**南京邮电大学计算机学院**

**2019／2020学年第2学期**

**《JAVA程序设计》课程大作业报告**

**课题代号（单选）☑课题1 □课题2 □课题3**

**课题名称：**\_ \_**剪刀石头布游戏程序**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**团队成员人数（单选）□1人 ☑2人 □3人 □4人 □5人**

**团队成员名单(学号和姓名）**：

**(1)学号\_\_\_\_B18060130\_\_\_\_\_\_姓名 俞博宸**

**(2)学号 B18030322 姓名 吴 雯**

**(3)学号 姓名**

**课题实现采用的操作系统（可多选，如为其他请填写操作系统名称）**

**☑Microsoft Windows □Unix ☑ Linux □Mac OS □其他\_\_\_\_\_\_**

**课题实现采用的软件开发工具（可多选，如为其他请填写具体工具名称）**

**☑NetBeans □Eclipse □JDK命令行 ☑其他\_\_IntelliJ idea\_\_**

**课题实现采用的文档工具（可多选，如为其他请填写具体工具名称）**

**☑Microsoft Office □JavaDoc □Markdown □LaTex □其他\_WPS\_**

**课题实现采用的团队协作管理工具（可多选，如为其他请填写具体工具名称）**

**☑人工进行 □SubVerion □GIT □其他\_\_\_\_\_\_\_\_\_\_\_\_\_**

**课题完成时间**：\_\_**2020/5/27\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**对应提交的电子文档文件夹名称(准确填写)**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**=================以下部分为教师填写区，请勿填写=================**

**评阅编号:（ ）－（ ）**

**成绩评定：详见大作业对应的“《JAVA程序设计》课程大作业评分记录表”**

**南京邮电大学计算机学院2019／2020学年第2学期**

**《JAVA程序设计》课程学习反馈表**

**说明：本反馈表用于了解课程学习情况以便今后更好地优化课程教学，请大作业团队成员根据情况认真填写本反馈表，谢谢！（请在选项中涂黑或打勾）**

**一．课程学习部分**

1.在选修本课程前大作业团队成员对JAVA技术的了解（单选）

□一点不了解 ☑有一些了解 □已开始自学 □已完成自学

2.在完成本课程后大作业团队成员对JAVA技术的了解（单选）

□一点不了解 □有一些了解 ☑有较多了解 □有很多了解

3.在完成本课程后大作业团队成员希望进一步学习的JAVA技术领域（多选）

☑JAVA企业及云计算 ☑JAVA WEB前端及手机应用 □JAVA嵌入式系统应用 ☑JAVA数据分析及大数据处理

4.在完成本课程后大作业团队成员认为本课程中需要加强的教学内容（多选）

☑面向对象理论 □JAVA基础语法 ☑课后作业 □课内实验

**二．大作业部分**

1.大作业团队成员认为本课程的大作业课题难度情况（单选）

□很难 □比较难 ☑简单 □非常简单

2.在完成大作业过程中对软件开发工具的使用情况（单选）

□不会使用工具，需要帮助 □基本会使用工具，但有困难 □会使用工具，基本没有困难 ☑会使用工具，并且能对外提供帮助

3.在完成大作业过程中对软件文档工具的使用情况（单选）

□不会使用工具，需要帮助 □基本会使用工具，但有困难 □会使用工具，基本没有困难 ☑会使用工具，并且能对外提供帮助

4.在完成大作业过程中对软件协作工具的使用情况（单选）

□不会使用工具，需要帮助 □基本会使用工具，但有困难 □会使用工具，基本没有困难 ☑会使用工具，并且能对外提供帮助

5.完成大作业对上述各种信息技术工具使用能力的总体提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

6.在完成大作业过程中获取的工程能力中的软件设计能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

7.在完成大作业过程中获取的工程能力中的代码编写能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

8.在完成大作业过程中获取的工程能力中的代码调试能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

9.在完成大作业过程中获取的工程能力中的代码测试能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

10.在完成大作业过程中获取的工程能力中的代码优化能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

11.在完成大作业过程中获取的工程能力中的算法设计能力的提升（单选）

□没有提升 □有一点提升 ☑有较多的提升 □有很大的提升

12.在完成大作业过程中获取的工程能力中的文档撰写能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

13.在完成大作业过程中获取的工程能力中的团队协作能力的提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

14.完成大作业对上述各种实际工程能力的总体提升（单选）

□没有提升 □有一点提升 □有较多的提升 ☑有很大的提升

**三．其它反馈意见（欢迎更多的意见和建议，请填写在下面空白处，谢谢）**

**目录**

1. 软件系统分析和设计方案..................................1

二．软件实现和代码编写.....................................4

三．算法分析................................................44

四．软件调试和测试.........................................45

五．课题项目工程实践总结..................................46

六．参考文献和材料.........................................49

七．团队成员姓名和联系方法................................49

八．附录.....................................................49

**一．软件系统分析和设计方案**

**1.需求分析：**

基于Java swing设计一个剪刀石头布的游戏程序，其可以实现人机对抗以及玩家对抗。每一种游戏方式都有四种游戏模式可以选择，分别为一局胜负、三局两胜、五局三胜以及无尽之战。

**2.功能分析：**

主界面共有四个选项，分别为人机游戏、人人游戏、退出游戏、关于四个模块(如图1.1 游戏主界面)。人机与人人游戏分别有四种模式可以选择(如图1.2 模式选择)。



图1.1 游戏主界面 图1.2 模式选择

2.1人机游戏

即可鼠标操作又可键盘操作。点击进入人机模式的选择；在一局胜负模式下，玩家猜拳的同时机器也做出相应的选择并将两者的选择展示出来并弹出结果框(如图1.3所示)，点击“是”则继续游戏，点击“否”则返回游戏主界面(图1.1);三局两胜和五局三胜的模式下，遇到平局则此局不计入直至一方胜利次数达到2或3判断输赢，游戏界面实时显示玩家输赢结果结束时弹出结果框(如图1.4所示)；无尽之战模式下，玩家可以进行较多次数的游戏，游戏结果实时显示但不会有消息弹框，需要人为结束游戏。

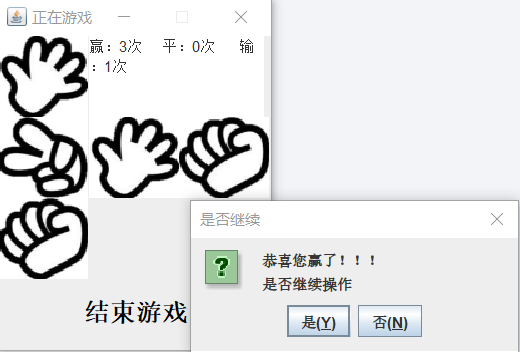


图1.3 图1.4

2.2人人游戏

两个玩家分别用键盘控制自己想要出的手势，先出的玩家的手势会以未知手势形式显示出来，、（鉴于游戏流畅性考虑，我们未采用多次按下触发作弊机制，而是一旦按下某个手势之后便被锁定不能按其他手势），同时触发3s倒计时，另一玩家需要三秒内做出选择，然后会将双方的选择显示出来并实时显示甲乙输赢结果(图1.5所示)，如若超时直接判输(图1.6所示)。

