X3D Graphics for Web Authors

X3D Scene Graph Tutorial

Plus ça change, plus c'est la même chose.

The more something changes, the more it's the same thing.

web3D CONSORTIUM

Tutorial Contents

X3D Scene Graph Introduction

1. Technical Overview
2. Shape and Geometry
3. Grouping and Transformation
4. Viewing and Navigation
5. Appearance, Material and Textures

- Animation Behavior Examples

Tutorial Summary

References

Technical Overview

Historical background: VRML

Virtual Reality Modeling Language (VRML) began in 1994, seeking to create 3D markup for Web

- Numerous candidates considered by an open community of interested practitioners
- SGI's OpenInventor won the initial competition
- VRML 1.0 developed over the next year
- VRML 2.0 restructured some nodes, added features

VRML advanced to International Standard 14772 by ISO in 1997

Web3D Consortium

Web3D Consortium founded in 1998 to protect, support and advance the VRML specification

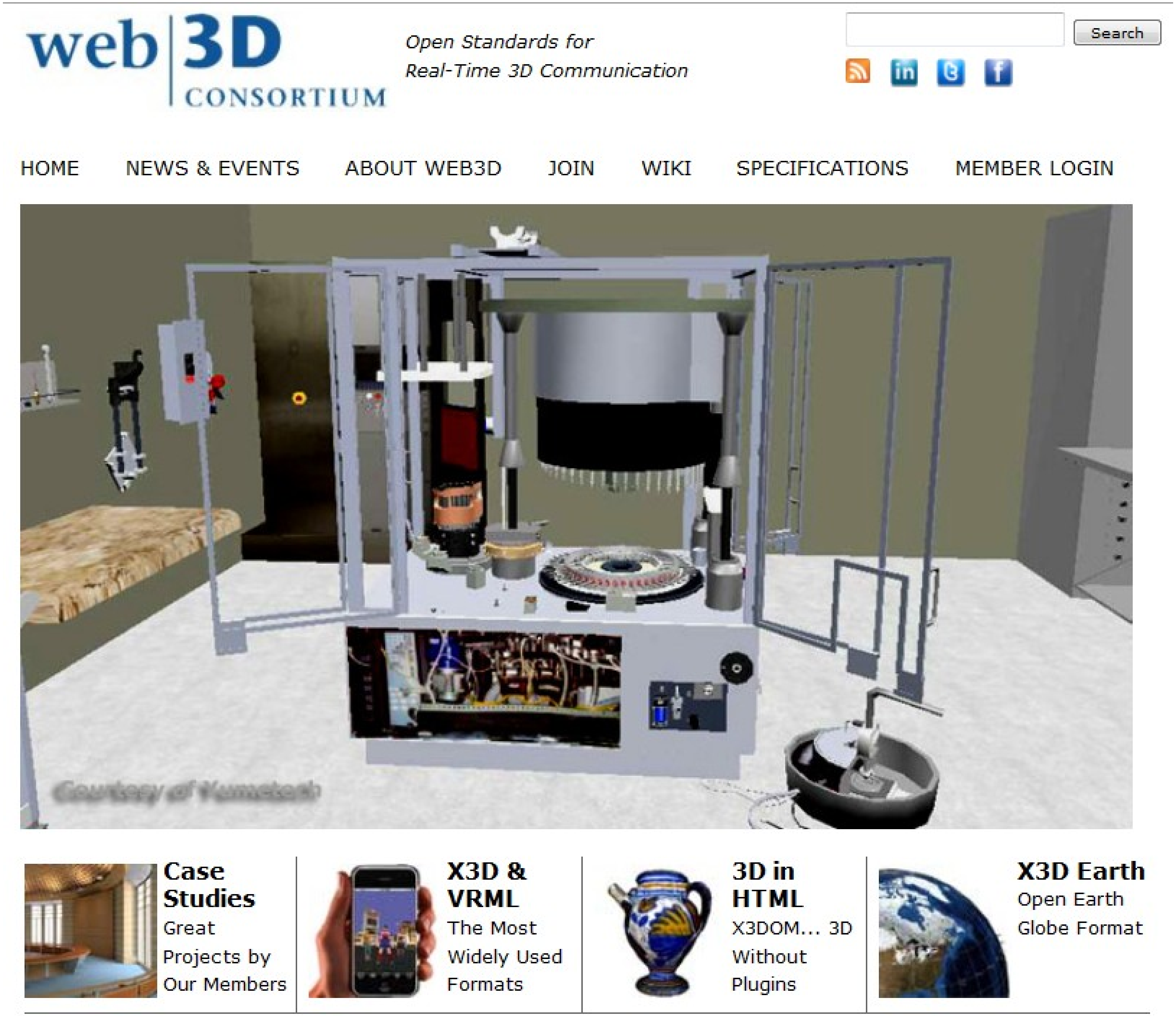
- http://www.web3D.org

Continued efforts on new technology by multiple working groups led its successor, X3D

- http://www.web3D.org/x3d

Non-profit organization of many stakeholders ensures that X3D remains royalty free, relevant

- Partnership of industry, agency, academic and professional members



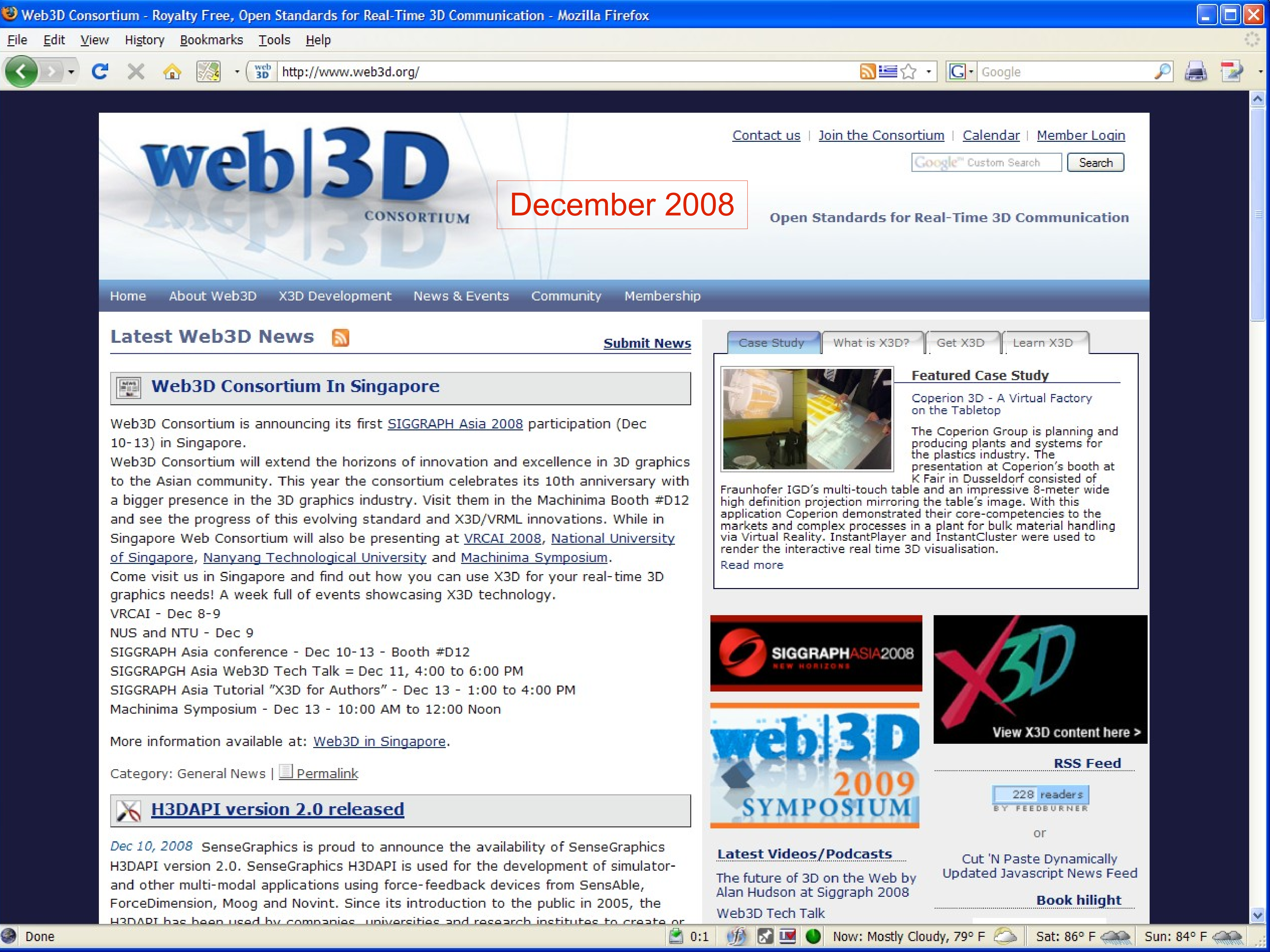
Open Standards for
Real-Time 3D Communication

Case Studies
Great Projects by Our Members

X3D & VRML
The Most Widely Used Formats

3D in HTML
X3DOM... 3D Without Plugins

X3D Earth
Open Earth Globe Format



Web3D Consortium - Royalty Free, Open Standards for Real-Time 3D Communication - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.web3d.org/
Google

December 2008

Contact us | Join the Consortium | Calendar | Member Login
Google Custom Search
Search

Open Standards for Real-Time 3D Communication

Home
About Web3D
X3D Development
News & Events
Community
Membership

Latest Web3D News

Submit News

Web3D Consortium In Singapore

Web3D Consortium is announcing its first SIGGRAPH Asia 2008 participation (Dec 10-13) in Singapore.

Web3D Consortium will extend the horizons of innovation and excellence in 3D graphics to the Asian community. This year the consortium celebrates its 10th anniversary with a bigger presence in the 3D graphics industry. Visit them in the Machinima Booth #D12 and see the progress of this evolving standard and X3D/VRML innovations. While in Singapore Web Consortium will also be presenting at VRCAI 2008, National University of Singapore, Nanyang Technological University and Machinima Symposium.

Come visit us in Singapore and find out how you can use X3D for your real-time 3D graphics needs! A week full of events showcasing X3D technology.

VRCAI - Dec 8-9
NUS and NTU - Dec 9
SIGGRAPH Asia conference - Dec 10-13 - Booth #D12
SIGGRAPGH Asia Web3D Tech Talk = Dec 11, 4:00 to 6:00 PM
SIGGRAPH Asia Tutorial "X3D for Authors" - Dec 13 - 1:00 to 4:00 PM
Machinima Symposium - Dec 13 - 10:00 AM to 12:00 Noon

More information available at: Web3D in Singapore.

Category: General News | Permalink

H3DAPI version 2.0 released

Dec 10, 2008 SenseGraphics is proud to announce the availability of SenseGraphics H3DAPI version 2.0. SenseGraphics H3DAPI is used for the development of simulator- and other multi-modal applications using force-feedback devices from SensAble, ForceDimension, Moog and Novint. Since its introduction to the public in 2005, the H3DAPI has been used by companies, universities and research institutes to create or

Featured Case Study

Coperion 3D - A Virtual Factory on the Tabletop

The Coperion Group is planning and producing plants and systems for the plastics industry. The presentation at Coperion's booth at K Fair in Dusseldorf consisted of Fraunhofer IGD's multi-touch table and an impressive 8-meter wide high definition projection mirroring the table's image. With this application Coperion demonstrated their core-competencies to the markets and complex processes in a plant for bulk material handling via Virtual Reality. InstantPlayer and InstantCluster were used to render the interactive real time 3D visualisation.

Read more

RSS Feed

or

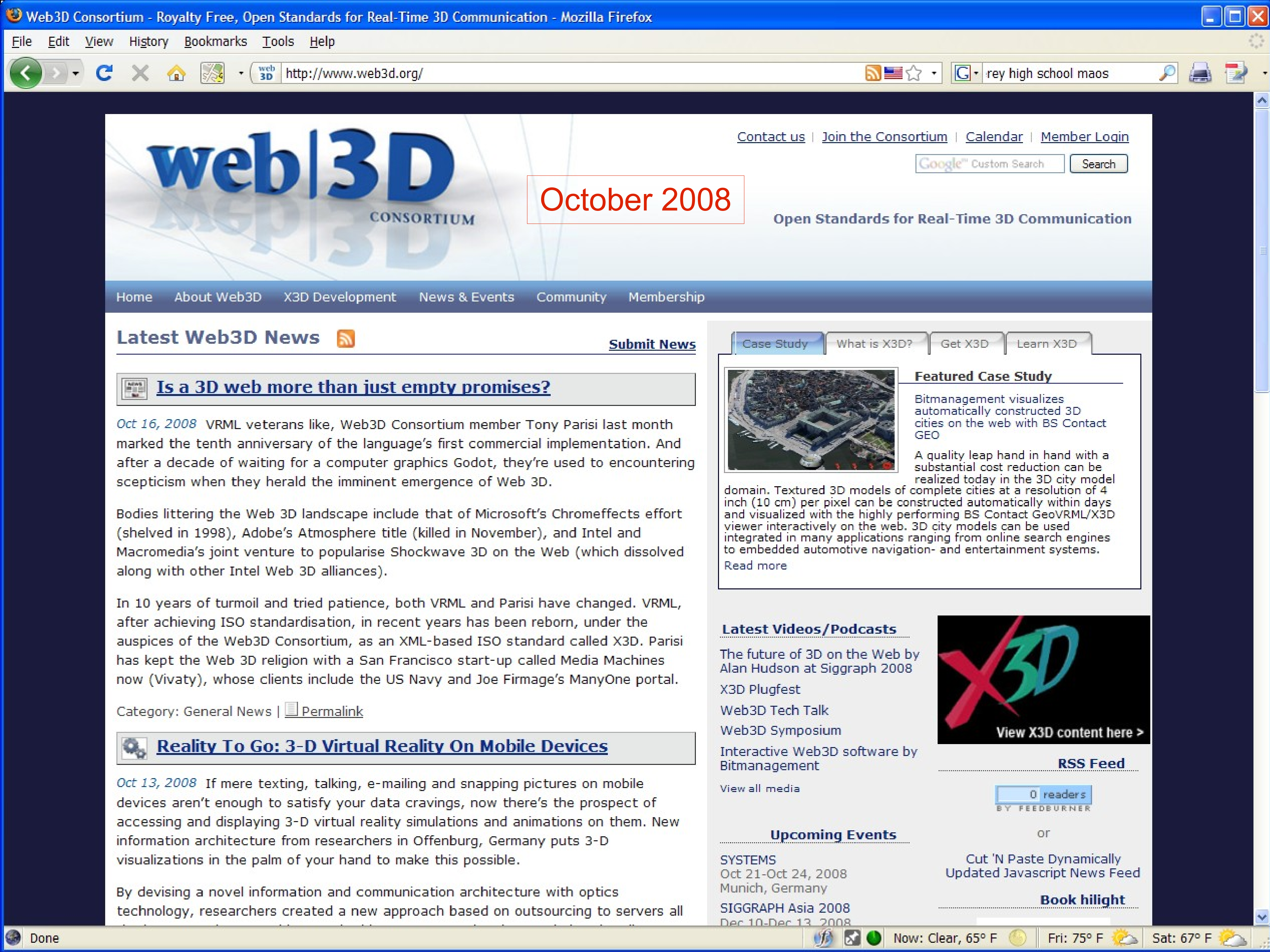
Cut 'N Paste Dynamically Updated Javascript News Feed

Book hilight

Latest Videos/Podcasts

The future of 3D on the Web by Alan Hudson at Siggraph 2008

Web3D Tech Talk



Web3D Consortium - Royalty Free, Open Standards for Real-Time 3D Communication - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.web3d.org/
rey high school maos

October 2008

web|3D CONSORTIUM
Contact us | Join the Consortium | Calendar | Member Login
Google Custom Search
Search
Open Standards for Real-Time 3D Communication
Home
About Web3D
X3D Development
News & Events
Community
Membership

Latest Web3D News

Submit News

Is a 3D web more than just empty promises?

Oct 16, 2008 VRML veterans like, Web3D Consortium member Tony Parisi last month marked the tenth anniversary of the language's first commercial implementation. And after a decade of waiting for a computer graphics Godot, they're used to encountering scepticism when they herald the imminent emergence of Web 3D.

Bodies littering the Web 3D landscape include that of Microsoft's Chromeffects effort (shelved in 1998), Adobe's Atmosphere title (killed in November), and Intel and Macromedia's joint venture to popularise Shockwave 3D on the Web (which dissolved along with other Intel Web 3D alliances).

In 10 years of turmoil and tried patience, both VRML and Parisi have changed. VRML, after achieving ISO standardisation, in recent years has been reborn, under the auspices of the Web3D Consortium, as an XML-based ISO standard called X3D. Parisi has kept the Web 3D religion with a San Francisco start-up called Media Machines now (Vivaty), whose clients include the US Navy and Joe Firmage's ManyOne portal.

Category: General News | Permalink

Reality To Go: 3-D Virtual Reality On Mobile Devices

Oct 13, 2008 If mere texting, talking, e-mailing and snapping pictures on mobile devices aren't enough to satisfy your data cravings, now there's the prospect of accessing and displaying 3-D virtual reality simulations and animations on them. New information architecture from researchers in Offenburg, Germany puts 3-D visualizations in the palm of your hand to make this possible.

By devising a novel information and communication architecture with optics technology, researchers created a new approach based on outsourcing to servers all

Featured Case Study

Bitmanagement visualizes automatically constructed 3D cities on the web with BS Contact GEO

A quality leap hand in hand with a substantial cost reduction can be realized today in the 3D city model domain. Textured 3D models of complete cities at a resolution of 4 inch (10 cm) per pixel can be constructed automatically within days and visualized with the highly performing BS Contact GeoVRML/X3D viewer interactively on the web. 3D city models can be used integrated in many applications ranging from online search engines to embedded automotive navigation- and entertainment systems.

Read more

Latest Videos/Podcasts

The future of 3D on the Web by Alan Hudson at Siggraph 2008

X3D Plugfest

Web3D Tech Talk

Web3D Symposium

Interactive Web3D software by Bitmanagement

View all media

RSS Feed

or

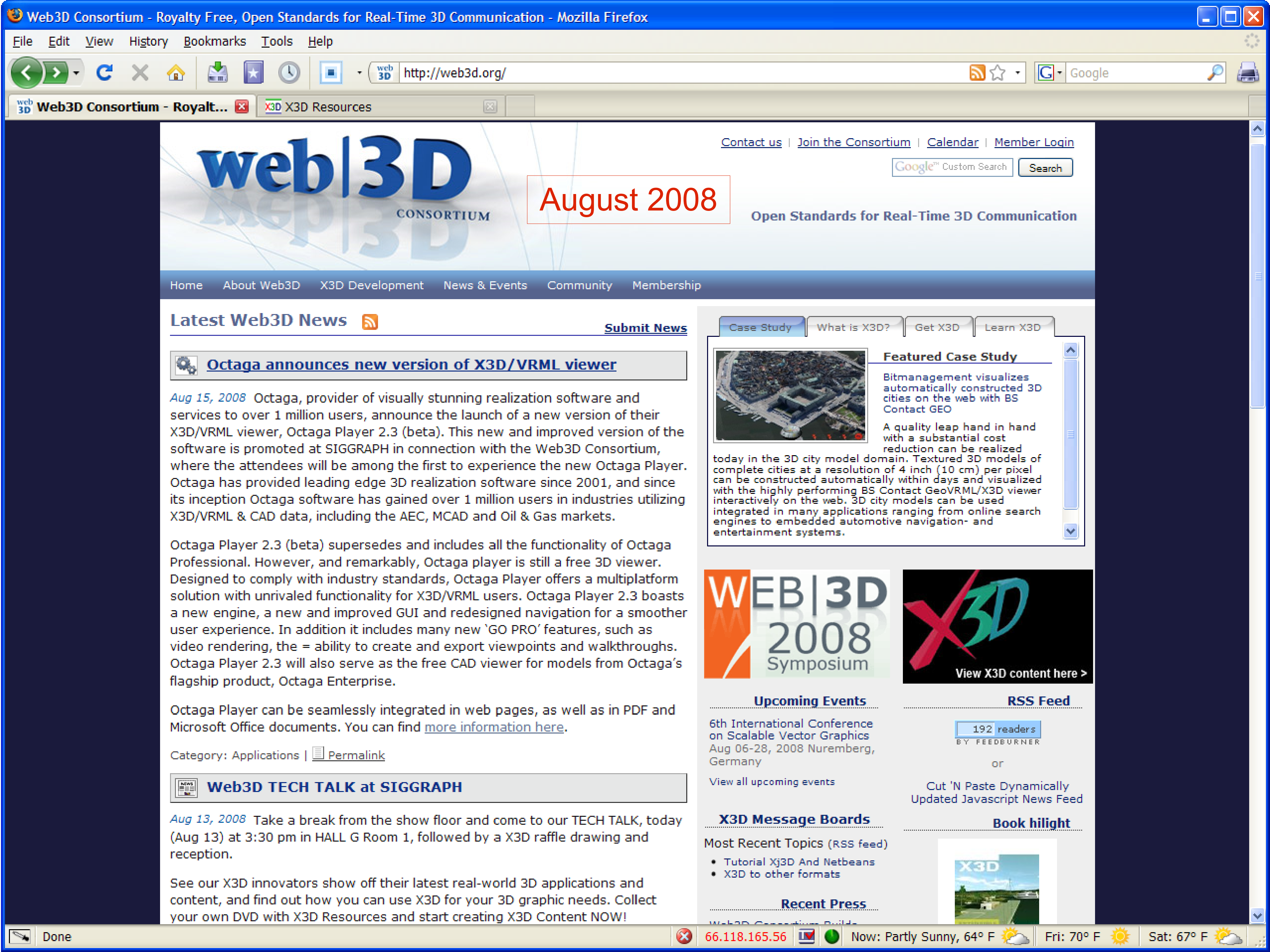
Cut 'N Paste Dynamically Updated Javascript News Feed

Book hilight

Upcoming Events

SYSTEMS
Oct 21-Oct 24, 2008
Munich, Germany

SIGGRAPH Asia 2008
Dec 10-Dec 13, 2008



Contact us | Join the Consortium | Calendar | Member Login

Google™ Custom Search Search

August 2008

Open Standards for Real-Time 3D Communication

Home About Web3D X3D Development News & Events Community Membership

Latest Web3D News

Submit News

Octaga announces new version of X3D/VRML viewer

Aug 15, 2008 Octaga, provider of visually stunning realization software and services to over 1 million users, announce the launch of a new version of their X3D/VRML viewer, Octaga Player 2.3 (beta). This new and improved version of the software is promoted at SIGGRAPH in connection with the Web3D Consortium, where the attendees will be among the first to experience the new Octaga Player. Octaga has provided leading edge 3D realization software since 2001, and since its inception Octaga software has gained over 1 million users in industries utilizing X3D/VRML & CAD data, including the AEC, MCAD and Oil & Gas markets.

Octaga Player 2.3 (beta) supersedes and includes all the functionality of Octaga Professional. However, and remarkably, Octaga player is still a free 3D viewer. Designed to comply with industry standards, Octaga Player offers a multiplatform solution with unrivaled functionality for X3D/VRML users. Octaga Player 2.3 boasts a new engine, a new and improved GUI and redesigned navigation for a smoother user experience. In addition it includes many new 'GO PRO' features, such as video rendering, the = ability to create and export viewpoints and walkthroughs. Octaga Player 2.3 will also serve as the free CAD viewer for models from Octaga's flagship product, Octaga Enterprise.

Octaga Player can be seamlessly integrated in web pages, as well as in PDF and Microsoft Office documents. You can find more information here.

Category: Applications | Permalink

Web3D TECH TALK at SIGGRAPH

Aug 13, 2008 Take a break from the show floor and come to our TECH TALK, today (Aug 13) at 3:30 pm in HALL G Room 1, followed by a X3D raffle drawing and reception.

See our X3D innovators show off their latest real-world 3D applications and content, and find out how you can use X3D for your 3D graphic needs. Collect your own DVD with X3D Resources and start creating X3D Content NOW!

Case Study | What is X3D? | Get X3D | Learn X3D

Featured Case Study

Bitmanagement visualizes automatically constructed 3D cities on the web with BS Contact GEO

A quality leap hand in hand with a substantial cost reduction can be realized today in the 3D city model domain. Textured 3D models of complete cities at a resolution of 4 inch (10 cm) per pixel can be constructed automatically within days and visualized with the highly performing BS Contact GeoVRML/X3D viewer interactively on the web. 3D city models can be used integrated in many applications ranging from online search engines to embedded automotive navigation- and entertainment systems.

Upcoming Events

6th International Conference on Scalable Vector Graphics
Aug 06-28, 2008 Nuremberg, Germany

View all upcoming events

X3D Message Boards

Most Recent Topics (RSS feed)

- Tutorial Xj3D And Netbeans
- X3D to other formats

Recent Press

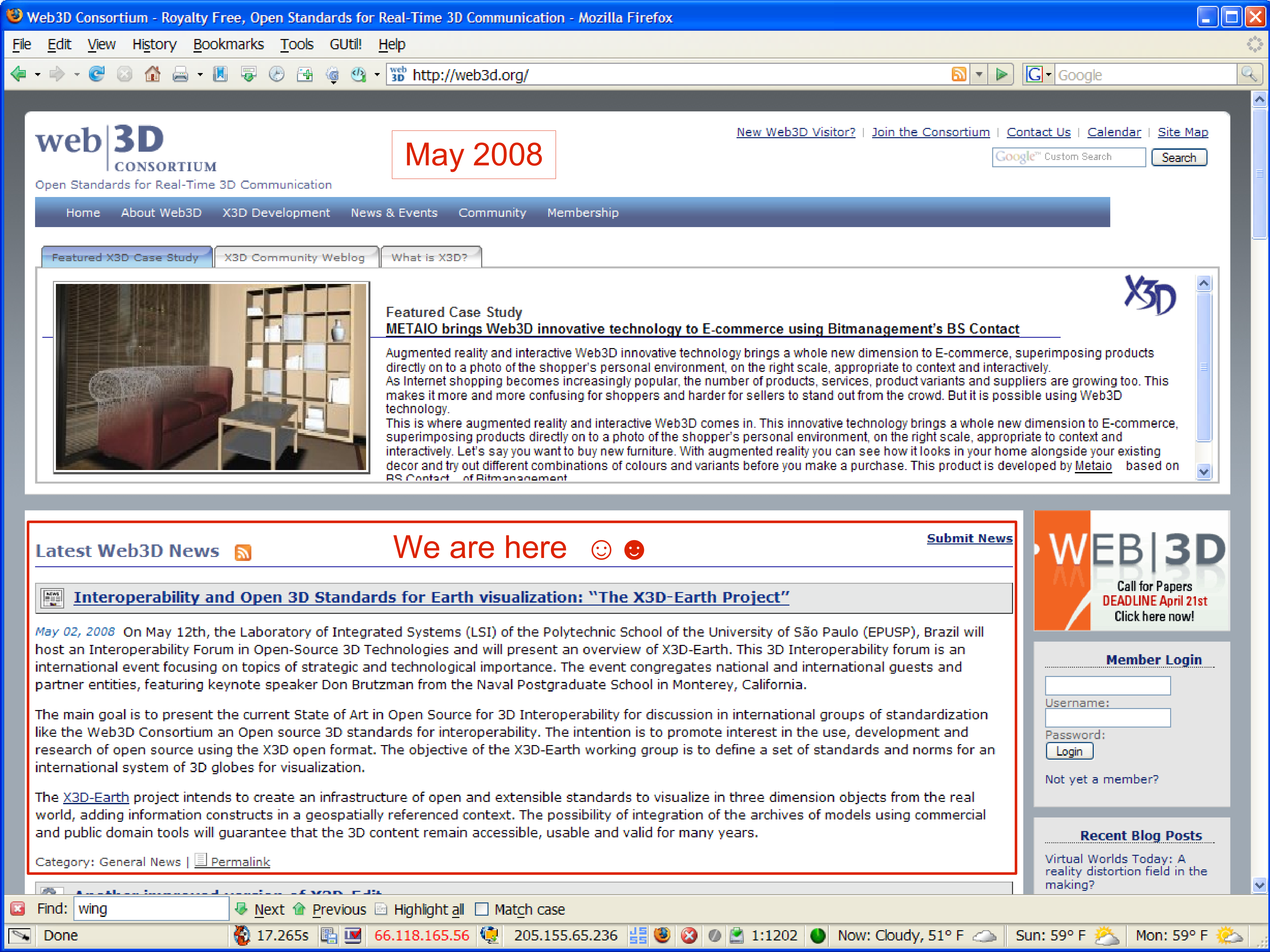
Web3D Consortium Builds

RSS Feed

or

Cut 'N Paste Dynamically Updated Javascript News Feed

Book hilight



Open Standards for Real-Time 3D Communication

May 2008

Featured Case Study

METAIO brings Web3D innovative technology to E-commerce using Bitmanagement's BS Contact

Augmented reality and interactive Web3D innovative technology brings a whole new dimension to E-commerce, superimposing products directly on to a photo of the shopper's personal environment, on the right scale, appropriate to context and interactively.

As Internet shopping becomes increasingly popular, the number of products, services, product variants and suppliers are growing too. This makes it more and more confusing for shoppers and harder for sellers to stand out from the crowd. But it is possible using Web3D technology.

This is where augmented reality and interactive Web3D comes in. This innovative technology brings a whole new dimension to E-commerce, superimposing products directly on to a photo of the shopper's personal environment, on the right scale, appropriate to context and interactively. Let's say you want to buy new furniture. With augmented reality you can see how it looks in your home alongside your existing decor and try out different combinations of colours and variants before you make a purchase. This product is developed by Metaio based on BS Contact of Bitmanagement

Latest Web3D News

We are here ☺ ☻

Submit News

Interoperability and Open 3D Standards for Earth visualization: "The X3D-Earth Project"

May 02, 2008 On May 12th, the Laboratory of Integrated Systems (LSI) of the Polytechnic School of the University of São Paulo (EPUSP), Brazil will host an Interoperability Forum in Open-Source 3D Technologies and will present an overview of X3D-Earth. This 3D Interoperability forum is an international event focusing on topics of strategic and technological importance. The event congregates national and international guests and partner entities, featuring keynote speaker Don Brutzman from the Naval Postgraduate School in Monterey, California.

The main goal is to present the current State of Art in Open Source for 3D Interoperability for discussion in international groups of standardization like the Web3D Consortium an Open source 3D standards for interoperability. The intention is to promote interest in the use, development and research of open source using the X3D open format. The objective of the X3D-Earth working group is to define a set of standards and norms for an international system of 3D globes for visualization.

The X3D-Earth project intends to create an infrastructure of open and extensible standards to visualize in three dimension objects from the real world, adding information constructs in a geospatially referenced context. The possibility of integration of the archives of models using commercial and public domain tools will guarantee that the 3D content remain accessible, usable and valid for many years.

Category: General News | Permalink

Another improved version of X3D-Edit

Member Login

Username:

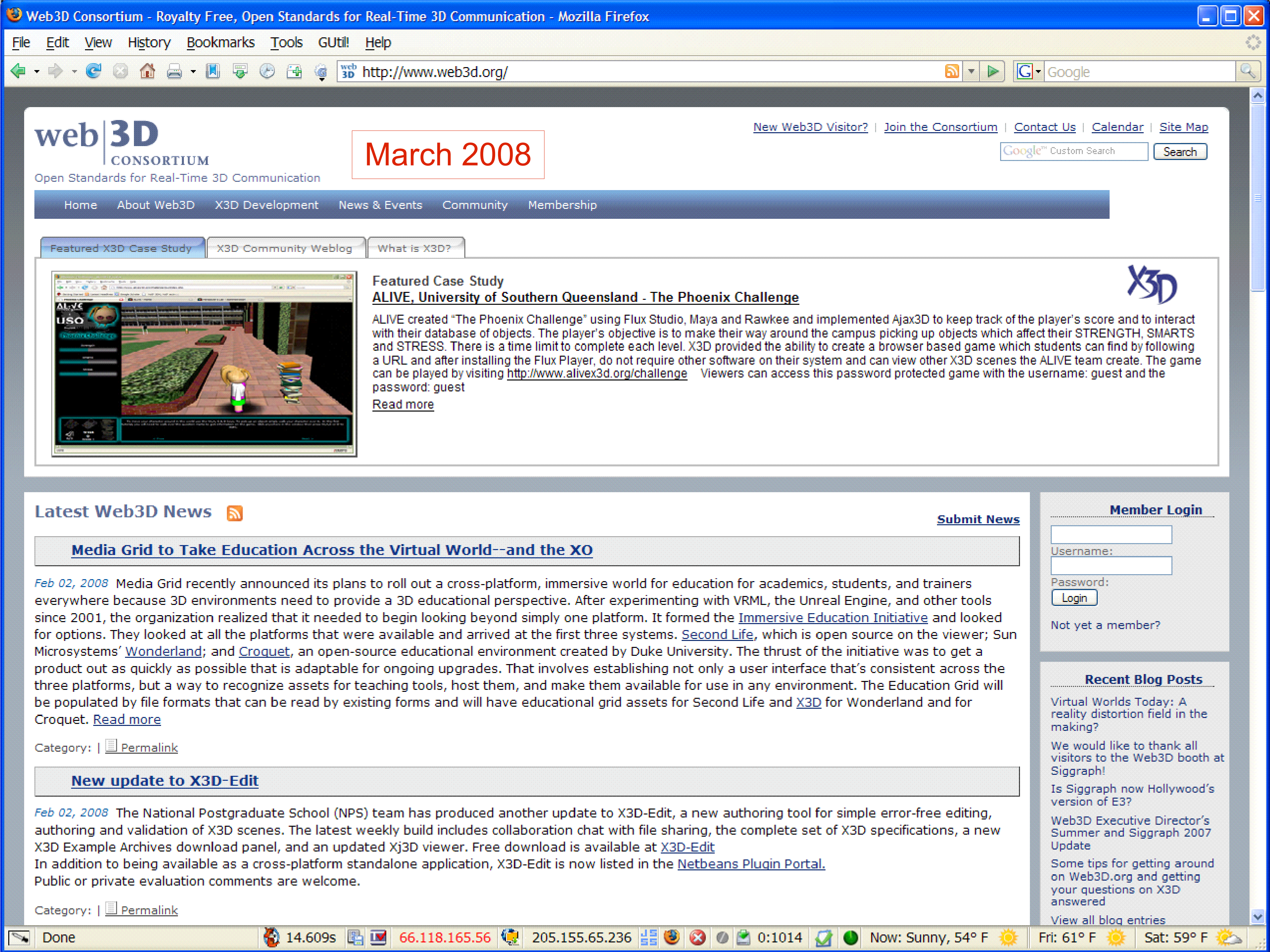
Password:

Login

Not yet a member?

Recent Blog Posts

Virtual Worlds Today: A reality distortion field in the making?



web 3D CONSORTIUM

Open Standards for Real-Time 3D Communication

March 2008

New Web3D Visitor? | Join the Consortium | Contact Us | Calendar | Site Map

Google™ Custom Search Search

Home About Web3D X3D Development News & Events Community Membership

Featured X3D Case Study | X3D Community Weblog | What is X3D?

X3D

Featured Case Study

ALIVE, University of Southern Queensland - The Phoenix Challenge

ALIVE created "The Phoenix Challenge" using Flux Studio, Maya and Rawkee and implemented Ajax3D to keep track of the player's score and to interact with their database of objects. The player's objective is to make their way around the campus picking up objects which affect their STRENGTH, SMARTS and STRESS. There is a time limit to complete each level. X3D provided the ability to create a browser based game which students can find by following a URL and after installing the Flux Player, do not require other software on their system and can view other X3D scenes the ALIVE team create. The game can be played by visiting http://www.alivex3d.org/challenge Viewers can access this password protected game with the username: guest and the password: guest

Read more

Latest Web3D News

Submit News

Media Grid to Take Education Across the Virtual World--and the XO

Feb 02, 2008 Media Grid recently announced its plans to roll out a cross-platform, immersive world for education for academics, students, and trainers everywhere because 3D environments need to provide a 3D educational perspective. After experimenting with VRML, the Unreal Engine, and other tools since 2001, the organization realized that it needed to begin looking beyond simply one platform. It formed the Immersive Education Initiative and looked for options. They looked at all the platforms that were available and arrived at the first three systems. Second Life, which is open source on the viewer; Sun Microsystems' Wonderland; and Croquet, an open-source educational environment created by Duke University. The thrust of the initiative was to get a product out as quickly as possible that is adaptable for ongoing upgrades. That involves establishing not only a user interface that's consistent across the three platforms, but a way to recognize assets for teaching tools, host them, and make them available for use in any environment. The Education Grid will be populated by file formats that can be read by existing forms and will have educational grid assets for Second Life and X3D for Wonderland and for Croquet. Read more

Category: | Permalink

New update to X3D-Edit

Feb 02, 2008 The National Postgraduate School (NPS) team has produced another update to X3D-Edit, a new authoring tool for simple error-free editing, authoring and validation of X3D scenes. The latest weekly build includes collaboration chat with file sharing, the complete set of X3D specifications, a new X3D Example Archives download panel, and an updated Xj3D viewer. Free download is available at X3D-Edit
In addition to being available as a cross-platform standalone application, X3D-Edit is now listed in the Netbeans Plugin Portal.
Public or private evaluation comments are welcome.

Category: | Permalink

Member Login

Username:

Password:

Login

Not yet a member?

Recent Blog Posts

Virtual Worlds Today: A reality distortion field in the making?

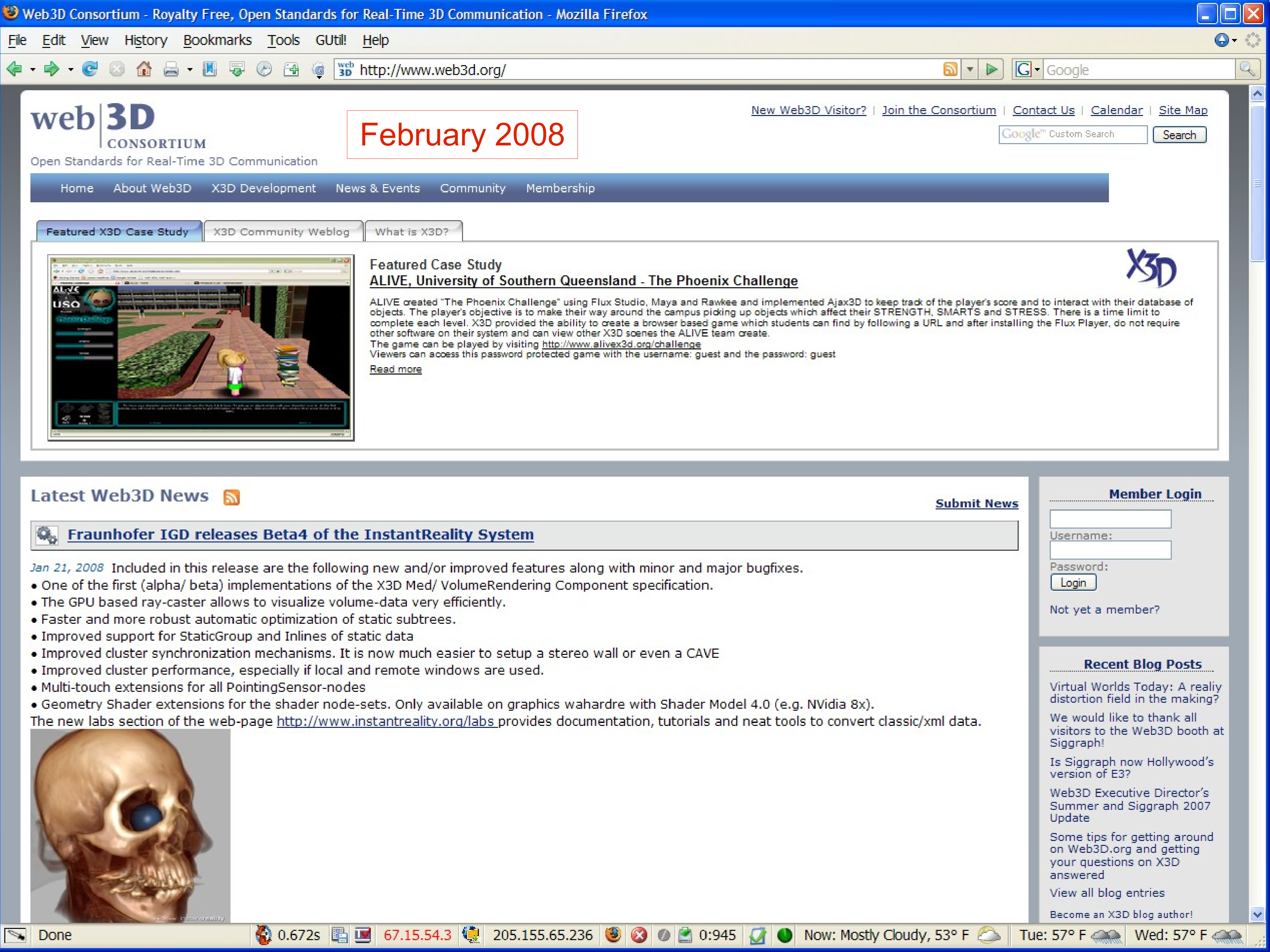
We would like to thank all visitors to the Web3D booth at Siggraph!

Is Siggraph now Hollywood's version of E3?

Web3D Executive Director's Summer and Siggraph 2007 Update

Some tips for getting around on Web3D.org and getting your questions on X3D answered

View all blog entries



web|3D CONSORTIUM

Open Standards for Real-Time 3D Communication

February 2008

New Web3D Visitor? | Join the Consortium | Contact Us | Calendar | Site Map

Google™ Custom Search [Search]

Home | About Web3D | X3D Development | News & Events | Community | Membership

Featured X3D Case Study | X3D Community Weblog | What is X3D?

Featured Case Study

ALIVE, University of Southern Queensland - The Phoenix Challenge

ALIVE created "The Phoenix Challenge" using Flux Studio, Maya and Rawkee and implemented Ajax3D to keep track of the player's score and to interact with their database of objects. The player's objective is to make their way around the campus picking up objects which affect their STRENGTH, SMARTS and STRESS. There is a time limit to complete each level. X3D provided the ability to create a browser based game which students can find by following a URL and after installing the Flux Player, do not require other software on their system and can view other X3D scenes the ALIVE team create.
The game can be played by visiting http://www.alivex3d.org/challenge
Viewers can access this password protected game with the username: guest and the password: guest

Read more

Latest Web3D News

Submit News

Fraunhofer IGD releases Beta4 of the InstantReality System

Jan 21, 2008 Included in this release are the following new and/or improved features along with minor and major bugfixes.

- One of the first (alpha/ beta) implementations of the X3D Med/ VolumeRendering Component specification.
- The GPU based ray-caster allows to visualize volume-data very efficiently.
- Faster and more robust automatic optimization of static subtrees.
- Improved support for StaticGroup and Inlines of static data
- Improved cluster synchronization mechanisms. It is now much easier to setup a stereo wall or even a CAVE
- Improved cluster performance, especially if local and remote windows are used.
- Multi-touch extensions for all PointingSensor-nodes
- Geometry Shader extensions for the shader node-sets. Only available on graphics wahardre with Shader Model 4.0 (e.g. NVidia 8x).

The new labs section of the web-page http://www.instantreality.org/labs provides documentation, tutorials and neat tools to convert classic/xml data.

Member Login

Username:

Password:

[Login]

Not yet a member?

Recent Blog Posts

Virtual Worlds Today: A realiy distortion field in the making?

We would like to thank all visitors to the Web3D booth at Siggraph!

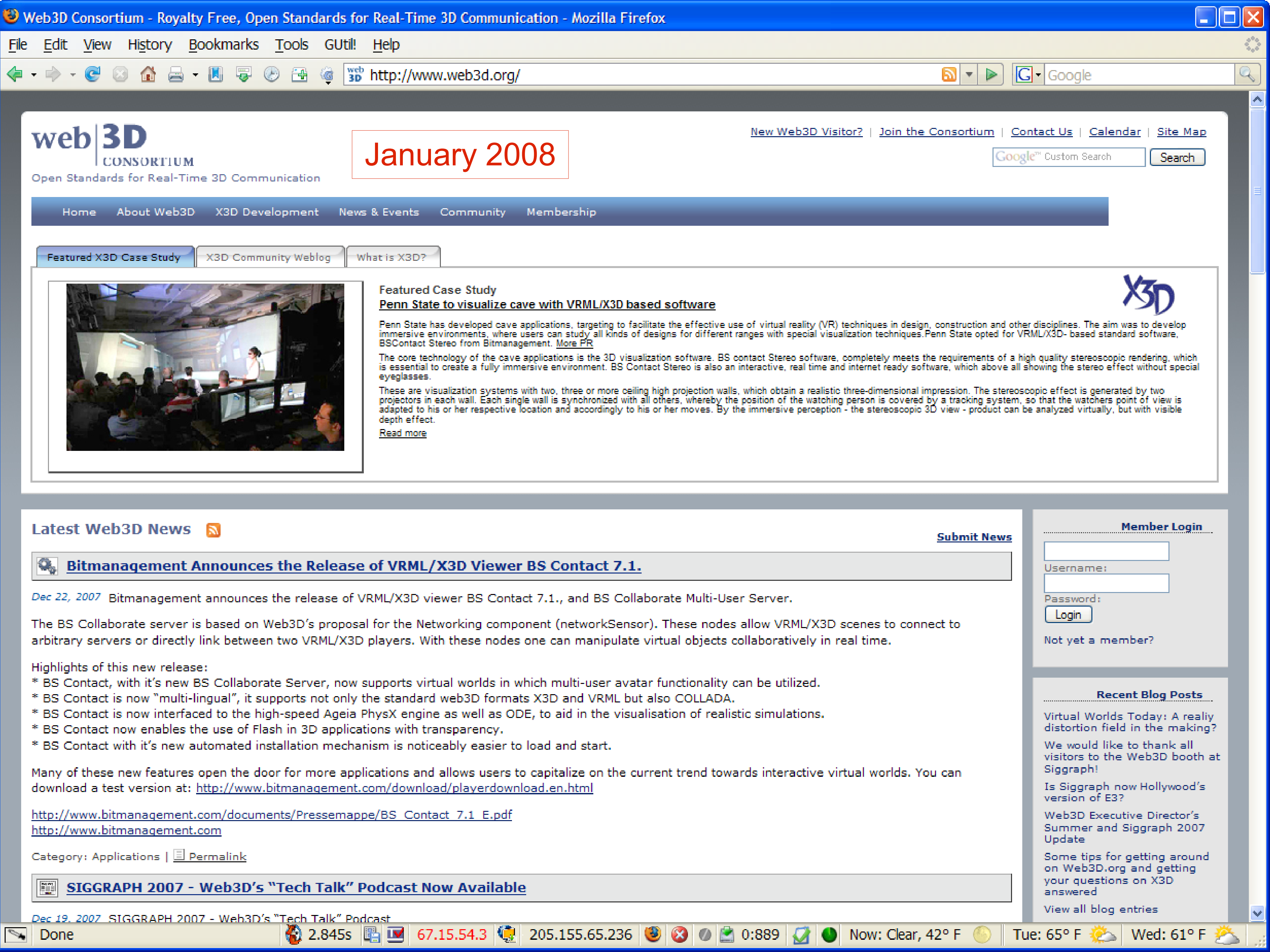
Is Siggraph now Hollywood's version of E3?

Web3D Executive Director's Summer and Siggraph 2007 Update

Some tips for getting around on Web3D.org and getting your questions on X3D answered

View all blog entries

Become an X3D blog author!



Web3D Consortium - Royalty Free, Open Standards for Real-Time 3D Communication - Mozilla Firefox
File Edit View History Bookmarks Tools GUtil! Help
http://www.web3d.org/
Google

web 3D CONSORTIUM
Open Standards for Real-Time 3D Communication

January 2008

New Web3D Visitor? | Join the Consortium | Contact Us | Calendar | Site Map
Google™ Custom Search
Search
Home
About Web3D
X3D Development
News & Events
Community
Membership
Featured X3D Case Study
X3D Community Weblog
What is X3D?

Featured Case Study

Penn State to visualize cave with VRML/X3D based software

Penn State has developed cave applications, targeting to facilitate the effective use of virtual reality (VR) techniques in design, construction and other disciplines. The aim was to develop immersive environments, where users can study all kinds of designs for different ranges with special visualization techniques.Penn State opted for VRML/X3D- based standard software, BSContact Stereo from Bitmanagement. More PR

The core technology of the cave applications is the 3D visualization software. BS contact Stereo software, completely meets the requirements of a high quality stereoscopic rendering, which is essential to create a fully immersive environment. BS Contact Stereo is also an interactive, real time and internet ready software, which above all showing the stereo effect without special eyeglasses.

These are visualization systems with two, three or more ceiling high projection walls, which obtain a realistic three-dimensional impression. The stereoscopic effect is generated by two projectors in each wall. Each single wall is synchronized with all others, whereby the position of the watching person is covered by a tracking system, so that the watchers point of view is adapted to his or her respective location and accordingly to his or her moves. By the immersive perception - the stereoscopic 3D view - product can be analyzed virtually, but with visible depth effect.

Read more

Latest Web3D News

Submit News

Bitmanagement Announces the Release of VRML/X3D Viewer BS Contact 7.1.

Dec 22, 2007 Bitmanagement announces the release of VRML/X3D viewer BS Contact 7.1., and BS Collaborate Multi-User Server.

The BS Collaborate server is based on Web3D's proposal for the Networking component (networkSensor). These nodes allow VRML/X3D scenes to connect to arbitrary servers or directly link between two VRML/X3D players. With these nodes one can manipulate virtual objects collaboratively in real time.

Highlights of this new release:
* BS Contact, with it's new BS Collaborate Server, now supports virtual worlds in which multi-user avatar functionality can be utilized.
* BS Contact is now "multi-lingual", it supports not only the standard web3D formats X3D and VRML but also COLLADA.
* BS Contact is now interfaced to the high-speed Ageia PhysX engine as well as ODE, to aid in the visualisation of realistic simulations.
* BS Contact now enables the use of Flash in 3D applications with transparency.
* BS Contact with it's new automated installation mechanism is noticeably easier to load and start.

Many of these new features open the door for more applications and allows users to capitalize on the current trend towards interactive virtual worlds. You can download a test version at: http://www.bitmanagement.com/download/playerdownload.en.html

http://www.bitmanagement.com/documents/Pressemappe/BS_Contact_7.1_E.pdf
http://www.bitmanagement.com

Category: Applications | Permalink

SIGGRAPH 2007 - Web3D's "Tech Talk" Podcast Now Available

Dec 19, 2007 SIGGRAPH 2007 - Web3D's "Tech Talk" Podcast

Member Login

Username:

Password:

Login

Not yet a member?

Recent Blog Posts

Virtual Worlds Today: A realiy distortion field in the making?

We would like to thank all visitors to the Web3D booth at Siggraph!

Is Siggraph now Hollywood's version of E3?

Web3D Executive Director's Summer and Siggraph 2007 Update

Some tips for getting around on Web3D.org and getting your questions on X3D answered

View all blog entries

X3D Specifications

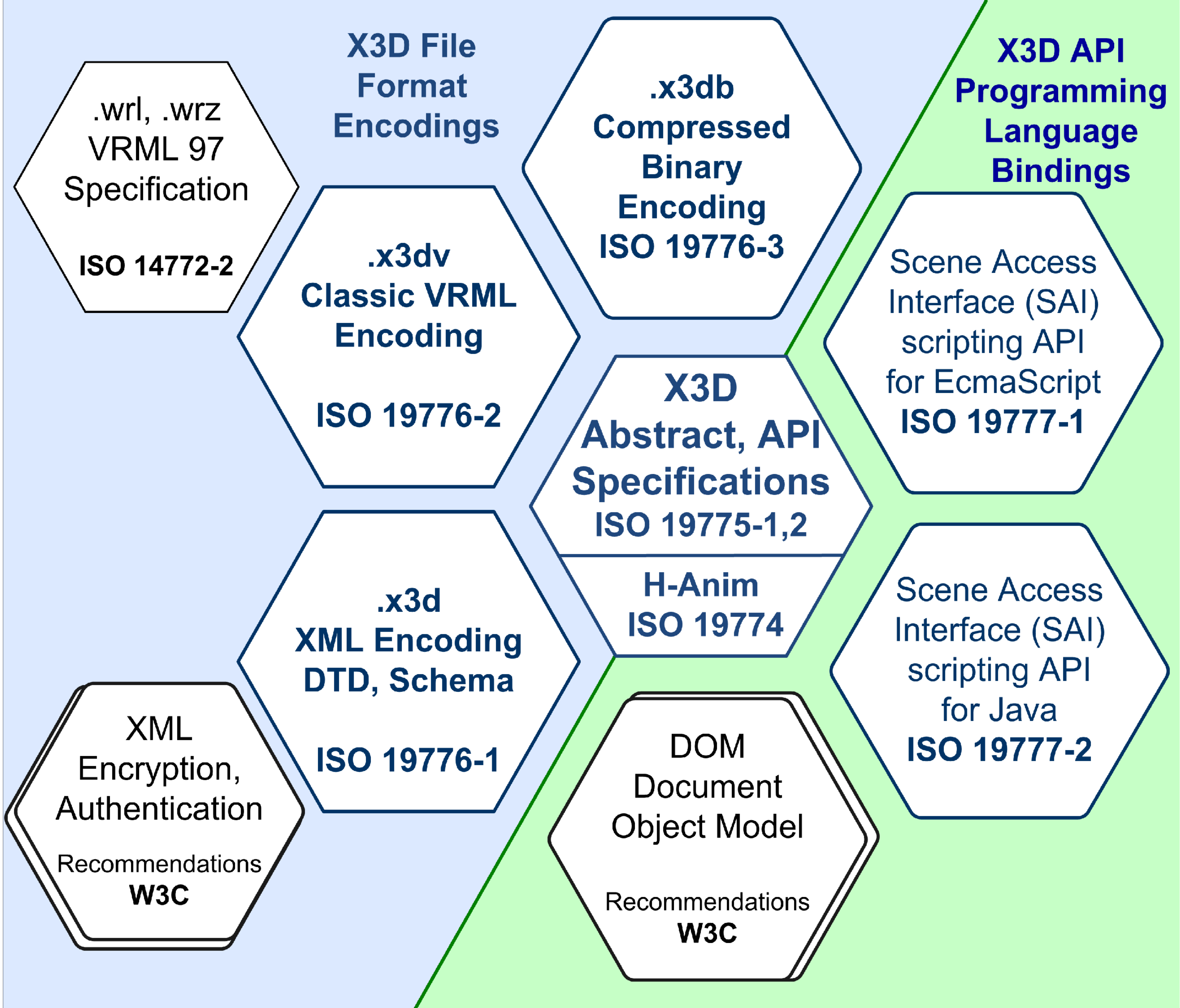
X3D graphics is defined by a set of specifications

These "specs" are developed by working-group volunteers as part of the Web3D Consortium

- Nonprofit organization with business, nonprofit, academic and professional members
- http://www.web3D.org
- Efforts include editing, implementing and evaluating

Specification results reviewed and approved by International Organization of Standards (ISO)

- http://www.iso.ch



X3D File Format Encodings

.wrl, .wrz
VRML 97
Specification
ISO 14772-2

.x3dv
Classic VRML Encoding
ISO 19776-2

.x3db
Compressed Binary Encoding
ISO 19776-3

.x3d
XML Encoding
DTD, Schema
ISO 19776-1

XML Encryption, Authentication
Recommendations
W3C

X3D
Abstract, API Specifications
ISO 19775-1,2

H-Anim
ISO 19774

X3D API Programming Language Bindings

Scene Access Interface (SAI) scripting API for EcmaScript
ISO 19777-1

Scene Access Interface (SAI) scripting API for Java
ISO 19777-2

DOM
Document Object Model
Recommendations
W3C

Reading the X3D specification

The X3D Specification is highly detailed, primarily written for 3D graphics experts.

Requirements must be described as strictly and precisely as possible so that X3D browsers can be implemented consistently. This precision means that X3D content is more likely to render and animate correctly.

Nevertheless the X3D specification is a great learning resource for additional graphics details. It is also the authoritative reference for questions.

