Layout Managers and the EDT

Reese Hatfield

Structure

- How can we organize our components
- So far, everything kind of flows together

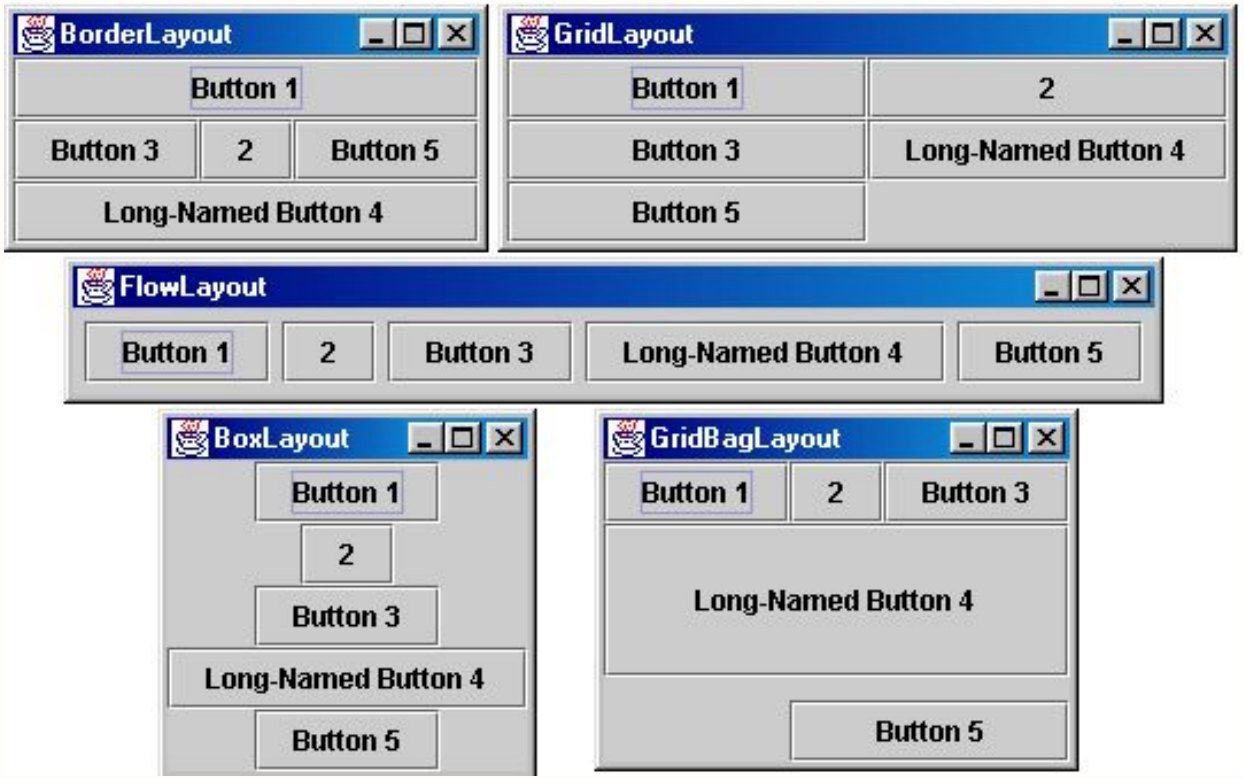
- FlowLayout = Default Layout
- There are many others

Layout Managers

- Usually placed onto JPanels
- Tell Swing how to organize components
  - As they get added

