***贪吃蛇python源码***

*贪吃蛇源码，初始源码来自于网络。*

*初学python，做了部分修改，加了些有趣的功能进行测试。*

*增加了部分注释；增加背景设置；增加历史高分统计；增加当前玩家排名；增加幸运食物，吃了有惊喜！*

*开始：回车键*

*暂停：空格键*

*操作：控制上下左右（ w、s、a、d ） or（ up、down、left、right）*

*自动背景开启和关闭： L*

*手动背景RGB值微调：r、g、b 键（L开启才可用）*

*游戏界面*

**

*Python 源码*

*"""贪吃蛇"""*import random  
import sys  
import time  
import copy  
import pygame  
from pygame.locals import \*  
from collections import deque  
  
SCREEN\_WIDTH = 800 # 屏幕宽度  
SCREEN\_HEIGHT = 480 # 屏幕高度  
SIZE = 20 # 小方格大小  
LINE\_WIDTH = 3 # 网格线宽度  
  
# 游戏区域的坐标范围  
SCOPE\_X = (0, SCREEN\_WIDTH // SIZE - 1)  
SCOPE\_Y = (2, SCREEN\_HEIGHT // SIZE - 1)  
  
# 食物的分值及颜色  
# FOOD\_STYLE\_LIST = [(10, (255, 100, 100)), (20, (100, 255, 100)), (30, (100, 100, 255))]  
FOOD\_STYLE\_LIST = [(10, (255, 0, 0)), (20, (0, 255, 0)), (30, (0, 0, 255)), (100, (255, 255, 0))]  
# 幸运食物出现概率（蛇身减一，速度不增加）  
FOOD\_LUCK=(0,7)  
# LIGHT = (100, 100, 100)  
DARK = (200, 200, 200) # 蛇的颜色  
BLACK = (0, 0, 0) # 网格线颜色  
RED = (200, 30, 30) # 红色，GAME OVER 的字体颜色  
BGCOLOR = (40, 40, 60) # 背景色  
  
  
  
  
def print\_text(screen, font, x, y, text, fcolor=(255, 255, 255)):  
 imgText = font.render(text, True, fcolor)  
 screen.blit(imgText, (x, y))  
  
  
# 初始化蛇  
def init\_snake():  
 snake = deque()  
 snake.append((2, SCOPE\_Y[0]))  
 # snake.append((1, SCOPE\_Y[0]+1)), snake.append((1, SCOPE\_Y[0])), snake.append((1, SCOPE\_Y[0]+2))  
 snake.append((1, SCOPE\_Y[0]))  
 snake.append((0, SCOPE\_Y[0]))  
 return snake  
  
  
def create\_food(snake):  
 food\_x = random.randint(SCOPE\_X[0], SCOPE\_X[1])  
 food\_y = random.randint(SCOPE\_Y[0], SCOPE\_Y[1])  
 while (food\_x, food\_y) in snake:  
 # 如果食物出现在蛇身上，则重来  
 food\_x = random.randint(SCOPE\_X[0], SCOPE\_X[1])  
 food\_y = random.randint(SCOPE\_Y[0], SCOPE\_Y[1])  
 return food\_x, food\_y  
  
  
def get\_food\_style():  
 if random.randint(FOOD\_LUCK[0],FOOD\_LUCK[1])==FOOD\_LUCK[1]:  
 return FOOD\_STYLE\_LIST[3]  
 else:  
 return FOOD\_STYLE\_LIST[random.randint(0, 2)]  
  
  
def main():  
 pygame.init()  
 screen = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT)) #pygame创建窗口  
 pygame.display.set\_caption('贪吃蛇') #窗口名称  
  
 font1 = pygame.font.SysFont('SimHei', 16) # 得分的字体  
 font\_ypos = 12 #位置  
  
 font2 = pygame.font.Font(None, 72) # GAME OVER 的字体  
 fwidth, fheight = font2.size('GAME OVER')  
  
 # 如果蛇正在向右移动，那么快速点击向下向左，由于程序刷新没那么快，向下事件会被向左覆盖掉，导致蛇后退，直接GAME OVER  
 # b 变量就是用于防止这种情况的发生  
 b = True  
  
 # 蛇  
 snake = init\_snake()  
 # 食物  
 food = create\_food(snake)  
 food\_style = get\_food\_style()  
 # 方向  
 pos = (1, 0) #(左-1 右1，上-1 下1）  
  
 game\_over = True  
 start = False # 是否开始，当start = True，game\_over = True 时，才显示 GAME OVER  
 score = 0 # 得分  
 orispeed =0.5 # 原始速度  
 speed = orispeed  
 luck\_times=0  
 last\_move\_time = None  
 last\_draw\_time = None  
 pause = False # 暂停  
  
