Graphical User Interfaces
GUIs
The Plan

- Components
- Flat Layouts
- Hierarchical Layouts
- Designing a GUI
- Coding a GUI
Components

- JLabel
text/image display
- JTextField
single line for text input/output
- JTextArea
multiple lines for text input/output
- JButton
used for decisions
- JFrame
a basic window
Components

- JLabel
  text/image display
- JTextField
  single line for text input/output
- JTextArea
  multiple lines for text input/output
- JButton
  used for decisions
- JFrame
  a basic window
Components

- JLabel: text/image display
- JTextField: single line for text input/output
- JTextArea: multiple lines for text input/output
- JButton: used for decisions
- JFrame: a basic window
Flat Layouts

GridLayout

BorderLayout

NORTH

CENTER

WEST

SOUTH

EAST
Flat Layouts

GridLayout
• Added left to right, top to bottom
• Expands to fill horizontally and vertically
• Each space equal width and height

BorderLayout
• Not all positions must be filled
• CENTER expands horizontally and vertically
• NORTH and SOUTH expand horizontally
• WEST and EAST expand vertically
Flat Layouts

BorderLayout

Above is a JLabel in the BorderLayout NORTH

JButton in BorderLayout.WEST

TextArea Here in BorderLayout.CENTER

JButton in BorderLayout.EAST

Below is a JTextField in BorderLayout.SOUTH
Flat Layouts

GridLayout
Hierarchical Layouts

You can put layouts within layouts:
Hierarchical Layouts

Identify the BorderLayout and GridLayouts in the application on the right.
Hierarchical Layouts

CENTER

EAST
Hierarchical Layouts

GridLayout
Hierarchical Layouts

GridLayout
Hierarchical Layouts
Hierarchical Layouts

CENTER

Jeff: Jenny, I like your star, how did you do that?
Jenny: I just used the polygon shape. Just click on 5 points and you should
Jeff: be able to do it
Jenny: Yeah, that’s not too bad. I like the zig-zags too.
Kathy: I’ll bet you guys can’t get a polygon like this one.
Kathy: Jeff, draw a square on the Community board and I’ll do one of those
Kathy: cool polygons again.
Jeff: Okay, but you’ve got to show us how to do it.
Chris: I prefer the plain stuff like ovals, rectangles, and triangles.
Jenny: That’s just because you can’t do the cool polygons stuff.
Kathy: Try a polygon Chris, here I’ll put a circle on the Community board.
Chris: How about that! I can too do the polygons. Let’s see you do one Jenny.
Jenny: I’ll do a crown.

SOUTH
Hierarchical Layouts

Jeff: Jenny, I like your star; how did you do that?
Jenny: I just used the polygon shape. Just click on 5 points and you should Jenny: be able to do it.
Jeff: Yeah, that's not too bad. I like the zig-zags too.
Kathy: I'll bet you guys can't get a polygon like this one.
Kathy: Jeff, draw a square on the Community board and I'll do one of those Kathy: cool polygons again.
Jeff: Okay, but you've got to show us how to do it.
Chris: I prefer the plain stuff like ovals, rectangles, and triangles.
Jenny: That's just because you can't do the cool polygons stuff.
Kathy: Try a polygon Chris, here I'll put a circle on the Community board.
Chris: How about that! I can too do the polygons, let's see you do one Jenny.
Jenny: I'll do a crown.
Hierarchical Layouts

• Virtually every layout I make is a hierarchy of GridLayout and BorderLayout

• Other Layouts include
  – BoxLayout
  – GridBagLayout
  – FlowLayout
  – CardLayout
Designing a GUI

• What components are needed?
• Which components are of primary importance? Secondary?
• How do the components relate to each other?
• How big are the components?
• How can they be arranged into BorderLayout and GridLayout?
Coding a GUI

