

# 《生与死：语音交流与 AI 托管系统》产品设计文档 (PRD)

产品架构视角

2026 年 4 月 9 日

## 1 产品愿景与核心玩法 (Core Vision)

本系统的核心理念在于“**躯体的延续与灵魂的隔离**”。在多人合作/对抗场景中，玩家的角色 (Avatar) 不会因为玩家死亡而消失，而是由 AI 智能体接管躯干继续战斗。同时，真实玩家的“灵魂”（即语音交流与操作控制）将被强制剥离当前战局，以保证战术的公平性并提供独特的观战体验。

## 2 核心体验旅程 (User Journey)

我们把单局游戏分为三个阶段进行设计：

### 2.1 阶段一：存活期 (The Living)

- **控制权**：玩家拥有 100% 的角色控制权 (Human Input)。
- **语音策略**：玩家加入 GVoice 的“存活者小队房间” (**Room\_Alive**)，能够与存活的队友实时交流战术。
- **UI 表现**：正常显示血条、技能 CD 和语音波纹。

### 2.2 阶段二：死亡瞬间 (The Transition)

- **触发条件**：角色生命值归零（或触发特定淘汰机制）。
- **控制权剥离**：系统立刻禁用玩家的输入脚本 (Input Controller)，玩家无法再操纵角色移动或开火。
- **语音隔离**：为防止“死人报点” (Ghosting, 破坏竞技公平性)，系统通过 GVoice SDK 立刻执行 `CloseMic()` 或 `QuitRoom("Room_Alive")`。
- **视觉反馈**：屏幕短暂灰态或出现“系统已托管”的 UI 动效，让玩家明确知道自己已经“死亡”。

### 2.3 阶段三：托管期 (The AI Takeover)

- **AI 唤醒**：角色的底层控制权移交给 AI Controller（如行为树或有限状态机）。AI 瞬间“复活”或维持满血/残血状态，继续在战场上执行基础逻辑（如：跟随存活队友、攻击最近敌人）。

- **亡语频道**：为了保证阵亡玩家的留存率和参与度，系统将阵亡玩家的 GVoice 接口接入“亡灵房间” (Room\_Dead)。死去的玩家可以在这个专属频道里复盘、聊天，看着自己的 AI 在战场上奔跑。
- **身份标识**：在存活玩家的视角中，该角色的头顶 ID 颜色需发生改变（例如从绿色变成灰色），或者加上 [BOT] 前缀，明确告知队友“这已经是个人工智能了，不要指望它听懂复杂的战术”。

### 3 基于 GVoice 的通信逻辑流

根据你在前期 PoC 中打通的 GVoice 链路，产品上的流转逻辑如下：

#### 1. 游戏开局：

- 调用 `GVoice.JoinTeamRoom(RoomID_Alive)`。
- 进房回调成功后，调用 `GVoice.OpenMic()` 和 `GVoice.OpenSpeaker()`。

#### 2. OnPlayerDeath() 触发：

- 调用 `GVoice.CloseMic()`，掐断输入。
- 调用 `GVoice.QuitRoom(RoomID_Alive)`，退出生者频道。

#### 3. 进入亡语频道（可选功能）：

- 延迟 0.5 秒（等待退房回调）。
- 调用 `GVoice.JoinTeamRoom(RoomID_Death)`。
- 进房成功后，重新调用 `GVoice.OpenMic()`，让死者之间可以交流。

### 4 AI 托管的拟真度与接管策略

作为多智能体系统的应用延伸，死亡瞬间的 AI 接管需要避免“生硬感”。产品设计上建议采用“惯性状态继承”策略：

- 如果玩家在奔跑中死亡，AI 接管后的前 1-2 秒应继续保持原来的奔跑方向（惯性），然后再由 NavMesh 寻路系统接管计算新路径。
- 如果玩家在开火状态下死亡，AI 接管后应当打完当前弹夹，再进入掩体寻找逻辑。
- 这种设计能够极大程度消除突兀感，让局内的存活玩家感觉该角色依旧“有血有肉”。

### 5 边界异常处理 (Edge Cases)

- **断线重连**：如果玩家并非血量归零，而是网络断开 (UDP/TCP 掉线)。系统同样触发 AI 接管，但一旦玩家重连成功，系统需立即剥夺 AI 控制权，将 Input 权限交还给玩家，并重新将玩家拉入 Room\_Alive。
- **全军覆没**：当最后一个真人玩家死亡时，无需再进行 AI 托管，游戏直接判定结束，展示结算 UI 并允许所有人开启全屏麦克风复盘。