

MEL Scripting for Maya Animators



Maya动画师 MEL脚本编程全攻略

[美] Mark R. Wilkins 著
Chris Kazmier
Stephan Osterburg

唐俊华 王东安 等译



MK[®]
MORGAN
KAUFMANN



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

MEL Scripting for Maya Animators

Maya动画师MEL脚本 编程全攻略

Mark R. Wilkins

〔美〕 Chris Kazmier 著

Stephan Osterburg

唐俊华 王东安 等译

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

MEL脚本语言是Maya编程的核心,也是利用Maya在三维世界里自由创作所必需的工具语言。本书将MEL编程和整个动画制作过程相结合,并通过详细的实例来展示具体问题的解决方法。通过应用于粒子动力学、刚体动力学、群集系统和人物造型的大型实例,将本书推向Maya应用的颠峰,为读者最终成为熟练的程序员奠定了实践基础。

本书适用于已经掌握了Maya用户界面操作的读者,同时也可以作为Maya高级用户的参考书籍。



Copyright©2003 by Elsevier Science (USA). Translation Copyright©
2003 by Publishing House of Electronics Industry. All rights reserved.
本书英文版由美国Elsevier Science公司出版,Elsevier Science公司已
将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有
限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号 图字: 01-2004-0336

图书在版编目(CIP)数据

Maya动画师MEL脚本编程全攻略/(美)威尔金斯(Wilkins, M. R.)等著;唐俊华等译.一北京:电子工业出版社,2004.7

书名原文: MEL Scripting for Maya Animators

ISBN 7-120-00117-5

I. M… II. ①威… ②唐… III. 三维-动画-图形软件, Maya MEL-程序设计 IV. TP391.41

中国版本图书馆CIP数据核字(2004)第055023号

责任编辑:徐云鹏

特约编辑:卢国俊

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:25.5 字数:650千字

印 次:2004年7月第1次印刷

定 价:40.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

关于作者

Mark R. Wilkins是PDI/DreamWorks公司的高级技术导演，在那里他帮助开发了一个使用Maya制作效果和人物动画的产品流水线。**Mark**也为Maya动画师提供培训和技术支持。他也在Walt Disney动画工厂担任过许多不同的职位，包括软件工程师和场景设置主管。他也为很多电影做出过贡献，包括Dinosaur（恐龙）、The Legend of Bagger Vance（重返荣耀）、Mission Impossible 2（碟中谍2）、A.I.: Artificial Intelligence（人工智能）和Minority Report（少数派报告）。**Mark**拥有Harvey Mudd大学的物理学学位。

Chris Kazmier是DreamWorks Feature Animation（梦工厂动画电影）公司的效果动画师和技术导演，在那里他从事传统动画电影以及完全由计算机图形学所生成动画电影的效果制作。早先，**Chris**在PDI/DreamWorks公司从事Intel Aliens广告宣传方面的活动，并且作为三维动画导演也在Fox动画工作室从事过影片Titan AE（冰冻星球）、Bartok the Magnificent（巴托克的辉煌）和Anastasia（真假公主）方面的工作。**Chris**也教授过各种计算机图形学课程，包括针对中级和高级学员的三维动画课程。

前 言

在1997~2002年间, Alias|Wavefront公司的软件产品Maya被动画制作、视觉效果以及游戏制作界广泛接受,在很大程度上是因为它鼓励通过开发插件以及编写Maya嵌入式语言(MEL)脚本进行大范围的定制。包括Disney, Dreamworks, ILM, 以及Sony Picture Imageworks在内的一些主要制片厂, 难于找到足够的、有使用MEL开发有用工具经验的熟练技术导演。

了解到这一事实, 在2001年中, 我们惊讶地发现有关MEL脚本编程方面可利用的资料相当匮乏。在各种各样Maya课程及书中, 对于MEL的讨论更趋向于关注其作为一种编程语言的结构而很少讨论它是如何与整个动画相结合的。

由上述讨论诞生了《Maya动画师MEL脚本编程全攻略》这本书。本书是在实际环境中如何使用MEL和Maya表达式语言的教程指南和例库; 教程部分介绍在Maya中MEL是怎样与整个动画制作过程相结合的, 而例子部分的章给出对具体问题的解决方法, 从这些例子可以推演出自己的解决方法。

本书定位于对Maya界面的使用具备中等程度知识的用户。如果以前根本没有接触过Maya, 你可能能够完成那些例子, 但如果从类似Alias|Wavefront《学习Maya》这样的好书, 或者不断由第三方推出的针对初学者的书中的一本开始学习, 将发挥本书最大的作用。

