

需求文档 v1.0.0

《DeltaNights ---- 二次元搜打撤战术竞技游戏》

需求规格说明书

版本：v1.0.0

日期：2025.11.5

1. 引言

1.1 文档目的

本需求规格说明书旨在明确《DeltaNights》的系统功能、非功能需求、用户角色、交互用况及性能约束，为开发、测试、运维及投资决策提供一致的参考依据。该文档为后续系统设计、测试及维护提供基础支撑。

1.2 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。

1.3 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。

2. 项目概述

2.1 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，它更侧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。

2.2 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。

2.3 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。

3. 涉众与用户分析

3.1 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

3.2 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

4. 用户类别与用户画像

4.1 用户类别

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

4.2 目标用户画像

- 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。

4.3 可玩角色定位

游戏角色不仅是视觉形象的载体，更是差异化玩法与团队策略的核心。

- 定位原则：
 - a. 技能互补性：每个角色拥有独特的战术技能（如侦察、突击、支援、控制），迫使团队进行角色搭配，形成“1+1>2”的协同效应。
 - b. 攻击多元化：突破传统“枪械”限制，角色可配备或天生拥有能量武器、实体刀剑、无人机、陷阱等特殊攻击手段，实现近、中、远程攻击覆盖。
 - c. 策略导向性：角色技能设计服务于“搜（信息）、打（战斗）、撤（转移）”的核心循环，提供如位移、隐身、架设传送门等改变战局的战术选项。

