

JAVA 作品

# --記憶大考驗遊戲--

報告者:陳泓睿

# 目錄

- 簡介
- 流程圖
- 工具
- 作品展示

# 簡介

製作記憶大考驗遊戲，進入遊戲後共8張圖片，剛開始圖片全部顯示為？，一次可翻兩張圖片，根據兩張圖片比較，相同圖片會消失，不同則會變回？，當8張圖片都消失即破關。休閒時，快來體驗記憶大考驗遊戲的樂趣吧。

# 製作流程圖

使用java語言製作拉記憶  
大考驗遊戲

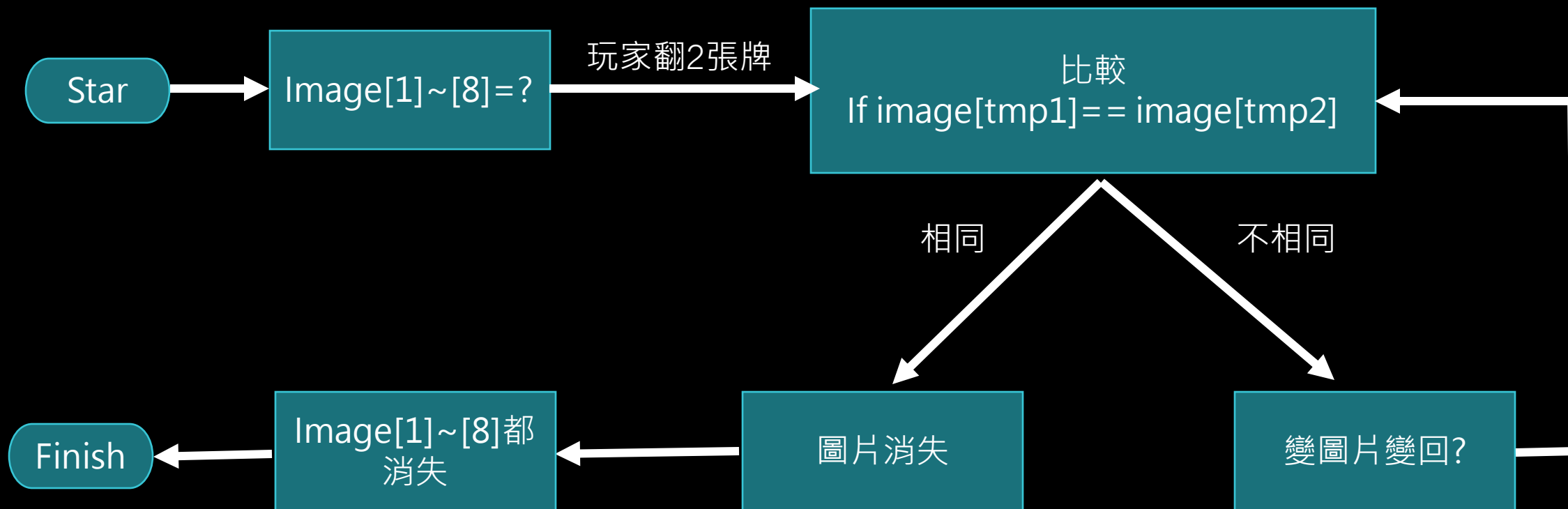


使用vscode測試操作、功  
能均正常



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

# 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 memory_game {
3     public static void main(String[] args) {
4         // 建立MyJFrame視窗 ( 記憶大考驗遊戲 )
5         new MyJFrame(); // 呼叫MyJFrame預設建構式
6     }
7 }
```



```
1 // package Memory Game;
2
3 import java.awt.*; // 使用Font類別請匯入java.awt.*套件
4 import java.awt.event.*; // 使用事件請匯入java.awt.event.*套件
5 import javax.swing.*; // 使用swing元件請匯入javax.swing.*套件
6
7 // MyJFrame(記憶大考驗遊戲視窗)繼承JFrame視窗元件
8 class MyJFrame extends JFrame {
9     // 宣告icons[0]~icons[4]用來存放
10    // 0.jpg(? 問號圖), 1.jpg, 2.jpg, 3.jpg, 4.jpg五個圖示
11    private ImageIcon[] icons = new ImageIcon[5];
12    private JButton[] jbtn = new JButton[8]; // 宣告jbtn[0]~jbtn[7]八個按鈕
13    private JButton jbtnOk, jbtnf, jbtns; // 宣告jbtnOk確定鈕 · jbtnf表示第一次按下的按鈕 · jbtns表示第二次按下的按鈕
14    String f = "", s = ""; // 宣告f表示第一次按下按鈕取得的字串 · s表示第二次按下按鈕取得的字串
15    int num = 0, win = 0; // 宣告num表示按下按鈕的次數; win表示共猜對幾組圖示
16    int[] rnd = new int[8]; // 建立rnd[0]~rnd[7]用來存放記憶大考驗每張圖所代表的編號
17}
```



