**海南大学计算机科学与技术学院**

**《计算机图形学》课内实验报告四**

****

**班 级：计算机科学与技术2021-3班**

**成 员： 李季鸿 20213002624**

**报告名称： 实验报告四**

**指导老师： 高新瑞**

**完成日期： 2023年10月12日**

**《计算机图形学》实验报告四**

**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：**2023.10.12  **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**

**实验一目的：**

1.学习如何使用Java 3D库创建三维图形应用程序。

2.理解如何创建和配置3D场景，包括添加背景、光照和交互性。

3.熟悉三维图形中的基本概念，如顶点、颜色、光照和多边形渲染。

4.实践创建和渲染基本的三维几何形状，如三角形。

**实验二目的：**

学习如何使用Java 3D库创建三维图形应用程序。

理解如何创建和配置3D场景，包括添加背景、光照和交互性。

熟悉三维图形中的基本概念，如顶点、颜色、光照和多边形渲染。

实践创建和渲染基本的三维几何形状，如三角带。

**实验三目的：**

学习如何使用Java 3D库创建三维图形应用程序。

理解如何创建和配置3D场景，包括添加背景、光照和交互性。

熟悉三维图形中的基本概念，如顶点、颜色、光照和多边形渲染。

实践创建和渲染基本的三维几何形状，如三角扇。

**二、实验过程**

## 实验内容一：

（1）代码

/\*\*  
 \* \\* Created with IntelliJ IDEA.  
 \* \\* @ProjectName: 例3.7 TriangleArray类程序实例  
 \* \\* @FileName: DisplayTriangles  
 \* \\* @author: li-jihong  
 \* \\* Date: 2023-09-21 14:14  
 \*/  
  
import com.sun.j3d.utils.applet.MainFrame;  
import com.sun.j3d.utils.behaviors.mouse.MouseRotate;  
import com.sun.j3d.utils.behaviors.mouse.MouseTranslate;  
import com.sun.j3d.utils.behaviors.mouse.MouseZoom;  
import com.sun.j3d.utils.universe.SimpleUniverse;  
  
import javax.media.j3d.\*;  
import javax.vecmath.Color3f;  
import javax.vecmath.Point3d;  
import javax.vecmath.Vector3f;  
import java.applet.Applet;  
import java.awt.\*;  
  
public class DisplayTriangles extends Applet {  
 public DisplayTriangles() {  
 setLayout(new BorderLayout());  
 GraphicsConfiguration gc = SimpleUniverse.getPreferredConfiguration();  
 Canvas3D c = new Canvas3D(gc);  
 add("Center", c);  
 BranchGroup BranchGroupScene = createBranchGroup();  
 SimpleUniverse u = new SimpleUniverse(c);  
 u.getViewingPlatform().setNominalViewingTransform();  
 u.addBranchGraph(BranchGroupScene);  
 }  
  
 public static void main(String[] args) {  
 new MainFrame(new DisplayTriangles(), 450, 450);  
 }  
  
 public BranchGroup createBranchGroup() {  
 BranchGroup BranchGroupRoot = new BranchGroup();  
 BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);  
 Color3f bgColor = new Color3f(1.0f, 1.0f, 1.0f);  
 Background bg = new Background(bgColor);  
 bg.setApplicationBounds(bounds);  
 BranchGroupRoot.addChild(bg);  
 Color3f directionalColor = new Color3f(1.f, 0.f, 0.f);  
 Vector3f vec = new Vector3f(0.f, 0.f, -1.0f);  
 DirectionalLight directionalLight = new DirectionalLight(directionalColor, vec);  
 directionalLight.setInfluencingBounds(bounds);  
 BranchGroupRoot.addChild(directionalLight);  
 TransformGroup transformgroup = new TransformGroup();  
 transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  
 transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_READ);  
 BranchGroupRoot.addChild(transformgroup);  
 MouseRotate mouserotate = new MouseRotate();  
 mouserotate.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mouserotate);  
 mouserotate.setSchedulingBounds(bounds);  
 MouseZoom mousezoom = new MouseZoom();  
 mousezoom.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mousezoom);  
 mousezoom.setSchedulingBounds(bounds);  
 MouseTranslate mousetranslate = new MouseTranslate();  
 mousetranslate.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mousetranslate);  
 mousetranslate.setSchedulingBounds(bounds);  
 transformgroup.addChild(new TriangleArrays());  
 BranchGroupRoot.compile();  
 return BranchGroupRoot;  
 }  
}  
  
