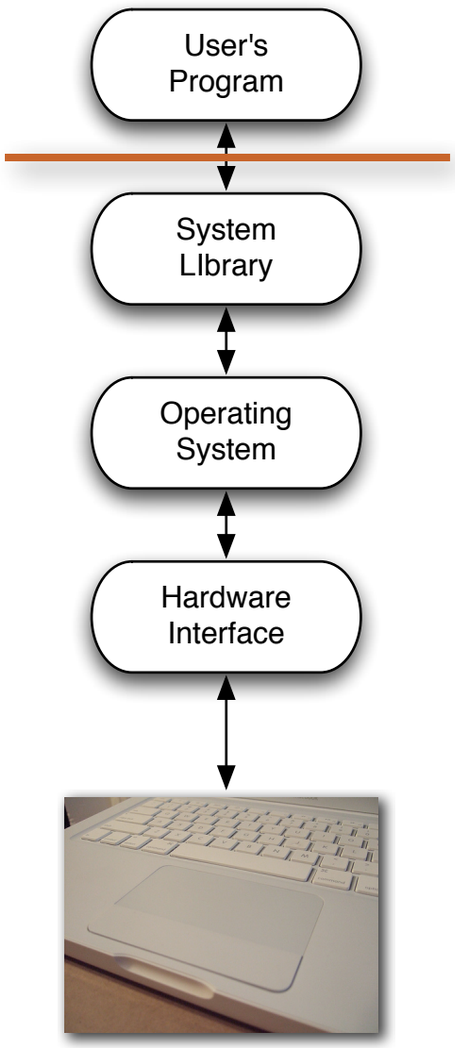
User Interaction: Intro to Multi-Touch

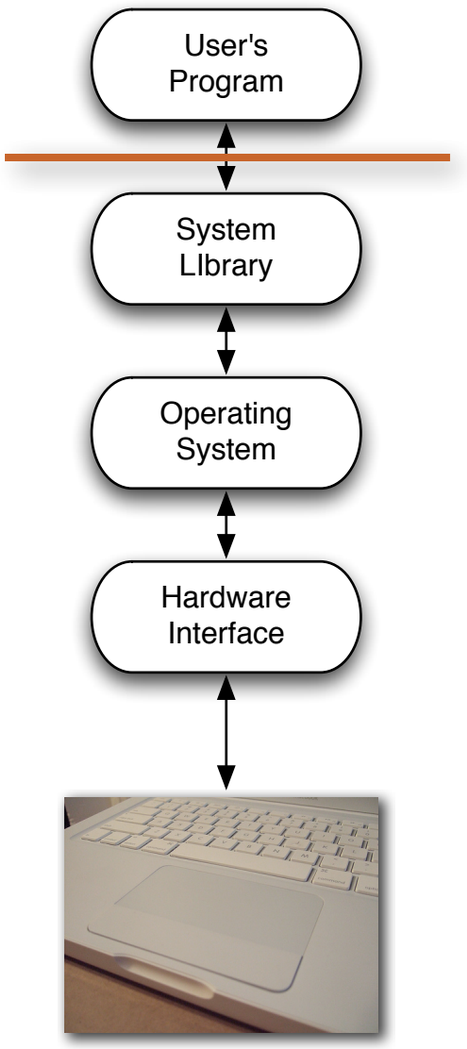
Associate Professor Donald J. Patterson
INF 133 Fall 2013

Multi-Touch Approach #1 - rebuild the Observer Pattern



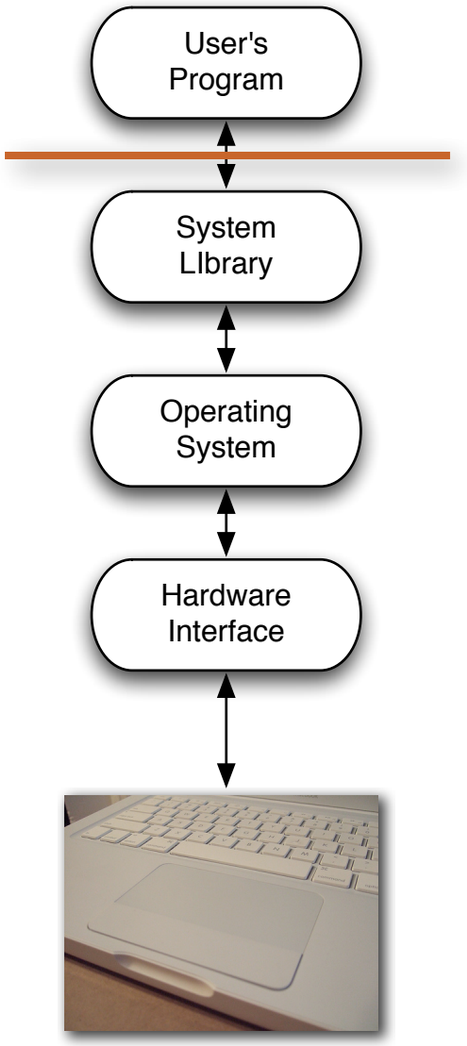
- Design specific multi-touch/gesture events that you can register for:
  - Pinching movements (in or out)
    - meaning zoom out or zoom in
  - Rotate: Two fingers moving in opposite semicircles is a gesture meaning rotate.
  - Swipe: Three fingers brushing across the trackpad surface in a common direction.
  - Scroll: Two fingers moving vertically or horizontally is a scroll gesture.

