FUNDAMENTALS

CS062

DATA STRUCTURES AND ADVANCED PROGRAMMING

4: Java GUIs and Graphics

Alexandra Papoutsaki
LECTURES

Mark Kampe
LABS

Lecture 4: Java GUIs and Graphics

- Java GUIs
- Graphics
- Events

Some slides adopted from Oracle tutorials

Inheritance

- AWT: The Abstract Windowing Toolkit is found in the package `java.awt`
  - Heavyweight components.
  - Implemented with native code written for that particular computer.
  - The AWT library was written in six weeks!
- Swing: Java 1.2 extended AWT with the `javax.swing` package.
  - Lightweight components.
  - Written in Java.

JFrame

- `javax.swing.JFrame` inherits from `java.awt.Frame`
- The outermost container in an application.
- To display a window in Java:
  - Create a class that extends `JFrame`.
  - Set the size.
  - Set the location.
  - Set it visible.

JFrame

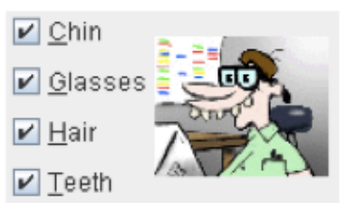
```
import javax.swing.JFrame;

public class MyFirstGUI extends JFrame {

    public MyFirstGUI() {
        super("First Frame");
        setSize(500, 300);
        setLocation(100, 100);
        setVisible(true);
    }

    public static void main(String[] args) {
        MyFirstGUI mfgui = new MyFirstGUI();
    }

}
```

Closing a GUI

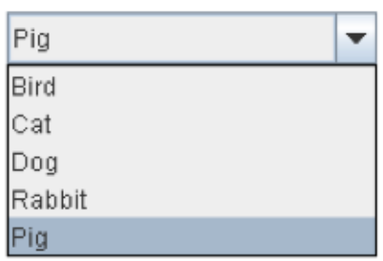
- The default operation of the quit button is to set the visibility to false. The program does not terminate!
- `setDefaultCloseOperation` can be used to control this behavior.
- `mfgui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
- More options (hide, do nothing, etc).

