

DVD DVD DVD DVD
ROM ROM ROM ROM

超值

20小时高清视频教程
素材及工程文件

Maya

总动员

——动画编程篇

张宝荣 吕新欣 等编著

清华大学出版社

2001 2002 2003 2004
2005 2006 2007 2008
2009 2010 2011 2012
2013 2014 2015 2016

Mava

2016 2017 2018 2019 2020

2021 2022 2023 2024 2025

2026 2027 2028 2029 2030

2031 2032 2033 2034 2035

DVD DVD DVD DVD
ROM ROM ROM ROM

超值 20小时高清视频教程
素材及工程文件

Maya 总动员

——动画编程篇

张宝荣 吕新欣 等编著

清华大学出版社
北 京

内 容 简 介

本书以 Maya 2010 版及 Maya 2011 版本为依托,详细地介绍了各种 Maya 动画编程技术。具体内容包括动画表达式、粒子表达式、常用函数、表达式编辑器、MEL 脚本基础内容、数值与变量、数据类型、创建界面、句法、脚本范例、Python 的使用、Python 和线程、MEL 与 Python 的重要差异、程序编译简介、导出插件程序、使用 API 编辑节点、API 多线程插件、多边形 API 和 Maya Python API、分配 Maya 插件等。

本书附带 4 张 DVD,近 20 小时的视频教学,全方位地以视频教学的方式展现了 Maya 的强大程序功能,使学习枯燥的程序编写变得简单易懂。光盘内容包括表达式的常用函数学习、表达式的应用、粒子表达式应用、MEL 脚本命令操作、函数定义操作、自定义窗口、Python 介绍、Python 语法规则、Python 元素的引用方法、Python 内置函数、Maya API 插件编写及应用实例等。光盘内容是图书内容的延伸和补充,是全面学习 Maya 程序高级功能最为经典的不可多得的珍贵教学资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Maya 总动员:动画编程篇 / 张宝荣等编著. —北京:清华大学出版社, 2011.1

ISBN 978-7-302-24178-2

I. ①M… II. ①张… III. ①三维-动画-图形软件, Maya IV. ①TP391.41

中国版本图书馆 CIP 数据核字(2010)第 234639 号

责任编辑:冯志强

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:38.25 字 数:955 千字

附光盘 4 张

版 次:2011 年 1 月第 1 版

印 次:2011 年 1 月第 1 次印刷

印 数:1~4000

定 价:79.80 元

前言

当今三维动画领域（包括影视动画、电视广告、3D动画片、游戏、建筑、产品展示、可视化……）的软件可谓是百花争鸣，但其中却是良莠不齐，绝大部分软件功能都不是很全面，因此严重地影响了软件的使用率。而唯独 Maya 拥有着全面的功能，而且用户群非常之大，因此 Maya 也就成为了影视动画的工业标准。

Maya 这个软件工具，作为 3D 动画领域中的著名软件，其强大的功能已经被世人所熟知。在中国几乎所有的三维制作公司都用 Maya 作为制作工具。但无奈，由于技术原因，Maya 功能的利用率在国内是很可怜的，基本都在 20% 以下，远远达不到国外的利用率。究其原因，就是国内的制作人员大多是学艺术出身，所以在 Maya 程序这一块很难跃进，使 Maya 这一最强有力、最核心的工具无法发挥出它强大的威力，这样就造成了国内制作水准较之国外要差很多。

我们经常看的国外大片如《2012》、《阿凡达》、《变形金刚》、《指环王》、《哈利波特》、《金刚》、《黑客帝国》、《后天》等，几乎所有品质一流的影片，都使用 Maya 作为其主要制作工具。而众多制作公司选用 Maya 的唯一一个原因，就是 Maya 的开放性非常好。所谓 Maya 的开放性就是利用 Maya 的编程工具进行程序设计，以拓展 Maya 的功能。而在这一块，恰恰是国内无法涉及的领域，现在市面上几乎所有的书，都是围绕 Maya 软件界面的某些现成的功能来讲述的，都是千篇一律，不外乎就是建模、材质灯光、角色动画、动力学特效等。整体来说技术价值不高，含金量低。

