

การออกแบบและพัฒนาเกมเป่ายิงฉุบด้วยภาษา Java

กลุ่มที่ 02

6830300444 นางสาว เบญญาพร ปัญญา

6830300517 นางสาว พรลภัส ผาสุข

6830300886 นางสาว สิริภัทร ประภัสสรารมย์

เสนอ

ผศ.ดร.กุลวดี สมบูรณ์วิวัฒน์

วิชา

03603112 หลักการโปรแกรมเบื้องต้น II

สารบัญ

กฎของเกม Magic Elements	3
ตัวเลือกที่เล่นได้	
กติกาการแพ้-ชนะ	
การให้คะแนน	
ตัวอย่างการเล่นเกม	4
การออกแบบโปรแกรม	5
โครงสร้างโปรแกรม	6-10
ซอร์สโค้ดบน GitHub	11

โปรแกรมเกม Magic Elements

1. กฎของเกม Magic Elements

Magic Elements เป็นเกมที่ผู้เล่นต้องเล่นด้วยการพิมพ์คำสั่งอย่างใดอย่างหนึ่ง และคอมพิวเตอร์จะสุ่มเลือก 1 อย่างเช่นกัน โดยการตัดสินใจแพ้-ชนะ เป็นไปตามกฎกติกาที่กำหนด

1.1 ตัวเลือกที่เล่นได้

- Fire
- Water
- Plant
- พิมพ์ !score เพื่อดูคะแนน
- พิมพ์ !exit เพื่อออกเกม

1.2 กติกาการแพ้-ชนะ

- Fire ชนะ Plant
- Water ชนะ Fire
- Plant ชนะ Water
- ถ้าเลือกเหมือนกันจะถือว่าเสมอ

