boss分身逻辑详解

附件为2个类算法、注意这只是为了说明逻辑，真正实际应用请结合实际架构算法。这不是可以运行的程序，也不是真正的应用算法。

BossNew为boss本体的部分说明逻辑，而BossOther则为分身逻辑。

本文档仅用于对Boss设计方案的补充，并解说两个逻辑的部分功能方便阅读。具体方案请结合原文档。

## Bossnew

BossNew不包含虚弱逻辑，只有weak变量判定是否虚弱。BossNew内也不含技能的选择判断。技能判断之后默认直接调用private void Boss\_Attack(int damageType) 。

BossNew可以分为分身前与分身后，分身前主要是private void Boss\_Attack(int damageType)对于分身的创建和transform定义。并且主动开启分身技能。分身后的处理函数主要是public void Boss\_OtherReturn(BossOther bossOther)。这个主要用于分身后boss位置恢复逻辑、boss数值判定逻辑。

需要调整数值的接口基本都是public变量。

注意boss护盾上限我默认是2。而boss护盾上限这个应该正常留出策划可以调整的接口。

## BossOther

BossOther内方法基本只会更改这个实例内部的数值，而不会去影响本体或者其他BossOther实例。

BossOther内基本逻辑其实和Boss第一阶段差不多。抛开和本体的通信其他改动很小。

激光射击的颜色是固定值几乎是与boss本体第一阶段的唯一差别。

BossOther唯一注意的是调用了本体BossNew.Boss\_OtherReturn(this)的方法来标志boss分身阶段的结束。