

1.

C/C++:

```
#pragma warning(disable: 4996)
#include <stdio.h>
#include <stdlib.h>

// ----- 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 则落子相应颜色(flag对应颜色) 否则不落子

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，玩家可以在这里选择进入游戏，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()
{
    menuview();
    return 0;
}

void init() {
    //将map初始化为0
    for (int i = 0; i < 19; i++) for (int j = 0; j < 19; j++) map[i][j] = 0;
    //将flag初始化为0
    flag = 0;
}

int iswin(int x, int y) {
    // 四条线 分别是从上往下 从左上往右下 从左往右 从左下往右上
    int dir[4][4] = {
        //x起始 y起始 x移动方向 y移动方向
        {0,-4,0,1}, //从上往下
        {-4,-4,1,1}, //从左上 往右下
        {-4,0,1,0}, //从左往右
        {-4,4,1,-1} //从左下 往右上
    };
    int count;
    for (int i = 0; i < 4; i++) {
        count = 0;

        //从起始点开始循环 一直循环到终点 也就是从上往下 从左上往右下 从左往右 从左下
        //往右上
        /*
            j和k 分别表示起始点相对的x位置和y位置
            比如从上往下判断 则起始相对位置是(0,-4) x每次+0 y每次+1 循环9次

            map[x+j][y+k] 表示当前判断的位置
            比如要判断(5,6)从上往下(每次循环x+0,y+1)
            则是从(5,6-4)开始往下移动 一路到(5,6+4)

        */
        for (int j = dir[i][0], k = dir[i][1], n = 0;
            n < 9;
            j += dir[i][2], k += dir[i][3], n++) {

            //如果超过棋盘 直接下一次循环
            if (x + j >= 19 || y + k >= 19 || x + j < 0 || y + k < 0) continue;

            //如果是当前玩家的子 则count+1

```

```

        if (map[x + j][y + k] == (flag % 2) + 1) {
            if (count == 4) return (flag % 2) + 1;
            count++;
        }
        //否则 count = 0 重新计数
        else {
            count = 0;
        }
    }
}
return 0;
}

int playerMove(int x, int y) {
    if (map[x][y]) {
        return 0;
    }
    map[x][y] = (flag % 2) + 1;
    return 1;
}

void menuview() {
    int choose = 0;
    while(1){
        printf("输入1进入游戏界面\n输入2进入设置界面\n输入0退出游戏\n");
        scanf("%d", &choose);

        switch (choose) {
            case 1:
                gameview();
                break;
            case 2:
                printf("敬请期待...\n");
                break;
            case 0:
                exit(0);
        }
    }
}

/*
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)

```
    }
*/
void gameview() {
    init();
    int x = 0, y = 0;
    while (1) {
        gameview_ShowMap();
        // 三元运算符 a ? b : c
        printf(flag % 2 == 0 ? "黑棋落子" : "白棋落子");
        printf("请输入xy坐标(空格分隔 如: 3 6):");
        scanf("%d %d", &x, &y);

        //c语言中非0表示真    !是取反
        //于是这里返回0为真    返回非0为假
        if (!playerMove(x, y)) {
            printf("这个点不能落子, 请重新输入\n");
            continue;
        }

        if (iswin(x, y)) {
            gameview_ShowMap();
            winView();
            return;
        }

        flag++;
    }
}

/*
    负责人: 张三
    功能: 根据map变量 打印游戏棋盘
    参数: void
    返回值: void
*/
void gameview_ShowMap() {
    for (int i = 0; i < 19; i++) {
        for (int j = 0; j < 19; j++) {
            printf(map[i][j] == 0 ? "+" : map[i][j] == 1 ? "龍" : "口");
        }
        printf("\n");
    }
}

/*
    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() {
```

```
printf((flag % 2) == 0 ? "黑子胜利\n" : "白子胜利\n");
}
```

Java:

```
package view;

import java.util.Scanner;
import java.util.Arrays;
class ChessDao {

    /**
     * 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
     */
    public static int[][] map = new int[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
    public static int flag;

}

class ChessService {

    /**
     * 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
     */
    public void init() {
        for (int i = 0; i < ChessDao.map.length; i++) {
            Arrays.fill(ChessDao.map[i], 0);
        }
    }

    /**
     * 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
     * @param x 当前回合落子的x坐标
     * @param y 当前回合落子的y坐标
     * @return 0表示没有获胜
     * 1表示黑子胜利
     * 2表示白子胜利
     */
    public int iswin(int x, int y){
```

