Slide 1

Administrivia

- Reminder: Homework 6 design due Thursday.
- Quiz 5 moved to next Tuesday. Likely topic is GUIs/graphics.

Slide 2

Custom GUI Components, Recap/Review

- “Custom components” allow you to define what’s drawn, etc.
- Put code to draw what you want in `paintComponent` method; call `repaint` to regenerate image.
- Parameter to `paintComponent` is a `Graphics` object — a “graphics context”, or something you can draw on. As you know from your game, you can also get one of these for an `Image` object.
- So, you can look at the `Graphics` class . . . However, the object is really a `Graphics2D` object, and `Graphics2D` has a lot more functionality.

Slide 3

Drawing and Filling Shapes

- “Draw” means draw outline only; “fill” to draw and fill.
- `Graphics` provides methods for doing simple shapes. `Graphics2D` provides more general methods. (Look at some shapes in `java.awt.geom`.)
- Use these to finish example from last time . . .

Slide 4

Drawing and Filling Shapes, Continued

- You already know (from your game) about simple way to control color of what’s painted.
- The `Graphics2D` class provides a lot more options. You can use:
  - `setPaint` to fill shapes with simple color, gradient fill, etc.
  - `setStroke` to draw outlines with different widths, etc.
  - `setFont` to draw text in different fonts. (This works for text components such as `JLabel` too.)
- And there’s more — “clipping”, affine transformations (e.g., rotation — transformations in which parallel lines stay parallel), etc., etc.
- Look at another example . . .

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 asks 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

- In the first example shown today, I kept a list of shapes to be drawn and redrew everything in `paintComponent`. Can you think of another approach that wouldn't involve keeping a list of shapes? (Hint: Consider what can be done with a `BufferedImage`.)

Slide 6

Minute Essay Answer

- It would probably work to maintain a `BufferedImage` on which the shapes are drawn, and then in `paintComponent` combine that with the bounding ellipse/rectangle showing the outline of the shape being created. (I haven't tried it, though.)

Slide 7