

第3章 互联网3D几何造型设计



互联网 3D 几何造型组件主要用于创建基本三维立体场景和造型的开发与设计,它包含基本立体几何节点 Sphere 球体节点、Box 盒子节点、Cone 圆锥体节点、Cylinder 圆柱体节点以及 Text 文本造型节点等。利用互联网 3D 立体几何节点创建的造型编程简洁、快速、方便,有利于浏览器的快速浏览,提高软件编程和运行的效率。本章重点介绍简单三维立体几何节点设计语法定义,并结合实例源程序理解软件开发与设计全过程。在互联网 3D 三维立体网页编程语言中,互联网 3D 文件由各种各样的节点组成,“节点”是互联网 3D 的内核,节点之间可以并列或层层嵌套使用。节点在互联网 3D 文件中起着主导的作用,它贯穿于第二代三维立体网络程序设计语言互联网 3D 编程语言始终。可以说,如果没有节点,互联网 3D 文件也就不存在了。理解和掌握互联网 3D 编程语言的“节点”是至关重要的,因为它是互联网 3D 编程设计的灵魂,是互联网 3D 编程的精髓,互联网 3D 三维立体空间造型就是由许许多多“节点”构成并创建的。互联网 3D 简单三维立体几何节点设计主要由 Shape 模型节点、三维立体造型节点以及相关几何节点组成。

3.1 互联网 3D 模型设计

在互联网 3D 文件 Scene 根节点中,添加开发与设计所需要三维立体场景和造型时,在 Shape 模型节点中包含两个子节点,分别为 Appearance 外观节点与 Geometry 几何造型节点。Appearance 外观子节点定义了物体造型的外观,包括纹理映像、纹理坐标变换以及外观的材料节点;Geometry 几何造型子节点定义了立体空间物体的几何造型,如 Box 节点、Cone 节点、Cylinder 节点和 Sphere 节点等原始的几何结构。

Shape 模型节点在互联网 3D 节点与 Scene 场景根节点的基础上,设计添加场景与造型。互联网 3D 节点是互联网 3D 文件中最高一级的互联网 3D 节点,包含 Profile 概貌与 Head 头文件。Head 头文件节点包括 component 组件,metadata 描述说明标签,head 标签节点是互联网 3D 标签的第一个子对象,放在场景的开头。在 head 头文件元素中,加入 meta 子元素描述说明,表示文档的作者、说明、创作日期或著作权等相关信息。如果想使用指定概貌 profile 的集合范围之外的节点,可以在 head 头文件中,加入 component 组件语句,用于描述场景之外的其他信息。

Scene 场景节点表示包含所有互联网 3D 场景语法结构的根节点。在此根节点下增加 Shape 模型节点和子节点以创建三维立体场景和造型,在每个文件里只允许有一个 Scene 根节点。Shape 模型节点是建立在 Scene 场景根节点之下的模型节点,在 Shape 模型节点下,可以创建外观子节点和几何造型子节点,对三维立体空间场景和造型进行外观和几何体描述。

3.1.1 互联网 3D Shape 语法定义

Shape 模型节点是在互联网 3D 文件中(Scene)根场景节点基础上,选择或添加一个 Shape 模型节点或其他节点可以编辑各种三维立体场景和造型。Shape 模型节点定义了一个互联网 3D 立体空间造型所具有的几何尺寸、材料、纹理和外观特征等,这些特征定义了互联网 3D 虚拟空间中创建的空间造型。Shape 节点是互联网 3D 的内核节点,互联网 3D 的所有立体空间造型均使用 Shape 节点创建,所以 Shape 节点在互联网 3D 文件中显得尤为重要。

Shape 模型节点可以放在互联网 3D 文件中任何组节点下,Shape 模型节点可以包含 Appearance 子节点和 geometry 子节点,可以用符合类型定义的原型 ProtoInstance 来替代。Shape 模型节点语法定义如下:

```
< Shape
DEF          ID
USE          IDREF
bboxCenter  0 0 0          SFVec3f          initializeOnly
bboxSize    -1 -1 -1      SFVec3f          initializeOnly
containerField children
class
/>
```

Shape 模型节点 **3** 包含域名、域值、域数据类型以及存储/访问类型等,节点中数据内容包含在一对尖括号中,用“<、/>”表示。Shape 模型节点包含 DEF、USE、bboxCenter、bboxSize、containerField、appearance、geometry 以及 class 域等。

- DEF 为节点定义一个名字,给该节点定义了唯一的 ID,在其他节点中就可以引用这个节点。用 DEF 为节点命名时,使用有意义的描述性的名称可以规范文件,以提高文件可读性。
- USE 用来引用 DEF 定义的节点 ID,即引用 DEF 定义的节点名字,同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。
- bboxCenter 域——表示边界盒的中心,默认值为[0 0 0],域数据类型为一单值三维矢量空间,包含三个浮点数,数与数之间用空格分离,该值表示从原点到所给定点的矢量。存储/访问类型为 initializeOnly。
- bboxSize 域——表示边界盒尺寸大小,默认值为[-1 -1 -1],域数据类型为一个单值三维矢量空间,包含三个浮点数。存储/访问类型为 initializeOnly。为优化三维立体场景,也可以强制指定赋值。
- containerField 域——表示容器域是 field 域标签的前缀,表示了子节点和父节点的关系。该容器域名称为 children,涵盖 appearance 子节点和 geometry 子节点。如 geometry Box、children Group、proxy Shape。containerField 属性只有在 X3D 场景用 XML 编码时才使用。
- appearance 域——定义了一个 Appearance 节点,Appearance 节点定义了物体造型的外观,包括纹理映像、纹理坐标变换以及外观的材料节点。Appearance 域的默认值为 NULL,表示其外观为白色光。该域值为一个单值节点。

- geometry 域——定义了一个几何造型节点,包含 Box 节点、Cone 节点、Cylinder 节点和 Sphere 节点等原始的几何结构。geometry 域默认值为 NULL,表示没有任何几何造型节点。该域值为一个单值节点。
- class 域——用空格分开的类的列表,保留给 XML 样式表使用。只有 X3D 场景用 XML 编码时才支持 class 属性。

3.1.2 互联网 3D Shape 案例分析

利用 Shape 模型节点中的 Geometry 子节点创建各种几何造型,使三维立体空间场景和造型更具真实感。虚拟现实瓶子三维立体造型设计利用互联网 3D 虚拟现实程序设计语言进行设计、编码和调试。利用现代软件开发的极端编程思想,采用绝对编程、自动测试、简单设计以及先测试后设计开发理念。融合结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用虚拟现实语言的各种简单节点创建生动、逼真的瓶子三维立体造型。使用互联网 3D 节点、背景节点、简单几何节点以及坐标变换节点进行设计和开发。

【实例 3-1】 利用 Shape 空间物体造型模型节点、背景节点、基本几何节点、坐标变换节点等在三维立体空间背景下,创建一个颜色为灰色的瓶子造型。简单几何节点将在下面详细讲述,虚拟现实三维立体场景设计互联网 3D 文件源程序展示如下:

```
< Scene >
< Background skyColor = "0.98 0.98 0.98" />
  < Transform translation = "0 5 0" rotation = "0 1 0 0" scale = "0.9 1 1" >
    < Shape >
      < Appearance >
        < Material ambientIntensity = "0.4" diffuseColor = "0.8 0.8 0.8"
          shininess = "0.2" specularColor = "0.7 0.7 0.7" />
      < /Appearance >
    < Sphere radius = "5" />
  < /Shape >
< /Transform >
  < Transform translation = "0 -1 0" rotation = "0 1 0 0" scale = "0.9 1 1" >
    < Shape >
      < Appearance >
        < Material ambientIntensity = "0.4" diffuseColor = "0.8 0.8 0.8"
          shininess = "0.2" specularColor = "0.7 0.7 0.7" />
      < /Appearance >
    < Sphere radius = "4" />
  < /Shape >
< /Transform >
  < Transform translation = "-1 4 4" rotation = "0 1 0 0" scale = "0.9 1 0.8" >
    < Shape >
      < Appearance >
        < Material ambientIntensity = "0.4" diffuseColor = "0.8 0.8 0.8"
          shininess = "0.2" specularColor = "0.7 0.7 0.7" />
      < /Appearance >
    < Sphere radius = "2.5" />
  < /Shape >
< /Transform >
  < Transform translation = "1 4 4" rotation = "0 1 0 0" scale = "0.9 1 0.8" >
```

```

    <Shape>
      <Appearance>
        <Material ambientIntensity = "0.4" diffuseColor = "0.8 0.8 0.8"
          shininess = "0.2" specularColor = "0.7 0.7 0.7"/>
      </Appearance>
    <Sphere radius = "2.5"/>
  </Shape>
</Transform>
<Transform translation = "0.2 5.5 5.2" rotation = "0 1 0 0" scale = "0.8 1 1">
  <Shape>
    <Appearance>
      <Material ambientIntensity = "0.4" diffuseColor = "0.2 0 0"
        shininess = "0.2" specularColor = "0.7 0.7 0.7"/>
    </Appearance>
  <Sphere radius = "1"/>
</Shape>
</Transform>
  :
</Scene>

