BUFF

通常指游戏中由实体间交互引起的用来更改实体数据的实体。

一个通用的BUFF系统如何定义BUFF元数据？

根据具体需求的差异，BUFF元数据自然分门别类。

业界常用的元数据：

1：ID

2：表现相关数据向：

名称

详细信息（描述）

图标

3：表现相关配置向：

是否显示图标

是否显示文字提示（ID为XXXX或0）（对应消息提示系统）

4：生命周期：

寿命（-1为永久，其他为计时）

生命周期计时方式（挂载时/生效时）

同实体转生间隔（冷却时间）

死亡方式：

切换场景移除

载体死亡移除

完成关卡移除

更换装备移除

提前结束方式 + 参数

5：BUFF从出生到挂载规则方面：

BUFF种类（增益、减益）

BUFF类型（控制、回复、伤害、属性增强）

组别ID

触发方式Enum + 参数组（可以唯一确定触发条件）

触发目标状态条件Enum + 参数组（确定释放者是否可以释放）

挂载目标状态条件Enum + 参数组（确定挂载者是否可以挂载）

作用目标类型

作用范围

作用区域ID（对应区域系统）

作用目标条件Enum + 参数组（确定是否可以挂载）

最大作用数量

目标优先级

作用效果生效几率

6：BUFF效果：

属性效果参数组：

属性效果类型

固定值

固定值随等级

百分比

百分比随等级

属性效果延迟

特殊作用效果：

特殊作用效果目标状态条件Enum + 参数组

特殊作用效果：

效果类型 + 参数 + 效果延迟 参数组

移除作用效果：

效果类型 + 参数 + 效果延迟 参数组

最大叠加层数

叠加额外效果层数

叠加额外效果类型:  
 效果类型 + 参数 + 效果延迟 参数组

限定BUFFFlag

变更时移除

7：音效相关：

挂载时

作用时

移除时

8：组：

同源组内冲突

同源自身冲突

不同源组内冲突

不同源自身冲突

Act游戏案例：雷锤龙释放雷降

1:PartCollisionBaseMonsterAttack ：PartCollisionBase 触发器发现可攻击目标player

2:HeroCollisionSystem 根据玩家状态看看打得怎么样

3:如果能打上，计算伤害相关（BUFF啊，能不能打死呀）

1：有BUFF就挂载，由BuffManager进行源、挂载者BUFF处理

2：挂载者进行BUFF挂载---->挂载成功/失败

3：挂载成功的话开始数据管理 + BUFF 挂载start逻辑处理

1：BUF FSTART ----> trigger ----> preform 进行表现处理

2: 发放给服务器通知所有人

4:计算伤害的方向

网络同步规则相关

调用BuffManager挂载函数AttachBuff（源、载、buffConfigID）回动态ID

给所有效果实体发送消息（判别该载体是否可以挂载，活动的物体对象都是由这种效果实体去控制属性表现的，查看该载体是否拒绝挂载此ID的buff，比如免疫效果实体，需要传递的参数看效果实体需求而定）

得到/创建挂载者

调用挂载者方法：AttachBuff（源、载、buffConfigID）

得到表格BUFF

得到挂载者实体

判断挂载者状态是否允许添加此BUFF（由状态和BUFF字段决定），可能返回

判断BUFF之间的冲突，可能返回

加上去AddAttachCmd

通过Buff工厂制造Buff的动态唯一ID

向服务器发送广播

调用AttachBuffInternal

通过Buff工厂创建BUFF

记录BUFF时间

Buff挂载触发调用

计算状态

返回BUFF

记录BUFF技能ID

发放BUFF更新事件

返回动态ID

返回动态ID

返回动态ID