**第5组**

《Java程序设计》课程报告

（08306099）

学 期 ： **2018 -2019 年度冬季学期**

项目名称 ： **五子棋小游戏**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 角色 | 学号 | 姓名 | 签名 | 格式 | 项目概述 | 功能设计 | 关键代码 | 体会 | 难度系数 | 合计成绩 |
| 5 | 5 | 10 | 50 | 10 | 20 | 100 |
| 组长 | 16121783 | 阳鹏 |  |  |  |  |  |  |  |  |
| 组员1 | 16121813 | 柳静楠 |  |  |  |  |  |  |  |  |
| 组员2 | 16122160 | 曾灵灵 |  |  |  |  |  |  |  |  |

五子棋小游戏

1. 项目概述

**1.项目背景**

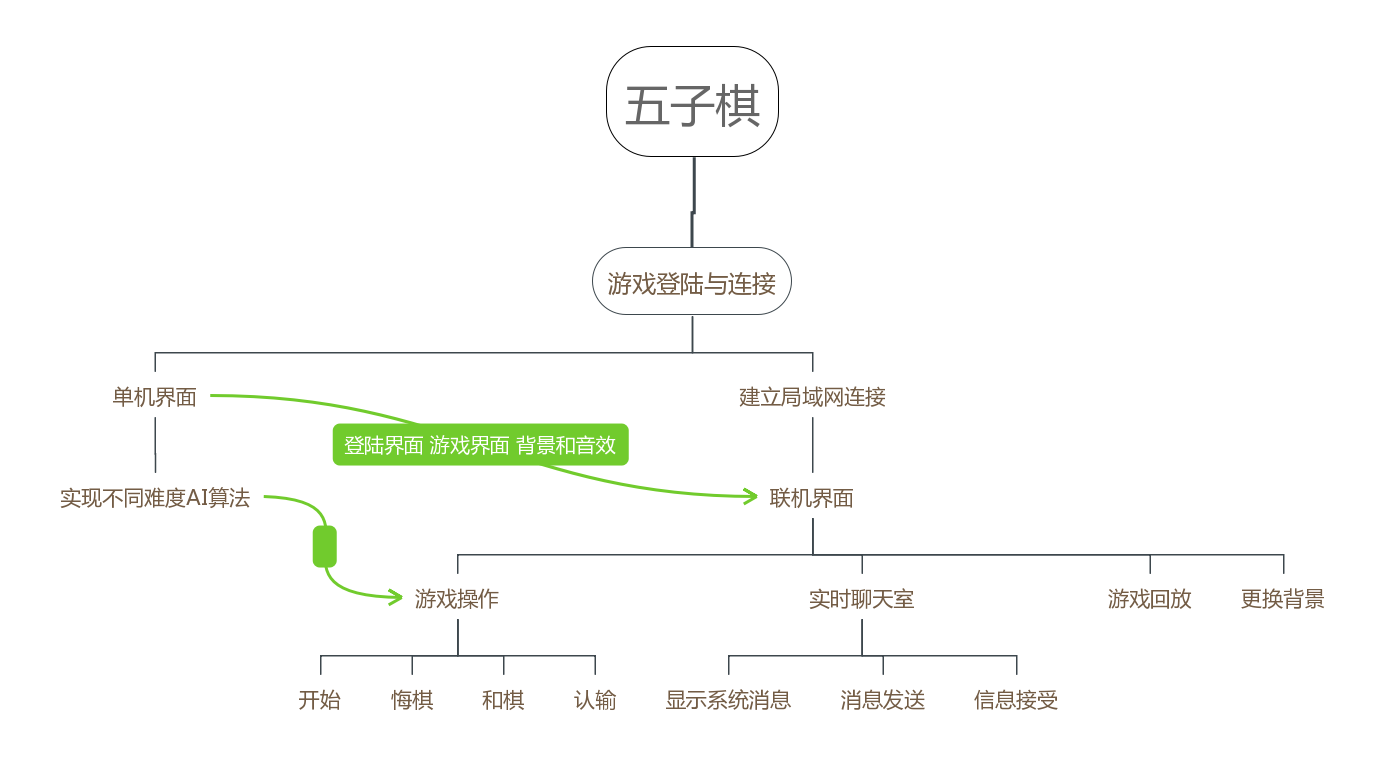
五子棋是一个简单而又经典的棋类游戏，是一种两人对弈的策略游戏，也叫做“串珠”，是中国民间熟知的古老的棋种。相传起源于四千多年前的尧帝时期，比围棋历史更为悠久。它发展于日本，流行于欧美。容易上手，老少皆宜，而且具有趣味性。五子棋作为一种益智类游戏，能够陶冶情操，开发智力。因为其门槛较低，所以覆盖群众广。为了满足大众需求，我们就选择去实现五子棋这个项目。