Multi-Touch Approach #1 - rebuild the Observer Pattern

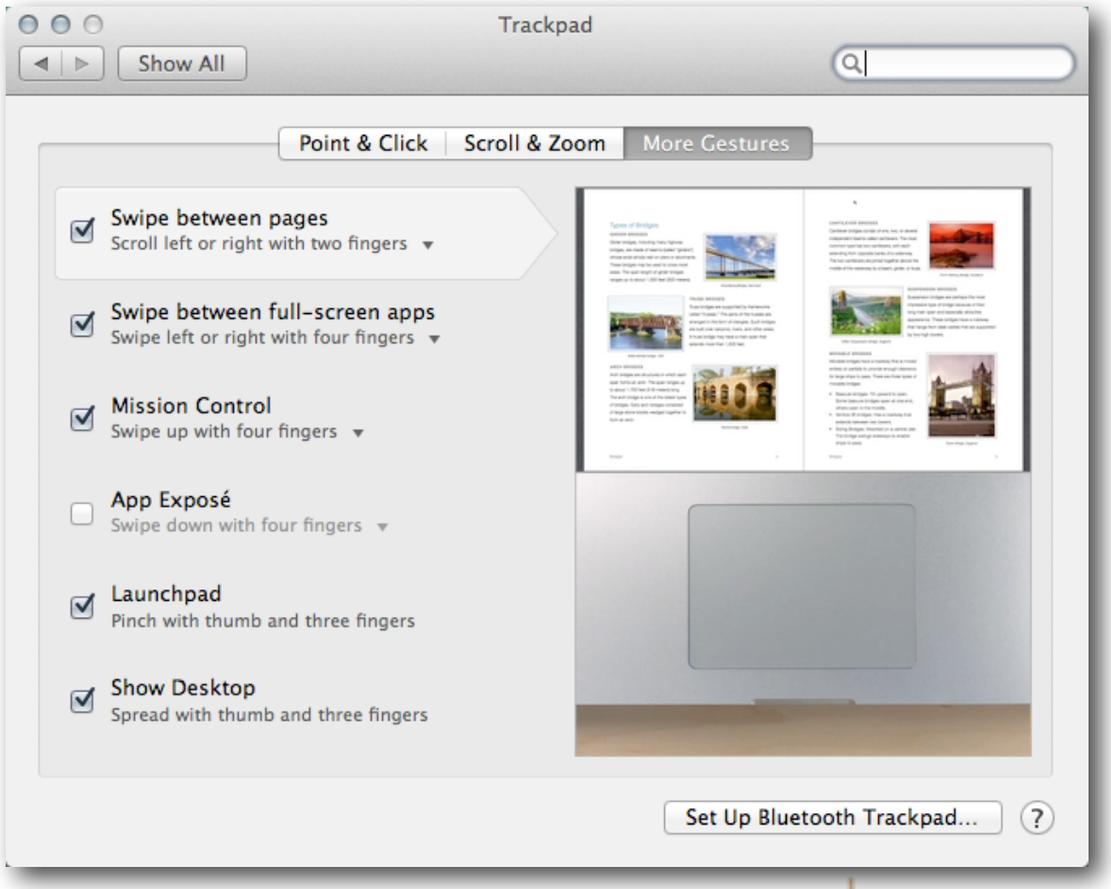


- Advantages:
  - Simple to code
  - Library/OS does all the work
- Disadvantages
  - No flexibility
  - Limited to supported events

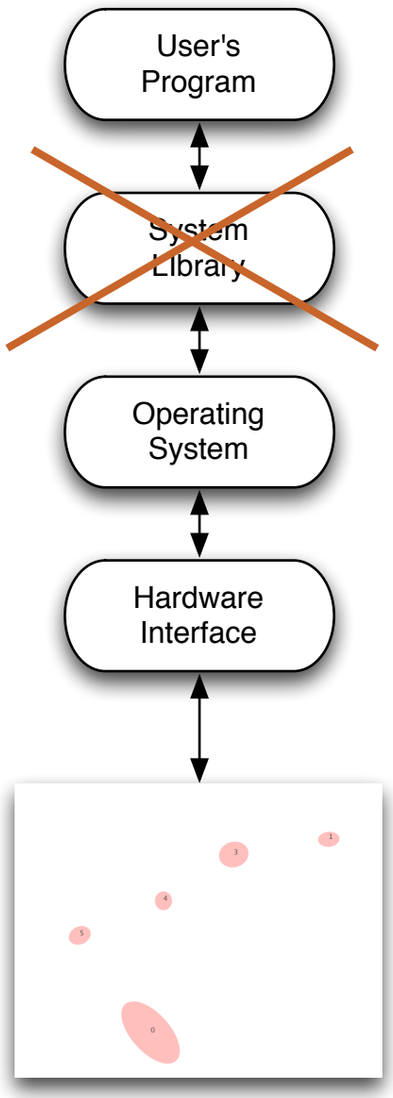
Multi-Touch Approach #1 - rebuild the Observer Pattern



- Examples (demo):
  - Document browsing in Preview
    - Zoom
    - Scale
    - Swipe

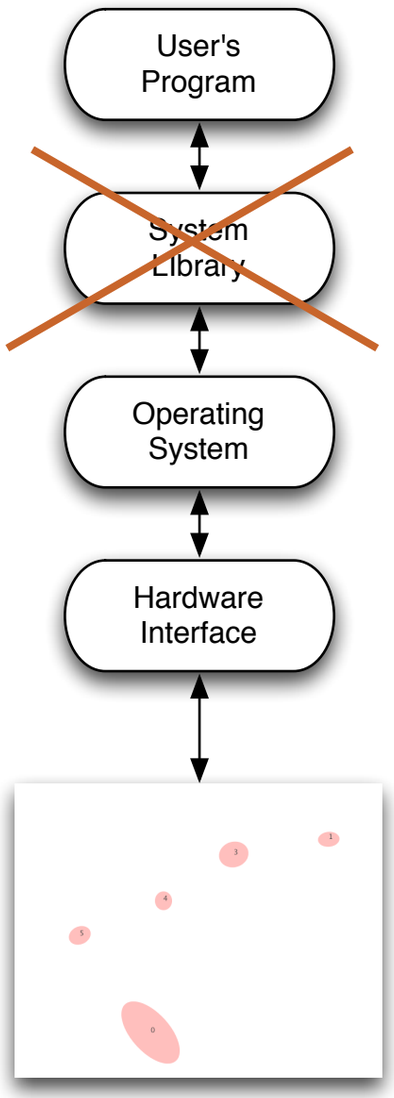


Multi-Touch Approach #2 - Blob interpreter



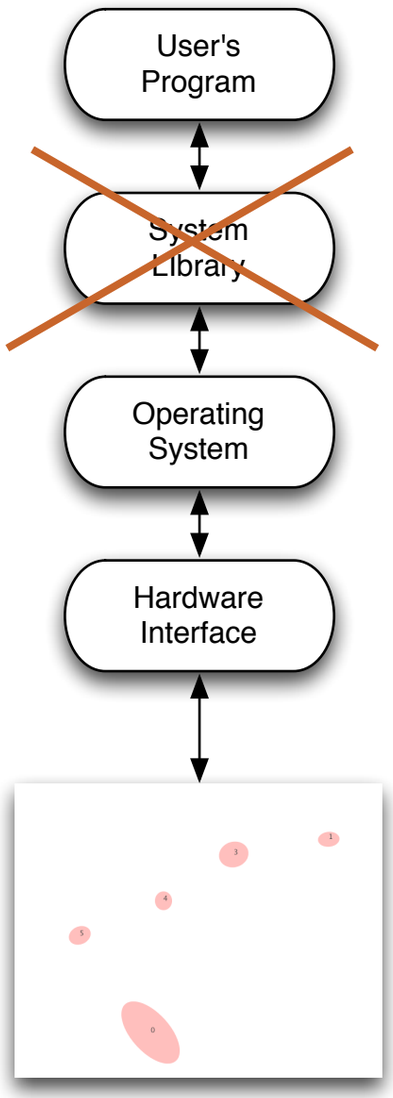
- Blob interpretation by program
  - A program receives information about the location/"pressure"/orientation of multiple touches
  - Each touch gets an id to uniquely identify it
  - This is a stream of data
    - continuously updating locations and ids

