

Internet 上描述三维虚拟场景的 Java 3D API

伍祥生* 王克宏

Java 3D API (Java Three-Dimension Application Programming Interface) 是用于实现三维图形显示和基于 Web 的 3D 小应用程序 (applets) 的 Java 编程接口, 它具备了从网络设备编程到三维几何图形编程等各方面的功能, 技术实现于今年 3 月在美国旧金山 JavaOne 会议上首次亮相, 为用户在 Internet 上创建和操作三维几何图形、描述宽大的虚拟世界提供了新的技术。

1 Java 3D API 的由来

1.1 Java 技术在 Internet 上广泛应用的结果

Java 作为一门面向对象的程序设计语言自推出以来, 以其自身的优势, 在短短的 3 年时间里, 已由一种编程语言发展成为一门 Java 技术。它被认为是第一个专门用于 Internet 和 Web 环境的, 面向对象的编程语言。Java 技术的发展极大地促进了 Internet 上应用的开发, 电子购物、电子银行、远程教学、远程医疗等系统相继在 Internet 上出现。为了能使人们从一个真实的物质世界进入 Internet 上的虚拟现实, 实现 Internet 上的电子购物, 电子银行、远程教学和远程医疗等系统的可视化, 就必须加快实现人的视觉与计算机之间理想实时交互技术的研究。基于上述情况, 由 Intel、Silicon Graphics、Apple 和 Sun 4 家公司共同研究、开发了 Java 3D 软件包。

1.2 Java 3D API 的特点

1.2.1 Java 3D API 提供了下面创新的特点:

(1) 高层 (High-level)、面向对象、基于场景图的编程范例实现的复杂的应用程序和小应用程序能在适合于 Java 运行的任何地方快速使用。

(2) 3D 图示数据高性能处理

a. 处理能够实现最佳化, 例如: 挑选、删除和并行场景图横越等。

b. 处理能够位于现存低层 (Low-level) 3D API

的顶部。

(3) 用于创建和观察有趣味的 3D 世界丰富的特点组合。包括: 三角、直线、点、3D 文本、压缩的几何形状、3D 立体性声音、3D 转换、光照、雾、纹理图、音响特性、复杂的视图模型、高级的输入/传感器模型对 6 自由度跟踪装置的支持等等。

(4) 支持运行时输入器容纳广泛的文件格式的多样化, 例如: 特定销售商 CAD 格式、互换格式、VRML 1.0 和 VRML 2.0

1.2.2 Java 3D API 支持的技术特点

(1) 即时存取方式 (Immediate Mode) Java 3D 包括一个公平、完整、一般的即时存取方式。Java 3D 即时存取方式的设计为使用 Java 3D API 编写的所有应用程序实现平台交互能力。

(2) 观察模型 (View Model) Java 3D 提供了一个很复杂的基于观察模型的虚拟现实。从典型的应用的观点来说, Java 3D 的观察模型是相当简单的: 应用者放置、定位和定标一个观察平台周期对象, Java 3D 进行剩下部分的工作。

(3) 跟踪器模型 (Tracker Model) 由于性能原因, Java 3D 观察模型需要暴露一个 6 自由度的头状物体和手跟踪器作为外形对象。暴露允许 API 的实现依靠这些跟踪器使观察模型的计算成为最佳。

(4) 能力比特 (Capability Bits) 在 Java 3D 节点对象中, 能力比特对大部分可修改状态进行校正检查。现在绝大多数的 3D 环境是运行时环境, 不是编辑环境。

(5) 矢量数学库 (Vector Mathematics Library) 在 Java 3D 中建立了一个十分广大的矢量数学库, 所开发的这组数学类扮演 Java 3D 进程的角色。

(6) 浮点和双倍 (Float and Double) 对于单精度和双倍浮点形式, 矢量数学库特别给予支持, 而

* 现为清华大学计算机科学与技术系访问学者

许多 Java 3D 接口给予一般性的支持,似乎它们的支持是相等的。

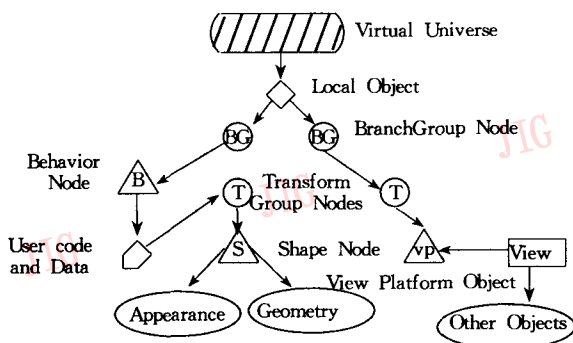
(7) 几何图压缩形式(Geometry Compression Format) Java 3D 是一个运行时 API,它不定义一个外部的文件形式。几何图压缩形式是一个特殊情况,它既是一种运行时二进制形式(用于硬件支持压缩几何图的平台),又是一个应用程序能够使用在文件和网络层的形式。

