

项目文档v0.1.0

摘要

《DeltaNights》是一款以PvPvE“搜打撤”模式为主的TPS多人战术竞技游戏，以二次元科幻漫画风格作为主要表现形式，面向爱好二次元风格与团队战术策略的战术竞技玩家。与主流“搜打撤”相比，它更注重于画面的表现形式、角色技能的组合与差异化、攻击形式的多元化与对战撤离策略，为玩家提供前所未有的沉浸体验。

关键词：搜打撤，PvPvE，二次元

| 日期 | 修改 | 描述 |
|----------------|-------|-----------------|
| 2025.11.4 | 0.0.1 | 构建文档框架 |
| 2025.11.2 3 | 0.1.0 | 总结需求文档等文档内容到此文档 |
| | | |

| git提交名 | github账号 | 学号 |
|------------|-------------------|---------------|
| SL_Skuy | SLSkuy | 2023051604044 |
| 星焰 | StarFlameDisorder | 2023051604032 |
| | | |
| kish666 | kishizi | 2023051604046 |
| Warfarinrl | warfarinrl | 2023051604050 |

1. 立项

1.1 项目起源与提案

1.1.1 项目背景

近年来FPS类游戏市场竞争激烈，以“搜打撤”（Search - Fight - Evacuate）模式为主要玩法的战术类游戏成为市场主流。但该赛道仍存在明显空白--缺乏将二次元风格与战术竞技机制深度融合的产品。本项目旨在补缺市场漏洞，构建一个融合团队战术、角色技能与策略机制的竞技体验。

1.1.2 商机与竞争机会

- 市场缺口：搜打撤模式与二次元的交叉生态位存在空白
- 玩家群体：喜好二次元与竞技游戏的玩家数量高，潜在用户基数大
- 内容创新：传统TPS游戏缺乏角色差异化、攻击形式单一
- 近似游戏：《卡拉彼丘》运营状况较差产能低下，《APEX》反作弊系统差，游玩体验糟糕，《命运扳机》潜在玩家较多，《三角洲行动》的运营决策失误导致玩家流失，《守望先锋》因角色与技能数过多且高度依赖团队配合导致上手门槛高。《无畏契约》因游戏模式过于单一且竞技性强导致玩家无法长期留存。

1.1.3 项目定位

《DeltaNights》是一款以PvPvE“搜打撤”模式为主的TPS多人战术竞技游戏，以二次元科幻漫画风格作为主要表现形式，面向爱好二次元风格与团队战术策略的战术竞技玩家。与主流“搜打撤”相比，它更注重于画面的表现形式、角色技能的组合与差异化、攻击形式的多元化与对战撤离策略，为玩家提供前所未有的沉浸体验。

1.1.4 核心循环

核心玩法为PvPvE“搜 - 打 - 撤”：

- 搜（Search）：玩家在地图中搜索战利品与资源；
- 打（Fight）：与敌方玩家及AI敌人进行战斗；
- 撤（Evacuate）：在安全区撤离并带走搜集成果。

1.1.5 系统组成

- 客户端：角色控制、UI交互、语音通信、技能释放、动画展示；
- 服务端：匹配、战斗同步、数据统计、经济系统；
- 第三方服务：反作弊服务、云平台、语音与存储服务。

1.2 Business Case

1.2.1 市场机会与规模

- 目标用户群体明确：二次元+战术竞技双栖玩家，具备高付费意愿与社区活跃度。
- 市场缺口显著：目前尚无成功融合“二次元美术 + 搜打撤玩法 + 角色技能系统”的产品。
- 竞争环境分析：

- 《APEX》反作弊差，《卡拉彼丘》运营乏力，《三角洲行动》决策失误——均未完全满足目标用户需求。
- 《DeltaNights》通过美术风格、角色技能、攻击形式、战术标记系统等维度实现差异化。

1.2.2 收益预测

- 主要收入来源：
 - 角色皮肤、武器外观、表情动作等虚拟商品销售
 - 战斗通行证
 - 高级礼包

1.2.3 风险评估与缓解策略

| 风险 | 影响 | 缓解策略 |
|--------|----|----------------------|
| 市场竞争加剧 | 中 | 快速迭代、强化社区运营、持续内容更新 |
| 技术实现难度 | 高 | 采用Unity引擎、分阶段开发、原型验证 |
| 玩家接受度低 | 中 | 早期测试反馈、差异化宣传 |

1.2.4 关键成功因素

- 清晰的愿景与定位：明确“二次元战术竞技”细分市场
- 持续的内容更新：保持玩家活跃与社区热度
- 强反作弊与公平性：保障竞技环境健康
- 社区驱动开发：通过玩家反馈优化体验

