

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

Team No.10

6830300088 เจตนิพัทธ์ พรมศร

6830300631 กัตตารากรณ์ นาคลิทธี

6830300908 เสฎฐพงศ์ โพธิ์ครี

Submit to

Asst.Prof.Dr. Kullawadee Somboonviwat

03603112 Programming Fundamentals

Kasetsart University, Sriracha Campus

តារប័ណ្ណ

Title Page	၁
Table of Contents	၂
Rock-Paper-Scissors Program	၁
Rules of the Rock-Paper-Scissors Game	၁
Sample Runs	၂
Program Design	၃
Flowchart	၃
Program Structure	၄
Implementation	၅
Source Code in GitHub	၅
How to Compile and Run the Program	၆
Appendix	၉

Rock-Paper-Scissors Program

Rules of the rock-paper-scissors game

ในเกมเป้าปั้งชูบันนี่ ผู้เล่นและคอมพิวเตอร์จะเลือกรหัสว่าง rock, paper หรือ scissors ตามกติกามาตรฐานคือ rock ชนะ scissors, scissors ชนะ paper และ paper ชนะ rock หากเลือกเหมือนกันจะอีกว่าเสมอ โดยระบบจะให้คะแนนผู้เล่นเมื่อชนะ 100 คะแนน เสมอ 50 คะแนน และแพ้ 0 คะแนน พร้อมทั้งมีระบบสกิลสุ่มในแต่ละรอบด้วยโอกาส 30% ซึ่งสกิลอาจตกเป็นของผู้เล่นหรือคอมพิวเตอร์ โดยสกิล Shield จะทำให้ผลการแพ้กลับเป็นเสมอ และสกิล Double จะทำให้คะแนนชนะของผู้เล่นสูงถึงสองเท่า ทั้งหมดนี้จะถูกนำมาสะสมเป็นคะแนนรวมเพื่อปลดล็อกเหรียญ Bronze ที่ 1000 คะแนน Silver ที่ 5000 คะแนน และ Gold ที่ 10000 คะแนน จนกว่าผู้เล่นจะออกจากเกมด้วยคำสั่ง !exit

Sample runs

ตัวอย่างผลลัพธ์ที่ได้จากการรันโปรแกรมแสดงให้เห็นการเลือกของผู้เล่นและคอมพิวเตอร์ รวมถึงผลลัพธ์ของเกม

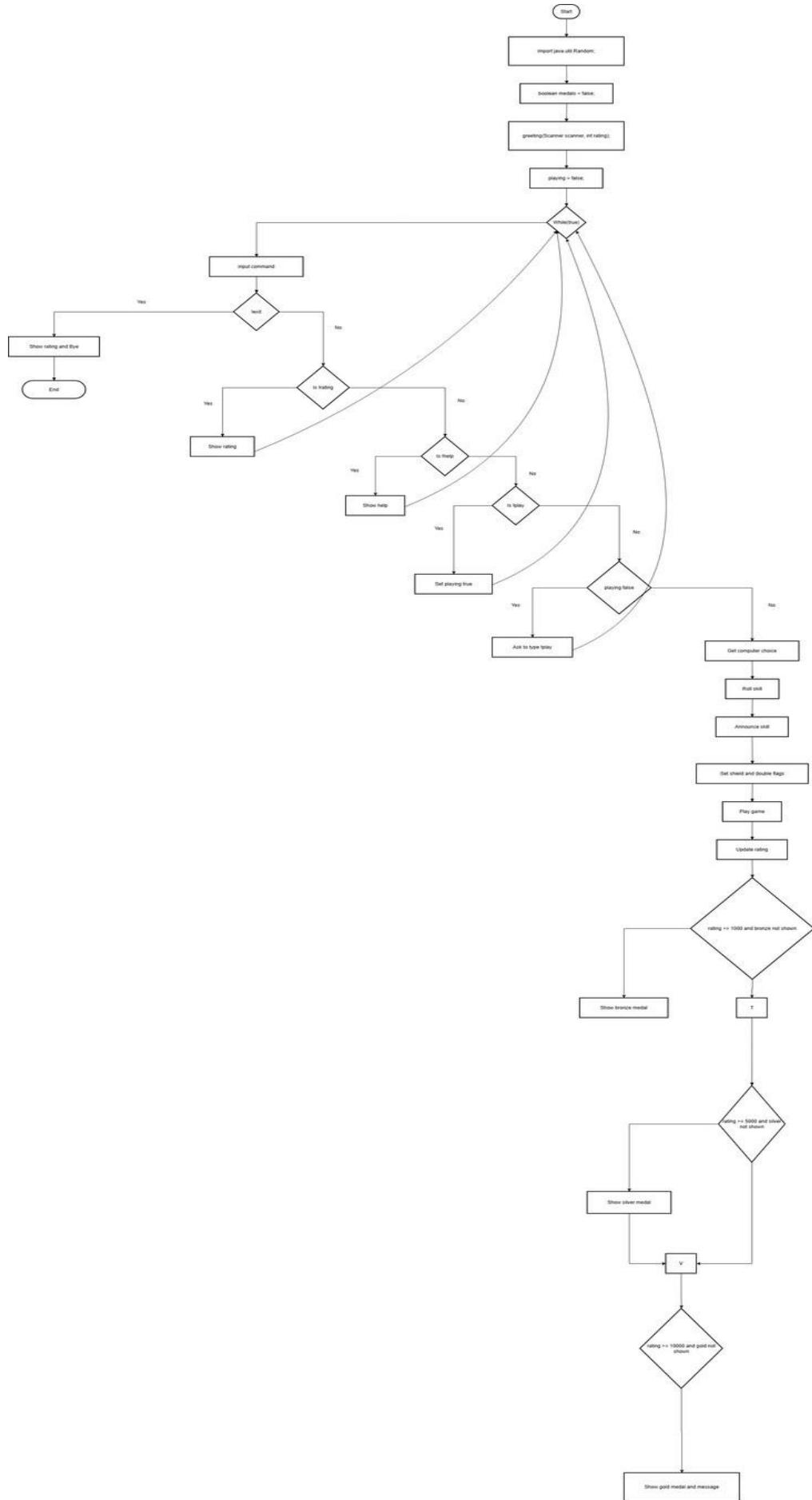
```

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