**2.需求分析**

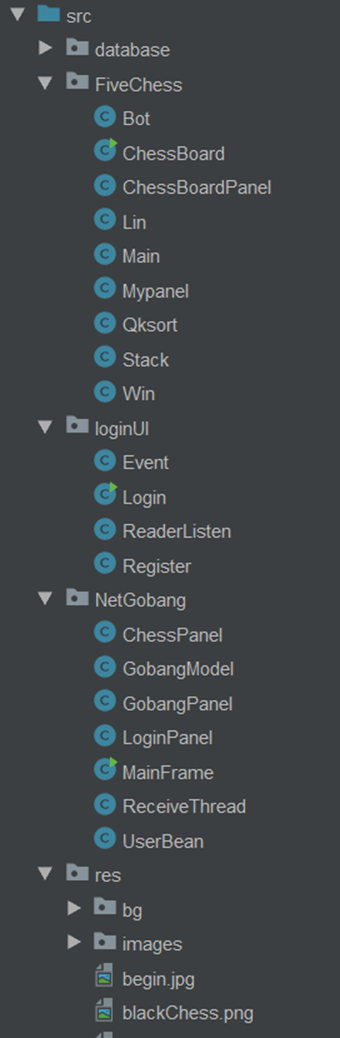
1. 开发五子棋游戏要求具有以下功能：
2. 系统操作简单,界面友好
3. 支持游戏悔棋与游戏回放功能
4. 提供游戏背景更换功能,避免视疲劳。
5. 支持聊天功能,增强游戏沟通能力