MEL的应用领域非常广泛, 所以各种不同的读者对不同章节感兴趣的程度不同。如果你不是一个熟练的程序员, 可能希望阅读第1章至第3章, 以对表达式和简单MEL脚本背后的概念有个基本的了解; 继续学习第6章、第7章和第9章以便了解如何管理数据, 如何使用MEL命令工作, 以及如何自动化一些重复性的工作; 然后着手解决有更复杂例子的第4章和第5章。不要过多关注于对这些章中数学的理解, 而更多关注于对脚本是如何构成的以及脚本所做工作的理解。接着, 可以转到第12章以领会如何向用户提问并呈现信息。到那时, 就可以跳到一些例子部分阅读。

如果你是一个对人物动画感兴趣的熟练程序员, 克制住完全略过第4章和第5章的冲动, 因为这些章节讨论了一些有用的主题, 譬如向量和数组是怎样彼此相联系的, 以及与表达式相反, 能够将MEL脚本用于什么目的。一般地, 第6章到第11章中对MEL语言的讨论也可以用于人物造型。

如果你是一个对动画效果感兴趣的熟练程序员, 通读本书会非常惬意。仔细研究群集系统的例子, 因为在我们看来使用Maya的动力学求解器而不是表达式系统实现对象扭动着跑出场景的策略极少被论述。

推荐访问Mel Scripting for Maya Animators网站, 网址是www.melscripting.com。通过点击“Supplementary Material”链接, 能找到很多例子中的脚本文件及场景文件, 以及已勘误表的一个列表。

目 录

第1章	Maya内部运行机制	1
	本章将学习以下内容	1
	为什么要深入学习Maya	1
	依赖图、属性、连接	2
	变换层次结构和父/子关系	9
	查看层次结构	11
	变换节点和形状节点	12
	MEL以及Maya用户界面	15
	Maya内部运行机制要点	15
第2章	MEL命令基础	17
	本章将学习以下内容	17
	不编写脚本能使用MEL么	17
	命令行和命令反馈行	18
	命令解释器	18
	脚本编辑器	19
	脚本编辑器与命令解释器的对比	20
	将脚本编辑器中的消息作为MEL代码	20
	为MEL脚本制作搁板按钮	22
	保存MEL脚本	22
	状态消息区域的危险诱惑	23
	whatIs命令	24
	MEL命令的基本结构	24
	在互联网上得到有关Maya和MEL命令的资料	25
	不编写脚本而使用MEL的要点	26
第3章	使用表达式	27
	本章将学习以下内容	27
	什么是表达式	27
	表达式如何工作	28
	等号：相等和赋值	28
	Maya怎么实现表达式	29
	Maya表达式语言与MEL相同吗	30
	什么时候能（什么时候不能）使用表达式	31
	在属性之间定义关系	31
	什么是操作符优先	33

	Maya表达式语言的预览 (Walkthrough)	33
	关于使用表达式应该记住的	45
第4章	利用表达式控制粒子	47
	本章将学习以下内容	47
	粒子对象的两种属性: 每对象属性和每粒子属性	47
	向量	48
	两种表达式: 常规表达式和粒子表达式	52
	有效使用粒子表达式的几点提示	62
	Maya粒子表达式要点	74
第5章	利用MEL脚本进行问题求解	75
	本章将学习以下内容	75
	MEL在Maya中的作用: 构建场景	75
	设计MEL应用的策略	76
	最简单的用户界面	77
	在MEL中创建、编辑并查询节点	78
	在MEL中添加、设置和获取属性的值	79
	在MEL中连接属性	80
	在MEL中创建并连接表达式节点	81
	编写MEL脚本的要点	88
第6章	变量和数据类型	91
	本章将学习以下内容	91
	声明变量 (及取消对变量的声明)	91
	环境变量	96
	MEL命令和类型检查	96
	简单数据类型和聚合数据类型	97
	MEL中变量和数据类型的要点	106
第7章	使用MEL命令	109
	本章将学习以下内容	109
	什么是MEL命令	109
	MEL命令的结构	110
	在MEL脚本中使用MEL命令	111
	避免在表达式中使用MEL命令	113
	使用MEL命令的要点	113
第8章	在MEL中操纵节点	115
	本章将学习以下内容	115
	使用ls命令以通过名称得到节点或其他属性	115
	使用select命令管理选中对象列表	118
	在Maya场景中创建节点	119
	查找节点的父节点和子节点	120

