

# 多人联机射击游戏总结

## 游戏流程:

服务端的GameMode负责核心游戏事件，通过修改ServerGameState的数据来驱动游戏流程变化。

## 依赖属性同步导致的一些Bug:

断网重练期间队友阵亡，回到游戏瞬间看到队友重复死亡

原因:

判断和触发角色死亡是依赖角色的生命值组件的回调，角色的生命值HP默认等于预设的最大生命值MaxHP。当客户端A断线重连到服务器后，阵亡队友的HP从MaxHP同步为0，由于角色构造时的默认状态为存活并且站立，因此会导致原地再次死亡。

正确做法：通过RPC触发玩家死亡逻辑

## 依赖RPC导致的一些Bug:

客户端击杀之后使用旧战绩更新记分面板UI

击杀之后，可能属性同步的最新值晚于RPC，导致RPC中的参数并不是最新的KDA数据，并未完成同步，导致延迟

正确做法：通过属性同步的回调触发UI更新，确保数据同步之后再刷新UI

## 总结:

属性同步注重数据，可能会丢失中间过程，也就是丢失产生变化这一事件，RPC重视逻辑，也就是变化本身。对于流程状态，用枚举值或者布尔变量来记录，使用属性同步记录是最佳选择

## 时序问题:

本质原因是在GamePlay的流程中很多资源并未来得及初始化，访问为空，有很多类似的问题。解决方法是不要假定BeginPlay是最早执行的，需要确保对方创建时有一个主动去拿数据设置的流程，可以依赖回调函数完成资源的初始化以确保正确初始化

## 角色死亡后抽搐:

角色死亡时组件的碰撞信息并不是自动同步的，需要手动保持一致，双端碰撞信息不一致导致客户端的角色位置频繁被修正

## UI管理:

PC负责声明/销毁/隐藏UI，Pawn负责创建和显示UI，如果角色死亡，通过委托通知PC隐藏UI，Gun本身负责更新自己，因为Gun可以第一时间感知弹药的数量变化

