

# X3D Graphics for Web Authors

## Chapter 4

### Viewing and Navigation

*But the eyes, though they are no sailors, will never be satisfied with any model, however fashionable, which does not answer all the requisitions of art.*

Henry David Thoreau, 1849



# Contents

Chapter Overview and Concepts

X3D Nodes and Examples

Additional Resources

Chapter Summary

References



# Chapter Overview



# Overview: Viewing and Navigation

Users explore X3D worlds by choosing predefined viewpoints and **navigating** through 3D space.

- **Viewpoint** lets authors identify key camera locations
- **NavigationInfo** provides options for moving around
- **Bindable nodes**, so only one is active at a time

Related nodes improve navigability, interaction

- **Anchor** makes geometric shapes linkable
- **Billboard** keeps child geometry facing the user
- **Collision** can allow or prevent a user's view from passing through geometry



[back to Table of Contents](#)

# Concepts



# Viewing and navigation

It is helpful to think of X3D scenes as fixed at different locations in 3D space

- Viewpoints are like cameras, prepositioned in locations (and directions) of interest
- Users can move their current camera viewpoint further and change direction they are looking at
- This process is called *navigation*

Making navigation easy for users is important

- Authors provide viewpoints of interest with scenes
- Browsers enable camera rotation, pan, zoom, etc.



# Goals of viewing and navigation

- Viewing a scene from different vantage points that reveal aspects of interest, document key locations, or help to tell a story
- Navigating changes in the user's viewpoint effectively by moving from place to place in an intuitive manner
- Making geometric objects selectable so that users can transport to another viewpoint, launch into another scene, or receive other web content
- Taking advantage of viewpoint location for special interactive techniques, such as user-facing billboard rotations and terrain following



# Bindable nodes

Bindable nodes have a special property:  
only one can be active at a time

- Viewpoint, NavigationInfo, Background, TextureBackground, Fog
- Each implements X3DBindableNode type interface for consistency

Implemented using a stack

- Similar to spring-loaded tray of plates in cafeteria
- One (and only one) is active, on top
- One can be pulled off top, sent off to the side
- One can be pulled to top, pushing down others



# Binding example

Basic user operation is pretty simple:

- just select the desired Viewpoint

Complex example follows, stepping through binding stack operations

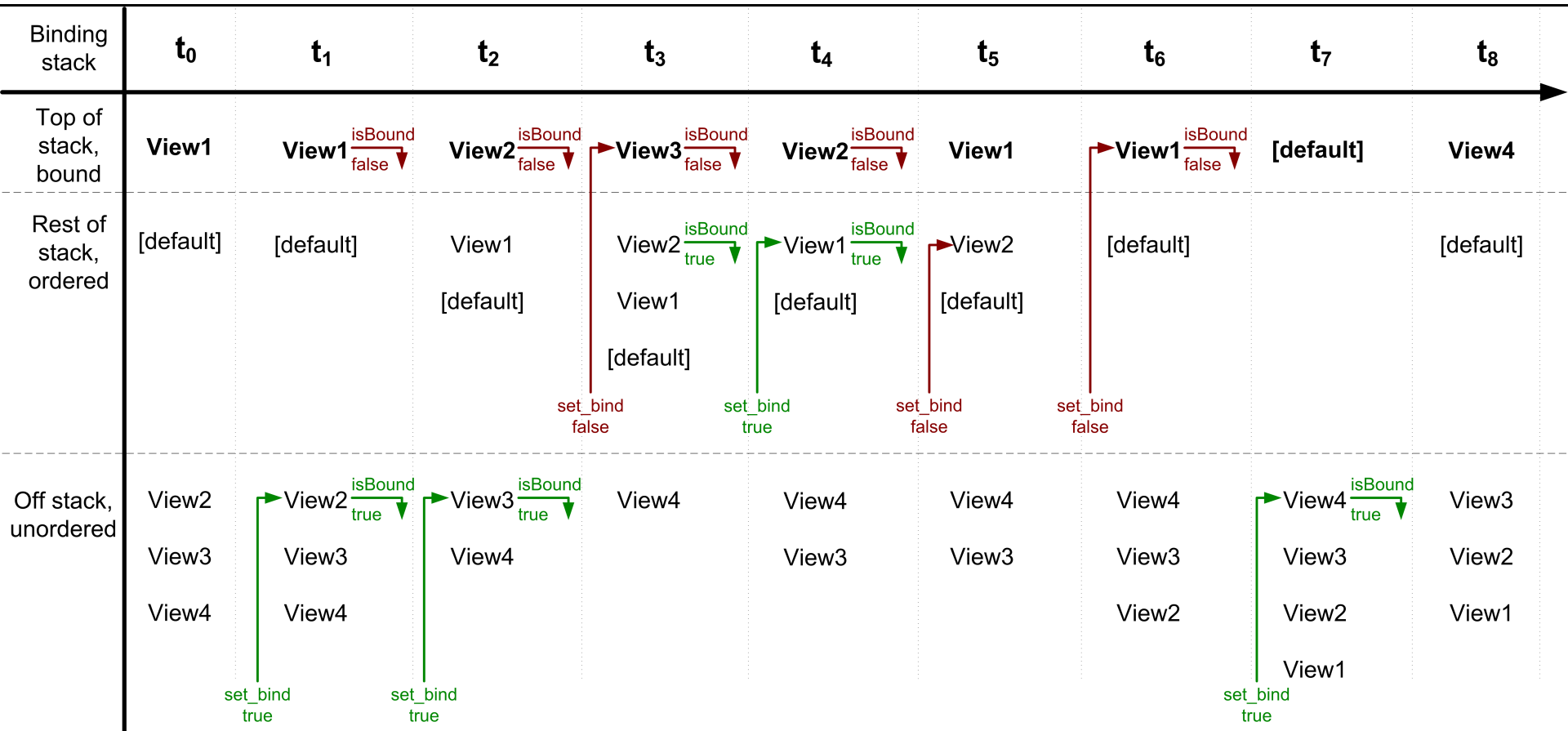
- Advanced details
- `BindingOperations.x3d`
- Animated with scripting
- Console results found in `BindingOperations.console.txt`



New users please skip ahead to **Nodes and Examples**



# Binding node operations diagram



Viewpoints are activated (bound) upon selection,  
ordering is governed by stack operations



# Binding node operations 1

- **Time t0.** The initial loading of the scene has first `<Viewpoint DEF='View1' />` active and bound to the top of the binding stack. Other viewpoints are off the binding stack.
- If no viewpoints are provided in the scene, then the default `<Viewpoint position='0 0 10' />` defined in the X3D Specification is used.
- **Time t1.** When the user selects View2 from the viewpoint list, it receives a *set\_bind="true"* event and goes to the top of the binding stack. View2 also issues an *isBound="true"* event, and View1 issues an *isBound="false"* event as it moves down the stack.



# Binding node operations 2

- **Time t2.** Similar to the previous transitions in step t1, View3 receives a *set\_bind="true"* event and responds with an *isBound="true"* event, while View2 issues an *isBound="false"* event and pushes View1 further down the stack.
- **Time t3.** View3 receives a *set\_bind="false"* event, triggering a corresponding *isBound="false"* event and dropping off the stack completely. Because View2 is the next node on the binding stack, it pops to the top to become the active Viewpoint node. View2 also issues an *isBound="true"* event.



# Binding node operations 3

- **Time t4.** The user now selects View1 from the browser's viewpoint list, so View1 receives a *set\_bind="true"* event and sends a corresponding *isBound="true"* event. View2 is no longer bound, and is pushed down the binding stack.
- **Time t5.** View2 receives a *set\_bind="false"* event while on the binding stack but unbound, and as a result, it is taken completely off the binding stack.
- **Time t6.** View1 is now removed off the binding stack via a *set\_bind="false"* event, leaving no other defined Viewpoint nodes on the stack.



# Binding node operations 4

- **Time t7.** With no Viewpoint nodes remaining on the stack to bind, default viewpoint values are used: `<Viewpoint position='0 0 10' />`. The user then selects the previously unbound View4 from the viewpoint list.
- **Time t8.** View4 remains as the bound viewpoint with no further viewpoints remaining on the stack.

Final note: same for other X3D bindable nodes

- Viewpoint, NavigationInfo, Background, TextureBackground, Fog



# X3D Nodes and Examples



# Viewpoint node

It is helpful to think of X3D scenes as being fixed solidly in 3D space, positioned and oriented exactly where placed by the scene author

Viewing a scene is thus a matter of navigating the current user point of view through space

Viewpoint nodes let X3D scene authors predefine locations and orientations of particular interest

- Sometimes viewpoints are animated and moving
- Freedom of viewpoint is exciting and engaging, also a major advantage over fixed-viewpoint video



# Viewpoint *position, orientation*

A Viewpoint node defines a specific *position* and *orientation* for looking at a 3D scene

- Similar to a “virtual camera” vantage point

Default Viewpoint *position* is (0 0 10)

- out 10 m on +Z axis, looking back towards origin

Any changes to Viewpoint *orientation* are made relative to that default direction (along -X axis)

- Different initial direction than other orientations
- Visualize the situation and then use right-hand rule to figure out the correct *orientation* value



# Viewpoint *description*

Each Viewpoint is given a *description* string to help users decide which view to select

- Clear, understandable descriptions can guide users
- Use an object's name first when many viewpoints follow, so they are more easily identified in a list
- Use whitespace instead of underscores for better readability

Viewpoints are primary user tool for navigation

- Browsers provide Viewpoint List to show and select descriptions



# Viewpoint *fieldOfView*, *centerOfRotation*

*fieldOfView* describes the angular width shown

- Horizontal breadth displayed
- Default is 45 degrees =  $\pi/4$  radians = 0.785
- Vertical field determined by browser aspect ration

*centerOfRotation* is a local position

- User's current view rotates about this point if the bound NavigationInfo node is in EXAMINE mode
- Can be changed by a user's LOOKAT operation picking some other geometry as new center



# Viewpoint *jump*

*jump* can be a tricky field (but is not often used)

- *jump*='true' when a Viewpoint is selected means that the current view position and orientation is modified according to NavigationInfo transitionType
- *jump*='true' is usual default
- *jump*='false' is an advanced technique
  - User's view doesn't appear to change when new Viewpoint is selected
  - New Viewpoint is bound, but given offsets to match prior user position and orientation (hence no jumping)
  - Example use: changing bound viewpoint when moving from one floor into an elevator, then to another floor



# Viewpoint hints and warnings

Use parent Transform node(s) for complex Viewpoint orientation, position values

- One axis of rotation at a time can work

Keyboard shortcuts are helpful

- PageUp PageDown Home End to select Viewpoint
- Arrow keys to examine (rotate), pan, zoom, etc. depending on current NavigationInfo mode
- Browser may allow Viewpoint reset after navigating

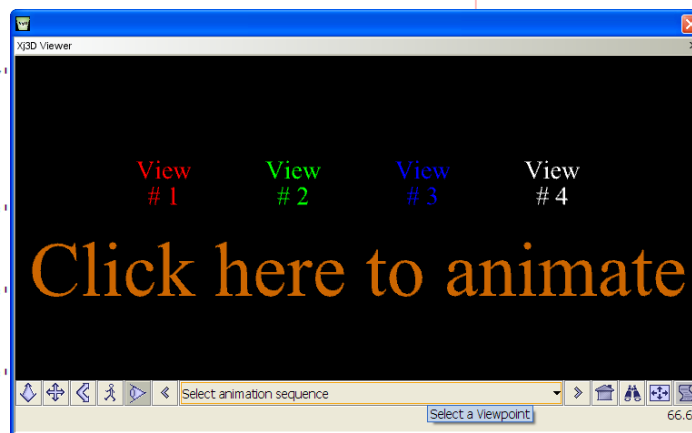
Distinguish between defined Viewpoint and current navigated user-view location, direction



