Opening Discussion

- Questions on the assignment?
- Finishing the example from last class.
Motivation

- While text based programs still play a very big role in computing, it is mostly behind the scenes.
- You are far more used to working with Graphical User Interfaces (GUIs).
- It is time that we learn how to write GUIs in Scala.
There are three libraries that will wind up being relevant to our discussion.

- javax.swing – Swing was built on top of AWT to be more flexible.
- scala.swing – Scala code wrapped around Java Swing to aid Scala GUI programming.
In order to write a GUI we need to start by popping up a window.

For the main window of a GUI, we will make a MainFrame. For other windows there are Frame and Dialog types.

We can set the title and size fields of the MainFrame when we create it.

Set visible to true to bring up the window.

Oddly, we have to prevent the script from stopping.
Active Components

- GUIs are made from components. Use scala.swing package.
  - Button(text:String)(action : => Unit).
  - new CheckBox(label:String)
    - selected:Boolean
  - new ComboBox(items:Seq[A])
    - selection.index to get the index of the current selection
  - new EditorPane(contentType:String,text:String)
More Components

- new FormattedTextField(format:String)
  - text:String that will tell you the text
- new Label(text:String)
- new ListView(items:Seq[A])
  - Use collection selection.indices to interact with the index values that are selected.
- new PasswordField or new PasswordField(text:String)
  - text:String will tell you the text
More Components

- new ProgressBar
  - min: Int, max: Int, and value: Int
- new RadioButton(text: String)
  - selected: Boolean
- new ScrollBar
  - minimum, maximum, and value are all Ints
  - Generally use ScrollPane
- new Slider
  - min, max, value
  - orientation
Still More Components

- new Table(rowData: Array[Array[Any]], columnNames: Seq[Any])
- new TextArea(text: String)
  - text: String
- new TextField(text: String)
  - text: String
We build complex GUIs by nesting panels and panes.

- **BorderPanel**
  - Can hold up to five different components in the north, south, east, west, and center positions. Add to the layout as a tuple of (Component, Position).

- **BoxPanel**
  - Can hold a number of components either vertically or horizontally, each takes the space it needs. Use new BoxPanel(Orientation.Vertical). Use contents+=Button(“text”)(action).
More Panels

- **FlowPanel**
  - Components are laid out from left to right wrapping like text in a word processor. You can pass a variable length list of components as an argument at construction or add the components to contents.

- **GridBagPanel**
  - This panel is more complex.

- **GridPanel**
  - Holds a regular grid of components. You specify how many rows and columns the grid has at creation.
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

- Windows can set the MenuBar.
- Add Menu objects to the contents of the MenuBar.
- Add MenuItems to the contents of the Menus.
  - `new MenuItem(Action("Exit"){ exit(0) })`
Let's spend the rest of class laying out and coding up a GUI we could use to view players and edit teams.
Is there some type of GUI you would like to have as IcP #6.

Assignment #2 is due on Wednesday.