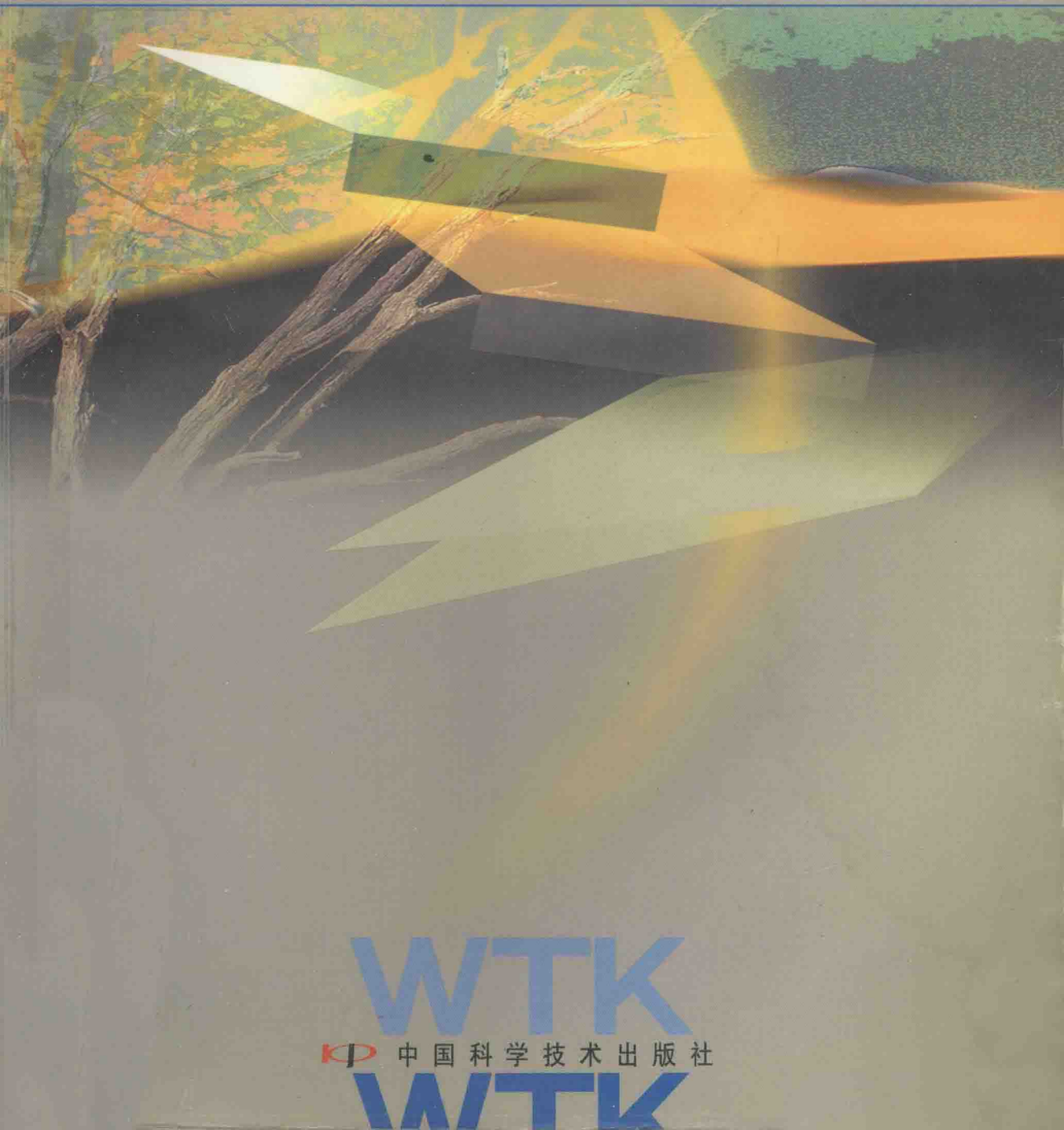


虚拟仿真开发平台 WTK

王永超 胡月明 谢健文 陈飞香 编著



WTK
中国科学技术出版社
WTK

虚拟仿真开发平台WTK

王永超 胡月明 谢健文 陈飞香 编著
何

中国科学技术出版社
·北 京·

图书在版编目(CIP)数据

虚拟仿真开发平台 WTK/王永超等编著. —北京:中国科学技术出版社, 2004. 10

ISBN 7-5046-3931-1

I. 虚... II. 王... III. 计算机仿真-程序设计
IV. TP391.9

中国版本图书馆 CIP 数据核字(2004)第 111234 号

中国科学技术出版社出版
北京市海淀区中关村南大街 16 号 邮政编码:100081
电话:010-62103210 传真:010-62183872
科学普及出版社发行部发行
北京长宁印刷有限公司印刷