Multi-Touch Approach #2 - Blob interpreter



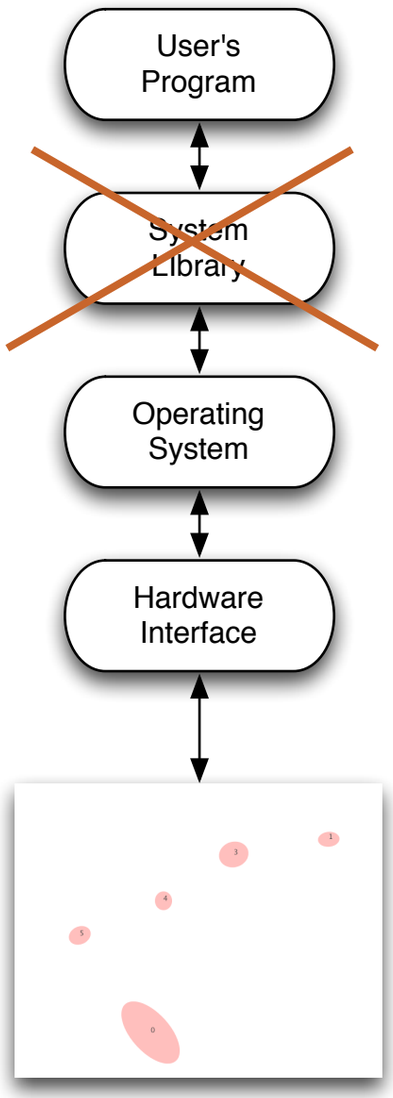
- Advantages
  - Supports unlimited numbers of touches
    - two hands / multiple people
  - Programs can have gestures that make unique sense for them
  - OS does a lot of work to find and report blobs

Multi-Touch Approach #2 - Blob interpreter



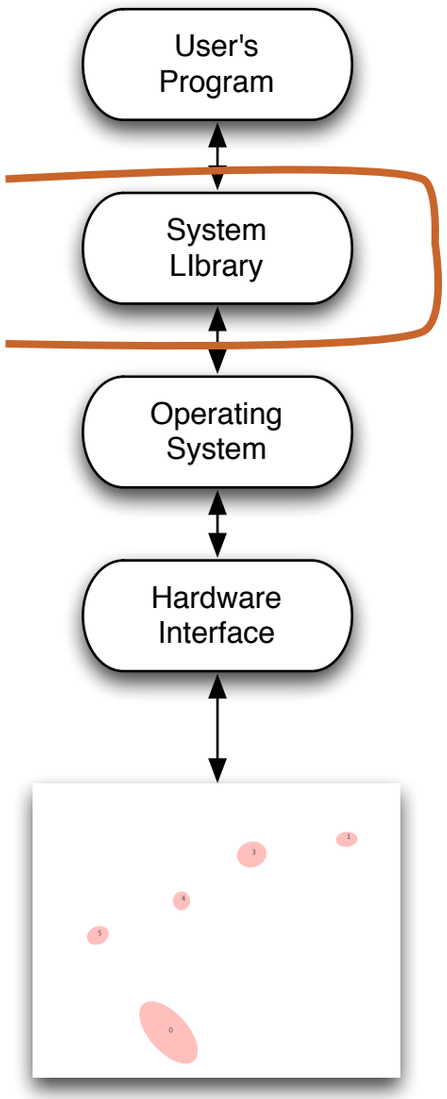
- Disadvantages
  - Each program has to interpret events itself
    - Was that a pinch?
    - Was that a rotate?
    - Where is the thumb?

Multi-Touch Approach #2 - Blob interpreter



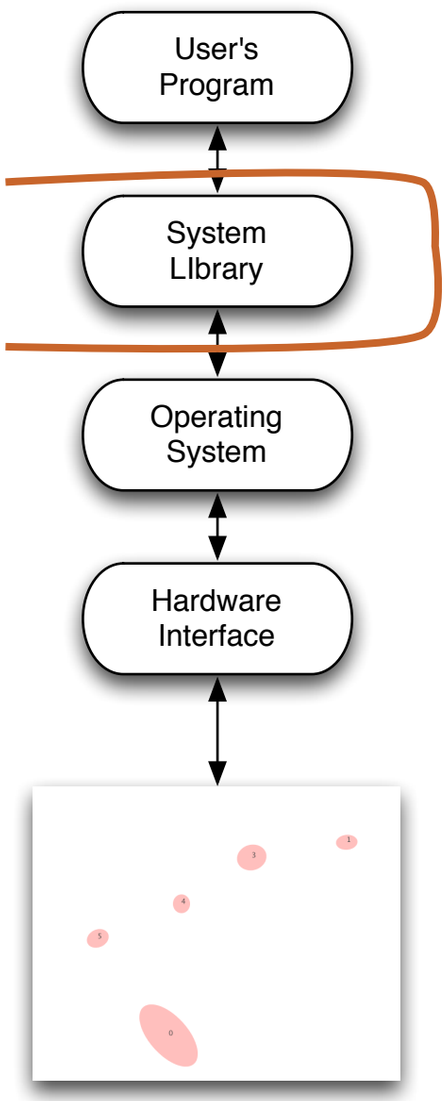
- Examples
  - MacMultitouch Demo
    - FingerMgmt
    - In this example the program plots the blobs that would be interpreted by a full-program

