



X3D 互动游戏建模交互分析与设计主要完成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 X3D 互动游戏立体几何节点创建造型编程简洁、快速、方便,有利于浏览器的快速浏览,能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的语法定义,并结合案例源程序帮助读者理解软件开发与设计的全过程。在 X3D 互动游戏的三维立体网页互动游戏动画设计中,X3D 互动游戏文件由各种各样的节点组成,节点之间可以并列或层层嵌套使用。节点在 X3D 互动游戏文件中起着主导的作用,它贯穿于 X3D 互动游戏开发的始终。理解和掌握 X3D 互动游戏技术的“节点”是至关重要的,因为它是 X3D 互动游戏编程设计的灵魂,是 X3D 互动游戏编程的精髓,X3D 互动游戏的三维立体空间造型就是由许许多多“节点”构成并创建的。X3D 互动游戏的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。

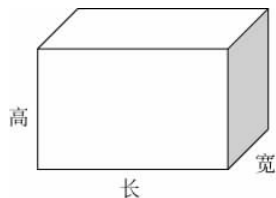
3.1 X3D 互动游戏几何建模算法分析

X3D 互动游戏三维建模分析与设计是对 X3D 互动游戏中的基本几何节点(如球体、圆柱体、立方体、圆锥体、长方体、椭球体)以及文字造型等进行算法分析、结构分析以及语法剖析等,即从理论和实际层面进行分析和设计。算法分析是运用数学公式对三维立体几何造型进行设计,提高用户对三维几何体的理解,通过结构分析使用户对三维几何体有一个宏观掌握,通过语法剖析使用户对几何节点有更深刻的理解,最终实现 X3D 互动游戏三维几何造型的开发与设计。

X3D 互动游戏几何建模算法分析对球体、圆柱体、立方体以及长方体等用数学方法来分析和设计,通过计算机编程来实现三维几何体造型。

1. 长方体算法

根据长、宽、高 3 个参数设计一个长方体,若一个长方体的长、宽、高相等,则为一个立方体,如图 3-1 所示。



2. 球体算法

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

图 3-1 长方体算法分析

果 $M(X, Y, Z)$ 为球体表面上的任意一点, 则有 $|M_0M| = R$ 。

得到球体表面的通用坐标方程:

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

当球体的球心坐标为 $M_0(0, 0, 0)$ 时得到球体表面的特殊坐标方程:

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

对三维球体坐标进一步细化, 将球体在 X, Y 平面进行极限分割, 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范围为 $[0, R]$ 。

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

把复杂三维运算简化为二维运算, 得到三维球体坐标的简化公式, 其中 X, Y 为截面圆上的坐标, 球心到截面的距离为 d , r 为截面圆的半径, Z 表示球体的三维坐标为一个常量, 取值范围为 $[-R, +R]$, 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. 圆柱体算法

在虚拟现实立体空间中建立三维坐标系 (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 3-3 所示。

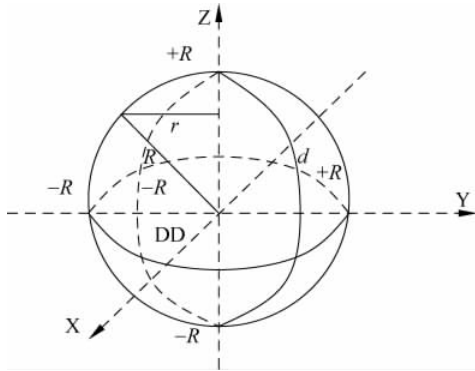


图 3-2 球体算法分析

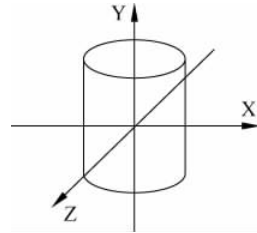


图 3-3 圆柱体三维坐标系

假设圆柱体的中心点在坐标原点 $(0, 0, 0)$ 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, 设圆柱表面上的任意一点 (θ, γ, δ) 在圆柱体表面上的投影坐标为 (x, y, z) , 设圆柱体的中心点为 O , 圆柱体的半径为 R , 圆柱体宽度为 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) 。

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

3.2 X3D 互动游戏几何建模分析

X3D 互动游戏几何建模分析涵盖 X3D 互动游戏几何建模结构分析和 X3D 互动游戏几何语法剖析。X3D 互动游戏几何建模结构分析包含 X3D 球体、X3D 圆锥体、X3D 立方体、X3D 圆柱体、X3D 文本等分类和设计，X3D 互动游戏几何语法剖析主要针对 X3D 球体语法、X3D 圆锥体语法、X3D 立方体语法、X3D 圆柱体语法、X3D 文本语法等进行详细的剖析和设计。

3.2.1 X3D 互动游戏几何建模结构分析

X3D 互动游戏三维几何建模结构分析指对基本几何体进行分析设计，包括球体、圆柱体、立方体、椭球体以及圆锥体等。首先根据球体的半径设计一个球体；然后编写圆柱的底半径和高确定一个圆柱体；根据长、宽、高创建一个长方体；通过圆锥的底半径和高编写一个圆锥体造型，如图 3-4 所示。

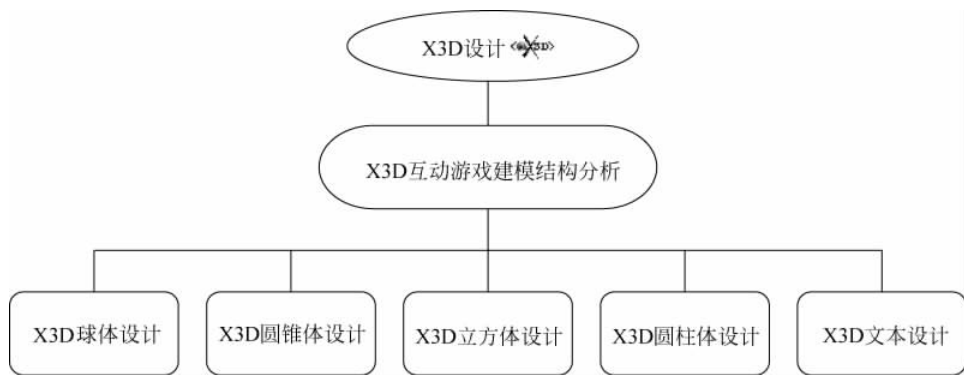


图 3-4 X3D 互动游戏几何体结构分析与设计

X3D 互动游戏三维几何建模结构分析与设计：利用球节点创建球体造型需要给出球体半径，生成一个三维立体球；使用 X3D 互动游戏圆锥体节点需要提供圆锥的半径、高、侧面等信息创建一个三维立体圆锥体；利用 Box 节点设计立方体和长方体，只需要给出长、宽、高等信息，生成一个三维立体的立方体或长方体造型；使用 X3D 互动游戏圆柱体节点需要提供半径和高以及侧面等参数创建一个圆柱体；利用文本节点创建三维立体文字造型等。

