

南 开 大 学

本 科 生 课 程 设 计 论 文（设 计）

中文题目： Java 课程设计—五子棋游戏

外文题目： Java Course Design-The GoBang Game

学 号： 2010239

姓 名： 李思凡

年 级： 2020 级

专 业： 计算机科学与技术

系 别： 计算机系

学 院： 计算机学院

指导教师： 刘嘉欣

完成日期： 2022 年 1 月 2 日

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：李思凡

2022 年 1 月 2 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

在学习完本学期的 Java 课程后，为了提升实际解决问题的能力并综合巩固所学知识，我采用 Java 编程语言及相关工具，设计并实现了一个具有网络通信功能的五子棋游戏。本文综合描述开发工具的选择、程序框架设计、游戏功能实现、游戏效果展示，在此基础上提出对当前游戏设计的展望。

首先，我采用 eclipse 开发平台和 Java Swing 图形界面开发工具，根据希望实现的游戏功能，设计程序的大体框架为模型-控制-视图，并采用 TCP 传输层协议，利用 socket 进行网络通信。接下来，我具体实现了五子棋游戏的基础功能，包括开局，轮流下棋，判断输赢，悔棋，认输，复盘，文字聊天。最后，我对游戏进行了细节优化，包括美化界面，添加背景音乐和音效，添加倒计时。本文将对功能的逻辑实现和可视化实现进行详述。

关键词：Java；五子棋游戏；网络编程；图形化编程

Abstract

After completing the Java course in this semester, with the aim of improving my ability to solve real problems and review knowledge learned, I use Java programming language and relative tools to design and implement a GoBang Game with the network communication function. This paper comprehensively describes the selection of development tools, framework design, the implementation of game functions and the display of the game effect, as well as puts forwards some prospects.

First of all, I choose eclipse and Java Swing as my main tools and Model-View-Controller as the main frame. To realize network communication, I choose Transmission Control Protocol and socket. Next, I implement the basic functions including starting the game, taking turns playing chess, judging win or loss, taking back a move, giving up the game, recapping the last game and text chatting. Finally, I optimize some details, with beautifying the interface, adding background music, sound effects and countdowns included. This paper will elaborate the logical and visual implementations of these functions.

Keywords: Java; GoBang game; network programming; graphic programming

目 录

声 明.....	I
摘 要.....	II
Abstract.....	III
目 录.....	IV
第一章 引言.....	1
第二章 开发工具选择.....	2
一、编程语言选择.....	2
二、集成开发环境选择.....	3
三、Java GUI 工具包选择.....	3
第三章 软件框架设计.....	4
一、整体框架.....	4
二、模型.....	4
三、视图.....	6
四、控制器.....	7
五、网络通信.....	8
第四章 功能实现.....	9
一、单机模式.....	9
（一）开局.....	9
（二）用户落子.....	10

(三) 电脑随机落子.....	10
(四) 判断输赢.....	12
(五) 复盘.....	12
二、联网模式.....	13
(一) 网络连接与通信.....	13
(二) 开局.....	13
(三) 用户落子.....	13
(四) 悔棋.....	14
(五) 认输.....	14
(六) 判断输赢.....	14
(七) 复盘.....	14
(八) 文字聊天.....	14
三、细节补充.....	15
(一) 倒计时.....	13
(二) 背景音乐和音效.....	13
致 谢.....	16

第一章 引言

在学习完本学期的 Java 课程后，我了解了 Java 语言的基本知识，熟悉了面向对象设计的有关知识，同时也学习了图形化编程和网络程序设计的基础知识。为了综合巩固所学知识，培养解决实际问题的能力，我采用 Java 编程语言及相关工具，设计并实现了一个具有网络通信功能的五子棋游戏。

本文综合描述了游戏从设计到实现的全过程，包括开发工具与程序框架的选择，游戏相关功能的具体实现过程，程序能够达到的效果，在此基础上提出未来的改进方向，达到综合运用所学知识，完整解决实际问题的目的。

