# 回合资源争夺战

# 需求分析与概要设计

## 项目说明

## 项目目标：

简要介绍项目需要研发出一个什么样的软件产品。

## 软硬件环境需求

介绍软件的运行环境，包括操作系统、数据库、网络等方面等要求。

## 使用的关键技术：

介绍软件使用到的关键技术，包括技术难点和亮点。

## 需求分析

## 系统用例



图 1 系统用例图

1. **查询考试成绩**

**参与者**：学生、教师

**基本事件流**：学生进入个人成绩查询页面，选择学期，点击查看按钮，系统显示本学期该生的所有课程成绩。如果是教师，进入课程成绩查询页面，选择学期和课程名称，点击查询按钮，系统显示该课程对应的考试成绩。

1. **查看选修课程**

……

说明：

（1）用例图使用StarUML或华为的CodeArts Modeling绘制，要画出用户角色和对应用例

（2）用例是描述用户使用软件的场景，不要把用户看不到的后台功能画在用例图上。这里的用户不仅是人，也可以是外部系统。

（3）一个用例可以有不同粒度，但一个用例要是一个完整使用场景，不要把单个步骤作为用例。

（4）用例之间有泛化、包含和扩展关系。泛化适用于表示一个抽象用例的多种不同实现。包含是一个复杂用例的多个细粒度用例。扩展表示一个用例的扩展能力，没有这个扩展用例原用例还是完整的。