```

在互联网 3D 源文件中,在 Scene 场景根节点下添加 Background 背景节点和 Shape 模型节点,背景节点的颜色取白色以突出三维立体几何造型的显示效果。利用坐标变换节点和三维立体几何节点创建三维立体造型。

互联网 3D 虚拟现实瓶子三维立体造型设计运行程序,首先,启动 X3D 浏览器,然后在浏览器中即可运行互联网 3D 虚拟现实三维立体造型场景程序,Shape 模型节点源程序运行效果如图 3-1 所示。

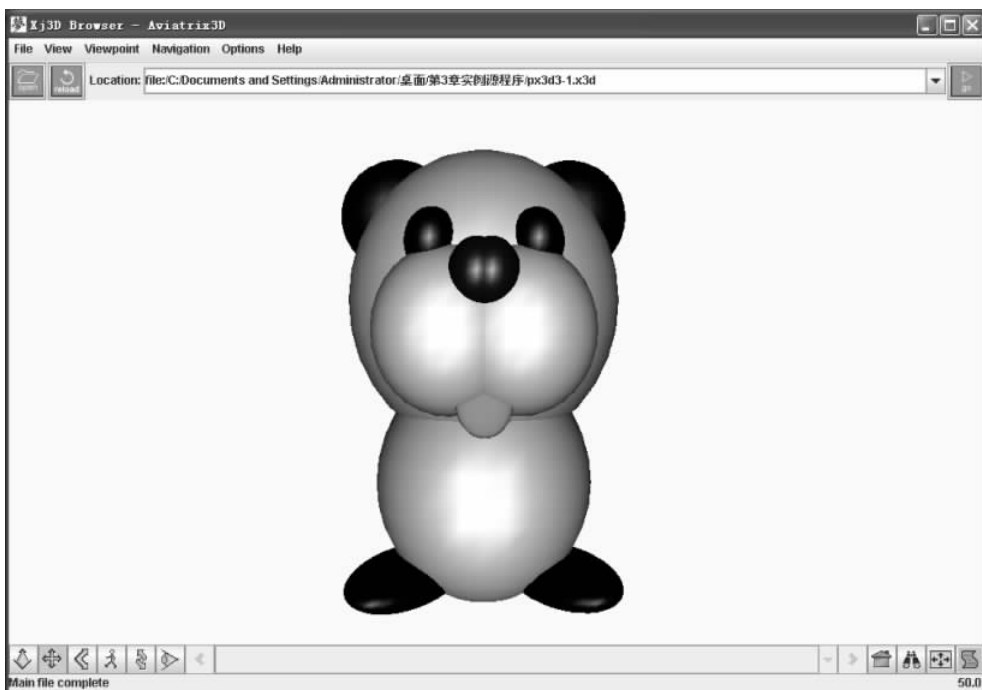


图 3-1 互联网 3D 虚拟现实 Shape 模型节点运行效果

3.2 互联网 3D Sphere 设计

互联网 3D Sphere 球体节点设计描述了一个球体的几何造型。根据球体半径大小的不同,可以改变球体的大小。球体节点通常作为 Shape 节点中 Geometry 子节点。Sphere 球体节点是一个几何节点,用来创建一个三维立体球,根据开发与设计需求可以为 Sphere 球体节点粘贴纹理、设置各种需要颜色以及透明度等。Shape 模型节点可以包含 Appearance 子节点和 Geometry 子节点,Sphere 球体节点作为 Shape 模型节点下 Geometry 几何节点域中的一个子节点。而 Appearance 外观和 Material 材料节点用于描述 Sphere 球体节点的纹理材质、颜色、发光效果、明暗、光的反射以及透明度等。

3.2.1 互联网 3D Sphere 算法分析

互联网 3D 虚拟现实 Sphere 球体表面映射算法分析与实现。

球体表面坐标算法。设球体的球心坐标为 $M_0(X_0, Y_0, Z_0)$, 已知球体半径为 R , 如果 $M(X, Y, Z)$ 为球体表面上任意一点, 则有 $|M_0M| = R$ 。

得到球体表面通用坐标方程如下:

$$(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 = R^2 \quad (3-1)$$

当球体的球心坐标为 $M_0(0, 0, 0)$ 时, 得到球体表面特殊坐标方程为:

$$X^2 + Y^2 + Z^2 = R^2 \quad (3-2)$$

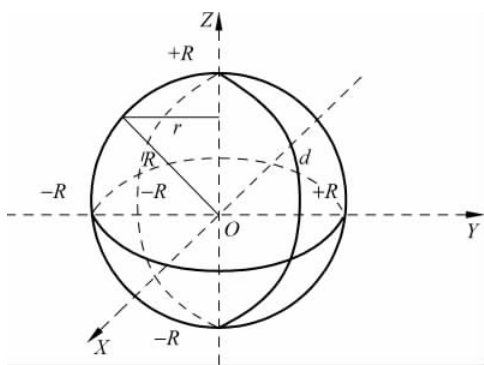


图 3-2 球体算法分析与现实

对三维球体坐标进一步细化,将球体在 X, Y 平面进行极限分隔,形成无数截面,截面圆的半径为 r ,球心到截面的距离为 d ,所得截面圆的半径取值在 $[0, R]$ 之间。

$$r = \sqrt{R^2 - d^2} \quad r \in [-R, +R] \quad (3-3)$$

把复杂三维运算简化为二维运算。得到三维球体坐标简化公式,其中 X, Y 为截面圆上的坐标,球心到截面的距离为 d, r 为截面圆半径, Z 为球体的三维坐标为一个常量,取值范围在 $[-R, +R]$, R 为球体半径,如图 3-2 所示。

$$\begin{cases} X^2 + Y^2 = r^2 & r \in [-R, +R] \\ Z = d & d \in [-R, +R] \end{cases} \quad (3-4)$$

3.2.2 互联网 3D Sphere 语法定义

互联网 3D Sphere 球体节点语法定义了一个三维立体球体的属性和域值,通过 Sphere 球体节点的域名、域值、域的数据类型以及事件的存储访问权限的定义来描述一个三维立体空间球体造型。主要利用球体半径(radius)和实心(solid)参数创建(设置)互联网 3D 球体文件。Sphere 球体节点语法定义如下:

```

< Sphere
  DEF          ID
  USE          IDREF
  radius      1          SFFloat          initializeOnly
  solid       true       SFBool           initializeOnly
  containerField geometry
  class
/>

```

Sphere 球体节点包含域名、域值、域数据类型以及存储/访问类型等,节点中数据内容包含在一对尖括号中,用“<、/>”表示。域数据类型描述如下:

- SFFloat 域——单值单精度浮点数。
- SFBool 域——一个单值布尔量,取值范围为[true|false]。

事件的存储/访问类型描述:表示域(属性)的存储/访问类型,包括 inputOnly(输入类型)、outputOnly(输出类型)、initializeOnly(初始化类型)以及 inputOutput(输入/输出类型)等,用来描述该节点必须提供该属性值。Sphere 球体节点包含 DEF、USE、radius、solid、containerField 以及 class 域等。

- DEF 为节点定义一个名字,给该节点定义了唯一的 ID,在其他节点中就可以引用这个节点。用 DEF 为节点命名时,使用有意义的描述性的名称可以规范文件,以提高 X3D 文件可读性。该属性是可选项。
- USE 用来引用 DEF 定义的节点 ID,即引用 DEF 定义的节点名字,同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。该属性是可选项。
- radius 域——定义一个以原点为球心的三维球体的半径。SFFlot 域类型表示一个单值单精度浮点数,该域值不可小于 0.0,其默认值为 1.0。如果要改变三维立体球体的大小,可以通过改变球体的 radius 半径的域值完成,也可以使用 Transform 节点对三维立体球进行定位、缩放和旋转等设计。
- solid 域——定义了一个布尔量,当该域值 true 时,表示只构建球体对象的表面,不构建背面;当该域值 false 时,表示球体对象的正面和背面均构建。该域值的取值范围为[true|false],其默认值为 true。
- containerField 域——表示容器域是 field 域标签的前缀,表示子节点和父节点的关系。该容器域名称为 geometry,包含几何节点。如 geometry Sphere、children Group、proxy Shape。containerField 属性只有在 X3D 场景用 XML 编码时才使用。
- class 域——是用空格分开的类的列表,保留给 XML 样式表使用。只有 X3D 场景用 XML 编码时才支持 class 属性。

