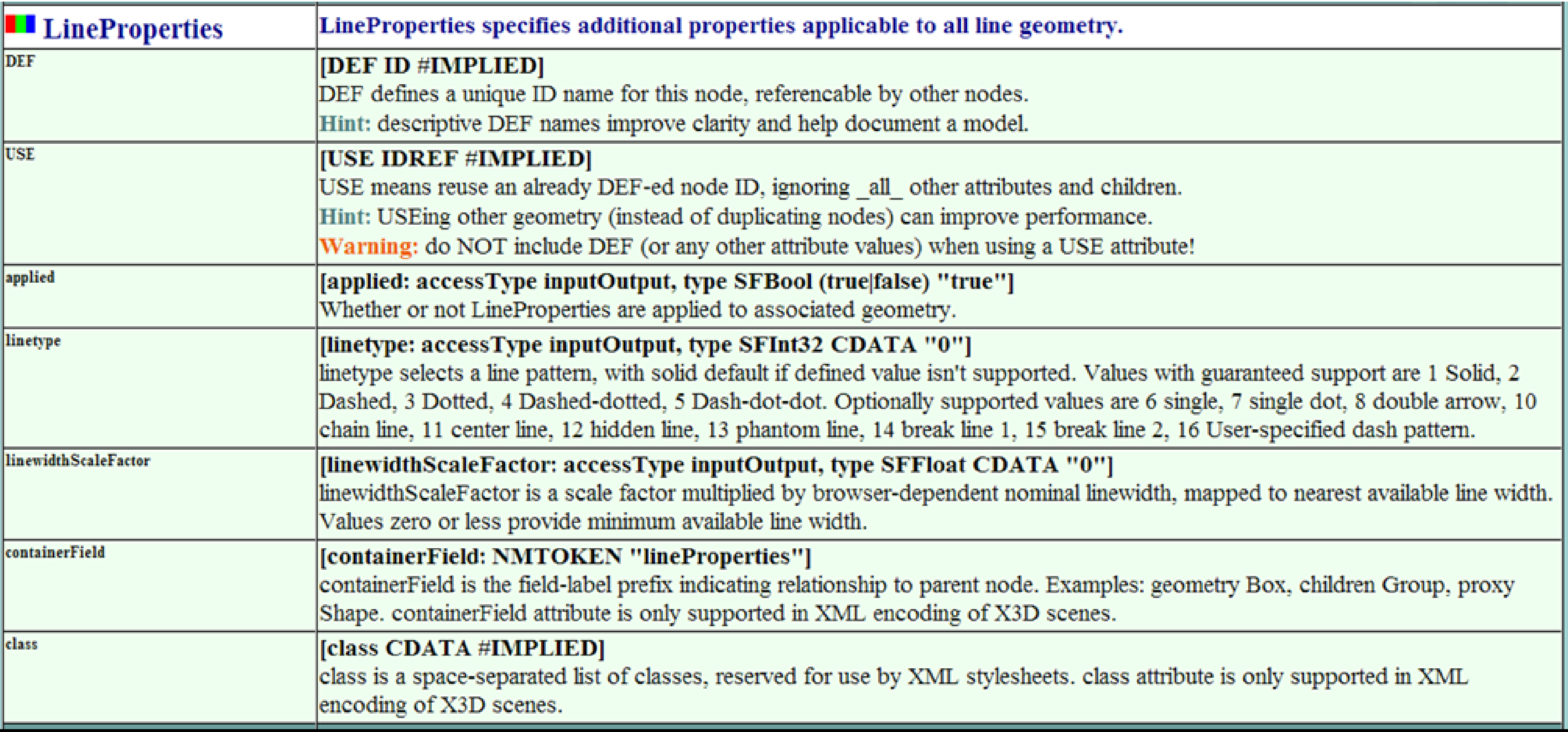
12 (hidden line)

13 (phantom line)

14 (break line 1)

15 (break line 2)

16 (user-specified dash pattern)



LineProperties	LineProperties specifies additional properties applicable to all line geometry.
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!
applied	**[applied: accessType inputOutput, type SFBool (true|false) "true"]** Whether or not LineProperties are applied to associated geometry.
linetype	**[linetype: accessType inputOutput, type SFInt32 CDATA "0"]** linetype selects a line pattern, with solid default if defined value isn't supported. Values with guaranteed support are 1 Solid, 2 Dashed, 3 Dotted, 4 Dashed-dotted, 5 Dash-dot-dot. Optionally supported values are 6 single, 7 single dot, 8 double arrow, 10 chain line, 11 center line, 12 hidden line, 13 phantom line, 14 break line 1, 15 break line 2, 16 User-specified dash pattern.
linewidthScaleFactor	**[linewidthScaleFactor: accessType inputOutput, type SFFloat CDATA "0"]** linewidthScaleFactor is a scale factor multiplied by browser-dependent nominal linewidth, mapped to nearest available line width. Values zero or less provide minimum available line width.
containerField	**[containerField: NMTOKEN "lineProperties"]** 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.

Texture nodes

Texture nodes read 2D image (or movie) files and apply them pixel-by-pixel to the associated geometry sharing the same Shape node

- Thus wrapping picture images around an object
- ImageTexture, PixelTexture, MovieTexture
- Can be inexpensive way to achieve high fidelity

Texture images can be shifted, rotated, scaled

- TextureTransform, TextureCoordinate
- Thus modifying image application to geometry

Texture coordinates 1

Defined by a 2D *(s, t)* coordinate system

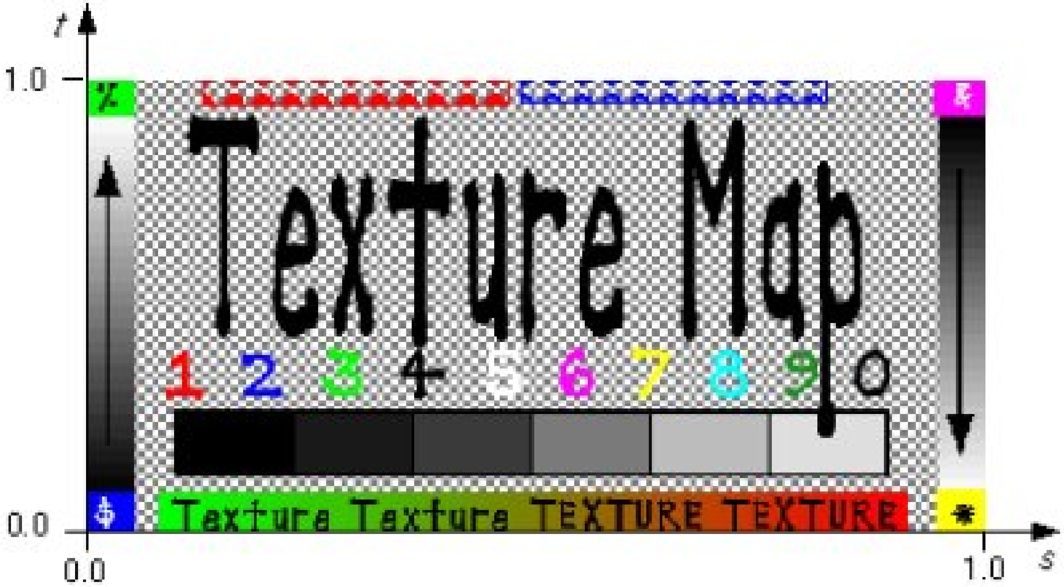
- Ranges from [0,1] along lateral *s* and vertical *t* axes
- Bottom edge of image is *s*-axis (*t*=0)
- Left edge of image is *t*-axis (*s*=0)
- Top-right corner is (*s*, *t*) = (1, 1)

Thus texture maps provide a 2D color function that find the pixel in an image at location *(s, t)* to return value of *color(s, t)*

Texture coordinates 2

s and *t* coordinates locate each pixel in an image

- Thus texture coordinates work independently of either file size (bytes), image size (pixel count) or aspect ratio (width:height)



Common fields for texture nodes

repeatS and *repeatT*

- These boolean fields indicate whether the texture image is repeated along a given axis once used
- Default is to use once along each axis, mapping the texture image once from coordinates (0,0) to (1,1)

Hint: rather than working with *repeatS repeatT* parameters or TextureTransform, it is often easiest to adjust a texture by modifying it within an image editor. Example follows.

Image file manipulation tools

Many tools are available for manipulating images, sometimes provided with the operating system

- Adobe Photoshop
- Microsoft Visio, Paint, Keynote (Mac)

One of best is free, open source, recommended:

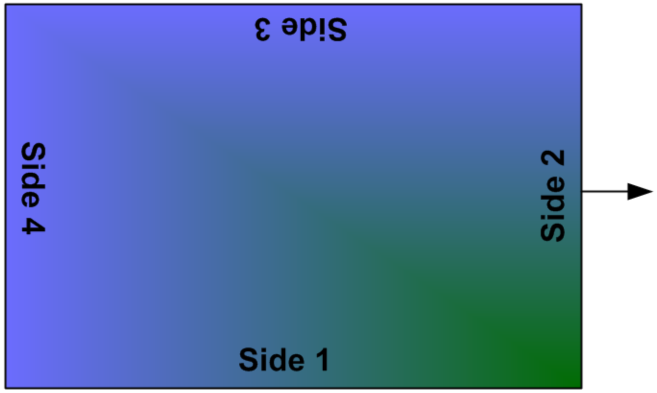
- Gnu Image Manipulation Program (GIMP) http://www.gimp.org

Drawing tools can also be helpful

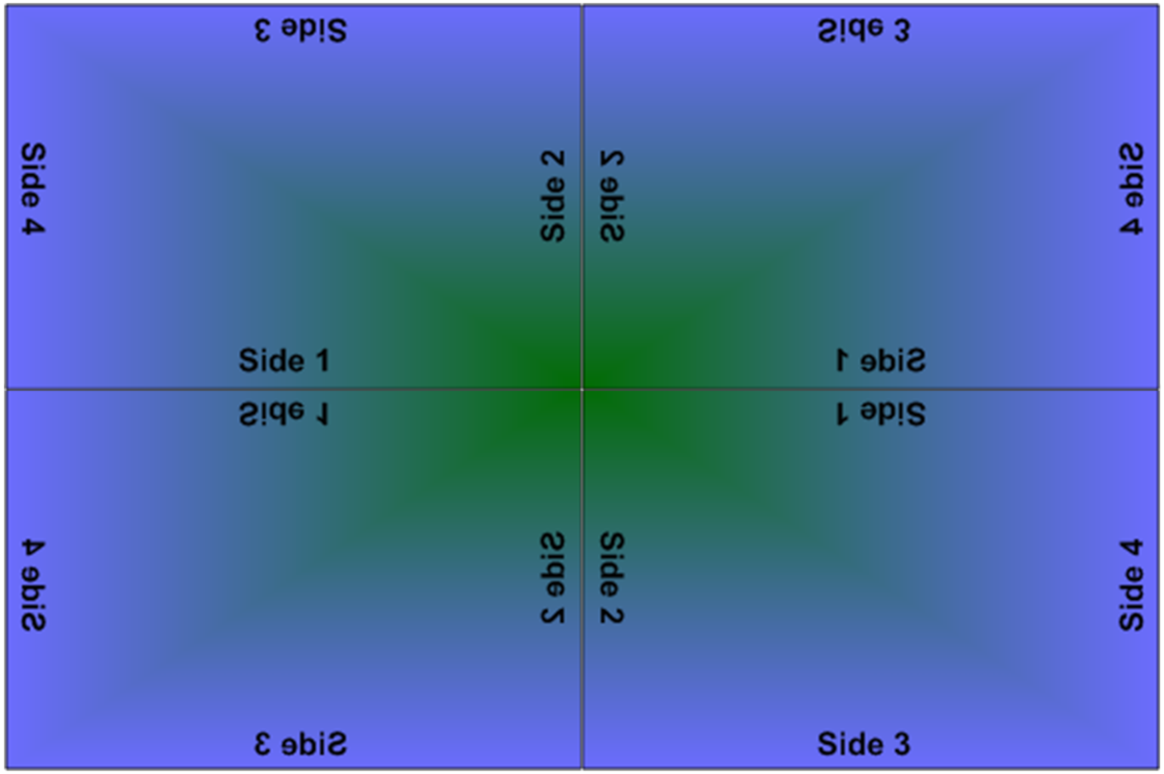
- OpenOffice Draw, Impress http://www.openOffice.org

Texture flipping for (s,t) tile repetition

a. Original image



b. First flip copy of first image across rightmost edge



c. Then flip copy of both images across bottommost edge

d. Note that all internal sides match as mirror images of each other.

Also note that external sides match:
top and bottom edges are both Side 3,
left and right edges are both Side 4.

Thus further (*s*,*t*) repetition also matches when additional texture tiling occurs.

ImageTexture node

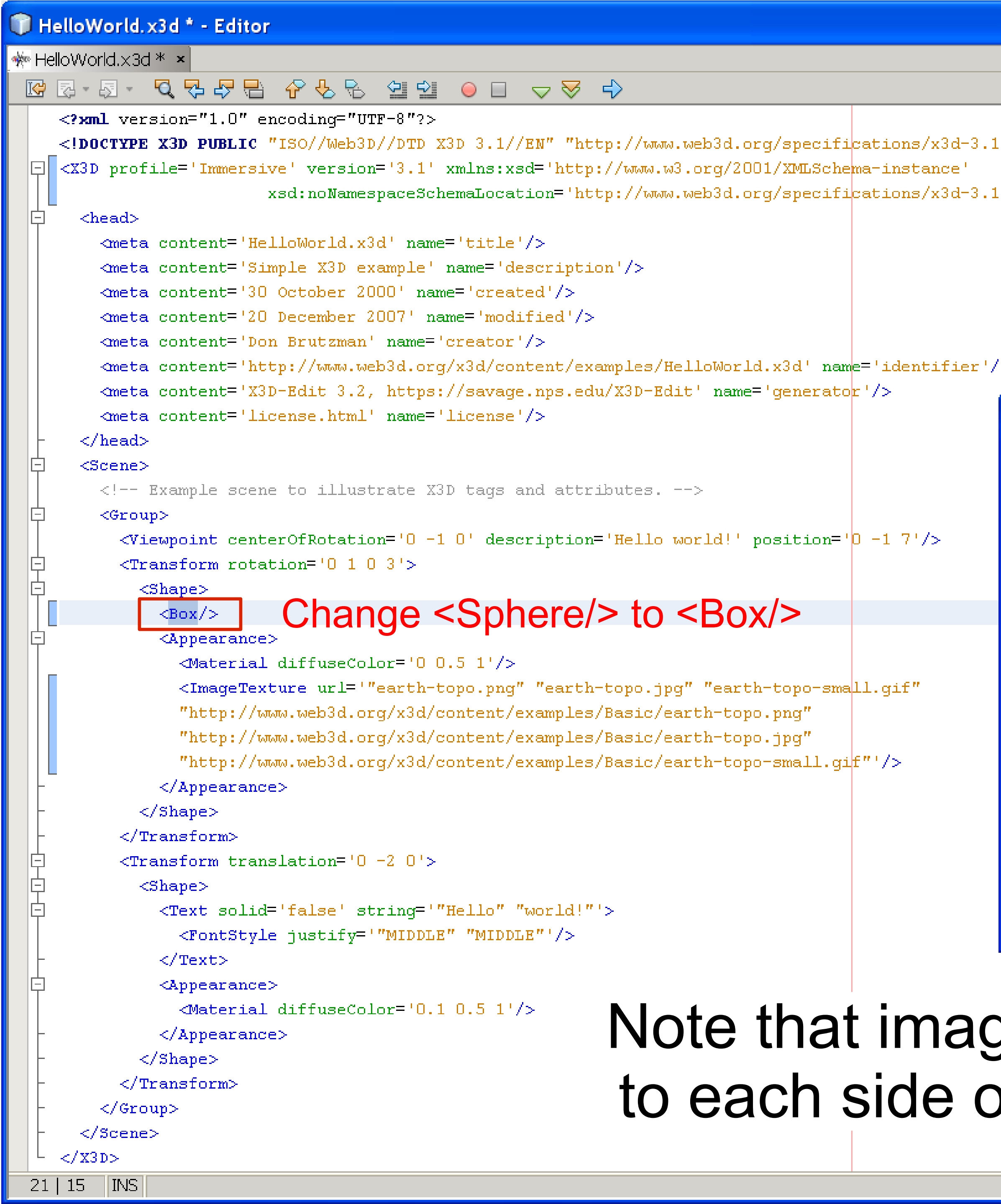
ImageTexture retrieves a 2D image file and applies it as a texture to geometry

- Commonly used technique, important to master

url described in Chapter 4 Grouping Nodes

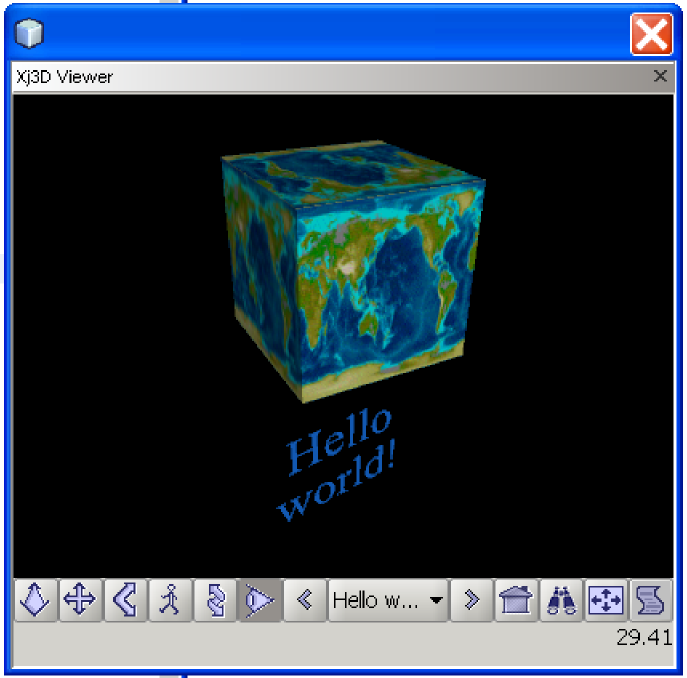
- as part of Inline and Anchor
- Recall that the url field is an ordered list which can include both local (relative) and online addresses to image files
- Might preferentially load online version first, perhaps if it can be updated, and keep a local url value for a backup image

ImageTexture example



```
<?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='HelloWorld.x3d' name='title'/>
    <meta content='Simple X3D example' name='description'/>
    <meta content='30 October 2000' name='created'/>
    <meta content='20 December 2007' name='modified'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='http://www.web3d.org/x3d/content/examples/HelloWorld.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>
    <!-- Example scene to illustrate X3D tags and attributes. -->
    <Group>
      <Viewpoint centerOfRotation='0 -1 0' description='Hello world!' position='0 -1 7'/>
      <Transform rotation='0 1 0 3'>
        <Shape>
          <Box/>
          <Appearance>
            <Material diffuseColor='0 0.5 1'/>
            <ImageTexture url='"earth-topo.png" "earth-topo.jpg" "earth-topo-small.gif"
            "http://www.web3d.org/x3d/content/examples/Basic/earth-topo.png"
            "http://www.web3d.org/x3d/content/examples/Basic/earth-topo.jpg"
            "http://www.web3d.org/x3d/content/examples/Basic/earth-topo-small.gif"'/>
          </Appearance>
        </Shape>
      </Transform>
      <Transform translation='0 -2 0'>
        <Shape>
          <Text solid='false' string='"Hello" "world!"'>
            <FontStyle justify='"MIDDLE" "MIDDLE"'/>
          </Text>
          <Appearance>
            <Material diffuseColor='0.1 0.5 1'/>
          </Appearance>
        </Shape>
      </Transform>
    </Group>
  </Scene>
</X3D>
```

Change <Sphere/> to <Box/>



Note that image is applied equally to each side of the Box geometry

ImageTexture file formats

Supported, required image file formats:

- Joint Photographic Expert Group (.jpg) which is good for photographic images
- Portable Network Graphics (.png) which is good for bit-mapped drawings and other images
- Both formats are royalty free, commonly used in Web

Also suggested (but not required)

- Graphics Image Format (.gif), has license restrictions

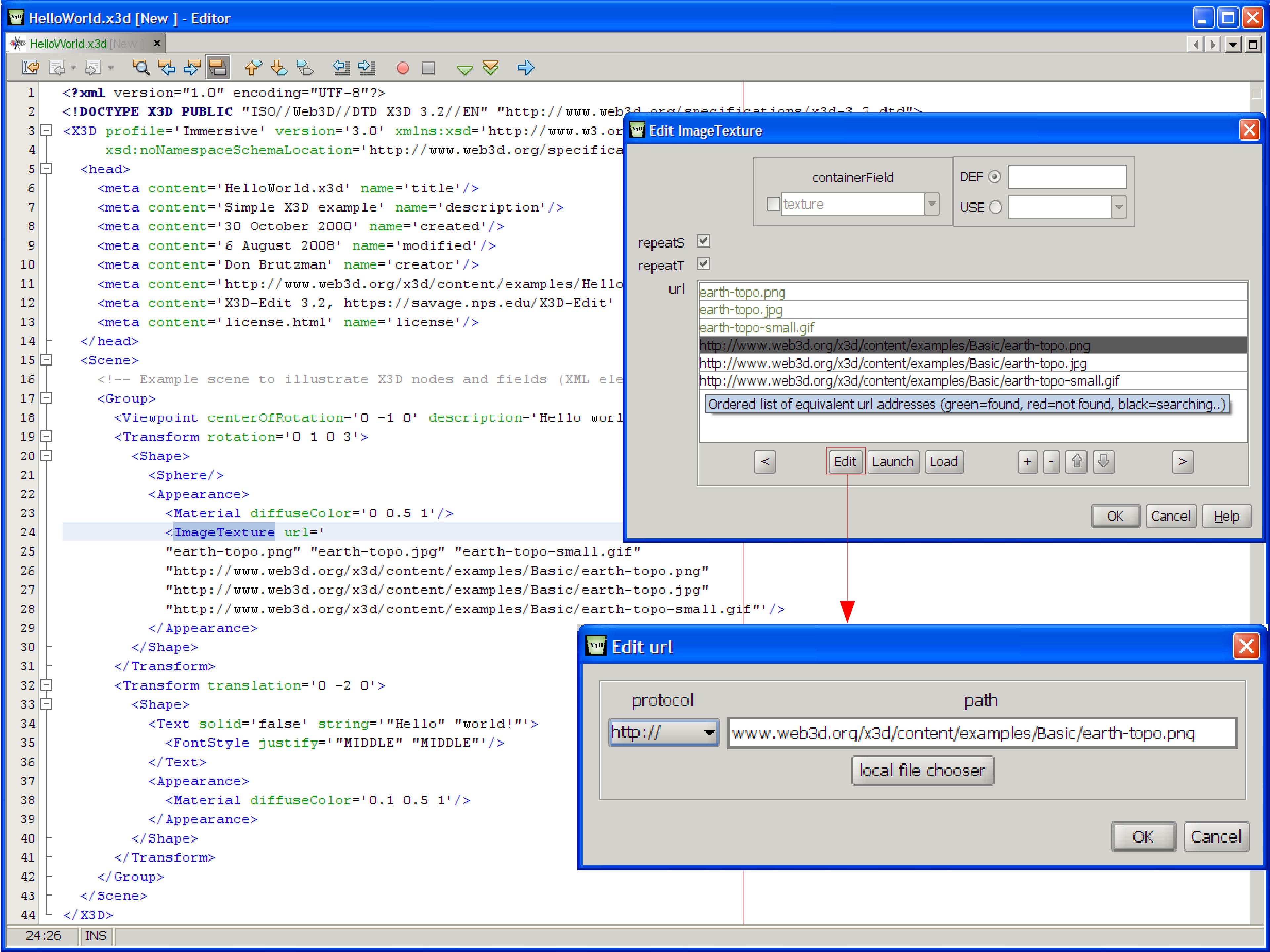
Other image formats are also allowed

- but support for users by X3D browser not guaranteed

ImageTexture and Material

It is good practice to accompany ImageTexture with a Material node

- Material is rendered first if network delays are encountered when loading the image file
- Carefully chosen Material *diffuseColor* can reduce sudden color changes when a delayed image file is finally applied
- Underlying Material values are further important and will show through if the texture image includes transparent pixels



HelloWorld.x3d [New] - Editor

```
<?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='3.0' xmlns:xsd='http://www.w3.or
     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifica
  <head>
    <meta content='HelloWorld.x3d' name='title'/>
    <meta content='Simple X3D example' name='description'/>
    <meta content='30 October 2000' name='created'/>
    <meta content='6 August 2008' name='modified'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='http://www.web3d.org/x3d/content/examples/Hello
    <meta content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit'
    <meta content='license.html' name='license'/>
  </head>
  <Scene>
    <!-- Example scene to illustrate X3D nodes and fields (XML ele
    <Group>
      <Viewpoint centerOfRotation='0 -1 0' description='Hello worl
      <Transform rotation='0 1 0 3'>
        <Shape>
          <Sphere/>
          <Appearance>
            <Material diffuseColor='0 0.5 1'/>
            <ImageTexture url='
            "earth-topo.png" "earth-topo.jpg" "earth-topo-small.gif"
            "http://www.web3d.org/x3d/content/examples/Basic/earth-topo.png"
            "http://www.web3d.org/x3d/content/examples/Basic/earth-topo.jpg"
            "http://www.web3d.org/x3d/content/examples/Basic/earth-topo-small.gif"'/>
          </Appearance>
        </Shape>
      </Transform>
      <Transform translation='0 -2 0'>
        <Shape>
          <Text solid='false' string='"Hello" "world!"'>
            <FontStyle justify='"MIDDLE" "MIDDLE"'/>
          </Text>
          <Appearance>
            <Material diffuseColor='0.1 0.5 1'/>
          </Appearance>
        </Shape>
      </Transform>
    </Group>
  </Scene>
</X3D>
```

24:26 INS

Edit ImageTexture

containerField: □ texture

DEF ●

USE ○

repeatS ☑

repeatT ☑

url:

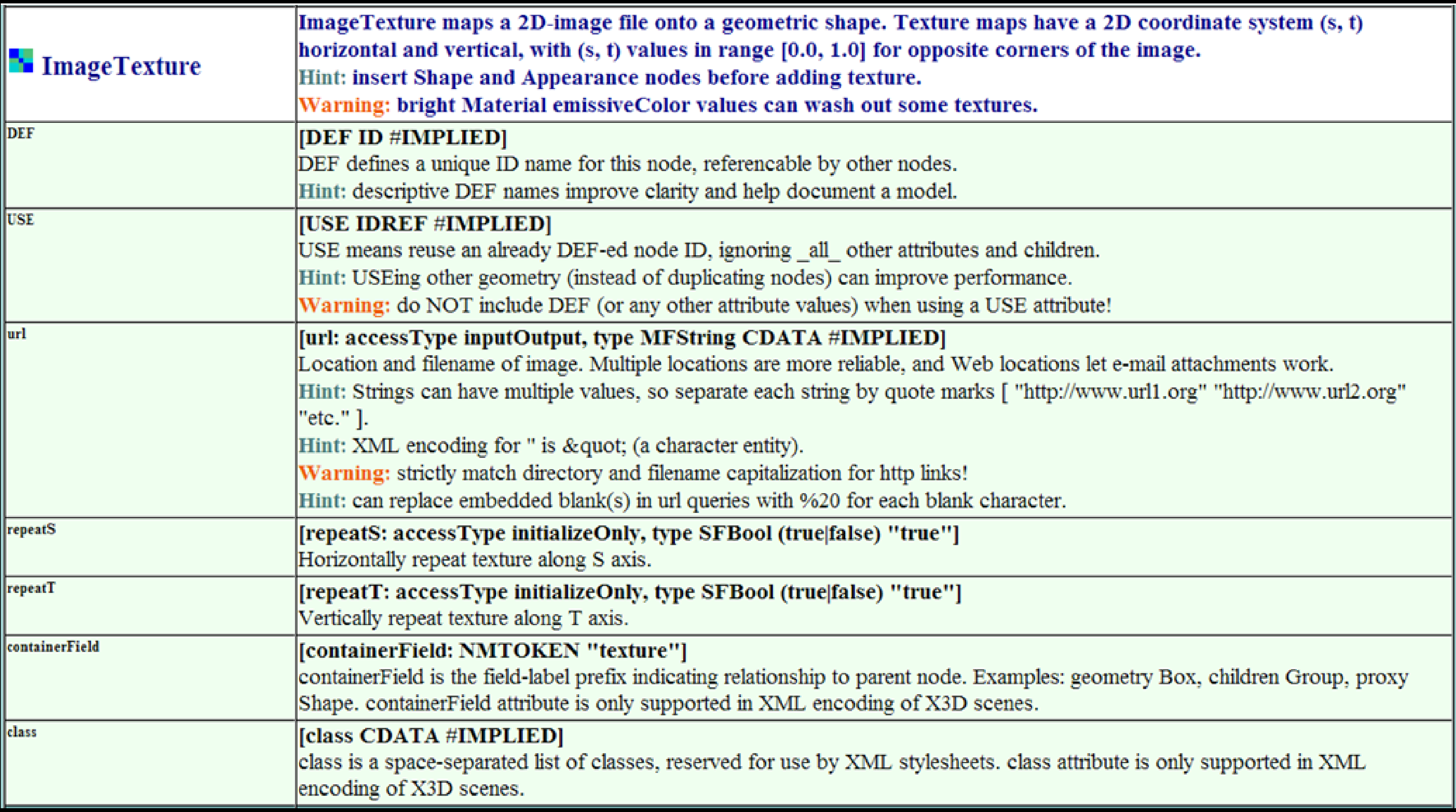
- earth-topo.png
- earth-topo.jpg
- earth-topo-small.gif
- http://www.web3d.org/x3d/content/examples/Basic/earth-topo.png
- http://www.web3d.org/x3d/content/examples/Basic/earth-topo.jpg
- http://www.web3d.org/x3d/content/examples/Basic/earth-topo-small.gif

Ordered list of equivalent url addresses (green=found, red=not found, black=searching..)

Edit url

protocol: http://

path: www.web3d.org/x3d/content/examples/Basic/earth-topo.png



ImageTexture	**ImageTexture maps a 2D-image file onto a geometric shape. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image.** **Hint: insert Shape and Appearance nodes before adding texture.** **Warning: bright Material emissiveColor values can wash out some textures.**
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!
url	**[url: accessType inputOutput, type MFString CDATA #IMPLIED]** Location and filename of image. Multiple locations are more reliable, and Web locations let e-mail attachments work. 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.
repeatS	**[repeatS: accessType initializeOnly, type SFBool (true|false) "true"]** Horizontally repeat texture along S axis.
repeatT	**[repeatT: accessType initializeOnly, type SFBool (true|false) "true"]** Vertically repeat texture along T axis.
containerField	**[containerField: NMTOKEN "texture"]** 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.

MovieTexture node

MovieTexture applies video imagery to geometry

- Same considerations for *url* field as ImageTexture

Often applied sparingly, because

- movie files are often quite large
- Applying high-frame rate pixels is computationally expensive, which can slow down frame rate and may present low-quality results in some browsers

3D wall or billboard helps make movies viewable

- Important to provide a clear Viewpoint to see it
- Authors might prefer for movie to instead play within Web browser outside X3D scene

MovieTexture file formats

Supported, required image file formats:

- Motion Picture Expert Group (.mpg format, MPEG-2) which is good for low (or moderate) bandwidth video

Other movie formats are also allowed

- but consistent support for users among X3D browsers is not guaranteed, unless royalty free (RF)

Expected future work

- Possible merger of streaming video, streaming X3D?
- Web3D Consortium will not approve video formats for Web use by X3D unless they are royalty free
- Related efforts: World Wide Web Consortium

MovieTexture fields 1

- *speed* is a rate factor to speed up or slow down movie playback, can be negative to go in reverse
- *startTime* and *stopTime* are used as input controls to begin and end play, usually by routing an SFTime event from a TimeSensor or TouchSensor
- *pauseTime* and *resumeTime* operate similarly, allowing the movie to pause/resume at same point in time (rather than starting over from beginning)
- *isActive* and *isPaused* are boolean output events that are sent by the MovieTexture node: true when the condition occurs, false when it ends

MovieTexture fields 2

- *duration_changed* is length of time in seconds for one cycle of the movie
- *elapsedTime* is SFTime output event sent continuously as movie is playing, cumulatively in seconds without counting any pause durations

Can use LoadSensor (chapter 12) to detect when movie is fully loaded

DEF and USE are important for multiple copies

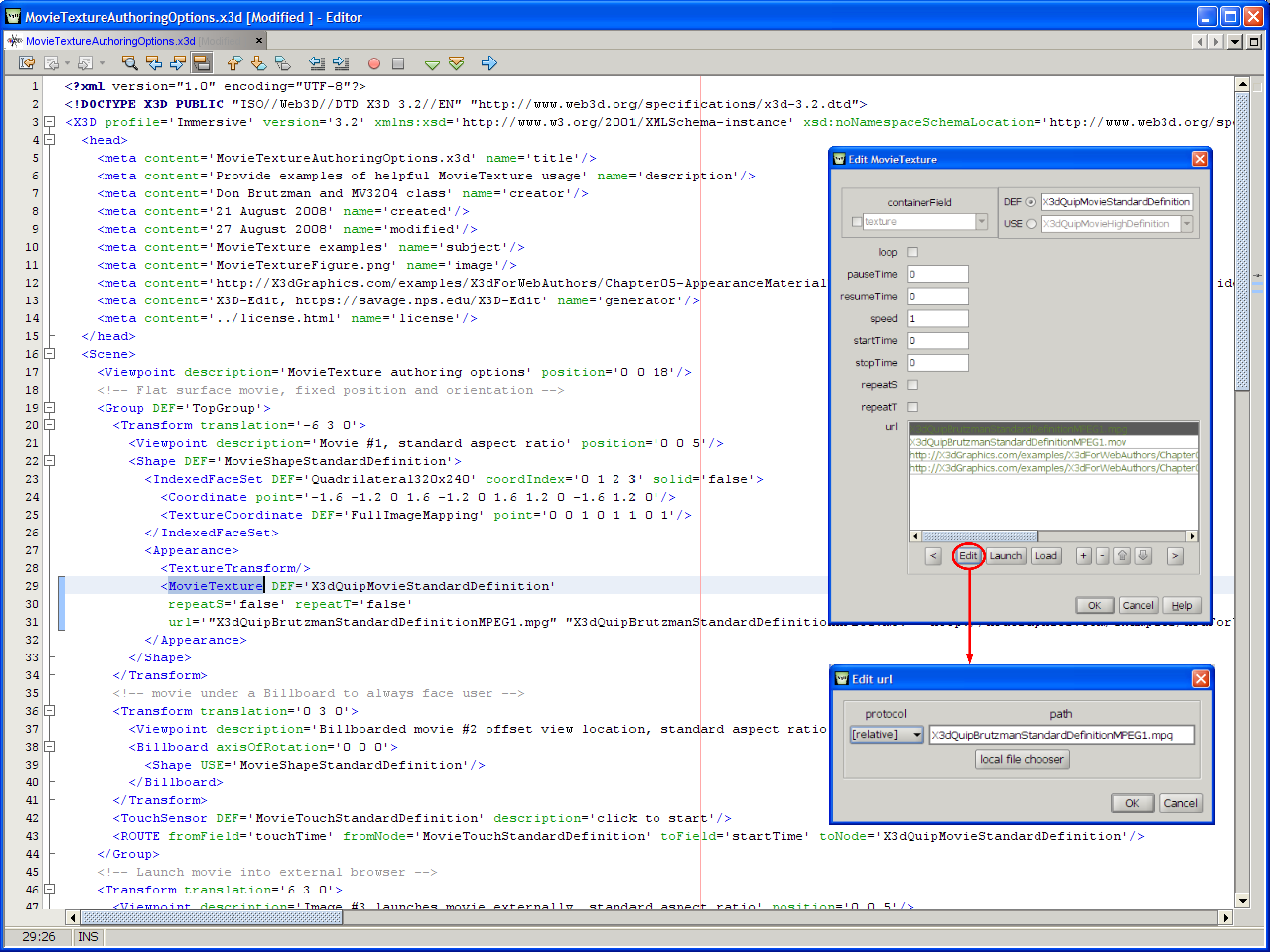
- Minimize download file size, bandwidth, and delay

MovieTexture and Material

It is good practice to accompany MovieTexture with a Material node

- Material is rendered first if network delays are encountered when loading the image file
- Carefully chosen Material *diffuseColor* can reduce sudden color changes when a delayed image file is finally applied
- Underlying Material values are further important and will show through if the MovieTexture images include transparent pixels

Same considerations as ImageTexture



MovieTextureAuthoringOptions.x3d [Modified] - Editor

MovieTextureAuthoringOptions.x3d

```
<?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/sp
  <head>
    <meta content='MovieTextureAuthoringOptions.x3d' name='title'/>
    <meta content='Provide examples of helpful MovieTexture usage' name='description'/>
    <meta content='Don Brutzman and MV3204 class' name='creator'/>
    <meta content='21 August 2008' name='created'/>
    <meta content='27 August 2008' name='modified'/>
    <meta content='MovieTexture examples' name='subject'/>
    <meta content='MovieTextureFigure.png' name='image'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterial                              id
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Viewpoint description='MovieTexture authoring options' position='0 0 18'/>
    <!-- Flat surface movie, fixed position and orientation -->
    <Group DEF='TopGroup'>
      <Transform translation='-6 3 0'>
        <Viewpoint description='Movie #1, standard aspect ratio' position='0 0 5'/>
        <Shape DEF='MovieShapeStandardDefinition'>
          <IndexedFaceSet DEF='Quadrilateral320x240' coordIndex='0 1 2 3' solid='false'>
            <Coordinate point='-1.6 -1.2 0 1.6 -1.2 0 1.6 1.2 0 -1.6 1.2 0'/>
            <TextureCoordinate DEF='FullImageMapping' point='0 0 1 0 1 1 0 1'/>
          </IndexedFaceSet>
          <Appearance>
            <TextureTransform/>
            <MovieTexture DEF='X3dQuipMovieStandardDefinition'
             repeatS='false' repeatT='false'
             url='"X3dQuipBrutzmanStandardDefinitionMPEG1.mpg" "X3dQuipBrutzmanStandardDefinitio                              or
          </Appearance>
        </Shape>
      </Transform>
      <!-- movie under a Billboard to always face user -->
      <Transform translation='0 3 0'>
        <Viewpoint description='Billboarded movie #2 offset view location, standard aspect ratio
        <Billboard axisOfRotation='0 0 0'>
          <Shape USE='MovieShapeStandardDefinition'/>
        </Billboard>
      </Transform>
      <TouchSensor DEF='MovieTouchStandardDefinition' description='click to start'/>
      <ROUTE fromField='touchTime' fromNode='MovieTouchStandardDefinition' toField='startTime' toNode='X3dQuipMovieStandardDefinition'/>
    </Group>
    <!-- Launch movie into external browser -->
    <Transform translation='6 3 0'>
      <Viewpoint description='Image #3 launches movie externally, standard aspect ratio' position='0 0 5'/>
```

29:26 INS

Edit MovieTexture

containerField □ texture

DEF ● X3dQuipMovieStandardDefinition

USE ○ X3dQuipMovieHighDefinition

loop □

pauseTime 0

resumeTime 0

speed 1

startTime 0

stopTime 0

repeatS □

repeatT □

url

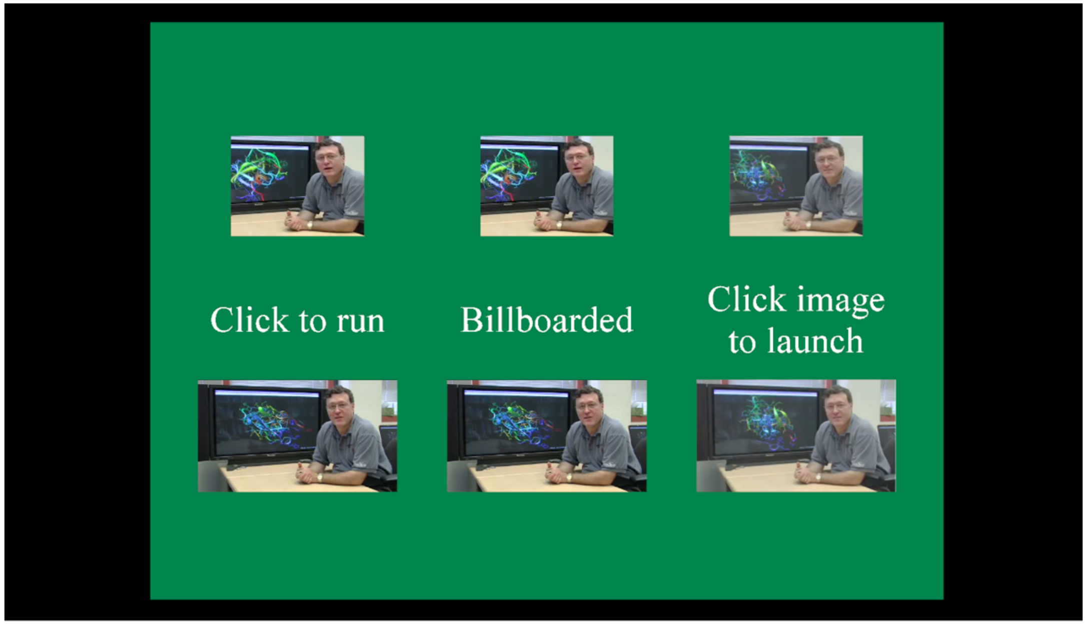
X3dQuipBrutzmanStandardDefinitionMPEG1.mpg
X3dQuipBrutzmanStandardDefinitionMPEG1.mov
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter0
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter0

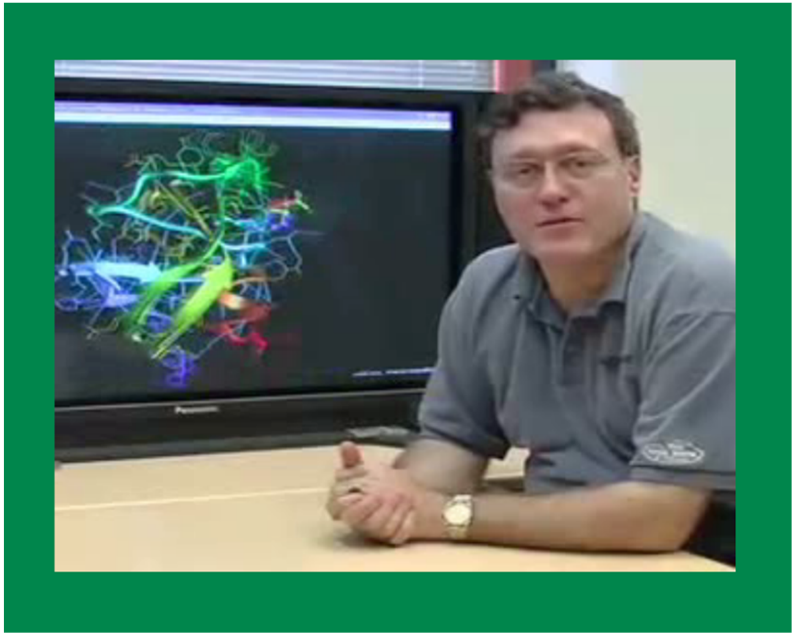
Edit url

protocol: [relative]

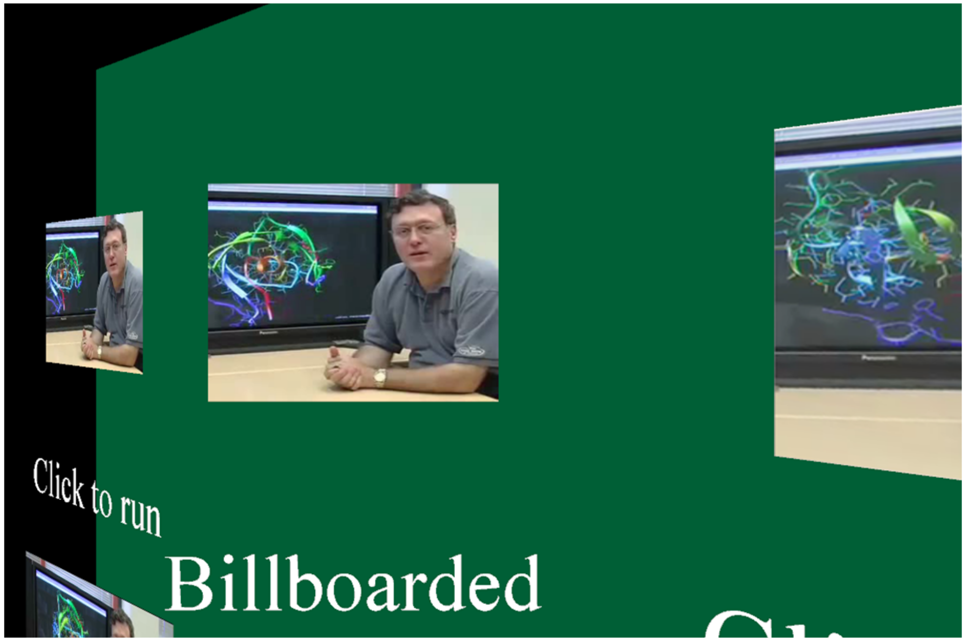
path: X3dQuipBrutzmanStandardDefinitionMPEG1.mpg

Entry view

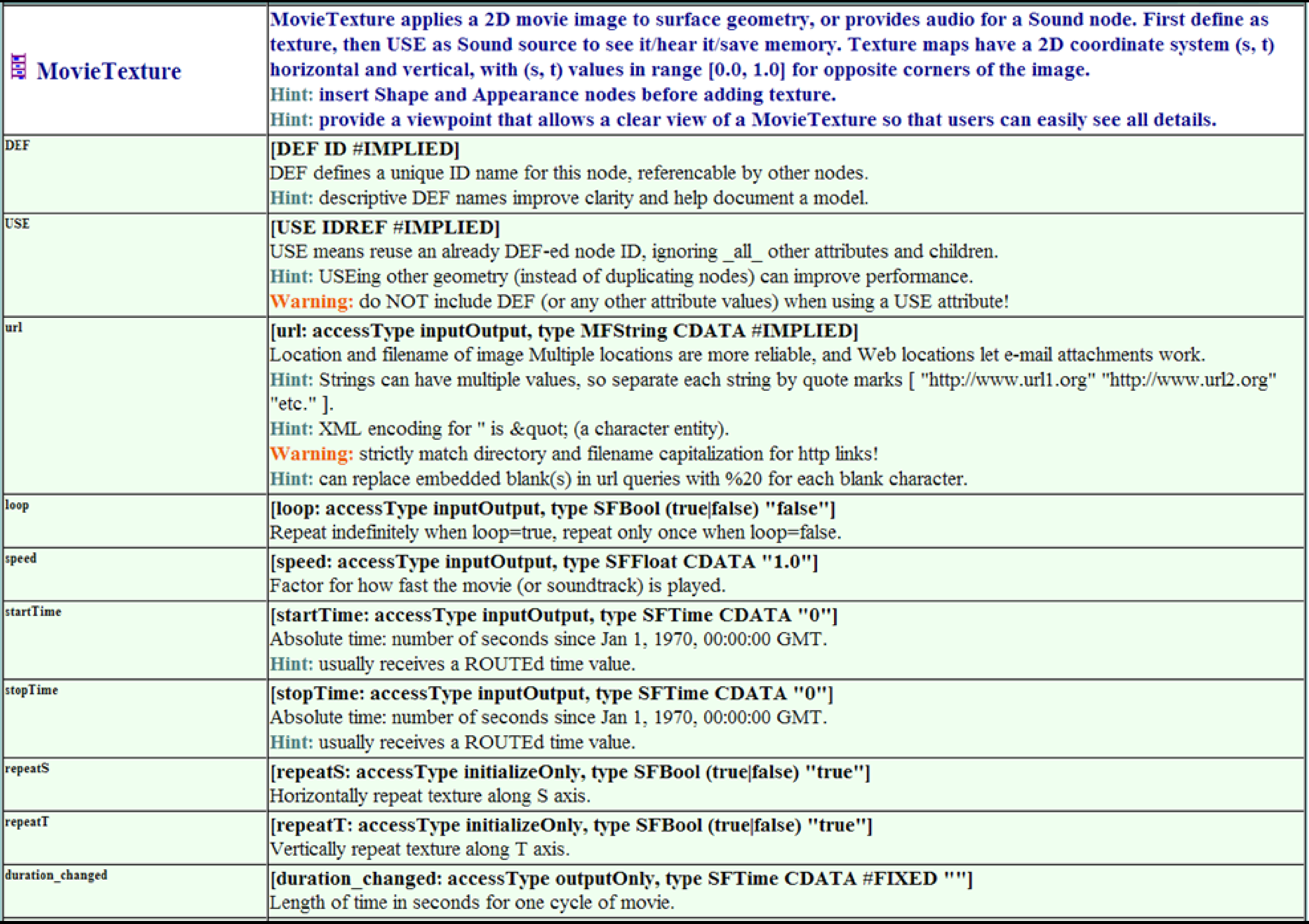




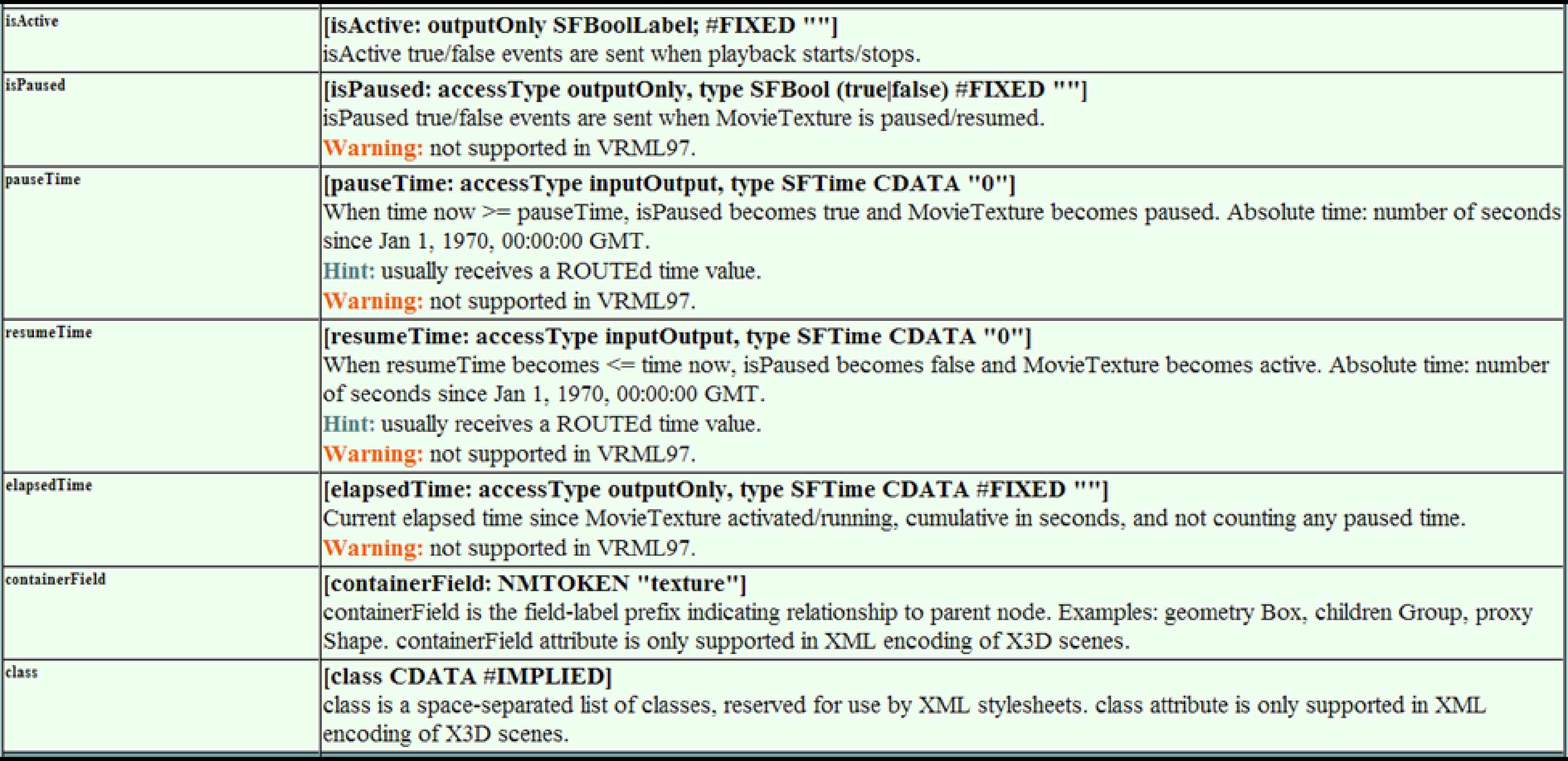
Upper left view



Upper center view



MovieTexture	MovieTexture applies a 2D movie image to surface geometry, or provides audio for a Sound node. First define as texture, then USE as Sound source to see it/hear it/save memory. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image. Hint: insert Shape and Appearance nodes before adding texture. Hint: provide a viewpoint that allows a clear view of a MovieTexture so that users can easily see all details.
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!
url	**[url: accessType inputOutput, type MFString CDATA #IMPLIED]** Location and filename of image Multiple locations are more reliable, and Web locations let e-mail attachments work. 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.
loop	**[loop: accessType inputOutput, type SFBool (true|false) "false"]** Repeat indefinitely when loop=true, repeat only once when loop=false.
speed	**[speed: accessType inputOutput, type SFFloat CDATA "1.0"]** Factor for how fast the movie (or soundtrack) is played.
startTime	**[startTime: accessType inputOutput, type SFTime CDATA "0"]** Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. Hint: usually receives a ROUTEd time value.
stopTime	**[stopTime: accessType inputOutput, type SFTime CDATA "0"]** Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. Hint: usually receives a ROUTEd time value.
repeatS	**[repeatS: accessType initializeOnly, type SFBool (true|false) "true"]** Horizontally repeat texture along S axis.
repeatT	**[repeatT: accessType initializeOnly, type SFBool (true|false) "true"]** Vertically repeat texture along T axis.
duration_changed	**[duration_changed: accessType outputOnly, type SFTime CDATA #FIXED ""]** Length of time in seconds for one cycle of movie.



isActive	**[isActive: outputOnly SFBoolLabel; #FIXED ""]** isActive true/false events are sent when playback starts/stops.
isPaused	**[isPaused: accessType outputOnly, type SFBool (true|false) #FIXED ""]** isPaused true/false events are sent when MovieTexture is paused/resumed. **Warning:** not supported in VRML97.
pauseTime	**[pauseTime: accessType inputOutput, type SFTime CDATA "0"]** When time now >= pauseTime, isPaused becomes true and MovieTexture becomes paused. Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. **Hint:** usually receives a ROUTEd time value. **Warning:** not supported in VRML97.
resumeTime	**[resumeTime: accessType inputOutput, type SFTime CDATA "0"]** When resumeTime becomes <= time now, isPaused becomes false and MovieTexture becomes active. Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. **Hint:** usually receives a ROUTEd time value. **Warning:** not supported in VRML97.
elapsedTime	**[elapsedTime: accessType outputOnly, type SFTime CDATA #FIXED ""]** Current elapsed time since MovieTexture activated/running, cumulative in seconds, and not counting any paused time. **Warning:** not supported in VRML97.
containerField	**[containerField: NMTOKEN "texture"]** 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.

