# 单双人游戏与伪AI贪吃蛇项目开发说明书

## 项目简介

1.项目背景（选择这个项目的原因、前景，面向的用户，优势）；

随着科技的发展, 人们的生活质量有了飞跃性的提升, 与此同时, 快节奏、强压力、高强度的生活节奏也席卷而来, 人们的烦躁情绪越来越严重, 因此, 我们决定开发一款面向全年龄段的轻松解压小游戏, 不管你是学生, 工作人, 还是老年人, 这款游戏都能给你带来乐趣, 能帮助用户在繁忙的工作学习之余减轻他们的压力, 体会到慢下来的生活.

且编写这个程序也是为了加深我们对面向对象程序设计的思想, 对类与对象的理解, 并将其运用到具体项目中.

1. 架构说明

前端: C# winform

后端: C#

数据库: 无

1. 整体功能说明

贪吃蛇游戏是一个经典小游戏，一条蛇在密闭的围墙内，在围墙内随机出, 现一个食物，通过按键盘上的四个光标键控制蛇向上下左右四个方向移动，蛇头撞到食物，则表示食物被蛇吃掉，时蛇的身体长一节，同时计1分，接着又出现食物，等待被蛇吃掉，如果蛇在移动过程中，撞到墙壁或身体交叉蛇头撞到身体则表示游戏结束. 双人模式下, 如碰撞到对方身体也表示游戏结束.

在此基础上加入双人游戏, 通过抢夺唯一的食物进行等级升级(等级越高速度越开, 身体越长), 增加竞技性

同时加入了人工智障模式, 提供解压观赏模式, 观赏人工自动寻路吃食物, 感受人工智障的魅力QAQ

## 团队介绍

**成员分工**

代码: 白永梁: 人工智障逻辑, 单双人游戏逻辑

李柯迪: 单双人游戏逻辑, 等级机制, 界面设计

陈豪: 颜色配色设计, 代码测试调试

PPT: 江欣蔚

文档: 解叶瀛

其他: 郑叶能, 陈逸昊, 章璐杳

## 技术说明

系统：Windows 10

环境：.Net Framework 4.7

平台：VS 2019

语言: C#

优劣势：

优势：我们的优势基于游戏经典简单的基础上，也就是代码的直观性较强，但又蕴含一些独特算法（双人模式的规则），既继承了游戏的经典，又适当对游戏进行了创新。

劣势：由于知识内容接受过少的原因，我们不能使用更高一级的算法实现功能，离完善程序还是有一些距离。

## 功能模块

## 单人游戏和双人游戏模块(两者差不多, 双人游戏更加复杂一些)

介绍:

基本逻辑:

1. 游戏主体

格子用Label类的二维数组表示, 分别用颜色来区别蛇的身体头部食物地图

1. 刷新

主要控件: timer用来每次的地图的更新, 主要是蛇的Move和时间的刷新

1. 控制的移动

单人游戏下, 就只用记录WASD四个方向, 双人模式下, 需要用到WASD与JKLI分别控制Player1与Player2的移动

1. 最高记录

这块用创建的二进制文件(.dat)来记录, 游戏开始的时候读入二进制文件, 游戏结束的时候, 写进二进制文件

1. 吃食物

使用的主要的类就是随机类 Random 然后通过设置位置出的label颜色来标为食物

在每次的timer中来对食物是否与蛇头坐标重合来做下一步操作

1. 输入(字符串队列优化输入)

这步如果裸写输入会造成如果过快连续输入操作符,会因为timer的interval值过小而导致部分的输入被忽略, 所以改为字符串模拟队列来写, 即每次键盘输入的时候都把操作符插在队尾, 然后每次都从队头取操作符, 保证每个操作符都能被执行.

