

Graphical User Interface using Swing

Part II - Layout Managers

Samir Genaim

Let Swing Organise your Widgets

- ◆ We used **Absolute Positioning** to place components:

```
JPanel mainPanel = new JPanel();
mainPanel.setLayout(null);
```

```
JPanel redPanel = new JPanel();
redPanel.setBackground(Color.red);
redPanel.setLocation(10, 10);
redPanel.setSize(50,50);
mainPanel.add(redPanel);
```

- ◆ Absolute positioning become difficult to manage when the GUI is complex, the frame resized, etc.
- ◆ Swing can do this for you, using different layout in Panels, Frames, etc.

Layout Managers in Swing

- ◆ Layout Managers available in Swing
 1. Flow Layout (default for JPanel)
 2. Border Layout (default for JFrame, ...)
 3. Box Layout
 4. Card Layout
 5. Grid Layout
 6. Grid Bag Layout
 7. Group Layout
 8. Spring Layout
- ◆ It is not possible to use Absolute Positioning and a Layout Manager at the same time ...

FlowLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new FlowLayout( FlowLayout.LEFT, 10, 20 ) );
```

```
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 50) );  
    mainPanel.add( createPanel(Color.green, 50, 50) );  
    mainPanel.add( createPanel(Color.orange, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 50, 50) );
```

```
    mainPanel.setOpaque(true);
```

```
    ...
```

```
}
```

```
private JPanel createPanel(Color color, int x, int y) {
```

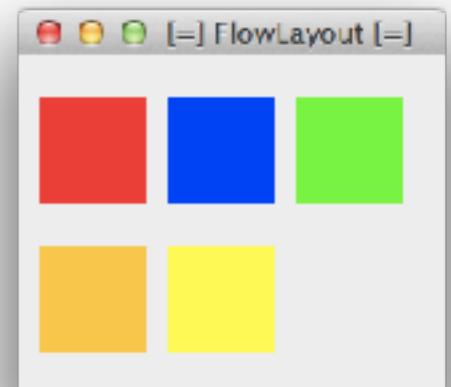
```
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;
```

```
}
```

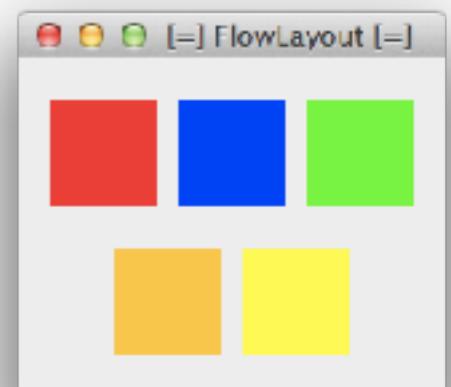
Components are positioned left-right-top-down,
in the order in which they were added.

see: examples.swing.miscFlowLayout

FlowLayout.LEFT



FlowLayout.CENTER



FlowLayout.RIGHT



FlowLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new FlowLayout( FlowLayout.LEFT, 10, 20 ) );
```

```
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 50) );  
    mainPanel.add( createPanel(Color.green, 50, 50) );  
    mainPanel.add( createPanel(Color.orange, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 50, 50) );
```

```
    mainPanel.setOpaque(true);
```

```
    ...
```

```
}
```

```
private JPanel createPanel(Color color, int x, int y) {
```

```
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;
```

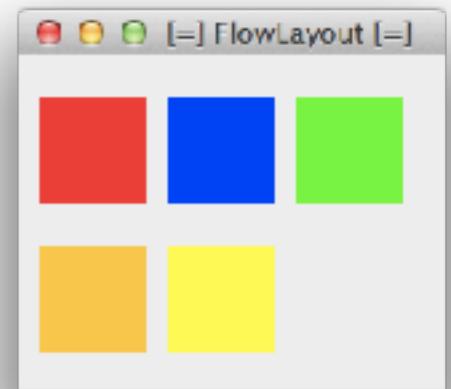
```
}
```

Reordered when the size
of the panel changes.

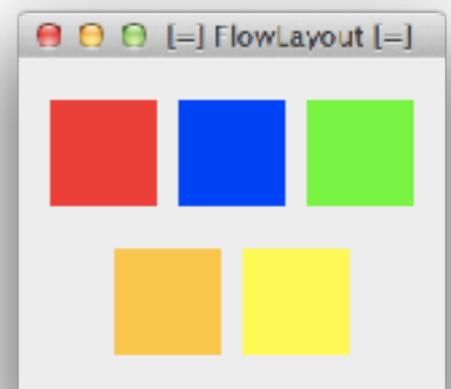
Components are positioned left-right-top-down,
in the order in which they were added.

see: examples.swing.miscFlowLayout

FlowLayout.LEFT



FlowLayout.CENTER



FlowLayout.RIGHT



FlowLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new FlowLayout( FlowLayout.LEFT, 10, 20 ) );
```

```
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 50) );  
    mainPanel.add( createPanel(Color.green, 50, 50) );  
    mainPanel.add( createPanel(Color.orange, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 50, 50) );
```

```
    mainPanel.setOpaque(true); "new JPanel()" uses FlowLayout()  
    ... which uses FlowLayout.CENTER  
}
```

```
private JPanel createPanel(Color color, int x, int y) {
```

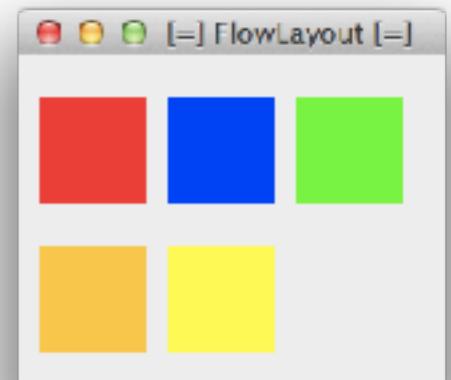
```
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;
```

Reordered when the size
of the panel changes.

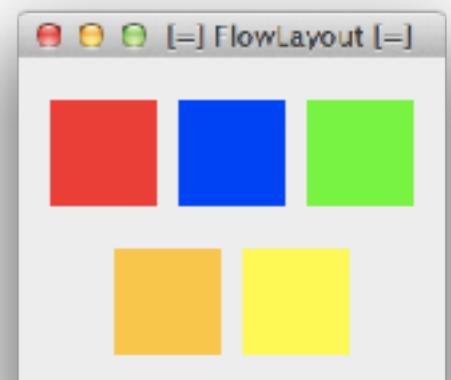
Components are positioned left-right-top-down,
in the order in which they were added.

see: examples.swing.miscFlowLayout

FlowLayout.LEFT



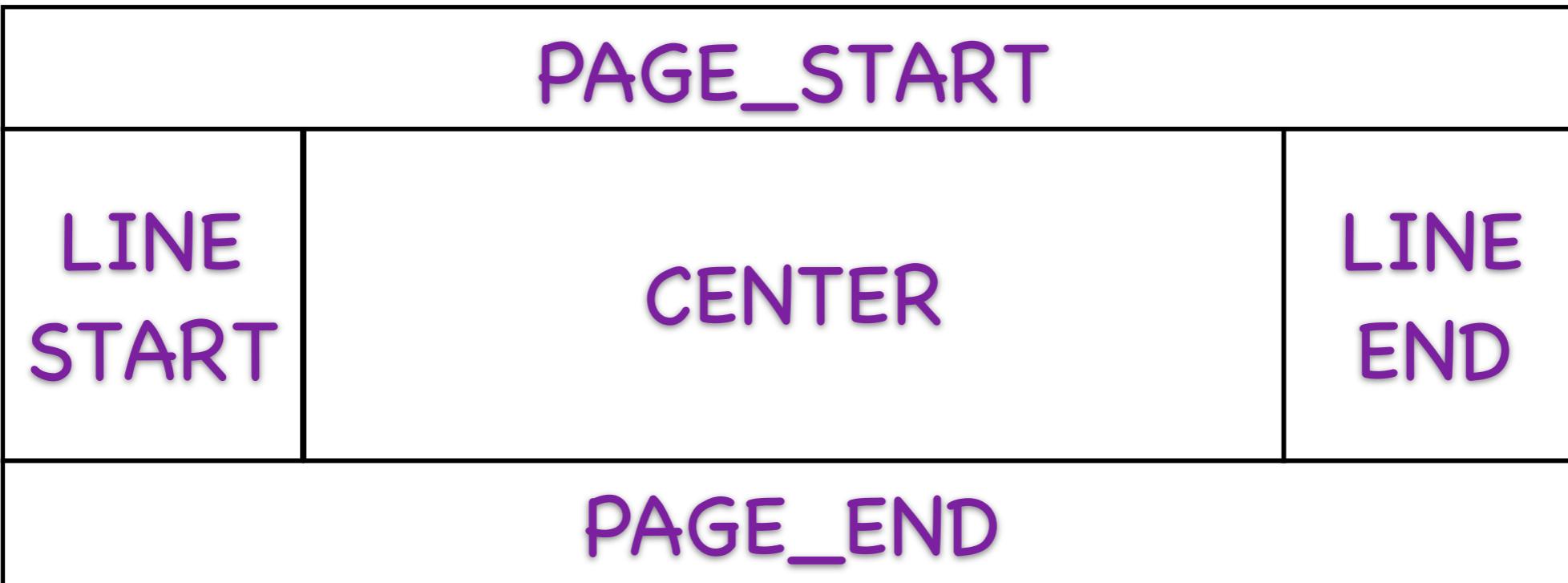
FlowLayout.CENTER



FlowLayout.RIGHT



BorderLayout

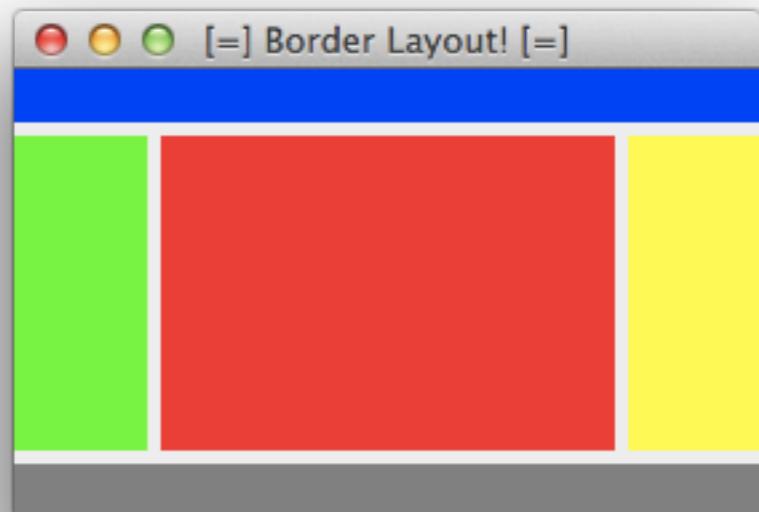


- ◆ The panel is divided into 5 areas, and in each one we can put **one** component (though, we don't have to)
- ◆ When resizing the Panel:
 1. All parts are resized to "best fit" their widgets
 2. PAGE_START/END do not change vertical size
 3. LINE_START/END do not change horizontal size
 4. When an area is resized, it asks the component placed in it to set its size accordingly

Border Layout - HowTo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

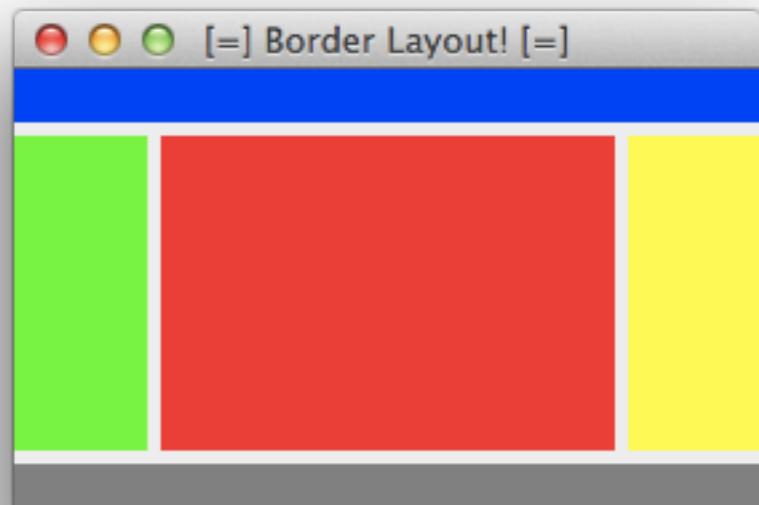
see: examples.swing.misc.BorderLayoutExample



Border Layout - HowTo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

the parameters are the horizontal and vertical gap between the areas



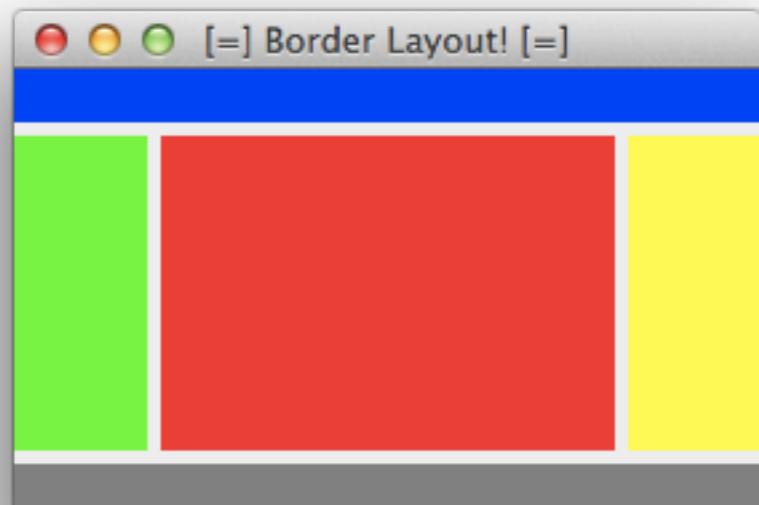
see: examples.swing.misc.BorderLayoutExample

Border Layout - HowTo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

the parameters are the horizontal and vertical gap between the areas

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```



see: examples.swing.misc.BorderLayoutExample

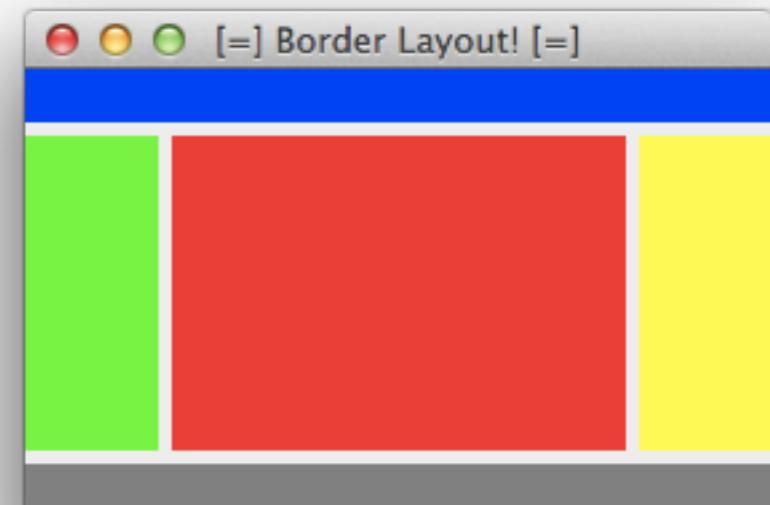
Border Layout - HowTo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

the parameters are the horizontal and vertical gap between the areas

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```



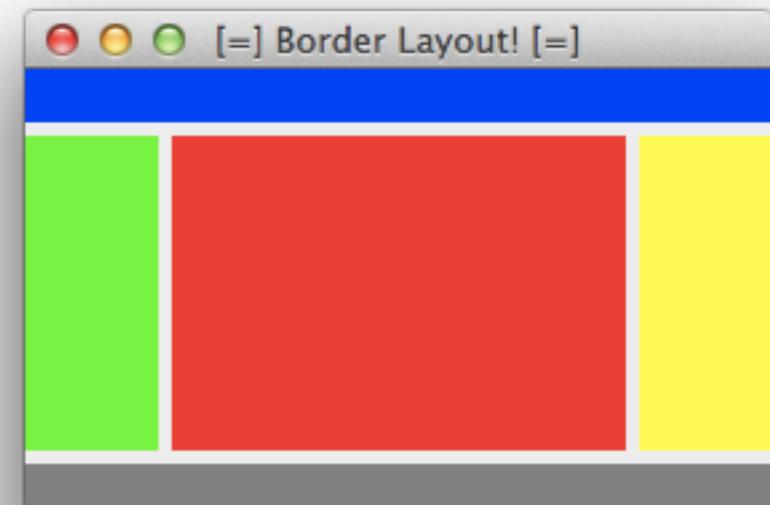
Border Layout - HowTo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

the parameters are the horizontal and vertical gap between the areas

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```



Border Layout - HowTo

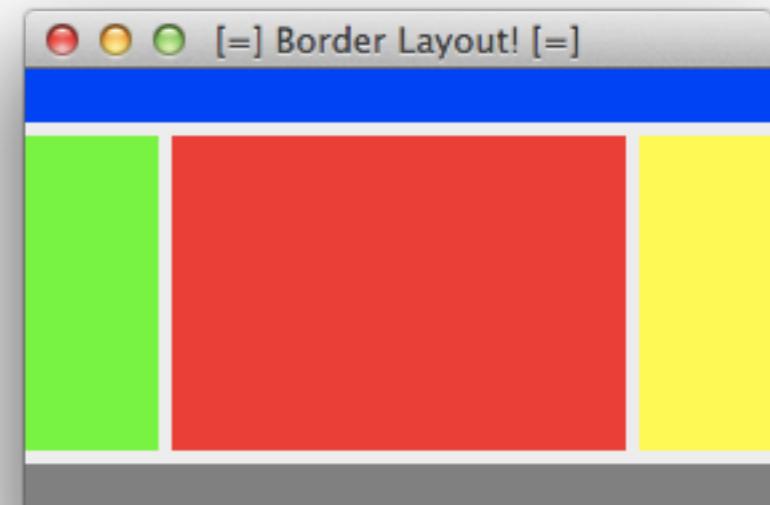
```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