PixelTexture node

PixelTexture contains the bit pattern of an image

- Written out as set of numeric data within the node

This allows single X3D scene to embed imagery

- Which helps when delivering a self-sufficient scene
- However may increase overall file size

Numeric image data is encoded pixel by pixel, using a special data type: SFImage

- After defining array dimensions, each individual number entered in pixel field corresponds to a black/white, black/white/alpha, RGB or RGBA value

SFImage data type

First three data values:

- Number of width pixels in image
- Number of height pixels in image
- Number of components in each pixel value (0-4)

Component count represented by each pixel value:

- 0 for no image, `<ImageTexture image='0 0 0'/>`
- 1 for black-white intensity
- 2 for black-white intensity, transparency
- 3 for red-green-blue colors
- 4 for red-green-blue colors, transparency

Array then holds appropriate number of pixel values

SFImage examples

Components	SFImage Value	Pixel count	Description	Image
0	0 0 0	0	Empty image	
1	1 2 1, 0xFF 0x00	2	Intensity (black & white) example: checkerboard pattern	
2	2 1 2, 0xCCFF 0x2277	2	Intensity & transparency example	
3	2 4 3, 0xFF0000 0xFF00 0 0 0 0 0xFFFFFF 0xFFFF00	8	Red-green-blue (RGB) example	
4	3 2 4, 0xFF0000FF 0x00FF00FF 0x0000FFFF 0xFF00007F 0x00FF007F **0x0000FF7F**	6	Red-green-blue-alpha (RGBA) example	

Each numeric pixel entry is a single component value

note erratum in book

Hexadecimal number representation

Hexadecimal (base 16) and decimal (base 10) are both permitted for any X3D numeric data

- Hex commonly used in SFImage, easier to read

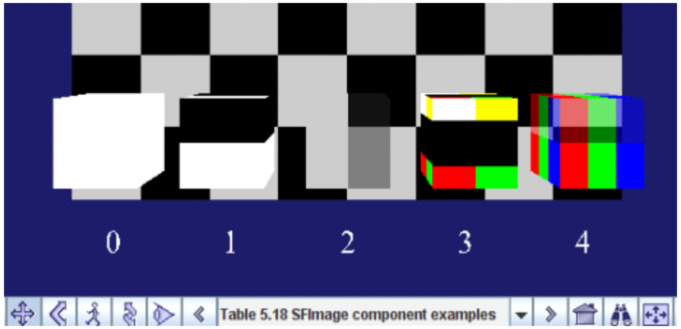
Base 10 and base 16 digits:

- 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
- 0 1 2 3 4 5 6 7 8 9 a b c d e f

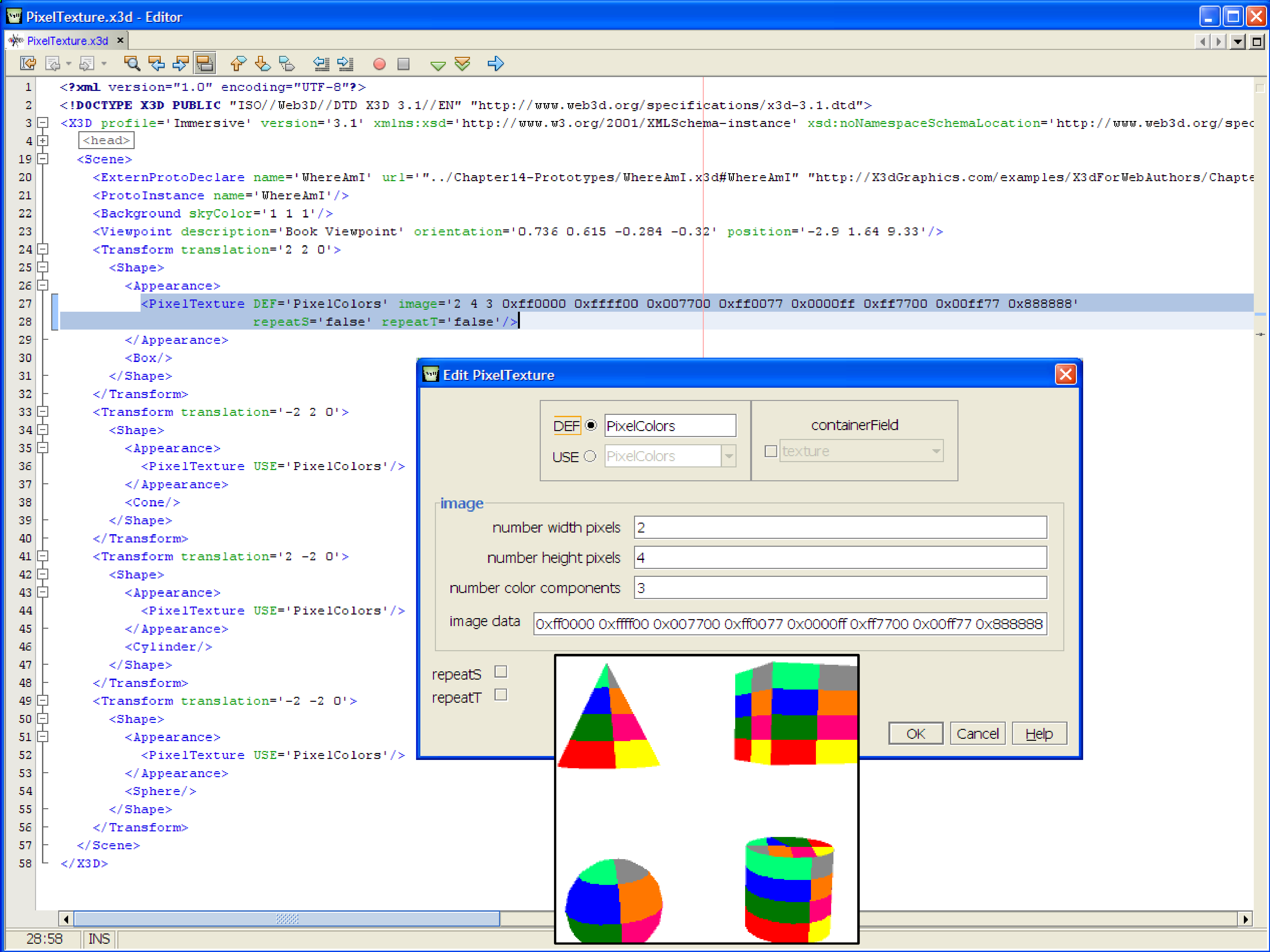
Syntax and examples:

- Precede hex values with 0x, thus hex 0xC=15_{10}
- 0x0012BE = $1 \cdot 16^3 + 2 \cdot 16^2 + 11 \cdot 16^1 + 14 = 4798_{10}$
  - ➢ = 0 red, (0x12=47)/255 green, (0xBE=190)/255 blue
  - ➢ = color value (0 red, 0.184 green, 0.745 blue)

Example value conversions

Hex value	0-255 range	bw	r	g	b	a	transparency	color
0xFF	255	1						⬜
0x00	0	0						⬛
0xCCFF	204	0.8				255	0.0	▒
0x2277	34	0.13				119	0.53	▓
0xFF0000	255 0 0		1	0	0			🟥
0xFF00	0 255 0		0	1	0			🟩
0xFFFFFF	255 255 255		1	1	1			⬜
0xFFFF00	255 255 0		1	1	0			🟨
 0 1 2 3 4 Table 5.18 SFImage component examples			1	0	0	255	0.0	🟥
			0	1	0	255	0.0	🟩
			0	0	1	255	0.0	🟦
			1	0	0	127	0.5	🟥
			0	1	0	127	0.5	🟩
			0	0	1	127	0.5	🟦

Note that alpha a = (1- transparency)



PixelTexture.x3d - Editor

PixelTexture.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='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/spec
  <head>
  <Scene>
    <ExternProtoDeclare name='WhereAmI' url='"../Chapter14-Prototypes/WhereAmI.x3d#WhereAmI" "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapte
    <ProtoInstance name='WhereAmI'/>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book Viewpoint' orientation='0.736 0.615 -0.284 -0.32' position='-2.9 1.64 9.33'/>
    <Transform translation='2 2 0'>
      <Shape>
        <Appearance>
          <PixelTexture DEF='PixelColors' image='2 4 3 0xff0000 0xffff00 0x007700 0xff0077 0x0000ff 0xff7700 0x00ff77 0x888888'
                  repeatS='false' repeatT='false'/>
        </Appearance>
        <Box/>
      </Shape>
    </Transform>
    <Transform translation='-2 2 0'>
      <Shape>
        <Appearance>
          <PixelTexture USE='PixelColors'/>
        </Appearance>
        <Cone/>
      </Shape>
    </Transform>
    <Transform translation='2 -2 0'>
      <Shape>
        <Appearance>
          <PixelTexture USE='PixelColors'/>
        </Appearance>
        <Cylinder/>
      </Shape>
    </Transform>
    <Transform translation='-2 -2 0'>
      <Shape>
        <Appearance>
          <PixelTexture USE='PixelColors'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Edit PixelTexture

DEF ● PixelColors

USE ○ PixelColors

containerField

□ texture

image

number width pixels: 2

number height pixels: 4

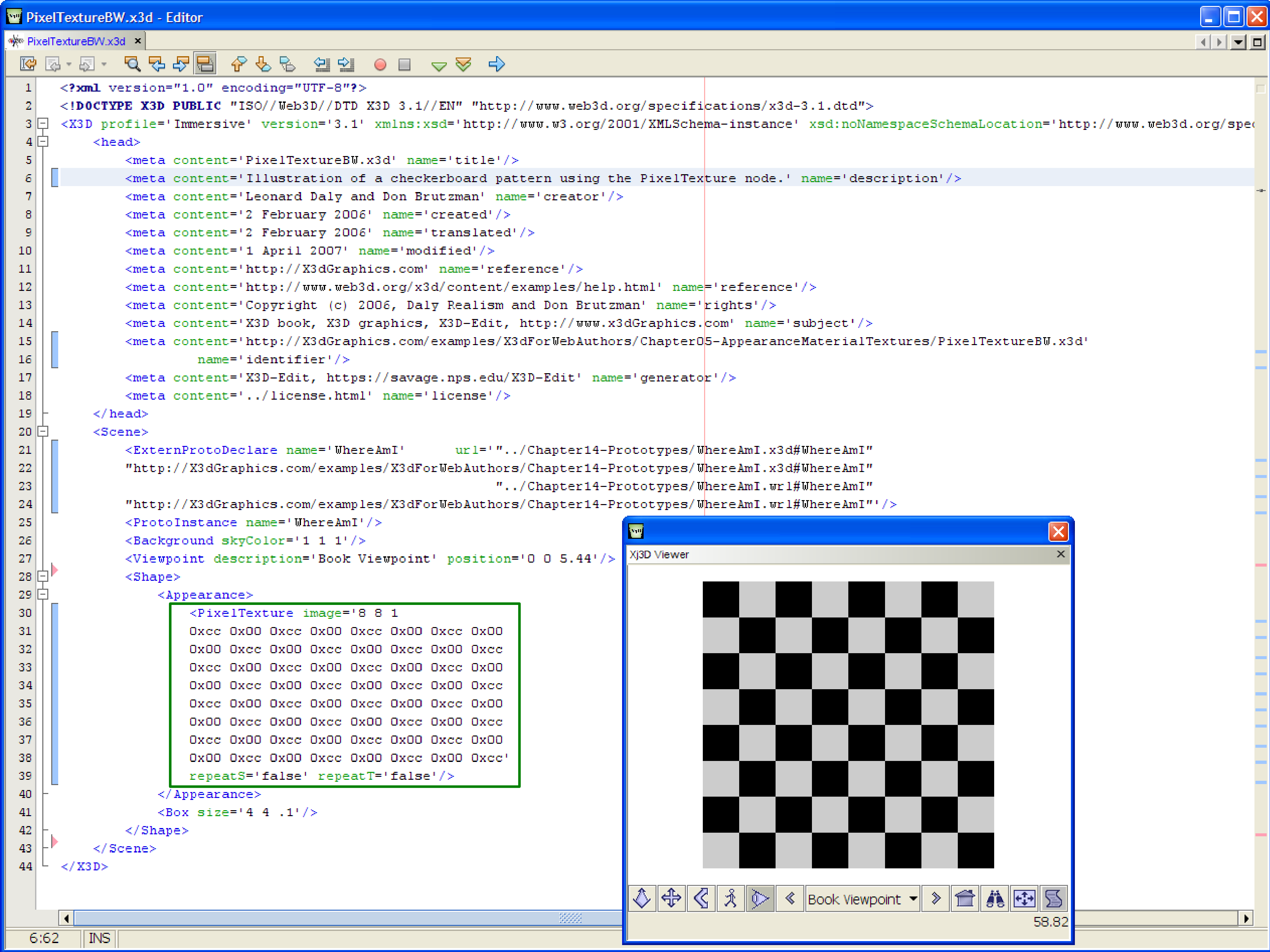
number color components: 3

image data: 0xff0000 0xffff00 0x007700 0xff0077 0x0000ff 0xff7700 0x00ff77 0x888888

repeatS □

repeatT □

28:58 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/spec
    <head>
        <meta content='PixelTextureBW.x3d' name='title'/>
        <meta content='Illustration of a checkerboard pattern using the PixelTexture node.' name='description'/>
        <meta content='Leonard Daly and Don Brutzman' name='creator'/>
        <meta content='2 February 2006' name='created'/>
        <meta content='2 February 2006' name='translated'/>
        <meta content='1 April 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/Chapter05-AppearanceMaterialTextures/PixelTextureBW.x3d'
                 name='identifier'/>
        <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
        <meta content='../license.html' name='license'/>
    </head>
    <Scene>
        <ExternProtoDeclare name='WhereAmI'      url='"../Chapter14-Prototypes/WhereAmI.x3d#WhereAmI"
        "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter14-Prototypes/WhereAmI.x3d#WhereAmI"
                                                      "../Chapter14-Prototypes/WhereAmI.wrl#WhereAmI"
        "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter14-Prototypes/WhereAmI.wrl#WhereAmI"'/>
        <ProtoInstance name='WhereAmI'/>
        <Background skyColor='1 1 1'/>
        <Viewpoint description='Book Viewpoint' position='0 0 5.44'/>
        <Shape>
            <Appearance>
                <PixelTexture image='8 8 1
                0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00
                0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc
                0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00
                0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc
                0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00
                0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc
                0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00
                0x00 0xcc 0x00 0xcc 0x00 0xcc 0x00 0xcc'
                repeatS='false' repeatT='false'/>
            </Appearance>
            <Box size='4 4 .1'/>
        </Shape>
    </Scene>
</X3D>
```

Converting image into PixelTexture

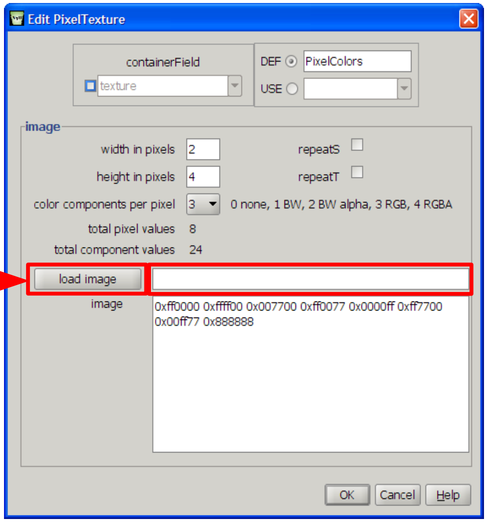
PixelTextureGenerator is a .java application to convert an image file into a PixelTexture node

- Available in Savage archive under Tools, Authoring
- https://savage.nps.edu/Savage/Tools/Authoring/PixelTextureGenerator.java
- Can be downloaded with open-source Savage archive

Command-line invocation

- `java PixelTextureGenerator imageName.ext [outputSceneName.x3d]`

X3D-Edit includes this capability within the PixelTexture editor, greatly simplifying image file conversion into pixel values



Edit PixelTexture

containerField □ texture

DEF ● PixelColors

USE ○

image

width in pixels 2

height in pixels 4

repeatS □

repeatT □

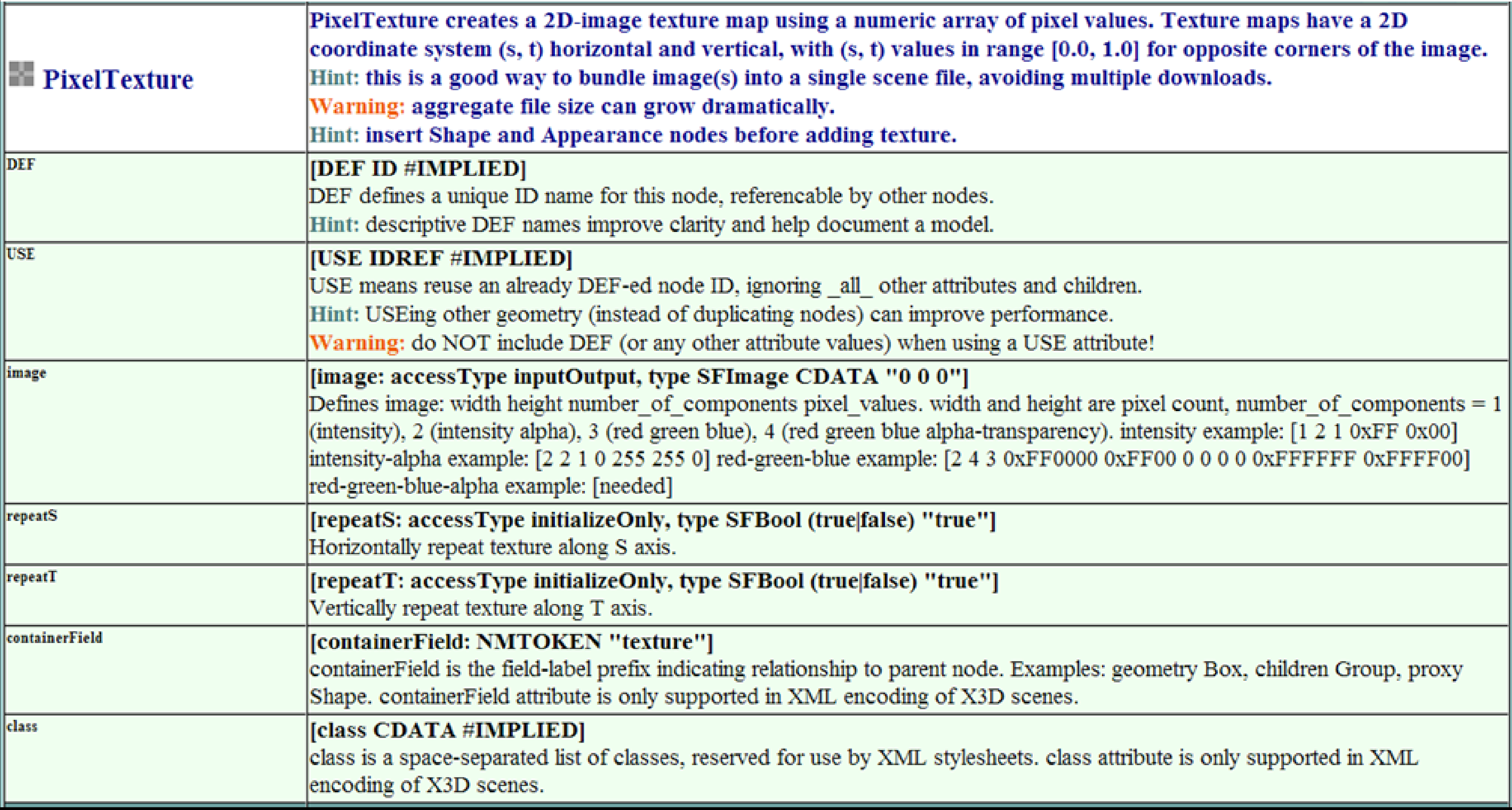
color components per pixel 3 — 0 none, 1 BW, 2 BW alpha, 3 RGB, 4 RGBA

total pixel values 8

total component values 24

image 0xff0000 0xffff00 0x007700 0xff0077 0x0000ff 0xff7700 0x00ff77 0x888888

Warning
file size increases are drastic!



PixelTexture	PixelTexture creates a 2D-image texture map using a numeric array of pixel values. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image. Hint: this is a good way to bundle image(s) into a single scene file, avoiding multiple downloads. Warning: aggregate file size can grow dramatically. Hint: insert Shape and Appearance nodes before adding texture.
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!
image	**[image: accessType inputOutput, type SFImage CDATA "0 0 0"]** Defines image: width height number_of_components pixel_values. width and height are pixel count, number_of_components = 1 (intensity), 2 (intensity alpha), 3 (red green blue), 4 (red green blue alpha-transparency). intensity example: [1 2 1 0xFF 0x00] intensity-alpha example: [2 2 1 0 255 255 0] red-green-blue example: [2 4 3 0xFF0000 0xFF00 0 0 0 0 0xFFFFFF 0xFFFF00] red-green-blue-alpha example: [needed]
repeatS	**[repeatS: accessType initializeOnly, type SFBool (true|false) "true"]** Horizontally repeat texture along S axis.
repeatT	**[repeatT: accessType initializeOnly, type SFBool (true|false) "true"]** Vertically repeat texture along T axis.
containerField	**[containerField: NMTOKEN "texture"]** 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.

TextureTransform node

TextureTransform defines a 2D *(s, t)* coordinate transformation for corresponding texture node, to better align images placed on geometry

- 2D translation left/right/up/down
- rotation angle about center
- 2D scaling, uniform or non-uniform

Transformation order remains significant

- translation, rotation, scale (same as Transform)
- However it is applied against coordinate system, **not** image file, so directions are counterintuitive

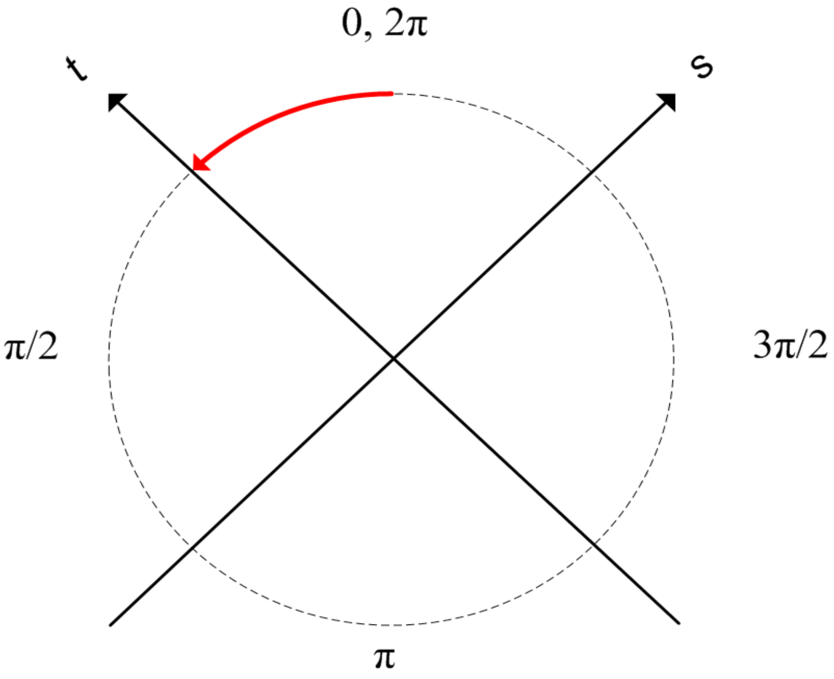
TextureTransform fields

Transformation are *(s,t)* axes-centric, not image centric, so direction differs from expectations

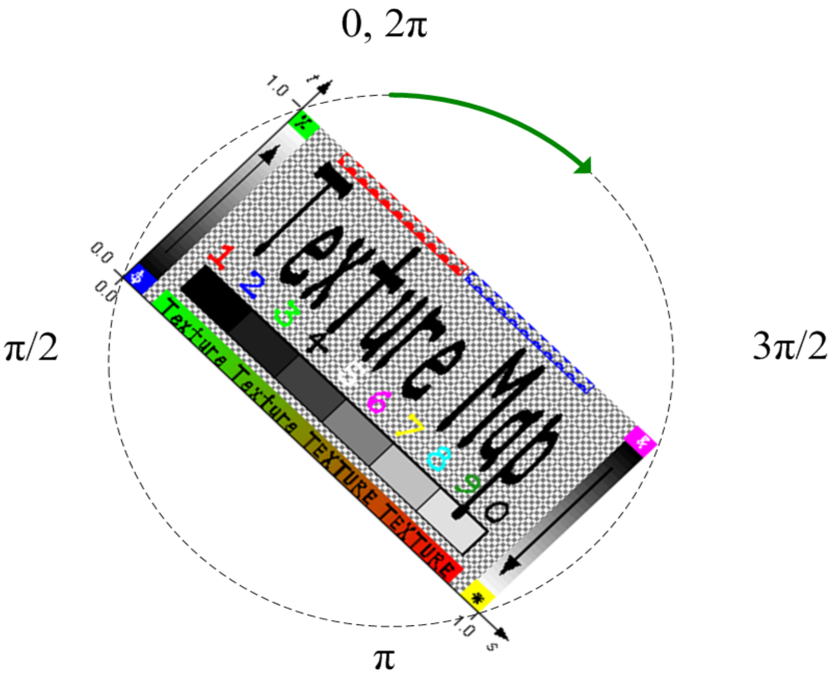
- *translation* controls lateral shift of image file along the polygonal surface, defined using *(s,t)* values
- *center* and *rotation* modify texture orientation: each makes a change in coordinate system, so the textured image rotates in opposite direction
  - *center* defined using *(s,t)* values
  - *rotation* defined using radians
- scale similarly opposite: *scale*='3 0.5' shows only 1/3 of texture along *s* axis, doubled along *t* axis

TextureTransform rotation

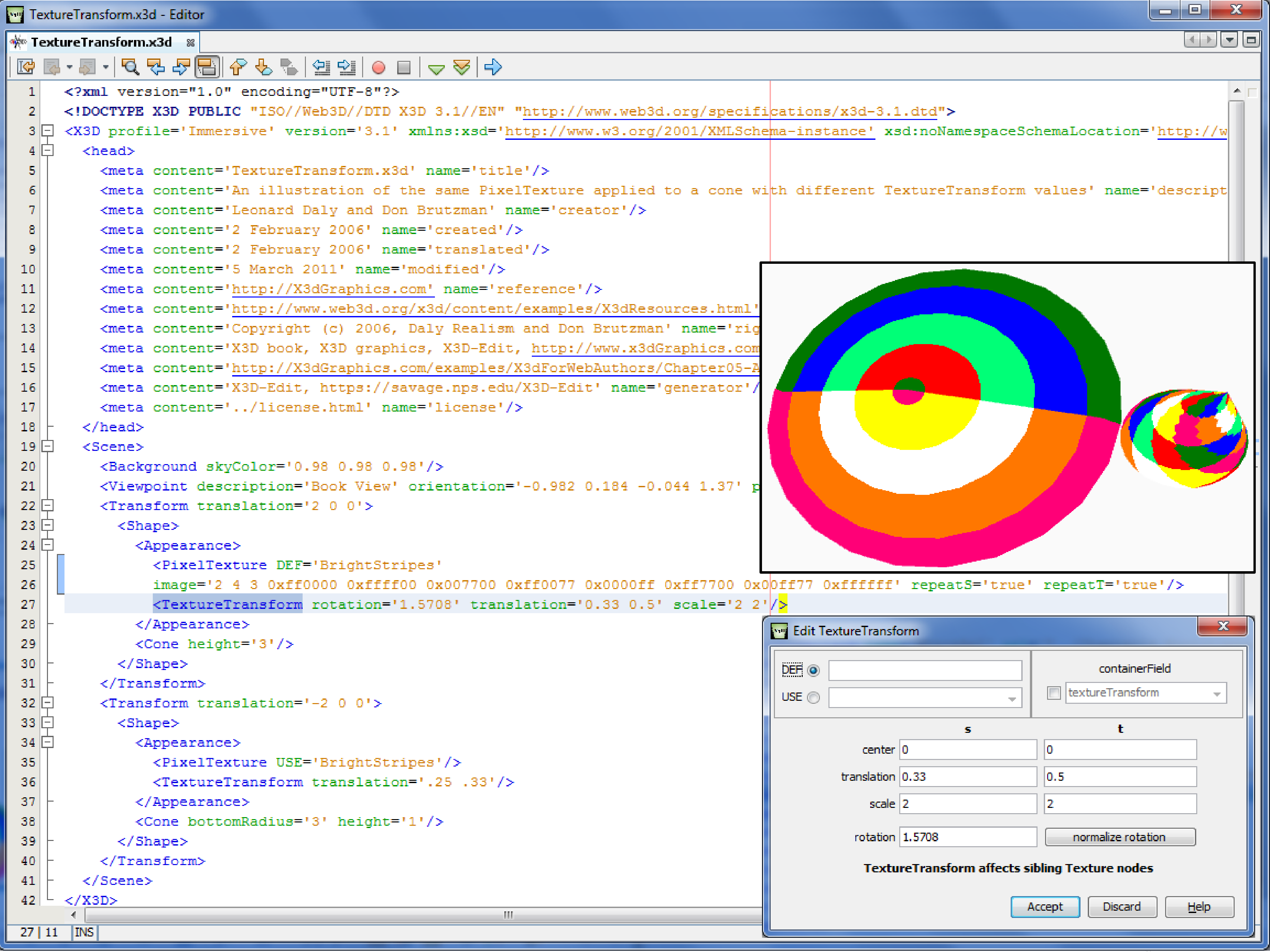
ccw rotation of
geometry texture coordinate axes



opposite rotation of
applied texture images



```
<TextureTransform rotation='0.78'/>
```



```
<?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://w
  <head>
    <meta content='TextureTransform.x3d' name='title'/>
    <meta content='An illustration of the same PixelTexture applied to a cone with different TextureTransform values' name='descript
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='5 March 2011' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/X3dResources.html'
    <meta content='Copyright (c) 2006, Daly Realism and Don Brutzman' name='rig
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-A
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='0.98 0.98 0.98'/>
    <Viewpoint description='Book View' orientation='-0.982 0.184 -0.044 1.37' p
    <Transform translation='2 0 0'>
      <Shape>
        <Appearance>
          <PixelTexture DEF='BrightStripes'
          image='2 4 3 0xff0000 0xffff00 0x007700 0xff0077 0x0000ff 0xff7700 0x00ff77 0xffffff' repeatS='true' repeatT='true'/>
          <TextureTransform rotation='1.5708' translation='0.33 0.5' scale='2 2'/>
        </Appearance>
        <Cone height='3'/>
      </Shape>
    </Transform>
    <Transform translation='-2 0 0'>
      <Shape>
        <Appearance>
          <PixelTexture USE='BrightStripes'/>
          <TextureTransform translation='.25 .33'/>
        </Appearance>
        <Cone bottomRadius='3' height='1'/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

27 | 11 INS

Edit TextureTransform

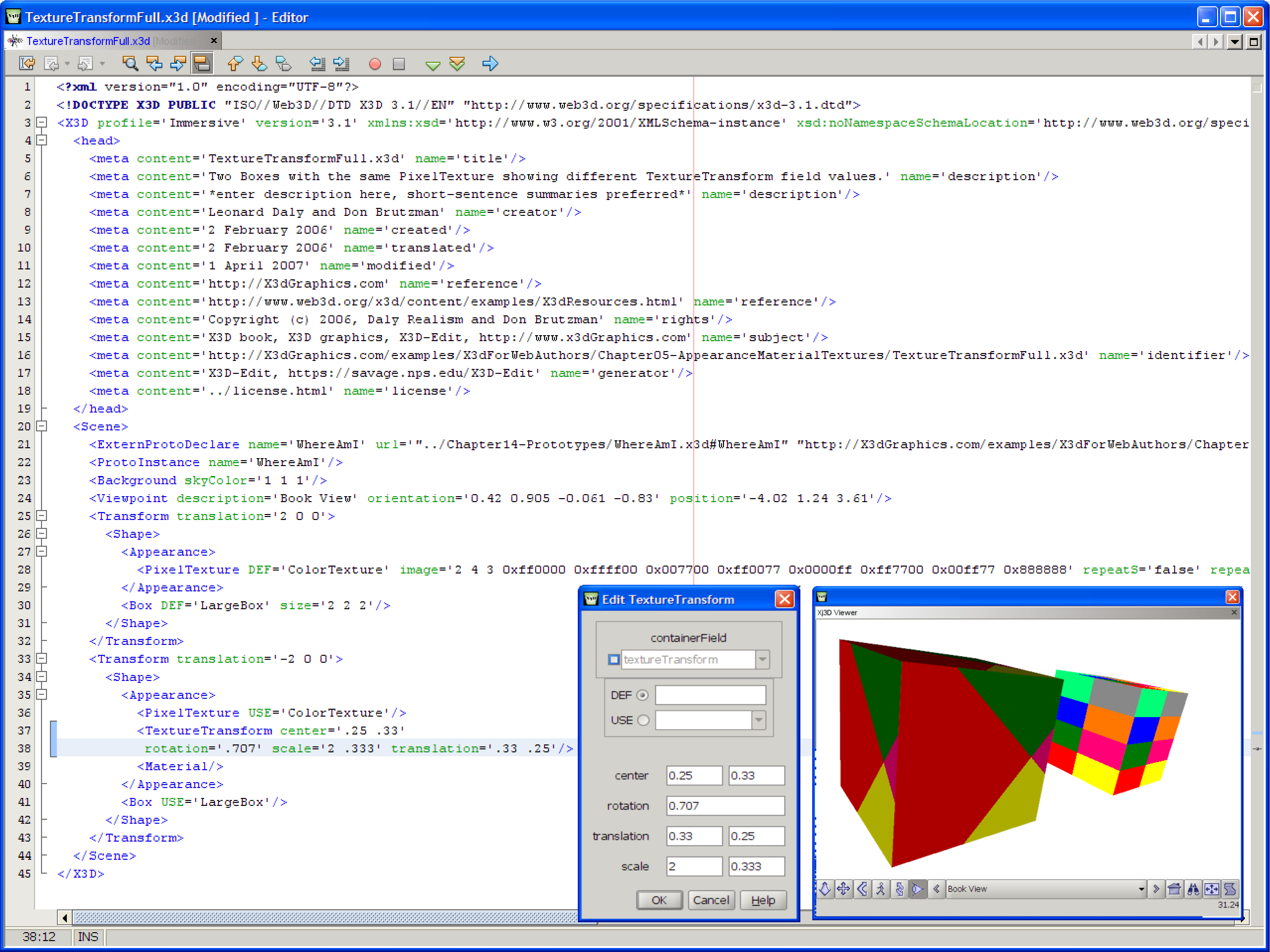
DEF ●

USE ○

containerField ☐ textureTransform

	s	t
center	0	0
translation	0.33	0.5
scale	2	2
rotation	1.5708	

TextureTransform affects sibling Texture nodes



TextureTransformFull.x3d [Modified] - Editor
TextureTransformFull.x3d [Modified]

```
<?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/speci
  <head>
    <meta content='TextureTransformFull.x3d' name='title'/>
    <meta content='Two Boxes with the same PixelTexture showing different TextureTransform field values.' name='description'/>
    <meta content='*enter description here, short-sentence summaries preferred*' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='1 April 2007' 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 (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/Chapter05-AppearanceMaterialTextures/TextureTransformFull.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <ExternProtoDeclare name='WhereAmI' url='"../Chapter14-Prototypes/WhereAmI.x3d#WhereAmI" "http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter
    <ProtoInstance name='WhereAmI'/>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book View' orientation='0.42 0.905 -0.061 -0.83' position='-4.02 1.24 3.61'/>
    <Transform translation='2 0 0'>
      <Shape>
        <Appearance>
          <PixelTexture DEF='ColorTexture' image='2 4 3 0xff0000 0xffff00 0x007700 0xff0077 0x0000ff 0xff7700 0x00ff77 0x888888' repeatS='false' repea
        </Appearance>
        <Box DEF='LargeBox' size='2 2 2'/>
      </Shape>
    </Transform>
    <Transform translation='-2 0 0'>
      <Shape>
        <Appearance>
          <PixelTexture USE='ColorTexture'/>
          <TextureTransform center='.25 .33'
           rotation='.707' scale='2 .333' translation='.33 .25'/>
          <Material/>
        </Appearance>
        <Box USE='LargeBox'/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Edit TextureTransform

containerField: ☐ textureTransform

DEF ○

USE ○

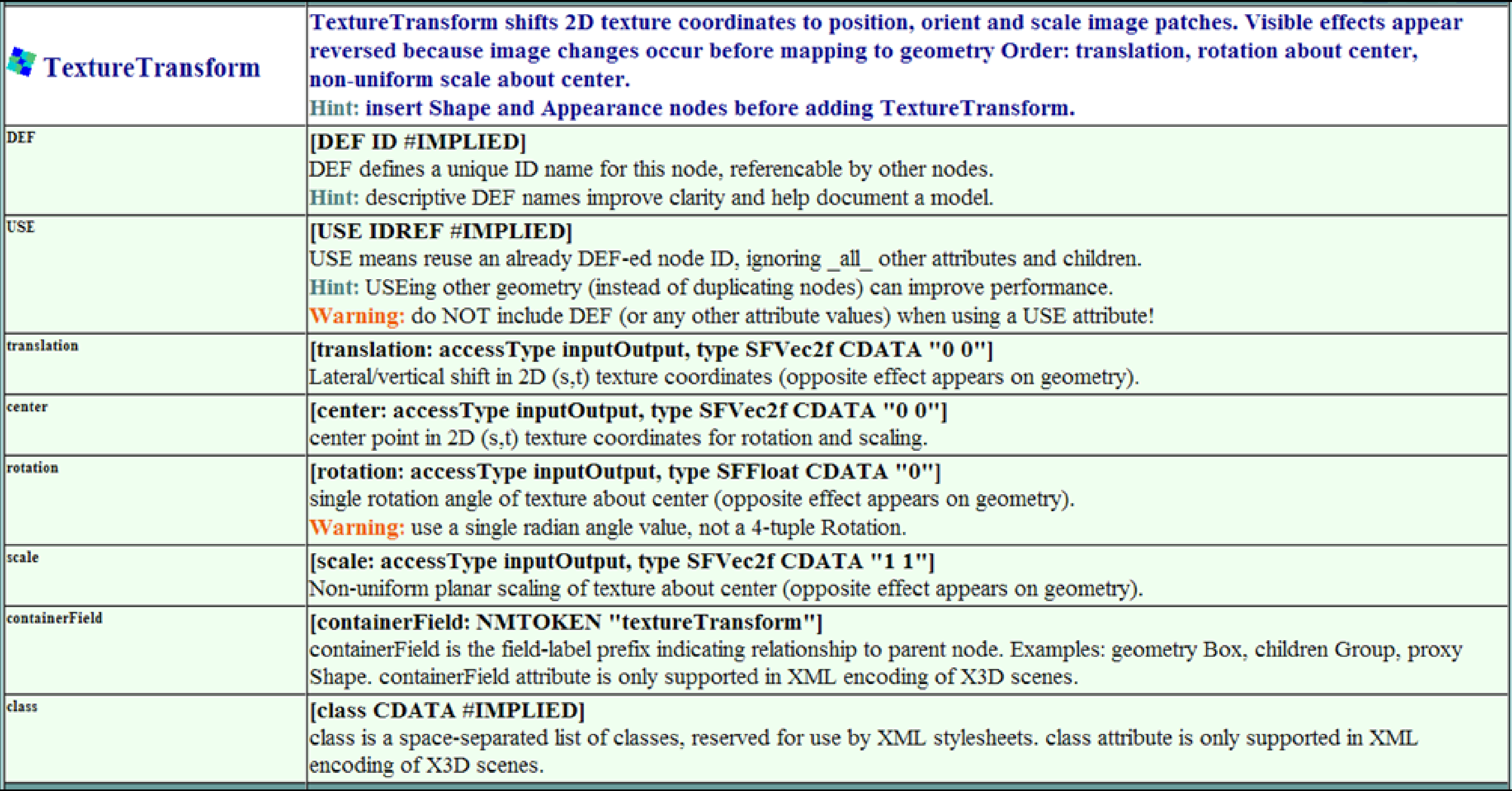
center: 0.25 0.33

rotation: 0.707

translation: 0.33 0.25

scale: 2 0.333

38:12 INS



TextureTransform	**TextureTransform shifts 2D texture coordinates to position, orient and scale image patches. Visible effects appear reversed because image changes occur before mapping to geometry Order: translation, rotation about center, non-uniform scale about center.** **Hint: insert Shape and Appearance nodes before adding TextureTransform.**
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 SFVec2f CDATA "0 0"]** Lateral/vertical shift in 2D (s,t) texture coordinates (opposite effect appears on geometry).
center	**[center: accessType inputOutput, type SFVec2f CDATA "0 0"]** center point in 2D (s,t) texture coordinates for rotation and scaling.
rotation	**[rotation: accessType inputOutput, type SFFloat CDATA "0"]** single rotation angle of texture about center (opposite effect appears on geometry). Warning: use a single radian angle value, not a 4-tuple Rotation.
scale	**[scale: accessType inputOutput, type SFVec2f CDATA "1 1"]** Non-uniform planar scaling of texture about center (opposite effect appears on geometry).
containerField	**[containerField: NMTOKEN "textureTransform"]** 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.

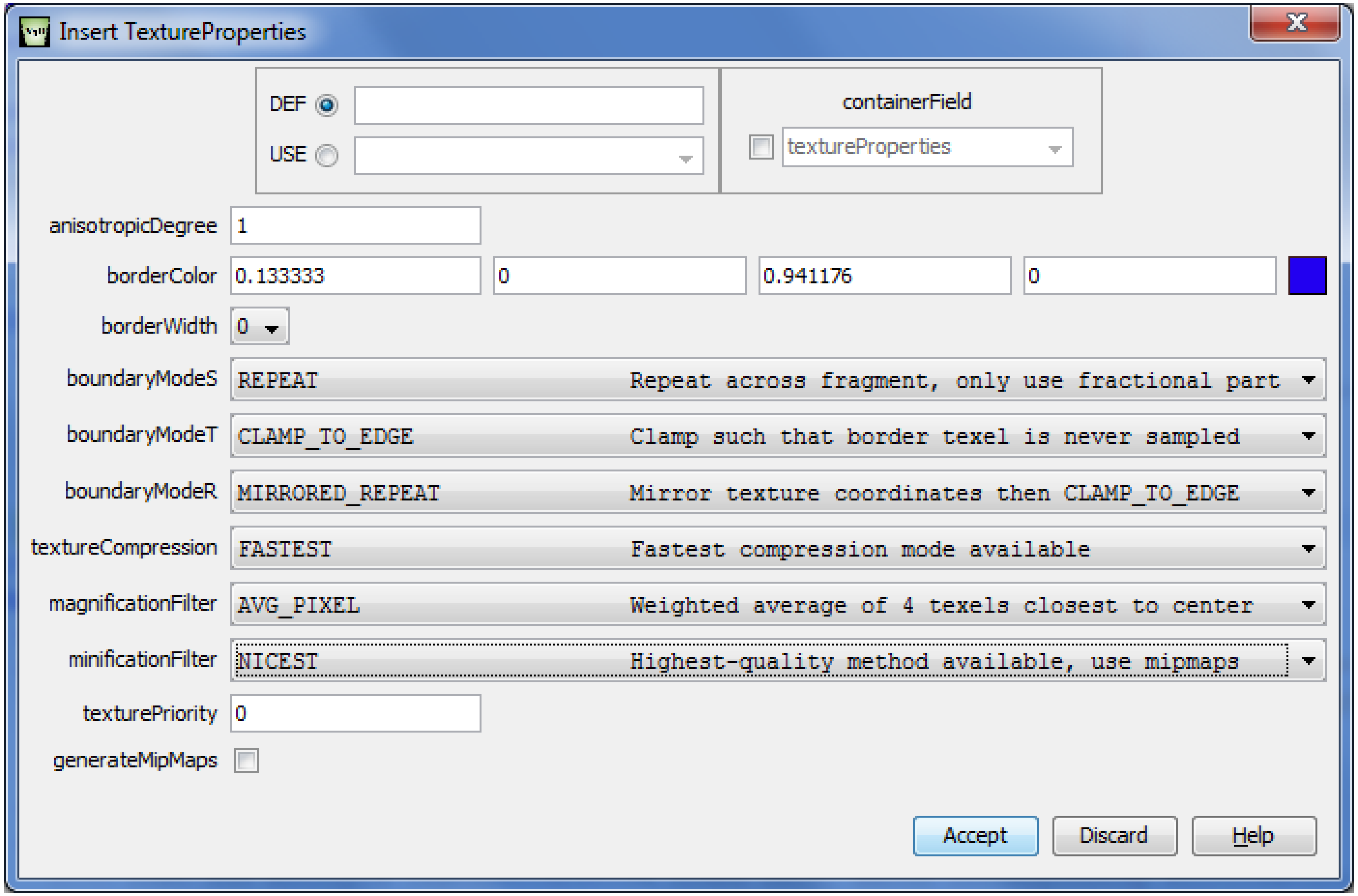
TextureProperties node

TextureProperties specifies additional properties applicable to all line geometry

- *anisotropicDegree* texture filtering
- *borderColor* SFColorRGBA 3-tuple color plus alpha
- *borderWidth* pixel boundary [0..1]
- *boundaryModeS*, *boundaryModeT*, *boundaryModeR*: CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT
- Others: *magnificationFilter*, *minificationFilter*, *textureCompression*, *texturePriority*, *generateMipMaps*
- Hint: include

```
<component name='Shape' level='2'/>
```

TextureProperties interface



DEF ●

USE ○

containerField ☐ textureProperties

anisotropicDegree 1

borderColor 0.133333 0 0.941176 0 🟦

borderWidth 0

boundaryModeS REPEAT Repeat across fragment, only use fractional part

boundaryModeT CLAMP_TO_EDGE Clamp such that border texel is never sampled

boundaryModeR MIRRORED_REPEAT Mirror texture coordinates then CLAMP_TO_EDGE

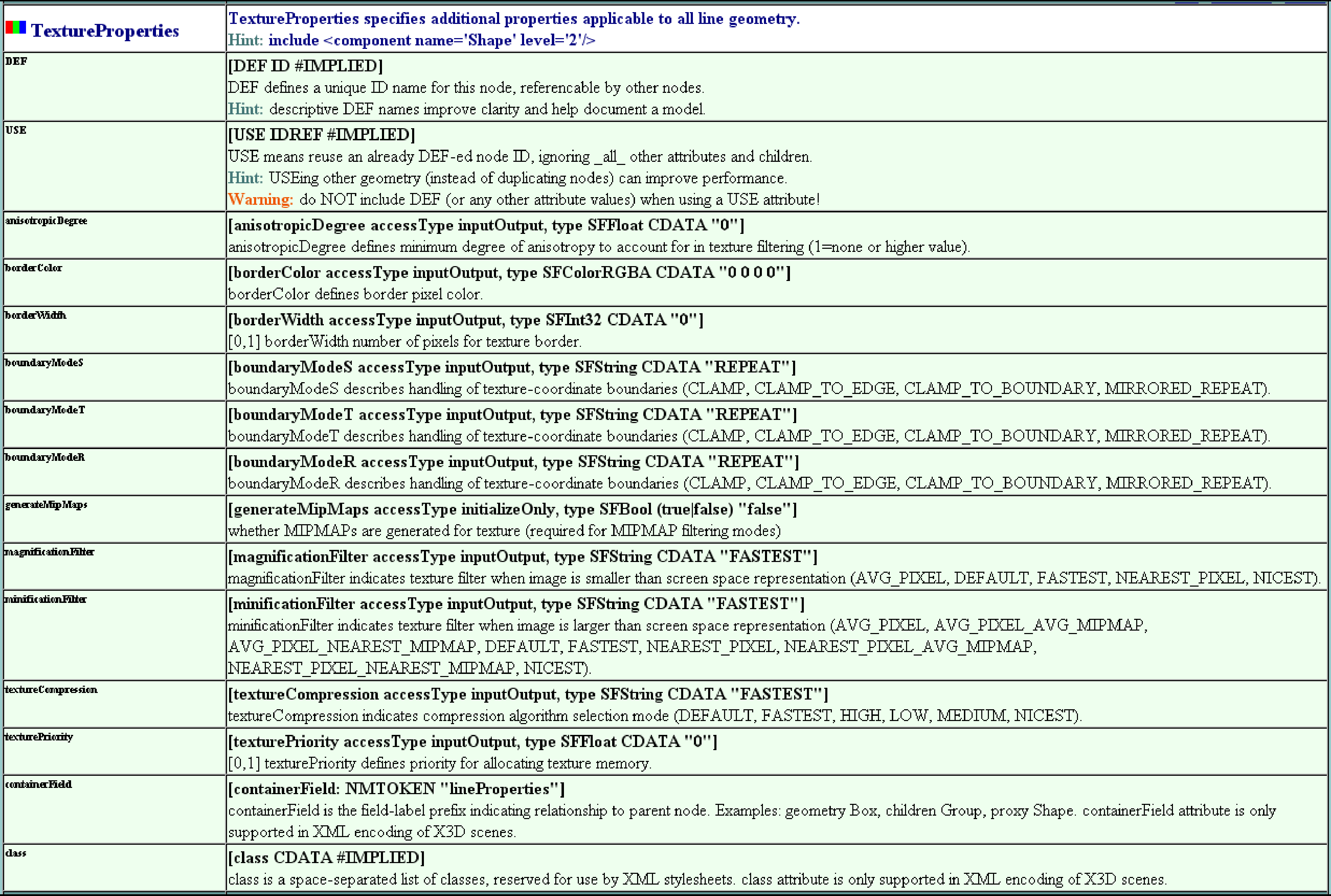
textureCompression FASTEST Fastest compression mode available

magnificationFilter AVG_PIXEL Weighted average of 4 texels closest to center

minificationFilter NICEST Highest-quality method available, use mipmaps

texturePriority 0

generateMipMaps ☐



TextureProperties	TextureProperties specifies additional properties applicable to all line geometry. Hint: include <component name='Shape' level='2'/>
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!
anisotropicDegree	**[anisotropicDegree accessType inputOutput, type SFFloat CDATA "0"]** anisotropicDegree defines minimum degree of anisotropy to account for in texture filtering (1=none or higher value).
borderColor	**[borderColor accessType inputOutput, type SFColorRGBA CDATA "0 0 0 0"]** borderColor defines border pixel color.
borderWidth	**[borderWidth accessType inputOutput, type SFInt32 CDATA "0"]** [0,1] borderWidth number of pixels for texture border.
boundaryModeS	**[boundaryModeS accessType inputOutput, type SFString CDATA "REPEAT"]** boundaryModeS describes handling of texture-coordinate boundaries (CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT).
boundaryModeT	**[boundaryModeT accessType inputOutput, type SFString CDATA "REPEAT"]** boundaryModeT describes handling of texture-coordinate boundaries (CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT).
boundaryModeR	**[boundaryModeR accessType inputOutput, type SFString CDATA "REPEAT"]** boundaryModeR describes handling of texture-coordinate boundaries (CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT).
generateMipMaps	**[generateMipMaps accessType initializeOnly, type SFBool (true|false) "false"]** whether MIPMAPs are generated for texture (required for MIPMAP filtering modes)
magnificationFilter	**[magnificationFilter accessType inputOutput, type SFString CDATA "FASTEST"]** magnificationFilter indicates texture filter when image is smaller than screen space representation (AVG_PIXEL, DEFAULT, FASTEST, NEAREST_PIXEL, NICEST).
minificationFilter	**[minificationFilter accessType inputOutput, type SFString CDATA "FASTEST"]** minificationFilter indicates texture filter when image is larger than screen space representation (AVG_PIXEL, AVG_PIXEL_AVG_MIPMAP, AVG_PIXEL_NEAREST_MIPMAP, DEFAULT, FASTEST, NEAREST_PIXEL, NEAREST_PIXEL_AVG_MIPMAP, NEAREST_PIXEL_NEAREST_MIPMAP, NICEST).
textureCompression	**[textureCompression accessType inputOutput, type SFString CDATA "FASTEST"]** textureCompression indicates compression algorithm selection mode (DEFAULT, FASTEST, HIGH, LOW, MEDIUM, NICEST).
texturePriority	**[texturePriority accessType inputOutput, type SFFloat CDATA "0"]** [0,1] texturePriority defines priority for allocating texture memory.
containerField	**[containerField: NMTOKEN "lineProperties"]** 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.

TextureCoordinate node

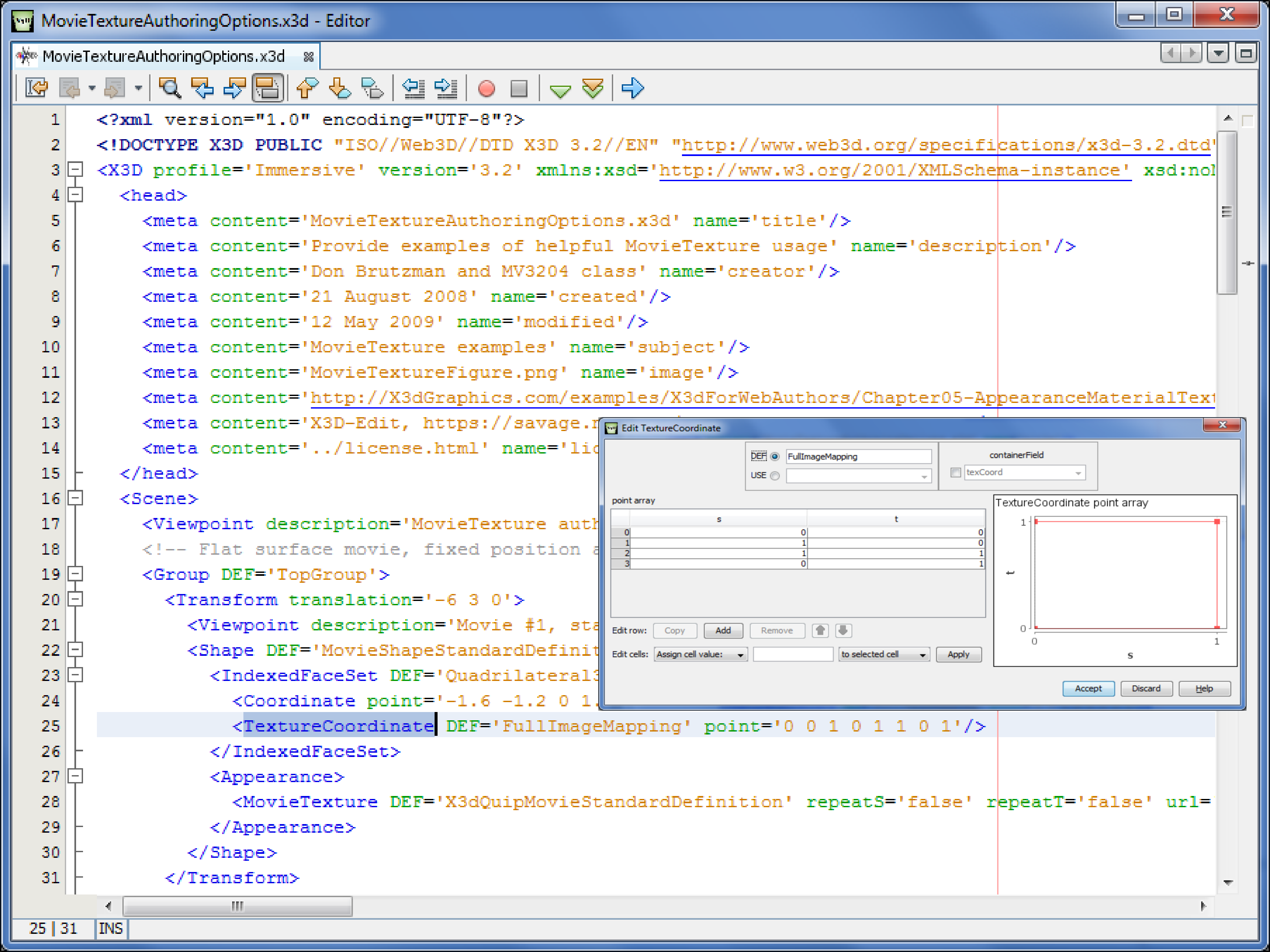
TextureCoordinate specifies a set of 2D texture coordinates used by vertex-based nodes

- Such as IndexedFaceSet and ElevationGrid, which are covered in Chapter 6

TextureCoordinate *point* field has *(s,t)* values corresponding to vertices in parent geometry

- Type MFVec2f, multiple field array of 2-tuple floats
- Default is empty array, corresponds to regular *(s,t)* values ranging (0,1)

Best approach: use special authoring tools



MovieTextureAuthoringOptions.x3d - Editor

MovieTextureAuthoringOptions.x3d

```
<?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:nol
  <head>
    <meta content='MovieTextureAuthoringOptions.x3d' name='title'/>
    <meta content='Provide examples of helpful MovieTexture usage' name='description'/>
    <meta content='Don Brutzman and MV3204 class' name='creator'/>
    <meta content='21 August 2008' name='created'/>
    <meta content='12 May 2009' name='modified'/>
    <meta content='MovieTexture examples' name='subject'/>
    <meta content='MovieTextureFigure.png' name='image'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialText
    <meta content='X3D-Edit, https://savage.n
    <meta content='../license.html' name='lic
  </head>
  <Scene>
    <Viewpoint description='MovieTexture auth
    <!-- Flat surface movie, fixed position a
    <Group DEF='TopGroup'>
      <Transform translation='-6 3 0'>
        <Viewpoint description='Movie #1, sta
        <Shape DEF='MovieShapeStandardDefinit
          <IndexedFaceSet DEF='Quadrilateral3
            <Coordinate point='-1.6 -1.2 0 1.
            <TextureCoordinate DEF='FullImageMapping' point='0 0 1 0 1 1 0 1'/>
          </IndexedFaceSet>
          <Appearance>
            <MovieTexture DEF='X3dQuipMovieStandardDefinition' repeatS='false' repeatT='false' url=
          </Appearance>
        </Shape>
      </Transform>
```

Edit TextureCoordinate

DEF ● FullImageMapping

USE ○

containerField

☐ texCoord

point array

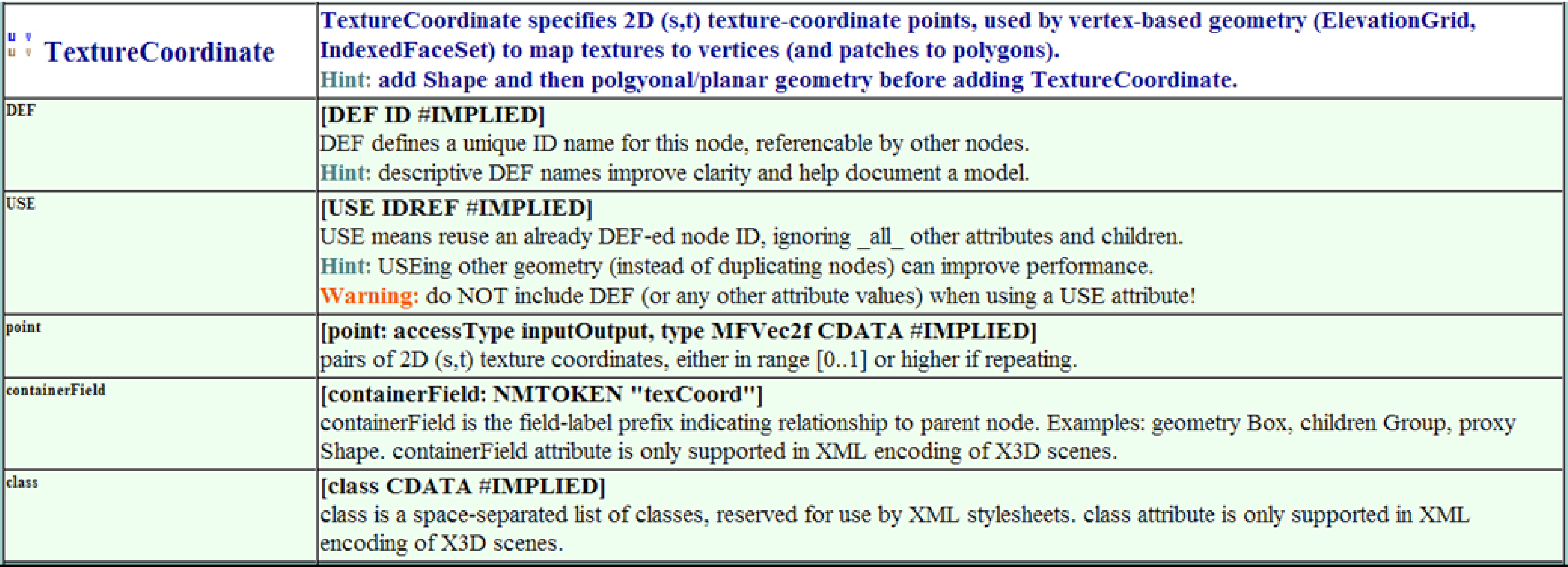
	s	t
0	0	0
1	1	0
2	1	1
3	0	1

Edit row: Copy Add Remove

Edit cells: Assign cell value: to selected cell Apply

Accept Discard Help

25 | 31 INS



TextureCoordinate	**TextureCoordinate specifies 2D (s,t) texture-coordinate points, used by vertex-based geometry (ElevationGrid, IndexedFaceSet) to map textures to vertices (and patches to polygons).** **Hint: add Shape and then polgyonal/planar geometry before adding TextureCoordinate.**
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!
point	**[point: accessType inputOutput, type MFVec2f CDATA #IMPLIED]** pairs of 2D (s,t) texture coordinates, either in range [0..1] or higher if repeating.
containerField	**[containerField: NMTOKEN "texCoord"]** 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.

TextureCoordinateGenerator node

TextureCoordinateGenerator enables the automatic computation and generation of texture coordinates for geometric shapes

- Can serve as substitute for TextureCoordinate node

Eleven procedural modes are provided

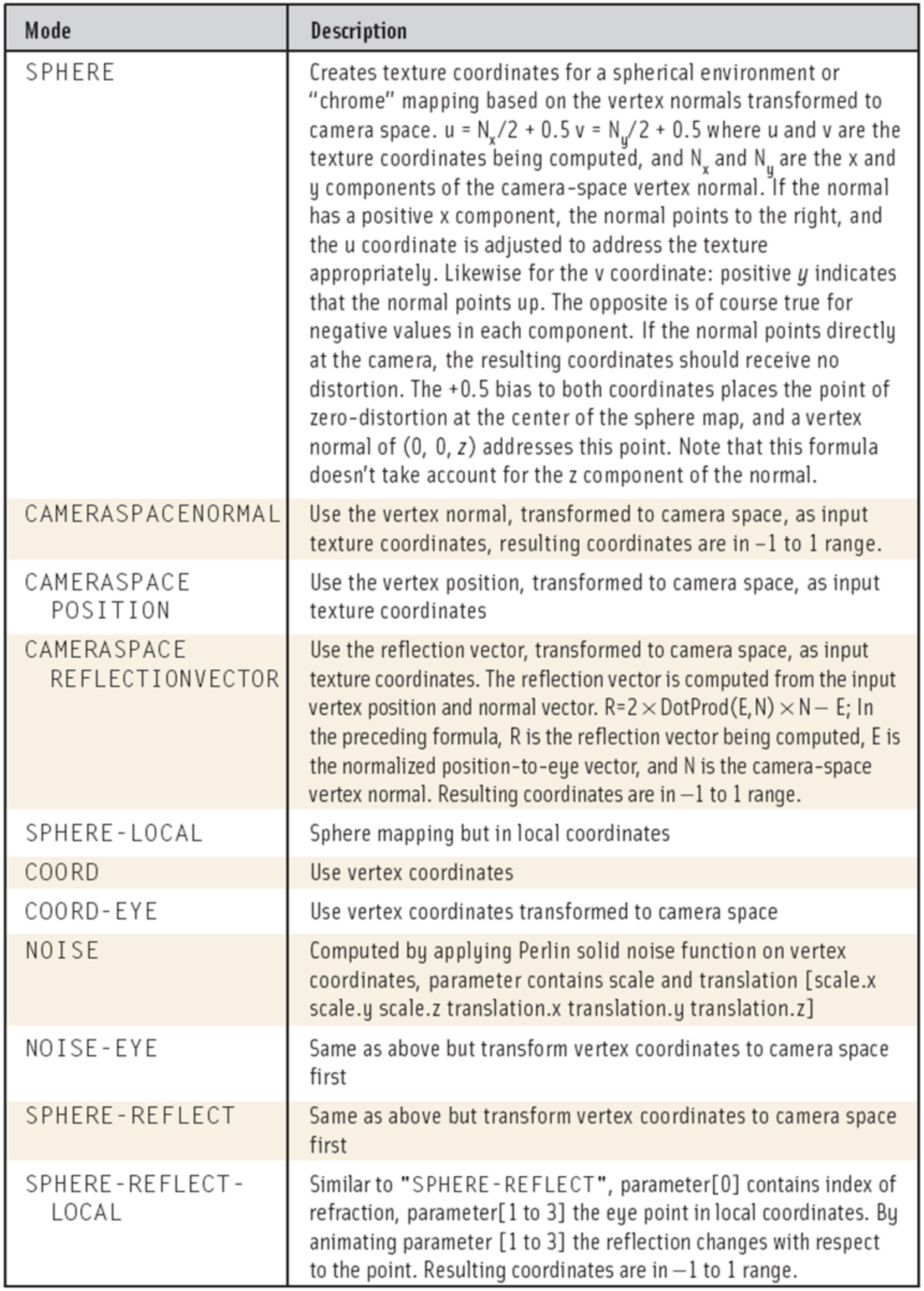
- *mode* field, following table explains possible values
- Associated *parameter* field provides setup values

This node is quite complicated

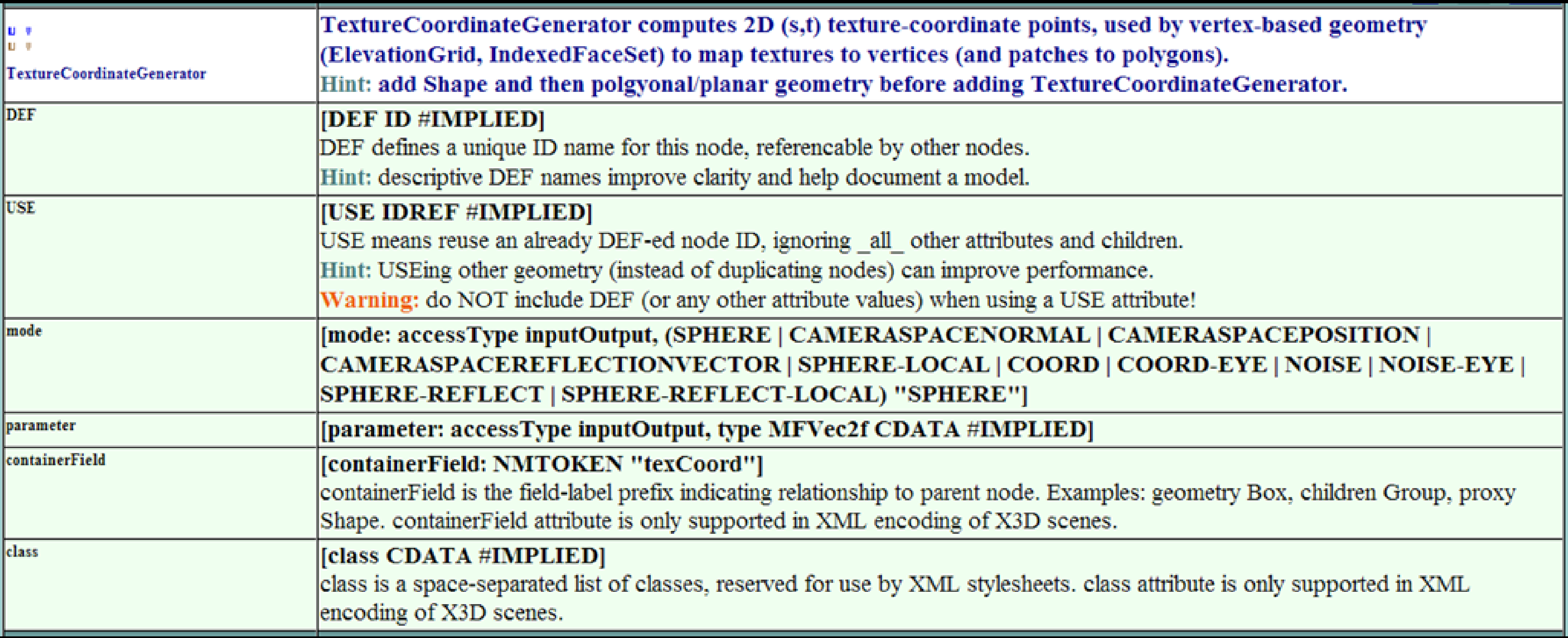
- May find support in 3D acceleration hardware

Best approach: use special authoring tools

TextureCoordinateGenerator *mode* enumerations and *parameter* values



Mode	Description
SPHERE	Creates texture coordinates for a spherical environment or "chrome" mapping based on the vertex normals transformed to camera space. $u = N_x/2 + 0.5$ $v = N_y/2 + 0.5$ where u and v are the texture coordinates being computed, and N_x and N_y are the x and y components of the camera-space vertex normal. If the normal has a positive x component, the normal points to the right, and the u coordinate is adjusted to address the texture appropriately. Likewise for the v coordinate: positive *y* indicates that the normal points up. The opposite is of course true for negative values in each component. If the normal points directly at the camera, the resulting coordinates should receive no distortion. The +0.5 bias to both coordinates places the point of zero-distortion at the center of the sphere map, and a vertex normal of (0, 0, *z*) addresses this point. Note that this formula doesn't take account for the z component of the normal.
CAMERASPACENORMAL	Use the vertex normal, transformed to camera space, as input texture coordinates, resulting coordinates are in −1 to 1 range.
CAMERASPACE POSITION	Use the vertex position, transformed to camera space, as input texture coordinates
CAMERASPACE REFLECTIONVECTOR	Use the reflection vector, transformed to camera space, as input texture coordinates. The reflection vector is computed from the input vertex position and normal vector. $R=2 \times \text{DotProd}(E,N) \times N - E$; In the preceding formula, R is the reflection vector being computed, E is the normalized position-to-eye vector, and N is the camera-space vertex normal. Resulting coordinates are in −1 to 1 range.
SPHERE-LOCAL	Sphere mapping but in local coordinates
COORD	Use vertex coordinates
COORD-EYE	Use vertex coordinates transformed to camera space
NOISE	Computed by applying Perlin solid noise function on vertex coordinates, parameter contains scale and translation [scale.x scale.y scale.z translation.x translation.y translation.z]
NOISE-EYE	Same as above but transform vertex coordinates to camera space first
SPHERE-REFLECT	Same as above but transform vertex coordinates to camera space first
SPHERE-REFLECT-LOCAL	Similar to "SPHERE-REFLECT", parameter[0] contains index of refraction, parameter[1 to 3] the eye point in local coordinates. By animating parameter [1 to 3] the reflection changes with respect to the point. Resulting coordinates are in −1 to 1 range.



TextureCoordinateGenerator	**TextureCoordinateGenerator computes 2D (s,t) texture-coordinate points, used by vertex-based geometry (ElevationGrid, IndexedFaceSet) to map textures to vertices (and patches to polygons).** **Hint: add Shape and then polgyonal/planar geometry before adding TextureCoordinateGenerator.**
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!
mode	**[mode: accessType inputOutput, (SPHERE | CAMERASPACENORMAL | CAMERASPACEPOSITION | CAMERASPACEREFLECTIONVECTOR | SPHERE-LOCAL | COORD | COORD-EYE | NOISE | NOISE-EYE | SPHERE-REFLECT | SPHERE-REFLECT-LOCAL) "SPHERE"]**
parameter	**[parameter: accessType inputOutput, type MFVec2f CDATA #IMPLIED]**
containerField	**[containerField: NMTOKEN "texCoord"]** 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.

back to Table of Contents

Advanced Nodes: MultiTextures

MultiTexture node

MultiTexture applies several textures together to achieve more complex visual effects

MultiTexture contains multiple ImageTexture, MovieTexture and PixelTexture nodes

- Texture maps have a 2D coordinate system
- (s, t) horizontal and vertical
- (s, t) values [0.0, 1.0] at opposite corners of image

MultiTextureTransform is associated sibling node

MultiTextureCoordinate is child of associated polygonal geometry node

MultiTexture fields

MultiTexture fields define how each image in series of texture children are composed and applied

- *mode, function, source* are matching MFString arrays

mode enumerations indicate type of blending operation, both for color and for alpha channel

- Numerous enumeration values, default MODULATE

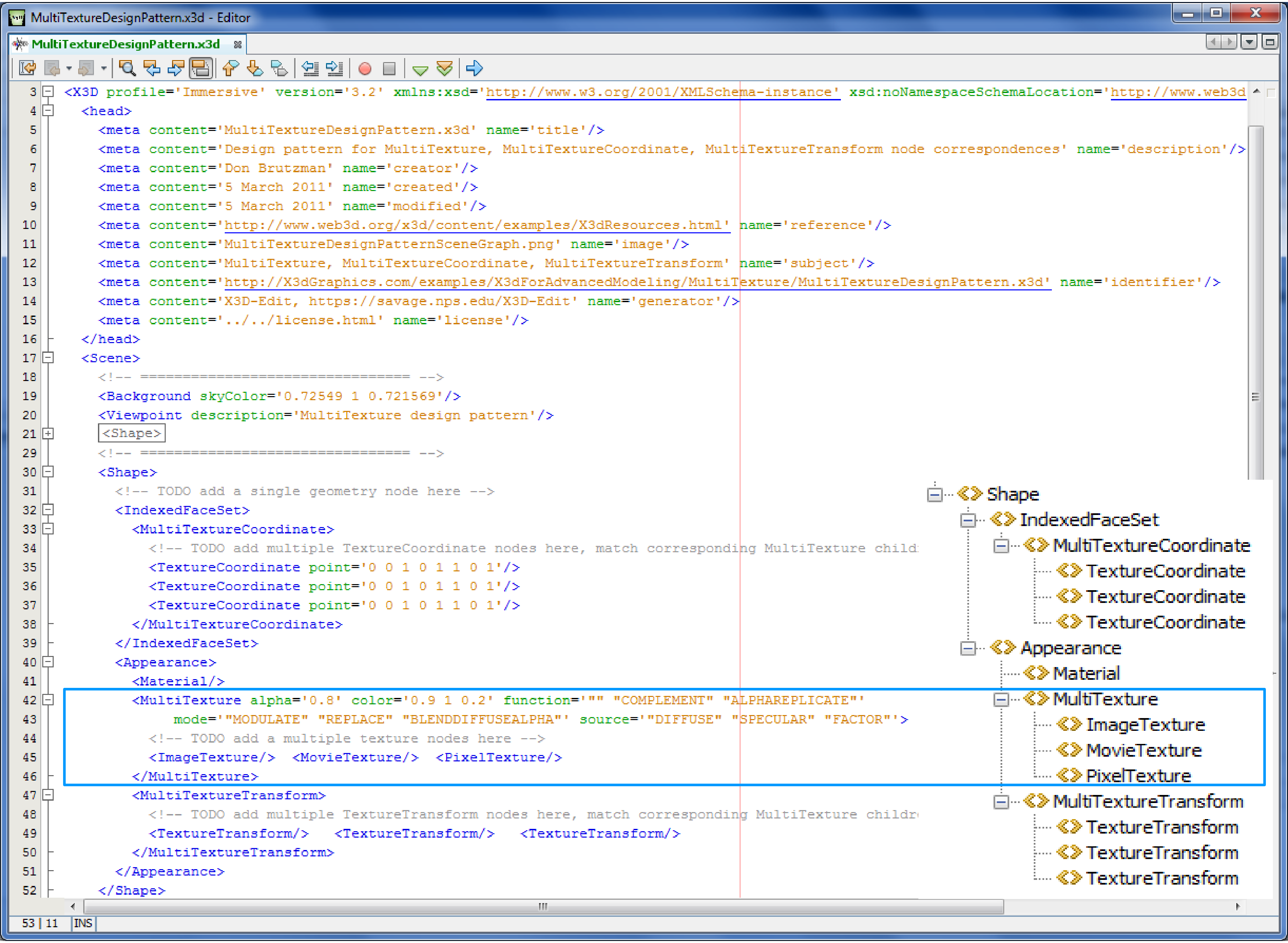
function operators COMPLEMENT, ALPHAREPLICATE optionally applied after *mode* blending operation

source determines if image source pixels are treated as DIFFUSE, SPECULAR or multiplicative FACTOR

alpha, color define baseline (1-transparency), RGB

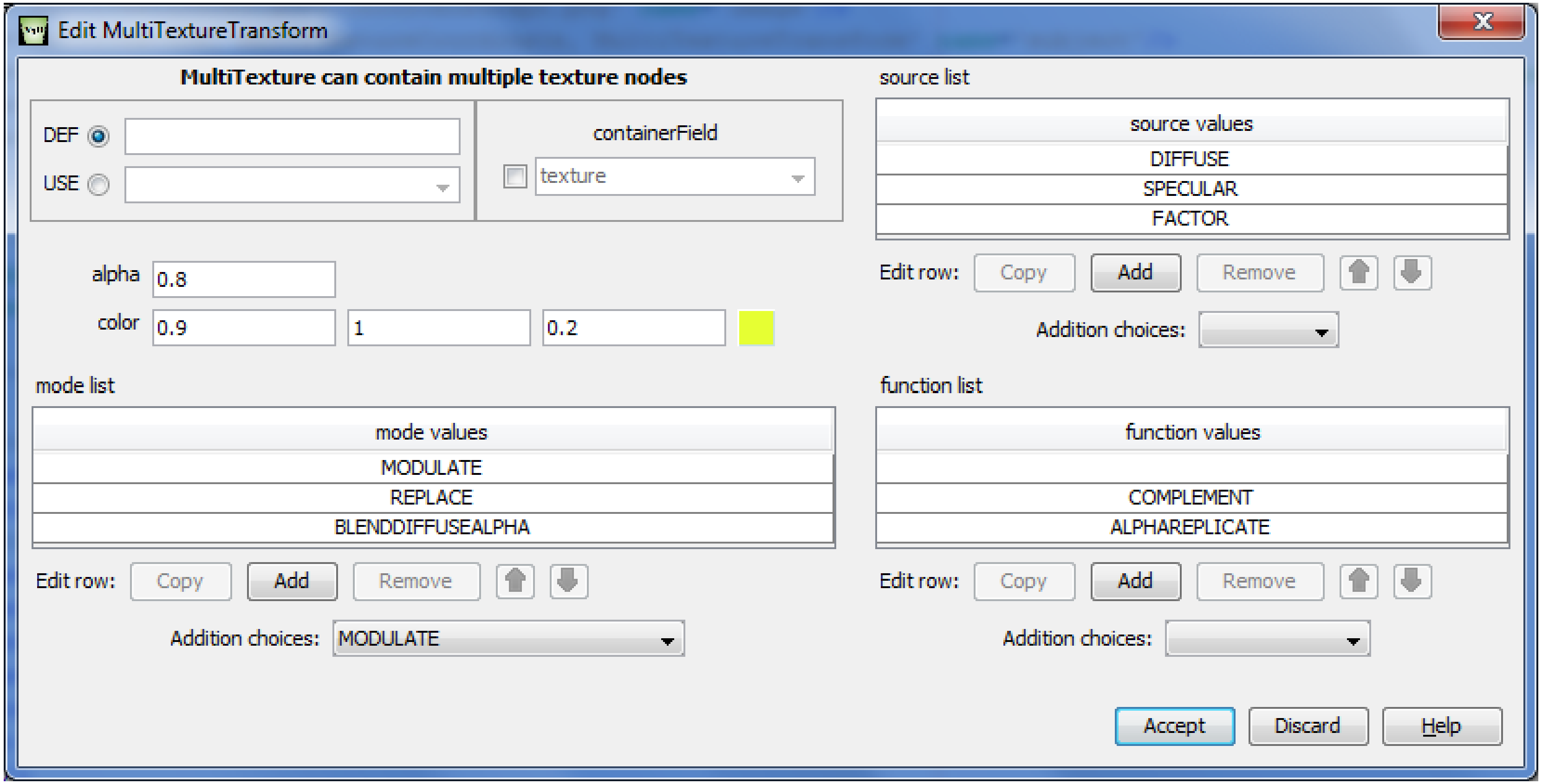
MultiTexture *mode* enumeration values

"MODULATE"	Multiply texture color with current color, Arg1 × Arg2
"REPLACE"	Replace current color, Arg2
"MODULATE2X"	Multiply components of arguments, shift products left 1 bit (multiplying by 2) for brightening
"MODULATE4X"	Multiply components of arguments, shift products left 2 bits (multiplying by 4) for brightening
"ADD"	Add the components of the arguments, Arg1 + Arg2
"ADDSIGNED"	Add components of arguments with -0.5 bias, effective range becomes -0.5 through 0.5
"ADDSIGNED2X"	Add components of arguments with -0.5 bias, shift products to left 1 bit
"SUBTRACT"	Subtract components of second argument from first argument, Arg1 - Arg2
"ADDSMOOTH"	Add first and second arguments, then subtract product from sum. Arg1 + Arg2 - Arg1 × Arg2 = Arg1 + (1 - Arg1) × Arg2
"BLENDDIFFUSEALPHA"	Linearly blend this texture stage using interpolated alpha from each vertex, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"BLENDTEXTUREALPHA"	Linearly blend this texture stage using alpha from this stage's texture, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"BLENDFACTORALPHA"	Linearly blend this texture stage using alpha factor from MultiTexture node, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"BLENDCURRENTALPHA"	Linearly blend this texture stage using alpha taken from previous texture stage, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"MODULATEALPHA_ADDCOLOR"	Modulate color of second argument using alpha of first argument, then add result to argument one, Arg1.RGB + Arg1.A × Arg2.RGB
"MODULATEINVALPHA_ADDCOLOR"	Similar to MODULATEALPHA_ADDCOLOR but use inverse of alpha of first argument, (1 - Arg1.A) × Arg2.RGB + Arg1.RGB
"MODULATEINVCOLOR_ADDALPHA"	Similar to MODULATECOLOR_ADDALPHA but use inverse of color of first argument, (1 - Arg1.RGB) × Arg2.RGB + Arg1.A
"OFF"	No texture composition for this stage
"SELECTARG1"	Use color argument 1, Arg1
"SELECTARG2"	Use color argument 1, Arg2
"DOTPRODUCT3"	Modulate components of each argument (as signed components), add their products, then replicate sum to all color channels, including alpha



