

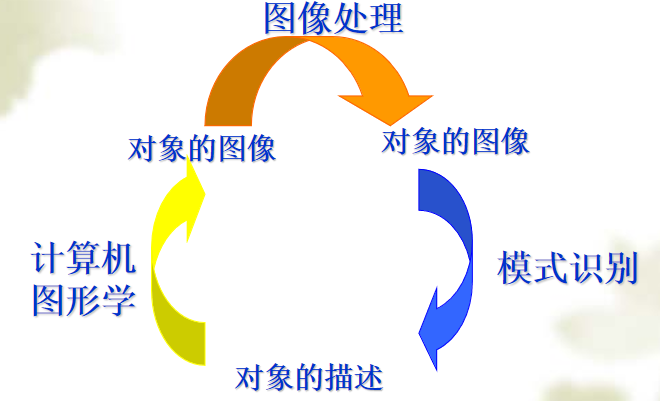
主要知识点：

1,数字图像处理、 计算机图形学、模式识别三门课之间的关系

图像处理(image processing)：利用计算机对图像进行分析处理，继而再现图像。

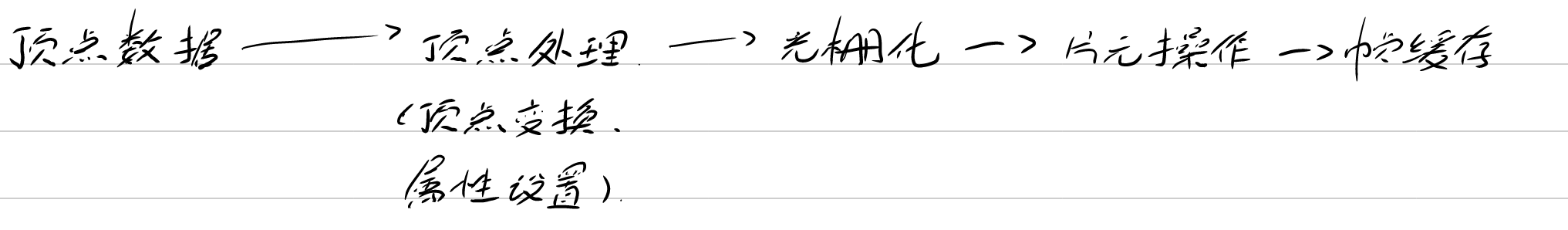
计算机图形学(computer graphics)：研究图形的计算机生成和基本图形操作，是从数据描述到图形生成的过程。

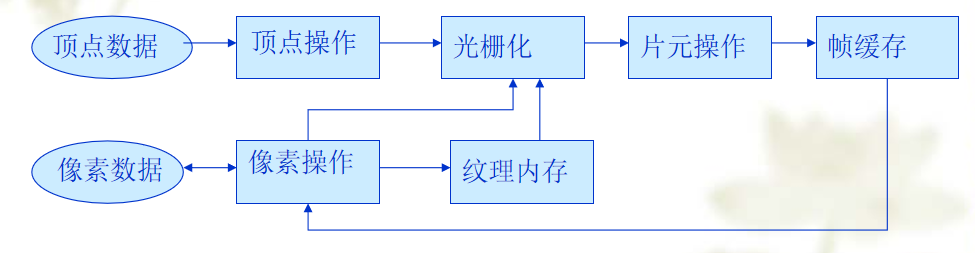
模式识别(pattern recognition)：计算机对图形信息进行的识别和分析描述，是从图形（图像）到描述表达的过程。



彼此交叉、界线模糊、相互渗透

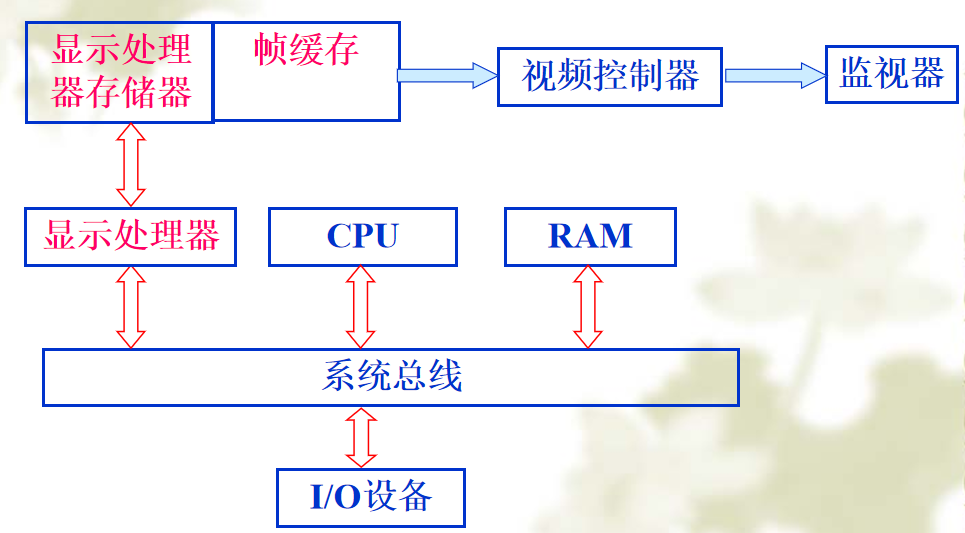
2,就能够根据图进行分析。图形管线的各个阶段工作。





3,计算机图形处理架构（有GPU模式和无GPU模式）

4,简述光栅式扫描显示系统



**简单的光栅扫描系统结构：**

显示处理器：可将应用程序给出的图形定义数字化为一组像素强度值，并存放在帧缓冲器中

帧缓冲存储器(frame buffer) ：保存一组对应屏幕所有点的强度值，经视频控制器可绘制出图形

视频控制器：有两个寄存器用来存放屏幕像素的坐标，可实现图形的基本刷新操作

5,屏幕分辨率三种描述，扫描频率，带宽计算

屏幕分辨率三种表示方法：

（1），光点直径（CRT屏幕上发光荧光层亮点的亮度分布图）

（2），水平方向上的光点数×垂直方向上的光点数r(x×y)

