

## 任务:

## 内容:

技术官:

完成全套项目练习题

(可选)根据文件"项目常用思路.pdf"的内容, 优化五子棋游戏的页面切换逻辑

其他组员:

完成第一题(其中 isWin()和gameView()不需要完成)

注意: 不要求用图形界面 控制台实现即可

## 截止时间:

项目设计直播启动前

## 提交方式:

将源码打包后提交给组长

命名格式: qq-昵称.zip/rar/7z

如: 198237645-张三.zip

1. 根据以下的[数据设计]和[函数声明], 完成函数实现. [完成五子棋游戏](#)

```
// ----- 数据设计 -----
/*
    棋盘:
        map[i][j]表示坐标(i,j)的值
        0表示空地
        1表示黑子
        2表示白子
    如: map[3][6] = 1 表示(3,6)的位置是黑子
*/
int map[19][19];

// 表示当前回合数 偶数表示黑棋落子 奇数表示白棋落子
// 如: flag = 20 表示当前是第[20]次落子 由黑方落子
int flag;

// ----- 数据设计 -----

// ----- service -----
/*
    负责人: 张三
    功能: 初始化游戏数据
        将棋盘的值初始化为0
        当前回合设为黑棋(flag设为0)
    参数: void
*/
```

```

        返回值: void
    */
    void init();

    /**
     *难点1
     负责人: 张三
     功能: 根据传入的坐标(map对应位置)和flag值 判断落点后是否获胜
     参数:
         x: 当前回合落子的x坐标
         y: 当前回合落子的y坐标
     返回值:
         0表示没有获胜
         1表示黑子胜利
         2表示白子胜利
    */
    int iswin(int x, int y);

    /**
     负责人: 张三
     功能: 在指定位置落子
         如果map[x][y]是空地 则修改map[x][y]的值:改为相应颜色(flag对应颜色)
         否则不操作
     参数:
         x: 当前回合落子的x坐标
         y: 当前回合落子的y坐标
     返回值:
         0表示落子失败 (棋盘已经有子)
         1表示落子成功
    */
    int playerMove(int x, int y);
    // ----- service -----

    // ----- view -----
    /**
     功能: 展示选项, 玩家可以在此选择进入游戏, 进入设置或退出游戏
         进入游戏: 调用游戏界面函数gameview();
         进入设置: 敬请期待...
         退出游戏: 调用exit(0);
    */
    void menuview();

    /**
     负责人: 张三
     功能: 根据map数组 打印游戏棋盘
     参数: void
     返回值: void
    */
    void gameview_ShowMap();

    /**
     负责人: 张三

```

```

    功能：根据flag的值 打印游戏胜利界面 用户可以按任意键回到主菜单
    参数：void
    返回值：void
*/
void winview();

/*
    *难点2
    负责人：张三
    功能：游戏界面整合
        初始化游戏数据(调用函数init())
        while(1){
            打印游戏界面(调用函数gameview_ShowMap())
            接收玩家坐标输入

            落子(调用落子函数playerMove())
            (如果落子失败 重新开始循环)

            判断游戏是否胜利(调用胜利判断函数iswin())
            (如果游戏胜利 调用胜利界面函数 然后结束当前界面)
            切换玩家(修改flag值)
        }
    参数：void
    返回值：void
*/
void gameview();
// ----- view -----

int main()
{
    int testflag = 0;
    //init测试代码
    /*
    init();
    if (flag != 0) {
        printf("init()错误");
        exit(0);
    }
    for (int i = 0; i < 19; i++) {
        for (int j = 0; j < 19; j++) {
            if (map[i][j]) {
                printf("init()错误");
                exit(0);
            }
        }
    }
    printf("init() 测试成功\n");
    testflag++;
    */

    //playerMove测试代码
    /*
    int result = 1;
    result &= playerMove(2, 2);
    result &= playerMove(2, 3);

```

```

        result &= playerMove(2, 4);
        result &= playerMove(2, 5);
        if (result != 1 || (map[2][2] && map[2][3] && map[2][4] && map[2][5]) != 1)
    {
        printf("playerMove()错误");
        exit(0);
    }
    flag = 1;
    result &= playerMove(2, 5);
    if (result != 0 || map[2][5] != 1) {
        printf("playerMove()错误");
        exit(0);
    }
    printf("playerMove()测试成功\n");
    testflag++;
    */

    //iswin测试代码
    /*
    playerMove(2, 1);
    if (iswin(2, 1)) {
        printf("iswin()错误");
        exit(0);
    }
    playerMove(1, 0);
    playerMove(3, 2);
    playerMove(4, 3);
    playerMove(5, 4);
    if (iswin(1, 0) != 2) {
        printf("iswin()错误");
        exit(0);
    }
    printf("iswin()测试成功\n");
    testflag++;
    */

    if (testflag == 3) {
        printf("service代码测试成功\n");
    }
    return 0;
}

void init(){
    //代码实现
}

int iswin(int x, int y){
    //代码实现
}

int playerMove(int x, int y){
    //代码实现
}

void menuview(){

```

```

        //代码实现
    }

    void gameview_ShowMap(){
        //代码实现
    }

    void winView(){
        //代码实现
    }

    void gameview(){
        //代码实现
    }

```

## 2. 根据第一题的函数描述内容, 回答以下问题. 拓展知识点: 三层架构

### 1. 以下内容的代码实现, 由view还是service负责? (填写view或service)

游戏逻辑判断:

用户输入:

用户输入数据合法判断/类型判断(比如区分输入是[方向]还是[空格]):

界面展示:

游戏数据修改:

### 2. 以下说法是否正确? (填写true或false)

1. view有时需要调用service的功能 以实现逻辑判断和数据修改:
2. service有时需要调用view的功能 以实现界面展示:
3. service中可以写scanf和printf 用于接收用户输入和展示界面:
4. 涉及游戏数据判断和修改的, 大多在service实现:
5. 数据的设计不需要描述很清晰, 自己能看懂即可:

## 3. 根据第一题的题目思路, 完成推箱子游戏的[数据设计]和[函数声明]. 不需要写代码实现

该游戏至少要有以下功能:

开始界面: 可以选择: [选关界面]/[设置界面]/[排行榜界面]/退出游戏  
 选关界面: 可以选择关卡进入[游戏界面], 或者回到[开始界面]  
 游戏界面: 可以移动人物, 回退到[选关界面], 或者游戏胜利后展示[胜利界面]  
 胜利界面: 提示游戏胜利, 并回到[选关界面]  
 设置界面: 展示[未完待续...]  
 排行榜界面: 展示[未完待续...]