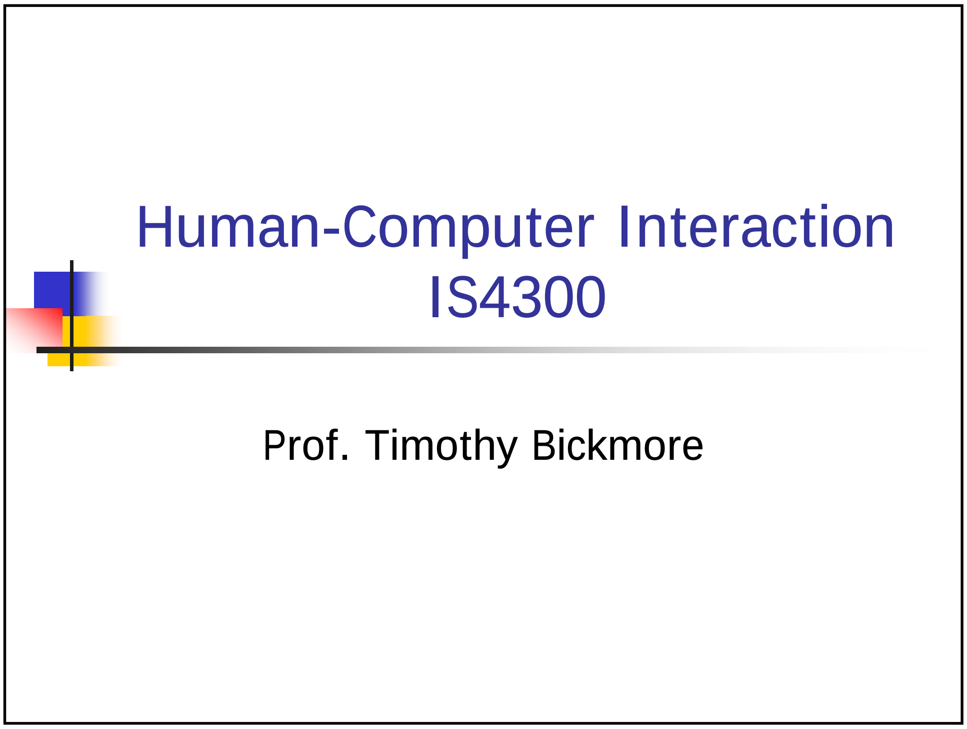
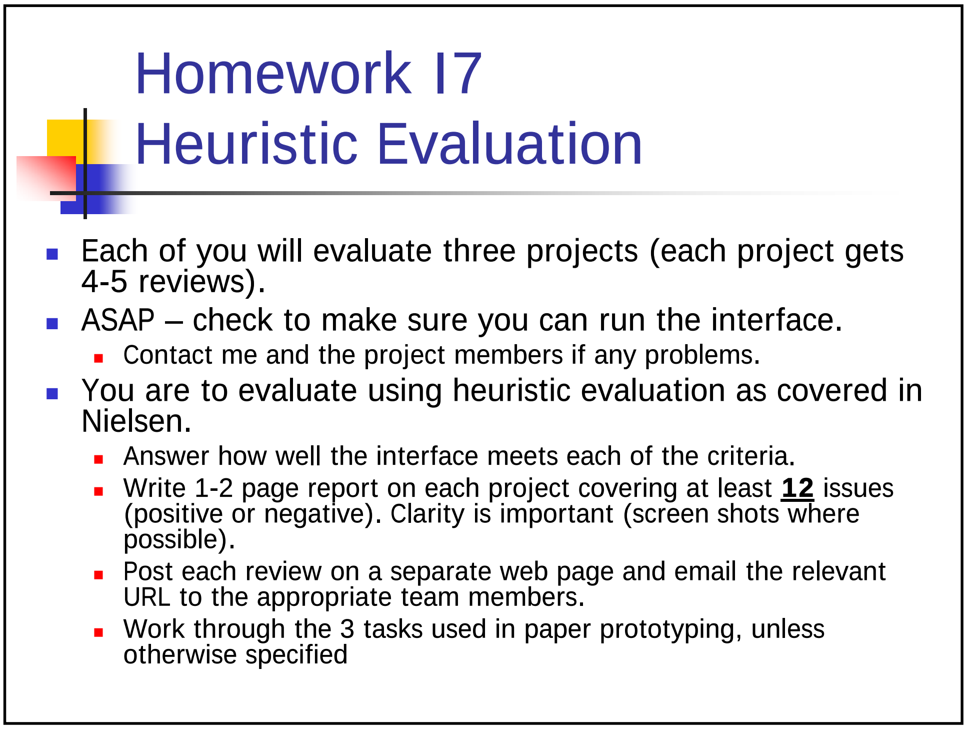


Human-Computer Interaction IS4300

Prof. Timothy Bickmore



Homework I7 Heuristic Evaluation

- Each of you will evaluate three projects (each project gets 4-5 reviews).
- ASAP – check to make sure you can run the interface.
  - Contact me and the project members if any problems.
- You are to evaluate using heuristic evaluation as covered in Nielsen.
  - Answer how well the interface meets each of the criteria.
  - Write 1-2 page report on each project covering at least **12** issues (positive or negative). Clarity is important (screen shots where possible).
  - Post each review on a separate web page and email the relevant URL to the appropriate team members.
  - Work through the 3 tasks used in paper prototyping, unless otherwise specified

