java3D与计算机图形学期末复习 第三章 java3D基本图形功能

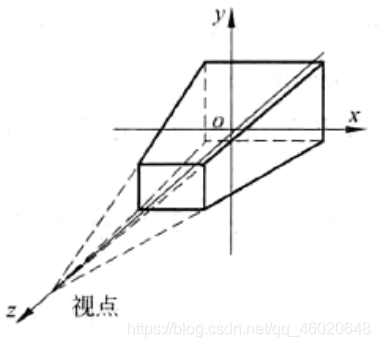
1、SimpleUniverse类

该类可以快速的设置一个最小的用户环境，并且很容易使一个Java3D应用程序运行起来。

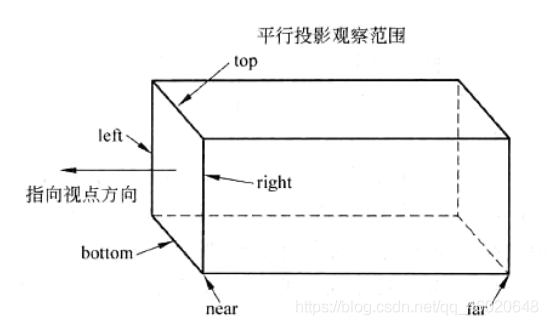
该类创建一个Locale(场景)、一个单独的ViewingPlatform(观察平台)和一个Viewer(观察者对象)，其中,该ViewingPlatform和Viewer观察者都是用其所有默认参数创建的。

2、ViewingPlatform类

透视投影(Perspective Projection)：模拟人眼睛看东西和相机照相的原理，特点是离视点近的形体显得大，离视点远的形体显得小。其中，在投影平面上的投影通过相似三角形的比例关系进行计算。



平行投影(Parallel Projection)：将空间的三维形体分别向三个坐标平面进行平行投影，形成三视图。最常用的就是工程制图中的主视图、俯视图和侧视图。



3、Shape3D类

Shape3D类定义所有的几何体。它包含几何体与该几何体的外观属性。几何体部分定义三维空间体的几何形状,外观属性部分定义颜色、材质等属性。

在一个Shape3D的几何对象列表中的所有元素必须属于同一个等价类，也就是属于同一种基本的几何类型。

等价类：对Geometry Array类的所有子类，所有的点对象是等价的，所有线对象是等价的，所有多边形对象是等价的。

4、Appearance类

Appearance类对象定义所有与显示相关的外观状态，这些状态可设置为一个Shape3D节点的组件对象(Component Object)。

这些显示状态包括下列几个方面：

颜色属性、点的属性、线属性、多边形属性、可视化模式、透明度属性、材质、纹理、纹理属性、纹理坐标的产生。

5、BranchGroup类

一个 BranchGroup作为一个场景图分支的根。BranchGroup对象是唯一能插入到一个Locale对象中的对象。以 BranchGroup节点为根的场景图子图是一个编辑单元。

注意：如果一个 BranchGroup节点包含在另一个子图中，作为其他Group的子节点，则该BranchGroup节点不能插入到一个Locale 节点。

6、TransformGroup类

TransformGroup节点通过Transform3D对象定义了一个唯一的3D空间坐标变换,该变换可对其子节点进行位置、方向及比例变换。（仿射变换）

7、GeometryArray类

GeometryArray类是 PointArray类,、LineArray类、TriangleArray类、QuadArray类、GeometryStripArray类与IndexedGeometryArray类的直接父类。

