

## **X3D: Extensible 3D Graphics for Web Authors**

## In Praise of X3D

*There will be no problem understanding these concise, clear, comprehensible background concepts for readers new to Extensible 3D (X3D). There are many notes and examples that compare X3D to Virtual Reality Modeling Language (VRML) features. Don Brutzman and Leonard Daly clearly and thoroughly illustrate each logical concept and feature of X3D with diagrams, tables, code snippets, screenshots of 3D objects/environments, and example scenes, while making use of the very latest specifications and implementations. Their approach contributes greatly to an easy and in-depth understanding of the X3D language. This book is the ultimate introductory guide to X3D!*

—Dr. Vladimir Geroimenko, University of Plymouth, School of Computing Communications and Electronics, Plymouth, UK

*This book is required reading for anybody interested in Web3D. The authors are well known and respected in the X3D community as pioneers. Their writing style is concise and engaging, set at an appropriate level to encourage understanding, and uses the concepts being introduced. Their “Hints and warnings” sections provide added value above what is available from X3D specification documents. Hard to achieve in a reference manual!*

—Professor Nigel W. John, School of Computer Science, University of Wales, Bangor

*How many times have we heard “The ISO specification is hard to read, do you have something more approachable?” This book is the answer. It provides a detailed explanation of each node in the Immersive profile and gives many reusable examples. After reading this book you’ll be well prepared to develop your own X3D content.*

—Alan Hudson, President Web3D Consortium, Yumetech Inc.

*This is a much-needed book about the X3D standard and X3D content development. The book follows the structure of the X3D standard specifications which helps readers understand and apply the X3D standard. It can also be used as a reference material in virtual reality and graphics-related courses.*

—Professor Denis Gracanin, Virginia Polytechnic Institute & State University

# X3D: Extensible 3D Graphics for Web Authors

**DON BRUTZMAN**

*Naval Postgraduate School, Monterey, California, USA*

*and*

**LEONARD DALY**

*Daly Realism, Los Angeles, California, USA*



AMSTERDAM • BOSTON • HEIDELBERG • LONDON  
NEW YORK • OXFORD • PARIS • SAN DIEGO  
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO  
Morgan Kaufmann Is an Imprint of Elsevier



Publisher: Denise E. M. Penrose  
Publishing Services Manager: George Morrison  
Senior Project Manager: Brandy Lilly  
Assistant Editor: Michelle Ward  
Cover Image: Christian Greuel/Planet 9  
Text Design: Frances Baca  
Composition: SPI India  
Illustration: Graphic World Illustration Studio  
Copyeditor: Graphic World Publishing Services  
Proofreader: Graphic World Publishing Services  
Indexer: Graphic World Publishing Services  
Interior printer: Hing Yip Printing Co. Ltd.  
Cover printer: Hing Yip Printing Co. Ltd.

Morgan Kaufmann Publishers is an imprint of Elsevier.  
500 Sansome Street, Suite 400, San Francisco, CA 94111

This book is printed on acid-free paper.  
© 2007 by Elsevier Inc. All rights reserved.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, scanning, or otherwise—without prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+ 44) 1865 843830, fax: (+ 44) 1865 853333, E-mail: [permissions@elsevier.com](mailto:permissions@elsevier.com). You may also complete your request on-line via the Elsevier homepage (<http://elsevier.com>), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."

**Library of Congress Cataloging-in-Publication Data**  
Application submitted

ISBN-13: 978-0-12-088500-8  
ISBN-10: 0-12-088500-X

For information on all Morgan Kaufmann publications,  
visit our Web site at [www.mkp.com](http://www.mkp.com) or [www.books.elsevier.com](http://www.books.elsevier.com)

Printed in China  
07 08 09 10 11      5 4 3 2 1

Working together to grow  
libraries in developing countries

[www.elsevier.com](http://www.elsevier.com) | [www.bookaid.org](http://www.bookaid.org) | [www.sabre.org](http://www.sabre.org)

ELSEVIER    BOOK AID  
International    Sabre Foundation

## *Dedication*

We dedicate this book to our families,  
with grateful thanks for their patient support and love.

—Don Brutzman and Leonard Daly



# Contents

<b>Preface</b> . . . . .	xix
1. Goals . . . . .	xix
2. Motivation . . . . .	xxi
3. Reader background . . . . .	xxi
4. Software support . . . . .	xxi
5. Book structure . . . . .	xxi
5.1. Typographic conventions . . . . .	xxi
5.2. Chapter organization . . . . .	xxii
6. Chapter descriptions . . . . .	xxii
6.1. Chapter topics . . . . .	xxii
6.2. Appendices . . . . .	xxiv
6.3. How to use this book . . . . .	xxiv
<b>Contributor List</b> . . . . .	xxv
<b>About the Authors</b> . . . . .	xxvii

## CHAPTER 1

<b>Technical Overview</b> . . . . .	1
1. Introduction . . . . .	1
2. Concepts . . . . .	2
2.1. Historical background: VRML, ISO, and the Web3D Consortium . . . . .	2
2.2. X3D browsers . . . . .	3
2.3. X3D specifications . . . . .	5
2.4. Scene graph . . . . .	7

