

JAVA 程式設計

期末專題報告

題目名稱：顏色敏感度測試

108 學年度第 1 學期

海洋工程學院 資訊工程系 三甲

製作者:黃崧嘉

學號:1106405037



目錄

● 使用 JAVA 環境以及編譯器軟體	-----1
● 程式流程圖	-----2
● 程式介面介紹	-----3
● 程式碼介紹 - 內建函式庫及變數宣告	-----6
● 程式碼介紹 - 建構介面	-----8
● 程式碼介紹 - 畫布繪圖	-----11
● 程式碼介紹 - 滑鼠事件	-----12
● 程式碼介紹 - 函式	-----14
● 程式測試結果	-----17
● 匯出程式檔	-----20

使用 JAVA 環境以及編譯器軟體

使用 JAVA 環境為: jre 1.8.0_231

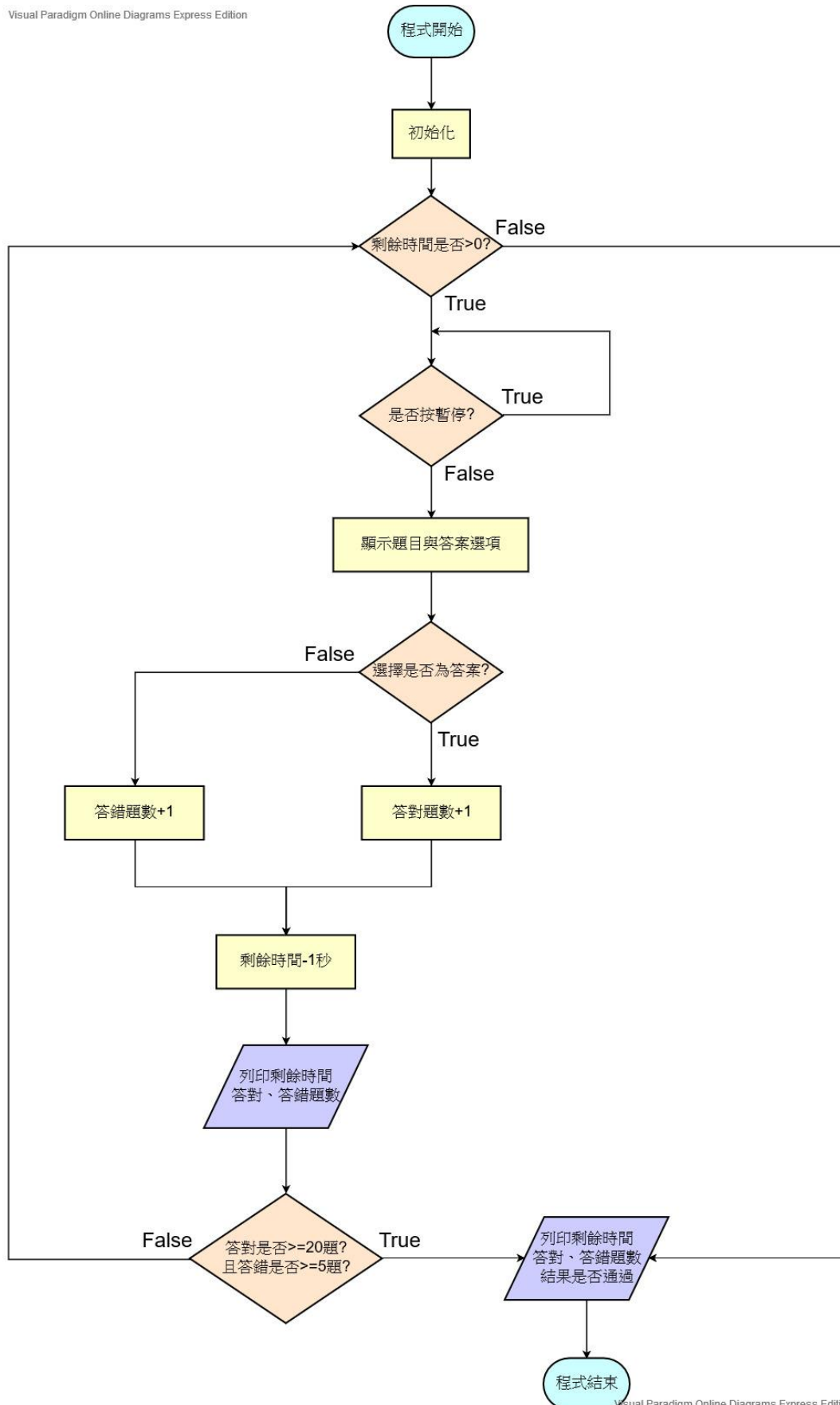


使用編譯器為: Eclipse IDE 2019-09



程式流程圖

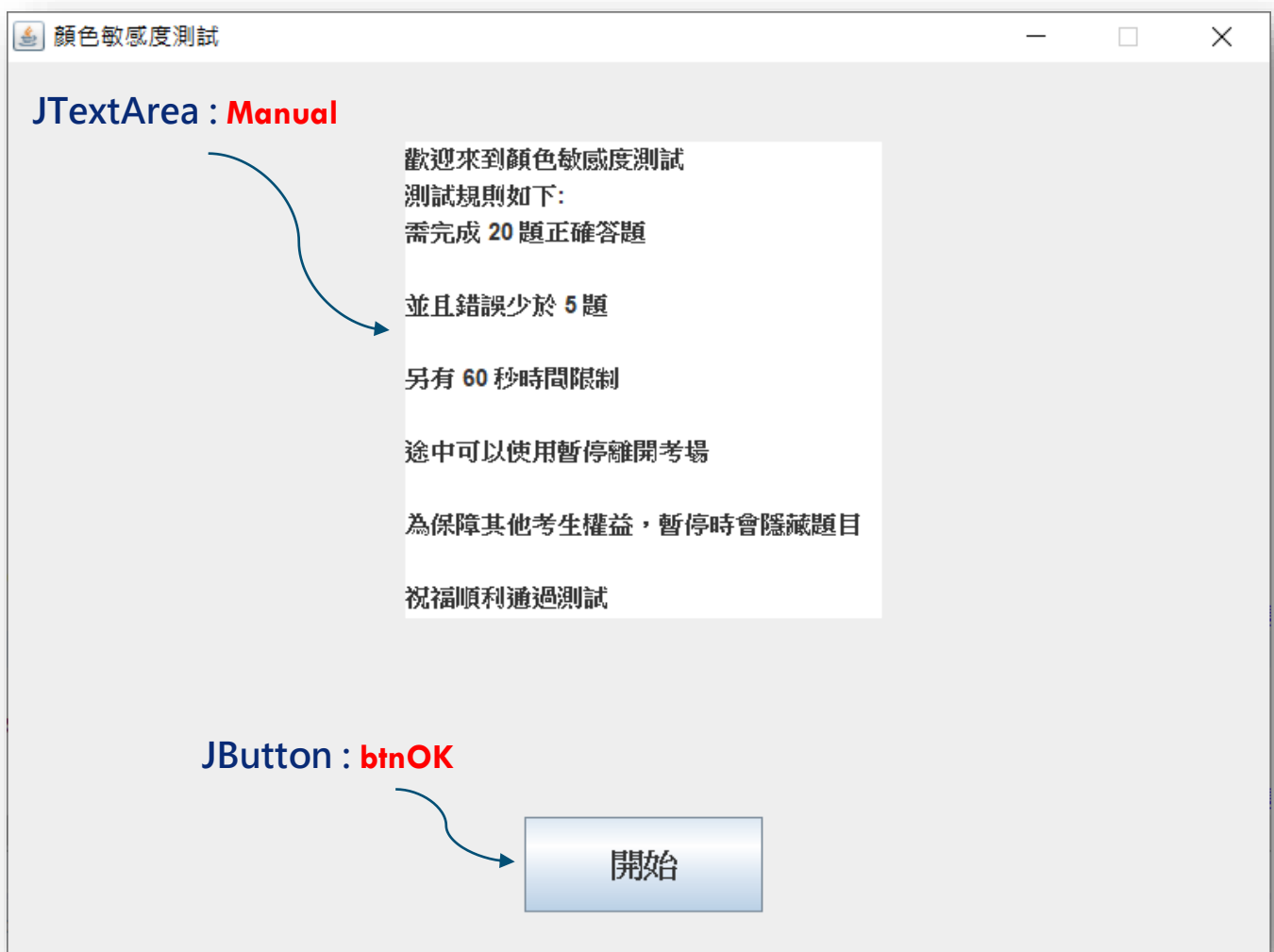
Visual Paradigm Online Diagrams Express Edition



