



AUTODESK
Official Press

Todd Palamar



Mastering Autodesk® Maya® 2016

 SYBEX
A Wiley Brand



Mastering Autodesk® Maya® 2016

Todd Palamar

 **AUTODESK**
Official Press

 **SYBEX**
A Wiley Brand

Acquisitions Editor: Stephanie McComb
Development Editor: Gary Schwartz
Technical Editor: Keith Reicher
Production Editor: Rebecca Anderson
Copy Editor: Linda Recktenwald
Editorial Manager: Mary Beth Wakefield
Production Manager: Kathleen Wisor
Associate Publisher: Jim Minatel
Book Designers: Maureen Forys, Happenstance Type-O-Rama; Judy Fung
Proofreader: Kim Wimpsett
Indexer: Robert Swanson
Project Coordinator, Cover: Brent Savage
Cover Designer: Wiley
Cover Image: Courtesy of Todd Palamar

Copyright © 2016 by John Wiley & Sons, Inc., Indianapolis, Indiana
Published simultaneously in Canada

ISBN: 978-1-119-05982-0
ISBN: 978-1-119-05970-7 (ebk.)
ISBN: 978-1-119-05985-1 (ebk.)

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, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2015945407

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Autodesk and Maya are registered trademarks of Autodesk, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

10 9 8 7 6 5 4 3 2 1

Acknowledgments

I would like to thank Eric Keller, whose work on previous editions has built a foundation for this current edition. A big thanks to Keith Reicher, our technical editor, who worked above and beyond the call of duty.

I would also like to thank our development editor, Gary Schwartz. Gary is a pleasure to work with, and he always put the content of the book first.

About the Author

Todd Palamar is a 23-year veteran of the computer animation industry. After transitioning early in his career from traditional special effects to computer-generated imagery, Todd did effects work for several direct-to-video movies. He later went on to work on numerous video games, including Sega of Japan's coin-operated title *Behind Enemy Lines*, as well as *Dukes of Hazzard* and *Trophy Buck 2* for the Sony PlayStation console.

For six years, Todd taught at Full Sail University in Winter Park, Florida. During this time, he received numerous accolades as an outstanding educator. Todd currently runs his own company, Surrealistic Producing Effects, making and distributing movies. Todd has written several books, among them *Maya Cloth for Characters* (Surrealistic Producing Effects, 2008), *Maya Studio Projects: Dynamics* (Sybex, 2009), and *Maya Studio Projects: Photorealistic Characters* (Sybex, 2011). The breadth of his experience has allowed him to work on location-based entertainment, military simulations, television commercials, and corporate spots. You can see more of Todd's work on his company's website, www.speffects.com.

Contents at a Glance

<i>Introduction</i>	<i>xix</i>
Chapter 1 • Working in Autodesk Maya	1
Chapter 2 • Introduction to Animation	39
Chapter 3 • Hard-Surface Modeling	111
Chapter 4 • Organic Modeling	163
Chapter 5 • Rigging and Muscle Systems	201
Chapter 6 • Animation Techniques	275
Chapter 7 • Lighting with mental ray	313
Chapter 8 • mental ray Shading Techniques	347
Chapter 9 • Texture Mapping	373
Chapter 10 • Paint Effects.....	419
Chapter 11 • Rendering for Compositing	473
Chapter 12 • Introducing nParticles.....	507
Chapter 13 • Dynamic Effects.....	581
Chapter 14 • Hair and Clothing	631
Chapter 15 • Maya Fluids.....	677
Chapter 16 • Scene Management and Virtual Filmmaking	725
Appendix A • The Bottom Line.....	783
Appendix B • Autodesk Maya 2016 Certification	803
<i>Index</i>	<i>807</i>

Contents

<i>Introduction</i>	<i>xix</i>
---------------------------	------------

Chapter 1 • Working in Autodesk Maya1

Color Management	1
Creating and Editing Nodes	3
Using the Hypergraph	3
Connecting Nodes with the Node Editor	8
Creating Node Hierarchies in the Outliner	13
Displaying Options in the Outliner	17
The Channel Box	20
The Attribute Editor	24
Working with Shader Nodes in the Hypershade	27
Creating Maya Projects	35
Creating a New Project	36
Editing and Changing Projects	37
The Bottom Line	38

Chapter 2 • Introduction to Animation39

Using Joints and Constraints	39
Joint Basics	39
Point Constraints	41
Aim Constraints	41
Inverse Kinematics	44
IK Handle Tool	45
Creating a Master Control	48
Keyframe Animation	51
Creating Keyframes	52
Auto Keyframe	54
Moving and Scaling Keyframes on the Timeline	55
Copy, Paste, and Cut Keyframes	57
The Graph Editor	59
Animation Curves	60
Editing Animation Curves	65
Weighted Tangents	69
Additional Editing Tools	70
Breakdowns and In-Betweens	74
Pre- and Post-Infinity	76
Playblast and FCheck	79
Driven Keys	81
Creating a Driven Key	81
Looping Driven Keys	84

Copying and Pasting Driven Keys	85
Motion-Path Animation	88
Motion Trails	90
Animating Constraints	93
Animation Layers	97
Creating an Animation Layer	97
Layer Mode	99
Other Options in the Layer Editor	102
Layer Hierarchy	103
Merging Layers	106
Grease Pencil	107
The Bottom Line	109

Chapter 3 • Hard-Surface Modeling 111

Understanding Polygon Geometry	111
Polygon Vertices	111
Polygon Edges	112
Polygon Faces	113
Working with Smooth Polygons	114
Understanding NURBS	115
Understanding Curves	116
Understanding NURBS Surfaces	118
Surface Seams	121
NURBS Display Controls	121
Using Subdivision Surfaces	122
Employing Image Planes	122
Modeling NURBS Surfaces	126
Lofting Surfaces	134
Attaching Surfaces	136
Converting NURBS Surfaces to Polygons	139
Modeling with Polygons	140
Using Booleans	141
Cleaning Topology	146
Creating Your Own Polygons	148
Multi-Cut Tool	151
Combining and Merging Geometry	153
Bridge Polygon	155
Mirror Cut	159
The Bottom Line	160

Chapter 4 • Organic Modeling 163

Implement Box Modeling	163
Shaping Using Smooth Mesh Polygon Geometry	164
Multi-Cut with Edge Flow	175
Slide Edge Tool	177
Offset Edge Loops	178

Employ Build-Out Modeling	179
Extrude along a Curve	181
Sculpt Polygons	185
Soft Select Tool	185
Sculpting Tools	187
Use Retopology Tools	189
Importing and Exporting	189
Alembic Cache Files	190
Slide on Surface	190
Quad Draw	193
Reduce	198
The Bottom Line	199

Chapter 5 • Rigging and Muscle Systems 201

Understanding Rigging	201
Creating and Organizing Joint Hierarchies	203
Orienting Joints	211
Naming Joints	212
Mirroring Joints	215
Rigging the Giraffe	216
IK Legs	216
FK Blending	219
Rotate-Plane Solver	221
Creating Custom Attributes	225
Spline IK	230
Human Inverse Kinematics	237
Skeleton Generator	237
Character Controls	239
Interoperability	241
Skinning Geometry	242
Interactive/Smooth Binding	243
Weighting the Giraffe	243
Geodesic Voxel Binding	250
Painting Skin Weights	253
Editing Skin Weights in the Component Editor	258
Copying Skin Weights	259
Mirroring Skin Weights	260
The Maya Muscle System	260
Understanding the Maya Muscle System	260
Using Capsules	261
Creating a Muscle Using the Muscle Builder	262
Editing Muscle Parameters	268
Converting the Smooth Skin to a Muscle System	270
Sliding Weights	272
The Bottom Line	273

Chapter 6 • Animation Techniques	275
Working with Deformers	275
ShrinkWrapping Geometry	275
Using Textures to Deform Objects	278
Delta Mush	281
Animating Facial Expressions Using Blend Shapes	283
Creating Blend Shape Targets	286
Creating Blend Shapes	292
Painting Blend Shape Weights	294
Adding Targets	297
Animating a Scene Using Nonlinear Deformers	298
Creating a Wave Deformer	299
Squashing and Stretching Objects	300
Twisting Objects	302
Creating a Jiggle Effect	304
Applying Jiggle Deformers	304
Painting Jiggle Weights	305
Optimizing Animations with the Geometry Cache	307
Creating a Geometry Cache	307
Editing the Cache Playback	308
Applying Motion Capture	309
The Bottom Line	311
Chapter 7 • Lighting with mental ray	313
Shadow-Casting Lights	313
Shadow Preview	314
Depth Map Shadows	316
mental ray Shadow Map Overrides	320
Raytrace Shadows	322
Indirect Lighting: Global Illumination	323
Color Bleeding	326
Indirect Illumination: Final Gathering	326
Light-Emitting Objects	327
Using Lights with Final Gathering	331
Image-Based Lighting	332
Enabling IBL	332
IBL and Final Gathering	333
Physical Sun and Sky	335
Enabling Physical Sun and Sky	335
Editing the Sky Settings	337
mental ray Area Lights	338
Light Shaders	341
Physical Light Shader	341
Tone Mapping	343
Photometric Lights and Profiles	344
The Bottom Line	345

Chapter 8 • mental ray Shading Techniques	347
Shading Concepts	347
Diffusion	350
Reflection	351
Refraction	351
The Fresnel Effect	353
Anisotropy	353
Layering Shaders	354
Creating Reflections and Refractions	355
Creating Metals and Plastics	362
Adding Shaders to Individual Polygons	364
Building a Layered Car Paint Shader	365
Base Parameters	367
Flake Parameters	369
Specular Reflection Layer	370
Glossy Reflection Parameters	370
The Bottom Line	371
Chapter 9 • Texture Mapping	373
UV Texture Layout	373
What Are UV Texture Coordinates?	374
Mapping the Giraffe Leg	376
Unfolding UVs	381
Mapping the Giraffe Head	381
Mirroring UVs	384
More UV Tools	387
Arranging UV Shells	388
Additional UV Mapping Considerations	391
Transferring UVs	392
Multiple UV Sets	392
Optimizing Textures	392
Bump and Normal Mapping	393
Bump Maps	393
Normal Maps	394
Creating Normal Maps	396
Applying Normal Maps	400
Displacement Mapping	402
Subsurface Scattering	407
Fast, Simple Skin-Shader Setup	407
Subsurface Specularity	411
ShaderFX	415
The Bottom Line	417
Chapter 10 • Paint Effects	419
Using the Paint Effects Canvas	419
The Paint Effects Window	420

Painting in Scene Mode	424
Painting on 3D Objects	425
Understanding Strokes	427
The Anatomy of a Paint Effects Stroke	427
Brush Sharing	430
Understanding Brush Curve Nodes	431
Designing Brushes	433
Starting from Scratch	433
Tubes	436
Growing Flowers	438
Adding Leaves	444
Create Complexity by Adding Strokes to a Curve	447
Shaping Strokes with Behavior Controls	450
Applying Forces	450
Displacement, Spiral, and Bend	451
Animating Strokes	453
Animating Attribute Values	455
Adding Turbulence	456
Animating Growth	457
Modifiers	458
Surface Collisions	459
Rendering Paint Effects	460
Illumination	461
Shadow Effects	461
Shading Strokes and Tubes	463
Texturing Strokes	466
Converting Strokes to Geometry	469
The Bottom Line	471
Chapter 11 • Rendering for Compositing	473
Render Layers	473
Creating Render Layers	474
Render Layer Overrides	477
Creating Overrides for Rendering Cameras	479
Material Overrides	481
Render Layer Blend Modes	481
Render Passes	486
Rendering Multiple Passes from a Single Render Layer	488
Creating an Ambient Occlusion Render Pass	492
Setting Up a Render with mental ray	494
File Tokens	494
Specifying Frame Range	497
Starting a Batch Render	498
Command-Line Rendering	499
mental ray Quality Settings	502
Tessellation and Approximation Nodes	502

Sampling	504
Filtering	504
The Bottom Line.....	504

Chapter 12 • Introducing nParticles..... 507

Creating nParticles.....	507
Drawing nParticles Using the nParticle Tool.....	508
Spawning nParticles from an Emitter.....	512
Emitting nParticles from a Surface	515
Filling an Object with nParticles	518
Making nParticles Collide with nRigids	523
Passive Collision Objects.....	523
Collide Strength and Collision Ramps	528
Using nParticles to Simulate Liquids	531
Creating Liquid Behavior	531
Converting nParticles to Polygons.....	536
Shading the nParticle Mesh	537
Emitting nParticles Using a Texture.....	539
Surface Emission.....	539
Using Wind.....	545
Shading nParticles and Using Hardware Rendering to Create Flame Effects.....	549
Shading nParticles to Simulate Flames.....	549
Creating an nCache	551
Using the Hardware Render Buffer	553
Controlling nParticles with Fields.....	556
Using Multiple Emitters.....	556
Volume Axis Curve.....	560
Working with Force Fields	566
Painting Field Maps	569
Using Dynamic Fields	572
Rendering Particles with mental ray.....	576
The Bottom Line.....	579

Chapter 13 • Dynamic Effects..... 581

Creating nCloth Objects	581
Making a Polygon Mesh Dynamic.....	582
Applying nCloth Presets	585
Making Surfaces Sticky	587
Creating nConstraints	589
Making nCloth Objects Expand Using Pressure.....	593
Additional Techniques.....	595
Creating an nCache	595
Creating nCloth and nParticle Interactions	597
Creating an nParticle Goal	598
Controlling Collision Events.....	601
Bursting an Object Open Using Tearable nConstraints	603
Crumbling Tower	604

Soft Body Dynamics	606
Creating Flying Debris Using nParticle Instancing	607
Adding nParticles to the Scene	607
Sending the Debris Flying Using a Field	610
Creating a More Convincing Explosion by Adjusting nParticle Mass	612
Instancing Geometry	613
Animating Instances Using nParticle Expressions	615
Randomizing Instance Index	615
Connecting Instance Size to nParticle Mass	620
Controlling the Rotation of nParticles	624
Bullet Physics	626
The Bottom Line	629

Chapter 14 • Hair and Clothing 631

Understanding XGen	631
Creating an XGen Description	632
XGen Library	637
Rendering an XGen Description	638
Animating Using Dynamic Curves	642
Using Dynamic Curves with IK Splines	642
Creating an IK Spline Handle from the Dynamic Curve	647
Using Forces	648
Adding Hair to a Character	649
Applying Hair to a Surface	649
Determining Hair Shape	653
Styling Hair	656
Start and Rest Positions	656
Painting Follicle Attributes	658
Modifying Curves	660
Curling, Noise, Sub Clumping, and Braids	660
Rendering Hair	661
Creating Clothing for Characters	662
Modeling Clothes for nCloth	662
Using Constraints	664
Connecting Buttons to the Shirt	670
Applying Forces	671
Painting nCloth Properties	671
The Bottom Line	675

Chapter 15 • Maya Fluids 677

Using Fluid Containers	677
Using 2D Containers	678
Adding an Emitter	679
Using Fields with Fluids	683
Using 3D Containers	686

Fluid Interactions	687
Emitting Fluids from a Surface.	687
Making Flames	690
Igniting the Fuel	693
Filling Objects	694
Rendering Fluid Containers	700
Creating Fluids and nParticle Interactions	702
Emitting Fluids from nParticles	702
Creating Flaming Trails	706
Creating Water Effects	708
Bifrost Liquid Simulation	708
Shading Bifrost Liquids	714
Guiding Liquid	717
Creating an Ocean	720
The Bottom Line	722
 Chapter 16 • Scene Management and Virtual Filmmaking	 725
Organizing Complex Node Structures with Assets	725
Creating an Asset	726
Publishing Asset Attributes	730
Using the Asset Editor	731
Viewing Assets in the Node Editor	733
File References	733
Referencing a File	734
Bounding-Box Representations	736
Determining the Image Size and Film Speed of the Camera	737
Setting the Size and Resolution of the Image	738
Setting the Film Speed	740
Creating and Animating Cameras	740
Creating a Camera	741
Setting Camera Attributes	744
Limiting the Range of Renderable Objects with Clipping Planes	747
Composing the Shot Using the Film-Back Settings	750
Creating a Camera-Shake Effect	752
Using an Expression to Control Alpha Offset	755
Creating Custom Camera Rigs	758
Swivel Camera Rig	758
Swivel Camera Rig Asset	760
Applying Depth of Field and Motion Blur	764
Rendering Using Depth of Field	764
Creating a Rack Focus Rig	767
Adding Motion Blur to an Animation	771
Using Orthographic and Stereo Cameras	774
Orthographic Cameras	774
Stereo Cameras	775
Using the Camera Sequencer	778
The Bottom Line	782

Appendixes

Appendix A • The Bottom Line

783

Chapter 1: Working in Autodesk Maya

783

Chapter 2: Introduction to Animation

784

Chapter 3: Hard-Surface Modeling

785

Chapter 4: Organic Modeling

786

Chapter 5: Rigging and Muscle Systems

787

Chapter 6: Animation Techniques

788

Chapter 7: Lighting with mental ray

789

Chapter 8: mental ray Shading Techniques

790

Chapter 9: Texture Mapping

791

Chapter 10: Paint Effects

792

Chapter 11: Rendering for Compositing

794

Chapter 12: Introducing nParticles

795

Chapter 13: Dynamic Effects

796

Chapter 14: Hair and Clothing

797

Chapter 15: Maya Fluids

798

Chapter 16: Scene Management and Virtual Filmmaking

799

Appendix B • Autodesk Maya 2016 Certification

803

Index

807

Introduction

The Autodesk® Maya® program is big. It is really, really huge. The book you hold in your hands, and all the exercises within it, represents a mere sliver of what can be created in Maya. Mastering Maya takes years of study and practice. I have been using Maya almost every day since 1999, and I'm still constantly facing new challenges and making new discoveries.

This book is meant to be a guide to help you not only understand Maya but also learn about Maya. The title *Mastering Autodesk Maya 2016* implies an active engagement with the software. This book is packed with hands-on tutorials. If you're looking for a quick-reference guide that simply describes each and every button, control, and tool in the Maya interface, turn to the Maya documentation that comes with the software. This book is not a description of Maya; it is an explanation illustrated with practical examples.

The skills you acquire through the examples in this book should prepare you for using Maya in a professional environment. To that end, some features, such as lighting and rendering with mental ray®, nDynamics, Fluids, and Maya Muscle, have received more emphasis and attention than others. Features that have not changed significantly over the past few versions of the software, such as Maya Software rendering, standard Maya shaders, and older rigging techniques, receive less attention since they have been thoroughly covered elsewhere.

When you read this book and work through the exercises, do not hesitate to use the Maya help files. I won't be insulted! The Maya documentation has a very useful search function that allows you to find complete descriptions of each control in the software. To use the help files, click the Help menu in the Maya menu interface. The documentation consists of a large library of Maya resources, which will appear in your default web browser when you access the help files. Experienced Maya artists never hesitate to use the help files to find more information about the software; there is no shame in asking questions! In addition, hovering over a tool or setting will give you a brief description. Features new to Maya, highlighted in green throughout the interface, have links to larger descriptions as well as movies.

Who Should Buy This Book

This book is written for intermediate Maya users and users who are advanced in some aspects of Maya and want to learn more about other facets of the program. The book is intended for artists who are familiar with Maya and the Maya interface or those who have significant experience using similar 3D packages. If you have used older versions of Maya, this book will help you catch up on the features in Maya 2016.

If you have never used Maya or any other 3D software on a computer before, this book will be too challenging and you will quickly become frustrated. You are encouraged to read

Introducing Autodesk Maya 2016 by Dariush Derakhshani (Sybex, 2015) or to read through the tutorials in the Maya documentation before diving into this book.

You should be familiar with the following before reading this book:

- ◆ The Maya interface.
- ◆ Computer image basics such as color channels, masking, resolution, and image compression.
- ◆ Computer animation basics, such as keyframes, squash and stretch, and 3D coordinate systems.
- ◆ Standard Maya shaders, such as the Blinn, Phong, Lambert, Layered, and Anisotropic materials, as well as standard textures, such as Fractal, Ramp, Noise, and Checker.
- ◆ Lighting and rendering with standard Maya lights and the Maya software renderer.
- ◆ The basics of working with NURBS curves, polygon surfaces, and NURBS surfaces.
- ◆ Your operating system. You need to be familiar with opening and saving files and the like. Basic computer networking skills are helpful as well.

What's Inside

The topics in this book move in a progressive order from introductory to complex. They also loosely follow a typical production pipeline for starting and completing assets. The following are brief explanations of the contents of each chapter.

There is also a companion website, which is home to all of the project files and samples referenced in the book. Go to www.sybex.com/go/masteringmaya2016, and click the Downloads tab to access the files.

- ◆ **Chapter 1: Working in Autodesk Maya** This chapter discusses how to work with the various nodes and the node structure that make up a scene. Using the Hypergraph, Outliner, Hypershade, Attribute Editor, and Connection Editor to build relationships between nodes is demonstrated through a series of exercises.
- ◆ **Chapter 2: Introduction to Animation** This chapter demonstrates basic rigging with inverse kinematics as well as animating with keyframes, expressions, and constraints. Animation layers are explained.
- ◆ **Chapter 3: Hard-Surface Modeling** This chapter introduces the various types of surfaces you can use to model. It walks you through numerous approaches for modeling parts of a bicycle.
- ◆ **Chapter 4: Organic Modeling** This chapter focuses on building a humanoid mesh, using polygon and subdivision surface techniques. Smooth mesh polygons, creasing, and soft selection are demonstrated on various parts of the model.
- ◆ **Chapter 5: Rigging and Muscle Systems** This chapter explains joints, expands on inverse kinematics, and covers smooth binding and proper rigging techniques. Maya Muscle is introduced and demonstrated on a character's leg.
- ◆ **Chapter 6: Animation Techniques** This chapter takes you through the numerous deformation tools available in Maya. Creating a facial-animation rig using blend shapes

is demonstrated, along with texture deformers, nonlinear deformers, and the geometry cache. We also take a look at importing motion capture.

