User Interaction: Intro to Multi-Touch

Associate Professor Donald J. Patterson
INF 133 Fall 2014
Traditional Mouse Input

User's Program
→ Java Virtual Machine
→ Operating System
→ Hardware Interface

User's Program
→ System Library
→ Operating System
→ Hardware Interface
Java uses a “MouseListener” model

- The developer asks the virtual machine to tell it when mouse events occur
- Mouse movements
- Mouse button press, release, click
  - button 1, 2, 3
- Mouse wheel movements
Java uses a “MouseListener” model

- The developer asks the virtual machine to tell it when mouse events occur
  - Mouse movements
  - Mouse button press, release, click
    - button 1,2,3
  - Mouse wheel movements
Java uses a “MouseListener” model

- The developer asks the virtual machine to tell it when mouse events occur
  - Mouse movements
  - Mouse button press, release, click
    - button 1, 2, 3
  - Mouse wheel movements
Java uses a “MouseListener” model

- The developer asks the virtual machine to tell it when mouse events occur
  - Mouse movements
  - Mouse button press, release, click
    - button 1, 2, 3
  - Mouse wheel movements
Java uses a “MouseListener”

- “Observer” design pattern
- Example:
  - [http://java.sun.com/docs/books/tutorialJWS/uiswing/events/ex6/MouseEventDemo.jnlp](http://java.sun.com/docs/books/tutorialJWS/uiswing/events/ex6/MouseEventDemo.jnlp)
The “Observer” gets callbacks from the “Subject”

Same pattern we saw in AJAX

Common pattern in event-driven User Interface software

“MouseListener” maps to the “Observer”

Traditional Mouse Input

```java
public class MouseEventArgsExample implements MouseListener {
    // where initialization occurs:
    // Register for mouse events on blankArea and the panel.
    blankArea.addMouseListener(this);
    addMouseListener(this);
    ...

    public void mousePressed(MouseEvent e) {
        saySomething("Mouse pressed; # of clicks: ",
            e.getClickCount(), e);
    }

    public void mouseReleased(MouseEvent e) {
        saySomething("Mouse released; # of clicks: ",
            e.getClickCount(), e);
    }

    public void mouseEntered(MouseEvent e) {
        saySomething("Mouse entered", e);
    }

    public void mouseExited(MouseEvent e) {
        saySomething("Mouse exited", e);
    }

    public void mouseClicked(MouseEvent e) {
        saySomething("Mouse clicked (# of clicks: ",
            e.getClickCount() + ")", e);
    }

    void saySomething(String eventDescription, MouseEvent e) {
        textArea.append(eventDescription + " detected on "
            + e.getComponent().getName() + "." + newline);
    }
}
```

http://java.sun.com/docs/books/tutorial/uiswing/events/mouselistener.html
Traditional Mouse Input

Tell me when the mouse is pressed, released, enters the boundaries, exits the boundaries, or is clicked.

public class MouseEventDemo ... implements MouseListener {
    // where initialization occurs:
    // Register for mouse events on blankArea and the panel.
    blankArea.addMouseListener(this);
    addMouseListener(this);
    ...

    public void mousePressed(MouseEvent e) {
        saySomething("Mouse pressed; # of clicks: "+e.getClickCount(), e);
    }

    public void mouseReleased(MouseEvent e) {
        saySomething("Mouse released; # of clicks: "+e.getClickCount(), e);
    }

    public void mouseEntered(MouseEvent e) {
        saySomething("Mouse entered", e);
    }

    public void mouseExited(MouseEvent e) {
        saySomething("Mouse exited", e);
    }

    public void mouseClicked(MouseEvent e) {
        saySomething("Mouse clicked (# of clicks: "+e.getClickCount()+")", e);
    }

    void saySomething(String eventDescription, MouseEvent e) {
        textarea.append(eventDescription + " detected on "
                        + e.getComponent().getClass().getName() + "." + newline);
    }
}
Mouse Event

- When your program is told that something happened, you get extra info with the event (as a parameter)
- Single or double click?
- \((X, Y)\) of event
  - global and local coordinates
- which button was pushed \((1,2,3)\)
- Modifier keys
  - “Shift” click
Mouse Event (cont)

- When your program is told that something happened, you get extra info with the event
- Which UI component is reporting
  - “blankArea”
- timestamp
- and a few more things
## Method Summary