本文将从相关工具选择，框架结构设计，功能实现，效果展示，总结与展望等方面详细阐述。

第二章 开发工具选择

一、编程语言选择

（一）Java 语言特点

Java 是一门面向对象编程语言，吸收了 C++ 的优点，摒弃了难以理解的多继承和指针等概念。因此，Java 语言功能强大且简单易用。Java 具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点，可以实现网络编程。Java 基本库的内容非常齐全，社区功能强大。Java 可以比较方便地编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等。

（二）C/C++ 语言特点

C 是面向过程的结构化语言，在程序设计时不如面向对向语言方便。C 语言的处理能力很强，可以直接访问内存的物理地址，实现了对硬件的编程操作。C++ 是对 C 语言的扩展，具有面向对象语言的特征。

（三）Python 语言特点

Python 可以有效的进行面向对象编程，是一门解释型语言，具有丰富的标准库，提供适用于各个主要系统平台的源码和机器码。主要用于人工智能、机器学习、自然语言处理等方面。

（四）比较与总结

经过对三门主要的面向对象语言的比较，可以发现 Java 面向对象的功能强大，相对 C++ 较为容易。具有极强的平台独立与可移植性、丰富的基本库，并且可以实现并发编程和网络编程。考虑网络五子棋游戏的特点，选择 Java 语言最为合适。

二、集成开发环境选择

目前主要的 Java 集成开发环境有 Eclipse 和 IntelliJ IDEA。Eclipse 具有开源免费、无服务、免安装、即开即用的特点；IntelliJ IDEA 功能强大、界面友好，是非常智能的集成开发环境。考虑到自己为初学者，希望加深对 Java 语言的理解和实际代码能力，我选择了智能性相对较弱的 Eclipse 作为集成开发环境。

三、Java GUI 工具包选择

Java 中讨论 GUI 时，主要有以下四种选项：Java AWT, Java Swing, SWT 和 Java FX。Java AWT 非常简陋，能支持的组件过于有限。Java Swing 是 Java 标准库的一部分，非常坚固。Swing 的空间是通过 Java 图形功能绘制出来的，实现独立于平台。SWT 是 Eclipse 的开源项目，它的功能没有通过 Java 虚拟机实现。Java FX 需要 Java FX 编译器，使用并不普遍。综合比较后，我选择了功能比较齐全、稳定性较高的 Java Swing，同时也可以作为熟悉 Java 类库的练习。

