

JAVA 作品

--拉霸機遊戲--

報告者:陳泓睿

目錄

- 簡介
- 流程圖
- 工具
- 作品展示

簡介

製作拉霸機遊戲，在進入遊戲時初始金額為50，可輸入下注金額，下注金額不可超過總金額，每次點選下注會隨機出現三張圖，當三張圖相同時為中獎，會根據連線圖案獲得相應獎金倍數，當總金額為0時宣告破產。休閒時，快來體驗拉霸機的樂趣吧。

製作流程圖

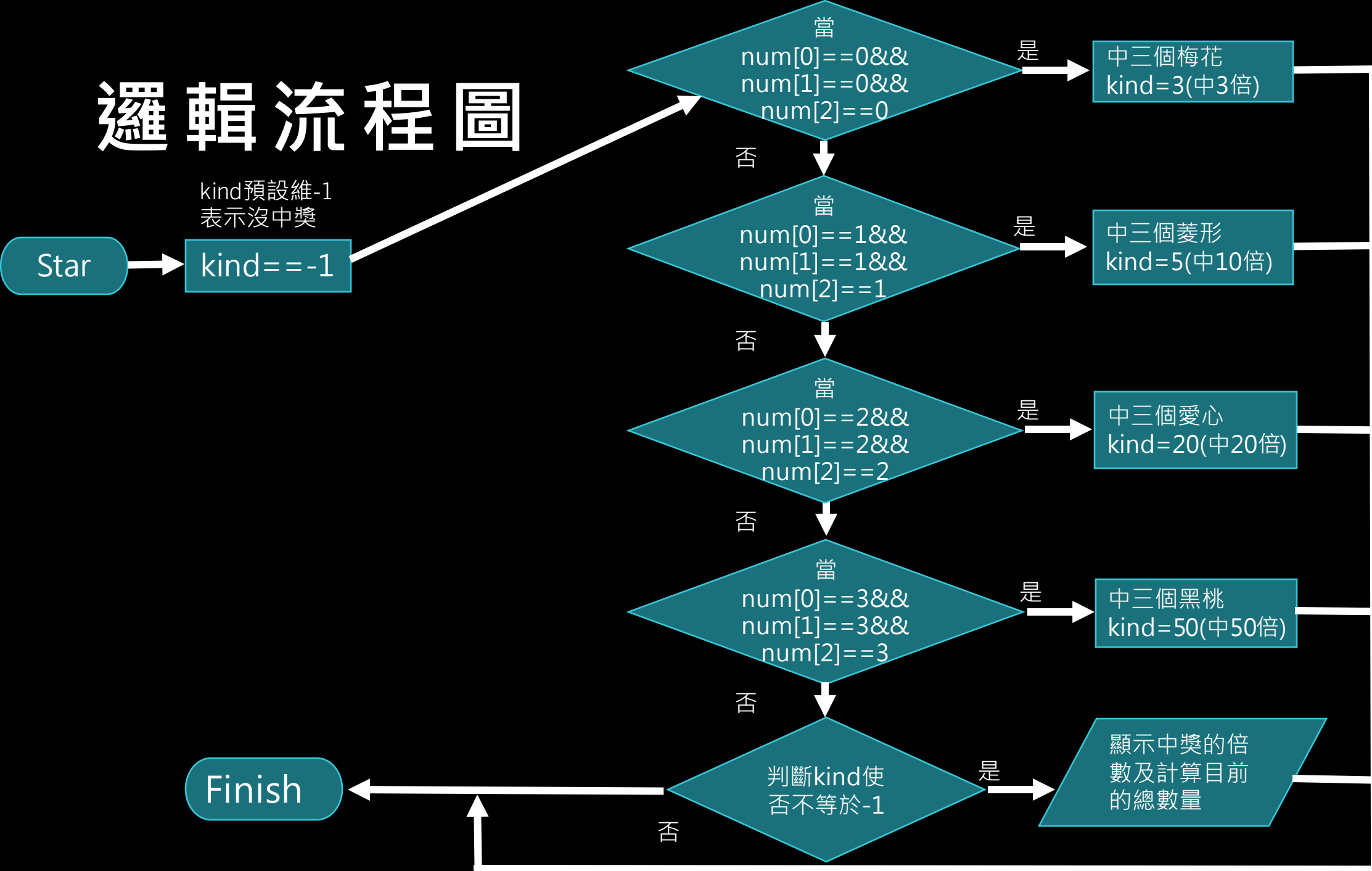
使用java語言及內建類別
方法製作拉霸機視窗

將拉霸機遊戲功能物件化
逐一加進架構

使用exe4j將java及jar檔打
包成exe執行檔，確認exe
檔，可正常執行

使用vscode測試操作、功
能均正常

邏輯流程圖



使用工具



使用vscode撰寫code及程式執行測試



使用pyinstaller將程式相關的所有東西給打包成一個.exe執行檔



使用Java語言撰寫相關程式碼

作品展示



```
1 // 主程式
2 public class slot_machine {
3     public static void main(String[] args) {
4         // 建立MyJFrame視窗 (拉霸遊戲)
5         new MyJFame(); // 呼叫MyJFrame()預設建構式
6     }
7 }
8
```



```
1 // package Slot Machine;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 // MyJFame(拉霸遊戲視窗)繼承JFrame視窗元件
8 // MyJFame實作ActionListener介面的actionPerformed方法用來處理按鈕的按一下事件
9 class MyJFame extends JFrame implements ActionListener {
10     private JLabel[] jlbl = new JLabel[3]; // 宣告jlbl[0]~jlbl[2]用來當拉霸機遊戲三個圖
11     // 宣告icon[0]~icon[3]用來存放櫻桃、星星、西瓜、bar四個圖
12     // 四個圖依序為 0.jpg, 1.jpg, 2.jpg, 3.jpg
13     private ImageIcon[] icons = new ImageIcon[4];
14     // 宣告jlblSum標籤用來顯示"總數量:"訊息
15     // 宣告jlblBetting標籤用來顯示"投注量:"訊息
16     private JLabel jlblSum, jlblBetting;
17     private JTextField jtxtBetting; // 宣告jtxtBetting文字方塊用來讓使用者輸入投注量
18     private JButton jbbtnOk; // 宣告jbbtnOk"下注"按鈕
19     private int sum = 50;

```



```
1 // 實作ActionListener介面的actionPerformed方法
2 public void actionPerformed(ActionEvent evt) {
3     // 建立執行緒t物件，並導入Runnable介面物件
4     // 此執行緒用來啟動拉霸遊戲
5     // 讓jlbl[0]~jlbl[2]以亂數方式顯示櫻桃、星星、西瓜、bar四個圖示
6     // 並判斷是否中獎
7     Thread t = new Thread(
8         new Runnable() {
9             public void run() { // 實作Runnable介面的run方法
10                 // k用來計算拉霸遊戲的換圖次數
11                 // kind用來表示中獎次數，kind等於-1表示沒中獎
12                 int k = 0, kind = -1;
13                 int[] n = new int[jlbl.length]; // n[0]~n[2]用來存放產生的亂數值
14                 int betting = 0; // 用來存放投注量
15
16                 try {
17                     // 若sum總數量等於0，表示沒有可用的投注額即離開遊戲
18                     if (sum == 0) {
19                         JOptionPane.showMessageDialog(null, "您已破產了！即將離開遊戲");
20                         System.exit(0);
21                     }
22
23                     betting = Integer.parseInt(jtxtBetting.getText()); // 取得使用者的投注額，並指定給betting