**3.系统设计**

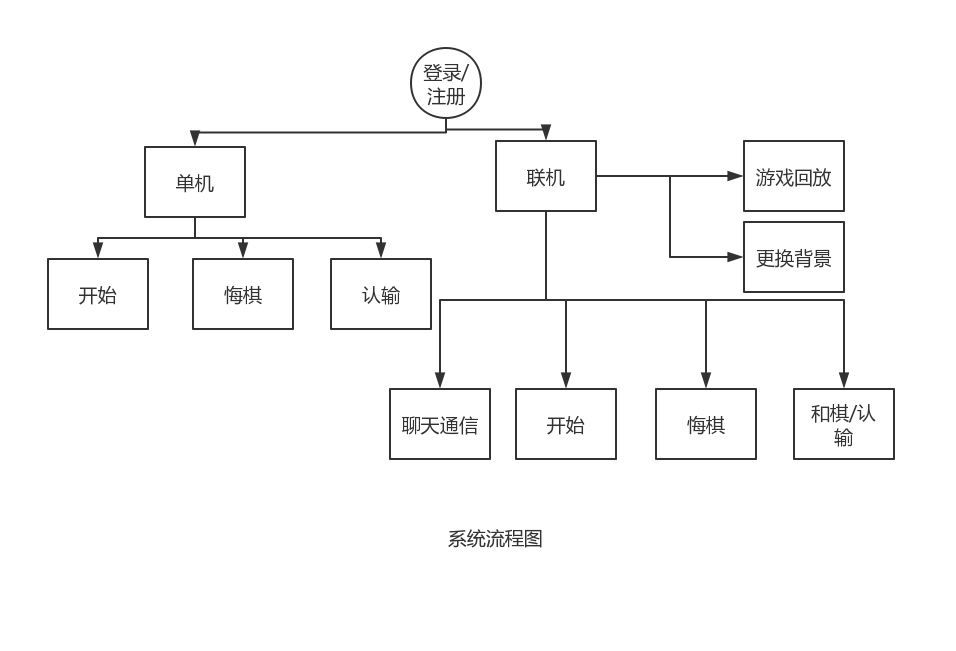
五子棋界面包括四大板块：登录注册等，单机，联机，聊天等界面。



**项目文件列表：**

****

1. 系统功能设计



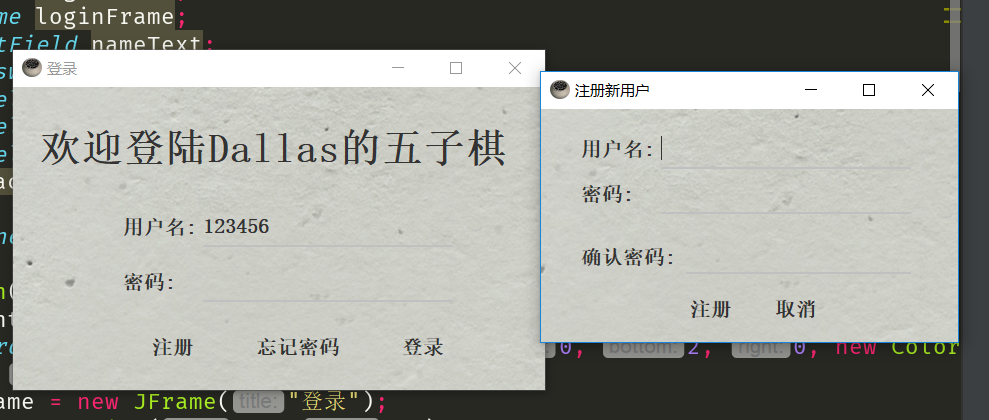
**游戏选项卡:**

1. 登录界面：注册、忘记密码、登录
2. 主界面：单机游戏、联机游戏
3. 联机界面：开始、悔棋、和棋、认输、聊天、设置、棋局回放
4. 单机界面：继承自联机界面

**页面布局：**

1. 登陆界面后进入主界面，分设单机游戏和联机游戏选项；
2. 联机界面居中靠左设置棋盘，棋盘两边设置用户信息框，下方设置开始、悔棋等选项。
3. 界面右栏设置一个简单的实时聊天界面。

