

/// 构建个性化网络虚拟世界——

21世纪网络虚拟世界入门与提高丛书之二

吴小华 主编 李竞杰 赵鹏 编著

VRML 与Java 编程



国防工业出版社

前言



Internet 及 WWW 给我们的生活带来了巨大的变化，它逐渐成为联系全世界信息的纽带，而在此基础上的虚拟空间的实现，将为我们带来一场革命。

VRML (Virtual Reality Modeling Language) ——虚拟现实建模语言正是实现这一革命的中流砥柱，VRML 有着巨大的使用前景，它可以实现细腻的三维虚拟场景，大至星球，山脉，湖泊，小至房间，建筑，个人物品等等。并且该虚拟空间的娱乐性，可控性，交互性都能得到充分的实现。市面上几乎所有的浏览器，如 Internet Explorer, Netscape 都能通过插件来支持 VRML。VRML 的另一重要特点就是代码精练、占用空间很小、编译快，作为一种描述性语言，它使用性与可视性高度统一。

Java 是一种面向网络的程序设计语言，它使人们对网络有了一个崭新的认识。Java applet 可以达到良好的通用性，在目前所有的操作系统下都能稳定地运行。当你浏览一个嵌有 applet 的网页时，服务器将以一种与机器无关的方式发送到本地机，然后解释执行。用户无需担心其通用性。

本书主要向读者介绍了怎样将这二者结合使用，并如何在互联网上创建三维的虚拟空间。

本书的第 1~6 章介绍了 VRML 的基础知识，即使读者从未接触过 VRML，通过这几章的学习，也可以很轻松地创建出复杂的三维场景。而对于熟练的使用者而言，这一部分也可以作为备查资料，不可不用。

本书的第 7 章讲述了传感器和插入器的使用，由于我们在后面的实例程序中大量地用到了这两种节点，因此编者建议读者详细阅读这两章的内容。

本书的第 8 章介绍了脚本的使用，使用脚本和使用 Java 程序的方法和作用有许多相似之处，如果读者熟练掌握了脚本的使用方法，后面的学习会相当轻松。在很多情况下，脚本可以与 Java 程序结合使用，以取长补短。

本书的第 9 章和第 10 章主要是面向没有 Java 编程基础的读者，如果读者已经掌握 Java 的编程方法可以跳过这两章阅读，当然也可以作为语法查询。

第 11 章介绍了通过 Java 程序实现特效和交互功能的方法，在这一章中，编者尽可能地与脚本对照说明，以帮助读者理解。

第 12 章介绍了动态创建场景的方法，通过这一章的学习，读者将掌握通过 Java 程序来即时生成 VRML 三维场景的方法。

在第 13 章中，编者将向读者介绍 Java 在 VRML 场景中的一些高级应用技巧，这一章着重介绍怎样利用 Java 本身提供的强大功能来增强自己的三维场景。

本书的第 14 章介绍了将制作好的三维网页发布到互联网上的操作步骤，以及一些需要注意的问题。

参加本书编写的还有王洁、陈果，在本书撰写的过程中，得到了李晓龙的大力协助，本书的所有程序均在 Windows 2000 环境下编译通过。由于 Java 和 VRML 的平台无关性，这些例子适用于绝大部分的操作系统。因此本书尽可能地提供完整的例子，读者可以尝试在自己的计算机上实现网络虚拟世界。

本书中的程序代码可以在国防工业出版社计算机室网站（<http://www.NDIP.com.cn/computer/>）上免费获得。

编 者



目 录

第 1 章 VRML 概述	1
1.1 VRML 的诞生、现状及前景	1
1.2 VRML 虚拟现实实例	2
1.3 VRML 2.0 的功能和应用	4
1.4 VRML 常用浏览器及使用	5
1.4.1 World View 2.0 的浏览方式控制	6
1.4.2 World View 2.0 的浏览视点控制	7
1.4.3 World View 2.0 的弹出菜单	7
第 2 章 VRML 基本概念	9
2.1 VRML 约定及基本概念	9
2.1.1 节点和域	9
2.1.2 事件、路由和原型	10
2.1.3 场景、场景结构图、交互及脚本	11
2.2 VRML 空间计量单位	11
2.2.1 VRML 空间坐标系	11
2.2.2 长度单位	11
2.2.3 角度单位	12
2.3 VRML 文件基本要素	12
2.3.1 文件头	12
2.3.2 节点	12
2.4 VRML 的节点与域	13
2.4.1 外形节点 Shape	14
2.4.2 组节点 Group	14
2.4.3 造型节点	15
2.4.4 父节点和子节点	15
2.5 创建第一个 VRML 场景	15
第 3 章 基本造型及其属性	19
3.1 提供三维物体对象 (Shape 节点)	19
3.2 创建基本几何造型	21
3.2.1 长方体造型 (Box 节点)	21
3.2.2 圆锥体造型 (Cone 节点)	22
3.2.3 圆柱体造型 (Cylinder 节点)	24