Specification availability

The X3D specifications are online at

- http://www.web3d.org/x3d/specifications
- also embedded in the X3D-Edit help system

The X3D specifications are published by the Web3D Consortium and International Organization of Standards (ISO)

- Web3D versions are published in HTML for free
- ISO publishes .pdf versions and requires purchase

Feedback on X3D specifications is always welcome

- http://www.web3d.org/x3d/specifications/spec_feedback

X3D plugins, Web browsers, applications

X3D browsers parse (read) X3D scene models and render (draw) them

- Also provide simulation capabilities for animation and user interaction
- http://www.web3d.org/x3d/content/examples/X3dResources.html#Applications

Often implemented as plugins to web browsers:

- Internet Explorer http://www.microsoft.com
- Mozilla Firefox http://www.mozilla.com
- Opera http://www.opera.com

Can also operate as a standalone application

- Xj3D http://www.xj3d.org
- Instant Reality http://www.instantreality.org

X3D browser plugin list (partial)

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

Xj3D Open Source for X3D/VRML97. Version 2.0 release using Java OpenGL (JOGL) rendering. Includes a Java WebStart version (Java standalone, Windows MacOSX Linux Solaris)

CRC's FreeWRL X3D/VRML browser (open-source C). Also available via Apple website (MacOSX Linux)

BitManagement's BS Contact X3D/VRML97 plugin for Internet Explorer (Windows MacOSX Linux)

Octaga X3D/VRML browser with high performance and community support forum (Windows MacOSX Linux)

InstantReality is a high-performance Mixed Reality (MR) system (Windows MacOSX Linux)

Vivaty's Flux Player X3D/VRML97 plugin for Internet Explorer (Windows)

SwirlX3D Free Viewer by Pine Coast Software (Windows)

Heilan X3D Browser open-source C++ browser for audio research (Windows Linux)

NuGraf 3D Rendering, Translation, Viewing & Data Optimization System by Okino (Windows and authoring-tool plugins)

Example software architecture for X3D browser

3D graphics algorithms and implementations are intensely technical and performance-sensitive

X3D browsers are thus allowed to implement in any manner which they choose

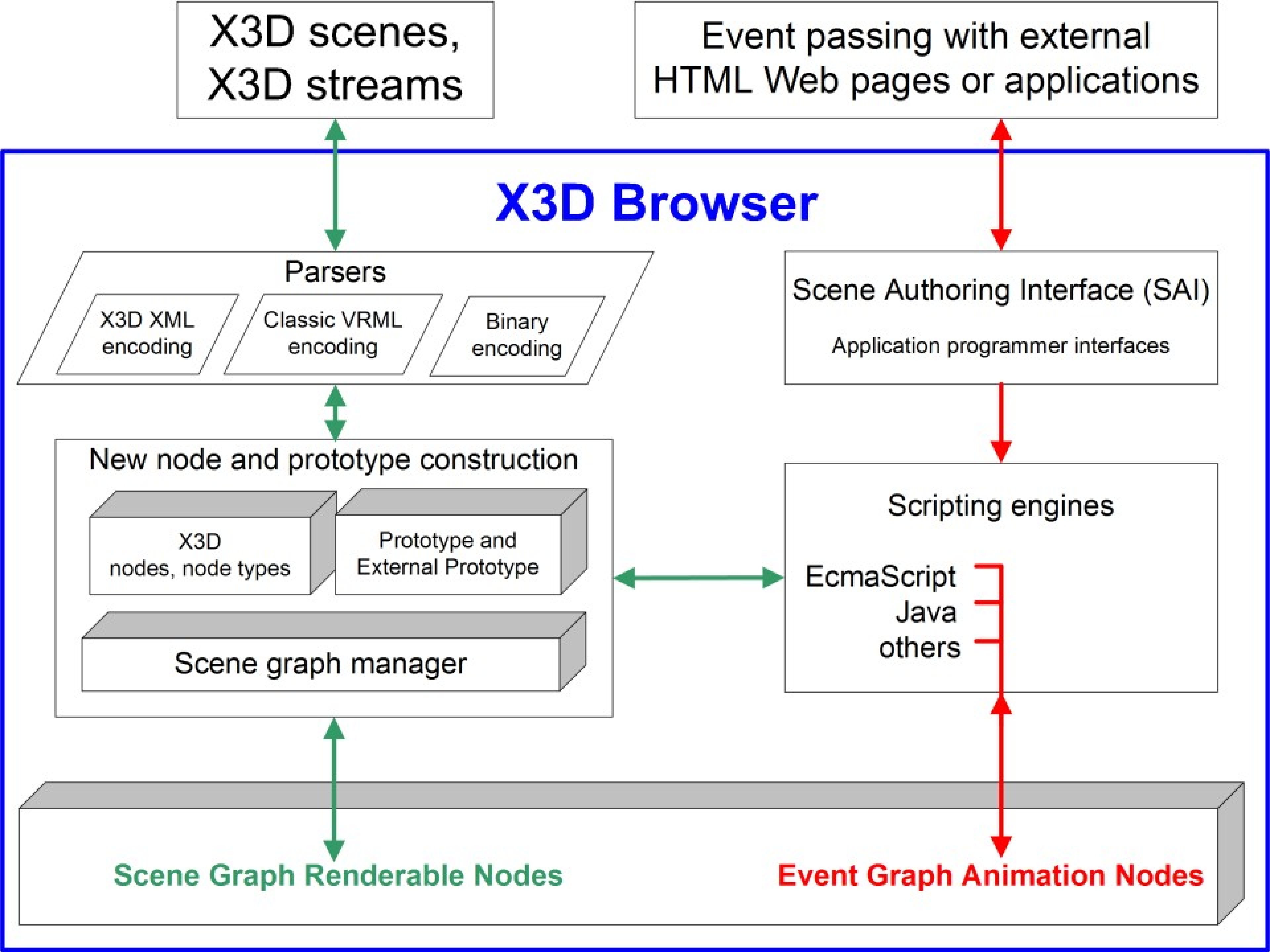
- As long as the author's X3D scene works properly

This is a healthy division of responsibilities

- Each gets to excel at what they are good at

Commonalities and shared lessons learned continue to build up nicely

- Next diagram shows example architecture


X3D scenes,
X3D streams
Event passing with external
HTML Web pages or applications
X3D Browser
Parsers
X3D XML encoding
Classic VRML encoding
Binary encoding
Scene Authoring Interface (SAI)
Application programmer interfaces
New node and prototype construction
X3D nodes, node types
Prototype and External Prototype
Scene graph manager
Scripting engines
EcmaScript
Java
others
Scene Graph Renderable Nodes
Event Graph Animation Nodes

Scene graph concepts

Scene graphs are a model-centric approach to 3D that hierarchically defines geometry shape, appearance, position and orientation, etc. etc.

- Directed acyclic graph (DAG), meaning a tree with a root node and no loops
- Declarative listing of parameters of interest
- Similar to defining a Computer Aided Design (CAD) model

Unlike most imperative programming approaches

- draw this triangle, that triangle, recompute, etc.

Scene graph terminology

Scene graph data file

- contains model description, may refer to data files

Scene graph viewer

- Reads and renders scene-graph models
- Implemented as application or web browser plugin

Scene graph editor

- Special text editor for scene graph development

Executable application

- Specific 3D model capable of running on a specific operating system

Scene graph rendering

The browser traverses the scene graph, updating any values within nodes and building an image

- New image then replaces previous screen image, process known as *double buffering*
- Rapid repetitions are very important
- Frame rate faster than 7-10 Hz (cycles per second) provides appearance of smooth motion

Rendering defined as this drawing process

Off-line rendering is performing such operations to image or movie files, rather than display

Performance optimizations

Scene graphs have performance optimizations sometimes not available in other Application Programming Interface (API) approaches

- Scene graph structure designed to take advantage of graphics hardware acceleration
- Can refer to (and reuse) subgraphs (X3D DEF, USE)
- “dirty bit” indicates whether a scene subgraph has been modified, avoiding needless recomputations
- Browser can rearrange or simplify geometry
- Scoping of lights to reduce computational impact
- Widely repeated interchange patterns

Scene-graph advantages relative to OpenGL, DirectX render layers

Scene graphs often a close match to simulation models, easier for authors to make and modify

OpenGL and DirectX APIs are thin software layers that expose underlying 3D graphics-acceleration hardware for real-time rendering

Each is a state machine, optimized for drawing triangles textures etc., not designed to have memory for modeling high-level simulation objects, remembering user actions, etc.

Scene graph compared to ray tracing

Ray tracing emulates physical properties of light interaction with material surfaces

- Ray vectors are propagated, computed, added
- Computational time can be intensive, usually best for high-fidelity rendering (rather than real-time)

Variety of different approaches, programs

- Persistence of Vision Raytracer (www.povray.org)
- Movies, e.g. Renderman (renderman.pixar.com)

Scene graph designed for real-time rendering

- But X3D Specification has no rendering prohibitions
- Okino Polytrans supports both (www.okino.com)

Other scene graph architectures

OpenInventor (OI), predecessor of VRML

- http://oss.sgi.com/projects/inventor

Virtual Reality Modeling Language (VRML), direct predecessor of X3D

- http://www.web3d.org/x3d/specifications

Java3D quite similar to X3D scene graph

- https://java3d.dev.java.net

OpenSceneGraph (OSG)

- http://www.openscenegraph.org

OpenSG

- http://www.opensg.org

X3D file structure

X3D scene files have a common file structure

- File header (XML, ClassicVRML, Compressed Binary)
- X3D header statement
- Profile statement
- Component statements (optional)
- Meta statements (optional)
- X3D root node
- X3D scene graph child nodes

XML file encoding

The Extensible Markup Language (XML) is a plain-text format used by many Web languages

- Including Hypertext Markup Language (HTML)

XML is used to define other data-oriented languages

- Thus XML is not a language by itself, rather it is a language about languages, a *metalanguage*
- Common XML basis enables better interoperability, opens a “path of least resistance” for data flow

XML has many benefits and is well-suited for X3D

XML in 10 Points

http://www.w3.org/XML/1999/XML-in-10-points

XML is for structuring data

XML looks a bit like HTML

XML is text, but isn't meant to be read

XML is verbose by design

XML is a family of technologies

XML in 10 Points is a key reference for understanding the common underlying design principles underlying the great diversity of XML.

Only 4 pages long – essential reading.

XML is new but not that new

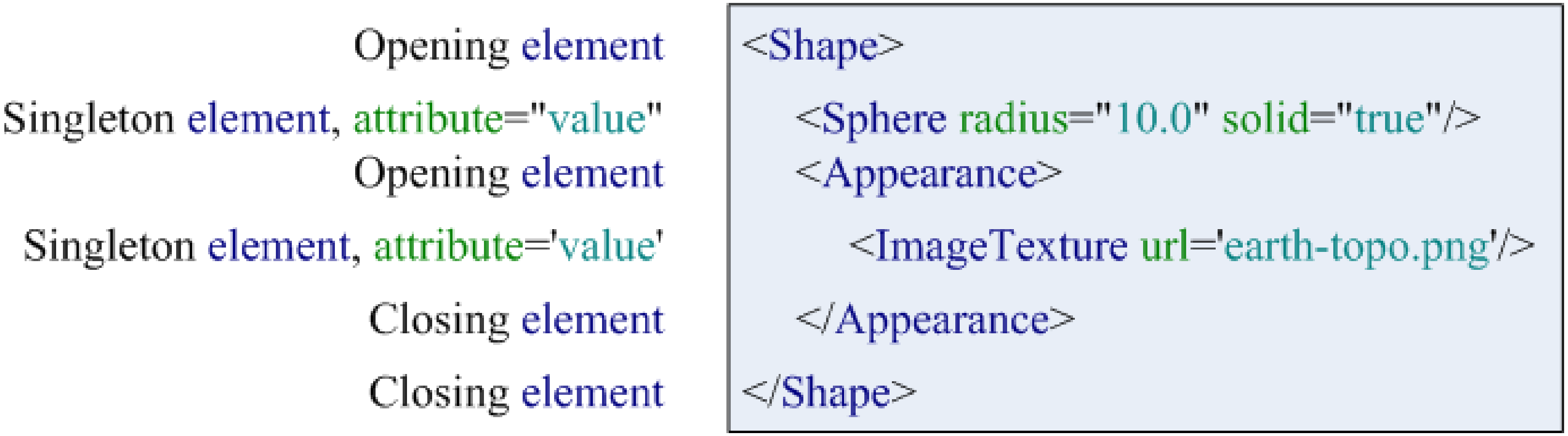
XML leads HTML to XHTML

XML is modular

XML is basis for RDF and the Semantic Web

XML is license-free, platform-independent and well-supported

XML and X3D correspondence



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

- Elements correspond to X3D nodes
- Attributes correspond to X3D simple-type fields
- Parent-child relationships define containerField
- Validatable XML using X3D DTD, schema

Need for subdivisions and subsets

3D graphics is a big and complicated subject

- Beginning authors just want simple scenes
- Experienced authors want to use everything

Similar needs for browser software builders

- Small rapid download for simple web graphics
- Full-capability software for every possible technique

Challenge: how to consistently support both?

- Object-oriented decomposition for consistency
- Key design criteria for bottom-up X3D extensibility

Profiles and components

Profiles are predefined collections of components

- Can augmented each by adding other components

Components are predefined collections of nodes

- Further defined by *level* of complexity
- Components match chapters in X3D specification

Authors define the expected complexity of scene by defining profile level in the X3D header

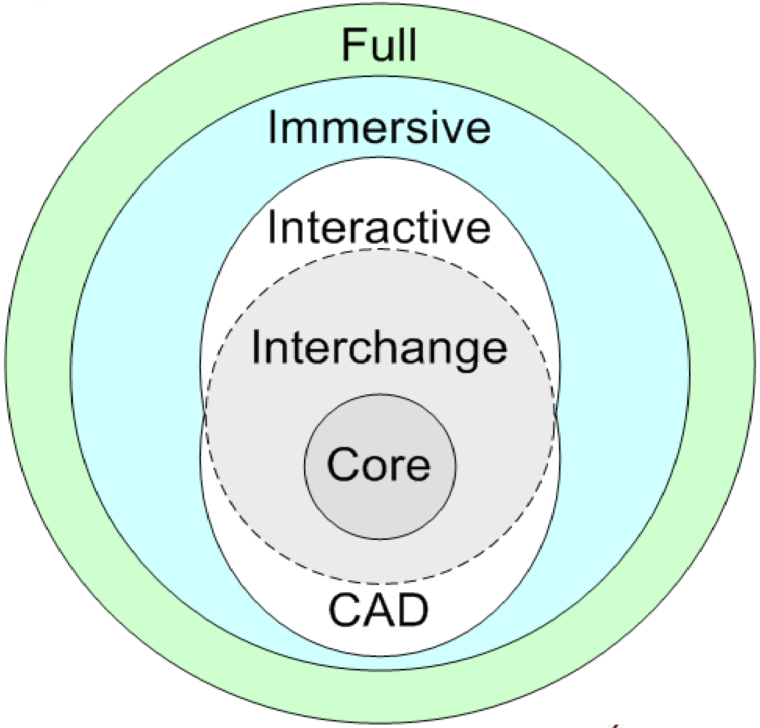
- Can also add optional components, if desired
- This tells the X3D browser what level of support is needed for run-time operation

Profiles cover common use cases

Profiles are a collection of components matching common levels of complexity

Profiles are X3D subsets

- Collection of X3D nodes for for author's palette
- *Interchange* suitable for simple geometry conversion
- *Interactive* adds simple user interactivity (clicking etc.)
- *Immersive* matches VRML97
- *Full* profile includes all nodes



meta statements

meta statements provide information about the X3D scene

- Document metadata, not scene metadata

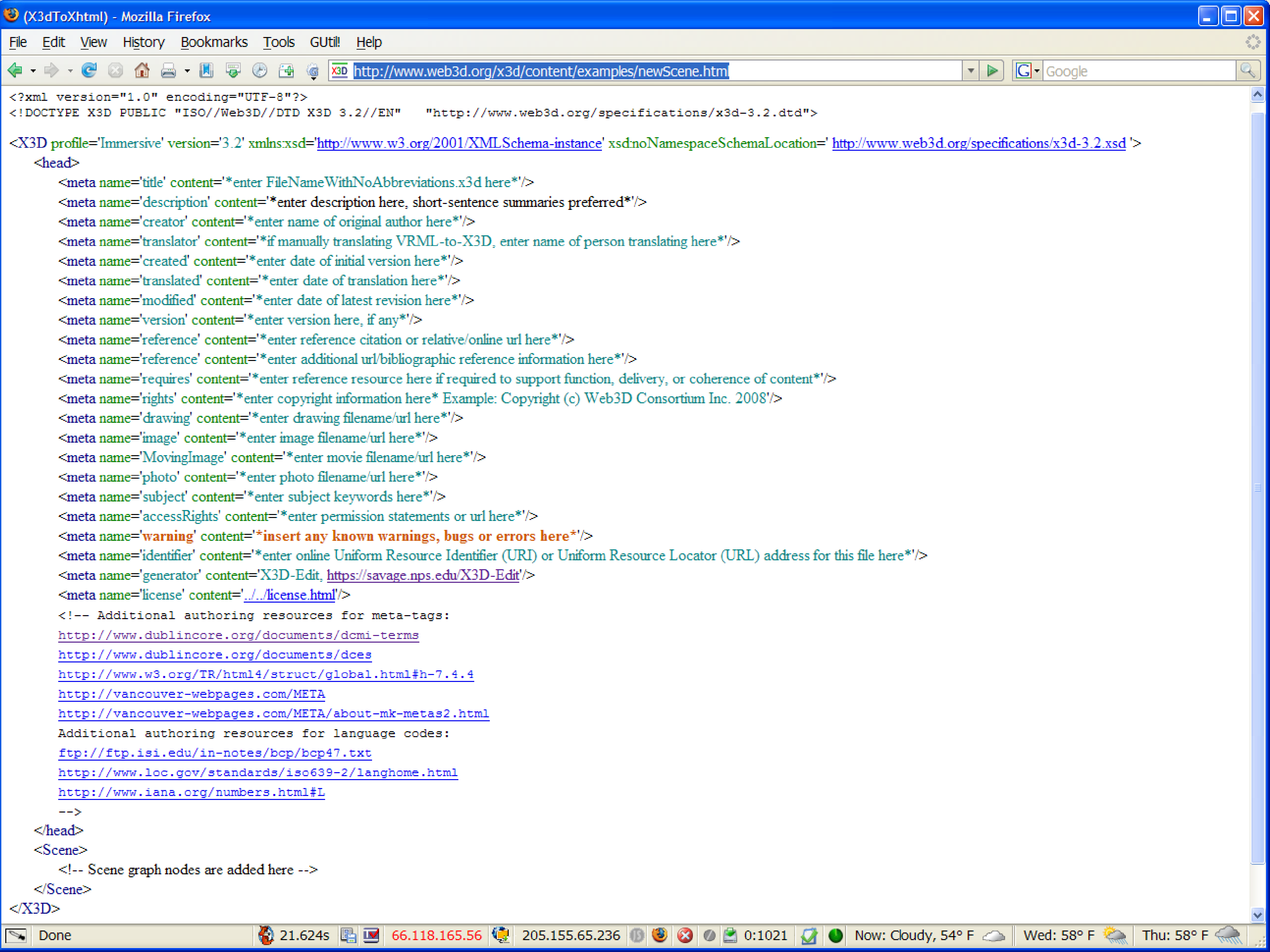
Information provided as name-value pairs

- Example:

```
<meta name='created' value='1 January 2008'/>
```

This approach is thus very general

- Wide variety of metadata can be represented
- Matches same approach used by HTML for regular hypertext web pages



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN"   "http://www.web3d.org/specifications/x3d-3.2.dtd">

<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation=' http://www.web3d.org/specifications/x3d-3.2.xsd '>
    <head>
        <meta name='title' content='*enter FileNameWithNoAbbreviations.x3d here*'/>
        <meta name='description' content='*enter description here, short-sentence summaries preferred*'/>
        <meta name='creator' content='*enter name of original author here*'/>
        <meta name='translator' content='*if manually translating VRML-to-X3D, enter name of person translating here*'/>
        <meta name='created' content='*enter date of initial version here*'/>
        <meta name='translated' content='*enter date of translation here*'/>
        <meta name='modified' content='*enter date of latest revision here*'/>
        <meta name='version' content='*enter version here, if any*'/>
        <meta name='reference' content='*enter reference citation or relative/online url here*'/>
        <meta name='reference' content='*enter additional url/bibliographic reference information here*'/>
        <meta name='requires' content='*enter reference resource here if required to support function, delivery, or coherence of content*'/>
        <meta name='rights' content='*enter copyright information here* Example: Copyright (c) Web3D Consortium Inc. 2008'/>
        <meta name='drawing' content='*enter drawing filename/url here*'/>
        <meta name='image' content='*enter image filename/url here*'/>
        <meta name='MovingImage' content='*enter movie filename/url here*'/>
        <meta name='photo' content='*enter photo filename/url here*'/>
        <meta name='subject' content='*enter subject keywords here*'/>
        <meta name='accessRights' content='*enter permission statements or url here*'/>
        <meta name='warning' content='*insert any known warnings, bugs or errors here*'/>
        <meta name='identifier' content='*enter online Uniform Resource Identifier (URI) or Uniform Resource Locator (URL) address for this file here*'/>
        <meta name='generator' content='X3D-Edit, https://savage.nps.edu/X3D-Edit'/>
        <meta name='license' content='../../license.html'/>
        <!-- Additional authoring resources for meta-tags:
        http://www.dublincore.org/documents/dcmi-terms
        http://www.dublincore.org/documents/dces
        http://www.w3.org/TR/html4/struct/global.html#h-7.4.4
        http://vancouver-webpages.com/META
        http://vancouver-webpages.com/META/about-mk-metas2.html
        Additional authoring resources for language codes:
        ftp://ftp.isi.edu/in-notes/bcp/bcp47.txt
        http://www.loc.gov/standards/iso639-2/langhome.html
        http://www.iana.org/numbers.html#L
        -->
    </head>
    <Scene>
        <!-- Scene graph nodes are added here -->
    </Scene>
</X3D>
```

DEF and USE

DEF names provide a label for any node

- Including child nodes making up that subgraph
- Equivalent to ID type in XML: must be unique
- Provides target for routing events
- Multiple DEFs: legal in X3D, illegal in XML, harmful

USE labels reference a DEF node

- Spelling is case sensitive, must be identical

DEF label must precede USE reference in scene

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

DEF naming

Names are important!

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

Naming convention: CamelCaseNaming

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

Units of measurement

Linear measurements in meters

- 1 m = 39.3”

Angular measurements in radians

- 2 pi = 360 degrees

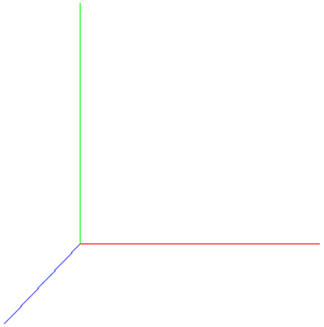
Time measured in seconds

- Starting 1 January 1970

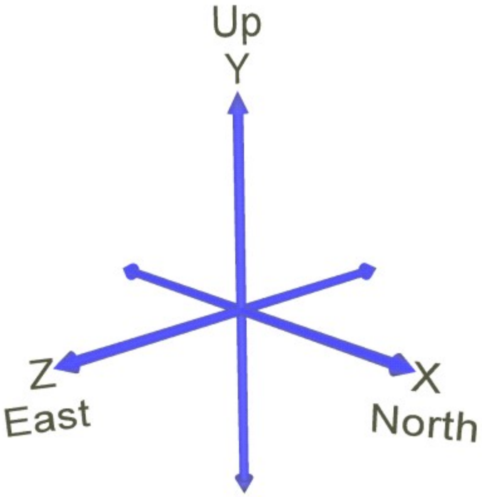
Colors

- RGB red-green-blue floating points ranging 0..1 (vice HTML which has integers 0..255)

Coordinate systems



Right hand rule for X Y Z order

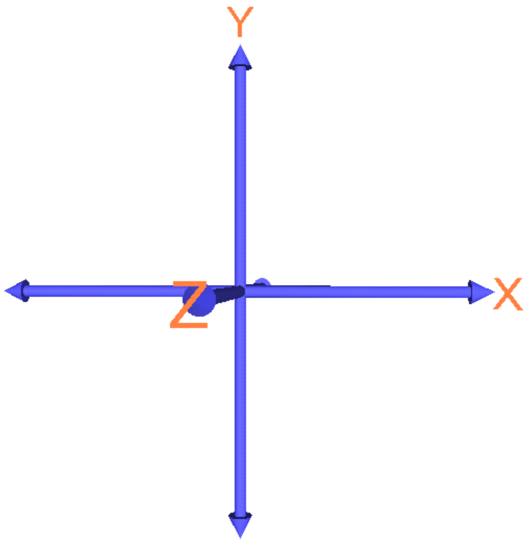


Y axis is up

Correspondence: North, Up, East

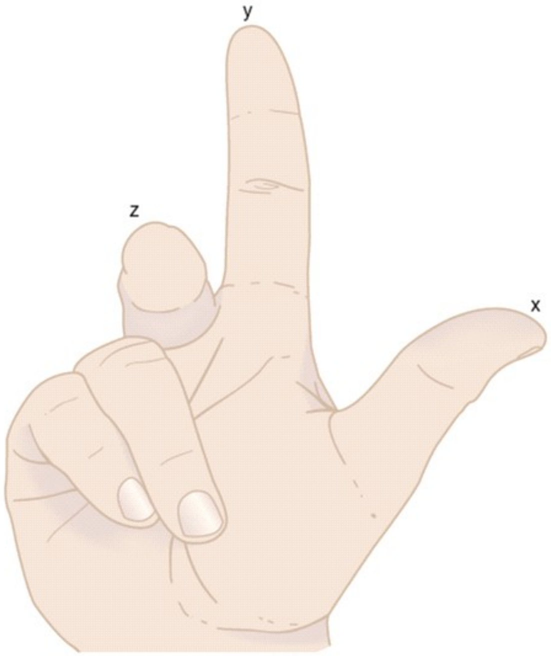
Accept no substitutes!

- or at least realign them ☺

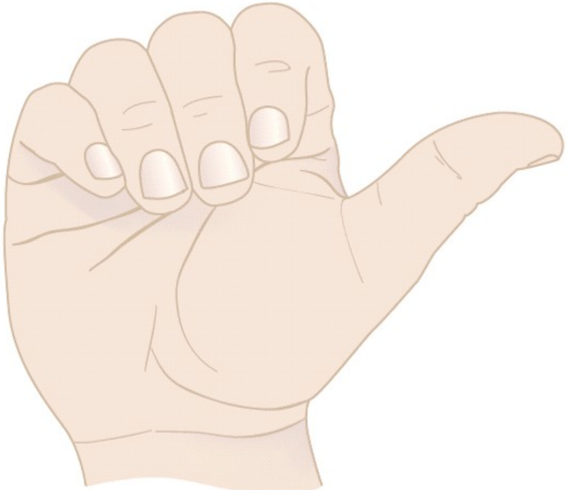


Right hand rules!

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



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



Shape and Geometry

Chapter 2

Shape and geometry

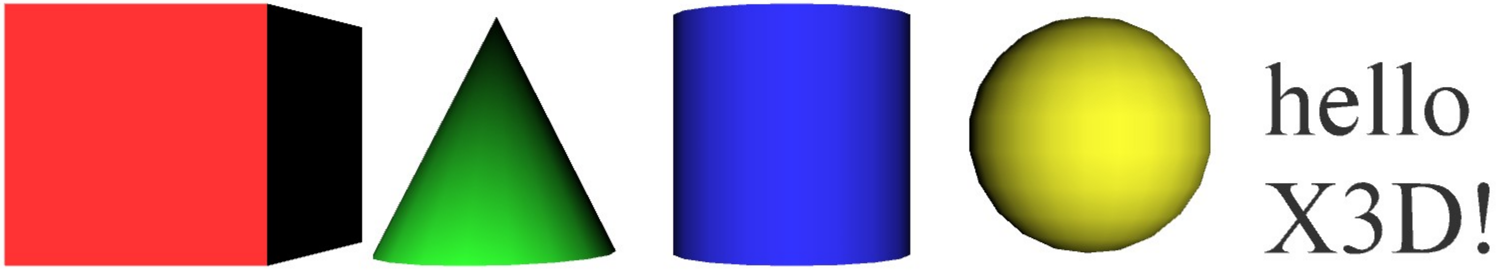
Shape nodes can contain a single geometry node

- For example, one of the five geometry primitive nodes
- Alternatively contains a more-advanced geometry node
  - Chapter 2: Geometric primitives
  - Chapter 6: Points Lines and Polygon nodes
  - Chapter 10: Geometry2D nodes
  - Chapter 13: Triangle nodes

Shape nodes can also contain an Appearance node

- Which in turn contains a Material node for coloring
- Covered in Chapter 3

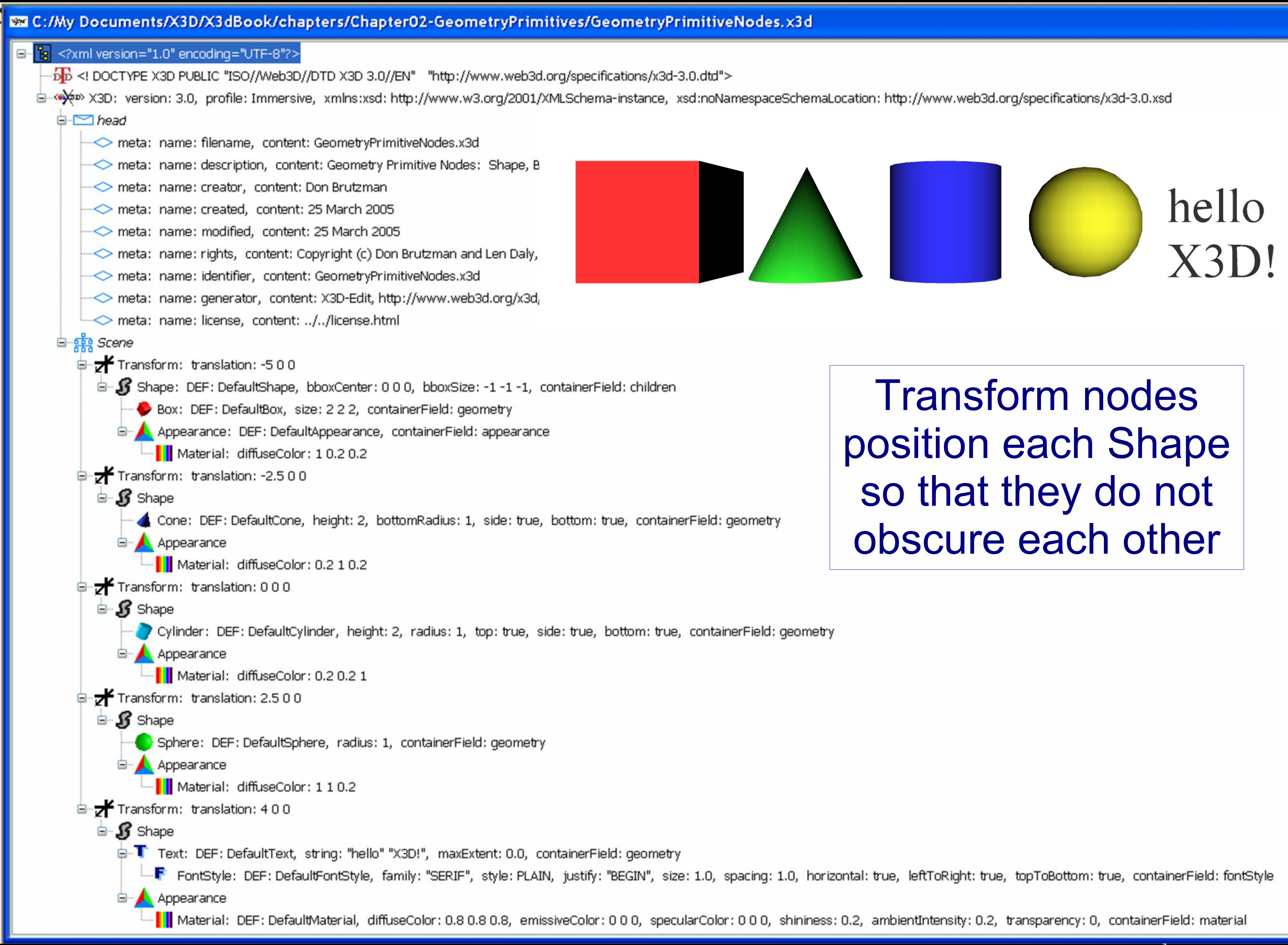
Geometry Primitives



Primitives are simple geometric constructs

Shape, geometry, Appearance, Material pattern

Browsers decide implementation details, including tessellation (polygon count) and thus quality



C:/My Documents/X3D/X3dBook/chapters/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d

```
<?xml version="1.0" encoding="UTF-8"?>
  <! DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"   "http://www.web3d.org/specifications/x3d-3.0.dtd">
  X3D:  version: 3.0,  profile: Immersive,  xmlns:xsd: http://www.w3.org/2001/XMLSchema-instance,  xsd:noNamespaceSchemaLocation: http://www.web3d.org/specifications/x3d-3.0.xsd
    head
      meta:  name: filename,  content: GeometryPrimitiveNodes.x3d
      meta:  name: description,  content: Geometry Primitive Nodes:  Shape, B
      meta:  name: creator,  content: Don Brutzman
      meta:  name: created,  content: 25 March 2005
      meta:  name: modified,  content: 25 March 2005
      meta:  name: rights,  content: Copyright (c) Don Brutzman and Len Daly,
      meta:  name: identifier,  content: GeometryPrimitiveNodes.x3d
      meta:  name: generator,  content: X3D-Edit, http://www.web3d.org/x3d,
      meta:  name: license,  content: ../../license.html
    Scene
      Transform:  translation: -5 0 0
        Shape:  DEF: DefaultShape,  bboxCenter: 0 0 0,  bboxSize: -1 -1 -1,  containerField: children
          Box:  DEF: DefaultBox,  size: 2 2 2,  containerField: geometry
          Appearance:  DEF: DefaultAppearance,  containerField: appearance
            Material:  diffuseColor: 1 0.2 0.2
      Transform:  translation: -2.5 0 0
        Shape
          Cone:  DEF: DefaultCone,  height: 2,  bottomRadius: 1,  side: true,  bottom: true,  containerField: geometry
          Appearance
            Material:  diffuseColor: 0.2 1 0.2
      Transform:  translation: 0 0 0
        Shape
          Cylinder:  DEF: DefaultCylinder,  height: 2,  radius: 1,  top: true,  side: true,  bottom: true,  containerField: geometry
          Appearance
            Material:  diffuseColor: 0.2 0.2 1
      Transform:  translation: 2.5 0 0
        Shape
          Sphere:  DEF: DefaultSphere,  radius: 1,  containerField: geometry
          Appearance
            Material:  diffuseColor: 1 1 0.2
      Transform:  translation: 4 0 0
        Shape
          Text:  DEF: DefaultText,  string: "hello" "X3D!",  maxExtent: 0.0,  containerField: geometry
            FontStyle:  DEF: DefaultFontStyle,  family: "SERIF",  style: PLAIN,  justify: "BEGIN",  size: 1.0,  spacing: 1.0,  horizontal: true,  leftToRight: true,  topToBottom: true,  containerField: fontStyle
          Appearance
            Material:  DEF: DefaultMaterial,  diffuseColor: 0.8 0.8 0.8,  emissiveColor: 0 0 0,  specularColor: 0 0 0,  shininess: 0.2,  ambientIntensity: 0.2,  transparency: 0,  containerField: material
```

Transform nodes position each Shape so that they do not obscure each other

Shape parent with geometry child

```
<Shape>
    <Box size='1 2 3'/>
    <Appearance>
        <Material/>
    </Appearance>
</Shape>
```

Shape must be parent node, can only hold one geometry node

Appearance and Material nodes define colors, transparency, etc.

```
<Shape>
    <Sphere radius='1'/>
    <Appearance>
        <Material/>
    </Appearance>
</Shape>
```

Primitives have simple dimensions

- Typical volume ~1 m radius

All units are in meters

Geometry nodes

Chapter 2, Primitives

- Box, Cone, Cylinder, Sphere, Text / FontStyle

Chapter 6, Points Lines and Polygons

- PointSet, IndexedLineSet, IndexedFaceSet, ElevationGrid, Extrusion

Chapter 10, Geometry2D

- Arc2D,ArcClose2D, Circle2D, Disk2D, Polyline2D, Polypoint2D, Rectangle2D, TriangleSet2D

Chapter 13, Triangles and Quadrilaterals

- TriangleSet, TriangleStripSet, TriangleFanSet, QuadSet
- Both regular and Indexed versions

Advanced geometry nodes

Geospatial component

- GeoElevationGrid

NURBS component

- NurbsCurve, NurbsPatchSurface, NurbsSweptSurface, NurbsSwungSurface, NurbsTrimmedSurface

Programmable shaders component

- ComposedShader, PackagedShader, ProgramShader

Further information available in X3D Specification

- http://www.web3d.org/x3d/specifications

Grouping and Transformation

Chapter 3

Grouping rationale

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

Grouping nodes help provide sensible structure

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

Bounding boxes

Provides a hint to browsers about object size

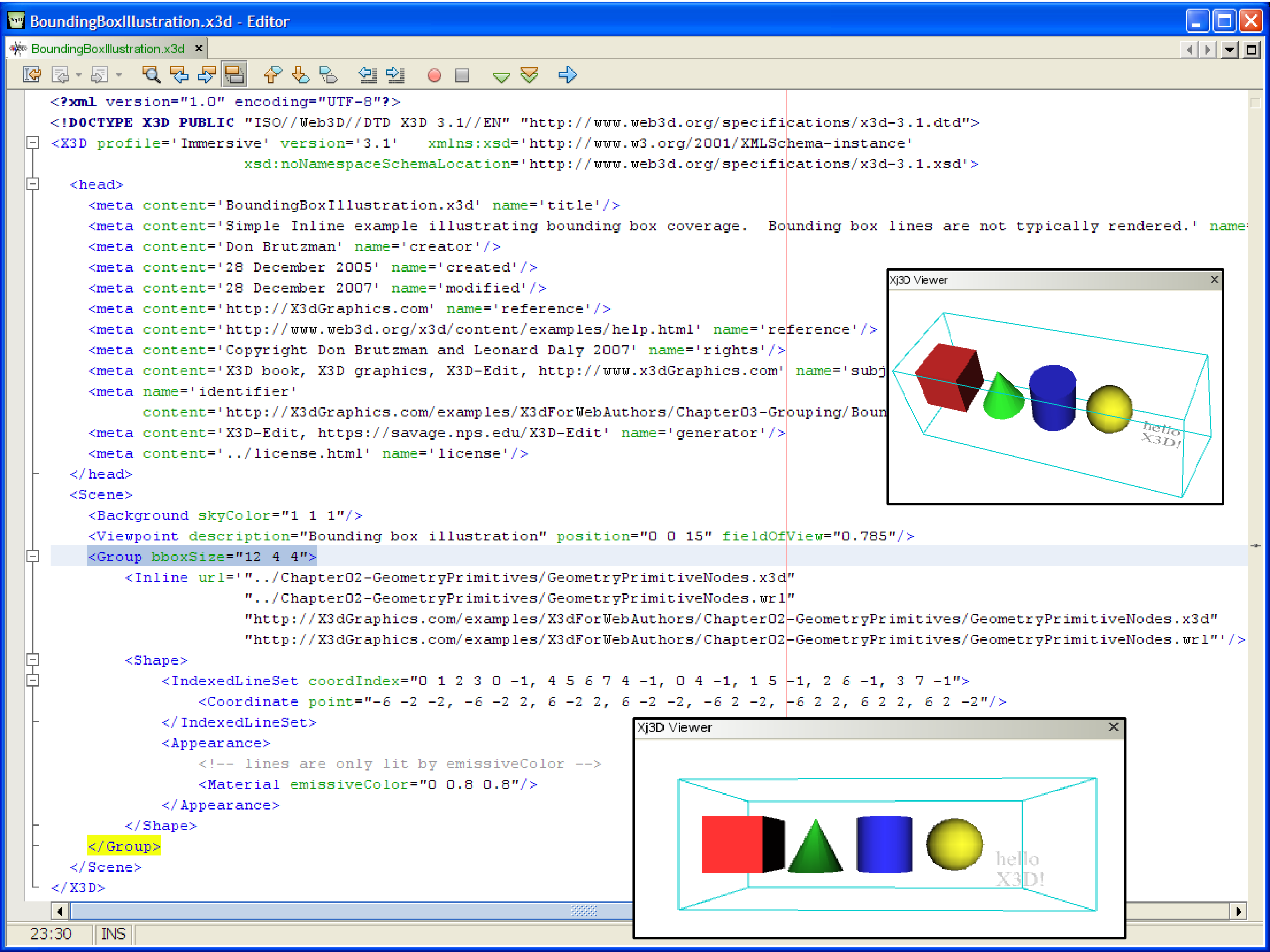
- Does not affect how an object is rendered (drawn) if it is actually larger than the bounding box
- Are never drawn themselves
- Defined by *bboxSize* and *bboxCenter*

Goal is to reduce computational complexity

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

Bounding boxes can be ignored by authors

- some authoring tools can provide them if needed


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

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

Transform node

Grouping node that defines a coordinate system for its children

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

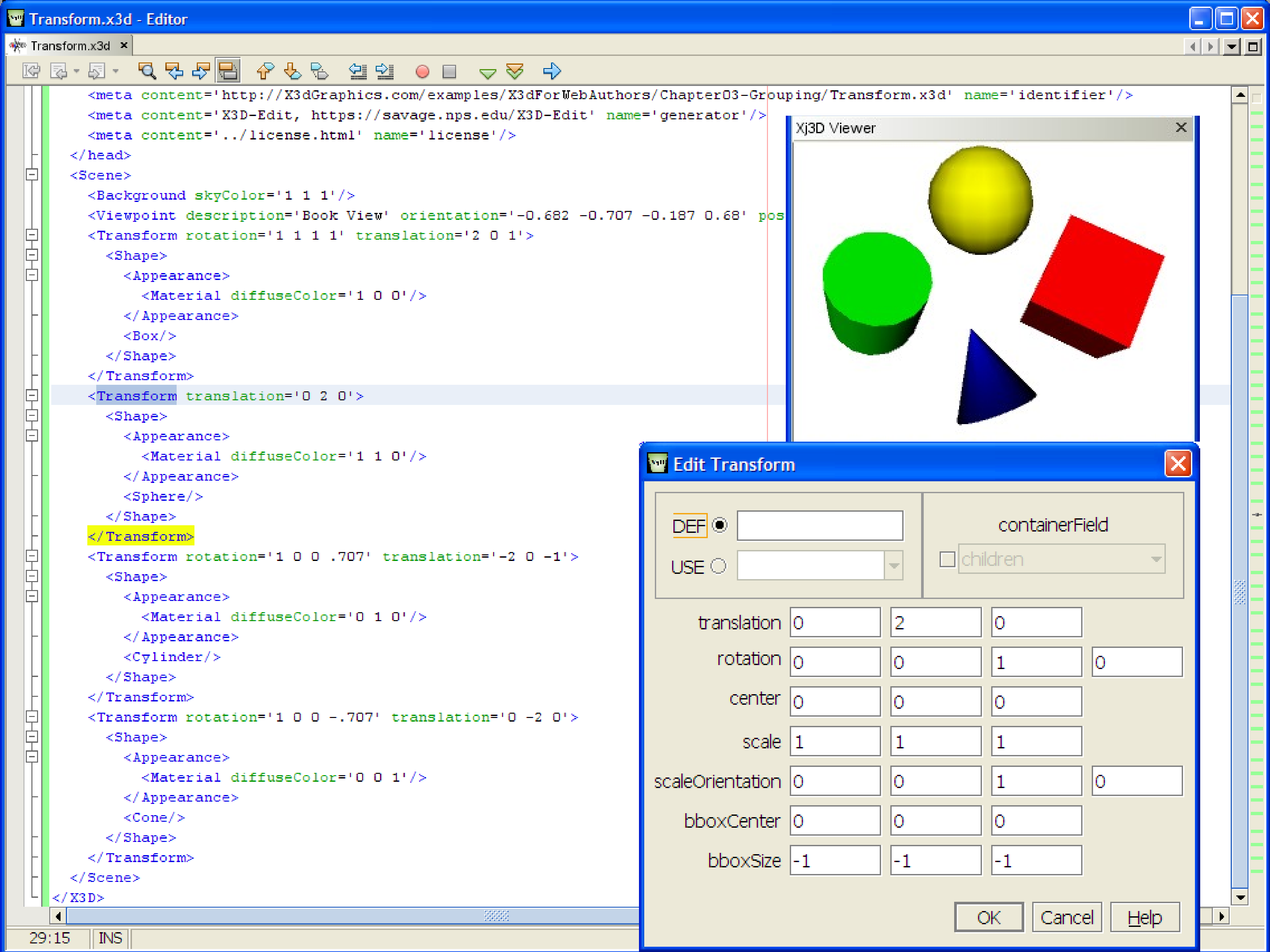
Transform nodes can

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

Group and Transform are among most commonly used nodes

Transform fields

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



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

DEF ●

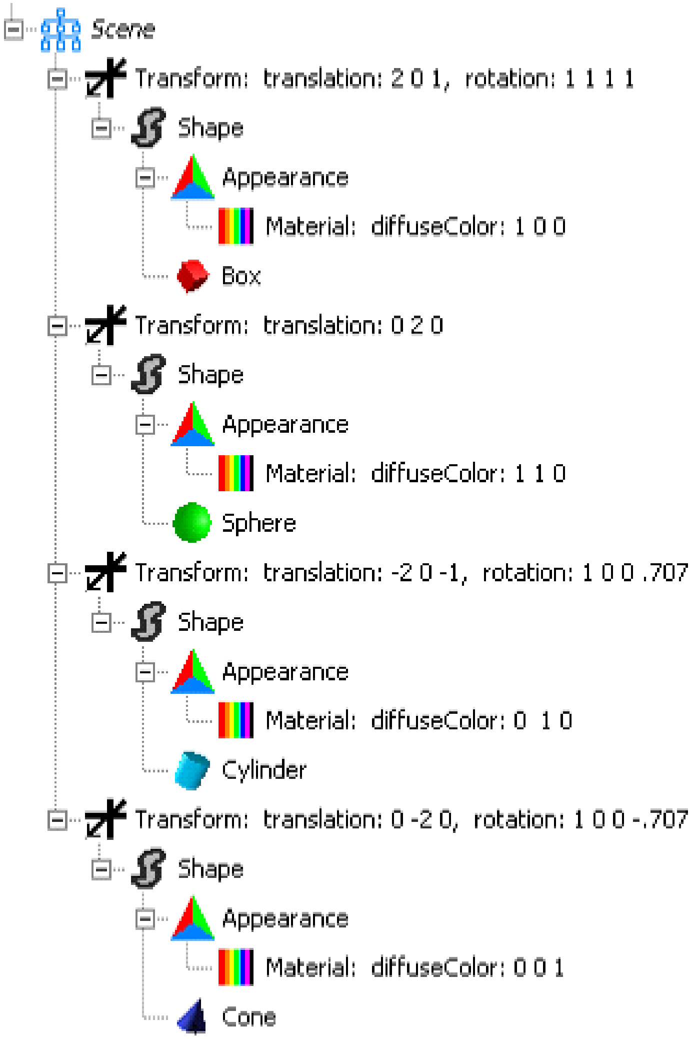
USE ○

containerField

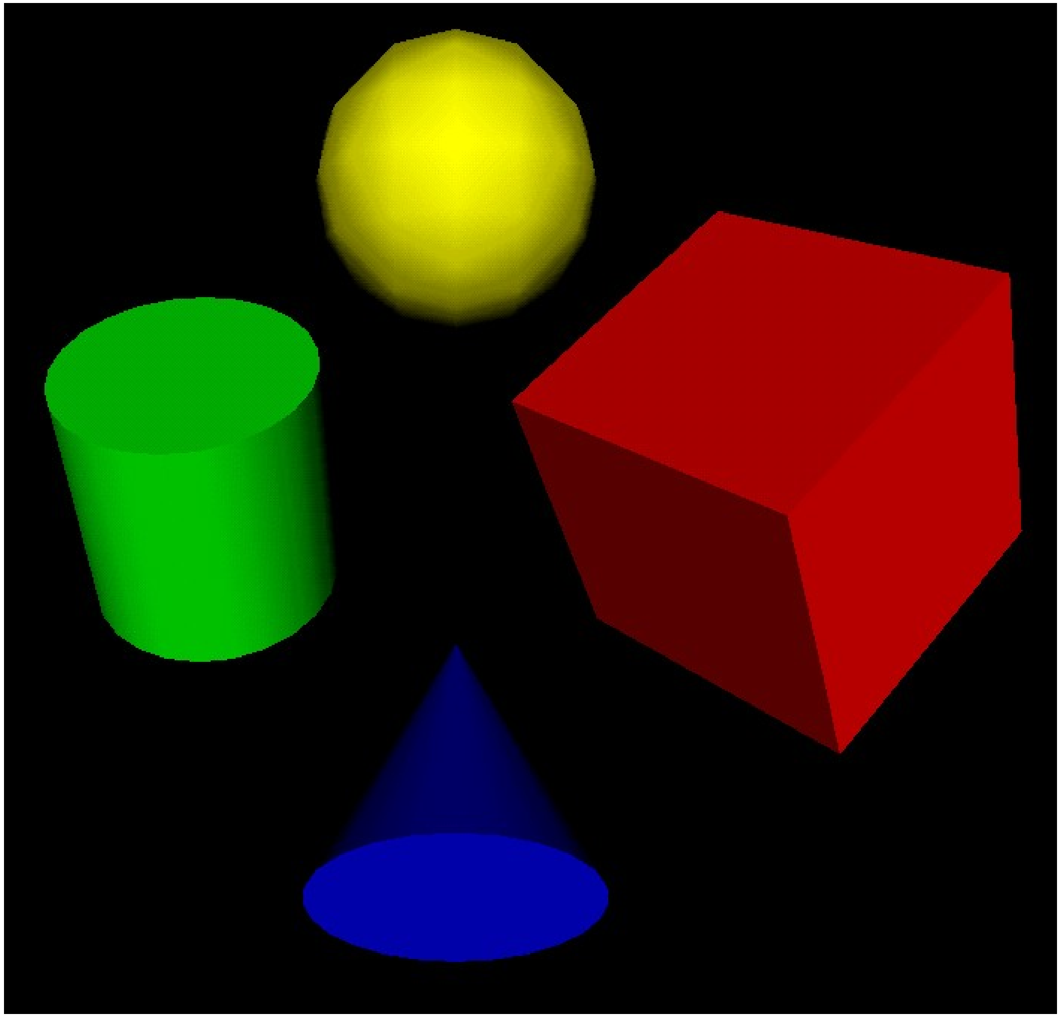
□ children

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

Each Transform is a scene subgraph



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



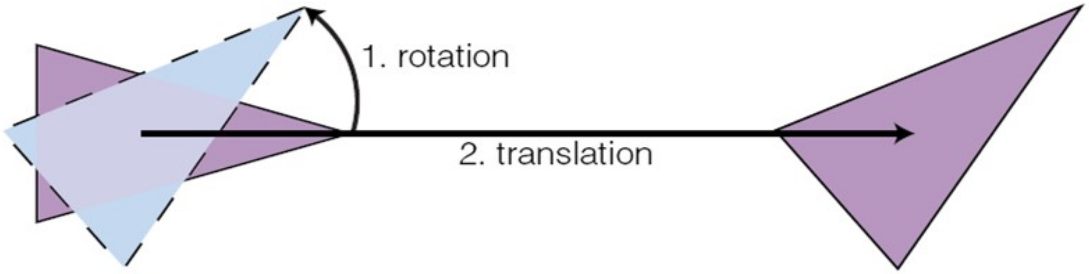
Order of transformation operations

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

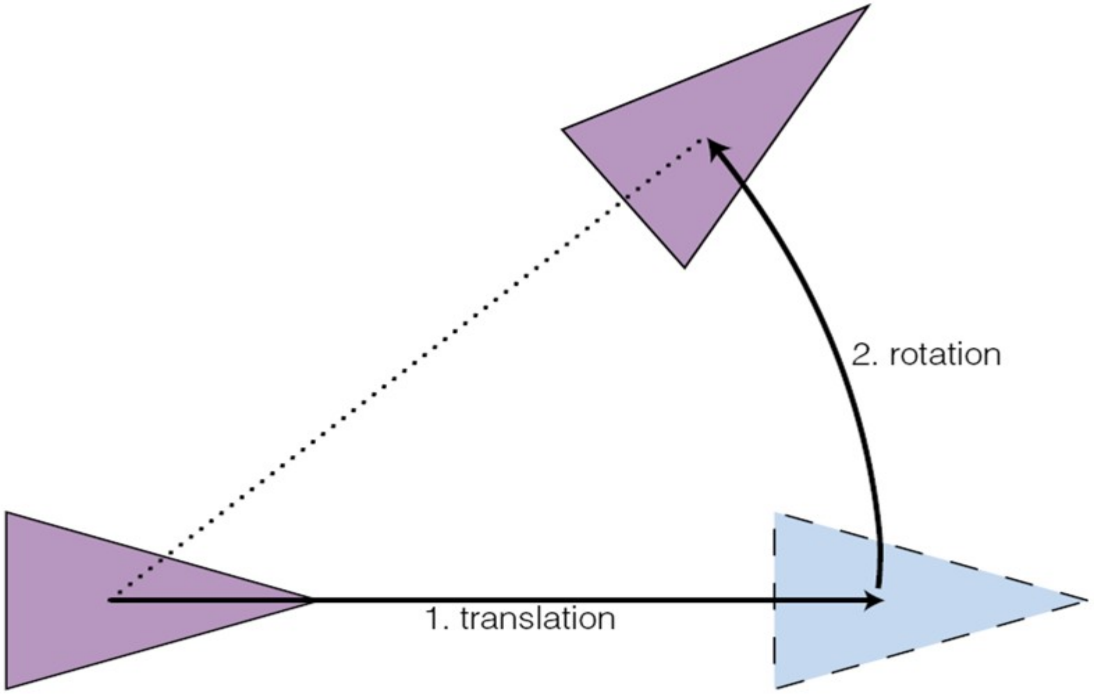
- Apply reverse *center* offset to set up for properly centered scaling and orientation operations
- Apply reverse *scaleOrientation*, then apply *scale* operation, then apply forward *scaleOrientation* to regain initial frame
- Apply *rotation* to final direction, then apply forward *center* offset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

Comparing out-of-order operations

Case 1



Case 2



Equivalent transformations

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

Using matrix transformation notation, where

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

are the equivalent transformation matrices, then

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

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

Matrix operations

Matrix operations are not directly exposed in X3D

- Unlike most imperative programming interfaces
- Instead Transform nodes provide a regularized way to perform translation, rotation, scaling

Transform includes a specific order of operations

- Illustrated in next slides

Flexible: multiple Transform nodes can be nested

- Each Transform establishes new coordinate frame

Inline node

Loads another X3D world within current scene

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

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

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

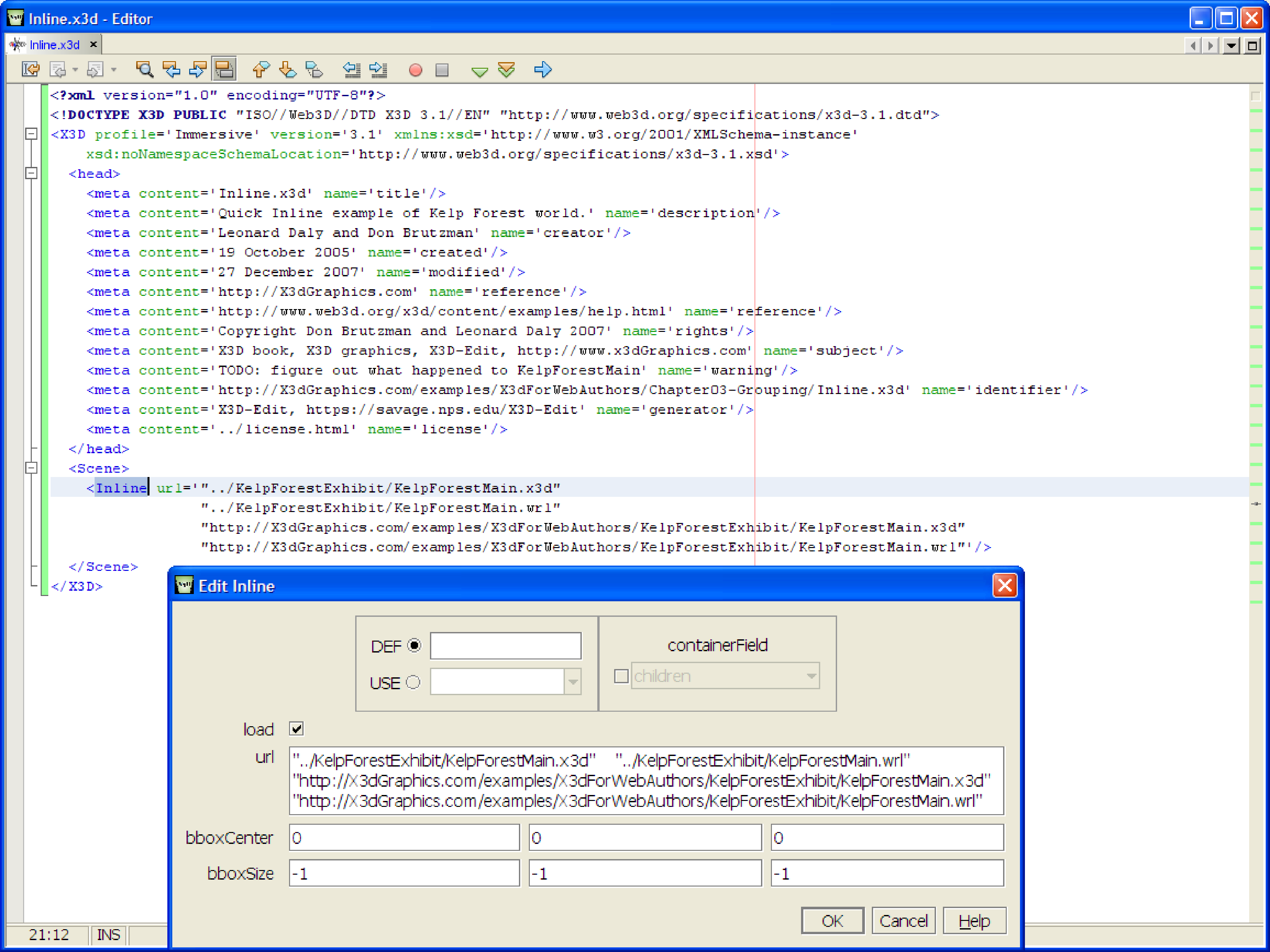
url field

url = uniform resource locator

- Equivalent to universal resource identifier (uri)

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

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



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//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='Inline.x3d' name='title'/>
    <meta content='Quick Inline example of Kelp Forest world.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='19 October 2005' name='created'/>
    <meta content='27 December 2007' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright Don Brutzman and Leonard Daly 2007' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='TODO: figure out what happened to KelpForestMain' name='warning'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter03-Grouping/Inline.x3d' name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../license.html' name='license'/>
  </head>
  <Scene>
    <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>
</X3D>
```