第三章 软件框架设计

一、整体框架

（一）MVC 模式简介

MVC 模式代表了 Model-View-Controller（模型-视图-控制器）模式，用于应用程序的分层开发。通过 MVC 可以实现视图与模型代码的分离。

M(Model)表示模型，模型是一个存取数据的对象，封装应用程序状态和功能，可以带有逻辑。控制器的状态改变会映射到模型，模型会响应状态查询，并通知视图改变。

V(View)表示视图，视图表示模型包含的数据的可视化，是用来解释模型，反映模型请求的更新。它发送用户输入给控制器，允许控制器选择视图。

C(Controller)表示控制器，它定义应用程序的行为，将用户动作映射成模型更新。控制数据流向模型对象，并在数据变化时更新视图，使视图与模型分离开。

本游戏采用 MVC 模式作为大体框架进行设计，目的是使程序设计更清晰明了。

MVC 的控制方式如图 1 所示。

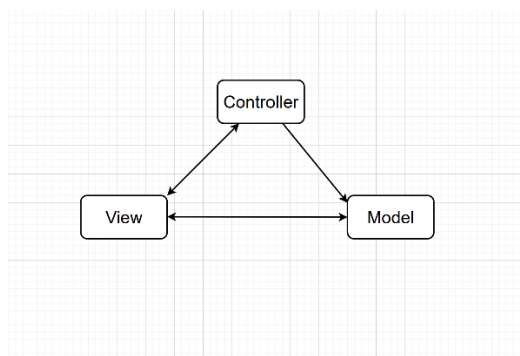


图 1

二、模型

Model 层中存储五子棋对象，封装状态和功能。其中状态指封装的五子棋类 (Chess)，数据域包括棋子的横纵坐标 x 和 y，圆形棋子的直径 h，当前棋子的颜

色 `color` 以及是否是最后一步落子的标记 `last`。方法域包括绘制棋子的方法 `draw`，通过类 `Graphics` 的对象，调用 `drawImage` 方法，根据当前的棋子颜色，在相应的坐标处放上相应颜色的图片，图片的宽度和高度都与所设的直径 `h` 相同。若当前标记的就是最后一步棋，则在棋子上画一个红色正方形作为标记。类中的方法还包括棋子的横坐标 `x`，纵坐标 `y`，颜色 `color`，其中 `color` 为 1 表示是黑棋，`color` 为 2 表示是白棋，是否是最后一步标记的获取方法 `get()` 和设定方法 `set()`，这些方法的访问权限均设为 `public`，方便在其它类中调用。

`Model` 中的状态还包括指示器类(`Pointer`)，指示器是指鼠标移动到能够下棋的点时，该点周围显示的红色方框，如图 2 所示。指示器类中的数据域包括当前指示的点的二维数组下标 `i` 和 `j`；当前点的实际坐标的 `x` 和 `y`；指示器的小正方形的边长 `h`；是否显示指示器的标记 `isShow`；标记当前指示器所指节点棋子状态 `tag`，其中 `tag=0` 表示该点无棋子，`tag=1` 表示该点有黑棋，`tag=2` 表示该点有白棋，`tag` 的作用是只有 `tag=0` 即无棋子时才显示指示器，该点才能下棋。指示器类的方法域包括绘制指示器的方法 `draw` 和 `drawPointer`，在方法 `draw` 中判断指示器的显示状态，标记 `isShow` 为 `true` 时，才调用 `drawPointer` 方法绘制。`drawPointer` 方法中，将点的坐标 `x` 和 `y`，作为所绘制正方形的中心，从正方形的每个顶点各画两条两个方向的小短线，正方形的边长为设定的 `h`。类中的方法还包括判断什么时候显示指示器，需要传进来鼠标当前所在位置的坐标 `xx` 和 `yy`，当 `xx` 和 `yy` 在当前指示器的小正方形范围内（通过对坐标大小进行判定得到），才返回展示 `true`，否则返回 `false`。与五子棋类相似，指示器类中也有其数据成员 `x`，`y`，`i`，`j`，`isShow`，`tag` 的设定方法 `set` 和获取方法 `get`，以便方便的设定值和获取值。



图 2

`Model` 类的数据域包括保存棋子的链表 `chesses`，其中的每个元素都是一个 `Chess` 类对象；当前的棋子颜色 `color`；指示器数组 `pointers`，其中每个元素都是一个 `Pointer` 类对象，二维数组的下标与棋盘点的二维数组下标对应；还包括游戏的状态，复盘的步骤等，与后面的功能相照应。

Model 中的功能部分包括五子棋游戏实现的一些功能。包括开局功能，落子功能，悔棋功能，判断输赢功能，复盘功能，文字聊天功能，电脑随机落子功能，倒计时功能，背景音乐和音效。这些功能的具体实现将在下一张进行详述。

三、视图

View 层中主要体现了可视化部分的实现。本游戏中我采用了 Java Swing 作为主要的图形化工具。View 层继承自 JFrame 类，表示窗口类。在窗口类中主要添加了三个部分，菜单，背景和面板。添加菜单即向窗口中添加 JMenuBar 的对象，JMenu 的对象表示菜单项，JMenuItem 表示每个菜单项的子项，需要将 JMenuItem 的对象添加到对应的 JMenu 对象中，再将 JMenu 对象依次添加到 JMenuBar 中。我设置的菜单项包括“游戏”和“功能”。其中“游戏”中包含子项“开始”“帮助”“退出”，“功能”中包含子项“悔棋”“认输”，通过点击相应的菜单项，将实现对应的功能。

背景的添加是通过组件标签 JLabel 实现的。将背景图片文件存入一个 ImageIcon 对象 image 中，再将这个 image 添加到一个 JLabel 对象 imglabel 中，设置这个 imglabel 的大小和窗口类的大小相同。需要将背景图片添加到窗口类的底层 LayeredPane 中，并将 LayeredPane 上面一层 ContentPane 设置为透明，这样就能将该图片作为背景。