*

开本:787 毫米×1092 毫米 1/16 印张:18.125 字数:450 千字
2004 年 11 月第 1 版 2004 年 11 月第 1 次印刷
定价:36.00 元

本书由

**广东技术师范学院出版基金
华南农业大学地理信息系统研究室**

资助出版

责任编辑：吕秀齐
封面设计：赵一东
责任校对：刘红岩
责任印制：安利平

目 录

第 1 章 WTK 概述	1
1.1 WTK 简介	1
1.2 两个重要概念	1
1.2.1 场景结构	1
1.2.2 仿真引擎	1
1.3 WTK 的主要特点	2
1.3.1 代码的高效性	2
1.3.2 跨平台性	2
1.3.3 支持多种外设	3
1.3.4 支持多管道/多处理器	3
1.3.5 其他特点	3
1.4 WTK 的子类(class)总览	4
1.5 WTK 的系统配置	5
1.5.1 通常的系统配置	5
1.5.2 Windows 下的系统配置	5
1.5.3 Windows 下支持的外部设备	6
1.6 WTK 的安装	6
1.6.1 WTK 的安装	6
1.6.2 环境变量的设置	8
1.7 WTK 应用程序示例	9
第 2 章 虚拟环境的管理	10
2.1 从建模坐标系到屏幕坐标系	10
2.1.1 绘制过程中的坐标系	10
2.1.2 摄像机标定	11
2.2 WTK 中的宇宙	14
2.2.1 宇宙的概念、构造和析构	14
2.2.2 仿真管理	16
2.2.3 宇宙行为函数	18
2.2.4 宇宙中的实体	19
2.2.5 整体渲染函数	20
2.2.6 性能函数和定时函数	22
2.2.7 宇宙选项设置函数	22
2.2.8 资源文件	24
2.3 WTK 中的视点	27
2.3.1 视点的基本管理	28

2.3.2	传感器与视点的关联	29
2.3.3	视点位姿的存取	30
2.3.4	坐标系的使用和坐标转换	32
2.3.5	立体显示	33
2.4	WTK 中的窗口	36
2.4.1	窗口的构造和集成	37
2.4.2	窗口大小位置以及与场景的关联	40
2.4.3	窗口和视点	41
2.4.4	窗口投影函数	42
2.4.5	多边形拾取和光线投射(Ray Casting)	44
2.4.6	窗口绘制属性	45
2.4.7	视口	47
第 3 章	场景图及其结点	49
3.1	虚拟场景结构和场景图	49
3.1.1	虚拟场景结构和场景图	49
3.1.2	场景图的层次关系	50
3.1.3	结点分类	50
3.1.4	WTK 对场景图的绘制	51
3.2	位姿结点	52
3.2.1	位姿结点的基本概念	52
3.2.2	位姿结点相关函数	53
3.2.3	状态积累和状态传播	54
3.2.4	场景中的坐标系	55
3.3	描述位姿的数据类型	56
3.3.1	平面位置矢量	57
3.3.2	空间位置矢量 $WTp3$	57
3.3.3	四元数 WTq	58
3.3.4	坐标系 $WTpq$	60
3.3.5	旋转矩阵 $WTm3$	60
3.3.6	位姿矩阵 $WTm4$	61
3.3.7	姿态描述方法的相互转换	61
3.4	普通分离结点	63
3.5	LOD 结点和开关结点	65
3.6	停靠结点、内联结点和烟雾结点	67
3.6.1	停靠结点和内联结点	67
3.6.2	烟雾结点	68
3.7	OpenGL 回调结点	69
3.8	可移动结点	72
3.8.1	可移动结点的概念	72