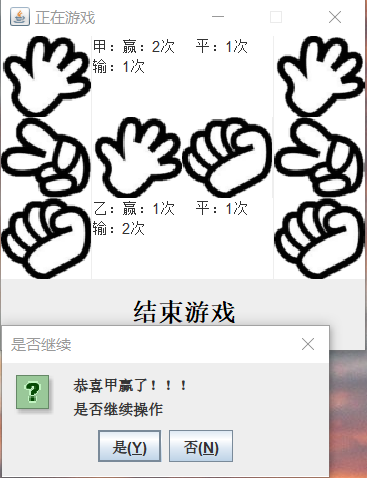


图1.4 图1.5

2.3关于

点击关于可以查阅该游戏程序的相关说明。

2.4退出游戏

点击退出游戏将会退出程序，终止运行。

**3.系统设计方案：**

3.1游戏流程图：

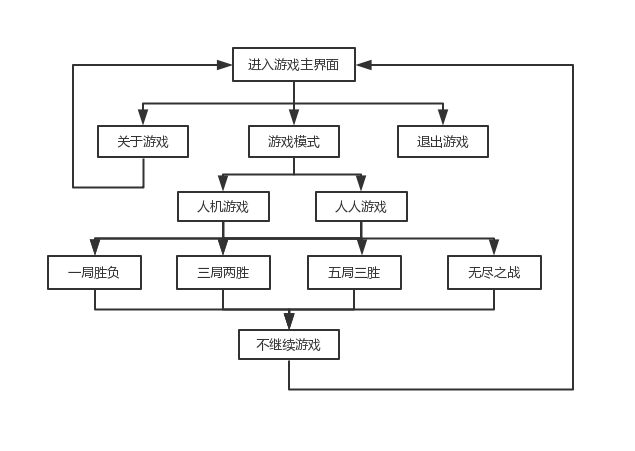


图1.6 游戏流程图

3.2游戏面板

人机游戏模式下，既可以鼠标点击游戏界面的图标，也可以用键盘控制；左键头代表布，下箭头代表剪刀，右箭头代表石头；

人人模式下，双方通过键盘按键来操纵出拳选择；甲方：A键代表布，S键代表剪刀，D键代表石头；乙方：左键头代表布，下箭头代表剪刀，右箭头代表石头。

3.3 弹框提示

在相应的游戏模式一轮得出输赢后会弹出输赢判断以及是否继续的弹框(如图1.7)。

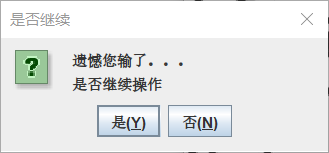


图1.7 弹框

**二．软件实现和代码编写**

**1.rock\_paper\_scissors.java**

1. **package** rock\_paper\_scissors;
3. **import** java.awt.Font;
4. **import** java.awt.event.MouseAdapter;
5. **import** java.awt.event.MouseEvent;
6. **import** java.io.BufferedReader;
7. **import** java.io.File;
8. **import** java.io.FileReader;
9. **import** java.io.IOException;
10. **import** javax.swing.JButton;
11. **import** javax.swing.JFrame;
13. **public** **class** rock\_paper\_scissors {
14. JFrame jf;
15. JButton jb1, jb2, jb3, jb4;
16. MyPanel jp;
18. **private** **void** init(String path) {//初始化
19. jf = **new** JFrame();//东西南北中，五个位置。
20. jp = **new** MyPanel();
21. **if** (path == **null**) {
22. MyPanel.path = "src\\rock\_paper\_scissors\\背景0.jpg";
23. } **else** {
24. MyPanel.path = path;
25. }
26. //jp.setOpaque(false);将jpanel设置为透明将显示jfreame的背景颜色
27. jp.setSize(400 + 8 + 8, 400 + 31 + 8);
28. jf.setTitle("剪刀石头布");
29. jf.setSize(400 + 8 + 8, 400 + 31);
30. jf.setResizable(**false**);//设置此窗体是否可由用户调整大小。
31. jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
32. /\*
33. \* :将contentPane的颜色设置为你想要的颜色，而不是对JFrame本身设置；
34. 核心代码：this.getContentPane().setBackground(Color.red);//设置contentPane为红色
35. \*/
36. //jf.getContentPane().setBackground(Color.red);
37. jf.setLocationRelativeTo(**null**);
38. jf.setLayout(**null**);
39. jp.setLayout(**null**);//jpanel为流式布局改为空
40. jb1 = **new** JButton("人机游戏");
41. jb1.setFont(**new** Font("华文行楷", Font.BOLD, 30));
42. //  jb1.setSize(100, 50);
43. jb1.setBorderPainted(**false**);//不打印边框
44. jb1.setBorder(**null**);//除去边框
45. jb1.setFocusPainted(**false**);//除去焦点的框
46. jb1.setContentAreaFilled(**false**);
47. jb1.addMouseListener(**new** MouseAdapter() {
49. @Override
50. **public** **void** mouseExited(MouseEvent e) {
51. // TODO 自动生成的方法存根
52. jb1.setFont(**new** Font("华文行楷", Font.BOLD, 30));
53. }
55. @Override
56. **public** **void** mouseEntered(MouseEvent e) {
57. // TODO 自动生成的方法存根
58. jb1.setBounds(115, 50, 170, 50);
59. jb1.setFont(**new** Font("华文行楷", Font.BOLD, 40));
60. }
62. @Override
63. **public** **void** mouseClicked(MouseEvent e) {
64. // TODO 自动生成的方法存根
65. String PATH = Configuration.getPath();
66. AllFrames.mainUI.getJf().setVisible(**false**);
67. //GameUI gameUI =new GameUI();
68. //gameUI.init();
69. ChooseUI chooseUI = **new** ChooseUI();
70. AllFrames.chooseUI = chooseUI;
71. chooseUI.init(Type.MAN\_MACHINE,PATH);
72. }
73. });
75. jb2 = **new** JButton("人人游戏");
76. jb2.setFont(**new** Font("华文行楷", Font.BOLD, 30));
77. jb2.setBorderPainted(**false**);//不打印边框
78. jb2.setBorder(**null**);//除去边框
79. jb2.setFocusPainted(**false**);//除去焦点的框
80. jb2.setContentAreaFilled(**false**);
81. jb2.addMouseListener(**new** MouseAdapter() {
83. @Override
84. **public** **void** mouseExited(MouseEvent e) {
85. // TODO 自动生成的方法存根
86. jb2.setFont(**new** Font("华文行楷", Font.BOLD, 30));
87. }
89. @Override
90. **public** **void** mouseEntered(MouseEvent e) {
91. // TODO 自动生成的方法存根
92. jb2.setBounds(115, 150, 170, 50);
93. jb2.setFont(**new** Font("华文行楷", Font.BOLD, 40));
94. }
96. @Override
97. **public** **void** mouseClicked(MouseEvent e) {
98. // TODO 自动生成的方法存根
99. //AllFrames.mainUI.getJf().setEnabled(false);
100. //ChangeBackground changeBackground =new ChangeBackground();
101. AllFrames.mainUI.getJf().setVisible(**false**);
102. ChooseUI chooseUI = **new** ChooseUI();
103. //AllFrames.changeBackground=changeBackground;
104. //changeBackground.init();
105. chooseUI.init(Type.MAN\_MAN,Configuration.PATH);
106. }
107. });
109. jb3 = **new** JButton("退出游戏");
110. jb3.setFont(**new** Font("华文行楷", Font.BOLD, 30));
111. //  jb3.setSize(100, 50);
112. jb3.setBorderPainted(**false**);//不打印边框
113. jb3.setBorder(**null**);//除去边框
114. jb3.setFocusPainted(**false**);//除去焦点的框
115. jb3.setContentAreaFilled(**false**);
116. jb3.addMouseListener(**new** MouseAdapter() {
118. @Override
119. **public** **void** mouseExited(MouseEvent e) {
120. // TODO 自动生成的方法存根
121. jb3.setFont(**new** Font("华文行楷", Font.BOLD, 30));
122. }
124. @Override
125. **public** **void** mouseEntered(MouseEvent e) {
126. // TODO 自动生成的方法存根
127. jb3.setBounds(115, 250, 170, 50);
128. jb3.setFont(**new** Font("华文行楷", Font.BOLD, 40));
129. }
131. @Override
132. **public** **void** mouseClicked(MouseEvent e) {
133. // TODO 自动生成的方法存根
134. System.exit(0);
135. }
136. });