2.5.	File structure . . . . .	10
2.5.1.	File header . . . . .	10
2.5.2.	X3D header statement . . . . .	10
2.5.3.	Profile statements . . . . .	12
2.5.4.	Component statements . . . . .	14
2.5.5.	Meta statements . . . . .	15
2.5.6.	Scene graph body . . . . .	18
2.6.	Field types . . . . .	18
2.7.	Abstract node types . . . . .	20
2.8.	File encodings: XML, ClassicVRML, and Compressed . . . . .	20
2.8.1.	Extensible Markup Language (XML) encoding: .x3d files . . . . .	21
2.8.1.1.	XML motivations . . . . .	22
2.8.1.2.	XML design for X3D . . . . .	24
2.8.1.3.	XML validation . . . . .	25
2.8.2.	ClassicVRML encoding: .x3dv files . . . . .	27
2.8.3.	Binary encoding: .x3db files . . . . .	29
2.9.	Hello World example using X3D-Edit and an X3D browser . . . . .	32
3.	Summary . . . . .	35
3.1.	Key ideas . . . . .	35
3.2.	Next chapters . . . . .	35

## CHAPTER 2

<b>Geometry Nodes, Part 1: Primitives . . . . .</b>	<b>37</b>
1. What this chapter covers . . . . .	37
2. Concepts . . . . .	38
2.1. Purpose and common functionality . . . . .	38
2.2. Common field . . . . .	40
2.2.1. Solid . . . . .	40
2.3. Abstract node types . . . . .	41
2.3.1. X3DShapeNode type . . . . .	41
2.3.2. X3DGeometryNode type . . . . .	41
2.3.3. X3DFontStyleNode type . . . . .	41
2.4. Hints and warnings . . . . .	41
3. Node descriptions . . . . .	42
3.1. Shape node . . . . .	42
3.1.1. Hints and warnings . . . . .	42



- 3.2. Box node . . . . . 44
  - 3.2.1. Size . . . . . 46
  - 3.2.2. Hints and warnings. . . . . 46
- 3.3. Cone node . . . . . 46
  - 3.3.1. bottomRadius and height. . . . . 48
  - 3.3.2. bottom and side . . . . . 48
- 3.4. Cylinder node . . . . . 49
  - 3.4.1. radius and height . . . . . 49
  - 3.4.2. bottom, side, and top . . . . . 50
  - 3.4.3. Hints and warnings. . . . . 50
- 3.5. Sphere node . . . . . 50
- 3.6. Text node . . . . . 54
- 3.7. FontStyle node . . . . . 55
- 4. Summary . . . . . 62
  - 4.1. Key ideas . . . . . 62
  - 4.2. Related nodes and concepts . . . . . 63
  - 4.3. Next chapter . . . . . 63

**CHAPTER 3**

- Grouping Nodes. . . . . 65**
  - 1. What this chapter covers . . . . . 65
  - 2. Concepts . . . . . 66
    - 2.1. Purpose and common functionality . . . . . 66
    - 2.2. Units of measurement and coordinate systems . . . . . 66
    - 2.3. Coordinate system details . . . . . 69
    - 2.4. DEF and USE . . . . . 71
    - 2.5. Abstract node types. . . . . 72
      - 2.5.1. X3DChildNode type . . . . . 72
      - 2.5.2. X3DBoundedObject type . . . . . 73
      - 2.5.3. X3DGroupingNode type . . . . . 73
      - 2.5.4. X3DInfoNode type . . . . . 74
      - 2.5.5. X3DUrlObject abstract interface . . . . . 75
  - 3. Node descriptions . . . . . 75
    - 3.1. Group and StaticGroup nodes . . . . . 75
    - 3.2. Transform node . . . . . 77
    - 3.3. Inline node . . . . . 82
    - 3.4. LOD node . . . . . 85
    - 3.5. Switch node . . . . . 89

- 4. Summary . . . . . 93
  - 4.1. Key ideas . . . . . 93
  - 4.2. Related nodes and concepts . . . . . 93
  - 4.3. Next chapter . . . . . 93

**CHAPTER 4**

**Viewing and Navigation . . . . . 95**

- 1. What this chapter covers . . . . . 95
- 2. Concepts . . . . . 95
  - 2.1. Purpose and common functionality . . . . . 96
  - 2.2. Viewing model . . . . . 96
  - 2.3. Navigation model . . . . . 97
  - 2.4. Collision detection and terrain following . . . . . 97
  - 2.5. Abstract node types . . . . . 97
    - 2.5.1. X3DBindableNode type . . . . . 97
- 3. Node descriptions . . . . . 101
  - 3.1. Viewpoint node . . . . . 101
  - 3.2. NavigationInfo node . . . . . 106
  - 3.3. Anchor node . . . . . 111
  - 3.4. Billboard node . . . . . 114
  - 3.5. Collision node . . . . . 117
- 4. Summary . . . . . 118
  - 4.1. Key ideas . . . . . 118
  - 4.2. Related nodes and concepts . . . . . 120
  - 4.3. Next chapter . . . . . 120