```

int color = ChessDao.flag % 2 + 1;
int left = x < 4 ? -x : -4;
int right = x + 4 > ChessDao.map[0].length - 1 ? ChessDao.map[0].length
- x - 1 : 4;
int top = y < 4 ? -y : -4;
int blow = y + 4 > ChessDao.map.length - 1 ? ChessDao.map.length - y - 1
: 4;

for (int i = 0; i < 4; i++) {
    int num = 1;
    switch (i) {
        case 0:
            for (int j = left; j <= right && num != 5; j++) {
                if (ChessDao.map[y][x + j] == color) {
                    num++;
                } else {
                    num = 0;
                }
            }
            break;
        case 1:
            for (int j = top; j <= blow && num != 5; j++) {
                if (ChessDao.map[y + j][x] == color) {
                    num++;
                } else {
                    num = 0;
                }
            }
            break;
        case 2:
            for (int j = left, k = top; j <= right && k <= blow && num
!= 5; j++, k++) {
                if (ChessDao.map[y + k][x + j] == color) {
                    num++;
                } else {
                    num = 0;
                }
            }
            break;
        case 3:
            for (int j = right, k = top; j >= left && k <= blow && num
!= 5; j--, k++) {
                if (ChessDao.map[y + k][x + j] == color) {
                    num++;
                } else {
                    num = 0;
                }
            }
            break;
    }
    if (num == 5) {
        return color;
    }
}

```

```

        return 0;
    }

    /**
     *
     * 在指定位置落子 如果map[x][y]是空地 则落子相应颜色(flag对应颜色) 否则不落子
     * @param x 当前回合落子的x坐标
     * @param y 当前回合落子的y坐标
     * @return 0表示落子失败 (棋盘已经有子) 1表示落子成功
     */
    public int playerMove(int x, int y){
        if (ChessDao.map[y][x] != 0) {
            return 0;
        }
        ChessDao.map[y][x] = ChessDao.flag % 2 == 0 ? 1 : 2;
        return 1;
    }
}

public class Chessview {

    ChessService service = new ChessService();
    Scanner scanner = new Scanner(System.in);

    /**
     * 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);
     */
    public void menuView() {
        System.out.println("1.进入游戏");
        System.out.println("2.进入设置");
        System.out.println("3.退出游戏");
        System.out.print("请选择: ");
        int input = scanner.nextInt();
        while (input < 1 || input > 3) {
            System.out.print("输入有误，请重新选择: ");
            input = scanner.nextInt();
        }
        switch (input) {
            case 1: gameView();break;
            case 2:
                System.out.println();
                System.out.print("| 敬请期待... |");
                System.out.println();
                break;
            case 3: System.exit(0);break;
        }
    }
}

```



```

}

/**
 * 功能：根据map数组 打印游戏棋盘
 * 参数：void
 * 返回值：void
 */
public void gameview_ShowMap() {
    for (int i = 0; i < ChessDao.map.length; i++) {

        for (int j = 0; j < ChessDao.map[i].length; j++) {
            System.out.print(" ---");
        }

        System.out.print("\n| ");
        for (int j = 0; j < ChessDao.map[i].length; j++) {
            if (ChessDao.map[i][j] != 0) {
                System.out.printf("%c ", ChessDao.map[i][j] == 1 ? 'X' :
'o');
            }else {
                System.out.print(" ");
            }
            System.out.print("| ");
        }
        System.out.println();
    }
    for (int j = 0; j < ChessDao.map[0].length; j++) {
        System.out.print(" ---");
    }
    System.out.println();
}

/**
 * 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
 */
public void winview() {
    System.out.println("恭喜获胜! ");
    System.out.println("输入任意键返回主菜单");
    if (scanner.nextLine() != null) {
        menuview();
    }
}

/**
 * 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
*/
public void gameView() {
    service.init();
    while (true) {
        gameView_ShowMap();
        if (ChessDao.flag % 2 == 0) {
            System.out.println("轮到白子下棋");
        }else{
            System.out.println("轮到黑子下棋");
        }
        System.out.print("请输入坐标: ");
        int x = scanner.nextInt();
        int y = scanner.nextInt();
        if (service.playerMove(x, y) == 0) {
            continue;
        }

        int win = service.iswin(x, y);

        if (win == 0) {
            ChessDao.flag++;
        }else {
            winview();
            break;
        }
    }
}

public static void main(String[] args) {
    //gameView_ShowMap();
    new ChessView().menuView();
}
}

```

2.

1. service, view, view, view, service

service负责游戏数据相关的内容 如碰撞检测, 移动, 胜利判断等等

view负责与用户交互相关 如接收输入 判断输入类型 输出等等

2. 1. true: view需要通过调用service来修改/判断游戏逻辑
2. false: 从三层架构的思路来讲, service不能反过来调用view 通常只能是view调用service
3. false: view负责界面交互: 如接收用户输入, 将界面展示给用户等等
4. true: service负责逻辑代码: 如判断/修改数据等等
5. false: 团队项目需要每个人都理解[数据的设计] 和自己需要负责的[函数思路].

3.

C/C++:

```
// ----- 数据设计 -----
/*
    当前关卡的地图：
        map[i][j]表示坐标(i,j)的物品
        0表示空地
        1表示墙
        2表示玩家
        3表示箱子
        4表示目标点(空)
        5表示目标点(箱子)
        6表示目标点(玩家)

