



18

Introduction to VRML97

Organizer

David Nadeau

San Diego Supercomputer Center

Lecturers

Michael Heck

Template Graphics Software, Inc.

John Moreland

David Nadeau

San Diego Supercomputer Center

25th International Conference on Computer Graphics and Interactive Techniques

Exhibition **21-23 July** 1998 Conference **19-24 July** 1998

Orlando, Florida USA

course notes

SIGGRAPH 98

Introduction to VRML 97

Lecturers

David R. Nadeau

nadeau@sdsc.edu
<http://www.sdsc.edu/~nadeau>
San Diego Supercomputer Center
University of California at San Diego

John L. Moreland

moreland@sdsc.edu
<http://www.sdsc.edu/~moreland>
San Diego Supercomputer Center
University of California at San Diego

Michael M. Heck

mmh@tgs.com
Template Graphics Software

Tutorial notes sections

Abstract

Preface

Lecturer information

Using the VRML examples

Using the JavaScript examples

Using the Java examples

Tutorial slides

Introduction to VRML 97

Abstract

VRML (the Virtual Reality Modeling Language) is an international standard for describing 3-D shapes and scenery on the World Wide Web. VRML's technology has very broad applicability, including web-based entertainment, distributed visualization, 3-D user interfaces to remote web resources, 3-D collaborative environments, interactive simulations for education, virtual museums, virtual retail spaces, and more. VRML is a key technology shaping the future of the web.

Participants in this tutorial will learn how to use VRML 97 (a.k.a. *ISO VRML*, *VRML 2.0*, and *Moving Worlds*) to author their own 3-D virtual worlds on the World Wide Web. Participants will learn VRML concepts and terminology, and be introduced to VRML's text format syntax. Participants also will learn tips and techniques for increasing performance and realism. The tutorial includes numerous VRML examples and information on where to find out more about VRML features and use.

Preface

Welcome to the *Introduction to VRML 97* tutorial notes! These tutorial notes have been written to give you a quick, practical, example-driven overview of *VRML 97*, the Web's Virtual Reality Modeling Language. Included are over 500 pages of tutorial material with nearly 200 images and over 100 VRML examples.

To use these tutorial notes you will need an HTML Web browser with support for viewing VRML worlds. An up to date list of available VRML browsing and authoring software is available at:

The VRML Repository
(<http://vrml.sdsc.edu>)

What's included in these notes

These tutorial notes primarily contain two types of information:

1. General information, such as this preface
2. Tutorial slides and examples

The tutorial slides are arranged as a sequence of 500+ hyper-linked pages containing VRML syntax notes, VRML usage comments, or images of sample VRML worlds. Clicking on a sample world's image loads the VRML world into your browser for you to examine yourself.

Clicking on a sample world's file name, shown underneath the image, loads into your browser a text page showing the VRML code itself. Using these links, or text editor, you can view the VRML code and see how a particular effect is created. In most cases, the VRML files contain extensive comments providing information about the techniques the file illustrates.

The tutorial notes provide a necessarily terse overview of VRML. It is recommended that you invest in one of the VRML books on the market to get a more thorough coverage of the language. The book we recommend is one we co-authored:

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

Several other good VRML books are on the market as well.

A word about VRML versions

VRML has evolved through several versions of the language, starting way back in late 1994. These tutorial notes cover *VRML 97*, the latest version of the language. To provide context, the following table provides a quick overview of these VRML versions and the names they have become known by.

