

COMPLETE MAYA PROGRAMMING
An Extensive Guide to MEL and the C++ API



Maya 5 编程全攻略

〔美〕 David A.D.Gould 著

夏旺盛 于旭升 等译

如果你希望开启**Maya**神秘的
工具箱，这本书就是你的向导

M K[®]
MORGAN
KAUFMANN



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

STUDENTS OF THE MAYA 5 PROGRAM CAN RECEIVE
A 10% DISCOUNT ON THE BOOK PRICE.



Maya 5 编程全攻略

作者：[美] 约翰·H·汉密尔顿
译者：陈伟 陈伟 陈伟

本书是 Maya 5 编程的权威指南

它涵盖了 Maya 5 编程的所有内容

清华大学出版社

清华大学出版社
地址：北京清华大学学研大厦
邮编：100084
电话：(010) 62770175

COMPLETE MAYA PROGRAMMING

An Extensive Guide to MEL and the C++ API

Maya 5编程全攻略

〔美〕 David A. D. Gould 著

夏旺盛 于旭升 等译

電子工業出版社.

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是一本实用的、逐步引导的指南,它从最基本的知识开始,向读者展示了如何最大限度地利用Maya所带来的优势。本书主要内容包括:如何利用MEL(Maya Embedded Language)控制Maya并将多种操作自动化;如何利用C++应用编程接口开发自己的自定义工具和功能。通过书中大量详尽的例子,读者能够很快就完全控制Maya的各个方面,并根据自己的需要将它扩展。

本书既是Maya初学者的一本基本教程,又是有经验的开发人员的一本很好的参考书。



MORGAN KAUFMANN

Copyright©2003 by Elsevier Inc.

Translation Copyright©2004 by Publishing House of Electronics Industry & Beijing Media Electronic Information Co., Ltd. All rights reserved.

本书英文版由Elsevier公司出版,Elsevier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:01-2004-0338

图书在版编目(CIP)数据

Maya 5编程全攻略/(美)古尔德(Gould, D. A. D.)著;夏旺盛等译.一北京:电子工业出版社, 2004.7

书名原文: COMPLETE MAYA PROGRAMMING

ISBN 7-120-00104-3

I. M… II. ①古… ②夏… III. 三维—动画—图形软件, Maya 5 IV. TP319.41

中国版本图书馆CIP数据核字(2004)第052170号

责任编辑:徐云鹏 朱 巍

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:22.875 字数:580千字

印 次:2004年7月第1次印刷

定 价:36.00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

David Gould在计算机图形行业已经有逾十年的经验，同时在程序设计和动画制作方面有很深的造诣。这种将技术和艺术才华集于一身的罕见能力为他带来了很多的奖项和荣誉。他曾在一项惊人的技术混合开发（包括为Pangolin开发的一种获奖的激光渲染系统）中扮演着关键角色；他还开发出用于控制Kuper运动控制装置以及Monkey停止运动puppet的软件。他曾亲自开发出Illustrate！这一市场领先的动画和技术图像渲染软件，该软件已得到NASA、英国航空、沃尔特·迪斯尼、索尼影视娱乐等公司的广泛使用。

David的职业生涯遍及各大洲的多家公司。在巴黎，他主持制作了多部三维立体科技影片，其中包括获奖影片“Inside the Cell”。在伦敦，他开发出一套已申请专利的面部动画系统。在纽约，他加入后期制作界，参与制作了许多出色的商业广告片，这些都进一步丰富了他的制作经验。

在位于洛杉矶的沃尔特·迪斯尼动画制作公司时，David开发出一种尖端的动画和建模技术，以用于他们的动画影片制作。后来他加入了位于伯克利市的Exluna，这是一家由包括Larry Gritz在内的前Pixar公司的渲染研究人员创立的软件公司。在这里，他在Entropy（一种兼容Renderman的渲染软件）以及其他产品的设计和开发中扮演着重要角色。加入位于加利福尼亚州的圣克拉拉市的NVIDIA公司后，David通过协助设计未来的3D图形芯片而继续着他的渲染开发工作。此后他加入了位于新西兰的Weta Digital，参与《指环王》三部曲影片的动画制作。他在图像制作中从事着色器开发、灯光和CG特效等多项工作，这一切都展现了他全面的才华。

