



Rock Paper Scissor Game

เสนอ

ผศ.ดร. กุลลาวดี สมบูรณ์วิวัฒน์

โดย

- | | | |
|----------------|---------|------------|
| 1. นายณฐพงศ์ | กองธรรม | 6830300185 |
| 2. นายณพวรรณ | ทันสุข | 6830300193 |
| 3. นายศุภวิชญ์ | การิก | 6830300843 |

รายงานประกอบวิชา 03603112 – 65 หลักการโปรแกรมเบื้องต้น II
มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา ภาคการศึกษาที่ 2 ปีการศึกษา 2568

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา หลักการโปรแกรมเบื้องต้น II โดยมีจุดประสงค์เพื่อศึกษาหาความรู้ในเรื่อง เกมค้อน กรรไกร กระดาษ ซึ่งรายงานฉบับนี้มีเนื้อหาเกี่ยวกับการออกแบบโครงสร้าง การเขียนโปรแกรม ให้สามารถรันได้ในรูปแบบของเกม ค้อน กรรไกร กระดาษ ที่แปลกใหม่ของกลุ่มเรา

ผู้จัดทำคาดหวังเป็นอย่างยิ่งว่า การจัดทำเอกสารฉบับนี้จะมีข้อมูลที่เป็นประโยชน์ต่อผู้ที่สนใจศึกษาเรื่อง เกม ค้อน กรรไกร กระดาษ หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

17 ธันวาคม 2568

สารบัญ

คำนำ	ก
สารบัญ.....	๗
โครงสร้าง Rock Paper Scissor	
ส่วนที่ 1: การนำเข้าไลบรารี (Import)	1
ส่วนที่ 2: การสร้างคลาสและเมธอดหลัก	2
ส่วนที่ 3: การสร้างออบเจกต์และตัวแปร	2
ส่วนที่ 4: วนลูการเล่นเกม	2
ส่วนที่ 5: การรับค่าจากผู้เล่น.....	2
ส่วนที่ 6: การสุ่มของคอมพิวเตอร์	3
ส่วนที่ 7: การตัดสินใจแพ้-ชนะ	3
ส่วนที่ 8: การแสดงผลระหว่างเกม	3
ส่วนที่ 9: สรุปผลเมื่อจบเกม	4
สรุปแนวคิดของโปรแกรม	4
ผังงาน (Flowchart).....	5
ความหมายของผังงาน RockPaperScissor	5

Start (เริ่มต้นโปรแกรม).....	5
กำหนดค่าเริ่มต้น	6
แสดงเมนูให้ผู้เล่นเลือก	6
รับค่าจากผู้เล่น.....	6
ตรวจสอบเงื่อนไขออกจากเกม	6
คอมพิวเตอร์สุ่มตัวเลือก	7
เปรียบเทียบผลการเล่น	7
แสดงผลลัพธ์รอบนั้น	7
เล่นต่อหรือไม่	7
สรุปผล (End).....	7
สรุปความสำคัญของผังงาน	8

โครงสร้าง Rock Paper Scissor

โปรแกรมนี้เป็นเกม เป่ายิ้งฉุบ (Rock – Paper – Scissors)

ที่ให้ผู้เล่นแข่งกับคอมพิวเตอร์ผ่านทาง Terminal โดยระบบจะรับค่าที่ผู้เล่นเลือก,

ปุ่มตัวเลือกของคอมพิวเตอร์, เปรียบเทียบผลลัพธ์-ชนะ, นับสถิติ Win / Lose / Draw และคะแนน,

