贪吃蛇大冒险

目录

[一、 引言 0](#_Toc23679)

[1.1游戏背景及简介 0](#_Toc4242)

[1.2游戏操作及游戏规则 0](#_Toc32419)

[二、 游戏总体设计 0](#_Toc29123)

[三、编译语言以及用到的库文件描述 1](#_Toc3033)

[3.1 Python语言 1](#_Toc8427)

[3.2 pygame库 1](#_Toc16039)

[3.3 random库 1](#_Toc31183)

[3.4 time库 1](#_Toc18865)

[3.5 sys库 1](#_Toc23214)

[四、游戏设计与实现 2](#_Toc22458)

[4.1游戏整体设计 2](#_Toc31417)

[4.2主要函数解析 2](#_Toc1223)

[五、运行模块代码结构 4](#_Toc2348)

[六、游戏运行测试（截图） 5](#_Toc24898)

[七、总结 7](#_Toc608)

[八、参考文献 7](#_Toc22352)

[九、游戏完整代码 7](#_Toc21078)

# 引言

## 1.1游戏背景及简介

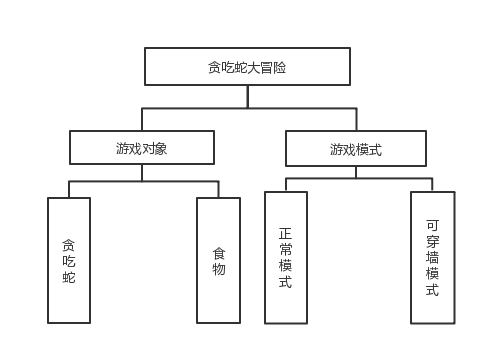
随着生活节奏的不断加快，人们面对的压力越来越大，如果不及时缓解，可能会使人们受到严重的影响。游戏作为缓解压力的一种有效方式，一直受到人们的欢迎。而贪吃蛇大冒险游戏作为一款经典的小型益智游戏，现有PC和手机等多平台版本，受到人们的喜爱。

在本次课设中，我结合所学知识，使用Python程序设计语言，编写了简易版的贪吃蛇大冒险小游戏。

## 1.2游戏操作及游戏规则

1. 游戏开始后，通过键盘上下左右键控制贪吃蛇移动寻找食物；
2. 贪吃蛇每吃一个食物，身长会增加一个单位；
3. 在正常模式中，贪吃蛇如果碰到墙面或咬到自身，则游戏失败；
4. 在可穿墙模式中，贪吃蛇可以穿过墙面，但咬到自身后，游戏失败。

# 游戏总体设计



**图1** 游戏总体设计框架图

# 三、编译语言以及用到的库文件描述

## 3.1 Python语言

Python是一种跨平台的计算机程序设计语言。是一种面向对象的动态类型语言，最初被设计用于编写自动化脚本(shell)，Python语言具有非常简洁且清晰的语法特点，随着版本的不断更新和语言新功能的添加，越来越多被用于独立的、大型项目的开发。目前，基于Python语言的相关技术正在飞速发展，用户数量急剧增大，相关的资源越来越多。

## 3.2 pygame库

Pygame是跨平台Pyth，Pygame 作者是 Pete Shinners， 协议为 GNU Lesser General Public License。Pygame包含图像、声音。建立在SDL基础上，允许实时电子游戏研发而无需被低级语言（如机器语言和汇编语言）束缚。基于这样一个设想，所有需要的游戏功能和理念（主要是图像方面）都完全简化为游戏逻辑本身，所有的资源结构都可以由高级语言提供，如Python。

## 3.3 random库

Random是产生随机数的库，即随机数发生器，random产生的是伪随机数或者说是用一种复杂的方法计算得到的序列值，因此每次运算时需要一个不同的种子值。种子值不同，得到的序列值也不同。