 BG\_r = 40 #手动背景调整  
 BG\_g = 40   
 BG\_b = 60  
  
 BG\_auto=True  
  
 REC\_score=[]  
 REC\_save=False  
  
 while True:  
 for event in pygame.event.get():  
 if event.type == QUIT: #关闭窗口事件直接退出  
 sys.exit()  
 elif event.type == KEYDOWN: #处理按键事件  
 if event.key == K\_RETURN: #处理回车键事件  
 if game\_over:  
 start = True #游戏开始  
 game\_over = False  
 b = True  
 snake = init\_snake()  
 food = create\_food(snake)  
 food\_style = get\_food\_style()  
 pos = (1, 0)  
 # 得分  
 score = 0  
 REC\_save=True  
 REC\_score.append(score)  
 last\_move\_time = time.time()  
 last\_draw\_time = last\_move\_time  
 luck\_times = 0  
 elif event.key == K\_SPACE: #处理空格键事件 暂停和继续  
 if not game\_over:  
 pause = not pause  
 elif event.key in (K\_w, K\_UP):  
 # 这个判断是为了防止蛇向上移时按了向下键，导致直接 GAME OVER  
 if b and not pos[1]:  
 pos = (0, -1)  
 b = False  
 elif event.key in (K\_s, K\_DOWN):  
 if b and not pos[1]:  
 pos = (0, 1)  
 b = False  
 elif event.key in (K\_a, K\_LEFT):  
 if b and not pos[0]:  
 pos = (-1, 0)  
 b = False  
 elif event.key in (K\_d, K\_RIGHT):  
 if b and not pos[0]:  
 pos = (1, 0)  
 b = False  
 elif event.key in (K\_r,K\_g,K\_b): #背景色调整 L键控制开关(R,G,B)键控制3原色  
 if event.key==K\_r:  
 BG\_r=BG\_r+5  
 if BG\_r>=125:  
 BG\_r=40  
 elif event.key==K\_g:  
 BG\_g=BG\_g+5  
 if BG\_g>=125:  
 BG\_g=40  
 elif event.key==K\_b:  
 BG\_g=BG\_b+5  
 if BG\_b>=125:  
 BG\_b=40  
 elif event.key==K\_l:  
 BG\_auto=not BG\_auto  
  
 # 填充背景色  
 if BG\_auto:  
 screen.fill(BGCOLOR)  
 else:  
 screen.fill((BG\_r,BG\_g,BG\_b))  
  
  
  
  
 # 画网格线 竖线  
 for x in range(SIZE, SCREEN\_WIDTH, SIZE):  
 pygame.draw.line(screen, BLACK, (x, SCOPE\_Y[0] \* SIZE), (x, SCREEN\_HEIGHT), LINE\_WIDTH)  
 # 画网格线 横线  
 for y in range(SCOPE\_Y[0] \* SIZE, SCREEN\_HEIGHT, SIZE):  
 pygame.draw.line(screen, BLACK, (0, y), (SCREEN\_WIDTH, y), LINE\_WIDTH)  
  
 if not game\_over:  
 curTime = time.time()  
 if curTime - last\_move\_time > speed: #控制移动速度speed值越小，刷新越快  
 if not pause:  
 b = True  
 last\_move\_time = curTime  
 next\_s = (snake[0][0] + pos[0], snake[0][1] + pos[1]) #按方向读取下一个前进位  
 if next\_s == food:  
 # 吃到了食物  
 snake.appendleft(next\_s)  
 score += food\_style[0]  
 if REC\_save: #记录得分  
 REC\_score.pop()  
 REC\_score.append(score)  
  
 if food\_style[0] == FOOD\_STYLE\_LIST[3][0] and len(snake) > 3:  
 snake.pop()  
 snake.pop()  
 luck\_times+=1  
 else:  
 speed = orispeed - 0.03 \* (score // 100-luck\_times)  
 food = create\_food(snake)  
 food\_style = get\_food\_style()  
 else:  
 if SCOPE\_X[0] <= next\_s[0] <= SCOPE\_X[1] and SCOPE\_Y[0] <= next\_s[1] <= SCOPE\_Y[1] \  
 and next\_s not in snake: #判断下一步是否越界和碰到自己  
 snake.appendleft(next\_s)  
 snake.pop() #没吃到食物，前进一步  
 else:  
 game\_over = True  
  
 # 画食物  
 if not game\_over:  
 # 避免 GAME OVER 的时候把 GAME OVER 的字给遮住了  
 if food\_style[0] == FOOD\_STYLE\_LIST[3][0]:  
 curTime = time.time()  
 if curTime-last\_draw\_time>0.01: #闪烁时间  
 pygame.draw.rect(screen, food\_style[1], (food[0] \* SIZE, food[1] \* SIZE, SIZE, SIZE), 0)  
 last\_draw\_time=curTime  
 else:  
 pygame.draw.rect(screen, food\_style[1], (food[0] \* SIZE, food[1] \* SIZE, SIZE, SIZE), 0)  
  
 # 画蛇  
 for s in snake:  
 pygame.draw.rect(screen, DARK, (s[0] \* SIZE + LINE\_WIDTH, s[1] \* SIZE + LINE\_WIDTH,  
 SIZE - LINE\_WIDTH \* 2, SIZE - LINE\_WIDTH \* 2), 0)  
  
 print\_text(screen, font1, 30, font\_ypos, f'速度: {score//100-luck\_times}')  
  
  
 print\_text(screen, font1, 630, font\_ypos, f'得分: {score}')  
  
 if game\_over:  
 if start:  
 print\_text(screen, font2, (SCREEN\_WIDTH - fwidth) // 2, (SCREEN\_HEIGHT - fheight) // 2, 'GAME OVER', RED)  
  
  
 TMP\_score=copy.deepcopy(REC\_score)  
 TMP\_score.sort()  
 TMP\_score.reverse()  
 if len(REC\_score):  
 print\_text(screen, font1, 230, font\_ypos, f'最高得分: {TMP\_score[0]}')  
 print\_text(screen, font1, 430, font\_ypos, f'当前排名: {TMP\_score.index(REC\_score[len(REC\_score)-1])+1}')  
 else:  
 print\_text(screen, font1, 230, font\_ypos, f'最高得分: {score}')  
 print\_text(screen, font1, 430, font\_ypos, f'当前排名: {score}')  
 pygame.display.update()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()