```
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d
  <head>
    <meta content='MultiTextureDesignPattern.x3d' name='title'/>
    <meta content='Design pattern for MultiTexture, MultiTextureCoordinate, MultiTextureTransform node correspondences' name='description'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='5 March 2011' name='created'/>
    <meta content='5 March 2011' name='modified'/>
    <meta content='http://www.web3d.org/x3d/content/examples/X3dResources.html' name='reference'/>
    <meta content='MultiTextureDesignPatternSceneGraph.png' name='image'/>
    <meta content='MultiTexture, MultiTextureCoordinate, MultiTextureTransform' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForAdvancedModeling/MultiTexture/MultiTextureDesignPattern.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='0.72549 1 0.721569'/>
    <Viewpoint description='MultiTexture design pattern'/>
    <Shape>
    <!-- ================================ -->
    <Shape>
      <!-- TODO add a single geometry node here -->
      <IndexedFaceSet>
        <MultiTextureCoordinate>
          <!-- TODO add multiple TextureCoordinate nodes here, match corresponding MultiTexture child
          <TextureCoordinate point='0 0 1 0 1 1 0 1'/>
          <TextureCoordinate point='0 0 1 0 1 1 0 1'/>
          <TextureCoordinate point='0 0 1 0 1 1 0 1'/>
        </MultiTextureCoordinate>
      </IndexedFaceSet>
      <Appearance>
        <Material/>
        <MultiTexture alpha='0.8' color='0.9 1 0.2' function='"" "COMPLEMENT" "ALPHAREPLICATE"'
             mode='"MODULATE" "REPLACE" "BLENDDIFFUSEALPHA"' source='"DIFFUSE" "SPECULAR" "FACTOR"'>
          <!-- TODO add a multiple texture nodes here -->
          <ImageTexture/>  <MovieTexture/>  <PixelTexture/>
        </MultiTexture>
        <MultiTextureTransform>
          <!-- TODO add multiple TextureTransform nodes here, match corresponding MultiTexture childr
          <TextureTransform/>  <TextureTransform/>  <TextureTransform/>
        </MultiTextureTransform>
      </Appearance>
    </Shape>
```

MultiTexture editing panel



Edit MultiTextureTransform

MultiTexture can contain multiple texture nodes

DEF ●

USE ○

containerField

☐ texture

alpha 0.8

color 0.9 1 0.2 🟨

mode list

mode values
MODULATE
REPLACE
BLENDDIFFUSEALPHA

Addition choices: MODULATE

source list

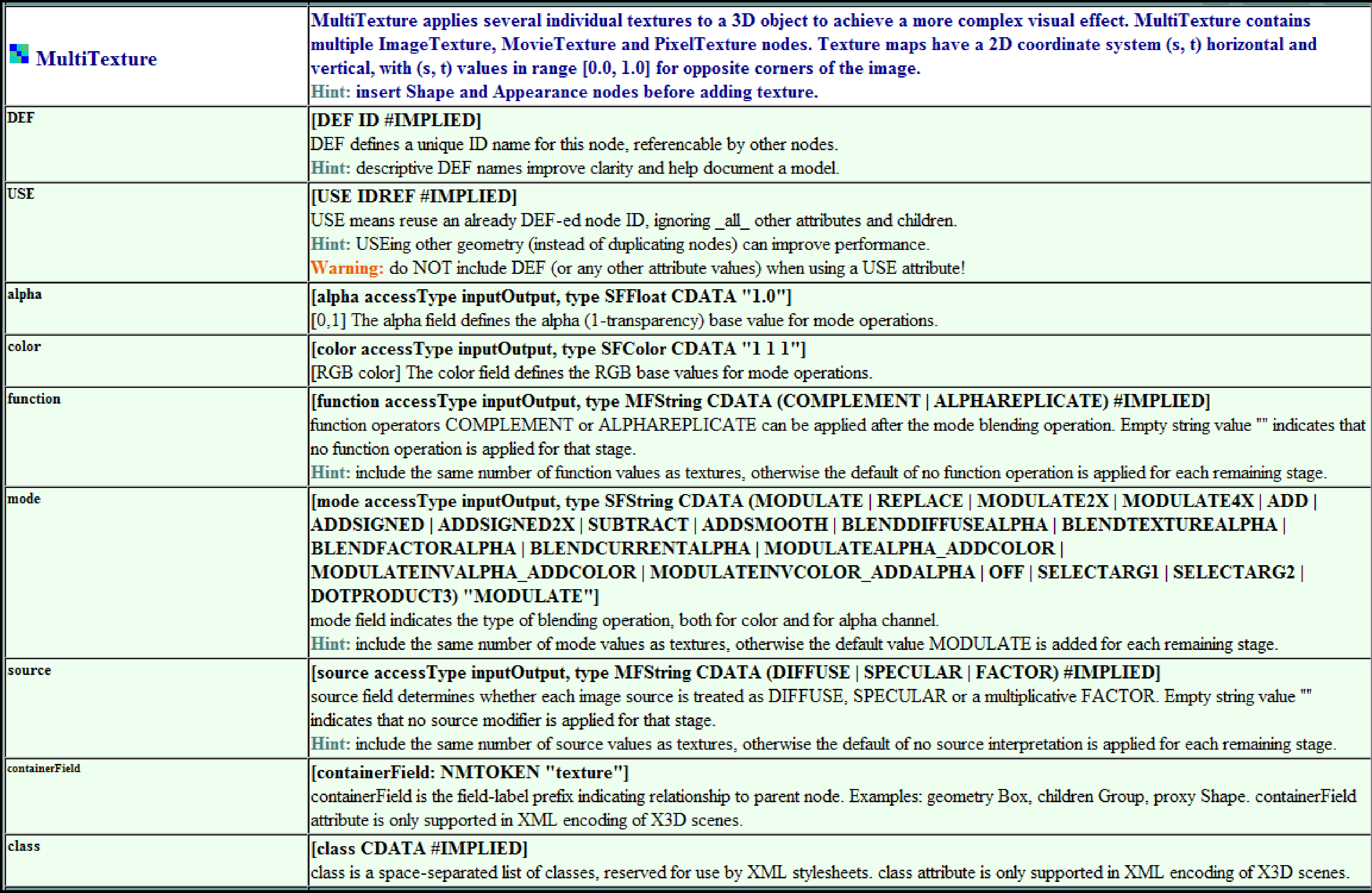
source values
DIFFUSE
SPECULAR
FACTOR

Addition choices:

function list

function values
COMPLEMENT
ALPHAREPLICATE

Addition choices:



MultiTexture	**MultiTexture applies several individual textures to a 3D object to achieve a more complex visual effect. MultiTexture contains multiple ImageTexture, MovieTexture and PixelTexture nodes. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image.** **Hint: insert Shape and Appearance nodes before adding texture.**
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!
alpha	**[alpha accessType inputOutput, type SFFloat CDATA "1.0"]** [0,1] The alpha field defines the alpha (1-transparency) base value for mode operations.
color	**[color accessType inputOutput, type SFColor CDATA "1 1 1"]** [RGB color] The color field defines the RGB base values for mode operations.
function	**[function accessType inputOutput, type MFString CDATA (COMPLEMENT | ALPHAREPLICATE) #IMPLIED]** function operators COMPLEMENT or ALPHAREPLICATE can be applied after the mode blending operation. Empty string value "" indicates that no function operation is applied for that stage. **Hint:** include the same number of function values as textures, otherwise the default of no function operation is applied for each remaining stage.
mode	**[mode accessType inputOutput, type SFString CDATA (MODULATE | REPLACE | MODULATE2X | MODULATE4X | ADD | ADDSIGNED | ADDSIGNED2X | SUBTRACT | ADDSMOOTH | BLENDDIFFUSEALPHA | BLENDTEXTUREALPHA | BLENDFACTORALPHA | BLENDCURRENTALPHA | MODULATEALPHA_ADDCOLOR | MODULATEINVALPHA_ADDCOLOR | MODULATEINVCOLOR_ADDALPHA | OFF | SELECTARG1 | SELECTARG2 | DOTPRODUCT3) "MODULATE"]** mode field indicates the type of blending operation, both for color and for alpha channel. **Hint:** include the same number of mode values as textures, otherwise the default value MODULATE is added for each remaining stage.
source	**[source accessType inputOutput, type MFString CDATA (DIFFUSE | SPECULAR | FACTOR) #IMPLIED]** source field determines whether each image source is treated as DIFFUSE, SPECULAR or a multiplicative FACTOR. Empty string value "" indicates that no source modifier is applied for that stage. **Hint:** include the same number of source values as textures, otherwise the default of no source interpretation is applied for each remaining stage.
containerField	**[containerField: NMTOKEN "texture"]** 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.

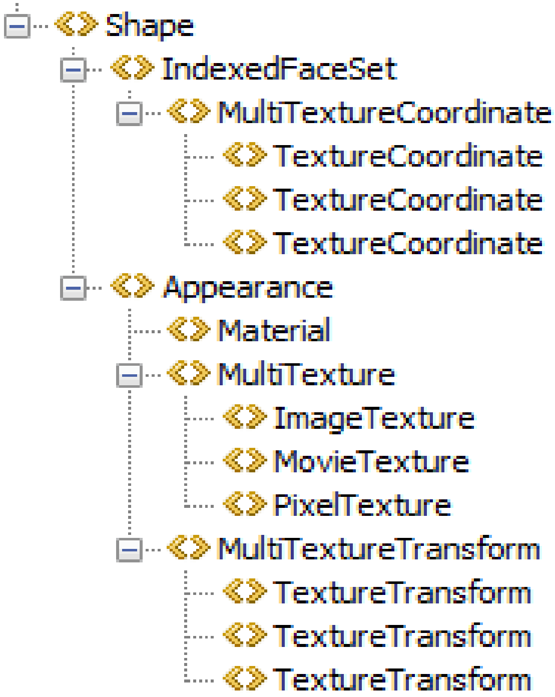
MultiTextureTransform

MultiTextureTransform contains multiple child TextureTransform nodes

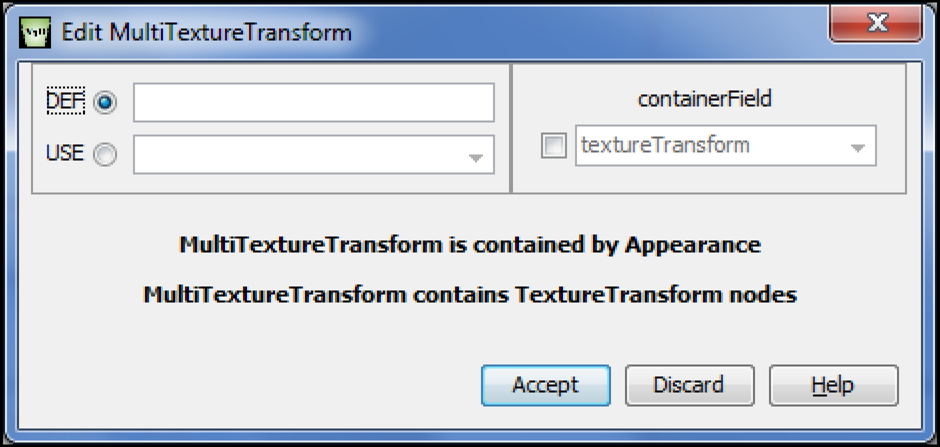
Each of the contained TextureTransform nodes correspond to the multiple texture nodes found in sibling MultiTexture node

- ImageTexture MovieTexture or PixelTexture

Thus supports application of multiple textures



- Shape
  - IndexedFaceSet
    - MultiTextureCoordinate
      - TextureCoordinate
      - TextureCoordinate
      - TextureCoordinate
  - Appearance
    - Material
    - MultiTexture
      - ImageTexture
      - MovieTexture
      - PixelTexture
    - MultiTextureTransform
      - TextureTransform
      - TextureTransform
      - TextureTransform



Edit MultiTextureTransform

DEF ●

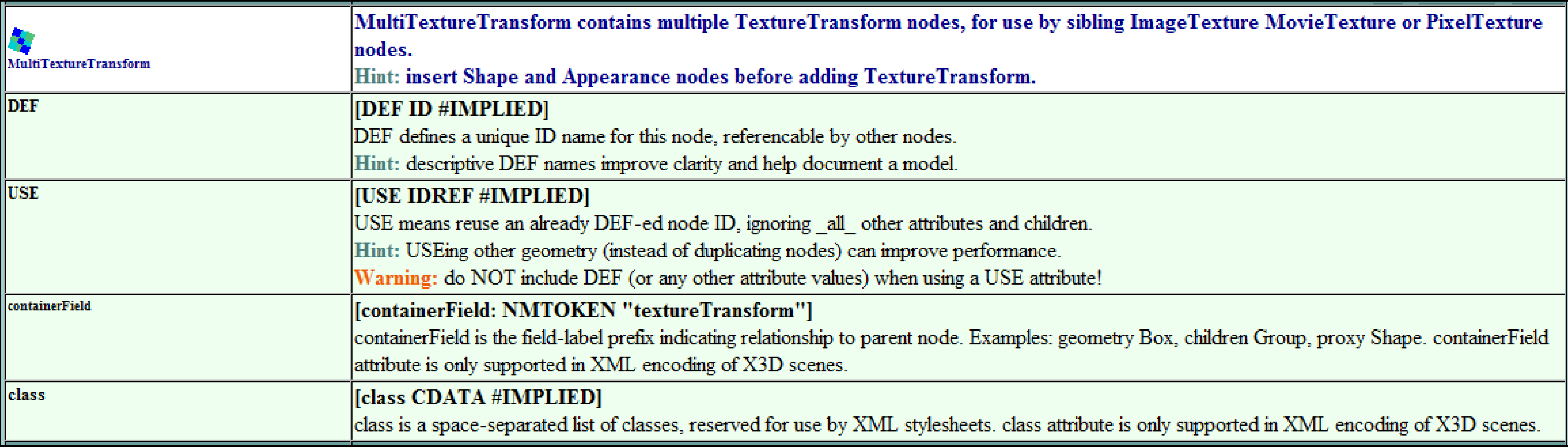
USE ○

containerField

☐ textureTransform

MultiTextureTransform is contained by Appearance

MultiTextureTransform contains TextureTransform nodes



MultiTextureTransform	**MultiTextureTransform contains multiple TextureTransform nodes, for use by sibling ImageTexture MovieTexture or PixelTexture nodes.** **Hint: insert Shape and Appearance nodes before adding TextureTransform.**
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!
containerField	**[containerField: NMTOKEN "textureTransform"]** 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.

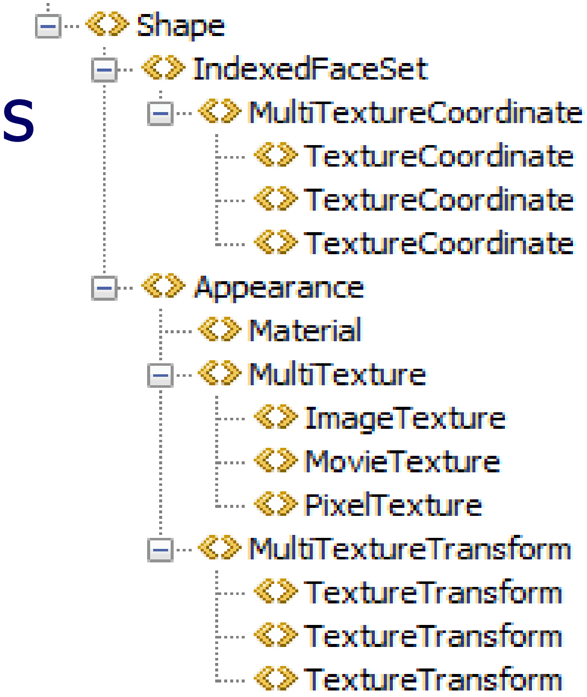
MultiTextureCoordinate

MultiTextureCoordinate contains multiple child TextureCoordinate, TextureCoordinateGenerator nodes

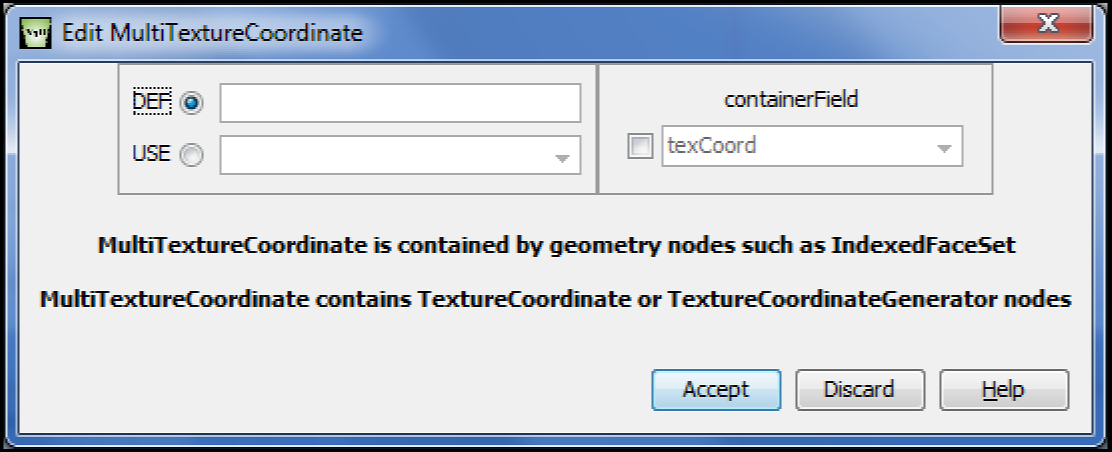
- for use by a parent polygonal geometry node such as IndexedFaceSet or Triangle* node

Contained TextureCoordinate nodes correspond to texture nodes

- which in turn are contained in sibling Appearance/MultiTexture



```
Shape
    IndexedFaceSet
        MultiTextureCoordinate
            TextureCoordinate
            TextureCoordinate
            TextureCoordinate
    Appearance
        Material
        MultiTexture
            ImageTexture
            MovieTexture
            PixelTexture
        MultiTextureTransform
            TextureTransform
            TextureTransform
            TextureTransform
```



DEF ●

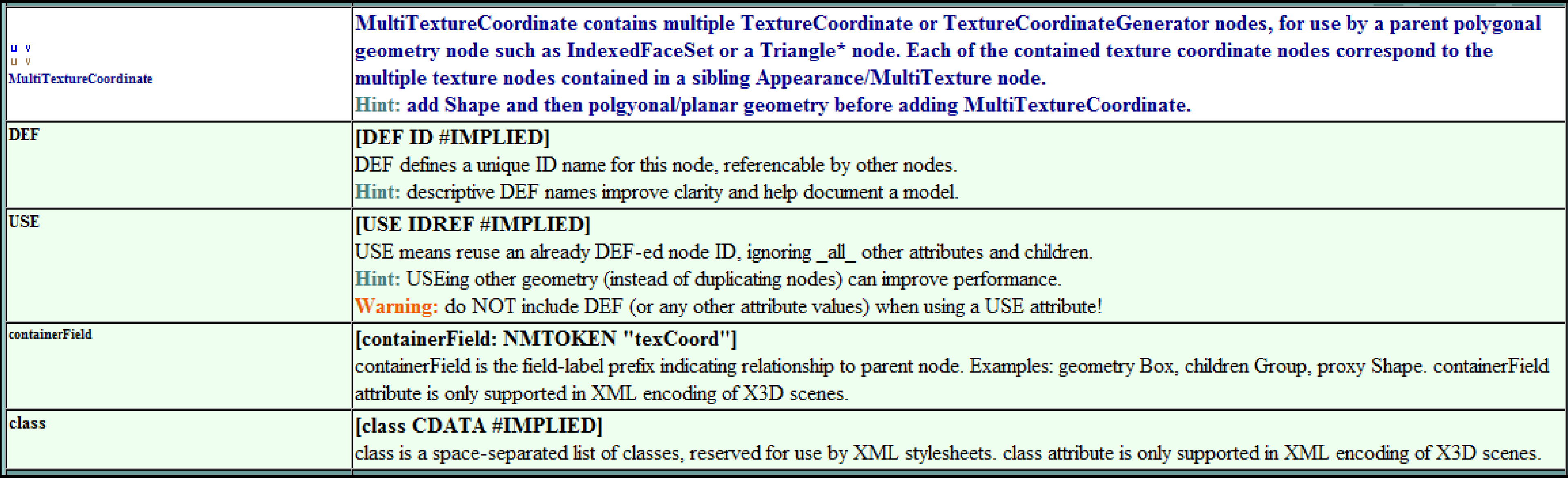
USE ○ ▾

containerField

☐ texCoord ▾

MultiTextureCoordinate is contained by geometry nodes such as IndexedFaceSet

MultiTextureCoordinate contains TextureCoordinate or TextureCoordinateGenerator nodes



MultiTextureCoordinate	**MultiTextureCoordinate contains multiple TextureCoordinate or TextureCoordinateGenerator nodes, for use by a parent polygonal geometry node such as IndexedFaceSet or a Triangle* node. Each of the contained texture coordinate nodes correspond to the multiple texture nodes contained in a sibling Appearance/MultiTexture node.** **Hint: add Shape and then polgyonal/planar geometry before adding MultiTextureCoordinate.**
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!
containerField	**[containerField: NMTOKEN "texCoord"]** 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.

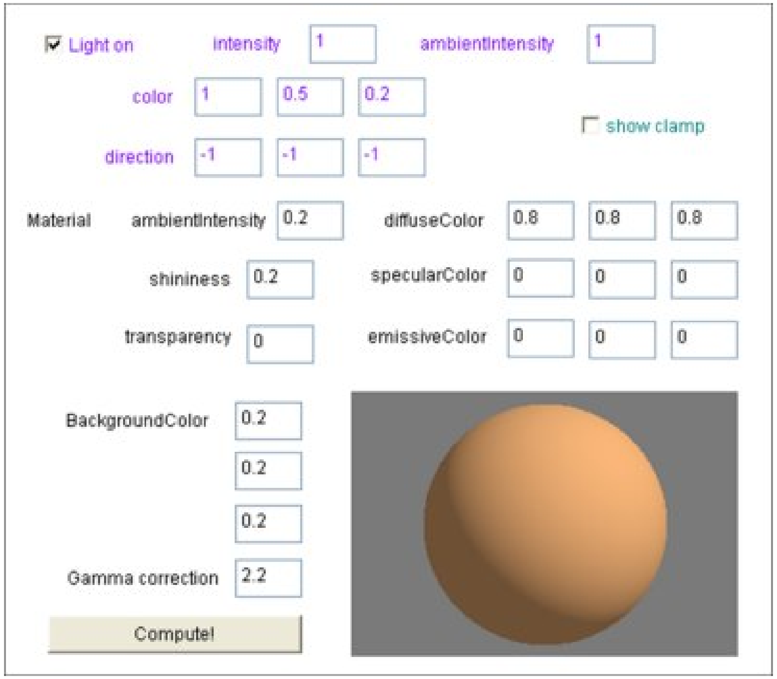
Additional Resources

Additional Resources

Pellucid materials editor

Pellucid materials editor provides high-fidelity rendering of the VRML (X3D) materials model

- Eric Haines, copyright (c) 1997
- http://tog.acm.org/resources/applets/vrml/pellucid.html



☑ Light on

intensity 1

ambientIntensity 1

color 1 0.5 0.2

□ show clamp

direction -1 -1 -1

Material

ambientIntensity 0.2

diffuseColor 0.8 0.8 0.8

shininess 0.2

specularColor 0 0 0

transparency 0

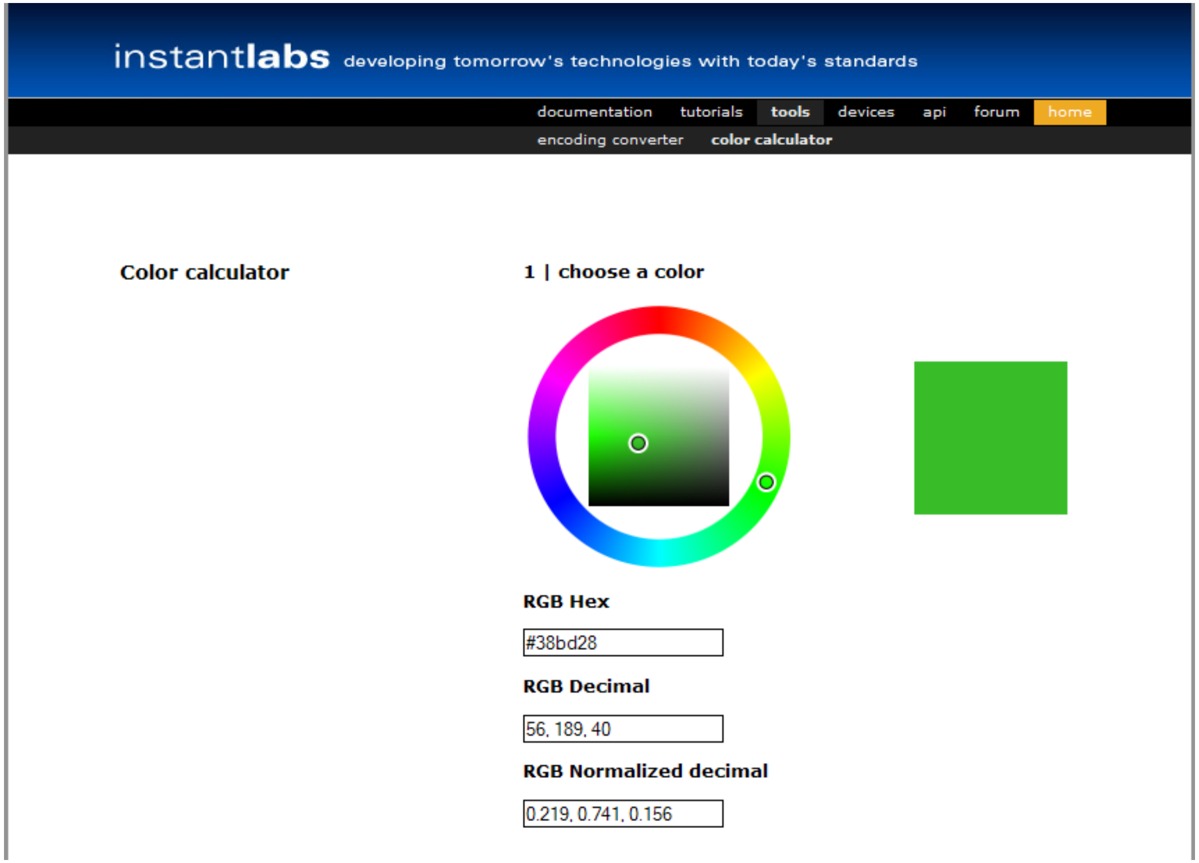
emissiveColor 0 0 0

BackgroundColor 0.2 0.2 0.2

Gamma correction 2.2

InstantReality color calculator

- http://instantreality.org/tools/color_calculator



Color calculator

1 | choose a color

RGB Hex

#38bd28

RGB Decimal

56, 189, 40

RGB Normalized decimal

0.219, 0.741, 0.156

PNG and JPEG image formats

Portable Network Graphics (PNG) image format

- http://www.w3.org/Graphics/PNG W3C activity
- http://www.w3.org/TR/PNG specification
- http://www.libpng.org/pub/png PNG home page
- http://www.libpng.org/pub/png/slashpng-1999.html Story of PNG

JPEG

- JPEG File Interchange Format (JFIF) http://www.w3.org/Graphics/JPEG/jfif.txt
- http://www.jpeg.org JPEG committee home page

Scalable Vector Graphics (SVG)

SVG describes 2D graphics using scalably sized vector definitions, rather than raster pixels

- SVG supports 2D images, interactive applications
- Support for SVG not directly required in X3D

Multiple specifications sharing a common core

- SVG, Mobile Profiles (Basic and Tiny), SVG Print
- W3C home: http://www.w3.org/Graphics/SVG
- Community: http://svg.org

Conversion to other formats available using Batik

- http://xmlgraphics.apache.org/batik open-source

MPEG and other video formats

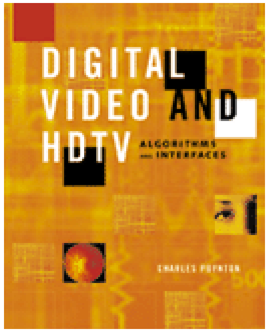
Caveat: the video format arena is dominated by proprietary, incompatible formats.

MPEG

- http://www.mpeg.org MPEG reference website
- http://www.chiariglione.org/mpeg MPEG home

Video

- Poynton, Charles, *Digital Video and HDTV: Algorithms and Interfaces,* Morgan Kaufmann Publishers, 2003. http://www.poynton.com/DVAI



Chapter Summary

Chapter Summary

Appearance affects associated geometry, containing the following fields

Visual surface properties that interact with lights

- Material and TwoSidedMaterial
- LineProperties and FillProperties

Texture nodes wrap images onto geometry

- ImageTexture, MovieTexture, PixelTexture and MultiTexture
- TextureTransform, TextureCoordinate and TextureCoordinateGenerator

Suggested exercises

Compare different materials on identical shapes

Demonstrate the use of ImageTexture nodes by taking (or finding) photos of interest and then applying them to corresponding geometry

- Be sure to give credit for someone else's content, do not use unlicensed imagery without permission

Build a PixelTexture image, apply it to geometry

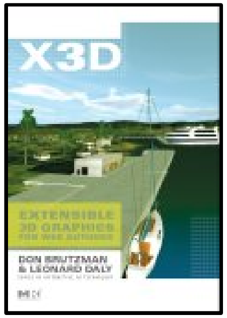
Demonstrate use of MovieTexture video applied to square geometry within a Billboard node

- Always observe credit, licensing requirements

References

References 1

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



- Chapter 5, Appearance Material and Textures
- 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

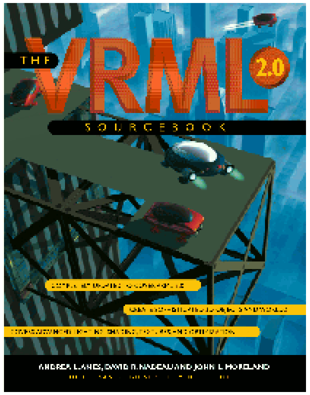
X3D Scene Authoring Hints

- 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



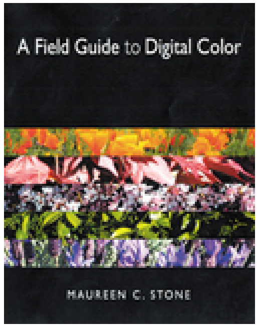
VRML 2.0 Sourcebook by Andrea L. Ames, David R. Nadeau, and John L. Moreland, John Wiley & Sons, 1996

- http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm
- http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook
- Chapter 10 – Materials
- Chapter 17 – Textures
- Chapter 18 – Texture Mapping
- Chapter 21 – Shiny Materials

References 4

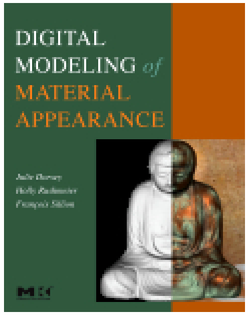
Stone, Maureen C., *A Field Guide to Digital Color,* A.K. Peters Publishing, 2003

- http://www.akpeters.com/product.asp?ProdCode=1616

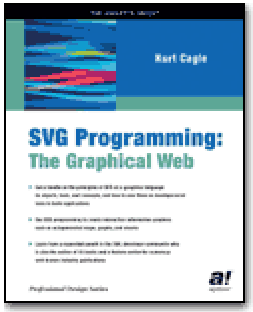


Dorsey, Julie, Rushmeier, Holly and Sillion, François, *Digital Modeling of Material Appearance*, Morgan Kaufmann Publishing, December 2007

- http://www.elsevierdirect.com/product.jsp?lid=0&iid=16&sid=0&isbn=9780122211812
- http://www.siggraph.org/s2005/main.php?f=conference&p=courses&s=24



References 5



Kurt Cagle

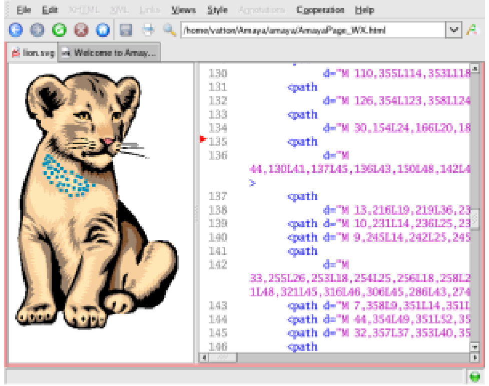
SVG Programming:
The Graphical Web

Kurt Cagle, *SVG Programming: The Graphical Web*, Apress, 2002.

- http://www.apress.com/book/view/1590590198

Amaya open-source editor from W3C supports HTML, CSS, MathML and Scalable Vector Graphics (SVG).

- http://www.w3.org/Amaya

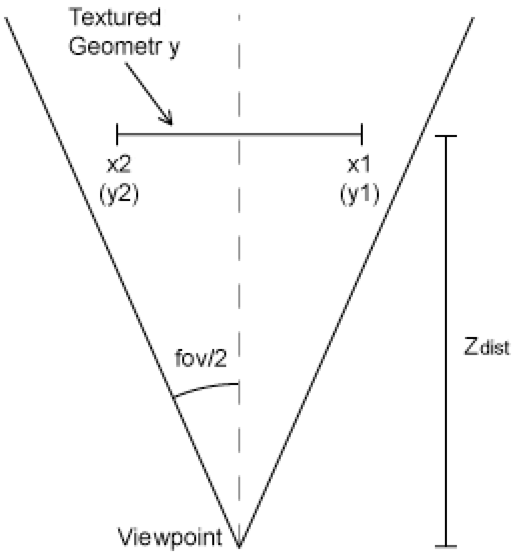


```
                d="M 110,355L114,353L118
        <path
                d="M 126,354L123,358L124
        <path
                d="M 30,154L24,166L20,18
        <path
                d="M
    44,130L41,137L45,136L43,150L48,142L4
    >
        <path
                d="M 13,216L19,219L36,23
        <path d="M 10,231L14,236L25,23
        <path d="M 9,245L14,242L25,245
        <path
                d="M
    33,255L26,253L18,254L25,256L18,258L2
    1L48,321L45,316L46,306L45,286L43,274
        <path d="M 7,358L9,351L14,351L
        <path d="M 44,354L49,351L52,35
        <path d="M 32,357L37,353L40,35
        <path
```

References 6

Pixel Perfect Text by David Frerichs

- Overcome poor pixelation of Text nodes by creating a texture image of the desired text, along with a matching Viewpoint at the right distance
- http://www.frerichs.net/vrml2/pp/pixel_perfect.html



References 7

Charles Poynton, *Frequently Asked Questions (FAQ) about Color* and *Gamma FAQ/FQA*

- www.poynton.com/PDFs/ColorFAQ.pdf
- www.poynton.com/notes/color/GammaFQA.html

COLOURLovers.com

- "A creative community where people from around the world create and share colors, palettes and patterns, discuss the latest trends and explore colorful articles"
- www.colourLovers.com

References 8

Wikipedia: material and color

- **3D rendering** http://en.wikipedia.org/wiki/3D_rendering
- **List of colors, values** http://en.wikipedia.org/wiki/List_of_colors (compact)
- Color gamut http://en.wikipedia.org/wiki/Color_gamut
- **RGB color model** http://en.wikipedia.org/wiki/RGB_color_model
- **RGB color space** http://en.wikipedia.org/wiki/RGB_color_space
- **Web colors, values** http://en.wikipedia.org/wiki/Web_colors

References 9

Wikipedia: video

- **Video** and **Video digital encoding formats**
  http://en.wikipedia.org/wiki/Video_formats
  http://en.wikipedia.org/wiki/Video#Digital_encoding_formats
- **Video codec (coder/decoder)**
  http://en.wikipedia.org/wiki/Video_codec
- **List of codecs** and **Comparison of codecs**
  http://en.wikipedia.org/wiki/List_of_codecs#Video_codecs
  http://en.wikipedia.org/wiki/Comparison_of_video_codecs

Slideset TODO

- Add TestCube.x3d
- Consider adding a triangular version of TestCube that uses triangles and TextureTransform slicing instead of quads
- add example scene using flipped texture
- Hugin panoramic image stitching program http://sourceforge.net/projects/hugin
- Skypaint http://www.skypaint.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! ☺

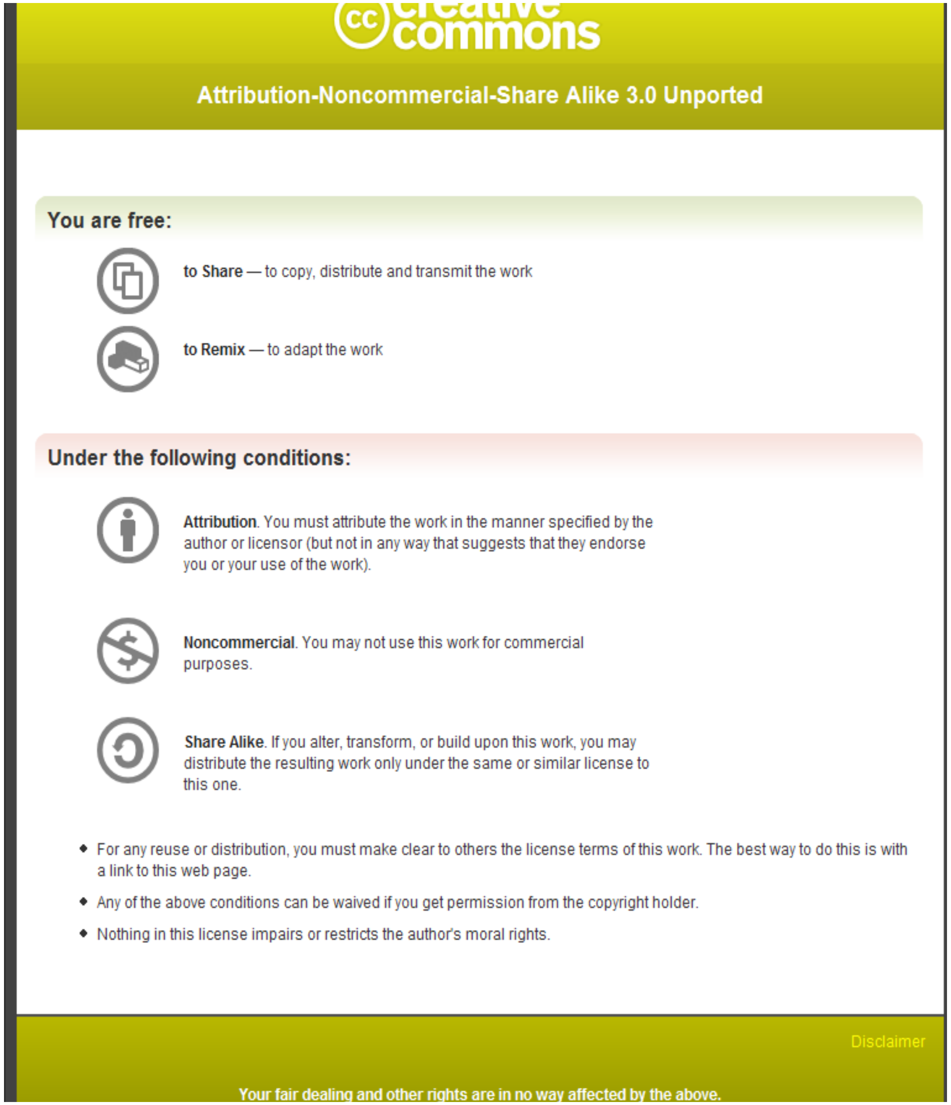
- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics

cgems ACMSIGGRAPH

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.

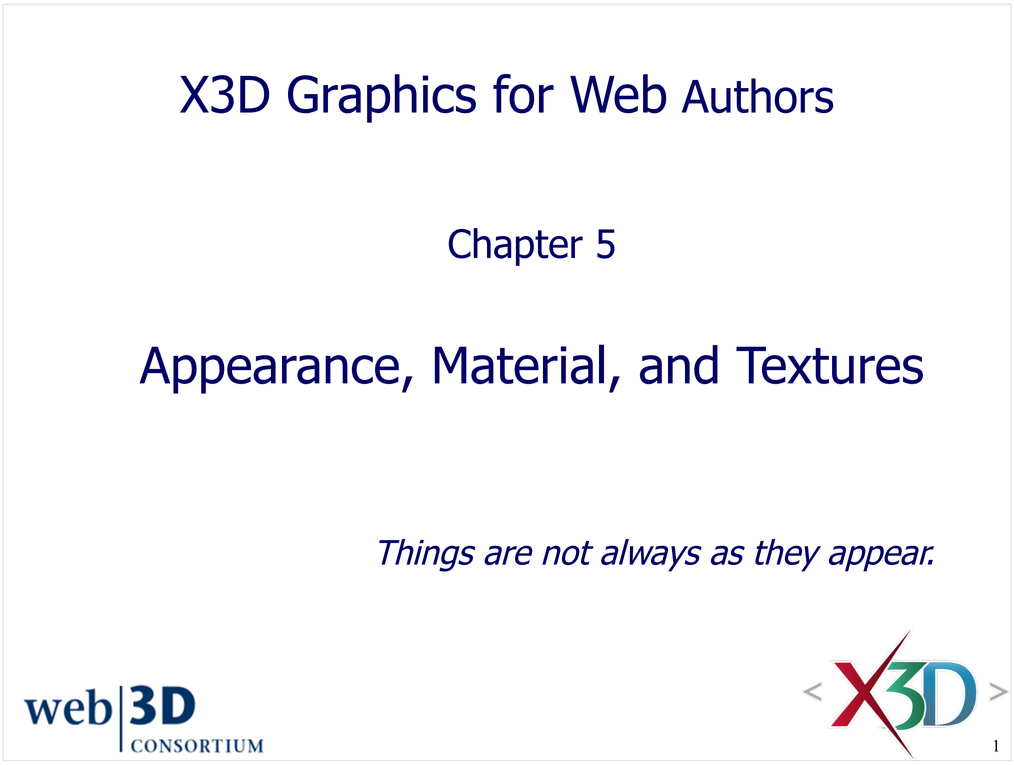
- For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's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

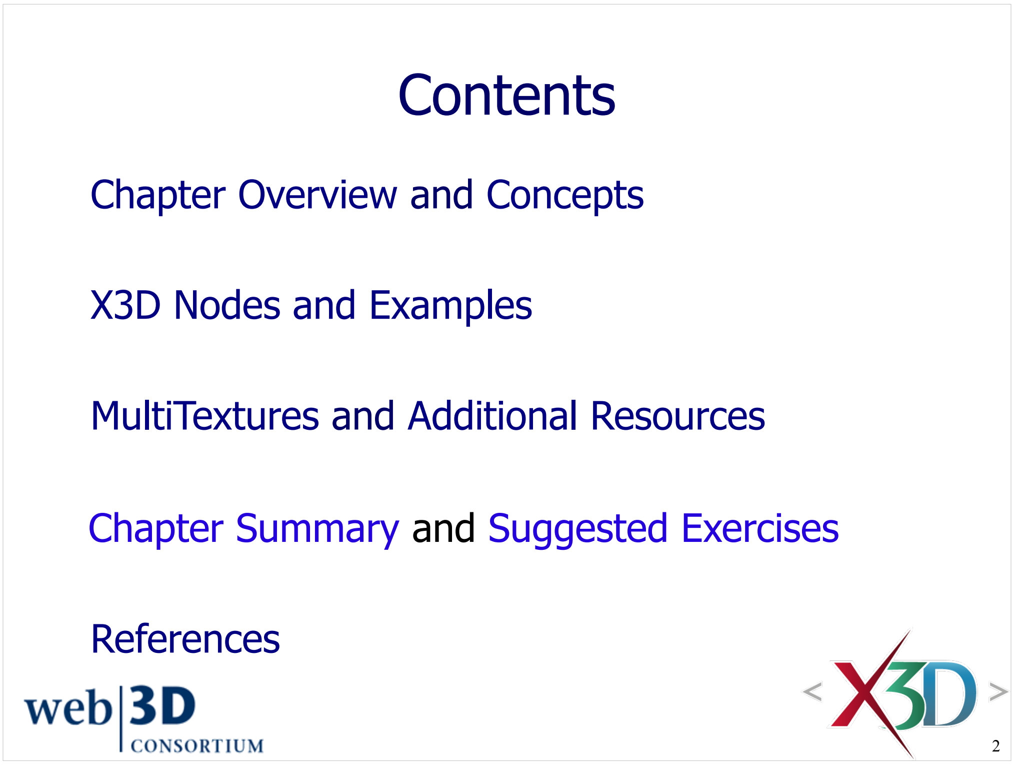


X3D Graphics for Web Authors

Chapter 5

Appearance, Material, and Textures

Things are not always as they appear.



Contents

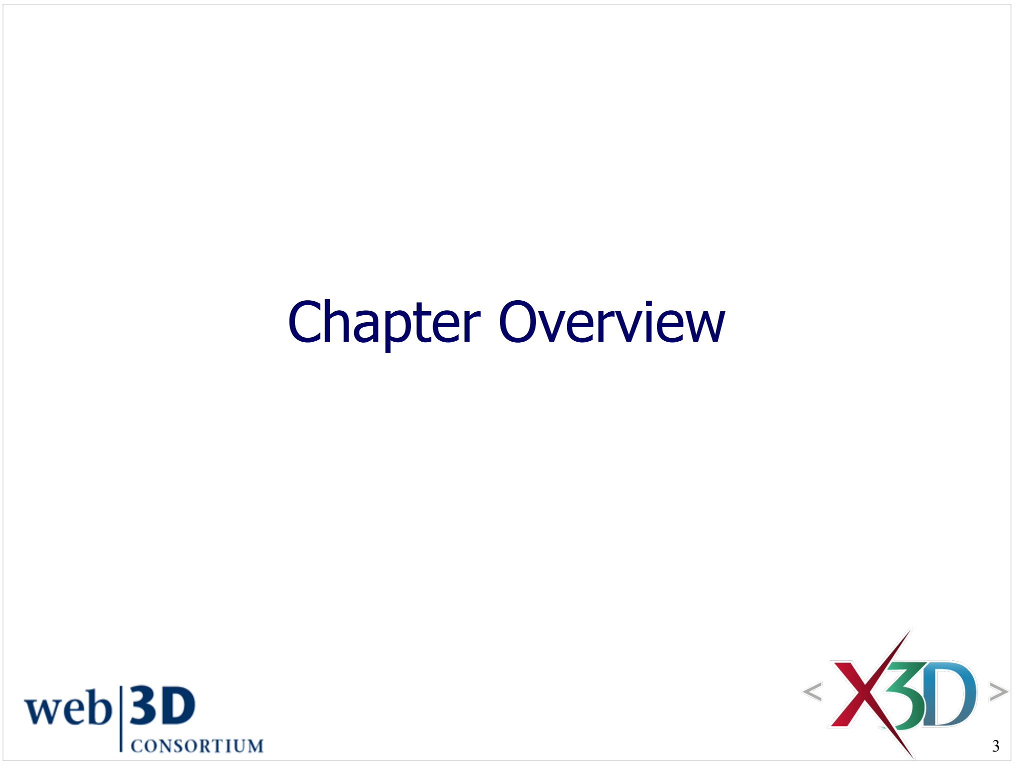
Chapter Overview and Concepts

X3D Nodes and Examples

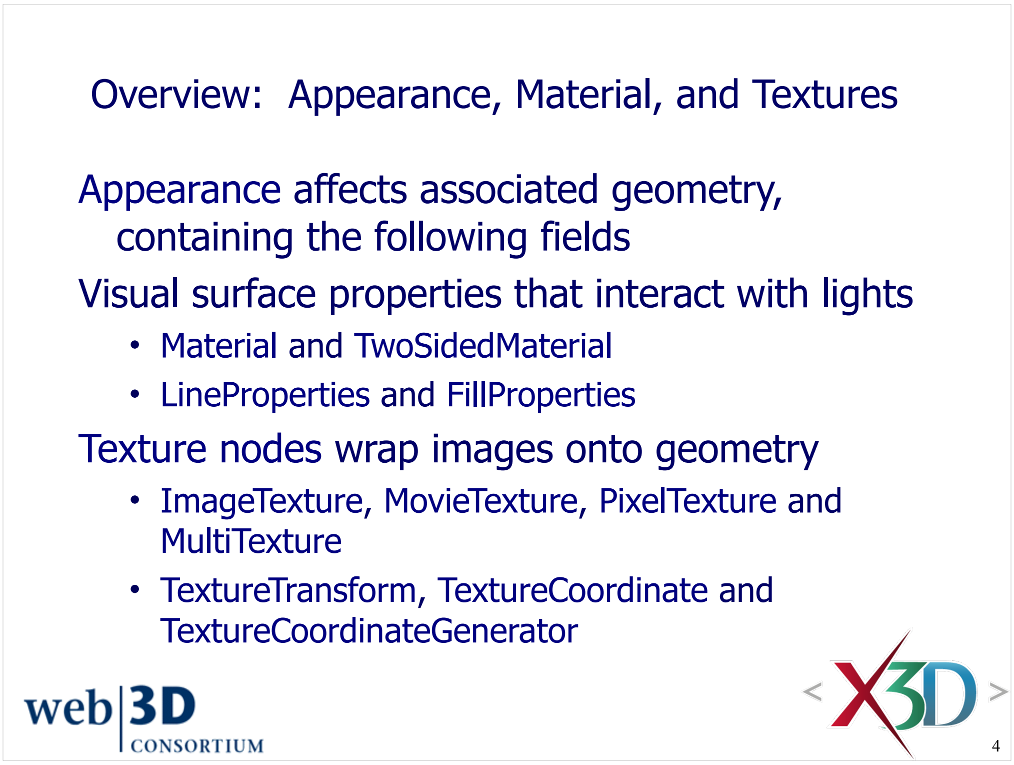
MultiTextures and Additional Resources

Chapter Summary and Suggested Exercises

References



Chapter Overview



Overview: Appearance, Material, and Textures

Appearance affects associated geometry, containing the following fields

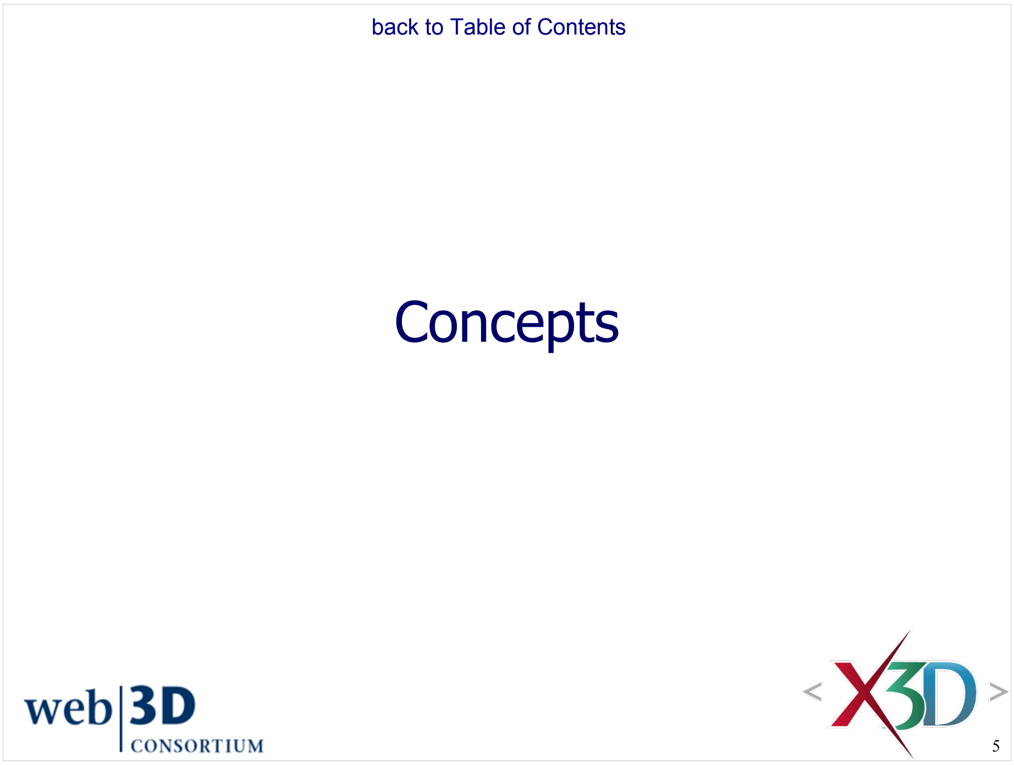
Visual surface properties that interact with lights

- Material and TwoSidedMaterial
- LineProperties and FillProperties

Texture nodes wrap images onto geometry

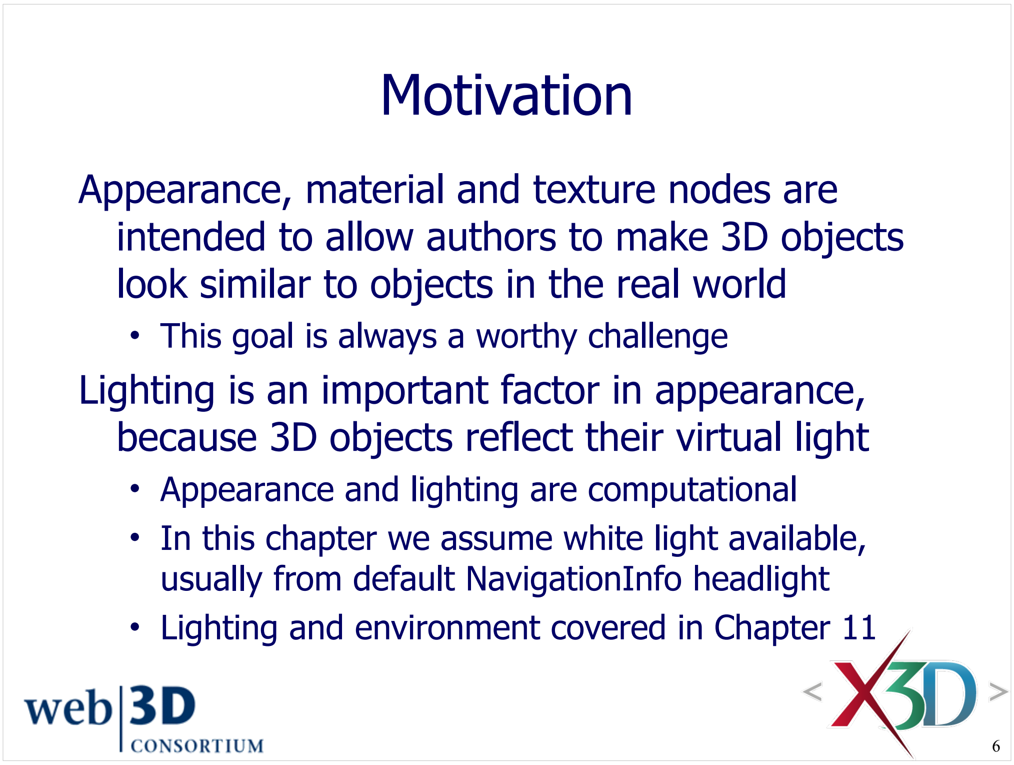
- ImageTexture, MovieTexture, PixelTexture and MultiTexture
- TextureTransform, TextureCoordinate and TextureCoordinateGenerator

4



back to Table of Contents

Concepts



Motivation

Appearance, material and texture nodes are intended to allow authors to make 3D objects look similar to objects in the real world

- This goal is always a worthy challenge

Lighting is an important factor in appearance, because 3D objects reflect their virtual light

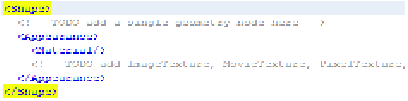
- Appearance and lighting are computational
- In this chapter we assume white light available, usually from default NavigationInfo headlight
- Lighting and environment covered in Chapter 11

In a number of ways, lighting and appearance in 3D graphics are the theoretical inverses of optical properties. Graphics attempts to recreate optical properties computationally.

Parent-child constraints

Each Shape node can contain

- Single geometry node
- Single Appearance node



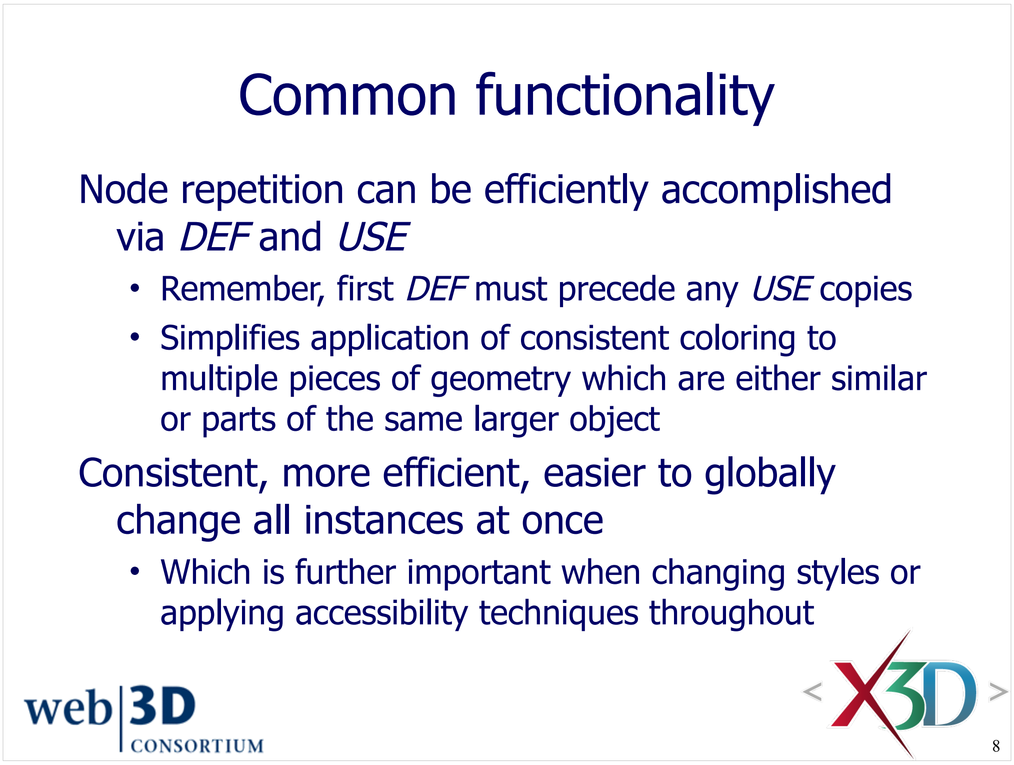
```
<Shape>
  <!-- TODO add a single geometry node here -->
  <Appearance>
    <Material/>
    <!-- TODO add ImageTexture, MovieTexture, PixelTexture,
  </Appearance>
</Shape>
```

Each Appearance node can contain

- A single Material (or TwoSidedMaterial) node
- FillProperties, LineProperties, TextureProperties
- A single Texture node (image, pixel or movie)

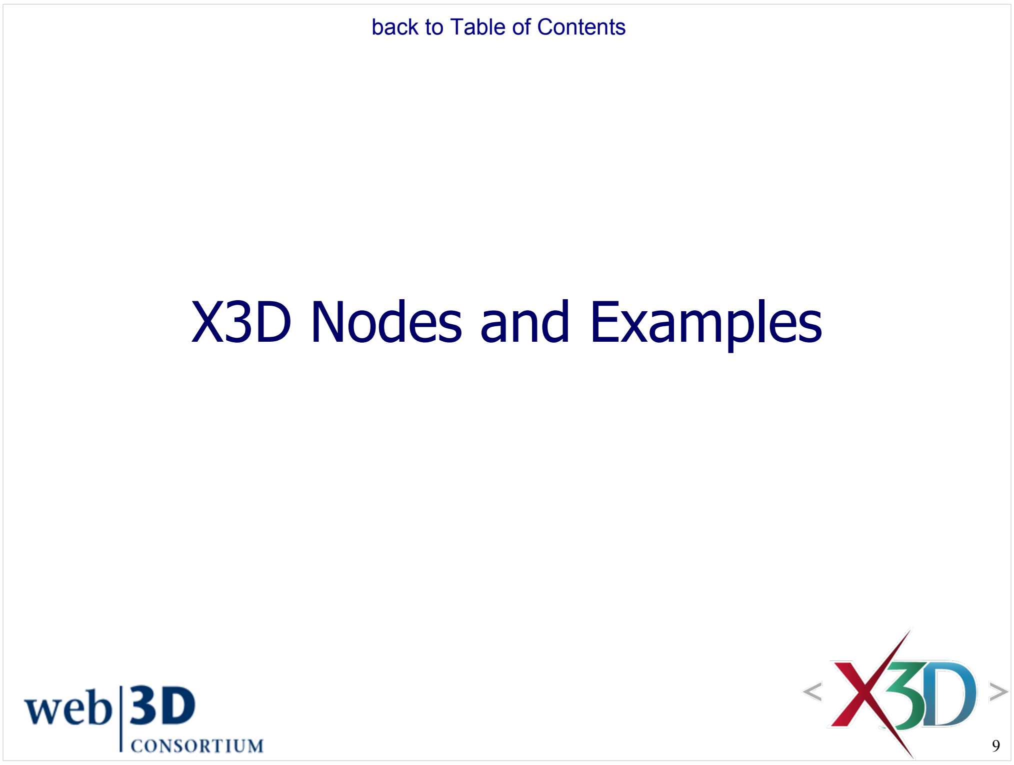
Each Texture node can contain

- Single TextureTransform or TextureTransformGenerator node



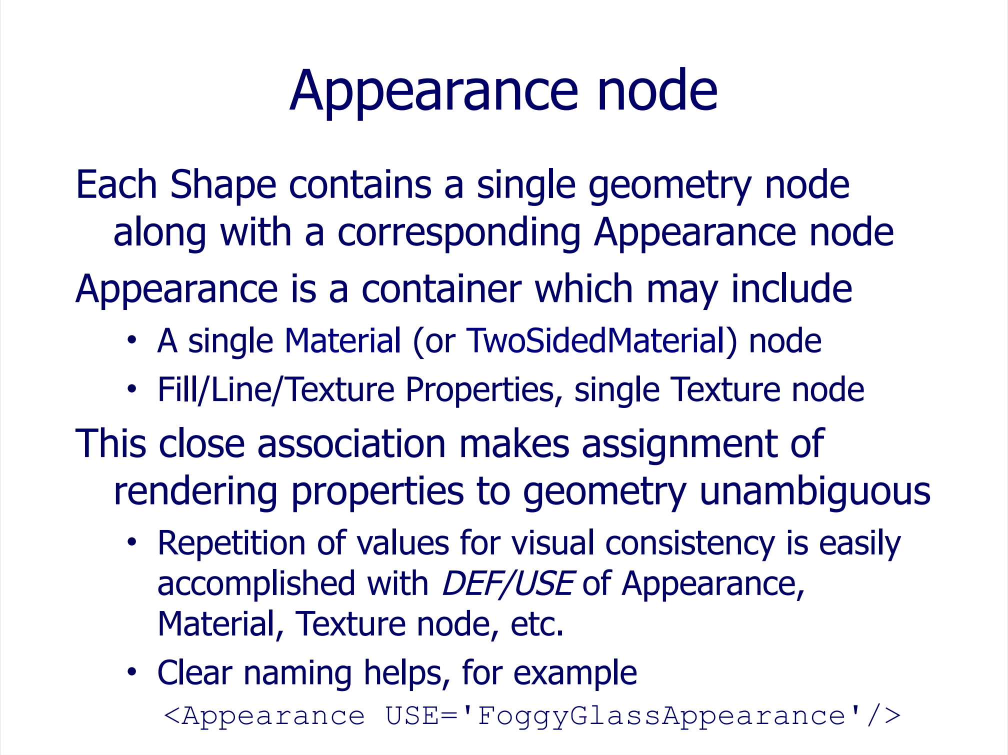
Common functionality

- Node repetition can be efficiently accomplished via *DEF* and *USE*
  - Remember, first *DEF* must precede any *USE* copies
  - Simplifies application of consistent coloring to multiple pieces of geometry which are either similar or parts of the same larger object
- Consistent, more efficient, easier to globally change all instances at once
  - Which is further important when changing styles or applying accessibility techniques throughout



back to Table of Contents

X3D Nodes and Examples



Appearance node

Each Shape contains a single geometry node along with a corresponding Appearance node

Appearance is a container which may include

- A single Material (or TwoSidedMaterial) node
- Fill/Line/Texture Properties, single Texture node

This close association makes assignment of rendering properties to geometry unambiguous

- Repetition of values for visual consistency is easily accomplished with *DEF/USE* of Appearance, Material, Texture node, etc.
- Clear naming helps, for example