class TriangleArrays extends Shape3D {  
 public TriangleArrays() {  
 int vCount = 12;  
 float vertexes[] = {-0.6f, 0.9f, 0.0f, -0.6f, -0.9f, 0.2f,  
 -0.4f, 0.9f, -0.2f, -0.2f, -0.9f, 0.2f,  
 0.0f, 0.9f, -0.2f, 0.0f, -0.9f, 0.2f,  
 0.2f, 0.7f, 0.0f, 0.2f, -0.9f, 0.3f,  
 0.5f, 0.8f, -0.3f, 0.6f, -.9f, 0.0f,  
 0.8f, 0.9f, 0.2f, 0.8f, -0.8f, 0.3f};  
 float colors[] = {0.0f, 0.5f, 1.0f, 0.0f, 0.5f, 1.0f,  
 0.0f, 0.8f, .0f, 1.0f, 0.0f, 0.3f,  
 0.0f, 1.0f, 0.5f, 0.9f, 1.0f, 0.0f,  
 0.5f, 0.0f, 1.0f, 0.0f, 0.5f, 1.0f,  
 1.0f, 0.5f, 0.0f, 1.0f, 0.0f, 0.5f,  
 1.0f, 0.8f, 0.0f, 1.0f, 0.5f, 0.0f};  
 TriangleArray trianglearray = new TriangleArray  
 (vCount, TriangleArray.COORDINATES | TriangleArray.COLOR\_3);  
 trianglearray.setCoordinates(0, vertexes);  
 trianglearray.setColors(0, colors);  
 PolygonAttributes polygonattributes = new PolygonAttributes();  
// polygonattributes.setCullFace(PolygonAttributes.CULL\_NONE);  
// polygonattributes.setCullFace(PolygonAttributes.CULL\_FRONT);  
 polygonattributes.setCullFace(PolygonAttributes.CULL\_BACK);  
 Appearance app = new Appearance();  
 app.setPolygonAttributes(polygonattributes);  
 this.setGeometry(trianglearray);  
 this.setAppearance(app);  
 }  
}

（2）结果截图

|  |  |  |
| --- | --- | --- |
|  |  |  |
| CULL\_NONE | CULL\_FRONT | CULL\_BACK |

## 实验内容二：

（1）代码

package week7\_fourth.\_3\_8; /\*\*  
 \* \\* Created with IntelliJ IDEA.  
 \* \\* @ProjectName: 例3.8 TriangleStripArray类程序实例  
 \* \\* @FileName: week7\_fourth.\_3\_8.TriangleStripArrays  
 \* \\* @author: li-jihong  
 \* \\* Date: 2023-09-21 14:15  
 \*/  
  
import com.sun.j3d.utils.applet.MainFrame;  
import com.sun.j3d.utils.behaviors.mouse.MouseRotate;  
import com.sun.j3d.utils.behaviors.mouse.MouseTranslate;  
import com.sun.j3d.utils.behaviors.mouse.MouseZoom;  
import com.sun.j3d.utils.universe.SimpleUniverse;  
  
import javax.media.j3d.\*;  
import javax.vecmath.Color3f;  
import javax.vecmath.Point3d;  
import javax.vecmath.Vector3f;  
import java.applet.Applet;  
import java.awt.\*;  
  
public class TriangleStripArrays extends Applet {  
 public TriangleStripArrays() {  
 setLayout(new BorderLayout());  
 GraphicsConfiguration gc = SimpleUniverse.getPreferredConfiguration();  
 Canvas3D c = new Canvas3D(gc);  
 add("Center", c);  
 BranchGroup BranchGroupScene = createBranchGroup();  
 SimpleUniverse u = new SimpleUniverse(c);  
 u.getViewingPlatform().setNominalViewingTransform();  
 u.addBranchGraph(BranchGroupScene);  
 }  
  
 public static void main(String[] args) {  
 new MainFrame(new TriangleStripArrays(), 450, 450);  
 }  
  