	找出节点连接的相关信息	121
	关于Maya节点类型，DG节点和属性参考	122
	MEL中节点管理的要点	122
第9章	控制执行流程	125
	本章将学习以下内容	125
	控制脚本执行的流程	125
	基本条件运算：if-else和switch	126
	循环	135
	在MEL中控制执行流程的要点	138
第10章	过程和函数	139
	本章将学习以下内容	139
	自顶向下的设计方法	139
	过程和函数是什么	141
	MEL中过程、函数以及自顶向下设计方法的要点	150
第11章	给节点、脚本和变量命名	153
	本章将学习以下内容：	153
	为什么命名规范很重要	153
	脚本命名	154
	给变量命名	155
	给节点命名	156
	什么是名字空间	162
	使用名字空间的方法	164
	关于给脚本、变量和节点命名应该记住的	164
第12章	设计MEL用户界面	167
	本章将学习以下内容	167
	什么是用户界面	167
	Maya用户希望从MEL脚本中看到什么	168
	设计用户界面之前需要回答的问题	169
	设计和测试用户界面	170
	对话框的结构	171
	关于在MEL中设计用户界面时应该记住的	172
第13章	简单的MEL用户界面	173
	本章将学习以下内容	173
	从用户收集信息	173
	什么时候要验证用户输入，为什么	174
	使用confirmDialog对话框询问确认信息	175
	使用promptDialog让用户输入文本串	176
	使用fileDialog让用户选择文件或目录	177
	使用警告和错误命令处理警告和错误	177

	使用正则表达式和匹配来验证数据	178
	正则表达式如何工作	178
	验证整数	180
	验证浮点数	181
	验证对象的名字（不需要名字空间）	182
	关于简单的MEL用户界面和输入验证应该记住的	186
第14章	定制对话框	189
	本章将学习以下内容	189
	如何编写使用定制对话框作为输入界面的脚本	189
	对话框及其内容	190
	常见的控件类型	199
	常用布局类型	205
	在MEL中构建定制对话框的要点	211
第15章	安装MEL脚本	213
	本章将学习以下内容	213
	安装脚本使其在所有场景中能够被访问	213
	安装Maya启动时就会运行的脚本	214
	将脚本作为脚本节点安装到场景	214
	安装定制菜单	215
	管理按钮搁板并创建定制搁板图标	216
	安装MEL脚本的要点	216
第16章	MEL应用于粒子动力学的例子	219
第17章	MEL应用于刚体动力学的例子	255
第18章	一个简单群系统的例子	303
	可供参考的完整脚本: crowdSystem.mel	358
第19章	在人物装配中使用MEL的例子	369
	完整的mrBlahControls.mel文件文本	391
	明确问题	395
	实现主脚本	395
附录A	设计MEL脚本的步骤清单	395
后记		397

1

Maya内部运行机制

本章将学习以下内容

- 对于Maya来说，所定义的场景就是节点以及它们之间连接的集合，这些节点及它们之间的连接构成了依赖图。
- 什么是依赖图节点以及它与三维浏览器中所看到的对象有什么关系。
- 通道框和属性编辑器窗口是Maya界面中操纵依赖图节点属性最常用的工具。
- 某些依赖图节点，当然并非所有依赖图节点，是变换层次结构中的一部分，这个层次结构建立了对象间的空间关系。
- 怎样使用大纲窗口、超图窗口以及连接编辑器来检查节点在依赖图及变换层次结构中的位置。
- 一些如创建对象、设置动画关键帧，对NURBS表面建模等常见工作是怎样改变依赖图的。
- Maya的图形用户界面包括：菜单、工具条、时间线等，是由MEL脚本构建的，并能使用MEL脚本来改变。

为什么要深入学习Maya

一些Maya动画师发现MEL脚本语言难以学习，原因之一是学习MEL脚本语言需要理解一个场景中的不同部分协同工作的众多细节问题，而这些细节是Maya用户界面所尽力隐藏的。这一章揭示了很多这方面的细节，并将它们与Maya动画师在不使用脚本语言工作时，在用户界面中所看到的对象相联系。

即使完全不使用MEL脚本语言编程，了解一个场景在Maya中是如何组合也很重要。如果是一名富有经验的Maya动画师，你可能已经了解本章所述的大部分内容。然而，当使用MEL脚本语言开发时，搞清Maya场景中的组件以及它们是如何组合的非常关键，因为不像使用用户界面，使用MEL脚本语言迫使你在这种较低的层次上频繁地与所建的Maya场景打