the parameters are the horizontal and vertical gap between the areas

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```

mainPanel.add(p) adds to
BorderLayout.CENTER



Border Layout - HowTo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

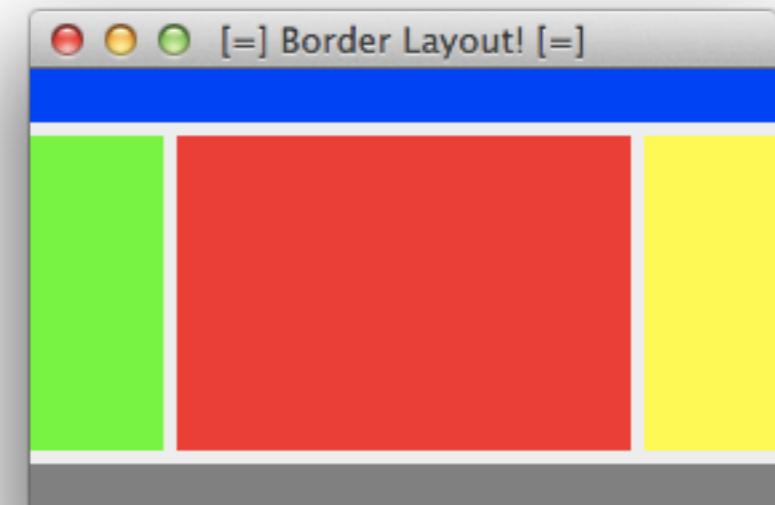
see: examples.swing.misc.BorderLayoutExample

the parameters are the horizontal and vertical gap between the areas

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```

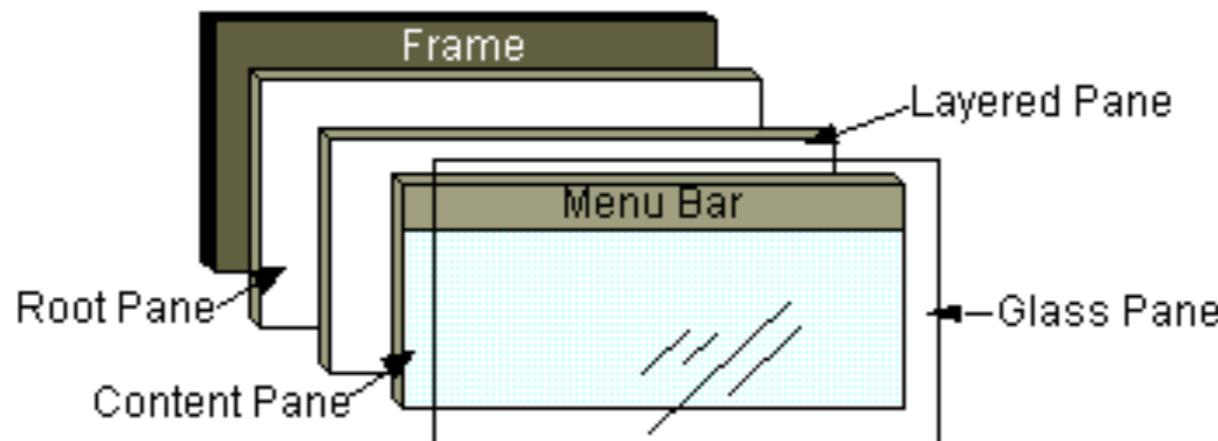
mainPanel.add(p) adds to
BorderLayout.CENTER

why the size of mainPanel
changes when changing
the size of the frame?



BorderLayout as Default Manager

ContentPane and JFrame use BorderLayout by default

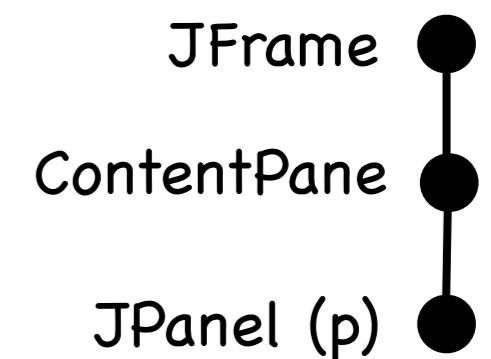


changes the content pane to be the panel p

```
JFrame frame = new JFrame("...");  
JPanel p = ...  
frame.setContentPane(p);
```

adds p to the CENTER of the content pane

```
JFrame frame = new JFrame("...");  
JPanel p = ...  
frame.getContentPane().add(p);
```



BorderLayout - more Examples

```
JPanel mainPanel = new JPanel( new BorderLayout(5,5) );
```

```
centerPanel = createPanel(Color.red,50,50);  
mainPanel.add(centerPanel, BorderLayout.CENTER);  
centerPanel.add( createButton("Hola!", 50, 20) );
```

...

```
JPanel mainPanel = new JPanel( new BorderLayout(5,5) );
```

```
centerPanel = createPanel(Color.red, 50, 50);  
mainPanel.add(centerPanel, BorderLayout.CENTER);  
centerPanel.setLayout( new BorderLayout() );  
centerPanel.add( createButton("Hola!", 50, 20) );
```

...

```
JPanel mainPanel = new JPanel( new BorderLayout(5,5) );
```

```
mainPanel.add( createButton("Hola!", 50, 20) );
```

...

Explain what would happen when resizing the window in each of these examples.



BoxLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.Y_AXIS ) );  
  
    addButton("Hola", mainPanel);  
    addButton("Holaa", mainPanel);  
    addButton("Holaaa", mainPanel);  
    addButton("Holaaaa", mainPanel);  
    addButton("Holaaaaa", mainPanel);  
  
    mainPanel.setOpaque(true);  
    ...  
    frame.pack();  
    frame.setVisible(true);  
}  
  
private void addButton(String text, Container container) {  
    JButton button = new JButton(text);  
    button.setAlignmentX(Component.CENTER_ALIGNMENT);  
    container.add(button);  
}
```

see: examples.swing.misc.BoxLayoutExample

BoxLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.Y_AXIS ) );
```

```
    addButton("Hola", mainPanel);  
    addButton("Holaa", mainPanel);  
    addButton("Holaaa", mainPanel);  
    addButton("Holaaaa", mainPanel);  
    addButton("Holaaaaa", mainPanel);
```

```
    mainPanel.setOpaque(true);  
    ...  
    frame.pack();  
    frame.setVisible(true);  
}
```

```
private void addButton(String text, Container container) {  
    JButton button = new JButton(text);  
    button.setAlignmentX(Component.CENTER_ALIGNMENT);  
    container.add(button);  
}
```

The Window is resized to fit
the preferred size and layouts
of its subcomponents.



BoxLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.Y_AXIS ) );  
  
    addButton("Hola", mainPanel);  
    addButton("Holaa", mainPanel);  
    addButton("Holaaa", mainPanel);  
    addButton("Holaaaa", mainPanel);  
    addButton("Holaaaaa", mainPanel);  
  
    mainPanel.setOpaque(true);  
    ...  
    frame.pack();  
    frame.setVisible(true);  
}  
  
private void addButton(String text, Container container) {  
    JButton button = new JButton(text);  
    button.setAlignmentX(Component.CENTER_ALIGNMENT);  
    container.add(button);  
}
```



LEFT



RIGHT



BoxLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.Y_AXIS ) );  
  
    addButton("Hola", mainPanel);  
    addButton("Holaa", mainPanel);  
    addButton("Holaaa", mainPanel);  
    addButton("Holaaaa", mainPanel);  
    addButton("Holaaaaa", mainPanel);  
  
    mainPanel.setOpaque(true);  
    ...  
    frame.pack();  
    frame.setVisible(true);  
}  
  
private void addButton(String text, Container container) {  
    JButton button = new JButton(text);  
    button.setAlignmentX(Component.CENTER_ALIGNMENT);  
    container.add(button);  
}
```

BoxLayout honoring the minimum and maximum sizes of each of the components. Any extra space appears at the bottom (resp. right) of the container. Use setMaximumSize, setMinimumSize, setPreferredSize, etc.



LEFT



RIGHT

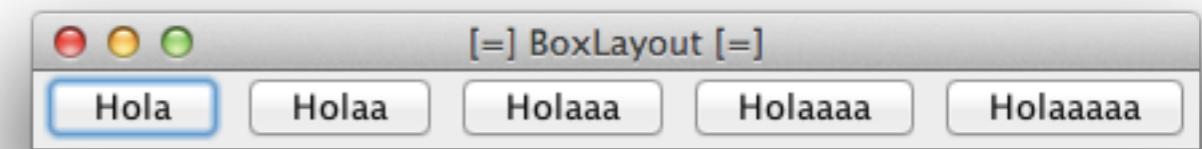


BoxLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.Y_AXIS ) );  
  
    addButton("Hola", mainPanel);  
    addButton("Holaa", mainPanel);  
    addButton("Holaaa", mainPanel);  
    addButton("Holaaaa", mainPanel);  
    addButton("Holaaaaa", mainPanel);  
  
    mainPanel.setOpaque(true);  
    ...  
    frame.pack();  
    frame.setVisible(true);  
}
```

```
private void addButton(String text, Container container) {  
    JButton button = new JButton(text);  
    button.setAlignmentX(Component.CENTER_ALIGNMENT);  
    container.add(button);  
}
```

X_AXIS



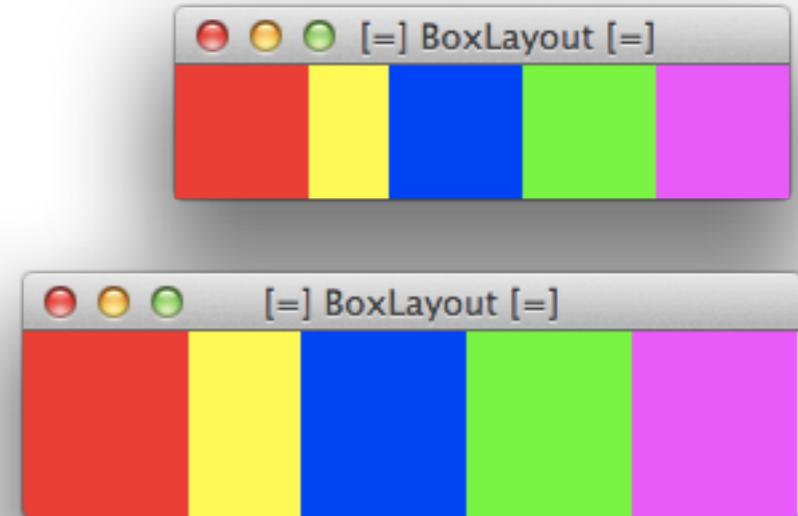
setAlignmentY(...)

CENTER_ALIGNMENT
TOP_ALIGNMENT
BOTTOM_ALIGNMENT

BoxLayout honoring the minimum and maximum sizes of each of the components. Any extra space appears at the bottom (resp. right) of the container. Use setMaximumSize, setMinimumSize, setPreferredSize, etc.

BoxLayout - More on Resizing

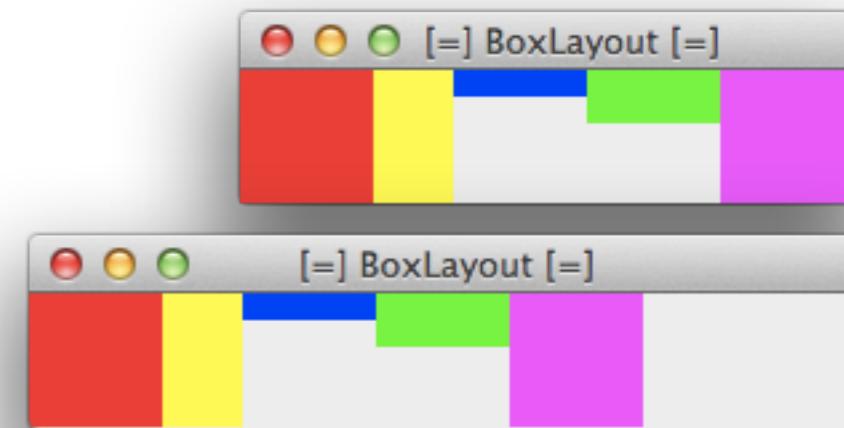
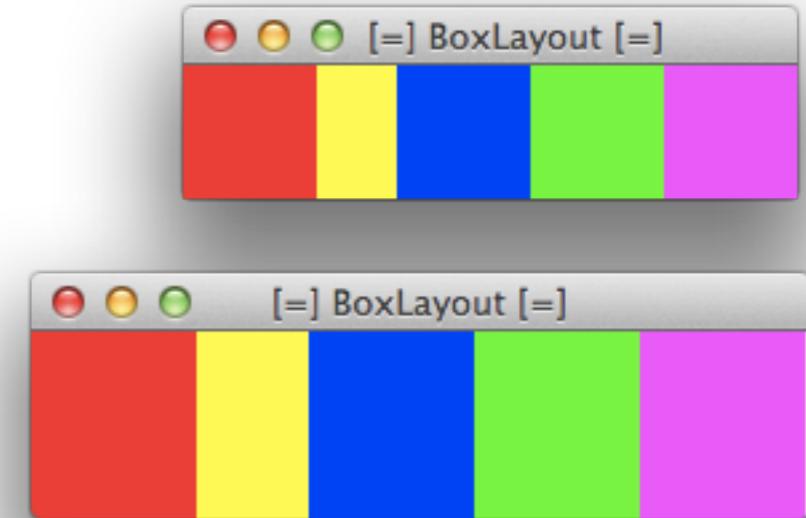
```
private void initGUI() {  
  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.X_AXIS ) );  
  
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 30, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 10) );  
    mainPanel.add( createPanel(Color.green, 50, 20) );  
    mainPanel.add( createPanel(Color.magenta, 50, 50) );  
  
    ...  
  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setAlignmentY(Component.TOP_ALIGNMENT);  
    panel.setPreferredSize(new Dimension(x, y));  
  
    return panel;  
}
```



The maximum size of a panel is very large by default, for example 32767x32767. Thus when resizing the window the panels change size as well

BoxLayout - More on Resizing

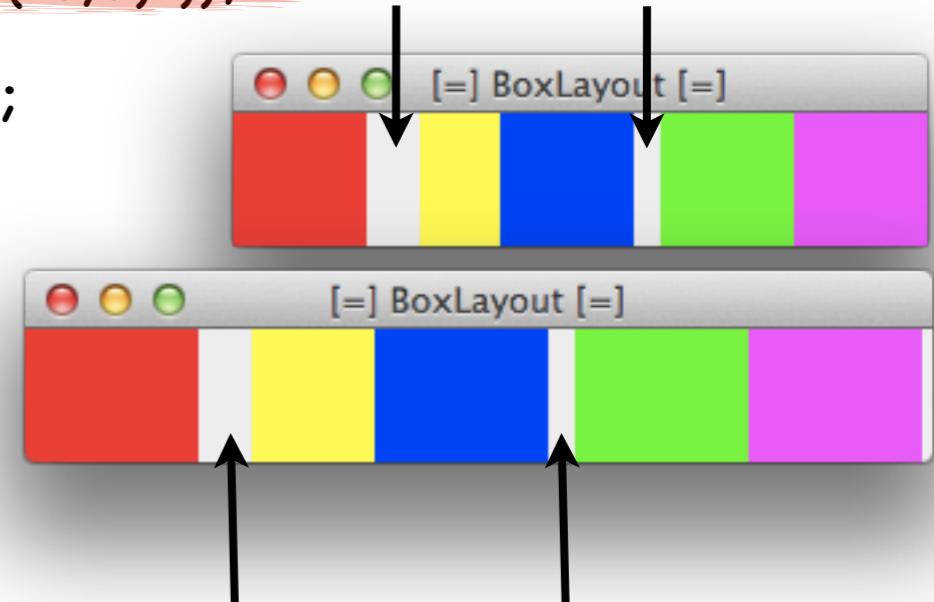
```
private void initGUI() {  
  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.X_AXIS ) );  
  