Edit Inline

DEF ●

USE ○

containerField ☐ children

load ☑

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

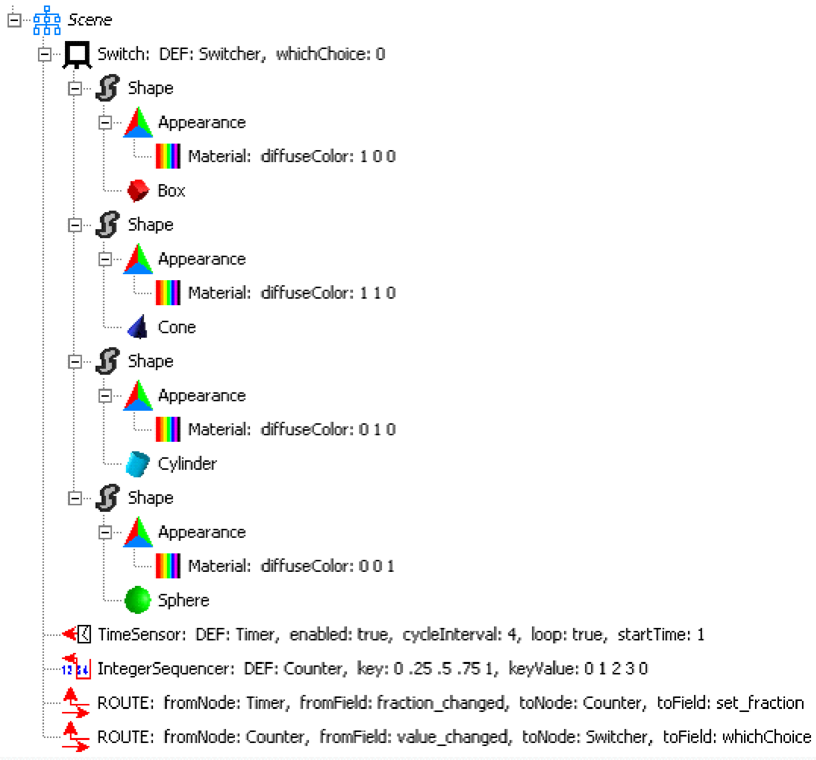
bboxCenter 0 0 0

bboxSize -1 -1 -1

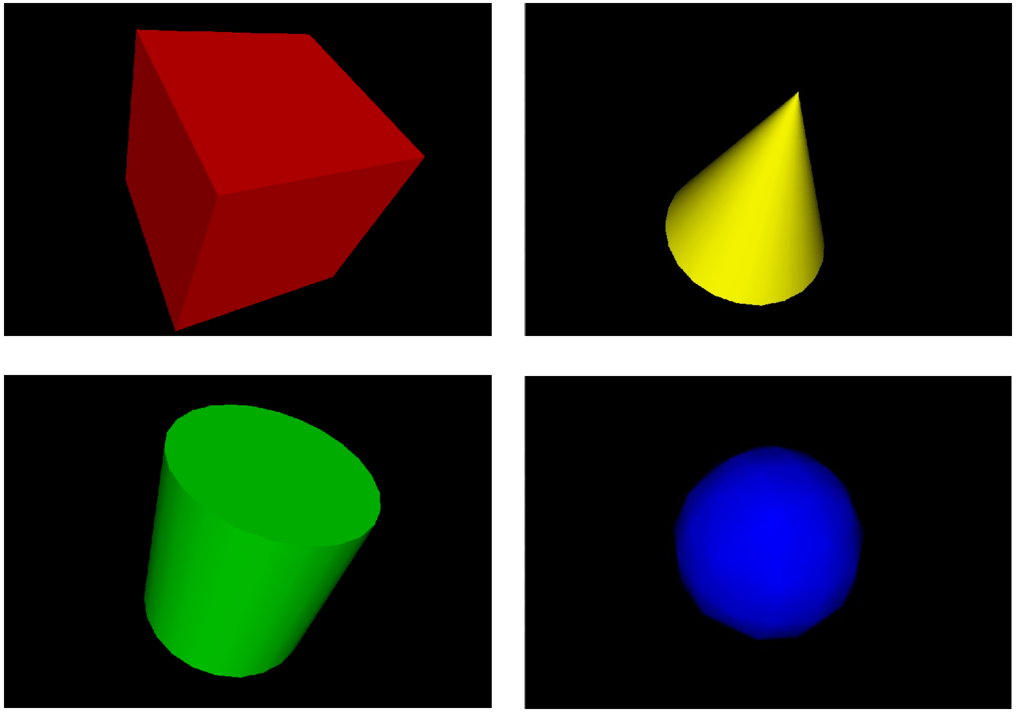
Switch node

- Switch selects only one (or none) of its children for rendering
  - Initial child index is *whichChoice*='0'
  - *whichChoice*='-1' indicates no child is selected
- Can manually change values
  - Sometimes better to hide geometry rather than to comment out large blocks
    - (which may already have embedded comments)
  - Chapter 7 Event Animation describes how to change selections using event animation

Switch node example



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



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

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

Viewing and Navigation

Chapter 4

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

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

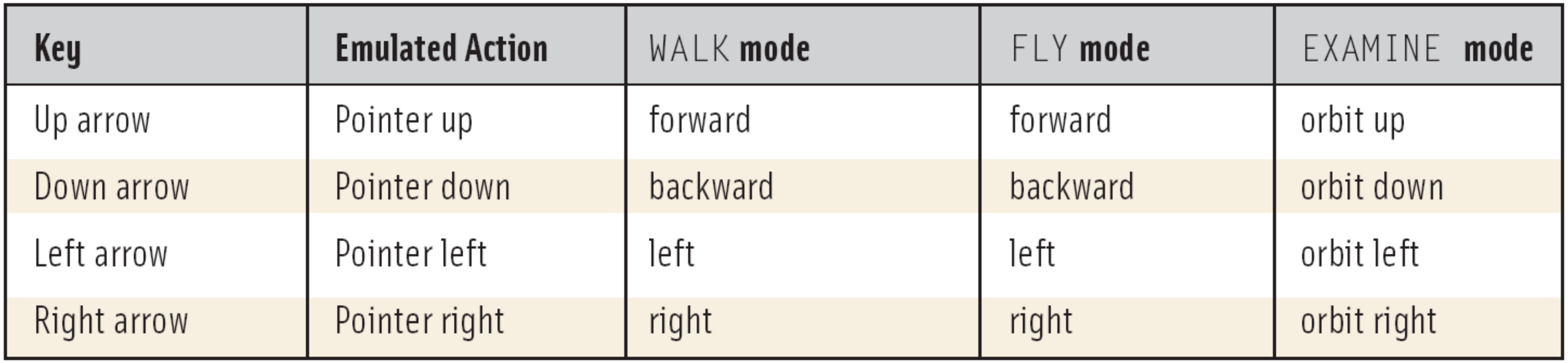
Navigation model 1

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

Anchor node

Anchor is another grouping node that can contain other nodes

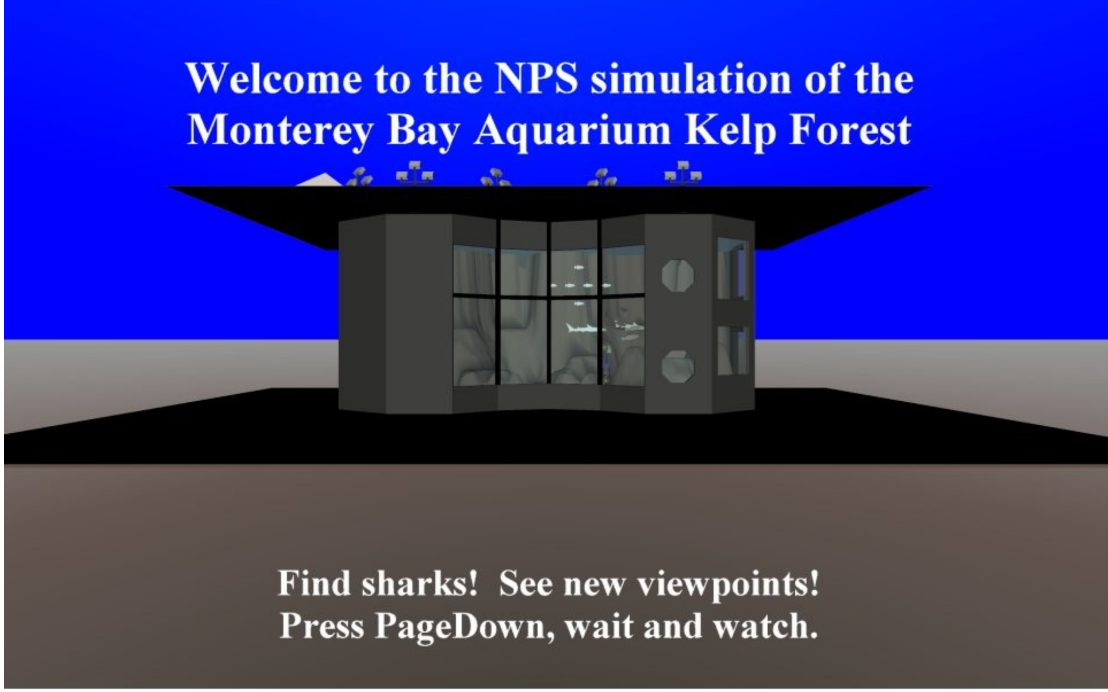
Geometry rendered by contained nodes is activated and can be selected

- 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

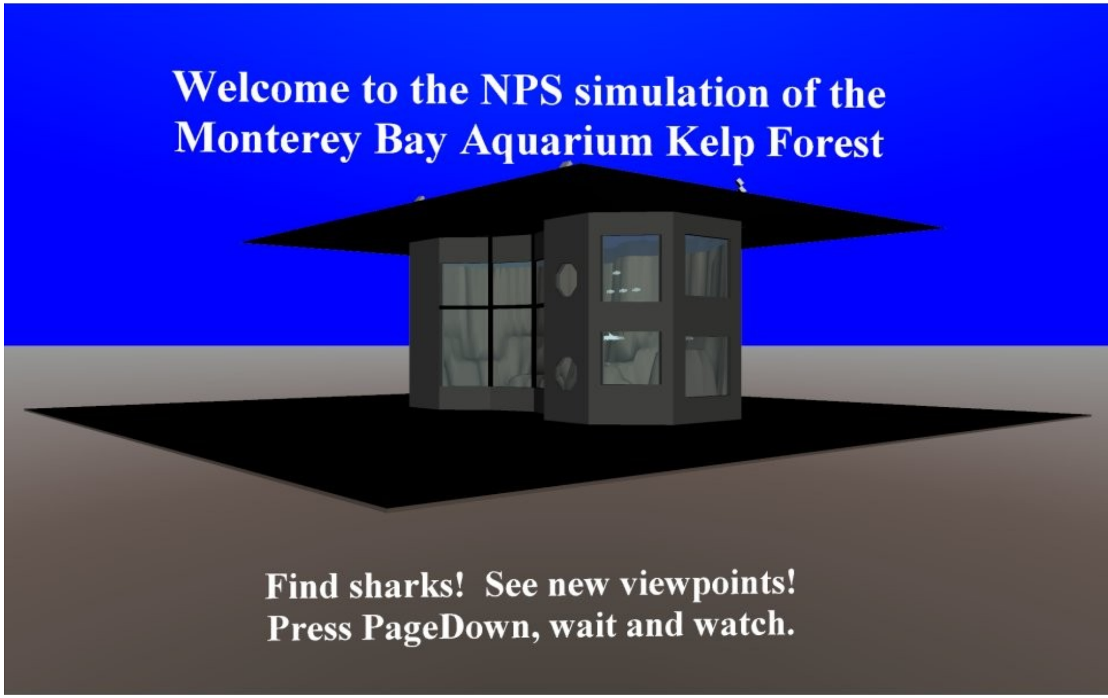
Billboard example

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



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.

web3D CONSORTIUM

Appearance, Material and Textures

Chapter 5

Appearance node

Each Shape contains some geometry along with a corresponding Appearance node

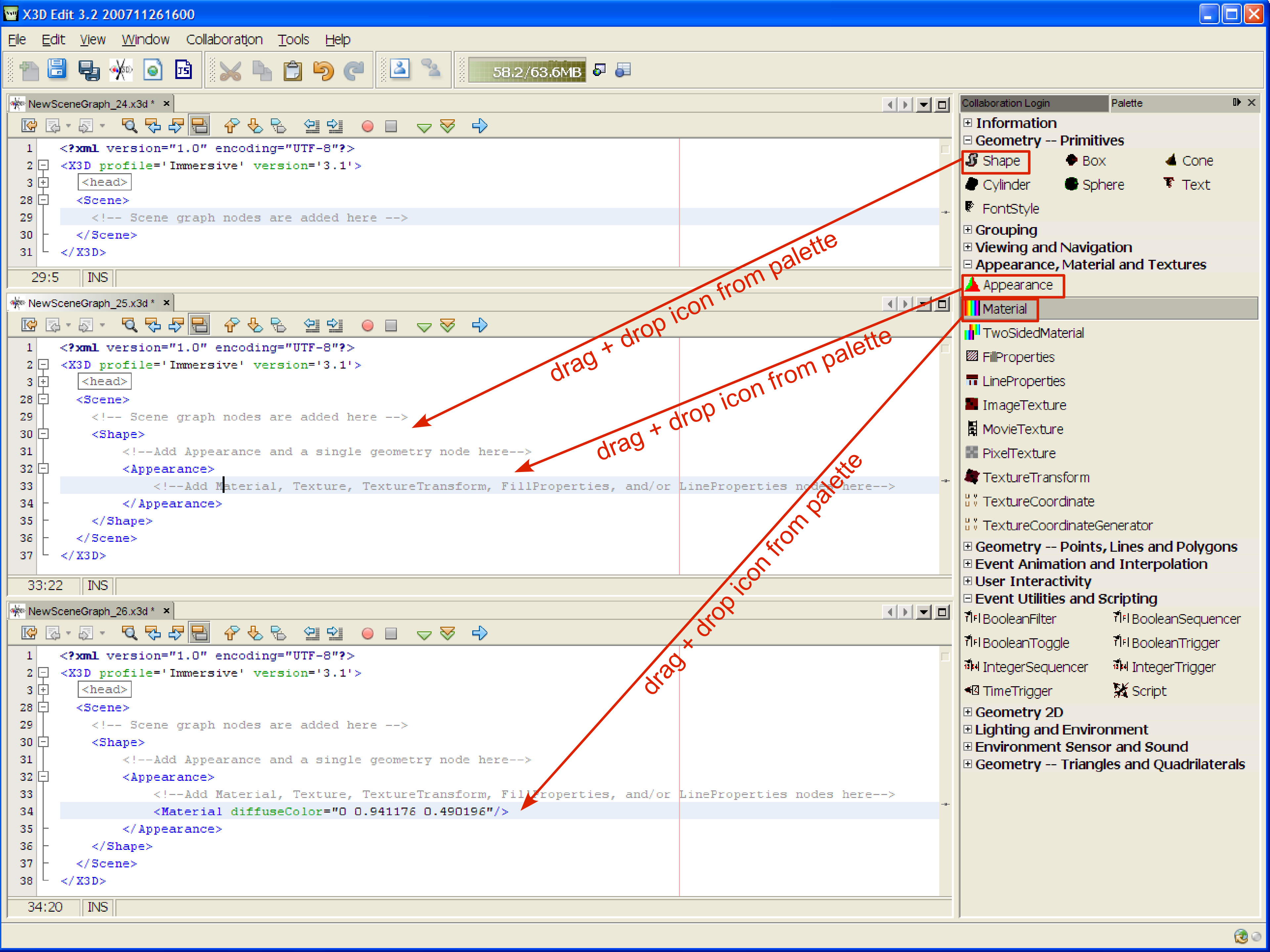
Appearance is a container which may include

- Single Material (or TwoSidedMaterial) node
- FillProperties, LineProperties, single Texture node

This close association makes assignment of rendering properties to geometry unambiguous

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

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



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

drag + drop icon from palette

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

drag + drop icon from palette

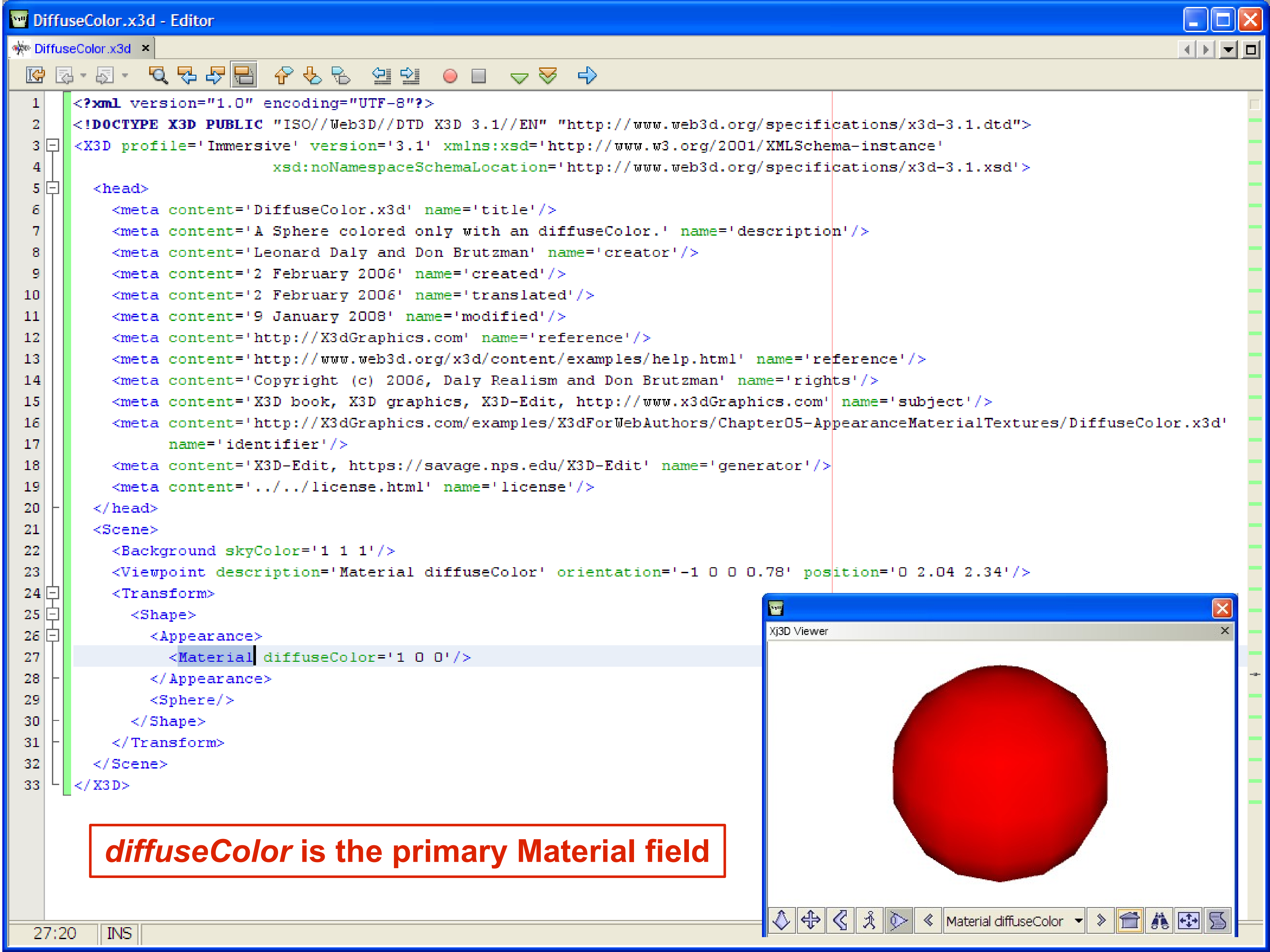
```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' version='3.1'>
  <head>
  <Scene>
    <!-- Scene graph nodes are added here -->
    <Shape>
        <!--Add Appearance and a single geometry node here-->
        <Appearance>
            <!--Add Material, Texture, TextureTransform, FillProperties,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

drag + drop icon from palette

Material fields

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

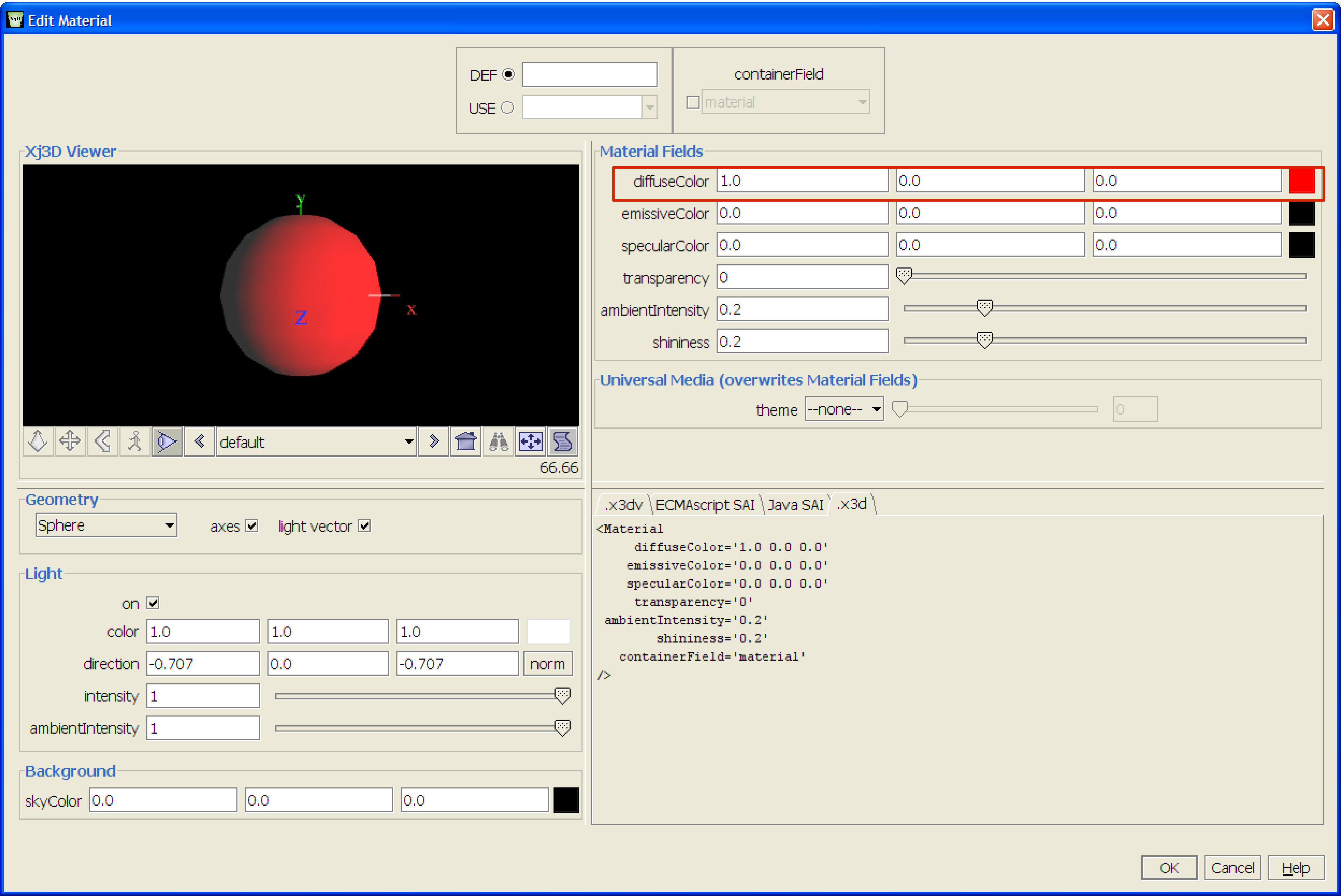
- *diffuseColor* reflects all X3D light sources, depending on viewing angles towards each light
- *ambientIntensity* is reflection multiplication factor
- *emissiveColor* is glowing component, normally off, independent of reflected light
- *specularColor* governs reflection highlights
- *shininess* controls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



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Immersive' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='DiffuseColor.x3d' name='title'/>
    <meta content='A Sphere colored only with an diffuseColor.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator'/>
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='9 January 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/help.html' name='reference'/>
    <meta content='Copyright (c) 2006, Daly Realism and Don Brutzman' name='rights'/>
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.x3dGraphics.com' name='subject'/>
    <meta content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter05-AppearanceMaterialTextures/DiffuseColor.x3d'
          name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='../../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <Viewpoint description='Material diffuseColor' orientation='-1 0 0 0.78' position='0 2.04 2.34'/>
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

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

Material editor: *diffuseColor*



Edit Material

DEF ● USE ○

containerField □ material

Xj3D Viewer

default

66.66

Geometry

Sphere axes ☑ light vector ☑

Light

on ☑

color 1.0 1.0 1.0

direction -0.707 0.0 -0.707 norm

intensity 1

ambientIntensity 1

Background

skyColor 0.0 0.0 0.0 ■

Material Fields

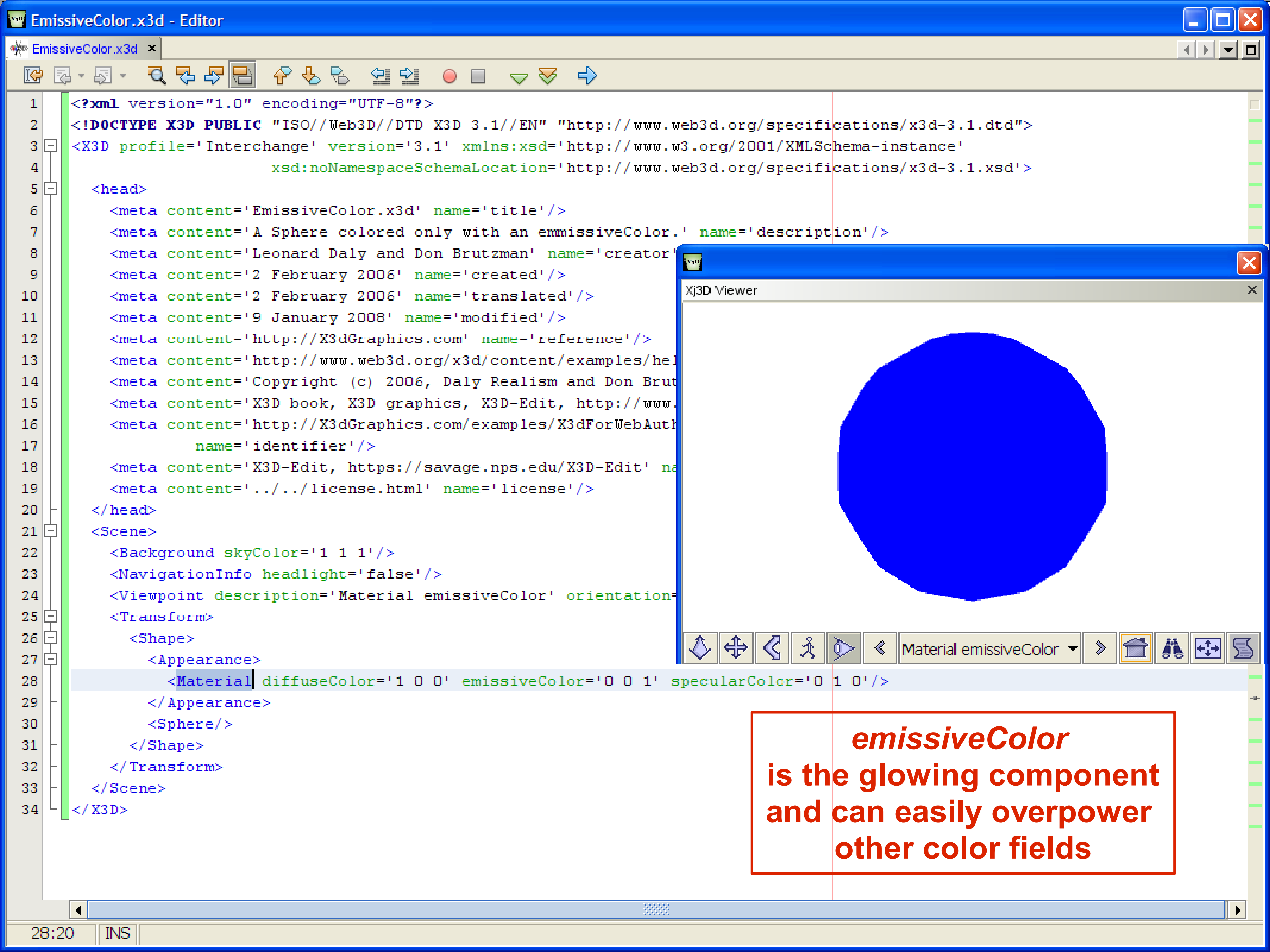
diffuseColor	1.0	0.0	0.0	🟥
emissiveColor	0.0	0.0	0.0	■
specularColor	0.0	0.0	0.0	■
transparency	0			
ambientIntensity	0.2			
shininess	0.2			

Universal Media (overwrites Material Fields)

theme --none-- 0

.x3dv ECMAscript SAI Java SAI .x3d

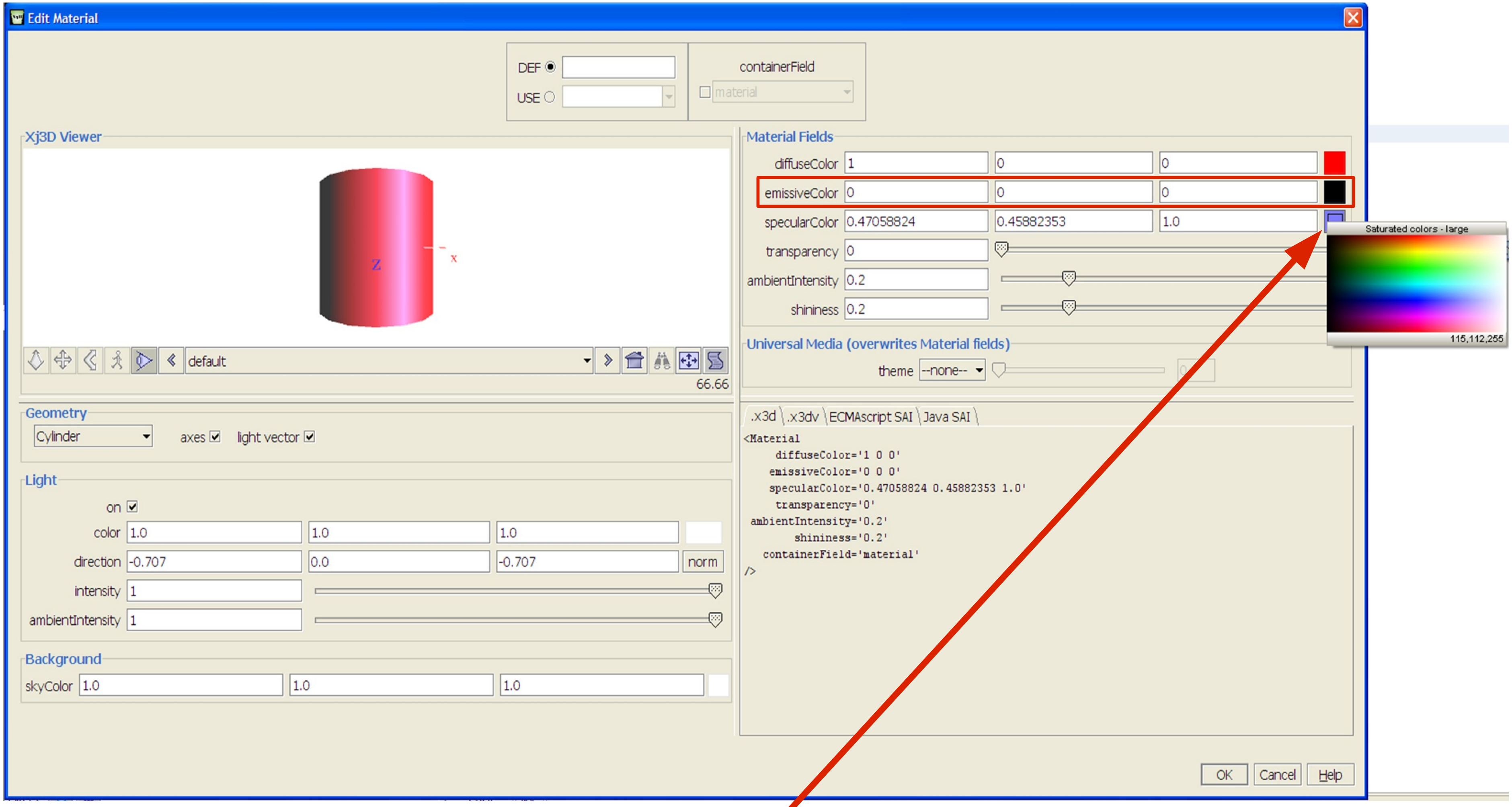
```
<Material
     diffuseColor='1.0 0.0 0.0'
    emissiveColor='0.0 0.0 0.0'
    specularColor='0.0 0.0 0.0'
     transparency='0'
 ambientIntensity='0.2'
        shininess='0.2'
   containerField='material'
/>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.1.dtd">
<X3D profile='Interchange' version='3.1' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
                     xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.1.xsd'>
  <head>
    <meta content='EmissiveColor.x3d' name='title'/>
    <meta content='A Sphere colored only with an emmissiveColor.' name='description'/>
    <meta content='Leonard Daly and Don Brutzman' name='creator
    <meta content='2 February 2006' name='created'/>
    <meta content='2 February 2006' name='translated'/>
    <meta content='9 January 2008' name='modified'/>
    <meta content='http://X3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/he
    <meta content='Copyright (c) 2006, Daly Realism and Don Brut
    <meta content='X3D book, X3D graphics, X3D-Edit, http://www.
    <meta content='http://X3dGraphics.com/examples/X3dForWebAut
             name='identifier'/>
    <meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' na
    <meta content='../../license.html' name='license'/>
  </head>
  <Scene>
    <Background skyColor='1 1 1'/>
    <NavigationInfo headlight='false'/>
    <Viewpoint description='Material emissiveColor' orientation=
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor='1 0 0' emissiveColor='0 0 1' specularColor='0 1 0'/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

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

Material editor color selector



DEF ● USE ○

containerField ☐ material

Geometry: Cylinder — axes ☑ light vector ☑

Light

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

Background

skyColor	1.0	1.0	1.0

Material Fields

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

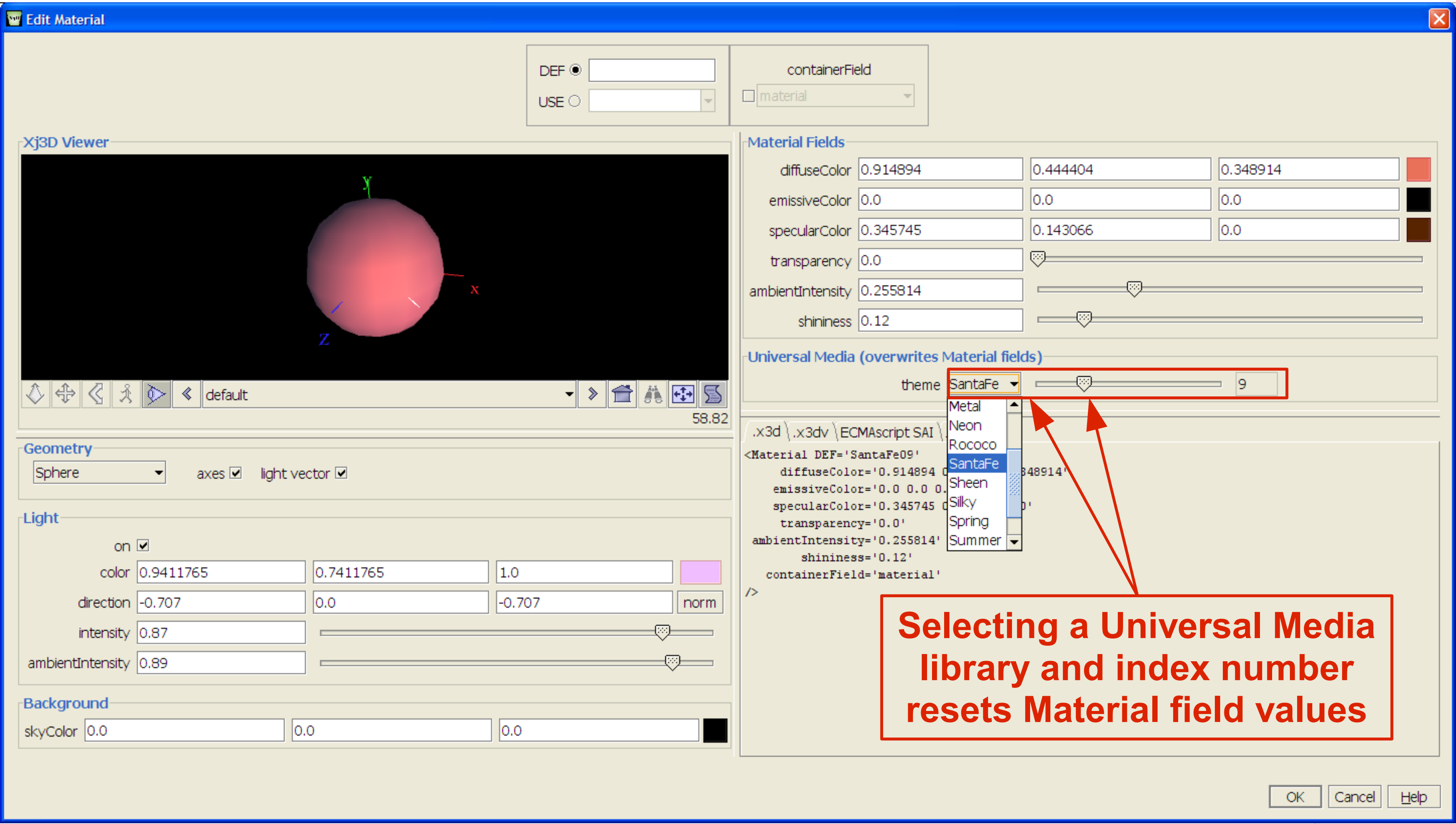
Universal Media (overwrites Material fields)

theme --none--

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

Click colored box to select a color

Selecting a Universal Material value



DEF ●

USE ○

containerField

□ material

Material Fields

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

Universal Media (overwrites Material fields)

theme SantaFe 9

Geometry

Sphere axes ☑ light vector ☑

Light

on ☑

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

Background

skyColor	0.0	0.0	0.0

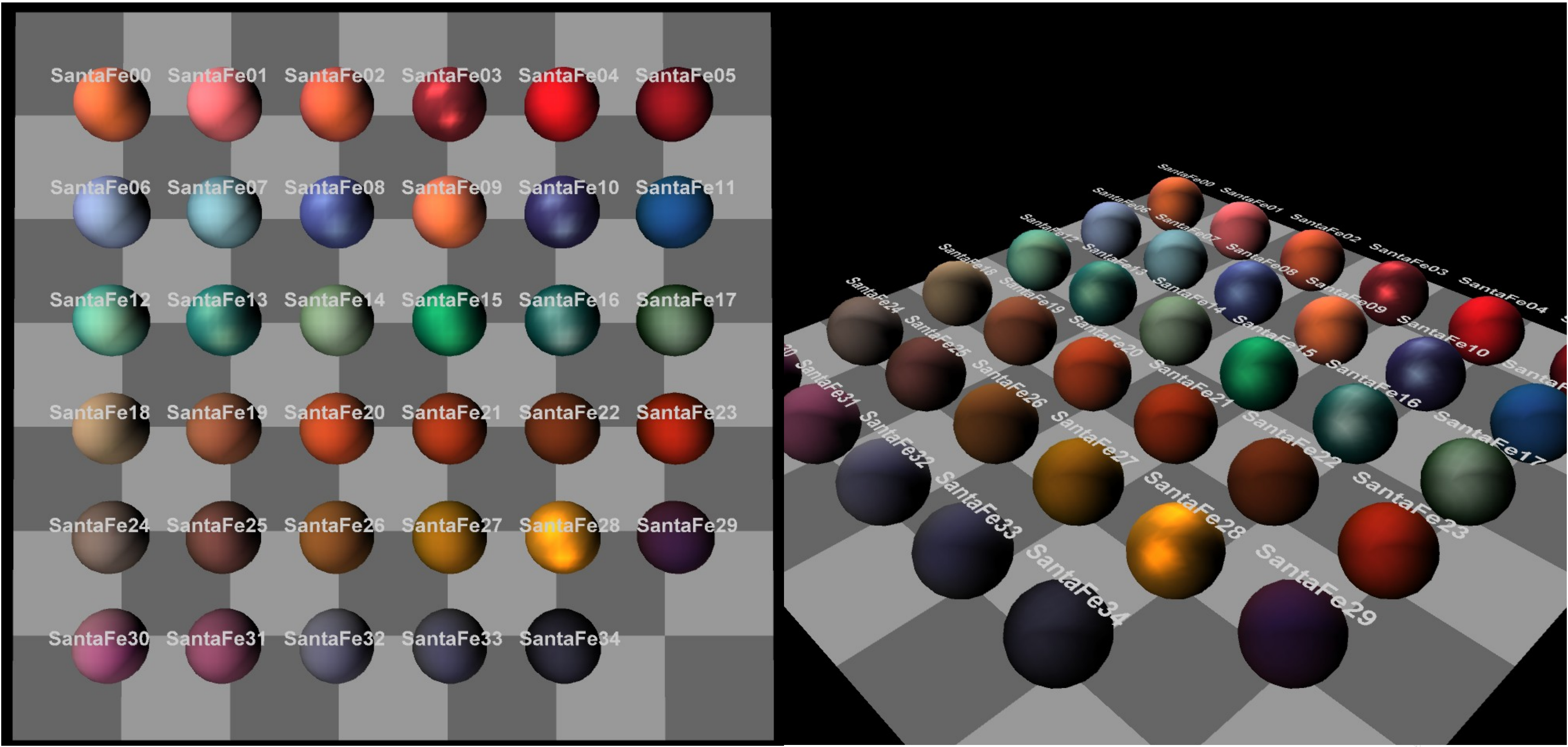
```
<Material DEF='SantaFe09'
     diffuseColor='0.914894 0            348914'
    emissiveColor='0.0 0.0 0.
    specularColor='0.345745 0            0'
     transparency='0.0'
 ambientIntensity='0.255814'
        shininess='0.12'
   containerField='material'
/>
```

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

Universal Media libraries include

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

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



Texture nodes

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

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

Texture images can be shifted, rotated, scaled

- TextureTransform, TextureCoordinate
- Thus modifying image application to geometry

Animation Behavior Examples

Behaviors

Behavior defined as changing the value of some field contained by some node in scene graph

Animation nodes, user interaction nodes and network updates can produce updated values

ROUTE statements connect output of one node as an input to field in another node

Event defined as the time-stamped value passed by a ROUTE, from one field to another

Thus the values held by nodes in scene graph can change as time advances

Behavior traversal of scene graph

Once frame is swapped to update screen image, need to update values in the scene

Event model consists of

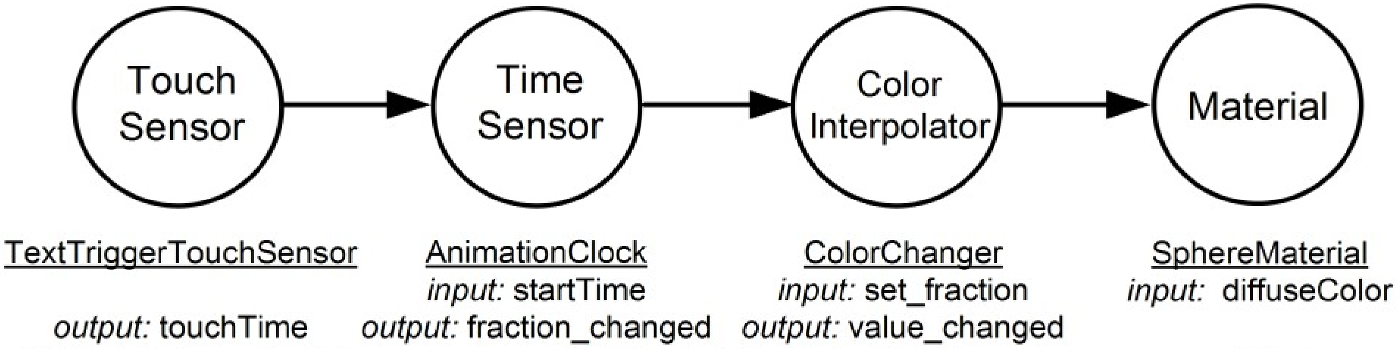
- Examining clock-driven and user-initiated events
- Updating scene-graph values
- Triggering and updating new events as appropriate
- Continue until all events handled, loops not allowed

Event updates modify the scene graph

- Changing rendering properties, or
- Generating further event outputs

Example behavior event chain

- User clicks button to start a timer clock
- Clock outputs new event at start of each frame,
- ... which stimulates linear-interpolation function which produces another output value
- ... which updates some target value in scene graph
- Repeat event traversal after each frame redraw



ROUTE connections

ROUTE connection enables the output field of one node to pass a value that then stimulates the input field of another node

- The passed value also includes a time stamp

Field data type and accessType must both match between node/field of source and target

- Chapter 1, Technical Introduction lists field types
- Also provided in tooltips and specification
- Authors usually must carefully check these

Animation as scene-graph modification

Behavior = changing a field value in a node, somewhere in the scene graph

Event = time-stamped value going over a ROUTE

Event cascade = a series of events being sent, each one triggering the next

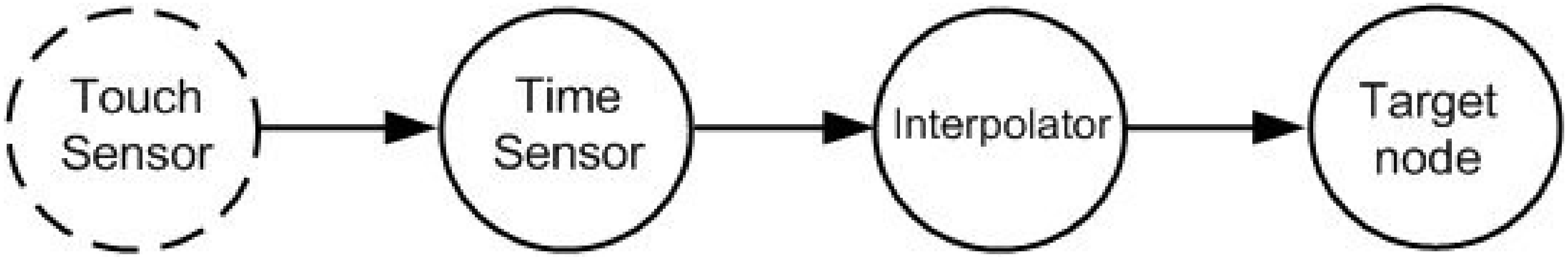
- No event loops allowed, guaranteeing completion

Thus all X3D animation can be considered as modification of the scene graph at run time

Event-animation design pattern

X3D can be imposing, there are many nodes

Nevertheless a simple design pattern is used for nearly every kind of animation



This consistent event ROUTE pattern enables you to expertly animate most X3D scene behaviors

Visualizing scenes on paper

It is good practice to sketch out 3D scene drafts

- Consider what models are needed, and how multiple models might be composed

Consider user experience, from their perspective

- What tasks and goals, what use cases
- What might things look like when first seen

Storyboarding can help build long-form content

- Series of vignettes to tell a larger story
- Each scene defines needed models and behaviors
- Build each piece, put them together

Importance of user interaction

Animated scenes are more interesting than static unchanging geometry

X3D interaction consists of sensing user actions and then prompting appropriate responses

Scenes that include behaviors which respond to user direction and control are more lively

Freedom of navigation and interaction contribute to user's sense of presence and immersion

Thus animation behaviors tend to be reactive and declarative, responding to the user

Sensors produce events

Sensors detect various kinds of user interaction and produce events to ROUTE within a scene

- Each sensor detects a certain kind of interaction, then produces one or more events

Authors decide how the events describing user interaction are interpreted and handled

- This approach allows great flexibility for authors

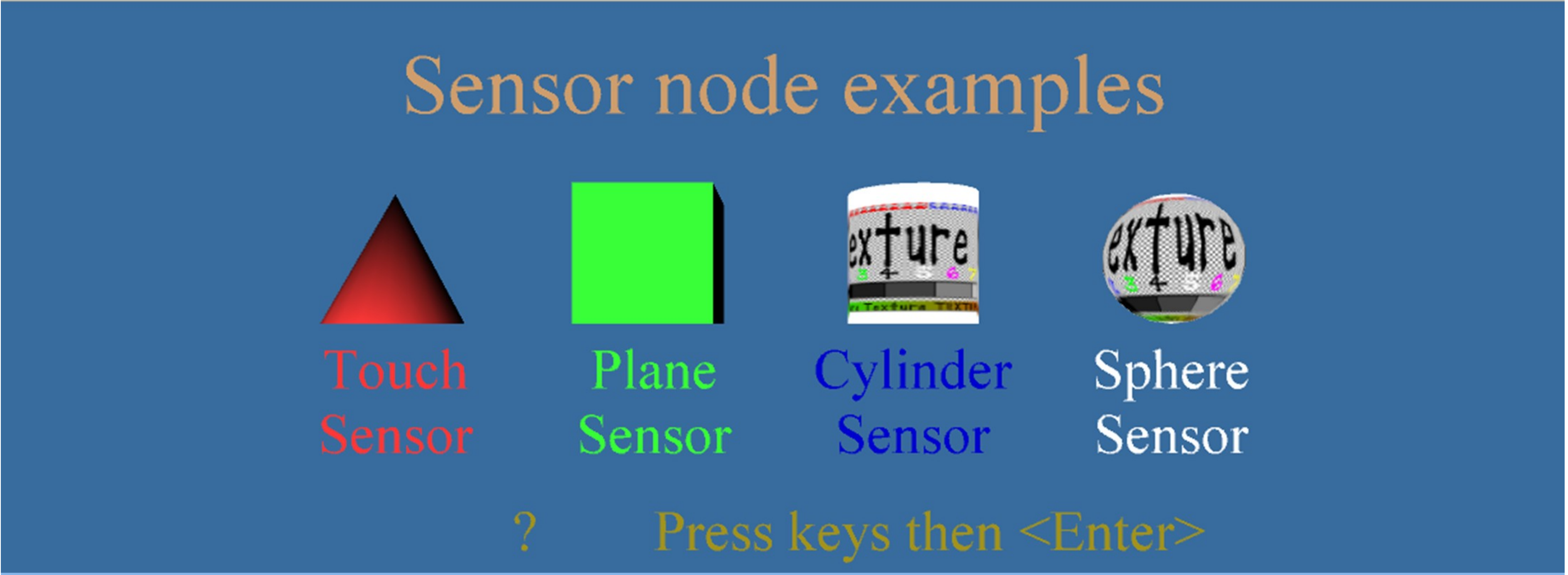
Example: user-interactivity sensor nodes

UserInteractivitySensorNodes.x3d

- Select (click and hold) TouchSensor Cone to alternate Background nodes
- Select and drag PlaneSensor Box around the screen
- Select and rotate CylinderSensor Cylinder
- Select and spin SphereSensor Sphere

Keyboard inputs are also activated

- KeySensor indicates keyPress
- StringSensor shows *finalText* once <Enter> pressed
- Console shows *enteredText* (includes deletes if any)



Sensor node examples

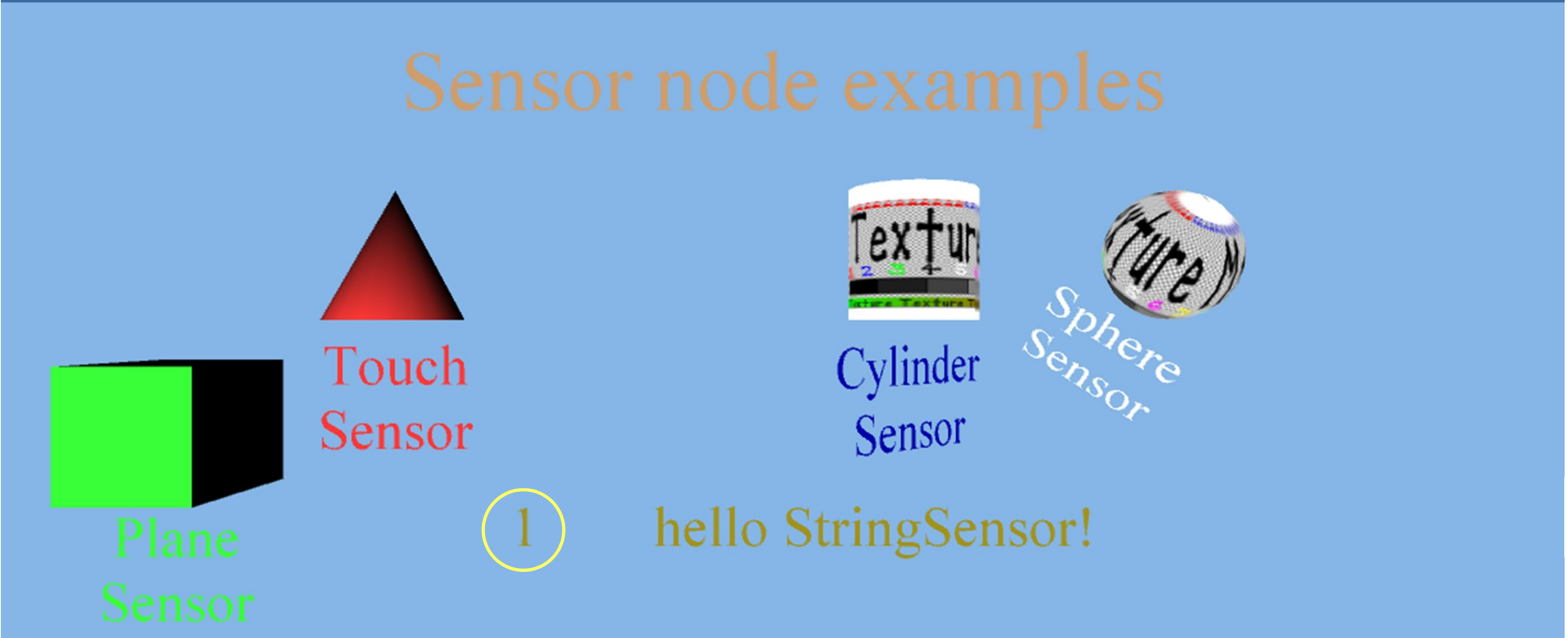
Touch Sensor

Plane Sensor

Cylinder Sensor

Sphere Sensor

? Press keys then <Enter>



Sensor node examples

Touch Sensor

Cylinder Sensor

Sphere Sensor

Plane Sensor

1 hello StringSensor!