```
<Appearance USE='FoggyGlassAppearance'/>
```

DEF/USE names can get confusing in a large X3D scene, unless good patterns and habits are used when giving names to nodes.

For example, a DEF name of `FoggyGlass` certainly describes what is intended, but it is not clear whether the node is an Appearance, Material, or even some kind of Texture. Therefore, including the name of the defining node in the DEF name (e.g. `FoggyGlassAppearance`) makes it easy to copy.

In other words, it is more likely to later say

```
<Appearance USE='FoggyGlassAppearance'/>
```

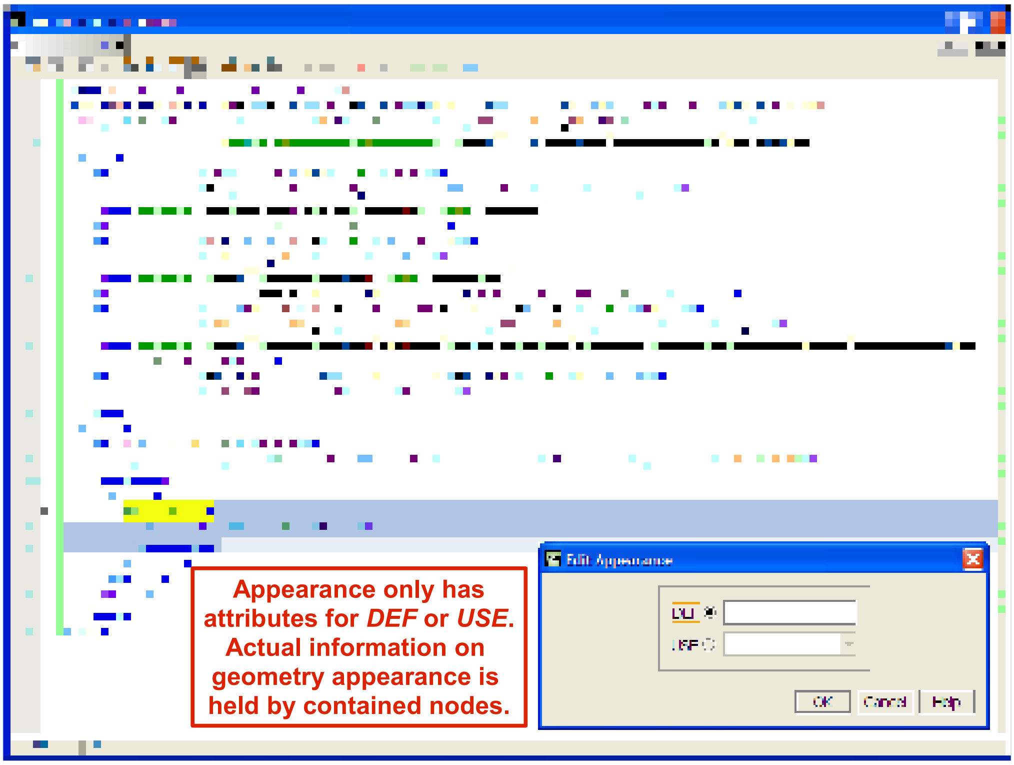
instead of making the node-typing mistake

```
<Material USE='FoggyGlass'/> <!-- run-time error -->
```

Since such run-time errors are often not caught until an end user is trying to view a scene with unintended errors, it is better to adopt good naming practices early to avoid puzzling problems later.

Thumbrules on node-naming conventions are given in the X3D Scene Authoring Hints, provided in the X3D-Edit help system and also online at

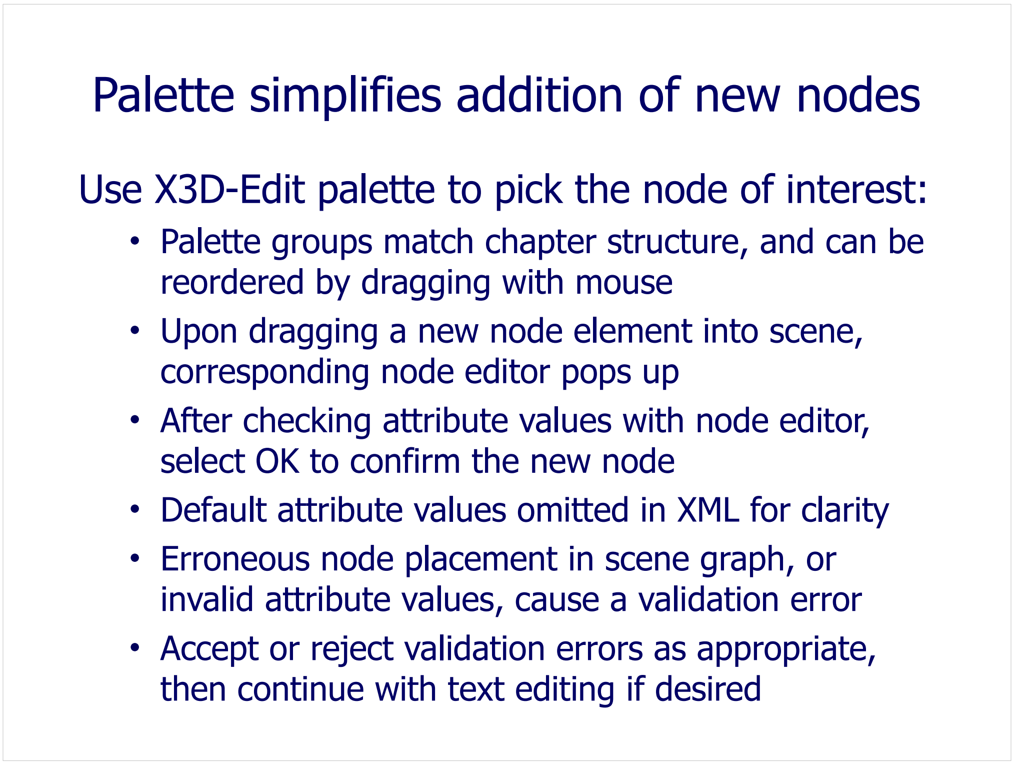
http://www.web3d.org/x3d/content/examples/X3dSceneAuthoringHints.html#NamingConventions



Appearance only has attributes for *DEF* or *USE*. Actual information on geometry appearance is held by contained nodes.

When adding an Appearance node from the palette, the following text appears in the editing pane, prompting further node addition(s):

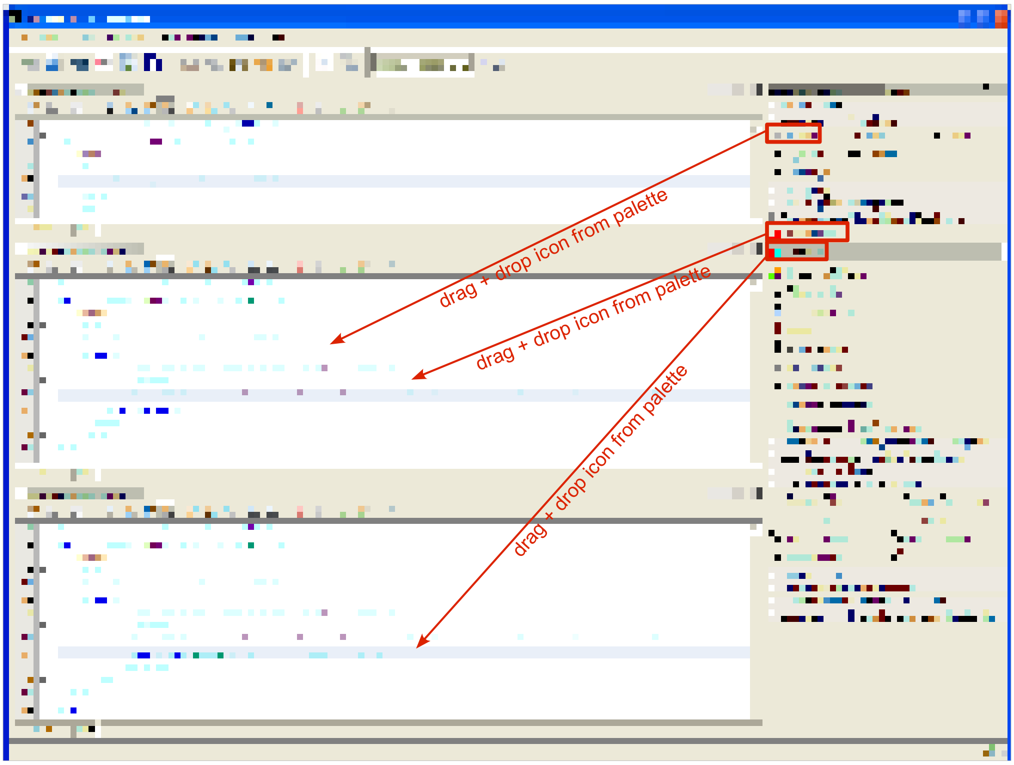
```
<Appearance>
    <!--Add Material, Texture, TextureTransform, FillProperties, and/or
        LineProperties nodes here-->
</Appearance>
```



Palette simplifies addition of new nodes

Use X3D-Edit palette to pick the node of interest:

- Palette groups match chapter structure, and can be reordered by dragging with mouse
- Upon dragging a new node element into scene, corresponding node editor pops up
- After checking attribute values with node editor, select OK to confirm the new node
- Default attribute values omitted in XML for clarity
- Erroneous node placement in scene graph, or invalid attribute values, cause a validation error
- Accept or reject validation errors as appropriate, then continue with text editing if desired



drag + drop icon from palette
drag + drop icon from palette
drag + drop icon from palette

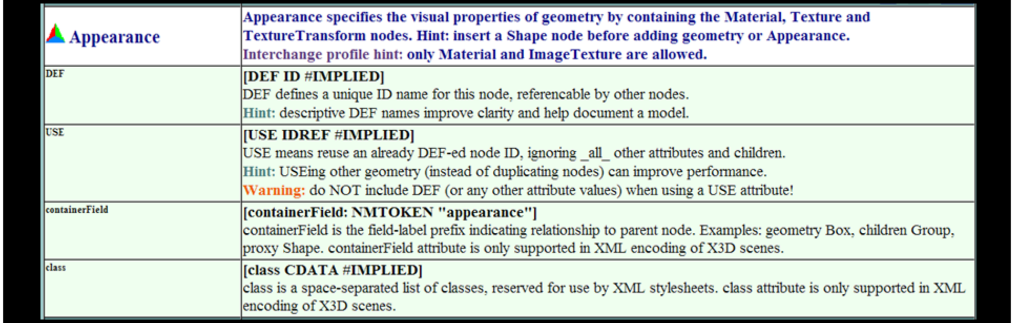
Hint: place the cursor before comments and closing tags, and then press Enter (return key for line feeds), to get proper line spacing and to make the scene easier to read.

Embedded comments (that prompt where new nodes are inserted) can be deleted.

When all nodes are in place, you can reformat by selecting

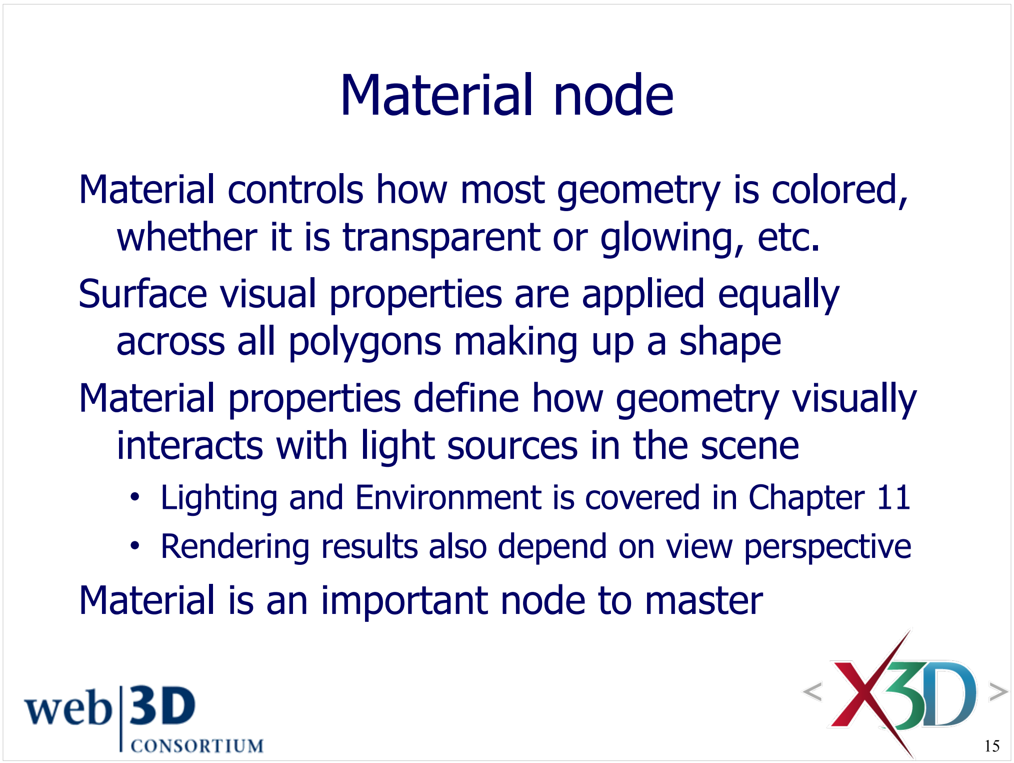
- Control+A to select all nodes
- Alt+Shift+F to format the XML (also available via right-click context menu)

Note that `head` element is iconized and DOCTYPE deleted in these scenes for clarity.



Appearance	Appearance specifies the visual properties of geometry by containing the Material, Texture and TextureTransform nodes. Hint: insert a Shape node before adding geometry or Appearance. Interchange profile hint: only Material and ImageTexture are allowed.
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!
containerField	**[containerField: NMTOKEN "appearance"]** 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#Appearance



Material node

Material controls how most geometry is colored, whether it is transparent or glowing, etc.

Surface visual properties are applied equally across all polygons making up a shape

Material properties define how geometry visually interacts with light sources in the scene

- Lighting and Environment is covered in Chapter 11
- Rendering results also depend on view perspective

Material is an important node to master

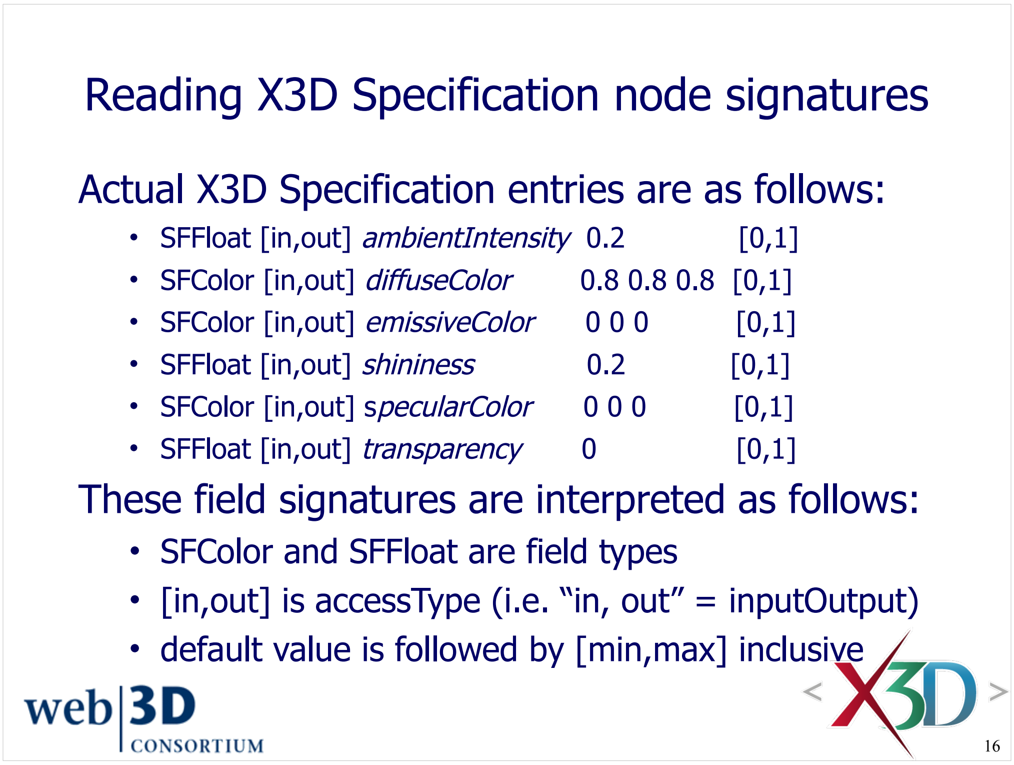
web|3D CONSORTIUM

<X3D>

15

This is a good time to look at the X3D specification entry for Material, either within X3D Help or online at

http://www.web3d.org/x3d/specifications/ISO-IEC-FDIS-19775-1.2-X3D-AbstractSpecification/Part01/components/shape.html#Material



Reading X3D Specification node signatures

Actual X3D Specification entries are as follows:

- SFFloat [in,out] *ambientIntensity* 0.2 [0,1]
- SFColor [in,out] *diffuseColor* 0.8 0.8 0.8 [0,1]
- SFColor [in,out] *emissiveColor* 0 0 0 [0,1]
- SFFloat [in,out] *shininess* 0.2 [0,1]
- SFColor [in,out] s*pecularColor* 0 0 0 [0,1]
- SFFloat [in,out] *transparency* 0 [0,1]

These field signatures are interpreted as follows:

- SFColor and SFFloat are field types
- [in,out] is accessType (i.e. “in, out” = inputOutput)
- default value is followed by [min,max] inclusive

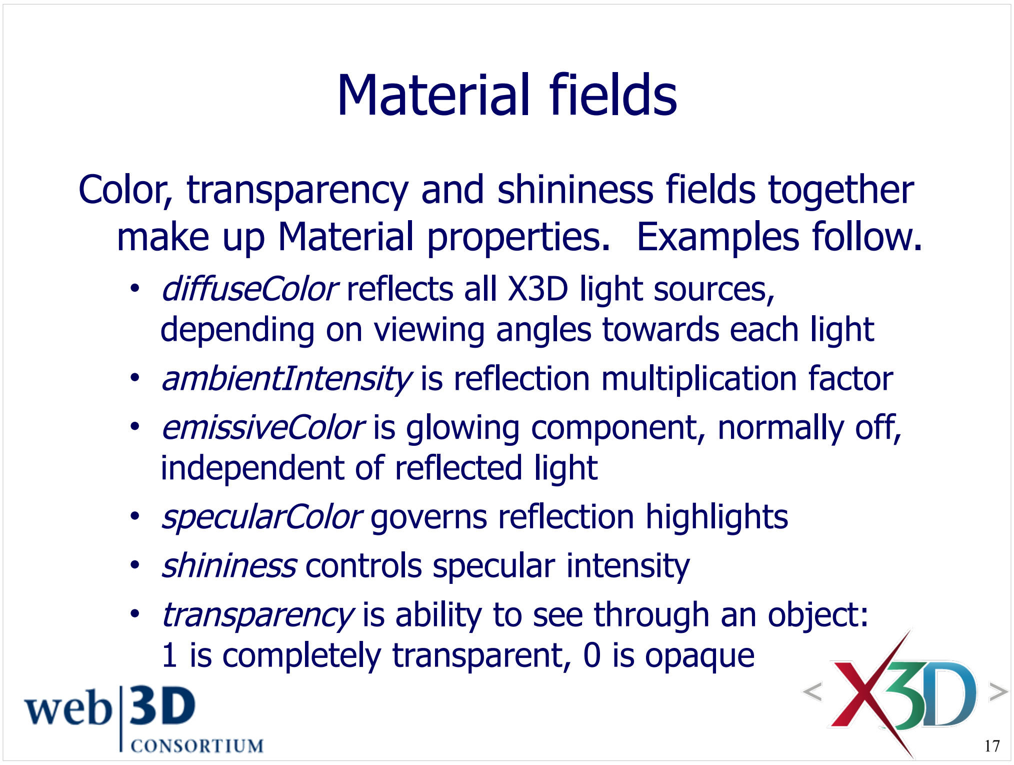
Note that [0,1] means any value from 0 to one.

[Square brackets] mean inclusive, (parentheses) mean exclusive.

Thus [0, ∞) means a range from zero inclusive to infinity exclusive.

This is a good time to look at the X3D specification entry for TwoSidedMaterial, either within X3D Help or online at

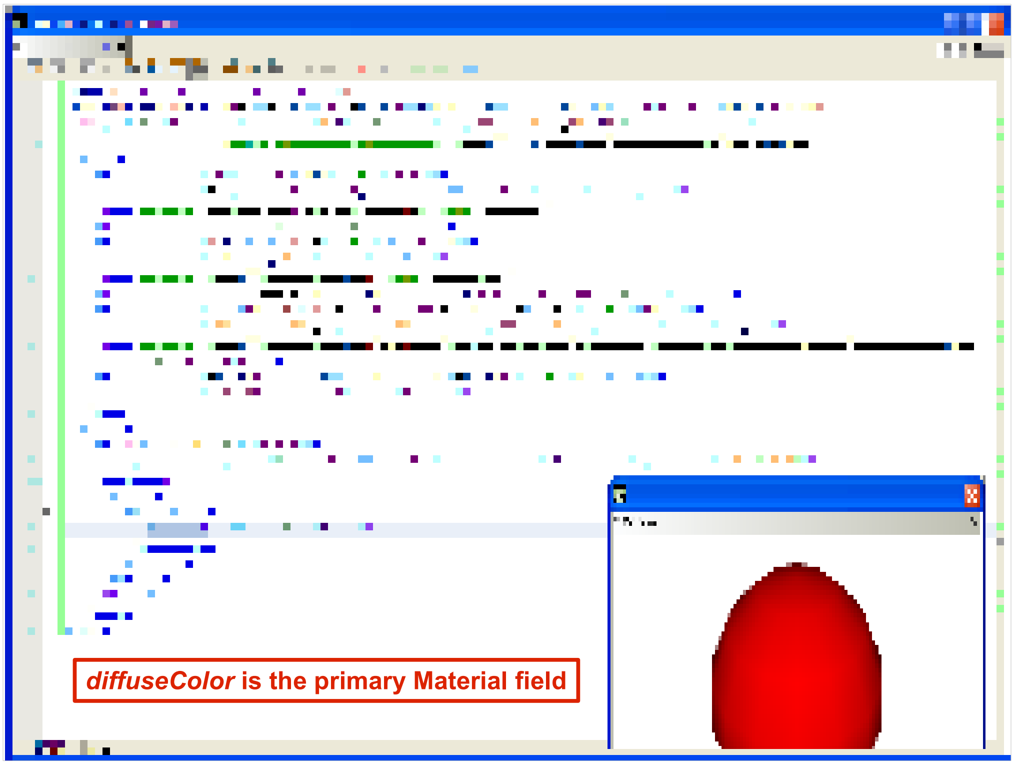
http://www.web3d.org/x3d/specifications/ISO-IEC-FDIS-19775-1.2-X3D-AbstractSpecification/Part01/components/shape.html#TwoSidedMaterial



Material fields

Color, transparency and shininess fields together make up Material properties. Examples follow.

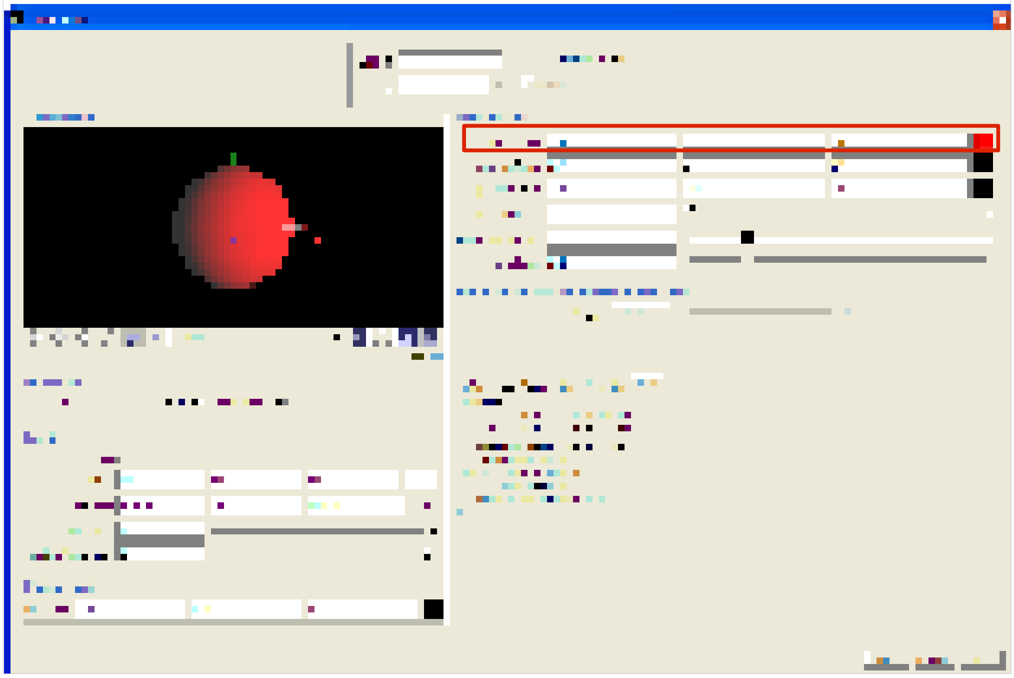
- *diffuseColor* reflects all X3D light sources, depending on viewing angles towards each light
- *ambientIntensity* is reflection multiplication factor
- *emissiveColor* is glowing component, normally off, independent of reflected light
- *specularColor* governs reflection highlights
- *shininess* controls specular intensity
- *transparency* is ability to see through an object: 1 is completely transparent, 0 is opaque

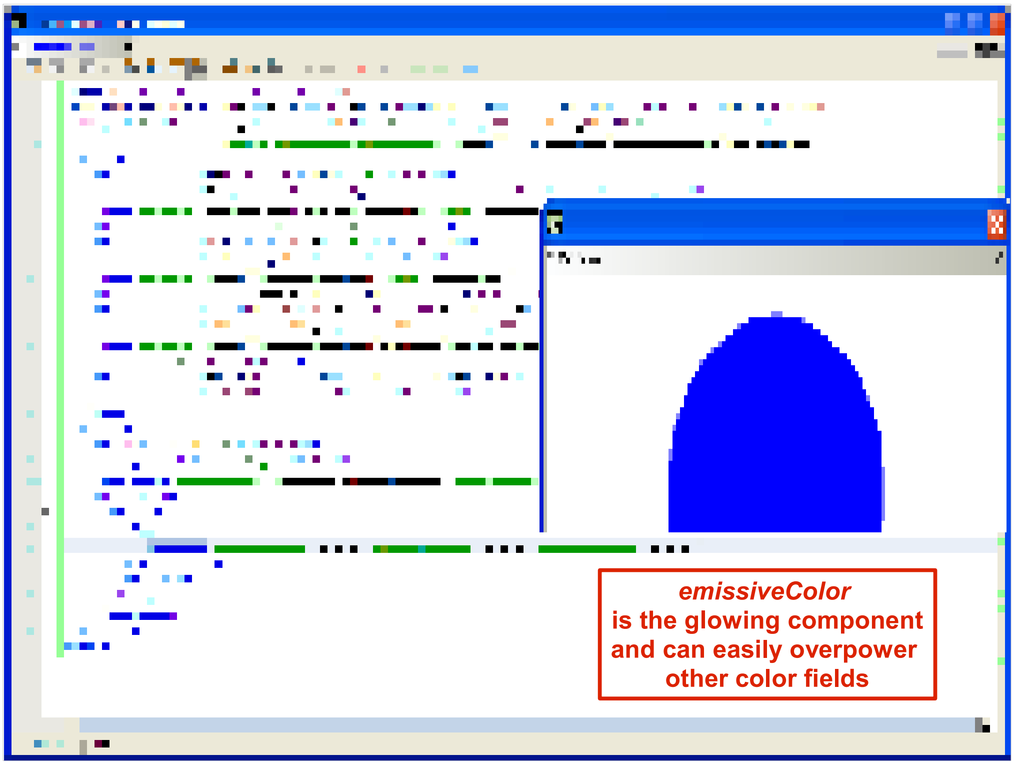


***diffuseColor* is the primary Material field**

http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/DiffuseColor.x3d

Material editor: *diffuseColor*





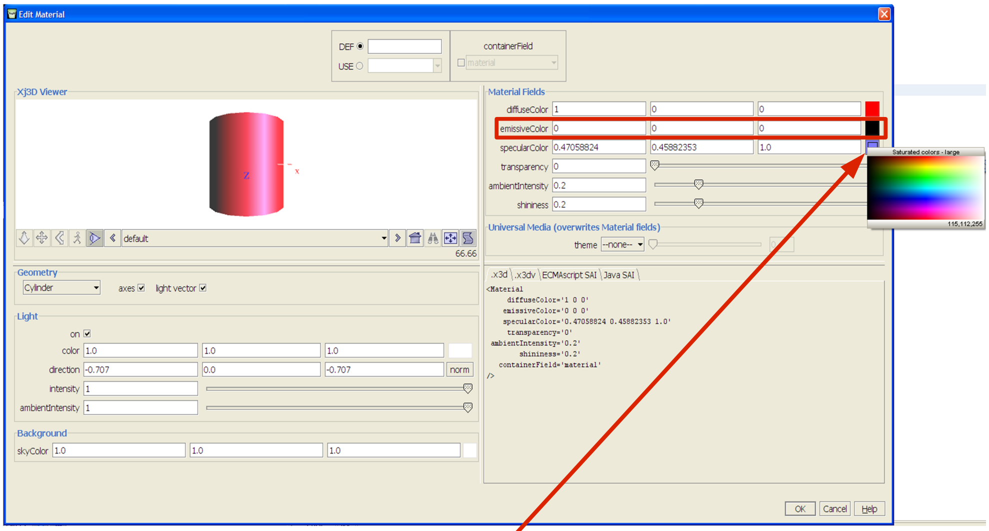
emissiveColor is the glowing component and can easily overpower other color fields

http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/EmissiveColor.x3d

Also note how all highlights are washed out, the sense of perspective provided by the shading of reflected light is completely lost.

Because of this side effect, emissiveColor should be used sparingly (if at all) and is usually reserved for visualizing energy or other special effects.

Material editor color selector



DEF ● USE ○

containerField □ material

Xj3D Viewer

default

66.66

Geometry

Cylinder axes ☑ light vector ☑

Light

on ☑			
color	1.0	1.0	1.0
direction	-0.707	0.0	-0.707
intensity	1		
ambientIntensity	1		

Background

skyColor	1.0	1.0	1.0

Material Fields

diffuseColor	1	0	0
emissiveColor	0	0	0
specularColor	0.47058824	0.45882353	1.0
transparency	0		
ambientIntensity	0.2		
shininess	0.2		

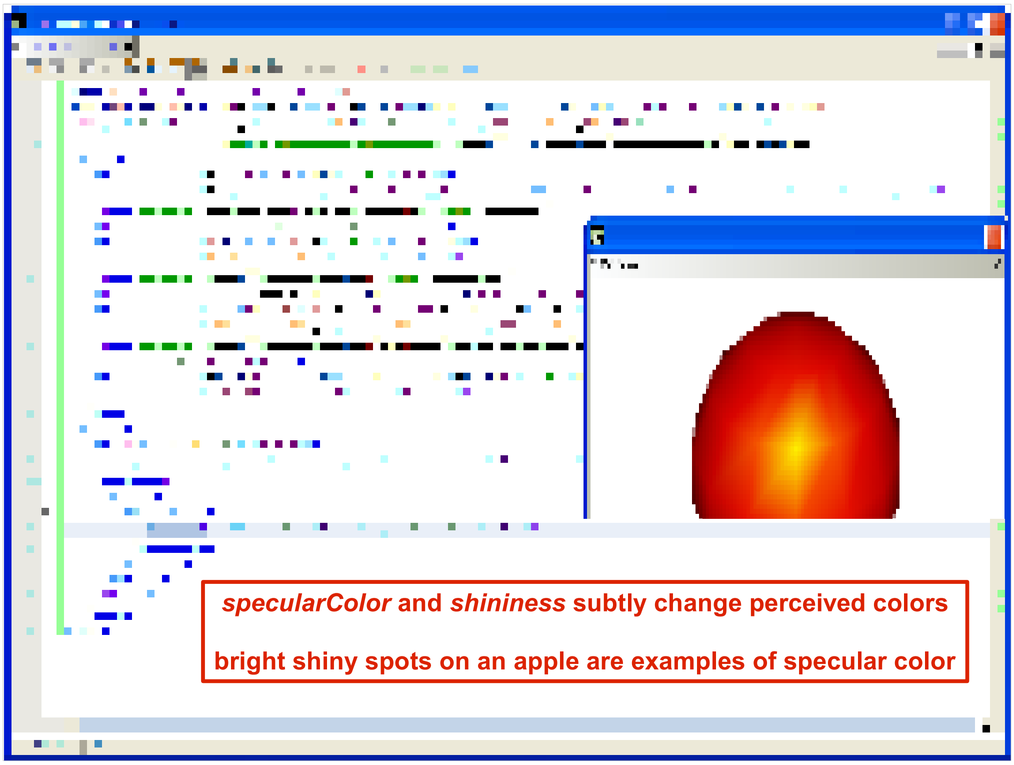
Universal Media (overwrites Material fields)

theme --none-- 0

.x3d .x3dv ECMAscript SAI Java SAI

```
<Material
     diffuseColor='1 0 0'
    emissiveColor='0 0 0'
    specularColor='0.47058824 0.45882353 1.0'
     transparency='0'
 ambientIntensity='0.2'
        shininess='0.2'
   containerField='material'
/>
```

Click colored box to select a color

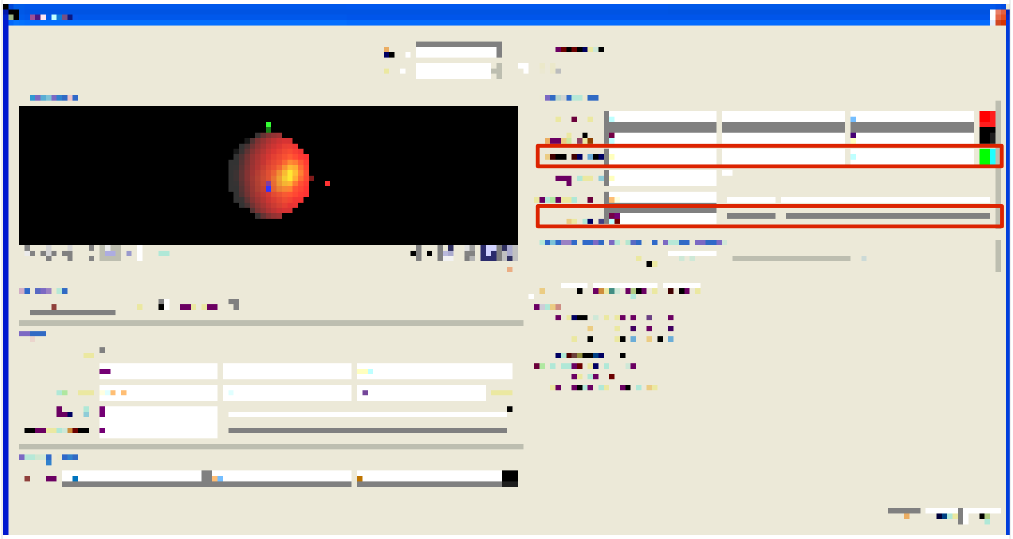


specularColor **and** ***shininess*** **subtly change perceived colors**

bright shiny spots on an apple are examples of specular color

http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/SpecularColor.x3d

Material editor *specularColor, shininess*



View of semitransparent shiny Sphere,
in front of nontransparent (opaque) Box,
in front of opaque Cylinder

http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/Transparency.x3d

Universal Media materials library

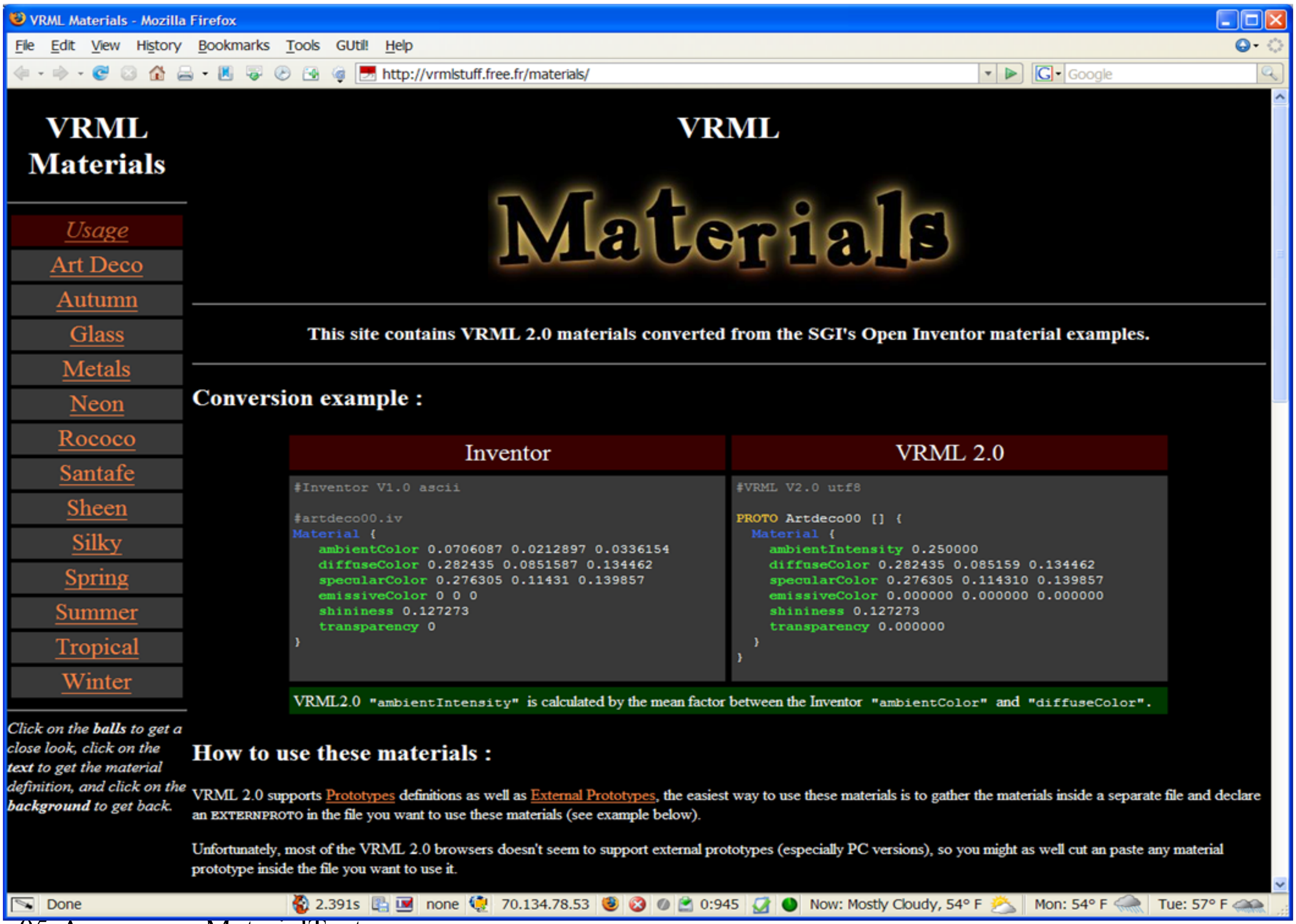
The Universal Media materials were originally created by SGI as part of OpenInventor in the 1990s as a convenience to authors

Each set of materials is grouped for visual compatibility and aesthetic appeal

Now converted and available for X3D use

- David Rousseau converted to VRML97
- Aaron Walsh created VRML Universal Media archive
- Don Brutzman translated into X3D as prototypes, cut/paste field values, also embedded in X3D-Edit
- http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials

David Rousseau's VRML site for these materials is http://vrmlstuff.free.fr/materials



*Click on the **balls** to get a close look, click on the **text** to get the material definition, and click on the **background** to get back.*

VRML

Materials

This site contains VRML 2.0 materials converted from the SGI's Open Inventor material examples.

Conversion example :

Inventor

```
#Inventor V1.0 ascii

#artdeco00.iv
Material {
   ambientColor 0.0706087 0.0212897 0.0336154
   diffuseColor 0.282435 0.0851587 0.134462
   specularColor 0.276305 0.11431 0.139857
   emissiveColor 0 0 0
   shininess 0.127273
   transparency 0
}
```

VRML 2.0

```
#VRML V2.0 utf8

PROTO Artdeco00 [] {
  Material {
    ambientIntensity 0.250000
    diffuseColor 0.282435 0.085159 0.134462
    specularColor 0.276305 0.114310 0.139857
    emissiveColor 0.000000 0.000000 0.000000
    shininess 0.127273
    transparency 0.000000
  }
}
```

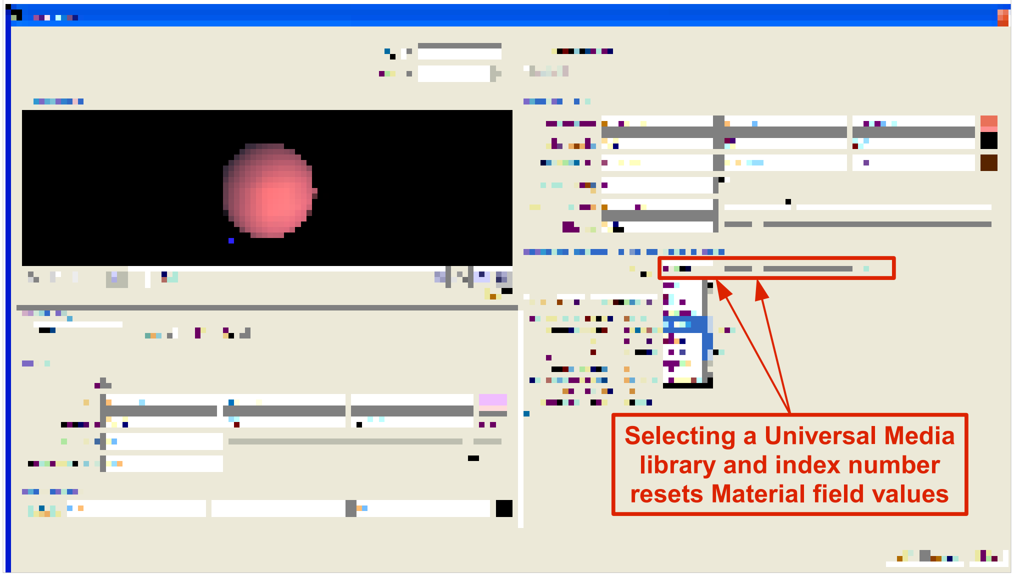
VRML2.0 `"ambientIntensity"` is calculated by the mean factor between the Inventor `"ambientColor"` and `"diffuseColor"`.

How to use these materials :

VRML 2.0 supports Prototypes definitions as well as External Prototypes, the easiest way to use these materials is to gather the materials inside a separate file and declare an EXTERNPROTO in the file you want to use these materials (see example below).

Unfortunately, most of the VRML 2.0 browsers doesn't seem to support external prototypes (especially PC versions), so you might as well cut an paste any material prototype inside the file you want to use it.

Selecting a Universal Material value



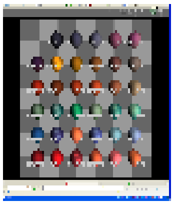
Selecting a Universal Media library and index number resets Material field values

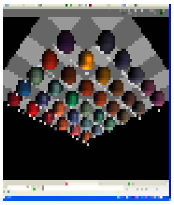
Universal Media Material libraries include

ArtDeco, Autumn, Glass, Metal, Neon, Rococo, SantaFe, Sheen, Silky, Spring, Summer, Tropical, Winter

http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials

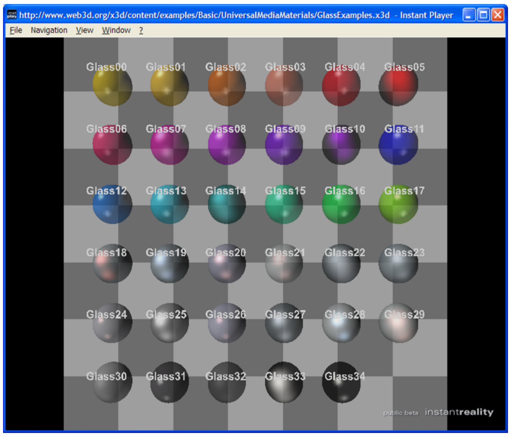
Universal Media screenshots





Universal Media Material library online

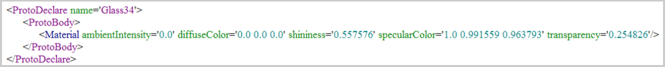
http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials



http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassExamples.x3d - Instant Player
File Navigation View Window ?
Glass00 Glass01 Glass02 Glass03 Glass04 Glass05
Glass06 Glass07 Glass08 Glass09 Glass10 Glass11
Glass12 Glass13 Glass14 Glass15 Glass16 Glass17
Glass18 Glass19 Glass20 Glass21 Glass22 Glass23
Glass24 Glass25 Glass26 Glass27 Glass28 Glass29
Glass30 Glass31 Glass32 Glass33 Glass34
public beta instantreality

- Example scenes good for browsing
- Click sphere for closeup view
- Click text for X3D material source
- Click grid to restore full view

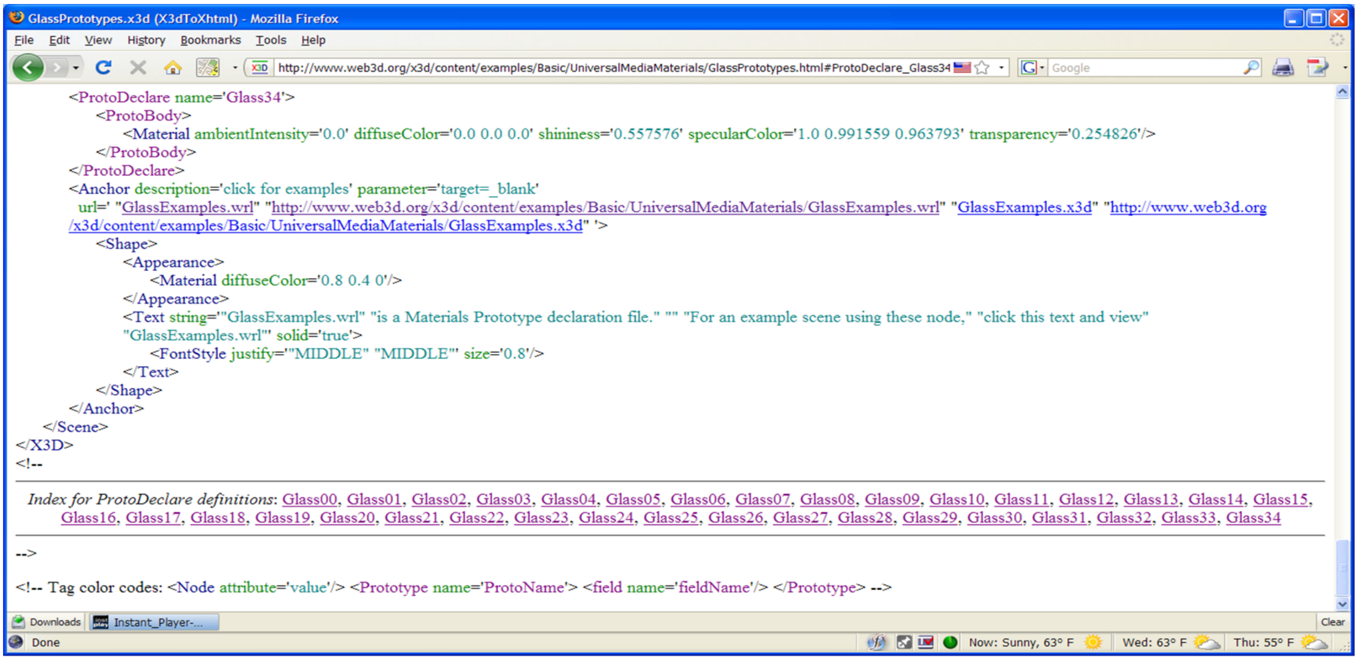
http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassExamples.x3d



```
<ProtoDeclare name='Glass34'>
    <ProtoBody>
        <Material ambientIntensity='0.0' diffuseColor='0.0 0.0 0.0' shininess='0.557576' specularColor='1.0 0.991559 0.963793' transparency='0.254826'/>
    </ProtoBody>
</ProtoDeclare>
```

Pretty-print HTML version of X3D material declaration is popped up when the user selects some text (such as “Glass 34”). Then the Material values can be copied/pasted into your X3D scene.

http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassPrototypes.html#ProtoDeclare_Glass34



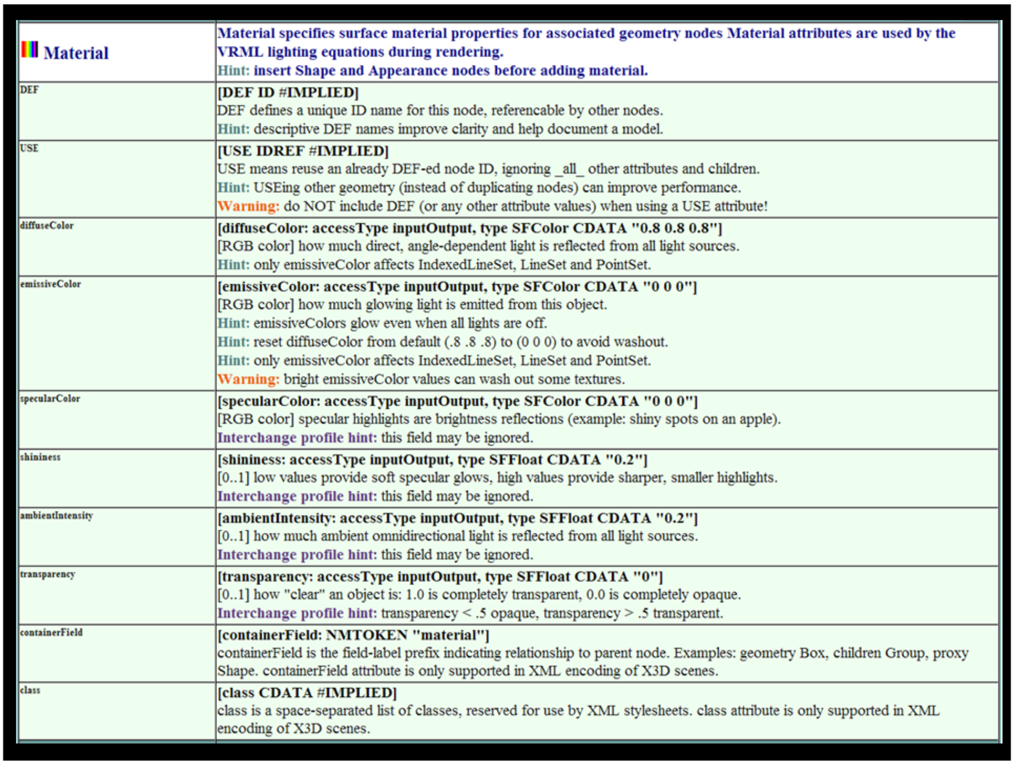
GlassPrototypes.x3d (X3dToXhtml) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassPrototypes.html#ProtoDeclare_Glass34
Google
Downloads
Instant_Player-...
Clear
Done
Now: Sunny, 63° F
Wed: 63° F
Thu: 55° F

```
      <ProtoDeclare name='Glass34'>
          <ProtoBody>
              <Material ambientIntensity='0.0' diffuseColor='0.0 0.0 0.0' shininess='0.557576' specularColor='1.0 0.991559 0.963793' transparency='0.254826'/>
          </ProtoBody>
      </ProtoDeclare>
      <Anchor description='click for examples' parameter='target=_blank'
        url=' "GlassExamples.wrl" "http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassExamples.wrl" "GlassExamples.x3d" "http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaMaterials/GlassExamples.x3d" '>
          <Shape>
              <Appearance>
                  <Material diffuseColor='0.8 0.4 0'/>
              </Appearance>
              <Text string='"GlassExamples.wrl" "is a Materials Prototype declaration file." "" "For an example scene using these node," "click this text and view"
              "GlassExamples.wrl"' solid='true'>
                  <FontStyle justify='"MIDDLE" "MIDDLE"' size='0.8'/>
              </Text>
          </Shape>
      </Anchor>
  </Scene>
</X3D>
<!--
```

Index for ProtoDeclare definitions: Glass00, Glass01, Glass02, Glass03, Glass04, Glass05, Glass06, Glass07, Glass08, Glass09, Glass10, Glass11, Glass12, Glass13, Glass14, Glass15, Glass16, Glass17, Glass18, Glass19, Glass20, Glass21, Glass22, Glass23, Glass24, Glass25, Glass26, Glass27, Glass28, Glass29, Glass30, Glass31, Glass32, Glass33, Glass34

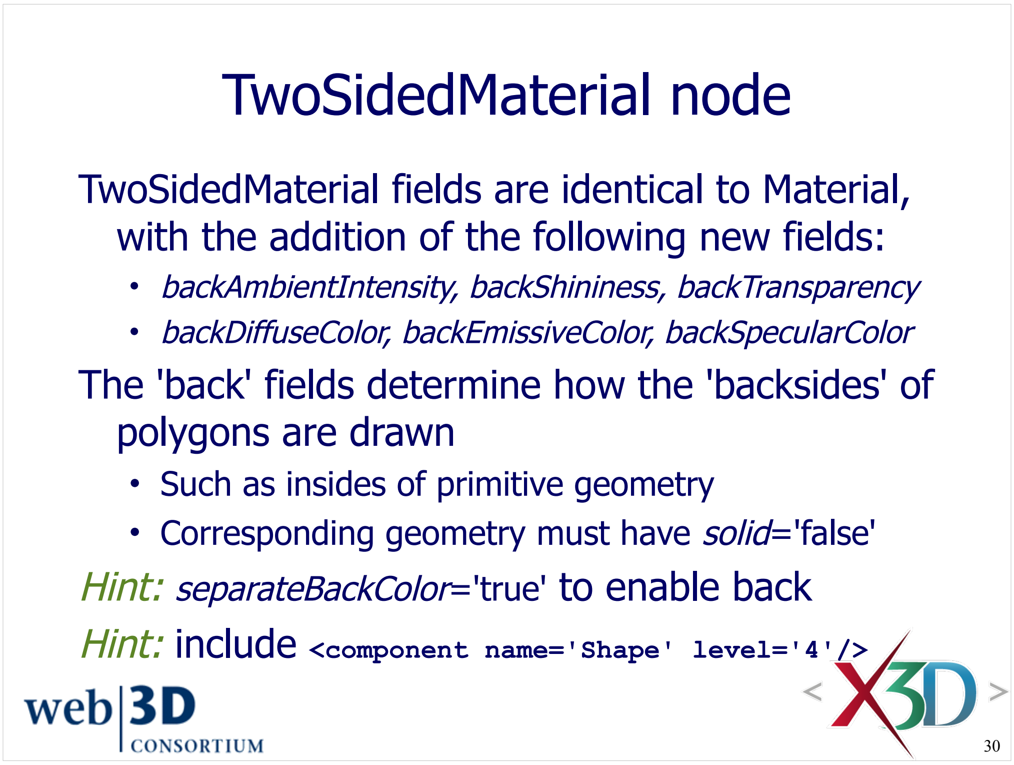
```
-->

