Programming Languages and Techniques (CIS120)

Lecture 34

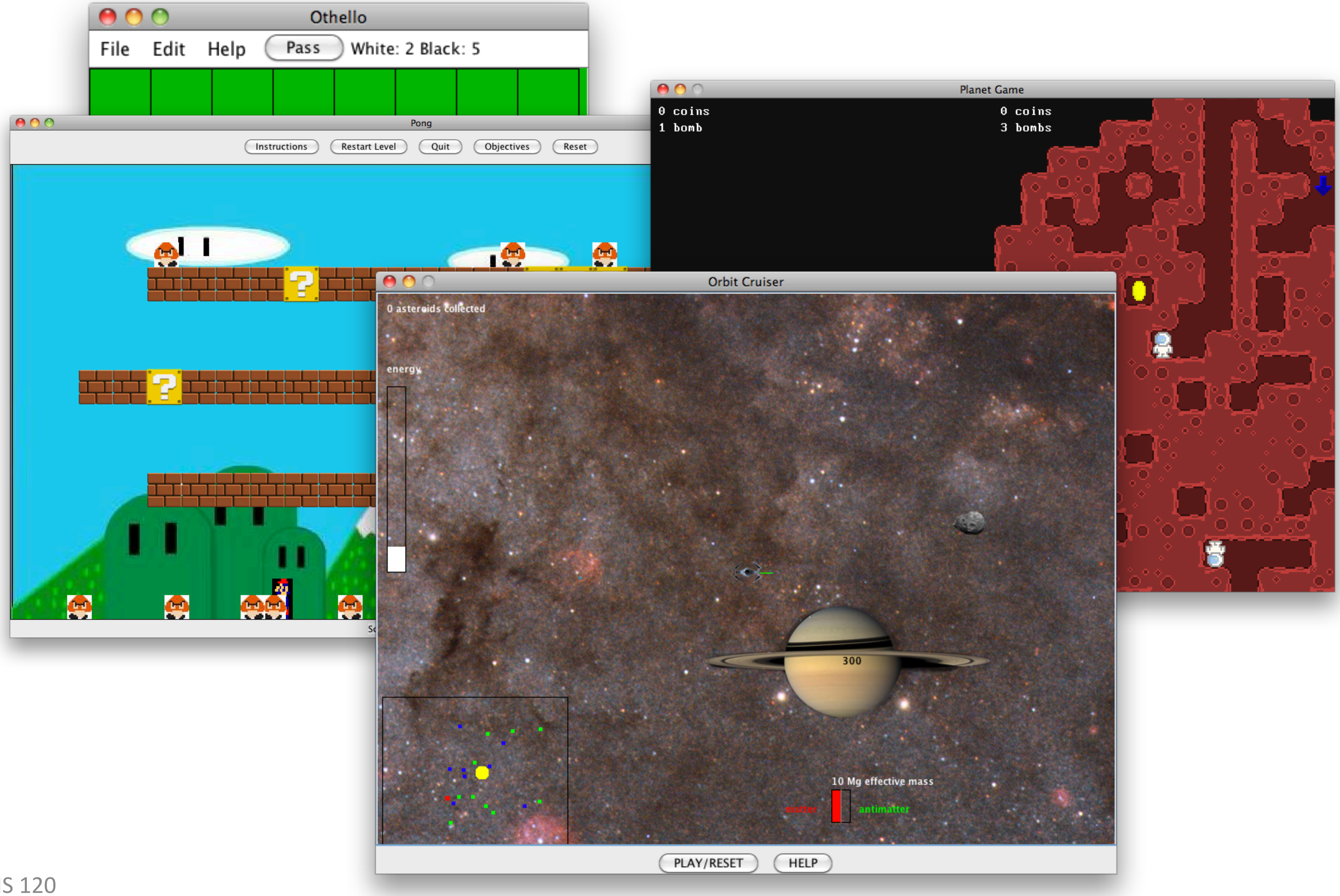
November 23, 2015

Swing II: Inner Classes and Layout

Announcements

- HW8: Spellchecker
  - Due: *TOMORROW* Tuesday, November 24th
  - Parsing, working with I/O, more practice with collections
- This Week: No Lab Sections
- Wednesday: Bonus Lecture "Consequences of Code as Data"
  - Attendance not required (but encouraged if you are around!)