2. 愿景

2.1 问题陈述

- 产品类型的同质化：当前市场的竞争十分激烈，开发厂商急于从市场竞争中蹭热度赚一笔，多数倾向于对成功案例的粗劣换皮模仿，产品缺乏创造性、创新性、有趣性，使得市面上的产品同质化严重
- 产品的互动性较低：互动形式局限于固定表情/快捷语音，缺乏深度的合作机制、物品获得受算法机制影响，使得用户体验局限于算法框架之下，自由度较低，可互动性较弱，大幅降低用户从随机事件中获取到的刺激感
- 策略性：缺乏有效的队内沟通工具和战术标记系统，团队配合成本高、游戏内教学引导不足，核心机制未被清晰传达，导致学习曲线陡峭

- PVP和PVE：PVP模式受玩家对于产品的熟练度影响，游戏对局的匹配机制可能出现高熟练玩家与低熟练玩家竞争，迫使双方从中获取的正反馈降低；PVE模式受敌人的AI算法影响，过高的或过低的敌人AI算法都会大幅降低用户从中获取到的正反馈
- 玩家流失：若产品内容在长期内都得不到更新迭代，玩家可能会因重复体验相同的内容而感到乏味；适度提高竞技强度可以提高玩家的兴奋感，但过高的竞技强度会导致休闲玩家从中获取到的正反馈降低，进而导致玩家流失

| 要素 | 描述 |
|------|--|
| 问题 | 市场竞争激烈，创新不足、缺乏深度的合作机制、物品获得过于随机、团队配合成本高、游戏内教学引导不足、匹配机制、AI行为模式固定、刷新点背板 |
| 影响 | 玩家、开发团队、内容创作者、运营团队 |
| 结果 | 玩家流失、新玩家减少 |
| 解决方案 | 加强创新和调研，开放测试服，将做出的更改提前试验在一部分玩家身上，通过反馈做出决定 |

2.2 涉众与用户

2.2.1 涉众

2.2.1.1 涉众类型与代表

| 类型 | 代表 | 简述 | 关注点 |
|-------|--------------------|------------------------|-----------------|
| 项目发起者 | 投资方、制作人 | 项目资金与方向决策 | 确定项目能够按期开发上线 |
| 开发团队 | 策划、程序、美术、测试 | 实际开发游戏 | 项目成功、技术实现、创意落地 |
| 用户 | 玩家、主播、电竞选手 | 直接参与游戏的游玩的玩家 | 社交与策略 |
| 运维 | 维护团队、后续开发团队 | 保证游戏正常运转、服务器维护、漏洞修复和更新 | 稳定性、更新速度质量 |
| 服务商 | 服务器提供商、游戏平台、媒体、反作弊 | 提供游戏的运行基础、宣发、提供 | 市场份额、资金回报 |
| 运营 | 宣发、客服 | 添加游戏活动、玩家账号维护、媒体官号维护 | 用户留存、服务器稳定、活动效果 |
| 支持性涉众 | 自媒体创作者、社区、媒体 | 维护游戏生态、宣传游戏 | 用户增加和留存 |

2.2.1.2 涉众优先级

| 涉众 | 权力 | 兴趣 | 优先级 | 沟通策略 |
|-------|----|----|-----|-------------|
| 投资方 | 高 | 高 | 高 | 定期汇报与决策会议 |
| 玩家 | 中 | 高 | 高 | 社区反馈与测试活动 |
| 开发团队 | 高 | 高 | 高 | 敏捷迭代同步 |
| 合作者 | 中 | 中 | 中 | 定期沟通与协议签订 |
| 运维 | 高 | 中 | 中 | 技术同步与维护报告 |
| 支持性涉众 | 低 | 中 | 低 | 提供创作素材与活动参与 |

2.2.2 用户

| 用户类型 | 特征 | 核心需求 |
|-------|----------------------|--|
| 玩家 | 关注社交游戏体验、追求更高的操作技术 | 易上手、团队协作、情绪价值、操作性与深度策略 |
| 运营人员 | 关注游戏运营情况，对运营策略做出调整 | 让公司尽可能获利的情况下满足用户需求 |
| 运维工程师 | 修复游戏出现bug，定期维护服务器 | bug及时反馈及时修复，避免用户出现不愉快体验 |
| 游戏策划 | 思考游戏未来方向，为玩家游戏体验做出优化 | 能够提供合理的游戏未来开发方案给开发团队，使游戏体验得到优化，使游戏内容更加丰富 |
| 系统管理员 | 获取系统日志，局内反作弊监管 | 保证游戏日志稳定输出，对可疑战局采取人为监管，利用系统服务运营与玩家 |

2.3 关键涉众和用户的需要

- 核心玩家 - “二次元战术家”
 - 特征：年龄18-28岁，同时是二次元文化爱好者与硬核竞技游戏玩家。追求角色个性、视觉风格与深度策略的结合。
 - 核心诉求：强烈的角色认同感、丰富的战术组合空间、高回报的团队协作体验、公平且有深度的竞技环境。
 - 游戏动机：钻研BD（角色构建）、与固定队达成战术默契、在竞技中获得优越感、收集个性化外观。

