

Design and Development of Rock-Paper-Scissors Game using Java

Team No.10

6830300088 เจตนิพัทธ์ พรหมศรี

6830300631 ภัทราภรณ์ นาคสิทธิ์

6830300908 เสฎฐพงษ์ โพธิ์ศรี

Submit to

Asst.Prof.Dr. Kullawadee Somboonviwat

03603112 Programming Fundamentals

Kasetsart University, Sriracha Campus

สารบัญ

Title Page	ก
Table of Contents	ข
Rock-Paper-Scissors Program	1
Rules of the Rock-Paper-Scissors Game	1
Sample Runs	2
Program Design	3
Flowchart	3
Program Structure	4
Implementation	5
Source Code in GitHub	5
How to Compile and Run the Program	6
Appendix	9

Rock-Paper-Scissors Program

Rules of the rock-paper-scissors game

ในเกมเป่าชิงจูบนี้ ผู้เล่นและคอมพิวเตอร์จะเลือกระหว่าง rock, paper หรือ scissors ตามกติกามาตรฐานคือ rock ชนะ scissors, scissors ชนะ paper และ paper ชนะ rock หากเลือกเหมือนกันจะถือว่าเสมอ โดยระบบจะให้คะแนนผู้เล่นเมื่อชนะ 100 คะแนน เสมอ 50 คะแนน และแพ้ 0 คะแนน พร้อมทั้งมีระบบสกิลสุ่มในแต่ละรอบด้วยโอกาส 30% ซึ่งสกิลอาจตกเป็นของผู้เล่นหรือคอมพิวเตอร์ โดยสกิล Shield จะทำให้ผลการแพ้กลายเป็นเสมอ และสกิล Double จะทำให้คะแนนชนะของผู้เล่นถูกคูณสอง ทั้งหมดนี้จะถูกนำมาสะสมเป็นคะแนนรวมเพื่อปลดล็อกเหรียญ Bronze ที่ 1000 คะแนน Silver ที่ 5000 คะแนน และ Gold ที่ 10000 คะแนน จนกว่าผู้เล่นจะออกจากเกมด้วยคำสั่ง !exit