Version/Released	Comments
VRML 1.0 May 1995	<p>Begun in late 1994, the first version of VRML was largely based upon the <i>Open Inventor</i> file format developed by Silicon Graphics Inc. (SGI). The VRML 1.0 specification was completed in May 1995 and included support for shape building, lighting, and texturing.</p> <p>VRML 1.0 browser plug-ins became widely available by late 1995, though few ever supported the full range of features defined by the VRML 1.0 specification.</p>
VRML 1.0c January 1996	<p>As vendors began producing VRML 1.0 browsers, a number of ambiguities in the VRML 1.0 specification surfaced. These problems were corrected in a new VRML 1.0c (clarified) specification released in January 1996. No new features were added to the language in VRML 1.0c.</p>
VRML 1.1 canceled	<p>In late 1995, discussion began on extensions to the VRML 1.0 specification. These extensions were intended to address language features that made browser implementation difficult or inefficient. The extended language was tentatively dubbed VRML 1.1. These enhancements were later dropped in favor of forging ahead on VRML 2.0 instead.</p> <p>No VRML 1.1 browsers exist.</p>
Moving Worlds January 1996	<p>VRML 1.0 included features for building static, unchanging worlds suitable for architectural walk-throughs and some scientific visualization applications. To extend the language to support animation and interaction, the VRML architecture group made a call for proposals for a language redesign. Silicon Graphics, Netscape, and others worked together to create the <i>Moving Worlds</i> proposal, submitted in January 1996. That proposal was later accepted and became the starting point for developing VRML 2.0. The final VRML 2.0 language specification is still sometimes referred to as the <i>Moving Worlds</i> specification, though it differs significantly from the original <i>Moving Worlds</i> proposal.</p>
VRML 2.0 August 1996	<p>After seven months of intense effort by the VRML community, the <i>Moving Worlds</i> proposal evolved to become the final VRML 2.0 specification, released in August 1996. The new specification redesigned the VRML syntax and added an extensive set of new features for shape building, animation, interaction, sound, fog, backgrounds, and language extensions.</p> <p>While multiple VRML 2.0 browsers exist today, as of this writing, none are <i>complete</i>. All of the browsers are missing a few features. Fortunately, most of the missing features are obscure aspects of VRML.</p>
VRML 97	<p>In early 1997, efforts got under way to present the VRML 2.0 specification</p>

December 1997 to the International Standards Organization (ISO) which oversees most of the major language specifications in use in the computing community. The ISO version of VRML 2.0 was reviewed and the specification significantly rewritten to clarify issues. A few minor changes to the language were also made. The final ISO VRML was dubbed *VRML 97*. The VRML 97 specification features finalized in March 1997 and its explanatory text finalized in September 1997. This specification was ratified by ISO in December 1997.

Most major VRML 2.0 browsers are now VRML 97 browsers.

VRML 1.0 and VRML 2.0 differ radically in syntax and features. A VRML 1.0 browser cannot display VRML 2.0 worlds. Most VRML 2.0 browsers, however, can display VRML 1.0 worlds.

VRML 97 differs in a few minor ways from VRML 2.0. In most cases, a VRML 2.0 browser will be able to correctly display VRML 97 files. However, for 100% accuracy, you should have a VRML 97 compliant browser for viewing the VRML files contained within these tutorial notes.

How these tutorial notes were created

These tutorial notes were developed and tested on a PC with a Diamond Multimedia FireGL 1000 3D accelerator card, and on a Silicon Graphics High Impact UNIX workstation. HTML and VRML text was hand-authored using a text editor. In some cases Perl and C programs were used to automatically generate smooth surfaces and animation paths.

A Perl script, called `mktalk`, developed by John Moreland, was used to process raw tutorial notes text and produce the 500+ individual HTML files, one per tutorial slide.

HTML text was displayed using Netscape Navigator 4.04 on Silicon Graphics and PC systems and Microsoft Internet Explorer 4.01 on PC systems. Colors were checked for viewability in 24-bit, 16-bit, and 8-bit display modes on a PC. Text sizes were chosen for viewability at a normal 12 point font on-screen, and at an 24 point font for presentation during the tutorial. The large text, white-on-black colors, and terse language are used to insure that slides are readable when displayed for the tutorial audience.