- ◆ **Chapter 7: Lighting with mental ray** This chapter demonstrates a variety of lighting tools and techniques that can be used when rendering scenes with mental ray. Indirect lighting using global illumination, Final Gathering, and the Physical Sun and Sky network are all demonstrated.
- ◆ **Chapter 8: mental ray Shading Techniques** This chapter describes commonly used mental ray shaders and how they can be employed to add material qualities that mimic real-world surfaces. Tips on how to use the shaders together as well as how to light and render them using mental ray are offered.
- ◆ **Chapter 9: Texture Mapping** This chapter demonstrates how to create UV texture coordinates for a giraffe. Applying textures painted in other software packages, such as Adobe Photoshop, is discussed, as are displacement and normal maps and subsurface scattering shaders.
- ◆ **Chapter 10: Paint Effects** This chapter provides a step-by-step demonstration of how to create a custom Paint Effects brush as well as how to animate and render with Paint Effects.
- ◆ **Chapter 11: Rendering for Compositing** This chapter introduces render layers and render passes, which can be used to split the various elements of a render into separate files that are then recombined in compositing software.
- ◆ **Chapter 12: Introducing nParticles** This chapter provides numerous examples of how to use nParticles. You'll use fluid behavior, particle meshes, internal force fields, and other techniques to create amazing effects.
- ◆ **Chapter 13: Dynamic Effects** This chapter demonstrates a variety of techniques that can be used with nCloth to create effects. Traditional rigid body dynamics are compared with nCloth, and combining nCloth and nParticles is illustrated.
- ◆ **Chapter 14: Hair and Clothing** This chapter discusses how to augment your Maya creatures and characters using XGen, Maya nHair, and nCloth. Using dynamic curves to create a rig for a dragon's tail is also demonstrated.
- ◆ **Chapter 15: Maya Fluids** This chapter explains how 2D and 3D fluids can be used to create smoke, cloud, and flame effects, and it provides a demonstration of how to render using the Ocean shader. Bifrost is introduced as a way of creating liquid simulation.
- ◆ **Chapter 16: Scene Management and Virtual Filmmaking** This chapter provides an in-depth discussion of the Maya virtual camera and its attributes. A number of exercises provide examples of standard and custom camera rigs. Stereo 3D cameras are also introduced. References and the Asset Editor are also introduced. These features aid with large Maya projects that are divided between teams of artists.
- ◆ **Appendix A: The Bottom Line** This appendix contains all of the solutions from the Master It section at the end of each chapter.
- ◆ **Appendix B: Autodesk Maya 2016 Certification** This appendix contains the Autodesk Maya 2016 Certified Professional Objectives table that lists the topic, exam objective, and chapter where the information can be found.

NOTE Go to www.autodesk.com/certification to find information about the Maya 2016 Certified Professional exam covered in this book, as well as other Maya certification exams.

Conventions

Navigating in Maya is slightly different in the Windows and Mac operating systems. You can navigate the Hypergraph by using the same hot-key combination that you use in the viewport: Alt+MMB-drag/Option+MMB-drag pans through the Hypergraph workspace, and Alt+RMB-drag/Option+RMB-drag zooms in and out. (MMB refers to the middle mouse button, and RMB refers to the right mouse button.)

It is also important to note that Maya uses three digits for values listed within its tools and editors. The book may show only one or two digits when the last one or two digits are 0.

FREE AUTODESK SOFTWARE FOR STUDENTS AND EDUCATORS

The Autodesk Education Community is an online resource with more than five million members that enables educators and students to download—for free (see website for terms and conditions)—the same software as used by professionals worldwide. You can also access additional tools and materials to help you design, visualize, and simulate ideas. Connect with other learners to stay current with the latest industry trends and get the most out of your designs. Get started today at www.autodesk.com/joinedu.

How to Contact the Author

You can contact me with questions, comments, or concerns through his website at www.speffects.com, where you can see other books and productions on which he has worked.

Sybex strives to keep you supplied with the latest tools and information that you need for your work. Please check this book's website at www.sybex.com/go/masteringmaya2016, where we'll post additional content and updates that supplement this book should the need arise.



Chapter 1

Working in Autodesk Maya

The Autodesk® Maya® working environment has evolved to accommodate the individual artist as well as a team of artists working in a production pipeline. The interface presents tools, controls, and data in an organized fashion to allow you to bring your fantastic creations to life easily.

Autodesk Maya 2016 introduces a new, flattened GUI. The 3D beveled look of the interface's icons has been removed in favor of a simpler 2D look. Many of the colors have been modified as well. As a result, the GUI is now more modern and streamlined.

Understanding the way Maya organizes data about the objects, animations, textures, lights, dynamics, and all of the other elements contained within the 3D environment of a scene is essential to understanding how the interface is organized. Maya uses what's known as the *Dependency Graph* to keep track of the various packets of data, called nodes, and how they affect each other. Any single element of a Maya scene consists of multiple nodes connected in a web, and each one of these nodes is dependent on another. The Maya interface consists of editing windows that allow you to connect these nodes in an intuitive way and edit the information contained within each node.

There is usually more than one way to accomplish a task in Maya. As you grow comfortable with the interface, you'll discover which editing windows best suit your working style.

This chapter is a brief overview of what professionals need to understand when working in Maya. You'll learn about the types of nodes with which you'll be working and how they can be created and edited in Maya. You'll also learn how to work with projects and scene data as well as the various windows, panels, and controls that make up the interface. This will help you, whether you are working alone or as part of a team of artists.

This chapter is about working with nodes, but it is not meant to be a comprehensive guide to each and every control in Maya. You will find that information in the Maya documentation. If you've never used Maya, I strongly encourage you to read the Maya documentation as well as *Introducing Autodesk Maya 2016* by Dariush Derakhshani (Sybex, 2016).

In this chapter, you will learn to

- ◆ Work with Color Management
- ◆ Understand transform and shape nodes
- ◆ Create a project

Color Management

New!

Autodesk Maya 2016 implements a new management system for controlling the way colors are displayed in the viewport and in Render View. The Color Management system is based

on Autodesk Color Management, or SynColor, which is shared across several Autodesk applications. Color Management lets you switch between sRGB and linear color space. You can also switch to many other common color space environments. The Color Management system makes it easy to render your images to be color-corrected within your favorite compositing package.

The Color Management controls are visible in the Viewport 2.0 viewport, in the render view, and also within your preferences. Figure 1.1 shows the Color Management controls from the viewport. The viewport controls are the same controls that are located in the render view. Figure 1.2 shows the Color Management settings in the Preferences window.

FIGURE 1.1

The Color Management controls within the viewport

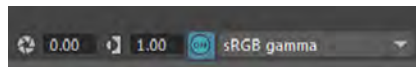
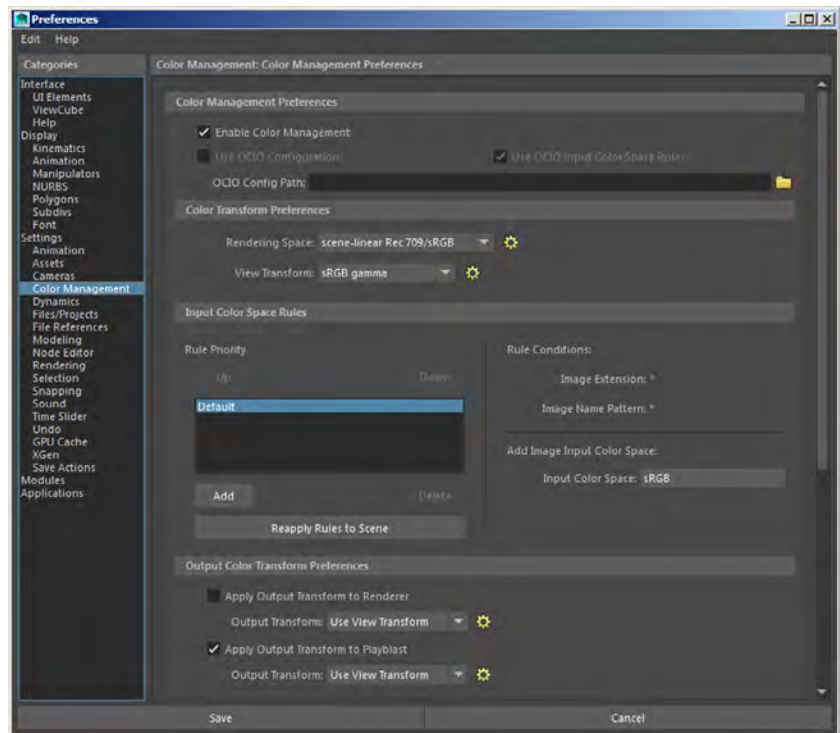


FIGURE 1.2

The Preferences for Color Management



Color Management affects all aspects of a production. In Maya 2016, you should establish your color space at the beginning of any project. To see more features and uses of Maya's Color Management system, you can watch `Color_Management.mov` in the `chapter1\movies` folder at the book's web page.

Creating and Editing Nodes

A Maya *scene* is a system of interconnected nodes that are packets of data. The data within a node tells the software what exists within the world of a Maya scene. The nodes are the building blocks that you, as the artist, put together to create the 3D scene and animation that will finally be rendered for the world to see. So if you can think of the objects in your scene, their motion, and their appearance as nodes, think of the Maya interface as the tools and controls that you use to connect those nodes. The relationship between these nodes is organized by the Dependency Graph (DG), which describes the hierarchical relationship between connected nodes. The interface provides many ways to view the graph, and these methods are described in this chapter.

Any given workflow in Maya is much like a route on a city map. You usually have many ways to get to your destination, and some of them make more sense than others, depending on where you're going. In Maya, the best workflow depends on what you're trying to achieve, and there is typically more than one possible ideal workflow.

Maya has many types of nodes that serve any number of different functions. All of the nodes in Maya are considered DG nodes. Let's say that you have a simple cube and you subdivide it once, thus quadrupling the number of faces that make up the cube. The information concerning how the cube has been subdivided is contained within a DG node that is connected to the cube node.

A special type of DG node is the *directed acyclic graph* (DAG) node. These nodes are made up of two specific types of connected nodes: transform and shape. The arrangement of DAG nodes consists of a hierarchy in which the shape node is a child of the transform node. Most of the objects with which you work in the Maya viewport, such as surface geometry (cubes, spheres, planes, and so on), are DAG nodes.

To understand the difference between the transform and shape node types, think of a transform node as describing where an object is located and a shape node as describing what an object is.

The simple polygon cube in Figure 1.3 consists of six flat squares attached at the edges to form a box. Each side of the cube is subdivided twice, creating four polygons per side. That basically describes what the object is, and the description of the object would be contained in the shape node. This simple polygon cube may be 4.174018 centimeters above the grid, rotated 35 degrees on the x-axis, and scaled four times its original size based on the cube's local x- and y-axes and six times its original size in the cube's local z-axis. That description would be in the transform node.

Maya has a number of workspaces that enable you to visualize and work with the nodes and their connections. The following sections describe how these workspaces work together when building a node network in a Maya scene.

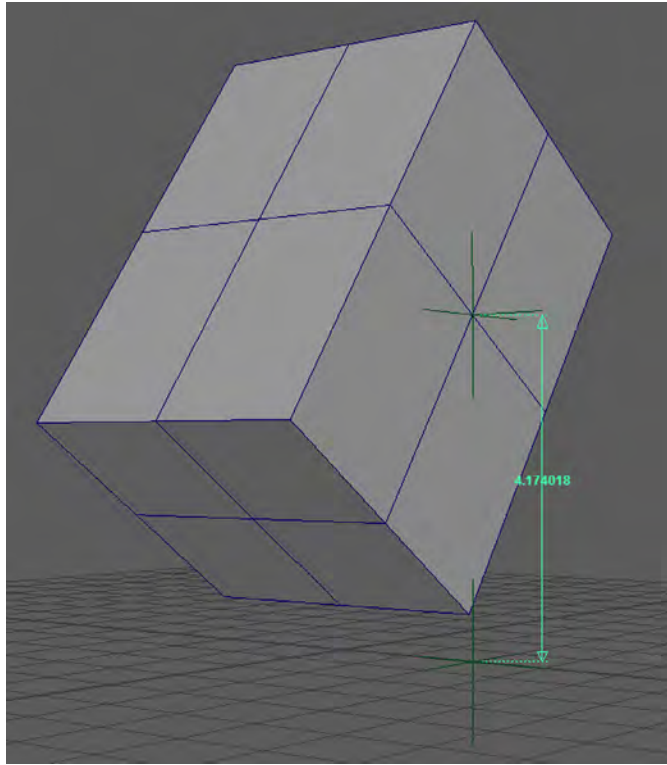
Using the Hypergraph

The *Hypergraph* is a visual representation of the nodes and their connections in Maya. A complex scene can look like an intricate web of these connections. When you need to know how a network of nodes is connected, the Hypergraph gives you the most detailed view. There are two ways to view the Hypergraph:

- ◆ The *hierarchy view* shows the relationships between nodes as a tree structure.
- ◆ The *connections view* shows how the nodes are connected as a web.

FIGURE 1.3

A shape node describes the shape of an object and how it has been constructed; a transform node describes where the object is located in the scene.



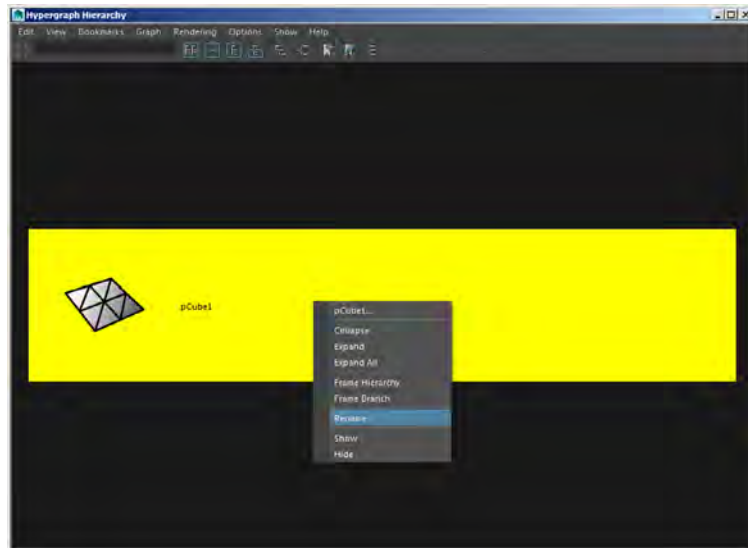
You can have more than one Hypergraph window open at the same time, but you are still looking at the same scene with the same nodes and connections.

This short exercise gives you a sense of how you would typically use the Hypergraph:

1. Create a new Maya scene.
2. Create a polygon cube by choosing **Create > Polygon Primitives > Cube**.
3. Select the cube in the viewport, and choose **Windows > Hypergraph: Hierarchy** to open the Hypergraph in Hierarchy mode.
4. Press **f** to frame your selection. You'll see a yellow rectangle on a black field labeled **pCube1**. The rectangle turns gray when deselected.
5. Move the mouse over the rectangle labeled **pCube1** and then right-click. Choose **Rename** from the pop-up window. Rename the cube **myCube** (see Figure 1.4).
6. Select **myCube** and, from the Hypergraph menu, choose **Graph > Input And Output Connections**. This switches the view to the connections view just as if you had originally opened the Hypergraph by choosing **Windows > Hypergraph: Connections**. It's the same Hypergraph, but the view mode has changed, allowing you to see more of the scene.

FIGURE 1.4

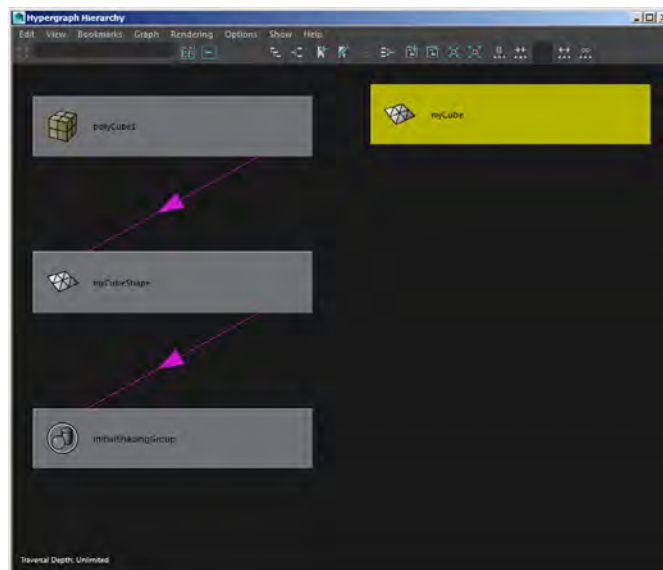
You can rename nodes in the Hypergraph through the context menu when you RMB-click.



When you graph the input and output connections, you see the connected nodes that make up an object and how the object appears in the scene. In the current view, you should see the myCube node next to a stack of connected nodes labeled polyCube1, myCubeShape, and initialShadingGroup, as shown in Figure 1.5. (The nodes may also be arranged in a line; the actual position of the nodes in the Hypergraph does not affect the nodes themselves.)

FIGURE 1.5

The node network appears in the Hypergraph. This shape node (myCubeShape) is connected to two other nodes, whereas the transform node (myCube) appears off to the side.



NAVIGATING THE HYPERGRAPH

You can navigate the Hypergraph by using the same hot-key combination that you use in the viewport: Alt+MMB-drag/Option+MMB-drag pans through the Hypergraph workspace, and Alt+RMB-drag/Option+RMB-drag zooms in and out. (MMB means clicking with the middle mouse button, and RMB means clicking with the right mouse button.) Selecting a node and pressing the **f** hot key focuses the view on the currently selected node. It is also possible to zoom in using the scroll wheel on your mouse.

The myCube node is the transform node. The myCubeShape node is the shape node. In the Hypergraph, the shape and transform nodes are depicted as unconnected; however, there is an implied connection, as you'll see later. This is demonstrated when you rename the myCube node; the shape node is renamed as well.

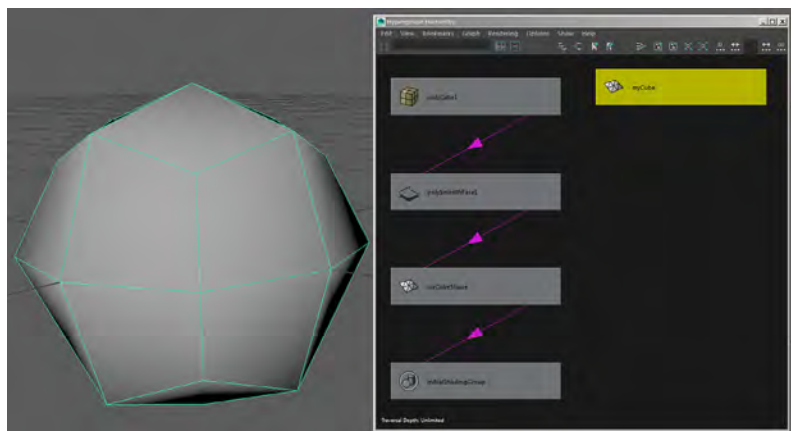
In Maya, the construction history feature stores a record of the changes used to create a particular node. The polyCube1 node is the construction history node for the myCubeShape node. When you first create a piece of geometry, you can set options to the number of subdivisions, spans, width, height, depth, and many other features that are stored as a record in this history node. Additional history nodes are added as you make changes to the node. You can go back and change these settings as long as the history node still exists. Deleting a history node makes all of the previous changes to the node permanent (however, deleting history is undoable). Use the following steps to guide you through the process of modifying history nodes:

1. Keep the Hypergraph open, but select the cube in the viewport.
2. Change the menu set in the upper left of the main interface to Modeling.
3. Press the **5** key on the keyboard to switch to Shaded mode. Choose Mesh > Smooth. The cube will be subdivided and smoothed in the viewport.

In the Hypergraph, you'll see a new polySmoothFace1 node between the polyCube1 node and the myCubeShape node (see Figure 1.6). This new node is part of the history of the cube.

FIGURE 1.6

Performing a smooth operation on the cube when construction history is activated causes a new polySmoothFace1 node to be inserted into the node network.

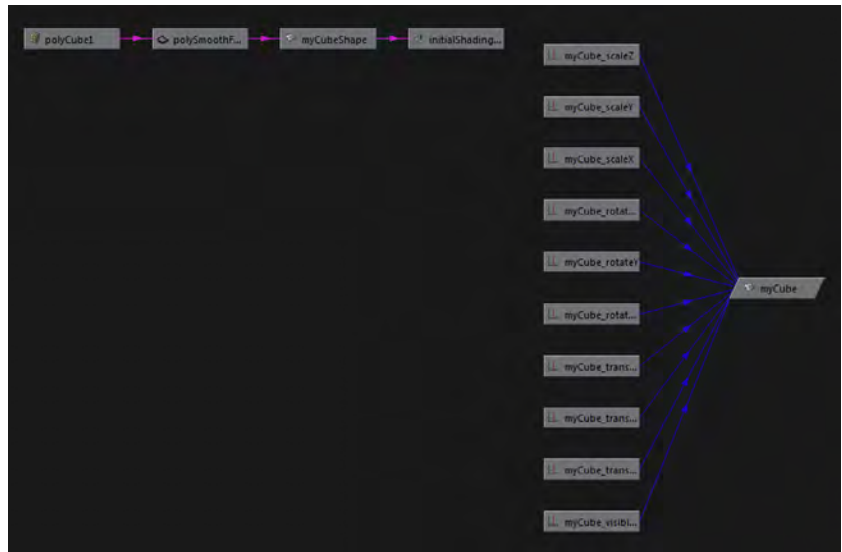


4. Select the polySmoothFace1 node, and delete it by pressing the Backspace key on the keyboard. The cube will return to its unsmoothed state.
5. Select the transform node (myCube), and press the **s** hot key. This creates a keyframe on all of the channels of the transform node. A keyframe stores the current attribute values at a particular time on the timeline. Animation is created by interpolating between keyframed values.

You'll see a new node icon appear for each keyframed channel with a connection to the transform node (see Figure 1.7).

FIGURE 1.7

The attributes of myCube's transform node have been keyframed. The keyframe nodes appear in the Hypergraph.



