小蛇快跑——反应力小游戏

作者：王其钲 赵梓赫

这是一个操纵长度为2的蛇躲避障碍物的游戏。从传统游戏贪吃蛇中得到灵感，本游戏中蛇的长度不再发生变化，且恒定为2，玩家需要通过单片机上的两个键来控制蛇的转向躲避地图上生成的障碍。地图上的障碍每隔一段时间就会随机生成一次，一旦玩家的蛇碰到了障碍，则宣告游戏失败。

代码实现

- 蛇的行为 `class snake`

```
class snake:
    def __init__(self,x,y,a,b):
        self.location1=(x,y)
        self.location2=(a,b)
    def lit(self):
        alist=[["0","0","0","0","0"],["0","0","0","0","0"]\
               ,["0","0","0","0","0"]\
              ,["0","0","0","0","0"],["0","0","0","0","0"]]
        x,y=self.location1
        a,b=self.location2
        alist[x % 5][y % 5]="9"
        alist[a % 5][b % 5]="9"
        return alist
    def right(self):
        x,y=self.location1
        a,b=self.location2
        x,y=a,b
        a,b=a,b+1
        self.location1=(x,y)
        self.location2=(a,b)
    def down(self):
        x,y=self.location1
        a,b=self.location2
        x,y=a,b
        a,b=a+1,b
        self.location1=(x,y)
        self.location2=(a,b)
    def isright(self):
        x,y=self.location1
        a,b=self.location2
        return x==a and y==b-1
    def isdown(self):
        x,y=self.location1
        a,b=self.location2
```

```
        return x==a-1 and y==b ```
```

这一部分定义了一个蛇的类，通过一个列表来表示25个点中亮的部分来显示这个长度为2的蛇。通过 `lit` 函数来使蛇亮起来，并将亮度设置为最大以让蛇更加明显，与后边定义的障碍物所区别。接着使用了 `right` 和 `left` 函数来操纵蛇的转向，分别使用A，B键控制。最后是状态函数 `isright` 和 `isleft` 来表示蛇当前是横着还是竖着。

```
class trap:
    def __init__(self,dif):#难度参数
        self.difficulty = dif
    def map(self,snakelist):#随机生成，返回一个列表
        maplist=[["0","0","0","0","0"],["0","0","0","0","0"]\
               ,["0","0","0","0","0"]\
              ,["0","0","0","0","0"],["0","0","0","0","0"]]
        for i in range(5):
            for j in range(self.difficulty):
                p = random.randint(0,4)
                if snakelist[i][p] == '0':
                    maplist[i][p] = '5'

        return maplist
```

这一部分通过定义类 `map` 来为棋盘随机生成陷阱。

```
def showmap(a,b):#将map和snake一起显示
    maplist = [["0","0","0","0","0"],["0","0","0","0","0"]\
               ,["0","0","0","0","0"]\
              ,["0","0","0","0","0"],["0","0","0","0","0"]]
    for i in range(5):
        for j in range(5):
            maplist[i][j] = str(int(a[i][j]) +int(b[i][j]))
            if int(maplist[i][j]) > 9:
                maplist[i][j] = '9'
    mapstr = "".join(maplist[0])+":"+"".join(maplist[1])+\
                    ":"+"".join(maplist[2])+":"+"".join(maplist[3])+\
                    ":"+"".join(maplist[4])
    return mapstr
while True:
    display.show('press A')
    if button_a.was_pressed():
        break
display.show(Image('90000:'
                   '00000:'
                   '00000:'
                   '00000:'
                   '00000'))
while True:
    if button_a.was_pressed():
        asnake=snake(0,0,0,1)#按钮a代表向右
```

```
        asnake.lit()
        break
    elif button_b.was_pressed():
        asnake=snake(0,0,1,0)#按钮b代表向下
        asnake.lit()
        break
x = 1
difficulty = 1
time = 1000
```

这一部分是游戏的初始化，先是设置将陷阱和蛇一起显示，然后设置初始动画，接着设置按下A或B键来开始游戏，其中按下A键开始时，蛇直接向右运动，而按下B键时直接向左运动。

```
def attack(oldlist,newlist):#蛇与障碍重合时结束游戏
    for row in range(5):
        for col in range(5):
            if oldlist[row][col]!= '0' and \
            newlist[row][col] != '0':
                return True
    else:
        return False
```

这一部分是定义了 `attack` 函数来判定蛇是否与陷阱重叠。

```
while True:
    tra = trap(difficulty)
    if button_a.was_pressed() and asnake.isdown()\
    or button_b.was_pressed() and asnake.isright():
        if asnake.isdown():
            asnake.right()
            lit = asnake.lit()
            if x >0:#new_set a map
                map = tra.map(asnake.lit())
            display.show(Image(showmap(lit,map)))
            sleep(time)
        else:
            asnake.down()
            lit = asnake.lit()
            if x >0:#new_set a map
                map = tra.map(asnake.lit())
            display.show(Image(showmap(lit,map)))
            sleep(time)
    else:
        if asnake.isright():
            asnake.right()
            lit = asnake.lit()
            if x >0:#new_set a map
                map = tra.map(asnake.lit())
            display.show(Image(showmap(lit,map)))
            sleep(time)
        elif asnake.isdown():
```

```
            asnake.down()
            lit = asnake.lit()
            if x >0:#new_set a map
                map = tra.map(asnake.lit())
            display.show(Image(showmap(lit,map)))
            sleep(time)
    if attack(lit,map):
        display.show(Image(showmap(lit,map)))
        sleep(2000)
        break
    if pin_logo.is_touched():#加多
        difficulty += 1
    if pin_logo.is_touched() and button_b.was_pressed():#加速（电脑模拟暂时不可用）
        time -= 100
    x = -x

display.show('lose')
while True:#结束
    if pin_logo.is_touched():
        display.show(Image.HAPPY)
```

这一部分是游戏进行的主体部分，其中除了实现前边的函数以外，还增加了按住logo键的操作，当按下logo键时，意味着难度系数增加，陷阱将变多以及蛇的速度将变快。

分工合作

本次大作业中，赵梓赫负责完成代码中的蛇类及其移动，其余部分包括陷阱类，游戏主体等由王其钲完成。此外，本文档由赵梓赫完成，介绍视频由王其钲录制。