  78

**《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 王 萌

专业班级 信管16-3

学生学号 311609030318

教师姓名 徐 文 鹏

成 绩**：**

评 语：

教师签名：

日期：

目录

[1.1 课程设计教学目的： 2](#_Toc535318480)

[1.2 课设选择题目：贪吃蛇小游戏 2](#_Toc535318481)

[2、设计任务和要求 3](#_Toc535318482)

[2.1 课程设计教学任务和要求 3](#_Toc535318483)

[2.2 本课程设计具体任务和要求 3](#_Toc535318484)

[3、总体设计 4](#_Toc535318485)

[3.1设计图纸 4](#_Toc535318486)

[3.2任务分配： 9](#_Toc535318487)

[4、设计实现 9](#_Toc535318488)

[4.1图片展示 9](#_Toc535318489)

[4.2 实现结果评价：优秀。 12](#_Toc535318490)

[5、详细设计 12](#_Toc535318491)

[5.1 食物及毒物的产生 12](#_Toc535318492)

[5.2 判断蛇死了没： 13](#_Toc535318493)

[5.3蛇吃是否吃到食物或毒物： 14](#_Toc535318494)

[5.4蛇吃到毒物后的反应： 15](#_Toc535318495)

[6、调试与测试 17](#_Toc535318496)

[6.1蛇吃到食物时食物的产生。 17](#_Toc535318497)

[7、设计总结 18](#_Toc535318498)

1、设计目的

## 1.1 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。

## 1.2 课设选择题目：贪吃蛇小游戏

1、初步想法是经典游戏贪吃蛇是一定要敲的，就像学代码，一定要敲一句“hello，world！”一样。

2、我们对于做一个贪吃蛇小游戏是相当有兴趣，而且决定投入精神、精力与热情于其中的。倘若我们做一个不感兴趣的项目，想必最终我们也就是浑水摸鱼，随便应付过去了，这样既浪费我们的时间与精力，又会使老师的教学目的事倍功半。