Basic components

JButton

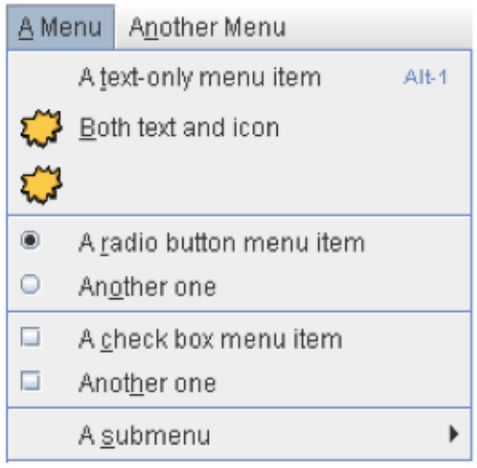


JCheckBox

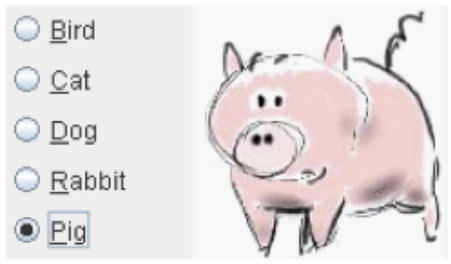


JComboBox

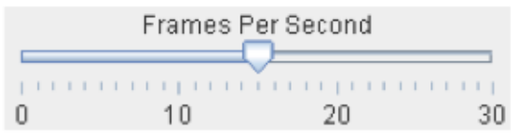
JList



JMenu



JRadioButton



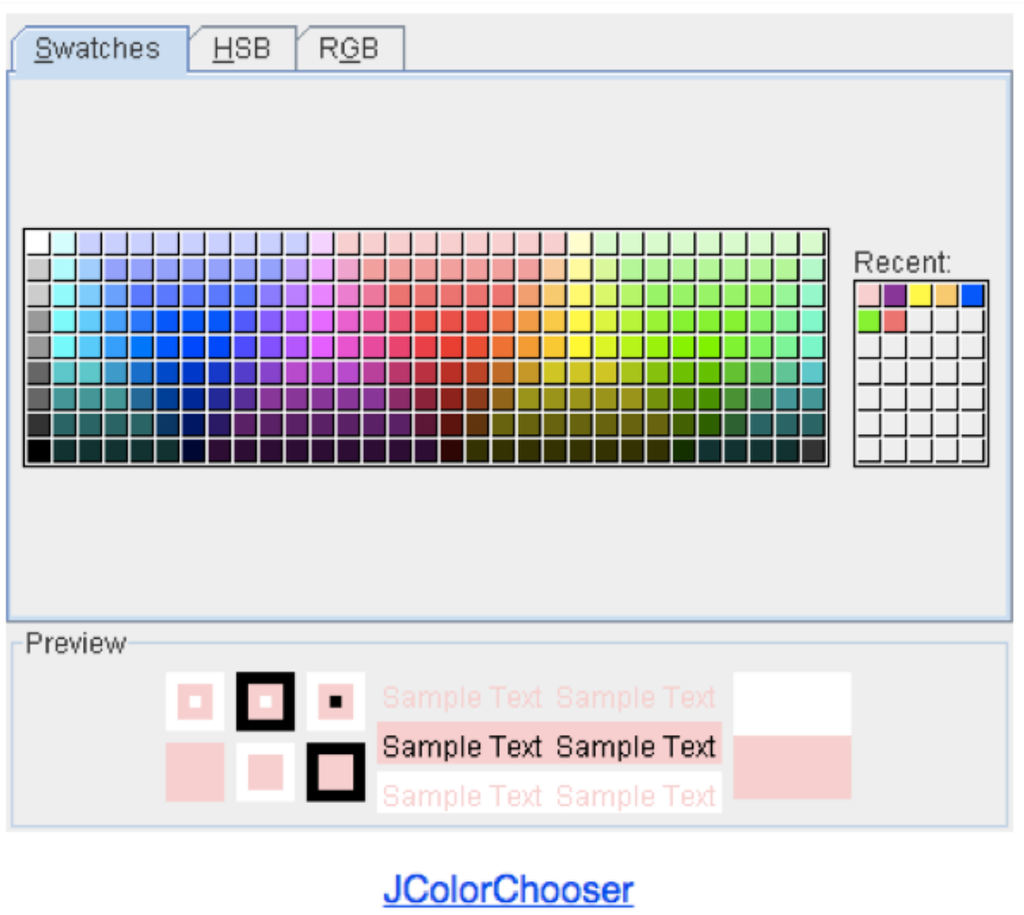
JSlider

JSpinner

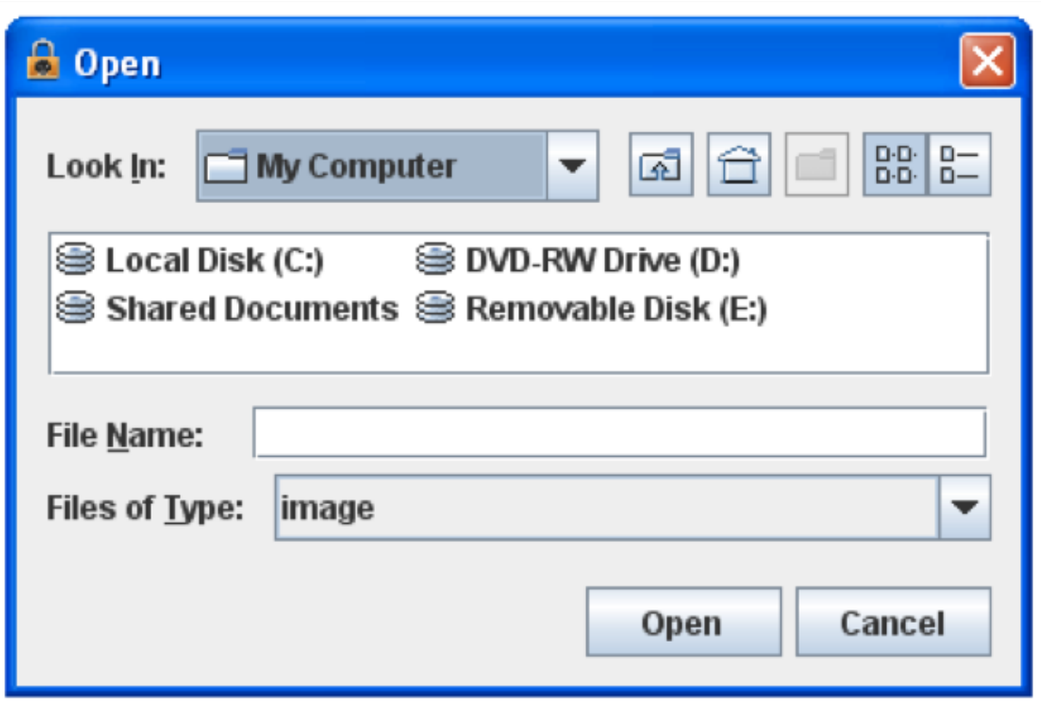
JTextField

JPasswordField

Interactive displays



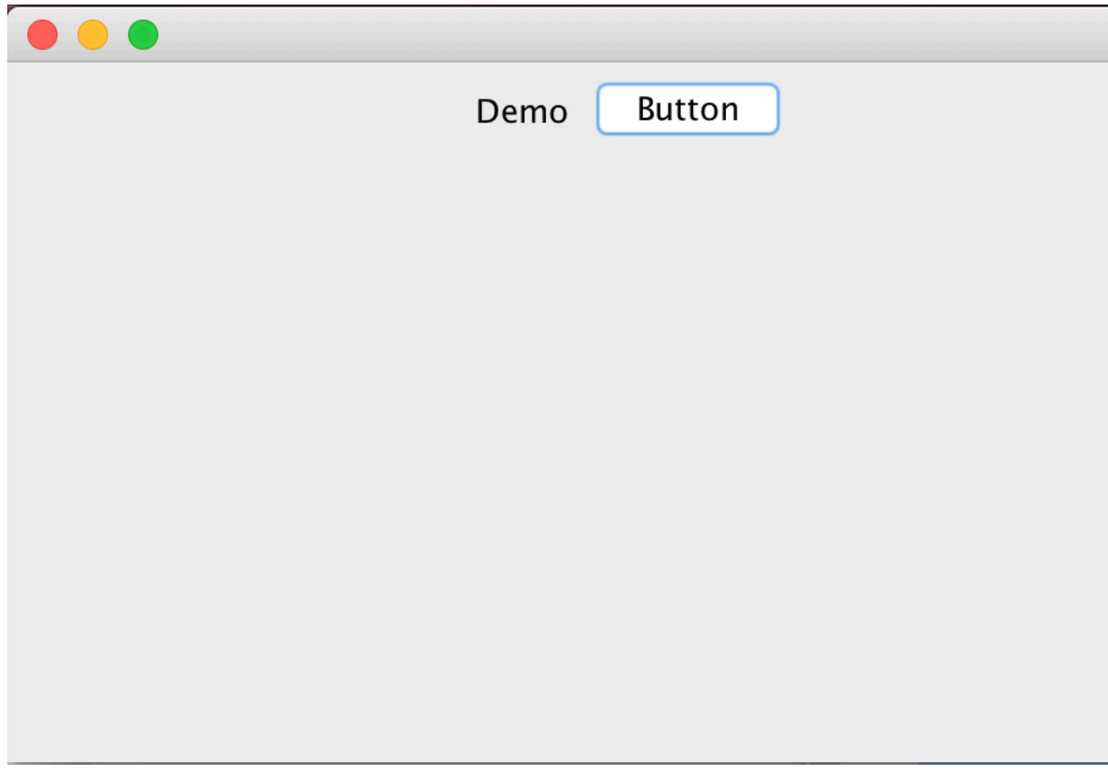
JColorChooser



JFileChooser

Adding JComponents to JFrame

```
import java.awt.Container;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class GUIDemo extends JFrame {
    public GUIDemo() {
        // Container cp = getContentPane();
        // cp.setLayout(new FlowLayout());
        // cp.add(new JLabel("Demo"));
        // cp.add(new JButton("Button"));
        JPanel mainPanel = new JPanel(new FlowLayout());
        mainPanel.add(new JLabel("Demo"));
        mainPanel.add(new JButton("Button"));
        setContentPane(mainPanel);
        setSize(500, 300);
        setVisible(true);
    }

    public static void main(String[] args) {