3.2.3 互联网 3D Sphere 案例分析

在 Shape 模型节点中,利用 Geometry 子节点下的 Sphere 球体节点创建三维立体球造型。虚拟现实 Sphere 球体节点三维立体造型设计利用虚拟现实程序设计语言互联网 3D 进行设计、编码和调试。利用结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用互联网 3D 虚拟现实技术的各种节点创建生动、逼真的 Sphere 球体节点三

维立体造型。使用互联网 3D 节点、背景节点、Shape 模型节点以及 Sphere 球体节点进行设计和开发。

【实例 3-2】 在互联网 3D 三维立体空间场景环境下,利用背景节点、Shape 空间物体造型模型节点、Appearance 外观子节点和 Material 外观材料节点以及 Sphere 球体节点在三维立体空间背景下,创建一个三维立体球的互联网 3D 源程序。虚拟现实 Sphere 球体节点立体场景设计互联网 3D 文件源程序展示如下:

```
<?xml version = "1.0" encoding = "UTF - 8"?>
<X3D profile = 'Immersive' version = '3.2' >
  <head>
    <meta content = "px3d3 - 2. x3d" name = "filename"/>
    <meta content = "zjz - zjr - zjd" name = "author"/>
  </head>
  <Scene>
    <! -- Scene graph nodes are added here -->
    <Background skyColor = '1 1 1'/>
    <Shape>
      <Appearance>
        <Material ambientIntensity = "0.4" diffuseColor = "1.0 0.2 0.2"
          shininess = "0.2" specularColor = "0.7 0.7 0.7"/>
      </Appearance>
      <Sphere radius = '1.5'/>
    </Shape>
  </Scene>
</X3D>
```

</X3D>在互联网 3D 源文件中,在 Scene 场景根节点下添加 Background 背景节点和 Shape 模型节点,背景节点的颜色取白色 以突出三维立体几何造型的显示效果。利用基本三维立体几何节点,即球体节点在三维立体创建一个灰色的球体造型显示效果。

互联网 3D 虚拟现实三维立体球造型设计运行程序,首先,启动 X3D 浏览器,即可运行虚拟现实三维立体球造型场景,Sphere 节点序运行效果如图 3-3 所示。

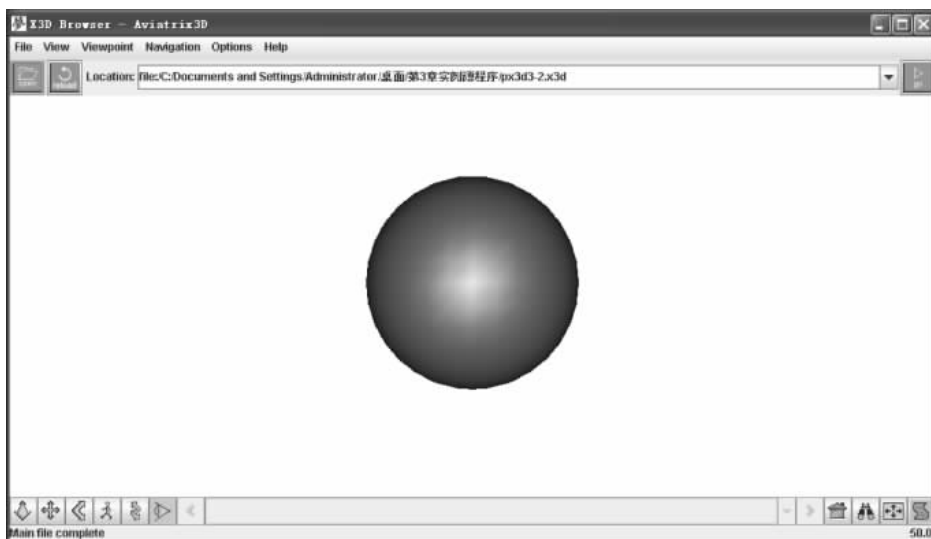


图 3-3 互联网 3D 虚拟现实 Sphere 节点运行效果

3.3 互联网 3D Box 设计

互联网 3D Box 立方体节点设计是一个三维立体基本几何节点,用来创建立方体、长方体以及立体平面的原始几何造型,该节点一般作为 Shape 节点中 Geometry 域的子节点。Box 立方体节点描述了一个立方体的几何造型。根据立方体长、宽和高尺寸大小的不同,可以改变立方体的大小和长短。Box 立方体节点通常作为 Shape 节点中 Geometry 子节点。

3.3.1 互联网 3D Box 语法定义

Box 立方体节点语法定义了一个三维空间立方体造型的属性名、域值、域数据类型、存储和访问类型,通过 Box 立方体节点的域名、域值等来描述一个三维空间立方体造型。主要利用立方体 size 尺寸大小分别定义立方体的长、高和宽和 solid 参数创建互联网 3D 立方体造型。Box 立方体节点语法定义如下:

```
< Box
  DEF          ID
  USE          IDREF
  size         2 2 2          SFVec3f          initializeOnly
  solid        true          SFBool            initializeOnly
  containerField geometry
  class
/>
```

Box 立方体节点包含域名、域值、域数据类型以及存储/访问类型等,节点中数据内容包含在一对尖括号中,用“<、/>”表示。

域数据类型描述如下:

- SFVec3f 域——定义一个三维向量空间。一个 SFVec3f 域值包含三个浮点数,数与数之间用空格分离。该值表示从原点到所给定点的向量。
- SFBool 域——是一个单值布尔量,取值范围为[true|false]。

事件的存储/访问类型描述:表示域(属性)的存储/访问类型,包括 inputOnly(输入类型)、outputOnly(输出类型)、initializeOnly(初始化类型)以及 inputOutput(输入/输出类型)等,用来描述该节点必须提供该属性值。Box 立方体节点包含 DEF、USE、size、solid、containerField、class 域等。

- DEF 为节点定义一个名字,给该节点定义了唯一的 ID,在其他节点中就可以引用这个节点。用 DEF 为节点命名时,使用有意义的描述性的名称可以规范文件,以提高 X3D 文件可读性。该属性是可选项。
- USE 用来引用 DEF 定义的节点 ID,即引用 DEF 定义的节点名字,同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。该属性是可选项。
- size 域——指定一个以原点为中心的空间三维立方体或长方体的尺寸大小。该域值为三维数组,第一个数值为长方体在 X 轴方向上的宽度,第二个数值为长方体在 Y 轴方向上的高度,第三个数值为长方体在 Z 轴方向上的深度。size 域的域值必须

大于 0.0,其默认值为 2.0 2.0 2.0,即立方体的长、宽、高均为 2.0。

- solid 域——定义一个立方体造型表面和背面绘制的布尔量,当该域值 true 时,表示只构建立方体对象的表面,不构建背面;当该域值 false 时,表示立方体对象的正面和背面均构建。该域值的取值范围为[true|false],其默认值为 true。
- containerField 域——表示容器域是 field 域标签的前缀,表示子节点和父节点的关系。该容器域名称为 geometry,包含几何节点。如 geometry Sphere、children Group、proxy Shape。containerField 属性只有在 X3D 场景用 XML 编码时才使用。
- class 域——是用空格分开的类的列表,保留给 XML 样式表使用。只有 X3D 场景用 XML 编码时才支持 class 属性。

3.3.2 互联网 3D Box 案例分析

互联网 3D Box 立方体三维立体造型设计利用虚拟现实程序设计语言互联网 3D 进行设计、编码和调试。利用现代软件开发编程思想,融合结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用虚拟现实语言的各种节点创建生动、逼真的 Box 立方体节点三维立体造型。使用互联网 3D 内核节点、背景节点以及 Box 立方体节点进行设计和开发。在 Shape 模型节点中,利用 Geometry 几何子节点下的 Box 立方体节点创建三维空间立方体或长方体造型,使互联网 3D 三维立体空间场景和造型更具真实感。

【实例 3-3】 在互联网 3D 三维立体空间场景环境下,利用背景节点、视点节点、Shape 空间物体造型模型节点、Appearance 外观子节点和 Material 外观材料节点以及 Box 立方体节点在三维立体空间背景下,创建一个三维空间 Box 长方体的互联网 3D 源程序。Box 立方体节点立体场景设计互联网 3D 文件源程序展示如下:

```
< Scene >
  < Background skyColor = "1 1 1" />
  < Viewpoint orientation = "0 1 0 0.524" position = "3 0 5" />
  < Shape >
    < Appearance >
      < Material ambientIntensity = "0.4" diffuseColor = "0.0 1.0 1.0"
        shininess = "0.2" specularColor = "0.7 0.7 0.7" />
    < /Appearance >
    < Box size = "4 2 2" />
  < /Shape >
< /Scene >
```