所以综上所述，在我们了解到有很多人也是做贪吃蛇这个项目的前提下，仍然固执的选择了做一个贪吃蛇小游戏。

# 2、设计任务和要求

## 2.1 课程设计教学任务和要求

本次课程设计的主要任务是以Python为开发语言完成一个100~300行左右规模的程序项目开发。

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

（1）实现贪吃蛇的基本功能；

（2）可以通过按来控制游戏过程；

（3）利用pyinstaller工具对程序进行打包；

（4）游戏界面美观，对用户友好；

（5）游戏提示显然，使用户一目了然；

（6）始终以一个用户的心理体验和制作游戏，使用户游戏体验良好。

# 3、总体设计

## 3.1设计图纸

软件总体流程图分析

模块初始化

游戏运行主体

main函数

游戏进入界面

游戏结束界面

蛇死亡

按其余键

游戏结束

按其余键

按esc键

按esc键

按esc键或q键

按esc键

函数调用关系图

贪吃蛇

Main函数

主体运行函数

结束界面

画食物

画毒物

画蛇

开始界面

蛇移动

蛇速度控制

蛇死亡判定

食物随机生成

蛇吃食物判定

程序终止

画成绩

画网格

略

无边界

蛇撞墙

软件功能图

游戏进入界面

游戏界面

退出

游戏结束界面

游戏暂停界面

按esc键

按其余任意键

按P键

按esc键

按esc键或q键

用键盘控制蛇的移动

蛇加速减速

蛇吃苹果加分

蛇加速减速

蛇吃毒苹果变色，混乱

幸运传送门

蛇的设计

**[0]**

**[1]**

**[-1]**

**[2]**

**头**

**身体**



**一节身体：**

**{‘x’:X,‘y’:Y}**

**[**

**]**

（毒）苹果的设计



**{‘x’:X,‘y’:Y}**

蛇的移动设计

**添加**

**删除**

蛇的方向控制设计

**向上**

**新头**

**{x,y }**

**{x,y+1 }**

**向下**

**新头**

**{x,y-1}**

**{x,y}**

**向左**

**新头**

**{x,y}**

**{x-1,y}**

**向右**

**新头**

**{x+1,y}**

**{x,y}**

蛇吃果实

**头**

**果实**

## 3.2任务分配：

李文鹏：贪吃蛇的创造和移动以及游戏具体优化；

# 王萌：（毒）苹果的创建、蛇吃到毒苹果的效果以及蛇的死亡判定。4、设计实现

## 4.1图片展示

游戏启动界面：

（1）最终实现结果



4-1 游戏开始界面

（2）实现结果评价：优秀。

基本实现基本想法，界面美观大方。

游戏结束界面：

（1）最终实现效果



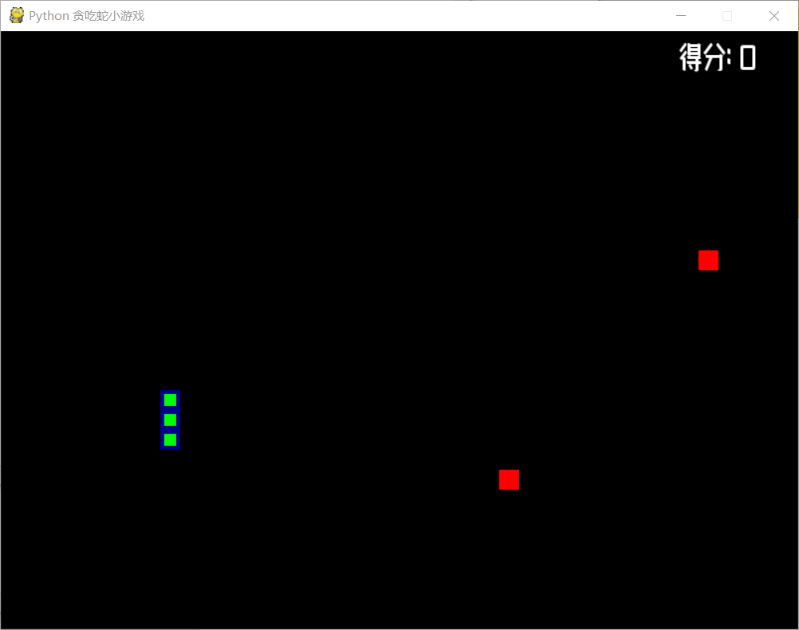
4-2游戏结束界面

（2）实现结果评价：优秀。

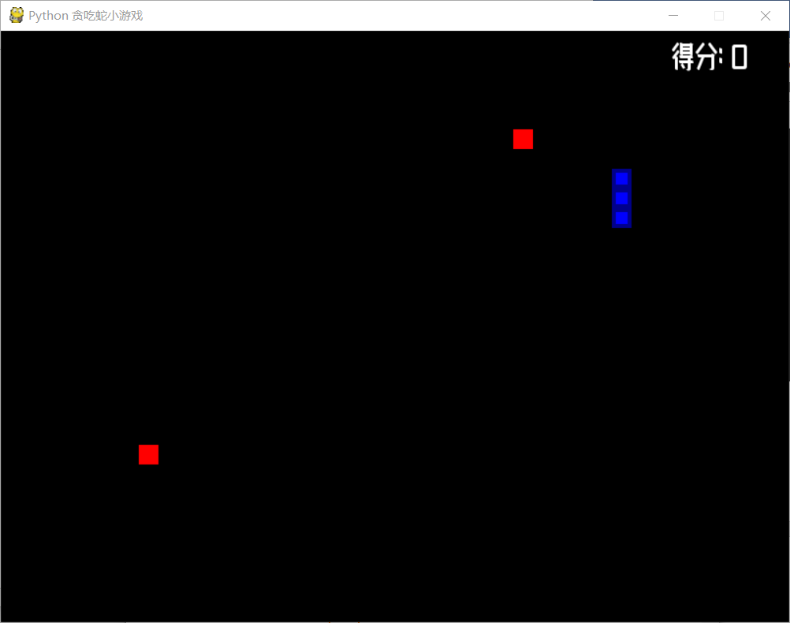
基本实现当初设想，遵循简约美的原则，颜色柔和不刺眼。

游戏界面

1. 最终实现效果



4-3 贪吃蛇吃到毒苹果



4-4 贪吃蛇游戏界面

## 4.2 实现结果评价：优秀。

本来有设想过，找张草地之类的图片拿来当背景，但仔细思考后，觉得有两点不妥之处：1、草地图案太繁琐，这与我们这款软件本身所贯彻的简约美原则相违背；2、草地颜色是绿色和中毒之后的蛇的颜色一样，会影响游戏观感。我最终还是没有改变，选择了黑色当底色，好处也有几点：1、黑色简约，但又不失去一丝神秘和冷酷；2、黑色对于任何鲜艳颜色都是很好的衬色，对游戏玩家的视觉感不会造成负面影响。所以，也许这个界面会让一些人感觉差强人意，但是，我还是蛮满意的，当然也还是有很大的进步空间。嗯，我觉得我超额完成自己的任务。

# 5、详细设计

## 5.1 食物及毒物的产生

代码：

food = get\_random\_location()

nofood = get\_random\_location()

#将食物画出来

def draw\_food(screen, food):