5. 问题陈述与需求金字塔

5.1 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

5.2 需求金字塔

6. 用况建模

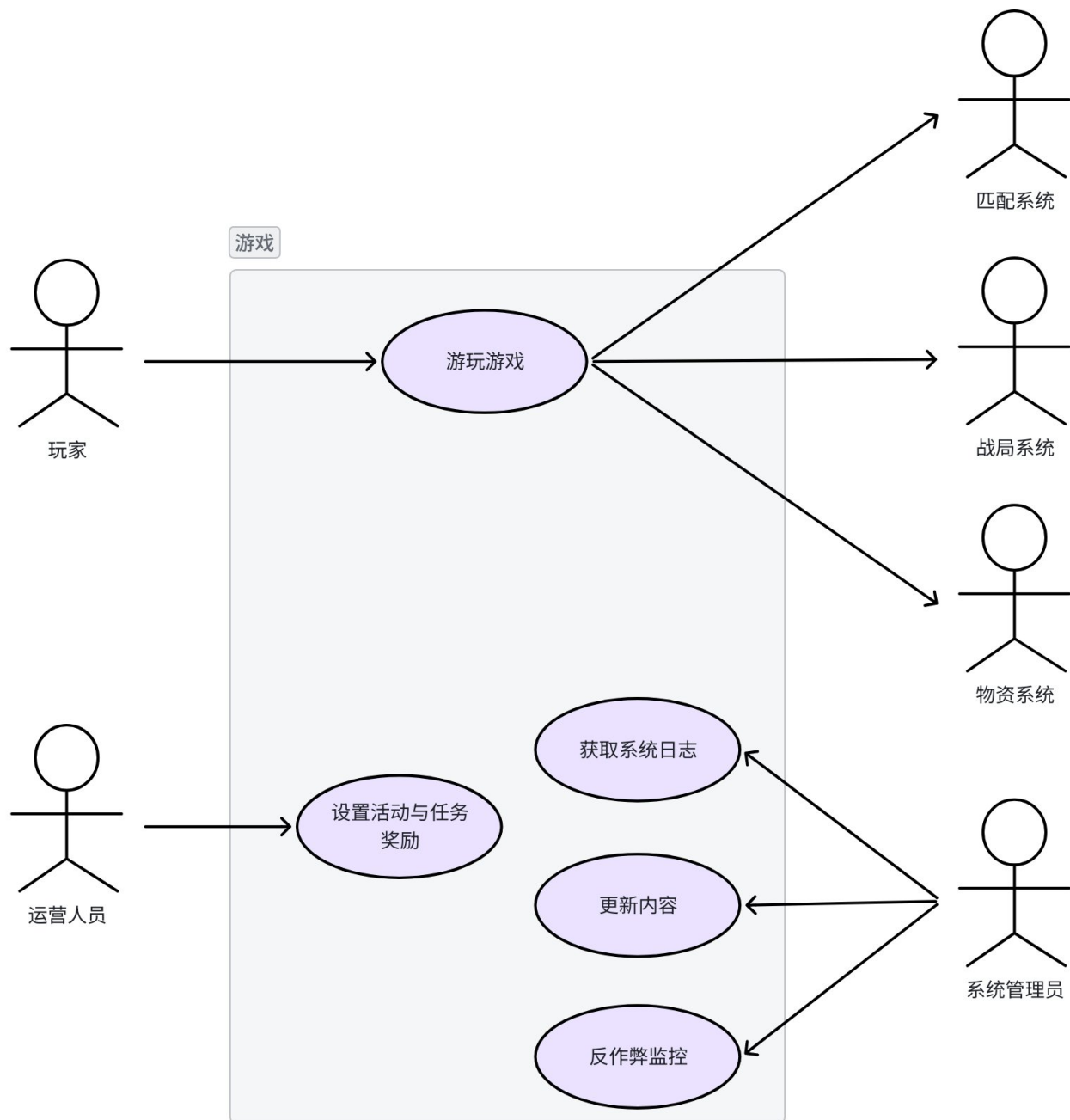
6.1 参与者确定

- 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

6.2 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

6.3 用况图



6.4 用况详细描述

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 匹配
3. 执行子流S3 生成战局
4. 执行子流S4 战局内交互

5. 执行子流S5 结算对局
6. 重复步骤2-5，直到玩家退出
7. 结束

子流：

子流S1 登录

参与者：用户

基本流：

1. 用户在登录界面输入密码和账号
2. 用户使用其他验证方式登录
3. 系统提供验证检测
4. 用户输入验证码
5. 用户成功登录后更改状态
6. 结束

备选流：

A1 账号不存在

- 触发：数据库中没有当前登录信息
- 处理：为当前用户创建新的账号
- 恢复：返回登录界面等待操作

A2 玩家忘记密码

- 触发：用户点击忘记密码提示入口
- 处理：用户重制密码
- 恢复：返回登录界面

A3 网络连接中断

- 触发：用户网络连接断开
- 处理：刷新界面
- 恢复：返回登陆界面

A4 服务器维护中

- 触发：服务器不能提供正常服务
- 处理：运营人员维护服务器
- 恢复：服务器提供正常服务

子流S2 匹配

参与者：匹配系统

前置条件：

- 满足当前匹配需求

基本流：

1. 开始匹配
2. 匹配系统为其提供服务
3. 执行 S1: 执行匹配算法
4. 匹配成功后接受玩家处理
5. 结束

子流：

- S1 执行匹配算法

系统根据玩家MMR（匹配等级）寻找相近对手

考虑地理位置优化网络延迟

备选流：

A1 玩家取消匹配

- 触发：在步骤1-4，玩家在匹配过程中取消匹配
- 处理：修改玩家状态，停止匹配系统服务
- 恢复：返回匹配界面

A2 玩家未接受对局

- 触发：在步骤4，玩家未接受对局
- 处理：修改玩家状态，停止匹配系统服务
- 恢复：返回匹配界面

A3 玩家网络中断

- 触发：在步骤1-4，匹配过程中玩家网络中断
- 处理：刷新
- 恢复：返回当前状况

A4 匹配失败：

- 触发：在步骤3，未找到满足条件玩家

- 处理：扩大匹配范围继续匹配，记录日志
- 恢复：返回匹配界面

A5 连接战局系统失败

- 触发：无法将玩家信息录入战局系统
- 处理：匹配系统记录错误并尝试重新连接或切换到备用战局服务器
- 恢复：返回失败信息，继续连接战局

子流S3 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. 分配玩家位置
5. 系统创建游戏房间并邀请所有玩家
6. 系统确认所有玩家准备状态
7. 游戏对局正式开始
8. 结束