- 潜力玩家 - “风格化探索者”
 - 特征：来自传统竞技游戏或二次元RPG领域，被独特的二次元科幻美术风格所吸引，但对“搜打撤”模式可能较为陌生。
 - 核心诉求：友好的上手体验、直观的视觉反馈、丰富的角色与攻击形式、可持续的成长目标。
 - 游戏动机：体验新颖玩法、享受视觉盛宴、完成角色与装备收集、与朋友进行轻度团队合作。

2.4 产品概述

2.4.1 产品定位陈述

| 项目 | 内容 |
|----------------------|---|
| for(目标) | 二次元文化与战术竞技游戏爱好者，年龄18-28岁，追求角色个性与团队策略 |
| who(需求) | 对现有游戏的美术风格、角色差异化与策略深度不满，渴望更具个性与协作性的游戏体验 |
| the(产品名称) | 《DeltaNights》是一款二次元风格PVPVE“搜打撤”战术竞技游戏 |
| that(核心价值) | 提供融合角色技能、多元攻击与动态撤离策略的团队PVPVE体验 |
| unlike(竞品) | 《APEX》《卡拉彼丘》《三角洲行动》 |
| Our product (差异化) | 更强调角色技能组合、画面表现力、战术自由度与持续内容更新 |

2.4.2 完整的产品概述

能力概述

《DeltaNights》是一款二次元风格PVPVE“搜打撤”战术竞技游戏。核心能力包括：

- 角色战术系统：操作技能独特的干员，实现团队互补与环境互动。
- 多元战斗探索：融合枪械、技能等多元素，进行资源搜索与动态PvE挑战。
- 动态撤离机制：通过守护信标或争夺公共点完成策略性撤离。
- 成长与收集：提供局外技能解锁与丰富的外观收藏。

客户效益

- 沉浸体验：独特二次元美术风格带来强烈视觉享受与角色认同。
- 策略成就感：丰富的战术组合与高价值目标带来高强度正反馈。
- 友好体验：智能标记系统降低沟通门槛，动态事件确保对局新鲜感。

竞争定位

本产品核心竞争力在于“二次元画风 + 角色技能驱动 + PVPVE搜打撒”的完整融合，以玩家体验为核心，创造了独特的细分市场定位。

2.5 产品特性

| 编号 | 特性名称 | 描述 | 优先级 |
|-----|----------|------------------------------------|-----|
| F01 | 核心射击系统 | 第一/第三人称自由切换，包含枪械、手雷等武器系统 | 必须 |
| F02 | 搜打撤机制 | 在地图中搜索战利品、击败敌人、激活撤离点 | 必须 |
| F03 | 角色技能系统 | 各角色拥有被动、普通与终极技能 | 必须 |
| F04 | 多人匹配与排位 | 实时联网匹配战斗 | 必须 |
| F05 | 经济与仓库系统 | 玩家撤离携带的战利品进入个人仓库，可用于交易或制作 | 必须 |
| F06 | 反作弊与公平竞技 | 集成反作弊系统，对开挂账号的严厉惩罚，保障竞技公平性 | 必须 |
| F07 | 战术标记系统 | 支持快速标记敌人位置、战利品、战术指令降低语音沟通门槛 | 应该 |
| F08 | 新手引导与训练场 | 指引玩家掌握“搜打撤”核心循环，提供可测试所有枪械和角色技能的训练场 | 应该 |
| F09 | 对局回放系统 | 查询对局细节，死亡回防等功能，以供玩家复盘战术漏洞 | 应该 |
| F10 | 角色/武器皮肤 | 提供不影响平衡性的个性化外观，是重要的商业化手段 | 应该 |

2.6 其他产品需求

- 性能需求

- 响应时间：95% 的页面加载时间应在 2 秒以内
- 吞吐量：系统应能支持每秒 1000 个并发用户登录
- 容量：数据库应能存储至少 5 年的用户使用记录

- 安全性需求

- 身份认证：支持双因素认证（2FA）
 - 数据加密：所有用户密码必须以加盐哈希方式存储；传输数据必须使用 TLS 1.2+ 加密
 - 权限控制：实现基于角色的访问控制（RBAC），区分管理员、运营、普通用户权限
 - 合规性：个人数据处理必须符合 GDPR/《个人信息保护法》要求，提供用户数据导出和删除功能
- 兼容性需求

1. 操作系统/设备：支持 pc、手柄和移动端；需适配平板电脑的横屏与竖屏模式
- 可靠性、可用性和可拓展性需求
 1. 可靠性：系统月度正常运行时间不低于 99.9%
 2. 可用性：任何计划内的维护需提前 24 小时通知用户，且每月计划内停机时间不得超过 4 小时
 3. 可扩展性：架构设计应支持在 2 小时内通过增加服务器节点，将系统处理能力提升 50%
- 运营与监控需求
 1. 日志：系统需记录所有关键用户操作和安全事件，并集中存储至少 90 天
 2. 监控：需配置关键业务指标（如 API 响应时间）的监控仪表盘和报警机制
 3. 后台：运营人员需可通过管理后台对用户内容进行审核和封禁操作
- 国际化与本地化需求
 1. 国际化：代码架构需支持多语言，字符串需外部化处理
 2. 本地化：首发支持简体中文和英文；UI 布局需适应不同语言的长度（如德语通常更长）
- 法律与合规需求
 1. 隐私政策必须明确告知用户数据收集和使用方式，并获取用户同意
 2. 在欧盟地区提供服务，必须遵守 GDPR 规定