6. Hold the cursor over any line that connects one node to another. A label appears describing the output and input attributes indicated by the connection line.
7. It is common to have tens to hundreds of connections in the Hypergraph. To help accommodate numerous connections, you can change the layout of your nodes under Options > Graph Layout Style (see Figure 1.8).

WORKING WITH HISTORY

Over the course of a modeling session, the history for any given object can become quite long and complex. This can slow down performance. It's a good idea to delete history periodically on an object by selecting the object and choosing Edit > Delete By Type > History. You can also choose to delete the history of all of the objects in the scene at once by choosing Edit > Delete All By Type > History. Once you start animating a scene using deformers and joints, you can use the Delete By Type > Non-Deformer History option, which will remove the construction history nodes while preserving connections to animation nodes such as deformers.

Continues

(Continued)

You can turn off the history globally by clicking the Construction History toggle switch on the status line, as shown here:

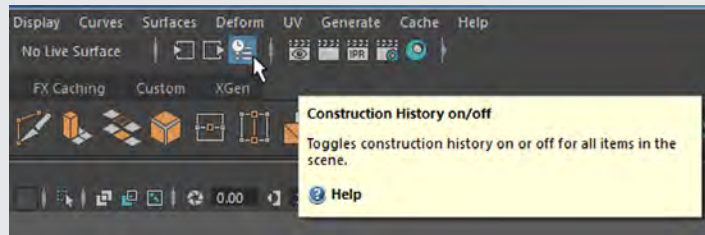
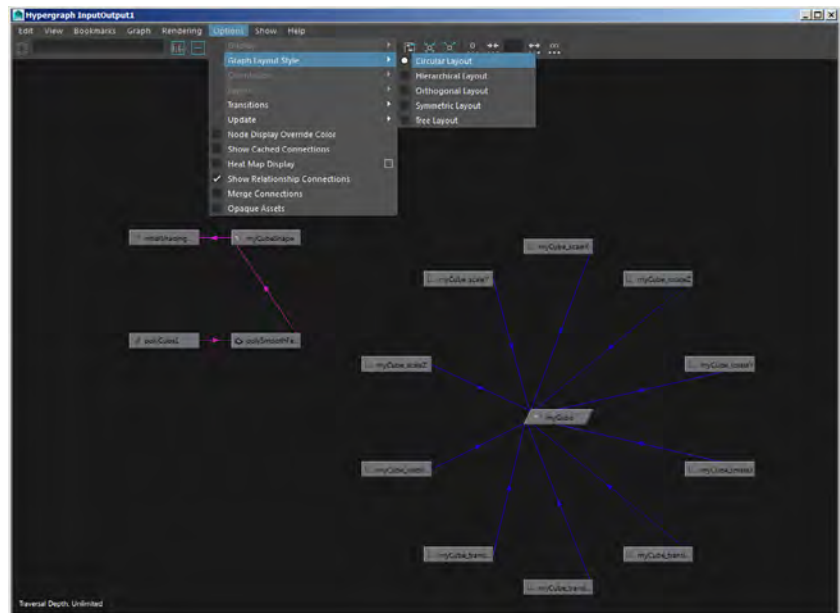


FIGURE 1.8

You can change the graph layout to accommodate numerous nodes and connections.



Connecting Nodes with the Node Editor

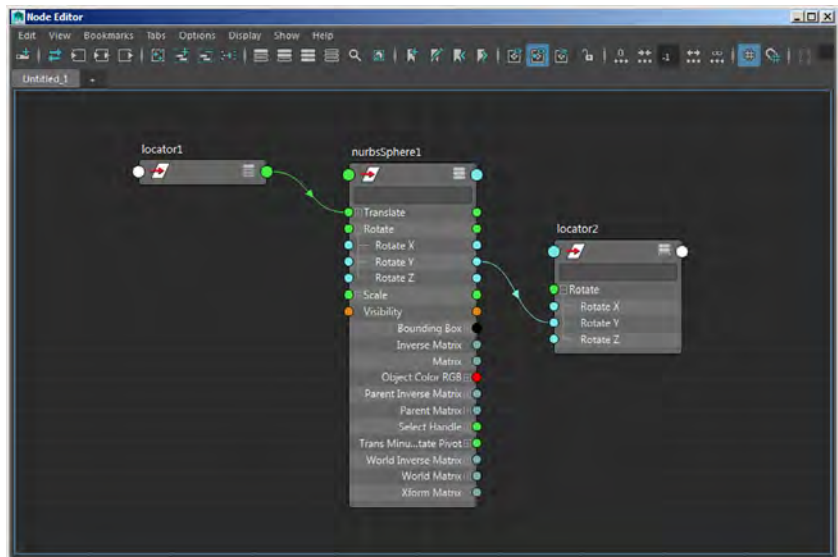
Connections between nodes can be added, deleted, or changed using the Hypergraph and the Connection Editor. The Node Editor combines features of the Hypergraph, Hypershade, and Connection Editor into a single graphical interface. Maya 2016 introduces a few changes and enhancements.

When you open the Node Editor you are presented with an empty, grid-lined space. To view a selected node, you can choose the type of connections you wish to graph: input, output, or both. After establishing a graph, you can add additional nodes by choosing the icon of the two-node hierarchy and plus symbol from the Node Editor toolbar.

Every node has a series of ports for connecting attributes between nodes. By default, the nodes are shown in Simple mode, meaning that none of their attributes or other ports are shown. In Simple mode, you can click the dot on either side of the node to access a pop-up menu for a node's input or output connections. When unconnected, the superport is white. After connecting, the port takes on the color of the connected attribute. You do not always have to use the superport and subsequent pop-up menu to make connections. You can expose the lesser ports by changing the nodes' display from their simple, default display to Connected, exposing the connected attributes, or to Full mode. Click the bars on the right side of each node to change its display. With the node selected, you can also press **1**, **2**, or **3** on the keyboard to change its mode. To change all of the nodes' modes at once, use the Edit menu at the top left of the Node Editor. Full mode allows you to see all of the connectable ports (see Figure 1.9).

FIGURE 1.9

The various display modes, starting with locator1 in Default, nurbsSphere1 in Full, and locator2 in Connected mode



As with all editors in Maya, you can customize the colors of the Node Editor using the Windows > Settings/Preferences > Color Settings window. The Attribute Types rollout under the Node Editor rollout allows you to change the color of the various types of connections.

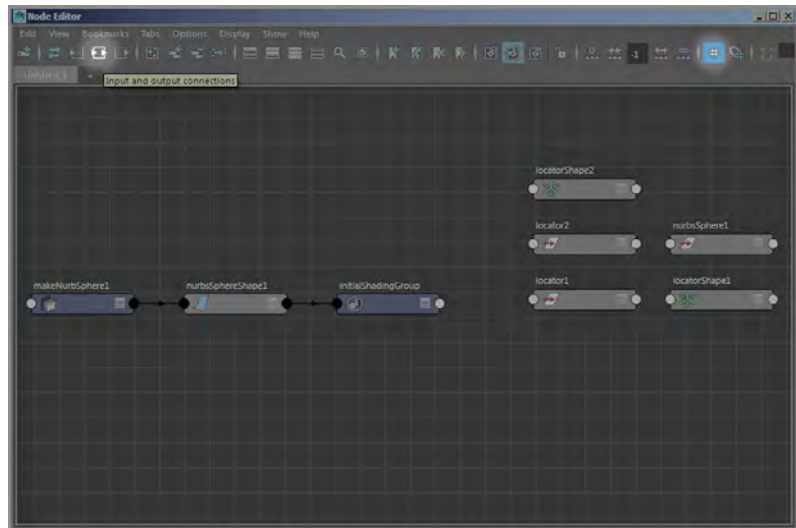
The following steps walk you through the basic uses of the Node Editor and how to make connections:

1. Start a new Maya scene.
2. Create a locator in the scene by choosing Create > Locator. A simple cross appears at the center of the grid in the viewport. This locator is a simple nonrendering null that indicates a point in space. Locators are handy tools that can be used for a wide variety of things in Maya.
3. Press the **w** hot key to switch to the Move tool; select the locator at the center of the grid, and move it out of the way.

4. Press the **g** hot key to create another locator. The **g** hot key repeats the last action that you performed, in this case the creation of the locator.
5. Create a NURBS sphere in the viewport by choosing **Create > NURBS Primitives > Sphere**.
6. Move the sphere away from the center of the grid so that you can clearly see both locators and the sphere.
7. Use the Select tool (hot key = **q**) to drag a selection marquee around all three objects.
8. Open the Node Editor by choosing **Windows > Node Editor**. A grid is drawn in the workspace. The grid can be toggled on or off by choosing the grid-visibility button (highlighted in Figure 1.10) from the Node Editor's toolbar. Choose the Input And Output Connections button, also highlighted in Figure 1.10, to graph your selection.

FIGURE 1.10

The input and output connections of the two locators and the sphere are graphed in the Node Editor.



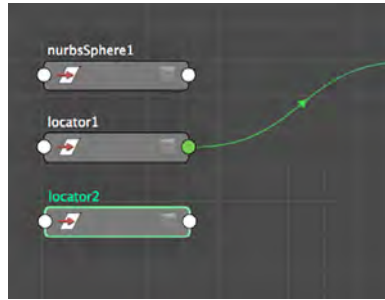
Stacked on top of one another are the two transform nodes for **locator1** and **locator2**. **locatorShape1** and **locatorShape2** are the shape nodes for the locators. **nurbsSphere1** is the transform node for the NURBS sphere. And **nurbsSphereShape1** is the shape node for the sphere; it's connected to **makeNurbSphere1**, which is the history node, and to **initialShadingGroup**. The **initialShadingGroup** node is the default shading group that is applied to all geometry; without this node, the geometry can't be shaded or rendered. When you apply a new shader to an object, the connection to **initialShadingGroup** is replaced with a connection to the new shader. A *shader* is a node that determines how a surface appears in Maya, as well as how it reacts to virtual lights.

9. Select the **locator1**, **locator2**, and **nurbsSphere1** transform nodes, and drag them away from the other nodes so that you can work on them in their own space. To keep your graph neat, you can use snap to grid to align your nodes with the grid. To activate, click the icon to the right of the grid icon.

10. Click the white superport on the right side of the locator1 node. This is the output.
11. From the pop-up menu, choose Translate > Translate. A yellow wire extends from the translate node (see Figure 1.11). The wire can be connected to a white port on another node.

FIGURE 1.11

Use the wire to connect the output of one node to the input of another.



12. Connect the yellow wire to the left side of the nurbsSphere1 node by clicking its white superport and choosing Translate from the pop-up menu. You can connect the yellow wire to either side of a node. The connection will be the same. A green wire shows the finished connection.

You can also choose Other from the pop-up menu. Doing so brings up the Input Selection window. The window lists the attributes of the node. Any of the attributes that have a plus sign next to them can be expanded to reveal nested attributes. For instance, find the Translate attribute and expand it by clicking the plus sign. (The term *translate* in Maya refers to an object's position. When you use the Move tool to change the position of an object in 3D space, you are "translating" the object.) You'll see that Translate has Translate X, Translate Y, and Translate Z. This means that you can choose either to select the Translate attribute, which will automatically use all three nested attributes as the input connection, or to expand Translate and choose one or more of the nested Translate X, Y, or Z attributes as the input connection. In some situations, a connection becomes unavailable (grayed out), indicating that the connection between the two attributes cannot be made, usually because the connection is not appropriate for the selected attributes (see Figure 1.12).

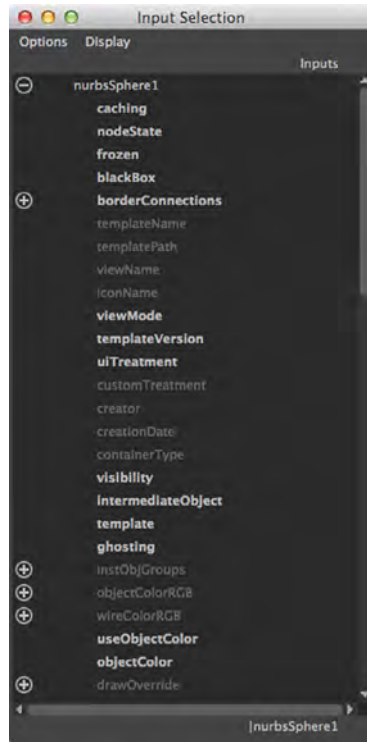
13. In the viewport, switch to Wireframe mode if you are not already in it. You can do this by pressing **4** on the keyboard or by clicking the wireframe icon on the icon bar at the top of the viewport window; the wireframe icon is the wireframe cube.
14. In the viewport, you'll notice that the sphere has snapped to the same position as the locator. Select the sphere, and try to move it using the Move tool (hot key = **w**). The sphere is locked to the locator, so it cannot be moved. Select the locator, and try to move it; the sphere moves with the locator. The output of the locator's Translate attributes is the input for the sphere's Translate.

INCOMING CONNECTIONS

In Wireframe view, an object will be highlighted in purple if it has an incoming connection from the selected object.

FIGURE 1.12

The Input Selection window specifies which attributes can be connected between nodes.



15. Click the `nurbsSphere1` node, and press **3** to display the node's attributes in full. Click the plus sign next to `Rotate` to expand the `Rotate` rollout. Repeat the procedure for `locator2`.
16. Click the right-side port or output for `Rotate X` on `nurbsSphere1`.
17. Drag the yellow wire to the left side (the input side) of `locator2`, and connect it to `Rotate Y`. The yellow wire turns cyan to match the color of its port of origin and the connection is made. The difference in color indicates rotational values as opposed to numeric values, like those used by the `Translate` and `Scale` attributes.
18. In the viewport, select the sphere and switch to the `Rotate` tool (hot key = **e**).
19. Drag up and down on the red circle of the tool to rotate the sphere around its x-axis. The locator rotates around its y-axis.

USING THE NODE EDITOR TO MAKE SIMPLE CONNECTIONS

The Node Editor is perfect for making one-to-one relationships between attributes on two nodes. In other words, the value of the output connection needs to equal the value of the input connection exactly. You can also create nodes from inside the editor by using the RMB and subsequent marking menu.

New!

The Node Editor can get cluttered quickly. To combat a messy graph, you can pin your nodes to their current position and size regardless of regraphing. To activate, choose Options ➤ Pin All Nodes By Default or RMB-click in the Node Editor to access the option through the marking menu.

You can also generate multiple tabs within the Node Editor to keep your workspace organized. Additional tabs allow you to view only the nodes on which you want to do work. For instance, you can create a new tab called NurbsSphere and only add the nodes related to the nurbsSphere1 object. Now the NURBS sphere is part of two different tabs. You can edit either one, and both tabs will update. You can create as many tabs as needed. All tabs are saved with your scene.

It is also possible to reduce the number of attributes displayed in each node to help simplify your view. To activate Edit Custom Attribute List, RMB-click a port of a node that you wish to customize. In Edit mode, the node is divided by a yellow bar. The attributes above the yellow bar are the ones that display. Those below the yellow bar are hidden. You can click an attribute to hide or unhide it. Attributes can also be rearranged by MMB-dragging to a new location within the node. To accept your changes, RMB-click the node's graphic and deselect Edit Custom Attribute List. You can then press 4 to see your edited node. To revert the node to its original settings, RMB-click any attribute in Edit mode and choose Revert from the context menu.

Experiment with making connections between the various attributes with the Node Editor. You can break a connection by selecting and dragging the arrow along the wire into empty space. Notice the changes in the port colors when making or breaking connections.

Creating Node Hierarchies in the Outliner



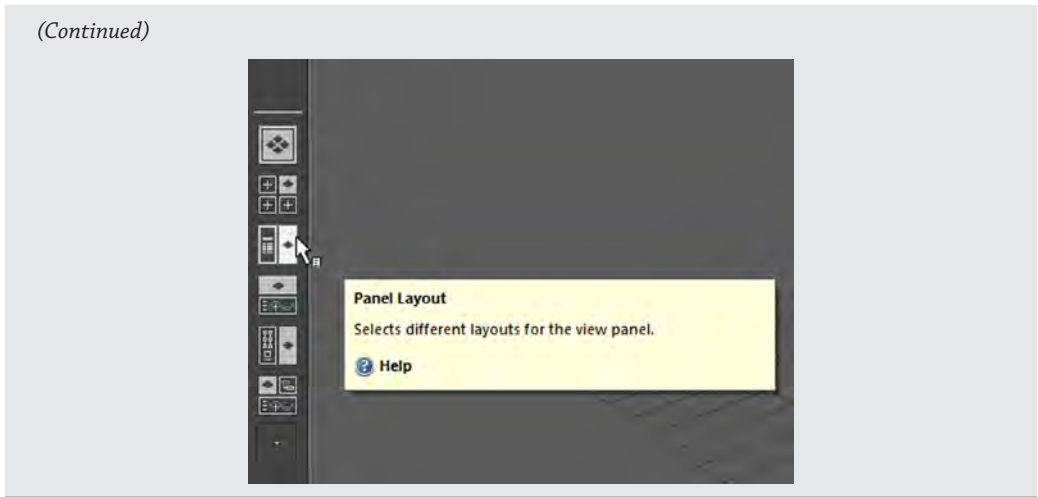
The Outliner shows a hierarchical list of the nodes in the scene in a form similar to the outline of a book. It is another way to view the transform and shape nodes in a scene and a way to create hierarchical relationships between nodes through parenting. The Outliner does not show the connections between nodes as the Hypergraph does; rather, it shows the hierarchy of the nodes in the scene. To see how this works, try the following exercise:

1. Open `miniGun_v01.ma` from the `Chapter1/scenes` directory at the book's web page (www.sybex.com/go/masteringmaya2016). The scene consists of a minigun model in three parts.
2. Open the Outliner by choosing Windows ➤ Outliner.

OUTLINER LAYOUT PRESETS

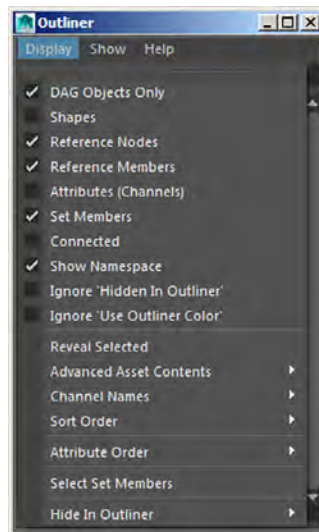
The Outliner can be opened as a separate panel or, like many of the panels in Maya, it can be opened in a viewport. A popular window arrangement is to split the viewports into two views, with the left view set to the Outliner and the right view set to the perspective view. You can open this arrangement by going to the menu bar in a viewport window and choosing Panels ➤ Saved Layouts ➤ Persp/Outliner. You can also click the third layout button on the left side of the interface just below the toolbox (as shown on the next page).

Continues

(Continued)

3. At the top of the Outliner is a menu bar. In the Display menu, make sure that DAG Objects Only is selected and that Shapes is deselected (see Figure 1.13).

FIGURE 1.13
The Display menu
at the top of the
Outliner



In the Outliner, you'll see three nodes listed—gunBarrels, housing, and mount—in addition to the four default cameras and several set nodes (don't worry about the set nodes). These are the three transform nodes for the pieces of the minigun. Select each node, and you'll see the corresponding part highlighted in the perspective view. At the moment, each piece is completely separate and unconnected.

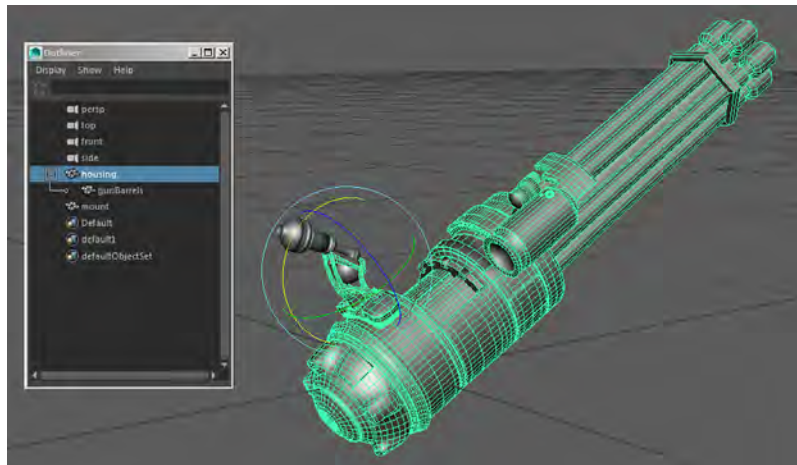
4. Select the housing node, and switch to the Rotate tool (hot key = **e**).
5. Rotate the objects; nothing else is affected. Try translating housing using the Move tool (hot key = **w**); again, nothing else is affected.
6. Use Undo (hot key = **Ctrl/Cmd+z**) a few times until the housing node returns to its original location and orientation.
7. In the Outliner, MMB-click the gunBarrels object. A white box appears around the gunBarrels name. Drag gunBarrels to housing; gunBarrels is parented to housing.

Parenting one object to another means making one transform node the child of the second. When an object is a child node, it inherits its position, rotation, scale, and visibility from the parent node. When you have multiple objects selected, the last object selected becomes the parent. In the Outliner, you'll notice that the housing node has a plus sign beside it and the gunBarrels node is not visible. The plus sign indicates that the node has a child node.

8. Click the plus sign next to the housing node to expand this two-node hierarchy. The gunBarrels node is now visible as the child of the housing node.
9. Select the housing node, and try rotating and translating it. The gunBarrels node follows the rotation and translation of the housing node (see Figure 1.14).

FIGURE 1.14

When the gunBarrels node is made a child of the housing object, it inherits changes made to the housing object's transform node.



Unlike the situation presented in the “Connecting Nodes with the Node Editor” section earlier in this chapter, the rotation and translation of the gunBarrels object are not locked to the rotation and translation of the housing node; rather, as a child, its rotation, translation, scale, and visibility are all relative to that of its parent.

10. Select the gunBarrels node, and try rotating and translating the object; then rotate and translate the housing node. You'll see the gun barrels maintain their position relative to the housing node. You could create an animation in which the gun barrels rotate on their own z-axis to spin around while firing at the same time the housing node is animated, rotating on all three axes in order to aim.

