

CS120: Even More GUIs

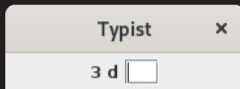
W. Michael Petullo

University of Wisconsin–La Crosse

As of October 6, 2021



Even More GUIs: Typist Specification



Label containing a score s

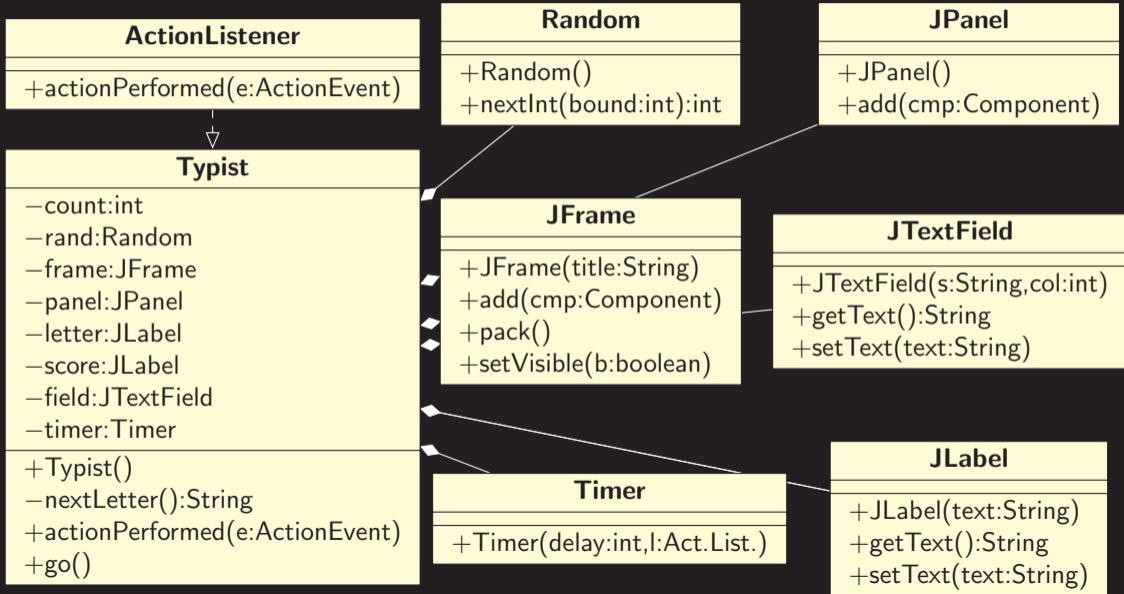
Label containing a letter l

Text-entry field f

- 1 s initially displays '0'.
- 2 l initially displays a random letter.
- 3 f is initially blank, and has a width sufficient to enter two characters.
- 4 Each time the timer expires, the program must compare l and f . If they contain the same value, then the program increments s and updates its display.
- 5 Each time the timer expires, the program must also randomly select another letter before updating l to display it.

User aims to observe each letter displayed by l and type the same letter in f before the timer expires. Properly responding to each letter quickly enough accumulates points.

Even More GUIs: Typist UML



Even More GUIs: Typist Code



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

public class Typist implements ActionListener {
    private Random    rand;
    private int       count;
    private JFrame    frame;
    private JPanel    panel;
    private JLabel    letter;
    private JLabel    score;
    private JTextField field;
    private Timer     timer;

    public Typist() {
        rand    = new Random();
        count   = 0;
        frame   = new JFrame("Typist");
        panel   = new JPanel();
        letter  = new JLabel(nextLetter());
        score   = new JLabel(String.valueOf(0), 2);
        field   = new JTextField("", 2);
        timer   = new Timer(2000, this);
    }

    private String nextLetter() {
        final String alpha = "abcdefghijklmnopqrstuvwxyz";
        String next = rand.nextInt(alpha.length());
        return String.valueOf(alpha.charAt(next));
    }
}
```

```
    public void actionPerformed(ActionEvent e) {
        String t1 = field.getText();
        String t2 = letter.getText();
        if (t1.equals(t2)) {
            count++;
        }

        score.setText(String.valueOf(count));
        letter.setText(nextLetter());
        field.setText("");
    }

    public void go() {
        panel.add(score);
        panel.add(letter);
        panel.add(field);
        frame.add(panel);

        frame.pack();
        frame.setVisible(true);

