

<<Maya总动员>>

图书基本信息

书名：<<Maya总动员>>

13位ISBN编号：9787302241782

10位ISBN编号：7302241783

出版时间：2011-1

出版时间：清华大学

作者：张荣宝，吕新欣 编著

页数：602

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Maya总动员>>

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总动员>>

内容概要

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。

<<Maya总动员>>

作者简介

总动员公司创始人，Maya专家。

国内第一代三维动画技术的研究者与教学者，曾担任过多部电影、电视剧的特技导演，有丰富的行业技术经验。

曾任多家知名学府及培训机构的技术总监和教学总监，具有丰富的三维动画教学经验。

著有多部动画技术专著，出版有《Maya总动员》系列大型多媒体教学软件。

吕新欣

Houdini专家，Maya专家。

从事三维动画工作十多年，曾供职于多家知名动画制作企业，任职技术总监，并在多家高校及培训机构任教，是一位技术全面而精湛的三维动画专家。

<< Maya总动员 >>

书籍目录

第1章 表达式 1.1 动画表达式 1.1.1 动画表达式 1.1.2 创建动画表达式 1.1.3 每个属性只有一个驱动程序 1.1.4 时间和帧关键词 1.1.5 找到之前创建的动画表达式 1.1.6 编辑动画表达式中的文本 1.1.7 使用文本编辑器编辑动画表达式 1.1.8 删除动画表达式 1.2 粒子表达式 1.2.1 关于粒子表达式 1.2.2 粒子表达式创建表达式的执行 1.2.4 运行时间表达式的执行 1.2.5 设置动态开始帧 1.2.6 设置原始状态使用的属性 1.2.7 编写创建表达式 1.2.8 编写运行时间表达式 1.2.9 运行粒子属性 1.2.10 添加动态属性 1.2.11 了粒子和每对象属性 1.2.12 了解原始状态属性 1.2.13 赋值自定义属性 1.2.14 赋值不同长度的粒子排列属性 1.2.15 在运行表达式中使用创建表达式的数值 1.2.16 运行位置、速度和加速度 1.2.17 运行颜色 1.2.18 运行发散的粒子 1.2.19 运行碰撞 1.2.20 运行使用期 1.2.21 运行特定粒子 1.2.22 赋值矢量和矢量排列