（3），显示器精度dpi

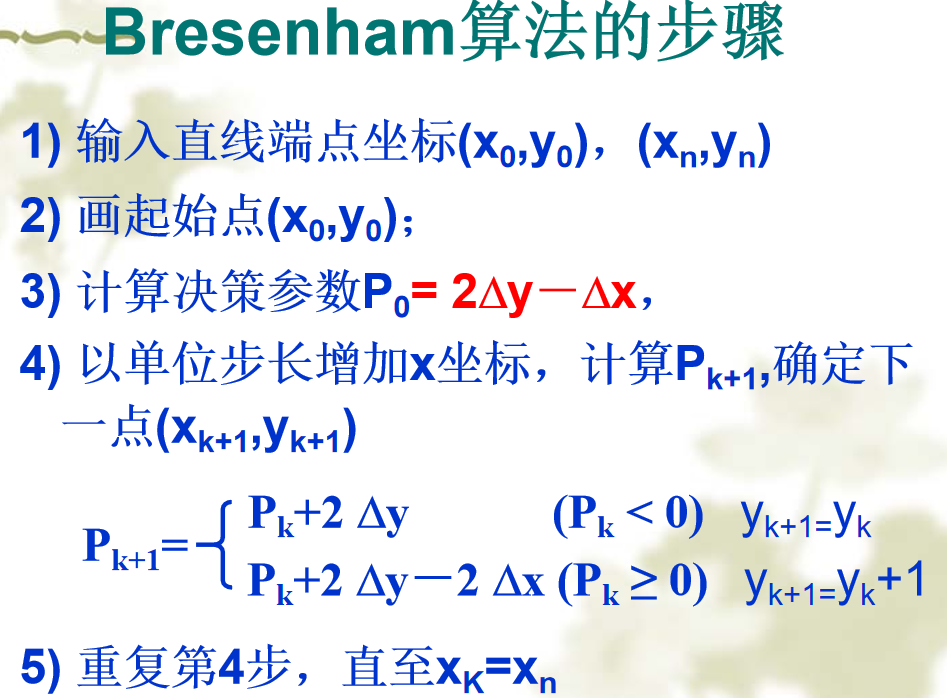
显示器的扫描频率：也叫刷新率，分为行频（水平扫描频率）和场频（重绘率/垂直扫描频率/帧频）