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 30, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 10) );  
    mainPanel.add( createPanel(Color.green, 50, 20) );  
    mainPanel.add( createPanel(Color.magenta, 50, 50) );  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setAlignmentY(Component.TOP_ALIGNMENT);  
    panel.setPreferredSize(new Dimension(x, y));  
  
    panel.setMaximumSize(new Dimension(x, y));  
    panel.setMinimumSize(new Dimension(x, y));  
  
    return panel;  
}
```



see: examples.swing.misc.BoxLayoutExample_1, examples.swing.misc.BoxLayoutExample_2

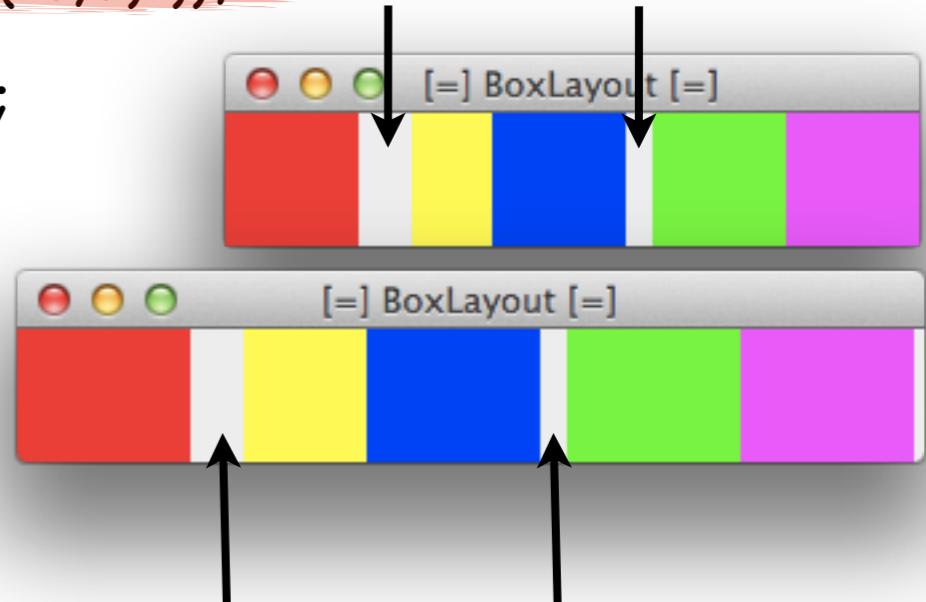
BoxLayout - Rigid Area

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.X_AXIS ) );  
  
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( Box.createRigidArea( new Dimension(20,0) ));  
    mainPanel.add( createPanel(Color.yellow, 30, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 10) );  
    mainPanel.add( Box.createRigidArea( new Dimension(10,0) ));  
    mainPanel.add( createPanel(Color.green, 50, 20) );  
    mainPanel.add( createPanel(Color.magenta, 50, 50) );  
  
    mainPanel.setOpaque(true);  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setAlignmentY(Component.TOP_ALIGNMENT);  
    panel.setPreferredSize(new Dimension(x, y));  
  
    return panel;  
}
```



BoxLayout - Rigid Area

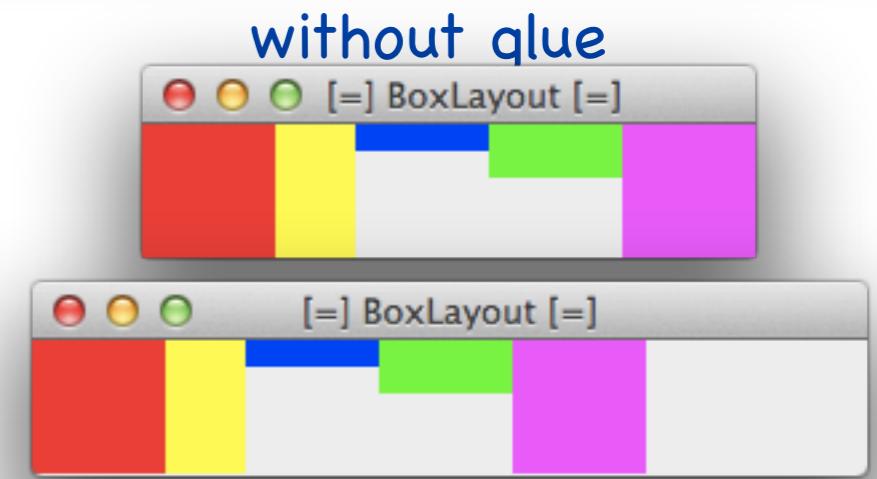
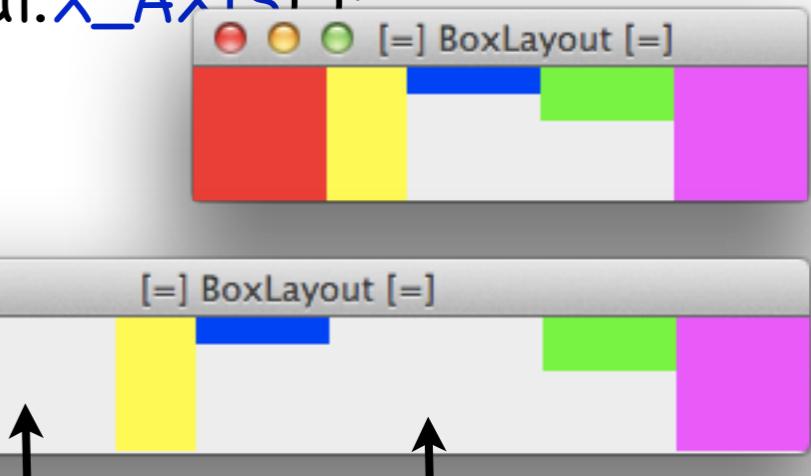
```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.X_AXIS ) );  
  
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( Box.createRigidArea( new Dimension(20,0) ));  
    mainPanel.add( createPanel(Color.yellow, 30, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 10) );  
    mainPanel.add( Box.createRigidArea( new Dimension(10,0) ));  
    mainPanel.add( createPanel(Color.green, 50, 20) );  
    mainPanel.add( createPanel(Color.magenta, 50, 50) );  
  
    mainPanel.setOpaque(true);  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setAlignmentY(Component.TOP_ALIGNMENT);  
    panel.setPreferredSize(new Dimension(x, y));  
  
    return panel;  
}
```



You can do the same with an auxiliary panel, or any class that extends JComponent as we will see later.

BoxLayout - Glue

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout( new BoxLayout( mainPanel, BoxLayout.X_AXIS ) );  
  
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( Box.createHorizontalGlue() );  
    mainPanel.add( createPanel(Color.yellow, 30, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 10) );  
    mainPanel.add( Box.createHorizontalGlue() );  
    mainPanel.add( createPanel(Color.green, 50, 20) );  
    mainPanel.add( createPanel(Color.magenta, 50, 50) );  
  
    mainPanel.setOpaque(true);  
}  
  
...  
  