常用函数 1.3.1 极限函数 1.3.2 指数函数 1.3.3 三角函数 1.3.4 矢量函数 1.3.5 转化函数 1.3.6 数组函数 1.3.7 随机数函数 1.3.8 曲线函数 1.3.9 通用命令 1.4 表达式编辑器 第2章 MEL脚本 2.1 背景情况 2.1.1 MEL脚本概述 2.1.2 MEL和表达式手册 2.1.3 编程器的MEL 2.2 运行MEL 2.2.1 运行MEL命令 2.2.2 查看或记录与作相关的MEL命令 2.2.3 为脚本制作一个工具栏按钮 2.2.4 从MEL命令上得到帮助 2.3 数值和变量 2.3.1 整数和浮点数 2.3.2 字符串 2.3.3 显式输入和隐式输入 2.3.4 变量 2.3.5 分配数值到变量和属性上 2.3.6 输出值 2.3.7 选择任意数字 2.4 排列、矢量和矩阵 2.4.1 排列 2.4.2 获取和改变排列的大小 2.4.3 矢量 2.4.4 矩阵 2.5 类型 2.5.1 MEL合适类型 2.5.2 使用空白区域 2.5.3 添加注释 2.5.4 变量命名 2.5.5 程序和脚本 2.6 控制脚本流量 2.6.1 测试和比较数值 2.6.2 布尔值 2.6.3 if...else if...else 2.6.4 ? : 运算符 2.6.5 switch...case 2.6.6 while 2.6.7 do...while 2.6.8 for 2.6.9 for-in 2.6.10 break 2.6.11 continue 2.6.12 测试命令、对象和属性的存在 2.6.13 =和==的不同 2.6.14 常见问题 2.7 属性 2.7.1 属性概述 2.7.2 属性名称 2.7.3 属性的数据类型 2.7.4 获取和设置属性 2.7.5 获取和设置多值属性 2.8 程序 2.8.1 程序 2.8.2 程序 2.8.3 调用程序 2.8.4 全局和本地变量 2.8.5 测试功能在MEL中是否可用 2.8.6 检查程序的来源 2.9 用户交互 2.9.1 用户交互 2.9.2 读取和编写文件 2.9.3 测试文件存在、权限和其他属性 2.9.4 控制文件 2.9.5 目录 2.9.6 执行系统命令 2.9.7 读取并写入系统命令管线 2.9.8 从AppleScript中调用MEL, 反之亦然 2.10 调试、优化和疑难解答 2.10.1 MEL调试特点 2.10.2 优化脚本和表达式速度 2.10.3 疑难解答 2.11 创建界面 2.11.1 ELF命令 2.11.2 窗口 2.11.3 控件 2.11.4 布局 2.11.5 分组 2.11.6 菜单 2.11.7 集合 2.11.8 母体 2.11.9 命名 2.11.10 UI命令模板 2.11.11 删除UI元素 2.11.12 连接命令到UI元素 2.11.13 简单窗口 2.11.14 模式对话框 2.11.15 使用系统事件和scriptJobs 2.12 脚本节点 2.12.1 MEL脚本节点 2.12.2 生成或编辑脚本节点 2.12.3 打开文件时阻止脚本节点的运行 2.13 高级版本 2.13.1 高级程序课题 2.13.2 高级动画表达式课题 2.14 句法 2.14.1 祈使句法 2.14.2 分隔符和空白区域 2.14.3 表达式、运算符和语句 2.14.4 运算符优先级 2.14.5 组块 2.14.6 注释 2.14.7 表达式和MEL句法之间的不同 2.15 疑难问题解答 2.15.1 任务 2.15.2 与句法 2.15.3 建模 2.15.4 动画、动力学和渲染 2.16 脚本范例 2.16.1 从Maya自带的脚本文件中学习 2.16.2 从文档文件中读取动画参数值 2.16.3 粒子碰撞边界 2.16.4 点爆炸 2.16.5 测试附加粒子属性 2.16.6 测试力学事件 2.16.7 动态时间播放 2.16.8 寻找着色对象 第3章 Python 3.1 Python in Maya 3.2 Python的使用 3.3 Python和线程 3.4 Python from an external interpreter 3.5 MEL与Python中的重要差异 3.6 当前限制条件 3.7 Tips and tricks for scripters new to Python 第4章 程序编译和导出端口 4.1 程序编译 4.1.1 程序编译简介 4.1.2 FBX文件程序编译 4.1.3 IGES程序编译插件 4.1.4 StudioImport插件程序和程序编译 4.1.5 导出插件程序 4.1.6 Maya OpenFlight导入端口/导出端口 4.2 游戏 4.2.1 简介 4.2.2 VRML2程序编译 4.2.3 RTG应用和文件格式 4.2.4 游戏交流2程序编译 4.2.5 MDt API 第5章 Maya API简介 5.1 简介 5.2 安装插件程序 5.3 加载插件程序 5.4 编写简单的插件程序 5.5 重要的插件程序特性 5.6 与Maya组合 5.7 MObject 5.8 包装器 5.9 对象和函数集 5.10 命名惯例 5.11 添加参数 5.12 错误检查 5.13 MStatus类 5.14 错误记录 第6章 使用选择 6.1 使用API选择概述 6.2 MGlobal::setActive-SelectionList() 6.3 MSelectionList 6.4 MItSelectionList 6.5 setObject()方法 6.6 MFn::Type列举 6.7 MGlobal::selectByName() 第7章 命令插件 7.1 添加命令给Maya概述 7.2 注册命令 7.3 MFnPlugin 7.4 initializePlugin() 7.5 uninitializePlugin() 7.6 创建者方法 7.7 MPxCommand 7.8 返回结果到MEL 7.9 语法对象 7.10 标志 7.11 背景 7.12 MPxContext 7.13 MPxContextCommand 7.14 工具属性表 7.15 MPxToolCommand 7.16 连接插件到Maya菜单 7.17 第三方插件程序的安装 第8章 DAG层次 8.1 DAG层次概述 8.2 节点 8.3 变形和形状 8.4 DAG路径 8.5 一般化

