



一、项目分工

学号	名字	角色	班级	职责	贡献
15331209	林子琪	组长	上午班	负责实验报告和素材准备	25%
15331213	刘芳	组员	上午班	负责登陆和排行界面	25%
15331232	罗文璐	组员	晚上班	负责游戏主要实现	25%
15331207	林泽云	组员	上午班	负责游戏主要实现	25%

二、开发环境

编程语言：C++

Window10,

Visual Studio 2015

Cocos2d

三、项目阐述

1. 名称

老鹰抓母鸡

2. 简介

农场里和平共处的农民和一群母鸡，母鸡负责下蛋，农夫收集鸡蛋换取钱财（积分）。天上盘旋着老鹰，老鹰瞄准母鸡，不定时飞下来把母鸡抓走。农夫作为母鸡的守护者，负责狙击天上的老鹰。当母鸡全部被老鹰抓走，游戏结束。

3. 玩法

控制农夫 WSAD 上下左右移动，空格键发射子弹狙击天上的老鹰，鼠标点击鸡蛋收集鸡蛋得分。当母鸡数量为 0，游戏结束，统计得分。

4. 应用知识点

- ① 小人发射子弹控制：屏幕触发触屏相应事件、监听器
- ② .鸡蛋的选中：鼠标触摸事件，touch 坐标



- ③ 自定义调度器(schedule)---游戏界面
- ④ 动画、动画切割、字体格式、音效
- ⑤ CC_CALLBACK_X 回调函数、Moveby\MoveTo 等
- ⑥ 随机数、碰撞检测
- ⑦ HttpClient, sessionId
- ⑧

