《Java 程序设计》实验报告

年级、专业、班级	2019 级计卓 02 班	姓名	李燕琴
实验题目	联网对战俄罗斯方块游戏		
实验时间	2020.11.15-2020.12.15	实验地点	DS401
实验成绩		实验性质	□验证性 □设计性 ■综合性

教师评价：

□ 算法/实验过程正确； □ 源程序/实验内容提交 □ 程序结构/实验步骤合理；

□ 实验结果正确； □ 语法、语义正确； □ 报告规范；

其他：

评价教师签名：

一、实验目的

利用 Java GUI、Java Graphics、多线程和网络技术，编写实现联网对战俄罗斯方块游戏。

二、实验项目内容

1、实现俄罗斯方块的基本游戏逻辑。

2、利用 Socket 编写服务器端和客户端程序。

3、实现俄罗斯方块的联网对战功能。

4、实现计分、音效等辅助功能。

三、实验过程或算法（源程序）

1、需求分析

1）实现俄罗斯方块的基本游戏逻辑：如方块的自动下落、操作左右下移动、操作旋转、操作直落、方块消行、方块碰底及碰顶判断。这些操作，是对游戏数据的底层修改，可以将其封装在数据操作层（model 构建）里面，进行统一管理，每当监听（controller 构建）到相应事件输入（view 构建即图形界面和用户进行交互）后，程序调用该数据操作即可。

2）利用 Socket 编写服务器端和客户端程序：根据“三次握手四次挥手”原理编写，用 ObjectInputSream 和 ObjectOutputSream 收发游戏数据层类，以此定义网络通信协议。

报告创建时间：2020/12/15

3）实现俄罗斯方块的联网对战功能：如创建服务端后，客户端连接服务器，传送己方数据给服务端，服务端接收数据并传给另一个客户端，另一个客户端将收到他方数据，这样即可完成数据通信，根据他方数据做成图形界面进行直观对比，达到对战效果。