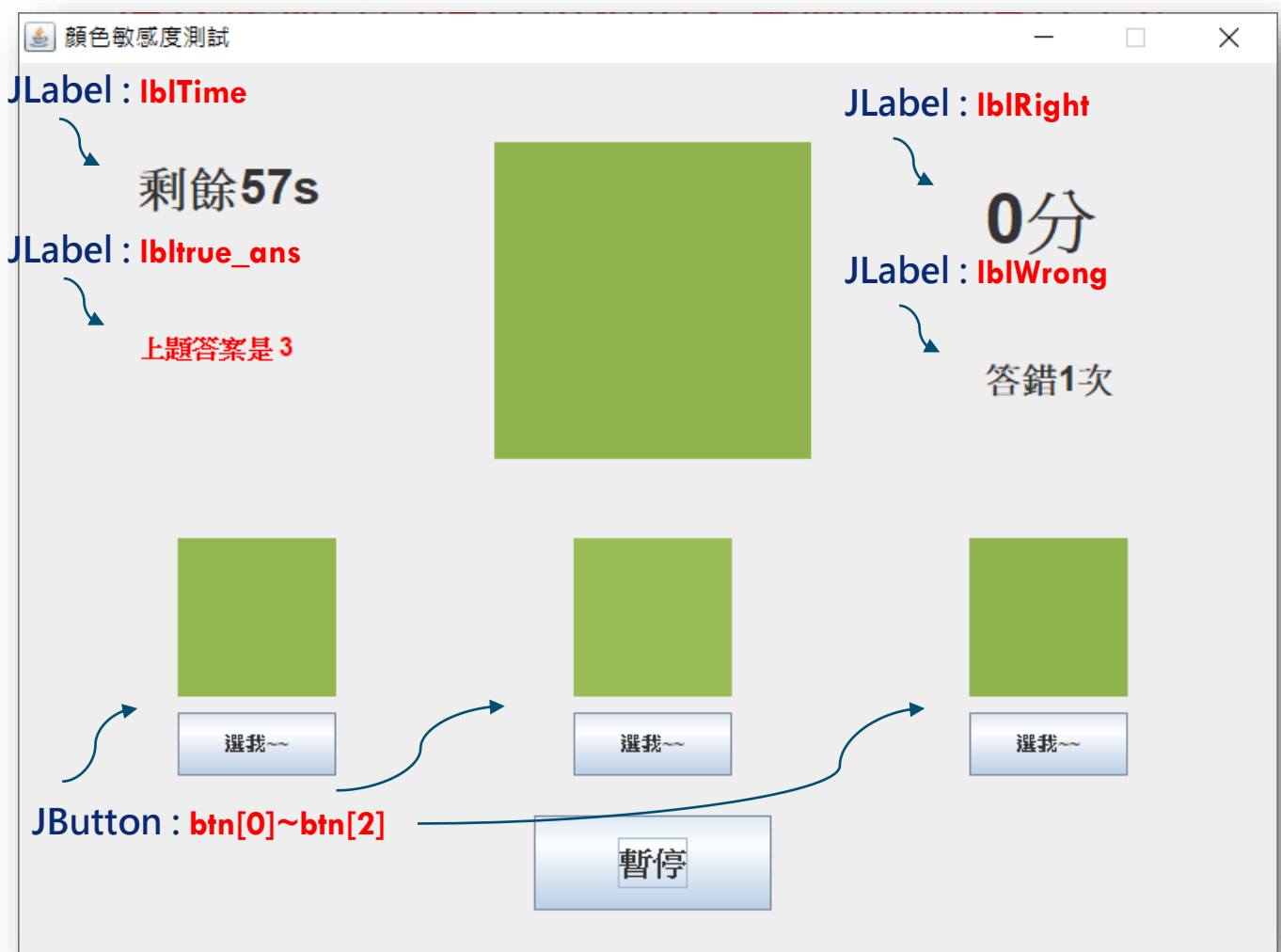
Visual Paradigm Online Diagrams Express Edition

程式介面介紹

進入程式且尚未初始化之畫面



進入程式且已進行初始化之畫面(測試進行中)



進入程式且已進行初始化之畫面(測試暫停)



程式碼介紹 – 內建函式庫及變數宣告

導入內建函式庫 import 區

```
import java.awt.Color;           //繪製顏色及形狀
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D; //-----
import java.awt.event.ActionEvent; //事件傾聽者(ActionListener)
import java.awt.event.ActionListener; //-----
import javax.swing.JButton;       //程式介面物件
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;     //-----
import javax.swing.Timer;         //計時器物件
```


變數宣告區

```
final static int quest=25;
private static int[] ans = new int[quest];
private static JTextArea Manual;
private static JLabel lblWrong, lblRight;
private static JLabel lblTime, lbltrue_ans;
private static JButton btnOK;
private static JButton[] btn = new JButton[3];
private static Canvas pan;
private static Color[][] Q = new Color[quest][3];
private static boolean is_pause=true;
private static boolean is_initial=true;
private static boolean is_clock=false;
private static long tot_time, con_time;
private static int wrong=0, right=0;
private static int R,G,B, choas;
private static int set_time=60, remain_time=99, now=0;

//quest:題目數(固定常數)
//ans[]:正確答案紀錄陣列
//Manual:說明書
//lblWorng:錯誤分數(顯示) lblRight:正確分數(顯示)
//lblTime:時間倒數(顯示)      lbltrue_ans:答題錯誤時的真正答案(顯示)
//btnOK:開始or暫停or繼續or重置
//btn[]:答案按鈕陣列
//pan:畫筆
//Q[]:顏色陣列
//is_pause:暫停or繼續(判斷)
//is_initial:初始化(判斷)
//is_clock:時間計數(判斷)
//tot_time:累計經過時間(計時) con_time:系統當前時間(計時)
//wrong:錯誤分數(計數)      right:正確分數(計數)
//R,G,b:紅,綠,藍 choas:-25~25隨機數
//set_time:設定秒數 remain_time:剩餘秒數 now:現在題目
```