```
1 // MyJFrame建構式
2 MyJFrame() {
3     super.setLayout(null); // 不使用版面配置
4     super.setTitle("記憶大考驗"); // 視窗標題設為"記憶大考驗"
5
6     // 設定icons[0]~icons[4]元件的圖示為membering資料夾下的0.jpg~4.jpg
7     // 其中0.jpg為問號圖
8     for (int i = 0; i < icons.length; i++) {
9         icons[i] = new ImageIcon(".\\Memory Game\\Image\\" + String.valueOf(i) + ".jpg");
10    }
11
12    // 在窗放入jbtnOk確定鈕
13    jbtnOk = new JButton("確定");
14    jbtnOk.setBounds(28, 260, 80, 30);
15    jbtnOk.setFont(new Font("微軟中黑體", Font.PLAIN, 18));
16    add(jbtnOk);
17
18    // 指定jbtnOk確定鈕的傾聽者為ActionListener匿名物件
19    // 按下確定鈕將會執行該物件的actionPerformed方法
20    jbtnOk.addActionListener(new ActionListener() { // 按下確定鈕執行此處
21        public void actionPerformed(ActionEvent evt) {
22            int[] ary = new int[] { 1, 1, 2, 2, 3, 3, 4, 4 }; // 建立ary[0]~ary[7]用來存放圖示的編號，編號相同的為同一組
23            int n = 0; // 用來存放產生的亂數
24            int max = ary.length - 1;
25
26            // 使用迴圈jbtn[0]~jbtn[7]進行亂數存放1.jpg~4.jpg
27            // 編號相同為同一組
28            for (int i = 0; i < ary.length; i++) {
29                n = (int) Math.round((Math.random() * max));
30                rnd[i] = ary[n];
31                ary[n] = ary[max];
32                max--;
33
34                jbtn[i].setActionCommand(String.valueOf(rnd[i]));
35                jbtn[i].setToolTipText(String.valueOf(i));
36                jbtn[i].setIcon(icons[0]);
37                jbtn[i].setEnabled(true);
38            }
39        }
40    });
```



```
1 // 建立jbtn[0]~jbtn[7]八個按鈕 · 排成兩行 · 一行有四個按鈕
2 int x = 0, y = 0;
3
4 for (int i = 0; i < jbtn.length; i++) {
5     jbtn[i] = new JButton();
6     jbtn[i].setBounds(x * 100 + 20, y * 120 + 20, 80, 100);
7     jbtn[i].setIcon(icons[0]); // 按鈕預設顯示?問號圖
8     jbtn[i].setEnabled(false);
9     x++;
10
11    if (i % 4 == 3) {
12        y++;
13        x = 0;
14    }
15
16    add(jbtn[i]); // 在視窗放入jbtn[0]~jbtn[7]八個按鈕
17
18    // 指定jbtn[0]~jbtn[7]八個按鈕的傾聽者為ActionListener匿名物件
19    // 當按下jbtn[0]~jbtn[7]時會執行該物件的actionPerformed方法
20    jbtn[i].addActionListener(new ActionListener() {
21        public void actionPerformed(ActionEvent evt) {
22            num++; // 按下按鈕次數加1
23
24            if (num == 1) { // 第一次按下
25                f = evt.getActionCommand(); // 取得第一次按下按鈕代表的字串
26                jbtnf = (JButton) evt.getSource(); // 取得第一次按下按鈕
27                jbtn[Integer.parseInt(jbtnf.getToolTipText())].setIcon(icons[Integer.parseInt(f)]);
28            } else if (num == 2) { // 第二次按下
29                s = evt.getActionCommand(); // 取得第二次按下按鈕代表的字串
30                jbtns = (JButton) evt.getSource(); // 取得第二次按下按鈕
31                jbtn[Integer.parseInt(jbtns.getToolTipText())].setIcon(icons[Integer.parseInt(s)]);
32
33                // 若第一次按下按鈕的f字串與第二次按下按鈕的s字串相等
34                // 且第一次按下按鈕與第二次按下按鈕不是同一個 · 則表示猜對一組圖示
35                if (f.equals(s) && jbtns != jbtnf) {
36                    JOptionPane.showMessageDialog(null, "猜對了!");
37                    jbtnf.setEnabled(false); // 第一次按鈕的按鈕停用
38                    jbtns.setEnabled(false); // 第二次按鈕的按鈕停用
39                    win++; // 猜對組數加一
40
41                    if (win == 4) { // 若猜對四組
42                        JOptionPane.showMessageDialog(null, "全對了...ya!");
43                    }
44                } else { // 若沒有猜對任一組圖示 · 則之前按下的按鈕皆還原成?問號圖示
45                    JOptionPane.showMessageDialog(null, "不對哦!");
46                    jbtnf.setIcon(icons[0]);
47                    jbtns.setIcon(icons[0]);
48                }
49
50                f = "";
51                s = "";
52                num = 0;
53            }
54        }
55    });
56
57    setSize(430, 360); // 設定視窗大小為寬430 · 高360
58    setVisible(true); // 顯示視窗
59    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // 設定按視窗的關閉鍵會結束程式
60 }
61 }
62 }
63 }
```

# 作品展示