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
3  <X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
4    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
5    <head>
6      <meta content='BindingOperations.x3d' name='title' />
7      <meta content='Illustrate Viewpoint binding operations as described in Chapter 4 concepts.
8        Display the browser console to see an event' name='description' />
9      <meta content='Don Brutzman' name='creator' />
10     <meta content='5 January 2008' name='created' />
11     <meta content='5 January 2008' name='modified' />
12     <meta content='http://X3dGraphics.com' name='reference' />
13     <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
14     <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights' />
15     <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject' />
16     <meta content='BindingOperations.console.txt' name='reference' />
17     <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/BindingOperations.x3d'
18       name='identifier' />
19     <meta content='X3D-Edit, http://www.web3d.org/x3d/content/README.X3D-Edit.html' name='generator' />
20     <meta content='../..//license.html' name='license' />
21   </head>
22   <Scene>
23     <Viewpoint DEF='View1' centerOfRotation='-6 0 0' description='Viewpoint 1' position='-6 0 5' />
24     <Viewpoint DEF='View2' centerOfRotation='-2 0 0' description='Viewpoint 2' position='-2 0 5' />
25     <Viewpoint DEF='View3' centerOfRotation='2 0 0' description='Viewpoint 3' position='2 0 5' />
26     <Viewpoint DEF='View4' centerOfRotation='6 0 0' description='Viewpoint 4' position='6 0 5' />
27     <Group>
28       <Transform DEF='Text1' translation='-6 0 0'>
29         <Shape>
30           <Text string='View" "# 1"'>
31             <FontStyle DEF='CenterJustify' />
32           </Text>
33           <Appearance>
34             <ImageTexture DEF='Image1' url='http://www.web3d.org/x3d/content/examples/AdvancedAnimation/anim.gif' />
35           </Appearance>
36         </Shape>
37       </Transform>
38       <Transform DEF='Text2' translation='0 0 0'>
39         <Shape>
40           <Text string='View" "# 2"'>
41             <FontStyle DEF='CenterJustify' />
42           </Text>
43           <Appearance>
44             <ImageTexture DEF='Image2' url='http://www.web3d.org/x3d/content/examples/AdvancedAnimation/anim.gif' />
45           </Appearance>
46         </Shape>
47       </Transform>
48       <Transform DEF='Text3' translation='0 0 0'>
49         <Shape>
50           <Text string='View" "# 3"'>
51             <FontStyle DEF='CenterJustify' />
52           </Text>
53           <Appearance>
54             <ImageTexture DEF='Image3' url='http://www.web3d.org/x3d/content/examples/AdvancedAnimation/anim.gif' />
55           </Appearance>
56         </Shape>
57       </Transform>
58       <Transform DEF='Text4' translation='0 0 0'>
59         <Shape>
60           <Text string='View" "# 4"'>
61             <FontStyle DEF='CenterJustify' />
62           </Text>
63           <Appearance>
64             <ImageTexture DEF='Image4' url='http://www.web3d.org/x3d/content/examples/AdvancedAnimation/anim.gif' />
65           </Appearance>
66         </Shape>
67       </Transform>
68     </Group>
69     <!-- The following advanced animation sequence uses nodes covered in Chapters 7, 8 and 9. -->
70     <!-- It does not need to be studied in this chapter. -->

```



Edit Viewpoint

DEF ☒ View1
USE ☐ View1
containerField children

centerOfRotation -6 0 0
description Viewpoint 1
fieldOfView 0.785398
jump ☒
orientation 0 0 1 0
position -6 0 5

OK Cancel Help



X3D for Web Authors, Kelp Forest Exhibit, Kelp Forest Main - Mozilla Firefox

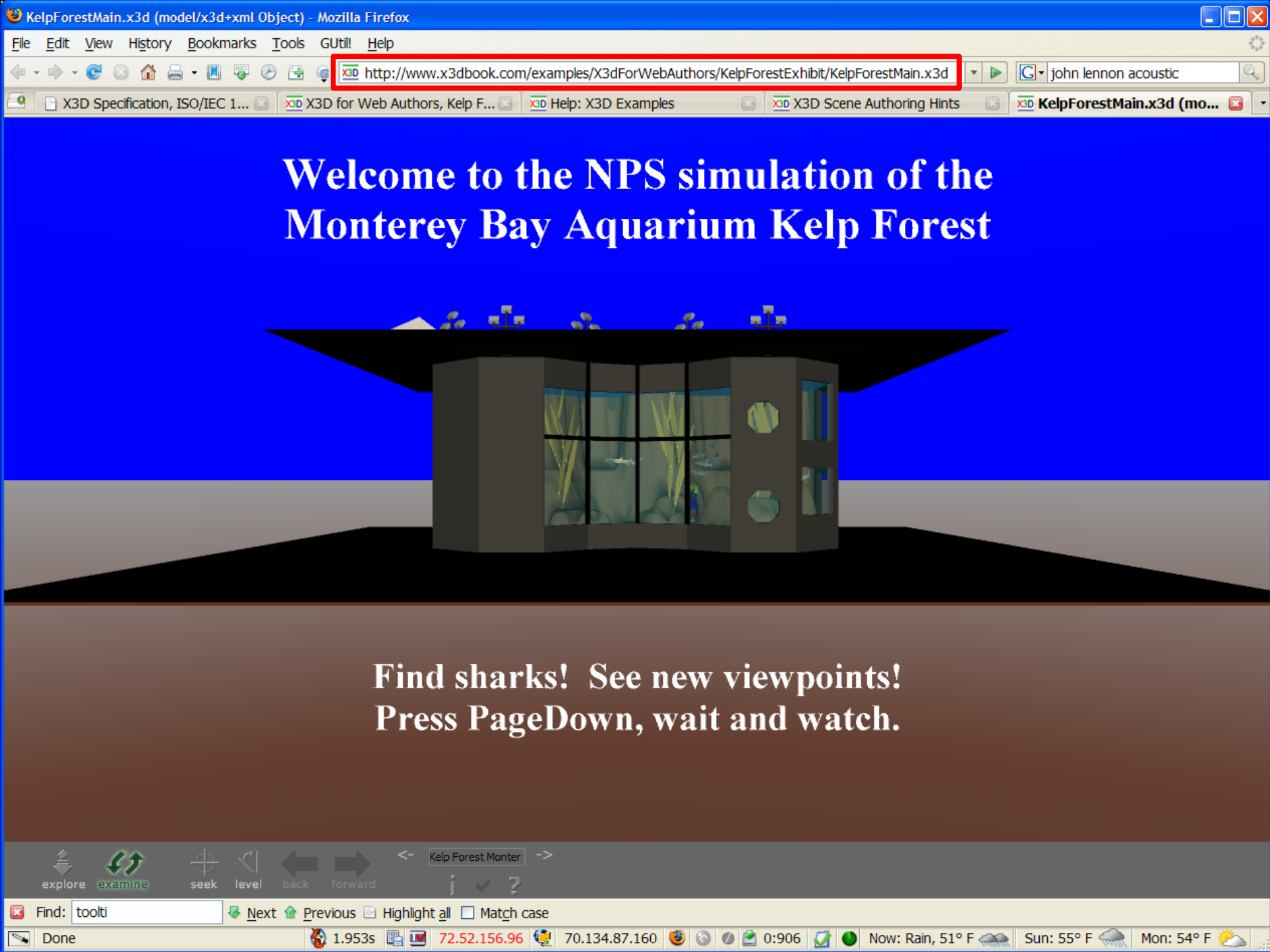
FileEditViewHistoryBookmarksToolsGUTilHelp

http://www.x3dbook.com/examples/X3dForWebAuthors/KelpForestExhibit/\_pages/page23.htmljohn lennon acoustic

X3D Specification, ISO/IEC 1...X3D for Web Authors, Ke...X3D Help: X3D ExamplesX3D X3D Scene Authoring HintsX3D KelpForestMain.x3d (model/...

<





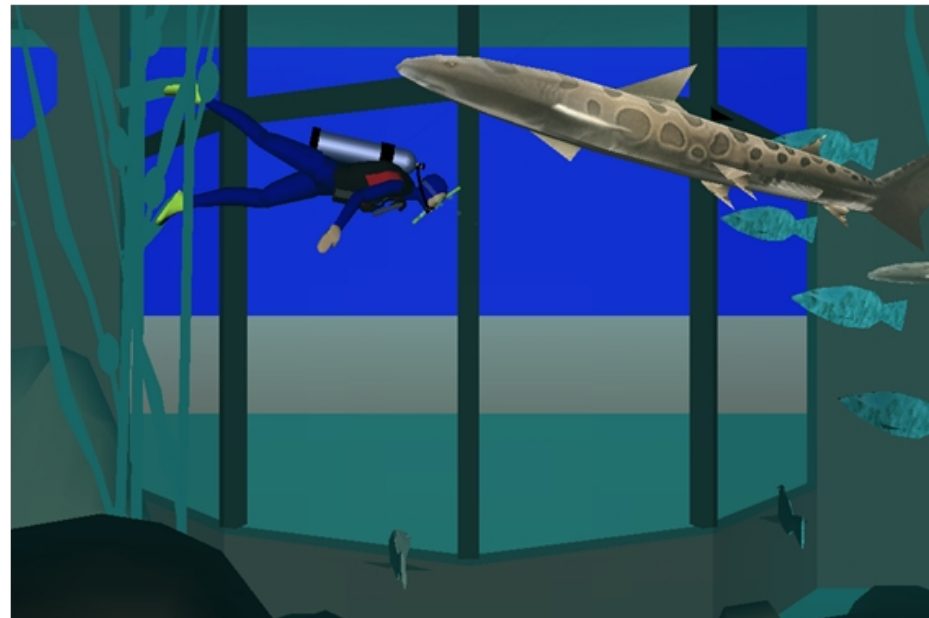
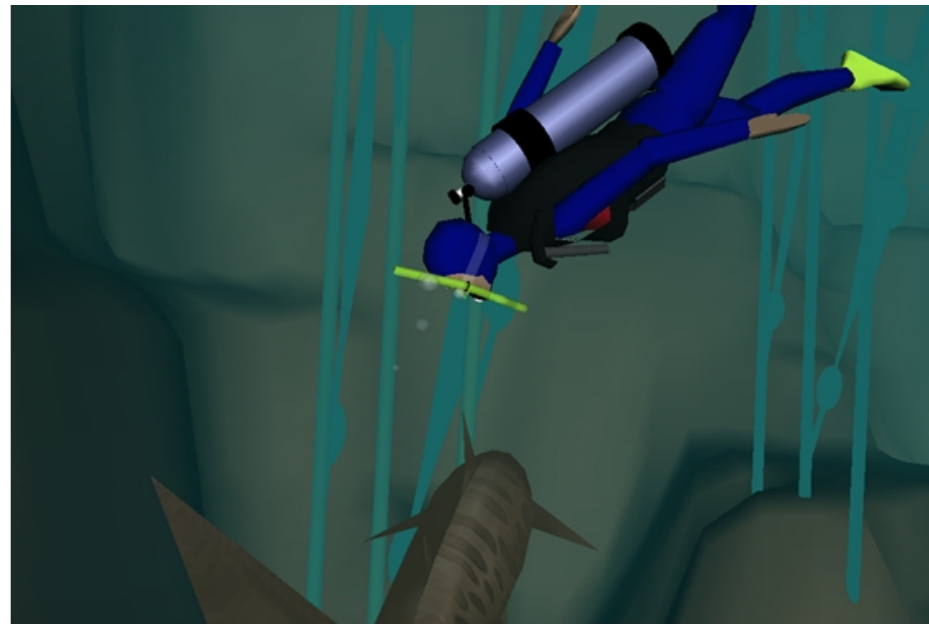
# Welcome to the NPS simulation of the Monterey Bay Aquarium Kelp Forest

Find sharks! See new viewpoints!  
Press PageDown, wait and watch.




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

**Find sharks! See new viewpoints!  
Press PageDown, wait and watch.**





 <b>Viewpoint</b>	Viewpoint provides a specific location and direction where the user may view the scene. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes.
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p><b>Hint:</b> descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring <code>_all_</code> other attributes and children.</p> <p><b>Hint:</b> USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p><b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
description	<p>[description: accessType initializeOnly, type SFString CDATA #IMPLIED]</p> <p>Text description or navigation hint to be displayed for this Viewpoint.</p> <p><b>Hint:</b> use spaces, make descriptions clear and readable.</p> <p><b>Warning:</b> without description, Viewpoint is unlikely to appear on browser Viewpoints menu. Hint: many XML tools substitute XML character references automatically if needed (like <code>&amp;#38;</code> for <code>&amp;</code> or <code>&amp;#34;</code> for <code>"</code>).</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
position	<p>[position: accessType inputOutput, type SFVec3f CDATA "0 0 10"]</p> <p>Position (x, y, z in meters) relative to local coordinate system.</p>
orientation	<p>[orientation: accessType inputOutput, type SFRotation CDATA "0 0 1 0"]</p> <p>Rotation (axis, angle in radians) of Viewpoint, relative to default -Z axis direction in local coordinate system.</p> <p><b>Hint:</b> this is orientation <code>_change_</code> from default direction (0 0 -1).</p> <p><b>Hint:</b> complex rotations can be accomplished axis-by-axis using parent Transforms.</p>
fieldOfView	<p>[fieldOfView: accessType inputOutput, type SFFloat CDATA "0.785398" (0..pi)]</p> <p>Preferred minimum viewing angle from this viewpoint in radians. Small field of view roughly corresponds to a telephoto lens, large field of view roughly corresponds to a wide-angle lens.</p> <p><b>Hint:</b> modifying Viewpoint distance to object may be better for zooming.</p> <p><b>Warning:</b> fieldOfView may not be correct for different window sizes and aspect ratios.</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
jump	<p>[jump: accessType inputOutput, type SFBool (true/false) "true"]</p> <p>Transition instantly by jumping, or smoothly animate to this Viewpoint.</p> <p><b>Hint:</b> set <code>jump=true</code> for smooth camera motion when going to this viewpoint.</p>
centerOfRotation	<p>[centerOfRotation: accessType inputOutput, type SFVec3f CDATA "0 0 0"]</p> <p>centerOfRotation point relates to NavigationInfo EXAMINE mode.</p>
set_bind	<p>[set_bind: accessType inputOnly, type SFBool (true/false) #FIXED ""]</p> <p>Sending event <code>set_bind=true</code> makes this node active. Sending event <code>set_bind=false</code> makes this node inactive. Thus setting <code>set_bind</code> to true/false will pop/push (enable/disable) this Viewpoint.</p>
bindTime	<p>[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]</p> <p>Event sent when node becomes active/inactive.</p>
isBound	<p>[isBound: accessType outputOnly, type SFBool (true/false) #FIXED ""]</p> <p>Event true sent when node becomes active, event false sent when unbound by another node.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>



# Navigation model 1

Users can select predefined Viewpoints

- Defines both position and direction of view

Users can further navigate around scene

- Using pointing device or hot keys
- Chosen viewpoint remains bound

Key	Emulated Action	WALK mode	FLY mode	EXAMINE mode
Up arrow	Pointer up	forward	forward	orbit up
Down arrow	Pointer down	backward	backward	orbit down
Left arrow	Pointer left	left	left	orbit left
Right arrow	Pointer right	right	right	orbit right



# Navigation model 2

User's current view can be animated

- ROUTE new position/direction event values to the Viewpoint itself, or to parent Transform nodes
- User navigation offsets to that view remain in effect
- Thus “over the shoulder” viewpoints can follow a moving object around, while still allowing user to look around while in that moving viewpoint

Lefty and Lucy shark in the Kelp Forest Main scene use this technique as virtual tour guides



# NavigationInfo node

NavigationInfo indicates how a browser might best support user navigation in the scene

Multiple NavigationInfo nodes may exist in scene

- Or in multiple Inline scenes loaded together

NavigationInfo is an X3DBindableNode

- So only one can be active at a given time
- Follow the same binding rules as Viewpoint, but not easily selectable
- Can be linked to a given Viewpoint by ROUTE that connects isBound of one node to set\_bind of other



# NavigationInfo *type*

Primary field is *type* which indicates which of the various modes of navigation are relevant

- "EXAMINE" best for rotating solitary objects
- "FLY" allows zooming in, out and around
- "WALK" also allows exploration, but on the ground
- "LOOKAT" use pointer to select geometry of interest
- "ANY" lets user select any mode
- "NONE" gives user zero control of navigation

MFString array default *type*=' "EXAMINE" "ANY" '

- which gives users plenty of flexibility



# NavigationInfo *type* details 1

- **"EXAMINE"** Used to view individual objects. Scene navigation consists of rotating the user viewpoint about the center of the observed object. The centerOfRotation field of the currently bound Viewpoint node values determines which local point centers the view rotation.
- **"WALK"** Used when exploring a virtual world on the ground. The user's eye level stays above the ground geometry and collision detection prevents the user from falling if underlying geometry is present.



# NavigationInfo *type* details 2

- **"FLY"** Similar to "WALK", but terrain following and collision detection is ignored. This type of navigation has the fewest constraints. Shifts the current view and related *centerOfRotation* values to track or zoom toward objects of interest to user.
- **"ANY"** Browser is allowed to provide whichever navigation type seems appropriate for the task at hand, modifying the user interface if necessary.
- **"NONE"** All navigation disabled and hidden. Navigation remains possible via animation of viewpoint fields or by binding other viewpoints (using viewpoint-list selection or Anchor node).



# NavigationInfo speed, headlight

*speed* determines how fast navigation occurs

- Default value 1 meter/second is usually pretty slow
- Might need to vary widely from ground to space

*headlight* is whether a light is shining ahead from user's point of view

- Otherwise one or more Light nodes is needed (covered in Chapter 11), or else world goes black



# NavigationInfo transitions

*transitionType* determines type of path followed when transitioning between viewpoints

- "ANIMATE" browser chooses smoothing algorithm
- "LINEAR" interpolation of position, orientation
- "TELEPORT" immediate repositioning to destination

*transitionTime*

- initial array value used for linear, otherwise multiple values can be used by browser-specific "ANIMATE"

*transitionComplete* lets author know when done

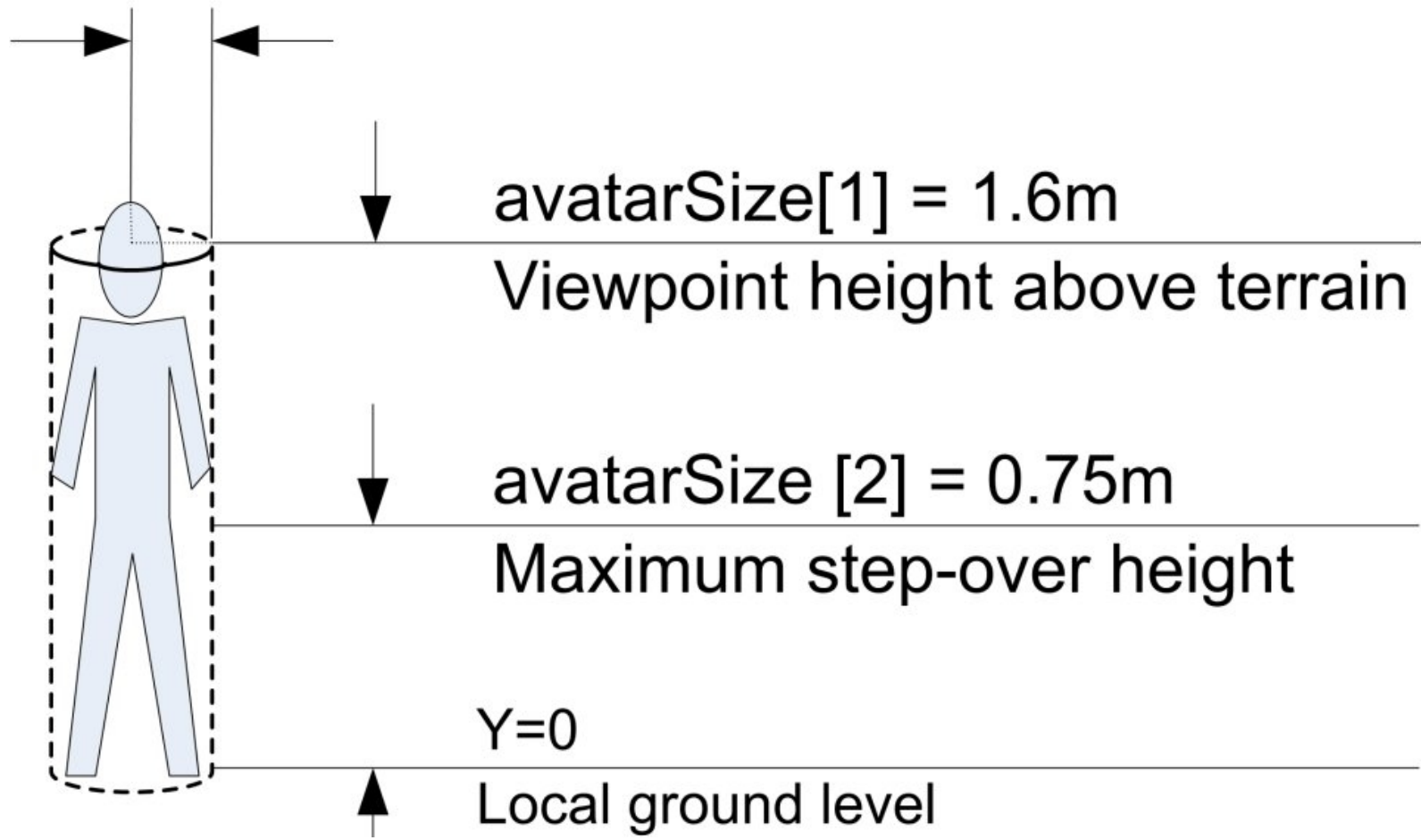
- SFBool boolean event sent when move is finished



# *avatarSize* SFVec3f array

`avatarSize[0] = 0.25m`

Allowed collision distance





# NavigationInfo *visibilityLimit*

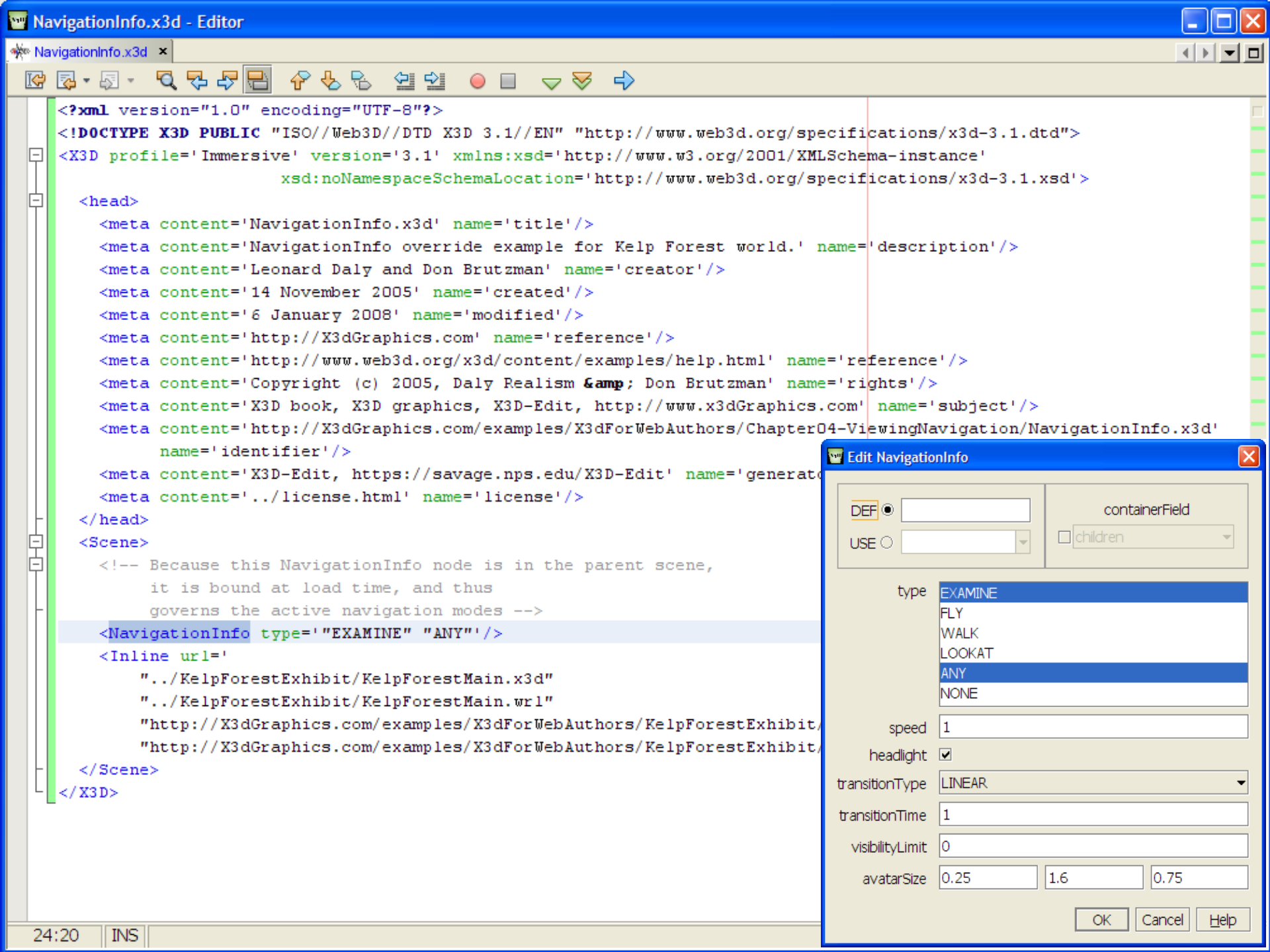
*visibilityLimit* defines the maximum range that may be rendered by the browser

- Measured from the user's point of view
- Geometry beyond that distance are not drawn
- *visibilityLimit*='0.0' means no limits are imposed

Quality thumbrule: meet following relationship

- $avatarSize.collisionDistance / visibilityLimit < 10,000$
- Avoids floating-point roundoff error on graphics card and almost-coplanar polygon tearing/aliasing
- Exactly coplanar polygons still suffer from aliasing





