

# 左涵俊-四川大学-开局一课客户端大作业技术说明(29届)

UE5 First Person 多人协作清理训练场 Demo

## 1. 项目概述

本项目基于 Unreal Engine 5.7 官方 First Person 模板开发，实现了一个可演示的多人协作清理训练场 Demo。玩家进入训练场后使用第一人称射线武器攻击敌人；敌人会在服务器端寻找最近的存活玩家，使用 NavMesh 追击并播放攻击动画；敌人被击败后增加队伍分数，达到目标数量后所有玩家看到胜利提示。

Demo 支持单人模式和本地 Listen Server 双人联机模式。多人测试时，一个客户端选择创建房间，另一个客户端选择加入游戏，人数达到 2 人后由 Lobby 自动切换到战斗关卡。

## 2. 作业要求对应关系

作业要求	实现内容
会移动和攻击玩家的敌人	敌人服务器端索敌，使用 AI MoveTo 追击玩家，并在攻击范围内播放攻击动画和造成伤害。
玩家可以击败敌人	玩家左键射击调用 ServerFire，服务器执行射线检测和 Apply Damage，敌人血量归零后销毁。
基础得分和胜利机制	GameMode 修改 GameState 中的 TeamScore，达到 TargetScore 后设置 GameWon，HUD 显示胜利。
多人网络对战	Listen Server 模式，伤害、AI、死亡、得分和胜利由服务器权威处理，并通过 Replication/RPC 同步。

## 3. GitHub 链接与多人网络实现要点

GitHub 仓库: [https://github.com/Left-Jun/Left\\_Jun-Unreal-Afterclasswork\\_Tencent2026](https://github.com/Left-Jun/Left_Jun-Unreal-Afterclasswork_Tencent2026)

Demo 视频随提交文件一同上传，文件名为 左涵俊-四川大学-开局一课客户端大作业.mp4。

项目使用 Listen Server 模式进行多人演示。服务器负责敌人

AI、伤害、死亡、重生、得分和胜利判定；客户端负责本地输入、UI 显示和本地射击视觉反馈。

- 玩家射击：客户端输入后调用 ServerFire，由服务器执行射线检测和伤害。
- 射击反馈：客户端本地执行 LocalFireVisual 绘制射线，避免等待网络往返。
- 敌人 AI：敌人在服务器 Authority 分支中定时索敌和移动，位置通过 Actor Replication 同步。
- 敌人攻击动画：使用 Multicast RPC 播放攻击动画，保证所有窗口表现一致。
- 玩家死亡 UI：服务器判定死亡后调用 ClientShowDeathMenu，只在死亡玩家自己的客户端显示死亡菜单。
- 玩家重生：复活按钮调用 ServerRespawn，由服务器恢复玩家状态并传送到 PlayerStart。
- 分数与胜利：保存在 GameState 中并复制给所有客户端。

## 4. 玩家系统

### 4.1 服务器权威射击

玩家射击基于 First Person 模板扩展。按下鼠标左键后，客户端先执行 LocalFireVisual，在本地绘制射线反馈；随后调用 ServerFire RPC，由服务器根据玩家第一人称摄像机位置和朝向执行 Line Trace By Channel。命中目标后，服务器调用 Apply Damage，对敌人造成 25 点伤害。