程式碼介紹 – 建構介面

繼承 JFrame 物件建構子區

```
chapter37() {
    setTitle("顏色敏感度測試"); setSize(800,600); setLocationRelativeTo(null);
    setVisible(true); getContentPane().setLayout(null); setResizable(false);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    pan = new Canvas();
    pan.setLayout(null);
    pan.setBounds(0,0,800,600); getContentPane().add(pan);

    Manual = new JTextArea("歡迎來到顏色敏感度測試\n測試規則如下:\n需完成 20 題正確答題\n\n並且錯誤少於 5 題\n\n"
        + "另有 60 秒時間限制\n\n途中可以使用暫停離開考場\n\n為保障其他考生權益,暫停時會隱藏題目\n\n祝福順利通過測試");
    Manual.setBounds(250,50,300,300); Manual.setEditable(false);
    Manual.setFont(new Font("",Font.BOLD,15)); Manual.setLayout(null);
    Manual.setLineWrap(true); Manual.setWrapStyleWord(true); pan.add(Manual);

    btn[0] = new JButton(); btn[0].setBounds(100,410,100,40); btn[0].setVisible(false);
    btn[0].setText("選我~~"); btn[0].addActionListener(new mouseAction()); pan.add(btn[0]);
    btn[1] = new JButton(); btn[1].setBounds(350,410,100,40); btn[1].setVisible(false);
    btn[1].setText("選我~~"); btn[1].addActionListener(new mouseAction()); pan.add(btn[1]);
    btn[2] = new JButton(); btn[2].setBounds(600,410,100,40); btn[2].setVisible(false);
    btn[2].setText("選我~~"); btn[2].addActionListener(new mouseAction()); pan.add(btn[2]);
    btnOK = new JButton(); btnOK.setBounds(325,475,150,60);
    btnOK.setFont(new Font("",Font.BOLD,20));
    btnOK.setText("開始"); btnOK.addActionListener(new mouseAction()); pan.add(btnOK);

    lblRight = new JLabel(); lblRight.setBounds(610,70,100,60); lblRight.setVisible(false);
    lblRight.setText("0分"); lblRight.setFont(new Font("",Font.BOLD,45)); pan.add(lblRight);
    lblWrong = new JLabel(); lblWrong.setBounds(610,170,100,60); lblWrong.setVisible(false);
    lblWrong.setText("答錯0題"); lblWrong.setFont(new Font("",Font.BOLD,22)); pan.add(lblWrong);
    lbltrue_ans = new JLabel(); lbltrue_ans.setBounds(75,150,100,60); lbltrue_ans.setVisible(false);
    lbltrue_ans.setFont(new Font("",Font.BOLD,16)); lbltrue_ans.setForeground(Color.RED); pan.add(lbltrue_ans);

    lblTime = new JLabel(); lblTime.setBounds(75,50,150,60); lblTime.setVisible(false);
    lblTime.setText(""); lblTime.setFont(new Font("",Font.BOLD,30)); pan.add(lblTime);
    pan.repaint();

    Timer();
}
```

第一段為設定繼承 JFrame 屬性之 chapter37 視窗介面之設定，設定標題(Title)為“顏色敏感度測試”；視窗大小(Size) 800*600；視窗位置置中(LocationRelativeTo) 於屏幕；視窗顯示設定(Visible)為 真(true)；視窗佈局(Layout)為 空(不設限)；視窗大小調整(Resizable) 關閉，預設視窗關閉方式(DefaultCloseOperation)為 完全關閉程式並退出。

pan 為 Canvas 物件，其作用為繼承畫布功能，設定畫布大小(Size) 800*600，等同於涵蓋了 JFrame 的大小，設定佈局(Layout)為 空(不設限)。