        timer.start();
    }

    public static void main(String[] args) {
        Typist t = new Typist();
        t.go();
    }
}
```



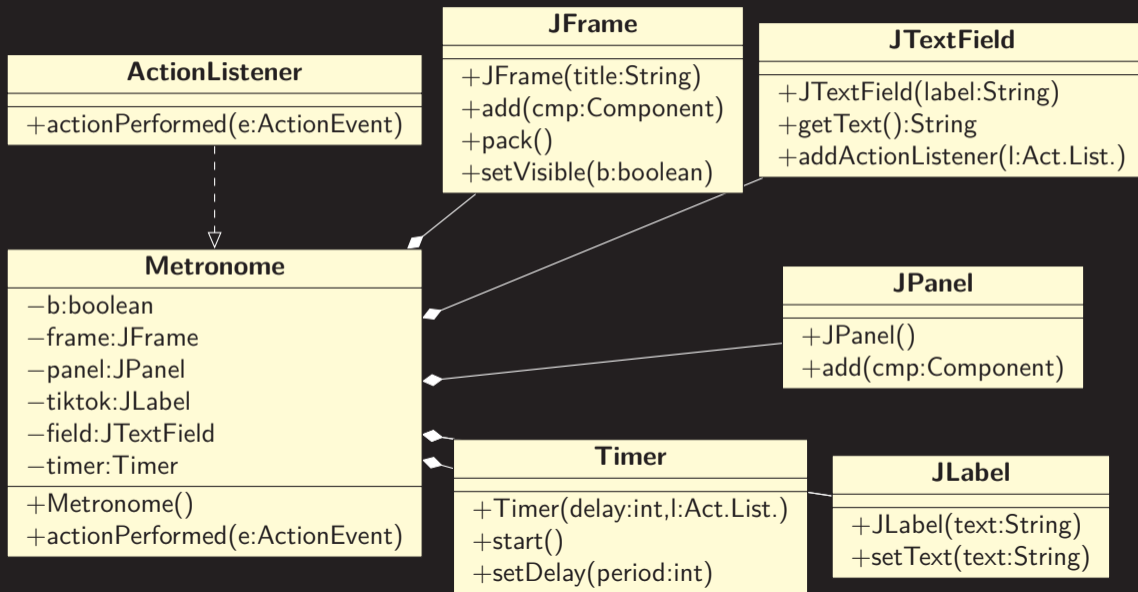
Even More GUIs: Metronome Specification

Label / ("TIK" or "tok")	p
--------------------------	-----



- 1 / initially displays "TIK".
- 2 p initially contains 1000.
- 3 Timer causes "TIK" and "tok" to cycle every p milliseconds.
- 4 Changing p changes period of timer.

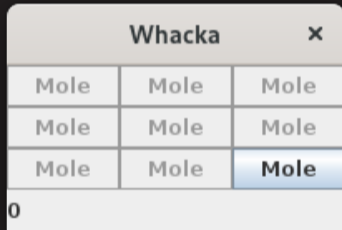
Even More GUIs: Metronome UML





Demo Metronome

Even More GUIs: Whacka Specification

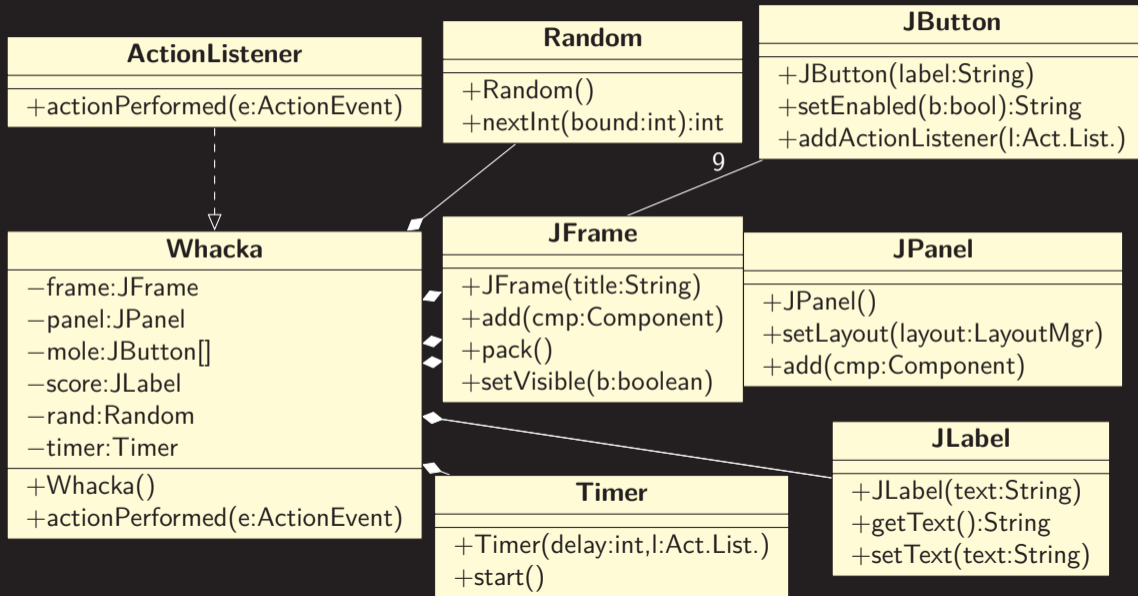


m_1	m_2	m_3
m_4	m_5	m_6
m_7	m_8	m_9
Label containing a score l		

- 1 l initially displays '0'.
- 2 A timer expires every second.
- 3 Each time the timer expires, the program selects and makes active a random mole m_n .
- 4 Each time the user presses an active mole the value displayed by l increments.

User aims to accumulate points by whacking moles.

Even More GUIs: Whacka UML





Demo Whacka



Demo demo.html



Even More GUIs: Assignments

Graded Homework Aquinas: whacka in Java

Ungraded Labs Aquinas: metronome and recursion in Java

<https://www.flyn.org/courses/cs120-2021-fall/schedule>