交道，而很少隐藏底层的复杂性。

依赖图、属性、连接

从内部结构来说，场景中的每一部分不管它是三维几何图形、动画、表达式关系、灯光、纹理，还是用于创建有历史信息对象的参数，都被表示为一个或者多个节点，更完整地称其为依赖图节点或DG节点。每一个节点都有一个属性集，每一属性保存着该节点所代表对象的一个特性。所有这些节点以及它们的连接被称为依赖图或者场景图（图1.1）。

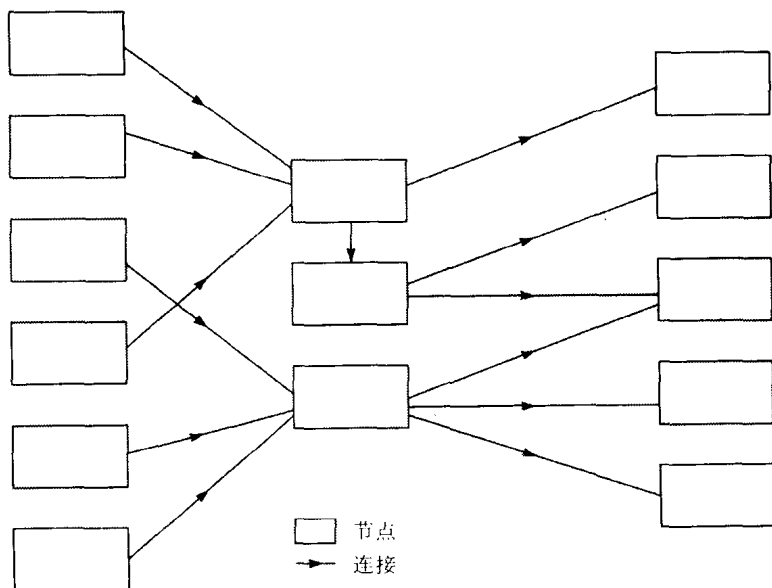


图1.1 依赖图

对于DG节点，一个有用的参考信息是Maya的DG节点与属性参考，通过帮助菜单可以获得该参考信息，该文档对大多数（如果不是所有的）Maya中的内建DG节点、节点的属性以及它们在场景中的作用进行了描述。

属性自身也具有一些特性，这些特性对能以何种方式操纵属性进行控制。属性可以被锁定（Locked）以防止它们被改变；属性能够被标示为可关键化（keyable），以允许通过定义关键帧形成动画，或者被标示为不可关键化（nonkeyable），禁止通过定义关键帧形成动画。

同样，每一个属性都具有数据类型，该数据类型说明该属性能存储哪种信息。在Maya中，属性能够存储的数据包括：

- 整数（没有小数部分）。
- 浮点数（有小数部分）。
- 字符串，字符串是文本和数字的组合。
- 布尔值，布尔值可以取“开/关”或“真/假”值对。

- 枚举类型值，它存储在创建属性时所定义选项列表中一项的值。
- 同样，一些属性可以存储上述类型数据的集合，包括数组、向量和矩阵。

MEL脚本的一个重要作用是创建并连接DG节点，正是这些节点以及它们之间的连接使场景回放时产生特殊效果。最初，一个学习节点以及连接如何生成动画的好方法是亲手制作动画，然后检查由Maya所生成的节点及连接。稍后，随着对MEL语言进一步学习，会学到如何使用脚本来创建节点网络及属性连接。查看以不使用编写脚本的方式创建动画时Maya所完成的工作可以为编写复杂脚本提供重要的思想来源。

通道框作为Maya标准布局的一部分，显示组成所选对象（图1.2）的一个或多个节点（在本例中为directionalLight1和directionalLightShape1）。通道框仅显示那些被标示为可关键化（keyable）的属性，因为它们是在工作时编辑最频繁的属性。对其他属性的编辑通常是通过属性编辑器来实现的（图1.3）。即使是属性编辑器也不显示每一个属性。某些属性只能通过MEL脚本和表达式来处理。