四、项目展示

初始登陆界面，输入用户名，进入游戏。



图 1-初始界面

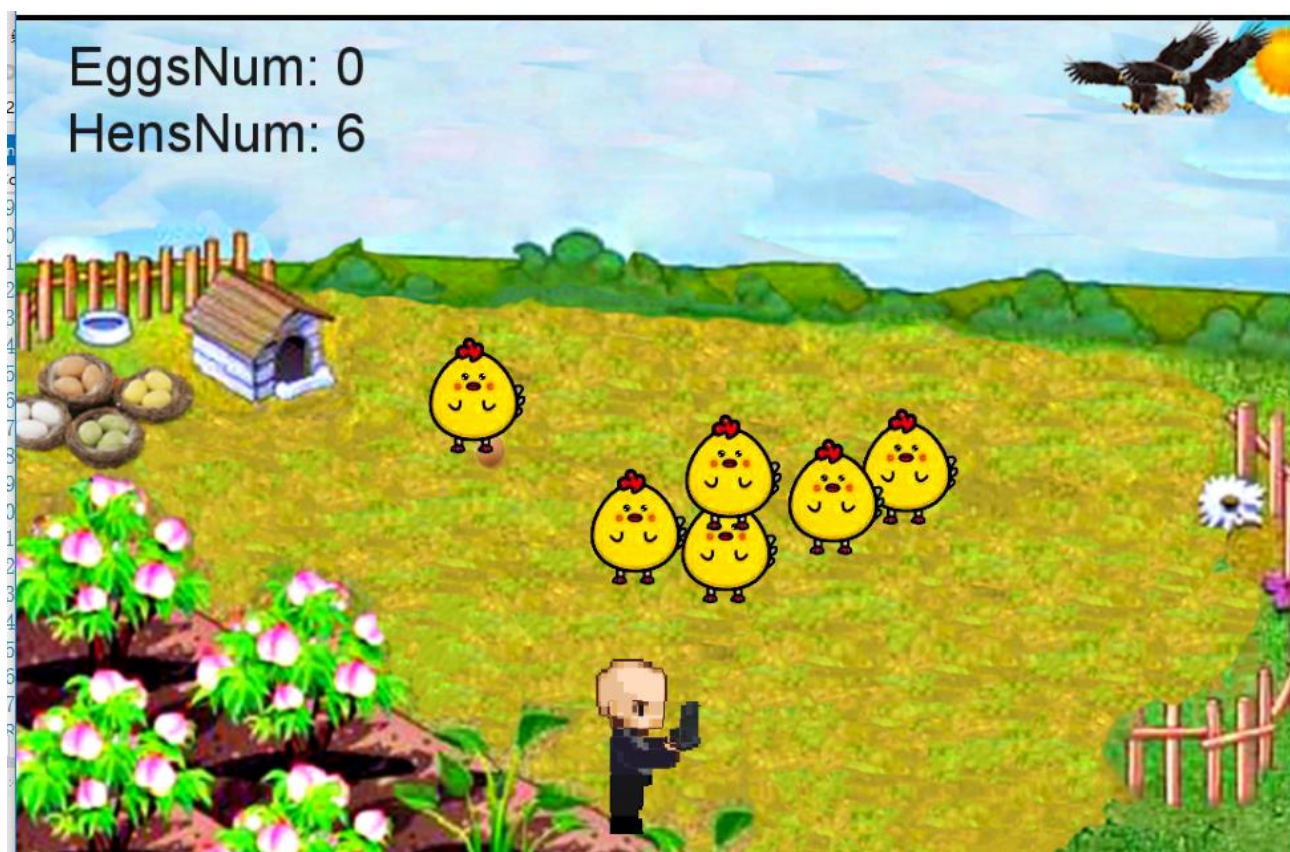


图 2-游戏主界面

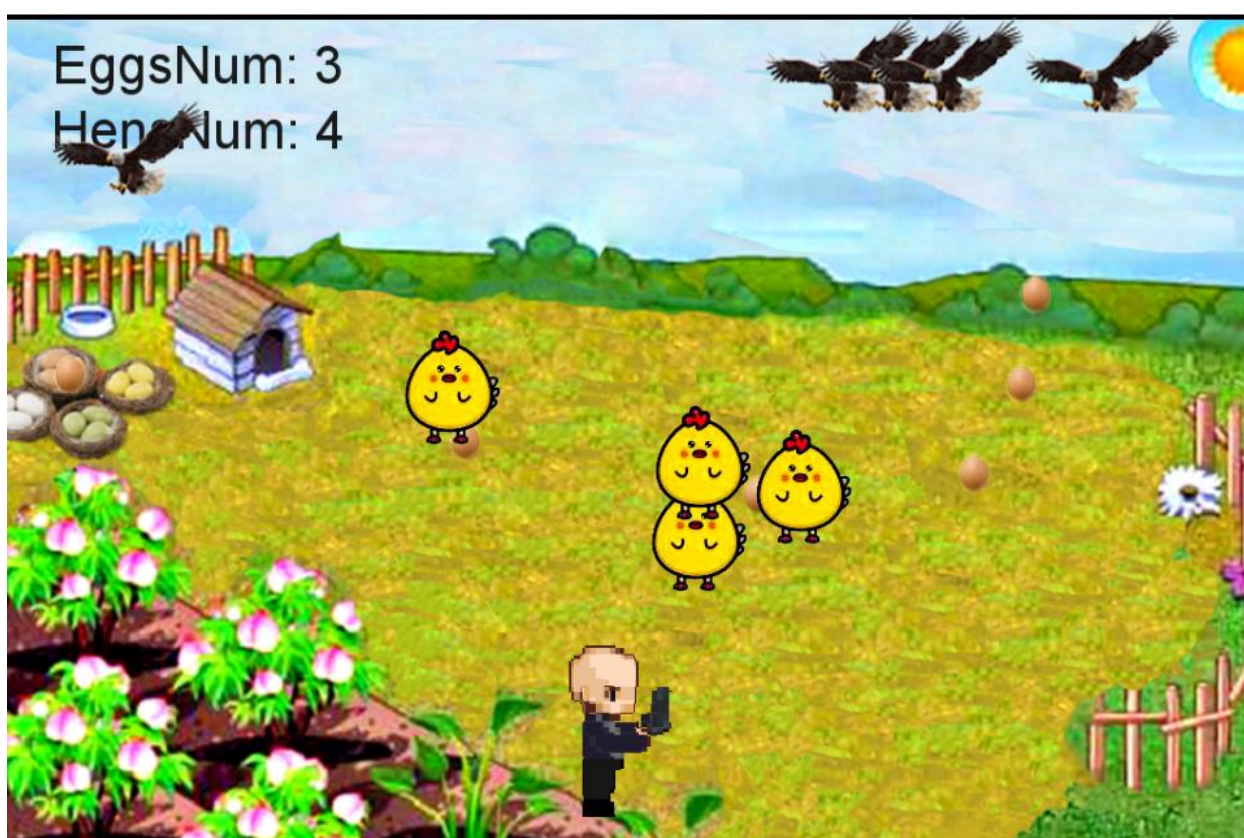


图 3-老鹰抓走母鸡



图 4-农夫狙击老鹰

母鸡全部被抓走，游戏结束。

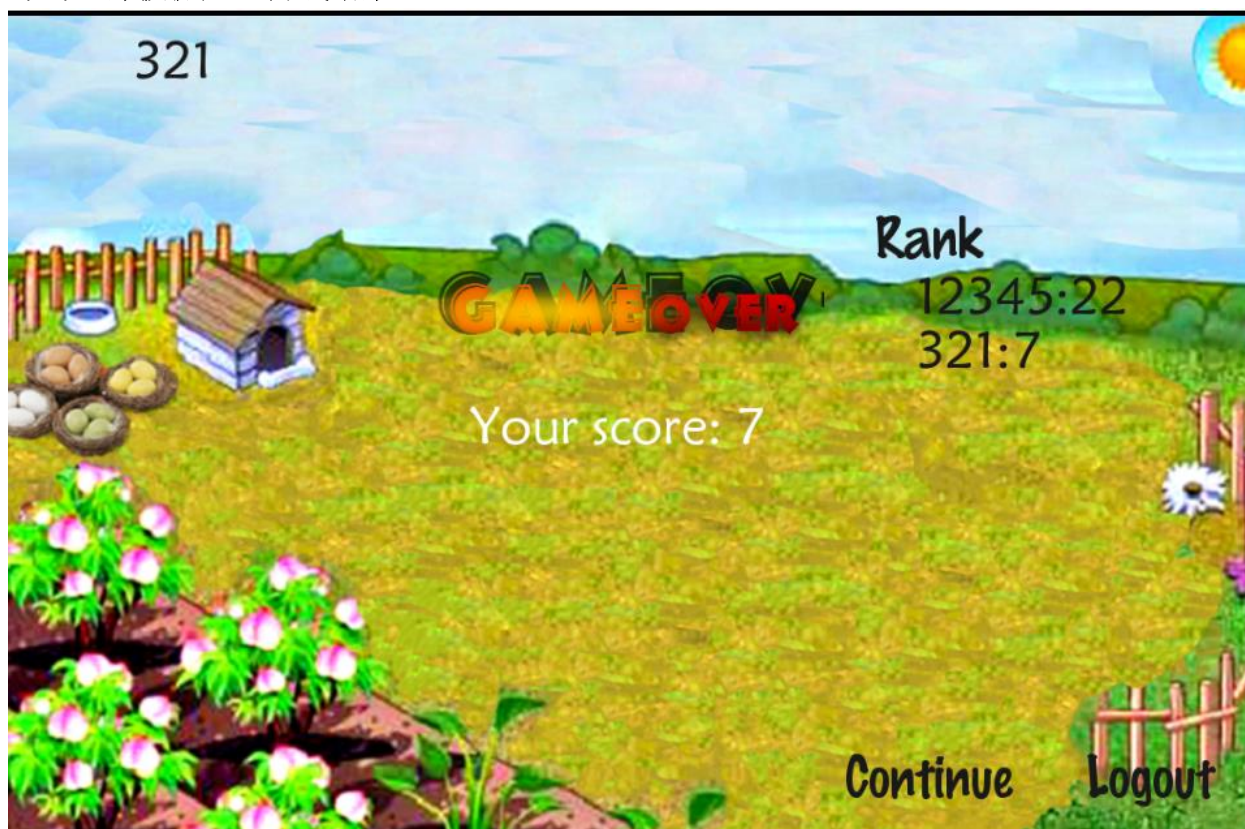


图 5-结束画面及排行显示



五、项目难点及解决方案

- ① 素材比较难找，特别是农夫的枪口朝向上，需要用 PS 解决
- ② 动画帧切割，使用 `visibleSize` 一直出错，原因是在 `init` 函数中用了 `auto visibleSize = Director::getInstance()->getVisibleSize();` 而不是 `visibleSize = Director::getInstance()->getVisibleSize();`
- ③ 子弹不是直接从 `player->getPosition()` 开始发射，需要采用偏移，当人物面向左和面向右时水平偏移量为相反数，需要采用 `bool` 变量进行控制。
- ④ 老鹰抓母鸡：