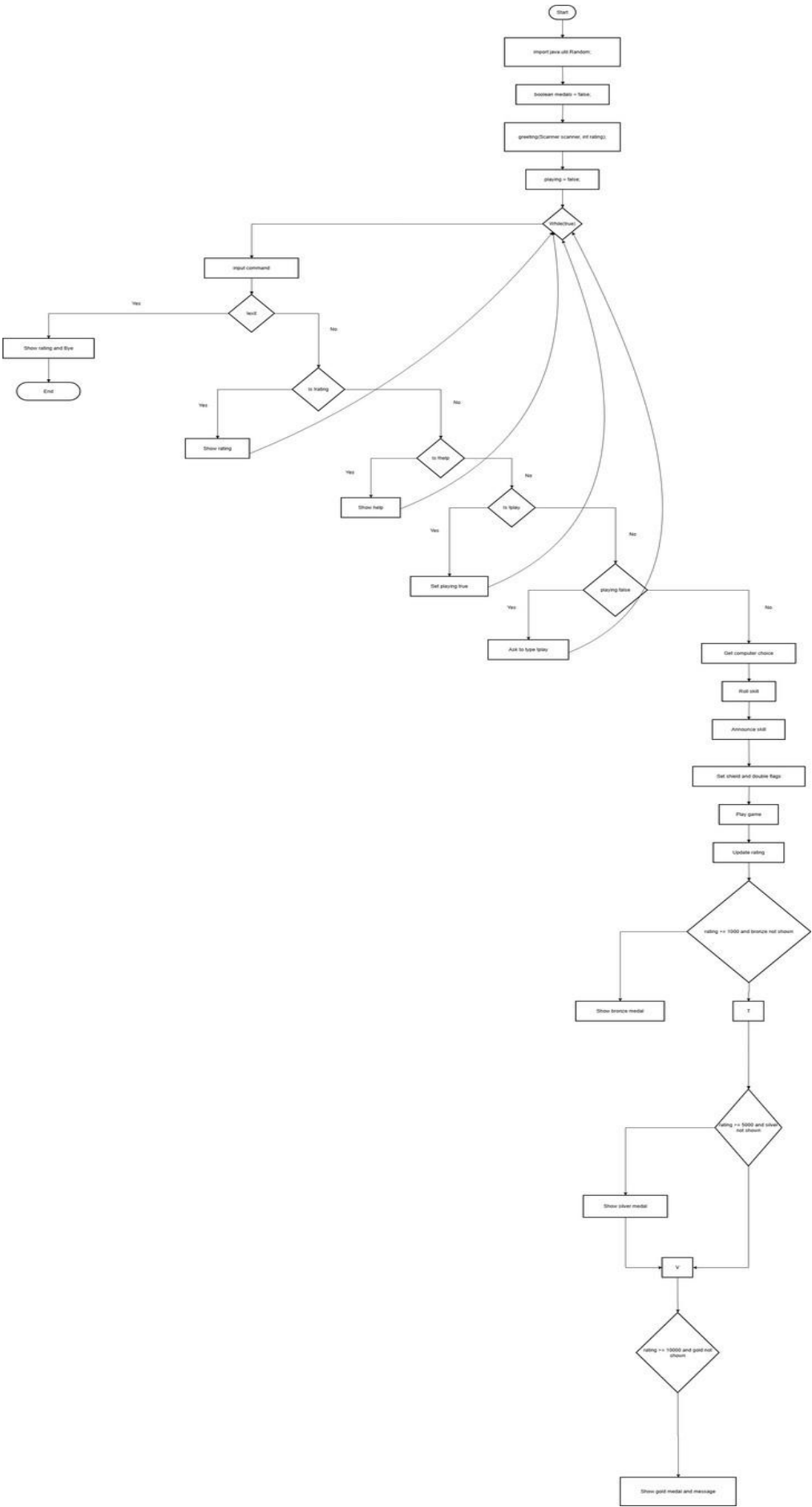
Sample runs

ตัวอย่างผลลัพธ์ที่ได้จากการรัน โปรแกรมแสดงให้เห็นการเลือกของผู้เล่นและคอมพิวเตอร์ รวมถึงผลลัพธ์ของเกม

```
PS C:\Users\jetniphat\Documents\F2-256802-T10> javac .\Rock_Paper_Scissors.java
PS C:\Users\jetniphat\Documents\F2-256802-T10> java Rock_Paper_Scissors
Enter your name: Jeff
Hello, Jeff!
Current Score: 0
Let's play Rock-Paper-Scissors!
Commands:
!play = start playing
!rating = show your current score
!help = show commands
!exit = exit the game
> !play
Game started! Type: rock / paper / scissors
> rock
There is a draw (rock)
> rock
There is a draw (rock)
> rock
Well done. The computer chose scissors and failed
> rock
[Skill] Computer got: Attack x2 (Double)
Sorry, but the computer chose paper
> !help
Commands:
!play = start playing
!rating = show your current score
!help = show commands
!exit = exit the game
Moves: rock, paper, scissors
> !rating
Your rating: 200
> !exit
Your rating: 200
Bye!
```