这种设计将即时视觉反馈和权威伤害判定分离：客户端能立刻看到开火射线，真正的命中和伤害由服务器统一确认。

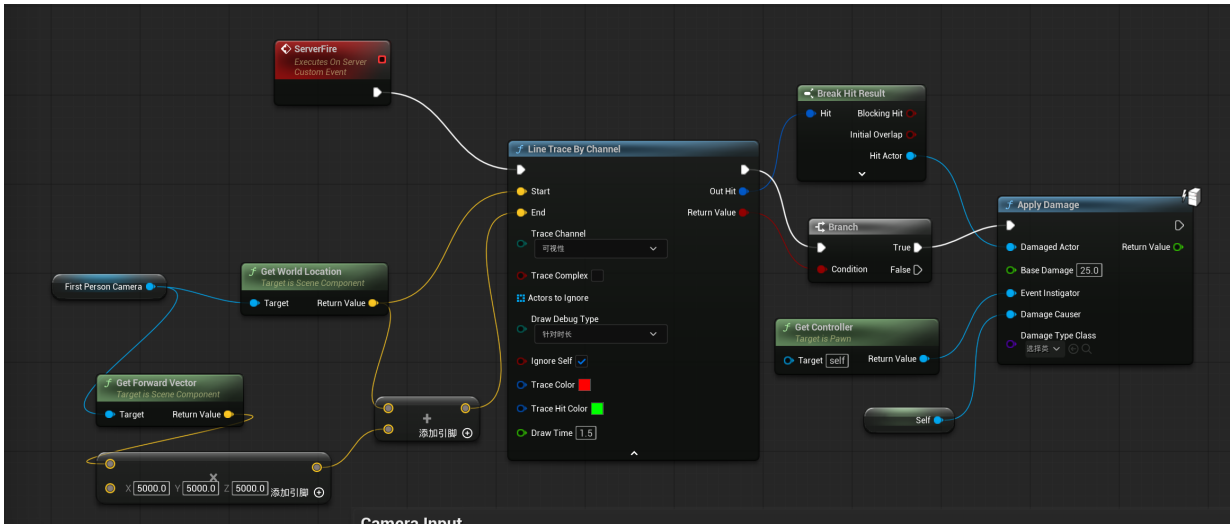


图 19: ServerFire 射线伤害

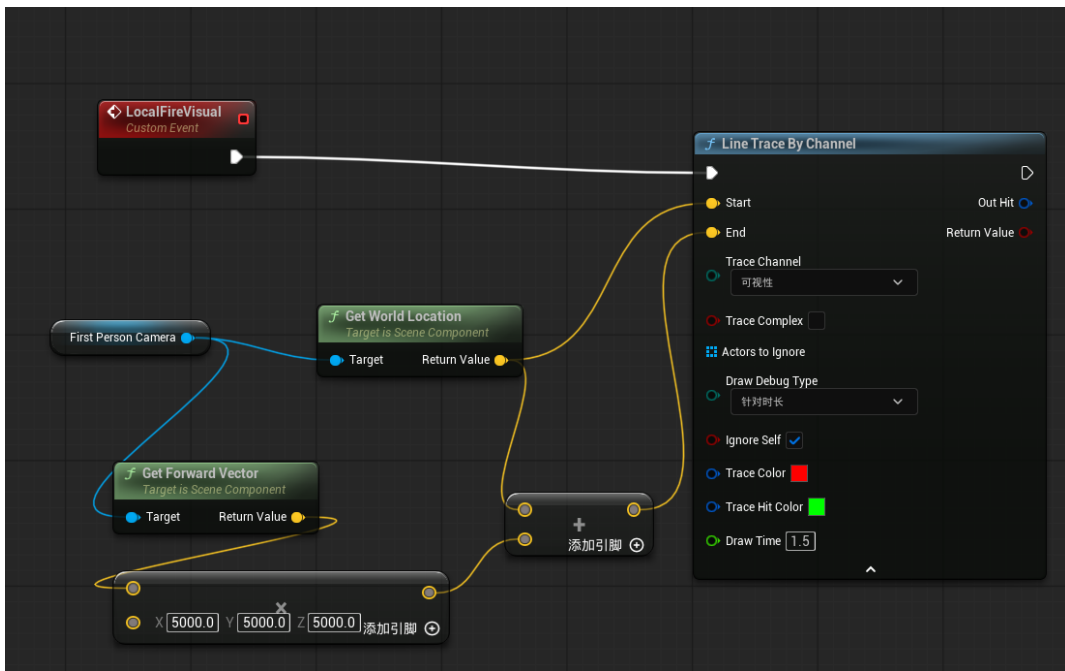


图 23: LocalFireVisual 本地射击反馈

### 4.2 玩家受伤、死亡与重生

玩家角色维护 Health、MaxHealth 和 IsDead 变量，其中 Health 与 IsDead 设置为 Replicated。敌人攻击玩家时由服务器调用 Apply Damage。玩家角色在 Event AnyDamage 中先判断 IsDead，避免死亡后重复扣血；若仍存活，则将血量 Clamp 到 0 - MaxHealth 范围内。

当 Health  $\leq 0$  时，服务器设置 IsDead = true，并调用 ClientShowDeathMenu。该事件设置为 Run on Owning Client，因此死亡菜单只会出现在死亡玩家自己的窗口中，而不会错误显示在服务器或其他客户端窗口。

