

GUIs in Java



2-24-2005

Opening Discussion



- Do you have any questions on the quiz?
- What did we talk about last class? Do you have code to show?
- Do you have any questions about the assignment?
- Private iterator? Writing the Iterator loop?
- GUIs are great examples of the object model. What types of objects can you think of in a GUI? Are there any places where inheritance and polymorphism come into play?

A Tale of Two Libraries



- Java actually has two GUI libraries in it. The original one is the Abstract Windowing Toolkit (AWT). This can still be used and might provide better performance for some things.
- On top of AWT they built Swing to be more platform independent. Swing also has some more powerful classes that you can use.
- Go to the docs and look at them.

Components and Containers



- In AWT, GUIs are built from components. Each individual element of a GUI will be some subclass of the Component class.
- One subclass of Component is Container which represents a component that can hold other components.
- In your code you are likely to use the JPanel class as a container and a basic component.

Layout Managers



- While you can explicitly place components in containers to build your GUIs, that is not the recommended method in Java. You should use layout managers instead to make your GUIs more flexible and portable.
- You can set the layout used by any container and nest containers inside containers to create complex GUIs.

Basic Layout Managers

- There are a handful of `LayoutManager` subclasses provided with AWT and Swing. These are the most basic ones.
 - `FlowLayout`: puts one component after another using the preferred size of the component and wraps like text.
 - `GridLayout`: simple 2D grid of equal sized components.
 - `BorderLayout`: north and south get what they need followed by east and west. Center gets the rest.

Other Layout Managers



- Here are a few more:
 - `CardLayout`: used for quickly flipping between different views.
 - `GridBagLayout`: very powerful but rather complex layout with the ability to place things quite freely and give borders and separations between them.
 - Others: `BoxLayout`, `OverlayLayout`, `ScrollPaneLayout`

Easier in Swing



- Swing has some custom panels that use their own layouts that can be nice.
 - JScrollPane is nice when you have something bigger than the screen to display. Embed it in this and you can scroll around it.
 - JTabbedPane gives the functionality you typically want a card layout for.
 - JSplitPane allows you to have two panels where you can move the dividing line between them.
 - Box is a panel with a BorderLayout.

Frames and Dialogs



- Of course, your GUI components need to go inside a window. These are represented by Frames and Dialogs (or JFrames and JDialogs). These subclasses of the Window class represent windows and dialog boxes and act as containers.
- In your game you might bring these up to prompt the user for extra information.
- JoptionPane can be used for simple stuff.

Menus



- The last separate piece of the GUI puzzle is the ability to add menus.
- A MenuBar holds the Menu objects that you can add to it. Each Menu can hold several MenuItem objects.
- Java also allows you to put in popup menus in your applications.
- GameSetup has a function that returns the menu for the frame. You can edit it.

Code



- Let's go and look at how the GUIs are set up for our semester example.

Minute Essay



- I want a GUI with 5 JTextFields where each has a JLabel next to it. How could you lay that out?
- Remember to turn in assignment #3 today.