

THỰC HÀNH CÔNG CỤ VÀ MÔI TRƯỜNG VÀ LẬP TRÌNH 2

TS. Võ Phương Bình – Email: binhvp@dlu.edu.vn
Information Technology Faculty - Dalat University
Website: <http://it.dlu.edu.vn/ivp-lab>

LAB 7 (4 tiết): Tổng hợp GUI và xử lý sự kiện

A. Mục tiêu:

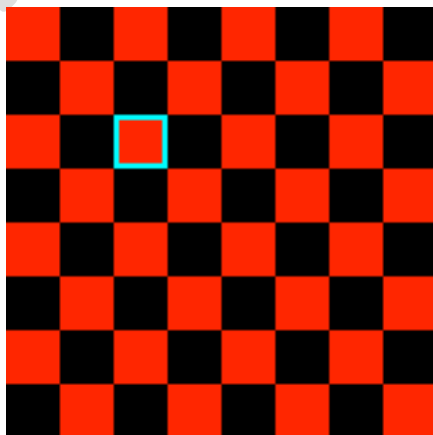
- Sử dụng các sự kiện chuột, bàn phím kết hợp các hàm vẽ tạo ra các hình ảnh trên môi trường Frame.
- Xử lý các sự kiện trên môi trường GUI.

B. Kết quả sau khi hoàn thành:

- Sử dụng được các thành phần thiết kế để có thể tạo ra các giao diện, vẽ các hình ảnh và tô màu cho các hình ảnh đó.
- Xây dựng các ứng dụng có sự kiện chuột và bàn phím.

C. Yêu cầu:

Draw a checkerboard. For this exercise, write a program where the user can select a square by clicking on it. (Use a *JPanel* for the checkerboard.) Highlight the selected square by drawing a colored border around it. When the program starts, no square is selected. When the user clicks on a square that is not currently selected, it becomes selected (and the previously selected square, if any, is unselected). If the user clicks the square that is selected, it becomes unselected. Assume that the size of the panel is exactly 160 by 160 pixels, so that each square on the checkerboard is 20 by 20 pixels. Here is my checkerboard, with the square in row 3, column 3 selected:



The Solution

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

/**
 * This program draws a red-and-black checkerboard.
 * It is assumed that the size of the panel is 160
 * by 160 pixels. When the user clicks a square, that
 * square is selected, unless it is already selected.
 * When the user clicks the selected square, it is
 * unselected. If there is a selected square, it is
 * highlighted with a cyan border.
 */
public class ClickableCheckerboard extends JPanel implements
MouseListener {

    /**
     * A main routine lets this class be run as an application.
     */
    public static void main(String[] args) {
        JFrame window = new JFrame("Clickable Checkerboard");
        ClickableCheckerboard content = new ClickableCheckerboard();
        window.setContentPane(content);
        window.pack(); // Size the window to the preferred size of its
content.
        window.setLocation(100,100);
        window.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        window.setResizable(false); // User can't change the window's
size.
        window.setVisible(true);
    }

    //-----

    int selectedRow; // Row and column of selected square. If no
int selectedCol; // square is selected, selectedRow is -1.

    /**
     * Constructor. Set selectedRow to -1 to indicate that
     * no square is selected. And set the board object
     * to listen for mouse events on itself.
     */
    public ClickableCheckerboard() {
        selectedRow = -1;
        addMouseListener(this);
        setPreferredSize( new Dimension(160,160) );
    }

    /**
     * Draw the checkerboard and highlight selected square, if any.
     * (Note: super.paintComponent(g) is not necessary, since this
     * method already paints the entire surface of the object.
     * This assumes that the object is exactly 160-by-160 pixels.
     */
    public void paintComponent(Graphics g) {
```

```

int row;    // Row number, from 0 to 7
int col;    // Column number, from 0 to 7
int x,y;    // Top-left corner of square

for ( row = 0; row < 8; row++ ) {

    for ( col = 0; col < 8; col++ ) {
        x = col * 20;
        y = row * 20;
        if ( (row % 2) == (col % 2) )
            g.setColor(Color.red);
        else
            g.setColor(Color.black);
        g.fillRect(x, y, 20, 20);
    }

} // end for row

if (selectedRow >= 0) {
    // Since there is a selected square, draw a cyan
    // border around it. (If selectedRow < 0, then
    // no square is selected and no border is drawn.)
    g.setColor(Color.CYAN);
    y = selectedRow * 20;
    x = selectedCol * 20;
    g.drawRect(x, y, 19, 19);
    g.drawRect(x+1, y+1, 17, 17);
}

} // end paint()

/**
 * When the user clicks on the panel, figure out which
 * row and column the click was in and change the
 * selected square accordingly.
 */
public void mousePressed(MouseEvent evt) {

    int col = evt.getX() / 20;    // Column where user clicked.
    int row = evt.getY() / 20;    // Row where user clicked.

    if (selectedRow == row && selectedCol == col) {
        // User clicked on the currently selected square.
        // Turn off the selection by setting selectedRow to -1.
        selectedRow = -1;
    }
    else {
        // Change the selection to the square the user clicked
on.
        selectedRow = row;
        selectedCol = col;
    }
    repaint();

} // end mousePressed()

public void mouseReleased(MouseEvent evt) { }

```

```
public void mouseClicked(MouseEvent evt) { }  
public void mouseEntered(MouseEvent evt) { }  
public void mouseExited(MouseEvent evt) { }  
  
} // end class ClickableCheckerboard
```

D. Kết quả thực hành.

- Sinh viên thực hành ứng dụng trên GUI.
- Thời gian thực hành: 4 tiết.

F. Đánh giá:

- Kiểm tra lại chương trình, thử các kết quả.
- Bắt các lỗi bằng cách sử dụng các phần bắt lỗi: try – catch.

-----Hết-----