Lecture 4: Graphics and Lists

CS 62
Fall 2018
Alexandra Papoutsaki & William Devanny
Graphics

• Create objects you want to draw:
  • `Rectangle2D.Double`, `Line.Double`, etc.
  • Constructors take x,y coords and dimensions, but don’t actually draw items.

• All drawing takes place in `paint` method using a “graphics context”
  • an object you can use to draw graphics primitives

• Triggered implicitly by uncovering window or explicitly by calling `repaint` method.
  • Adds `repaint` event to event queue – eventually draws it
Graphics context

• All drawing is done in `paint` method of component

• `public void paint(Graphics g)`
  • `g` is a Graphics context provided by system
  • “pen” that does the drawing
  • Programmer calls `repaint()`, not `paint()`!!

• Need to import classes from `java.awt.*`, `java.geom.*`, `javax.swing.*`

• See `MyGraphicsDemo`
General Graphics Applications

- Create an extension of component (either JPanel, JFrame, or JApplet) and implement the `paint` method in the subclass.
  - See main method of demo to get window to show
  - At start of paint method cast `g` to Graphics2D to get access to new methods

- Call `repaint()` on component every time you make a change.
  - Causes OS to schedule call of paint in event queue
  - Called automatically if window obscured and revealed
Geometric Objects

- Objects from classes `Rectangle2D.Double`, `Line2D.Double`, etc. from `java.awt.geom`
  - There are also float versions
  - Constructors take params x, y, width, height, but don’t draw object

- `Rectangle2D.Double`
- `RoundRectangle2D.Double`
- `Ellipse2D.Double`
- `Arc2D.Double`
- `Line2D.Double`, ...
java.awt.Color
Methods

- `myObj.setFrame(x, y, width, height)`: can move object
- `g2.draw(myObj)`: gives outline
- `g2.fill(myObj)`: gives filled version
- `g2.drawString("a string", x, y)`: draws string
MyGraphicsDemo

• Class extends JFrame, which creates window.

• Constructor calls super with title of window.

• main method creates object, sets size, visibility, and enables go-away box.

• paint method creates and draws objects.
BorderLayout
PostItApplication

• More sophisticated.

• JFrame contains two JPanels.

• JFrame uses BorderLayout, so add controls to JPanel in SOUTH, drawing canvas in CENTER of the JFrame.

• DrawingCanvas extends JPanel -- contains paint method
  • Note use of ArrayList to hold PostIt.
PostIt

• Represents the rectangles being dragged:
  • Contains getter(accessor) and setter(mutator) methods to allow it to be manipulated by drawing program.
  • Could add features (title bar, go-away box) without affecting PostItApplication code.
PostItApplication

- **PostItApplication** class responsible for
  - setting up the GUI
  - Responding to button pressed and menu selections
  - Sets up **ArrayList** of items on canvas.

- Class has 3 inner classes
  - **DrawingCanvas**
  - **DrawingMouseListener**
  - **DrawingMouseMotionListener**
  - *Inner classes have access to private features of containing class*
List Operations

- Review list operations from library interface `List` in Java 8 documentation.
  - Bailey’s List is slightly different.

- Think about how to implement with array.

- `size`, `isEmpty`, `get`, `set` functions
ArrayList

• See Bailey’s **ArrayList**
  • Similar to Java 8’s **ArrayList**
  • Instance variables:
    • `elts`: array instance variable,
    • `eltsFilled`: number of slots filled.

• Some operations very cheap:
  • `size`, `isEmpty`, `get`, `set` take constant time (no search)

• Others more expensive
Lab and Assignment 1

• Strip with 12 squares and 5 silver dollars placed randomly on the board.
• Move silver dollars to fill 5 leftmost squares
• Coins move only to the left.
• No coin may pass another.
• No square may hold more than one coin.
• Last person to move wins.
• Complete description in textbook.
Arrays

- `int arr[] = new int[10]`
- Hold a sequence of primitives or objects.
- Public instance variable `length`
- Fixed length
- Don’t play nice with generics
ArrayList

- `import java.util.ArrayList`
- `class ArrayList<E> implements List<E>`
- Important methods:
  - `add`, `get`, `set`, `indexOf`, `isEmpty`, `remove`, `size`, `contains`, `clear`
    - `size`, `isEmpty`, `get`, `set` \(\rightarrow\) constant time
    - `add(E e)` \(\rightarrow\) "amortized constant" time

- See javadoc at: [https://docs.oracle.com/javase/8/docs/api/](https://docs.oracle.com/javase/8/docs/api/)
- Text uses `Vector` instead of `ArrayList`.
  - `ArrayList` more efficient if no concurrency