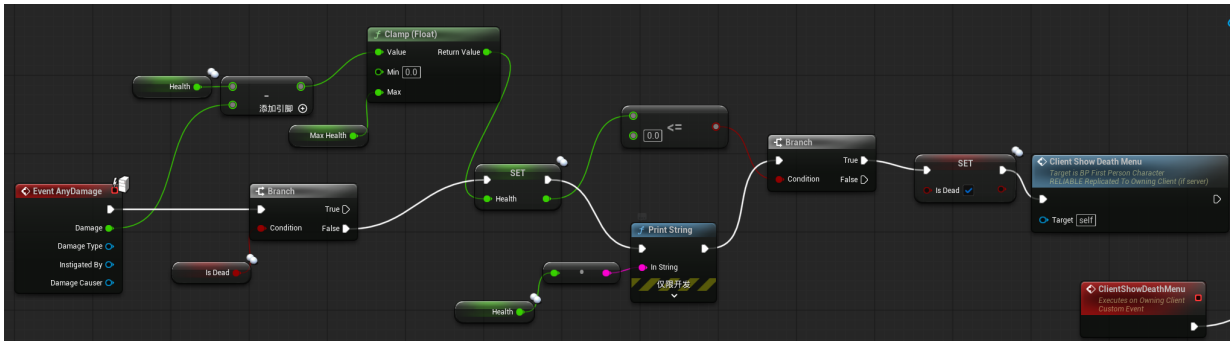


图 20: 玩家扣血与死亡判断

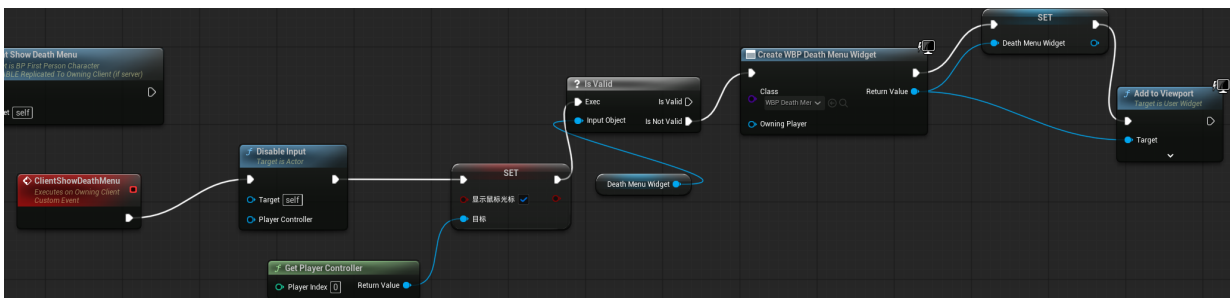


图 21: 客户端显示死亡菜单

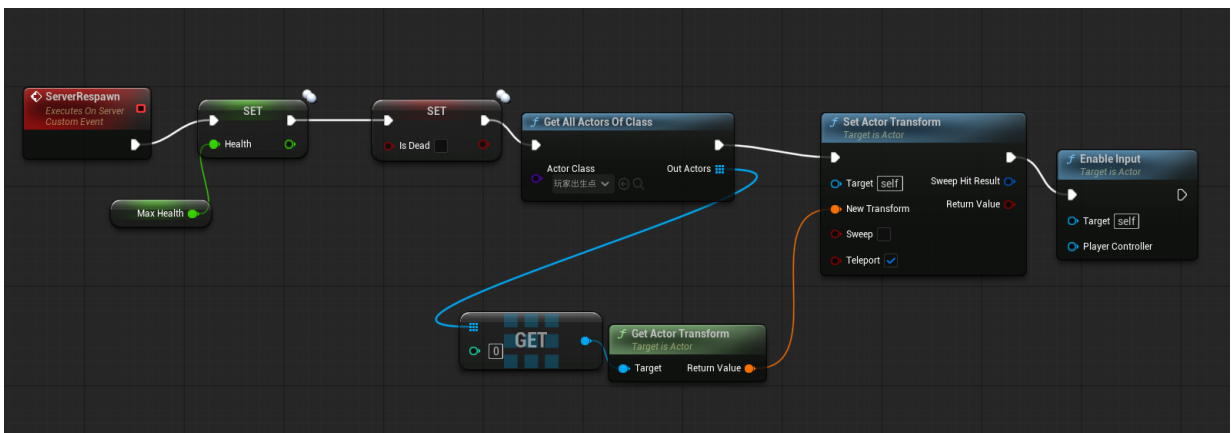


图 22: 服务器重生逻辑

### 4.3 HUD 创建与输入恢复

玩家 BeginPlay 中恢复输入模式、隐藏鼠标并创建 HUD。该步骤解决了从主菜单或死亡界面切换到战斗地图后输入模式仍停留在 UI 状态的问题。

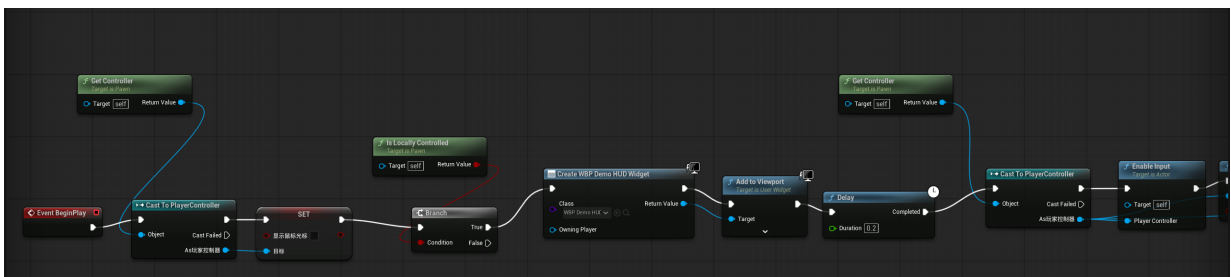


图 24: 玩家 BeginPlay 初始化

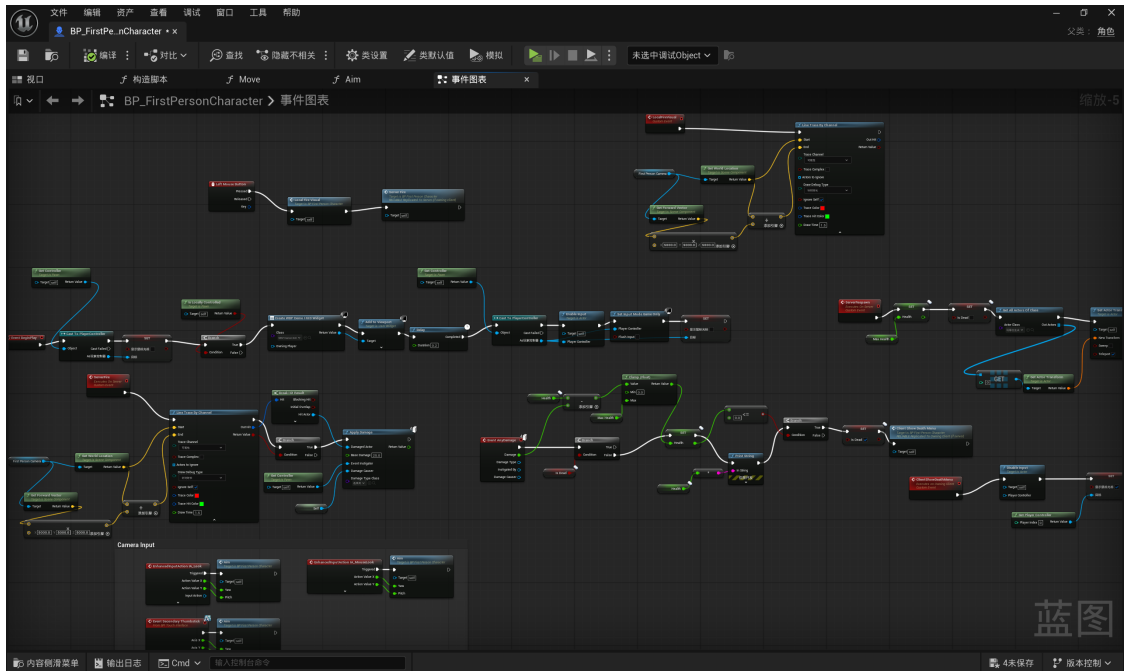


图 25: 玩家蓝图总览

## 5. 敌人系统

### 5.1 敌人受伤与死亡

敌人蓝图 BP\_EnemyCharacter 在 Event AnyDamage 中处理受伤逻辑。该逻辑通过 Switch Has Authority 保证只在服务器端扣减敌人生命值。敌人血量小于等于 0 后，调用 BP\_DemoGameMode.EnemyKilled 增加队伍分数，然后销毁敌人 Actor。

该设计确保敌人死亡、得分和胜利判断都由服务器统一处理，避免多个客户端分别计算造成状态不一致。

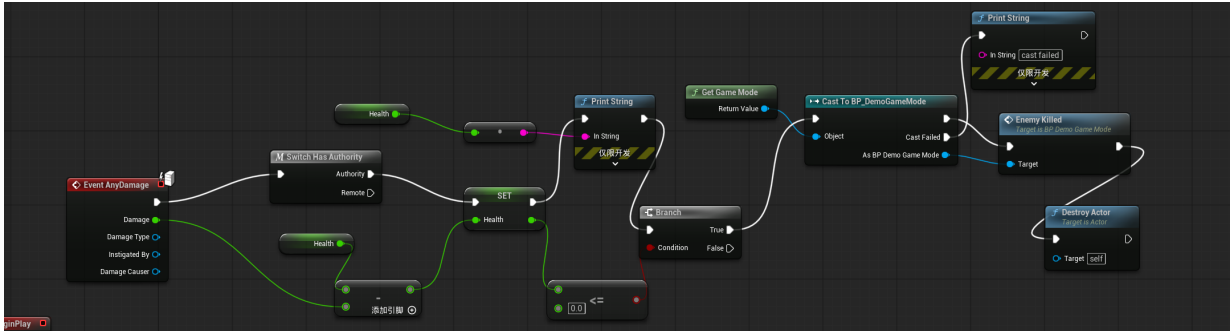


图 11: 敌人受伤与死亡

### 5.2 服务器端索敌与移动

敌人 BeginPlay 中在 Authority 分支启动定时器，周期性调用 UpdateEnemy。UpdateEnemy 遍历所有 BP\_FirstPersonCharacter，跳过 IsDead 为 true 的玩家，选择最近的存活玩家作为 TargetPlayer，并使用 AI MoveTo 追击。地图中放置 Nav Mesh Bounds Volume，使敌人能够基于 NavMesh 寻路。