3.2.2 X3D 互动游戏几何建模语法剖析

X3D 互动游戏建模设计主要涵盖 Shape 模型节点、Sphere 球体节点、Box 立方体节点、Cone 圆锥体节点、Cylinder 圆柱体节点、Text 文本节点等的语法定义、解释和分析。

在 X3D 互动游戏文件 Scene(场景)根节点中添加开发与设计所需要的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，例如 Box 节

点、Cone 节点、Cylinder 节点和 Sphere 节点等原始的几何结构。

X3D 互动游戏节点是 X3D 互动游戏文件中最高一级的 X3D 互动游戏节点,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 X3D 互动游戏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、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 X3D 互动游戏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。


1. Shape 模型节点

在 X3D 互动游戏文件中的 Scene 根场景节点的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 X3D 互动游戏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 X3D 互动游戏虚拟空间中创建的空间造型。Shape 节点是 X3D 互动游戏的内核节点,X3D 互动游戏的所有立体空间造型均使用 Shape 节点创建,所以 Shape 节点在 X3D 互动游戏文件中显得尤为重要。

Shape 模型节点可以放在 X3D 互动游戏文件中的任何组节点下,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 来代替。Shape 模型节点的语法定义如下:

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

```
>/>
```

 Shape 模型节点包含域名、域值、域数据类型以及存储/访问类型等,节点中的数据内容包含在一对尖括号中,用一对</>表示。Shape 模型节点包含 DEF、USE、bboxCenter、bboxSize、containerField 以及 class 域等。

2. Sphere 球体节点

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 节点中的 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、设置各种需要的颜色以及透明度等。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。

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中的球体造型,主要利用球体半径(radius)和实心(solid)参数创建(设置)X3D 互动游戏球体文件。Sphere 球体节点的语法定义如下:

```
< 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,节点中的数据内容包含在一对尖括号中,用一对</>表示。域数据类型描述 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。Sphere 球体节点包含 DEF、USE、radius、solid、containerField 以及 class 域等。

3. Box 立方体节点

Box 立方体节点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。根据立方体的长、宽和高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 节点中的 geometry 子节点。

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/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,主要利用立方体的 size 大小分别定义立方体的长、高和宽,用 solid 等参数创建 X3D 互动游戏中的立方体造型。Box 立方体节点的语法定义如下:

```
< 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,节点中的数据内容包含在一对尖括号中,用一对</>表示。域数据类型描述 SFVec3f 域定义了一个三维向量空间,一个 SFVec3f 域值包含 3 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 以及 class 域等。

4. Cone 圆锥体节点

Cone 圆锥体节点设计是在 X3D 互动游戏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、设置各种需要的颜色以及透明度等,使浏览者体验三维立体空间中各种物体造型的浏览效果。Cone 圆锥体节点定义了一个圆锥体的原始造型,是 X3D 互动游戏中的基本几何造型节点,一般作为 Shape 节点中的 geometry 子节点。Shape 节点中的 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、圆锥体的高度等参数可以改变圆锥体的尺寸大小。

Cone 圆锥体节点语法定义了一个三维立体空间中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中的 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)参数创建 X3D 互动游戏中的圆锥体造型。Cone 圆锥体节点的语法定义如下:

```
< 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,节点中的数据内容包含在一对尖括号中,用一对<、/>表示。域数据类型描述 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。Cone 圆锥体节点包含 DEF、USE、height、bottomRadius、side、bottom、solid、containerField 以及 class 域等。

5. Cylinder 圆柱体节点

Cylinder 圆柱体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尺寸。圆柱体节点通常作为 Shape 节点中的 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 X3D 互动游戏中的基本几何造型节点。Shape 节点中的 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。

Cylinder 圆柱体节点语法定义了一个三维立体空间中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中的 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)参数创建 X3D 互动游戏中的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 节点中的 geometry 域的值。Cylinder 圆柱体节点的语法定义如下:

```
<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,节点中的数据内容包含在一对尖括号中,用一对<、/>表示。域数据类型描述 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。

6. Text 文本节点

Text 文本节点设计用来在 X3D 互动游戏空间中创建文本造型,通常使用 Shape 节点的 geometry 域的值。Text 文本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、定义文本的长度以及文本的外观特征等。Text 文本节点通常作为 Shape 节点中的 geometry 几何子节点。Text 文本节点用于在虚拟现实立体空间中创建文字,Text 文本节点包含有 string、length、maxExtent、lineBounds、textBounds 以及 fontStyle 等域,其中 fontStyle 为文本外观子节点。

Text 文本节点语法定义了一个三维立体空间中文本的属性名和域值,利用 Text 文本节点的域名、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中的 Text 文本,主要利用 Text 文本节点中的文本内容(string)、文本长度(length)、文本最大有效长度(maxExtent)以及实心(solid)等参数创建 X3D 互动游戏中的三维立体文本造型。Text 文本节点的语法定义如下:

```
<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
/>

```

T Text 文本节点包含域名、域值、域数据类型以及存储/访问类型等,节点中的数据内容包含在一对尖括号中,用一对</>表示。域数据类型描述 SFFloat 域是单值单精度浮点数; MFFloat 域是多值单精度浮点数; SFBool 域是一个单值布尔量,取值范围为[true | false]; MFString 域是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 域是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 文本节点包含 DEF、USE、string、length、maxExtent、solid、lineBounds、textBounds、containerField 以及 class 域等。