**页面截图：**





主界面

|  |
| --- |
|  |

1. 关键技术与核心代码

|  |
| --- |
| **所用到的关键技术知识点如下：**  在进行五子棋的网络联机部分的设计过程中使用到了socket类库。   1. Java.net.ServerSocket 2. 构造函数：   public ServerSocket();  构造一个ServerSocket，但不绑定任何端口，所以也不能接受任何的请求连接。  public ServerSocket(int port);  通过一个端口来构造一个ServerSocket对象。  public ServerSocket(int port, int backlog); 通过一个端口和TCP队列大小来构造一个ServerSocket对象。  public ServerSocket(int port, int backlog, InetAddress bindAddr);  通过一个端口、TCP队列大小和一个InetAddress 来构造一个ServerSocket对象。   1. 主要方法   Socket accept();  监听端口与发送请求的socket建立连接。  　　 void bind(SocketAddress endpoint, int backlog);  　　 为初始没有绑定端口的server socket绑定端口，endpoint即要绑定的端口，backlog可以省略，功能同构造函数中。  　　 void close();  　　 关闭这个server socket。   1. Java.net.Socket 2. 构造函数   Socket();  　　 建立一个无连接的socket。  　　 Socket(InetAddress address, int port);  　　 新建socket，连接到指定的IP地址和端口。  　　 Socket(String host, int port);  新建socket，连接到指定的主机名和端口。   1. 主要方法   void bind(SocketAddress bindpoint)  　　 void close();  InetAddress getLocalAddress();int getLocalPort();int getPort()； 获取本地的IP地址，端口等信息  　　 InputStream getInputStream();OutputStream getOutputStream()  　　 返回这个socket的输入流和输出流。   1. Runtime 2. 每个应用程序都有一个runtime类实例，使得应用程序能够与运行环境相连接。可以通过getRuntime()方法来获取当前运行时对象。   Runtime类的大多数方法为实例方法。  public static Runtime getRuntime();返回与当前 Java 应用程序相关的运行时对象。  public Process exec(String command) throws IOException  在单独的进程中执行指定的字符串命令。command - 一条指定的系统命令。   1. 与数据库相关的方法与接口。 2. java.sql.Connection;(Connection 接口)   负责连接数据库并担任传送数据的任务，如：连接数据库，要有一个通道，那么这个通道就是Connection。   1. java.sql.DriverManager; 用于依据数据库的不同，管理JDBC驱动   常用方法为：getConnection()方法。  public static Connection getConnection(String url) throws SQLException 其中 String url：数据库URL  public static Connection getConnection(String url,Properties info) throws SQLException 其中String url：数据库URL；Properties info：一系列字符串键值对用来作为连接参数，一般至少包括user和password两个属性。  public static Connection getConnection(String url, String user,String password) throws SQLException  String url：数据库URL；String user:连接数据库的用户名；String password：连接数据库的密码。   1. Statement 接口   由 Connection 产生、负责执行SQL语句，如：有了通道了，我开个车过去就是Statement，执行SQL语句。   1. ResultSet 接口   负责保存Statement执行后所产生的查询结果，如，开了车过去了，把东西拿回来，那个东西就是ResultSet.   1. JDBC的工作原理   https://images2015.cnblogs.com/blog/1045520/201612/1045520-20161213151002464-1670495149.pnghttps://images2015.cnblogs.com/blog/1045520/201612/1045520-20161213150344854-1032739970.png 如图所示：  **核心代码：**  1. 数据库的连接  利用jdbc提供的connection接口来实现数据库的连接。  public class DatabaseConnect {  private static Connection conn = null;  public static Connection getConnect() {  try {  String url = "jdbc:sqlserver://localhost:1433;DatabaseName=account;";  String user = "sa";  String password = "123456";  conn = DriverManager.getConnection(url, user, password);  } catch (SQLException sqlException) {  sqlException.printStackTrace();  System.out.println("数据库连接失败");  }  return conn;  }  }  2. 网络互联  public void startServer() {  try {  // 创建Socket服务器对象  final ServerSocket chatSocketServer = new ServerSocket(9528);  // 创建接收信息的线程  new ReceiveThread(chatSocketServer, this).start();  } catch (IOException ex) {  JOptionPane.showMessageDialog(this, "本程序禁止重复运行，只能同时存在一个实例。","你确定重复运行？", JOptionPane.ERROR\_MESSAGE);  System.exit(0);  Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null,  ex);  }  }  public void run() {  while (true) {  try {  frame.serverSocket = chatSocketServer.accept(); // 接收Socket连接  Socket serverSocket = frame.serverSocket;  host = serverSocket.getInetAddress().getHostName(); // 获取对方主机信息  String ip = serverSocket.getInetAddress().getHostAddress(); // 获取对方IP地址  int link = JOptionPane.showConfirmDialog(frame, "收到" + host  + "的联机请求，是否接受？"); // 询问是否接受联机  if (link == JOptionPane.YES\_OPTION) { // 如果接受联机  LoginPanel loginPanel = (LoginPanel) frame.getRootPane()  .getGlassPane(); // 获取登录面板的实例  loginPanel.setLinkIp(ip); // 设置登录面板的对家IP信息  }  serverSocket.setOOBInline(true); // 启用紧急数据的接收  InputStream is = serverSocket.getInputStream(); // 获取网络输入流  ObjectInputStream objis = new ObjectInputStream(is);// 创建对象输入流  while (frame.isVisible()) {  serverSocket.sendUrgentData(255); // 发送紧急数据  Object messageObj = objis.readObject(); // 从对象输入流读取Java对象  if (messageObj instanceof String) { // 如果读取的对象是String类型  String name = frame.getTowardsUser().getName();// 获取对家昵称  frame.appendMessage(name + "：" + messageObj); // 将字符串信息添加到通讯面板  } else if (messageObj instanceof byte[][]) { // 如果读取的是字节数组对象  