## 3.4 time库

time库是Python中处理时间的标准库，用于计算机时间的表达，提供获取系统时间并格式化输出功能，提供系统级精确计时功能，也可用于程序性能分析。

## 3.5 sys库

sys.exit（[ arg ]）用于程序的退出，sys.exit(n) 退出程序，正常退出时exit(0)。这是通过引发SystemExit 异常来实现的，因此遵循finally语句的子句所指定的清理操作try ，并且可以拦截外层的退出尝试。可选参数arg可以是一个整数，给出退出状态（默认为零）或其他类型的对象。如果它是整数，则零被认为是“成功终止”，并且任何非零值等视为“异常终止”。

# 四、游戏设计与实现

## 4.1游戏整体设计

1. 定义贪吃蛇对象，包括蛇头和蛇身两个部分，蛇头单独设置，蛇身使用列表来存储；
2. 定义食物对象，包括食物对象的初始化、随机颜色和随机位置的设置；
3. 定义初始化界面，在该界面中设置文本为“正常模式”、“可穿墙模式”和“退出”的三个按钮供玩家选择；
4. 定义正常模式，包括游戏窗口显示、主循环和判断贪吃蛇死亡的设置；
5. 定义可穿墙模式，包括游戏窗口显示、主循环和判断贪吃蛇死亡的设置；
6. 定义游戏结束窗口，包括窗口显示和最后得分显示的设置。

## 4.2主要函数解析

#### 4.2.1 new\_food()函数

食物对象初始化函数，传入形参为贪吃蛇蛇头坐标。当贪吃蛇吃掉食物后，通过该函数生成新的食物。通过传入形参判断新生成的食物坐标是否与蛇头坐标相同，若相同则重新生成新的坐标。

#### 4.2.2 start\_game()函数

正常模式的主循环函数，通过该函数进行正常模式下贪吃蛇和食物的初始化、玩家控制贪吃蛇运动、蛇头和蛇身的判断更新和得分统计实现。

#### 4.2.3 die\_snake()函数

正常模式下贪吃蛇死亡判断函数，传入形参为贪吃蛇蛇头和蛇身坐标数据。在该函数中设置布尔型变量die\_flag，若死亡设置为Ture并返回，没有死亡为False。通过遍历蛇身存储列表，判断蛇身坐标是否与蛇头坐标相同，若相同则判定贪吃蛇咬到自身，死亡。同时，判断贪吃蛇是否撞墙，及判断蛇头的x和y坐标是否窗体的宽高，若超过则死亡。

#### 4.2.4 start\_kgame()函数

可穿墙模式的主循环函数，通过该函数进行可穿墙模式下贪吃蛇和食物的初始化、玩家控制贪吃蛇运动、蛇头和蛇身的判断更新和得分统计实现。

#### 4.2.5 through\_snake()函数

可穿墙模式下贪吃蛇死亡判断函数，传入形参为贪吃蛇蛇头和蛇身坐标数据。在该函数中设置布尔型变量die\_flag，若死亡设置为Ture并返回，没有死亡为False。通过遍历蛇身存储列表，判断蛇身坐标是否与蛇头坐标相同，若相同则判定贪吃蛇咬到自身，死亡。

