

VRML SANWEI JIAOHUSHI CHANGJING KAIFA JISHU

VRML三维交互式 场景开发技术

 刘颖 王淑良 陈松 编著



西南交通大学出版社

VRML SANWEI JIAOHUSHI CHANGJING KAIFA JISHU

VRML三维交互式 场景开发技术



刘颖 王淑良 陈松 编著



西南交通大学出版社
· 成 都 ·

内容简介

本书以 VRML 平台的互动式编程为目标, 结合实例介绍了 VRML 编程的基本概念、方法和核心技术。VRML 互动式漫游部分, 主要介绍了在三维世界漫游中, 观察者与三维世界的互动机制。VRML 结合模型部分, 主要介绍了在第三方平台中建立的立体模型如何在 VRML 中使用。VRML 的互动控制部分, 主要介绍了在 VRML 中如何实现三维世界中的目标控制。

本书采用最新的案例驱动模式编写, 不但适合计算机专业的高年级学生和研究生选作教材, 而且适合有一定 VRML 编程基础的开发人员参考。

图书在版编目 (C I P) 数据

VRML 三维交互式场景开发技术 / 刘颖, 王淑良, 陈松编著. — 成都: 西南交通大学出版社, 2015.8

ISBN 978-7-5643-4202-9

I. ①V… II. ①刘… ②王… ③陈… III. ①VRML 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 195870 号

VRML 三维交互式场景开发技术

刘 颖 王淑良 陈 松 编著

责任编辑	张 波
特邀编辑	黄庆斌
封面设计	何东琳设计工作室
出版发行	西南交通大学出版社 (四川省成都市金牛区交大路 146 号)
发行部电话	028-87600564 028-87600533
邮政编码	610031
网 址	http://www.xnjdcbs.com
印 刷	成都勤德印务有限公司
成品尺寸	185 mm × 240 mm
印 张	24.75
字 数	540 千
版 次	2015 年 8 月第 1 版
印 次	2015 年 8 月第 1 次
书 号	ISBN 978-7-5643-4202-9
定 价	75.00 元

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前言

VRML (Virtual Reality Modeling Language), 即虚拟现实建模语言。它是一项与多媒体通信、因特网、虚拟现实等领域密切相关的第二代 WWW 标准语言。作为唯一 Web 3D 国际标准的 VRML 技术, 得到了众多企业的支持, 在网络上得到了广泛应用。与实现三维场景的技术手段 OpenGL、DirectX 相比, VRML 非常适合一般用户, 而使用 OpenGL 和 DirectX 实现时, 必须配合其他语言来实现涉及的接口; VRML 脚本化的语句可以编写三维动画片、三维游戏、计算机三维辅助教学课件, 且文件容量较小, 因此可以嵌在网页中显示, 更适合于网络传输; VRML 虚拟空间生成系统可以使用户通过拖放的方法, 人机交互生成 VRML 虚拟空间; 3D Max 对 VRML 的制作提供了更好支持, 大部分的 VRML 效果、交互都可以在 3D Max 中完成, 所以前期的开发模型、贴图等都可以在 VRML 的开发中使用; 在 VRML 中, 用户能使用 LOD、动态多分辨率分块 LOD 技术、可见性选择、单像素对应等, 不仅可以实现场景大小的无限制, 还可保证整个场景运行流畅; VRML 脚本为 Javascript 脚本, 在 VRML 场景中通过交互节点可以实现与浏览者的实时交互, VRML 提供 Java、Javascript、XML、Flash 等网络对象的支持, VRML 也支持 MPG、Flash、RealPlayer 等流媒体视频格式; VRML 场景的渲染是实时的, 完全不同于关键帧三维动画, VRML 支持雾、立体声、实时灯光渲染等特性, 可增加场景的真实感和用户的切身体验感; VRML 的开发是基于 Internet 环境的, 所以 Internet 的新技术都可以应用到 VRML 技术的开发中, VRML 支持 Java、Javascript、XML、Flash、MPG、RealPlayer、SQL、GIS 等技术, 同时也提供高级开发语言接口, 如 C++。

本书从不同的方面展示了 VRML 所涉及的技术在具体开发中的应用, 其主要分为三个方面:

(1) VRML 互动式漫游: ① 彩色山间漫游系统。重点建造海拔栅格, 并对海拔栅格上各个染色点进行染色和明暗控制; ② 海底世界。在设计中运用多种节点(如导航节点)实现了鱼自由游动的动画效果和人机交互功能; ③ 梦幻小屋。本设计通过编程建立实体模型, 采用内联的方法, 将复杂的场景和模型分解成独立的文件, 以方便修改和重构; ④ 两层展厅漫游。建立的虚拟展厅漫游系统, 包括空间模块的建立、调入纹理、调入声音、添加光源、背景等模块, 构成了整个展厅; ⑤ 虚拟医院漫游。本系统采用模块化、组件化设计思想进行开