在互联网 3D 源文件中的 Scene 场景根节点下添加 Background 背景节点、视点节点和 Shape 模型节点,背景节点的颜色取银白色以突出三维立体几何造型的显示效果。利用基本三维立体几何节点,即立方体节点在三维立体创建一个立方体造型,根据设计需求设置立方体或长方体的尺寸大小,也可以设置 solid 域来绘制立体造型的不同表面。此外还增加了 Appearance 外观节点和 Material 材料节点,以提高空间三维空间立方体造型的显示效果。

互联网 3D 虚拟现实三维立方体造型设计运行程序,首先,启动 X3D 浏览器,可运行虚拟现实三维空间立方体造型场景,在三维立体空间背景下,Box 长方体节点运行后的效果如图 3-4 所示。

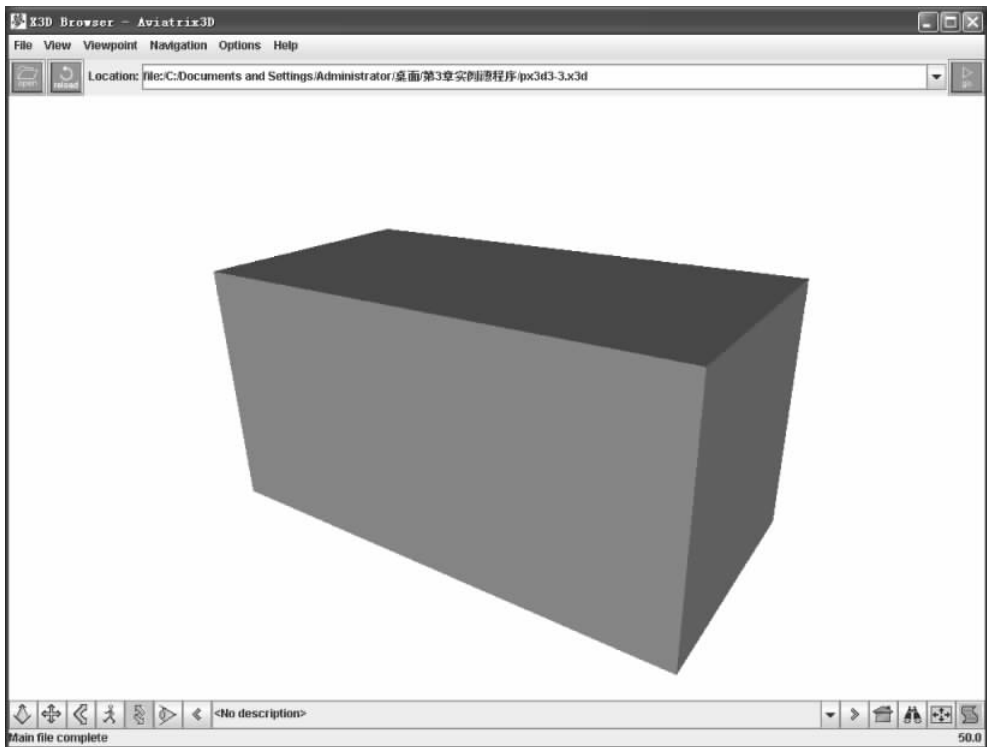


图 3-4 互联网 3D 虚拟现实三维立体 Box 节点运行效果

3.4 互联网 3D Cone 设计

互联网 3D Cone 圆锥体设计是在互联网 3D 文件中创建一个三维立体几何节点,用来在三维立体空间中创建一个圆锥体造型,根据开发与设计需求可以为 Cone 圆锥体节点粘贴纹理、设置各种需要颜色以及透明度等,使浏览者体验三维立体空间各种物体造型的浏览效果。Cone 圆锥体定义了一个圆锥体的原始造型,是互联网 3D 基本几何造型节点,一般作为 Shape 节点中 Geometry 子节点。利用 Shape 节点中 Appearance 外观和 Material 材料子节点用于描述 Cone 圆锥体节点的纹理材质、颜色、发光效果、明暗、光的反射以及透明度等。提高开发与设计的效果 Cone 圆锥体节点描述了一个圆锥体的几何造型。通过设置圆锥体的半径大小、圆锥体高度等参数的不同,可以改变圆锥体的尺寸大小。圆锥体节点通常作为 Shape 节点中 Geometry 几何子节点。

3.4.1 互联网 3D Cone 语法定义

互联网 3D Cone 圆锥体节点语法定义了一个三维立体空间圆锥体造型的属性名和域值,利用 Cone 圆锥体节点的域名、域值、域的数据类型以及事件的存储访问权限的定义来创建一个三维立体空间 Cone 圆锥体造型。主要使用 Cone 圆锥体节点中的高度(height)、圆锥底半径(bottomRadius)、侧面(side)、底面(bottom)以及实心(solid)参数设置创建互联网 3D 圆锥体造型。Cone 圆锥体节点语法定义如下:

```
< Cone
```

```

    DEF          ID
    USE          IDREF
    height       2          SFFloat      initializeOnly
    bottomRadius 1          SFFloat      initializeOnly
    side         true       SFBool       initializeOnly
    bottom       true       SFBool       initializeOnly
    solid        true       SFBool       initializeOnly
    containerField geometry
    class
  />
```

Cone 圆锥体节点包含域名、域值、域数据类型以及存储/访问类型等，节点中数据内容包含在一对尖括号中，用“<、/>”表示。

域数据类型描述如下：

- SFFloat 域——单值单精度浮点数。
- SFBool 域——一个单值布尔量，取值范围为[true|false]。

事件的存储/访问类型描述：表示域(属性)的存储/访问类型，包括 inputOnly(输入类型)、outputOnly(输出类型)、initializeOnly(初始化类型)以及 inputOutput(输入/输出类型)等，用来描述该节点必须提供该属性值。Cone 圆锥体节点包含 DEF、USE、height、bottomRadius、side、bottom、solid、containerField 以及 class 域等。

- DEF 为节点定义一个名字，给该节点定义了唯一的 ID，在其他节点中就可以引用这个节点。用 DEF 为节点命名时，使用有意义的描述性的名称可以规范文件，以提高 X3D 文件的可读性。该属性是可选项。
- USE 用来引用 DEF 定义的节点 ID，即引用 DEF 定义的节点名字，同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。该属性是可选项。
- height 域——指定圆锥体的高。圆锥体的高位于 Y 轴方向，且以原点为中点。该域值为单精度浮点数，存储/访问类型是 initializeOnly。该域值必须大于 0.0，其默认值为 2.0，就是圆锥体的底部的中心位于 Y 轴的 -1.0 处，顶点位于 Y 轴 1.0 处。
- bottomRadius 域——指定以原点为中心，以 Y 轴为中心轴的圆锥体的底部圆的半径。该域值为单精度浮点数，存储/访问类型是 initializeOnly(初始化类型)。该域值必须大于 0.0，其默认值为 1.0，就是该圆锥体底部的半径为 1.0。
- side 域——指定该圆锥体是否有锥面。该域值为单值布尔类型，存储/访问类型是 initializeOnly(初始化类型)。如果该域值为 true，则创建锥面；如果该域值为 false，则不创建。该域值的默认值为 true。
- bottom 域——指定一个是否创建该圆锥体的底部。该域值为单值布尔类型，存储/访问类型是 initializeOnly(初始化类型)。如果该域值为 true，则创建底部；如果该域值为 false，则不创建。该域值的默认值为 true。
- solid 域——定义一个圆锥体造型表面和背面绘制的布尔量，当该域值 true 时，表示

只构建圆锥体对象的表面,不构建背面;当该域值 false 时,表示圆锥体对象的正面和背面均构建。该域值的取值范围为[true|false],其默认值为 true。

- containerField 域——表示容器域是 field 域标签的前缀,表示子节点和父节点的关系。该容器域名称为 geometry,包含几何节点。如 geometry Sphere、children Group、proxy Shape。containerField 属性只有在 X3D 场景用 XML 编码时才使用。
- class 域——是用空格分开的类的列表,保留给 XML 样式表使用。只有 X3D 场景用 XML 编码时才支持 class 属性。

3.4.2 互联网 3D Cone 案例分析

互联网 3D Cone 圆锥体三维立体造型设计利用虚拟现实程序设计语言互联网 3D 进行设计、编码和调试。利用现代软件开发的思想,采用结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用虚拟现实语言的各种节点创建生动、逼真的 Cone 圆锥体节点三维立体造型。使用互联网 3D 节点、背景节点以及 Cone 圆锥体节点进行设计和开发。在 Shape 模型节点中,利用 Geometry 几何子节点下的 Cone 圆锥体节点创建三维立体圆锥体造型,使用 Appearance 外观子节点和 Material 外观材料子节点使三维立体空间场景和造型更具真实感。