#### 4.2.6 show\_end()函数

游戏结束时，结束界面设置函数。在该函数中进行结束界面窗体的初始化，在窗体中显示玩家的最终得分，并在改函数中调用pygame库quit方法，使该库停止工作。

#### 4.2.7 into\_game()函数

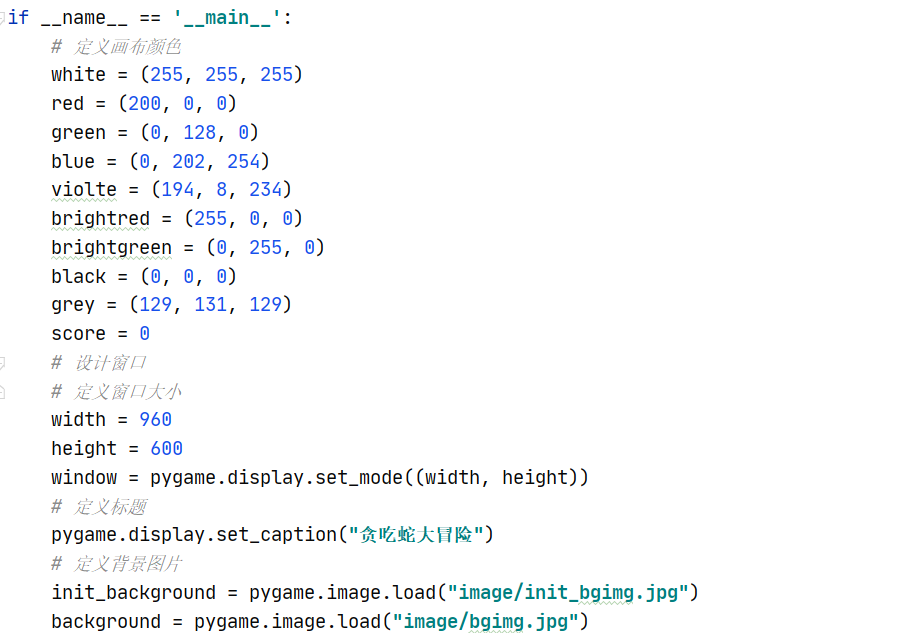
游戏初始界面实现函数，在该函数中进行游戏初始界面窗体的初始化。在该界面窗体中设置文本为“正常模式”、“可穿墙模式”和“退出”的三个点击按钮，通过设置循环，调用button函数（详细见6.2.8）监听玩家的点击。

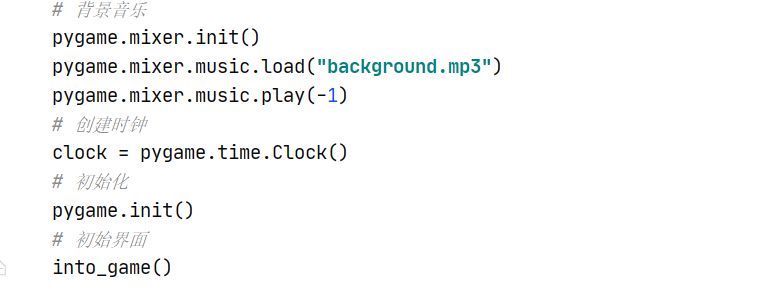
#### 4.2.8 button()函数

游戏初始界面按钮监听函数，在该函数中实现对玩家鼠标点击事件和键盘输入事件的监听。并根据玩家选择，运行按钮对应的触发函数。即点击“正常模式”按钮，运行正常模式的主循环函数，以此类推。