        GUIDemo gd = new GUIDemo();
        gd.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



Lecture 4: Java GUIs and Graphics

- Java GUIs
- Graphics
- Events

Java Graphics

- Create arbitrary objects you want to draw:
  - `Rectangle2D.Double`, `Line.Double`, etc.
  - Constructors take x, y coordinates and dimensions, but don't actually draw items.
- All drawing takes place in `paint` method using a "graphics content".
- Triggered implicitly by uncovering window or explicitly by calling the `repaint` method.
  - Adds repaint event to draw queue and eventually draws it.

Graphics context

- All drawing is done in `paint` method of component.
- `public void paint (Graphics g)`
- `g` is a graphics context provided by the system.
- "pen" that does the drawing.
- You call `repaint()` not `paint()`.
- Need to import classes from `java.awt.*`, `java.geom.*`, `javax.swing.*`
- See `MyGraphicsDemo`.

General graphics applications

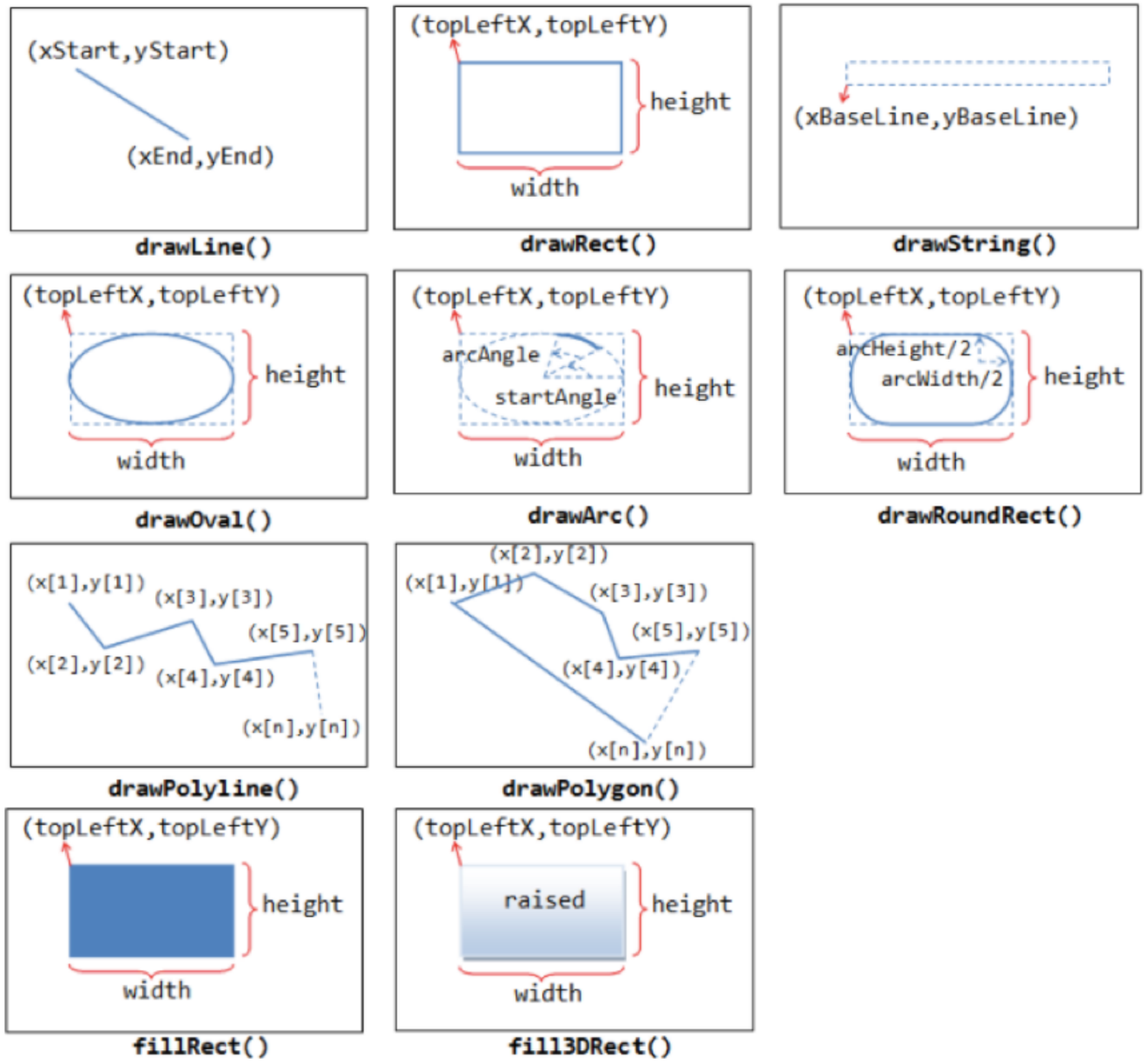
- Create an extension of component (`JPanel` or `JFrame`) and implement `paint` method in subclass.
- At start of `paint()` method cast `g` to `Graphics2D`.
- Call `repaint()` every time you want the component to be redrawn.

Geometric objects

- Objects from classes `Rectangle2D.Double`, `Line2D.Double`, etc. from `java.awt.geom`
- Constructors take parameters x, y, width, height but don't draw object.
- `Rectangle2D.Double`
- `Ellipse2D.Double`
- `Arc2D.Double`
- etc.

Drawing

- `myObj.setFrame(x, y, width, height)`: moves and sets size of component
- `g2.draw(myObj)`: gives outline
- `g2.fill(myObj)`: gives filled version
- `g2.drawString("a string", x, y)`: draws string



java.awt.Color

Lecture 4: Java GUIs and Graphics

- Java GUIs
- Graphics
- Events

Action listeners

- Define what should be done when a user performs certain operations.
  - e.g., clicks a button, chooses a menu item, presses Enter, etc.
- The application should implement the ActionListener interface.
- An instance of the application should be registered as a listener on one or more components.
- Implement the `actionPerformed` method.