3.2.4	球体造型 (Sphere 节点)	26
3.2.5	组合造型	27
3.3	创建文本造型	30
第 4 章	虚拟环境的设置	40
4.1	光源的使用	40
4.1.1	头顶灯的开关	40
4.1.2	点光源 PointLight	42
4.1.3	平行光源 DirectionalLight	46
4.1.4	聚光光源 SpotLight	49
4.2	环境背景的设置	53
4.3	如何控制视点	57
4.4	信息节点的使用	67
第 5 章	造型分组	70
5.1	Group 节点	70
5.2	Transform 节点	74
第 6 章	复杂造型的结构	87
6.1	用于复杂造型的几何体属性节点	87
6.1.1	坐标参数节点 Coordinate	87
6.1.2	颜色参数节点 Color	88
6.2	基于顶点的几何造型	89
6.2.1	PointSet 节点	89
6.2.2	IndexedLineSet 节点	91
6.2.3	IndexedFaceSet 节点	95
6.3	创建标高网格	111
6.4	创建挤压造型	120
第 7 章	VRML 中的动画与交互	124
7.1	VRML 中的事件和路由	124
7.1.1	事件和路由	124
7.1.2	VRML 中的事件操作	124
7.1.3	时间传感器	126
7.1.4	完整的例子	127
7.2	VRML 中的插入器	130
7.2.1	ColorInterpolator (色彩插入器)	130
7.2.2	ScalarInterpolator (梯度插入器)	132
7.2.3	CoordinateInterpolator (坐标插入器)	134
7.2.4	OrientationInterpolator (方向插入器)	137
7.3	VRML 中的传感器	140
7.3.1	CylinderSensor (圆柱体传感器)	140
7.3.2	PlaneSensor (平移传感器)	142
7.3.3	SphereSensor (球面传感器)	143

7.3.4	TouchSensor (接触传感器)	145
7.3.5	VisibilitySensor (视觉传感器)	147
7.3.6	ProximitySensor (接近传感器)	149
第8章	VRML 脚本	154
8.1	脚本的概念	154
8.2	VRMLScript 语法规则	155
8.2.1	脚本语言的结构和基础	155
8.2.2	VRMLScript 常用语句	160
8.2.3	对象与函数	165
8.3	VRML 中的事件处理	174
8.3.1	事件输入	174
8.3.2	事件输出	175
8.4	Script 节点	176
8.4.1	Script 节点语法	176
8.4.2	Script 节点应用举例	177
8.5	扩展用户交互功能	179
8.5.1	单事件处理	179
8.5.2	多事件处理	181
第9章	Java 语法	188
9.1	Java 简介	188
9.1.1	Java 简史	188
9.1.2	Java 的现状	189
9.2	Java 程序的结构	193
9.2.1	声明一个类	193
9.2.2	注释	194
9.2.3	声明一个方法	194
9.3	数据类型	194
9.3.1	变量	194
9.3.2	变量的声明	194
9.3.3	常量	195
9.3.4	变量的命名约定	196
9.3.5	变量的作用域	197
9.4	数组	199
9.4.1	数组的创建	199
9.4.2	数组的访问	200
9.4.3	多维数组	201
9.5	字符串	201
9.5.1	字符串的定义	201
9.5.2	字符串的连接	202
9.5.3	取子串	202