发设计,实现了虚拟场景的碰撞检测、开关门、楼梯感应等;⑥ 太空漫游。本系统设置了行星的运行路径,路径由多个坐标组成,而每个坐标对应一个相应的动画时钟;⑦ 美丽风景虚拟场景。本系统中瀑布的设计采用粒子代替了小水滴,让粒子从上而下的运动形成了类似于实际生活中的瀑布场景。

(2) VRML 结合模型:① 虚拟校园。利用 3D Max 实现建筑物的三维建模,用 VRML 添加光源、背景、材质等场景信息以渲染场景,用 Script 节点设定对象的行为,使用时间检测技术以及邻近检测技术;② 避暑山庄。建立了避暑山庄漫游系统中风车、木屋、凉亭、邮箱等模型,再合理运用 VRML 中所提供的动画效果和交互功能,实现风车的自由旋转、鹰的自由飞翔和门的自动打开;③ 汽车模型。创建了一个汽车在环形跑道上行驶的虚拟场景,包括三维场景的建模,实现场景动画和交互控制功能。同时采用 3D Max 建好的场景模型实现了模型的集成;④ 室内场景。以虚拟现实建模语言并结合 3D Max 的方法,实现了一个室内外结合,强调人机交互的分层室内漫游系统。实现了感应门、碰撞检测、上下楼梯、键盘响应、影音播放等功能;⑤ 虚拟室内。借助 3D Max 制作部分复杂模型,实现了具有较高真实感场景的模型构建、多媒体展示、场景漫游和物体碰撞检测等;⑥ 虚拟教室。本系统在三维坐标上建立虚拟教室的三维模型,使用 Inline 节点来统一整合场景模型,有机结合高精度建模与低精度建模,利用导航、路由等节点相结合实现自动漫游、交互漫游。

(3) VRML 的互动控制:① 七彩立体拼图。此设计有两个主要功能,一个是用户可以通过鼠标点击拖动操作将拼图块移动到相应的位置的功能,另一个是用户通过按钮的拖动来完成拼图块旋转的功能。还实现了通过拖动旁边的控制板的按钮来实现从不同角度观赏立体图等功;② 飞机起飞。在本系统中,可以通过鼠标的点击来触发飞机的运行,通过调整视角来观赏飞机起飞的过程,同时根据真实场景使用背景节点、飞机造型节点等创建出飞机以及周围环境的模型在运动部分,采用了朝向、位置等各种插补器来实现飞机的运动,运用时间感应器来控制程序的运动时间周期,使用触摸感应器来对飞机运动的触发进行控制。

由作者水平有限,加之时间仓促,书中难免存在不足之处,恳请读者批评指正。

衷心感谢为本书做了大量前期准备工作的赵文俊、彭静、江轲、李海燕、徐洪飞、姚宇、姚智芳、程红梅、郭彪勇、钟昌毅、苟顺涛、黄帅、王寒、张克诚、孙东等同志。

作者

2015 年 5 月

目 录

第 1 章 关于虚拟现实语言 VRML	1
1.1 VRML 概述	1
1.2 VRML 的特点	15
1.3 VRML 中的交互	19
1.4 VRML 的空间坐标	20
第 2 章 VRML 研究和应用现状	22
2.1 VRML 研究的关键技术	22
2.2 VRML 各国研究现状	22
2.3 VRML 发展趋势	29
第 3 章 VRML 互动式漫游	31
3.1 彩色山间漫游系统	31
3.2 海底世界系统	45
3.3 梦幻小屋系统	62
3.4 两层展厅漫游系统	81
3.5 虚拟医院漫游系统	101
3.6 太空漫游	115
3.7 美丽风景虚拟场景	141
第 4 章 VRML 结合模型	162
4.1 虚拟校园漫游系统	162
4.2 虚拟避暑山庄漫游系统	188
4.3 汽车模型漫游控制系统	220
4.4 基于 VRML 的分层室内场景漫游系统	234

4.5 虚拟室内漫游	259
4.6 虚拟教室漫游	274
第 5 章 VRML 的互动控制	295
5.1 七彩立体拼图程序	295
5.2 飞机起飞介绍	310
附 录	330
附录一 VRML 常用节点和域说明	330
附录二 VRML 资源网址大全	380
参考文献	385

第1章 关于虚拟现实语言 VRML

1.1 VRML 概述

1.1.1 基本概念

VRML (Virtual Reality Modeling Language), 即虚拟现实建模语言, 它是一项与多媒体通信、因特网、虚拟现实等领域密切相关的第二代 WWW 标准语言。