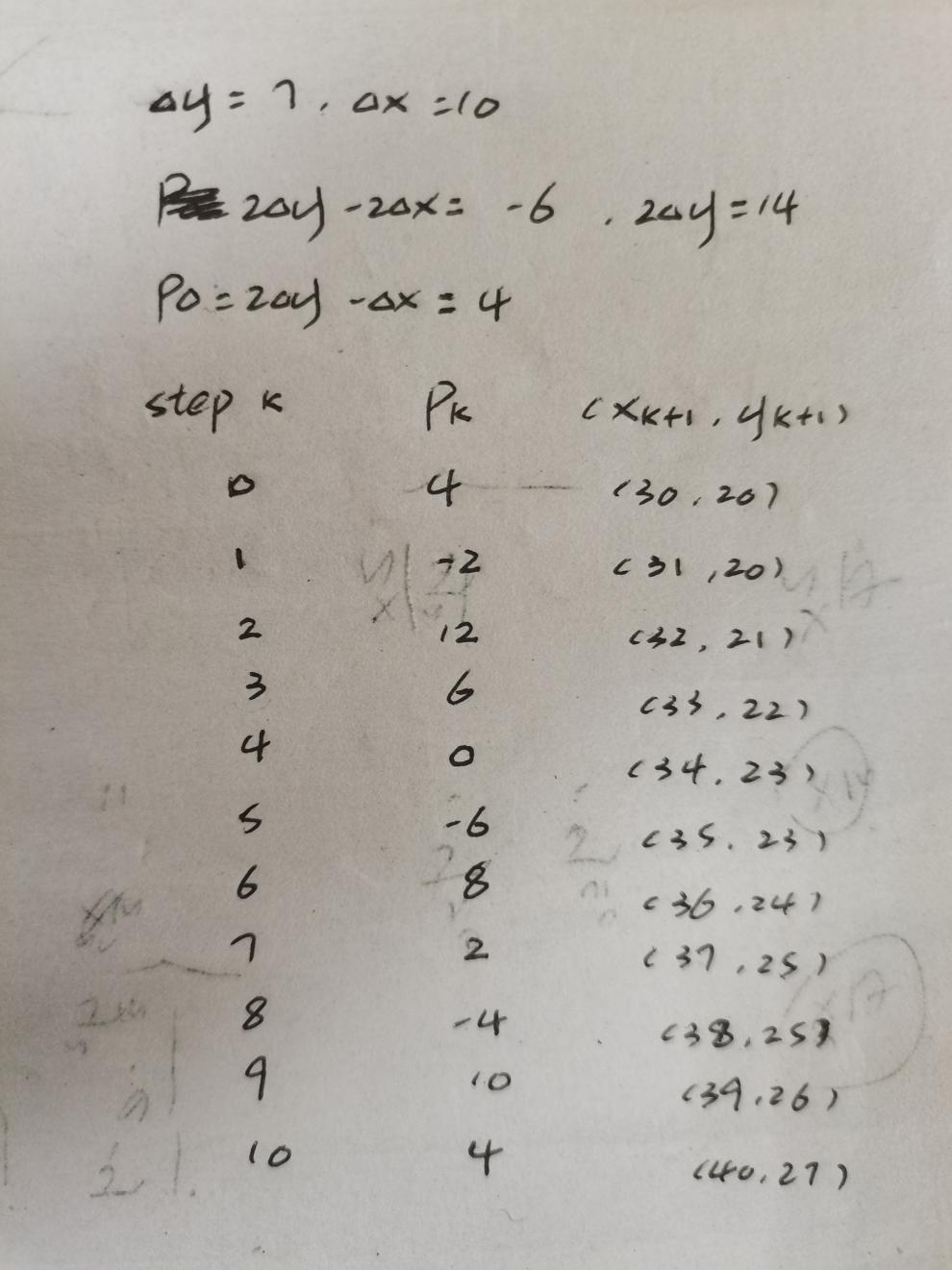
行频：电子枪每秒在屏幕上所扫描过的水平线数

场频：每秒钟重复绘制显示画面的次数

视频宽带：单位时间内所扫描的点数

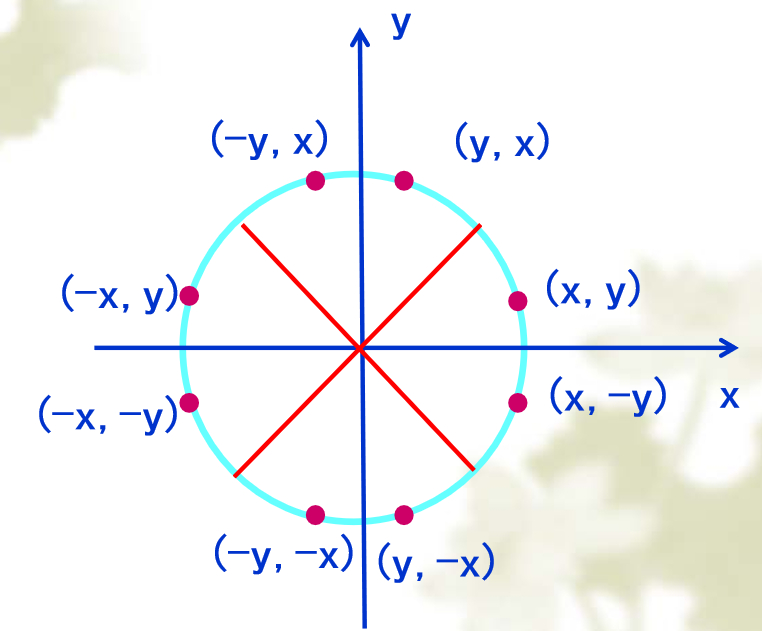
6,用bresenham算法光栅化线段P1(30,20)，P2(40,27)



【

7,画图描述圆对称性坐标计算，

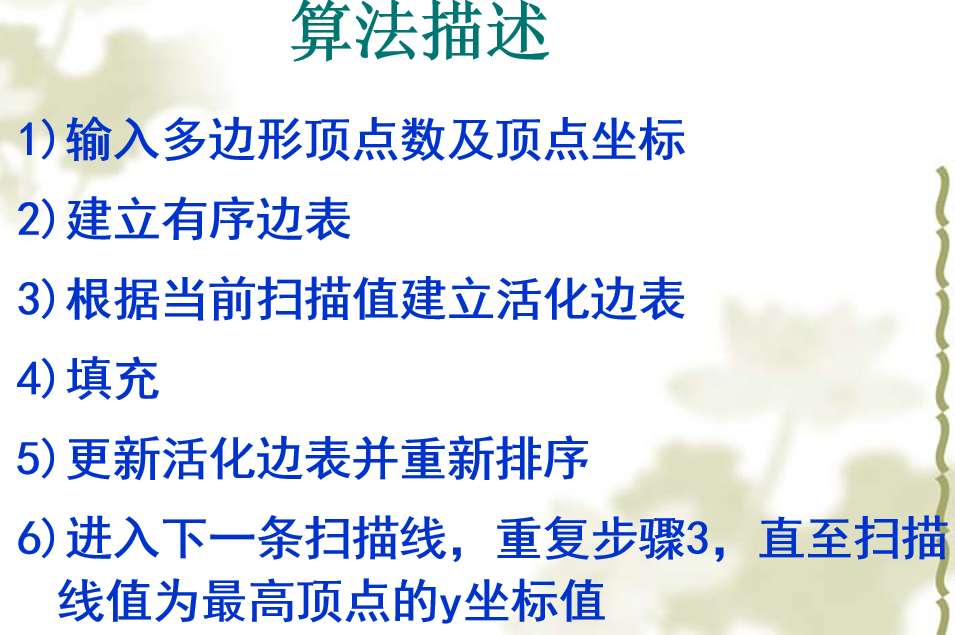
(x,y)