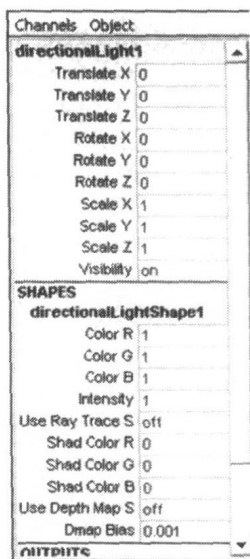


图1.2 通道框

属性编辑器以一系列组的方式来显示所选节点的属性，组可以通过点击它们左边的箭头按钮来扩展。在图1.3的例子中，Directional Light Attributes就是这样的一个组。相互连接的节点以标签的方式处于属性编辑器的顶端，以允许方便地访问其他与所编辑节点相关的节点。

属性编辑器也允许通过使用Attributes（属性）菜单中的选项添加自定义属性。本章将在后面的内容中讨论在表达式和MEL脚本中怎样发挥自定义属性的用途。

属性的最后一个重要特性是它们之间能够互相连接。在两个属性之间建立连接使得其中的一个属性的值与另一个属性的值保持相同。这种连接是有方向性的，这意味着，如果将一属性连接到另一属性，可以随意更改第一个属性，第二个属性将会跟着改变；而不能更改第二个属性的值，因为它的值是由第一个属性的连接驱动的。通过使用连接编辑器（图1.4）

可以建立节点之间的连接。

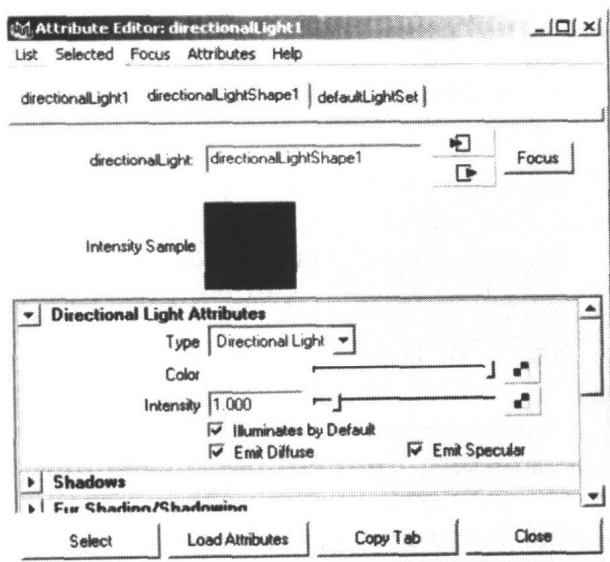


图1.3 属性编辑器

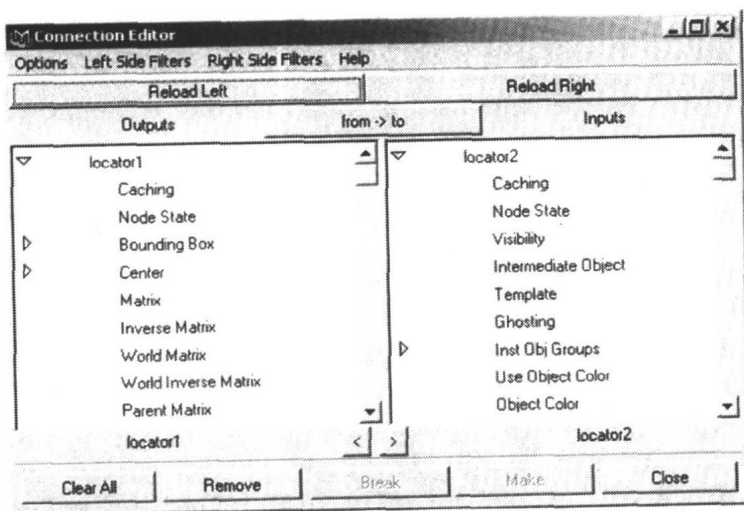


图1.4 连接编辑器

连接的这种特性是依赖图得名的由来。首先它是一个“图”，“图”是描述相互连接节点网络的术语，“依赖”指的是每一被连接的节点依赖于连接到该节点的节点值。其属性被连接到当前节点的那些节点为上游节点，那些依赖于当前节点的节点为下游节点。类似Maya依赖图的场景图的思想在计算机动画系统中是常见的，例如：在3D Studio MAX和SoftImage中都使用场景图结构。

查看多个节点之间相互连接的一个十分有用的工具是超图窗口（图1.5），通过选择主菜单中的**Window>Hypergraph...**菜单项可以打开该窗口。通过选中节点，并使用超图窗口中**Graph>Up Connections**和**Graph>Downstream Connections**菜单项，就能够查看与当前所选节点所建立的连接。