3.8.2 可移动结点的构造函数	73
3.8.3 可移动结点的位姿控制函数	74
3.8.4 可移动结点的层次结构	74
3.8.5 可移动结点的层次结构的相关函数	75
3.8.6 实例化和结点路径	76
3.8.7 实例化和结点路径的相关函数	77
3.9 场景图的构建及相关坐标系	79
3.9.1 场景图的构建和合成位姿	79
3.9.2 场景装配函数	80
3.9.3 外部文件函数	81
3.10 结点属性函数和场景查询函数	82
3.10.1 结点属性函数	82
3.10.2 场景查询函数	84
第4章 光照材质声音和纹理建模	85
4.1 光源	85
4.1.1 WTK 中的光源	85
4.1.2 光源结点构造函数	86
4.1.3 光源结点属性函数	88
4.2 材质	91
4.2.1 材质属性和像素颜色的计算	91
4.2.2 材质表及其文件格式	93
4.2.3 材质表函数	94
4.3 声音建模	97
4.3.1 支持硬件	98
4.3.2 设备级函数	98
4.3.3 声音级函数	101
4.4 纹理建模	103
4.4.1 纹理的应用	104
4.4.2 透明纹理	107
4.4.3 明确指定 uv 进行纹理映射	108
4.4.4 动画纹理	110
4.4.5 纹理过滤	112
4.4.6 纹理的操作	113
4.4.7 其他操作	115
第5章 几何建模	117
5.1 多边形建模	117
5.1.1 多边形属性	117
5.1.2 多边形的操作	119
5.1.3 多边形的动态构造	120

5.1.4	多边形相交检测	122
5.1.5	多边形的拾取函数	123
5.2	几何模型文件	125
5.2.1	WTK 支持的几何模型文件格式	125
5.2.2	模型中的点和面	128
5.3	几何建模函数	131
5.3.1	构建预定义的几何模型	131
5.3.2	构建自定义的几何模型	134
5.3.3	其他几何构建函数	136
5.4	几何模型的处理	137
5.4.1	几何模型的材质函数	137
5.4.2	几何模型中的多边形与顶点	138
5.4.3	几何模型的修改	139
5.4.4	几何模型的优化	140
5.4.5	顶点编辑函数	141
5.5	范围盒框和碰撞检测	144
5.5.1	包围盒框及其相关函数	144
5.5.2	碰撞检测及其相关函数	145
5.5.3	多边形的拾取函数	147
5.6	绘图函数	148
5.6.1	二维绘图函数	148
5.6.2	三维绘图函数	150
第 6 章	传感器	151
6.1	引言	151
6.2	构造与析构	153
6.3	传感器的访问和处理	156
6.3.1	访问传感器的状态	156
6.3.2	传感器输入的旋转	159
6.4	定制传感器驱动程序	162
6.4.1	传感器驱动程序概要	163
6.4.2	传感器驱动函数	164
6.4.3	驱动程序示例	166
6.5	鼠标	171
6.5.1	存取鼠标原始数据	172
6.5.2	鼠标更新函数	172
6.5.3	自己编写更新函数	174
6.5.4	用鼠标拖动物体	175
6.5.5	检查鼠标的输入焦点窗口	176
6.5.6	用鼠标代替跟踪球	177

6.6 升鸟、鸟群和扩展鸟	178
6.6.1 升鸟传感器	178
6.6.2 鸟群	180
6.6.3 扩展范围鸟	183
6.7 几何球(CIS 图形几何球 Jr)	183
6.8 BOOM 设备	184
6.8.1 BOOM 设备	184
6.8.2 BOOM 游戏杆	186
6.9 数据手套	187
6.9.1 捏攥式手套	187
6.9.2 5DT 手套	189
6.10 游戏杆	190
6.11 Logitech 系列设备	193
6.11.1 Logitech 三维鼠标(Red Baron)	193
6.11.2 Logitech 头部跟踪器	195
6.11.3 空间控制鼠标(Magellan)	197
6.12 Polhemus 系列设备	199
6.12.1 ISOTRAK 跟踪器	199
6.12.2 ISOTRAK II 跟踪器	201
6.12.3 InsideTRAK 跟踪器	201
6.12.4 FASTRAK 跟踪器	202
6.13 精确漫游探路者	203
6.14 Spacetec 系列设备	205
6.14.1 Spacetec IMC 空间球	205
6.14.2 Spacetec IMC 空间球空间控制器	207
6.15 立体显示设备	209
6.16 ThrustMaster 系列设备	210
6.16.1 ThrustMaster 方程式 T2 驾驶台	210
6.16.2 ThrustMaster 串行游戏杆	211
6.17 HMD 设备和 Virtual i-O i-glasses! 设备	214
6.17.1 HMD 设备	214
6.17.2 Virtual i-O i-glasses! 设备	215
6.18 CyberGlove	215
6.18.1 CyberGlove 设备的初始化	216
6.18.2 CyberGlove 的校准	217
6.18.3 创建 CyberGlove 的可视化手模型	217
6.18.4 访问手模型对象	219
6.18.5 CyberGlove 弯曲角度数据的访问	220
6.18.6 CyberGlove 手模型常数	221