HUMAN-COMPUTER INTERACTION THIRD EDITION

DIX FINLAY ABOWD BEALE

chapter 20

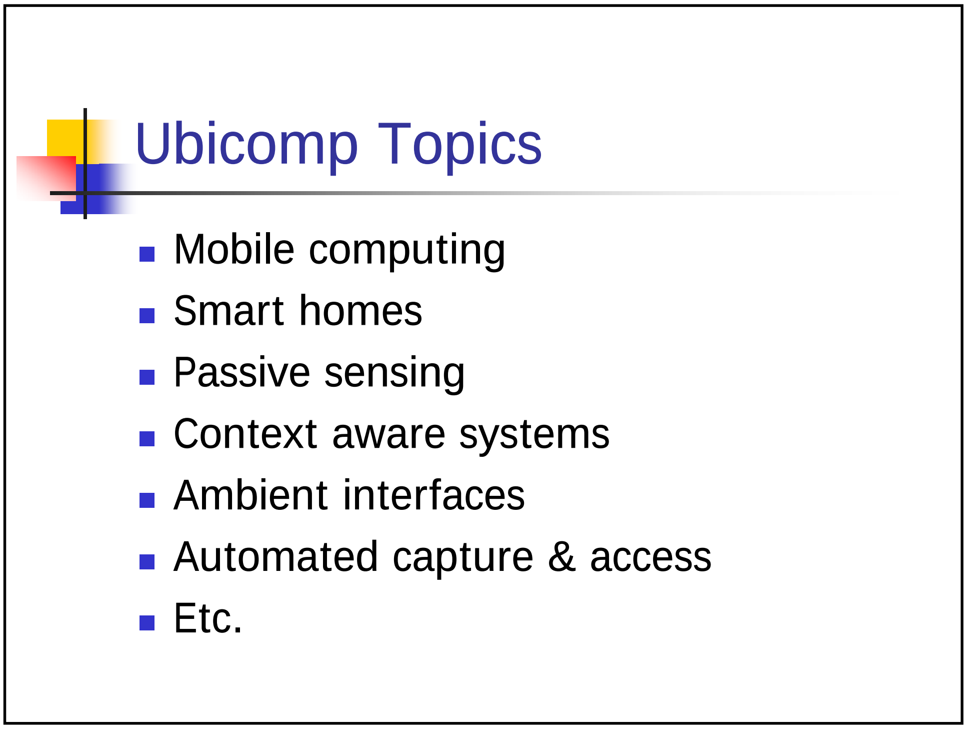
You must be joking

- Ubicomp
- VR?
- Mobile

Ubicomp

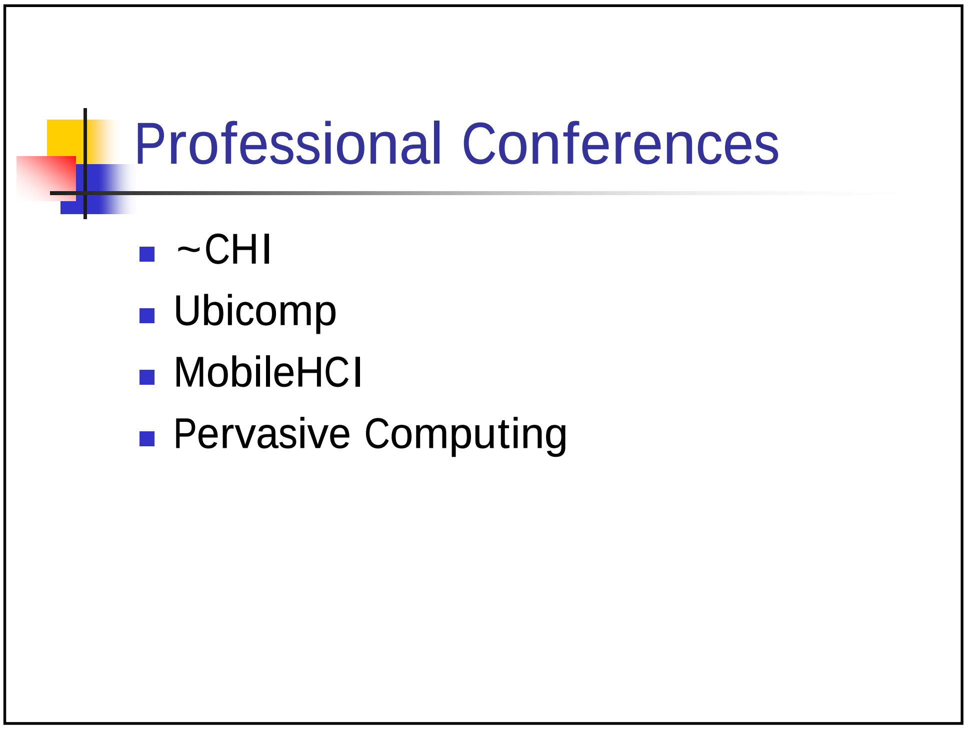
- Ubiquitous Computing, aka
- Pervasive Computing

- “Computing off the desktop”
- Mark Weiser @ Xerox PARC 1990’s



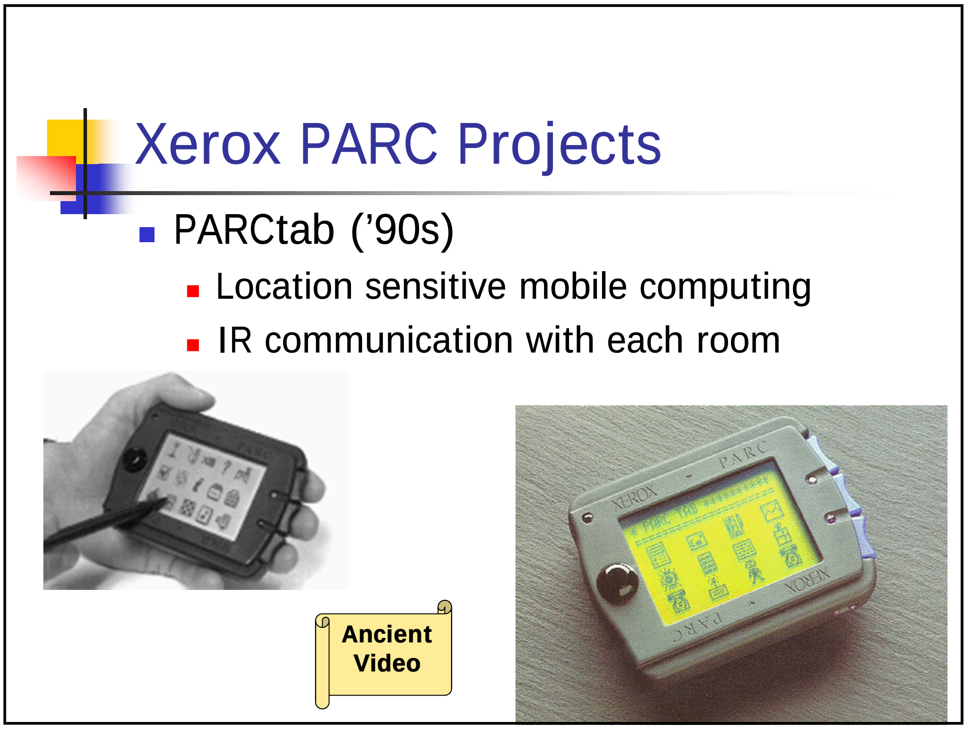
Ubicomp Topics

- Mobile computing
- Smart homes
- Passive sensing
- Context aware systems
- Ambient interfaces
- Automated capture & access
- Etc.