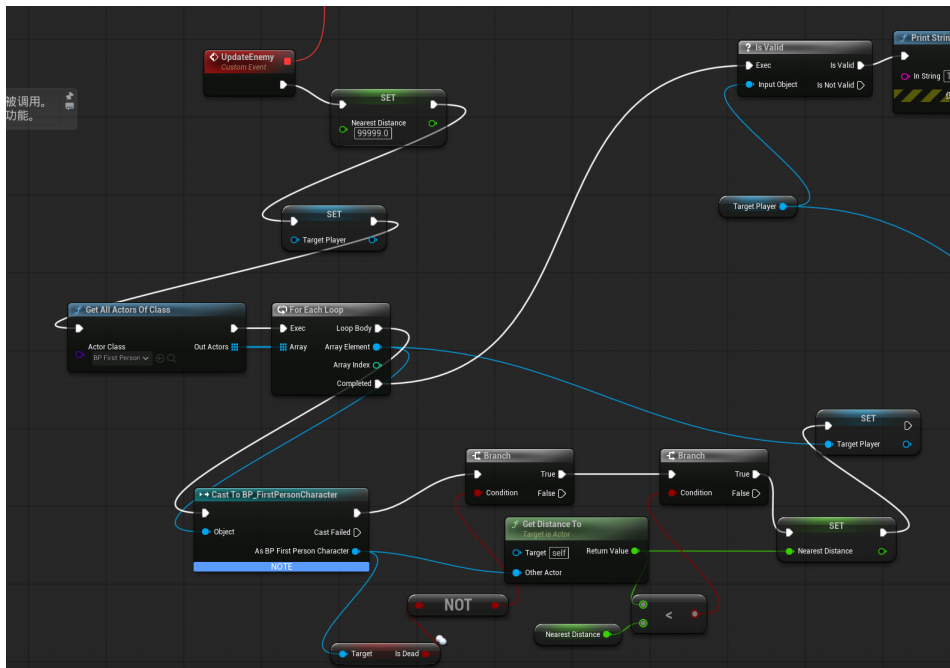


图 12: 敌人选择最近存活玩家

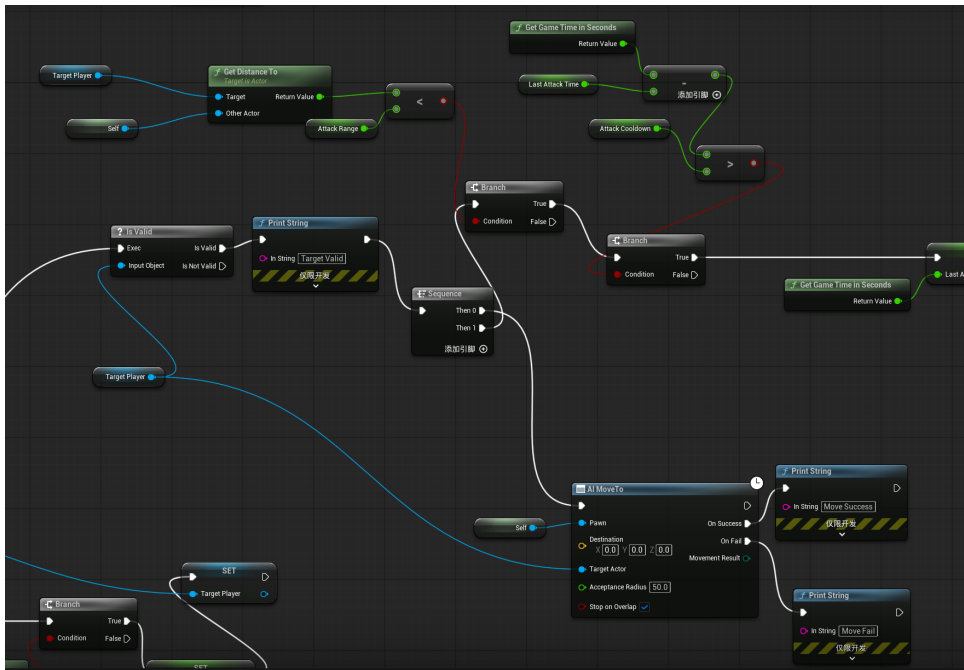


图 13: 敌人追击与攻击入口

### 5.3 攻击动画与延迟伤害

敌人进入攻击范围后，使用 LastAttackTime 和 AttackCooldown 控制攻击频率。攻击触发时调用 Multicast\_PlayAttack，让所有客户端都能看到攻击动画。动画播放后延迟约 0.35 秒，再检查目标有效且在攻击范围内，然后执行 Apply Damage。

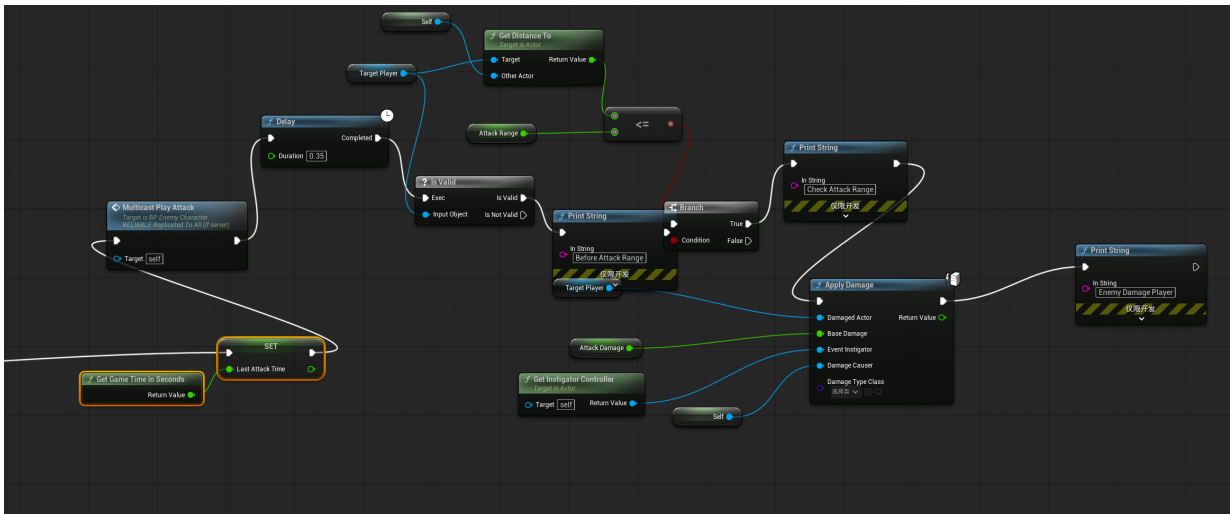


图 14: 敌人攻击延迟伤害

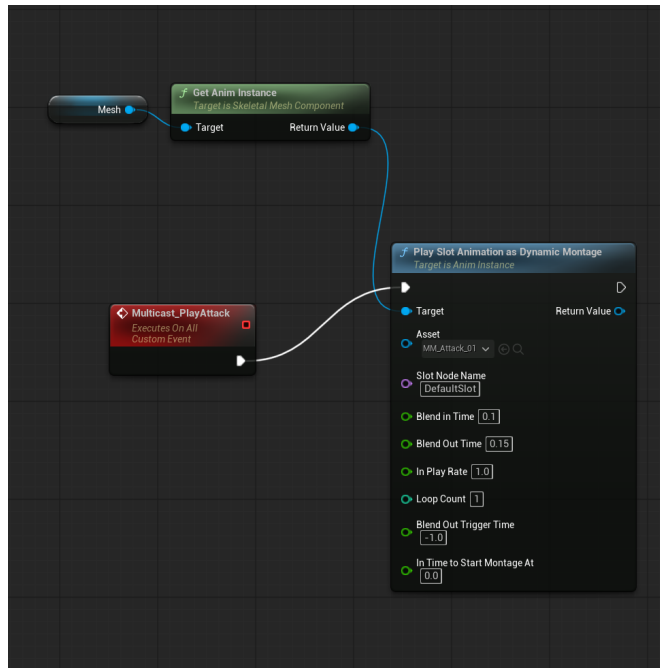


图 15: 敌人攻击动画 Multicast

## 5.4 敌人动画蓝图

ABP\_Energy 使用 Try Get Pawn Owner、Get Velocity 和 Vector Length 计算敌人移动速度，并写入 Speed。AnimGraph 中使用 BS\_Idle\_Walk\_Run 混合空间实现待机、行走、奔跑切换，攻击动画通过 DefaultSlot 叠加播放。

