

### Administrivia

- Homework 5 code due tomorrow. Questions?

Slide 1

### GUIs in Java — Recap of Main Ideas

- Components and containers.
- Layout — grouping components in containers, layout managers. Components have minimum and preferred sizes; layout managers readjust sizes differently.
- Event handling — listener interfaces, “add listener” methods.
- Painting — predefined components do for you, or you can roll your own using graphics classes.
- Threads — more later, but be aware this is an issue.

Slide 3

### Minute Essay From Last Lecture

- The game framework will allow you to add panels to any or all four sides of your game. You can display info (text is easiest) or include GUI components for additional user input (e.g., click a button to speed up the player). You can also add to the menu bar. How might this be helpful for your game?  
(Almost everyone had some good ideas!)

Slide 2

### GUI-Based Programs in Java

- A common structure for the main class of a simple program is to extend `JFrame`, put all/most setup code in constructor, have `main` method just create an instance and call `setVisible`. Call `pack` to set everything to minimum sizes.
- Usually need instance variables to represent persistent data — anything needed by more than one method (including ones to handle events). This includes components needed by more than one method.  
If data becomes complex, considering making a separate class for it — “Model / View / Controller” paradigm.
- There are some compiler pitfalls with variables used in anonymous inner classes — if local, may have to declare `final`.

Slide 4

### GUIs and the Project

- Overall layout of game is `BorderLayout`, with screen in middle and "game status panels" on four sides — returned by `getGameStatusPanel` (in `Player`), usually a `JPanel`.
- Menu bar is in `GameSetup`, can be modified.
- Screen editor program has support for "editing properties" (of screens, blocks, entities) — `getEditPropertiesPanel`. Could use this to give slightly different properties to different instances (e.g., walls of different colors, enemies with different speeds).
- Homework 6 will ask you to use these features to (1) display something, and (2) get input from the user (either in the game or in the screen editor).

Slide 5

### Minute Essay

- Try writing code for a program that works as follows:  
Displays a text field the user can type into, an area to display the text, and a "convert" button.  
When the button is pressed, converts the text in the text field to uppercase and displays it.

Slide 6