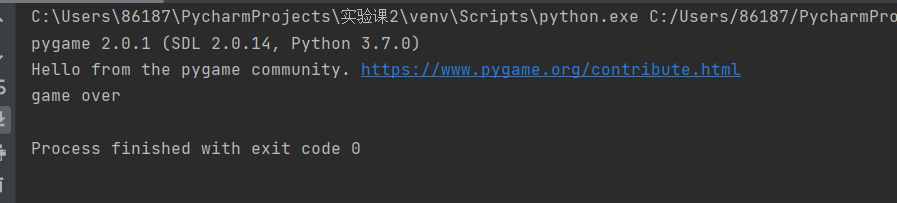
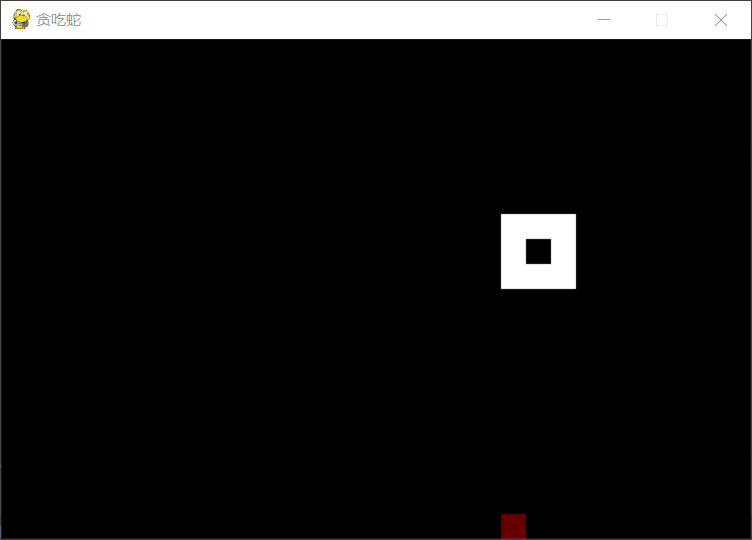
**实验题目1：**增加死亡条件：（头接触身体会死亡）。

**代码及运行：**

# 这是一个pygame的最小开发框架  
import pygame  
import sys  
import random  
from pygame.locals import \*  
pygame.init() # 初试化pygame  
  
white\_colour = pygame.Color(255, 255, 255) # 白色  
black\_colour = pygame.Color(0, 0, 0) # 黑色  
red\_colour = pygame.Color(100,0,0) # 红色  
game\_surface = pygame.display.set\_mode((600, 400)) # 设置pygame游戏框大小  
pygame.display.set\_caption("贪吃蛇") # 设置游戏标题  
  
def new\_direction(key,direction):  
 if key == 119 and direction != "down":  
 direction = "up"  
 elif key == 115 and direction != "up":  
 direction = "down"  
 elif key == 100 and direction != "left":  
 direction = "right"  
 elif key == 97 and direction != "right":  
 direction = "left"  
 elif key == 27:  
 sys.exit()  
 else:  
 pass  
 return direction  
def new\_position(head\_position,direction):  
 if direction == "up":  
 head\_position[1] -= 20  
 elif direction == "down":  
 head\_position[1] += 20  
 elif direction == "left":  
 head\_position[0] -= 20  
 elif direction == "right":  
 head\_position[0] += 20  
 else:  
 pass  
 return head\_position  
  