Multi-Touch Approach #3 - DIY Observer Pattern



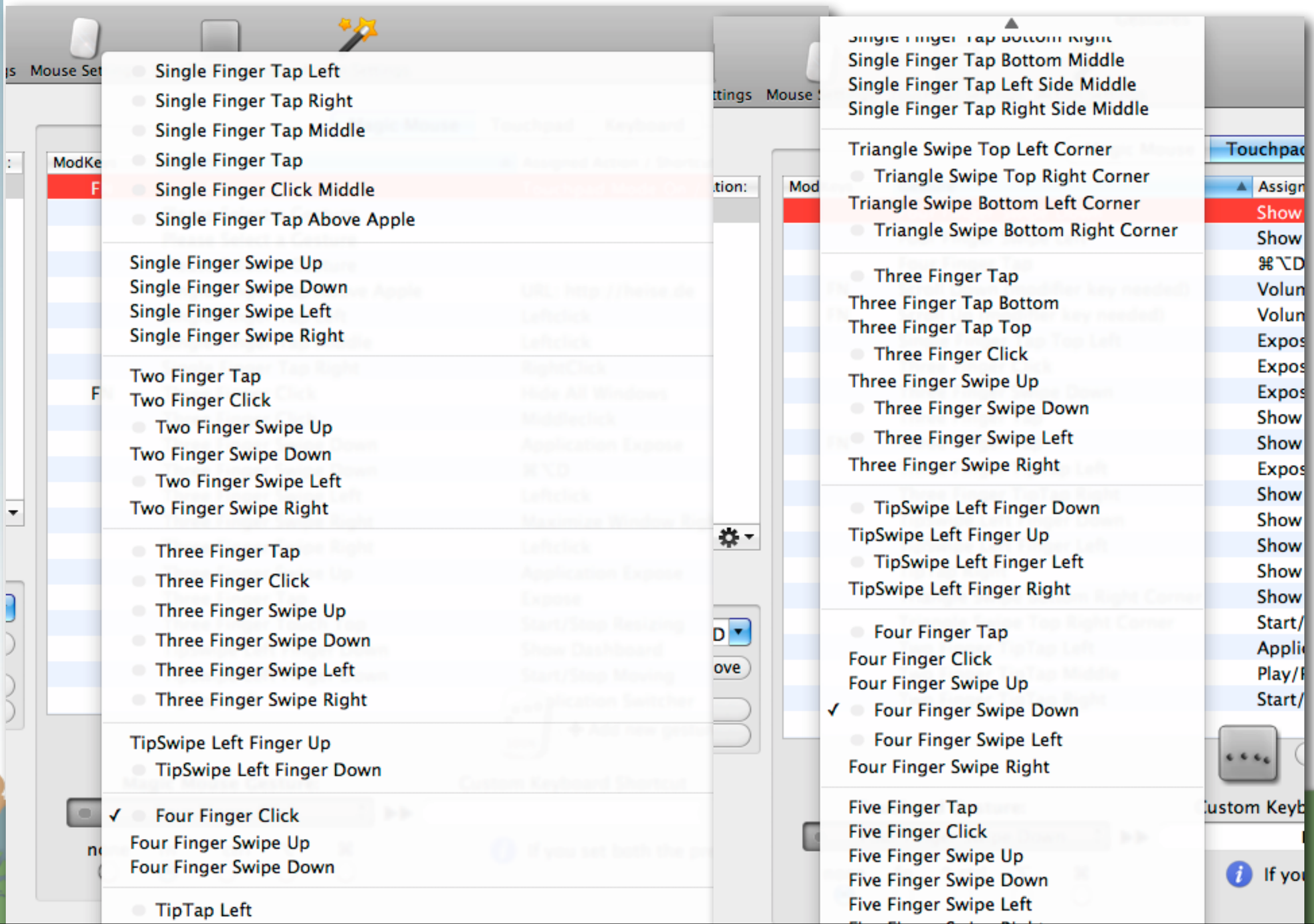
- Create your own event layer for everyone b/c
  - Everyone wants to … detect triangle touches
  - Everyone wants to … interpret for multiple people
  - Everyone needs a … “tiptap” interaction

Multi-Touch Approach #3 - DIY Observer Pattern

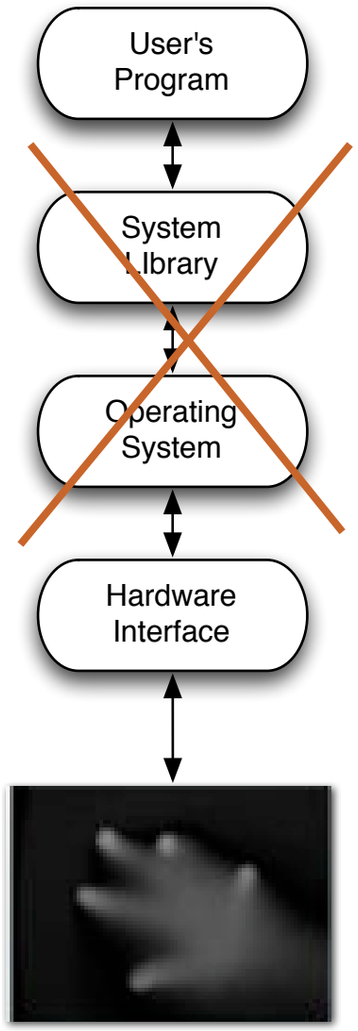


- Advantages:
  - Scalable (Other people can use it)
  - Allows completely new interface design
    - “3-finger pinch”
  - Lots of potential for innovation
- Disadvantages
  - Lots to code
  - Limited application support

Multi-Touch Approach #3: Better Touch Tool (http://boastr.net/)

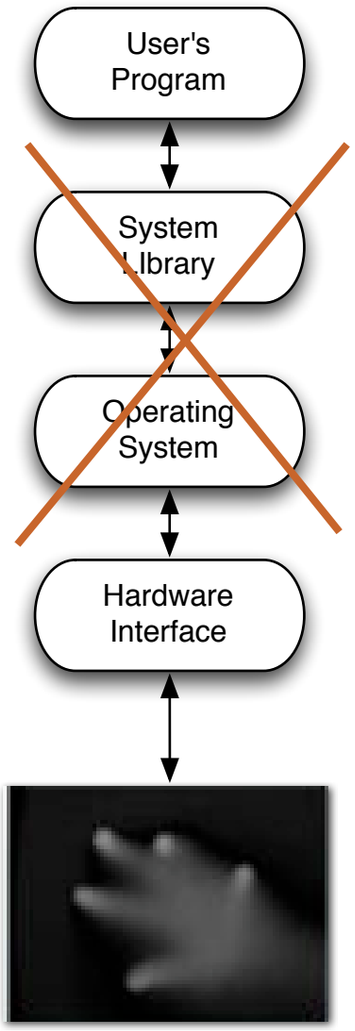


Multi-Touch Approach #4 - Do Everything Yourself



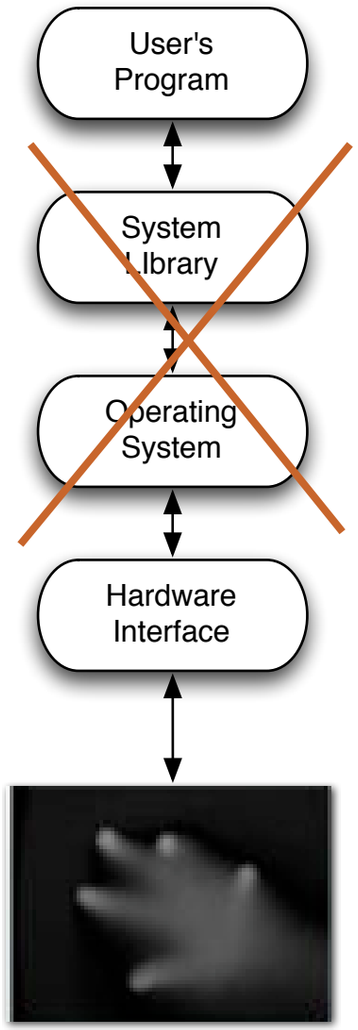
- Grayscale input
  - A program receives a stream of images
  - Darker (or lighter) colors indicates pressure or proximity

