# 实验4 曲线、消隐与真实感图形绘制

## 一、实验目的与要求：

1. 掌握Hermite、Bezier曲线的表达方法，了解OpenGL绘制曲线方法
2. 了解OpenGL消隐
3. 了解OpenGL光照与材质
4. 了解OpenGL纹理映射
5. 了解OpenGL半透明

## 二、实验内容与步骤：

### 1 Hermite、Bezier曲线

* 曲线曲面的**插值**（拟合）：当用一组型值点来指定曲线曲面的形状时，形状完全通过给定的型值点列；曲线曲面的**逼近**：当用一组控制点来指定曲线曲面的形状时，求出的形状不必通过控制点列。
* n次样条参数多项式曲线的矩阵：

，或表达成：



* 三次Hermite样条：假定型值点Pk和Pk+1之间的曲线段为p(t),t∈[0,1]，给定矢量Pk、Pk+1、Rk和Rk+1,则参数方程为：



通常将T•Mk称为Hermite基函数（或称混合函数，调和函数）：



Drawing2

* Bezier曲线定义：



三次Bezier曲线（n=3）定义：



或表达成矩阵形式：





* OpenGL绘制Bezier曲线：利用求值器计算并绘制基于控制点逼近的大量的线段。相关函数及操作有：

// 启动求值器

glEnable(GL\_MAP1\_VERTEX\_3);

// 设定求值器

void glMap1f (GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat \*points);

// 方式1：计算从u1到u2生成均匀分布的un个参数值

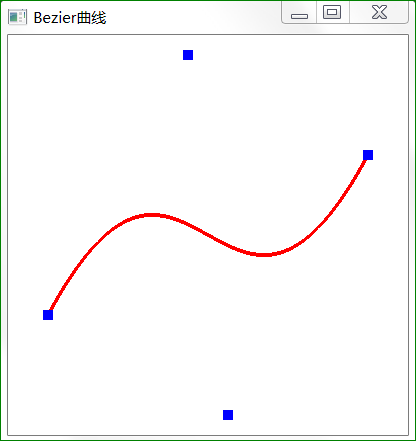
void glMapGrid1f (GLint un, GLfloat u1, GLfloat u2);

// 以折线方式绘制Bezier曲线

void glEvalMesh1 (GLenum mode, GLint i1, GLint i2);

// 方式2：通过求值器计算并绘制曲线上的任一位置的坐标

void glEvalCoord1f (GLfloat u);



### 2 OpenGL消隐

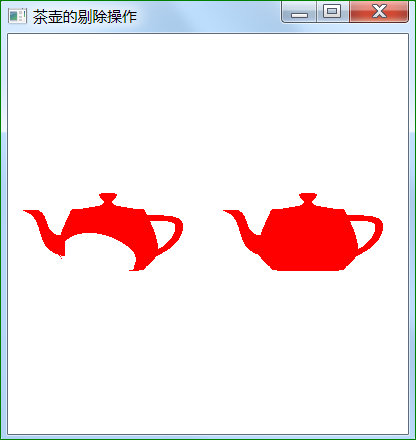
* 物体的消隐或隐藏线面的消除：在给定视点和视线方向后，决定场景中哪些物体的表面是可见的，哪些是被遮挡不可见的。
* 消隐算法按实现方式分类
  + 图像空间消隐算法以屏幕像素为采样单位，确定投影于每一像素的可见景物表面区域，并将其颜色作为该像素的显示颜色。如深度缓冲器算法、A缓冲器算法、区间扫描线算法等。
  + 景物空间消隐算法直接在景物空间（观察坐标系）中确定视点不可见的表面区域，并将它们表达成同原表面一致的数据结构。如BSP算法、多边形区域排序算法等。
  + 介于二者之间的算法，如深度排序算法、区域细分算法、光线投射算法等。
* 多边形剔除：主要用于去除多边形物体本身的不可见面，以提高图形系统的性能。

glEnable(GL\_CULL\_FACE);