11. Press Undo a few times (hot key = **Ctrl/Cmd+z**) until both the housing and gunBarrels objects are back to their original positions.
12. In the Outliner, select the housing node and MMB-drag it on top of the mount node.
13. Click the plus signs next to the mount and housing nodes in the Outliner to expand the hierarchy. The lines indicate the organization of the hierarchy; the gunBarrels node is parented to the housing node, which is parented to the mount node.

SHIFT+CLICK TO EXPAND THE HIERARCHY

You can expand an entire hierarchy with one click in the Outliner. Just Shift+click the plus sign for the hierarchy that you want to expand.

14. Select the mount node, and choose Edit > Duplicate (hot key = **Ctrl/Cmd+d**). This makes a copy of the entire hierarchy. The duplicated mount node is named mount1.
15. Select the mount1 node, and switch to the Move tool (hot key = **w**). Pull on the tool's red arrow to move the duplicate along the x-axis about two units.
16. Select the mount node, and then Ctrl/Cmd+click the mount1 node in the Outliner.
17. Choose Edit > Group (hot key = **Ctrl/Cmd+g**) to group these two nodes under a single parent node.

A group node is a transform node that has no shape node. It's just a location in space used to organize a hierarchy. Like a parent node, its children inherit its rotation, translation, scale, and visibility.
18. Select the group1 node, and Shift+click the plus sign next to it in the Outliner to expand the group and all of its children.
19. Double-click the label for the group1 node in the Outliner to rename it; rename the group **guns**.

RENAMING NODES

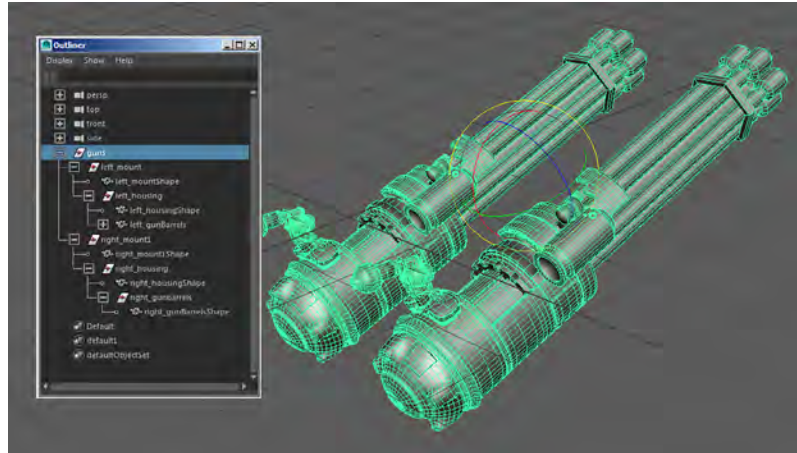
You'll notice that the duplicate mount node has been renamed mount1 automatically. Nodes on the same level of the hierarchy can't have the same name. The child nodes do have the same name, and this is usually a bad idea. It can confuse Maya when more complex connections are made between nodes. Whenever you encounter this situation, you should take the time to rename the child nodes so that everything in the scene has a unique name.

20. Select the mount1 node in the guns hierarchy, and choose Modify > Prefix Hierarchy Names.
21. In the pop-up window, type **right_**. This renames the top node and all of its children so that "right_" precedes the name. Do the same with the other mount node, but change the prefix to **left_**.

22. Select the guns group, and choose **Modify > Center Pivot**. This places the pivot at the center of the group. Try rotating the guns group, and you'll see both guns rotate together (see Figure 1.15).

FIGURE 1.15

The guns group is rotated as a single unit.



Each member of the hierarchy can have its own animation, so both gun barrels can rotate around their z-axes as they fire, the two housing nodes could be animated to aim in different directions, and the two guns could rotate as one unit, all at the same time. The entire group can be parented to another node that is part of a vehicle.

Displaying Options in the Outliner

There are several options in the Outliner for displaying nodes and their hierarchical arrangements. You can see that the default perspective, top, side, and front cameras are visible as nodes at the top of the Outliner. Also, a number of sets, such as the defaultLightSet, appear at the bottom of the Outliner. These sets are mainly used for organization of data by Maya and are usually not directly edited or altered.

1. In the Display menu of the Outliner, select the Shapes option to display the shape nodes of the objects. The shape nodes appear parented to their respective transform node. You can select either the transform node or the shape node in the Outliner to select the object.

ACCESSING OUTLINER OPTIONS

You can right-click in the Outliner to access the Outliner's display options quickly, rather than use the menu at the top of the Outliner.

2. In the Display menu, activate the visibility of attributes by selecting the Attributes (Channels) option.

Each node now has an expandable list of attributes. Most of the time you may want this option off, because it clutters the Outliner and there are other ways to get to these attributes. Ultimately, how you use these options is up to you.

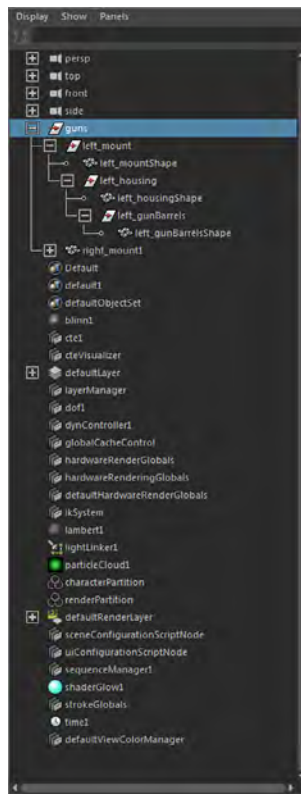
3. Turn off the Attributes display, and turn off the DAG Objects Only option. This allows you to see all of the nodes in the scene in the Outliner list, as opposed to just the DAG nodes.

DAG stands for “directed acyclic graph,” and DAG objects are those that have both a shape and a transform node. It’s not crucial to understand exactly what directed acyclic graph means as long as you understand that it is an arrangement in which a shape node is parented to a transform node. When you turn off DAG Objects Only in the Outliner, you’ll see all of the nodes in the Maya scene appear. Many of these are default utility nodes required to make Maya function, such as the layerManager node or the dynController1 node. Many other nodes appear when you create a new node or connection. An example of this is a keyframe or an expression node.

When you turn off DAG Objects Only, the list can get quite long. To find a node quickly, type the node’s name in the field at the very top of the Outliner. This hides all nodes except the named node. Clearing the field restores the visibility of all nodes in the Outliner (see Figure 1.16).

FIGURE 1.16

The Outliner can display shape nodes as well as other types of nodes in the scene.



Additional viewing options are available in the Show menu, which contains options for displaying only nodes of a certain type. Throughout this book, the Outliner will be used extensively, so you'll have lots of practice working with this panel.

4. In the Display menu of the Outliner, turn the DAG Objects Only option back on. Save the scene as **minGun_v02.ma**.

To see a finished version of the scene, open `miniGun_v02.ma` from the `chapter1\scenes` directory on the book's web page.

SEARCHING FEATURES IN THE OUTLINER

A complex scene in Maya can easily have hundreds of nodes. Just one character often has associated geometry, dozens of animation controls, joints, textures, and so on. Add another character to the scene with its own set of node networks, and the Outliner can become very cluttered very quickly. Establishing an organized naming system for the nodes in your scenes has many benefits, one of which is that you can use the search feature in the Outliner to filter what is displayed, thus making it easy to access the nodes that you need. Take the time to name your nodes in such a way as to make searching easy. The following illustrates how the search feature can be used in a complex scene.

Let's say that you have a scene with two complex characters, one named Larry and the other named Cecil. Both characters have similar rigs that use NURBS curves to control their animation rigs, and both have geometry, joints, shaders, and so on. When naming the nodes associated with each character, you make sure that all Larry nodes start with the name "larry." So, Larry's skin geometry might be named `larry_SKIN_GEO`, whereas his clothes would use names like `larry_PANTS_GEO`. Using capital letters in this case is purely a personal preference; the important thing is that the name of each node starts with `larry_`. Cecil would use the same convention; his skin geometry would be `cecil_SKIN_GEO`, and his pants would be `cecil_PANTS_GEO`. You end the names using `GEO` so that you know that this is a geometry node.

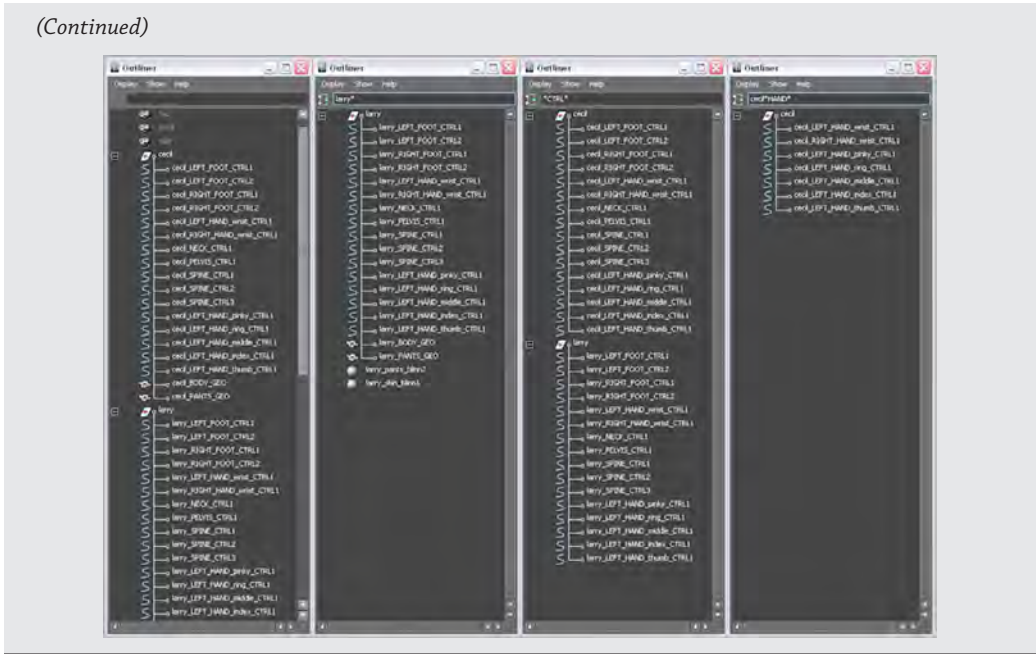
The controls for the animation rig use names like `larry_LEFT_HAND_wrist_CTRL1`, `larry_SPINE_CTRL1`, and `larry_NECK_CTRL1`. You get the idea. You can see that each of these nodes belongs to Larry, nodes for the left side of the body are clearly identified, the associated body part is identified, and they end with the letters `CTRL`. The same goes for Cecil.

Now here's where this type of organization, or something similar, is helpful in the Outliner. At the top of the Outliner is a blank field. To filter the nodes listed in the Outliner, you need to type some text and either precede or follow the text with an asterisk (*). The asterisk tells Maya to show all nodes that use the text before or after the asterisk in the name. So if you want to see all nodes associated with Larry, type **larry***. If you want to see all of the control nodes for both Cecil and Larry, type ***CTRL***. In this case, there may be text before and after the `CTRL` letters, so use an asterisk before and after `CTRL`. If you want to see the controls associated with Cecil's hands, type **cecil*HAND***, and so on.

The following images show variations on how to search through the Outliner with this method. If nothing appears in the Outliner when you type some text, check to see whether the asterisk is in the right place. To find one specific node, type its full name without the asterisk.

Continues

(Continued)



The Channel Box

Certification
Objective

The term *channel* is, for the most part, interchangeable with *attribute*. You can think of a channel as a container that holds the attribute's value. The *Channel Box* is an editor that lists a node's attributes for quick access. The Channel Box displays the node's attributes, which are most frequently keyframed for animation.

The Channel Box is located on the right side of the screen when the view mode at the end of the status bar is set to show the Channel Box/Layer Editor (see Figure 1.17).

FIGURE 1.17

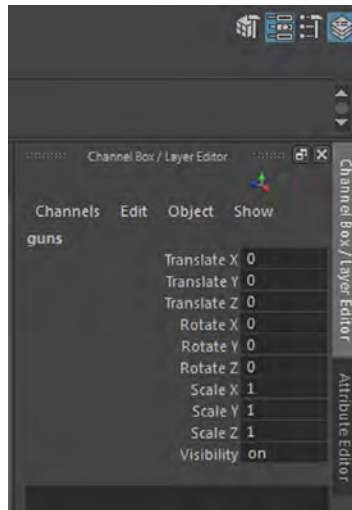
The icon in the upper right of the interface toggles the visibility of the Channel Box/Layer Editor.



Two tabs on the far right side of the screen allow you to switch quickly between the Channel Box and the Attribute Editor, as shown in Figure 1.18. (The Attribute Editor is discussed in detail later in this chapter.) These tabs are visible when both the Attribute Editor icon and the Channel Box icon are activated on the status bar in the upper-right corner of the interface.

FIGURE 1.18

The two tabs on the right side of the screen allow you to switch quickly between the Channel Box and the Attribute Editor.

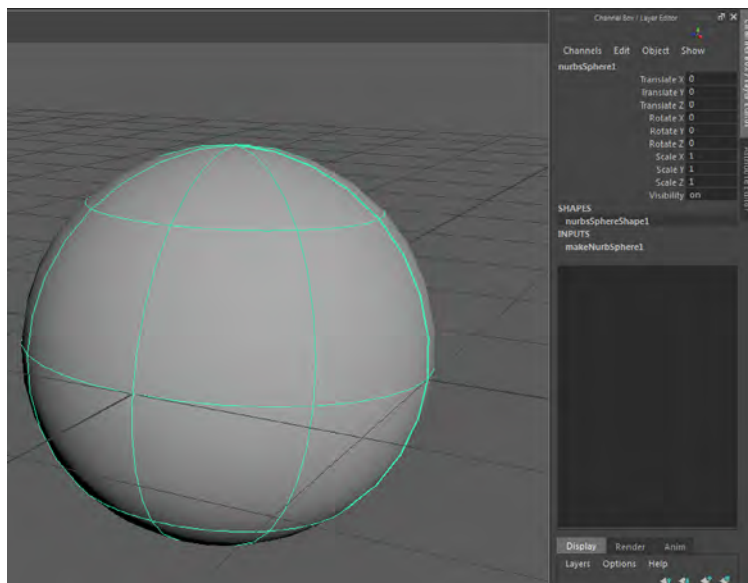


This exercise provides a quick tour of how to work in the Channel Box:

1. Create a new scene in Maya, and create a NURBS sphere (Create > NURBS Primitives > Sphere).
2. Make sure that the Channel Box is visible on the right side of the screen. To do this, click the icon at the farthest right of the status bar (shown earlier in Figure 1.18). This is a toggle to display the Channel Box. Click it until the Channel Box appears, as shown in Figure 1.19.

FIGURE 1.19

The Channel Box displays the channels for the currently selected object.



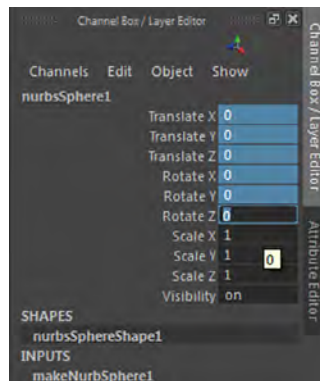
3. The Channel Box will list the currently selected object. Select the sphere, and you'll see `nurbsSphere1` appear. The list below it shows the attributes for `nurbsSphere1`'s transform node.

The lower half of the Channel Box lists the connections to this node. You'll see the name of the associated shape node under SHAPES, and below this is a section for the inputs. In this case, the input is the history node, named `makeNurbSphere1`, which contains the original settings used to create the sphere. If you delete history on the sphere, these attributes will no longer be accessible.

4. In the upper section of the Channel Box, under `nurbsSphere1`, try selecting the fields and inputting different values for Translate, Scale, and Rotate. The sphere updates its position, size, and orientation.
5. In the Visibility channel, select the word On in the field, and type 0. The sphere disappears. Input the value 1, and it reappears. Visibility is a Boolean, meaning that it is either on or off, 1 or 0.
6. Select the Translate X field so that it is highlighted. Shift+click the Rotate Z value, and all the values in between are also selected.
7. Type 0 in the Translate X field while they are selected, and press the Enter key. Doing so sets all of the Translate and Rotate values to the same value, places the sphere at the center of the grid, and returns it to its original orientation (see Figure 1.20).

FIGURE 1.20

You can quickly “zero out” the Translate and Rotate channels by Shift+clicking their fields and entering 0.



8. In the `makeNurbSphere1` section under INPUTS, highlight the Start Sweep channel. Enter a value of 90, and the sphere opens up. If this is hard to see, switch to shaded mode by pressing 5 on the keyboard. You're altering the construction history of the sphere so that it is no longer a closed surface.
9. Select the word Sections so that it is highlighted in blue. MMB-drag back and forth in the viewport. Doing so creates a virtual slider so that you can change the value of the field interactively instead of numerically. This should work for all of the channels (at least, most of the time).

10. Set the timeline to frame 1 by clicking the far left of the time slider where it is labeled 1, and press the **s** hot key. You'll see all of the channels turn red, indicating that the channels have a keyframe at the current time. The **s** hot key keyframes all of the available channels.
11. Move the timeline to frame 24. Notice that the channels' colors are now pink, indicating that the channels are animated but that they are not keyed at the current time. Change some settings both on the transform node (the upper half of the Channel Box) and under makeNurbSphere1 in the INPUTS section.
12. Press the **s** hot key again to set another key. Play the animation, and you'll see the sphere update based on the keyframed changes.

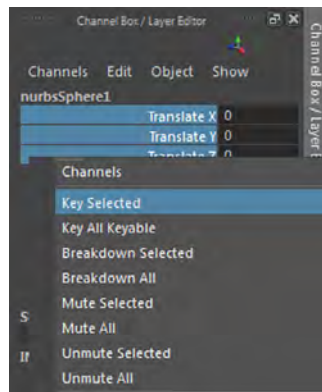
If the animation seems to play too quickly, you need to change the preferences so that playback matches the frame speed of the animation. To do so, choose Windows > Settings/Preferences > Preferences. In the Preferences window, choose Time Slider on the left column and set PlayBack Speed to Real-Time [24 FPS].

The **s** hot key keyframes everything, even those channels that you may not need to keyframe. You can use the Channel Box to keyframe specific channels.

13. Rewind the timeline, and choose Edit > Keys > Delete Keys to remove all of the keyframes on the sphere.
14. Highlight Translate X and Shift+click Translate Z so that the translation channels are all selected.
15. Right-click these values, and choose Key Selected (see Figure 1.21).

FIGURE 1.21

Right-click the selected channels, and choose Key Selected to keyframe just those specific channels.



16. Move to frame 24, and enter different values in the Translate fields.
17. Shift+click the Translate fields in the Channel Box, right-click, and choose Key Selected. This places a keyframe on just the selected channels—often a cleaner and more efficient way to work because you're placing keyframes only on the channels that you need to animate and not on every keyable channel, which is what happens when you use the **s** hot key. Now try playing the animation.

BE THRIFTY WITH KEYFRAMES

Creating extra, unnecessary keys leads to a lot of problems, especially when you start to refine the animation on the Graph Editor (discussed in Chapter 2, “Introduction to Animation.”) Keyframes also can increase the scene size (the amount of storage space the scene uses on disk). Be cheap with your keyframes, and use the Key Selected feature to keyframe only the channels that you need. Avoid using the **s** hot key to create keys on everything.

- 18.** To remove keys, you can highlight the channels, right-click, and choose Break Connections. This removes any inputs to those channels. The values for the current keyframe will remain in the channels.

The channels are color-coded to show what kind of input drives the channel:

- ◆ Red indicates a keyframe.
- ◆ Pink indicates that animation exists.
- ◆ Purple indicates an expression.
- ◆ Yellow indicates a connection (as in a connection from another node or channel, made in the Connection Editor or Node Editor).
- ◆ Brown indicates a muted channel.
- ◆ Gray means that the channel is locked.

LOCKING AND MUTING CHANNELS

You can mute a channel by right-clicking it and choosing Mute Selected from the context menu. When you mute a channel, the keyframes on that channel are temporarily disabled; as long as the channel is muted, the animation will not update. This is useful when you want to disable the keyframes in a channel so that you can focus on other aspects of the animation. Locking a channel is another option available when you right-click selected channels in the Channel Box. A locked channel prevents you from adding keyframes to a channel regardless of whether it has been animated. Creating animation is examined further in Chapter 2.

The Channel Box will be explored throughout the book and used frequently, particularly in the chapters concerning animation.

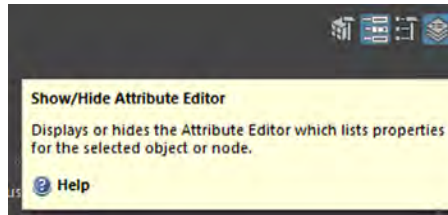
The Attribute Editor

The Attribute Editor is a tabbed panel that gives detailed information and access to a node’s attributes. The tabs at the top of the editor allow you to move between the attributes of all of the upstream (input) and downstream (output) connected nodes. This exercise provides a brief tour of how to use the Attribute Editor:

1. Create a new scene in Maya. Create a polygon cube on the grid (Create > Polygon Primitives > Cube).
2. Select the cube, and open its Attribute Editor. There are several ways to do this:
 - ◆ Right-click and hold the right mouse button over the cube, and then choose pCube1 from the marking menu.
 - ◆ Select the cube, and choose Windows > Attribute Editor.
 - ◆ Click the Show/Hide Attribute Editor icon in the upper right of the Maya interface (Figure 1.22).

FIGURE 1.22

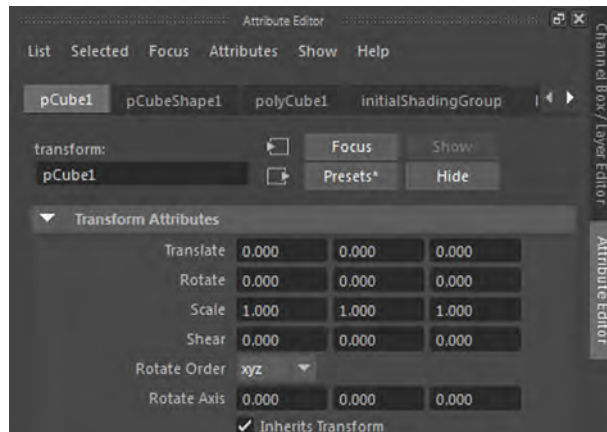
The Show/Hide Attribute Editor icon resides in the upper-right corner of the Maya interface.



