# 某某某软件

# 需求分析与概要设计

## 项目说明

## 项目目标：

双人联机协作：通过Photon实现实时联机，两名玩家需合作解谜、对抗敌人。

多样化玩法：包含光影谜题、陷阱机关、自动追踪敌人等机制。

关卡设计：每关需结合不同机制

## 软硬件环境需求

操作系统：Windows

.net框架：Unity 2022.3.53f1c1，Photon PUN 2

数据库：无需求

网络：Photon服务器实现双人联机

## 使用的关键技术：

Unity引擎：2D物理系统、Tilemap地图构建、动画状态机

Photon网络框架：玩家同步、状态传输、房间管理

动态光影：使用Unity 2D Light实现光影谜题

AI路径追踪：A\*算法实现敌人自动追踪

（技能系统：基于事件驱动的技能触发与同步）

## 需求分析

## 系统用例



图 1 系统用例图

**1. 创建/加入房间**

**参与者：**玩家

**事件流：**

玩家A创建房间并等待；玩家B通过房间号加入。

Photon同步双方状态，加载联机大厅界面。

**2. 协作解谜**

**参与者：**玩家A、玩家B

**事件流：**

玩家A与玩家B

双人同时操作机关开启通道。

**3. 对抗敌人**

**参与者：**玩家、自动追踪敌人

**事件流：**

敌人通过A\*算法追踪最近玩家。

玩家协作引诱敌人至陷阱区域。

## 业务流程



图 2图书续借流程图

说明：

1. 使用UML活动图（使用StarUML或华为的CodeArts Modeling绘制）画出系统的使用和操作过程。
2. 不用画出所有的使用场景，画出一两个关键用例的使用过程即可
3. 在需求阶段，活动图可以画系统做了什么，但不用画系统内部如何做的。

## 概要设计

## 功能模块设计



图 3功能模块图

说明：

1. 对各个功能模块需要完成的事情进行简要介绍。
2. 注意功能模块图和用例图不同，这里偏重于软件实现需要做功能模块，有些模块可能用户看不见，但也要画处理。
3. 按照功能来划分模块，不要按照前后端来划分模块。一个模块可能包括前端也包括后端功能。功能模块和代码结构并对应。

**1. 联机模块（Photon）**

输入：玩家IP/房间号

输出：同步玩家位置、状态、关卡进度

功能：

房间创建/加入

实时位置同步（通过PhotonView组件）

事件同步（如机关触发、敌人死亡）

**2. 双人操作模块**

输入：玩家A/B的键盘输入（WASD + 技能键）

输出：角色移动、技能释放

功能：

移动限制：一格一格移动（网格化地图）

协作判定：双人同时触发机关的逻辑检测

**3. 敌人AI模块**

输入：玩家坐标、地图数据

输出：敌人移动路径

功能：

A\*算法计算最短路径

状态切换：巡逻 → 追踪 → 攻击

## 核心类图

说明：

1. 类图可以使用StarUML或CodeArts Modeling绘制
2. 画出系统中的关键类以及相互之间的关系。这里关键类包括实体类、控制类和边界类三种。实体类描述系统中使用的数据，控制类描述系统的操作和业务过程，边界类描述系统与外界（如用户或数据库）的交互。



图 4核心类图

## 界面设计

说明：

1. 页面流程图：画一个系统的页面及其如何相互跳转的
2. 界面原型：画出系统主要操作界面，手绘示意图即可, 也可以用墨刀在线画更正式的UI原型

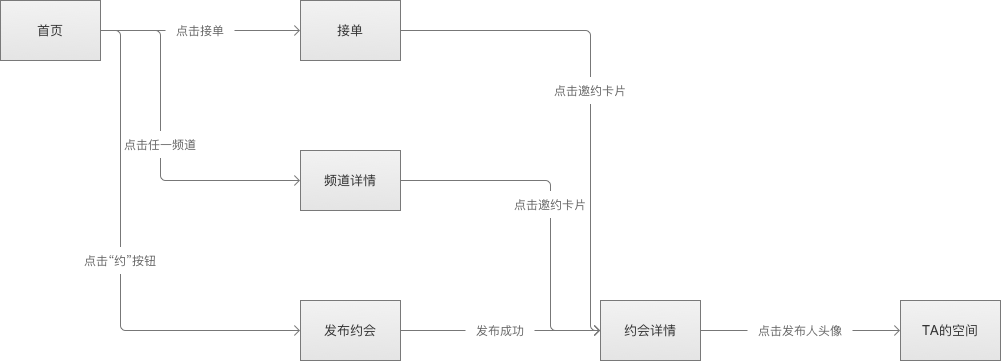


图 5 页面流程图



图 5 系统界面