**CHAPTER 5**

**Appearance, Material, and Textures . . . . . 121**

- 1. What this chapter covers . . . . . 121
- 2. Concepts . . . . . 122
  - 2.1. Purpose and common functionality . . . . . 122
  - 2.2. Abstract node types . . . . . 123
    - 2.2.1. X3DAppearanceNode and X3DAppearanceChildNode types . . . . . 123
    - 2.2.2. X3DMaterialNode type . . . . . 124
    - 2.2.3. X3DTextureNode, X3DTextureCoordinateNode, and X3DTextureTransformNode types . . . . . 124

- 2.2.4. X3DTexture2DNode type . . . . . 124
    - 2.2.5. X3DTextureTransform2DNode type . . . . . 125
  - 3. Node descriptions . . . . . 125
    - 3.1. Appearance node. . . . . 125
    - 3.2. Material node . . . . . 126
    - 3.3. FillProperties node . . . . . 132
    - 3.4. LineProperties node. . . . . 134
    - 3.5. ImageTexture node . . . . . 137
    - 3.6. MovieTexture node . . . . . 140
    - 3.7. PixelTexture node . . . . . 143
    - 3.8. TextureTransform node . . . . . 146
    - 3.9. TextureCoordinate node . . . . . 150
    - 3.10. TextureCoordinateGenerator node. . . . . 152
  - 4. Summary. . . . . 155
    - 4.1. Key ideas . . . . . 155
    - 4.2. Related nodes and concepts . . . . . 155
    - 4.3. Next chapter. . . . . 155

**CHAPTER 6**

**Geometry Nodes, Part 2: Points, Lines, and Polygons . . . . 157**

- 1. What this chapter covers. . . . . 157
- 2. Concepts . . . . . 158
  - 2.1. Purpose and common functionality . . . . . 158
  - 2.2. Common geometry fields. . . . . 159
- 3. Node descriptions . . . . . 162
  - 3.1. Color and ColorRGBA nodes . . . . . 162
  - 3.2. Coordinate and CoordinateDouble nodes. . . . . 163
  - 3.3. PointSet node . . . . . 165
  - 3.4. IndexedLineSet node . . . . . 167
  - 3.5. LineSet node . . . . . 170
  - 3.6. IndexedFaceSet node . . . . . 172
  - 3.7. ElevationGrid node . . . . . 174
  - 3.8. Extrusion node . . . . . 178
- 4. Summary. . . . . 184
  - 4.1. Key ideas . . . . . 184
  - 4.2. Related nodes and concepts . . . . . 185
  - 4.3. Next chapter. . . . . 185

CHAPTER 7

**Event Animation and Interpolation** . . . . . 187

1. What this chapter covers . . . . . 187
2. Concepts . . . . . 188
  - 2.1. Animation as scene-graph modification . . . . . 188
  - 2.2. Purpose and common functionality . . . . . 188
  - 2.3. ROUTE connections . . . . . 189
  - 2.4. Interpolation . . . . . 190
  - 2.5. Constructing animation chains: 10 steps . . . . . 192
  - 2.6. Abstract node types . . . . . 196
    - 2.6.1. X3DTimeDependentNode type . . . . . 196
    - 2.6.2. X3DSensorNode type . . . . . 196
    - 2.6.3. X3DInterpolatorNode type . . . . . 196
3. Node descriptions . . . . . 200
  - 3.1. TimeSensor node . . . . . 200
  - 3.2. ScalarInterpolator node . . . . . 204
  - 3.3. ColorInterpolator node . . . . . 205
  - 3.4. PositionInterpolator node . . . . . 210
  - 3.5. OrientationInterpolator node . . . . . 211
  - 3.6. NormalInterpolator node . . . . . 214
  - 3.7. CoordinateInterpolator node . . . . . 215
  - 3.8. PositionInterpolator2D node . . . . . 216
  - 3.9. CoordinateInterpolator2D node . . . . . 218
4. Summary . . . . . 219
  - 4.1. Key ideas . . . . . 219
  - 4.2. Related nodes and concepts . . . . . 220
  - 4.3. Next chapter . . . . . 220

CHAPTER 8

**User Interactivity Nodes** . . . . . 221

1. What this chapter covers . . . . . 221
2. Concepts . . . . . 222
  - 2.1. Purpose and common functionality . . . . . 222
  - 2.2. X3DSensorNode type . . . . . 223
  - 2.3. X3DPointingDeviceSensorNode type . . . . . 223

- 2.4. X3DTouchSensorNode type . . . . . 225
- 2.5. X3DDragSensorNode type . . . . . 225
- 2.6. X3DKeyDeviceSensorNode type . . . . . 226
- 3. Node descriptions . . . . . 227
  - 3.1. TouchSensor node . . . . . 227
  - 3.2. PlaneSensor node . . . . . 231
  - 3.3. CylinderSensor node . . . . . 234
  - 3.4. SphereSensor node . . . . . 239
  - 3.5. KeySensor node . . . . . 241
  - 3.6. StringSensor node . . . . . 244
- 4. Summary . . . . . 246
  - 4.1. Key ideas . . . . . 246
  - 4.2. Related nodes and concepts . . . . . 247
  - 4.3. Next chapter . . . . . 247