虚拟现实建模语言是一种三维造型和渲染的图形描述语言, 它是描述虚拟环境中场景的一种标准语言。它与语言比较类似, 本身为码, 通过浏览器解释出代码。通过面向对象技术, 利用计算机构建起接近真实的虚拟三维场景, 建立三维物体, 渲染虚拟场景, 并通过多种知觉的传感器实现用户与虚拟现实世界的交互, 从而使用户产生身临其境的感觉。虚拟现实建模语言具有平台无关性, 它是基于三维交互网站的解释性语言。虚拟现实建模语言集成多媒体技术、计算机图形图像技术、仿真技术、传感技术等多种技术于一体, 涉及人工智能、计算机网络、多媒体、自然与社会科学、哲学等多个技术领域与学科领域。与虚拟现实技术有所区别的是, 虚拟现实技术的实现需要进入虚拟三维世界的用户借助各个交互设备(感觉手套、数据头盔等)来实现交互过程, 而虚拟现实建模语言是一种三维建模和渲染的图形描述性语言。它超越语言的地方在于: 在 VRML 之前, 网页均为二维平面的, 尽管能够通过脚本为网页添加互动效果, 让静态的网页更加生动, 但总体来说, 依然停留在平面阶段。而利用它可以在 Internet 建立交互式的三维多媒体的境界。利用 VRML 所创造出来的三维世界更加立体真实, 用户进入虚拟世界之后始终占据主导地位, 视觉呈现的景象也随着用户位置的改变而改变, 交互性、融入性更强。在安装一个 VRML 浏览器后, 用户可通过网络访问虚拟现实世界, 其操作方法简单易懂。由此可见, 虚拟现实技术并非高高在上, 已经如同一般的多媒体技术和网络技术一样渐渐被大众所接受, 成为标准和核心技术。简单地说, 以 VRML 为基础的第二代 WWW=多媒体+虚拟现实+Internet, 它的主要特征有三维性、交互性、动态性、实时性、多媒体集成、境界逼真性等。

与其他 Web 技术语言相比, VRML 语法简单、易懂, 编辑操作方便, 因此学习相对容易。VRML 可创建三维造型与场景, 也能实现很好的交互效果, 而且可以嵌入 Java、JavaScript 等程序, 使其表现能力得到极大扩充。它能够实现人机交互, 形成更为逼真的虚拟环境。

VRML 文件描述的基于时间的三维空间称为虚拟境界 (Virtual World)。它由对象构成, 而对象及其属性用节点来描述, 节点是 VRML 的基本单元。每个节点由类型、域、事件、实

现、名字组成,节点按一定规则构成场景图(Scene Graph),场景图中分两类节点:第一类节点用于从视觉、听觉角度表现对象,它们按层次体系组织,反映了境界的空间结构,提供颜色、灯光、超链接、材质、化身、重力、碰撞、地形随动、飞行等功能,支持局部坐标系;第二类节点参与事件产生和路由机制,形成路由图,能够确定境界随时间推移如何动态变化,提供环境变化、用户交互、时间推移产生事件,传感器(Sensor)检测并发出初始事件,实践产生其他事件或修改场景图结构,从而提供动态特性。插入器(Interpolator)是特殊事件处理器,利用它可以设计动画。

对于复杂行为处理则需利用脚本节点(Script node)来完成。它包含一组脚本描述语言编写的函数,Script节点收到事件后,将执行相应的函数。该函数可以通过常规的事件路由机制发送事件或直接向Script节点指定节点发送事件,脚本也能动态增加和删除路由。

1.1.2 VRML 中较为深入的几个概念

1. 构建 VRML 文件的几种方法

构建 VRML 文件一般有三种方法:

① 使用文本编辑器直接进行 VRML 文件的书写。这样做的不足之处是只能创建比较简单的物体和场景。用户若要创建复杂场景,几乎不可能使用文本编辑器来直接编写,有三个原因:一是因为场景之间相互关系和位置确定比较复杂;二是对于复杂物体的建模显得无能为力(精确到小数位两位已经是计算的一个极限了);三是对数学能力要求较高。

② 一般的方法是使用可视化编辑器来建模。使用可视化编辑器避免了直接使用文本建模时的一些问题,例如不能生成复杂场景等。这些工具的一个突出优点就是使用起来相当方便,它不需要手工地输入大量的命令,而只需要轻轻拖动鼠标即可实现大量程序才可能完成的功能。

常见的几个可视化编辑器有:Cosmo World 2.0、Paragraph's Virtual Home Space Builder、Home Space、Caligari's Pioneer、Virtus Walk Through Pro 和 3-D Website Builder 等。

③ 第三种方法就是使用其他常见的图形格式进行转换。在 Web 上,存在着大量的各种各样的三维文件格式,转换工具的作用就是在这些文件格式之间进行转换。需要注意的是,所有的转换程序都有局限性,例如在转换特定文件中的光照和纹理贴图时就会失败。