Tutorial Summary

Tutorial Summary

X3D scene graph has a tremendous amount of capability and flexibility

X3D playback is suitable for

- Real-time rendering of 3D models
- Efficient animation using ROUTE-based event passing for any scene-graph parameter
- Reacting to user behaviors, overt and implicit

X3D authoring is straightforward

- Tools help, XML interoperability helps more
- Web deployment opens up new horizons for 3D

Exercise: deploy a 3D model

Deploy a 3D model using X3D, HTML on the Web

- Use existing model from another tool (e.g. Blender)
- Save as in XML as .x3d file (or #VRML 2.0, 3.0)
- Load (or import) into X3D-Edit, fix bugs (if any)
- Add meta tags in header documenting the scene
- Create parent scene that loads first via Inline
- Add further X3D content to parent scene
- Create HTML page containing the X3D scene that adds further information to user
- Deploy on a web site or as .zip archive to users

Review topics

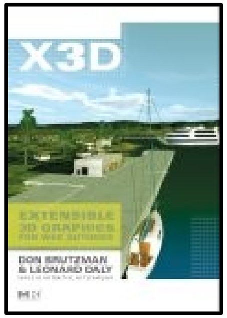
- Create a proper scene graph structure for a given scene
- List content and functionality that can be embedded in a scene graph
- State the contents of internal nodes and leaf nodes
- Visualize on paper the scene contained in a scene graph
- Explain the various scene-graph traversals, their order and purpose
- Translate between between scene graph and OpenGL with respect to modeling transformations, rendering attributes, geometry, animations
- Explain the connection between the matrix stack and a scene graph
- Name the advantages of using a scene graph over OpenGL
- Explain the relationship between scene graphs and raytracing
- Explain why and how bounding volumes are used in scene graphs
- Name performance optimizations that a scene graph affords
- Use X3D as a concrete scene graph architecture
- Use a graphical scene graph editor to create and modify graphs
- Use a text editor to modify graphs
- Conceptually explain the relationship between a scene graph data file, a scene graph viewer, a scene graph editor, a geometry data file and an OpenGL executable

back to Table of Contents

References

References 1

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



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

X3D Resources

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

References 2

X3D-Edit Authoring Tool

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

X3D Scene Authoring Hints

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

X3D Graphics Specification

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

Contact

Don Brutzman

brutzman@nps.edu

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

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.2149 voice
1.831.656.7599 fax

CGEMS, SIGGRAPH, Eurographics

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

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

X3D for Web Authors recognized by CGEMS! ☺

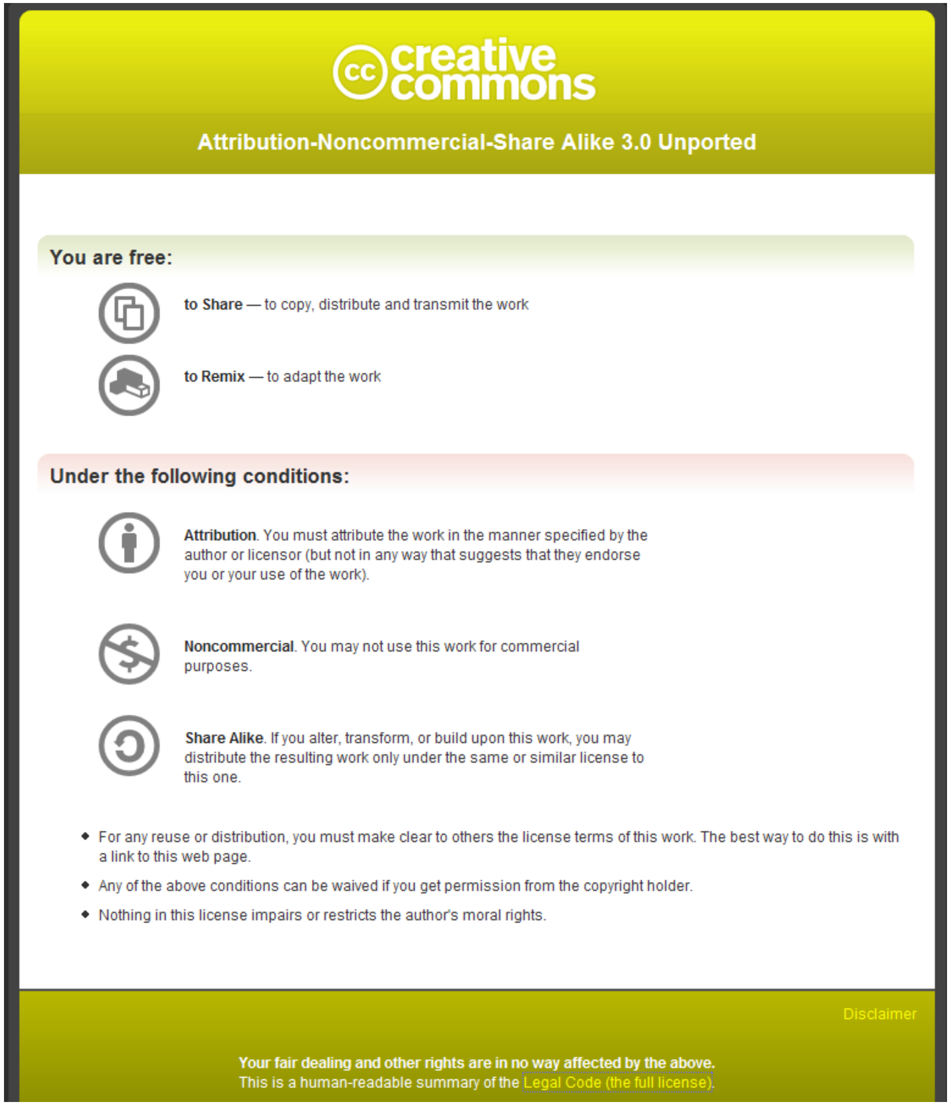
- Book materials: X3D-Edit tool, examples, slidesets
- Received jury award for Best Submission 2008

CGEMS supported by SIGGRAPH, Eurographics

cgems ACMSIGGRAPH EG

Creative Commons open-source license

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



creative commons

Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

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

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

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

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

Disclaimer

Your fair dealing and other rights are in no way affected by the above.
This is a human-readable summary of the Legal Code (the full license).

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

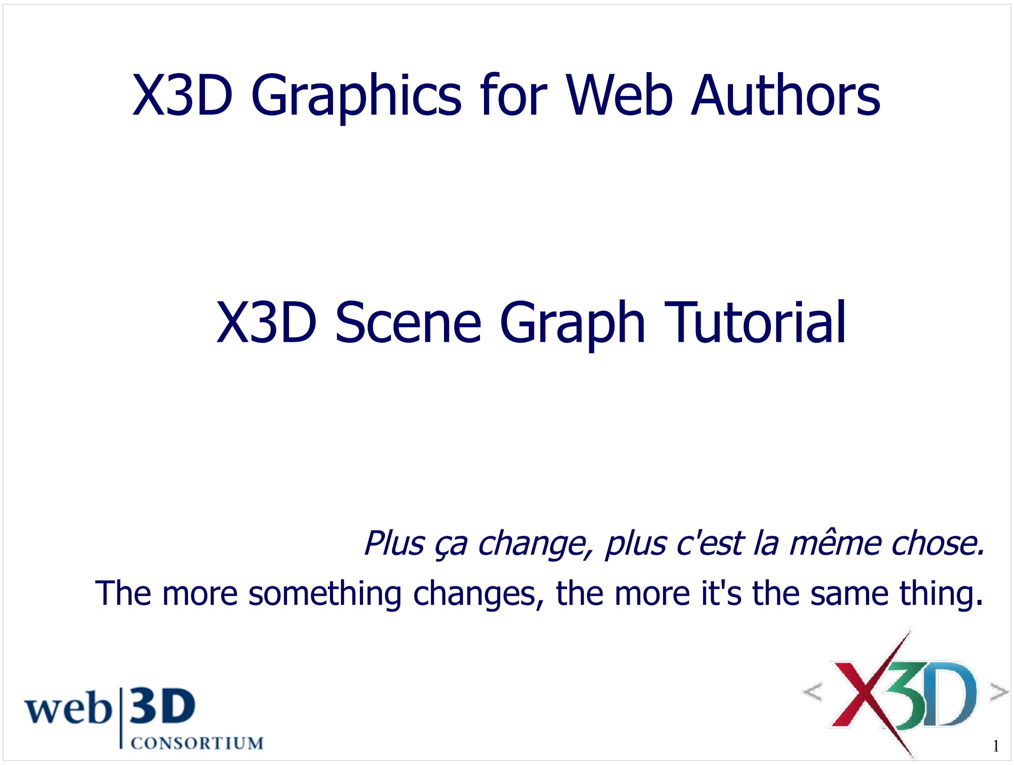
http://www.web3d.org/x3d/content/examples/license.html

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

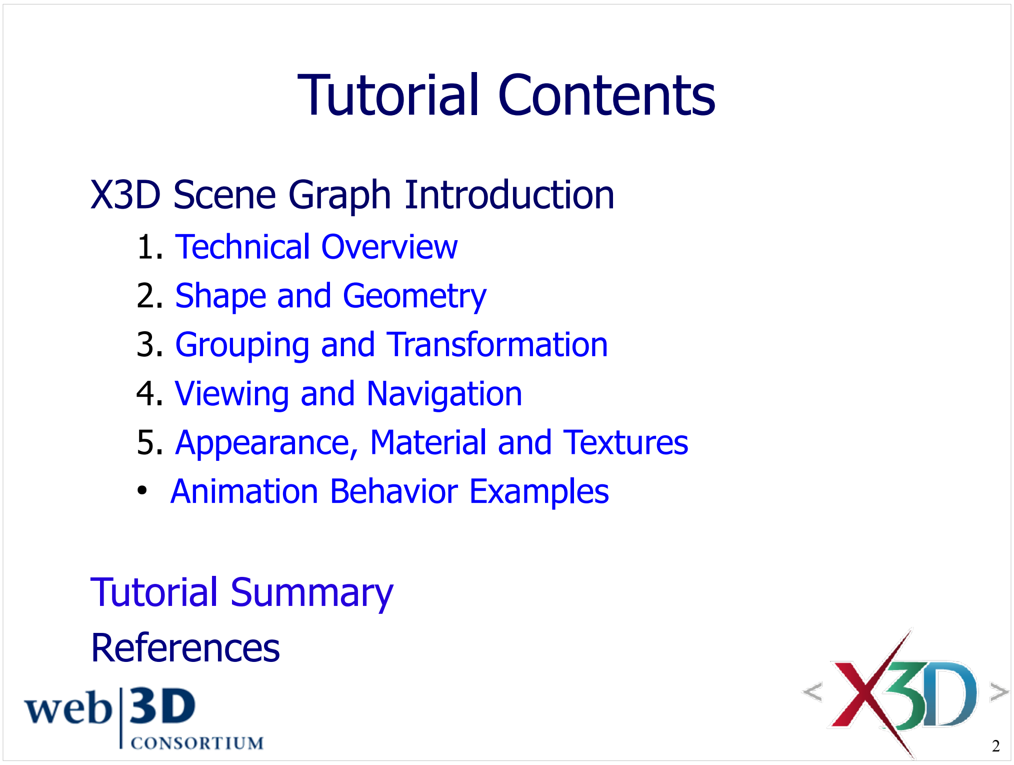
X3D Scene Graph Tutorial

Plus ça change, plus c'est la même chose.

The more something changes, the more it's the same thing.

web|3D CONSORTIUM

<X3D>



Tutorial Contents

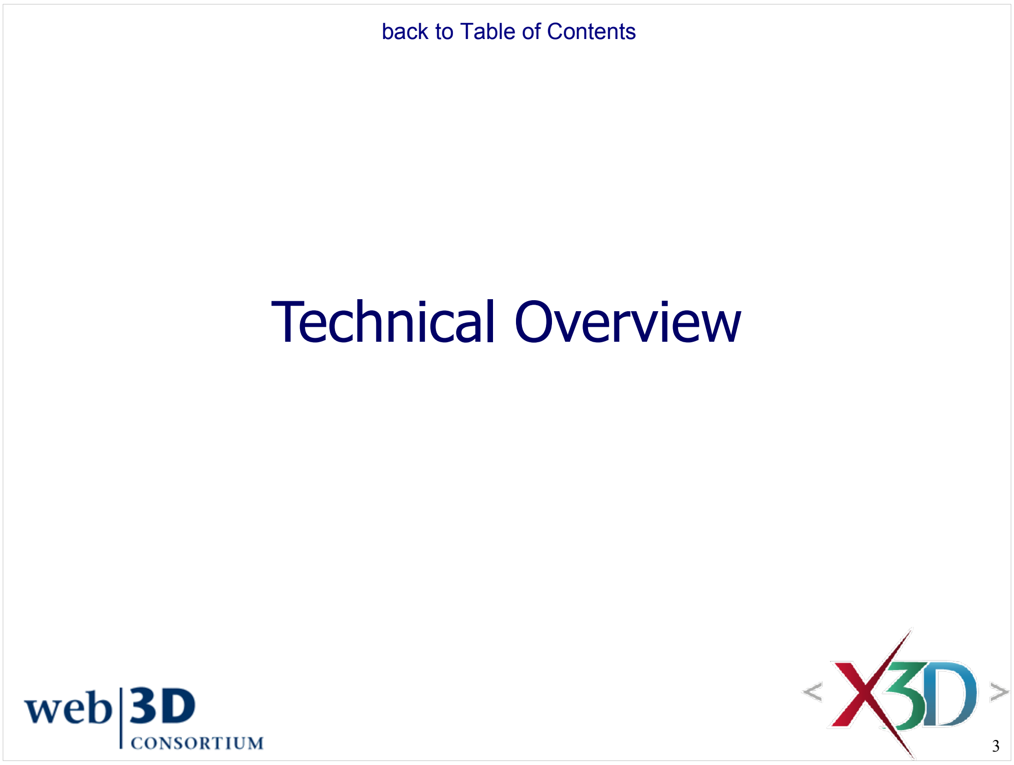
X3D Scene Graph Introduction

1. Technical Overview
2. Shape and Geometry
3. Grouping and Transformation
4. Viewing and Navigation
5. Appearance, Material and Textures

- Animation Behavior Examples

Tutorial Summary

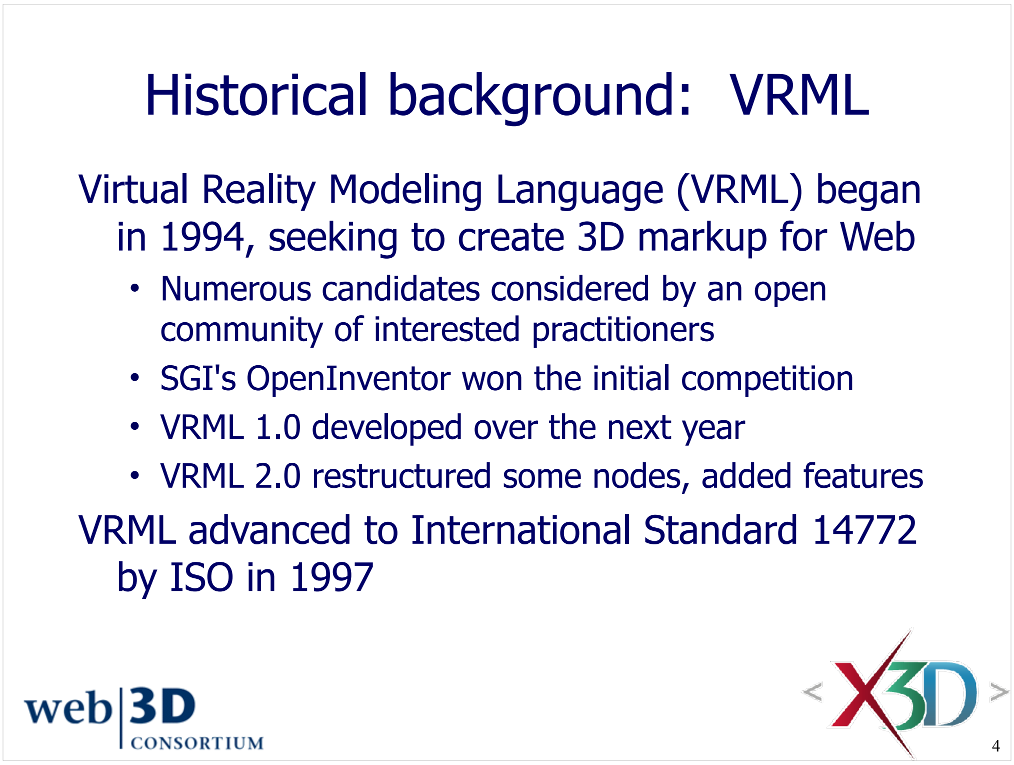
References



back to Table of Contents

Technical Overview

3



Historical background: VRML

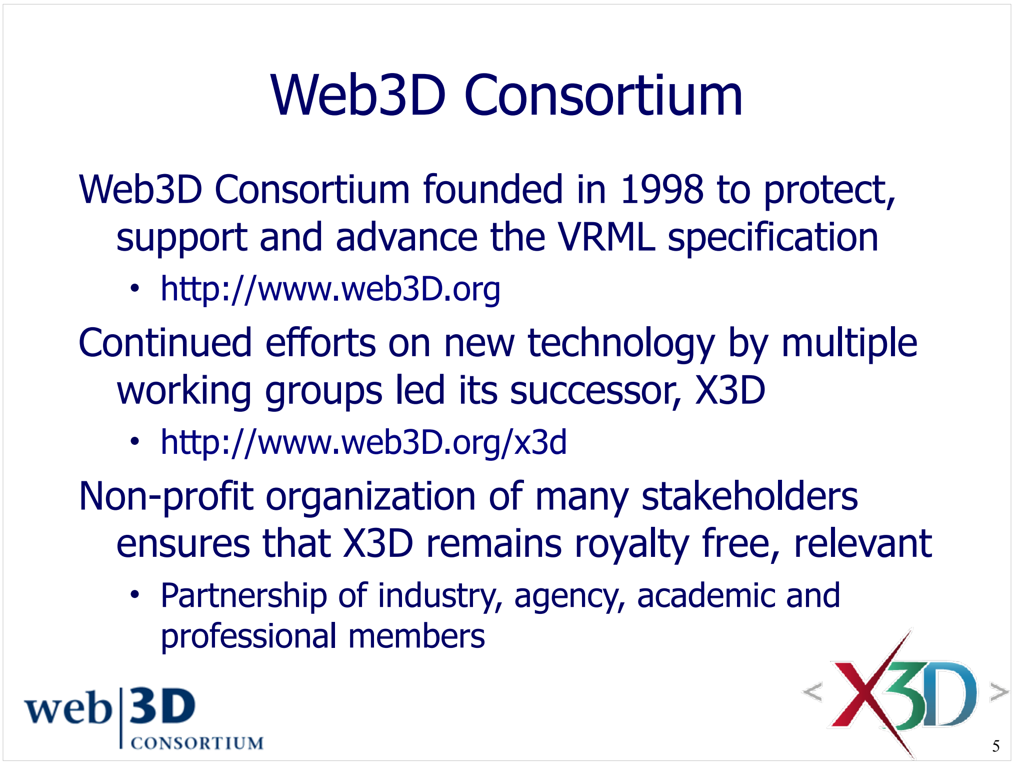
Virtual Reality Modeling Language (VRML) began in 1994, seeking to create 3D markup for Web

- Numerous candidates considered by an open community of interested practitioners
- SGI's OpenInventor won the initial competition
- VRML 1.0 developed over the next year
- VRML 2.0 restructured some nodes, added features

VRML advanced to International Standard 14772 by ISO in 1997

4

Lots more can be said here. Indeed numerous books have been written about VRML.



Web3D Consortium

Web3D Consortium founded in 1998 to protect, support and advance the VRML specification

- http://www.web3D.org

Continued efforts on new technology by multiple working groups led its successor, X3D

- http://www.web3D.org/x3d

Non-profit organization of many stakeholders ensures that X3D remains royalty free, relevant

- Partnership of industry, agency, academic and professional members

Perhaps the key test of 'openness' for any self-proclaimed 'open' organization: exactly who is allowed to join? Many industry associations only allow preselected (usually paying) companies to participate.

The Web3D Consortium includes industry, government-agency, college/university and individual professional memberships. This makes it one of the most open organizations around.

Further information on membership and joining available online at http://www.web3d.org/membership

web3D CONSORTIUM

Open Standards for Real-Time 3D Communication

HOME NEWS & EVENTS ABOUT WEB3D JOIN WIKI SPECIFICATIONS MEMBER LOGIN

Case Studies Great Projects by Our Members

X3D & VRML The Most Widely Used Formats

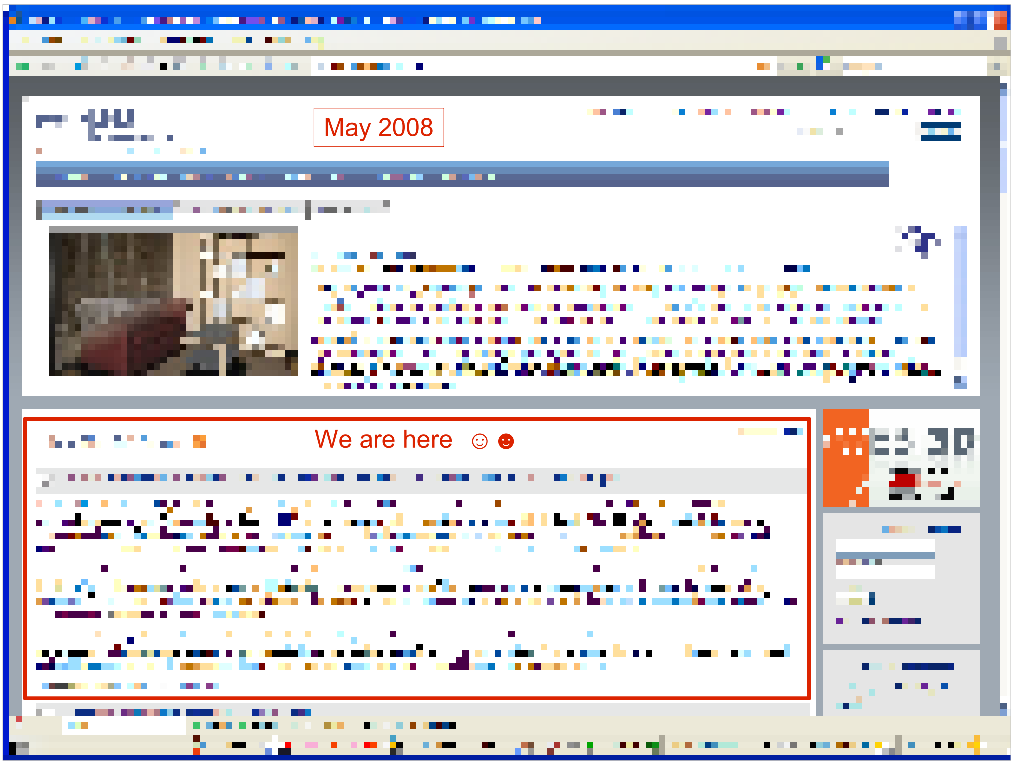
3D in HTML X3DOM... 3D Without Plugins

X3D Earth Open Earth Globe Format

December 2008

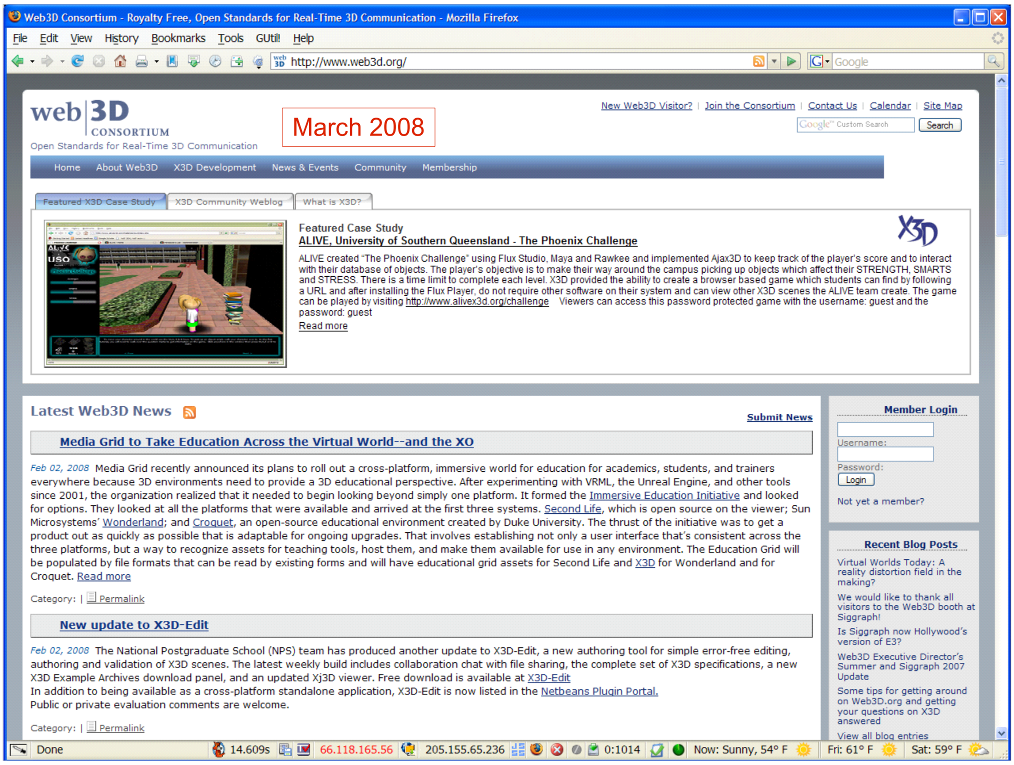
http://www.web3d.org January 2008

August 2008



http://www.web3d.org May 2008

http://www.lsi.usp.br/forumx3d


Web3D Consortium - Royalty Free, Open Standards for Real-Time 3D Communication - Mozilla Firefox
File Edit View History Bookmarks Tools GUtil! Help
http://www.web3d.org/
web 3D CONSORTIUM
Open Standards for Real-Time 3D Communication
New Web3D Visitor? | Join the Consortium | Contact Us | Calendar | Site Map
Google Custom Search
Search
Home About Web3D X3D Development News & Events Community Membership
Featured X3D Case Study
X3D Community Weblog
What is X3D?
X3D
Member Login
Username:
Password:
Login
Not yet a member?
Done
14.609s
66.118.165.56
205.155.65.236
0:1014
Now: Sunny, 54° F
Fri: 61° F
Sat: 59° F

March 2008

Featured Case Study

ALIVE, University of Southern Queensland - The Phoenix Challenge

ALIVE created "The Phoenix Challenge" using Flux Studio, Maya and Rawkee and implemented Ajax3D to keep track of the player's score and to interact with their database of objects. The player's objective is to make their way around the campus picking up objects which affect their STRENGTH, SMARTS and STRESS. There is a time limit to complete each level. X3D provided the ability to create a browser based game which students can find by following a URL and after installing the Flux Player, do not require other software on their system and can view other X3D scenes the ALIVE team create. The game can be played by visiting http://www.alivex3d.org/challenge Viewers can access this password protected game with the username: guest and the password: guest

Read more

Latest Web3D News

Submit News

Media Grid to Take Education Across the Virtual World--and the XO

Feb 02, 2008 Media Grid recently announced its plans to roll out a cross-platform, immersive world for education for academics, students, and trainers everywhere because 3D environments need to provide a 3D educational perspective. After experimenting with VRML, the Unreal Engine, and other tools since 2001, the organization realized that it needed to begin looking beyond simply one platform. It formed the Immersive Education Initiative and looked for options. They looked at all the platforms that were available and arrived at the first three systems. Second Life, which is open source on the viewer; Sun Microsystems' Wonderland; and Croquet, an open-source educational environment created by Duke University. The thrust of the initiative was to get a product out as quickly as possible that is adaptable for ongoing upgrades. That involves establishing not only a user interface that's consistent across the three platforms, but a way to recognize assets for teaching tools, host them, and make them available for use in any environment. The Education Grid will be populated by file formats that can be read by existing forms and will have educational grid assets for Second Life and X3D for Wonderland and for Croquet. Read more

Category: | Permalink

New update to X3D-Edit

Feb 02, 2008 The National Postgraduate School (NPS) team has produced another update to X3D-Edit, a new authoring tool for simple error-free editing, authoring and validation of X3D scenes. The latest weekly build includes collaboration chat with file sharing, the complete set of X3D specifications, a new X3D Example Archives download panel, and an updated Xj3D viewer. Free download is available at X3D-Edit
In addition to being available as a cross-platform standalone application, X3D-Edit is now listed in the Netbeans Plugin Portal.
Public or private evaluation comments are welcome.

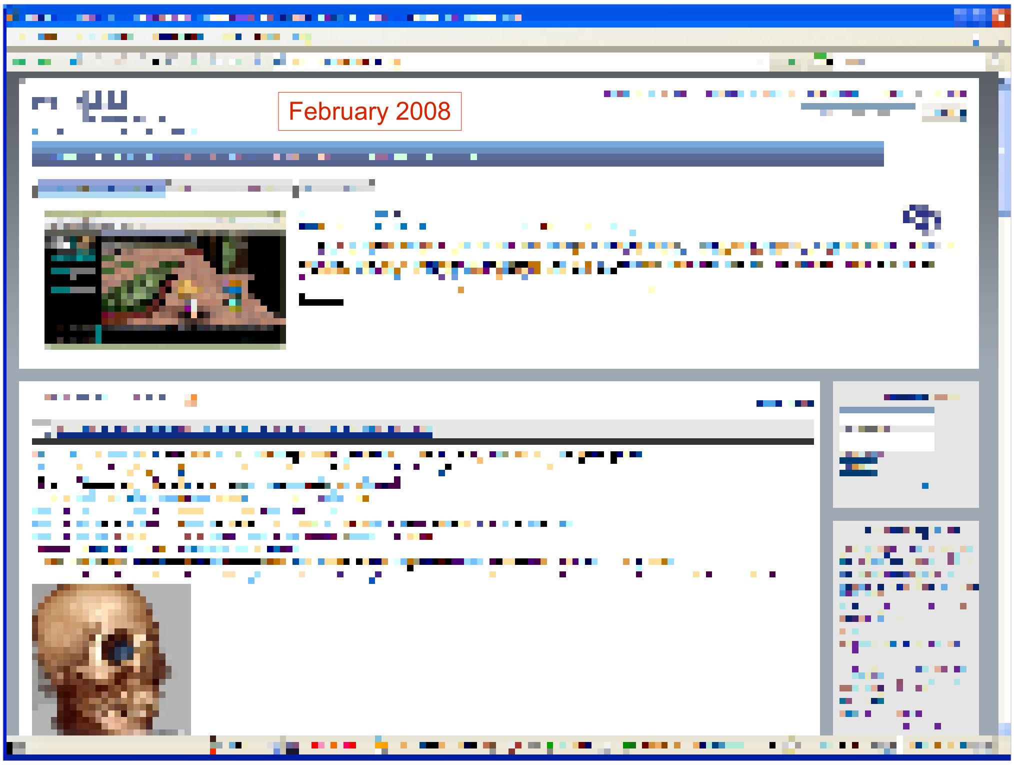
Category: | Permalink

Recent Blog Posts

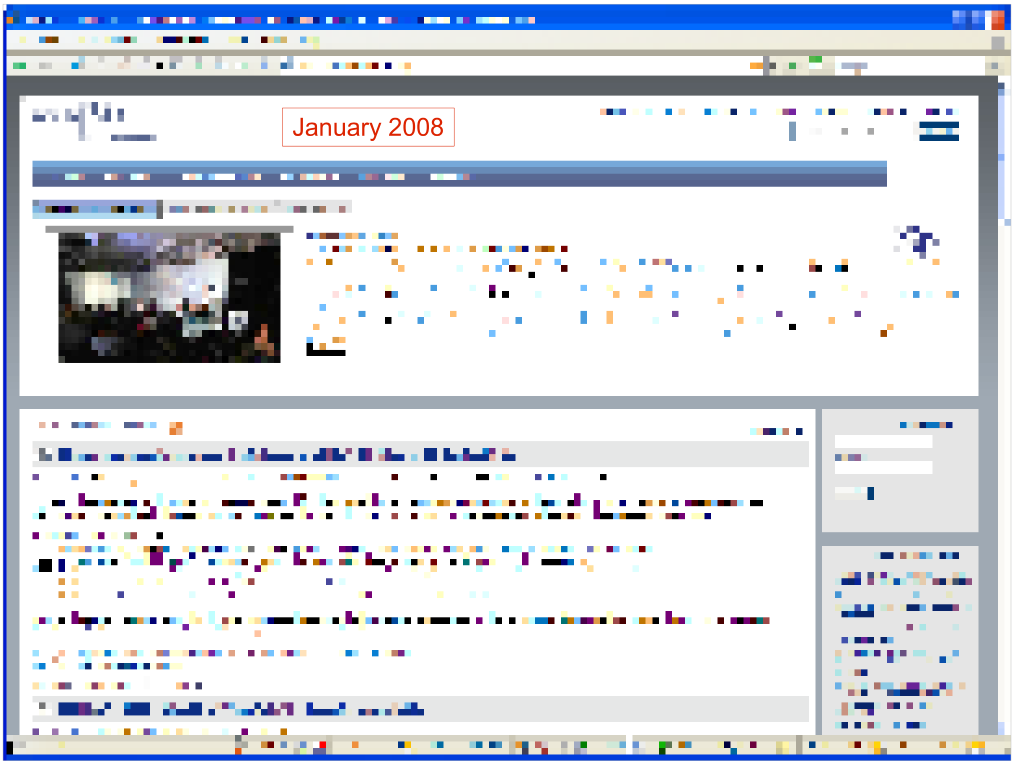
- Virtual Worlds Today: A reality distortion field in the making?
- We would like to thank all visitors to the Web3D booth at Siggraph!
- Is Siggraph now Hollywood's version of E3?
- Web3D Executive Director's Summer and Siggraph 2007 Update
- Some tips for getting around on Web3D.org and getting your questions on X3D answered

View all blog entries

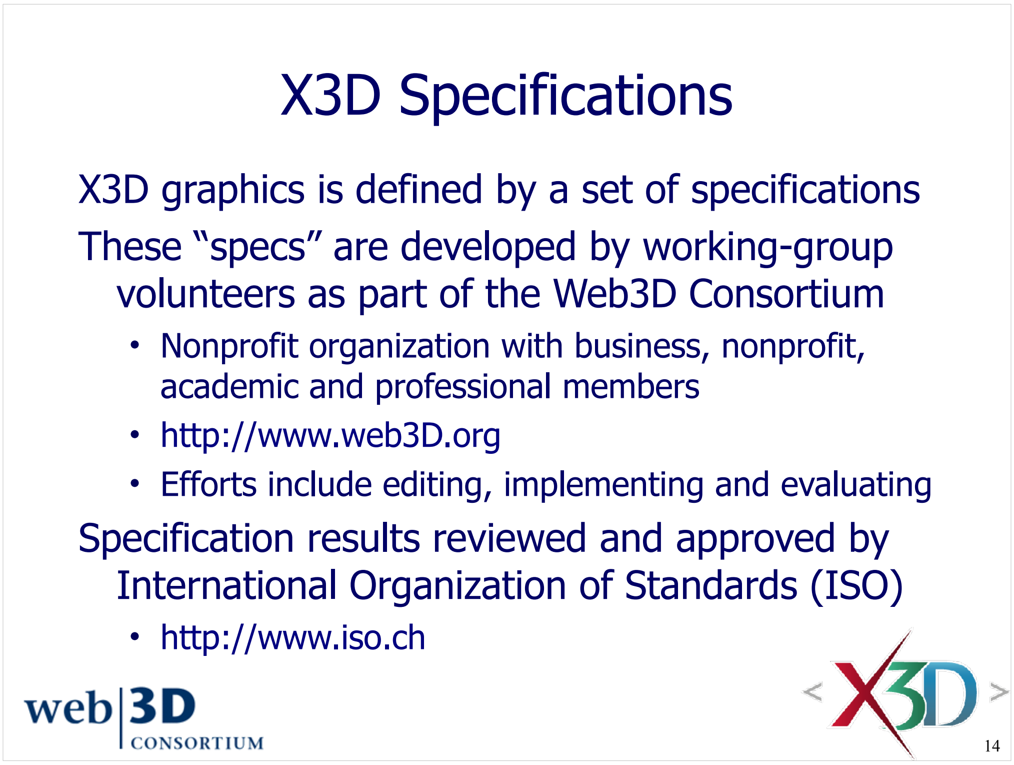
http://www.web3d.org February 2008



http://www.web3d.org January 2008



http://www.web3d.org December 2007



X3D Specifications

X3D graphics is defined by a set of specifications

These “specs” are developed by working-group volunteers as part of the Web3D Consortium

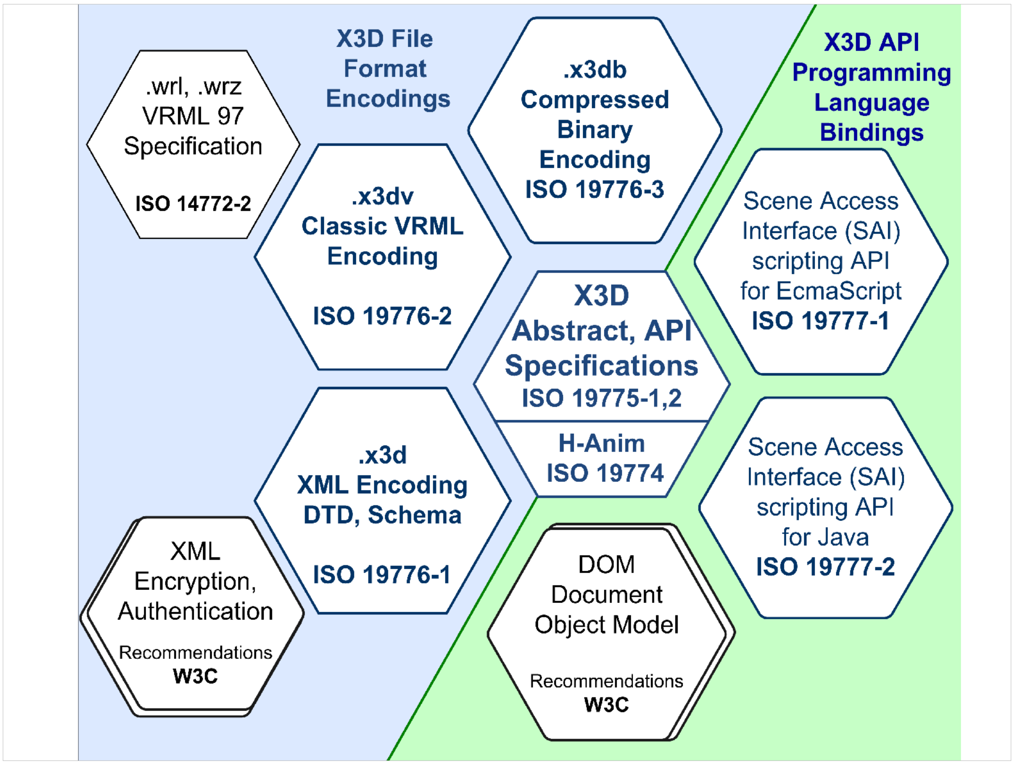
- Nonprofit organization with business, nonprofit, academic and professional members
- http://www.web3D.org
- Efforts include editing, implementing and evaluating

Specification results reviewed and approved by International Organization of Standards (ISO)

- http://www.iso.ch

14

Typically 10-15 member nations review and vote on the X3D Specification

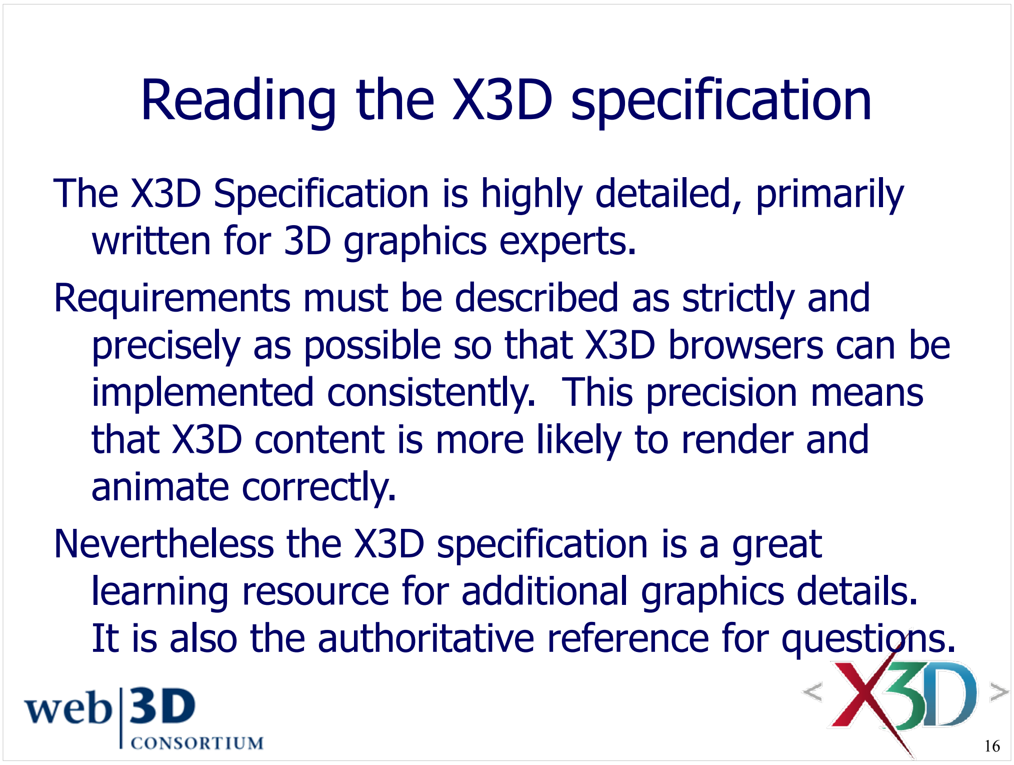

X3D File Format Encodings
.wrl, .wrz VRML 97 Specification ISO 14772-2
.x3db Compressed Binary Encoding ISO 19776-3
X3D API Programming Language Bindings
.x3dv Classic VRML Encoding ISO 19776-2
Scene Access Interface (SAI) scripting API for EcmaScript ISO 19777-1
X3D Abstract, API Specifications ISO 19775-1,2
H-Anim ISO 19774
.x3d XML Encoding DTD, Schema ISO 19776-1
Scene Access Interface (SAI) scripting API for Java ISO 19777-2
XML Encryption, Authentication Recommendations W3C
DOM Document Object Model Recommendations W3C

Encodings define file formats.

Each Scene Access Interface (SAI) binding is a specific Application Programming Interface (API).

ECMAScript is the formal-specification name for JavaScript.

ECMA was originally named the European Computer Manufacturers Association and is now ECMA International - European association for standardizing information and communication systems. http://www.ecma-international.org

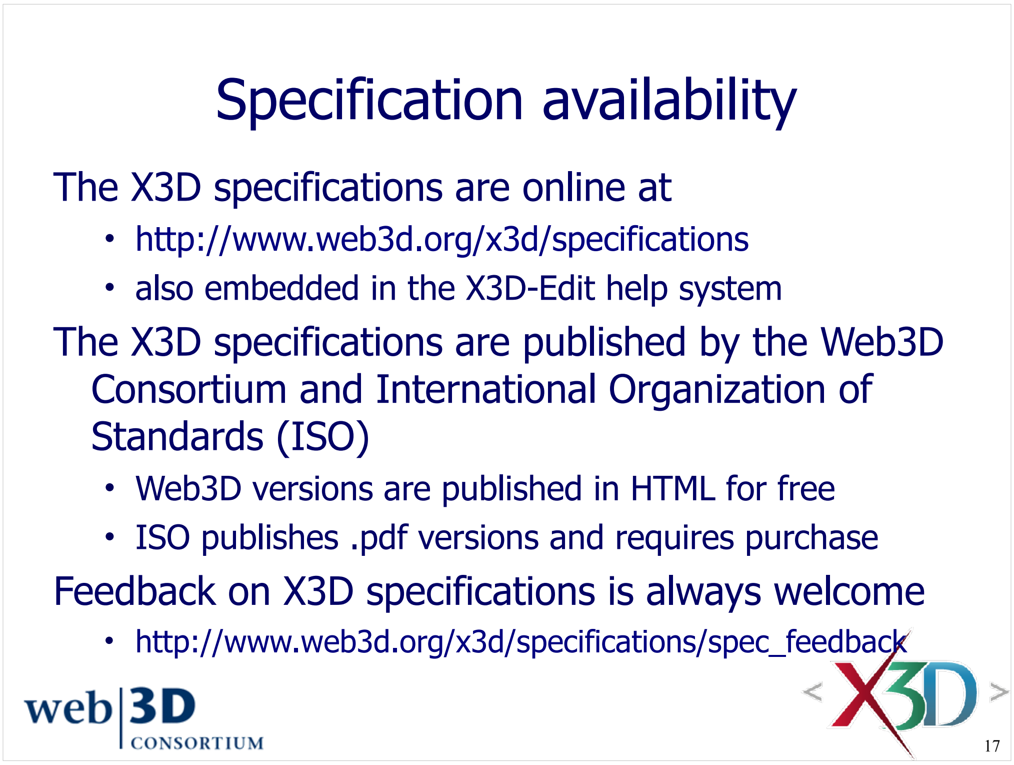


Reading the X3D specification

The X3D Specification is highly detailed, primarily written for 3D graphics experts.

Requirements must be described as strictly and precisely as possible so that X3D browsers can be implemented consistently. This precision means that X3D content is more likely to render and animate correctly.

Nevertheless the X3D specification is a great learning resource for additional graphics details. It is also the authoritative reference for questions.



Specification availability

The X3D specifications are online at

- http://www.web3d.org/x3d/specifications
- also embedded in the X3D-Edit help system

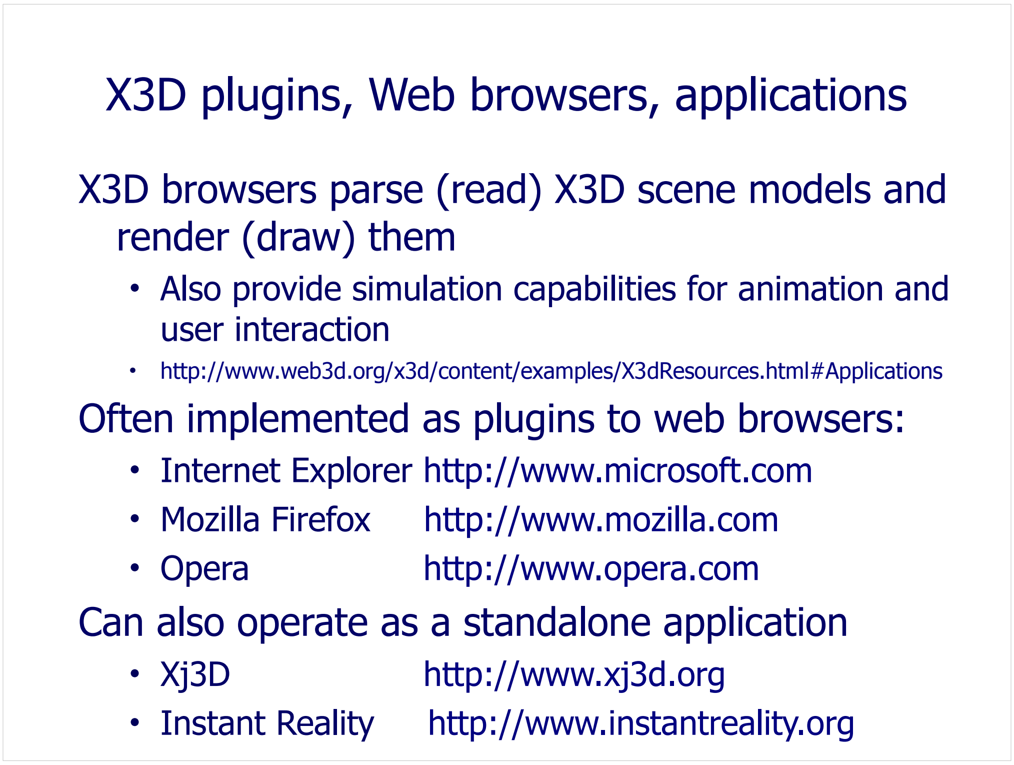
The X3D specifications are published by the Web3D Consortium and International Organization of Standards (ISO)

- Web3D versions are published in HTML for free
- ISO publishes .pdf versions and requires purchase

Feedback on X3D specifications is always welcome

- http://www.web3d.org/x3d/specifications/spec_feedback

The Web3D Consortium was the first organization to request (and receive) permission to place final versions of approved ISO specifications online for free retrieval using HTML. Purchase of hard-copy bound and electronic versions from ISO remains available.



X3D plugins, Web browsers, applications

X3D browsers parse (read) X3D scene models and render (draw) them

- Also provide simulation capabilities for animation and user interaction
- http://www.web3d.org/x3d/content/examples/X3dResources.html#Applications

Often implemented as plugins to web browsers:

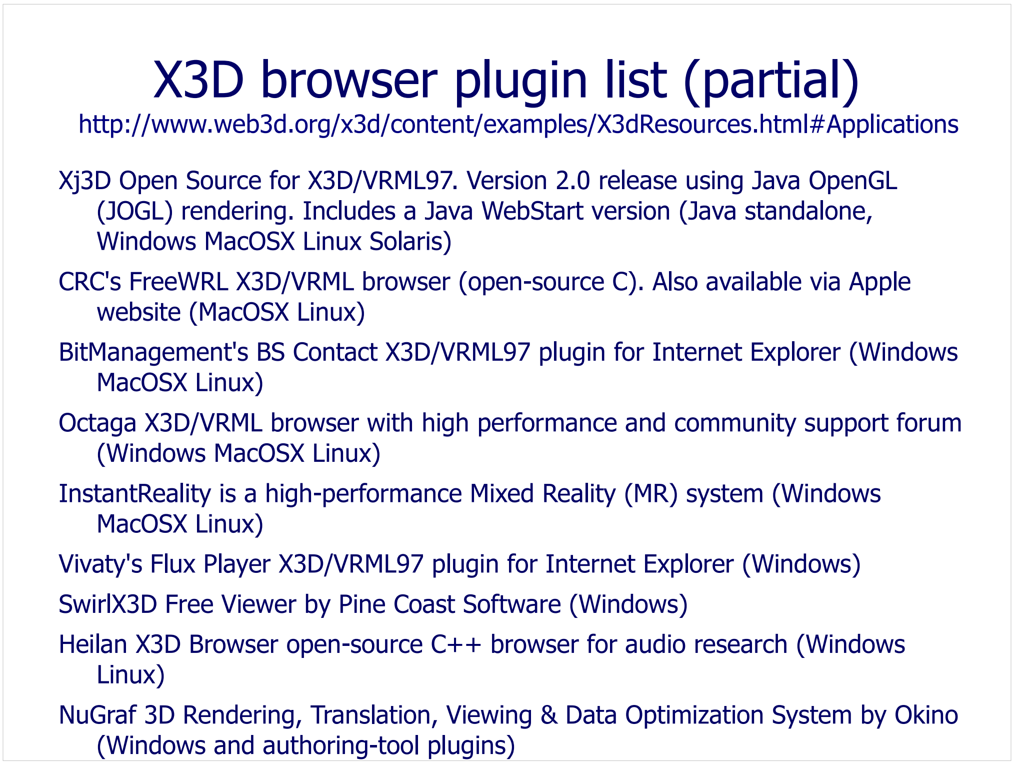
- Internet Explorer http://www.microsoft.com
- Mozilla Firefox http://www.mozilla.com
- Opera http://www.opera.com

Can also operate as a standalone application

- Xj3D http://www.xj3d.org
- Instant Reality http://www.instantreality.org

It is a good idea to install an X3D plugin in your web browser. Available via

http://www.web3d.org/x3d/content/examples/help.html#Applications



X3D browser plugin list (partial)

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

Xj3D Open Source for X3D/VRML97. Version 2.0 release using Java OpenGL (JOGL) rendering. Includes a Java WebStart version (Java standalone, Windows MacOSX Linux Solaris)

CRC's FreeWRL X3D/VRML browser (open-source C). Also available via Apple website (MacOSX Linux)

BitManagement's BS Contact X3D/VRML97 plugin for Internet Explorer (Windows MacOSX Linux)

Octaga X3D/VRML browser with high performance and community support forum (Windows MacOSX Linux)

InstantReality is a high-performance Mixed Reality (MR) system (Windows MacOSX Linux)

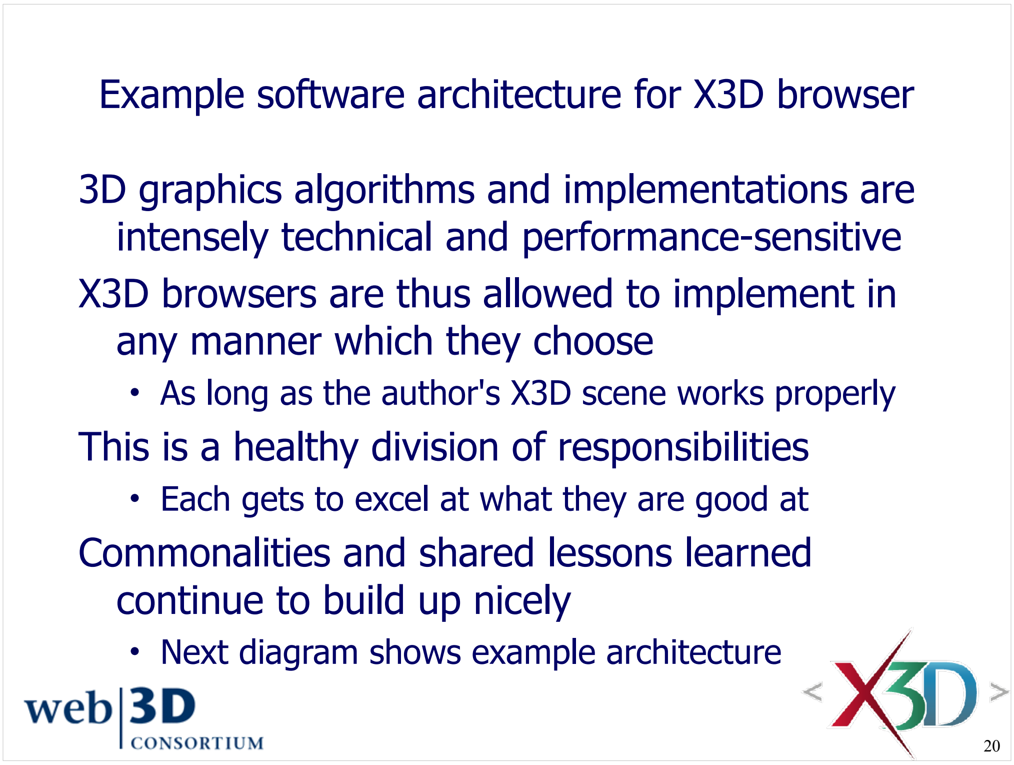
Vivaty's Flux Player X3D/VRML97 plugin for Internet Explorer (Windows)

SwirlX3D Free Viewer by Pine Coast Software (Windows)

Heilan X3D Browser open-source C++ browser for audio research (Windows Linux)

NuGraf 3D Rendering, Translation, Viewing & Data Optimization System by Okino (Windows and authoring-tool plugins)

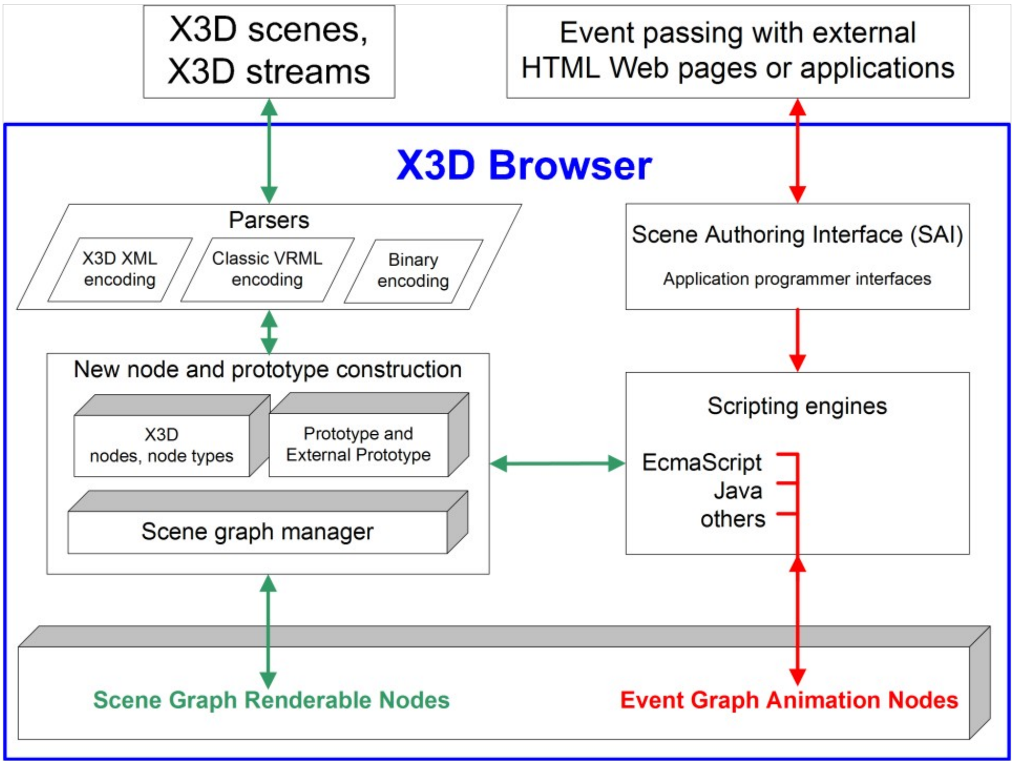
Browser support for the various X3D components is now available at
http://www.web3d.org/x3d/wiki/index.php/Player_support_for_X3D_components



Example software architecture for X3D browser

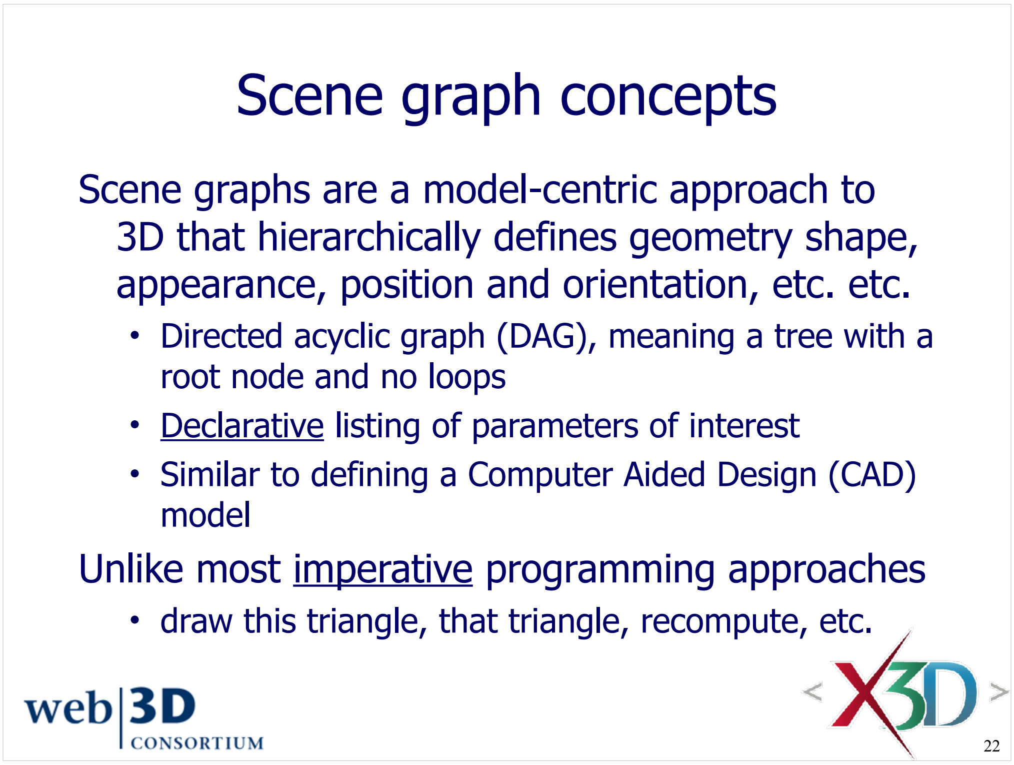
- 3D graphics algorithms and implementations are intensely technical and performance-sensitive
- X3D browsers are thus allowed to implement in any manner which they choose
  - As long as the author's X3D scene works properly
- This is a healthy division of responsibilities
  - Each gets to excel at what they are good at
- Commonalities and shared lessons learned continue to build up nicely
  - Next diagram shows example architecture

20


X3D scenes, X3D streams
Event passing with external HTML Web pages or applications
X3D Browser
Parsers
X3D XML encoding
Classic VRML encoding
Binary encoding
Scene Authoring Interface (SAI)
Application programmer interfaces
New node and prototype construction
X3D nodes, node types
Prototype and External Prototype
Scene graph manager
Scripting engines
EcmaScript
Java
others
Scene Graph Renderable Nodes
Event Graph Animation Nodes

X3D browser implementers can use any approach they choose. This architecture diagram is generic to illustrate common approaches.