**CHAPTER 9**

**Event Utilities and Scripting . . . . . 249**

- 1. What this chapter covers . . . . . 249
- 2. Concepts . . . . . 250
  - 2.1. Purpose and common functionality . . . . . 250
  - 2.2. X3DScriptNode type . . . . . 251
  - 2.3. X3DSequencerNode type . . . . . 251
  - 2.4. X3DTriggerNode type . . . . . 254
- 3. Node Descriptions . . . . . 254
  - 3.1. BooleanFilter node . . . . . 254
  - 3.2. BooleanSequencer node . . . . . 256
  - 3.3. BooleanToggle node . . . . . 258
  - 3.4. BooleanTrigger node . . . . . 259
  - 3.5. IntegerSequencer node . . . . . 259
  - 3.6. IntegerTrigger node . . . . . 261
  - 3.7. TimeTrigger node . . . . . 263
  - 3.8. Script node . . . . . 264
- 4. Summary . . . . . 278
  - 4.1. Key ideas . . . . . 278
  - 4.2. Related nodes and concepts . . . . . 278
  - 4.3. Next chapter . . . . . 278

CHAPTER 10

**Geometry Nodes, Part 3: Geometry2D Nodes . . . . . 279**

- 1. What this chapter covers . . . . . 279
- 2. Concepts . . . . . 280
  - 2.1. Purpose and common functionality . . . . . 280
  - 2.2. Common geometry fields . . . . . 281
- 3. Node descriptions . . . . . 281
  - 3.1. Arc2D node . . . . . 281
  - 3.2. ArcClose2D node . . . . . 282
  - 3.3. Circle2D node . . . . . 285
  - 3.4. Disk2D node . . . . . 287
  - 3.5. Polyline2D node . . . . . 289
  - 3.6. Polypoint2D node . . . . . 291
  - 3.7. Rectangle2D node . . . . . 292
  - 3.8. TriangleSet2D node . . . . . 293
- 4. Summary . . . . . 296
  - 4.1. Key ideas . . . . . 296
  - 4.2. Related nodes and concepts . . . . . 296
  - 4.3. Next chapter . . . . . 296

CHAPTER 11

**Lighting and Environment Nodes . . . . . 299**

- 1. What this chapter covers . . . . . 299
- 2. Concepts . . . . . 300
  - 2.1. Purpose and common functionality . . . . . 302
  - 2.2. X3DLightNode type . . . . . 303
  - 2.3. X3DBackgroundNode type . . . . . 306
- 3. Node descriptions . . . . . 308
  - 3.1. DirectionalLight node . . . . . 308
  - 3.2. Headlight . . . . . 309
  - 3.3. PointLight node . . . . . 311
  - 3.4. SpotLight node . . . . . 315
  - 3.5. Background node . . . . . 319
  - 3.6. TextureBackground node . . . . . 321
  - 3.7. Fog node . . . . . 322

- 4. Summary . . . . . 325
  - 4.1. Key ideas . . . . . 325
  - 4.2. Related nodes and concepts . . . . . 326
  - 4.3. Next chapter . . . . . 326

**CHAPTER 12**

**Environment Sensor and Sound Nodes . . . . . 327**

- 1. What this chapter covers . . . . . 327
- 2. Concepts . . . . . 328
  - 2.1. X3DEnvironmentSensorNode type . . . . . 328
  - 2.2. X3DNetworkSensorNode type . . . . . 329
  - 2.3. X3DSoundNode type . . . . . 330
  - 2.4. X3DSoundSourceNode type . . . . . 330
- 3. Node descriptions . . . . . 330
  - 3.1. LoadSensor node . . . . . 330
  - 3.2. ProximitySensor node . . . . . 334
  - 3.3. VisibilitySensor node . . . . . 339
  - 3.4. Sound node . . . . . 340
  - 3.5. AudioClip node . . . . . 347
- 4. Summary . . . . . 351
  - 4.1. Key ideas . . . . . 351
  - 4.2. Related nodes and concepts . . . . . 351
  - 4.3. Next chapter . . . . . 351

**CHAPTER 13**

**Geometry Nodes, Part 4: Triangles and Quadrilaterals . . . 353**

- 1. What this chapter covers . . . . . 353
- 2. Concepts . . . . . 354
  - 2.1. Purpose and common functionality . . . . . 354
  - 2.2. Abstract node types . . . . . 355
    - 2.2.1. X3DColorNode, X3DCoordinateNode,  
X3DGeometricPropertyNode, X3DGeometryNode,  
and X3DNormalNode types . . . . . 355
    - 2.2.2. X3DComposedGeometryNode type . . . . . 356
  - 2.3. Common geometry fields . . . . . 356