```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
      xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>

  <head>
    <meta content='NavigationInfo.x3d' name='title' />
    <meta content='NavigationInfo override example for Kelp Forest world.' name='description' />
    <meta content='Leonard Daly and Don Brutzman' name='creator' />
    <meta content='14 November 2005' name='created' />
    <meta content='6 January 2008' name='modified' />
    <meta content='http://X3dGraphics.com' name='reference' />
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
    <meta content='Copyright (c) 2005, Daly Realism & 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/Chapter04-ViewingNavigation/NavigationInfo.x3d'
          name='identifier' />
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
    <meta content='../license.html' name='license' />
  </head>
  <Scene>
    <!-- Because this NavigationInfo node is in the parent scene,
         it is bound at load time, and thus
         governs the active navigation modes -->
    <NavigationInfo type='EXAMINE' "ANY" />
    <Inline url='
      ../KelpForestExhibit/KelpForestMain.x3d"
      ../KelpForestExhibit/KelpForestMain.wrl"
      http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/
      http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/
    </Scene>
  </X3D>
```

**Edit NavigationInfo**

DEF ☐  USE ☐

containerField

☐ children

type EXAMINE  
FLY  
WALK  
LOOKAT  
ANY  
NONE

speed 1

headlight ☒

transitionType LINEAR

transitionTime 1

visibilityLimit 0

avatarSize 0.25 1.6 0.75

OK Cancel Help



 NavigationInfo	<p>NavigationInfo describes the viewing model and physical characteristics of the viewer's avatar.</p> <p><b>Hint:</b> for inspection of simple objects, usability often improves with type="EXAMINE" "ANY"</p> <p><b>Hint:</b> NavigationInfo types ""WALK" "FLY"" support camera-to-object collision detection. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p><b>Hint:</b> descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p><b>Hint:</b> USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p><b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
type	<p>[type: accessType inputOutput, type MFString CDATA "EXAMINE" "ANY"]</p> <p>Enter one or more quoted Strings: "EXAMINE" "WALK" "FLY" "LOOKAT" "ANY" "NONE". <b>Hint:</b> for inspection of simple objects, usability often improves with type="EXAMINE" "ANY". <b>Hint:</b> types WALK and FLY force strict camera-to-object collision detection. <b>Hint:</b> see Collision node for further details on camera-to-object collision detection. <b>Hint:</b> Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ]</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
speed	<p>[speed: accessType inputOutput, type SFFloat CDATA "1.0"]</p> <p>[0..+infinity) Default rate at which viewer travels through scene, meters/second.</p> <p><b>Warning:</b> default 1 m/s usually seems slow for ordinary navigation.</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
headlight	<p>[headlight: accessType inputOutput, type SFBool (true/false) "true"]</p> <p>Enable/disable directional light that always points in the direction the user is looking.</p>
avatarSize	<p>[avatarSize: accessType inputOutput, type MFFloat CDATA "0.25 1.6 0.75"]</p> <p>avatarSize triplet values are: (a) collision distance between user and geometry (near culling plane of the view frustum) (b) viewer height above terrain (c) tallest height viewer can WALK over. <b>Hint:</b> keep (visibilityLimit / avatarSize.CollisionDistance) &lt; 10,000 to avoid aliasing artifacts (i.e. polygon "tearing").</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
visibilityLimit	<p>[visibilityLimit: accessType inputOutput, type SFFloat CDATA "0.0"]</p> <p>Geometry beyond the visibilityLimit may not be rendered (far culling plane of the view frustum). visibilityLimit=0.0 indicates an infinite visibility limit. <b>Hint:</b> keep visibilityLimit &gt;= zero. <b>Hint:</b> keep (visibilityLimit / avatarSize.CollisionDistance) &lt; 10,000 to avoid aliasing artifacts (i.e. polygon "tearing").</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
transitionType	<p>[transitionType: accessType inputOutput, type MFString CDATA "ANIMATE"]</p> <p>Enter one or more quoted Strings: "ANIMATE" "LINEAR" "TELEPORT".</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
transitionTime	<p>[transitionTime: accessType inputOutput, type MFFloat CDATA "1.0"]</p> <p>Duration of viewpoint transition. <b>Hint:</b> If transitionType is "ANIMATE", transitionTime provides browser-dependent animation parameters.</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
transitionComplete	<p>[transitionComplete: accessType outputOnly, type MFFloat CDATA #FIXED ""]</p> <p>Event signaling viewpoint transition complete.</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
set_bind	<p>[set_bind: accessType inputOnly, type SFBool (true/false) #FIXED ""]</p> <p>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.</p>
bindTime	<p>[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""]</p> <p>Event sent when node becomes active/inactive.</p>
isBound	<p>[isBound: accessType outputOnly, type SFBool (true/false) #FIXED ""]</p> <p>Event true sent when node becomes active, event false sent when unbound by another node.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>

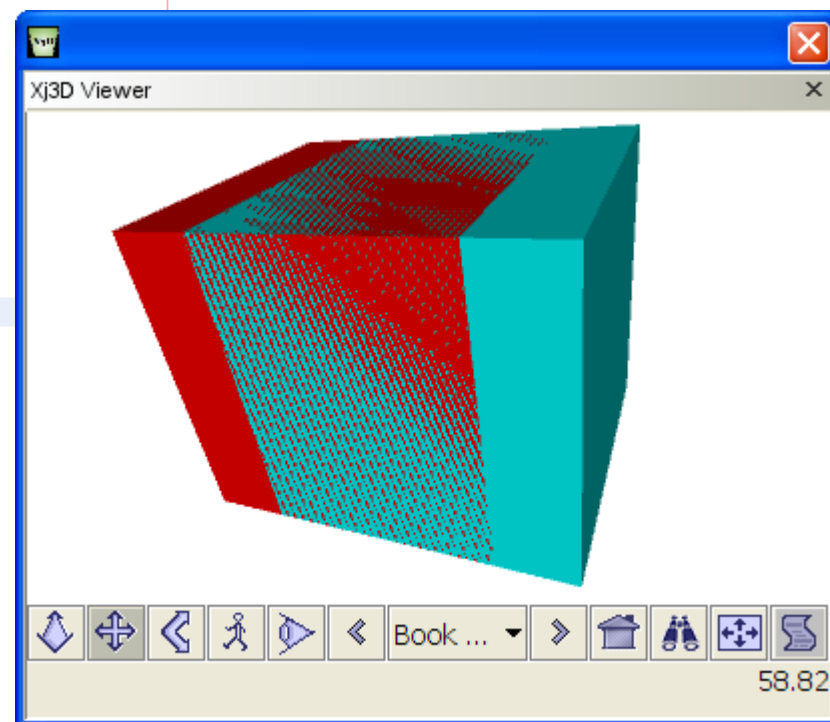


```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
      xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>

  <head>
    <meta content='AliasingExample.x3d' name='title'/>
    <meta content='Illustrates the effect of aliasing (tearing) by the coplanar overlap of two cubes with different colors.'
      name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='19 June 2006' name='created'/>
    <meta content='7 January 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright (c) 2006, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/AliasingExample.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='true' type="EXAMINE" "ANY"/>
    <Viewpoint description='Book View'/>
    <Transform translation='-0.25 0 0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Box/>
      </Shape>
    </Transform>
    <Transform translation='0.25 0 0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0 1 1'/>
        </Appearance>
        <Box/>
      </Shape>
    </Transform>
  </Scene>
</X3D>

```





# Anchor node

Anchor is another grouping node that can contain other nodes

Geometry rendered by contained nodes is activated and can be selected

- Clicking on Anchor geometry launches url link
- Alternatively can select a viewpoint in the scene (similar to HTML bookmark)
- Thus similar to HTML anchor tag

Selected link can replace current X3D scene, or else can launch into another browser window



# Anchor *description*

The *description* field provides the user with a single-string summary of what is selected when the Anchor geometry is selected, e.g.

- *description*='click door, open portal to new world'
- *description*='jump to next viewpoint...'

X3D browsers usually pop up the text description when the pointing device is over the selection geometry



# *url* Uniform Resource Locator

The *url* field provides either

- Address to new X3D scene, HTML page, or another Web resource, or else
- Viewpoint bookmark within the scene

MFString array provides alternate url addresses

- url addresses can be either local or online
- Point to alternate versions of same resource
- X3D browser goes sequentially through ordered list, one at a time, until one retrieval succeeds



# Anchor *parameter*

*parameter* provides additional information to browser regarding redirection of loaded result

- *parameter*='target=\_blank' sends to new frame
- *parameter*='target=frame4' sends to named frame
- May be ignored if browser is solely X3D capable, rather than (for instance) a Web-browser plugin

Once again, designed to match HTML anchor tag



# Anchor hints and warnings

Strictly match capitalization of directories and filenames

- Unix and http are case sensitive and fail otherwise
- Windows is forgiving but actually this hides errors

XML escape characters

- & (ampersand)           &amp;
- ' (apostrophe)         &apos;
- " (double quote)       &quot;



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
    xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>

  <head>
    <meta content='Anchor.x3d' name='title' />
    <meta content='Anchor example using the Kelp Forest world.' name='description' />
    <meta content='Leonard Daly and Don Brutzman' name='creator' />
    <meta content='14 November 2005' name='created' />
    <meta content='7 January 2007' name='modified' />
    <meta content='Show Anchor link to Monterey Bay Aquarium web site for Kelp Forest exhibit' name='reference' />
    <meta content='http://www.mbayaq.org/efc/kelp.asp' name='reference' />
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference' />
    <meta content='Copyright (c) 2005, Daly Realism & 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/Chapter04-ViewingNavigation/Anchor.x3d'
      name='identifier' />
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
    <meta content='../license.html' name='license' />
  </head>
  <Scene>
    <Transform translation='0 8 30'>
      <Viewpoint description="view Anchor text" position="0 -1 12"/>
      <Anchor DEF='AnchorExample' description='Aquarium Exhibit Web Site'
        parameter='target=_blank' url='http://www.mbayaq.org/efc/kelp.asp'>
        <Shape bboxSize='-1 -1 -1'>
          <Text string='"Click orange text for" "Monterey Bay Aquarium" "kelp forest web site"'>
            <FontStyle justify='MIDDLE' MIDDLE' size='0.6' />
          </Text>
          <Appearance>
            <Material DEF='Autumn11' ambientIntensity='0.25641' diffuseColor='0.795918 0.273554 0.006861'
              shininess='1' specularColor='0.48655 0.319155 0.444036' />
          </Appearance>
        </Shape>
      </Anchor>
    </Transform>
    <Inline url='"../KelpForestExhibit/KelpForestMain.x3d"
      "../KelpForestExhibit/KelpForestMain.wrl"
      "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d"
      "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.wrl"' />
  </Scene>

```



# Anchor example scene, editor



**Edit Anchor**

DEF ☒ AnchorExample  
USE ☐ AnchorExample

containerField  
☐ children

description  
Aquarium Exhibit Web Site

url  
<http://www.mbayaq.org/efc/kelp.asp>


parameter  
target=\_blank

bboxCenter  
0 0 0

bboxSize  
-1 -1 -1

OK Cancel Help



 <b>Anchor</b>	<p><b>Anchor is a Grouping node that can contain most nodes. Clicking Anchored geometry loads content specified by the url field. Loaded content completely replaces current content, if parameter is same window.</b></p> <p><b>Hint:</b> insert a Shape node before adding geometry or Appearance.</p>
DEF	<p><b>[DEF ID #IMPLIED]</b></p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p><b>Hint:</b> descriptive DEF names improve clarity and help document a model.</p>
USE	<p><b>[USE IDREF #IMPLIED]</b></p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p><b>Hint:</b> USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p><b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
description	<p><b>[description: accessType inputOutput, type SFString CDATA #IMPLIED]</b></p> <p>Text description to be displayed for action of this node. Hint: many XML tools substitute XML character references automatically if needed (like &amp;#38; for &amp; or &amp;#34; for ").</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
url	<p><b>[url: accessType inputOutput, type MFString CDATA #IMPLIED]</b></p> <p>Address of replacement world, activated by clicking Anchor geometry.</p> <p><b>Hint:</b> jump to a world's internal viewpoint by appending viewpoint name (e.g. #ViewpointName, someOtherCoolWorld.wrl#GrandTour).</p> <p><b>Hint:</b> jump to a local viewpoint by only using viewpoint name (e.g. #GrandTour).</p> <p><b>Hint:</b> Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ].</p> <p><b>Hint:</b> XML encoding for " is &amp;quot; (a character entity).</p> <p><b>Warning:</b> strictly match directory and filename capitalization for http links!</p> <p><b>Hint:</b> can replace embedded blank(s) in url queries with %20 for each blank character.</p> <p><b>Hint:</b> pop up a new window with url value as follows: "JavaScript:window.open('popup.html','popup','width=240,height=240');location.href='HelloWorld.wrl'"</p>
parameter	<p><b>[parameter: accessType inputOutput, type MFString CDATA #IMPLIED]</b></p> <p>Passed parameter that signals web browser how to redirect url loading. Hint: set parameter to target=_blank to load target url into a blank frame. Hint: set parameter to target=frame_name to load target url into another frame. Hint: Strings can have multiple values, so separate each string by quote marks. [ "http://www.url1.org" "http://www.url2.org" "etc." ].</p> <p><b>Interchange profile hint:</b> this field may be ignored.</p>
bboxCenter	<p><b>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</b></p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p><b>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</b></p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p><b>[containerField: NMTOKEN "children"]</b></p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p><b>[class CDATA #IMPLIED]</b></p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>



# Billboard node

Billboard is another X3DGroupingNode

Child-content geometry faces user

- Special effect that improves readability or visibility

*axisOfRotation* field determines Billboard pivot

- Relative to local coordinate system
- Default is *axisOfRotation*='0 1 0' which swivels about vertical (Y axis)
- Rotations unpredictable when above (on axis)
- Define *axisOfRotation*='0 0 0' for circular rotation to always fully face user without axis constraints