แสดงผลสรุปเมื่อผู้เล่นออกจากเกม

```

1  import java.util.Scanner;
2  import java.util.Random;
3
4  public class RockPaperScissor {
5      public static void main(String[] args) {
6
7          Scanner sc = new Scanner(System.in);
8          Random rand = new Random();
9
10         int score = 0;
11         int win = 0;
12         int lose = 0;
13         int draw = 0;
14
15         while (true) {
16
17             System.out.print("Enter rock/paper/scissor or !exit : ");
18             String player = sc.nextLine();
19
20             if (player.equals("!exit")) {
21                 break;
22             }
23
24             if (player.equals("secret")) {
25                 System.out.println("คุณใช้คำสั่งลับ! ชนะทันที (+100)");
26                 win++;
27                 score += 100;
28                 continue;
29             }
30
31             if (player.equals("bonus")) {
32                 System.out.println("ได้รับโบนัส +20 คะแนน!");
33                 score += 20;
34                 continue;
35             }
36
37             if (!player.equals("rock") &&
38                 !player.equals("paper") &&
39                 !player.equals("scissor")) {
40                 System.out.println("Invalid input.");
41                 continue;
42             }
43
44             int r = rand.nextInt(3);
45             String computer;
46             if (r == 0) {
47                 computer = "paper";
48             }
49             else if (r == 1) {
50                 computer = "scissor";
51             }
52             else {
53                 computer = "rock";
54             }
55
56             System.out.println("Computer chose: " + computer);
57
58             if (player.equals(computer)) {
59                 System.out.println("Result: Draw");
60                 draw++;
61             }
62             else if (player.equals("rock") && computer.equals("scissor")) {
63                 System.out.println("Result: You Win (+100)");
64                 win++;
65                 score += 100;
66             }
67             else if (player.equals("paper") && computer.equals("rock")) {
68                 System.out.println("Result: You Win (+100)");
69                 win++;
70                 score += 100;
71             }
72             else if (player.equals("scissor") && computer.equals("paper")) {
73                 System.out.println("Result: You Win (+100)");
74                 win++;
75                 score += 100;
76             }
77             else {
78                 System.out.println("Result: You Lose");
79                 lose++;
80             }
81
82             System.out.println();
83         }
84     }
85 }

```

ส่วนที่ 1: การนำเข้าไลบรารี (Import)

import java.util.Scanner; Scanner → ใช้รับค่าจากผู้ใช้จากคีย์บอร์ด

import java.util.Random; Random → ใช้คำสั่งสุ่มเลือก

ส่วนที่ 2: การสร้างคลาสและเมธอดหลัก

```
public class RockPaperScissor {    เป็นคลาสหลักของโปรแกรมโดยจะตั้งชื่อตามชื่อไฟล์

    public static void main(String[] args) { main() คือจุดเริ่มต้นการทำงานของโปรแกรม Java
```

ส่วนที่ 3: การสร้างออบเจกต์และตัวแปร

```
Scanner sc = new Scanner(System.in);
```

```
Random rand = new Random();
```

```
int score = 0;                score → คะแนนรวม
```

```
int win = 0;                  win → จำนวนครั้งที่ชนะ
```

```
int lose = 0;                 lose → จำนวนครั้งที่แพ้
```

```
int draw = 0;                 draw → จำนวนครั้งที่เสมอ
```

ส่วนที่ 4: วนลูปการเล่นเกม

```
while (true) {
```

ใช้ลูป **while(true)** เพื่อให้ผู้เล่นสามารถเล่นเกมได้ ไม่จำกัดรอบ โปรแกรมจะหยุดก็ต่อเมื่อผู้เล่นเลือกออกจากเกม

ส่วนที่ 5: การรับค่าจากผู้เล่น

ภายในลูป โปรแกรมจะให้ผู้เล่นเลือก:

Rock Paper Scissors !exit = ออกจากเกม

ใช้ Scanner รับค่าที่ผู้ใช้พิมพ์เข้ามา

ส่วนที่ 6: การสุ่มของคอมพิวเตอร์

```
int computer = rand.nextInt(3);
```

สุ่มเลข 3 ตัว (0 , 1 , 2)

แทนค่าการเลือกของคอมพิวเตอร์ (Rock / Paper / Scissors)

ส่วนที่ 7: การตัดสินใจแพ้-ชนะ

โปรแกรมใช้ if - else เปรียบเทียบค่า:

ผู้เล่นชนะ → เพิ่ม win และ score

ผู้เล่นแพ้ → เพิ่ม lose

เสมอ → เพิ่ม draw

เมื่อพิม secret --> score +100

เมื่อพิม bonus --> score +20

ตัวอย่างแนวคิด:

Rock ชนะ Scissors Paper ชนะ Rock Scissors ชนะ Paper

ส่วนที่ 8: การแสดงผลระหว่างเกม

หลังจากจบแต่ละรอบ โปรแกรมจะแสดงตัวเลือกให้ผู้เล่น

และตัวเลือกของคอมพิวเตอร์ จากนั้นนำมาเทียบกับ และแสดงผลลัพธ์ของรอบนั้น (Win / Lose / Draw)

```

87         }
88
89
90         System.out.println("\n--- Game Summary ---");
91         System.out.println("Win : " + win);
92         System.out.println("Lose : " + lose);
93         System.out.println("Draw : " + draw);
94         System.out.println("Score: " + score);
95     }
96 }