HW9: Game Project Available Soon



Game project grading

- Game Design Proposal Milestone Due: (12 points)
  Tuesday December 1st at 11:59pm
  - (Should take about 1 hour)
- Final Program Due: (88 points)
  Tuesday December 8th at 11:59pm
  - Submit zipfile online, submission *only* checks if your code compiles
- Grade based on demo with your TA during reading days
  - Make sure that you test your program in Moore 100, especially if you use outside libraries
  - Grading rubric on the assignment website
  - Recommendation: don't be too ambitious.
- *NO LATE SUBMISSIONS PERMITTED*

Inner Classes

Anonymous Inner Classes

- Define a class and create an object from it all at once, inside a method

```
quit.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
       System.exit(0);
   }
 });
```

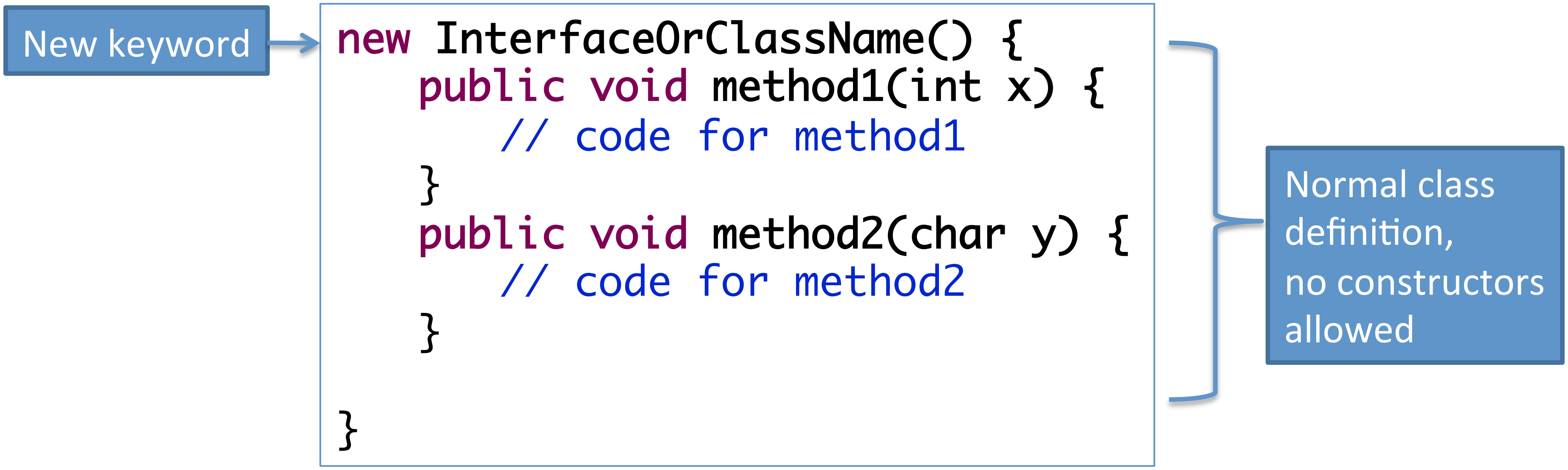
Puts button action right with button definition

```
line.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
       shapes.add(new Line(…));
       canvas.repaint();
   }
});
```

Can access fields and methods of outer class, as well as final local variables

Anonymous Inner class

- New *expression* form: define a class and create an object from it all at once



New keyword

```
new InterfaceOrClassName() {
    public void method1(int x) {
        // code for method1
    }
    public void method2(char y) {
        // code for method2
    }

}
```

Normal class definition, no constructors allowed

Static type of the expression is the Interface/superclass used to create it

Dynamic class of the created object is anonymous! Can't refer to it.