第 7 章 任务、路径、运动连接	222
7.1 任务	222
7.1.1 任务的创建	222
7.1.2 任务函数	223
7.2 路径	224
7.2.1 路径的构造和析构	225
7.2.2 路径管理	226
7.2.3 路径的装入和保存	227
7.2.4 路径的记录和播放	228
7.2.5 路径点的管理	232
7.2.6 其他函数	234
7.3 运动连接	235
7.3.1 运动连接构成和创建	235
7.3.2 运动连接相关函数	236
7.3.3 运动连接的坐标系和约束	238
第 8 章 串行口和可移植性	241
8.1 串行口	241
8.2 可移植性	242
8.2.1 键盘读入和文件读入	242
8.2.2 消息和错误	244
第 9 章 用户界面对象	247
9.1 UI 应用程序的创建	247
9.1.1 创建步骤和完整示例	247
9.1.2 创建步骤的详细说明	249
9.2 用户界面对象	252
9.3 UI 对象实用函数	257
9.3.1 UI 基本函数	257
9.3.2 仿真循环控制函数	259
第 10 章 网络	261
10.1 客户端服务器网络	261
10.1.1 属性	261
10.1.2 共享组	264
10.1.3 网络连接	268
10.1.4 列举	272
10.2 多点传送式网络	275
10.2.1 WTK 的联网功能	275
10.2.2 相关概念	277
10.2.3 网络函数	278

第 1 章 WTK 概述

1.1 WTK 简介

WTK 是 World Tool Kit 的简称,是美国 Sense8 公司产品,主要用以开发虚拟仿真应用程序。简单地说,WTK 提供了 1 000 多个高效率的 C 函数。一个 C 函数可完成数百行一般的 C 语言代码,因此这些函数的效率很高。利用 WTK 提供的这些函数,我们可以开发三维图形应用程序,在此基础上,还可编写传感器驱动程序,从而开发虚拟现实(Virtual Reality)应用程序。也就是说,我们通过调用 WTK 的函数,来构造自己的虚拟世界。

1.2 两个重要概念

场景结构和仿真引擎是 WTK 中的两个最基本、最重要的概念。

1.2.1 场景结构

WTK 采用层次型场景结构,即在创建虚拟世界应用程序时,我们可将结点组织装配成一个层次型的场景结构图,从而指示仿真引擎如何渲染场景,这样也有效利用了状态保存(state-preserving)和面向堆栈(stack-oriented)的渲染架构。

场景层次结构是场景数据库的一种有效的可视化表示方法,它的优点是,提高了 WTK 的性能、控制能力和柔性。例如可以在场景结构中按层次来选择物体,可以充分利用位姿结点信息,可以打开或关闭细节层次结点,可对结点进行组合,可与 VRML 兼容,可从互联网上载入模型和数据等。利用这种场景图方法,我们可以创建一个光源,然后指定其在场景图中的位置,这样,光源就只照射到我们选择的结点。

场景层次结构通过逐个结点表达场景数据,提高了构建场景的柔性。WTK 还提供了一些非常有用的场景构建函数,通过载入场景描述文件来构建场景。例如,从互联网上载入 VRML 文件到场景中,只要一个简单的函数调用即可。WTK 也提供了一些非常方便的场景修改和重构函数。

