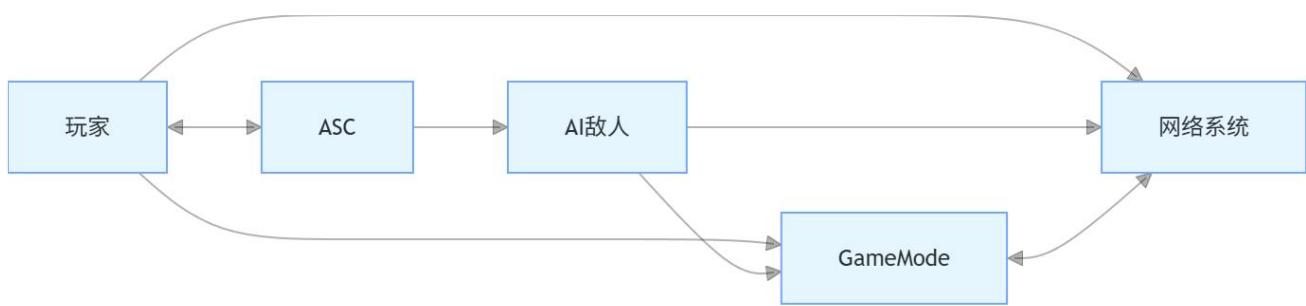


战斗系统核心参与者分析

参与者名称	角色定位	核心职责	与系统的主要交互	设计重点
玩家	核心外部交互者	控制角色进行战斗的人类用户	1. 输入战斗指令 2. 管理棍势蓄力 3. 选择战斗策略 4. 体验战斗反馈	操作流畅性、战斗深度、视觉反馈
AI 敌人	次要外部交互者	游戏中的敌对非玩家角色	1. 执行战斗行为 2. 与玩家角色战斗交互 3. 响应玩家攻击 4. 评估战斗态势	智能决策、行为多样性、挑战性平衡
ASC (能力系统组件)	核心功能组件	管理角色战斗能力的核心组件	1. 管理角色能力库 2. 处理能力激活与执行 3. 应用能力效果 4. 管理能力冷却 5. 处理棍势蓄力机制	模块化设计、性能优化、扩展性
GameMode	系统控制者	管理游戏进程和战斗规则的核心模块	1. 初始化战斗场景 2. 控制游戏规则 3. 处理战斗结果 4. 统计战斗数据 5. 管理多人对战匹配	规则一致性、流程控制、数据统计
网络系统	扩展功能组件	支持多人对战的网络通信系统	1. 处理玩家间网络通信 2. 同步战斗状态 3. 确保多人对战流畅性 4. 处理网络延迟和同步问题	低延迟、高可靠性、状态同步准确性

参与者交互关系



各参与者的关键实现细节

1. 玩家:

- 输入通过 InputComponent 映射到对应的 GameplayTag
- 角色类: ABMPlayerCharacter (继承自 ABMBaseCharacter)
- 支持多种输入设备 (键盘鼠标、手柄、触摸屏)

2. AI 敌人:

- 角色类: ABMNPCharacter (继承自 ABMBaseCharacter)
- AI 控制器: ABMAIController, 实现战斗决策逻辑
- 行为树: 定义 AI 敌人的行为模式和决策流程

3. ASC (能力系统组件) :

- 类: UBMAbilitySystemComponent (继承自 UBMASCBase)
- 管理 UBMGameplayAbility 实例和 UBMGameplayEffect
- 实现棍势蓄力机制的核心逻辑

4. GameMode:

- 类: ABMGameModeBase 或 ABMCombatGameMode
- 管理战斗匹配、开始/结束逻辑、胜利/失败条件
- 处理玩家加入/离开游戏的逻辑

5. 网络系统:

- 基于 UE5 的网络框架实现
- 关键状态通过 ReplicatedProperties 或 RPC 同步
- 支持客户端预测和服务器校正, 确保多人对战的流畅性

这些参与者共同构成了 BM 战斗系统的生态核心，覆盖了外部用户、核心功能组件、系统控制者和扩展功能组件，确保了系统功能的完整性和特色的实现。