面板部分是整个可视化的核心部分。为了按照自己的想法布局，我采用了绝对布局，需要将布局管理器设置为空。我将界面分为北部（即上方），中部和南部（即下方）。北部包括能选择游戏模式的组合框，通过下拉列表框，可以选择进行单机模式或是联网模式。在服务端，组合框的右侧是输入服务端提供的端口号的文本框和表示连接的按钮，在联网模式下，可以由服务端用户输入希望提供的端口号，并连接客户端。在客户端，组合框的右侧是输入服务端 IP 地址和端口号的文本框，在联网模式下，可以由客户端用户输入希望连接的服务端 IP 地址和端口号。

在中心面板的左侧是绘制的棋盘，棋盘的绘制通过重写 paint 方法实现。首先画棋盘背景，我选取的是一个棋盘背景图片，通过画笔类 Graphics 对象 g，将图片按指定大小绘制到指定位置。接下来是绘制棋盘格子，我画的是 15*15 格子

的棋盘，每个格子像素为 $64*64$ 。通过两次 for 循环，分别画 16 条横线和 16 条竖线，使得整个棋盘格子的大小与棋盘背景的大小一致，就将棋盘绘制完成，同时我也得到了棋盘每个点的坐标和对应的二维数组下标。接着需要绘制指示器和棋子。在 View 类里传入指示器二维数组 `pointers`，其中每个元素都是一个 `Pointer` 类对象，通过二重循环将每个 `pointer` 和棋盘格的二维数组元素对应。在 `paint` 里面将 `pointers` 的每个元素调用 `Pointer` 类的 `draw` 方法，绘制指示器。在 View 类中传入表示所有棋子的链表 `chesses`，在 `paint` 里面对 `chesses` 中每个棋子对象调用 `Chess` 类的 `draw` 方法，绘制每个棋子。这样有关棋盘部分的可视化就完成了。在棋盘的下方添加有关复盘功能的按钮，从左到右分别是“上一步”、“复盘”、“下一步”，这样通过点击按钮可以实现相应的复盘功能。在中心面板的右侧部分是联网模式下的文字聊天区域。上方是输入发送消息的多行文本框，用户可以输入想要发送给对方的文字；发送区下面是一个发送按钮，点击该按钮可以将消息发送给对面；按钮下面是一个显示双方聊天内容的日志。

南部主要负责显示计时器，在联网模式下，服务端下黑棋，客户端下白棋，每当轮到己方下棋时，己方的倒计时功能会启动，从一分钟开始倒计时，己方必须在一分钟内考虑清楚并落子，否则会弹出超时的提示框。