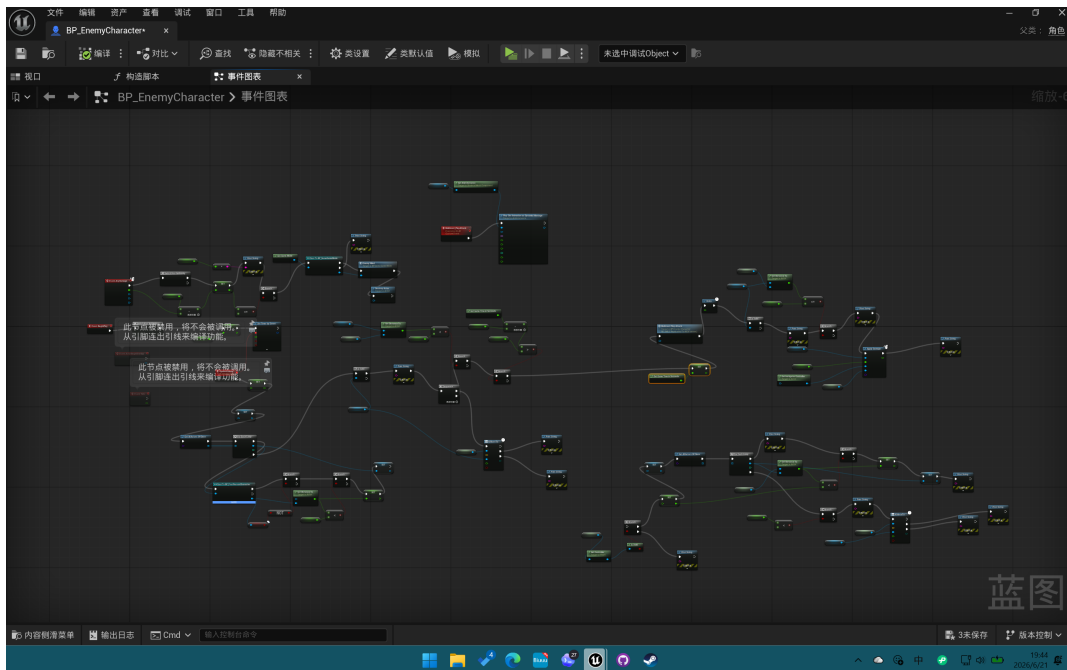


图 16: 敌人蓝图总览

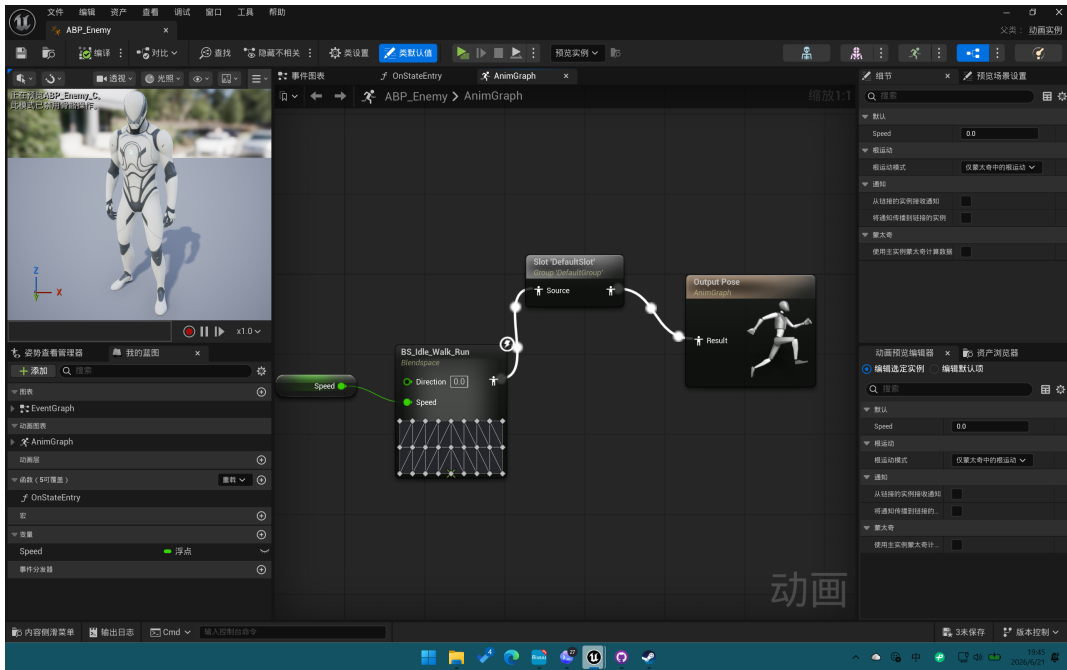


图 17: 敌人动画图表

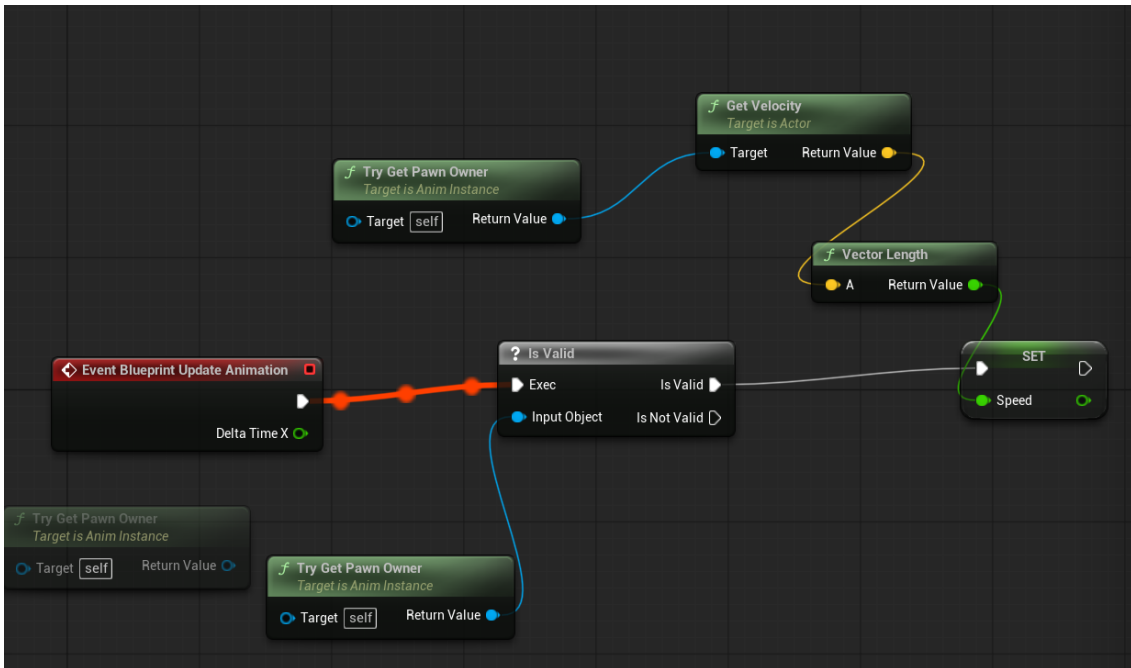


图 18: 敌人动画 Speed 更新

## 6. 联机、得分、胜利与大厅

玩家启动程序后首先进入主菜单。单人游戏直接打开战斗关卡；多人游戏通过控制台命令 `open /Game/Demo/Lvl_Lobby?listen` 创建 Listen Server；加入游戏通过 `open 127.0.0.1` 连接本机服务器。

在 Lobby 中，BP\_LobbyGameMode 通过 PostLogin 统计 GameState.PlayerArray 人数。当玩家数量达到 2 时，服务器执行 `servertravel /Game/Demo/Lvl_FirstPerson`，将所有客户端切换到战斗地图。

进入战斗地图后，玩家协作击败敌人。敌人追击最近的存活玩家，靠近后播放攻击动画并扣除玩家生命值。玩家死亡后会显示死亡菜单，可以点击复活按钮回到出生点。队伍击败 5 名敌人后，所有玩家 HUD 显示游戏胜利。