    如：map[3][6] = 1 表示(3,6)的位置是墙
*/
int map[10][10];

/*
    游戏默认地图 在玩家选关后 将mapData[关卡]的值 赋予map
*/
int mapData[9][10][10];

// ----- 数据设计 -----

// ----- service -----
/*
    负责人：张三
    功能：初始化游戏数据
        根据关卡号 初始化地图
    参数：void
    返回值：void
*/
void init(int level);

/*
    负责人：张三
    功能：根据当前的map 判断游戏是否胜利
        如果当前地图没有3和4（单独的箱子和单独的目标点）说明游戏胜利

    参数:void
    返回值：
        0表示没有获胜
        1表示胜利
*/
int iswin();

/*
    负责人：张三
    功能：玩家朝指定方向移动一格
        如果玩家前面是箱子 则箱子也跟着移动一格
            如果箱子前面有[箱子]或[墙]或[有箱子的终点] 则无法移动
        如果玩家前面是墙 无法移动
*/
```

参数：

input: 移动的方向

返回值：

0表示移动失败（撞墙 或者推的箱子撞墙）

1表示移动成功

```
*/
int move(char input);
// ----- service -----

// ----- view -----
/*
    功能：展示选项，玩家可以在这里选择进入游戏，进入设置，进入排行榜或退出游戏
    选关界面：调用选关界面函数levelSelectView();
    进入设置：敬请期待...
    进入排行榜：敬请期待...
    退出游戏：调用exit(0);
*/
void menuView();

/*
    负责人：张三
    功能：
        展示关卡选项
        让用户输入关卡号
        进入对应游戏界面(调用gameView(int level))
    参数：void
    返回值：void
*/
void levelSelectView();

/*
    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(int level))
        while(1){
            打印游戏界面(调用函数gameView_ShowMap())
            接收玩家输入
                如果是q：结束当前界面
                如果是wasd：移动(调用移动函数move(char input))

            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)
        }
    参数：void
    返回值：void
*/
void gameView(int level);

/*
    负责人：张三
    功能：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();
// ----- view -----

int main()
{
    menuview();
    return 0;
}

```

Java:

```

package view;

class BoxDao{
    /*
    当前关卡的地图:
        map[i][j]表示坐标(i,j)的物品
        0表示空地
        1表示墙
        2表示玩家
        3表示箱子
        4表示目标点(空)
        5表示目标点(箱子)
        6表示目标点(玩家)

    如: map[3][6] = 1 表示(3,6)的位置是墙
    */
    public static int map[][];

    /*
    游戏默认地图 在玩家选关后 将mapData[关卡]的值 赋予map
    */
    public static int mapData[][][];
}

class BoxService{
    /*
    负责人: 张三
    功能: 初始化游戏数据
        根据关卡号 初始化地图
    参数: void
    返回值: void
    */
    public void init(int level){

    }

    /*
    负责人: 张三
    功能: 根据当前的map 判断游戏是否胜利
        如果当前地图没有3和4 (单独的箱子和单独的目标点) 说明游戏胜利
    */
}

```

```

        参数: void
        返回值:
            0表示没有获胜
            1表示胜利
    */
    public int iswin(){
        return 0;
    }

    /*
        负责人: 张三
        功能: 玩家朝指定方向移动一格
            如果玩家前面是箱子 则箱子也跟着移动一格
            如果箱子前面有[箱子]或[墙]或[有箱子的终点] 则无法移动
            如果玩家前面是墙 无法移动

        参数:
            input: 移动的方向
        返回值:
            0表示移动失败 (撞墙 或者推的箱子撞墙)
            1表示移动成功
    */

    public int move(char input){
        return 0;
    }
}

public class BoxView {
    /*
        功能: 展示选项, 玩家可以在这里选择进入游戏, 进入设置, 进入排行榜或退出游戏
            选关界面: 调用选关界面函数levelSelectview();
            进入设置: 敬请期待...
            进入排行榜: 敬请期待...
            退出游戏: 调用exit(0);
    */

    public void menuView(){

    }

    /*
        负责人: 张三
        功能:
            展示关卡选项
            让用户输入关卡号
            进入对应游戏界面(调用gameView(int level))
        参数: void
        返回值: void
    */

    public void levelSelectview(){

    }
}

```

```

    /*
        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(int level))
        while(1){
            打印游戏界面(调用函数gameView_ShowMap())
            接收玩家输入
                如果是q：结束当前界面
                如果是wasd：移动(调用移动函数move(char input))

            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)
        }
        参数：void
        返回值：void
    */
    public void gameView(int level){

    }

    /*
        负责人：张三
        功能：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
    */
    public void winView(){

    }

    public static void main(String[] args) {
        new BoxView().menuView();
    }

}

```