# 五、运行模块代码结构

在运行模块，进行pygame库的初始化、设置游戏背景音乐、显示窗口各个参数的设置以及游戏初始界面的显示设置。



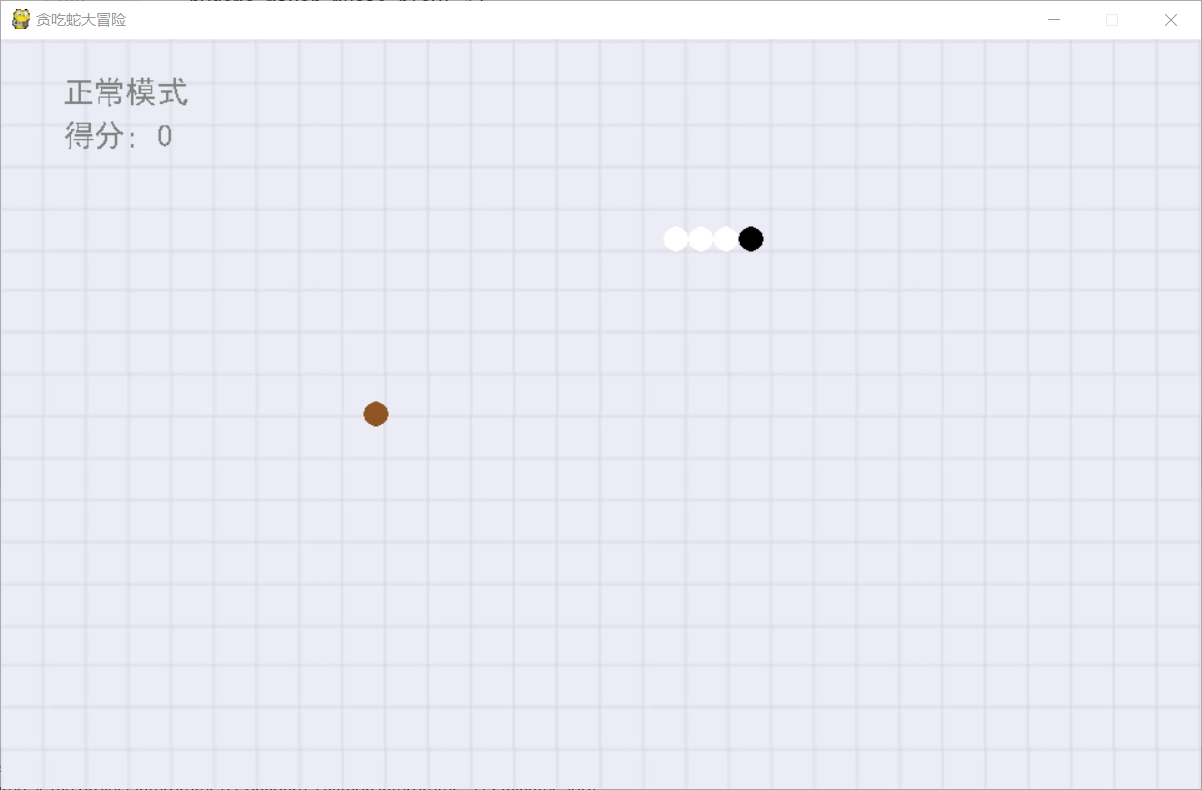


**图2** 运行模块代码结构

# 六、游戏运行测试（截图）



**图3** 游戏初始选择界面



**图4** 正常模式运行界面



**图5** 可穿墙模式运行界面



**图1** 游戏结束界面

# 七、总结

在本次课程设计中，我使用Python编程语言编写了贪吃蛇大冒险小游戏。主要进行了贪吃蛇对象和食物对象的设计，以及贪吃蛇小游戏正常模式和可穿墙模式的实现。使用到了Python的pygame、random、time、sys等库，将课上所学的知识与实践进行了结合。

通过本次课设，我对Python语言有了更深刻的了解，基本掌握了pygame等多个库中基本方法的使用，并提高了自己的编程能力。但仍有很多不足，我会在今后的学习中不断努力，不断提高自己的专业能力。

最后，非常感谢老师一个学期的耐心教导，让我对Python的很多库及其方法有了一定的了解，并对Python这门编程语言有了一定认识。听了您的课我也深受启发，让我对日常专业学习遇到的问题有了更多更深的思考，使我受益匪浅。

