### 简介

在本文，我们将看到什么是viewingVolume和viewport。我认为通过理解viewingVolume来了解openGl是如何工作的和如何从抽象的3d空间画一个顶点到窗口屏幕是非常必要的。

（openGl the red book）

### viewingVolume

3d空间被3条轴定义（向量）：x，y，z。这三条轴创建了一个坐标轴系统。这个3d空间中的所有点都能以这三个向量的线性组合方式来表达。一个点表示成（a，b，c），意味着

（ax+by+cz）。整个3d空间的体积是无限的。

viewingVolume通过创造封闭的体积来分割3d空间。在这个体积之外的点将会从屏幕上消失。这个特性解释了第一个调用glTranslte（紧接着glLoadIdentity之后）：我们来到

viewingVolume内部。一开始，我们坐标落在viewingVolume正面那个面上。

viewingVolume能够以两种不同的投影方式创建：

* 透视投影：渲染的物体受距离的影响，例如一辆远去的车看起来越来越小。
* 正投影：渲染的物体不受距离的影响，例如一个菜单、一段文字、2d物体等。

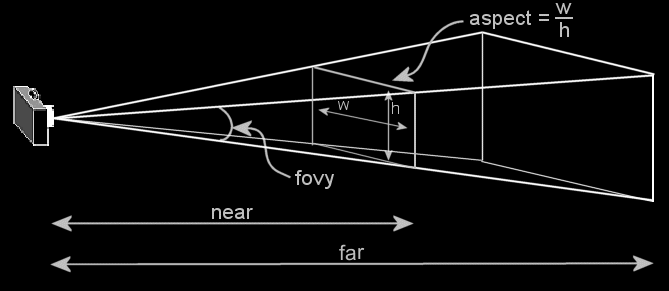
##### 透视投影

这种投影，viewingVolume形状像一个截头金字塔。

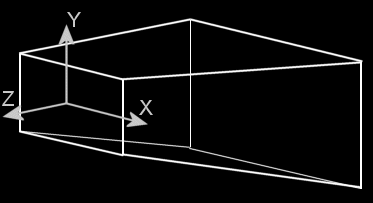
获取这种形状的viewingVolume需要6个截面：前面、后面、右面、左面、上面、下面。

这些截面可以通过glu.gluPerspective（fovy，aspect，near，far）来获得，fovy参数是向上的面和向下的面之间的夹角，aspect是窗口的宽高比（窗口是场景将要渲染到的屏幕区域）。

参数的具体含义如下图：



viewingVolume如下图：



x轴和y轴平行于最近的面（near plane）。坐标轴的原点和最近的面的中心重合。z轴和最近的面垂直，并且正方向指向视点。

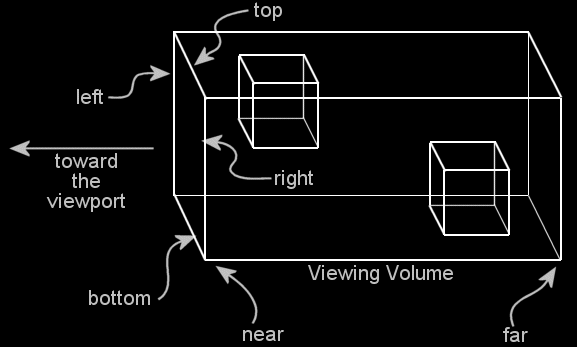
参数的值

透射投影viewingVolume的一个例子：gluPerspective（45.f，width/height，1.f，500.f）

aspect参数是窗口宽和高之间的比例（窗口是持有openGl场景的地方）。

##### 正投影

在这种投影情况下，viewingVolume的形状是一个平行六面体。分割空间的6个截面可以通过gl.glOrtho（left，right，bottom，top，near，far），每一个参数如下图所示：



viewingVolume如下图：