# Billboard hints and warnings

DEF, USE allowed for multiple Billboards nodes

- Each copy should independently face user

Put Billboard as close to moving geometry as possible, nested inside a positioning Transform

Do not put a Viewpoint under a Billboard

- Creates a feedback loop
- Unpredictable behavior likely to result



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
      xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>

  <head>
    <meta content='Billboard.x3d' name='title' />
    <meta content='Illustrates Billboard operation using Kelp Forest example scenes for Figure 4.4, X3D for Web Authors.' name='description' />
    <meta content='Don Brutzman and Leonard Daly' name='creator' />
    <meta content='13 June 2004' name='created' />
    <meta content='7 January 2007' 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='Revisions Copyright (c) 2006, Daly Realism and Don Brutzman. Original content has permissions for free use. Please provide' />
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/Billboard.x3d' name='identifier' />
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
    <meta content='../license.html' name='license' />
  </head>
  <Scene>
    <Background 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 0.0' />
    <Viewpoint description='Text message from ahead' position='0 -5 45' />
    <Viewpoint description='Text message Billboard effect' orientation='0 1 0 0.86' position='37.01 -5 29.59' />
    <Billboard axisOfRotation='0 1 0'>
      <Inline url='
        "../KelpForestExhibit/IntroductionMessage.wrl"
        "../KelpForestExhibit/IntroductionMessage.x3d"
        "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForest
        "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForest
      </Billboard>
    <Inline url='
      "../KelpForestExhibit/KelpTank.wrl"
      "../KelpForestExhibit/KelpTank.x3d"
      "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForest
      "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForest
    </Scene>
  </X3D>

```

**Edit Billboard**

☒ DEF 
☐ USE

☐ containerField
 

children

axisOfRotation

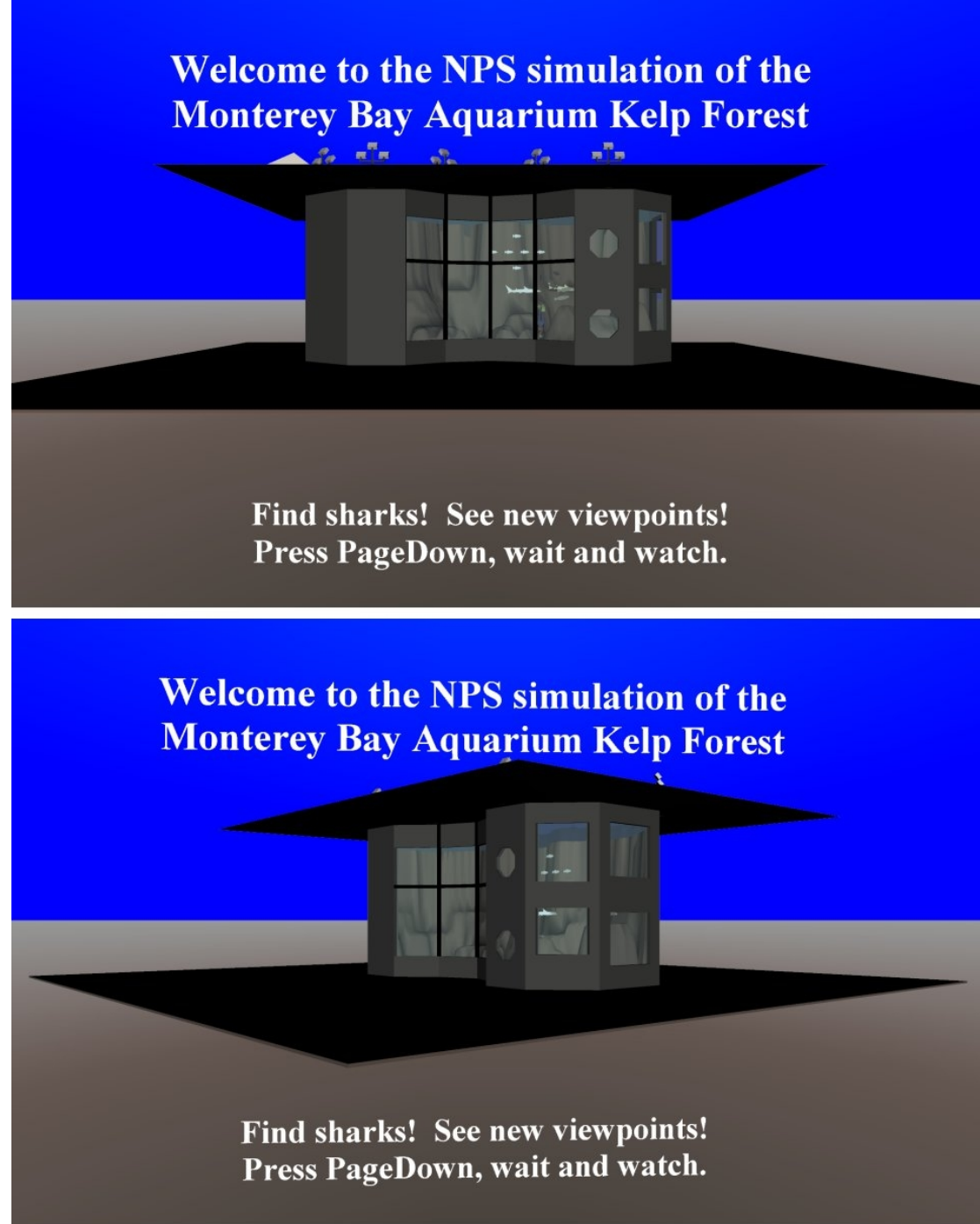
bboxCenter

bboxSize




## Billboard example

Starting at initial viewpoint and navigating with mouse or arrow keys reveals that Billboard Text remains facing the viewer, improving readability





 <b>Billboard</b>	<p>Billboard is a Grouping node that can contain most nodes. Content faces the user, rotating about the specified axis. Set axisOfRotation=0 0 0 to fully face the user's camera.</p> <p><b>Hint:</b> Put Billboard as close to the geometry as possible, nested inside Transform for local coordinate system.</p> <p><b>Hint:</b> don't put Viewpoint inside a Billboard.</p> <p><b>Hint:</b> insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p><b>Hint:</b> descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring __all__ other attributes and children.</p> <p><b>Hint:</b> USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p><b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
axisOfRotation	<p>[axisOfRotation: accessType inputOutput, type SFVec3f CDATA "0 1 0"]</p> <p>axisOfRotation direction is relative to local coordinate system.</p> <p><b>Hint:</b> axis 0 0 0 always faces viewer.</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>



# Collision node

Defines camera-to-object collision-detection properties between child geometry and user

- *enabled*='true' blocks user navigation through the geometry
- *enabled*='false' blocks user navigation through the geometry

Not used for object-to-object collision detection

Authors can detect when collision occurs

- SFTIME outputOnly event *collideTime*
- SFBool outputOnly event *isActive*



# Collision detection and terrain following

Terrain following depends on +Y axis being “up”

- Other coordinate systems are possible but do not match this X3D convention
- Thus datasets using other coordinates must be converted to match

WALK mode is another form of collision detection

- Viewer's camera drops until NavigationInfo avatar rests on geometry serving as the ground plane
- Step-over distance (an avatarSize parameter) governs whether user can rise over obstacles



# Collision *proxy* field

Child geometry may be quite detailed, irregular

- Complicating collision-detection calculations and thus slowing rendering performance

Can substitute SFNode *proxy* child as alternate

- Shape containing a Box, Sphere or Cylinder can provide simplifying geometric alternative
- *proxy* geometry is not rendered

```
<Collision DEF='Example' enabled='true'>
  <Shape containerField='proxy'><Cylinder/></Shape>
  <Inline url='SomeComplicatedObject.x3d' />
</Collision>
```



# Collision hints and warnings

```
<NavigationInfo type=' "WALK" "FLY" ' />
```

modes support camera-to-object collision detection

Only polygonal geometry can be used for collision detection

- No points or lines
- Special limitation: no Text node collisions
- Nevertheless can achieve same collision effects by adding a transparent Box or polygon, thus providing the necessary polygons as boundaries



Collision.x3d - Editor

Collision.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/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='Collision.x3d' name='title' />
    <meta content='Illustrate Collision node: set enabled="true" to block zooming through glass.' name='description' />
    <meta content='Don Brutzman and Leonard Daly' name='creator' />
    <meta content='1 June 1998' name='created' />
    <meta content='13 June 2004' name='translated' />
    <meta content='7 January 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='Revisions Copyright (c) 2006, Daly Realism and Don Brutzman. Original Content Copyright (c) 2006, Don Brutzman and Leonard Daly. All Rights Reserved.' name='copyright' />
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit' name='generator' />
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-Viewing' name='generator' />
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator' />
    <meta content='../license.html' name='license' />
  </head>
  <Scene>
    <NavigationInfo speed='3' type='FLY' "ANY"/>
    <Viewpoint description='Outside Tank' orientation='0.21 0.97 -0.14 1.202' position='0 0 0' />
    <Viewpoint description='Inside Tank' orientation='0.21 0.97 -0.14 1.202' position='0 0 0' />
    <Collision enabled='false'>
      <Inline url='../KelpForestExhibit/KelpTank.wrl' "../KelpForestExhibit/KelpTank.x3d" "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpTank.x3d" />
    </Collision>
  </Scene>
</X3D>

```

Edit Collision

DEF ☐

USE ☐

containerField

☐ children

enabled ☒

collideTime

axisOfRotation

bboxCenter

bboxSize

OK Cancel Help

initial view outside tank glass

zoom into tank, collision off

blocked by glass, collision on

Outside Tank

Outside Tank

side Tank




# Collision example

Example screen shots  
first show the viewer  
being stopped by  
glass geometry, then  
the viewer passing  
through the tank glass  
for a closer view.

Collision *enabled*="true"  
or *enabled*="false"  
result in different  
navigation responses.





 Collision	<p>Collision detects camera-to-object contact using current Viewpoint and NavigationInfo avatarSize. Collision is a Grouping node that handles collision detection for its children. Collision can contain a single proxy child node for substitute collision-detection geometry. Note: proxy geometry is not rendered. Note: PointSet, IndexedLineSet, LineSet and Text do not trigger collisions.</p> <p><b>Hint:</b> improve performance using proxy for simpler contact-calculation geometry.</p> <p><b>Hint:</b> NavigationInfo types ""WALK"" "FLY"" support camera-to-object collision detection.</p> <p><b>Hint:</b> insert a Shape node before adding geometry or Appearance.</p>
DEF	<p>[DEF ID #IMPLIED]</p> <p>DEF defines a unique ID name for this node, referencable by other nodes.</p> <p><b>Hint:</b> descriptive DEF names improve clarity and help document a model.</p>
USE	<p>[USE IDREF #IMPLIED]</p> <p>USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children.</p> <p><b>Hint:</b> USEing other geometry (instead of duplicating nodes) can improve performance.</p> <p><b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!</p>
bboxCenter	<p>[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"]</p> <p>Bounding box center: position offset from origin of local coordinate system.</p>
bboxSize	<p>[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"]</p> <p>Bounding box size: automatically calculated, can be specified as an optimization or constraint.</p>
enabled	<p>[enabled: accessType inputOutput, type SFBool (true false) "true"]</p> <p>Enables/disables collision detection for children and all descendants.</p> <p><b>Hint:</b> former name "collide" in VRML 97 specification.</p>
isActive	<p>[isActive: accessType outputOnly, type SFBool (true false) #FIXED ""]</p> <p>isActive true/false events are sent when triggering the sensor. isActive=true when view-object collision occurs, isActive=false when view-object collision no longer occurs.</p>
collideTime	<p>[collideTime: accessType outputOnly, type SFTime CDATA #FIXED ""]</p> <p>Time of collision between camera (avatar) and geometry.</p>
containerField	<p>[containerField: NMTOKEN "children"]</p> <p>containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.</p>
class	<p>[class CDATA #IMPLIED]</p> <p>class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.</p>



# Additional Resources



# File formatting

X3D-Edit has a Netbeans capability for formatting

- *Alt-shift-F* **Format** acts upon highlighted text blocks, also available via right-click menu
- Warning: do not reformat embedded ECMAScript source code

X3D Canonicalization (C14N) also reformats X3D

- Performed prior to examples being placed in archive
- Will be added to X3D-Edit Tools menu



# Pretty-print HTML capabilities

*Pretty print* means to reformat nicely in HTML, usually with color coding

- facilitates reading and printing

X3D-Edit has this Netbeans feature

- File > Print to HTML

X3dToXhtml.xslt stylesheet

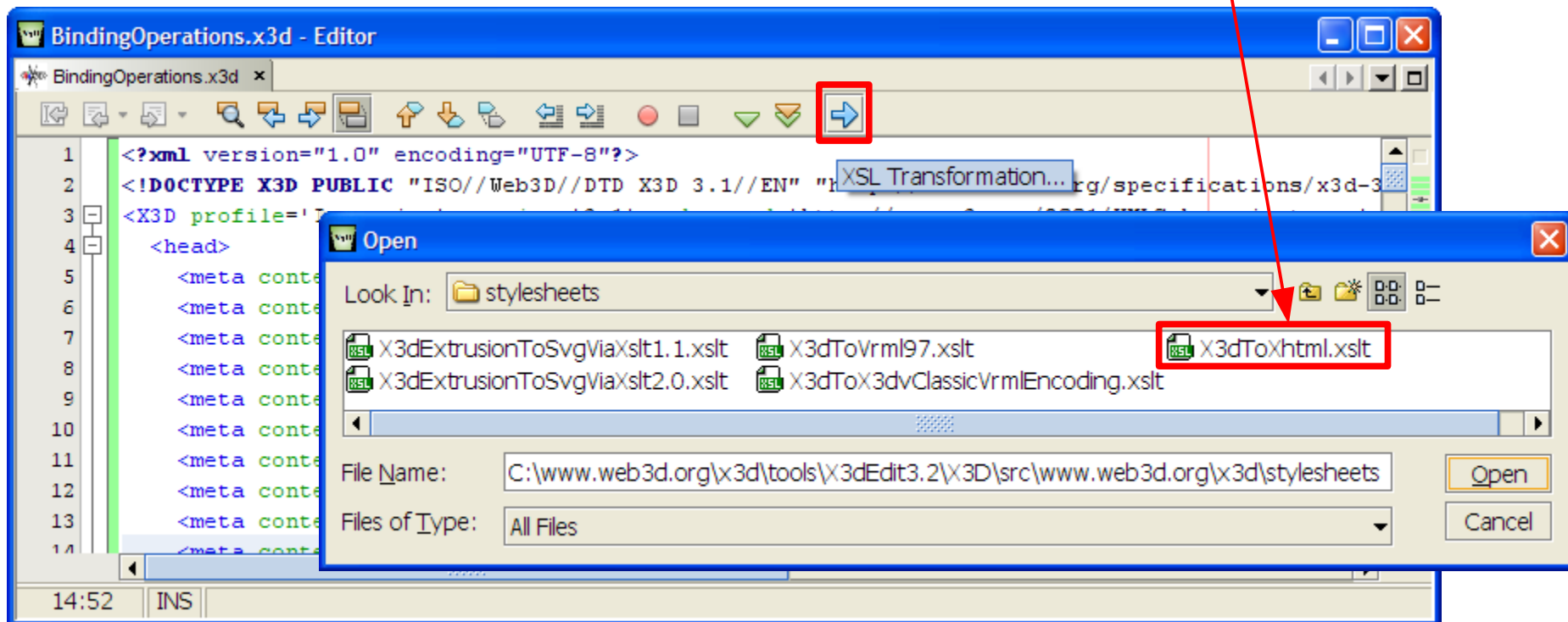
- Includes indices and hyperlinks to DEF/USE, ROUTEs, images, url values, prototypes, etc.
- Can be launched via XSL Transformation button
- Will be added to X3D-Edit Tools menu