private JPanel createPanel(Color color, int x, int y) {  
  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setAlignmentY(Component.TOP_ALIGNMENT);  
    panel.setPreferredSize(new Dimension(x, y));  
    panel.setMaximumSize(new Dimension(x, y));  
    panel.setMinimumSize(new Dimension(x, y));  
    return panel;  
}
```

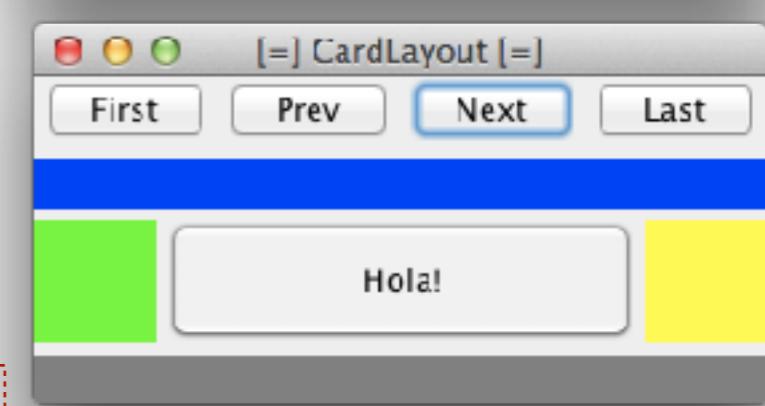
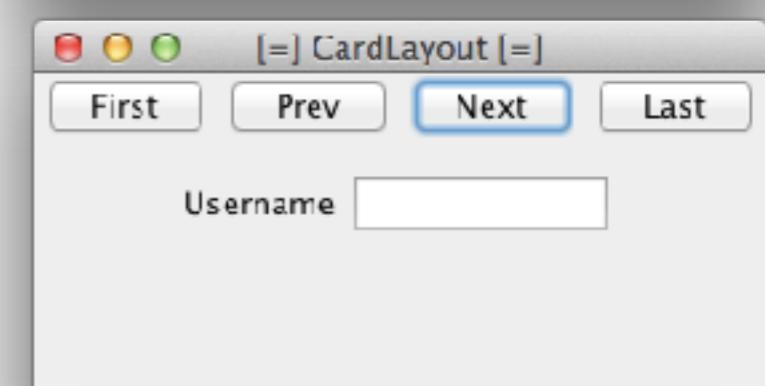
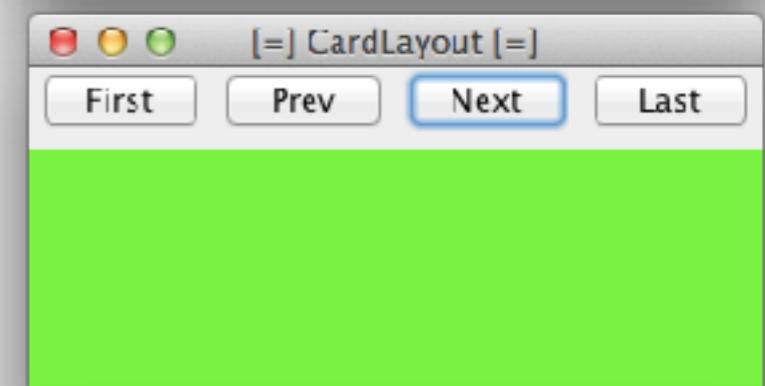
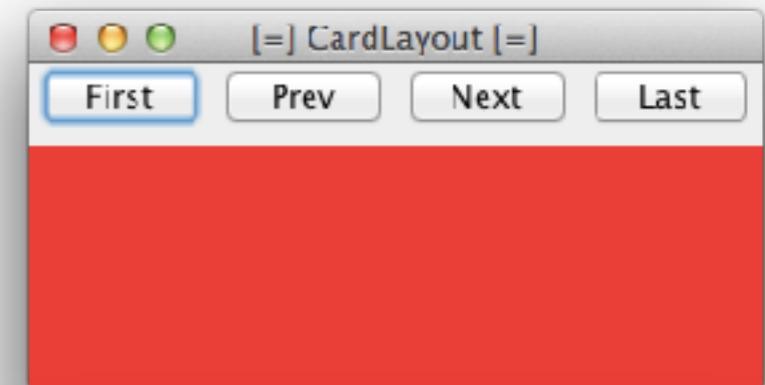


You can do something similar using an auxiliary panel (or any class that extends JComponent).

CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...), "Card 1");  
    ...  
    cardsPanel.add( someContent(), "Card 4");  
  
}  
...  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

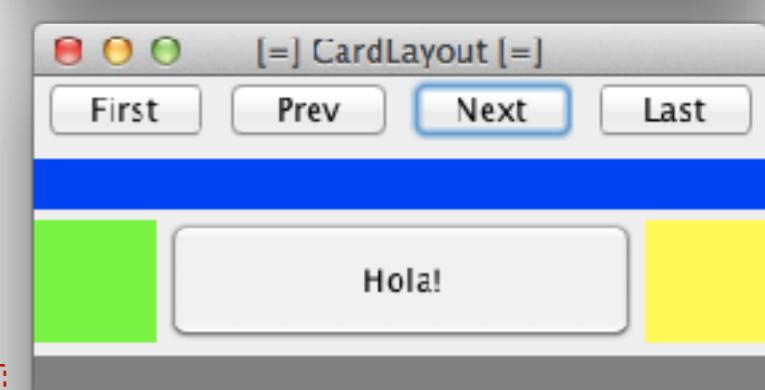
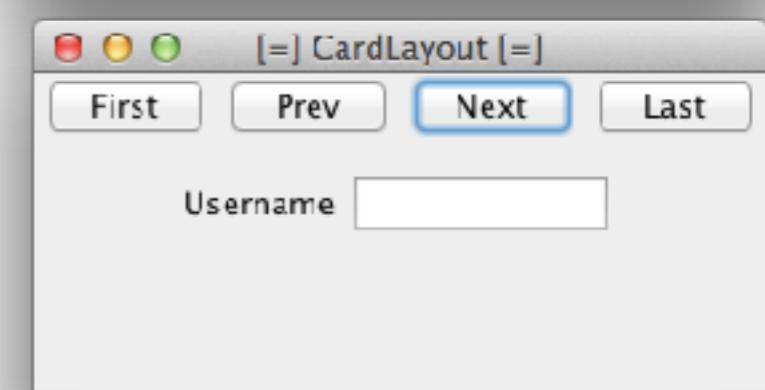
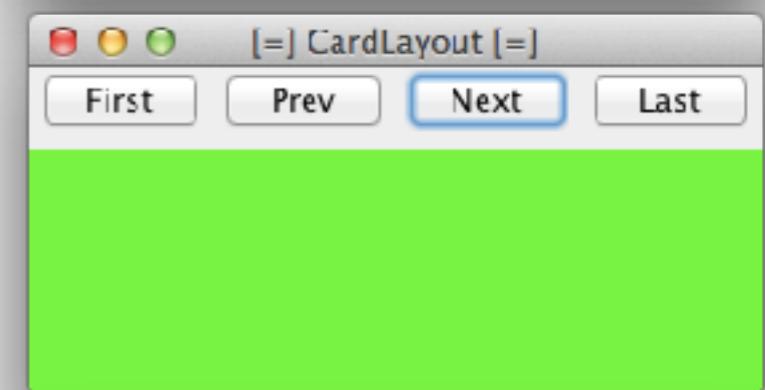
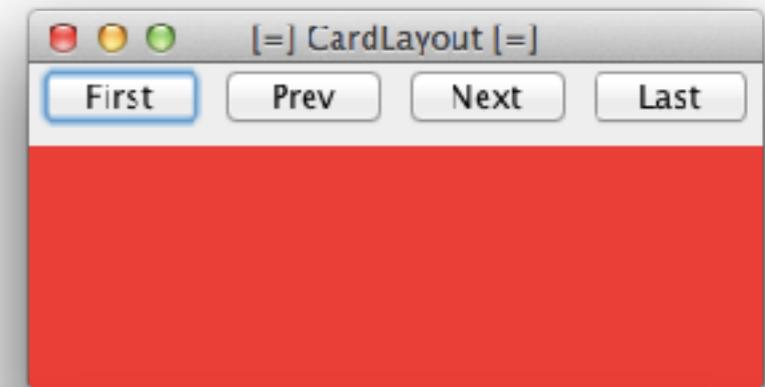
see: examples.swing.misc.CardLayout



CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...) , "Card 1");  
    ...  
    cardsPanel.add( someContent() , "Card 4");  
  
}  
...  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

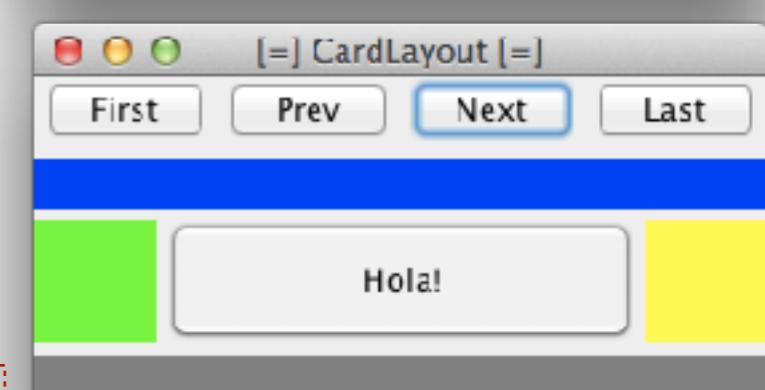
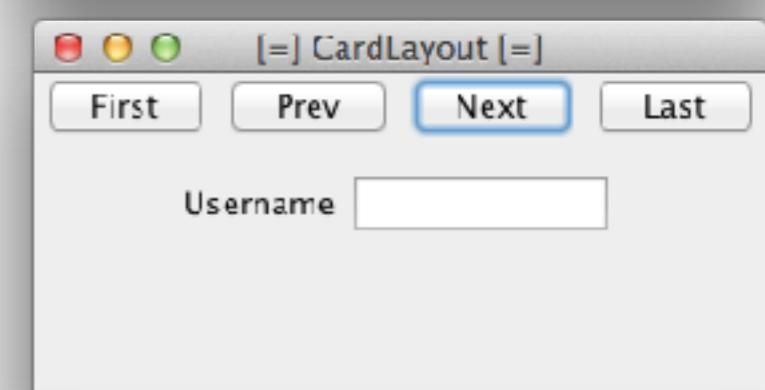
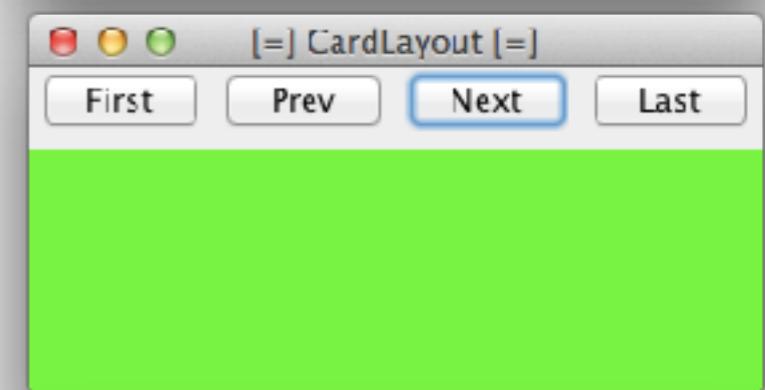
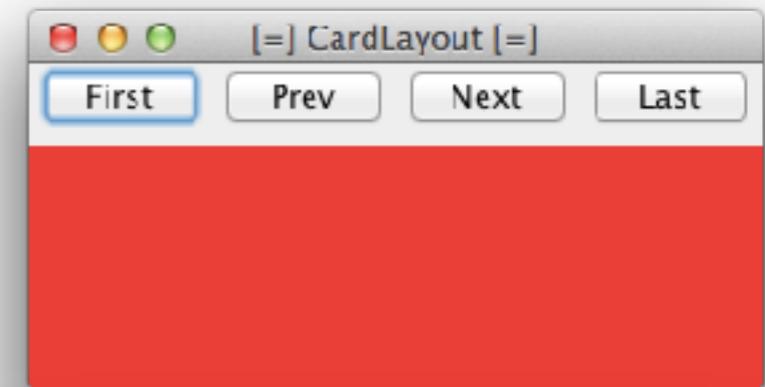
see: examples.swing.misc.CardLayout



CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...) , "Card 1");  
    ...  
    cardsPanel.add( someContent() , "Card 4");  
  
}  
...  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

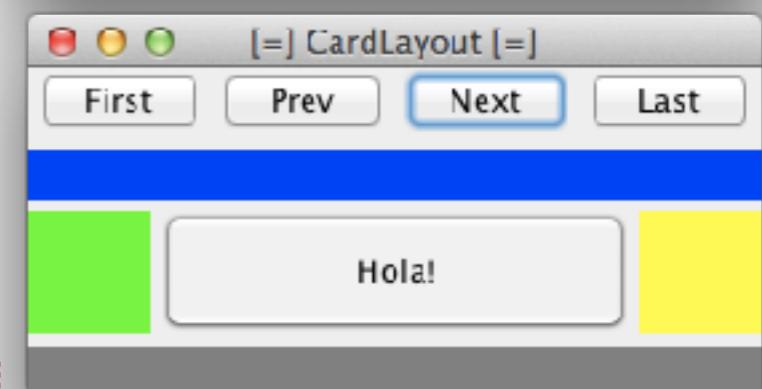
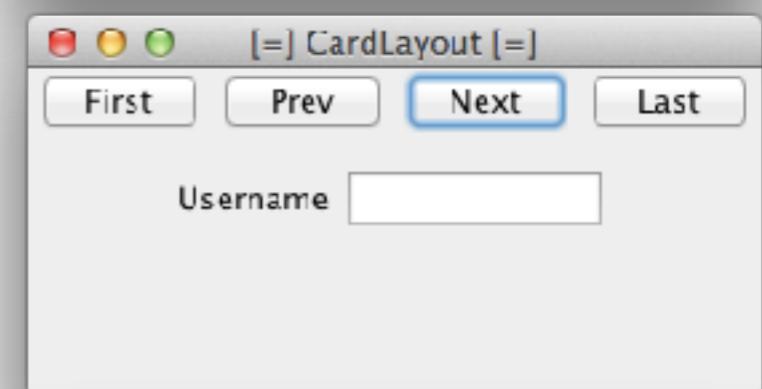
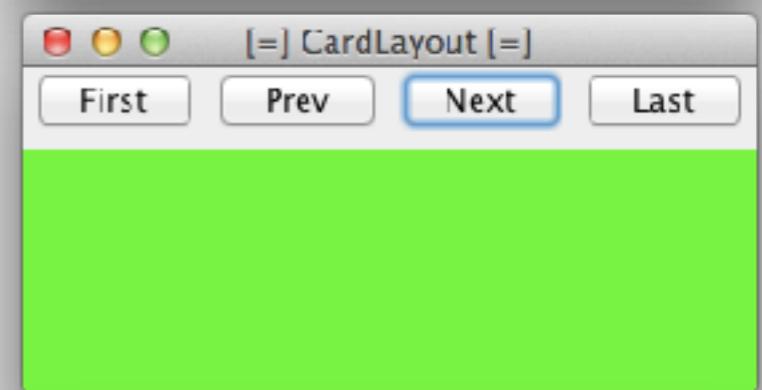
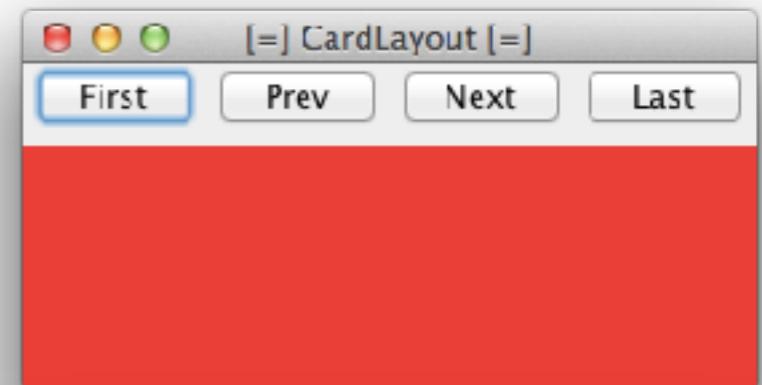
see: examples.swing.misc.CardLayout



CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...) , "Card 1");  
    ...  
    cardsPanel.add( someContent() , "Card 4");  
  
}  
...  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

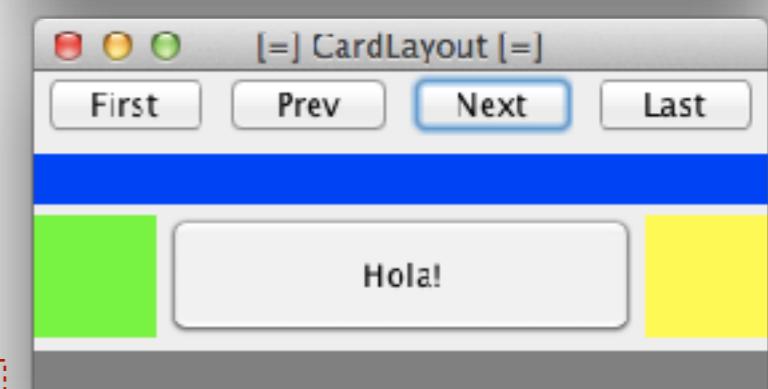
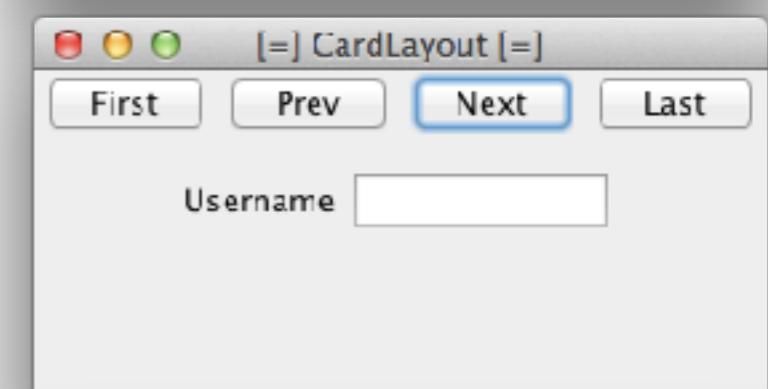
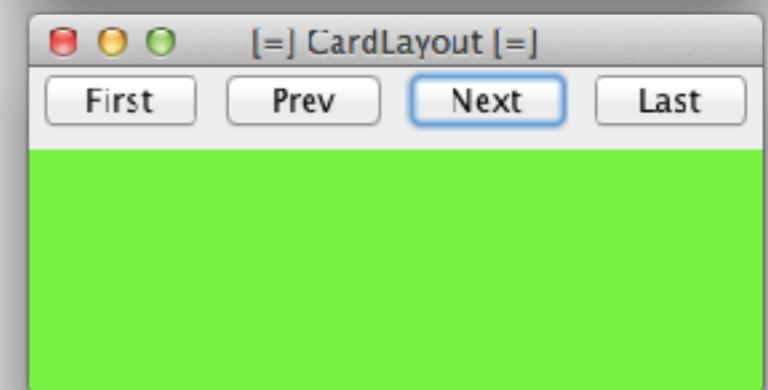
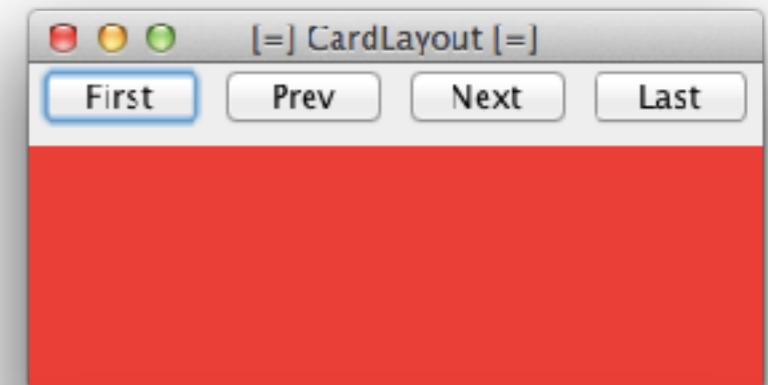
see: examples.swing.misc.CardLayout



CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...), "Card 1");  
    ...  
    cardsPanel.add( someContent(), "Card 4");  
    ...  
}  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

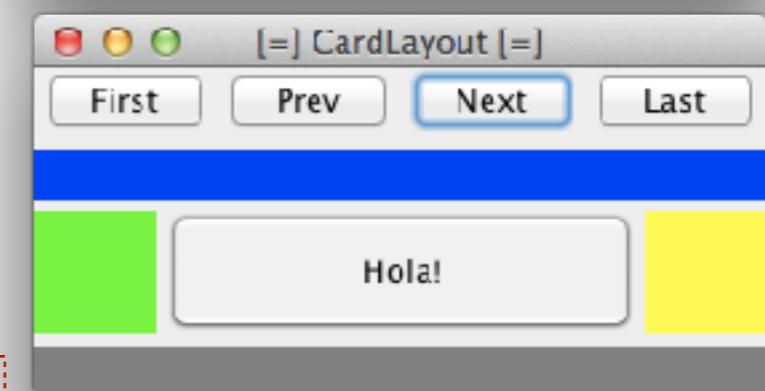
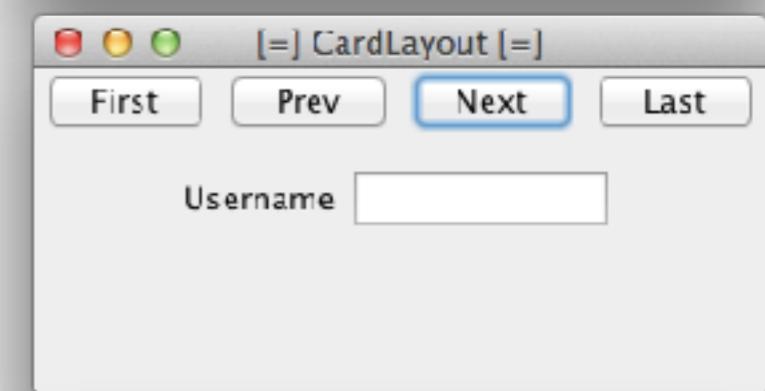
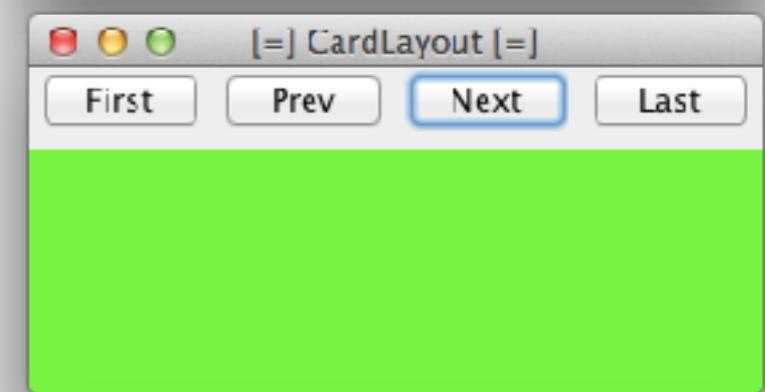
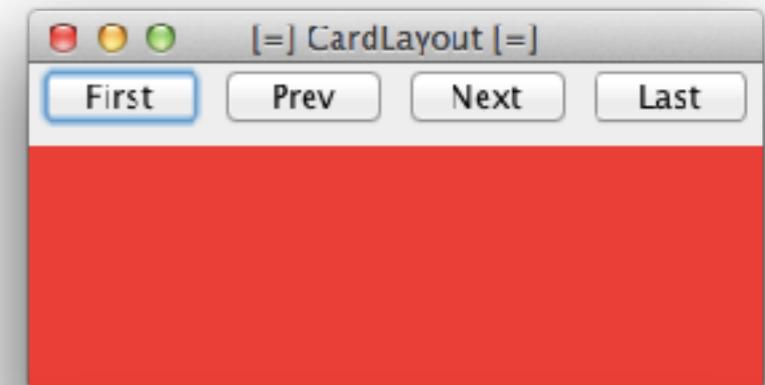
see: examples.swing.misc.CardLayout



CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...), "Card 1");  
    ...  
    cardsPanel.add( someContent(), "Card 4");  
    ...  
}  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

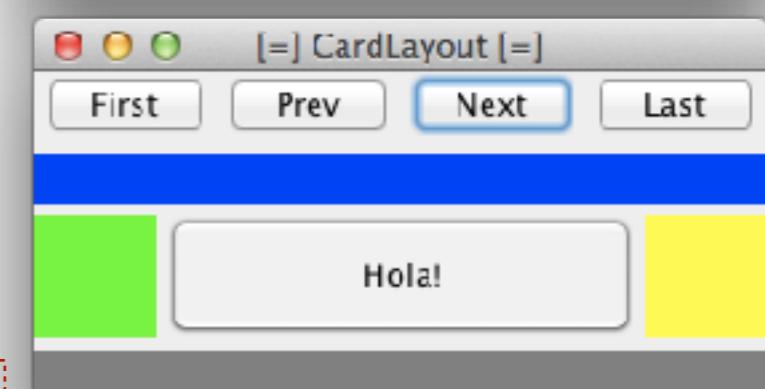
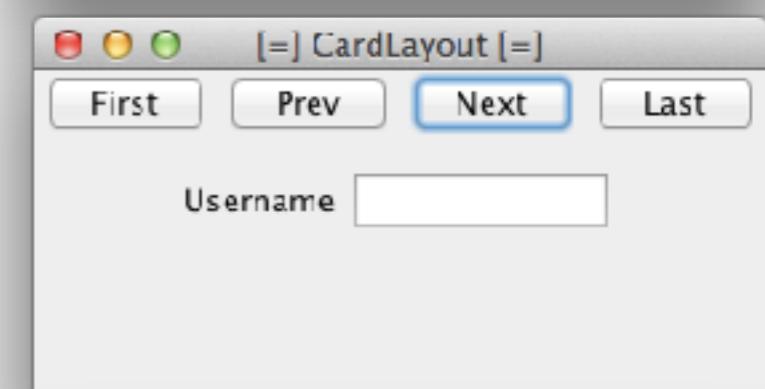
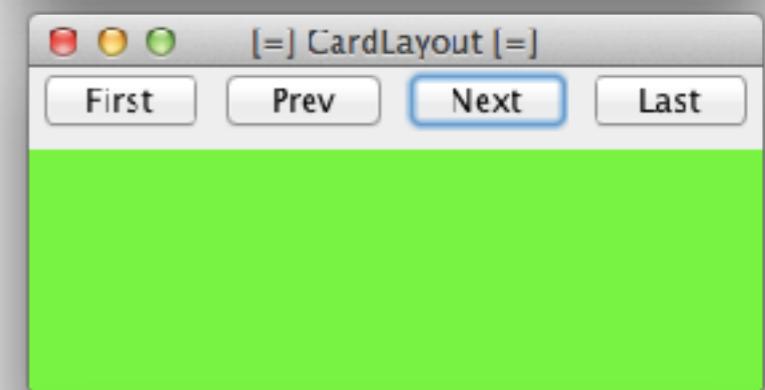
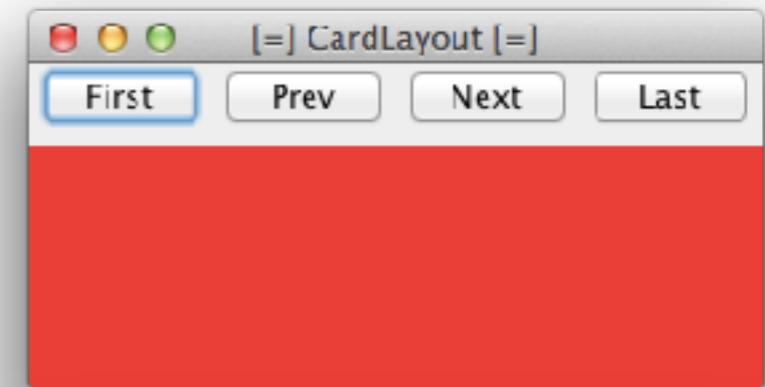
see: examples.swing.misc.CardLayout



CardsLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new BorderLayout());  
  
    buttonsPanel = createButtons();  
    mainPanel.add(buttonsPanel, BorderLayout.PAGE_START);  
  
    cardsPanel = new JPanel(new CardLayout());  
    mainPanel.add(cardsPanel, BorderLayout.CENTER);  
  
    cardsPanel.add( createPanel(...), "Card 1");  
    ...  
    cardsPanel.add( someContent(), "Card 4");  
    ...  
}  
  
public void actionPerformed(ActionEvent e) {  
    CardLayout cl = (CardLayout) cardsPanel.getLayout();  
    if ( e.getSource() == nextButton ) {  
        cl.next(cardsPanel);  
    } else if ( e.getSource() == prevButton ) {  
        cl.previous(cardsPanel);  
    } else if ( e.getSource() == firstButton ) {  
        cl.first(cardsPanel);  
    } else if ( e.getSource() == lastButton ) {  
        cl.last(cardsPanel);  
    }  
}
```

