

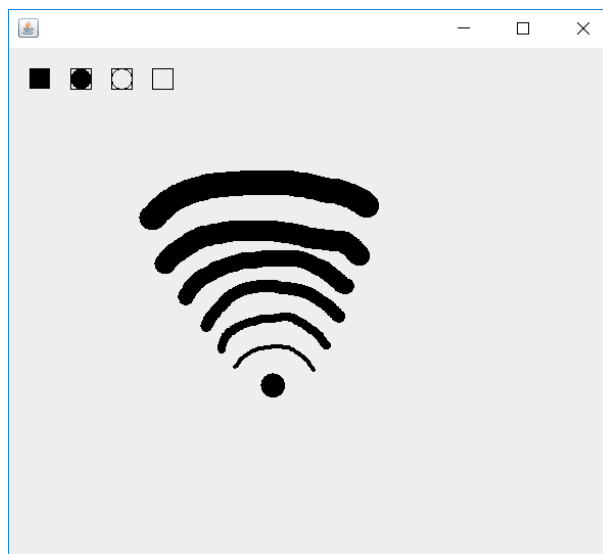
JAVA LAB 3 (4)

Problem:

Write a **JAVA Application** that user can draw something on the **JPanel**.

Requirement:

1. You can't declare any static field and method except the main function.
2. You need to provide four "button". Just like picture below.
3. The first button is to change Color to black.
4. The second and the third button is to increase or decrease the width of the stroke.
5. The last button is an "eraser".
6. All "buttons" are painted on JPanel , and is activated by mouse event.
7. The size of "button" is **20 * 20px**
8. The stroke width varies from **4.0f to 24.0f**. The interval of each step is **4.0f**.
9. **Paint on the JPanel directly**, you don't have to store the brush path.
10. **You need to make the painting smoothly**.
11. **Please use Graphics2D to draw the line**.



Class & method:

Class

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
```

Interface

MouseListener

```
public void mousePressed(MouseEvent event){}
public void mouseReleased(MouseEvent event){}
public void mouseClicked(MouseEvent event){}
public void mouseEntered(MouseEvent event){}
public void mouseExited(MouseEvent event){}
```

MouseMotionListener

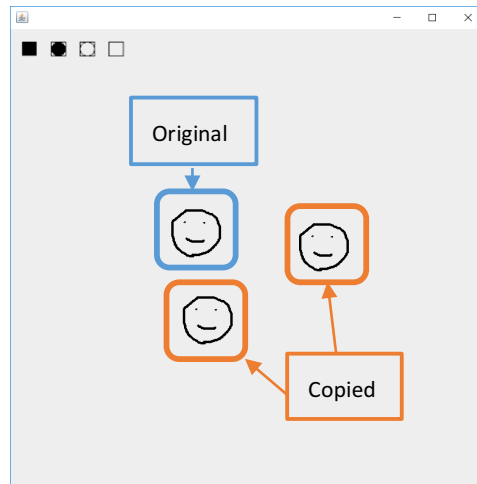
```
public void mouseDragged( MouseEvent event ){}
public void mouseMoved( MouseEvent event ){}

```

```
01. addMouseListener()
02. addMouseMotionListener()
03. event.getX()
04. event.getY()
05. Graphics g = getGraphics();
06. Graphics2D g2 = (Graphics2D) g;
07. setBackground();
08. drawLine(int x1, int y1, int x2, int y2)
09. setStroke(new BasicStroke(lineWidth, BasicStroke.CAP_ROUND,
    BasicStroke.JOIN_ROUND)); // lineWidth = 4.0f
10. drawRect (int x, int y, int width, int height);
11. fillRect (int x, int y, int width, int height);
12. setColor(Color color);
13. setVisible();
14. paintComponent();
15. setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
```

Advance:

Select a rectangle area and do copy/paste operation.



Step:

- 按右鍵選擇第一個點
- 再按右鍵選擇第二個點
- 複製由兩個點組成的方形範圍區域的影像
- 按右鍵放置複製的影像在指定位置

Tips:

- `event.getButton()`
- `MouseEvent.BUTTON3` // Right mouse button
- `MouseEvent.BUTTON1` // Left mouse button
- `copyArea(int x, int y, int width, int height, int dx, int dy)`

Referance:

1. Use JAVA Doc

<http://docs.oracle.com/javase/7/docs/api/overview-summary.html>

2. `getGraphics` VS `paintComponent`

`getGraphics` method

- Can be null
- Is a "snap shot" of the last paint process
- Anything painted to it will be lost on the next paint cycle

You should avoid using `getGraphics` and simply use what is past to the `paintComponent` method.

In theory, there is no difference between them, but if you want what you have painted to survive between repaints, then you should be using `paintComponent`

[source](#)

3. `Graphics` VS `Graphics2D`

[See here](#)