glCullFace (mode); // GL\_FRONT或GL\_BACK

glClipPlane(GL\_CLIP\_PLANE0, equ); // 定义裁剪平面 a*x*+b*y*+c*z*+d=0

glEnable(GL\_CLIP\_PLANE0); // GL\_CLIP\_PLANE0——GL\_CLIP\_PLANE5



* 深度测试： OpenGL中的深度测试是采用深度缓存器算法，消除场景中的不可见面。

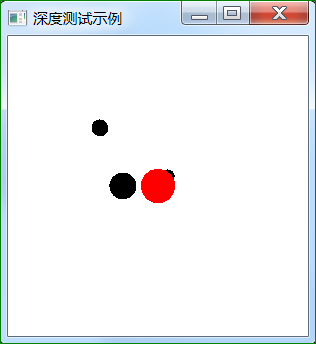
glEnable(GL\_DEPTH\_TEST);

glDepthRange (nearNormDepth, farNormalDepth);

glClearDepth (maxDepth);

glClear(GL\_DEPTH\_BUFFER\_BIT);

glDepthFunc(func);

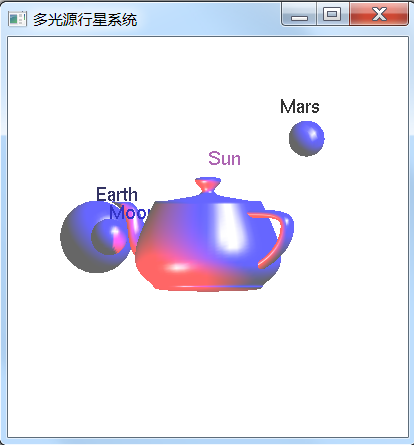


### 3 OpenGL光照与材质

* 简单光照模型：只考虑反射光的作用。反射光由环境光、漫反射光和镜面反射光三部分组成。



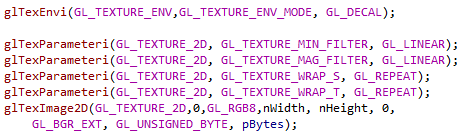


### 4 OpenGL纹理映射与透明处理

* **纹理贴图：生成颜色纹理的一般方法，是预先定义纹理模式，然后建立物体表面的点与纹理模式的点之间的对应。当物体表面的可见点确定之后，以纹理模式的对应点参与光照模型进行计算，就可把纹理模式附到物体表面上。这种方法称为纹理映射（Texture Mapping）。**

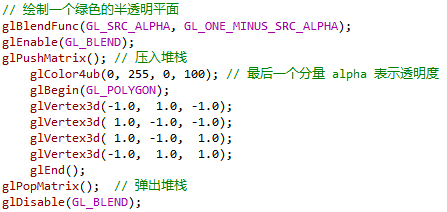


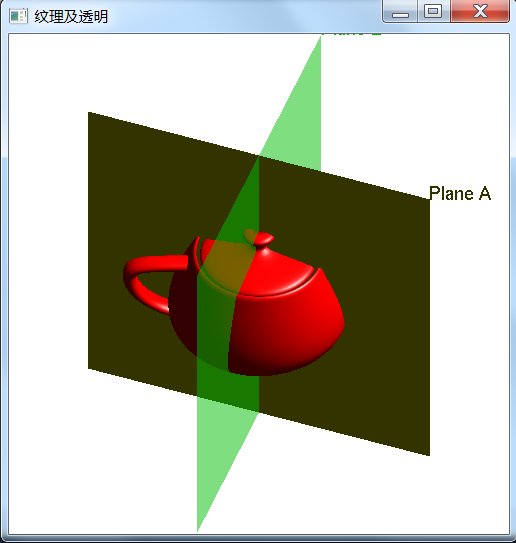
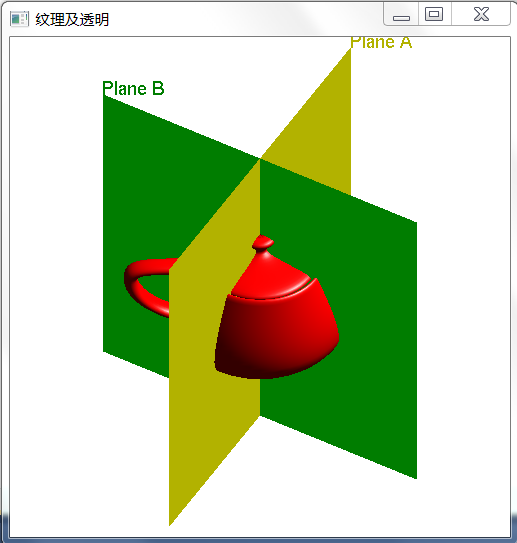