Part of the magic for X3D scene authors is that they don't have to care about underlying hard-core technical details “under the hood” of each browser. Rather, scenes are designed to capture shapes, appearance and behaviors from a content-authoring perspective that emphasizes modeling results.



Scene graph concepts

Scene graphs are a model-centric approach to 3D that hierarchically defines geometry shape, appearance, position and orientation, etc. etc.

- Directed acyclic graph (DAG), meaning a tree with a root node and no loops
- Declarative listing of parameters of interest
- Similar to defining a Computer Aided Design (CAD) model

Unlike most imperative programming approaches

- draw this triangle, that triangle, recompute, etc.

Scene graph terminology

Scene graph data file

- contains model description, may refer to data files

Scene graph viewer

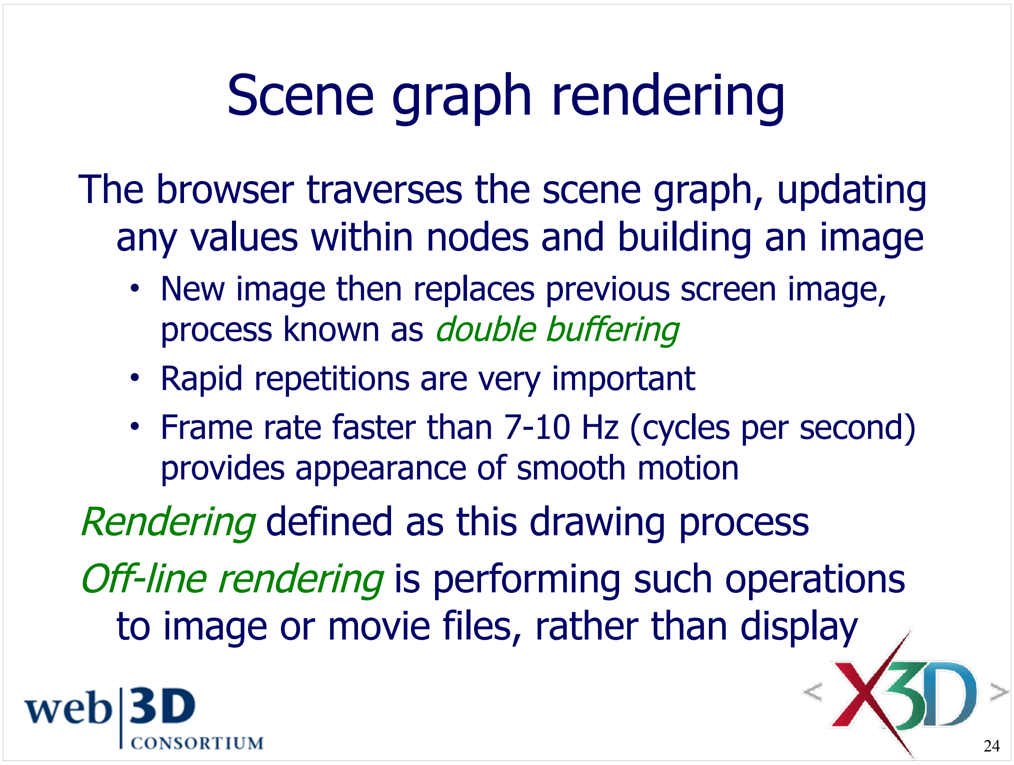
- Reads and renders scene-graph models
- Implemented as application or web browser plugin

Scene graph editor

- Special text editor for scene graph development

Executable application

- Specific 3D model capable of running on a specific operating system



Scene graph rendering

The browser traverses the scene graph, updating any values within nodes and building an image

- New image then replaces previous screen image, process known as *double buffering*
- Rapid repetitions are very important
- Frame rate faster than 7-10 Hz (cycles per second) provides appearance of smooth motion

Rendering defined as this drawing process

Off-line rendering is performing such operations to image or movie files, rather than display

Performance optimizations

Scene graphs have performance optimizations sometimes not available in other Application Programming Interface (API) approaches

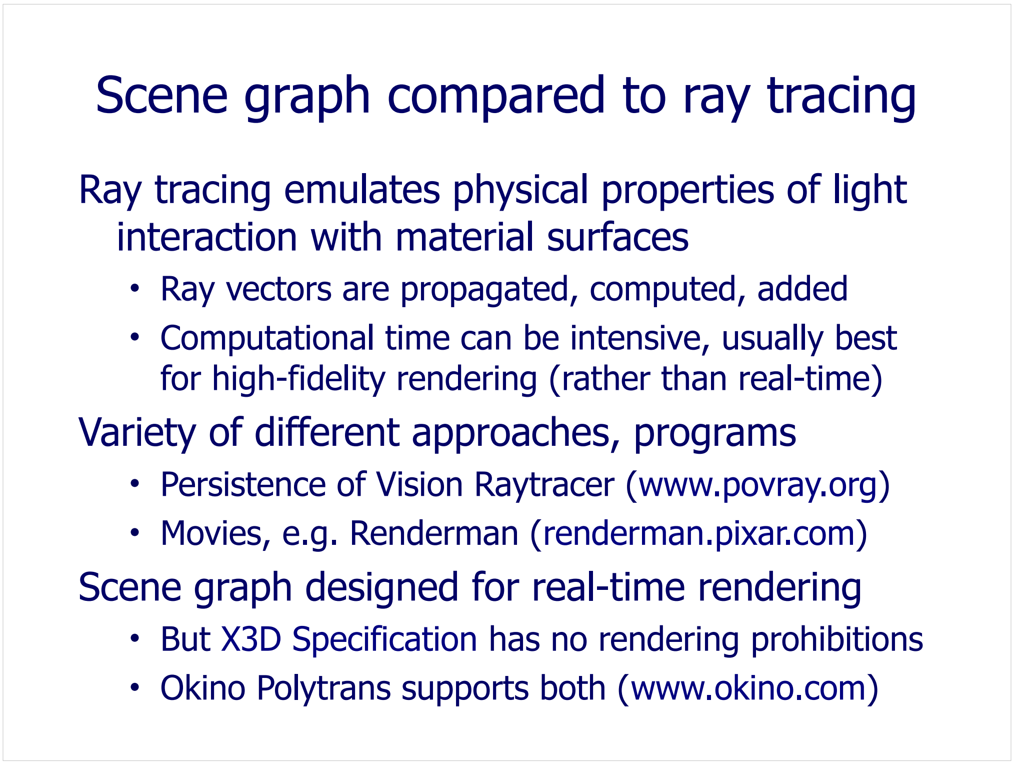
- Scene graph structure designed to take advantage of graphics hardware acceleration
- Can refer to (and reuse) subgraphs (X3D DEF, USE)
- “dirty bit” indicates whether a scene subgraph has been modified, avoiding needless recomputations
- Browser can rearrange or simplify geometry
- Scoping of lights to reduce computational impact
- Widely repeated interchange patterns

Scene-graph advantages relative to OpenGL, DirectX render layers

Scene graphs often a close match to simulation models, easier for authors to make and modify

OpenGL and DirectX APIs are thin software layers that expose underlying 3D graphics-acceleration hardware for real-time rendering

Each is a state machine, optimized for drawing triangles textures etc., not designed to have memory for modeling high-level simulation objects, remembering user actions, etc.



Scene graph compared to ray tracing

Ray tracing emulates physical properties of light interaction with material surfaces

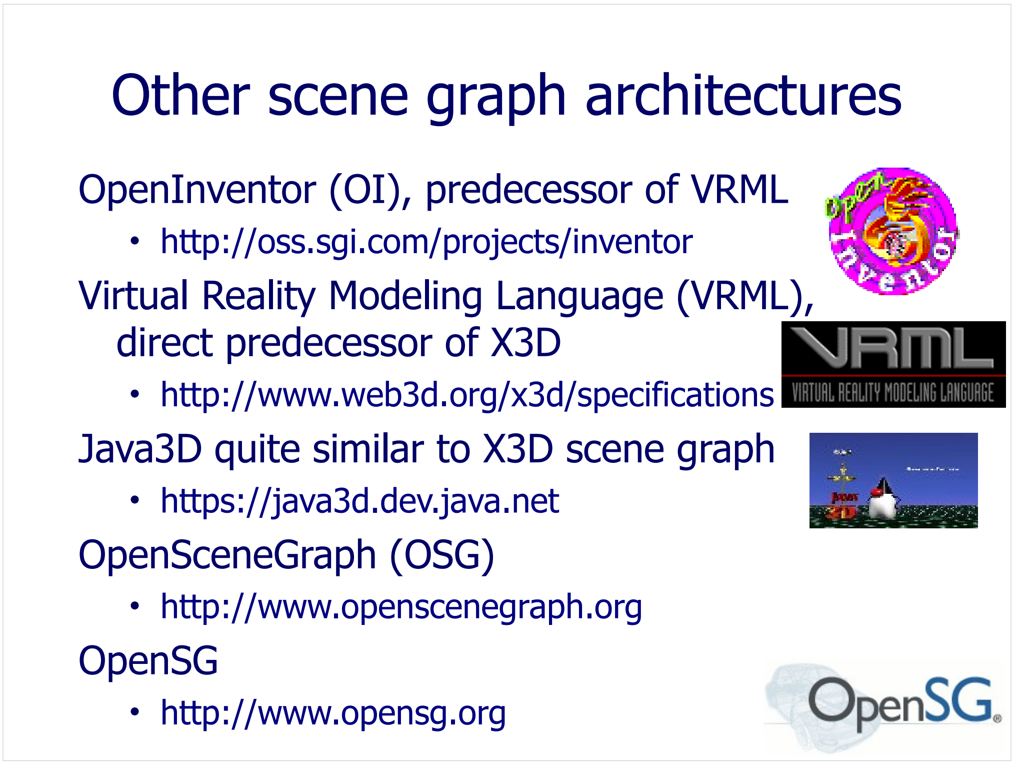
- Ray vectors are propagated, computed, added
- Computational time can be intensive, usually best for high-fidelity rendering (rather than real-time)

Variety of different approaches, programs

- Persistence of Vision Raytracer (www.povray.org)
- Movies, e.g. Renderman (renderman.pixar.com)

Scene graph designed for real-time rendering

- But X3D Specification has no rendering prohibitions
- Okino Polytrans supports both (www.okino.com)



Other scene graph architectures

OpenInventor (OI), predecessor of VRML

- http://oss.sgi.com/projects/inventor

Virtual Reality Modeling Language (VRML), direct predecessor of X3D

- http://www.web3d.org/x3d/specifications

Java3D quite similar to X3D scene graph

- https://java3d.dev.java.net

OpenSceneGraph (OSG)

- http://www.openscenegraph.org

OpenSG

- http://www.opensg.org

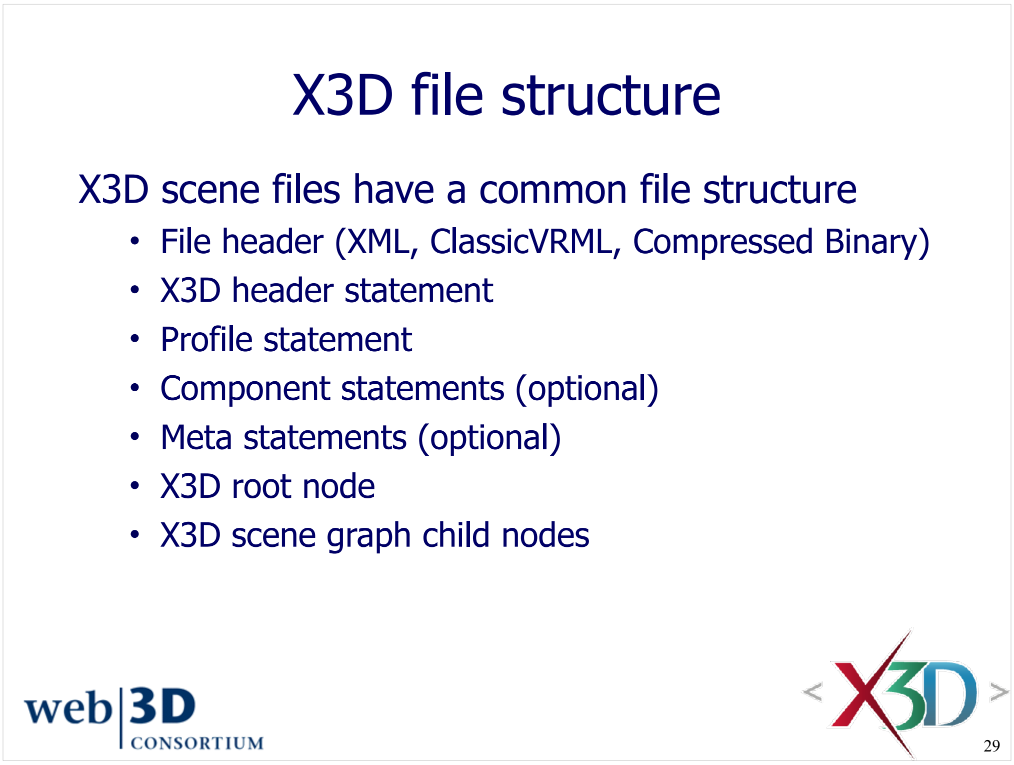
Open Inventor
VRML
VIRTUAL REALITY MODELING LANGUAGE
OpenSG

OpenInventor reference on WikiPedia: http://en.wikipedia.org/wiki/Open_Inventor

VRML97 is still an approved ISO specification. Furthermore the X3D ClassicVRML encoding is a direct extension of VRML 97, moving from version 2.0 to 3.0.

The Java3D scene graph has been described as over 90% similar to VRML and X3D. There are many good books and resources.

From the website: “The OpenSceneGraph is an open source high performance 3D graphics toolkit, used by application developers in fields such as visual simulation, games, virtual reality, scientific visualization and modelling.”

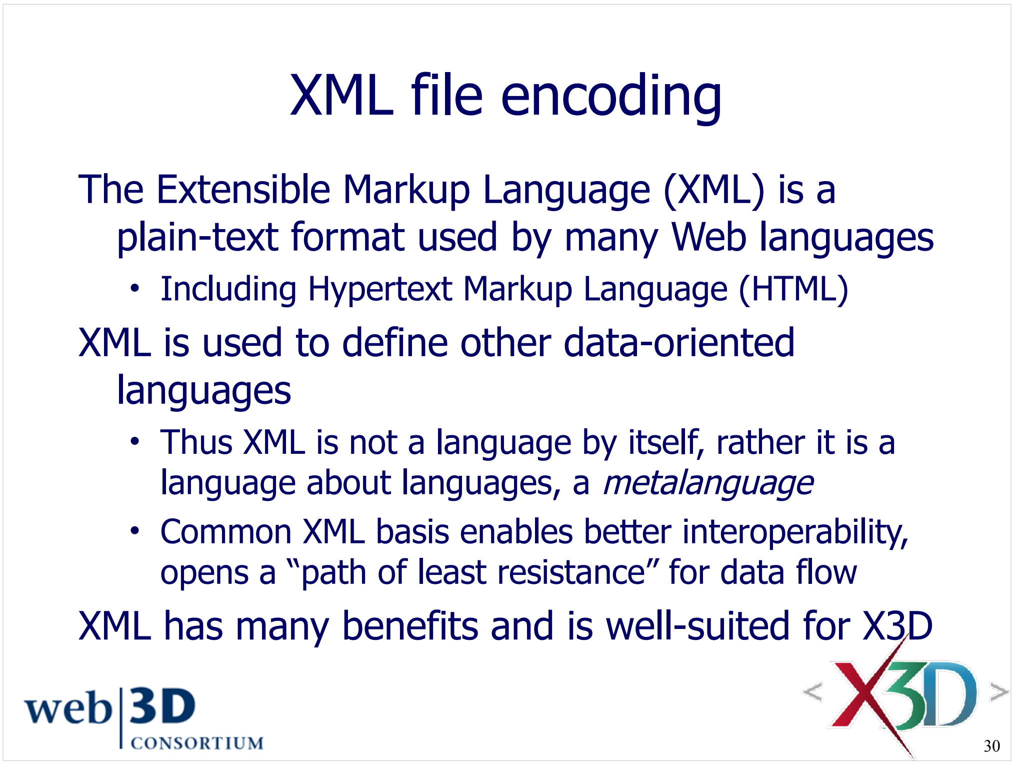


X3D file structure

X3D scene files have a common file structure

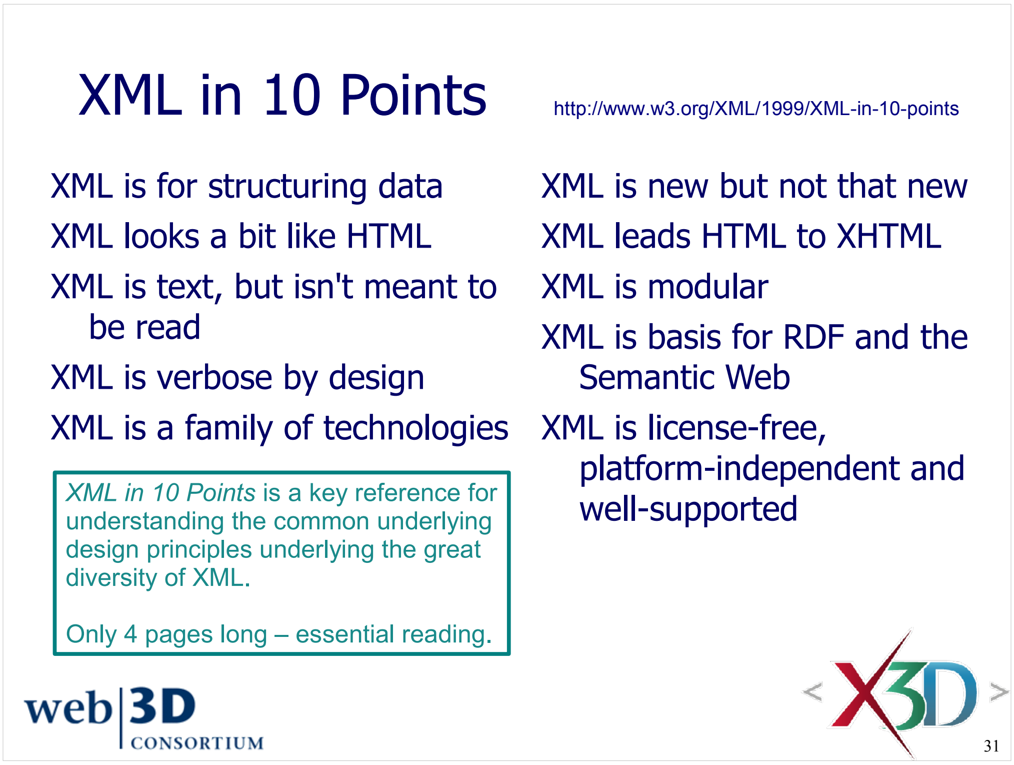
- File header (XML, ClassicVRML, Compressed Binary)
- X3D header statement
- Profile statement
- Component statements (optional)
- Meta statements (optional)
- X3D root node
- X3D scene graph child nodes

The X3D scene root node is implicit in ClassicVRML encoding and not listed per se.



XML file encoding

- The Extensible Markup Language (XML) is a plain-text format used by many Web languages
  - Including Hypertext Markup Language (HTML)
- XML is used to define other data-oriented languages
  - Thus XML is not a language by itself, rather it is a language about languages, a *metalanguage*
  - Common XML basis enables better interoperability, opens a “path of least resistance” for data flow
- XML has many benefits and is well-suited for X3D



XML in 10 Points

http://www.w3.org/XML/1999/XML-in-10-points

- XML is for structuring data
- XML looks a bit like HTML
- XML is text, but isn't meant to be read
- XML is verbose by design
- XML is a family of technologies

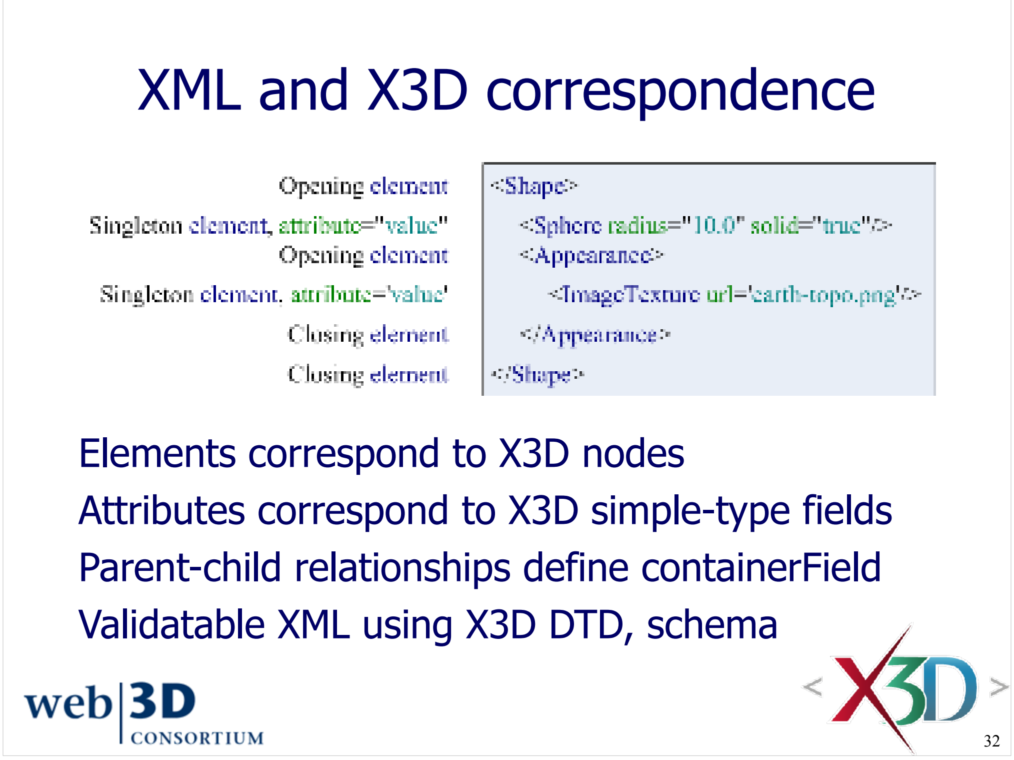
XML in 10 Points is a key reference for understanding the common underlying design principles underlying the great diversity of XML.

Only 4 pages long – essential reading.

- XML is new but not that new
- XML leads HTML to XHTML
- XML is modular
- XML is basis for RDF and the Semantic Web
- XML is license-free, platform-independent and well-supported

31

Bert Bos et al., “XML in 10 Points,: World Wide Web Consortium (W3C), created 1999, updated 2003. Available at http://www.w3.org/XML/1999/XML-in-10-points



XML and X3D correspondence

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

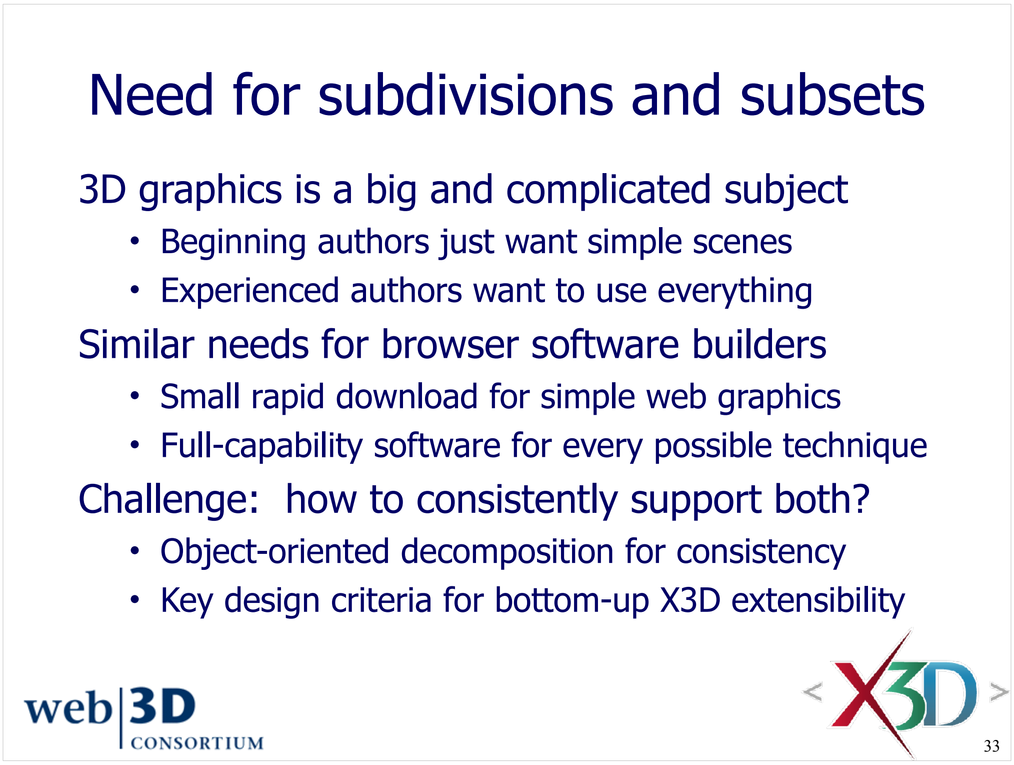
Elements correspond to X3D nodes

Attributes correspond to X3D simple-type fields

Parent-child relationships define containerField

Validatable XML using X3D DTD, schema

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



Need for subdivisions and subsets

3D graphics is a big and complicated subject

- Beginning authors just want simple scenes
- Experienced authors want to use everything

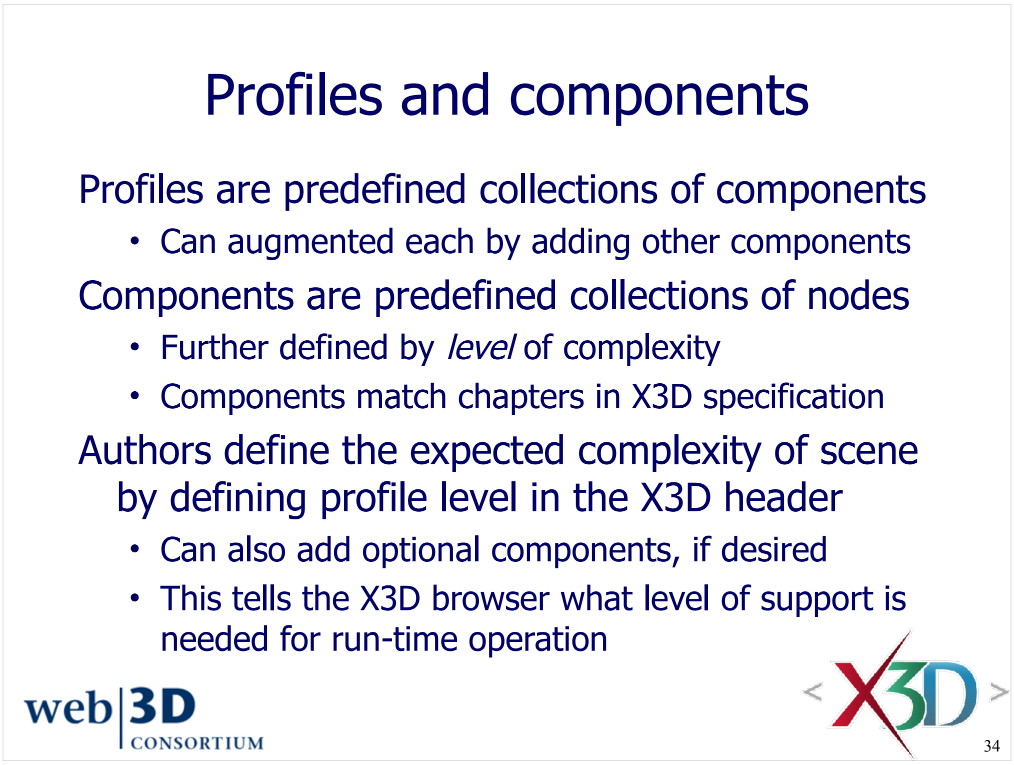
Similar needs for browser software builders

- Small rapid download for simple web graphics
- Full-capability software for every possible technique

Challenge: how to consistently support both?

- Object-oriented decomposition for consistency
- Key design criteria for bottom-up X3D extensibility

These points are some of the original design challenges that faced X3D architects when evolving from the successes and lessons learned of VRML97.



Profiles and components

Profiles are predefined collections of components

- Can augmented each by adding other components

Components are predefined collections of nodes

- Further defined by *level* of complexity
- Components match chapters in X3D specification

Authors define the expected complexity of scene by defining profile level in the X3D header

- Can also add optional components, if desired
- This tells the X3D browser what level of support is needed for run-time operation

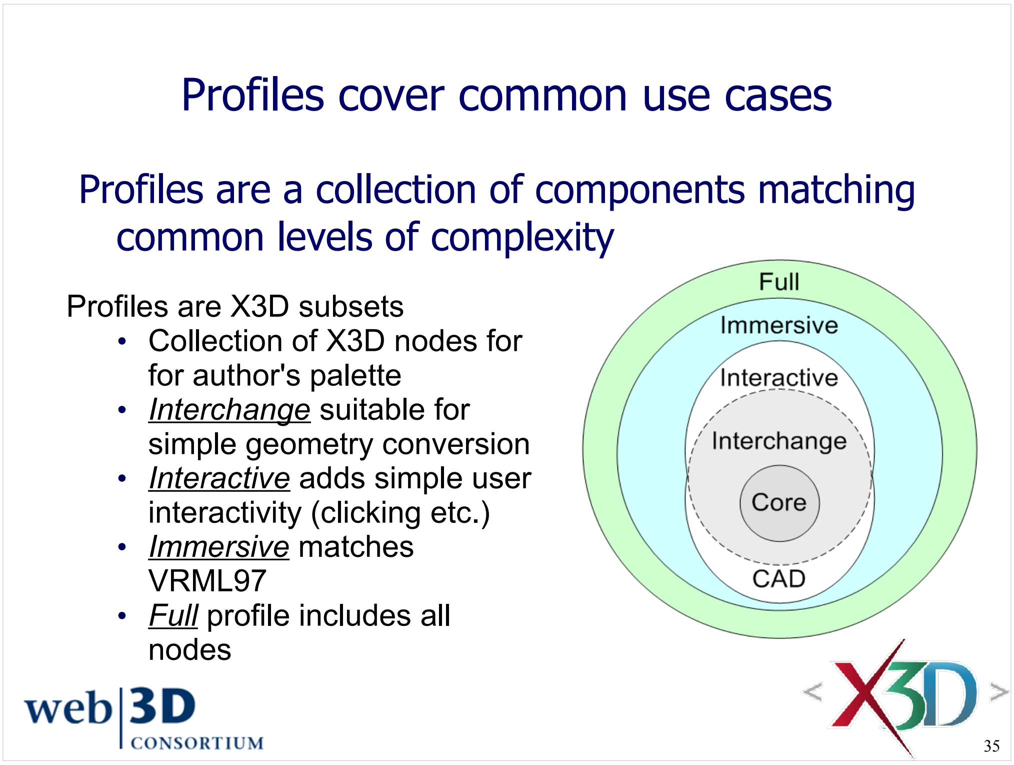
34

Someday X3D browser software applications might themselves begin to componentize, enabling a light-weight initial download followed by run-time addition of further components as needed.

Each specification chapter includes a table at the end that lists the nodes and fields which are included for each component level.

This might sound a bit complicated, but is actually a helpful thing architecturally. Authors can simply choose the best profile, rarely needing to worry about the components or levels that make them up.

Further customization within a scene is always possible using `component` statements.



Profiles cover common use cases

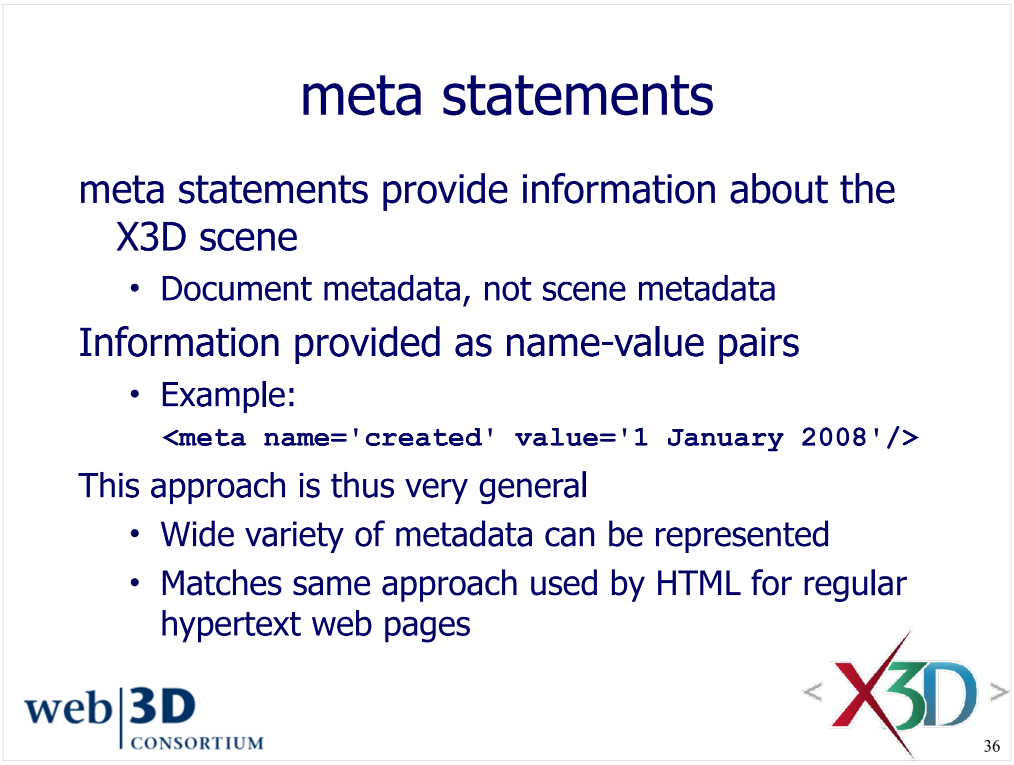
Profiles are a collection of components matching common levels of complexity

Profiles are X3D subsets

- Collection of X3D nodes for for author's palette
- *Interchange* suitable for simple geometry conversion
- *Interactive* adds simple user interactivity (clicking etc.)
- *Immersive* matches VRML97
- *Full* profile includes all nodes

Full
Immersive
Interactive
Interchange
Core
CAD
web3D CONSORTIUM
<X3D>
35

This is the profiles and components “onion” diagram



meta statements

meta statements provide information about the X3D scene

- Document metadata, not scene metadata

Information provided as name-value pairs

- Example:

```
<meta name='created' value='1 January 2008'/>
```

This approach is thus very general

- Wide variety of metadata can be represented
- Matches same approach used by HTML for regular hypertext web pages

36

newScene.x3d includes a number of prompts for authors to fill in the proper metadata

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

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

A variety of metadata standards exist that specify the proper metadata terms to use. This allows consistent searchability among data files that follow the metadata norms.

<!-- Additional authoring resources for meta-tags:

http://www.dublincore.org/documents/dcmi-terms

http://www.dublincore.org/documents/dces

http://www.w3.org/TR/html4/struct/global.html#h-7.4.4

http://vancouver-webpages.com/META

http://vancouver-webpages.com/META/about-mk-metas2.html

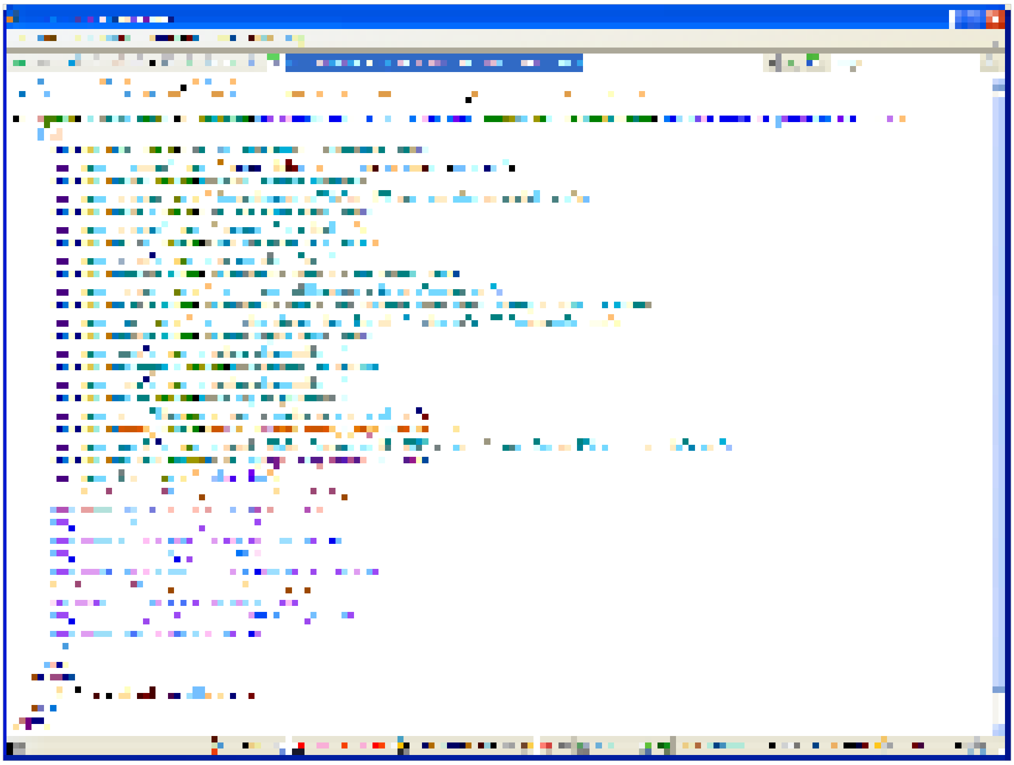
Additional authoring resources for language codes:

ftp://ftp.isi.edu/in-notes/bcp/bcp47.txt

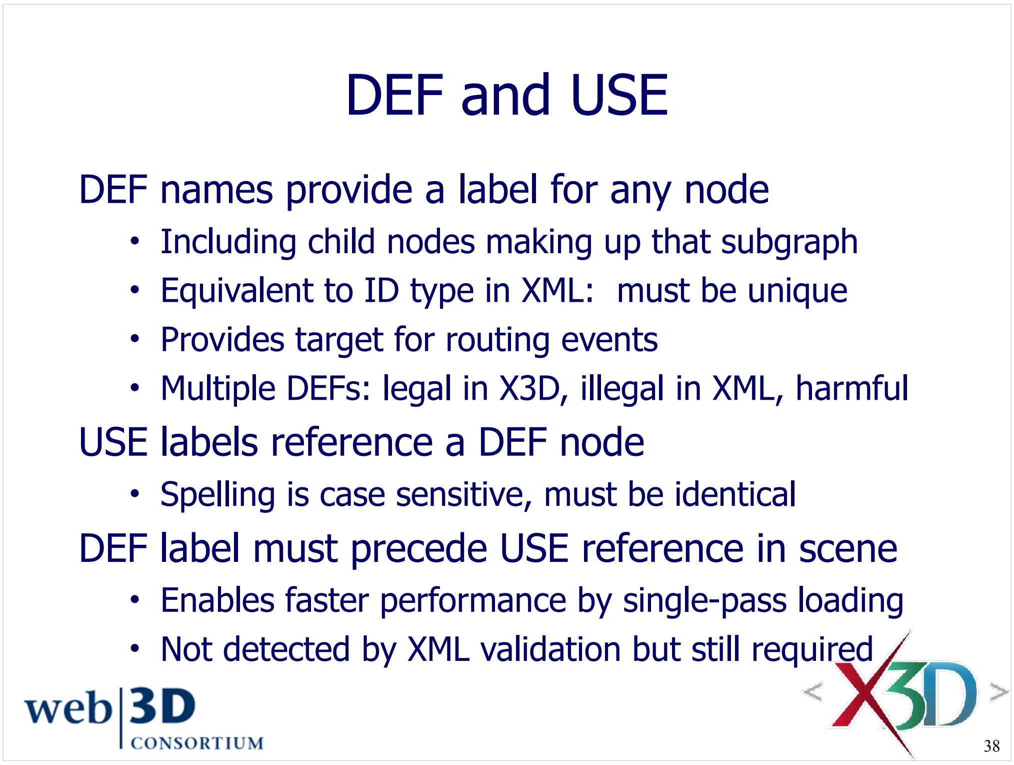
http://www.loc.gov/standards/iso639-2/langhome.html

http://www.iana.org/numbers.html#L

-->



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



DEF and USE

DEF names provide a label for any node

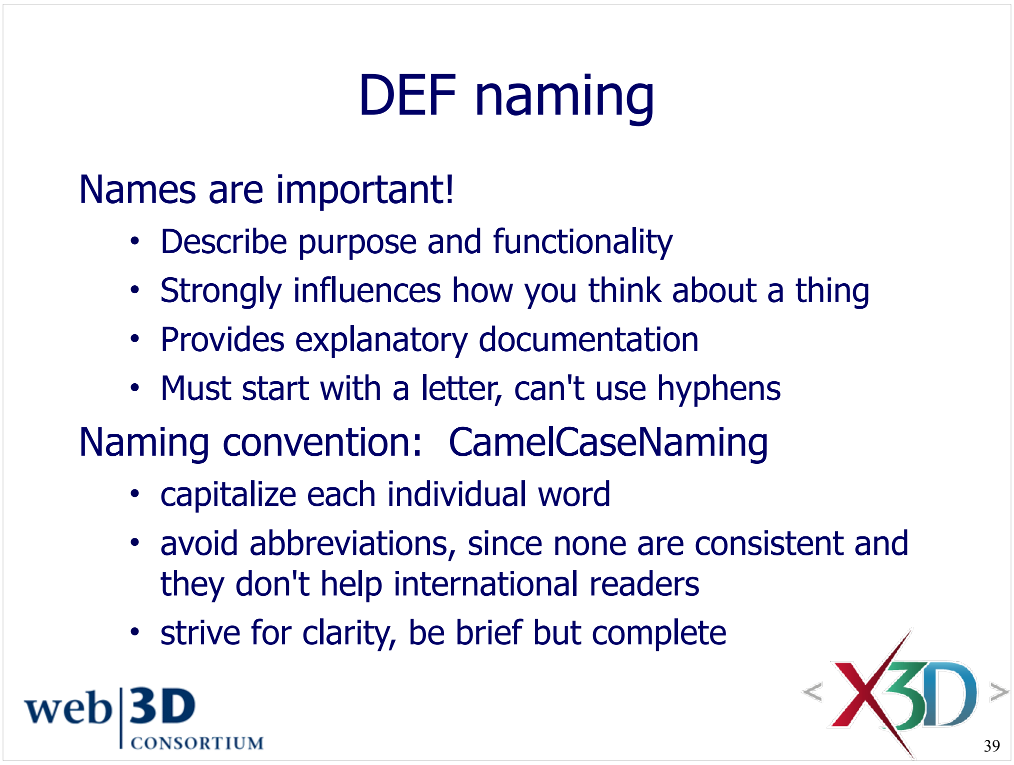
- Including child nodes making up that subgraph
- Equivalent to ID type in XML: must be unique
- Provides target for routing events
- Multiple DEFs: legal in X3D, illegal in XML, harmful

USE labels reference a DEF node

- Spelling is case sensitive, must be identical

DEF label must precede USE reference in scene

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



DEF naming

Names are important!

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

Naming convention: CamelCaseNaming

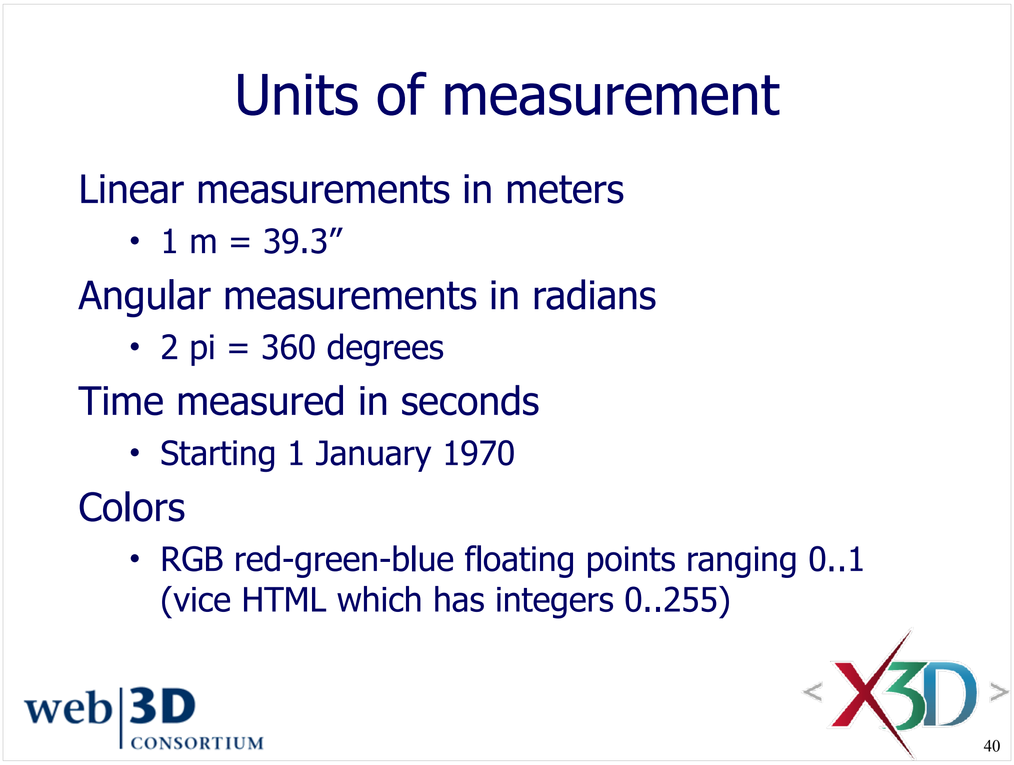
- capitalize each individual word
- avoid abbreviations, since none are consistent and they don't help international readers
- strive for clarity, be brief but complete

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

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

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

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



Units of measurement

Linear measurements in meters

- 1 m = 39.3"

Angular measurements in radians

- 2 pi = 360 degrees

Time measured in seconds

- Starting 1 January 1970

Colors

- RGB red-green-blue floating points ranging 0..1 (vice HTML which has integers 0..255)

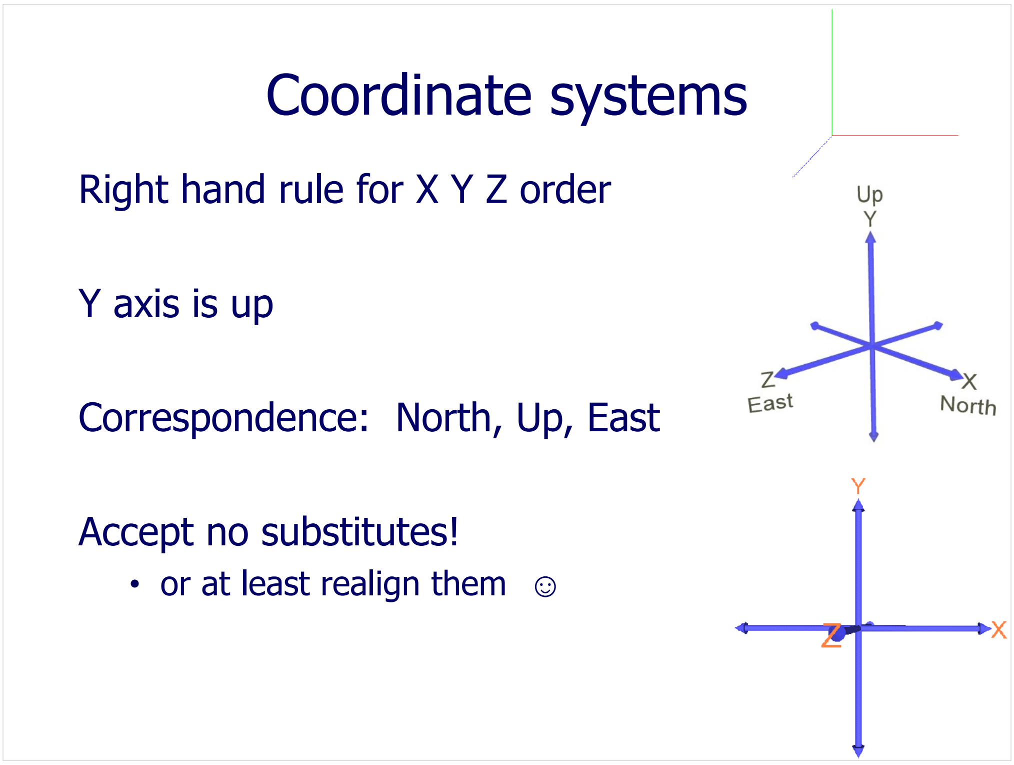
web 3D CONSORTIUM

<X3D>

40

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

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



Coordinate systems

Right hand rule for X Y Z order

Y axis is up

Correspondence: North, Up, East

Accept no substitutes!

- or at least realign them ☺

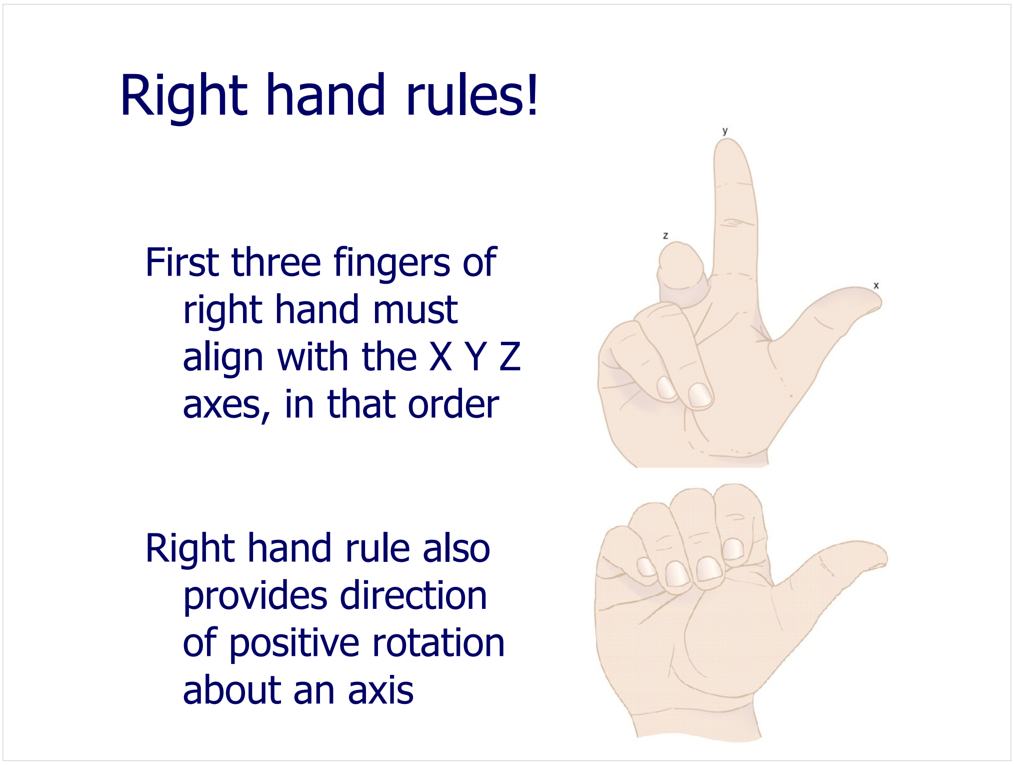
See Figures 3.1 and 3.1, page 68, *X3D for Web Authors*

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

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

The displayed example is

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



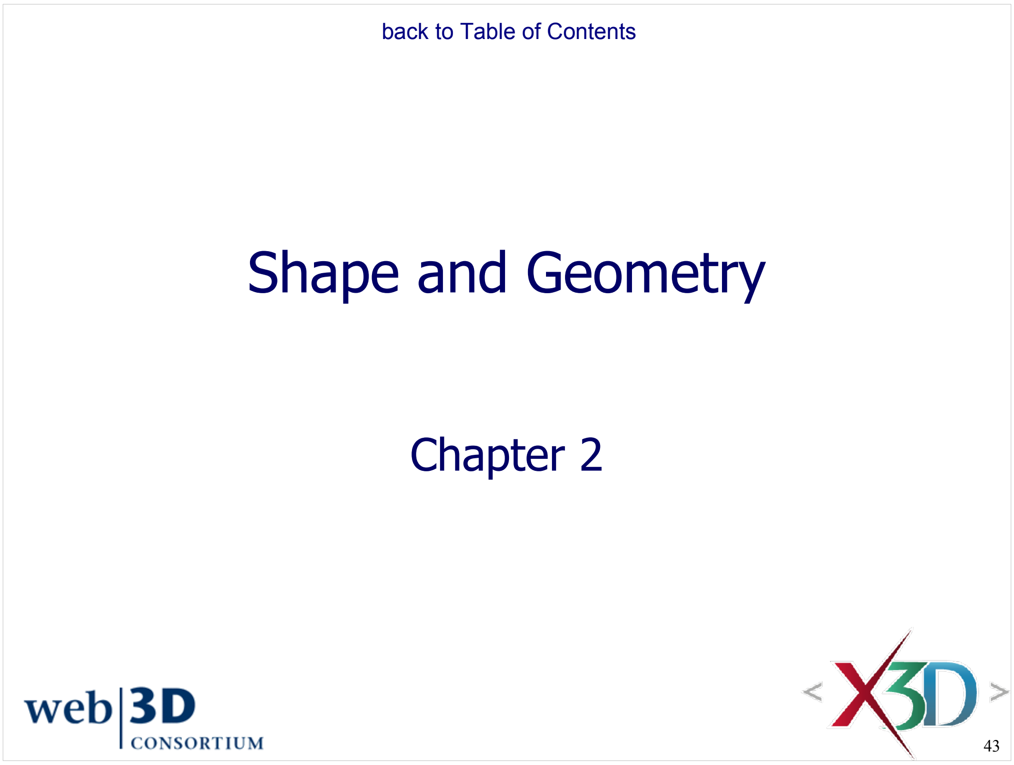
Right hand rules!