9.5.4	编辑	202
9.5.6	比较	203
9.6	类型转换和数值提升	204
9.6.1	转换	204
9.6.2	转换上下文	206
9.7	运算符和表达式	208
9.7.1	赋值运算符	208
9.7.2	数学运算符	209
9.7.3	其它运算符	212
9.7.4	运算符的优先级、结合方向	213
9.8	语句和块	213
9.8.1	条件语句	214
9.8.2	分支语句	215
9.8.3	循环语句	216
9.8.4	标号和转移语句	218
第 10 章	Java 进阶	220
10.1	面向对象的程序设计	220
10.1.1	对象与类	220
10.1.2	属性与方法	220
10.1.3	继承	224
10.1.4	接口和包	225
10.2	异常处理	228
10.2.1	调用堆栈	228
10.2.2	抛出异常	229
10.2.3	捕获异常	231
10.3	JDK	233
10.3.1	JDK 的安装	233
10.3.2	JDK 的使用	236
10.4	JavaApplet	237
10.4.1	JavaApplet 的安全性	238
10.4.2	将 JavaApplet 程序嵌入 html 文件	238
10.4.3	关于 JavaApplet 使用的建议	240
第 11 章	通过 Java 实现特效	241
11.1	运动	241
11.2	翻转	245
11.3	颜色选择	248
11.4	残像	254
第 12 章	通过 Java 动态创建 VRML 场景	262
12.1	删除和添加路由	262
12.2	动态添加物体	269

第 13 章	Java 在 VRML 场景中的高级应用	274
13.1	输出字符串	274
13.2	弹出窗口	278
13.3	控制窗口	282
第 14 章	网上发布	294
14.1	网上发布的注意事项	294
14.2	申请主页空间	295
14.2.1	到哪里申请	295
14.2.2	申请过程	295
14.3	主页的上传	296
14.3.1	文件传输协议	296
14.3.2	主页上传后无法显示的原因	301
14.4	加入搜索引擎	301
第 15 章	VRML 开发资源	303
15.1	VRML 浏览器	303
15.2	VRML 开发工具	306

第1章

VRML 概述



长久以来，人们就希望有一种可以真实再现现实生活中各种场景和事件的方法，由此促成了虚拟现实（VR）的出现。随着近年来网络的高速发展，尤其是 Internet 在全世界的影响力不断扩大，如何在网上实现这种引人入胜的虚拟技术，成了许多人关心的事情。而 VRML（Virtual Reality Modeling Language）正是一种基于 WWW（World Wide Web）上的具有交互性的虚拟现实建模语言。

1.1 VRML 的诞生、现状及前景

作为一个虚拟现实系统，它必须具有几个最基本的要素：

首先它必须能够建立一个三维立体的虚拟空间和空间中构造虚拟造型。在以往的二维图片中，它们所表达的只是从一个确切位置和方向以及某一个时间上看到的现实空间和物体的信息，其信息是十分有限的，我们是否想过换一个角度看物体的情况呢？在二维图片中是不能提供这种自由的，它最多只能多提供物体的一些侧面图片给我们而已。

第二，这个虚拟现实系统可以让我们在虚拟空间中自由地活动，也就是说让我们有一种好像在现实生活中的大街上漫步一样的感觉。

第三，在这个虚拟空间中，可以通过一些控制装置来实时操作和改变我们所在的虚拟空间。就像我们在生活中能通过按下灯的开关将灯打开或者关闭一样。

为了实现具有这样几个基本要素的虚拟现实系统，很早就有人提出了各种构想，但是由于各方面的制约，像缺乏技术支持、没有合适的传播载体等等，虚拟现实的产生一直拖到了 20 世纪末。1991 年万维网技术投入应用后，万维网迅速膨胀成为今天最有活力的商业热点。1992 年，SGI 公司又推出了新一代三维计算机图形接口“Open Inventor”，此时虚拟现实产生的条件都已具备。1994 年 5 月，在日内瓦举行的第一届因特网会议上，Mark Pesce 和 Tony Parisi 在对以虚拟现实界面访问万维网进行了长时间的研究以后，把他们的思想介绍给正在筹划中的 BOF（Birds-of-a-Feather）的一个关于虚拟现实的会议，并获得大力支持。在此次会议上大家一致决定要制定一种能连接万维网的三维场景描述语言，“VRML”这一



术语就是在这次会议上定下的。

VRML 的中文名叫做虚拟现实建模语言。它用来建立真实世界中各种场景的模型,当然,它也可以建立一个与真实世界无关的虚拟世界,就像许多游戏中的一样。

VRML 的第一个标准版本是 VRML 1.0,它是经过许多人共同合作的结果,其中最主要的是 SGI 公司的工作。SGI 公司同意无偿提供这种新的文件格式并放弃对其的所有权,在 1994 年 10 月的万维网会议上,VRML 1.0 规范正式出台。