see: examples.swing.misc.CardLayout



GridLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new GridLayout(5, 2, 3, 3));  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    JPanel blue = createSquareJPanel(Color.blue, 100);  
    JPanel green = createSquareJPanel(Color.green, 100);  
    JPanel orange = createSquareJPanel(Color.orange, 100);  
    JPanel yellow = createSquareJPanel(Color.yellow, 50);  
    JPanel pink = createSquareJPanel(Color.pink, 50);  
    JPanel cyan = createSquareJPanel(Color.cyan, 50);  
    JPanel gray = createSquareJPanel(Color.gray, 50);  
    JPanel black = createSquareJPanel(Color.black, 50);  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
  
    mainPanel.add(red);  
    mainPanel.add(blue);  
    ...  
}
```



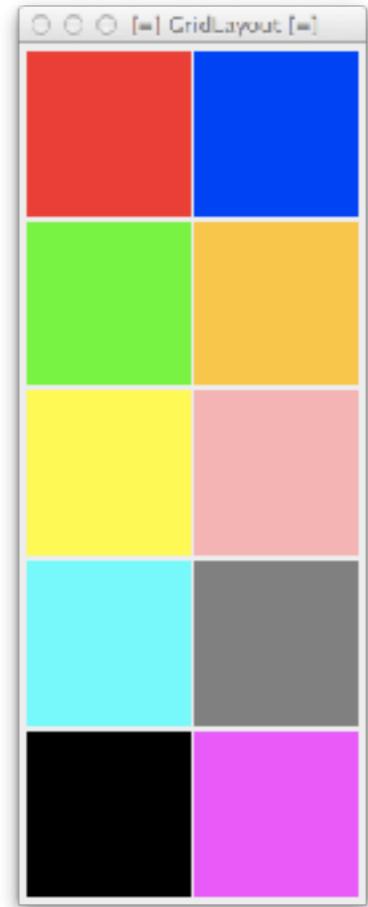
see: examples.swing.misc.GridLayout, examples.swing.misc.GridLayout_1

GridLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new GridLayout(5, 2, 3, 3));
```

rows x cols spacing

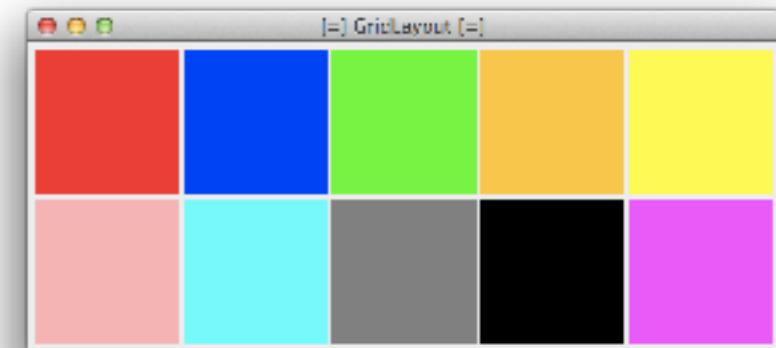
```
JPanel red = createSquareJPanel(Color.red, 100);  
JPanel blue = createSquareJPanel(Color.blue, 100);  
JPanel green = createSquareJPanel(Color.green, 100);  
JPanel orange = createSquareJPanel(Color.orange, 100);  
JPanel yellow = createSquareJPanel(Color.yellow, 50);  
JPanel pink = createSquareJPanel(Color.pink, 50);  
JPanel cyan = createSquareJPanel(Color.cyan, 50);  
JPanel gray = createSquareJPanel(Color.gray, 50);  
JPanel black = createSquareJPanel(Color.black, 50);  
JPanel magenta = createSquareJPanel(Color.magenta, 50);
```



```
mainPanel.add(red);  
mainPanel.add(blue);  
...  
}
```

- ◆ Places component row by row, from left to right
- ◆ All components are resized to fit the size of a cell

GridLayout(2,5)



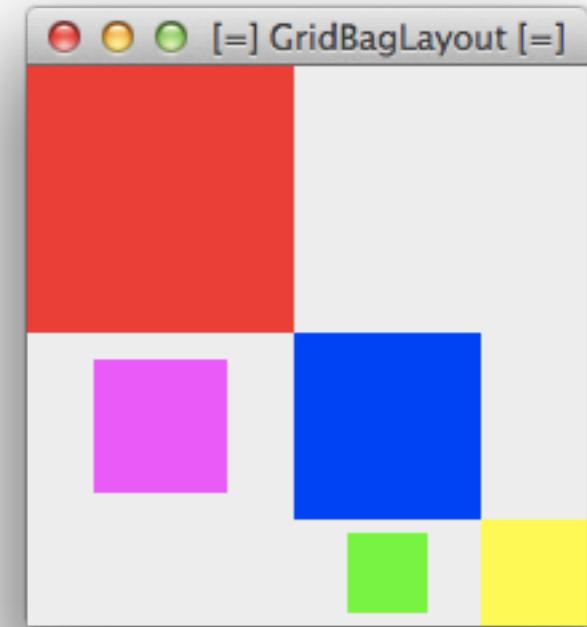
see: examples.swing.misc.GridLayout, examples.swing.misc.GridLayout_1

GridBagLayout

- ◆ Similar to GridLayout, but more flexible (and more complicated!)
- ◆ It allows placing a component in cell (i,j)
- ◆ Different sizes for cells, row, columns, etc.
- ◆ It allows spanning a component over several cells
- ◆ It allows better positioning inside the cell
- ◆ ...

GridLayout

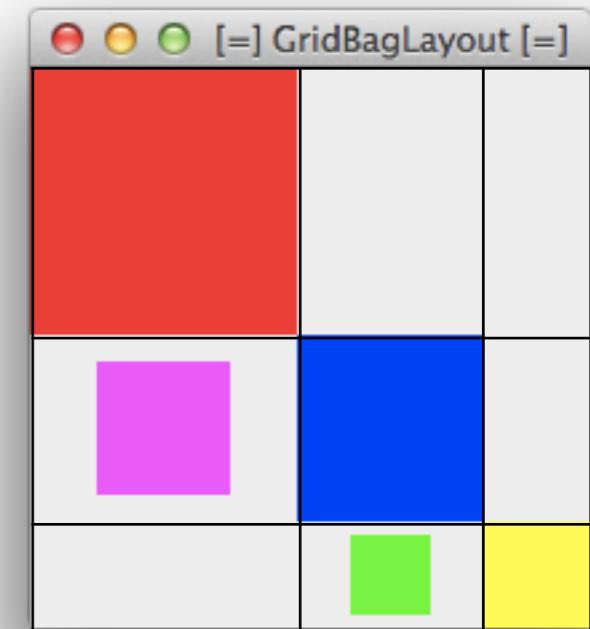
```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2;  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



see: [examples.swing.misc.GridBagLayout](#), [examples.swing.misc.GridBagLayout_1](#), [examples.swing.misc.GridBagLayout_2](#)

GridLayout

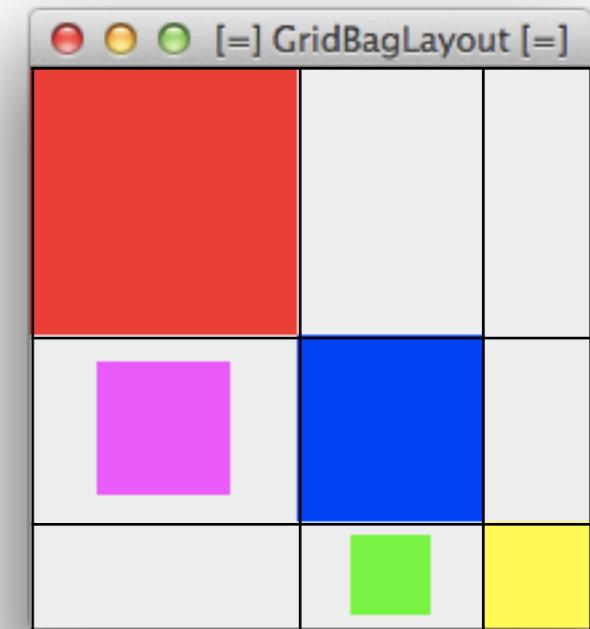
```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2;  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



see: [examples.swing.misc.GridBagLayout](#), [examples.swing.misc.GridBagLayout_1](#), [examples.swing.misc.GridBagLayout_2](#)

GridLayout

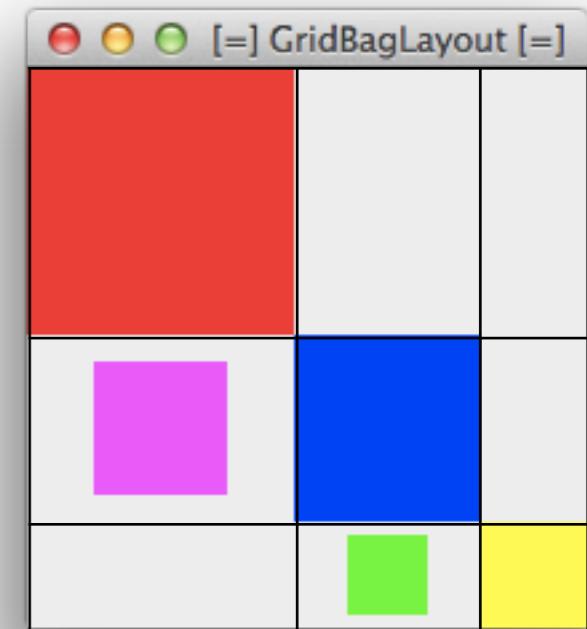
```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2;  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridLayout

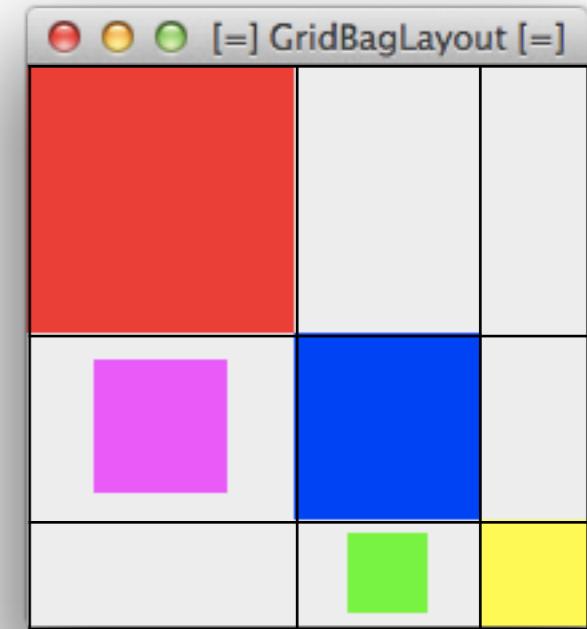
```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2;  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridLayout

```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2;  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```

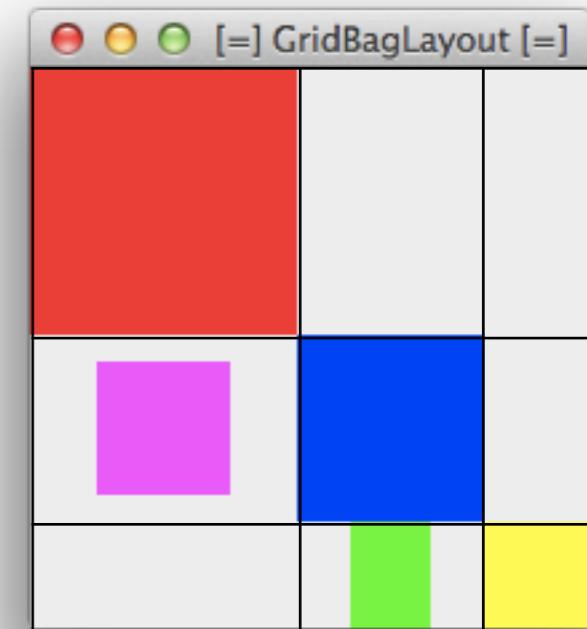


- ◆ A column width is the max width its components
- ◆ A row height is the max height its components

see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridBagLayout

```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridBagLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2; ←  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



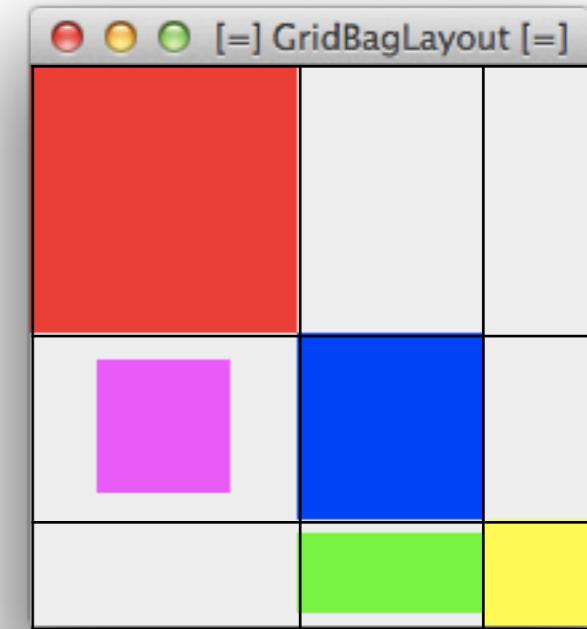
- ◆ A column width is the max width its components
- ◆ A row height is the max height its components

c.fill = GridBagConstraints.VERTICAL;

see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridLayout

```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2; ←  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



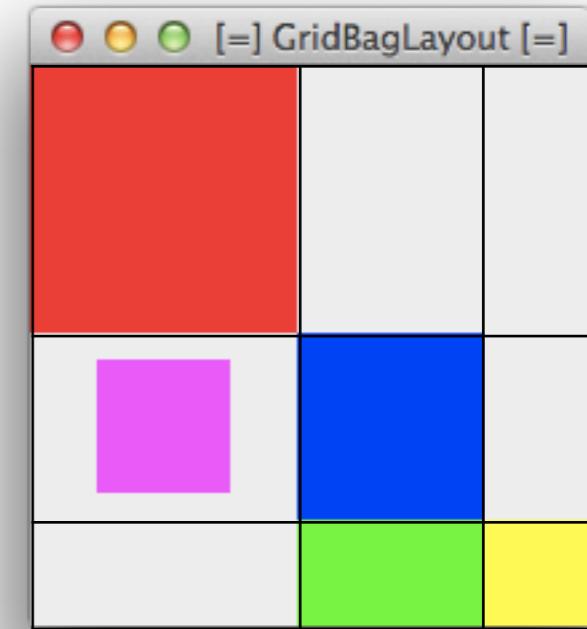
- ◆ A column width is the max width its components
- ◆ A row height is the max height its components

c.fill = GridBagConstraints.HORIZONTAL;

see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridLayout

```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2; ←  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1;  
    mainPanel.add(magenta, c);  
    ...  
}
```



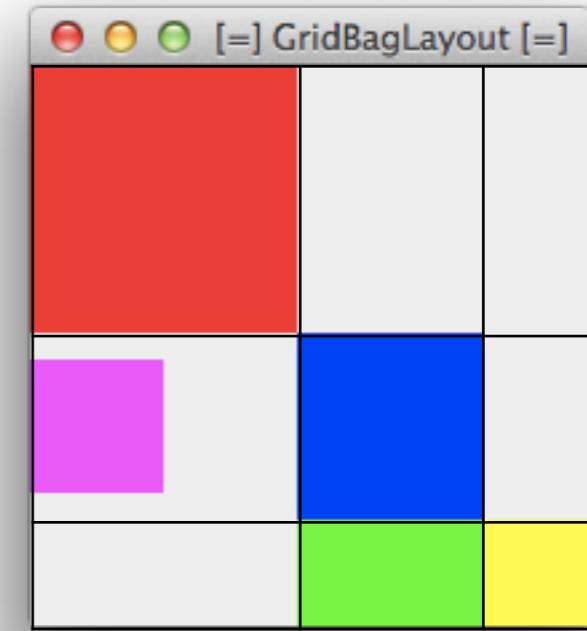
- ◆ A column width is the max width its components
- ◆ A row height is the max height its components

c.fill = GridBagConstraints.BOTH;

see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridLayout

```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green, 30);  
    c.gridx = 1; c.gridy = 2; ←  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.gridx = 0; c.gridy = 1; ←  
    mainPanel.add(magenta, c);  
    ...  
}
```



- ◆ A column width is the max width its components
- ◆ A row height is the max height its components

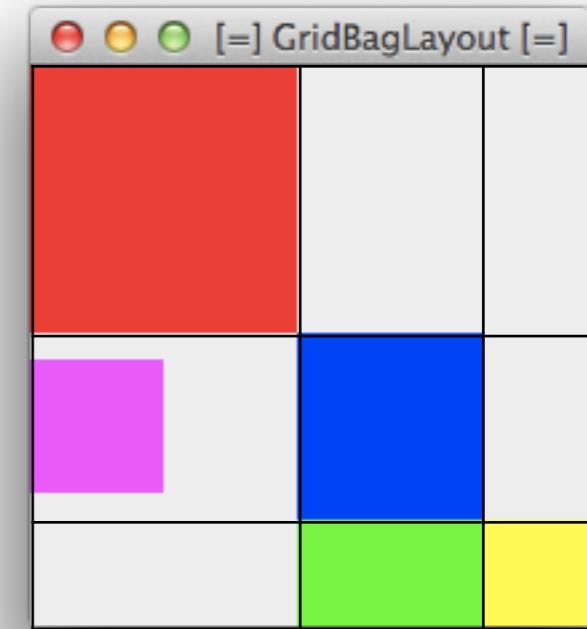
c.fill = GridBagConstraints.BOTH;

c.anchor = GridBagConstraints.LINE_START;

see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridLayout

```
private void initGUI() {  
    ...  
    JPanel mainPanel = new JPanel(new GridLayout());  
  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.gridx = 0; c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 70);  
    c.gridx = 1; c.gridy = 1;  
    mainPanel.add(blue, c);  
  
    JPanel yellow = createSquareJPanel(Color.yellow, 40);  
    c.gridx = 2; c.gridy = 2;  
    mainPanel.add(yellow, c);  
  
    JPanel green = createSquareJPanel(Color.green);  
    c.gridx = 1; c.gridy = 2; ←  
    mainPanel.add(green, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta);  
    c.gridx = 0; c.gridy = 1; ←  
    mainPanel.add(magenta, c);  
    ...  
}
```



- ◆ A column width is the max width its components
- ◆ A row height is the max height its components

FIRST_LINE_START PAGE_START FIRST_LINE_END

LINE_START CENTER LINE_END

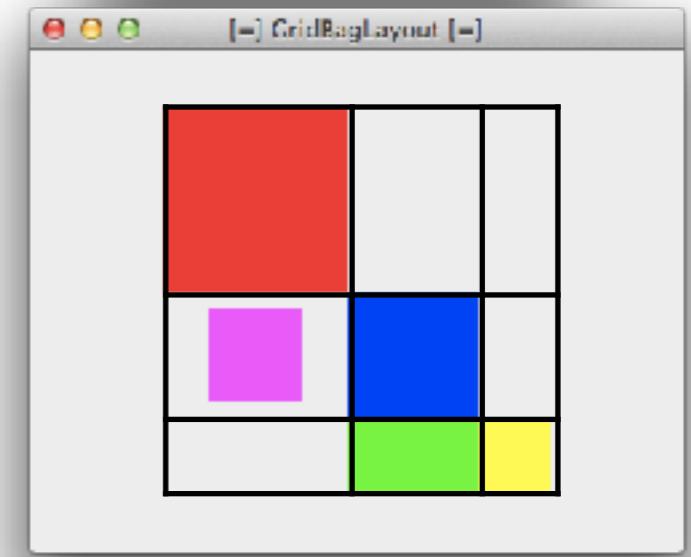
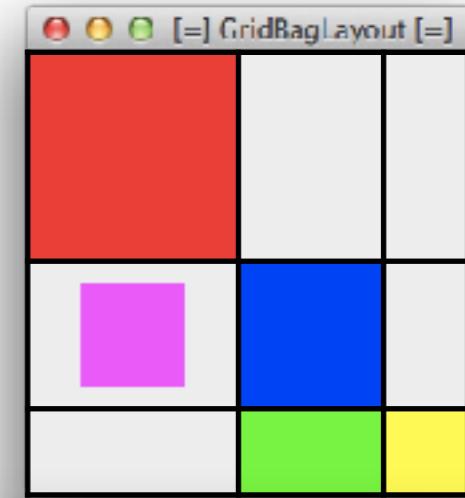
LAST_LINE_START PAGE_END LAST_LINE_END

c.anchor = GridBagConstraints.LINE_START;

see: examples.swing.misc.GridBagLayout, examples.swing.misc.GridBagLayout_1, examples.swing.misc.GridBagLayout_2

GridBagLayout - Resizing Cells

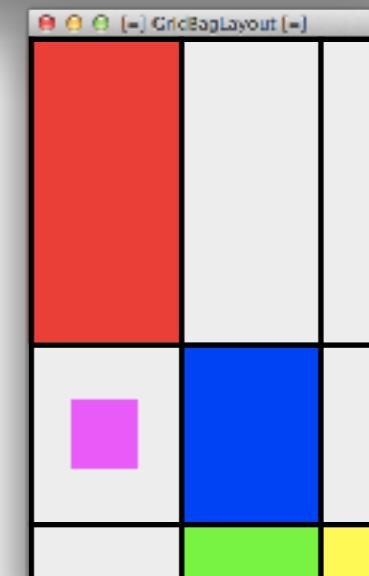
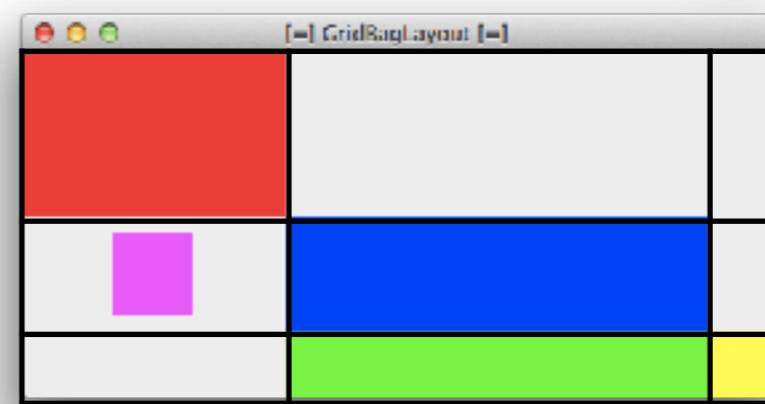
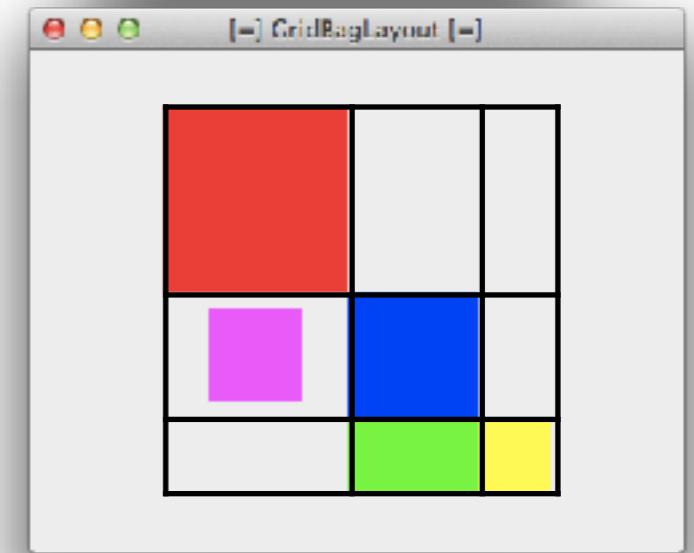
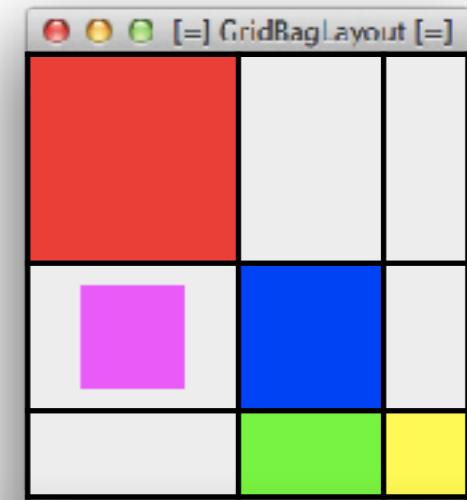
```
public JPanel createContentPane() {  
    ...  
  
    JPanel mainPanel = new JPanel(new GridBagLayout());  
    totalGUI.add(mainPanel);  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.fill = GridBagConstraints.BOTH;  
    c.gridx = 0;  
    c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 100);  
    c.fill = GridBagConstraints.BOTH;  
    c.gridx = 1;  
    c.gridy = 1;  
    mainPanel.add(blue, c);  
  
}
```



see: examples.swing.misc.GridBagLayout_3

GridBagLayout - Resizing Cells

```
public JPanel createContentPane() {  
    ...  
  
    JPanel mainPanel = new JPanel(new GridBagLayout());  
    totalGUI.add(mainPanel);  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.fill = GridBagConstraints.BOTH;  
    c.gridx = 0;  
    c.gridy = 0;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 100);  
    c.fill = GridBagConstraints.BOTH;  
    c.gridx = 1;  
    c.gridy = 1;  
    mainPanel.add(blue, c);  
  
}
```



see: examples.swing.misc.GridBagLayout_3

GridBagLayout - Resizing Cells

```
public JPanel createContentPane() {
```

```
...
```

```
JPanel mainPanel = new JPanel(new GridBagLayout());  
totalGUI.add(mainPanel);  
GridBagConstraints c = new GridBagConstraints();
```

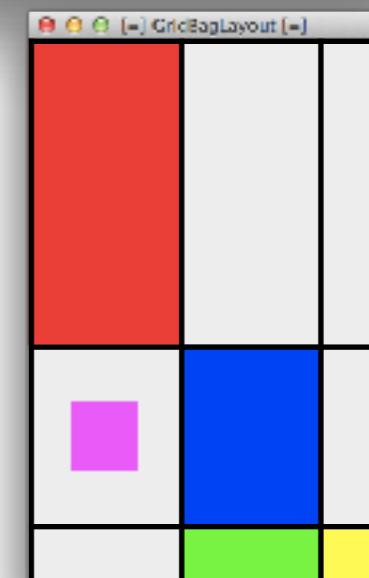
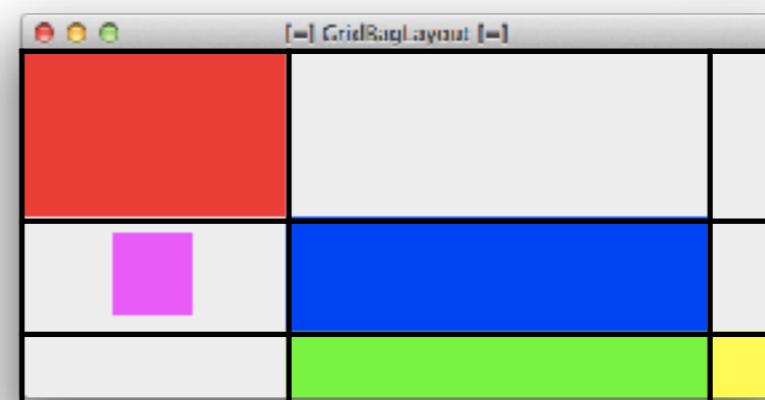
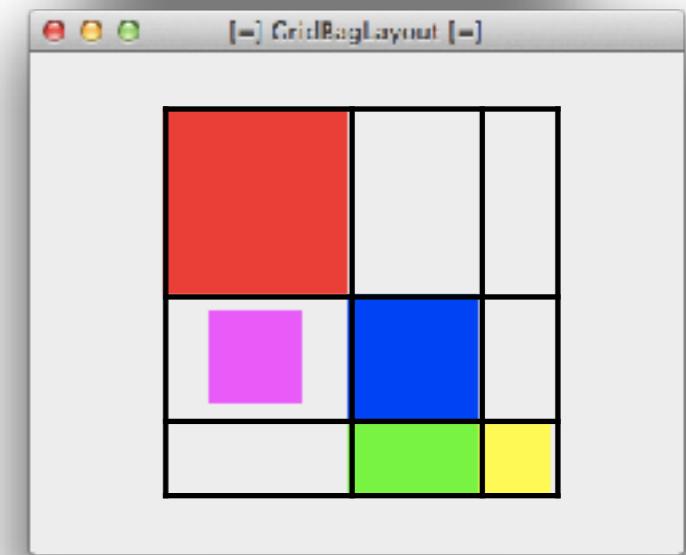
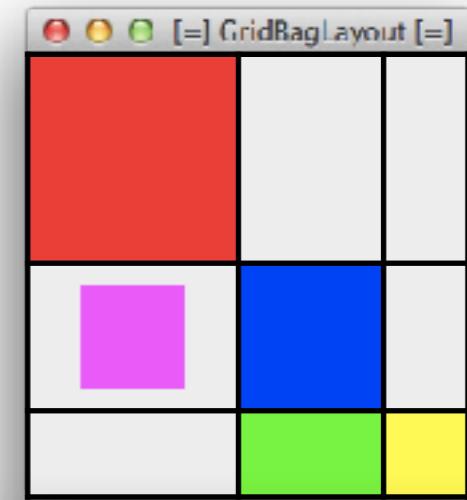
```
JPanel red = createSquareJPanel(Color.red, 100);  
c.fill = GridBagConstraints.BOTH;  
c.gridx = 0; c.weightx = 0.1;  
c.gridy = 0; c.weighty = 0.4;  
mainPanel.add(red, c);
```

```
JPanel blue = createSquareJPanel(Color.blue, 100);  
c.fill = GridBagConstraints.BOTH;  
c.gridx = 1; c.weightx = 0.3;  
c.gridy = 1; c.weighty = 0.2;  
mainPanel.add(blue, c);
```