Professional Conferences

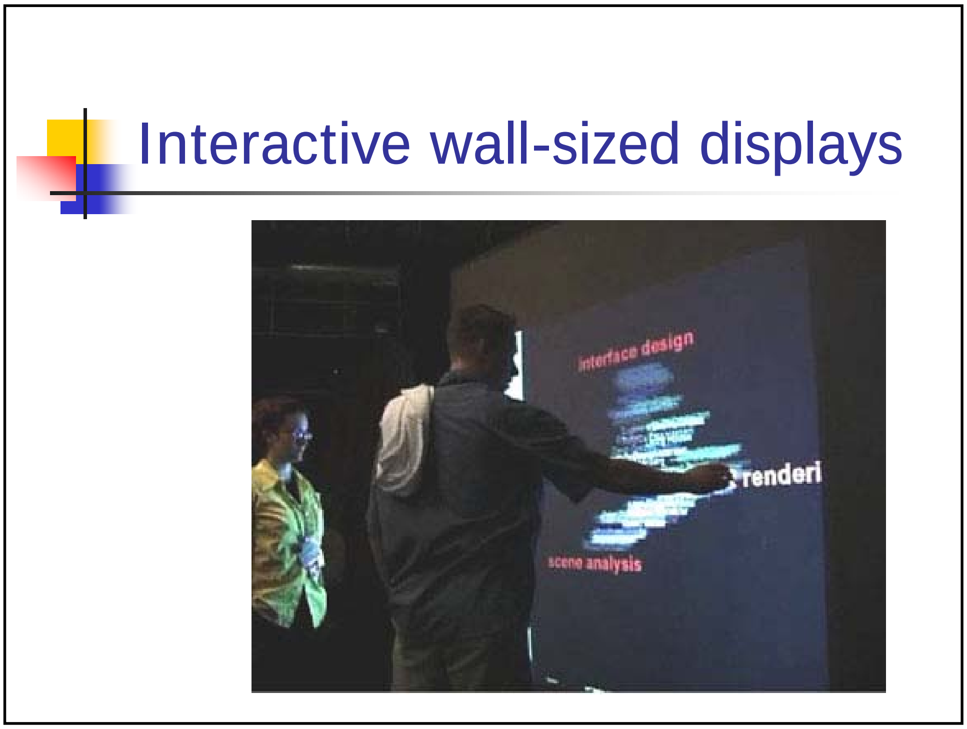
- ~CHI
- Ubicomp
- MobileHCI
- Pervasive Computing



Xerox PARC Projects

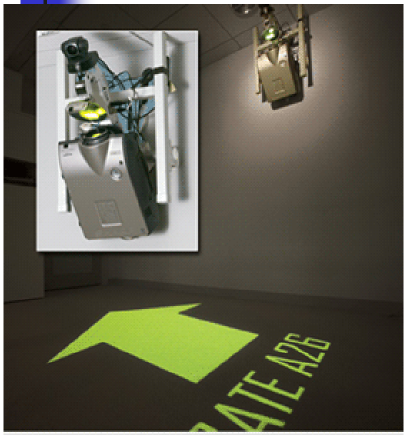
- PARCtab ('90s)
  - Location sensitive mobile computing
  - IR communication with each room

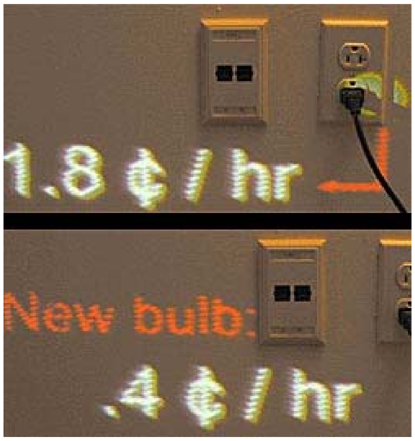
Ancient Video



Interactive wall-sized displays

IBM Anywhere Display





PlaceLab

Evaluation challenges?