4）实现计分、音效等辅助功能：可根据消行数来记录分数，音效响应相关按键或点击事件即可。

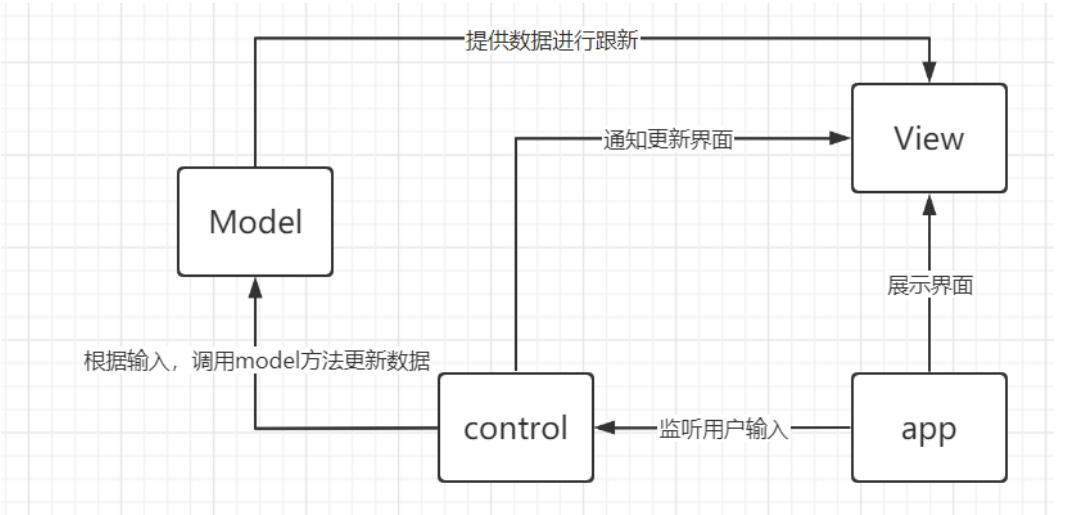
2、系统设计

2.1 设计模式：MVC 框架，即 model-view-controller（模型-视图-控制器），适用于交互编程项目开发。

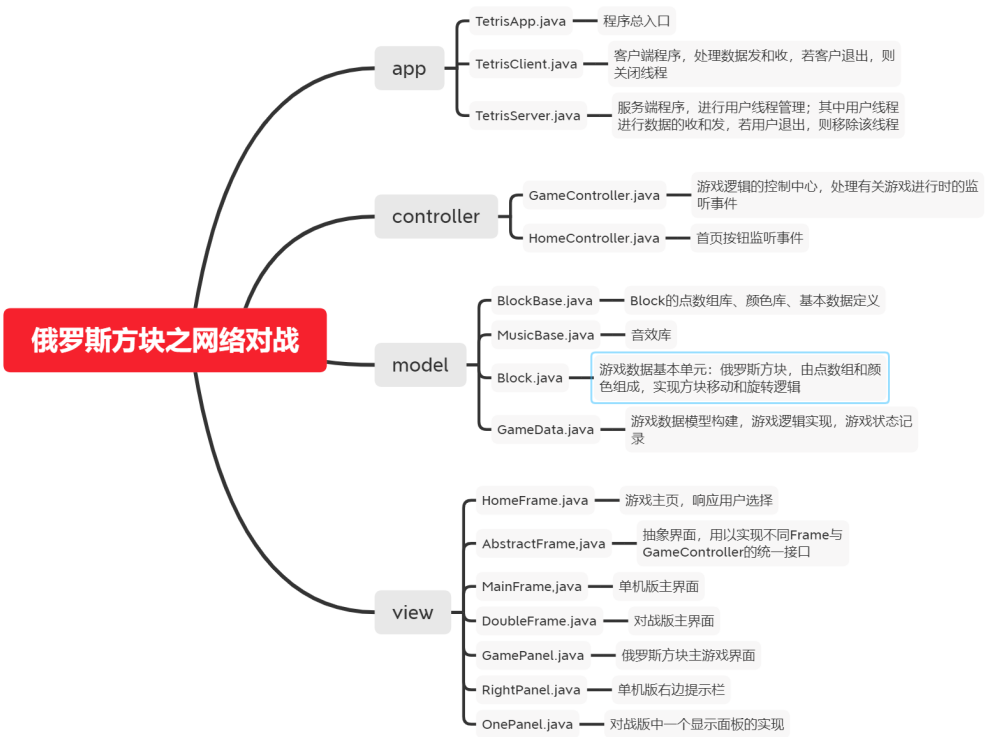
Model：游戏数据模型构建、游戏状态记录、游戏逻辑操作实现

View：游戏界面设计，响应用户操作，并呈现 Model 的操作结果给用户，主要实现数据到页面转换过程。

Control：负责控制 view 和 model 之间的联系，即 control 根据 view 响应的用户操作，调用 model 中的逻辑处理，并将处理结果传给 view 进行显示。



