# Unity2D杂交游戏

# 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。

本项目是一个以2D纵版平台闯关游戏为基础，融合一些常见游戏形式并加以结合的Unity项目游戏。

游戏的主要玩法是控制角色登上更高的平台，不断接受新的玩法，并在过程中获得基本物资从而提升能力。

开发一款2D纵版平台闯关游戏，玩家通过操控角色不断攀登平台，体验融合跳跃、解谜、战斗等多种机制的动态关卡。核心目标：

核心玩法​：垂直平台跳跃为核心，结合阶段性新机制（）。在地图的适当位置有NPC交互，会进入战斗/解密/挑战系统，交互后方可进入下一阶段。

​成长系统​：通过收集物资（金币、能量核心等）提升角色能力（攻击力等）。

​多样化体验​：每层平台引入不同玩法（如竞速、潜行、BOSS战），避免重复疲劳。

## 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。

目标平台​：

PC端（Windows/macOS，支持Steam集成） ~~+ 移动端（iOS/Android，触屏适配）~~。

​开发环境​：

​引擎: Unity 2021 LTS及以上版本。

​编程语言: C#，支持Unity ECS/DOTS优化性能。

​美术工具: Aseprite（像素动画）、~~Blender（3D素材转2D）~~。

​~~性能要求​：~~

~~PC端：最低GTX 1060，60FPS稳定运行；移动端：适配中端机型（如骁龙7系）。~~

## 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。

核心机制实现​：

​物理系统: Unity Rigidbody 2D + 自定义跳跃曲线控制。

​动画系统: 状态机控制角色动作（跳跃、攻击、受伤）。

​关卡生成: 分阶段预生成关卡 + 动态难度调整算法。

​技术难点​：

多机制无缝切换（如从跳跃转向潜行时的输入适配）。

跨平台操作优化（PC键鼠 vs. 移动触屏）。

动态镜头跟随（自动缩放适应平台高度）。

融合Roguelike元素（随机物资掉落，死亡后部分能力继承）。

## 需求分析

## 游戏用例

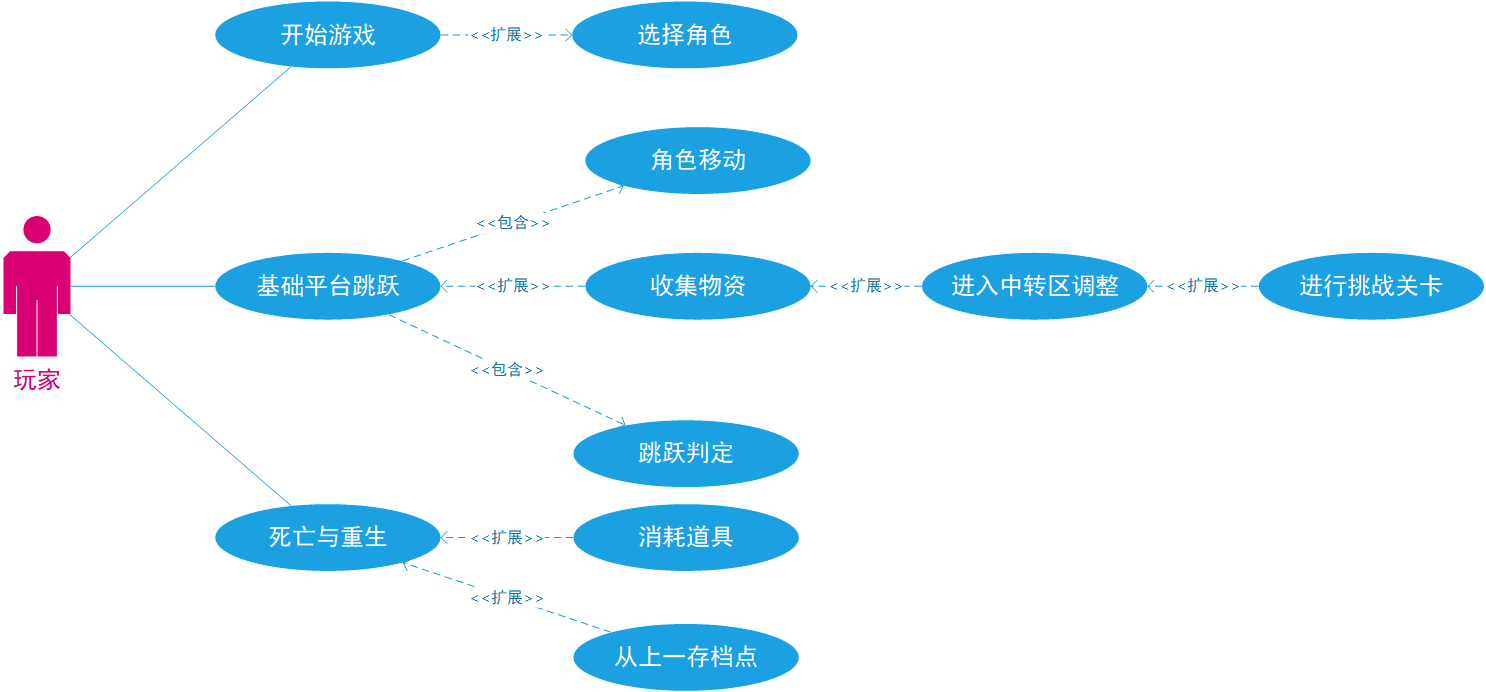


图 1 系统用例图

1. **基础平台跳跃**

**​参与者:** 玩家。

​**操作模式**：左右行走，八个可选择移动方向，跳跃，蹲（防御），二段跳，与地图物件交互。