1.2.2 仿真引擎

在用 WTK 编写的虚拟世界应用程序中,包含一个仿真引擎。仿真引擎管理着一个仿真循环,主要完成一系列非常重要的任务,如场景的绘制渲染、输入传感器的数据读入、对象状态的更新、几何模型的载入等。仿真循环每秒可处理 5~30 帧,因此可用以开发实时性的应用程序。WTK 的仿真循环和事件发送机制类似于 Windows 管理器,所以 WTK 应用程序的开发和 Windows 程序开发一样,我们只须把精力集中在应用程序的细节上,而不必去维护仿真循环。但与 Windows 应用程序不同的是,在 WTK 应用程序中,用户

的视点或宇宙中的对象在不断地改变。

WTK 结合了 OpenVR™ 的思想,也就是说它在跨平台方面非常方便,包括 SGI, Sun, DEC, Intel 和 Evans and Sutherland 等平台。WTK 进行了优化,以充分发挥它所支持的硬件平台能力,这样我们的应用程序就可运行于多种图形加速系统上。

1.3 WTK 的主要特点

1.3.1 代码的高效性

通常 WTK 中的一个 C 函数,可完成数百行一般的 C 语言代码的功能,因此这些函数的效率很高,可以极大缩短我们的开发周期。

代码的高效性还表现在它借鉴了面向对象的思想,以面向对象的方式来组织这些函数。WTK 虽然不支持继承和动态绑定,但开发 WTK 应用程序与现实世界中的情形类似,如虚拟世界中的对象有自己的属性和行为。

WTK 在函数命名方面也借鉴了面向对象的思想,将函数组织成 20 多个类,如宇宙 (Universe)、几何 (Geometry)、视点 (Viewpoint) 等。WTK 中函数的命名遵循一个规则,即每个对象的函数名称中有一个表示对象子类的类型。如函数 WTsensor 表示一个传感器对象,函数 WTserial 表示一个串行口对象。WTK 中的对象对用户是不透明的,这样有利于数据抽象。控制对象时常常通过指针进行, WTK 对象的内部状态的改变,只能通过调用 WTK 库函数来控制。

为方便起见,每个子类的所有函数均以子类名称作为开始。另外,用户可以存取的所有子类,均有一个对象构造函数和析构函数。构造函数名称以 _new 结尾,析构函数以 _delete 结尾。例如函数 WTviewpoint *WTviewpoint _new() 创建一个视点对象,并返回一个指向该对象的指针,如语句 newview = WTviewpoint _new()。要销毁这一视点对象,可以调用函数 WTviewpoint _delete(newview) 来实现。

绝大多数函数需要一个指向其子类对象的指针作为它的第一个参数,用以指引该函数。在复制一个视点时,必须调用 WTviewpoint _copy() 函数,此函数用一个已有视点作为参数,函数返回一个指针,指向已有视点的一个副本:

```
WTviewpoint *old_viewpoint, *new_viewpoint;  
new_viewpoint = WTviewpoint _copy(old_viewpoint);
```

宇宙对象比较特殊,因为在任一给定时间,只有一个宇宙对象,所以在有关宇宙对象的函数中,不需要宇宙对象指针作为其第一个参数。

1.3.2 跨平台性

利用 WTK 开发出来的应用程序,不仅有高性能的实时性,而且具有跨平台的优点,可在 Unix/Motif 系统和 Windows 系统中运行。WTK 还提供了跨平台的 UI 对象,在新的操作系统环境下,应用程序不必修改,只要重新编译即可运行。此时的 UI 对象在编译时自动更新,以适应当前的操作系统环境。例如我们在 X-Windows 下编写一个带有工具

栏的应用程序,如果在 Windows 环境下重新编译,那么此工具栏就呈现出 Windows 的界面风格。

1.3.3 支持多种外设

WTK 支持多种输入和输出设备,从简单的桌面式设备到复杂的传感器设备,如穿戴于人体、用于感知位置和方向的设备。WTK 提供了读入各种传感器的库函数,提高了用户与虚拟世界应用程序的交互性。对虚拟世界的控制也可通过这些输入传感器来完成,如从简单的鼠标到包含 6 个自由度的输入设备。虚拟世界的展现,可用一般的计算机屏幕作为观察虚拟世界的一个窗口,也可以利用具有位置跟踪的头盔式立体显示系统。必须指出的是,不是所有的传感器设备在所有的系统中均被支持,系统不同,支持的设备也有所不同,具体情况可参阅相关资料。