Ambient Interfaces: Ambient Orb

Ambientdevices.com

Pinwheels – Example Ambient Interface

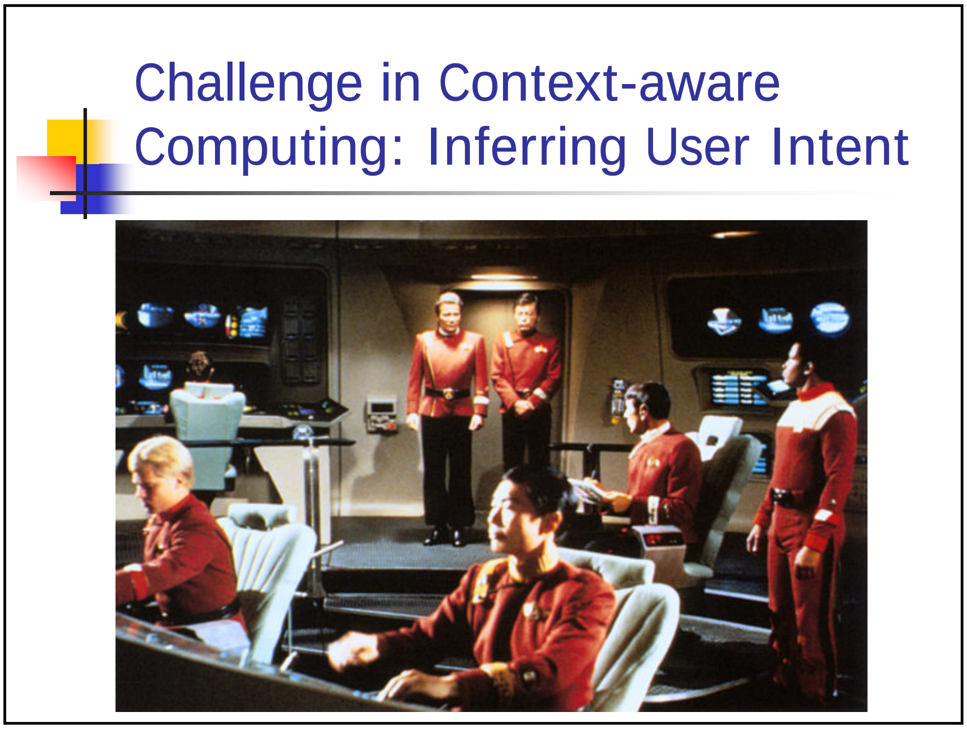
Automated Capture & Access

Vannevar Bush “memex” – Brian Clarkson “Life Patterns”

A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory. Vannevar Bush, 1945

Context-Aware Computing

- Apps that automatically respond to, or incorporate, context
  - Location
  - Time
  - Activity
  - Who
- Current examples?
- Trying to guess ‘user intent’ is notoriously difficult…

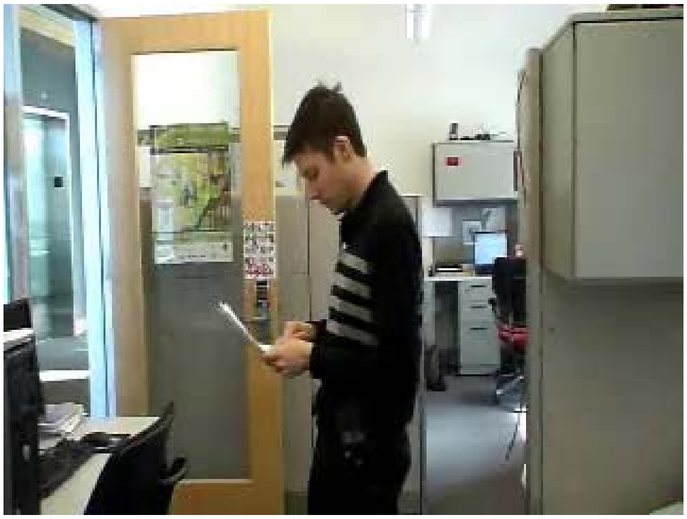


Challenge in Context-aware Computing: Inferring User Intent

Challenge in mobile/ubicomp/everyday: how to monitor and interact proactively

Okay , tomorrow night at 06:20 PM it is ! Talk to you then .

Challenge in mobile/ubicomp/everyday: how to monitor and interact proactively



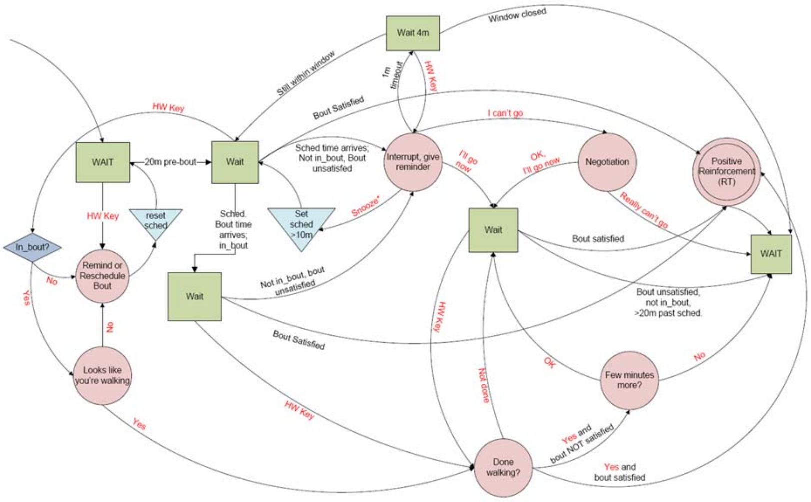
Context-awareness Study

- Compared automatic sensing of walking to explicit user signaling of walk start & end.
- Eight subject, 2-treatment (4day ea), within-subjects design.
- Results:
  - Awareness led to greater social bonding, but less walking.
  - Likely due to low perceived reliability & effective commitment of walk signaling.

Wearable Agent Field Study

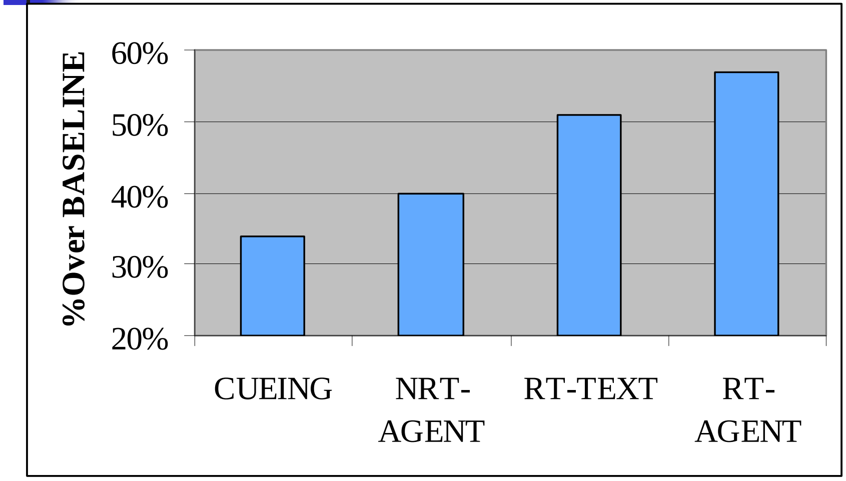
- Primary hypothesis: real-time intervention more effective than retrospective.
- 5-week, 5-treatment within-subjects design
- 100 free-living, sedentary adults

Challenge in mobile/ubicomp/everyday: how to monitor and interact proactively



Just-in-Time Study

N=51



How do our models of interaction need to change for ubicomp?