界面的样式如图 3 所示。

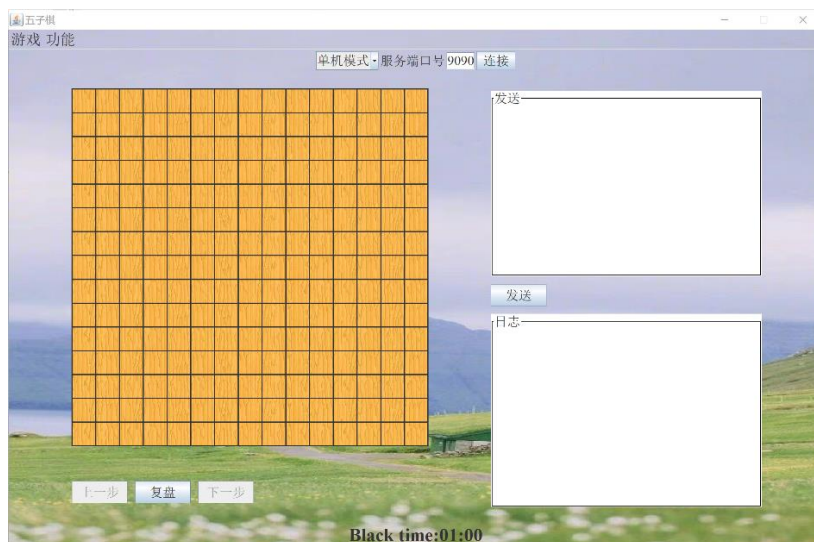


图 3 服务端界面

四、控制器

控制器主要负责控制数据流向模型，并在数据变化时及时更新视图。需要为各类事件（表现为鼠标的各种动作）添加监听器。单击按钮、选择菜单项都表现为 `ActionEvent` 事件，需要为每个菜单选项和每个按钮调用 `addActionListener` 方法，注册该事件，并在方法里通过匿名内部类实现接口 `ActionListener`，传入其对象。实现该接口需要重写 `actionPerformed` 方法，在该方法中写需要实现的事情。在下棋的过程中，我们需要知道鼠标在各时刻的位置，鼠标移动表现为鼠标移动事件，需要调用 `addMouseMotionListener` 方法监听该事件，在其中传入 `MouseAdapter` 对象。该适配器对象需要实现接口 `MouseMotionListener` 中的 `mouseMoved` 方法，在其中判断当前鼠标的位置是否在指示器二维数组的某个指示器的范围内且该位置没有棋子，若是则显示该指示器。在具体落子时，需要鼠标点击某位置，鼠标键的按下表现为鼠标事件，需要调用 `addMouseListener` 方法监听该事件，在其中传入 `MouseAdapter` 对象。该适配器对象需要实现接口 `MouseListener` 中的 `mouseClicked` 方法，在其中实现落子相关的操作。各功能的实现将在下一章详述。控制器中还包括网络连接相关部分，将在下一部分详述。

五、网络通信

（一）TCP 与 Socket 简介

TCP 是一种传输控制协议，是两主机之间有连接的、可靠的、端对端的数据流的传输。`Socket` 就是对网络中不同主机上的应用进程之间进行双向通信的端点的抽象，一个 `Socket` 就是网络上进程通信的一端。服务端的 `ServerSocket` 在某端口进行监听，等待并接受客户端的连接请求，客户端创建 `Socket`，向服务端的某端口发出请求，二者建立连接，并通过 `Socket` 进行通信。通信时，二者通过 I/O 流发送和接受消息。通信结束时，服务端和客户端都关闭 `Socket`，服务端关闭监听。

（二）本游戏网络通信实现

本游戏中的网络通信部分就按照上述原理进行。通过 TCP 和 `Socket` 建立连接，当需要进行网络通信（开局、下棋、悔棋、判断输赢、文字聊天等）时，二者通过 I/O 流进行发送和接受消息，并在视图上做出相应更新和处理。

第四章 功能实现

一、单机模式

通过在组合框里点击单机模式，即可将模式转换为单机。点击单机模式后，控制器中表示游戏模式的数据 `gameMode` 会切换为 1，会通过控制器调用模型中的 `changeMode2To1` 方法，将棋子清空，输入文字的文本框清空且都设为不可编辑，连接和发送按钮设置为不可使用。此时若不点击开始选项，鼠标将无法控制棋盘。

（一）开局

开局按键设置为一个菜单项，当点击“开始”选项后，会先判断当前的游戏状态（判断控制器中 `gameFlag` 是否为 `true`），若当前游戏为正在进行中，则会弹出消息框，提示游戏正在进行中，不能重新开局，如图 4。若当前游戏已结束或未开始，会判断当前的游戏模式（判断控制器中 `gameMode` 为 1 或 2），若为单机模式（`gameFlag=1`），则通过控制器调用模型中的 `restart` 方法，将游戏状态 `gameFlag` 设置为 `true`，棋子颜色设置为黑色（单机模式下黑棋先下），将二维数组中所有指示器的 `tag` 设置为 0，表示未下棋，显示状态设置为 `false`，并将存放棋子的链表清空。此时开局的准备工作完成。