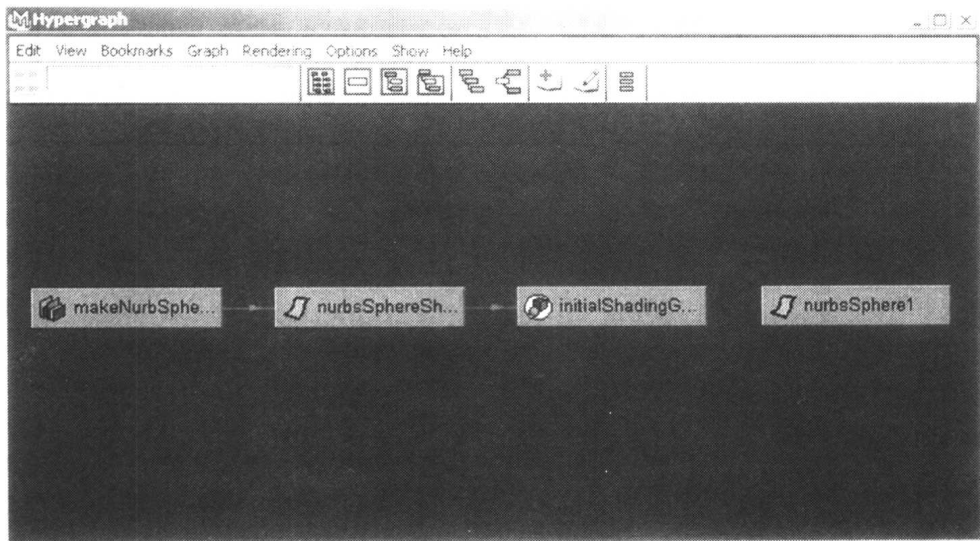


图1.5 使用超图窗口查看连接

需要注意的一点是，不可能通过**Up Connections**和**Downstream Connections**模式查看整个场景。因为即使在一个相对简单的场景中也存在太多的节点及连接，在超图窗口中仅显示连接到所选取对象的连接以使得在超图窗口中的工作易于管理。当以这种模式查看节点时，可以将鼠标指针停留在代表节点之间连接的箭头线上，以查看被连接的是哪些属性。

为了了解在Maya中一个简单的场景是如何被表示成依赖图的，检查在制作一个跳动球动画时发生的事情。为简单起见，假定该球只沿着一个坐标轴运动。

制作小球动画的第一步，可以为小球垂直方向的运动设置3个关键帧，假定为动画中的第0帧、第10帧和第20帧。

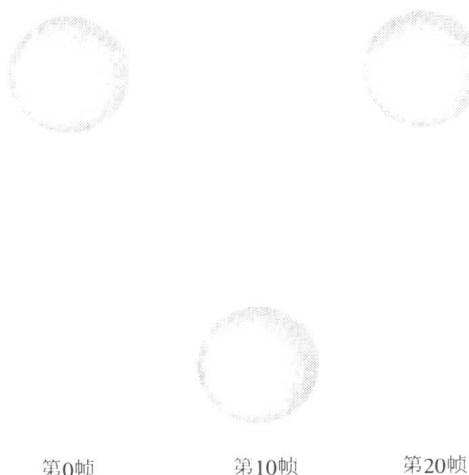


图1.6 跳动小球的关键帧

制作动画时，每一个动作都会在后台创建节点和连接，如图1.7中所看到的，创建球体就会生成3个互相连接的节点。

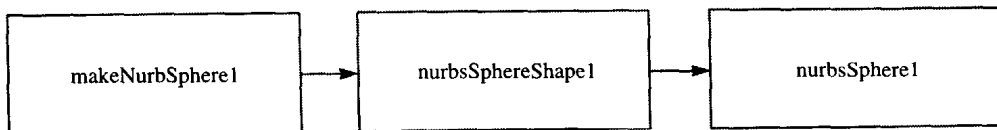


图1.7 组成新球体的节点

然后，设置动画关键帧导致另一个节点的创建和连接（图1.8），现在nurbsSphere1的translateY属性由一个动画曲线（animation curve）来驱动。当回放该场景时，Maya的time1节点，也就是场景的时钟，告诉动画曲线（animation curve）去查找第几帧的值。接着，动画曲线（animation curve）为translateY属性正确地赋值。最后，Maya在设定的位置画出该球体。