def main():  
 head\_position = [100, 100] # 蛇的初试位置  
 food\_position = [300, 100] # 食物的初始位置  
 snake\_position = [[100, 100]]  
 empty\_position = []  
 for i in range(30):  
 for j in range(20):  
 empty\_position.append([i\*20,j\*20])  
 direction = "right" # 初始化方向  
 EVENT\_time = pygame.USEREVENT+1  
 pygame.time.set\_timer(EVENT\_time, 400)  
 while True:  
 game\_surface.fill(black\_colour) # 背景填充为黑色  
 for i in range(len(snake\_position)):  
 pygame.draw.rect(game\_surface, white\_colour, Rect(snake\_position[i][0], snake\_position[i][1], 20, 20)) # 在[100, 100]处画20\*20的矩形  
 pygame.draw.rect(game\_surface, red\_colour, Rect(food\_position[0], food\_position[1], 20, 20)) # 在[100, 100]处画20\*20的矩形  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT: # 如果是退出键则退出游戏  
 sys.exit()  
 elif event.type == pygame.KEYDOWN:  
 direction = new\_direction(event.key, direction)  
 elif event.type == EVENT\_time:  
 head\_position = new\_position(head\_position, direction)  
 snake\_position.insert(0,[head\_position[0],head\_position[1]])  
 if head\_position in empty\_position:  
 empty\_position.remove(head\_position)  
 else:  
 print("game over")  
 sys.exit()  
 if food\_position == head\_position:  
 food\_position = random.choice(empty\_position)  
 else:  
 empty\_position.append(snake\_position.pop())  
 if head\_position[0] < 0 or head\_position[0]>590:  
 print("game over")  
 sys.exit()  
 elif head\_position[1] < 0 or head\_position[1] > 390:  
 print("game over")  
 sys.exit()  
 # 增加死亡判断功能  
 pygame.display.update()  
  
main()

****

**实验题目2：**为蛇提供加速功能。

**代码：**

# 这是一个pygame的最小开发框架  
import pygame  
import sys  
import random  
from pygame.locals import \*  
pygame.init() # 初试化pygame  
  
white\_colour = pygame.Color(255, 255, 255) # 白色  
black\_colour = pygame.Color(0, 0, 0) # 黑色  
red\_colour = pygame.Color(100,0,0) # 红色  
game\_surface = pygame.display.set\_mode((600, 400)) # 设置pygame游戏框大小  
pygame.display.set\_caption("贪吃蛇") # 设置游戏标题  
  
def new\_direction(key,direction):  
 if key == 119 and direction != "down":  
 direction = "up"  
 elif key == 115 and direction != "up":  
 direction = "down"  
 elif key == 100 and direction != "left":  
 direction = "right"  
 elif key == 97 and direction != "right":  
 direction = "left"  
 elif key == 27:  
 sys.exit()  
 else:  
 pass  
 return direction  
def new\_position(head\_position,direction):  
 if direction == "up":  
 head\_position[1] -= 20  
 elif direction == "down":  
 head\_position[1] += 20  
 elif direction == "left":  
 head\_position[0] -= 20  
 elif direction == "right":  
 head\_position[0] += 20  
 else:  
 pass  
 return head\_position  
def new\_positions(head\_position,direction):  
 if direction == "up":  
 head\_position[1] -= 40  
 elif direction == "down":  
 head\_position[1] += 40  
 elif direction == "left":  
 head\_position[0] -= 40  
 elif direction == "right":  
 head\_position[0] += 40  
 else:  
 pass  
 return head\_position  
def main():  
 t=1  
 head\_position = [100, 100] # 蛇的初试位置  
 food\_position = [300, 100] # 食物的初始位置  
 snake\_position = [[100, 100]]  
 empty\_position = []  
 for i in range(30):  
 for j in range(20):  
 empty\_position.append([i\*20,j\*20])  
 direction = "right" # 初始化方向  
 EVENT\_time = pygame.USEREVENT+1  
 pygame.time.set\_timer(EVENT\_time, 400)  
 while True:  
 game\_surface.fill(black\_colour) # 背景填充为黑色  
 for i in range(len(snake\_position)):  
 pygame.draw.rect(game\_surface, white\_colour, Rect(snake\_position[i][0], snake\_position[i][1], 20, 20)) # 在[100, 100]处画20\*20的矩形  
 pygame.draw.rect(game\_surface, red\_colour, Rect(food\_position[0], food\_position[1], 20, 20)) # 在[100, 100]处画20\*20的矩形  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT: # 如果是退出键则退出游戏  
 sys.exit()  
 elif event.type == pygame.KEYDOWN:  
 direction = new\_direction(event.key, direction)  
 if event.key == 32:  
 t = -t  
 elif event.type == EVENT\_time:  
 if t == 1:  
 head\_position = new\_position(head\_position, direction)  
 elif t == -1:  
 head\_position = new\_positions(head\_position, direction)  
 snake\_position.insert(0,[head\_position[0],head\_position[1]])  
 if head\_position in empty\_position:  
 empty\_position.remove(head\_position)  
 else:  
 print("game over")  
 sys.exit()  
 if food\_position == head\_position:  
 food\_position = random.choice(empty\_position)  
 else:  
 empty\_position.append(snake\_position.pop())  
 if head\_position[0] < 0 or head\_position[0]>590:  
 print("game over")  
 sys.exit()  
 elif head\_position[1] < 0 or head\_position[1] > 390:  
 print("game over")  
 sys.exit()  
 # 增加死亡判断功能  
 pygame.display.update()  
  
main()

