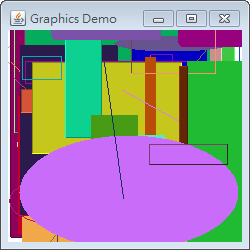
繪圖與音效

1. 請參考投影片內容，建立以下視窗應用程式

請將1.程式**執行結果**截圖置入作業中、2.程式**原始檔**置入作業中

1. 請參考範例程式與投影片,建立繪圖程式
   1. 滑鼠進入視窗時, 使用亂數,任意選擇繪出Line, 3DRect, Oval, Rect, RoundRect等不同形式的線條或區域,繪出的顏色也請使用亂數選擇
   2. 滑鼠離開視窗時,則停止繪圖



import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

public class GraphicsDemo extends java.awt.Frame implements MouseMotionListener { // 實作MouseListener介面

public static void main(String args[]){

new GraphicsDemo();

}

// 建構函式

public GraphicsDemo() {

super("Graphics Demo");

// 設定背景顏色

this.setBackground(Color.white);

// 註冊 MouseListener

this.addMouseMotionListener(this);

// 設定視窗的大小

this.setSize(250, 250);

// Center the frame

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = this.getSize();

if (frameSize.height > screenSize.height)

frameSize.height = screenSize.height;

if (frameSize.width > screenSize.width)

frameSize.width = screenSize.width;

this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

// 顯示視窗

this.setVisible(true);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public void paint(Graphics g) {

int r, x1, x2, y1, y2;

int red, green, blue;

Random random = new Random();

int width = this.getSize().width;

int height = this.getSize().height;

for (int i=0; i<100; i++) {

r = random.nextInt(9);

x1 = random.nextInt(width);

y1 = random.nextInt(height);

x2 = random.nextInt(width);

y2 = random.nextInt(height);

red = random.nextInt(255);

green = random.nextInt(255);

blue = random.nextInt(255);

// 設定Graphics物件之目前顏色

g.setColor(new Color(red, green, blue));

switch (r) {

case 0:

g.drawLine(x1, y1, x2, y2);

break;

case 1:

g.draw3DRect(x1, y1, x2-x1, y2-y1, true);

break;

case 2:

g.drawOval(x1, y1, x2-x1, y2-y1);

break;

case 3:

g.drawRect(x1, y1, x2-x1, y2-y1);

break;

case 4:

g.drawRoundRect(x1, y1, x2-x1, y2-y1, 10, 10);

break;

case 5:

g.fill3DRect(x1, y1, x2-x1, y2-y1, true);

break;

case 6:

g.fillOval(x1, y1, x2-x1, y2-y1);

break;

case 7:

g.fillRect(x1, y1, x2-x1, y2-y1);

break;

case 8:

g.fillRoundRect(x1, y1, x2-x1, y2-y1, 10, 10);

break;

default:

break;

}

}

}

public void mouseDragged(MouseEvent e) {}

public void mouseMoved(MouseEvent e){

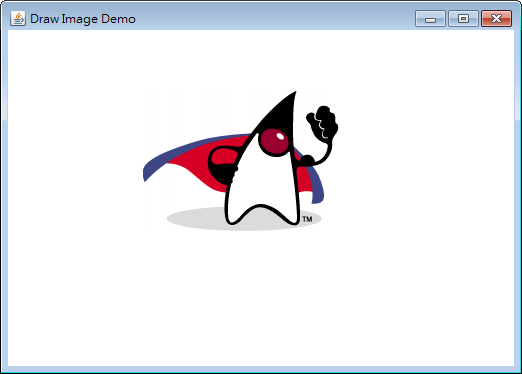
this.repaint();

}

}

1. 請參考範例程式與投影片,建立圖片繪製程式
   1. 滑鼠於視窗內任意點Click一下,則在Click的地方將圖載入與繪出,圖片可自行找

例如在座標10,10 滑鼠click一下,就在座標10,10 將圖片載入繪出



import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

public class DrawImageDemo extends java.awt.Frame implements MouseListener, MouseMotionListener {

int x, y;

Image image;

boolean blnDrag = false;

public static void main(String args[]){

new DrawImageDemo();

}

// 建構函式

public DrawImageDemo() {

super("Draw Image Demo");

// 設定背景顏色

setBackground(Color.white);

// 讀取Java Archive檔案內的圖像檔案

ClassLoader cl = this.getClass().getClassLoader();

Toolkit tk = Toolkit.getDefaultToolkit();