- 3. Node descriptions . . . . . 361
  - 3.1. Normal node . . . . . 361
  - 3.2. TriangleSet node . . . . . 363
  - 3.3. TriangleFanSet node . . . . . 366
  - 3.4. TriangleStripSet node. . . . . 368
  - 3.5. QuadSet node . . . . . 370
  - 3.6. IndexedTriangleSet node . . . . . 370
  - 3.7. IndexedTriangleFanSet node . . . . . 373
  - 3.8. IndexedTriangleStripSet node. . . . . 375
  - 3.9. IndexedQuadSet node . . . . . 377
- 4. Summary. . . . . 378
  - 4.1. Key ideas . . . . . 378
  - 4.2. Related nodes and concepts. . . . . 379
  - 4.3. Next chapter. . . . . 379

**CHAPTER 14**

**Creating Prototype Nodes . . . . . 381**

- 1. What this chapter covers . . . . . 381
- 2. Concepts . . . . . 382
  - 2.1. Extensibility . . . . . 382
  - 2.2. Strong typing of nodes. . . . . 383
  - 2.3. Comparison with Inline node. . . . . 383
  - 2.4. Syntax alert. . . . . 383
- 3. Functional descriptions . . . . . 384
  - 3.1. Prototype declarations: ProtoDeclare . . . . . 384
  - 3.2. External prototype declarations: ExternProtoDeclare . . . . . 394
  - 3.3. Prototype instances: ProtoInstance . . . . . 397
- 4. Summary. . . . . 398
  - 4.1. Key ideas . . . . . 398
  - 4.2. Related nodes and concepts. . . . . 399

**Afterword: Getting Involved . . . . . 401**

- Learning X3D . . . . . 401
- Building Software . . . . . 401



Creating Content . . . . .	402
X3D Development . . . . .	402
Other Web3D Activities . . . . .	402
<b>APPENDICES</b>	
<b>A. Online Resources . . . . .</b>	<b>403</b>
Web Site Description . . . . .	403
<b>B. Help: X3D/VRML Examples . . . . .</b>	<b>405</b>
Applications, Players, and Plugins for X3D/VRML Viewing . . . .	405
Web3D Consortium Members . . . . .	405
Not (yet) Web3D Consortium Members . . . . .	406
Authoring Tools . . . . .	406
Authoring Support . . . . .	407
Conversion and Translation Tools . . . . .	407
Examples . . . . .	408
License . . . . .	409
PowerPoint Support . . . . .	409
References . . . . .	409
<b>C. X3D Scene Authoring Hints . . . . .</b>	<b>411</b>
Authoring practices . . . . .	411
Credits . . . . .	414
Dates . . . . .	414
HTML . . . . .	415
Images . . . . .	415
Inline and Prototype Subscenes . . . . .	416
License . . . . .	417
meta tags . . . . .	418

Naming Conventions . . . . .	418
Naming of Multiple Similar Autogenerated Files . . . . .	419
Scripts . . . . .	419
URL Links . . . . .	420
Viewpoints . . . . .	421
List of References . . . . .	423
<b>Index . . . . .</b>	<b>425</b>

# Preface

*A journey of a thousand miles begins with a single step.*

—Chinese proverb

## 1. Goals

Welcome! This book describes Extensible 3D (X3D) Graphics, the open international standard for 3D on the Web. X3D is used for building 3-dimensional models, both simple and sophisticated. X3D can show animated objects from different viewpoint perspectives, allowing user insight and interaction. X3D models can be further combined and connected to build sophisticated 3D virtual environments running across the Internet.

The primary goal of this book is to show Web authors how to build and connect X3D models. Another goal is to teach students the essential principles of Webcapable 3D graphics. Because X3D scene-graph concepts are easily understood from the perspective of creating Extensible Markup Language (XML) documents, this material has been successfully used to teach 3D to students with or without prior programming experience. Finally the book should serve as a handy reference for X3D authors, explaining both the broad principles and specific details of X3D scene construction.

## 2. Motivation

More than 30 years of innovative research, theoretical mathematics, and application-development effort have made 3D graphics an exciting field that produces amazing results and frequent surprises. The heart of the 3D community is the Special Interest Group on Graphics (SIGGRAPH), which is one of many professional societies making up the Association for Computing Machinery (ACM). SIGGRAPH can be found online at [www.siggraph.org](http://www.siggraph.org). The classic text for this subject area is *Computer Graphics, Principles and Practice* by James D. Foley, Andries van Dam, Stephen K. Feiner, and John F. Hughes (Addison-Wesley, 2nd Edition, 1997). Hundreds of other books about

3D graphics are also available, covering a wide spectrum of technical and practical topics.

Nevertheless, for most Web authors and software programmers, 3D graphics remains a complex niche technology that is not easily mastered and not often used. For many people, 3D is best known from movies or computer games. It is something “special” created by others, and viewed only in movie theaters, by DVD playback, or by locally installed computer-game programs. 3D scenes are typically created only by small teams of experts who have worked hard to achieve programming mastery using expensive professional application tool kits.