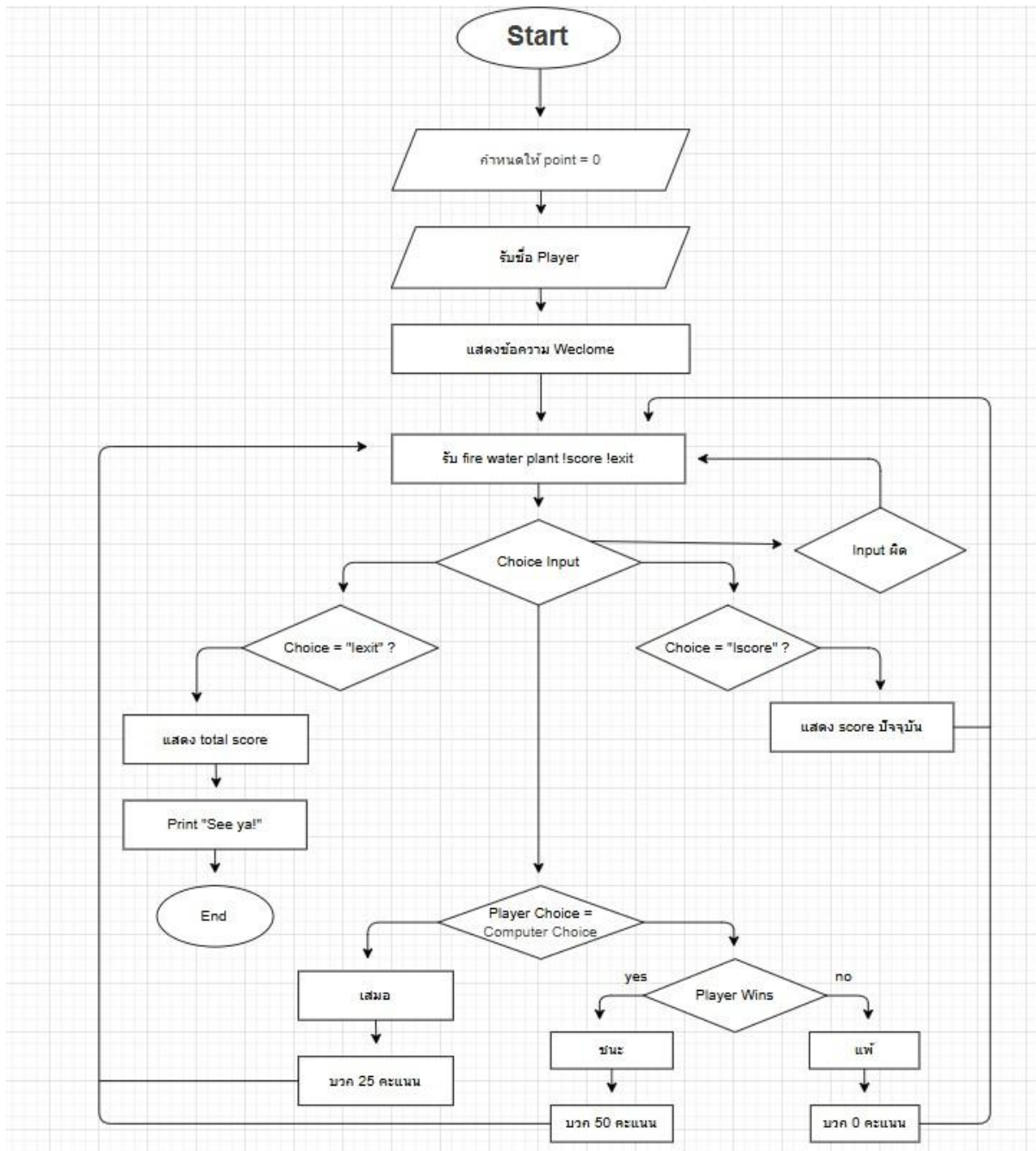
1.3 การให้คะแนน

- ชนะ → ได้ 50 คะแนน
- แพ้ → ได้ 0 คะแนน
- เสมอ → ได้ 25 คะแนน

2. ตัวอย่างการเล่นเกม

```
Enter your name: Bam
Hello, Bam!
Welcome to Magic Elements Game!
Enter fire water and plant to play
If you want to know the score enter "!score"
If you want to end this game enter "!exit"
Let's play Magic Elements!
> water
At least you didn't lose.
Because the computer pick water too.
> fire
Well played!! Because the computer ....
Pick plant!!
> fire
Good pick!! Because the computer ....
Pick plant!!
> plant
At least you didn't lose.
Because the computer pick plant too.
> !score
Your score: 150
> fire
Better luck next time...
Because the computer pick water !!
> !exit
Your total score is 150!
See ya!
```

3. การออกแบบโปรแกรม



4. โครงสร้างโปรแกรม

```
1 import java.util.Scanner;
2 import java.util.Random;
```

4.1) โปรแกรมได้ทำการนำเข้าไลบรารี Scanner และ Random จากแพ็คเกจ java.util เพื่อใช้ในการทำงานดังนี้

- Scanner

ใช้สำหรับรับข้อมูลที่ผู้ใช้ป้อนเข้ามาทางแป้นพิมพ์ (Input) เช่น ชื่อผู้เล่นและคำสั่งในการเล่นเกม

- Random

ใช้สำหรับสร้างค่าตัวเลขแบบสุ่ม เพื่อนำไปใช้ในการสุ่มตัวเลือกของเกมคอมพิวเตอร์ภายในเกม

```
1 import java.util.Scanner;
2 import java.util.Random;
3
4 public class RPS {
5
6     public static String greeting(Scanner scanner) {
7         String player;
8
9         System.out.print(s: "Enter your name: ");
10        player = scanner.nextLine();
11
12        System.out.println("Hello, " + player + "!");
13        System.out.println(x: "Welcome to Magic Elements Game!");
14        System.out.println(x: "Enter fire water and plant to play");
15        System.out.println(x: "If you want to know the score enter \"!score\"");
16        System.out.println(x: "If you want to end this game enter \"!exit\"");
17        System.out.println(x: "Let's play Magic Elements!");
18
19        return player;
20    }
21 }
```

4.2) ฟังก์ชัน greeting() ทำหน้าที่รับค่า Scanner เพื่อใช้ในการรับชื่อผู้เล่นจากผู้ใช้ทางแป้นพิมพ์ หลังจากนั้นโปรแกรมจะแสดงข้อความทักทายโดยแสดงคำว่า

“Hello, ชื่อผู้เล่น!”

นอกจากนี้ ฟังก์ชันนี้ยังทำหน้าที่แสดงข้อความต้อนรับผู้เล่น พร้อมทั้งอธิบายกติกาและวิธีการเล่นเกม รวมถึงคำสั่งต่าง ๆ ที่ผู้เล่นสามารถใช้ระหว่างการเล่นได้

```

21
22     public static int userWin(String win,String computer){
23         System.out.println(win +" Because the computer .... ");
24         System.out.println("Pick " + computer + "!!");
25         return 50;
26     }
27