如果想在 Maya 技术上有所突破，真正达到或是近似于好莱坞大片的效果，那必须利用 Maya 的程序拓展功能来实现，这是一条必由之路。Maya 在国内已发展了十多年，它的精髓何在大家都心知肚明，那就是 Maya 强大的程序内容。其实很多人都对 Maya 的程序开发这一块内容垂涎已久，但无奈的是目前国内还没有系统全面介绍该技术的书籍或是资料，使广大 Maya 用户只能观望而无法有效利用。藉于此，我们看到了国内用户之所求，看到 Maya 编程技术之空白，所以十分有必要策划一部内容全面翔实的 Maya 程序使用书籍。

Maya 程序主要包括三大部分：

第一是表达式（Expression），它可以完成传统方法无法完成的动画效果，例如大片里的变形金刚的变形动画效果，就是充分利用 Maya 表达式动画制作的，因此这是制作高度复杂动画的必用工具。

第二是 MEL 脚本及 Python 脚本，它们主要的作用是提高工作效率，制作中经常有重复性的工作，我们可以把重复的工作写成 MEL 脚本，这样即可省掉大量的手工劳动，从而提高效率。一个复杂的项目如果需要一个月时间，那如果充分利用 MEL 脚本可以把工期缩短至一周。

第三是 Maya 的 API 部分，这是一个程序接口，也就是说，在需要使用 Maya 软件没有涉及的功能时，制作人员可以自己为其添加功能，这也是 Maya 最重要的拓展工具，我们可以使用 C++ 语言来扩充 Maya 的功能。

如果能对 Maya 的这三部分程序内容善加利用的话，便可以充分挖掘出 Maya 深层次的功能以及无限的拓展能力，只有这样国内三维动画的制作技术才能达到一个新的高度，才能有所突破与创新，才能和国际化制作接轨。

对于很多人来说，只要提到编程就可能感到害怕或是厌烦，这是可以理解的，尤其是 Maya 软件的用户，他们大多不是学习计算机编程专业出身，对于这些程序更是头疼。鉴于此，本书假设读者没有任何的编程经验，完全从零学起，打破只有专业程序员才能对 Maya 编程的神话。当你牢固掌握本书内容后，任何人都可以在 Maya 的程序上做到游刃有余，使自身的制作能力有一个质的飞跃！这正是我们的目标，也是这本书的宗旨！

Maya 的程序内容是 Maya 软件的精髓所在。本书以 Maya 2010 版及 Maya 2011 版本为依托，详细地介绍了表达式、MEL 脚本、Python 脚本，并且重点讲述了 Maya 的 API 内容。具体内容包括动画表达式、粒子表达式、常用函数、表达式编辑器、MEL 脚本基础内容、数值与变量、数据类型、创建界面、句法、脚本范例、Python 的使用、Python 和线程、MEL 与 Python 的重要差异、程序编译简介、导出插件程序、使用 API 编辑节点、API 多线程插件、多边形 API 和 Maya Python API、分配 Maya 插件等。书中内容详细而丰富，是学习 Maya 程序高级功能不可多得的珍贵图书。

本书使用超过一半的篇幅，重点学习了 Maya API 的全部内容。讲述 Maya API 的高难技术，使其得以全面地向读者展现她那神秘而又无敌天下的强大功能。更为难得的是随书配备的大容量 DVD 光盘，更是以视频教学的方式，淋漓尽致地向读者展示了 Maya 她那迷人的魅力。使读者学习复杂的程序变得非常方便和易懂。

本书特别适合 Maya 的初、中、高级用户学习使用，更加适合渴望提高 Maya 高级应用技术的成熟用户，是全面学习 Maya 程序高级功能的不可或缺的资料。

本书在编写过程中得到了很多业内专家的指导和帮助，除了两位作者张宝荣和吕新欣外，参与编写的人员还有：赵辉、于治民、焦福海、张进霞、张庆中、尚东风、陈钢、马鹏程、王小慧、程瑞霞、白丽娜、余凤、史振旭、万凌清、杨庆、刘音、李孝楠、钱挺、刘伟、钟来进、李远剑、邹凯、姜忠雨、何艺蕾等。他们付出了辛苦的劳动，在此一并表示感谢。书中难免有不妥或值得商榷之处，希望读者批评指正。谢谢！