Multi-Touch Approach #4 - Do Everything Yourself



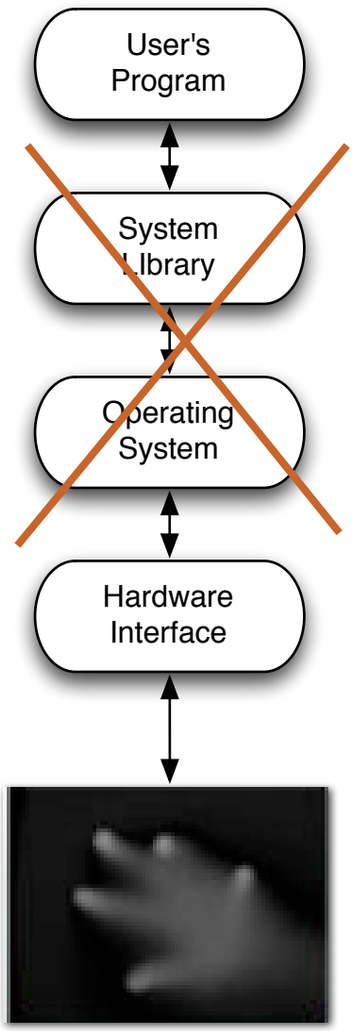
- Advantages
  - Maximum flexibility
  - Not restricted to “finger touch” paradigm
    - Can recognize a “cup down” event for example

Multi-Touch Approach #4 - Do Everything Yourself



- Disadvantages
  - This is full-fledged computer vision
  - Different technologies generate different quality images
  - Robustly and consistently recognizing events is hard.

Multi-Touch Approach #4 - Do Everything You



- Examples
  - iShred
  - http://www.youtube.com/watch?v=eZpnzzKbY2I&feature=player_embedded

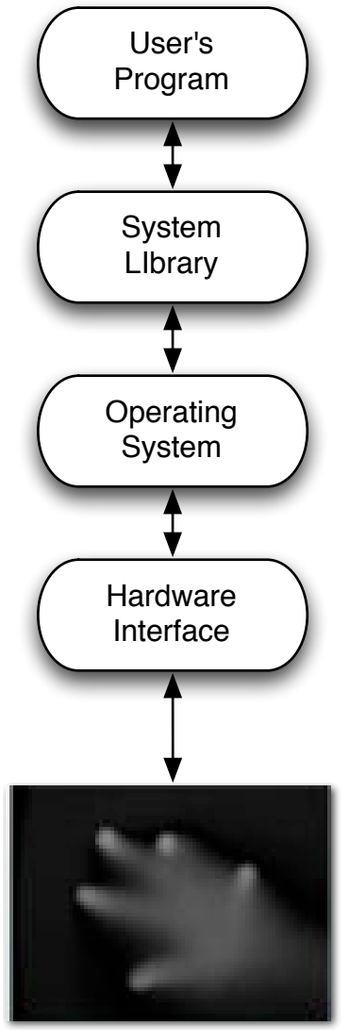
Microsoft Surface (table version)

- http://www.youtube.com/embed/C36rm5yS4c4?rel=0

How do you choose?

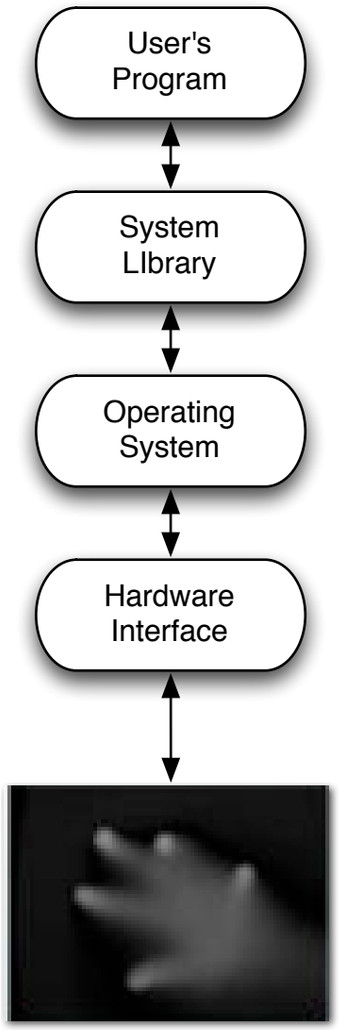
- How fast do you need to get your application done?
  - #1 is fastest, #4 is slowest
- Who are your users?
  - #1 is the most familiar to users, #4 requires users to adapt
- What is your application?
  - #1 is basically point and click extensions
  - #4 supports crazy gaming/applications
- Are you showcasing multi-touch? or supporting a task?

Our assignment



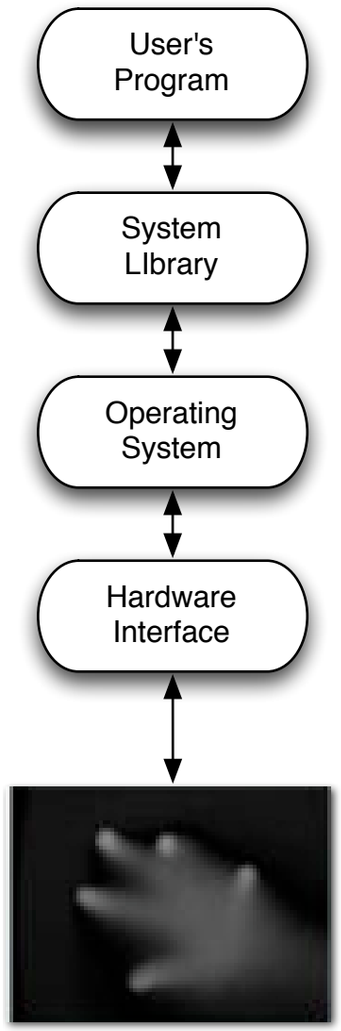
- Build a multi-touch Java paint application
  - No OS support