- ◆ Press Ctrl+a on the keyboard. This toggles between the Channel Box and Attribute Editor if they're docked.
3. With the Attribute Editor open, choose the pCube1 tab at the top (Figure 1.23). The panel that opens contains the attributes for the cube's transform node, much like the upper section of the Channel Box described in the previous section. It also contains options for setting limits on the transform attributes.

FIGURE 1.23

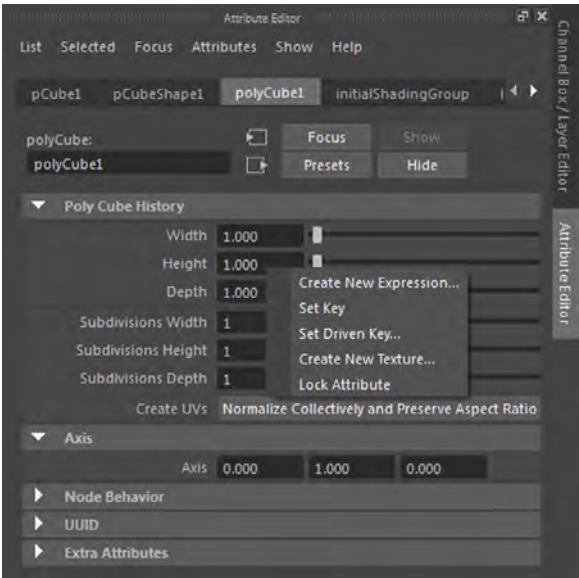
The Attribute Editor contains tabs that allow you to move through the connected nodes of a network.



Many of the settings can be accessed through the Attribute Editor's rollout panels. These are collapsible sections of grouped settings.

- 4. In the Attribute Editor, on the pCube1 tab, click the triangle next to mental ray. This reveals mental ray–specific settings related to the cube. Note that there are subsections under mental ray that are also collapsible.
- 5. Choose the pCubeShape1 tab at the top of the Attribute Editor. This tab contains settings related to the shape node. For example, expand the Render Stats section and you’ll see a list of settings that control how the shape will appear in a render.
- 6. Choose the polyCube1 tab, and you’ll see the construction history settings. If you delete history on the cube, this tab will no longer appear.
- 7. Expand the Poly Cube History rollout. If you right-click any of the fields, you get a menu that offers options such as creating expressions, key setting, or locking, much like the fields in the Channel Box (see Figure 1.24).

FIGURE 1.24
Right-clicking an attribute field reveals a menu with options for animating the attribute value.



- 8. In the Subdivisions Width field, highlight the text and type =. Doing so lets Maya know that you want to add an expression to the field.
- 9. Complete the expression by typing 9*2 after the equals sign (see Figure 1.25); then press the Enter key. Doing so adds an expression to this attribute that makes the Subdivisions Width value equal to 18. Note that the field turns purple and that the slider can no longer be moved.

FIGURE 1.25
You can enter simple mathematical expressions directly into a field in the Attribute Editor.



Note that a new tab called `expression1` is added to the top of the Attribute Editor; this is a new expression node that is now part of the cube's node network.

If the number of connected nodes is too large to fit within the tab listing at the top, you can use the two arrow buttons to the right of the tabs to move back and forth between the tab listings. Likewise, if not all connections are visible, you can use the Go To Input and Go To Output Connection buttons to the right of the field indicating the node name.

The Notes field at the bottom is useful for typing your own notes if you need to keep track of particular settings or leave a message for yourself or other users (see Figure 1.26). You can collapse this section by dragging the bar above it downward, thus making more room for the other settings in the Attribute Editor.

FIGURE 1.26

Messages can be entered in the Notes section at the bottom of the Attribute Editor.



LOAD ATTRIBUTES

You can use the Load Attributes button at the bottom of the Attribute Editor if the attribute display needs to be refreshed. Maya automatically updates the editor when new attributes are added, but occasionally it misses an update and needs to be refreshed.

The Attribute Editor is the workhorse panel of Maya. We will use it constantly throughout this book. Make sure that you are comfortable with the core concepts of how to switch between node settings using the tabs as well as how to change the available values.

Working with Shader Nodes in the Hypershade

New!

The *Hypershade* provides a visual display of how nodes in a Maya scene are connected. For Maya 2016, the Hypershade is much improved, and it is now combined with the Node Editor window. The Hypershade is mostly concerned with *shaders*—nodes used to define the color and material properties of renderable objects in a scene. These include materials (also known as shaders), textures, lights, cameras, and shading utilities. However, it is not unusual to use the Hypershade Work Area to make connections between other types of nodes as well. In this exercise, you'll use the Hypershade to connect several types of nodes.

1. Create a new scene in Maya with a NURBS cone.
2. Switch to smooth-shaded mode by pressing **6** on the keyboard, or click the Smooth Shade All and Textured icons on the viewport's menu bar (see Figure 1.27).

FIGURE 1.27

The Maya viewport menu bar allows you to choose between shading modes by toggling buttons.

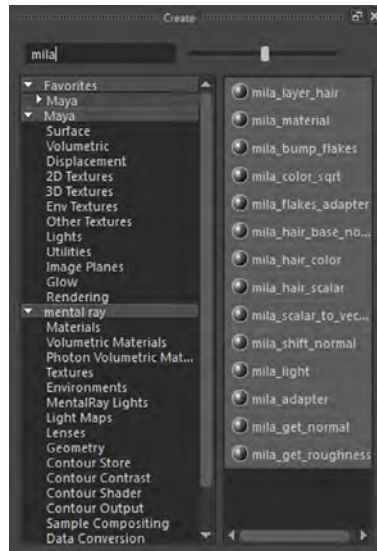


3. Open the Hypershade by choosing Windows > Rendering Editors > Hypershade. Alternatively, you can select the Hypershade icon from the main toolbar. It is a sphere with an extruded hole.

The Hypershade Editor is made up of several windows. The Browser window contains a visual display of the nodes in the scene. It is organized by tabs named Materials, Textures, Utilities, Rendering, Lights, Cameras, Shading Groups, Bake Sets, Projects, and Asset Nodes. If you want to access all of the file textures used in the scene, you can choose the Textures tab to see them listed with preview icons. Under the Browser window is the Create window. This is also referred to as the Create Render Node window, as you will see throughout the book. The Create window contains a list and a visual menu of the nodes that you can create in the Hypershade. The list is divided into sections for the Maya nodes, mental ray nodes, and a customizable list for your own favorites at the very top. Clicking a category in the list filters the node-creation buttons to the right of the list, which helps to cut down on the amount of time that you need to hunt for specific nodes. To add a node to the Favorites list, MMB-drag a node button from the right on top of the Favorites menu. You can also search through the nodes by typing in the field at the very top of the list. For example, typing **mila** in this field filters the node-creation buttons so that all the mila (mental images layers) nodes are displayed (see Figure 1.28).

FIGURE 1.28

The text field at the top of the Create tab allows you to filter the list of buttons. MMB-dragging a button on top of the Favorites section adds the node to the Favorites.

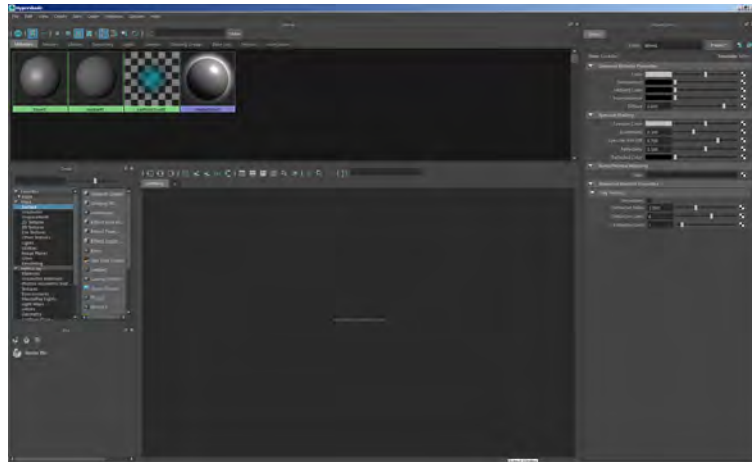


Beneath the Create window is the Bins window. Bins allow you to organize your shaders into separate groups. Add shaders to a bin by MMB-dragging from the Browser window to a desired bin. The Browser window automatically updates based on the bin that you have selected. The Master Bin displays all of the shaders.

The right side of the Hypershade contains the Property Editor. The Property Editor is similar to the Attribute Editor, but it is optimized only to show attributes that directly affect the look of your shader.

Centered between these windows is the Node Editor. Figure 1.29 shows the default Hypershade layout.

FIGURE 1.29
The Hypershade organizes render nodes and offers a workspace for constructing render node networks.



4. Under the Create window, click Surface in the list of Maya nodes. Click the Blinn button to create a new Blinn material.

You can see the new blinn1 material in the Browser; its attributes appear in the Property Editor.

DEFAULT SCENE MATERIALS

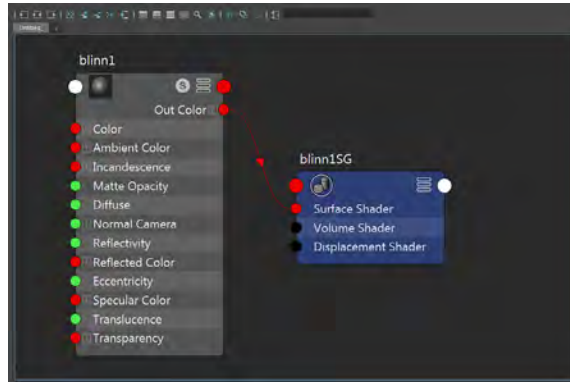
All Maya scenes start with three materials already created: lambert1, particleCloud1, and shaderGlow1. The lambert1 material is the default material applied to all newly created geometry, the particleCloud1 material is a special material reserved for particle-cloud objects, and the shaderGlow1 node sets the glow options for all shaders in the scene.

5. Select the blinn1 material in the Browser window and, from the icon bar at the top of the Node Editor, choose Input And Output Connections. This displays all of the upstream and downstream nodes connected to blinn1. Upstream nodes are nodes that plug into a node and affect its attributes; downstream nodes are ones that are affected by the node.

The blinn1SG node is a downstream node, known as a *shader group*, connected to blinn1. All materials have a shader group node connected to them. This node is a required part of the network that defines how the shader is applied to a surface, and it is often used when creating complex mental ray shader networks and overrides (see Figure 1.30).

FIGURE 1.30

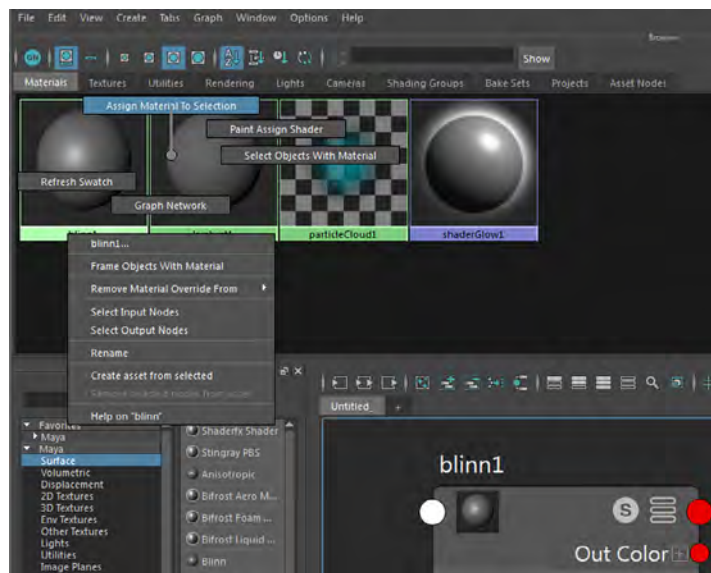
Shaders all have shading group nodes attached, which define how the shader is applied to the geometry.



6. In the viewport, select the cone. You can apply the blinn1 material to the cone in three ways:
 - ◆ MMB-drag the material icon from the Hypershade on top of the cone in the viewport window.
 - ◆ Select the cone, right-click the blinn1 node in the Hypershade, and choose Assign Material To Selection (see Figure 1.31).

FIGURE 1.31

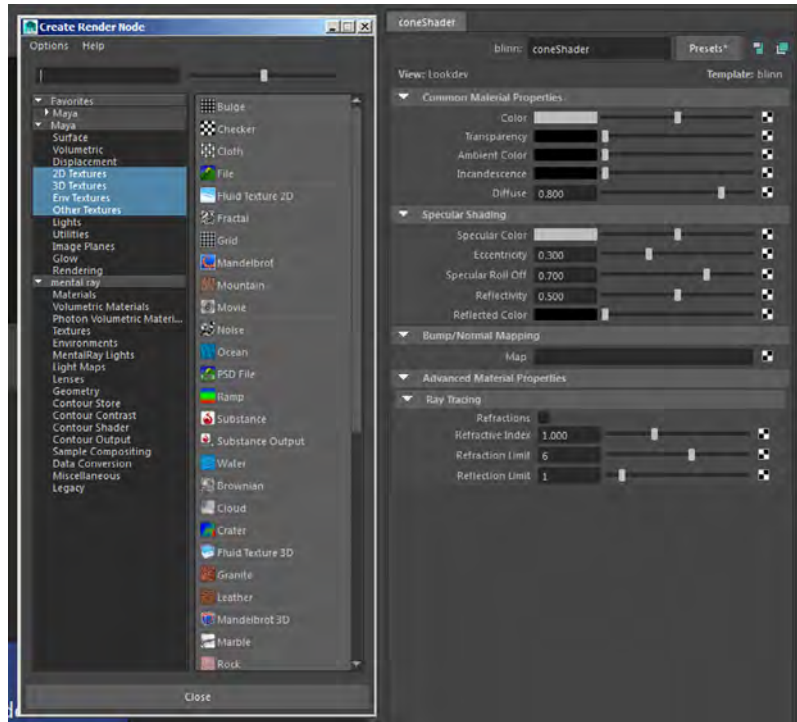
Right-click a shader, and drag upward on the marking menu to choose Assign Material To Selection.



- ◆ Right-click the surface in the viewport, and choose Assign New Material to create a new shader, or choose Assign Existing Material to assign a shader that you've already created.
- 7. In the Property Editor for the blinn1 node, rename the material **coneShader** by typing in the field at the top of the editor.
- 8. Click the checked box to the right of the Color slider. This opens the Create Render Node window (see Figure 1.32).

FIGURE 1.32

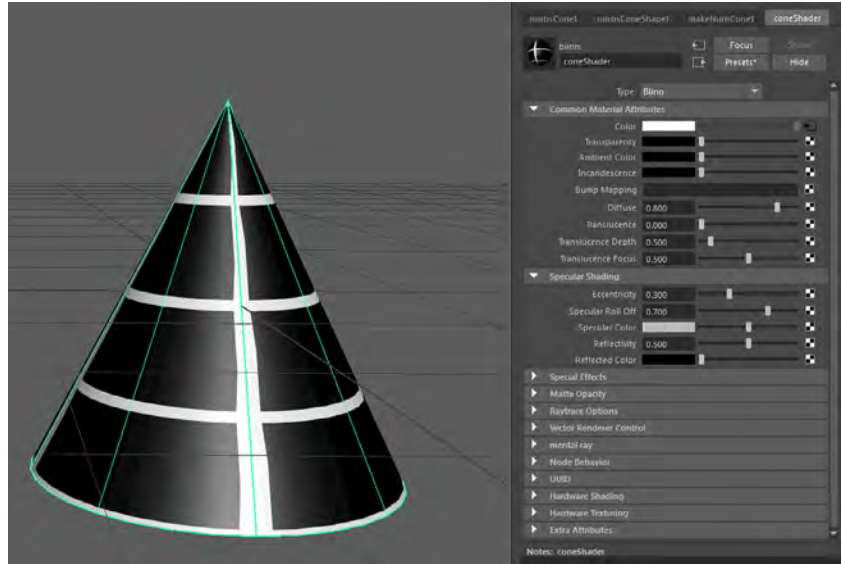
Click the checked box next to the Color slider to open the Create Render Node window.



- 9. Select 2D Textures from the list on the left, and click the Grid button on the right side of the panel to create a grid texture; this is applied to the color channel of the cone, and it is visible on the cone in the viewport when textured and smooth-shaded mode is on (hot key = 6) (see Figure 1.33).
- 10. Select coneShader in the Node Editor of the Hypershade. Choose the Input And Output Connections icon. You'll see that the coneShader node now has the grid1 texture node as well as the place2dTexture1 node attached.
- 11. The Hypershade uses a lot of screen real estate. Switching back and forth to the viewport in order to see changes to your shader is tedious. To alleviate this problem, you can add another window to your Hypershade. From the Hypershade, choose Window > Material Viewer.

FIGURE 1.33

The grid texture appears on the cone when the perspective view is set to Shaded mode.



12. A new window appears that contains a proxy object with your shader mapped to it. Drag the window to the dotted lines above the Property Editor. When you get within range, the interface shifts, making room to dock the Material Viewer. Drop it so that it docks itself above the Property Editor.
13. Hover your mouse over the black line between the Property Editor and Material Viewer. The cursor will change to two opposing arrows. Click and drag to resize the windows. Make the Material Viewer larger to get a better view of the proxy object (see Figure 1.34).
14. You can quickly make changes to your shader and have them reflected on the proxy object in the Material Viewer window. The proxy shader ball object can be changed to numerous other types of objects to display your shader better. Choose Cloth from the center drop-down menu to switch from the shader ball object to a draped cloth object.
15. Final renders are rarely done with default lighting. Therefore, it makes sense to test your shader with a more detailed lighting model. You first need to change the renderer from Hardware to mental ray. The Material Viewer will begin an iterative render process that still allows you to work. Next, choose Interior3Color from the right drop-down menu. Figure 1.35 shows the results.
16. Click the place2dTexture1 node from the Node Editor. Its attributes are displayed in the Property Editor.
17. Type **0.5** in the first field next to Coverage and press Enter. This reduces the coverage of the grid texture in U space by one-half.

FIGURE 1.34

The Material Viewer gives you a rendered representation of your shader.

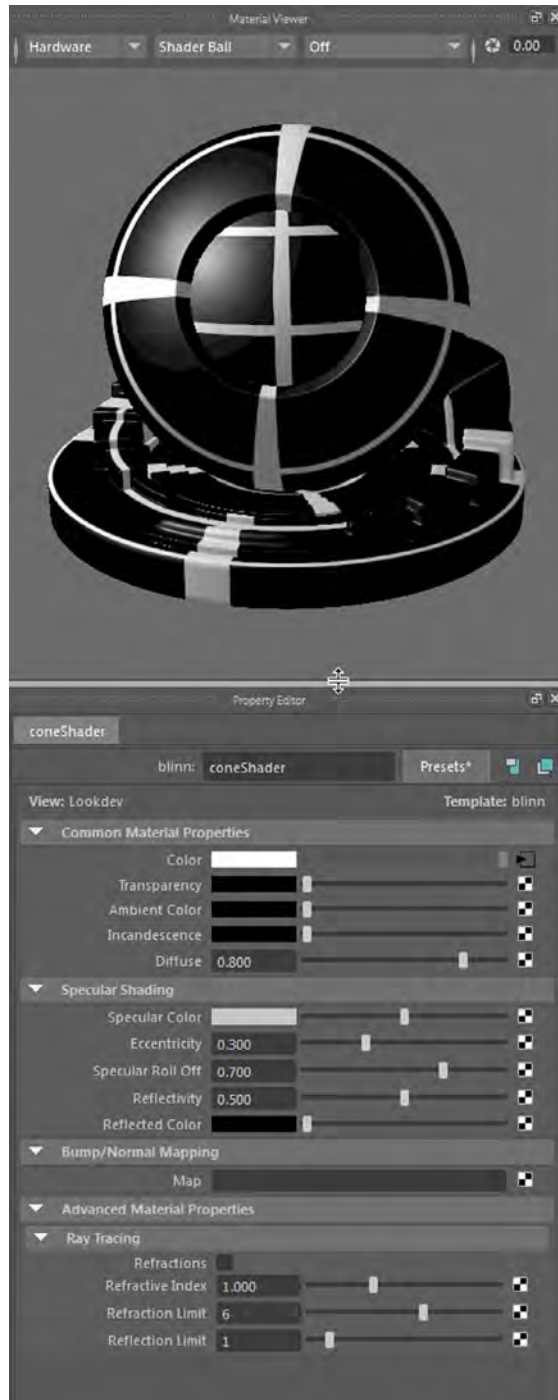


FIGURE 1.35

Change the Material Viewer to use the mental ray renderer and a detailed lighting model.



NURBS UV TEXTURE COORDINATES

NURBS surfaces have their U and V texture coordinates based on the parameterization of the surface, unlike polygon meshes, which require defined UV coordinates. You can use the attributes in the place2dTexture1 node to position textures on NURBS surfaces.

18. Select the grid1 node in the Node Editor.
19. Expand the Color Balance rollout, and click the color swatch next to Default Color. This opens the Color History. Set the color to **red**.

DEFAULT COLOR

The default color of the texture is the color “behind” the grid texture. Any part of the surface that is not covered by the grid (based on the settings in the place2dTexture1 node) will use the default color.

20. In the left panel of the Hypershade, select 2D Textures in the Maya list. MMB-drag the ramp texture all the way to the color swatch next to the Filler Color in the grid’s Property Editor.