子流：

- S1 初始化游戏地图

加载对应场景

根据人数计算生成概率和生成数量

生成对应物资和NPC

备选流：

A1 生成地图错误

- 触发：在步骤3，地图设置错误
- 处理：中止对局，后台显示报错

A2 玩家中途退出

- 触发：在步骤6，玩家意外退出对局

- 处理：删除对应玩家操作角色及位置
- 恢复：继续流程

子流S4 战局内交互

参与者：玩家、战局系统

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1-5，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

- 触发: 在步骤2，操作不合法（比如缺少对应道具）
- 处理: 系统拒绝操作，提示原因
- 恢复: 返回等待新操作

A2 反作弊检测触发

- 触发: 在步骤2，检测到可疑操作模式
- 处理: 系统记录日志，可能限制玩家操作
- 恢复: 继续正常流程，标记监控

A3 玩家阵亡

- 触发: 在步骤4计算玩家血量归零
- 处理: 进入战局结算，删除当前战局内玩家操作角色
- 恢复: 其他玩家继续正常流程

子流S5 结算对局

参与者：战局系统、物资系统

前置条件：对局已结束或玩家撤离或玩家阵亡

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

- 触发: 在步骤6，检测发放失败
- 处理: 重新发放，并在后台记录发放错误日志
- 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。
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
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。

子流：

- S1从集中式日志平台获取：如果使用了ELK等集中式日志系统，则可通过查询语言构建查询语句，更简单地获取日志内容

备选流：

A1：权限不足

- 触发：在步骤1中发现自身权限不足无法访问系统日志
- 处理：寻求拥有更高权限的管理员进行权限升级
- 恢复：回到基本流步骤1继续执行

A2：文件不存在

- 触发：在步骤2中无法定位到日志文件
- 处理：检查日志文件路径是否正确，文件是否存在
- 恢复：回到基本流步骤2

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
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）

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

- 触发：有需求执行该流程
- 处理：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），重新部署和启动上一个稳定版本的服务。
- 恢复：进行预览与自检，恢复游戏服务，分析失败原因

A2：自检发现异常

- 触发：步骤5中发现有错误
- 处理：执行备选流A1
- 恢复：同备选流A1恢复

A3：紧急热修复

- 触发：在游戏更新上线后发现小范围的BUG
- 处理：准备最小化的修复补丁，执行基本流，跳过部分非必要的步骤
- 恢复：事后补全更新文档，进行分析与复盘

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内广泛传播时，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

- 触发：在步骤5中，有需要高度怀疑的玩家
- 处理：反作弊团队收集相关信息给安全工程师，安全工程师进行深度调查并给出报告
- 恢复：通过安全工程师给出的报告对玩家进行封禁或解除高度怀疑

A2：处理误报

- 触发：在步骤4中确定为误报
- 处理：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
- 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

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

- 触发：通过玩家群体与社区反馈发现反作弊系统宕机
- 处理：团队内部将其视为最优先事项，分析系统宕机原因，尝试重启运行反作弊系统，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
- 恢复：对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

- 触发：在步骤4发现有错误存在
- 处理：运营人员自己对发现的错误进行修正
- 恢复：回到步骤4继续进行自检

A2：活动上线后发现错误

- 触发：在活动正式上线后，由玩家群体发现有错误存在
- 处理：运营团队将活动和任务紧急关闭，由技术人员对存在的错误进行处理，完成热更新
- 恢复：回到步骤4进行自检

7. 系统功能需求

7.1 系统功能架构

代码块

- 1 客户端
- 2 |—— 战斗系统（移动、射击、技能）
- 3 |—— 匹配系统（组队、匹配分段）
- 4 |—— 社交系统（聊天、语音、好友）

5	└─ UI系统（背包、地图、任务、撤离点）
6	└─ 市场系统（交易、仓库）
7	└─ 回放系统（复盘、战术分析）
8	
9	服务端
10	└─ 玩家数据服务
11	└─ 战斗逻辑与同步
12	└─ 匹配与战局管理
13	└─ 经济系统与市场管理
14	└─ 日志监控与反作弊模块
15	└─ 网络通讯模块

