# X3D Graphics for Web Authors

## Chapter 11

# Lighting and Environment Nodes

*Daylight encourages good behavior.*

Don Brutzman

# Contents

Chapter Overview

Lighting Concepts and Lighting Nodes

Environment Concepts and Environment Nodes

Chapter Summary and Suggested Exercises

References

# Chapter Overview

# Overview:  Lighting and Environment

Lighting concepts

- Illuminating scene geometry with virtual lights

Lighting nodes

- DirectionalLight
- NavigationInfo headlight
- PointLight, Spotlight

Environment concepts

- X3DBackgroundNode type and common fields

Environmental effects nodes

- Background, TextureBackground, Fog

# Lighting Concepts

# Illuminating objects in 3D scenes 1

So far we have studied the creation of 3D shapes and then positioning the camera to view them

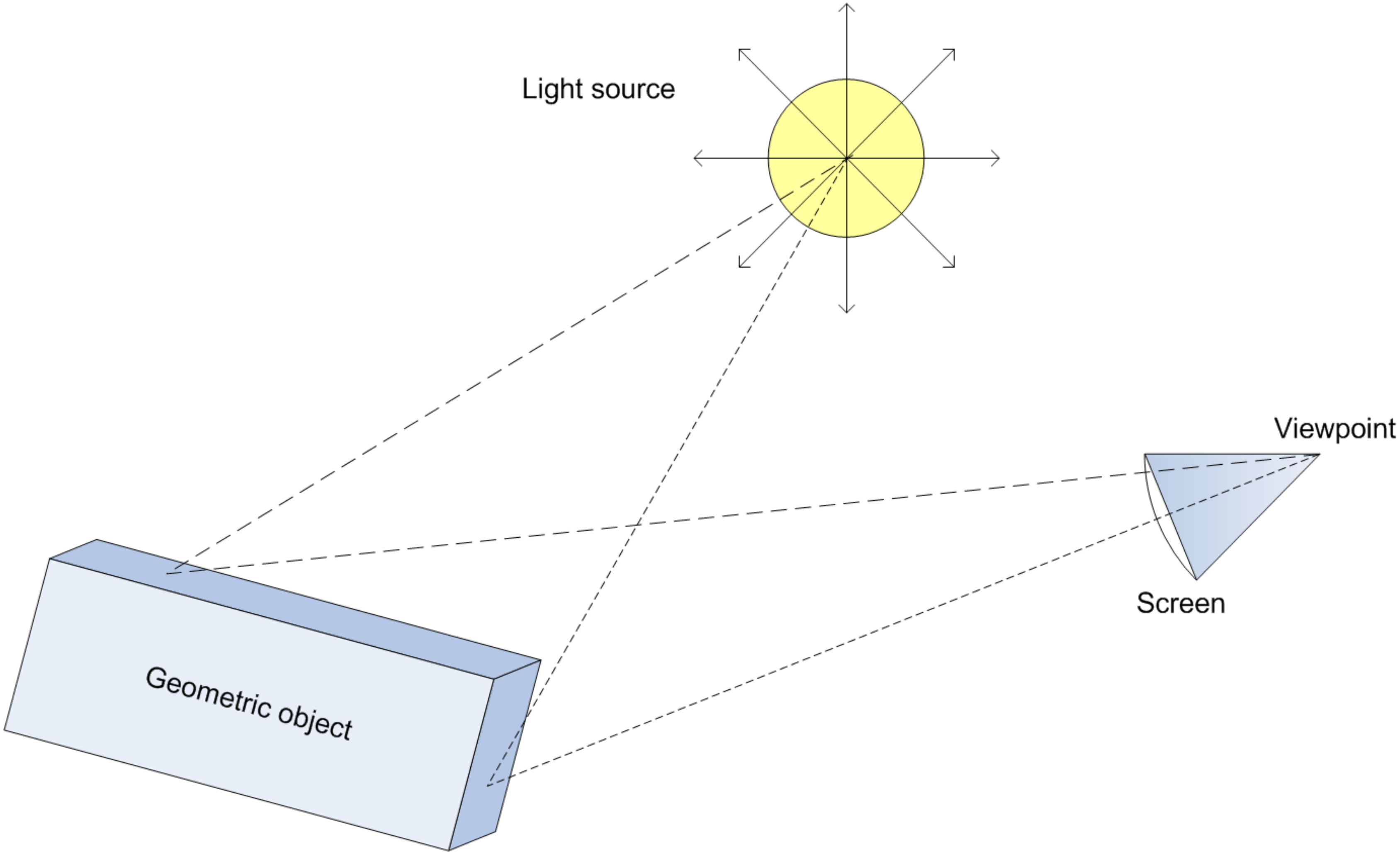
Third major component is needed for 3D scenes: virtual lights that emulate sources in real world

Such lighting occurs computationally – angles of incidence and reflection between lights and viewer are computed for each geometric shape

- Additional lighting factors are also considered

Pixel brightness reaching viewer thus depends on light sources, shape appearance, view angle

# Illuminating objects in 3D scenes   2



# Performance is paramount

Frame rate (measured in fps, frames per second) must be high for users to believe a 3D scene is interactive and responsive to their touch

- Human perception of smooth motion is 7-8 fps
- Keep frame rate above 10-15 fps in your scenes

Adding multiple lights in a scene can improve realism, if done properly

- Contributions of each can be added, pixel by pixel

However lights add significant computational cost, doubling/tripling/etc. rendering overhead

- Thus light management requires careful control

# Approximations necessary for interactive real-time performance

The mathematics of optics are quite complex

- Compare sizes of graphics and optics textbooks!

Ultra-high-fidelity computation of lighting effects is possible, and subject of ongoing research

Nevertheless the computational complexity of such effects are prohibitive for most computers

Thus a number of careful approximations made in lighting equations and material functions, allowing reasonable rendering to occur while supporting interactive real-time performance

# Other render techniques possible

A variety of higher-fidelity systems are available that use different techniques

- Ray tracing, volumetric rendering, light fields, etc.

Nevertheless most interactive 3D graphics systems use identical (or at least similar) lighting equations and rendering techniques

Following precomputation, advanced techniques can often export results for 3D rendering

Thus X3D is a good match for each approach when interactive export to Web is needed

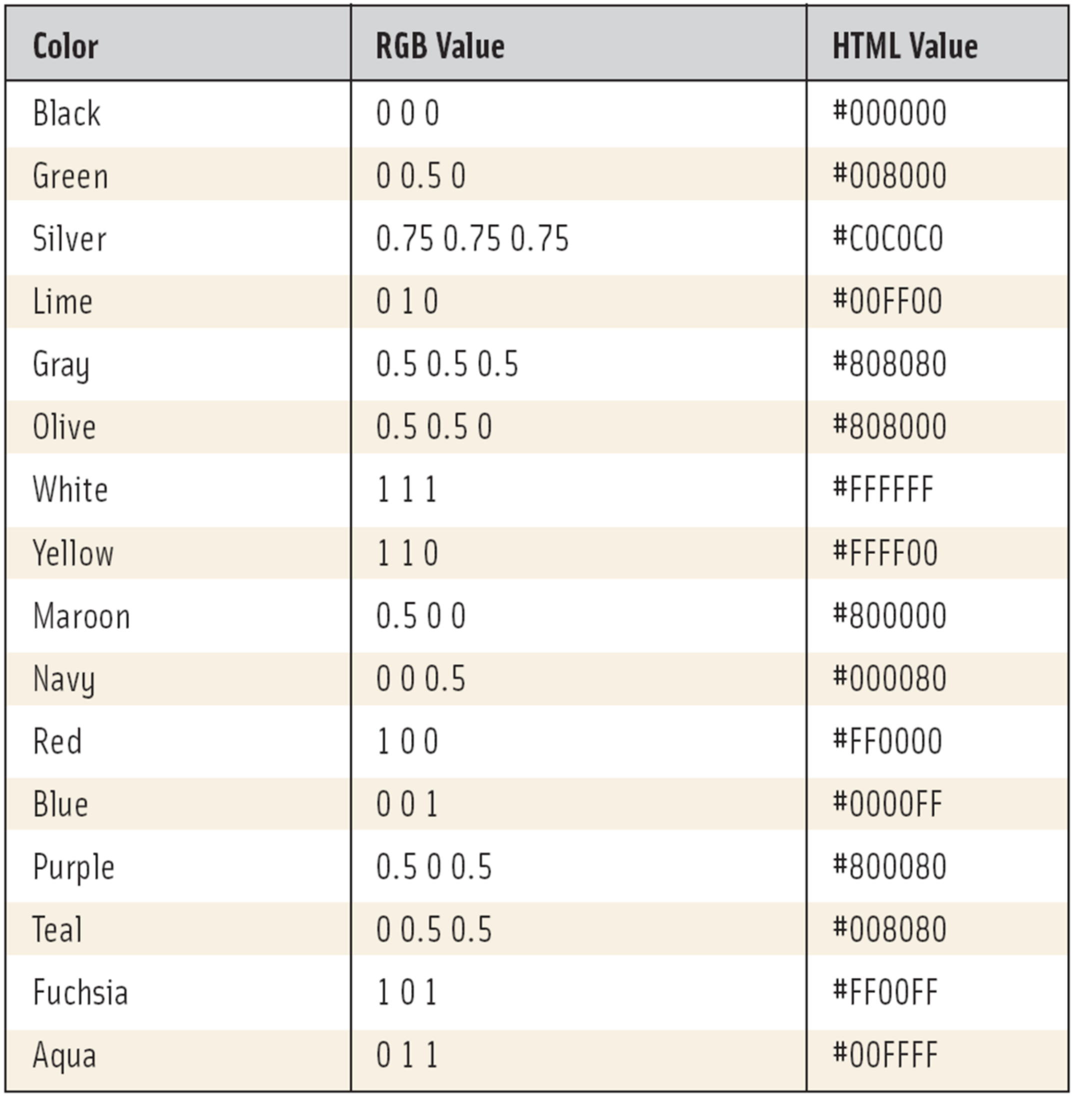
# Achieving a balance

- Authoring the right mix of lighting, appearance and viewing is both technical and artistic
  - This is why both types of activity are prevalent in the graphics community
- Making a scene "look good" sometimes requires technical shortcuts or workarounds that do not exactly match the original real-world effects
- Carefully applying these techniques to achieve an attractive, functional scene is the primary challenge facing all 3D graphics authors

# Common lighting fields 1

- *ambientIntensity* determines brightness of assumed reflected background ambient lighting, which occurs in most indoor and outdoor locations. This field also helps ensure that objects remain visible.
- *intensity* indicates the direct brightness of a light shining from the source location, and can be used to adjust relative strengths of multiple lights
- *color* specifies the red-green-blue (RGB) spectral components of the light source; adding to (or subtracting from) all three components affects the output by whitening (or darkening) the light source

# Common color values



| Color | RGB Value | HTML Value |
|---|---|---|
| Black | 0 0 0 | #000000 |
| Green | 0 0.5 0 | #008000 |
| Silver | 0.75 0.75 0.75 | #C0C0C0 |
| Lime | 0 1 0 | #00FF00 |
| Gray | 0.5 0.5 0.5 | #808080 |
| Olive | 0.5 0.5 0 | #808000 |
| White | 1 1 1 | #FFFFFF |
| Yellow | 1 1 0 | #FFFF00 |
| Maroon | 0.5 0 0 | #800000 |
| Navy | 0 0 0.5 | #000080 |
| Red | 1 0 0 | #FF0000 |
| Blue | 0 0 1 | #0000FF |
| Purple | 0.5 0 0.5 | #800080 |
| Teal | 0 0.5 0.5 | #008080 |
| Fuchsia | 1 0 1 | #FF00FF |
| Aqua | 0 1 1 | #00FFFF |

# Common lighting fields 2

- *global* indicates whether the scope of a light affects the entire 3D scene, or simply the local geometry shared within a common parent grouping node. This can be expensive in large scenes, do not set *global*='true' unless the effect is intentional

- *on* is the boolean field which turns a light on/off, allowing simple animation effects

# Lighting limitations 1

Light shines through geometry and illuminates objects that block each other

- Necessary limitation to support real-time rendering
- Can be surprising but usually not noticeable

No shadows

- Might be supported in a future X3D specification

Placing a light inside an object is problematic

- Only illuminates back faces, resulting in no light to externally visible faces
- Avoid, unless intentionally showing light location

# Lighting limitations 2

Maximum number of active lights: 8

- Can use more if turned off/on appropriately
- Matches limits of OpenGL, DirectX, GPU hardware
- Actually this is a high number for most applications

# Lighting Nodes and Examples

# DirectionalLight node

DirectionalLight provides uniform-intensity light as set of parallel rays sharing a single direction

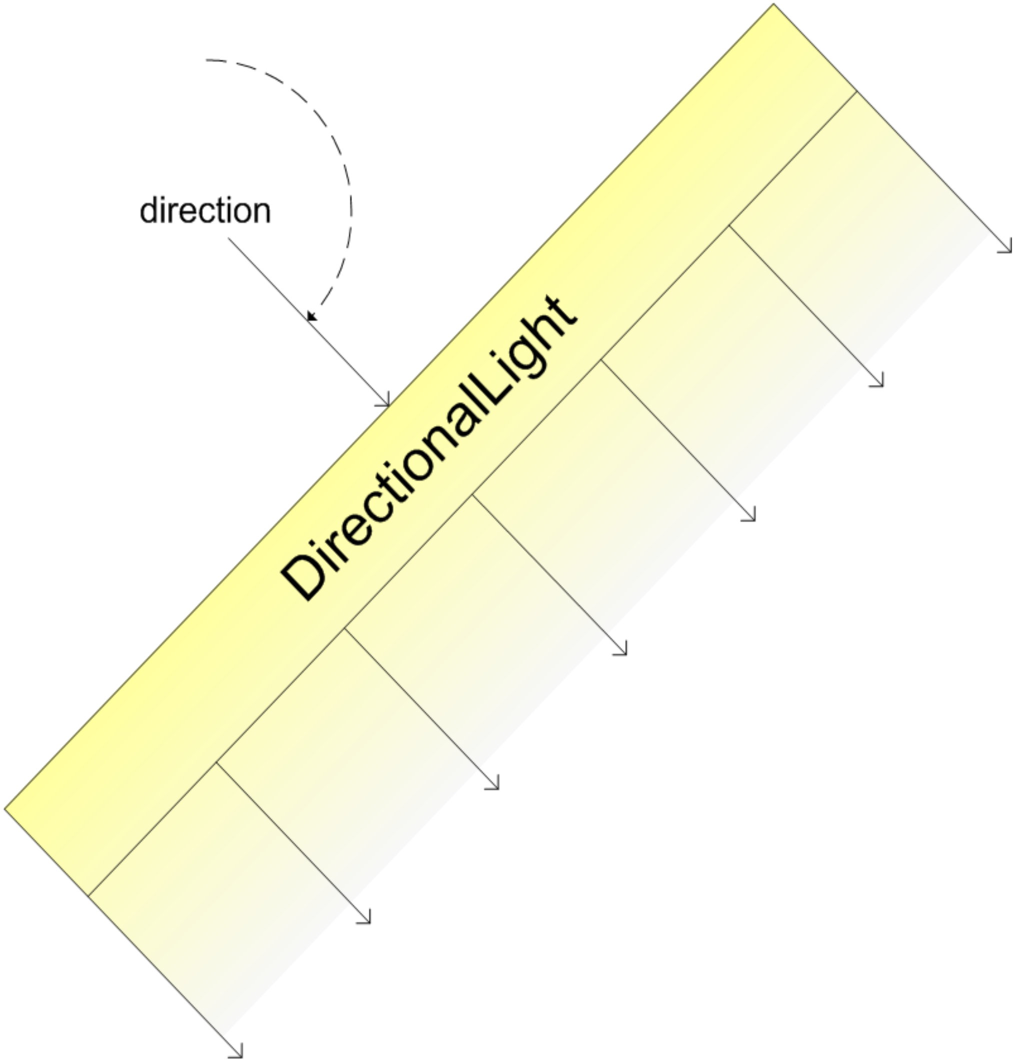
- Each surface facing the light gets some illumination
- Backface polygons, surfaces parallel to light rays remain unlit by direct *intensity*
- All surfaces lit by *ambientIntensity*, facing or not

Angle of incidence/reflection and appearance properties of lit shapes are primary factors on reflected color, brightness of lit objects

Emulates distant light source, such as the sun

- Intensity does not diminish with distance

# DirectionalLight geometry



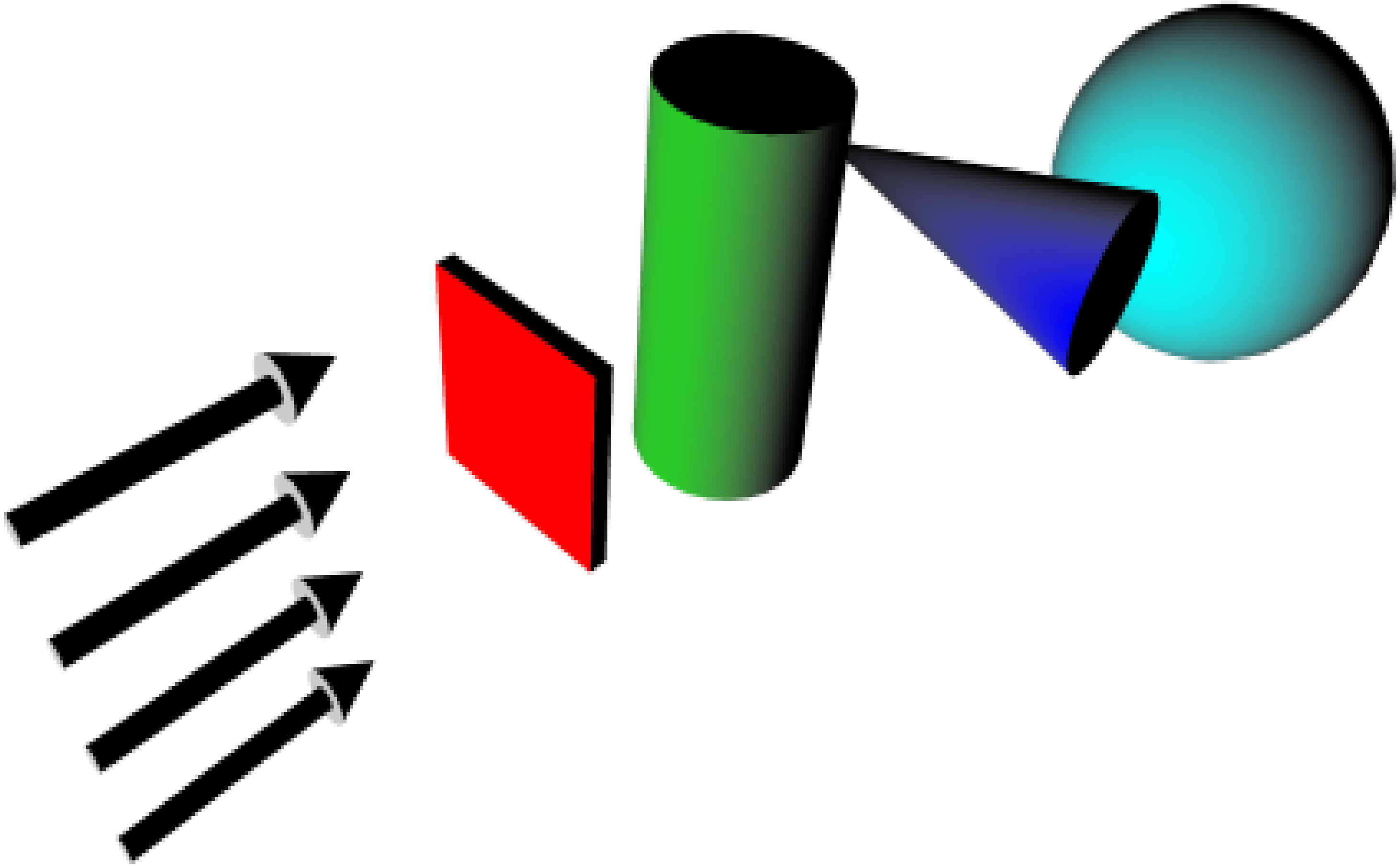
# DirectionalLight fields

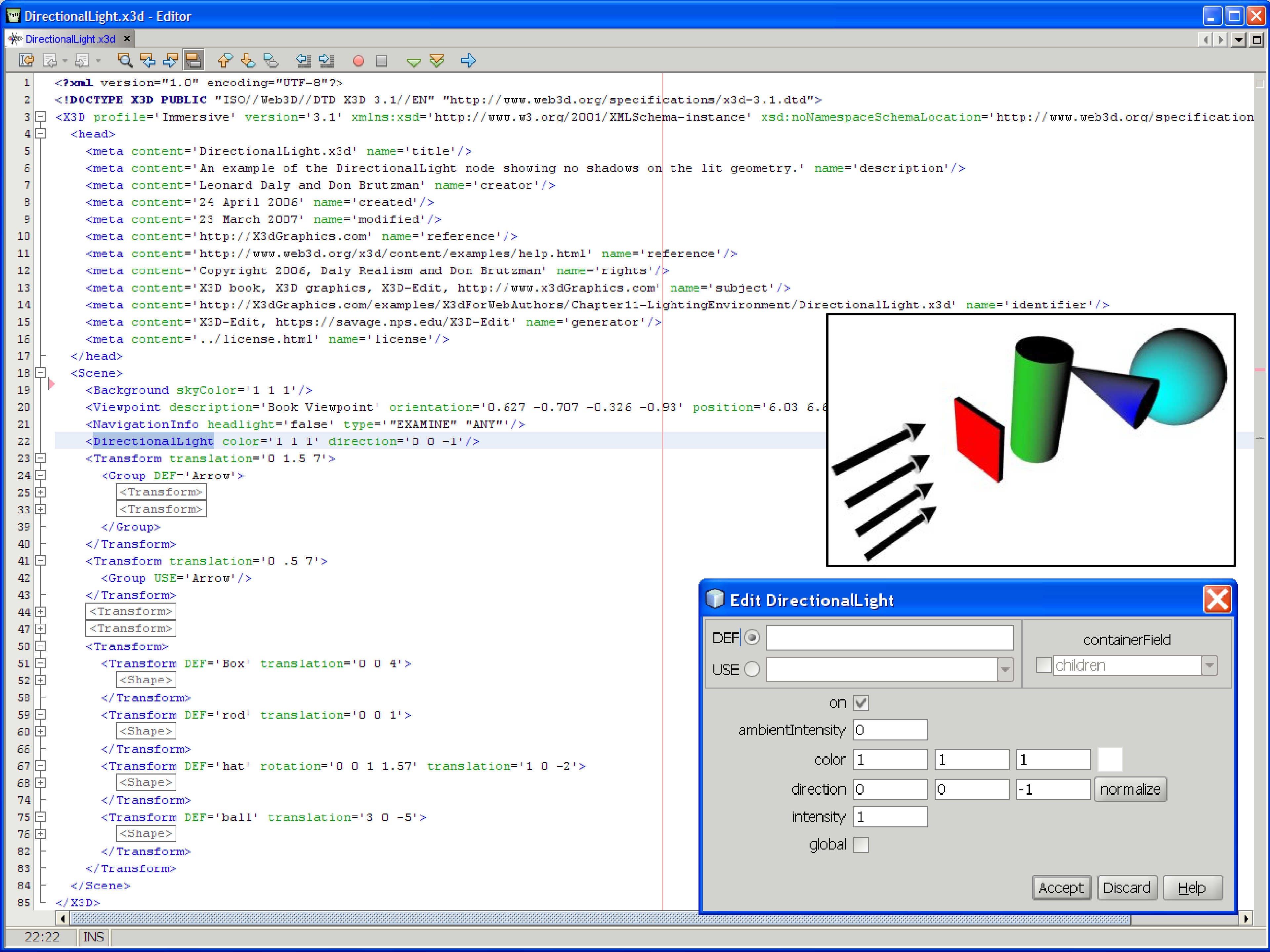
- *ambientIntensity, color, global, intensity, on* – same
- *direction* is x-y-z vector indicating direction of rays in local coordinate system, which in turn depends on rotations in parent Transform node hierarchy

## Hints

- Misdirected DirectionalLight nodes hard to debug, since light nodes themselves are not visible
- Substitute a PointLight node to get indication of where the light actually is located
- Can rotate DirectionalLight via parent Transform, also semitransparent Cone for direction, range
- Position irrelevant since constant intensity, direction

# No shadows, constant intensity



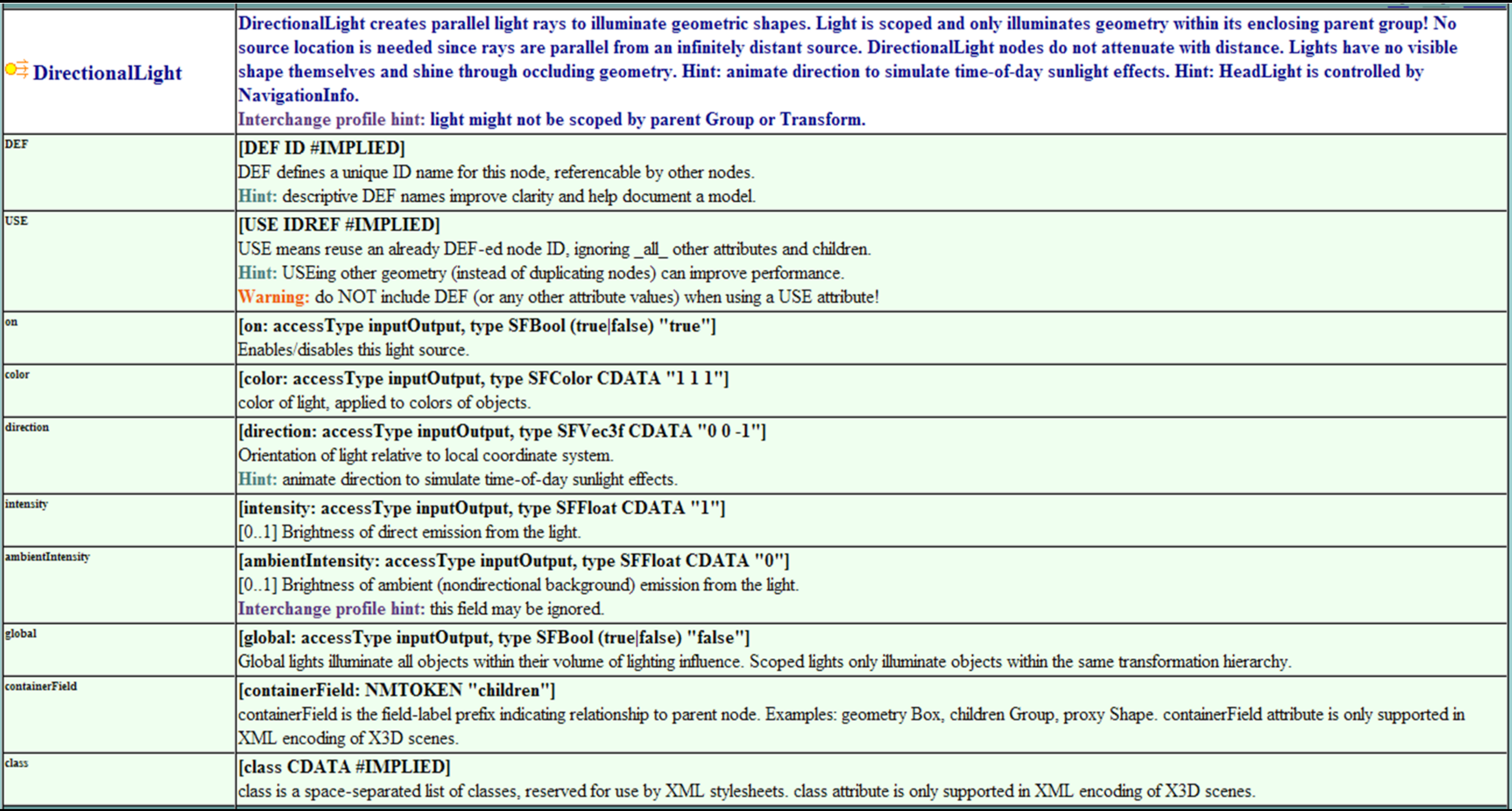


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specification
  <head>
    <meta content='DirectionalLight.x3d' name='title'/>
    <meta content='An example of the DirectionalLight node showing no shadows on the lit geometry.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='24 April 2006' name='created'/>
    <meta content='23 March 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 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/Chapter11-LightingEnvironment/DirectionalLight.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Book Viewpoint' orientation='0.627 -0.707 -0.326 -0.93' position='6.03 6.8
    <NavigationInfo headlight='false' type='"EXAMINE" "ANY"'/>
    <DirectionalLight color='1 1 1' direction='0 0 -1'/>
    <Transform translation='0 1.5 7'>
      <Group DEF='Arrow'>
        <Transform>
        <Transform>
      </Group>
    </Transform>
    <Transform translation='0 .5 7'>
      <Group USE='Arrow'/>
    </Transform>
    <Transform>
    <Transform>
    <Transform>
      <Transform DEF='Box' translation='0 0 4'>
        <Shape>
      </Transform>
      <Transform DEF='rod' translation='0 0 1'>
        <Shape>
      </Transform>
      <Transform DEF='hat' rotation='0 0 1 1.57' translation='1 0 -2'>
        <Shape>
      </Transform>
      <Transform DEF='ball' translation='3 0 -5'>
        <Shape>
      </Transform>
    </Transform>
  </Scene>
</X3D>
```

Edit DirectionalLight

DEF ●
USE ○
containerField □ children

on ☑
ambientIntensity 0
color 1 1 1
direction 0 0 -1
intensity 1
global □



| | |
|---|---|
| DirectionalLight | DirectionalLight creates parallel light rays to illuminate geometric shapes. Light is scoped and only illuminates geometry within its enclosing parent group! No source location is needed since rays are parallel from an infinitely distant source. DirectionalLight nodes do not attenuate with distance. Lights have no visible shape themselves and shine through occluding geometry. Hint: animate direction to simulate time-of-day sunlight effects. Hint: HeadLight is controlled by NavigationInfo.<br>Interchange profile hint: light might not be scoped by parent Group or Transform. |
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>**Hint:** descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>**Hint:** USEing other geometry (instead of duplicating nodes) can improve performance.<br>**Warning:** do NOT include DEF (or any other attribute values) when using a USE attribute! |
| on | **[on: accessType inputOutput, type SFBool (true\|false) "true"]**<br>Enables/disables this light source. |
| color | **[color: accessType inputOutput, type SFColor CDATA "1 1 1"]**<br>color of light, applied to colors of objects. |
| direction | **[direction: accessType inputOutput, type SFVec3f CDATA "0 0 -1"]**<br>Orientation of light relative to local coordinate system.<br>**Hint:** animate direction to simulate time-of-day sunlight effects. |
| intensity | **[intensity: accessType inputOutput, type SFFloat CDATA "1"]**<br>[0..1] Brightness of direct emission from the light. |
| ambientIntensity | **[ambientIntensity: accessType inputOutput, type SFFloat CDATA "0"]**<br>[0..1] Brightness of ambient (nondirectional background) emission from the light.<br>**Interchange profile hint:** this field may be ignored. |
| global | **[global: accessType inputOutput, type SFBool (true\|false) "false"]**<br>Global lights illuminate all objects within their volume of lighting influence. Scoped lights only illuminate objects within the same transformation hierarchy. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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. |

# NavigationInfo headlight

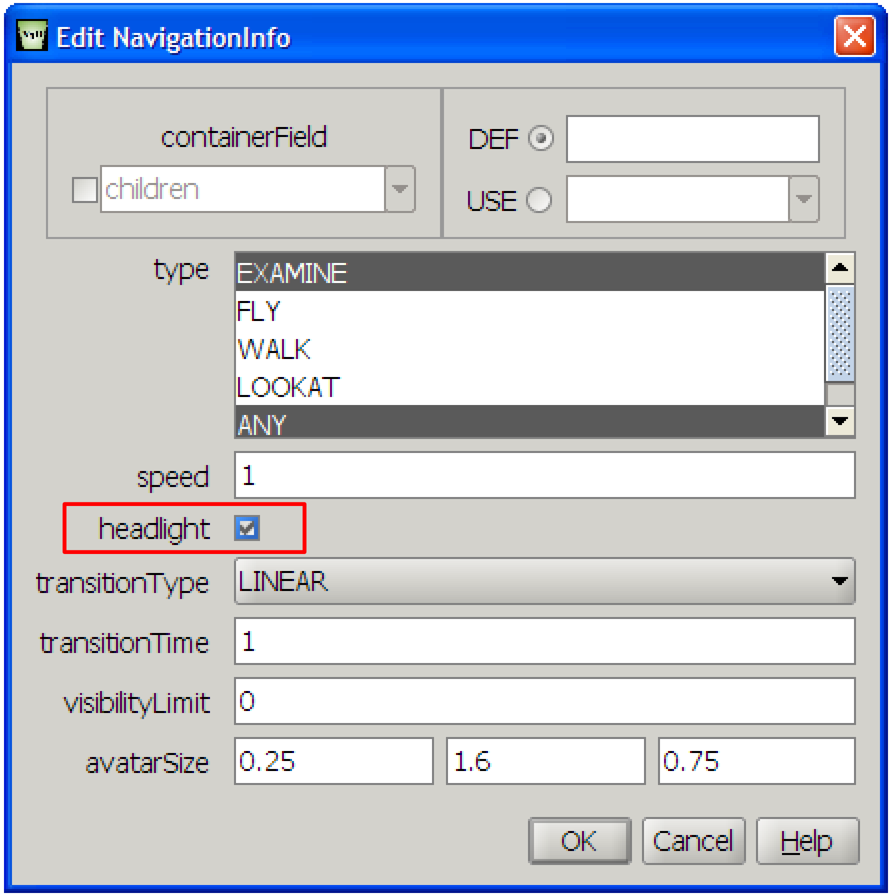
The NavigationInfo node's *headlight* field enables a special DirectionalLight that follows user's current view location, pointing straight ahead

- Enables a light source that is always pointing in direction of user's view, ensuring objects are lit
- *headlight*='true' by default, color is always white

*headlight* matches the following characteristics, pointed out from center of current active view:

```
<DirectionalLight color='1 1 1'
  intensity='1' ambientIntensity='1'
  direction='0 0 -1'/>
```

# NavigationInfo headlight, X3D-Edit



Edit NavigationInfo

containerField

□ children

DEF ●

USE ○

type: EXAMINE, FLY, WALK, LOOKAT, ANY

speed: 1

headlight ☑

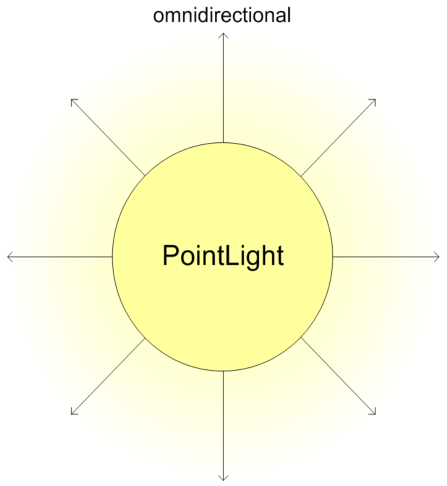
transitionType: LINEAR

transitionTime: 1

visibilityLimit: 0

avatarSize: 0.25 1.6 0.75

# PointLight node



PointLight emulates single light source that radiates equally in all directions

- Analogy: single small light

PointLight intensity varies with distance to object

- Longer travel means less light arrives from source

Rotation is irrelevant since light is omnidirectional

Shared traits with other lights:

- No shadows, light not blocked by other geometry
- No visible representation in space of light itself

# PointLight fields

- *ambientIntensity, color, global, intensity, on* – same
- *attenuation* array holds constant, linear, quadratic coefficients for distance-attenuation factor

  $\text{factor} = 1 / \max(\text{atten}[0] + \text{atten}[1] \cdot r + \text{atten}[2] \cdot r^2, 1)$

  Note that non-constant (linear and quadratic) coefficients are computationally expensive per pixel
- *location* is SFVec3f position indicating origin of rays relative to local coordinate system, which in turn depends on translations and rotations in parent Transform node hierarchy
- *radius* gives maximum effective range of light rays from source *location*, must be nonnegative

# PointLight considerations

Because light intensity falls off with distance, brightness can vary across a single object

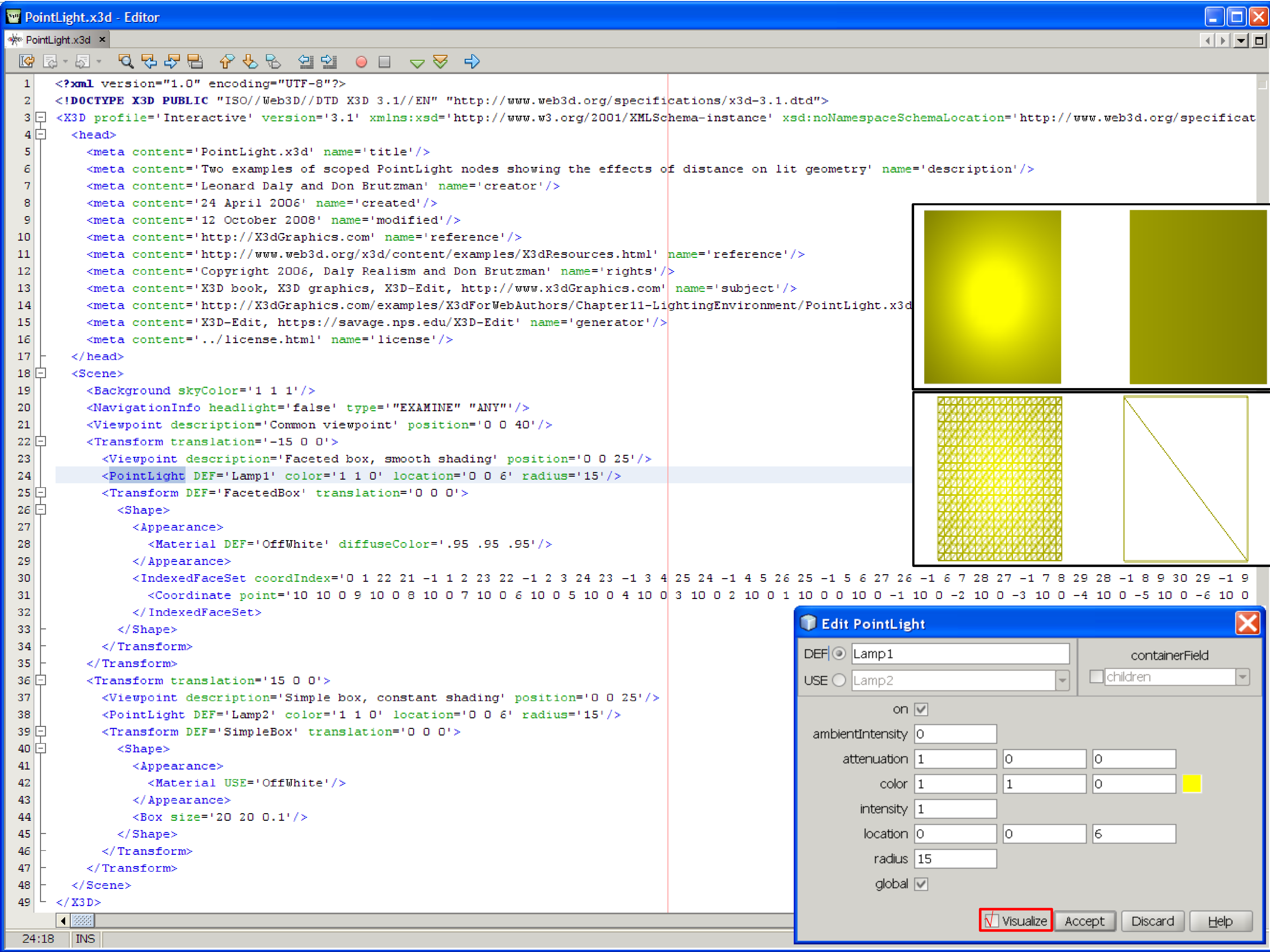
- Computed on a polygon-per-polygon basis for computational efficiency

This effect is not visible if a large polygon is used

- Since entire polygon receives consistent shading

Thus special tools for detailed tesselation of simple shapes might be needed to achieve special lighting effects

- Careful choice of viewing angle usually sufficient