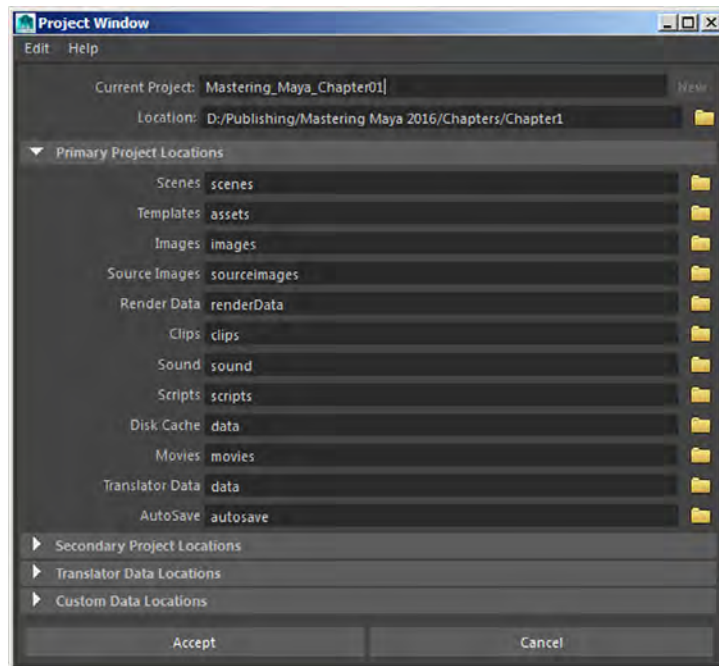
Creating a New Project

Creating a new project is simple. Projects can be created on your computer's primary hard drive, a secondary drive, or a network drive. The scene files used for each chapter in this book are stored in their own project directories at www.sybex.com/go/masteringmaya2016. Maya uses a default project directory when one has not been specified. This is located in your My Documents\maya\projects folder in Windows. As an example, you'll create a project directory structure for the examples used in this chapter.

1. Start a new Maya scene. You'll note that an empty scene is created when you start Maya.
2. Choose File > Project Window.
3. The Project Window dialog box opens. Click the New button to the right of the Current Project field. In the Current Project field, type **Mastering_Maya_Chapter01** (see Figure 1.37).

FIGURE 1.37

The Project Window dialog box lets you set the location of a new project directory and determine its directory structure on your hard drive.



4. To the right of the Location field, click the folder icon to browse your computer. The Select Location window opens and lets you determine where on your computer or network you want the project to be stored, or you can simply type the path to the directory. The project folder can be a subfolder of another folder if you like.

In the Primary Project Locations section, you'll see a large number of labeled fields. The labels indicate the various types of files a Maya scene may or may not use. The fields indicate the path to the subdirectory where these types of files will be located.

The fields contain the name of the subdirectory relative to the project file. Thus when you choose to use the default settings, all Maya scene files (files with the `.mb` or `.ma` file extension) will be stored in a folder labeled Scenes. The path to that folder will be, in this example, `Mastering_Maya_Chapter01\Scenes`.

EDITING DEFAULT FILE LOCATIONS

If you decide that you want to store the scene files in a different directory, you can type the path to that directory in the field or click the folder icon to the right of the field to set a different folder path.

5. Click Accept. Maya will take a few moments to create the project directory and all subfolders on the specified drive.
6. Use your computer's file browser to locate the new project; then expand the folder, and you'll see all of the subfolders.

Editing and Changing Projects

You may not need to use every folder that Maya creates for you, or you may decide to change where Maya looks for elements such as file textures. If you're working on a number of projects, you may need to switch the current project. All of these options are available in the Maya File menu.

1. To edit the current project, choose **File > Project Window**. The Project Window dialog box opens with all of the paths to the project subdirectories. You can type a new directory path in any one of these fields or click the folder icon to browse your computer's directory. Then click the Accept button. If a folder does not already exist, Maya will create it for you.

RELINKING FILES AFTER CHANGING PROJECT SETTINGS

If you edit the project, Maya will look in the newly specified folders from this point on, but files used prior to editing the project will not be copied or moved. You'll need to move these files using the computer's file browser if you want Maya to find them easily after editing the project.

2. To switch projects, you can choose **File > Set Project** or choose a project listed in the Recent Projects menu.

When working on a project with a number of other animators, you can choose to share the same project, which is a bit risky, or each animator can create their own project directory structure within a shared folder. The latter approach is a little safer because it prevents two people from having the same project open or overwriting each other's work. Later in this book, you'll learn how multiple animators can share parts of the scene using file references.

It is possible to work on a scene file outside the current project. This usually happens when you forget to set the project using the File > Set Project option. Make a habit of setting the current project each time that you start to work on a scene; otherwise, linked files such as textures, dynamic caches, and file references can become broken, causing the scene to behave unpredictably (which is a nice way of saying that the scene will fall apart and possibly crash Maya).

While working with the project files for this book, you'll want to copy the entire project to your local drive and then use the Project Window dialog box to choose the corresponding chapter project directory as the current project. This way, all of the linked files in the scenes should behave correctly and the exercises will function.

OVERRIDING PROJECT SETTINGS

You can choose to override a project setting for an individual scene element. For instance, by default Maya looks to the `sourceimages` directory for file textures. However, when you create a file texture node, you can use the Browse button to reference a file anywhere on your machine or the network. This is usually not a great idea; it defeats the purpose of organizing the files in the first place, and it can easily lead to broken links between the scene and the texture file. It's a better idea to move all file textures used in the scene to the `sourceimages` directory or whatever directory is specified in your project settings.

The Bottom Line

Work with Color Management. Color Management lets you switch between sRGB and linear color space. You can also switch to many other common color space environments. The Color Management system makes it easy to render your images to be color-corrected within your favorite compositing package.

Master It Establish a color space for a new project.

Understand transform and shape nodes. DAG nodes have both a transform node and a shape node. The transform node tells where an object is located; the shape node describes how it is made. Nodes can be parented to each other to form a hierarchy.

Master It Arrange the nodes in the `miniGun_v03.ma` file in a hierarchical structure so that the barrels of the guns can rotate on their z-axis, the guns can be aimed independently, and the guns rotate with the turret.

Create a project. Creating a project directory structure keeps Maya scene files and connected external files organized to ensure that the animation project is efficient.

Master It Create a new project named `Test`, but make sure that the project has only the `Scene`, `Source Images`, and `Data` subfolders.



Chapter 2

Introduction to Animation

Computer animation has revolutionized the way stories are told. Whether the animation is used for entertainment, to advertise a product, or to demonstrate a complex scientific principle, the ability to construct virtual three-dimensional objects and then manipulate their movements over time has given the artist an unprecedented amount of power. Creative Autodesk® Maya® users will find that there are few limits to what they can accomplish through animation.

Animation in Maya is accomplished through a wide variety of tools and techniques. The goal of this chapter is to show you how to use the tools in Maya to animate the attributes of various nodes. Gaining an understanding of how you can solve problems in Maya is essential to telling your story through computer animation.

In this chapter, you will learn to

- ◆ Use inverse kinematics
- ◆ Animate with keyframes
- ◆ Use the Graph Editor
- ◆ Preview animations with a playblast
- ◆ Animate with motion paths
- ◆ Use animation layers

Using Joints and Constraints

Most of this chapter is devoted to exercises that animate a simple mechanical bug model. Animation and rigging are closely related skills. *Rigging* is the process in which controls are created to allow you to manipulate the three-dimensional objects in order to generate animation. Even if you don't intend to do much of your own rigging, you'll have an easier time understanding how to animate if you know what goes into creating a rig. Chapter 5, "Rigging and Muscle Systems," delves into more advanced rigging concepts. In this chapter, you'll learn about the basic tools and techniques for rigging to get you started.

Joint Basics

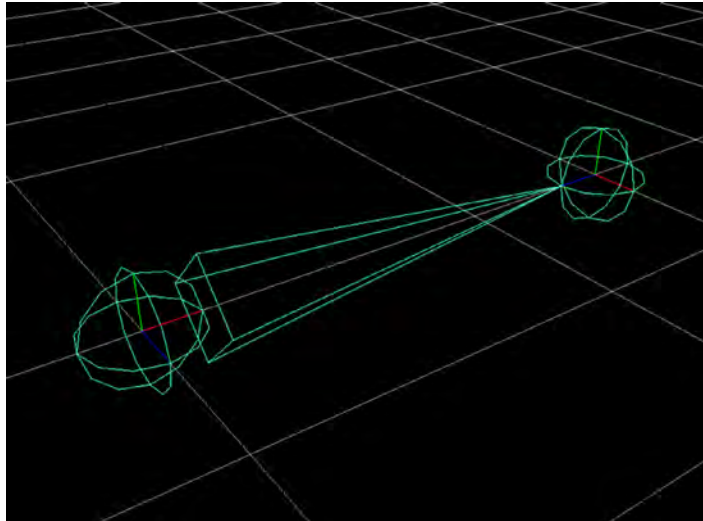
A *joint* is the basic animation control. It is essentially a point in space connected by a virtual bone, symbolized by an elongated pyramid. The geometry is bound to the joints in a process known as *skinning*. The skinned geometry is then deformed when the joints are moved to create such effects as the bend in an elbow or a knee.

When you are rigging mechanical objects and robots, it's not always necessary to use joints; you can parent the parts of the object together in an organized fashion and then set keyframes directly on the geometry. (This is demonstrated in Chapter 1, "Working in Autodesk Maya.") However, because of the built-in hierarchy of joints as well as the many animation controls available, using joints can make rigging and animating mechanical objects easier, even if you don't intend to bind or skin the geometry so that it is deformed by the joints.

You create joints using the Joint tool (from the Rigging menu set, choose Skeleton > Create Joints). A simple wire sphere represents a joint. As noted before, when one joint is connected to another, the space between the two joints is bridged with a wireframe pyramid shape referred to as a *bone*. The broad end of the bone is placed next to the parent joint, and the pointed end of the bone is pointed toward the child joint (see Figure 2.1).

FIGURE 2.1

Two joints are placed on the grid with a bone between them.



Joints are most often animated by rotating the parent joint or using *inverse kinematics*, which orients the joints based on the position of a goal called an *end effector*. This topic is discussed later in this chapter in the section "Inverse Kinematics." Joints can also be animated by placing keyframes on Translate or Scale channels. This approach, however, is slightly less common.

Joints are usually placed in a hierarchical relationship known as a *joint chain*. Joint chains can have many branches and numerous controls, depending on the complexity of the model. A series of joint chains is known as a *skeleton*.

A joint is a type of deformer that typically influences the shape of nearby components depending on how the components of the geometry are bound, or skinned, to the joints. Since the bug in these examples is mechanical, the joints do not need to deform the geometry of the model. You can simply parent the pieces of the leg geometry to the individual joints. Skinning geometry to joints is explored further in Chapter 5.

Point Constraints

Certification
Objective

A *point constraint* uses the world space position of one object to control the world space position of another object. World space coordinates tell exactly where an object is in relation to the rest of the scene. This is different from object space coordinates, which are relative to the object's initial position.

Certification
Objective

For instance, if you move an object on the grid, the Translate channels in the Channel Box indicate where an object is in object space. If you freeze the transformations on an object (Modify > Freeze Transformations) after moving the object, its Translate (and Rotate) channels become 0 in X, Y, and Z (Scale channels become 1 in X, Y, and Z), and its current position is now its starting position in object space.

If you create a joint in a scene and then reposition it and freeze transformations, you'll notice that the Rotate channels all become 0 for its new orientation; however, the Translate channels are not affected by freeze transformations. If you want to place a joint exactly at the position of another object, you can't rely on the object's Translate channels as an accurate description of the object's location in world space. One way to get around this is to use a point constraint to position the joint according to the object's world space coordinates.

Aim Constraints

Certification
Objective

An *aim constraint* constrains the orientation of an object relative to the position of one or more other objects. The following steps show you how to add aim constraints and use them to control parts of a mechanical bug:

1. Open the mechBugRig_v01.ma scene from the chapter2\scenes directory at the book's web page (www.sybex.com/go/masteringmaya2016).
2. Turn on the display layer for BODY; turn off the LEGS layer so that the legs are hidden and the body is displayed.
3. Create a locator (Create > Locator), and name it **eyeAimLoc**.
4. Set eyeAimLoc's Translate X, Y, and Z channels to 0, 0, and 2, respectively.
5. In the Outliner, expand bugBody group.
6. MMB-drag eyeAimLoc into the bugBody group.
7. Select eyeAimLoc, and choose Modify > Freeze Transformations so that the Translate channels become 0, 0, and 0.
8. Expand the frontBody group, the head group, and the face group.
9. Select eyeAimLoc, and Ctrl/Cmd+click the eyeBase group (see Figure 2.2).
10. From the Rigging menu set, choose Constrain > Aim > □.
11. In the Aim Constraint Options dialog box, choose Edit > Reset Settings to return the settings to the default.
12. Set the Aim Vector fields to 0, 0, and 1, respectively, so that the aim vector is set to the z-axis (see Figure 2.3).

FIGURE 2.2
Select the eyeAimLoc
locator and the
eyeBase object in the
Outliner.

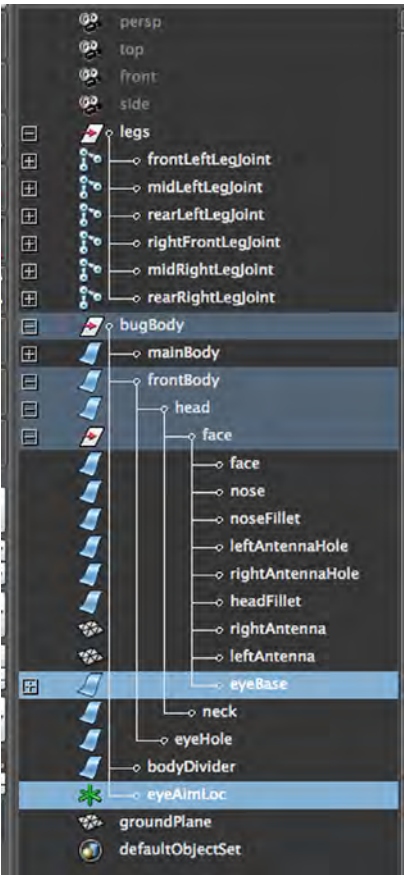
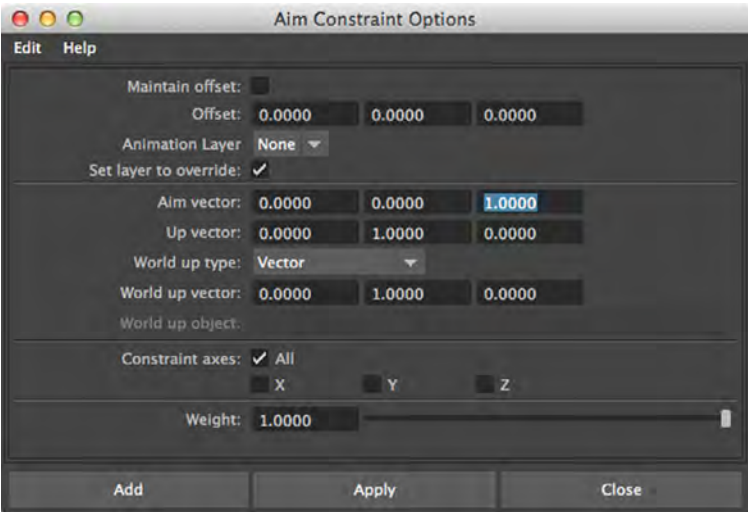
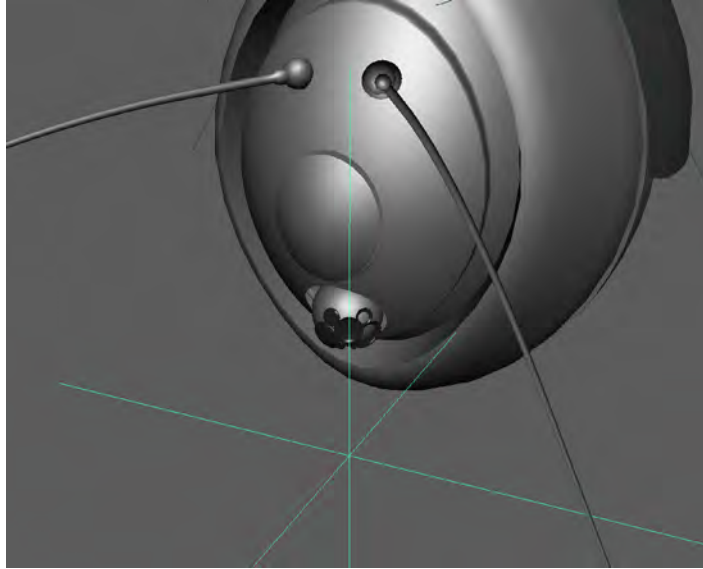


FIGURE 2.3
The Aim Constraint
Options window



13. Click Apply to create the constraint.
14. Move the locator around in the scene, and you'll see that the eyes follow it (see Figure 2.4). Because the locator is parented to the bugBody, when you animate the bug, the locator will move with it.

FIGURE 2.4
The eyes follow the position of the eyeAimLoc locator.

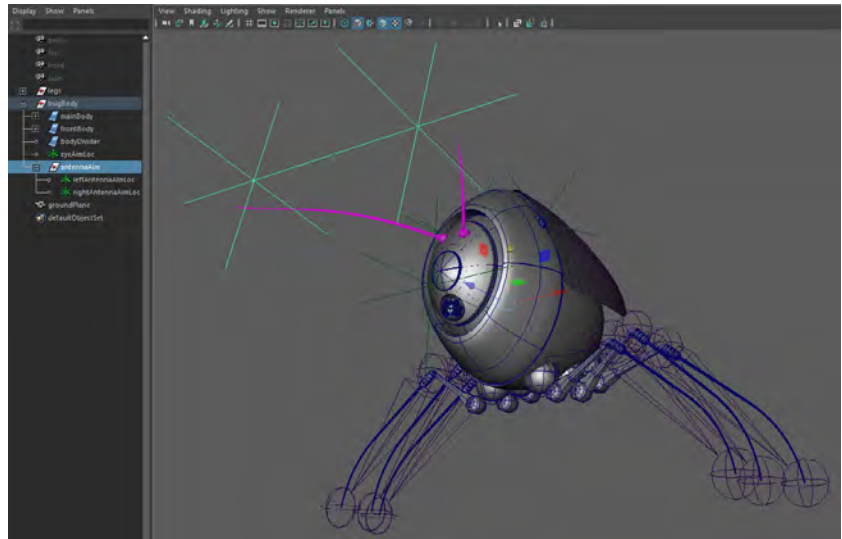


15. Create two new locators named **leftAntennaAimLoc** and **rightAntennaAimLoc**.
16. Set leftAntennaAimLoc's Translate X, Y, and Z to 1, 0, and 3, respectively, and set rightAntennaAimLoc's Translate X, Y, and Z to -1, 0, and 3 likewise.
17. Freeze transformations for both locators.
18. Shift+click both locators, and group them together. Name the group **antennaAim**. Leave the pivot point for the group at the center of the grid.
19. In the Outliner, MMB-drag antennaAim into the bugBody group.
20. Expand the antennaAim group, and make sure that the face group is expanded as well (in the bugBody/frontBody/head group).
21. Use an Aim constraint to constrain the leftAntenna to the leftAntennaAimLoc locator and the rightAntenna to the rightAntennaAimLoc locator.

When you move each antennaAimLoc locator, the antennae follow; when you rotate the antennaAim group, the locators move and the antennae follow. This gives you a way to animate the antennae separately and together (see Figure 2.5).
22. Save the scene as **mechBugRig_v02.ma**.

FIGURE 2.5

You can animate the antennae separately or together based on the grouping of the antennaAimLoc locators.



To see a version of the scene up to this point, open the `mechBugRig_v02.ma` scene from the `chapter2\scenes` directory at the book's web page.

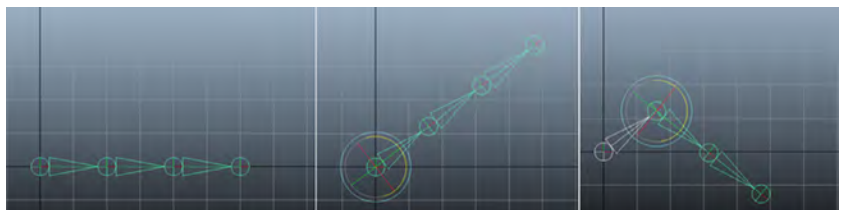
Inverse Kinematics

Kinematics is the study of the motion of objects. This is related to, but distinguished from, dynamics in that kinematics studies only the way in which objects move and not the cause of the objects' motion. In 3D computer graphics, the term *kinematics* describes how joints can be moved to animate objects and characters. There are two main types of kinematics: forward kinematics and inverse kinematics.

The term *forward kinematics* refers to a situation in which each joint in the chain inherits the motion of its parent joint. Thus, if you have four joints in a chain, when you rotate the root, the three child joints move based on the rotation of the root. When you rotate the second joint, the third and fourth joints inherit the motion of the second (see Figure 2.6).

FIGURE 2.6

When using forward kinematics, each joint inherits the rotation of its parent.



Forward kinematics can be useful in many situations; for instance, they are often used for basic arm animation for characters. However, they can be tedious and difficult to work with for other types of animation, particularly when you are animating the legs of a character walking or jumping. Constant adjustments have to be made to ensure that as the root joints are animated, the limbs of the character do not penetrate the floor or slide during a walk cycle.

Inverse kinematics (IK) cause the joints in a chain to orient themselves based on the position of a goal known as the *end effector* (see Figure 2.7). Inverse kinematics can be a more intuitive technique than forward kinematics in many situations. When used on legs, the Sticky option for inverse kinematics can prevent joints from penetrating the floor or sliding during a walk cycle. When animating an IK chain, you can simply change the position of the end effector using the IK Handle tool, and all of the joints in the chain will orient themselves based on the position of the end effector. The end effector itself is positioned using the IK Handle; the end effector is not actually visible or selectable in the viewport (it can be selected in the Hypergraph window).

FIGURE 2.7

Inverse kinematics cause the joints in a chain to rotate based on the position of a goal.



There are controls for blending between forward and inverse kinematics known as FK/IK Blend. You can switch between forward and inverse kinematics and even blend between the two. This topic is covered in more detail in Chapter 5.