# 八、参考文献

[1] 董付国编著.《Python程序设计基础（第2版）》，清华大学出版社 2017 北京

[2] [美]埃里克·马瑟斯（Eric Matthes）|译者袁国忠.《Python编程 从入门到实践》人民邮电出版社 2016

# 九、游戏完整代码

import sys  
import time  
import pygame  
from random import \*  
# Position类，通过其构造函数，设置x和y  
class Position(object):  
 def \_\_init\_\_(self, x, y):  
 self.x = x  
 self.y = y  
# 生成随机的食物  
def new\_food(head):  
 while True:  
 new\_food = Position(randint(0, 48) \* 20, randint(0, 29) \* 20)  
 # 判断新生成的事物是否和贪吃蛇蛇头重合，重合则不创键  
 if new\_food.x != head.x and new\_food.y != head.y:  
 break  
 else:  
 continue  
 return new\_food  
# 绘制，在窗体中绘制贪吃蛇、食物  
# color:颜色，position: 坐标  
def rect(color, position):  
 pygame.draw.circle(window, color, (position.x, position.y), 10)  
# 初始界面和游戏中点差退出游戏时  
def exit\_end():  
 pygame.quit()  
 quit()  
# 游戏结束时，显示得分的窗体的设置  
def show\_end():  
 # 设计窗口  
 # 定义窗口大小  
 small\_window = pygame.display.set\_mode((960, 600))  
 init\_background = pygame.image.load("image/init\_bgimg.jpg")  
 small\_window.blit(init\_background, (0, 0))  
 # 定义标题  
 pygame.display.set\_caption("贪吃蛇大冒险")  
 # 定义背景图片  
 font = pygame.font.SysFont("simHei", 40)  
 fontsurf = font.render('游戏结束! 你的得分为: %s' % score, False, black)  
 small\_window.blit(fontsurf, (250, 200))  
 pygame.display.update()  
 time.sleep(2)  
 pygame.quit()  
 sys.exit()  
# 正常模式死亡设置  
# head: 蛇头， snake\_body:蛇身  
def die\_snake(head, snake\_body):  
 # 定义标志物，默认为false，true时判定贪吃蛇碰到自己，死亡  
 die\_flag = False  
 # 遍历存放贪吃蛇位姿的列表，从第1个开始，(第0个位蛇头)  
 for body in snake\_body[1:]:  
 # 如果蛇头的xy和蛇身xy相等，则判定相撞，设置flag为ture  
 if head.x == body.x and head.y == body.y:  
 die\_flag = True  
 # 若蛇头的xy在显示窗体外，或flag为true，则显示结束界面，并退出游戏  
 if head.x < 0 or head.x > 960 or head.y < 0 or head.y > 600 or die\_flag:  
 pygame.mixer.music.stop()  
 show\_end()  
# 正常模式主体设置  
def start\_game():  
 # 定义存分数的全局变量  
 global score  
 global color  
 color = (randint(10, 255), randint(10, 255), randint(10, 255))  
 # 定义存放玩家键盘输入运动方向的变量，初始为向右  
 run\_direction = "right"  
 # 定义贪吃蛇运动方向的变量，初始为玩家键入方向  
 run = run\_direction  
 # 实例化蛇头、蛇身、食物对象  
 head = Position(160, 160)  
 # 初始化蛇身长度为3个单位  
 snake\_body = [Position(head.x, head.y + 20), Position(head.x, head.y + 40), Position(head.x, head.y + 60)]  
 # 初始化食物位置  
 food = Position(300, 300)  
 # 死循环  
 while True:  
 window.blit(background, (0,0))  
 # 监听玩家键盘输入的运动方向值，并根据输入转为up、down、right或left，方便程序中调用  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 show\_end()  
 elif event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_UP:  
 run\_direction = "up"  
 elif event.key == pygame.K\_RIGHT:  
 run\_direction = "right"  
 elif event.key == pygame.K\_LEFT:  
 run\_direction = "left"  
 elif event.key == pygame.K\_DOWN:  
 run\_direction = "down"  
 # 食物  
 rect(color, food)  
 # 蛇头  
 rect(black, head)  
 # 蛇身  
 for pos in snake\_body:  
 rect(white, pos)  
 # 判断贪吃蛇原运动方向与玩家键盘输入的运动方向是否违反正常运动情况  
 if run == "up" and not run\_direction == "down":  