- Model Human Processor / Norman’s Interaction Model, Assumes:
  - single user
  - uninterrupted task
  - state either on screen or in working memory
- Alternate theoretical frameworks
  - Activity theory, Distributed cognition, Ethnography

virtual and augmented reality

VR - technology & experience
web, desktop and simulators
AR – mixing virtual and real

virtual reality technology

- headsets allow user to "see" the virtual world
- gesture recognition achieved with DataGlove (lycra glove with optical sensors that measure hand and finger positions)
- eyegaze allows users to indicate direction with eyes alone

Immersion

- VR
  - computer simulation of the real world
    - mainly visual, but sound, haptic, gesture too
  - experience life-like situations
    - too dangerous, too expensive
  - see unseen things:
    - too small, too large, hidden, invisible
      - e.g. manipulating molecules
- the experience
  - aim is immersion, engagement, interaction

Coming out of the lab...finally

Mobile UIs

User Interface Design and Evaluation
Debbie Stone
Caroline Jarrett
Mark Woodroffe
Shailey Minocha
Patterns for Interaction Design
Designing Mobile Interfaces
O'REILLY
iOS Human Interface Guidelines
Developer

Stone: Design Issues for Information Appliances

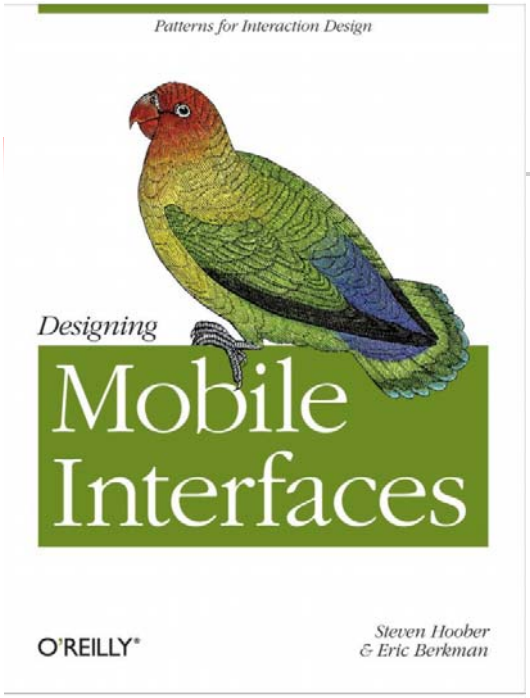
- Portability
- General Purpose versus Special Purpose
- Connectedness
- The Commercial Environment
  - Consumer products
  - Aesthetics of design even more important

Stone: “Block of Wood” prototyping

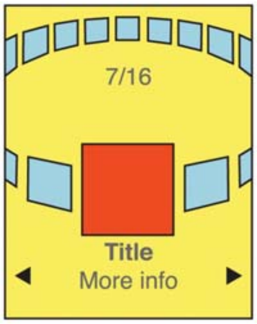
- Ask users to carry around
- Use as if a cell phone
- Jeff Hawkins used to prototype the Palm Pilot

UI Design Guidelines for Handheld Devices (Stone Table 18.1)

- Select vs. Type (typing is hard)
- Be consistent / Consistency between platforms
  - (External) Consistency
- Design stability
  - Robustness of connectivity
- Feedback
- Forgiveness - Error correction
- Use Metaphors
- Clickable graphics should look clickable (Visibility)
- Use icons to clarify concepts (Visiblity)