编 者

2010 年 8 月

目 录

第 1 章 表达式..... 1

1.1 动画表达式..... 1

1.1.1 动画表达式..... 1

1.1.2 创建动画表达式..... 2

1.1.3 每个属性只有一个 驱动程序..... 3

1.1.4 时间和帧关键词..... 3

1.1.5 找到之前创建的 动画表达式..... 4

1.1.6 编辑动画表达式中的 文本..... 6

1.1.7 使用文本编辑器编辑 动画表达式..... 7

1.1.8 删除动画表达式..... 10

1.2 粒子表达式..... 10

1.2.1 关于粒子表达式..... 10

1.2.2 粒子表达式..... 11

1.2.3 创建表达式的执行..... 12

1.2.4 运行时间表达式 的执行..... 12

1.2.5 设置动态开始帧..... 12

1.2.6 设置原始状态使用 的属性..... 13

1.2.7 编写创建表达式..... 13

1.2.8 编写运行时间表达式..... 14

1.2.9 运行粒子属性..... 18

1.2.10 添加动态属性..... 18

1.2.11 了解每粒子和每对 象属性..... 18

1.2.12 了解原始状态属性..... 19

1.2.13 赋值自定义属性..... 23

1.2.14 赋值不同长度的粒子

排列属性..... 24

1.2.15 在运行表达式中使用

创建表达式的数值..... 25

1.2.16 运行位置、速度和

加速度..... 25

1.2.17 运行颜色..... 27

1.2.18 运行发散的粒子..... 28

1.2.19 运行碰撞..... 28

1.2.20 运行使用期..... 32

1.2.21 运行特定粒子..... 32

1.2.22 赋值矢量和矢量

排列..... 35

1.3 常用函数..... 42

1.3.1 极限函数..... 42

1.3.2 指数函数..... 45

1.3.3 三角函数..... 46

1.3.4 矢量函数..... 54

1.3.5 转化函数..... 56

1.3.6 数组函数..... 58

1.3.7 随机数函数..... 59

1.3.8 曲线函数..... 64

1.3.9 通用命令..... 70

1.4 表达式编辑器..... 73

第 2 章 MEL 脚本..... 76

2.1 背景情况..... 76

2.1.1 MEL 概述..... 76

2.1.2 MEL 和表达式手册..... 76

2.1.3 编程器的 MEL..... 76

2.2 运行 MEL..... 79

2.2.1 运行 MEL 命令..... 79

2.2.2	查看或记录与作用 相关的 MEL 命令	81	2.6.11	continue	99
2.2.3	为脚本制作一个 工具栏按钮	81	2.6.12	测试命令、对象和 属性的存在	99
2.2.4	从 MEL 命令上得 到帮助	82	2.6.13	=和=的不同	100
2.3	数值和变量	82	2.6.14	常见问题	100
2.3.1	整数和浮点数	82	2.7	属性	102
2.3.2	字符串	82	2.7.1	属性概述	102
2.3.3	显式输入和隐式输入	83	2.7.2	属性名称	102
2.3.4	变量	83	2.7.3	属性的数据类型	103
2.3.5	分配数值到变量和 属性上	84	2.7.4	获取和设置属性	104
2.3.6	输出数值	85	2.7.5	获取和设置多值 属性	105
2.3.7	选择任意数字	86	2.8	程序	107
2.4	排列、矢量和矩阵	86	2.8.1	程序	107
2.4.1	排列	86	2.8.2	定义程序	107
2.4.2	获取和改变排列的 大小	87	2.8.3	调用程序	109
2.4.3	矢量	88	2.8.4	全局和本地变量	110
2.4.4	矩阵	89	2.8.5	测试功能在 MEL 中 是否可用	111
2.5	类型	89	2.8.6	检查程序的来源	112
2.5.1	MEL 合适类型	89	2.9	I/O 和互动	112
2.5.2	使用空白区域	90	2.9.1	用户互动	112
2.5.3	添加注释	90	2.9.2	读取和编写文件	114
2.5.4	变量命名	91	2.9.3	测试文件存在、权限 和其他属性	115
2.5.5	程序和脚本	91	2.9.4	控制文件	116
2.5.6	完美脚本	92	2.9.5	运行目录	116
2.6	控制脚本流量	92	2.9.6	执行系统命令	117
2.6.1	测试和比较数值	92	2.9.7	读取并写入系统 命令管线	118
2.6.2	布尔值	93	2.9.8	从 AppleScript 中调用 MEL, 反之亦然	119
2.6.3	if...else if...else	93	2.10	调试、优化和疑难解答	120
2.6.4	? : 运算符	94	2.10.1	MEL 调试特点	120
2.6.5	switch...case	94	2.10.2	优化脚本和表达式 速度	122
2.6.6	while	96	2.10.3	疑难解答	125
2.6.7	do...while	97	2.11	创建界面	129
2.6.8	for	97	2.11.1	ELF 命令	129
2.6.9	for-in	98			
2.6.10	break	98			

