**海南大学计算机科学与技术学院**

**《计算机图形学》课内实验报告一**

****

**班 级：计算机科学与技术2021-3班**

**成 员： 李季鸿 20213002624**

**报告名称： 实验报告一**

**指导老师： 高新瑞**

**完成日期： 2023年9月14日**

**《计算机图形学》实验报告一**

**学生姓名：李季鸿 班级：21级计科3班 学号：20213002624**

**实验地点：9-202 指导教师：高新瑞**

**实验日期：**2023.9.14  **实验课时：2学时**

**实验环境：**Windows 10+JDK1.8+记事本+IntelliJ IDEA

**一、实验目的**

1.了解并熟练掌握java3D编程

2.了解Java3D基本图形功能

3.了解并熟悉Java3D的场景式管理

4.通过球体和长方体组合体程序对Java场景式管理实践

**二、实验过程**

## 实验内容一：

（1）代码

1. *//加入程序运行时需要的Java、Java 3D的包*
2. import com.sun.j3d.utils.applet.MainFrame;
3. import com.sun.j3d.utils.behaviors.mouse.MouseRotate;
4. import com.sun.j3d.utils.behaviors.mouse.MouseTranslate;
5. import com.sun.j3d.utils.behaviors.mouse.MouseZoom;
6. import com.sun.j3d.utils.geometry.Box;
7. import com.sun.j3d.utils.geometry.Sphere;
8. import com.sun.j3d.utils.universe.SimpleUniverse;
9. import javax.media.j3d.\*;
10. import javax.vecmath.Color3f;
11. import javax.vecmath.Point3d;
12. import javax.vecmath.Vector3f;
13. import java.applet.Applet;
14. import java.awt.\*;
15. public class twoprimitivedisplay extends Applet {
16. public twoprimitivedisplay() {*//设置显示界面的相关参数*
17. setLayout(new BorderLayout());
18. *//创建投影平面Canvas3D*
19. GraphicsConfiguration gc = SimpleUniverse.getPreferredConfiguration();
20. Canvas3D c = new Canvas3D(gc);
21. *//将投影平面上的图象显示在显示平面的中间*
22. add("Center", c);
23. *//设置SimpleUniverse，由系统选择视点在z轴的正向，观察方向沿z轴反向*
24. BranchGroup BranchGroupScene = createBranchGroup();
25. SimpleUniverse u = new SimpleUniverse(c);
26. u.getViewingPlatform().setNominalViewingTransform();
27. *//将BranchGroup：BranchGroupScene加入到SimpleUniverse：u中*
28. u.addBranchGraph(BranchGroupScene);
29. }
30. public static void main(String[] args) {*//通过MainFrame显示图象*
31. new MainFrame(new twoprimitivedisplay(), 300, 300);
32. }
33. public BranchGroup createBranchGroup() {*//定义BranchGroup*
34. BranchGroup BranchGroupRoot = new BranchGroup();
35. *//创建球心在坐标系原点球形范围*
36. BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);
37. *//定义背景颜色*
38. Color3f bgColor = new Color3f(1.0f, 1.0f, 1.0f);
39. Background bg = new Background(bgColor);
40. bg.setApplicationBounds(bounds);
41. BranchGroupRoot.addChild(bg);
42. *//定义平行光、颜色、照射方向与作用范围*
43. Color3f directionalColor = new Color3f(1.f, 1.f, 1.f);
44. Vector3f vec = new Vector3f(-1.f, -1.f, -1.0f);
45. DirectionalLight directionalLight = new DirectionalLight(directionalColor, vec);
46. directionalLight.setInfluencingBounds(bounds);
47. BranchGroupRoot.addChild(directionalLight);
48. *//定义两个三维型体的外观*
49. Appearance app1 = new Appearance();
50. Material material1 = new Material();
51. material1.setDiffuseColor(new Color3f(1.0f, .0f, 0.0f));
52. app1.setMaterial(material1);
53. Appearance app2 = new Appearance();
54. Material material2 = new Material();
55. material2.setDiffuseColor(new Color3f(.0f, 1.0f, 0.0f));
56. app2.setMaterial(material2);
57. *//定义总的TransformGroup：transformgroup*
58. TransformGroup transformgroup = new TransformGroup();
59. *//设置对该TransformGroup的读写能力*
60. transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);
61. transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_READ);
62. *//将该TransformGroup加入到BranchGroupRoot中*
63. BranchGroupRoot.addChild(transformgroup);
64. *//定义鼠标对场景的旋转、平移与放大功能*
65. MouseRotate mouserotate = new MouseRotate();
66. mouserotate.setTransformGroup(transformgroup);
67. BranchGroupRoot.addChild(mouserotate);
68. mouserotate.setSchedulingBounds(bounds);
69. MouseZoom mousezoom = new MouseZoom();
70. mousezoom.setTransformGroup(transformgroup);
71. BranchGroupRoot.addChild(mousezoom);
72. mousezoom.setSchedulingBounds(bounds);
73. MouseTranslate mousetranslate = new MouseTranslate();
74. mousetranslate.setTransformGroup(transformgroup);
75. BranchGroupRoot.addChild(mousetranslate);
76. mousetranslate.setSchedulingBounds(bounds);
77. */\*定义一个球体与一个长方体的大小、外观属性与坐标变换，并定义相应的TransformGroup：tg1、tg2\*/*
78. TransformGroup tg1 = new TransformGroup();
79. tg1.addChild(new Sphere(0.4f, app1));
80. Transform3D t = new Transform3D();
81. t.setTranslation(new Vector3f(0.f, -0.425f, 0.f));
82. TransformGroup tg2 = new TransformGroup(t);
83. tg2.addChild(new Box(0.5f, 0.05f, 0.5f, app2));
84. *//将定义好的两个TransformGroup(tg1、tg2)加入到总的transformgroup*
85. transformgroup.addChild(tg1);
86. transformgroup.addChild(tg2);
87. *//对BranchGroupRoot预编译*
88. BranchGroupRoot.compile();
89. *//通过方法名返回BranchGroupRoot*
90. return BranchGroupRoot;
91. }
92. }

