

# 项目信息

## 项目名

鸡地求生 (Chicken eating)

(来自选题“鸡地求生”)

## 项目成员 & 贡献比

姓名	学号	贡献比 以平均值为100%	贡献比 以总分为100%	班级
王子安	2054088	100%	25%	朱老师
袁枫烨	2051977	100%	25%	赵老师
何庐坤	2053025	100%	25%	赵老师
孙 泽	2052635	100%	25%	赵老师

## 项目地址

[wzawza123/teamwork\\_chicken\\_eating\(github.com\)](https://github.com/wzawza123/teamwork_chicken_eating)

## 进度时间线

日期	功能
5.10	项目创建

- | 5.14 | 类基本设计
- | 5.17 | 第一周计划、确定代码风格
- | 5.23 | 开始界面、单/多人游戏运行界面
- | 5.24 | 暂停功能实现，地图的绘制与引入
- | 5.25 | 可以显示的英雄
- 英雄基本的移动、动画动作
- | 5.27 | 移动动画实现
- 部分场景UI
- | 5.30 | 初步实现了子弹层、子弹类、武器类
- | 5.31 | 地图的视野设置
- 角色UI
- | 6.1 | 使用物理引擎
- | 6.4 | 为角色以及道具设置刚体与碰撞层
- 角色类的优化
- | 6.6 | 实现了武器切换
- | 6.7 | 敌人的初步设计
- UI与武器的交互
- | 6.8 | 角色与武器的交互设计
- 角色与UI的交互

| 6.9 | 实现了子弹命中的功能  
添加了用于测试的敌人  
| 6.11 | 添加了更多的武器  
实现了武器拾取的方法  
| 6.12 | 实现了敌人的武器持有  
实现了交换手中与地面的武器  
| 6.13 | 敌人拥有了射击功能  
设计了敌人AI  
添加了子弹包  
| 6.14 | 实现了与子弹包/弹夹数的UI交互  
新增了大地图  
联机模式数据库的初步搭建  
| 6.15 | 加入了音效  
完善了敌人的生成  
联机模式UI  
联机数据层设计  
| 6.16 | 添加了护甲包  
护甲包功能实现  
实现了角色呼吸回血  
| 6.17 | 单人模式资源载入优化整理  
新UI的添加与优化  
联机模式数据交互  
| 6.18 | 联机模式图形化界面设计  
| 6.19 | 单人模式与联机模式的兼容整合  
| 6.20 | 整理工作  
| 每天 | 修复各种bug

# 项目开发

## 1. 架构设计

### 基类

- Actor
- Item
- Weapon
- Bullet
- myNetwork

具体类在代码中体现，一般一个文件 (.h + .cpp) 就是一个类。

## 2.分工

组员	完成项
王子安	英雄各种相关、技能、敌人各种相关、联机相关、对接组织
袁枫烨	各种武器相关、子弹相关、音效添加，局内道具拾取物相关
何庐坤	地图、物理引擎、刚体设置碰撞检测、联机相关
孙 泽	场景、场景UI、场景切换相关、局内道具拾取物相关

## 难点及解决

---

1. cocos 的坐标获取与设置过程中产生的问题。由于cocos内部的存在三套坐标系统，在getposition()时get到的坐标为该节点相对于父节点的位置坐标，所以需要根据不同情况通过数学手段或调用cocos的坐标转换函数来补偿并设置为正确的坐标位置。
2. 物理引擎的使用。通过物理引擎的使用优化了各种事件的监测的方法与游戏性能，在使用过程中，在碰撞掩饰码的设置、通过tag判断碰撞状态时遇到了很大的困难。对于物理引擎函数不甚了解，且网上的学习资料多数不适用于cocos2d-4.0版本，故学习成本较高。最终通过理解并实践各种函数，找到了各个函数在项目中的用法。
3. 项目的跨平台合作。组内成员使用不同的操作系统及平台进行项目的开发工作，过程中使用了3种不同的编码。导致开发初期时每次项目内容整合与跟进时均因编码不同而产生各种错误。之后找到了问题的解决方法：通过将代码复制进txt文本后在复制入文件或使用转编码软件进行批量转码。
4. 英雄视角跟随问题。由于吃鸡游戏需使用大地图，需要为人物设置动态视野，视野以主角为中心进行跟随，且不能露出地图外的区域。初期解决方案为利用数学方法对地图进行移动，利用相对移动实现视觉效果。但考虑到后期联网功能的实现，所以最后决定引入camera视角相机Layer来进行视野的移动。如此做也为跟随移动的UI的实现提供了途径。
5. 为多边形对象层设置刚体。难点在于资料甚少，相关资料只有为规则形状对象层设置刚体。最后通过对变量的解读与实践弄清了各个变量的含义与功用。
6. 联机游戏的实现问题。联机模式服务器的搭建，本地与服务器的数据交互、联机模式对类的机构与设计的要求均存在许多问题。初期从单人模式设计的角度出发来设计相关的类，但当开始联机模式的设计时发现之前的类的设计需要重新设计以适应联机的需求。

## 踩坑

---

1. cocos2d所能容纳图片的最高像素是4096\*4096，像素高于最高像素的地图资源无法加入项目中
2. cocos2d中的getpositon()得到的是该节点相对于父节点的相对坐标。
3. 类的设计在后期需要根据联机的实际需求进行返工。

## 亮点

---

1. 每一个物体与物体之间的判断都是依靠物理引擎来实现的，不同物体与英雄的碰撞响应不同。
2. 在游戏中引入了大地图，满足多人游戏的需求。
3. 引入视野系统，做到了视野跟随角色移动。
4. 引入了智能AI索敌系统，敌人采取会采取“风筝打法”，提高了游戏难度与趣味性。
5. 设计了多种武器与多种道具，大大提高了游戏的可玩性。
6. 用了一个 Const.h 管理全部的常量，在代码中尽量不使用裸值。
7. 引入了计时器，实现了游戏时间的记录与角色呼吸回血。
8. 功能全面的UI

## 基本项

---

单机部分全部完成

联机部分已完成数据层的搭建，实现基础的数据交互，房间创建、加入等功能，只是游戏逻辑尚未完全完成。

## 加分项

---

- 多种多样的局内道具物品（各种子弹包、护甲包）
- 游戏音效（包括BGM，攻击、换弹、换武器等多种音效）
- 画面精致，UI多类
- 人物延时呼吸回血

- 外挂（当某两个键同时按下时会有特殊效果）
- 移动方式采取八向移动方式。

## C++11 特性

---

- [初始化列表]
- [lamda 表达式]
- [类型推断 (auto) ]
- [范围 for 循环]