```
<?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='Interactive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specificat
  <head>
    <meta content='PointLight.x3d' name='title'/>
    <meta content='Two examples of scoped PointLight nodes showing the effects of distance on lit geometry' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='24 April 2006' name='created'/>
    <meta content='12 October 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 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/Chapter11-LightingEnvironment/PointLight.x3d
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo headlight='false' type='"EXAMINE" "ANY"'/>
    <Viewpoint description='Common viewpoint' position='0 0 40'/>
    <Transform translation='-15 0 0'>
      <Viewpoint description='Faceted box, smooth shading' position='0 0 25'/>
      <PointLight DEF='Lamp1' color='1 1 0' location='0 0 6' radius='15'/>
      <Transform DEF='FacetedBox' translation='0 0 0'>
        <Shape>
          <Appearance>
            <Material DEF='OffWhite' diffuseColor='.95 .95 .95'/>
          </Appearance>
          <IndexedFaceSet coordIndex='0 1 22 21 -1 1 2 23 22 -1 2 3 24 23 -1 3 4 25 24 -1 4 5 26 25 -1 5 6 27 26 -1 6 7 28 27 -1 7 8 29 28 -1 8 9 30 29 -1 9
            <Coordinate point='10 10 0 9 10 0 8 10 0 7 10 0 6 10 0 5 10 0 4 10 0 3 10 0 2 10 0 1 10 0 0 10 0 -1 10 0 -2 10 0 -3 10 0 -4 10 0 -5 10 0 -6 10 0
          </IndexedFaceSet>
        </Shape>
      </Transform>
    </Transform>
    <Transform translation='15 0 0'>
      <Viewpoint description='Simple box, constant shading' position='0 0 25'/>
      <PointLight DEF='Lamp2' color='1 1 0' location='0 0 6' radius='15'/>
      <Transform DEF='SimpleBox' translation='0 0 0'>
        <Shape>
          <Appearance>
            <Material USE='OffWhite'/>
          </Appearance>
          <Box size='20 20 0.1'/>
        </Shape>
      </Transform>
    </Transform>
  </Scene>
</X3D>
```

**Edit PointLight**

DEF ● Lamp1

USE ○ Lamp2

containerField □ children

on ☑

ambientIntensity 0

attenuation 1 0 0

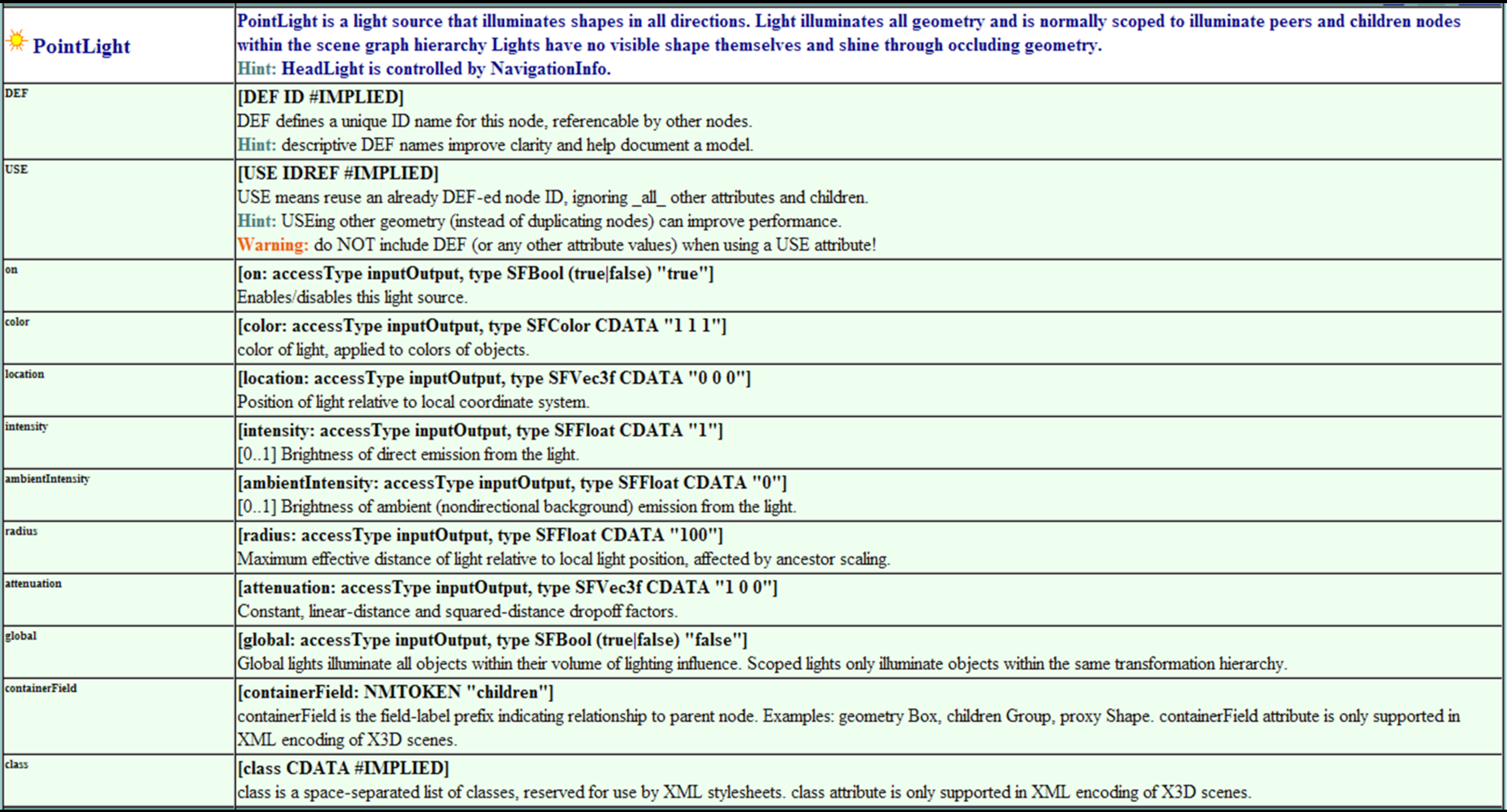
color 1 1 0 🟨

intensity 1

location 0 0 6

radius 15

global ☑



| | |
|---|---|
| **PointLight** | **PointLight is a light source that illuminates shapes in all directions. Light illuminates all geometry and is normally scoped to illuminate peers and children nodes within the scene graph hierarchy Lights have no visible shape themselves and shine through occluding geometry.**<br>**Hint: HeadLight is controlled by NavigationInfo.** |
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>Hint: descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.<br>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute! |
| on | **[on: accessType inputOutput, type SFBool (true\|false) "true"]**<br>Enables/disables this light source. |
| color | **[color: accessType inputOutput, type SFColor CDATA "1 1 1"]**<br>color of light, applied to colors of objects. |
| location | **[location: accessType inputOutput, type SFVec3f CDATA "0 0 0"]**<br>Position of light relative to local coordinate system. |
| intensity | **[intensity: accessType inputOutput, type SFFloat CDATA "1"]**<br>[0..1] Brightness of direct emission from the light. |
| ambientIntensity | **[ambientIntensity: accessType inputOutput, type SFFloat CDATA "0"]**<br>[0..1] Brightness of ambient (nondirectional background) emission from the light. |
| radius | **[radius: accessType inputOutput, type SFFloat CDATA "100"]**<br>Maximum effective distance of light relative to local light position, affected by ancestor scaling. |
| attenuation | **[attenuation: accessType inputOutput, type SFVec3f CDATA "1 0 0"]**<br>Constant, linear-distance and squared-distance dropoff factors. |
| global | **[global: accessType inputOutput, type SFBool (true\|false) "false"]**<br>Global lights illuminate all objects within their volume of lighting influence. Scoped lights only illuminate objects within the same transformation hierarchy. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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. |

# SpotLight node

SpotLight illuminates shapes within conical beam

- Intensity decreases with distance from source

Spotlight can be considered similar to PointLight with exception that computed light is constrained to be within conical beams, defined by solid angles

- Center portion of cone (indicated by *beamWidth*) receives full intensity
- Outer portion of light cone has linear drop off of reduced intensity (indicated by *cutOffAngle*)

# SpotLight fields 1

- *ambientIntensity, color, global, intensity, on* – same
- *attenuation* array holds constant, linear, quadratic coefficients for distance-attenuation factor

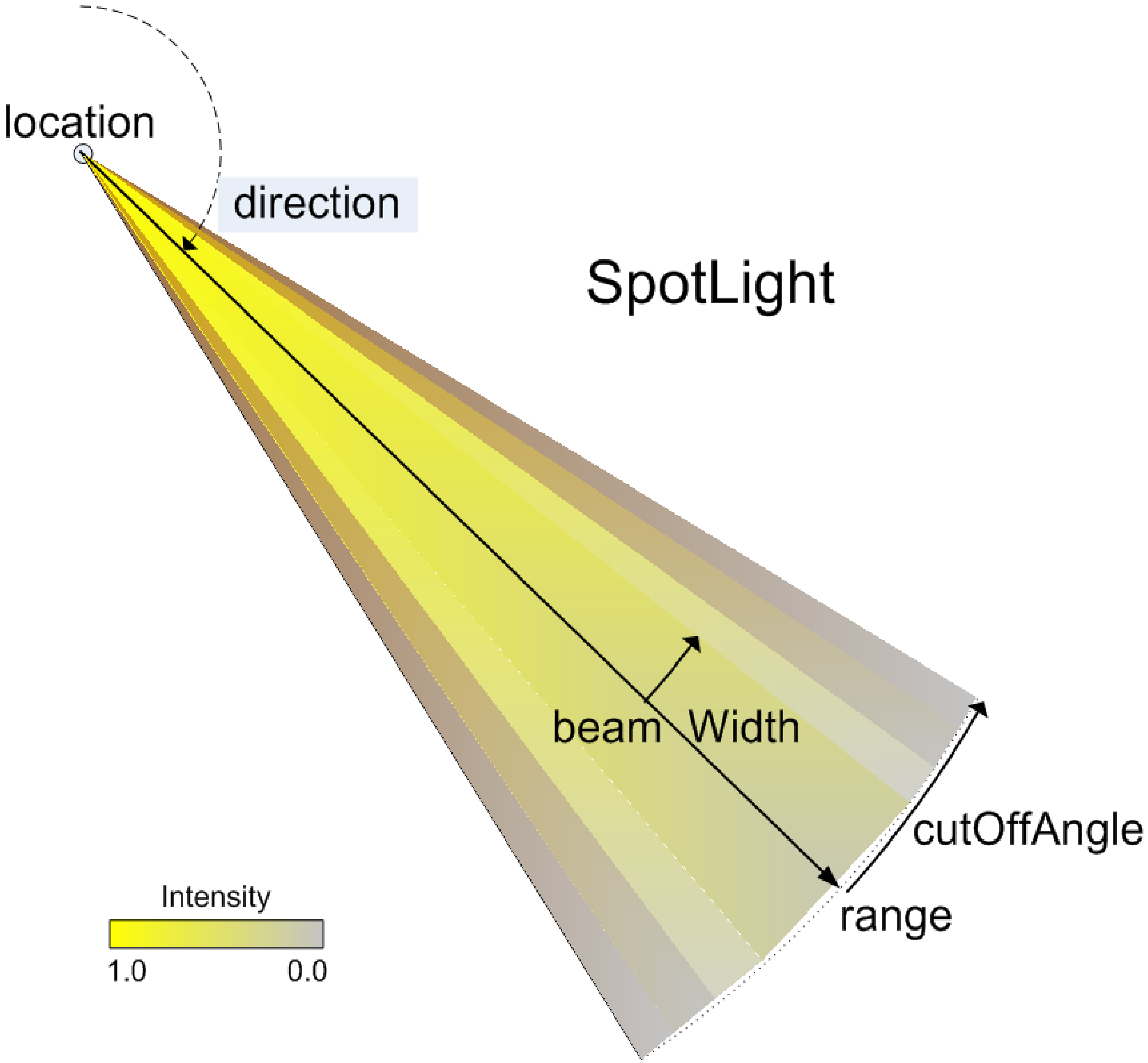
  $\text{factor} = 1 \,/\, \max\,(\text{atten}[0] + \text{atten}[1] \cdot r + \text{atten}[2] \cdot r^2,\ 1)$

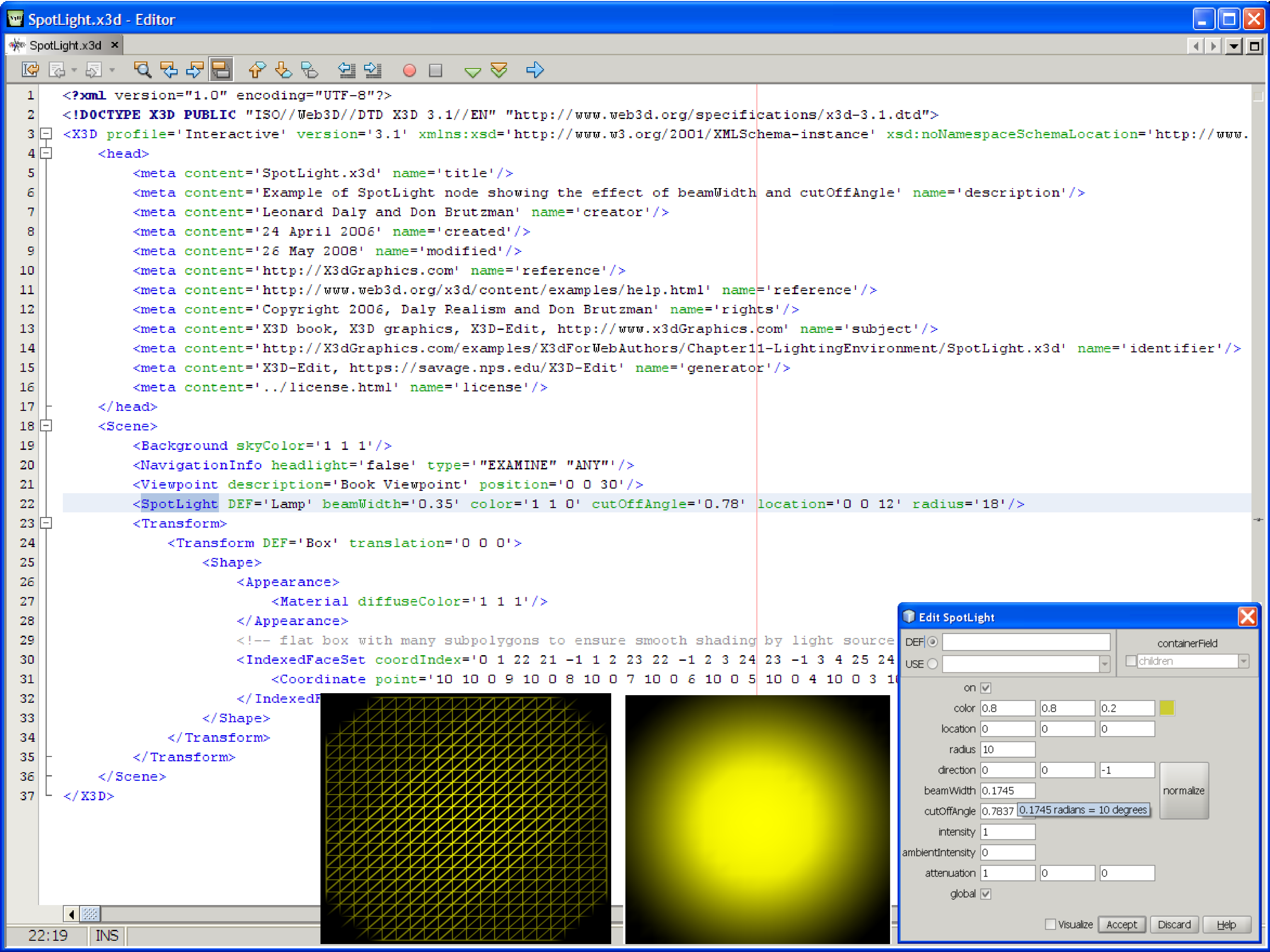
  Note that non-constant (linear and quadratic) coefficients are computationally expensive per pixel
- *location* is SFVec3f position indicating origin of rays relative to local coordinate system, which in turn depends on translations and rotations in parent Transform node hierarchy
- *radius* gives maximum effective range of light rays from source *location*, must be nonnegative

# SpotLight fields 2

- *beamWidth* specifies the half angle (in radians) about the SpotLight direction that corresponds to maximum uniform intensity
- *cutOffAngle* specifies the half angle (in radians) about the SpotLight direction that corresponds to outer bound of SpotLight node's effect. Outside of this solid angle, no light is provided.
- Constraint: 0 ≤ *beamWidth* ≤ *cutOffAngle* ≤ pi/2
- *direction* is x-y-z vector indicating direction of rays in local coordinate system, which in turn depends on rotations in parent Transform node hierarchy

# SpotLight fields diagram





```
<?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='Interactive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.
    <head>
        <meta content='SpotLight.x3d' name='title'/>
        <meta content='Example of SpotLight node showing the effect of beamWidth and cutOffAngle' name='description'/>
        <meta content='Leonard Daly and Don Brutzman' name='creator'/>
        <meta content='24 April 2006' name='created'/>
        <meta content='26 May 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 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/Chapter11-LightingEnvironment/SpotLight.x3d' name='identifier'/>
        <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
        <meta content='../license.html' name='license'/>
    </head>
    <Scene>
        <Background skyColor='1 1 1'/>
        <NavigationInfo headlight='false' type='"EXAMINE" "ANY"'/>
        <Viewpoint description='Book Viewpoint' position='0 0 30'/>
        <SpotLight DEF='Lamp' beamWidth='0.35' color='1 1 0' cutOffAngle='0.78' location='0 0 12' radius='18'/>
        <Transform>
            <Transform DEF='Box' translation='0 0 0'>
                <Shape>
                    <Appearance>
                        <Material diffuseColor='1 1 1'/>
                    </Appearance>
                    <!-- flat box with many subpolygons to ensure smooth shading by light source
                    <IndexedFaceSet coordIndex='0 1 22 21 -1 1 2 23 22 -1 2 3 24 23 -1 3 4 25 24
                        <Coordinate point='10 10 0 9 10 0 8 10 0 7 10 0 6 10 0 5 10 0 4 10 0 3 1
                    </IndexedF
                </Shape>
            </Transform>
        </Transform>
    </Scene>
</X3D>
```

**Edit SpotLight**

DEF ● 
USE ○

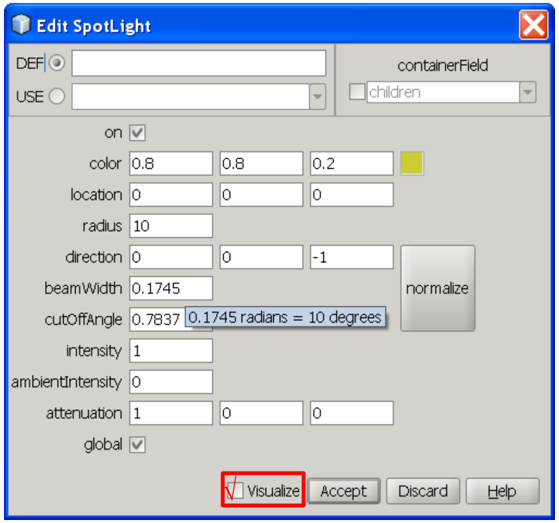
containerField
□ children

on ☑
color 0.8 0.8 0.2 🟨
location 0 0 0
radius 10
direction 0 0 -1
beamWidth 0.1745
cutOffAngle 0.7837 0.1745 radians = 10 degrees
intensity 1
ambientIntensity 0
attenuation 1 0 0
global ☑

# SpotLight visualization

X3D-Edit has an authoring assist to show outlines for SpotLight fields

- *cutOffAngle* shown by wireframe
- *beamWidth* shown by transparent Cone
- Example:  SpotLightVisualization.x3d



Edit SpotLight

DEF ●

USE ○

containerField □ children

on ☑

color 0.8 0.8 0.2

location 0 0 0

radius 10

direction 0 0 -1

beamWidth 0.1745

cutOffAngle 0.7837 0.1745 radians = 10 degrees

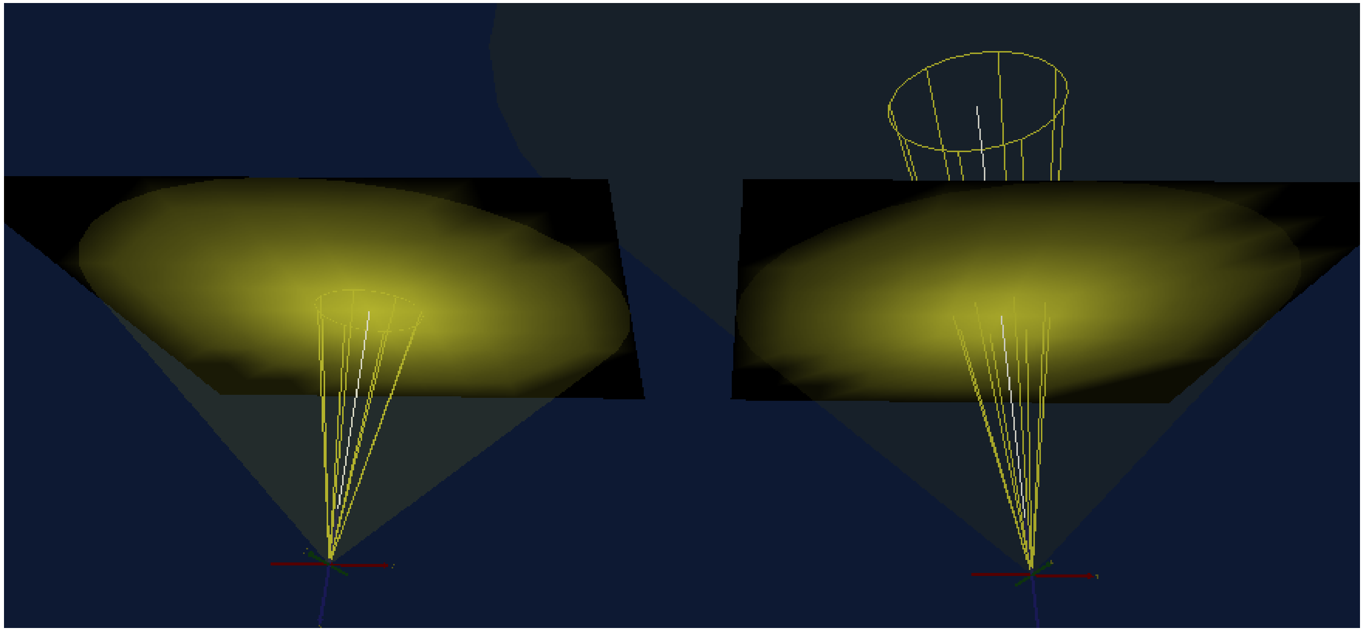
intensity 1

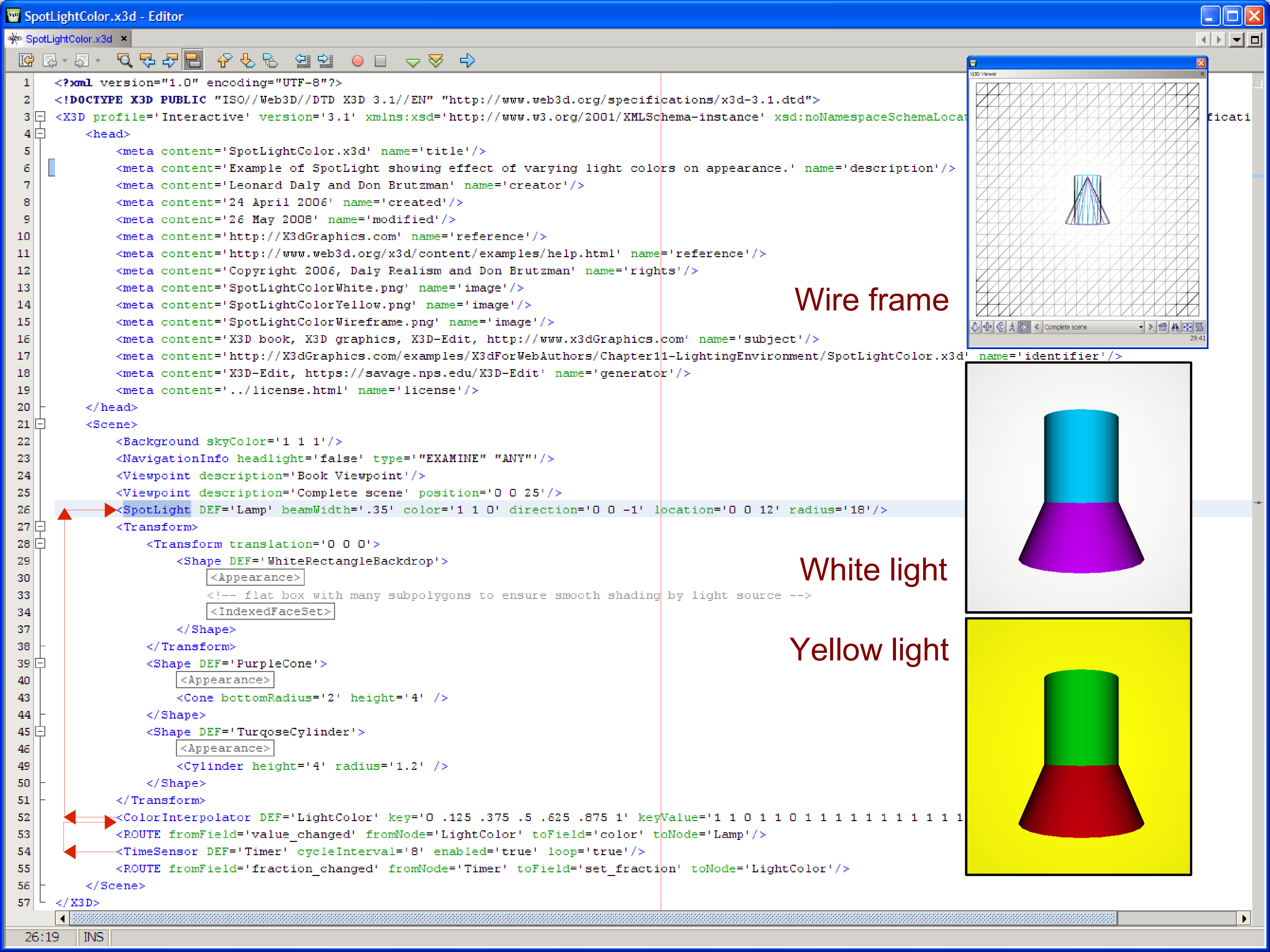
ambientIntensity 0

attenuation 1 0 0

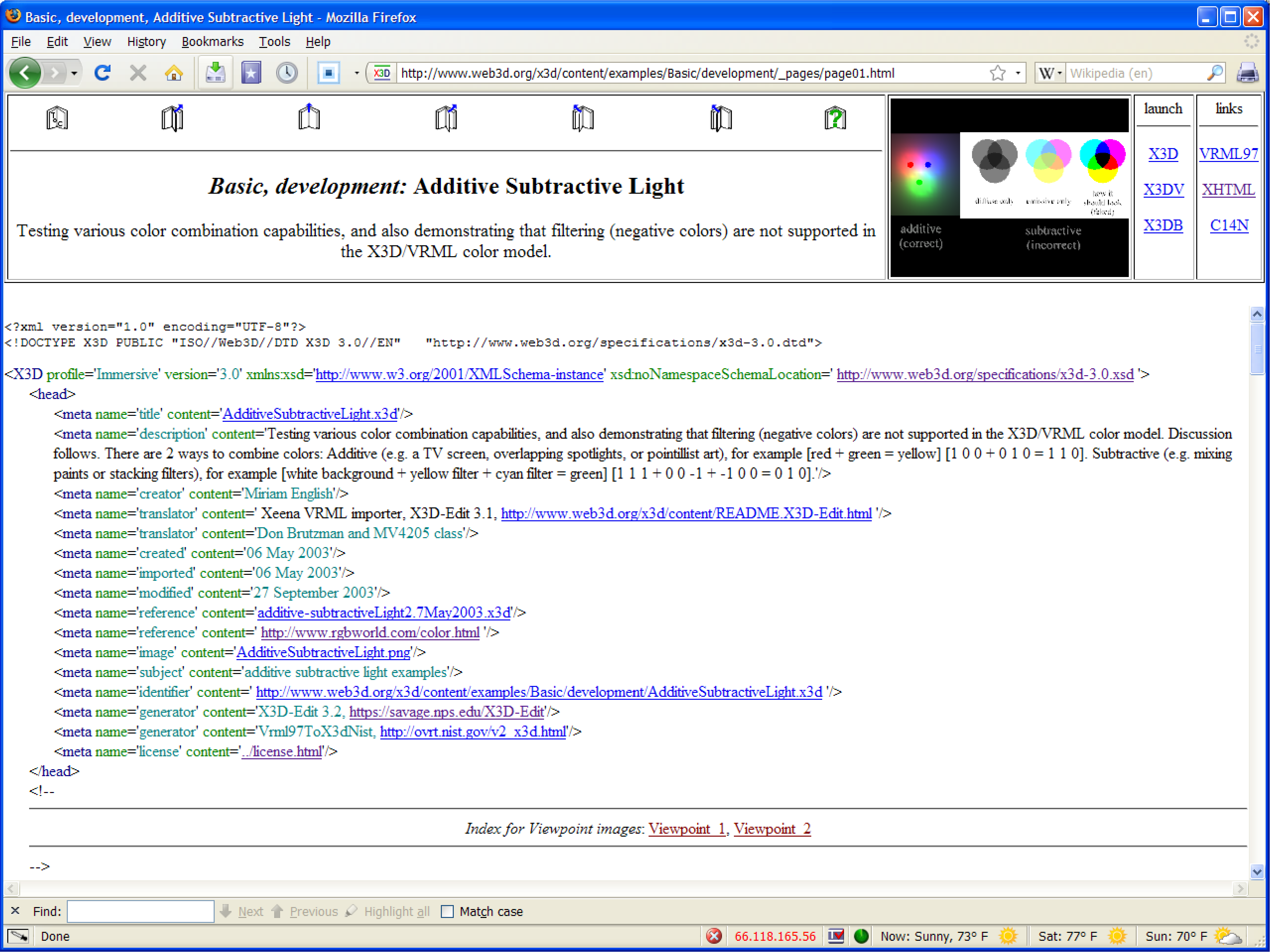
global ☑

☑ Visualize





```
<?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='Interactive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocat    ficati
    <head>
        <meta content='SpotLightColor.x3d' name='title'/>
        <meta content='Example of SpotLight showing effect of varying light colors on appearance.' name='description'/>
        <meta content='Leonard Daly and Don Brutzman' name='creator'/>
        <meta content='24 April 2006' name='created'/>
        <meta content='26 May 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 2006, Daly Realism and Don Brutzman' name='rights'/>
        <meta content='SpotLightColorWhite.png' name='image'/>
        <meta content='SpotLightColorYellow.png' name='image'/>
        <meta content='SpotLightColorWireframe.png' name='image'/>
        <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
        <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/SpotLightColor.x3d' name='identifier'/>
        <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
        <meta content='../license.html' name='license'/>
    </head>
    <Scene>
        <Background skyColor='1 1 1'/>
        <NavigationInfo headlight='false' type='"EXAMINE" "ANY"'/>
        <Viewpoint description='Book Viewpoint'/>
        <Viewpoint description='Complete scene' position='0 0 25'/>
        <SpotLight DEF='Lamp' beamWidth='.35' color='1 1 0' direction='0 0 -1' location='0 0 12' radius='18'/>
        <Transform>
            <Transform translation='0 0 0'>
                <Shape DEF='WhiteRectangleBackdrop'>
                    <Appearance>
                    <!-- flat box with many subpolygons to ensure smooth shading by light source -->
                    <IndexedFaceSet>
                </Shape>
            </Transform>
            <Shape DEF='PurpleCone'>
                <Appearance>
                <Cone bottomRadius='2' height='4' />
            </Shape>
            <Shape DEF='TurqoseCylinder'>
                <Appearance>
                <Cylinder height='4' radius='1.2' />
            </Shape>
        </Transform>
        <ColorInterpolator DEF='LightColor' key='0 .125 .375 .5 .625 .875 1' keyValue='1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1
        <ROUTE fromField='value_changed' fromNode='LightColor' toField='color' toNode='Lamp'/>
        <TimeSensor DEF='Timer' cycleInterval='8' enabled='true' loop='true'/>
        <ROUTE fromField='fraction_changed' fromNode='Timer' toField='set_fraction' toNode='LightColor'/>
    </Scene>
</X3D>
```



## *Basic, development:* Additive Subtractive Light

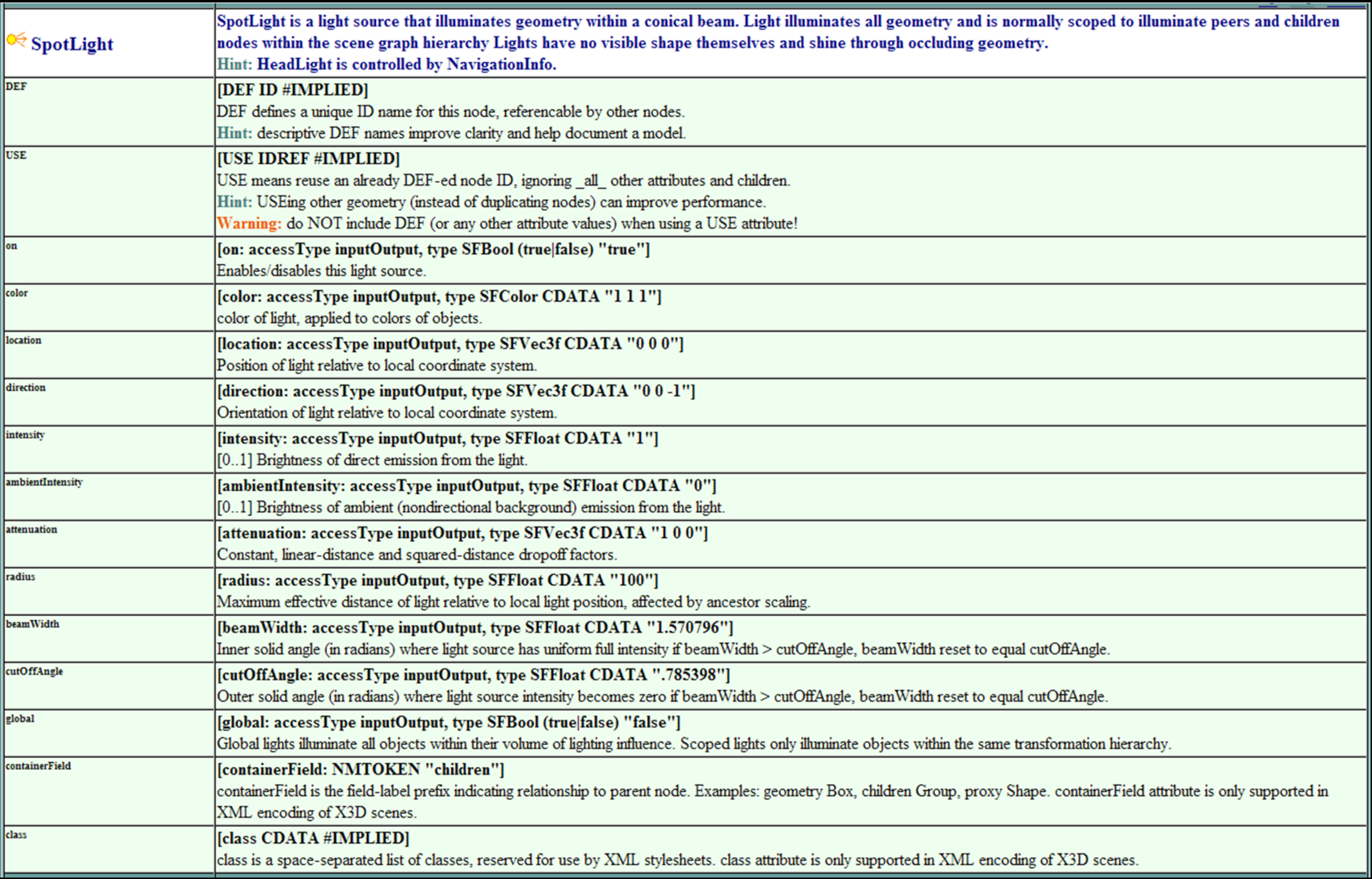
Testing various color combination capabilities, and also demonstrating that filtering (negative colors) are not supported in the X3D/VRML color model.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"   "http://www.web3d.org/specifications/x3d-3.0.dtd">

<X3D profile='Immersive' version='3.0' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation=' http://www.web3d.org/specifications/x3d-3.0.xsd '>
    <head>
        <meta name='title' content='AdditiveSubtractiveLight.x3d'/>
        <meta name='description' content='Testing various color combination capabilities, and also demonstrating that filtering (negative colors) are not supported in the X3D/VRML color model. Discussion follows. There are 2 ways to combine colors: Additive (e.g. a TV screen, overlapping spotlights, or pointillist art), for example [red + green = yellow] [1 0 0 + 0 1 0 = 1 1 0]. Subtractive (e.g. mixing paints or stacking filters), for example [white background + yellow filter + cyan filter = green] [1 1 1 + 0 0 -1 + -1 0 0 = 0 1 0].'/>
        <meta name='creator' content='Miriam English'/>
        <meta name='translator' content=' Xeena VRML importer, X3D-Edit 3.1, http://www.web3d.org/x3d/content/README.X3D-Edit.html '/>
        <meta name='translator' content='Don Brutzman and MV4205 class'/>
        <meta name='created' content='06 May 2003'/>
        <meta name='imported' content='06 May 2003'/>
        <meta name='modified' content='27 September 2003'/>
        <meta name='reference' content='additive-subtractiveLight2.7May2003.x3d'/>
        <meta name='reference' content=' http://www.rgbworld.com/color.html '/>
        <meta name='image' content='AdditiveSubtractiveLight.png'/>
        <meta name='subject' content='additive subtractive light examples'/>
        <meta name='identifier' content=' http://www.web3d.org/x3d/content/examples/Basic/development/AdditiveSubtractiveLight.x3d '/>
        <meta name='generator' content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit'/>
        <meta name='generator' content='Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html'/>
        <meta name='license' content='../license.html'/>
    </head>
    <!--
```

*Index for Viewpoint images:* Viewpoint 1, Viewpoint 2

```
    -->
```



| | |
|---|---|
| **SpotLight** | **SpotLight is a light source that illuminates geometry within a conical beam. Light illuminates all geometry and is normally scoped to illuminate peers and children nodes within the scene graph hierarchy Lights have no visible shape themselves and shine through occluding geometry.**<br>**Hint: HeadLight is controlled by NavigationInfo.** |
| **DEF** | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>**Hint:** descriptive DEF names improve clarity and help document a model. |
| **USE** | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>**Hint:** USEing other geometry (instead of duplicating nodes) can improve performance.<br>**Warning:** do NOT include DEF (or any other attribute values) when using a USE attribute! |
| **on** | **[on: accessType inputOutput, type SFBool (true\|false) "true"]**<br>Enables/disables this light source. |
| **color** | **[color: accessType inputOutput, type SFColor CDATA "1 1 1"]**<br>color of light, applied to colors of objects. |
| **location** | **[location: accessType inputOutput, type SFVec3f CDATA "0 0 0"]**<br>Position of light relative to local coordinate system. |
| **direction** | **[direction: accessType inputOutput, type SFVec3f CDATA "0 0 -1"]**<br>Orientation of light relative to local coordinate system. |
| **intensity** | **[intensity: accessType inputOutput, type SFFloat CDATA "1"]**<br>[0..1] Brightness of direct emission from the light. |
| **ambientIntensity** | **[ambientIntensity: accessType inputOutput, type SFFloat CDATA "0"]**<br>[0..1] Brightness of ambient (nondirectional background) emission from the light. |
| **attenuation** | **[attenuation: accessType inputOutput, type SFVec3f CDATA "1 0 0"]**<br>Constant, linear-distance and squared-distance dropoff factors. |
| **radius** | **[radius: accessType inputOutput, type SFFloat CDATA "100"]**<br>Maximum effective distance of light relative to local light position, affected by ancestor scaling. |
| **beamWidth** | **[beamWidth: accessType inputOutput, type SFFloat CDATA "1.570796"]**<br>Inner solid angle (in radians) where light source has uniform full intensity if beamWidth > cutOffAngle, beamWidth reset to equal cutOffAngle. |
| **cutOffAngle** | **[cutOffAngle: accessType inputOutput, type SFFloat CDATA ".785398"]**<br>Outer solid angle (in radians) where light source intensity becomes zero if beamWidth > cutOffAngle, beamWidth reset to equal cutOffAngle. |
| **global** | **[global: accessType inputOutput, type SFBool (true\|false) "false"]**<br>Global lights illuminate all objects within their volume of lighting influence. Scoped lights only illuminate objects within the same transformation hierarchy. |
| **containerField** | **[containerField: NMTOKEN "children"]**<br>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]**<br>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

# Environment Concepts

# Environmental effects

Three nodes provide environmental effects that can increase overall realism of an X3D scene

- Background defines either color arrays or images for defining the scene horizon, including vertical variations that emulate both sky and earth/sea
- TextureBackground similarly defines background but uses ImageTexture nodes instead of image urls
- Fog emulates actual fog by washing out pixels according to distance from camera, usually to match color (or blackness) of the background

Helpful techniques to improve scene realism

# X3DBackgroundNode type 1

The X3DBackgroundNode type defines common functionality and fields for background nodes

- Background and TextureBackground

Each node is bindable, on same binding stack

- Thus only one Background or TextureBackground node can be active at a single time
- Similar to Viewpoint, NavigationInfo, Fog

Binding nodes explained in detail for Viewpoint

- Chapter 4, Viewing and Navigation
- Same techniques apply for all four binding stacks

# X3DBackgroundNode type 2

Conceptually the background nodes apply colors or images to the horizon

- Color arrays are mapped to a distant sphere
- Images are mapped to a distant box

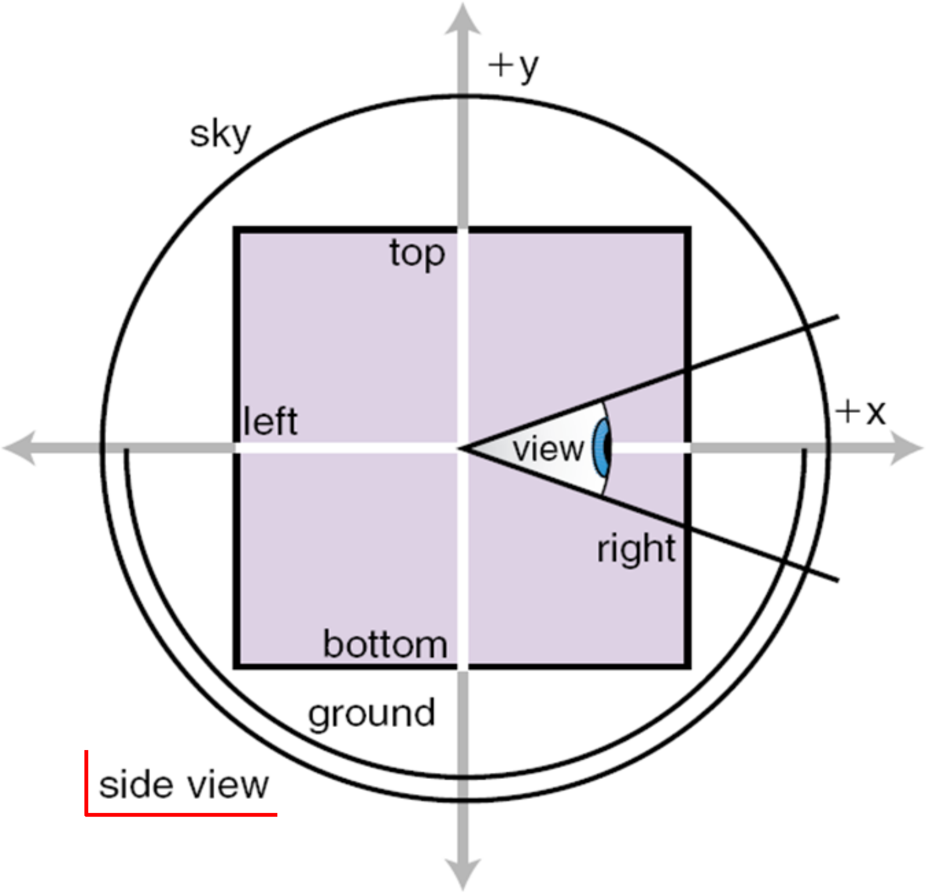
The user viewpoint is not able to approach these background colors and images, since they are always kept at the horizon

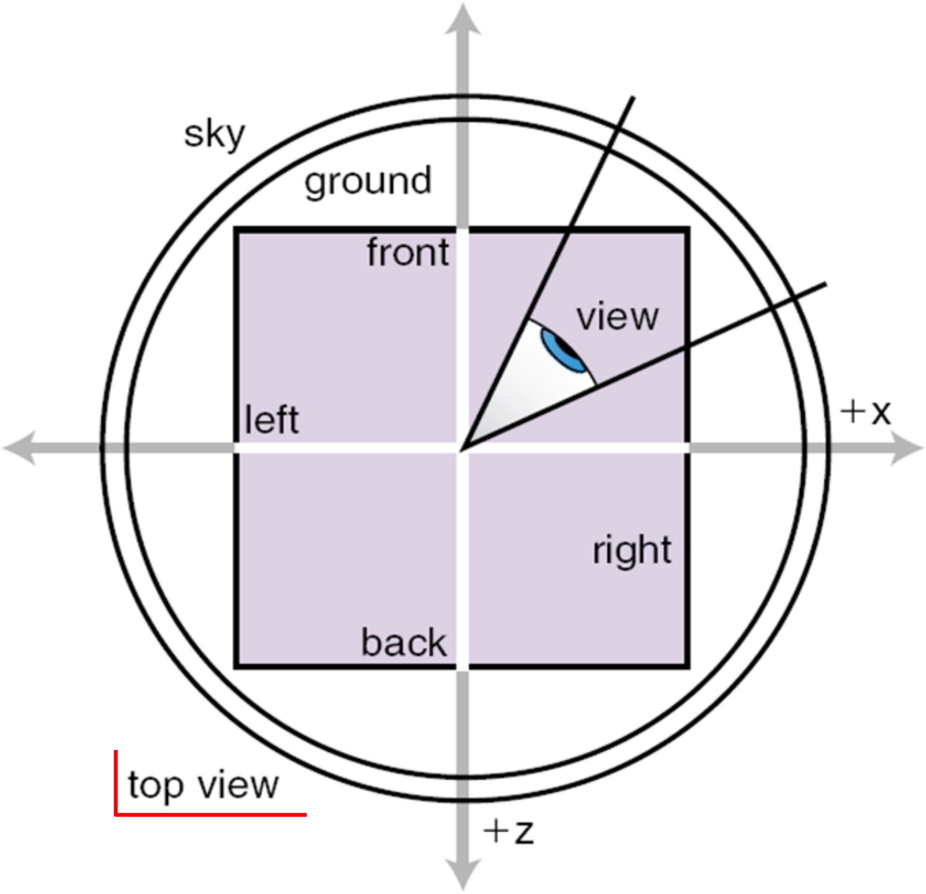
- Other objects in the scene always appear in front, and cannot be placed behind the background

Affected by parent rotations, not translations

# X3DBackgroundNode fields 1

Side and top views show geometric relationships for sky+ground, top+bottom, left+right, front+back





# X3DBackgroundNode fields 2

- *groundAngle* and *skyAngle* arrays list angles corresponding to color values for ground and sky
- *groundAngle* in radians monotonically increases from 0.0 (nadir, or straight down) to pi/2 (horizon), going upward
- *skyAngle* in radians monotonically increases from 0.0 (zenith, or straight up) to pi/2 (horizon) and on to pi (nadir, or straight down) , going downward
- *groundColor* and *skyColor* arrays provide corresponding color values for each angle, intermediate RGB colors are smoothly interpolated

# X3DBackgroundNode fields 3

Constraints on angles and colors:

- First color in array is zenith/nadir, no angle needed

  *skyAngle.*length + 1 = *skyColor.*length

  *groundAngle.*length + 1 = *groundColor.*length

- Each angle value must monotonically increase (thus repetition is allowed for constant color bands)
- When duplicate angular coverage is provided, *groundColor* values obscure *skyColor* values (but only as far as the horizon, by definition)
- If no *groundColor* values are provided, then final *skyColor* value is clamped and continues to nadir
  - Thus sole skyColor value defines entire background

# Environment Nodes and Examples

# Background node: color arrays

Background can provide simple colors or images for ground and sky, which appear behind any other geometric shapes in the scene

Background implements X3DBackgroundNode and X3DBindableNode types

Background colors form horizontal circular bands at the user's horizon which gradually change from one color value to the next along the vertical direction

# Background node: image files via url

Images (if used) form a box around user

- Once again, positioned at distance of horizon
- Top+bottom, left+right, front+back

Not all images required, but usually at least 4 side images included for visual coverage

Various url fields are each arrays

- Allowing both local, remote addresses for each file

Transparent pixels are allowed in image textures

- If present, background colors appear through them

# Background fields

*topUrl, bottomUrl, leftUrl, rightUrl, frontUrl, backUrl*

- Separate url arrays for each image, for reliability

X3DBindableNode fields

- *set_bind* is inputOnly event to bind or unbind node
- *isBound* is outputOnly event indicating whether the node is bound or unbound
- *bindTime* is outputOnly event providing the time that the node is bound