<table>
<thead>
<tr>
<th>Modifier and Type</th>
<th>Method and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td><code>getButton()</code></td>
</tr>
<tr>
<td></td>
<td>Returns which, if any, of the mouse buttons has changed state.</td>
</tr>
<tr>
<td>int</td>
<td><code>getClickCount()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the number of mouse clicks associated with this event.</td>
</tr>
<tr>
<td>Point</td>
<td><code>getLocationOnScreen()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the absolute x, y position of the event.</td>
</tr>
<tr>
<td>int</td>
<td><code>getModifiersEx()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the extended modifier mask for this event.</td>
</tr>
<tr>
<td>static String</td>
<td><code>getMouseModifiersText(int modifiers)</code></td>
</tr>
<tr>
<td></td>
<td>Returns a String instance describing the modifier keys and mouse buttons that were down during the event, such as &quot;Shift&quot;, or &quot;Ctrl+Shift&quot;.</td>
</tr>
<tr>
<td>Point</td>
<td><code>getPosition()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the x,y position of the event relative to the source component.</td>
</tr>
<tr>
<td>int</td>
<td><code>getX()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the horizontal x position of the event relative to the source component.</td>
</tr>
<tr>
<td>int</td>
<td><code>getXOnScreen()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the absolute horizontal x position of the event.</td>
</tr>
<tr>
<td>int</td>
<td><code>getY()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the vertical y position of the event relative to the source component.</td>
</tr>
<tr>
<td>int</td>
<td><code>getYOnScreen()</code></td>
</tr>
<tr>
<td></td>
<td>Returns the absolute vertical y position of the event.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>isPopupTrigger()</code></td>
</tr>
<tr>
<td></td>
<td>Returns whether or not this mouse event is the popup menu trigger event for the platform.</td>
</tr>
<tr>
<td>String</td>
<td><code>paramString()</code></td>
</tr>
<tr>
<td></td>
<td>Returns a parameter string identifying this event.</td>
</tr>
<tr>
<td>void</td>
<td><code>translatePoint(int x, int y)</code></td>
</tr>
<tr>
<td></td>
<td>Translates the event's coordinates to a new position by adding specified x (horizontal) and y (vertical) offsets.</td>
</tr>
</tbody>
</table>

http://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseEvent.html
Different types of input devices

- What about trackpads?
- What about tablets?
- What about rollerballs?
Different types of input devices

- As long as the OS can translate the hardware interaction into the same events then programs are compatible.
- A stylus on a tablet can “click”
- A rollerball “enters” and “exits”
- A finger on a trackpad has an (X,Y)
Poll

- Who should we make fun of?
  - Microsoft
  - Apple
Multi-touch is different
Multi-touch is different

http://www.youtube.com/embed/Cog8b8oiji0?rel=0
Multi-touch creates new interactions

- This breaks old programs
- unless the OS makes the multi-touch look like a mouse to the program
Multi-touch creates new interactions

- Watch Jeffrey Han Demo Reel
- What is different from working with a mouse?
Multi-touch is different

http://www.youtube.com/watch?v=EiS-W9aeG0s
Multi-touch creates new interactions

- pointer is gone
  - all interaction is active
- hover is gone
- you can’t see what you are clicking
- “clicking” isn’t [as] natural
- “swiping” is natural
Multi-touch creates new interactions

- Software has to be (re)written to be
  - “multi-touch” aware
- The OS can give some support
  - exposing new events like
    - “pinch” (tell me when a pinch occurs)
    - “rotate” (tell me when a rotate occurs)
    - “two finger swipe”
    - “three finger swipe”
Multi-touch creates new interactions

- But multi-touch is really computer vision

Where is the mouse clicking?

What abstractions will the OS expose?
Multi-touch creates new interactions

- Watch 10/GUI video
- http://10gui.com/video/
Multi-touch terminology

- **Multi-touch** – An interactive technique that allows single or multiple users to control graphical displays with more than one simultaneous finger.
- **Multi-point** – An interactive technique that makes use of points of contact rather than movement. A multi-point kiosk with buttons would be an example.
- **Multi-user** – A multi-touch device that accepts more than one user. Larger multi-touch devices are said to be inherently multi-user.
- **Multi-modal** – A form of interaction using multiple modes of interfacing with a system.
Multi-touch terminology

- **Tabletop Computing** – Interactive computer displays that take place in the form of tabletops.

- **Direct Manipulation** – The ability to use the body itself (hands, fingers, etc) to directly manage digital workspaces.