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时对文字的外观进行设计,FontStyle 文本外观节点包含 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中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)等参数创建 X3D 互动游戏中的三维立体文本外观造型。FontStyle 文本外观节点的语法定义如下:

```

< 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 文本外观节点包含域名、域值、域数据类型以及存储/访问类型等,节点中的数据内容包含在一对尖括号中,用一对</>表示。域数据类型描述 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];SFString 域是单值字符串类型;MFString 域是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。

3.3 X3D 互动游戏几何建模开发设计

X3D 互动游戏开发设计利用基本几何体构建一个三维造型,利用软件工程的思想对三维几何体进行开发与设计,使用模型节点、球节点、圆柱体节点、圆锥体节点、坐标变换节点、外观节点、材料节点以及纹理绘制节点等进行开发与设计。

3.3.1 X3D 球体造型设计

这里以简单的 X3D 互动游戏球体为例,利用集成开发环境快速建模与设计。首先下载 BS Content Studio 集成开发环境,安装并运行该程序。

在主菜单中找到 Node List View(节点列表视图),选择 Standard→Primitives→Sphere Primitive 创建一个球体造型,并调整球体半径为 3.5,如图 3-5 所示。

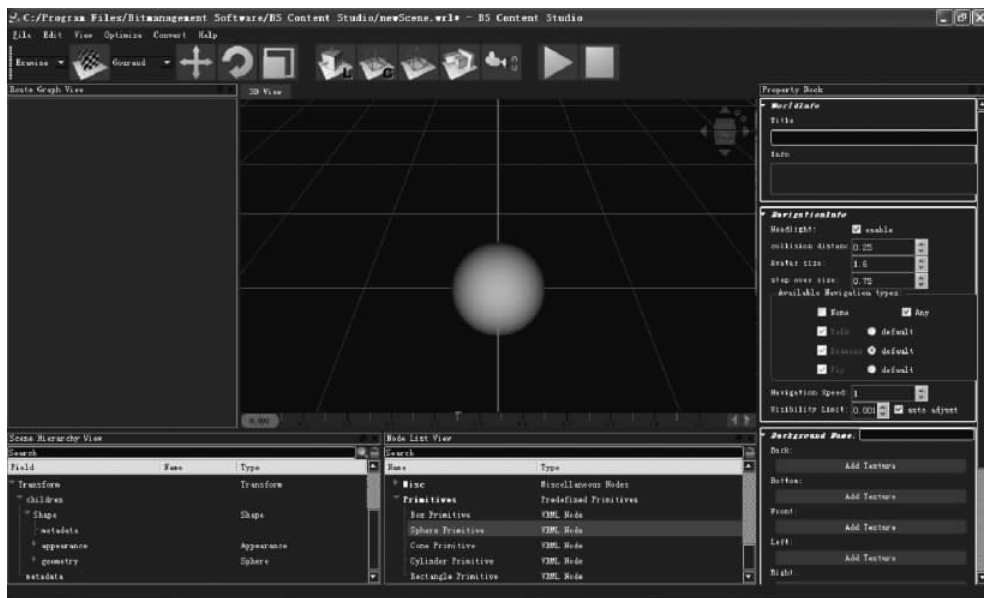


图 3-5 球体造型设计

X3D 互动游戏球体造型设计：利用 Shape 模型节点中的 geometry 子节点创建各种几何球节点造型；利用 X3D 互动游戏虚拟现实集成开发环境进行设计、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；利用虚拟现实集成开发环境中的各种基本节点创建生动、逼真的 3D 球造型；使用 X3D 互动游戏节点、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。

在本书附带光盘的 X3D 互动游戏案例源程序/第 3 章案例源程序目录下提供了 X3D 互动游戏源程序 px3d3-1. x3d。

【案例 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 3D 球造型。

X3D 互动游戏球体造型文件 px3d3-1. x3d 的源程序如下：

```
<?xml version = "1.0" encoding = "UTF - 8"?>
<X3D profile = "Immersive" version = "3.0">
  < head >
    < meta content = "zjz - zjr - zjd" name = "author"/>
    < meta content = ' * enter FileNameWithNoAbbreviations. x3d here * ' name = 'title'/>
    < meta content = ' * enter description here, short - sentence summaries preferred * ' name =
'description'/>
    < meta content = ' * enter version here, if any * ' name = 'version'/>
    < meta content = ' * enter copyright information here * Example: Copyright (c) Web3D
Consortium Inc. 2015' name = 'rights'/>
  </ head >
  < Scene >
    < Background DEF = "_1" skyColor = '0.1 0.1 0.1'>
    </ Background >
    < WorldInfo DEF = "_2">
    </ WorldInfo >
    < NavigationInfo DEF = "_3" type = "EXAMINE", "ANY">
    </ NavigationInfo >
    < Transform DEF = "_4">
      < Shape DEF = "_5">
        < Appearance DEF = "_6">
          < Material DEF = "_7">
          </ Material >
        </ Appearance >
        < Sphere DEF = "_8" radius = '3.5'>
        </ Sphere >
      </ Shape >
    </ Transform >
  </ Scene >
</ X3D >
```

3.3.2 X3D 圆锥体造型设计

根据圆锥体的高、底半径等参数来设计圆锥体，利用集成开发环境快速建模，首先启动开发环境，在主菜单中找到 Node List View，选择 Standard→Primitives→Cone Primitive 创建一个圆锥体造型，并调整圆锥体的高 height=12、圆锥体的底半径 bottomRadius=6。然后调

整圆锥体的颜色,在 Scene Hierarchy View(场景层次视图)中选择 Shape→Appearance→Material,在主窗口右侧的 Property Dock(属性窗口)中选择 diffuseColor=0.8 0 0.03921569,如图 3-6 所示。

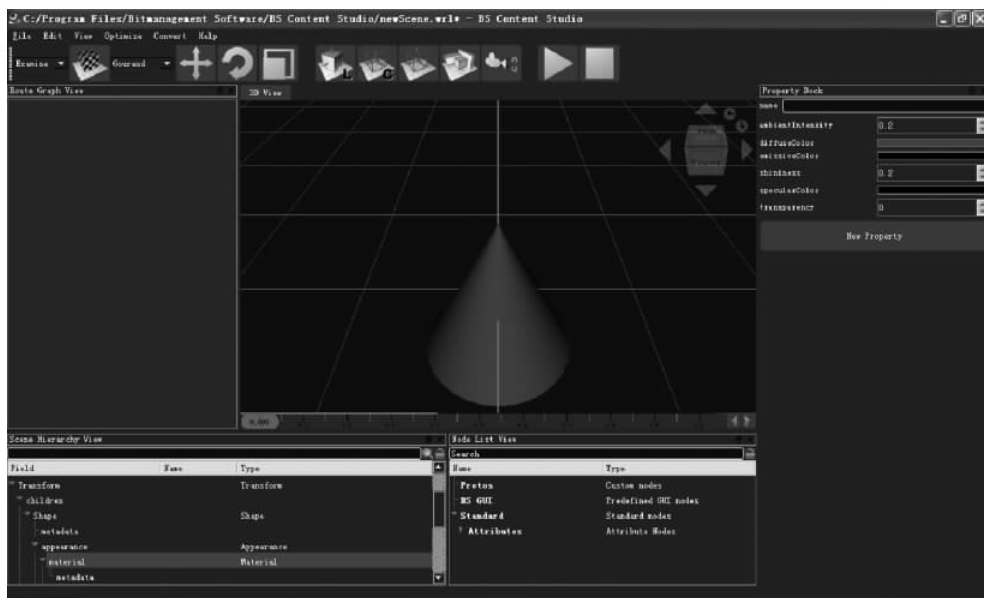


图 3-6 红色的圆锥体造型设计

在本书附带光盘的 X3D 互动游戏案例源程序/第 3 章案例源程序目录下提供了 X3D 互动游戏源程序 px3d3-2. x3d。

【案例 3-2】 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红色的圆锥体 3D 造型。

X3D 互动游戏圆锥体造型文件 px3d3-2. x3d 的源程序如下:

```
<?xml version = "1.0" encoding = "UTF - 8"?>
<X3D profile = "Immersive" version = "3.0">
< head>
< meta content = "px3d3 - 2. x3d" name = "filename"/>
    < meta content = "zjz - zjr - zjd" name = "author"/>
    < meta content = ' * enter FileNameWithNoAbbreviations. x3d here * ' name = 'title'/>
    < meta content = ' * enter description here, short - sentence summaries preferred * ' name =
'description'/>
    < meta content = ' * enter version here, if any * ' name = 'version'/>
    < meta content = ' * enter copyright information here * Example: Copyright (c) Web3D
Consortium Inc. 2015' name = 'rights'/>
</ head>
< Scene>
    < Background DEF = "_1" skyColor = '0.1 0.1 0.1'>
</Background>
    < WorldInfo DEF = "_2">
</WorldInfo>
    < NavigationInfo DEF = "_3" type = "EXAMINE", "ANY">
</NavigationInfo>
```

```

<Transform DEF = "_4">
  <Shape DEF = "_5">
    <Appearance DEF = "_6">
      <Material DEF = "_7" diffuseColor = '0.8 0 0.03921569'>
      </Material>
    </Appearance>
    <Cone DEF = "_8" bottomRadius = '6' height = '12'>
    </Cone>
  </Shape>
</Transform>
</Scene>
</X3D>

```

3.3.3 X3D 圆柱体造型设计

结合圆柱体节点和域值,根据圆柱体的高、底半径等参数来设计圆柱体。利用集成开发环境快速建模,首先启动开发环境,在主菜单中找到 Node List View(节点列表视图),选择 Standard→Primitives→Cylinder Primitive 创建一个圆柱体造型。然后选中圆柱体,调整圆柱体的高 height = 16、圆柱体的底半径 Radius = 5。接着设置圆柱体的颜色,在 Scene Hierarchy View(场景层次视图)中选择 Shape→Appearance→Material,在主窗口右侧的 Property Dock(属性窗口)中选择 diffuseColor=0 0 1,设置背景颜色为白色, Sky Color=1 1 1,如图 3-7 所示。

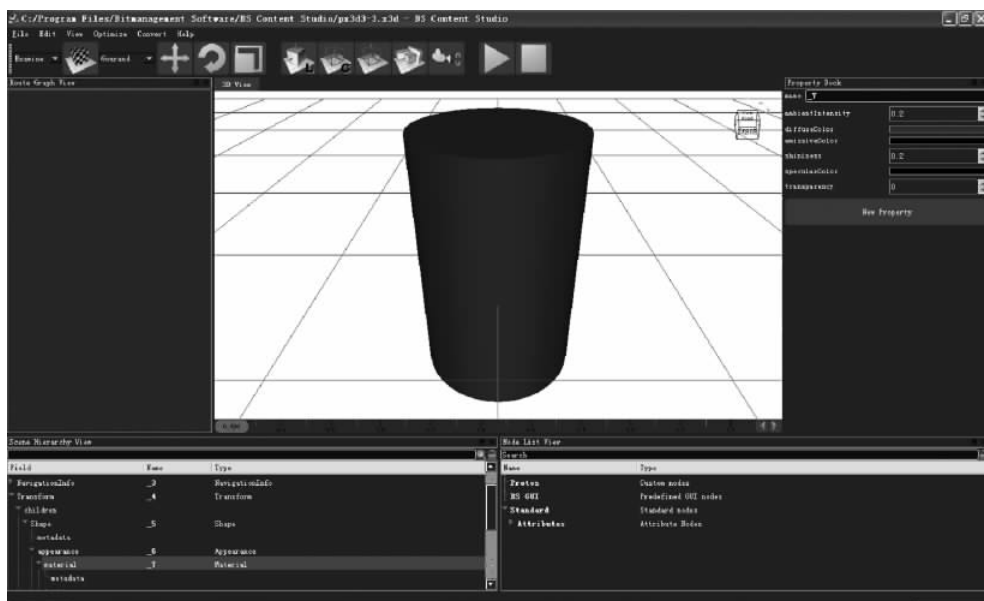


图 3-7 蓝色的圆柱体造型设计

在本书附带光盘的 X3D 互动游戏案例源程序/第 3 章案例源程序目录下提供了 X3D 互动游戏源程序 px3d3-3. x3d。

【案例 3-3】 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,设置白色的三维立体空间背景,创建一个蓝色的圆柱体 3D 造型。

X3D 互动游戏圆柱体造型文件 px3d3-3. x3d 的源程序如下：