2.11.2	窗口	129	2.15.3	建模	175
2.11.3	控件	130	2.15.4	动画、动力学 和渲染	176
2.11.4	布局	130	2.16	脚本范例	178
2.11.5	分组	134	2.16.1	从 Maya 自带的 脚本文件中学习	178
2.11.6	菜单	135	2.16.2	从文档文件中读取 动画参数值	178
2.11.7	集合	135	2.16.3	粒子碰撞边界	179
2.11.8	母代与子代	135	2.16.4	点爆炸	181
2.11.9	命名	137	2.16.5	测试附加粒子属性	184
2.11.10	UI 命令模板	138	2.16.6	测试动力学事件	187
2.11.11	删除 UI 元素	139	2.16.7	动态时间播放	190
2.11.12	连接命令到 UI 元素	139	2.16.8	寻找着色对象	192
2.11.13	简单窗口	141	第 3 章	Python	194
2.11.14	模式对话框	142	3.1	Python in Maya	194
2.11.15	使用系统事件和 scriptJobs	143	3.2	Python 的使用	195
2.12	脚本节点	145	3.3	Python 和线程	203
2.12.1	MEL 脚本节点	145	3.4	Python from an external interpreter	204
2.12.2	生成或编辑脚本 节点	145	3.5	MEL 与 Python 中的重要 差异	205
2.12.3	打开文件时阻止脚 本节点的运行	146	3.6	当前限制条件	207
2.13	高级版本	147	3.7	Tips and tricks for scripters new to Python	208
2.13.1	高级程序课题	147	第 4 章	程序编译和导出端口	212
2.13.2	高级动画表达式 课题	150	4.1	程序编译	212
2.14	句法	164	4.1.1	程序编译简介	212
2.14.1	祈使句法	164	4.1.2	FBX 文件程序编译	212
2.14.2	分隔符和空白区域	166	4.1.3	IGES 程序编译插件	214
2.14.3	表达式、运算符和 语句	167	4.1.4	StudioImport 插件 程序和程序编译	216
2.14.4	运算符优先顺序	168	4.1.5	导出插件程序	217
2.14.5	组块	168	4.1.6	Maya OpenFlight 导入 端口/导出端口	220
2.14.6	注释	169	4.2	游戏	230
2.14.7	表达式和 MEL 句法之间的不同	170	4.2.1	简介	230
2.15	疑难问题解答	171			
2.15.1	任务	171			
2.15.2	脚本与句法	172			

4.2.2 VRML2 程序编译	233	7.5 uninitializedPlugin()	271
4.2.3 RTG 应用程序和 文件格式	241	7.6 创建者方法	271
4.2.4 游戏交流 2 程序 编译	243	7.7 MpxCommand	271
4.2.5 MDt API	247	7.8 返回结果到 MEL	278
第 5 章 Maya API 简介	253	7.9 语法对象	278
5.1 简介	253	7.10 标志	279
5.2 安装插件程序	255	7.11 背景	281
5.3 卸载插件程序	255	7.12 MPxContext	281
5.4 编写简单的插件程序	256	7.13 MPxContextCommand	285
5.5 重要的插件程序特性	257	7.14 工具属性表	287
5.6 与 Maya 组合	260	7.15 MPxToolCommand	287
5.7 MObject	260	7.16 连接插件到 Maya 菜单	296
5.8 包装器	261	7.17 第三方插件程序的安装	297
5.9 对象和函数集	261	第 8 章 DAG 层次	298
5.10 命名惯例	262	8.1 DAG 层次概述	298
5.11 添加参数	262	8.2 节点	298
5.12 错误检查	263	8.3 变形和形状	299
5.13 MStatus 类	263	8.4 DAG 路径	299
5.14 错误记录	264	8.5 一般化的实例化	301
第 6 章 使用 API 选择	265	8.6 多个形状的变形	301
6.1 使用 API 选择概述	265	8.7 DAG 行走实例	302
6.2 MGlobal::setActive- SelectionList()	265	第 9 章 编写阴影节点	309
6.3 MSelectionList	265	9.1 阴影节点插件程序概述	309
6.4 MItSelectionList	266	9.2 编写阴影节点插件程序	309
6.5 setObject() 方法	267	9.3 阴影节点插件程序分析	310
6.6 MFn::Type 列举	268	9.4 InterNode 实例编码走查	312
6.7 MGlobal::selectByName()	268	9.5 阴影节点分类	320
第 7 章 命令插件	269	9.6 Hypershade 的阴影节点图标	323
7.1 添加命令给 Maya 概述	269	9.7 特殊的阴影节点	323
7.2 注册命令	269	9.8 阴影节点里的超级采样	323
7.3 MFnPlugin	269	9.9 赋值渲染背景外的阴影节点	324
7.4 initializePlugin()	270	第 10 章 从属图表插件程序	325
		10.1 从属图表插件程序概述	325
		10.2 母代类别描述	325
		10.3 基础	326