Our assignment



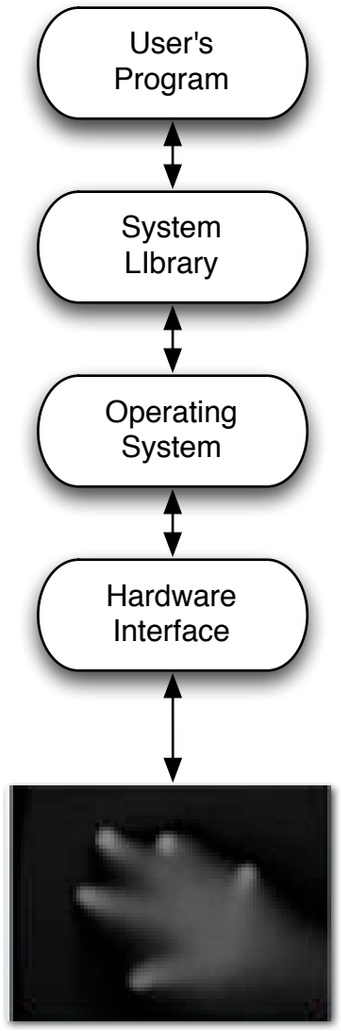
- Where are we going to get a grayscale input?
  - You can build your own
  - You can use prerecorded video

Our assignment



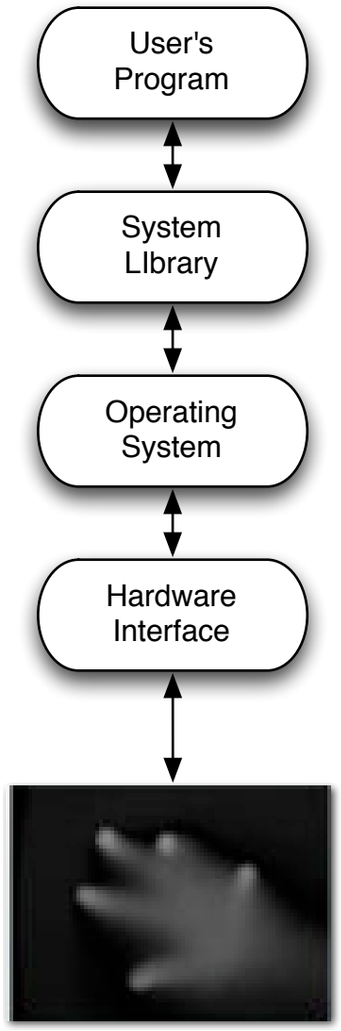
- How will we interface to the computer?
  - Use standard camera inputs

Our assignment



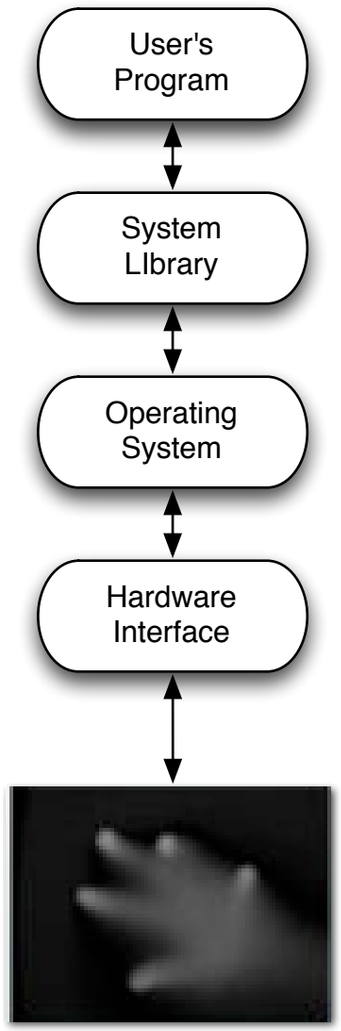
- How will we process it without OS support?
  - We will use Community Core Vision to process the grayscale images

Our assignment



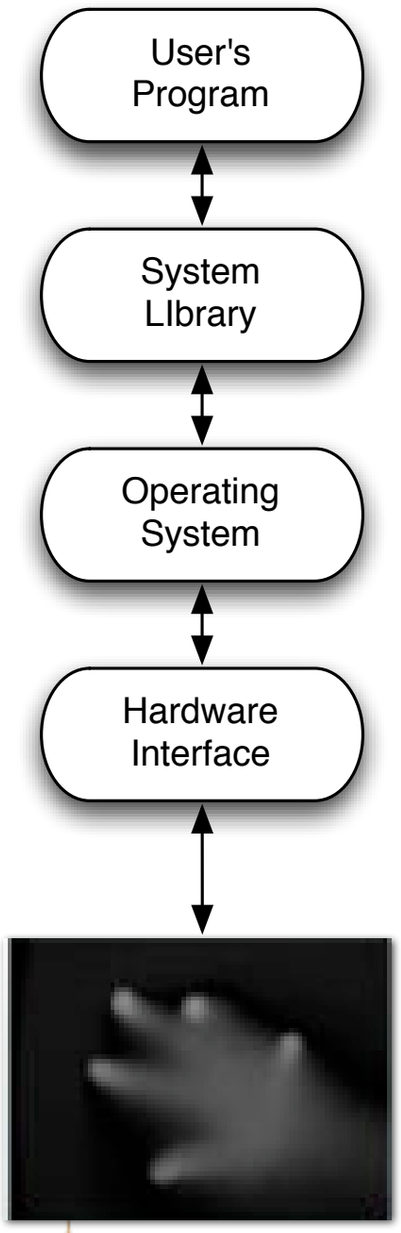
- How will will our application get information about multi-touch events?
  - Using the TUIO standard and a TUIO library for java

Our assignment



- How will I write a multi-touch application?
  - Register for multi-touch events and then respond when you receive them.

Getting Multi-Touch up and Running



- Your program
- MT4J

  TUIO
- Community Core Vision
- web cam

Getting Multi-Touch up and Running

- Community Core Vision (http://ccv.nuigroup.com/)
  - is a open source/cross-platform solution for computer vision and machine sensing. It takes an video input stream and outputs tracking data (e.g. coordinates and blob size) and events (e.g. finger down, moved and released) that are used in building multi-touch applications.