（5）对用例图中的各个用例进行详细说明，从使用者的角度说明如何使用系统，不用描述软件系统内部如何实现该功能

## 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

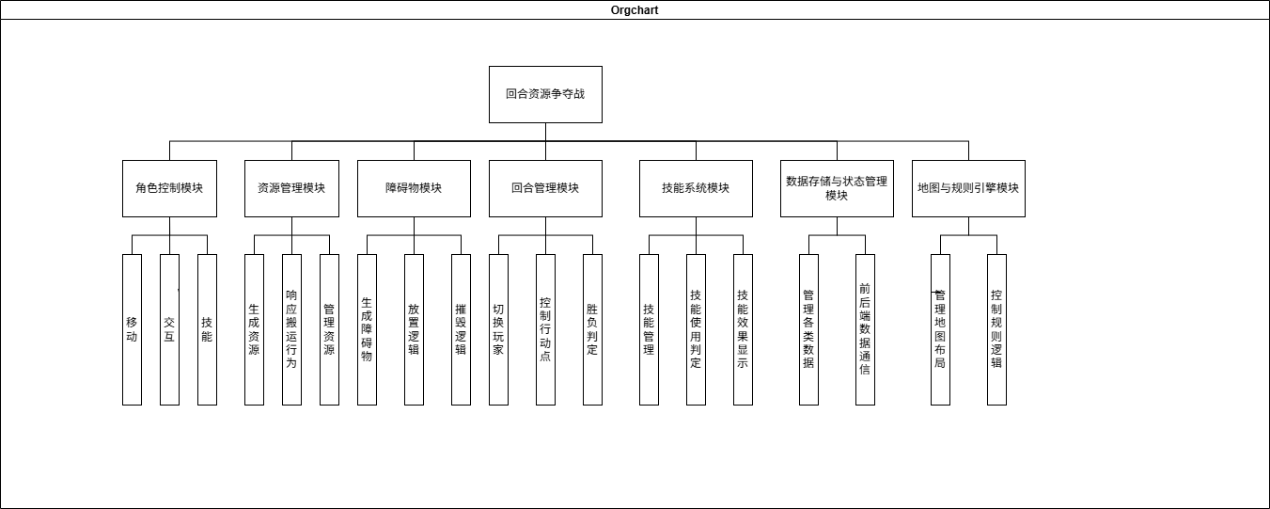


图 3功能模块图

**（1）角色控制模块**

**输入：**键盘产生的控制指令

**输出：**玩家角色的位置、状态（是否携带资源）、技能效果

**功能概述：**实现角色在地图上的移动功能，包括路径判定与动画播放。控制角色搬运资源的交互逻辑（靠近资源点并触发携带）。响应玩家技能释放的输入指令，执行技能效果（如传送、拆除障碍等）。与障碍物和资源的碰撞检测与处理。

**（2）资源管理模块**

**输入：**地图初始化配置、玩家交互行为（如采集资源）。

**输出：**在地图上生成的资源信息（类型、位置、状态）、当前各玩家得分。

**功能概述：**负责在地图指定区域生成资源点并定期刷新资源。管理不同种类资源（如能晶、黄金矿石）及其分值。响应玩家搬运行为，更新资源状态与得分板。控制资源拾取、投放、搬运中断等状态流转。

**（3）障碍物模块**

**输入：**玩家使用技能或手动放置/摧毁障碍物的行为。

**输出：**地图上障碍物状态的更新、动画反馈。

**功能概述：**以指定方式生成三种不同硬度的障碍物。判断角色行为是否满足摧毁条件（如行动点消耗是否足够）。控制障碍物的放置与摧毁逻辑，包括硬度判定和持续动画反馈。提供给角色系统与技能系统接口，配合技能影响障碍。

**（4）回合管理模块**

**输入：**玩家当前操作完成指令、全局游戏时间信息。

**输出：**当前回合玩家身份、剩余行动点、游戏是否结束。

**功能概述：**管理当前轮到哪位玩家进行操作。控制每位玩家行动点的分配与消耗。判断回合结束条件，自动或手动切换至下一玩家。在达到回合上限时进行胜负判定。与UI模块协作显示当前回合状态信息。

**（5）技能系统模块**

**输入：**玩家触发技能键位，资源状态、障碍物状态、行动点数。

**输出：**技能效果动画、资源/障碍物/玩家状态变化。

**功能概述：**管理角色拥有的多种技能，包含冷却时间与使用限制。技能可影响地图元素（如远程放置障碍、瞬移、干扰敌人等）。判断技能是否可以使用（如行动点、冷却等条件）。与障碍物系统、角色控制模块交互，触发具体效果

**（6）数据存储与状态管理模块**

**输入：**所有模块的状态数据与交互行为。

**输出：**游戏运行中的状态快照、持久化存档（如得分、玩家行为日志）。

**功能概述：**管理各类全局变量与玩家状态数据（如分数、剩余资源数等）。作为缓存中枢连接前后端数据，实现状态共享与通信。支持存档、回溯、重开游戏等逻辑。

**（7）地图与规则引擎模块**

**输入：**游戏配置文件、玩家行为数据。

**输出：**当前地图配置、可达区域、规则执行结果。

**功能概述：**管理地图布局、网格系统、障碍与资源初始位置。控制规则逻辑，如不可重复采集、障碍不能重叠等。提供路径可行性查询。供其他模块调用用于行为合法性判断。

## 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

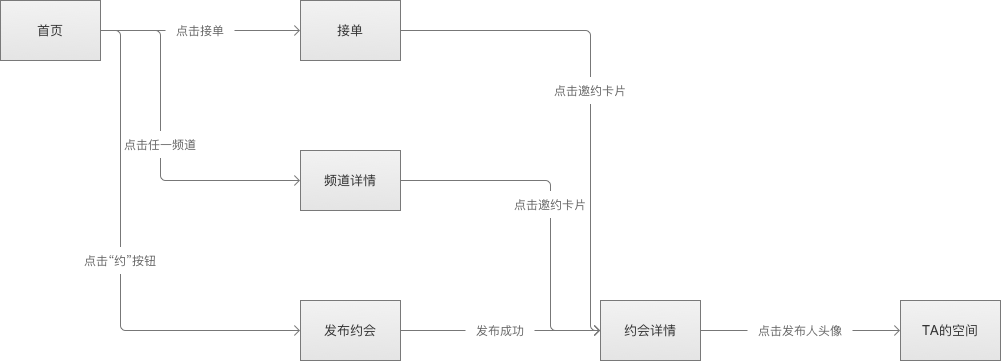


图 5 页面流程图



图 5 系统界面