

## 选5、数组的应用-五子棋

### 1、项目描述

利用python模拟五子棋游戏，共有10\*10的棋盘格子，通过输入行号和列号交替着下白子和黑子。直到一方在行、列、对角上率先达到连续的5颗，则提示哪方获胜。

通过数组qp来表示棋盘格子，qp的初始值都为-1，表示没有落子，0表示白子，1表示黑子。

### 2、定义数组

在Python中定义二维数组有个问题，如果通过qp=[[-1]\*10]\*10来定义，则因为浅拷贝的存在，修改某一个元素时，所有行的同一列元素都会被修改。因此需要通过列表生成式来定义：

```
qp=[[0 for i in range(10)] for j in range(10)]
```

另外需要定义当前下子的颜色，通过变量role来定义。

```
#五子棋
qp=[[-1 for i in range(10)] for j in range(10)]
role=0
qp
```

```
[[ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
 [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]]
```

### 3、编写判断一方是否获胜的函数

在每次落子后，都需要判断一方是否获胜，因此考虑模块化设计，通过编写函数win来实现获胜的判断。

实现判断的基本思想可以是，对于10\*10的棋盘格，需要遍历第0-5行、第0-5列中所有的棋子，对于该范围内的任意棋子，需要去判断向右、向下、向右上角、向右下角四个方向上是否有5颗相同的棋子。

为了进一步拆分问题，先编写函数is\_five来判断以某个位置为基准，在4个方向上是否有5颗相同的棋子：

```

def is_five(r,c,n):
    global qp
    #left
    ct=0
    for i in range(5):
        if c+i>9:
            break
        if qp[r][c+i]==n:
            ct+=1
    if ct==5:
        return True
    #down
    ct=0
    for i in range(5):
        if r+i>9:
            break
        if qp[r+i][c]==n:
            ct+=1
    if ct==5:
        return True
    #rise
    ct=0
    for i in range(5):
        if r-i<0 or c+i>9:
            break
        if qp[r-i][c+i]==n:
            ct+=1
    if ct==5:
        return True
    #fall
    ct=0
    for i in range(5):
        if r+i>9 or c+i>9:
            break
        if qp[r+i][c+i]==n:
            ct+=1
    if ct==5:
        return True
    return False

```

其中，r是行号，c是列号，n为需要判断的棋子颜色(0或者1)

在此基础上，实现win函数就比较简单了：

```

def win(n):
    global qp
    for i in range(6):
        for j in range(6):
            if is_five(i,j,role):
                return n
    return -1

```

函数win的参数n为需要判断的棋子颜色，如果存在5颗则返回n值，否则返回-1

#### 4、编写落子并判断是否获胜的主程序代码

需要注意的是，在主程序里，role才是表示颜色的变量。而且每次正常落子后，role需要通过1-role实现0和1的交替。

```

if role==0:
    print("请下白子")
else:
    print("请下黑子")
r,c=map(int,input("请输入下子行 列:").split())
if qp[r][c]==-1:
    qp[r][c]=role
    w=win(role)
    if w==0:
        print("白子获胜,游戏结束")
    elif w==1:
        print("黑子获胜,游戏结束")
    else:
        role=1-role
else:
    print("该处已经有子了")
qp

```

请下白子

请输入下子行 列:4 4

白子获胜

```

[[-1, -1, -1, -1, -1, -1, -1, -1, -1],
 [-1, -1, -1, -1, -1, -1, -1, -1, -1],
 [-1, 1, 1, -1, -1, -1, -1, -1, -1],
 [-1, 1, 0, 1, 1, 0, -1, -1, -1],
 [-1, 0, 0, 0, 0, 0, -1, -1, -1],
 [-1, 1, -1, -1, -1, 0, 1, -1, -1],
 [-1, -1, -1, -1, -1, -1, -1, -1, -1],
 [-1, -1, -1, -1, -1, -1, -1, -1, -1],
 [-1, -1, -1, -1, -1, -1, -1, -1, -1],
 [-1, -1, -1, -1, -1, -1, -1, -1, -1]]

```