Interactive GUIs

3-21-2012

Opening Discussion

- Minute essay comments:
  - Requires graphics:
    - Most games.
    - Schedule maker.
  - You will have the ability to write full programs operating from the GUI.
  - There will be a cheat sheet for the final. API access as well.
  - GPA calculator, or basic calculator.
  - Did we need to put Exit in the menu?
  - Spurs trade of RJ for SJ.

More

- GUIs vs. web apps.
- Complimentary majors to a CS minor.
- GUIs easier?
- Importing scala.swing vs. scala.swing._
- Details of types in API.

Panes

- ScrollPane
  - Holds a single component passed in as an argument at construction. Scroll bars automatic.
- SplitPane
  - Two components separated by a moveable bar.
  - new SplitPane(Orientation.Horizontal, leftComp,rightComp)
- TabbedPane
  - One component shown at a time. Tabs are always shown. Add components by adding Pages to the page object.
  - pages += new Page(“A Tab”,tabComponent)

Menus

- The frame has a MenuBar.
- Contents of a MenuBar should be Menus.
- Menus can hold the following:
  - MenuItem
  - Menu
  - CheckMenuItem
  - RadioMenuItem
  - Separator

Example GUI

- We want to have a simple GUI that has a few components for us to play with.

Interactive GUIs

- Last time we learned how to build a GUI with components.
- Our GUI wasn't interactive though.
- Buttons and MenuItems are easy because we give them an Action.
- Simple read model of console apps is insufficient.
- Need a way to deal with input from many sources.

Publisher/Reactor

- The basic model employed by scala.swing is Publishers and Reactors.
- A Reactor can listenTo events from a Publisher.
- Events are in scala.swing.event.
- To stop listening, set yourself to be deafTo.
- The Publisher can be the component or an object in the component.

Partial Functions

- Scala has a construct called a partial function. It is a function that only works on some inputs.
- The brief syntax for them is like a match with no match. So it has curly braces with cases in them.
  - {
    - case 1 => doOption1()
    - case 2 => doOption2()
    - ...
  - }

Events/reactions

- When a Reactor is listening to a Publisher, it needs to define reactions.
- You can add PartialFunctions into the reactions of the Reactor.
- They should respond to the events you are interested in.
  - reactions += {
    - case e:ActionEvent => ...
    - case e:SelectionChanged => ...
  - }

Making the GUI Interactive

- Let's use this new knowledge to make it so that our GUI interacts with the user.

Minute Essay

- Questions?
- Remember to turn in Assignment #2 by tonight.