

**信息与软件工程学院**

**项目报告**

课程名称： 程序设计项目实践（PBLF）

学 期： 2021-2022第1学期

项目名称： 烧脑游戏串烧

授课教师： 陈伟

|  |  |  |
| --- | --- | --- |
| 序号 | 学号 | 姓名 |
| 1（组长） | 2021090906016 | 刘芮利 |
| 2 | 2021090906020 | 张庭恺 |
| 3 | 2021090906002 | 赵晗 |
| 4 | 2021090906010 | 钟城伟 |
| 5 | 2021090906008 | 黄培坤 |

目录

[1 项目简介 3](#_Toc91795855)

[1.1 考核方式 3](#_Toc91795856)

[1.2 项目题目及内容简介 3](#_Toc91795857)

[1.3 项目组成员与分工 4](#_Toc91795858)

[2 需求分析 4](#_Toc91795859)

[2.1 选题的依据 4](#_Toc91795860)

[2.2 功能需求 5](#_Toc91795861)

[3 系统设计 6](#_Toc91795862)

[3.1 总体设计 6](#_Toc91795863)

[3.2 模块设计 7](#_Toc91795864)

[4 系统实现 8](#_Toc91795865)

[4.1 主函数 8](#_Toc91795866)

[4.2 其他函数 9](#_Toc91795867)

[5 测试 9](#_Toc91795868)

[6 总结 10](#_Toc91795869)

# 1 项目简介

## 1.1 考核方式

总成绩 = 项目和项目文档成绩(40%) + 汇报幻灯片成绩(20%)

+ 表达能力(20%) + 团队合作(20%)

## 1.2 项目题目及内容简介

* 项目题目：烧脑游戏串烧（小游戏合集）
* 内容简介：项目成果是一款益智游戏主题的小游戏合集，具体包括猜数游戏、猜拳游戏、取石子博弈、吃豆人、长城游戏、俄罗斯方块，每一个游戏有多个关卡和不同模式可供选择。支持多用户注册、登录和PK，天梯排名系统可以显示所有用户的游戏得分，同时每个游戏也可以查看历史得分记录排名。游戏数据使用文件保存和读取，为了保障数据安全，程序数据采用RSA非对称加密算法对游戏数据进行加密。每个游戏的简要介绍如下：
  + - 猜数游戏：玩家猜测程序随机生成的数字，有一定次数限制。根据猜测次数和用时计算得分。
    - 猜拳游戏：玩家与电脑进行猜拳，通过模拟石头剪刀布来实 现。玩家可进行多次游戏来对自己今天的运气进行没有科学依据的判断。
    - 取石子博弈：玩家和AI每次可从任意一堆石子里取出任意多枚 石子扔掉，可以取完，不能不取，每次只能从一堆里取。取走最 后一颗石子的一方获得胜利。根据AI难度、撤回次数和用时计算得分。
    - 吃豆人：玩家可使用键盘操控吃豆人躲避敌人攻击获取分数，可通过增幅道具增加速度
    - 长城游戏：在NN方阵中有N个‘/’，请玩家数出移动‘/’使之形成一条直线“长城”的最小步数。答案正确则获胜。有三次回答机会，并计时，所用时间越少或回答关卡数越靠后分数越高。
    - 俄罗斯方块：玩家可以进行四种难度模式选择，难度越大即俄罗斯方块下落速度越快，满行即清除并且记1000分

## 1.3 项目组成员与分工

成员：刘芮利、赵晗、黄培坤、张庭恺、钟城伟

刘芮利：游戏框架设计、头文件设计与编写、代码后期合成、登录/注册系统、关卡切换、天梯排名系统、加解密模块、数据存储、猜数游戏编写、取石子博弈游戏编写

赵晗：猜拳游戏编写（使用Windows.h头文件实现对system和sleep函数的应用，使程序更具有现实性和灵动性;独立设计case\_win和echoinput两个函数对主函数进行“瘦身”;利用时间作种子生成随机数实现真随机）

黄培坤：编写小游戏俄罗斯方块，学习并使用颜色函数，解决碰撞判断法性问题以及研究方块旋转逻辑

张庭恺：吃豆人游戏的编写

钟城伟：长城游戏编写（包括1.最短步数的计算，等价于最佳匹配，使用了KM算法，大大缩短了时间复杂度，由缩减为。2.作了获胜或失败后的美化。3.追踪生成答案方阵。4.利用字符数组读入用户输入，再进行判断，增强程序的排错能力，避免出现bug。5.使用时间函数，以达到计时目的。6.巧妙使用多个goto语句，以达成选择性循环。）

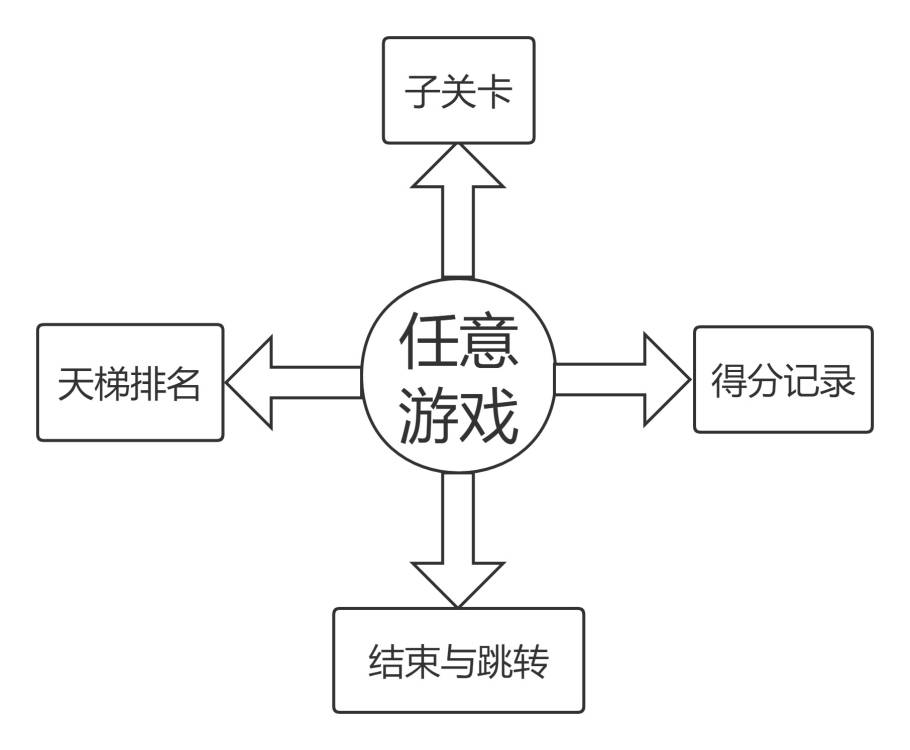
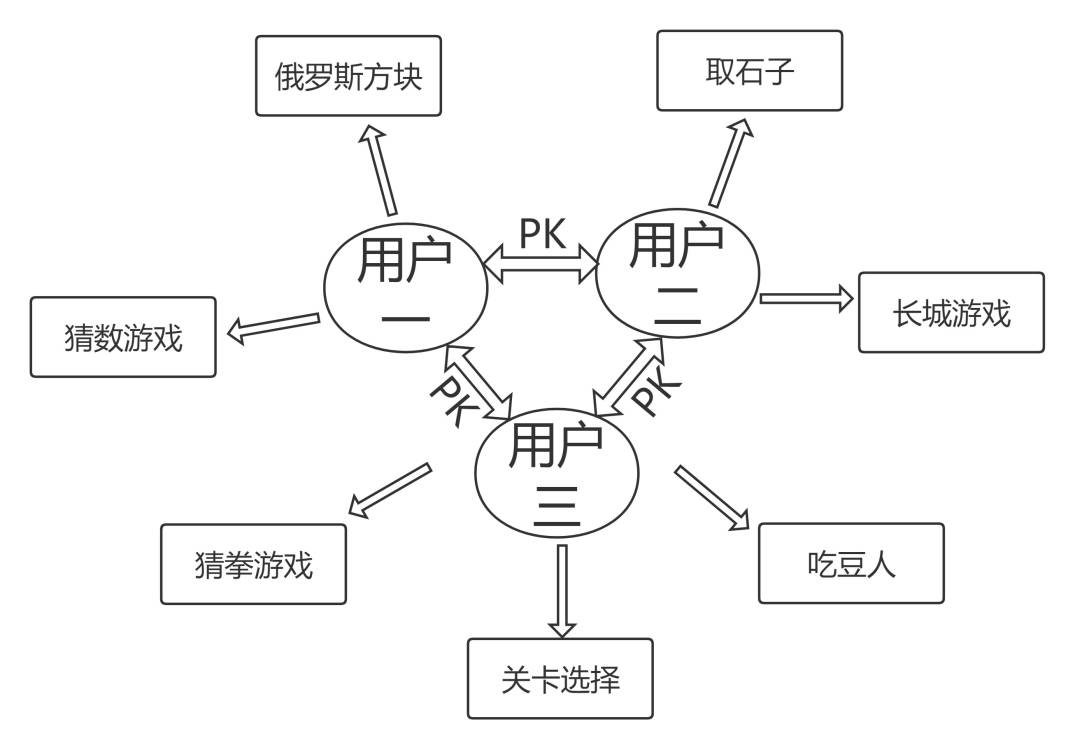
# 2 需求分析

## 2.1 选题的依据

* 俄罗斯方块、Nim博弈、吃豆人等都是经典小游戏，通过这学期所学知识，复现经典小游戏提高C语言代码能力，同时可供闲暇之余消遣娱乐
* 多用户登录/注册、用户得分排名、历史得分记录等是小游戏体验的基本需要，项目对这些功能进行了简易还原，以此提高了游戏的可玩性和竞技性
* 为防止游戏数据被恶意篡改，数据存储安全对程序正常、可靠的运行至关重要。游戏采用现今最广泛运用的RSA非对称加密算法对数据进行加解密，保障数据安全

### 2.2 功能需求

下面通过UML示例图说明程序功能：



# 3 系统设计

## 3.1 总体设计

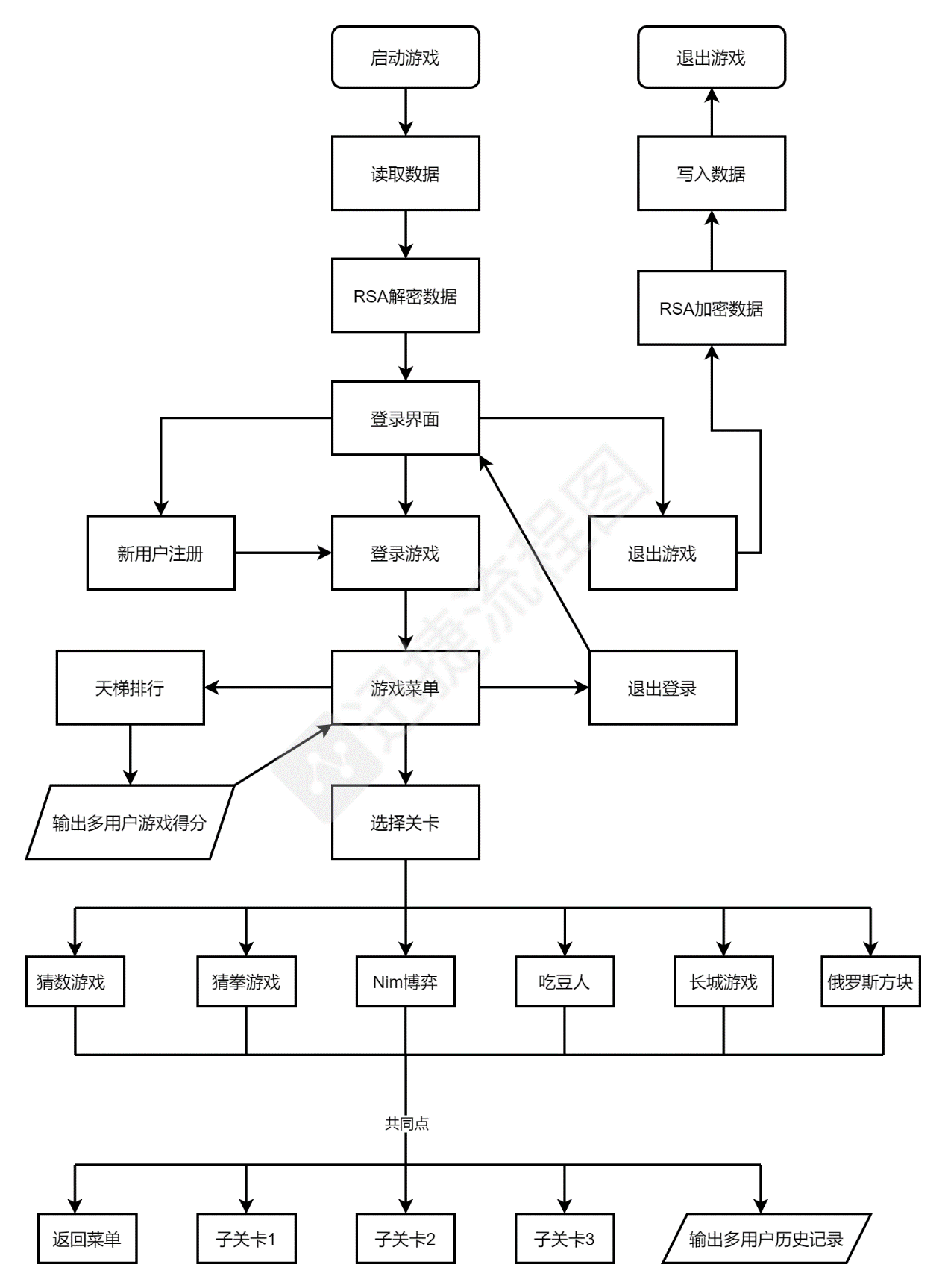
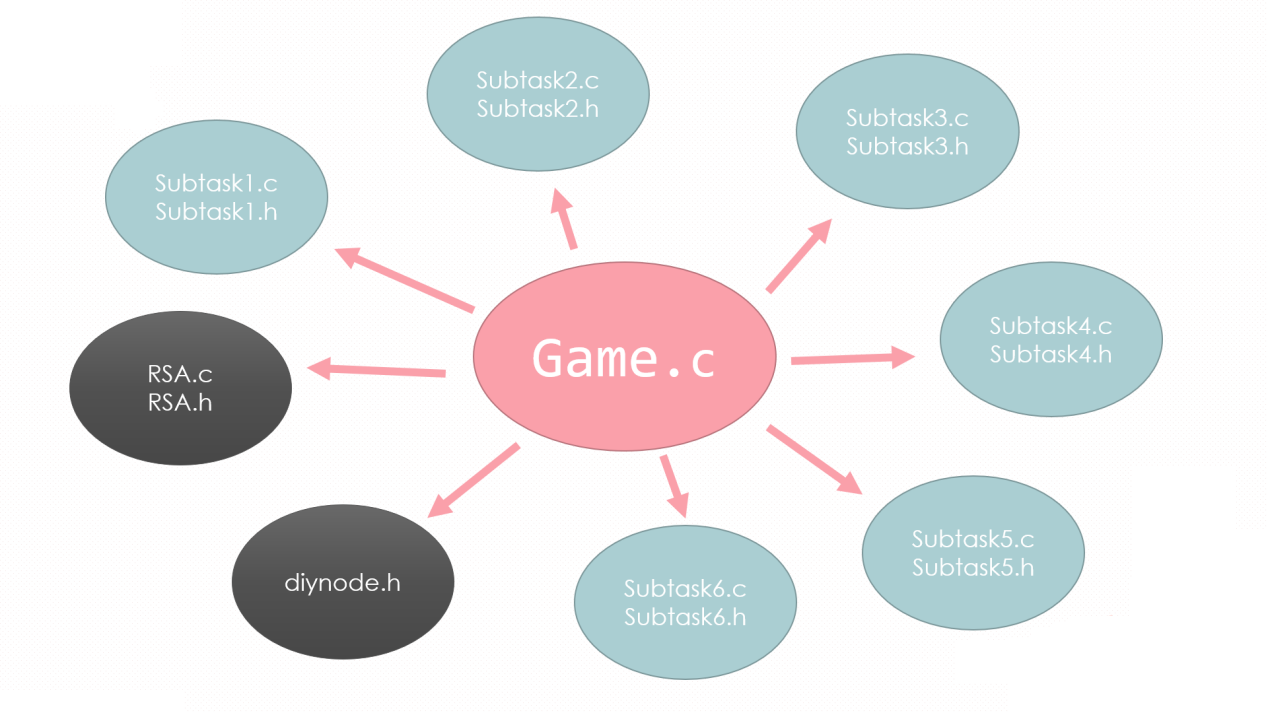


图3-1

## 3.2 模块设计



* **Subtask.h**

1. int Max(int, int); *//求最小值函数*
2. int Min(int, int); *//求最大值函数*
3. void Switchtask(); *//切换关卡函数*
4. void Subtask3(); *//你的游戏函数入口，这里以Subtask3举例*
5. void ShowRank(int); *//展示对应游戏用户排名*

* **Subtask.c**

1. #include "Subtask2.h" *//引入你的头文件，这里以Subtask2举例*
2. #include "diynode.h"  *//引入自定义结构体类型,*
3. #include<stdio.h>
4. extern node Data[]; *//共享游戏数据Data*
5. extern int id, taskid; *//共享id, taskid, id为当前用户对应的数字序号, 用作Data[]下标, taskid表示当前正在进行或即将进入的关卡, 你很可能会用到*
6. *//你可以在下面写函数，定义变量等等*
7. void Subtask2(){ *//下面将是你的游戏入口, 确保return后将taskid更改为合适的值, 1~Gamecnt表示对应关卡, -1表示退出游戏*
8. puts("Subtask2:"); *//这只是一个演示测试而已, 建议删除*
9. Data[id].taskid = Data[id].taskid > 2 ? Data[id].taskid : 2; *//更新用户的最高关卡记录*
10. Switchtask(); *//切换关卡, 可在退出时调用*
11. }

* **Diynode.h**

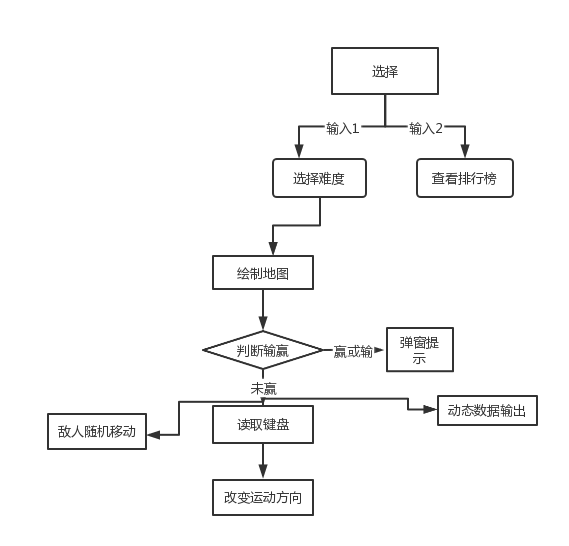
1. #define GameCnt 8
2. typedef struct Node{ *//存储游戏数据的自定义结构体*
3. char username[12];
4. char password[35]; int taskid;
5. int data[GameCnt+5][10];
6. }node;

**注：以上模块每个成员共用的代码模板，便于统一共享游戏数据和后期代码整合**

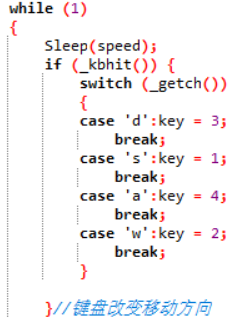
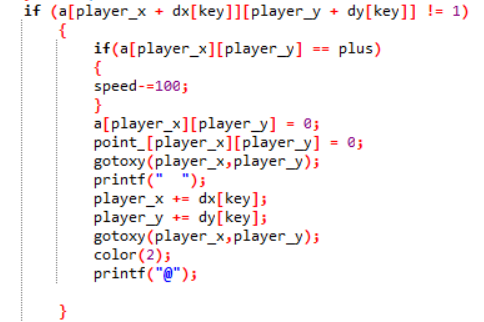
# 4 系统实现

## 4.1 主函数

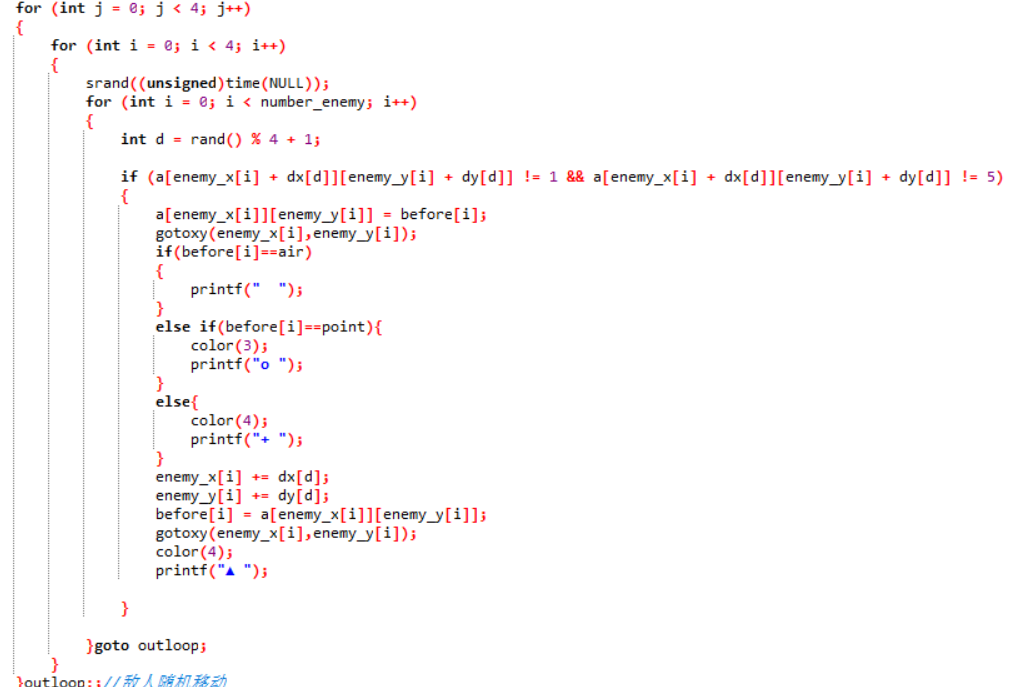
用程序流程图说明实现思路



关键代码分析



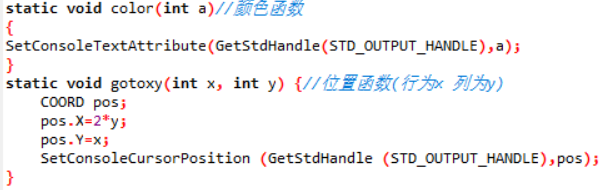
此部分代码实现了对键盘信息的收集，并以此控制玩家移动方向





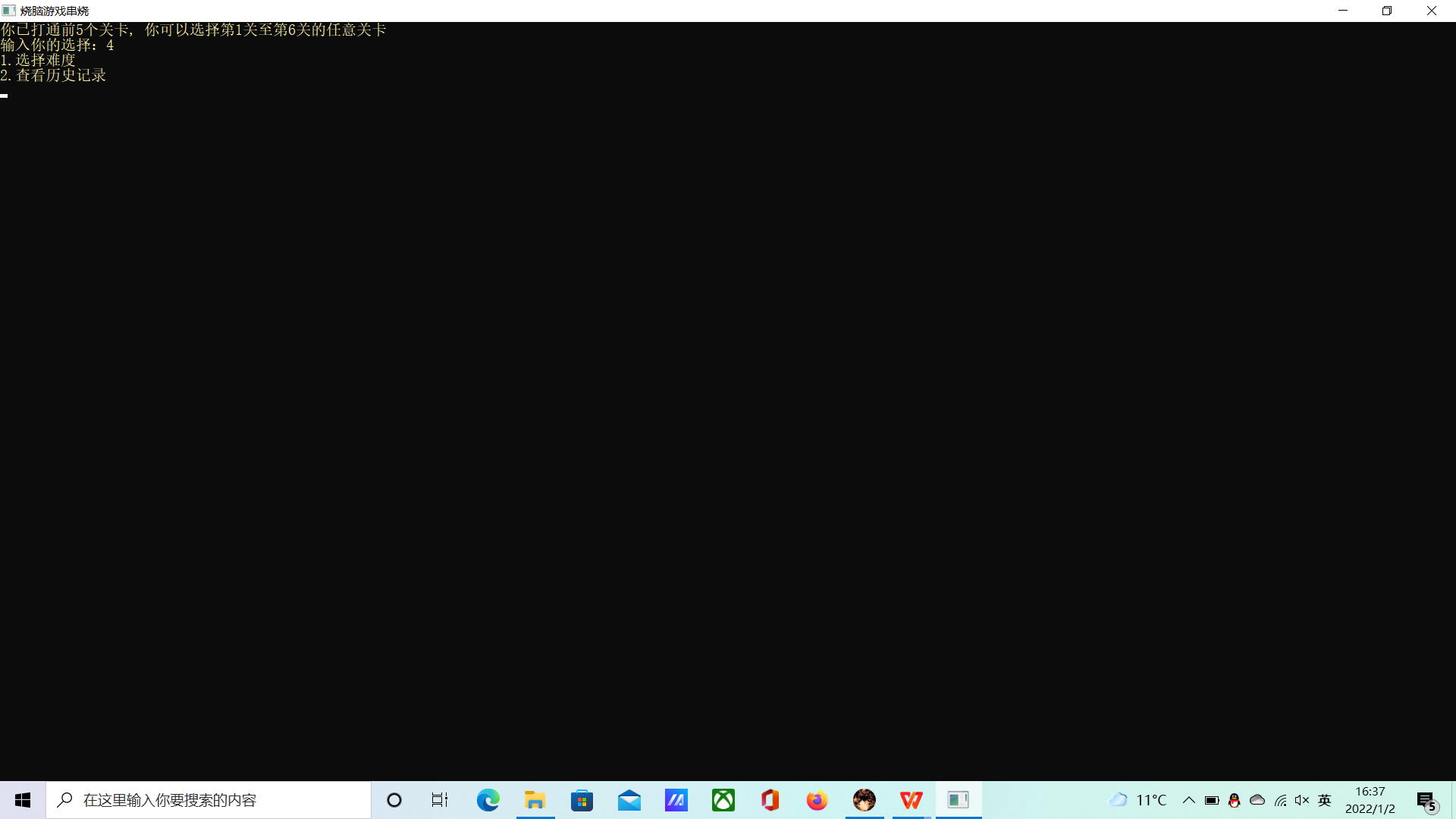
此部分通过随机步法和循环结构实现了每一个敌人的随机走动，同时通过对敌人足迹的记录，实现了地图的复原，并防止分数和增幅道具被敌人吃掉

## 4.2 其他函数

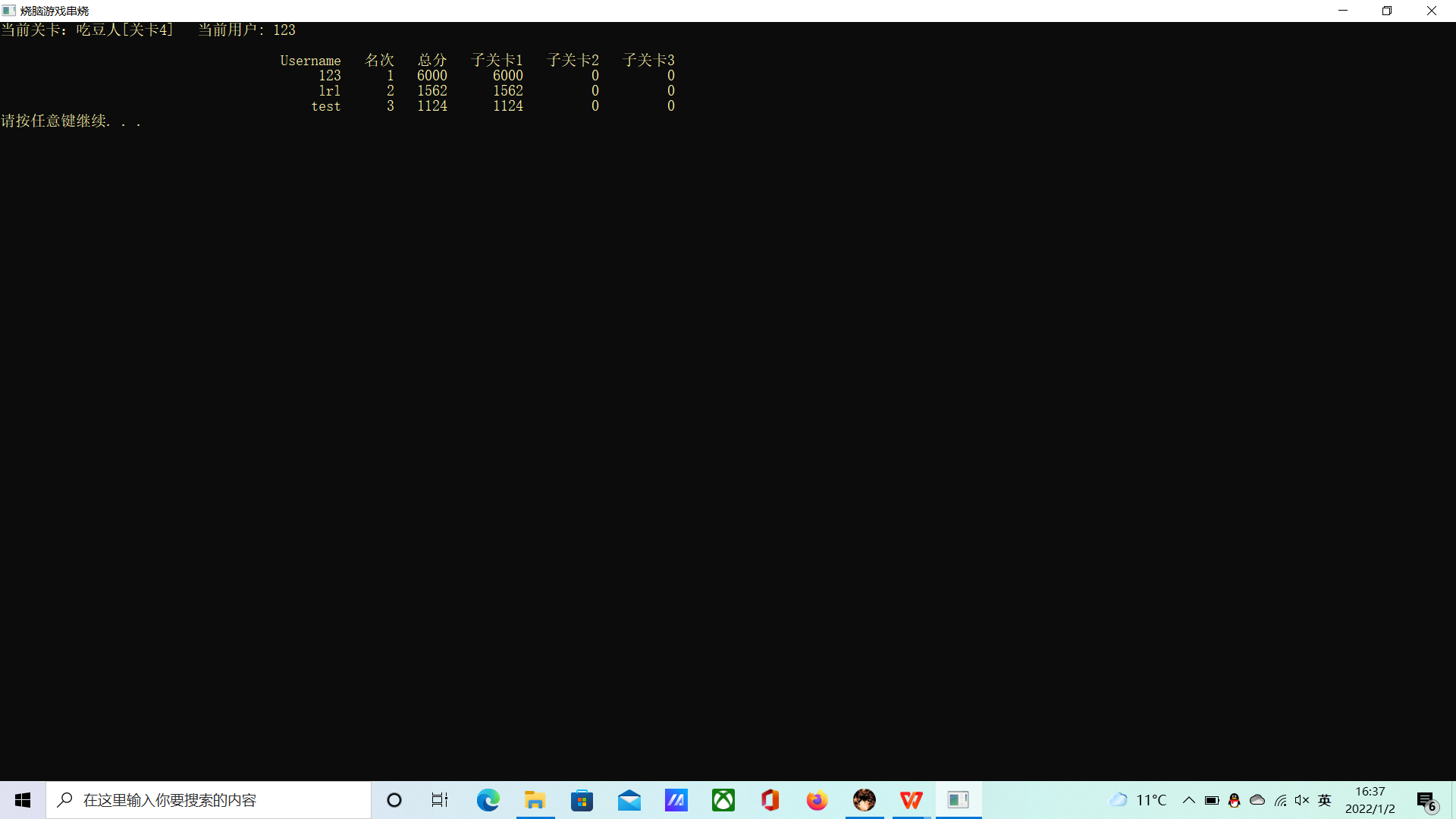


此部分是实现地图绘制及动态数据输出的主要函数，通过颜色和位置函数，能够较好的防止静态数据的重复输出

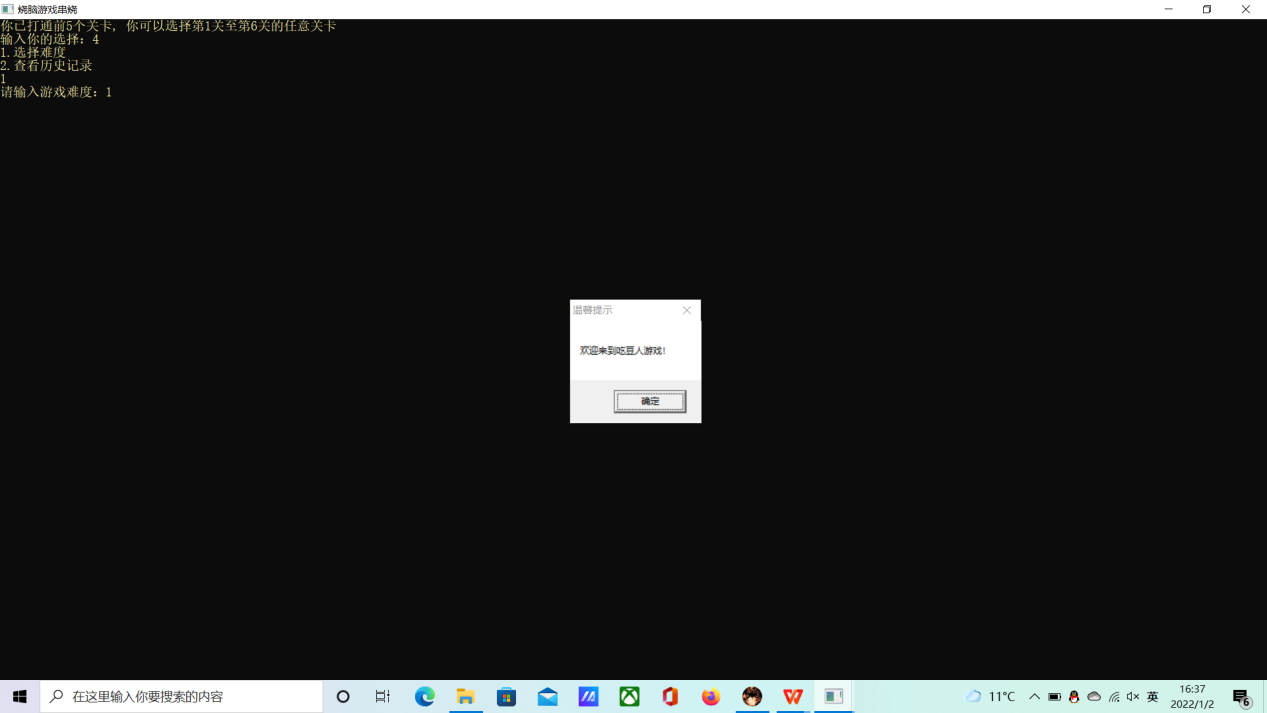
# 5 测试



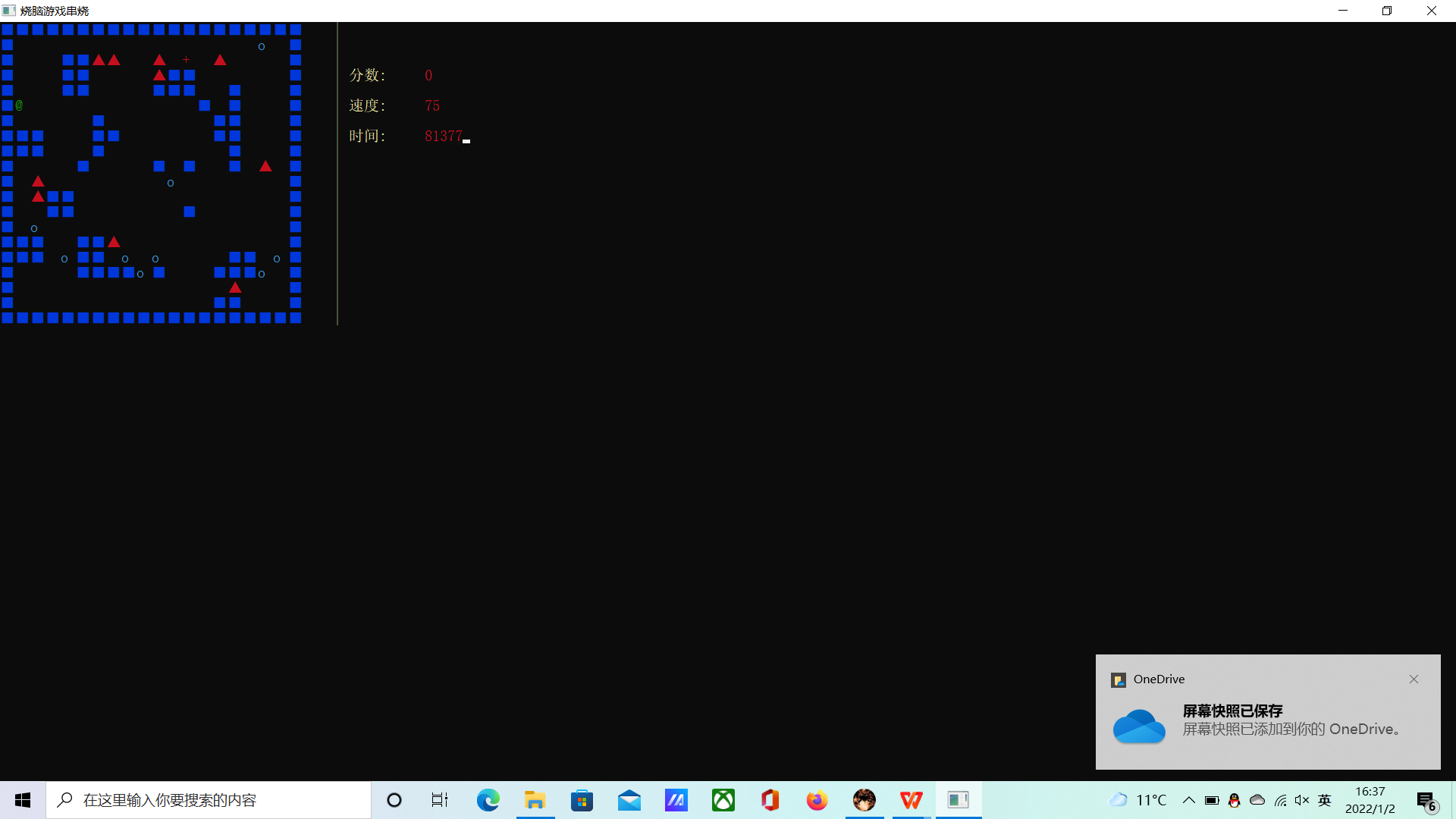
起始选择界面



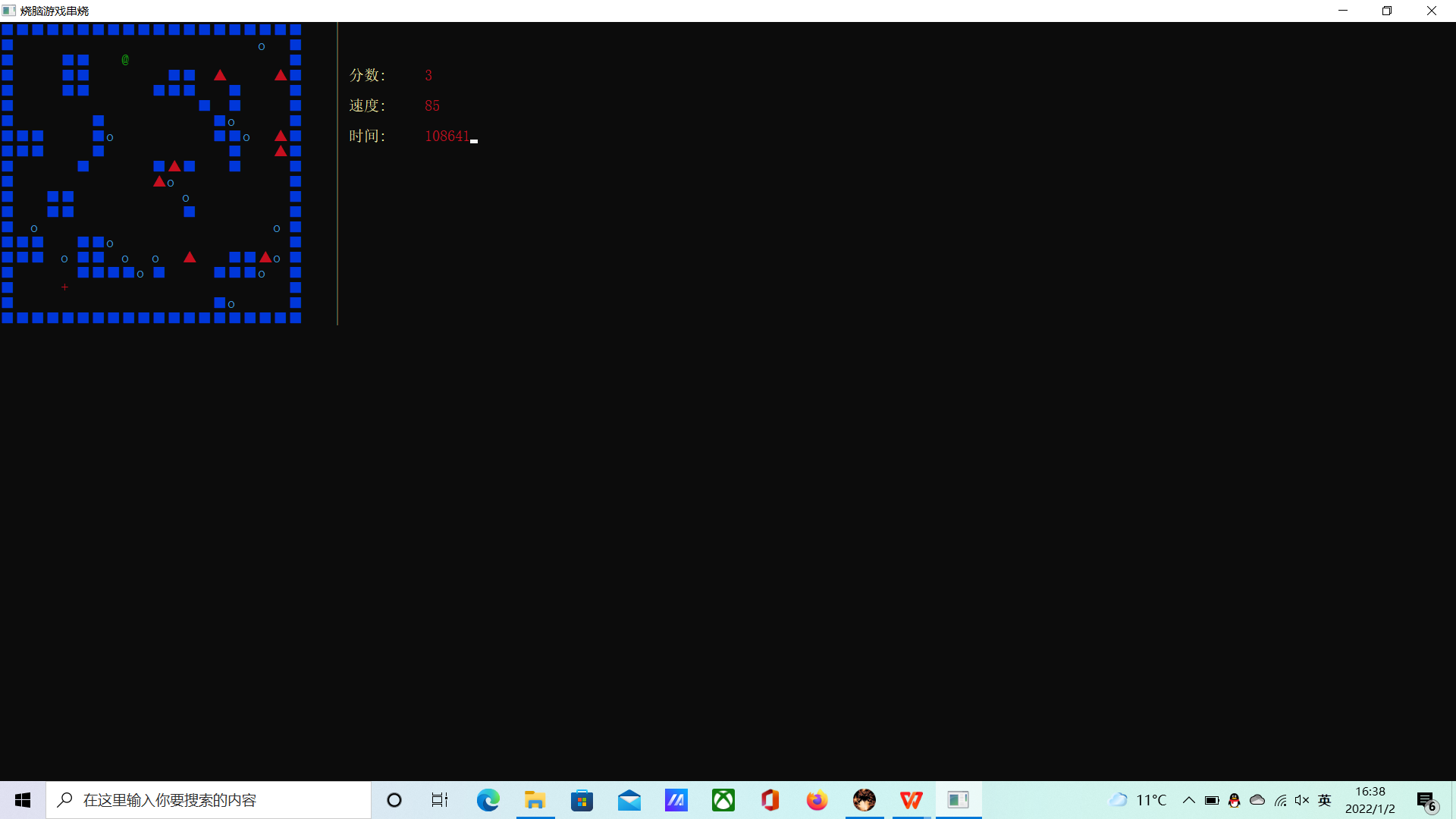
排行榜界面



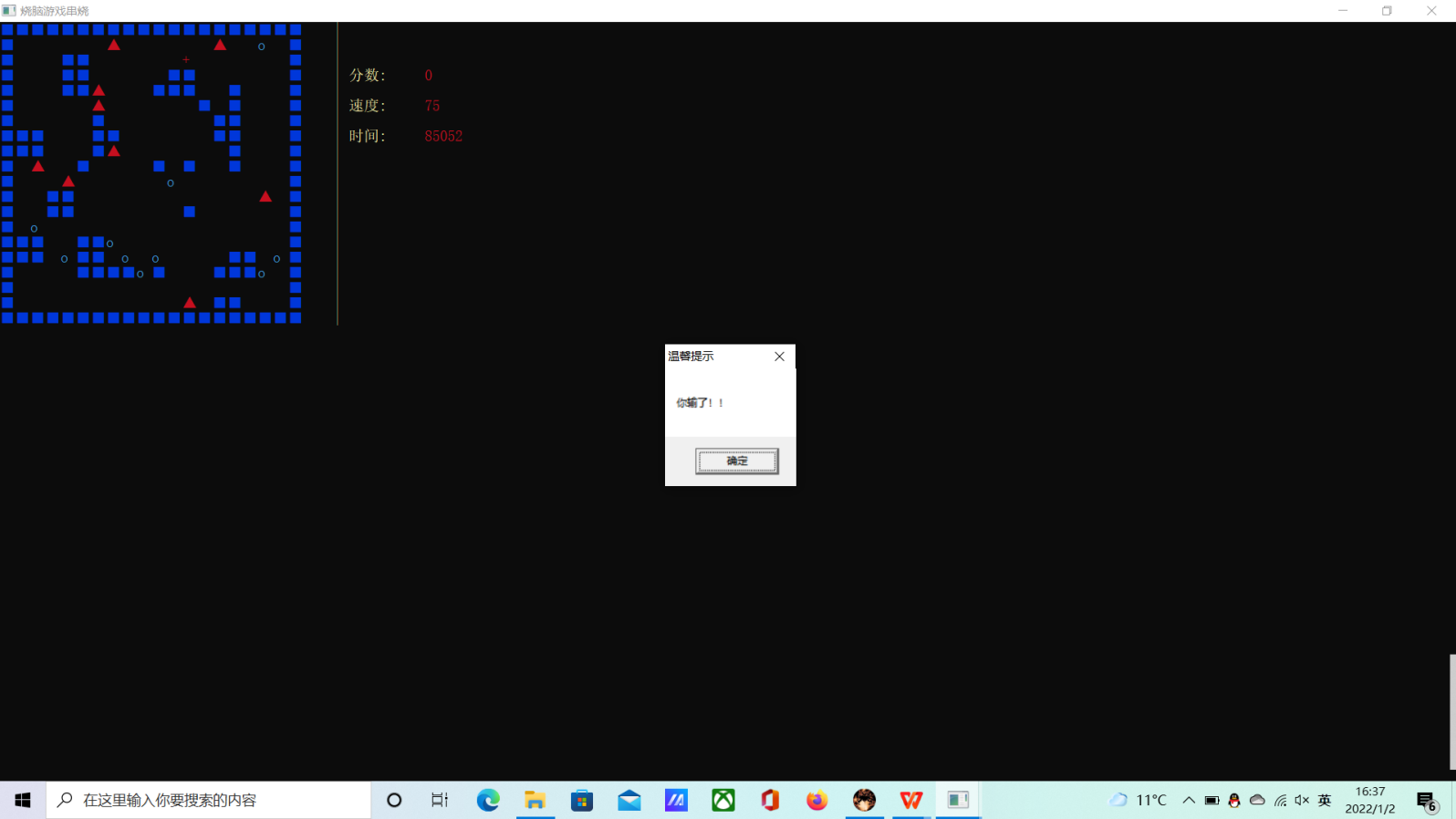
进入前弹窗提示



游戏界面



获得分数及增幅道具后，动态数据的改变



游戏结束后弹窗提示

# 6 总结

**张庭恺 2021090906020：你的总结**

学到了什么？

在编写项目的过程中我逐步了解到图形化编程及键鼠输入信息等的读取，让我对C语言的小黑窗有了更多的认识，让程序不仅仅局限于控制台，增加了人机交互的过程。

痛点和难点

编写过程中，由于一开始对于地图模型等静态数据以及敌人和玩家等动态数据简单的同时输出，导致画面频闪严重，体验感极差，但无从解决，同时由于敌人行进过程中会将走过的地图改为air，因此会导致部分分数和道具被敌人吃掉，出现bug。

自己的贡献

我通过逐步深入了解人机交互相关的函数，完成了吃豆人游戏的编写。

如何与他人合作？

自己的part在编写结束后，需要与队友的程序进行整合，这就需要我们线下进行有效的交流和沟通，通过线下的交流和代码的分析测试，我们呈现出了目前的项目。