10.4	从属图表 (DG) 节点	330	12.1.1	硬件着色节点插件 程序概述	365
10.5	节点	332	12.1.2	执行硬件着色节点 插件	366
10.6	属性和塞子	333	12.1.3	一个硬件着色节点 插件的实例	370
10.7	复杂属性	333	12.1.4	改进硬件着色器的 运行	379
10.8	资料区	335	12.1.5	硬件着色节点插件 实例	379
10.9	数据句柄	335	12.2	编写自定义变形节点	380
10.10	数据创建式	336	12.2.1	自定义变形概述	380
10.11	计算方法	336	12.2.2	执行自定义变形	381
10.12	更复杂的实例	337	12.2.3	自定义变形实例	382
10.13	MPxNode 及其衍生类别	339	12.2.4	自定义变形节点 实例	388
第 11 章	操纵器和形	342	12.3	编写变形器节点	388
11.1	操纵器	342	12.3.1	变形器概述	388
11.1.1	操纵器是什么	342	12.3.2	执行变形器节点	389
11.1.2	基础操纵器	342	12.3.3	变形器节点实例	389
11.1.3	编写操纵器	344	12.3.4	变形器实例	392
11.1.4	操纵器容器	345	12.4	编写文件编译程序	392
11.1.5	操纵器和节点间的 信息交流	347	12.4.1	文件编译程序概述	392
11.1.6	连接操纵器到 Show Manipulator Tool	351	12.4.2	执行文件编译程序	393
11.2	形状	354	12.4.3	文件编译程序实例	393
11.2.1	Maya 形状	354	第 13 章	多线程插件	394
11.2.2	形状分类	354	13.1	简介	394
11.2.3	写出形状	355	13.2	数据和任务对应	394
11.2.4	绘图与刷新	356	13.3	线程	394
11.2.5	选择	358	13.4	锁定	395
11.2.6	组件	358	13.5	OpenMP	396
11.2.7	调整和内部属性	360	13.6	Intel 线程建构 模块 (TBB)	398
11.2.8	几何体数据	361	13.7	线程任务	398
11.2.9	文件 IO	361	13.8	内存分配	402
11.2.10	变形器	361	13.9	线程和 Maya	402
11.2.11	标记菜单	363	13.10	其他对应类型	408
11.2.12	形状实例	363	13.11	蟒蛇线程和 Maya	409
第 12 章	编写节点和文件编译程序	365			
12.1	编写硬件着色节点	365			