 public BranchGroup createBranchGroup() {  
 BranchGroup BranchGroupRoot = new BranchGroup();  
 BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);  
 Color3f bgColor = new Color3f(1.0f, 1.0f, 1.0f);  
 Background bg = new Background(bgColor);  
 bg.setApplicationBounds(bounds);  
 BranchGroupRoot.addChild(bg);  
 Color3f directionalColor = new Color3f(1.f, 0.f, 0.f);  
 Vector3f vec = new Vector3f(0.f, 0.f, -1.0f);  
 DirectionalLight directionalLight = new DirectionalLight(directionalColor, vec);  
 directionalLight.setInfluencingBounds(bounds);  
 BranchGroupRoot.addChild(directionalLight);  
 TransformGroup transformgroup = new TransformGroup();  
 transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  
 transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_READ);  
 BranchGroupRoot.addChild(transformgroup);  
 MouseRotate mouserotate = new MouseRotate();  
 mouserotate.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mouserotate);  
 mouserotate.setSchedulingBounds(bounds);  
 MouseZoom mousezoom = new MouseZoom();  
 mousezoom.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mousezoom);  
 mousezoom.setSchedulingBounds(bounds);  
 MouseTranslate mousetranslate = new MouseTranslate();  
 mousetranslate.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mousetranslate);  
 mousetranslate.setSchedulingBounds(bounds);  
 transformgroup.addChild(new TriangleStrip());  
 BranchGroupRoot.compile();  
 return BranchGroupRoot;  
 }  
}  
  
class TriangleStrip extends Shape3D {  
 public TriangleStrip() {  
 int vertexesCount = 12;  
 int stripCount[] = new int[1];  
 float vertexes[] = {-0.9f, 0.9f, 0.0f, -0.8f, -0.9f, 0.2f,  
 -0.6f, 0.8f, -0.2f, -0.4f, -0.8f, 0.2f,  
 -0.3f, 0.9f, -0.2f, -0.2f, -0.9f, 0.2f,  
 0.4f, 0.7f, 0.0f, 0.4f, -0.7f, 0.3f,  
 0.6f, 0.9f, -0.3f, 0.6f, -.9f, 0.0f,  
 0.9f, 0.8f, 0.2f, 0.8f, -0.8f, 0.3f};  
 float colors[] = {0.0f, 0.5f, 1.0f, 0.0f, 0.5f, 1.0f,  
 0.0f, 0.8f, .0f, 1.0f, 0.0f, 0.3f,  
 0.0f, 1.0f, 0.5f, 0.9f, 1.0f, 0.0f,  
 0.5f, 0.0f, 1.0f, 0.0f, 0.5f, 1.0f,  
 1.0f, 0.5f, 0.0f, 1.0f, 0.0f, 0.5f,  
 1.0f, 0.8f, 0.0f, 1.0f, 0.5f, 0.0f};  
 //生成三个子Strip，每个含有四个顶点  
 stripCount[0] = 12;  
// stripCount[1] = 6;  
// stripCount[2] = 3;  
 TriangleStripArray triangleStriparray = new TriangleStripArray(vertexesCount,  
 TriangleStripArray.COORDINATES | TriangleStripArray.COLOR\_3, stripCount);  
 triangleStriparray.setCoordinates(0, vertexes);  
 triangleStriparray.setColors(0, colors);  
 PolygonAttributes polygonattributes = new PolygonAttributes();  
 polygonattributes.setCullFace(PolygonAttributes.CULL\_NONE);  
 Appearance app = new Appearance();  
 app.setPolygonAttributes(polygonattributes);  
 this.setGeometry(triangleStriparray);  
 this.setAppearance(app);  
 }  
}

（2）结果截图

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 三个Strip，每个含有顶点数目（4,4,4） | 三个Strip，每个含有顶点数目（3,6,3） | 三个Strip，每个含有顶点数目（3,4,5） |  |
|  |  |  |  |
| 两个Strip，每个含有顶点数目（3,9） | 三个Strip，每个含有顶点数目（4,8） | 三个Strip，每个含有顶点数目（5,7） | 三个Strip，每个含有顶点数目（6,6） |
|  | | | |
| 一个Strip，每个含有顶点数目（12） | | | |

## 实验内容三：

（1）代码

package week7\_fourth.\_3\_9;  
  
import com.sun.j3d.utils.applet.MainFrame;  
import com.sun.j3d.utils.behaviors.mouse.MouseRotate;  
import com.sun.j3d.utils.behaviors.mouse.MouseTranslate;  
import com.sun.j3d.utils.behaviors.mouse.MouseZoom;  
import com.sun.j3d.utils.universe.SimpleUniverse;  
  
import javax.media.j3d.\*;  
import javax.vecmath.\*;  
import java.applet.Applet;  
import java.awt.\*;  
  