(8) 高分辨率坐标(High-resolution Coordinates) 为支持大规模虚拟世界,许多应用需要一些标准化更高分辨率坐标系统的形式。为满足这个需要,Java 3D 支持高分辨率坐标。

(9) 其它技术特点 Java 3D 对其它技术特点的支持包括:光照范围(Scoping of Lights)、声音库(Sound Library), VRML (The Virtual Reality Modeling Language) 等等。

2 Java 3D API 规范

一个场景图由 Java 3D 对象组成,这些对象被称为节点,并以树型结构排列。用户创建一个或更多场景子图,并把它们放进一个虚拟的世界。在 Java 3D 节点之间的单独连接总是表示一个直接的关系:从父系到子系。Java 3D 在一个主要方面限制场景图:场景图不可以含有循环。因此,一个 Java 3D 场景图是一个直接的非循环图形(directed acyclic graph),如下图所示。



Java 3D 场景图

Java 3D 把节点对象类分成 2 个子类:簇(Group)和叶(Leaf)节点对象。簇节点对象聚集成一个或更多子节点。一个簇节点能够指向零或更多的子系,但只能有一个父系。SharedGroup 节点不能有任何父系。叶节点对象包含有几何形状、光照、雾、声音等实际定义。一个叶节点没有子系,并仅有

一个父系。

2.1 场景图结构(Scene Graph Structure)

一个场景图组织和控制它的构成对象的处理,Java 3D 处理器绘制一个场景图,还能绘制一个与其它对象无关的对象。Java 3D 能够允许这种独立性,是因为它的场景图有一个特殊的形状,不能在树型结构的分支中间分享状态。

2.1.1 空间分类(Spatial Separation)

场景图体系助长一个根据在图面上的几何图对象的物质空间组合,内部节点起聚集它们的子系在一起的作用。一个簇节点也定义一个空间边界,包括所有由它的子系定义的几何图形、空间组合允许有效的操作实现,例如:近似探测、冲突检测等等。

2.1.2 状态继承(State Inheritance)

在场景图的根和叶之间的一条直线路径上的节点能够定义一个叶节点状态。由于一个叶的图形范围仅取决于根部和那个节点之间的直线路径,Java 3D 处理器能够判定横越场景图的任何次序,它能够从左到右和从顶部到底部穿过场景图,或者从右到左,甚至是并行。这个规则的唯一例外,就是空间边界的属性。例如:光照和雾。

2.1.3 处理(Rendering)

Java 3D 处理器合并了所有的图形状态变化,这些变化是在叶对象的绘制中,由一个场景图根到一个叶对象的直接路径中产生的。Java 3D 为保留和保留编译 2 种方式提供了语义。

2.2 场景图对象(Scene Graph Objects)

一个 Java 3D 场景图由以树型结构连接的 Java 3D 节点的集合所组成。这些节点对象给其它称为节点组件对象的场景图对象作参考。所有的场景图节点和组件对象是一个公共的 SceneGraphObject 类的子类。SceneGraphObject 类是一个抽象类,该类定义的许多方法在节点和组件对象中间是通用的。

场景图对象的作图是通过创建一新的所期望类的实例,用对象的 Set 和 get 方法可以访问和操作场景图对象,一旦一个场景图对象被创建,并与其它场景对象连接形成一个子图,该子图能够全部附属一个虚拟世界。在把一个子图放到虚拟世界之前,整个的子图能够编译成为一个优化的、内部格式。

所有场景图对象的一个重要特征是:它们只能能够在场景图的创建期间被访问或修改,特殊允许的情况除外。下面看看场景图对象的构造程序和

一些方法。

(1) 构造程序 SceneGraphObject 指定一个构造程序 public SceneGraphObject() 构造一个新的 SceneGraphObject。

(2) 方法 下面的方法适合于所有的场景图对象。

```
public final boolean isCompiled()
public final boolean isLive()
```

第1个方法返回一个旗标,表明该节点是否扮演已被编译的一个场景图的角色。第2个方法返回一个旗标,表明该节点是否扮演已贴进一个虚拟世界的一个场景图的角色。

```
public final boolean getCapability (int bit)
public final boolean getCapability (int bit)
public final boolean getCapability (int bit)
```

第3个方法为一个场景图对象的能力比特 (Capability bits) 的访问和修改提供了方法的应用。

```
public void setUserData (Object userData)
public Object getUserData ()
```

这些方法结合这个场景图对象访问或修改 UserData 域。

2.2.1 节点对象(Node Objects)