139. jb4 = **new** JButton("关       于");
140. jb4.setFont(**new** Font("华文行楷", Font.BOLD, 30));
141. jb4.setBorderPainted(**false**);//不打印边框
142. jb4.setBorder(**null**);//除去边框
143. jb4.setFocusPainted(**false**);//除去焦点的框
144. jb4.setContentAreaFilled(**false**);
145. jb4.addMouseListener(**new** MouseAdapter() {
147. @Override
148. **public** **void** mouseExited(MouseEvent e) {
149. // TODO 自动生成的方法存根
150. jb4.setFont(**new** Font("华文行楷", Font.BOLD, 30));
151. }
153. @Override
154. **public** **void** mouseEntered(MouseEvent e) {
155. // TODO 自动生成的方法存根
156. jb4.setBounds(115, 350, 170, 50);
157. jb4.setFont(**new** Font("华文行楷", Font.BOLD, 40));
158. }
160. @Override
161. **public** **void** mouseClicked(MouseEvent e) {
162. // TODO 自动生成的方法存根
163. jf.setEnabled(**false**);
164. AboutUI aboutUI = **new** AboutUI();
165. aboutUI.init();
166. }
167. });
169. jp.add(jb1);
170. jb1.setBounds(125, 50, 150, 50);
171. jp.add(jb2);
172. jb2.setBounds(125, 150, 150, 50);
173. jp.add(jb3);
174. jb3.setBounds(125, 250, 150, 50);
175. jp.add(jb4);
176. jb4.setBounds(125, 350, 150, 50);
178. jf.add(jp);
179. jf.setVisible(**true**);//方法的意思是说数据模型已经构造好了，允许JVM可以根据数据模型执行paint方法开始画图并显示到屏幕上了，并不是显示图形，而是可以运行开始画图了。
180. jf.repaint();
181. }
183. **public** JFrame getJf() {
184. **return** jf;
185. }
187. **public** **void** setJf(JFrame jf) {
188. **this**.jf = jf;
189. }
191. **public** **static** **void** main(String[] args) {
192. String PATH = Configuration.getPath();
193. // TODO 自动生成的方法存根
194. rock\_paper\_scissors mainUI = **new** rock\_paper\_scissors();
195. mainUI.init(PATH);
196. AllFrames.mainUI = mainUI;
197. //System.out.println(System.getProperty("java.class.path"));
198. }
199. }

游戏主界面的样式编写及鼠标事件的响应代码，其绑定了ChooseUI.java和AboutUI.java，运行跳转界面进入游戏模式的选择和游戏说明。程序运行界面如图1.1 游戏主界面。

init(String path)函数中的参数为接收主界面背景的图片位置；函数体内容主要是实现主界面并添加了鼠标悬浮效果以及点击触发的响应。main函数入口在此类中，main函数调用AllFrames事先定义好的来保证安全性。

**2.ChooseUI.java**

1. **package** rock\_paper\_scissors;
3. **import** java.awt.Font;
4. **import** java.awt.event.MouseAdapter;
5. **import** java.awt.event.MouseEvent;
6. **import** java.io.BufferedReader;
7. **import** java.io.File;
8. **import** java.io.FileReader;
9. **import** java.io.IOException;
10. **import** java.util.ArrayList;
11. **import** java.util.HashMap;
12. **import** java.util.Map;
13. **import** javax.swing.JButton;
14. **import** javax.swing.JFrame;
16. **public** **class** ChooseUI {
17. JFrame jf;
18. JButton jb1, jb2, jb3, jb4;
19. MyPanel jp;
21. **public** **void** init(Type type,String path) {//初始化
22. jf = **new** JFrame();//东西南北中，五个位置。
23. jp = **new** MyPanel();
24. **if** (path == **null**) {
25. MyPanel.path = "src\\rock\_paper\_scissors\\背景0.jpg";
26. } **else** {
27. MyPanel.path = path;
28. }
29. //jp.setOpaque(false);将jpanel设置为透明将显示jfreame的背景颜色
30. jp.setSize(400 + 8 + 8, 400 + 31 + 8);
31. jf.setTitle("模式选择");
32. jf.setSize(400 + 8 + 8, 400 + 31);
33. jf.setResizable(**false**);//设置此窗体是否可由用户调整大小。
34. jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
36. jf.setLocationRelativeTo(**null**);
37. jf.setLayout(**null**);
38. jp.setLayout(**null**);//jpanel为流式布局改为空
40. Map<Integer,JButton> jButtonMap = **new** HashMap<>();//key存储init的参数
41. jb1 = **new** JButton("一局胜负");
42. jb2 = **new** JButton("三局两胜");
43. jb3 = **new** JButton("五局三胜");
44. jb4 = **new** JButton("无尽之战");
46. jButtonMap.put(1,jb1);
47. jButtonMap.put(3,jb2);
48. jButtonMap.put(5,jb3);
49. jButtonMap.put(10000,jb4);
50. **for**(Map.Entry<Integer,JButton> entry : jButtonMap.entrySet()){
51. entry.getValue().setFont(**new** Font("华文行楷", Font.BOLD, 30));
52. //  jb1.setSize(100, 50);
53. entry.getValue().setBorderPainted(**false**);//不打印边框
54. entry.getValue().setBorder(**null**);//除去边框
55. entry.getValue().setFocusPainted(**false**);//除去焦点的框
56. entry.getValue().setContentAreaFilled(**false**);
57. entry.getValue().addMouseListener(**new** MouseAdapter() {
59. @Override
60. **public** **void** mouseExited(MouseEvent e) {
61. // TODO 自动生成的方法存根
62. entry.getValue().setFont(**new** Font("华文行楷", Font.BOLD, 30));
63. }
65. @Override
66. **public** **void** mouseEntered(MouseEvent e) {
67. // TODO 自动生成的方法存根
68. //entry.getValue().setBounds(115, 50, 170, 50);
69. entry.getValue().setFont(**new** Font("华文行楷", Font.BOLD, 40));
70. }
72. @Override
73. **public** **void** mouseClicked(MouseEvent e) {
74. // TODO 自动生成的方法存根
75. jf.setVisible(**false**);
76. //AllFrames.mainUI.getJf().setVisible(false);
77. **if**(type == Type.MAN\_MACHINE){
78. GameUI gameUI = **new** GameUI();
79. gameUI.init(entry.getKey());
80. }**else** **if**(type == Type.MAN\_MAN){
81. HumanUI humanUI = **new** HumanUI();
82. humanUI.init(entry.getKey());
83. AllFrames.humanUI = humanUI;
84. }
85. }
86. });
87. }
89. jp.add(jb1);
90. jb1.setBounds(125, 50, 170, 50);
91. jp.add(jb2);
92. jb2.setBounds(125, 150, 170, 50);
93. jp.add(jb3);
94. jb3.setBounds(125, 250, 170, 50);
95. jp.add(jb4);
96. jb4.setBounds(125, 350, 170, 50);
98. jf.add(jp);
99. jf.setVisible(**true**);//方法的意思是说数据模型已经构造好了，允许JVM可以根据数据模型执行paint方法开始画图并显示到屏幕上了，并不是显示图形，而是可以运行开始画图了。
100. jf.repaint();
101. }
103. **public** JFrame getJf() {
104. **return** jf;
105. }
107. **public** **void** setJf(JFrame jf) {
108. **this**.jf = jf;
109. }
110. }

游戏模式选择界面，每一个选项给GameUI.java和HumaUI.java中传递不同的参数作为对战局数的选择，点击进入游戏界面。游戏模式选择界面是在主界面的基础上，改动触鼠标点击调用的函数，每种模式的区别近为调用函数的形参值不同。