Like first-class functions

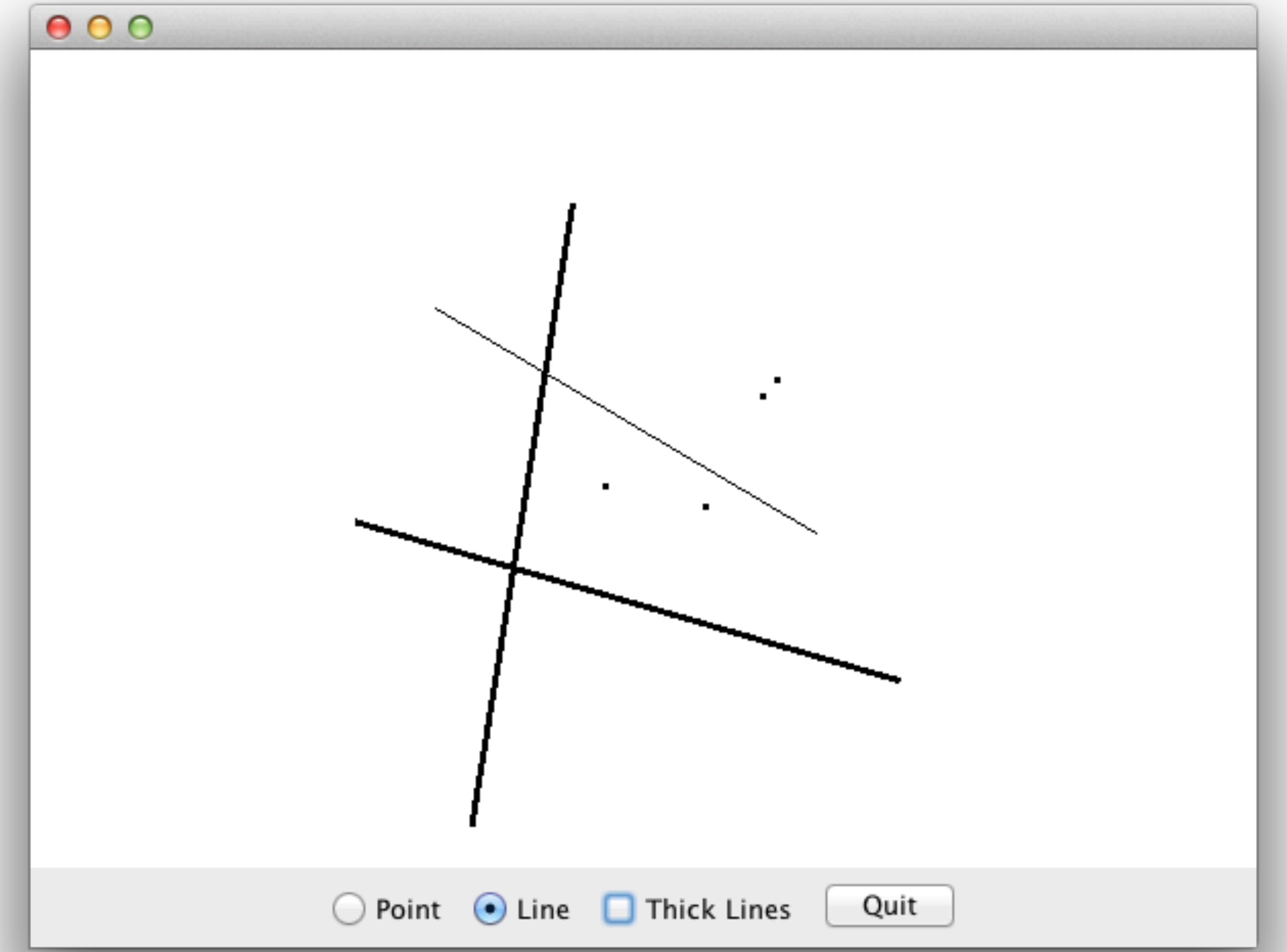
- Anonymous inner classes are the real Java equivalent of Ocaml first-class functions
- Both create "delayed computation" that can be stored in a data structure and run later
  - Code stored by the event / action listener
  - Code only runs when the button is pressed
  - Could run once, many times, or not at all
- Both sorts of computation can refer to variables in the current scope
  - OCaml: Any available variable
  - Java: only instance variables (fields) and variables marked final

Did you attend lecture today?