13.12	第三方工具.....	409	15.6	Windows 环境（32 位和 64 位）.....	472
13.13	API 类别和插件.....	410	15.7	安装 Maya Plug-in Wizard	474
13.14	Mac OS X 平台特定 线程库问题.....	411	15.7.1	Maya Plug-in Wizard 2.0（32 位的 版本）.....	474
13.15	供应商特定 OpenMP 问题.....	411	15.7.2	Maya Plug-in Wizard 2.0（64 位的 版本）.....	475
第 14 章 多边形 API 和 Maya Python API.....			475	15.8	Mac OS X 环境.....
14.1	多边形 API.....	412	第 16 章 分配 Maya 插件.....		
14.1.1	多边形 API 概述.....	412	16.1	分配单独文件.....	478
14.1.2	如何内部控制 多边形.....	412	16.2	分配多个文件的模块.....	479
14.1.3	五个基本多边形 API 类别.....	418	16.3	模块限制.....	480
14.1.4	构建历史和调整.....	422	16.3.1	共享库.....	480
14.1.5	polyModifierCmd 实例.....	424	16.3.2	渲染全能的着色器.....	481
14.1.6	splitUVCmd 实例.....	440	第 17 章 实例插件.....		
14.1.7	多边导出端口插件.....	452	17.1	实例插件概述.....	482
14.2	Maya Python API.....	453	17.2	独立应用程序举例.....	488
14.2.1	Maya Python API 简介.....	453	17.3	C++插件实例描述.....	488
14.2.2	开发工具.....	453	17.4	独立应用程序的描述举例.....	526
14.2.3	使用 Maya Python API.....	454	17.5	着色器源代码举例.....	526
14.2.4	C++ Maya API 和 Maya Python API 的差别.....	466	17.6	ASHLI 着色器.....	533
第 15 章 建立创建环境.....			482	17.7	脚本插件程序举例.....
15.1	创建插件和应用程序简介.....	468	第 18 章 附录.....		
15.2	编译程序的要求.....	468	18.1	附录 A: NURBS 图形.....	546
15.3	Linux 环境（64 位）.....	468	18.2	附录 B: 相关图渲染节点.....	549
15.4	Linux 编译程序需要.....	470	18.3	附录 C: 渲染属性.....	551
15.5	使用调试器调试插件.....	471	18.4	附录 D: 常见问题解答.....	556
			18.5	附录 E: API 和 Devkit 限制条件.....	571
			18.6	附录 F: 添加图像插件 程序.....	573

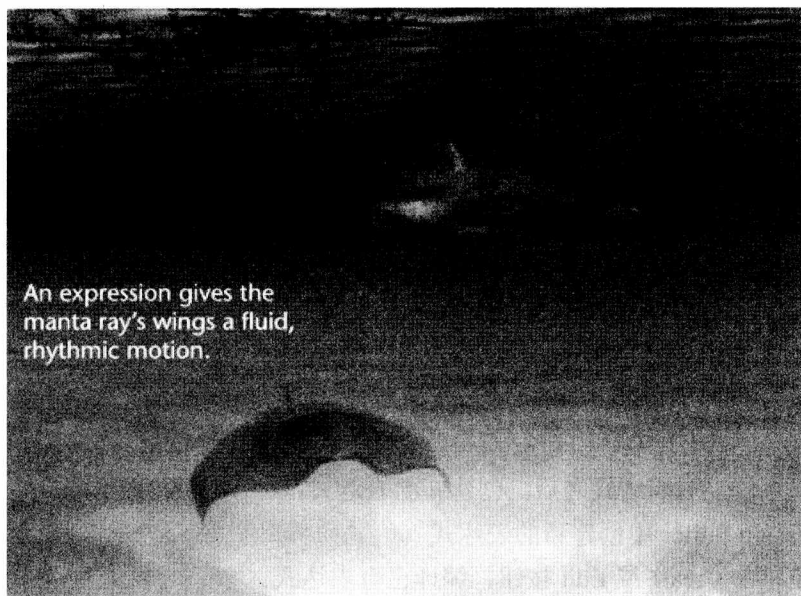
第 1 章 表 达 式

1.1 动画表达式

1.1.1 动画表达式

表达式是给定 Maya 在时间上控制属性的指令。属性是对象的特性，如 X 缩放、Y 平移、可见性等。

虽然可以创建一个表达式，为任意目的动态改变属性，但这是属性在时间上递增，随机、有节奏改变的一个概念标准，如图 1-1 所示。



Eric Saindon

图 1-1

表达式也可以用于链接不同对象间的属性——一个属性上的改变也会更改另一个的运行。例如，可以旋转依赖于汽车前后移动的轮胎。

表达式为困难的格点设定任务提供选择。在格点设定中，设置动画中选定关键帧的属性值，Maya 在关键帧之间插入操作。根据表达式，编写一个方程式，然后动画播放时 Maya 执行操作。

表达式总是简单的几个词或几行。下列表达式实例中，注意范围的变化和细节（不是其目的）。

实例

```
Ball.translateX = Cube.translateX + 4;
```

实例

```
if (frame == 1)
    Cone.scaleY = 1;
else
{
    Cone.scaleY = (0.25 + sin(time)) * 3;
    print(Cone.scaleY + "\n");
}
```

可以使用表达式动态改变任何帧范围内可设置关键帧、未锁定的对象属性。也可以使用表达式控制每个颗粒或每个对象的属性。每个颗粒属性分别控制颗粒对象的每个颗粒。每个对象属性共同控制颗粒对象的所有颗粒。