1.3.4 支持多管道/多处理器

计算机辅助虚拟环境(CAVE),在我国又称为洞穴式虚拟环境,它要求配备高分辨率的立体显示系统。WTK 支持对多个图形管道或屏幕的渲染,可以利用多处理器的计算资源。这一特点使 WTK 非常适用于洞穴式虚拟环境。

1.3.5 其他特点

- 素材及透明性 WTK 充分利用了 OpenGL 的特点,可以完全控制几何的着色,包括高光效果。
- 任务对象 通过对几何、几何结点甚至 C 中结构型数据指派任务,用户可以规定它们的行为。
- 面向渲染绘制的性能优化 支持三角形网格化(triangle stripping),状态分类等。
- 大气效果 支持烟雾、云层等特殊效果。
- 约束 可以约束几何结点或其他场景构件的位姿。
- 支持正射投影 可用于规划视图时或透视投影不适合的情形下。
- 从内存中载入纹理 如视频对象映射于物体表面,并可回放。
- 跨平台的二维图形调用 支持平面图形、线条、位图等。
- 支持三维字体 可在虚拟世界中生成三维字体。
- 支持多种文件格式 如 WRL、FLT、DXF、NFF、OBJ、3DS、BFF、SLP、GEO 等文件格式。
- C++ 包装 用户可选择用 C 或 C++ 进行编程。
- 如 WTK 提供了碰撞检测函数、设备实例化函数、几何动态生成函数、材质处理函数以及渲染控制函数等。声音建模、针对渲染的性能优化,支持多种几何模型数据文件格式等。

1.4 WTK 的子类(class) 总览

WTK 是以面向对象的方式来组织的,绝大多数 WTK 函数的命名也是遵循面向对象的思想,WTK 将这些函数组织成以下子类:

- 宇宙(Universe) 即仿真管理器,它管理着 WTK 虚拟世界中的所有对象,相当于虚拟世界中所有对象的一个容器。在一个虚拟世界应用程序中,只能存在一个宇宙对象。我们不但可以临时增加或删除容器中的对象,而且可以调整仿真循环中事件的执行顺序。
- 几何(Geometry) 是仿真中可见的图形对象,如长方体、球体、圆柱体、三维字体等。我们可以动态创建,也可以从外部导入这些图形对象。在创建一个与此几何对应的几何结点(Geometry Node)后,场景图中才能包含这一几何。
- 结点(Node)场景图由结点构成,结点分为可见结点和不可见结点两类。几何结点是唯一的可见结点。不可见结点有光源(Light)结点、位姿(Transform)结点、细节层次(Level of Detail)结点、开关(Switch)结点等。它们虽不可见,但可以影响几何结点的外观表现。
- 多边形 在 WTK 中可动态创建多边形,可利用多种图形文件对其进行纹理映射。多边形的渲染有三种模式,分别是线框模式、光滑阴影(smooth-shaded)模式和纹理模式。
- 顶点(Vertices) 除了可动态创建外,顶点还可从文件中读入。当顶点与顶点法矢相关联后,就可以进行高氏明暗处理(gouraud shading)。
- 光源(Light) 不但可动态创建,而且还可从光源文件中读入。
- 视点(Viewpoint) 虚拟世界中的几何投影于屏幕上时,视点定义了观察这些几何的位置和方向。WTK 支持一个或多个视点。如果在视点与传感器之间建立联接关系,那么我们就可通过传感器实现对视点的位置和方向的控制。
- 窗口(Windows) 用以显示场景。一个虚拟世界可对应多个窗口。
- 传感器(Sensors) 在与位姿结点、可移动结点或视点等建立联接关系后,可以控制有关对象的运动。WTK 支持多种传感器。
- 路径(Path) 使几何或视点沿事先设定的路径移动,我们动态创建、添加、记录和播放路径。
- 任务(Task) 用以指定单个对象的行为,如运动轨迹、改变外观等。
- 运动连接(Motion Links)把源对象与宿对象联接起来,从而利用源对象的位姿信息控制宿对象的运动。例如可在传感器与视点间建立运动联接关系。
- 声音(Sound) 装载声音对象后,与三维对象相联系,即可在虚拟世界中播放。
- 用户界面(User Interface) WTK 提供的 UI 既可应用于 X/Motif 环境下,又可应用于 Windows 环境下。
- 网络(Networking) 在 PC 机和工作站上的 WTK 应用程序,可通过以太网进行异步通信,这样就可进行分布式仿真,即在网络环境下多台机器运行一个虚拟仿真程序。