- We have already seen how things get added without changing anything

Layout Managers

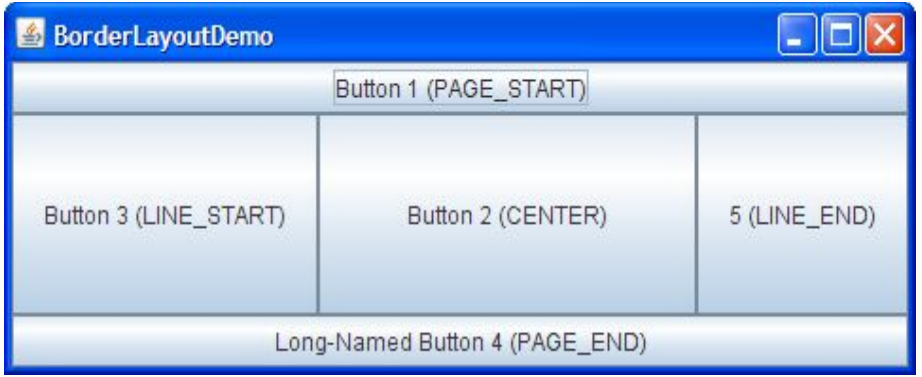


Flow Layout

- Default Layout Manager
- Positioning:
  - Horizontal Center
  - Vertical Top → Bottom

Border Layout

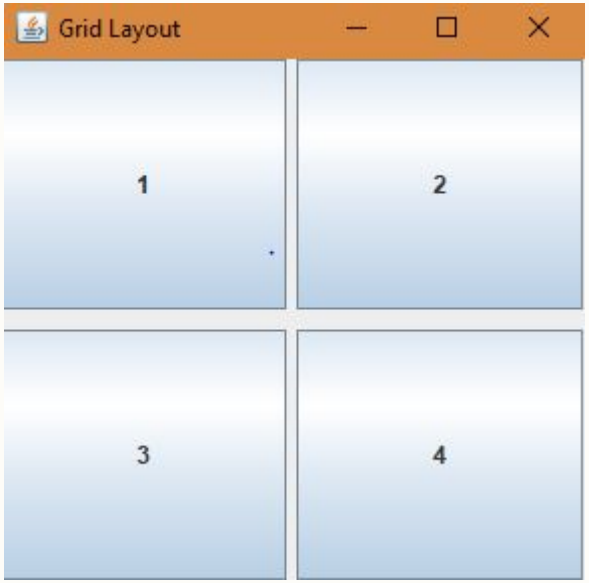
- Cardinal Organization
  - NORTH
  - SOUTH
  - EAST
  - WEST
  - CENTER



- target.add(component, position)

Grid Layout

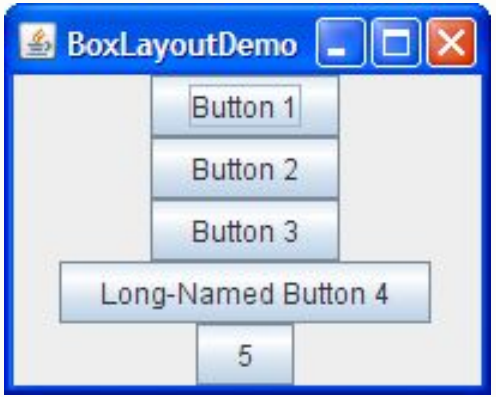
- Grid Organization
- Set rows and columns
  - Can also sets gaps
  - Left → Right
  - Top → Bottom



- New GridLayout(rows, cols)

Box Layout

- Provides different axis:
  - X_AXIS
  - Y_AXIS
  - etc.*
- Different syntax



- new BoxLayout(target, BoxLayout.Y_AXIS)