InterChange 是 Syndesis 公司(<http://www.threedee.com>)的一项产品,它可以执行大约 50 种文件格式之间的转换,VRML 就是这 50 种文件格式中的一种。

Extreme 3-D 是 Macromedia 公司(<http://www.macromedia.com>)的一项功能非常强大的产品,它对于 VRML 文件的转换功能在多媒体创作人员当中是深受欢迎的。使用 Extreme 3-D,用户就可以通过点击鼠标来把二维模型转换成三维模型。

DoomtoVRML2 使用户可以在网上发布自己的 Doom 文件,它使得用户可以把 Doom 层次的多边形和纹理信息转换成为 VRML 2.0 格式的文件(<http://vrm1.sig.com/tools/doomtovrml2/index.html>)。

VRML1to2 可以将符合 VRML 1.0 格式的文件转换成为符合 VRML 2.0 格式的文件, Sony 公司 (<http://vrml.sgi.com/cgi-bin/vrmlToVrml2.cgi>) 在这个领域开了先河。

VRML1toVRML2 来自 SGI 公司, 它执行的功能与上面的转换器一样。

2. 浏览 VRML 的方法及几种常见的 VRML 插件

浏览 VRML 的一般方法是使用 VRML 浏览器, 也可以在 IE 等通用浏览器上通过 VRML 浏览器插件对 VRML 文件进行浏览。

3. VRML 中的动画效果

要理解 VRML 中动画的概念, 首先必须理解路由和事件。为了使虚拟空间具有动感, 构造指令可以包含绑定指令, 绑定指令描述了如何将节点绑定在一起。VRML 绑定包括: 绑定在一起的节点和节点之间绑定的路由 (或者叫做路径)。绑定两个节点之后, 第一个节点通过这样的路径传送给第二个节点的信息叫做事件。事件包含一个值。当一个节点接收到一个事件时, 它将根据节点的特征开始播放动画或者进行其他操作。通过绑定多个节点, 用户可以创建许多路由, 从而使空间更具有动感。

大多数 VRML 节点都可以绑定在线路之上, 每个节点都有输入、输出插座。一些节点同时具有输入、输出插座, 而另外的一些节点仅有其中的一种。节点的输入插座称为 eventIn, 输出插座称为 eventOut。当链接一个路由时, eventIn 接受输入, eventOut 将事件输出。节点的每个输入、输出路由也有类型。例如, 一个 SFFloat 类型的 eventOut, 当它绑定一个路由时, 输出浮点数。SFFloat 类型的 eventIn 能够接收浮点值。创建路由之后, 线路路由将处于睡眠状态, 直到有一个事件从发送节点发送到接收节点。接收节点接收事件之后将作出反应, 反应类型依赖于:

- ① 接收事件节点的类型。
- ② 路由所绑定的节点输入插座。
- ③ 事件所包含的数值。
- ④ 当前节点的活动状态。

VRML 中的动画产生是由于变动了任何一个坐标系的位置、方向和形体比例, 从而使物体按你所想的方式飞行、平移、旋转或按比例缩放。

TimeSensor 节点的作用像一个时钟, 它可以被用来执行开始、停止或者其他控制动画的动作。随着时间的流逝, 这个传感器就会产生事件来表示时间的变化。通过将这些事件 TimeSensor 节点的 eventOut 路由到其他节点。当 TimeSensor 节点的时钟计时时, 可以使这些节点发生相应的变化。如果要使一个坐标系平移、旋转和按比例缩放的话, 可以将 TimeSensor 节点事件路由至 PositionInterpolator 和 OrientationInterpolator 节点。这些节点中的每一个产生新的位置和旋转值, 并通过它们的 eventOut 传送这些值。按顺序将这些值路由到 Transform 节点, 就可以使节点的坐标系随动画过程的发展而发生平移、旋转和按比例缩放。

4. VRML 中的自身交互

如果要使空间具有交互性,可以给一个造型附带一个传感器,该传感器使用一个定点设备来感知观察者的动作。当观察者点击到一个附带有传感器的造型时,传感器就输出一个事件,这个事件就被路由到其他的节点来开始一个动画。

TouchSensor 是一种用来检测观察者的接触和将事件输出的传感器。这些输出描述了在何时、何地观察者接触到了可感知的造型。CylinderSensor、PlaneSensor 和 SphereSensor 节点也可用来检测在何时观察者接触到了一个可感知的造型,并且提供了用来改变造型位置和方向的输出。

在 VRML 中,用户可以将一个传感器附在一个造型上,用它来检测观察者的移动、点击和拖动。当观察者与一个可感知的造型相互作用时,就可以把传感器的输出连入一个线路中,从而引起造型的移动和动画的播放。