2.2 类的设计：



重要类设计说明

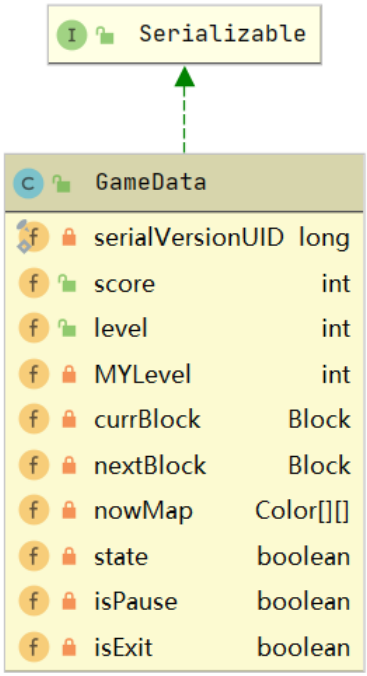
1）游戏核心数据 GameData

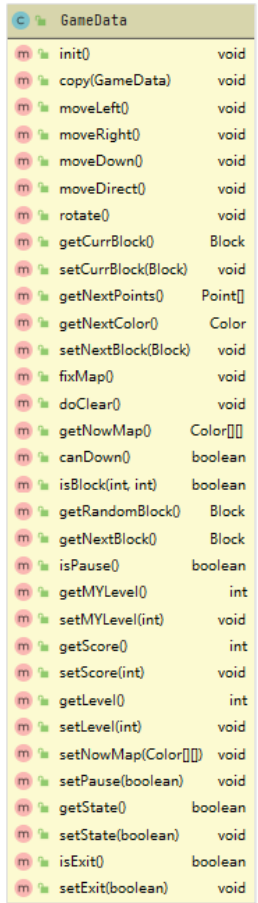
• 构造 GameData(),GameData(GameData)

• 数据模型：score,level,currBlock,nextBlock,nowMap。其中 score 记录消行数即得分；level 随着分数的增加取分数模 MyLevel 结果，以此设置游戏方块下降速度，自动增加难度；nowMap 存储 20*10 的主游戏面板的颜色 Color[][]数据。

•游戏状态：state,isExit,isPause，响应外界事件进行值的变化，state 为游戏是否进行，isExit 为界面是否关闭，isPause 判断游戏是否暂停，是则添加暂停图标。

• 游戏逻辑：move**()，控制 currBlcok 的移动，rotate()控制其旋转，getRandomBlock()获取随机方块更新 nextBlock，fixMap()填充 nowMap，doClear()执行消行操作并记录分数、更新等级，canDown()能否下移的判断，若不能则调用 fixMap()。





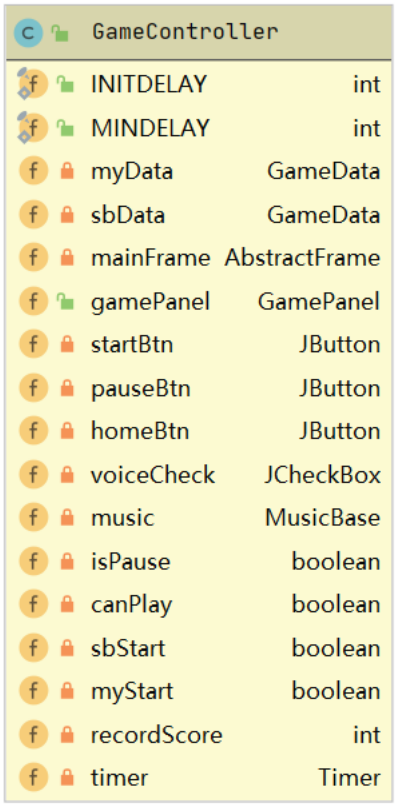
2）游戏核心控制 GameController

• 构造 GameController(GameData,AbstractFrame)

• 游戏数据：myData 记录己方数据；sbData 记录对方数据，如果是单人版则为 null；gamePanel 和 mainFrame 获取监听事件并控制更新界面；recordScore 记录之前的得分对比更新得分，若不同则播放得分音效；music 音效库。

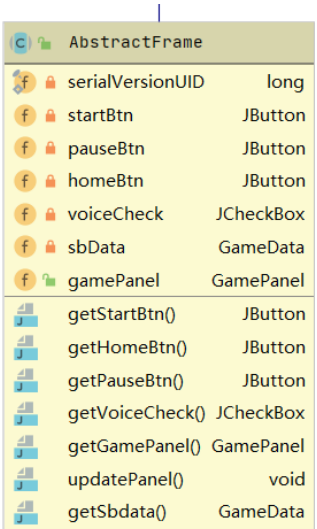
• 游戏状态：myStart 记录己方游戏是否开始，sbData 记录对方游戏是否开始，canPlay 记录是否开启音效，isPause 记录游戏是否暂停。