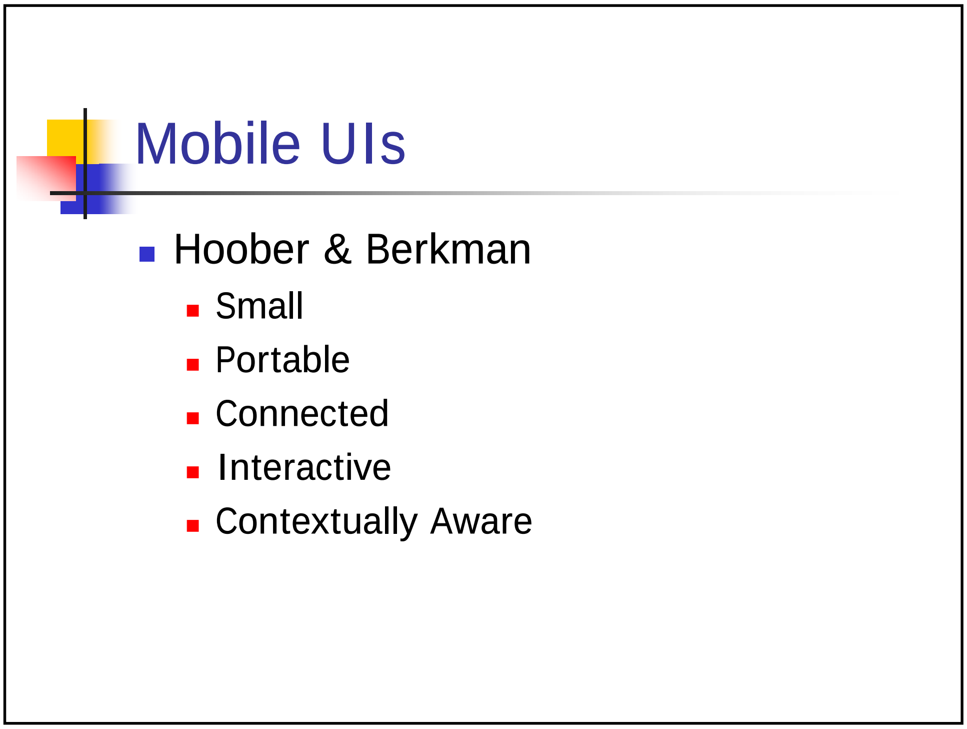


- 584 pages
- 11/11
- Design patterns



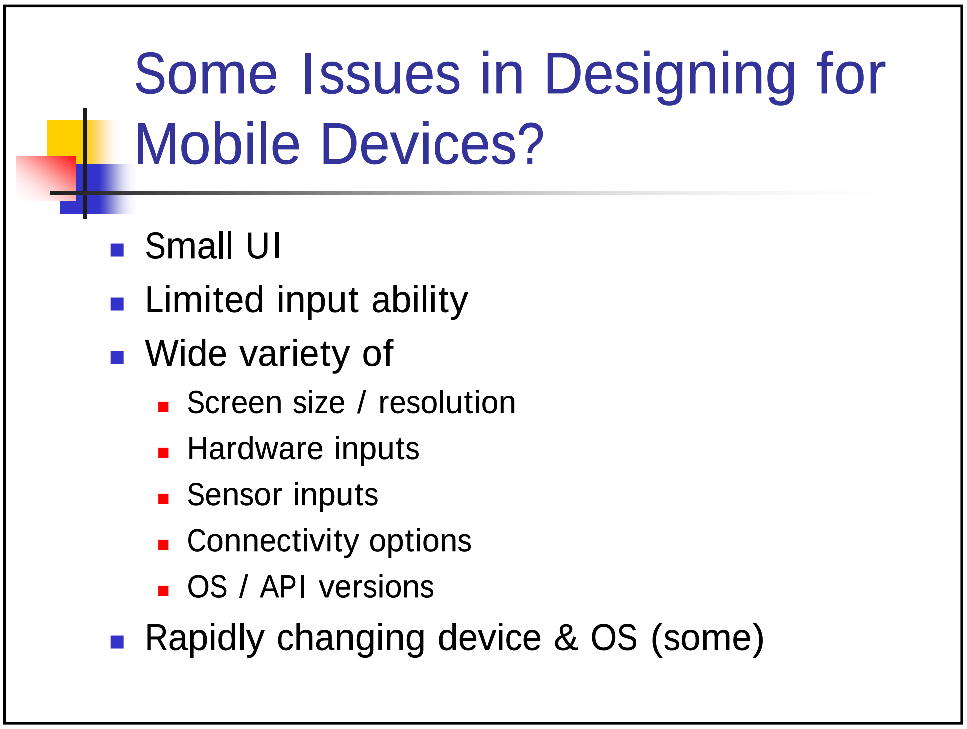
Mobile Interfaces

- What is a mobile UI?
- What is different in design methodology relative to desktop GUIs?



Mobile UIs

- Hoober & Berkman
  - Small
  - Portable
  - Connected
  - Interactive
  - Contextually Aware



Some Issues in Designing for Mobile Devices?

- Small UI
- Limited input ability
- Wide variety of
  - Screen size / resolution
  - Hardware inputs
  - Sensor inputs
  - Connectivity options
  - OS / API versions
- Rapidly changing device & OS (some)

Principles of Mobile Design
Hooker & Berkman

- Respect User-Entered Data
  - Input is hard
- Mobiles are Personal
  - Assume one user, with personal data active
- Lives Take Precedence
  - Don’t interrupt unless necessary
- Must Work in all Contexts
  - E.g., screen brightness
- Use Sensors & Smarts
  - Do things for the user when possible
- User Tasks Take Precedence
  - User-directed interaction
- Consistency (external & internal)
- Respect Information (present data precisely)

Page Layout Guidelines

- Mobile screen real estate is valuable.
  - Skip unnecessary banners, images, graphics (“administrative clutter” – Tufte)
- Consistent & simple navigation elements
- Keep everything as simple as possible
- For Serious tools (vs. games)
  - Minimal number of colors
  - Keep UI data-centered

Design Methodology

Hooker & Berkman

- Storyboard UIs (as before)
- Additional considerations
  - Gestural interface & finger size
  - Use contexts
  - Asynchronous events
  - Use of sensors, devices
  - Different display sizes, orientations (e.g., auto-switch landscape / portrait)

iPhone

https://developer.apple.com/library/ios/navigation/

iOS Human Interface Guidelines

Developer

Theme in iOS7:

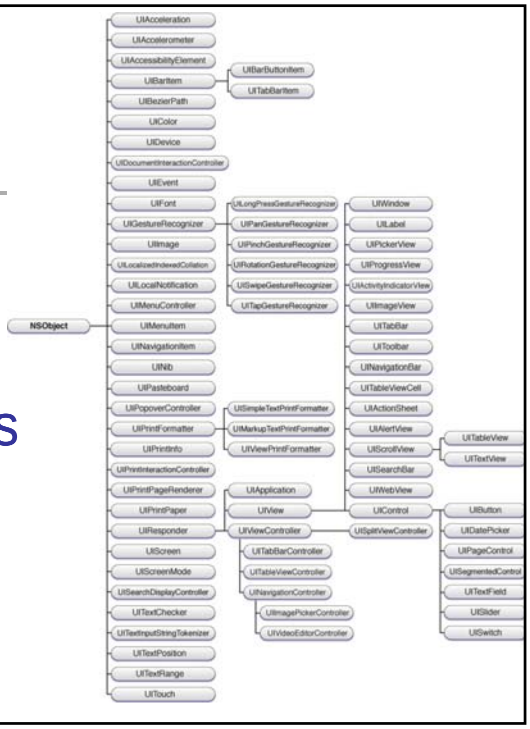
- Minimize UI ‘adornments’
- Focus on content

- e.g. ‘borderless buttons’

iPhone 5s / iOS 7

- 4” dia, 1136 x 640 screen resolution
- Multi-touch screen
- Audio output
- Vibration output
- Front and rear cameras
- Accelerometer (orientation, motion)
- Connectivity: GSM, CDMA, Wi-Fi, Bluetooth, etc.
- GPS, compass
- 1M+ apps