```
}
```

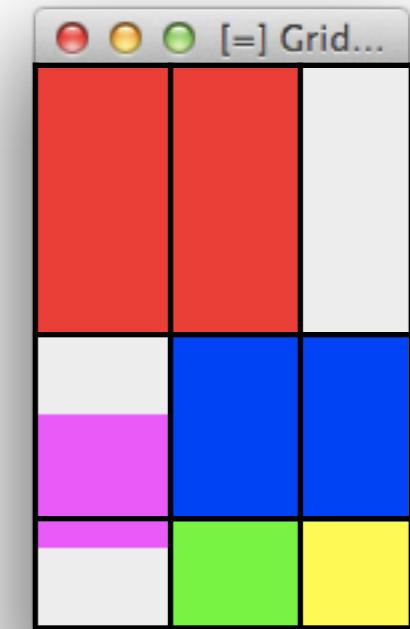
$0 \leq c.weightx \leq 1;$
 $0 \leq c.weighty \leq 1;$

see: examples.swing.misc.GridBagLayout_3



GridBagLayout - Spanning

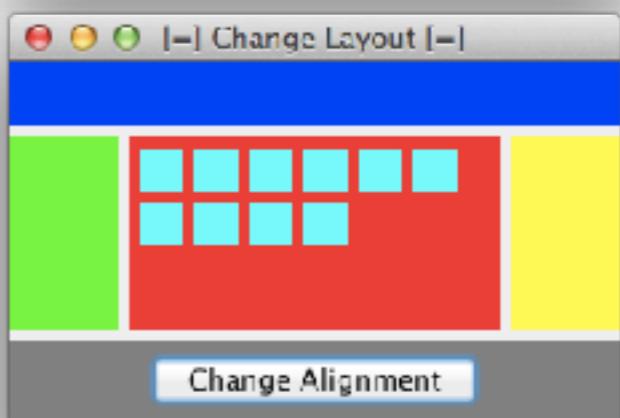
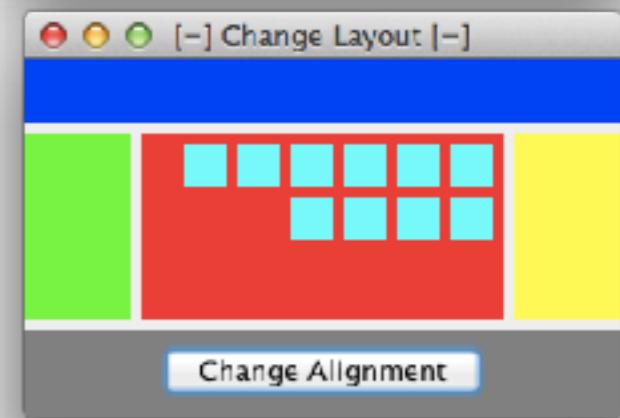
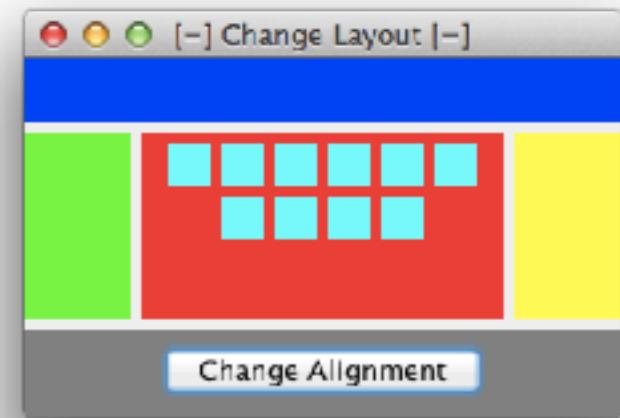
```
public JPanel createContentPane() {  
    ...  
    JPanel mainPanel = new JPanel(new GridBagLayout());  
    totalGUI.add(mainPanel);  
    GridBagConstraints c = new GridBagConstraints();  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    c.fill = GridBagConstraints.BOTH;  
    c.gridx = 0; c.gridy = 0;  
    c.gridwidth = 2; c.gridheight = 1;  
    mainPanel.add(red, c);  
  
    JPanel blue = createSquareJPanel(Color.blue, 100);  
    c.fill = GridBagConstraints.BOTH;  
    c.gridx = 1; c.gridy = 0;  
    c.gridwidth = 2; c.gridheight = 1;  
    mainPanel.add(blue, c);  
  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
    c.fill = GridBagConstraints.NONE;  
    c.gridx = 0; c.gridy = 1;  
    c.gridwidth = 1; c.gridheight = 2;  
    c.anchor = GridBagConstraints.CENTER;  
    mainPanel.add(magenta, c);  
    ...  
}
```



see: examples.swing.misc.GridBagLayout_4

Dynamic Change of Layout - Ex. I

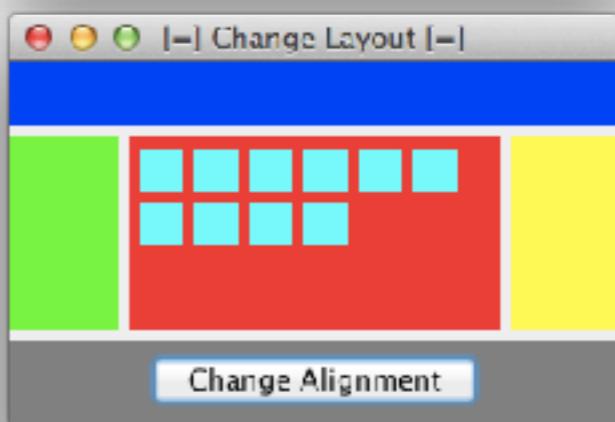
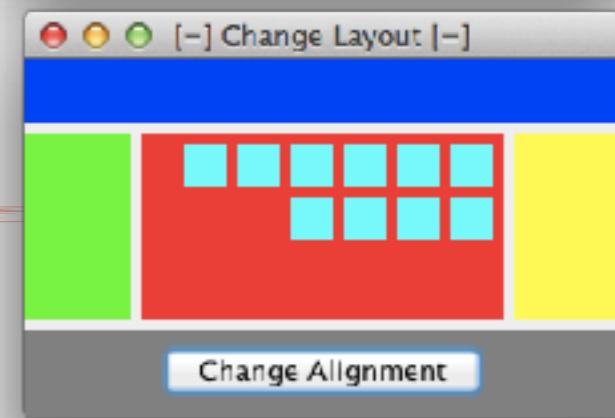
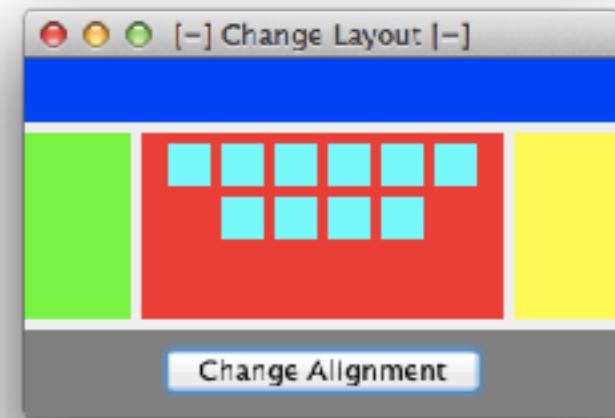
```
private void initGUI() {
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );
    ...
    for( int i=0; i<10; i++ )
        centerPanel.add( createPanel( Color.cyan, 20, 20 ) );
    alignButton = new JButton( "Change Alignment" );
    alignButton.addActionListener(this);
    botPanel.add( alignButton );
    ...
}
public void actionPerformed(ActionEvent arg0) {
    FlowLayout fl = ((FlowLayout)centerPanel.getLayout());
    int a = fl.setAlignment();
    switch ( a ) {
        case FlowLayout.LEFT:
            a = FlowLayout.CENTER;
            break;
        case FlowLayout.CENTER:
            a = FlowLayout.RIGHT;
            break;
        case FlowLayout.RIGHT:
            a = FlowLayout.LEFT;
    }
    fl.setAlignment(a);
    centerPanel.revalidate();
}
```



see: examples.swing.miscFlowLayoutExample_1

Dynamic Change of Layout - Ex. I

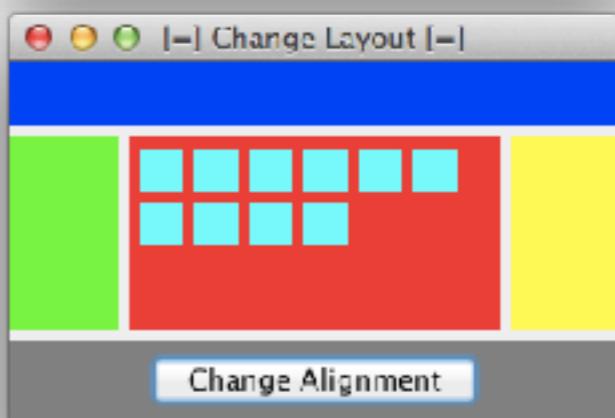
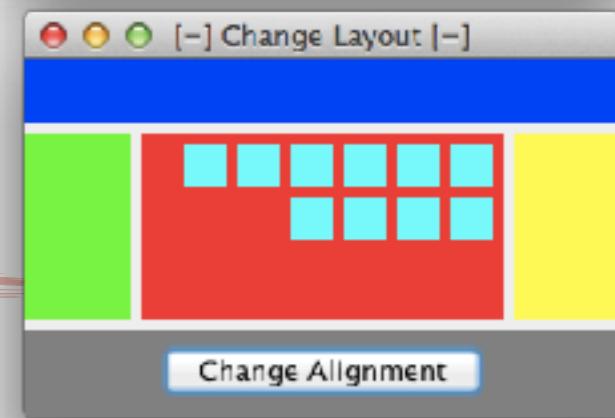
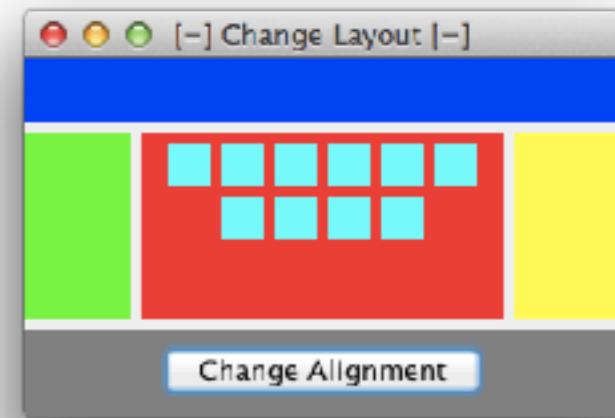
```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
    ...  
  
    for( int i=0; i<10; i++ )  
        centerPanel.add( createPanel( Color.cyan, 20, 20 ) );  
  
    alignButton = new JButton( "Change Alignment" );  
    alignButton.addActionListener(this);  
    botPanel.add( alignButton );  
    ...  
}  
  
public void actionPerformed(ActionEvent arg0) {  
    FlowLayout fl = ((FlowLayout)centerPanel.getLayout());  
    int a = fl.setAlignment();  
    switch ( a ) {  
        case FlowLayout.LEFT:  
            a = FlowLayout.CENTER;  
            break;  
        case FlowLayout.CENTER:  
            a = FlowLayout.RIGHT;  
            break;  
        case FlowLayout.RIGHT:  
            a = FlowLayout.LEFT;  
    }  
    fl.setAlignment(a);  
    centerPanel.revalidate();  
}
```



see: examples.swing.miscFlowLayoutExample_1

Dynamic Change of Layout - Ex. I

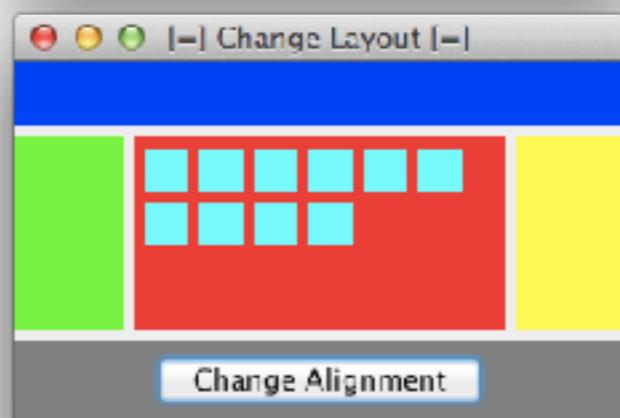
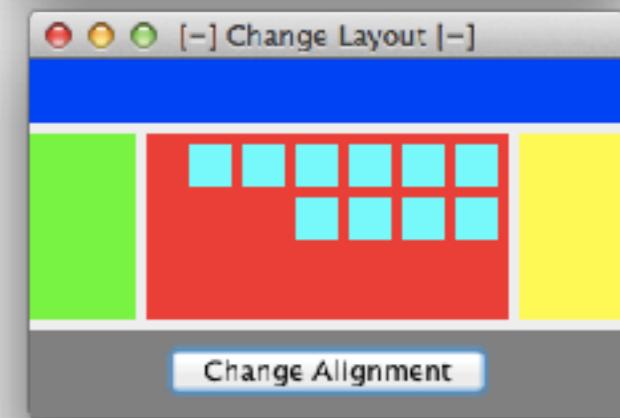
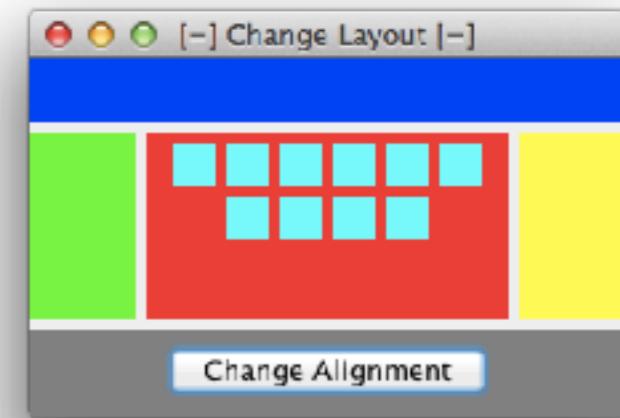
```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
    ...  
  
    for( int i=0; i<10; i++ )  
        centerPanel.add( createPanel( Color.cyan, 20, 20 ) );  
  
    alignButton = new JButton( "Change Alignment" );  
    alignButton.addActionListener(this);  
    botPanel.add( alignButton );  
    ...  
}  
  
public void actionPerformed(ActionEvent arg0) {  
    FlowLayout fl = ((FlowLayout)centerPanel.getLayout());  
    int a = fl.setAlignment();  
    switch ( a ) {  
        case FlowLayout.LEFT:  
            a = FlowLayout.CENTER;  
            break;  
        case FlowLayout.CENTER:  
            a = FlowLayout.RIGHT;  
            break;  
        case FlowLayout.RIGHT:  
            a = FlowLayout.LEFT;  
    }  
    fl.setAlignment(a);  
    centerPanel.revalidate();  
}
```



see: examples.swing.miscFlowLayoutExample_1

Dynamic Change of Layout - Ex. I

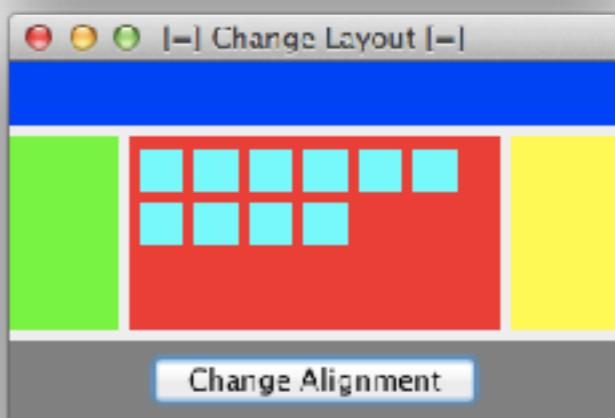
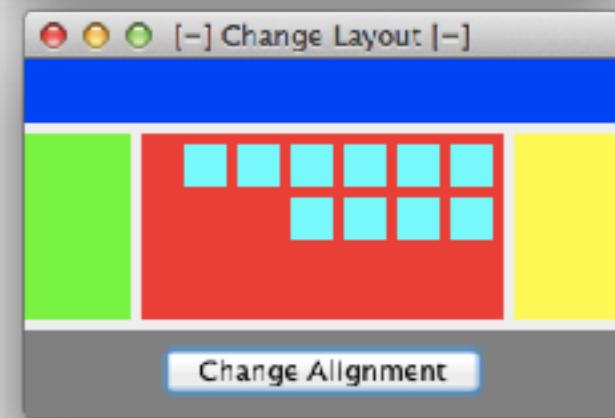
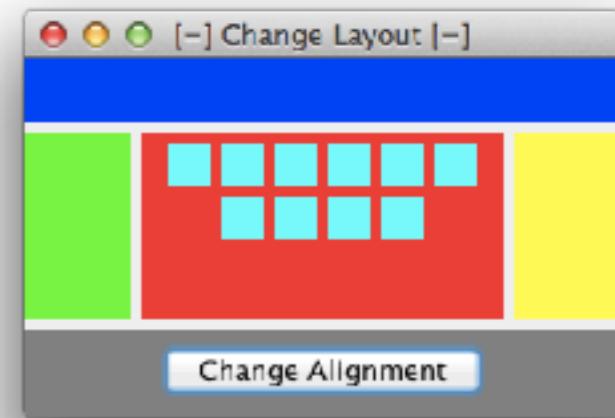
```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
    ...  
  