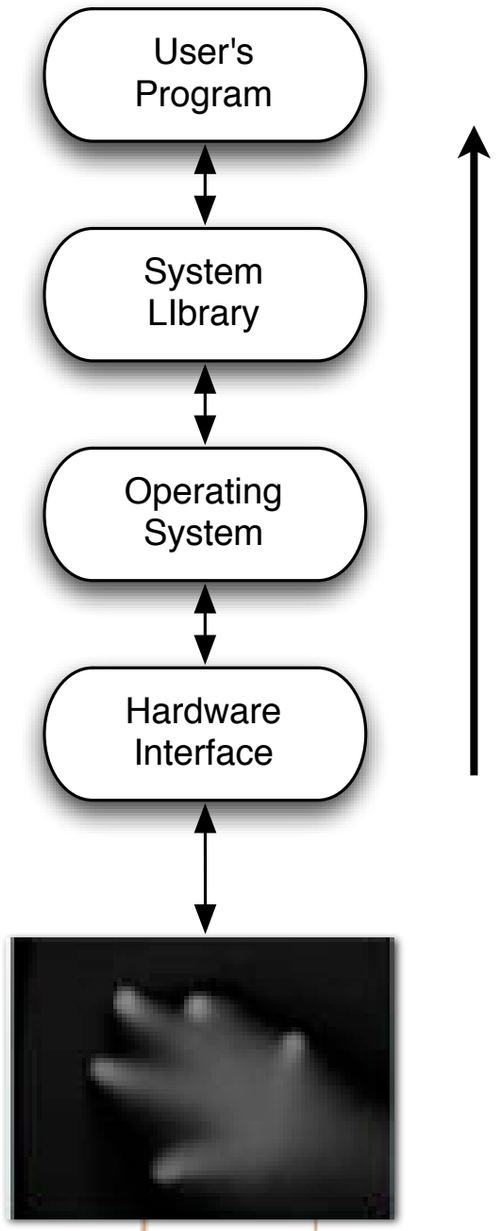
Getting Multi-Touch up and Running

- MT4J
  - MT4j - Multitouch for Java™ - is an open source Java™ development platform, created for rapid development of graphically rich applications. MT4j is designed to support different kinds of input devices with a special focus on multitouch support.
  - http://www.mt4j.org

Getting Multi-Touch up and Running

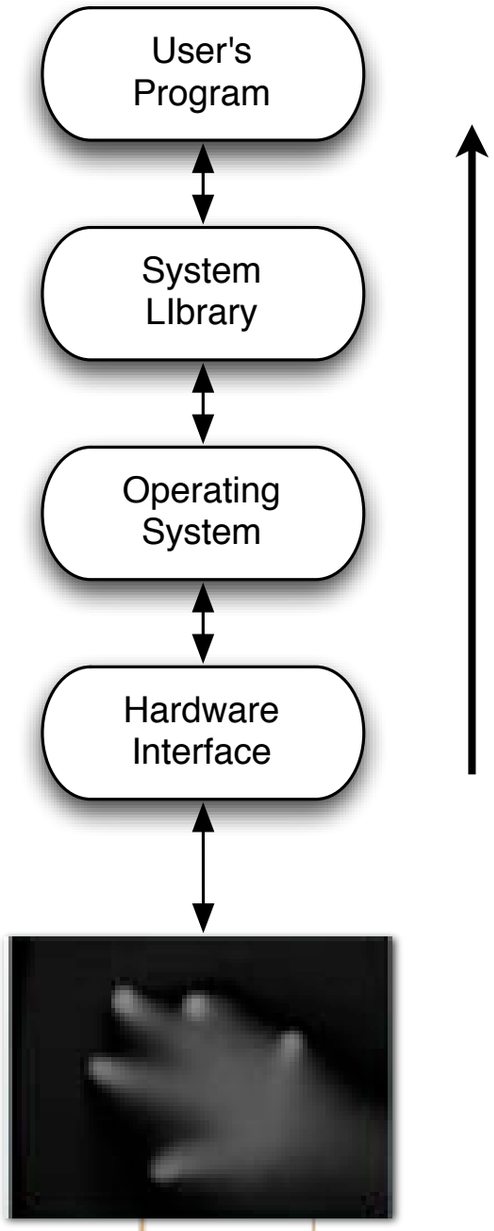
- TUIO
  - TUIO is an open framework that defines a common protocol and API for tangible multitouch surfaces. The TUIO protocol allows the transmission of an abstract description of interactive surfaces, including touch events and tangible object states.
  - http://www.tuio.org/

Getting Multi-Touch up and Running: Demo #1



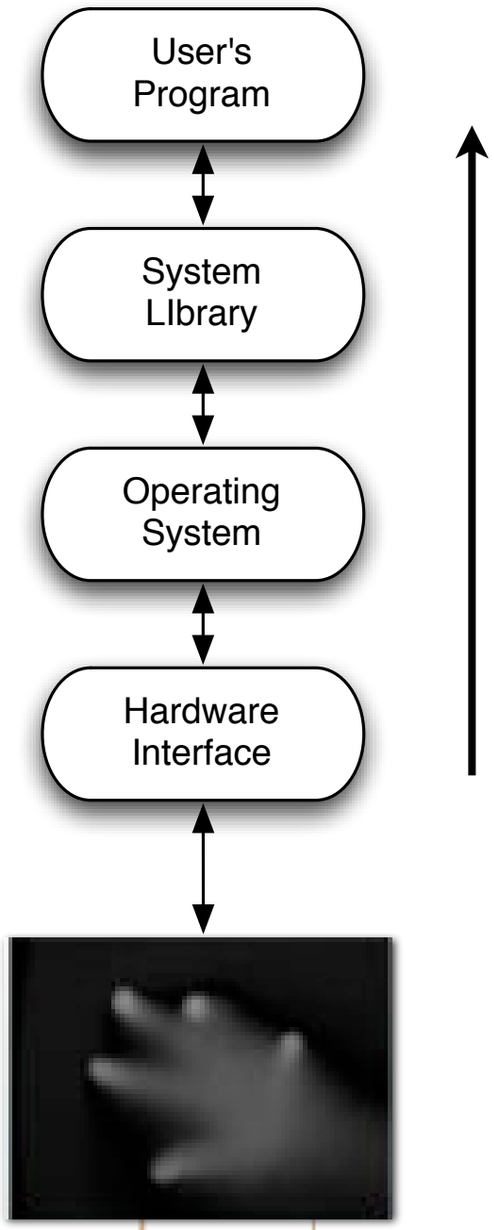
- To a flash application that is multi-touch aware
- Through a TUIO server
- Going through CCV
- Video from recorded gray scale

