

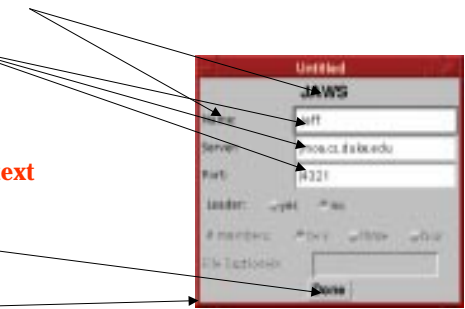
Graphical User Interfaces -- GUIs

❖ The Plan

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

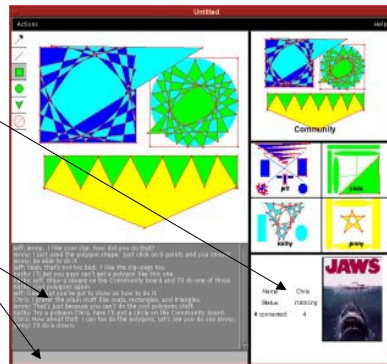
Components

- ❖ **JLabel**
text/image display
- ❖ **TextField**
single line for text input/output
- ❖ **TextArea**
multiple lines for text input/output
- ❖ **Button**
used for decisions
- ❖ **Frame**
a basic window



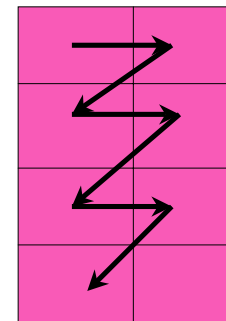
Components

- ❖ **JLabel**
text/image display
- ❖ **TextField**
single line for text input/output
- ❖ **TextArea**
multiple lines for text input/output
- ❖ **Button**
used for decisions
- ❖ **Frame**
a basic window

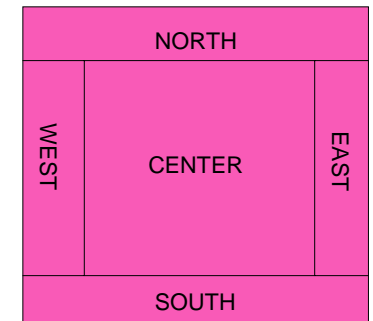


Flat Layouts

GridLayout



BorderLayout



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



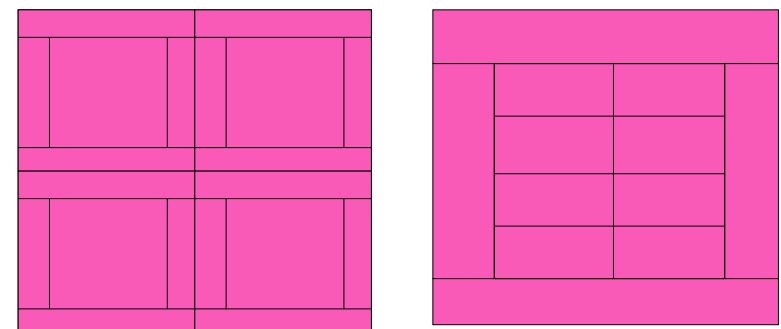
Flat Layouts

GridLayout



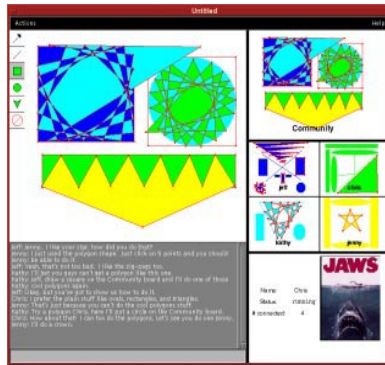
Hierarchical Layouts

You can put layouts within layouts:



Hierarchical Layouts

Identify the `BorderLayout` and `GridLayout` in the application on the right.



Hierarchical Layouts

- ❖ Virtually every layout we 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)

❖ In code directory (`GUIs.jar`)

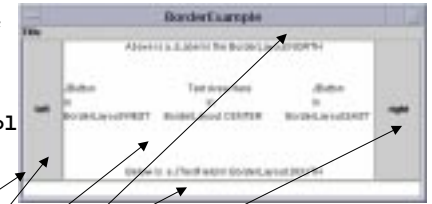
□ `GridExample.java`

□ `CombinedExample.java`

BorderExample.java

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

```
public class BorderExampl  
    extends JApplet  
{  
    JFrame frame;  
    JTextArea middle;  
    JTextField bottom;  
    JButton left, right;  
    JLabel title;
```



BorderExample.java



```
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");  
}
```

BorderExample.java

```
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();  
}
```

BorderExample.java

```
public BorderExample()
{
    makeComponents();
    makeLayout();
}

public void setVisible(boolean vis)
{
    frame.setVisible(vis);
}
```

BorderExample.java

```
public void init()
{
    main(null);
}

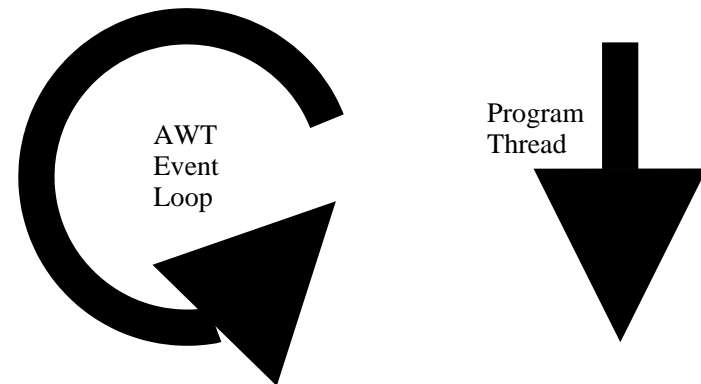
public static void main(String[] args)
{
    BorderExample example = new BorderExample();
    example.setVisible(true);
}
```

Event Handling

❖ The Plan

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

Event Model

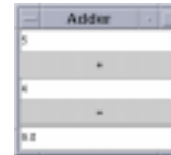


Making the GUI Interactive

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

Examples

AdderGUI.java



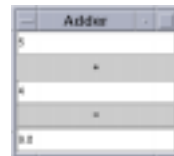
GameShell.java



Examples

AdderGUI.java

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

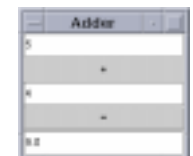


```
public class AdderGUI extends JApplet  
    implements ActionListener
```

Examples

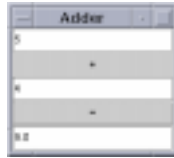
AdderGUI.java

```
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

AdderGUI.java



```
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);
}
```

CompSci 6

41.25

Examples

GameShell.java



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

public class GameShell extends JApplet
    implements ActionListener
```

CompSci 6

41.26

Examples

GameShell.java



```
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");
    }
}
```

CompSci 6

41.27

Examples

GameShell.java



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

CompSci 6

41.28