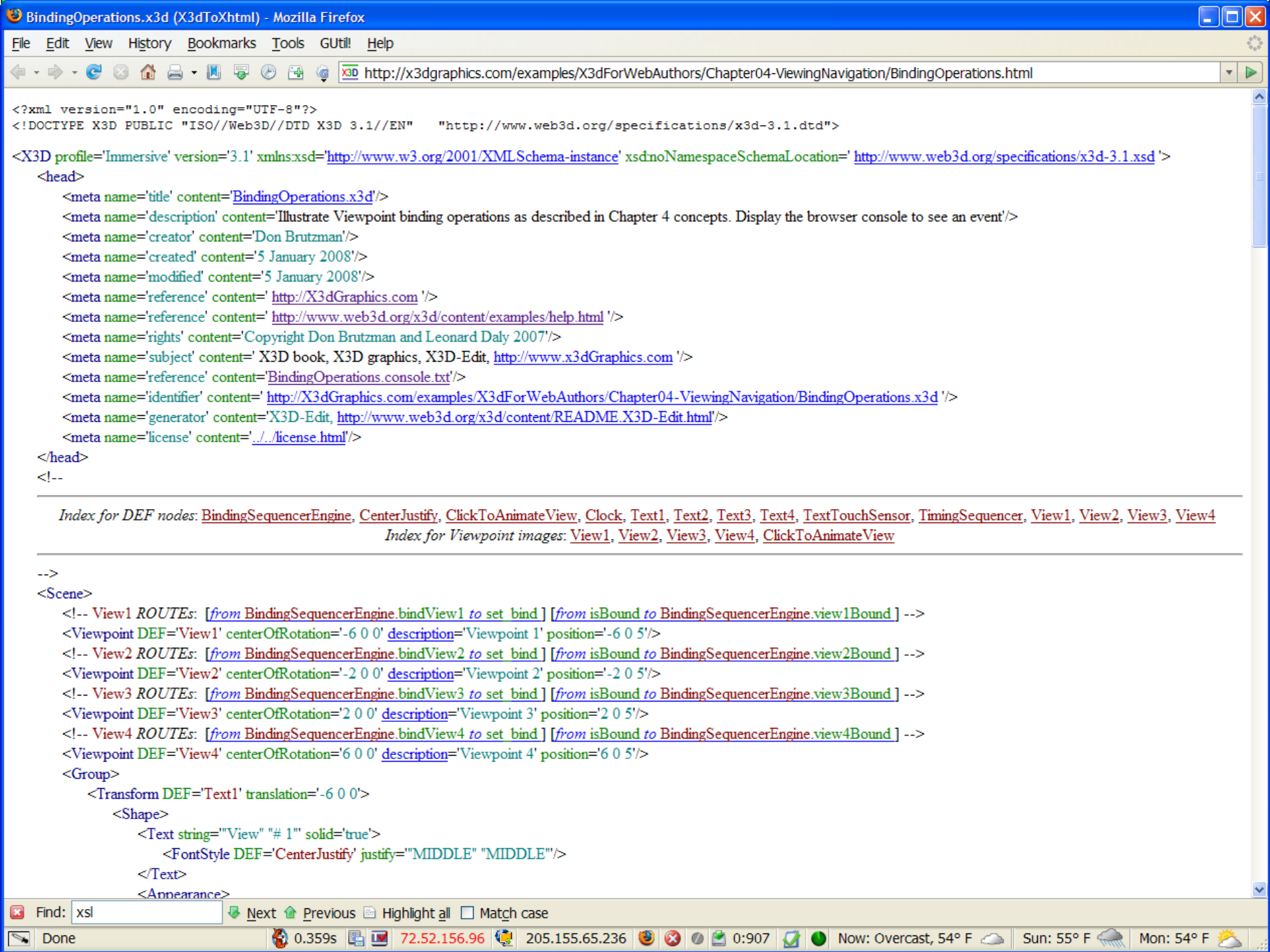
# Launching other XSLT stylesheets

Extensible Stylesheet Language for XML (XSLT) stylesheets support a variety of conversions

- X3dToVrml97.xslt
- X3dToClassicVRML.xslt
- X3dToXhtml.xslt
- others









# Chapter Summary



# Chapter Summary

Users explore X3D worlds by choosing predefined viewpoints and **navigating** through 3D space.

- **Bindable nodes**, so only one is active at a time
- **Viewpoint** lets authors identify key camera locations
- **NavigationInfo** provides options for moving around

Nodes to improve user navigability, interaction:

- **Anchor** makes geometric shapes linkable, like HTML
- **Billboard** for axis-aligned geometry facing the user
- **Collision** permits or blocks a user's current camera view from passing through collidable geometry



[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 4, Viewing and Navigation
- <http://x3dGraphics.com>
- <http://x3dgraphics.com/examples/X3dForWebAuthors>

## X3D Examples Help

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



# References 2

## X3D Scene Authoring Hints

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

## X3D Graphics Specification

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



# References 3

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

- <http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm>
- <http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook>
- Chapter 26 – Viewpoint



# Contact

**Don Brutzman**

*[brutzman@nps.edu](mailto:brutzman@nps.edu)*

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

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

1.831.656.2149 voice

1.831.656.7599 fax



# Open-source license

Copyright (c) 1995-2008 held by the author(s). All rights reserved.

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

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

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



# X3D Graphics for Web Authors

## Chapter 4

### Viewing and Navigation

*But the eyes, though they are no sailors, will never  
be satisfied with any model, however fashionable,  
which does not answer all the requisitions of art.*

Henry David Thoreau, 1849





# Contents

Chapter Overview and Concepts

X3D Nodes and Examples

Additional Resources

Chapter Summary

References





# Chapter Overview





## Overview: Viewing and Navigation

Users explore X3D worlds by choosing predefined viewpoints and navigating through 3D space.

- **Viewpoint** lets authors identify key camera locations
- **NavigationInfo** provides options for moving around
- **Bindable nodes**, so only one is active at a time

Related nodes improve navigability, interaction

- **Anchor** makes geometric shapes linkable
- **Billboard** keeps child geometry facing the user
- **Collision** can allow or prevent a user's view from passing through geometry





[back to Table of Contents](#)

# Concepts





# Viewing and navigation

It is helpful to think of X3D scenes as fixed at different locations in 3D space

- Viewpoints are like cameras, prepositioned in locations (and directions) of interest
- Users can move their current camera viewpoint further and change direction they are looking at
- This process is called *navigation*

Making navigation easy for users is important

- Authors provide viewpoints of interest with scenes
- Browsers enable camera rotation, pan, zoom, etc.



Difficult navigation leads to users becoming “lost in space” or, worse yet from an author's perspective, simply leaving the scene because it is incomprehensible.



# Goals of viewing and navigation

- Viewing a scene from different vantage points that reveal aspects of interest, document key locations, or help to tell a story
- Navigating changes in the user's viewpoint effectively by moving from place to place in an intuitive manner
- Making geometric objects selectable so that users can transport to another viewpoint, launch into another scene, or receive other web content
- Taking advantage of viewpoint location for special interactive techniques, such as user-facing billboard rotations and terrain following





## Bindable nodes

Bindable nodes have a special property:  
only one can be active at a time

- Viewpoint, NavigationInfo, Background, TextureBackground, Fog
- Each implements X3DBindableNode type interface for consistency

Implemented using a stack

- Similar to spring-loaded tray of plates in cafeteria
- One (and only one) is active, on top
- One can be pulled off top, sent off to the side
- One can be pulled to top, pushing down others

Stack (data structure) [http://en.wikipedia.org/wiki/Stack\\_%28data\\_structure%29](http://en.wikipedia.org/wiki/Stack_%28data_structure%29)



# Binding example

Basic user operation is pretty simple:

- just select the desired Viewpoint

Complex example follows, stepping through binding stack operations

- Advanced details
- [BindingOperations.x3d](#)
- Animated with scripting
- Console results found in [BindingOperations.console.txt](#)



New users please skip ahead to [Nodes and Examples](#)

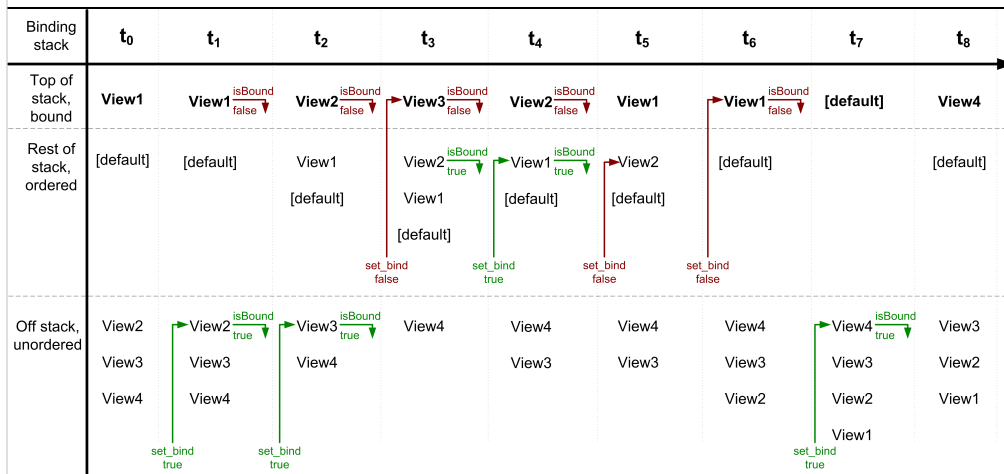
Related chapter for event passing: Chapter 8, User Interactivity Nodes

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/BindingOperations.x3d>

<http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/BindingOperations.console.txt>



# Binding node operations diagram



Viewpoints are activated (bound) upon selection, ordering is governed by stack operations

*X3D for Web Authors* Figure 4.1. Binding node operations: `set_bind` events control whether bindable nodes go to the top of the stack or pop off the stack.



# Binding node operations 1

- **Time t0**. The initial loading of the scene has first `<Viewpoint DEF='View1' />` active and bound to the top of the binding stack. Other viewpoints are off the binding stack.
- If no viewpoints are provided in the scene, then the default `<Viewpoint position='0 0 10' />` defined in the X3D Specification is used.
- **Time t1**. When the user selects View2 from the viewpoint list, it receives a *set\_bind="true"* event and goes to the top of the binding stack. View2 also issues an *isBound="true"* event, and View1 issues an *isBound="false"* event as it moves down the stack.

See preceding figure



## Binding node operations 2

- **Time t2**. Similar to the previous transitions in step t1, View3 receives a *set\_bind="true"* event and responds with an *isBound="true"* event, while View2 issues an *isBound="false"* event and pushes View1 further down the stack.
- **Time t3**. View3 receives a *set\_bind="false"* event, triggering a corresponding *isBound="false"* event and dropping off the stack completely. Because View2 is the next node on the binding stack, it pops to the top to become the active Viewpoint node. View2 also issues an *isBound="true"* event.

See preceding figure



## Binding node operations 3

- **Time t4.** The user now selects View1 from the browser's viewpoint list, so View1 receives a *set\_bind="true"* event and sends a corresponding *isBound="true"* event. View2 is no longer bound, and is pushed down the binding stack.
- **Time t5.** View2 receives a *set\_bind="false"* event while on the binding stack but unbound, and as a result, it is taken completely off the binding stack.
- **Time t6.** View1 is now removed off the binding stack via a *set\_bind="false"* event, leaving no other defined Viewpoint nodes on the stack.

See preceding figure



## Binding node operations 4

- **Time t7**. With no Viewpoint nodes remaining on the stack to bind, default viewpoint values are used: `<Viewpoint position='0 0 10'/>`. The user then selects the previously unbound View4 from the viewpoint list.
- **Time t8**. View4 remains as the bound viewpoint with no further viewpoints remaining on the stack.

Final note: same for other X3D bindable nodes

- Viewpoint, NavigationInfo, Background, TextureBackground, Fog

See preceding figure



[back to Table of Contents](#)

## X3D Nodes and Examples





# Viewpoint node

It is helpful to think of X3D scenes as being fixed solidly in 3D space, positioned and oriented exactly where placed by the scene author

Viewing a scene is thus a matter of navigating the current user point of view through space

Viewpoint nodes let X3D scene authors predefine locations and orientations of particular interest

- Sometimes viewpoints are animated and moving
- Freedom of viewpoint is exciting and engaging, also a major advantage over fixed-viewpoint video





## Viewpoint *position, orientation*

A Viewpoint node defines a specific *position* and *orientation* for looking at a 3D scene

- Similar to a “virtual camera” vantage point

Default Viewpoint *position* is (0 0 10)

- out 10 m on +Z axis, looking back towards origin

Any changes to Viewpoint *orientation* are made relative to that default direction (along -X axis)

- Different initial direction than other orientations
- Visualize the situation and then use right-hand rule to figure out the correct *orientation* value





## Viewpoint *description*

Each Viewpoint is given a *description* string to help users decide which view to select

- Clear, understandable descriptions can guide users
- Use an object's name first when many viewpoints follow, so they are more easily identified in a list
- Use whitespace instead of underscores for better readability

Viewpoints are primary user tool for navigation

- Browsers provide Viewpoint List to show and select descriptions





## Viewpoint *fieldOfView*, *centerOfRotation*

*fieldOfView* describes the angular width shown

- Horizontal breadth displayed
- Default is 45 degrees =  $\pi/4$  radians = 0.785
- Vertical field determined by browser aspect ration

*centerOfRotation* is a local position

- User's current view rotates about this point if the bound NavigationInfo node is in EXAMINE mode
- Can be changed by a user's LOOKAT operation picking some other geometry as new center





# Viewpoint *jump*

*jump* can be a tricky field (but is not often used)

- *jump*='true' when a Viewpoint is selected means that the current view position and orientation is modified according to NavigationInfo transitionType
- *jump*='true' is usual default
- *jump*='false' is an advanced technique
  - User's view doesn't appear to change when new Viewpoint is selected
  - New Viewpoint is bound, but given offsets to match prior user position and orientation (hence no jumping)
  - Example use: changing bound viewpoint when moving from one floor into an elevator, then to another floor



# Viewpoint hints and warnings

Use parent Transform node(s) for complex  
Viewpoint orientation, position values

- One axis of rotation at a time can work

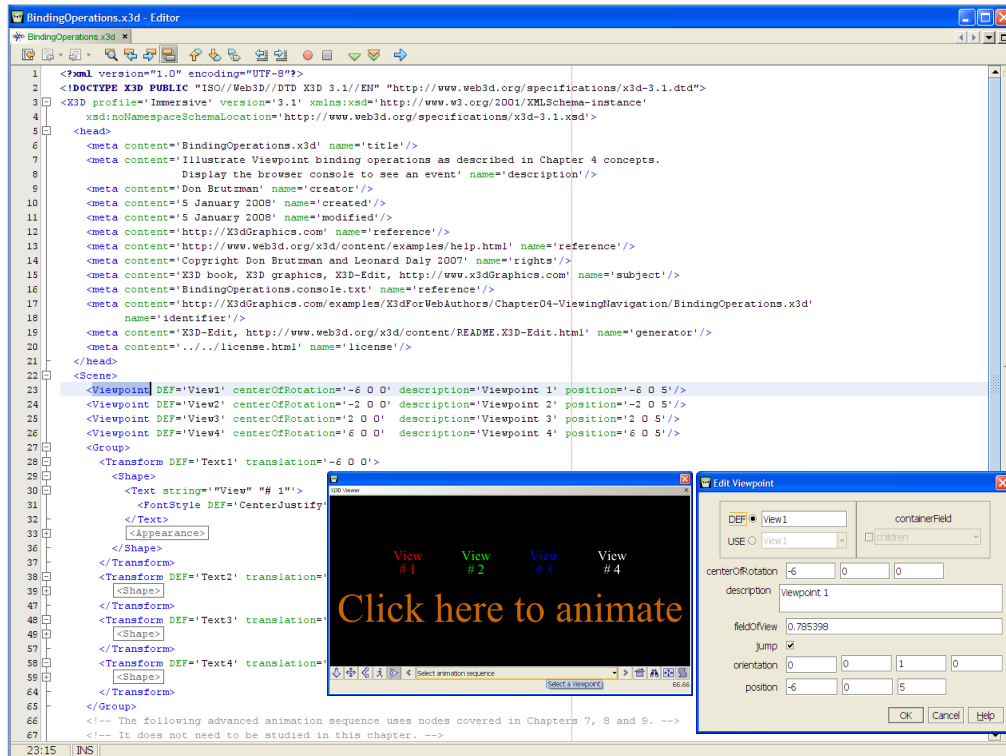
Keyboard shortcuts are helpful

- PageUp PageDown Home End to select Viewpoint
- Arrow keys to examine (rotate), pan, zoom, etc.  
depending on current NavigationInfo mode
- Browser may allow Viewpoint reset after navigating

Distinguish between defined Viewpoint and  
current navigated user-view location, direction