【实例 3-4】 在互联网 3D 三维立体空间场景环境下,利用 Shape 空间物体造型模型节点、Appearance 外观子节点和 Material 外观材料节点以及 Cone 圆锥体节点在三维立体空间背景下,创建一个三维立体蓝色圆锥体的互联网 3D 源程序。Cone 圆锥体节点立体场景设计互联网 3D 文件源程序展示如下:

```
< Scene >
  < Background skyColor = "1 1 1" />
  < Shape >
    < Appearance >
      < Material ambientIntensity = "0.4" diffuseColor = "0.0 1.0 0.0"
        shininess = "0.2" specularColor = "0.7 0.7 0.7" />
    < /Appearance >
    < Cone bottom = "true" bottomRadius = "2.0" height = "3" side = "true" />
  < /Shape >
< /Scene >
```

在互联网 3D 源文件中的 Scene 场景根节点下添加 Background 背景节点和 Shape 模型节点,背景节点的颜色取白色以突出三维立体几何造型的显示效果。在 Shape 模型节点下增加 Appearance 外观节点和 Material 材料节点,以提高空间三维立体圆锥体的显示效果。在几何节点中创建 Cone 圆锥体节点,根据设计需求设置 Cone 圆锥体半径和高的尺寸大小,也可以设置 solid 域来绘制立体造型的不同表面。

互联网 3D 虚拟现实三维 Cone 圆锥体造型设计运行程序,首先,启动 X3D 浏览器,可运行虚拟现实三维空间绿色圆锥体造型场景,在三维立体空间背景下,Cone 圆锥体节点运行后的场景效果如图 3-5 所示。

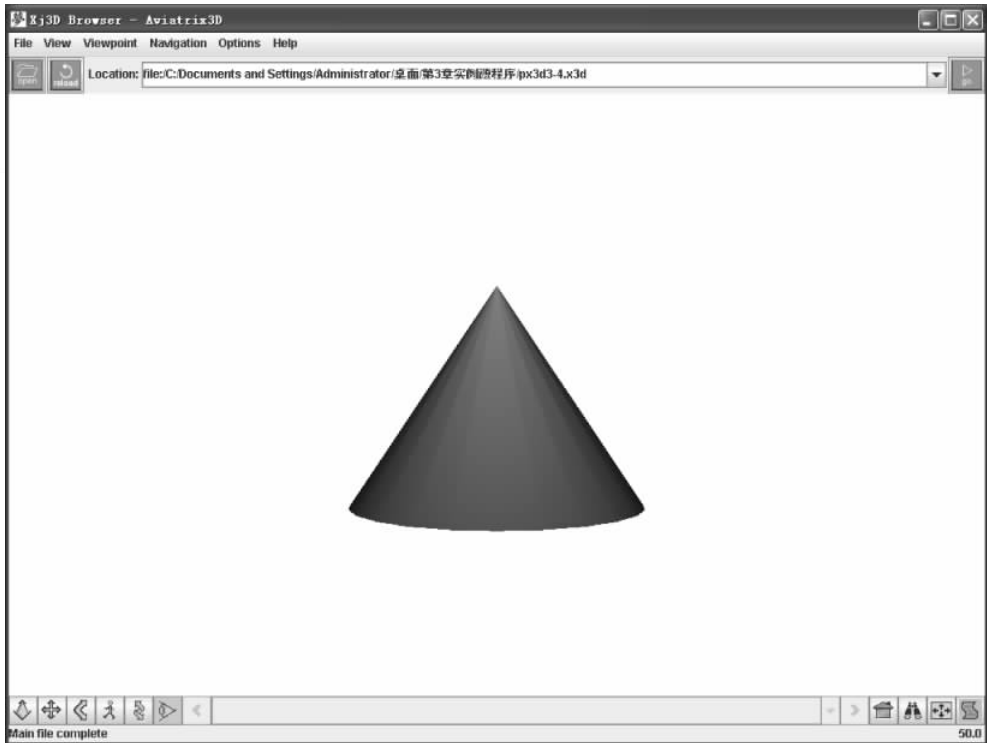


图 3-5 虚拟现实互联网 3D Cone 圆锥体运行效果

3.5 互联网 3D Cylinder 设计

互联网 3D Cylinder 节点设计描述了一个圆柱体的 3D 几何造型。根据圆柱体的半径大小、圆柱体高度的不同,可以改变圆柱体的大小尺寸。圆柱体节点通常作为 Shape 节点中 Geometry 几何子节点。Cylinder 圆柱体节点定义了一个圆柱体的原始造型,是互联网 3D 基本几何造型节点,一般作为 Shape 节点中 Geometry 子节点。利用 Shape 节点中 Appearance 外观和 Material 材料子节点用于描述 Cylinder 圆柱体节点的纹理材质、颜色、发光效果、明暗、光的反射以及透明度等,提高开发与设计的效果。

3.5.1 互联网 3D Cylinder 算法分析

Cylinder 算法分析与实现在虚拟现实立体空间建立三维坐标系(X, Y, Z),将圆柱体的中心线作为虚拟空间三维坐标的中轴线,对圆柱体表面的算法进行分析和设计,如图 3-6 所示。

假设圆柱体的重心点在坐标原点 $(0, 0, 0)$ 上,圆柱体与圆柱体表面构成三维立体空间造型。在三维立体坐标系中,设圆柱表面上任意一点 (θ, γ, δ) 在圆柱表面上的投影坐标为 (x, y, z) 。设圆柱体中心点为 O ,圆柱体的半径为 R ,照片宽

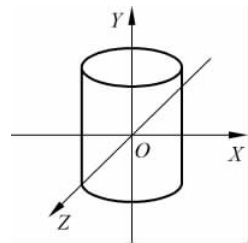


图 3-6 圆柱体三维坐标系

度为 W , 高度为 H 。运用空间解析几何的方法建立数学模型。

$$\begin{cases} x^2 + z^2 = R^2 \\ y = h, \quad h \in [-H/2, H/2] \\ x/\theta = y/\gamma = z/\delta \end{cases} \quad (3-5)$$

求得圆柱体表面上的坐标为 (x, y, z) 。

$$\begin{cases} x = \pm R \cdot \theta / \sqrt{R^2 + \theta^2} \\ y = \pm R \cdot \gamma / \sqrt{R^2 + \theta^2} \\ z = \pm R \cdot \delta / \sqrt{R^2 + \theta^2} \end{cases} \quad (3-6)$$

3.5.2 互联网 3D Cylinder 语法定义

互联网 3D Cylinder 节点语法定义了一个三维立体空间圆柱体造型的属性名和域值, 利用 Cylinder 圆柱体节点的域名、域值、域的数据类型以及事件的存储访问权限的定义来创建一个三维立体空间 Cylinder 圆柱体造型。主要利用 Cylinder 圆柱体节点中的高度 (height)、圆柱底半径 (bottomRadius)、侧面 (side)、底面 (bottom) 以及实心 (solid) 参数设置创建互联网 3D 三维立体圆柱体造型。Cylinder 圆柱体节点定义了一个圆柱体的三维立体造型。通常作为 Shape 节点中 Geometry 域的值。Cylinder 圆柱体节点语法定义如下:

```
<Cylinder
DEF          ID
USE          IDREF
height      2          SFFloat      initializeOnly
radius      1          SFFloat      initializeOnly
top         true       SFBool       initializeOnly
side        true       SFBool       initializeOnly
bottom      true       SFBool       initializeOnly
solid       true       SFBool       initializeOnly
containerField geometry
class
/>
```

Cylinder 圆柱体节点包含域名、域值、域数据类型以及存储/访问类型等, 节点中数据内容包含在一对尖括号中, 用“<、/>”表示。

域数据类型描述如下:

- SFFloat 域——是单值单精度浮点数。
- SFBool 域——是一个单值布尔量, 取值范围为 [true|false]。

事件的存储/访问类型描述: 表示域(属性)的存储/访问类型, 包括 inputOnly(输入类型)、outputOnly(输出类型)、initializeOnly(初始化类型)以及 inputOutput(输入/输出类型)等, 用来描述该节点必须提供该属性值。Cylinder 圆柱体节点包含 DEF、USE、height、radius、top、side、bottom、solid、containerField 以及 class 域等。

- DEF 为节点定义一个名字, 给该节点定义了唯一的 ID, 在其他节点中就可以引用这个节点。用 DEF 为节点命名时, 使用有意义的描述性的名称可以规范文件, 以提高互联网 3D 文件的可读性, 该属性是可选项。