<!-- Tag color codes: <Node attribute='value'/> <Prototype name='ProtoName'> <field name='fieldName'/> </Prototype> -->
```



Material	Material specifies surface material properties for associated geometry nodes Material attributes are used by the VRML lighting equations during rendering. Hint: insert Shape and Appearance nodes before adding material.
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!
diffuseColor	**[diffuseColor: accessType inputOutput, type SFColor CDATA "0.8 0.8 0.8"]** [RGB color] how much direct, angle-dependent light is reflected from all light sources. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet.
emissiveColor	**[emissiveColor: accessType inputOutput, type SFColor CDATA "0 0 0"]** [RGB color] how much glowing light is emitted from this object. Hint: emissiveColors glow even when all lights are off. Hint: reset diffuseColor from default (.8 .8 .8) to (0 0 0) to avoid washout. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet. Warning: bright emissiveColor values can wash out some textures.
specularColor	**[specularColor: accessType inputOutput, type SFColor CDATA "0 0 0"]** [RGB color] specular highlights are brightness reflections (example: shiny spots on an apple). Interchange profile hint: this field may be ignored.
shininess	**[shininess: accessType inputOutput, type SFFloat CDATA "0.2"]** [0..1] low values provide soft specular glows, high values provide sharper, smaller highlights. Interchange profile hint: this field may be ignored.
ambientIntensity	**[ambientIntensity: accessType inputOutput, type SFFloat CDATA "0.2"]** [0..1] how much ambient omnidirectional light is reflected from all light sources. Interchange profile hint: this field may be ignored.
transparency	**[transparency: accessType inputOutput, type SFFloat CDATA "0"]** [0..1] how "clear" an object is: 1.0 is completely transparent, 0.0 is completely opaque. Interchange profile hint: transparency < .5 opaque, transparency > .5 transparent.
containerField	**[containerField: NMTOKEN "material"]** 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#Material



TwoSidedMaterial node

TwoSidedMaterial fields are identical to Material, with the addition of the following new fields:

- *backAmbientIntensity, backShininess, backTransparency*
- *backDiffuseColor, backEmissiveColor, backSpecularColor*

The 'back' fields determine how the 'backsides' of polygons are drawn

- Such as insides of primitive geometry
- Corresponding geometry must have *solid*='false'

Hint: separateBackColor='true' to enable back

Hint: include `<component name='Shape' level='4'/>`

TwoSidedMaterial was introduced in X3D version 3.2.

http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification_Revision1_to_Part1/Part01/components/shape.html#TwoSidedMaterial

The front (and back) sides of a polygon are defined by normal direction of each polygon. For simple geometry primitives, these are simply the outside (and inside) respectively. For vertex-based polygons, the positive normal direction is defined using the right-hand rule: follow the points in order pointing from lower to higher using the curled fingers of the right hand. The right-hand thumb then points in the direction of the positive normal.

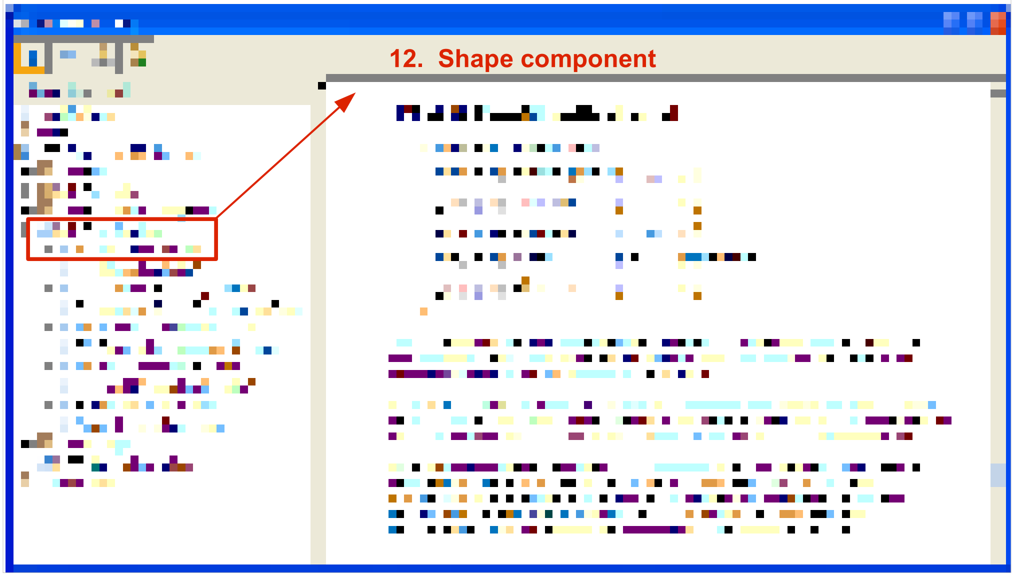
Normals are discussed further in Chapter 13, Triangles and Quadrilaterals.

Also note that many geometry nodes have a *solid* field which has an effect on TwoSidedMaterial.

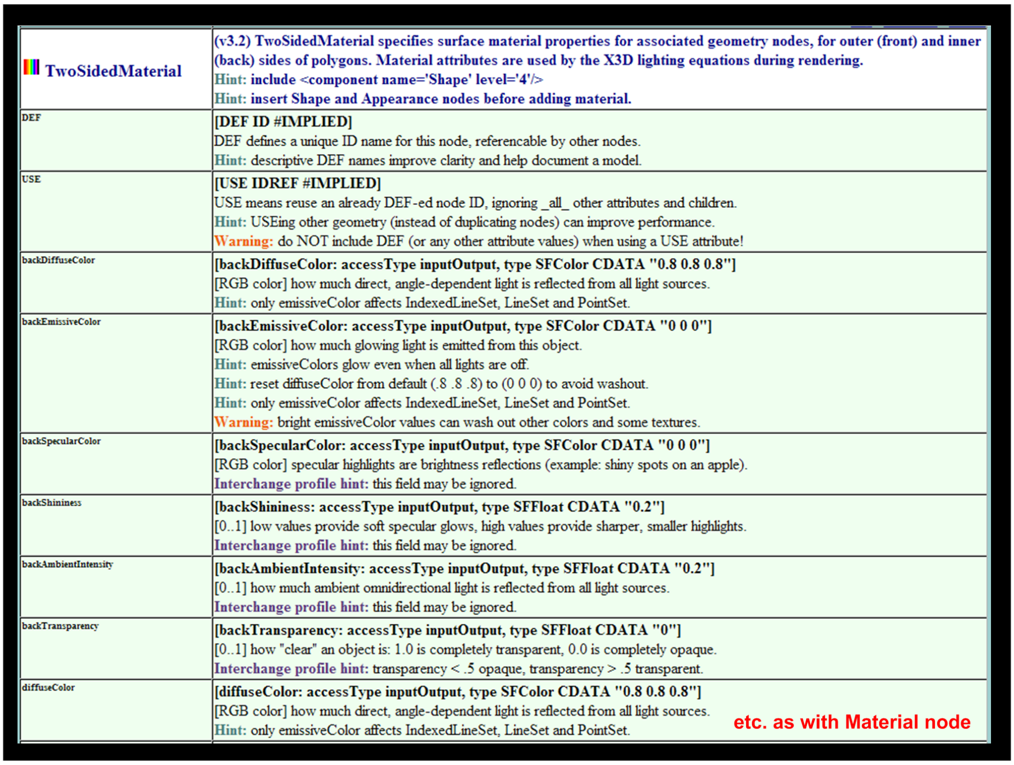
- *solid*='true' means that the object has one side and is considered solid (like a brick) with no inside faces to draw
- *solid*='false' means the object has two sides, both front (outer) faces and back (inner) faces. So for this case, TwoSidedMaterial is honored and used.

Advanced hint: *separateBackColor* presents interesting opportunities for animation.

TwoSidedMaterial specification help entry in X3D-Edit



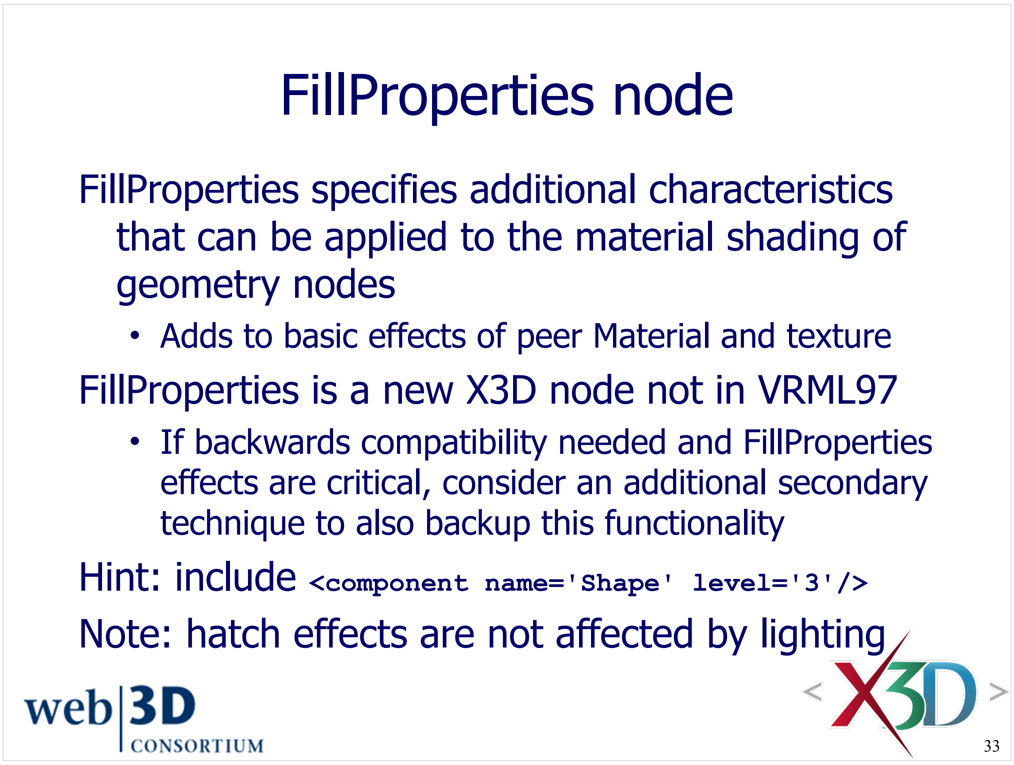
F1 is the X3D-Edit hot key to invoke the JavaHelp system



TwoSidedMaterial	(v3.2) TwoSidedMaterial specifies surface material properties for associated geometry nodes, for outer (front) and inner (back) sides of polygons. Material attributes are used by the X3D lighting equations during rendering. Hint: include <component name='Shape' level='4'/> Hint: insert Shape and Appearance nodes before adding material.
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!
backDiffuseColor	**[backDiffuseColor: accessType inputOutput, type SFColor CDATA "0.8 0.8 0.8"]** [RGB color] how much direct, angle-dependent light is reflected from all light sources. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet.
backEmissiveColor	**[backEmissiveColor: accessType inputOutput, type SFColor CDATA "0 0 0"]** [RGB color] how much glowing light is emitted from this object. Hint: emissiveColors glow even when all lights are off. Hint: reset diffuseColor from default (.8 .8 .8) to (0 0 0) to avoid washout. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet. Warning: bright emissiveColor values can wash out other colors and some textures.
backSpecularColor	**[backSpecularColor: accessType inputOutput, type SFColor CDATA "0 0 0"]** [RGB color] specular highlights are brightness reflections (example: shiny spots on an apple). Interchange profile hint: this field may be ignored.
backShininess	**[backShininess: accessType inputOutput, type SFFloat CDATA "0.2"]** [0..1] low values provide soft specular glows, high values provide sharper, smaller highlights. Interchange profile hint: this field may be ignored.
backAmbientIntensity	**[backAmbientIntensity: accessType inputOutput, type SFFloat CDATA "0.2"]** [0..1] how much ambient omnidirectional light is reflected from all light sources. Interchange profile hint: this field may be ignored.
backTransparency	**[backTransparency: accessType inputOutput, type SFFloat CDATA "0"]** [0..1] how "clear" an object is: 1.0 is completely transparent, 0.0 is completely opaque. Interchange profile hint: transparency < .5 opaque, transparency > .5 transparent.
diffuseColor	**[diffuseColor: accessType inputOutput, type SFColor CDATA "0.8 0.8 0.8"]** [RGB color] how much direct, angle-dependent light is reflected from all light sources. Hint: only emissiveColor affects IndexedLineSet, LineSet and PointSet.

etc. as with Material node

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



FillProperties node

FillProperties specifies additional characteristics that can be applied to the material shading of geometry nodes

- Adds to basic effects of peer Material and texture

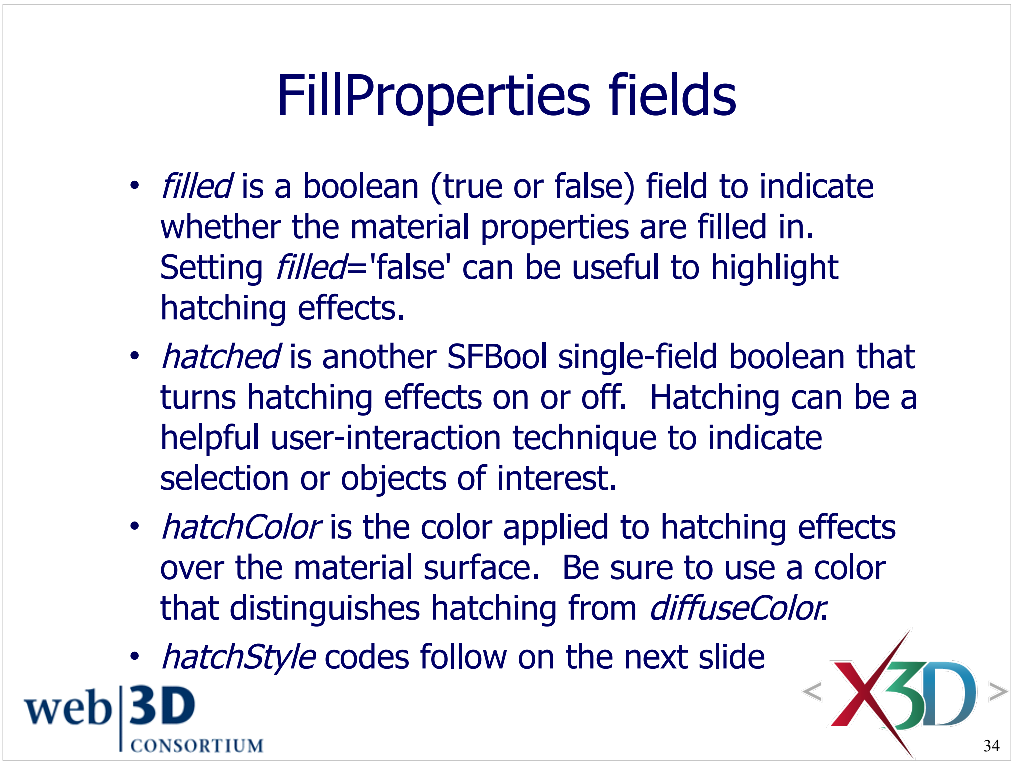
FillProperties is a new X3D node not in VRML97

- If backwards compatibility needed and FillProperties effects are critical, consider an additional secondary technique to also backup this functionality

Hint: include `<component name='Shape' level='3'/>`

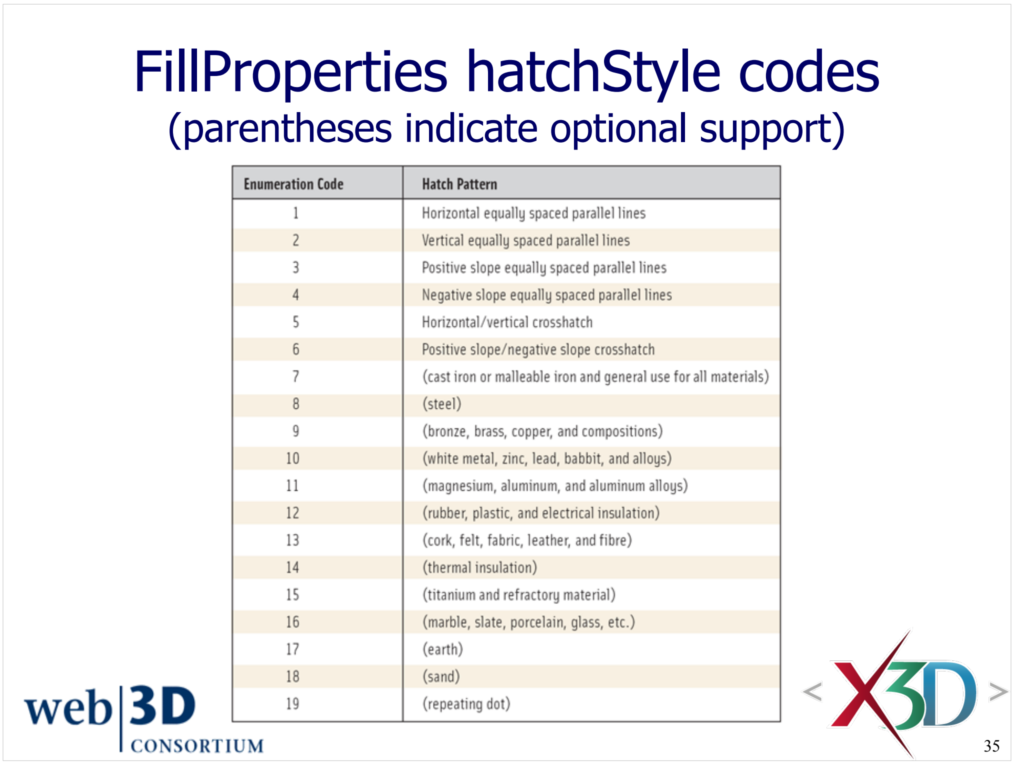
Note: hatch effects are not affected by lighting

Note: <component> declarations precede <meta> statements inside the scene <head> section



FillProperties fields

- *filled* is a boolean (true or false) field to indicate whether the material properties are filled in. Setting *filled*='false' can be useful to highlight hatching effects.
- *hatched* is another SFBool single-field boolean that turns hatching effects on or off. Hatching can be a helpful user-interaction technique to indicate selection or objects of interest.
- *hatchColor* is the color applied to hatching effects over the material surface. Be sure to use a color that distinguishes hatching from *diffuseColor*.
- *hatchStyle* codes follow on the next slide



FillProperties hatchStyle codes

(parentheses indicate optional support)

Enumeration Code	Hatch Pattern
1	Horizontal equally spaced parallel lines
2	Vertical equally spaced parallel lines
3	Positive slope equally spaced parallel lines
4	Negative slope equally spaced parallel lines
5	Horizontal/vertical crosshatch
6	Positive slope/negative slope crosshatch
7	(cast iron or malleable iron and general use for all materials)
8	(steel)
9	(bronze, brass, copper, and compositions)
10	(white metal, zinc, lead, babbit, and alloys)
11	(magnesium, aluminum, and aluminum alloys)
12	(rubber, plastic, and electrical insulation)
13	(cork, felt, fabric, leather, and fibre)
14	(thermal insulation)
15	(titanium and refractory material)
16	(marble, slate, porcelain, glass, etc.)
17	(earth)
18	(sand)
19	(repeating dot)

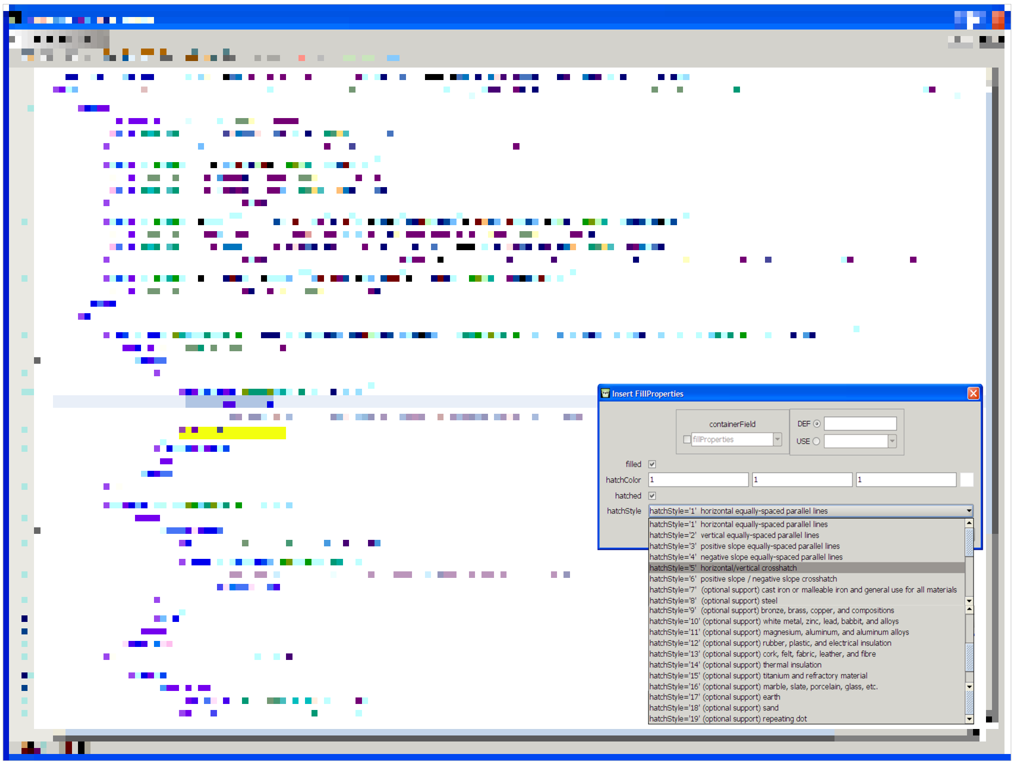
X3D for Web Authors, Table 5.8, p.136

ISO International Register of Graphical Items, Registration authority — National Imagery and Mapping Agency, c/o Joint Interoperability Test Command, Building 57305, Room 263A, Fort Huachuca, Arizona 85613-7020. USA.

http://jitc.fhu.disa.mil/nitf/graph_reg/graph_reg.html

Hatchstyle Section

http://jitc.fhu.disa.mil/nitf/graph_reg/class_pages/hatchstyle.html



Insert FillProperties

containerField ☐ fillProperties

DEF ●

USE ○

filled ☑

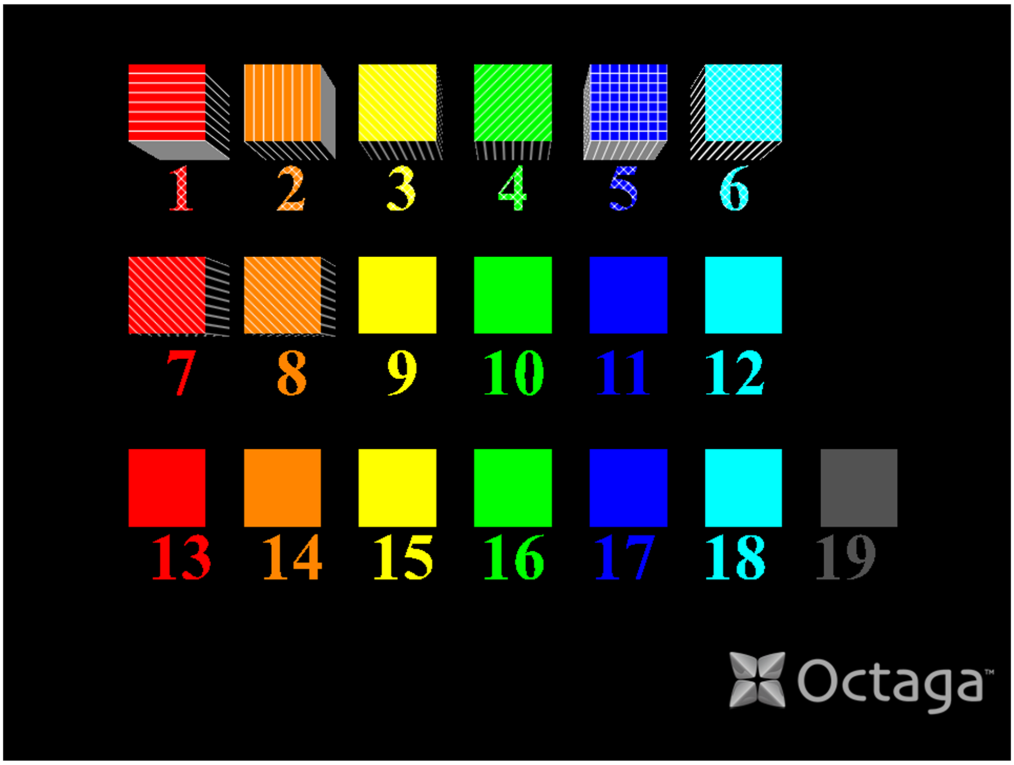
hatchColor 1 1 1

hatched ☑

hatchStyle hatchStyle='1' horizontal equally-spaced parallel lines

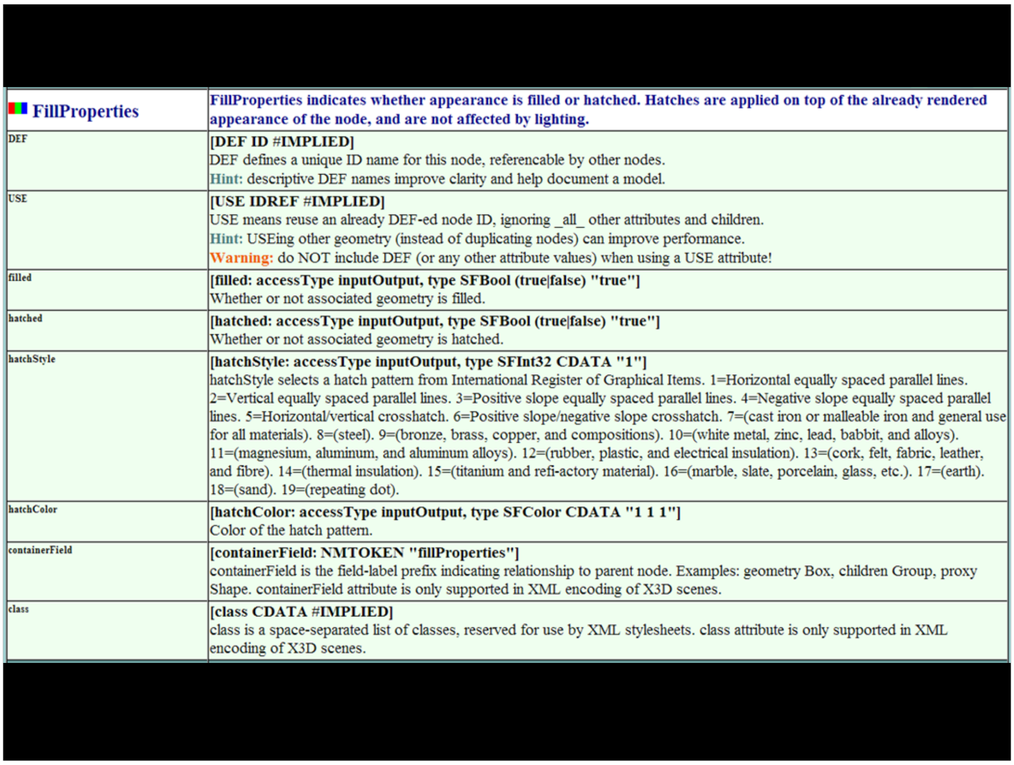
- hatchStyle='1' horizontal equally-spaced parallel lines
- hatchStyle='2' vertical equally-spaced parallel lines
- hatchStyle='3' positive slope equally-spaced parallel lines
- hatchStyle='4' negative slope equally-spaced parallel lines
- hatchStyle='5' horizontal/vertical crosshatch
- hatchStyle='6' positive slope / negative slope crosshatch
- hatchStyle='7' (optional support) cast iron or malleable iron and general use for all materials
- hatchStyle='8' (optional support) steel
- hatchStyle='9' (optional support) bronze, brass, copper, and compositions
- hatchStyle='10' (optional support) white metal, zinc, lead, babbit, and alloys
- hatchStyle='11' (optional support) magnesium, aluminum, and aluminum alloys
- hatchStyle='12' (optional support) rubber, plastic, and electrical insulation
- hatchStyle='13' (optional support) cork, felt, fabric, leather, and fibre
- hatchStyle='14' (optional support) thermal insulation
- hatchStyle='15' (optional support) titanium and refractory material
- hatchStyle='16' (optional support) marble, slate, porcelain, glass, etc.
- hatchStyle='17' (optional support) earth
- hatchStyle='18' (optional support) sand
- hatchStyle='19' (optional support) repeating dot

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/FillProperties.x3d



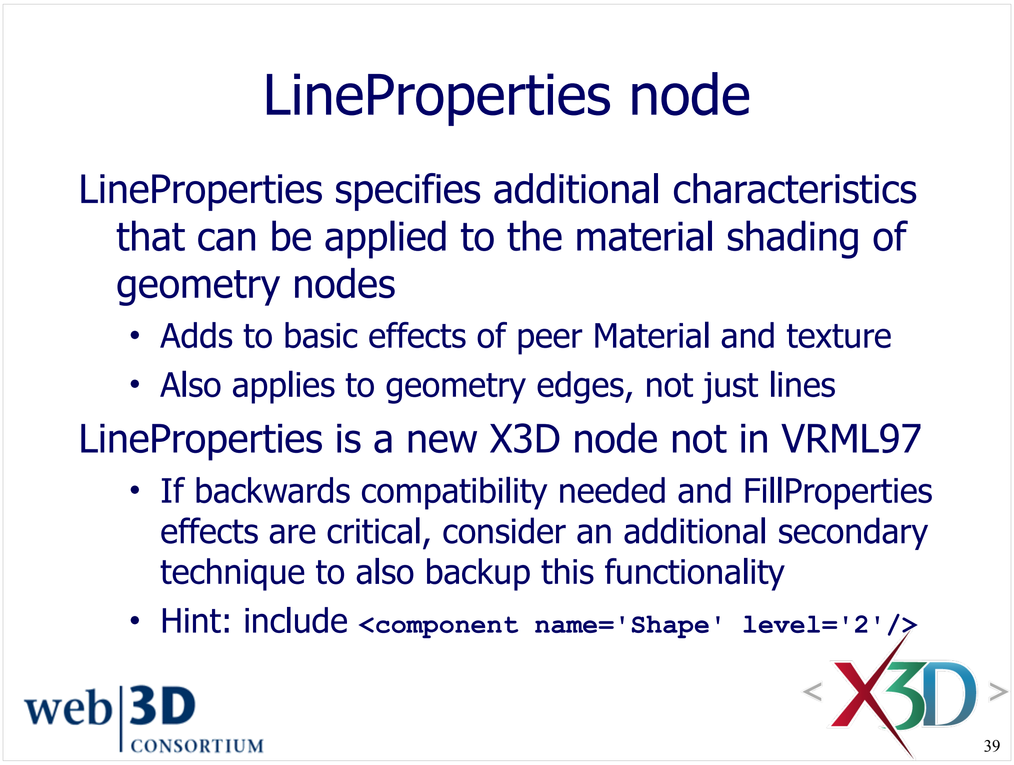
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/FillProperties.x3d

Warning: currently support for FillProperties is poor for many browsers. This screen snapshot shows that the optional hatchStyle values are not supported by this browser.



FillProperties	**FillProperties indicates whether appearance is filled or hatched. Hatches are applied on top of the already rendered appearance of the node, and are not affected by lighting.**
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!
filled	**[filled: accessType inputOutput, type SFBool (true|false) "true"]** Whether or not associated geometry is filled.
hatched	**[hatched: accessType inputOutput, type SFBool (true|false) "true"]** Whether or not associated geometry is hatched.
hatchStyle	**[hatchStyle: accessType inputOutput, type SFInt32 CDATA "1"]** hatchStyle selects a hatch pattern from International Register of Graphical Items. 1=Horizontal equally spaced parallel lines. 2=Vertical equally spaced parallel lines. 3=Positive slope equally spaced parallel lines. 4=Negative slope equally spaced parallel lines. 5=Horizontal/vertical crosshatch. 6=Positive slope/negative slope crosshatch. 7=(cast iron or malleable iron and general use for all materials). 8=(steel). 9=(bronze, brass, copper, and compositions). 10=(white metal, zinc, lead, babbit, and alloys). 11=(magnesium, aluminum, and aluminum alloys). 12=(rubber, plastic, and electrical insulation). 13=(cork, felt, fabric, leather, and fibre). 14=(thermal insulation). 15=(titanium and refi-actory material). 16=(marble, slate, porcelain, glass, etc.). 17=(earth). 18=(sand). 19=(repeating dot).
hatchColor	**[hatchColor: accessType inputOutput, type SFColor CDATA "1 1 1"]** Color of the hatch pattern.
containerField	**[containerField: NMTOKEN "fillProperties"]** 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#FillProperties



LineProperties node

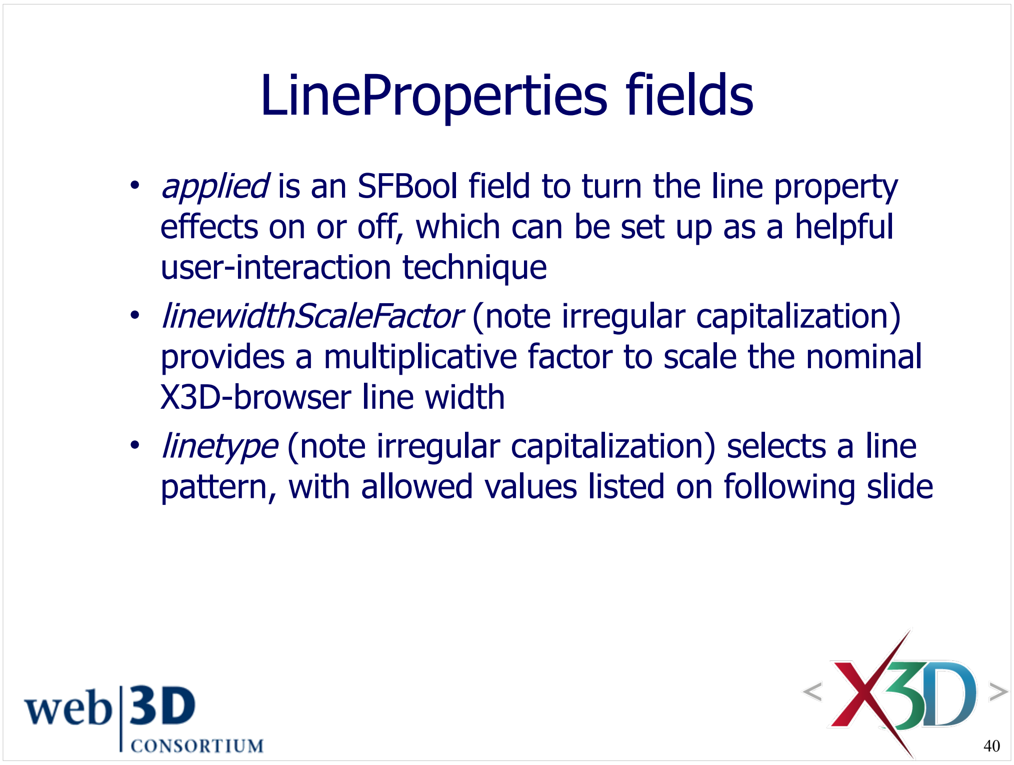
LineProperties specifies additional characteristics that can be applied to the material shading of geometry nodes

- Adds to basic effects of peer Material and texture
- Also applies to geometry edges, not just lines

LineProperties is a new X3D node not in VRML97

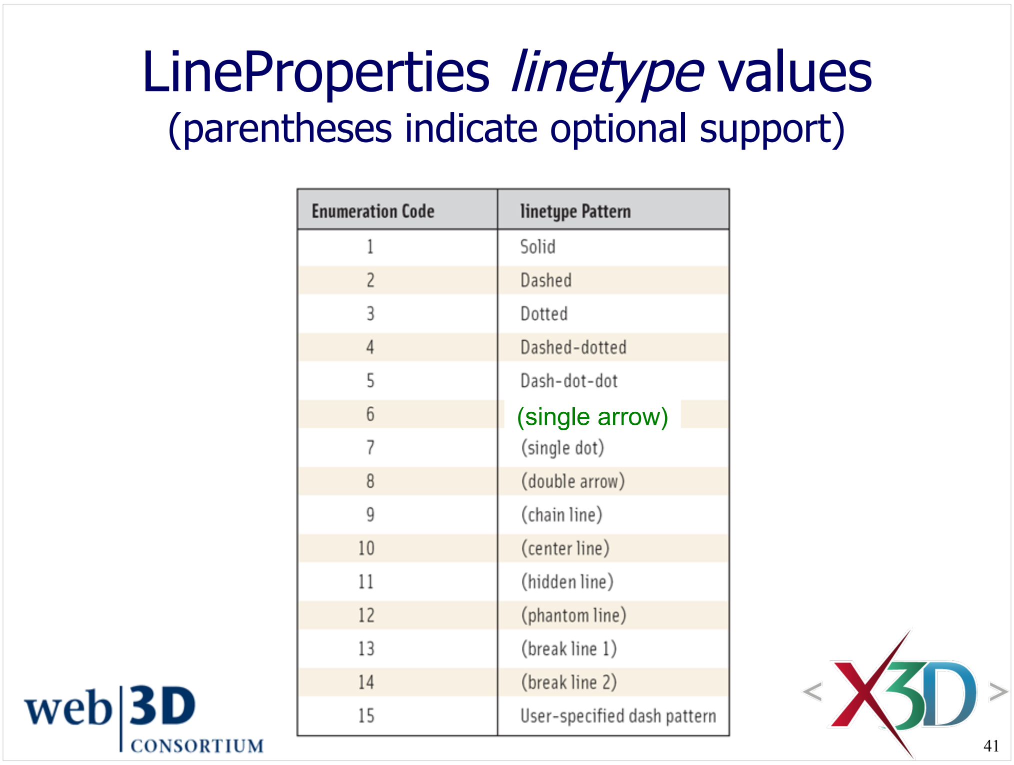
- If backwards compatibility needed and FillProperties effects are critical, consider an additional secondary technique to also backup this functionality
- Hint: include `<component name='Shape' level='2'/>`

Note: <component> declarations precede <meta> statements inside the scene <head> section



LineProperties fields

- *applied* is an SFBool field to turn the line property effects on or off, which can be set up as a helpful user-interaction technique
- *linewidthScaleFactor* (note irregular capitalization) provides a multiplicative factor to scale the nominal X3D-browser line width
- *linetype* (note irregular capitalization) selects a line pattern, with allowed values listed on following slide



LineProperties *linetype* values

(parentheses indicate optional support)

Enumeration Code	linetype Pattern
1	Solid
2	Dashed
3	Dotted
4	Dashed-dotted
5	Dash-dot-dot
6	(single arrow)
7	(single dot)
8	(double arrow)
9	(chain line)
10	(center line)
11	(hidden line)
12	(phantom line)
13	(break line 1)
14	(break line 2)
15	User-specified dash pattern

41

X3D for Web Authors, Table 5.11, p.138

X3D specification, Table 12.2 — International registry of graphical items linetypes

http://www.web3d.org/x3d/specifications/ISO-IEC-FDIS-19775-1.2-X3D-AbstractSpecification/Part01/components/shape.html#t-Linetypes

ISO International Register of Graphical Items, Registration authority — National Imagery and Mapping Agency, c/o Joint Interoperability Test Command, Building 57305, Room 263A, Fort Huachuca, Arizona 85613-7020. USA.

http://jitc.fhu.disa.mil/nitf/graph_reg/graph_reg.html

Linetype Section

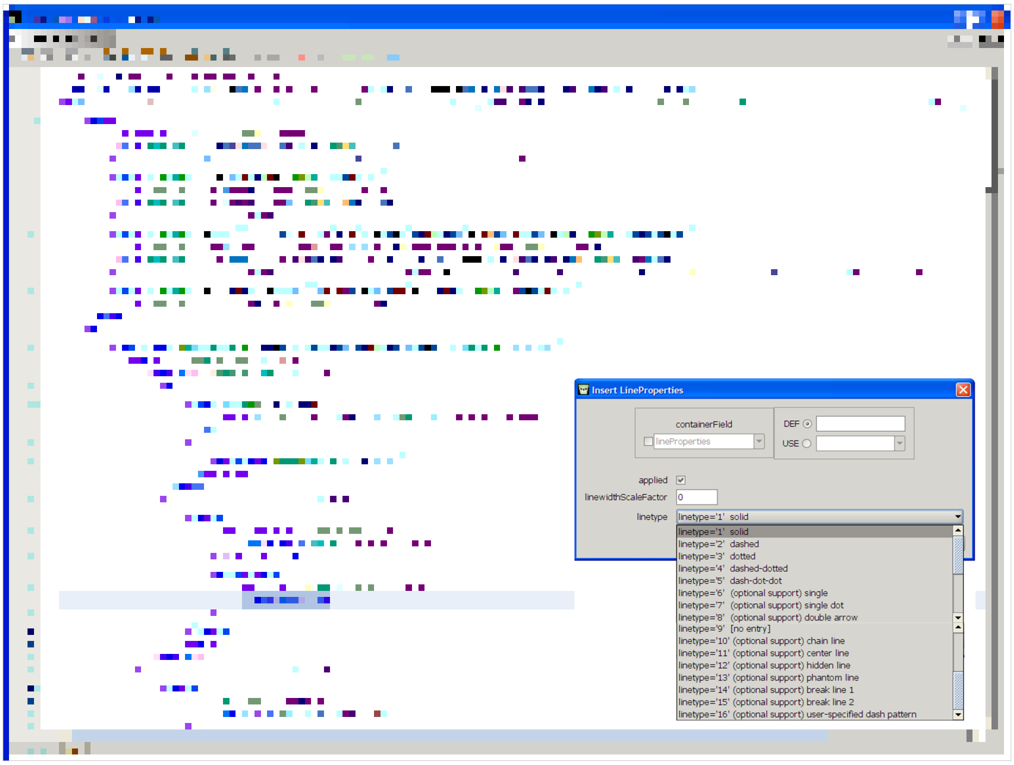
http://jitc.fhu.disa.mil/nitf/graph_reg/class_pages/linetype.html

The linetype erratum for enumeration code 6 was found and reported by a member of the X3D Working Group in August 2009. You can report specification comments and issues via online feedback forms available at

http://www.web3d.org/x3d/specifications/#x3dreporting

Book errata are tracked at

http://x3dgraphics.com/errata.php



containerField □ lineProperties

DEF ●

USE ○

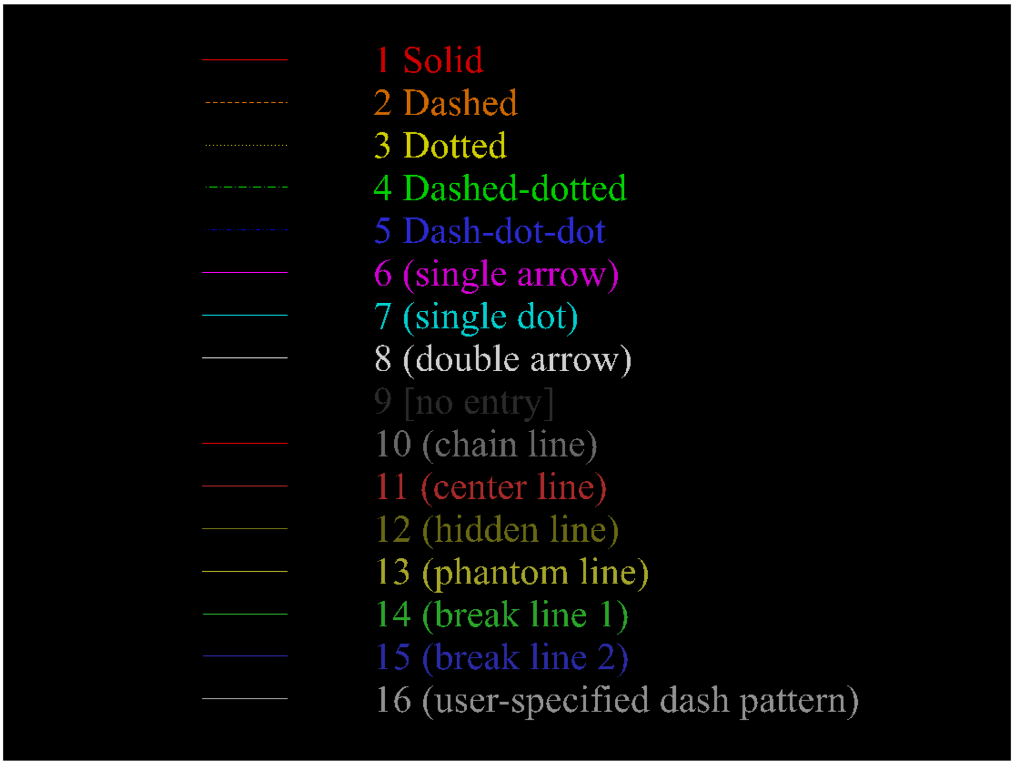
applied ☑

linewidthScaleFactor 0

linetype linetype='1' solid

linetype='1' solid
linetype='2' dashed
linetype='3' dotted
linetype='4' dashed-dotted
linetype='5' dash-dot-dot
linetype='6' (optional support) single
linetype='7' (optional support) single dot
linetype='8' (optional support) double arrow
linetype='9' [no entry]
linetype='10' (optional support) chain line
linetype='11' (optional support) center line
linetype='12' (optional support) hidden line
linetype='13' (optional support) phantom line
linetype='14' (optional support) break line 1
linetype='15' (optional support) break line 2
linetype='16' (optional support) user-specified dash pattern

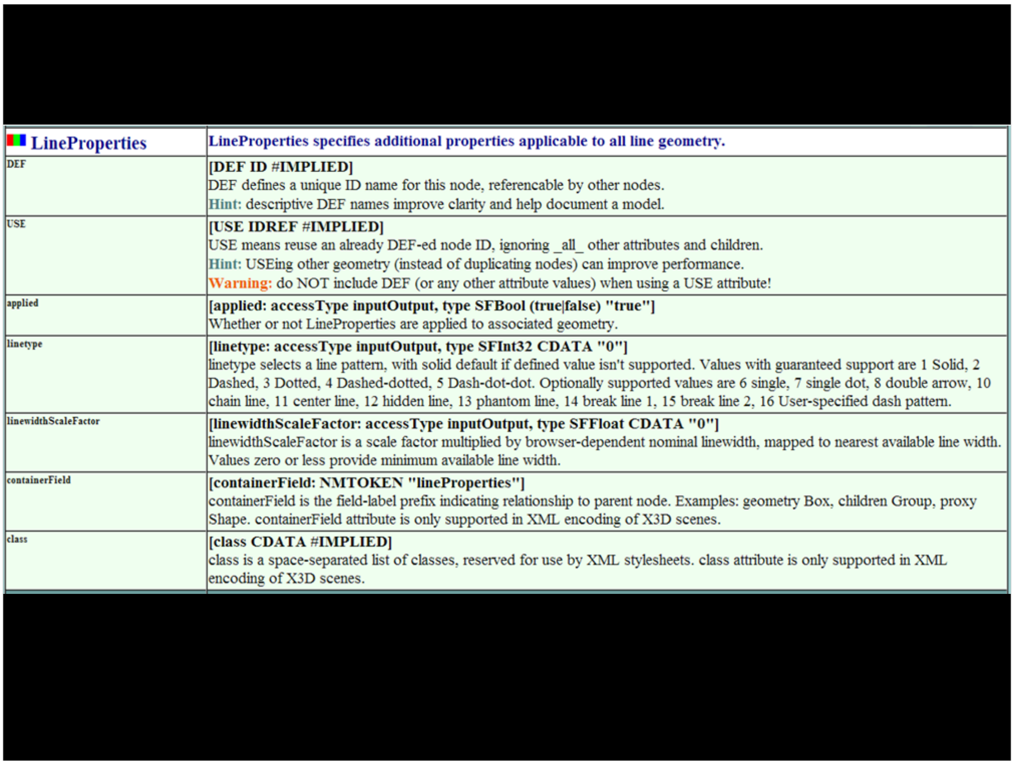
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/LineProperties.x3d



1 Solid
2 Dashed
3 Dotted
4 Dashed-dotted
5 Dash-dot-dot
6 (single arrow)
7 (single dot)
8 (double arrow)
9 [no entry]
10 (chain line)
11 (center line)
12 (hidden line)
13 (phantom line)
14 (break line 1)
15 (break line 2)
16 (user-specified dash pattern)

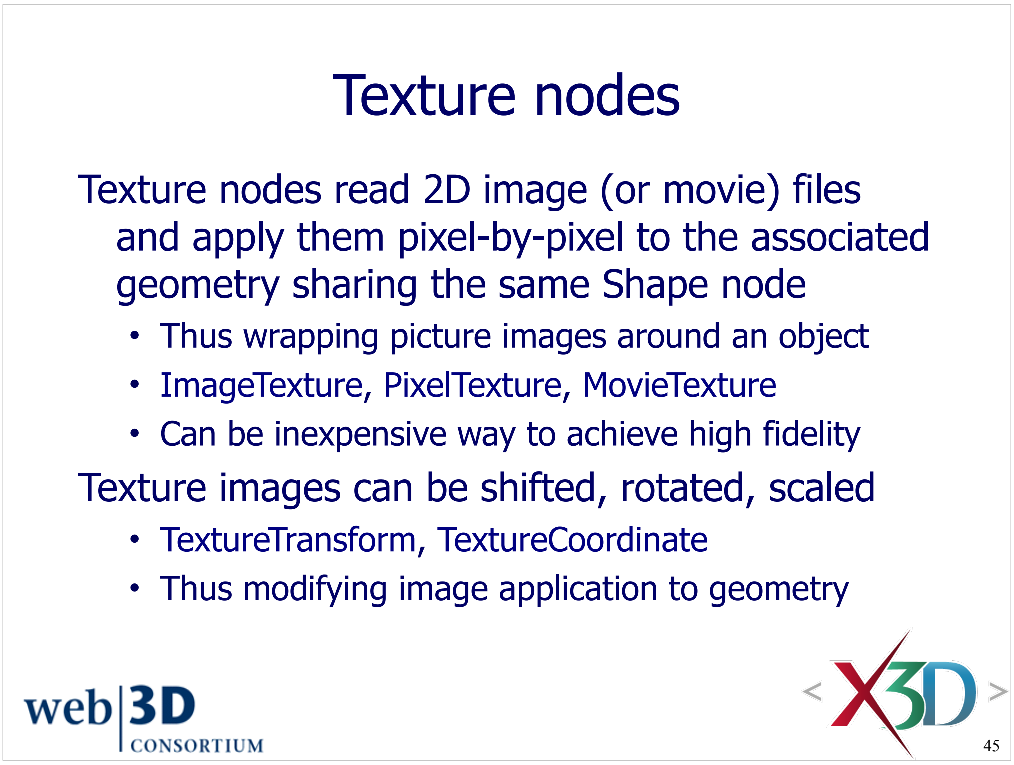
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/LineProperties.x3d

Warning: currently support for LineProperties is poor for many browsers.



LineProperties	LineProperties specifies additional properties applicable to all line geometry.
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!
applied	**[applied: accessType inputOutput, type SFBool (true|false) "true"]** Whether or not LineProperties are applied to associated geometry.
linetype	**[linetype: accessType inputOutput, type SFInt32 CDATA "0"]** linetype selects a line pattern, with solid default if defined value isn't supported. Values with guaranteed support are 1 Solid, 2 Dashed, 3 Dotted, 4 Dashed-dotted, 5 Dash-dot-dot. Optionally supported values are 6 single, 7 single dot, 8 double arrow, 10 chain line, 11 center line, 12 hidden line, 13 phantom line, 14 break line 1, 15 break line 2, 16 User-specified dash pattern.
linewidthScaleFactor	**[linewidthScaleFactor: accessType inputOutput, type SFFloat CDATA "0"]** linewidthScaleFactor is a scale factor multiplied by browser-dependent nominal linewidth, mapped to nearest available line width. Values zero or less provide minimum available line width.
containerField	**[containerField: NMTOKEN "lineProperties"]** 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#LineProperties



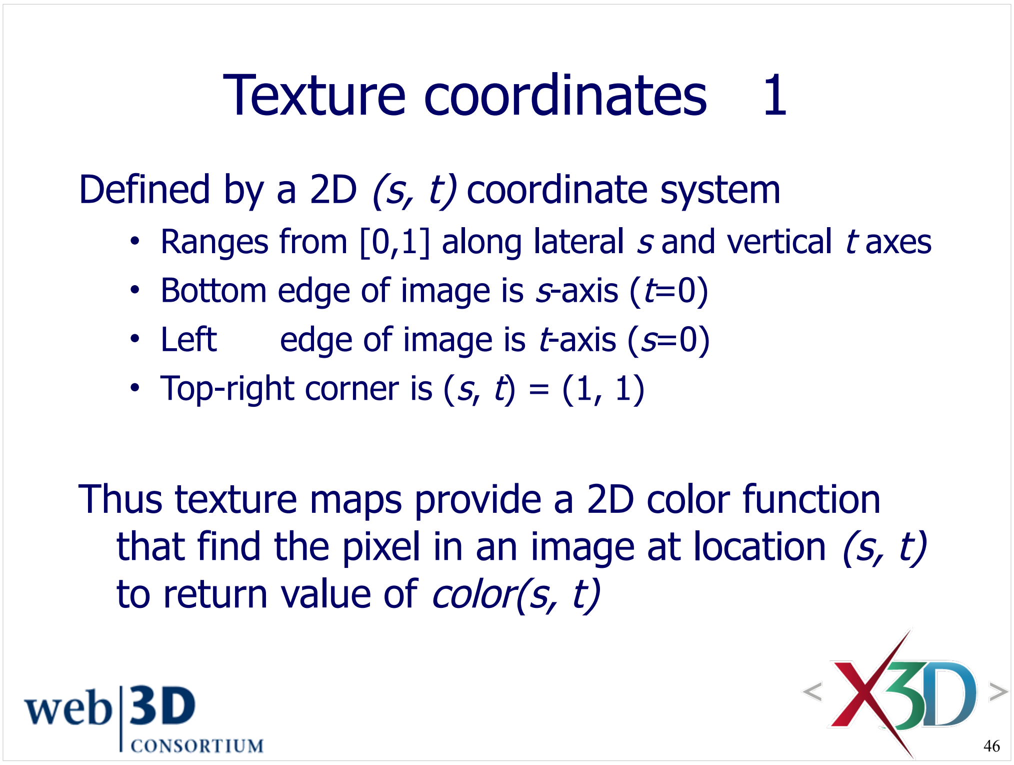
Texture nodes

Texture nodes read 2D image (or movie) files and apply them pixel-by-pixel to the associated geometry sharing the same Shape node

- Thus wrapping picture images around an object
- ImageTexture, PixelTexture, MovieTexture
- Can be inexpensive way to achieve high fidelity

Texture images can be shifted, rotated, scaled

- TextureTransform, TextureCoordinate
- Thus modifying image application to geometry



Texture coordinates 1

Defined by a 2D *(s, t)* coordinate system

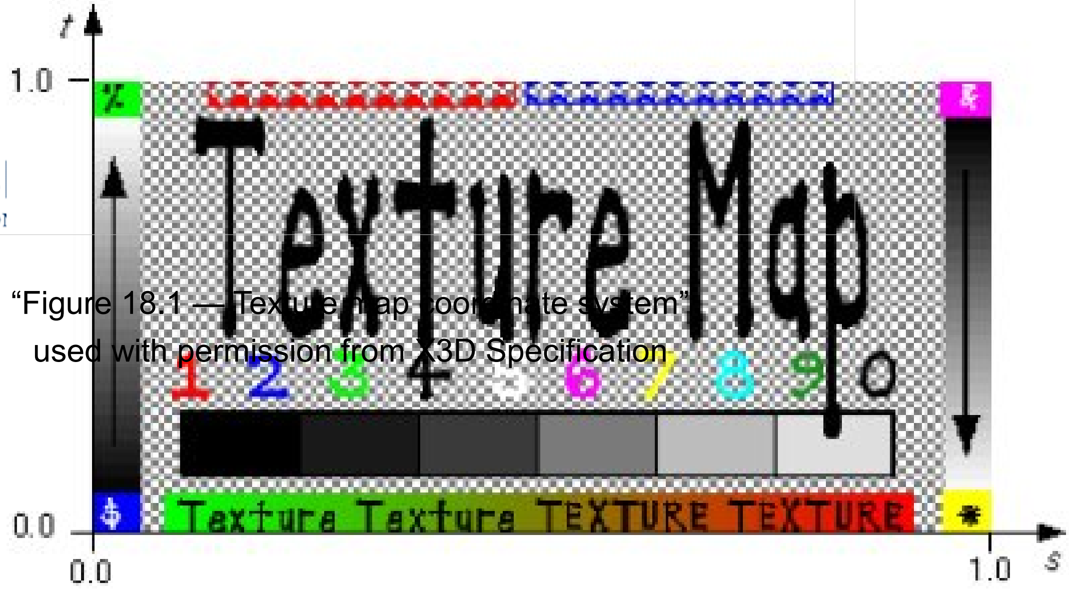
- Ranges from [0,1] along lateral *s* and vertical *t* axes
- Bottom edge of image is *s*-axis (*t*=0)
- Left edge of image is *t*-axis (*s*=0)
- Top-right corner is (*s*, *t*) = (1, 1)

Thus texture maps provide a 2D color function that find the pixel in an image at location *(s, t)* to return value of *color(s, t)*

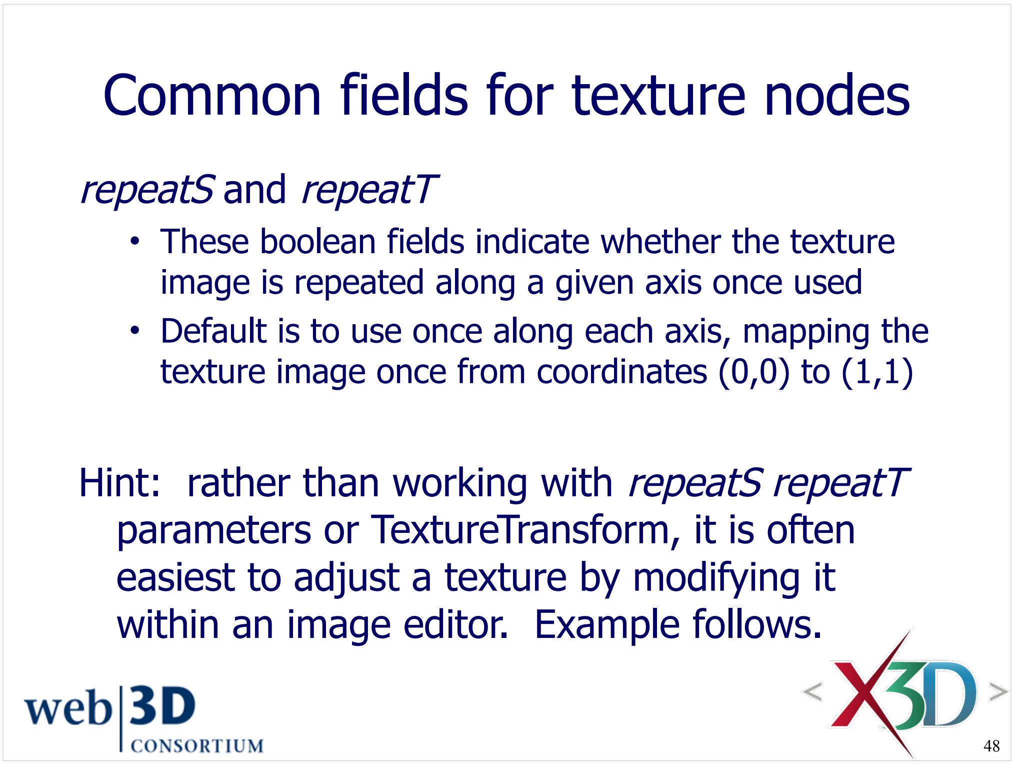
Texture coordinates 2

s and *t* coordinates locate each pixel in an image

- Thus texture coordinates work independently of either file size (bytes), image size (pixel count) or aspect ratio (width:height)



“Figure 18.1 — Texture map coordinate system”
used with permission from X3D Specification



Common fields for texture nodes

repeatS and *repeatT*

- These boolean fields indicate whether the texture image is repeated along a given axis once used
- Default is to use once along each axis, mapping the texture image once from coordinates (0,0) to (1,1)

Hint: rather than working with *repeatS repeatT* parameters or TextureTransform, it is often easiest to adjust a texture by modifying it within an image editor. Example follows.

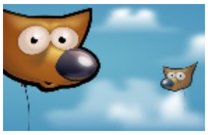
Image file manipulation tools

Many tools are available for manipulating images, sometimes provided with the operating system

- Adobe Photoshop
- Microsoft Visio, Paint, Keynote (Mac)

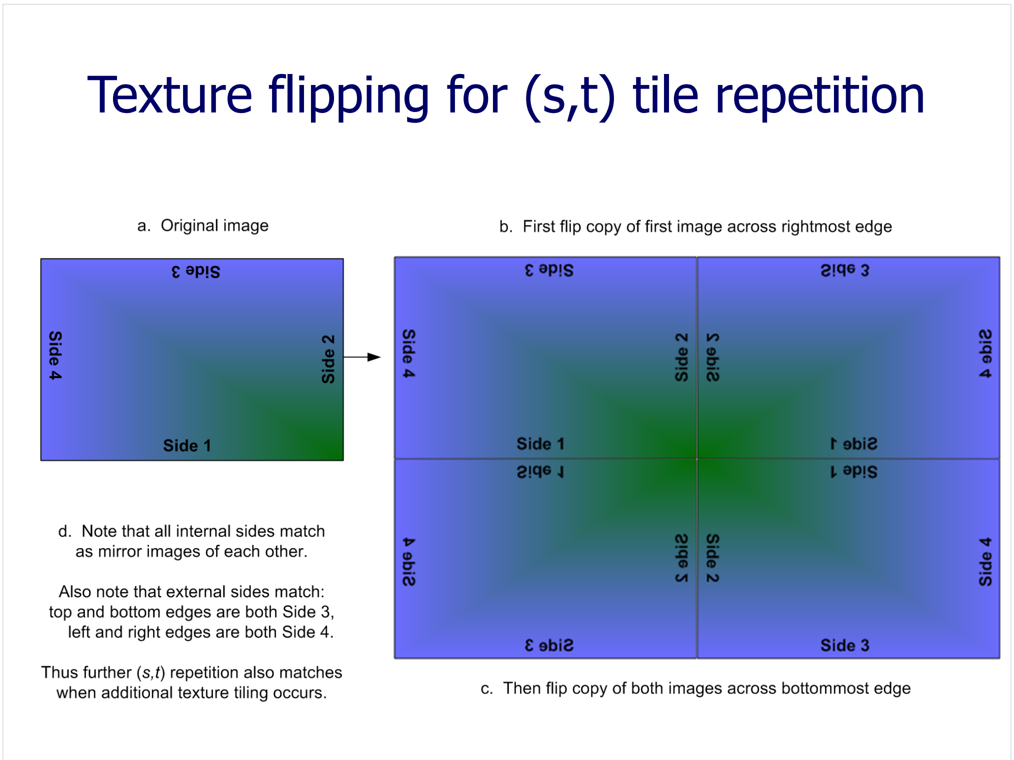
One of best is free, open source, recommended:

- Gnu Image Manipulation Program (GIMP) http://www.gimp.org



Drawing tools can also be helpful

- OpenOffice Draw, Impress http://www.openOffice.org



Texture flipping for (s,t) tile repetition

a. Original image

b. First flip copy of first image across rightmost edge

c. Then flip copy of both images across bottommost edge

d. Note that all internal sides match as mirror images of each other.

Also note that external sides match: top and bottom edges are both Side 3, left and right edges are both Side 4.

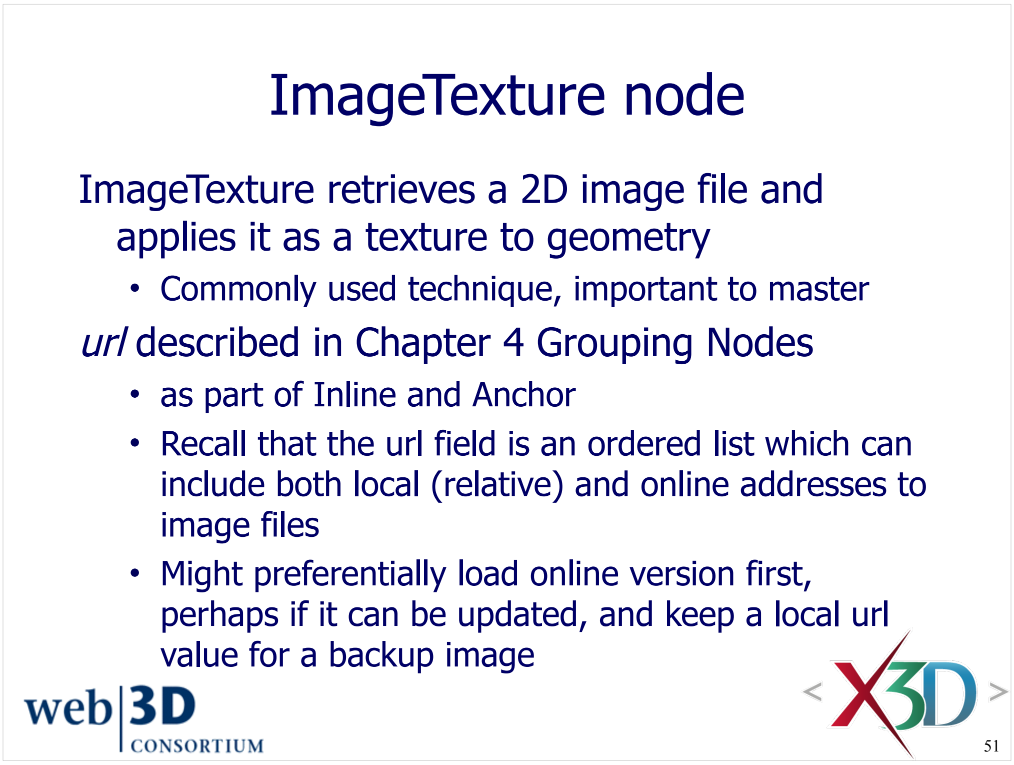
Thus further (*s*,*t*) repetition also matches when additional texture tiling occurs.

This is a nice trick for repetitive surfaces such as grass, water, sky, etc. that will show sharp, distracting edge artifacts if simply tiled as they originally appear.

Most image editors are capable of copying, flipping and aligning the quadrant images.

This is not a sufficient technique for smooth repetitive texturing if there are large color differences among the pixels within the original image being tiled.

TODO: add example scene using this texture



ImageTexture node

ImageTexture retrieves a 2D image file and applies it as a texture to geometry

- Commonly used technique, important to master

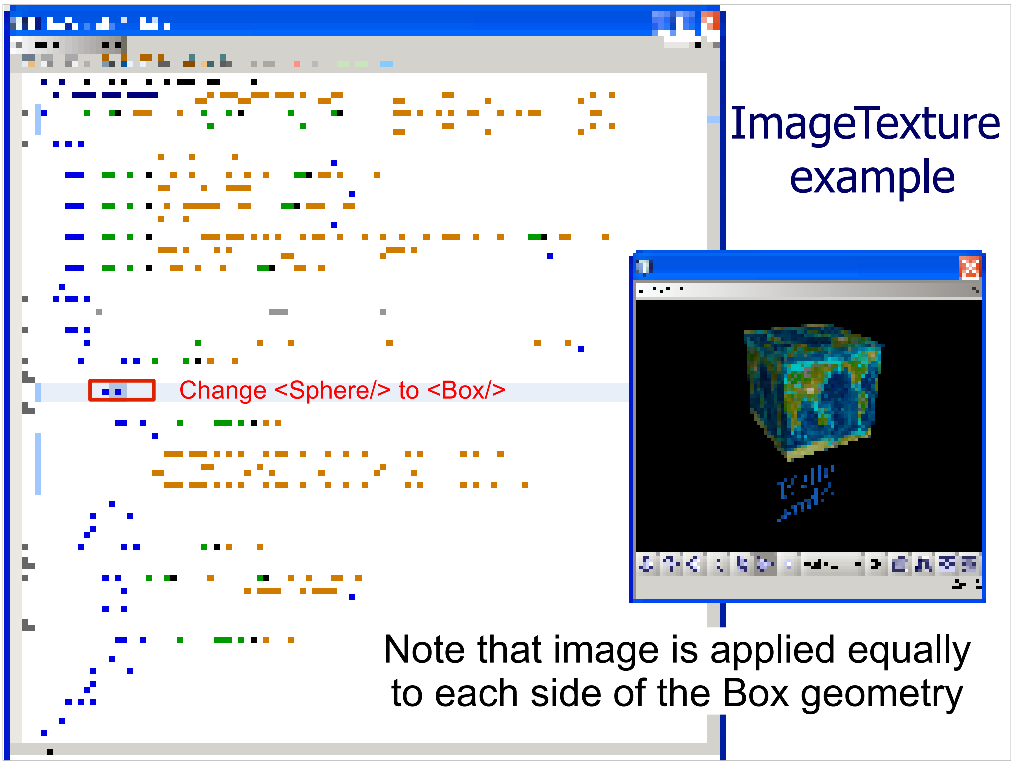
url described in Chapter 4 Grouping Nodes

- as part of Inline and Anchor
- Recall that the url field is an ordered list which can include both local (relative) and online addresses to image files
- Might preferentially load online version first, perhaps if it can be updated, and keep a local url value for a backup image

Examples of online imagery updates might be weather-related sky snapshots, webcam views, or perhaps other photography.

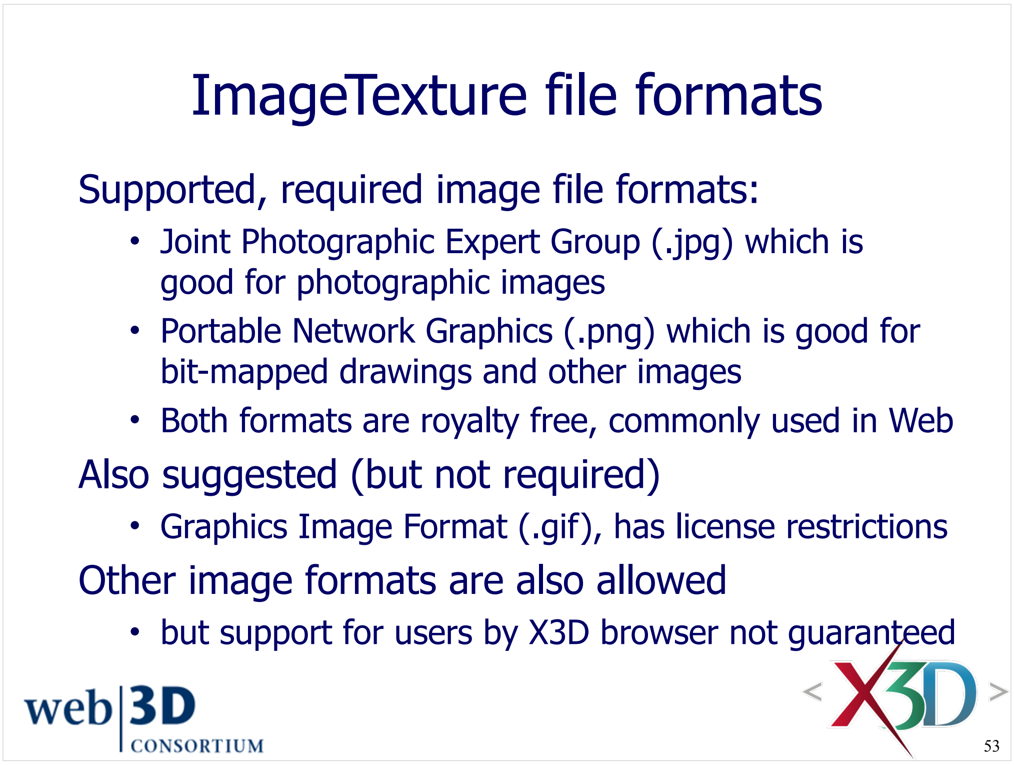
Further guidance on url links provided in X3D Scene Authoring Hints, provided within X3D-Edit help system or online at http://www.web3d.org/x3d/content/examples/X3dSceneAuthoringHints.html#urls

ImageTexture example



Change <Sphere/> to <Box/>

Note that image is applied equally to each side of the Box geometry



ImageTexture file formats

Supported, required image file formats:

- Joint Photographic Expert Group (.jpg) which is good for photographic images
- Portable Network Graphics (.png) which is good for bit-mapped drawings and other images
- Both formats are royalty free, commonly used in Web

Also suggested (but not required)

- Graphics Image Format (.gif), has license restrictions

Other image formats are also allowed

- but support for users by X3D browser not guaranteed

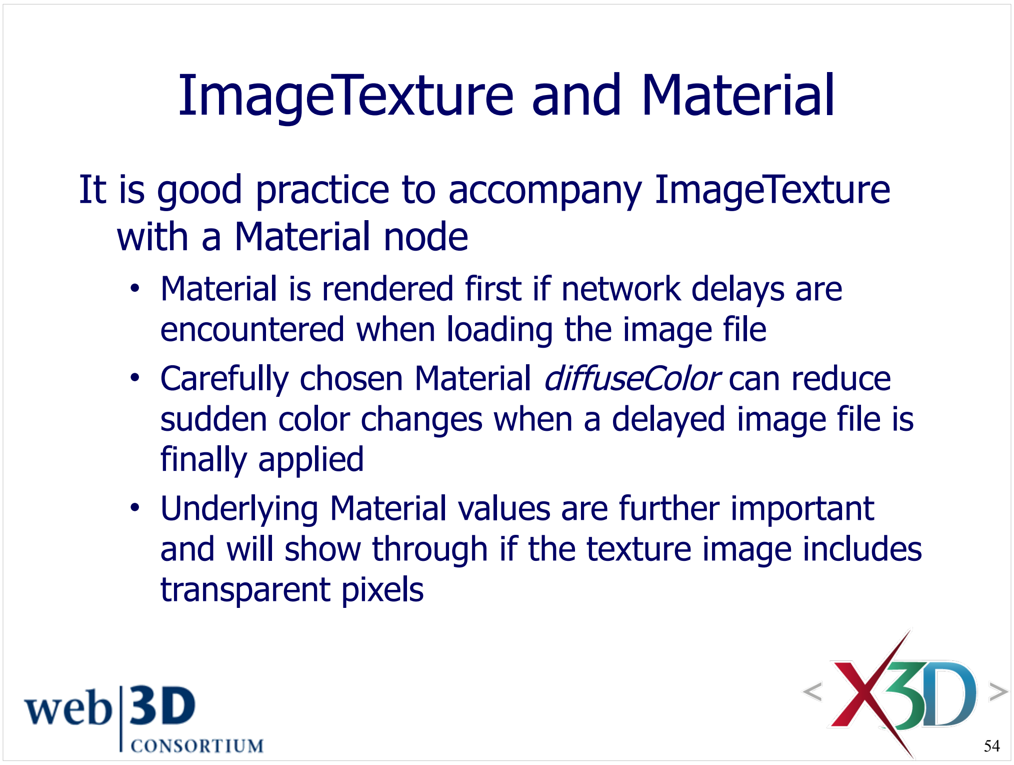
Specialty image file formats are allowed. There are no restrictions on what file formats can be referenced within the *url* values of an X3D scene.

One common approach to the use of specialty (perhaps high resolution) file formats is to list these first in the url ordered-list array, followed by an alternate version of the image file encoded in a format required to be supported (such as .png).

In that way, a broader-capability X3D browser can preferentially load the specialty image format first, but other regular X3D browsers will skip the unsupported format and then load the fallback url that lists the required format.

Further detail: X3D Specification, 18.2.2 Texture map image formats

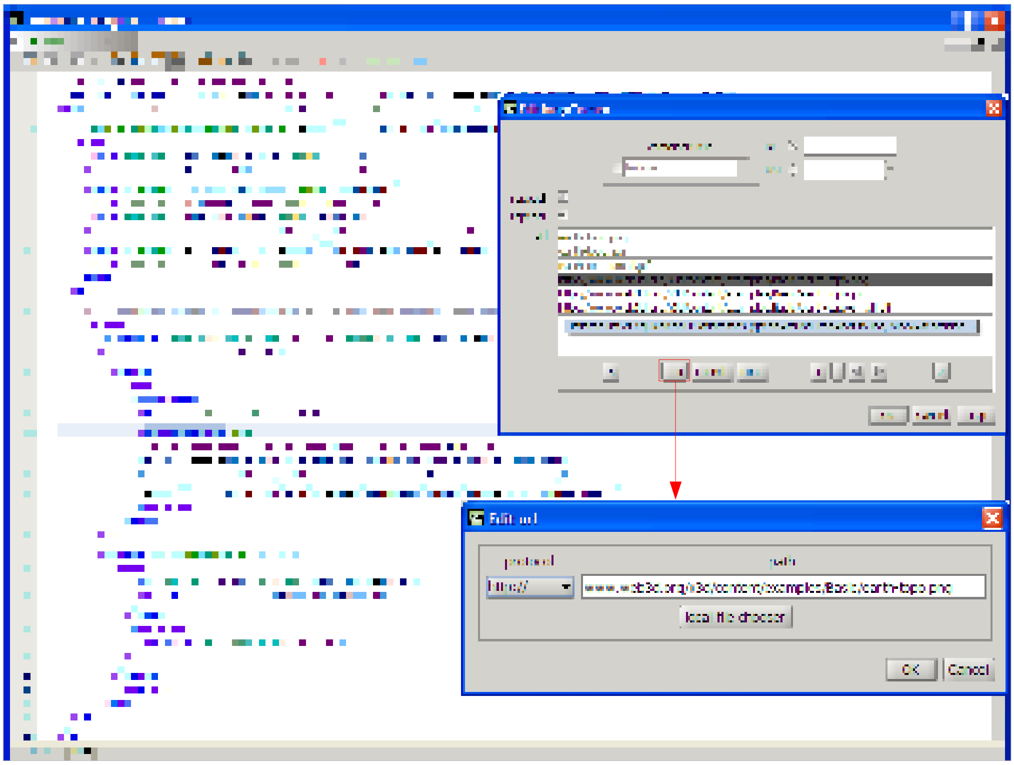
http://www.web3d.org/x3d/specifications/ISO-IEC-FDIS-19775-1.2-X3D-AbstractSpecification/Part01/components/texturing.html#Texturemapimageformats



ImageTexture and Material

It is good practice to accompany ImageTexture with a Material node

- Material is rendered first if network delays are encountered when loading the image file
- Carefully chosen Material *diffuseColor* can reduce sudden color changes when a delayed image file is finally applied
- Underlying Material values are further important and will show through if the texture image includes transparent pixels



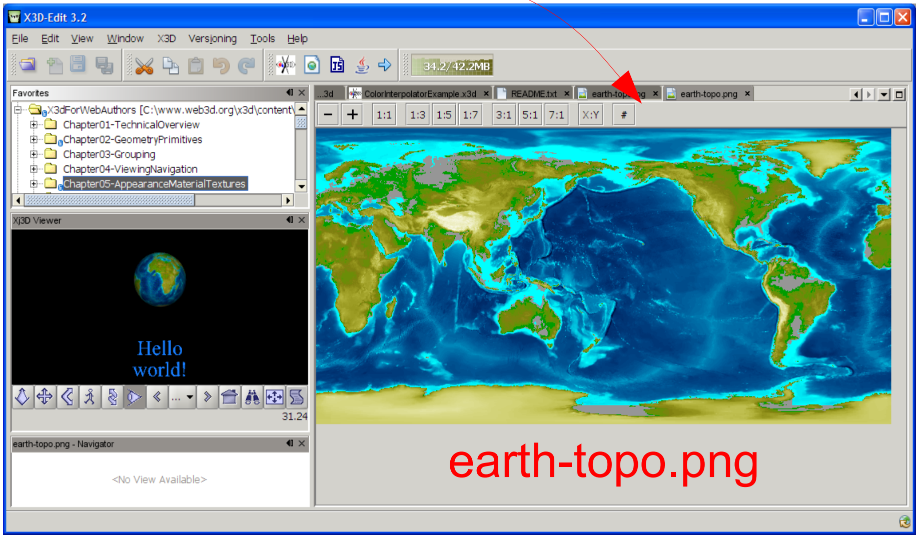
protocol: http://
path: www.web3d.org/x3d/content/examples/Basic/earth-topo.png

An old favorite.

http://www.web3d.org/x3d/content/examples/HelloWorld.x3d

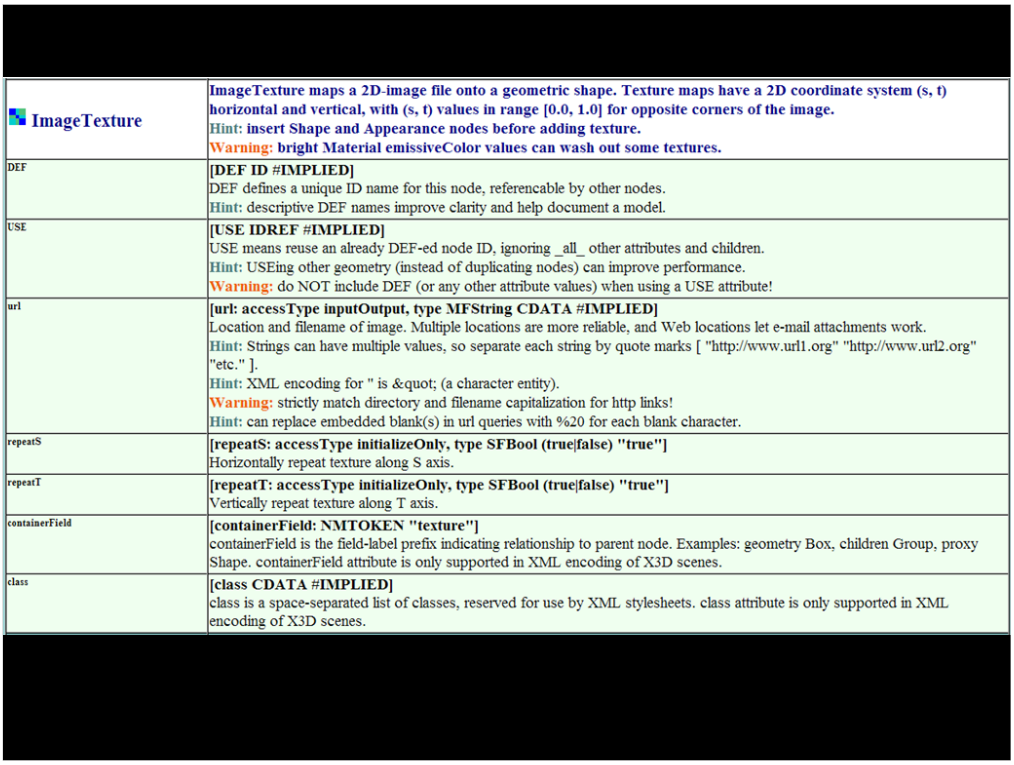
url list functionality:

- Color coding: green=resource found, red=resource not found, black=still searching
- Edit: edit url value, as shown in popup menu
- Launch: pass url to Web browser for display
- Load: load image in X3D-Edit



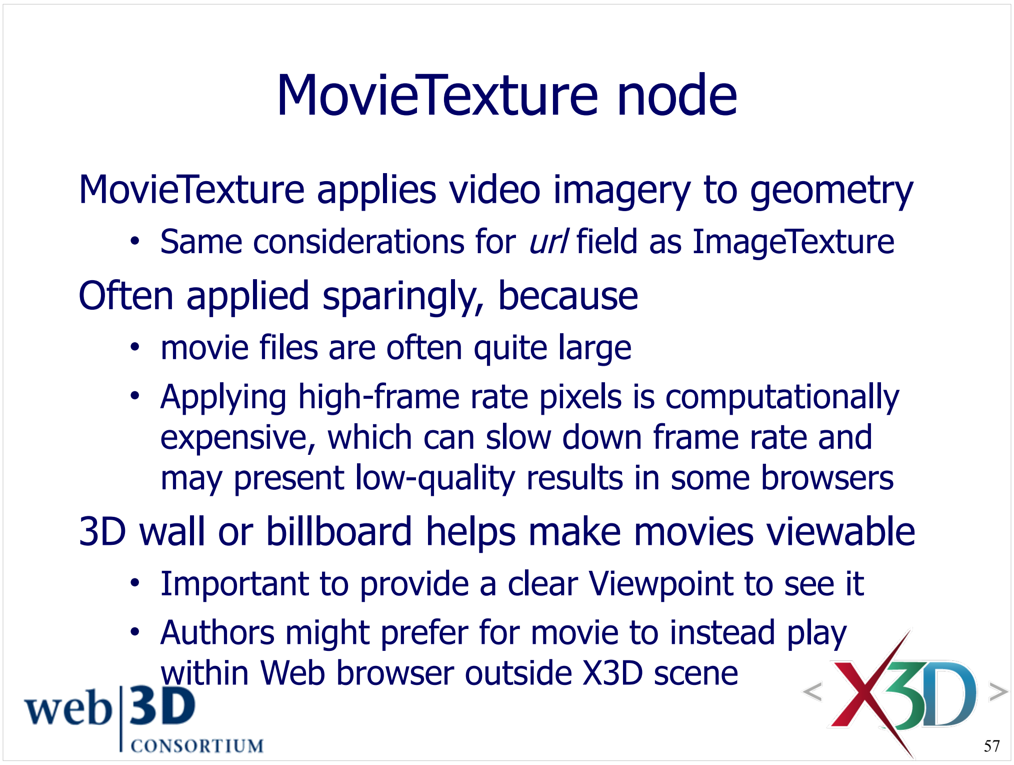
X3D-Edit 3.2
File
Edit
View
Window
X3D
Versioning
Tools
Help
34.2/42.2MB
Favorites
X3dForWebAuthors [C:\www.web3d.org\x3d\content\
Chapter01-TechnicalOverview
Chapter02-GeometryPrimitives
Chapter03-Grouping
Chapter04-ViewingNavigation
Chapter05-AppearanceMaterialTextures
Xj3D Viewer
Hello
world!
31.24
earth-topo.png - Navigator
<No View Available>
ColorInterpolatorExample.x3d
README.txt
earth-topo.png
earth-topo.png
1:1
1:3
1:5
1:7
3:1
5:1
7:1
X:Y
#

earth-topo.png



ImageTexture	ImageTexture maps a 2D-image file onto a geometric shape. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image. Hint: insert Shape and Appearance nodes before adding texture. Warning: bright Material emissiveColor values can wash out some textures.
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!
url	[url: accessType inputOutput, type MFString CDATA #IMPLIED] Location and filename of image. Multiple locations are more reliable, and Web locations let e-mail attachments work. 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.
repeatS	[repeatS: accessType initializeOnly, type SFBool (true|false) "true"] Horizontally repeat texture along S axis.
repeatT	[repeatT: accessType initializeOnly, type SFBool (true|false) "true"] Vertically repeat texture along T axis.
containerField	[containerField: NMTOKEN "texture"] 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#ImageTexture



MovieTexture node

MovieTexture applies video imagery to geometry

- Same considerations for *url* field as ImageTexture

Often applied sparingly, because

- movie files are often quite large
- Applying high-frame rate pixels is computationally expensive, which can slow down frame rate and may present low-quality results in some browsers

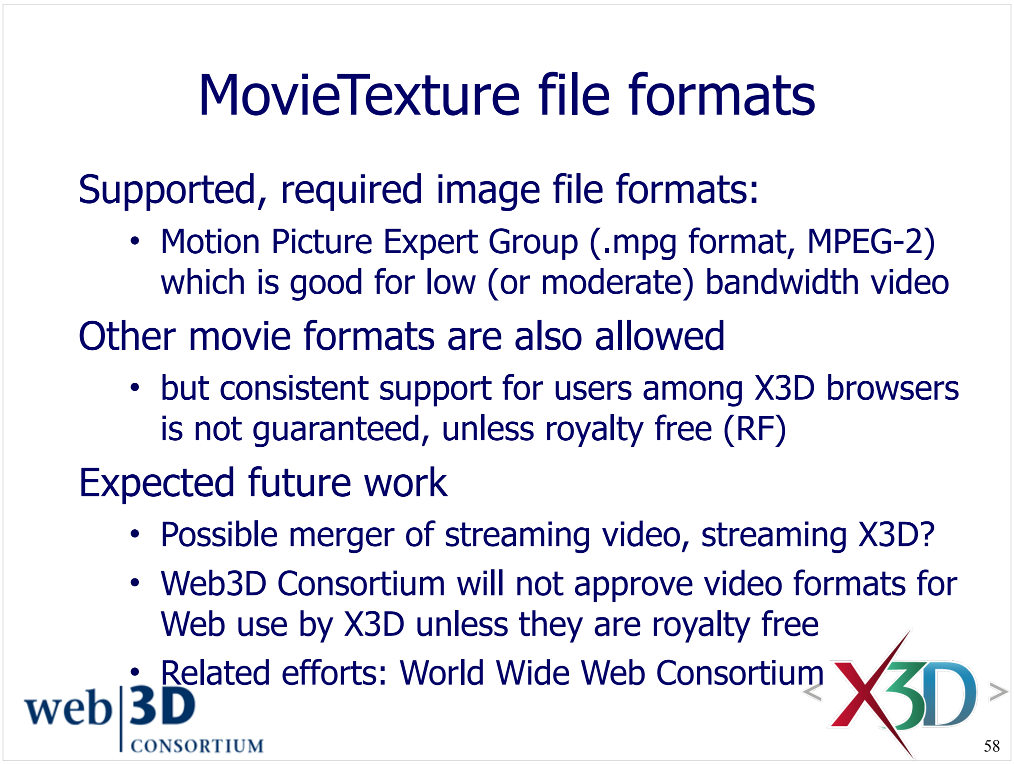
3D wall or billboard helps make movies viewable

- Important to provide a clear Viewpoint to see it
- Authors might prefer for movie to instead play within Web browser outside X3D scene

Of course, as systems and graphics cards get faster and faster, with more and more memory, the size of movie files is becoming much less of a problem these days.

MovieTexture examples are available in ConformanceNIST suite, locally downloadable using X3D-Edit or else found online at

http://www.web3d.org/x3d/content/examples/ConformanceNist/Appearance/MovieTexture



MovieTexture file formats

Supported, required image file formats:

- Motion Picture Expert Group (.mpg format, MPEG-2) which is good for low (or moderate) bandwidth video

Other movie formats are also allowed

- but consistent support for users among X3D browsers is not guaranteed, unless royalty free (RF)

Expected future work

- Possible merger of streaming video, streaming X3D?
- Web3D Consortium will not approve video formats for Web use by X3D unless they are royalty free
- Related efforts: World Wide Web Consortium

Specialty movie file formats are allowed. There are no restrictions on what file formats can be referenced within the *url* values of an X3D scene.

One common approach to the use of specialty (perhaps high resolution) file formats is to list these first in the url ordered-list array, followed by an alternate version of the image file encoded in the format required to be supported (which is .jpg).

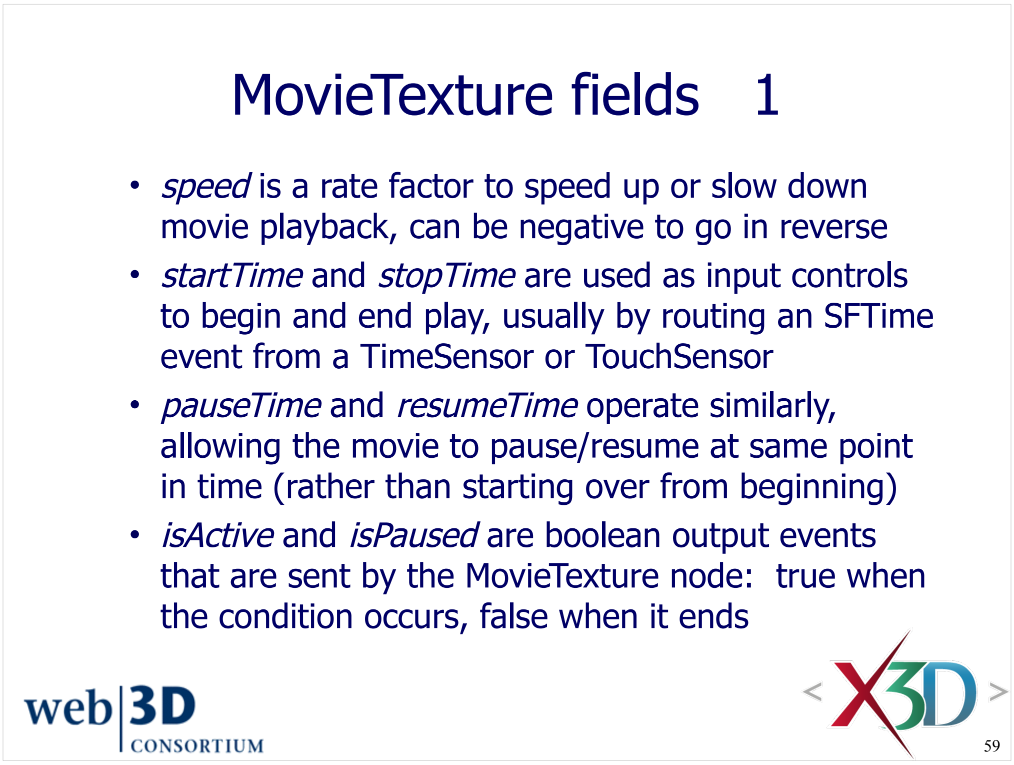
In that way, a broader-capability X3D browser can preferentially load the specialty image format first, but other regular X3D browsers will skip the unsupported format and then load the fallback url that lists the required format.

Related work at W3C: Video on the Web at http://www.w3.org/2008/WebVideo

Reference: *Video Requirements for Web-based Virtual Environments using Extensible 3D (X3D) Graphics*, Don Brutzman, Mathias Kolsch (Web3D Consortium, Naval Postgraduate School, Monterey California USA).

http://www.w3.org/2007/08/video/positions/Web3D.pdf

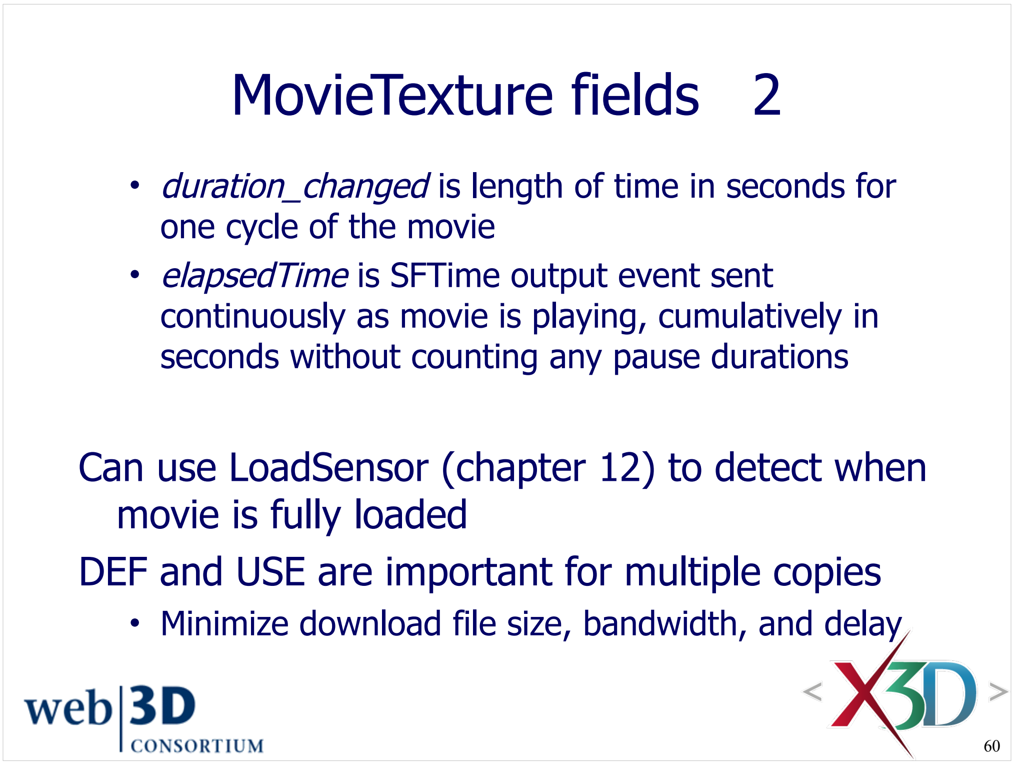
Abstract. Real-time interactive 3D graphics and virtual environments typically include a variety of multimedia capabilities, including video. The Extensible 3D (X3D) Graphics is an ISO standard produced by the Web3D Consortium that defines 3D scenes using a scene-graph approach. Multiple X3D file formats and language encodings are available, with a primary emphasis on XML for maximum interoperability with the Web architecture. A large number of functional capabilities are needed and projected for the use of video together with Web-based virtual environments. This paper examines numerous functional requirements for the integrated use of Web-compatible video with 3D. Three areas of interest are identified: video usage within X3D scenes, linking video external to X3D scenes, and generation of 3D geometry from video.



MovieTexture fields 1

- *speed* is a rate factor to speed up or slow down movie playback, can be negative to go in reverse
- *startTime* and *stopTime* are used as input controls to begin and end play, usually by routing an SFTime event from a TimeSensor or TouchSensor
- *pauseTime* and *resumeTime* operate similarly, allowing the movie to pause/resume at same point in time (rather than starting over from beginning)
- *isActive* and *isPaused* are boolean output events that are sent by the MovieTexture node: true when the condition occurs, false when it ends

Events are described further in Chapter 7



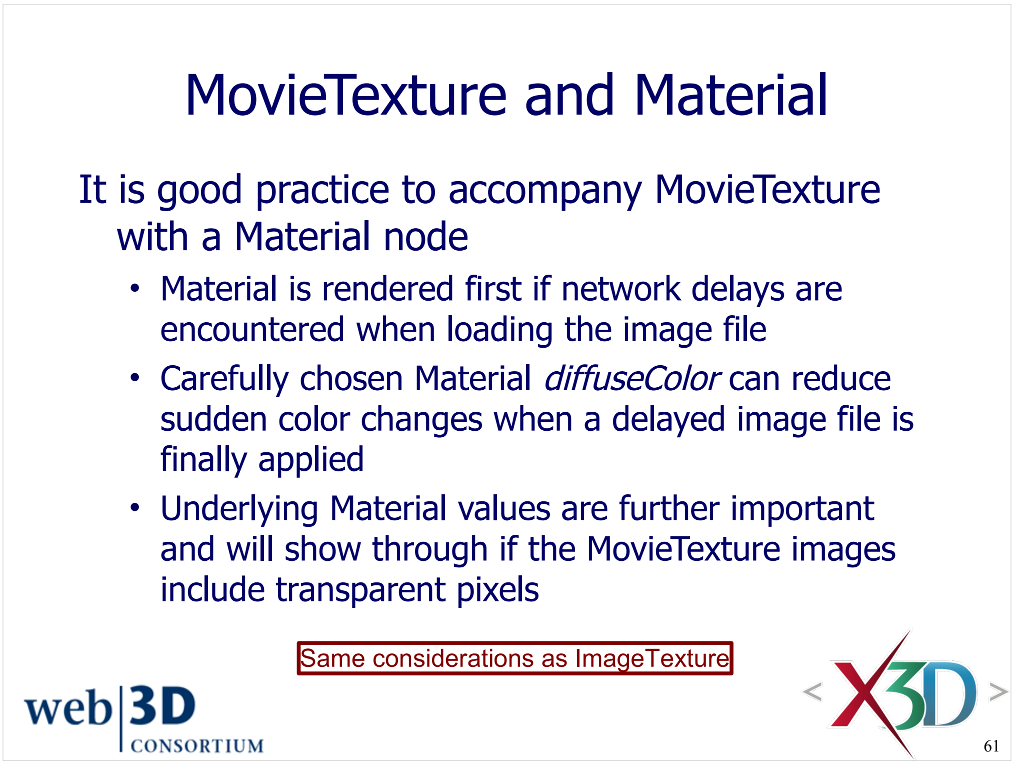
MovieTexture fields 2

- *duration_changed* is length of time in seconds for one cycle of the movie
- *elapsedTime* is SFTime output event sent continuously as movie is playing, cumulatively in seconds without counting any pause durations

Can use LoadSensor (chapter 12) to detect when movie is fully loaded

DEF and USE are important for multiple copies

- Minimize download file size, bandwidth, and delay

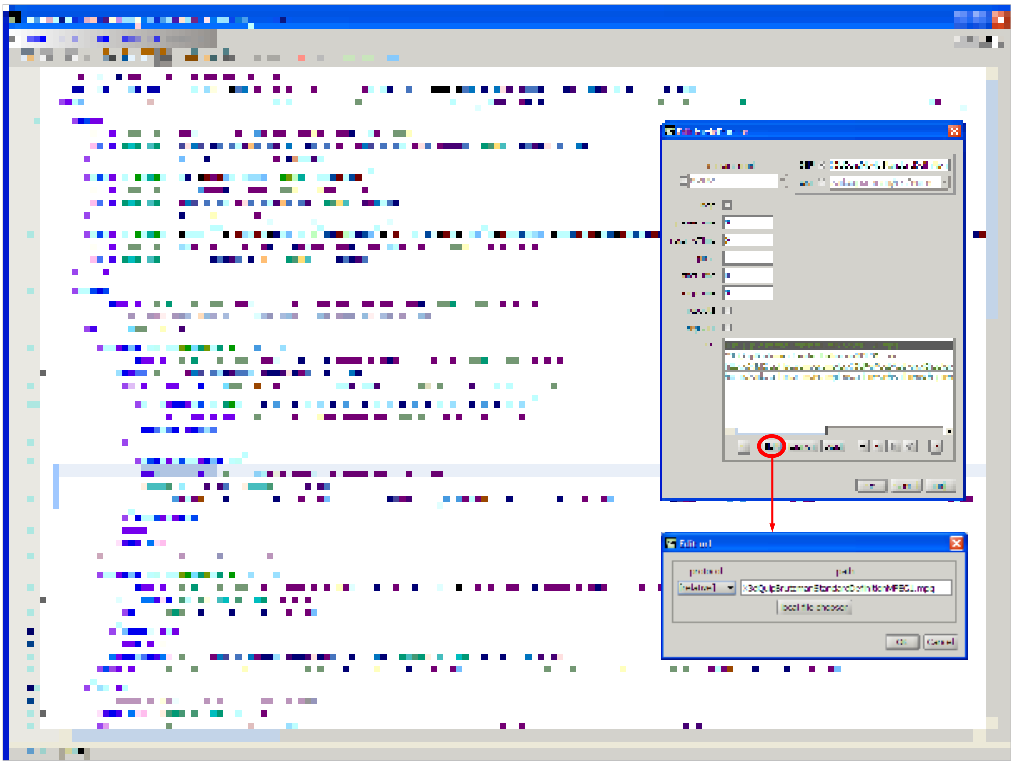


MovieTexture and Material

It is good practice to accompany MovieTexture with a Material node

- Material is rendered first if network delays are encountered when loading the image file
- Carefully chosen Material *diffuseColor* can reduce sudden color changes when a delayed image file is finally applied
- Underlying Material values are further important and will show through if the MovieTexture images include transparent pixels

Same considerations as ImageTexture



[relative] X3dQuipBrutzmanStandardDefinitionMPEG1.mpg

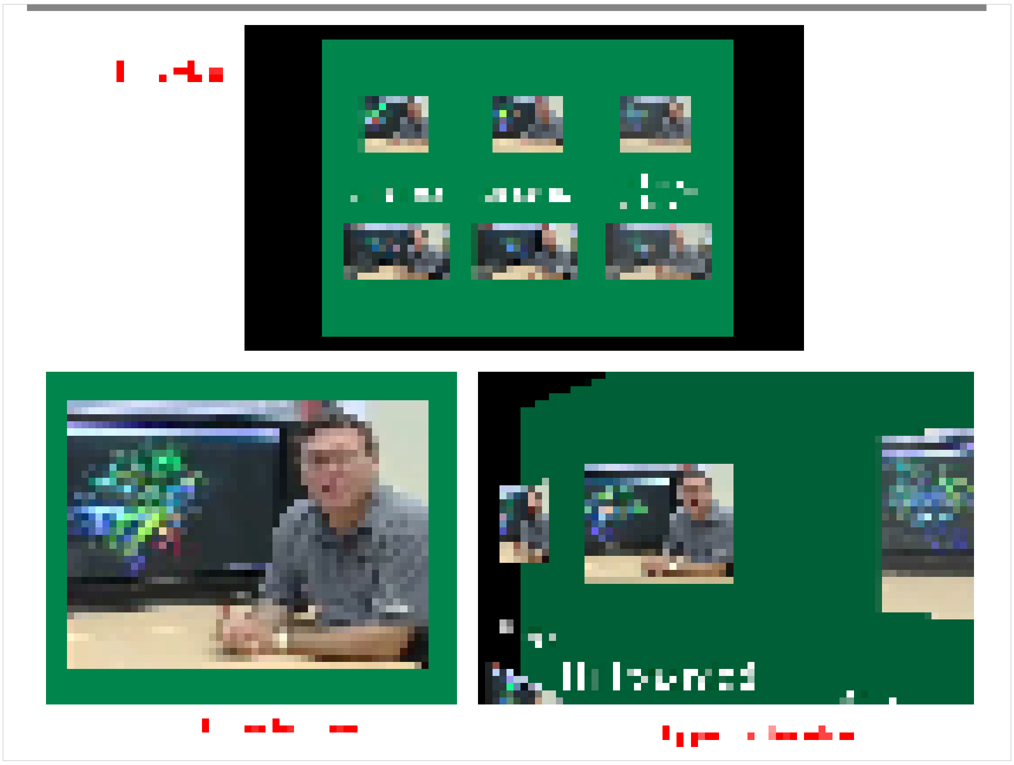
Note the presence of