• 事件监听：timer 为开启后的游戏进程，控制方块自动下落，监听己方或对方游戏是否开始或结束，实时更新界面并辅助 GamePanel 夺回焦点以便键盘控制；键盘监听上（旋转）、下（移动）、Enter（直降）、左右（移动）、空格键（暂停）；startBtn 监听游戏的开始和终止，pauseBtn 监听游戏的暂停和继续；voiceCheck 监听游戏音效使能。



3）抽象界面 `AbstractFrame`

内置数据和功能按钮，统一接口，获取按钮和对方数据，以此构造 `GameController`。



4）网络程序分析

• 服务端

①客户端连接服务器后，将不多于两个的客户端放入 `threads` 中进行数据交换。

```
public TetrisServer(int port) {
    threads = new ArrayList<GameThread>();
    try {
        server = new ServerSocket(port);
        System.out.println("服务器创建成功，等待连接");
        while (true) {
            client = server.accept();
            int num = threads.size();
            GameThread gt = new GameThread(client);
            gt.start();
            if (num == 2) {
                System.out.println("房间人数已满，请等待。。。。");
                continue;
            }
            System.out.println("第" + (num + 1) + "个用户进入房间。");
            threads.add(gt);
        }
    } catch (Exception e) {
        e.printStackTrace();
```

②设置内部线程类，为进入的客户端分配线程，进行数据收发，若用户退出，则关闭该 socket，并设置线程 canRun 为 false，将该线程移除 threads 中。

```
class GameThread extends Thread {
    private Socket s = null;
    private ObjectInputStream ois = null;
    private ObjectOutputStream oos = null;
    // 确保其他程序使用的是最新的值
    private volatile boolean canRun = true;

    public GameThread(Socket s) throws Exception {
        super();
        // 利用线程进行通信
        this.s = s;
        // NOTE 服务器先读后写，还需要先定义oos,再定义ois
        oos = new ObjectOutputStream(s.getOutputStream());
        ois = new ObjectInputStream(s.getInputStream());
    }

    @Override
    public void run() {
        try {
            while (canRun) {
                // DBUG
                // System.out.println("server");
                GameData dataFromClient = (GameData) ois.readObject();
                if (dataFromClient.isExit()) {
                    ois.close();
                    oos.close();
                    System.out.println("线程退出，并移除一个客户端");
                    threads.remove(this);
                    s.close();
                    this.canRun = false;
                    return;
                }
                for (GameThread gt : threads) {
                    // 这里的oos是调用的arraylist里面的oos哦，不是this的oos
                    // 给非我的其他客户端写数据
                    if (gt != this) {
                        gt.oos.writeObject(dataFromClient);
                        gt.oos.flush();
                        gt.oos.reset();
                    }
                }
                Thread.sleep(5000);
            }
        } catch (Exception e) {
```

