放视频时：

大家看啊，我们的游戏是基于webGL绘制的，其中绘制了许多基本体素，包括圆柱、球、圆锥等等，我们还导入了obj格式的模型，这个飞机就是。在飞机上进行了纹理的贴图和MTL材质的导入，大家注意看，飞机的窗户和表面对于光线的反射效果是不一样的。

基本几何变换功能也实现了，飞机可以旋转、平移，实现很逼真的俯冲、抬升、侧飞效果。随机的一些球会放大缩小。我们也实现了光线照明功能，并根据游戏运行时间改变光线的照明方向，当然，鼠标拖动改变观察视角、鼠标滚轮缩放也都实现了。

前面的功能都比较基础，接下来具体介绍一下实时碰撞检测、粒子效果、实时阴影绘制和云的效果。

实时碰撞检测的原理其实是在物体外套一个AABB包围盒，简化物体运动过程中的碰撞检测。在本游戏中，可能发生碰撞的只有飞机和球体，因此我们将所有可能与飞机产生碰撞的球体放入ballSet集合中，每一帧进行一次碰撞检测，判断飞机与每个球的包围盒的交叠部分是否大于预定的阈值，如果交叉部分大于阈值则视为发生了碰撞。碰撞后根据碰撞到的物体种类，会引发得分或爆炸的两种粒子效果。

粒子效果实际上是很多随机位置的点组成的，并设定他们会在一个随机时间后分别消散。webGL中绘制点的基本效果为一个个小正方形，为了实现比较合理的粒子效果，在着色器中判断片元与像素点的距离，舍弃较远的片元即得到了球体的粒子。

实时阴影绘制的原理为实时绘制一张阴影纹理，然后将其贴图于物体表面。之前老师上课的时候有讲到过，只要从光源的方向看向每一个物体，就可以得到他们的深度值，将这个深度数值以渲染到纹理的方式存入阴影纹理之中，在实际绘制物体时将该阴影纹理贴在每一个物体表面，判断物体深度值和阴影贴图中存储的深度值的大小，以此改变物体的颜色，即可实现实时阴影效果。

云实现了基本的体积云，云的形状采用了worley\_perlin 两种噪声的组合，在天空盒范围内生成连续变化并且具备蓬松结构的三维纹理。云的绘制使用的是光线步进法，从视点出发沿视线步进，若位置在云纹理的区域内，则进行深度的累计。最后根据该视线方向上的云的累计密度，以密度为参数将云的高光和低光做线性混合得到云的颜色，在片元上进行绘制。为了实现云的光照效果，我们需要知道云内一个点的厚度。我们沿光线方向取一定距离的位置的纹理深度，若与该点本身的深度值相差很大则说明这个点的云很薄，从而以云的厚度作为光照颜色的强弱参数。