8,描述扫描线多边形填充算法流程

基本思想：对于一条扫描线填充过程可以分为四个步骤：求交、排序、配对、填色

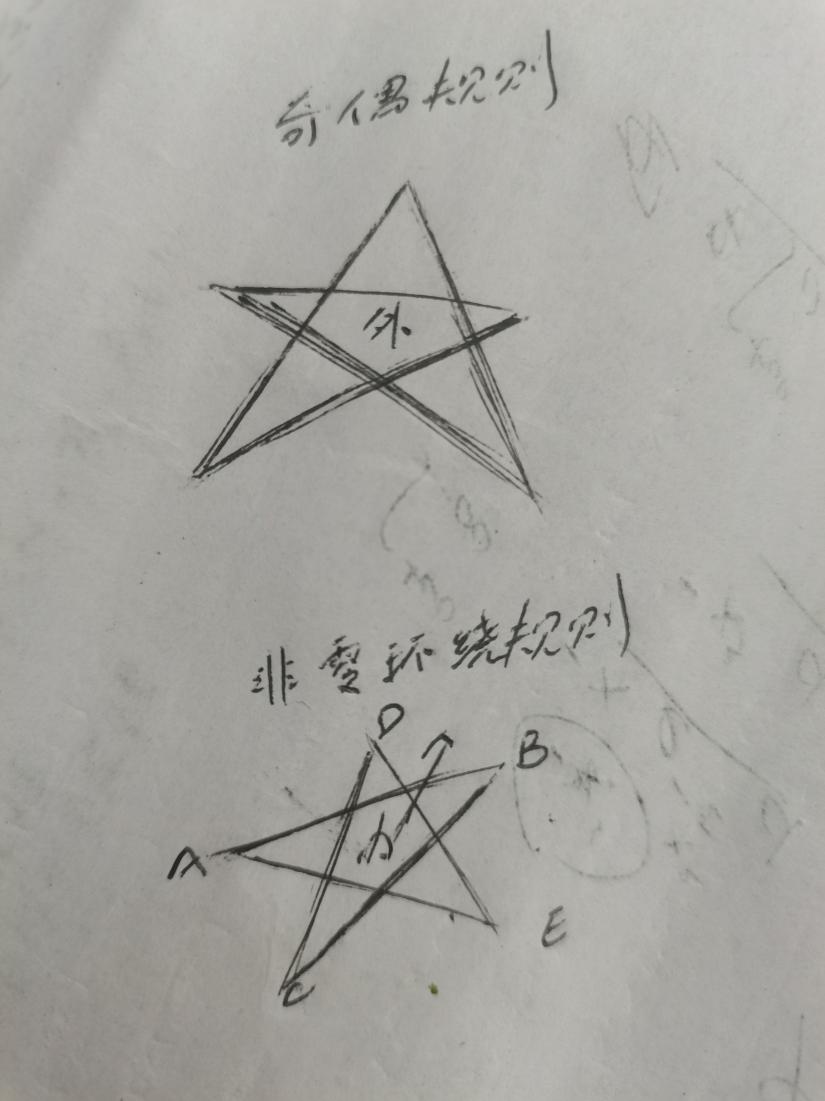
1. 输入多边形顶点数及顶点坐标，建立有序边表；扫描线自底进行扫描，计算扫描线与多边形边界的交点（注意多边形顶点处的交点处理），根据当前扫描值建立活化边表；根据活化边表进行配对并填色，更新活化边表并重新排序；然后进入下一条扫描线，重复计算交点、建立活化边表并进行填充，直至扫描线值为最高顶点的y坐标值