```

ส่วนที่ 9: สรุปผลเมื่อจบเกม

เมื่อผู้เล่นออกจากเกม โปรแกรมจะแสดงสรุป

```
System.out.println("\n--- Game Summary ---");
```

```
System.out.println("Win : " + win);
```

```
System.out.println("Lose : " + lose);
```

```
System.out.println("Draw : " + draw);
```

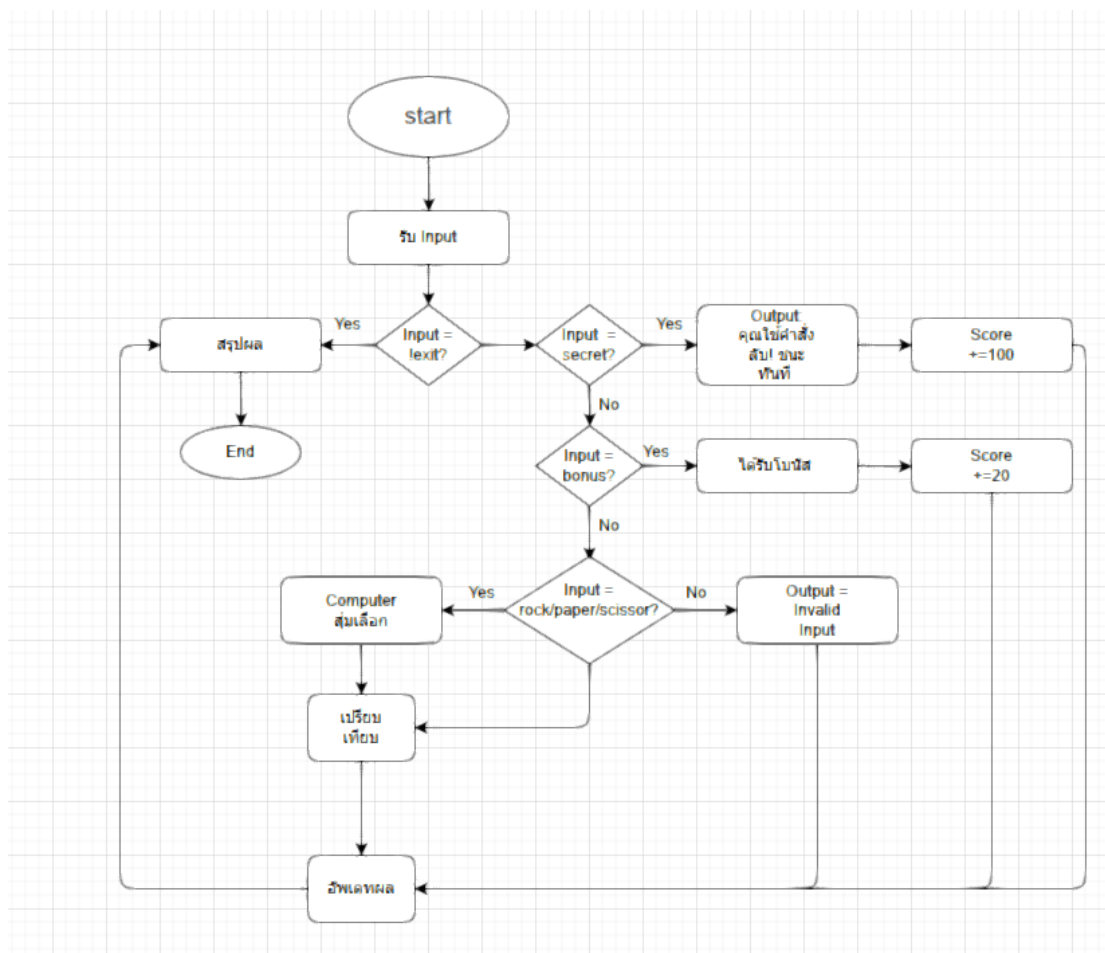
```
System.out.println("Score: " + score);
```