Program Design

Flowchart



Program Design

Program Structure

```

1  import java.util.Random;
2  import java.util.Scanner;
3
4  public class Rock_Paper_Scissors {
5
6      public static String greeting(Scanner scanner, int rating) {
7          String player;
8
9          System.out.print(s: "Enter your name: ");
10         player = scanner.nextLine();
11         System.out.println("Hello, " + player + "!");
12         System.out.println("Current Score: " + rating);
13         System.out.println(x: "Let's play Rock-Paper-Scissors!");
14         System.out.println(x: "Commands:");
15         System.out.println(x: "lplay = start playing");
16         System.out.println(x: "lrating = show your current score");
17         System.out.println(x: "lhelp = show commands");
18         System.out.println(x: "lexit = exit the game");
19
20         return player;
21     }
22
23     public static int userWin(String computer) {
24         System.out.println("Well done. The computer chose " + computer + " and failed");
25         return 100;
26     }
27
28     public static int userLose(String computer) {
29         System.out.println("Sorry, but the computer chose " + computer);
30         return 0;
31     }
32
33     public static int userDraw(String computer) {
34         System.out.println("There is a draw (" + computer + ")");
35         return 50;
36     }

```

```

37
38     public static int[] rollSkill(Random random) {
39         int skillOwner = 0;
40         int skillType = 0;
41
42         int chance = random.nextInt(bound: 100);
43         if (chance < 30) {
44             skillOwner = random.nextBoolean() ? 1 : 2;
45             skillType = random.nextBoolean() ? 1 : 2;
46         }
47
48         return new int[]{skillOwner, skillType};
49     }
50
51     public static void announceSkill(String playerName, int[] skill) {
52         int owner = skill[0];
53         int type = skill[1];
54
55         if (owner == 0)
56             return;
57
58         String who = (owner == 1) ? playerName : "Computer";
59         if (type == 1) {
60             System.out.println("[Skill] " + who + " got: AI attack has no effect (Shield)");
61         } else if (type == 2) {
62             System.out.println("[Skill] " + who + " got: Attack x2 (Double)");
63         }
64     }
65
66     public static int playGame(String playerChoice, String computer,
67                                boolean usershield, boolean aishield, boolean userDouble) {
68
69         int rating = 0;

```

```

4 public class Rock_Paper_Scissors {
66     public static int playGame(String playerChoice, String computer,
70
71         if (playerChoice.equals(computer)) {
72             return userDraw(computer);
73         }
74
75         boolean userWins = false;
76         boolean userLoses = false;
77
78         if (playerChoice.equals(anObject: "rock")) {
79             if (computer.equals(anObject: "scissors")) userWins = true;
80             else if (computer.equals(anObject: "paper")) userLoses = true;
81         } else if (playerChoice.equals(anObject: "paper")) {
82             if (computer.equals(anObject: "rock")) userWins = true;
83             else if (computer.equals(anObject: "scissors")) userLoses = true;
84         } else if (playerChoice.equals(anObject: "scissors")) {
85             if (computer.equals(anObject: "paper")) userWins = true;
86             else if (computer.equals(anObject: "rock")) userLoses = true;
87         } else {
88             System.out.println(x: "Invalid input!");
89             return 0;
90         }
91
92         if (userWins) {
93             rating = userWin(computer);
94             if (userDouble) {
95                 rating *= 2;
96                 System.out.println(x: "[Skill Effect] Your attack is x2! Bonus doubled.");
97             }
98         } else if (userLoses) {
99             if (userShield) {
100                 rating = userDraw(computer);
101                 System.out.println(x: "[Skill Effect] Shield activated! AI attack has no effect.");
102             } else if (aiShield) {
103                 rating = userDraw(computer);

```

```

4 public class Rock_Paper_Scissors {
66     public static int playGame(String playerChoice, String computer,
104         System.out.println(x: "[Skill Effect] Computer's Shield activated! Your attack has no effect.");
105     } else {
106         rating = userLose(computer);
107     }
108 }
109
110 return rating;
111 }
112
113 public static String getComputerChoice(Random random) {
114     String[] computerChoices = {"rock", "paper", "scissors"};
115     return computerChoices[random.nextInt(computerChoices.length)];
116 }
117
118 public static void printHelp() {
119     System.out.println(x: "Commands:");
120     System.out.println(x: "!play = start playing");
121     System.out.println(x: "!rating = show your current score");
122     System.out.println(x: "!help = show commands");
123     System.out.println(x: "!exit = exit the game");
124     System.out.println(x: "Moves: rock, paper, scissors");
125 }
126
127 Run | Debug
128 public static void main(String[] args) {
129     int rating = 0;
130     Scanner scanner = new Scanner(System.in);
131     Random random = new Random();
132
133     boolean bronzeShown = false;
134     boolean silverShown = false;
135     boolean goldShown = false;
136