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

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

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

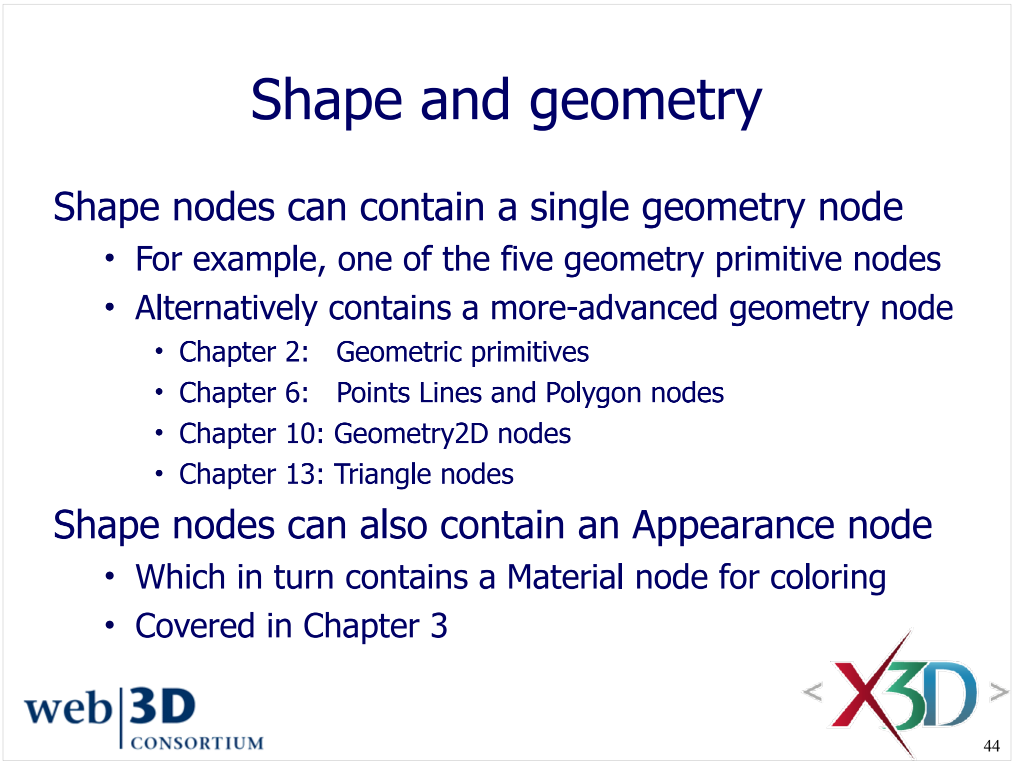
Instructors and students alike should frequently use their right hand to illustrate proper orientation relationships. It is a big help. Don't worry about onlookers.



back to Table of Contents

Shape and Geometry

Chapter 2



Shape and geometry

Shape nodes can contain a single geometry node

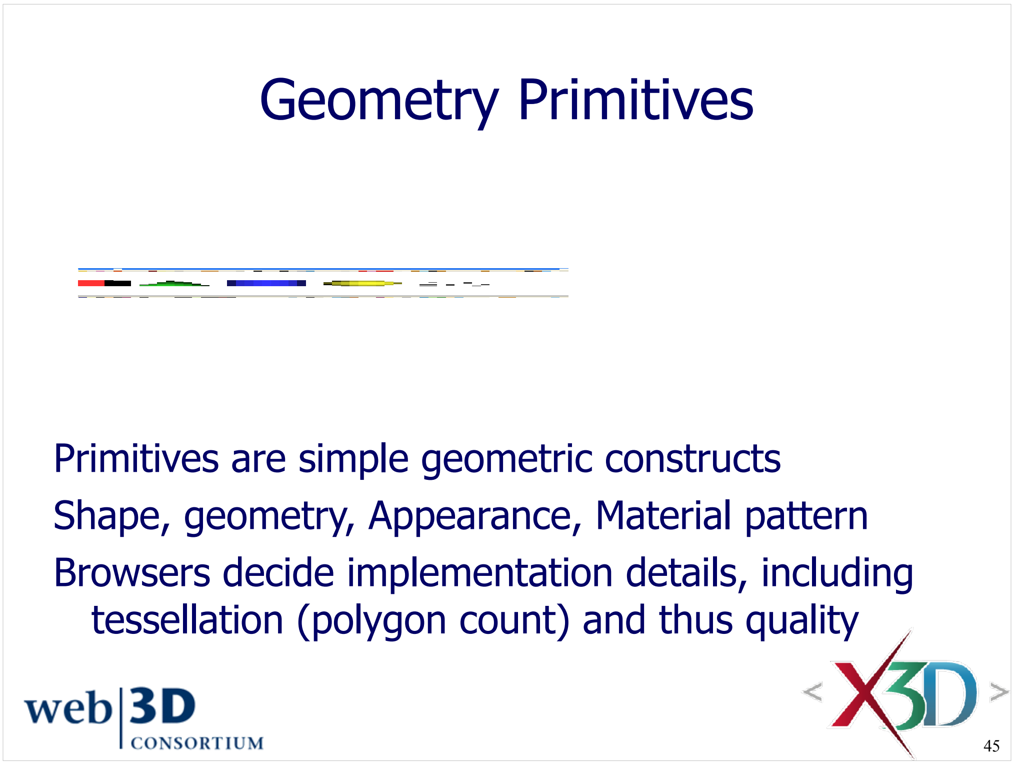
- For example, one of the five geometry primitive nodes
- Alternatively contains a more-advanced geometry node
  - Chapter 2: Geometric primitives
  - Chapter 6: Points Lines and Polygon nodes
  - Chapter 10: Geometry2D nodes
  - Chapter 13: Triangle nodes

Shape nodes can also contain an Appearance node

- Which in turn contains a Material node for coloring
- Covered in Chapter 3

Since every individual piece of geometry to be drawn must have a parent Shape node, expect to see a lot of Shape nodes in your X3D scenes.

The structure provided by having many Shape nodes helps keep a scene organized and clearly separates capabilities that might otherwise get unintentionally mixed up.



Geometry Primitives

- Primitives are simple geometric constructs
- Shape, geometry, Appearance, Material pattern
- Browsers decide implementation details, including tessellation (polygon count) and thus quality

45

Figure 2.2b, page 39, *X3D for Web Authors*

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d

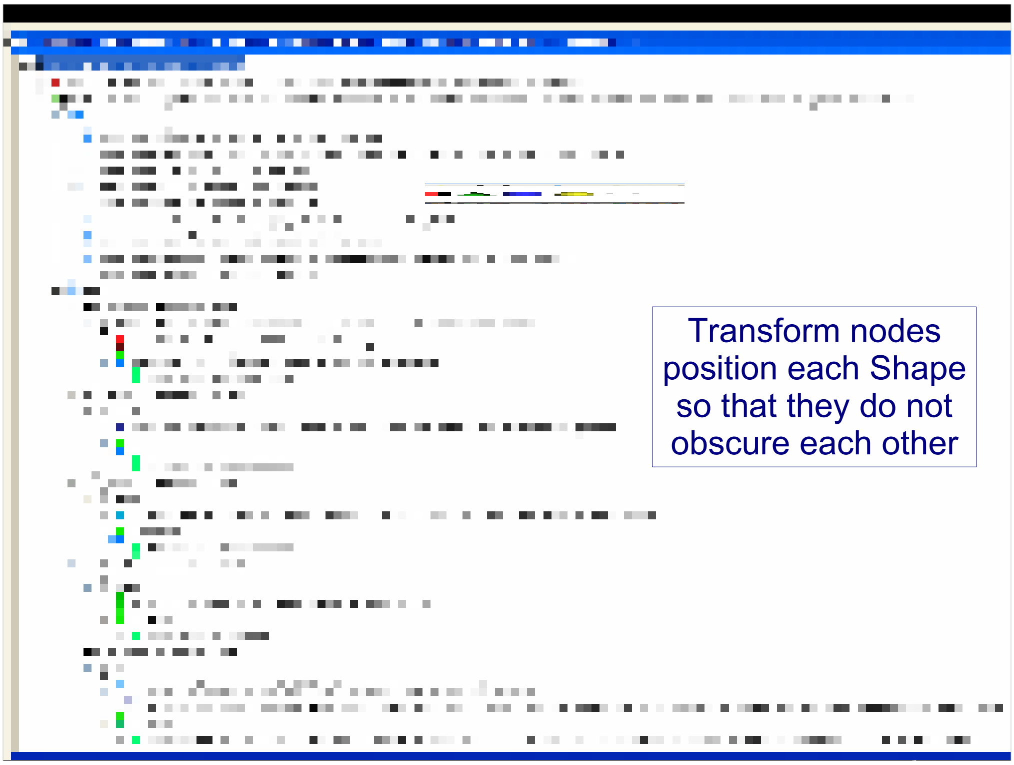
There are five primitive geometry nodes: Box Cone Cylinder Sphere and Text

Improving the polygon count of primitive geometry is a frequently requested X3D feature.

Some browsers (e.g. Xj3D) allow setting a parameter for primitive quality.

Maybe a new field will eventually be added to the X3D specification, or maybe not.

Authors can generate their own geometry (e.g. IndexedFaceSet) if they do not want to live with the uncertainty of browser quality when drawing geometry primitives.

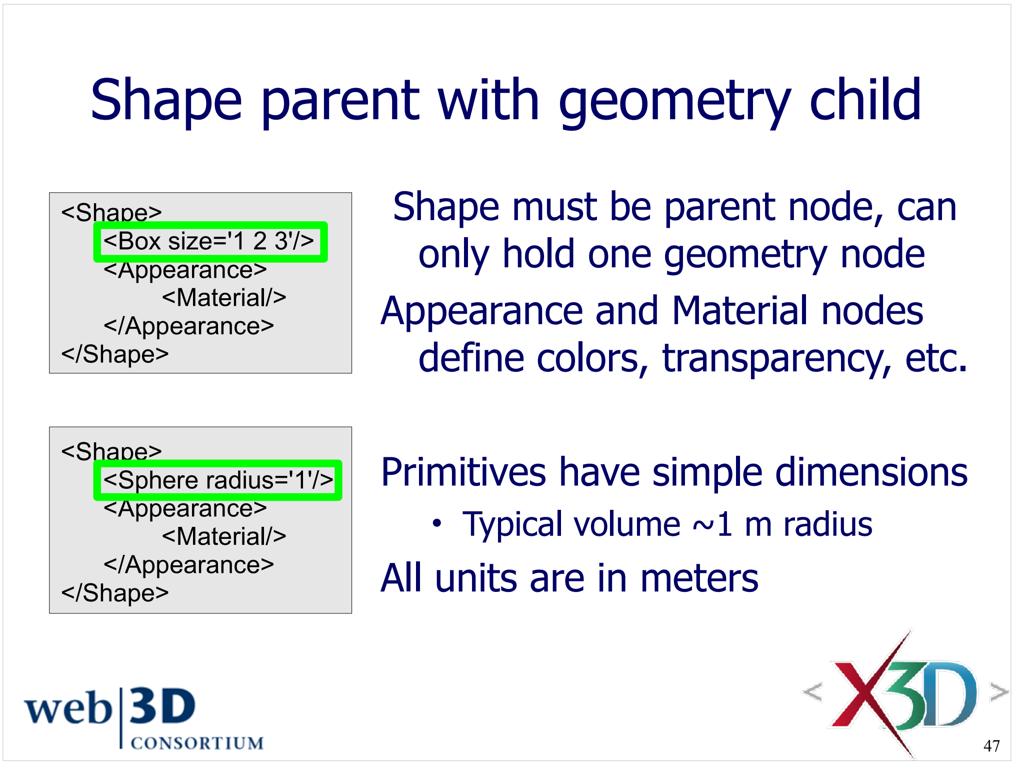


Transform nodes position each Shape so that they do not obscure each other

Figures 2.1 and 2.2, page 39, *X3D for Web Authors*

http://www.x3dbook.com/examples/X3dForWebAuthors/Chapter02-GeometryPrimitives/GeometryPrimitiveNodes.x3d

This scene-graph screen snapshot was taken using X3D-Edit 3.1.



Shape parent with geometry child

```
<Shape>
    <Box size='1 2 3'/>
    <Appearance>
         <Material/>
    </Appearance>
</Shape>
```

Shape must be parent node, can only hold one geometry node

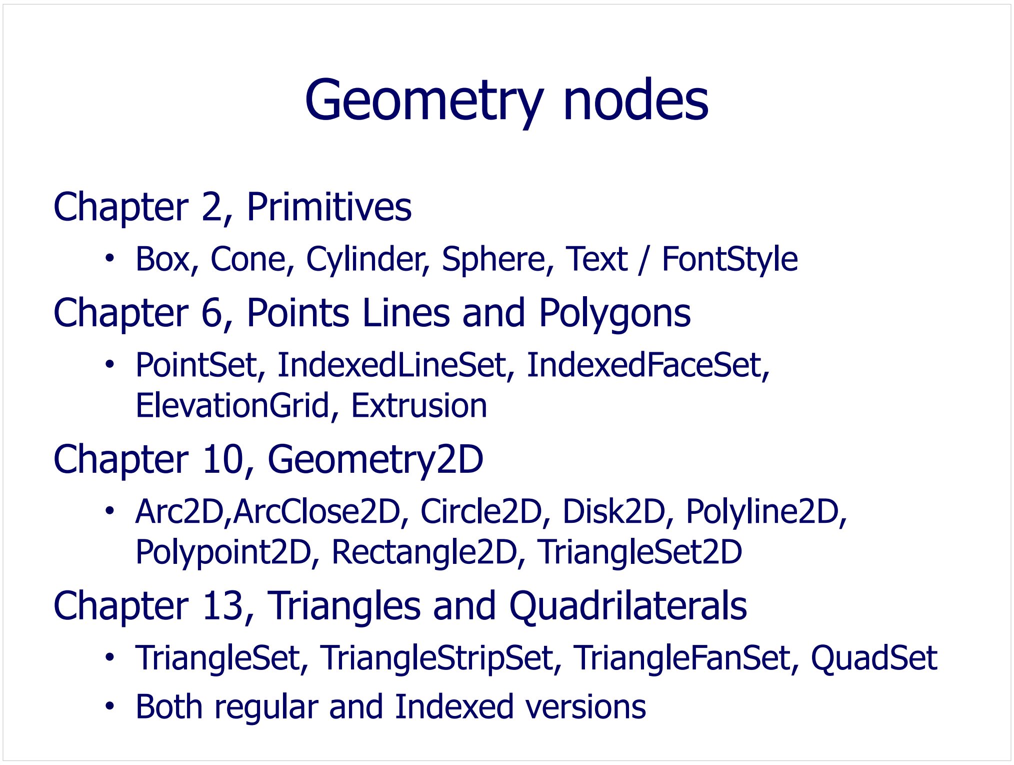
Appearance and Material nodes define colors, transparency, etc.

```
<Shape>
    <Sphere radius='1'/>
    <Appearance>
         <Material/>
    </Appearance>
</Shape>
```

Primitives have simple dimensions

- Typical volume ~1 m radius

All units are in meters



Geometry nodes

Chapter 2, Primitives

- Box, Cone, Cylinder, Sphere, Text / FontStyle

Chapter 6, Points Lines and Polygons

- PointSet, IndexedLineSet, IndexedFaceSet, ElevationGrid, Extrusion

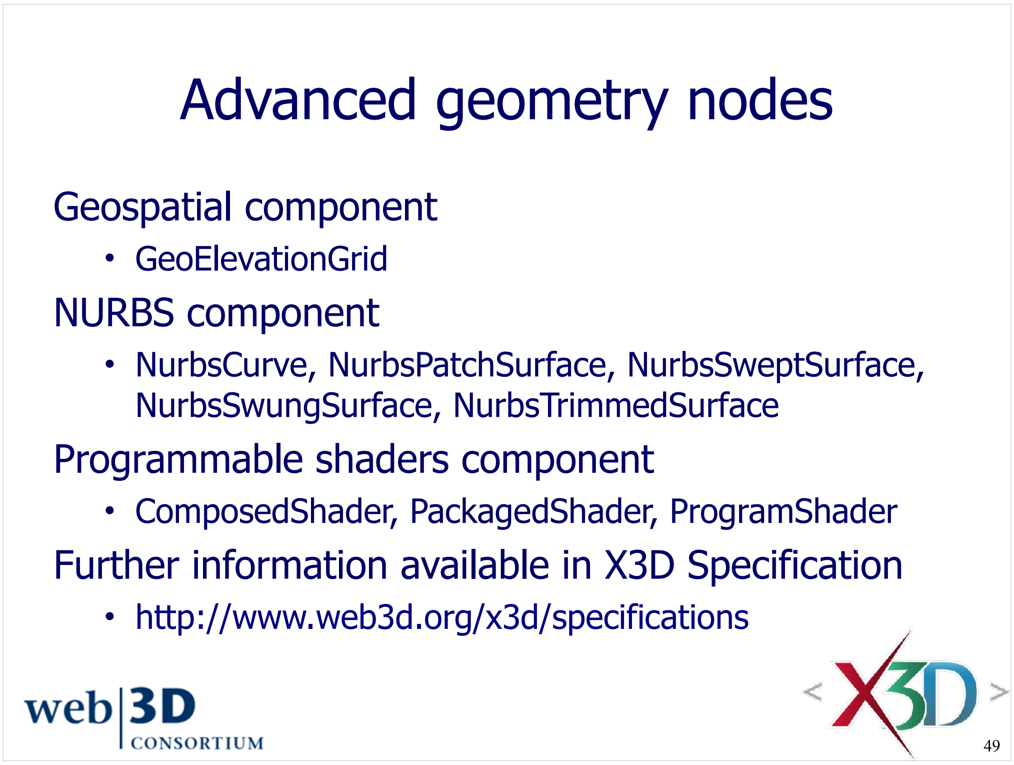
Chapter 10, Geometry2D

- Arc2D,ArcClose2D, Circle2D, Disk2D, Polyline2D, Polypoint2D, Rectangle2D, TriangleSet2D

Chapter 13, Triangles and Quadrilaterals

- TriangleSet, TriangleStripSet, TriangleFanSet, QuadSet
- Both regular and Indexed versions

The principle that one geometry node goes inside each Shape, and next to each Appearance, is consistent for all the different geometry nodes available in X3D.



Advanced geometry nodes

Geospatial component

- GeoElevationGrid

NURBS component

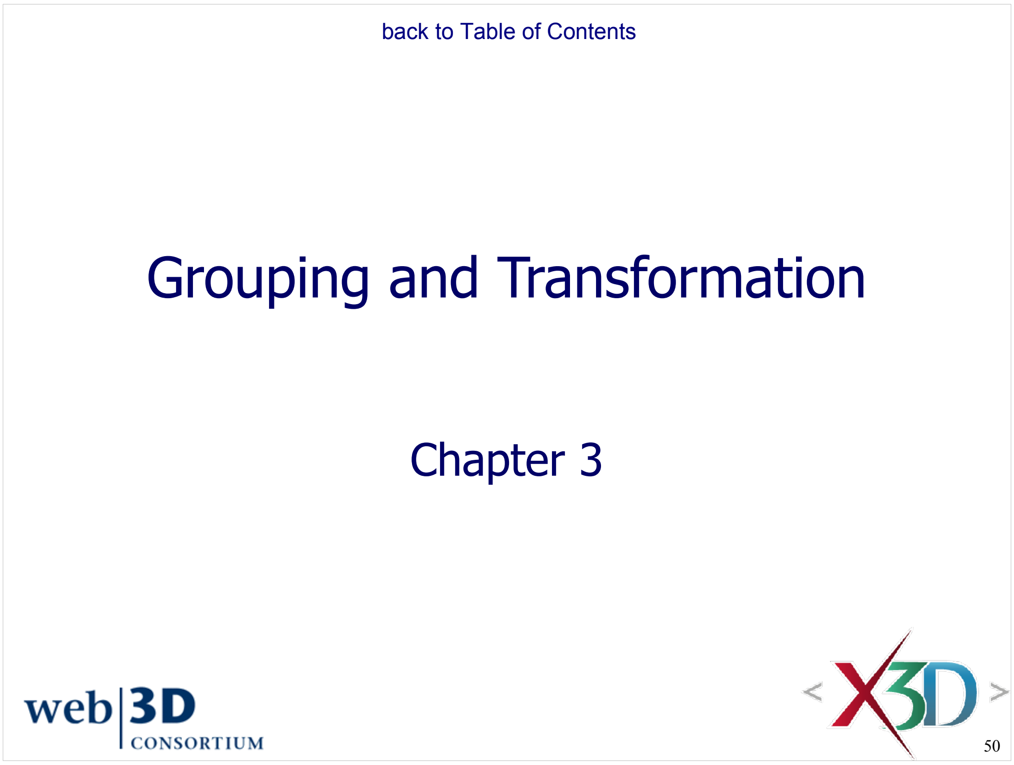
- NurbsCurve, NurbsPatchSurface, NurbsSweptSurface, NurbsSwungSurface, NurbsTrimmedSurface

Programmable shaders component

- ComposedShader, PackagedShader, ProgramShader

Further information available in X3D Specification

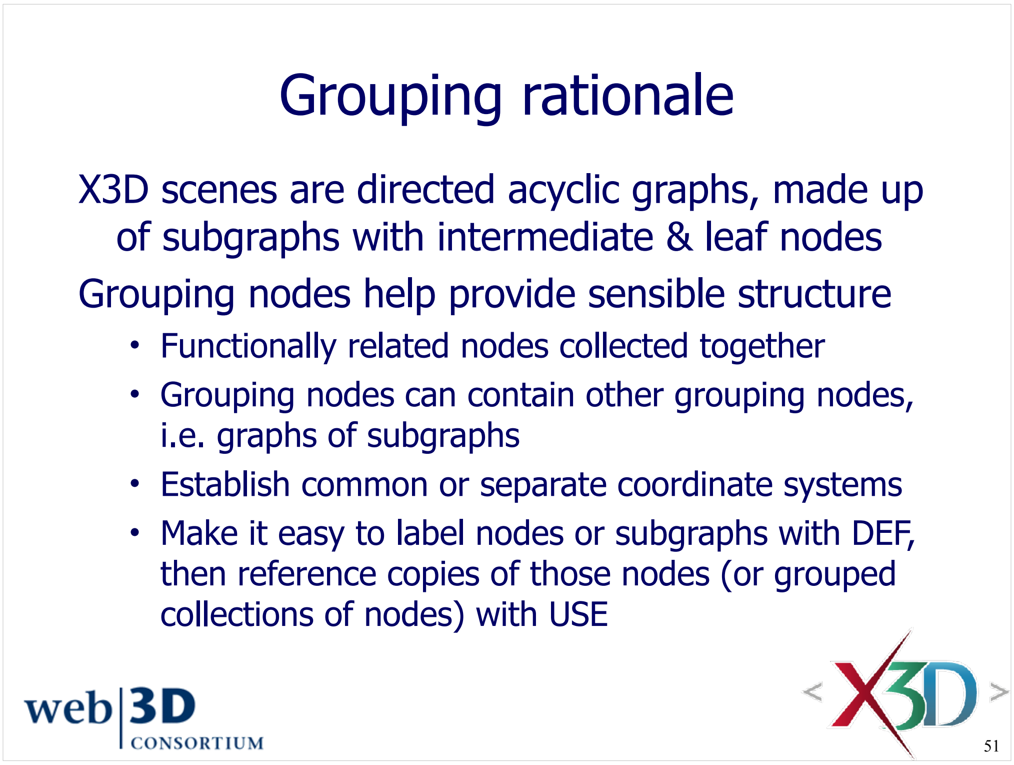
- http://www.web3d.org/x3d/specifications



back to Table of Contents

Grouping and Transformation

Chapter 3

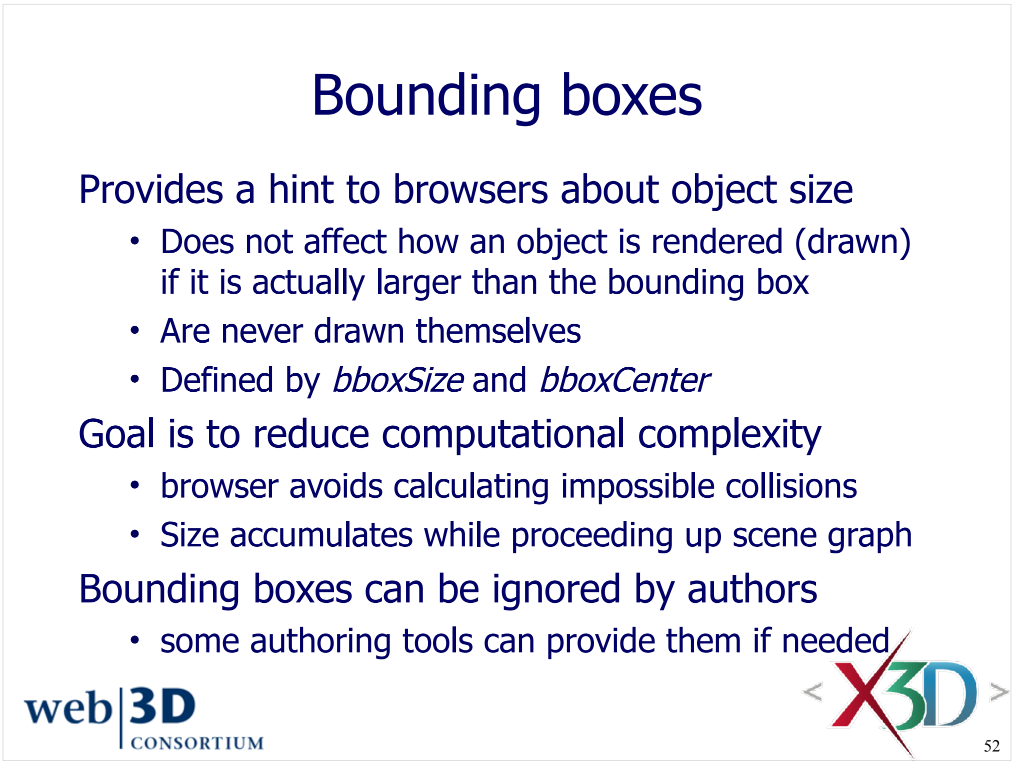


Grouping rationale

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

Grouping nodes help provide sensible structure

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



Bounding boxes

Provides a hint to browsers about object size

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

Goal is to reduce computational complexity

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

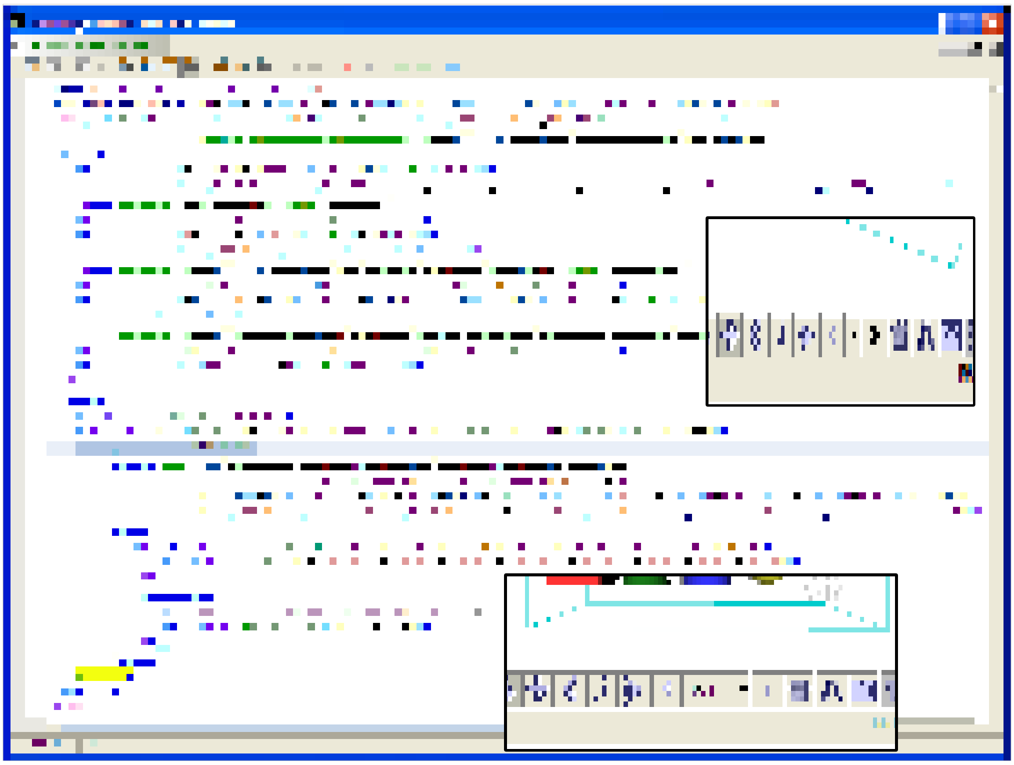
Bounding boxes can be ignored by authors

- some authoring tools can provide them if needed

52

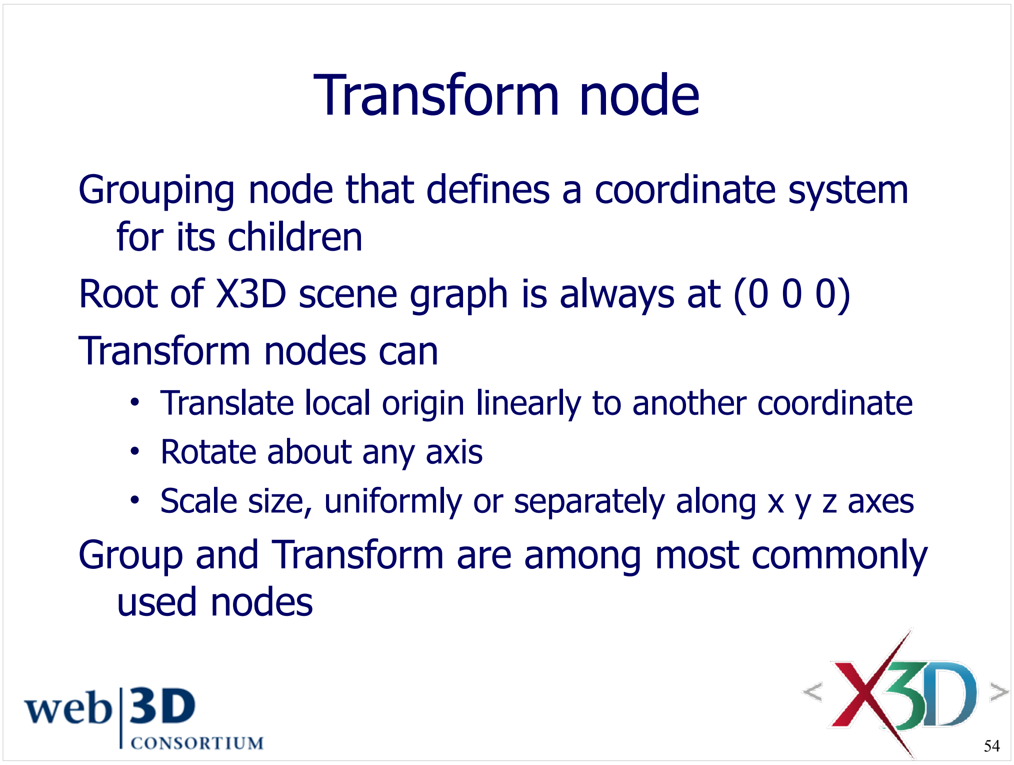
Note that bounding boxes are invisible and not displayed.

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



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

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



Transform node

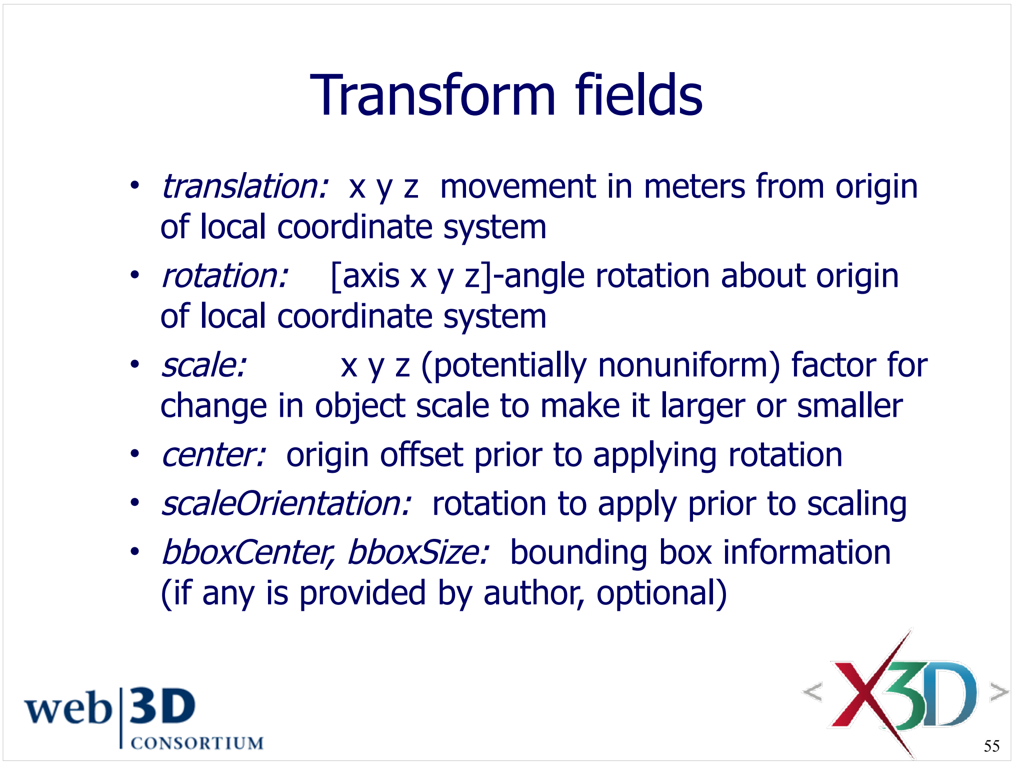
Grouping node that defines a coordinate system for its children

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

Transform nodes can

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

Group and Transform are among most commonly used nodes



Transform fields

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

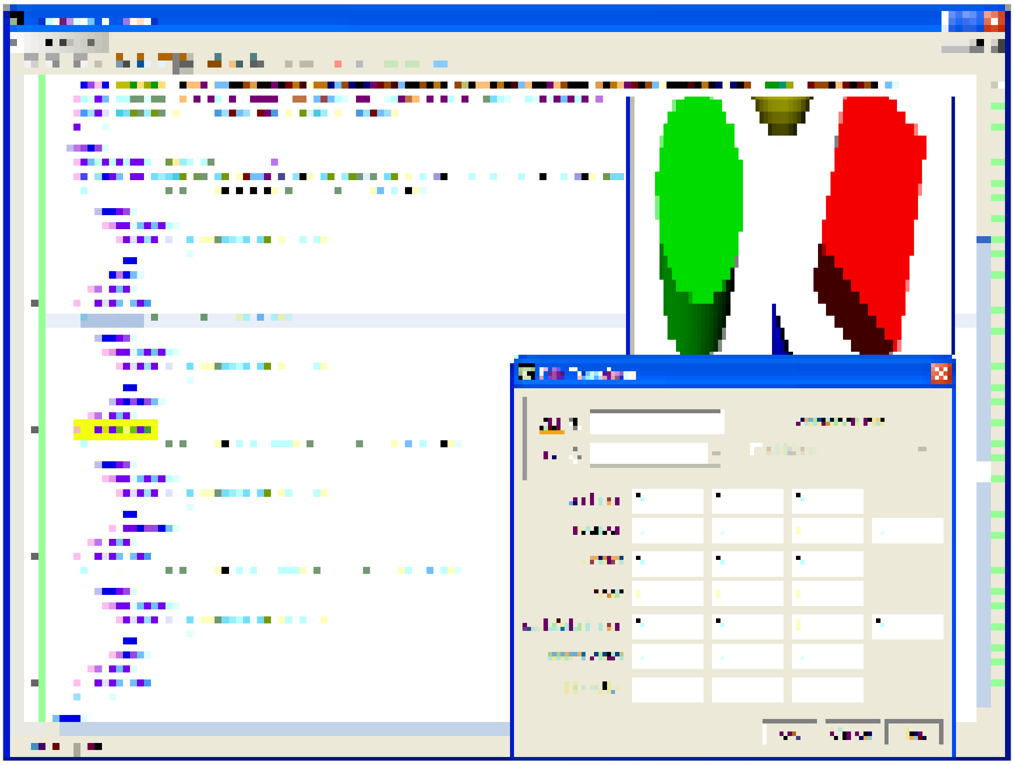
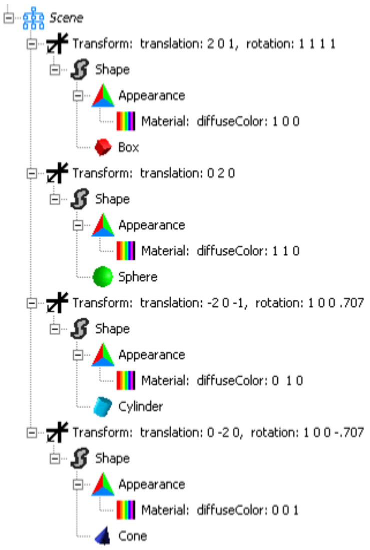


Figure 3.6, page 79, *X3D for Web Authors*

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

Each Transform is a scene subgraph



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

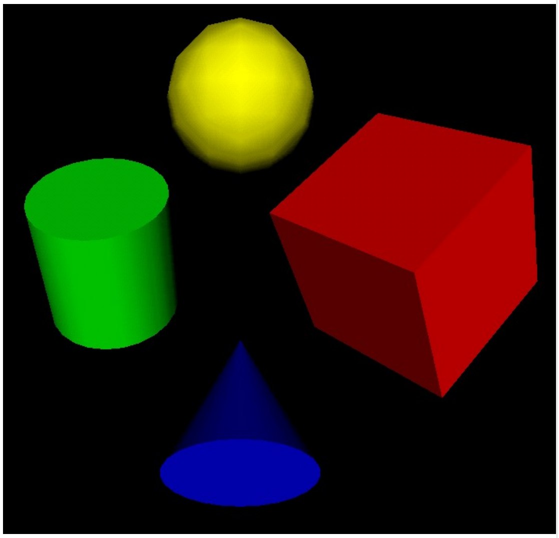
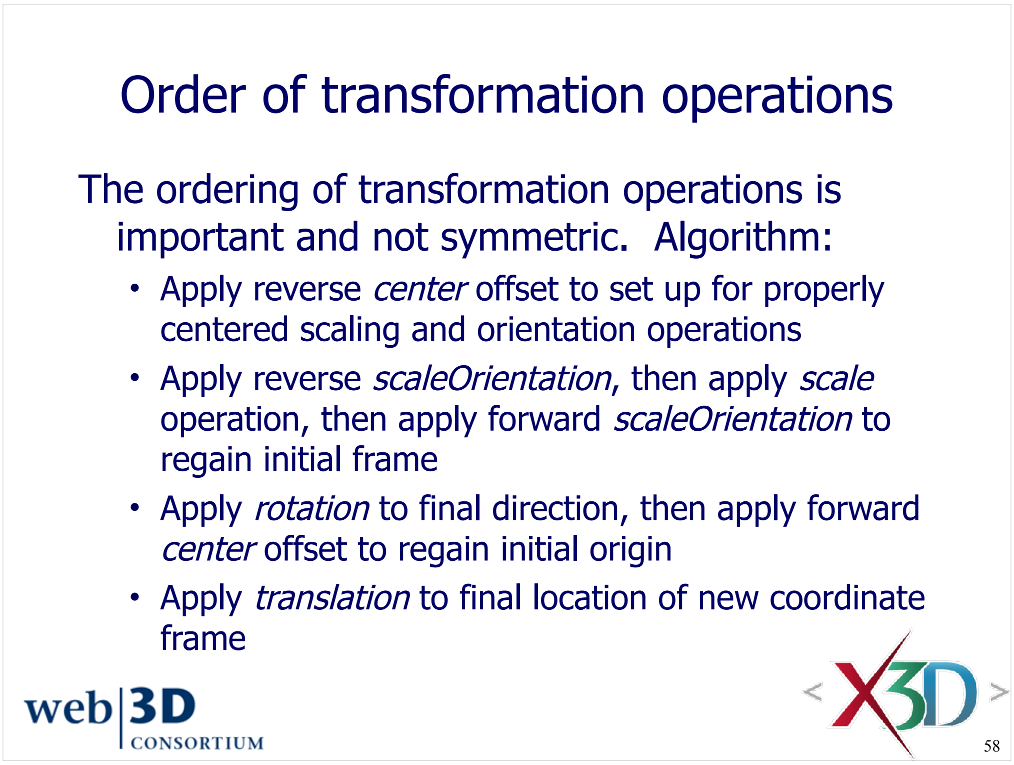


Figure 3.6, page 79, *X3D for Web Authors*

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

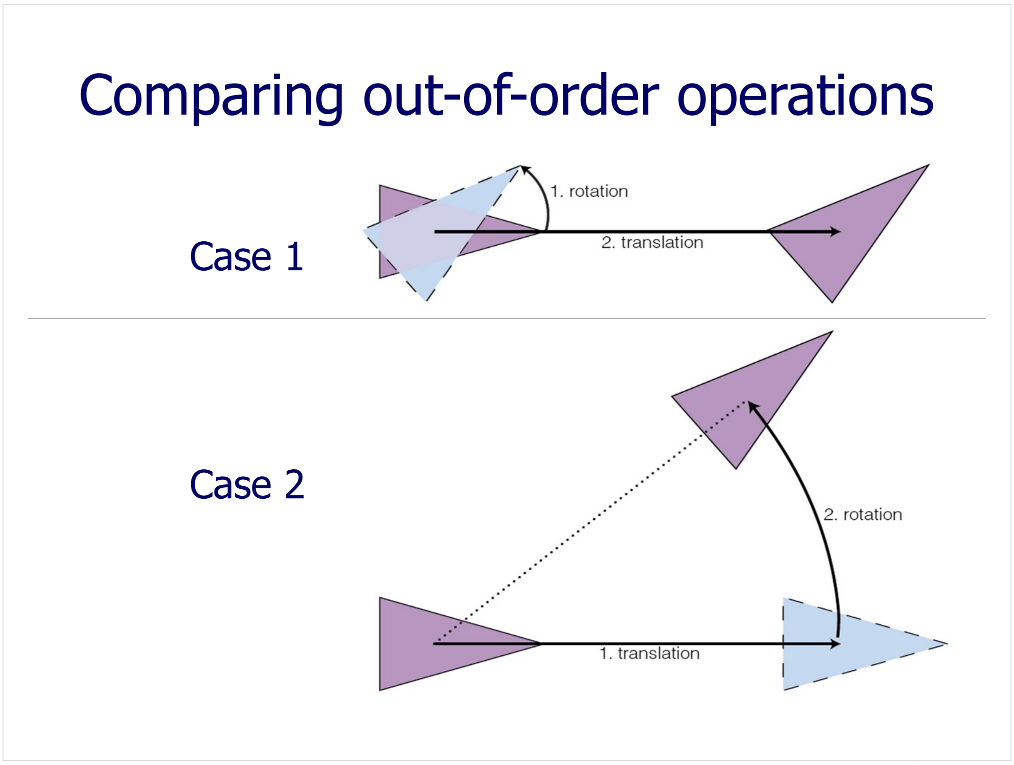


Order of transformation operations

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

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

The next slide illustrates these steps.



Comparing out-of-order operations

Case 1

Case 2

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

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

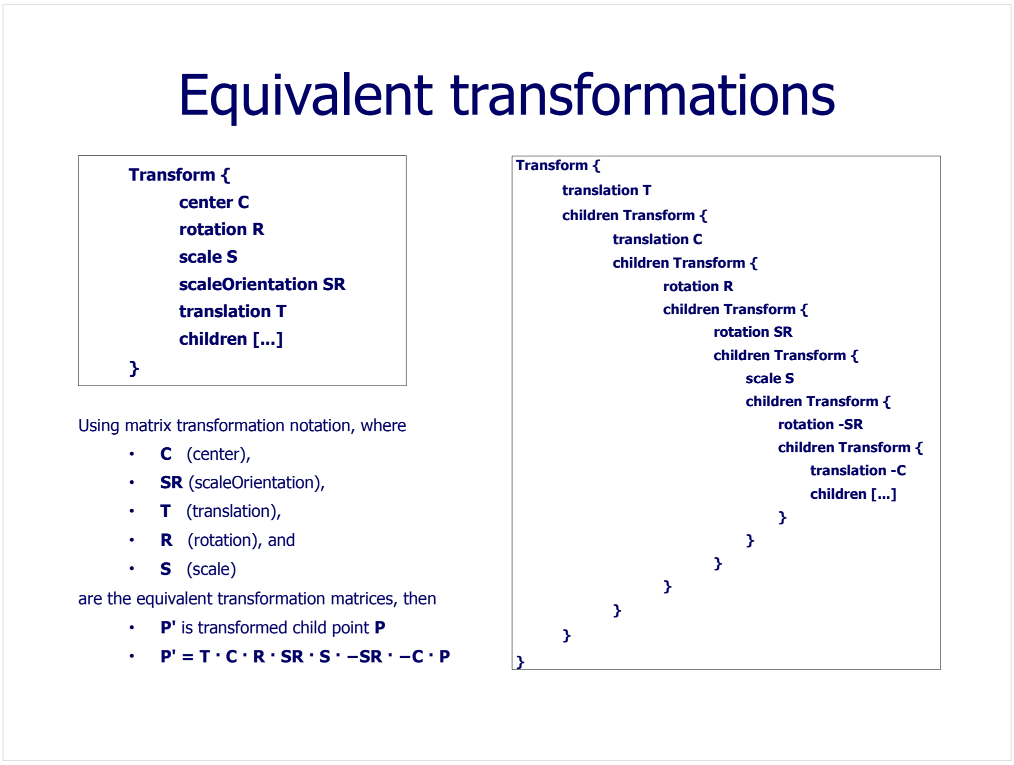
Case 2: first translation, then rotation. (Requires two Transform nodes in X3D)

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

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

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

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



Equivalent transformations

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

Using matrix transformation notation, where

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

are the equivalent transformation matrices, then

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

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

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

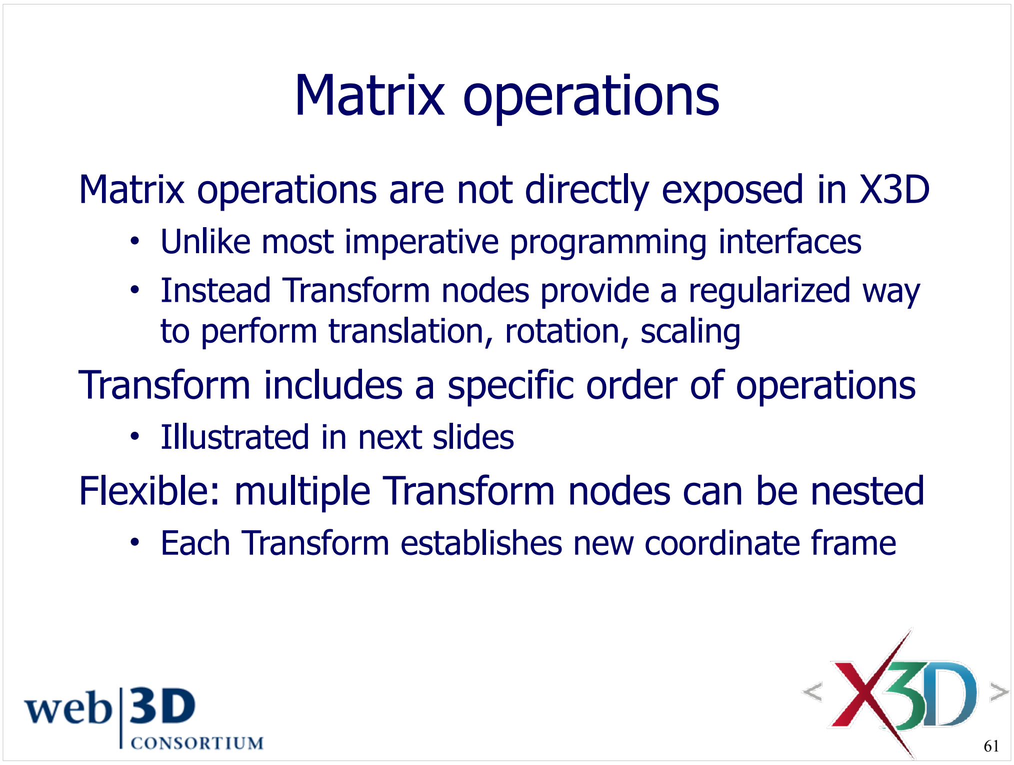
The Transform on the left is equivalent to the set of Transform nodes on the right.