More Layouts

- There are more layout managers
- Provides additional flexibility

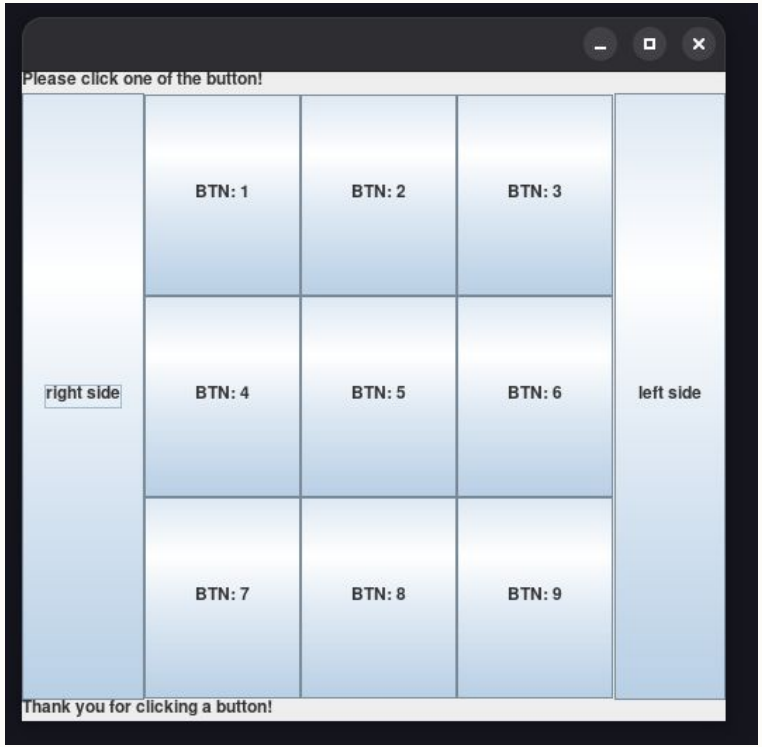
- “A Visual Guide to Layout Managers”
- Let’s take a look

Nesting Layouts

- We already saw how we can nest JPanels
- We can use this to next Layouts
- Can use this to design more complex applications

Nesting Layouts

- What if we wanted to make something like this?
- What layouts would we want?



Nesting Layouts

- Composition of layouts
- This idea transfers beyond what you'll do in swing

- Every UI framework has this same idea

More JFrames

- Right now our buttons do something simple
- Let’s do something more complicated
  - More (custom) JFrames
  - Dispose current JFrame

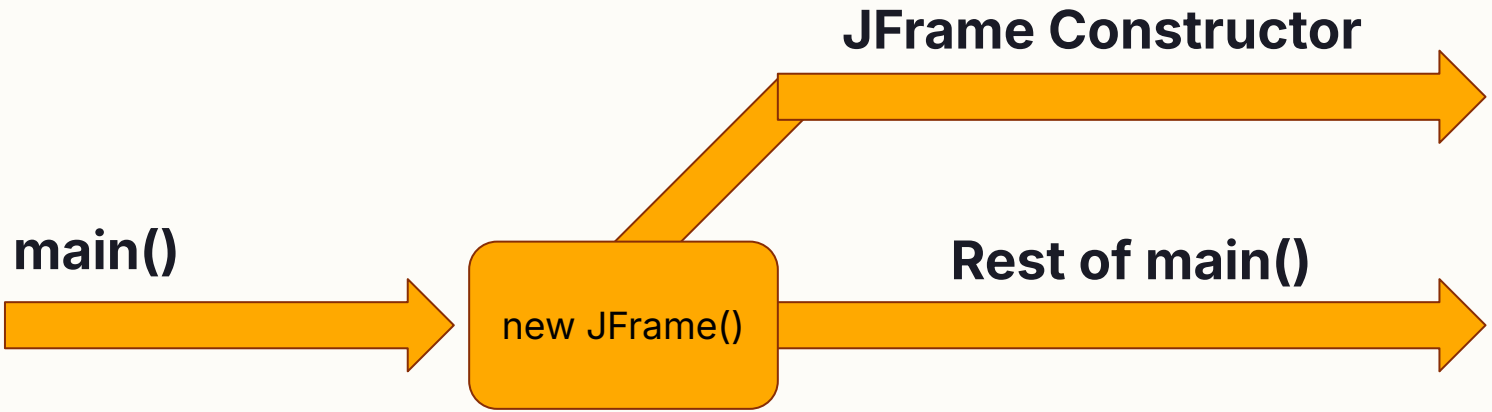
Work

- How did java make a new JFrame?
- Our code continued after dispose()

- This same thing happens in main()
- What is going on here?