感知观察者接近常使用三种方法:感知观察者的可视性、感知观察者的接近性和通过碰撞检测。

可见传感器可从观察者的位置和方向来感知在空间中的一个长方体区域是否可视。用户可以通过这些传感器来启动和停止动画或者控制其他的动作,这些动作仅当一个可感知的区域可见时是必要的。通过给出中心和尺寸,可以指定一个由 VisibilitySensor 节点感测的空间区域。

碰撞检测是检测空间中观察者与造型接近和碰撞的时间。碰撞节点在检测观察者的碰撞时做两件事:

- ① 通过 CollideTime eventOut 事件输出当前的绝对时间。
- ② 提示浏览器。

以上就是 VRML 场景中的几种交互,使用这些交互我们可以做到类似电灯开启,自动滑动门等效果。但是面对更为复杂的交互使用,它本身所提供的传感器就显得无能为力了,这就需要利用 Script、Proto 节点等对其进行功能上的扩展,最好是利用 Java 技术来完成这些扩展工作,因为 Sun 公司对 VRML 技术也非常看好。

1.1.3 发展历史

VRML 本质上是一种面向 Web、面向对象的三维造型语言,而且它是一种解释性语言。VRML 对象称为节点,子节点的集合可以构成复杂的景物。节点可以通过实例得到复用,对它们赋以名字,进行定义后,即可建立动态的 VR(虚拟世界)。

VRML 1.0 标准是许多研究人员共同合作的结果。1993 年 9 月, Tony Parisi 和 Mark Pesce 开发了第一个 VRML 浏览器。在第一届 WWW 大会上(1994 年秋于日内瓦),由 Tim Berners-Lee 和 Dave Raggett 所组织的一个名为 BOF 的小组提出了 VRML(Virtual Reality Markup Language),只是在后来由于为了反映三维世界的建立而将 Markup 改为了 Modeling,

缩写仍为 VRML。在这次大会后,一个名 www-vrmlmail list 的组织成立了,并于 1994 年秋在第二次 WWW 大会上发布了 VRML1.0 的草稿。VRML1.0 允许单个用户使用非交互功能,且没有声音和动画,它只允许建立一个可以探索的环境,但没有别的。虽然 VRML1.0 给人的最初印象看起来十分有限,但它的确形成了一组开发者可以用的工作核心,以便建立 VRML2.0、VRML3.0 或更新的版本。VRML2.0 的规范于 1996 年 8 月通过,它在 VRML1.0 的基础上进行了很大的补充和完善。它以 SGI 公司的 Move World 提案为基础,业界范围内对于 VRML2.0 的支持非常大。许多重要的厂商明确表示,VRML2.0 将是它们产品结构的基础。

SGI 公司,最有影响力的 VRML 厂商,已经引进 Cosmo3D——一个 VRML2.0 的 API,作为其新的工具结构 Viper 的基础。Cosmo3D 的附件支持为 SGI 公司已有工具而编写的应用程序,而这些应用程序也就变成了 Viper 内置的专用功能。SGI 公司许多广为传播的工具也支持 VRML2.0。

Sun Microsystems 公司是 Java 的先驱,最近选择 VRML2.0 作为其基于 Java 的 3DAPI——Java3D 的功能内核。Microsoft 公司已经采纳了 VRML2.0 标准,并已利用 DimensionX 基于 Java 的 Liquid Reality 来生成用于 VRML2.0 浏览的 ActiveX 组件。可在 WindowsNT4.0 上找到 Microsoft 公司优化了的 OpenGL 1.1,而 Microsoft 的 Direct3D 已通过硬件加速显示了它强大无比的功能。

IBM 公司和 Apple 公司正在与 ParaGraph 公司合作,开发以图形压缩为特点的 VRML 二进制格式,这项工作不仅可以使文件大小减少 50%,还可以把 VRML 使用者的范围拓宽。

VRML 已经发展成为新的国际标准 X3D,它在 VRML 的基础上做了很多改动。X3D 是一种支持 XML 编码格式的开放式 3D 标准,3D 数据可以通过网络实现实时交流,具有可移植性、页面整合性,易于与下一代的网络技术整合。另外,它采用了组件化结构设计,因此减少了对系统资源的占用,且具有很强的扩展性

1.1.4 VRML 的运行

1. 运行环境

VRML 对硬件与软件的环境要求都较低,一般配置了 Pentium2、Windows 98 操作系统的计算机都能运行,但在硬件配置较高时,其运行速度较快。一般推荐配置如下:

① 硬件环境:建议采用 Pentium4 以上的计算机,主频 2 G 以上,内存 128 M 以上,显存 64 M 以上,硬盘 10 G 以上。

② 软件环境:操作系统可采用 Windows 98/2000/XP/2003 等,但要求安装 VRML 的相关浏览插件。

③ 网络环境:浏览 VRML 场景时可采用拨号、宽带等网络接入方式,网络浏览器可采用 Windows 操作系统自带的 IE 浏览器或 Netscape 浏览器等。