Getting Multi-Touch up and Running: Demo #2



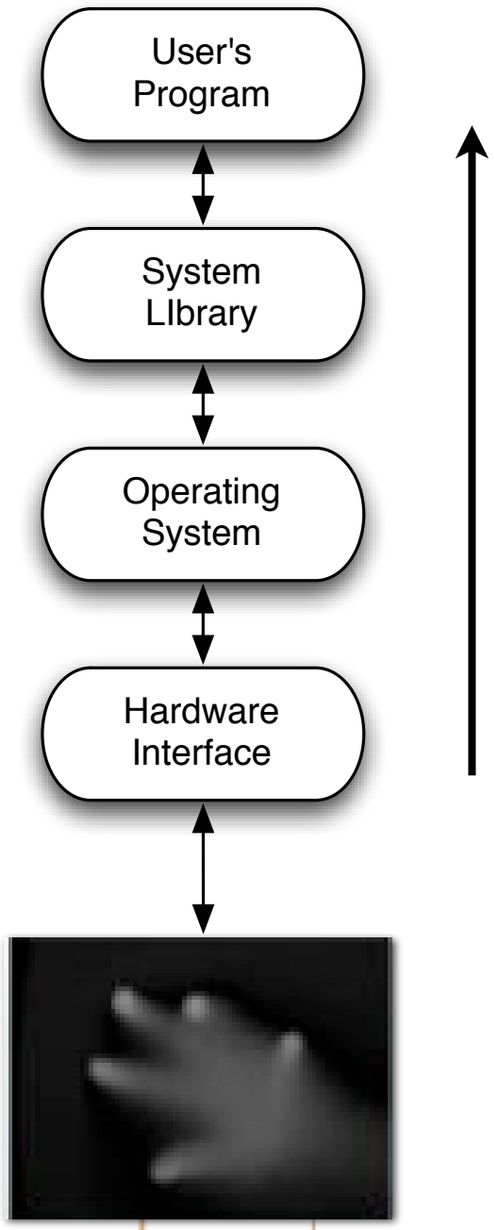
- To a flash application that is multi-touch aware
- Through a TUIO server
- Going through CCV
- Live video

Getting Multi-Touch up and Running: Demo #3



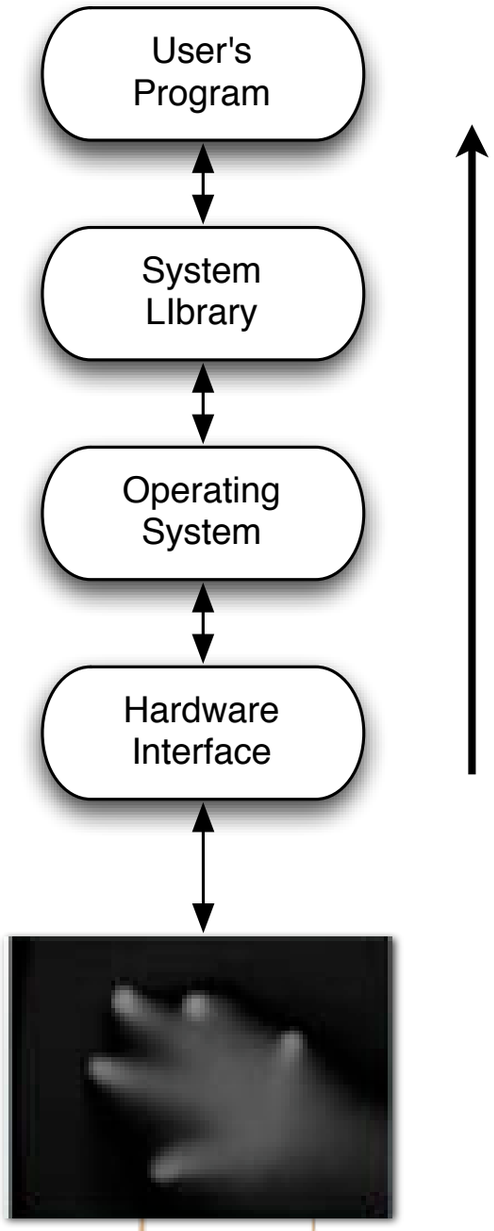
- To a MT4J application that is multi-touch aware
- Going through a MT4J TUIO server
- Multitouch from a TUIO simulator

Getting Multi-Touch up and Running: Demo #4



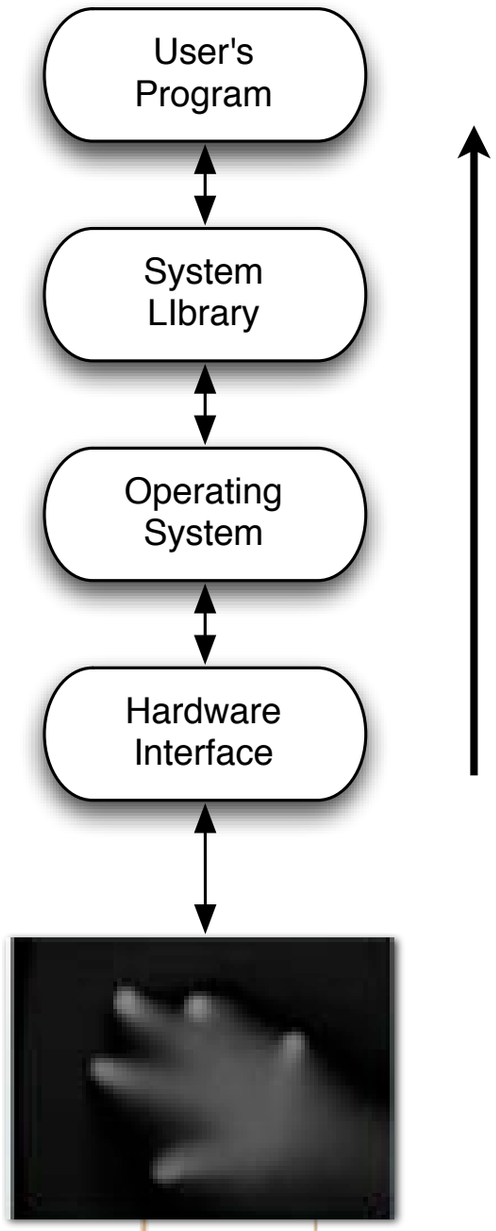
- To a MT4J application that is multi-touch aware
- Going through a MT4J TUIO server
- Live multitouch from iPhone

Getting Multi-Touch up and Running: Demo #4

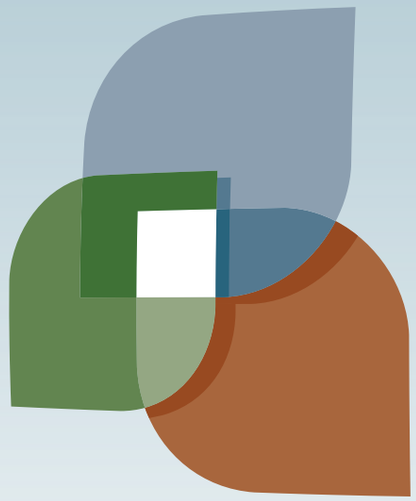


- To a MT4J application that is multi-touch aware
- Going through a MT4J TUIO server
- Going through CCV
- Live video

Getting Multi-Touch up and Running: Demo #4



- To a MT4J application that is multi-touch aware
- Going through a MT4J TUIO server
- Going through CCV
- Live video from lights



LUCI