来自业内专家的赞誉

David Gould是Maya的使用、编程和教学方面的专家，而这正如本书所展示的，需要进行Maya编程的人会发现本书是如此不可或缺。即使是那些不用进行大量编程的Maya用户也应该阅读本书，因为它可以令你更好地了解Maya的奥妙所在。本书紧凑而又全面，同时涵盖了MEL和C++ API。无论是刚入门的新手，还是老练的程序员，都可以从本书中获得所需的知识。不可不读！！

——Larry Gritz, Exluna/NVIDIA

无论是刚入门的新手，或是经验丰富的程序员，所有的Maya编程人员都应阅读本书。如果你是新手，它为你提供了对Maya的全面、构思巧妙且实践性强的指南和介绍。而本书最大的贡献，则是David将他对Maya基础概念和体系结构的深入理解与读者共享，所以，即使你是经验丰富的老手，也可以从本书中学习到如何更有效地利用Maya的丰富而又强大的编程接口。

——Philip J. Schneider, 迪斯尼动画制作公司

在对David Gould的《Maya 5编程全攻略》进行技术评论后，我必须说本书在Maya的脚本和插件开发方面是权威而可靠的作品。以前从未有过这样的简洁清晰的Maya编程方面的指南。选择本书的聪明读者无疑会受益匪浅。

——Chris Rock, 北加利福尼亚州一家大型动画公司技术总监

如果你希望开启Maya神秘的工具箱，这本书就是你的向导。利用本书提供的清晰而又渐进的说明，或通过MEL脚本语言，或通过C++ API，你很快就能定制和改进自己的应用程序，以及创建自定义插件。

——Christophe Hery, Industrial Light + Magic

前言

毫无疑问，**Maya**是一种创建计算机图形的非常强大的工具。然而，同时必须声明的是，它在功能方面的深度和广度也使它成为一个庞大的软件包，从而难于学习，更不用说掌握了。但是，打开这本书，你会发现一切是那么明了，这种感觉也许你从未有过。这就像是你站在山脚下，准备开始远程攀登时，惊喜地发现旅途会变得更轻松。**Maya**可以用你以前认为是绝不可能的方式进行定制和扩展。许多日常工作都能够实现自动化或大大简化。你可以创建工具，这样不仅可以提高工作效率，而且可以实现更好的控制。通过**Maya**编程，所有这些都可以实现。当然，还有更多。

对于很多人来说，只要提到编程就可能会感到害怕和厌烦。这是可以理解的，因为许多编程类书籍都将读者的编程经验假定在一个较高的层次。本书假设读者没有任何编程经验，并以此来打破只有专业程序员才能对**Maya**编程的神话。牢固掌握基本概念后，任何人都可以利用**Maya**为编程提供的各种控件开始编程。一度被视为畏途的旅程也会变成一次收获能力和信心的宝贵体验。

有了富于经验和耐心的向导，任何旅途都会变得更轻松。所以，本书的出发点不是单纯的说教，而是从最基本的地方逐步引导你理解**Maya**的原理。我们首先从**Maya**的核心开始探索，了解如何维护和处理数据。这些知识至关重要，因为它是**Maya**所有功能的基石。利用**Maya**最为简单的编程语言，**MEL**（**Maya Embedded Language**，**Maya**嵌入式语言），就可以开始学习如何控制**Maya**并将多种操作自动化。然后我们将介绍**C++**应用编程接口（**API**）。有了对**C++**语言的基本了解，就可以快速学习如何开发自己的自定义工具和功能。通过对每个真实例子的详细介绍，你可以进一步了解如何使用和控制**Maya**的功能。而通过结合使用**Maya**和**C++**，你很快就能完全控制**Maya**的各个方面，并可以根据自己的需要将它扩展。