1. **机制切换与物资收集**

**参与者：**玩家

​**流程:** 玩家达到特定层数→了解新的游戏机制→收集物资→提升能力。

​**扩展关系:** 可扩展为“战斗机制”（需击败敌人后继续攀登）。

1. **成长系统和商店系统**

**参与者**：玩家

**流程**：玩家根据游戏进度或收集的物资，对属性进行升级或者购买可单次使用物资。

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

## 游戏流程

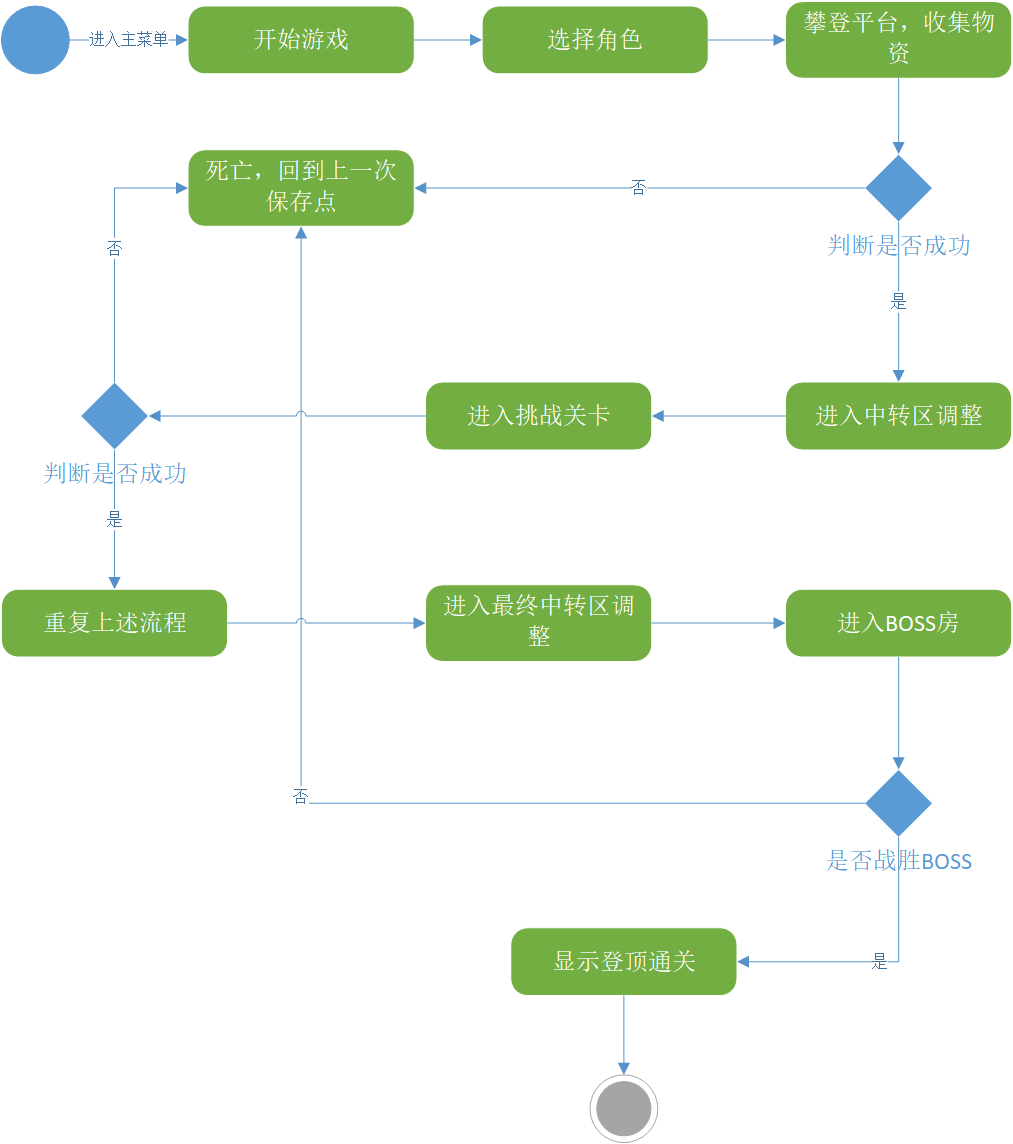


图 2游戏流程图

玩家单局流程​：

开始游戏→选择角色→攀登基础平台→触发机制A（如解谜）→收集物资→能力升级→触发机制B（如BOSS战）→登顶通关或死亡重生。

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。

**模块1：角色控制​**

**​输入: 键盘/触屏操作和地图给予的信号反馈。**

**​功能: 处理移动、跳跃、蹲防、二段跳、攻击指令，同步动画状态，判定角色与地图的交互状态（如攻击、受伤、交互操作）。