 run = run\_direction  
 elif run == "down" and not run\_direction == "up":  
 run = run\_direction  
 elif run == "left" and not run\_direction == "right":  
 run = run\_direction  
 elif run == "right" and not run\_direction == "left":  
 run = run\_direction  
 # 插入蛇头位置到蛇身列表中  
 snake\_body.insert(0, Position(head.x, head.y))  
 # 根据玩家键入方向进行蛇头xy的更新  
 if run == "up":  
 head.y -= 20  
 elif run == "down":  
 head.y += 20  
 elif run == "left":  
 head.x -= 20  
 elif run == "right":  
 head.x += 20  
 # 判断是否死亡  
 die\_snake(head, snake\_body)  
 # 判断蛇头和食物坐标，若相等，则加分，并生成新的食物  
 if head.x == food.x and head.y == food.y:  
 score += 1  
 food = new\_food(head)  
 color = (randint(10, 255), randint(10, 255), randint(10, 255))  
 else:  
 snake\_body.pop()  
 font = pygame.font.SysFont("simHei", 25)  
 mode\_title = font.render('正常模式', False, grey)  
 socre\_title = font.render('得分: %s' % score, False, grey)  
 window.blit(mode\_title, (50, 30))  
 window.blit(socre\_title, (50, 65))  
 # 绘制更新  
 pygame.display.update()  
 # 通过帧率设置贪吃蛇速度  
 clock.tick(8)  
# 可穿墙模式死亡设置  
# head:蛇头，snake\_body:蛇身  
def through\_snake(head, snake\_body):  
 # 定义标志位  
 die\_flag = False  
 # 遍历，蛇头碰到蛇身时，flag为true退出游戏  
 for body in snake\_body[1:]:  
 if head.x == body.x and head.y == body.y:  
 die\_flag = True  
 if die\_flag:  
 pygame.mixer.music.stop()  
 show\_end()  
 else: # 当蛇头的xy出窗体时  
 # 四种穿墙情况，分别设置  
 if head.x < 0:  
 head.x = 960  
 if head.x > 960:  
 head.x = 0  
 if head.y < 0:  
 head.y = 600  
 if head.y > 600:  
 head.y = 0  
# 穿墙模式主体设置  
def start\_kgame():  
 global score  
 global color  
 color = (randint(10, 255), randint(10, 255), randint(10, 255))  
 # 定义蛇初始方向  
 run\_direction = "up"  
 run = run\_direction  
 # 实例化蛇头、蛇身、食物对象  
 head = Position(160, 160)  
 # 三格  
 snake\_body = [Position(head.x, head.y + 20), Position(head.x, head.y + 40), Position(head.x, head.y + 60)]  
 # 初始化事物位置  
 food = Position(300, 300)  
 # 死循环，监听键盘键值  
 while True:  
 window.blit(background, (0, 0))  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 show\_end()  
 elif event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_UP:  
 run\_direction = "up"  
 elif event.key == pygame.K\_RIGHT:  
 run\_direction = "right"  
 elif event.key == pygame.K\_LEFT:  
 run\_direction = "left"  
 elif event.key == pygame.K\_DOWN:  
 run\_direction = "down"  
 # 食物  
 rect(color, food)  
 # 蛇头  
 rect(black, head)  
 # 蛇身  
 for pos in snake\_body:  
 rect(white, pos)  
 # 判断贪吃蛇原运动方向与玩家键盘输入的运动方向是否违反正常运动情况  
 if run == "up" and not run\_direction == "down": # 若运动方向为向上，玩家输入运动方向向下，则违背贪吃蛇正常运动情况  
 run = run\_direction  
 elif run == "down" and not run\_direction == "up":  
 run = run\_direction  
 elif run == "left" and not run\_direction == "right":  
 run = run\_direction  
 elif run == "right" and not run\_direction == "left":  
 run = run\_direction  