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

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

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

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



Matrix operations

Matrix operations are not directly exposed in X3D

- Unlike most imperative programming interfaces
- Instead Transform nodes provide a regularized way to perform translation, rotation, scaling

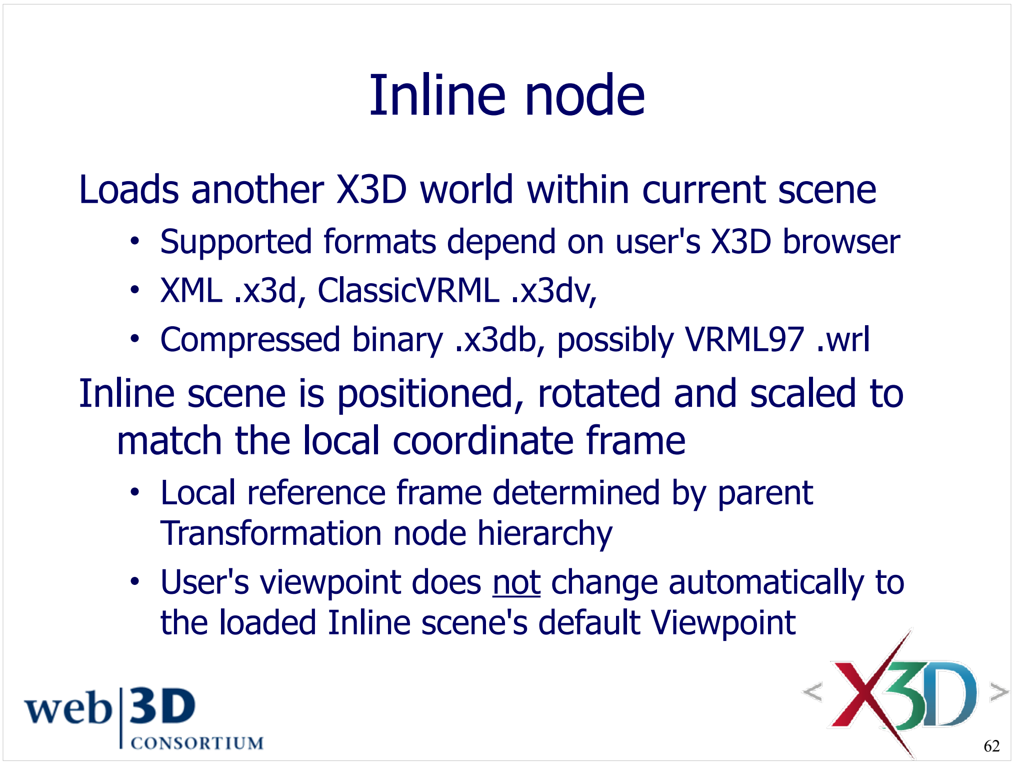
Transform includes a specific order of operations

- Illustrated in next slides

Flexible: multiple Transform nodes can be nested

- Each Transform establishes new coordinate frame

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



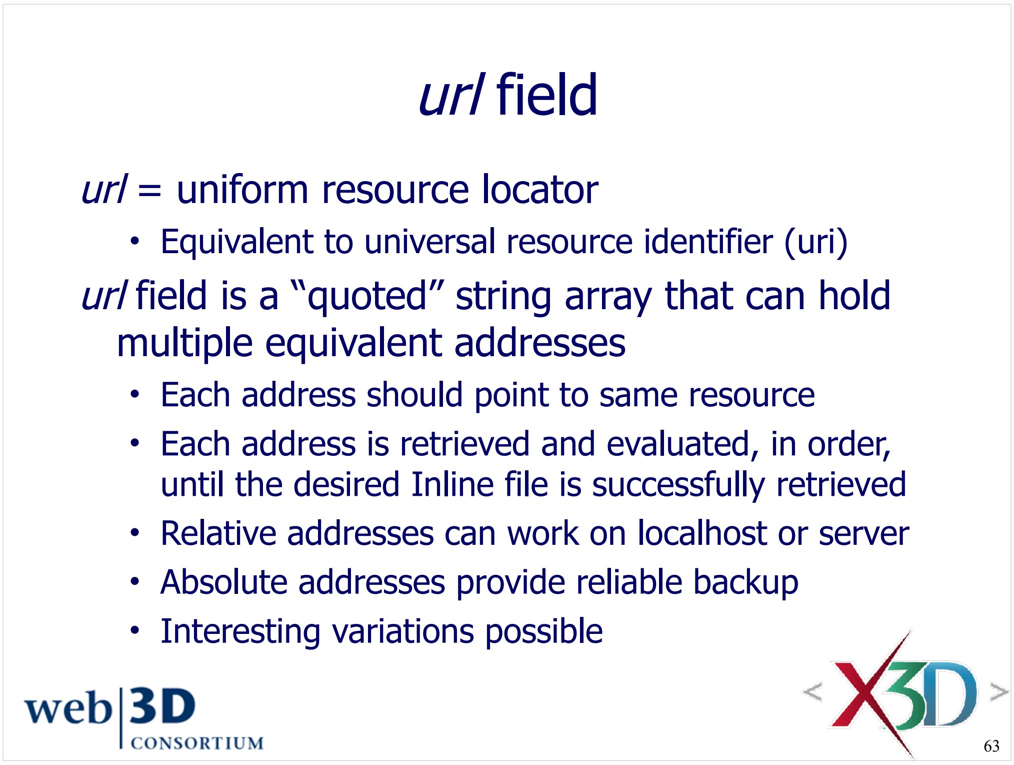
Inline node

Loads another X3D world within current scene

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

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

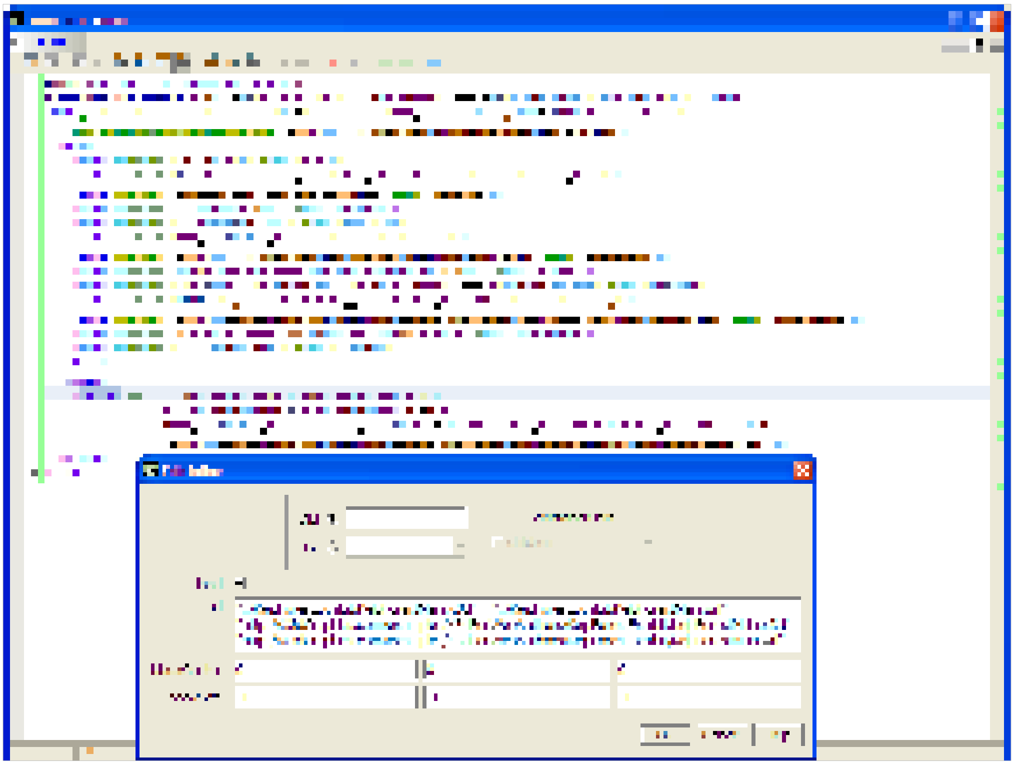
- Local reference frame determined by parent Transformation node hierarchy
- User's viewpoint does not change automatically to the loaded Inline scene's default Viewpoint



url field

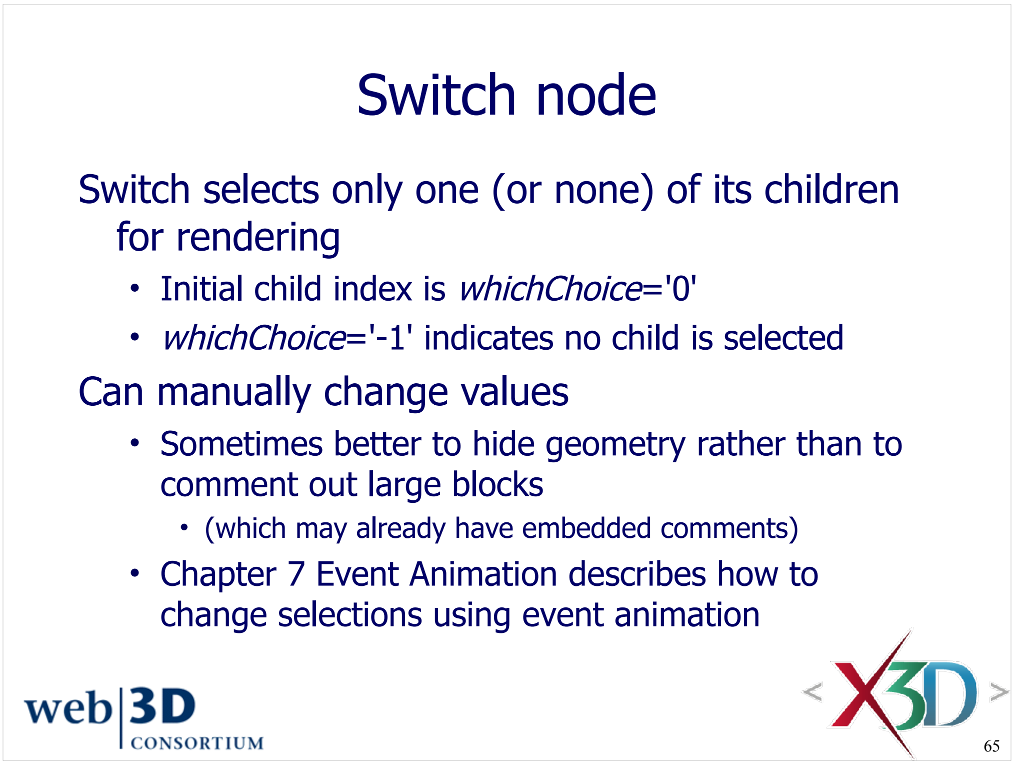
- *url* = uniform resource locator
  - Equivalent to universal resource identifier (uri)
- *url* field is a “quoted” string array that can hold multiple equivalent addresses
  - Each address should point to same resource
  - Each address is retrieved and evaluated, in order, until the desired Inline file is successfully retrieved
  - Relative addresses can work on localhost or server
  - Absolute addresses provide reliable backup
  - Interesting variations possible

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



An improved url editor is planned for X3D-Edit.

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



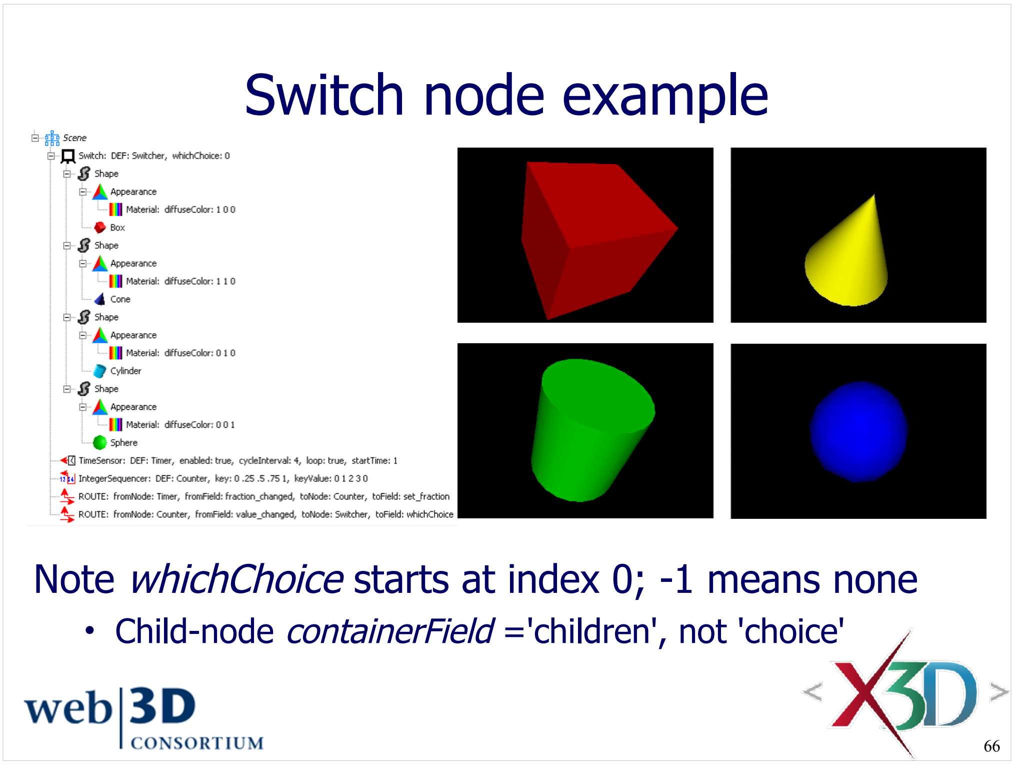
Switch node

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

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

Can manually change values

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



Switch node example

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

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

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

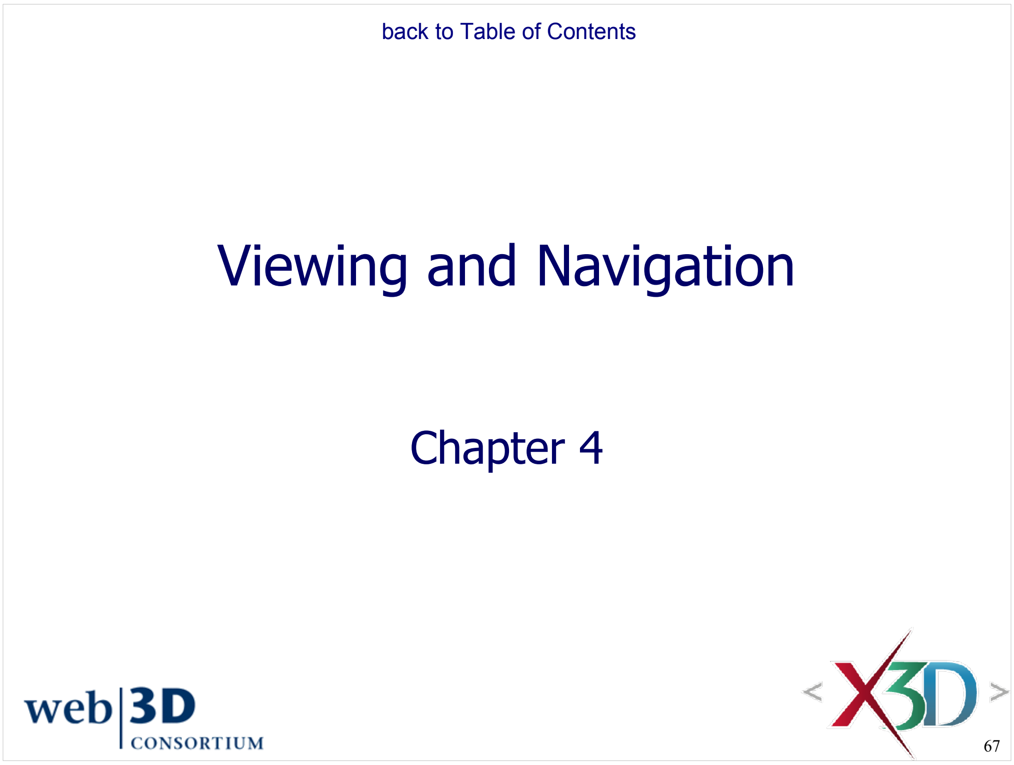
Figure 3.12, page 91, *X3D for Web Authors*

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

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

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

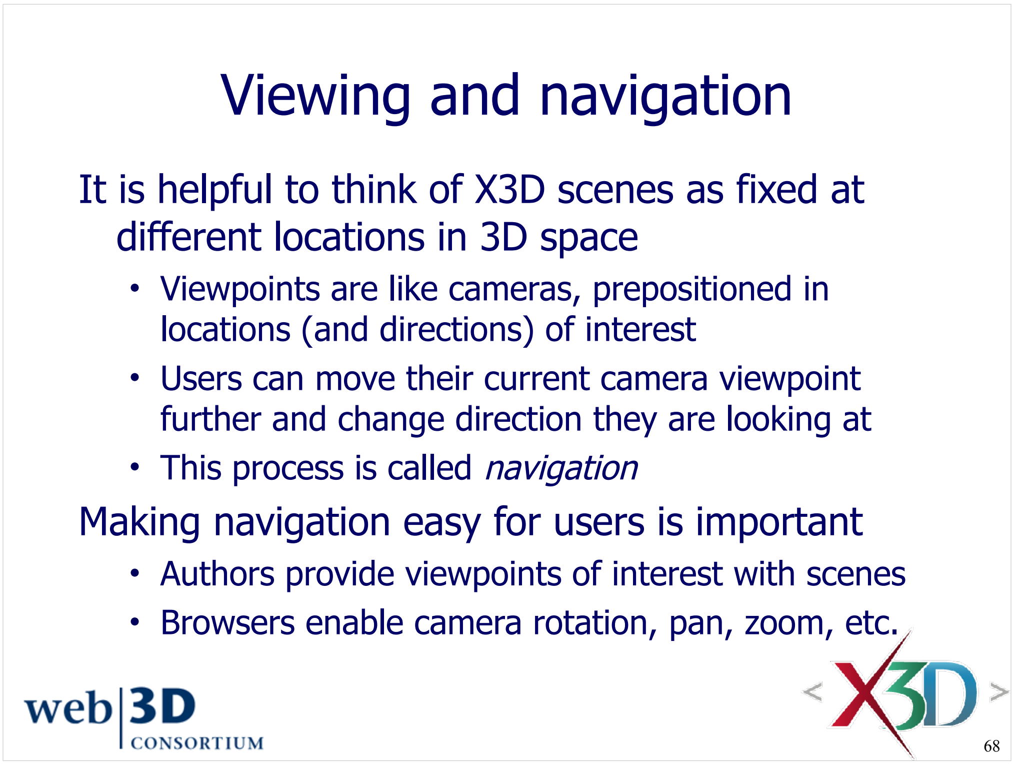
The default *containerField* value for Switch was changed to *containerField*='children' in X3D (from 'choice' in VRML97) in order to make Switch consistent with other X3DGroupingNode types.



back to Table of Contents

Viewing and Navigation

Chapter 4



Viewing and navigation

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

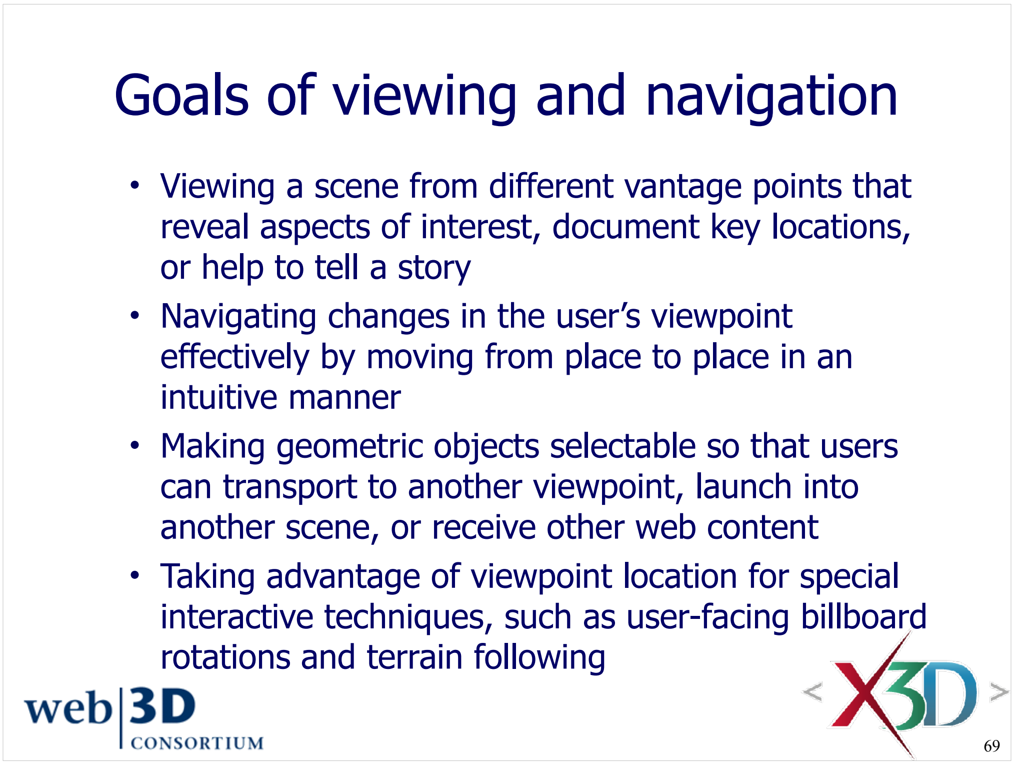
- Viewpoints are like cameras, prepositioned in locations (and directions) of interest
- Users can move their current camera viewpoint further and change direction they are looking at
- This process is called *navigation*

Making navigation easy for users is important

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

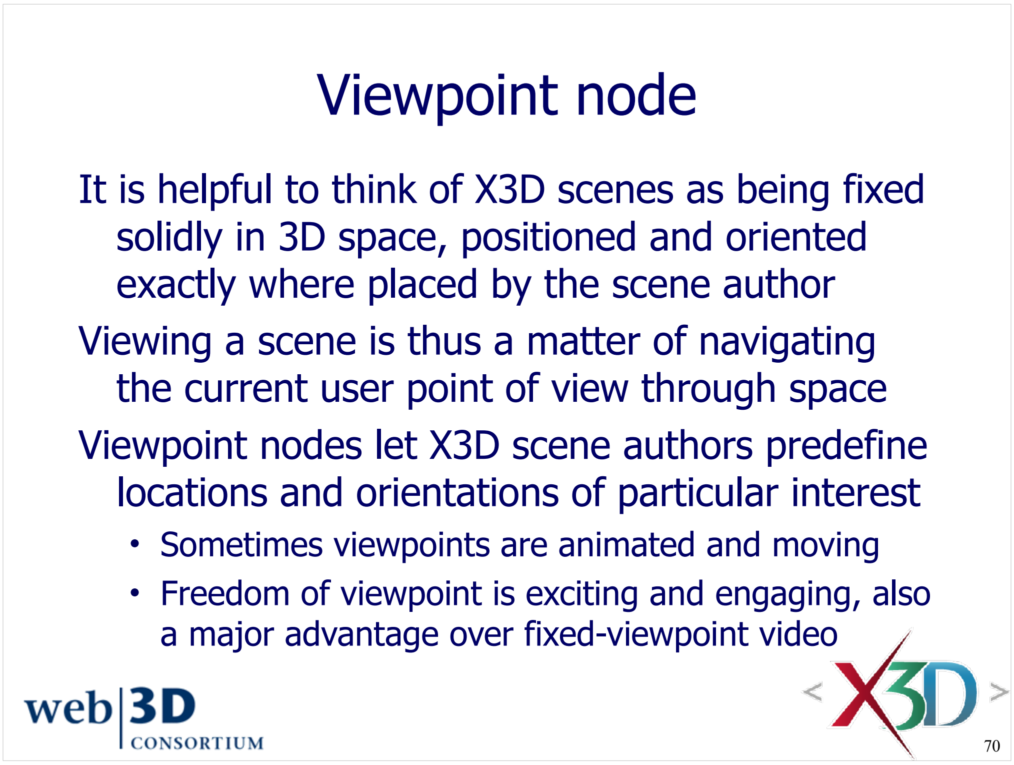
68

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



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

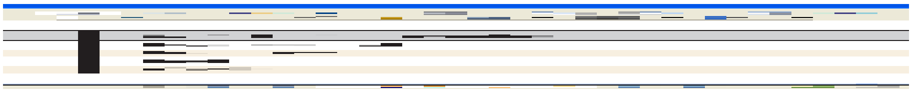
Navigation model 1

Users can select predefined Viewpoints

- Defines both position and direction of view

Users can further navigate around scene

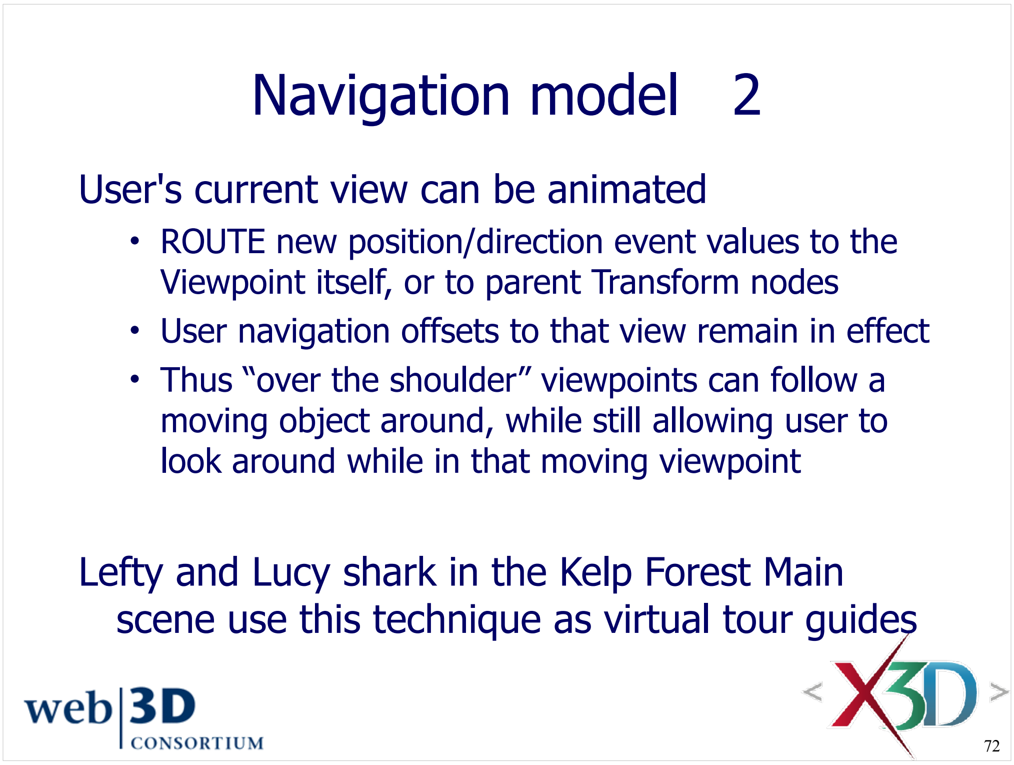
- Using pointing device or hot keys
- Chosen viewpoint remains bound
- Browser applies offsets using user-driven changes



web|3D CONSORTIUM

<X3D>

Figure 4.9. Recommended Keyboard Navigation Keys and Responses

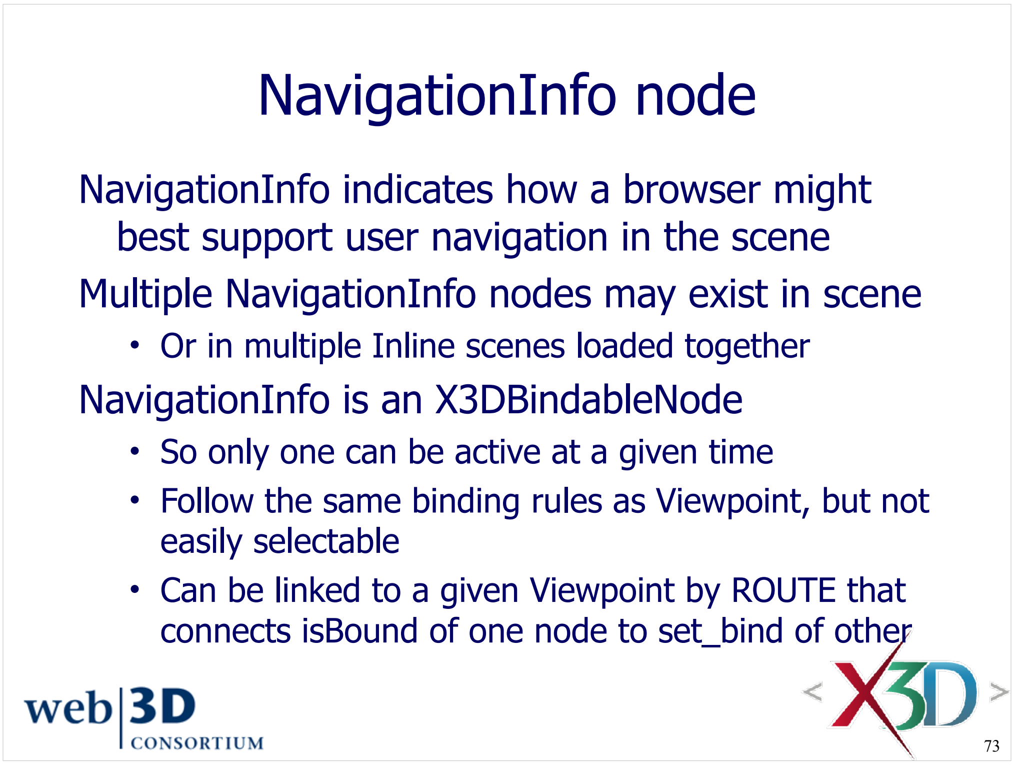


Navigation model 2

User's current view can be animated

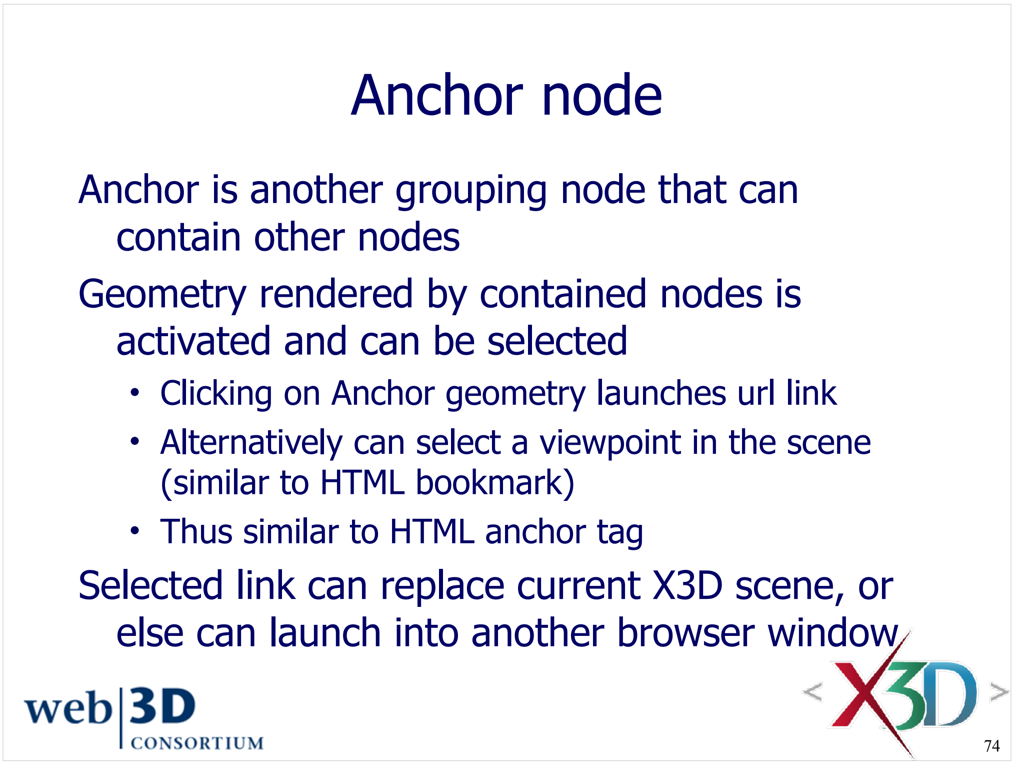
- ROUTE new position/direction event values to the Viewpoint itself, or to parent Transform nodes
- User navigation offsets to that view remain in effect
- Thus “over the shoulder” viewpoints can follow a moving object around, while still allowing user to look around while in that moving viewpoint

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



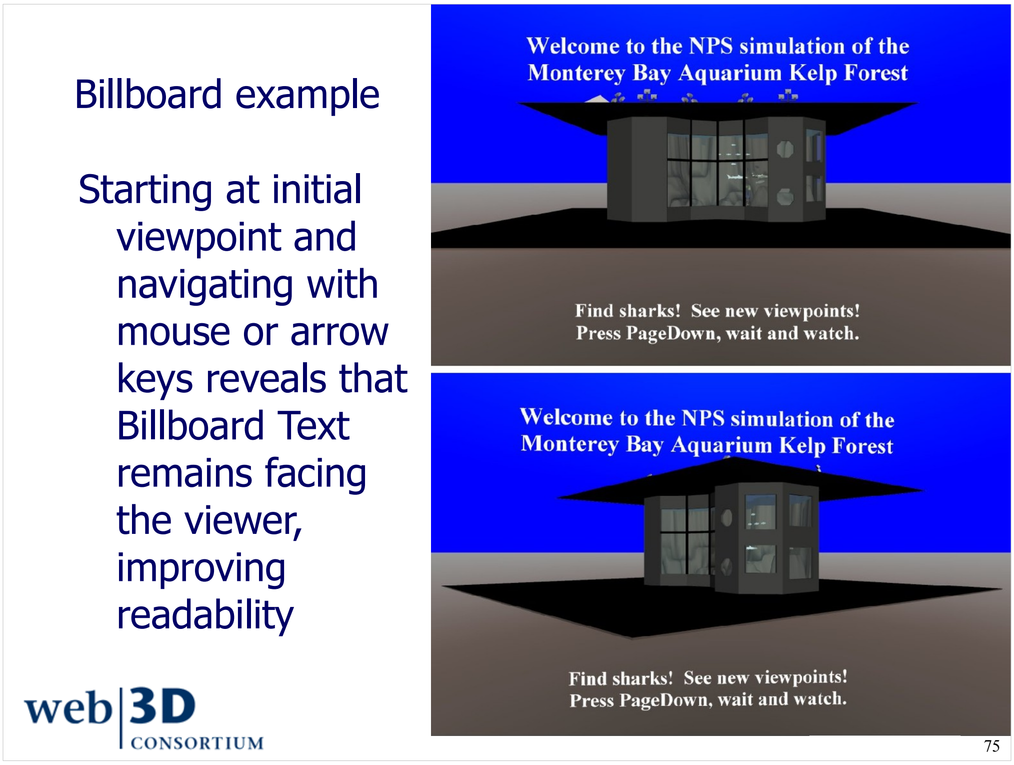
Anchor node

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



Billboard example

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

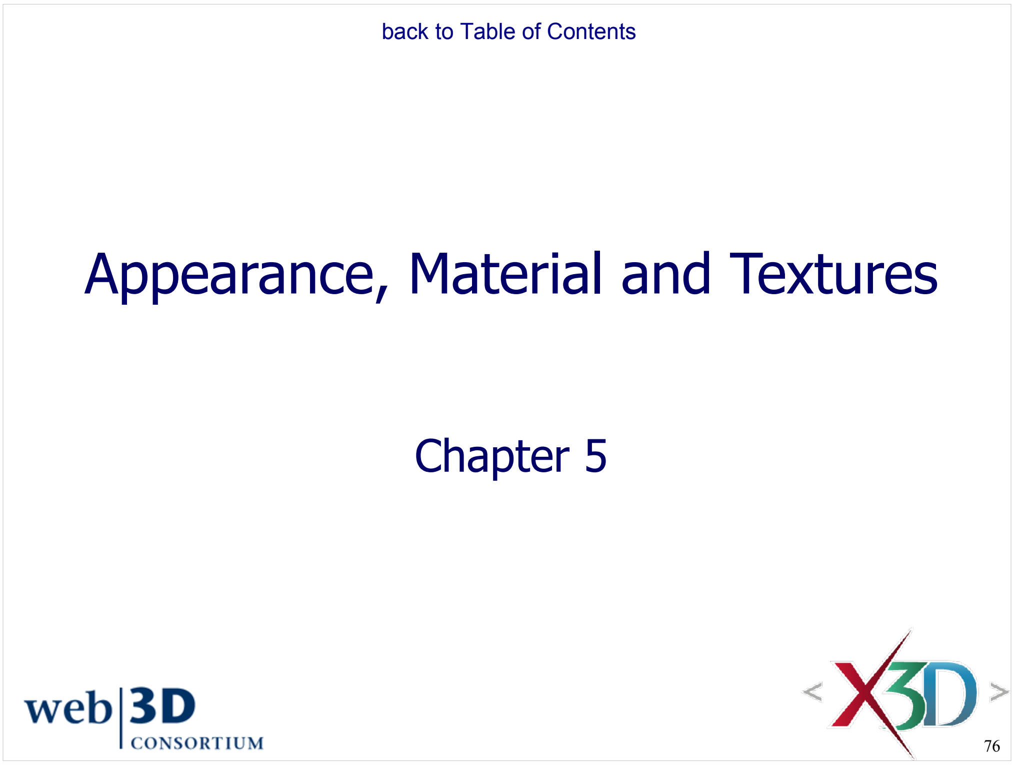
Welcome to the NPS simulation of the
Monterey Bay Aquarium Kelp Forest

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.

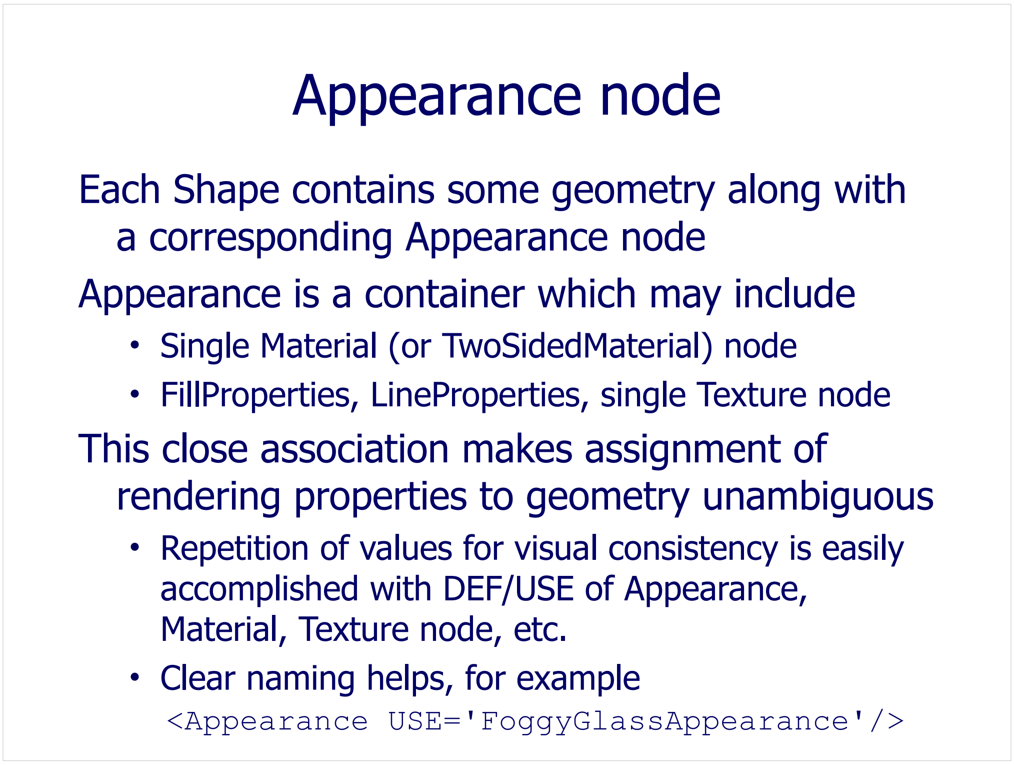
http://www.x3dbook.com/examples/X3dForWebAuthors/KelpForestExhibit/KelpForestMain.x3d



back to Table of Contents

Appearance, Material and Textures

Chapter 5



Appearance node

Each Shape contains some geometry along with a corresponding Appearance node

Appearance is a container which may include

- Single Material (or TwoSidedMaterial) node
- FillProperties, LineProperties, single Texture node

This close association makes assignment of rendering properties to geometry unambiguous

- Repetition of values for visual consistency is easily accomplished with DEF/USE of Appearance, Material, Texture node, etc.
- Clear naming helps, for example
  `<Appearance USE='FoggyGlassAppearance'/>`

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

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

In other words, it is more likely to later say

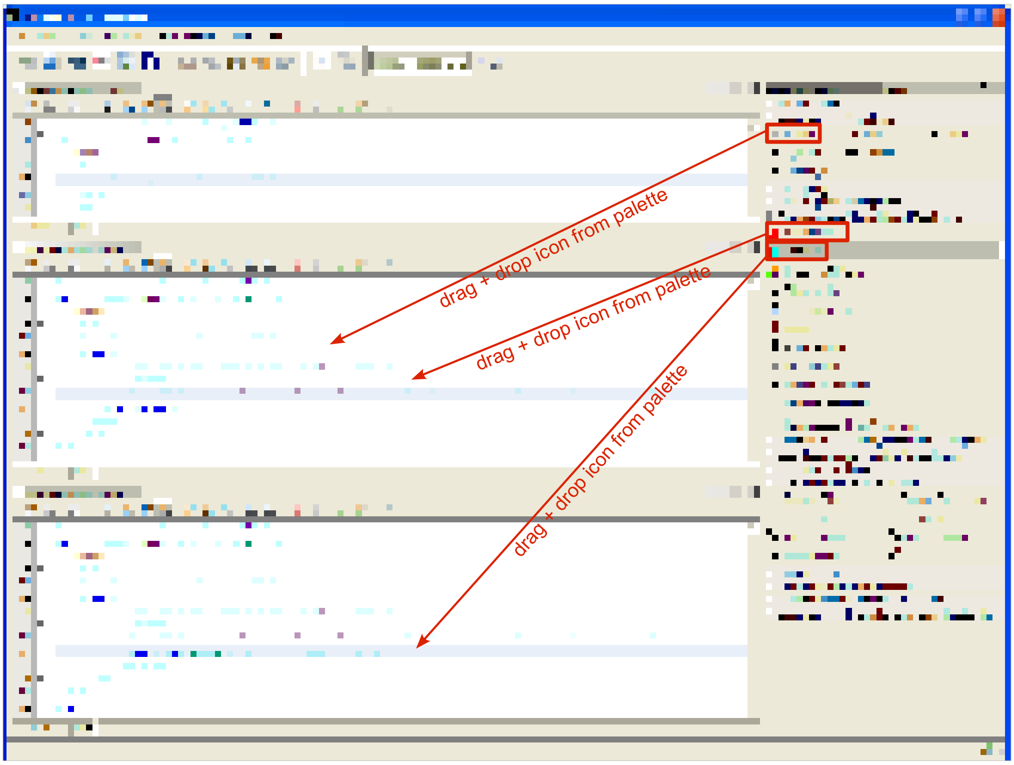
```
<Appearance USE='FoggyGlassAppearance'/>
```

instead of making the node-typing mistake

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

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

Thumbrules on node-naming conventions are given in the X3D Scene Authoring Hints, provided in the X3D-Edit help system and also online at
http://www.web3d.org/x3d/content/examples/X3dSceneAuthoringHints.html#NamingConventions


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

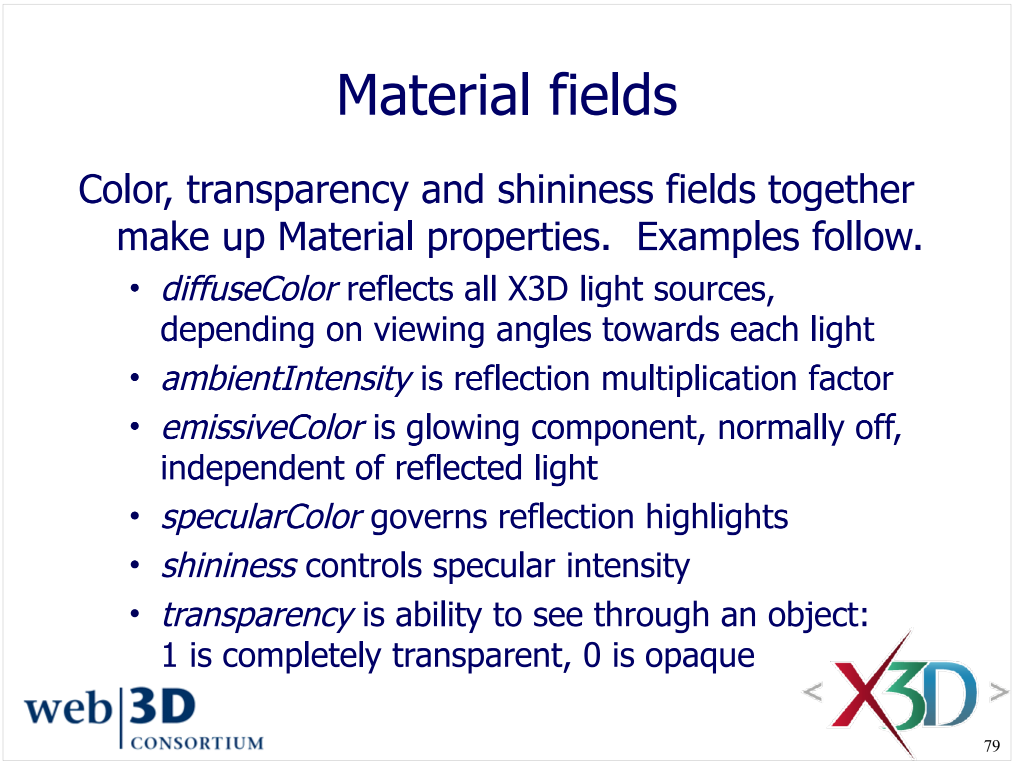
Hint: place the cursor before comments and closing tags, and then press Enter (return key for line feeds), to get proper line spacing and to make the scene easier to read.

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

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

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

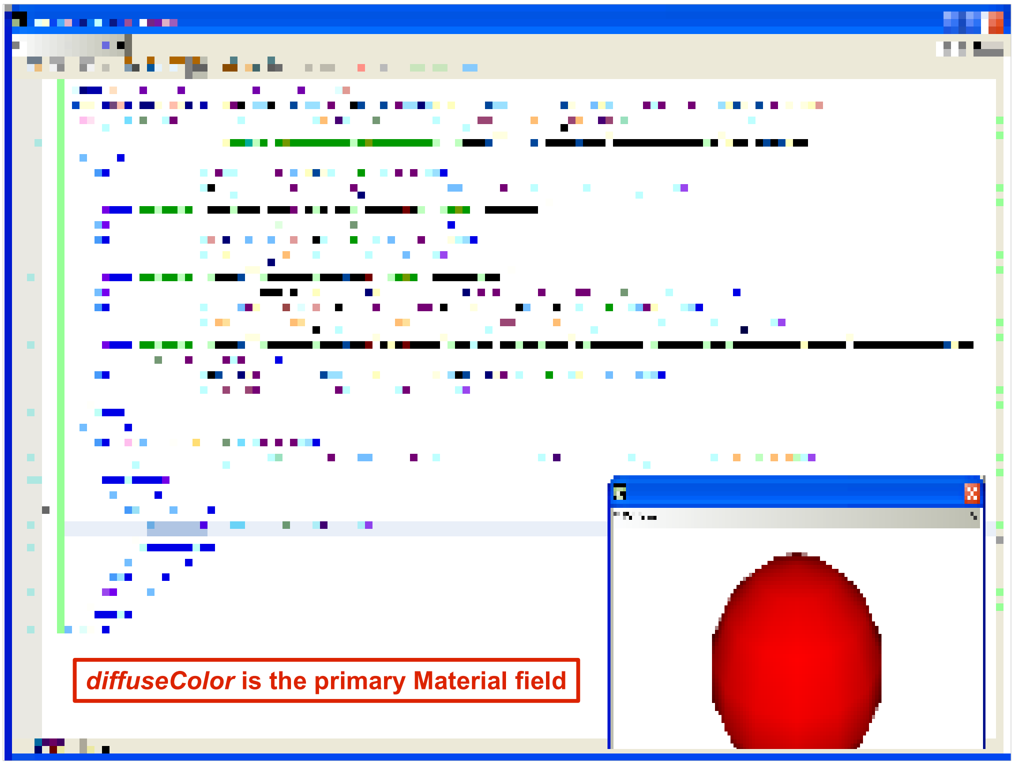
Note that `head` element is iconized and DOCTYPE deleted in these scenes for clarity.



Material fields

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

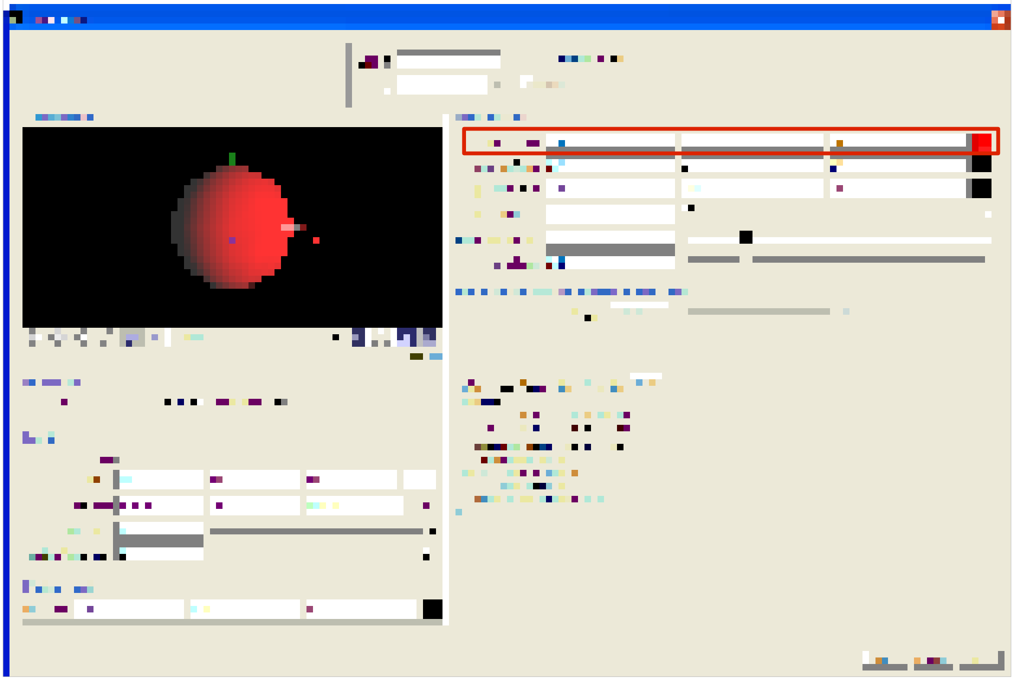
- *diffuseColor* reflects all X3D light sources, depending on viewing angles towards each light
- *ambientIntensity* is reflection multiplication factor
- *emissiveColor* is glowing component, normally off, independent of reflected light
- *specularColor* governs reflection highlights
- *shininess* controls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

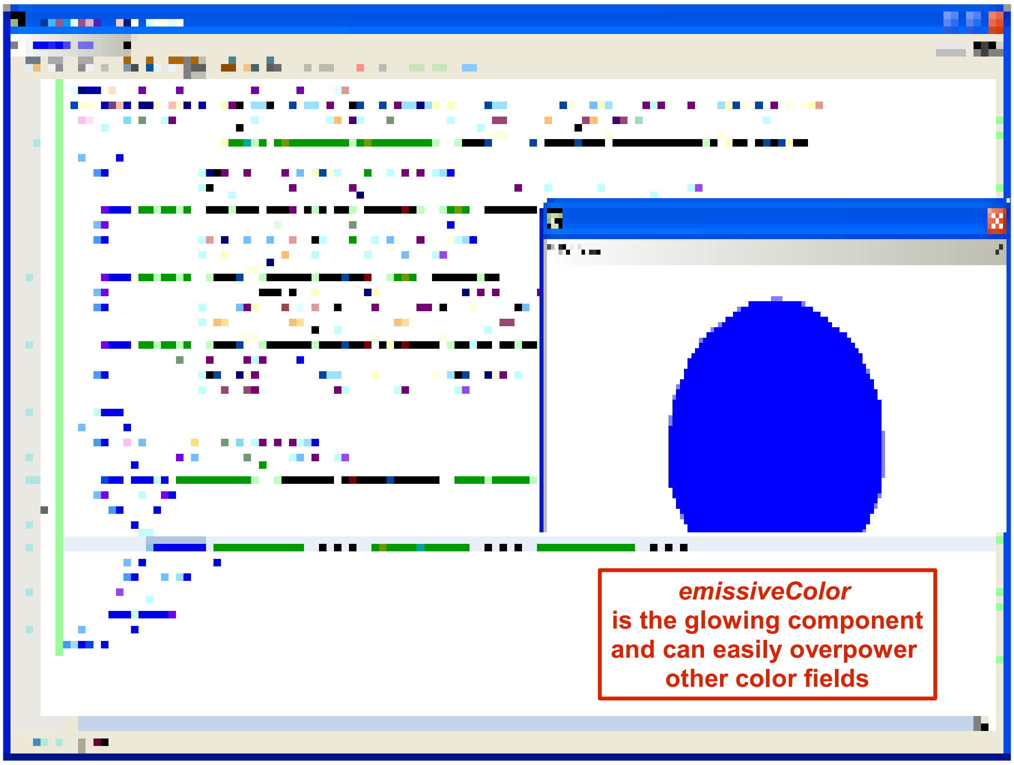


diffuseColor is the primary Material field

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

Material editor: *diffuseColor*





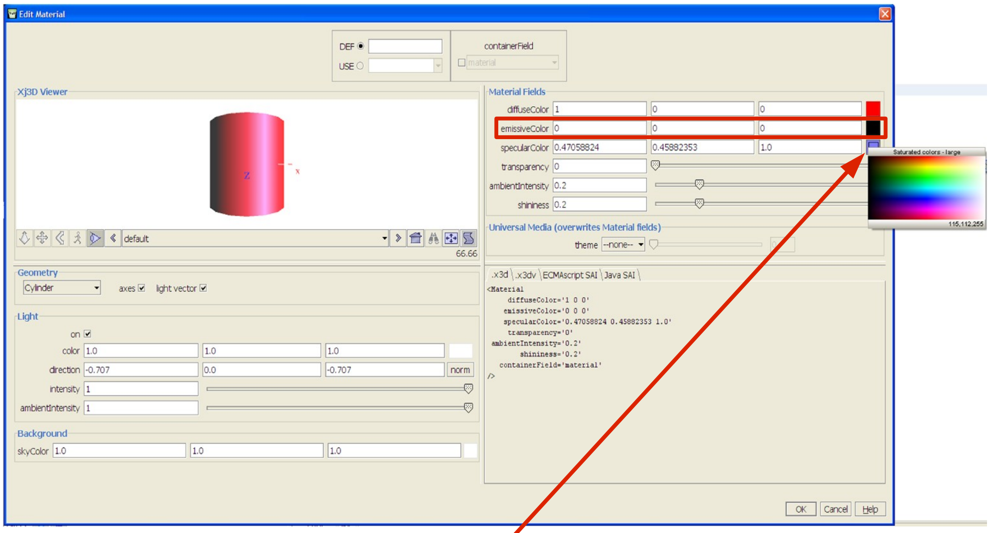
emissiveColor
is the glowing component
and can easily overpower
other color fields

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

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

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

Material editor color selector



DEF ● USE ○

containerField □ material

Material Fields

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

Universal Media (overwrites Material fields)

theme --none--

Geometry

Cylinder axes ☑ light vector ☑

Light

on ☑

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

Background

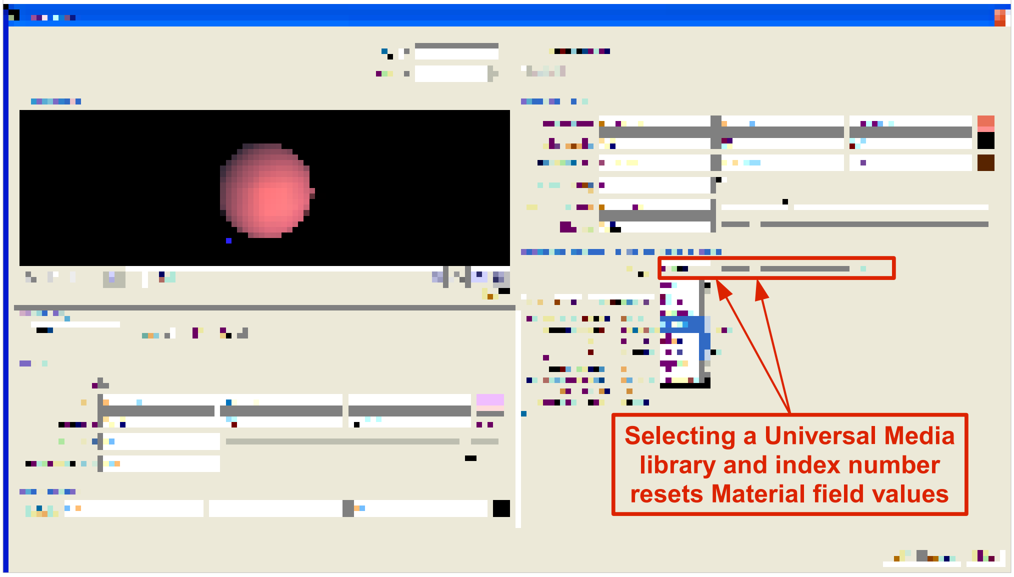
skyColor	1.0	1.0	1.0

.x3d .x3dv ECMAscript SAI Java SAI

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

Click colored box to select a color

Selecting a Universal Material value



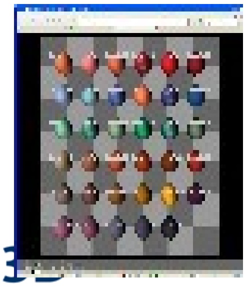
Selecting a Universal Media library and index number resets Material field values

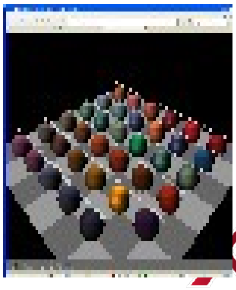
Universal Media libraries include

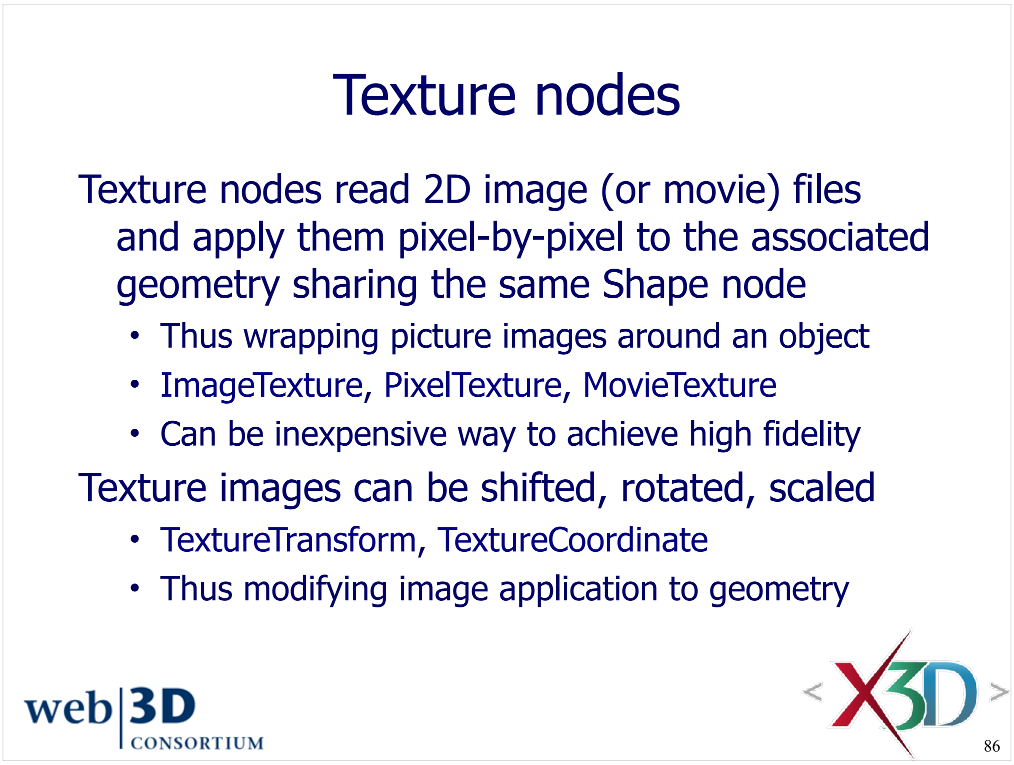
ArtDeco, Autumn, Glass, Metal, Neon, Rococo, SantaFe, Sheen, Silky, Spring, Summer, Tropical, Winter