```
  <TextureCoordinate DEF='FullImageMapping' point='0 0 1 0 1 1 0 1'/>
```

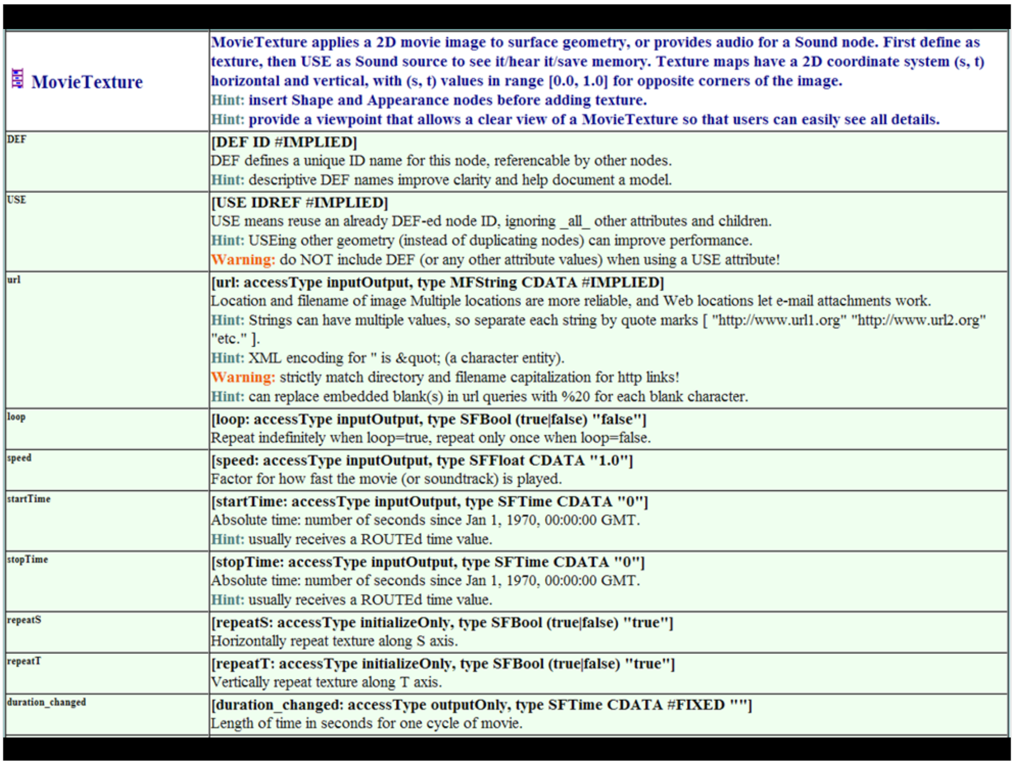
to guarantee proper (s, t) mapping of the MovieTexture image corners to a rectangular IndexedFaceSet.

Note that url values are listed in green, indicating that X3D-Edit was able to find the resources OK. Missing resources are listed in red, while a check in progress keeps the resource listed in black.

The url editor is also able to launch a resource into the browser for testing. If the file type is supported (X3D, image, HTML) then the author can also load it into X3D-Edit. X3D-Edit support of movie files isn't supported.

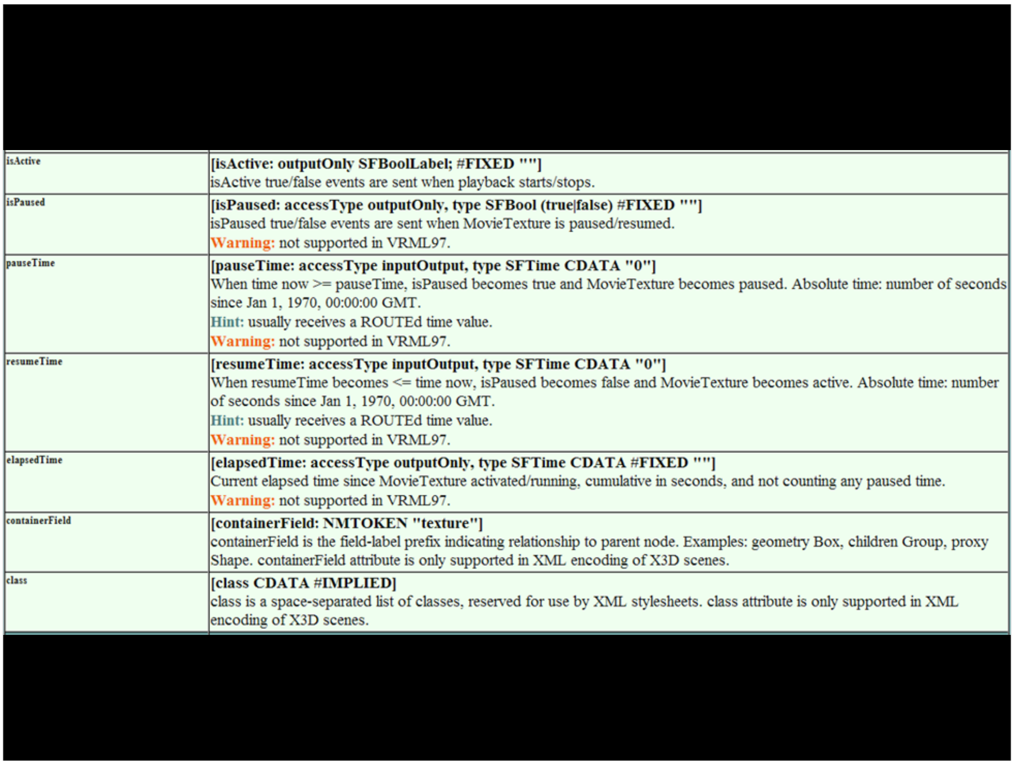


Billboard above a MovieTexture rotates it to always face the user.



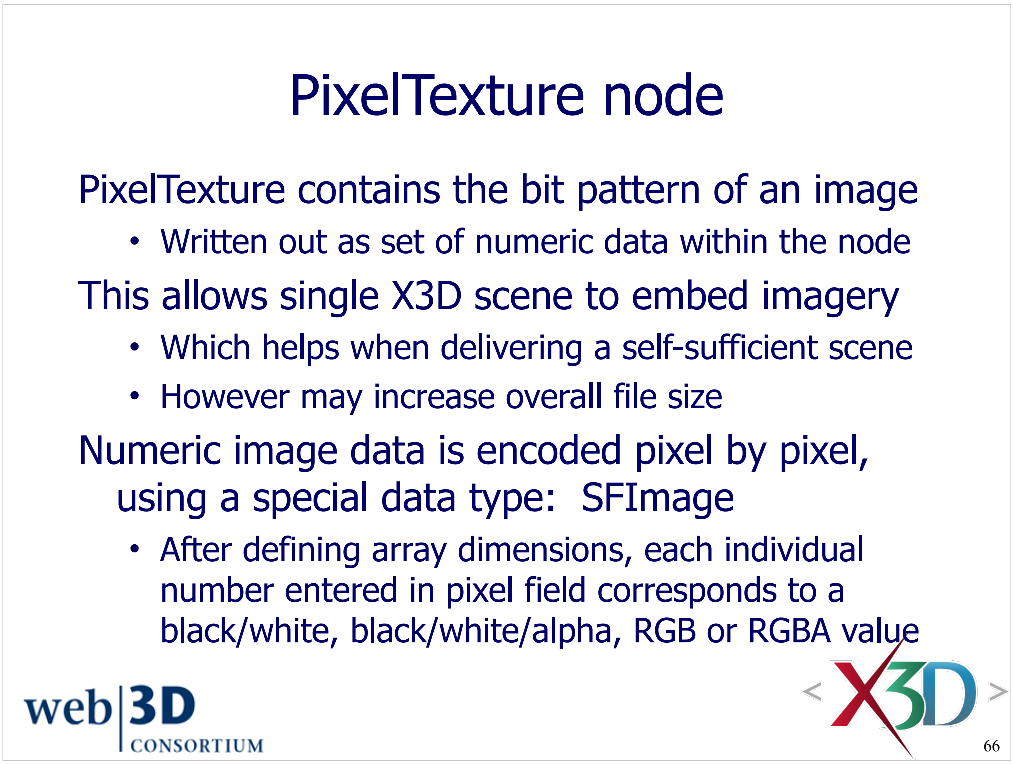
MovieTexture	**MovieTexture applies a 2D movie image to surface geometry, or provides audio for a Sound node. First define as texture, then USE as Sound source to see it/hear it/save memory. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image.** **Hint: insert Shape and Appearance nodes before adding texture.** **Hint: provide a viewpoint that allows a clear view of a MovieTexture so that users can easily see all details.**
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!
url	**[url: accessType inputOutput, type MFString CDATA #IMPLIED]** Location and filename of image Multiple locations are more reliable, and Web locations let e-mail attachments work. 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.
loop	**[loop: accessType inputOutput, type SFBool (true|false) "false"]** Repeat indefinitely when loop=true, repeat only once when loop=false.
speed	**[speed: accessType inputOutput, type SFFloat CDATA "1.0"]** Factor for how fast the movie (or soundtrack) is played.
startTime	**[startTime: accessType inputOutput, type SFTime CDATA "0"]** Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. Hint: usually receives a ROUTEd time value.
stopTime	**[stopTime: accessType inputOutput, type SFTime CDATA "0"]** Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. Hint: usually receives a ROUTEd time value.
repeatS	**[repeatS: accessType initializeOnly, type SFBool (true|false) "true"]** Horizontally repeat texture along S axis.
repeatT	**[repeatT: accessType initializeOnly, type SFBool (true|false) "true"]** Vertically repeat texture along T axis.
duration_changed	**[duration_changed: accessType outputOnly, type SFTime CDATA #FIXED ""]** Length of time in seconds for one cycle of movie.

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



isActive	**[isActive: outputOnly SFBoolLabel; #FIXED ""]** isActive true/false events are sent when playback starts/stops.
isPaused	**[isPaused: accessType outputOnly, type SFBool (true|false) #FIXED ""]** isPaused true/false events are sent when MovieTexture is paused/resumed. **Warning:** not supported in VRML97.
pauseTime	**[pauseTime: accessType inputOutput, type SFTime CDATA "0"]** When time now >= pauseTime, isPaused becomes true and MovieTexture becomes paused. Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. **Hint:** usually receives a ROUTEd time value. **Warning:** not supported in VRML97.
resumeTime	**[resumeTime: accessType inputOutput, type SFTime CDATA "0"]** When resumeTime becomes <= time now, isPaused becomes false and MovieTexture becomes active. Absolute time: number of seconds since Jan 1, 1970, 00:00:00 GMT. **Hint:** usually receives a ROUTEd time value. **Warning:** not supported in VRML97.
elapsedTime	**[elapsedTime: accessType outputOnly, type SFTime CDATA #FIXED ""]** Current elapsed time since MovieTexture activated/running, cumulative in seconds, and not counting any paused time. **Warning:** not supported in VRML97.
containerField	**[containerField: NMTOKEN "texture"]** 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#MovieTexture



PixelTexture node

PixelTexture contains the bit pattern of an image

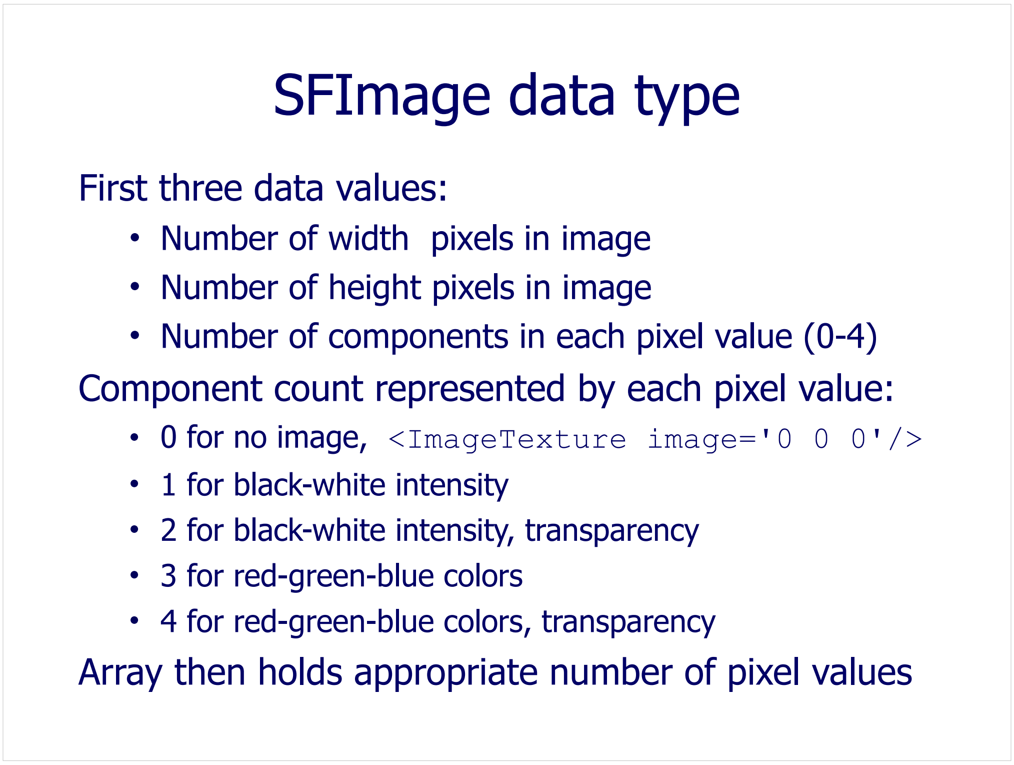
- Written out as set of numeric data within the node

This allows single X3D scene to embed imagery

- Which helps when delivering a self-sufficient scene
- However may increase overall file size

Numeric image data is encoded pixel by pixel, using a special data type: SFImage

- After defining array dimensions, each individual number entered in pixel field corresponds to a black/white, black/white/alpha, RGB or RGBA value



SFImage data type

First three data values:

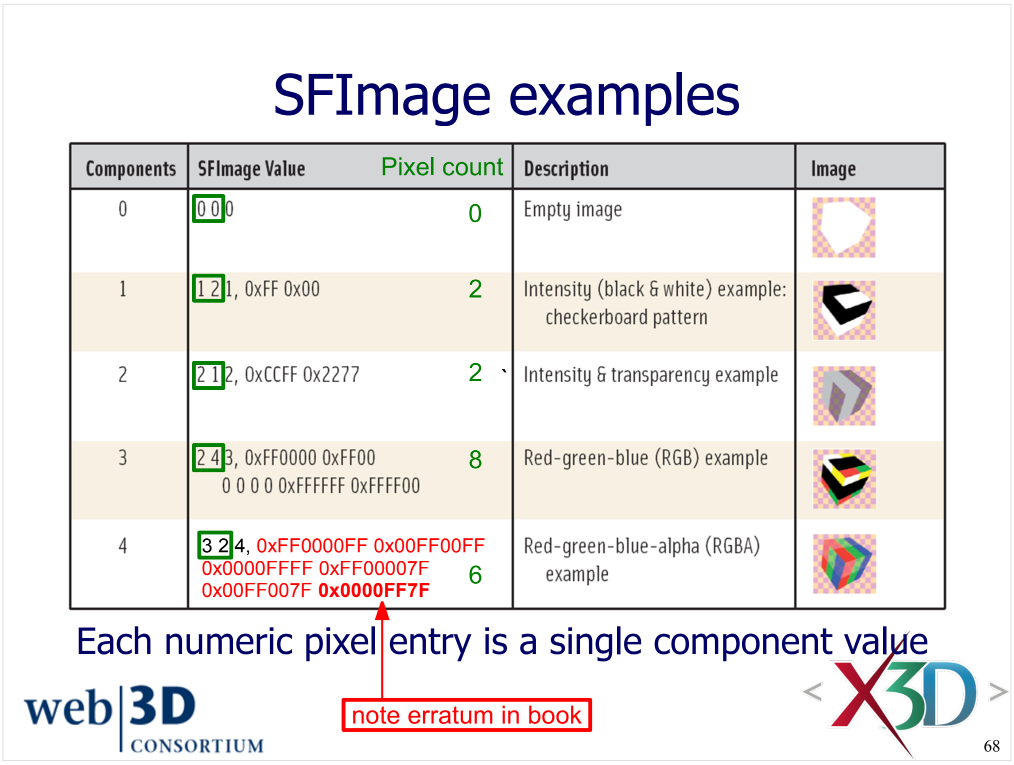
- Number of width pixels in image
- Number of height pixels in image
- Number of components in each pixel value (0-4)

Component count represented by each pixel value:

- 0 for no image, `<ImageTexture image='0 0 0'/>`
- 1 for black-white intensity
- 2 for black-white intensity, transparency
- 3 for red-green-blue colors
- 4 for red-green-blue colors, transparency

Array then holds appropriate number of pixel values

Pixel values (as well as any other X3D integer value) can be in decimal or hexadecimal format.



SFImage examples

Components	SFImage Value	Pixel count	Description	Image
0	0 0 0	0	Empty image	
1	1 2 1, 0xFF 0x00	2	Intensity (black & white) example: checkerboard pattern	
2	2 1 2, 0xCCFF 0x2277	2 `	Intensity & transparency example	
3	2 4 3, 0xFF0000 0xFF00 0 0 0 0 0xFFFFFF 0xFFFF00	8	Red-green-blue (RGB) example	
4	3 2 4, 0xFF0000FF 0x00FF00FF 0x0000FFFF 0xFF00007F 0x00FF007F **0x0000FF7F**	6	Red-green-blue-alpha (RGBA) example	

Each numeric pixel entry is a single component value

note erratum in book

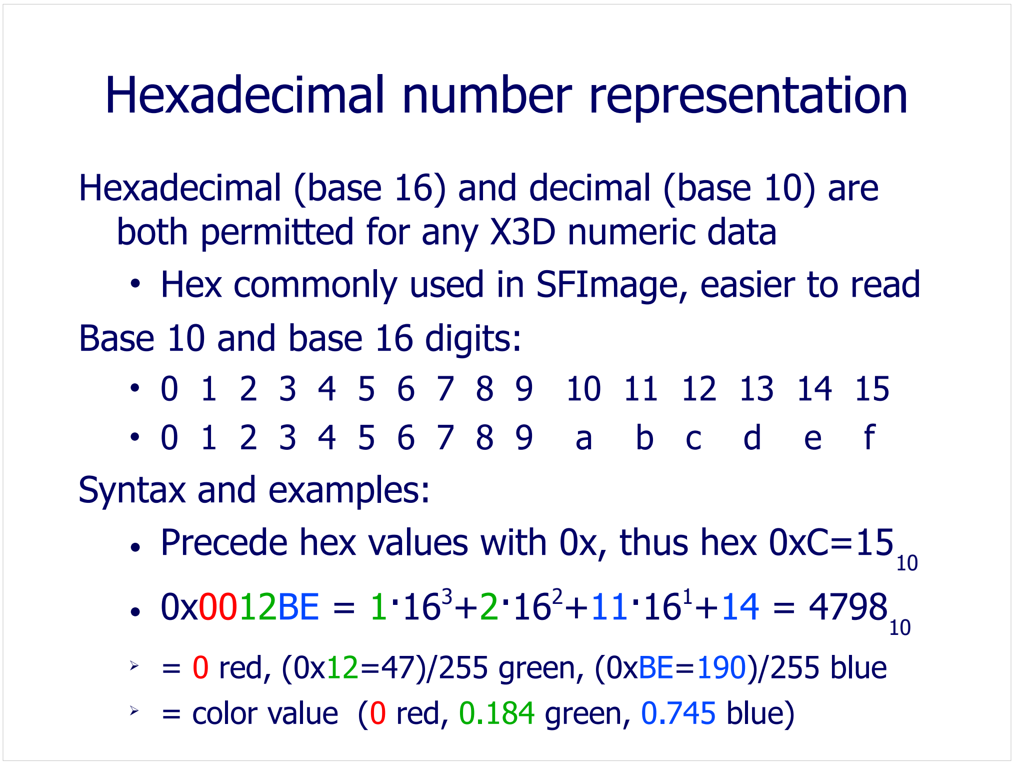
68

X3D for Web Authors, Table 5.18, p.145.

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/PixelTextureComponentExamples.x3d

Erratum correction applied to 4-component example (omit leading FF in final value).

http://x3dgraphics.com/errata.php



Hexadecimal number representation

Hexadecimal (base 16) and decimal (base 10) are both permitted for any X3D numeric data

- Hex commonly used in SFImage, easier to read

Base 10 and base 16 digits:

- 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
- 0 1 2 3 4 5 6 7 8 9 a b c d e f

Syntax and examples:

- Precede hex values with 0x, thus hex $0xC=15_{10}$
- $0x0012BE = 1 \cdot 16^3 + 2 \cdot 16^2 + 11 \cdot 16^1 + 14 = 4798_{10}$
  - = 0 red, (0x12=47)/255 green, (0xBE=190)/255 blue
  - = color value (0 red, 0.184 green, 0.745 blue)

Hexadecimal is base sixteen, which is more concise and suitable for binary data. Both decimal (base 10) and hexadecimal data appear in the above example.

Base 10 digits:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 16 digits:	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

Note X3D and ClassicVRML encoding prefix for hexadecimal data is '`0x`'

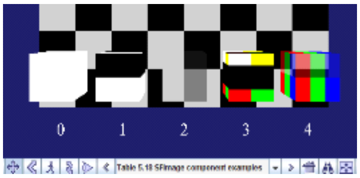
Hexadecimal data is unsigned, sign information is carried in most-significant bit.

binary	octal	decimal	hex	binary	octal	decimal	hex
0000	0	0	0	1000	10	8	8
0001	1	1	1	1001	11	9	9
0010	2	2	2	1010	12	10	a
0011	3	3	3	1011	13	11	b
0100	4	4	4	1100	14	12	c
0101	5	5	5	1101	15	13	d
0110	6	6	6	1110	16	14	e
0111	7	7	7	1111	17	15	f

Also note that leading zeros may be omitted. Thus, for the slide example above,

0x12BE = 0x0012BE

Example value conversions

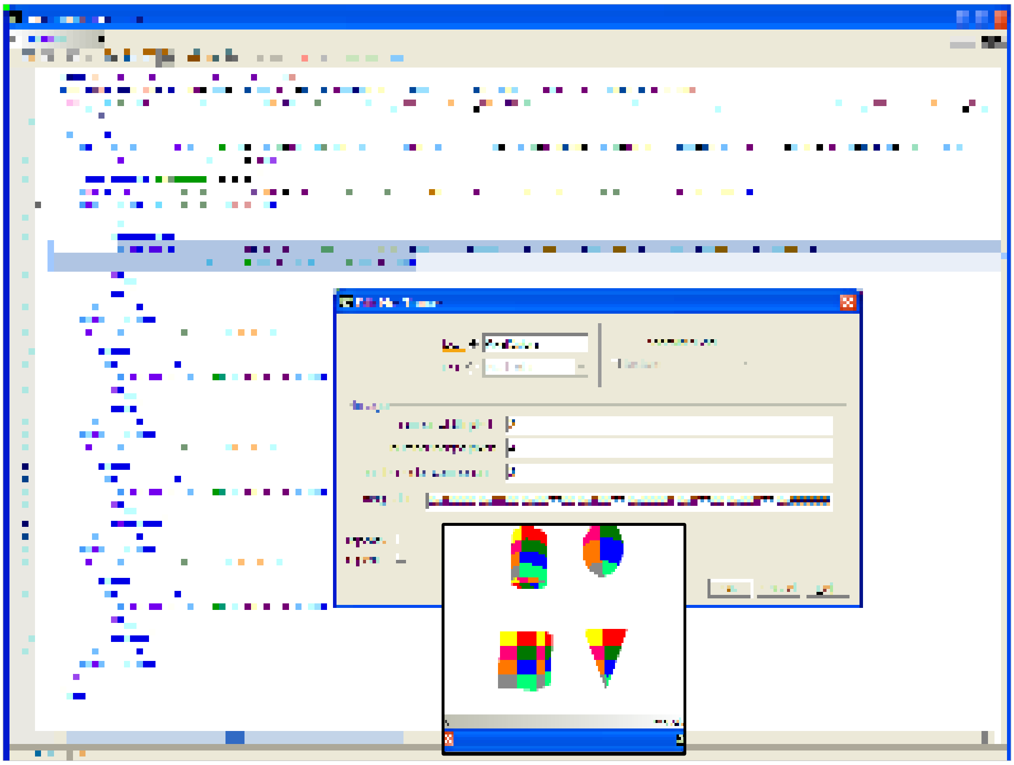
Hex value		0-255 range		bw	r	g	b	a	transparency	color
0xFF		255		1						⬜
0x00		0		0						⬛
0xCCFF		204		0.8				255	0.0	░
0x2277		34		0.13				119	0.53	▓
0xFF0000	255	0	0		1	0	0			🟥
0xFF00	0	255	0		0	1	0			🟩
0xFFFFFF	255	255	255		1	1	1			⬜
0xFFFF00	255	255	0		1	1	0			🟨
					1	0	0	255	0.0	🟥
					0	1	0	255	0.0	🟩
					0	0	1	255	0.0	🟦
					1	0	0	127	0.5	🟥
					0	1	0	127	0.5	🟩
					0	0	1	127	0.5	🟦

Note that alpha a = (1- transparency)

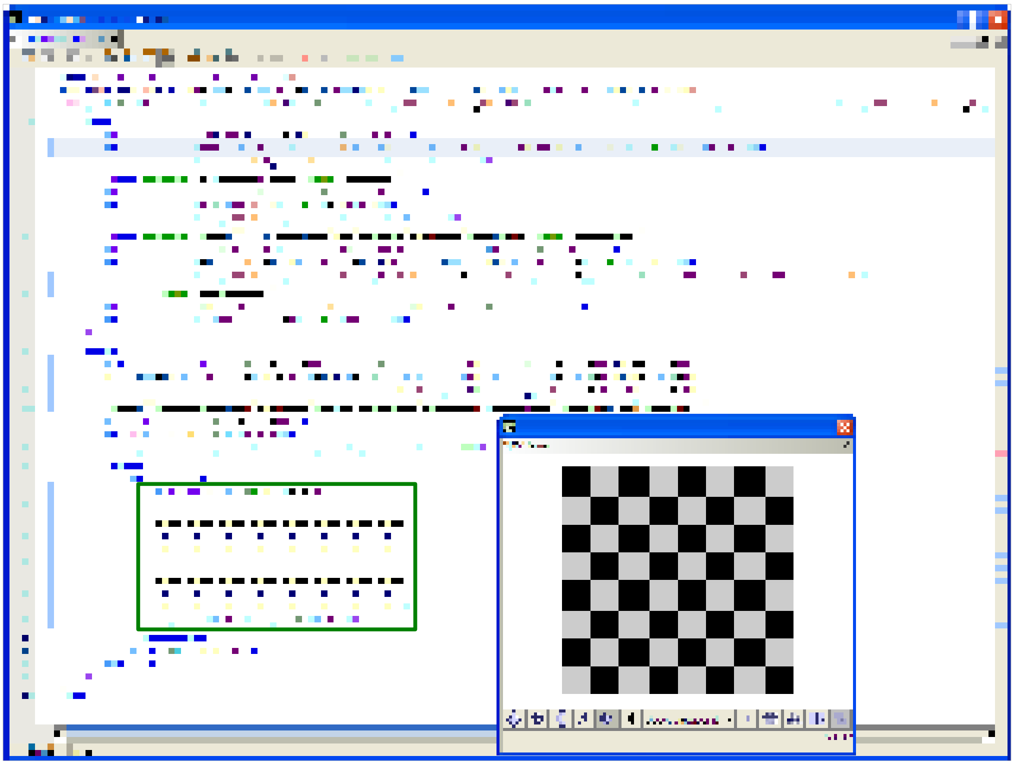
X3D for Web Authors, Table 5.18, p.145.

Errata correction applied to 4-component example.

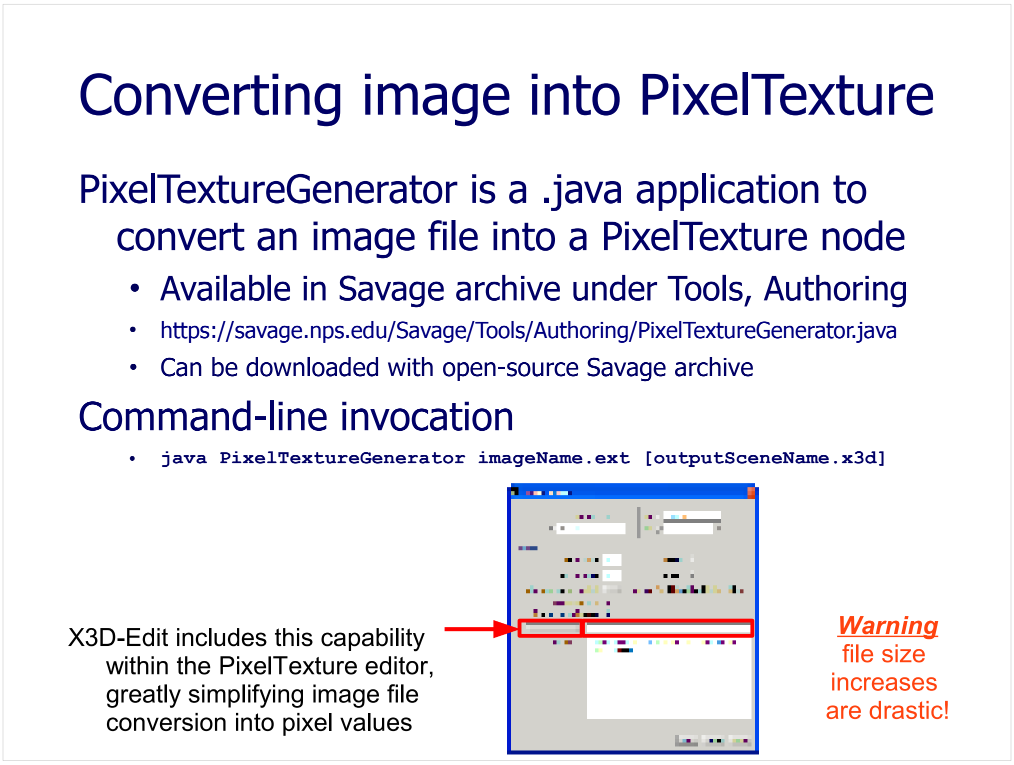
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/PixelTextureComponentExamples.x3d



http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/PixelTexture.x3d



http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/PixelTextureBW.x3d



Converting image into PixelTexture

PixelTextureGenerator is a .java application to convert an image file into a PixelTexture node

- Available in Savage archive under Tools, Authoring
- https://savage.nps.edu/Savage/Tools/Authoring/PixelTextureGenerator.java
- Can be downloaded with open-source Savage archive

Command-line invocation

- `java PixelTextureGenerator imageName.ext [outputSceneName.x3d]`

X3D-Edit includes this capability within the PixelTexture editor, greatly simplifying image file conversion into pixel values

Warning
file size increases are drastic!

Example showing use of PixelTextureGenerator to convert image file:
https://savage.nps.edu/Savage/Tools/Authoring/PixelTextureNavyJackDontTreadOnMe.x3d

Once a PixelTexture is generate by the class, it can be inserted into the following X3D template to make a complete Shape, At that point it is ready for copying and further use in an X3D scene.

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

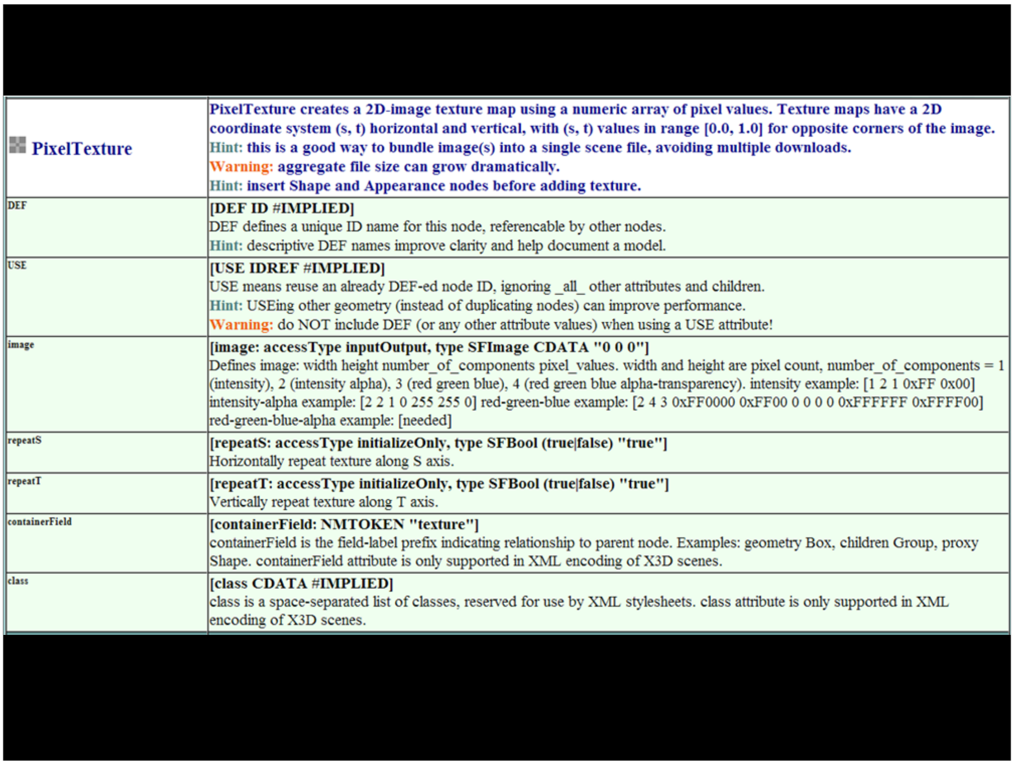
Use of the PixelTexture editing pane in X3D-Edit automates the use of this class and makes it easy to convert image files. Nevertheless be careful about file size, typically they may get about **25** times larger than the original!

Navy Jack references:

https://savage.nps.edu/Savage/Tools/Authoring/NavyJackDontTreadOnMe.gif

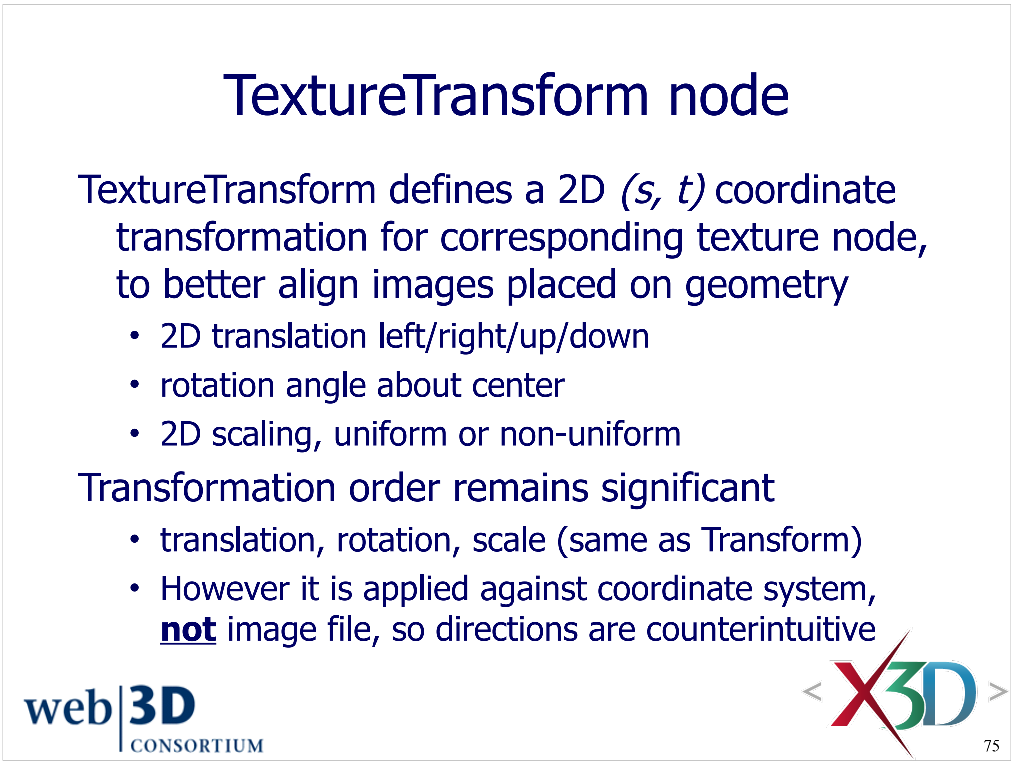
https://savage.nps.edu/Savage/Tools/Authoring/NavyJackDontTreadOnMe.txt

DONT TREAD ON ME



PixelTexture	**PixelTexture creates a 2D-image texture map using a numeric array of pixel values. Texture maps have a 2D coordinate system (s, t) horizontal and vertical, with (s, t) values in range [0.0, 1.0] for opposite corners of the image.** **Hint: this is a good way to bundle image(s) into a single scene file, avoiding multiple downloads.** **Warning: aggregate file size can grow dramatically.** **Hint: insert Shape and Appearance nodes before adding texture.**
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!
image	**[image: accessType inputOutput, type SFImage CDATA "0 0 0"]** Defines image: width height number_of_components pixel_values. width and height are pixel count, number_of_components = 1 (intensity), 2 (intensity alpha), 3 (red green blue), 4 (red green blue alpha-transparency). intensity example: [1 2 1 0xFF 0x00] intensity-alpha example: [2 2 1 0 255 255 0] red-green-blue example: [2 4 3 0xFF0000 0xFF00 0 0 0 0 0xFFFFFF 0xFFFF00] red-green-blue-alpha example: [needed]
repeatS	**[repeatS: accessType initializeOnly, type SFBool (true|false) "true"]** Horizontally repeat texture along S axis.
repeatT	**[repeatT: accessType initializeOnly, type SFBool (true|false) "true"]** Vertically repeat texture along T axis.
containerField	**[containerField: NMTOKEN "texture"]** 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#PixelTexture



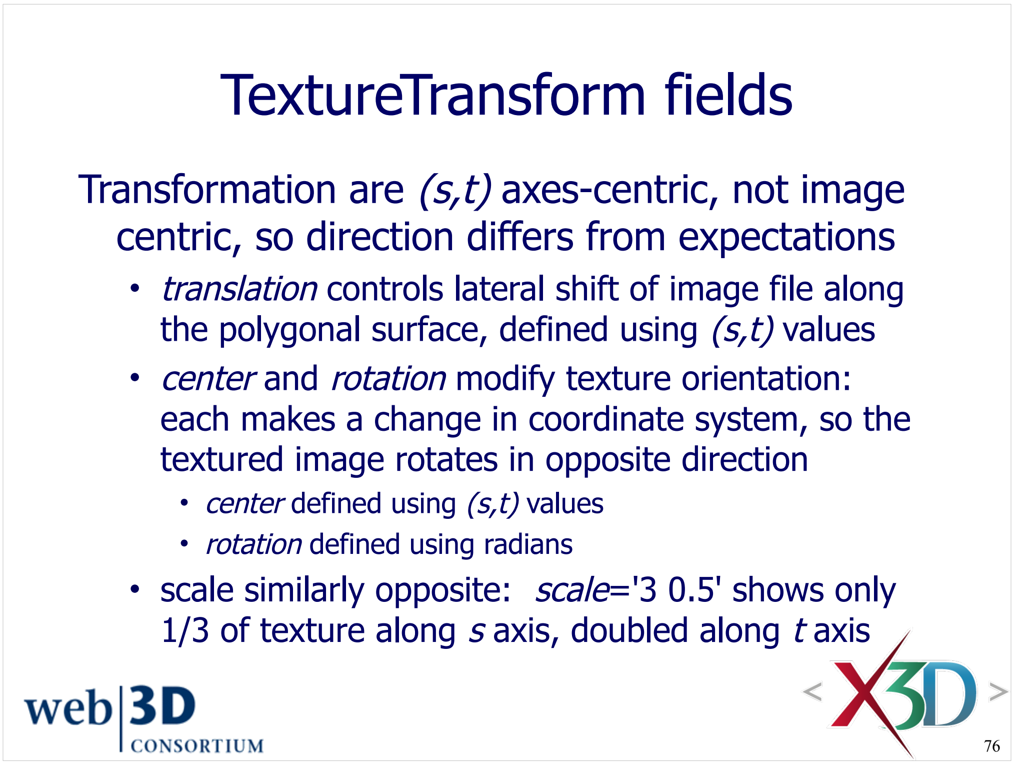
TextureTransform node

TextureTransform defines a 2D *(s, t)* coordinate transformation for corresponding texture node, to better align images placed on geometry

- 2D translation left/right/up/down
- rotation angle about center
- 2D scaling, uniform or non-uniform

Transformation order remains significant

- translation, rotation, scale (same as Transform)
- However it is applied against coordinate system, **not** image file, so directions are counterintuitive



TextureTransform fields

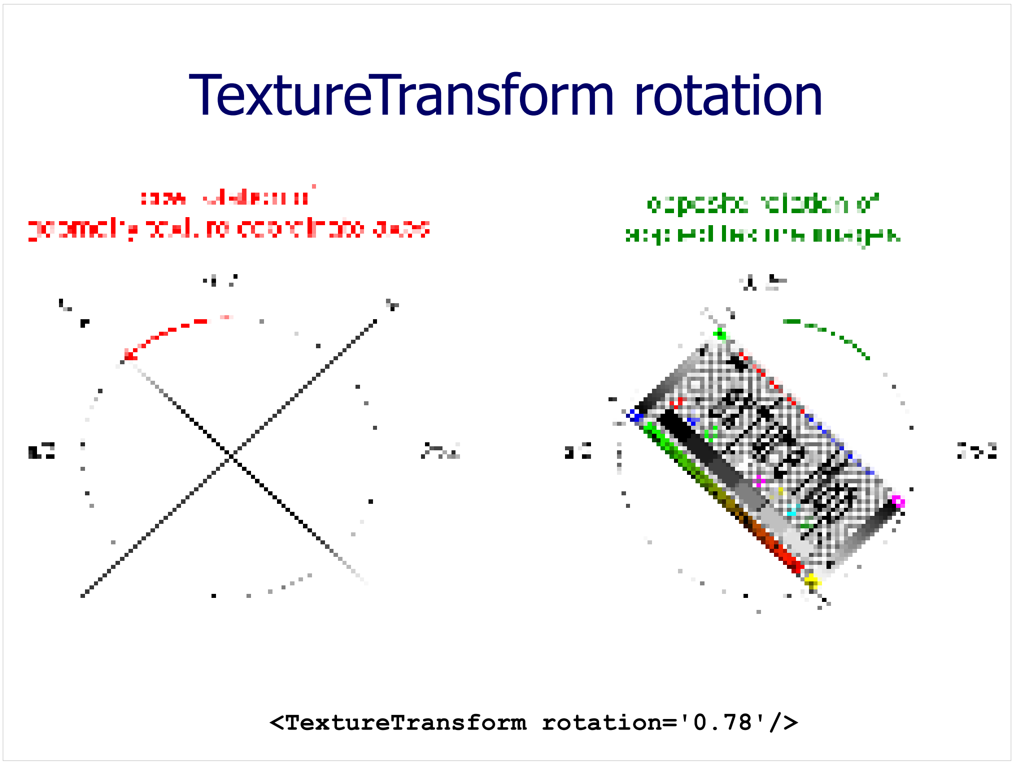
Transformation are *(s,t)* axes-centric, not image centric, so direction differs from expectations

- *translation* controls lateral shift of image file along the polygonal surface, defined using *(s,t)* values
- *center* and *rotation* modify texture orientation: each makes a change in coordinate system, so the textured image rotates in opposite direction
  - *center* defined using *(s,t)* values
  - *rotation* defined using radians
- scale similarly opposite: *scale*='3 0.5' shows only 1/3 of texture along *s* axis, doubled along *t* axis

web3D CONSORTIUM

<X3D>

76

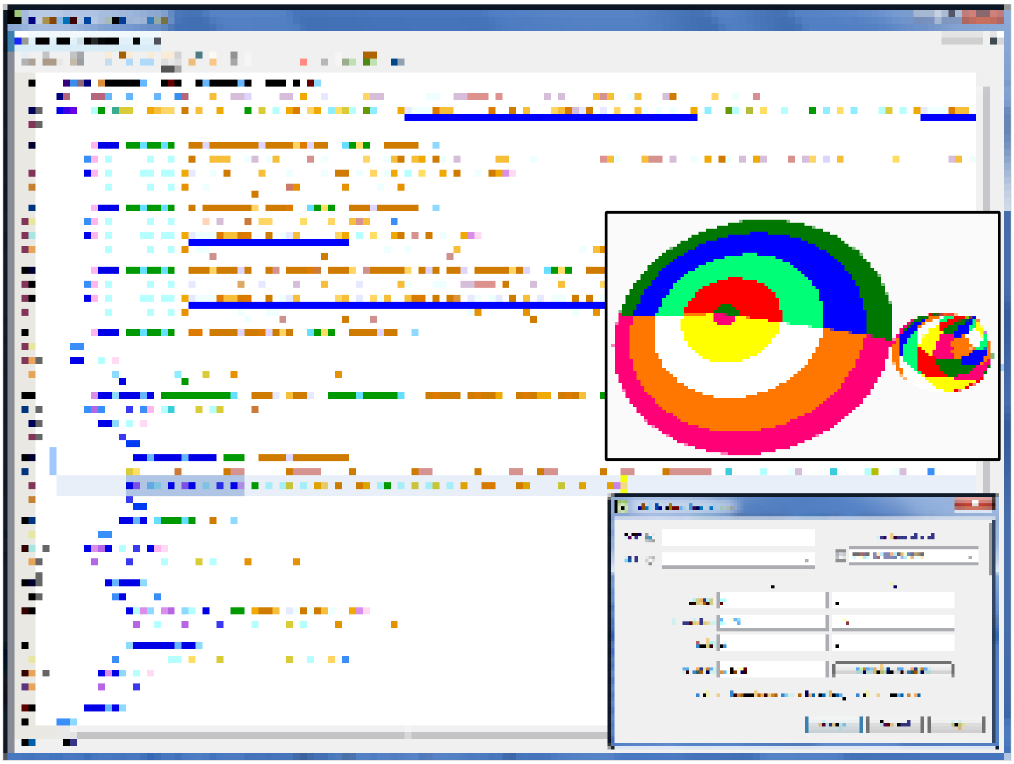


TextureTransform rotation

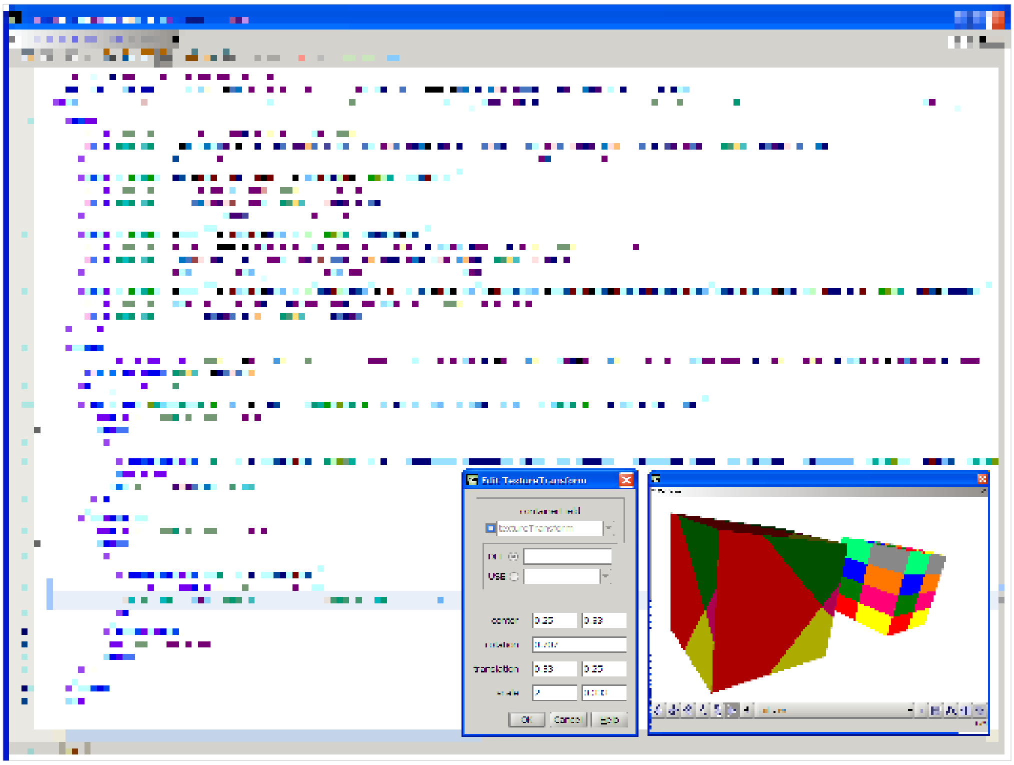
```
<TextureTransform rotation='0.78'/>
```

Key point: TextureTransform rotations turn in the direction opposite to that expected. This is because the rotation operates on the (s,t) coordinate system, rather than the individual coordinates themselves.

This convention has always been the case in computer graphics, so X3D remains consistent rather than trying to change the semantics of rotation direction (since that would cause many unnecessary incompatibility problems).



http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/TextureTransform.x3d



Edit TextureTransform

containerField textureTransform

DEF ○

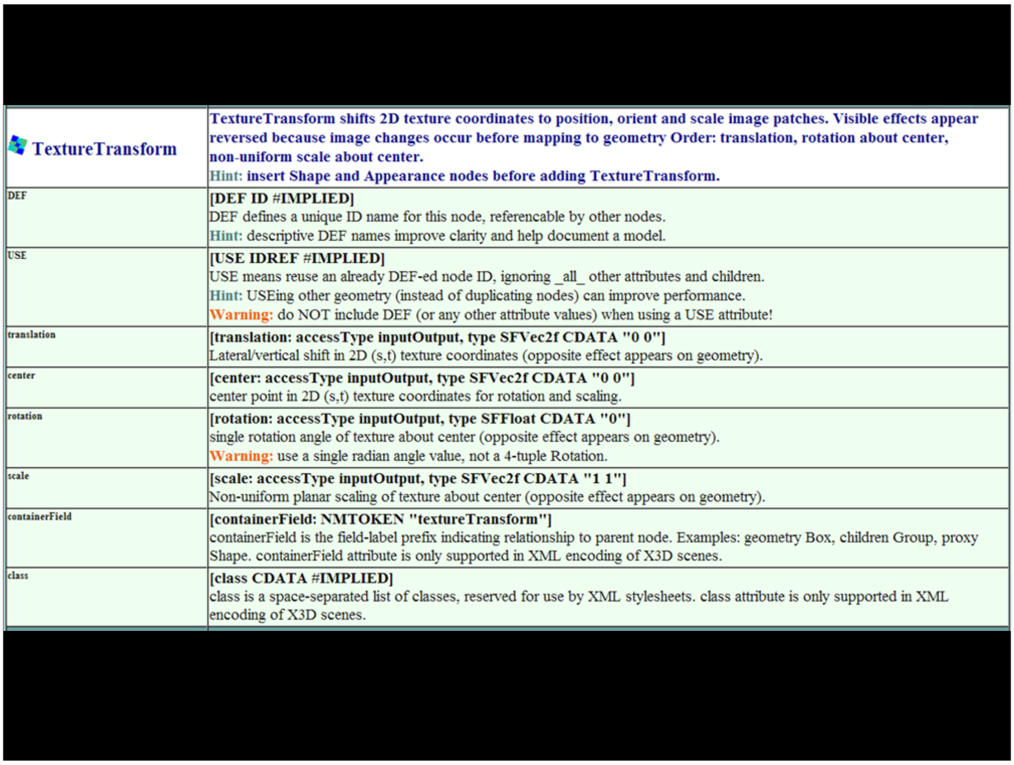
USE ○

center 0.25 0.33

rotation [illegible]

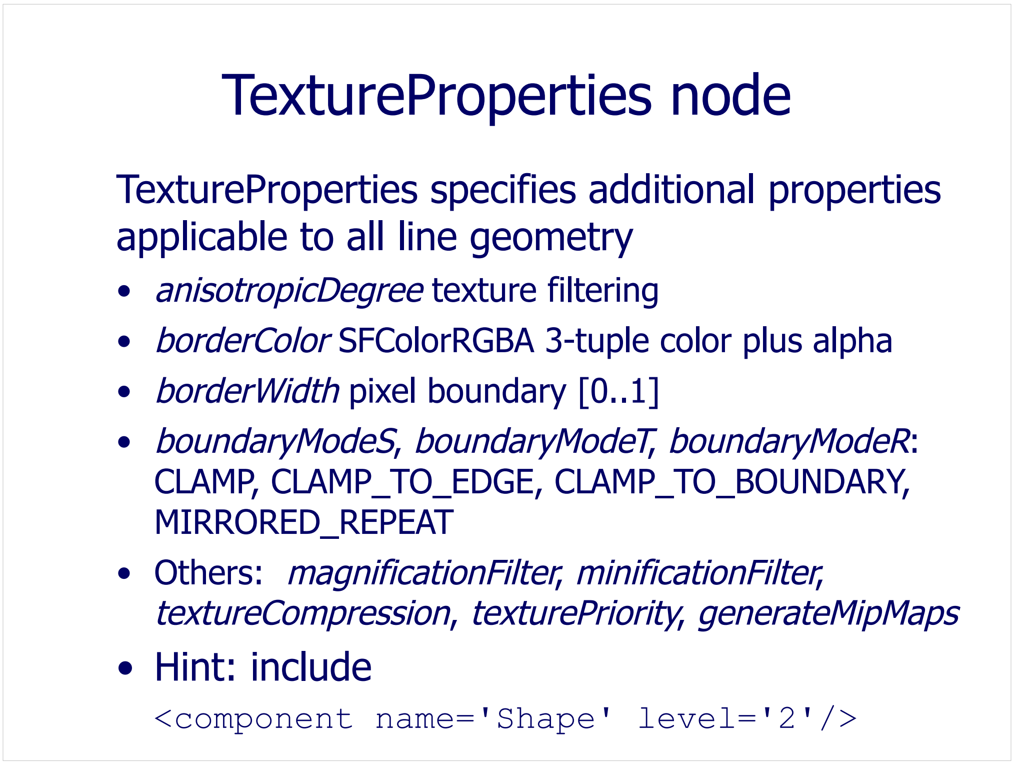
translation 0.33 0.25

scale 2 [illegible]



TextureTransform	TextureTransform shifts 2D texture coordinates to position, orient and scale image patches. Visible effects appear reversed because image changes occur before mapping to geometry Order: translation, rotation about center, non-uniform scale about center. Hint: insert Shape and Appearance nodes before adding TextureTransform.
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 SFVec2f CDATA "0 0"]** Lateral/vertical shift in 2D (s,t) texture coordinates (opposite effect appears on geometry).
center	**[center: accessType inputOutput, type SFVec2f CDATA "0 0"]** center point in 2D (s,t) texture coordinates for rotation and scaling.
rotation	**[rotation: accessType inputOutput, type SFFloat CDATA "0"]** single rotation angle of texture about center (opposite effect appears on geometry). Warning: use a single radian angle value, not a 4-tuple Rotation.
scale	**[scale: accessType inputOutput, type SFVec2f CDATA "1 1"]** Non-uniform planar scaling of texture about center (opposite effect appears on geometry).
containerField	**[containerField: NMTOKEN "textureTransform"]** 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#TextureTransform



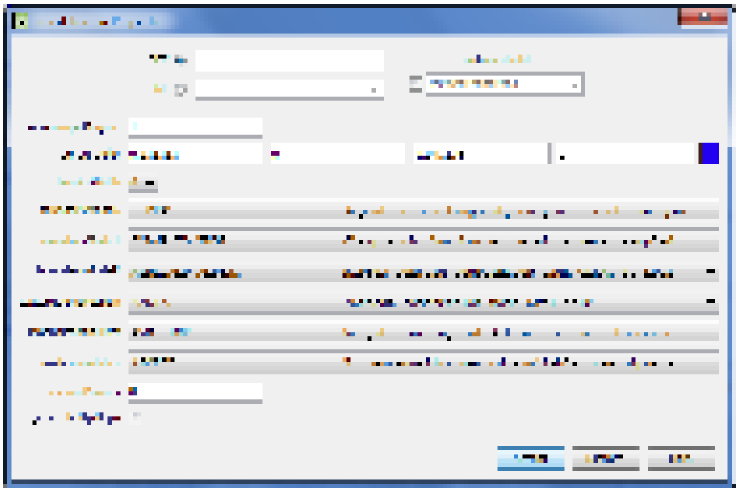
TextureProperties node

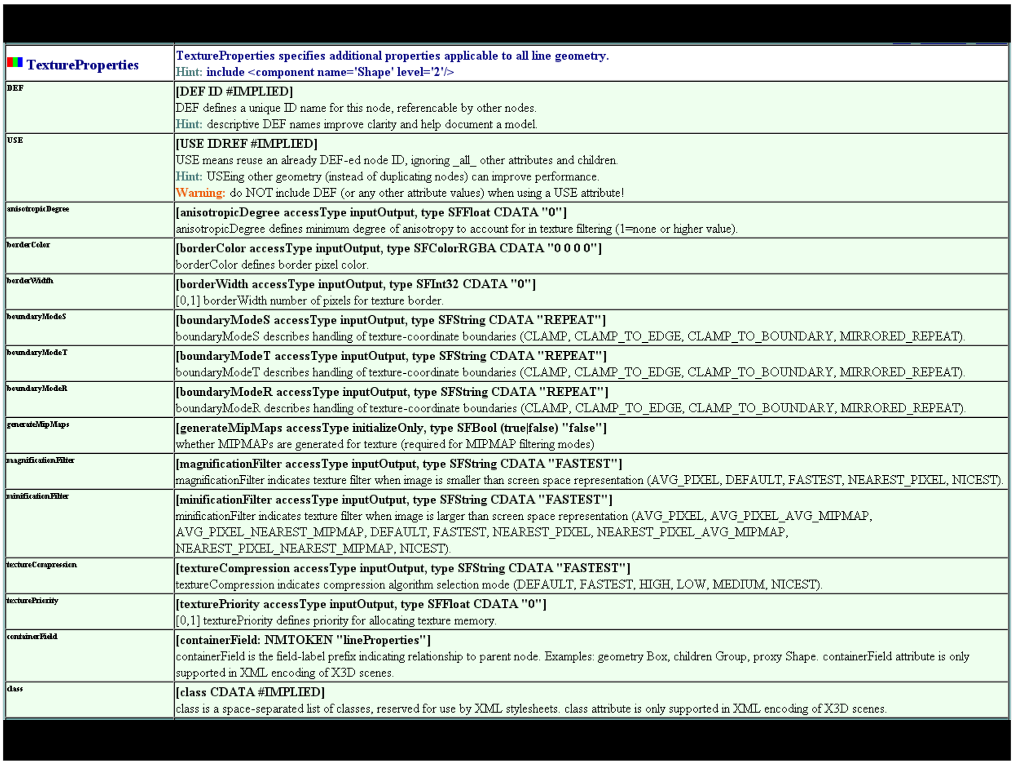
TextureProperties specifies additional properties applicable to all line geometry

- *anisotropicDegree* texture filtering
- *borderColor* SFColorRGBA 3-tuple color plus alpha
- *borderWidth* pixel boundary [0..1]
- *boundaryModeS, boundaryModeT, boundaryModeR*: CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT
- Others: *magnificationFilter, minificationFilter, textureCompression, texturePriority, generateMipMaps*
- Hint: include