除了教你**Maya**编程的细节之外，本书还着重说明以某种特定方式来设计某物品的原因。通常，编程为解决某问题提供了无限的可能方法。但是，**Maya**有其特定的设计理念，理解这一点非常重要。本书介绍了一组设计程序的准则，以便你编写的程序能够在**Maya**内无缝集成和运行。理解了为什么要采用某种特定方法，你就能够按照其中蕴涵的基本思想来解决自己的问题了。在你学习完本书后，不仅仅灵感勃发想像出新的激动人心的一切可能事物，而且拥有必要的知识和技巧将其变为现实，这正是我的目标。

目 录

第1章 概述	1
1.1 Maya的可编程性	2
1.2 编程接口	3
第2章 Maya基本概念	7
2.1 Maya体系结构	7
2.2 Dependency Graph	8
第3章 MEL	37
3.1 概述	37
3.2 MEL编程语言	39
3.3 脚本	80
3.4 对象	91
3.5 动画	106
3.6 图形用户界面	143
3.7 表达式	181
第4章 C++ API	201
4.1 简介	201
4.2 基本概念	203
4.3 开发插件	212
4.4 命令	226
4.5 节点	252
4.6 定位器	304
4.7 操纵器	311

4.8 变形器	321
4.9 高级C++ API	331
附录A 附加资源	339
附录B C程序员与MEL	343
附录C 补充阅读	345
词汇表	349

概述

说到变化，可以肯定地说，再没有任何行业像计算机图形行业的变化这样剧烈。虽然其诞生较晚，却已经渗透到众多不同的领域，其中包括建模、动画和渲染等。而且，每个领域都正在不断取得令人振奋的进步。结果是，我们制作出了前所未有的足以乱真的图像；我们完成了复杂程度前所未有的科学模拟；我们享受着交互式游戏带来的前所未有的痴迷于其中的乐趣。这一切在以前都无法想像。

在计算机图形行业，变化不仅不可避免，而且正以越来越快的速度在进行。与现在新的发明和理念为人们认可并立即实施的速度相比，以前的稳步进展便相形见绌了。推动这一进展的动力，是人们对更高深度和复杂程度的图像和体验的无休止追求。我们似乎看不到这一追求的尽头，而计算机图形行业也一直向前发展，以追求“终极”体验。

Alias | Wavefront一直处于这一变革浪潮的最前沿。其不懈的长期研究和开发为我们带来了一些最为先进的工具和技术，Maya就是其中之一。自1998年推出以来，Maya已经成为全球最好的动画工作室和公司的首选计算机图形软件包。其全面的工具箱覆盖了建模、动力学、动画和渲染等各个领域，使其用于生产工作的所有方面。

为了跟上快速变革的步伐，Maya也在进行不断的更新和改进。虽然已经具有全面的功能组，Maya的设计者们仍然明白，他们永远难以提供每个用户想要的所有特性和功能。他们也不能将Maya设计得适合每个人所喜好的工作流程。所以，他们将Maya设计得非常灵活，使用户能够根据需要自由定制和扩展软件包。用户可以自由对Maya进行塑造和更改，以适合其特定的工作环境和工作流程。从更改主图形用户界面到无缝添加新特性，用户可使原软件变得面目全非但更适合自己的。

事实上，Maya可被视为制作计算机图形的一种开放式体系结构。可以将它看做是一种框架，在这一框架中有许多三维组件（对象、动画、动力学等），可以重新设定这些组件的用途以实现自己的自定义应用程序。例如，可以使用Maya来完成自定义模拟或实时交互任务。Alias | Wavefront的一名工程师Mike Taylor向我们很好地演示了Maya可重新设定用途的自由程度。通过使用一些简单的脚本，他能够将Maya转变为一个三维Tetris（俄罗斯方块）游戏。最终的界面和交互性与原来截然不同，人们简直难以相信它实际上就是Maya。