 # 插入蛇头位置到蛇身列表中  
 snake\_body.insert(0, Position(head.x, head.y))  
 # 根据玩家键入方向进行蛇头xy的更新  
 if run == "up":  
 head.y -= 20  
 elif run == "down":  
 head.y += 20  
 elif run == "left":  
 head.x -= 20  
 elif run == "right":  
 head.x += 20  
 # 穿墙实现  
 through\_snake(head, snake\_body)  
 # 判断是否加分和随机生成新的食物  
 if head.x == food.x and head.y == food.y:  
 score += 1  
 food = new\_food(head)  
 color = (randint(10, 255), randint(10, 255), randint(10, 255))  
 else:  
 snake\_body.pop()  
 font = pygame.font.SysFont("simHei", 25)  
 mode\_title = font.render('穿墙模式', False, grey)  
 socre\_title = font.render('得分: %s' % score, False, grey)  
 window.blit(mode\_title, (50, 30))  
 window.blit(socre\_title, (50, 65))  
 # 绘制更新  
 pygame.display.update()  
 # 通过帧率设置贪吃蛇速度  
 clock.tick(8)  
# 监听函数，监听键盘输入  
# msg: 按钮信息，x: 按钮的x轴，y: 按钮的y轴，w: 按钮的宽，h: 按钮的高，ic: 按钮初始颜色，ac: 按钮按下颜色，action: 按钮按下的动作  
def button(msg, x, y, w, h, ic, ac, action=None):  
 # 获取鼠标位置  
 mouse = pygame.mouse.get\_pos()  
 # 获取键盘输入  
 click = pygame.mouse.get\_pressed()  
 if x + w > mouse[0] > x and y + h > mouse[1] > y:  
 pygame.draw.rect(window, ac, (x, y, w, h))  
 if click[0] == 1 and action != None:  
 action()  
 else:  
 pygame.draw.rect(window, ic, (x, y, w, h))  
 # 设置按钮中的文字样式和居中对齐  
 font = pygame.font.SysFont('simHei', 20)  
 smallfont = font.render(msg, True, white)  
 smallrect = smallfont.get\_rect()  
 smallrect.center = ((x + (w / 2)), (y + (h / 2)))  
 window.blit(smallfont, smallrect)  
# 游戏初始界面，选择模式  
def into\_game():  
 into = True  
 while into:  
 window.blit(init\_background, (0, 0))  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 exit\_end()  
 # 设置字体  
 font = pygame.font.SysFont("simHei", 50)  
 # 初始界面显示文字  
 fontsurf = font.render('欢迎来到贪吃蛇大冒险!', True, black) # 文字  
 fontrect = fontsurf.get\_rect()  
 fontrect.center = ((width / 2), 200)  
 window.blit(fontsurf, fontrect)  
 button("正常模式", 370, 370, 200, 40, blue, brightred, start\_game)  
 button("可穿墙模式", 370, 420, 200, 40, violte, brightred, start\_kgame)  
 button("退出", 370, 470, 200, 40, red,brightred, exit\_end)  
 pygame.display.update()  
 clock.tick(15)  
if \_\_name\_\_ == '\_\_main\_\_':  
 # 定义画布颜色  
 white = (255, 255, 255)  
 red = (200, 0, 0)  
 green = (0, 128, 0)  
 blue = (0, 202, 254)  
 violte = (194, 8, 234)  
 brightred = (255, 0, 0)  
 brightgreen = (0, 255, 0)  
 black = (0, 0, 0)  
 grey = (129, 131, 129)  
 score = 0  
 # 设计窗口  
 # 定义窗口大小  
 width = 960  
 height = 600  
 window = pygame.display.set\_mode((width, height))  
 # 定义标题  
 pygame.display.set\_caption("贪吃蛇大冒险")  
 # 定义背景图片  
 init\_background = pygame.image.load("image/init\_bgimg.jpg")  
 background = pygame.image.load("image/bgimg.jpg")  
 # 背景音乐  
 pygame.mixer.init()  
 pygame.mixer.music.load("background.mp3")  
 pygame.mixer.music.play(-1)  
 # 创建时钟  
 clock = pygame.time.Clock()  
 # 初始化  
 pygame.init()  
 # 初始界面  
 into\_game()