VRML 1.0 在设计时采用了面向对象的思想,因此由其构造的虚拟现实模型与所用的操作平台无关,并且其程序还具有可扩展性。VRML 1.0 不具有交互功能,也没有声音和动画,它只建立了一个可以用来探索的环境,除此之外,没有其他的功能。对于一个功能如此有限的语言,在网络上是很难被推广应用的,这也促使了新标准的产生。

VRML 的第二个标准为 VRML 2.0,它是在 VRML 1.0 的基础上经过补充和完善而建立起来的,其标准于 1996 年 8 月通过。

新的 VRML 标准不仅仅是对 VRML 1.0 中节点的补充,而是在很多方面作了改进,使 VRML 有了巨大的进步。它主要改进的方面包括:增强了静态世界的构造功能;增加了交互功能;增加了动画功能;增加了编程的能力;增加了原型的定义功能。

其中编程功能的增加,使 VRML 更符合“面向对象”编程的思想。交互功能、动画功能的增加,使我们不仅能够浏览虚拟空间,而且还能够改变虚拟空间,使虚拟空间更加接近现实空间的真实性。

1.2 VRML 虚拟现实实例

虚拟三维空间的文本描述是通过 VRML 文件,它可以通过任何文本编辑器来进行编写,但在保存时要将其保存为后缀名是 .wrl 的文件。常用的浏览器,像 Microsoft 的 Internet Explorer 4.0, Netscape 的 Communicator 4 等都可以通过自身集成的 VRML 浏览插件,就可以直接浏览 VRML 网页,1.4 节将详细介绍如何用这两种最常用的浏览器来浏览用 VRML 创建的虚拟三维世界。下面通过“HelloWorld.wrl”VRML 文件来初步了解 VRML。打开一个文本编辑器输入以下文本:

例 1-1:

```
#VRML V2.0 utf8
```

```
Group
```

```
{
```

```
  children[
```

```
    #环境背景
```

```
      Background{
```

```
        skyColor[
```

```
          1.0 0.5 0.8,
```

```
          0.1 0.1 1.0,
```

```
          1.0 1.0 2.0
```

```
        ]
```

```
    ]
```

2



```

skyAngle [1.5,1.0]
groundColor[
    3.0 0.1 0.1,
    0.4 0.3 0.2,
    0.6 0.6 0.6,
]
groundAngle [1.50,1.50]
},

```

#输入文字

```

Shape{
    appearance Appearance{
        material Material{
            diffuseColor 2.5 1.0 0.0
        }
    }
    geometry Text{
        string "Hello World!"
        fontStyle FontStyle{
            size 3
        }
    }
},

```

#创建造型

```

Transform{
    translation -5.0 1.5 0.0
    children[
        Shape
        {
            appearance DEF Green Appearance
            {
                material Material{ diffuseColor 0 1.0 0.8 }
            }
            geometry Cone
            {
                height 6
                bottomRadius 2.0
            }
        },
        Transform
    {

```

```

translation 0.0 -1.0 0.0
scale      1.0  0.3 1.0
children Shape
{
    appearance USE Green
    geometry Sphere
    {
        radius 4.0
    }
}
}
}
}
}
}
}
}
}
}

```

将上述文本输入完毕后，保存为“Hello World.wrl”。

下面通过用 Internet Explorer 5.0 来浏览刚才输入的代码。Internet Explorer 5.0 是通过自带插件 World View 2.0 来浏览 VRML 文件的。在 Internet Explorer 5.0 中打开 VRML 文件时将文件类型设置为“*.wrl”即可，也可直接双击“Hello World.wrl”文件将其打开。

在 Internet Explorer 5.0 中打开的“Hello World.wrl”如图 1.1 所示。

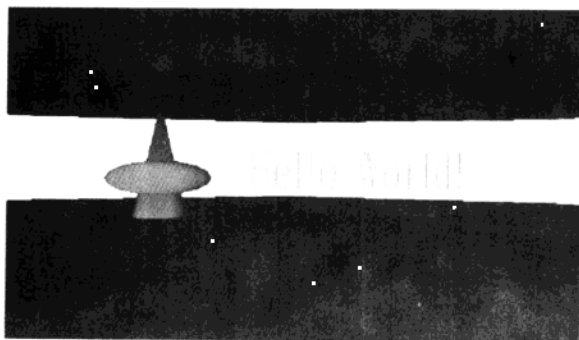


图 1.1 Hello World!示例

1.3 VRML 2.0 的功能和应用

VRML 1.0 允许开发者建立三维的静态场景，并且场景中的对象能超链接到别的场景，这和 HTML 文档非常相似。VRML 2.0 除继承 VRML 1.0 的造型和超链接功能外，还扩充了大量的内容。用 VRML 2.0，场景中的对象可以移动，能对基于时间的以及用户激发的时间产生响应。总体看来，新增加的内容可以分为以下几个方面：