**3.HumanUI.java**

1. **package** rock\_paper\_scissors;
3. **import** java.awt.Color;
4. **import** java.awt.Font;
5. **import** java.awt.event.ActionEvent;
6. **import** java.awt.event.ActionListener;
7. **import** java.awt.event.MouseAdapter;
8. **import** java.awt.event.MouseEvent;
9. **import** java.awt.event.MouseListener;
10. **import** java.awt.event.WindowAdapter;
11. **import** java.awt.event.WindowEvent;
12. **import** java.awt.event.WindowListener;
13. **import** java.awt.event.KeyEvent;
14. **import** java.awt.event.KeyListener;
15. **import** java.awt.event.KeyAdapter;
16. **import** javax.swing.\*;
18. **public** **class** HumanUI {
19. JPanel jPanel;
20. JFrame jf;
21. JButton jb, jb1, jb2, jb3;
22. JTextArea jta1, jta2;
23. **int** winNUM = 0;
24. **int** drawNUM = 0;
25. **int** lostNUM = 0;
26. Clock clock;
27. Select left;
28. Select right;
29. **boolean** leftLock = **false**;//锁，如果已经有了数字，就给上锁
30. **boolean** rightLock = **false**;
31. **int** returnNUM = 0;
33. **public** **void** init(**int** num) {
34. jf = **new** JFrame();
35. jPanel = **new** JPanel();
36. jPanel.setLayout(**null**);
37. jPanel.setSize(306, 250 + 31 + 8);
39. jf.getContentPane().add(jPanel);
40. //jp.setSize(200+8+8, 200+31+8);
41. jf.setTitle("正在游戏");
42. jf.setSize(306, 250 + 31 + 8);
43. jf.setResizable(**false**);
44. jf.setLocationRelativeTo(**null**);
45. jf.setLayout(**null**);
47. jb1 = **new** JButton();//human1猜拳
48. jb1.setBorderPainted(**false**);//不打印边框
49. jb1.setBorder(**null**);//除去边框
50. jb1.setFocusPainted(**false**);//除去焦点的框
51. jb1.setContentAreaFilled(**false**);
52. //jb1.setIcon(new ImageIcon());
53. jPanel.add(jb1);
54. jb1.setBounds(73, 65, 73, 65);
56. jb2 = **new** JButton();//human2猜拳
57. jb2.setBorderPainted(**false**);//不打印边框
58. jb2.setBorder(**null**);//除去边框
59. jb2.setFocusPainted(**false**);//除去焦点的框
60. jb2.setContentAreaFilled(**false**);
61. jPanel.add(jb2);
62. jb2.setBounds(145, 65, 73, 65);
64. **for** (**int** i = 1; i <= 3; i++) {
65. jb = **new** JButton();
66. jb.addKeyListener(**new** KeyAdapter() {
67. **public** **void** keyPressed(KeyEvent e) {
68. **if** (leftLock) {//如果上锁了，直接返回
69. ;
70. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_A) {
71. jb1.setIcon(**new** ImageIcon(getClass().getResource(4 + ".png")));
72. left = Select.PAPER;
73. leftLock = **true**;
74. **if**(clock==**null**){
75. clock = **new** Clock(Clock.RIGHT);
76. }
77. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_S) {
78. jb1.setIcon(**new** ImageIcon(getClass().getResource(4 + ".png")));
79. left = Select.SCISSORS;
80. leftLock = **true**;
81. **if**(clock==**null**){
82. clock = **new** Clock(Clock.RIGHT);
83. }
84. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_D) {
85. jb1.setIcon(**new** ImageIcon(getClass().getResource(4 + ".png")));
86. left = Select.ROCK;
87. leftLock = **true**;
88. **if**(clock==**null**){
89. clock = **new** Clock(Clock.RIGHT);
90. }
91. }
92. **if** (rightLock) {//如果上锁了，直接返回
93. ;
94. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_LEFT) {
95. jb2.setIcon(**new** ImageIcon(getClass().getResource(5 + ".png")));
96. right = Select.PAPER;
97. rightLock = **true**;
98. **if**(clock==**null**){
99. clock = **new** Clock(Clock.LEFT);
100. }
101. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_DOWN) {
102. jb2.setIcon(**new** ImageIcon(getClass().getResource(5 + ".png")));
103. right = Select.SCISSORS;
104. rightLock = **true**;
105. **if**(clock==**null**){
106. clock = **new** Clock(Clock.LEFT);
107. }
108. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_RIGHT) {
109. jb2.setIcon(**new** ImageIcon(getClass().getResource(5 + ".png")));
110. right = Select.ROCK;
111. rightLock = **true**;
112. **if**(clock==**null**){
113. clock = **new** Clock(Clock.LEFT);
114. }
115. }
116. **if** (operate()) {
117. leftLock = rightLock = **false**;//解锁
118. jta1.setText("甲：赢：" + winNUM + "次     平：" + drawNUM + "次      输：" + lostNUM + "次");
119. jta2.setText("乙：赢：" + lostNUM + "次     平：" + drawNUM + "次      输：" + winNUM + "次");
120. }
121. jf.repaint();


125. **if** (winNUM==num/2+1 ) {
126. **int** res = JOptionPane.showConfirmDialog(**null**, "恭喜甲赢了！！！\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
127. **if** (res == JOptionPane.YES\_OPTION) {
128. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
129. winNUM = drawNUM = lostNUM = 0;

132. } **else** {
133. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
134. //return;
135. AllFrames.mainUI.getJf().setVisible(**true**);
136. closeAction();
137. jf.dispose();
138. }
139. } **else** **if** (lostNUM==num/2+1 ) {
140. **int** res = JOptionPane.show这是左右哪边的1ConfirmDialog(**null**, "恭喜乙赢了！！！\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
141. **if** (res == JOptionPane.YES\_OPTION) {
142. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
143. winNUM = drawNUM = lostNUM = 0;

146. } **else** {
147. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
148. //return;
149. AllFrames.mainUI.getJf().setVisible(**true**);
150. closeAction();
151. jf.dispose();
152. }
153. }
154. }

157. });
159. jb.setBorderPainted(**false**);//不打印边框
160. jb.setBorder(**null**);//除去边框
161. jb.setFocusPainted(**false**);//除去焦点的框
162. jb.setContentAreaFilled(**false**);
163. jb.setIcon(**new** ImageIcon(getClass().getResource(i + ".png")));
164. jPanel.add(jb);
165. jb.setBounds(0, (i - 1) \* 65, 73, 65);
166. }
168. **for** (**int** i = 1; i <= 3; i++) {
169. jb = **new** JButton();
170. jb.setBorderPainted(**false**);//不打印边框
171. jb.setBorder(**null**);//除去边框
172. jb.setFocusPainted(**false**);//除去焦点的框
173. jb.setContentAreaFilled(**false**);
174. jb.setIcon(**new** ImageIcon(getClass().getResource(i + ".png")));
175. jPanel.add(jb);
176. jb.setBounds(219, (i - 1) \* 65, 73, 65);
177. }
179. jb3 = **new** JButton("结束游戏");
180. jb3.setBorderPainted(**false**);
181. jb3.setBorder(**null**);
182. jb3.setFocusable(**false**);
183. jb3.setContentAreaFilled(**false**);
184. jb3.setFont(**new** Font("华文行楷", Font.BOLD, 20));
185. jb3.addMouseListener(**new** MouseAdapter() {
186. @Override
187. **public** **void** mouseExited(MouseEvent e) {
188. // TODO 自动生成的方法存根
189. jb3.setForeground(Color.black);
190. }
191. @Override
192. **public** **void** mouseEntered(MouseEvent e) {
193. // TODO 自动生成的方法存根
194. jb3.setForeground(Color.blue);
195. }
196. @Override
197. **public** **void** mouseClicked(MouseEvent e) {
198. // TODO 自动生成的方法存根
199. AllFrames.mainUI.getJf().setVisible(**true**);
200. closeAction();
201. jf.dispose();
202. }
203. });
204. jPanel.add(jb3);
205. jb3.setBounds(96, 200, 100, 40);
207. jta1 = **new** JTextArea("甲：赢：  平： 输： ");
208. jta1.setLineWrap(**true**);
209. jta1.setEditable(**false**);
210. jPanel.add(jta1);
211. jta1.setBounds(73, 0, 145, 65);
213. jta2 = **new** JTextArea("乙：赢：  平： 输： ");
214. jta2.setLineWrap(**true**);
215. jta2.setEditable(**false**);
216. jPanel.add(jta2);
217. jta2.setBounds(73, 130, 145, 65);
219. jf.addWindowListener(**new** WindowAdapter() {
221. @Override
222. **public** **void** windowClosing(WindowEvent e) {
223. // TODO 自动生成的方法存根
224. AllFrames.mainUI.getJf().setVisible(**true**);
225. closeAction();
226. jf.dispose();
227. }
229. });
230. jf.setVisible(**true**);
231. }
233. //当两边都输入时，试图计算结果.返回true说明计算成功了
234. **public** **boolean** operate() {
235. **if** (leftLock && rightLock) {
236. clock.setVisible(**false**);
237. clock = **null**;
238. Result r = Tool.judge(left, right);
239. **switch** (r) {
240. **case** LEFT\_WIN:
241. winNUM++;
242. **break**;
243. **case** RIGHT\_WIN:
244. lostNUM++;
245. **break**;
246. **case** ALL\_WIN:
247. drawNUM++;
248. }
249. **switch** (left) {
250. **case** ROCK:
251. jb1.setIcon(**new** ImageIcon(getClass().getResource(3 + ".png")));
252. **break**;
253. **case** SCISSORS:
254. jb1.setIcon(**new** ImageIcon(getClass().getResource(2 + ".png")));
255. **break**;
256. **case** PAPER:
257. jb1.setIcon(**new** ImageIcon(getClass().getResource(1 + ".png")));
258. **break**;
259. }
260. **switch** (right) {
261. **case** ROCK:
262. jb2.setIcon(**new** ImageIcon(getClass().getResource(3 + ".png")));
263. **break**;
264. **case** SCISSORS:
265. jb2.setIcon(**new** ImageIcon(getClass().getResource(2 + ".png")));
266. **break**;
267. **case** PAPER:
268. jb2.setIcon(**new** ImageIcon(getClass().getResource(1 + ".png")));
269. **break**;
270. }
271. **return** **true**;
272. }**else** **if**(leftLock){
273. jPanel.add(clock);
274. }**else** **if**(rightLock){
275. jPanel.add(clock);
276. }
277. **return** **false**;
278. }
279. **public** **void** reStart(){
280. winNUM = lostNUM = drawNUM = 0;
281. leftLock = rightLock = **false**;
282. jb1.setIcon(**null**);
283. jb2.setIcon(**null**);
284. clock = **null**;
285. }
287. }