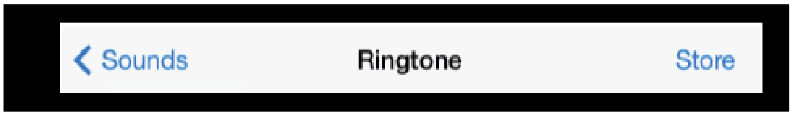
iOS 7 UI Widgets



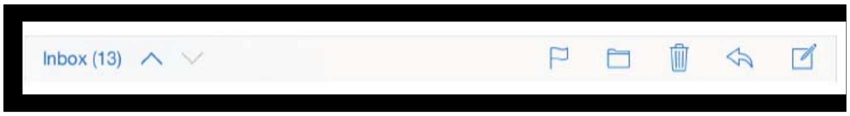
- NSObject
    - UIAcceleration
    - UIAccelerometer
    - UIAccessibilityElement
    - UIBarItem
        - UIBarButtonItem
        - UITabBarItem
    - UIBezierPath
    - UIColor
    - UIDevice
    - UIDocumentInteractionController
    - UIEvent
    - UIFont
    - UIGestureRecognizer
        - UILongPressGestureRecognizer
        - UIPanGestureRecognizer
        - UIPinchGestureRecognizer
        - UIRotationGestureRecognizer
        - UISwipeGestureRecognizer
        - UITapGestureRecognizer
    - UIImage
    - UILocalizedIndexedCollation
    - UILocalNotification
    - UIMenuController
    - UIMenuItem
    - UINavigationItem
    - UINib
    - UIPasteboard
    - UIPopoverController
    - UIPrintFormatter
        - UISimpleTextPrintFormatter
        - UIMarkupTextPrintFormatter
        - UIViewPrintFormatter
    - UIPrintInfo
    - UIPrintInteractionController
    - UIPrintPageRenderer
    - UIPrintPaper
    - UIResponder
        - UIApplication
        - UIView
            - UIWindow
            - UILabel
            - UIPickerView
            - UIProgressView
            - UIActivityIndicatorView
            - UIImageView
            - UITabBar
            - UIToolbar
            - UINavigationBar
            - UITableViewCell
            - UIActionSheet
            - UIAlertView
            - UIScrollView
                - UITableView
                - UITextView
            - UISearchBar
            - UIWebView
            - UIControl
                - UIButton
                - UIDatePicker
                - UIPageControl
                - UISegmentedControl
                - UITextField
                - UISlider
                - UISwitch
        - UIViewController
            - UISplitViewController
            - UITabBarController
            - UITableViewController
            - UINavigationController
                - UIImagePickerController
                - UIVideoEditorController
    - UIScreen
    - UIScreenMode
    - UISearchDisplayController
    - UITextChecker
    - UITextInputStringTokenizer
    - UITextPosition
    - UITextRange
    - UITouch

iOS Widgets

Navigation Bar
To traverse hierarchical information

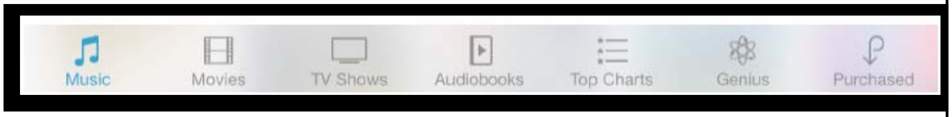


Tool Bar
Controls that perform actions related to objects in the screen or view.



iOS Widgets

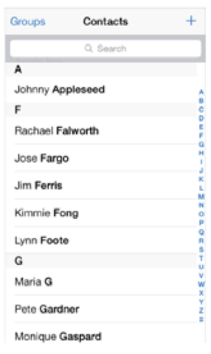
Tab Bar
ability to switch between different subtasks, views, or modes.



iOS Widgets

Table View
Display objects in single column

Simple

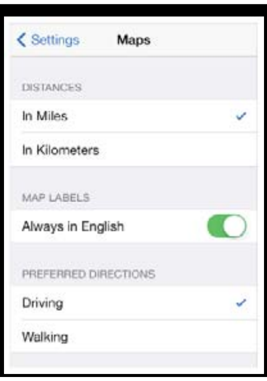


Groups Contacts +

Search

A
Johnny Appleseed
F
Rachael Falworth
Jose Fargo
Jim Ferris
Kimmie Fong
Lynn Foote
G
Maria G
Pete Gardner
Monique Gaspard

Grouped



< Settings Maps

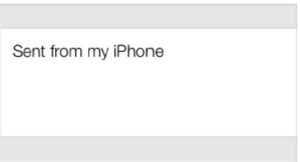
DISTANCES
In Miles ✓
In Kilometers

MAP LABELS
Always in English ●

PREFERRED DIRECTIONS
Driving ✓
Walking

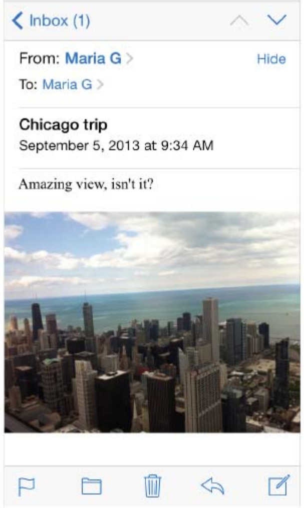
iOS Widgets

Text View
accepts & displays lines of text



Sent from my iPhone

Web View
displays HTML



< Inbox (1)

From: Maria G > Hide
To: Maria G >

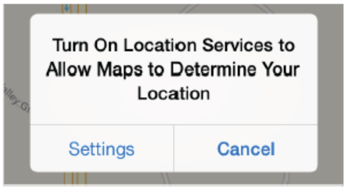
Chicago trip
September 5, 2013 at 9:34 AM

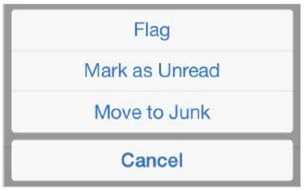
Amazing view, isn't it?