（2）结果截图

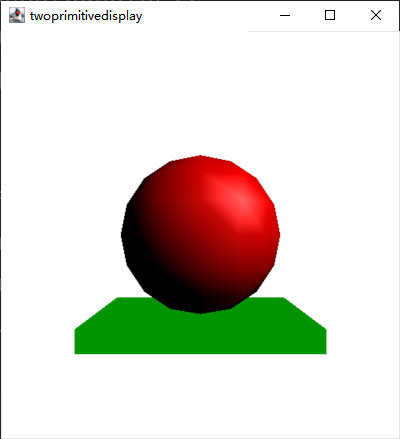


图1- 1 运行结果截图

## 实验内容二：

在上述基础上修改了代码：

（一）代码：

1. import com.sun.j3d.utils.applet.MainFrame;
2. import com.sun.j3d.utils.behaviors.mouse.MouseRotate;
3. import com.sun.j3d.utils.behaviors.mouse.MouseTranslate;
4. import com.sun.j3d.utils.behaviors.mouse.MouseWheelZoom;
5. import com.sun.j3d.utils.behaviors.mouse.MouseZoom;
6. import com.sun.j3d.utils.geometry.Box;
7. import com.sun.j3d.utils.geometry.Sphere;
8. import com.sun.j3d.utils.universe.SimpleUniverse;
9. import javax.media.j3d.\*;
10. import javax.vecmath.Color3f;
11. import javax.vecmath.Point3d;
12. import javax.vecmath.Vector3f;
13. import java.applet.Applet;
14. import java.awt.\*;
15. import java.awt.event.MouseEvent;
16. public class twoprimitivedisplay extends Applet {   *//第一次计算机图形学实验 2023-09-14晚上9-10节课*
17. public twoprimitivedisplay() {*//设置显示界面的相关参数*
18. setLayout(new BorderLayout());*//创建投影平面Canvas3D*
19. GraphicsConfiguration gc = SimpleUniverse.getPreferredConfiguration();
20. Canvas3D c = new Canvas3D(gc);
21. *//将投影平面上的图象显示在显示平面的中间*
22. add("Center", c);
23. *//设置SimpleUniverse，由系统选择视点在z轴的正向，观察方向沿z轴反向*
24. BranchGroup BranchGroupScene = createBranchGroup();
25. SimpleUniverse u = new SimpleUniverse(c);
26. u.getViewingPlatform().setNominalViewingTransform();*//将BranchGroup：BranchGroupScene加入到SimpleUniverse：u中*
27. u.addBranchGraph(BranchGroupScene);
28. }
29. public static void main(String[] args) {*//通过MainFrame显示图象*
30. new MainFrame(new twoprimitivedisplay(), 400, 400);
31. }
32. public BranchGroup createBranchGroup() {*//定义BranchGroup*
33. BranchGroup BranchGroupRoot = new BranchGroup();
34. *//        TransformGroup sceneTG=null;*
35. *//        BranchGroup scenceBranchGroupRoot =null;*
36. *//创建球心在坐标系原点球形范围*
37. BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);
38. *//定义背景颜色*
39. Color3f bgColor = new Color3f(1.0f, 1.0f, 1.0f);
40. Background bg = new Background(bgColor);
41. bg.setApplicationBounds(bounds);
42. BranchGroupRoot.addChild(bg);
43. *//定义平行光、颜色、照射方向与作用范围*
44. Color3f directionalColor = new Color3f(1.f, 1.f, 1.f);
45. Vector3f vec = new Vector3f(-1.f, -1.f, -1.0f);
46. DirectionalLight directionalLight = new DirectionalLight(directionalColor, vec);
47. directionalLight.setInfluencingBounds(bounds);
48. BranchGroupRoot.addChild(directionalLight);
49. *//定义两个三维型体的外观*
50. *//        Appearance app1 = new Appearance();*
51. *//        Material material1 = new Material();*
52. *//        material1.setDiffuseColor(new Color3f(0.0f, .0f, 1.0f));*
53. *//        app1.setMaterial(material1);*
54. *// 创建具有光滑表面属性的 Appearance*
55. Appearance app1 = new Appearance();
56. *// 设置高光反射属性*
