Hot and Cold

This is a game to train logical thinking in children.

Rules

- Hidden in a randomly selected square is a red or "hot" square. The child's goal is to find it.
- If the child selects a cell that is touching the "hot" square, the square turns orange.
- If the child selects a cell that is in the row or column of the "hot" square, the square turns yellow.
- The games keeps track of the number of turns that the child has taken. It also resets to another randomly selected hot square.

Step 1: Enter this code

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

public class WhereIsIt extends Applet implements ActionListener
{
    JButton [] a;
    int row = 10;     int col = 10;
    int total = row * col;
    int hiddenX = (int) ((Math.random () * 9) + 1);
    int hiddenY = (int) ((Math.random () * 9) + 1);

    public void init ()
    {
        Panel p = new Panel (new GridLayout (row, col, 0, 0));
        resize (500, 400);

        a = new JButton [total];

        for (int nNum = 0 ; nNum < total ; nNum++)
        {
            a [nNum] = new JButton ("0");
            p.add (a [nNum]);
            a [nNum].addActionListener (this);
            a [nNum].setBackground (Color.white);
            a [nNum].setActionCommand ("" + nNum);
        }
        add (p);

        JButton reset = new JButton ("Reset");
        reset.addActionListener (this);
        reset.setActionCommand ("-1");
        add (reset);
    }

    public void actionPerformed (ActionEvent e)
    {
        int pos = Integer.parseInt (e.getActionCommand ());
        if (pos == -1)
        { //add reset code here
        }
        else
        {
            //find i and j
            int i = pos / row;
            int j = pos % row;

            if (i == hiddenX && j == hiddenY)
            {
                a [pos].setBackground (Color.red);
                showStatus ("You win!");
            }
            else if ((i + 1 == hiddenX || i == hiddenX || i - 1 == hiddenX) && j + 1 ==
                  hiddenY)
            {
                a [pos].setBackground (Color.orange);
            }
            else if ((i + 1 == hiddenX || i - 1 == hiddenX) && j == hiddenY)
            {
                a [pos].setBackground (Color.orange);
            }
            else if ((i + 1 == hiddenX || i == hiddenX || i - 1 == hiddenX) && j - 1 ==
                  hiddenY)
            {
                a [pos].setBackground (Color.orange);
            }
            else if (i == hiddenX || j == hiddenY)
            {
                a [pos].setBackground (Color.yellow);
            }
        }
    }
}
```

Step 2: Make the square turn black if it isn't correct.

- Change the if in the actionPerformed to have an "else" condition that turns the button blue.
- (Do not make it an else if situation).

Step 3: Make the screen look better.

- Add a title "Where is it?" to the top of the applet.
- Change its font.
- Add a picture.

Step 4: Make the reset button work.

- In actionPerformed, there is a space for the reset button's code. (The action command is -1)
- Add a loop that starts at 0 and goes to total. Loop through the button array, set the colour of the background to white.
- Reset the hiddenX and the hiddenY variables. Be careful to not re-declare them. Leave out the int part.