```

```

4   public class Rock_Paper_Scissors {
127  public static void main(String[] args) {
137      String playerName = greeting(scanner, rating);
138      boolean playing = false;
139
140      while (true) {
141          System.out.print(s: "> ");
142          String choice = scanner.nextLine().toLowerCase();
143
144          if (choice.equals(anObject: "lexit")) {
145              System.out.println("Your rating: " + rating);
146              System.out.println(x: "Bye!");
147              break;
148          }
149
150          if (choice.equals(anObject: "lrating")) {
151              System.out.println("Your rating: " + rating);
152              continue;
153          }
154
155          if (choice.equals(anObject: "lhelp")) {
156              printHelp();
157              continue;
158          }
159
160          if (choice.equals(anObject: "lplay")) {
161              playing = true;
162              System.out.println(x: "Game started! Type: rock / paper / scissors");
163              continue;
164          }
165
166          if (!playing) {
167              System.out.println(x: "Type lplay to start playing (or lhelp for commands).");
168              continue;
169          }
170

```

```

4   public class Rock_Paper_Scissors {
127  public static void main(String[] args) {
170
171      String computer = getComputerChoice(random);
172
173      int[] skill = rollSkill(random);
174      announcesSkill(playerName, skill);
175
176      boolean usershield = (skill[0] == 1 && skill[1] == 1);
177      boolean aishield = (skill[0] == 2 && skill[1] == 1);
178      boolean userDouble = (skill[0] == 1 && skill[1] == 2);
179
180      rating += playGame(choice, computer, usershield, aishield, userDouble);
181
182      if (rating >= 1000 && !bronzeShown) {
183          System.out.println(x: "-----+");
184          System.out.println(x: "|          BRONZE MEDAL! |");
185          System.out.println(x: "| You reached 1000 points |");
186          System.out.println(x: "-----+");
187          bronzeShown = true;
188      }
189
190      if (rating >= 5000 && !silverShown) {
191          System.out.println(x: "-----+");
192          System.out.println(x: "|          SILVER MEDAL! |");
193          System.out.println(x: "| You reached 5000 points |");
194          System.out.println(x: "-----+");
195          silverShown = true;
196      }
197

```

```

4   public class Rock_Paper_Scissors {
177  public static void main(String[] args) {
178      boolean userDouble = (skill[0] == 1 && skill[1] == 2);
179
180      rating += playGame(choice, computer, userShield, aiShield, userDouble);
181
182      if (rating >= 1000 && !bronzeShown) {
183          System.out.println(x: "-----+");
184          System.out.println(x: "|          BRONZE MEDAL!      |");
185          System.out.println(x: "| You reached 1000 points  |");
186          System.out.println(x: "-----+");
187          bronzeShown = true;
188      }
189
190      if (rating >= 5000 && !silverShown) {
191          System.out.println(x: "-----+");
192          System.out.println(x: "|          SILVER MEDAL!      |");
193          System.out.println(x: "| You reached 5000 points  |");
194          System.out.println(x: "-----+");
195          silverShown = true;
196      }
197
198      if (rating >= 10000 && !goldShown) {
199          System.out.println(x: "-----+");
200          System.out.println(x: "|          GOLD MEDAL!      |");
201          System.out.println(x: "| You reached 10000 points  |");
202          System.out.println(x: "|");
203          System.out.println(x: "| Ai: You're really boring, you know. |");
204          System.out.println(x: "| But it's not like I'm playing with you because |");
205          System.out.println(x: "| I like you or anything! |");
206          System.out.println(x: "-----+");
207          goldShown = true;
208      }
209  }
210  }
211  }
212