```
<component name='Shape' level='2'/>
```

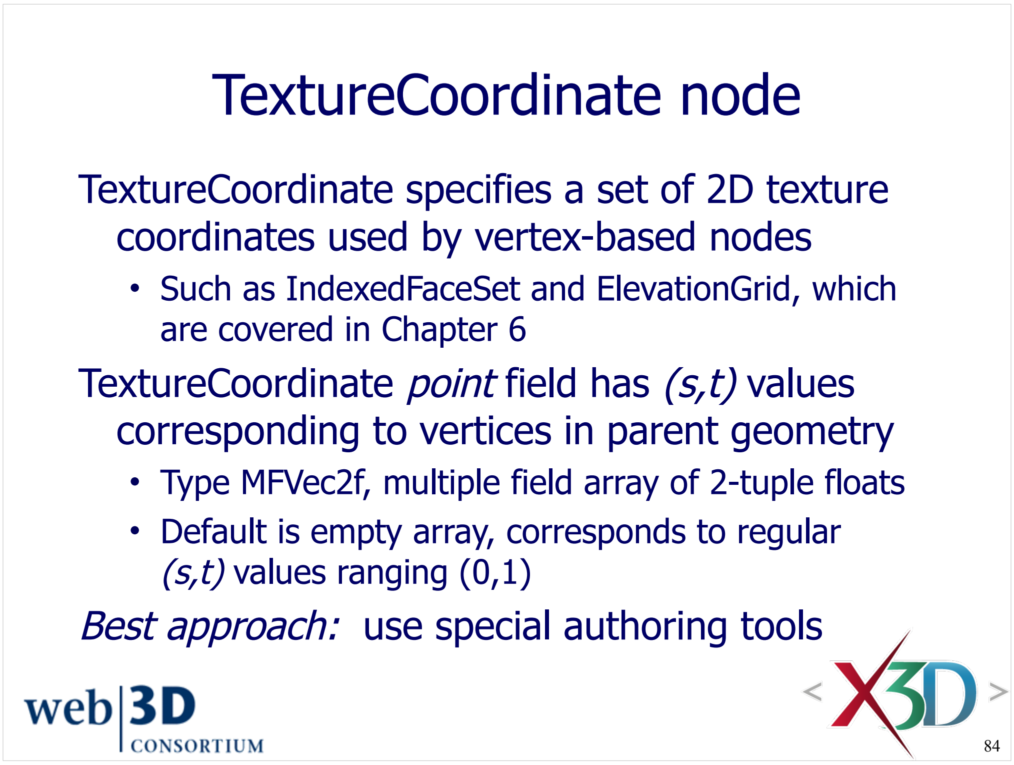
TextureProperties interface





🟥🟩🟦 TextureProperties	**TextureProperties specifies additional properties applicable to all line geometry.** Hint: **include <component name='Shape' level='2'/>**
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!
anisotropicDegree	**[anisotropicDegree accessType inputOutput, type SFFloat CDATA "0"]** anisotropicDegree defines minimum degree of anisotropy to account for in texture filtering (1=none or higher value).
borderColor	**[borderColor accessType inputOutput, type SFColorRGBA CDATA "0 0 0 0"]** borderColor defines border pixel color.
borderWidth	**[borderWidth accessType inputOutput, type SFInt32 CDATA "0"]** [0,1] borderWidth number of pixels for texture border.
boundaryModeS	**[boundaryModeS accessType inputOutput, type SFString CDATA "REPEAT"]** boundaryModeS describes handling of texture-coordinate boundaries (CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT).
boundaryModeT	**[boundaryModeT accessType inputOutput, type SFString CDATA "REPEAT"]** boundaryModeT describes handling of texture-coordinate boundaries (CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT).
boundaryModeR	**[boundaryModeR accessType inputOutput, type SFString CDATA "REPEAT"]** boundaryModeR describes handling of texture-coordinate boundaries (CLAMP, CLAMP_TO_EDGE, CLAMP_TO_BOUNDARY, MIRRORED_REPEAT).
generateMipMaps	**[generateMipMaps accessType initializeOnly, type SFBool (true|false) "false"]** whether MIPMAPs are generated for texture (required for MIPMAP filtering modes)
magnificationFilter	**[magnificationFilter accessType inputOutput, type SFString CDATA "FASTEST"]** magnificationFilter indicates texture filter when image is smaller than screen space representation (AVG_PIXEL, DEFAULT, FASTEST, NEAREST_PIXEL, NICEST).
minificationFilter	**[minificationFilter accessType inputOutput, type SFString CDATA "FASTEST"]** minificationFilter indicates texture filter when image is larger than screen space representation (AVG_PIXEL, AVG_PIXEL_AVG_MIPMAP, AVG_PIXEL_NEAREST_MIPMAP, DEFAULT, FASTEST, NEAREST_PIXEL, NEAREST_PIXEL_AVG_MIPMAP, NEAREST_PIXEL_NEAREST_MIPMAP, NICEST).
textureCompression	**[textureCompression accessType inputOutput, type SFString CDATA "FASTEST"]** textureCompression indicates compression algorithm selection mode (DEFAULT, FASTEST, HIGH, LOW, MEDIUM, NICEST).
texturePriority	**[texturePriority accessType inputOutput, type SFFloat CDATA "0"]** [0,1] texturePriority defines priority for allocating texture memory.
containerField	**[containerField: NMTOKEN "lineProperties"]** 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#TextureProperties



TextureCoordinate node

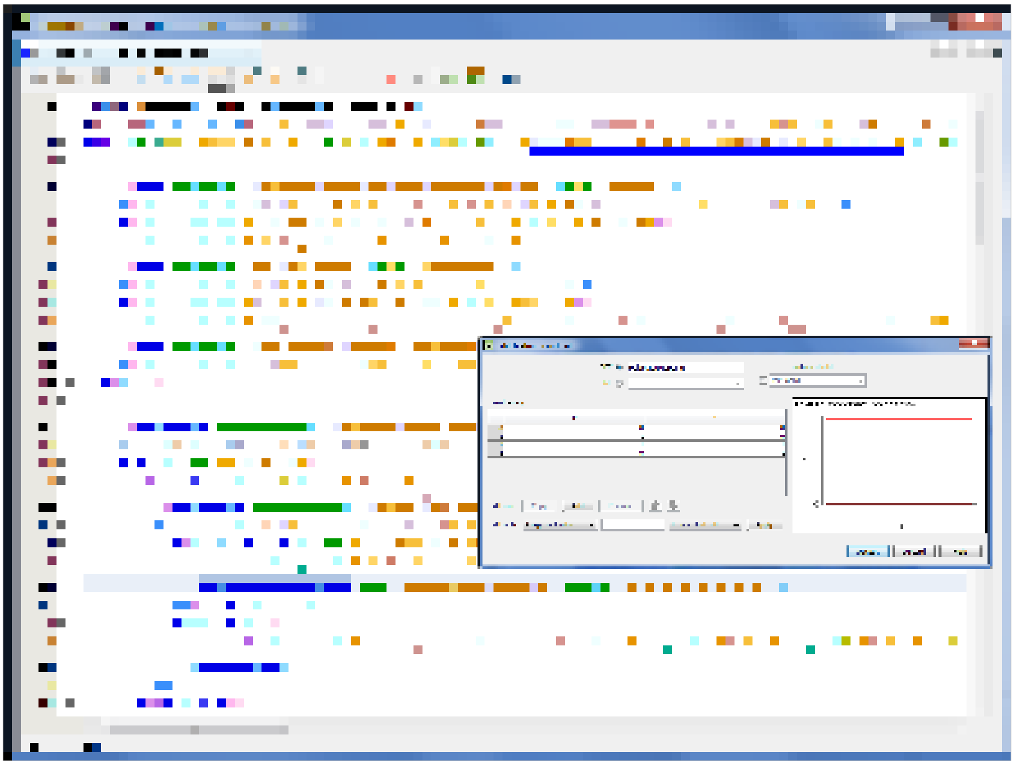
TextureCoordinate specifies a set of 2D texture coordinates used by vertex-based nodes

- Such as IndexedFaceSet and ElevationGrid, which are covered in Chapter 6

TextureCoordinate *point* field has *(s,t)* values corresponding to vertices in parent geometry

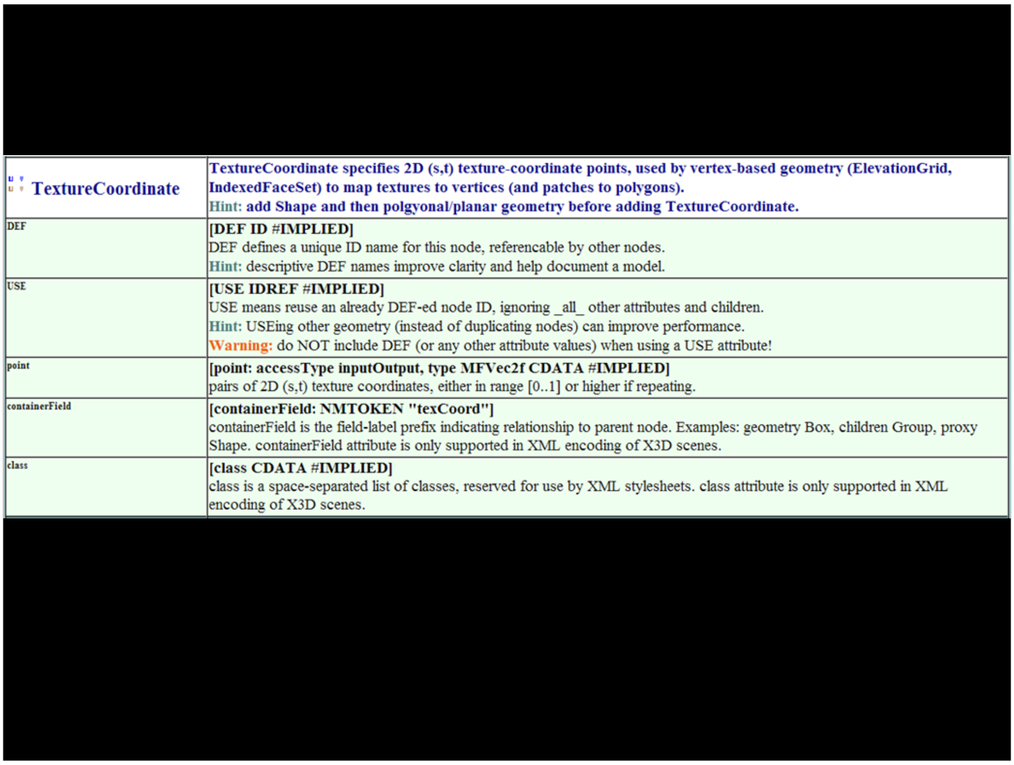
- Type MFVec2f, multiple field array of 2-tuple floats
- Default is empty array, corresponds to regular *(s,t)* values ranging (0,1)

Best approach: use special authoring tools



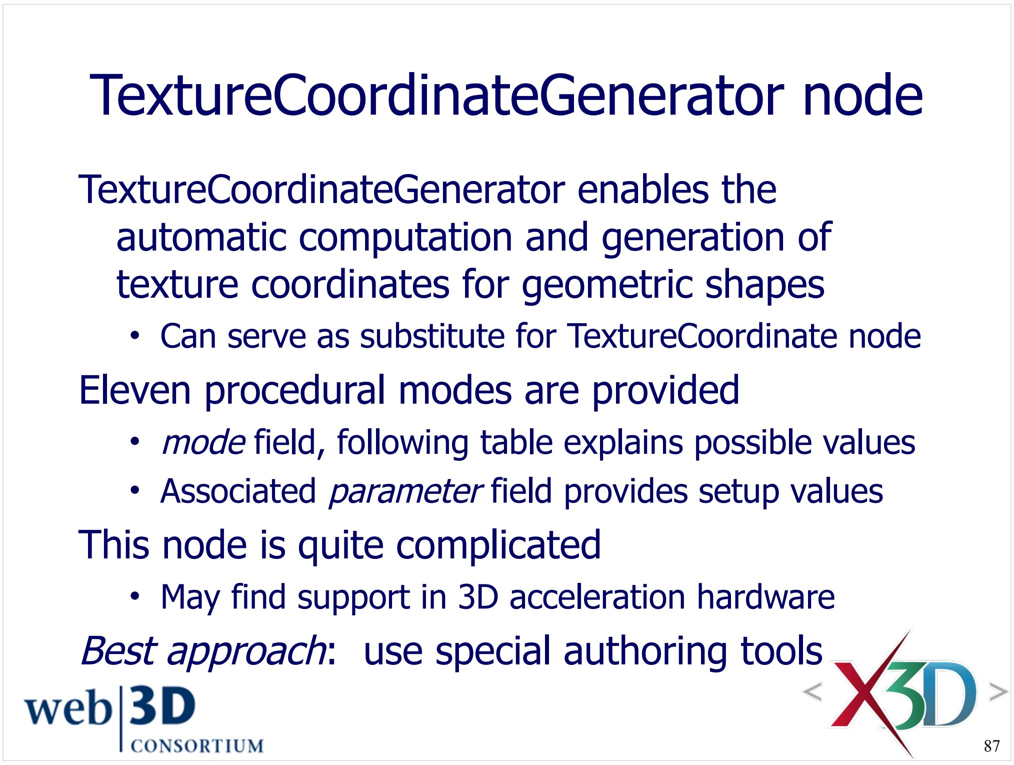
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/MovieTextureAuthoringOptions.x3d

TODO: open up window on plot 5-10% to avoid overwriting axes



TextureCoordinate	**TextureCoordinate specifies 2D (s,t) texture-coordinate points, used by vertex-based geometry (ElevationGrid, IndexedFaceSet) to map textures to vertices (and patches to polygons).** **Hint: add Shape and then polgyonal/planar geometry before adding TextureCoordinate.**
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!
point	**[point: accessType inputOutput, type MFVec2f CDATA #IMPLIED]** pairs of 2D (s,t) texture coordinates, either in range [0..1] or higher if repeating.
containerField	**[containerField: NMTOKEN "texCoord"]** 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#TextureCoordinate



TextureCoordinateGenerator node

TextureCoordinateGenerator enables the automatic computation and generation of texture coordinates for geometric shapes

- Can serve as substitute for TextureCoordinate node

Eleven procedural modes are provided

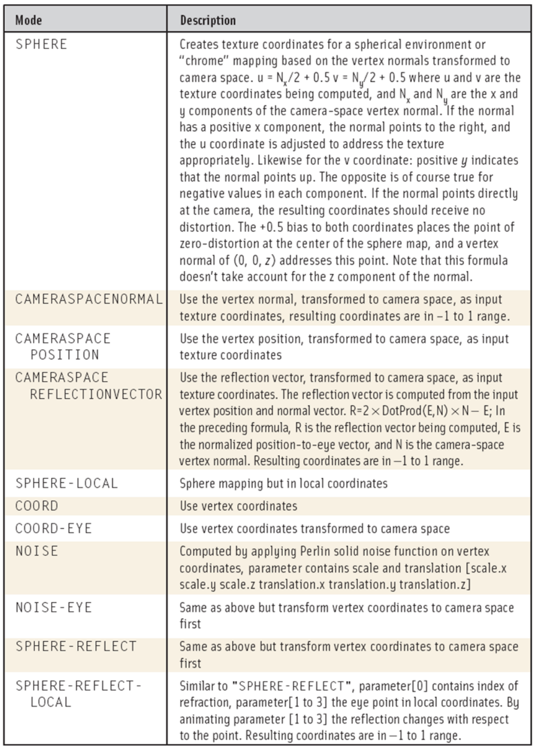
- *mode* field, following table explains possible values
- Associated *parameter* field provides setup values

This node is quite complicated

- May find support in 3D acceleration hardware

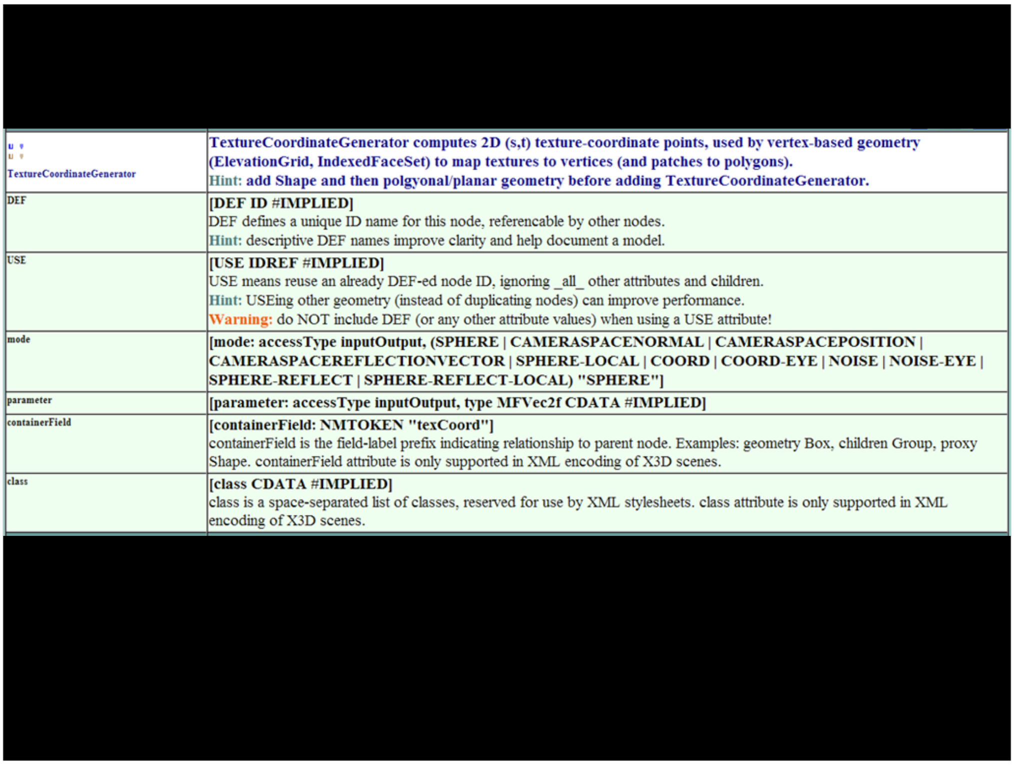
Best approach: use special authoring tools

TextureCoordinateGenerator *mode* enumerations and *parameter* values



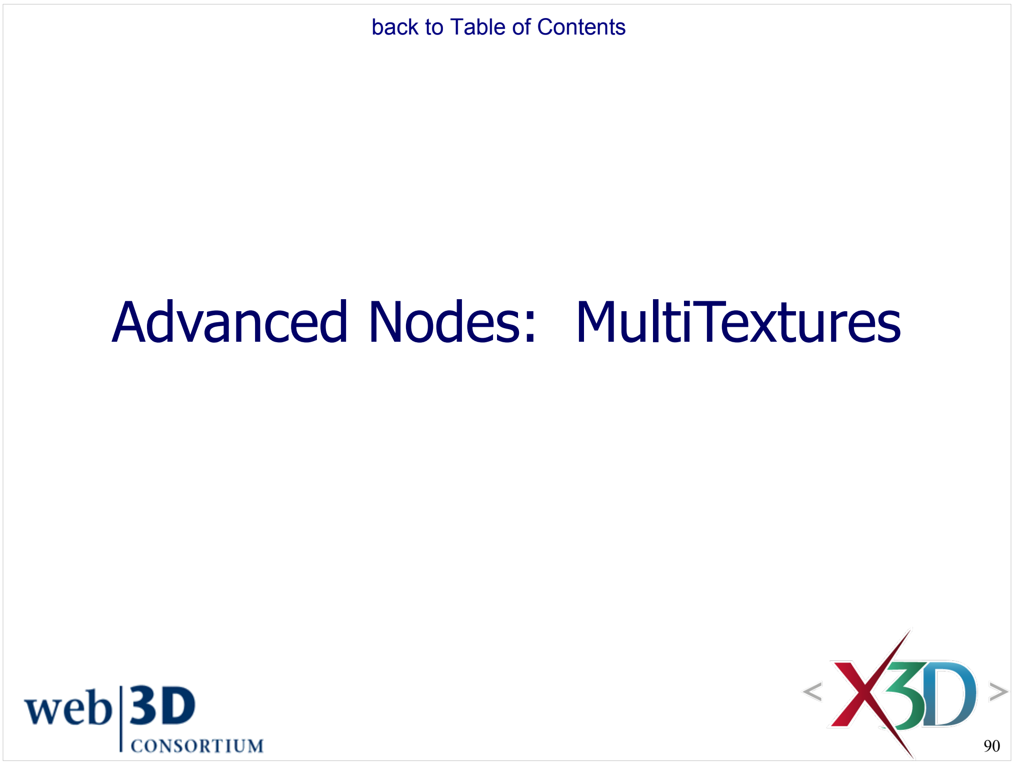
Mode	Description
SPHERE	Creates texture coordinates for a spherical environment or "chrome" mapping based on the vertex normals transformed to camera space. $u = N_x/2 + 0.5$ $v = N_y/2 + 0.5$ where u and v are the texture coordinates being computed, and N_x and N_y are the x and y components of the camera-space vertex normal. If the normal has a positive x component, the normal points to the right, and the u coordinate is adjusted to address the texture appropriately. Likewise for the v coordinate: positive *y* indicates that the normal points up. The opposite is of course true for negative values in each component. If the normal points directly at the camera, the resulting coordinates should receive no distortion. The +0.5 bias to both coordinates places the point of zero-distortion at the center of the sphere map, and a vertex normal of (0, 0, *z*) addresses this point. Note that this formula doesn't take account for the z component of the normal.
CAMERASPACENORMAL	Use the vertex normal, transformed to camera space, as input texture coordinates, resulting coordinates are in –1 to 1 range.
CAMERASPACEPOSITION	Use the vertex position, transformed to camera space, as input texture coordinates
CAMERASPACEREFLECTIONVECTOR	Use the reflection vector, transformed to camera space, as input texture coordinates. The reflection vector is computed from the input vertex position and normal vector. $R=2 \times DotProd(E,N) \times N - E$; In the preceding formula, R is the reflection vector being computed, E is the normalized position-to-eye vector, and N is the camera-space vertex normal. Resulting coordinates are in –1 to 1 range.
SPHERE-LOCAL	Sphere mapping but in local coordinates
COORD	Use vertex coordinates
COORD-EYE	Use vertex coordinates transformed to camera space
NOISE	Computed by applying Perlin solid noise function on vertex coordinates, parameter contains scale and translation [scale.x scale.y scale.z translation.x translation.y translation.z]
NOISE-EYE	Same as above but transform vertex coordinates to camera space first
SPHERE-REFLECT	Same as above but transform vertex coordinates to camera space first
SPHERE-REFLECT-LOCAL	Similar to "SPHERE-REFLECT", parameter[0] contains index of refraction, parameter[1 to 3] the eye point in local coordinates. By animating parameter [1 to 3] the reflection changes with respect to the point. Resulting coordinates are in –1 to 1 range.

X3D for Web Authors, Table 5.25, p.154



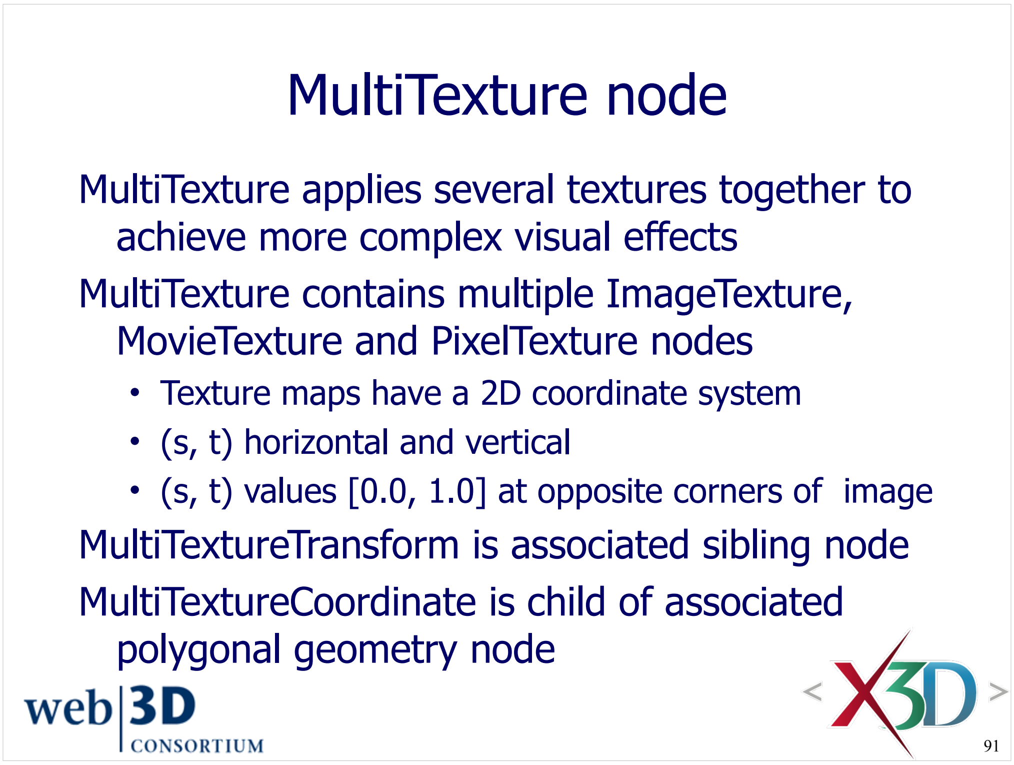
TextureCoordinateGenerator	**TextureCoordinateGenerator computes 2D (s,t) texture-coordinate points, used by vertex-based geometry (ElevationGrid, IndexedFaceSet) to map textures to vertices (and patches to polygons).** **Hint: add Shape and then polgyonal/planar geometry before adding TextureCoordinateGenerator.**
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!
mode	**[mode: accessType inputOutput, (SPHERE | CAMERASPACENORMAL | CAMERASPACEPOSITION | CAMERASPACEREFLECTIONVECTOR | SPHERE-LOCAL | COORD | COORD-EYE | NOISE | NOISE-EYE | SPHERE-REFLECT | SPHERE-REFLECT-LOCAL) "SPHERE"]**
parameter	**[parameter: accessType inputOutput, type MFVec2f CDATA #IMPLIED]**
containerField	**[containerField: NMTOKEN "texCoord"]** 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#TextureCoordinateGenerator



back to Table of Contents

Advanced Nodes: MultiTextures



MultiTexture node

MultiTexture applies several textures together to achieve more complex visual effects

MultiTexture contains multiple ImageTexture, MovieTexture and PixelTexture nodes

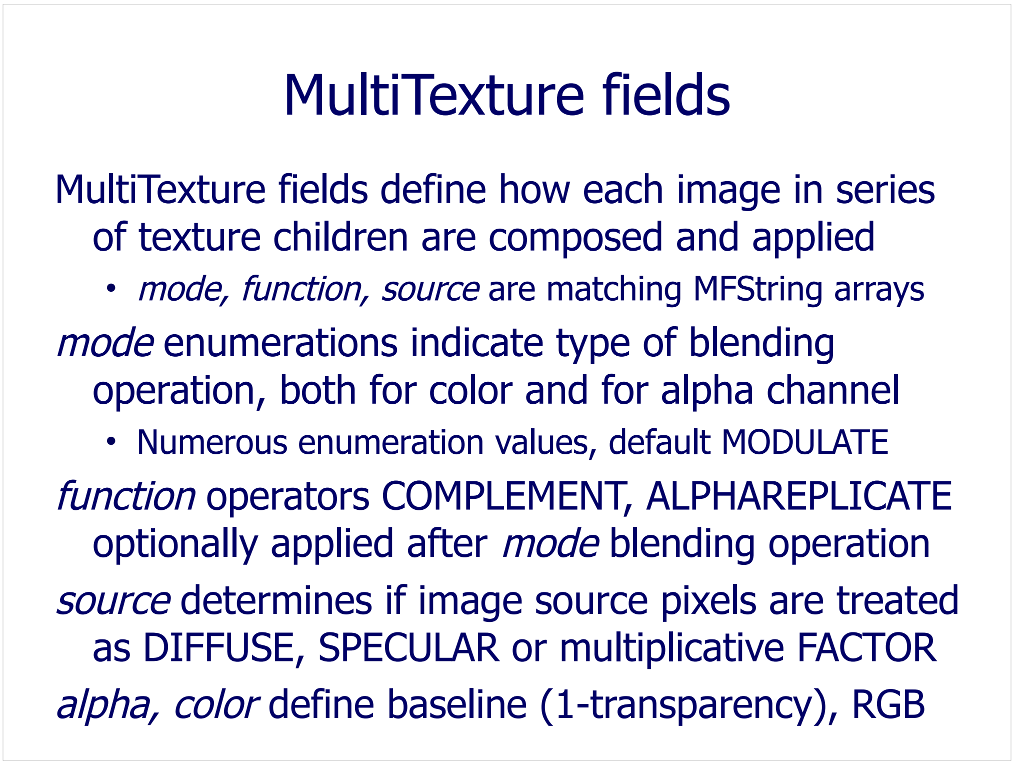
- Texture maps have a 2D coordinate system
- (s, t) horizontal and vertical
- (s, t) values [0.0, 1.0] at opposite corners of image

MultiTextureTransform is associated sibling node

MultiTextureCoordinate is child of associated polygonal geometry node

MultiTexture node in X3D Specification:

http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/Part01/components/texturing.html#MultiTexture



MultiTexture fields

MultiTexture fields define how each image in series of texture children are composed and applied

- *mode, function, source* are matching MFString arrays

mode enumerations indicate type of blending operation, both for color and for alpha channel

- Numerous enumeration values, default MODULATE

function operators COMPLEMENT, ALPHAREPLICATE optionally applied after *mode* blending operation

source determines if image source pixels are treated as DIFFUSE, SPECULAR or multiplicative FACTOR

alpha, color define baseline (1-transparency), RGB

MultiTexture node enumeration value tables are defined in X3D specification

Table 18.3 — Values for the *mode* field

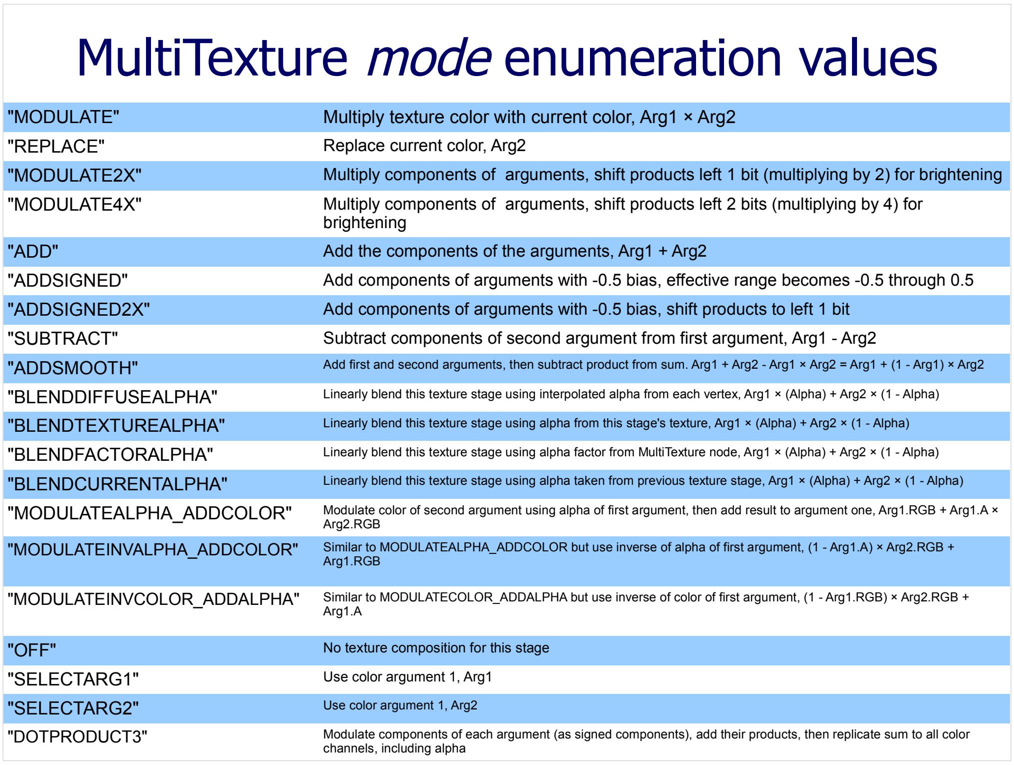
http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/Part01/components/texturing.html#t-multitexturemodes

Table 18.4 — Values for the *source* field

http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/Part01/components/texturing.html#t-ValuesForSourceField

Table 18.5 — Values for the *function* field

http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/Part01/components/texturing.html#t-ValuesForFunctionField



MultiTexture *mode* enumeration values

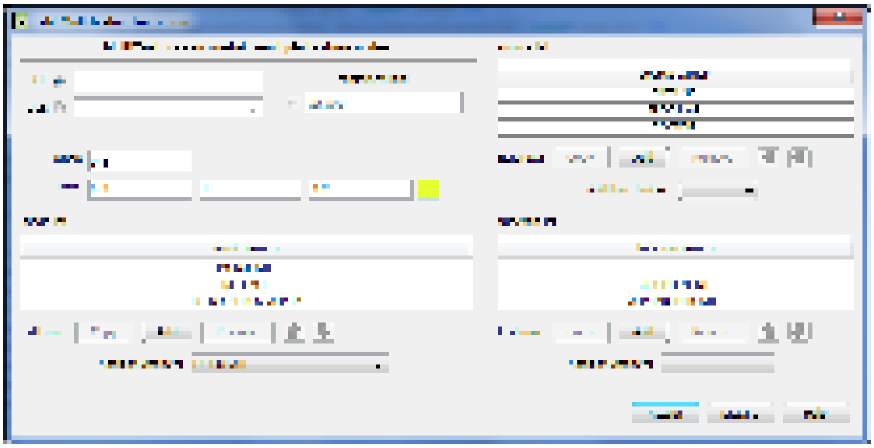
"MODULATE"	Multiply texture color with current color, Arg1 × Arg2
"REPLACE"	Replace current color, Arg2
"MODULATE2X"	Multiply components of arguments, shift products left 1 bit (multiplying by 2) for brightening
"MODULATE4X"	Multiply components of arguments, shift products left 2 bits (multiplying by 4) for brightening
"ADD"	Add the components of the arguments, Arg1 + Arg2
"ADDSIGNED"	Add components of arguments with -0.5 bias, effective range becomes -0.5 through 0.5
"ADDSIGNED2X"	Add components of arguments with -0.5 bias, shift products to left 1 bit
"SUBTRACT"	Subtract components of second argument from first argument, Arg1 - Arg2
"ADDSMOOTH"	Add first and second arguments, then subtract product from sum. Arg1 + Arg2 - Arg1 × Arg2 = Arg1 + (1 - Arg1) × Arg2
"BLENDDIFFUSEALPHA"	Linearly blend this texture stage using interpolated alpha from each vertex, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"BLENDTEXTUREALPHA"	Linearly blend this texture stage using alpha from this stage's texture, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"BLENDFACTORALPHA"	Linearly blend this texture stage using alpha factor from MultiTexture node, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"BLENDCURRENTALPHA"	Linearly blend this texture stage using alpha taken from previous texture stage, Arg1 × (Alpha) + Arg2 × (1 - Alpha)
"MODULATEALPHA_ADDCOLOR"	Modulate color of second argument using alpha of first argument, then add result to argument one, Arg1.RGB + Arg1.A × Arg2.RGB
"MODULATEINVALPHA_ADDCOLOR"	Similar to MODULATEALPHA_ADDCOLOR but use inverse of alpha of first argument, (1 - Arg1.A) × Arg2.RGB + Arg1.RGB
"MODULATEINVCOLOR_ADDALPHA"	Similar to MODULATECOLOR_ADDALPHA but use inverse of color of first argument, (1 - Arg1.RGB) × Arg2.RGB + Arg1.A
"OFF"	No texture composition for this stage
"SELECTARG1"	Use color argument 1, Arg1
"SELECTARG2"	Use color argument 1, Arg2
"DOTPRODUCT3"	Modulate components of each argument (as signed components), add their products, then replicate sum to all color channels, including alpha

X3D specification, Table 18.3 — Multitexture modes

http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/Part01/components/texturing.html#t-multitexturemodes

Shape
IndexedFaceSet
MultiTextureCoordinate
TextureCoordinate
TextureCoordinate
TextureCoordinate
Appearance
Material
MultiTexture
ImageTexture
MovieTexture
PixelTexture
MultiTextureTransform
TextureTransform
TextureTransform
TextureTransform

MultiTexture editing panel



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

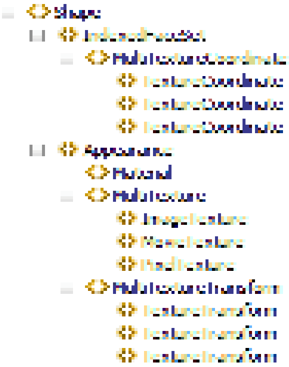
MultiTextureTransform

MultiTextureTransform contains multiple child TextureTransform nodes

Each of the contained TextureTransform nodes correspond to the multiple texture nodes found in sibling MultiTexture node

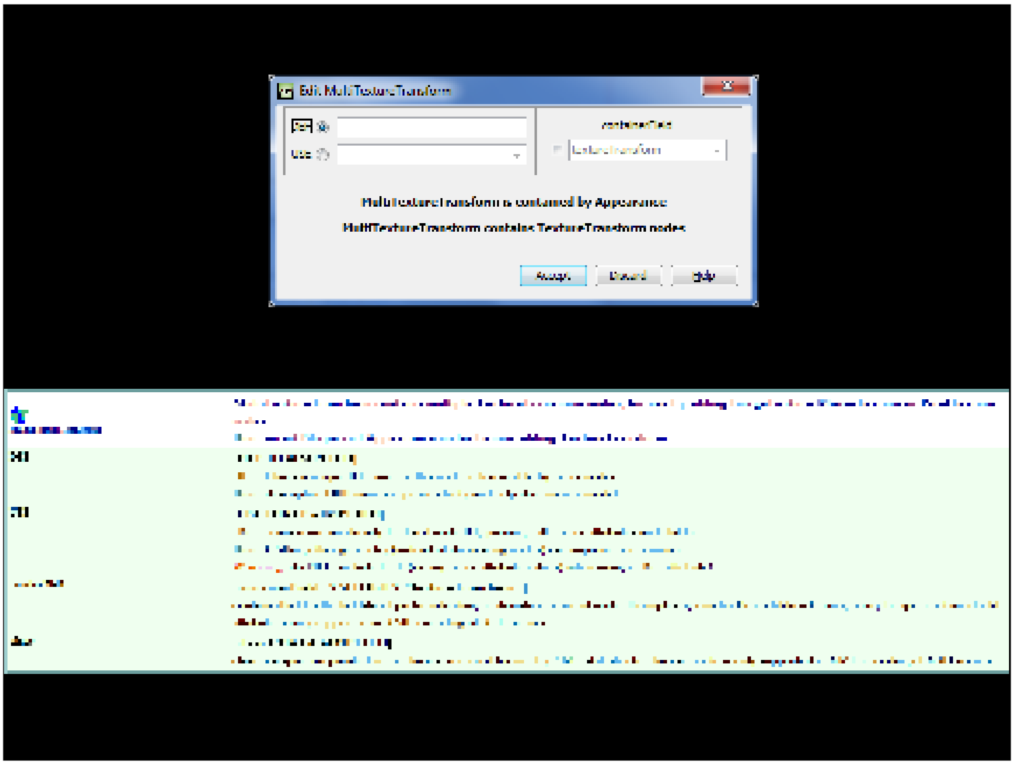
- ImageTexture MovieTexture or PixelTexture

Thus supports application of multiple textures