/\*\*  
 \* \\* Created with IntelliJ IDEA.  
 \* \\* @ProjectName: 例3.9 TriangleFanArray类程序实例  
 \* \\* @FileName: week7\_fourth.\_3\_9.TriangleFanArray  
 \* \\* @author: li-jihong  
 \* \\* Date: 2023-10-12 20:11  
 \*/  
  
public class TriangleFanArray2 extends Applet {  
 public TriangleFanArray2() {  
 setLayout(new BorderLayout());  
 GraphicsConfiguration gc = SimpleUniverse.getPreferredConfiguration();  
 Canvas3D c = new Canvas3D(gc);  
 add("Center", c);  
 BranchGroup BranchGroupScene = createBranchGroup();  
 SimpleUniverse u = new SimpleUniverse(c);  
 u.getViewingPlatform().setNominalViewingTransform();  
 u.addBranchGraph(BranchGroupScene);  
 }  
  
 public static void main(String[] args) {  
 new MainFrame(new TriangleFanArray2(), 450, 450);  
 }  
  
 public BranchGroup createBranchGroup() {  
 BranchGroup BranchGroupRoot = new BranchGroup();  
 BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);  
 Color3f bgColor = new Color3f(1.0f, 1.0f, 1.0f);  
 Background bg = new Background(bgColor);  
 bg.setApplicationBounds(bounds);  
 BranchGroupRoot.addChild(bg);  
 Color3f directionalColor = new Color3f(1.f, 0.f, 0.f);  
 Vector3f vec = new Vector3f(0.f, 0.f, -1.0f);  
 DirectionalLight directionalLight = new DirectionalLight(directionalColor, vec);  
 directionalLight.setInfluencingBounds(bounds);  
 BranchGroupRoot.addChild(directionalLight);  
 TransformGroup transformgroup = new TransformGroup();  
 transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  
 transformgroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_READ);  
 BranchGroupRoot.addChild(transformgroup);  
 MouseRotate mouserotate = new MouseRotate();  
 mouserotate.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mouserotate);  
 mouserotate.setSchedulingBounds(bounds);  
 MouseZoom mousezoom = new MouseZoom();  
 mousezoom.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mousezoom);  
 mousezoom.setSchedulingBounds(bounds);  
 MouseTranslate mousetranslate = new MouseTranslate();  
 mousetranslate.setTransformGroup(transformgroup);  
 BranchGroupRoot.addChild(mousetranslate);  
 mousetranslate.setSchedulingBounds(bounds);  
 transformgroup.addChild(new ShapeTriangleFanArray());  
 BranchGroupRoot.compile();  
 return BranchGroupRoot;  
 }  
}  
  
class ShapeTriangleFanArray extends Shape3D {  
 public ShapeTriangleFanArray() {  
 int vertexesCount = 12;  
// int stripCount[] = new int[1];  
// int stripCount[]=new int[2];  
 int stripCount[]=new int[3];  
 float vertexes[] = {.0f, 0.9f, 0.0f, -1.f, -0.8f, 0.f,  
 -0.8f, -0.6f, -0.2f, -0.6f, -0.9f, 0.2f,  
 -0.4f, -0.8f, -0.2f, 0.f, -0.8f, 0.2f,  
 0.2f, -0.5f, 0.0f, 0.4f, -0.6f, -0.5f,  
 0.6f, -0.8f, -0.3f, 0.8f, -0.9f, -0.2f,  
 0.9f, -0.7f, -0.2f, 1.1f, -0.8f, -0.3f};  
 float colors[] = {0.0f, 0.5f, 1.0f, 0.0f, 0.5f, 1.0f,  
 0.0f, 0.8f, .0f, 1.0f, 0.0f, 0.3f,  
 0.0f, 1.0f, 0.5f, 0.9f, 1.0f, 0.0f,  
 0.5f, 0.0f, 1.0f, 0.0f, 0.5f, 1.0f,  
 1.0f, 0.5f, 0.0f, 1.0f, 0.0f, 0.5f,  
 1.0f, 0.8f, 0.0f, 1.0f, 0.5f, 0.0f};  
 stripCount[0] = 4;  
 stripCount[1] = 4;  
 stripCount[2] = 4;  
 TriangleFanArray triangleFanarray = new TriangleFanArray(vertexesCount,  
 TriangleFanArray.COORDINATES | TriangleFanArray.COLOR\_3, stripCount);  
 triangleFanarray.setCoordinates(0, vertexes);  
 triangleFanarray.setColors(0, colors);  
 PolygonAttributes polygonattributes = new PolygonAttributes();  
 polygonattributes.setCullFace(PolygonAttributes.CULL\_NONE);  
  
// polygonattributes.setPolygonMode(PolygonAttributes.POLYGON\_LINE); // 设置为线模型  
  
 Appearance app = new Appearance();  
 app.setPolygonAttributes(polygonattributes);  
 this.setGeometry(triangleFanarray);  
 this.setAppearance(app);  
 }  
}