GeometryArray类的对象中包含独立的位置坐标数组、颜色数组、法向量数组、纹理坐标数组和顶点属性。

8、PointArray类

PointArray类是组织与定义点的数组类。

9、PointAttribute类

PointAttributes类的对象用来定义点的各种属性。

点的属性包括：

（1）Size，点的大小(默认为1个像素)；

（2）Anitialiasing，反走样(当一个点的大小大于一个像素单位时,为了使点在显示时外形接近于一个圆形,则需要设置该参数)。

10、IndexedPointArray类

从定义好的所有点的坐标与颜色数组中选择出一部分点进行显示。

1、LineArray类

LineArray类是组织与定义线的数组类。

该类生成的线段是不连续的,线段的连接方式:0-1，2-3，4-5等。其中,1、2之间与3、4之间不连接。

2、LineAttributes类

LineAttributes类定义所有与线的显示相关的属性与状态。

3、IndexedLineArray类

IndexedLineArray类从定义好的所有点的坐标中选择一部分连成线段显示。

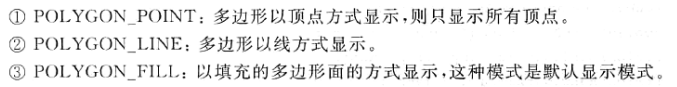
4、LineStripArray类

LineStripArray类将定义好的数据点分组，分别连成几段连续的线段，分成几个组，组间线段连续，各组之间互不连接。

5、多边形属性类PolygonAttributes类

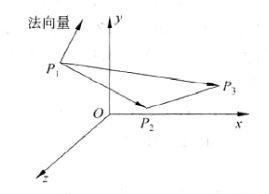
PolygonAttributes类用来定义多边形显示时的相关属性。

三种显示模式：

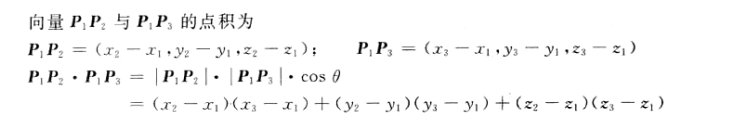


6、多边形面及其法向量的计算

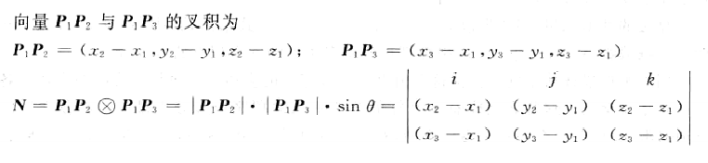
如图所示，求法向量？



先求出点乘，计算出两向量的夹角：



通过叉乘，计算出法向量N：

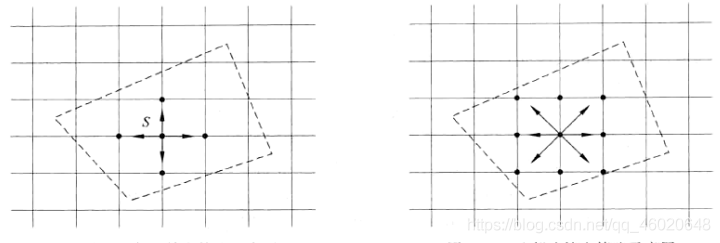


7、多边形可视化的填充算法

（1）种子填充算法

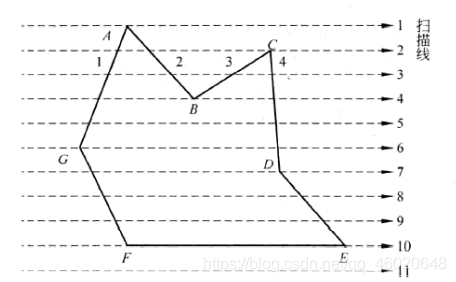
种子填色算法首先在要填充的多边形区域内选择一个填色种子点( Seed Point)，然后以该种子点为基础，通过与多边形边界的像素点相比较进行填色。

种子填色算法主要有四邻法（如左图）和八邻法（如右图）：



（2）扫描线填充算法

每一条扫描线从上向下以一个像素单位为间隔逐次向下扫描，采用奇偶计数法判断多边形内部的点对。



出现点重合时采取的处理方法：

首先按照多边形顶点顺时针方向标明：A、B、C、D、E、F、G

扫描线6：扫描到3个顶点，G点处两点重合，按奇偶计数法会出现错误。于是将入线FG的G点的y轴坐标向下移动1个像素单位（我理解为逆时针方向）。

扫描线7：扫描到3个顶点，D点处两点重合，按奇偶计数法会出现错误。于是将入线CD的D点的y轴坐标向上移动1个像素单位（我理解为逆时针方向）。

8、三角面TriangleArray类

TriangleArray类以一维顶点数组给出顶点的坐标值，从前向后依次以3个顶点形成一个三角形，并且上一个三角形与下一个三角形之间没有公用顶点。

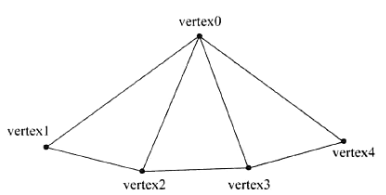
9、三角带TriangleStripArray类

在同一个三角形Strip内，生成三角形时，第一个三角形由前3个顶点生成，第一个三角形的后两个顶点与第四个顶点形成第二个三角形，其余依次类推。

两个子Strip之间没有交点。

10、三角扇TriangleFanArray类

TriangleFanArray类在组织三角形时，以第一个顶点为公用顶点，依次与其余顶点分别连接形成三角形。用这种方法生成的一系列三角形公用第一个顶点。如图：



1、四边面QuadArray类

QuadArray类以顶点坐标数组中给出的一维顶点数组，从前向后依次以4个顶点形成一个四边形面，并且相邻两个四边形面之间没有公用顶点。给定的总的顶点数必须是4的倍数。

2、颜色属性ColoringAttributes类

ColoringAttributes类定义所选择的颜色与光照模型(也称为阴影模型)。

3、材质Material类

Material类的对象定义一个三维物体在光照情况下的外观。

如果在一个Appearance类的对象中的Material类的对象一项为null，则光照效果对所有包含该Appearance类的对象的节点不起作用。

4、透明属性TransparencyAttributes类

TransparencyAttributes 类的对象定义所有几何体透明度的属性。

5、光源Light类

Light类叶子节点是一个抽象类，其中定义的一组参数（光颜色、光照开关标志和一个光的作用范围）属于所有类型的光。

光源的类型包括：

（1）点光源（Point Light）

（2）平行光源（Directional Light）

（3）环境光源（Ambient Light）

6、Vetor3f类

Vector3f类是一个包含有3个单精度浮点元素x、y、z的向量。

7、Java语言多维数组表示

没啥好说的，懂的都懂，不懂的我也不想说了（你搁这搁这呢？）