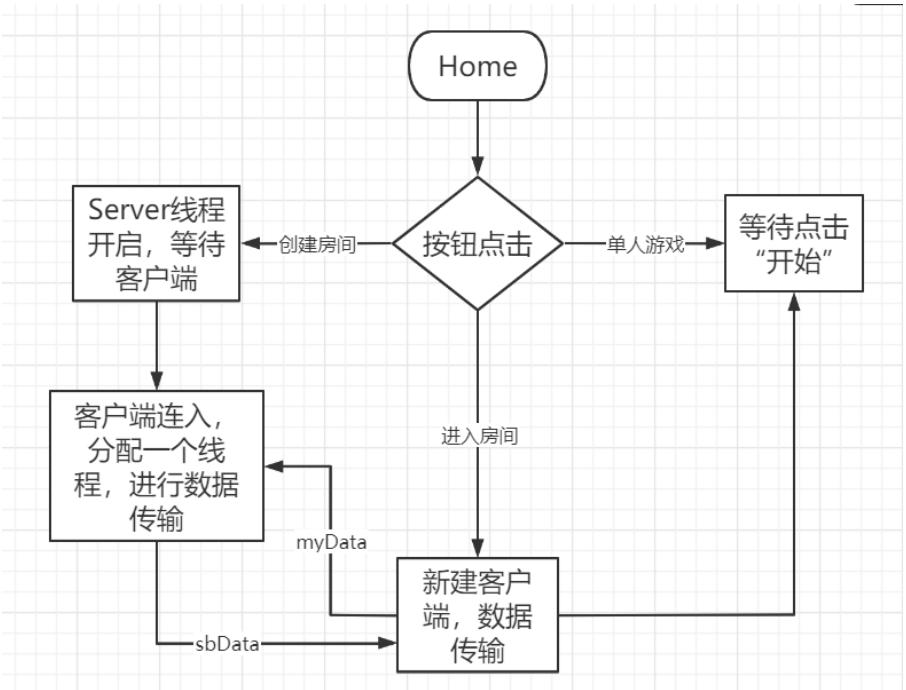
- 客户端

连接服务器后，开始发收数据，若用户退出，则关闭该 socket。

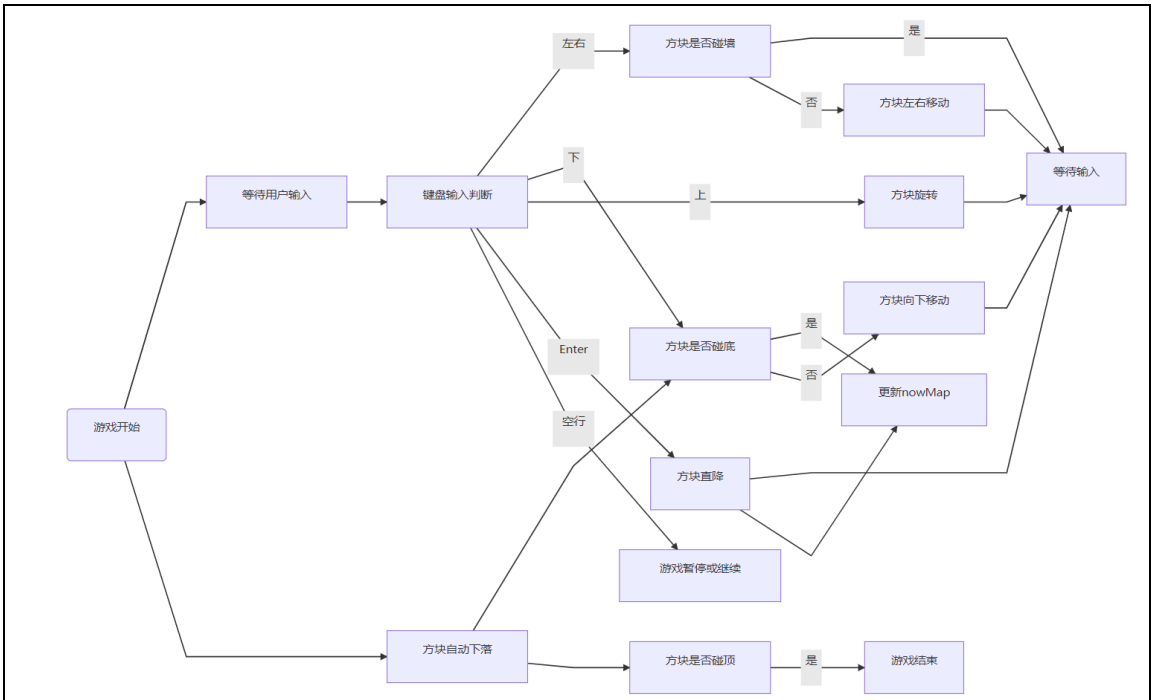
```java
public TetrisClient(int port, GameData mydata, GameData sbdata) {
    try {
        client = new Socket("localhost", port);
        System.out.println("成功连接服务器");
        ois = new ObjectInputStream(client.getInputStream());
        oos = new ObjectOutputStream(client.getOutputStream());
        // 客户端先写后读
        while (true) {
            // DBUG
            // System.out.println("\tclient");
            oos.writeObject(mydata);
            oos.flush();
            // 重置文件头
            oos.reset();
            if (mydata.isExit()) {
                System.out.println("客户端退出了");
                // sbdata.init();
                sbdata.setState(false);
                oos.close();
                ois.close();
                client.close();
                return;
            }
            GameData dataFromServer = (GameData) ois.readObject();
            // NOTE 不能直接使用"="，否则会产生一个新的对象，sbdata的指向地址会被更改，
            // 导致sbdata不能在原地址下进行修改，只能调用原对象的方法进行赋值，才能在原地址上修
            sbdata.copy(dataFromServer);
```