- USE 用来引用 DEF 定义的节点 ID,即引用 DEF 定义的节点名字,同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。该属性是可选项。
- height 域——指定了圆柱体的高。该域值必须大于 0.0,其默认值为 2.0,表示圆柱体的底部的位于 Y 轴的-1 处,顶部位于 Y 轴的 1 处。该域值的尺寸以米为单位,几何尺寸一旦初始化后就不可以再更改,通过使用 Transform 缩放尺寸。
- radius 域——指定了以原点为中心,Y 为轴的圆柱体的半径。该域值必须大于 0.0,其默认值为 1.0。该域值的尺寸以米为单位,几何尺寸一旦初始化后就不可以再更改,通过使用 Transform 缩放尺寸。
- top 域——指定了是否创建该圆柱体的顶部。如果该域值为 true,则创建顶部圆(不画内表面);如果该域值为 false,则不创建顶部圆。该域的默认值为 true,一旦初始化后就不可以再更改。
- side 域——指定是否创建该圆柱体的曲面。如果该域值为 true,则创建曲面(不画内表面);如果该域值为 false,则不创建曲面。该域的默认值为 true,一旦初始化后就不可以再更改。
- bottom 域——指定了是否创建该圆柱体的底部。如果该域值为 true,则创建底部圆(不画内表面);如果该域值为 false,则不创建底部圆。该域的默认值为 true,一旦初始化后就不可以再更改。
- solid 域——定义了一个圆柱体造型表面和背面绘制的布尔量,当该域值 true 时,表示只构建圆柱体对象的表面,不构建背面;当该域值 false 时,表示圆柱体对象的正面和背面均构建。该域值的取值范围为[true|false],其默认值为 true。
- containerField 域——表示容器域是 field 域标签的前缀,表示了子节点和父节点的关系。该容器域名称为 geometry,包含几何节点。如 geometry Sphere、children Group、proxy Shape。containerField 属性只有在互联网 3D 场景用 XML 编码时才使用。
- class 域——是用空格分开的类的列表,保留给 XML 样式表使用。只有互联网 3D 场景用 XML 编码时才支持 class 属性。

3.5.3 互联网 3D Cylinder 案例分析

互联网 3D Cylinder 圆柱体三维立体造型设计利用虚拟现实程序设计语言互联网 3D 进行设计、编码和调试。利用现代软件开发的极端编程思想,采用绝对编程、自动测试、简单设计以及先测试后设计开发理念。融合结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用虚拟现实语言的各种节点创建生动、逼真的 Cylinder 圆柱体节点三维立体造型。使用互联网 3D 内核节点、背景节点以及 Cylinder 圆柱体节点进行设计和开发。Cylinder 圆柱体节点是在 Shape 模型节点中的 Geometry 几何子节点下创建三维立体圆柱体造型,使用 Appearance 外观子节点和 Material 外观材料子节点使造型更具真实感。

【实例 3-5】 在互联网 3D 三维立体空间场景环境下,利用 Shape 空间物体造型模型节点、Appearance 外观子节点和 Material 外观材料节点,以及 Cylinder 圆柱体节点在三维立

体空间背景下,创建一个三维立体黄色圆柱体的互联网 3D 源程序。Cylinder 圆柱体节点立体场景设计互联网 3D 文件源程序展示如下:

```
< Scene >
  < Background skyColor = "1 1 1"/>
  < Shape >
    < Appearance >
      < Material ambientIntensity = "0.4" diffuseColor = "1.0 1.0 0.0"
        shininess = "0.2" specularColor = "1.0 1.0 0.2"/>
    </Appearance >
    < Cylinder height = "3.8" radius = "1.5"/>
  </Shape >
</Scene >
```

在互联网 3D 源文件中的 Scene 场景根节点下添加 Background 背景节点和 Shape 模型节点,背景节点的颜色取银白色以突出三维立体几何造型的显示效果。在 Shape 模型节点下增加 Appearance 外观节点和 Material 材料节点,以提高空间三维立体球的显示效果。在几何节点中创建 Cylinder 圆柱体节点,根据设计需求设置 Cylinder 圆柱体高和半径的尺寸大小,也可以设置 solid 域来绘制立体造型的不同表面。

互联网 3D 虚拟现实三维 Cylinder 圆柱体造型程序运行,首先,启动 X3D 浏览器,单击 Open 按钮,选择相应的文件,即可运行虚拟现实三维空间圆柱体造型场景,在三维立体空间背景下,Cylinder 圆柱体节点运行后的场景效果如图 3-7 所示。

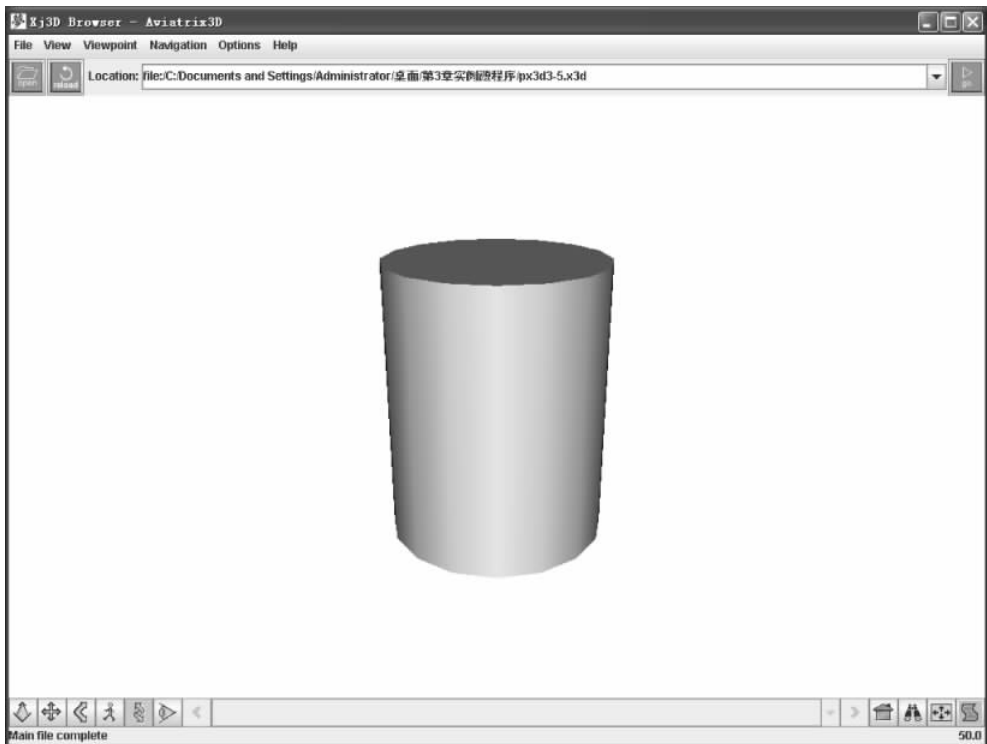


图 3-7 互联网 3DCylinder 节点运行效果

3.6 互联网 3D Text 设计

互联网 3D Text 节点设计用来在互联网 3D 空间中创建文本造型,通常使用 Shape 节点的 Geometry 域的域值。Text 文本造型节点描述了一个文字几何造型。根据文字的文本内容,创建一行或多行文本、定义文本造型的长度以及文本造型的外观特征等。Text 文本造型节点通常作为 Shape 节点中 Geometry 几何子节点。Text 文本造型节点用于在虚拟现实立体空间中创建文字,Text 文本造型节点分别包含 string、length、maxExtent、lineBounds、textBounds 以及 fontStyle 等域,其中 fontStyle 为文本外观子节点。

3.6.1 互联网 3D Text 语法定义

互联网 3D Text 节点语法定义了一个三维立体空间文本造型的属性名和域值,利用 Text 文本造型节点的域名、域值、域的数据类型以及事件的存储访问权限的定义来创建一个三维立体空间 Text 文本造型。主要利用 Text 文本造型节点中的文本内容(string)、文本长度(length)、文本最大有效长度(maxExtent)以及实心(solid)等参数设置创建互联网 3D 三维立体文本造型。Text 文本造型节点语法定义如下:

```
<Text
  DEF          ID
  USE          IDREF
  string       MFString          inputOutput
  length       MFFloat           inputOutput
  maxExtent    0.0               SFFloat           inputOutput
  solid        true              SFBool            initializeOnly
  lineBounds   MFVec2f           outputOnly
  textBounds   SFVec2f           outputOnly
  containerField geometry
  class
/>
```

Text 文本造型节点 **T** 包含域名、域值、域数据类型以及存储/访问类型等,节点中数据内容包含在一对尖括号中,用“<、/>”表示。

域数据类型描述如下:

- SFFloat 域——单值单精度浮点数。
- MFFloat 域——多值单精度浮点数。
- SFBool 域——一个单值布尔量,取值范围为[true|false]。
- MFString 域——一个含有零个或多个单值的多值域,指定了零个或多个字符串。
- SFVec2f 域——定义了一个二维矢量。
- MFVec2f 域——一个包含任意数量的二维矢量的多值域,指定零组或多组二维矢量。

事件的存储/访问类型描述:表示域(属性)的存储/访问类型,包括 inputOnly(输入类型)、outputOnly(输出类型)、initializeOnly(初始化类型)以及 inputOutput(输入/输出类型)等,用来描述该节点必须提供该属性值。Text 文本造型节点包含 DEF、USE、string、

length、maxExtent、solid、lineBounds、textBounds、containerField 以及 class 域等。

- DEF 为节点定义一个名字,给该节点定义了唯一的 ID,在其他节点中就可以引用这个节点。用 DEF 为节点命名时,使用有意义的描述性的名称可以规范文件,以提高 X3D 文件可读性,该属性是可选项。
- USE 用来引用 DEF 定义的节点 ID,即引用 DEF 定义的节点名字,同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。该属性是可选项。
- string 域——指定了要创建的文本内容,其域值可以是一行文本,也可以是多行文本。这些文本均包含在引号之内,在引号内的回车符将被忽略,所以不能使用回车键分行。在其域值中每一行文本用引号分隔每个字符串,不同的行用逗号分开。该域值的默认值为空,即不产生文本造型。另外,字符串中包含引号中套用的引号时,在内层的引号前加反斜杠如 "say \"hello\" please"。如果需要,许多 XML 工具自动替换涉及的 XML 字符(如 & 替换为 & #38; 或 " 替换为 & #34;)。
- length 域——用来指定文本字符串的长度,是以 X3D 单位为计量单位的,这里的长度指的是每一行文本的长度,参照局部坐标系。当设定一个值后,浏览器通过改变字符尺寸或字符间距来进行压缩或扩展,以满足设定长度的要求。Length 域的值和 string 域值是一一对应的,即一个数值控制一行文本。所以其实压缩或扩展并不是固定数值的,它与 string 域值有关。当所设定的长度大于其文本造型本来的长度时,则扩展,反之压缩。该域值的默认值为空列表,即为 0.0,表示即不扩展也不压缩。
- maxExtent 域——指定了文本造型中所对应的行的最大有效长度,也是以 X3D 单位为计量单位的,参照局部坐标系。该域值必须大于 0.0,而对那些长度大于所设定长度的行,通过改变字符尺寸或字符间距来进行压缩。该域值的默认值为 0.0,表示对文本造型的长度没有限制,可以为任意长度。
- solid 域——定义了一个 TEXT 文本造型表面和背面绘制的布尔量,当该域值 true 时,表示只构建文本造型对象的表面,不构建背面;当该域值 false 时,表示文本造型对象的正面和背面均构建。该域值的取值范围为[true|false],其默认值为 true。
- fontStyle 域——用来定义文本造型的外观特征。一般情况下,其域值为 FontStyle 子节点,该域值的默认值为 NULL,即没有定义外观特征。使用默认的外观特征,如左对齐,从左到右,文本尺寸为 1.0,文本间距为 1.0 以及默认的字体大小等。
- lineBounds 域——定义了一个 TEXT 文本造型在本地坐标系中,由每个线围成的二维区域范围的文本造型线段。该域值为 outputOnly(输出类型),是多值的二维矢量。
- textBounds 域——定义了一个 TEXT 文本造型在本地坐标系中,由所有线围成的二维文本造型区域。该域值为 outputOnly(输出类型),是一个单值二维矢量。
- containerField 域——表示容器域是 field 域标签的前缀,表示了子节点和父节点的关系。该容器域名称为 geometry,包含几何节点。如 geometry Box、children Group 和 proxy Shape。containerField 属性只有在 X3D 场景用 XML 编码时才使用。

- class 域——是用空格分开的类的列表,保留给 XML 样式表使用。只有 X3D 场景用 XML 编码时才支持 class 属性。

3.6.2 互联网 3D Text 案例分析

互联网 3D Text 文本造型节点三维立体设计利用虚拟现实程序设计互联网 3D 进行设计、编码和调试。利用现代软件开发的极端编程思想,采用绝对编程、自动测试、简单设计以及先测试后设计开发理念。融合结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用虚拟现实语言的各种节点创建生动、逼真的 Text 文本三维立体造型。使用互联网 3D 节点、背景节点以及 Text 文本造型节点进行设计和开发。Text 文本造型节点是在 Shape 模型节点中的 Geometry 几何子节点下创建三维立体文本造型,使用 Appearance 外观子节点和 Material 外观材料子节点描述空间物体造型的颜色、材料漫反射、环境光反射、物体镜面反射、物体发光颜色、外观材料的亮度以及透明度等,使三维立体空间场景和造型更具真实感。

【实例 3-6】 在互联网 3D 三维立体空间场景环境下,利用 Shape 空间物体造型模型节点、Appearance 外观子节点和 Material 外观材料节点以及 Text 文本造型节点在三维立体空间背景下,创建一个三维立体文字造型节点的互联网 3D 源程序。Text 文本造型节点立体场景设计互联网 3D 文件源程序展示如下:

```
< Scene >
  < Background skyColor = "1 1 1"/>
  < Shape >
    <!-- Add a single geometry node here -->
    < Appearance >
      < Material ambientIntensity = "0.4" diffuseColor = "1.0 0.0 0.0"
        shininess = "0.2" specularColor = "0.2 0.2 0.2"/>
    </ Appearance >
    < Text string = "Web3D http://www.x3d.com", "X3D - Augmented Reality",
      "X3D - Virtual Reality", "2012 - 05 - 28" >
      <!-- FontStyle justify = "MIDDLE" "MIDDLE" / -->
    </ Text >
  </ Shape >
</ Scene >
```

在互联网 3D 源文件中的 Scene 场景根节点下添加 Background 背景节点和 Shape 模型节点,背景节点的颜色取白色以突出三维立体几何文字造型的显示效果。利用基本三维立体 Text 文本造型节点创建一个白色背景下的文字造型,此外增加了 Appearance 外观节点和 Material 材料节点,对物体造型的外观颜色、物体发光颜色、外观材料的亮度以及透明度的设计,提高三维空间文字造型的显示效果。

互联网 3D 虚拟现实三维 Text 文本造型程序运行,首先,启动 X3D 浏览器,单击 Open 按钮,然后打开对应文件,可运行虚拟现实三维空间 Text 文本造型场景,在三维立体空间背景下,显示四行文本文字的效果图,在文本中四行不同的文字用逗号隔开,可以设定两行不同的长度以及设定字符串最大有效长度等。Text 文本造型节点运行后的场景效果如图 3-8 所示。



图 3-8 Text 文本造型节点运行效果

3.6.3 互联网 3D FontStyle 语法定义

互联网 3D FontStyle 文本外观节点是 Text 文本节点的子节点,用来控制文本造型的外观特征,通过设定 FontStyle 节点可以改变由 Text 节点创建的文本造型的外观、字体、字型、风格和尺寸大小等。FontStyle 文本外观节点用于在虚拟现实立体空间中创建文字的同时,利用该节点对文字的外观进行设计,FontStyle 文本外观节点分别包含 family、style、justify、size、spacing、language、horizontal、leftToRight 以及 topToBottom 域(属性)等。

FontStyle 文本外观节点语法定义了一个三维立体空间文本外观的属性名和域值,利用文本外观节点的域名、域值、域的数据类型以及事件的存储访问权限的定义来创建一个效果更加理想的三维立体空间文字造型。主要利用 FontStyle 文本外观节点中的 family(字体)、style(文本风格)、justify(摆放方式)、size(文字大小)、spacing(文字间距)、language(语言)、horizontal(文本排列方式)等参数设置创建互联网 3D 三维立体文本外观造型。FontStyle 文本外观节点语法定义如下:

```

< FontStyle
DEF                ID
USE                IDREF
family             SERIF           MFString       initializeOnly
style              "PLAIN"
                  [ PLAIN| BOLD| ITALIC|

```

```

                                BOLDITALIC]    SFString    initializeOnly
justify                          BEGIN        MFString    initializeOnly
size                              1.0        SFFloat    initializeOnly
spacing                           1.0        SFFloat    initializeOnly
language                          SFString
horizontal                        true        SFBool     initializeOnly
leftToRight                       true        SFBool     initializeOnly
topToBottom                       true        SFBool     initializeOnly
containerField    fontStyle
class
/>