3. 用况建模

3.1 参与者确定

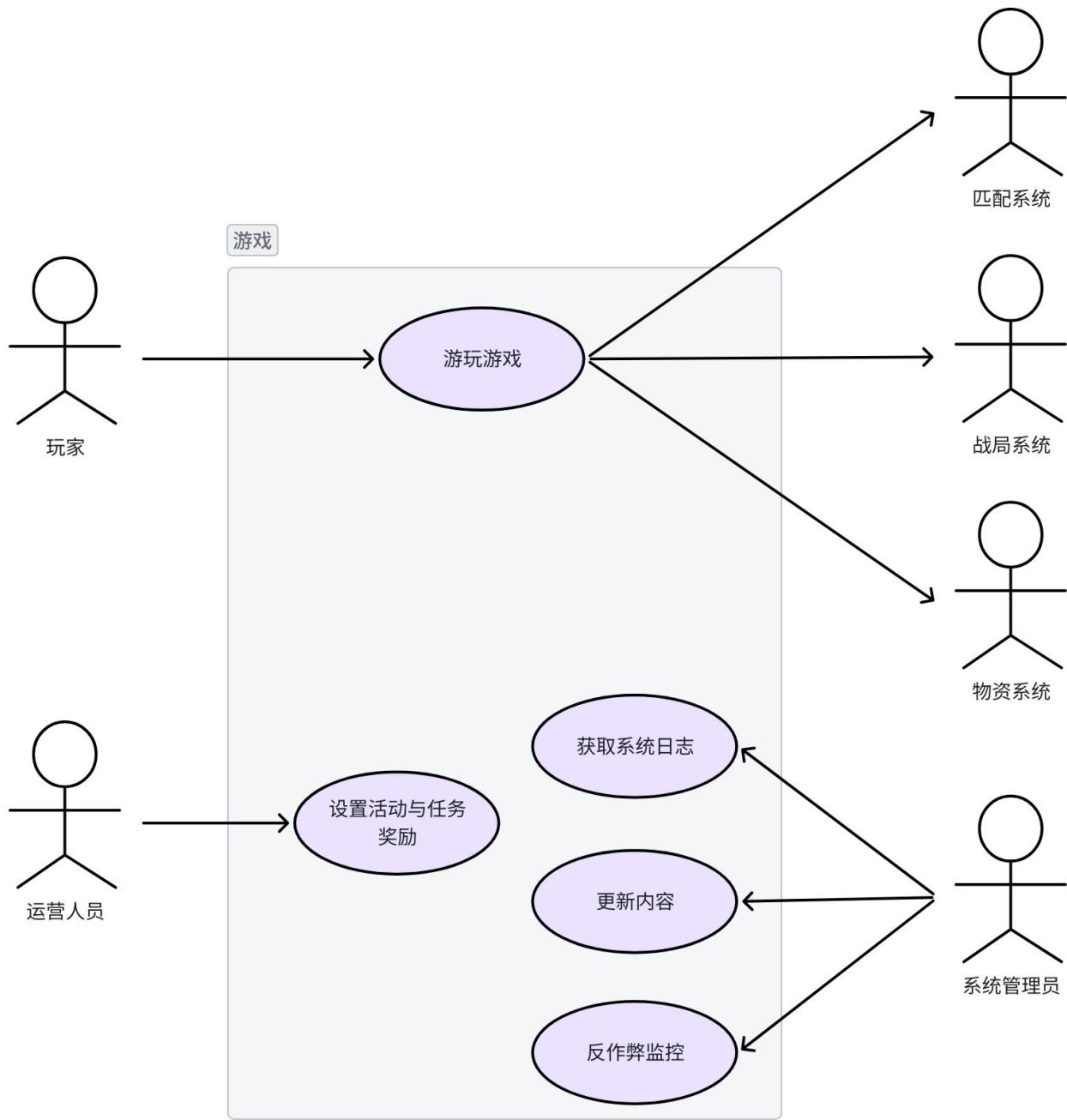
- 玩家 (Player)：游戏的直接用户，执行战术、拾取资源、释放技能、撤离战区
- 匹配系统 (Matchmaker)：执行战局匹配与房间分配逻辑
- 战局服务器 (Battle Server)：处理实时战斗逻辑、AI 逻辑、状态同步、战局结算等功能
- 运营人员 (Operator)：设计游戏活动，配置任务奖励，关注玩家游玩体验
- 系统管理员 (Administrator)：监控与维护服务器运行状态，保护数据安全
- 战局系统 (War Situation system)：分配玩家起始位置，分配地图资源分布
- 物资系统 (Material system)：管理物资信息