Manual 為 JTextArea 物件，其作用為說明本程式使用方法及成功與失敗之條件，設定文字區域內容為“

歡迎來到顏色敏感度測試

測試規則如下：

需完成 20 題正確答題

並且錯誤少於 5 題

另有 60 秒時間限制

途中可以使用暫停離開考場

為保障其他考生權益，暫停時會隱藏題目

祝福順利通過測試”；

設定大小(Size)為 300*300 ; 設定文字區域內不得做任何的修改(新增、刪除等.....) ; 設定字體大小為 15pt、粗體 ; 設定佈局(Layout)為空(不設限) ; 設定自動換行(LineWrap)為 真(true) ; 設定自動換行之方式(WrapStyleWord) 為真(true) 。 ex: WrapStyleWord 若為真(true)則換行動作會在單詞結束(空白)才進行換行，若為否(false)則自動換行時會直接做換行動作，此動作可能會斷開單字，所以設定為 true 。

btn[0]~btn[2]、btnOK 為 JButton 物件，其作用為選取答案的按鈕選項，設定大小(Size)為 100*40、150*60 ; 設定顯示(Visiable)為 假(false) ; 設定文字(Text)為“選我~~” ; 將按鈕加入滑鼠事件(mouseAction)中。

lblRight、lblWrong、lblTime、lbltrue_ans 為 JLabel 物件，其作用為顯示程式所需各項數值，設定文字顯示(Visible)為 假(false)，需經過初始化後才會顯示各項數值 ; 再以各文字需求做文字大小、粗細、顏色的調整。

Timer 為 Timer(計時器)物件，其作用為限制答題時間之倒數計時。

程式碼介紹 – 畫布繪圖

繼承 JPanel 物件 Canvas 之類別

```
class Canvas extends JPanel{
    private static final long serialVersionUID = 1L;

    public void paint(Graphics g) {
        super.paint(g);
        Graphics2D g1 = (Graphics2D)g;

        if(!is_pause) {
            for(int i=0;i<3;i++) {
                g1.setColor(Q[now][i]);
                if(i==ans[now]) {
                    g1.fill(new Rectangle2D.Double(100+(250*i),300,100,100));
                    g1.fill(new Rectangle2D.Double(300,50,200,200));
                }
                else {
                    g1.fill(new Rectangle2D.Double(100+(250*i),300,100,100));
                }
            }
        }
        else {
            g1.setColor(null);
        }
    }
}
```

一開始先繼承父類別 paint 並用 g 承接，在宣告 g1 為 g 的實作物件，若 is_pause(暫停觸發)為 false，則迴圈從 0 開始運作至 2，g1 畫筆顏色設定為 Q Color 陣列中的當前題數顏色，若迴圈運作至當題正確答案時，則題目與答案位置繪製成同一顏色，其餘兩格因非正確答案則畫上 Q Color 陣列中事先產生的兩種不同顏色。若 is_pause(暫停觸發)為 true，則將畫筆顏色設為 null，題目及答案在做繪製時則會空白。

程式碼介紹 – 滑鼠事件

實作 ActionListener 物件 mouseAction 之類別

```
class mouseAction implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        if(e.getSource()==btn[0] && !is_pause) {
            if(ans[now]==0) {
                lblRight.setText(++right+"分");
                lbltrue_ans.setText("");
            }
            else {
                lblWrong.setText("答錯"(++wrong)+"次");
                lbltrue_ans.setText("上題答案是 "+(ans[now]+1));
            }
            now++;
        }

        if(e.getSource()==btn[1] && !is_pause) {
            if(ans[now]==1) {
                lblRight.setText(++right+"分");
                lbltrue_ans.setText("");
            }
            else {
                lblWrong.setText("答錯"(++wrong)+"次");
                lbltrue_ans.setText("上題答案是 "+(ans[now]+1));
            }
            now++;
        }

        if(e.getSource()==btn[2] && !is_pause) {
            if(ans[now]==2) {
                lblRight.setText(++right+"分");
                lbltrue_ans.setText("");
            }
            else {
                lblWrong.setText("答錯"(++wrong)+"次");
                lbltrue_ans.setText("上題答案是 "+(ans[now]+1));
            }
            now++;
        }

        if(e.getSource()==btnOK && wrong<5 && right<20 && remain_time>0) {
            is_pause=!is_pause;

            if(is_initial) {
                begin();
                is_clock=true;
                con_time = System.currentTimeMillis();
            }
            else if(is_pause){
                btnOK.setText("繼續");
                is_clock=false;
            }
            else {
                btnOK.setText("暫停");
                is_clock=true;
                con_time = System.currentTimeMillis();
            }
        }
        end();
        pan.repaint();
    }
}
```