随着对Maya可以定制和扩展的自由程度的进一步了解，你会很快发现自己可以尽情发挥，而很少受到束缚。你的想像有多远，Maya就能支持你到多远。

1.1 Maya的可编程性

那么确切地说，用Maya能编写什么程序呢？本节介绍Maya功能的几个主要领域，这些功能可以使用编程界面直接访问和控制。

1.1.1 定制

Maya的整个图形用户界面（GUI）都采用Maya嵌入式语言（MEL）来编写和控制，所有图形用户界面元素的创建、修改和删除都用MEL语言来完成，也可以使用MEL语言来控制Maya界面。事实上，可以使用自己的MEL脚本来整个地替换标准Maya界面。在很多情况下，我们需要对Maya界面的某些部分进行专门的定制。例如，你可能希望开发一个特殊的界面，允许动画制作者不用学习Channel Box或其他Graph Editor就能使用软件。对于某些用户，也可以隐藏或移动多个Maya界面元素，以减少界面的复杂性。

除了用户界面以外，还可以定制Maya的内部设置。使用MEL，可以针对单个项目或整个系统来更改Maya的设置。例如，可以确保所有用户始终使用相同的时间、角度和长度单位。同时，任何用户每次打开某项目时，都可以应用不同的设置。

1.1.2 集成

在生产环境中，Maya经常不是惟一使用的软件包。人们可能仅用它的一部分功能，而用其他软件包来完成其他任务。因此，就需要在Maya和其他外部软件包之间进行数据传送。由于Maya标配有导入和导出某些数据格式的功能，其编程接口允许你编写自定义的数据导出程序和导入程序。这些程序也称为转换程序，因为它们将一种软件包中的数据转换为另一种软件包可以理解的格式。由于Maya允许你访问整个场景及其数据，你能够以任何需要的格式将此数据输出。例如，游戏公司通常需要将用Maya完成的作品（模型、动画等）转换为给定游戏引擎可直接使用的格式。

Maya的功能也可编译成一个独立的应用程序。这样，就可以结合使用Maya和自定义编程代码来提供一种完整的应用程序解决方案。

1.1.3 自动化

许多任务经常连续重复。编程非常适合于将重复性的任务自动化。用户只需编写一种MEL脚本即可将其完全自动化，而不必手动重复同一任务。无论是描影网络的应用、对象的定位还是任何需要重复进行的操作，都可以实现自动化。事实上，利用编程提供的通用性，可以根据当前环境使脚本执行不同的任务。例如，可以使一系列对象自动粘贴到另一位置的顶部或底部，而无需用户对每个对象逐一进行手动定位。

1.1.4 扩展

虽然Maya提供了非常广泛的功能集，用户仍需要使用自定义的工具和功能。幸运的是，Maya允许你添加自己的特性和功能。这些功能也可以与Maya原有的工具和功能无缝共用。从使用者的角度来看，自己创建的工具与标准的Maya工具没有任何区别。使用MEL和C++，可以为Maya软件包创建大型且众多的扩展工具和功能。

对于所扩展的功能，Maya几乎没有任何限制。举几个例子来说，它允许你创建自定义着色器、动力学、粒子、变形器和动画控制器等。事实上，创建的扩展特性和功能可以直接集成到Maya的核心部分Dependency Graph。集成后，这些功能的访问方式与标准Maya特性完全一样。

1.2 编程接口

影像制作者通常使用Maya的图形用户界面来完成其作品。该界面包括菜单、对话框和按钮等，使用户以可视化方式执行操作。此外，还有另外一种方式可以执行相同的操作。利用Maya编程，用户可以通过编写和执行一个程序来完成相同的任务。这一程序可以用MEL或C++这两种Maya编程接口中的一种来编写。

1.2.1 MEL

MEL是Maya Embedded Language（Maya嵌入式语言）的缩写。这是一种自定义编程语言，特别设计用于内部操作Maya。由于其更为简单的结构和语法，它比C++编程接口用起来更简单，使用面也更广。MEL的优势之一在于它是一种解释性语言。一般的编程语言要求你编译和链接源代码，而解释性语言则可以直接执行。这种直接执行指令的能力意味着MEL尤其适合于快速构建原型。由于无需编译和链接步骤，用户可以更轻松地设计和实现自己的构思。事实上，MEL可以完全在Maya内编写、调试和测试，而无需使用外部的编程器或调试程序。