GobangModel.getInstance().setChessmanArray( // 将数组对象设置为棋盘模型数据  (byte[][]) messageObj);  frame.getChessPanel1().getGobangPanel1().setTurn(true);// 获得走棋权限  byte myColor = frame.getChessPanel1().getGobangPanel1()  .getMyColor(); // 获取自己的棋子颜色  frame.getChessPanel1().getGobangPanel1().zhengliBoard(  myColor); // 整理棋盘  frame.getChessPanel1().backButton.setEnabled(true);// 悔棋按钮可用  } else if (messageObj instanceof Integer) {// 如果是整形对象  oprationHandler(messageObj);// 命令代码的接收和处理方法  } else if (messageObj instanceof UserBean) {// 如果是用户实体对象  UserBean user = (UserBean) messageObj;  frame.setTowardsUser(user); // 设置对家信息  }  }  } catch (SocketException ex) {  Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE,  null, ex);  JOptionPane.showMessageDialog(frame, "连接中断");  frame.getChessPanel1().reInit();  DefaultTableModel model = (DefaultTableModel) frame.userInfoTable.getModel();  model.setRowCount(0);  frame.getGlassPane().setVisible(true);  } catch (IOException ex) {  Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE,  null, ex);  } catch (ClassNotFoundException ex) {  Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE,  null, ex);  }  }  }  3. 联机按钮  private void loginButtonActionPerformed(java.awt.event.ActionEvent evt) {  try {  // 获取主窗体的实例对象  MainFrame mainFrame = (MainFrame) getParent().getParent();  String name = nameTextField.getText(); // 获取用户昵称  if (name.trim().isEmpty()) {  JOptionPane.showMessageDialog(this, "请输入昵称");  return;  }  String ipText = ipTextField.getText(); // 获取对家IP地址  if (ipText == null || ipText.isEmpty()) {  JOptionPane.showMessageDialog(this, "请输入对家IP地址");  return;  }  ipTextField.setEditable(true);  InetAddress ip = InetAddress.getByName(ipText);  if (ip.equals(InetAddress.getLocalHost())) {  JOptionPane.showMessageDialog(this, "不能输入自己的IP地址");  return;  }  socket = new Socket(ip, 9528); // 创建Socket连接对家主机  if (socket.isConnected()) { // 如果连接成功  user = new UserBean(); // 创建用户对象  // 获取当前时间对象  Time time = new Time(System.currentTimeMillis());  user.setName(name); // 初始化用户昵称  user.setHost(InetAddress.getLocalHost()); // 初始化用户IP  user.setTime(time); // 初始化用户登录时间  socket.setOOBInline(true); // 启用紧急数据的接收  mainFrame.setSocket(socket); // 设置主窗体的Socket连接对象  mainFrame.setUser(user); mainFrame.send(user); setVisible(false); // 隐藏登录窗体  }  } catch (UnknownHostException ex) {  Logger.getLogger(LoginPanel.class.getName()).log(Level.SEVERE,  null, ex);  JOptionPane.showMessageDialog(this, "输入的IP不正确");  } catch (IOException e) {  e.printStackTrace();  JOptionPane.showMessageDialog(this, "对方主机无法连接");  }  }  4.游戏悔棋  public synchronized void repentOperation() {  // 获取下棋队列  Deque<byte[][]> chessQueue = gobangPanel1.getChessQueue();  if (chessQueue.isEmpty()) {  return;  }  // 获取上两次次走棋的棋谱  for (int i = 0; i < 2 && !chessQueue.isEmpty(); i++) {  byte[][] pop = chessQueue.pop(); // 废弃走棋步骤  }  if (chessQueue.size() < 1) {  chessQueue.push(new byte[15][15]);  }  byte[][] pop = chessQueue.peek();  GobangModel.getInstance().updateChessmanArray(pop);// 更新棋盘的棋子布局  repaint();  }  6.游戏回放  private void backplayToggleButtonActionPerformed(  java.awt.event.ActionEvent evt) {  // 如果游戏进行中，提示用户游戏结束后在观看游戏回放  if (gobangPanel1.isStart()) {  JOptionPane.showMessageDialog(this, "请在游戏结束后，观看游戏回放。");  backplayToggleButton.setSelected(false);  return;  }  if (!backplayToggleButton.isSelected()) {  backplayToggleButton.setText("游戏回放");  } else {  backplayToggleButton.setText("终止回放");  new Thread() { // 开启新的线程播放游戏记录  public void run() {  Object[] toArray = gobangPanel1.getOldRec();  if (toArray == null) {  JOptionPane.showMessageDialog(ChessPanel.this,  "没有游戏记录", "游戏回放", JOptionPane.WARNING\_MESSAGE);  backplayToggleButton.setText("游戏回放");  backplayToggleButton.setSelected(false);  return;  }  // 清除界面的结局文字，包括对方胜利、你胜利了、此战平局  gobangPanel1.setTowardsWin(false);  gobangPanel1.setWin(false);  gobangPanel1.setDraw(false);  for (int i = toArray.length - 1; !gobangPanel1.isStart()  && backplayToggleButton.isSelected() && i >= 0; i--) {  try {  Thread.sleep(1000); // 线程休眠1秒  } catch (InterruptedException ex) {  Logger.getLogger(ChessPanel.class.getName()).log(  Level.SEVERE, null, ex);  }  GobangModel.getInstance().updateChessmanArray(  (byte[][]) toArray[i]); // 根据游戏记录跟换每一布游戏的棋谱  gobangPanel1.repaint(); // 重绘棋盘  }  backplayToggleButton.setSelected(false);  backplayToggleButton.setText("游戏回放");  }  }.start();  }  }  五子棋下棋算法太长此处不再黏贴，请参见源码 |