2. 编辑运行过程

VRML 的三维空间造型是由 VRML 文件产生的,VRML 文件是一个用 VRML 语言编写

的文本文件，其扩展名是.wrl。一般的文本编辑器（记事本等）都可以用来编辑 VRML 文件，不过我们推荐使用由 ParallerGraphics 公司开发的专用编辑器 VrmIpad 来编写 VRML 文件。VrmIpad 为 VRML 文件的编辑和运行调试提供了很多其他文本编辑器所不具备的方便之处。

VRML 文件需要安装专用的浏览器才能运行，目前常用的浏览器有微软公司开发的 VRML 浏览器、SGI 公司开发的 CosmoPlayer 浏览器等，还有其他一些开发商的浏览器。其中 CortonaPlayer 和 CosmoPlayer 浏览器具有相对较好的性能，在操作系统中可以同时安装这两种浏览器，它们的作用有所不同。CortonaPlayer 在 VrmIpad 编辑器中用于文件的预览调试，而 CosmoPlayer 浏览器安装后用于在 IE 浏览器中观察 VRML 的运行效果。当然也可以只装一种浏览器，建议安装 CortonaPlayer 浏览器。以上所说 VrmIpad 编辑器和浏览器可以从网上下载。

3. 浏览方法

浏览 VRML 的一般方法是使用 VRML 浏览器，也可以在 IE 等通用浏览器上通过 VRML 浏览器插件对 VRML 文件进行浏览。较常用的插件有 BS-Contact、blaxxun、Cortona 等。

1.1.5 工作原理

VRML 的工作原理是首先用文本信息描述三维场景，通过在 Internet 网上传输，再在本地机上由 VRML 的浏览器解释生成三维场景，解释生成的标准规范即是 VRML 规范。正是基于 VRML 的这种工作机制，才使其可能在网络应用中有很快的发展。当初 VRML 的设计者们考虑的也正是文本描述的信息在网络上的传输比图形文件迅速，考虑避开在网络上直接传输图形文件而改用传输图形文件的文本描述信息，将复杂的处理任务交给本地机，从而减轻了网路负荷。

1.1.6 VRML 的主要内容

1. VRML 文件格式及 MIME 类型

VRML 文件可以包括下列主要成分：VRML 文件头、造型和事件、脚本、路由、原型。并不是所有的文件都包括这些要素，但必须包括 VRML 文件头。

VRML 2.0 标准的文件头为“#VRML V2.0 utf8”。这不同于 1.0 标准中的文件头，1.0 标准中文件只支持 ASCII 字符集。为了向下兼容，即 VRML 2.0 文件格式兼容 1.0 文件格式，这个文件头有三个含义：表明这个文件是一个 VRML 文件；符合 VRML 2.0 版本；文件使用的是 utf8 字符集。

VRML 注释允许在不影响 VRML 空间外观情况下，在 VRML 中包括其他信息。可以加入对文件内容、文件绘制的不同部分的注释。注释以#符号开始，结束于该行的最后。

VRML 中的场景由造型组成，而造型则由节点创建，这些是 VRML 的构件。单个节点描述造型、颜色、光照、视点以及造型，动画定时器、传感器、内插器等的定位和朝向等等。

节点可以有零个或多个域, VRML 定义了 20 多种基本数据类型, 主要有 SF 的单值类型及 MF 的多值类型。节点除了域属性外, 还具有事件属性, 事件是 VRML 实现用户交互及场景动态变化的最主要内容。概括说来, 节点由域和事件组成, 其中域的取值决定了节点的取值, 从而决定了当前虚拟环境的状态; 事件则为节点提供了接收外界信息以及向外界发送信息的能力。

MIME (Multipurpose Internet Mail Extensions), 即多用途 Internet 邮件扩展。它用来定义 Internet 上传送的文件内容类型的软件标准, 所有的 Web 浏览器都能够理解 MIME 所定义的文件类型, 并使用它们在浏览器上自动决定显示的信息内容。例如: 如果一个 MIME 类型显示它的文件中包含了 VRML 文本, 那么浏览器将格式化 VRML 文本以使其显示在浏览器的窗口中。一个 MIME 内容类型由用斜杠分开的两部分指定: 第一部分说明内容的一般类型, 如文本、音频、视频类型; 第二部分说明内容的子类型, 用于指定内容使用的确切格式。

MIME 内容类型由 Internet 协会标准化, 临时的或最新的 MIME 内容类型都以 x-开始 (x 表示扩展)。VRML 非常新, 以至于它现在使用下列扩展的 MIME 内容类型: x-world/x-vrml。将来, VRML 的内容类型将融合进正式的 MIME 标准, 并成为下列 MIME 内容类型: model/vrml。

