12. Java Swing
Individual Project 2

Minhaeng Lee
Java Swing?

- Java GUI building library
- Not very famous for standalone application
  - Slow (because of JVM)
  - Current trends move to web
- Still useful and worth to try
- Logics are same
- http://docs.oracle.com/javase/tutorial/uiswing/
Necessary Components

• JK, JW
  – Drawing

• JM
  – CoverFlow

• Aiden
  – Miglayout

• Sien
  – Flipping animation?
Why others’ source?

• Difficult and inefficient to make from scratch
Basic JComponents

• JFrame
  – Most Basic Component to make window
  – One per each window
  – Window as is

• JPanel
  – Common component inside of a window
  – Multiple in each window

• JButton, JLabel, JTextArea ... etc.
Components
Your first Swing Example

Run `HelloWorldSwing.java` in ExampleSwing project

Does your own look like this?

```java
import javax.swing.*;

public class HelloWorldSwing {
    /**
     * Create the GUI and show it. For thread safety,
     * this method should be invoked from the
     * event-dispatching thread.
     */
    private static void createAndShowGUI() {
        // Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

        // Display the window.
        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        // Schedule a job for the event-dispatching thread:
        // creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}
```
File Chooser

Run **FileChooserDemo.java**
in ExampleSwing project

```java
int returnVal = fc.showOpenDialog(FileChooserDemo.this);
if (returnVal == JOptionPane.APPROVE_OPTION) {
    File file = fc.getSelectedFile();
    //This is where a real application would open the file.
    log.append("Opening: " + file.getName() + "." + newline);
} else {
    log.append("Open command cancelled by user." + newline);
}
log.setCaretPosition(log.getDocument().getLength());
```
Run *BorderDemo.java* in ExampleSwing project
Text, Password, ....more and more

Run `TextDemo.java` in ExampleSwing project

Run `PasswordDemo.java` in ExampleSwing project

What do you need?
Check here great examples:  [http://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html](http://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html)
LookAndFeel

- Skin for java GUI
- Easy to apply
- A number of resources: ex) http://www.jyloo.com/synthetica/themes/

```java
try {
    UIManager.setLookAndFeel(new SubstanceBusinessBlueSteelLookAndFeel());
} catch (final UnsupportedLookAndFeelException e1) {
}
```
Source management tools

• Famous tools
  – CVS
  – SubVersion
  – GIT
  – Etc.

• Necessary for source management
  – Version control
  – Cooperation
  – Useful to show your work to public!
Using External Library for Eclipse

• Adding Library path to build path
• When library import problem happened!
Using External Library for Eclipse (1/3)

Right click on the project -> Properties

Click “Add JARs”
Using External Library for Eclipse (2/3)

Select Required “jar”s
Then click ok

Check the jar list and then Click ok
Successfully add and error fixed!
How can I get required library?

- Mostly on the web
- Use Google
- Read publisher’s document carefully
- Search using package name

In this case
- Keyword: “org. pushingpixels.substance jar”

```
import org.pushingpixels.substance.api.skin.SubstanceB
```

Download substance-6.1.jar : substance « s « Jar File ...
java2s.com/Code/Jars/Downloadsubstance61jar.htm
Files contained in substance-6.1.jar: Looks.license META-INF/MANIFEST. ...
SubstanceLookAndFeel.class org.pushingpixels.substance.api.
Individual Project : Tetris

• **Basic**
  – Get Block (and other) images
  – Generation 2D Map
  – Key input
  – **Thread based Timing Control**
    • Frame control
  – Game, Score, Next zone

• **Advanced**
  – Continue from previous
  – Item
  – Auto Play
Individual Project : Sneak Game

• **Basic**
  – Get Sneak Parts images
  – Generate 2D Map
  – Key input
  – **Thread based Time Control**
    • Frame control
  – Game, Score zone

• **Advanced**
  – Continue from previous
  – Item
  – Auto Play
Individual Project : Music Player

• **Basic**
  – Get required button images (play, stop ... etc.)
  – **CoverFlow (template provided)**
    • ExampleGUI – CoverFlowDemo.java
  – File Scanning
  – File load/save
  – Play List Management
  – Music Play/Stop

• **Advanced**
  – Music Equalize
Individual Project: Calendar

• Basic
  – Date control
  – JComponents
  – Layout using MigLayout (Example Provided)
    • ExampleGUI – DashboardDemo.java, SwingDemo.java
  – Event Add/Delete/Edit

• Advanced
  – Sync to the web
Individual Project : Flash cards

• Basic
  – Layout
  – File management (read/write)

• Advanced
  – Fancy GUI
  – Online data management
Homework

• **Record Your Progress during week**
  – What Problem you have
  – What have you done
  – No progress, nothing to learn

• **Read Layout Source**
  – ExampleGUI – DashboardDemo.java
Java Swing Menu
Java Swing Menu

Read MenuDemo.java
Button and event

Read ButtonDemo.java
Layouts

-FlowLayout
-BorderLayout
-AbsoluteLayout...
References

• http://docs.oracle.com/javase/tutorial/uiswing/