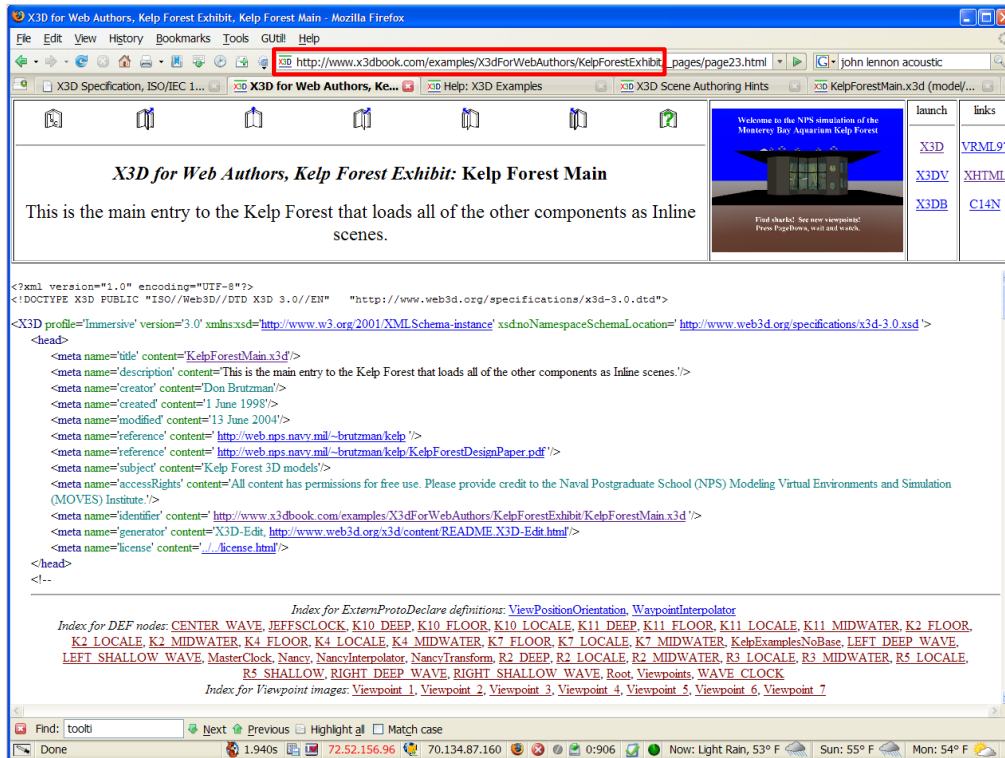






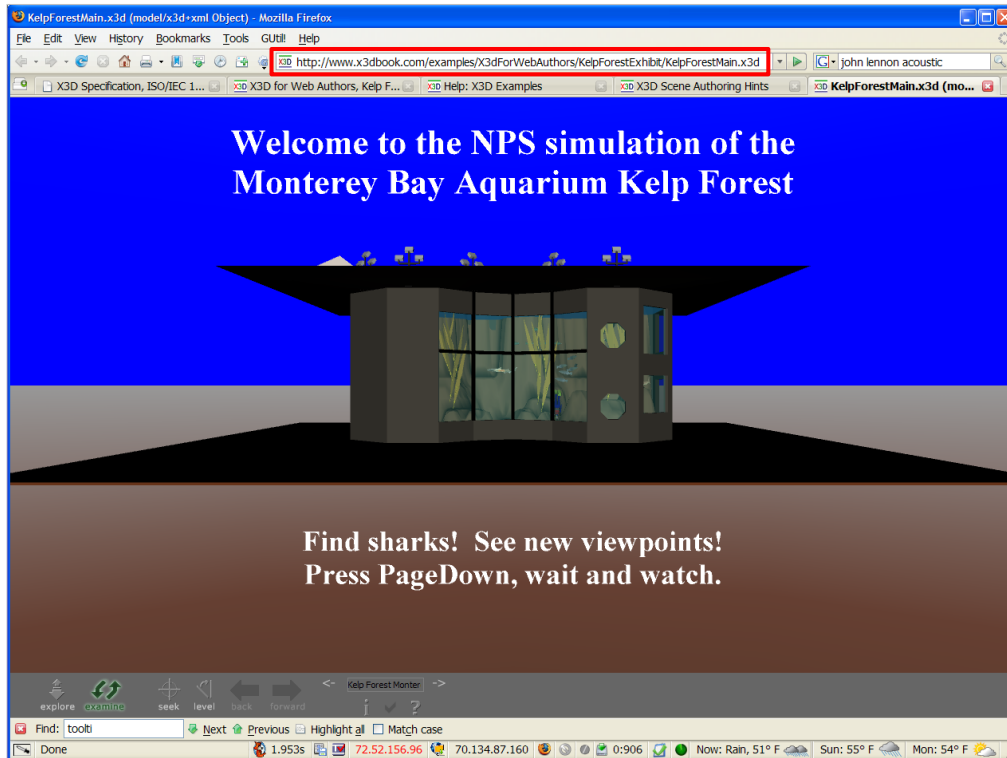
<http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/BindingOperations.x3d>



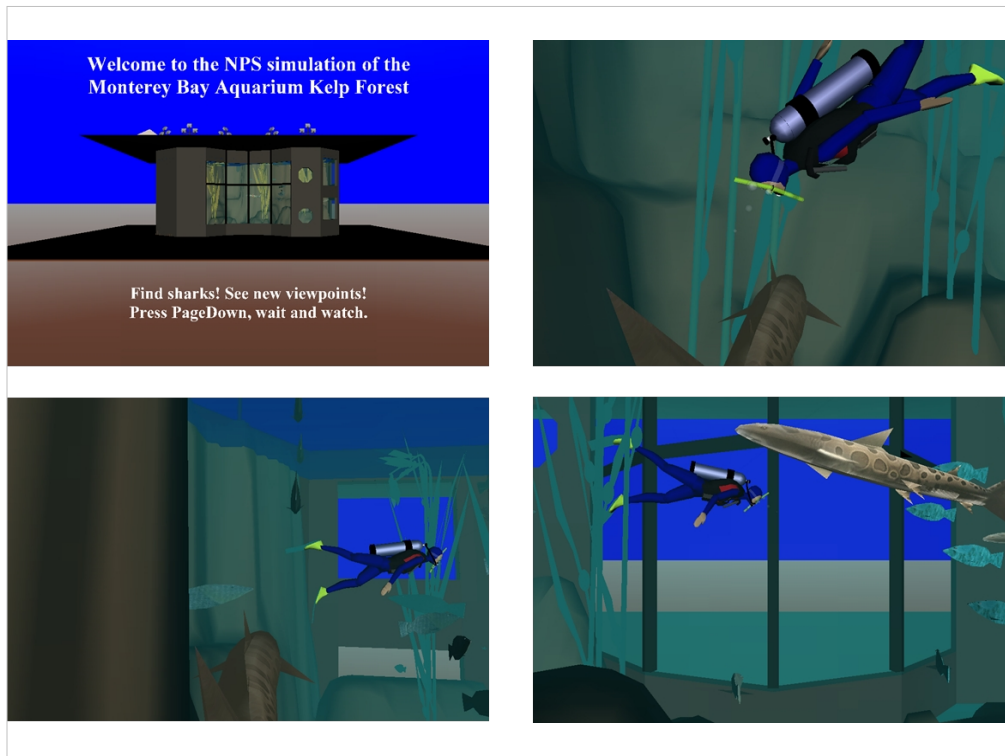


<http://www.x3dbook.com/examples/X3dForWebAuthors/KelpForestExhibit>  
then select Kelp Forest Main












 <b>Viewpoint</b>	Viewpoint provides a specific location and direction where the user may view the scene. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: USING other geometry (instead of duplicating nodes) can improve performance. <b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!
description	[description: accessType initializeOnly, type SFString CDATA #IMPLIED] Text description or navigation hint to be displayed for this Viewpoint. Hint: use spaces, make descriptions clear and readable. <b>Warning:</b> without description, Viewpoint is unlikely to appear on browser Viewpoints menu. Hint: many XML tools substitute XML character references automatically if needed (like &#38; for & or &#34; for "). Interchange profile hint: this field may be ignored.
position	[position: accessType inputOutput, type SFVec3f CDATA "0 0 10"] Position (x, y, z in meters) relative to local coordinate system.
orientation	[orientation: accessType inputOutput, type SFRotation CDATA "0 0 1 0"] Rotation (axis, angle in radians) of Viewpoint, relative to default -Z axis direction in local coordinate system. Hint: this is orientation_change_ from default direction (0 0 -1). Hint: complex rotations can be accomplished axis-by-axis using parent Transforms.
fieldOfView	[fieldOfView: accessType inputOutput, type SFFloat CDATA "0.785398" (0..pi)] Preferred minimum viewing angle from this viewpoint in radians. Small field of view roughly corresponds to a telephoto lens, large field of view roughly corresponds to a wide-angle lens. Hint: modifying Viewpoint distance to object may be better for zooming. <b>Warning:</b> fieldOfView may not be correct for different window sizes and aspect ratios. Interchange profile hint: this field may be ignored.
jump	[jump: accessType inputOutput, type SFBool (true/false) "true"] Transition instantly by jumping, or smoothly animate to this Viewpoint. Hint: set jump=true for smooth camera motion when going to this viewpoint.
centerOfRotation	[centerOfRotation: accessType inputOutput, type SFVec3f CDATA "0 0 0"] centerOfRotation point relates to NavigationInfo EXAMINE mode.
set_bind	[set_bind: accessType inputOnly, type SFBool (true/false) #FIXED ""] Sending event set_bind=true makes this node active. Sending event set_bind=false makes this node inactive. Thus setting set_bind to true/false will pop/push (enable/disable) this Viewpoint.
bindTime	[bindTime: accessType outputOnly, type SFTime CDATA #FIXED ""] Event sent when node becomes active/inactive.
isBound	[isBound: accessType outputOnly, type SFBool (true/false) #FIXED ""] Event true sent when node becomes active, event false sent when unbound by another node.
containerField	[containerField: NMTOKEN "children"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

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



# Navigation model 1

Users can select predefined Viewpoints

- Defines both position and direction of view

Users can further navigate around scene

- Using pointing device or hot keys
- Chosen viewpoint remains bound

Key	Emulated Action	WALK mode	FLY mode	EXAMINE mode
Up arrow	Pointer up	forward	forward	orbit up
Down arrow	Pointer down	backward	backward	orbit down
Left arrow	Pointer left	left	left	orbit left
Right arrow	Pointer right	right	right	orbit right

web|3D  
CONSORTIUM



Figure 4.9. Recommended Keyboard Navigation Keys and Responses



## Navigation model 2

User's current view can be animated

- ROUTE new position/direction event values to the Viewpoint itself, or to parent Transform nodes
- User navigation offsets to that view remain in effect
- Thus “over the shoulder” viewpoints can follow a moving object around, while still allowing user to look around while in that moving viewpoint

Lefty and Lucy shark in the Kelp Forest Main scene use this technique as virtual tour guides



## NavigationInfo node

NavigationInfo indicates how a browser might best support user navigation in the scene

Multiple NavigationInfo nodes may exist in scene

- Or in multiple Inline scenes loaded together

NavigationInfo is an X3DBindableNode

- So only one can be active at a given time
- Follow the same binding rules as Viewpoint, but not easily selectable
- Can be linked to a given Viewpoint by ROUTE that connects isBound of one node to set\_bind of other





## NavigationInfo *type*

Primary field is *type* which indicates which of the various modes of navigation are relevant

- "EXAMINE" best for rotating solitary objects
- "FLY" allows zooming in, out and around
- "WALK" also allows exploration, but on the ground
- "LOOKAT" use pointer to select geometry of interest
- "ANY" lets user select any mode
- "NONE" gives user zero control of navigation

MFString array default *type*=' "EXAMINE" "ANY" '

- which gives users plenty of flexibility





## NavigationInfo *type* details 1

- **"EXAMINE"** Used to view individual objects. Scene navigation consists of rotating the user viewpoint about the center of the observed object. The centerOfRotation field of the currently bound Viewpoint node values determines which local point centers the view rotation.
- **"WALK"** Used when exploring a virtual world on the ground. The user's eye level stays above the ground geometry and collision detection prevents the user from falling if underlying geometry is present.



## NavigationInfo *type* details 2

- **"FLY"** Similar to "WALK", but terrain following and collision detection is ignored. This type of navigation has the fewest constraints. Shifts the current view and related *centerOfRotation* values to track or zoom toward objects of interest to user.
- **"ANY"** Browser is allowed to provide whichever navigation type seems appropriate for the task at hand, modifying the user interface if necessary.
- **"NONE"** All navigation disabled and hidden. Navigation remains possible via animation of viewpoint fields or by binding other viewpoints (using viewpoint-list selection or Anchor node).





## NavigationInfo speed, headlight

*speed* determines how fast navigation occurs

- Default value 1 meter/second is usually pretty slow
- Might need to vary widely from ground to space

*headlight* is whether a light is shining ahead from user's point of view

- Otherwise one or more Light nodes is needed (covered in Chapter 11), or else world goes black



## NavigationInfo transitions

*transitionType* determines type of path followed when transitioning between viewpoints

- "ANIMATE" browser chooses smoothing algorithm
- "LINEAR" interpolation of position, orientation
- "TELEPORT" immediate repositioning to destination

*transitionTime*

- initial array value used for linear, otherwise multiple values can be used by browser-specific "ANIMATE"

*transitionComplete* lets author know when done

- SFBool boolean event sent when move is finished



*transitionType* is an inputOutput MFString array of quoted string values

*transitionTime* is an inputOutput MFBool boolean array of time intervals

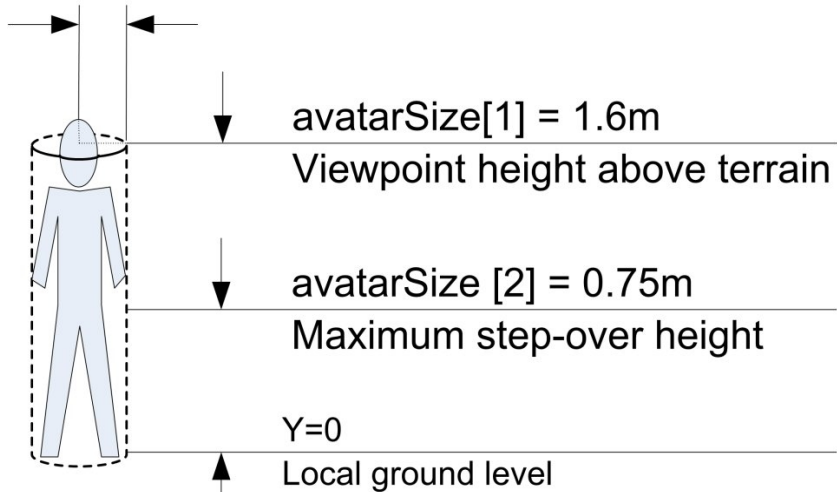
*transitionComplete* is an outputOnly SFTime value



## *avatarSize* SFVec3f array

`avatarSize[0] = 0.25m`

Allowed collision distance



Note that an SFVec3f is a 3-tuple with 3 component values



## NavigationInfo *visibilityLimit*

*visibilityLimit* defines the maximum range that may be rendered by the browser

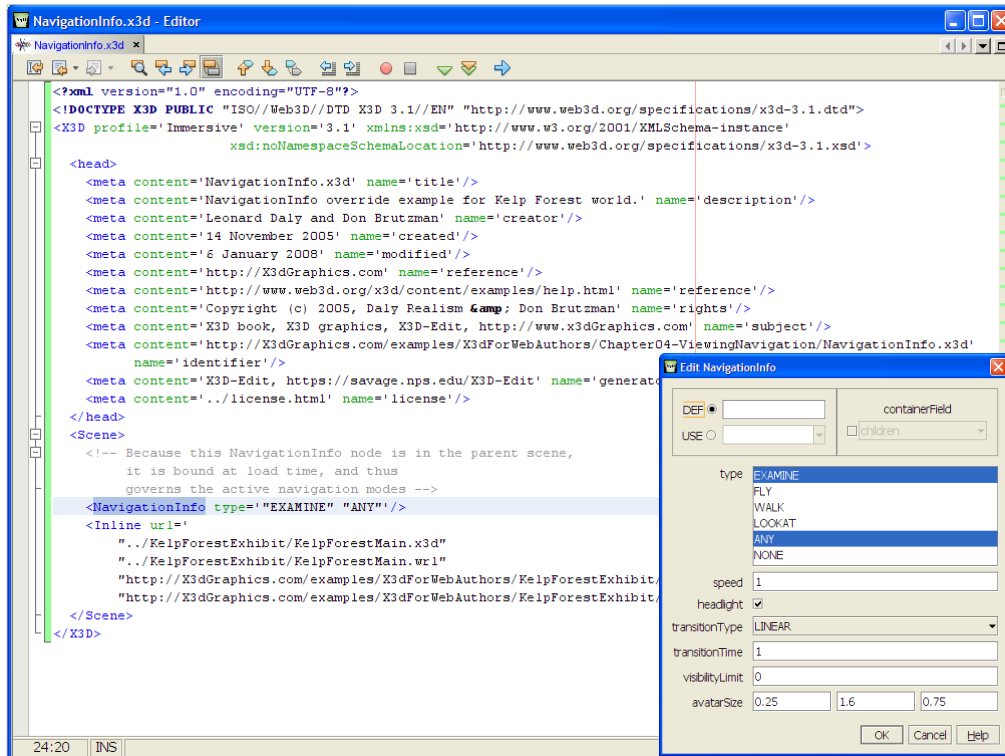
- Measured from the user's point of view
- Geometry beyond that distance are not drawn
- *visibilityLimit*='0.0' means no limits are imposed

Quality thumbrule: meet following relationship

- $avatarSize.collisionDistance / visibilityLimit < 10,000$
- Avoids floating-point roundoff error on graphics card and almost-coplanar polygon tearing/aliasing
- Exactly coplanar polygons still suffer from aliasing