若按鈕來源(e.getSource)為 btn[0]且暫停按鈕尚未被按下(is_pause==false)，則再判斷當題正確答案是否也為第 0 題，若為是(true)則正確答題累計(right)+1，lbltrue_ans 因答題正確所以字串內容將為空，不做報錯訊息之字串輸出；若為非(false)則錯誤答題累計(wrong)+1，lbltrue_ans 因答題錯誤所以字串內容更改為正確答案當格之數字並進行報錯，直到下一次答對為止。最後將當前題數(now)+1 往下推進至下一題。若暫停按鈕已經被按下(is_pause==true)，因及閘(AND)運算特性影響，該按鈕無法進行觸發。

btn[1]、btn[2]之判斷與上段 btn[0]之判斷無異，就不另贅述。

btnOK 預設文字為“開始”，做為一次性觸發初始化(is_initial)之開關。若開始被按下後開始執行初始化 begin()函式，時間觸發(is_clock)被觸發；每次此按鈕被按下時，暫停觸發(is_pause)會被反閘(Not)反相當前狀態，按鈕文字也會不斷在“暫停”、“繼續”兩者中做來回切換，時間觸發(is_clock)狀態也會不斷的切換。若剩餘時間(remain_time)或正確答題(right)或錯誤答題(wrong)任一已達標準或暫停觸發為真(is_pause==true)，則此按鈕因及閘(AND)運算特性影響，此按鈕無法進行觸發。

最後執行結束程式判斷 end()函式，查看程式是否已達結束程式的標準，並會進行重繪畫布(repaint)以達到更新畫面的作用。

程式碼介紹 – 函式

初始化 begin() 函式

```
public void begin() {  
    is_initial=false;  
  
    btnOK.setText("暫停");  
    lblRight.setVisible(true);  
    lblWrong.setVisible(true);  
    lbltrue_ans.setVisible(true);  
    lblTime.setVisible(true);  
    btn[0].setVisible(true);  
    btn[1].setVisible(true);  
    btn[2].setVisible(true);  
    Manual.setVisible(false);  
  
    for(int i=0;i<quest;i++) {  
        ans[i]=(int)(Math.random()*3);  
        R=(int)(Math.random()*206)+25;  
        G=(int)(Math.random()*206)+25;  
        B=(int)(Math.random()*206)+25;  
  
        for(int j=0;j<3;j++) {  
            choas=(int)(Math.random()*51)-25;  
            if(choas==0) {j--; continue;}  
  
            if(ans[i]==j) {  
                Q[i][j] = new Color(R,G,B);  
            }  
            else {  
                Q[i][j] = new Color(R+choas,G+choas,B+choas);  
            }  
        }  
    }  
}
```

進到 begin() 函式中，初始化觸發(is_initial)會被關閉，btnOK 按鈕文字被改為“暫停”，Manual 文字說明欄會被隱藏，其餘按鈕與數字顯示欄位狀態從隱藏更改為顯示，每題先產生一隨機值(0~2)的正確答案儲存在 ans[] 陣列中，再產生從 25~225 之隨機值分別儲存在 R、G、B 裡，之後產生-25~25 之隨機亂數(但不能為 0 否則與答案重覆)，最後將

三組不同 RGB 值儲存至 Color 的 Q 陣列中。ex:因 R、G、B 值範圍為 0~255，為防止溢位(Overflow)之發生，RGB 隨機亂數配合-25~25 隨機亂數則產生出 25~225 的範圍。

結束程式判斷 end() 函式

```
public void end() {  
    if(remain_time<=0) {  
        JOptionPane.showMessageDialog(this,"您的成績如下:\n答對題數: "+right+"題\n答錯題數: "+wrong+"  
            "題\n剩餘時間: "+remain_time+"秒","時間已到",JOptionPane.INFORMATION_MESSAGE);  
        JOptionPane.showMessageDialog(this,"也許是判斷不夠果斷，反應力也是很重要的，要相信自己啊\n下次再來過吧 年輕人",  
            "未通過測試",JOptionPane.ERROR_MESSAGE);  
    }  
    else if(wrong>=5) {  
        is_clock=false;  
        JOptionPane.showMessageDialog(this,"您的成績如下:\n答對題數: "+right+"題\n答錯題數: "+wrong+"  
            "題\n剩餘時間: "+remain_time+"秒","成績未達標準",JOptionPane.INFORMATION_MESSAGE);  
        JOptionPane.showMessageDialog(this,"也許是天分不夠，也可能準備不足\n下次再來過吧 年輕人",  
            "未通過測試",JOptionPane.ERROR_MESSAGE);  
    }  
    else if(right>=20) {  
        is_clock=false;  
        JOptionPane.showMessageDialog(this,"您的成績如下:\n答對題數: "+right+"題\n答錯題數: "+wrong+"  
            "題\n剩餘時間: "+remain_time+"秒","成績已達標準",JOptionPane.INFORMATION_MESSAGE);  
        JOptionPane.showMessageDialog(this,"看來您是可造之材啊，一定有雙銳利的眼睛\n\n再次恭喜您通過本測試!!!",  
            "已通過測試",JOptionPane.PLAIN_MESSAGE);  
    }  
    else {  
        return;  
    }  
    System.exit(0);  
}
```

end() 函式負責判斷程式是否已達結束標準。若剩餘時間 (remain_time) ≤ 0 秒或者答錯題數(wrong) ≥ 5 題或者答對題數(right) ≥ 20 題，則會用彈跳訊息視窗(MessageDialog)進行成績的結果宣告，並針對不同的結束方式有不同的情境對話，最後做 System.exit(0)進行程式的完全關閉與離開。若上述三個條件都尚未觸發，則回傳空值至呼叫 end() 函式處繼續進行程式。