2. 域

域可以包含各种类型的数据和单个或更多的值。每个域都有缺省值。域的标识符叫域名。在节点的作用范围内, 域名是唯一的。有两类常见的域: 一类只包含单值。所谓单值, 可以是一个单独的数, 也可以是定义一个向量或颜色的几个数, 甚至可以是定义一副图像的一组数, 单值类型的域名以“SF”开始; 另一类包含多个值, 叫多值域, 多值域名称以“MF”开始。

使用节点和域时要注意其书写方式, 特别要注意大小写, VRML 对大小写的区分是非常严格的。

3. 程序注释

同其他语言一样, VRML 也允许在不影响 VRML 文件所创建的 VRML 场景的前提下, 在 VRML 文件中包含其他信息。可以在 VRML 文件的任一部分添加注释信息。VRML 文件中的程序注释部分以符号 # 开头, 结束于该行的行尾, 也就是说它不支持多行注释, 新一行的注释也必须以 # 开头。

4. VRML 语言的骨干——路由

路由的作用在于将各个不通的节点绑定在一起以使虚拟空间具有动感和交互性。大多数的 vrml 节点都有输入、输出的接口, 其中输入接口称为 eventIn, 输出接口称为 eventOut。一个节点通常具有多个不同输入接口和输出接口, 但有一些节点并不同时具有输入和输出接口。

若在两个节点之间存在着路由, 事件就可以通过路由由这个节点传递到另外一个节点上。这样传递的事件通常可以改变相应节点的某些域值。例如, 在虚拟世界中分别创建了一盏灯

和一个开关,通过合适的路由将两者绑定后,可以通过鼠标单击开关来控制灯的亮灭,此时通过路由传输的事件就是灯的外观控制。

5. VRML 节点

节点是对现实世界中各种对象和概念(如球、光和质材等)的抽象描述,它是 VRML 文件中最基本的组成部分,其作用是描述空间造型及其属性。由多个节点之间并列使用或层层嵌套就构成了 VRML 文件。

使用 VRML 语言编写的文档称为 VRML 文档(或 VRML 程序),其扩展名为.wrl。VRML 文档的格式有两种: utf8 文本格式和二进制格式,目前我们所用的都是文本格式。每个 VRML 文档是一个 Web3D 页面,VRML 虚拟世界由一个或多个 VRML 文档共同展现。不同的 VRML 文档之间通过超链接组织在一起,共同构成了遍布全球的非线性超媒体系统。

节点(Node)是 VRML 文档基本的组成单元,用于描述对象某一方面的特征,如形状、材质等。VRML 虚拟世界的对象往往由一组具有一定层次结构关系的节点来构造。节点具有节点名、节点类型、域、事件接口和实现五个组成部分。节点可以用 DEF 语句命名,用 USE 语句引用。

1) VRML 中的节点分类。

节点类型可分为基本类型和用户自定义类型两大类。基本类型由系统提供,自定义类型由用户在基本类型的基础上通过原型机制构造,它们都是对虚拟世界的某些共性的提炼。如: Appearance 节点描述实体的外观; Material 节点描述对象的材质等,域、事件接口和实现通常包含在节点类型的定义中,并在构造节点时被使用。域描述节点的静态固有属性,域可以有多个,每个域都必须具有一个域值;事件接口描述节点的动态交互属性,它提供节点与外界的通信接口,事件接口也可以有多个。接口类型有事件入口和事件出口两种:事件入口(eventIn)是节点的逻辑接收器,它负责监听和接收外界事件;事件出口(eventOut)是节点的逻辑发送器,它负责向外界发送节点产生的事件。实现则是域和事件接口机制的实现代码,它一般由系统提供。需要提醒注意的是,事件具有事件值和时间戳属性。事件值即该控制信息本身,时间戳则表示事件发生或传送的时间。每个域(或事件)都具有一个数据类型,用于描述数据的结构、值域、个数等属性。数据类型按照包含的数据个数分为单值型和多值型两类。单值型以 SF 打头,只能包含一个数据;多值型以 MF 打头,可包含一个数据列表。

VRML 中的节点有以下几种:

- 造型尺寸、外观节点: Shape、Appearance、Material。
- 原始几何造型节点: Box、Cone、Cylinder、Sphere。
- 造型编组节点: Group、Switch、Billboard。
- 文本造型节点: Text、FrontStyle。
- 造型定位、旋转、缩放节点: Transform。