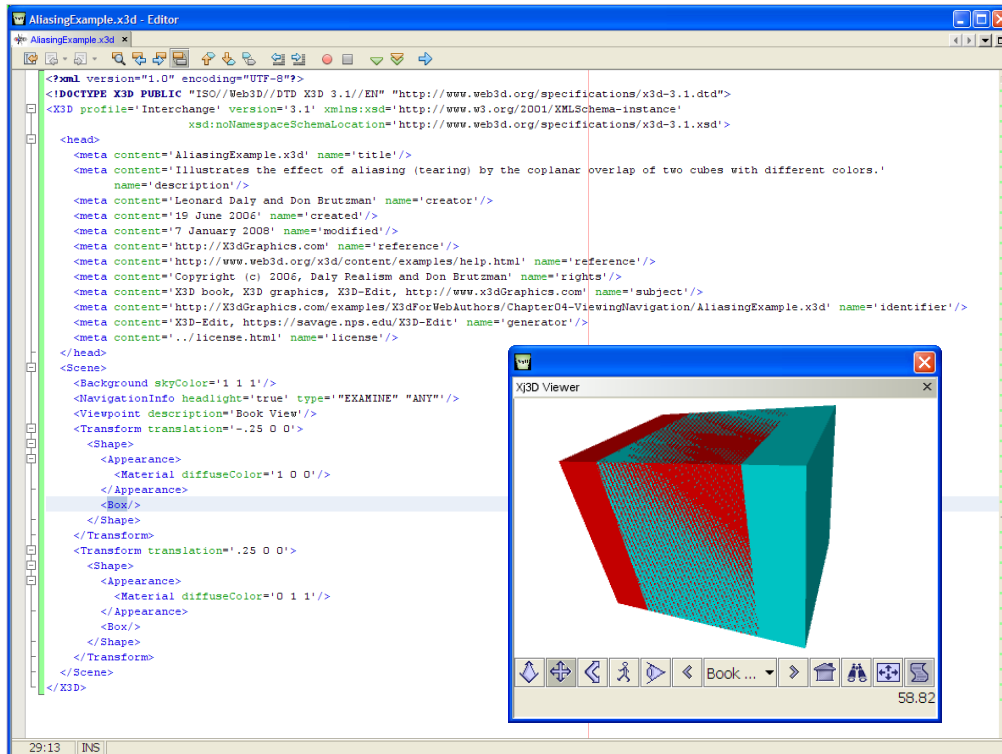




NavigationInfo	NavigationInfo describes the viewing model and physical characteristics of the viewer's avatar. Hint: for inspection of simple objects, usability often improves with type="EXAMINE" "ANY" Hint: NavigationInfo types "WALK" "FLY" support camera-to-object collision detection. Background, Fog, NavigationInfo, TextureBackground and Viewpoint are bindable nodes.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referenceable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring all other attributes and children. Hint: Using other geometry (instead of duplicating nodes) can improve performance. <b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!
type	[type: accessType inputOutput, type MFSString CDATA "EXAMINE" "ANY"] Enter one or more quoted Strings: "EXAMINE" "WALK" "FLY" "LOOKAT" "ANY" "NONE". Hint: for inspection of simple objects, usability often improves with type="EXAMINE" "ANY". Hint: types WALK and FLY force strict camera-to-object collision detection. Hint: see Collision node for further details on camera-to-object collision detection. Hint: Strings can have multiple values, so separate each string by quote marks [ "http://www.url1.org" "http://www.url2.org" "etc." ] Interchange profile hint: this field may be ignored.
speed	[speed: accessType inputOutput, type SFFloat CDATA "1.0"] [0..infinity) Default rate at which viewer travels through scene, meters/second. <b>Warning:</b> default 1 m/s usually seems slow for ordinary navigation. Interchange profile hint: this field may be ignored.
headlight	[headlight: accessType inputOutput, type SBool (true/false) "true"] Enable/disable directional light that always points in the direction the user is looking.
avatarSize	[avatarSize: accessType inputOutput, type MFFloat CDATA "0.25 1.6 0.75"] avatarSize triplet values are: (a) collision distance between user and geometry (near culling plane of the view frustum) (b) viewer height above terrain (c) tallest height viewer can WALK over. Hint: keep (visibilityLimit / avatarSize.CollisionDistance) < 10,000 to avoid aliasing artifacts (i.e. polygon "tearing"). Interchange profile hint: this field may be ignored.
visibilityLimit	[visibilityLimit: accessType inputOutput, type SFFloat CDATA "0.0"] Geometry beyond the visibilityLimit may not be rendered (far culling plane of the view frustum). visibilityLimit=0.0 indicates an infinite visibility limit. Hint: keep visibilityLimit >= zero. Hint: keep (visibilityLimit / avatarSize.CollisionDistance) < 10,000 to avoid aliasing artifacts (i.e. polygon "tearing"). Interchange profile hint: this field may be ignored.
transitionType	[transitionType: accessType inputOutput, type MFSString CDATA "ANIMATE"] Enter one or more quoted Strings: "ANIMATE" "LINEAR" "TELEPORT". Interchange profile hint: this field may be ignored.
transitionTime	[transitionTime: accessType inputOutput, type MFFloat CDATA "1.0"] Duration of viewpoint transition. Hint: If transitionType is "ANIMATE", transitionTime provides browser-dependent animation parameters. Interchange profile hint: this field may be ignored.
transitionComplete	[transitionComplete: accessType outputOnly, type MFFloat CDATA #FIXED ""] Event signaling viewpoint transition complete. Interchange profile hint: this field may be ignored.
set_bind	[set_bind: accessType inputOnly, type SBool (true/false) #FIXED ""] 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 ""] Event sent when node becomes active/inactive.
isBound	[isBound: accessType outputOnly, type SBool (true/false) #FIXED ""] Event true sent when node becomes active, event false sent when unbound by another node.
containerField	[containerField: NMTOKEN "children"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.

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





Aliasing (sometimes called tearing) occurs when the rendering engine is not able to detect which polygon is closer to the viewer. Here most of each unit box is directly superimposed on the other.

Moving the viewpoint varies the aliasing pattern a lot.

Moral of the story: don't do that, avoid coplanar polygons.

<http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/AliasingExample.x3d>



# Anchor node

Anchor is another grouping node that can contain other nodes

Geometry rendered by contained nodes is activated and can be selected

- Clicking on Anchor geometry launches url link
- Alternatively can select a viewpoint in the scene (similar to HTML bookmark)
- Thus similar to HTML anchor tag

Selected link can replace current X3D scene, or else can launch into another browser window





## Anchor *description*

The *description* field provides the user with a single-string summary of what is selected when the Anchor geometry is selected, e.g.

- *description*='click door, open portal to new world'
- *description*='jump to next viewpoint...'

X3D browsers usually pop up the text description when the pointing device is over the selection geometry



## *url* Uniform Resource Locator

The *url* field provides either

- Address to new X3D scene, HTML page, or another Web resource, or else
- Viewpoint bookmark within the scene

MFString array provides alternate url addresses

- url addresses can be either local or online
- Point to alternate versions of same resource
- X3D browser goes sequentially through ordered list, one at a time, until one retrieval succeeds





## Anchor *parameter*

*parameter* provides additional information to browser regarding redirection of loaded result

- *parameter*='target=\_blank' sends to new frame
- *parameter*='target=frame4' sends to named frame
- May be ignored if browser is solely X3D capable, rather than (for instance) a Web-browser plugin

Once again, designed to match HTML anchor tag





# Anchor hints and warnings

Strictly match capitalization of directories and filenames

- Unix and http are case sensitive and fail otherwise
- Windows is forgiving but actually this hides errors

XML escape characters

- & (ampersand)      &amp;
- ' (apostrophe)      &apos;
- " (double quote)      &quot;





```

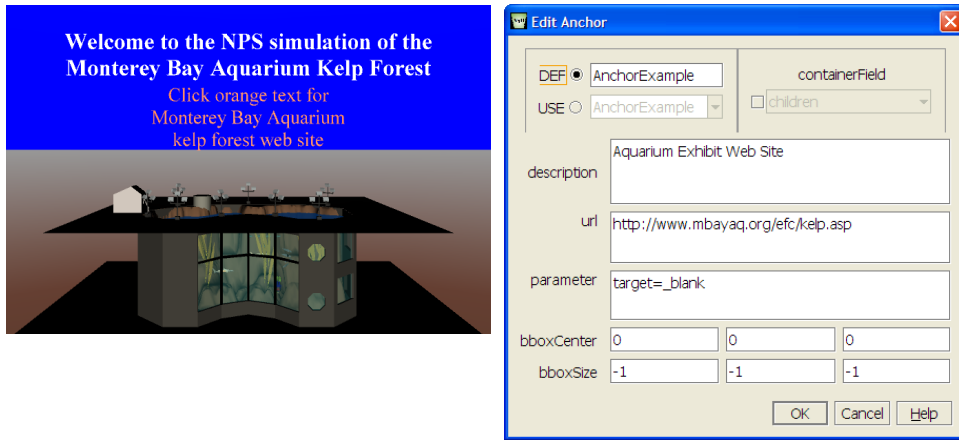
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile="Immersive" version="3.1" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
  xsd:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.1.xsd">
  <head>
    <meta content="Anchor.x3d" name="title"/>
    <meta content="Anchor example using the Kelp Forest world." name="description"/>
    <meta content="Leonard Daly and Don Brutzman" name="creator"/>
    <meta content="14 November 2005" name="created"/>
    <meta content="7 January 2007" name="modified"/>
    <meta content="Show Anchor link to Monterey Bay Aquarium web site for Kelp Forest exhibit" name="reference"/>
    <meta content="http://www.mbayaq.org/efc/kelp.asp" name="reference"/>
    <meta content="http://www.web3d.org/x3d/content/examples/help.html" name="reference"/>
    <meta content="Copyright (c) 2005, Daly Realism Camp; 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/Chapter04-ViewingNavigation/Anchor.x3d"
      name="identifier"/>
    <meta content="X3D-Edit, https://savage.nps.edu/X3D-Edit" name="generator"/>
    <meta content="../license.html" name="license"/>
  </head>
  <Scene>
    <Transform translation="0 8 30">
      <Viewpoint description="view Anchor text" position="0 -1 12"/>
      <Anchor DEF="AnchorExample" description="Aquarium Exhibit Web Site"
        parameter="target=_blank" url="http://www.mbayaq.org/efc/kelp.asp">
        <Shape bboxSize="-1 -1 -1">
          <Text string="Click orange text for " "Monterey Bay Aquarium" "Kelp forest web site""
            <FontStyle justify="MIDDLE" "MIDDLE" size="0.6"/>
          </Text>
          <Appearance>
            <Material DEF="Autumn11" ambientIntensity="0.25641" diffuseColor="0.795918 0.273554 0.006861"
              shininess="1" specularColor="0.48655 0.319155 0.444036"/>
          </Appearance>
        </Shape>
      </Anchor>
    </Transform>
    <Inline url="../KelpForestExhibit/KelpForestMain.x3d"
      "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d"
      "http://X3dGraphics.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.wrl"/>
  </Scene>

```

<http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/Anchor.x3d>



# Anchor example scene, editor



web|3D  
CONSORTIUM





⚓ Anchor	Anchor is a <i>Grouping</i> node that can contain most nodes. Clicking Anchored geometry loads content specified by the <code>url</code> field. Loaded content completely replaces current content, if parameter is same window. Hint: insert a <i>Shape</i> node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring <code>_all</code> other attributes and children. Hint: USEing other geometry (instead of duplicating nodes) can improve performance. <b>Warning:</b> do NOT include DEF (or any other attribute values) when using a USE attribute!
description	[description: accessType <i>inputOutput</i> , type <i>SFString CDATA</i> #IMPLIED] Text description to be displayed for action of this node. Hint: many XML tools substitute XML character references automatically if needed (like <code>&amp;#38;</code> for <code>&amp;</code> or <code>&amp;#34;</code> for <code>"</code> ). Interchange profile hint: this field may be ignored.
url	[url: accessType <i>inputOutput</i> , type <i>MFString CDATA</i> #IMPLIED] Address of replacement world, activated by clicking Anchor geometry. Hint: jump to a world's internal viewpoint by appending viewpoint name (e.g. <code>#ViewpointName, someOtherCoolWorld.wrl#GrandTour</code> ). Hint: jump to a local viewpoint by only using viewpoint name (e.g. <code>#GrandTour</code> ). Hint: Strings can have multiple values, so separate each string by quote marks [ <code>"http://www.url1.org" "http://www.url2.org" "etc."</code> ]. Hint: XML encoding for <code>"</code> is <code>&amp;quot;</code> (a character entity). <b>Warning:</b> strictly match directory and filename capitalization for http links! Hint: can replace embedded blank(s) in url queries with <code>%20</code> for each blank character. Hint: pop up a new window with url value as follows: <code>JavaScript:window.open('popup.html','popup','width=240,height=240',location.href='HelloWorld.wrl')</code>
parameter	[parameter: accessType <i>inputOutput</i> , type <i>MFString CDATA</i> #IMPLIED] Passed parameter that signals web browser how to redirect url loading. Hint: set parameter to <code>target=_blank</code> to load target url into a blank frame. Hint: set parameter to <code>target=frame_name</code> to load target url into another frame. Hint: Strings can have multiple values, so separate each string by quote marks. [ <code>"http://www.url1.org" "http://www.url2.org" "etc."</code> ]. Interchange profile hint: this field may be ignored.
bboxCenter	[bboxCenter: accessType <i>initializeOnly</i> , type <i>SFVec3f CDATA</i> "0 0 0"] Bounding box center: position offset from origin of local coordinate system.
bboxSize	[bboxSize: accessType <i>initializeOnly</i> , type <i>SFVec3f CDATA</i> "-1 -1 -1"] Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	[containerField: NMTOKEN "children"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

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



# Billboard node

Billboard is another X3DGroupingNode

Child-content geometry faces user

- Special effect that improves readability or visibility

*axisOfRotation* field determines Billboard pivot

- Relative to local coordinate system
- Default is *axisOfRotation*='0 1 0' which swivels about vertical (Y axis)
- Rotations unpredictable when above (on axis)
- Define *axisOfRotation*='0 0 0' for circular rotation to always fully face user without axis constraints



## Billboard hints and warnings

DEF, USE allowed for multiple Billboards nodes

- Each copy should independently face user

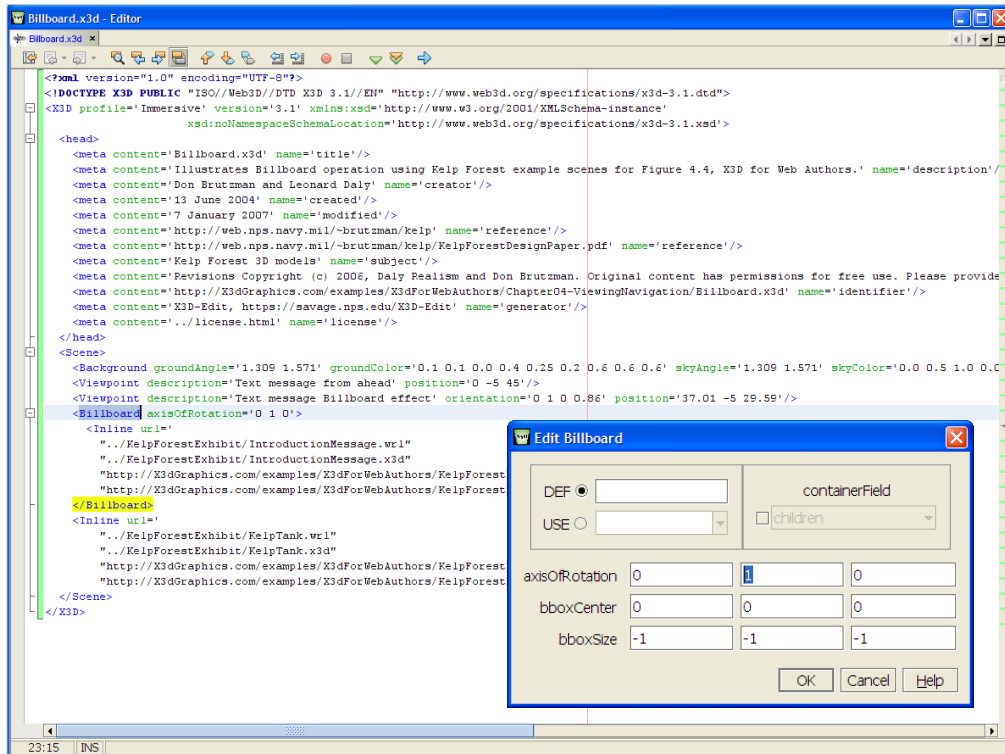
Put Billboard as close to moving geometry as possible, nested inside a positioning Transform

Do not put a Viewpoint under a Billboard

- Creates a feedback loop
- Unpredictable behavior likely to result







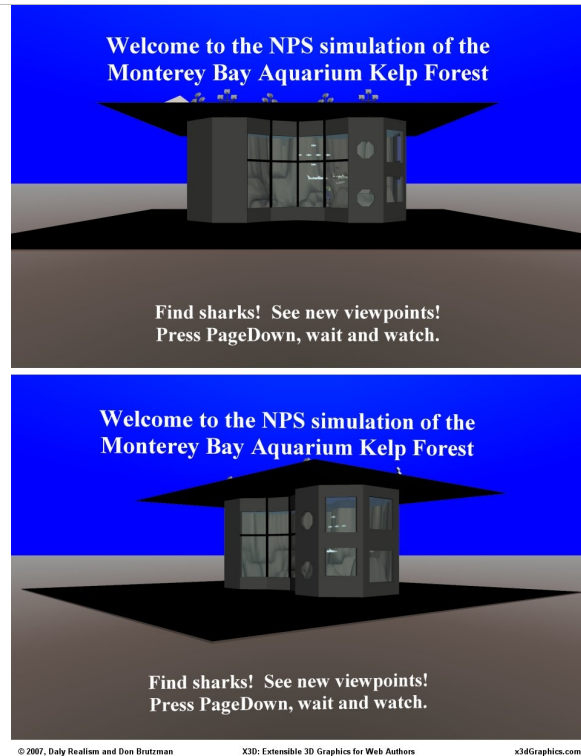
<http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/Billboard.x3d>