由于MEL是一种解释性语言，也就有其固有的缺点：它运行的速度要比同等的C++程序慢得多。C++程序对源代码进行编译，生成实际的本地机器指令，因而运行很快；解释性语言则要在运行中对源代码进行解译。Maya在遇到MEL指令时，必须进行解释并最终转换为本地机器语言。虽然Maya努力加快这一过程，但很多情况下MEL在速度方面仍远远不及C++。尽管如此，考虑到C++程序的安装、复杂性和编译成本，快速创建和执行MEL程序所带来的优势在很多情况下仍很明显。这样，二者谁更有优势完全取决于所要执行任务的类型和复杂性。

1.2.2 C++

Maya可以使用标准的C++编程语言进行编程。虽然对于不了解这种语言的用户有些困难，但C++却是迄今为止扩展Maya的一种最强大的方式。使用C++，可以创建与软件包中其他工具和功能无缝共用的本地Maya插件。

通过C++ API（应用程序编程接口）提供了编程访问方式。该接口包括一系列的C++类库。要创建插件，只需编写一个C++程序，由该程序使用和扩展基本Maya类。因此，学习C++ API编程就涉及到学习各种不同的类以及如何使用它们。幸运的是，这些类的设计方式是一致的，在学习一些较简单的类之后将有助于学习较复杂的类。虽然这些类的数目最初可能令人生畏，但一个普通的Maya插件仅使用其中的一小部分。总的来讲，所有这些类中仅有约1/3是常用的。有些较深奥的类则很少用到。

1.2.3 用MEL还是用C++

对Maya进行编程有两种选择，现在就有选择使用哪种语言的问题了。在选择编程接口时，要根据自己的需要对二者进行权衡。最终的选择可能会取决于一些外部因素，比如紧迫的时间要求或特定的速度要求。总的来讲，MEL为你提供了所需的所有编程功能。它允许你无需C++就可以访问Maya的许多功能。在使用C++编程接口时，通常是要利用MEL接口中未提供的特殊功能。

还有很重要的一点需要理解，那就是选择一种编程语言不会自动排斥另一种。C++接口不是MEL的超集；也就是说，C++接口不包含MEL接口中所具有的一切，也不比后者包含的内容更多。MEL接口中的某些功能在C++接口中并不存在，反之亦然。因此，有些问题必须将二者结合使用才能解决。随着你对开发Maya更为熟练且有经验，同时熟练掌握MEL和C++ API将帮助你快速得出给定问题的最佳解决方案。而更好地理解这两种编程接口，就能够识别哪种接口更适合于给定任务的哪一部分，从而实现最佳搭配。

易于使用

选择使用何种编程接口可能仅取决于你的编程熟练程度。虽然MEL和C编程语言有惊人的相似之处，但正是其不同之处，使其更利于缺少经验的开发人员掌握和使用。由于舍弃了指针、内存分配和解除分配等结构，MEL更易于理解和使用。虽然MEL不提供C的底层访问功能，但在进行Maya编程时很少有这种需求。此外，没有这些结构可避免一些造成程序崩溃和整体不稳定的因素。

MEL脚本由于比其他语言更接近于伪代码而更易于阅读。MEL还更为“宽容”，因为它不强制进行类型检查。这意味着可以快速编写脚本，而不必担心变量是什么类型。这同时是把双刃剑，因为这样可能会产生一些以后很难发现的bug。幸运的是，可以通过显式定义变量的类型来改进类型检查。

如果你对C++更有经验，可能想要使用C++ API。Maya的C++类层级结构能够访问大部分的Maya功能。然而，仍然需要做一些MEL编程，哪怕是打算用C++编写所有程序。深入理解C++肯定会有助于学习MEL，因为MEL的语法非常类似于C。附录B描述了C和MEL编程语言之间的主要区别。

如果编程任务要求复杂的数据结构，最可能需要使用的是C++ API。MEL只包括了各种变量类型的有限集，不允许你定义自己的数据结构。

特殊功能

由于C++ API比MEL更复杂，人们很容易推断前者能够实现更高级别的访问和控制。这种推断在一定程度上是正确的。使用C++ API可以创建插件和编写软件，从而实现与核心Maya软件包更紧密的集成。但认为C++ API是MEL功能的超集则是错误的，事实上它们是互相补充的。