24
25                     // 當總數量小於投注額或投注額小於0，表示金額不足
26                     if (sum < betting || betting <= 0) {
27                         JOptionPane.showMessageDialog(null, "金額不足或金額不對！");
28                         return;
29                     }
30
31                     sum -= betting;
32                     jlblSum.setText("總數量:" + String.valueOf(sum));
33                     jbbtnOk.setEnabled(false); // 按下注額啟動的開關遊戲機馬上立即停用下注額，防止使用者重複按下
34                 } catch (Exception ex) { // 抓取沒輸入下注數量的錯誤
35                     JOptionPane.showMessageDialog(null, "請輸入數字");
36                     return;
37                 }
38
39                 try {
40                     do {
41                         // 產生0~3之間的亂數並指定給n[0]~n[2]
42                         // 並在jlbl[0]~jlbl[2]隨機顯示櫻桃、星星、西瓜、bar圖示
43                         for (int i = 0; i < jlbl.length; i++) {
44                             n[i] = (int) Math.round(Math.random() * 3);
45                             jlbl[i].setIcon(icons[n[i]]);
46                         }
47
48                         k++;
49                         Thread.currentThread().sleep(100); // 目前執行緒暫停0.1秒
50                     } while (k < 10); // 若k大於0，則停止拉霸遊戲
51                 } catch (InterruptedException ex) {
52                 }
53
54                 // 判斷中獎與否
55                 if (n[0] == 0 && n[1] == 0 && n[2] == 0) {
56                     kind = 3; // 三個圖均為櫻桃，得3倍
57                 } else if (n[0] == 1 && n[1] == 1 && n[2] == 1) {
58                     kind = 10; // 三個圖均為星星，得10倍
59                 } else if (n[0] == 2 && n[1] == 2 && n[2] == 2) {
60                     kind = 20; // 三個圖均為西瓜，得20倍
61                 } else if (n[0] == 3 && n[1] == 3 && n[2] == 3) {
62                     kind = 50; // 三個圖均為bar，得50倍
63                 }
64
65                 // 判斷是否中獎，若kind不等於-1表示中獎
66                 if (kind != -1) {
67                     JOptionPane.showMessageDialog(null, "中獎囉！" + String.valueOf(kind) + "倍");
68                     sum = kind * betting; // 目前總數量(總投注額)累加中獎數量
69                     jlblSum.setText("總數量:" + String.valueOf(sum));
70                 }
71
72                 jbbtnOk.setEnabled(true); // 下注額啟用
73             }
74         });
75
76     t.start(); // 啟動執行緒，使拉霸機啟動，此時jlbl[0]~jlbl[2]即以亂數秀圖
77 }
78 }