```
<?xml version = "1.0" encoding = "UTF - 8"?>
<X3D profile = "Immersive" version = "3.0">
  < head >
< meta content = "px3d3 - 3. x3d" name = "filename"/>
    < meta content = "zjz - zjr - zjd" name = "author"/>
    < meta content = ' * enter FileNameWithNoAbbreviations. x3d here * ' name = 'title'/>
    < meta content = ' * enter description here, short - sentence summaries preferred * ' name =
'description'/>
    < meta content = ' * enter version here, if any * ' name = 'version'/>
    < meta content = ' * enter copyright information here * Example: Copyright (c) Web3D
Consortium Inc. 2015' name = 'rights'/>
  </ head >
  < Scene >
    < Background DEF = "_1" skyColor = '1 1 1'>
    </ Background >
    < WorldInfo DEF = "_2">
    </ WorldInfo >
    < NavigationInfo DEF = "_3" type = "EXAMINE", "ANY">
    </ NavigationInfo >
    < Transform DEF = "_4">
      < Shape DEF = "_5">
        < Appearance DEF = "_6">
          < Material DEF = "_7" diffuseColor = '0 0 1'>
          </ Material >
        </ Appearance >
        < Cylinder DEF = "_8" height = '16' radius = '5'>
        </ Cylinder >
      </ Shape >
    </ Transform >
  </ Scene >
</ X3D >
```

3.3.4 X3D 立方体造型设计

设置立方体的宽、高、深参数，默认为 $X=2, Y=2, Z=2$ 。利用集成开发环境快速建模，首先启动开发环境，在主菜单中找到 Node List View (节点列表视图)，选择 Standard → Primitives → Box Primitive 创建一个立方体造型。然后选中立方体，调整立方体的参数为 $X=10, Y=10, Z=10$ 。接着绘制立方体纹理，在 Scene Hierarchy View (场景层次视图) 中选择 Shape → Appearance → texture，在 Node List View (节点列表视图) 中选择 Texturing → Image Texture，双击该选项，然后在 Scene Hierarchy View (场景层次视图) 中选中 texture Image Texture，这时在主窗口右侧的 Property Dock (属性窗口) 中显示 url 路径，选择纹理图 m3698.jpg 文件所在的路径，设置背景颜色为白色，Sky Color = 1 1 1，如图 3-8 所示。

在本书附带光盘的 X3D 互动游戏案例源程序/第 3 章案例源程序目录下提供了 X3D 互动游戏源程序 px3d3-4. x3d。

【案例 3-4】 利用 Shape 空间物体造型模型节点、背景节点、基本几何节点、纹理图像以及坐标变换节点等，设置白色的三维立体空间背景，创建一个立方体纹理 3D 造型。

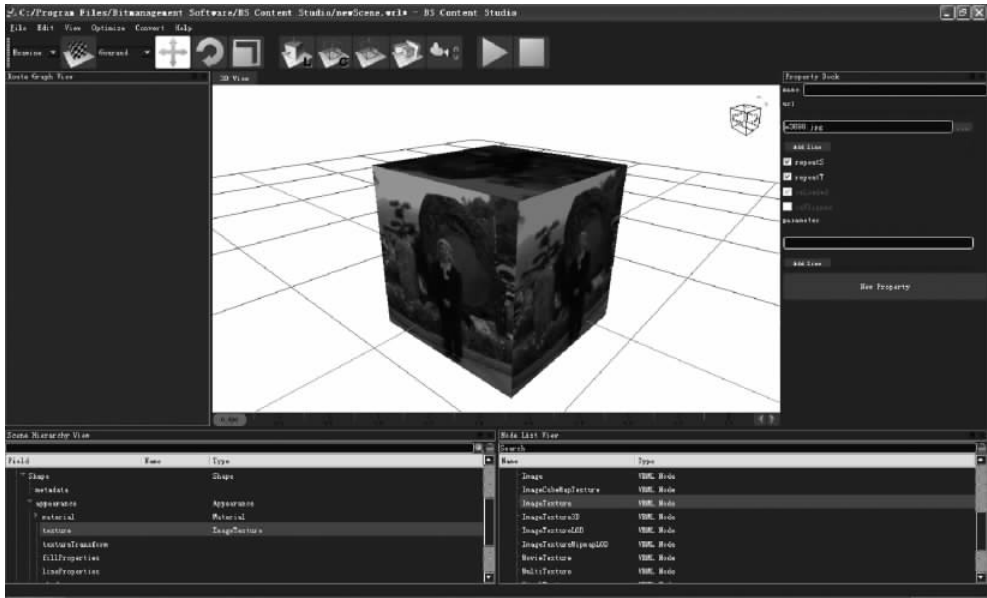


图 3-8 立方体纹理造型设计

X3D 互动游戏立方体纹理造型文件 px3d3-4. x3d 的源程序如下：

```

<?xml version = "1.0" encoding = "UTF - 8"?>
<X3D profile = "Immersive" version = "3.0">
  <head>
<meta content = "px3d3 - 4. x3d" name = "filename"/>
    <meta content = "zjz - zjr - zjd" name = "author"/>
    <meta content = ' * enter FileNameWithNoAbbreviations. x3d here * '
name = 'title'/>
    <meta content = ' * enter description here, short - sentence summaries
preferred * ' name = 'description'/>
    <meta content = ' * enter version here, if any * ' name = 'version'/>
    <meta content = ' * enter copyright information here * Example: Copyright (c)
Web3D Consortium Inc. 2015' name = 'rights'/>
  </head>
  <Scene>
    <Background DEF = "_1" skyColor = '1 1 1'>
  </Background>
    <WorldInfo DEF = "_2">
  </WorldInfo>
    <NavigationInfo DEF = "_3" type = "EXAMINE", "ANY">
  </NavigationInfo>
    <Transform DEF = "_4" translation = '- 2.149768 4 - 2.112515'>
      <Shape DEF = "_5">
        <Appearance DEF = "_6">
          <Material DEF = "_7">
            <ImageTexture DEF = "_8" url = "'m3698. jpg" '>
          </ImageTexture>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>

```