（2）结果截图

|  |  |
| --- | --- |
|  |  |
| 一个Fan | 与其线模型 |
|  |  |
| 两个Fan | 三个Fan |
|  |  |
| （4,8）Fan | （4,4,4）Fan |

**三、实验总结**

**实验一总结：**

在这个实验中，创建了一个使用Java 3D库的三维图形应用程序，名为"DisplayTriangles"。以下是实验的总结：

创建三维场景：使用Java 3D库创建了一个三维场景。这包括设置背景颜色和光照条件，以提供更真实的外观。

绘制三角形：通过创建TriangleArrays类，绘制了一组三角形。每个三角形由一系列顶点和颜色组成。这些三角形组成了我们的三维几何形状。

添加交互性：使用Java 3D库的鼠标交互行为类（MouseRotate、MouseZoom和MouseTranslate）来实现用户与场景的交互。这允许用户旋转、缩放和平移整个场景，以便更好地观察三维图形。

实验目的达成：通过完成这个实验，我们达到了以下目标：

学习了如何使用Java 3D库创建基本的三维图形应用程序。

理解了三维场景的配置和渲染，包括光照和背景设置。

掌握了三维几何形状的创建和呈现，包括顶点和颜色的设置。

实践了用户与三维场景的交互性实现。

总的来说，这个实验帮助初步掌握了Java 3D库的使用，并为进一步研究和开发复杂的三维图形应用程序打下了基础。

**实验二总结：**

实验总结： 在这个实验中，创建了一个使用Java 3D库的三维图形应用程序，名为"TriangleStripArrays"。以下是实验的总结：

创建三维场景：使用Java 3D库创建了一个三维场景。这包括设置背景颜色和光照条件，以提供更真实的外观。

绘制三角带：通过创建TriangleStrip类，绘制了一个三角带。每个三角带由一系列顶点和颜色组成。这些三角带组成了三维几何形状。

添加交互性：使用Java 3D库的鼠标交互行为类（MouseRotate、MouseZoom和MouseTranslate）来实现用户与场景的交互。这允许用户旋转、缩放和平移整个场景，以便更好地观察三维图形。

实验目的达成：通过完成这个实验，达到了以下目标：

学习了如何使用Java 3D库创建基本的三维图形应用程序。

理解了三维场景的配置和渲染，包括光照和背景设置。

掌握了三维几何形状的创建和呈现，包括顶点和颜色的设置。

实践了用户与三维场景的交互性实现。

总的来说，这个实验帮助学习者初步掌握了Java 3D库的使用，并为进一步研究和开发复杂的三维图形应用程序打下了基础。

**实验三总结：**

当分析你的Java 3D实验时，可以关注一些更详细的方面，包括以下内容：

1. 导入的包：你导入了`com.sun.j3d`和`javax.media.j3d`等包，这些包是Java 3D库的一部分，用于构建和渲染三维图形对象。

2. 构造函数：你的`TriangleFanArray2`类有一个构造函数，用于设置和初始化应用程序的基本组件。这包括创建`Canvas3D`用于渲染，创建`BranchGroup`作为场景图的根等。

3. 主方法：`main`方法用于启动Java 3D应用程序，创建`TriangleFanArray2`实例，并在窗口中显示它。这是Java应用程序的入口点。

4. 创建场景：`createBranchGroup`方法负责创建场景图。这包括设置背景、光源、交互行为（如旋转、平移、缩放）以及`ShapeTriangleFanArray`，它是包含你自定义的三维形状的`Shape3D`对象。

5. 自定义三维形状：`ShapeTriangleFanArray`类的构造函数用于创建三维图形对象。在这里，你定义了顶点坐标、颜色和三角形扇形的顶点连接方式。你还设置了多边形属性，控制了多边形的渲染方式。

6. 线模型：在注释中提到你尝试设置线模型渲染，但它是注释掉的。如果你想要以线框模式渲染，你可以取消注释`polygonattributes.setPolygonMode(PolygonAttributes.POLYGON\_LINE);`这一行，这将使多边形以线条形式显示而不是填充。

7. 编译和显示：最后，你使用`compile`方法编译`BranchGroup`，然后返回它，以便在主程序中添加到Java 3D视图中。你的应用程序使用`SimpleUniverse`来设置视图平台并显示三维场景。

总之，这个实验涵盖了创建、渲染和交互式查看三维图形的基本步骤，包括了许多Java 3D编程的关键方面。