节点对象划分为簇节点对象和叶节点对象,簇节点用于根据簇节点的语义聚集它们子节点对象在一起。叶节点特别规定 Java 3D 使用在处理中的实际的元素,如:几何图对象、光照和声音。下面看看节点对象的构造程序和方法。

(1) 构造程序 节点对象指定的构造程序如下:public Node()

这个构造程序构造和初始化一个节点对象,节点类为所有的簇和叶节点提供了一个抽象类。它为构造一个 Java 3D 场景图提供了一个公共的框架。

(2) 方法 下面的方法适合于节点对象。

```
public final Node getParent ()
```

它检索该节点的父亲或者零位(如果这个节点没有父系)。这种方法只在场景图构造期间有效,如果这个对象扮演运行或编译场景图的角色,一个 RestrictedAccessException 例外将被抛出。

```
public final Bounds getBounds ()
public final void setBounds (Bounds bounds)
```

这些方法访问或修改这个节点的几何图形的边界。

```
public final void getLocalToWorld (Trans-
```

```
form3D t)
```

```
public final void getLocalToWorld (Scene-
GraphPath path, Transform3D t)
```

这些方法为这个节点访问本地座标到虚拟世界座标变换和安置结果进入指定的 Transform3D argument。

```
public final void setBoundsAutoCompute
(boolean autoCompute)
```

```
public final boolean getBoundsAutoCompute (
)
```

这些方法设置和获取一个值,该值确定节点和几何图边界是否能自动地计算,在什么情况下这些边界只能被读或者手工设置,什么情况下由 set-Bounds 指定的值能使用。

```
public void setPickable (boolean pickable)
public boolean getPickable ()
```

这些方法设置和保持旗标,指出这个节点是否能被选择。

```
public void setCollidable (boolean collidable)
public boolean getCollidable ()
```

以上2个方法中,Set 方法设置碰撞值,get 方法返回碰撞值,该值确定是不是这个节点和它的子系。如果是一个簇节点,就能够被考虑供碰撞用途。

```
public Node cloneNode (boolean forceDupli-
cate)
```

这个方法创建一个新的节点实例。

```
public void duplicateNode (Node originalN-
ode, boolean forceDuplicate)
```

这个方法将所有来自 originalNode 的节点信息复制到现行节点。

```
public Node cloneTree ()
```

```
public Node cloneTree (boolean forceDupli-
cate)
```

```
public Node cloneTree (boolean forceDupli-
cate, boolean allowDanglingReference)
```

这些方法复写指定的子图的所有节点。

2.2.2 节点组件对象(Node Component Objects)

节点组件对象包括实际的几何图和被用来处理几何图的外观属性。

(1) 构造程序 节点组件对象指定下面的构造程序

```
public Node Component ()
```

这个构造程序构造和初始化一个节点组件对象,节点组件类为所有的组件对象提供了一个抽象

类。

(2) 方法 下面的方法适合于节点组件对象。

```
public void setDuplicateOnCloneTree (boolean
duplicate)
```

```
public boolean getDuplicateOnCloneTree (
)
```

这些方法访问或者修改节点组件对象的 duplicateOnCloneTree 的值。由 CloneTree 方法使用 duplicateOnCloneTree 的值来确定节点组件对象是否在没有子叶对象中被复写或者只是被引用。

```
public Node Component CloneNodeComponent(
)
```

这个方法创建了一个新的节点组件对象的实例。

```
public void duplicateNodeComponent (Node-
Component originalNodeComponent)
```

这个方法将来自原来节点组件的所有节点信息复制到现行的节点。

2.3 场景图上层结构对象 (Scene Graph Super-structure Objects)

Java 3D 定义了 2 个场景图上层结构对象: Virtual Universe 和 Locale。它们用来容纳构成场景图的子图收集。

2.3.1 Virtual Universe 对象

一个 Virtual Universe 对象由一序列 Locale 对象组成,它们包括存在这个领域中的场景图节点的收集。典型地,一个应用程序仅需要一个 Virtual Universe,甚至在一个很大的虚拟数据库。一个 Virtual Universe 操作包括列举这个领域的 Locale 对象的个数。

2.3.2 Locale 对象

Locale 对象作为一个容器对场景图的子图的收集,这些子图是经一个 BranchGroup 节点寻找的。一个 Locale 也在虚拟世界中定义了一个特定区域,并使用高分辨率座标指定它的位置。高分辨率座标作为包括在 Locale 中间的所有场景图对象的原点。

一个 Locale 在场景图中没有父系,但当它一构成,隐含地附属于一个虚拟世界。一个 Locale 可为一个任意数量的 BranchGroup 节点作参考,但它没有明确的子系。所有场景图的座标与所包括的 Locale 的高分辨率座标相关联。一个 Locale 的操作包括 Locale 高分辨率座标的设置和获得、增加一个子图和删除一个子图。

2.4 场景图观察对象 (Scene Graph Viewing Objects)