- 内插器节点: TimeSensor、PositionInterpolator、OrientationInterpolator、ColorInterpolator、ScalarInterpolator、CoordinateInterpolator。
- 感知节点: TouchSensor、CylinderSensor、PlaneSensor、SphereSensor、VisibilitySensor、ProximitySensor、Collision。
- 点、线、面集节点: PointSet、IndexedLineSet、IndexedFaceSet、Coordinate。
- 海拔节点: ElevationGrid。
- 挤出节点: Extrusion。
- 颜色、纹理、明暗节点: Color、ImageTexture、PixelTexture、MovieTexture、Normal。
- 控制光源的节点: PointLight、DirectionalLight、SpotLight。
- 背景节点: Background。
- 声音节点: AudioClip、MovieTexture、Sound。
- 细节控制节点: LOD。
- 雾节点: Fog。
- 空间信息节点: WorldInfo。
- 锚点节点: Anchor。
- 脚本节点: Script。
- 控制视点的节点: Viewpoint、NavigationInfo。
- 用于创建新节点类型的节点: PROTO、EXTERNPROTO、IS。

2) 关键节点介绍。

(1) VRML 动画的发源地: TimeSensor 节点。

```
TimeSensor {
  cycleInterval 1      # 暴露域值 类型 STime
  enabled TRUE         # 暴露域值 类型 SBool
  loop FALSE           # 暴露域值 类型 SBool
  startTime 0           # 暴露域值 类型 STime
  stopTime 0           # 暴露域值 类型 STime
}
```

① 域。

cycleInterval: 每个周期的长度, 以秒为单位, 取大于 0 的值。

enabled: 若为 TRUE, 表示当条件成立时产生时间相关事件; 若为 FALSE, 表示在任何条件下都不会产生时间相关事件。无论 enabled 为 TRUE 或 FALSE, 通域的 set_事件都被处理并产生_changed 事件。

loop: 表明时间传感器是无限循环, 还是在一个周期后被终止。

startTime: 开始产生事件的时间。

stopTime: 终止产生事件的时间。其值若小于等于起始时间, 则被忽略。

② 事件。

isActive eventOut SBool: 表明时间传感器当前是否在运行。若在运行, 则返回 TRUE; 若处于停止状态, 则返回 FALSE。

cycleTime eventOut STime: 在每个周期开始时, 返回当前时间。

fraction_changed eventOut SFloat: 当前周期的完成比。从 0(周期开始)到 1(周期结束)。

time eventOut STime: 从格林威治时间 1970 年 1 月 1 日子时(午夜 12 点)至今所经过的秒数。

③ 说明。

随着时间的推移, TimeSensor 节点产生事件。它被用来生成动画(通常与插值器一起), 能在一个指定时间引发一个动作或者以固定时间间隔产生事件。

通常情况下, 时间传感器的 **startTime** 被另一个对用户动作作串反应的传感器或脚本传来的时间事件设置。在 **startTime** 设定的时刻以前, 时间传感器不做任何工作。在该时刻到来时, 它产生值为 TRUE 的 **isActive** 事件, 并开始产生 **time**、**fraction_changed** 和 **cycleTime** 事件。

time 事件是总保持有一个当前时刻值的连续发生的事件。而产生的其他与时间相关的事件是周期性发生的事件。当 **loop** 为 FALSE 时, 时间传感器仅仅运行 **cycleInterval** 中设置的一个时间周期(或在第一个周期完成前到达 **stopTime** 中设定的时间); 而当 **loop** 为 TRUE 时, 时间传感器不断地运行, 直到到达 **stopTime** 中设定的时间或 **enabled** 被设置为 FALSE。在每个周期的开始, 时间传感器发出 **fraction_changed** 事件(值为 0)和 **cycleTime** 事件(值为当前时间)。在每一个周期中, 当浏览器允许传感器产生一个事件时(通常每帧一次), **fraction_changed** 的值从 0 增加到 1, 表明当前周期已完成多少。在 **fraction_changed** 值为 1 时, 当前周期结束, 新的周期开始。

cycleTime 事件仅仅在周期开始时产生。若要生成一个一次性事件(如闹钟), 则应把 **start_Time** 设置为期望时间, 把 **loop** 设置为 FALSE。这样在 **startTime** 到来时, 该时间传感器只运行一个周期。

在产生时间相关事件时, 时间传感器忽略 **set_cycleInterval** 和 **set_startTime** 事件。**cycleInterval** 和 **startTime** 域的值将不会发生变化, 而且不会产生 **cycleInterval_changed** 和 **startTime_changed** 事件。若想要重启一个时间传感器, 应该首先发送一个与当前时间值相同的 **set_stopTime** 事件来停止当前的时间传感器, 再发送一个 **set_startTime** 事件。

当一个活动的时间传感器收到一个 **set_stopTime** 事件时, 若值大于 **startTime** 的值但小于当前的时间值, 则它将把 **stopTime** 设定为指定值。但像 **stopTime** 中设置的时间是当前时间那样, 发出结束事件。若 **set_stopTime** 事件值小于等于 **startTime** 的值, 则这一事件将被忽略。

设置 **loop** 为 TRUE 时, 时间传感器在 **startTime** 到来时开始产生事件, 事件将不断地产