57. Material material1 = new Material();
58. material1.setDiffuseColor(new Color3f(0.0f, 0.0f, 1.0f)); *// 设置漫反射颜色*
59. material1.setSpecularColor(new Color3f(1.0f, 1.0f, 1.0f)); *// 设置高光反射颜色*
60. material1.setShininess(2500.0f); *// 设置光滑度，值越高越光滑*
61. app1.setMaterial(material1);
62. *//        Appearance app2 = new Appearance();*
63. *//        Material material2 = new Material();*
64. *//        material2.setDiffuseColor(new Color3f(.0f, 0.0f, 0.0f));*
65. *//        app2.setMaterial(material2);*
66. *// 创建具有透明度的 Appearance*
67. Appearance app2 = new Appearance();
68. TransparencyAttributes transparency = new TransparencyAttributes();
69. transparency.setTransparencyMode(TransparencyAttributes.BLENDED);
70. transparency.setTransparency(0.5f); *// 设置透明度，范围从 0.0（完全不透明）到 1.0（完全透明）*
71. app2.setTransparencyAttributes(transparency);
72. *// 其他属性设置*
73. Material material2 = new Material();
74. material2.setDiffuseColor(new Color3f(0.0f, 1.0f, 0.0f));
75. app2.setMaterial(material2);
76. *//定义总的TransformGroup：transformgroup*
77. TransformGroup transformgroup = new TransformGroup();
78. *//设置对该TransformGroup的读写能力*
79. transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);
80. transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_READ);
81. *//将该TransformGroup加入到BranchGroupRoot中*
82. BranchGroupRoot.addChild(transformgroup);
83. *//定义鼠标对场景的旋转、平移与放大功能*
84. *//鼠标旋转功能*
85. MouseRotate mouserotate = new MouseRotate();
86. mouserotate.setTransformGroup(transformgroup);
87. BranchGroupRoot.addChild(mouserotate);
88. mouserotate.setSchedulingBounds(bounds);
89. *//这是之前的不能进行缩放*
90. MouseZoom mousezoom = new MouseZoom();
91. mousezoom.setTransformGroup(transformgroup);
92. BranchGroupRoot.addChild(mousezoom);
93. mousezoom.setSchedulingBounds(bounds);
94. *//重新定义鼠标缩放功能 默认滚轮*
95. MouseWheelZoom mouseWheelZoom = new MouseWheelZoom();
96. mouseWheelZoom.setTransformGroup(transformgroup);
97. BranchGroupRoot.addChild(mouseWheelZoom);
98. mouseWheelZoom.setSchedulingBounds(bounds);
99. *//鼠标平移功能 默认鼠标右键*
100. MouseTranslate mousetranslate = new MouseTranslate();
101. mousetranslate.setTransformGroup(transformgroup);
102. BranchGroupRoot.addChild(mousetranslate);
103. mousetranslate.setSchedulingBounds(bounds);
104. *//        /\*定义一个球体与一个长方体的大小、外观属性与坐标变换，并定义相应的TransformGroup：tg1、tg2\*/*
105. *//        TransformGroup tg1 = new TransformGroup();*
106. *//        tg1.addChild(new Sphere(0.4f, app1));*
107. *//        Transform3D t = new Transform3D();*
108. *//        t.setTranslation(new Vector3f(0.f, -0.425f, 0.f));*
109. *//        TransformGroup tg2 = new TransformGroup(t);*
110. *//        tg2.addChild(new Box(0.5f, 0.005f, 0.5f, app2));*
