

十三、终章

本章将作为系列文章的最后一章，我们将前面的知识点汇总整合，编写一个基本可用的 3D 引擎。

这里我们可以对其做一些优化，前面几章对于屏幕空间中的线性插值方式都是采用的重心坐标插值方法，我们可以修改成双线性插值来提升程序的运行速度。在多边形裁剪的时候，我们都是使用了 STL 提供的动态容器来存放裁剪结果，动态容器的速度相对固定容器还是略慢的。实际上我们知道，一个凸 n 边形，被一个线性边界所采集，最多只能变成 $n+1$ 边形，所以每次裁剪的时候我们可以创建一个长度为 $n+1$ 的数组用于存放裁剪结果，比如第一次裁剪，我们创建一个长度为 4 的数组用存放三角形裁剪结果，下一次创建一个长度为 5 的数组，注意在裁剪 n 边形的时候不一定是变成 $n+1$ 边形，有可能存在边数量减少的情况(某条边的两个顶点都被拒绝)，所以我们还需要保存每次裁剪之后的边数量。