```

4.3) ฟังก์ชัน userWin() ทำหน้าที่รับค่าข้อมูลชนิด String จำนวน 2 ค่า ได้แก่ ข้อความที่ถูกสุ่มขึ้นมาเพื่อแสดงเมื่อผู้เล่นชนะเกม และตัวเลือกที่คอมพิวเตอร์สุ่มได้และส่งเข้ามาในฟังก์ชัน จากนั้นฟังก์ชันจะทำการแสดงข้อความแจ้งว่าผู้เล่นเป็นฝ่ายชนะ พร้อมทั้งแสดงตัวเลือกที่คอมพิวเตอร์สุ่มได้ และคืนค่าคะแนนจำนวน 50 คะแนน กลับไปยังโปรแกรมหลัก

```

27
28     public static int userLose(String lose,String computer){
29         System.out.println(lose);
30         System.out.println("Because the computer pick " + computer + " !!");
31         return 0;
32     }
33

```

4.4) ฟังก์ชัน userLose() ทำหน้าที่รับค่าข้อมูลชนิด String จำนวน 2 ค่า ได้แก่ ข้อความที่ถูกสุ่มขึ้นมาเพื่อแสดงเมื่อผู้เล่นแพ้เกม และตัวเลือกที่คอมพิวเตอร์สุ่มได้และส่งเข้ามาในฟังก์ชัน จากนั้นฟังก์ชันจะทำการแสดงข้อความปลอบใจผู้เล่นเมื่อแพ้ พร้อมทั้งแสดงตัวเลือกที่คอมพิวเตอร์สุ่มได้ และคืนค่าคะแนนจำนวน 0 คะแนน กลับไปยังโปรแกรมหลัก

```

34     public static int userDraw(String computer) {
35         System.out.println(x: "At least you didn't lose.");
36         System.out.println("Because the computer pick "+ computer + " too.");
37         return 25;
38     }

```

4.5) ฟังก์ชัน userDraw() ทำหน้าที่รับค่าข้อมูลชนิด String ซึ่งเป็นตัวเลือกที่คอมพิวเตอร์สุ่มได้และส่งเข้ามาในฟังก์ชัน จากนั้นฟังก์ชันจะแสดงข้อความแจ้งผลว่าเกมจบลงด้วยการเสมอ พร้อมทั้งแสดงตัวเลือกที่คอมพิวเตอร์สุ่มได้ และคืนค่าคะแนนจำนวน 25 คะแนน กลับไปยังโปรแกรมหลัก

```

39
40
41 public static int playGame(String win,String lose,String playerChoice, String computer){
42     int point =0 ;
43
44     if(playerChoice.equals(computer)){
45         point = userDraw(computer);
46     }else if (playerChoice.equals(anObject: "fire")){
47         if(computer.equals(anObject: "plant")){
48             point = userWin(win, computer);
49         }else{
50             point = userLose(lose,computer);
51         }
52     }else if(playerChoice.equals(anObject: "water")){
53         if(computer.equals(anObject: "fire")){
54             point = userWin(win, computer);
55         }else{
56             point = userLose(lose,computer);
57         }
58     }else if(playerChoice.equals(anObject: "plant")){
59         if(computer.equals(anObject: "water")){
60             point = userWin(win, computer);
61         }else{
62             point = userLose(lose,computer);
63         }
64     }
65
66     else {
67         System.out.println(x: "Invalid input!");
68     }
69
70     return point;
71 }

```

4.6) เริ่มต้นด้วยการกำหนดค่า point เท่ากับ 0

-หาก playerChoice มีค่าเท่ากับ computer

→ ให้เรียกใช้ฟังก์ชัน userDraw() เพื่อคำนวณคะแนนในกรณีเสมอ แล้วเก็บค่าใน point

-หาก playerChoice เท่ากับ "fire"

 ถ้า computer เท่ากับ "plant"

→ ให้เรียกใช้ฟังก์ชัน userWin() เพื่อคำนวณคะแนนในกรณีชนะ

 ถ้าไม่ใช่

→ ให้เรียกใช้ฟังก์ชัน userLose() เพื่อคำนวณคะแนนในกรณีแพ้

-หาก playerChoice เท่ากับ "water"

 ถ้า computer เท่ากับ "fire"

→ ให้เรียกใช้ฟังก์ชัน userWin()

 ถ้าไม่ใช่

→ ให้เรียกใช้ฟังก์ชัน userLose()

-หาก playerChoice เท่ากับ "plant"

 ถ้า computer เท่ากับ "water"

→ ให้เรียกใช้ฟังก์ชัน userWin()

 ถ้าไม่ใช่

→ ให้เรียกใช้ฟังก์ชัน userLose()