    for( int i=0; i<10; i++ )  
        centerPanel.add( createPanel( Color.cyan, 20, 20 ) );  
  
    alignButton = new JButton( "Change Alignment" );  
    alignButton.addActionListener(this);  
    botPanel.add( alignButton );  
    ...  
}  
  
public void actionPerformed(ActionEvent arg0) {  
    FlowLayout fl = ((FlowLayout)centerPanel.getLayout());  
    int a = fl.setAlignment();  
    switch ( a ) {  
        case FlowLayout.LEFT:  
            a = FlowLayout.CENTER;  
            break;  
        case FlowLayout.CENTER:  
            a = FlowLayout.RIGHT;  
            break;  
        case FlowLayout.RIGHT:  
            a = FlowLayout.LEFT;  
    }  
    fl.setAlignment(a);  
    centerPanel.revalidate();  
}
```



see: examples.swing.miscFlowLayoutExample_1

Dynamic Change of Layout - Ex. I

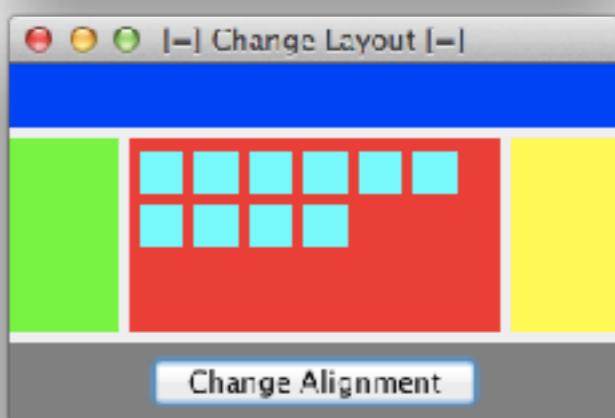
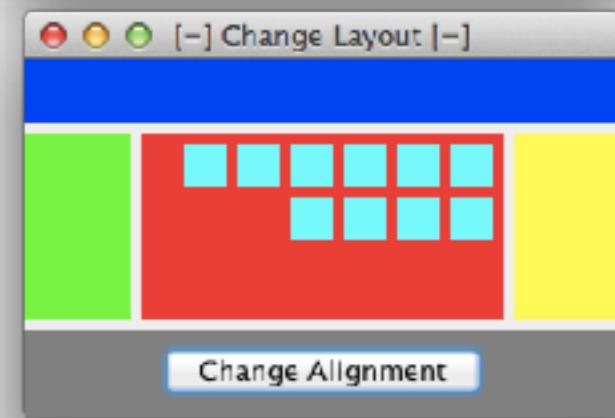
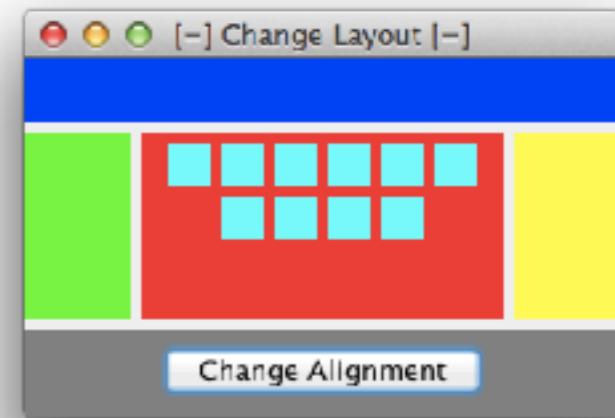
```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
    ...  
  
    for( int i=0; i<10; i++ )  
        centerPanel.add( createPanel( Color.cyan, 20, 20 ) );  
  
    alignButton = new JButton( "Change Alignment" );  
    alignButton.addActionListener(this);  
    botPanel.add( alignButton );  
    ...  
}  
  
public void actionPerformed(ActionEvent arg0) {  
    FlowLayout fl = ((FlowLayout)centerPanel.getLayout());  
    int a = fl.setAlignment();  
    switch ( a ) {  
        case FlowLayout.LEFT:  
            a = FlowLayout.CENTER;  
            break;  
        case FlowLayout.CENTER:  
            a = FlowLayout.RIGHT;  
            break;  
        case FlowLayout.RIGHT:  
            a = FlowLayout.LEFT;  
    }  
    fl.setAlignment(a);  
    centerPanel.revalidate();  
}
```



see: examples.swing.miscFlowLayoutExample_1

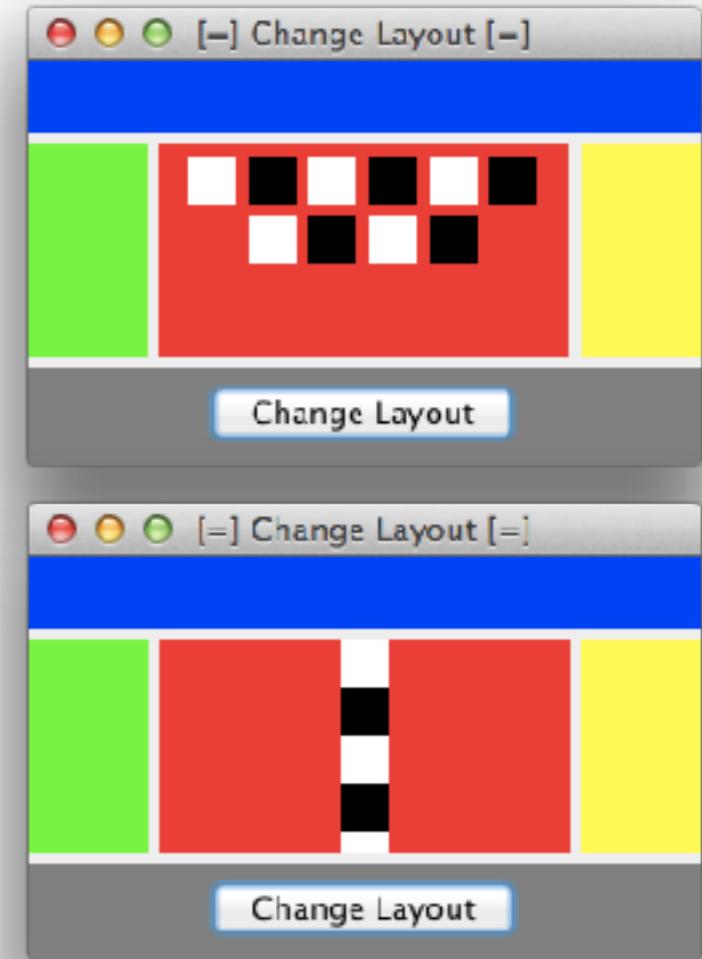
Dynamic Change of Layout - Ex. I

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
    ...  
  
    for( int i=0; i<10; i++ )  
        centerPanel.add( createPanel( Color.cyan, 20, 20 ) );  
  
    alignButton = new JButton( "Change Alignment" );  
    alignButton.addActionListener(this);  
    botPanel.add( alignButton );  
    ...  
}  
  
public void actionPerformed(ActionEvent arg0) {  
    FlowLayout fl = ((FlowLayout)centerPanel.getLayout());  
    int a = fl.setAlignment();  
    switch ( a ) {  
        case FlowLayout.LEFT:  
            a = FlowLayout.CENTER;  
            break;  
        case FlowLayout.CENTER:  
            a = FlowLayout.RIGHT;  
            break;  
        case FlowLayout.RIGHT:  
            a = FlowLayout.LEFT;  
    }  
    fl.setAlignment(a);  
    centerPanel.revalidate();  
}
```



Dynamic Change of Layout - Ex. II

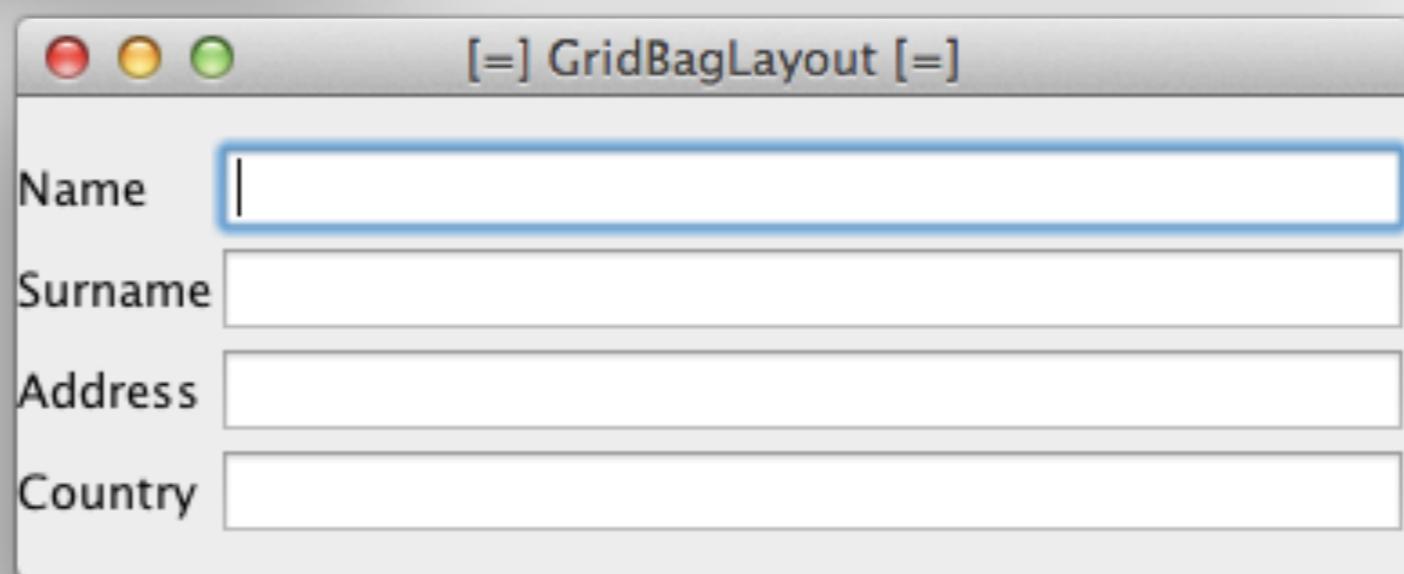
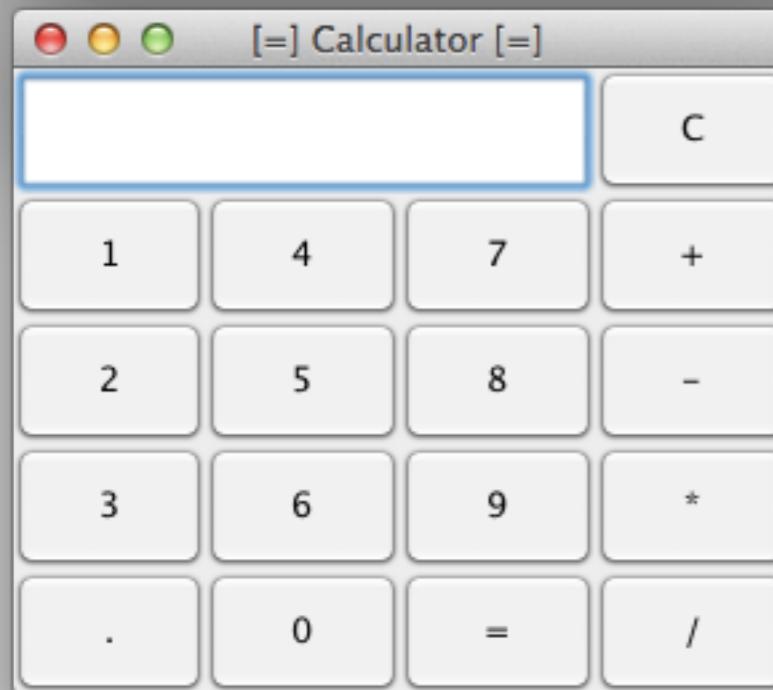
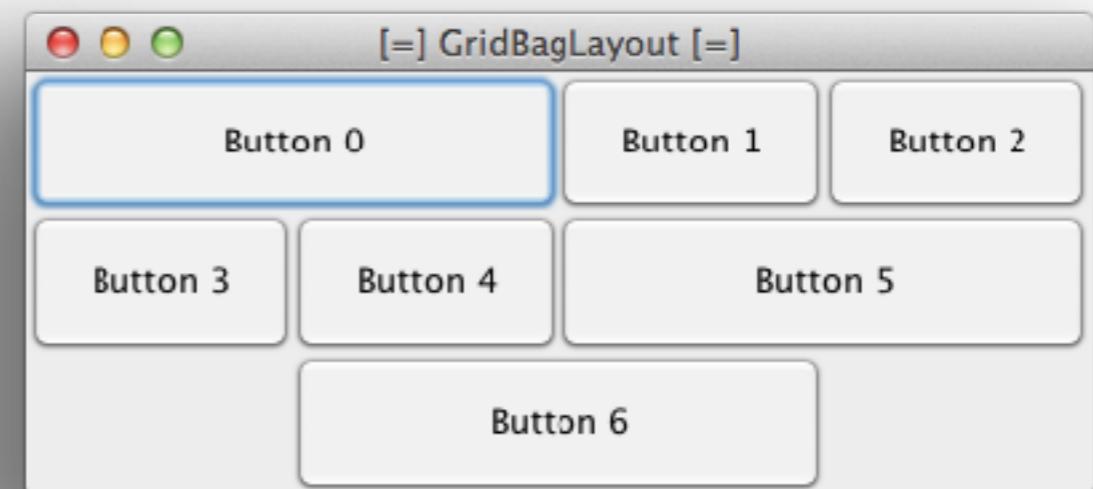
```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
    ...  
  
    for( int i=0; i<5; i++ ) {  
        centerPanel.add( createPanel( Color.white, 20, 20 ) );  
        centerPanel.add( createPanel( Color.black, 20, 20 ) );  
    }  
  
    layoutButton = new JButton( "Change Layout" );  
    layoutButton.addActionListener(this);  
    botPanel.add( alignButton );  
    ...  
  
    public void actionPerformed(ActionEvent arg0) {  
        if (lastLayout == 0) {  
            centerPanel.setLayout(  
                new BoxLayout(centerPanel, BoxLayout.Y_AXIS));  
            lastLayout = 1;  
        } else {  
            centerPanel.setLayout(  
                new FlowLayout(FlowLayout.CENTER));  
            lastLayout = 0;  
        }  
        centerPanel.revalidate();  
    }  
}
```



It is not something that we typically want to do, so be careful as you might get some unexpected behavior.

Exercises

Construct the following GUIs using **GridBagLayout**



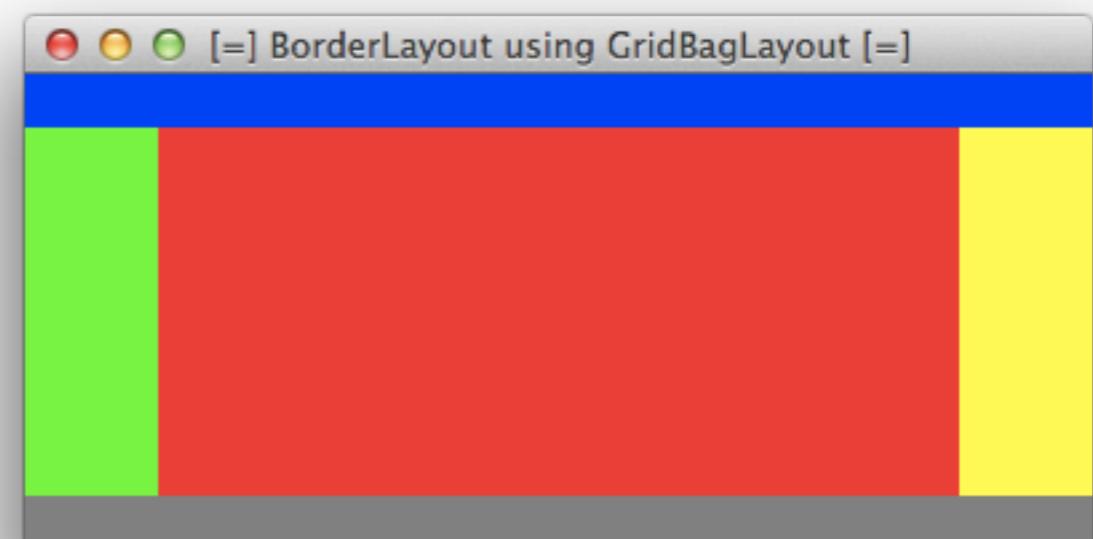
see: examples.swing.misc.GridBagLayout_5,...,examples.swing.misc.GridBagLayout_8

Exercise

Implement a class **BLPanel** to behave like **JPanel** with **BorderLayout**, using only **GridBagLayout**

```
public class BLJPanel extends JPanel {  
    final static int LEFT = 0;  
    final static int CENTER = 1;  
    final static int RIGHT = 2;  
    final static int TOP = 3;  
    final static int BOT = 4;  
  
    GridBagLayout lm;  
  
    public BLJPanel() {  
        lm = new GridBagLayout();  
        this.setLayout(lm);  
    }  
  
    public void add(JComponent x, int pos) {  
        ...  
    }  
}
```

```
BLJPanel totalGUI = new BLJPanel();  
  
center = createPanel(Color.red, 50, 50);  
totalGUI.add(center, BLJPanel.CENTER);  
  
left = createPanel(Color.green, 50, 50);  
totalGUI.add(left, BLJPanel.LEFT);  
  
...
```



see: examples.swing.misc.BLJPabel, examples.swing.misc.BLJPabelExample