<<Maya总动员>>

化 8.6 多个形状的变形 8.7 DAG行走实例 第9章 编写阴影节点 9.1 阴影节点插件程序概述 9.2 编写阴影节点插件程序 9.3 阴影节点插件程序分析 9.4 InterNode实例编码走查 9.5 阴影节点分类 9.6 Hypershade的阴影节点图标 9.7 特殊的阴影节点 9.8 阴影节点里的超级采样 9.9 赋值渲染背景外的阴影节点 第10章 从图表插件程序 10.1 从属图表插件程序概述 10.2 母代类别描述 10.3 基础 10.4 从属图表 (DG) 节点 10.5 节点 10.6 属性和塞子 10.7 复杂属性 10.8 资料区 10.9 数据句柄 10.10 数据创建式 10.11 计算方法 10.12 复杂的实例 10.13 MPxNode及其衍生类别 第11章 操纵器和形 11.1 操纵器 11.1.1 操纵器是什么 11.1.2 基础操纵器 11.1.3 编写操纵器 11.1.4 操纵器容器 11.1.5 操纵器和节点间的信息交流 11.1.6 连接操纵器到ShowManipulator Tool 11.2 形状 11.2.1 Maya形状 11.2.2 形状分类 11.2.3 写出形状 11.2.4 绘图与刷新 11.2.5 选择 11.2.6 组件 11.2.7 调整和内部属性 11.2.8 几何体数据 11.2.9 文件IO 11.2.10 变形器 11.2.11 记录菜单 11.2.12 形状实例 第12章 编写节点和文件编译程序 12.1 编写硬件着色节点 12.1.1 硬件着色节点插件程序概述 12.1.2 执行硬件着色节点插件 12.1.3 一个硬件着色节点插件的实例 12.1.4 改进硬件着色器的运行 12.1.5 硬件着色节点插件实例 12.2 编写自定义变形节点 12.2.1 自定义变形概述 12.2.2 执行自定义变形 12.2.3 自定义变形实例 12.2.4 自定义变形节点实例 12.3 编写变形器节点 12.3.1 变形器概述 12.3.2 变形器节点 12.3.3 变形器节点实例 12.3.4 变形器实例 12.4 编写文件编译程序 12.4.1 文件编译程序概述 12.4.2 执行文件编译程序 12.4.3 文件编译程序实例 第13章 多线程插件 13.1 简介 13.2 数据和任务 13.3 线程 13.4 锁定 13.5 OpenMP 13.6 Intel线程建构模块 (TBB) 13.7 线程任务 13.8 内存分配 13.9 编程和Maya 13.10 其他对应类型 13.11 蟒蛇线程和Maya 13.12 第三方工具 13.13 API类别和插件 13.14 MacOS X平台特定线程库问题 13.15 供应商特定OpenMP问题 第14章 多边形API和MayaPython API 14.1 多边形API 14.1.1 多边形API概述 14.1.2 如何内部控制多边形 14.1.3 五个基本多边形API类别 14.1.4 构建历史和调整 14.1.5 polyModifierCmd实例 14.1.6 splitUVCmd实例 14.1.7 多边导出端口插件 14.2 Maya Python API 14.2.1 Maya Python API简介 14.2.2 开发工具 14.2.3 使用Maya Python API 14.2.4 C++ Maya API和Maya Python API的差别 第15章 建立创建环境 15.1 创建插件和应用程序简介 15.2 编译程序的要求 15.3 Linux环境 (64位) 15.4 Linux编译程序需要 15.5 使用调试器调试插件 15.6 Windows环境 (32位和64位) 15.7 安装Maya Plug-in Wizard 15.7.1 Maya Plug-in Wizard2.0 (32位的版本) 15.7.2 Maya Plug-in Wizard2.0 (64位的版本) 15.8 Mac OS X环境 第16章 分配Maya插件 16.1 分配单独文件 16.2 分配多个文件的模块 16.3 模块限制 16.3.1 共享库 16.3.2 渲染全能的着色器 第17章 实例插件 17.1 实例插件概述 17.2 独立应用程序 17.3 C++插件实例描述 17.4 独立应用程序的描述举例 17.5 着色器源代码举例 17.6 ASHLI着色器脚本插件程序举例 第18章 附录 18.1 附录A: NURBS图形 18.2 附录B: 相关图渲染节点 18.3 附录C: 渲染属性 18.4 附录D: 常见问题解答 18.5 附录E: API和Devkit限制条件 18.6 附录F: 添加图像插件程序

<<Maya总动员>>

章节摘录

插图：重新计算在没有暗淡输入的第一个节点上停止。

例如，如果更改曲线旋转度数，但不更改曲面本身，然后重建旋转的曲面会使旋转节点重新计算，但不影响曲线。

这是高级介绍，实际执行提供大量信息，这样可以避免不必要的计算。

流经图形的数据是模拟水流经管道。

管道本身是连接的，但除非有重新定向和修改的数据，否则实际上不进行操作。

扩展这个类比，节点像脉冲线、花洒、喷泉、WC、源泉和水签。

它们都是以自己独特的方式用水进行某个操作，但必须有水。

使用DG的有意思的副作用是很难直接影响对象。

例如，细想上图中的球体。

如果没有连接到Transform3的DG节点影响球体缩放，通过UI或API设置的数值要重新缩放。

但是，如果图形中Time影响球体的缩放，Transform3的缩放通过UI或API进行另外修改时，效果不同。

如果设置缩放，可能直到第二次重新计算从属图表时，才会覆盖Time设置的缩放。

Time再次设置球体缩放时，通过UI或API设置的数值会丢失。

更复杂的实例应该是旋转的曲面。

如果试着在生成曲面上移动CV，会发生什么？

只有DG重新计算时，CV才会移动到新位置，这时CV会移动回到旋转节点强制命令的位置。

<<Maya总动员>>

编辑推荐

《Maya总动员:动画编程篇》：超值20小时高清视频教程，素材及工程文件。

<<Maya总动员>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>