111. *//*
112. *//        //将定义好的两个TransformGroup(tg1、tg2)加入到总的transformgroup*
113. *//        transformgroup.addChild(tg1);*
114. *//        transformgroup.addChild(tg2);*
115. *//*
116. *//        //对BranchGroupRoot预编译*
117. *//        BranchGroupRoot.compile();*
118. */\*定义一个球体与一个长方体的大小、外观属性与坐标变换，并定义相应的TransformGroup：tg1、tg2\*/*
119. TransformGroup tg1 = new TransformGroup();
120. tg1.addChild(new Sphere(0.4f, app1));
121. *// 创建上方的长方体*
122. TransformGroup tg2Above = new TransformGroup();
123. Transform3D tAbove = new Transform3D();
124. tAbove.setTranslation(new Vector3f(0.f, 0.5f, 0.f)); *// 将长方体放在球的上方*
125. TransformGroup tg2 = new TransformGroup(tAbove);
126. tg2.addChild(new Box(0.5f, 0.005f, 0.5f, app2));
127. *// 创建下方的长方体*
128. TransformGroup tg2Below = new TransformGroup();
129. Transform3D tBelow = new Transform3D();
130. tBelow.setTranslation(new Vector3f(0.f, -0.5f, 0.f)); *// 将长方体放在球的下方*
131. TransformGroup tg3 = new TransformGroup(tBelow);
132. tg3.addChild(new Box(0.5f, 0.005f, 0.5f, app2));
133. *// 创建前方的长方体*
134. TransformGroup tg2Front = new TransformGroup();
135. Transform3D tFront = new Transform3D();
136. tFront.setTranslation(new Vector3f(0.f, 0.f, -0.5f)); *// 将长方体放在球的前方*
137. TransformGroup tg4 = new TransformGroup(tFront);
138. tg4.addChild(new Box(0.5f, 0.5f, 0.005f, app2));
139. *// 创建后方的长方体*
140. TransformGroup tg2Back = new TransformGroup();
141. Transform3D tBack = new Transform3D();
142. tBack.setTranslation(new Vector3f(0.f, 0.f, 0.5f)); *// 将长方体放在球的后方*
143. TransformGroup tg5 = new TransformGroup(tBack);
144. tg5.addChild(new Box(0.5f, 0.5f, 0.005f, app2));
145. *// 创建左边的长方体*
146. TransformGroup tg2Left = new TransformGroup();
147. Transform3D tLeft = new Transform3D();
148. tLeft.setTranslation(new Vector3f(-0.5f, 0.f, 0.f)); *// 将长方体放在球的左边*
149. TransformGroup tg6 = new TransformGroup(tLeft);
150. tg6.addChild(new Box(0.005f, 0.5f, 0.5f, app2));
151. *// 创建右边的长方体*
152. TransformGroup tg2Right = new TransformGroup();
153. Transform3D tRight = new Transform3D();
154. tRight.setTranslation(new Vector3f(0.5f, 0.f, 0.f)); *// 将长方体放在球的右边*
155. TransformGroup tg7 = new TransformGroup(tRight);
156. tg7.addChild(new Box(0.005f, 0.5f, 0.5f, app2));
157. *// 将定义好的所有TransformGroup加入到总的transformgroup*
158. transformgroup.addChild(tg1);
159. transformgroup.addChild(tg2);
160. transformgroup.addChild(tg3);
161. transformgroup.addChild(tg4);
162. transformgroup.addChild(tg5);
163. transformgroup.addChild(tg6);
164. *//        transformgroup.addChild(tg7);*
165. *// 对BranchGroupRoot预编译*
166. BranchGroupRoot.compile();
167. *//通过方法名返回BranchGroupRoot*
168. return BranchGroupRoot;
169. }
170. }

