

## Objectives

- GUIs in Java
- Event Handling
- Animation

Nov 7, 2008

Sprenkle - CS209

1

## Discussion of Roulette Assignment

- How easy/difficult to refactor for extensibility?
- Was it easier to add to your refactored code?
  - What would your refactored classes have looked like if I hadn't told you that you were going to add the three other bets?
- How easy/difficult was it to test your classes?

Nov 7, 2008

Sprenkle - CS209

2

## GUI Review

- What is the purpose of a Layout Manager?
- Describe two different layout managers
- How can we create a customized layout?
- What are the parts of event handling?

Nov 7, 2008

Sprenkle - CS209

3

## Review: Fonts

- Bold & Italic

```
Font sansbold14 = new Font("SansSerif",
    Font.BOLD + Font.ITALIC, 14);
```

Nov 7, 2008

Sprenkle - CS209

4

## Review: Event-Driven Programming

- Flow of program is determined by user actions (e.g., mouse clicks, key presses), sensor outputs, or messages from other applications
- Application architecture:

```
while ( true ) {
    event = waitForEvent();
    handleEvent(event);
}
```

Nov 7, 2008

Sprenkle - CS209

5

## Review: Event Basics

- An **event** is generated from an **event source** and is transmitted to an **event listener**
- Event sources allow event listeners to register with them
  - Means: registered listener requests event source sends its event to listener when event occurs
- All events are objects of event classes, which derive from `java.util.EventObject`

Nov 7, 2008

Sprenkle - CS209

6

## Review: Anonymous Class Listener

```
void makeButton(String label, final Color bgColor) {
    JButton button = new JButton(label);
    add(button);
    button.addActionListener( new ActionListener() {
        public void actionPerformed(ActionEvent evt)
        {
            setBackground(bgColor);
            repaint();
        }
    });
}
```

Nov 7, 2008

Sprenkle - CS209

7

## Anonymous Inner Classes

- Confusing syntax!
- Create a new class that implements `ActionListener` interface
  - Define required method, `actionPerformed()`, inside braces
- Any needed parameters are inside the parentheses, following the *supertype* name:

```
new SuperType(construction parameters) {
    inner class methods and data
};
```

Nov 7, 2008

Sprenkle - CS209

8

## Anonymous Inner Classes

- *Supertype* can be an *interface* or a *class*
  - If an interface, inner class implements the interface and required methods
  - If a class, the inner class extends that class
- Anonymous inner classes do **not** have *constructors*
  - Parameters are passed to *superclass's* constructor
  - If inner class implements an interface, rather than extend a class, you cannot have construction parameters

Nov 7, 2008

Sprenkle - CS209

9

## Anonymous Inner Classes

- Carefully differentiate between
  - Construction of a new object of a class
  - Construction of an object of an anonymous inner class that extends that class...

```
// this is a Person object
Person queen = new Person("Mary");

// this is an object of an anonymous
// inner class extending the Person class
Person count = new Person("Dracula") { . . . };
```

Nov 7, 2008

Sprenkle - CS209

10

## GRAPHICS PROGRAMMING

Nov 7, 2008

Sprenkle - CS209

11

## Graphics Object

- Abstract class
  - Implementation different for each platform
- A collection of settings for drawing images and text, such as colors and fonts
- Where used:
  - `paintComponent(Graphics g)`

Nov 7, 2008

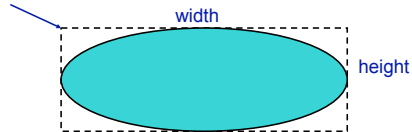
Sprenkle - CS209

12

## Drawing Lines, Rectangles, Ovals

- Draw ovals, rounded rectangles within bounding rectangle

Starting Position of oval



- Can also draw arcs, polygons, polylines

Nov 7, 2008

Sprenkle - CS209

13

## Colors

- Colors made up of three components
  - Red, Green, Blue component
  - RGB values
    - Components: either 0 to 255 or 0.0 to 1.0
- Thirteen predefined colors in the `Color` class:
  - black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, and yellow
  - Also defined in all caps
  - See API

`ColorJPanel.java`

Nov 7, 2008

Sprenkle - CS209

14

## CODE REVIEWS

Nov 7, 2008

Sprenkle - CS209

15

## Understanding Code

- Goblin Game
  - How draws
  - How does event handling
  - How animates
- Bouncers
  - How draws
  - How animates

Nov 7, 2008

Sprenkle - CS209

16

## Understanding Code: Screen Savers

- In Eclipse, import an existing project:
  - `/home/courses/cs209/handouts/screensavers.tar`
- Run `Main` class (which `Main` class?)
- Answer questions about code
  - What represents an object in the screen saver?
  - How generates screen saver objects?
  - How handles animation?
  - How handles events?

Nov 7, 2008

Sprenkle - CS209

17

## Assignment 12

- Complete screen savers for
  - Racers
  - Random Walkers
  - "Attractors"
- Due Wednesday
- Heads up: Exam next Friday
  - All about design principles (tradeoffs), GUIs
  - Terminology

Nov 7, 2008

Sprenkle - CS209

18