x = food['x'] \* cell\_size#在字典中获取数据随机坐标

y = food['y'] \* cell\_size

appleRect = pygame.Rect(x, y, cell\_size, cell\_size)

pygame.draw.rect(screen, Red, appleRect)

#把毒物画出来

def draw\_nofood(screen,nofood,snake\_coords,food):

x = nofood['x'] \* cell\_size

y = nofood['y'] \* cell\_size

appleRect = pygame.Rect(x, y, cell\_size, cell\_size)

pygame.draw.rect(screen, Green, appleRect)

def get\_random\_location():return {'x': random.randint(0, map\_width - 1), 'y': random.randint(0, map\_height - 1)}

分析：先定义一个locatioon函数，通过location获取随机坐标，然后食物刚好为窗口大小的一个单位，

## 5.2 判断蛇死了没：

代码：

ret = snake\_is\_alive(snake\_coords) #蛇是否还活着

if not ret:

dieSound.play()

break

def snake\_is\_alive(snake\_coords):

tag = True

if snake\_coords[HEAD]['x'] == -1 or snake\_coords[HEAD]['x'] == map\_width or snake\_coords[HEAD]['y'] == -1 or \

snake\_coords[HEAD]['y'] == map\_height:

chuanyue = random.choice([0,1])

if chuanyue == 0:

tag = True

luckDog[0]+=10

#print(luckDog)

else:

tag = False

for snake\_body in snake\_coords[1:]:

if snake\_body['x'] == snake\_coords[HEAD]['x'] and snake\_body['y'] == snake\_coords[HEAD]['y']:

tag = False # 蛇碰到自己身体啦

return tag

分析：判断蛇是否死亡，根据蛇头的坐标是否与身体坐标相等，来判断蛇是否碰到自己身体，还有就是根据蛇头的坐标是否出界，判断蛇是否碰墙，如果碰到自己的身体或者撞墙均为死亡跳出while true循环。从列表中的字典获取蛇头的坐标，判断以上内容

## 5.3蛇吃是否吃到食物或毒物：

def snake\_is\_eat\_food(snake\_coords, food, nofood, eatFoodSound, eatNoFood): #如果是列表或字典，那么函数内修改参数内容，就会影响到函数体外的对象。

if snake\_coords[HEAD]['x'] == food['x'] and snake\_coords[HEAD]['y'] == food['y']:

eatFoodSound.play()

food['x'] = random.randint(0, map\_width - 1)

food['y'] = random.randint(0, map\_height - 1) # 实物位置重新设置

nofood['x'] = random.randint(0, map\_width - 1)

nofood['y'] = random.randint(0, map\_height - 1)#毒物位置刷新

elif snake\_coords[HEAD]['x'] == nofood['x'] and snake\_coords[HEAD]['y'] == nofood['y']:

eatNoFood.play()

food['x'] = random.randint(0, map\_width - 1)

food['y'] = random.randint(0, map\_height - 1) # 实物位置重新设置

nofood['x'] = random.randint(0, map\_width - 1)

nofood['y'] = random.randint(0, map\_height - 1)#毒物位置刷新

global poison

poison = not poison

#print(poison)

del snake\_coords[-1]

else:

del snake\_coords[-1] #

分析：根据蛇头的坐标以及食物或毒物的坐标是否相等，判断蛇是否吃到食物或毒物，利用的是他们横纵坐标分别相等

## 5.4蛇吃到毒物后的反应：

代码：

while True:

event = pygame.event.poll()

if poison:

if event.type == QUIT:

terminate()

elif event.type == KEYDOWN:

if (event.key == K\_LEFT or event.key == K\_a) and direction != LEFT:

direction = RIGHT

elif (event.key == K\_RIGHT or event.key == K\_d) and direction != RIGHT:

direction = LEFT

elif (event.key == K\_UP or event.key == K\_w) and direction != UP:

direction = DOWN

elif (event.key == K\_DOWN or event.key == K\_s) and direction != DOWN:

direction = UP

elif event.key == K\_ESCAPE:

terminate()

elif event.key == K\_p:

pause\_flag = 0

分析：如果蛇吃到毒物控制蛇的按键控制方向与之前的相反，而且蛇的长度不变，如果没有吃到毒物，蛇继续前进，这个功能其实就是在按键控制方向上进行修改，先用poison判断是否中毒，再根据中毒进行操作，再次吃到毒物时蛇会恢复原状

# 6、调试与测试

## 6.1蛇吃到食物时食物的产生。

def snake\_is\_eat\_food(snake\_coords, food, nofood, eatFoodSound, eatNoFood): #如果是列表或字典，那么函数内修改参数内容，就会影响到函数体外的对象。

if snake\_coords[HEAD]['x'] == food['x'] and snake\_coords[HEAD]['y'] == food['y']:

eatFoodSound.play()

food['x'] = random.randint(0, map\_width - 1)

food['y'] = random.randint(0, map\_height - 1) # 实物位置重新设置

nofood['x'] = random.randint(0, map\_width - 1)

nofood['y'] = random.randint(0, map\_height - 1)#毒物位置刷新

elif snake\_coords[HEAD]['x'] == nofood['x'] and snake\_coords[HEAD]['y'] == nofood['y']:

eatNoFood.play()

food['x'] = random.randint(0, map\_width - 1)

food['y'] = random.randint(0, map\_height - 1) # 实物位置重新设置

nofood['x'] = random.randint(0, map\_width - 1)

nofood['y'] = random.randint(0, map\_height - 1)#毒物位置刷新

global poison

poison = not poison

#print(poison)

del snake\_coords[-1]

else:

del snake\_coords[-1] # 如果没有吃到实物, 就向前移动, 那么尾部一格删掉

#食（毒物）物随机生成

刚开始是产生一个随机食物，由于毒食物位置不变，在测试过程中，感觉比较容易判断哪个是毒物，然后进行改进，增加游戏难度，变为不管吃到食物或毒物，都会产生毒物和食物。由于蛇吃到毒物，控制蛇的按键方向与之前相反，刚开始在中毒函数引用控制蛇的方向，发现没有作用，最后在while true中引用成功

# 7、设计总结

经过这次课设，我收获很多，在贪吃蛇游戏制作过程，我们也引用了网上的代码，经过我们的一些改进，使得我们这个贪吃蛇游戏初见成果，在设计的过程中也遇到了一些问题，有得是因为粗心，有得是因为自己对知识掌握的不牢固，在小组当中，我也很荣幸担当了游戏测试员一职，在操作的过程中也发现了一些问题，过程很艰辛，结果是好的。游戏设计过程，非常感谢老师和同学对我的帮助。