VRML worlds were displayed on Silicon Graphics systems using the Silicon Graphics Cosmo Player 1.02 VRML 97 compliant browser for Netscape Navigator. The same worlds were displayed on PC systems using Silicon Graphics Cosmo Player 2.0 for Netscape Navigator and Microsoft Internet Explorer.

Texture images were created using Adobe PhotoShop 4.0 on a PC with help from KAI's PowerTools 3.0 from MetaTools. Image processing was also performed using the Image Tools suite of applications for UNIX workstations from the San Diego Supercomputer Center.

PDF tutorial notes for printing were created by dumping individual tutorial slides to PostScript on a Silicon Graphics workstation. The PostScript was transferred to a PC where it was converted to PDF and assembled into a single PDF file using Adobe's Distiller and Exchange.

Use of these tutorial notes

Can you use these tutorial notes for your own purposes? The answer is:

Parts of these tutorial notes are copyright (c) 1997 by David R. Nadeau) 1997 John L. Moreland, and (c) 1997 Michael M. Heck. Users and possessors of these tutorial notes are hereby granted a nonexclusive, royalty-free copyright and design patent license to use this material in individual applications. License is not granted for commercial resale, in whole or in part, without prior written permission from the authors. This material is provided "AS IS" without express or implied warranty of any kind.

You are free to use these tutorial notes in whole or in part to help you teach your own VRML tutorial. You may translate these notes into other languages and you may post copies of these notes on your own Web site, as long as the above copyright notice is included as well. You may not, however, sell these tutorial notes for profit or include them on a CD-ROM or other media product without written permission.

If you use these tutorial notes, please:

1. Give credit for the original material
2. Tell us since we like hearing about the use of the material!

If you find bugs in the notes, please tell us. We have worked hard to try and make the notes bug-free, but if something slipped by, we'd like to fix it before others are confused by the mistake.

Contact

For bug reports, comments, and questions, please contact:

David R. Nadeau

San Diego Supercomputer Center
P.O. Box 85608
San Diego, CA 92186-9784

UPS, Fed Ex: 10100 Hopkins Dr.
La Jolla, CA 92093-0505

(619) 534-5062
FAX: (619) 534-5152

nadeau@sdsc.edu
<http://www.sdsc.edu/~nadeau>

Introduction to VRML 97
Lecturer Information

David R. Nadeau (Organizer)

Title Principal Scientist
Affiliation San Diego Supercomputer Center (SDSC)
 University of California, San Diego (UCSD)
Address P.O. Box 85608
 San Diego, CA 92186-9784

 UPS, Fed Ex: 10100 Hopkins Dr.
 La Jolla, CA 92093-0505
Work phone (619) 534-5062
Fax phone (619) 534-5152
Email nadeau@sdsc.edu
Home page <http://www.sdsc.edu/~nadeau>

Dave Nadeau is a principal scientist at the San Diego Supercomputer Center (SDSC), a national research center specializing in computational science and engineering, located on the campus of the University of California, San Diego (UCSD). Specializing in scientific visualization and virtual reality, he is the author of technical papers and articles on 3D graphics and VRML and is a co-author of two books on VRML (*The VRML Sourcebook* and *The VRML 2.0 Sourcebook*, published by John Wiley & Sons). He is the founder and lead librarian for *The VRML Repository* and *The Java3D Repository*, principal Web sites for information on VRML, Java3D, and related software.

Dave has taught VRML at multiple conferences including SIGGRAPH 96-97, WebNet 96-97, VRML 97-98, WMC/SCS 98, Eurographics 97, and Visualization 97. He was a co-chair for the *VRML Behavior Workshop* in October 1995, the first workshop on VRML behavior technology, and a co-chair for the *VRML 95* conference in December 1995, the first conference on VRML. He was on the program committees for VRML 97 and VRML 98 and is SDSC's representative to the VRML Consortium.