1. 等级机制
2. 具体流程

图示

描述已自动生成

## 2. 人工智障贪吃蛇(对画图什么的都进行了改进)

1. 画图的方式的改变:

Pen类主要用来绘制线的宽度样式

Color对颜色的定义

SolidBrush 用于填充

界面用的pictureBox

2. 更多的面对对象思维的融入:

点用class写, 这样判断时候override一下判等函数即可

不单单写数组, 而是使用更多的List<T>来写

3. 因为是人工智障, 所以算法是重中之重(遗憾是没能用哈密顿回路来完成)

基本思路:

主要实现分了两个部分，一是蛇的可以怎么走, 二是能不能这么走

所以这里我用了BFS

策略：

a.蛇如果吃了苹果后能找到自己的尾巴，才去吃这个苹果。(这是在开始的时候裸BFS会导致蛇不管身体而直接去强行走最短路, 会把自己围死)

否则在头部找一个方向，这个方向满足

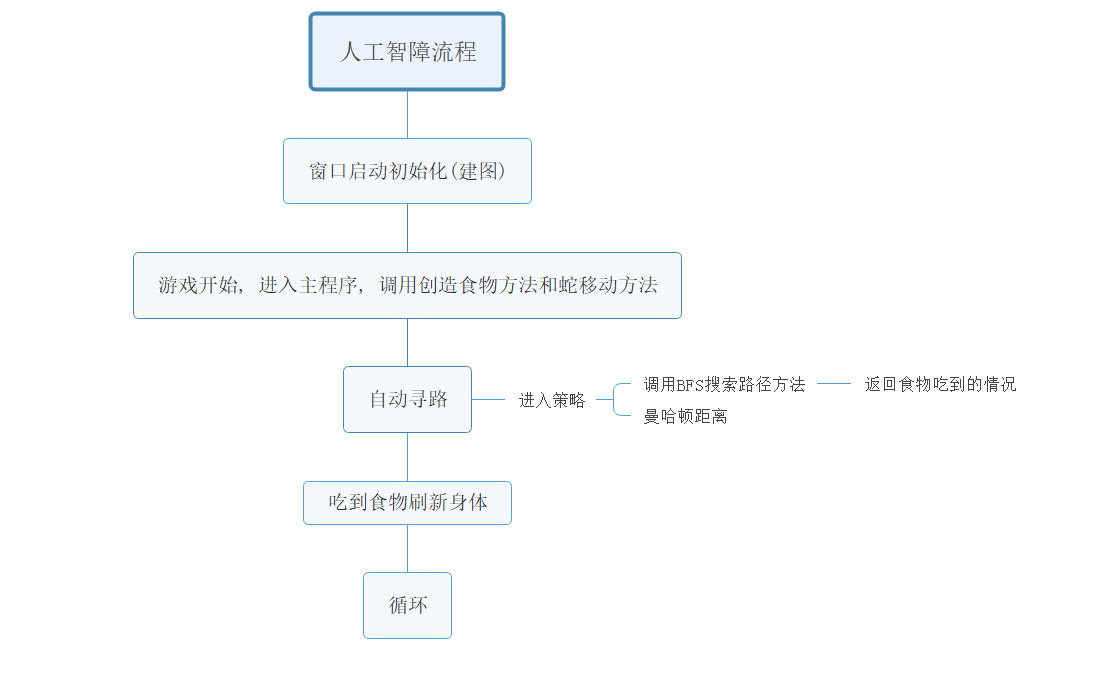
1. 离苹果最远距离（用的曼哈顿距离, 因为矩形距离, 话说曼哈顿距离 思路来源:电影唐人街探案）。2.走完后能找到尾巴。

b.如果吃了苹果后，生成的新苹果寻路算法无法找到这个新苹果。那么就在头部找一个方向，这个方向满足

1. 离苹果最远。2.走完后能找到尾巴。

过程：

规划了一下(吸取教训)，先用一个list存蛇的身体的坐标，行走求解都单独完成。输出显示单独用一个method，先把蛇坐标装进邻接矩阵，然后输出邻接矩阵。



## 使用说明

系统：Windows 10

环境：.Net Framework 4.7

平台：VS 2019