**

**​模块2：动态关卡系统​**

**​输入: 当前层数、玩家能力等级。**

**​输出: 加载对应预置关卡（含地图设计、特殊机制方块、物资点位）。**

**​规则: 每一大关切换机制（跳跃闯关→与地图互动，了解机制→利用机制对抗最终敌人→循环）。**

**​模块3：成长系统​**

**​数据表: 能力升级树（分为必须能力和可选能力，必选能力在通过一大关后即可解锁）。**

**​功能: 物资收集事件触发/关卡通过事件触发→更新玩家属性→保存至本地存档。**

**模块4：战斗系统**

**输入：角色操作，敌人行为与地图判定条件。**

**功能：角色基础数值的更改，敌人攻击行为和战利品，注意结合地图机制以及角色技能。**

## 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。

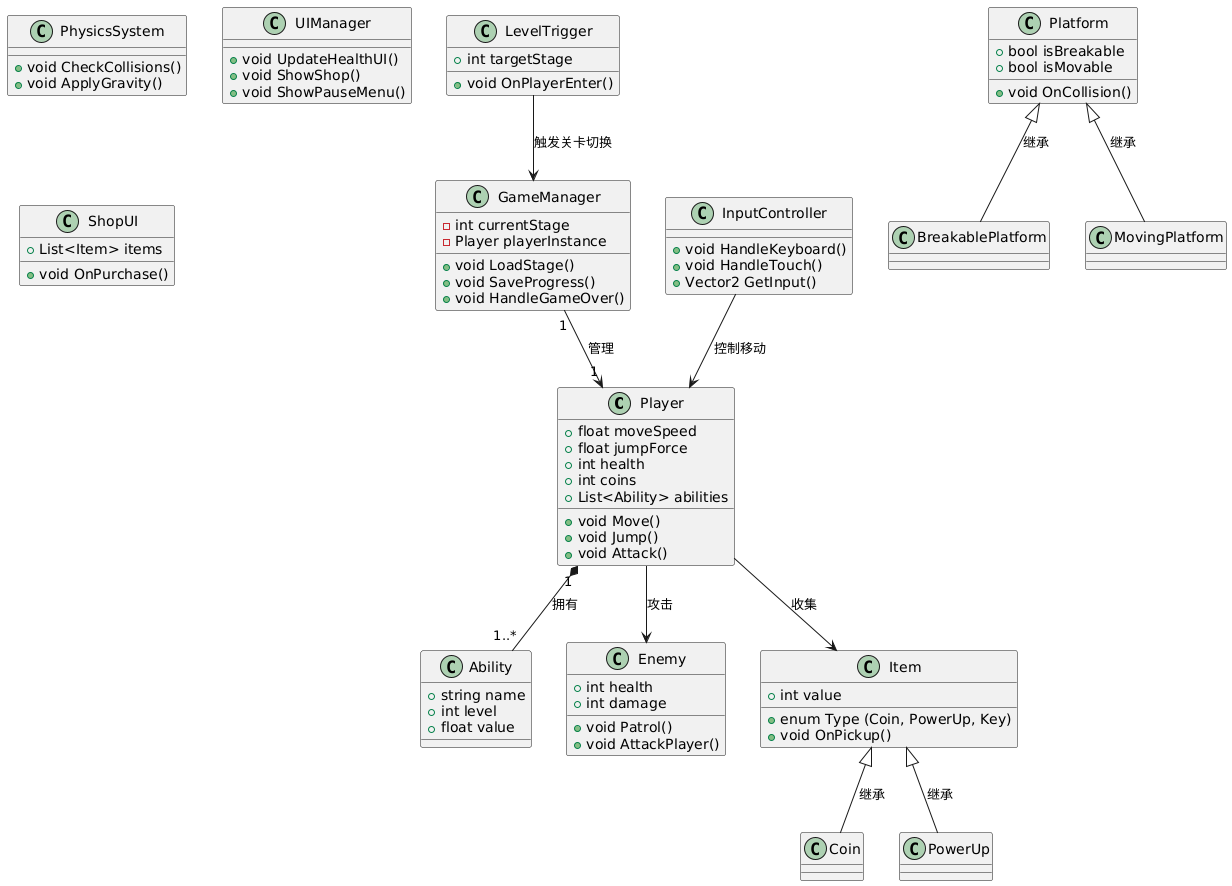


图 4核心类图

## 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

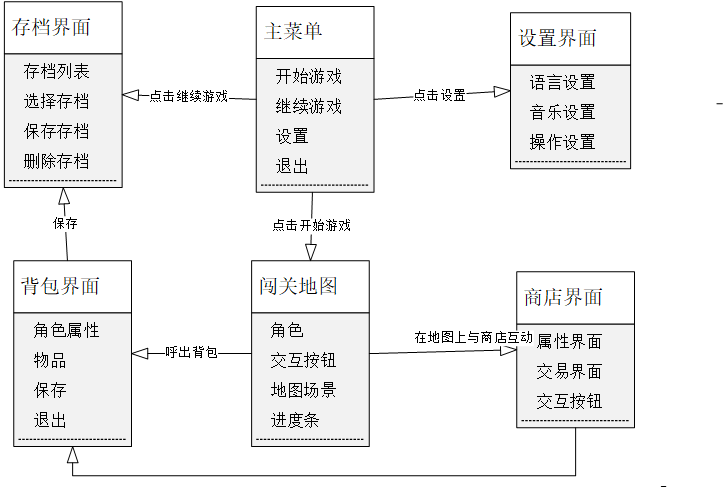


图 5 页面流程图

图 5 系统界面