1. 感应控制器功能

例如, 我们按下键盘按键, 在 VRML 程序中, 必须要有 `Sensor{}` 节点, 才能知道使用者有行为发生, 进而依据 `Script` 节点中的设定, 产生预定的反应。

2. Script 脚本功能 ()

在 `Script{}` 节点中加入程序语言, 以进行对象行为的设定, 编写的语言倾向于使用 Java, 也可用其他的 CGI 程序, 例如, Perl。另外 SGI 公司则发展了一种类似 Java 的 VRML Script 语言, 前途可观。

3. 立体音效

在 `Sound{}` 节点中, 可以进行声音大小、位置、方向等空间性质的设定, 让声音的表现有远近不同的效果, 以增加真实性。

4. 交互性

新增加的对三维对象的行为描述使得进行实时完全仿真成为可能。碰撞检测 (collision detection) 节点包含了碰撞检测信息, 这样就避免了穿墙而过的现象发生; 感知器 (sensors) 节点等待一个特定事件的发生, 然后做出响应。例如, 可以设计一个感知器完成当门被打开时就自动开灯的功能。

5. 增强的静态场景

VRML 2.0 中一些新增的特征使得场景看上去更加真实。例如, 声音 (sound) 节点允许为场景加上背景音乐、动作声响、对话等等; 背景 (background) 节点允许为场景加一幅背景图画; 雾 (fog) 节点能产生逼真的雾化大气效果。

6. 多重使用者

使虚拟环境成为让数个使用者通过同时彼此互动与沟通的空间, 比笔谈的 E-Mail、对话的 IRC、甚至是未能普及的视讯会议更上一级, 其中最受瞩目的画为 Living Worlds (LW)。Living Worlds 画 (http://www.livingworlds.com/draft_1/index.htm) 为 VRML 2.0 标准制定后的另一个 VRML 开发团队整合, 由 Black Sun、ParaGraph 与 Sony 公司共同提出, 并且得到 30 多个大厂的支持, 所要达成的理想是建立一个多重使用者共享的虚拟环境开发标准, 可以让进入的使用者利用其替身 (Avatars), 在虚拟空间中彼此交谈或沟通, 并体验到多媒体, 甚至是置身其中的生理感觉。

1.4 VRML 常用浏览器及使用

现在支持 VRML 浏览器软件种类很多, 常用的像 Netscape 公司的 Netscape, Microsoft 公司的 Internet Explorer; 其他的例如 Paper Software 公司的 WebFX, SGI 和 Template Graphics Software 公司的 Web Space, InterVista 软件公司的 World View 以及 Microsoft 公司的 Virtual Explorer 等等也是经常用到的。

下面将介绍在用 Netscape 公司的 Netscape 或者 Microsoft 公司的 Internet Explorer 时插

件的配置问题。

如果我们使用的浏览器是 Netscape 4.0 以下的版本, 则可从网上下载 Cosmo Player(<http://cosmo.sgi.com>); 若用的是 Netscape 4.0 或更高版本, 则浏览器已内置 CosmoPlayer 2.0, 但安装 Netscape 时要注意选中相应的选项; 如果使用的浏览器是 Internet Explorer 5.0, 则可能已经内置了 VRML 2.0 浏览器, 判断是否内置的方法很简单, 看它能否打开 VRML 文件(*.wrl, *.wrz)即可。如果无法打开, 可以从 <http://www.microsoft.com/vrml/> 下载 VRML 浏览器插件, 对于 Internet Explorer 5.x 以下的版本, 还需要我们下载一些辅助插件。当然在开始之前应基本熟悉 VRML 浏览器的操作方法。