The kinematic controls work very well for many standard situations. Most professional riggers prefer to create their own custom controls and solutions. Creating custom controls is discussed in Chapter 5. In this section, a basic, simple IK technique is used on the legs of the mechanical bug. This technique makes animating the legs easier and more intuitive because you only need to worry about the position of a single goal when animating all of the parts of one of the legs.

IK Handle Tool

In this section, you'll add inverse kinematics to each of the legs on the mechanical bug:

1. Continue with the scene from the previous section, or open the `mechBugRig_v02.ma` scene from the `chapter2\scenes` directory at the book's web page.
2. In the Layer Editor, turn off Visibility for the body layer.
3. From the Rigging menu set, choose **Skeleton > Create IK Handle > □**.
4. In the Tool Settings dialog box, set the Current Solver option to Single-Chain Solver. This solver works well for simple joint chains, such as the legs in this bug.

The other option is the Rotate Plane solver. This solver has extra controls that can be used to solve joint-flipping problems, which can occur with more complex joint arrangements. If you create an IK Handle using the Single-Chain solver and your joints behave unpredictably, try switching to the Rotate-Plane solver. (You can do this after creating the handle using the menu options in the IK Handle's Attribute Editor.) The various types of IK solvers are discussed in Chapter 5, "Rigging and Muscle Systems."

In general, when adding inverse kinematics to a joint chain using the IK Handle tool, you don't want the joints to be aligned in a straight line. There should be at least a small bend in the joint chain. This helps the solver figure out how the joints should bend as they attempt to rotate based on the position of the end effector. It's also a good idea to freeze transformations on the joints so that their X-, Y-, and Z-rotation channels are set to 0 before adding the IK Handle (in the `mechBug_v03.ma` scene, this has already been done).

5. Turn on Solver Enable, and turn off Snap Enable.

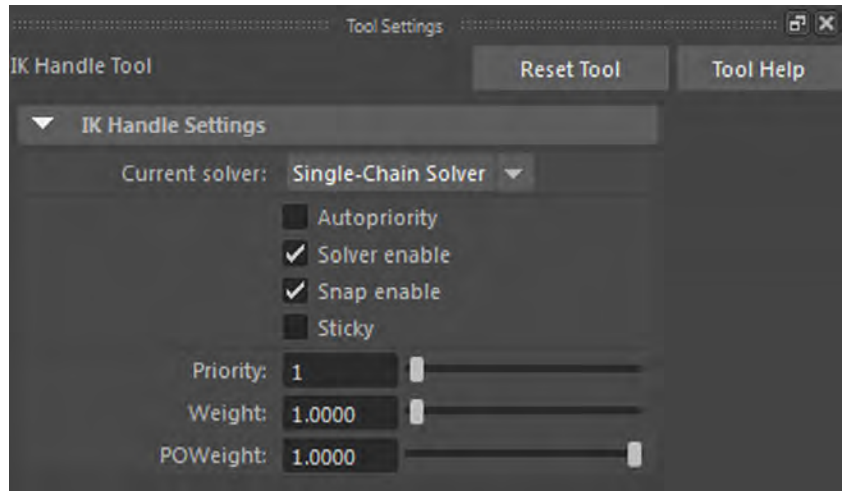
Snap Enable causes the IK Handle to snap back to the position of the end joint when you release it. You'll create a custom handle using a curve, so this option should be off.

6. Turn on Sticky.

Sticky keeps the IK Handle's position constant as you pose the rest of the skeleton. This option is very useful for keeping feet from moving through the floor when animating the other parts of the skeleton (see Figure 2.8).

FIGURE 2.8

The settings for the IK Handle tool

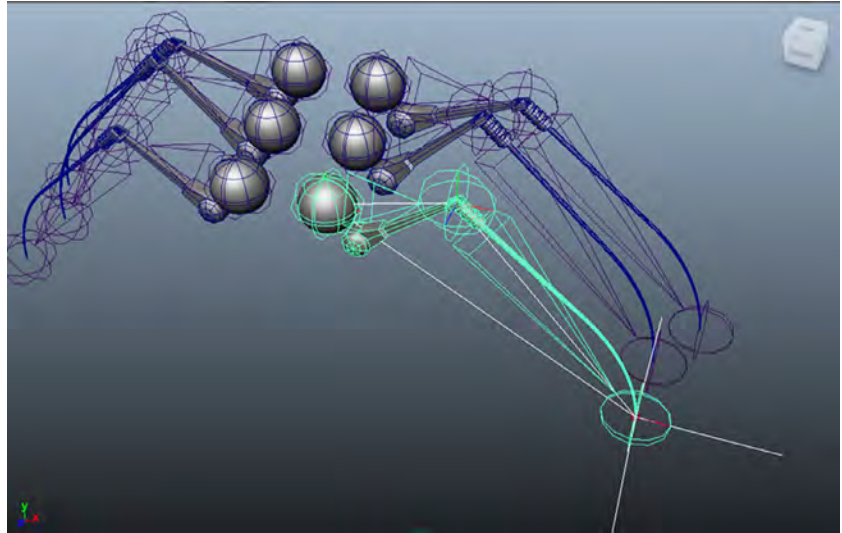


7. The other settings can be left at their defaults. With the IK Handle tool activated, click the `frontLeftLegJoint` (the joint at the root of the front left leg); then click the joint at the end of the `frontLeftLeg` chain. The IK Handle appears at the end of the joint chain (see Figure 2.9).
8. Try moving the IK Handle; you'll see the rest of the joint rotate based on the position of the handle.
9. Click Undo until the IK Handle returns to its original location.


It's usually a good idea to use a curve or another easily selectable object as a control for the IK Handle. This makes it easy to select the handle directly in the scene without having to hunt around in the Outliner.

FIGURE 2.9

The IK Handle tool creates a handle and adds the IK solver for the front left leg.



10. Create a NURBS circle (Create > NURBS Primitives > Circle).
11. In the Channel Box for the NURBS circle, expand the makeNurbCircle 1 section under INPUTS. This gives you access to the shape node settings for the circle. Set the radius to 0.4.
12. Position the circle at the same location of the IK Handle by snapping it in place.
13. Select the nurbsCircle1, and rename it **frontLeftFootCtrl**.
14. Choose Modify > Freeze Transformations so that the current position of the curve becomes its home position.
15. Select the frontLeftFootCtrl curve, and Ctrl/Cmd/Shift+click the ikHandle1.
16. Create a point constraint so that the handle is constrained to the frontLeftFoot circle (Constrain > Point).
17. You can turn off visibility of the ikHandle1 object. To move the leg, select the frontLeftLegCtrl circle and move it around. To reset its position, set its Translate X, Y, and Z channels to 0.
18. Repeat steps 2 through 17 to create controls for the other five legs. (The options for the IK Handle tool should already be stored, so you can just activate the tool and use it to add IK to the other legs.)
19. When you have finished making the controls for the other legs, select the control circles and group them. Name the group **legsControlGroup**.
20. Select the legsControlGroup group, and choose Modify > Center Pivot. This places the pivot of the group at the center of the controls.

21. Make another NURBS circle; name the circle **legsCtrl**.
22. Position the circle at the same location of the legsControlGroup's pivot.
23. Freeze transformations on the legsCtrl.
24. Select the legsCtrl circle, and Ctrl/Cmd/Shift+click the legsControlGroup.
25. Choose Constrain > Parent > .
26. In the Parent Constraint Options dialog box, make sure that Maintain Offset is on and both Translate and Rotate are set to All.
27. Click Apply to create the constraint. The Parent constraint constrains both the translation and the rotation of one object to another.

Now you have a selectable control for moving all of the legs as well as each individual leg. This will mean less time hunting around in the Outliner.

COLOR-CODING CONTROLS

You can create different colors for the NURBS curve that you use as a control. To do so, open the Attribute Editor for the curve, and expand the Object Display rollout and then the Drawing Overrides rollout. Click Enable Overrides, and use the Color slider to choose a new color for the circles. This helps them stand out in the scene.

28. Finally, you can straighten up the Outliner by grouping the IK Handles. Name the group **feetIKHandles**.
29. Group the legs and bugBody together, and name this group **bug**.
30. Save the scene as **mechBugRig_v03.ma**.

To see a version of the scene up to this point, open the mechBugRig_v03.ma scene from the chapter2\scenes directory at the book's web page.

Creating a Master Control

To finish the rig, you can create a selectable control to animate the position and the rotation of the bugBody and then group all of the parts of the bug together so that it can be moved easily or scaled in the scene.

1. Continue with the scene from the previous section, or open the mechBugRig_v03.ma scene from the chapter2\scenes directory at the book's web page.
2. Create a new display layer named **controls**.
3. Turn off the visibility of the legs and body display layers.
4. Select the NURBS circles, and add them to the controls layer.


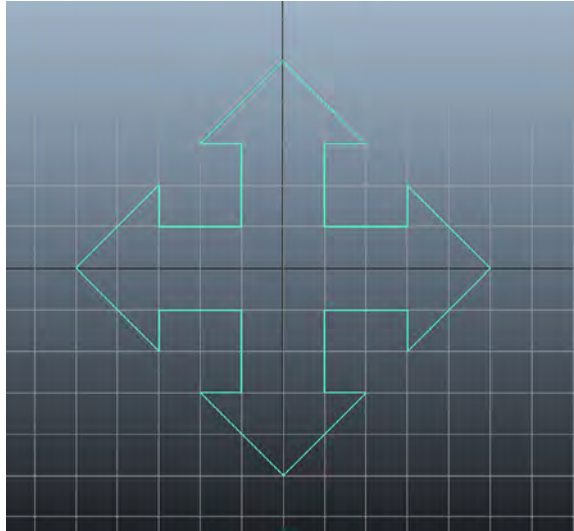
5. Turn off the visibility of the controls layer so that the scene is blank.
6. Turn on the visibility of the grid, and turn on Snap To Grids.
7. Choose Create ➤ Curve Tools ➤ CV Curve Tool ➤ . Set Curve Degree to 1 Linear.
8. Switch to the top view, and use the curve to draw a shape like the one in Figure 2.10. The shape should be a cross with an arrowhead at each end.

FIGURE 2.10

Create a cross with an arrowhead at each end using a linear CV curve.



9. Press Enter to complete the curve.
10. Choose Modify ➤ Center Pivot, and name the curve **bodyCtrl**.
11. Scale the bodyCtrl curve in X, Y, and Z to 0.35.
12. Freeze transformations on the curve.

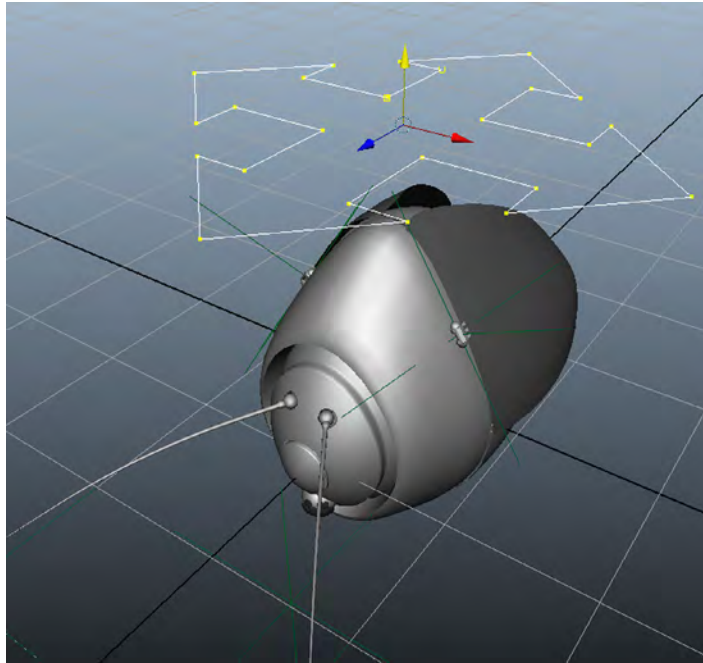
Next, you want to move the curve above the body so that you can easily select it. This curve controls the translation and rotation of the bugBody. However, you want to keep the pivot point of the control at the center of the bugBody. Since you snapped the curve to the center of the bugBody, the pivot point of the curve is at the same position as the bugBody. So, how do you move the curve without moving the pivot point? Simple—you move all of the CVs of the curve above the bugBody. This method moves the curve without changing its pivot point.

13. Select the bodyCtrl curve, and switch to CV Selection mode (right-click the curve and choose Control Vertex).
14. Drag a selection marquee over all of the CVs of the curve.

15. Turn the visibility of the body back on. Using the perspective view, switch to the Move tool and drag up on the y-axis to position the CVs of the curve above the body. Since the pivot point has not changed, it doesn't matter how high you place the curve above the bug; it just has to be easily selectable (see Figure 2.11).

FIGURE 2.11

Move the CVs of the bodyCtrl curve above the bugBody; the pivot point of the curve remains at the center of the bugBody.



16. Select the bodyCtrl curve in the Outliner, and Ctrl/Cmd+click the bug group.
17. Choose Constrain > Parent > ☐. In the Parent Constraint Options dialog box, make sure that Maintain Offset is selected and Translate and Rotate are set to All. Create the constraint.

The parent constraint constrains both the translation and rotation of one object to another object.
18. Turn on the LEGS layer.
19. Select the bodyCtrl curve, and try moving and rotating it. The legs stay on the ground (within a certain range).

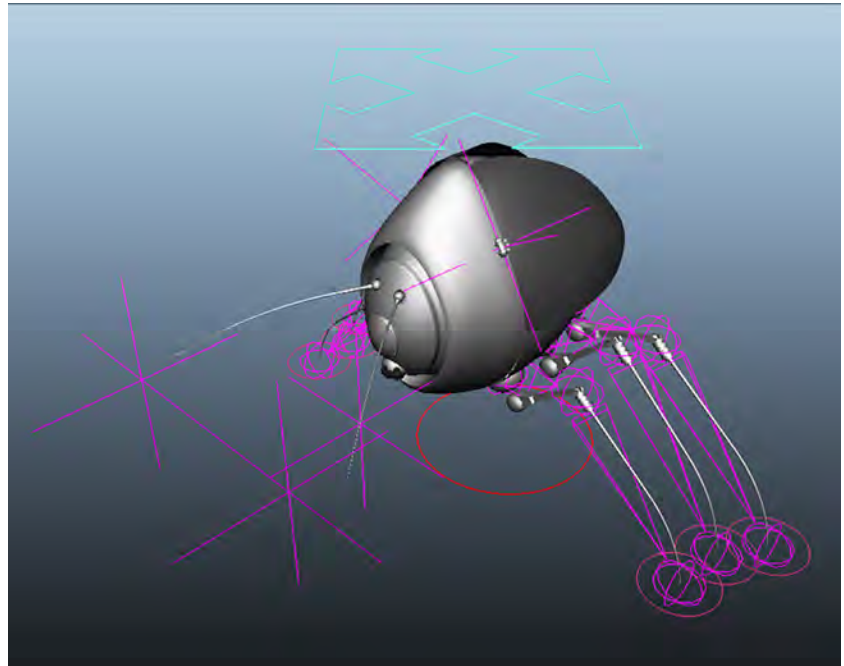
At this point, you should have a nice set of simple controls for both the body and the legs of the bug. All of this work will make animating the bug a much more pleasant experience.
20. Finally, select all of the nodes except the ground plane and cameras. Group them together. Name the group **mechanicalBug**.

21. Add the bodyCtrl curve and the three locators for the eyes and antenna controls to the controls layer.
22. Save the scene as **mechBugRig_v04.ma**.

To see a version of the scene up to this point, open the `mechBugRig_v04.ma` scene from the `chapter2\scenes` directory at the book's web page (see Figure 2.12).

FIGURE 2.12

The completed bug rig is ready for animation.



One last control that you can add to the bug is a selectable rotational control for the antennaAim group. Use the same techniques that you used to create the bodyCtrl curve. The `mechBug_v01.ma` scene has an example of this control.

Keyframe Animation

The simplest way to animate an object in Maya is to use keyframes. A *keyframe* records the value of an attribute at a point in time on the timeline. When two keyframes are set for an attribute at different points in time, Maya interpolates the value between the two keyframes on the timeline, and the result is an animation.

Keyframes can be set on almost any attribute in Maya. You can use keyframes to animate an object's position, the color of the shader applied to the object, the visibility of the object, whether the object becomes dynamic, and so on.

Now that you have a rig for the mechanical bug that can be animated, you can get to work bringing it to life.

Creating Keyframes

In this exercise, you'll see the various ways that you can set and manipulate keyframes in the Maya interface.

The mechanical bug in this scene has been rigged using the techniques discussed in the first part of the chapter. The major difference is that a circle has been added between the two antenna controls that you can select and animate to control both antennae at the same time.

The controls are color-coded so that they are more organized visually. The visibility of the joints is turned off in the Show menu of the perspective view; however, they are still present in the scene.

FRAME RATE

When you begin an animation, it's always a good idea to set the frame rate in the Maya preferences. The default frame rate is 24 frames per second, which is standard for film. To change the frame rate, choose **Windows > Settings/Preferences > Preferences**. In the Preferences window, select **Settings** from the list of Categories, and then choose **Frame Rate** from the Time drop-down list.

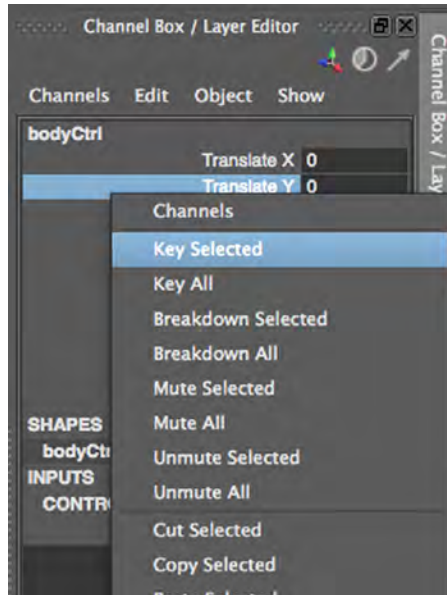
Keyframes can be placed on the individual channels of an object using the Channel Box. It's usually a good idea to keyframe on the channels that need to be animated rather than on all of the channels and attributes of an object. To keyframe individual channels, follow these steps:

1. Open the `mechBug_v01.ma` scene in the `chapter2\scenes` directory at the book's web page.
2. In the perspective view, select the blue `bodyCtrl` curve above the bug (the curve that looks like a cross with an arrowhead on each end).
3. Move the current frame in the timeline to frame 20 by clicking and dragging the timeline marker until the marker is at frame 20. Alternatively, type **20** in the box to the far right of the timeline.
4. With the `bodyCtrl` selected, highlight the **Translate Y** channel in the Channel Box by clicking it; then right-click and choose **Key Selected** from the context menu (see Figure 2.13).
5. Set the timeline to frame 48.
6. Use the Move tool to pull the `bodyCtrl` curve up to about two units, and set another keyframe on the **Translate Y** channel. The keys are represented on the timeline by a thin red vertical line (see Figure 2.14).
7. Rewind and play the animation. You'll see the bug start moving upward on frame 20 and stop on frame 48.

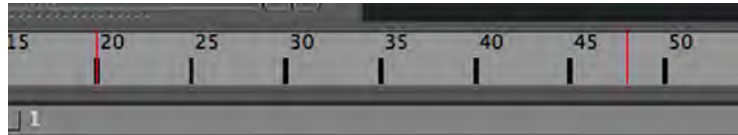
For the most part, setting keyframes is pretty straightforward. If you want to set a keyframe on all of the **Translate** channels at the same time, the hot key is **Shift+w**. To set a keyframe on all **Rotate** channels at once, use **Shift+e**. To set a keyframe on all of the **Scale** channels at once, use **Shift+r**.

FIGURE 2.13

Set a keyframe on the Translate Y channel of the bodyCtrl.

**FIGURE 2.14**

Keyframes are represented on the timeline by a thin red vertical line.



PLAYBACK SPEED

You can change the playback speed of the animation in Maya in the timeline Preferences window. (You can also access these options quickly by right-clicking the timeline and choosing a playback-speed option from the context menu.) Open the options by choosing **Windows > Settings/Preferences > Preferences**. Choose **Time Slider** from the list of categories. The **Playback Speed** menu allows you to choose the playback speed.

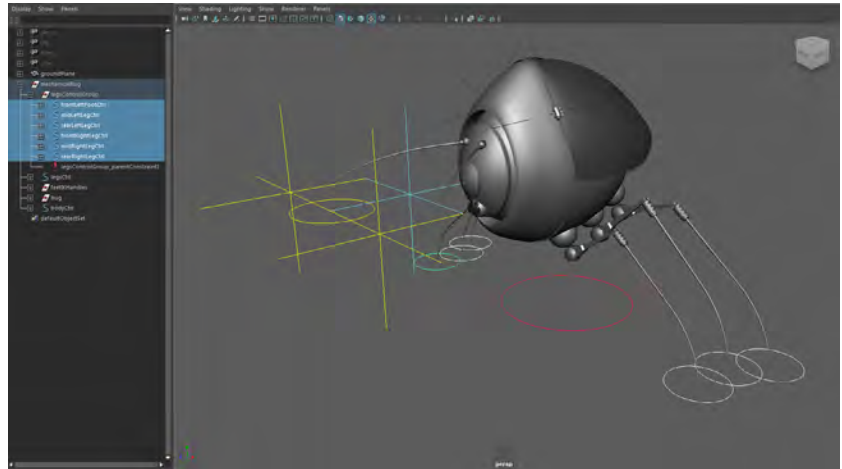
You can choose **Play Every Frame**, **Real-Time** (24 frames per second), **Half** (12 frames per second), or **Twice** (48 frames per second), or you can set a custom playback speed. Generally, the most useful speeds are **Play Every Frame** and **Real-Time**. If there are dynamics in the scene (such as particles, nCloth, rigid bodies, and so on), set this option to **Play Every Frame** so that the dynamics calculate correctly. You can set the **Max Playback Speed** as well so that the animation speed cannot exceed a certain rate. Setting **Max Playback Speed** to **Free** means that the animation will play back as fast as your processor allows, which can be faster than the final rendered animation.