有些功能只能从MEL访问而不能从C++ API访问，反之亦然。所以为了执行某些任务，必须将二者结合使用而别无选择。事实上，常有一些C++插件要调用少许MEL命令。但最常见的是，MEL提供了足够的功能，以至于无需编写C++插件。

事情并非总是如此，C++ API提供的某些特性是MEL无法企及的，只有通过C++ API才能创建自己的节点。创建自己的节点意味着允许你创建自定义工具、变形器、形状、着色器、定位器、操纵器等等。MEL还不允许你创建自定义命令，但可以创建一些运行起来类似于某些命令的MEL过程。

无论选择哪种编程接口来开发核心功能，在开始处理任何图形用户界面问题时，必须使用MEL，因为它是控制界面的惟一方式。幸运的是，我们能够从C++ API开始执行MEL命令。

跨平台可移植性

考虑到Maya广泛部署于多种平台，可移植性就是一个首要考虑的问题。如果需要开发的产品只针对一种平台，就不存在这个问题了。但是，随着混合平台日益广泛用于数据处理和渲染，应该慎重考虑现在的决策可能会带来的潜在后果。

由于MEL是一种脚本语言，它被设计为可以轻松地移植到其他平台。MEL很少包括依赖于平台的功能。事实上，无论在哪个平台上运行，几乎所有命令都可以执行。这意味着在多种平台上部署MEL脚本通常更轻松，因为它很少依赖于特定平台的特性。由于MEL用于所有图形用户界面功能，你永远不必关心不同窗口系统的细节。用MEL在一个平台上开发的界面，在其他平台上的显示和操作都是类似的。

由于多种原因，开发跨平台的C++软件一直都极为困难。首先，不同的C++编译器有其自身的不兼容性。其次，它们遵照ANSI C++语言的程度也各有不同。此外，不是所有的编译器都使用相同的标准C库。由于Maya C++类和库很少依赖于其他外部库和高级语言特性，所以如果只使用它们的话，将使跨平台开发工作变得轻松。

在跨平台兼容性很重要的情况下，应尽可能采用MEL来开发功能。仅在绝对必要时才可使用C++。

速度

由于MEL是一种解释性语言，它可能比C++要慢。但并非总是如此，速度很大程度上取决于程序的复杂程度。对于某些任务，速度差别并不十分明显，也就无需另外用C++来编程。对于真正的时间关键型和复杂的操作，C++程序无疑会运行得更快些。事实上，速度可以提高10倍。当在场景中用到大而复杂的几何图形时，就应该考虑采用C++来编程。额外的C++编程工作可能会在普通用户交互性和整体生产效率方面为你带来丰厚的回报。

公开

在有些情况下，你可能希望发布某些功能给外来方，而又不想透漏如何实现此功能的细节。这通常意味着脚本和插件中包含了一些专有的算法或技术。要保护这些信息，同时又允许他人使用所生成的功能。

由于**MEL**脚本既是源代码，又是要最终执行的程序，所以没有方法将二者分开。用户要运行此脚本，就需要有**MEL**脚本。用户有了这些脚本，就可以将其在文本编辑器中打开，看出它是如何工作的。而**C++**插件采用的则是机器代码，它是编译和链接后的最终结果。由于最终结果采用机器代码的形式，用户就不清楚实际程序是如何编写的。这样就可以有效地保护用于编写程序的方法和算法。

所以，如果不希望**MEL**脚本的用户确切地知道脚本的工作原理，最好将此功能编写为一个**C++**插件，并且谨慎使用附加脚本。值得注意的是，此时**C++**插件仍可调用**MEL**命令。

2

Maya基本概念

地基决定着建筑物的牢固程度。这一原理同样适用于Maya编程。熟练掌握Maya的基本概念将使你在开发工作中受益匪浅。由于要在Maya系统的框架内工作，所以理解该系统的工作原理非常重要。任何系统都会对什么能做什么不能做加一些限制，虽然Maya在支持你尽情发挥方面无疑是极为灵活的，但仍要学习和遵守一些基本规则。