Enabling Web authors to independently create and compose 3D objects has been difficult for the same reasons, because such graphics have almost always been produced by highly trained programmers and animators using specialty 3D software packages. Unfortunately, because most 3D software packages are created using proprietary commercial products or computer programming languages, users can only rarely combine diverse 3D models and interact directly with them. This challenging state of affairs has prevented 3D graphics from becoming a commonly authored form of multimedia.

Rather than creating yet another technical niche, X3D is designed for interchange and interoperability that allows authors to build simple or sophisticated 3D models of interest. X3D collects the most common 3D techniques and provides common-denominator graphics capabilities that map satisfactorily to many industry approaches (which often vary more in style than in substance). Thus translators and converters can import and export X3D scenes both to and from many other 3D graphics formats, allowing easier 3D interchange. This approach allows the various specialty tools to do what they are good at, noncompetitively, while making 3D models available for composition together in an effective, efficient way. Thus X3D can enable 3D graphics to transition from isolated islands of functionality to become a first-class media type used in web pages and documents everywhere.

According to the International Standards Organization (ISO), Extensible 3D Graphics (X3D) is the “ISO standard that defines a royalty-free run-time system and delivery mechanism for real-time 3D content and applications running on a network.” Supported X3D file-format encodings enable the construction of interactive, animated, 3D virtual environments that stably incorporate the latest advances in graphics hardware capabilities, file compression, and data security. Networking is optional, allowing X3D scenes to work compatibly over the Internet via http servers, on local disk drives or CDs, and even as standalone email and chat attachments.

Because licensing restrictions for all of the underlying contributions in the X3D standards are royalty free, X3D technologies can be used freely without fear that they will “time out” or become unusable as a result of corporate-profitability concerns or one-sided policy changes. An open process guided by the nonprofit Web3D Consortium has consolidated 12 years of work and achieved full certification from ISO as an International Standard. Thus X3D models can be used and improved by individuals, companies, government agencies, and educators without fear that the underlying technology might eventually become unavailable.

The 3D graphics community is highly technical, artistic, sensitive, and motivated, usually all at the same time. Many X3D authors and users already hope that bringing

3D graphics to the Web can make all kinds of models more widely available, thus making 3D graphics a useful part of everyday life. Building compatible X3D models for the Web is a worthy effort, showing progress that steadily grows with each passing day.

Hopefully this book will help you learn the fundamentals of X3D, master the techniques needed to show others your work, and bootstrap the Web into the third dimension. Good luck with your efforts!

### 3. Reader background

Prior experience in 3D graphics programming or XML authoring is helpful but not required to use this book. The creation of X3D scenes is presented with an emphasis on XML so that content production might appear familiar to Web authors. Many examples are developed as part of the book, and many more are available online.

### 4. Software support

A free Java-based authoring tool designed especially for X3D is available on the Web site: X3D-Edit. This scene-graph editor has been used to produce all of the book examples. X3D-Edit has been tested under Windows, Macintosh, and Linux operating systems. Features include tooltips for each X3D node and field in multiple languages: English, Chinese, French, German, Italian, Portuguese, and Spanish. Models and tooltips have also been tested satisfactorily using other XML-based tools.

Authored X3D scenes are viewed using separate browser plugins, usually installed into the local Web browser. Multiple X3D viewer plugins and applications are provided on the support Web site and online.

### 5. Book structure

#### 5.1. Typographic conventions

The following typographic conventions are observed in this book.

- Definition terms are identified using *italics*.
- Source content is shown using a fixed-width font.
- Null or absent nodes are represented as NULL.
- Empty arrays may be represented as empty square brackets [ ].
- Empty string values may be represented as empty double-quote marks, either "" or &quot; &quot; .

## 5.2. Chapter organization

### 5.2.1. Introductory chapters

This preface describes the structure of the book. Chapter 1 is the in-depth Technical Overview, which describes the architectural principles governing how X3D works. Topics include scene-graph structure, file encodings, the Extensible Markup Language (XML), various related ISO specifications, and the underlying technical detail common to all X3D components and nodes.

### 5.2.2. X3D Component chapters

Each of the subsequent chapters cover the primary nodes provided by each component in the X3D Immersive Profile. The layout of each chapter is organized consistently in order to best support clarity of presentation. Chapter structure follows.

The Concepts section discusses shared usage issues, abstract node-type interfaces, and the common fields shared by nodes presented in the chapter. Each node is explained using both XML syntax (for .x3d files) and ClassicVRML syntax (for .x3dv files). Each of the fields is then described in detail, emphasizing allowed values and the types of functionality provided. Usage hints and warnings emphasize lessons learned and problems to avoid.

Each chapter concludes with a summary of the key ideas covered, a list of related nodes and concepts, and a brief preview of the next chapter.

### 5.2.3. Examples

Working example scenes are provided for each node. First modeled by masters students at the Naval Postgraduate School, the Kelp Forest exhibit of Monterey Bay Aquarium provides a common theme for most of the examples in the book. These example X3D scenes are designed to illustrate specific features of X3D.