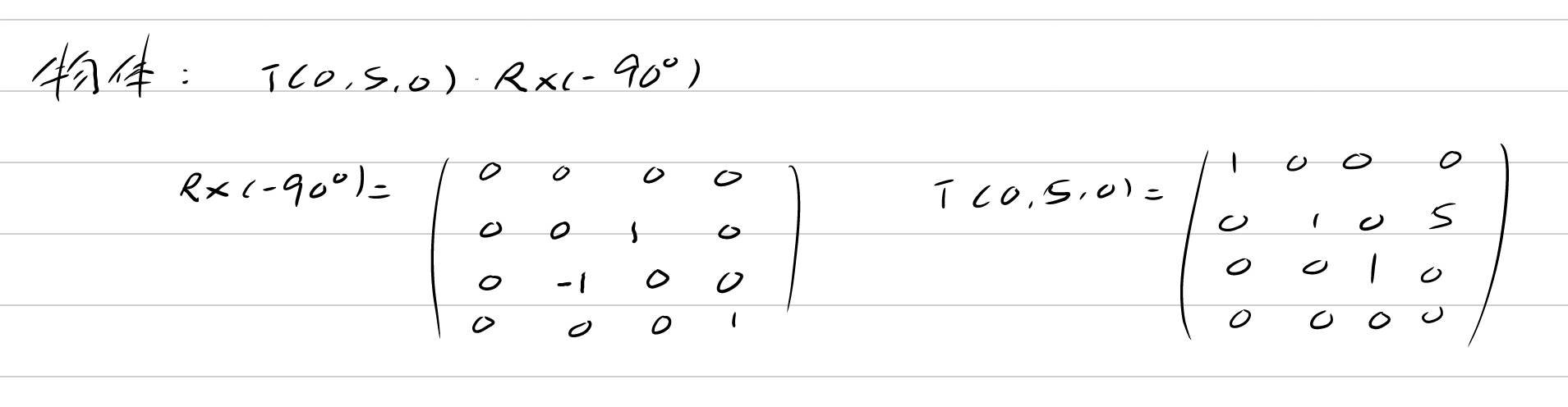
9,分别用奇偶规则和非零环绕规则判断下列内外区域

奇偶规则：从任意位置P作不经过顶点的射线，计算射线穿过的多边形边的数目，奇数为内部点，否则为外部点

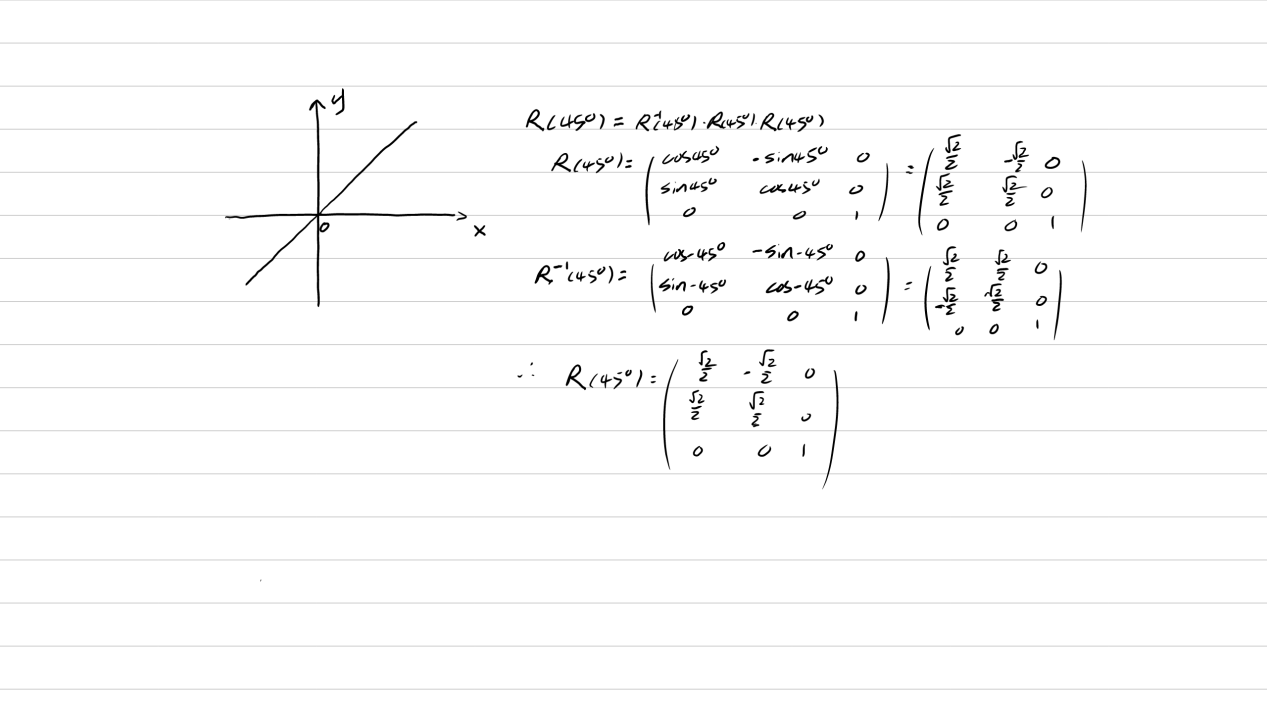
非零环绕规则：环绕数初始为零，从位置P作不经过顶点的射线，多边形边从右至左穿过射线，加1，多边形边从左至右穿过射线，减1；非零为内部点；否则为外部点