```

F2-256802-T10 2 J Rock_Paper_Scissors.java > % Rock_Paper_Scissors > greeting(Scanner, int)
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("Enter your name: ");
10        player = scanner.nextLine();
11        System.out.println("Hello, " + player + "!");
12        System.out.println("Current Score: " + rating);
13        System.out.println("Let's play Rock-Paper-Scissors!");
14        System.out.println("Commands:");
15        System.out.println("play = start playing");
16        System.out.println("rating = show your current score");
17        System.out.println("help = show commands");
18        System.out.println("exit = 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,
67        ...
68        if (playerChoice.equals(computer)) {
69            return userDraw(computer);
70        }
71
72        boolean userWins = false;
73        boolean userLoses = false;
74
75        if (playerChoice.equals(anObject: "rock")) {
76            if (computer.equals(anObject: "scissors")) userWins = true;
77            else if (computer.equals(anObject: "paper")) userLoses = true;
78        } else if (playerChoice.equals(anObject: "paper")) {
79            if (computer.equals(anObject: "rock")) userWins = true;
80            else if (computer.equals(anObject: "scissors")) userLoses = true;
81        } else if (playerChoice.equals(anObject: "scissors")) {
82            if (computer.equals(anObject: "paper")) userWins = true;
83            else if (computer.equals(anObject: "rock")) 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,
67        ...
68        System.out.println(x: "[Skill Effect] Computer's shield activated! Your attack has no effect.");
69    } else {
70        rating = userLose(computer);
71    }
72
73    return rating;
74 }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
Run|Debug
public static void main(String[] args) {
    ...
    int rating = 0;
    Scanner scanner = new Scanner(System.in);
    Random random = new Random();
    ...
    boolean bronzeShown = false;
    boolean silverShown = false;
    boolean goldShown = false;
}

```

```

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: "exit")) {
145          System.out.println("Your rating: " + rating);
146          System.out.println(x: "Bye!");
147          break;
148        }
149
150        if (choice.equals(anObject: "rating")) {
151          System.out.println("Your rating: " + rating);
152          continue;
153        }
154
155        if (choice.equals(anObject: "help")) {
156          printHelp();
157          continue;
158        }
159
160        if (choice.equals(anObject: "play")) {
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 !play to start playing (or !help for commands).");
168          continue;
169        }
170      }
}

```