两人对战的游戏界面及输赢逻辑处理部分。当一方的胜利次数到达num(总局数)/2+1次且另外一人胜利次数比其少时判断游戏输赢，避免了必须完成num轮游戏后判断输赢的呆板体验。在输赢结果出来之后会弹出结果框，点击“是”则继续游戏，点击“否”则返回到游戏主界面。

代码设计中，通过添加键盘的监听器实现对按键的监听。当左方玩家按下A、S、D时代表左方出了布、剪刀、石头，右方玩家按下←、↓、→时同理，具体操作为，给left和right赋值Select类型的值。之后运行operate()函数进行胜负判断相应处理等行动。在添加防作弊功能时，注意到游戏的流畅性，于是“禁止重复按下”的设计。具体到代码上，是用布尔属性leftLock和rightLock储存是否上锁（禁止进行按键处理）的信息，同时按键按下时将对应的属性赋值为true.

在operate（）中，若左右同时上锁——代表着左右两方玩家均已按下，则调用工具类的judge（left,right）来进行胜负判定，根据返回的信息来进行图片的显示。若只有一方按下，则只进行图片的显示。

在restart()中，进行整个界面的重启。包括让倒计时消失，胜利局数的归零，leftLock和rightLock同时置false以及让图片消失。

**4.AI.java**

1. **package** rock\_paper\_scissors;
3. **import** java.util.Random;
5. **public** **class** AI {
6. **static** **int** winNUM=0;
7. **static** **int** drawNUM=0;
8. **static** **int** lostNUM=0;
10. //1布2剪刀3石头  -1输 0平 1赢
11. **static** Random random =**new** Random();
12. **public** **static** **int** ai(**int** num) {
13. **int** rand=random.nextInt(1200)+1;
14. **switch** (rand%3+1) {//1布2剪刀3石头ai出上次玩家出的
15. **case** 1:
16. **switch** (num) {
17. **case** 1:
18. drawNUM++;
20. **break**;
21. **case** 2:
22. winNUM++;
24. **break**;
25. **case** 3:
26. lostNUM++;
28. **break**;
30. }
32. **break**;
34. **case** 2://1布2剪刀3石头
35. **switch** (num) {
36. **case** 1:
37. lostNUM++;
39. **break**;
40. **case** 2:
41. drawNUM++;
43. **break**;
44. **case** 3:
45. winNUM++;
47. **break**;
49. }
50. **break**;
52. **case** 3://1布2剪刀3石头
53. **switch** (num) {
54. **case** 1:
55. winNUM++;
57. **break**;
58. **case** 2:
59. lostNUM++;
61. **break**;
62. **case** 3:
63. drawNUM++;
65. **break**;
67. }
68. **break**;
69. }
71. **return** rand%3+1;
72. }
73. }

机器出拳的思路实现，在GameUI中调用此类的init()函数来实现机器的出拳选择。

**5.Clock.java**

1. **package** rock\_paper\_scissors;
3. **import** javax.swing.\*;
4. **import** java.awt.\*;
6. //倒计时
7. **public** **class** Clock **extends** JPanel **implements** Runnable {
8. **private** **int** num;//到达0秒时判负
9. //static int i = 1;
10. **public** **final** **static** **int** LEFT = 0;//左边的倒计时
11. **public** **final** **static** **int** RIGHT = 1;//右边的倒计时
12. **private** Thread thread;
13. **private** **int** position;
14. **private** JLabel jLabel;
15. **private** **boolean** isStop;
17. **public** Clock(**int** where) {
18. **this**.num = 3;
19. position = where;
20. **this**.jLabel = **new** JLabel("还有" + num + "秒！");
21. jLabel.setForeground(Color.RED);
22. **if**(where == LEFT){
23. **this**.setBounds(22, 200, 100, 40);
24. }**else**{
25. **this**.setBounds(188, 200, 100, 40);
26. }
27. //System.out.println("?"+i++);
28. **this**.add(jLabel);
29. thread = **new** Thread(**this**);
30. isStop = **false**;
31. thread.start();
32. }
34. @Override
35. **public** **void** run() {
36. **while** (!isStop){
37. **try**{
38. Thread.sleep(1000);
39. }**catch** (Exception e){
40. e.printStackTrace();
41. }
42. num--;
43. jLabel.setText("还有" + num + "秒！");
44. **if**(num == 0 && **this**.isVisible()){
45. String message = "";
46. **if** (position == LEFT) {
47. message = "乙";
48. } **else** **if**(position == RIGHT){
49. message = "甲";
50. }
51. **int** res = JOptionPane.showConfirmDialog(**null**, "恭喜"+message+"赢了！！！\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
52. **if** (res == JOptionPane.YES\_OPTION) {
53. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
54. AllFrames.humanUI.reStart();
56. } **else** {
57. AllFrames.mainUI.getJf().setVisible(**true**);
58. AllFrames.humanUI.closeAction();
59. AllFrames.humanUI.jf.dispose();
60. }
61. stop();
62. }
63. }
64. }
66. **public** **void** stop(){
67. isStop = **true**;
68. **this**.setVisible(**false**);
69. thread = **null**;
70. }
71. }

倒计时三秒功能的实现，在HumanUI.java中一方做出选择时调用此函数来限定另一方3s内做出选择。

设置两个属性LEFT和RIGHT用以表示这是左右哪方的倒计时。同时接受外部参数position（参数值为LEFT或者RIGHT）来初始化显示的Label的位置。此处不使用enum而是直接使用静态的int来表现该属性是因为这项属性仅用于该类。

内置方法stop(),调用他可将run()方法内的循环判定条件置为false且进行善后工作。

根据设计，在倒计时结束时仍未满足条件时将弹出失败的弹窗，在点击确定按钮后将重置对局。在代码的体现是，点击确定后在AllFrame类中获得HumanUI类的唯一实例humanUI,调用其方法restart()（该方法完成对局的重置）。