BP\_DemoGameMode 中实现 EnemyKilled 函数。敌人死亡时，服务器调用该函数，获取 BP\_DemoGameState 并将 TeamScore + 1。当 TeamScore >= TargetScore 时，服务器将 GameWon 设置为 true。GameState 中的 TeamScore、TargetScore 和 GameWon 设置为 Replicated，使所有客户端 HUD 都能同步显示击杀进度和胜利提示。

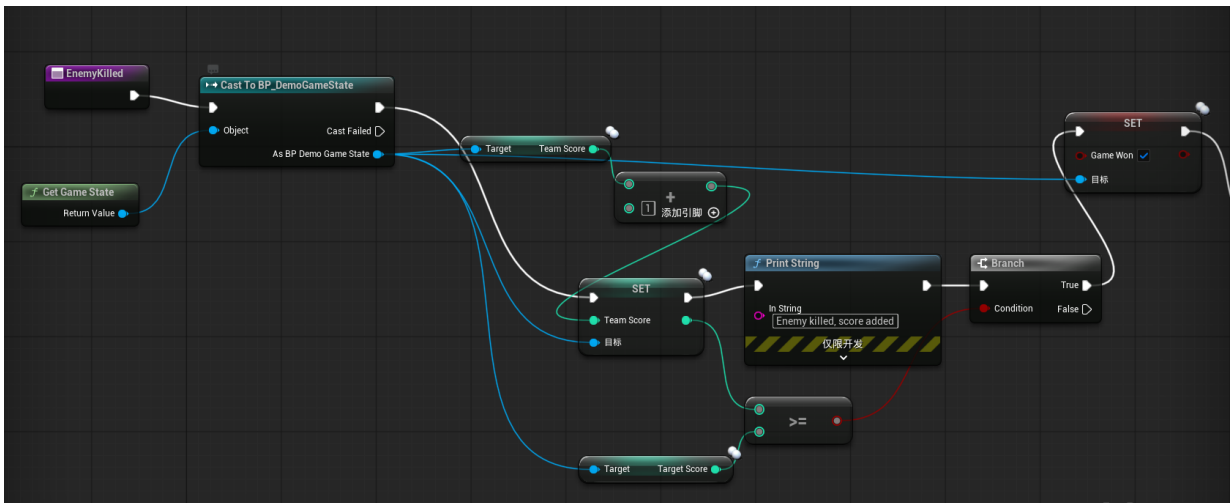


图 8: GameMode 击杀得分逻辑

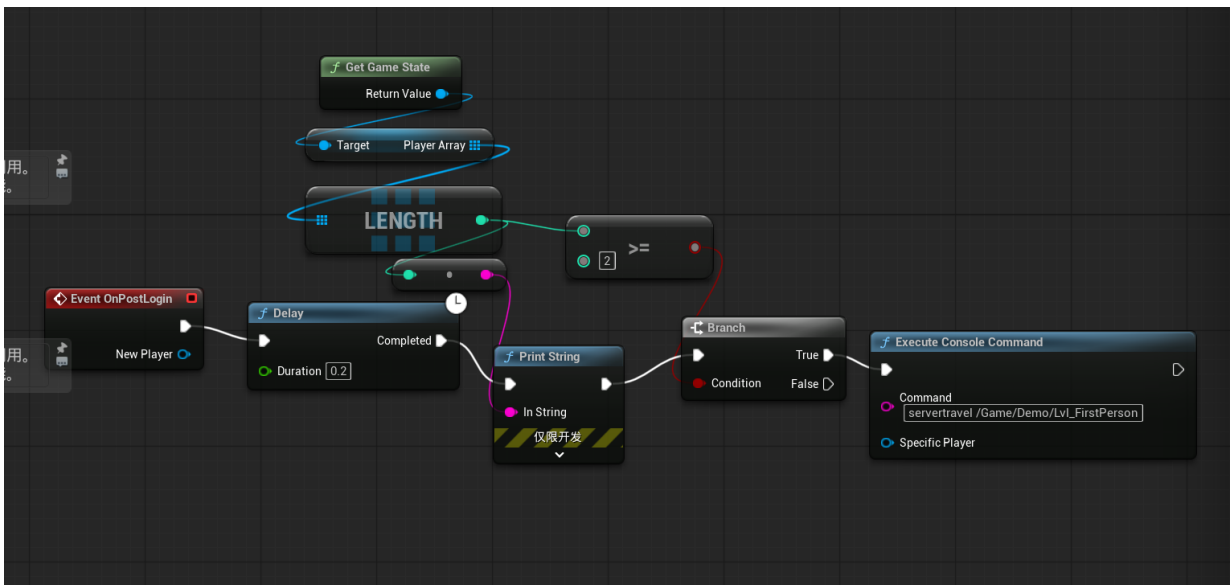


图 9: Lobby PostLogin 跳转逻辑

## 7. UI 系统

### 7.1 主菜单

WBP\_MainMenu 是项目入口界面，包含标题、作者信息、单人游戏、多人游戏和加入游戏按钮。单人按钮使用 `Open Level by Name` 打开战斗关卡；多人按钮使用 `Execute Console Command` 创建 Listen Server

Lobby; 加入按钮连接本机地址。主菜单关卡的 Level Blueprint 在 BeginPlay 中创建该 Widget, 并设置 UI Only 输入模式和鼠标显示。