Auto Keyframe

The Auto Keyframe feature automatically places keyframes on an object when a change is made to one of its attributes. For Auto Keyframe to work, the attribute must have an existing keyframe. To turn Auto Keyframe on, click the circular arrows icon to the right of the timeline. This exercise shows you how to use Auto Keyframe:

1. On the timeline for the `mechBug_v01.ma` scene, set the current frame to 20.
An easy way to do this is to use the Step Back One Frame or Step Forward One Frame button to the right of the timeline. The hot keys for moving back and forth one key are **Alt/Option+,** (comma) and **Alt/Option+.** (period).
2. Shift+click all six of the small purple leg-control circles below the bug's feet (see Figure 2.15).

FIGURE 2.15
Select the leg-control circles.

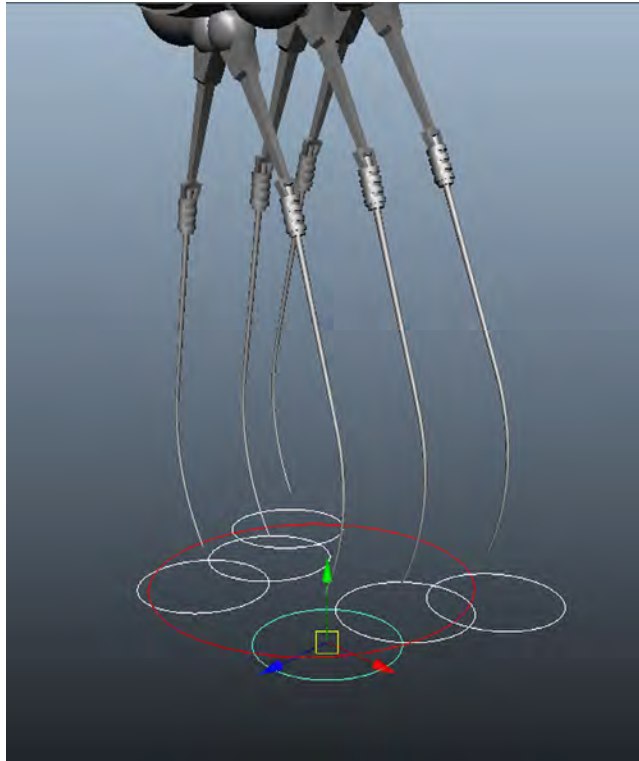


3. With the control circles selected, press **Shift+w** to place a keyframe on the Translate channels for all of the selected curves.
4. Click the icon with circular arrows to the far right of the timeline to turn on Auto Keyframe.
5. Set the timeline to frame 40.
6. Select each of the circles, and move them in toward the center (see Figure 2.16).
7. When you play the animation, you'll see that the legs move inward without you having to set a keyframe in the Channel Box.
8. Save the scene as **mechBug_v02.ma**.

To see a version of the scene up to this point, open the `mechBug_v02.ma` scene from the `chapter2\scenes` directory at the book's web page.

FIGURE 2.16

Move the leg-control circles in toward the center of the bug.



HANDLE WITH CARE

Using Auto Keyframe is a matter of personal preference. You can easily set unwanted keys on an object by mistake when using this feature, so remember to use it with caution.

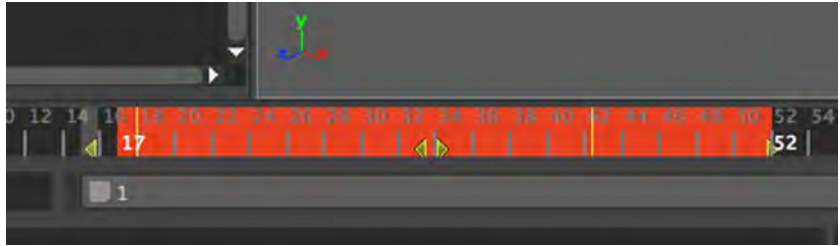
Moving and Scaling Keyframes on the Timeline

You can reposition keys on the timeline interactively by selecting the key markers and sliding them back and forth:

1. Continue with the scene from the previous section, or open the mechBug_v02.ma scene from the chapter2\scenes directory at the book's web page.
2. In the Perspective window, select the blue bodyCtrl curve.
3. Hold down the Shift key, and drag a selection directly on the timeline. You'll see a red area appear as you drag. Keep dragging this area so that it covers both keyframes set on the bodyCtrl curve (see Figure 2.17).

FIGURE 2.17

Hold down the Shift key, and drag the timeline to select a range of keys.



4. To move the keys forward or backward in time, drag the two arrows at the center of the selection. Drag these arrows to move the keys so that the first key is on frame 10.

To scale the keys, drag one of the arrows on either side of the selection. The other end of the selection acts as the pivot for the scaling operation; you may need to reposition the keys on the timeline again after scaling.

5. Scale the keys down by dragging the arrow on the right end of the selection toward the left.
6. After scaling the keys, drag the arrows at the center to reposition the keys so that the animation starts on frame 10.
7. Save the scene as **mechBug_v03.ma**.

SNAP KEYS

As you scale and move keys, you'll notice that a keyframe can be positioned on the fraction of a key. In other words, a key might end up on frame 26.68. You can fix this by choosing **Edit > Keys > Snap Keys > ☐**. In the Snap Keys Options dialog box, set the value for Snap Times To A Multiple Of to **1** and set Snap to Times. You can apply this to selected objects—or all objects, selected channels, or all channels—and define the time range. You can also right-click selected keyframes on the timeline and choose Snap. This automatically snaps keyframes to the nearest integer value.

You can move an individual key by Shift+clicking the key marker on the timeline and then dragging the arrows to move the key.

Repositioning and scaling keys directly on the timeline is usually good for simple changes. To make more sophisticated edits to the animation, you can use the Graph Editor discussed later in the chapter in the section “Additional Editing Tools.”

When you play the animation, you'll see that the bug jumps up fairly quickly and much sooner than before. You'll also notice that the animation of the legs has not changed.

Spend a few minutes practicing these techniques on each of the legs objects. Changes made to the position of the keyframes affect selected objects. You can edit each legsCtrl circle separately or all of them at the same time.

To see a version of the scene up to this point, open the **mechBug_v03.ma** scene from the **chapter2\scenes** directory at the book's web page.

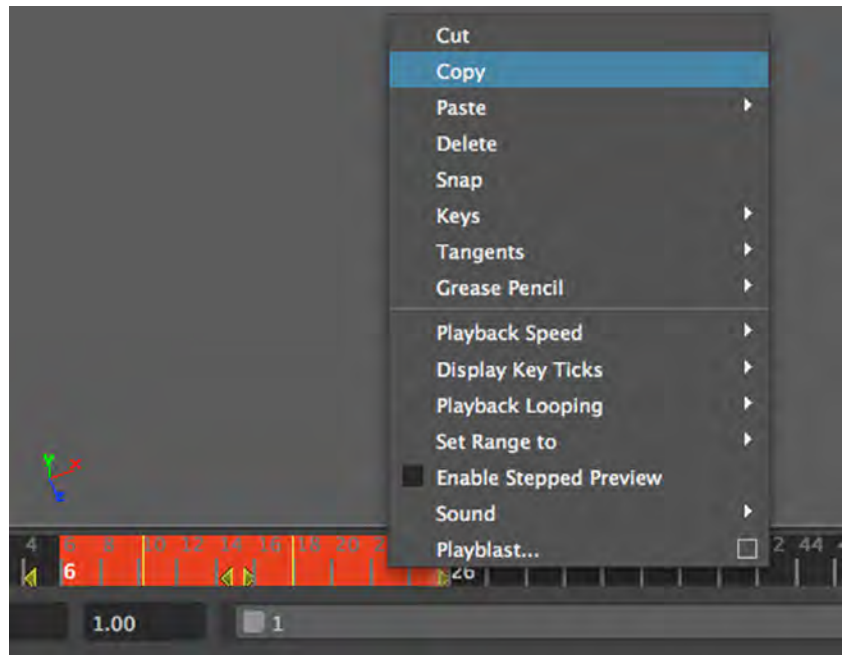
Copy, Paste, and Cut Keyframes

There are a number of ways that you can quickly copy, paste, and cut keyframes on the timeline and in the Channel Box:

1. Continue with the scene from the previous section, or open the `mechBug_v03.ma` scene from the `chapter2\scenes` directory at the book's web page.
2. Select the `bodyCtrl` curve in the perspective view.
3. On the timeline, Shift-drag a selection over both of the keyframes.
4. Right-click the selection, and choose Copy (see Figure 2.18).

FIGURE 2.18

You can copy keyframes directly onto the timeline.



5. Deselect the keys by clicking the timeline, move to frame 70, right-click the timeline, and choose Paste > Paste.

The keys for the Translate Y channel are pasted on the timeline at frame 70. If you play the animation, you'll see that the bug jumps up, moves back down, and then jumps up again.

Other options include the following:

Paste Connect Pastes the copied key with an offset starting at the value of the previous key

Cut Removes the keyframes from the timeline but copies their values to the clipboard so that they can be pasted anywhere on the timeline

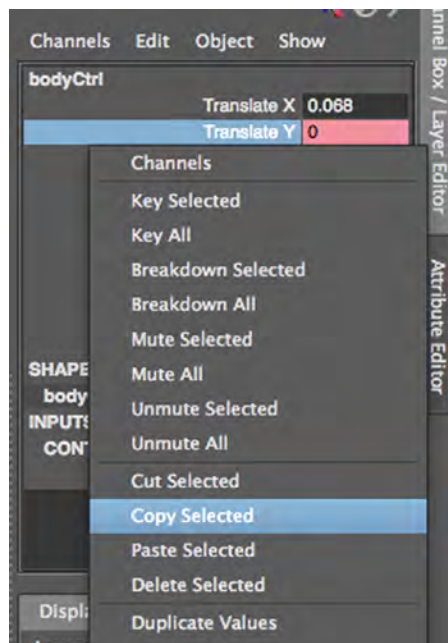
Delete Removes the keys

You can copy and paste keys from two different channels by using the options in the Channel Box.

6. Select the bodyCtrl curve in the perspective view.
7. In the Channel Box, right-click the Translate Y channel.
8. Choose Copy Selected (see Figure 2.19).

FIGURE 2.19

You can copy keyframes using the menu options in the Channel Box.



9. Highlight the Translate Z channel, right-click, and choose Paste Selected.

This pastes the values (including keyframes) copied from the Translate Y channel to the Translate Z channel. The starting point of the pasted keyframes is based on the current time in the timeline. When you play the animation, the bug moves forward as it moves upward. The forward motion will be offset (in time) depending on the current frame in the timeline.

You can also cut and delete keyframes in the Channel Box. The Duplicate keyframe operation allows you to copy keyframes from one object to another:

1. Create a NURBS sphere, and set its Translate X to 5.
2. Shift+click the bodyCtrl curve.

3. Right-click the Translate Y channel, and choose Duplicate Values.

This connects the keyframes on the Translate Y channel of the bodyCtrl curve to the NURBS sphere. If you change the values of these keys, both the bodyCtrl curve and the sphere will reflect the changes made to the keys.

When duplicating keyframes, the order in which you select the objects is important. Select the object that you want to use as the source of the duplicate keyframes last. If you graph the objects on the Hypergraph, you'll see that the same keyframe nodes are attached to both objects.

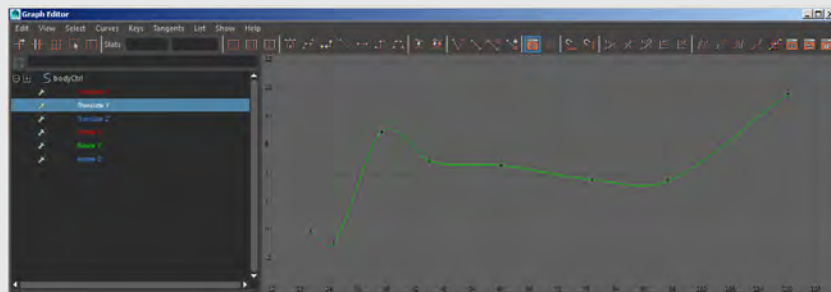
The Graph Editor

The timeline and the Channel Box offer a few basic controls for creating and editing keyframes. However, when you're refining an animation, the controls in the Graph Editor offer greater flexibility as well as visual feedback on the interpolation between keyframes. To open the Graph Editor, choose Windows > Animation Editors > Graph Editor. You can also set a viewport to the Graph Editor view.

The Graph Editor can be synchronized to the Channel Box. When they're in sync, selecting an attribute from the Channel Box also selects it in the Graph Editor. You can also synchronize the timeline with the Channel Box, limiting the displayed keyframes to your selection. Turning both of these options on allows you to manipulate keys easily using all three tools. These features can be toggled through the Channel Box under the Channels menu.

THE GRAPH EDITOR AXES

The Graph Editor is a two-dimensional display of the animation in a scene (shown here). The Graph Editor has two axes: x and y. The x-, or horizontal, axis typically displays the time of the scene in frames. As you move from left to right on the graph, time moves forward. Moving from right to left means going backward. It is possible to go into negative keyframe values if you create keys to the left of the zero marker.



The y-, or vertical, axis displays the values of the keys in units for translation and scale and in degrees (or radians, depending on the setting in the rotation settings in the preferences) for rotation. The higher you go on the y-axis, the higher the value of the key. Of course, anything below the zero line indicates negative values.

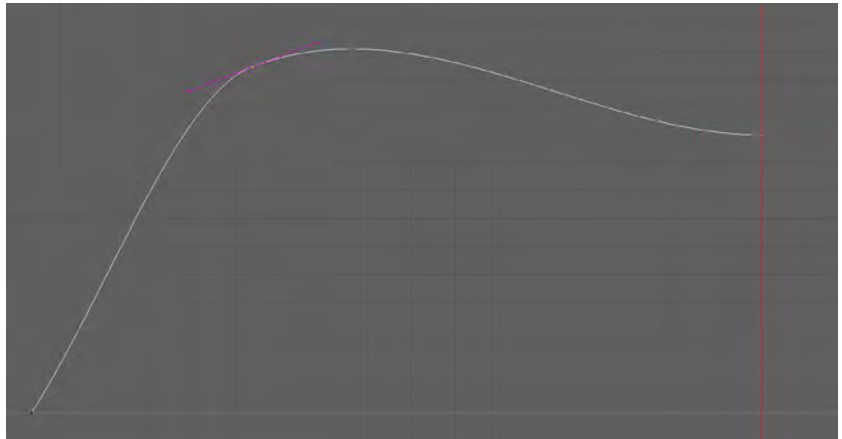


Animation Curves

Animation curves visually describe how the values between two keyframes are interpolated over time. A keyframe is represented by a point on the curve. The portion of the curve on the graph to the left of a keyframe represents the animation before the key, and the line and the portion on the right represent the animation after the key. The keys themselves have handles that can be used to fine-tune the shape of the curve and thus the behavior of the animation both before and after the key. The shape of the curve to the left of the key is known as the *incoming tangent* (or in tangent), and the shape of the curve to the right of the key is known as the *outgoing tangent* (or out tangent). This is shown in Figure 2.20.

FIGURE 2.20

The curves on the Graph Editor represent the interpolation of values between keyframes for a selected object.



Each animated channel has its own curve on the graph. You can use the menu in the Graph Editor to edit the keys and change the way that the curves are displayed on the Graph Editor.

In this exercise, you'll use the Graph Editor to refine a simple animation for the mechanical bug. In this animation, the bug will leap in the air, hover for two seconds, lean forward, and then fly off the screen in a gentle arc. You'll start by setting keys on the Translate channels of the bug. To make things easier, you'll use Auto Keyframe.

The setting applied to the tangents of the curves specifies the overall shape of the curve before and after each key. You can apply a tangent setting to one or more selected keys on the graph, and the in and out tangents can also have their own settings.

The settings are listed in the Tangents menu in the Graph Editor and are also represented visually by the icons at the top of the Graph Editor. Clicking one of these icons or applying a setting from the Tangents menu changes the interpolation of the selected key(s) or tangent handles (see Figure 2.21).

FIGURE 2.21

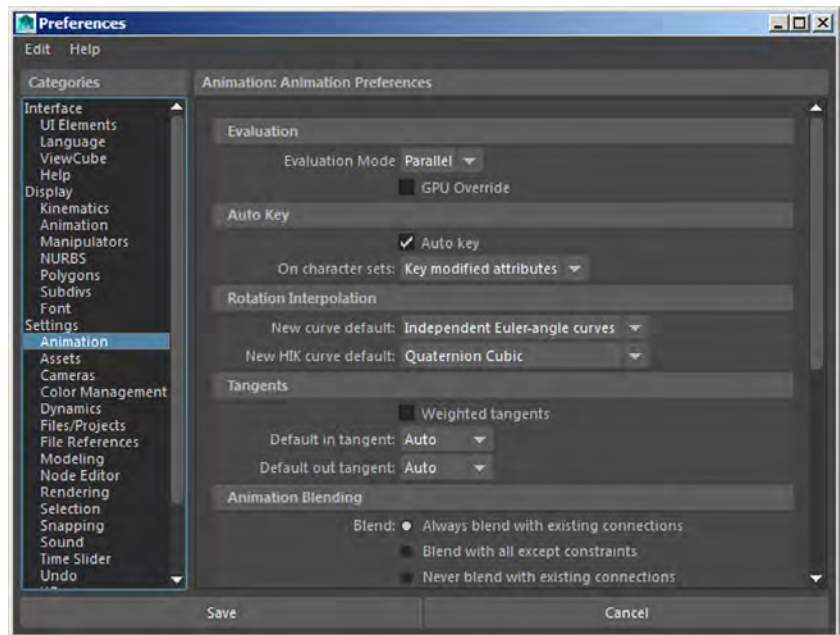
The icons for the tangent settings at the top of the Graph Editor



When you are blocking out an animation, you'll often find the Stepped or Step Tangents setting useful. Stepping the tangent eliminates the interpolation between keys so that an object's animated values do not change between keyframes; instead, the animation appears to pop instantly from key to key. Using stepped keys when you block out the animation gives you a clear idea of how the object will move without the additional motion that can be added by curved tangents.

1. Open the mechBug_v01.ma scene from the chapter2\scenes directory at the book's web page. This is the original rigged bug without the keyframes that you added in the previous sections.
2. Open the Preferences window by choosing Windows > Settings/Preferences > Preferences.
3. Click the Animation category under Settings.
4. Make sure that Auto Key is enabled. Click Save to save the preferences (see Figure 2.22).

FIGURE 2.22
The animation preferences are established in the Preferences window.



5. Set the length of the timeline to 120.
6. Set the current frame to 20.
7. RMB-click the timeline, and check Enable Stepped Preview.
8. Select the blue bodyCtrl curve in the perspective view.
9. Press **Shift+w** and **Shift+e** to set keyframes on the Translate and Rotate channels, respectively.

When you are blocking out the animation, the only object that needs to be keyframed at this point is the bodyCtrl curve.

- 10.** Set the timeline to frame **25**, and move the bodyCtrl curve down so that the Translate Y channel is about **-0.78**.
- 11.** Set the timeline to frame **35**.
- 12.** Drag up on the bodyCtrl curve until the Translate Y channel is around **7** units.
- 13.** Set the timeline to frame **45**.
- 14.** Rotate the bodyCtrl slightly, and move the bug a little forward and to the side. This gives the bug a slight wobble as it hovers. Try these settings:

Translate X: **-0.68**

Translate Y: **4.968**

Translate Z: **0.532**

Rotate X: **-11**

Rotate Y: **7**

Rotate Z: **10**

Many of these values were arrived at simply by moving and rotating the bodyCtrl curve in the scene. You can use these exact values or something that's fairly close. Most of the time when blocking in the animation, you'll move the objects in the scene rather than type in precise values, but the keyframe values are included here as a rough guide. Remember, right now the only object being keyframed is the bodyCtrl curve.

- 15.** Set the timeline to frame **60**. The bug is starting to turn as it decides which way to fly. Rotate it to the left a little and add a bit more variation to its position. Try these settings:

Translate X: **-0.057**

Translate Y: **4.677**

Translate Z: **-1.283**

Rotate X: **-18**

Rotate Y: **20**

Rotate Z: **-13**

- 16.** Move the time slider to frame **79**. Now the bug is starting to fly away. Rotate it so it is facing downward slightly. Try these settings:

Translate X: **1.463**

Translate Y: **3.664**

Translate Z: **-0.064**

Rotate X: **31**

Rotate Y: 35

Rotate Z: 1.5

17. Set the time slider to frame 95. The bug is beginning its flight, so it turns more to the left and dips down a little. Try these settings:

Translate X: 4.421

Translate Y: 3.581

Translate Z: 1.19

Rotate X: 1.5

Rotate Y: 46

Rotate Z: 2

18. In the final keyframe on frame 120, the bug is flying away. Try these settings:

Translate X: 11.923

Translate Y: 9.653

Translate Z: 6.794

Rotate X: 49

Rotate Y: 62

Rotate Z: 24

19. Play back the animation a few times. You'll see the bug pop from one position to another. Make changes if you like, but try not to add any more keys just yet. It's best to use as few keys as possible; you'll let the curves do all of the work in a moment.
20. Select the red legsCtrl circle, and keyframe its Translate channels so that it follows the flight of the bug. To keep it interesting, place the keys at different frames than the bodyCtrl curve. Remember to set an initial keyframe before using Auto Keyframe.

You can set keys on the individual foot controls, but at this point let's keep things simple and focus on just the bodyCtrl curve and the translation of the legsCtrl curve.
21. From the perspective view, choose Panels > Saved Layouts > Persp/Graph/Outliner so that the interfaces are split between the perspective in the Outliner and the Graph Editor.
22. Select the bodyCtrl curve, and hold the cursor over the Graph Editor.
23. Press the **f** hot key so that you can see all of the animation curves for the bodyCtrl object. Since the timeline is set to preview stepped tangents, they look like straight lines in a stepped pattern (see Figure 2.23).
24. Save the scene as **mechBug_v04.ma**.

To see a version of the scene up to this point, open the mechBug_v04.ma scene in the chapter2\scenes folder at the book's web page.