```



```
1 // 建構式
2 MyJFame() {
3
4     super.setLayout(null); // 不使用版面配置
5
6     super.setTitle("拉霸遊戲機"); // 視窗標題是設為"拉霸遊戲機"
7
8     // 設定icons[0]~icons[3]元件的圖示為barImg資料夾下的 0.jpg~3.jpg
9     for (int i = 0; i < icons.length; i++) {
10         icons[i] = new ImageIcon(".\\Slot Machine\\Image\\" + String.valueOf(i) + ".jpg");
11     }
12
13     // 建立jlbl[0]~jlbl[2]，並指定三個標籤為櫻桃圖(0.jpg)，最後放入視窗內
14     for (int i = 0; i < jlbl.length; i++) {
15         jlbl[i] = new JLabel();
16         jlbl[i].setBounds(i * 100 + 10, 10, 86, 86);
17         jlbl[i].setIcon(icons[i]);
18         add(jlbl[i]);
19     }
20
21     jlblSum = new JLabel("總數量:" + String.valueOf(sum)); // 在視窗放入jlblSum標籤，該標籤顯示"總數量:"
22     jlblSum.setBounds(10, 120, 160, 20); // 設定jlblSum標籤座標x座標10,y座標120,寬160,高20
23     jlblSum.setFont(new Font("微軟黑體", Font.PLAIN, 18));
24     add(jlblSum);
25
26     jlblBetting = new JLabel("投注量:"); // 在視窗放入jlblBetting標籤，該標籤顯示"投注量:"
27     jlblBetting.setBounds(160, 120, 80, 20);
28     jlblBetting.setFont(new Font("微軟黑體", Font.PLAIN, 18));
29     add(jlblBetting);
30
31     jtxtBetting = new JTextField(); // 在視窗放入jtxtBetting文字方塊，讓使用者輸入投注量
32     jtxtBetting.setBounds(240, 120, 50, 25);
33     jlblSum.setFont(new Font("微軟黑體", Font.PLAIN, 18));
34     add(jtxtBetting);
35
36     jbbtnOk = new JButton("下注"); // 在視窗放入jbbtnOk下注按鈕
37     jbbtnOk.setBounds(10, 160, 80, 30);
38     jbbtnOk.setFont(new Font("微軟黑體", Font.PLAIN, 18));
39     add(jbbtnOk);
40
41     // 指定jbbtnOk下注按鈕的候聽者為目前的物件
42     // 因此按下按鈕時會執行目前類別的actionPerformed方法
43     jbbtnOk.addActionListener(this);
44     setSize(320, 250); // 設定視窗大小為寬320、高250
45     setVisible(true); // 顯示視窗
46     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // 設定按視窗的關閉按鈕結束程式
47 }

```

作品展示