```
public class MultiButtonApp implements ActionListener {
    ...
        //where initialization occurs:
        button1.addActionListener(this);
        button2.addActionListener(this);

    ...
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == button1){
            //do something
        }
    }
}
```

https://docs.oracle.com/javase/tutorial/uiswing/events/actionlistener.html

Mouse listeners

- Define what should be done when a user enters a component, presses or releases one of the mouse buttons.
- The application should implement the MouseListener interface
  - Implement methods `mousePressed`, `mouseReleased`, mouseEntered, mouseExited, and `mouseClicked`.
- **Or** extend the MouseAdapter class
  - Which has default implementations of all of them.

```
public class MouseEventDemo ... implements MouseListener {
        //where initialization occurs:
        //Register for mouse events on blankArea and the panel.
        blankArea.addMouseListener(this);
        addMouseListener(this);
    ...

    public void mousePressed(MouseEvent e) {
       saySomething("Mouse pressed; # of clicks: "
                    + e.getClickCount(), e);
    }
```

https://docs.oracle.com/javase/tutorial/uiswing/events/mouselistener.html

Lecture 4: Java GUIs and Graphics

- Java GUIs
- Graphics
- Events

Readings:

- Java Graphics: https://github.com/pomonacs622019fa/Handouts/blob/master/graphics.md
- Programming with GUIs: http://www.cs.pomona.edu/classes/cs062/handouts/JavaGUI.pdf
- Swing/GUI Cheat Sheet: https://github.com/pomonacs622019fa/Handouts/blob/master/swing.md
- Writing Event Listeners: https://docs.oracle.com/javase/tutorial/uiswing/events/index.html