**此版本bug较多，演示看视频。**

**实验题目3：**优化界面，解决bug

**代码：**

# 这是一个pygame的最小开发框架  
import pygame  
import sys  
import random  
from pygame.locals import \*  
pygame.init() # 初试化pygame  
  
white\_colour = pygame.Color(200, 255, 200) # 白色  
black\_colour = pygame.Color(0, 0, 0) # 黑色  
red\_colour = pygame.Color(100,0,0) # 红色  
green\_colour = pygame.Color(30,100,0) # 绿色  
game\_surface = pygame.display.set\_mode((600, 400)) # 设置pygame游戏框大小  
pygame.display.set\_caption("贪吃蛇") # 设置游戏标题  
  
def new\_direction(key,direction):  
 if key == 119 and direction != "down":  
 direction = "up"  
 elif key == 115 and direction != "up":  
 direction = "down"  
 elif key == 100 and direction != "left":  
 direction = "right"  
 elif key == 97 and direction != "right":  
 direction = "left"  
 elif key == 27:  
 sys.exit()  
 else:  
 pass  
 return direction  
def new\_position(head\_position,direction):  
 if direction == "up":  
 head\_position[1] -= 20  
 elif direction == "down":  
 head\_position[1] += 20  
 elif direction == "left":  
 head\_position[0] -= 20  
 elif direction == "right":  
 head\_position[0] += 20  
 else:  
 pass  
 return head\_position  
  
def main():  
 print("按空格开始")  
 t = 1  
 head\_position = [100, 100] # 蛇的初试位置  
 food\_position = [300, 100] # 食物的初始位置  
 snake\_position = [[100, 100]]  
 empty\_position = []  
 for i in range(30):  
 for j in range(20):  
 empty\_position.append([i\*20,j\*20])  
 direction = "right" # 初始化方向  
 EVENT\_time = pygame.USEREVENT + 1  
 while True:  
 game\_surface.fill(black\_colour) # 背景填充为黑色  
 for i in range(len(snake\_position)):  
 pygame.draw.rect(game\_surface, white\_colour, Rect(snake\_position[i][0], snake\_position[i][1], 20, 20)) # 在[100, 100]处画20\*20的矩形  
 pygame.draw.rect(game\_surface, red\_colour, Rect(food\_position[0], food\_position[1], 20, 20))  
 pygame.draw.rect(game\_surface, green\_colour, Rect(head\_position[0], head\_position[1], 20, 20))# 在[100, 100]处画20\*20的矩形  
 for event in pygame.event.get():  
 if event.type == pygame.KEYDOWN and event.key == 32:  
 t = -t  
 if t == -1:  
 pygame.time.set\_timer(EVENT\_time, 600)  
 break  
 elif t == 1:  
 pygame.time.set\_timer(EVENT\_time, 200)  
 break  
 if event.type == pygame.QUIT: # 如果是退出键则退出游戏  
 sys.exit()  
 elif event.type == pygame.KEYDOWN:  
 direction = new\_direction(event.key, direction)  
 elif event.type == EVENT\_time:  
 head\_position = new\_position(head\_position, direction)  
 snake\_position.insert(0,[head\_position[0],head\_position[1]])  
 if head\_position in empty\_position:  
 empty\_position.remove(head\_position)  
 else:  
 print("game over")  
 sys.exit()  
 if food\_position == head\_position:  
 food\_position = random.choice(empty\_position)  
 else:  
 empty\_position.append(snake\_position.pop())  
 if head\_position[0] < 0 or head\_position[0]>590:  
 print("game over")  
 sys.exit()  
 elif head\_position[1] < 0 or head\_position[1] > 390:  
 print("game over")  
 sys.exit()  
 # 增加死亡判断功能  
 pygame.display.update()  
  
main()

**演示看视频。**