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

Universal Media screenshots







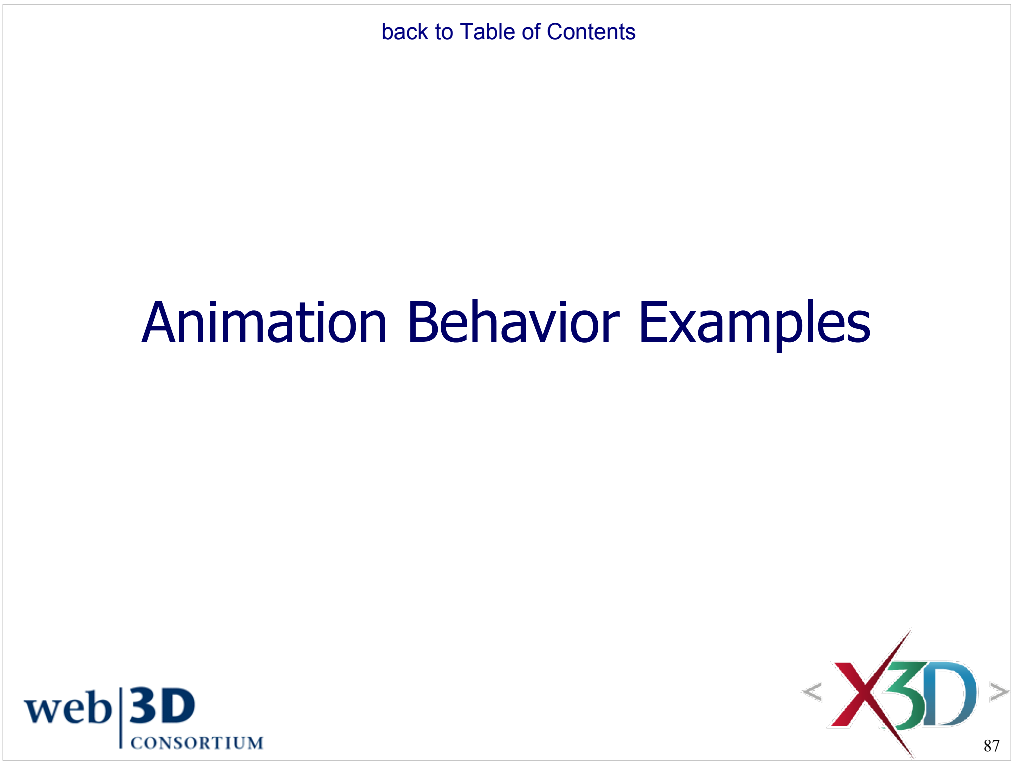
Texture nodes

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

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

Texture images can be shifted, rotated, scaled

- TextureTransform, TextureCoordinate
- Thus modifying image application to geometry

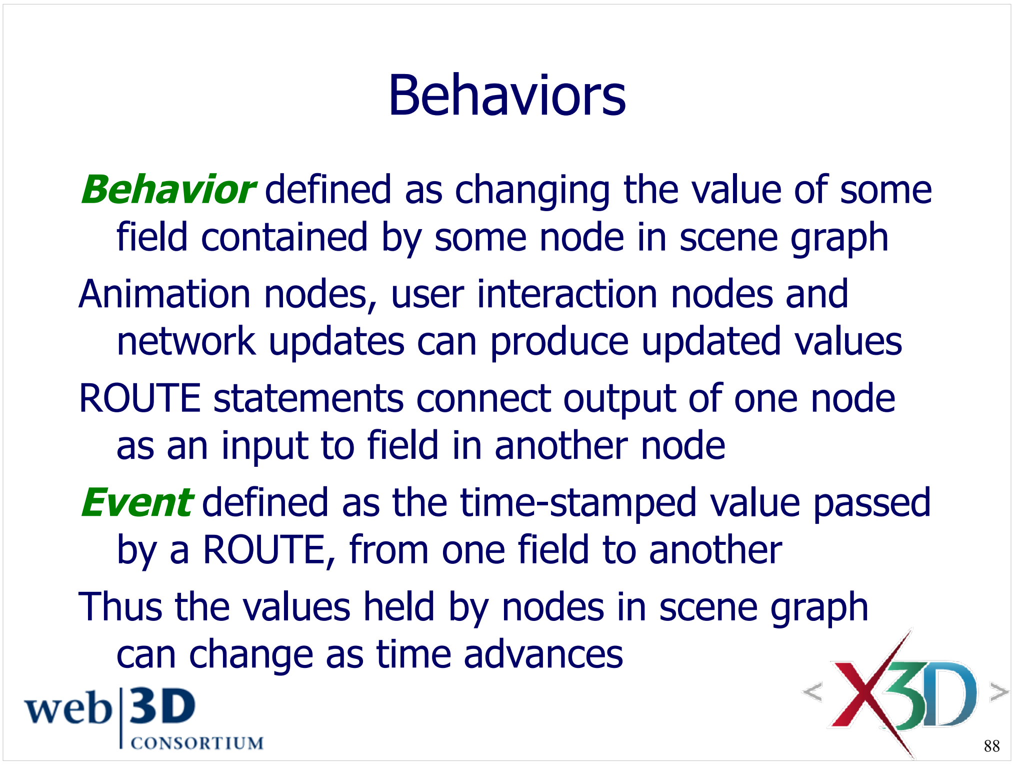


back to Table of Contents

Animation Behavior Examples

Slides from

- Chapter 1, Technical Overview
- Chapter 7, Event Animation and Interpolation
- Chapter 8, User Interaction



Behaviors

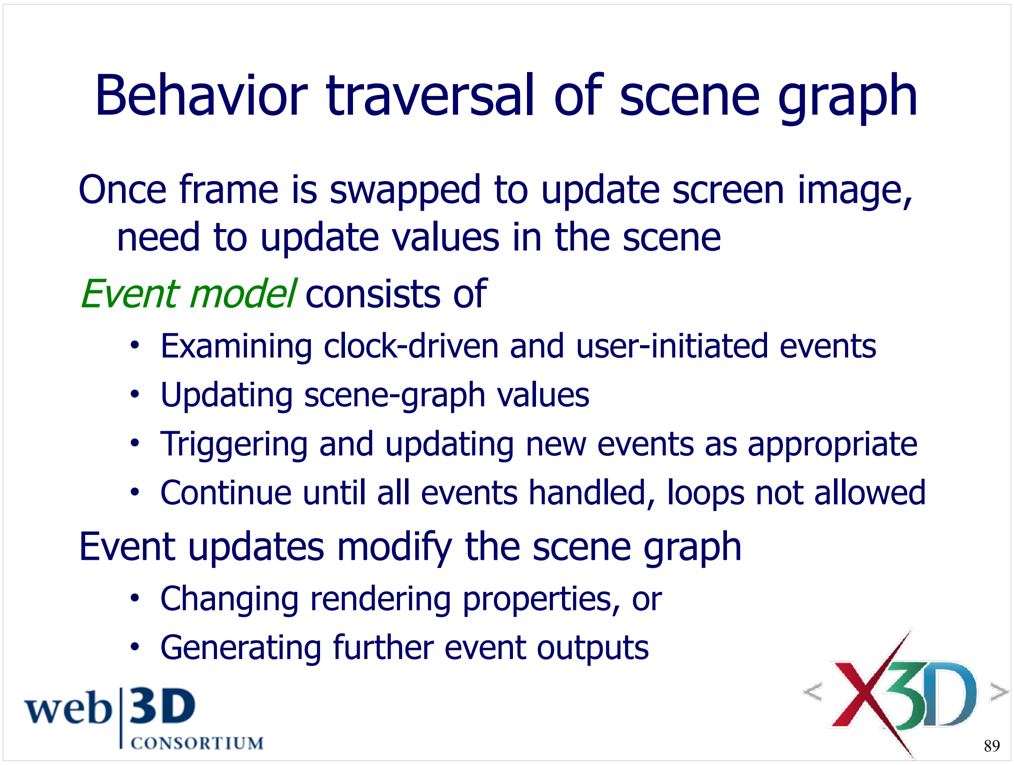
Behavior defined as changing the value of some field contained by some node in scene graph

Animation nodes, user interaction nodes and network updates can produce updated values

ROUTE statements connect output of one node as an input to field in another node

Event defined as the time-stamped value passed by a ROUTE, from one field to another

Thus the values held by nodes in scene graph can change as time advances



Behavior traversal of scene graph

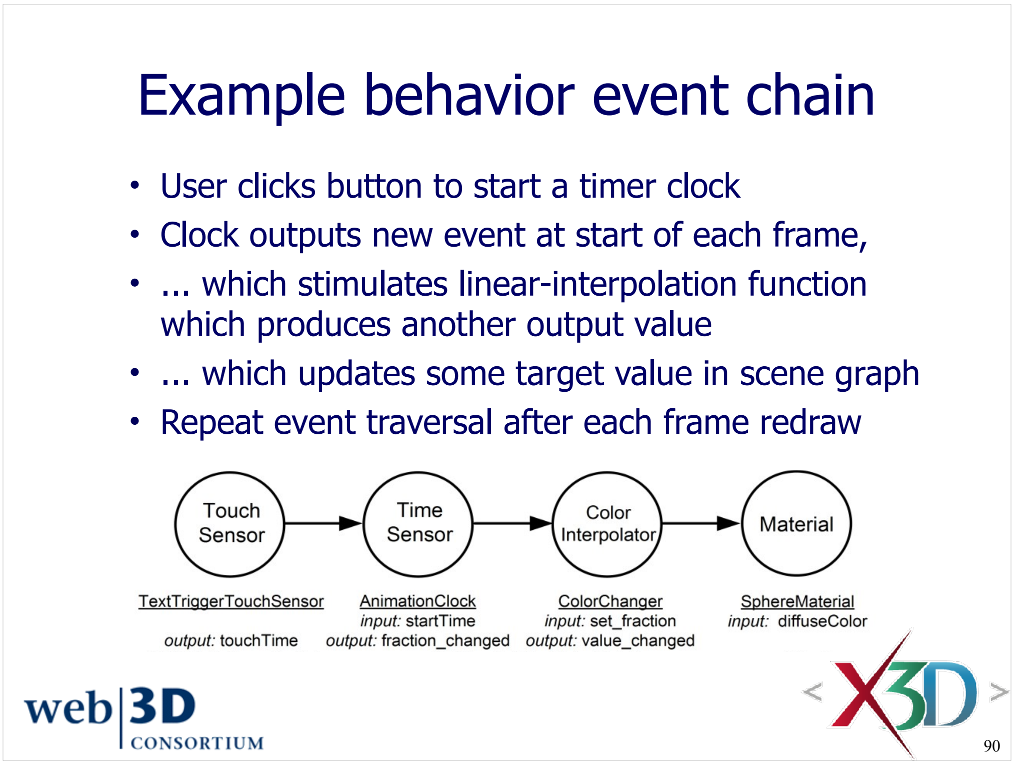
Once frame is swapped to update screen image, need to update values in the scene

Event model consists of

- Examining clock-driven and user-initiated events
- Updating scene-graph values
- Triggering and updating new events as appropriate
- Continue until all events handled, loops not allowed

Event updates modify the scene graph

- Changing rendering properties, or
- Generating further event outputs



Example behavior event chain

- User clicks button to start a timer clock
- Clock outputs new event at start of each frame,
- ... which stimulates linear-interpolation function which produces another output value
- ... which updates some target value in scene graph
- Repeat event traversal after each frame redraw

Touch Sensor → Time Sensor → Color Interpolator → Material

TextTriggerTouchSensor
output: touchTime

AnimationClock
input: startTime
output: fraction_changed

ColorChanger
input: set_fraction
output: value_changed

SphereMaterial
input: diffuseColor

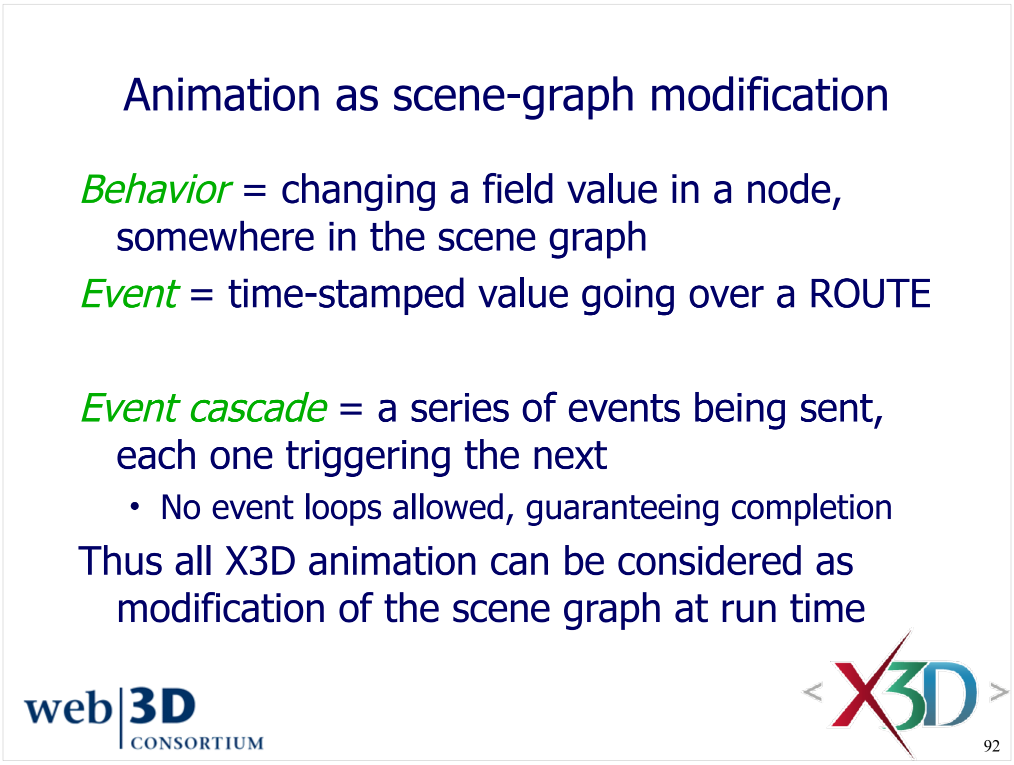
ROUTE connections

ROUTE connection enables the output field of one node to pass a value that then stimulates the input field of another node

- The passed value also includes a time stamp

Field data type and accessType must both match between node/field of source and target

- Chapter 1, Technical Introduction lists field types
- Also provided in tooltips and specification
- Authors usually must carefully check these



Animation as scene-graph modification

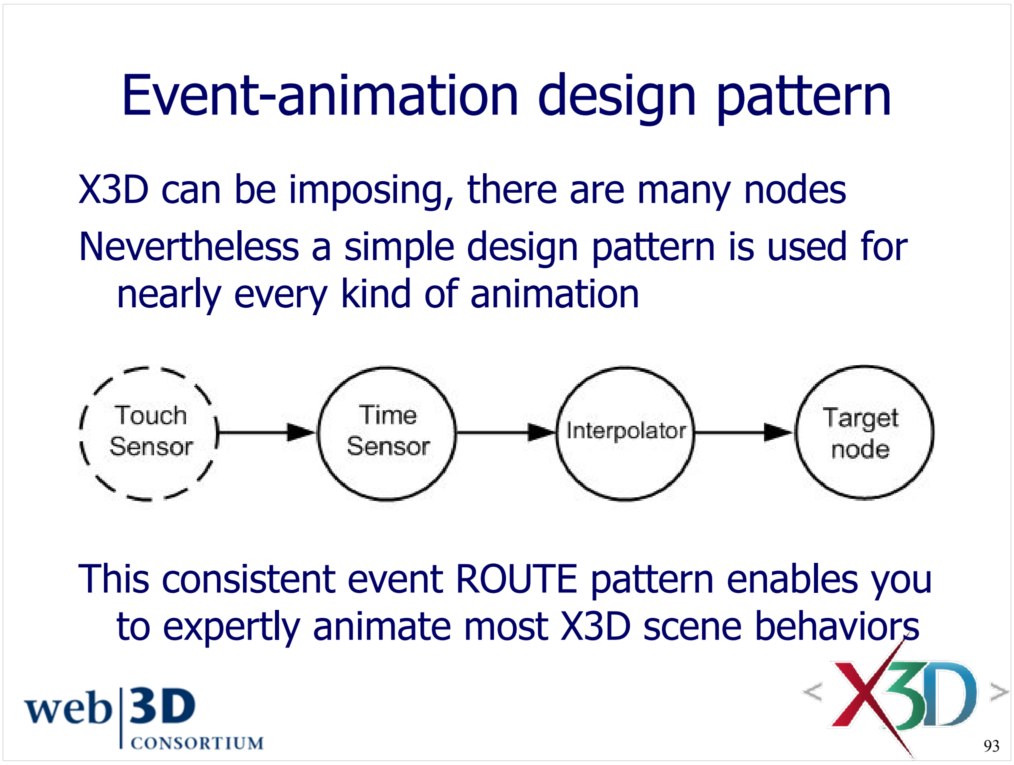
Behavior = changing a field value in a node, somewhere in the scene graph

Event = time-stamped value going over a ROUTE

Event cascade = a series of events being sent, each one triggering the next

- No event loops allowed, guaranteeing completion

Thus all X3D animation can be considered as modification of the scene graph at run time



Event-animation design pattern

- X3D can be imposing, there are many nodes
- Nevertheless a simple design pattern is used for nearly every kind of animation

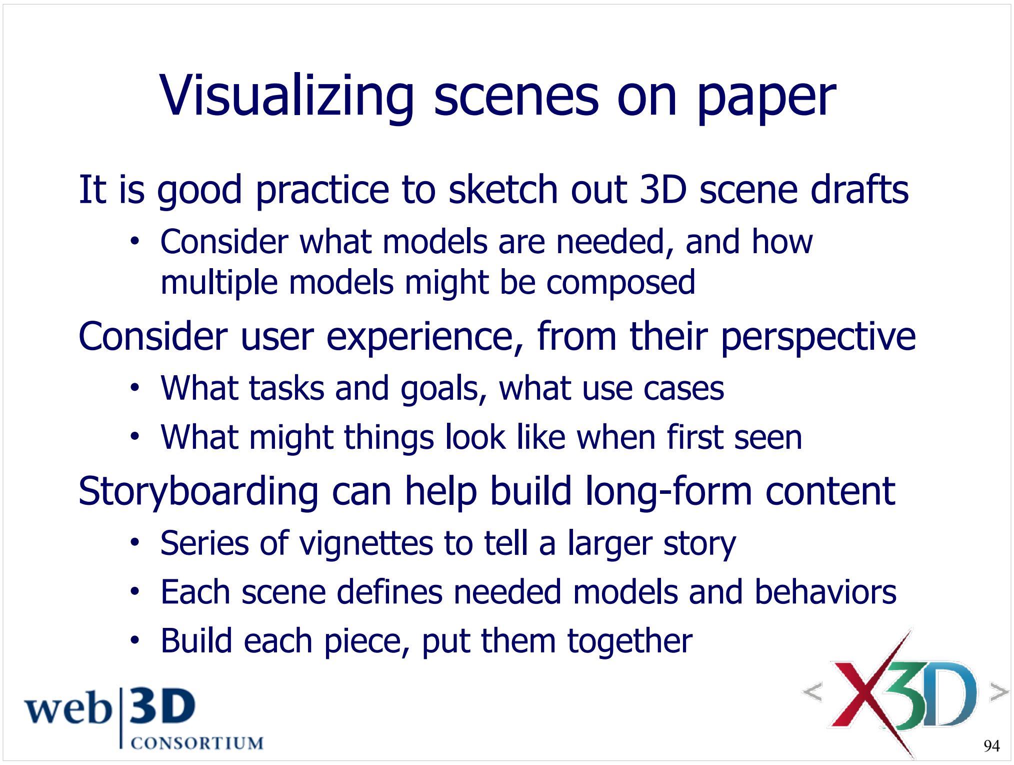
- This consistent event ROUTE pattern enables you to expertly animate most X3D scene behaviors

X3D for Web Authors, Figure 7.1, p. 189.

TouchSensor is optional. Some other triggering event may be provided to start the animation chain, or the TimeSensor may be looping indefinitely.

There are many interpolator nodes. The choice of which interpolator to utilize is determined by the data type of the target field in the target node.

A sequencer node is used instead of an interpolator node if the target field is boolean or integer. Sequencer nodes are described in Chapter 9, Event Utilities and Scripting.



Visualizing scenes on paper

It is good practice to sketch out 3D scene drafts

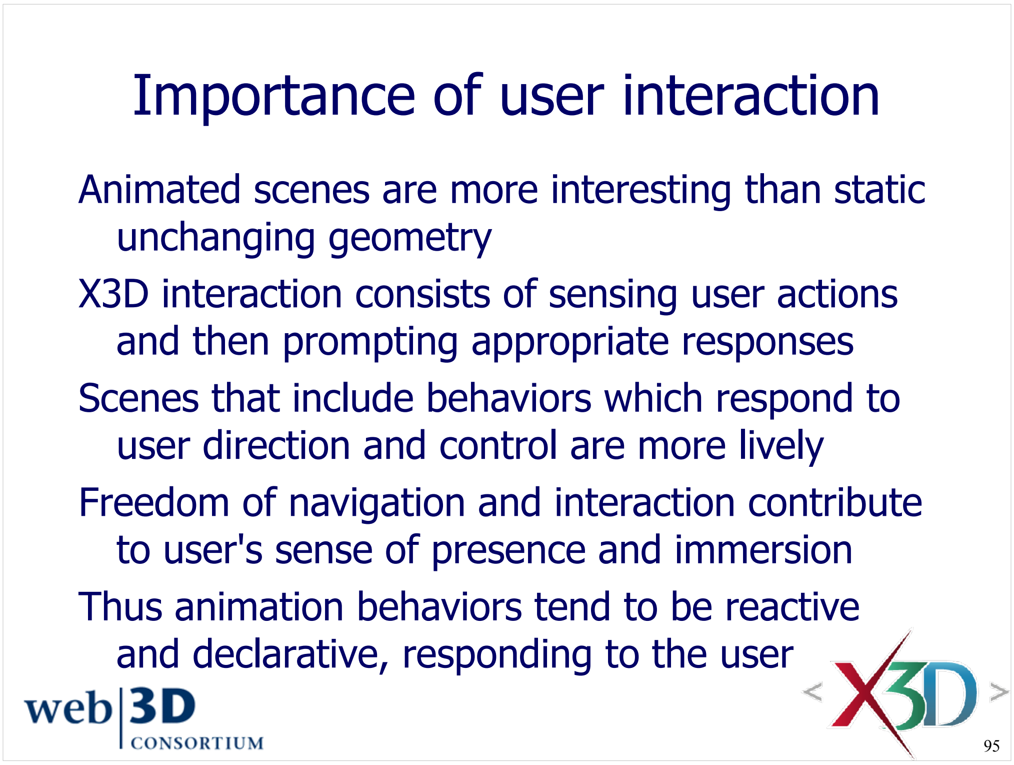
- Consider what models are needed, and how multiple models might be composed

Consider user experience, from their perspective

- What tasks and goals, what use cases
- What might things look like when first seen

Storyboarding can help build long-form content

- Series of vignettes to tell a larger story
- Each scene defines needed models and behaviors
- Build each piece, put them together



Importance of user interaction

Animated scenes are more interesting than static unchanging geometry

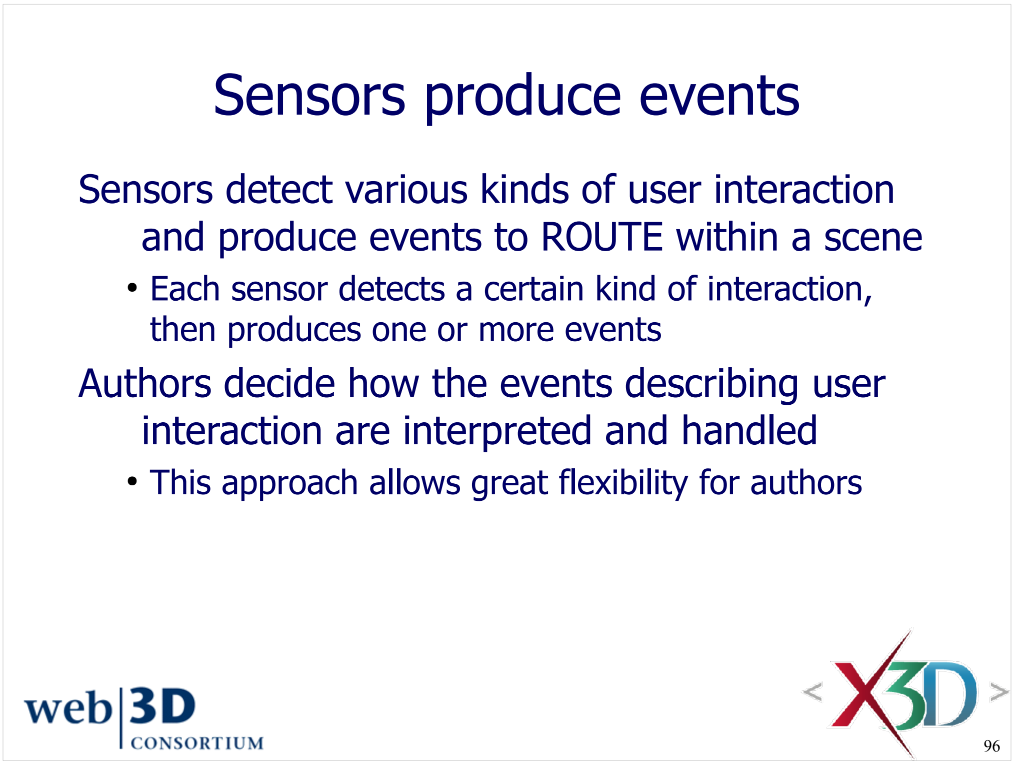
X3D interaction consists of sensing user actions and then prompting appropriate responses

Scenes that include behaviors which respond to user direction and control are more lively

Freedom of navigation and interaction contribute to user's sense of presence and immersion

Thus animation behaviors tend to be reactive and declarative, responding to the user

There is a large body of work in 3D user interaction. See the Additional Resources section.



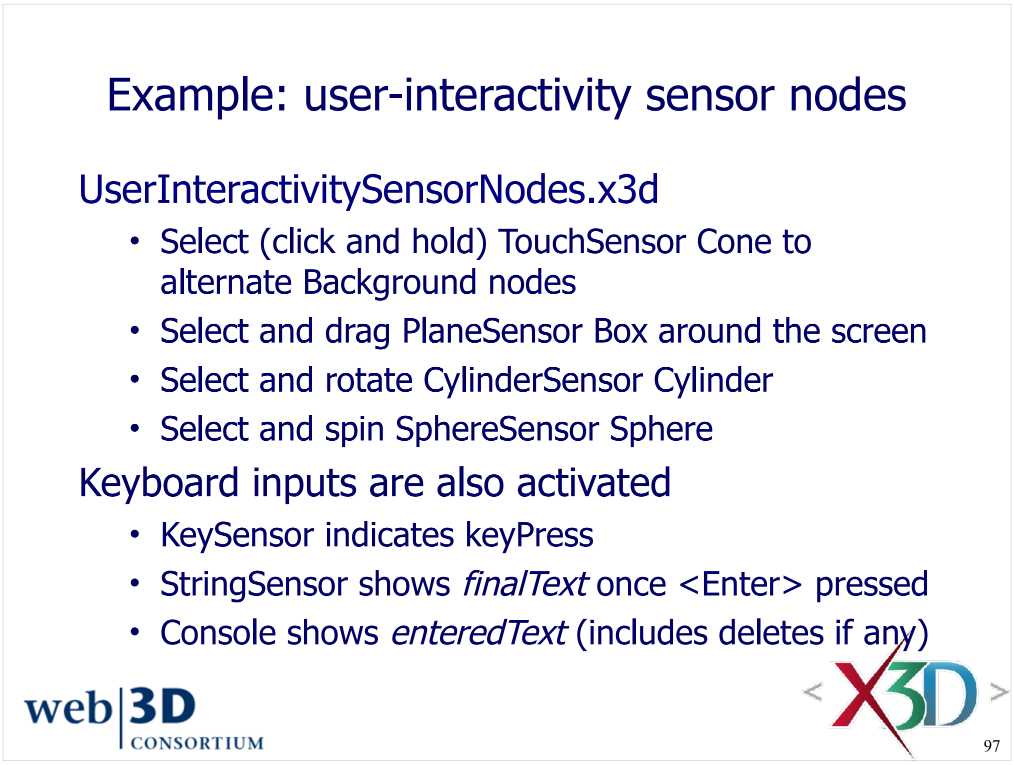
Sensors produce events

Sensors detect various kinds of user interaction and produce events to ROUTE within a scene

- Each sensor detects a certain kind of interaction, then produces one or more events

Authors decide how the events describing user interaction are interpreted and handled

- This approach allows great flexibility for authors



Example: user-interactivity sensor nodes

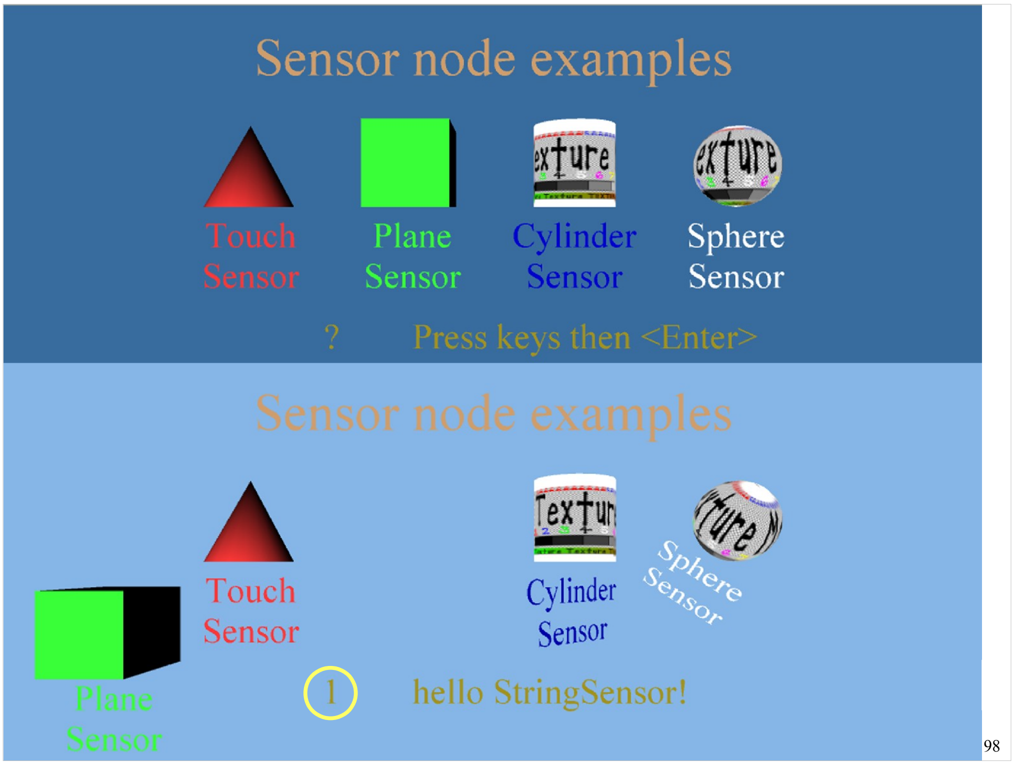
UserInteractivitySensorNodes.x3d

- Select (click and hold) TouchSensor Cone to alternate Background nodes
- Select and drag PlaneSensor Box around the screen
- Select and rotate CylinderSensor Cylinder
- Select and spin SphereSensor Sphere

Keyboard inputs are also activated

- KeySensor indicates keyPress
- StringSensor shows *finalText* once <Enter> pressed
- Console shows *enteredText* (includes deletes if any)

http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter08-UserInteractivity/UserInteractivitySensorNodes.x3d



http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter08-UserInteractivity/UserInteractivitySensorNodes.x3d

The top screen is the initial view. Click and hold to select the Cone TouchSensor that binds the light-blue Background. Releasing unbinds that Background, restoring the original.

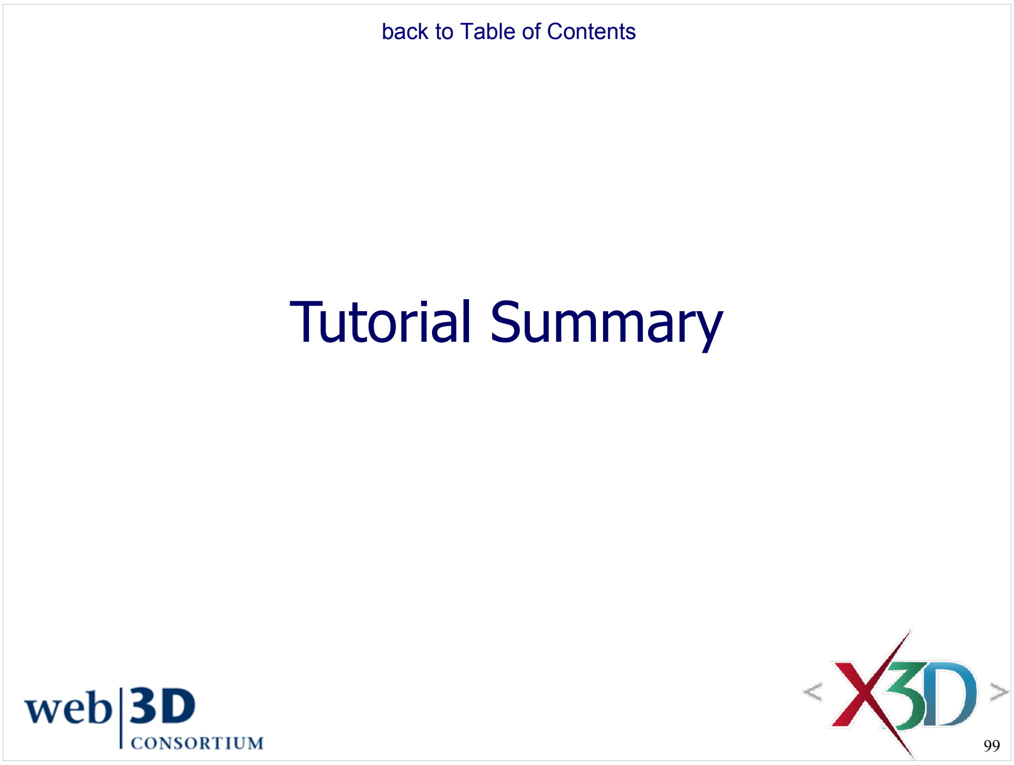
PlaneSensor, CylinderSensor and SphereSensor can each be selected and dragged. Their output values (SFVec3f, SFRotation, SFRotation) have ROUTE connections to either translate or rotate the respective parent Transform node.

Default KeySensor output text is a ? question mark. Note that the key output shows only a capital-letter character (or the primary character) for the key being pressed.

Default StringSensor output text is 'Press keys then <Enter>' - be patient since the *finalText* field doesn't send an output string until the <Enter> key is pressed.

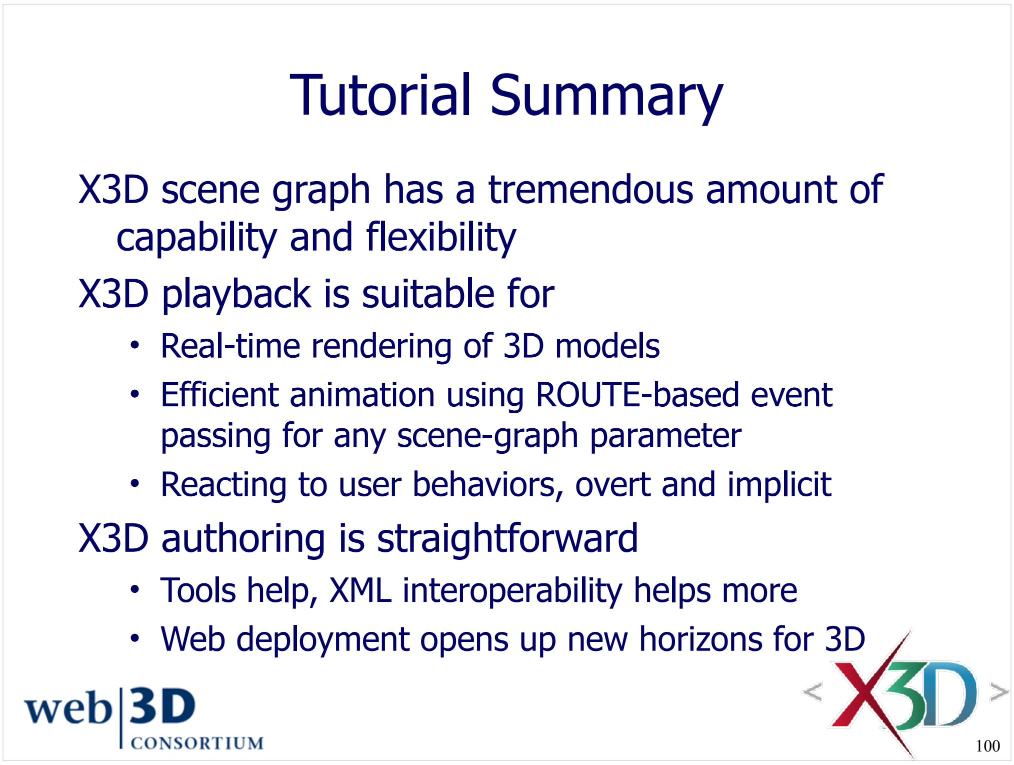
The console shows the *enteredText*, as it is typed key by key, including <Backspace> or <Delete> effects (if any).

```
enteredText=H               enteredText=Hello Strin
enteredText=He              enteredText=Hello String
enteredText=Hel             enteredText=Hello StringS
enteredText=Hell            enteredText=Hello StringSe
enteredText=Hello           enteredText=Hello StringSen
enteredText=Hello           enteredText=Hello StringSens
enteredText=Hello S         enteredText=Hello StringSenso
enteredText=Hello St        enteredText=Hello StringSensor
enteredText=Hello Str       enteredText=Hello StringSensor!
enteredText=Hello Stri      enteredText=Hello StringSensor!
```



back to Table of Contents

Tutorial Summary



Tutorial Summary

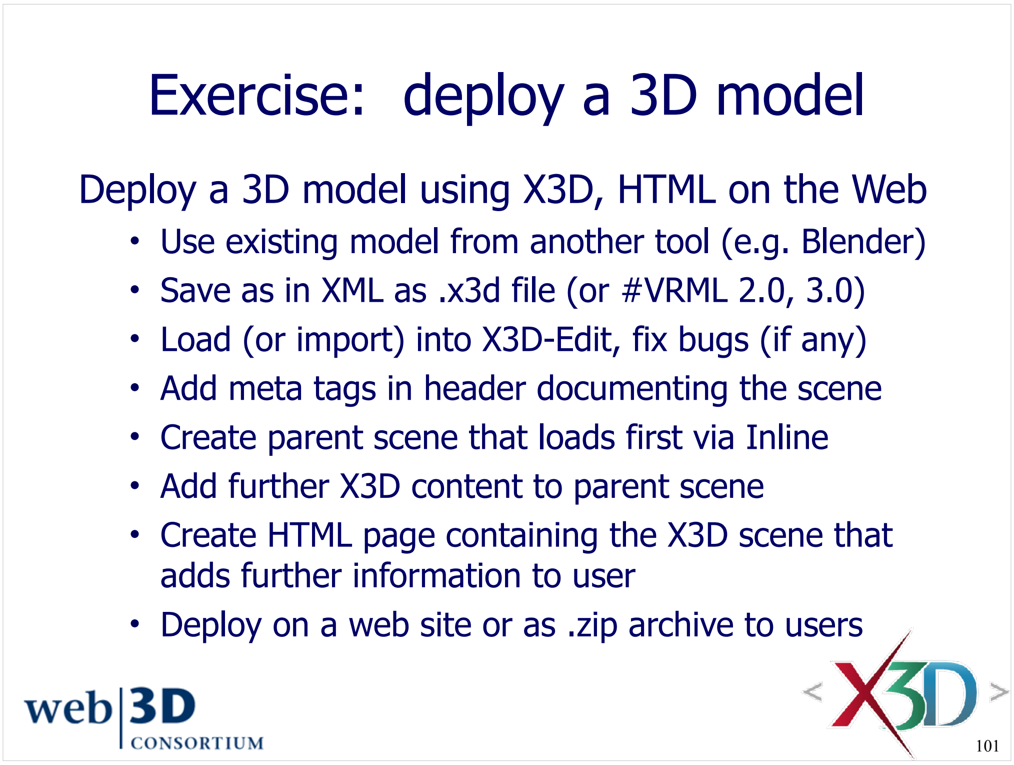
X3D scene graph has a tremendous amount of capability and flexibility

X3D playback is suitable for

- Real-time rendering of 3D models
- Efficient animation using ROUTE-based event passing for any scene-graph parameter
- Reacting to user behaviors, overt and implicit

X3D authoring is straightforward

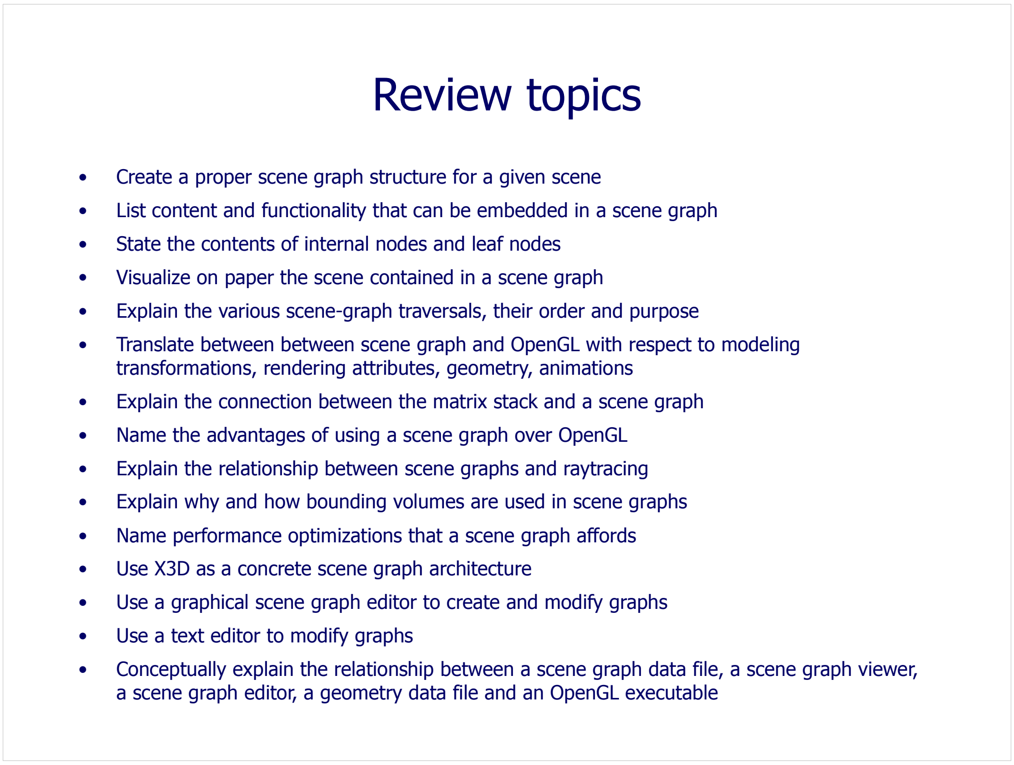
- Tools help, XML interoperability helps more
- Web deployment opens up new horizons for 3D



Exercise: deploy a 3D model

Deploy a 3D model using X3D, HTML on the Web

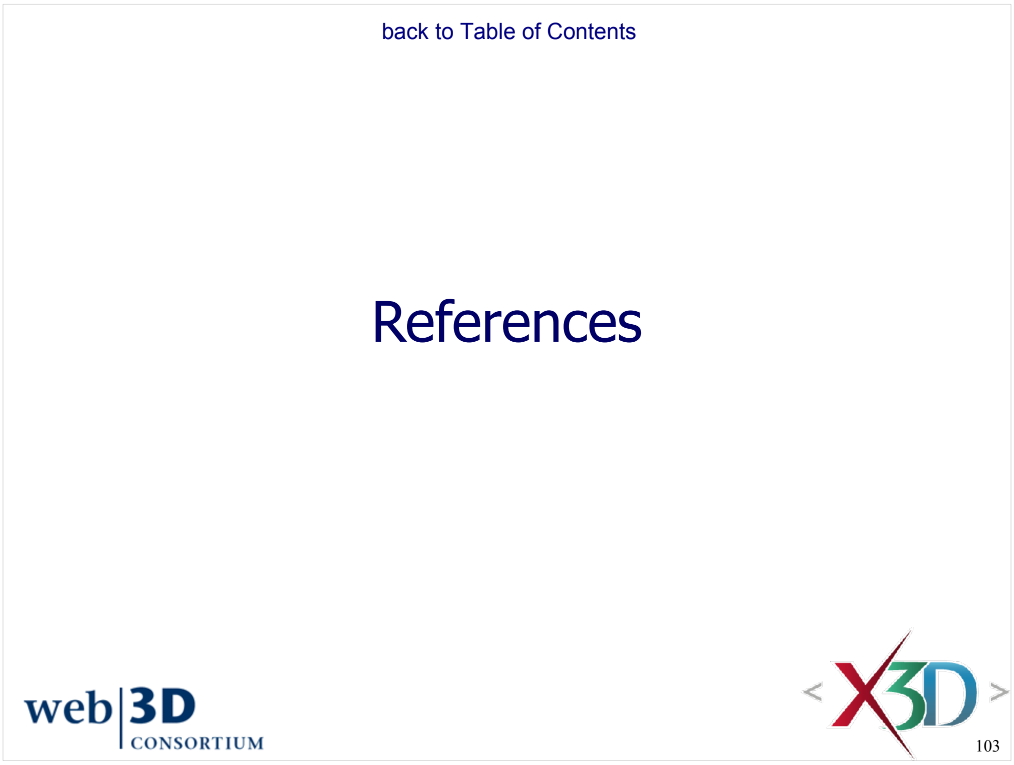
- Use existing model from another tool (e.g. Blender)
- Save as in XML as .x3d file (or #VRML 2.0, 3.0)
- Load (or import) into X3D-Edit, fix bugs (if any)
- Add meta tags in header documenting the scene
- Create parent scene that loads first via Inline
- Add further X3D content to parent scene
- Create HTML page containing the X3D scene that adds further information to user
- Deploy on a web site or as .zip archive to users



Review topics

- Create a proper scene graph structure for a given scene
- List content and functionality that can be embedded in a scene graph
- State the contents of internal nodes and leaf nodes
- Visualize on paper the scene contained in a scene graph
- Explain the various scene-graph traversals, their order and purpose
- Translate between between scene graph and OpenGL with respect to modeling transformations, rendering attributes, geometry, animations
- Explain the connection between the matrix stack and a scene graph
- Name the advantages of using a scene graph over OpenGL
- Explain the relationship between scene graphs and raytracing
- Explain why and how bounding volumes are used in scene graphs
- Name performance optimizations that a scene graph affords
- Use X3D as a concrete scene graph architecture
- Use a graphical scene graph editor to create and modify graphs
- Use a text editor to modify graphs
- Conceptually explain the relationship between a scene graph data file, a scene graph viewer, a scene graph editor, a geometry data file and an OpenGL executable

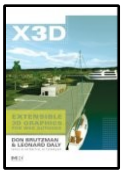
With thanks to Dr. Mathias Kolsch NPS for these guiding questions to support a MV3202 course tutorial.



back to Table of Contents

References

References 1

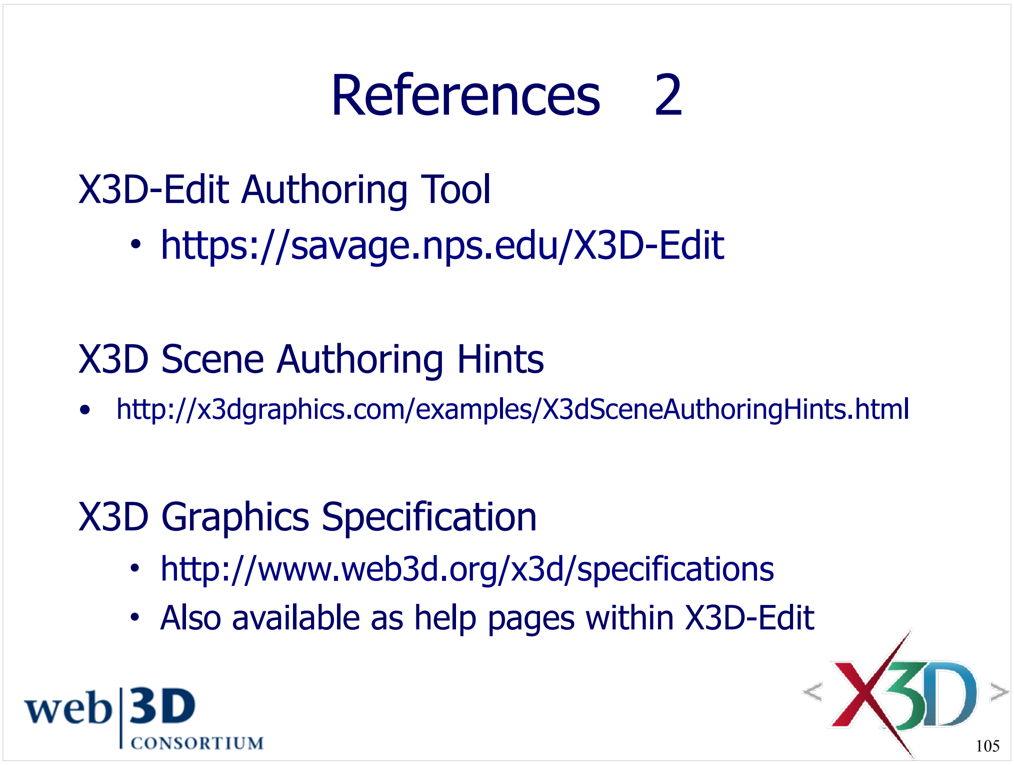


X3D: Extensible 3D Graphics for Web Authors by Don Brutzman and Leonard Daly, Morgan Kaufmann Publishers, April 2007, 468 pages.

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

X3D Resources

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



References 2

X3D-Edit Authoring Tool

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

X3D Scene Authoring Hints

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

X3D Graphics Specification

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

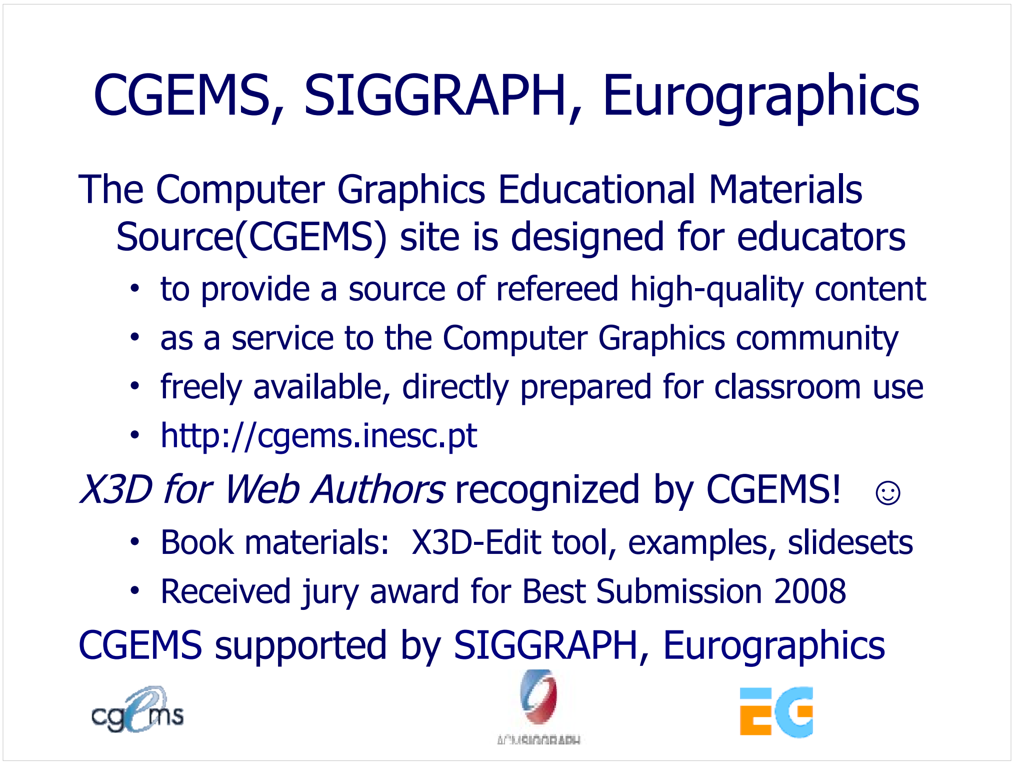
Contact

Don Brutzman

brutzman@nps.edu

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

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.2149 voice
1.831.656.7599 fax



CGEMS, SIGGRAPH, Eurographics

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

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

X3D for Web Authors recognized by CGEMS! ☺

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

CGEMS supported by SIGGRAPH, Eurographics

From the CGEMS home page:

- http://cgems.inesc.pt

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

List of all published modules:

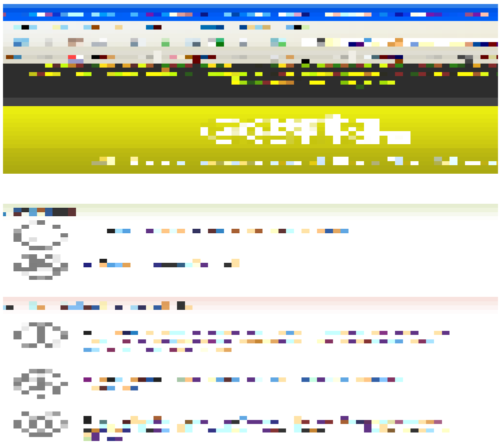
- http://cgems.inesc.pt/authors/ListModules.aspx

CGEMS Editorial Policy:

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

Creative Commons open-source license

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



Attribution-Noncommercial-Share Alike 3.0 Unported

You are free:

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

* to Remix — to adapt the work

Under the following conditions:

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

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

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

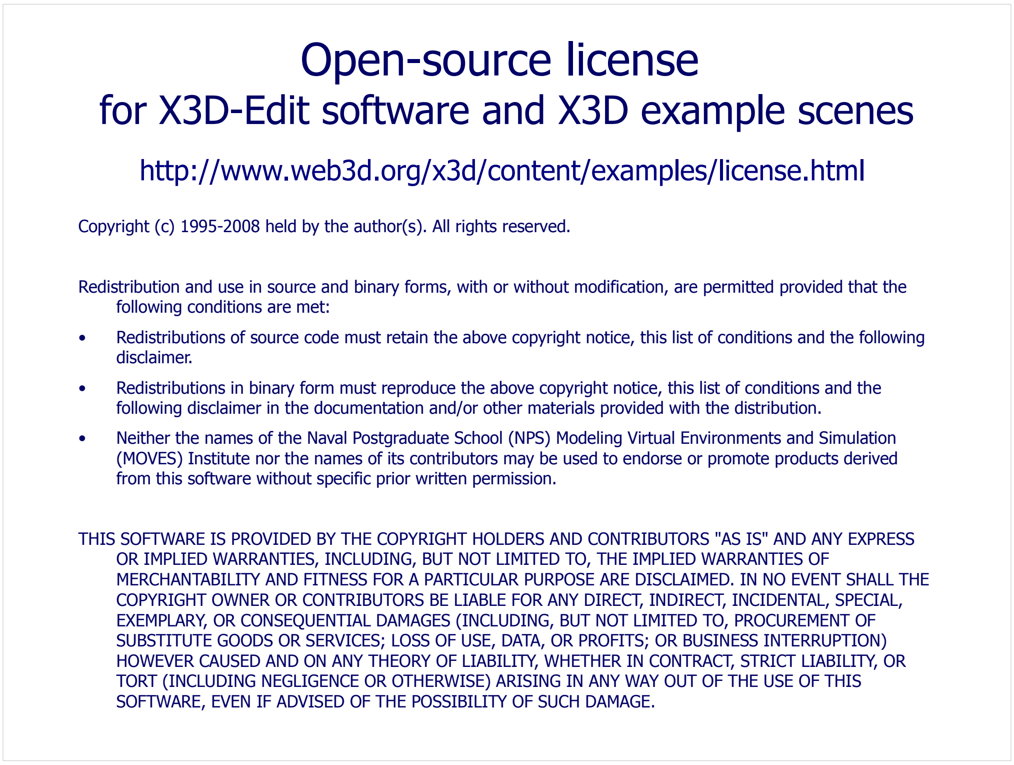
* Noncommercial. You may not use this work for commercial purposes.

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

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

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

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



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

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

Copyright (c) 1995-2008 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.

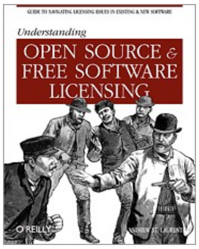
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE 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.

License available at

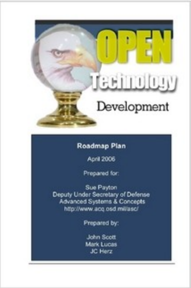
http://www.web3d.org/x3d/content/examples/license.txt

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

Good references on open source:



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



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