3.2 术语表

| 名称 | 定义 |
|----|----|
| 玩家 | |

| | |
|-------|--|
| | 游戏的直接用户，能够执行登录、匹配、战斗、交易等所有游戏操作。在战局中，玩家操控一个特定的游戏角色进行活动 |
| 战局 | 也称“对战局”。指一名或一组玩家从匹配成功、进入特定游戏地图开始，到最终成功撤离、全体阵亡或对局时间结束为止的完整、独立的一局游戏。 |
| 匹配系统 | 负责组织和管理战局中队伍的后台服务系统，将匹配成功的玩家组分配到指定的战局服务器中 |
| 战局服务器 | 运行战局的服务器，负责托管并执行特定战局内的所有实时游戏逻辑 |
| 战局系统 | 对战局内的资源，任务和玩家进行分配 |
| 物资系统 | 负责管理玩家在局内，局外获取到的物资 |
| 撤离 | 玩家在战局内抵达撤离点，在倒计时结束后便完成撤离 |

3.3 用况图



3.4 用况详细描述

游玩游戏

参与者：玩家

前置条件：

- 玩家网络和服务器网络可用

基本流：

1. 执行子流S1 登录
2. 执行子流S2 组队
3. 执行子流S3 匹配

4. 执行子流S4 生成战局
5. 执行子流S5 战局内交互
6. 执行子流S6 结算对局
7. 重复步骤2-6，直到玩家退出
8. 结束

子流：

子流S1 登录

参与者： 用户，登录系统

前置条件：

- 用户已启动游戏应用，并进入登录界面。

基本流：

1. 用户在登录界面选择登录方式
2. 系统根据用户选择提供登录验证
3. 用户提供登录验证凭证
4. 系统验证凭证通过后，加载用户个人资料和游戏数据，并跳转至游戏首页。
5. 系统显示“登录成功”提示信息
6. 系统修改用户状态
7. 用况终止。

备选流：

A1 “处理无效凭据”（特定备选流）

扩展点及触发条件：在基本流第3步，当系统验证发现用户名或密码错误时。

事件流：

1. 系统提示“用户名或密码错误，请重试”。
2. 系统清空密码输入框，保留用户名输入框内容。
3. 事件流恢复：事件流恢复到基本流第1步，用户可重新输入凭据或终止登录。

A2 “处理网络或服务器错误”（特定备选流）

扩展点及触发条件：在基本流第3步，当系统因网络连接失败或服务器无响应导致登录请求失败时。

事件流：

1. 系统提示“网络连接失败，请检查网络后重试”。
2. 事件流恢复：事件流恢复到基本流第1步，用户可重新尝试登录或终止。

子流S2 组队

参与者：用户，匹配系统，社交系统

前置条件：

- 用户登录成功

基本流：

1. 用户在首页或活动界面点击“组队”功能之后，用况开始启动。
2. 系统加载组队界面，展示可加入队伍列表、创建队伍入口等信息。
3. 用户选择执行以下任一操作：
 - 创建新队伍
 - 加入现有队伍
 - 向现有队伍发送加入申请

{执行组队操作}

4. 系统根据用户所选操作进行处理：
 - 创建队伍：系统要求填写队伍名称、简介、人数上限等并创建队伍。
 - 加入队伍：系统将用户加入该队伍（若该队伍无需审核）。
 - 申请加入：系统向队长/管理员发送入队申请。
5. 系统更新用户的队伍状态，并显示加入成功、创建成功或申请已发送等反馈信息。
6. 用况终止。

备选流：

A1 “处理队伍已满”（特定备选流）

扩展点及触发条件：在 {执行组队操作} 扩展点上，当用户尝试加入的队伍人数已满。

事件流：

1. 系统提示“队伍人数已满，无法加入”。
2. 系统返回队伍列表页面。

事件流终止：组队用况终止。

A2 “处理组队操作失败”（特定备选流）

扩展点及触发条件：

在 {执行组队操作} 扩展点上，当系统执行加入、申请、创建队伍时由于网络错误、权限问题等导致失败。

事件流：

1. 系统提示“操作失败，请重试”。

事件流恢复：事件流恢复到基本流第3步，用户可重新选择组队方式或终止。

子流S3 匹配

参与者： 用户， 匹配系统

前置条件：

- 用户登录成功，并处于游戏大厅或匹配入口界面。
- 用户已满足匹配条件（例如，已完成组队或选择单人匹配）。

基本流：

1. 用户在大厅界面点击“开始匹配”按钮后，用况开始启动。
2. 系统加载匹配界面，显示匹配队列状态（如预计等待时间、当前队列人数）。
3. 系统根据用户匹配请求（如模式选择、队伍状态）加入匹配队列。
4. 系统在队列中寻找符合条件的其他玩家，并进行匹配计算。