在倒计时结束时，先判定是否显示，若不显示，则不进行后续操作。

**6.AboutUI.java**

1. **package** rock\_paper\_scissors;
3. **import** java.awt.Color;
4. **import** java.awt.Desktop;
5. **import** java.awt.Font;
6. **import** java.awt.event.MouseAdapter;
7. **import** java.awt.event.MouseEvent;
8. **import** java.awt.event.WindowAdapter;
9. **import** java.awt.event.WindowEvent;
10. **import** java.awt.event.WindowListener;
11. **import** java.io.IOException;
12. **import** java.net.URI;
13. **import** java.net.URISyntaxException;
15. **import** javax.swing.JButton;
16. **import** javax.swing.JEditorPane;
17. **import** javax.swing.JFrame;
18. **import** javax.swing.JLabel;
19. **import** javax.swing.JTextArea;
21. **public** **class** AboutUI {
22. JFrame jf;
23. JTextArea jta;
24. JButton jb;
26. **public** **void** init() {
28. jf=**new** JFrame();
29. jf.setTitle("关于");
30. jf.setSize(300,300);
31. jf.setResizable(**false**);
33. jf.setLocationRelativeTo(**null**);
34. jf.setLayout(**null**);
36. jta =**new** JTextArea("        人机游戏：即可鼠标操作，又可按键操作(left代表PAPER，down代表SCISSORS，right代表ROCK);\n\n        人人游戏：甲玩家(A键代表PAPER，S键代表SCISSORS，D键代表ROCK)；乙玩家(left代表PAPER，down代表SCISSORS，right代表ROCK)",5,10);
37. jta.setLineWrap(**true**);
38. jta.setEditable(**false**);
39. jf.add(jta);
40. jta.setBounds(0, 0, 285, 250);
42. jf.addWindowListener(**new** WindowListener() {
44. @Override
45. **public** **void** windowOpened(WindowEvent e) {
46. // TODO 自动生成的方法存根
47. }
49. @Override
50. **public** **void** windowIconified(WindowEvent e) {
51. // TODO 自动生成的方法存根
52. }
54. @Override
55. **public** **void** windowDeiconified(WindowEvent e) {
56. // TODO 自动生成的方法存根
57. }
59. @Override
60. **public** **void** windowDeactivated(WindowEvent e) {
61. // TODO 自动生成的方法存根
63. }
65. @Override
66. **public** **void** windowClosing(WindowEvent e) {
67. // TODO 自动生成的方法存根
68. AllFrames.mainUI.getJf().setEnabled(**true**);
69. }
70. @Override
71. **public** **void** windowClosed(WindowEvent e) {
72. // TODO 自动生成的方法存根
73. }
75. @Override
76. **public** **void** windowActivated(WindowEvent e) {
77. // TODO 自动生成的方法存根
78. }
79. });
80. jf.setVisible(**true**);
81. }
83. **public** JFrame getJf() {
84. **return** jf;
85. }
87. **public** **void** setJf(JFrame jf) {
88. **this**.jf = jf;
89. }
90. }

游戏的关于界面的样式编写，展示了游戏相关的信息，同时提供了返回主界面的功能。改界面的主要目的就是向玩家展示该游戏程序的玩法，键盘按键对应的出拳手势，提前让用户熟悉游戏玩法。

**7.GameUI.java**

1. **package** rock\_paper\_scissors;
3. **import** java.awt.Color;
4. **import** java.awt.Font;
5. **import** java.awt.event.ActionEvent;
6. **import** java.awt.event.ActionListener;
7. **import** java.awt.event.MouseAdapter;
8. **import** java.awt.event.MouseEvent;
9. **import** java.awt.event.MouseListener;
10. **import** java.awt.event.WindowAdapter;
11. **import** java.awt.event.WindowEvent;
12. **import** java.awt.event.WindowListener;
13. **import** java.awt.event.KeyAdapter;
14. **import** java.awt.event.KeyEvent;
16. **import** javax.swing.JOptionPane;
18. **import** javax.swing.ImageIcon;
19. **import** javax.swing.JButton;
20. **import** javax.swing.JFrame;
21. **import** javax.swing.JPanel;
22. **import** javax.swing.JTextArea;
24. **public** **class** GameUI {
25. JFrame jf;
26. JButton jb, jb1, jb2, jb3;
27. JTextArea jta;
28. **int** returnNUM = 0;
30. // static int NUM;
31. **public** **void** init(**int** num) {
32. jf = **new** JFrame();
34. //jp.setSize(200+8+8, 200+31+8);
35. jf.setTitle("正在游戏");
36. jf.setSize(217 + 8 + 8, 250 + 31 + 8);
37. jf.setResizable(**false**);
38. jf.setLocationRelativeTo(**null**);
39. jf.setLayout(**null**);
40. **for** (**int** i = 1; i <= 3; i++) {
41. jb = **new** JButton();
42. **final** **int** j = i;
43. jb.addActionListener(**new** ActionListener() {
44. @Override
45. **public** **void** actionPerformed(ActionEvent e) {
46. // TODO 自动生成的方法存根
47. jb1.setIcon(**new** ImageIcon(getClass().getResource(j + ".png")));
48. returnNUM = AI.ai(j);
49. jb2.setIcon(**new** ImageIcon("src\\rock\_paper\_scissors\\" + returnNUM + ".png"));
51. jta.setText("赢：" + AI.winNUM + "次     平：" + AI.drawNUM + "次      输：" + AI.lostNUM + "次");
52. jf.repaint();

55. **if** (AI.winNUM == num / 2 + 1) {
56. **int** res = JOptionPane.showConfirmDialog(**null**, "恭喜您赢了！！！\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
57. **if** (res == JOptionPane.YES\_OPTION) {
58. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
59. AI.winNUM = AI.drawNUM = AI.lostNUM = 0;
61. } **else** {
62. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
63. //return;
64. AllFrames.mainUI.getJf().setVisible(**true**);
65. closeAction();
66. jf.dispose();
67. }
68. } **else** **if** (AI.lostNUM == num / 2 + 1 ) {
69. **int** res = JOptionPane.showConfirmDialog(**null**, "遗憾您输了。。。\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
70. **if** (res == JOptionPane.YES\_OPTION) {
71. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
72. AI.winNUM = AI.drawNUM = AI.lostNUM = 0;

75. } **else** {
76. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
77. //return;
78. AllFrames.mainUI.getJf().setVisible(**true**);
79. closeAction();
80. jf.dispose();
81. }
82. }
83. }
85. });
87. jb.addKeyListener(**new** KeyAdapter() {
88. **public** **void** keyPressed(KeyEvent e) {
89. **if** (e.getKeyCode() == KeyEvent.VK\_LEFT) {
90. jb1.setIcon(**new** ImageIcon(getClass().getResource(1 + ".png")));
91. returnNUM = AI.ai(1);
92. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_DOWN) {
93. jb1.setIcon(**new** ImageIcon(getClass().getResource(2 + ".png")));
94. returnNUM = AI.ai(2);
95. } **else** **if** (e.getKeyCode() == KeyEvent.VK\_RIGHT) {
96. jb1.setIcon(**new** ImageIcon(getClass().getResource(3 + ".png")));
97. returnNUM = AI.ai(3);
98. }
100. jb2.setIcon(**new** ImageIcon("src\\rock\_paper\_scissors\\" + returnNUM + ".png"));
101. jta.setText("赢：" + AI.winNUM + "次     平：" + AI.drawNUM + "次      输：" + AI.lostNUM + "次");
102. jf.repaint();
104. **if** (AI.winNUM == num / 2 + 1 && AI.lostNUM <= num / 2) {
105. **int** res = JOptionPane.showConfirmDialog(**null**, "恭喜您赢了！！！\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
106. **if** (res == JOptionPane.YES\_OPTION) {
107. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
108. AI.winNUM = AI.drawNUM = AI.lostNUM = 0;
110. } **else** {
111. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
112. //return;
113. AllFrames.mainUI.getJf().setVisible(**true**);
114. closeAction();
115. jf.dispose();
116. }
117. } **else** **if** (AI.lostNUM == num / 2 + 1 && AI.winNUM <= num / 2) {
118. **int** res = JOptionPane.showConfirmDialog(**null**, "遗憾您输了。。。\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
119. **if** (res == JOptionPane.YES\_OPTION) {
120. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
121. AI.winNUM = AI.drawNUM = AI.lostNUM = 0;

124. } **else** {
125. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
126. //return;
127. AllFrames.mainUI.getJf().setVisible(**true**);
128. closeAction();
129. jf.dispose();
130. }
131. }
133. }
134. });
136. jb.setBorderPainted(**false**);//不打印边框
137. jb.setBorder(**null**);//除去边框
138. jb.setFocusPainted(**false**);//除去焦点的框
139. jb.setContentAreaFilled(**false**);
140. jb.setIcon(**new** ImageIcon(getClass().getResource(i + ".png")));
141. jf.add(jb);
142. jb.setBounds(0, (i - 1) \* 65, 73, 65);
143. }
145. jb1 = **new** JButton();
146. jb1.setBorderPainted(**false**);//不打印边框
147. jb1.setBorder(**null**);//除去边框
148. jb1.setFocusPainted(**false**);//除去焦点的框
149. jb1.setContentAreaFilled(**false**);
150. //jb1.setIcon(new ImageIcon());
151. jf.add(jb1);
152. jb1.setBounds(73, 65, 73, 65);
154. jb2 = **new** JButton();//ai猜拳
155. jb2.setBorderPainted(**false**);//不打印边框
156. jb2.setBorder(**null**);//除去边框
157. jb2.setFocusPainted(**false**);//除去焦点的框
158. jb2.setContentAreaFilled(**false**);
159. jf.add(jb2);
160. jb2.setBounds(145, 65, 73, 65);