有些读者可能会迫不及待地跳过这一章而开始实际编程。我提醒你最好不要这样做。我曾见过不少多年使用Maya进行开发的人，虽然他们也能完成工作，但却忽视了或未掌握Maya的一些基本概念。其结果是，他们开发的应用程序并未完全利用Maya的全部优势。在很多情况下，这些开发人员基本是将他们的方案硬塞进Maya框架中。不幸的是，这样做注定会遭到失败。学习完本章的内容后，开发人员可以掌握Maya的一些主要规则，从而开发出更好和运行更快的应用程序。

产生以上这种情况的原因，还在于Maya允许你“按错误的方式行事”。由于Maya框架极为灵活，它很少对你实现某项任务的方式进行严格限制。这一点极具吸引力，因为几乎任何问题都可以从多个不同的角度来解决。但这种灵活性也具有一定的缺点。有时候，误入歧途的开发者会创建一些实际上不适于这一框架的应用程序。这通常会导致他们花费很多时间来对不工作或运行混乱的脚本和插件进行调试和测试。在学习本书的过程中，首先要掌握必需的基本知识。

本章提供了大量材料来概要介绍Maya的内部工作机制。这些材料同样适用于动画制作者和程序开发人员。虽然本章不涉及具体编程，但可以帮助你更好地了解Maya系统，从而在以后编写出更好的程序。后面关于MEL和C++的两章将详细介绍如何进行Maya编程。

2.1 Maya体系结构

如果在使用Maya之前你曾经使用过其他3D软件包，就必须完成一次心理上的转换，从习惯于使用其他软件包转到使用有些陌生的Maya。虽然大多数3D软件包都是要完成大体相同的任务，但它们采用的方式却是大不相同。为了完成特定操作而必须执行的步骤可能是给

定软件包所特有的，从而使你的工作流程受到某些预定义结构的约束。幸运的是，Maya的工作流程非常灵活，不会受到很多的约束。解决特定问题一般都会有多种方法，这意味着你可以采用最能适合自己需要的方法。在后面将学习到如何创建自己的自定义工作流程。

2.1.1 概述

如图2.1所示，可将整个Maya系统拆分为几个主要组成部分。

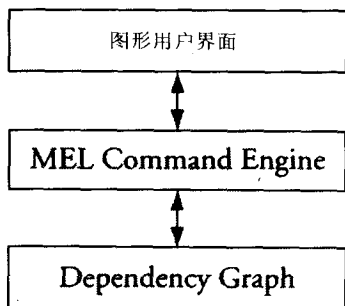


图2.1 Maya系统

作为Maya用户，你将与Maya GUI进行交互，可以选择菜单项、更改参数、使对象动画化以及移动对象，等等。当与用户界面进行交互时，Maya实际是在发出MEL命令。这些命令被发送到Command Engine，并在这里被解释并执行。你可能知道，Maya也能够以批处理模式运行。在这种模式中，不存在GUI（图形用户界面），所以MEL命令将直接发送到Command Engine并执行。

大部分MEL命令运行于Dependency Graph。这样做是因为Dependency Graph可以被直观地看做是一个完整的场景，这个场景是组成3D世界的所有重要数据和信息，其中包括对象、动画、动力学、材质等。Dependency Graph不仅定义了当前场景中有什么数据，它的结构和布局还定义了数据的处理方式。Dependency Graph实际就像Maya的心脏和大脑。鉴于其重要性，我们这里将更为详细地对它进行研究。

2.2 Dependency Graph

2.2.1 3D应用程序设计

为了更好地理解Maya体系结构与其他3D应用程序的不同之处，我们需要着重探讨一下最“典型”的3D应用程序是如何设计的。理解Maya编程方式与众不同的原因，肯定会有助于你设计自己的插件。

以前的设计方式

一般情况下，3D应用程序会提供一组特定的工具来执行建模、动画、光影和渲染等任