10,构造大矩阵实现，视点（0,0,0）从z轴负方向，到y轴上一点（0,5,0）向（0,0,0）观察变换。



11,在二维平面上，构造大矩阵实现，绕y=x轴旋转45度角的旋转矩阵。

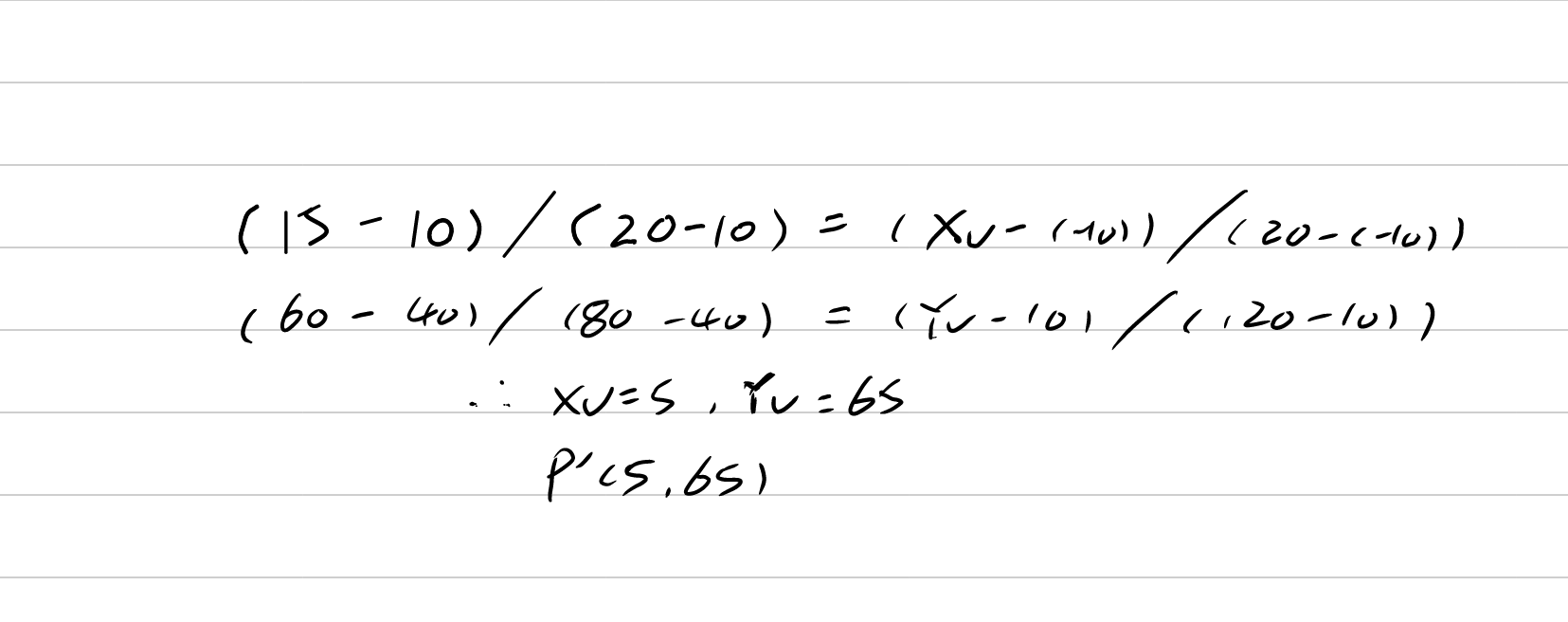


12,描述观察变换的坐标变换关系

已知w1=10, w2=20, w3=40, w4=80,

v1=-10, v2=20, v3=10, v4=120,

窗口中一点P(15，60)，求视口中的映射点P'