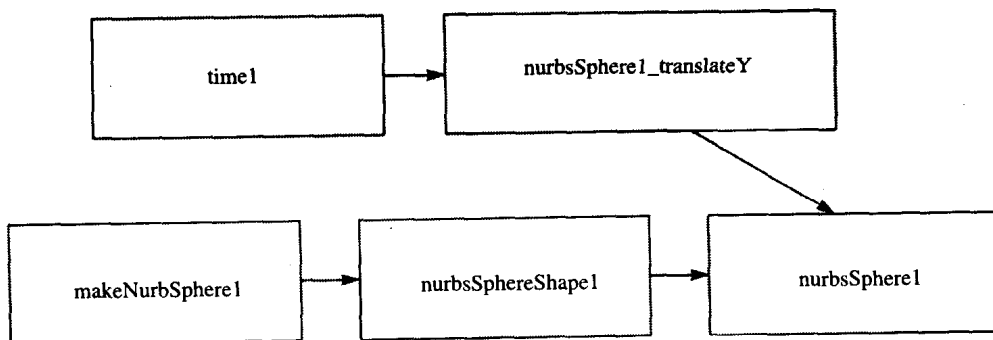


图1.8 组成小球动画的节点

正如所看到的，在Maya中一些诸如建立原型或属性动画这样的简单操作，在底层都有复杂的含义。MEL语言的作用之一是它能使为完成某项任务而建立大规模节点网络的工作更简单。

在Maya中，表达式概念比简单的属性值连接概念复杂得多，因为表达式能够运用数学运算或者任何使用MEL语言编写的方法，从一个属性计算另一属性的值。连接只是简单地将一个属性与另一属性设置为相同的值。

它的工作原理是使用表达式编辑器或者在通道框中输入的方式创建表达式将创建一个新节点，即表达式节点。这个节点中包含该表达式脚本，并能够使用输入数据计算表达式的输出。如果创建的表达式是通过使用节点node1的一个属性值计算节点node2的一个属性值，得到的结果如图1.9所示。在第3章中有对表达式更加详尽的讨论。

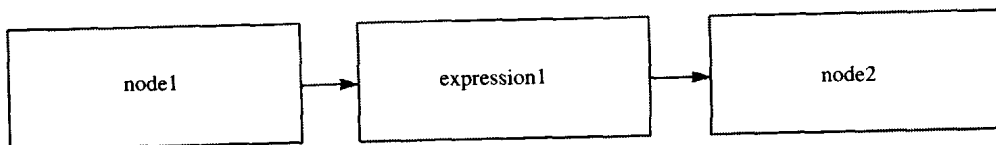


图1.9 表达式节点及其连接

例1: 使用超图窗口浏览依赖图

本例将示范如何使用超图窗口来查看一个场景中的节点是怎样连接的。首先, 看一个跳动小球动画的例子。

1. 选择主菜单中File>New Scene菜单项创建一个新场景。
2. 点击球体工具或者选择Create>NURBS Primitives>Sphere菜单项创建一个球体。
3. 在通道框Translate Y域中输入10, 以将该球体向上移动10个单位。
4. 在通道框Translate Y域中数字10的位置点击鼠标。然后, 在该域上单击鼠标右键, 并从弹出菜单中选择Key Selected项。Translate Y域将改变颜色。
5. 在时间线上点击第10帧, 当前时间标记将前进到10, 而在第0帧的位置留下一道红色标记以表明第0帧是一个关键帧。
6. 在Translate Y域中输入0, 按下回车键以提交变更, 并且使用弹出菜单的Key Selected项设置另一关键帧。
7. 在时间线上点击第20帧以前进时间, 设置一个Translate Y域中值为10的关键帧。
8. 倒回并回放该动画。
9. 确信球体被选中, 然后选择Windows>Hypergraph...菜单项。当超图窗口出现时, 通过使用Alt+MMB (鼠标中键) 的方式将所选的球体对象拖动到视图中央。注意梯形形状表示该球体已经被做成动画了。
10. 选择超图窗口Graph>Up Connections和Graph>Downstream Connections菜单项, 应该能够看到构成该球体的节点和多个与之相连接的节点, 如图1.10所示。