 - a. 因为老鹰有很多，母鸡有很多，所以不能采用随机捕捉，不仅实现难，而且界面乱。队头的老鹰飞到要捕捉的母鸡（队头的母鸡）后飞回天空的位置的确定----为了不会出现一直是同一只老鹰在抓鸡的情况，老鹰飞回的位置应该需要设置在接近 `position.x = 0` 的位置。
 - b. 老鹰的移动需要时间，但是又需要对队头老鹰和队头母鸡进行距离的判断，从而确定是否已经捕捉到。最后采用的是在 `update` 中通过静态变量来控制时间，因为老鹰飞向母鸡需要 1s 的时间，而 `update` 函数每 0.04s 更新一次，所以可以实现随时跟踪当前老鹰与母鸡之间的距离。

六、项目总结

这次游戏项目规划成三个模块来实现（除去登录和结束界面），老鹰、母鸡和蛋、农民。分而治之，用接口实现他们之间的交互，再具体地进行函数实现。

这个游戏是灵光一闪想要实现的，一开始只是有一个大概的轮廓觉得这也可以那也可以，实现起来才发现其实有很多细节不像当初设计的那么简单。

举个例子，一开始设想的是母鸡在院子里乱走，随机停下然后生蛋。具体实现的时候发现要考虑动画素材来源的问题，还要考虑生蛋过程中如果有老鹰飞下



来的情况，还有由于随意乱走引发的碰撞和“蛋碎”的问题，于是决定简化这个设定：鸡蛋用随机数产生，不需要“绑定”某一只母鸡。还有老鹰飞下来抓住母鸡这一个“核心”，要考虑抓住当前离自己最近的那一只，但是要考虑到母鸡如果能够随意走动，那么 `moveTo` 的坐标就会发生变化，如果考虑忽略母鸡走动的距离，即把 `getdistance` 的碰撞检测范围加大，就很可能抓到另一只鸡.....虽然看上去效果是差不多的，但是这个效果并不受设计者的控制，就很不好。至于鸡蛋产生和收集鸡蛋，需要对蛋设计两个链表，在这里出现过很多次错误，比如 `eggsnum` 和 `hensnum` 是标签，不用++，否则出现中断错误；随机产生的 `eggs` 和点击蛋后产生的 `clিকেeggs`，尤其注意要判断蛋的位置和鼠标点击位置是否一致时要遍历的 `eggs` 是否为空等等等等。

实现期末项目的时候正面临期末考试，大家都比较随意，也想过直接把某次作业修改之后当成期末项目交上去，方便省事。不过最终还是努力把这个项目完成了，虽然画面并不是很精美，也没有特别“好玩”，但也算团队合作实现的第一个游戏项目，还是很有意义的。

通过这次游戏项目，更加能够理解团队合作的重要性，在找不到素材、出现各种错误的情况下，团队成员之间能够互相帮助，虽然不是分配给自己的任务，也能尽心尽力地去帮助实现。也更加能够体会到完成一个项目的复杂性，不仅需要统筹兼顾整个项目的总进度，还要对分而治之的各个模块进行及时的修正。总而言之，收获巨大。