```

        <Box DEF = "_9" size = '10 10 10'>
            </Box>
        </Shape>
    </Transform>
</Scene>
</X3D>

```

3.3.5 X3D 文字造型设计

设置文字造型字符串,并对文字外观进行设计。利用集成开发环境快速创建一个文字造型和纹理地球,首先启动开发环境,在主菜单中找到 Node List View(节点列表视图),选择 Background→metdata,在主窗口右侧的 Property Dock(属性窗口)中找到 Sky Color 属性,选择天空颜色为白色,然后单击 OK 确定。

在 Node List View(节点列表视图)中分别选择 Groups→Transform、Appearances→Appearance、Geometry→Text 添加到 Scene Hierarchy View(场景层次视图)中。

在 Scene Hierarchy View(场景层次视图)中单击“Text Text”,在主窗口右侧的 Property Dock(属性窗口)中找到 String 属性,输入相应文字或字符,例如“X3D 地球”。接下来对文字进行着色,选择 Appearance→material,选择类型为 Material,在主窗口右侧的 Property Dock(属性窗口)中调整 diffuseColor=0 0 1。接着调整文字的大小及外观,在 Scene Hierarchy View(场景层次视图)中选择 Text→fontStyle,选择类型为 FontStyle,在主窗口右侧的 Property Dock(属性窗口)中调整字体的大小和摆放方式,调整属性 size=2,justify=MIDDLE。在 Scene Hierarchy View(场景层次视图)中选择 Transform,在 Property Dock(属性窗口)中找到 translation,设置 X=0,Y=8,Z=0。

创建一个纹理地球,在主菜单中找到 Node List View(节点列表视图),选择 Standard→Primitives→Sphere Primitive 创建一个球体造型,并调整球体半径为 5.0。然后绘制地球纹理,在 Scene Hierarchy View(场景层次视图)中选择 Shape→Appearance→texture,在 Node List View(节点列表视图)中选择 Texturing→Image Texture,双击该选项,在 Scene Hierarchy View(场景层次视图)中选中 texture Image Texture,这时在主窗口右侧的 Property Dock(属性窗口)中显示 url 路径,选择纹理图 D0258.jpg 文件所在的路径,如图 3-9 所示。

在本书附带光盘的 X3D 互动游戏案例源程序/第 3 章案例源程序目录下提供了 X3D 互动游戏源程序 px3d3-5.x3d。

【案例 3-5】 利用 Shape 空间物体造型模型节点、背景节点、基本几何节点、纹理图像以及坐标变换节点等,设置白色的三维立体空间背景,创建一个 3D 文字和一个纹理地球造型。

X3D 互动游戏 3D 文字和纹理地球造型文件 px3d3-5.x3d 的源程序如下:

```

<?xml version = "1.0" encoding = "UTF - 8"?>
<X3D profile = "Immersive" version = "3.0">
    <head>
        <meta content = "px3d3 - 5.x3d" name = "filename"/>
        <meta content = "zjz - zjr - zjd" name = "author"/>
        <meta content = ' * enter FileNameWithNoAbbreviations.x3d here * ' name = 'title'/>
        <meta content = ' * enter description here, short - sentence summaries preferred * ' name =
'description'/>

```

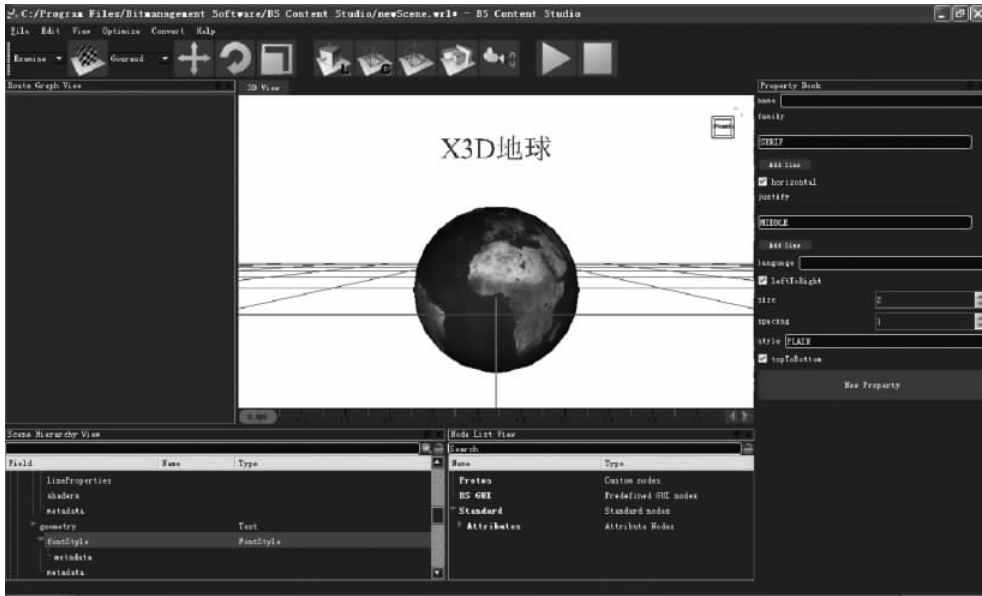


图 3-9 立方体纹理造型设计