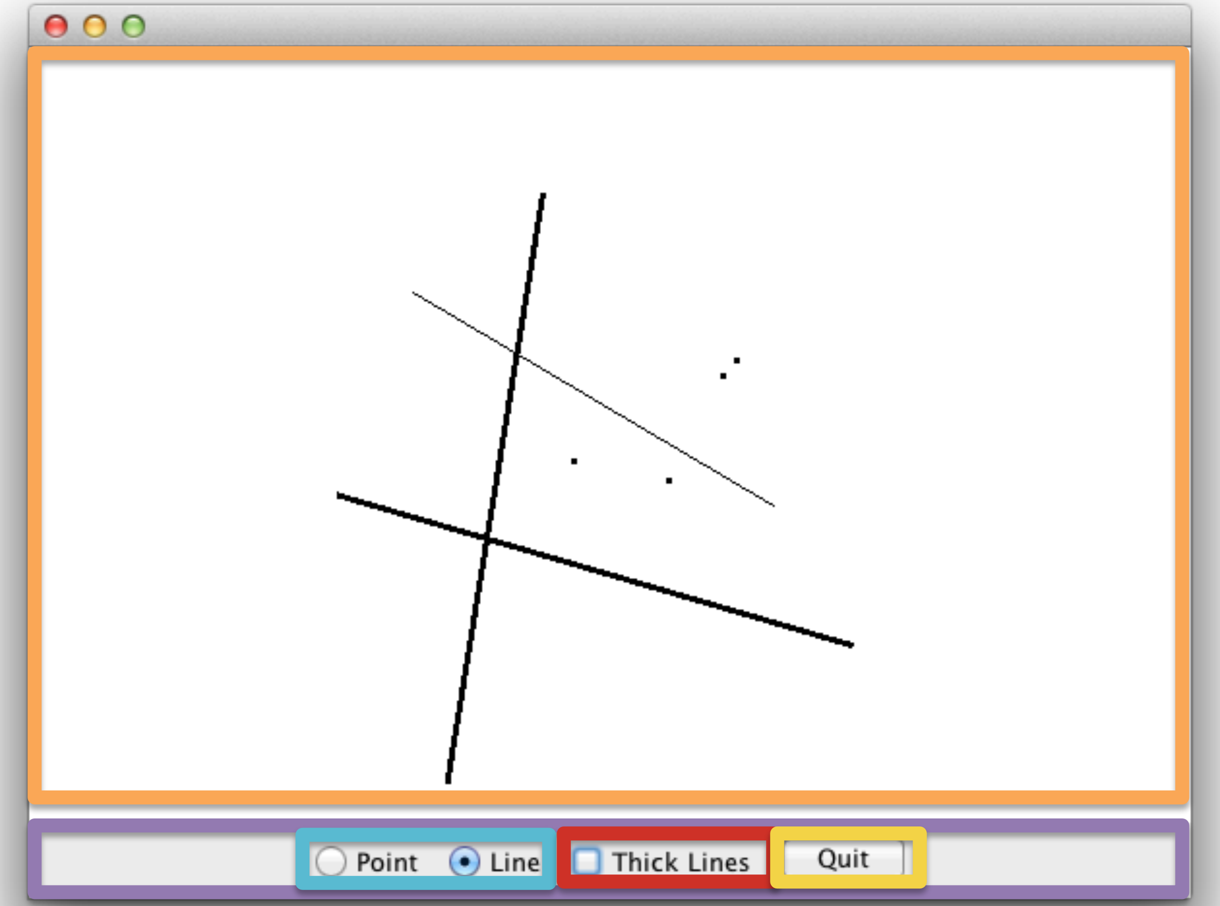
1. yes
2. yes
3. yes
4. yes

Swing Programming Demo

Layout



What layout would you use for this app? What components would you use?



Canvas subclass of JPanel (canvas)

JPanel (toolbar)

JRadioButton (point, line)

JCheckbox (thick)

JButton (quit)