```

FontStyle 文本外观节点^①包含域名、域值、域数据类型以及存储/访问类型等,节点中数据内容包含在一对尖括号中,用“<、/>”表示。

域数据类型描述如下:

- SFFloat 域——单值单精度浮点数。
- SFBool 域——一个单值布尔量,取值范围为[true|false]。
- SFString 域——单值字符串类型。
- MFString 域——一个含有零个或多个单值的多值域,指定了零个或多个字符串。

事件的存储/访问类型描述:表示域(属性)的存储/访问类型,包括 inputOnly(输入类型)、outputOnly(输出类型)、initializeOnly(初始化类型)以及 inputOutput(输入/输出类型)等,用来描述该节点必须提供该属性值。FontStyle 文本外观节点包含 DEF、USE、family、style、justify、size、spacing、language、horizontal、leftToRight、topToBottom、containerField 以及 class 域等。

- DEF 为节点定义一个名字,给该节点定义了唯一的 ID,在其他节点中就可以引用这个节点。用 DEF 为节点命名时,使用有意义的描述性的名称可以规范文件,以提高 X3D 文件可读性,该属性是可选项。
- USE 用来引用 DEF 定义的节点 ID,即引用 DEF 定义的节点名字,同时忽略其他的属性和子对象。使用 USE 来引用其他的节点对象而不是复制节点可以提高性能和编码效率。该属性是可选项。
- family 域——用来指定在 X3D 文件中使用的系列字体名,浏览器按排列顺序优先使用第一个可用字体。支持值包括 SERIF、SANS、TYPEWRITER。其中 SERIF 是指 serif 字体,是一种变宽的字体,如 Times Roman 字体;SANS 是指 sans 字体,也是一种变宽的字体,如 Helvetica 字体;TYPEWRITER 是指 typewriter 字体,是一种等宽字体,如 Coutier 字体。这里要注意的是,在 X3D 浏览器中实际显示的字体类型是与浏览器本身有关的,当该域值设定为 SERIF 时,浏览器也有可能显示 New York 字符集,这是由浏览器本身的设置决定的。该域值的默认值为 SERIF。另外,字符串变量可以是多值,由用引号" "分开每一个字符串(如"so separate"、"each string"、"by"、"quote marks")。
- style 域的值用来指定所显示的文本的风格。设置文字是常规体、粗体、斜体或粗斜

体等,该域值通常包括 PLAIN、BOLD、ITALIC 和 BOLDITALIC,这些都是浏览器所能支持的风格。其中,PLAIN 为常用字体,既不加粗又不倾斜; BOLD 为加粗字体; ITALIC 为倾斜字体; BOLDITALIC 为既加粗又倾斜的字体。该域值的默认值为 PLAIN。

- justify 域——用来指定文本造型中文本块的摆放方式,既是设置左对齐,右对齐,还是居中对齐,这是相对 X 轴或 Y 轴来说的。该域值为一个含有一个或两个值的列表。当含有两个选项值时,要用逗号分开,且都包含在括号中。其中第 1 个值为主对齐方式,第 2 个值为次对齐方式。这些值可从 FIRST、BEGIN、MIDDLE 和 END 中选择。当是水平排列的文本造型时,选 MIDDLE 时,则表示该文本造型的中点在 Y 轴上。该域值的默认值为 BEGIN,这是主对齐方式,次对齐方式按照 FIRST 处理。另外,字符串变量可以是多值,可用引号" " 分开每一个字符串(如" so separate"、"each string"、"by"、"quote marks")。
- size 域——用来指定所显示的文本字符的高度,单位为 X3D 文件单位,参照局部坐标系,也设定了字符的默认行间距。改变其高度可以进而改变文本字符的尺寸大小,该域值的默认值为 1.0。
- spacing 域——用来指定所显示的文本字符的间距,即调节行间距的比例。当文本是水平排列时,该间距指的是水平间距,而当文本是垂直排列时,该间距指的是垂直间距。该域值的默认值为 1.0。
- language 域——用来指定 string 域值中所使用的语言,如英语、法语、德语等。在 X3D 中可以使用的 language 域值是基于在 POSIX 和 RFC 1766 等几个国际标准中的规范的。该语言编码包括主编码和一系列子编码(可能是空)。
- horizontal 域——表示控制着文本造型的排列方式,既是水平排列还是垂直排列。该域值的默认值为 true,表示文本为水平排列;如将 horizontal 域值设定为 false,表示两行文本造型是垂直排列的。
- leftToRight 域——用来指定相邻字符在水平方向上的摆放,即决定字符是从左到右(true)还是从右到左(false)。当 leftToRight 域值为 true 时,相邻字符沿 X 正方向从左到右排列;当 leftToRight 域值为 false 时,则从右到左排列。
- topToBottom 域——用来指定相邻字符在垂直方向上的摆放,即决定字符方向是顶到底(true)还是底到顶(false)。如果当 topToBottom 域值为 true 时,相邻字符沿 Y 负方向从上到下排列;而当 topToBottom 域值为 false 时,相邻字符沿 Y 正方向从下往上排列。
- containerField 域——表示容器域是 field 域标签的前缀,表示子节点和父节点的关系。该容器域名称为 fontStyle,包含几何节点,如 geometry Box、children Group、proxy Shape。containerField 属性只有在 X3D 场景用 XML 编码时才使用。
- class 域——是用空格分开的类的列表,保留给 XML 样式表使用。只有 X3D 场景用 XML 编码时才支持 class 属性。

3.6.4 互联网 3D FontStyle 案例分析

互联网 3D FontStyle 文本外观节点是 Text 文本造型节点,而 Text 文本造型节点又是在 Shape 模型节点中的 Geometry 几何子节点下的三维立体文本造型,使用 Appearance 外观子节点和 Material 外观材料子节点描述空间物体造型的颜色、材料漫反射、环境光反射、物体镜面反射、物体发光颜色、外观材料的亮度以及透明度等,使三维立体空间场景和文字造型更生动和鲜活,具有真实感受。

FontStyle 文本外观节点三维立体设计利用虚拟现实程序设计语言互联网 3D 进行设计、编码和调试。利用现代软件开发思想,融合结构化、组件化和模块化的设计思想,使软件开发设计层次清晰、结构合理。利用虚拟现实语言的各种节点创建生动、逼真的文本三维立体造型。使用互联网 3D 节点、背景节点、Text 文本造型节点以及 FontStyle 文本外观节点进行设计和开发。

【实例 3-7】 在互联网 3D 三维立体空间场景环境下,利用 Shape 空间物体造型模型节点、Appearance 外观子节点和 Material 外观材料节点、Text 文本造型节点以及 FontStyle 文本外观节点等在三维立体空间背景下,显示两行文本造型。具有 BOLDITALIC 为既加粗又倾斜的字体,并且文本造型位于 X、Y 轴的中心点上,创建一个三维立体文字造型的互联网 3D 源程序。FontStyle 文本外观节点立体文字场景设计互联网 3D 文件源程序展示如下:

```
< Scene >
  < Background skyAngle = "1.571" skyColor = "0.2 0.2 1.0 1.0 1.0 1.0" />
  < Shape >
    <!-- Add a single geometry node here -->
    < Appearance >
      < Material ambientIntensity = "0.4" diffuseColor = "1.0 0.0 0.0"
        shininess = "0.2" specularColor = "0.2 0.2 0.2" />
    </ Appearance >
    < Text string = "广东珠海哈尔滨三维立体动画游戏设计", "X3D - Augmented Reality",
      "X3D - Virtual Reality", "2012 - 05 - 28" ' >
      < FontStyle justify = "MIDDLE" "MIDDLE" ' />
    </ Text >
  </ Shape >
</ Scene >
```

在互联网 3D 源文件中的 Scene 场景根节点下添加 Background 背景节点和 Shape 模型节点,背景节点的颜色取淡蓝色以突出三维立体几何文字造型的显示效果。利用基本三维立体 Text 文本造型节点和 FontStyle 文本外观节点共同创建一个淡蓝色背景下的文字造型,此外增加了 Appearance 外观节点和 Material 材料节点,对物体造型的外观颜色、物体发光颜色、外观材料的亮度以及透明度的设计,提高三维空间文字造型的显示效果。

互联网 3D 虚拟现实三维 Text 文本造型和 FontStyle 文本外观节点程序运行,首先,启动 X3D 浏览器,单击 Open 按钮,然后打开相应程序文件,运行虚拟现实三维空间 Text 文本造型和 FontStyle 文本外观节点场景,在三维立体空间背景下,显示四行文本文字的效果

图,选用 BOLDITALIC(既加粗又倾斜)的字体,并且文本造型位于 X、Y 轴的中心点上。文本外观节点程序运行后的场景效果如图 3-9 所示。



图 3-9 Text 和 FontStyle 节点运行效果