（二）截图：

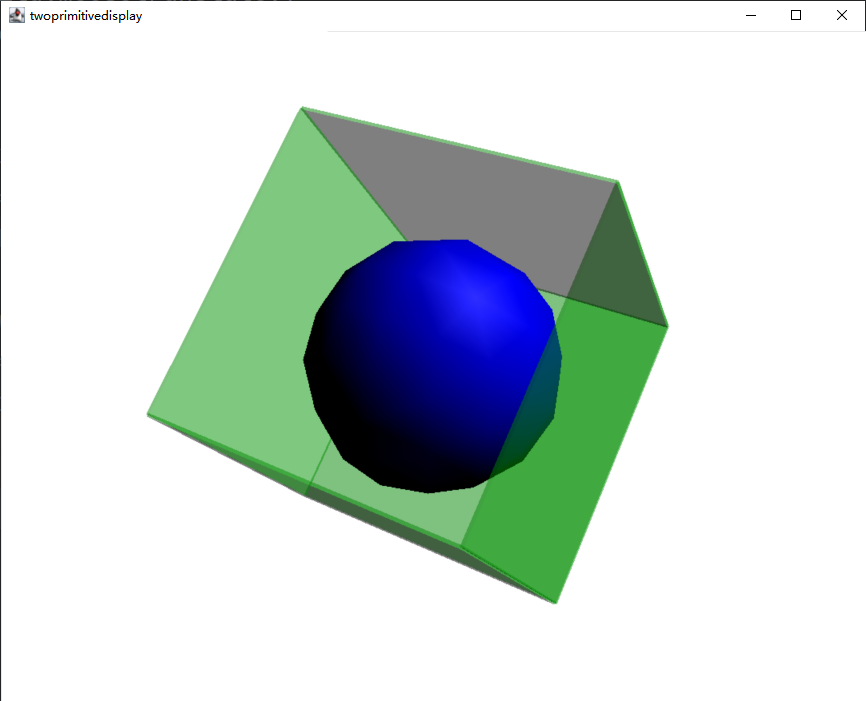


图1- 2 运行结果截图

**三、实验总结**

本次实验是关于使用Java 3D库创建三维图形的实践，主要集中在对球形和长方体组成的组合体进行可视化的编程实现。以下是对本次实验的总结：

1. 实验目标和内容：

本次实验的主要目标是使用Java 3D库创建一个简单的三维场景，其中包括球体和长方体，并添加鼠标交互功能，以便用户可以旋转、平移和缩放场景中的物体。实验涉及了多个关键类，如BranchGroup、TransformGroup、Appearance等。

2. 场景构建：

- 使用BranchGroup作为场景的容器，用于组织和管理三维元素。

- 定义了一个BoundingSphere，用于指定场景的范围。

- 设置背景颜色，为场景提供了背景。

- 添加了平行光源，定义了光源的颜色和照射方向，以影响物体的外观。

3. 三维物体：

- 使用Sphere和Box类创建了一个球体和一个长方体。

- 通过Appearance类定义了物体的外观属性，如颜色。

- 使用TransformGroup类对物体进行了坐标变换，以确定它们的位置和大小。

4. 鼠标交互功能：

- 使用MouseRotate、MouseTranslate和MouseZoom类添加了鼠标交互功能，允许用户通过鼠标操作来改变场景中的视角和物体位置。

1. 界面设置：

- 在twoprimitivedisplay构造函数中设置了显示界面的相关参数，包括布局和Canvas3D的创建。

- 通过SimpleUniverse管理三维场景，并设置默认视图变换。

6. 总结：

通过本次实验，我深入理解了Java 3D编程的基本原理和使用方法。学会了如何创建和配置三维场景，如何定义物体的外观和坐标变换，以及如何添加交互功能以提供用户友好的界面。这个实验为进一步探索三维图形编程和应用提供了坚实的基础。

7. 学习收获：

本次实验让我更加熟悉了Java 3D编程，特别是在构建三维场景和处理交互方面。这对于未来可能涉及到三维可视化的项目或研究非常有用。同时，我也锻炼了代码编写和问题解决的能力。

在未来的学习和实践中，我可以进一步扩展这个基础，创建更复杂的三维场景，并探索更多Java 3D库的功能和特性，以满足不同应用的需求。