- **Blob tracking** - Assigning each blob an ID (identifier). Each frame we try to determine which blob is which by comparing each with the previous frame.

- **Blob detection** - Process of picking out bright areas of a camera image and somehow relaying them to a computer as a touch.
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/)

Hardware

Interface

Operating System

System

Library

User’s Program
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 Yourself

• 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
• How will we process it without OS support?
• We will use Community Core Vision to process the grayscale images
• How 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
• 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
• 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 TUlO server
- Going through CCV
- Video from recorded gray scale

Utilities:
- Hardware Interface
- Operating System
- System Library
- User's Program

[Link to resources to do this yourself]
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

link to resources to do this yourself
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

link to resources to do this yourself
Getting Multi-Touch up and Running: Demo #5

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

link to resources to do this yourself
Getting Multi-Touch up and Running: Demo #6

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

link to resources to do this yourself
Getting Multi-Touch up and Running

How to Make an Inexpensive Multitouch Pad

in Just a Few Minutes
Flash notes

- I had very inconsistent behavior using the Flash stack
- Regardless of Flash security settings I could not get things to work:
  - In Flash Player
  - In a browser
- I could get everything to work consistently using the full Flash program
  - Running flosc-2_0_5.jar on the command line
    - `java -jar ./flosc-2_0_5.jar 3333 3000`
  - Having CCV send output via OSC
Things you have to worry about

- Is your operating system 32-bit or 64-bit?
  - Your java virtual machine should match
- What Java version are you running?
  - Most recent is 1.7
  - MT4J on a Mac needs Java 1.6 JRE in Eclipse
- On Mac you have to make CCV executable
- Download flosc for demoing here
  - https://code.google.com/p/flosc/
- Which version of CCV?
  - 1.5 for Window
  - 1.2 for Mac
Getting Multi-Touch up and Running
Getting Multi-Touch up and Running
Getting Multi-Touch up and Running

JAVA APPLICATION

MT4J

TUIO SERVER

TUIO SIMULATOR

CCV

LIVE VIDEO

LED

iPHONE APP
Getting Multi-Touch up and Running

- **APPLICATION #1**
- **APPLICATION #N**

**TUIO SERVER**

- **HARDWARE**
- **MAT**

**ISSUES**
- CCV VERSION 1.5 WINDOWS
- 1.3/1.2 MAC/LINUX
Review Assignment
How to do what we did in class

- Download Community Core Vision
- http://nuicode.com/#projects
How to do what we did in class
How to do what we did in class

Choose the source

Source Image

Tracked Image

Choose the source

Source Properties
- Camera Settings (v)
- Flip Vertical (j)
- Flip Horizontal (h)

GPU Properties
- GPU Mode (g)

Communication
- Send Tuio Osc (t)
- Send Tuio Top | For Flash (f)
- Send Height & Width

Calibration
- Enter Calibration (c)

Files
- Save Settings (s)

Calc. Time [ms]: 5
Video [Res]: 320 x 240
Video [fps]: 30

Sending OSC messages to:
Host: 127.0.0.1
Port: 3333

Press spacebar to toggle fast mode
How to do what we did in class
How to do what we did in class

Subtract static background
How to do what we did in class
How to do what we did in class

Smooth the video
How to do what we did in class
How to do what we did in class

Eliminate “static”
How to do what we did in class

Source Image

Tracked Image

Boost the signal
How to do what we did in class
How to do what we did in class

Decide what counts as a blob

Source Image

Tracked Image

Decide what counts as a blob
How to do what we did in class
How to do what we did in class

Settings you may need to mess with
How to do what we did in class

- There is also a configuration file for CCV
  - config.xml
- GUI settings can be typed in this file
- It may get overwritten from the GUI
How to do what we did in class

```xml
<?xml version="1.0" ?>
<CONFIG>
  <!---=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

  YOU CAN MANUALLY EDIT THE FEATURES BELOW

  """"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

  -->

  --
  --
  --

  //----------- CAMERA SETTINGS // -->
</CAMERA_0>

  --
  --
  --

  //----------- NETWORK COMMUNICATION SETTINGS // -->
</NETWORK>

  --
  --
  --

  //----------- VIDEO SETTINGS // -->
</VIDEO>

  --
  --
  --

  //----------- BLOB SETTING // -->
</BLOBS>

```
How to do what we did in class

<!--//-------------------------------

DO NOT EDIT THE VALUES BELOW

//--------------------------------->

<APPLICATION>
  <TITLE>Community Core Vision</TITLE>
  <VERSION>1.2.0</VERSION>
</APPLICATION>