{执行匹配操作}

5. 系统成功匹配到足够玩家后，创建游戏房间，并分配队伍。
6. 系统跳转至游戏加载界面，显示匹配成功信息（如队友列表、地图信息）。
7. 用况终止。

备选流：

A1 “处理匹配取消”（特定备选流）

扩展点及触发条件： 在{执行匹配操作}扩展点上，当用户主动取消匹配（如点击“取消匹配”按钮）时。

事件流：

1. 系统立即将用户移出匹配队列。
2. 系统提示“匹配已取消”。
3. 事件流终止：匹配用况终止，返回大厅界面。

A2 “处理匹配超时”（特定备选流）

扩展点及触发条件： 在{执行匹配操作}扩展点上，当匹配等待时间超过预设阈值（如5分钟）仍未成功时。

事件流：

1. 系统提示“匹配超时，请重新尝试”。
2. 系统自动将用户移出匹配队列。
3. 事件流恢复：事件流恢复到基本流第1步，用户可重新发起匹配或终止。

A3 “处理匹配失败”（特定备选流）

扩展点及触发条件：在{执行匹配操作}扩展点上，当系统因错误（如服务器故障、数据不一致）导致匹配失败时。

事件流：

1. 系统提示“匹配失败，请重试”。
 2. 系统自动将用户移出匹配队列。
 3. 事件流恢复：事件流恢复到基本流第1步，用户可重新发起匹配或终止。
-

子流S4 生成战局

参与者：战局系统

前置条件：

- 战局匹配成功
- 服务器可用

基本流：

1. 系统分配游戏服务器资源
2. 系统加载游戏规则
3. 执行 S1: 初始化游戏地图
4. 玩家选择干员
5. 分配玩家位置
6. 系统初始化玩家干员
7. 系统确认所有玩家准备状态
8. 游戏对局正式开始
9. 结束

备选流：

A1 "生成地图错误"（特定备选流）

扩展点及触发条件：在{初始化游戏地图}扩展点上，当系统初始化游戏地图发生错误时

事件流：

1. 系统操作：延长玩家在加载界面的时间
2. 系统重新初始化地图
3. 事件流恢复：玩家进入战局或玩家回到匹配界面

A2 "玩家中途退出" (特定备选流)

扩展点及触发条件：在{玩家选择干员-结束}扩展点上，当玩家意外退出对局时

事件流：

1. 系统保留玩家当前信息
 2. 系统删除对应玩家操作角色及位置
 3. 在玩家重连时恢复保留信息，继续流程
-

子流S5 战局内交互

参与者：玩家、战局系统，系统管理员

前置条件：

- 对局已生成完备
- 玩家正常联机

基本流：

1. 系统接收玩家操作输入（移动、攻击、使用技能等）
2. 系统验证操作合法性
3. 系统处理操作结果(执行S1或S2或S3)
4. 系统更新和同步游戏状态
5. 重复步骤1-4，直到游戏结束条件满足
6. 结束

子流：

- S1 处理战斗操作:

计算状态效果

计算伤害

更新状态和血量

- S2 处理道具使用:

验证道具

应用效果

- S3 处理团队协作:

验证团队沟通信息

转发团队沟通信息

备选流

A1 "操作验证失败"

扩展点及触发条件：在{系统验证玩家操作合法性}扩展点上，当玩家操作输入不合法时

事件流：

1. 系统处理：拒绝操作，提示拒绝原因
2. 事件流恢复：返回等待新操作

A2 "反作弊检测触发"

扩展点及触发条件：在{系统验证玩家操作合法性}扩展点上，当玩家操作与实际输入不符合

事件流：

1. 系统处理：返回当前玩家信息
2. 系统将信息提交至系统管理员
3. 系统管理员对该玩家进行检测
4. 系统记录日志
5. 系统对该玩家做出相应操作

A3 "玩家阵亡"

扩展点及触发条件：在{系统处理操作结果(执行S1或S2或S3)}扩展点上，当玩家血量归零时

事件流：

1. 停止对玩家干员的操作
2. 返回玩家被淘汰面板
3. 等待玩家操作（观战，离开等）
4. 事件流恢复：继续战局流程

子流S6 结算对局

参与者：战局系统、物资系统

前置条件：

- 对局已结束
- 玩家撤离
- 玩家阵亡

基本流：

1. 系统收集战局最终数据

2. 执行 S1: 计算基础奖励
3. 执行 S2: 计算加成
4. 更新玩家统计
5. 系统调用物资系统发放奖励
6. 执行 S3: 验证发放
7. 系统生成战局报告
8. 结束

子流：

- S1 计算基础奖励:

根据胜负结果计算基础奖励

- S2 计算加成:

根据玩家表现和活动计算额外奖励

- S3 验证奖励发放

向物资系统确认物资发放状况

备选流：

A1 "奖励发放失败"