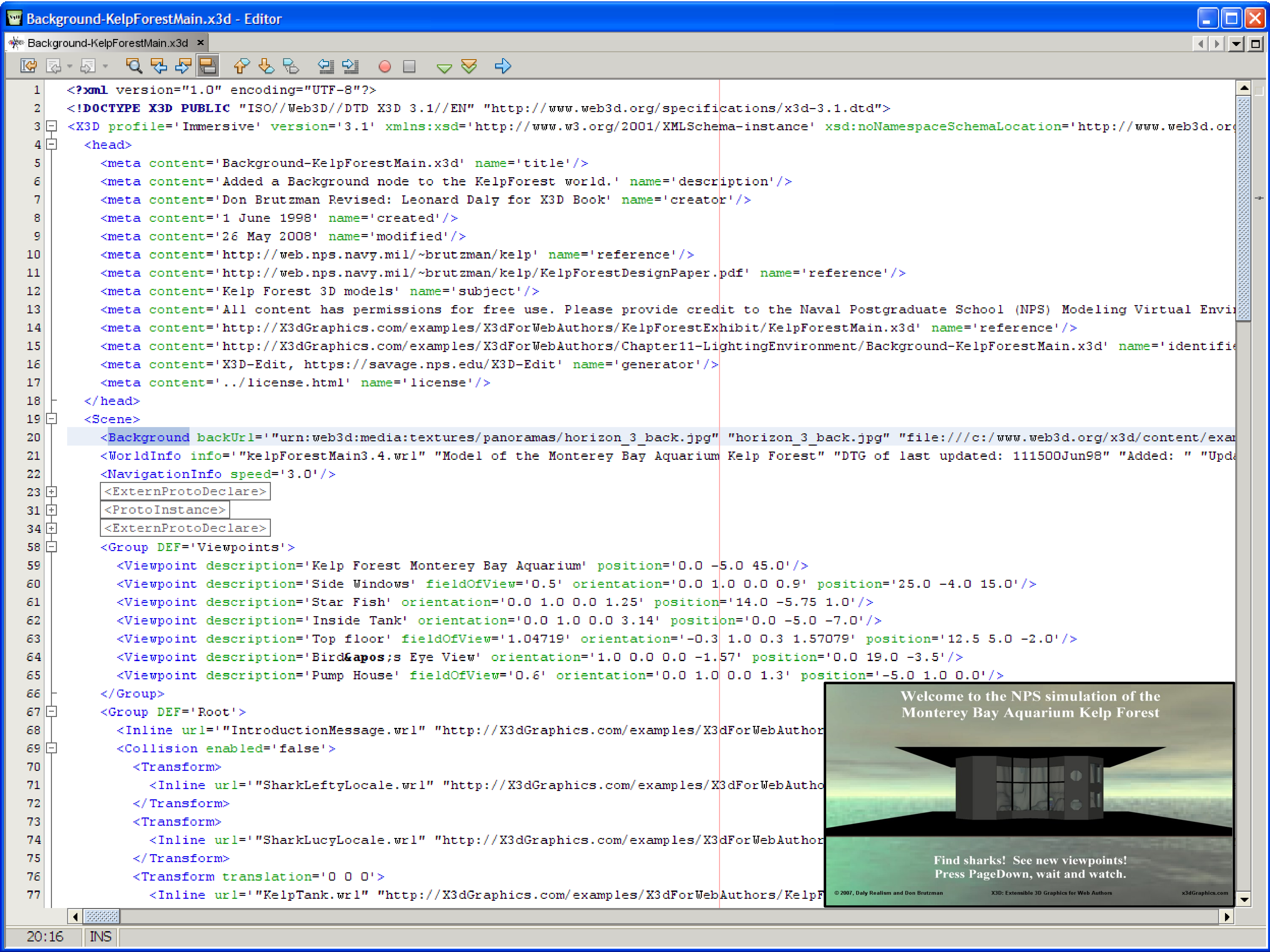
# UniversalMediaPanorama backgrounds

The UniversalMedia project includes a number of high-quality image backgrounds

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

Excerpt showing large number of url addresses:

```
<Background
   backUrl='
   "urn:web3d:media:textures/panoramas/desert_1_back.jpg"
   "desert_1_back.jpg"
   "file:///c:/www.web3d.org/x3d/content/examples/UniversalMediaPanoramas/desert_1_back.jpg"
   "http://www.web3d.org/WorkingGroups/media/textures/panoramas/desert_1_back.jpg"
   "http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaPanoramas/desert_1_back.jpg"
   "http://www.web3dmedia.com/UniversalMedia/textures/panoramas/desert_1_back.jpg"
   "http://www.officetowers.com/UniversalMedia/textures/panoramas/desert_1_back.jpg"
   "http://geometrek.com/UniversalMedia/textures/panoramas/desert_1_back.jpg"
   "http://www.sc.ehu.es/ccwgamoa/UniversalMedia/textures/panoramas/desert_1_back.jpg" '
            /> <!-- etc. for other fields -->
```



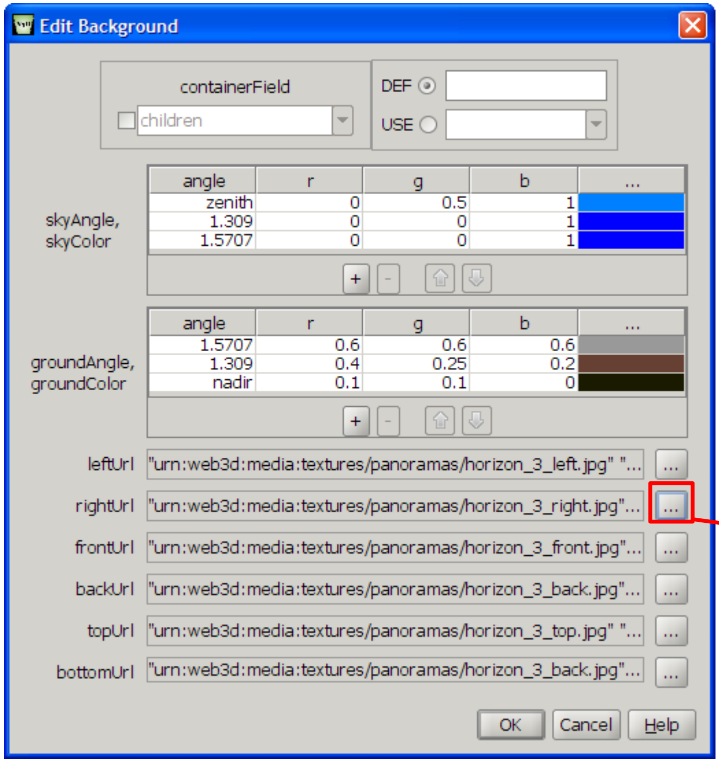
Background-KelpForestMain.x3d - Editor

Background-KelpForestMain.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.or
  <head>
    <meta content='Background-KelpForestMain.x3d' name='title'/>
    <meta content='Added a Background node to the KelpForest world.' name='description'/>
    <meta content='Don Brutzman Revised: Leonard Daly for X3D Book' name='creator'/>
    <meta content='1 June 1998' name='created'/>
    <meta content='26 May 2008' name='modified'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp' name='reference'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp/KelpForestDesignPaper.pdf' name='reference'/>
    <meta content='Kelp Forest 3D models' name='subject'/>
    <meta content='All content has permissions for free use. Please provide credit to the Naval Postgraduate School (NPS) Modeling Virtual Envi
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/Background-KelpForestMain.x3d' name='identifie
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Background backUrl='"urn:web3d:media:textures/panoramas/horizon_3_back.jpg" "horizon_3_back.jpg" "file:///c:/www.web3d.org/x3d/content/exam
    <WorldInfo info='"kelpForestMain3.4.wrl" "Model of the Monterey Bay Aquarium Kelp Forest" "DTG of last updated: 111500Jun98" "Added: " "Upda
    <NavigationInfo speed='3.0'/>
    <ExternProtoDeclare>
    <ProtoInstance>
    <ExternProtoDeclare>
    <Group DEF='Viewpoints'>
      <Viewpoint description='Kelp Forest Monterey Bay Aquarium' position='0.0 -5.0 45.0'/>
      <Viewpoint description='Side Windows' fieldOfView='0.5' orientation='0.0 1.0 0.0 0.9' position='25.0 -4.0 15.0'/>
      <Viewpoint description='Star Fish' orientation='0.0 1.0 0.0 1.25' position='14.0 -5.75 1.0'/>
      <Viewpoint description='Inside Tank' orientation='0.0 1.0 0.0 3.14' position='0.0 -5.0 -7.0'/>
      <Viewpoint description='Top floor' fieldOfView='1.04719' orientation='-0.3 1.0 0.3 1.57079' position='12.5 5.0 -2.0'/>
      <Viewpoint description='Bird&apos;s Eye View' orientation='1.0 0.0 0.0 -1.57' position='0.0 19.0 -3.5'/>
      <Viewpoint description='Pump House' fieldOfView='0.6' orientation='0.0 1.0 0.0 1.3' position='-5.0 1.0 0.0'/>
    </Group>
    <Group DEF='Root'>
      <Inline url='"IntroductionMessage.wrl" "http://X3dGraphics.com/examples/X3dForWebAuthor
      <Collision enabled='false'>
        <Transform>
          <Inline url='"SharkLeftyLocale.wrl" "http://X3dGraphics.com/examples/X3dForWebAutho
        </Transform>
        <Transform>
          <Inline url='"SharkLucyLocale.wrl" "http://X3dGraphics.com/examples/X3dForWebAuthor
        </Transform>
        <Transform translation='0 0 0'>
          <Inline url='"KelpTank.wrl" "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpF
```

Welcome to the NPS simulation of the
Monterey Bay Aquarium Kelp Forest
Find sharks! See new viewpoints!
Press PageDown, wait and watch.
© 2007, Daly Realism and Don Brutzman
X3D: Extensible 3D Graphics for Web Authors
x3dGraphics.com
20:16
INS

# Background node X3D-Edit



Edit Background

containerField ☐ children

DEF ●

USE ○

skyAngle, skyColor

| angle | r | g | b | ... |
|---|---|---|---|---|
| zenith | 0 | 0.5 | 1 | 🟦 |
| 1.309 | 0 | 0 | 1 | 🟦 |
| 1.5707 | 0 | 0 | 1 | 🟦 |

groundAngle, groundColor

| angle | r | g | b | ... |
|---|---|---|---|---|
| 1.5707 | 0.6 | 0.6 | 0.6 | ▨ |
| 1.309 | 0.4 | 0.25 | 0.2 | 🟫 |
| nadir | 0.1 | 0.1 | 0 | ⬛ |

leftUrl "urn:web3d:media:textures/panoramas/horizon_3_left.jpg" "...

rightUrl "urn:web3d:media:textures/panoramas/horizon_3_right.jpg"...

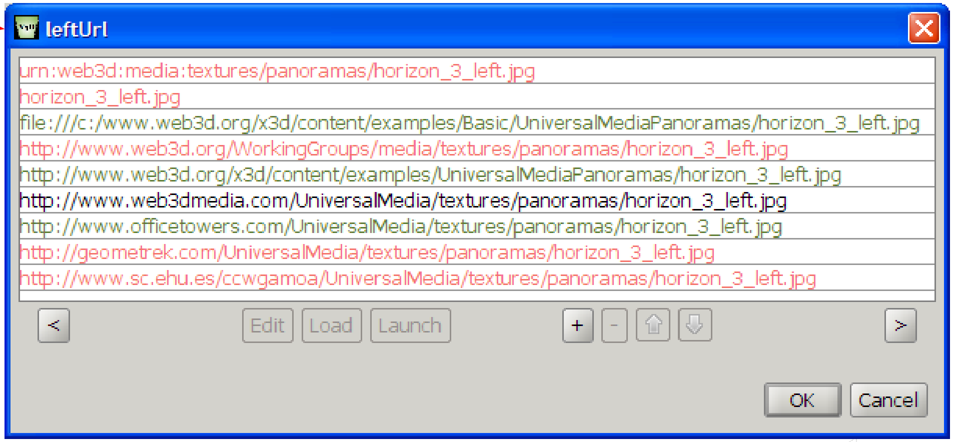
frontUrl "urn:web3d:media:textures/panoramas/horizon_3_front.jpg"...

backUrl "urn:web3d:media:textures/panoramas/horizon_3_back.jpg"...

topUrl "urn:web3d:media:textures/panoramas/horizon_3_top.jpg" "...

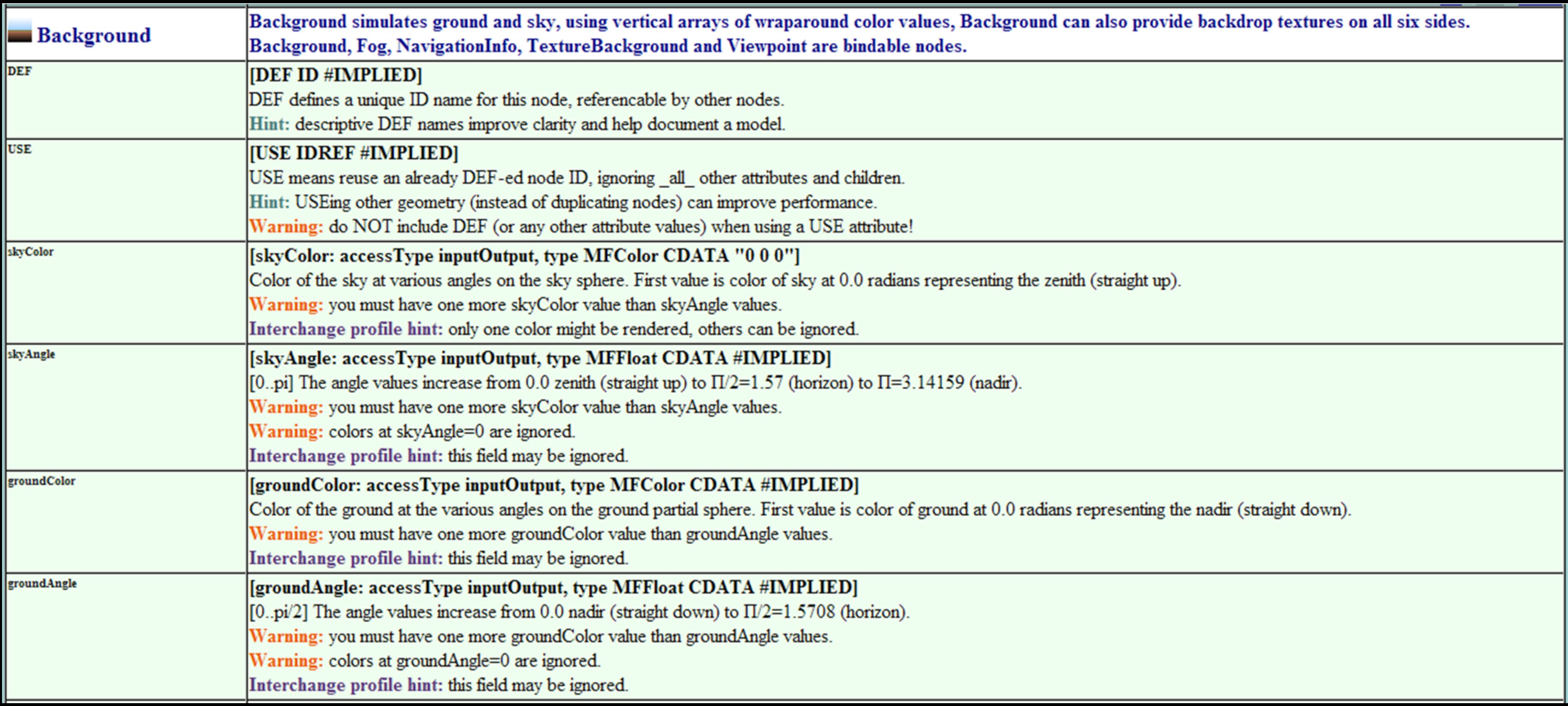
bottomUrl "urn:web3d:media:textures/panoramas/horizon_3_back.jpg"...

- *skyColor* values go down from zenith to nadir
- *groundColor* values go up from nadir to horizon

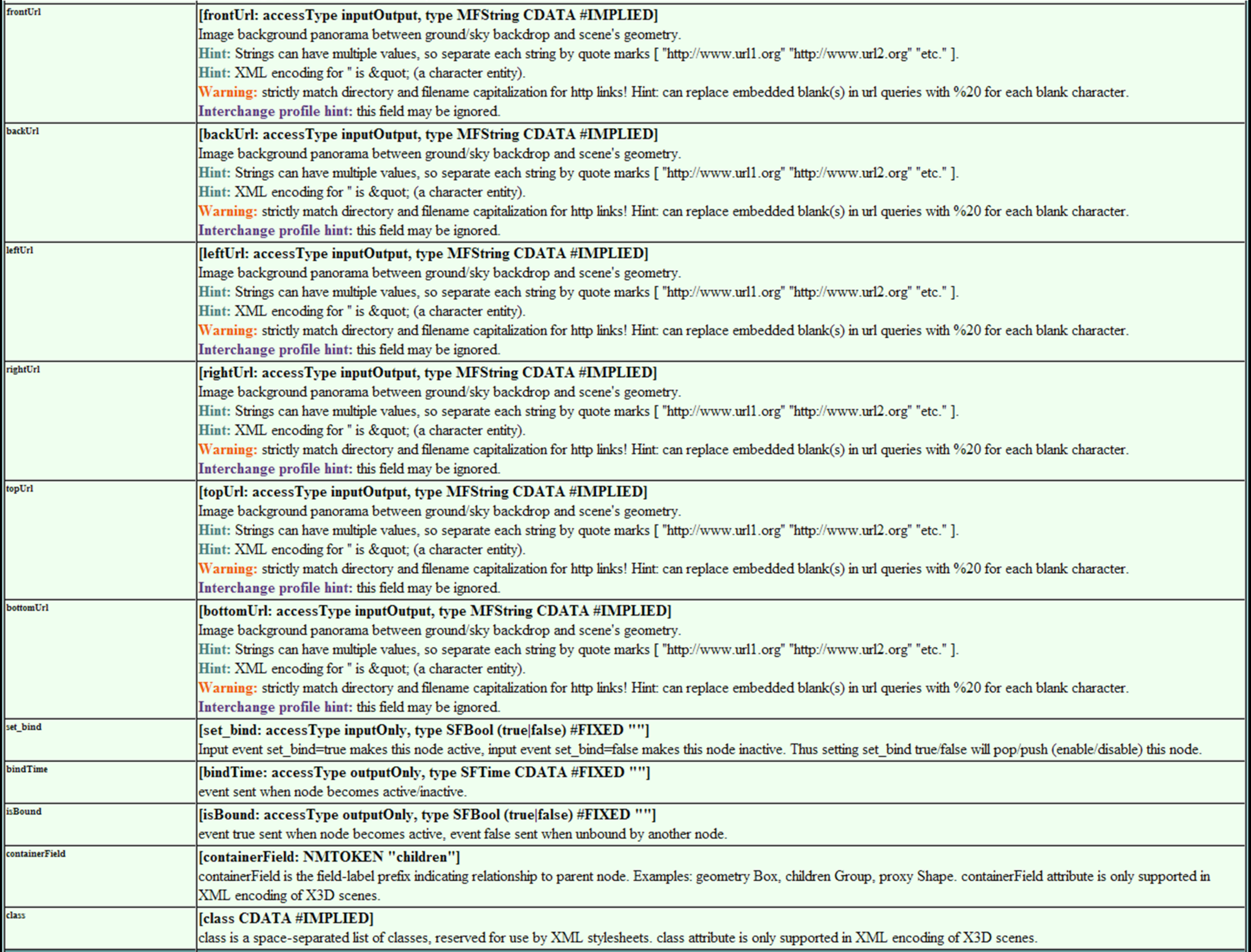


leftUrl

urn:web3d:media:textures/panoramas/horizon_3_left.jpg
horizon_3_left.jpg
file:///c:/www.web3d.org/x3d/content/examples/Basic/UniversalMediaPanoramas/horizon_3_left.jpg
http://www.web3d.org/WorkingGroups/media/textures/panoramas/horizon_3_left.jpg
http://www.web3d.org/x3d/content/examples/UniversalMediaPanoramas/horizon_3_left.jpg
http://www.web3dmedia.com/UniversalMedia/textures/panoramas/horizon_3_left.jpg
http://www.officetowers.com/UniversalMedia/textures/panoramas/horizon_3_left.jpg
http://geometrek.com/UniversalMedia/textures/panoramas/horizon_3_left.jpg
http://www.sc.ehu.es/ccwgamoa/UniversalMedia/textures/panoramas/horizon_3_left.jpg



| | |
|---|---|
| **Background** | **Background simulates ground and sky, using vertical arrays of wraparound color values, Background can also provide backdrop textures on all six sides.**<br>**Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes.** |
| **DEF** | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>**Hint:** descriptive DEF names improve clarity and help document a model. |
| **USE** | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>**Hint:** USEing other geometry (instead of duplicating nodes) can improve performance.<br>**Warning:** do NOT include DEF (or any other attribute values) when using a USE attribute! |
| **skyColor** | **[skyColor: accessType inputOutput, type MFColor CDATA "0 0 0"]**<br>Color of the sky at various angles on the sky sphere. First value is color of sky at 0.0 radians representing the zenith (straight up).<br>**Warning:** you must have one more skyColor value than skyAngle values.<br>**Interchange profile hint:** only one color might be rendered, others can be ignored. |
| **skyAngle** | **[skyAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi] The angle values increase from 0.0 zenith (straight up) to Π/2=1.57 (horizon) to Π=3.14159 (nadir).<br>**Warning:** you must have one more skyColor value than skyAngle values.<br>**Warning:** colors at skyAngle=0 are ignored.<br>**Interchange profile hint:** this field may be ignored. |
| **groundColor** | **[groundColor: accessType inputOutput, type MFColor CDATA #IMPLIED]**<br>Color of the ground at the various angles on the ground partial sphere. First value is color of ground at 0.0 radians representing the nadir (straight down).<br>**Warning:** you must have one more groundColor value than groundAngle values.<br>**Interchange profile hint:** this field may be ignored. |
| **groundAngle** | **[groundAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi/2] The angle values increase from 0.0 nadir (straight down) to Π/2=1.5708 (horizon).<br>**Warning:** you must have one more groundColor value than groundAngle values.<br>**Warning:** colors at groundAngle=0 are ignored.<br>**Interchange profile hint:** this field may be ignored. |



| | |
|---|---|
| frontUrl | **[frontUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| backUrl | **[backUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| leftUrl | **[leftUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| rightUrl | **[rightUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| topUrl | **[topUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| bottomUrl | **[bottomUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| set_bind | **[set_bind: accessType inputOnly, type SFBool (true\|false) #FIXED ""]**<br>Input event set_bind=true makes this node active, input event set_bind=false makes this node inactive. Thus setting set_bind true/false will pop/push (enable/disable) this node. |
| bindTime | **[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]**<br>event sent when node becomes active/inactive. |
| isBound | **[isBound: accessType outputOnly, type SFBool (true\|false) #FIXED ""]**<br>event true sent when node becomes active, event false sent when unbound by another node. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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. |

# TextureBackground node

TextureBackground nearly same as Background

- Includes ground and sky color, angle arrays
- Coexists on same bindable node stack

Instead of using six *url* arrays, instead use six ImageTexture nodes for background images

- Thus allows reuse of ImageTexture nodes if needed

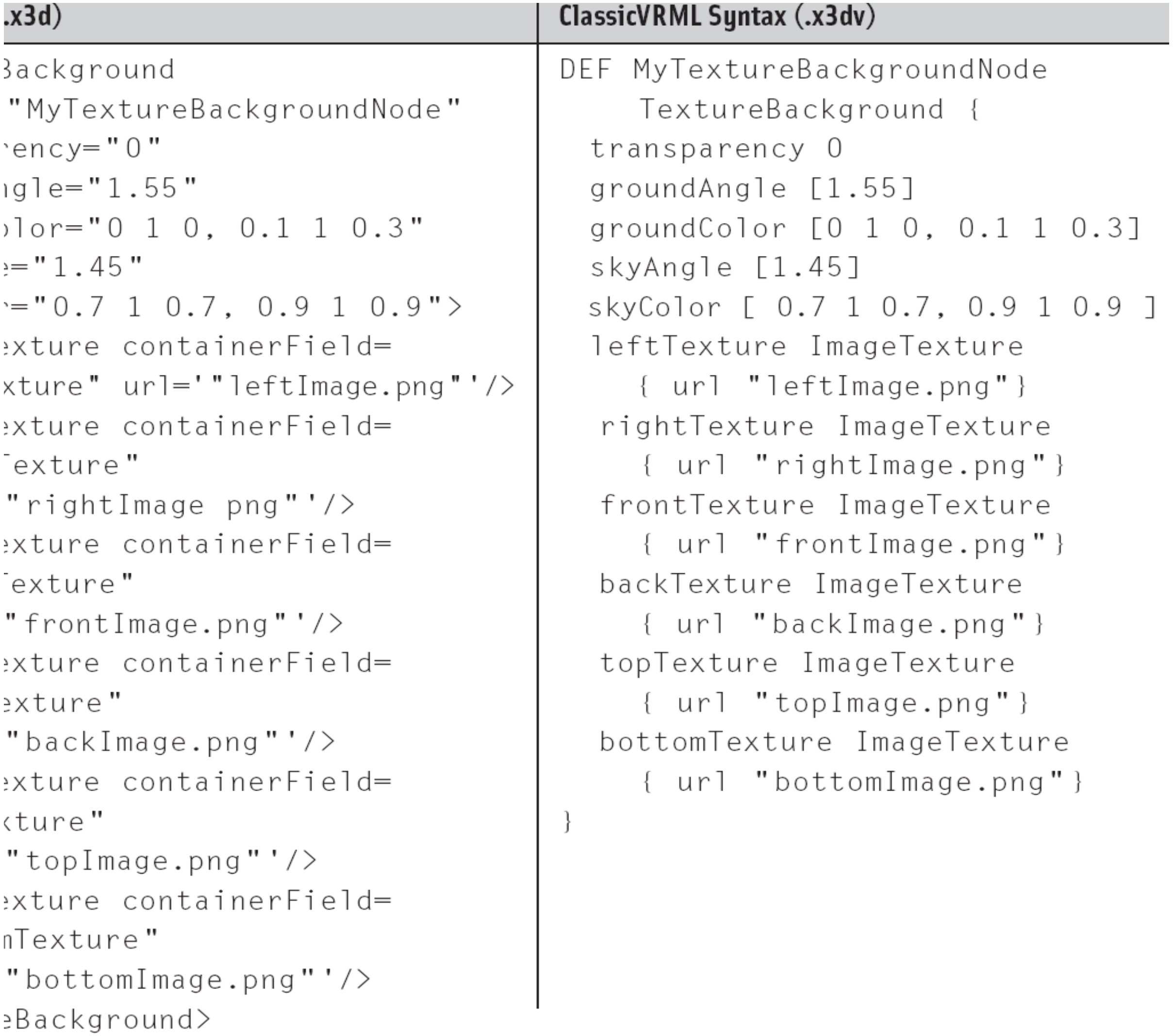
Requires `<component name='EnvironmentalEffects' level='3'/>`

Also allows overall *transparency* value, so that anything on user's desktop or behind browser can be seen behind objects in the scene

- Might be a useful specialty display technique

# TextureBackground syntax comparison

Note differences in use of *containerField* for XML syntax, versus use of field name for ClassicVRML syntax

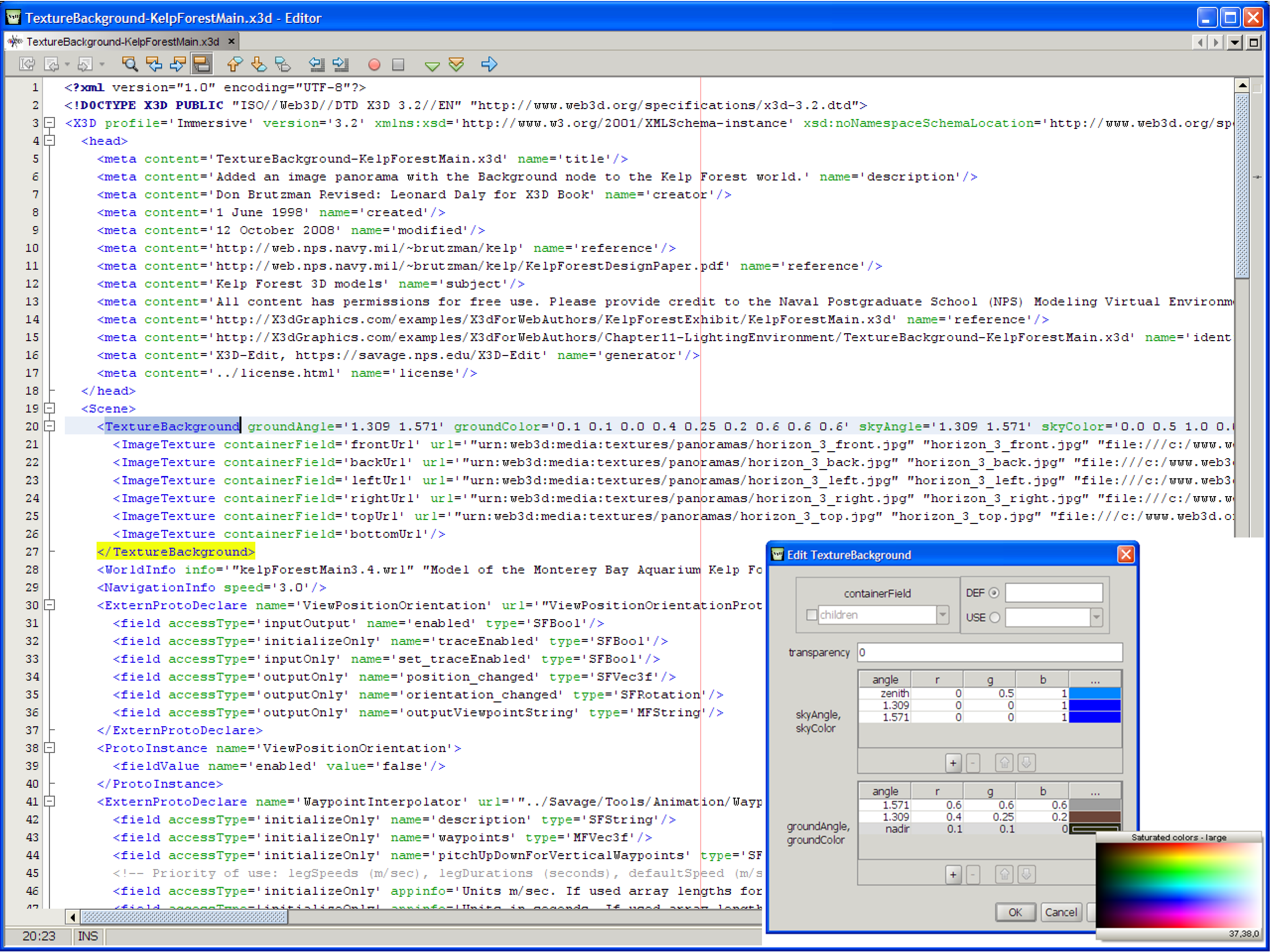


.x3d)

```
ackground
 "MyTextureBackgroundNode"
ency="0"
gle="1.55"
lor="0 1 0, 0.1 1 0.3"
="1.45"
="0.7 1 0.7, 0.9 1 0.9">
xture containerField=
xture" url='"leftImage.png"'/>
xture containerField=
exture"
 "rightImage png"'/>
xture containerField=
exture"
 "frontImage.png"'/>
xture containerField=
xture"
 "backImage.png"'/>
xture containerField=
ture"
 "topImage.png"'/>
xture containerField=
Texture"
 "bottomImage.png"'/>
Background>
```

ClassicVRML Syntax (.x3dv)

```
DEF MyTextureBackgroundNode
    TextureBackground {
  transparency 0
  groundAngle [1.55]
  groundColor [0 1 0, 0.1 1 0.3]
  skyAngle [1.45]
  skyColor [ 0.7 1 0.7, 0.9 1 0.9 ]
  leftTexture ImageTexture
     { url "leftImage.png"}
  rightTexture ImageTexture
     { url "rightImage.png"}
  frontTexture ImageTexture
     { url "frontImage.png"}
  backTexture ImageTexture
     { url "backImage.png"}
  topTexture ImageTexture
     { url "topImage.png"}
  bottomTexture ImageTexture
     { url "bottomImage.png"}
}
```



TextureBackground-KelpForestMain.x3d - Editor
TextureBackground-KelpForestMain.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='TextureBackground-KelpForestMain.x3d' name='title'/>
    <meta content='Added an image panorama with the Background node to the Kelp Forest world.' name='description'/>
    <meta content='Don Brutzman Revised: Leonard Daly for X3D Book' name='creator'/>
    <meta content='1 June 1998' name='created'/>
    <meta content='12 October 2008' name='modified'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp' name='reference'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp/KelpForestDesignPaper.pdf' name='reference'/>
    <meta content='Kelp Forest 3D models' name='subject'/>
    <meta content='All content has permissions for free use. Please provide credit to the Naval Postgraduate School (NPS) Modeling Virtual Environm
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/TextureBackground-KelpForestMain.x3d' name='ident
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <TextureBackground groundAngle='1.309 1.571' groundColor='0.1 0.1 0.0 0.4 0.25 0.2 0.6 0.6 0.6' skyAngle='1.309 1.571' skyColor='0.0 0.5 1.0 0.
      <ImageTexture containerField='frontUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_front.jpg" "horizon_3_front.jpg" "file:///c:/www.w
      <ImageTexture containerField='backUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_back.jpg" "horizon_3_back.jpg" "file:///c:/www.web3
      <ImageTexture containerField='leftUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_left.jpg" "horizon_3_left.jpg" "file:///c:/www.web3
      <ImageTexture containerField='rightUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_right.jpg" "horizon_3_right.jpg" "file:///c:/www.w
      <ImageTexture containerField='topUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_top.jpg" "horizon_3_top.jpg" "file:///c:/www.web3d.o
      <ImageTexture containerField='bottomUrl'/>
    </TextureBackground>
    <WorldInfo info='"kelpForestMain3.4.wrl" "Model of the Monterey Bay Aquarium Kelp Fo
    <NavigationInfo speed='3.0'/>
    <ExternProtoDeclare name='ViewPositionOrientation' url='"ViewPositionOrientationProt
      <field accessType='inputOutput' name='enabled' type='SFBool'/>
      <field accessType='initializeOnly' name='traceEnabled' type='SFBool'/>
      <field accessType='inputOnly' name='set_traceEnabled' type='SFBool'/>
      <field accessType='outputOnly' name='position_changed' type='SFVec3f'/>
      <field accessType='outputOnly' name='orientation_changed' type='SFRotation'/>
      <field accessType='outputOnly' name='outputViewpointString' type='MFString'/>
    </ExternProtoDeclare>
    <ProtoInstance name='ViewPositionOrientation'>
      <fieldValue name='enabled' value='false'/>
    </ProtoInstance>
    <ExternProtoDeclare name='WaypointInterpolator' url='"../Savage/Tools/Animation/Wayp
      <field accessType='initializeOnly' name='description' type='SFString'/>
      <field accessType='initializeOnly' name='waypoints' type='MFVec3f'/>
      <field accessType='initializeOnly' name='pitchUpDownForVerticalWaypoints' type='SF
      <!-- Priority of use: legSpeeds (m/sec), legDurations (seconds), defaultSpeed (m/s
      <field accessType='initializeOnly' appinfo='Units m/sec. If used array lengths for
      <field accessType='initializeOnly' appinfo='Units in seconds. If used array length
```

20:23 INS

Edit TextureBackground

containerField ☐ children

DEF ●

USE ○

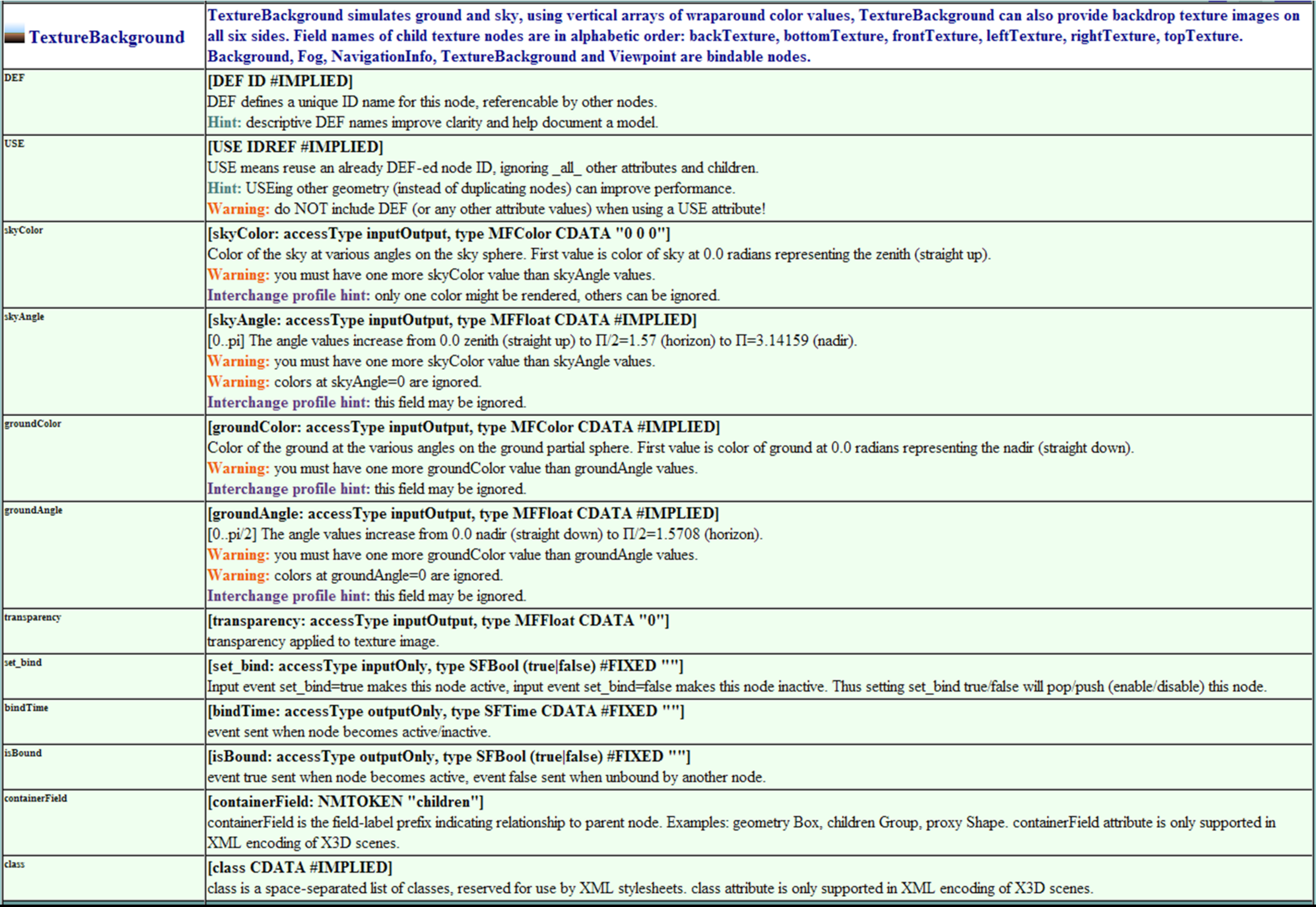
transparency 0

skyAngle, skyColor

| angle | r | g | b | ... |
|---|---|---|---|---|
| zenith | 0 | 0.5 | 1 | 🟦 |
| 1.309 | 0 | 0 | 1 | 🟦 |
| 1.571 | 0 | 0 | 1 | 🟦 |

groundAngle, groundColor

| angle | r | g | b | ... |
|---|---|---|---|---|
| 1.571 | 0.6 | 0.6 | 0.6 | ⬜ |
| 1.309 | 0.4 | 0.25 | 0.2 | 🟫 |
| nadir | 0.1 | 0.1 | 0 | ⬛ |



| TextureBackground | TextureBackground simulates ground and sky, using vertical arrays of wraparound color values, TextureBackground can also provide backdrop texture images on all six sides. Field names of child texture nodes are in alphabetic order: backTexture, bottomTexture, frontTexture, leftTexture, rightTexture, topTexture. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes. |
|---|---|
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>Hint: descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.<br>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute! |
| skyColor | **[skyColor: accessType inputOutput, type MFColor CDATA "0 0 0"]**<br>Color of the sky at various angles on the sky sphere. First value is color of sky at 0.0 radians representing the zenith (straight up).<br>Warning: you must have one more skyColor value than skyAngle values.<br>Interchange profile hint: only one color might be rendered, others can be ignored. |
| skyAngle | **[skyAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi] The angle values increase from 0.0 zenith (straight up) to Π/2=1.57 (horizon) to Π=3.14159 (nadir).<br>Warning: you must have one more skyColor value than skyAngle values.<br>Warning: colors at skyAngle=0 are ignored.<br>Interchange profile hint: this field may be ignored. |
| groundColor | **[groundColor: accessType inputOutput, type MFColor CDATA #IMPLIED]**<br>Color of the ground at the various angles on the ground partial sphere. First value is color of ground at 0.0 radians representing the nadir (straight down).<br>Warning: you must have one more groundColor value than groundAngle values.<br>Interchange profile hint: this field may be ignored. |
| groundAngle | **[groundAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi/2] The angle values increase from 0.0 nadir (straight down) to Π/2=1.5708 (horizon).<br>Warning: you must have one more groundColor value than groundAngle values.<br>Warning: colors at groundAngle=0 are ignored.<br>Interchange profile hint: this field may be ignored. |
| transparency | **[transparency: accessType inputOutput, type MFFloat CDATA "0"]**<br>transparency applied to texture image. |
| set_bind | **[set_bind: accessType inputOnly, type SFBool (true\|false) #FIXED ""]**<br>Input event set_bind=true makes this node active, input event set_bind=false makes this node inactive. Thus setting set_bind true/false will pop/push (enable/disable) this node. |
| bindTime | **[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]**<br>event sent when node becomes active/inactive. |
| isBound | **[isBound: accessType outputOnly, type SFBool (true\|false) #FIXED ""]**<br>event true sent when node becomes active, event false sent when unbound by another node. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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. |

# Fog node

Fog is an important environmental effect

- Use it sparingly for best effect
- Another X3DBindableNode

White or grey fog can emulate fog in real world

Black fog can emulate night-time effects

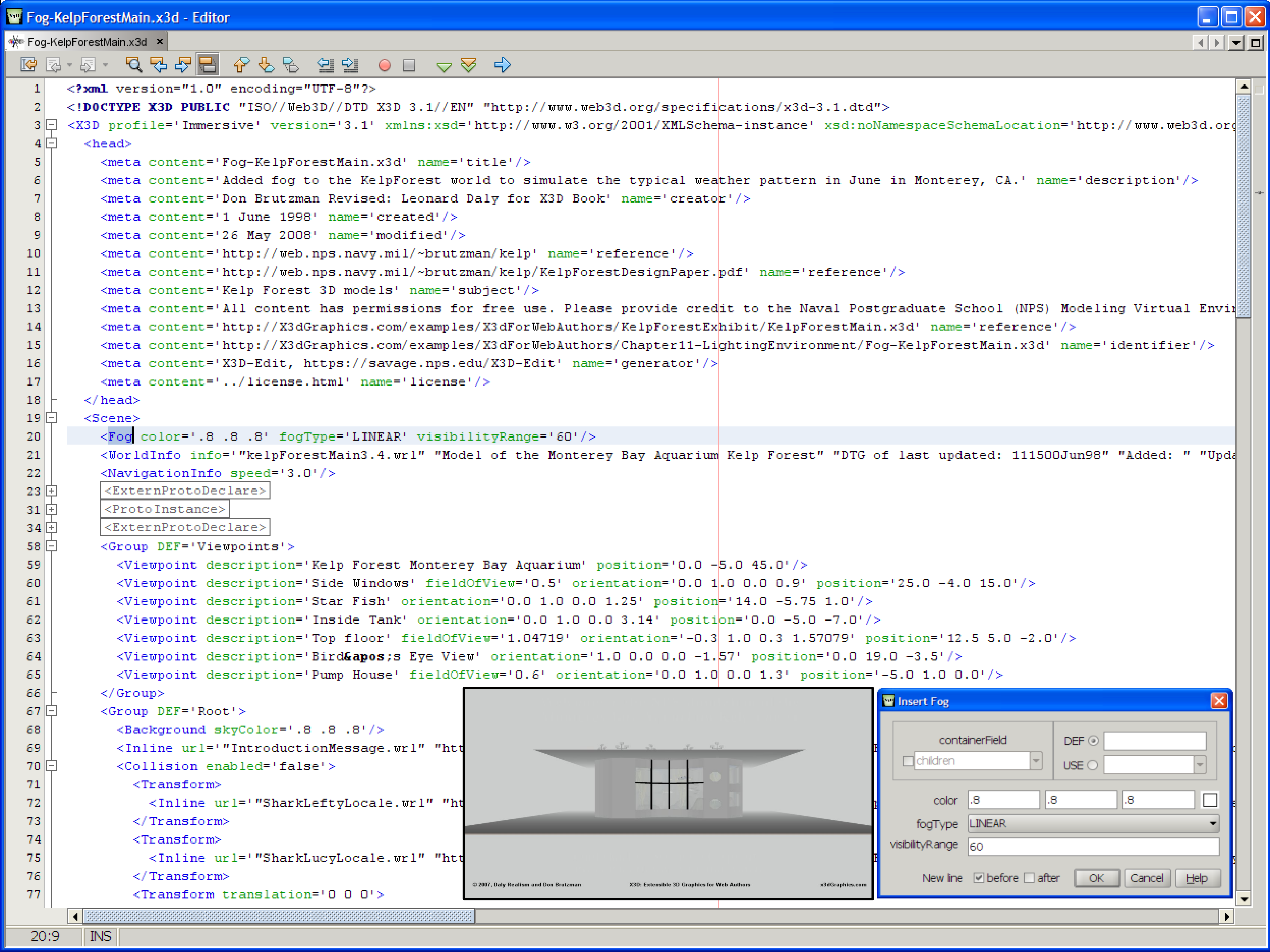
Fog works by gradually recoloring scene pixels to match fog color, depending on range to viewer