<WINDOW>
  <WIDTH>950</WIDTH>
  <HEIGHT>600</HEIGHT>
</WINDOW>

<BOOLEAN>
  <PRESSURE>4</PRESSURE>
  <LABELS>1</LABELS>
  <OUTLINES>1</OUTLINES>
  <LEARNBG>0</LEARNBG>
  <TUIO>1</TUIO>
  <VMIRROR>0</VMIRROR>
  <HMIRROR>0</HMIRROR>
  <HIGHPASS>1</HIGHPASS>
  <AMPLIFY>1</AMPLIFY>
  <SMOOTH>0</SMOOTH>
  <GPU>1</GPU>
  <DYNAMICBG>1</DYNAMICBG>
  <SNAPSHOT>0</SNAPSHOT>
  <MINMODE>0</MINMODE>
  <HEIGHTWIDTH>1</HEIGHTWIDTH>
  <OSCMODE>1</OSCMODE>
  <TCPMODE>0</TCPMODE>
  <TRACKDARK>1</TRACKDARK>
</BOOLEAN>

<INT>
  <MINMOVEMENT>2</MINMOVEMENT>
  <MINBLOBSIZE>11</MINBLOBSIZE>
  <MAXBLOBSIZE>86</MAXBLOBSIZE>
</INT>

<-- FILTERS SETTINGS -->

<THRESHOLD>43</THRESHOLD>
  <HIGHPASSBLUR>40</HIGHPASSBLUR>
  <HIGHPASSNOISE>8</HIGHPASSNOISE>
  <HIGHPASSTAMP>35</HIGHPASSTAMP>
  <SMOOTH>1</SMOOTH>
  <BGLERNRATE>450.099976</BGLERNRATE>
</INT>
</CONFIG>
How to do what we did in class

- For Flash Demos run the TUIO server standalone

```
java -jar flosc-2_0_5.jar 3333 3000

Set the Flash port to 3000
Set the OSC port to 3333
and start it

Launch CCV
Make sure "Send TUIO OSC" is checked

Launch Flash application
```
How to do what we did in class

- Download Flash Demos
- Originally here:
- Now on our course assignment page:
How to do what we did in class

• Download Flash Demos
How to do what we did in class

- Download TUIO Simulator
How to do what we did in class

- Run as a java program

```
$ java -jar TuioSimulator.jar
sending TUIO messages to 127.0.0.1:3333
```
How to do what we did in class

- Setup MT4J in Eclipse
Getting the software going

- Download and Install Eclipse
  - http://www.eclipse.org/
- Download MT4J
- Pick a directory and create a new workspace in Eclipse
Getting the software going

• Import a new archive Java Project in Eclipse
Getting the software going

- When you run an MT4J program the TUIO server is built in
  - Hello World
  - Test Scene
  - MTGestures
  - Space Scene
  - Fluid Simulator
Getting the software going

- The MT4J page has information on how to use MT4J
- It’s a visualization system
- It is complicated
- Start with an example
- Then modify it
How to go about doing multi-touch assignment

- Get an end-to-end demo running with off the shelf stuff
- Then slowly replace the pieces with your implementations
How to go about doing multi-touch assignment

- Get an end-to-end demo running with off the shelf stuff
- Then slowly replace the pieces with your implementations
How to go about doing multi-touch assignment

- Get an end-to-end demo running with off the shelf stuff
- Then slowly replace the pieces with your implementations
How to go about doing multi-touch assignment

- Get an end-to-end demo running with off the shelf stuff
- Then slowly replace the pieces with your implementations
How to go about doing multi-touch assignment

- Get an end-to-end demo running with off the shelf stuff
- Then slowly replace the pieces with your implementations
How to go about doing multi-touch assignment

- Get an end-to-end demo running with off the shelf stuff
- Then slowly replace the pieces with your implementations
• Options for increased awesomeness
  • Do a different application than a paint program
    • Make sure it needs multi-touch (talk to prof.)
  • Do a different input than a webcam
    • Make sure it is more than just a download (talk to prof.)
• Remember this can be a portfolio piece!
How to go about doing multi-touch assignment
• On Mac if you get:
  • LSOpenURLsWithRole() failed with error "-10810"
  • You need to make this file executable:
    • chmod a+x Community Core Vision.app/Contents/MacOS/Community Core Vision
  • If you run a flash demo you may have to give it permission to access locations on your hard drive