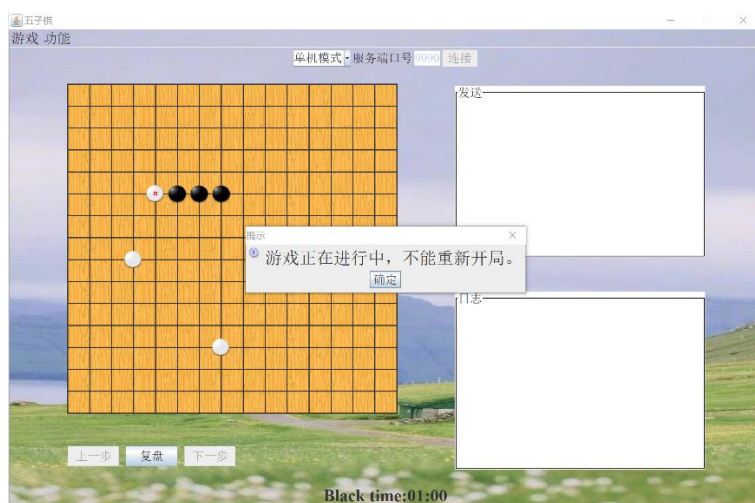


图 4

（二）用户落子

当游戏状态 `gameFlag` 为 `true` 时，可以通过鼠标点击未落子点下棋。在鼠标点击事件的监听中，判断当前鼠标坐标是否在某个未落子指示器范围内，若该点可下棋且鼠标点击，则将新建一个棋子对象，保存当前的横纵坐标和棋子颜色，将这个棋子添加到棋子链表 `chesses` 中，更新该点的指示器状态。转变下一步棋的颜色，并调用 `repaint` 函数，更新视图的显示状态。

（三）电脑随机落子

当用户下完一步棋后，电脑需要也下一步。电脑落子的位置通过计算权重分来找到。具体的算法如下：

由于用户先落子，所以用户下黑棋，电脑下白棋，以下黑棋均表示用户所下。根据五子棋的游戏规则，当同色五子相连时即获胜，因此需要在用户出现三个黑棋相连时，电脑及时在一端下白棋堵住。左开表示在左边可以落子，右开表示在右边可以落子。此处“左边”指左侧，上侧，左上侧，右上侧。“右边”指右侧，下侧，右下侧，左下侧。

以三子为例解释三子左开、三子、三子右开，四子的情况与三子相同。

三子左开，表示已有三个黑子相连，白棋可以下在左边，见图 5。

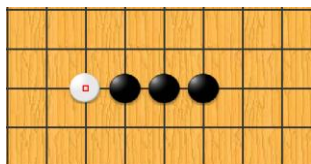


图 5

三子，表示三个黑子间只有一个空格，白棋可以下在该空格，见图 6。

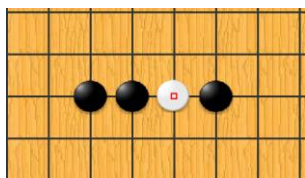


图 6

三子右开，表示三个黑子相连，左边已经被白子堵住，白棋可以下在右边，见图 7。

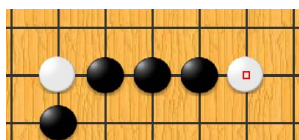


图 7

对于已有三个黑子相连或中间隔一个空格的情况，权重分如下：

1.三个黑子相连或只间隔一个空格时权重分

三子左开	三子	三子右开
32	30	31

对于已有四个黑子相连或中间隔一个空格的情况，权重分如下：

2.四个黑子相连或只间隔一个空格时权重分

四子左开	四子	四子右开
42	40	41

电脑下一步落子需要落在当前权重分最高的位置，需要通过计算和排序找出当前权重分最高的位置。步骤如下：

- 1.拿到指示器数组的每一个元素，只对该位置有棋子的指示器操作。
- 2.循环 $4*2$ 个方向，横向（从左到右，从右到左），竖向（从上到下，从下到上），右斜下（从左上到右下，从右下到左上），左斜下（从右上到左下，从左下到右上），对每个位置的棋子类型判断。
- 3.若颜色相同（为黑棋），则计数器加一；若颜色不同（为白棋），退出循环；若第一次遇到空格，计数器加一，第二次遇到空格，退出循环。
- 4.判断当前状态属于左开、中间还是右开，根据该状态计算权重分，并存到一个集合中。
- 5.循环所有有效点后，对集合中元素进行排序，取权重分最大的位置作为下一步落子的位置。
- 6.若没有权重分最大的位置（权重分都为 0），则电脑随机选取一个未落子的位置进行落子。

