**代码无错便是优？真正踏上架构设计之路！**

1. 对象池模式：弹幕游戏必行优化

弹幕游戏往往涉及大量子弹对象的生成与销毁。对象池模式是面对此类场景的经典优化。预先创建子弹管理器类，设为单例，并维护一个包含预先生成子弹的容器（如栈）。需要子弹时将对象激活并取出，销毁对象时将对象冻结并加入，以此回避对象大量生成销毁之开销。

1. 基于对象池模式：使用委托分离弹幕对象与行为

由于弹幕游戏的特殊性，子弹的行为逻辑往往相当复杂。在传统的面向对象程序设计方法中，我们声明一个子弹基类（可以是抽象类）与大量具体子弹子类。利用多态性可以统一以父类的名义执行子类的特有行为。然而此思路无法利用对象池进行优化，因为往往无法保证取出的子弹是需要的类型，且父类子类之间的类型转换往往无法进行。考虑对每一种子弹维护单独的池，然而由于游戏性质子弹种类繁多，此方法效率退化严重。因此考虑利用委托，在子弹类中声明更新事件的委托，子弹生成器生成子弹的同时将想要的更新方法注册进委托，此后在生命周期内对子弹的委托进行逐帧调用，冻结时将委托清空。以上思路使子弹的对象与具体行为相分离，而子弹对象完全相同，因此可以用一个对象池进行高效的维护。

1. 曲线激光：基础数据结构与高级绘制方法实践

弹幕游戏中一重要的游戏要素即为激光与曲线激光，声明链表头类，维护子弹序列，获取子弹位置并且使用Mesh进行纹理到坐标的对应，从而实现曲线激光的效果。

1. 特效实现：缓动插值函数的使用

良好的画面效果往往预示着顺滑的移动，顺滑的移动离不开插值函数的使用。

利用正弦函数进行插值

1. 属性与字段：

利用属性“伪造”成员方法与进行信息更新

利用属性的访问器实现无参数成员方法的定义

同时由于弹幕的属性为单独的结构体

因此可以在该属性的访问器中进行数据的同步

1. 利用现代IDE的力量：xml文档注释、变量命名规范与统一、可读性为王