計時器 Timer()函式

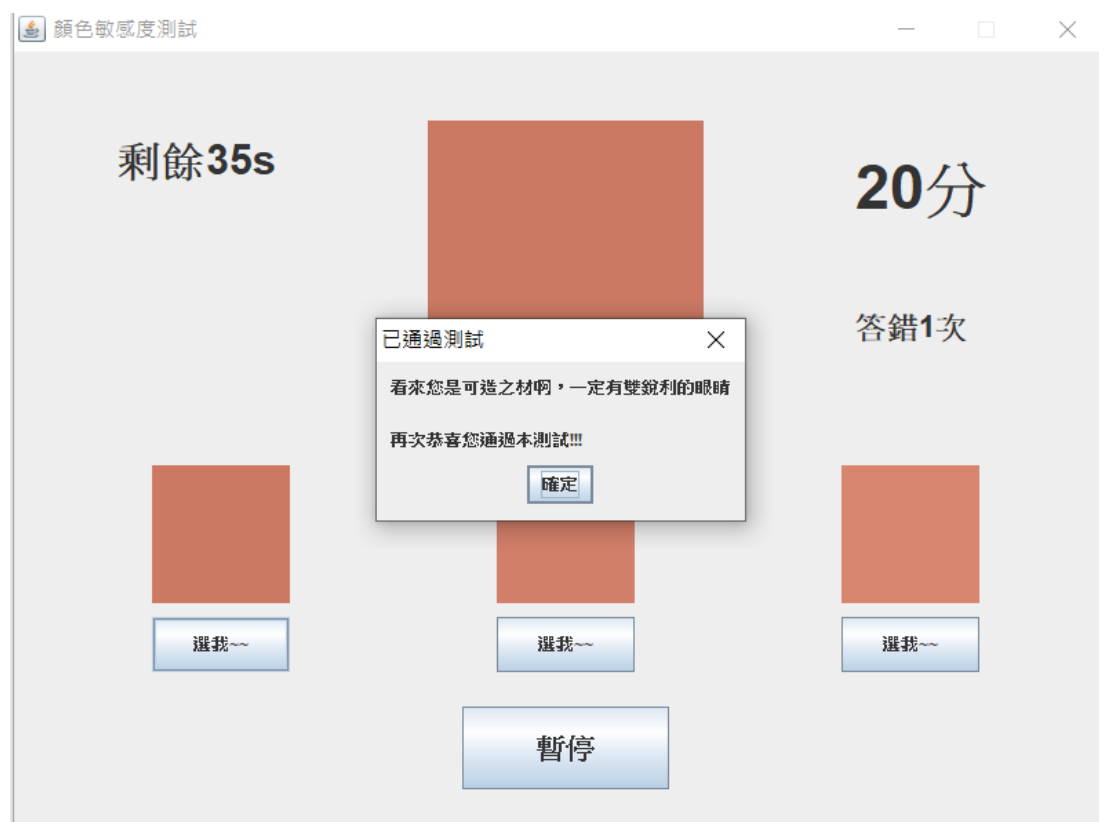
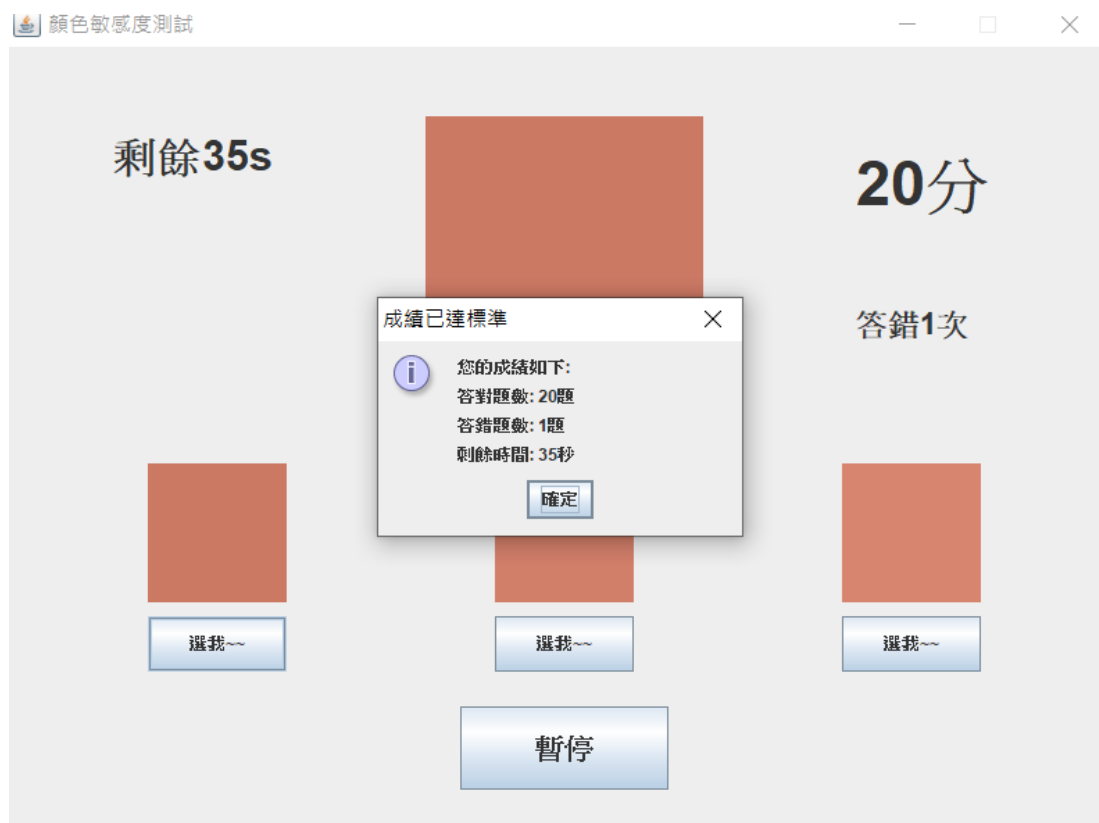
```
public void Timer() {
    ActionListener Array = new ActionListener(){
        public void actionPerformed(ActionEvent actionEvent) {
            if(is_clock) {
                tot_time += System.currentTimeMillis() - con_time;
                remain_time = set_time - (int)(tot_time/1000);
                if(remain_time<=0) {remain_time=0; is_clock=false; is_pause=true; end();}
                lblTime.setText("剩餘"+String.valueOf(remain_time)+"s");
                con_time = System.currentTimeMillis();
            }
        }
    };
    Timer timer = new Timer(1000,Array);
    if(is_initial) {
        timer.start();
    }
    else if(is_pause){
        timer.stop();
    }
    else {
        timer.restart();
    }
}
```

