CSE 340

Winter Quarter 2020

1/31/2020

Lecturer: Lauren Bricker

Note-taker email: jial8@uw.edu

Link to Notes on Google Docs:

https://docs.google.com/document/d/1TnWARQotjjPScjkTiQfEUooY-1PuvJlAUWgY6B2HFFQ/edit?usp=sharing

Week 4 Lecture 3
Event Records and more on Interactor Hierarchy

Interacting with the User

- In CSE 143, how did we continue to ask the user for input?
  - Scanner, System.in
  - Use a while loop until hit sentinel! Don't know when user wants to stop.
- In Layout, we were **Interface Programmers** (some people Component Developers, ex: if you made a new CaptionImage object)
- Hierarchies we have discussed:
  - Views
  - Interactors
  - Inheritance
  - Hardware → Toolkit
- Events → can be *abstract* concepts
  - Constantly running in background
  - App becomes inactive, do something!
  - Something is done being created
  - Timed events (animations, tic and toc…)

- How does Toolkit Architecture deliver events?
  - While True…
  - Deliver to correct window

Input *Dispatch Process*

- Use a **Queue** (first in first out) to handle ordering of events
- Input Thread → Dispatch Thread:
  - Front event removed from queue
  - Focus or Positional input?
    - **Focus-based = in order by interest**
      - Hovering over a button/caption
    - **Positional list (z-order under cursor from interactor hierarchy)**
      - Multiple windows, only affect the top one hovered by mouse
      - Bottom-first and Top-Down
- Event Dispatch
  - Goes down tree
  - **Find first thing that will accept event, apply event**
    - Might want to stop there or apply event to multiple things
  - *Callbacks handle application response to events*
    - Best to use custom listeners

Event Listeners

- Is an **interface** on the View class that acts as a *Callback Method*
- Must be attached to a particular View
- Standard listeners provided by Android:
  - .OnClickListener, .OnFocusListener
  - See more:
    https://developer.android.com/guide/topics/ui/ui-events

How to implement Listener?

1. Create Anonymous Class (private….)
2. Some Variable a = new View.onClickListener() { ….. }
3. **Abstract void, so need to use brackets**, has not been declared, so we need to do it
   a. Implement onClick public void method inside
   b. … etc
4. Put semicolon to indicate end of anonymous class

How to set button to a listener?

- If some button != null, mButton.setOnClickListener(mClickListener)
- Can register >= 1 listener to one widget