



链滴

# 游戏引擎的认识

作者: [xu365082218](#)

原文链接: <https://ld246.com/article/1501542074806>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

个人对U3D引擎的认识应该还是比较偏颇的

因为U3D的功能太多，所以掌握起来，很难度量掌握程度。

但是实话，现在觉得越是封装的东西多的引擎，问题和坑也就越来越多。

拿U3D来说，很多东西，类似在 U3D4的时候 打包assetbundle之类的，需要自己写依赖链分析之类工具

到U3D5，他自己把自己的一些4版本的不合理的地方给改了，然后又推出了新打包系统，这里面要熟又不知道有多少坑

其实，最简单的方法是，公开资源的格式

但是一个闭源的引擎，由于涉及到商业化，设计到他的利益的时候，他都会封闭起来

封闭起来，实际上就是给别人踩坑

还有各种小问题

类似动画系统，今天推一个这，明天推一个那。实际上更好的是了解其中的原理，往往这些东西才是稳定少发生变化的

说动画 现在指代的基本是 骨骼动画，在 顶点受到几个骨骼权重，这个基础不变的情况 来算动画帧时所有的骨骼动画系统，那就都是这样算的。

除非有一天有人发现新的骨骼动画算法 而不需要用到与多个骨骼的权重来计算动画帧。

这与学习微软的，和学习linux其实挺相似

学习微软的，就要不停的接触了解学习微软的新技术,而且更新的速度还是比较快的

学习linux的，就是知道各种不同的原理后去利用新的东西

学习U3D的，就会总是接触U3D提供的各种新技术。新技术只告诉怎么用，因为内部太复杂，也无法诉那么多细节和原理

学习COCOS的，就会接触到各种技术，如果有兴趣，还可以了解内部实现。

学习第三方的，就会被第三方牵着鼻子走

学习数学原理，那么离开了他，我们还是可以继续走

在用U3D模拟流星蝴蝶剑时，深刻的认识到，很多技术，最终面对的都是数学问题.而到U3D这里来，都不会了解有这些数学在里面，可以说被隐藏的内部实在太多了

在流星蝴蝶剑那个时代，连U3D都没有，那些昱泉开发流星的，都是用自己的思考来解决那些飞檐走，人物控制问题，这个程度，还是在15年前，可以说是比较匪夷所思了。