- 串行口(Serial Port) 主要用以简化串行口通信任务。

1.5 WTK 的系统配置

1.5.1 通常的系统配置

一个基本的 WTK 开发系统包括数种硬件和软件,如下所示,方括号中的是可选的硬件或软件。

- 主机
- [图形加速卡]
- WTK 库函数
- C 编译器
- [三维建模软件]
- [位图编辑软件]

除了用鼠标作为输入传感器之外,建议至少配置一个 3 自由度或 6 自由度的输入传感器,如空间球 Spacetec IMC Spaceball。

为满足虚拟世界应用程序开发的要求,必须对以上基本配置进行扩展,增加新硬件,如下所示:

- 头盔式立体显示系统或立体投影系统。
- 视频信号转换器,特别是从图形设备的 RGB 信号到头盔式显示系统的 NTSC 视频输入。
- 一个或多个位置跟踪设备,以跟踪记录头部或身体其他部位的位姿。

1.5.2 Windows 下的系统配置

在 Windows 系统中开发或运行 WTK 应用程序时,操作系统必须是 Windows NT 4.0,Windows 95 或 Windows 98。其配置为:

- 必须是 486/66 以上的机器。
- 系统至少有 24M 的 RAM。
- 硬盘至少有 85M。
- 一个或多个图形加速卡,对 OpenGL 图形管道进行加速。
- 光驱一个,用以安装 WTK。

RAM 至少必须满足图形加速卡材质内存的需求,因此系统所需的内存与具体的 WTK 应用程序有关。如果机器使用 3DLab、Dynamic Pictures 和 Permedia 的 OpenGL 驱动程序,那么 WTK 还要在内存中保存有一个材质的副本。另外,如果 WTK 应用程序装入了材质图像,那么 OpenGL 就在虚拟内存中复制这一材质图像。应用程序运行时如果与硬盘交换图像,其速度会降低,而与虚拟内存交换图像时,就可更快地运行应用程序,因此,在 Windows 中使虚拟内存的分配最大化时,可有效提高应用程序的运行速度。

其软件开发环境为 Visual C++ 5.0 或更高版本。

1.5.3 Windows 下支持的外部设备

- 任何标准鼠标（两键或三键）。
- Ascension Bird 和 Ascension Extended Range Bird。
- Ascension Flock of Birds。
- Ascension Mouse。
- CIS Graphics Geometry Ball, Jr.。
- Fakespace 单色 BOOM, 双色 BOOM2C 和全色 BOOM3C(按钮模式和游戏杆模式)。
- Fakespace Pinch Glove System。
- Fifth Dimension 公司的 5DT Glove。
- Gameport 游戏游戏杆。
- Logitech 的 3D 鼠标 (Red Baron)。
- Logitech 的头部跟踪器。
- Logitech 的空间控制鼠标 (Magellan)。
- Polhemus ISOTRAK。
- Polhemus ISOTRAK II。
- Polhemus InsideTRAK (仅 NT 3.51 适用)。
- Polhemus FASTRAK。
- Polhemus Stylus。
- Precision Navigation Wayfinder - VR。
- Spacetec IMC 空间球 2003 型和 Model 3003 型 (只用于点拾按钮)。
- Spacetec IMC 空间球 SpaceController。
- StereoGraphics CrystalEyes 和 CrystalEyesVR LCD 晶闸眼镜。
- ThrustMaster Formula T2 操纵控制台 (仅 NT 适用)。
- ThrustMaster 串行游戏杆(Mark II 飞行控制/武器控制系统)(仅 NT 适用)。
- VictorMaxx 公司的 CyberMaxx2 头盔显示器。
- Virtual i-O i-glasses! 带有头部跟踪的平面和立体显示 (仅 Intergraph 适用)。
- Virtual Technologies 的 Cyberglove。

1.6 WTK 的安装

本节内容除如何在机器上安装 WTK 外,还包含如何配置软件许可证。要编译和运行 WTK 应用程序,必须配置软件许可证。但是 WTK 提供的演示程序已经预先编译过,可以在没有软件许可证的机器上直接运行。

1.6.1 WTK 的安装

以下是在 Windows NT(或 Windows 95, Windows 98)上安装 WTK 的步骤: