**《Python程序设计》**

**课程设计报告**

　　　　　　　　　　（2018 —2019 学年第 一 学期）

题 目 贪吃蛇小游戏

学生姓名 樊 璟 琳

专业班级 信管1602班

学生学号 311609030209

教师姓名 徐 文 鹏

成 绩**：**

评 语：

教师签名：

日期：

**目录**

**1 设计目的3**

1.1 课程设计目的3

1.2 本课程设计具体目的3

**2 设计任务和要求3**

2.1 课程设计教学任务和要求3

2.2 本课程设计具体任务和要求4

**3 总体设计5**

3.1 游戏设计任务分配5

3.2 游戏设计功能图5

3.3 游戏设计流程图6

**4 设计实现7**

4.1 最终实现结果7

4.2 实现结果评价8

**5 详细设计8**

5.1 整体简介8

5.2 个人完成部分函数分析9

**6 调试与测试13**

**7 设计总结17**

7.1 尚未解决的问题17

7.2 课程设计体会17

附录**19**

**1 设计目的**

* 1. 课程设计教学目的

本课程设计是本专业的一门重要实践性教学环节。在学习了专业基础课和《Python程序设计》课程的基础上，本课程设计旨在加深对Python程序设计的认识，对Python语言及其语言生态有一个进一步的掌握和应用，学会运用Python标准库及外接相关库来解决实际问题的基本能力，培养和提高学生分析问题、解决问题的能力，尤其是提高学生使用Python为开发语言来进行问题描述、交流与思考的能力，为毕业设计和以后的工程实践打下良好的基础。

* 1. 课程设计具体目的

贪吃蛇游戏是一款经典的小游戏，深受大众喜爱。当我们学习了python程序设计课程后，了解到使用python第三方库pygame可以实现这款游戏，但经查阅资料，我们发现使用python内置的tkinter库，采用面向对象的方法也可以实现这款游戏。在此基础上，我们小组讨论决定做贪吃蛇小游戏，使用面向对象方法来实现这款游戏，以此来完成我们的课程设计。鉴于python面向对象我们没有学习，希望挑战一下自己，在团队合作的基础上尝试设计出这款小游戏。一来挑战自己、接受新知；二来自己体验自己做出的游戏，会有一丝小小的成就感；最后就是希望顺利完成这次课程设计。

**2 设计任务和要求**

2.1 课程设计教学任务和要求

本课程设计主要任务是以Python为开发语言完成一个100~300行左右规模的程序项目开发。设计参考题目见附件1。

课程设计的基本要求是：在课程设计的各个阶段严格、规范地完成相关的文档，例如在初期按时完成设计目的、设计要求、总体设计，在后期完成详细设计、调试与测试、设计总结等。要求能完成所选题目的主要功能，程序运行健壮、正确，代码要有详细注释，可读性好；所写文档结构合理、内容完整、叙述清晰。更高要求是：有创意、系统界面美观。

由于课程设计项目具有一定的综合性，鼓励具有不同特长和不同能力的学生互相组队。项目小组自己推荐一名组长，实行“组长负责制”。组长组织组员进行项目选题、任务分配、方案确定、方案设计、系统调试测试，组员分工协作。小组成员开展项目讨论，互相支持，形成协作意识。

2.2 本课程设计具体任务和要求

使用python面向对象方法，结合tkinter库、random库和tkinter.messagebox弹窗库，实现贪吃蛇小游戏。具体的功能设计任务和游戏规则要求如下：

（1）功能设计任务

玩家运行应用程序，进入游戏界面，游戏直接开始。游戏界面由四部分组成，包括得分栏（Score:分值）、游戏操作框、小蛇、食物。初始化分数为0，小蛇的位置、颜色和大小初始化，食物的位置随机出现，初始颜色固定。之后，玩家通过控制方向键或鼠标使贪吃小蛇在屏幕上不停的移动，寻找食物。小蛇的移动速度正常情况下始终保持不变，通过连续点击前进键（小蛇当前前进方向）使小蛇行进加速。在小蛇移动的过程中，每当吃到食物后，游戏界面会出现新食物，食物分为3种类型：绿色代表普通食物，吃了得分并增加小蛇的长度；红色为特殊食物1，吃了得分并减短小蛇的长度；金色为特殊食物2，吃了得分并回到小蛇的最初长度状态；特殊食物3为变色食物，小蛇吃了变色食物会根据食物颜色随机改变自身的颜色。小蛇吃的食物越多，得分就越高。如果小蛇在移动的过程中碰到了自身（蛇不可以向自己反方向走），则游戏结束，弹出弹窗，显示得分，玩家点击确认，可直接进入下一局游戏。每次游戏结束，进入新一局游戏，本局游戏已被初始化。

（2）游戏规则要求

点击运行进入游戏界面，显示得分栏、小蛇、食物。

食物随机出现、小蛇根据初始化位置和方向移动。

玩家点击方向键，小蛇根据玩家的按下的方向键进行移动。

当小蛇头部触及食物时，得分框数据更新，分数加1，小蛇吞掉食物，下一个食物随机出现。

当小蛇吃到绿色食物（普通食物）时，小蛇长度增加1个格；当蛇吃到红色食物（特殊食物1）时，长度减少3个格；当小蛇吃到黄色食物时（特殊食物2），变回蛇的初始长度；当蛇吃到变色食物（特殊食物3）时，返回碰到食物时食物的颜色。

当蛇头触碰到自己身体时游戏结束，弹出弹窗显示得分，点击“确定”，初始化游戏，游戏重新开始。

**3 总体设计**

3.1 游戏设计任务分配

郭盼盼：总体功能方案分析设计、游戏整体界面设计实现（包括贪吃蛇移动界面，小蛇得分栏、小蛇和食物的显示）、后期整合完善。

樊璟琳：总体功能方案分析设计、游戏规则功能的实现（实现小蛇吃到不同颜色食物后的反应）、后期整合完善。

方留龙：总体功能方案分析设计、游戏结束功能的实现（游戏结束规则功能的实现）、后期整合完善。

3.2 游戏设计功能图



图3-1 贪吃蛇游戏功能图

3.3 游戏设计流程图

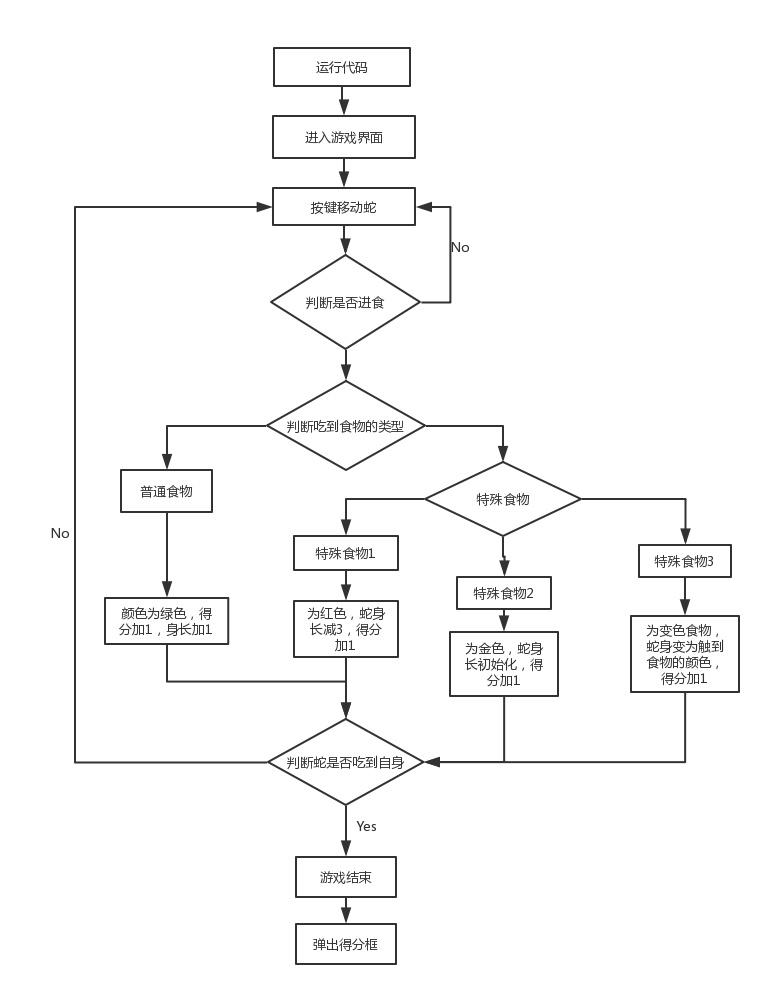


图3-2 贪吃蛇游戏流程图

**4 设计实现**

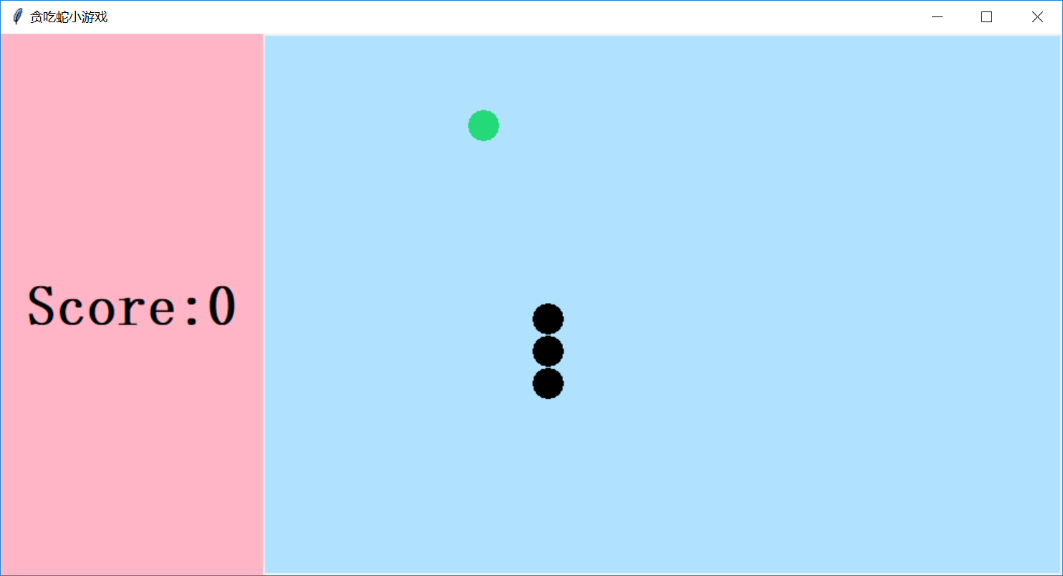
4.1 最终实现结果

图4-1 玩家首次进入游戏界面

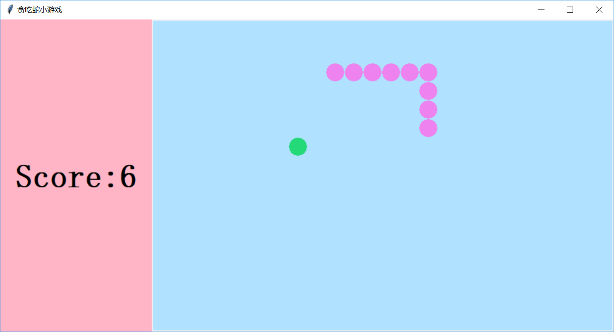
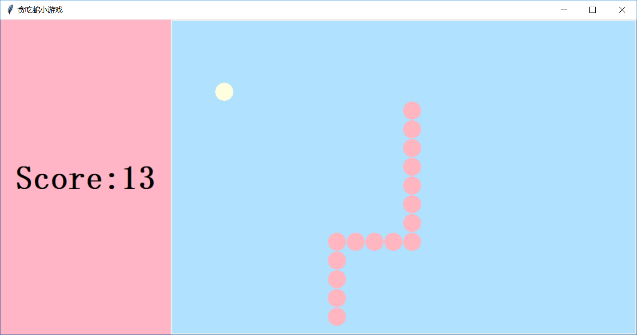
 

图4-2 小蛇吃到变色食物，身体变色图

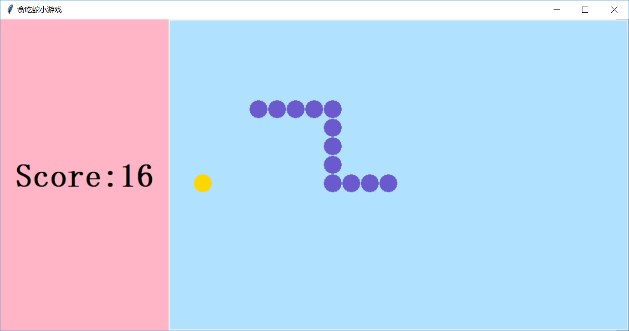
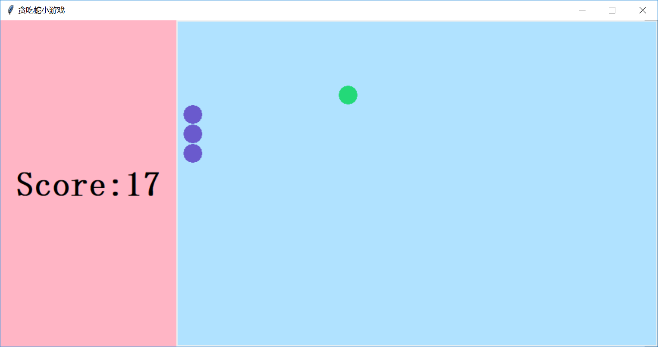
 

图4-3 小蛇吃到金色食物，体长被初始化图

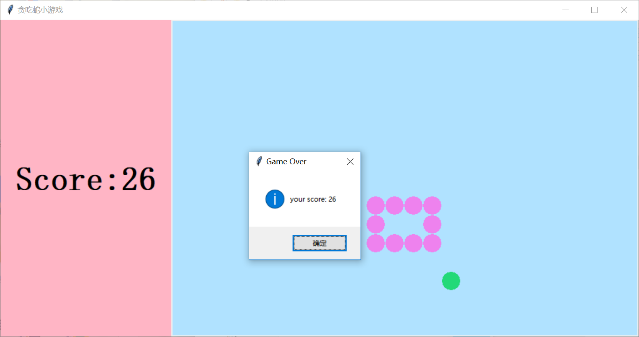
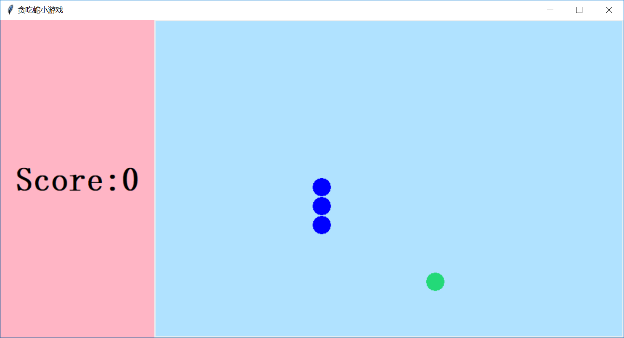
 

图4-4 游戏结束，点击“确定”进入新一轮图

4.2 实现结果评价

程序实现了最初的设计想法和要求，游戏整体界面比较清新美观，功能没有很高大上，但基本功能都已实现，游戏操作界面功能完备，尤其在食物方面，分出了普通食物和特殊食物，小蛇吃到不同事物进行不同反应，这使得游戏更有趣味性。小蛇碰到自身，游戏结束，显示得分，点击“确定”，进行下一轮游戏。游戏开始，游戏运行，游戏结束后开始新一轮游戏功能都已实现。

总体来说，我们认为我们小组的python课程设计“贪吃蛇小游戏”实现了最初的设计想法和要求，可以达到“良好”的成绩，希望也可以得到老师的认可。

**5 详细设计**

5.1 整体简介

（1）整体代码所使用的库

Thinter库：Tkinter 是 Python 的标准 图形界面开发（GUI） 库

Random库：随机数（random）库

Tkinter.messagebox库：弹窗库，这里的`messagebox`就是我们平时看到的弹窗。

（2）定义的类

Grid：定义一个网格（Gird）类，用于设置游戏总界面，蛇轨迹绘画等

Food：定义一个食物类，用于设置食物的各种参数

Snake：定义一个蛇类，用于定义蛇的各种参数

SnakeGame：定义一个SnakeGame类，Frame框架，用于游戏规则的设定和进行游戏控制。

（3）类中定义的方法

Grid类：

a.\_\_init\_\_()：采用\_\_init\_\_()构造方法初始化Gird类的宽和高、背景颜色

b.draw()：定义一个draw函数，作用是绘制小蛇的行进路径

2、Food类：

a.\_\_init\_\_()：采用\_\_init\_\_()构造方法对食物（形状为网格）进行初始化

b.set\_pos()：定义食物出现位置的set\_pos()方法

c.display()：定义display方法显示食物和蛇

3、Snake类：

a.\_\_init\_\_()：采用\_\_init\_\_()构造方法对蛇进行初始化

b.initial()：定义initial()方法用于游戏重新开始时初始化贪吃蛇的位置

c.move()：定义move()方法使蛇发生移动

d.add()：定义一个add()方法， 蛇吃到了普通食物，用于增加蛇的长度

e.cut\_down()：定义一个cut\_down()方法，蛇吃到了特殊食物1，用于减短蛇的长度

f.init()：定义init()方法，蛇吃到了特殊食物2，返回最初的长度

g.change\_color()：定义change\_color()方法，蛇吃到了特殊食物3，用于改变蛇的颜色

4、SnakeGame类：

a.\_\_init\_\_()：采用\_\_init\_\_()构造方法对SnakeGame进行初始化

b.initial()：定义一个initial()方法，用于游戏重新开始进行游戏

c.display\_food()：定义display\_food()，用于显示食物的颜色、位置和类型

d.key\_release()：定义一个key\_release()方法，用于监听键盘事件

e.run()：定义一个run()方法判断游戏进程

f.move()：定义move()方法，用与判断蛇的移动方向

5.2 个人完成部分函数分析

我主要负责设置食物那部分，就是令小蛇吃到的食物随机出现，且出现不同类型的食物，分别是普通食物，为绿色，吃到增加体长；特殊食物1，为红色，吃到蛇体长减3；特殊食物2，为金色，吃到蛇体长初始化；特殊食物3，为变色食物，吃到蛇身变为吃到食物时食物的颜色，通过使用random库来实现。我完成了Sanke()类里面的move()方法、add()方法、cut\_down()方法、init()方法和change\_color()方法，还有就是SnakeGame()类里面的initial()方法和display\_food()方法。下面我对自己负责部分的方法进行详细说明。

表5-1 食物与功能表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实体 食物 | 普通食物  （type1） | 特殊食物1  （type2） | 特殊食物2  （type3） | 特殊食物3  （type4） |
| 小蛇状态 | 身长+1 | 身长-3 | 身长初始化 | 变色 |
| 食物颜色 | 绿色 | 红色 | 金色 | 变色 |
| 得分 | +1 | +1 | +1 | +1 |

（一）输入输出分析

我这个出现不同食物部分没有用户输入，就是通过random库里面的randin来出现不同的整数，并将不同随机数数赋给一个食物类型，即出现不同随机数就相当于出现不同食物，最终出现的效果就是游戏界面随小蛇吃到不同食物随机出现新的不同的食物。

（二）方法详述

（1）move()这个方法就是令蛇向一个指定点移动，一旦移动之后蛇的位置变化，蛇尾减短一个格子。

def move(self, new): #定义move()方法使蛇发生移动

self.body.insert(0, new)

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

self.grid.draw(new, self.color)

（2）add()方法是针对普通食物，即type1绿色食物。如果出现这个食物，小蛇吃到后产生的反应，即体长加1，得分加1。

def add(self ,new): #定义一个add()方法,用于增加蛇的长度

self.body.insert(0, new) #更新后蛇的长度

self.grid.draw(new, self.color) #显示更新后的蛇

（3）cut\_down()方法是针对特殊食物1，即type2红色食物。如果出现这个食物，小蛇吃到后产生的反应，即小蛇吃到该类食物后体长减3（通过for循环来实现），得分加1。

def cut\_down(self,new): #定义一个cut\_down()方法，用于减短蛇的长度

self.body.insert(0, new) #更新后蛇的长度

self.grid.draw(new, self.color) #显示更新后的蛇

for i in range(0,3): #遍历for循环，将蛇的长度减少3个格

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

（4）init()法是针对特殊食物2，即type3金色食物。如果出现这个食物，小蛇吃到后产生的反应，即小蛇吃到该类食物后体长回到最初的长度（通过while循环来实现），得分加1。

def init(self, new): #定义init()方法用于返回蛇最初的长度

self.body.insert(0, new)

self.grid.draw(new, self.color)

while len(self.body) > 3: #判断蛇的长度，如果大于3个格，则用while循环返回蛇的初始长度

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

（5）change\_color()方法是针对特殊食物3，即type4变色食物。如果出现这个食物，小蛇吃到后产生的反应，即小蛇吃到该类食物后蛇身变色，变色范围限制在self.color\_c = ("#FFB6C1","#6A5ACD","#0000FF","#F0FFF0","#FFFFE0","#F0F8FF","#EE82EE","#000000","#5FA8D9","#32CD32")，得分加1。

def change\_color(self, new, color): #定义change\_color()方法，改变蛇的颜色

self.color = color #蛇改变为特殊食物的颜色

self.body.insert(0, new) #显示更新后的蛇

for item in self.body: #遍历蛇的身体，全部改变为改变之后的颜色

self.grid.draw(item, self.color)

（6）display\_food()方法用到了random库，通过使用randint随机出现整数，给不同类型的食物在不同范围内赋不同的值，使其出现的概率不同，这样主函数通过调用这个方法来显示不同食物。

def display\_food(self): #定义display\_food()，用于显示食物的颜色、位置和类型

self.food.color = "#23D978" #初始化食物的颜色为绿色

self.food.type = 1 #初始化食物的类型为第一个类型

if randint(0, 20) == 5: #判断出现的（0,20）范围内出现随机数是否为5，对随机数为5的数字设置其食物颜色和类型

self.food.color = "#FFD700"

self.food.type = 3

while (self.food.pos in self.snake.body): #判断蛇是否吃到食物，若吃到食物，该类食物随机出现

self.food.set\_pos()

self.food.display()

elif randint(0, 4) == 2: #判断出现的（0,4）范围内出现随机数是否为2，对随机数为2的数字设置其食物颜色和类型

self.food.color = "#EE82EE"

self.food.type = 4

while (self.food.pos in self.snake.body): #判断蛇是否吃到食物，若吃到食物，该类食物随机出现

self.food.set\_pos()

self.food.display()

elif len(self.snake.body) > 10 and randint(0, 16) == 1: #当蛇的长度>10个格子，且随机数为（0,16）设置其食物颜色和类型

self.food.color = "#FF0000"

self.food.type = 2

while (self.food.pos in self.snake.body): #判断蛇是否吃到食物，若吃到食物，该类食物随机出现

self.food.set\_pos()

self.food.display()

else: #若以上均不满足，当蛇吃到食物时，新食物随机出现

while (self.food.pos in self.snake.body):

self.food.set\_pos()

self.food.display()

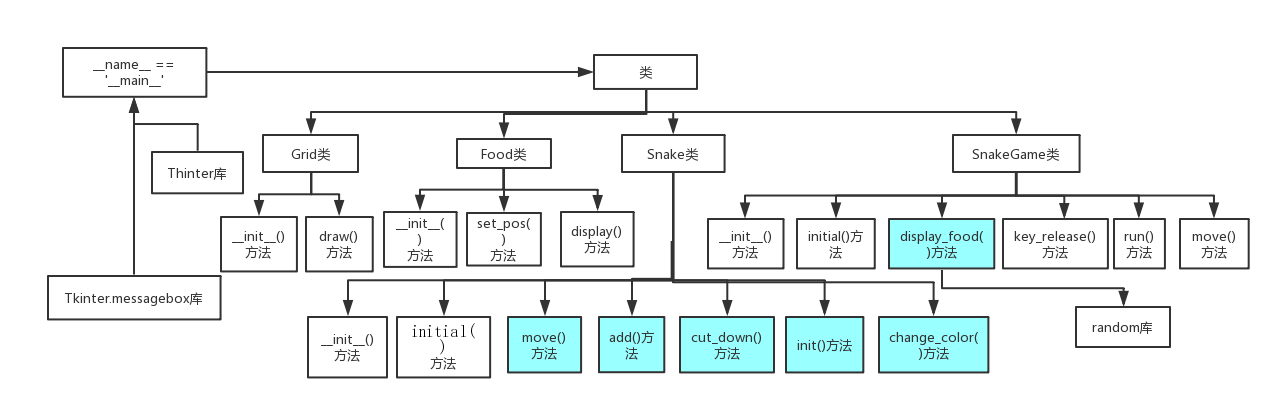
（三）程序流程图（浅蓝色标注部分为我完成的部分）

图5-1 贪吃蛇小游戏程序流程图

**6 调试与测试**

我的调试过程是在我们三个人都完工以后把代码整合到一起后完成的，主要调试的是看我想实现的四种食物是否随机出现，颜色是否不同，会有变化，以及小蛇吃到食物后的反应和当初的要求是否达到一致。刚开始会出现小蛇长度不超过蛇而出现特殊食物1，后来发现是随机数的问题，我又修改了随机数，才解决这个问题。由于利用随机数设定食物相对容易，而我的这部分主要就是食物一旦效果达成，之后就没再出现什么其它的问题了。程序具体运行效果如下：

（1）初始进入根据的初始化，显示蛇的颜色为黑色，长度为3，食物颜色为绿色，位置是随机的。

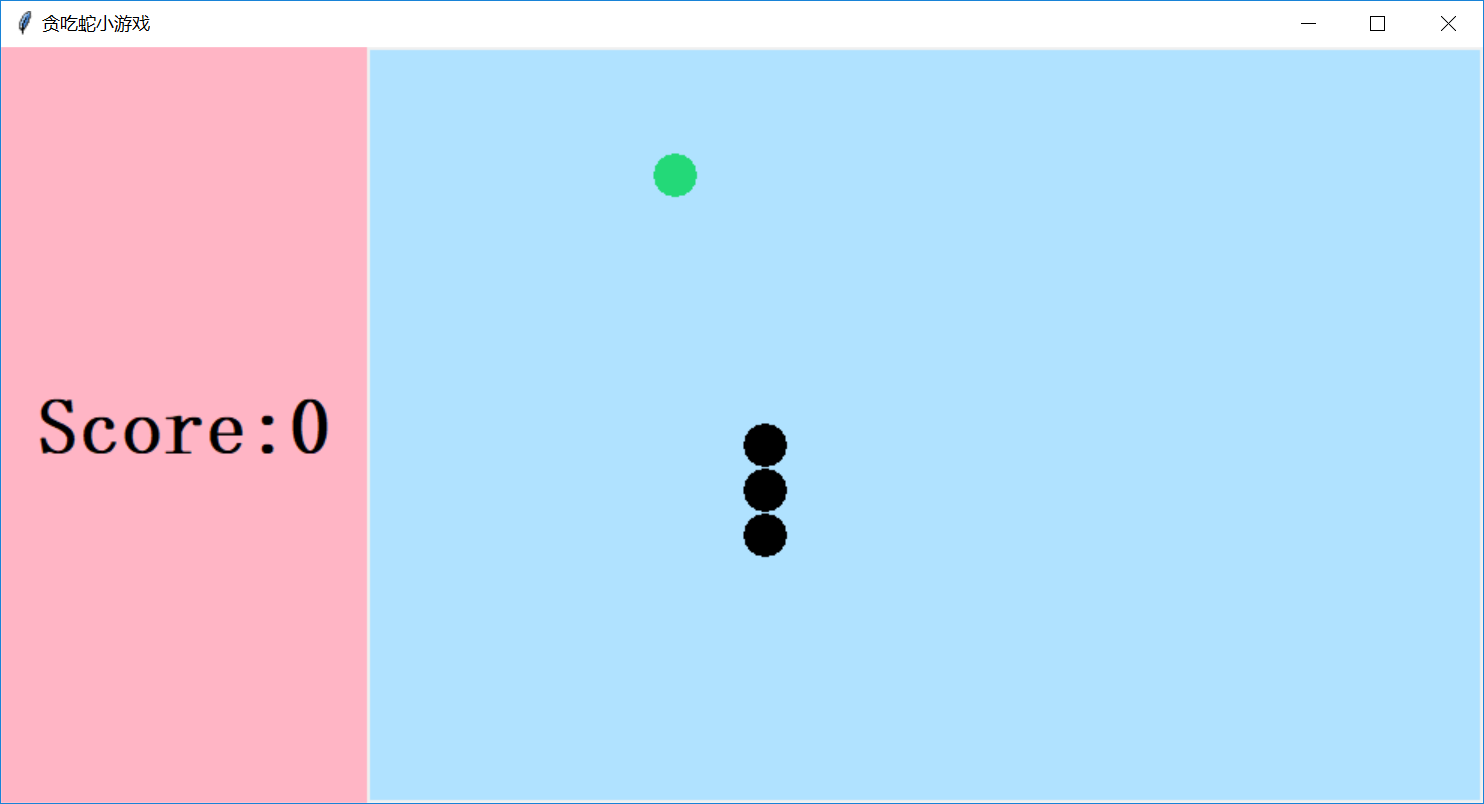


图6-1 初始游戏界面显示图

（2）由下面一张截图可以看出来小蛇吃到普通食物后身长加1，下一个食物位置随机出现。

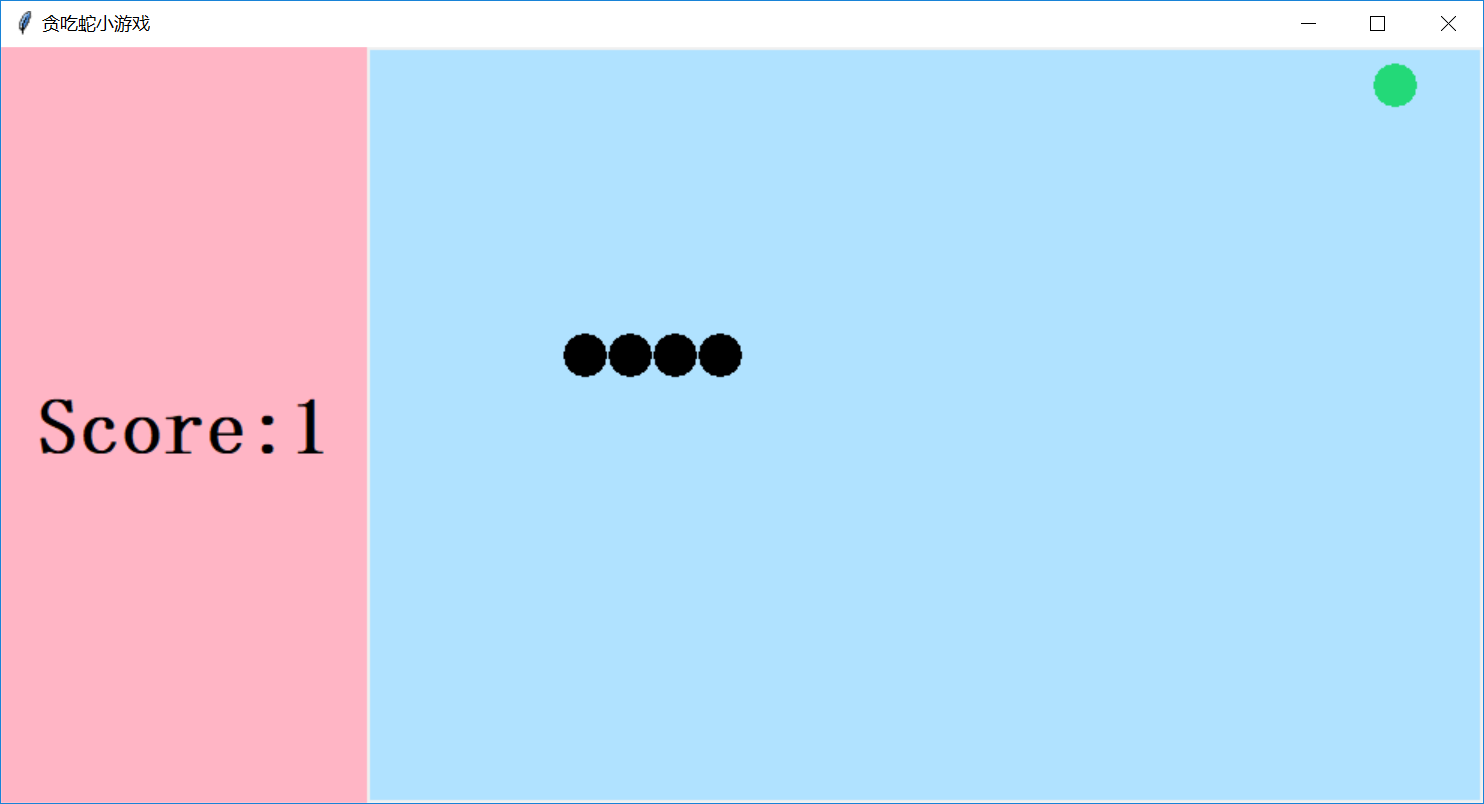


图6-2 小蛇吃到第一类食物后的反应图

（3）由于食物在我给定的几个颜色里面变化，截图只能展示静态效果，下面两张截图展示小蛇吃特殊食物3，即变色食物的运行情况。由图可知，变色食物的效果已经实现。

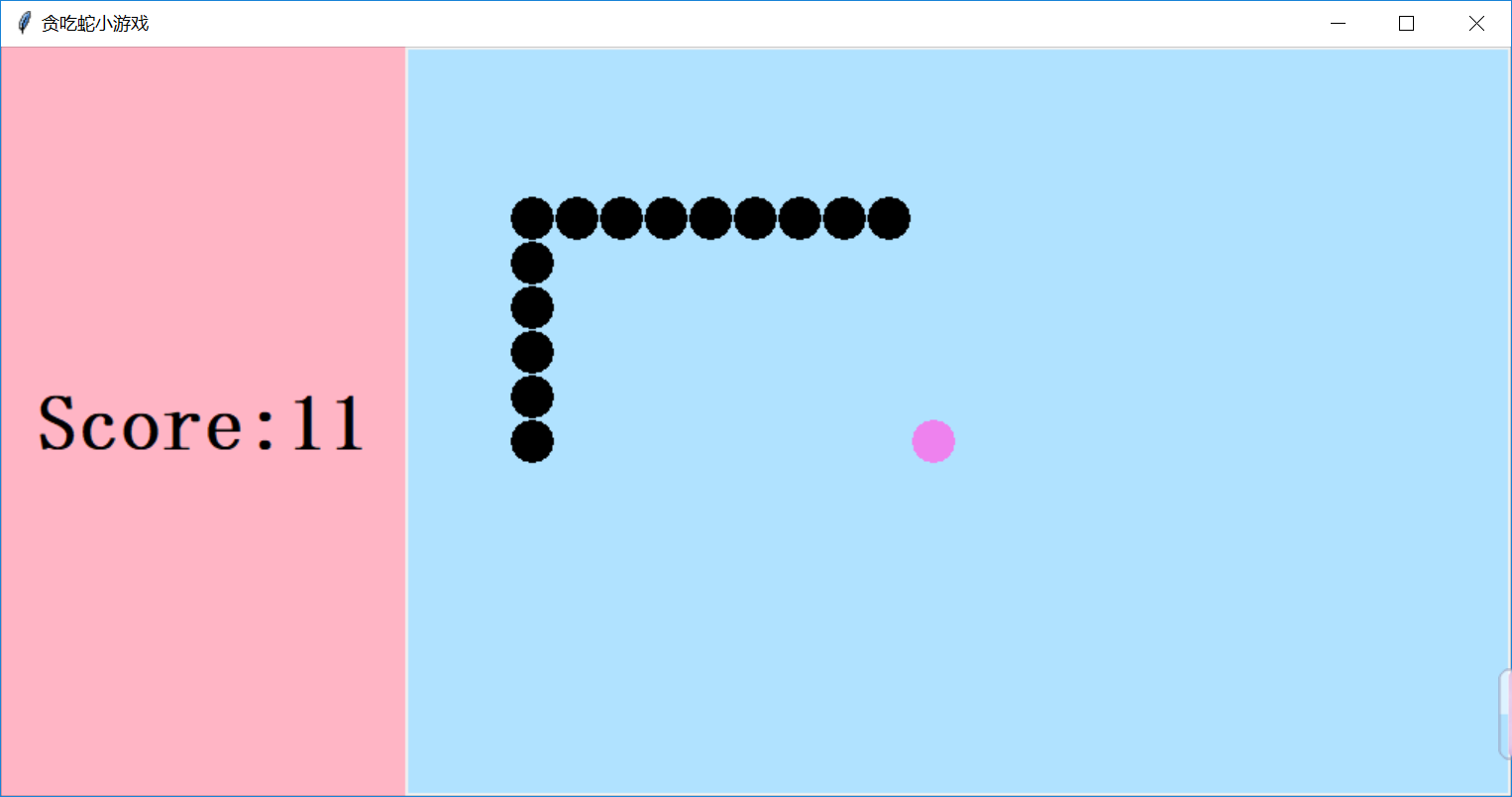


图6-3 小蛇吃变色食物前食物和小蛇的状态图

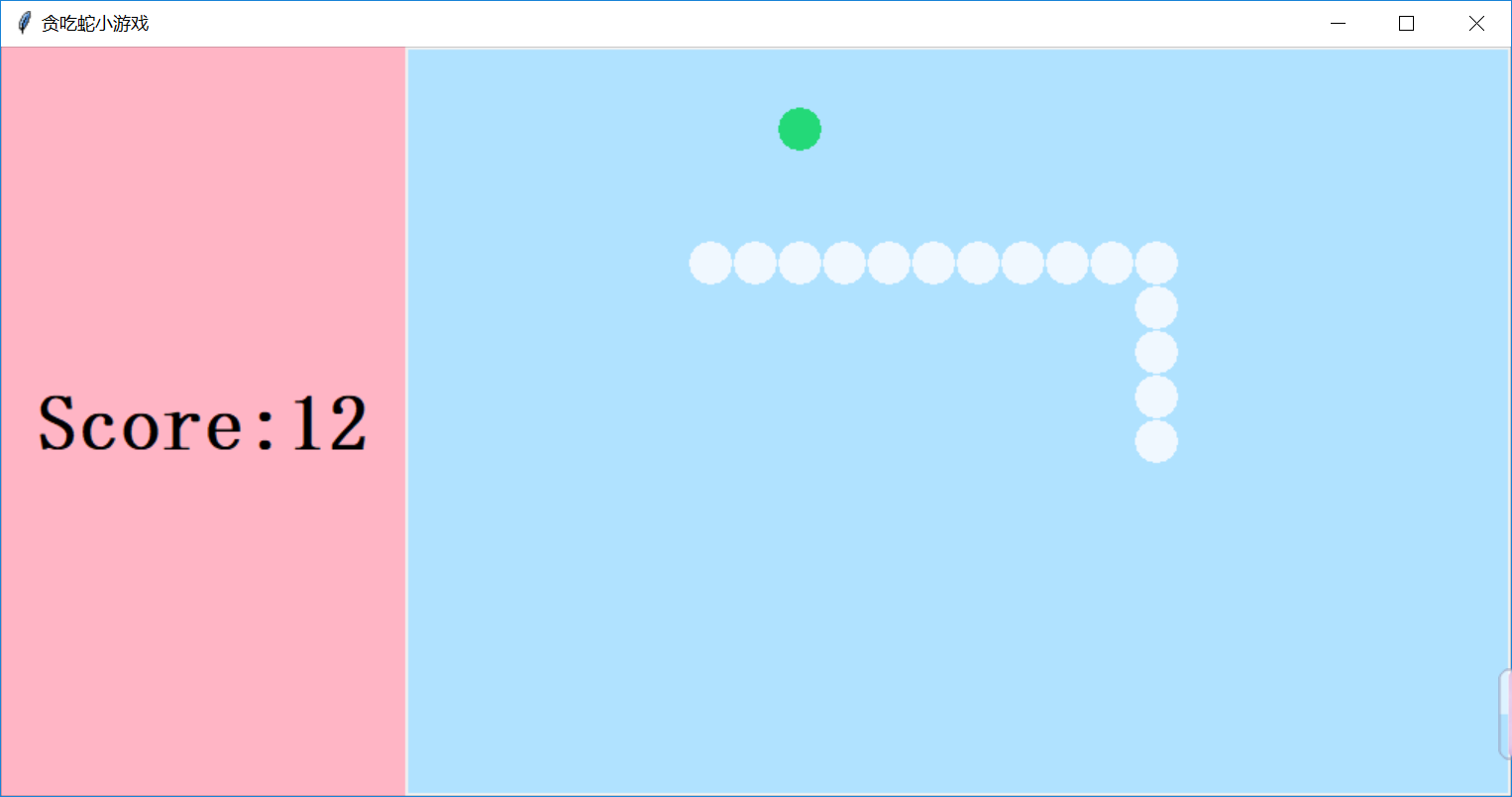


图6-4 小蛇吃变色食物后食物和小蛇的状态图

（4）下面展示小蛇遇到特殊食物2的情况，由以下两张截图可以看出，小蛇吃到这个食物后体长回到最初体长。对于特殊食物2的功能要求也已实现。

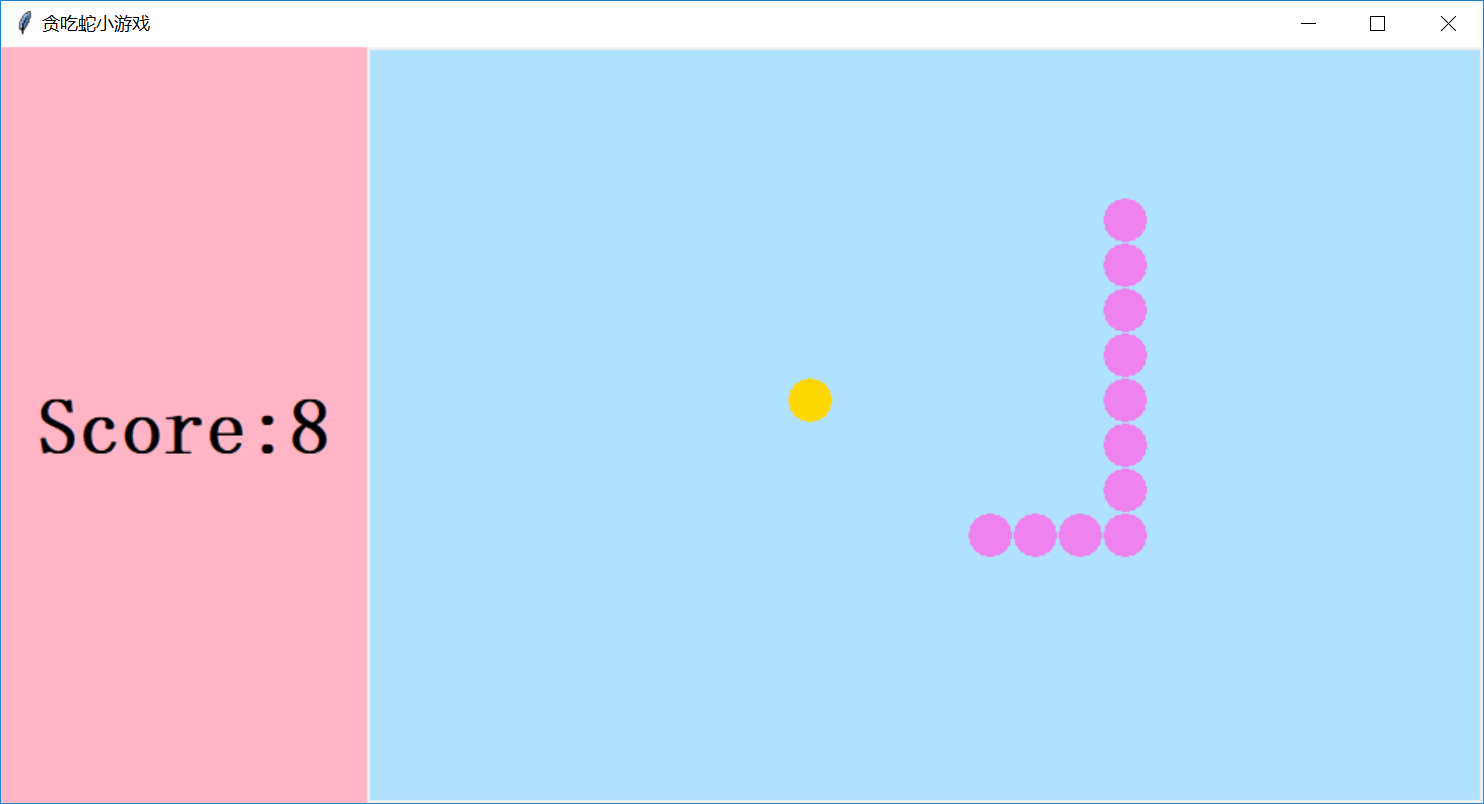


图6-5 小蛇吃特殊食物2前的状态

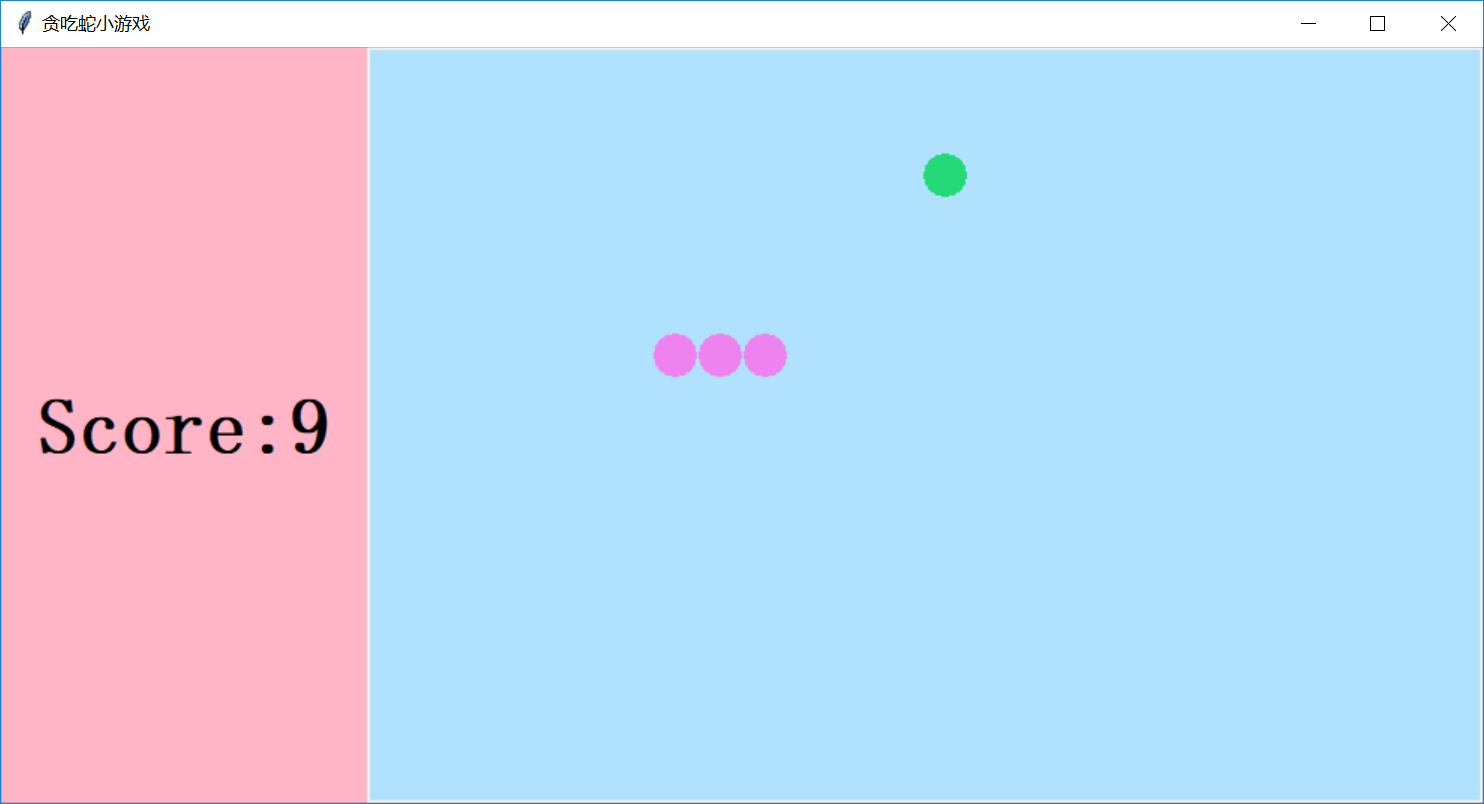


图6-6 小蛇吃特殊食物2后的状态

（5）下面展示小蛇吃到特殊食物1的情况，由以下两张截图可以发现，小蛇吃到食物前体长为12，吃时变为13，吃后变为10。符合当初的想法，对特殊食物1的功能要求实现。

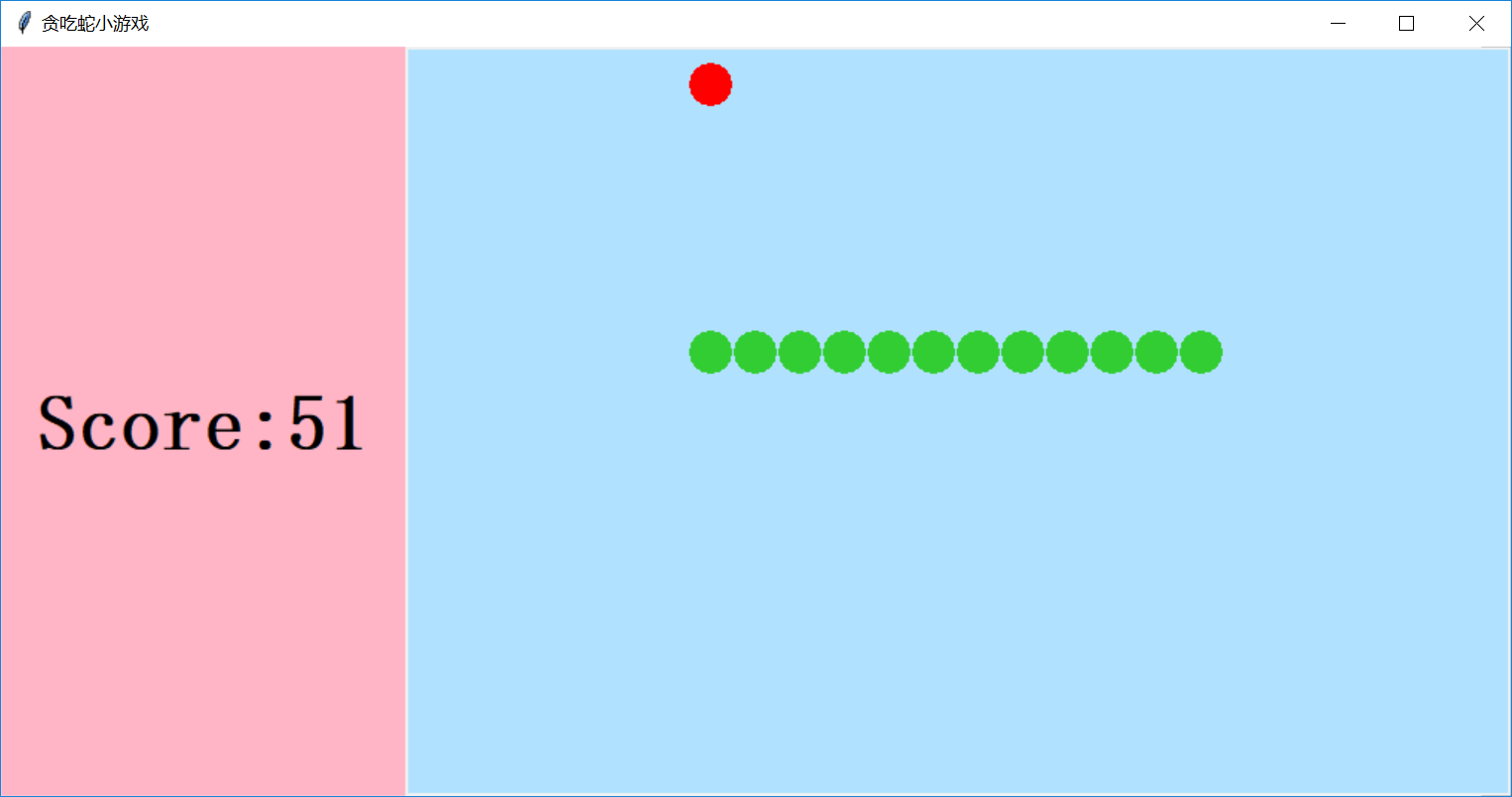


图6-7 小蛇吃特殊食物1前的状态

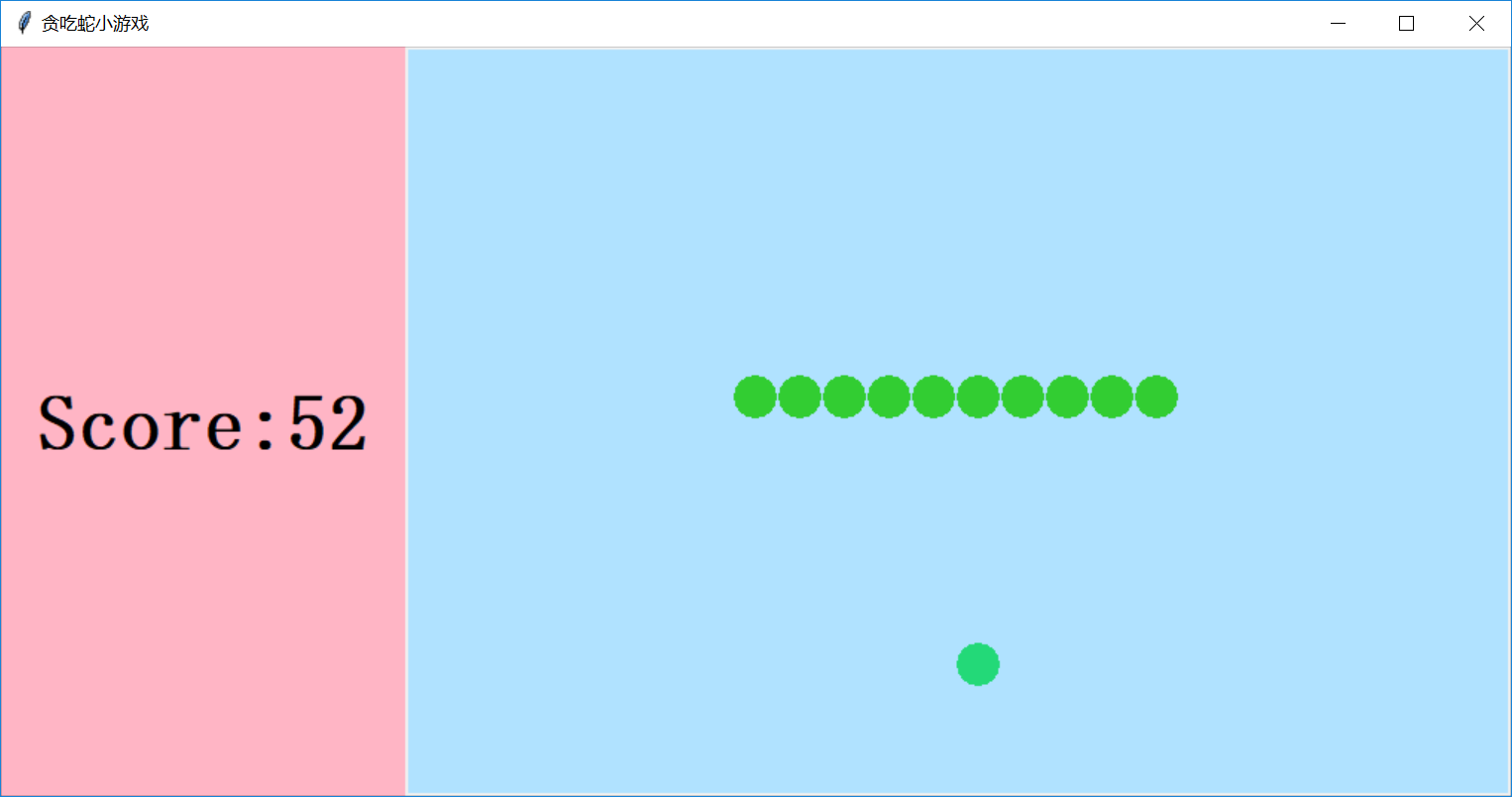


图6-8 小蛇吃特殊食物1后的状态

**7 设计总结**

7.1 尚未解决的问题

在做的过程中，很多东西并非一下完成，出错是难免，我们在进行的过程中出现了一些问题，有一些问题被解决，还有一些问题是尝试过很多次还是没有解决掉的。甚至有些问题可能是根本解决不掉的。我没解决的问题主要有一下几个。

（1）第一个就是因为食物随机出现，如第一次游戏结束，下一次游戏开始食物还是上一次结束时的没被吃到的食物，对于其它食物来说可能是没有问题的，但是对于特殊食物1，即红色食物，吃了自身长度减短3，这个是有一些不合理的，因为当时在设计这种类型的时候就规定蛇长大于10才会出现这种食物，吃了蛇长减3， 但是这种情况出现就使得吃掉后蛇长变为1，而且不满足特殊食物3出现的第一个条件。

（2）第二个问题是想在小蛇吃食物时加入声音，通过百度、查资料，尝试了很多种方法都没有解决这个问题，后来就放弃了加入声音。

（3）第三个问题是想改变弹窗的大小，但是后来百度用root.geometry()方法改变不了弹窗的大小，而是直接改变的界面的大小。

（4）还有问题就是通过展示我们发现有几组同学也做了贪吃蛇小游戏，他们的功能和我们的有所不同，还有进入页面和退出页面，或者通过点击不同键改变速度，吃到不同事物改变方向等，我们当时没有想到这几个方面，感觉以后可以多多交流，扩充思路。

总体说来，我们的问题就是这些了，有优点，也有缺陷，还是要继续学习，不断进步。

7.2 课程设计体会

通过这次课程设计，真的是锻练了自己读代码，写代码，改代码的能力。也学习了一些面向对象的知识。加深了自己对python的进一步理解。通过自己和他人合作，完成一项任务，真的是感觉到一丝满足感，尤其是自己做出来小时候经常玩的游戏，心里会有些许自豪。虽然还有很多缺陷，但是鉴于自己的能力问题，我还是有一丝满足。今后还会继续能力，学习，学无止境，只有当自己能力够强的时候才能做出来让别人欣赏的东西。同时面临当前的时代，在前行的道路上，要学着去接受新事物、学着合作学习、共同进步。其实感觉很多时候我们所学的知识并不是来源于课本，而是来源于身边人的影响，自己的想法都是很受限的，老师通过各小组展示我就深有体会，看到每个小组都有自己的特色，都有自己的创意，尤其是看到和我们做一样题目的同学，他们比我们做的好的地方，我们优于他们的地方，感受就更深了。所以，在今后前行的路途中，当自己面对困难的时候，应该多多和他人交流，寻求解决办法。还有一个感悟就是，不要过度怀疑自己，当你努力了，就应该相信自己能行。最后，感谢小组人员的齐心协力和辛苦付出，感谢老师付出时间和经历，陪着我们一起进步。

**附录：完整代码**

from tkinter import \* #导入tkinter库，是python的内置库，导入即可使用

from random import randint #导入random模块中的randint函数，用于生成随机整数

import tkinter.messagebox #导入弹窗库，用于信息提示

class Grid(object): #定义一个网格（Gird）类，用于设置游戏总界面，蛇轨迹绘画等

#初始化函数，\_\_init\_\_()方法,被称为类的构造函数或初始化方法，当创建了类的实例时就会调用该方法

def \_\_init\_\_(self, master=None,height=16, width=24, offset=10, grid\_width=30, grid\_height=30,bg="#B0E2FF"): #采用\_\_init\_\_()构造方法初始化Gird类的宽和高、背景颜色(浅蓝)

self.height = height #把参数height赋值给对象变量height，值为16，用于设置canvas纵坐标中网格的个数

self.width = width #把参数width 赋值给对象变量width ，值为24，用于设置canvas横坐标中网格的个数

self.offset = offset #把参数offset 赋值给对象变量offset，为允许发生的偏移量

self.grid\_width = grid\_width #把参数grid\_width赋值给对象变量grid\_width，单个网格的宽度为50

self.grid\_height=grid\_height #把参数grid\_height赋值给对象变量grid\_height，单个网格的高度为50

self.bg = bg #把参数bg赋值给对象变量bg，设置背景色

self.canvas = Canvas(master, width=self.width\*self.grid\_width+2\*self.offset, height=self.height\*self.grid\_height+ #设置画布(蛇活动范围)的大小和背景颜色

2\*self.offset, bg=self.bg)

self.canvas.pack(side=RIGHT,fill=Y) #用pack()函数对画布进行布局，side=RIGHT设置得分栏位于canvas的右侧，fill=Y填充Y方向的空间

def draw(self, pos, color): #定义一个draw函数，作用是绘制小蛇的行进路径、形状实物的形状

#pos位置坐标参数

x = pos[0] \* self.grid\_width + self.offset

y = pos[1] \* self.grid\_width + self.offset

#outline属性要与网格的背景色（self.bg）相同，更加美观

self.canvas.create\_oval(x, y, x + self.grid\_width, y + self.grid\_width, fill=color, outline=self.bg)

#绘制一个圆形((a,b,c,d),值为左上角和右下角的坐标)，路径及小蛇、实物的形状，轮廓outline属性设置为网格的背景色

class Food(object): #定义一个食物类，用于设置食物的各种参数

def \_\_init\_\_(self, grid,color="#23D978"): #采用\_\_init\_\_()构造方法定义一个食物

self.grid = grid #引用grid类为食物网格

self.color = color #食物的颜色

self.set\_pos() #食物的位置

self.type = 1 #第一次游戏开始时食物的类型

def set\_pos(self): #定义食物出现位置的set\_pos()方法

x = randint(0, self.grid.width - 1) #调用randint随机出现食物横坐标的位置

y = randint(0, self.grid.height - 1) #调用randint随机出现食物纵坐标的位置

self.pos = (x, y) #返回食物出现的位置坐标

def display(self): #定义display方法显示食物和蛇

self.grid.draw(self.pos, self.color)

class Snake(object): #定义一个蛇类，用于定义蛇的各种参数

def \_\_init\_\_(self, grid, color = "#000000"): #采用\_\_init\_\_()构造方法对蛇进行初始化

self.grid = grid #引用grid类为蛇的网格

self.color = color #蛇的初始化颜色

self.body = [(8, 11), (8, 12), (8, 13)] #蛇的初始化长度

self.direction = "Up" #定义蛇的初始行进方向向上

for i in self.body: #用for循环遍历蛇

self.grid.draw(i, self.color)

#定义initial()方法用于游戏重新开始时初始化贪吃蛇的位置

def initial(self):

while not len(self.body) == 0: #进行while循环，知道蛇的长度为0，退出while循环

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

self.body = [(8, 11), (8, 12), (8, 13)] #初始化蛇的大小

self.direction = "Up" #游戏重新开始初始化蛇的方向

self.color = "blue" #游戏重新开始初始化蛇的颜色

for i in self.body: #用for循环遍历蛇

self.grid.draw(i, self.color)

#蛇向一个指定点移动

def move(self, new): #定义move()方法使蛇发生移动

self.body.insert(0, new)

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

self.grid.draw(new, self.color)

#蛇向一个指定点移动，并增加长度

def add(self ,new): #定义一个add()方法,用于增加蛇的长度

self.body.insert(0, new) #更新后蛇的长度

self.grid.draw(new, self.color) #显示更新后的蛇

#蛇吃到了特殊食物1，减短自身的长度

def cut\_down(self,new): #定义一个cut\_down()方法，用于减短蛇的长度

self.body.insert(0, new) #更新后蛇的长度

self.grid.draw(new, self.color) #显示更新后的蛇

for i in range(0,3): #遍历for循环，将蛇的长度减少3个格

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

#蛇吃到了特殊食物2，回到最初长度

def init(self, new): #定义init()方法用于返回蛇最初的长度

self.body.insert(0, new)

self.grid.draw(new, self.color)

while len(self.body) > 3: #判断蛇的长度，如果大于3个格，则用while循环返回蛇的初始长度

pop = self.body.pop()

self.grid.draw(pop, self.grid.bg)

#蛇吃到了特殊食物3，改变了自身的颜色

def change\_color(self, new, color): #定义change\_color()方法，改变蛇的颜色

self.color = color #蛇改变为特殊食物的颜色

self.body.insert(0, new) #显示更新后的蛇

for item in self.body: #遍历蛇的身体，全部改变为改变之后的颜色

self.grid.draw(item, self.color)

class SnakeGame(Frame): #定义一个SnakeGame类，Frame框架，用于游戏规则的设定和进行游戏控制

def \_\_init\_\_(self, master): #采用\_\_init\_\_()构造方法对SnakeGame进行初始化

Frame.\_\_init\_\_(self, master) #调用用\_\_init\_\_()构造方法对框架进行初始化

self.grid = Grid(master)

self.snake = Snake(self.grid)

self.food = Food(self.grid)

self.gameover = False #设置游戏结束的初始状态为False

self.score = 0 #设置游戏的初始得分为0

self.status = ['run', 'stop'] #设置蛇的状态有run和stop

self.speed = 300 #设置蛇的初始速度为300

self.grid.canvas.bind\_all("<KeyRelease>", self.key\_release)

self.display\_food() #调用初始化食物的显示颜色

#用于设置变色食物

self.color\_c = ("#FFB6C1","#6A5ACD","#0000FF","#F0FFF0","#FFFFE0","#F0F8FF","#EE82EE","#000000","#5FA8D9","#32CD32")

self.i = 0

#界面左侧显示分数

self.m = StringVar()

self.ft1 = ('Fixdsys', 40, "bold")

self.m1 = Message(master, textvariable=self.m, aspect=5000, font=self.ft1, bg="#FFB5C5")

self.m1.pack(side=LEFT, fill=Y)

self.m.set("Score:"+str(self.score))

#这个方法用于游戏重新开始时初始化游戏

def initial(self): #定义一个initial()方法，用于游戏重新开始进行游戏

self.gameover = False #初始化游戏结束时游戏重新开始初始的状态

self.score = 0 #初始化游戏结束时游戏重新开始初始时游戏的得分

self.m.set("Score:"+str(self.score))

self.snake.initial() #调用上面的initial()方法回到游戏重新开始时蛇的状态

#type1:普通食物 type2:减少3 type3:回到最初状态 type4:吃了会变色

def display\_food(self): #定义display\_food()，用于显示食物的颜色、位置和类型

self.food.color = "#23D978" #初始化食物的颜色为绿色

self.food.type = 1 #初始化食物的类型为第一个类型

if randint(0, 20) == 5: #判断出现的（0,20）范围内出现随机数是否为5，对随机数为5的数字设置其食物颜色和类型

self.food.color = "#FFD700"

self.food.type = 3

while (self.food.pos in self.snake.body): #判断蛇是否吃到食物，若吃到食物，该类食物随机出现

self.food.set\_pos()

self.food.display()

elif randint(0, 4) == 2: #判断出现的（0,4）范围内出现随机数是否为2，对随机数为2的数字设置其食物颜色和类型

self.food.color = "#EE82EE"

self.food.type = 4

while (self.food.pos in self.snake.body): #判断蛇是否吃到食物，若吃到食物，该类食物随机出现

self.food.set\_pos()

self.food.display()

elif len(self.snake.body) > 10 and randint(0, 16) == 1: #当蛇的长度>10个格子，且随机数为（0,16）设置其食物颜色和类型

self.food.color = "#FF0000"

self.food.type = 2

while (self.food.pos in self.snake.body): #判断蛇是否吃到食物，若吃到食物，该类食物随机出现

self.food.set\_pos()

self.food.display()

else: #若以上均不满足，当蛇吃到食物时，新食物随机出现

while (self.food.pos in self.snake.body):

self.food.set\_pos()

self.food.display()

def key\_release(self, event): #定义一个key\_release()方法，用于监听键盘事件

key = event.keysym #获取一个键盘事件

key\_dict = {"Up": "Down", "Down": "Up", "Left": "Right", "Right": "Left"} #定义一个字典，后续进行蛇是否向自己的反方向走进行判断

#蛇不可以向自己的反方向走

if key in key\_dict and not key == key\_dict[self.snake.direction]: #判断蛇是否是向自己的反方向走，如果不是则蛇前进

self.snake.direction = key

self.move()

def run(self): #定义一个run()方法判断游戏进程

#首先判断游戏是否暂停

if not self.status[0] == 'stop':

#判断游戏是否结束

if self.gameover == True: #如果游戏结束，则显示界面Game Over,your score是多少分

message = tkinter.messagebox.showinfo("Game Over", "your score: %d" % self.score)

if message == 'ok': #点击确定按钮，则调用initial函数游戏重新开始，初始化游戏。

self.initial()

if self.food.type == 4: #判断食物类型是否为4,是的话执行以下操作，令变色食物闪烁

color = self.color\_c[self.i]

self.i = (self.i+1)%10 #调用color\_c(包含颜色)进行食物颜色闪烁

self.food.color = color

self.food.display()

self.move(color)

else: #如果游戏不结束、食物类型不为4，调用move()方法

self.move()

self.after(self.speed, self.run) #调用after()方法改变蛇行进速度和游戏进程

def move(self, color="#EE82EE"): #定义move()方法，用与判断蛇的移动方向

# 计算蛇下一次移动的点

head = self.snake.body[0]

if self.snake.direction == 'Up': #如果蛇向上移动，执行以下操作

if head[1] - 1 < 0:

new = (head[0], 16)

else:

new = (head[0], head[1] - 1)

elif self.snake.direction == 'Down': #如果蛇向上移动，执行以下操作

new = (head[0], (head[1] + 1) % 16)

elif self.snake.direction == 'Left': #如果蛇向左移动，执行下面的操作

if head[0] - 1 < 0:

new = (24, head[1])

else:

new = (head[0] - 1, head[1])

else: #如果蛇不符合上述3个移动方向，执行下面的一行代码

new = ((head[0] + 1) % 24, head[1])

#撞到自己，设置游戏结束的标志位，等待下一循环

if new in self.snake.body:

self.gameover=True

#如果吃到食物

elif new == self.food.pos:

if self.food.type == 1: #如果吃到的食物类型是1，则调用add函数，蛇向一个指定点移动，并增加自身函数

self.snake.add(new)

elif self.food.type == 2: #如果吃到的食物类型是2，则调用cut\_down函数，剪短自身长度

self.snake.cut\_down(new)

elif self.food.type == 4: #如果吃到的食物类型是4，则调用change函数，改变自身颜色

self.snake.change\_color(new, color)

else: #如果没有吃到上述类型的食物，执行下面的方法

self.snake.init(new) #蛇吃到了黄色食物则回到原始长度。

self.display\_food() #蛇吃到了食物，则调用display\_food函数，随机生成食物

self.score = self.score+1 #只要蛇吃到了食物，不管什么食物，则分数自动加一

self.m.set("Score:" + str(self.score)) #更新分数显示区域的分数显示。

#什么都没撞到，继续前进

else:

self.snake.move(new)

if \_\_name\_\_ == '\_\_main\_\_': #\_\_name\_\_就是标识模块的名字的一个系统变量,通过if判断这样就可以执行“\_\_mian\_\_

root = Tk() #创建一个窗口

root.title("贪吃蛇小游戏")

snakegame = SnakeGame(root)

snakegame.run() #游戏开始

snakegame.mainloop() #进入消息循环