The best approach to learning this material is to load each scene in an X3D browser, view it, and verify that it works as described, and then modify values for various nodes and fields to confirm your understanding of the concepts discussed.

## 6. Chapter descriptions

This book is designed to be read in order. Each chapter builds upon previous chapters with only occasional forward referencing. The ordering of chapters provides a clear path to learn how to build increasingly sophisticated 3D worlds.

### 6.1. Chapter topics

The chapters are organized by function and ordered as follows:

1. **Technical Overview.** General introduction of the fundamentals of 3D, including scene graphs, events, node reuse, file structure and encodings, components and profiles, and conformance.
2. **Geometry Nodes, Part 1: Primitives.** The basic primitive shapes: Box, Sphere, Cylinder, Cone, and Text.

3. **Grouping Nodes.** Collecting and positioning objects in the 3D world using Inline, LOD, Group and StaticGroup, Switch, Transform, and Anchor.
4. **Viewing and Navigation.** How to view and navigate in the 3D world with Viewpoint and Navigation Info.
5. **Appearance, Material, and Textures.** Adding colors, shininess, and transparency using Material, or adding image-file textures using PixelTexture, ImageTexture, MovieTexture, TextureTransform, TextureCoordinate, and TextureCoordinateGenerator.
6. **Geometry Nodes, Part 2: Points, Lines, and Polygons.** Geometric creations that are more advanced than the basic shapes, including Coordinate, Color, PointSet, LineSet, IndexedLineSet, IndexedFaceSet, ElevationGrid, and Extrusion.
7. **Event Animation and Interpolation.** Making objects move, twist, wiggle, and shake with TimeSensor and the interpolation nodes: ScalarInterpolator, PositionInterpolator, PositionInterpolator2D, ColorInterpolator, OrientationInterpolator, and CoordinateInterpolator.
8. **User Interactivity Nodes.** Allowing users to interact with the world by connecting TouchSensor, PlaneSensor, CylinderSensor, SphereSensor, KeySensor, and StringSensor nodes.
9. **Event Utilities and Scripting.** Event type conversion, improved animation using the event-utility nodes BooleanFilter, BooleanSequencer, BooleanToggle, BooleanTrigger, IntegerSequencer, IntegerTrigger, and the author-programmable Script node.
10. **Geometry Nodes, Part 3: Geometry2D Nodes.** Flat geometry can be helpful for building 2D constructs that face the viewer. Planar nodes include Polypoint2D, Rectangle2D, TriangleSet2D, Polyline2D, Circle2D, Arc2D, ArcClose2D, and Disk2D.
11. **Lighting and Environment Nodes.** Lighting and scene background effects are accomplished using DirectionalLight, PointLight, SpotLight, Background, TextureBackground, Fog, and Sound.
12. **Environment Sensor and Sound Nodes.** User activity in the environment can be detected and processed by using LoadSensor, Collision, Billboard, ProximitySensor, and VisibilitySensor.
13. **Geometry Nodes, Part 4: Triangles and Quadrilaterals.** Fundamental low-level geometry creation using triangles: TriangleSet, TriangleStripSet, TriangleFanSet, IndexedTriangleSet, IndexedTriangleStripSet, and IndexedTriangleFanSet.
14. **Creating Prototype Nodes.** Probably the most powerful extension feature in X3D is the ability to define new reusable nodes, known as prototypes. Prototype declarations are combinations of already-existing nodes and (optionally) other prototypes. Prototype instances can then be used like any other X3D node. External prototype declarations allow authors to collect reusable prototype definitions together in a single file that can be accessed by other scenes.

15. **Metadata and Information Nodes.** Metadata includes background information describing the makeup of the X3D scene itself. Nodes of interest include WorldInfo, MetadataSet, MetadataDouble, MetadataFloat, MetadataInteger, and MetadataString. This chapter is provided online.

## 6.2. Appendices

Several appendices provide supplementary information.

- Description of resources available online
- Help: X3D/VRML examples
- X3D scene authoring hints

## 6.3. How to use this book

X3D has many capabilities, so there is a lot to learn. Chapter 1, Technical Overview, provides a thorough background study of how X3D works. Each of the following chapters presents a set of related X3D nodes, each providing new functionality.

You will learn best if you load and test each example using X3D-Edit or another authoring tool. Usually, you need to modify nodes and field values to fully observe how the displayed scene might change. Do not worry about breaking things. A common occurrence in 3D graphics is that a problem or bug actually turns out to be a feature!

Chapters 2–6 provide the essentials of X3D scene graphs and are essential reading. Chapters 7–9 explain animation and interaction and are best read together. Because X3D is consistently designed, readers can skip around among the chapters if they wish. This is especially helpful when building new parts of a specific project.

The suggested sequence for proceeding through the chapters depends on the reader. Guidelines follow.