如果说 VRML 是一种语言, 那么 VRML 浏览器就是它的解释器。的确, VRML 浏览器的主要功能是读入 VRML 代码文件, 并把它解释成一个图形映像。

VRML 浏览器基本上实现了物体的变换效果, 如, 灯光、视角变换、模糊、裁剪、阴影、投影及碰撞。

下面将介绍如何用 Microsoft 的 Internet Explorer 5.0 浏览器来浏览 VRML 网页。

在 Internet Explorer 5.0 浏览窗口的控制栏可以分为两部分, 一部分是窗口左侧的浏览方式控制栏, 另一部分是窗口底部的浏览视点控制栏, 还有弹出菜单控制。

1.4.1 World View 2.0 的浏览方式控制

World View 2.0 的浏览方式控制分为移动模式和观察模式两种。而且处在不同的浏览方式下鼠标的显示方式也各不相同。

1. 移动模式

移动模式可分为 Walk、Pan、Turn 和 Roll 四种浏览方式。

(1) **Walk** 的浏览方式是模拟在虚拟三维世界中的走动和转向等。其实现方法是在控制栏中选择 Walk 选项使之高亮度显示, 然后在浏览器窗口内按住鼠标左键并拖动鼠标或通过键盘上的方向键来进行控制。其快捷键为 Ctrl+Shift+W。

(2) **Pan** 的浏览方式是模拟在虚拟三维世界中的垂直面上的上下或者左右方向的平移。其实现方法是在控制栏中选择 Pan 选项使之高亮度显示, 然后在浏览器窗口中按住鼠标左键并拖动鼠标或通过键盘上的方向键来进行控制。其快捷键为 Ctrl+Shift+P。

(3) **Turn** 的浏览方式为用来改变浏览者的视野角度。其实现方式是在控制栏中选择 Turn 选项使之高亮度显示, 然后在浏览器窗口中按住鼠标左键并拖动鼠标或通过键盘上的方向键组合来进行控制。其快捷键为 Ctrl+Shift+T。

(4) **Roll** 的浏览方式是在虚拟三维世界中的一个垂直平面上实现顺时针或者逆时针方向的视角变化。其实现方法是在控制栏中选择 Roll 选项并使之高亮度显示, 然后在浏览器窗口中按住鼠标左键并向左或者向右拖动鼠标, 视角便沿顺时针或者逆时针方向旋转, 也可通过键盘上的左右方向键来进行控制。其快捷键为 Ctrl+Shift+R。

2. 观察模式

观察模式可分为 Goto 和 Study 两种浏览方式。



(1)  Goto 的浏览方式是走近所要观察的对象。其实现方法是在控制栏中选择 Goto 选项并使之高亮度显示,然后在浏览器窗口中用鼠标左键单击所要观察的对象就可以靠近观察了。其快捷键为 Ctrl+Shift+G。

(2)  Study 的浏览方式是以不同的方向和不同的旋转角度来旋转几个漂浮在空间中的造型,对造型的某些细节进一步仔细观察。其实现方法是在控制栏中选择 Study 选项并使之高亮度显示,然后在浏览器窗口中按住鼠标左键并拖动鼠标或通过键盘单个方向键或者组合键来进行控制。其快捷键为 Ctrl+Shift+S。

1.4.2 World View 2.0 的浏览视点控制

VRML 中的视点就是浏览者在三维空间中的观察位置,视点的设置是在 VRML 文件中完成的。World View 2.0 共有 4 种视点控制方式: View、Zoom Out、Straighten Up 和 Re-store。View 方式主要用于切换观察者的视点,后三者主要是用来帮助浏览者快速地从不熟悉的环境或者错误的观察角度中摆脱出来,恢复到正常的视野方向和观察角度。

(1)  View 视点控制:如果在 VRML 文件中设置了多个视点,那么单击 View 两侧的箭头按钮就可切换视点为当前视点的前一个视点或者后一个视点。如果单击 View 则返回当前视点,而如果右击 View 视点将弹出一个包括所有视点的右键快捷菜单,选择其中之一即可实现视点之间的快速切换。切换到下一个视点的快捷键为 PageDown,切换到上一个视点的快捷键为 PageUp。

(2)  Zoom Out 视点控制:Zoom Out 控制是用来自动切换到可以浏览整个虚拟世界的视点的。当浏览者在一个并不怎么熟悉的虚拟世界中迷失了方向时,就可以用到 Zoom Out,浏览器将自动切换到一个远景的视点上,在此视点上,浏览者就可以看到整个虚拟三维世界。其快捷键为 Ctrl+Shift+Z。