3、系统实现

HomeController 按钮点击控制游戏



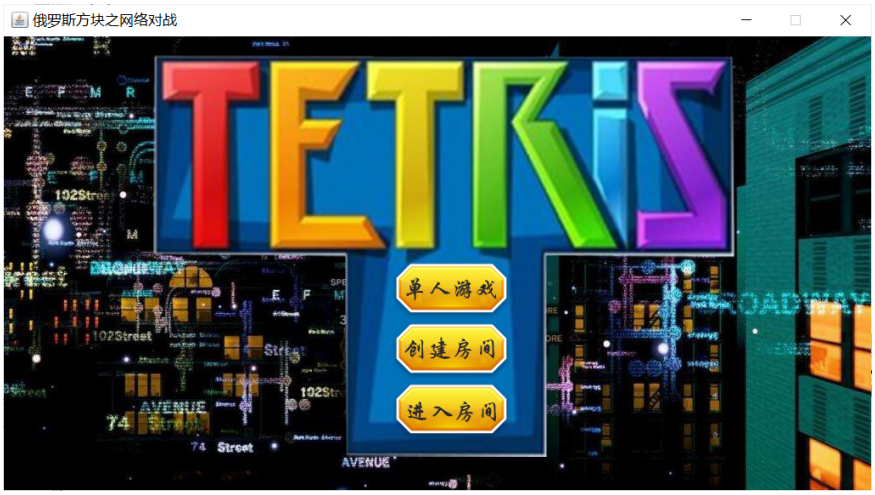
GameController 键盘输入控制线



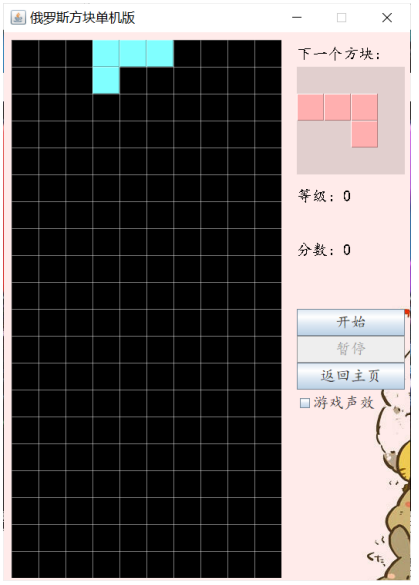
四、实验结果及分析和（或）源程序调试过程

1、程序界面展示

HomeFrame



MainFrame(单机版)



DoubleFrame(对战版)

具体运行见附件视频。

2、问题记录

1）GamePanel 的焦点获取

运行程序后，GamePanel 无法实现键盘监听：按钮类一直占据屏幕焦点，需要使 GamePabel 夺回屏幕焦点，此时通过 GameController 中的 timer，辅助其获取屏幕焦点。

```
if (!gamePanel.isFocusable()) {
    gamePanel.setFocusable(true);
}
if (!gamePanel.isFocusOwner()) {
    gamePanel.requestFocus(true);
}
```

