Toutris

(touch tetris)

组名:三个 python 匠

组长: 关尔佳(518120910160)

组员: 罗媛(518120910170)

王朝能(518120910162)

指导老师:鲍杨

前言

python 作为现今流行的一款计算机程序语言，拥有优秀的泛用性。python 拥有为数不少的跨平台 python 模块，其中的 pygame 模块因其对游戏功能的简洁实现而受到了我们关注。在小组寻找利用 pygame 模块所制作的传统游戏的过程中，我们发现家喻户晓的俄罗斯方块拥有创新改进的可能性。因此，小组希望能利用 pygame 模块，为其添加新的规则，制作新型俄罗斯方块。

一、 背景

俄罗斯方块规则十分简单，在 10X20 的场地内，每次会有一个方块从上方落下，方块共有七种不同形状(图 1)，玩家通过调整方块的位置和方向，使它们在屏幕底部拼出完整的一条或几条。这些完整的横条会随即消失，给新落下来的板块腾出空间，没有被消除掉的方块不断堆积起来，一旦堆到屏幕顶端，玩家便会失败，游戏结束。

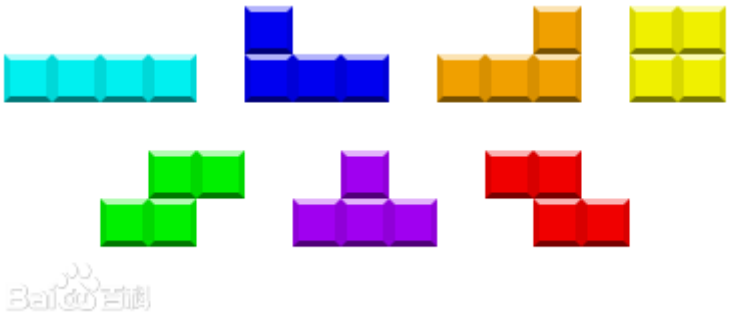

图 1 - 方块的 7 种形状

在网上我们可以见到不少利用 python 制作的古典俄罗斯方块。在此，我们希望能制作新型的俄罗斯方块，我们计划通过鼠标准心的移动设置方块。我们会在俄罗斯方块中设置关卡，关卡内有一定数量且不规则的方块在场地上方出现，玩家需要在下方 10X4 的区域内描绘方块(形状限定为传统方块的七种形状)，当玩家完成方块的描绘后，方块会上升。同样的，当能拼出完整的一条或几条后，方块消失，而通关则需要把所有方块弄消失。

二、 模块介绍

1） Pygame

是一个免费和开源的 python 编程语言库，用于制作基于优秀 SDL 库之上的游戏等多媒体应用程序。与 SDL 一样，pygame 具有高度可移植性，几乎可在所有平台和操作系统上运行。下述为我们所用到的代码 。

1. pygame.init :pygame 的初始函数，使用 pygame 模块的第一步

2. pygame.display :控制显示窗口和屏幕的 Pygame 模块
   - 2.1 pygame.display.set_mode (resolution=(0,0), flags=0, depth=0)
     初始化要显示的窗口或屏幕
   - 2.2 pygame.display.set_caption (title, icontitle=None)
     设置当前窗口标题

3. pygame.draw :绘制图形的 pygame 模块
   - 3.1 pygame.draw.rect(Surface, color, Rect, width=0)
     在指定面上绘制矩形
   - 3.2 pygame.draw.line (Surface, color, start_pos, end_pos, width=1)
     在指定面上绘制直线段

4. pygame.mouse :使用鼠标的 PyGame 模块
   - 4.1 pygame.mouse.get_pressed()
     获取鼠标按钮的状态
   - 4.2 pygame.mouse.get_pos()
     获取鼠标光标位置

5. pygame.event :用于与事件和队列交互的 PyGame 模块
   - 5.1 pygame.event.get()
     从队列中获取事件

2） numpy

Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。NumPy 是一个运行速度非常快的数学库，主要用于数组计算，其中包含强大的 N 维数组对象 ndarray，下述为我们所用到的代码

1. numpy.array() — 创建一个 N 维数组对象 ndarray

三、 代码

1. 导入必须的库

```
import pygame , time
from random import random
import numpy as np
from pygame.locals import MOUSEMOTION,MOUSEBUTTONDOWN,MOUSEBUTTONUP #导入 pygame 库定量
```

2. 设置游戏窗口

```
pygame.init()     #初始化 pygame
size = width,height = 360,570 #游戏窗口尺寸为 800x600 像素
#now 矩阵为 19 行 12 列，因此可以推算出每个小正方形块的边长为 30 像素
screen = pygame.display.set_mode(size)     #实例化一个窗口
pygame.display.set_caption("俄罗斯方块")  #设置窗口名称
```

3. 设置所需参数，其中 select 和 se 存的都是是鼠标描绘的方块坐标（由于循环会重新定义方块坐标，因此需要两个变量保存），now 存的是一个 19*12 的矩阵，相当于“游戏板”的作用

```
now=np.array([[0 if random()>0.8 else 1 for j in range(12) ] for i in range(6)]+[[0]*12]*13,dtype='bool')
score=0
select=[]
se=[]
```