(3)  Straighten Up 视点控制: Straighten Up 控制可以使浏览者的视线快速转移到正前方,并且使其与水平面平行,给浏览者重新配置一个合乎现实的视角。尤其是当浏览者在使用了 Study 或者 Turn 之后,通常会处在一个特别古怪的视角上, Straighten Up 能够使浏览者快速地恢复到正确的视角上。其快捷键为 Ctrl+Shift+U。

(4)  Restore 视点控制: Restore 控制能使浏览者的视点切换到 VRML 文件默认的视点上,即浏览者进入虚拟世界的第一个视点。但是它与浏览器的刷新 (Refresh) 不同, Restore 并没有重载文件,它仅仅是切换了视点而已。

1.4.3 World View 2.0 的弹出菜单

在浏览器窗口中的任意位置单击鼠标右键便会弹出如图 1.2 所示的右键快捷菜单,其中各个命令详细说明如下:

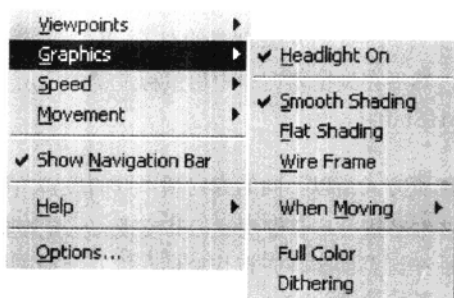


图 1.2 World View 2.0 的级联菜单

(1) Viewpoints 命令：用来给出所有可用视点的列表，可以实现视点的快速切换。

(2) Graphics 命令：用来控制显示的画质水平和顶灯的开关。画质水平从高到低依次为：Smooth Shading、Flat Shading 和 Wire Frame。浏览的速度则与画质水平相反。When Moving 命令则允许浏览者在移动时设置画质水平的高低。在移动时不要打开 Full Color，但可以打开 Dithering 设置，这样就可以使场景中色彩光滑过渡，而且能较大地改善各种场景的浏览效果。

(3) Speed 命令：用于设置浏览时的速度。要注意的是，设置的浏览速度并不是绝对速度，而是与浏览者所浏览场景的复杂程度、设置的画质水平高低以及所用计算机的硬件配置高低有关。

(4) Movement 命令：用于设置不同的浏览方式。

(5) Show Navigation Bar 命令：用来显示或者隐藏控制栏。

(6) Help 命令：用来显示帮助信息。

(7) Options 命令：选项，主要包括以下三种设置选项：

1) Downloading Textures：用于在打开 VRML 文件时是否同时下载纹理贴图。如果在打开时选择该选项，则可以大大加强场景的浏览效果。

2) Using Hardware Acceleration：是否使用硬件加速。如果所使用的计算机的显卡具有 3D 加速性能，则可以加快浏览的速度，也可以提高浏览的效果。

3) Rotating Objects Automatically：是否自动旋转所浏览的物体。如果选择该选项，在 Study 模式下，单击所浏览的物体后，物体将会自动旋转。

第2章

VRML基本概念



从本章开始，将介绍 VRML 的核心内容，而本章主要介绍 VRML 的基本概念。

VRML 涉及到的基本概念包括以下几个方面：

- (1) 节点、域、事件、路由、原型、场景和场景结构图、交互以及脚本。
- (2) VRML 空间计量单位。
- (3) VRML 文件基本要素，并创建第一个 VRML 场景介绍给读者。

下面对本章内容进行详述。

2.1 VRML 约定及基本概念

VRML 作为一门编程语言，有其自己约定的规范与标准，涉及到节点、域、事件、路由、场景等重要的基本概念，下面进行详述。

2.1.1 节点和域

节点是 VRML 文件最基本的组成要素，而 VRML 文件就是由许多节点层层嵌套而构成的。一部分节点表示对象，如，长方体 (Box)、圆柱体 (Cylinder)、圆锥体 (Cone) 等等；一部分节点则把相关节点放在一起，例如，外形节点 (shape) 包含一个几何节点和一个外观节点，而这两种节点又可以包含其他节点。此外，VRML 还定义了一些提供特殊功能的节点，如，超链接、碰撞检测节点等。

每个节点都是由域组成的，每个域都有域值，这些域值指定了一个场景的特征，每个域值都有自己的默认值。

现在举例说明一个节点所包含的基本要素：

```
Cylinder
{
  radius 2.0
  height 3.0
```