1. 成员角色与分工

|  |
| --- |
| 1. 阳鹏：负责五子棋游戏逻辑，走法，赢法，AI算法等 2. 柳静楠：负责界面设计，包括登录界面游戏界面等等 3. 曾灵灵：联机模块，负责Socket，实现联机界面各种实时交互。 |

1. 体会

阳鹏：作为组长，本次项目一方面我要统筹兼顾整体项目，另一方面我主要负责实现五子棋规则的各种算法。相比于做界面的同学会涉及到一些数据库知识，做实时通信的同学会用到Socket网络编程知识，我主要涉及到的还是如何用Java封装好一个个算法，以供游戏运行时调用各个模块实现其功能。本次项目中主要会涉及到的一些算法有赢法判断，着子生成，以及AI中简单的赢法数组和赢法统计数组等等。通过这次实践学习，不仅提升了我的Java编程水平，也锻炼了不少算法能力，看着自己的项目能真正运行起来，内心还是很兴奋的。

柳静楠：本次项目我主要负责的是界面设计。为了与五子棋相匹配，我选择了比较古风的背景。登录，主界面比较简洁，聊天室界面像以前软件的界面。这些在之后的学习中还可以改进。虽然这次项目遇到了许多困难，但在我们的努力下都一一解决了。通过这次项目，不仅锻炼了我JAVA的编程能力，也让我研究了前端的一些知识，体会到了一个项目的产生的过程，最终终于完成了项目。

曾灵灵：本次项目的编写，我主要是在socket网络聊天的部分。由于之前在计算机网络课程上学了有关socket的知识。所以在这次的项目过程中，我对socket的一些理论知识理解上更加透彻。在Java socket的类库中有两个socket server和socket，对这两个类的一些关系以及作用的了解是我这次项目收获最多的。此外，通过这次的项目合作，使我对多人协作开发有了初步的理解。也通过这次的项目经历收获了团队协作的一些技巧。但我清楚的知道自己这次的团队协作表现并没有特别完美，希望在今后的团队协作中能够更加注重细节，更加的注重团队精神。

1. 参考文献

|  |
| --- |
| <https://www.jb51.net/article/69909.htm>  <https://blog.csdn.net/shecanwin/article/details/70846690>  <https://blog.csdn.net/to_be_better/article/details/50913168>  <https://blog.csdn.net/qq_40577923/article/details/80269813>  <https://blog.csdn.net/xiaocxks/article/details/52847371>  <https://wenku.baidu.com/view/b7738a6b3b3567ec102d8a75.html>  <https://blog.csdn.net/ytyggg/article/details/83831180>  <https://blog.csdn.net/scbiaosdo/article/details/80526655>  <https://blog.csdn.net/qq_26805137/article/details/70159118>  <http://www.cnblogs.com/StarKL/p/6165717.html>  <https://blog.csdn.net/min996358312/article/details/62231033> |