// 取得圖像

image = tk.createImage(cl.getResource("images/SuperDuke.gif"));

// 註冊 MouseListener

this.addMouseListener(this);

// 註冊 MouseMotionListener

this.addMouseMotionListener(this);

// 設定視窗的大小

this.setSize(250, 250);

// Center the frame

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = this.getSize();

if (frameSize.height > screenSize.height)

frameSize.height = screenSize.height;

if (frameSize.width > screenSize.width)

frameSize.width = screenSize.width;

this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

// 顯示視窗

this.setVisible(true);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public void update(Graphics g) {

paint(g);

}

public void paint (Graphics g) {

if (blnDrag) {

// 設定Graphics物件之目前顏色

g.setColor(Color.white);

// 繪製矩形並以目前指定之顏色填滿面積

// 則清除畫面

g.fillRect(0, 0, this.getSize().width, this.getSize().height);

// 繪製圖像

g.drawImage(image, x, y, null);

}

}

// 實作MouseListener介面之方法

public void mousePressed(MouseEvent e){

blnDrag = true;

x = e.getX();

y = e.getY();

this.repaint();

}

public void mouseReleased(MouseEvent e){

blnDrag = false;

}

public void mouseEntered(MouseEvent e){}

public void mouseExited(MouseEvent e){}

public void mouseClicked(MouseEvent e){

}

// 實作MouseMotionListener介面之方法

public void mouseDragged(MouseEvent e){

x = e.getX();

y = e.getY();

repaint();

}

public void mouseMoved(MouseEvent e){

x = e.getX();

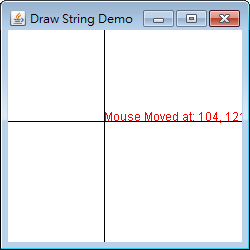
y = e.getY();

repaint();

}

}

1. 請參考範例程式與投影片,建立字串繪製程式
   1. 範例程式中是將字串繪製於最上方,請改寫程式 將字串繪製於滑鼠的位置
   2. 例如滑鼠移到在座標10,10 ,就在座標10,10繪製字串 或者
   3. 滑鼠在座標20,10 click一下,就在座標20,10 將字串繪出



import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

public class DrawStringDemo extends java.awt.Frame implements MouseListener, MouseMotionListener {

String str = "";

int x, y;

public static void main(String args[]){

new DrawStringDemo();

}

// 建構函式

public DrawStringDemo() {

super("Draw String Demo");

// 設定背景顏色

setBackground(Color.white);

// 預設滑鼠指標

setCursor(new Cursor(Cursor.DEFAULT\_CURSOR));

// 註冊 MouseListener

this.addMouseListener(this);

// 註冊 MouseMotionListener

this.addMouseMotionListener(this);

// 設定視窗的大小

this.setSize(250, 250);

// Center the frame

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = this.getSize();

if (frameSize.height > screenSize.height)

frameSize.height = screenSize.height;

if (frameSize.width > screenSize.width)

frameSize.width = screenSize.width;

this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

// 顯示視窗

this.setVisible(true);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public void update(Graphics g) {

paint(g);

}

public void paint (Graphics g) {

// 設定Graphics物件之目前顏色

g.setColor(Color.white);

// 繪製矩形並以目前指定之顏色填滿面積

// 則清除畫面

g.fillRect(0, 0, this.getSize().width, this.getSize().height);

// 設定Graphics物件之目前顏色

g.setColor(Color.RED);

// 設定Graphics物件之文字字型

g.setFont(new Font("dialog", Font.PLAIN, 12));

// 繪製文字

g.drawString(str, x, y);

// 設定Graphics物件之目前顏色

g.setColor(Color.BLACK);

g.drawLine(x, 0, x, this.getSize().height);

g.drawLine(0, y, this.getSize().width, y);

}

// 實作MouseListener介面之方法

public void mousePressed(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Pressed at: " + x + ", " + y;

repaint();

}

public void mouseReleased(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Released at: " + x + ", " + y;

repaint();

}

public void mouseClicked(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Clicked at: " + x + ", " + y;

repaint();

}

public void mouseEntered(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Entered at: " + x + ", " + y;

repaint();

}

public void mouseExited(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Exited at: " + x + ", " + y;

repaint();

}

// 實作MouseMotionListener介面之方法

public void mouseDragged(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Dragged at: " + x + ", " + y;

repaint();

}

public void mouseMoved(MouseEvent e){

x = e.getX();

y = e.getY();

str = "Mouse Moved at: " + x + ", " + y;

repaint();

}

}