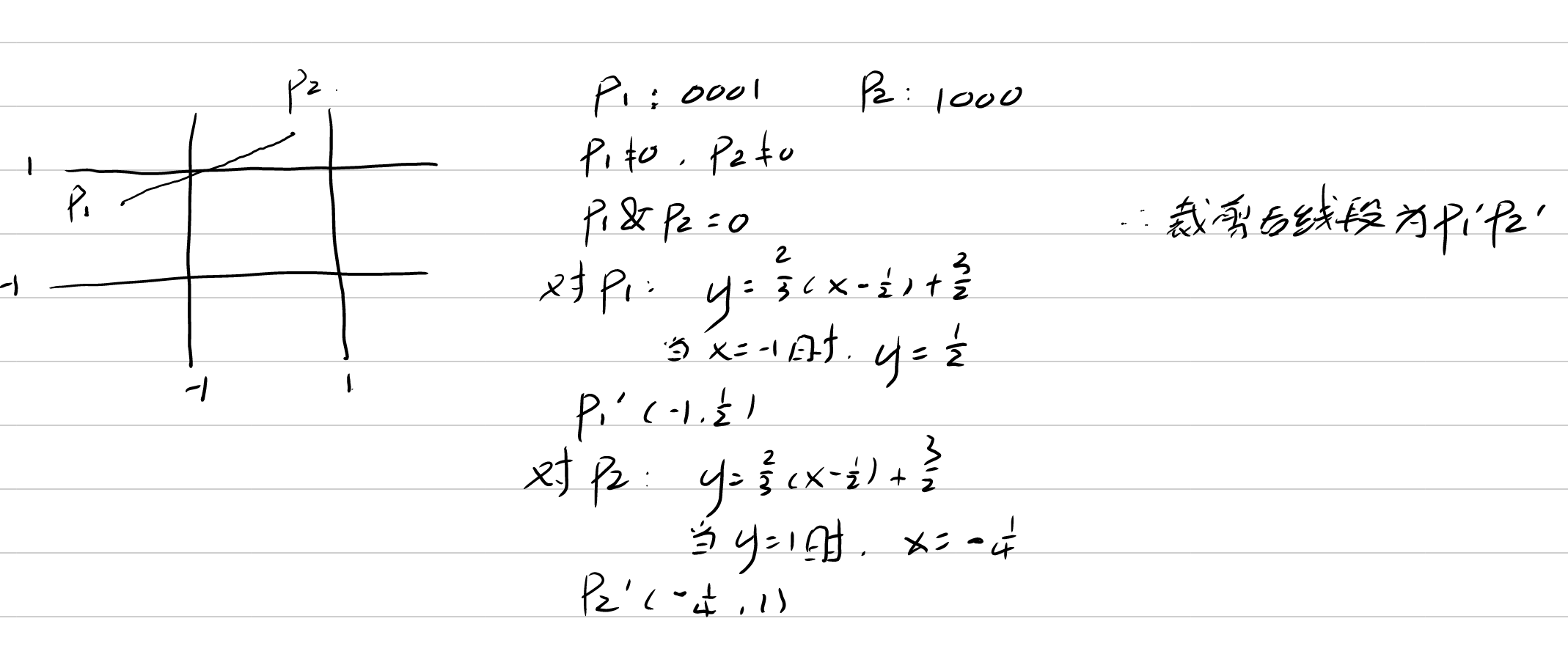


13,已知线段的两个端点

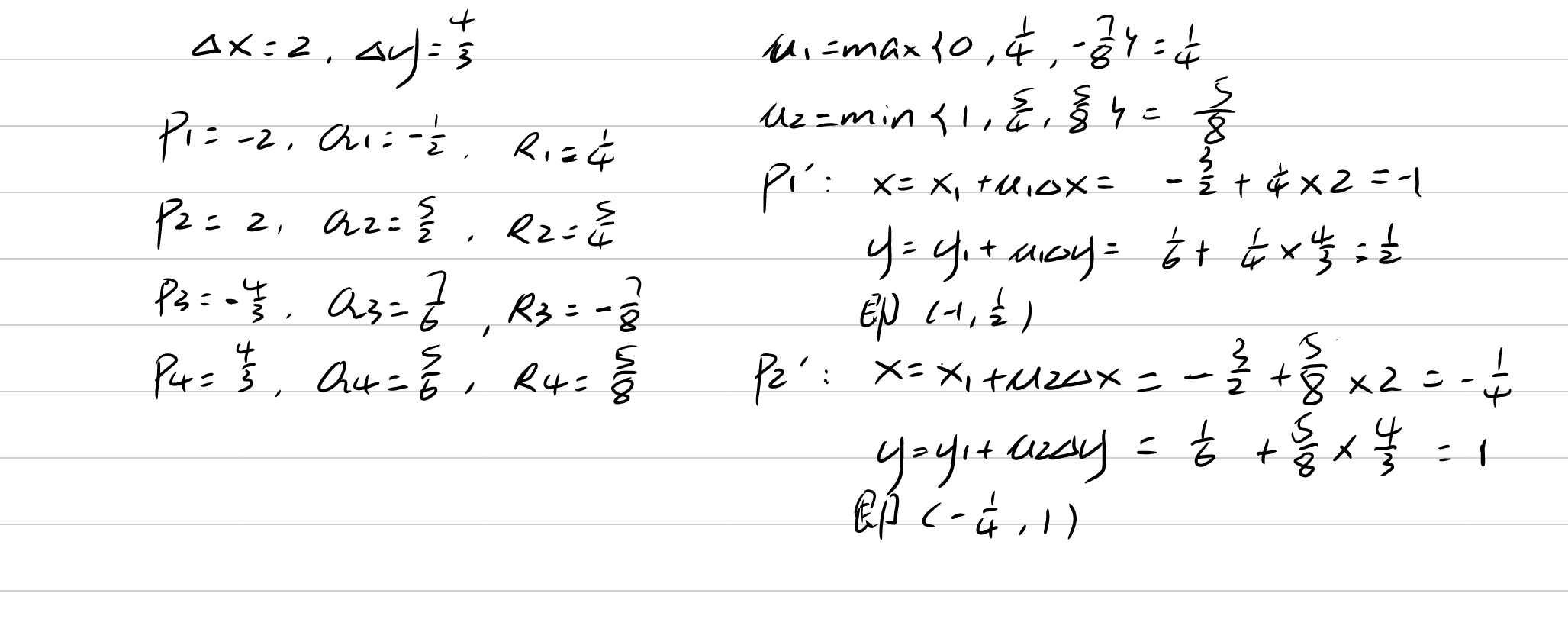
P1(-3/2, 1/6)，P2(1/2, 3/2)