```
Shape
  IndexedFaceSet
    MultiTextureCoordinate
      TextureCoordinate
      TextureCoordinate
      TextureCoordinate
  Appearance
    Material
    MultiTexture
      ImageTexture
      MovieTexture
      PixelTexture
    MultiTextureTransform
      TextureTransform
      TextureTransform
      TextureTransform
```

web 3D CONSORTIUM



MultiTextureTransform is contained by Appearance

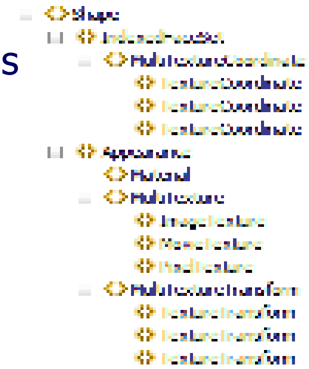
MultiTextureTransform contains TextureTransform nodes

[illegible]

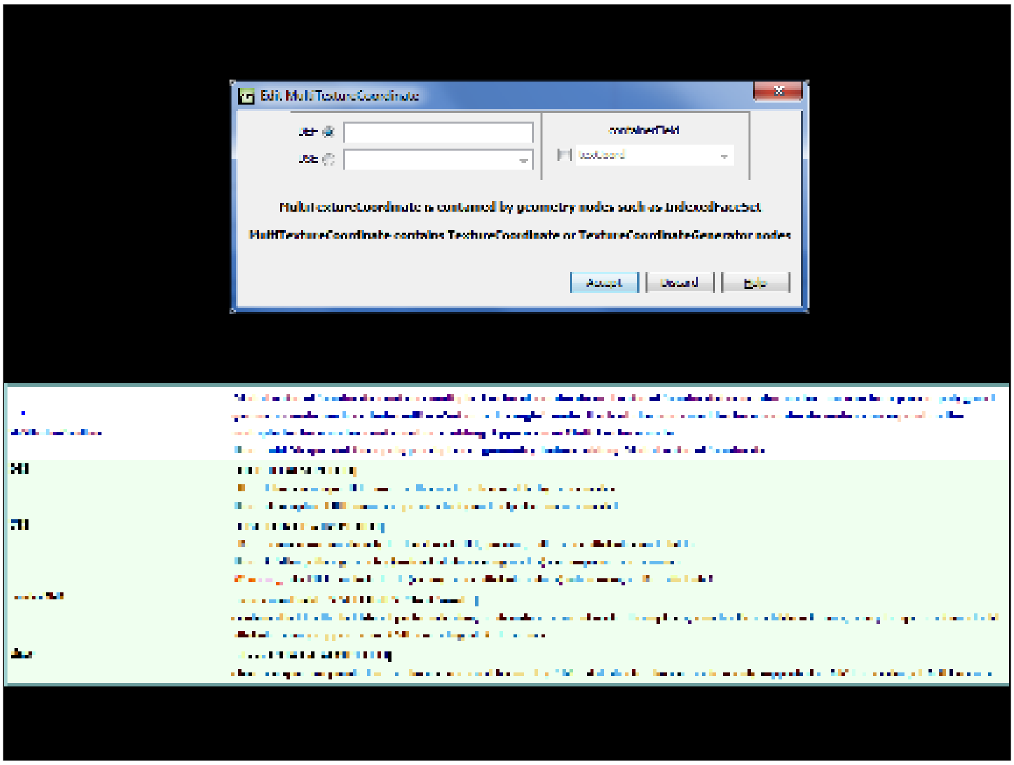
http://www.web3d.org/x3d/content/X3dTooltips.html#TextureCoordinateGenerator

MultiTextureCoordinate

- MultiTextureCoordinate contains multiple child TextureCoordinate, TextureCoordinateGenerator nodes
  - for use by a parent polygonal geometry node such as IndexedFaceSet or Triangle* node
- Contained TextureCoordinate nodes correspond to texture nodes
  - which in turn are contained in sibling Appearance/MultiTexture



- Shape
  - IndexedFaceSet
    - MultiTextureCoordinate
      - TextureCoordinate
      - TextureCoordinate
      - TextureCoordinate
  - Appearance
    - Material
    - MultiTexture
      - ImageTexture
      - MovieTexture
      - PixelTexture
    - MultiTextureTransform
      - TextureTransform
      - TextureTransform
      - TextureTransform

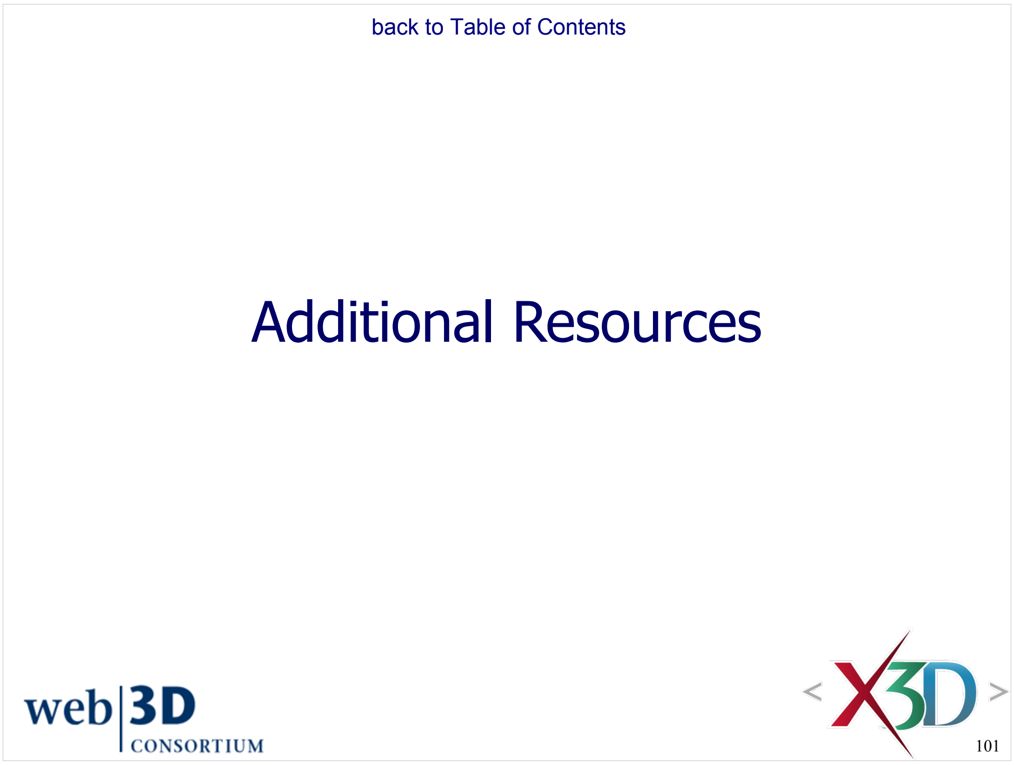


MultiTextureCoordinate is contained by geometry nodes such as IndexedFaceSet

MultiTextureCoordinate contains TextureCoordinate or TextureCoordinateGenerator nodes

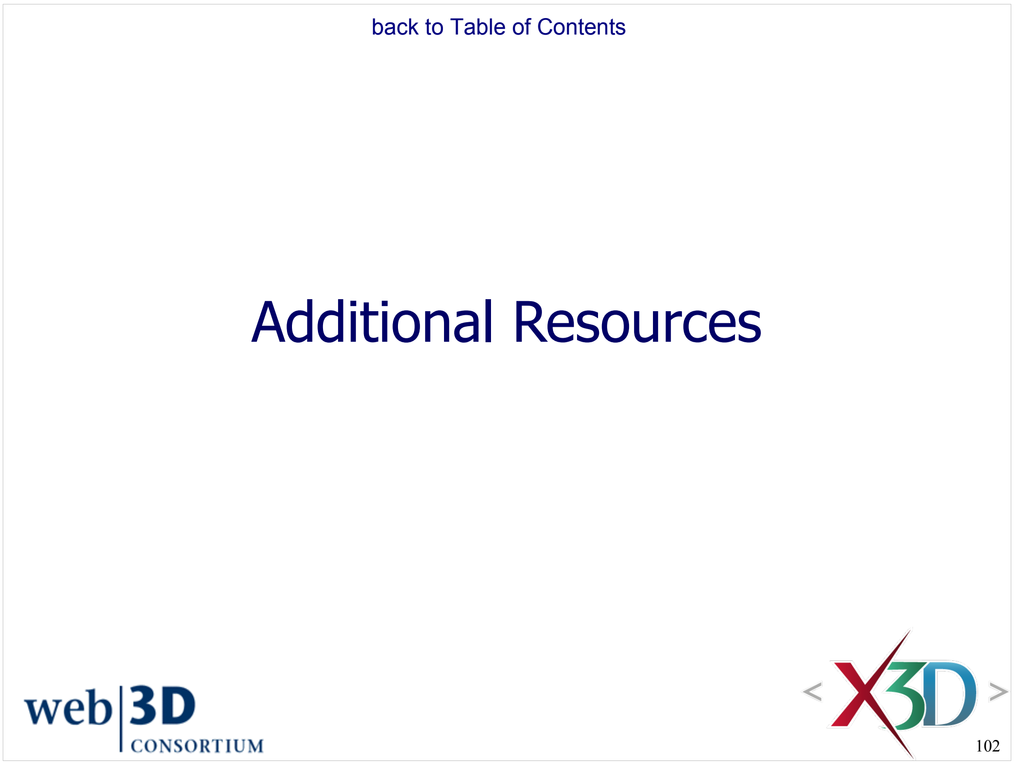
[illegible]

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



back to Table of Contents

Additional Resources



back to Table of Contents

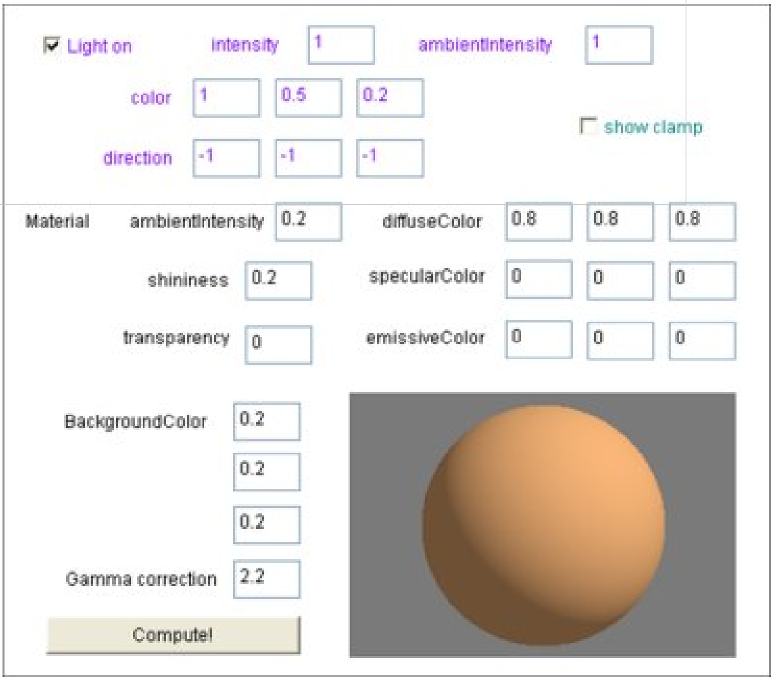
Additional Resources

Pellucid materials editor

Pellucid materials editor provides high-fidelity rendering of the VRML (X3D) materials model

- Eric Haines, copyright (c) 1997
- http://tog.acm.org/resources/applets/vrml/pellucid.html

web3D CONSORTIUM



☑ Light on intensity 1 ambientIntensity 1

color 1 0.5 0.2

direction -1 -1 -1

□ show clamp

Material ambientIntensity 0.2 diffuseColor 0.8 0.8 0.8

shininess 0.2 specularColor 0 0 0

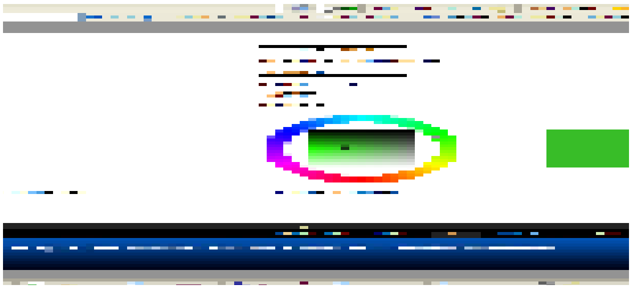
transparency 0 emissiveColor 0 0 0

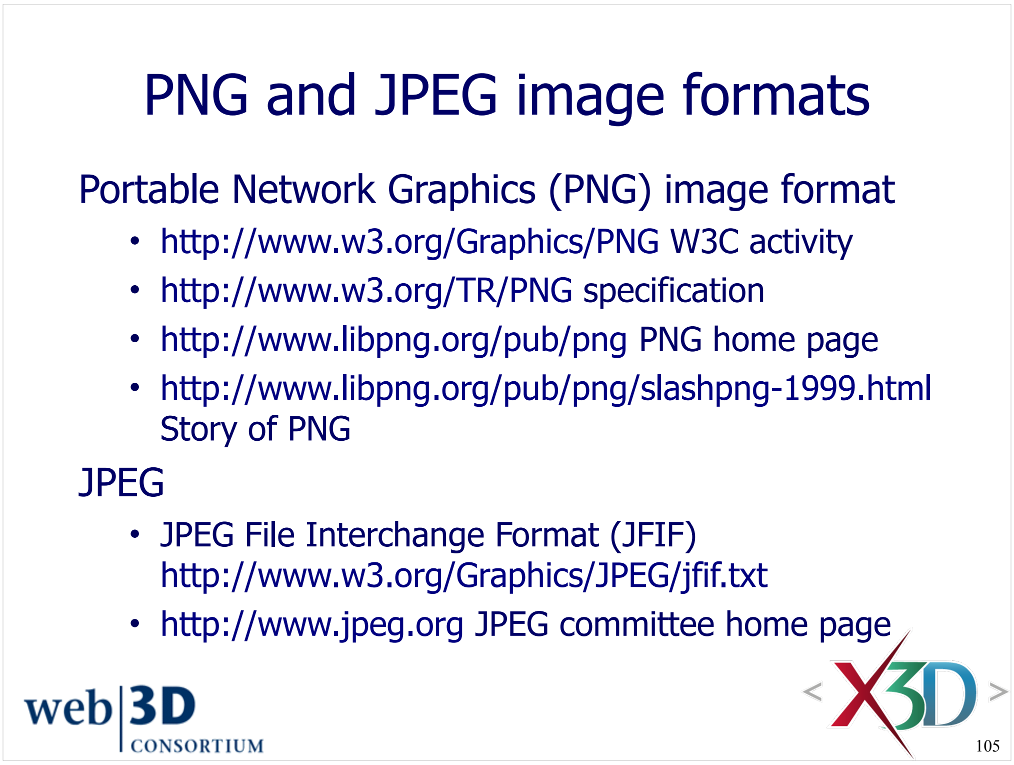
BackgroundColor 0.2 0.2 0.2

Gamma correction 2.2

InstantReality color calculator

- http://instantreality.org/tools/color_calculator





PNG and JPEG image formats

Portable Network Graphics (PNG) image format

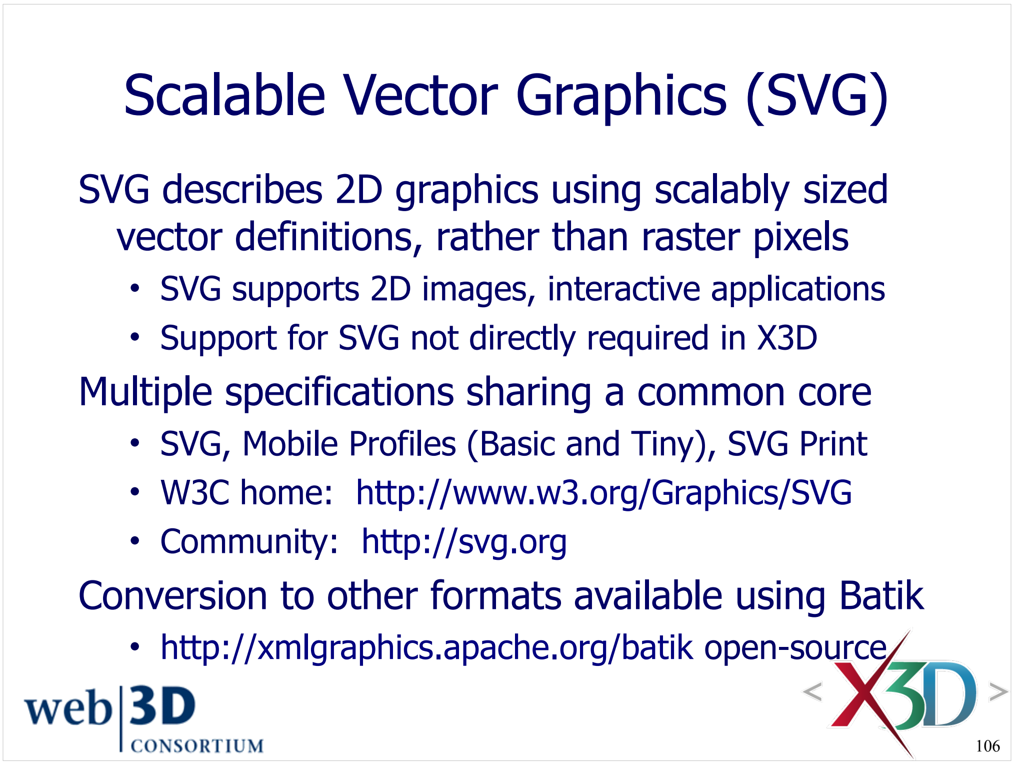
- http://www.w3.org/Graphics/PNG W3C activity
- http://www.w3.org/TR/PNG specification
- http://www.libpng.org/pub/png PNG home page
- http://www.libpng.org/pub/png/slashpng-1999.html Story of PNG

JPEG

- JPEG File Interchange Format (JFIF) http://www.w3.org/Graphics/JPEG/jfif.txt
- http://www.jpeg.org JPEG committee home page

JPEG 2000

- http://www.jpeg.org/jpeg2000



Scalable Vector Graphics (SVG)

SVG describes 2D graphics using scalably sized vector definitions, rather than raster pixels

- SVG supports 2D images, interactive applications
- Support for SVG not directly required in X3D

Multiple specifications sharing a common core

- SVG, Mobile Profiles (Basic and Tiny), SVG Print
- W3C home: http://www.w3.org/Graphics/SVG
- Community: http://svg.org

Conversion to other formats available using Batik

- http://xmlgraphics.apache.org/batik open-source

web3D CONSORTIUM

<X3D>

106

SVG support not is directly required by the X3D specification. Nevertheless, SVG support theoretically might be legally provided by any X3D player that wants to support SVG as an ImageTexture format.

Online support for converting SVG to PNG, JPEG or TIFF formats:

- http://www.fileformat.info/convert/image/svg2raster.htm
- This server exposes Batik functionality

SVG implementations

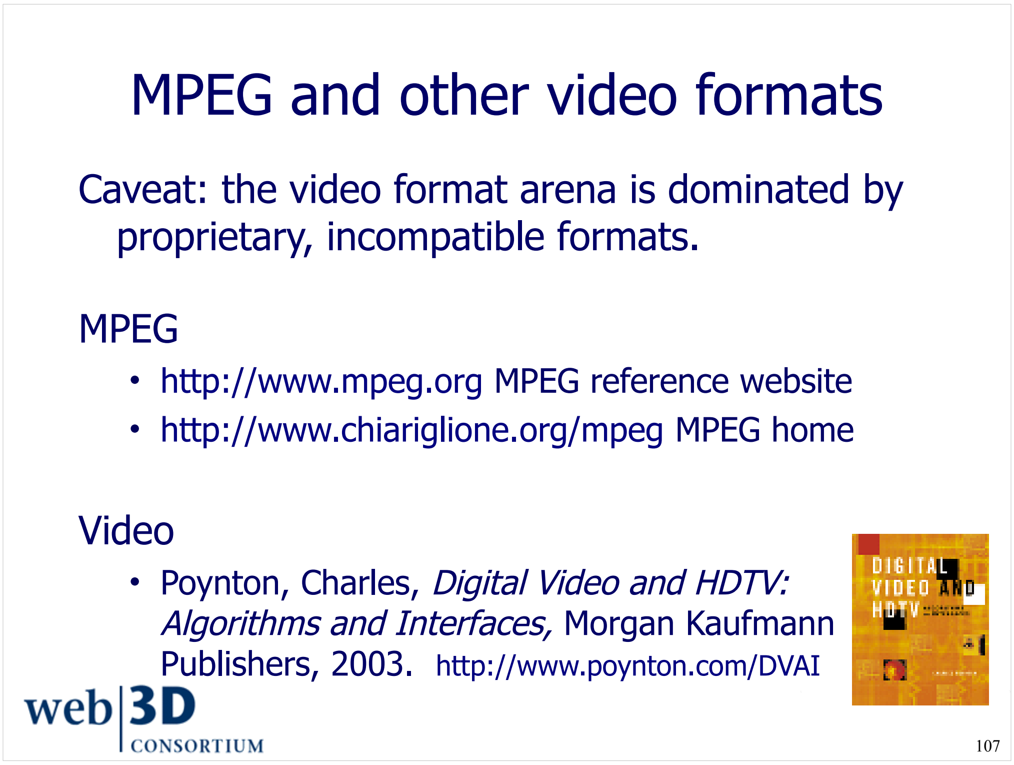
- http://wiki.svg.org/Viewer_Matrix

SVG support matrix for various browsers

- http://www.codedread.com/svg-support.php

SVG test suite

- http://www.w3.org/Graphics/SVG/Test



MPEG and other video formats

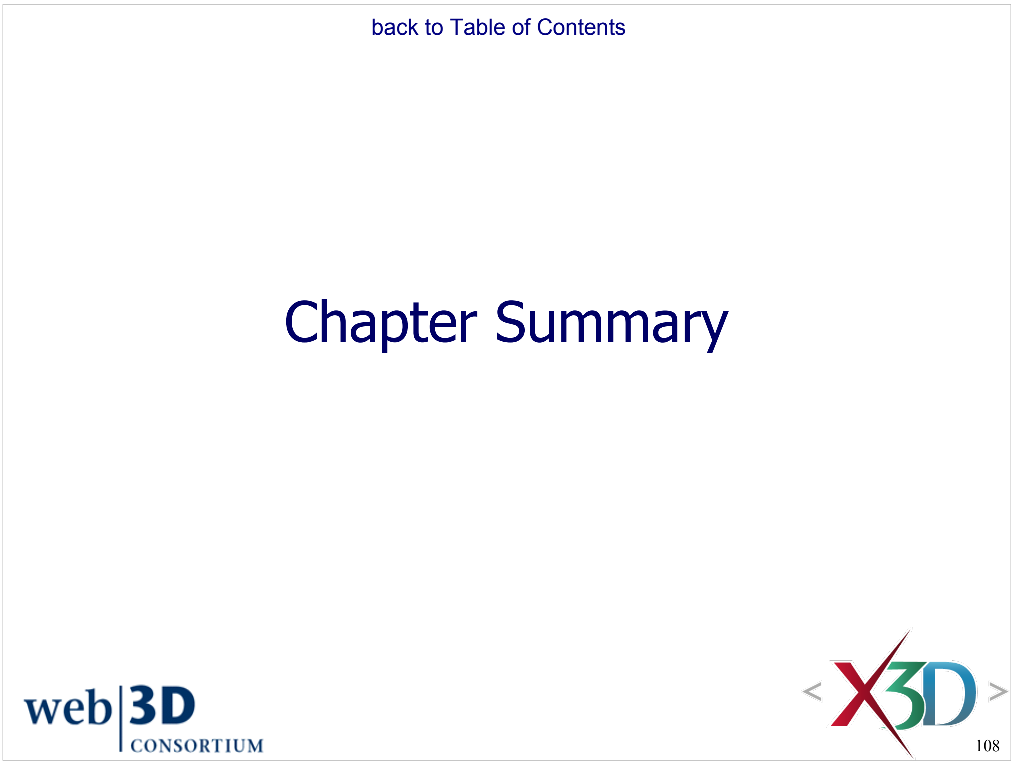
Caveat: the video format arena is dominated by proprietary, incompatible formats.

MPEG

- http://www.mpeg.org MPEG reference website
- http://www.chiariglione.org/mpeg MPEG home

Video

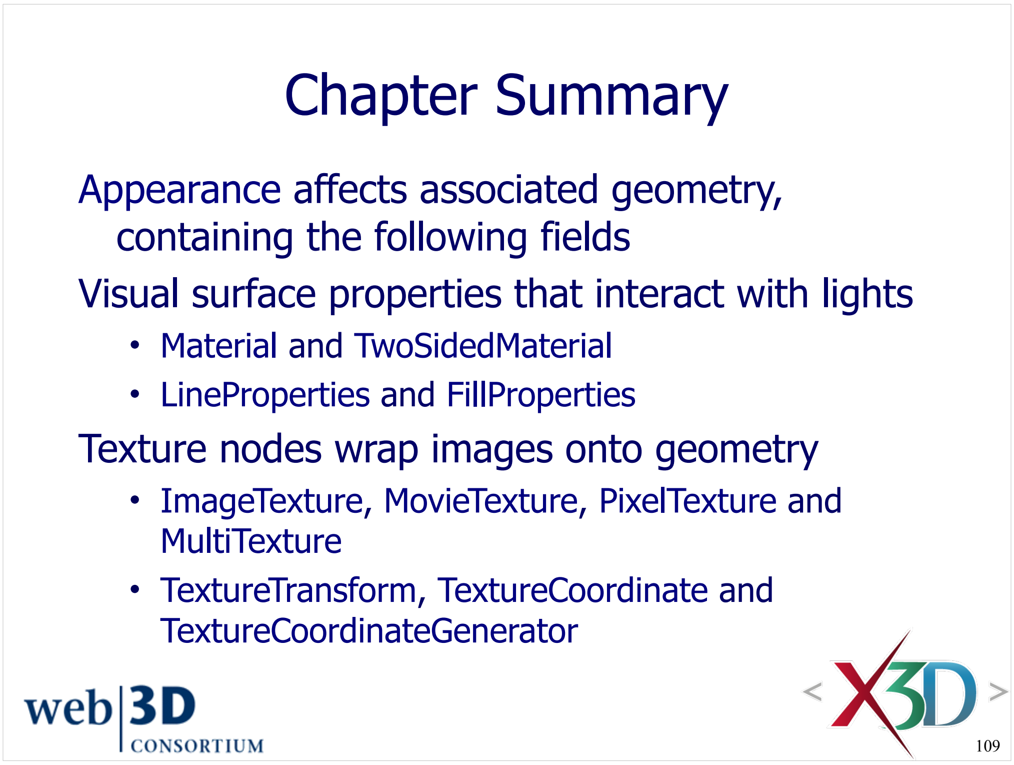
- Poynton, Charles, *Digital Video and HDTV: Algorithms and Interfaces,* Morgan Kaufmann Publishers, 2003. http://www.poynton.com/DVAI



back to Table of Contents

Chapter Summary

108



Chapter Summary

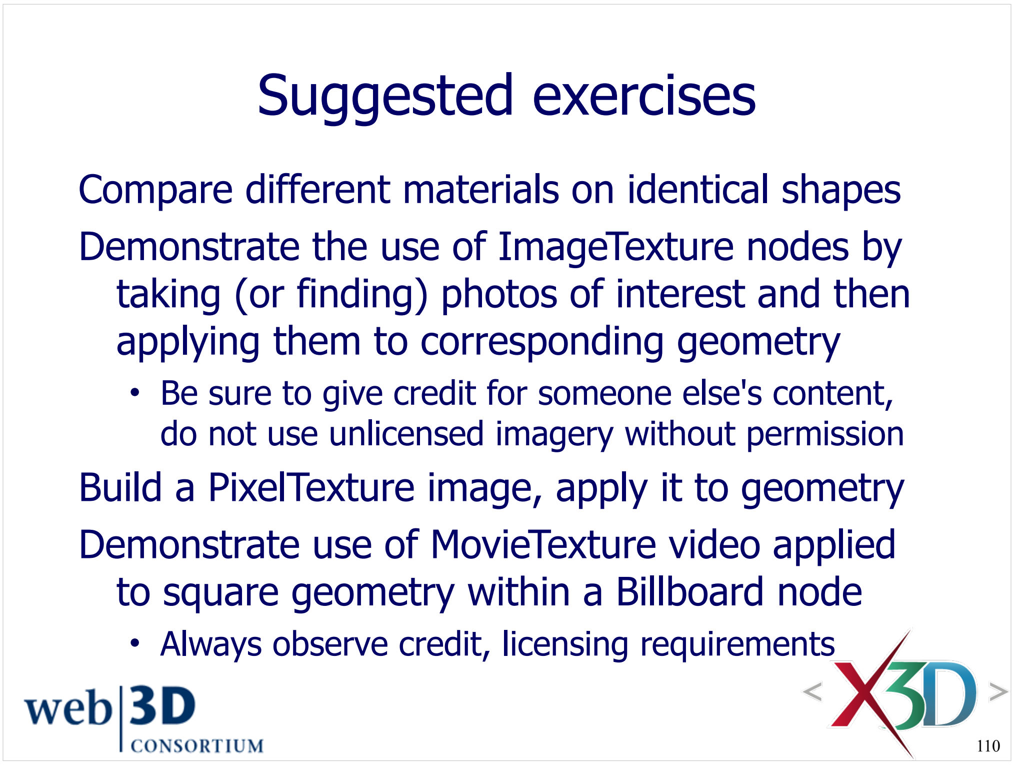
Appearance affects associated geometry, containing the following fields

Visual surface properties that interact with lights

- Material and TwoSidedMaterial
- LineProperties and FillProperties

Texture nodes wrap images onto geometry

- ImageTexture, MovieTexture, PixelTexture and MultiTexture
- TextureTransform, TextureCoordinate and TextureCoordinateGenerator



Suggested exercises

Compare different materials on identical shapes

Demonstrate the use of ImageTexture nodes by taking (or finding) photos of interest and then applying them to corresponding geometry

- Be sure to give credit for someone else's content, do not use unlicensed imagery without permission

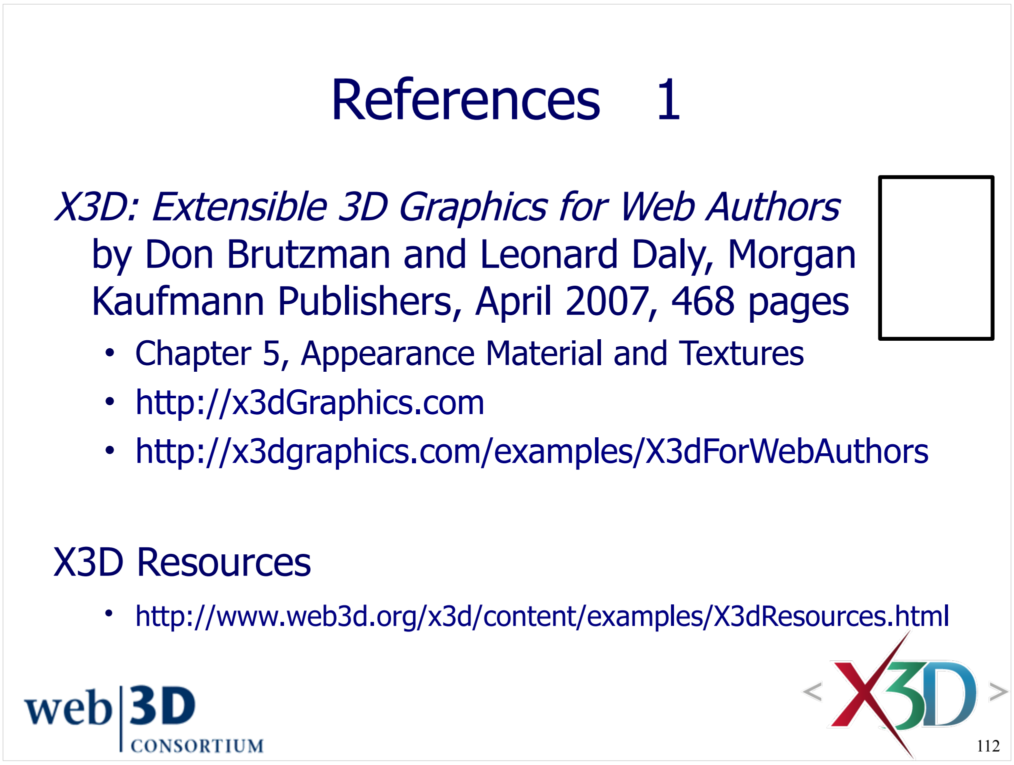
Build a PixelTexture image, apply it to geometry

Demonstrate use of MovieTexture video applied to square geometry within a Billboard node

- Always observe credit, licensing requirements

back to Table of Contents

References



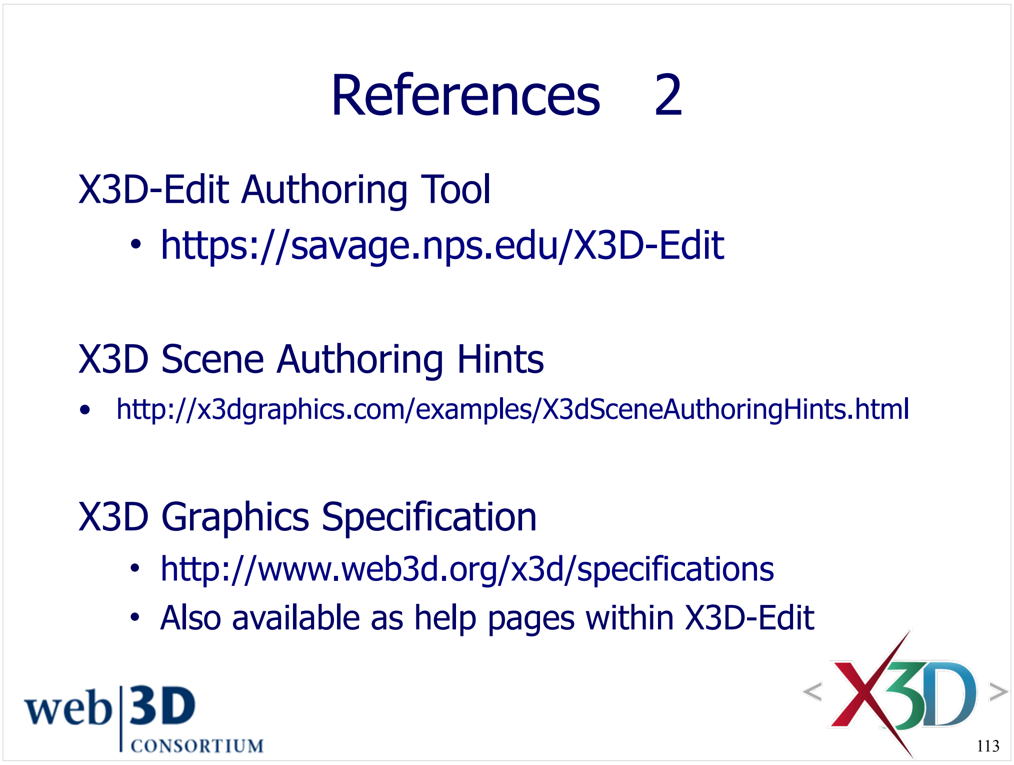
References 1

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

- Chapter 5, Appearance Material and Textures
- 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

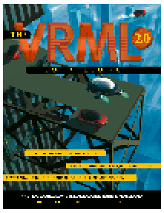
X3D Scene Authoring Hints

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

X3D Graphics Specification

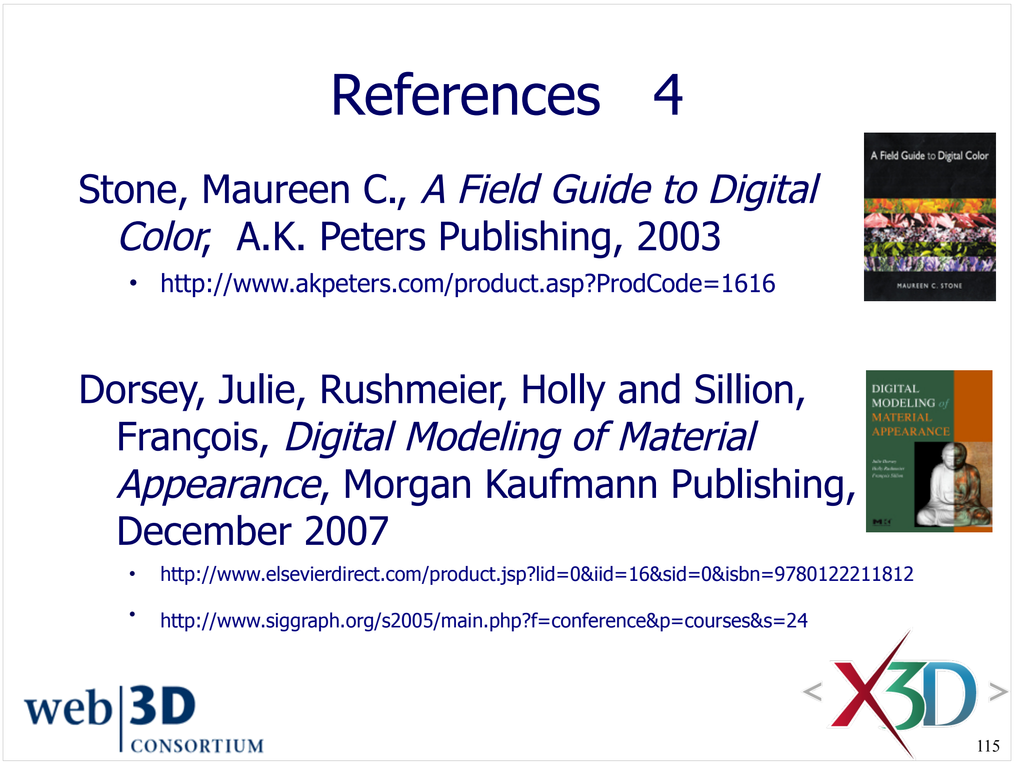
- http://www.web3d.org/x3d/specifications
- Also available as help pages within X3D-Edit

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 10 – Materials
- Chapter 17 – Textures
- Chapter 18 – Texture Mapping
- Chapter 21 – Shiny Materials



References 4

Stone, Maureen C., *A Field Guide to Digital Color*, A.K. Peters Publishing, 2003

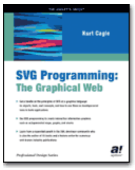
- http://www.akpeters.com/product.asp?ProdCode=1616

Dorsey, Julie, Rushmeier, Holly and Sillion, François, *Digital Modeling of Material Appearance*, Morgan Kaufmann Publishing, December 2007

- http://www.elsevierdirect.com/product.jsp?lid=0&iid=16&sid=0&isbn=9780122211812
- http://www.siggraph.org/s2005/main.php?f=conference&p=courses&s=24

A Field Guide to Digital Color
MAUREEN C. STONE
DIGITAL MODELING of MATERIAL APPEARANCE
web 3D CONSORTIUM
X3D

References 5


Kurt Cagle

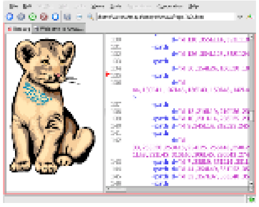
SVG Programming: The Graphical Web

Kurt Cagle, *SVG Programming: The Graphical Web*, Apress, 2002.

- http://www.apress.com/book/view/1590590198

Amaya open-source editor from W3C supports HTML, CSS, MathML and Scalable Vector Graphics (SVG).

- http://www.w3.org/Amaya

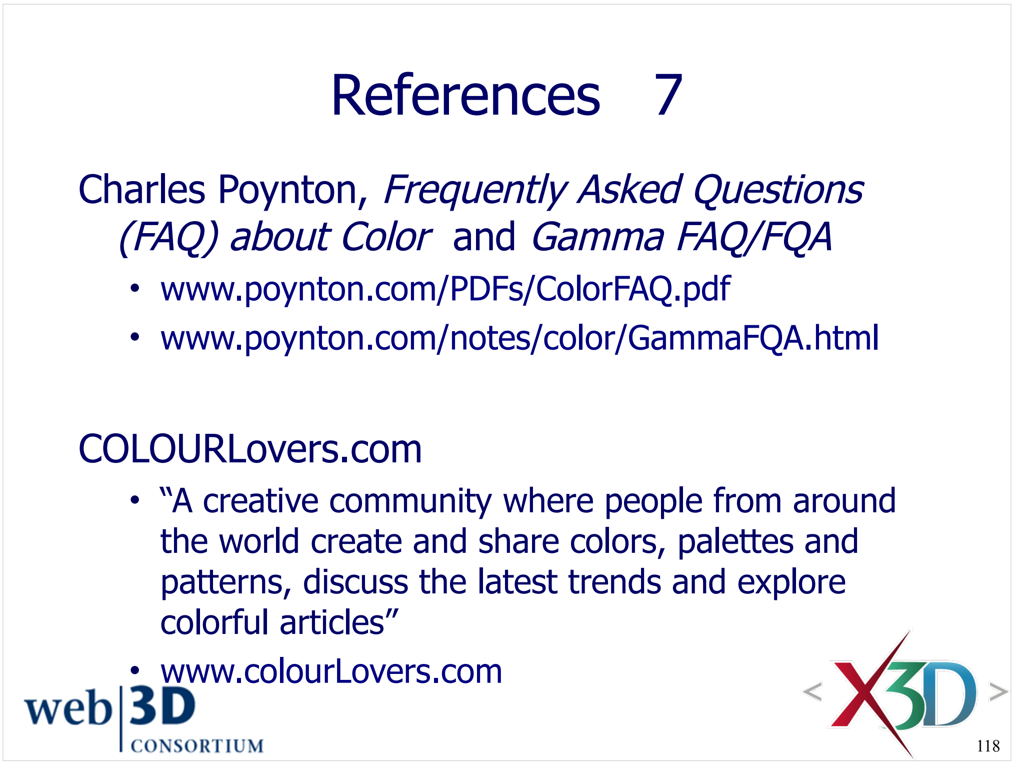


References 6

Pixel Perfect Text by David Frerichs

- Overcome poor pixelation of Text nodes by creating a texture image of the desired text, along with a matching Viewpoint at the right distance
- http://www.frerichs.net/vrml2/pp/pixel_perfect.html

Text geometry is covered in Chapter 2, Geometry Nodes: Primitives.



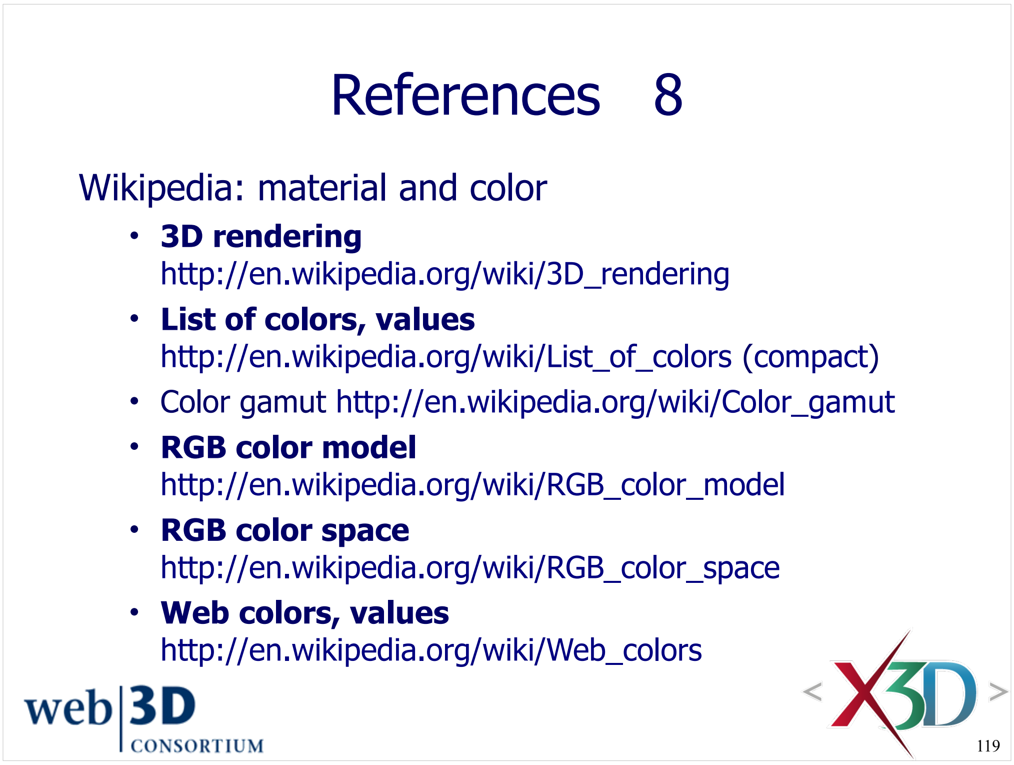
References 7

Charles Poynton, *Frequently Asked Questions (FAQ) about Color* and *Gamma FAQ/FQA*

- www.poynton.com/PDFs/ColorFAQ.pdf
- www.poynton.com/notes/color/GammaFQA.html

COLOURLovers.com

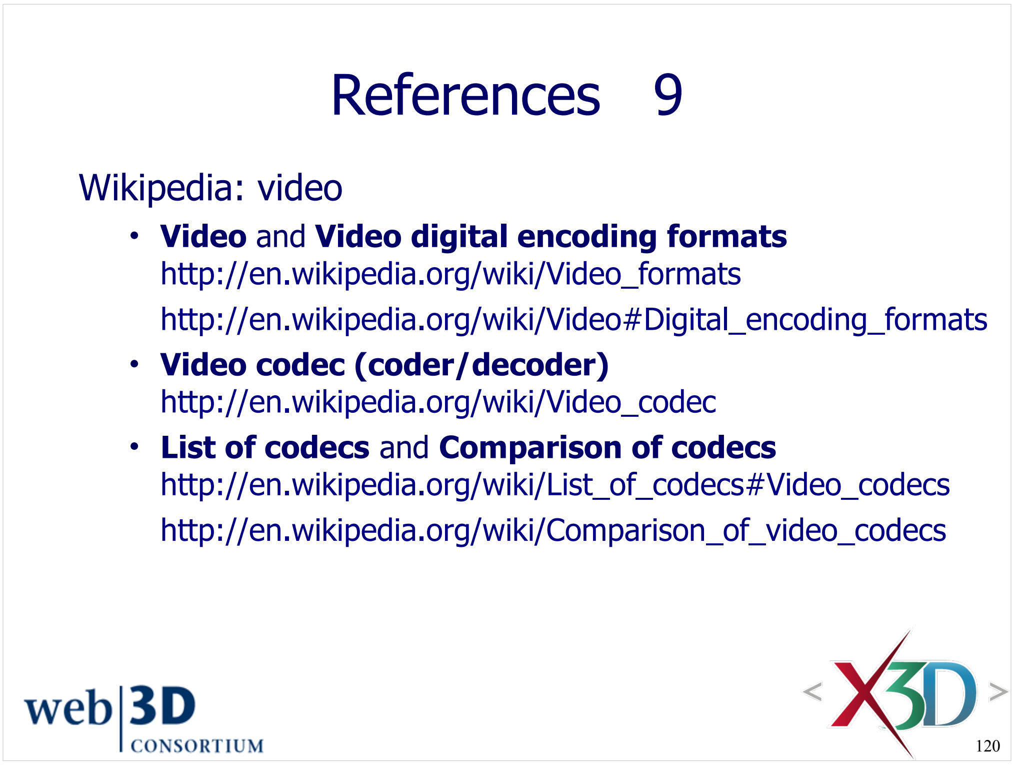
- "A creative community where people from around the world create and share colors, palettes and patterns, discuss the latest trends and explore colorful articles"
- www.colourLovers.com



References 8

Wikipedia: material and color

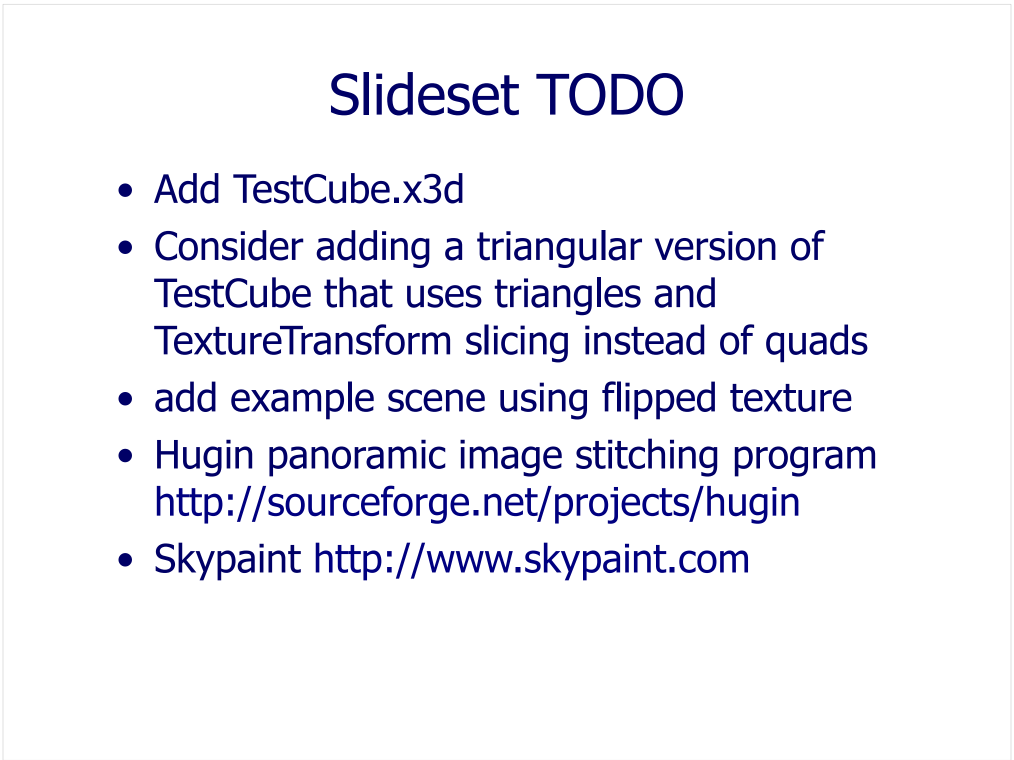
- **3D rendering** http://en.wikipedia.org/wiki/3D_rendering
- **List of colors, values** http://en.wikipedia.org/wiki/List_of_colors (compact)
- Color gamut http://en.wikipedia.org/wiki/Color_gamut
- **RGB color model** http://en.wikipedia.org/wiki/RGB_color_model
- **RGB color space** http://en.wikipedia.org/wiki/RGB_color_space
- **Web colors, values** http://en.wikipedia.org/wiki/Web_colors



References 9

Wikipedia: video

- **Video** and **Video digital encoding formats**
  http://en.wikipedia.org/wiki/Video_formats
  http://en.wikipedia.org/wiki/Video#Digital_encoding_formats
- **Video codec (coder/decoder)**
  http://en.wikipedia.org/wiki/Video_codec
- **List of codecs** and **Comparison of codecs**
  http://en.wikipedia.org/wiki/List_of_codecs#Video_codecs
  http://en.wikipedia.org/wiki/Comparison_of_video_codecs



Slideset TODO

- Add TestCube.x3d
- Consider adding a triangular version of TestCube that uses triangles and TextureTransform slicing instead of quads
- add example scene using flipped texture
- Hugin panoramic image stitching program http://sourceforge.net/projects/hugin
- Skypaint http://www.skypaint.com

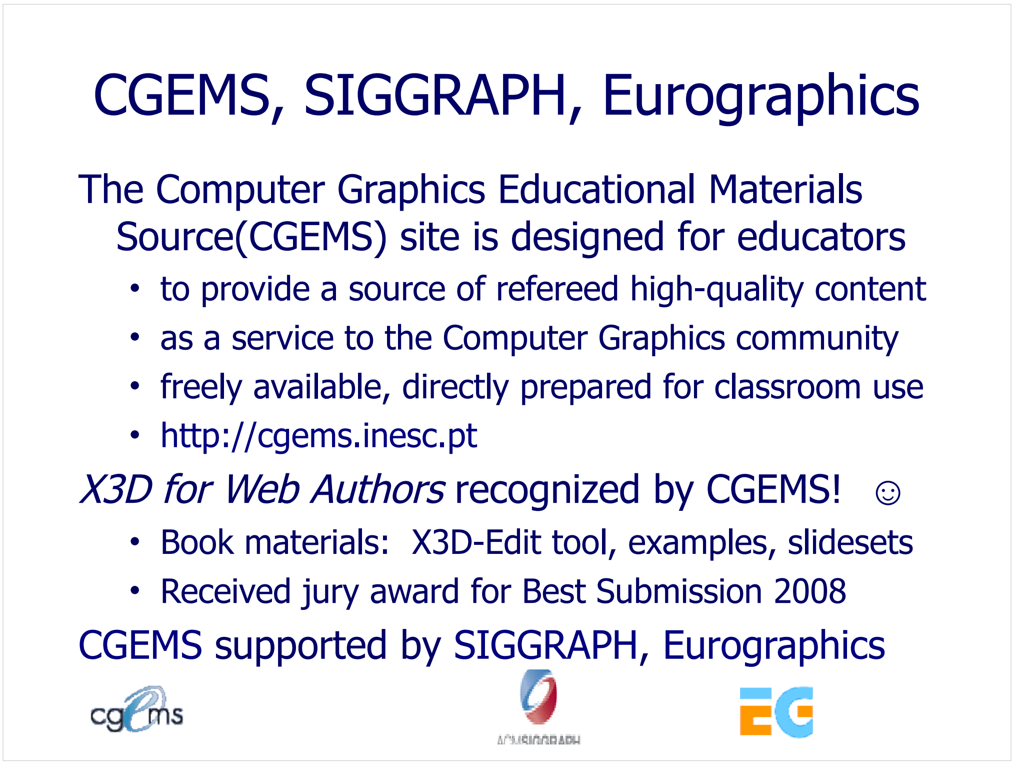
Contact

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:

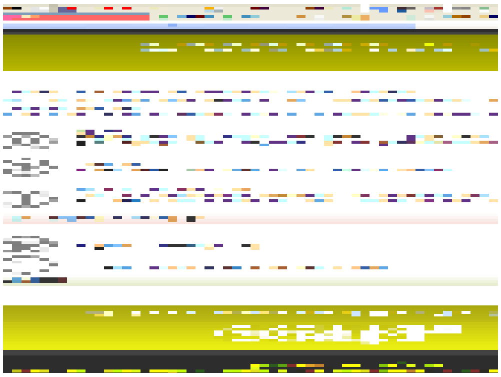
- http://cgems.inesc.pt/authors/ListModules.aspx

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).

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.

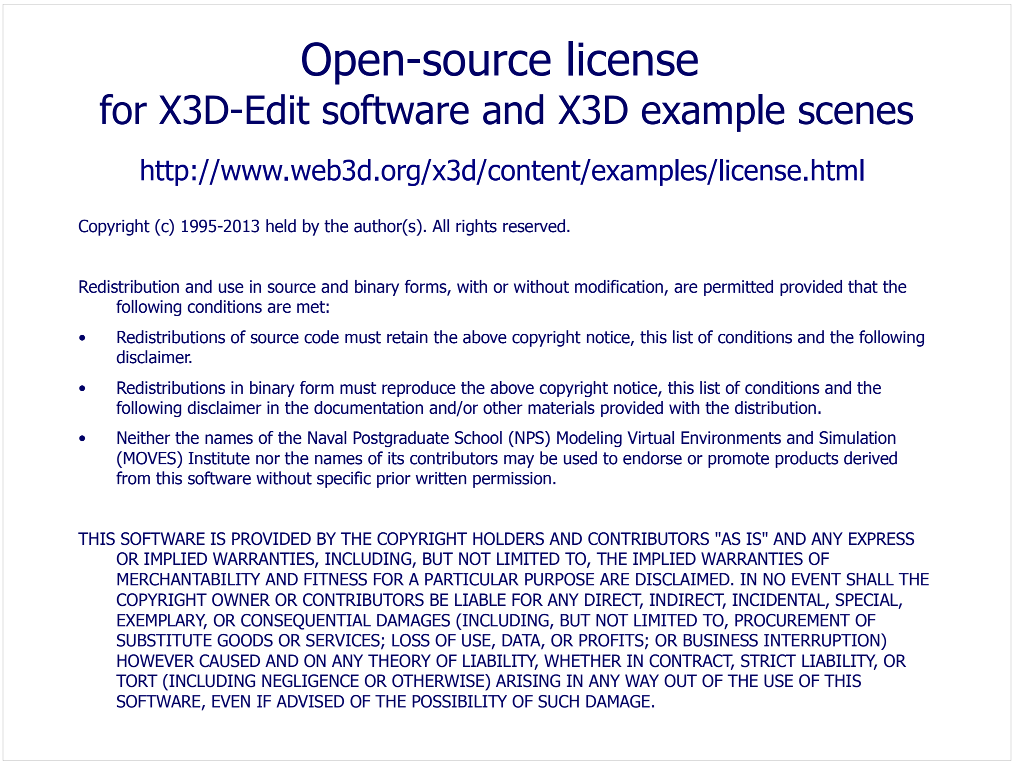
* Noncommercial. You may not use this work for commercial purposes.

* 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