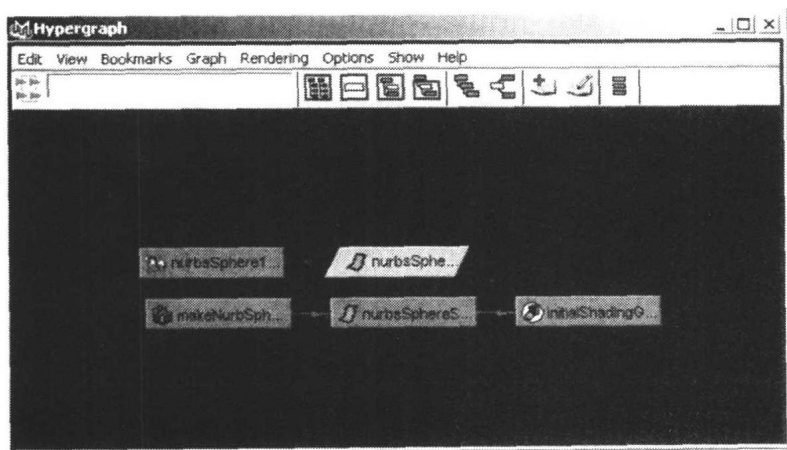


图1.10 小球动画的上行连接和下行连接

在超图窗口中, 允许使用与在透视视图和投影视图中漫游时相同的鼠标控制方法。使用Alt+LMB (鼠标左键)+MMB (鼠标中键), 可以放大和缩小视图; 使用Alt+MMB (鼠标中键) 可以上下、左右移动视图。当然, 在其他视图中也一样, 可以通过点击一个节点来

选中它。

试着将鼠标指针移动到一些连接上。当这样做时，能看到被连接属性的名字。如果想删除一个连接，可以用鼠标选中该连接并按（退格键）**Backspace**或者（删除键）**Delete**。如果想要建立新的连接，可以在按下鼠标中键的状态下，将鼠标指针从一个节点拖动到另一个节点，这个操作将弹出装载有这两个节点的连接编辑器，这时就可以建立属性之间的连接。

连接的颜色表明箭头线所代表的被连接属性的数据类型是简单数字类型（蓝色）、数组类型（绿色），还是如矩阵或几何图形这样一些更加复杂的数据类型（粉红色）。

在图1.10中，一个名为**nurbsSphere1_translateY**的动画曲线连接到梯形的**nurbsSphere1**节点。这个节点被称为变换节点，而与之相关的**nurbsSphere1**节点被称为形状节点。本章稍后有对变换节点与形状节点之间关系的讨论。现在，知道变换节点定义了小球在空间中的位置，形状节点定义了小球的几何形状就足够了。

早先讨论的两个连接没有在超图窗口的视图中出现。第一个是从变换节点到形状节点的连接，第二个是从被称为**time1**的Maya场景时钟节点到动画曲线的连接。因为大多数动画曲线节点都是由当前时钟驱动，并且大多数形状节点都连接到名称相似的变换节点，为避免混乱，在默认情况下这两种连接都被隐藏。

现在，来看一个更复杂的场景。首先将创建一系列的关节点以使球体对象弯曲，然后看一看所生成节点是如何连接的。

1. 建立一个新场景。
2. 翻转透视视图，使**XZ**平面朝向自己，创建球体对象。
3. 将此球体在**Scale X**域的值设置为3。
4. 选择关节点工具（**Joint Tool**），在球体对象内建立3个关节点，按下回车键以完成操作。
5. 选定这些关节点及球体对象，然后选择**Skin>Bind Skin>Rigid Bind**菜单项。
6. 选中中间的关节点，并选择旋转工具（**Rotate Tool**），然后旋转该关节点以查看球体对象被关节点所限制的情况。
7. 选定球体对象，并从主菜单中选择**Window>Hypergraph**菜单项。
8. 选择超图窗口中的**Graph>Up Connections**和**Graph>Stream Connections**菜单项。结果如图1.12所示。

在本例中，跟踪浏览粉红色的连接会使人对Maya在计算关节点如何影响球体形状时操纵几何图形的方式有个大体的概念。在图1.12中最左端是**makeNurbSphere1**历史节点，此历史节点被连接到老的**nurbsSphereShape1**节点，这个老的**nurbsSphereShape1**节点被重新命名为**nurbsSphereShape1Orig**并处于隐藏状态。在超图窗口中隐藏节点以灰色显示。

进一步往右看，在选择**Rigid Bind**菜单时，Maya自动建立的很多变形（**tweak**）节点、组（**group**）节点和关节群节点之后是新的**nurbsSphereShape1**节点，它包含被变形的几何图形。最后，在超图中包含一些隐藏群的句柄，这些隐藏群包括由每一关节点所控制的点所构成的群和定义每一关节点所影响点的集合节点。