扩展点及触发条件：当{执行 S3:验证发放}扩展点上，验证到发放错误或未发放到玩家仓库或邮件中时

事件流：

1. 物资系统重新对其发放
2. 再次验证是否成功
3. 记录错误日志
4. 事件流恢复：继续正常流程

获取系统日志

参与者：系统管理员

前置条件：

- 明确获取需求
- 具备访问权限

基本流：

1. 接收并确认请求：系统管理员接收到获取日志的请求，并明确需求细节。

{查看有无权限访问}

2. 定位日志文件：根据系统类型和应用程序配置，找到目标日志文件的存储路径。（满足条件进入子流S1）

{查看文件是否存在}

3. 确定时间范围：根据时间戳过滤出指定时间范围内的日志条目。
4. 检索与打包日志：将过滤后的日志输出重定向到一个临时文件。
5. 传输至本地：将打包好的日志文件从目标系统传输到本地分析环境或指定的安全存储区。
6. 执行初步检查：对下载的日志文件进行快速检查，例如验证文件完整性（未损坏）、确认时间范围是否正确、是否包含预期的内容（如相关的错误信息）。
7. 交付日志并记录：将日志文件交付给请求方（如开发工程师）。在工单或工作日志中记录此次操作。
8. 结束。

备选流：

A1 “权限不足”（特定备选流）

扩展点及触发条件：在{查看有无权限访问}扩展点上，发现系统管理员没有访问日志权限。

事件流：

1. 尝试使用 `sudo` 提权（如果该管理员拥有sudo权限）。
2. 如果权限依然不足，联系安全团队或系统所有者申请临时提升的权限。
3. 如果情况紧急且符合安全策略，可能有预先设定的“应急响应”流程可以触发，以获得临时访问权。

事件流恢复：事件流恢复到基本流第1步，重新检查是否有权限

A2 “文件不存在”（特定备选流）

扩展点及触发条件：在{查看文件是否存在}扩展点上，发现日志文件不存在。

事件流：

1. 检查日志文件路径是否正确，文件是否存在
2. 联系该系统的负责开发或运维人员，确认日志的正确位置。

事件流恢复：事件流恢复到基本流第2步，重新检查文件是否存在

子流S1：从集中式日志平台获取

参与者：系统管理员

前置条件：

- 使用了ELK等集中式日志系统

基本流：

1. 登录到日志平台的Web管理界面。
 2. 使用查询语言（KQL, SPL, LogQL）构建查询语句，指定主机、应用和时间范围。
 3. 执行查询并验证结果。
 4. 将查询结果导出为JSON、CSV或文本文件。
 5. 用况中止。
-

更新内容

参与者：系统管理员

前置条件：

- 更新包已经准备就绪

基本流：

1. 执行预更新检查：检查系统状态，内存与CPU资源是否充足
2. 进入维护模式：在游戏内进行全局操作，通知玩家即将进行更新
3. 停止游戏服务：按顺序停止游戏服务器，后台服务等
4. 部署更新内容：将更新的资源包部署到目标服务器上
5. 预览与自我检查：对更新完毕的游戏进行预览和自检

{系统运行}

6. 恢复游戏服务：按顺序恢复游戏服务器，后台服务等
7. 更新完成通知：通知所有玩家群体本次更新已结束
8. 更新内容文档化：记录本次更新的详细信息（时间，版本号等）
9. 后续等待小规模bug反馈
10. 结束

子流：

- S1执行灰度更新：将一小撮玩家部署到新版本验证更新内容的可行性并且在此阶段不发布更新公告，验证完更新可行性后将更新内容应用到所有玩家上

备选流：

A1：“执行回滚流程”

扩展点及触发条件：在基本流中任何情况下均可触发

事件流：

1. 停止有问题的更新版本服务，从备份中恢复被修改的配置文件或数据（或执行数据库回滚脚本）
2. 重新部署和启动上一个稳定版本的服务。

事件流恢复：对更新资源包进行检查修改，事件流恢复到基本流第4步

A2：“自检发现异常”

扩展点及触发条件：在{系统运行}扩展点发现游戏更新内容有错误，如bug，实际内容和预期不符

事件流：

1. 执行备选流A1

事件流恢复：同备选流A1恢复

“A3：紧急热修复”

扩展点及触发条件：在{后续等待小规模bug反馈}扩展点获取到玩家反馈

事件流：

1. 准备最小化的修复补丁
2. 执行基本流，跳过部分非必要的步骤

事件流恢复：事件流恢复到基本流第9步

反作弊监控

参与者：系统管理员

前置条件：

- 监控系统已经部署并运行

基本流：

1. 日常监控与巡检
 - 系统管理员查看核心指标的整体健康状况。
 - 关注异常峰值：如特定API错误率飙升、来自特定IP区域的登录激增、游戏对局数据异常等。
2. 告警触发：监控系统基于预设规则自动触发告警。
3. 初步分析与调查
 - 系统管理员查看告警详情，关联相关日志进行深入分析。
 - 操作包括：
 - 查询该玩家的详细行为日志（登录、对局、物品变动）。
 - 分析网络数据包捕获（如有）。
 - 检查该玩家客户端的反作弊心跳报告或扫描结果。