Java 3D 定义了 5 个场景图观察对象,这些对象不是扮演每秒多少场景图的角色,而是作为定义观察参数和提供与物质世界联系的方法。这些对象是 Canvas3D、Screen3D View, PhysicalBody 和 PhysicalEnvironment。

2.4.1 画布 3D 对象 (Canvas3D Object)

画布 3D 对象封装了所有与处理窗口相关联的参数,当一个画布 3D 对象附属于一个观察对象,Java 3D 横贯器处理指定的视图在画布上,多样画布 3D 对象能够指向相同的视图对象。

2.4.2 屏幕 3D 对象 (Screen3D Object)

屏幕 3D 对象封装了所有包括在画布中与物质屏幕相关联的参数,例如:以像素为单位的屏幕的宽和高,屏幕的物质维数和各种各样的物质校验值。

2.4.3 观察对象 (View Object)

观察对象指定需要处理场景图的信息。所有在 Java 3D 的视图参数要么直接包括在观察对象中或者由一观察对象指向这些对象。Java3D 支持多样的同时作用的观察对象,每一个观察对象能处理一个或更多的画布。

2.4.4 物质体对象 (PhysicalBody Object)

物质体对象封装了与物质物体相关联的所有参数。例如:右和左眼的位置等等。

2.4.5 物质环境对象 (PhysicalEnvironment Object)

物质环境对象封装了与物质环境相关联的所有参数,例如:用于头状物体或头盔式跟踪器的校验信息。

3 Java 3D API 的应用

Java 3D API 是一个应用程序编程接口,用它编写三维几何应用程序或基于 Web 的 3D 小应用程序。Java 3D API 可以把一个物体或一个场景准确地描述出来,实行“编写一次,到处运行”的功能。

3.1 三维真实感物体显示

在 Internet 上,用户可以在不同的平台,利用各自的浏览器,随心所欲地观察用 Java 3D API 描述的本地或远程的,有三维真实感的物体,通过 Java

3D API 提供的语言工具,随意对观察物体进行旋转、平移等操作。

3.2 虚拟场景游览

随着电子商务、远程教学系统不断在 Internet 上出现,人们可以利用 Java 3D API 描绘出虚拟的商场,虚拟的教室和医院等。在不久的将来,用户可以通过 Java 3D API,操纵自己的替身在商场购物,在虚拟的教室上课学习。

3.3 多模式的信息访问

计算机技术的发展逐渐地改变了信息的表现形式。Java 3D API 为实现信息的多模式的表现形式提供了良好的基础,现在 Java 3D API 能够实现信息的可视化、可听化和可操作化,相信不久的将来,

随着 Java 3D API 的不断完善和提高,还可能实现信息的可触、可摸,甚至可闻的表现形式。

4 结束语

Java 3D API 的出现为 Java 在 Internet 上的应用和开发注入了新的活力,随着 Java 软件包的不断充实、网络速度的逐渐提高和硬件计算能力的加强,电子购物、远程教学等系统将使人置入一个复杂逼真的虚拟场景。可以预见,这些在 Internet 上的虚拟场景的广泛实现和应用为期不会太远。

参考文献

The Java 3D API White Paper (<http://Java.sun.com:80/Marketing/Collateral/3d-api.html>)

我国首套纯 Java 中文应用产品问世

据报道,美国 SUN 公司与四通利方公司(SRS)宣布,由双方合作建立的“SRS/SUN Java 中文应用开发中心”推出中国首套纯 Java 中文应用产品——跨平台中文信息处理解决方案。

此次推出的我国首套纯 Java 中文应用产品包括 RICHWRITE 和 RICHDICTIONARY 2 个软件。RICHWRITE 为各种平台编写中文文档而设计。它采用双字节内核技术,完全支持中文,不会产生汉字乱码。RICHWRITE 提供全拼、词组拼音、英中和 GB2312 内码等 4 种常用中文输入法。RICHWRITE 的 EXPORT TO HTML 可把文件保存为 HTML 超文本文件,生成的 HTML 文件可用 Netscape Navigator、Microsoft IE、HotJava 等浏览。它具有丰

富的排版功能,支持多种字体、字号、颜色、样式属性及排版格式。

RICHDICTIONARY 是一个简便实用的英汉—汉英双向词典,用 100% 纯 JAVA 设计,可运行于 Solaris、Windows 95、Windows NT 等多种操作系统。

这 2 个软件的最大特点是可以进行跨平台的中文信息处理。SUN 公司首家在其产品中捆绑销售这套解决方案的主体——集成了 RICHWRITE 和 RICHDICTIONARY 的“利方 Java 套餐”。

此处,四通利方不久还将推出用纯 Java 编写的中文 E-mail 系统。利用它,无论在何种语言环境和平台下均可成功收发中文信件。