Thus fog color must match Background color to work properly, otherwise silhouettes created

- Experimentation is helpful to get things right

# Fog fields

- *color* field is single RGB value for replacement color
- *fogType* has two defined enumeration values for rate of dropoff in visibility as viewer range increases
  - LINEAR indicates linear dropoff
  - EXPONENTIAL indicates exponential dropoff, which increases the fog effect and is more computationally expensive
- *visibilityRange* indicates distance from view camera where objects become completely obscured
  - Avoid scaling in parent transformation hierarchy, otherwise this value will no longer be in meters



Fog-KelpForestMain.x3d - Editor

Fog-KelpForestMain.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
  <head>
    <meta content='Fog-KelpForestMain.x3d' name='title'/>
    <meta content='Added fog to the KelpForest world to simulate the typical weather pattern in June in Monterey, CA.' name='description'/>
    <meta content='Don Brutzman Revised: Leonard Daly for X3D Book' name='creator'/>
    <meta content='1 June 1998' name='created'/>
    <meta content='26 May 2008' name='modified'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp' name='reference'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp/KelpForestDesignPaper.pdf' name='reference'/>
    <meta content='Kelp Forest 3D models' name='subject'/>
    <meta content='All content has permissions for free use. Please provide credit to the Naval Postgraduate School (NPS) Modeling Virtual Envi
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/Fog-KelpForestMain.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Fog color='.8 .8 .8' fogType='LINEAR' visibilityRange='60'/>
    <WorldInfo info='"kelpForestMain3.4.wrl" "Model of the Monterey Bay Aquarium Kelp Forest" "DTG of last updated: 111500Jun98" "Added: " "Upda
    <NavigationInfo speed='3.0'/>
    <ExternProtoDeclare>
    <ProtoInstance>
    <ExternProtoDeclare>
    <Group DEF='Viewpoints'>
      <Viewpoint description='Kelp Forest Monterey Bay Aquarium' position='0.0 -5.0 45.0'/>
      <Viewpoint description='Side Windows' fieldOfView='0.5' orientation='0.0 1.0 0.0 0.9' position='25.0 -4.0 15.0'/>
      <Viewpoint description='Star Fish' orientation='0.0 1.0 0.0 1.25' position='14.0 -5.75 1.0'/>
      <Viewpoint description='Inside Tank' orientation='0.0 1.0 0.0 3.14' position='0.0 -5.0 -7.0'/>
      <Viewpoint description='Top floor' fieldOfView='1.04719' orientation='-0.3 1.0 0.3 1.57079' position='12.5 5.0 -2.0'/>
      <Viewpoint description='Bird&apos;s Eye View' orientation='1.0 0.0 0.0 -1.57' position='0.0 19.0 -3.5'/>
      <Viewpoint description='Pump House' fieldOfView='0.6' orientation='0.0 1.0 0.0 1.3' position='-5.0 1.0 0.0'/>
    </Group>
    <Group DEF='Root'>
      <Background skyColor='.8 .8 .8'/>
      <Inline url='"IntroductionMessage.wrl" "htt
      <Collision enabled='false'>
        <Transform>
          <Inline url='"SharkLeftyLocale.wrl" "ht
        </Transform>
        <Transform>
          <Inline url='"SharkLucyLocale.wrl" "htt
        </Transform>
        <Transform translation='0 0 0'>
```

© 2007, Daly Realism and Don Brutzman

X3D: Extensible 3D Graphics for Web Authors

x3dGraphics.com

Insert Fog

containerField □ children

DEF ○

USE ○

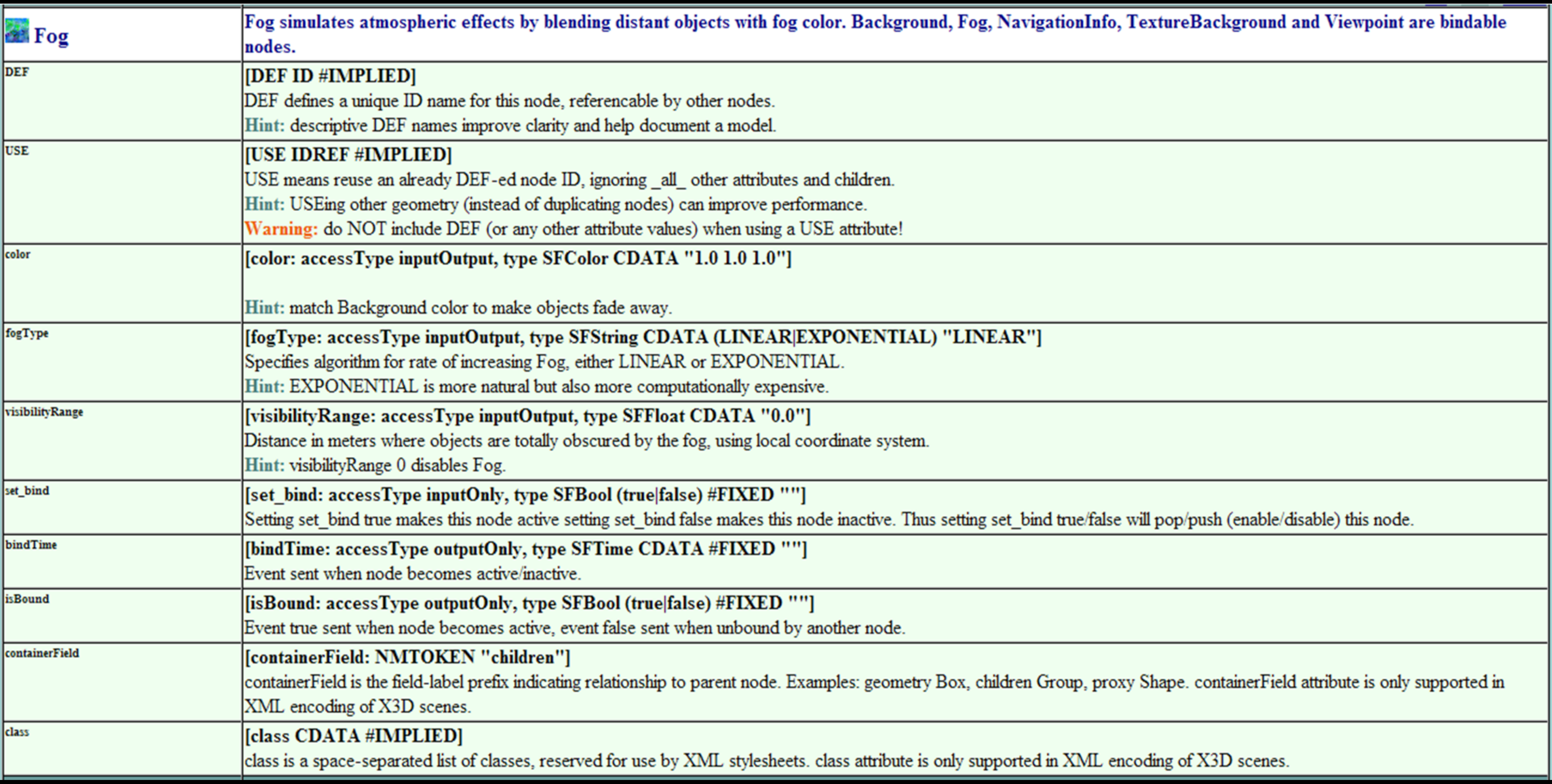
color .8 .8 .8

fogType LINEAR

visibilityRange 60

New line ☑ before □ after

20:9 INS



| Fog | Fog simulates atmospheric effects by blending distant objects with fog color. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes. |
|---|---|
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>Hint: descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.<br>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute! |
| color | **[color: accessType inputOutput, type SFColor CDATA "1.0 1.0 1.0"]**<br>Hint: match Background color to make objects fade away. |
| fogType | **[fogType: accessType inputOutput, type SFString CDATA (LINEAR\|EXPONENTIAL) "LINEAR"]**<br>Specifies algorithm for rate of increasing Fog, either LINEAR or EXPONENTIAL.<br>Hint: EXPONENTIAL is more natural but also more computationally expensive. |
| visibilityRange | **[visibilityRange: accessType inputOutput, type SFFloat CDATA "0.0"]**<br>Distance in meters where objects are totally obscured by the fog, using local coordinate system.<br>Hint: visibilityRange 0 disables Fog. |
| set_bind | **[set_bind: accessType inputOnly, type SFBool (true\|false) #FIXED ""]**<br>Setting set_bind true makes this node active setting set_bind false makes this node inactive. Thus setting set_bind true/false will pop/push (enable/disable) this node. |
| bindTime | **[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]**<br>Event sent when node becomes active/inactive. |
| isBound | **[isBound: accessType outputOnly, type SFBool (true\|false) #FIXED ""]**<br>Event true sent when node becomes active, event false sent when unbound by another node. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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

# Chapter Summary

# Chapter Summary

Lighting concepts

- Illuminating scene geometry with virtual lights

Lighting nodes

- DirectionalLight
- NavigationInfo headlight
- PointLight, Spotlight

Environment concepts

- X3DBackgroundNode type and common fields

Environmental effects nodes

- Background, TextureBackground, Fog

# Suggested exercises

Use a TouchSensor to turn a light on/off

- Include BooleanToggle so that state is persistent

Also indicate the physical location of a light source by using a Cone or Sphere

- Be sure to set *emissiveColor* and *transparency*

Demonstrate use of Background colors

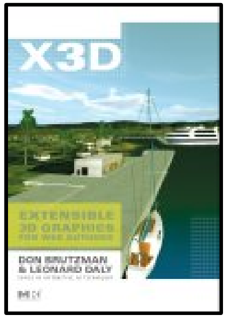
Demonstrate use of (Texture)Background images

Demonstrate use of Fog for day or night effects

# References

# References 1

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


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

- Chapter 11, Lighting and Environment Nodes
- http://x3dGraphics.com
- http://x3dgraphics.com/examples/X3dForWebAuthors

X3D Resources

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

# References 2

X3D-Edit Authoring Tool

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

X3D Scene Authoring Hints

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

X3D Graphics Specification

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

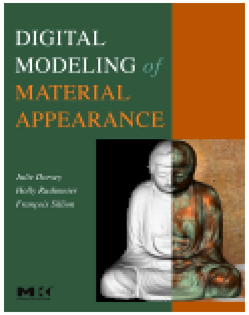
# References 3

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

- http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm
- http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook
- Chapter 20 – Lighting
- Chapter 22 - Background
- Chapter 23 - Fog

# References 4

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

RGB World explains additive, subtractive color

- http://www.rgbworld.com/color.html

# 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

web3D CONSORTIUM

<X3D>

# 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! ☺

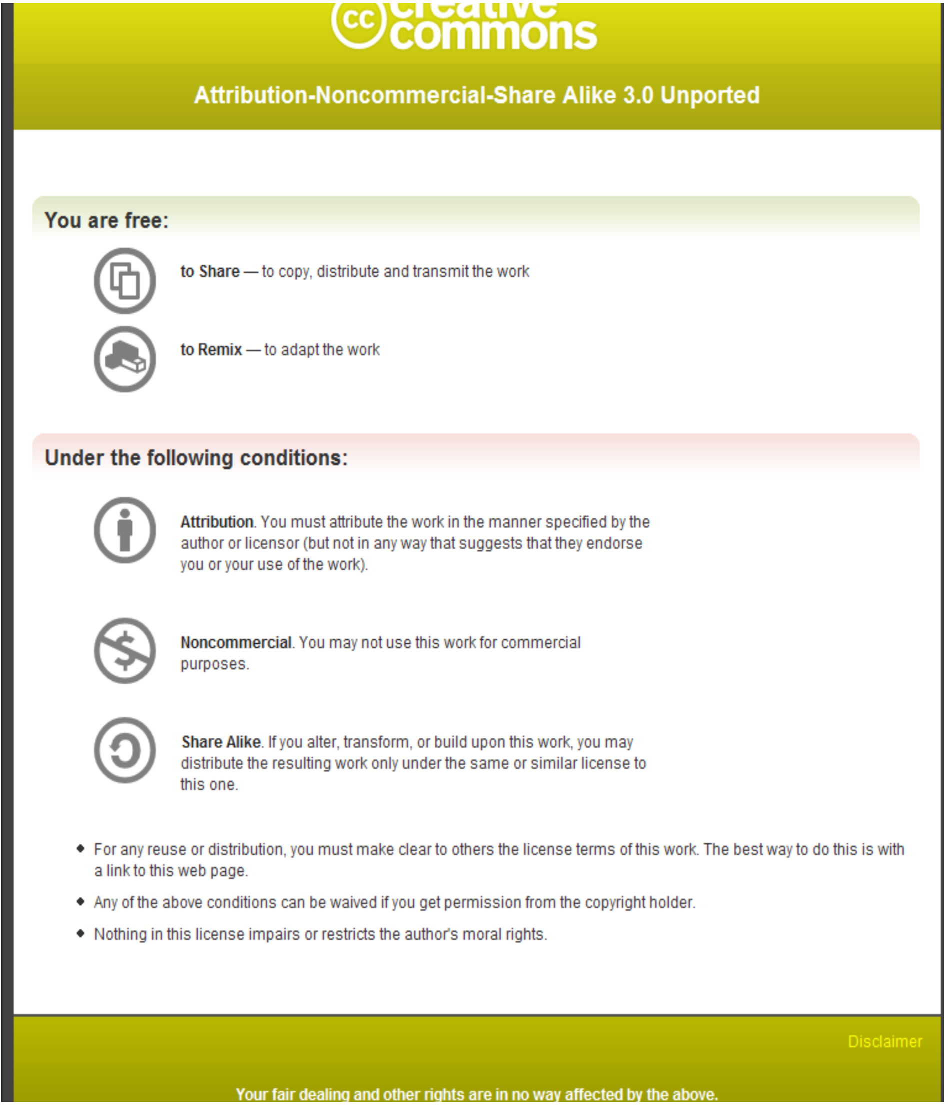
- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics

cgems ACMSIGGRAPH EG

# Creative Commons open-source license

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



Attribution-Noncommercial-Share Alike 3.0 Unported

**You are free:**

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

**to Remix** — to adapt the work

**Under the following conditions:**

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

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

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

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

Disclaimer

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

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

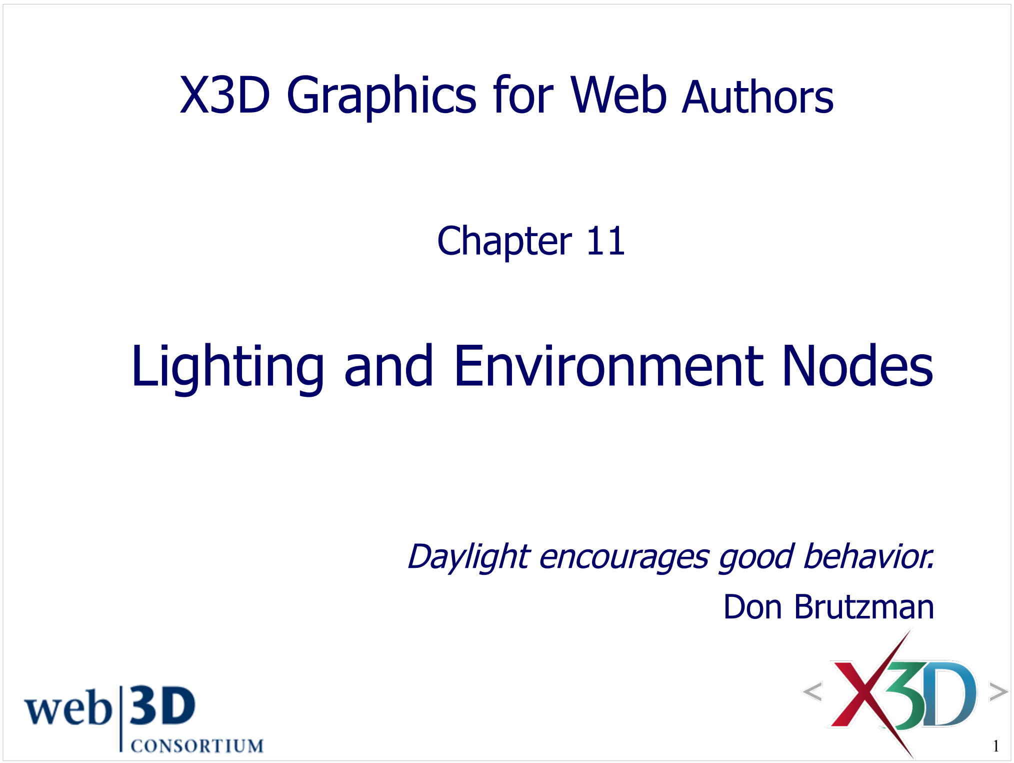
http://www.web3d.org/x3d/content/examples/license.html

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

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

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

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



X3D Graphics for Web Authors

Chapter 11

# Lighting and Environment Nodes

*Daylight encourages good behavior.*
Don Brutzman

web|3D CONSORTIUM

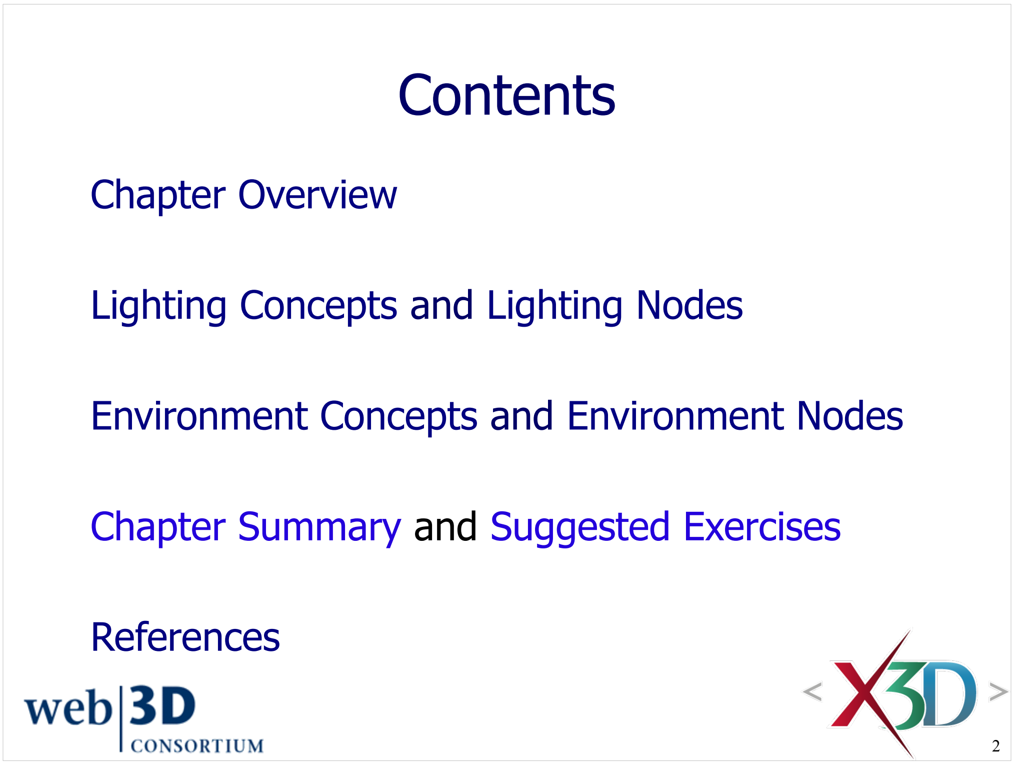
1

Alternate quote:

*Fiat lux*, let there be light!
Book of Genesis, Old Testament, Bible
http://en.wikipedia.org/wiki/Let_there_be_light

Of further interest:

Graphica Obscura by Paul Haeberle
http://www.graficaobscura.com



# Contents

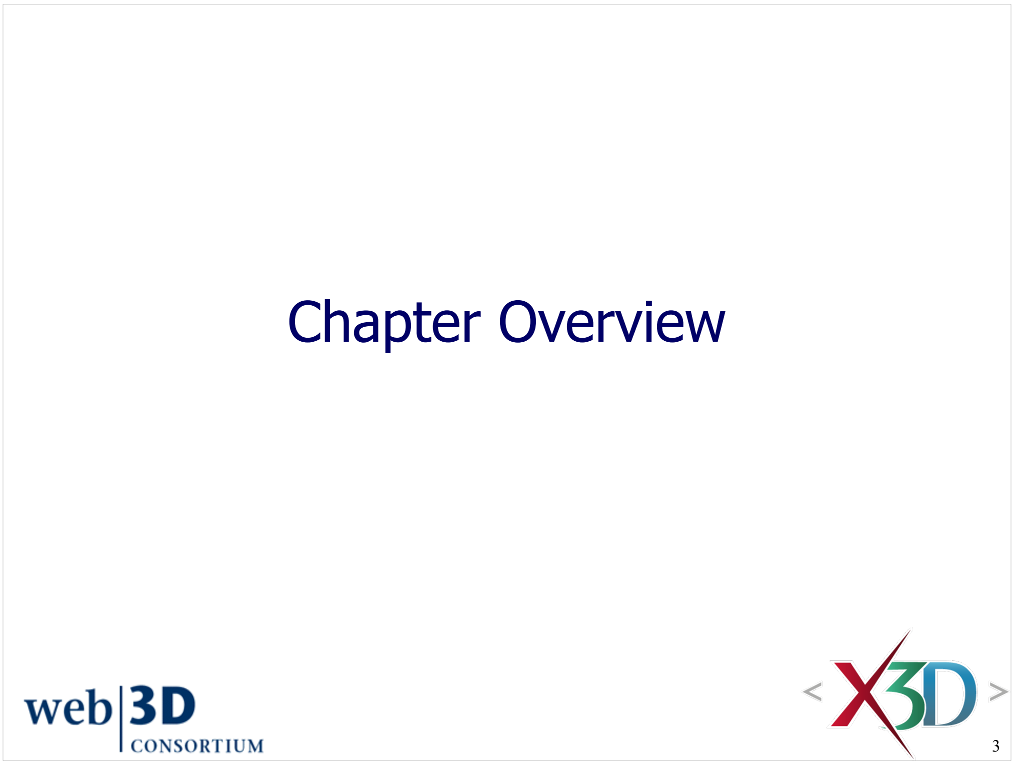
Chapter Overview

Lighting Concepts and Lighting Nodes

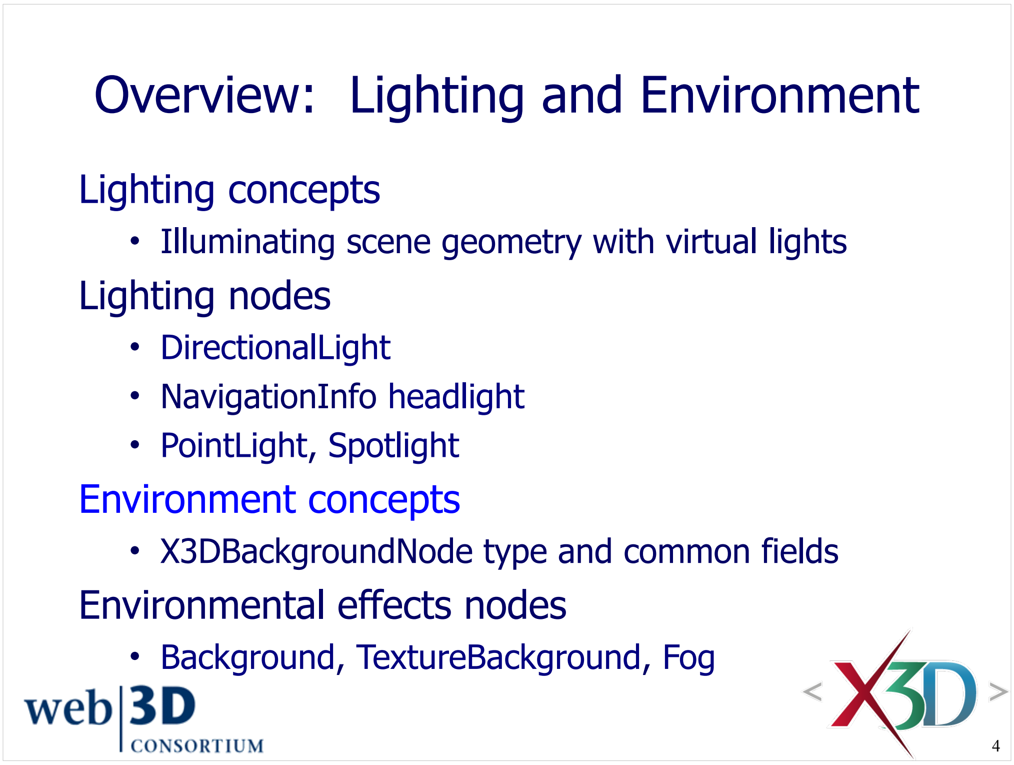
Environment Concepts and Environment Nodes

Chapter Summary and Suggested Exercises

References



# Chapter Overview



## Overview: Lighting and Environment

Lighting concepts

- Illuminating scene geometry with virtual lights

Lighting nodes

- DirectionalLight
- NavigationInfo headlight
- PointLight, Spotlight

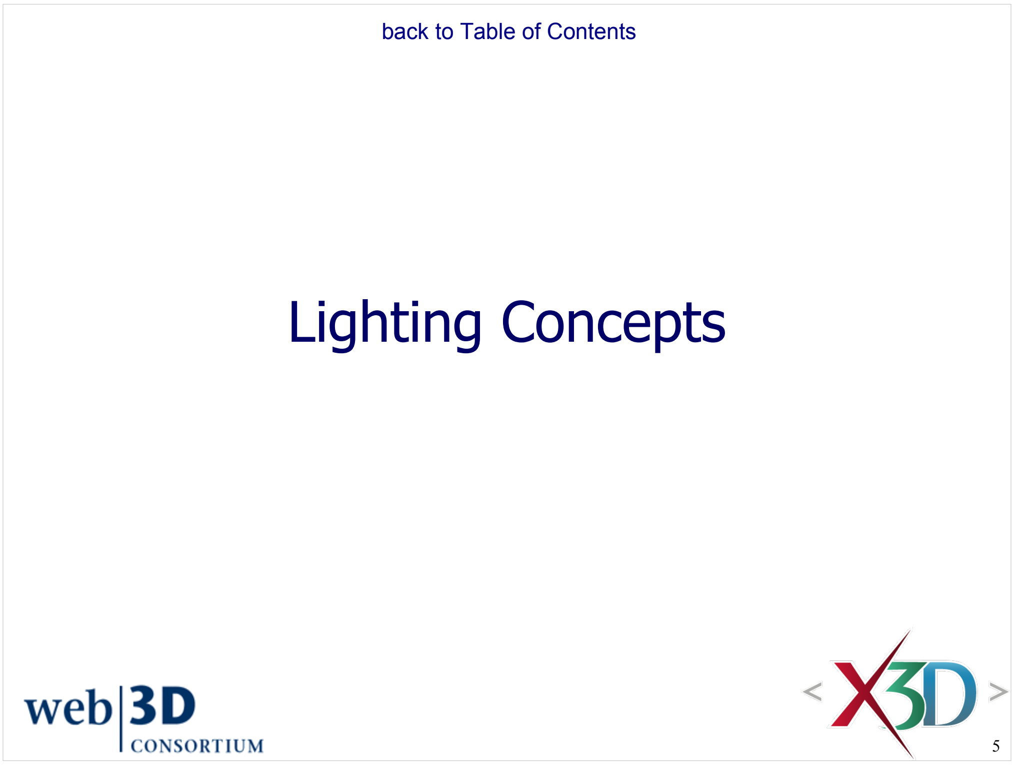
Environment concepts

- X3DBackgroundNode type and common fields

Environmental effects nodes

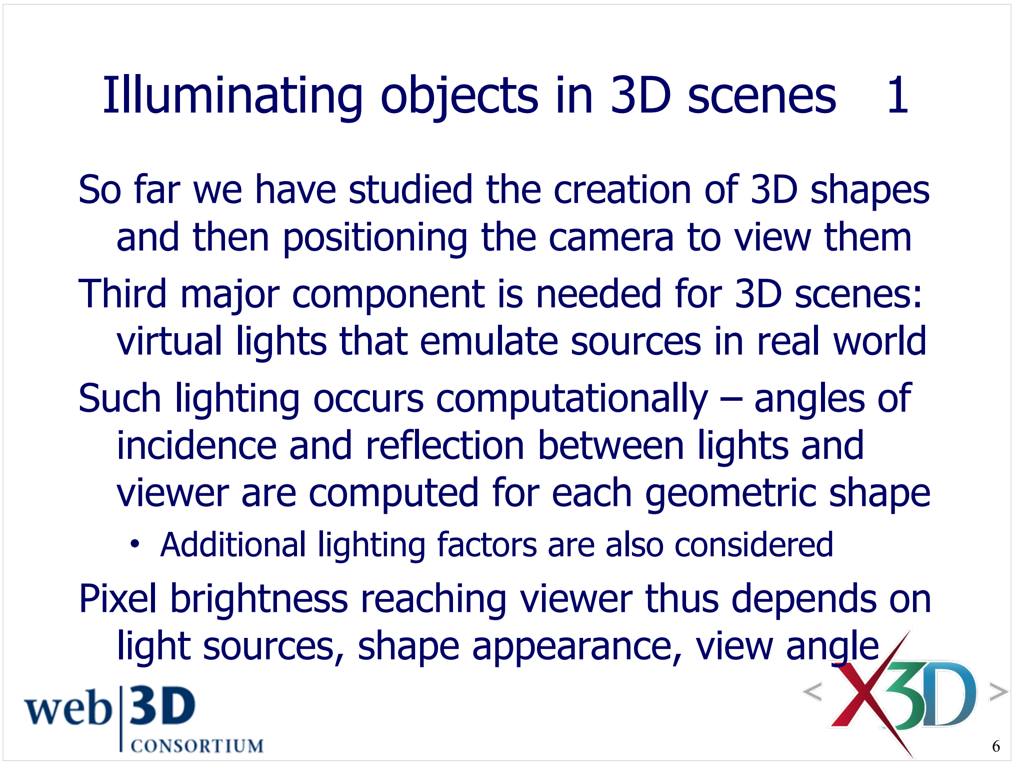
- Background, TextureBackground, Fog

4



back to Table of Contents

# Lighting Concepts



## Illuminating objects in 3D scenes 1

So far we have studied the creation of 3D shapes and then positioning the camera to view them

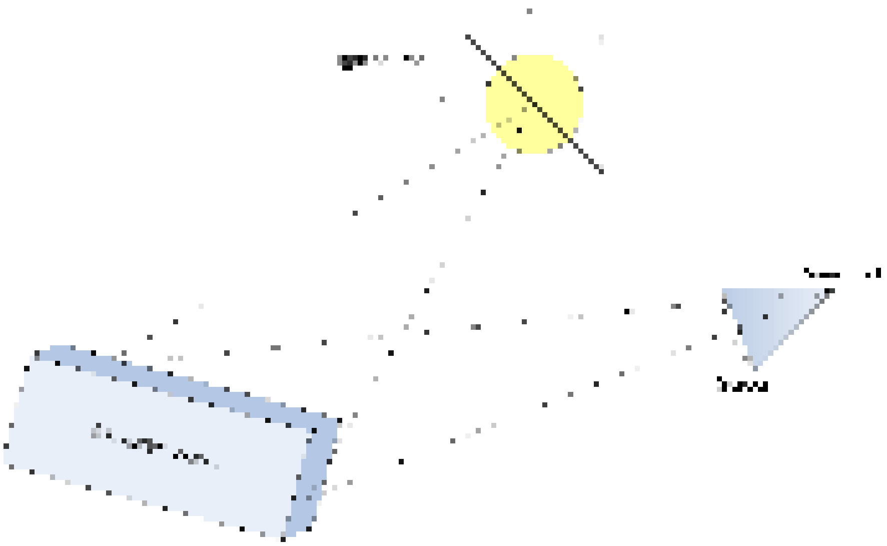
Third major component is needed for 3D scenes: virtual lights that emulate sources in real world

Such lighting occurs computationally – angles of incidence and reflection between lights and viewer are computed for each geometric shape

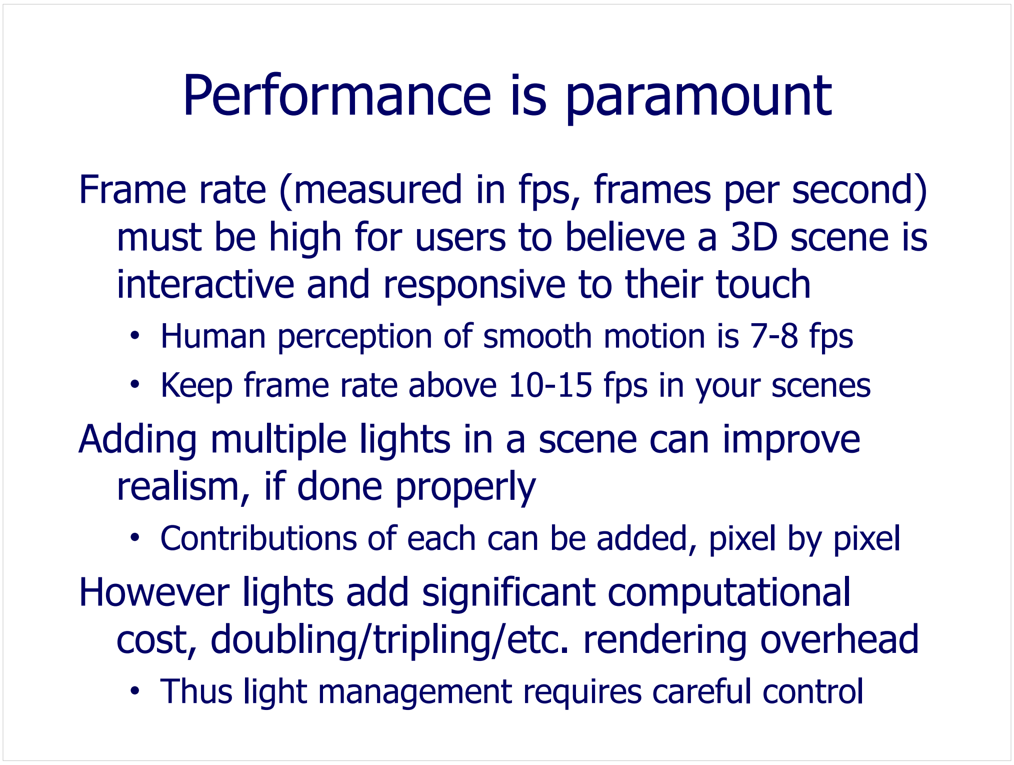
- Additional lighting factors are also considered

Pixel brightness reaching viewer thus depends on light sources, shape appearance, view angle

# Illuminating objects in 3D scenes 2



Page 301, Figure 11.1. X3D lighting and rendering is made up of a light source, rays, geometric object surfaces, reflection, and the screen rendering that displays a user’s point of view (POV).



## Performance is paramount

Frame rate (measured in fps, frames per second) must be high for users to believe a 3D scene is interactive and responsive to their touch

- Human perception of smooth motion is 7-8 fps
- Keep frame rate above 10-15 fps in your scenes

Adding multiple lights in a scene can improve realism, if done properly

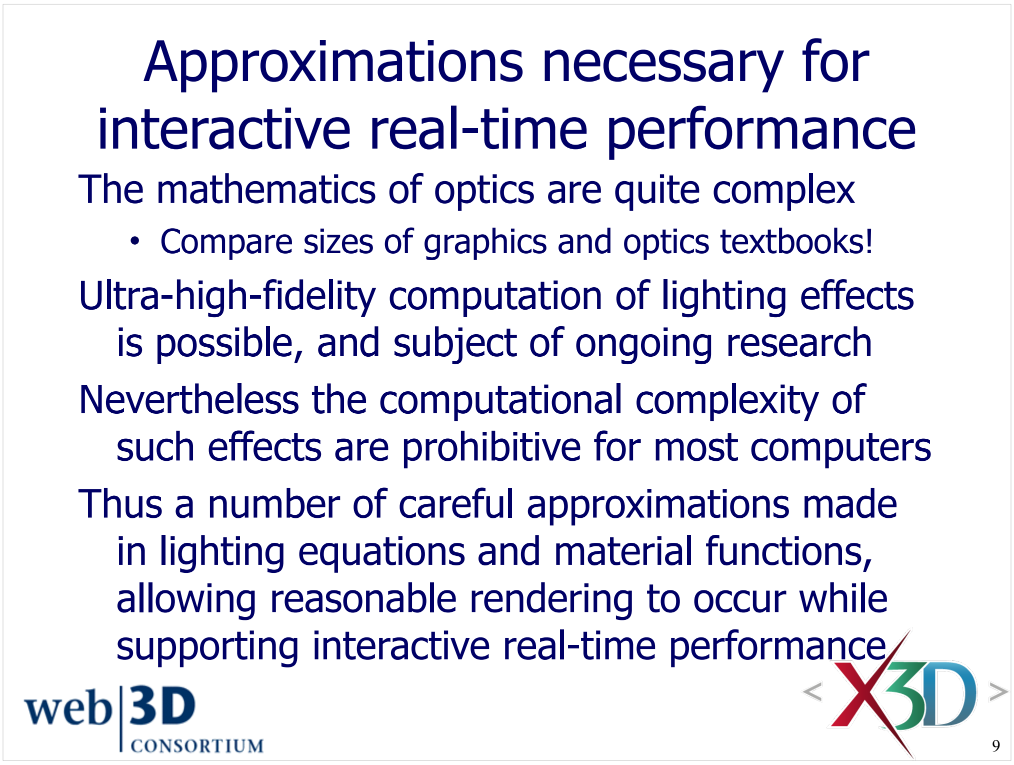
- Contributions of each can be added, pixel by pixel

However lights add significant computational cost, doubling/tripling/etc. rendering overhead

- Thus light management requires careful control

Each added light with global scope adds the same amount of computational complexity as the first light, thus making the addition of lights one of the easiest ways to change an interactive scene into something slow and unresponsive.

Keeping PointLight and SpotLight definitions scoped to only affect local peer and child nodes (default *global*='false') is a good way to easily manage multiple lights.  This approach ensures that local lights (such as individual lights for individual rooms) don't accumulate unnecessarily.



## Approximations necessary for interactive real-time performance

The mathematics of optics are quite complex

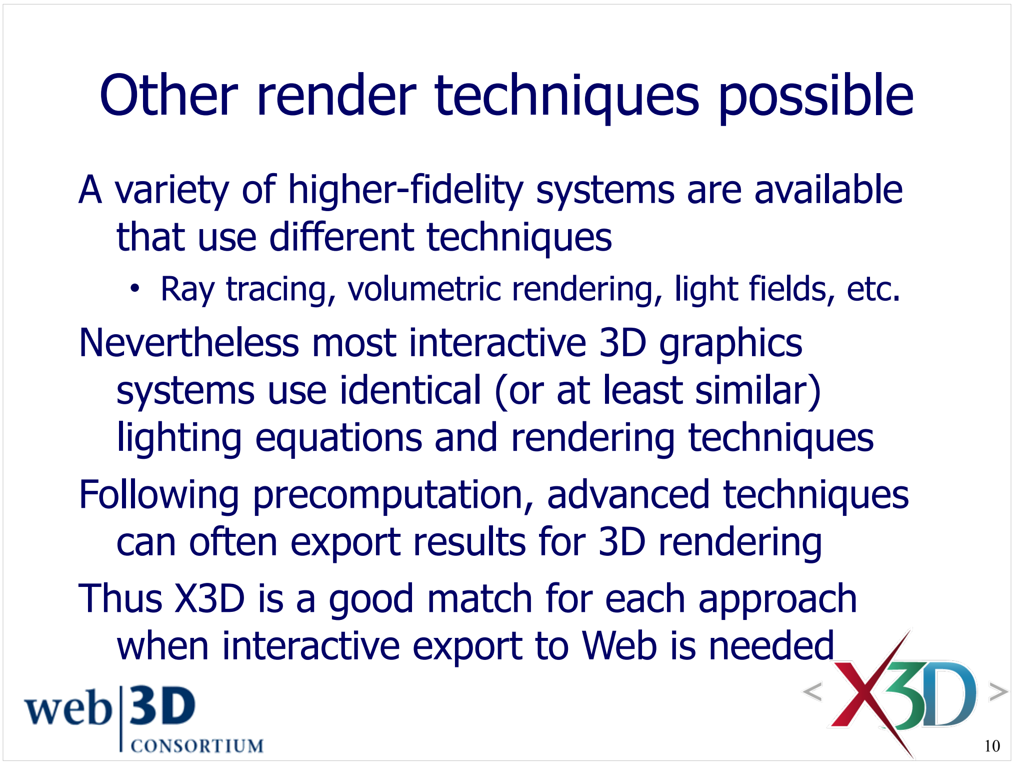
- Compare sizes of graphics and optics textbooks!

Ultra-high-fidelity computation of lighting effects is possible, and subject of ongoing research

Nevertheless the computational complexity of such effects are prohibitive for most computers

Thus a number of careful approximations made in lighting equations and material functions, allowing reasonable rendering to occur while supporting interactive real-time performance

The full lighting equations are included in the X3D Abstract Specification. However these are rarely (if ever) needed by authors, so we don't study them here.



## Other render techniques possible

A variety of higher-fidelity systems are available that use different techniques

- Ray tracing, volumetric rendering, light fields, etc.

Nevertheless most interactive 3D graphics systems use identical (or at least similar) lighting equations and rendering techniques

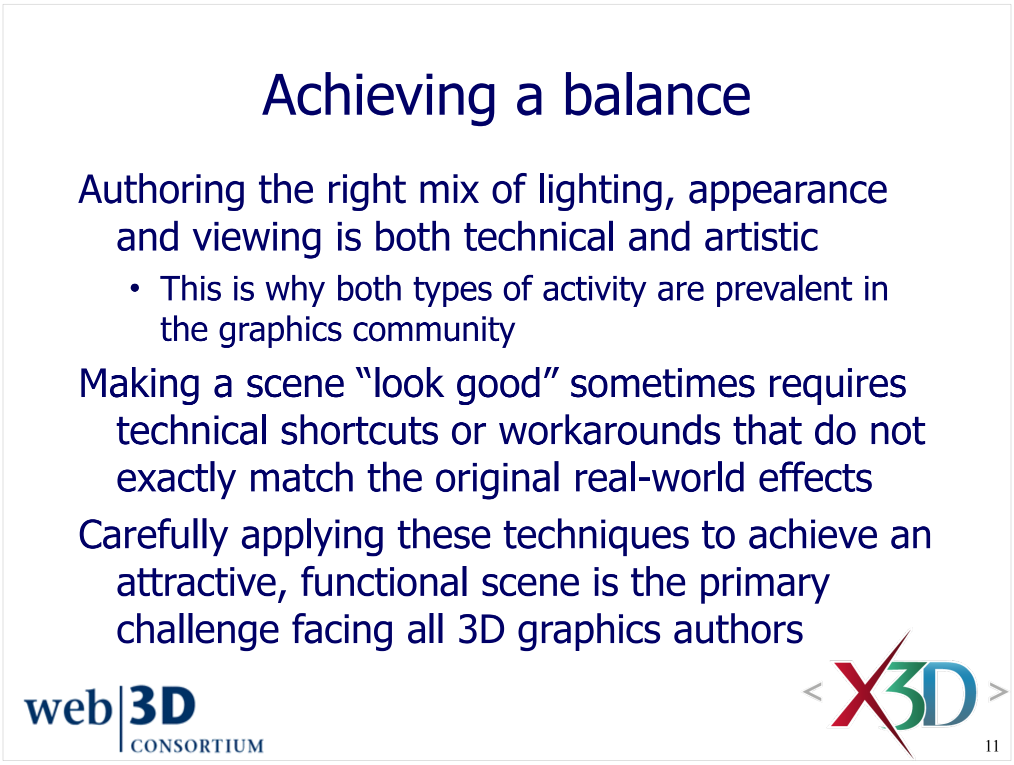
Following precomputation, advanced techniques can often export results for 3D rendering

Thus X3D is a good match for each approach when interactive export to Web is needed

10

Individual modeling tools can be checked for export capabilities.

X3D conversion and translation tools are another good option for adapting the results of other rendering techniques.  A list of such tools is maintained on the X3D Help page at http://www.web3d.org/x3d/content/examples/help.html#Conversions



## Achieving a balance

Authoring the right mix of lighting, appearance and viewing is both technical and artistic

- This is why both types of activity are prevalent in the graphics community

Making a scene “look good” sometimes requires technical shortcuts or workarounds that do not exactly match the original real-world effects

Carefully applying these techniques to achieve an attractive, functional scene is the primary challenge facing all 3D graphics authors

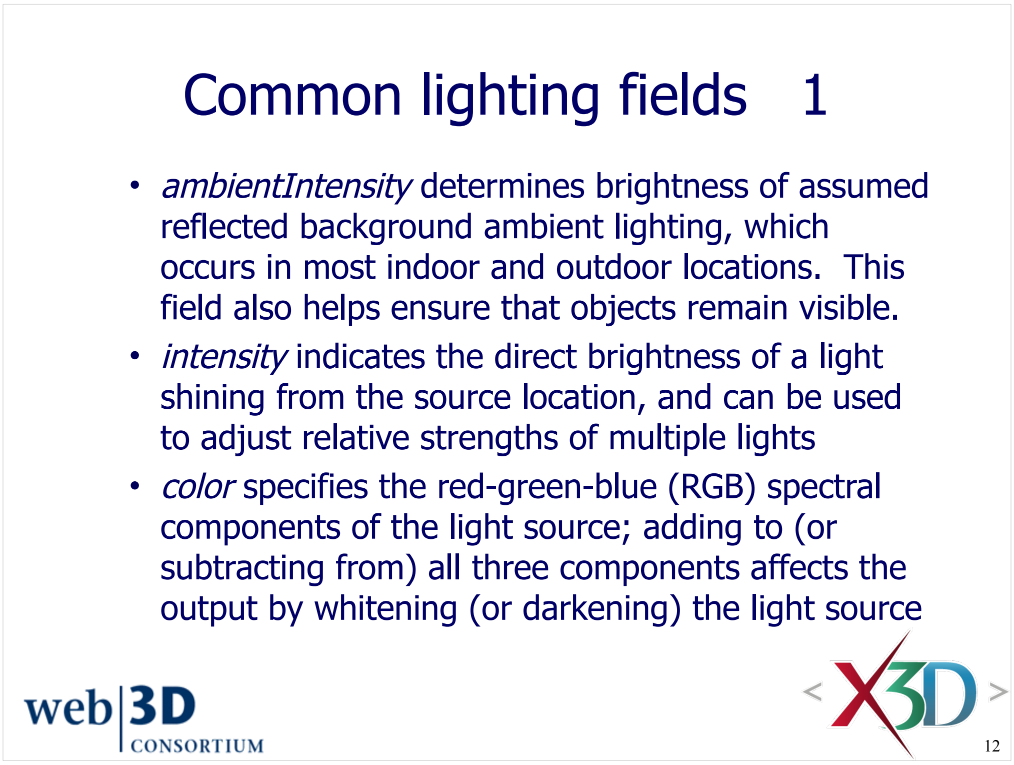
web3D CONSORTIUM

<X3D>

11

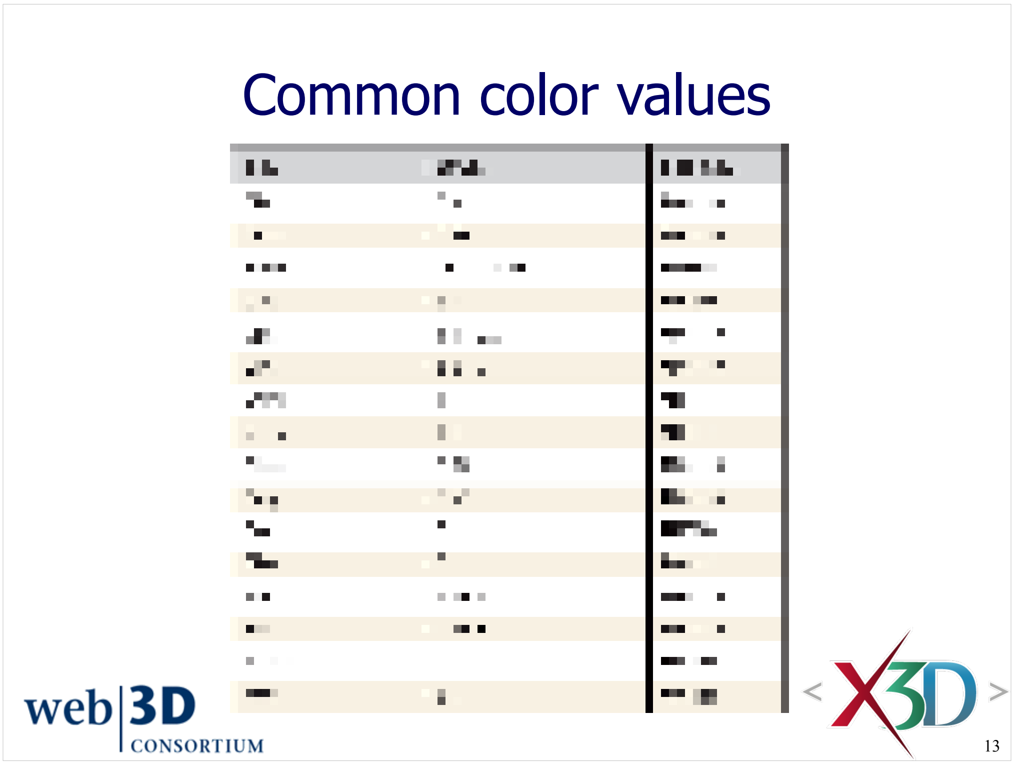
Often a scene which might be sluggish can be optimized and frame rate improved.

Over time such scenes are likely to run even better as graphics hardware and software continues to improve rapidly.



# Common lighting fields 1

- *ambientIntensity* determines brightness of assumed reflected background ambient lighting, which occurs in most indoor and outdoor locations. This field also helps ensure that objects remain visible.
- *intensity* indicates the direct brightness of a light shining from the source location, and can be used to adjust relative strengths of multiple lights
- *color* specifies the red-green-blue (RGB) spectral components of the light source; adding to (or subtracting from) all three components affects the output by whitening (or darkening) the light source



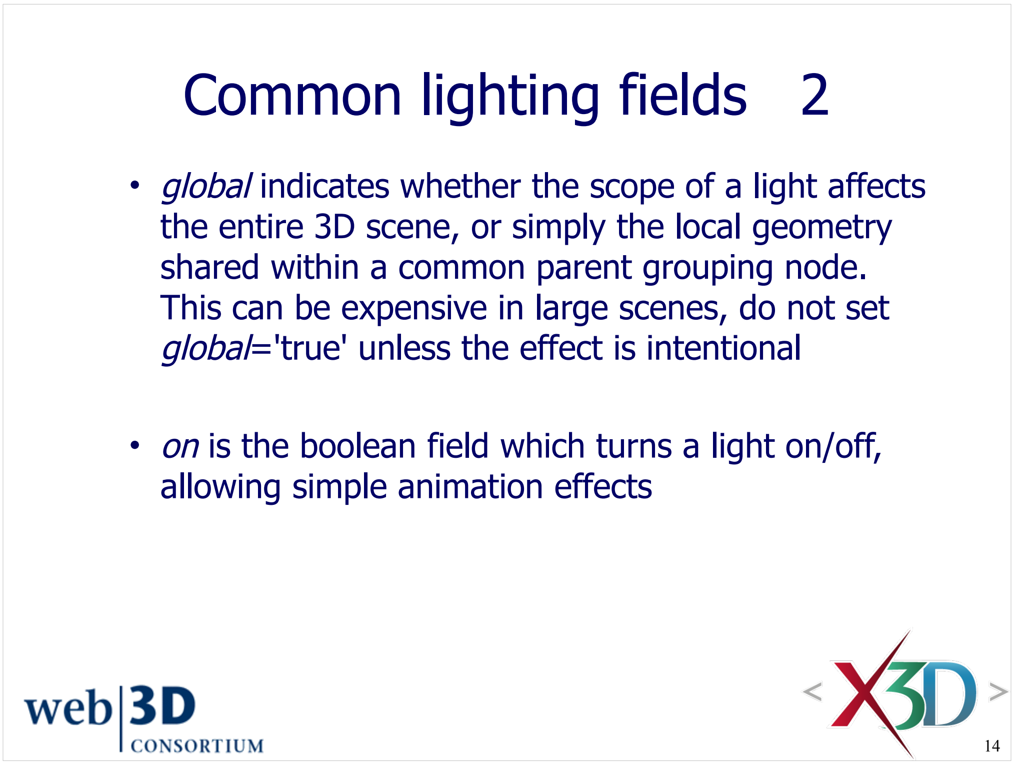
## Common color values

| [illegible] | [illegible] | [illegible] |
|---|---|---|
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |

13

Page 305, Table 11.2. Common Colors and Corresponding RGB and HTML Values

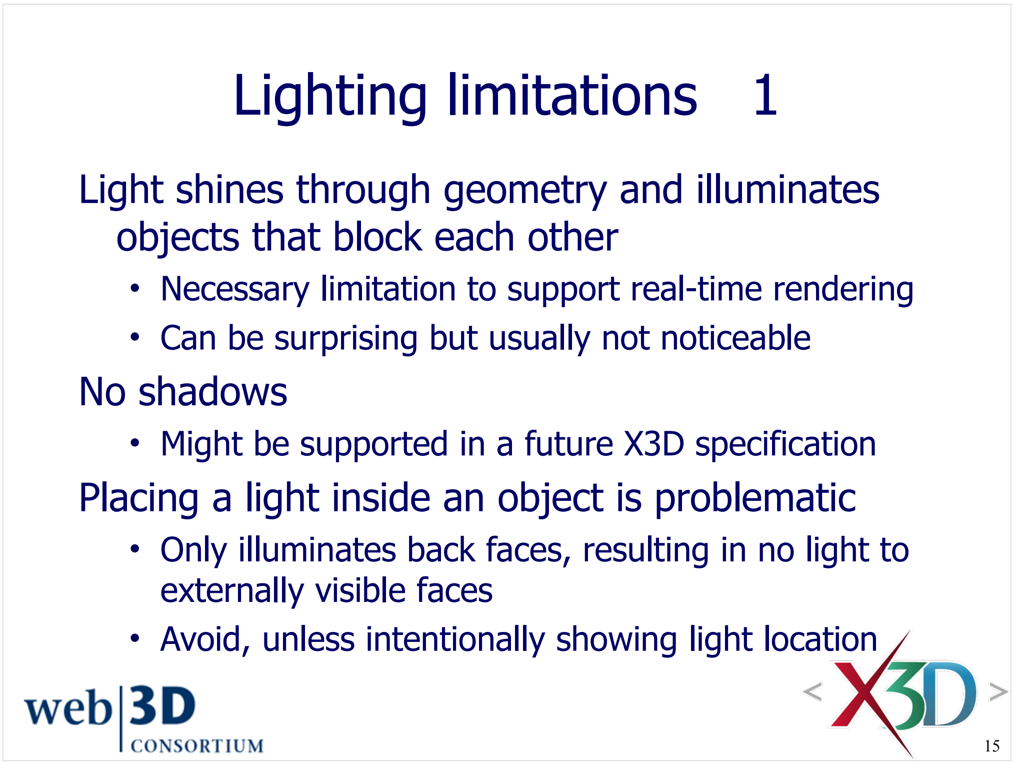
It is worth noting that lights are typically white, thus letting colors depend on the appearance of each shape.



# Common lighting fields 2

- *global* indicates whether the scope of a light affects the entire 3D scene, or simply the local geometry shared within a common parent grouping node. This can be expensive in large scenes, do not set *global*='true' unless the effect is intentional

- *on* is the boolean field which turns a light on/off, allowing simple animation effects



## Lighting limitations 1

Light shines through geometry and illuminates objects that block each other

- Necessary limitation to support real-time rendering
- Can be surprising but usually not noticeable

No shadows

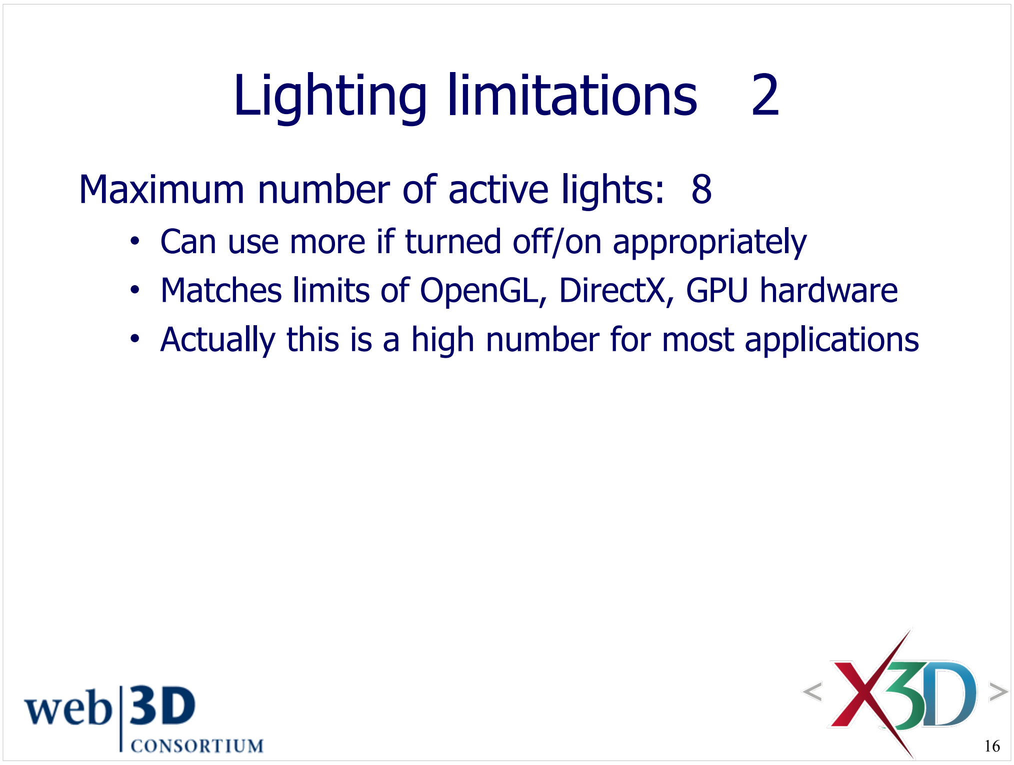
- Might be supported in a future X3D specification

Placing a light inside an object is problematic

- Only illuminates back faces, resulting in no light to externally visible faces
- Avoid, unless intentionally showing light location

Shadows proposal

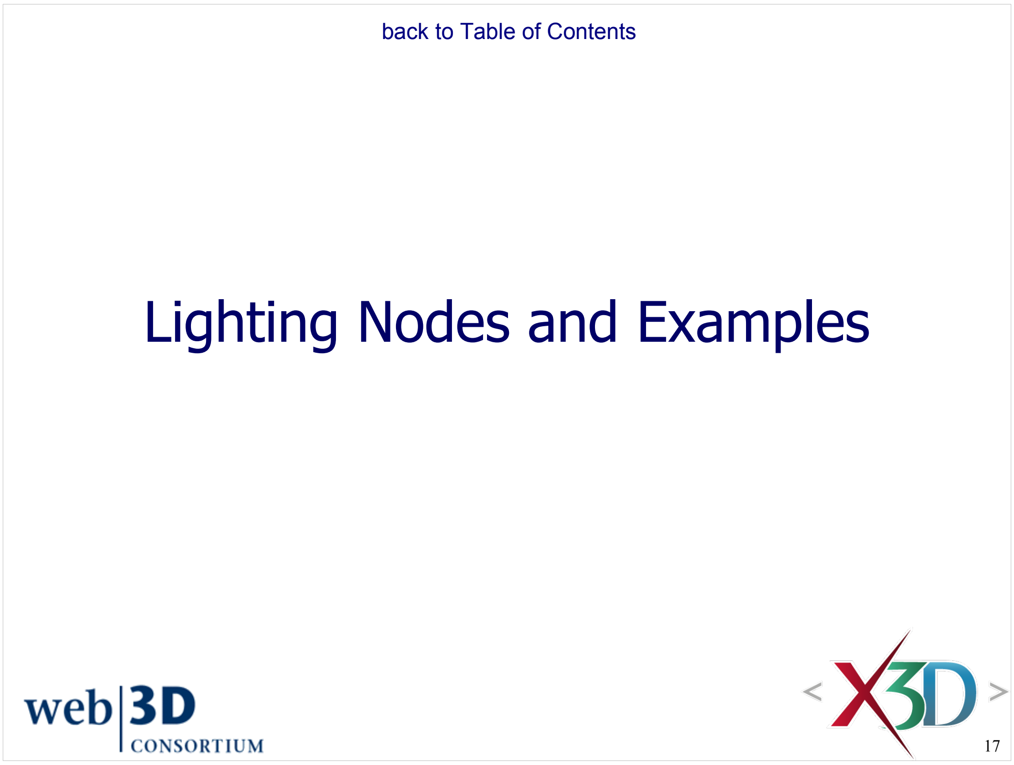
- Yvonne Jung, Tobias Franke, Patrick D¨ahne, and Johannes Behr, “Enhancing X3D for advanced MR appliances,” *Twelfth ACM SIGGRAPH International Symposium on Web3D Graphics,* Peruggia Italy, 15-18 April 2007, pp. 27-36 and 207.
- InstantReality: http://www.instantreality.org/tutorial/dynamic-shadows
- BS Contact: http://www.bitmanagement.com/demos/dynamic_shadow.en.html
- Octaga: http://www.octaga.com then search for 'shadows'



# Lighting limitations 2

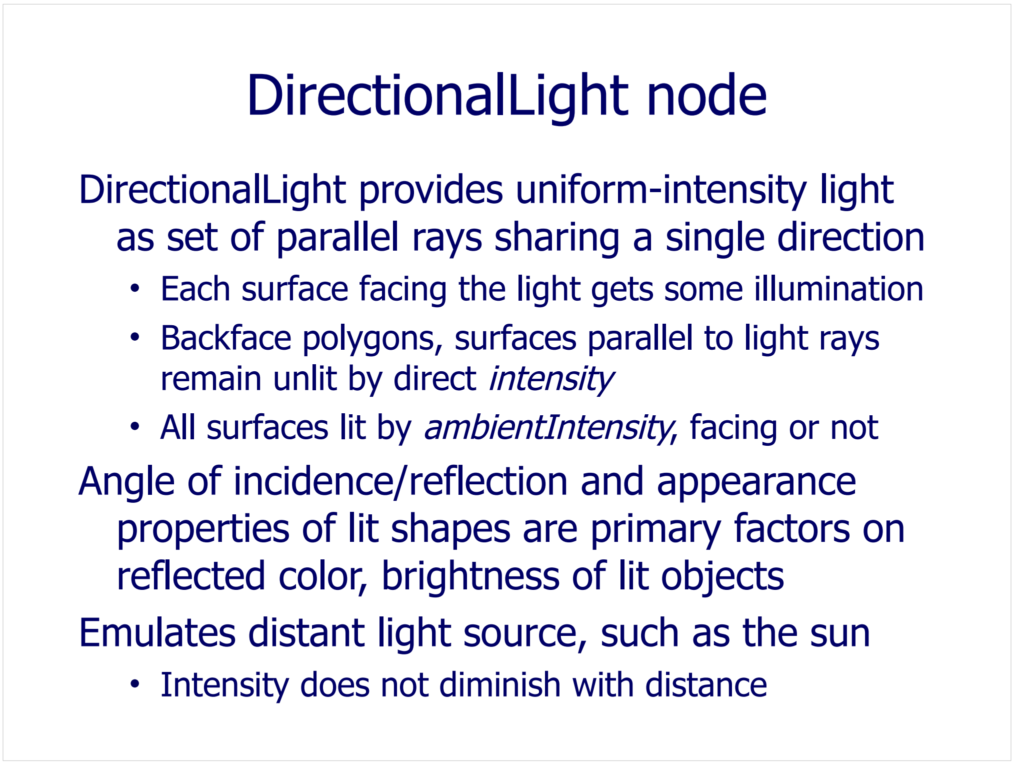
Maximum number of active lights: 8

- Can use more if turned off/on appropriately
- Matches limits of OpenGL, DirectX, GPU hardware
- Actually this is a high number for most applications



back to Table of Contents

# Lighting Nodes and Examples



# DirectionalLight node

DirectionalLight provides uniform-intensity light as set of parallel rays sharing a single direction

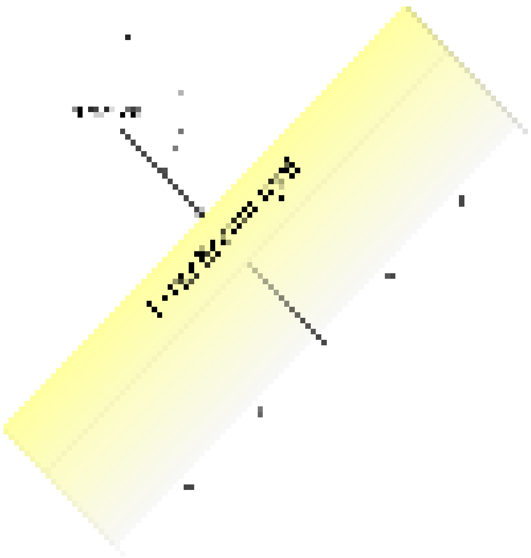
- Each surface facing the light gets some illumination
- Backface polygons, surfaces parallel to light rays remain unlit by direct *intensity*
- All surfaces lit by *ambientIntensity*, facing or not

Angle of incidence/reflection and appearance properties of lit shapes are primary factors on reflected color, brightness of lit objects

Emulates distant light source, such as the sun

- Intensity does not diminish with distance

# DirectionalLight geometry



19

Page 310, Figure 11.3. DirectionalLight provides a virtual light source where all rays are parallel.

# DirectionalLight fields

- *ambientIntensity, color, global, intensity, on* – same
- *direction* is x-y-z vector indicating direction of rays in local coordinate system, which in turn depends on rotations in parent Transform node hierarchy

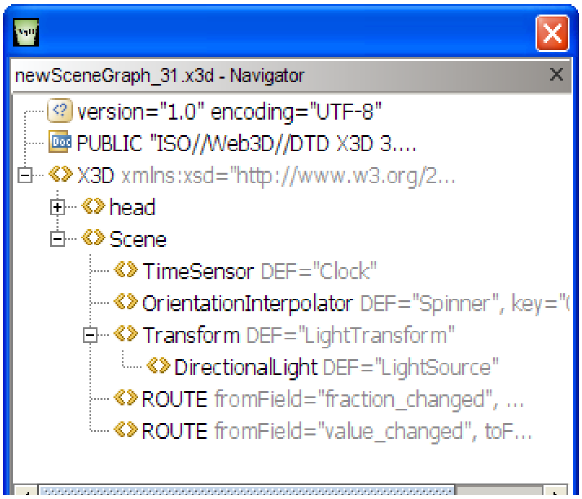
## Hints

- Misdirected DirectionalLight nodes hard to debug, since light nodes themselves are not visible
- Substitute a PointLight node to get indication of where the light actually is located
- Can rotate DirectionalLight via parent Transform, also semitransparent Cone for direction, range
- Position irrelevant since constant intensity, direction

Could animate DirectionalLight directly by using a PositionInterpolator (SFVec3f).

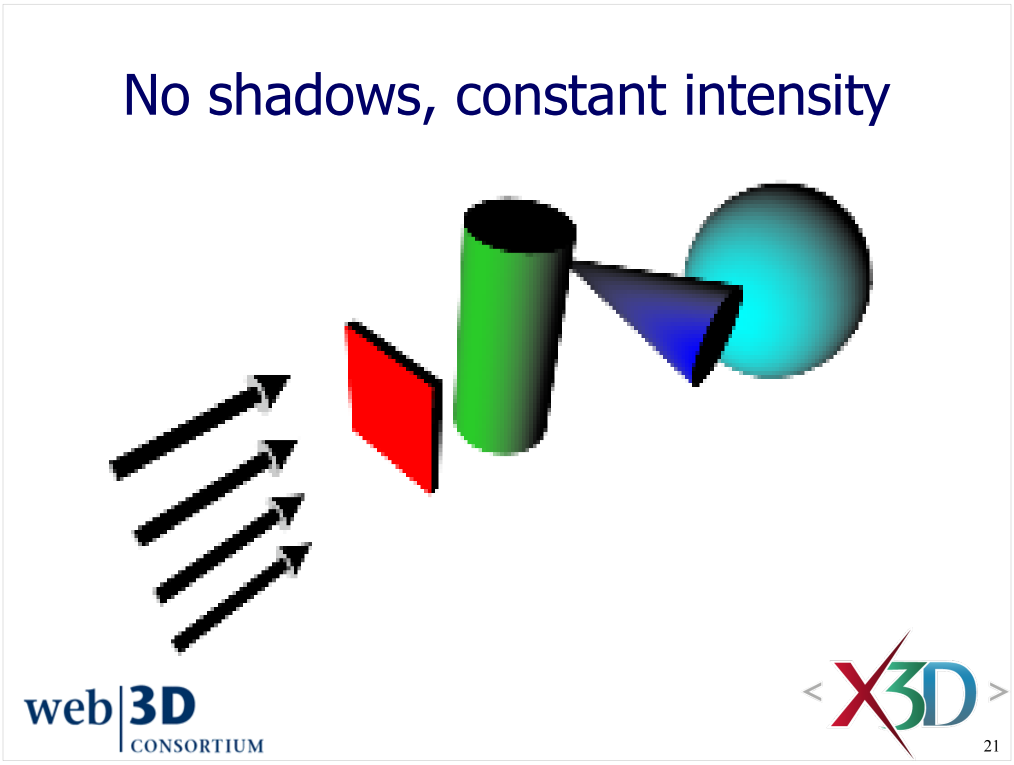
Interesting pattern for SFRotation animation of DirectionalLight: rotate parent Transform as follows.

```
<TimeSensor DEF='Clock'/>
<OrientationInterpolator DEF="Spinner" key='0 1' keyValue='0 1 0 0, 0 1 0 1.57'/>
<Transform DEF='LightTransform'>
  <DirectionalLight DEF='LightSource' />
</Transform>
<ROUTE fromField='fraction_changed' fromNode='Clock' toField='set_fraction' toNode='Spinner'/>
<ROUTE fromField='value_changed' fromNode='Spinner' toField='rotation' toNode='LightTransform'/>
```



newSceneGraph_31.x3d - Navigator

```
version="1.0" encoding="UTF-8"
PUBLIC "ISO//Web3D//DTD X3D 3....
X3D xmlns:xsd="http://www.w3.org/2...
  head
  Scene
    TimeSensor DEF="Clock"
    OrientationInterpolator DEF="Spinner", key="(
    Transform DEF="LightTransform"
      DirectionalLight DEF="LightSource"
    ROUTE fromField="fraction_changed", ...
    ROUTE fromField="value_changed", toF...
```

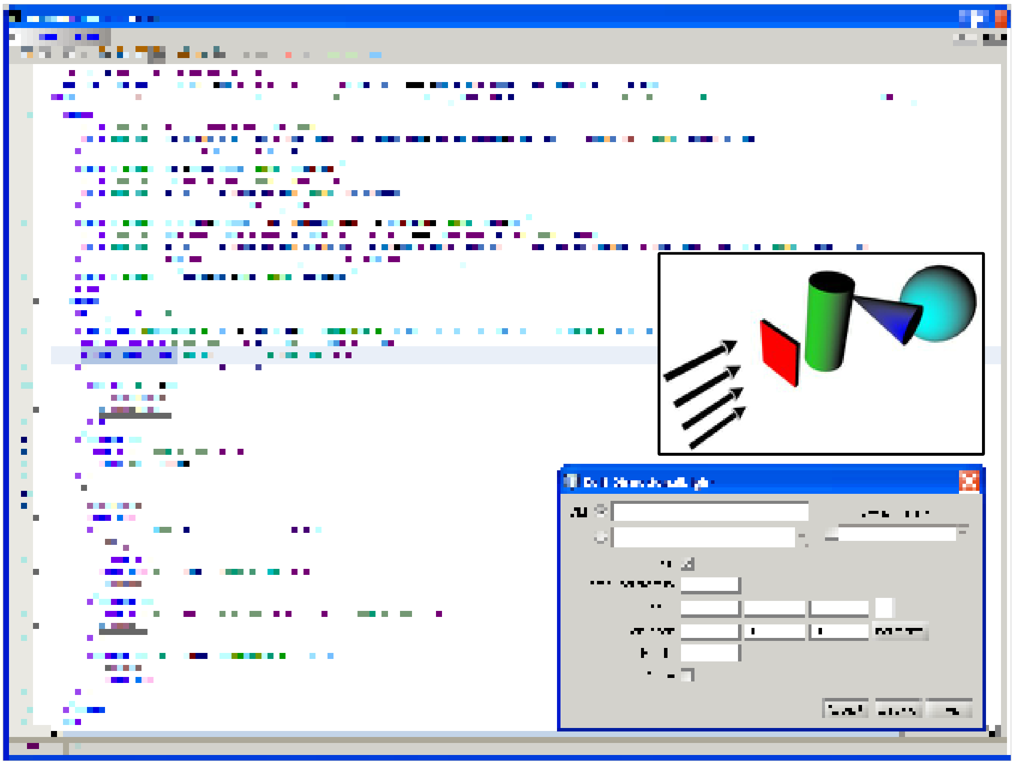


## No shadows, constant intensity

21

Page 310, Figure 11.4. DirectionalLight nodes do not interact with objects to cast shadows, and distance does not decrease their intensity.

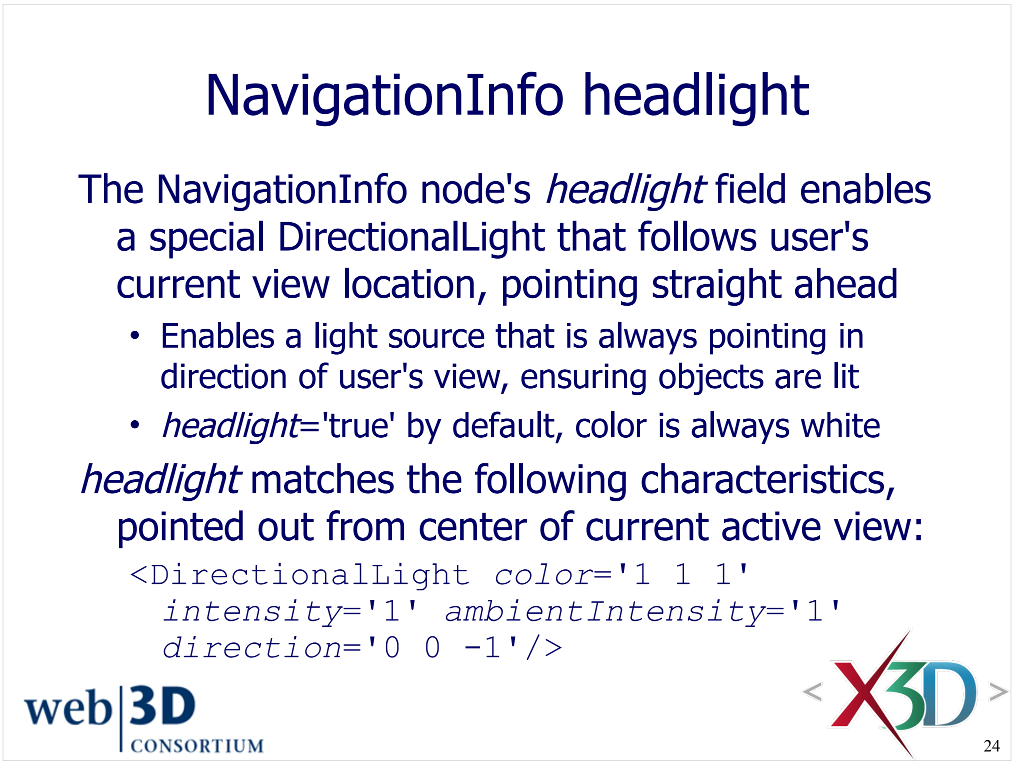
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/DirectionalLight.x3d



Page 310, Figure 11.4. DirectionalLight nodes do not interact with objects to cast shadows, and distance does not decrease their intensity.

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/DirectionalLight.x3d

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



# NavigationInfo headlight

The NavigationInfo node's *headlight* field enables a special DirectionalLight that follows user's current view location, pointing straight ahead

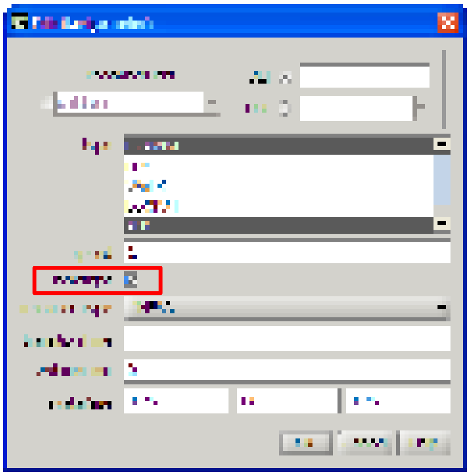
- Enables a light source that is always pointing in direction of user's view, ensuring objects are lit
- *headlight*='true' by default, color is always white

*headlight* matches the following characteristics, pointed out from center of current active view:

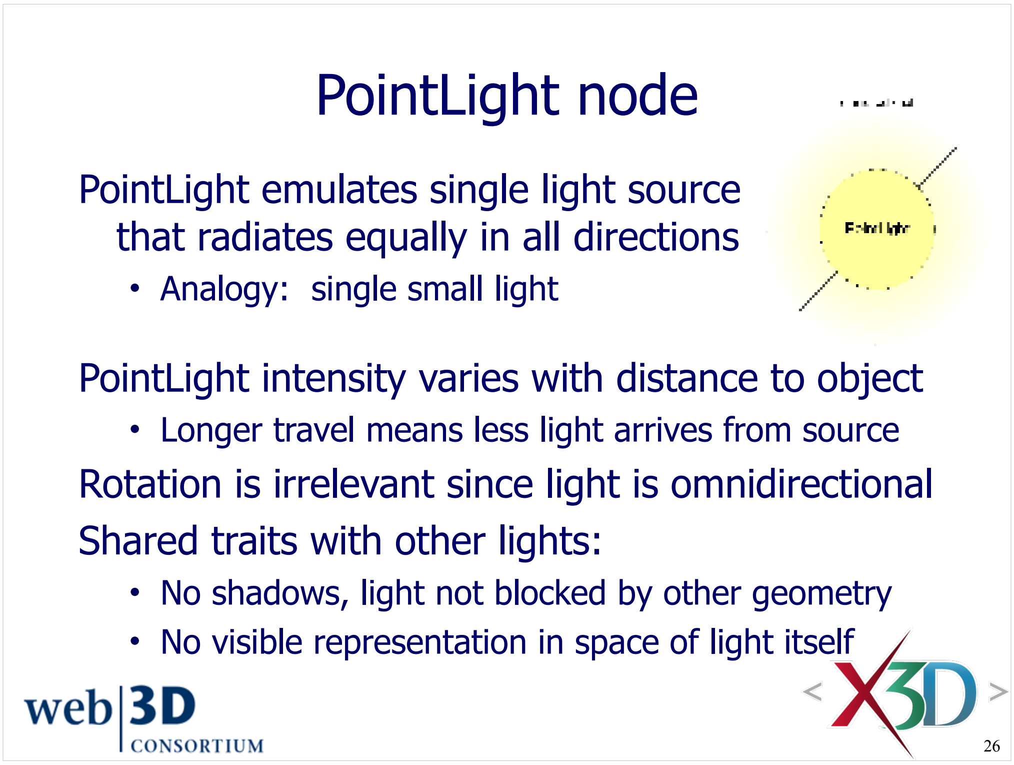
```
<DirectionalLight color='1 1 1'
  intensity='1' ambientIntensity='1'
  direction='0 0 -1'/>
```

Further details in Chapter 4, Viewing and Navigation

# NavigationInfo headlight, X3D-Edit



Further details in Chapter 4, Viewing and Navigation



## PointLight node

PointLight emulates single light source that radiates equally in all directions

- Analogy:  single small light

PointLight intensity varies with distance to object

- Longer travel means less light arrives from source

Rotation is irrelevant since light is omnidirectional

Shared traits with other lights:

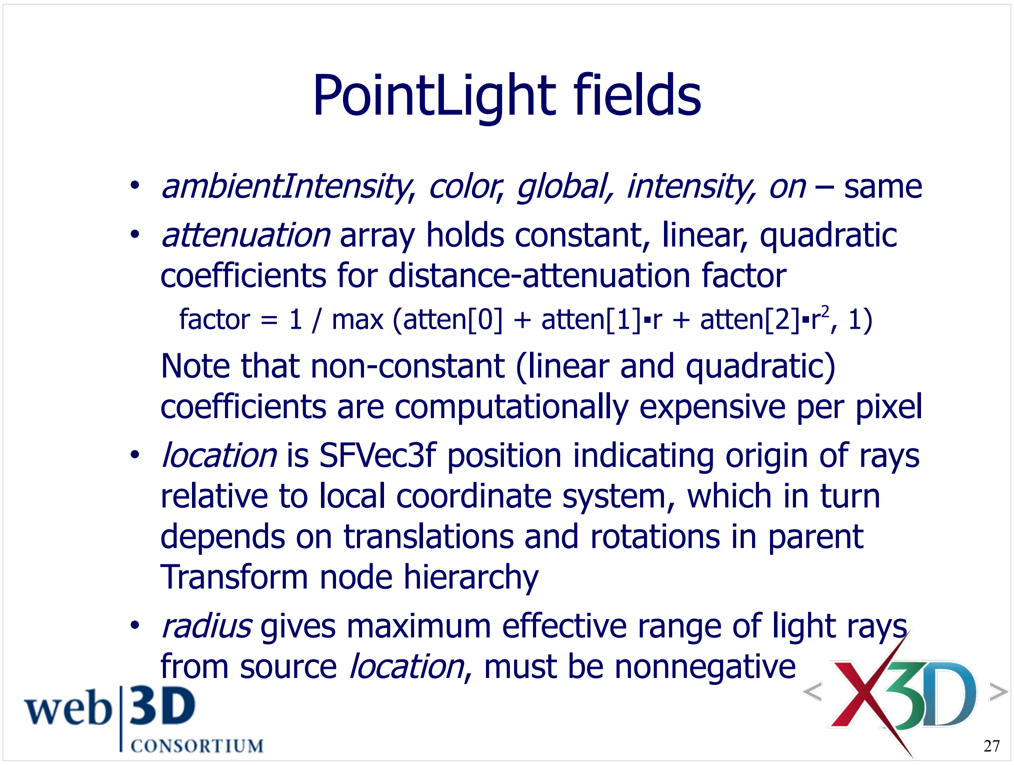
- No shadows, light not blocked by other geometry
- No visible representation in space of light itself

web3D CONSORTIUM

<X3D>

26

Page 312, Figure 11.5. PointLight provides a virtual light source such that rays emanate radially in all directions from a single location.

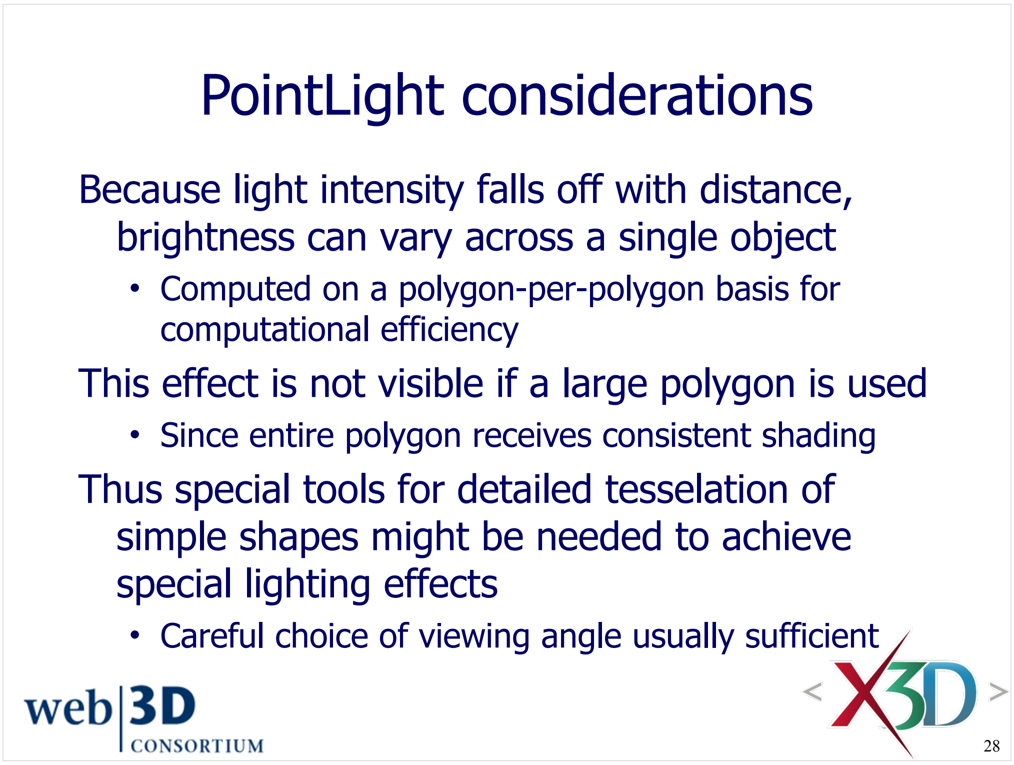


# PointLight fields

- *ambientIntensity, color, global, intensity, on* – same
- *attenuation* array holds constant, linear, quadratic coefficients for distance-attenuation factor

  factor = 1 / max (atten[0] + atten[1]▪r + atten[2]▪$r^2$, 1)

  Note that non-constant (linear and quadratic) coefficients are computationally expensive per pixel
- *location* is SFVec3f position indicating origin of rays relative to local coordinate system, which in turn depends on translations and rotations in parent Transform node hierarchy
- *radius* gives maximum effective range of light rays from source *location*, must be nonnegative



# PointLight considerations

Because light intensity falls off with distance, brightness can vary across a single object

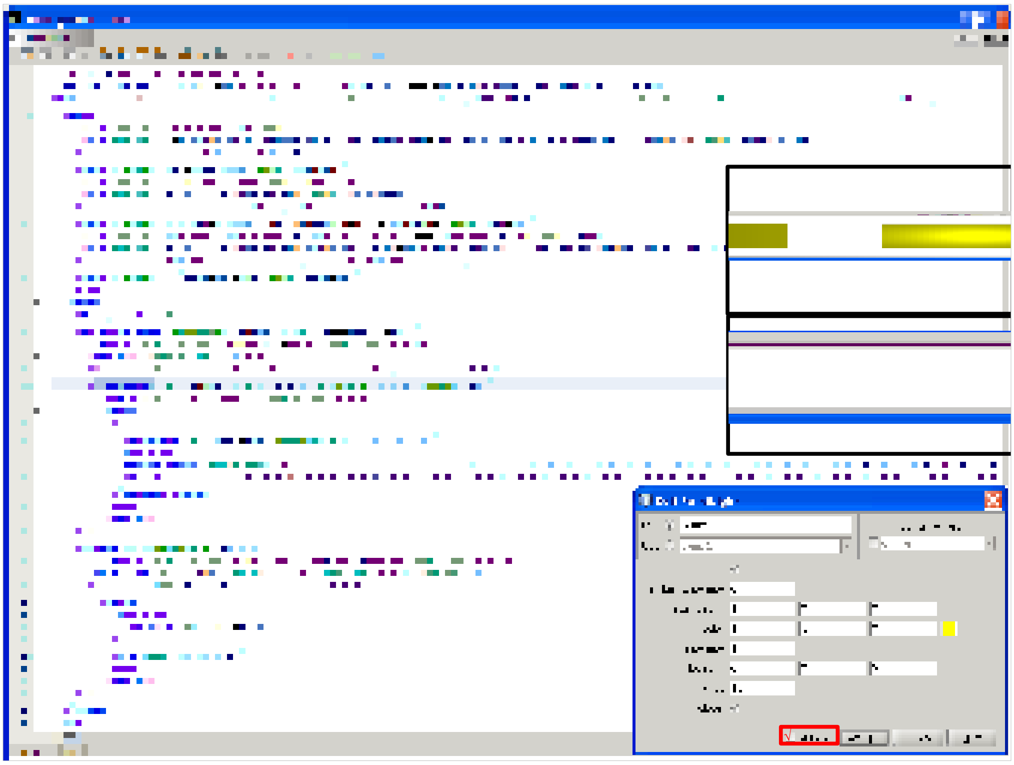
- Computed on a polygon-per-polygon basis for computational efficiency

This effect is not visible if a large polygon is used

- Since entire polygon receives consistent shading

Thus special tools for detailed tesselation of simple shapes might be needed to achieve special lighting effects

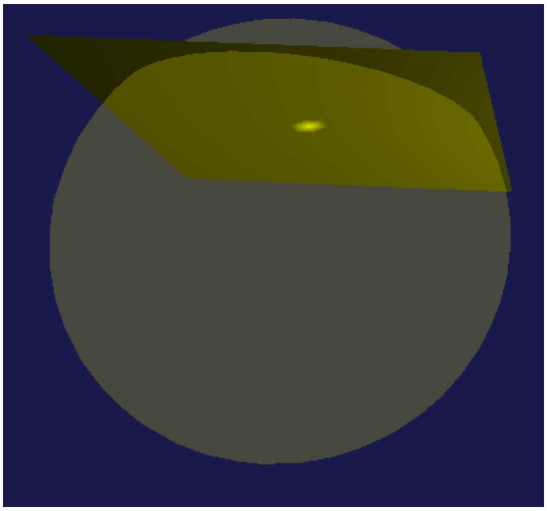
- Careful choice of viewing angle usually sufficient

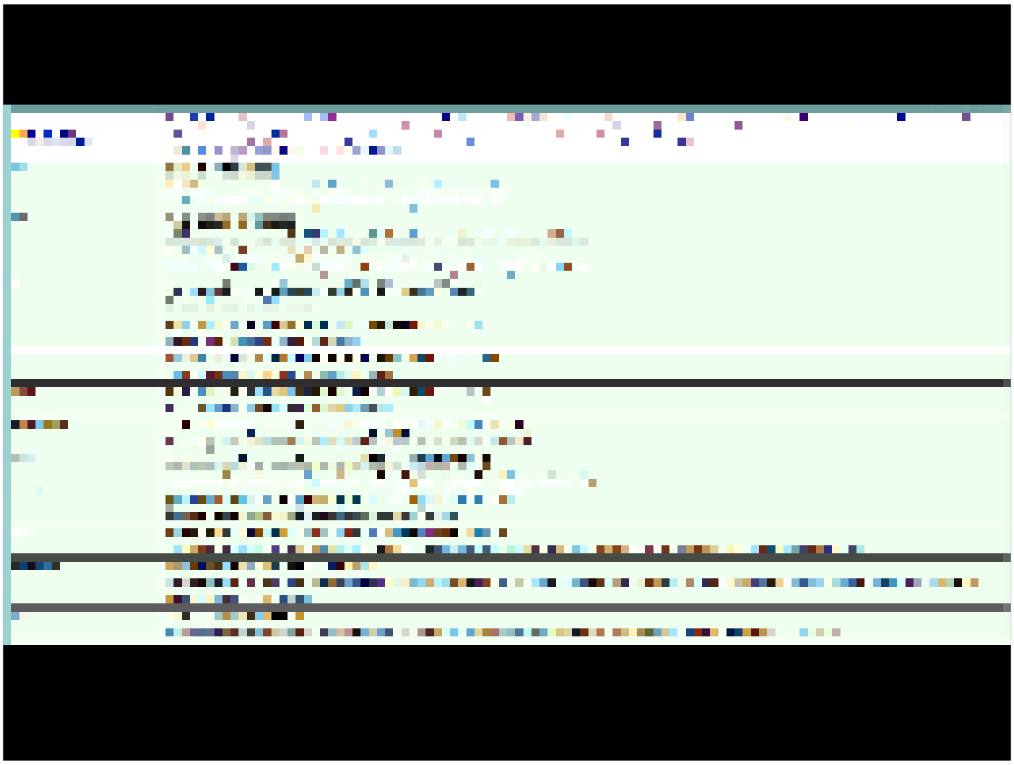


Page 313, (improved snapshot)

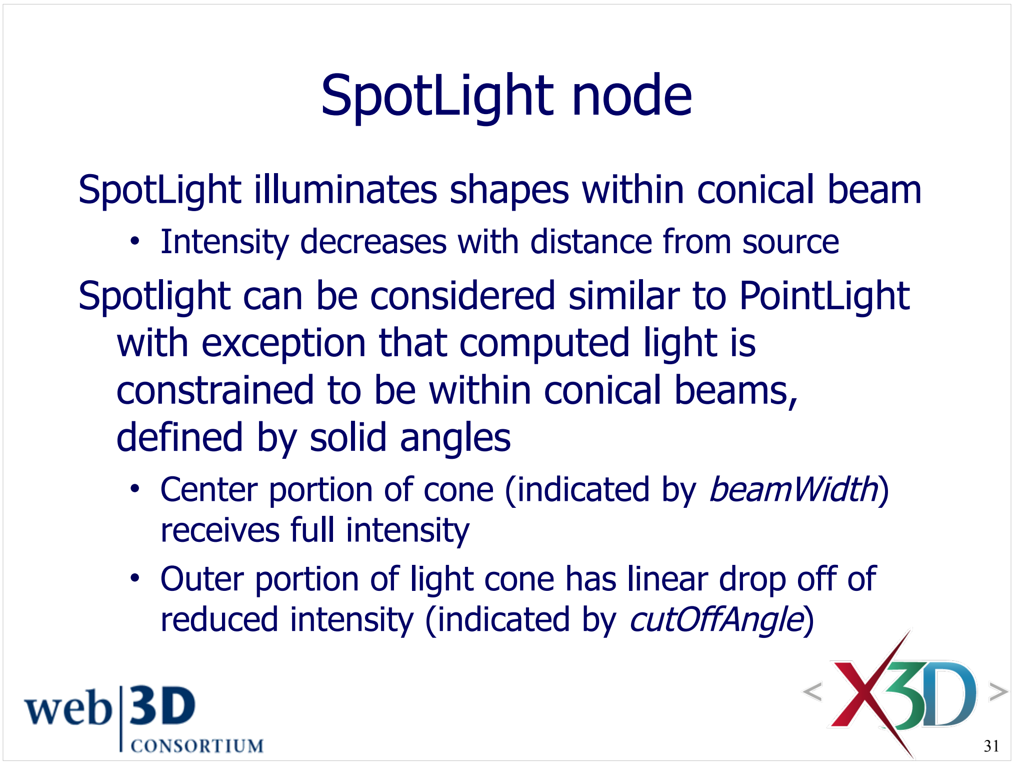
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/PointLight.x3d

Visualizing PointLight: when the “show outline” checkbox is selected, a Sphere with the correct location and radius is inserted in the scene with the same *emissiveColor* as the PointLight.  This authoring assist can help visualize the coverage of lights.





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



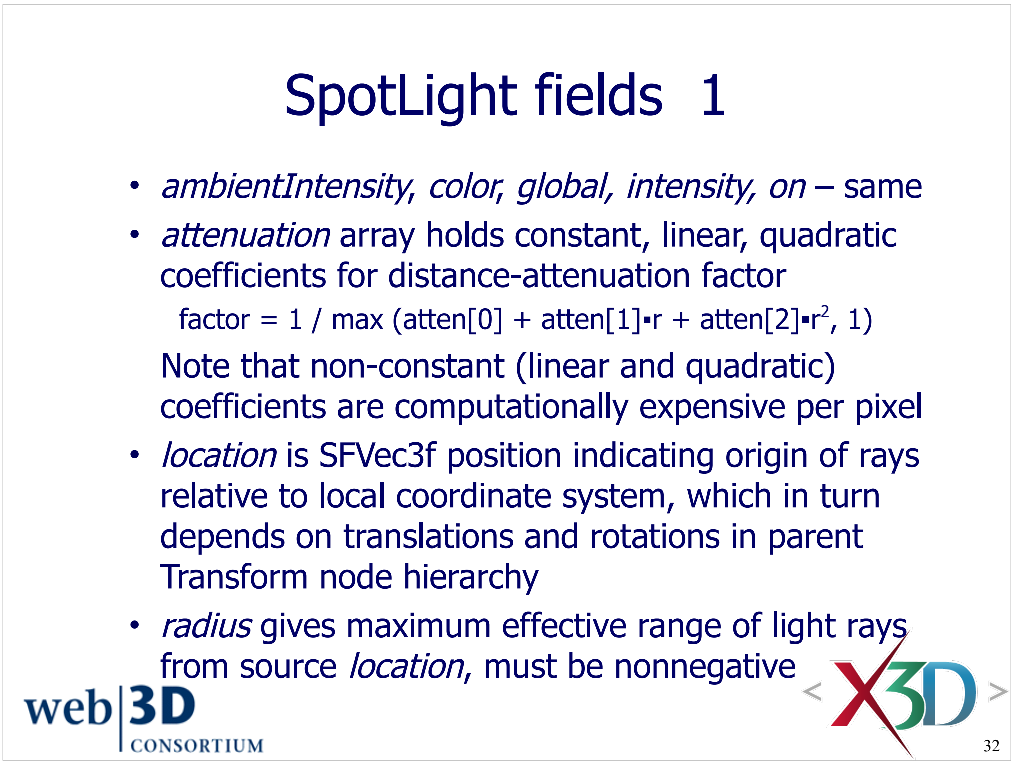
# SpotLight node

SpotLight illuminates shapes within conical beam

- Intensity decreases with distance from source

Spotlight can be considered similar to PointLight with exception that computed light is constrained to be within conical beams, defined by solid angles

- Center portion of cone (indicated by *beamWidth*) receives full intensity
- Outer portion of light cone has linear drop off of reduced intensity (indicated by *cutOffAngle*)



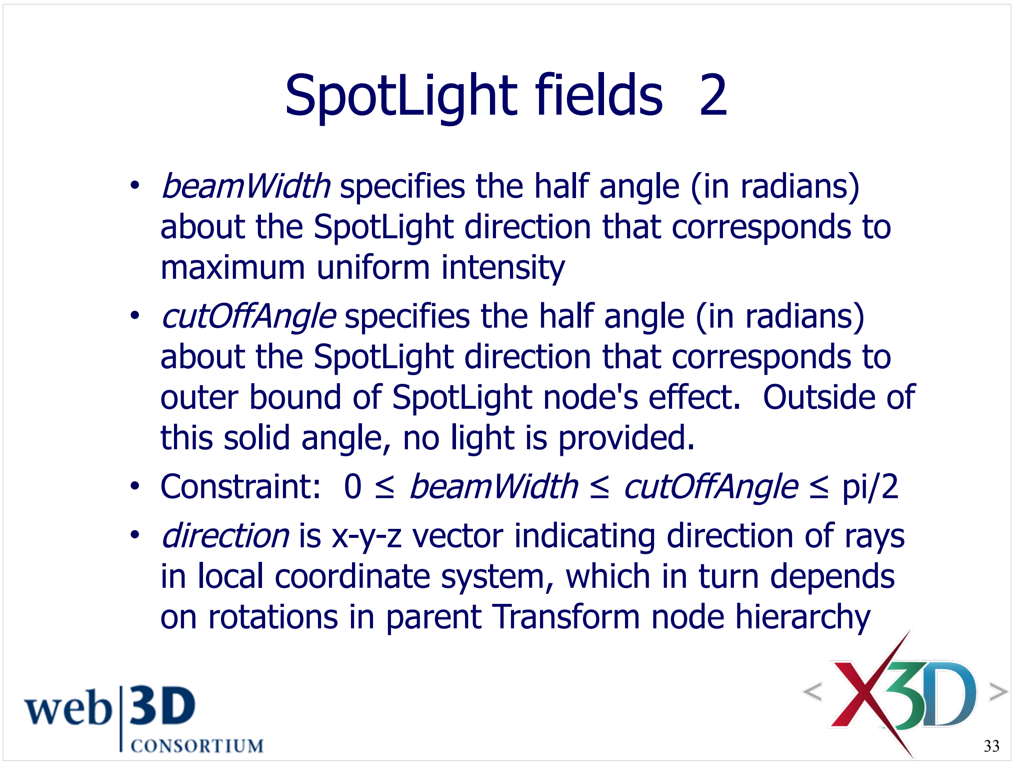
## SpotLight fields 1

- *ambientIntensity, color, global, intensity, on* – same
- *attenuation* array holds constant, linear, quadratic coefficients for distance-attenuation factor

  $factor = 1 / max\ (atten[0] + atten[1]\blacksquare r + atten[2]\blacksquare r^2, 1)$

  Note that non-constant (linear and quadratic) coefficients are computationally expensive per pixel
- *location* is SFVec3f position indicating origin of rays relative to local coordinate system, which in turn depends on translations and rotations in parent Transform node hierarchy
- *radius* gives maximum effective range of light rays from source *location*, must be nonnegative

These Spotlight fields are identical to the PointLight fields. SpotLight contains a superset of the PointLight fields.



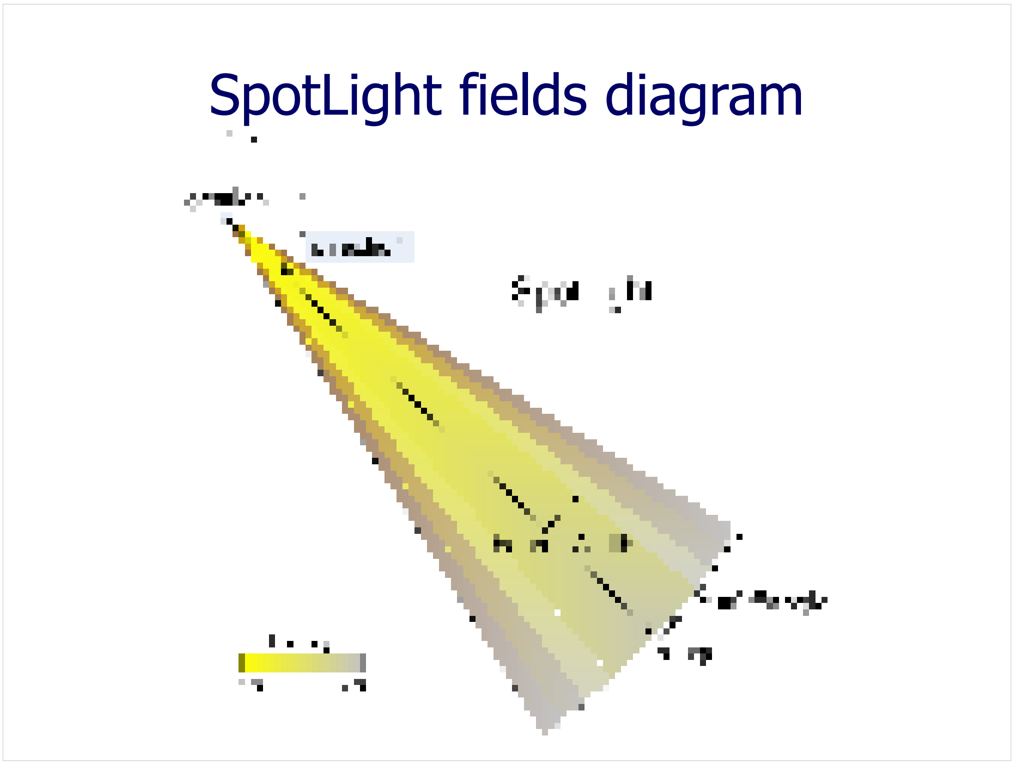
# SpotLight fields 2

- *beamWidth* specifies the half angle (in radians) about the SpotLight direction that corresponds to maximum uniform intensity
- *cutOffAngle* specifies the half angle (in radians) about the SpotLight direction that corresponds to outer bound of SpotLight node's effect. Outside of this solid angle, no light is provided.
- Constraint: 0 ≤ *beamWidth* ≤ *cutOffAngle* ≤ pi/2
- *direction* is x-y-z vector indicating direction of rays in local coordinate system, which in turn depends on rotations in parent Transform node hierarchy

SpotLight *direction* field is the same as PointLight *direction* field.

If the inner *beamWidth* angle is mistakenly set greater than the outer *cutOffAngle*, then *cutOffAngle* still determines the maximum width of the SpotLight cone and internal intensity is uniform throughout.

The X3D-Edit SpotLight editor will offer to swap values if inner *beamWidth* angle is mistakenly set greater than the outer *cutOffAngle.*

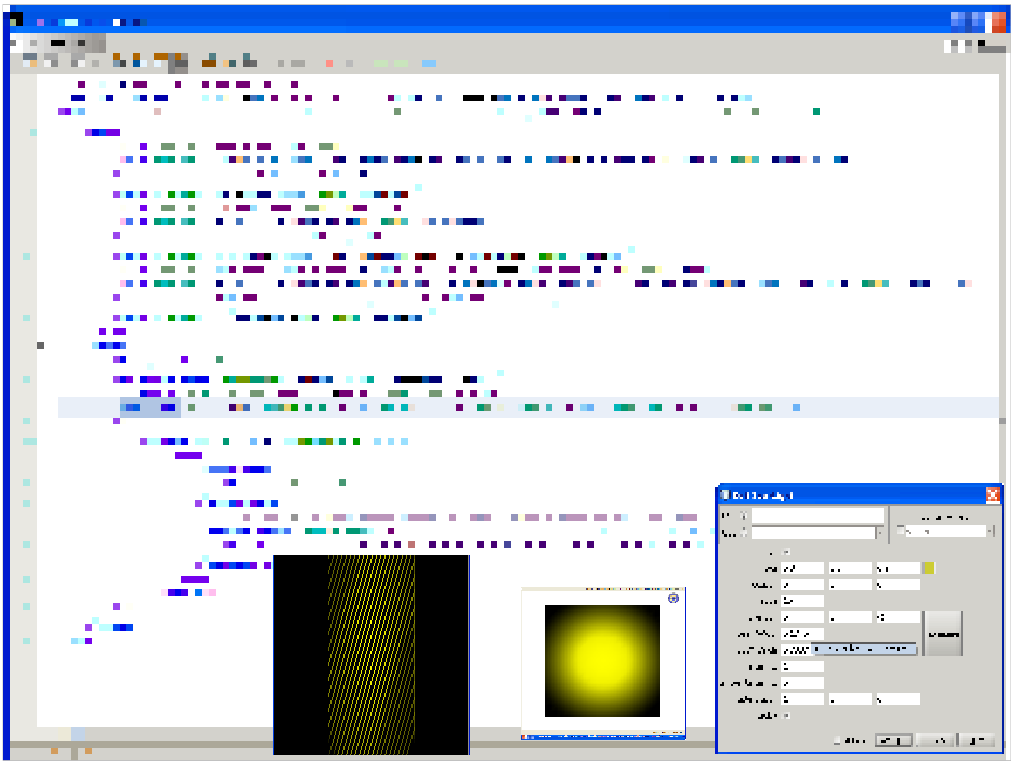


# SpotLight fields diagram

Page 316, Figure 11.7. SpotLight provides a virtual light source such that rays emanate radially with a conical outline from a single location.

**Note earlier error** on original figure in the published book which incorrectly swapped *beamWidth* and *cutOffAngle*.

Book errata are maintained at http://x3dgraphics.com/errata.php



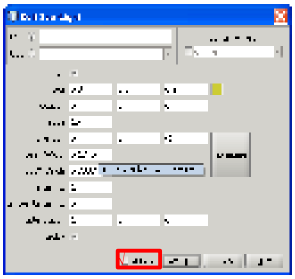
Page 317, Figure 11.8. Central beam intensity and fall-off are visible in this SpotLight example.

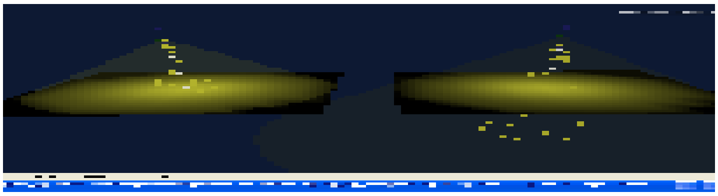
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/SpotLight.x3d

## SpotLight visualization

X3D-Edit has an authoring assist to show outlines for SpotLight fields

- *cutOffAngle* shown by wireframe
- *beamWidth* shown by transparent Cone
- Example: SpotLightVisualization.x3d





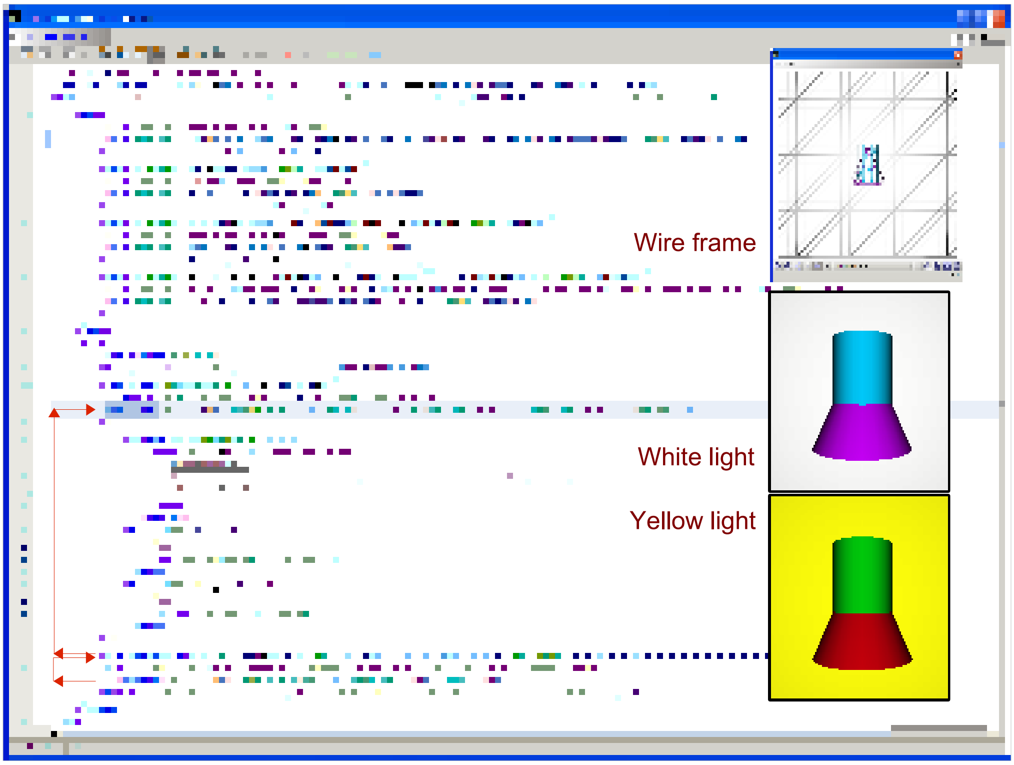
A common difficulty when authoring SpotLight nodes is to not be able to find their effect. The SpotLight may be pointed in a different direction than expected, or perhaps located in an unexpected position. Such surprises may be due to parent Transform nodes.

One simple technique to test where a SpotLight is located is to change it to a PointLight (for omnidirectional effect) and perhaps adding a sphere at the expected location. This approach can be somewhat tedious, however.

X3D-Edit includes a selectable checkbox for adding geometry to a scene that matches SpotLight dimensions and directions. A special example scene (SpotLightVisualization.x3d) is also provided to illustrate this author-assist feature. Edit the first SpotLight in the scene and select 'show outline' to see the pair of cones appear on the right-hand side.

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironmentalEffects/SpotLightVisualization.x3d

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/PointLightVisualization.png

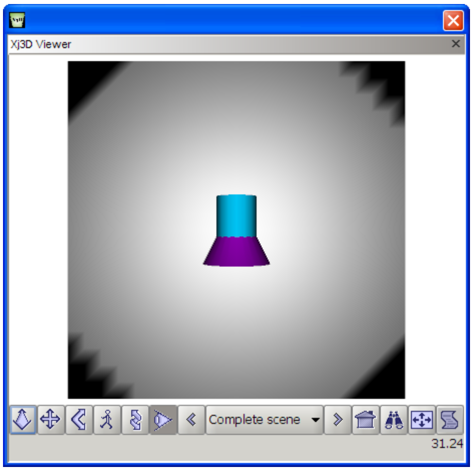


SpotLightColor.x3d provides an example of SpotLight showing effect of varying light colors on appearance, the color of objects can appear quite different.

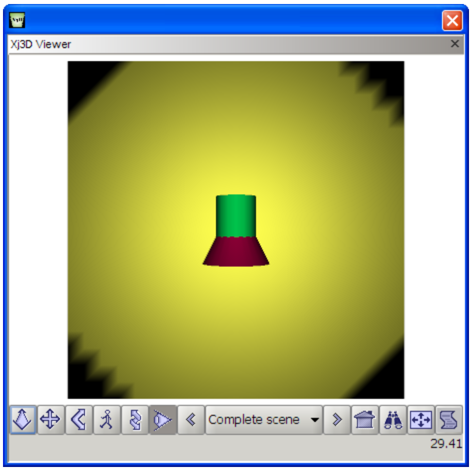
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/SpotLightColor.x3d

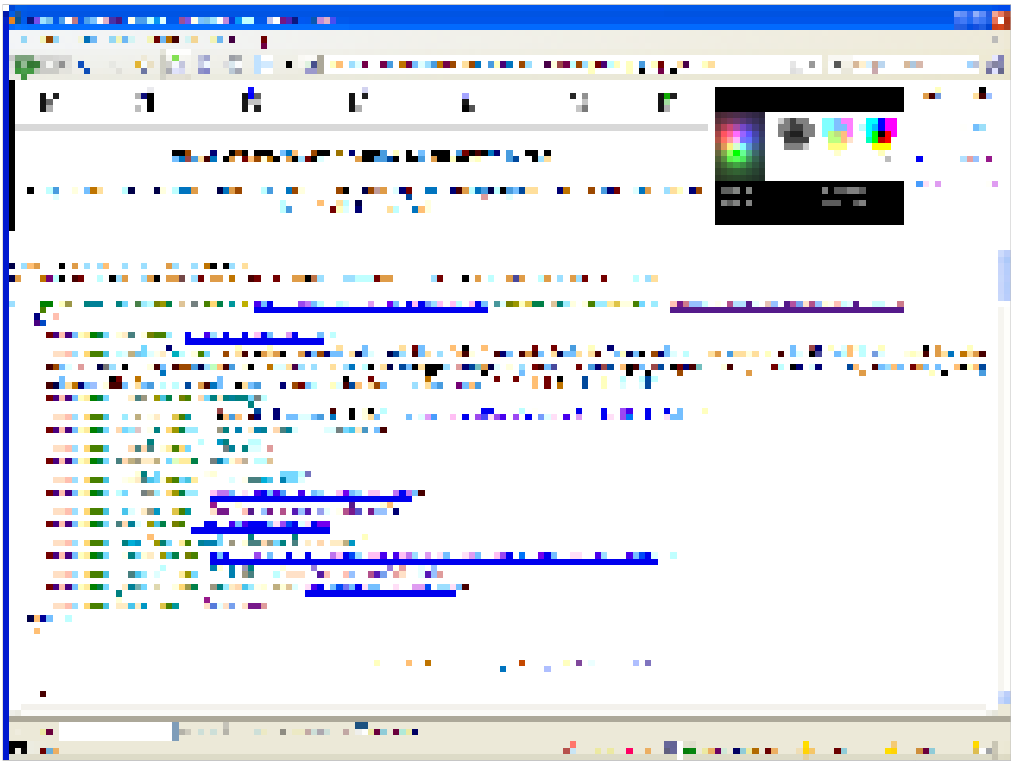
Zooming out to see the complete scene shows the background grid behind the cylinder and cone. Note that the scene background color is unaffected by lighting.

White light



Yellow light



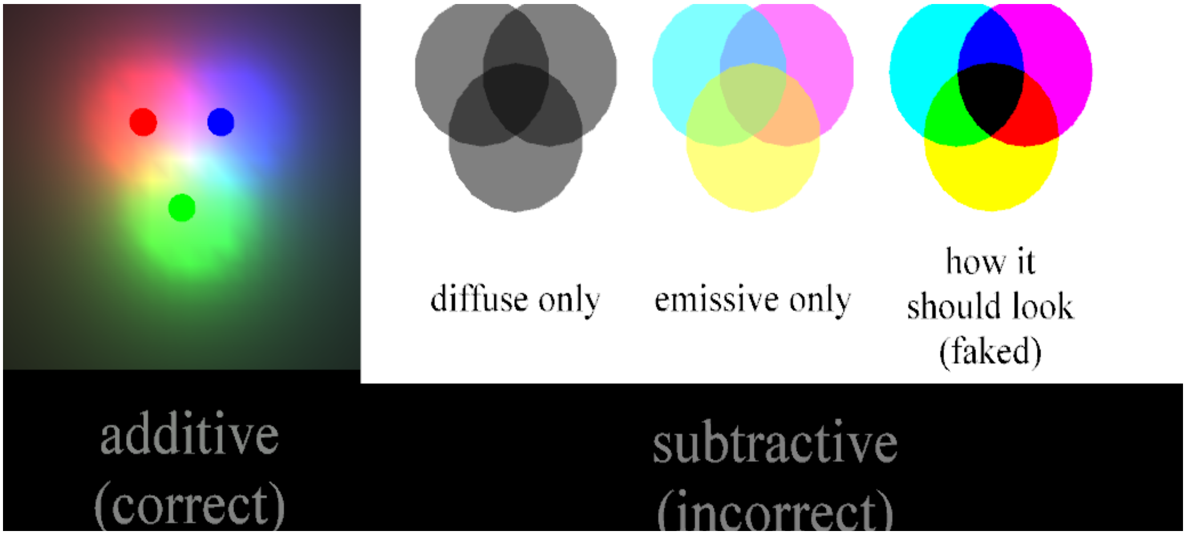


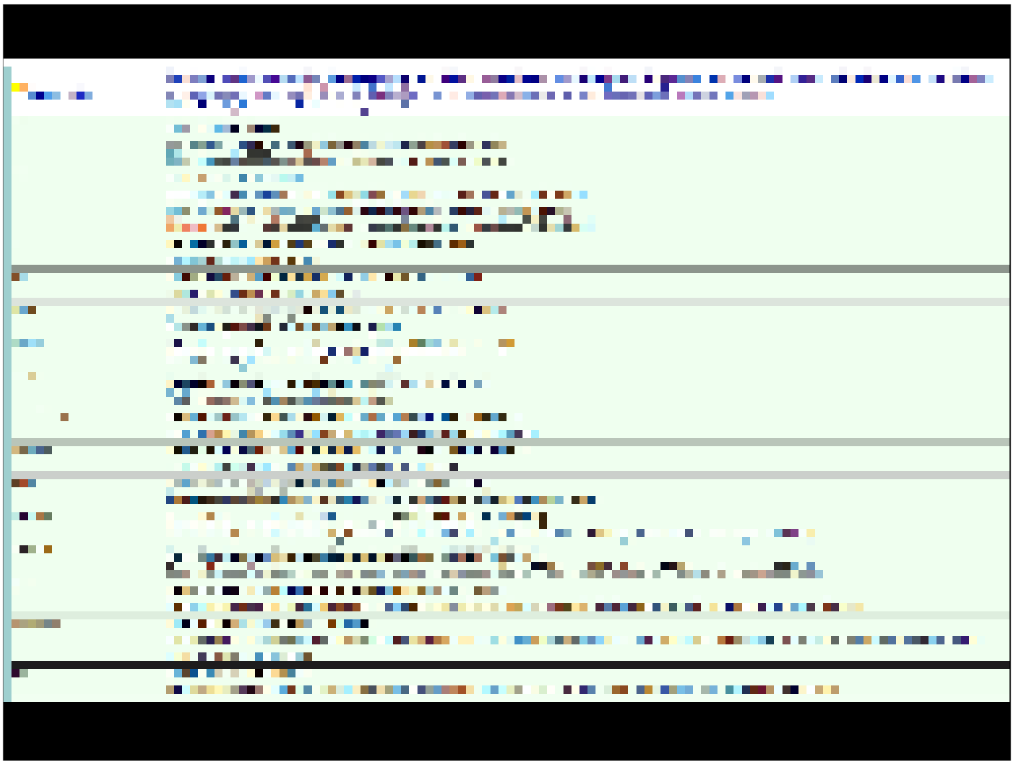
Interesting example from the X3D Basic archives.  Original author Miriam English.

http://www.web3d.org/x3d/content/examples/Basic/development/_pages/page01.html
http://www.web3d.org/x3d/content/examples/Basic/development/AdditiveSubtractiveLight.x3d

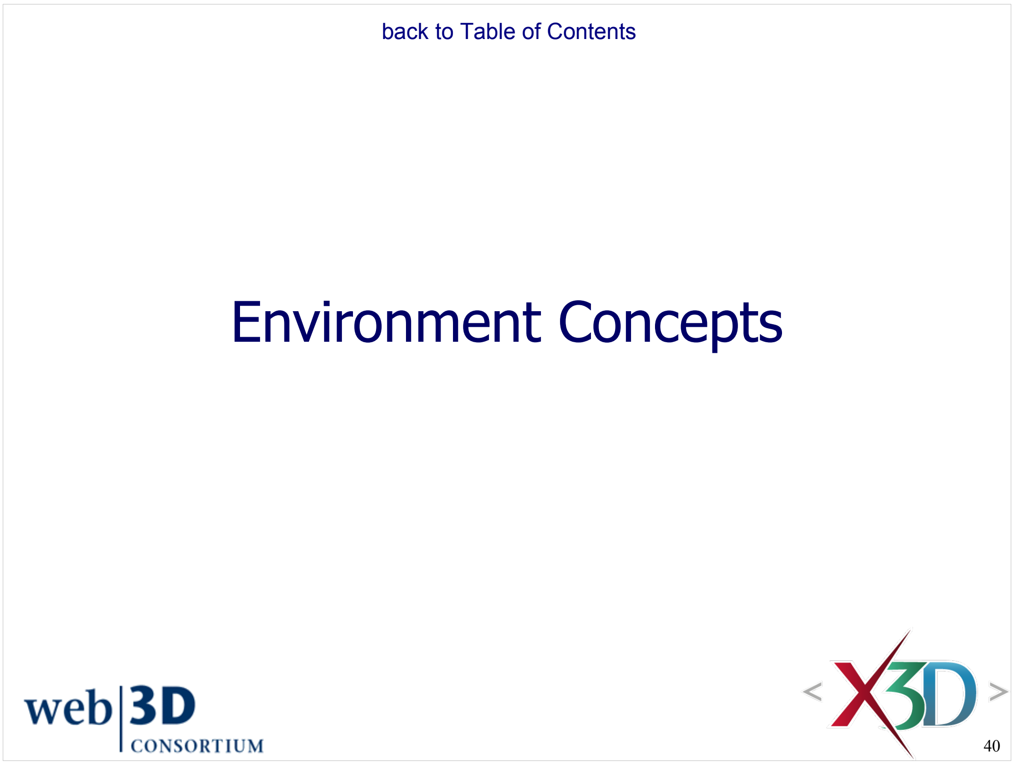
*Additive Subtractive Light* example description.  This example tests various color combination capabilities, and also demonstrating that filtering (negative colors) are not supported in the X3D/VRML color model. Discussion follows. There are 2 ways to combine colors:

- Additive (e.g. a TV screen, overlapping spotlights, or pointillist art), for example
  [red + green = yellow] [1 0 0 + 0 1 0 = 1 1 0].
- Subtractive (e.g. mixing paints or stacking filters), for example
  [white background + yellow filter + cyan filter = green] [1 1 1 + 0 0 -1 + -1 0 0 = 0 1 0].



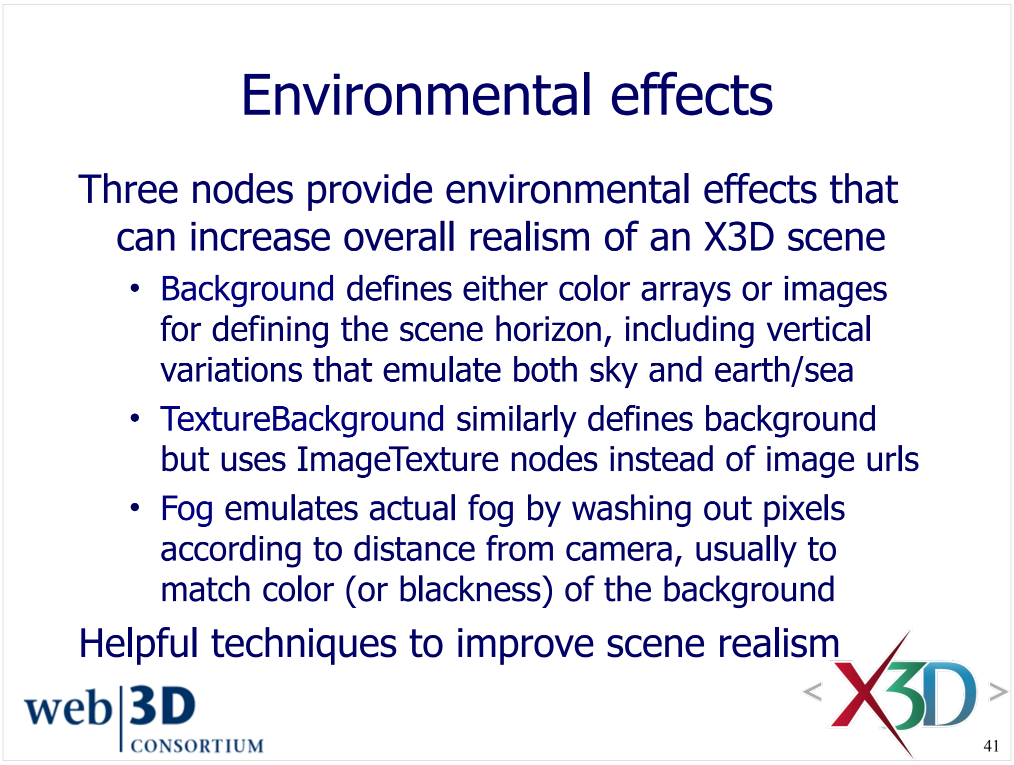


http://www.web3d.org/x3d/content/X3dTooltips.html#SpotLight



back to Table of Contents

# Environment Concepts

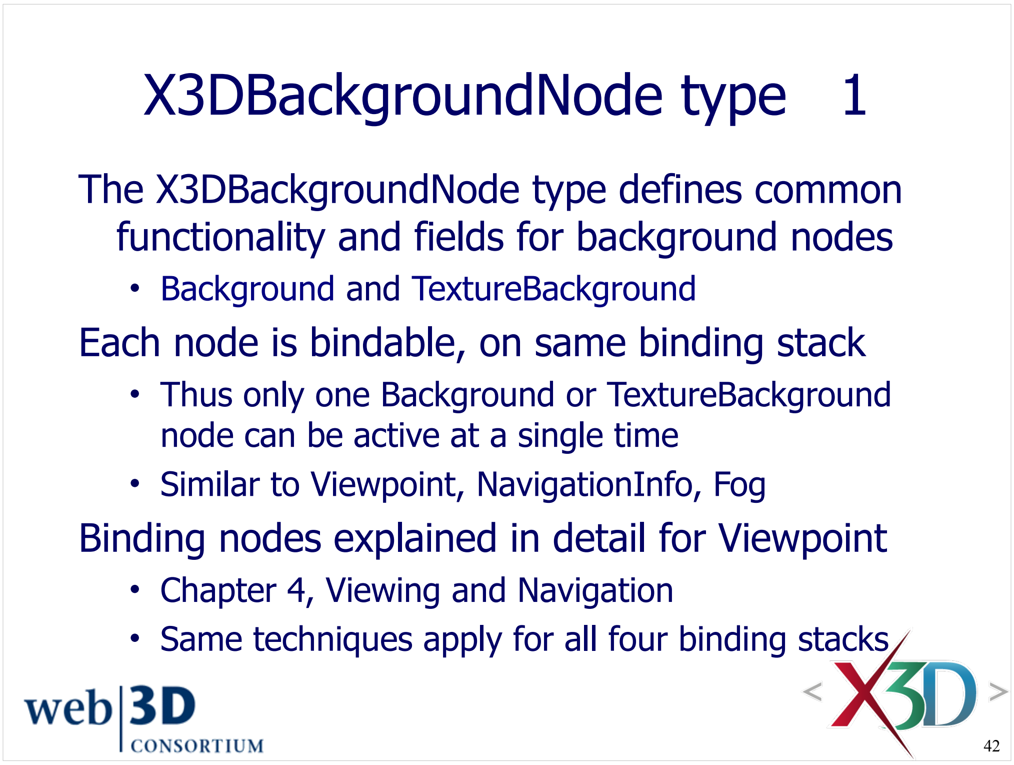


# Environmental effects

Three nodes provide environmental effects that can increase overall realism of an X3D scene

- Background defines either color arrays or images for defining the scene horizon, including vertical variations that emulate both sky and earth/sea
- TextureBackground similarly defines background but uses ImageTexture nodes instead of image urls
- Fog emulates actual fog by washing out pixels according to distance from camera, usually to match color (or blackness) of the background

Helpful techniques to improve scene realism



# X3DBackgroundNode type 1

The X3DBackgroundNode type defines common functionality and fields for background nodes

- Background and TextureBackground

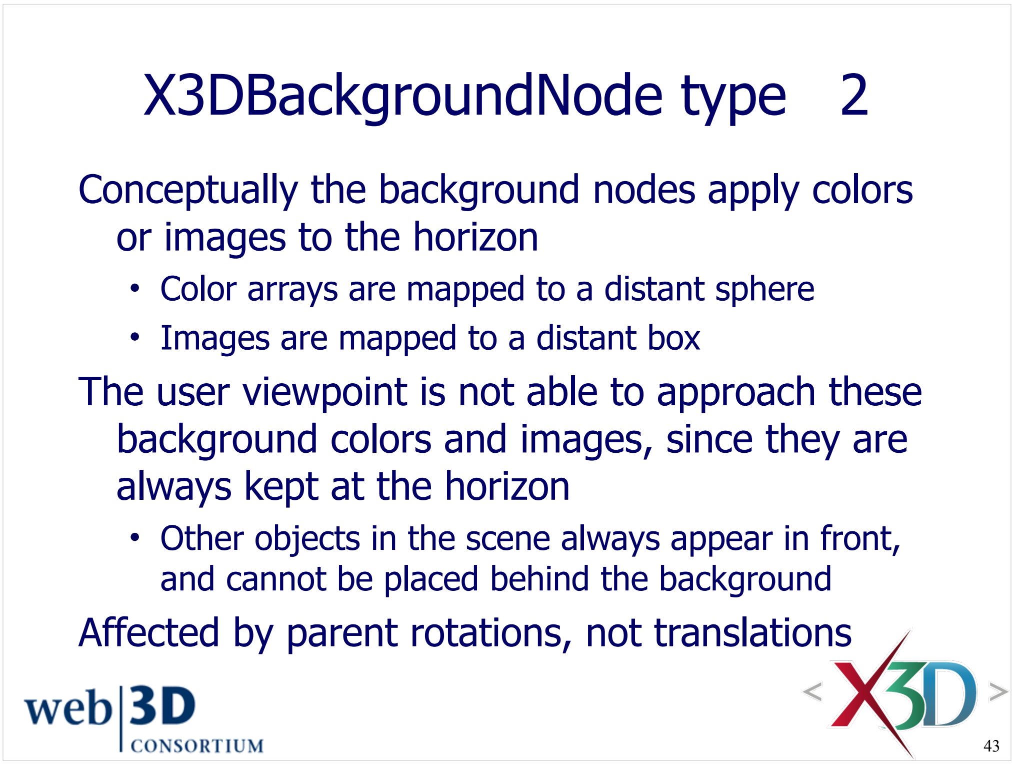
Each node is bindable, on same binding stack

- Thus only one Background or TextureBackground node can be active at a single time
- Similar to Viewpoint, NavigationInfo, Fog

Binding nodes explained in detail for Viewpoint

- Chapter 4, Viewing and Navigation
- Same techniques apply for all four binding stacks

Typically browsers only provide selection controls for the Viewpoint stack and do not provide selection controls for the Background, Fog or NavigationInfo stacks. Thus control and selection of these nodes is only handled by animation chains created by the author.



# X3DBackgroundNode type 2

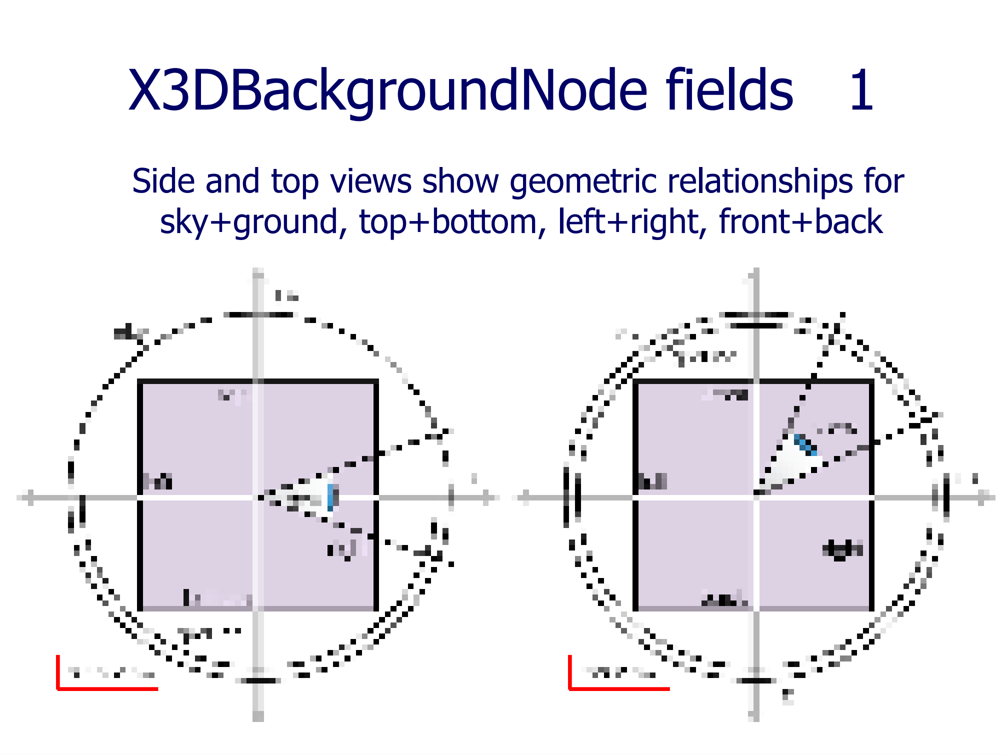
Conceptually the background nodes apply colors or images to the horizon

- Color arrays are mapped to a distant sphere
- Images are mapped to a distant box

The user viewpoint is not able to approach these background colors and images, since they are always kept at the horizon

- Other objects in the scene always appear in front, and cannot be placed behind the background

Affected by parent rotations, not translations

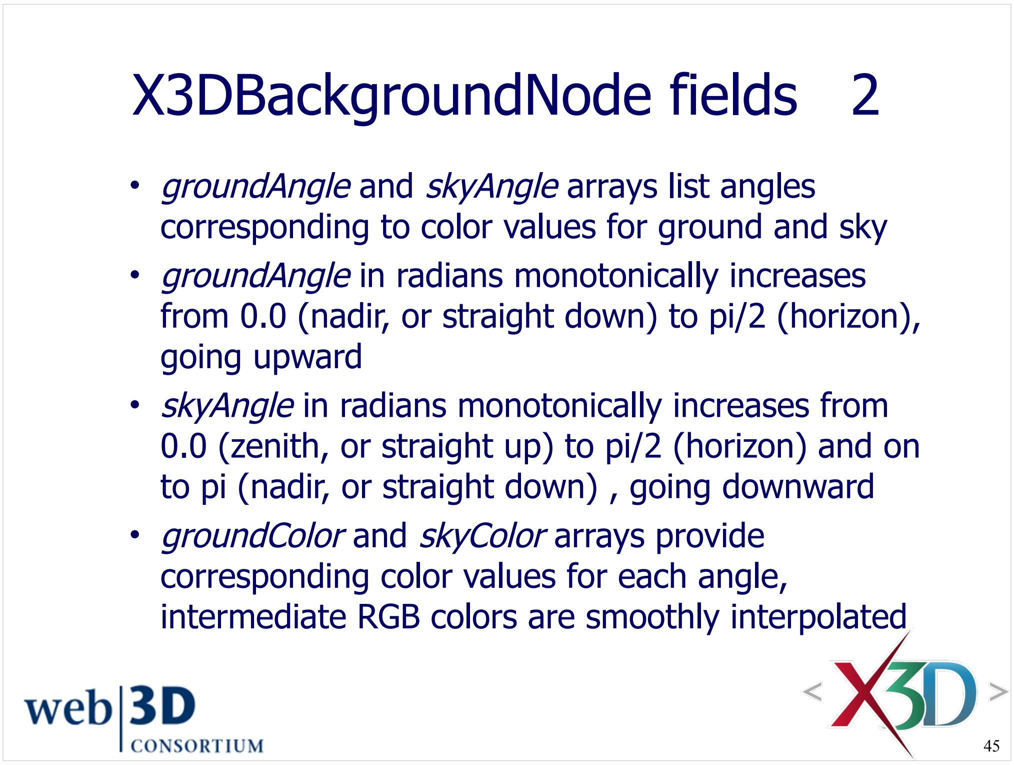


# X3DBackgroundNode fields 1

Side and top views show geometric relationships for sky+ground, top+bottom, left+right, front+back

Page 307, Figure 11.2. Placement of background colors and images relative to the user’s viewpoint.

Adapted with permission from X3D Abstract Specification, Figure 24.1.



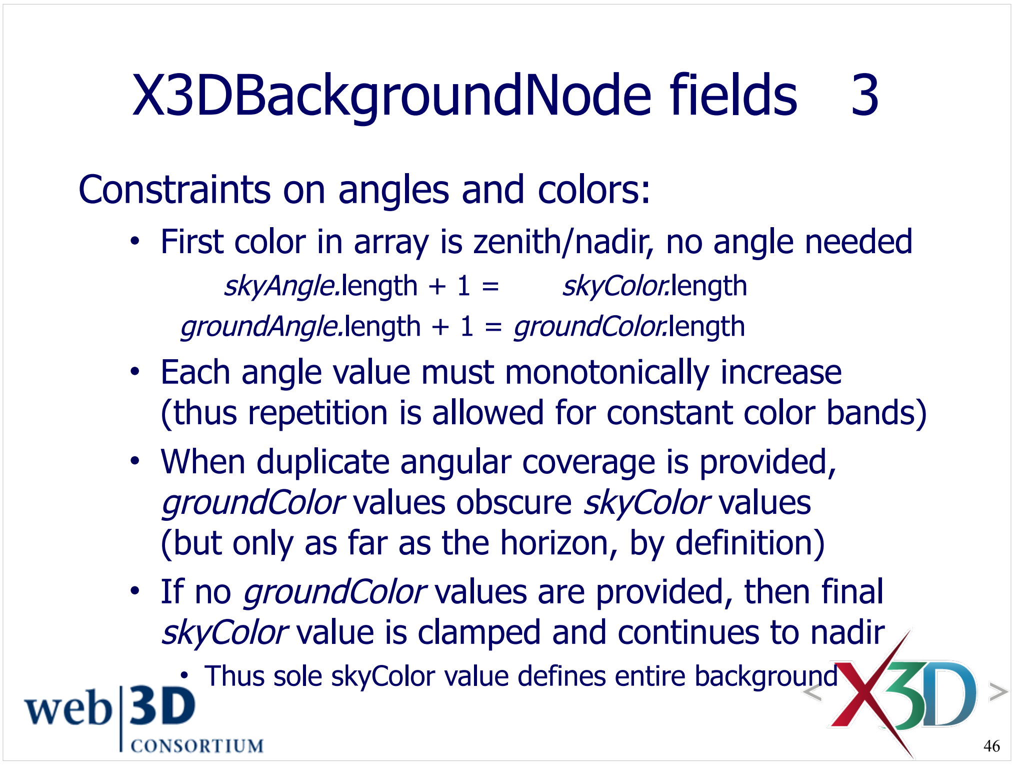
# X3DBackgroundNode fields 2

- *groundAngle* and *skyAngle* arrays list angles corresponding to color values for ground and sky
- *groundAngle* in radians monotonically increases from 0.0 (nadir, or straight down) to pi/2 (horizon), going upward
- *skyAngle* in radians monotonically increases from 0.0 (zenith, or straight up) to pi/2 (horizon) and on to pi (nadir, or straight down) , going downward
- *groundColor* and *skyColor* arrays provide corresponding color values for each angle, intermediate RGB colors are smoothly interpolated

web3D CONSORTIUM

<X3D>

45



# X3DBackgroundNode fields 3

Constraints on angles and colors:

- First color in array is zenith/nadir, no angle needed

  *skyAngle*.length + 1 = *skyColor*.length

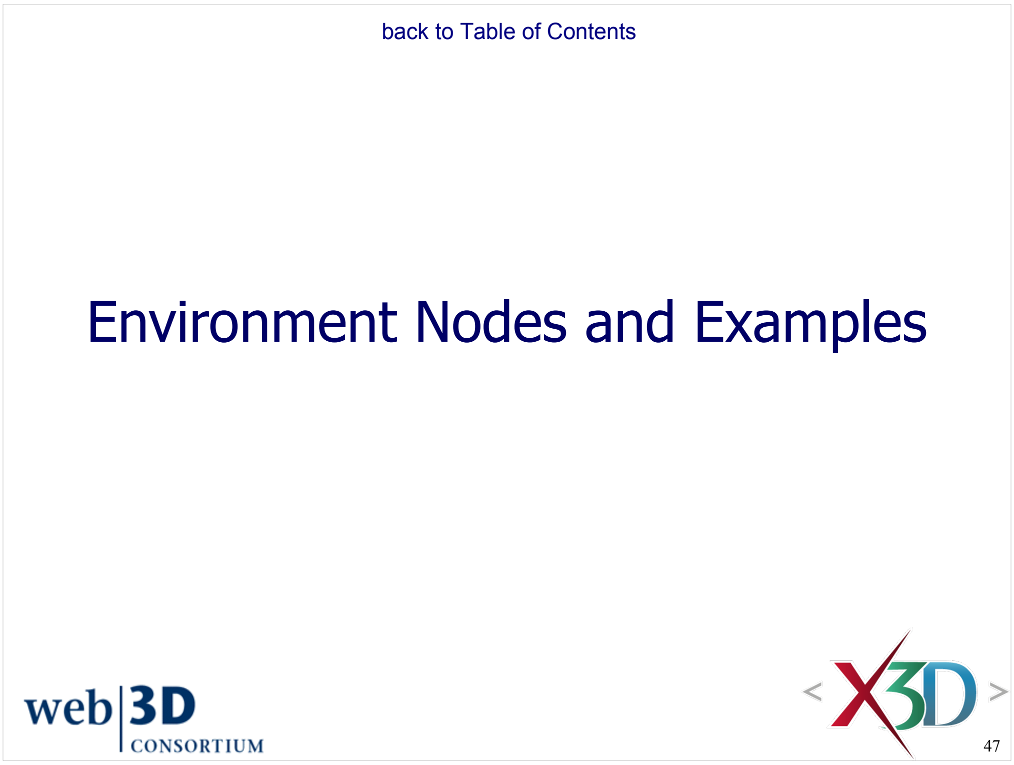
  *groundAngle*.length + 1 = *groundColor*.length

- Each angle value must monotonically increase (thus repetition is allowed for constant color bands)
- When duplicate angular coverage is provided, *groundColor* values obscure *skyColor* values (but only as far as the horizon, by definition)
- If no *groundColor* values are provided, then final *skyColor* value is clamped and continues to nadir
  - Thus sole skyColor value defines entire background

46

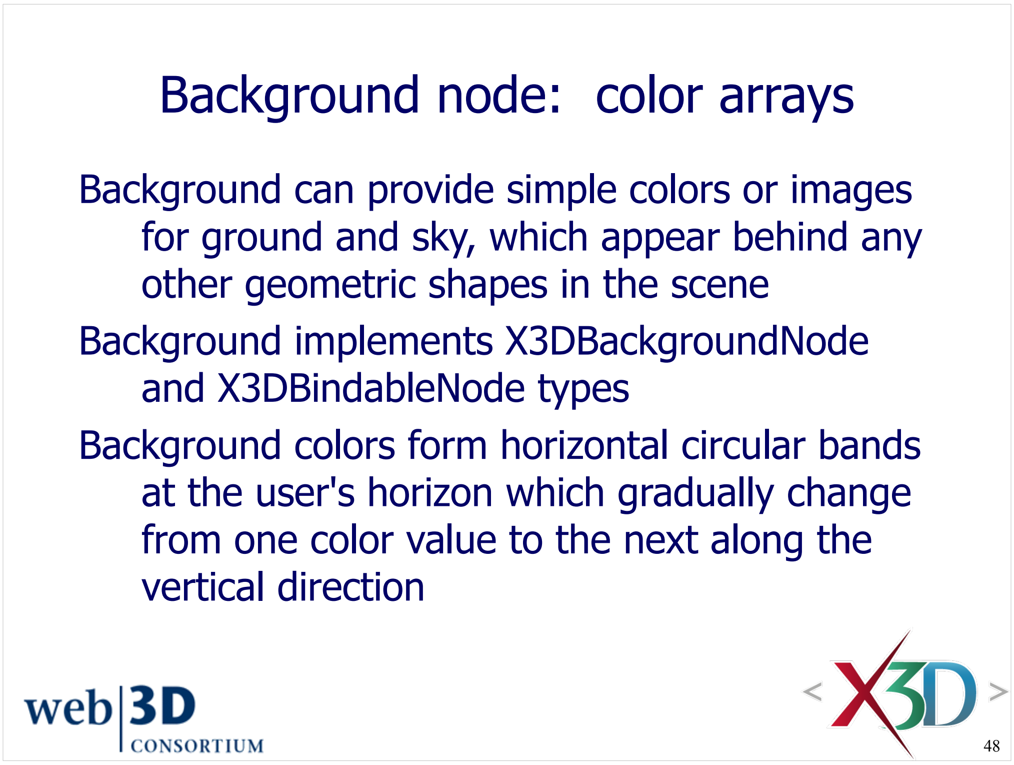
“Monotonically increase” is a mathematical term that indicates each value in the array must be greater than, or equal to, the value which precedes it.

If provided, the groundColor array obscures the skyColor array values. If no groundColor array is provided, then skyColor provides complete vertical coverage.



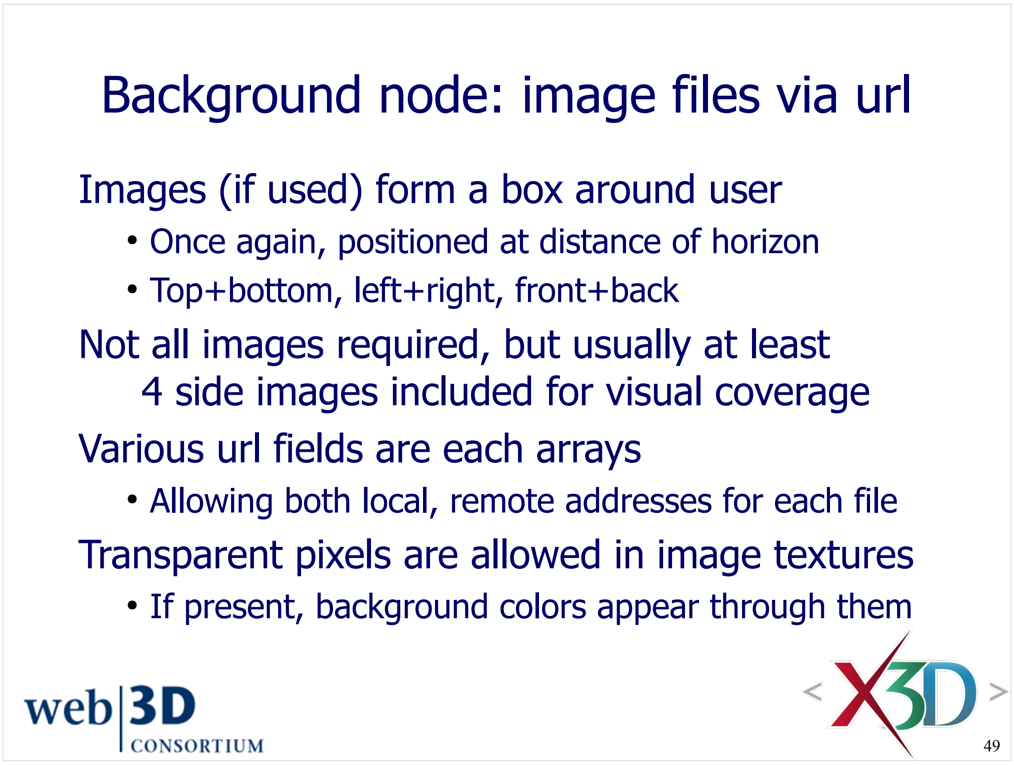
back to Table of Contents

# Environment Nodes and Examples



## Background node: color arrays

- Background can provide simple colors or images for ground and sky, which appear behind any other geometric shapes in the scene
- Background implements X3DBackgroundNode and X3DBindableNode types
- Background colors form horizontal circular bands at the user's horizon which gradually change from one color value to the next along the vertical direction



## Background node: image files via url

Images (if used) form a box around user

- Once again, positioned at distance of horizon
- Top+bottom, left+right, front+back

Not all images required, but usually at least
4 side images included for visual coverage

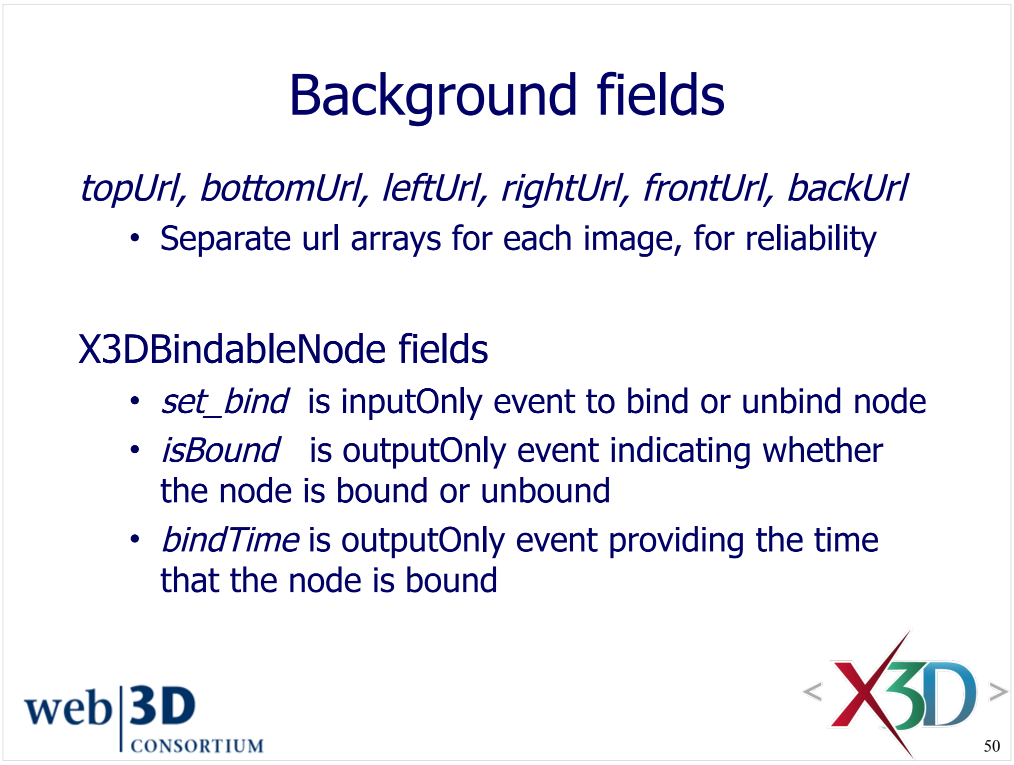
Various url fields are each arrays

- Allowing both local, remote addresses for each file

Transparent pixels are allowed in image textures

- If present, background colors appear through them

Image files are covered in detail in Chapter 5, Appearance Material and Textures.



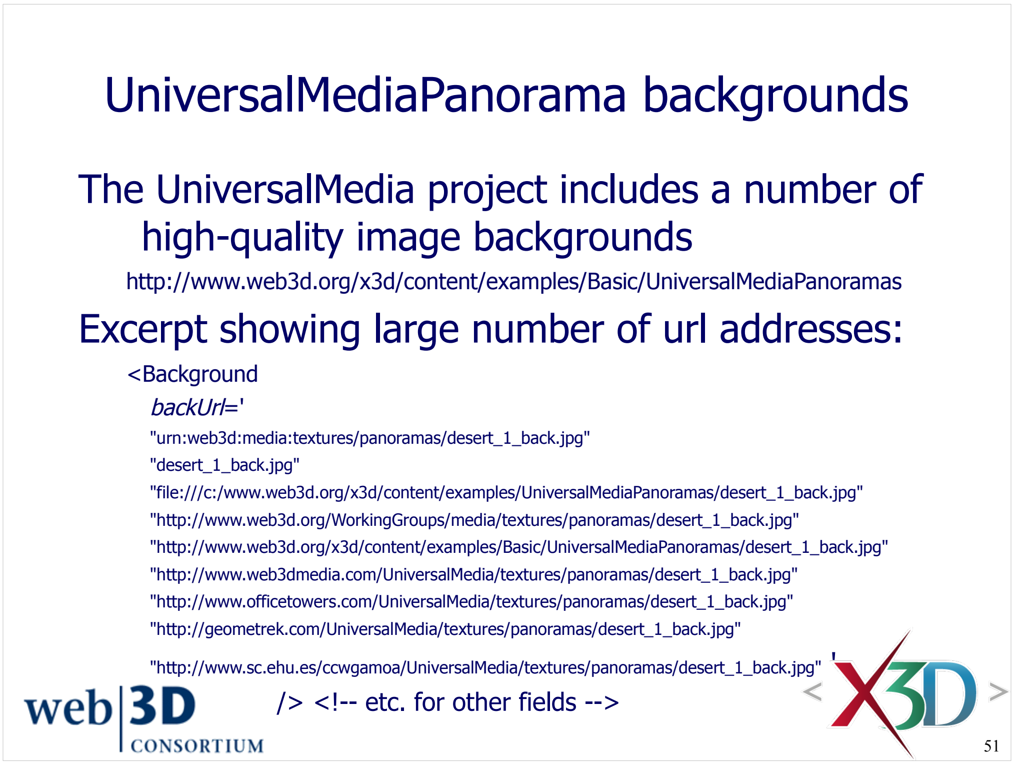
# Background fields

*topUrl, bottomUrl, leftUrl, rightUrl, frontUrl, backUrl*

- Separate url arrays for each image, for reliability

X3DBindableNode fields

- *set_bind* is inputOnly event to bind or unbind node
- *isBound* is outputOnly event indicating whether the node is bound or unbound
- *bindTime* is outputOnly event providing the time that the node is bound



# UniversalMediaPanorama backgrounds

- The UniversalMedia project includes a number of high-quality image backgrounds
  - http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaPanoramas
- Excerpt showing large number of url addresses:

```
<Background
  backUrl='
  "urn:web3d:media:textures/panoramas/desert_1_back.jpg"
  "desert_1_back.jpg"
  "file:///c:/www.web3d.org/x3d/content/examples/UniversalMediaPanoramas/desert_1_back.jpg"
  "http://www.web3d.org/WorkingGroups/media/textures/panoramas/desert_1_back.jpg"
  "http://www.web3d.org/x3d/content/examples/Basic/UniversalMediaPanoramas/desert_1_back.jpg"
  "http://www.web3dmedia.com/UniversalMedia/textures/panoramas/desert_1_back.jpg"
  "http://www.officetowers.com/UniversalMedia/textures/panoramas/desert_1_back.jpg"
  "http://geometrek.com/UniversalMedia/textures/panoramas/desert_1_back.jpg"
  "http://www.sc.ehu.es/ccwgamoa/UniversalMedia/textures/panoramas/desert_1_back.jpg" '
          /> <!-- etc. for other fields -->
```

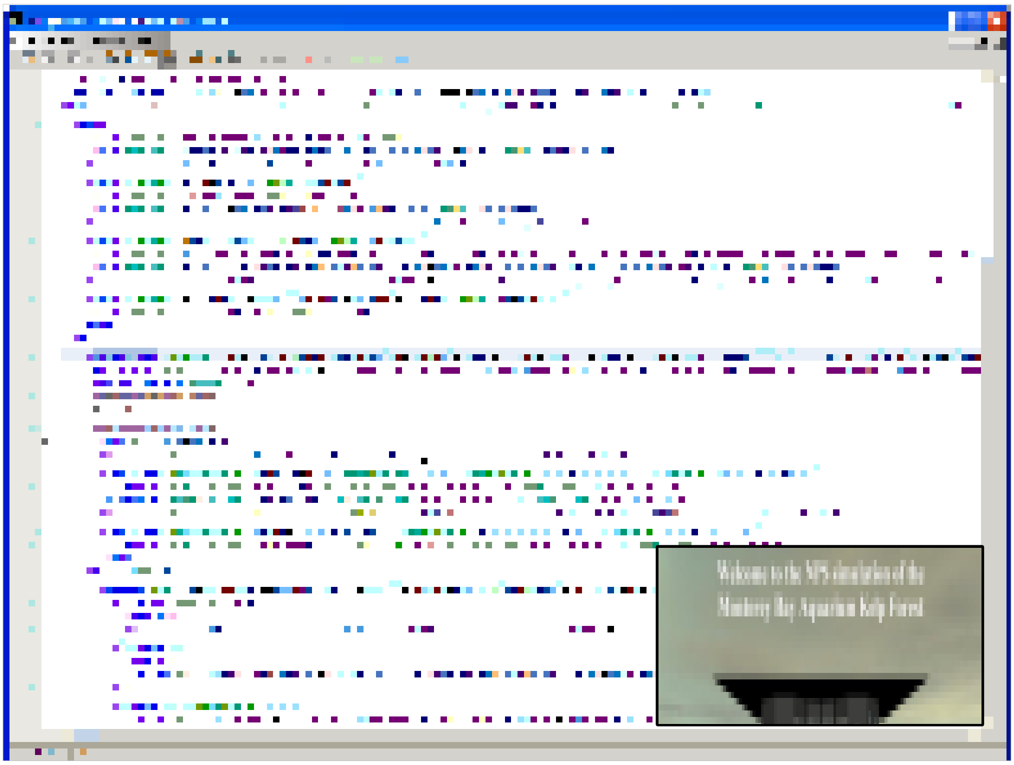
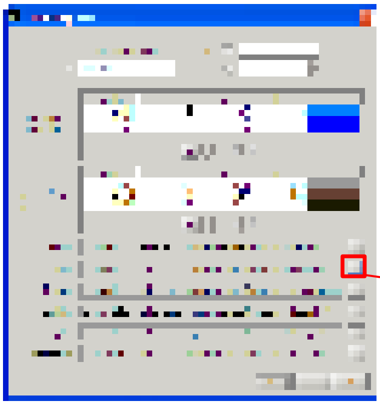

Welcome to the NPS simulation of the
Monterey Bay Aquarium Kelp Forest

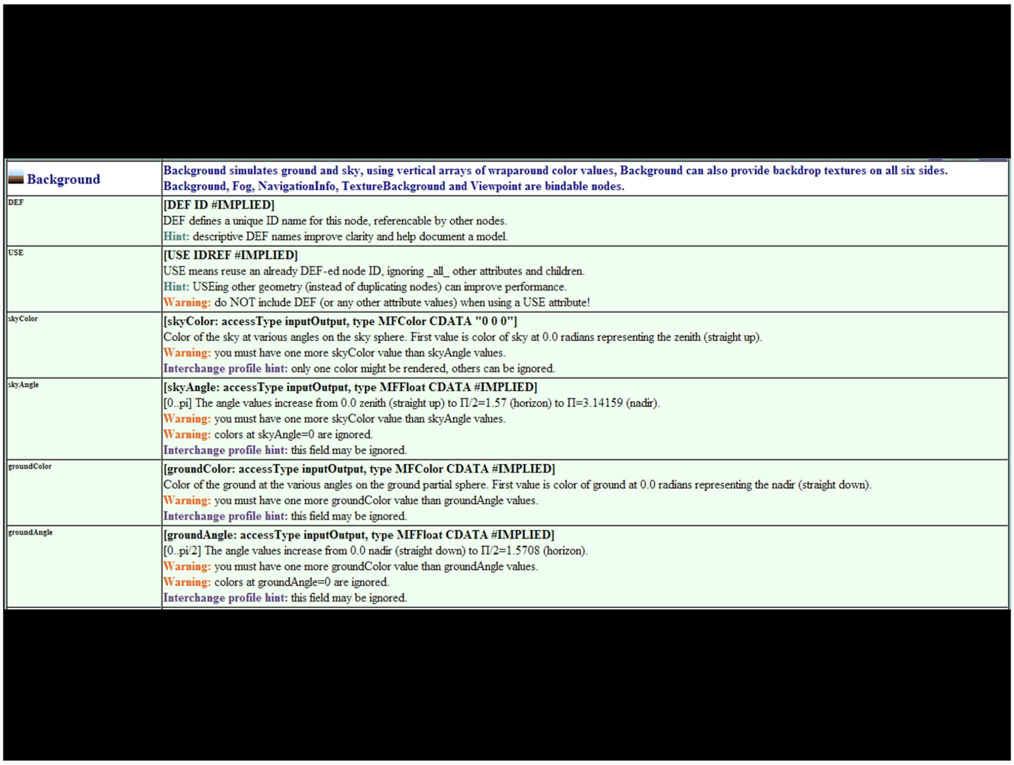
Figure 11.9. This Background example provides a backdrop panorama of an ocean environment.

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/Background-KelpForestMain.x3d

# Background node X3D-Edit

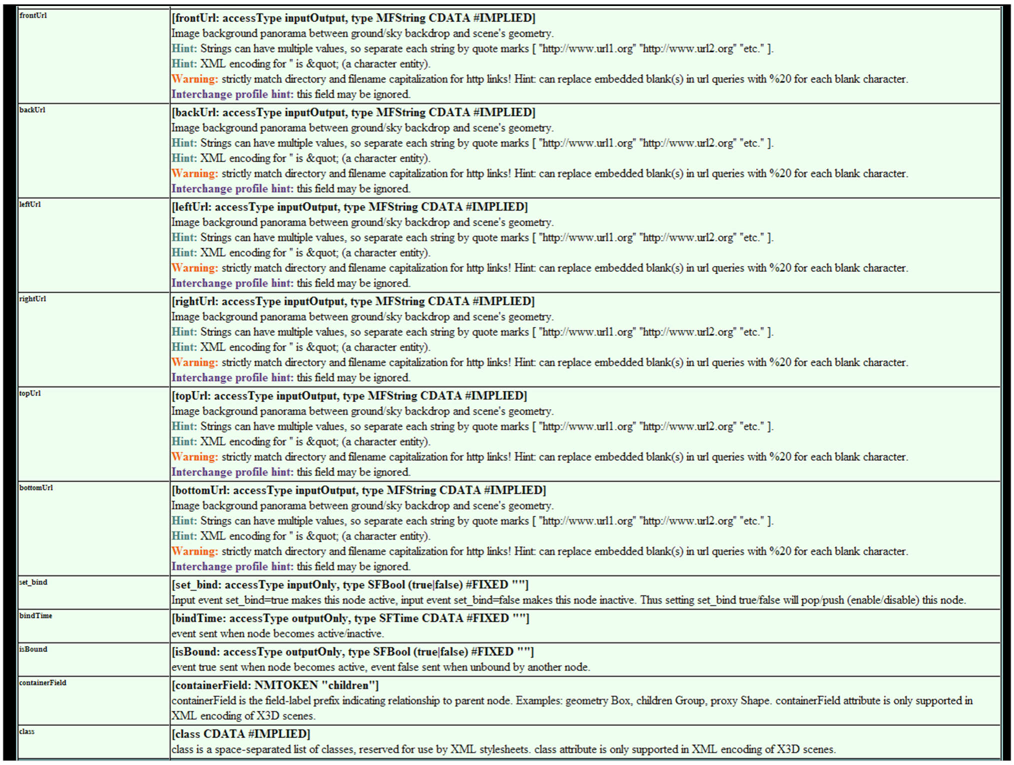


- *skyColor* values go down from zenith to nadir
- *groundColor* values go up from nadir to horizon



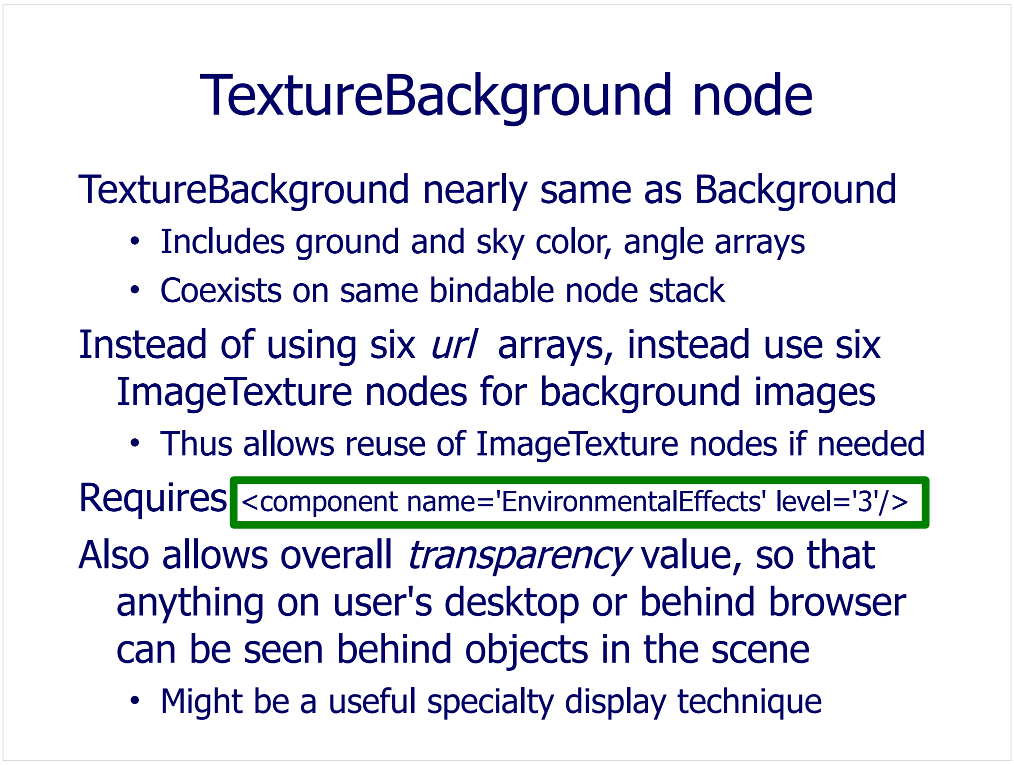
| **Background** | **Background simulates ground and sky, using vertical arrays of wraparound color values, Background can also provide backdrop textures on all six sides. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes.** |
|---|---|
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>**Hint:** descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>**Hint:** USEing other geometry (instead of duplicating nodes) can improve performance.<br>**Warning:** do NOT include DEF (or any other attribute values) when using a USE attribute! |
| skyColor | **[skyColor: accessType inputOutput, type MFColor CDATA "0 0 0"]**<br>Color of the sky at various angles on the sky sphere. First value is color of sky at 0.0 radians representing the zenith (straight up).<br>**Warning:** you must have one more skyColor value than skyAngle values.<br>**Interchange profile hint:** only one color might be rendered, others can be ignored. |
| skyAngle | **[skyAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi] The angle values increase from 0.0 zenith (straight up) to Π/2=1.57 (horizon) to Π=3.14159 (nadir).<br>**Warning:** you must have one more skyColor value than skyAngle values.<br>**Warning:** colors at skyAngle=0 are ignored.<br>**Interchange profile hint:** this field may be ignored. |
| groundColor | **[groundColor: accessType inputOutput, type MFColor CDATA #IMPLIED]**<br>Color of the ground at the various angles on the ground partial sphere. First value is color of ground at 0.0 radians representing the nadir (straight down).<br>**Warning:** you must have one more groundColor value than groundAngle values.<br>**Interchange profile hint:** this field may be ignored. |
| groundAngle | **[groundAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi/2] The angle values increase from 0.0 nadir (straight down) to Π/2=1.5708 (horizon).<br>**Warning:** you must have one more groundColor value than groundAngle values.<br>**Warning:** colors at groundAngle=0 are ignored.<br>**Interchange profile hint:** this field may be ignored. |

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



| | |
|---|---|
| frontUrl | **[frontUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| backUrl | **[backUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| leftUrl | **[leftUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| rightUrl | **[rightUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| topUrl | **[topUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| bottomUrl | **[bottomUrl: accessType inputOutput, type MFString CDATA #IMPLIED]**<br>Image background panorama between ground/sky backdrop and scene's geometry.<br>**Hint:** Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].<br>**Hint:** XML encoding for " is &quot; (a character entity).<br>**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.<br>**Interchange profile hint:** this field may be ignored. |
| set_bind | **[set_bind: accessType inputOnly, type SFBool (true\|false) #FIXED ""]**<br>Input event set_bind=true makes this node active, input event set_bind=false makes this node inactive. Thus setting set_bind true/false will pop/push (enable/disable) this node. |
| bindTime | **[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]**<br>event sent when node becomes active/inactive. |
| isBound | **[isBound: accessType outputOnly, type SFBool (true\|false) #FIXED ""]**<br>event true sent when node becomes active, event false sent when unbound by another node. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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#Background



# TextureBackground node

TextureBackground nearly same as Background

- Includes ground and sky color, angle arrays
- Coexists on same bindable node stack

Instead of using six *url* arrays, instead use six ImageTexture nodes for background images

- Thus allows reuse of ImageTexture nodes if needed

Requires <component name='EnvironmentalEffects' level='3'/>

Also allows overall *transparency* value, so that anything on user's desktop or behind browser can be seen behind objects in the scene

- Might be a useful specialty display technique

Reuse of ImageTexture nodes is accomplished by utilizing DEF/USE nodes.

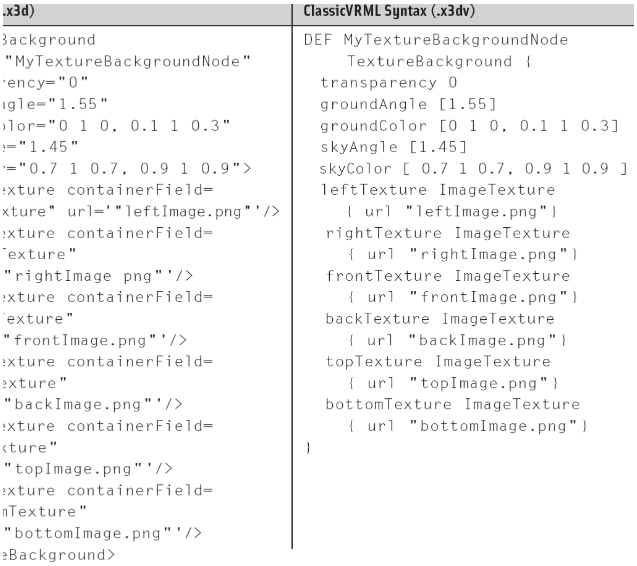
TextureBackground was introduced in X3D version 3.0. Nevertheless, usage under Immersive profile requires addition of

```
<component name='EnvironmentalEffects' level='3'/>
```

http://x3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/TextureBackground-KelpForestMain.x3d

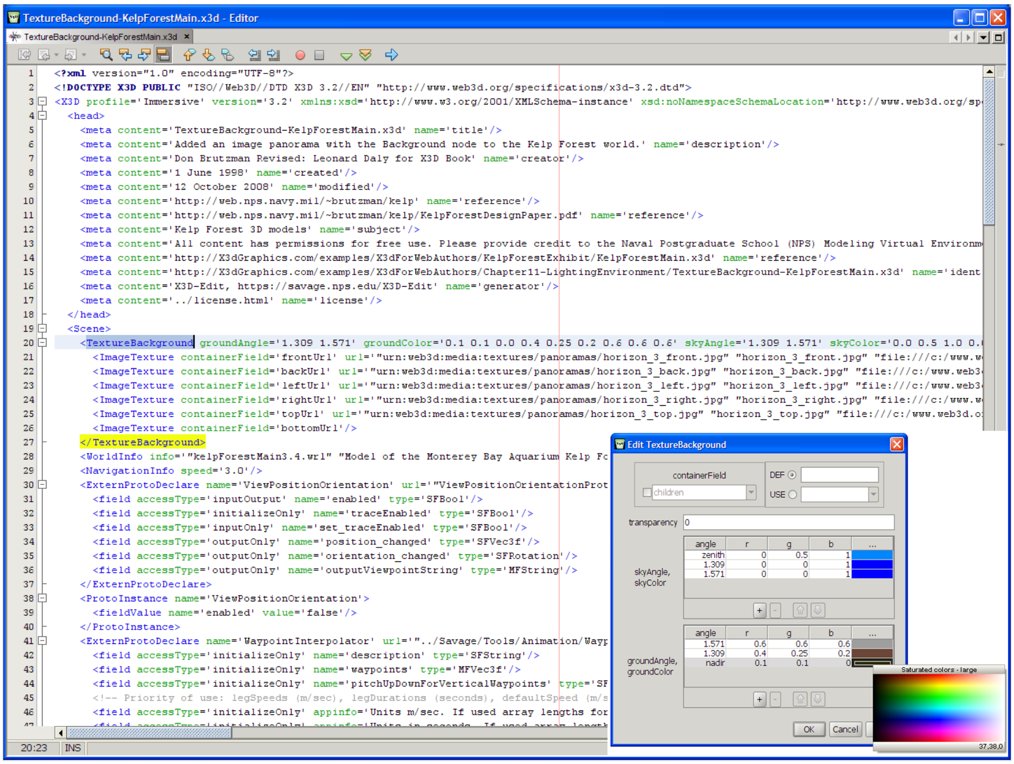
# TextureBackground syntax comparison

Note differences in use of *containerField* for XML syntax, versus use of field name for ClassicVRML syntax



| .x3d) | ClassicVRML Syntax (.x3dv) |
|---|---|
| ackground | DEF MyTextureBackgroundNode |
| "MyTextureBackgroundNode" | TextureBackground { |
| ency="0" | transparency 0 |
| gle="1.55" | groundAngle [1.55] |
| lor="0 1 0, 0.1 1 0.3" | groundColor [0 1 0, 0.1 1 0.3] |
| ="1.45" | skyAngle [1.45] |
| ="0.7 1 0.7, 0.9 1 0.9"> | skyColor [ 0.7 1 0.7, 0.9 1 0.9 ] |
| xture containerField= | leftTexture ImageTexture |
| xture" url='"leftImage.png"'/> | { url "leftImage.png"} |
| xture containerField= | rightTexture ImageTexture |
| exture" | { url "rightImage.png"} |
| "rightImage png"'/> | frontTexture ImageTexture |
| xture containerField= | { url "frontImage.png"} |
| exture" | backTexture ImageTexture |
| "frontImage.png"'/> | { url "backImage.png"} |
| xture containerField= | topTexture ImageTexture |
| xture" | { url "topImage.png"} |
| "backImage.png"'/> | bottomTexture ImageTexture |
| xture containerField= | { url "bottomImage.png"} |
| ture" | } |
| "topImage.png"'/> | |
| xture containerField= | |
| nTexture" | |
| "bottomImage.png"'/> | |
| Background> | |

Page 323, Table 11.13. Node Syntax for TextureBackground



TextureBackground-KelpForestMain.x3d - Editor
TextureBackground-KelpForestMain.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='TextureBackground-KelpForestMain.x3d' name='title'/>
    <meta content='Added an image panorama with the Background node to the Kelp Forest world.' name='description'/>
    <meta content='Don Brutzman Revised: Leonard Daly for X3D Book' name='creator'/>
    <meta content='1 June 1998' name='created'/>
    <meta content='12 October 2008' name='modified'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp' name='reference'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp/KelpForestDesignPaper.pdf' name='reference'/>
    <meta content='Kelp Forest 3D models' name='subject'/>
    <meta content='All content has permissions for free use. Please provide credit to the Naval Postgraduate School (NPS) Modeling Virtual Environm
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/TextureBackground-KelpForestMain.x3d' name='ident
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <TextureBackground groundAngle='1.309 1.571' groundColor='0.1 0.1 0.0 0.4 0.25 0.2 0.6 0.6 0.6' skyAngle='1.309 1.571' skyColor='0.0 0.5 1.0 0.
      <ImageTexture containerField='frontUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_front.jpg" "horizon_3_front.jpg" "file:///c:/www.w
      <ImageTexture containerField='backUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_back.jpg" "horizon_3_back.jpg" "file:///c:/www.web3
      <ImageTexture containerField='leftUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_left.jpg" "horizon_3_left.jpg" "file:///c:/www.web3
      <ImageTexture containerField='rightUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_right.jpg" "horizon_3_right.jpg" "file:///c:/www.w
      <ImageTexture containerField='topUrl' url='"urn:web3d:media:textures/panoramas/horizon_3_top.jpg" "horizon_3_top.jpg" "file:///c:/www.web3d.o
      <ImageTexture containerField='bottomUrl'/>
    </TextureBackground>
    <WorldInfo info='"kelpForestMain3.4.wrl" "Model of the Monterey Bay Aquarium Kelp Fo
    <NavigationInfo speed='3.0'/>
    <ExternProtoDeclare name='ViewPositionOrientation' url='"ViewPositionOrientationProt
      <field accessType='inputOutput' name='enabled' type='SFBool'/>
      <field accessType='initializeOnly' name='traceEnabled' type='SFBool'/>
      <field accessType='inputOnly' name='set_traceEnabled' type='SFBool'/>
      <field accessType='outputOnly' name='position_changed' type='SFVec3f'/>
      <field accessType='outputOnly' name='orientation_changed' type='SFRotation'/>
      <field accessType='outputOnly' name='outputViewpointString' type='MFString'/>
    </ExternProtoDeclare>
    <ProtoInstance name='ViewPositionOrientation'>
      <fieldValue name='enabled' value='false'/>
    </ProtoInstance>
    <ExternProtoDeclare name='WaypointInterpolator' url='"../Savage/Tools/Animation/Wayp
      <field accessType='initializeOnly' name='description' type='SFString'/>
      <field accessType='initializeOnly' name='waypoints' type='MFVec3f'/>
      <field accessType='initializeOnly' name='pitchUpDownForVerticalWaypoints' type='SF
      <!-- Priority of use: legSpeeds (m/sec), legDurations (seconds), defaultSpeed (m/s
      <field accessType='initializeOnly' appinfo='Units m/sec. If used array lengths for
```

Edit TextureBackground

containerField ☐ children

DEF ●

USE ○

transparency 0

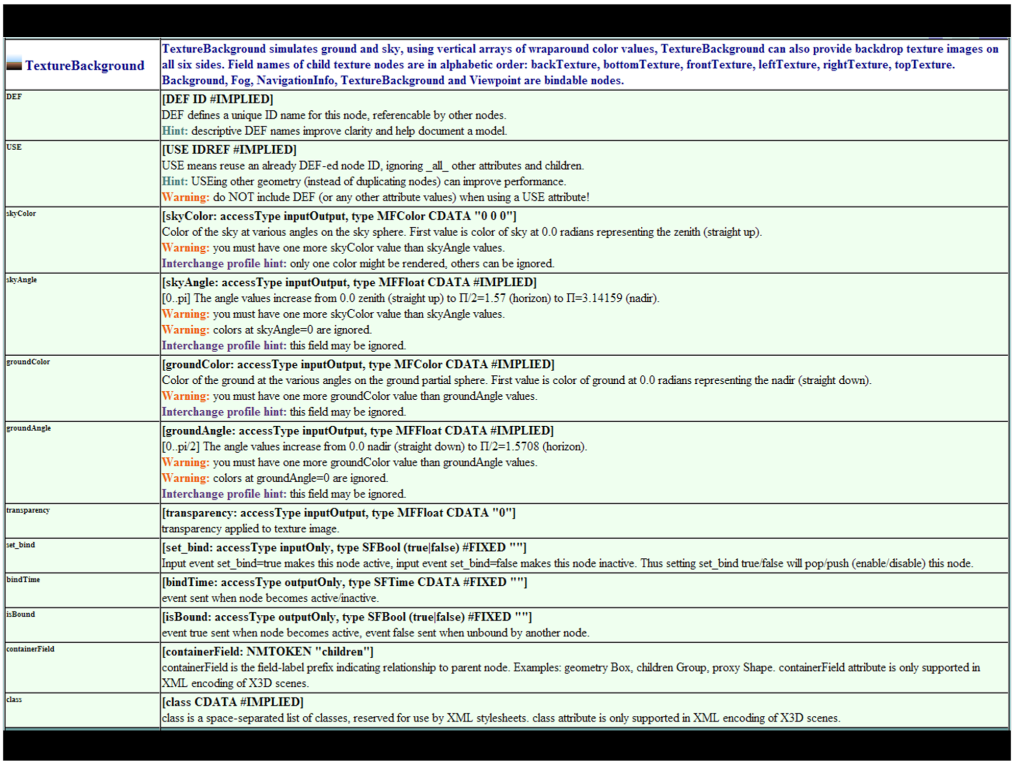
skyAngle, skyColor

| angle | r | g | b | ... |
|---|---|---|---|---|
| zenith | 0 | 0.5 | 1 | 🟦 |
| 1.309 | 0 | 0 | 1 | 🟦 |
| 1.571 | 0 | 0 | 1 | 🟦 |

groundAngle, groundColor

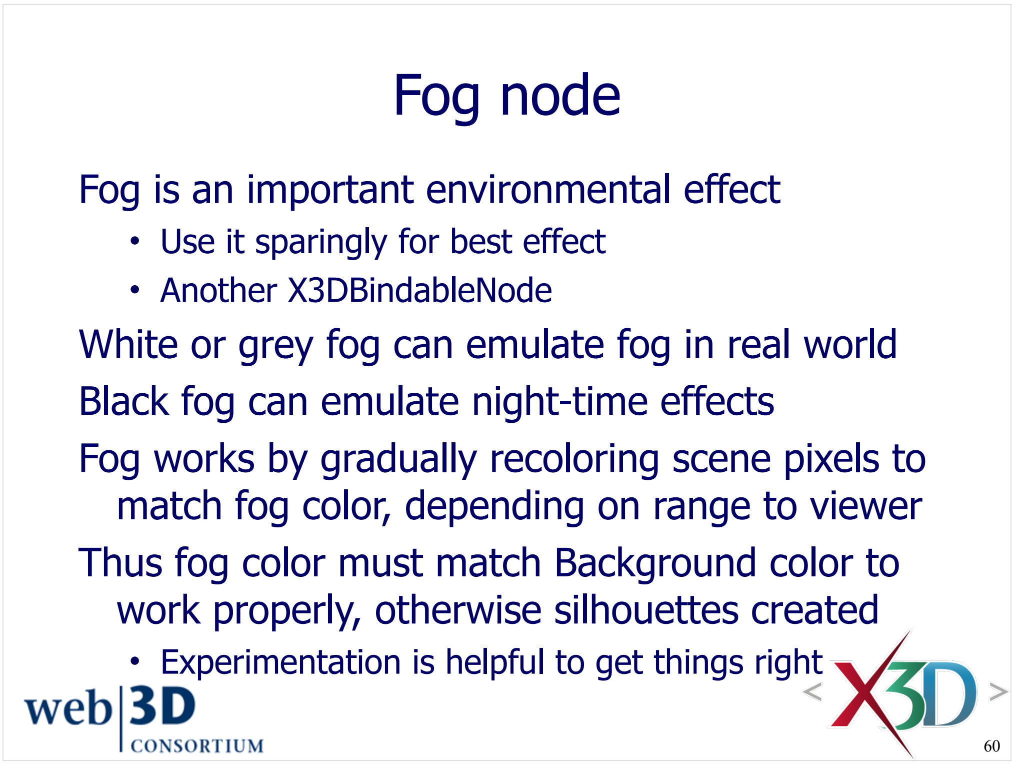
| angle | r | g | b | ... |
|---|---|---|---|---|
| 1.571 | 0.6 | 0.6 | 0.6 | ⬜ |
| 1.309 | 0.4 | 0.25 | 0.2 | 🟫 |
| nadir | 0.1 | 0.1 | 0 | ⬛ |

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/TextureBackground-KelpForestMain.x3d



| TextureBackground | TextureBackground simulates ground and sky, using vertical arrays of wraparound color values, TextureBackground can also provide backdrop texture images on all six sides. Field names of child texture nodes are in alphabetic order: backTexture, bottomTexture, frontTexture, leftTexture, rightTexture, topTexture. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes. |
|---|---|
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>Hint: descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.<br>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute! |
| skyColor | **[skyColor: accessType inputOutput, type MFColor CDATA "0 0 0"]**<br>Color of the sky at various angles on the sky sphere. First value is color of sky at 0.0 radians representing the zenith (straight up).<br>Warning: you must have one more skyColor value than skyAngle values.<br>Interchange profile hint: only one color might be rendered, others can be ignored. |
| skyAngle | **[skyAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi] The angle values increase from 0.0 zenith (straight up) to Π/2=1.57 (horizon) to Π=3.14159 (nadir).<br>Warning: you must have one more skyColor value than skyAngle values.<br>Warning: colors at skyAngle=0 are ignored.<br>Interchange profile hint: this field may be ignored. |
| groundColor | **[groundColor: accessType inputOutput, type MFColor CDATA #IMPLIED]**<br>Color of the ground at the various angles on the ground partial sphere. First value is color of ground at 0.0 radians representing the nadir (straight down).<br>Warning: you must have one more groundColor value than groundAngle values.<br>Interchange profile hint: this field may be ignored. |
| groundAngle | **[groundAngle: accessType inputOutput, type MFFloat CDATA #IMPLIED]**<br>[0..pi/2] The angle values increase from 0.0 nadir (straight down) to Π/2=1.5708 (horizon).<br>Warning: you must have one more groundColor value than groundAngle values.<br>Warning: colors at groundAngle=0 are ignored.<br>Interchange profile hint: this field may be ignored. |
| transparency | **[transparency: accessType inputOutput, type MFFloat CDATA "0"]**<br>transparency applied to texture image. |
| set_bind | **[set_bind: accessType inputOnly, type SFBool (true\|false) #FIXED ""]**<br>Input event set_bind=true makes this node active, input event set_bind=false makes this node inactive. Thus setting set_bind true/false will pop/push (enable/disable) this node. |
| bindTime | **[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]**<br>event sent when node becomes active/inactive. |
| isBound | **[isBound: accessType outputOnly, type SFBool (true\|false) #FIXED ""]**<br>event true sent when node becomes active, event false sent when unbound by another node. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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. |

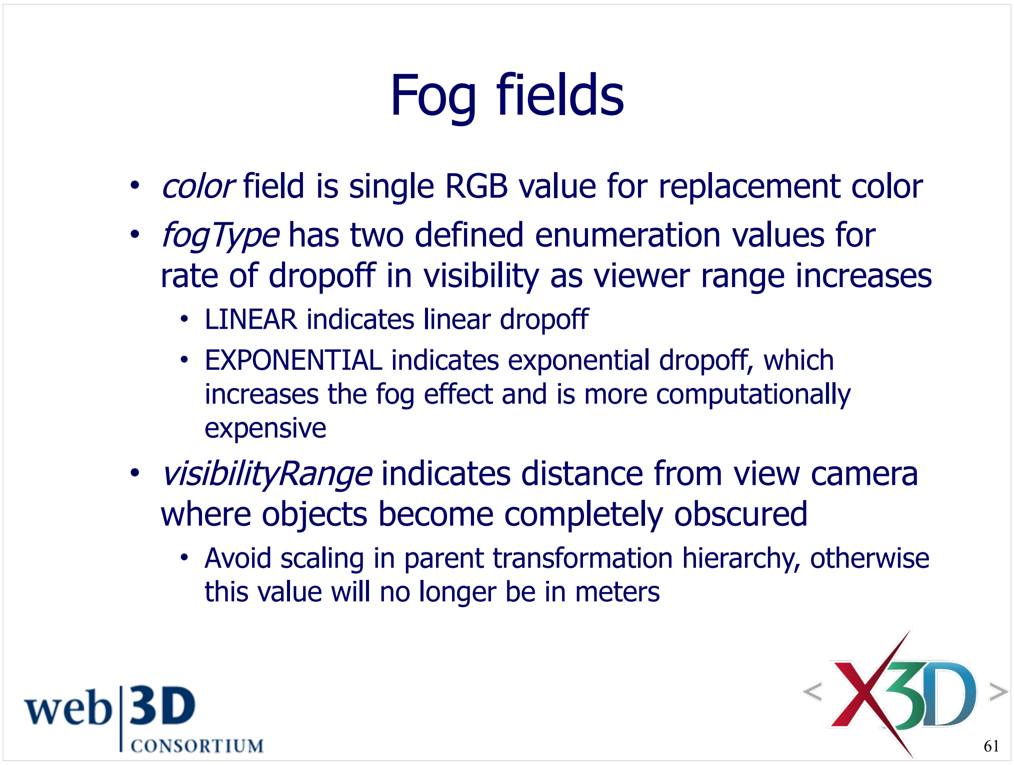
http://www.web3d.org/x3d/content/X3dTooltips.html#TextureBackground



# Fog node

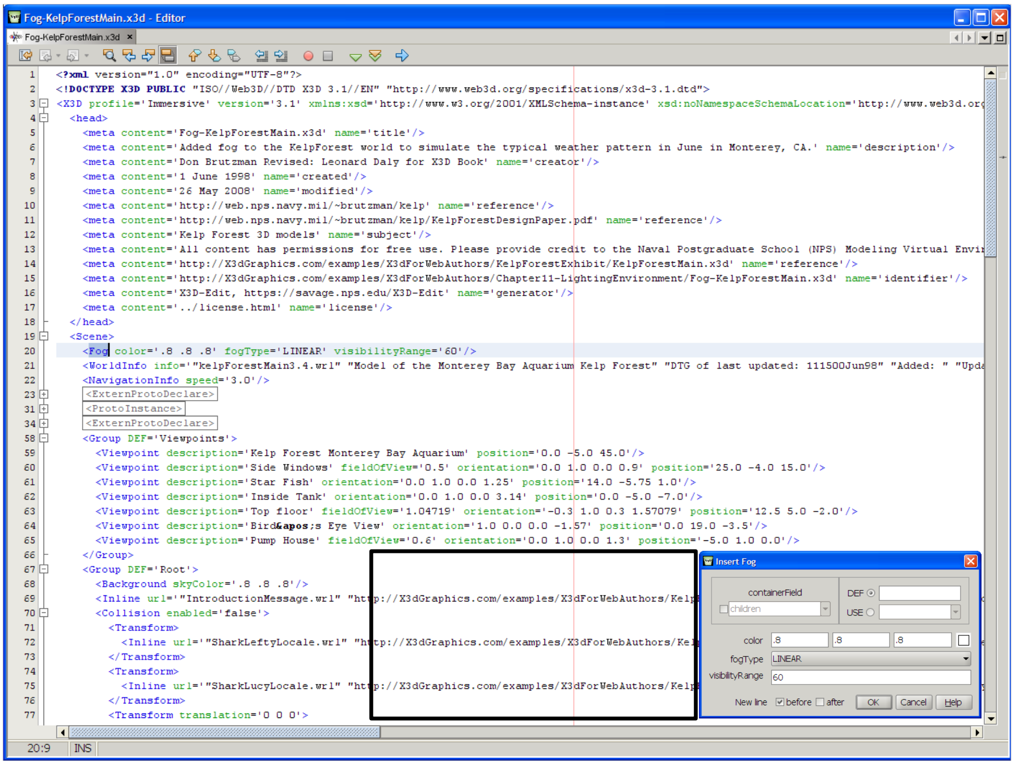
- Fog is an important environmental effect
  - Use it sparingly for best effect
  - Another X3DBindableNode
- White or grey fog can emulate fog in real world
- Black fog can emulate night-time effects
- Fog works by gradually recoloring scene pixels to match fog color, depending on range to viewer
- Thus fog color must match Background color to work properly, otherwise silhouettes created
  - Experimentation is helpful to get things right

Constraint:  Fog nodes within Inline scenes are not automatically bound when loaded, even if they are the only Fog nodes available in the scene.



# Fog fields

- *color* field is single RGB value for replacement color
- *fogType* has two defined enumeration values for rate of dropoff in visibility as viewer range increases
  - LINEAR indicates linear dropoff
  - EXPONENTIAL indicates exponential dropoff, which increases the fog effect and is more computationally expensive
- *visibilityRange* indicates distance from view camera where objects become completely obscured
  - Avoid scaling in parent transformation hierarchy, otherwise this value will no longer be in meters



```
<?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
  <head>
    <meta content='Fog-KelpForestMain.x3d' name='title'/>
    <meta content='Added fog to the KelpForest world to simulate the typical weather pattern in June in Monterey, CA.' name='description'/>
    <meta content='Don Brutzman Revised: Leonard Daly for X3D Book' name='creator'/>
    <meta content='1 June 1998' name='created'/>
    <meta content='26 May 2008' name='modified'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp' name='reference'/>
    <meta content='http://web.nps.navy.mil/~brutzman/kelp/KelpForestDesignPaper.pdf' name='reference'/>
    <meta content='Kelp Forest 3D models' name='subject'/>
    <meta content='All content has permissions for free use. Please provide credit to the Naval Postgraduate School (NPS) Modeling Virtual Envi
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d' name='reference'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/Fog-KelpForestMain.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <Fog color='.8 .8 .8' fogType='LINEAR' visibilityRange='60'/>
    <WorldInfo info='"kelpForestMain3.4.wrl" "Model of the Monterey Bay Aquarium Kelp Forest" "DTG of last updated: 111500Jun98" "Added: " "Upda
    <NavigationInfo speed='3.0'/>
    <ExternProtoDeclare>
    <ProtoInstance>
    <ExternProtoDeclare>
    <Group DEF='Viewpoints'>
      <Viewpoint description='Kelp Forest Monterey Bay Aquarium' position='0.0 -5.0 45.0'/>
      <Viewpoint description='Side Windows' fieldOfView='0.5' orientation='0.0 1.0 0.0 0.9' position='25.0 -4.0 15.0'/>
      <Viewpoint description='Star Fish' orientation='0.0 1.0 0.0 1.25' position='14.0 -5.75 1.0'/>
      <Viewpoint description='Inside Tank' orientation='0.0 1.0 0.0 3.14' position='0.0 -5.0 -7.0'/>
      <Viewpoint description='Top floor' fieldOfView='1.04719' orientation='-0.3 1.0 0.3 1.57079' position='12.5 5.0 -2.0'/>
      <Viewpoint description='Bird&apos;s Eye View' orientation='1.0 0.0 0.0 -1.57' position='0.0 19.0 -3.5'/>
      <Viewpoint description='Pump House' fieldOfView='0.6' orientation='0.0 1.0 0.0 1.3' position='-5.0 1.0 0.0'/>
    </Group>
    <Group DEF='Root'>
      <Background skyColor='.8 .8 .8'/>
      <Inline url='"IntroductionMessage.wrl" "http://X3dGraphics.com/examples/X3dForWebAuthors/Kelp
      <Collision enabled='false'>
        <Transform>
          <Inline url='"SharkLeftyLocale.wrl" "http://X3dGraphics.com/examples/X3dForWebAuthors/Kel
        </Transform>
        <Transform>
          <Inline url='"SharkLucyLocale.wrl" "http://X3dGraphics.com/examples/X3dForWebAuthors/Kelp
        </Transform>
        <Transform translation='0 0 0'>
```

Insert Fog

containerField ☐ children

DEF ●

USE ○

color .8 .8 .8 □

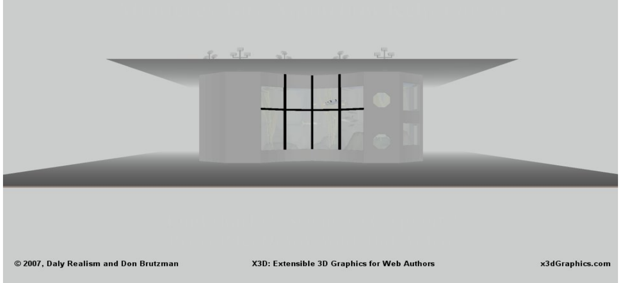
fogType LINEAR

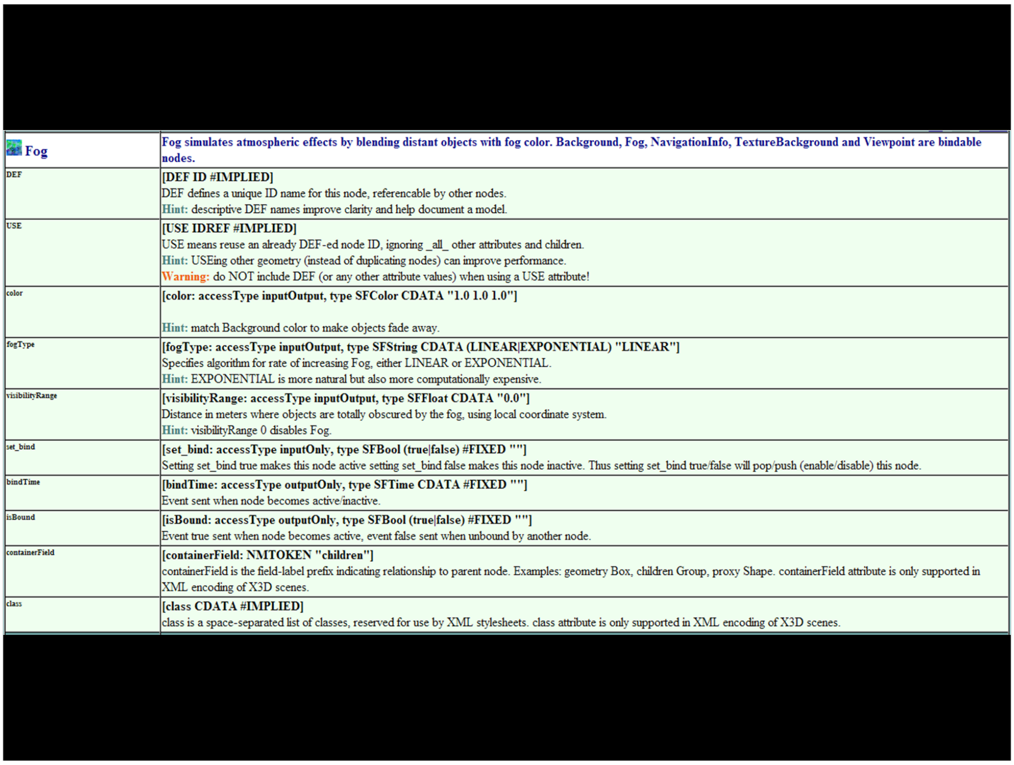
visibilityRange 60

New line ☑ before ☐ after

Figure 11.10. A Fog node blends geometry colors and textures to the fog color.

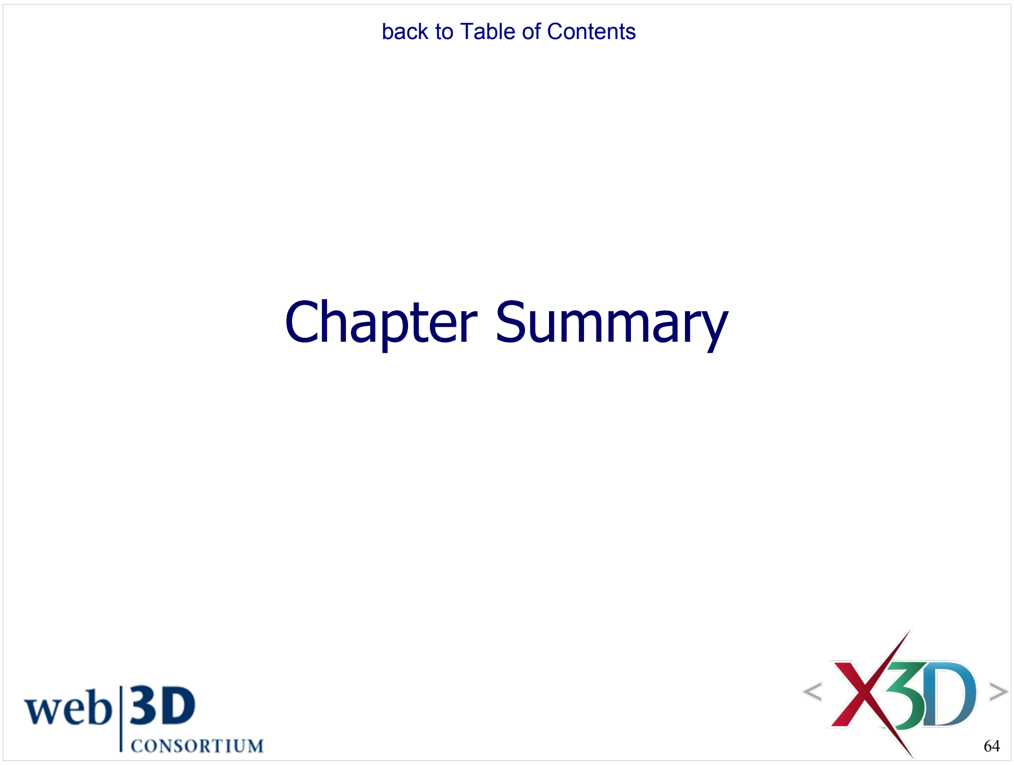
http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter11-LightingEnvironment/Fog-KelpForestMain.x3d


© 2007, Daly Realism and Don Brutzman
X3D: Extensible 3D Graphics for Web Authors
x3dGraphics.com



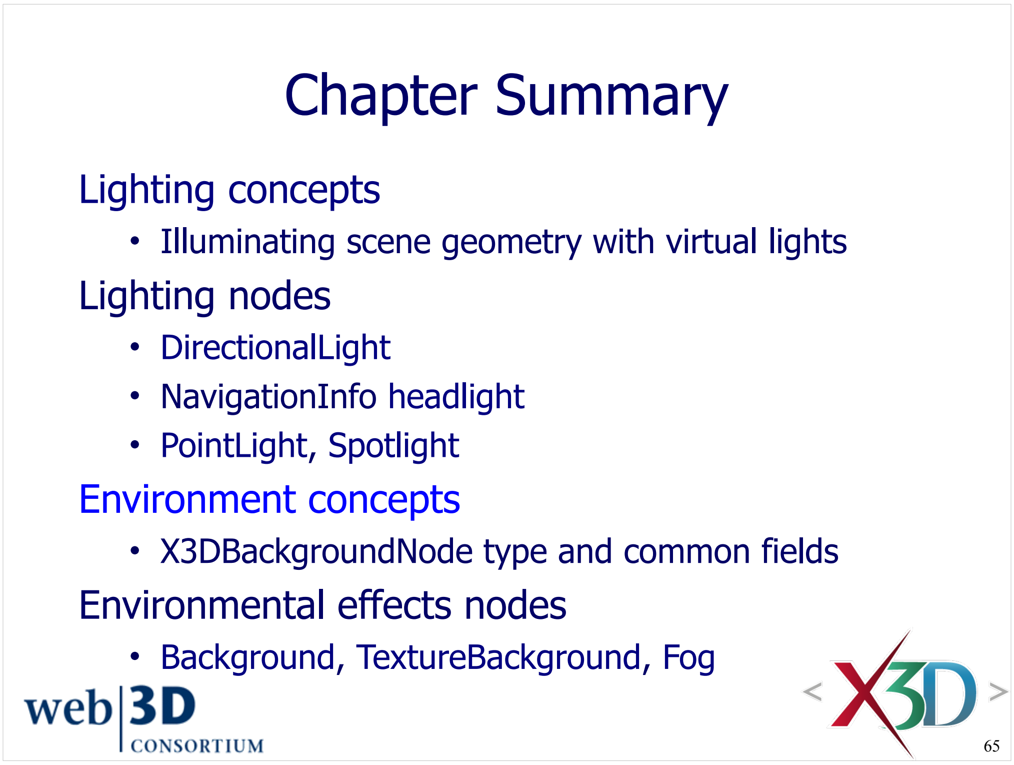
| Fog | Fog simulates atmospheric effects by blending distant objects with fog color. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes. |
|---|---|
| DEF | **[DEF ID #IMPLIED]**<br>DEF defines a unique ID name for this node, referencable by other nodes.<br>Hint: descriptive DEF names improve clarity and help document a model. |
| USE | **[USE IDREF #IMPLIED]**<br>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.<br>Hint: USEing other geometry (instead of duplicating nodes) can improve performance.<br>Warning: do NOT include DEF (or any other attribute values) when using a USE attribute! |
| color | **[color: accessType inputOutput, type SFColor CDATA "1.0 1.0 1.0"]**<br>Hint: match Background color to make objects fade away. |
| fogType | **[fogType: accessType inputOutput, type SFString CDATA (LINEAR\|EXPONENTIAL) "LINEAR"]**<br>Specifies algorithm for rate of increasing Fog, either LINEAR or EXPONENTIAL.<br>Hint: EXPONENTIAL is more natural but also more computationally expensive. |
| visibilityRange | **[visibilityRange: accessType inputOutput, type SFFloat CDATA "0.0"]**<br>Distance in meters where objects are totally obscured by the fog, using local coordinate system.<br>Hint: visibilityRange 0 disables Fog. |
| set_bind | **[set_bind: accessType inputOnly, type SFBool (true\|false) #FIXED ""]**<br>Setting set_bind true makes this node active setting set_bind false makes this node inactive. Thus setting set_bind true/false will pop/push (enable/disable) this node. |
| bindTime | **[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]**<br>Event sent when node becomes active/inactive. |
| isBound | **[isBound: accessType outputOnly, type SFBool (true\|false) #FIXED ""]**<br>Event true sent when node becomes active, event false sent when unbound by another node. |
| containerField | **[containerField: NMTOKEN "children"]**<br>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]**<br>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#Fog



back to Table of Contents

# Chapter Summary



# Chapter Summary

Lighting concepts

- Illuminating scene geometry with virtual lights

Lighting nodes

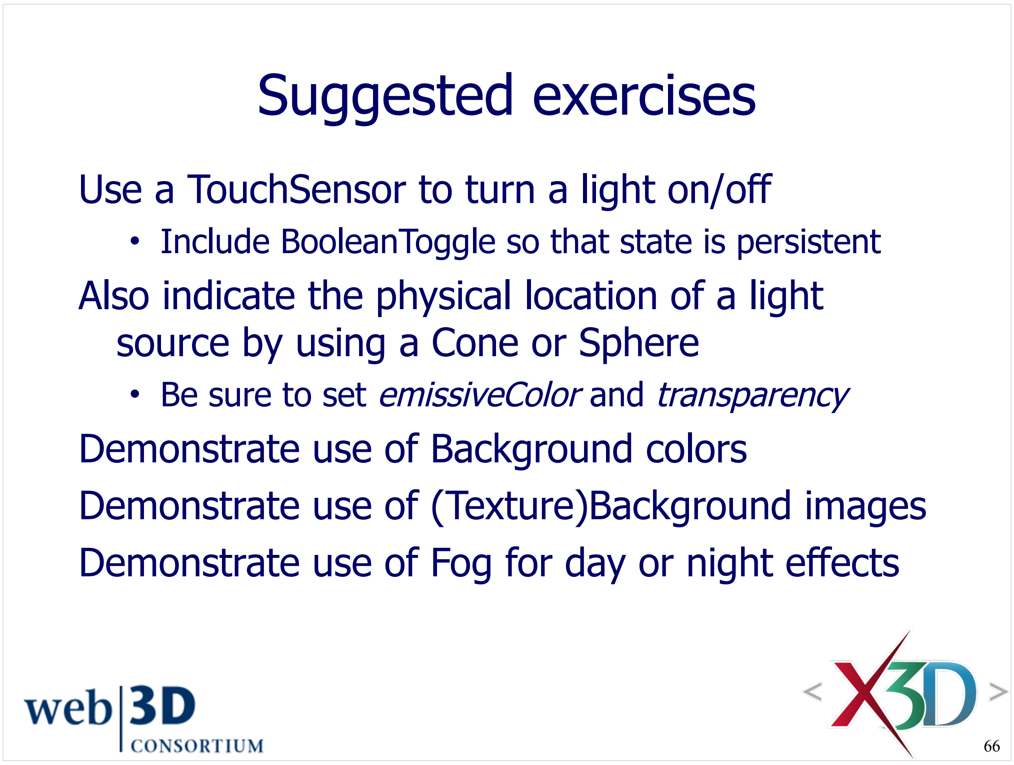
- DirectionalLight
- NavigationInfo headlight
- PointLight, Spotlight

Environment concepts

- X3DBackgroundNode type and common fields

Environmental effects nodes

- Background, TextureBackground, Fog



# Suggested exercises

Use a TouchSensor to turn a light on/off

- Include BooleanToggle so that state is persistent

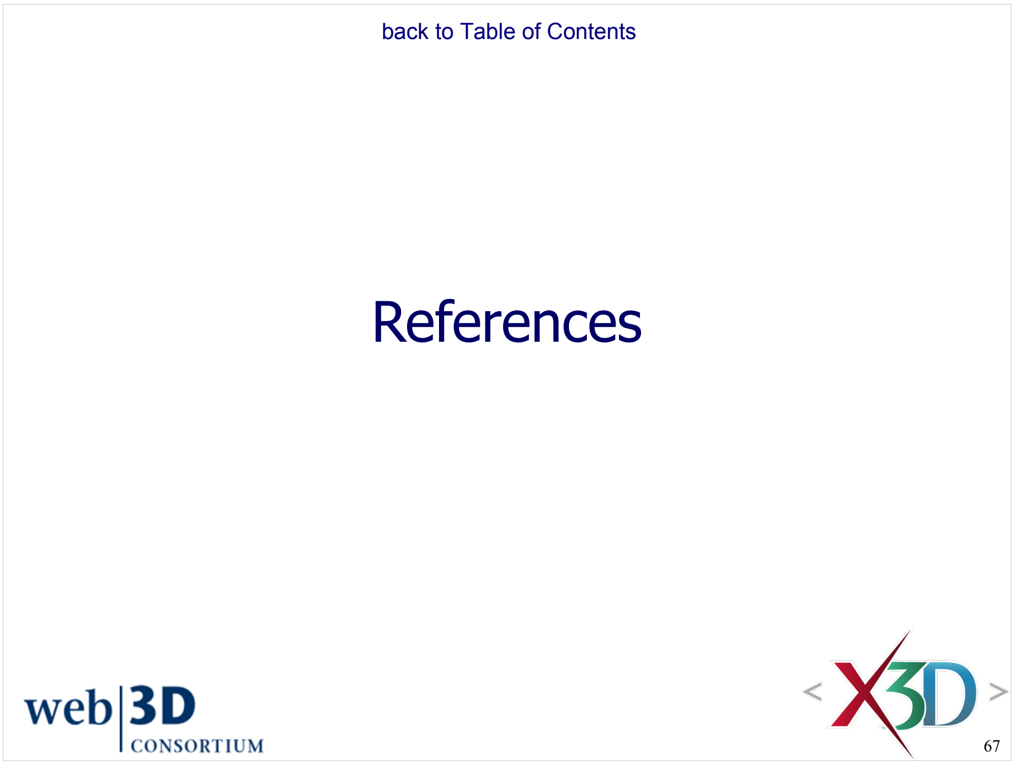
Also indicate the physical location of a light source by using a Cone or Sphere

- Be sure to set *emissiveColor* and *transparency*

Demonstrate use of Background colors

Demonstrate use of (Texture)Background images

Demonstrate use of Fog for day or night effects



back to Table of Contents

# References

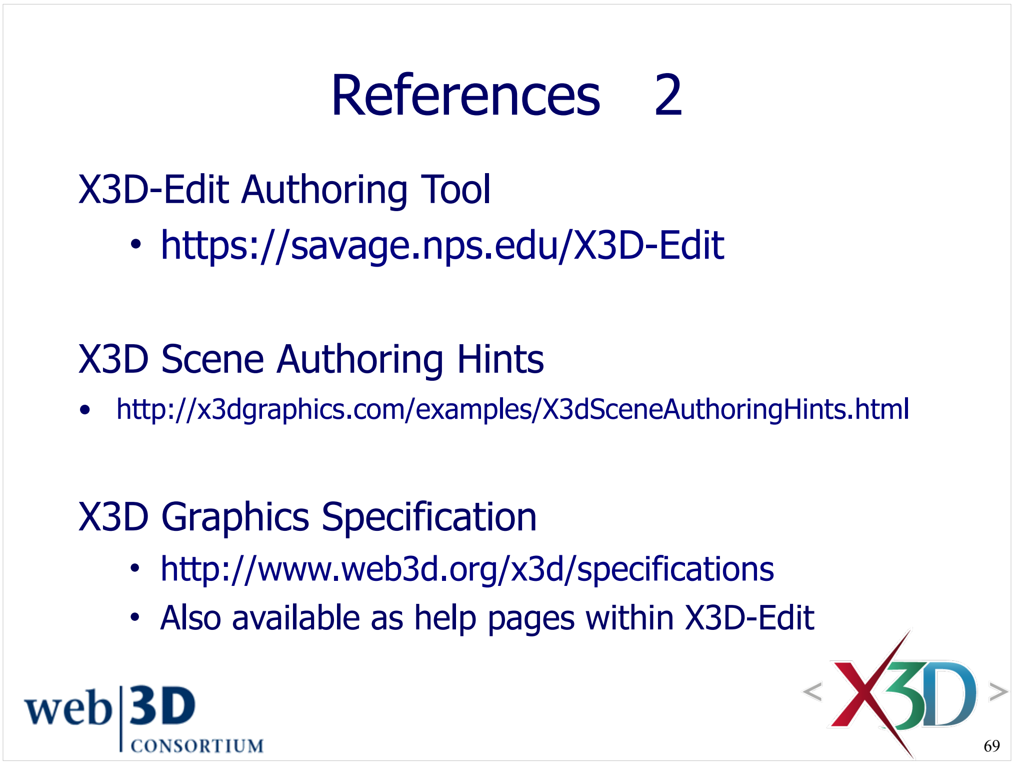
# References 1

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

- Chapter 11, Lighting and Environment Nodes
- http://x3dGraphics.com
- http://x3dgraphics.com/examples/X3dForWebAuthors

X3D Resources

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



# References 2

X3D-Edit Authoring Tool

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

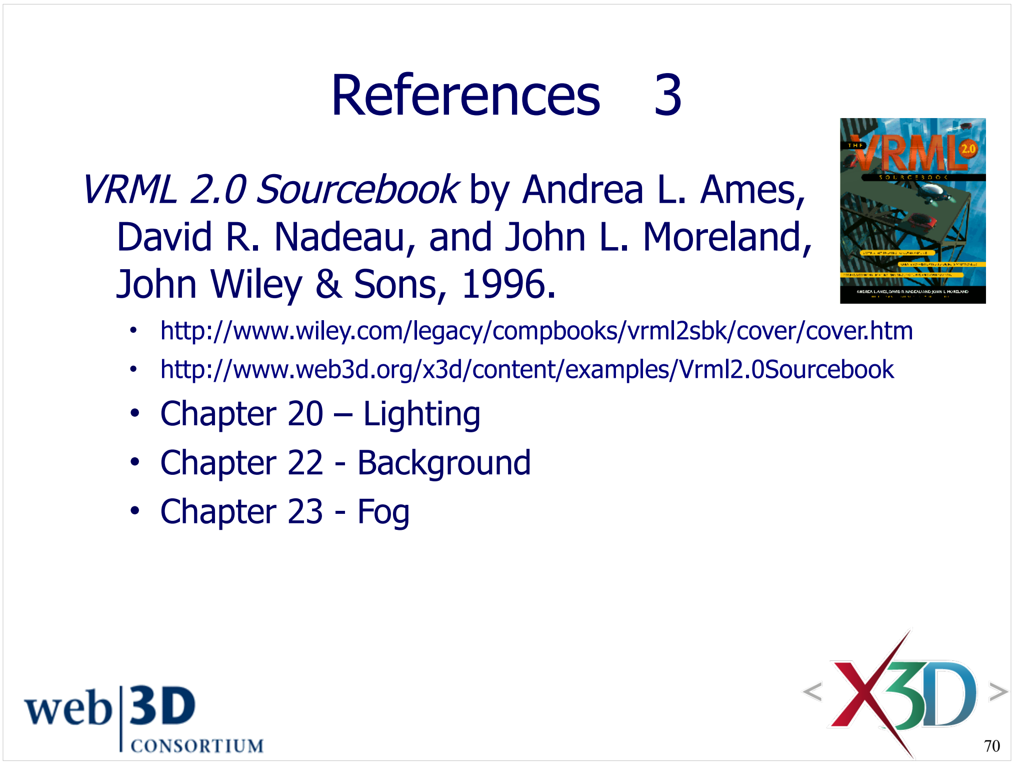
X3D Scene Authoring Hints

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

X3D Graphics Specification

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

69

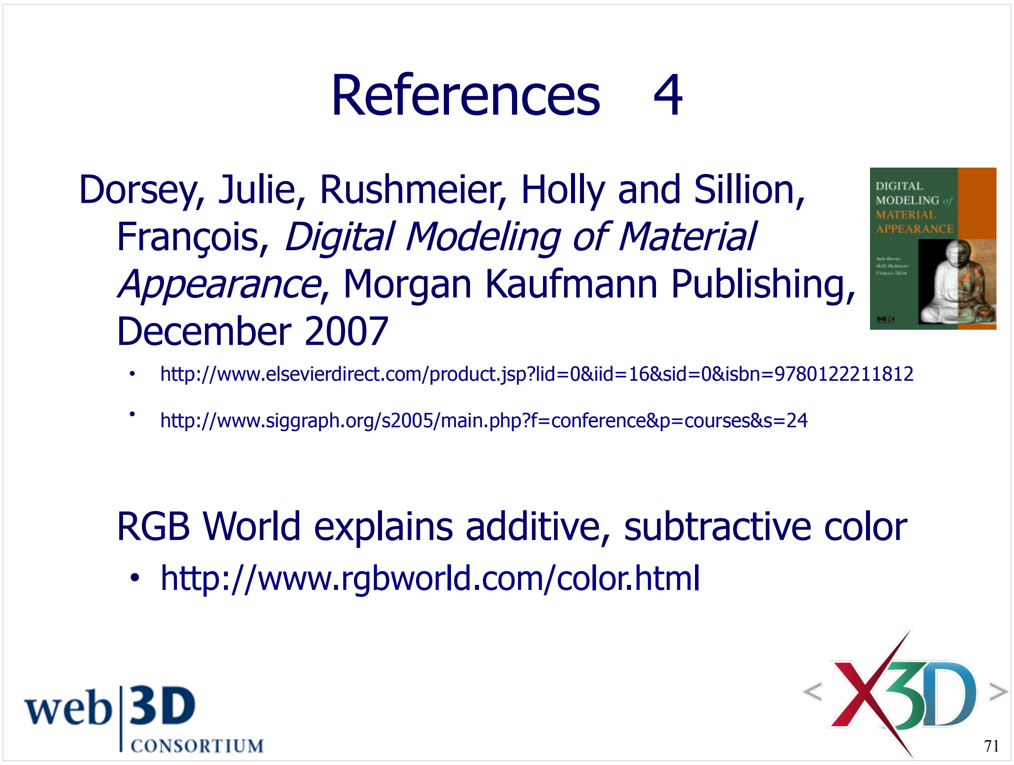


# References 3

*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 20 – Lighting
- Chapter 22 - Background
- Chapter 23 - Fog

web3D CONSORTIUM
<X3D>
70



DIGITAL
MODELING of
MATERIAL
APPEARANCE

# References 4

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

RGB World explains additive, subtractive color

- http://www.rgbworld.com/color.html

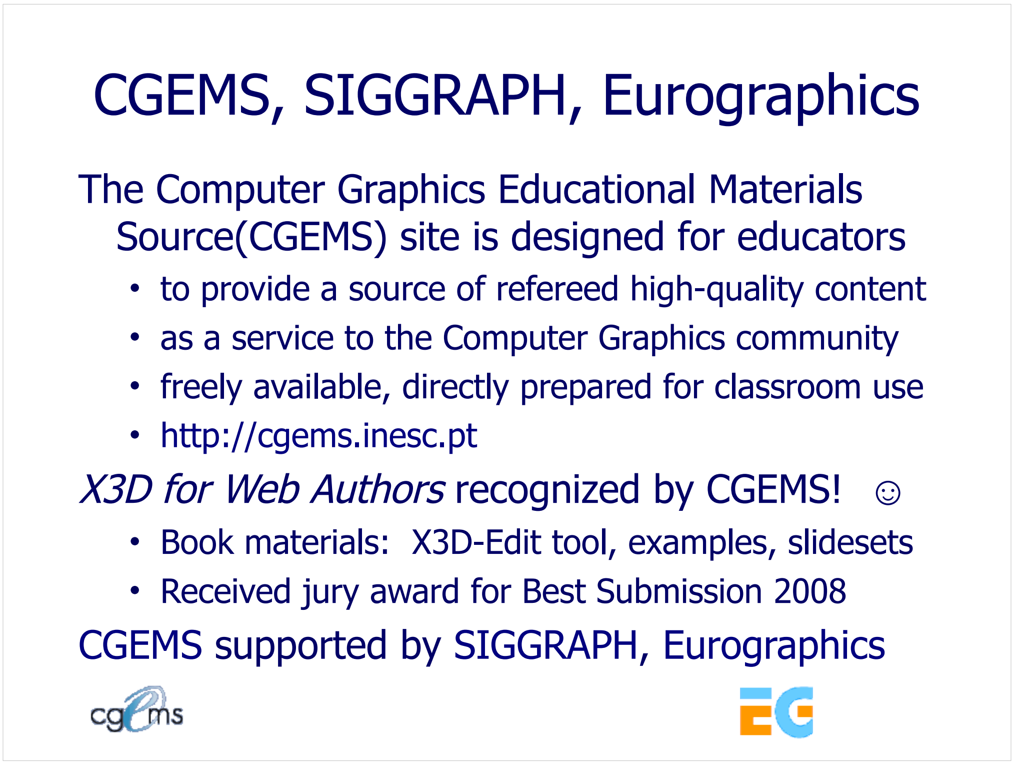
# Contact

**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:

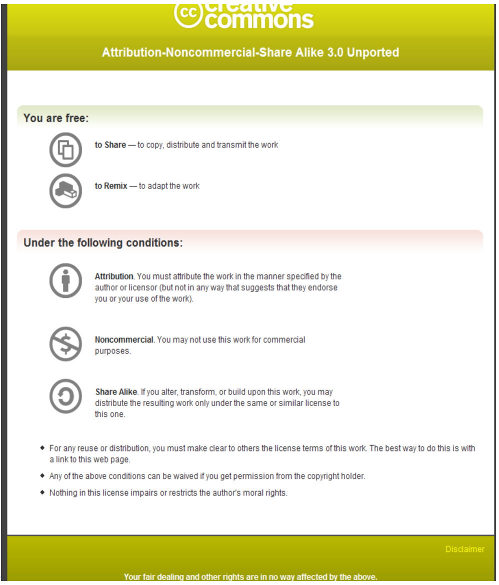
- http://cgems.inesc.pt/authors/ListModules.aspx

CGEMS Editorial Policy:

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

## Creative Commons open-source license

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



creative commons

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.

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.

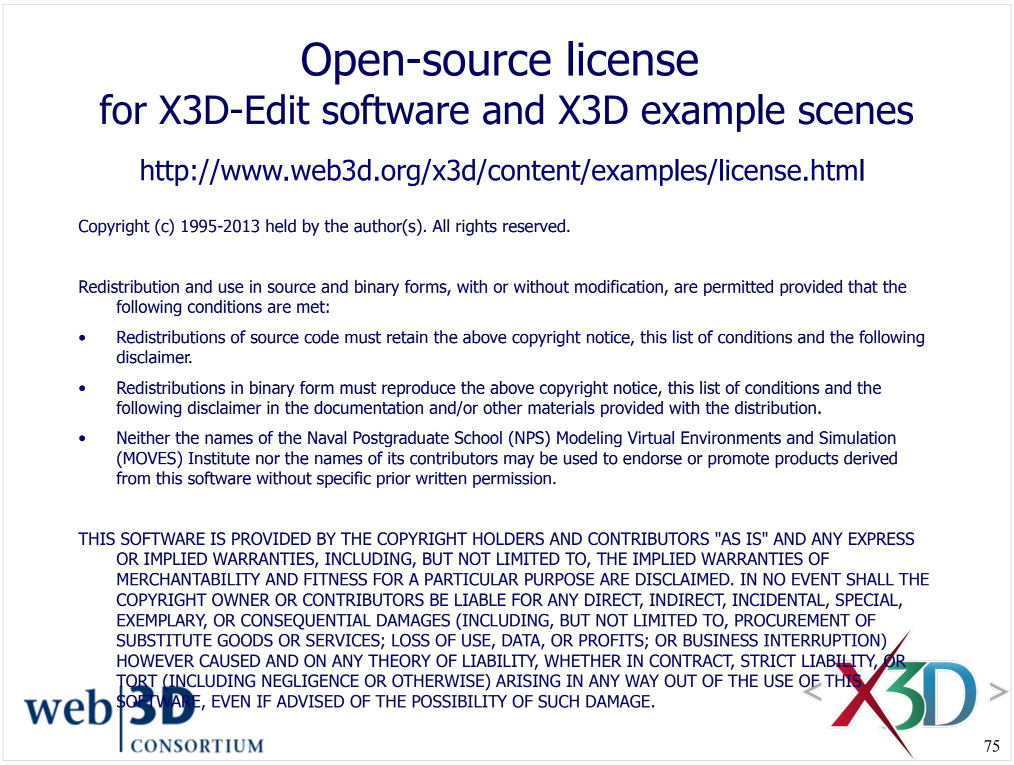
* Noncommercial. You may not use this work for commercial purposes.

* 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.

75

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*, Oreilly 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