Dave holds a B.S. in Aerospace Engineering from the University of Colorado, Boulder, an M.S. in Mechanical Engineering from Purdue University, and is in the PhD program in Electrical and Computer Engineering at the University of California, San Diego.

John L. Moreland

Title Senior programmer/analyst
Affiliation San Diego Supercomputer Center (SDSC)
 University of California, San Diego (UCSD)
Address P.O. Box 85608
 San Diego, CA 92186-9784

UPS, Fed Ex: 10100 Hopkins Dr.
La Jolla, CA 92093-0505

Work phone (619) 534-5170

Fax phone (619) 534-5152

Email moreland@sdsc.edu

Home page <http://www.sdsc.edu/~moreland>

John Moreland is a senior programmer/analyst at the San Diego Supercomputer Center and specializes in scientific visualization and multimedia. He is an author of papers on VRML, and is a co-author of two books on VRML (*The VRML Sourcebook*, and *The VRML 2.0 Sourcebook*, published by John Wiley & Sons). He has developed multiple large VRML environments for scientific visualization, including a CAVE-based environment presented at Supercomputing 95, and a multi-user environment presented at Supercomputing 96-97.

John has taught VRML courses at SIGGRAPH 96-97, InternetWorld Asia 96-97, and Web 97 East and West, and is a frequent presenter of courses in computer graphics, visualization, and the PERL programming language. He was a co-chair for the *VRML Behavior Workshop* in October 1995, and for the *VRML 95* conference in December 1995.

John holds a B.S. in Applied Mathematics from the University of California, San Diego.

Michael M. Heck

Title Vice President of Research and Development

Affiliation Template Graphics Software, Inc.

Address Template Graphics Software, Inc.
9920 Pacific Heights Boulevard, Suite 200
San Diego, CA 92121-4331

Work phone (619) 457-2547

Fax phone (619) 452-2547

Email mmh@tgs.com

Mike Heck is the Vice President for Research and Development at Template Graphics Software (TGS), a developer of cross-platform graphics tools for browsing and authoring VRML. Both Mike and TGS are now focused upon making OpenGL, OpenInventor, and VRML available everywhere, including Sun, DEC, HP, and IBM workstations, as well as Macintosh and PC platforms.

Mike is an author of technical papers on computer graphics and the first book on 3D programming with PHIGS (*Understanding PHIGS*). He has been a speaker in multiple conference courses, including SIGGRAPH courses on PHIGS (1986), PEX (1993), OpenInventor (1995), and VRML (1996-1997), as well as a course in 3D graphics with MFC at the Visual C++ Developers Conference (1996).

Mike holds a B.S. in Electrical Engineering from the University of Pittsburgh.

Table of contents

Morning

Section 1 - Shapes, geometry, and appearance

Welcome!	1
Introducing VRML	5
Building a VRML world	16
Building primitive shapes	28
Transforming shapes	49
Controlling appearance with materials	71
Grouping nodes	84
Naming nodes	101
Summary examples	111

Section 2 - Animation, sensors, and geometry

Introducing animation	116
Animating transforms	133
Sensing viewer actions	161
Building shapes out of points, lines, and faces	175
Building elevation grids	199
Building extruded shapes	208
Controlling color on coordinate-based geometry	221
Controlling shading on coordinate-based geometry	238
Summary examples	253

Afternoon

Section 3 - Textures, lights, and environment

Mapping textures	259
Controlling how textures are mapped	276
Lighting your world	299
Adding backgrounds	311
Adding fog	325

Adding sound	333
Controlling the viewpoint	352
Controlling navigation	358
Sensing the viewer	366
Summary examples	382

Section 4 - Scripts and prototypes

Controlling detail	387
Introducing script use	399
Writing program scripts with JavaScript	409
Writing program scripts with Java	435
Accessing the browser from JavaScript and Java	459
Creating new node types	471
Providing information about your world	491
Summary examples	494
Miscellaneous extensions	501
Conclusion	506