Work

- These pieces of code ran simultaneously



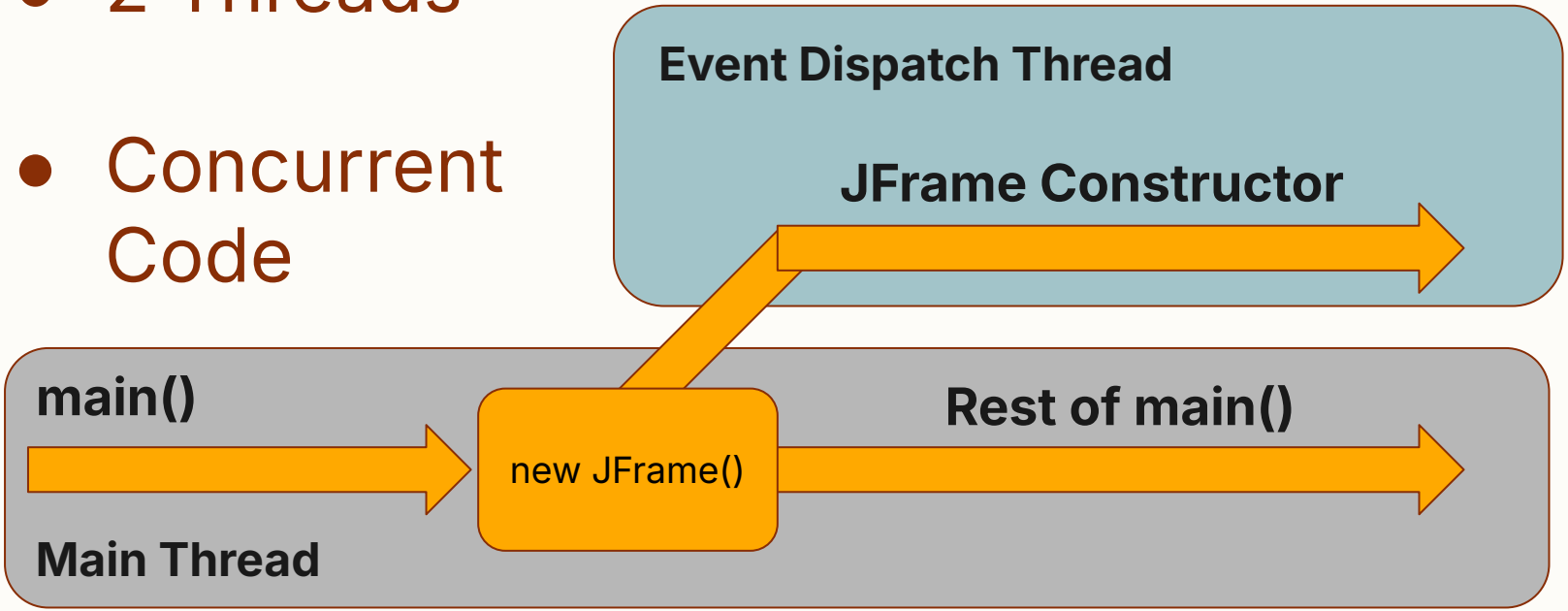
Event Dispatch Thread (EDT)

- Thread = Sequence of java code execution

- All your code so far has run on the “main” thread
- Swing code runs on the **“Event Dispatch Thread”**

Event Dispatch Thread (EDT)

- 2 Threads
- Concurrent Code



Event Dispatch Thread (EDT)

- This can get us into trouble
- By default, *all* our Swing code will run on the EDT

- EDT is responsible for all swing events (movement, graphics, etc)
- What if we did a lot of work?

Practice with Swing

- Let’s build something actually useful
- To-Do app
    - Common example
  - Design:
    - How should it look?
  - Data Model
    - How should we code it?

Design

- How do we want it to look?
  - Let’s draw it
  - Think about layouts

Data Modeling

- How do we want to code it?
  - Think about our design
  - How can we link those with our current tools?