163. jb3 = **new** JButton("结束游戏");
164. jb3.setBorderPainted(**false**);
165. jb3.setBorder(**null**);
166. jb3.setFocusable(**false**);
167. jb3.setContentAreaFilled(**false**);
168. jb3.setFont(**new** Font("华文行楷", Font.BOLD, 20));
169. jb3.addMouseListener(**new** MouseAdapter() {

172. @Override
173. **public** **void** mouseExited(MouseEvent e) {
174. // TODO 自动生成的方法存根
175. jb3.setForeground(Color.black);
176. }
178. @Override
179. **public** **void** mouseEntered(MouseEvent e) {
180. // TODO 自动生成的方法存根
181. jb3.setForeground(Color.blue);
182. }
184. @Override
185. **public** **void** mouseClicked(MouseEvent e) {
186. // TODO 自动生成的方法存根
187. AllFrames.mainUI.getJf().setVisible(**true**);
188. closeAction();
189. jf.dispose();
190. }
191. });
192. jf.add(jb3);
193. jb3.setBounds(60, 200, 100, 40);
195. jta = **new** JTextArea("赢：  平： 输： ");
196. jta.setLineWrap(**true**);
197. jta.setEditable(**false**);
198. jf.add(jta);
199. jta.setBounds(73, 0, 140, 65);
201. jf.addWindowListener(**new** WindowAdapter() {
203. @Override
204. **public** **void** windowClosing(WindowEvent e) {
205. // TODO 自动生成的方法存根
206. AllFrames.mainUI.getJf().setVisible(**true**);
207. closeAction();
208. jf.dispose();
209. }
211. });
212. jf.setVisible(**true**);
213. }
215. **public** **void** closeAction() {
216. AI.drawNUM = AI.lostNUM = AI.winNUM = 0;
218. }
219. }

人机游戏模式的代码，该类的init(int num)函数的形参为ChooseUI中传递过来的游戏局数，分别对应四种游戏模式；在jb.addActionListener()的参数中重写鼠标和按键的触发响应事件，即输赢判断，函数体体中实现将布、剪刀、石头可视化游戏显示在游戏屏幕上。接下来的代码就是游戏界面的绘制。

代码设计中，处理监听鼠标的点击之外，通过添加键盘的监听器实现对按键的监听。当玩家通过鼠标点击游戏界面的布、剪刀、石头的图标或者按下←、↓、→时代表出了布、剪刀、石头。

**8.Result.java**

1. **package** rock\_paper\_scissors;
3. **public** **enum** Result {
4. LEFT\_WIN,
5. RIGHT\_WIN,
6. ALL\_WIN
7. }

枚举类型Result,用以表示胜负的判断类型。其中ALL\_WIN代表平局。用于HumanUI.java,Tool.java等。

**9.Select.java**

1. **package** rock\_paper\_scissors;
3. **public** **enum** Select {
4. ROCK,
5. PAPER,
6. SCISSORS
7. }

枚举类型Select,用以表示选择的出拳。用于HumanUI.java等

**10.Type.java**

1. **package** rock\_paper\_scissors;
3. **public** **enum** Type {
4. MAN\_MAN,
5. MAN\_MACHINE
6. }

枚举类型Type,用以表示生成的类型的判断类型。用于ChooseUI.java和主函数。

**11.Tool.java**

1. **package** rock\_paper\_scissors;
3. //工具类
4. **public** **class** Tool {
5. **public** **static** Result judge(Select A, Select B) {//胜负判断
6. **if** (A == B) {
7. **return** Result.ALL\_WIN;
8. } **else** **if** (
9. (A.equals(Select.ROCK) && B.equals(Select.SCISSORS)) ||
10. (A.equals(Select.SCISSORS) && B.equals(Select.PAPER)) ||
11. (A.equals(Select.PAPER) && B.equals(Select.ROCK))) {
12. **return** Result.LEFT\_WIN;
13. } **else** {
14. **return** Result.RIGHT\_WIN;
15. }
16. }
17. }

工具类，拥有静态方法judge,可以通过两方的出拳类型判别结果。用于HumanUI.java等。

代码实现中，如果A和B相同，则返回ALL\_WIN,代表平局，否则进行更细致的胜负判定。为了简洁起见，可将左边胜利的条件写在一个条件判断中，若符合则返回LEFT\_WIN,不符合则返回RIGHT\_WIN.

**12.Configuration.java**

1. **package** rock\_paper\_scissors;
3. **import** java.io.BufferedReader;
4. **import** java.io.BufferedWriter;
5. **import** java.io.File;
6. **import** java.io.FileReader;
7. **import** java.io.FileWriter;
8. **import** java.io.IOException;
10. **public** **class** Configuration {
11. **static** String PATH = **null**;
13. **public** **static** String getPath() {
14. File file = **new** File("src\\rock\_paper\_scissors\\Configuration.txt");
16. **if** (!file.exists()) {
18. **try** {
19. file.createNewFile();
20. } **catch** (IOException e) {
21. // TODO 自动生成的 catch 块
22. e.printStackTrace();
23. }
24. }
25. **try** {
26. BufferedReader in = **new** BufferedReader(**new** FileReader(file));
27. PATH = in.readLine();
28. in.close();
29. } **catch** (IOException e) {
30. // TODO 自动生成的 catch 块
31. e.printStackTrace();
32. }
33. //System.out.println(PATH);
34. **return** PATH;
35. }
37. **public** **static** **void** setPATH(String pATH) {
38. File file = **new** File("src\\rock\_paper\_scissors\\Configuration.txt");
39. File file1 = **new** File("src\\rock\_paper\_scissors\\Configuration1.txt");
40. **if** (file.exists()) {
41. file.delete();
42. }
43. **if** (!file1.exists()) {
45. **try** {
46. file1.createNewFile();
47. } **catch** (IOException e) {
48. // TODO 自动生成的 catch 块
49. e.printStackTrace();
50. }
51. }
52. **try** {
53. BufferedWriter out = **new** BufferedWriter(**new** FileWriter(file1));
54. out.write(pATH);
55. out.flush();
56. out.close();
57. file1.renameTo(file);
58. } **catch** (IOException e) {
59. // TODO 自动生成的 catch 块
60. e.printStackTrace();
61. }
62. PATH = pATH;
63. }
64. }

配置类，此处仅用于设置背景图片并在文本文档中输入有关信息。

**13.AllFrames.java**

1. **package** rock\_paper\_scissors;
3. **public** **class** AllFrames {
5. **public** **static** rock\_paper\_scissors mainUI;
6. **public** **static** HumanUI humanUI;
7. **public** **static** ChooseUI chooseUI;
8. }

保证这几个类至多只有一个实例。同时方便其他方法访问这些类的实例。

**14.MyPanel.java**

1. **package** rock\_paper\_scissors;
3. **import** java.awt.Graphics;
4. **import** java.awt.Image;
5. **import** javax.swing.ImageIcon;
6. **import** javax.swing.JPanel;
8. **public** **class** MyPanel  **extends** JPanel{
9. **static** String path=**null**;
10. @Override
11. **protected** **void** paintComponent(Graphics g) {
13. ImageIcon icon = **new** ImageIcon(path);
14. Image img = icon.getImage();
15. g.drawImage(img, 0, 0, icon.getIconWidth(), icon.getIconHeight(), icon.getImageObserver());
16. Configuration.setPATH(path);
17. }
18. }

该类仅用于绘制背景图片。

15.Configuration.txt

1. src\rock\_paper\_scissors\背景0.jpg

**三．算法分析**

由于该程序主要是实现界面之间事件的触发响应，真正需要算法处理的部分仅为输赢逻辑的处理，这部分用简单的if和switch语句就可以轻松实现。

以如下程序段为例：

1. **if** (AI.winNUM == num / 2 + 1) {
2. **int** res = JOptionPane.showConfirmDialog(**null**, "恭喜您赢了！！！\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
3. **if** (res == JOptionPane.YES\_OPTION) {
4. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
5. AI.winNUM = AI.drawNUM = AI.lostNUM = 0;
6. } **else** {
7. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
8. //return;
9. AllFrames.mainUI.getJf().setVisible(**true**);
10. closeAction();
11. jf.dispose();
12. }
13. } **else** **if** (AI.lostNUM == num / 2 + 1 ) {
14. **int** res = JOptionPane.showConfirmDialog(**null**, "遗憾您输了。。。\n是否继续操作", "是否继续", JOptionPane.YES\_NO\_OPTION);
15. **if** (res == JOptionPane.YES\_OPTION) {
16. //System.out.println("选择是后执行的代码"); // 点击“是”后执行这个代码块
17. AI.winNUM = AI.drawNUM = AI.lostNUM = 0;
18. } **else** {
19. //System.out.println("选择否后执行的代码"); // 点击“否”后执行这个代码块
20. //return;
21. AllFrames.mainUI.getJf().setVisible(**true**);
22. closeAction();
23. jf.dispose();
24. }
25. }