电脑落子与用户落子相同，都是新建一个棋子对象，传入坐标和颜色，将该棋子添加到棋子链表中，视图调用 `repaint` 方法画出。电脑的最后一步落子会用红色小正方形标出，每当用户下完一步棋后，清楚上一步的标记。

（四）判断输赢

用户和电脑每下完一步棋，都要判断是否赢棋。判断方法如下：

- 1.以当前落子点为基准位置，横向、竖向、右斜下、左斜下四个方向判断是否出现同色五子相连。
- 2.每个方向都要向左遍历加向右遍历。
- 3.若在某个方向出现五子连珠，则会弹出提示框。若用户操控的黑棋五子连珠，则提示“您胜利了！”，否则提示“您失败了！”。若四个方向遍历后都未出现五子连珠，则无输赢，游戏继续进行。
- 4.若用户胜利或用户失败，本局游戏结束，游戏状态 `gameFlag` 被设置为 `false`。用户可以选择重新开局、联网模式或者复盘。

（五）复盘

每一局结束后，用户可以选择复盘功能，从本局开始到结束，获得每一步下棋情况。实现步骤如下：

- 1.设置一个临时链表 `chessesTemp`，当本局游戏结束时，将棋子链表 `chesses` 复制过来。
- 2.点击复盘按钮，会先判断当前的游戏状态。若游戏正在进行中，不能复盘，会弹出提示框提示游戏正在进行中。若游戏已经结束，会清空棋子链表 `chesses`，并将临时链表的第一个元素放入棋子链表 `chesses` 中，调用 `repaint` 方法，将第一步画在棋盘上。
- 3.若当前复盘到第 i 步，点击“下一步”按钮，会将临时链表的前 $i+1$ 个元素放入棋子链表 `chesses` 中，调用 `repaint` 方法；点击“上一步”按钮，会将临时链表的前 $i-1$ 个元素放入棋子链表 `chesses` 中，调用 `repaint` 方法。
- 4.若已经进行到最后一步，“下一步”按钮将不可用；若当前为第一步，“上一步”按钮会不可用。

二、联网模式

通过在组合框里点击联网模式，即可将模式转换为联网。点击联网模式后，控制器中表示游戏模式的数据 `gameMode` 会切换为 2，会通过控制器调用模型中的 `changeModelTo2` 方法，将棋子清空，输入文字的文本框清空且都设为可编辑，连接和发送按钮设置为可使用。此时若不点击开始选项，鼠标将无法控制棋盘。

（一）网络连接与通信

在服务端的端口号文本框内输入提供的端口号，点击连接按钮；客户端输入服务器的 IP 地址和相同端口号，点击连接按钮，若 IP 地址和端口号输入正确，则显示成功连接。

服务端与客户端的通信通过 I/O 流来实现。以服务端为例，需要建立一个新的线程用来接受从客户端读取的信息。调用服务端 `Socket` 的 `getInputStream` 方法读取信息，将其用 `InputStreamReader` 和 `BufferedReader` 包装，调用 `readLine` 方法一次读取一行从客户端发来的字符串。通过设置“协议”，将每个字符串的开头设置为信息类型描述，根据不同类型的信息做出相应的处理。向客户端写入过程类似，将 `Socket` 的 `getOutputStream` 方法写出的信息包装成 `PrintWriter` 类，当需要与对方通信时，通过 `println` 方法发送消息。消息的开头应设置为此消息类型的协议，方便对方接受并处理。

（二）开局

联网模式下的开局功能整体与单机模式下类似。区别在于，游戏未开始状态下，当服务端或客户端其中一方点击开局按钮，都会向对方发送一条信息，以“start”开头，表示开局。对方线程读取到此消息后，会弹出提示框“对方准备开始游戏，您同意吗？”。若点击“是”，会发送一条以“start:yes”开头的消息，并通过控制器调用 `restart` 方法开局。己方收到对方发来的以“start:yes”开头的消息后，也通过控制器调用 `restart` 方法开局。游戏开始，并由服务端（黑棋）先下棋。