4. 定义消行动作

```
def clear():
    global score
    h,w=now.shape
    for i in range(h):    #消除空行，即将第 i+1 行至第 12 行的所有数组元素上移一行，最后一行变为全 0（这里把最上面一行看做第 0 行
        if sum(now[i])==0 and now[i:].sum()!=0:    #本行全是 0，本行以下有不为 0 的元素
            now[i:-1]=now[i+1:]
            now[-1]=0
            time.sleep(0.1)
            return None
    for i in range(h):    #消除整行及记分
        if sum(now[i])==w:
```

```
            score+=1
            now[i]=0
            print(score)
```

5. 定义方块落地的判定与落地前的移动

```
def move():
    global se
    se.sort()
    h,w=now.shape
    top=set(se)-set([i for i in se for j in se if i-j ==100])
    #取补集操作，第一个 set 为所有选中格子的位置，第二个 set 为所有“同列”方块中不在，
    最上的位置，取补集后即为“同列”方块中最上的位置
    def land():       #检测落地与否
        for i in top:
            if i>w and now[i//100-1,i%100]==1 or i<w:
                #i>w and now[i//100-1,i%100]==1 意味着该位置在游戏界面内（第二行及
                  以下），且上方有方块
                #i<w 意味着该位置已经在游戏界面第一行，两者满足其一即应该着地
                return True
        return False
    if not land():        #移动一步
        for i in se:
            now[i//100-1,i%100]=1
            now[i//100,i%100]=0
             #i 代表位置的上方一格变为“有方块”，而 i 自身代表的位置变为“无方块”，即
             往上平移一格
        for i in range(len(se)):
            se[i]-=100
             #se 中的“位置数”也相应调整
        time.sleep(0.1)
    else:
        se=[]
    if not se:   #空 list 意味着 false，不空为 true
        clear()
```

6. 定义鼠标画方块的过程，亦是主程序所在位置

```
def main():
    global select,se
    pos=[0,0]
    while True:
        screen.fill((204,204,204))#刷新 204 度白色背景
```

```
move()

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        return 0
        #如果点击窗口的X则退出窗口
    if event.type == MOUSEMOTION: #鼠标移动
        pos = pygame.mouse.get_pos()  #得到鼠标新坐标


     elif event.type == MOUSEBUTTONDOWN:#鼠标按下
        pressed_array = pygame.mouse.get_pressed()
         #返回一个由布尔值组成的列表，代表所有鼠标按键被按下的情况。True 意
         味着在调用此方法时该鼠标按键正被按下。
        if pressed_array[0]:#按下鼠标左键
            if pos[1]>height-120:      #在黑线方格区域内
                select.append(pos[1]//30*100+pos[0]//30)
                 #append 里面的数（即 loc）是个四位数，千位百位为鼠标所在
                 行数，十位个位为所在列数
    if pygame.mouse.get_pressed():
        if pos[1]>height-120 and len(select)<4:
            loc=pos[1]//30*100+pos[0]//30
            if loc not in select:
                for i in select:
                    if abs(i-loc)==100 or abs(i-loc)==1:
                     #即上下左右四个格子内
                        select.append(loc)
                        break
    if event.type == MOUSEBUTTONUP:
        if len(select)==4:
            for i in select:
                now[i//100,i%100]=1

            se=select[:]
        select=[]

for i in range(now.shape[0]):
    for j in  range(now.shape[1]):

         pygame.draw.rect(screen,(204,204*(1-now[i,j]),204),(j*30,i*30,30,3
         0),0)
         #第二个参数是线条（或填充）的颜色，第三个参数 Rect 的形式是 ((x, y),
         (width, height))，表示的是所绘制矩形的区域，其中第一个元组(x, y)表
```

```
                示的是该矩形左上角的坐标，第二个元组 (width, height)表示的是矩形的
                宽度和高度。width 表示线条的粗细，单位为像素；默认值为 0，表示填充矩
                形内部。(204,204,204)为淡灰色，(204,0,204)为淡紫色
        for i in select:
            pygame.draw.rect(screen,(204,0,204),((i%100)*30,(i//100)*30,30,30),0)
        for i in range(15,20):
            pygame.draw.line(screen,(0,0,0),(0,i*30),(width,i*30),1)
            #线条两端自然结束，没有明显的端点（如实心黑点）。
        for i in range(1,12):
            pygame.draw.line(screen,(0,0,0),(i*30,450),(i*30,height),1)
             #这两个循环在画下方横竖黑线
        pygame.display.flip()#刷新屏幕

main()
```

四、 结论

此次的项目利用 python 的 pygame 模块以及 numpy 模块实现了对传统游戏项目俄罗斯方块的实现和创新，为其增添了新玩法。但我们仍有更多的想法与创新点，希望能在未来的研究中对其实行逐步的改进与实装。

五、 展望

1. 引入 OPENCV 模块，将鼠标操作变为手指操作

   目前方块的描绘由鼠标的移动来完成，期望可以结合电脑摄像头，引入 OPENCV 模块，通过捕捉玩家手指的移动轨迹来描绘方块，增加游戏的互动性与趣味性。

2. 控制方块生成，增加关卡设计
   目前关卡内的方块为随机生成，期望以后可以通过计算后设置方块的生成位置，由此可以做到更精细巧妙的关卡设计，为游戏增添趣味性

3. 提高游戏难度，设置步数限制

   在运行程序时，玩家可以看到方块的绘画次数，期望可以为玩家设置步数限制，只要玩家可以在特定步数内过关，便可获得一定奖励(如游戏内代币，奖牌)，增加玩家过关后的满足感以及对游戏的黏性。