在外层if else语句中，通过玩家获胜次数与总局数的比较来判断是哪一方获胜；在外层if else语句中嵌套了一个结果弹框，点击“是”则结果清零从头开始，点击“否”则关闭游戏界面回到主界面。该程序段能够简单实现输赢结果的判断。由于if else语句内无循环语句，其时间复杂度由其内部调用的函数复杂度决定，为O(1);

**四．软件调试和测试**

1.倒计时多次触发

通过添加System.out.pringtln("?"+i)和属性i（Clock.java）使得每次新建该类的实例都会在控制台打印一个数字以表明这是第几个实例。经运行程序发现，当两名玩家都完成了输入后，仍然会触发生成Clock的实例。检查得，两名玩家都输入时，leftLock和rightLock同时为true,然后根据是否为true生成Clock，然后执行operate恢复false（HumanUI.java）.通过在生成Clock实例前检查clock属性是否为null,同时在结束倒计时使得属性clock = null解决了这个问题（HumanUI.java）。

2.可多次生成游戏界面

通过运行程序发现，通过选择界面选择完毕游戏类型之后，选择界面仍然会留在屏幕上。这可能会导致有用户多次点击选择界面从而产生多个游戏界面。这与设计初衷不符。比较了“生成界面后禁用原界面的功能”和“生成界面后原界面不可视”后，选择了较美观的后一种。即在生成新界面完成后使界面调用setVisible(false);（Choose.java）

3.胜负判定方式

一开始胜负判断是必须完成num次对局才会判断出输赢结果，其中还会因为存在平局导致三局两胜和五局三胜存在平局的情况，后来经过将判断条件改成一方胜利次数达到num/2+1且另一方胜利次数少于等于num/2次来判断输赢，既解决了机械性判断，也减少了不必要的冗余回合，实现智能提前判断输赢，避免不必要的浪费。

**五．课题项目工程实践总结**

**软件开发工具的选择和使用方案**

在拿到课题后了解课题需求后，预计此课题的实现只需一些基本的Java库便可完成，所以随即决定采用NetBeans和IntelliJ idea这些常用开发工具。

团队成员甲用的NetBeans编辑代码，体验感十分不好：

1、代码会出现无法对齐的情况；

2、智能代码补充功能不是很完善，降低编码效率；

3、NetBeans的界面很不友善，关闭Projects任务栏之后不易还原等；

相比之下，团队成员乙使用的idea开发工具就相对智能友善，界面美观、智能补码，提高编码效率。idea的代码纠错功能也比NetBeans强大，其万能键的使用更加方便了代码的编写与调试。

**文档工具的选择和使用方案**

大作业报告的撰写文档工具我们采用的是word+WPS在线文档。

word在Windows本机是很强大的办公软件，但是对于报告团队协作撰写的本地化就显得不高效，团队成员无法实时查看队友的编辑情况，造成资源浪费；

WPS在线协作可以利于团队成员交流对报告内容见解，在撰写报告中分工合作，在线文档可以帮助成员交流想法，方便修改润饰，提高了报告内容的一致性，最后以word的形式导出报告。

**项目团队协作的管理方法**

团队协作管理我们采用人工进行，由于成员少，任务量不是很大，人工进行效率也不低。

团队在进行程序设计时，采用了面向对象编程的编程思想（OOP），即将整个石头剪刀布程序视作一组对象的集合而不是一些功能的顺序实现。这更加符合人对世界的认知，简化了设计负担。

团队在进行程序设计时，采用了单例设计模式，即每个类只创建一个实例，并提供直接访问这些实例的途径。

团队在程序设计时的目标，尽量朝着“低耦合高内聚”的方向进行编写。每个程序块只负责一个功能，让模块与模块之间的联系，尽量少而简单。

**软件架构理论的体现**

1、软件粒度最小的单元通常是“类”；在本课题中，我们采用了不下于十种类，其中最为重要的有rock\_paper\_scissors、ChooseUI、GameUI、HumanUI、ClockUI等。

2、几个类紧密协作形成“模块”；每个类都与其他类紧密相连，谁也无法离开谁，彼此相互作用。比如rock\_paper\_scissors中调用了类ChooseUI，ChooseUI又与GameUI和HumanUI等类紧密相连，HumanUI类中又联系着ClockUI类等；这么多类都无法脱离整体而单独存在，紧密协作。

3、完成相对独立的功能的多个模块构成了“子系统”。在此程序中，人机游戏与人人游戏是两个相对独立的功能，其实现与对方没有必然联系。

4、多个子系统相互配合满足了一个完整应用的需求，从而构成了软件“系统”；正如主界面上三个模块构成了这个完整的剪刀石头布游戏程序。

**问题及解决方法：**

1.我们发现背景图片难以显示。借助搜索引擎得，可能是图片过于大的原因。最后通过压缩图片至10kb左右得以解决。

2.出现了预料之外的错误。在进行程序编写时，出现了无法给按钮添加位置的情况。经调试无法找到错误现的位置，最后通过重新开始编写程序解决。

3.软件调试过程中遇到的问题见“四.软件调试与测试”。

**心得体会(吴 雯)：**

通过此次的课题程序设计，我们对这学期所学的Java语言程序设计有一个更深刻的了解，将所学知识应用于实践。为了使游戏能够实现预先的构想，我们也查阅了网上相关程序，比较好的流程图及功能模块，然后再着手实现程序并不断修改以达到预先的构想，完成之后感觉对Java程序又有了新的认识。

当完成程序设计之后开始补充完善文档的时候，一方面觉得自己取得了一定的成绩，从想法到设计到完成，确实学到了很多，虽然现在Java学习还不是很深入，但相信通过自己的努力会做得更好。

在Java设计之前也做过其他语言的课程设计，但模块之间联系没有这么紧因为各个模块之间的联系十分紧密，代码重复执行率高，经过调试、修改后，终于看到成品。同时也发现自己的不足，我们这个程序的体系并不标准；文件层次结构有待完善。通过本次的课题设计，我平时学到的理论知识得到了很好的应用，理论与实际相结合，受益匪浅。

**心得体会(俞博宸)：**

通过这次课题程序设计，较好的将所学知识贯穿联合起来，同时也实现了一个较为完整的程序，是一次宝贵的经验。

在这个过程中，我们了解到了团队合作的必要性和对团队成员的提高功能，在编写程序的过程中，互相学习彼此的代码，发现了自己的错漏之处并予以补强。同时在讨论过程中对要做怎样的功能逐渐完善，也跟让我们明白了讨论的重要性。

在完成整个程序的进程里，我们采用了“先完成主体功能，再逐渐细化其他要求”的方式。即先完成了石头剪刀布的胜负判定和显示，再完成了倒计时，弹窗判胜负，以及选择几局几胜的功能。这种思想会继续影响我们之后的编程生涯。

我们也认识到了自己的不足。在程序编写过程中并没有将不同的类放到不同的包中，同时部分代码也存在着冗余。在之后的编程过程中，应当吸取教训，尽量避免这类问题。

纸上得来终觉浅，绝知此事要躬行。通过本次的课题设计，我们愈发感受到了实践在学习过程中起到的重要作用。

**六．参考文献和材料**

参考书籍：《Java语言程序设计》

网络参考:[1]<https://blog.csdn.net/sys1159378858/article/details/82621564>

[2]<https://blog.csdn.net/xietansheng/article/details/72814492>

开源项目：<https://github.com/pengxiandyou/RockPaperScissors-BC-201908>

**七．团队成员姓名和联系方法**

俞博宸 邮箱1239048739@qq.com 电话：19805189210

吴 雯 邮箱2191226423@qq.com 电话：19805182358

**八．附录**

具体操作可在游戏主界面关于中查看。

以下为游戏的简单演示视频：