（三）落子

联网模式下落子的实现与单机模式相同，需要添加的是向对方发送己方落子的消息，和接受对方落子消息并绘制。当己方下了一个棋子后，将向对方发送

格式为“Chess:x...y...c...”的消息，其中 x 后为当前落子点的 x 坐标，y 后为当前落子点的坐标，c 后为当前棋子的颜色。当对方收到以“Chess:”开头的信息时，将从字符串中找出 x, y 坐标和棋子颜色，新建一个棋子对象，传入该数据，并添加到链表 chesses 中，进行重绘。这样一来，在服务端或客户端任何一方下的棋子都能被对方获得并绘制在棋盘上，即实现了联网下棋的功能。

（四）悔棋

悔棋按键设置为一个菜单项，点击后，通过控制器向对方发送消息“regret”，对方读取到“regret”字符串后，会弹出提示框“对方想悔棋，您同意吗？”。若点击“是”，会发送格式为“regret:yes”的字符串，己方收到该消息后，双方会从 chesses 链表中获得最后一步棋子，并将其删除，再调用 repaint 方法，棋盘上就不会出现上一步下的棋子，此时对方的游戏状态 gameFlag 设置为 false，己方的游戏状态设置为 true，将重新选择下棋的位置进行下棋。

（五）认输

认输功能的原理与悔棋非常相似。也设置为一个菜单项。点击后通过控制器向对方发送消息“giveup”，对方收到该类型消息后，将弹出提示框“对方想认输，您同意吗？”。若选择“是”，会发送格式为“giveup:yes”的字符串，并弹出提示框提示胜利。己方收到反馈回来的消息后，将给出提示框提示失败。

（六）判断输赢

判断输赢的实现也与单机模式下相同。需要添加的是一方胜利后向另一方发送消息，为“blackWin”或者“whiteWin”。对方收到消息后，也需要将游戏状态 gameFlag 设置为 false，标志游戏结束。

（七）复盘

联网模式下也可以进行复盘操作。由于复盘是“单机”性质的，不需要与对方进行通信，所以复盘的实现与单机模式下完全相同。

（八）文字聊天

联网模式下双方可以进行文字聊天。将文字通过键盘输入到文本框里，点击发送按钮后，通过输出流将文本框内的内容发送到对方。前面的协议若为“Server:”，表示这是由服务端向客户端发送的文字聊天内容；若为“Client:”，

表示这是由客户端向服务端发送的文字聊天内容。对方收到聊天内容后，会将该字符串显示在己方的日志文本区域内。即可实现文字聊天。

三、细节补充

（一）倒计时

联网模式下，双方交替下棋，可以添加倒计时功能。初始时时间都设置为 1 分钟。倒计时功能主要通过 `Timer` 类和 `Date` 类来实现，当启动 `Timer` 后，时间将从 1 分钟开始，间隔 1 秒的减少，倒计时截止后将给出提示信息。每当轮到己方下棋时，计时器将启动，当落子结束后，计时器将停止倒计时，并恢复为初始时长。

（二）背景音乐和音效

背景音乐的添加需要在启动函数里，新建一个线程用来循环播放音乐。音频通过 `AudioInputStream` 导入，通过调用 `Clip` 类对象的 `open` 方法和 `start` 方法进行添加和播放。因为背景音乐要循环播放，可以设置一个死循环。由于播放音乐是一个新建的线程，所以不会影响其他功能的执行。

音效的添加原理与背景音乐的添加相同，在相应位置新建一个线程，加载音频文件，播放一遍音乐。落子时，鼠标点击落子点后会触发落子的音效；游戏一方胜利时，会触发游戏胜利音效。

致谢

在本论文的完成过程之际，感谢在我完成此游戏设计过程中帮助我的人们。首先感谢我的 Java 课程老师刘嘉欣老师，在本学期课程中为我们讲授 Java 语言相关的基础知识和实用的拓展功能，帮助我把握到每一部分内容的重点，其中图形化部分和网络通信部分对我完成网络五子棋游戏设计帮助很大。

再者，在游戏设计过程中，感谢在一些细节问题的处理上为我解答疑惑和提供帮助的朋友和同学，帮助我节省了解决问题的时间成本，我们也在相互学习中共同进步。

最后衷心的感谢在百忙之中参加答辩和评阅论文的老师们！