2）碰底和碰墙逻辑：最开始陷入的误区是碰底和碰墙，一起用一个 if 或语句判断，

但是碰底需要额外的更新nowMap，一起判断容易造成逻辑失当。只通过一个canDown()来判断碰底逻辑，碰墙比较简单就不用单独的方法判断。

```
public void moveRight() {
    for (Point p : currBlock.points) {
        if (p.x + 1 >= GamePanel.COLS || isBlock(p.x + 1, p.y)) {
            return;
        }                          碰墙+碰到Block
    }
    currBlock.moveRight();
}

public boolean canDown() {
    for (Point p : currBlock.points) {
        if (p.y + 1 >= GamePanel.ROWS || isBlock(p.x, p.y + 1)) {
            fixMap();
            return false;          碰底+碰到Block
        }
    }
    return true;
}
```

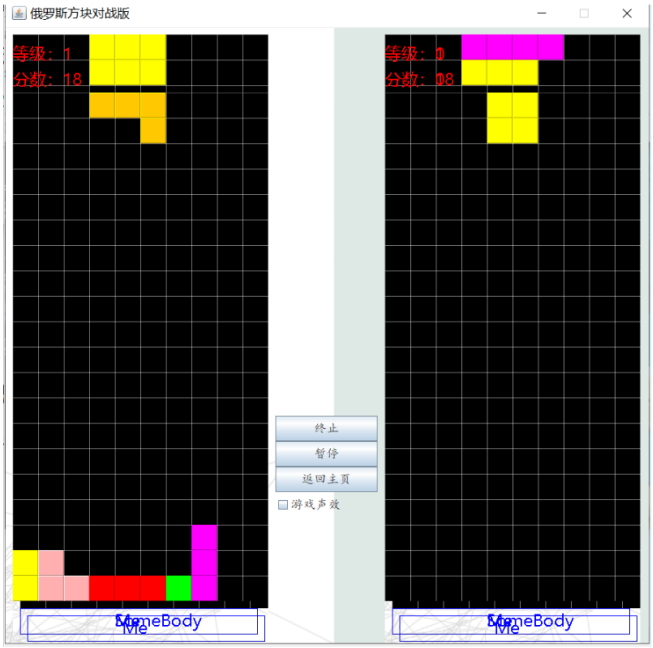
3）随机数的产生：

```
// 每次都是0
shapeIndex = ((int)Math.random()*10000) % BlockBase.NUM;

// 成功解决
Random r = new Random();
shapeIndex = Math.abs(r.nextInt()) % 7 + 1;
```

4）定点添加的 Panel 未能显示出来：检查 Panel 附属的容器的 Layout 是否设置为 null。

5)Panel 重绘异常：未能继承之前的图，需要加上 super.paint(g)，在原图像的基础上画图，否则重绘时会将原有的绘制清空，在根据自己设定的容器坐标进行绘制，导致重绘异常。



5）网络对战数据传输

①传输数据后，对方的界面 sbPanel 没有变化：传输数据时，不能直接写成

```
sbdata = (GameData) ois.readObject();
```

“=”产生新的对象，使 sbdata 指向发生改变，而没有实质性地改变原地址的那个 sbdata 的值。解决：需要调用 sbdata 内部的方法进行复制赋值。

```
GameData dataFromServer = (GameData) ois.readObject();
// NOTE 不能直接使用"="，否则会产生一个新的对象，sbdata的指向地址会被更改，
// 导致sbdata不能在原地址下进行修改，只能调用原对象的方法进行赋值，才能在原地址上修改
sbdata.copy(dataFromServer);
```

```
public void copy(GameData data) {
    this.currBlock = new Block(data.getCurrBlock());
    this.nextBlock = new Block(data.getNextBlock());
    this.score = data.score;
    level = data.level;
    state = data.getState();
    isPause = data.isPause;
    Color[][] dataMap = data.getNowMap();
    for (int i = 0; i < nowMap.length; i++) {
        for (int j = 0; j < nowMap[i].length; j++) {
            this.nowMap[i][j] = dataMap[i][j];
        }
    }
}
```