- **Web authors and X3D students.** First, read just the Section 1 Introduction in Chapter 1, Technical Overview, then skip ahead to Chapter 2. Follow the chapters in order. Read the other Chapter 1 sections later when you are ready to learn more about the technical underpinnings and design rationale of X3D.
- **Experienced 3D programmers.** Read Chapter 1, Technical Overview, first to see how X3D differs from (and is similar to) the technologies that you are already familiar with. Gain familiarity with Chapters 2–6 for scene-graph fundamentals, and then Chapters 7–9 for animation and scripting. The remaining chapters can be followed in any order.
- **Experienced X3D authors.** Read the descriptions of both the XML and ClassicVRML encodings in Chapter 1, Technical Overview, so that you understand the syntax of both file formats. They are functionally equivalent. You can then read the rest of the book in any order and use it as a ready reference manual. You may find the sections providing hints and warnings especially helpful.

Feedback from readers is encouraged and welcome. Errata and improvements are posted on the book's Web site.



# Contributor List

Many talented individuals contributed unselfishly to the technologies and capabilities described in this book. In particular, we want to thank the following people:

Dr. Michael Aratow (Web3D Medical Working Group); Christian Bouville (France Telecom), Len Bullard (independent consultant); Timothy Childs (RoundUP Productions), David Colleen and Christian Greuel (Planet 9 Studios); Dr. David Duce (Oxford Brooks University, Oxford, United Kingdom), Miriam English (miriam-english.org); Dr. Pablo Figueroa (Universidad de los Andes Colombia); Dr. Vladimir Geroimenko (University of Plymouth, United Kingdom); Rob Glidden (Sun Microsystems), Rick Goldberg (Aniviza); Dr. Julian Gomez (Polished Pixels), Barb Helfer (Capital University, Columbus, Ohio); Alan Hudson, Justin Couch, and Stephen Matsuba (Yumetech Inc.); Dr. Nigel John (University of Wales, Bangor, United Kingdom); Robert Lansdale (Okino Graphics), Braden McDaniel (endoframe); Tony Parisi, Keith Victor, and Dave Arendash (Media Machines); Dr. Nicholas Polys and Dennis Graconin (Virginia Tech); Dr. Richard Puk (Intelligraphics Inc.); Dr. Mark Pullen (George Mason University); Sandy Ressler (NIST); Larry Rosenthal (Cube3.com); Dr. Cristina Russo dos Santos (European Patient Office); Peter Schickel, Holger Grahn, Cecile Muller, Herbert Stocker, and Brian Hay (Bit Management); Hyunju Shim (University of Florida, Gainesville); John A. Stewart and Sarah Dumoulin (Communications Research Centre Canada); Chris Thorne and Viveka Weiley (Ping Interactive Australia); Neil Trevett (nVidia and Khronos); Rita Turkowski (Web3D Consortium); Joe D. Williams (HyperMultiMedia); Mitch Williams (3D-online); Jeff Weekley, Don McGregor, Curt Blais, and Terry Norbraten (Naval Postgraduate School); several dozen NPS graduate students, and the many contributors who have worked on X3D.



# About the Authors

Don Brutzman is a computer scientist and Associate Professor working in the Modeling Virtual Environments & Simulation (MOVES) Institute at the Naval Postgraduate School in Monterey, California, USA.

He is one of the original participants involved with Virtual Reality Modeling Language (VRML) since this community effort began in 1994. He is a founding member of the nonprofit Web3D Consortium, serving on the Board of Directors. He cochairs the X3D and X3DEarth Working Groups and leads a variety of X3D technical development efforts. He also serves as the Web3D Consortium liaison to the World Wide Web Consortium (W3C) Advisory Committee and the Open Geospatial Consortium (OGC). He has been teaching VRML and X3D since 1996.

Dr. Brutzman's research interests include underwater robotics, real-time 3D computer graphics, artificial intelligence, and high-performance networking. He is a member of the Institute of Electrical and Electronic Engineers (IEEE), the Association for Computing Machinery (ACM) Special Interest Group on Graphics (SIGGRAPH), and the American Association for Artificial Intelligence (AAAI). Dr. Brutzman directs numerous related research and development projects as part of the Extensible Modeling and Simulation Framework (XMSF).

A retired submarine officer, Dr. Brutzman holds degrees from the U.S. Naval Academy in Annapolis, Maryland, and the Naval Postgraduate School.

Leonard Daly is President of Daly Realism, an Internet Consultant working in 3D graphics and e-business, based in Los Angeles, California, USA.

He is secretary of the X3D working group, managing and contributing to the X3D specification development and ISO approval process. He has been working in 2D and 3D computer graphics since 1980 and with the X3D/Virtual Reality Modeling Language (VRML) community since 1997. He was founder and Treasurer of the Los Angeles VRML User's Group (LAVUG). He was the organizer and lead presenter of X3D courses at SIGGRAPH Conferences (2002 and 2003) and the Web3D Symposium (2000, 2002, and 2003).

Mr. Daly's primary research interest is in large interactive networked systems with real-time 3D graphical plus audio interfaces. He is a member of the Association for Computing Machinery (ACM) Special Interest Group on Graphics (SIGGRAPH) and has served on the nonprofit Web3D Consortium Board of Directors. He holds a Bachelor of Science in Mathematics from Harvey Mudd College.