```

<meta content = ' * enter version here, if any * ' name = 'version' />
<meta content = ' * enter copyright information here * Example: Copyright (c) Web3D Consortium Inc. 2015' name = 'rights' />
</head>
< Scene>
  < Background DEF = "_1" skyColor = '1 1 1'>
  </Background>
  < WorldInfo DEF = "_2">
  </WorldInfo>
  < NavigationInfo DEF = "_3" type = "EXAMINE", "ANY">
  </NavigationInfo>
  < Transform DEF = "_4" translation = '0 8 0'>
    < Shape DEF = "_5">
      < Appearance DEF = "_6">
        < Material DEF = "_7" diffuseColor = '0 0 1'>
        </Material>
      </Appearance>
      < Text DEF = "_8" string = "X3D&#xe5;&#x9c;&#xb0;&#xe7;&#x90;&#x83;"
solid = 'false'>
        < FontStyle DEF = "_9" justify = "MIDDLE" size = '2'>
        </FontStyle>
      </Text>
    </Shape>
  </Transform>
  < Transform DEF = "_10">
    < Shape DEF = "_11">
      < Appearance DEF = "_12">
        < Material DEF = "_13">
        </Material>
        < ImageTexture DEF = "_14" url = "D0258.jpg" />

```



```

        </ImageTexture >
    </Appearance >
    < Sphere DEF = "_15" radius = '5' >
    </Sphere >
    </Shape >
    </Transform >
</Scene >
</X3D >

```

3.4 X3D 互动游戏路灯造型项目开发案例

利用基本几何体构建一个三维路灯造型,利用 X3D 互动游戏虚拟现实技术在三维立体空间中创建一个逼真、生动、鲜活的 3D 路灯造型;采用软件工程的思想对三维路灯造型项目进行开发与设计,由总体设计、详细设计、编码测试以及运行维护等环节组成;使用坐标变换节点、背景节点、视点节点、模型节点、圆柱体节点、球体节点、外观节点以及材料节点等开发与设计。

3.4.1 X3D 路灯造型项目开发设计

利用几何体节点创建一个 X3D 互动游戏三维路灯造型,X3D 互动游戏三维路灯造型由灯头部分、路灯支架、路灯主干、路灯底座等构成,路灯的灯头由灯罩和灯头两个部分组成,灯罩运用透明技术创建一个透明球形灯罩来衬托三维路灯造型,路灯支架由圆柱体造型组成,路灯主干和底座由圆柱体造型构成,使 X3D 互动游戏三维路灯造型更加逼真和生动,如图 3-10 所示。

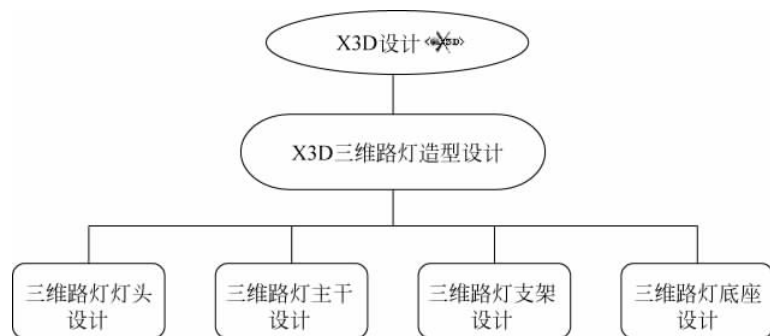


图 3-10 X3D 互动游戏三维路灯造型设计

3.4.2 X3D 路灯造型项目案例

X3D 互动游戏三维路灯造型由灯头部分、路灯支架、路灯主干、路灯底座等构成,利用集成开发环境快速创建一个三维路灯造型。首先启动开发环境,在主菜单中找到 Node List View(节点列表视图),选择 Background→metdata,在主窗口右侧的 Property Dock(属性窗口)中找到 Sky Color 属性,选择天空颜色为白色后单击 OK 按钮确定。

分别创建一个路灯灯头和灯罩,在主菜单中找到 Node List View(节点列表视图),选择 Standard→Primitives→Sphere Primitive 创建一个球体造型,分别调整球体半径为 0.2 和 0.8。然后调整路灯灯头和灯罩的色彩,在 Scene Hierarchy View(场景层次视图)中分别选择灯头和灯罩,选择 Shape→Appearance→material,在主窗口右侧的 Property Dock(属性窗口)中显示各种颜色,选择 diffuseColor="1.0 0.2 0.2"(红色)。

分别创建一个路灯支架、主干和底座,在主菜单中找到 Node List View(节点列表视图),选择 Standard→Primitives→Cylinder Primitive 创建一个圆柱体造型,然后选择 diffuseColor="0.5 0.5 0.7"(灰色),分别对其进行坐标定位、旋转和缩放,设计效果如图 3-11 所示。

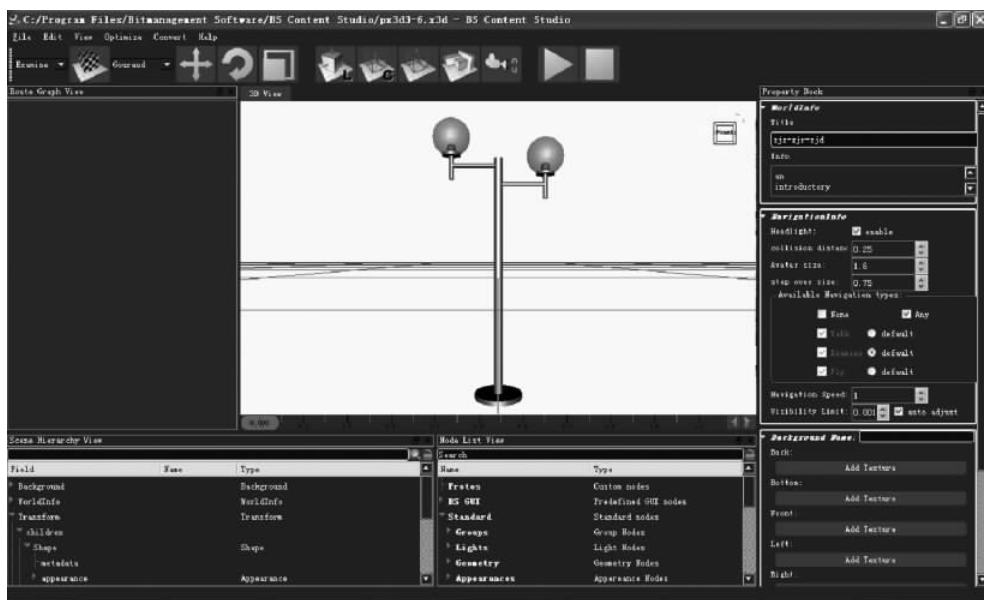


图 3-11 X3D 互动游戏路灯造型设计效果

X3D 三维路灯造型项目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的开发理念;使用 X3D 集成开发环境提高开发效率,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;利用坐标变换节点、模型节点、几何节点创建各种几何体造型,构造一个更加逼真的三维立体路灯造型,虚拟现实路灯三维立体造型利用 X3D 互动游戏虚拟现实程序开发、设计、编码和调试,利用 X3D 互动游戏虚拟现实技术的各种基本节点创建生动、逼真的三维立体路灯造型,使用 X3D 互动游戏节点、坐标变换节点、重定义和使用节点、背景节点、基本几何体节点以及视点节点进行开发、设计、编码和测试。

在本书附带光盘的 X3D 互动游戏案例源程序/第 3 章案例源程序目录下提供了 X3D 互动游戏源程序 px3d3-6. x3d。

【案例 3-6】 利用坐标变换节点、重定义和使用节点、视点节点、背景节点、Shape 空间物体造型模型节点、基本几何体节点等在三维立体空间背景下创建一个逼真、鲜活的路灯三维造型。

X3D 互动游戏三维路灯造型文件 px3d3-6. x3d 的源程序如下:

```
<?xml version = "1.0" encoding = "UTF - 8"?>
```

```

<!DOCTYPE X3D PUBLIC "http://www.web3d.org/specifications/x3d-3.2.dtd"
    "file:///www.web3d.org/TaskGroups/x3d/translation/x3d-3.2.dtd">
<!-- Warning: transitional DOCTYPE in source .x3d file -->
<X3D profile = "Immersive" version = "3.2"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema-instance" xsd:noNamespaceSchemaLocation = "
http://www.web3d.org/specifications/x3d-3.2.xsd">
  <head>
    <meta content = "px3d3-6.x3d" name = "filename"/>
    <meta content = "zjz-zjr-zjd" name = "author"/>
    <meta content = " * enter name of original author here * " name = "creator"/>
    <meta content = " * enter copyright information here * Example: Copyright (c) Web3D
Consortium Inc. 2015" name = "rights"/>
    <meta content = " * enter online Uniform Resource Identifier (URI) or Uniform Resource Locator
(URL) address for this file here * " name = "identifier"/>
    <meta content = "X3D - Edit, http://www.web3d.org/x3d/content/README.X3D - Edit.html" name =
"generator"/>
  </head>
  <Scene>
    <Background skyColor = "0.98 0.98 0.98"/>
    <WorldInfo info = "an introductory scene" title = "zjz-zjr-zjd"/>
    <!-- Viewpoint description = "zjz-zjr-zjd" orientation = "0 0 0" position = "0 1 18"/ -->
    <Transform translation = "0 0.1 0">
      <Shape>
        <Appearance>
          <Material ambientIntensity = "0.4" diffuseColor = "0.5 0.5 0.7"
            shininess = "0.2" specularColor = "0.8 0.8 0.9"/>
        </Appearance>
        <Cylinder bottom = "true" height = "10" radius = "0.2" side = "true" top = "true"/>
      </Shape>
    </Transform>
    <Transform translation = "-1 4.8 0" rotation = "0 0 1 1.571">
      <Shape>
        <Appearance>
          <Material ambientIntensity = "0.4" diffuseColor = "0.5 0.5 0.7"
            shininess = "0.2" specularColor = "0.8 0.8 0.9"/>
        </Appearance>
        <Cylinder bottom = "true" height = "2" radius = "0.1" side = "true" top = "true"/>
      </Shape>
    </Transform>
    <Transform translation = "1 4 0" rotation = "0 0 1 1.571">
      <Shape>
        <Appearance>
          <Material ambientIntensity = "0.4" diffuseColor = "0.5 0.5 0.7"
            shininess = "0.2" specularColor = "0.8 0.8 0.9"/>
        </Appearance>
        <Cylinder bottom = "true" height = "2" radius = "0.1" side = "true" top = "true"/>
      </Shape>
    </Transform>
    <Transform translation = "0 -5 0">
      <Shape>
        <Appearance>
          <Material ambientIntensity = "0.4" diffuseColor = "0.5 0.5 0.7"

```

```

        shininess = "0.2" specularColor = "0.8 0.8 0.9"/>
    </Appearance>
    <Cylinder bottom = "true" height = "0.2" radius = "1" side = "true" top = "true"/>
</Shape>
</Transform>
<Group DEF = 'ld1'>
<Transform translation = "- 2 6 0">
    <Shape>
        <Appearance>
            <Material ambientIntensity = "0.1" diffuseColor = "1.0 0.2 0.2"
                shininess = "0.15" specularColor = "0.8 0.8 0.8" transparency = "0.5"/>
        </Appearance>
        <Sphere radius = "0.8"/>
    </Shape>
</Transform>
<Transform translation = "- 2 5.52 0">
    <Shape>
        <Appearance>
            <Material ambientIntensity = "0.1" diffuseColor = "1.0 0.2 0.2"
                shininess = "0.15" specularColor = "0.8 0.8 0.8" />
        </Appearance>
        <Sphere radius = "0.2"/>
    </Shape>
</Transform>
<Transform translation = "- 2 5.25 0">
    <Shape>
        <Appearance>
            <Material ambientIntensity = "0.4" diffuseColor = "0.5 0.5 0.7"
                shininess = "0.2" specularColor = "0.8 0.8 0.9"/>
        </Appearance>
        <Cylinder bottom = "true" height = "0.10" radius = "0.4" side = "true" top = "true"/>
    </Shape>
</Transform>
<Transform translation = "- 2 4.8 0">
    <Shape>
        <Appearance>
            <Material ambientIntensity = "0.4" diffuseColor = "0.5 0.5 0.7"
                shininess = "0.2" specularColor = "0.8 0.8 0.9"/>
        </Appearance>
        <Cylinder bottom = "true" height = "1.2" radius = "0.1" side = "true" top = "true"/>
    </Shape>
</Transform>
</Group>
<Transform translation = "4 - 0.8 0">
    <Group USE = 'ld1' />
</Transform>
</Scene>
</X3D>

```