```

- โปรแกรมเริ่มที่ main() สร้าง rating = 0, Scanner, Random และตัวแปรเช็คเหรียญ (bronze/silver/gold)
- เรียก greeting(scanner, rating) รับชื่อผู้เล่นจาก Scanner → แสดงข้อความต้อนรับและคำสั่ง → ส่งชื่อกลับมาเก็บใน playerName
- ตั้งค่า playing = false แล้วเข้า while(true) (ลูปหลักของโปรแกรม)
- ทุกครั้งในลูป รับอินพุตผู้ใช้ด้วย scanner.nextLine() เก็บใน choice
- ถ้า choice == "!exit" แสดงคะแนนสุดท้าย → พิมพ์ Bye! → ออกจากลูป → โปรแกรมจบ
- ถ้า choice == "!rating" แสดงคะแนนปัจจุบัน → วนลูปรอบใหม่
- ถ้า choice == "!help" เรียก printHelp() → แสดงคำสั่ง → วนลูปรอบใหม่
- ถ้า choice == "!play" ตั้ง playing = true → แข่งเริ่มเกม → วนลูปรอบใหม่
- ถ้า playing == false และพิมพ์อย่างอื่น แจ้งให้พิมพ์ !play ก่อน → วนลูปใหม่
- เมื่อเริ่มเล่นแล้ว เรียก getComputerChoice(random) เพื่อสุ่มท่าของคอมพิวเตอร์
- เรียก rollSkill(random) สุ่มโอกาส 30% ว่ามีสกิลหรือไม่ → คืนค่า [owner, type]
- เรียก announceSkill(playerName, skill) ถ้ามีสกิล → แสดงว่าใครได้และเป็นสกิลอะไร
- แปลงผลสกิลเป็น Boolean กำหนด userShield, aiShield, userDouble
- เรียก playGame(choice, computer, userShield, aiShield, userDouble) ใช้กติกา Rock-Paper-Scissors คัดสินแพ้/ชนะ/เสมอ พร้อมผลสกิล
- ภายใน playGame()
 - เสมอ → เรียก userDraw() ได้ 50 คะแนน
 - ผู้เล่นชนะ → เรียก userWin() ได้ 100 คะแนน (x2 ถ้ามี Double)
 - ผู้เล่นแพ้ → เช็ค Shield → เสมอ หรือ userLose() ได้ 0 คะแนน
 - คืนค่าคะแนนรอบนั้นกลับไป
- main() นำคะแนนที่ได้มาบวกกับ rating
- เช็คคะแนนรวม
 - ≥ 1000 → แสดง Bronze (ครั้งเดียว)
 - ≥ 5000 → แสดง Silver (ครั้งเดียว)
 - ≥ 10000 → แสดง Gold (ครั้งเดียว)
- วนกลับไปข้อ 4 รออินพุตใหม่ จนกว่าจะพิมพ์ !exit

Implementation

Source code in GitHub

ซอร์สโค้ดของโปรแกรมถูกจัดเก็บไว้บน GitHub เพื่อความสะดวกในการจัดการและตรวจสอบ

— ชื่อ Repository ของกลุ่ม: F2-256802-T10

— โฟลเดอร์: AS01-Rock-Paper-Scissors

How to compile and run the program

— เปิด Command Prompt หรือ Terminal

— ไปยังโฟลเดอร์ที่เก็บไฟล์ Rock_Paper_Scissors.java

— ใช้คำสั่ง `javac Rock_Paper_Scissors.java` เพื่อคอมไพล์โปรแกรม

— ใช้คำสั่ง `java Rock_Paper_Scissors` เพื่อรันโปรแกรม