- 关联其他数据，判断是误报还是确凿的作弊行为。

4. 分类与判定

- 根据分析结果，对事件进行分类：
 - 确认作弊：证据确凿。
 - 高度可疑：需要进一步观察或上报安全工程师进行深度分析。
 - 误报：由网络延迟、游戏BUG或其他合法原因引起。

5. 执行标准响应

- 对于【确认作弊】：
 - 根据预定义的策略，在游戏后台管理系统中执行封禁操作。封禁类型可能包括：账号封禁、硬件ID封禁、IP段封禁（谨慎使用）。
 - 记录封禁原因、证据和操作人员。
- 对于【高度可疑】：
 - 将玩家加入观察列表，进行更密集的日志记录。
 - 创建工单并转交给安全工程师（备选流 A-1）进行深入调查。

6. 记录与报告：在安全事件管理系统中完整记录本次事件、分析过程和处置结果。

7. 规则优化：根据本次事件中发现的误报（备选流 A-2）或新型作弊手段（备选流 A-3），提出优化检测规则的请求，并转交给安全或开发团队。

子流：

- S1进行玩家行为分析：需要对一个可疑玩家进行深度调查时，调取该玩家一段时间内的所有对局数据（K/D，爆头率，视角移动方式等），分析其历史数据是否有突然且不合理的表现提升，生成分析报告作为判断证据
- S2处理大规模作弊事件：检测到新型外挂在游戏中广泛传播时，立即在运营团队内部进行警告，安全团队分析新型外挂的特征，制定相应检测与封禁规则，严格封禁作弊玩家，公示封禁名单以维护游戏正义

备选流：

A1：“上报安全工程师”

扩展点及触发条件：在{执行标准响应}扩展点，发现有需要高度怀疑的玩家

事件流：

1. 反作弊团队收集相关信息给安全工程师
2. 安全工程师进行深度调查并给出报告

事件流恢复：回到基本流6，公布执行结果

A2：“处理误报”

扩展点及触发条件：在步骤4中确定为误报

事件流：

1. 解除对玩家的限制，如果导致玩家游戏体验受损，对玩家发出致歉邮件与补偿

事件流恢复：回到基本流7，汇报信息进行反作弊规则的优化

A3：“应对未知作弊手段”

扩展点及触发条件：反作弊监测系统没有检测到，但通过大量玩家举报和游戏社区反馈推测有新型作弊手段

事件流：

1. 通过玩家举报与社区反馈推测判断新型作弊手段
2. 制定临时反作弊规则部署到游戏
3. 进行作弊者封禁

事件流恢复：回到基本流7，汇报反作弊信息进行反作弊规则优化

A4：“反作弊系统本身宕机”

扩展点与触发条件：通过玩家群体与社区反馈发现反作弊系统宕机

事件流：

1. 团队内部将其视为最优先事项，分析系统宕机原因
2. 尝试重启运行反作弊系统
3. 如果有必要暂停游戏服务则进行服务器维护修理反作弊系统

事件流恢复：回到基本流6，发布玩家和社区发布反作弊维护公告

设置活动和任务奖励

参与者：运营人员

前置条件：

- 运营目标已明确：团队已经确定了本次活动的具体目标（如提升用户留存率）
- 资源已就位：美术和文案资源就位，活动与任务的奖励已在数据库准备好

基本流：

1. 登录管理后台：运营专员使用授权账号登录游戏运营管理后台。
2. 进入活动管理模块：在后台中使用“活动管理”或“任务管理”功能模块。
3. 配置基础信息
 - 输入活动名称、活动类型（如登录活动、充值活动、挑战任务等）。

- 设置活动时间：包括开始时间与结束时间（精确到分钟）。
- 设置活动描述和规则说明。
- 设置活动或任务的完成条件。
- 设置对应的奖励信息（S1, S2）。

4. 预览与自我检查：对设置好的任务与活动进行前端展示，确认是否无BUG存在。
5. 发布活动：系统将活动状态变为“已发布”。活动在预设的开始时间自动对玩家生效。
6. 记录活动或任务的日志：记录本次活动任务的详细信息。

子流：

- S1配置多档位奖励：适用于充值返利、累计登录等多档位活动。
- S2配置概率型奖励：适用于开箱、抽奖类活动。

备选流：

A1：“自检时发现错误”

扩展点及触发条件：在步骤4发现有错误存在

事件流：

1. 运营人员自己对发现的错误进行修正

事件流恢复：回到基本流4继续进行自检

A2：“活动上线后发现错误”

扩展点及触发条件：在活动正式上线后，由玩家群体发现有错误存在

事件流：

1. 运营团队将活动和任务紧急关闭
2. 由技术人员对存在的错误进行处理
3. 完成热更新

事件流恢复：回到基本流4进行自检

4. 需求分析

4.1 健壮性分析

4.2 交互建模

5. 架构设计

6. 详细设计

后记

参考文献