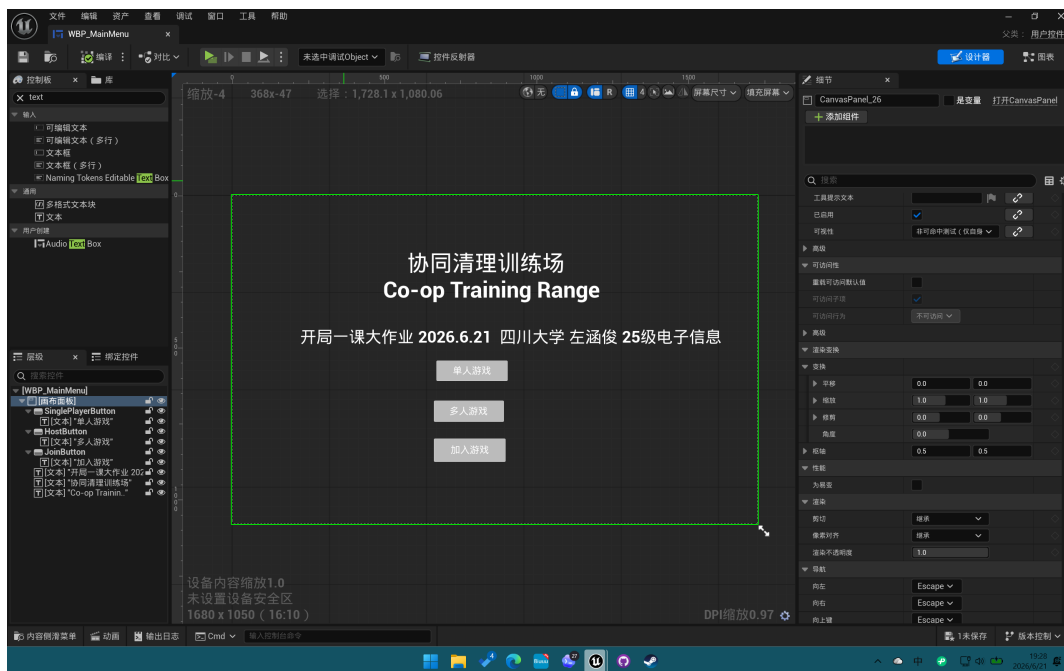


图 6: 主菜单界面

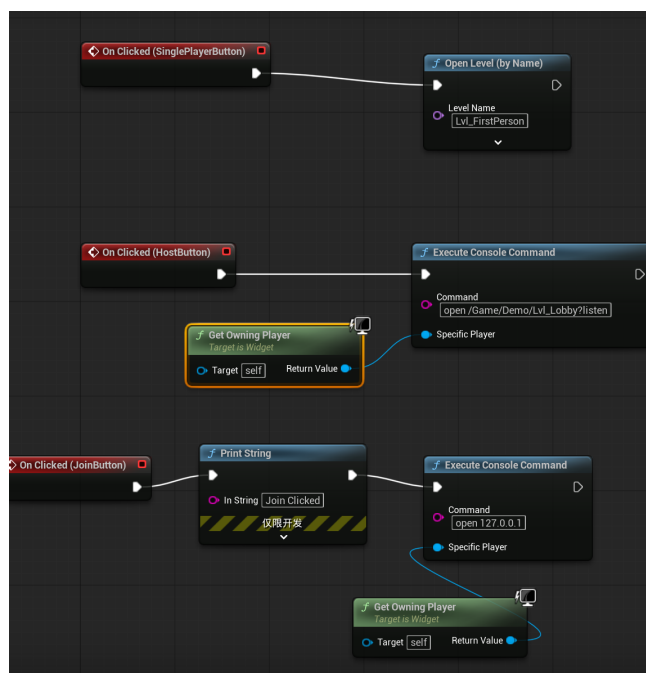


图 7: 主菜单按钮逻辑

## 7.2 战斗 HUD

WBP\_DemoHUD 显示本地玩家生命值、队伍击杀数量、胜利提示和准星。HUD 在玩家角色 BeginPlay 中通过 Is Locally Controlled 判断, 仅为本地控制的玩家创建, 避免服务器或远程角色重复生成 UI。

HUD 的 Tick 中使用 Get Owning Player Pawn 读取本地玩家的健康, 并从 BP\_DemoGameState 中读取 TeamScore 和 GameWon。这样每个客户端显示的是自己的生命值, 同时共享同一份服务器同步的得分与胜利状态。