## Billboard example


Starting at initial viewpoint and navigating with mouse or arrow keys reveals that Billboard Text remains facing the viewer, improving readability

web|3D  
CONSORTIUM



<http://www.x3dbook.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d>



 Billboard	Billboard is a Grouping node that can contain most nodes. Content faces the user, rotating about the specified axis. Set axisOfRotation=0 0 0 to fully face the user's camera. Hint: Put Billboard as close to the geometry as possible, nested inside Transform for local coordinate system. Hint: don't put Viewpoint inside a Billboard. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referenceable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring _all_ other attributes and children. Hint: Using other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
axisOfRotation	[axisOfRotation: accessType inputOutput, type SFVec3f CDATA "0 1 0"] axisOfRotation direction is relative to local coordinate system. Hint: axis 0 0 0 always faces viewer.
bboxCenter	[bboxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"] Bounding box center: position offset from origin of local coordinate system.
bboxSize	[bboxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"] Bounding box size: automatically calculated, can be specified as an optimization or constraint.
containerField	[containerField: NMTOKEN "children"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

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



## Collision node

Defines camera-to-object collision-detection properties between child geometry and user

- *enabled*='true' blocks user navigation through the geometry
- *enabled*='false' blocks user navigation through the geometry

Not used for object-to-object collision detection

Authors can detect when collision occurs

- SFTIME outputOnly event *collideTime*
- SFBool outputOnly event *isActive*



## Collision detection and terrain following

Terrain following depends on +Y axis being “up”

- Other coordinate systems are possible but do not match this X3D convention
- Thus datasets using other coordinates must be converted to match

WALK mode is another form of collision detection

- Viewer's camera drops until NavigationInfo avatar rests on geometry serving as the ground plane
- Step-over distance (an avatarSize parameter) governs whether user can rise over obstacles



## Collision *proxy* field

Child geometry may be quite detailed, irregular

- Complicating collision-detection calculations and thus slowing rendering performance

Can substitute SFNode *proxy* child as alternate

- Shape containing a Box, Sphere or Cylinder can provide simplifying geometric alternative
- *proxy* geometry is not rendered

```
<Collision DEF='Example' enabled='true'>  
  <Shape containerField='proxy'><Cylinder/></Shape>  
  <Inline url='SomeComplicatedObject.x3d' />  
</Collision>
```





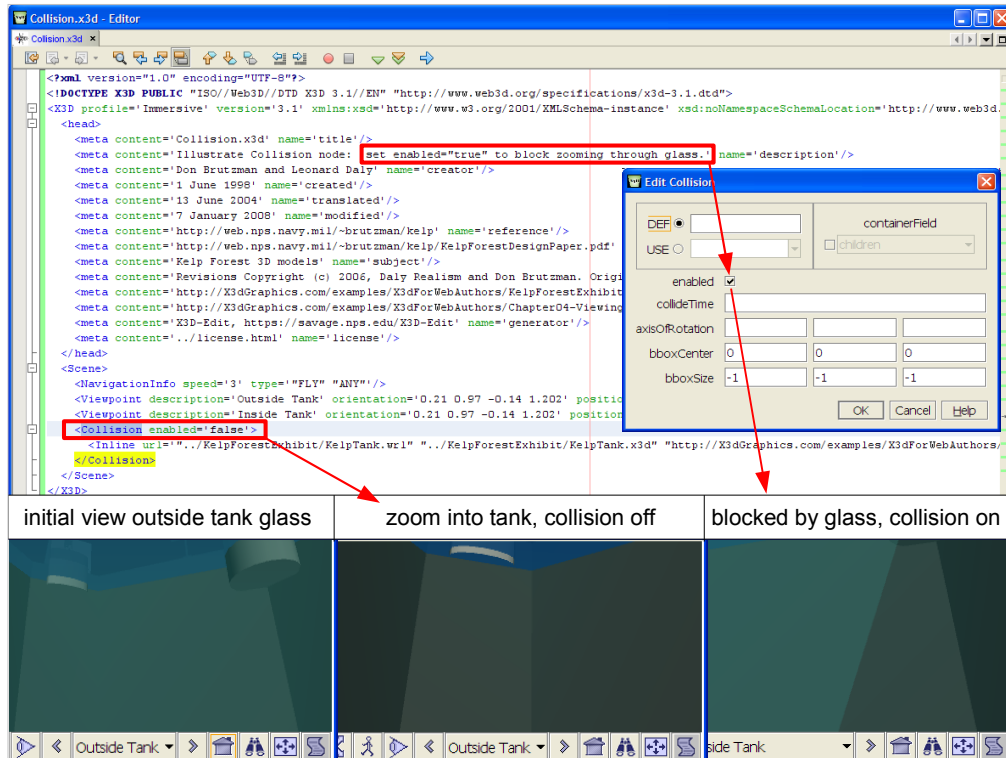
## Collision hints and warnings

`<NavigationInfo type=' "WALK" "FLY" ' />`  
modes support camera-to-object collision  
detection

Only polygonal geometry can be used for  
collision detection

- No points or lines
- Special limitation: no Text node collisions
- Nevertheless can achieve same collision effects  
by adding a transparent Box or polygon, thus  
providing the necessary polygons as boundaries





<http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/Collision.x3d>



## Collision example

Example screen shots  
first show the viewer  
being stopped by  
glass geometry, then  
the viewer passing  
through the tank glass  
for a closer view.

Collision *enabled*="true"  
or *enabled*="false"  
result in different  
navigation responses.

web|3D  
CONSORTIUM




© 2007, Daly Realism and Don Brutzman

X3D: Extensible 3D Graphics for Web Authors

x3dGraphics.com

*X3D for Web Authors*, Figure 4.5



 Collision	Collision detects camera-to-object contact using current Viewpoint and NavigationInfo avatarSize. Collision is a Grouping node that handles collision detection for its children. Collision can contain a single proxy child node for substitute collision-detection geometry. Note: proxy geometry is not rendered. Note: PointSet, IndexedLineSet, LineSet and Text do not trigger collisions. Hint: improve performance using proxy for simpler contact-calculation geometry. Hint: NavigationInfo types "WALK" "FLY" support camera-to-object collision detection. Hint: insert a Shape node before adding geometry or Appearance.
DEF	[DEF ID #IMPLIED] DEF defines a unique ID name for this node, referencable by other nodes. Hint: descriptive DEF names improve clarity and help document a model.
USE	[USE IDREF #IMPLIED] USE means reuse an already DEF-ed node ID, ignoring all other attributes and children. Hint: USING other geometry (instead of duplicating nodes) can improve performance. Warning: do NOT include DEF (or any other attribute values) when using a USE attribute!
boxCenter	[boxCenter: accessType initializeOnly, type SFVec3f CDATA "0 0 0"] Bounding box center: position offset from origin of local coordinate system.
boxSize	[boxSize: accessType initializeOnly, type SFVec3f CDATA "-1 -1 -1"] Bounding box size: automatically calculated, can be specified as an optimization or constraint.
enabled	[enabled: accessType inputOutput, type SFBool (true/false) "true"] Enables/disables collision detection for children and all descendants. Hint: former name "collide" in VRML 97 specification.
isActive	[isActive: accessType outputOnly, type SFBool (true/false) #FIXED ""] isActive true/false events are sent when triggering the sensor; isActive=true when view-object collision occurs, isActive=false when view-object collision no longer occurs.
collideTime	[collideTime: accessType outputOnly, type SFTime CDATA #FIXED ""] Time of collision between camera (avatar) and geometry.
containerField	[containerField: NMTOKEN "children"] containerField is the field-label prefix indicating relationship to parent node. Examples: geometry Box, children Group, proxy Shape. containerField attribute is only supported in XML encoding of X3D scenes.
class	[class CDATA #IMPLIED] class is a space-separated list of classes, reserved for use by XML stylesheets. class attribute is only supported in XML encoding of X3D scenes.

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



[back to Table of Contents](#)

## Additional Resources





# File formatting

X3D-Edit has a Netbeans capability for formatting

- *Alt-shift-F **Format*** acts upon highlighted text blocks, also available via right-click menu
- Warning: do not reformat embedded ECMAScript source code

X3D Canonicalization (C14N) also reformats X3D

- Performed prior to examples being placed in archive
- Will be added to X3D-Edit Tools menu



Source Editing in Netbeans 6.0

<http://www.netbeans.org/kb/60/java/editor-tips.html>

X3D Canonical Form

<http://www.web3d.org/x3d/specifications/ISO-IEC-19776-3-X3DEncodings-CompressedBinary/Part03/concepts.html#X3DCanonicalForm>

Issue-tracker entry for C14N addition to X3D-Edit

[https://www.movesinstitute.org/bugzilla/show\\_bug.cgi?id=1461](https://www.movesinstitute.org/bugzilla/show_bug.cgi?id=1461)



## Pretty-print HTML capabilities

*Pretty print* means to reformat nicely in HTML, usually with color coding

- facilitates reading and printing

X3D-Edit has this Netbeans feature

- File > Print to HTML

X3dToXhtml.xslt stylesheet

- Includes indices and hyperlinks to DEF/USE, ROUTEs, images, url values, prototypes, etc.
- Can be launched via XSL Transformation button
- Will be added to X3D-Edit Tools menu



Tagset pretty-printing in XHTML (.html encoding), includes cross linking of DEF/USE/ROUTE/etc.: X3dToXhtml.xslt and X3dToXhtml.bat

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

Issue-tracker entry for X3dToXhtml.xslt addition to X3D-Edit

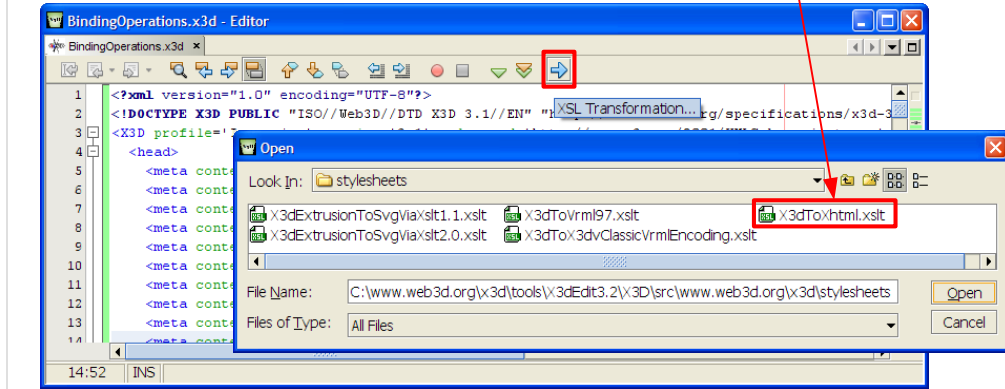
[https://www.movesinstitute.org/bugzilla/show\\_bug.cgi?id=1549](https://www.movesinstitute.org/bugzilla/show_bug.cgi?id=1549)



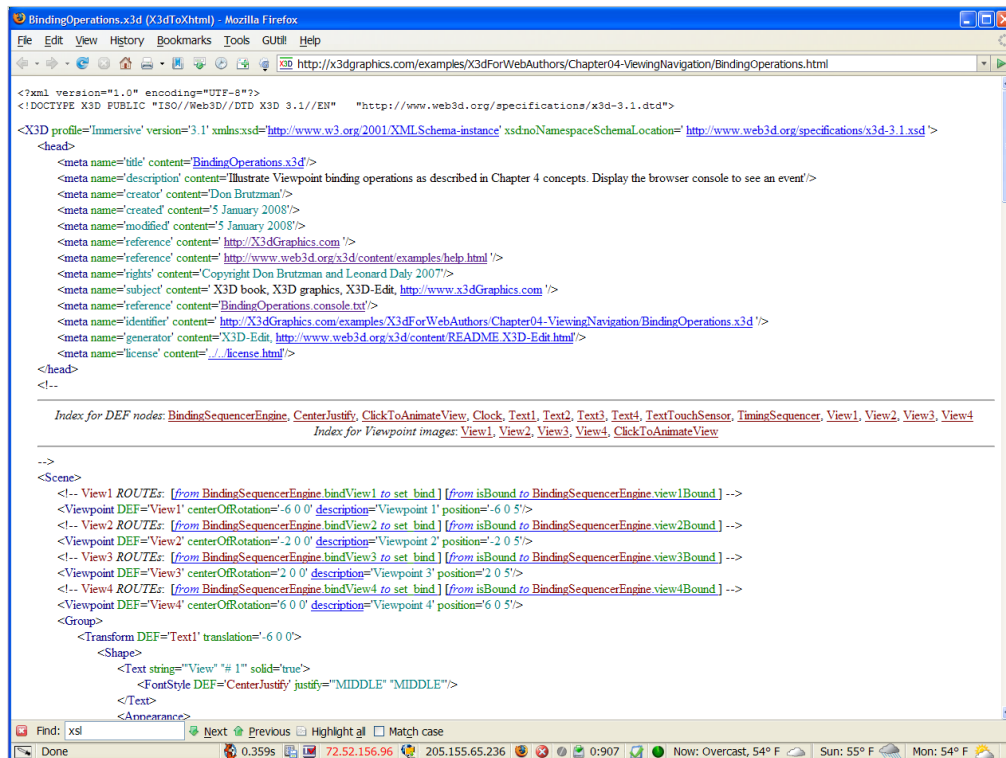
# Launching other XSLT stylesheets

Extensible Stylesheet Language for XML (XSLT) stylesheets support a variety of conversions

- X3dToVrmI97.xslt
- X3dToClassicVRML.xslt
- X3dToXhtml.xslt
- others







```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D/DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile="Immersive" version="3.1" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" xsd:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.1.xsd">
  <head>
    <meta name="title" content="BindingOperations x3d"/>
    <meta name="description" content="Illustrate Viewpoint binding operations as described in Chapter 4 concepts. Display the browser console to see an event"/>
    <meta name="creator" content="Don Brutzman"/>
    <meta name="created" content="5 January 2008"/>
    <meta name="modified" content="5 January 2008"/>
    <meta name="reference" content="http://X3dGraphics.com"/>
    <meta name="reference" content="http://www.web3d.org/x3d/content/examples/help.html"/>
    <meta name="rights" content="Copyright Don Brutzman and Leonard Daly 2007"/>
    <meta name="subject" content="X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com"/>
    <meta name="reference" content="BindingOperations console.txt"/>
    <meta name="identifier" content="http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/BindingOperations x3d"/>
    <meta name="generator" content="X3D-Edit, http://www.web3d.org/x3d/content/README.X3D-Edit.html"/>
    <meta name="license" content="./../license.html"/>
  </head>
  <!--
    Index for DEF nodes: BindingSequencerEngine, CenterJustify, ClickToAnimateView, Clock, Text1, Text2, Text3, Text4, TextTouchSensor, TimingSequencer, View1, View2, View3, View4
    Index for Viewpoint images: View1, View2, View3, View4, ClickToAnimateView
  -->
  <Scene>
    <!-- View1 ROUTEs: [from BindingSequencerEngine bindView1 to set_bind] [from isBound to BindingSequencerEngine view1Bound] -->
    <Viewpoint DEF=View1 centerOfRotation=-6 0 0 description=Viewpoint 1 position=-6 0 5/>
    <!-- View2 ROUTEs: [from BindingSequencerEngine bindView2 to set_bind] [from isBound to BindingSequencerEngine view2Bound] -->
    <Viewpoint DEF=View2 centerOfRotation=-2 0 0 description=Viewpoint 2 position=-2 0 5/>
    <!-- View3 ROUTEs: [from BindingSequencerEngine bindView3 to set_bind] [from isBound to BindingSequencerEngine view3Bound] -->
    <Viewpoint DEF=View3 centerOfRotation=2 0 0 description=Viewpoint 3 position=2 0 5/>
    <!-- View4 ROUTEs: [from BindingSequencerEngine bindView4 to set_bind] [from isBound to BindingSequencerEngine view4Bound] -->
    <Viewpoint DEF=View4 centerOfRotation=6 0 0 description=Viewpoint 4 position=6 0 5/>
    <Group>
      <Transform DEF=Text1 translation=-6 0 0>
        <Shape>
          <Text string="View" "# 1" solid=true>
            <FontStyle DEF=CenterJustify justify="MIDDLE" "MIDDLE"/>
          </Text>
        </Appearance>
      </Transform>
    </Group>
  </Scene>

```

<http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter04-ViewingNavigation/BindingOperations.html>



[back to Table of Contents](#)

## Chapter Summary





## Chapter Summary

Users explore X3D worlds by choosing predefined viewpoints and **navigating** through 3D space.

- **Bindable nodes**, so only one is active at a time
- **Viewpoint** lets authors identify key camera locations
- **NavigationInfo** provides options for moving around

Nodes to improve user navigability, interaction:

- **Anchor** makes geometric shapes linkable, like HTML
- **Billboard** for axis-aligned geometry facing the user
- **Collision** permits or blocks a user's current camera view from passing through collidable geometry





[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 4, Viewing and Navigation
- <http://x3dGraphics.com>
- <http://x3dgraphics.com/examples/X3dForWebAuthors>

## X3D Examples Help

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





## References 2

### X3D Scene Authoring Hints

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

### X3D Graphics Specification

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





## References 3

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

- <http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm>
- <http://www.web3d.org/x3d/content/examples/Vrml2.0Sourcebook>
- Chapter 26 – Viewpoint





# Contact

**Don Brutzman**

*[brutzman@nps.edu](mailto:brutzman@nps.edu)*

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

Code USW/Br, Naval Postgraduate School

Monterey California 93943-5000 USA

1.831.656.2149 voice

1.831.656.7599 fax





# Open-source license

Copyright (c) 1995-2008 held by the author(s). All rights reserved.

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

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

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