Wiki troubles

We’re investigating. Causes still unknown. Keep sending messages when things fail.
HW2

Due next Friday 5pm - get started now.
125 Cory access requests received by UCPD yesterday - should work today?
HW2 - Conceptual Overview

Decide on interaction scheme
Extract useful events from input
Design the menu widget
Link menu levels

Can be decoupled!
Decide on an Interaction Scheme

Example:

Hand as 1D pointer in Y to navigate
Swipe right in X to select, swipe left to back up a level.
Extract events from input

Example:

Use absolute Y position of the hand joint of the tracked skeleton for navigation. Map Y position into screen space.

Detect swipes by thresholding velocity and direction of movement.

Consider: What to do when there is no data? Noise?
Design the Menu Widget

You need different visual rendering of an item depending on a) previous state and b) current input.

State machines are a common way to implement this logic.
Design the Menu Widget

To determine which item the user is interacting with, you may need **hit testing**.
Link menu levels

Keep application state which menu you are currently showing, and add functions to swap out menu levels.
User Interface Components

Each component is an object with

Bounding box

Paint method for drawing itself

Drawn in the component’s coordinate system

Callbacks to process input events

Mouse clicks, typed keys

Java:
public void paint(Graphics g) {
  g.fillRect(…); // interior
  g.drawString(…); // label
  g.drawRect(…); // outline
}

Cocoa:
(void)drawRect:(NSRect)rect
Containment Hierarchy
Event Dispatch Loop

Event Queue
- Queue of input events

Event Loop (runs in dedicated thread)
- Remove next event from queue
- Determine event type
- Find proper component(s)
- Invoke callbacks on components
- Repeat, or wait until event arrives

Component
- Invoked callback method
- Update application state
- Request repaint, if needed

Mouse moved \((t_0, x, y)\)
Event Dispatch Loop

1) Events from input devices enter here

2) Event is added to FIFO event queue

3) Main loop processes one event per iteration

Apple, Cocoa Event-Handling Guide
Event Dispatch

Event Queue
- Mouse moved \((t_0, x, y)\)
- Mouse pressed \((t_1, x, y, 1)\)
- Mouse dragged \((t_2, x, y, 1)\)
- Key typed \((t_3, 'F1')\)
- ...
(queues and dispatches incoming events in a dedicated thread)

/* callback for TextArea */
public void mouseMoved(e) {
    // process mouse moved event
}
Mouse/Touch vs. Keyboard Events

Mouse Events are (usually) routed to the top-most (in z-order) visible component underneath the cursor using hit testing.
Exception: “captured” mouse events after beginning interaction

Keyboard events are (usually) routed to the component that has key focus.
Exceptions: keys that change focus, accelerator keys
Surface Computing
History: Video Place (1980s)

http://www.youtube.com/watch?v=dqZyZrN3Pl0 (short segment)

Myron Krueger’s Videoplace
History: Buchla Thunder (1990)
Digital Desk (Wellner, 1991)

Video:
Terminology (from Buxton)

- Touch tablets vs. touch screens
- Multi-finger vs. multi-hand
- Multi-person vs. multi-touch
- Beyond Points: Pressure, Pose, Gestures
- Fingers vs. Tokens vs. Arbitrary Objects
Status Quo
The FourBySix Table
The FourBySix Table
Table Infrastructure

- 12 MP Digital Still Camera
- XGA Projector with 45 Degree Mirror
- VGA Camera with IR-Pass Filter
- Acrylic tabletop with vellum cover
- IR Illuminant Panel

Dimensions:
- 120 cm
- 91 cm
- 180 cm
Top-down Image (12MP, ~60dpi)
Layered Model for Input Device Design

Hartmann, Follmer, Hanrahan
Physical Properties

A horizontal, flat sheet of semi-transparent material that the user touches or places objects / tokens upon
Sensing

Camera senses light in the infrared spectrum (~850nm)

In particular, it senses reflected light from an IR illuminant inside the device (+ambient IR, but that’s noise)

Fingers/objects that come in contact with the surface will reflect more light.

*Implies that the camera has to be positioned so it can image the entire touch surface area*
Electromagnetic Spectrum

- Gamma-rays
- X-rays
- Ultraviolet
- Visible
  - Near IR
- Infra-red
  - Thermal IR
  - Far IR
- Wavelength
  - 0.1 Å
  - 1 Å (0.1 nm)
  - 1 nm
  - 10 nm
  - 100 nm
  - 1000 nm (1 μm)
  - 10 μm
  - 100 μm
  - 1000 μm
- Frequency (Hz)
Sensing

Diffuse transmission and reflectance.
Coding

Sensed light is coded as an 8-bit grayscale image at 30fps.
Transformation

Image processing recipe:
Reverse lens distortion
Binarize image (thresholding, bg subtraction)
Find blobs (connected components)
Compute centroid, moments of blobs
Track blob ids across frames

Open source package:
Community Core Vision
UNDISTORT → NORMALIZE

EDGE IMAGE: FIND CONTACTS

SHAPE IMAGE: ESTIMATE POS, ANGLE
Transformation

Source Image

Tracked Image

Community core vision
Device Abstraction

The device from the application programmer’s point of view:
Object down/move/up events
Touch objects: location, size, orientation
Fiducial markers: id, orientation

e.g.: Surface API, TUIO, Win7 Touch,...
And then you finally get to interaction design...
Searching

B. Hartmann, M. Morris, H. Benko, A. Wilson,
*Augmenting Interactive Surfaces with Mice & Keyboards*, Proc. UIST 2009

Pictionaire: Supporting collaborative design work by integrating physical and digital artifacts
Drag-Off Gesture
Snap-To-Fit Light Table
Capture + Annotate
Common Issues

Attention
Requires free use of (both) hands and eyes.

Precision
C:D ratio and fat fingers

Gesture vocabulary
Multi-point Gestures

Select Single₁: tap
Select Single₂: lasso
Select Group₁: hold and tap
Select Group₂ and Select Group₃: Use Select Single₁ or Select Single₂ on all items in the group.

Move₁: drag
Move₂: jump
Pan: drag hand
Rotate: drag corner
Object jumps to index finger location.
Finger touches corner to rotate.

Cut: slash
Paste₁: tap
Paste₂: drag from offscreen
Paste₃: Use Move₂ with off-screen source and on-screen destination.
After duplicating, source object is no longer selected.

Delete₁: drag offscreen
Delete₂: Use Move₁ with on-screen source and off-screen destination.
Accept: draw check
Reject: draw 'X'
Help: draw '?'
Reject₂: Reject₂: If rejecting an object/dialog with an on-screen representation, use Delete₁ or Delete₂.
Menu: pull out
Undo: scratch out

Issues at Table Size

Orientation
Reach & Territoriality
Group interaction techniques (identity?)
Issues at Table-Size
Issues at Tab/Pad size

Two hands required

Fat Fingers:
Precision, Occlusion

Use outside?
The “Fat Finger” Problem

Graphics: Patrick Baudisch, nanoTouch
A Software Solution

scenario 1: ambiguous target due to occlusion

(a) (b) (c) (d) (e)

Graphics: D. Vogel, P. Baudisch - Shift
A Hardware Solution: Use the Backside

pointer

Graphics: Patrick Baudisch, nanoTouch
Use outside?

New Kindle Pool Ad