3.1 核心战斗循环（搜-打-撤）

功能模块	用户故事（作为【玩家】，我希望…）	核心验收标准
战术地图	…能在进入对局前与队友查看一张包含资源区、任务点、撤离点信息的动态地图，以便规划初始落点与行动路线。	<ul style="list-style-type: none">• 地图清晰展示不同危险等级的区域。• 撤离点位置在对局后期动态刷新。
高价值目标	…能在地图中发现并完成随机出现的PvE首领挑战或物资空投任务，以便获得顶级装备，并引发其他队伍的争夺。	<ul style="list-style-type: none">• 系统会通过视觉与语音提示全图玩家。• 完成挑战奖励显著高于普通搜刮。
动态撤离	…能在准备撤离时，与队友共同启动一个需要时间守护的撤离信标，或在限时内赶往一个随机出现的公共撤离点，以便增加撤离的紧张与不确定性。	<ul style="list-style-type: none">• 启动信标会暴露队伍位置并吸引AI敌人。• 公共撤离点有最大队伍人数限制。

3.2 角色与技能系统

功能模块	用户故事（作为【玩家】，我希望…）	核心验收标准
角色技能	…能操作拥有一个被动天赋、一个战术技能和一个终极技能的角色进行战斗，并且技能能与我携带的武器产生联动（如用电击技能引爆水坑），以便实现多元化的攻击与战术。	<ul style="list-style-type: none">• 每个角色拥有独特的三技能组合。• 技能与部分环境元素、武器特效有明确的交互逻辑。
局外成长	…能在对局外通过使用游戏内货币和材料，为角色解锁新的技能变体或升级被动天赋，以便定制我的玩法风格。	<ul style="list-style-type: none">• 成长系统提供明确的、非数值碾压的玩法分支（如改变技能效果）。

3.3 战术沟通与交互

功能模块	用户故事（作为【玩家】，我希望…）	核心验收标准
智能标记	…能通过一个径向菜单，快速、精准地标记敌人位置、物资、或建议下一步行动（如“前往这里”、“需要治疗”），以便在无需语音的情况下与队友进行高效沟通。	<ul style="list-style-type: none">• 标记在场景中有明确的3D图标显示。• 针对不同可交互物（物资/敌人/门）有对应的标记类型。
战场遗迹	…能在击败其他玩家或搜刮其物资时，能选择性地捡起并带走其一件“核心装备”，并在局外将其解锁为我的永久收藏品，以便增加对战的收益与仪式感。	<ul style="list-style-type: none">• 被带出的装备将进入玩家的永久仓库。

非功能性需求

4.1 性能需求

1. 客户端性能
- 帧率：在主流配置（如GTX 1060或同级别显卡）下，1080p分辨率中画质运行时，战斗复杂场景帧率不低于60 FPS，基准场景帧率不低于90 FPS。
 - 加载时间：从点击“开始游戏”到进入主界面的冷启动时间不超过15秒；从匹配成功到进入战局地图的加载时间不超过10秒。
2. 服务器性能
- 延迟：95%的玩家连接游戏服务器的网络延迟（Ping值）应低于80ms。
 - 同步：玩家关键操作（如射击、受击、技能释放）的服务器同步频率不低于64Tick，确保战斗反馈及时准确。
 - 承载：单组服务器需支持至少100名玩家同时在一张地图内进行游戏，且AI单位数量增加时，服务器性能损耗在可控范围内。

4.2 可用性与用户体验需求

1. 易上手性
- 新玩家在不超过15分钟的引导流程内，能理解“搜-打-撤”的核心循环、基本操作和首次撤离目标。
 - 所有角色技能的描述必须简洁、无歧义，并配有简短的动态预览图。
2. 界面与交互

- 所有关键游戏信息（血量、弹药、技能冷却、任务目标）必须在UI上清晰呈现，玩家在1秒内即可获取所需信息。
- 自定义按键布局功能必须支持，且设置更改能实时生效。

3. 强反馈

- 要能够在玩家在游玩时为玩家带来强烈的视觉反馈或心理反馈