```

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

```

```

4  public class Rock_Paper_Scissors {
127    public static void main(String[] args) {
128      boolean userDouble = (skill[0] == 1 && skill[1] == 2);
129
130      rating += playGame(choice, computer, userShield, aiShield, userDouble);
131
132      if (rating >= 1000 && !bronzeShown) {
133        System.out.println(x: "+-----+");
134        System.out.println(x: "| BRONZE MEDAL! |");
135        System.out.println(x: "| You reached 1000 points |");
136        System.out.println(x: "+-----+");
137        bronzeShown = true;
138      }
139
140      if (rating >= 5000 && !silverShown) {
141        System.out.println(x: "+-----+");
142        System.out.println(x: "| SILVER MEDAL! |");
143        System.out.println(x: "| You reached 5000 points |");
144        System.out.println(x: "+-----+");
145        silverShown = true;
146      }
147
148      if (rating >= 10000 && !goldShown) {
149        System.out.println(x: "+-----+");
150        System.out.println(x: "| GOLD MEDAL! |");
151        System.out.println(x: "| You reached 10000 points |");
152        System.out.println(x: "| Ai: You're really boring, you know. |");
153        System.out.println(x: "| But it's not like I'm playing with you because |");
154        System.out.println(x: "| I like you or anything! |");
155        System.out.println(x: "+-----+");
156        goldShown = true;
157      }
158    }
159  }
160}

```

1. โปรแกรมเริ่มที่ main() สร้าง rating = 0, Scanner, Random และตัวแปรเชื้อค่าเหรียญ (bronze/silver/gold)
2. เรียก greeting(scanner, rating) รับชื่อผู้เล่นจาก Scanner → แสดงข้อความต้อนรับและคำสั่ง → ส่งชื่อกลับมาเก็บใน playerName
3. ตั้งค่า playing = false และเข้า while(true) (ลูปหลักของโปรแกรม)
4. ทุกครั้งในลูป รับอินพุตผู้ใช้ด้วย scanner.nextLine() เก็บใน choice
5. ถ้า choice == "!exit" แสดงคะแนนสุดท้าย → พิมพ์ Bye! → ออกจากรูป → โปรแกรมจบ
6. ถ้า choice == "!rating" แสดงคะแนนปัจจุบัน → วนลูปรอบใหม่
7. ถ้า choice == "!help" เรียก printHelp() → แสดงคำสั่ง → วนลูปรอบใหม่
8. ถ้า choice == "!play" ตั้ง playing = true → แจ้งวิ่งเกม → วนลูปรอบใหม่
9. ถ้า playing == false และพิมพ์อย่างอื่น แจ้งให้พิมพ์ !play ก่อน → วนลูปใหม่
10. เมื่อเริ่มเล่นแล้ว เรียก getComputerChoice(random) เพื่อสุ่มท่าของคอมพิวเตอร์
11. เรียก rollSkill(random) ลุ้นโอกาส 30% ว่ามีสกิลหรือไม่ → คืนค่า [owner, type]
12. เรียก announceSkill(playerName, skill) ถ้ามีสกิล → แสดงว่าใครได้และเป็นสกิลอะไร
13. แปลงผลสกิลเป็น Boolean กำหนด userShield, aiShield, userDouble
14. เรียก playGame(choice, computer, userShield, aiShield, userDouble) ใช้คติกา Rock-Paper-Scissors ตัดสินแพ้/ชนะ/เสมอ พร้อมผลสกิล
15. 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
16. main() นำคะแนนที่ได้มา加กับ rating
17. เช็คคะแนนรวม
 - ≥ 1000 → แสดง Bronze (ครั้งเดียว)
 - ≥ 5000 → แสดง Silver (ครั้งเดียว)
 - ≥ 10000 → แสดง Gold (ครั้งเดียว)
18. วนกลับไปข้อ 4 รออินพุตใหม่ จนกว่าจะพิมพ์ !exit

Implementation

Source code in GitHub

ชอร์สโค้ดของโปรแกรมถูกจัดเก็บไว้บน GitHub เพื่อความสะดวกในการจัดการและตรวจสอบ

— ชื่อ Repository ของกลุ่ม: F2-256802-T10

— ไฟล์เดอร์: AS01-Rock-Paper-Scissors

How to compile and run the program

— เปิด Command Prompt หรือ Terminal

— ไปยังไฟล์เดอร์ที่เก็บไฟล์ Rock_Paper_Scissors.java

— ใช้คำสั่ง javac Rock_Paper_Scissors.java เพื่อคอมpile โปรแกรม

— ใช้คำสั่ง java Rock_Paper_Scissors เพื่อรันโปรแกรม