因計時器物件格式要求為 `Timer(int Delay,ActionListener Listener)`，所以宣告 `Array` 為 `ActionListener` 物件專為 `timer` 專用之事件傾聽者。若時間觸發(`is_clock`)為真(`true`)，`con_time` 在滑鼠事件的 `btnOK` 中事先儲存了系統的當前時間(`System.currentTimeMillis`)，回到 `Timer()`中，待 `timer` 延遲毫秒(`Delay`)後，累計時間(`tot_time`)會累加系統當前時間扣除 `con_time` 延遲前紀錄的系統時間，設定時間(`set_time`)再與累計時間(`tot_time`)相減後即可推導出程式剩餘時間。

當初始化觸發(`is_initial`)觸發後，`timer` 計時開始(`timer.start()`)，若暫停被按下(`is_pause==false`)，計時器也會停止運作(`timer.stop()`)，直到再度按下繼續 (`is_pause==true`)，計時器才會繼續運作(`timer.restart()`)。

程式測試結果

當答對題數滿 20 題時



當答錯題數滿 5 題時



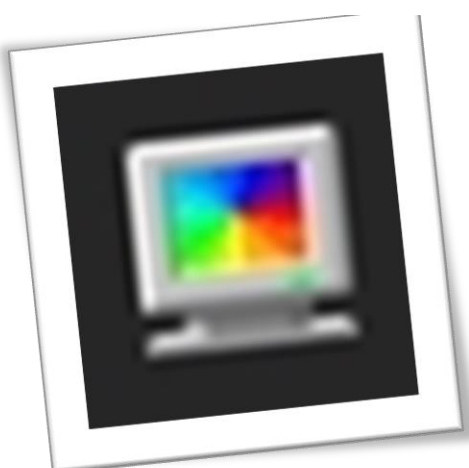
當剩餘時間為 0 秒時



匯出程式檔



[Thematic37.jar](#)



免費軟體 [ColorSPY](#)

說明:偵測滑鼠在螢幕上該座標的 RGB 值