1.1.2 创建动画表达式

在 Expression Editor 中创建并编辑表达式。开启 Expression Editor 有几种方法:

- 选择 Window→Animation Editors→Expression Editor。
- 在 Channel Box 中, 单击属性名称, 然后右击, 从弹出的菜单中选择 Editors→Expressions。
- 在 Attribute Editor 中, 在属性框中右击, 选择 Create New Expression。

如果之前创建了一个表达式, 分配数值到属性, 选择 Edit Expression 代替, 如图 1-2 所示。

表达式文本编辑框随着输入文本进行扩展, 所以可以编写不限定长度的表达式。也可以启动文本编辑框上边的 Editor 下拉菜单中的外部文本编辑器, 并在编辑器里编辑表达式。

也可以在编辑已有表达式之后, 创建新的表达式。

在 Expression Editor 中创建新的表达式

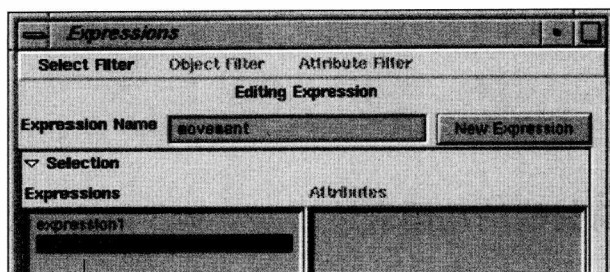
- (1) 确定单击 Create 或 Edit 按钮, 编译已有表达式。
- (2) 选择 Filter→By Expression Name。
- (3) 单击 New Expression 按钮。

这样清除了 Expression Name 框和表达式文本编辑框, 所以可以创建新的表达式。

创建表达式时, Expression Editor 与表达式的对象名称结合。这意味着使用除表达式名称外的对象名称可以缩小表达式的搜索范围。

不需要选择 Attributes 列表中的属性。可以使表达式只结合一个对象。

对于颗粒形状节点, 不需要选择属性, 因为可以创建每个颗粒形状只有一个创建表达式和两个运行时间表达式(动态计算前后)。对于非颗粒形状的对象, 可以创建每个属性一个表达式。



表达式列表

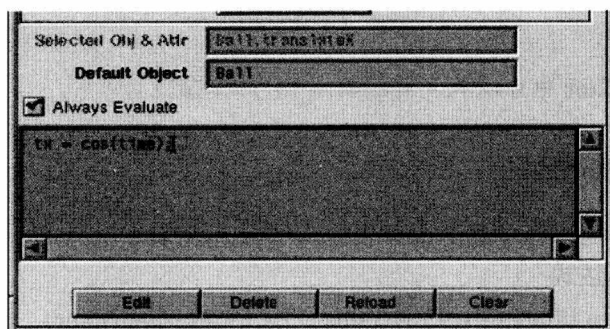


图 1-2

1.1.3 每个属性只有一个驱动程序

不能应用表达式到已经使用下列技术动态改变的属性上：

- 键
- 设置驱动键
- 约束条件
- 运动路径
- 另一个表达式
- 其他直接联系

如果应用了表达式，会在 Script Editor 和 Command Line 的反应区域看到错误信息。

虽然前面技术中的两个改变的一个属性不能被控制，但是可以控制使用关键帧属性，使用表达式的另一个属性，使用约束条件的另一个属性，等等。

也可以使用一个表达式分配数值到一个或多个对象的几个属性上。

1.1.4 时间和帧关键词

在动画表达式中，可以使用两个不在标准 MEL 中的非常有用的关键词：

- **time** 返回时间线上的当前时间位置。
- **frame** 返回时间线上的当前帧位置。

可以在动画表达式中式中使用这些关键词，就好像它们是变量：

```
persp.translateY = frame / 2;  
persp.rotateY = time;
```