②客户端传给服务器的数据，服务器每次收到的数据和第一次传过去数据一样。ObjectOutputStream 会自动添加发送的 Object 的头尾信息，且再次读入数据的时候，会在其后继续添加数据和头尾信息。而每当服务器读入时，读到第一个尾信息便停止读入，即每次都只收到第一次写入的数据。解决：客户端和服务器的 oos 每次写完数据边重置一次，即 oos.reset()。

3、项目优点

1）符合 MVC 架构，整个游戏的逻辑比较清晰。

2）各个类的封装性比较好，设置新增功能比较快捷

3）根据分数自动增加游戏难度（level），增加了游戏的有趣度。

4、项目缺点或改进

1）未能实现的线程逻辑：当第三个人想要加入游戏，只有等到第二个人退出游戏再连接服务器才行，不能在其退出之前加入游戏，否则会错过进入 threads 的机会，源自用 Arraylist 来管理线程的 bug，有必要时应当使用同步锁处理此逻辑。

2）当双人对战时，第二个用户进入房间时，需要手动使能“开始”，无法自动开始。

备注：

1、教师在布置需撰写实验报告的实验前，应先将报告书上的“实验题目”、“实验性质”、“实验目的”、“实验项目内容”等项目填写完成，然后再下发给学生。

2、教师在布置需撰写报告的实验项目时，应告知学生提交实验报告的最后期限。

3、学生应按照要求正确地撰写实验报告：

1) 在实验报告上正确地填写“实验时间”、“实验地点”等栏目。

2) 将实验所涉及的源程序文件内容（实验操作步骤或者算法）填写在**“实验过程或算法（源程序）”**栏目中。

3) 将实验所涉及源程序调试过程（输入数据和输出结果）或者实验的分析内容填写在**“实验结果及分析和（或）源程序调试过程”**栏目中。

4) 在实验报告页脚的“报告创建时间：”处插入完成实验报告时的日期和时间。

5) 学生将每个实验完成后，按实验要求的文件名通过网络提交（上载）到指定的服务器所规定的共享文件夹中。每个实验一个电子文档，如果实验中有多个电子文档（如源程序或图形等），则用 WinRAR 压缩成一个压缩包文档提交，压缩包文件名同实验报告文件名（见下条）。

6) 提交的实验报告电子文档命名为：“组号（2 位数字）年级（两位数字不要“级”字）专业（缩写：计算机科学与技术专业（计科）、网络工程专业（网络）、信息安全专业（信息）、物联网工程（物联网））项目组成员（学号（八位数字）姓名）实验序号（一位数字）. doc。如第 1 组完成第 1 个 Project，专业为“计算机科学与技术”专业，项目组成员有：张三（学号 20115676），李四（学号 20115676），王五（学号 20115676），完成的课程设计报告命名为：01_10 计科_20115676 张三

_20115676 李四_20115676 王五 1.doc，以后几次实验的报告名称以此类推。

4、教师（或助教）在评价学生实验时，应根据其提交的其他实验相关资料（例如源程序文件等）对实验报告进行仔细评价。评价后应完成的项目有:

1) 在“成绩”栏中填写实验成绩。每个项目的实验成绩按照五级制（优、良、中、及格、不及格）方式评分，实验总成绩则通过计算每个项目得分的平均值获得（平均值计算时需将五级制转换为百分制优=95、良=85、中=75、及格=65、不及格=55）。

2) 在“教师评价”栏中用符号标注评价项目结果（用√表示正确，用×表示错误，用≈表示半对半错）。

3) 在“教师评价”栏中“评价教师签名”填写评价教师（或助教）姓名。将评价后的实验报告转换为 PDF 格式文件归档。

4) 课程实验环节结束后，任课教师将自己教学班的实验报告文件夹进行清理。在提交文件夹中，文件总数为实验次数×教学班学生人数（如，教学班人数为 90 人，实验项目为 5，其文件数为：90×5=450）。任课教师一定要认真清理，总数相符，否则学生该实验项目不能得分。最后将学生提交的实验报告刻光盘连同实验成绩一起放入试卷袋存档。