1. Declare the components as instance variables
2. Write a makeComponents method to initialize the components
3. Write a layoutComponents methods to arrange the components
4. Write a constructor to call the above two methods
5. Write a setVisible method to set the primary component’s visibility (usually a JFrame).
Examples

- BorderExample.java (today)
- GridExample.java (in the code directory)
- CombinedExample.java (in code directory)
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BorderExample extends JApplet {
    JFrame frame;
    JTextArea middle;
    JTextField bottom;
    JButton left, right;
    JLabel title;
}
private void makeComponents()
{
    frame=new JFrame("BorderExample");
    middle=new JTextArea(10, 40);
    bottom=new JTextField();
    left=new JButton("left");
    right=new JButton("right");
    title=new JLabel("Title");
}
private void makeLayout()
{
    Container container = frame.getContentPane();
    container.setLayout(new BorderLayout());
    container.add(new JScrollPane(middle), BorderLayout.CENTER);
    container.add(title, BorderLayout.NORTH);
    container.add(left, BorderLayout.WEST);
    container.add(right, BorderLayout.EAST);
    container.add(bottom, BorderLayout.SOUTH);
    frame.pack();
}
public BorderExample()
{
    makeComponents();
    makeLayout();
}

public void setVisible(boolean vis)
{
    frame.setVisible(vis);
}
public void init()
{
    main(null);
}

public static void main(String[] args)
{
    BorderExample example=new BorderExample();
    example.setVisible(true);
}
Event Handling
The Plan

- Sequential (Single Thread) Model
- Event Model
- Making the GUI interactive
- Examples
- Practice
Sequential (Single Thread) Model
Event Model

AWT Event Loop

Program Thread
Event Model

- main is called
- GUI is constructed
- Thread terminates
Event Model

- Check for input
- Find event listeners
- Notify listeners
Making the GUI Interactive

1) import java.awt.event.*
2) implements ActionListener
3) write method
   public void actionPerformed(ActionEvent e)
4) call addActionListener(this) to all JButtons
Examples

**AdderGUI.java**

```java
AdderGUI.java
```

**GameShell.java**

```java
GameShell.java
```

![Adder GUI example](image1)

![GameShell example](image2)
Examples

AdderGUI.java

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

public class AdderGUI
    extends JApplet
    implements ActionListener
```
public void actionPerformed(ActionEvent ae) {
    String addend0Text=addend0.getText();
    double addend0Number=Double.parseDouble(addend0Text);
    String addend1Text=addend1.getText();
    double addend1Number=Double.parseDouble(addend1Text);
    double answer=addend0Number+addend1Number;
    sum.setText(""+answer);
}
Examples

private void makeComponents()
{
    frame=new JFrame("Game Shell");
    addend0=new JTextField(8);
    addend1=new JTextField(8);
    sum=new JTextField(8);
    compute=new JButton("=");
    compute.addActionListener(this);
    plus=new JLabel("+");
    plus.setHorizontalAlignment(SwingConstants.CENTER);
    plus.setHorizontalAlignment(SwingConstants.CENTER);
}

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

public class GameShell
    extends JApplet
    implements ActionListener
public void actionPerformed(ActionEvent ae) {
    Object cause=ae.getSource();
    if(cause==pause) {
        if(pause.getText().equals("Pause")) {
            pause.setText("Resume");
            shell.setText("Paused");
        } else {
            pause.setText("Pause");
            shell.setText("Game Running");
        }
    }
    if(cause==reset) {
        pause.setText("Start");
        shell.setText("Splash");
    }
}
GameShell.java

```java
pause = new JButton("Start");
pause.addActionListener(this);
reset = new JButton("Start New Game");
reset.addActionListener(this);
```
Practice

• Make a 2x2 tic-tac-toe board out of initially blank JButtons.
• Make the JButton text change to X when the user clicks on it.
• Make the JButton text change to X and O alternatively as the user clicks on the buttons.
  Hint: use a boolean instance variable.
• Make the fonts larger, and maybe add images.