1.1.5 找到之前创建的动画表达式

创建完表达式之后，会决定稍后进行更改创建不同的动画结果。要编辑表达式，在 Expression Editor 中进行显示。下面描述如何为编辑找到并显示表达式。

注意

在 Expression Editor 中可以找到并编辑 MEL 脚本节点。

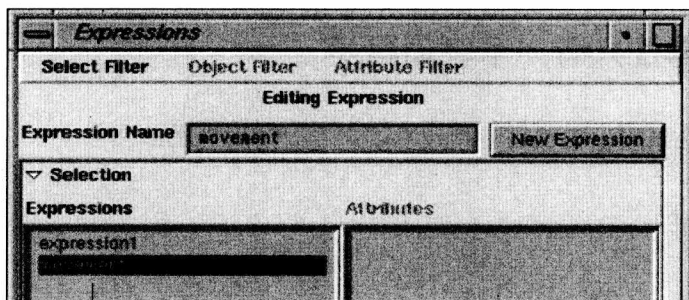
1. 通过名称找到表达式

要找到表达式，可以在场景的所有表达式列表中选择。

通过名称搜索表达式

(1) 从 Expression Editor 中，选择 Filter→By Expression Name。

Expressions 列表出现在 Expression Editor 中。列表显示了为这个场景创建的所有表达式。如图 1-3 所示。



表达式列表

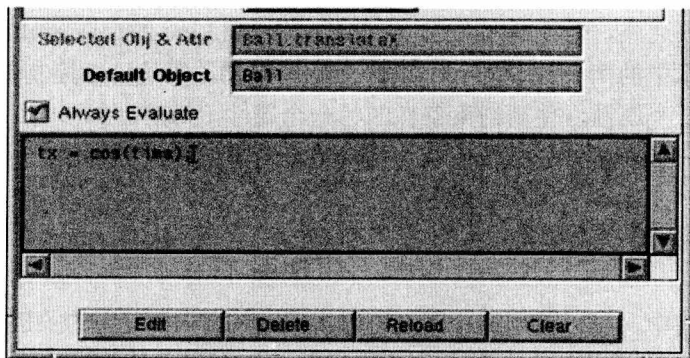


图 1-3

(2) 单击列表中的表达式。

表达式内容出现在表达式文本编辑框中。

如果忘记了表达式的名称，单击列表中的每个名称，直到预期表达式出现在表达式文本框中。

注意

对于颗粒形状节点，可以创建一个创建表达式，在动态执行之前创建一个运行时间表达式，动态执行之后创建一个运行时间表达式，或者三个都创建。所有的表达式都列表显示在一个名称下——颗粒形状节点的名称。不能命名或重命名这样的表达式。

要找到这样的表达式，在 Expressions 列表中寻找颗粒形状节点的名称。

单击适当的动态前/后 Runtime 或 Creation 复选框，显示想要的表达式。

2. 通过选定对象找到表达式

如果忘记了表达式的给定名称，可以选择受影响的对象找到表达式。对于非颗粒形状节点，也可以从 Attributes 列表中受影响的属性缩小表达式的搜索范围。

通过对象和属性名称搜索表达式

(1) 选择 Outliner、Hypergraph 或工作空间中的对象或节点。

(2) 在 Expression Editor (Window→Animation Editors→Expression Editor) 中，选择 Filter→By Object/Attribute Name。这是 Expression Editor 的默认搜索设置。

(3) 选择 Object Filter→Selected Objects。

选定的对象名称和适当属性显示在窗口中，如图 1-4 所示。

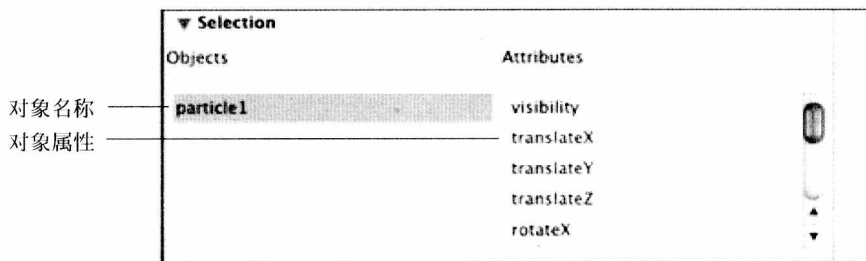


图 1-4

(4) 对于颗粒形状节点以外的对象，单击表达式控制的属性名称。

如果忘记了表达式控制的属性名称，选择 Attribute Filter→Connected to Expressions。Attributes 列表中只显示选定对象的表达式控制的属性。单击 Attributes 列表中的每个属性，直到在表达式文本编辑框中看到想要的表达式。

不能像对其他类型的对象那样为颗粒形状的每个属性编写不同的表达式。因为每个颗粒形状只能编写一个创建表达式和两个运行时间表达式（动态计算前后），不需要从 Expression Editor 的 Attributes 列表中选择属性。