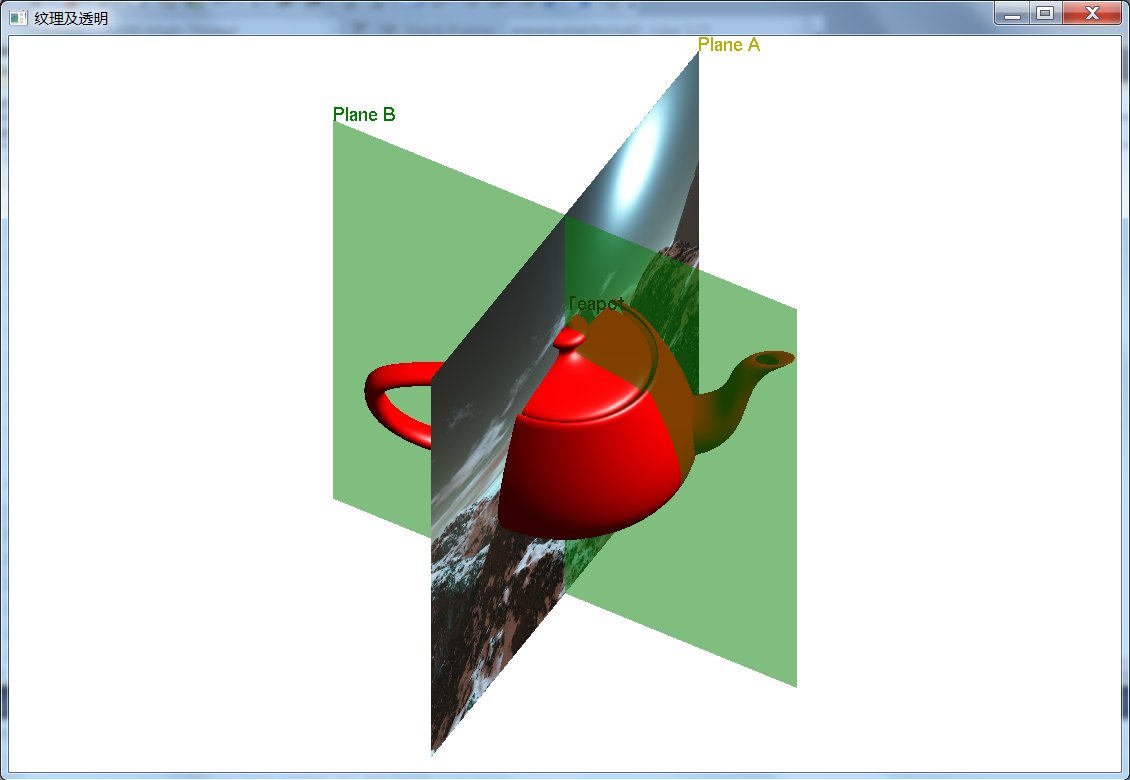


* **透明处理：OpenGL中使用混合实现透明处理。混合操作是指将输入对象（源）的颜色值与当前存储在帧缓存中的像素（目标）颜色值合并的过程。在RGBA颜色模式（A表示透明度）中，已知源像素的颜色值为（Sr，Sg，Sb，Sa），目标像素的颜色值为（Dr，Dg，Db，Da），颜色混合后像素的颜色为：**

**（RS·Sr +RD·Dr，GS·Sg +GD·Dg，BS·Sb +BD·Db，AS·Sa +AD·Da）**







## 三、实验练习：

###### 1 掌握Hermite、Bezier三次样条曲线的参数化表达、计算及绘制方法；

###### 2 了解消隐原理，练习OpenGL深度测试；

###### 3 练习OpenGL光照、纹理与透明处理；

###### 4 验证教材中的OpenGL例子。

## 四、提交报告：无

建议可利用计算机图形学的原理方法，尝试设计并实现一个有趣的三维动画系统（比如太阳系、弹跳球等）。不要求写报告。