แสดง : จำนวนครั้งที่ ชนะ-แพ้ เสมอ คะแนนรวมทั้งหมด

สรุปแนวคิดของโปรแกรม

โปรแกรม RockPaperScissor เป็นเกมเป่ายิ้งฉุบที่พัฒนาด้วยภาษา Java โดยใช้หลักการรับข้อมูลจากผู้ใช้ การสุ่มค่าด้วยคลาส Random และการตัดสินใจด้วยโครงสร้างเงื่อนไข if-else ผู้เล่นสามารถเล่นได้หลายรอบและระบบจะบันทึกสถิติการเล่น พร้อมแสดงผลสรุปเมื่อจบเกม

ผังงาน (Flowchart)



ความหมายของผังงาน RockPaperScissor

ผังงานนี้แสดง ลำดับขั้นตอนการทำงานของโปรแกรม ตั้งแต่เริ่มต้นเกม การรับค่าจากผู้เล่น การสุ่มของคอมพิวเตอร์ การตัดสินใจ ไปจนถึงการสรุปคะแนน

Start (เริ่มต้นโปรแกรม)

- โปรแกรมเริ่มทำงาน

เตรียมระบบสำหรับเล่นเกม

กำหนดค่าเริ่มต้น

สร้างตัวแปร

- คะแนน (score) จำนวนครั้งที่ชนะ (win)
- แพ้ (lose) เสมอ (draw)
- สร้าง Scanner และ Random

แสดงเมนูให้ผู้เล่นเลือก

โปรแกรมแสดงตัวเลือก เช่น

Rock Paper Scissors !exit = จบเกม

รับค่าจากผู้เล่น

- รับค่าที่ผู้เล่นพิมพ์จากแป้นพิมพ์
- ตรวจสอบว่าเลือก ออกจากเกมหรือไม่

ตรวจสอบเงื่อนไขออกจากเกม

ถ้าผู้เล่นพิมพ์ !exit

- ออกจากลูปการเล่น
- ไปสู่ขั้นตอนสรุปผล

ถ้าไม่ใช่

- โปรแกรมทำงานต่อ

ผู้เล่นใช้คำสั่งลับ Secret หรือไม่

- ถ้าใช่ --> เพิ่มคะแนน 100
- ถ้าไม่ใช่ --> เช็คว่าผู้เล่นใช้คำสั่ง bonus หรือไม่
 ↳ ถ้าใช่ --> เพิ่มคะแนน 20

คอมพิวเตอร์สุ่มตัวเลือก

- ใช้ Random สุ่มค่า 0 1 2
- แทนค่า Rock / Paper / Scissors

เปรียบเทียบผลการเล่น

- ใช้เงื่อนไข if-else ตรวจสอบผล
- ผู้เล่นชนะ → เพิ่ม win และ score
- ผู้เล่นแพ้ → เพิ่ม lose
- เสมอ → เพิ่ม draw

แสดงผลพัลซ์รอบนั้น

- ผู้เล่นเลือกอะไร
- คอมพิวเตอร์เลือกอะไร
- ใครชนะในรอบนั้น

เล่นต่อไหม

- โปรแกรมวนกลับไปที่ ขั้นตอนแรก แสดงตัวเลือกให้ดู
- ผู้เล่นสามารถเล่นซ้ำได้เรื่อย ๆ

สรุปผล (End)

- เมื่อผู้เล่นออกจากเกม (lexit) :
- แสดงจำนวนครั้งที่ ชนะ / แพ้ / เสมอ

- แสดงคะแนนรวม
- โปรแกรมจบการทำงาน

สรุปความสำคัญของผังงาน

ผังงานนี้ช่วยให้เข้าใจลำดับการทำงานของโปรแกรมง่ายขึ้น

ทำให้เห็นโครงสร้างที่ใช้อธิบายลำดับการทำงานในรายงานหรือการนำเสนอได้อย่างชัดเจน

ลิงค์ Github: <https://github.com/noppawattansuk/Rock-Paper-Scissor-17->

- Team Repository name: Rock-Paper-Scissor-17
- Folder: RockPaperScissor

วิธี Compile

- เปิด Terminal แล้วพิมพ์ `javac RockPaperScissor.java` เพื่อสร้าง class
- จากนั้นพิมพ์ `java RockPaperScissor` เพื่อรันโค้ด

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\GitAssignment\RockPaperScissor> javac RockPaperScissor.java
PS D:\GitAssignment\RockPaperScissor> java RockPaperScissor
Enter rock/paper/scissor or !exit : █
```