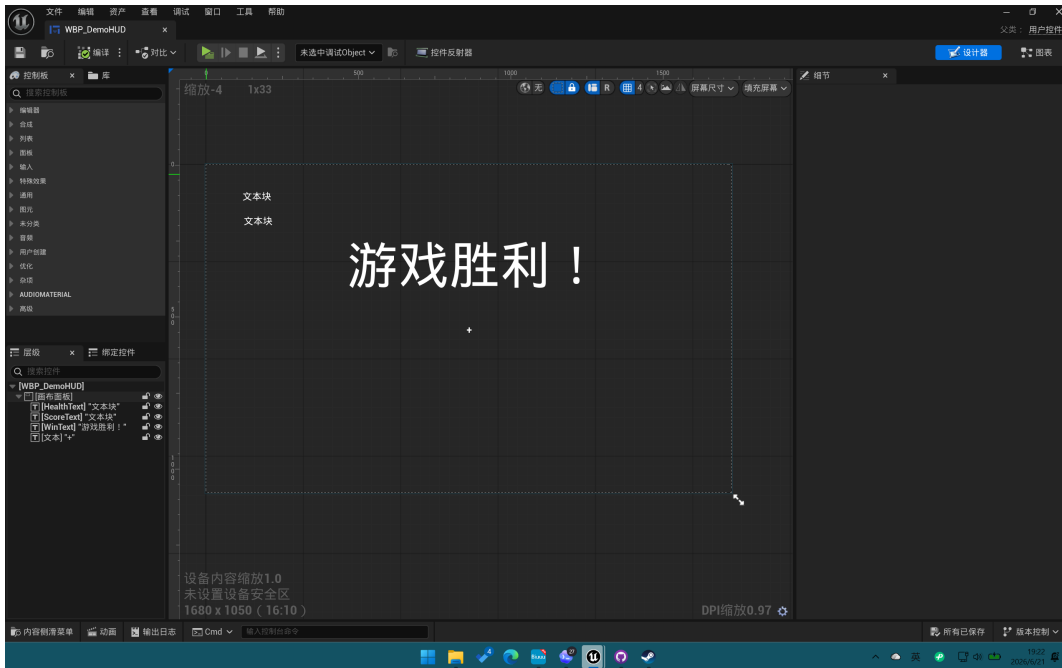


图 3: HUD 设计

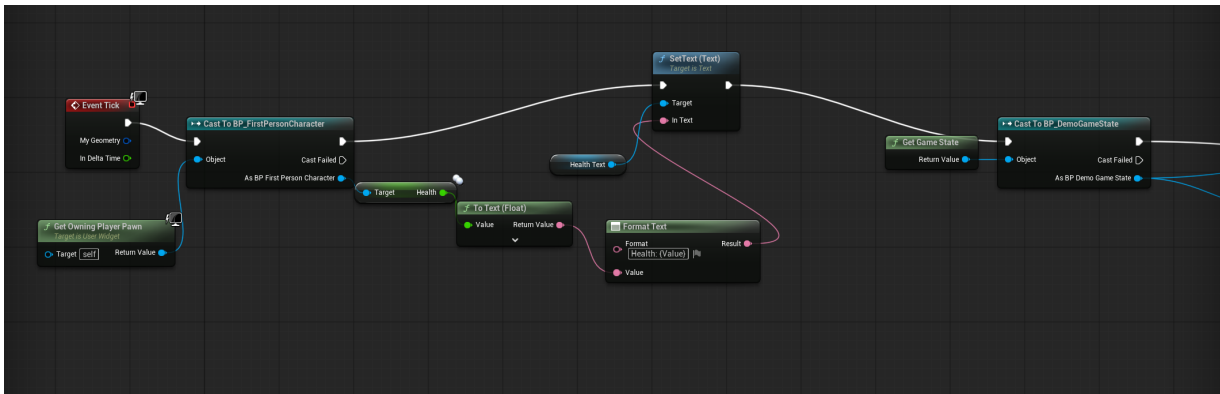


图 4: HUD 生命值更新

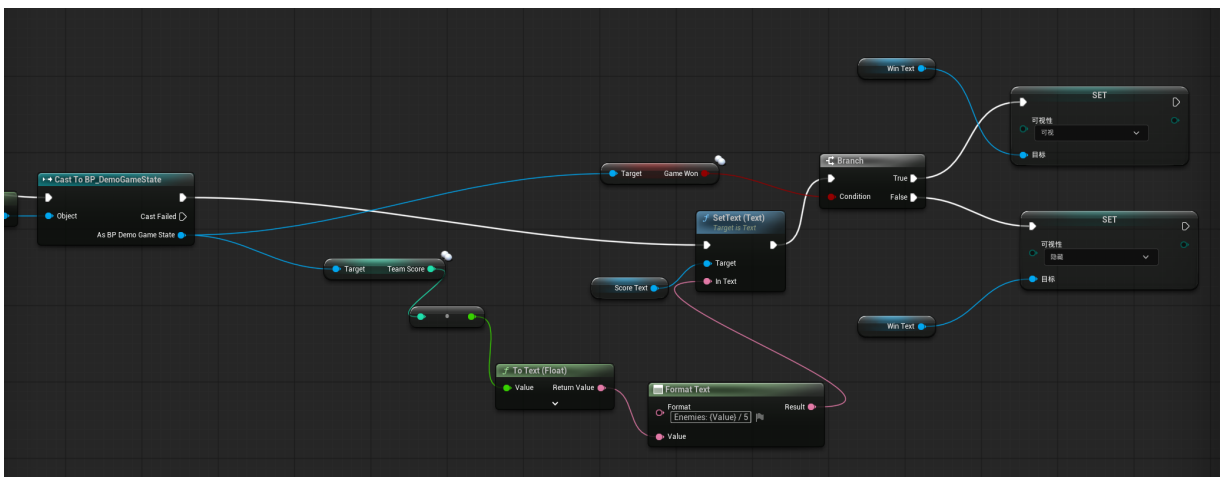


图 5: HUD 分数与胜利更新

### 7.3 死亡与复活菜单

WBP\_DeathMenu 在玩家死亡时显示，包含 You Died 提示和复活按钮。按钮点击后调用玩家角色的 ServerRespawn，由服务器恢复生命值、重置死亡状态并将玩家传回出生点；随后本地 Widget 移除自身，并恢复 Game Only 输入模式。

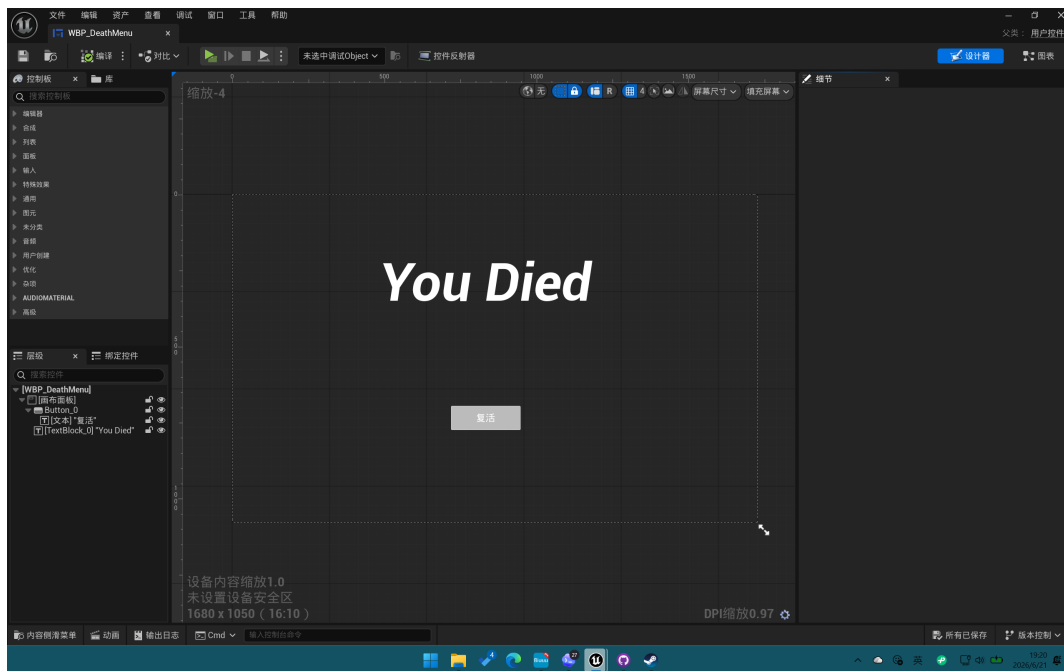


图 1: 死亡菜单设计

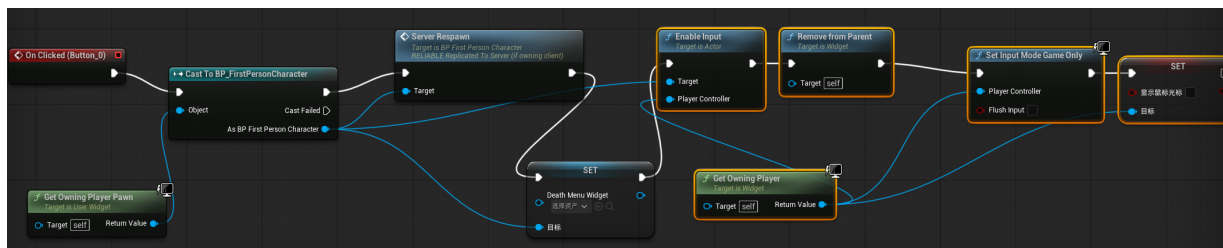


图 2: 复活按钮逻辑

## 8. 主要资源与关卡

- 主菜单关卡: /Game/Demo/Lvl\_MainMenu
- 等待房间关卡: /Game/Demo/Lvl\_Lobby
- 战斗关卡: /Game/Demo/Lvl\_FirstPerson
- 玩家蓝图: /Game/FirstPerson/Blueprints/BP\_FirstPersonCharacter
- 敌人蓝图: /Game/Blueprints/ENEMY/BP\_EnemyCharacter
- 敌人动画蓝图: /Game/Blueprints/ENEMY/ABP\_Enemy
- 游戏规则: /Game/Blueprints/GAME/BP\_DemoGameMode
- 游戏状态: /Game/Blueprints/GAME/BP\_DemoGameState
- 大厅规则: /Game/Blueprints/GAME/BP\_LobbyGameMode
- UI: WBP\_MainMenu、WBP\_DemoHUD、WBP\_DeathMenu、WBP\_LobbyMenu

## 9. 打包与测试

项目已配置主菜单为默认地图，并添加窗口化启动设置。为了避免 UE 打包时中文路径和临时 C++ target 造成的问题，项目禁用了未使用的 GameplayStateTree 插件，并将关键蓝图资源迁移到英文目录。打包配置使用 Windows Development 版本进行测试。

- 打开第一个 exe，点击“多人游戏”创建 Listen Server。
- 第一个窗口进入 Lobby 等待。
- 打开第二个 exe，点击“加入游戏”连接 127.0.0.1。
- 人数达到 2 后，服务器自动切换到战斗关卡。
- 测试敌人追击、攻击、玩家死亡重生、玩家射击、击杀得分与胜利提示。

## 10. 总结

本 Demo 在 UE5 First Person 模板基础上完成了敌人 AI、玩家射击、生命值、死亡复活、得分胜利、UMG 界面和 Listen Server 多人流程。实现过程中采用服务器权威模型处理核心玩法状态，使用 Replication、Server RPC、Client RPC 和 Multicast RPC 分别解决状态同步、本地 UI 和动画表现问题，满足题目对敌人、得分胜利与多人网络对战的要求。