窗口边界x=-1, x=1, y=-1, y=1

用CS算法对线段进行剪裁



14,利用LB算法实现上述例子



15,简述平行投影，透视投影概念

1）平行投影

将物体表面上的点沿平行线投影到显示平台上

三维场景中的平行线在投影到二维显示平面中后仍然是平行线

(2) 透视投影

沿会聚路径将点投影到显示平面上

远小近大，平行线投影后成了会聚线

平行投影：坐标位置沿平行线变换到观察平面上

透视投影：物体位置沿收敛于某点的线变换到观察平面

16,投影的三要素

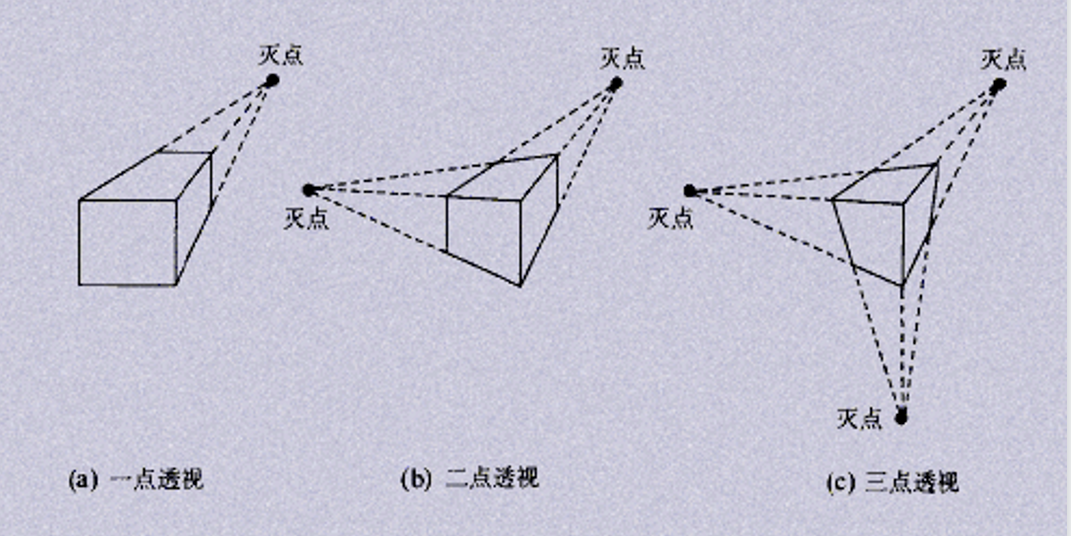
投影中心、投影平面和投影射线

17,主灭点，一点透视，两点透视，三点透视

一组平行线投影后收敛于一点称之为灭点

物体中平行于某一坐标轴的平行线的灭点

透视投影按照主灭点数目分为一点透视、二点透视、三点透视



18,观察体调整

调整的意义：使得计算、处理方便快捷

正平行投影：观察体为正平行六面体，不用调整 斜平行投影和透视投影的观察体均需要调整

19,概念，多边形网格模型表示中，基本数据表形式，顶点表，边表，面表

多边形表数据表分为两组进行组织：

几何表：顶点坐标和用来标识多边形表面空间方向的参数

属性表：指明物体透明度及表面反射度的参数和纹理特征



20,概念，用函数描述的二次曲面模型，如球体表面，如何进行绘制

给定二次曲面函数，图形包将指定曲线方程投影到显示平面上，且沿着投影函数路径绘制像素位置，最终绘制成二次曲面。

21,概念， 样条曲线，样条曲面

样条曲线在计算机图形学中的含义：由多项式曲线段连接而成的曲线、在每段的边界处满足特定的连续性条件 样条曲面：使用两组正交样条曲线进行描述

22,概念，样条曲线的两种类型

插值样条曲线：选取的多项式使得曲线通过每个控制点

逼近样条曲线：选取的多项式不一定使曲线通过每个控制点

23,概念， 凸壳的概念

包含一组控制点的凸多边形边界

24,概念，分段连续中连续的定义

参数连续性条件和几何连续性条件

**参数连续性条件：**

两个相邻曲线段在相交处的参数导数相等

零阶连续(C0连续)：简单地表示曲线连接，两个曲线必在相交点处有相同的坐标

一阶连续(C1连续)：说明代表两个相邻曲线的方程在相交点处有相同的一阶导数（切线）

二阶连续(C2连续)：两个曲线段在交点处有相同的一阶和二阶导数，交点处的切向量变化率相等

**几何连续性条件：**

两个相邻曲线段在相交处的参数导数成比例

零阶连续（G0连续）：与0阶参数连续性相同，即两个曲线必在公共点处有相同的坐标

一阶连续（G1连续）：表示一阶导数在两个相邻曲线的交点处成比例

二阶连续（G2连续）：表示两个曲线段在相交处的一阶和二阶导数均成比例

25,简述Bezier的几个特点

Bezier曲线总是通过第一个和最后一个控制点

Bezier曲线在第一个控制点P0处与直线P0P1相切，在最后一个控制点Pn处与直线Pn-1Pn相切。

Bezier曲线总是落在控制点的凸壳内，保证了曲线沿控制点的平稳前进

26,threejs的浏览器中如何调试的方法，文件相对路径关系（src=“/images/ddd.jpg”与 “../images/ddd.jpg” 与 “./images/ddd.jpg”与 “images/ddd.jpg”的区别）

27,VBO,PBO,FBO的概念

28,如何创建一个vbo，并基于vbo进行绘制（写出相应的伪代码过程）

29,深度缓存的意义，基于深度检测的方法

深度缓存的意义：把一个距离观察平面(近裁剪面)的深度值(或距离)与窗口中的每个像素相关联。

深度测试是采用深度缓存器算法，消除场景中的不可见面。在默认情况下，深度缓存中深度值的范围在0.0到1.0之间，这个范围值可以通过函数：

glDepthRange(nearNormDepth,farNormalDepth);

glClearDepth(maxDepth);

glClear(GL\_DEPTH\_BUFFER\_BIT);

glDepthFunc(func);

30,描述固定管线中的光照模型

Lambert模型(漫反射)、Phong模型(镜面反射)、Blinn-Phong光照模型(修正镜面光)、Rendering Equation(全局光照模型）

最后有实验题，而且都是大题，一般共占40分左右

1. threejs的程序主框架（初始化和主循环，）

初始化：

ar scene = new THREE.Scene();

var camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 );

var renderer = new THREE.WebGLRenderer(); renderer.setSize( window.innerWidth, window.innerHeight ); document.body.appendChild( renderer.domElement );

主循环框架：

function update () {

renderer.render(scene, camera);

requestAnimationFrame(update);

}

requestAnimationFrame(update);

1. 利用threejs实现某个场景的绘制，写出设计和具体代码
2. 利用threejs实现对象的运动控制的设计，如太阳系运动，以及汽车与轮子运动等。
3. 利用threejs实现天空盒的搭建
4. 利用threejs设计实现一个简单游戏：用鼠标控制旋转方向，键盘控制向前向后移动，空格键发射子弹，射线检测碰撞结果

6， 利用threejs设计实现一个函数曲线的绘制

7， 利用threejs设计实现一个曲面的绘制