-หากผู้เล่นป้อนข้อมูลไม่ถูกต้อง

→ แสดงข้อความว่า Invalid input

-สุดท้าย

→ คำนวณค่าตัวแปร point กลับไปยังโปรแกรมหลัก

```

72
73     public static String getComputerChoice(){
74         String[] computerChoice = {"fire","water","plant"};
75         Random random = new Random();
76         int computerChoiceIndex = random.nextInt(computerChoice.length);
77
78         return computerChoice[computerChoiceIndex];
79     }
80

```

4.7) ฟังก์ชัน getComputerChoice() ทำหน้าที่สุ่มตัวเลือกของคอมพิวเตอร์

โดยจะสร้าง **อาร์เรย์** ชื่อ computerChoice ซึ่งเก็บค่าตัวเลือก "fire", "water", และ "plant" จากนั้น

ฟังก์ชันจะใช้ค่าที่สุ่มตำแหน่งของอาร์เรย์นี้เพื่อนำตัวเลือกที่สุ่มได้ส่งกลับไปใช้ในการเล่นเกม

```

81     public static String Winchooice(){
82         String[] Win = {"Nice pick!!", "Nice choice!!", "Good pick!!", "Well played!!"};
83         Random rd = new Random();
84         int WinchoiceIndex = rd.nextInt(Win.length);
85
86         return Win[WinchoiceIndex];
87     }
88
89     public static String Losechooice(){
90         String[] Lose = {"Bad luck T-T", "Unlucky T-T", "Better luck next time...", "Nice try."};
91         Random RD = new Random();
92         int WinchoiceIndex = RD.nextInt(Lose.length);
93
94         return Lose[WinchoiceIndex];
95     }
96

```

4.8) ฟังก์ชัน Winchooice() ทำหน้าที่สุ่มข้อความคำชมเมื่อผู้เล่นชนะเกม โดยจะสร้าง **อาร์เรย์** ชื่อ

Win ซึ่งเก็บข้อความคำชมหลาย ๆ ข้อความ จากนั้นฟังก์ชันจะสุ่มตำแหน่งของอาร์เรย์นี้และส่ง

ข้อความที่สุ่มได้กลับไปใช้

ฟังก์ชัน Losechooice() ทำหน้าที่สุ่มข้อความปลอบใจเมื่อผู้เล่นแพ้เกม โดยจะสร้าง **อาร์เรย์** ชื่อ

Lose ซึ่งเก็บข้อความปลอบใจหลาย ๆ ข้อความ จากนั้นฟังก์ชันจะสุ่มตำแหน่งของอาร์เรย์นี้และส่ง

ข้อความที่สุ่มได้กลับไปใช้

```

Run | Debug
97 public static void main(String[] args) {
98     int point = 0;
99     Scanner scanner = new Scanner(System.in);
100
101     greeting(scanner, point);
102
103     while(true){
104         System.out.print(s: "> ");
105         String choice = scanner.nextLine();
106
107         if(choice.equals(anObject: "lexit")){
108             System.out.println("Your total score is " + point + "!");
109             System.out.println(x: "See ya!");
110             break;
111         }
112         if(choice.equals(anObject: "lscore")){
113             System.out.println("Your score: " + point);
114             continue;
115         }
116
117         String computer = getComputerChoice();
118         String win = Winchooice();
119         String lose = Losechooice();
120         point += playGame(win,lose,choice, computer);
121     }
122 }
123
124
125 }

```

4.9) เริ่มต้นโปรแกรมด้วยการ เรียกใช้ฟังก์ชัน `greeting()` เพื่อรับชื่อผู้เล่นและแสดงข้อความ

ต้อนรับพร้อมกติกาการเล่น

-จากนั้นโปรแกรมจะเข้าสู่ **ลูป** ซึ่งทำหน้าที่วนซ้ำเพื่อรับคำสั่งจากผู้เล่น

-ภายในลูป โปรแกรมจะตรวจสอบคำสั่งพิเศษ เช่น

- `lexit` → ออกจากเกมและแสดงคะแนนรวม
- `lscore` → แสดงคะแนนปัจจุบัน
- หากผู้เล่นป้อนตัวเลือก "fire", "water", หรือ "plant" โปรแกรมจะ
- สุ่มตัวเลือกของคอมพิวเตอร์ (`getComputerChoice()`)
- สุ่มข้อความคำชม (`Winchooice()`) และข้อความปลอบใจ (`Losechooice()`)
- เรียกใช้ฟังก์ชัน `playGame()` เพื่อตรวจสอบผลเกมและคำนวณคะแนน
- คะแนนที่ได้จากแต่ละรอบจะถูกสะสมในตัวแปร `point`

5. ซอร์สโค้ดบน GitHub

<https://github.com/pornlaphatph/F2-256802-T02/tree/main>