iOS Widgets

Action Sheet

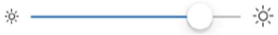
Alerts


Turn On Location Services to Allow Maps to Determine Your Location

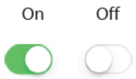


iOS Widgets

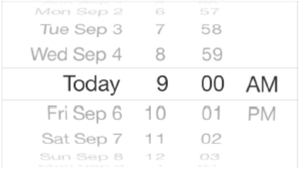
Slider



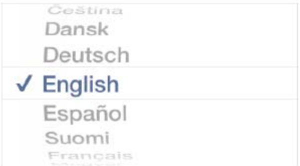
Switch



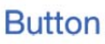
Date/Time Picker



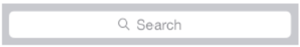
Picker



Button



Search bar

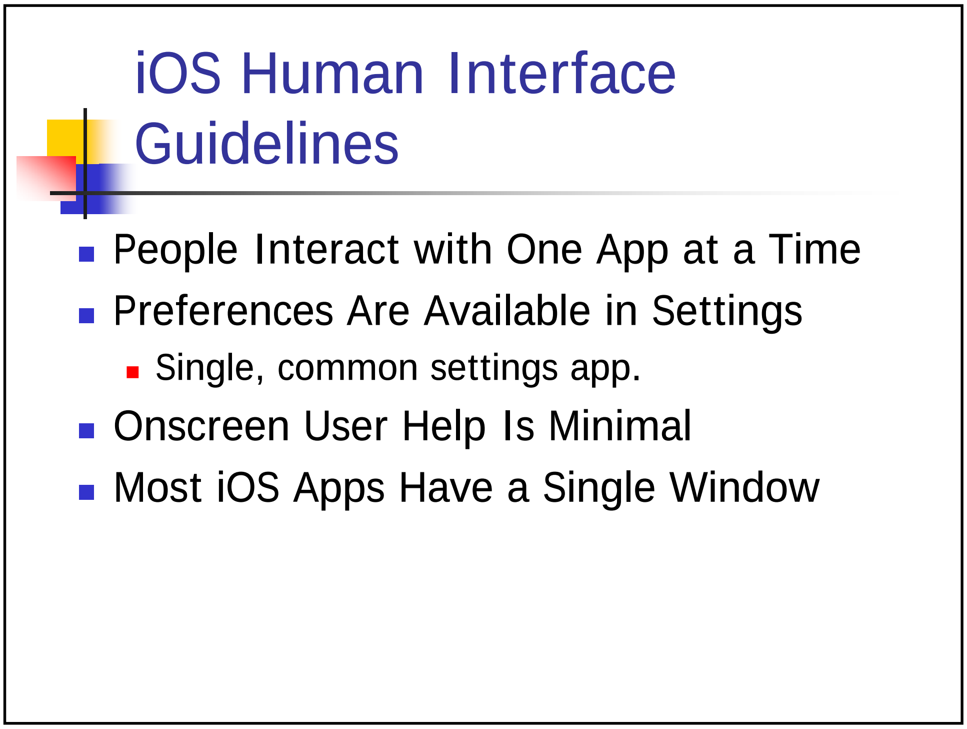


iOS Human Interface Guidelines

- The Display Is Paramount
  - The display of an iOS-based device is at the heart of the user’s experience.
  - The display encourages people to forget about the device and to focus on their content or task.
- Device Orientation Can Change

Apps Respond to Gestures, Not Clicks

- Tap
  - To press or select a control or item
- Drag
  - To scroll or pan; To drag an element.
- Flick
  - To scroll or pan quickly.
- Swipe
  - To reveal hidden content / widgets.
- Double tap
  - Zoom in and center; Zoom out.
- Pinch
  - Zoom in ; Zoom out



iOS Human Interface Guidelines

- People Interact with One App at a Time
- Preferences Are Available in Settings
  - Single, common settings app.
- Onscreen User Help Is Minimal
- Most iOS Apps Have a Single Window

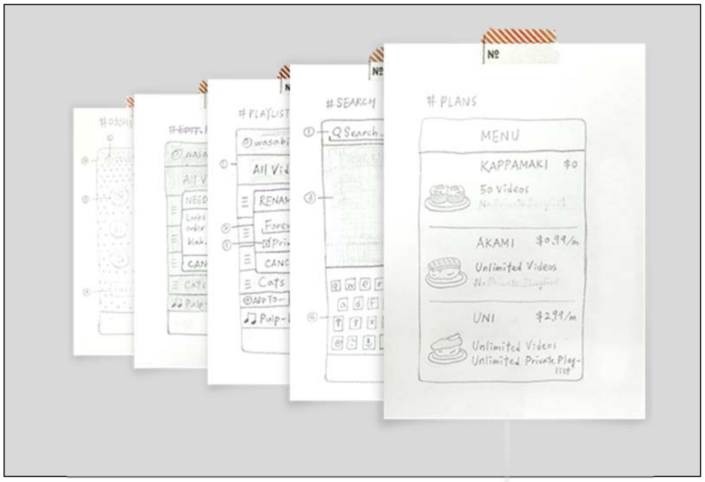
iOS Design Methodology

1. Create an App Definition Statement (aka requirements analysis)
   1. List All the Features (tasks) You Think Users Might Like
   2. Determine Who Your Users Are
   3. Filter the Feature List Through the Audience Definition

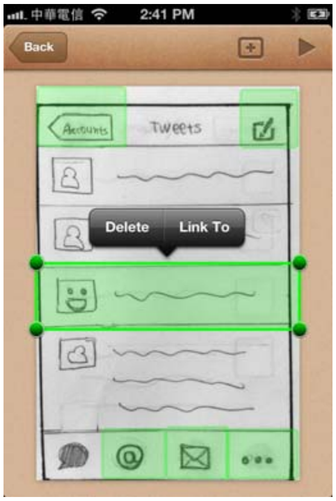
iOS Design Methodology

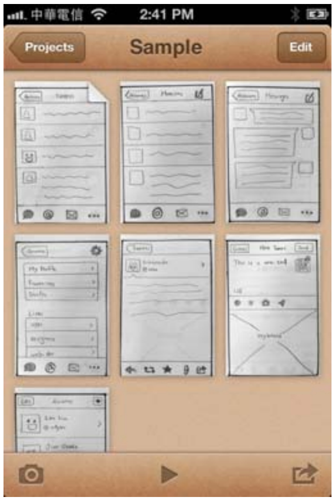
2. Design the App for the Device
   - Follow iOS UI Paradigms
     - Controls should look tappable
     - App structure should be clean and easy to navigate
     - User feedback should be subtle, but clear
   - Reconsider Web-Based Designs
     - Focus your app – narrow set of tasks
     - Make sure your app lets people do something – interactive
     - Design for touch
     - Let people scroll
     - Relocate the homepage icon

Remember SILK? Try POP – Prototyping On Paper



Remember SYLK?
Try POP – Prototyping On Paper





Usability Testing for Mobile

How to do usability studies of *in situ* mobile users?

Oulasvirta & Nyyssönen, “Flexible Hardware Configurations for Studying Mobile Usability”

Mobile Usability Lab…

Example Apps

30 Superb Examples of iPhone Interface Design

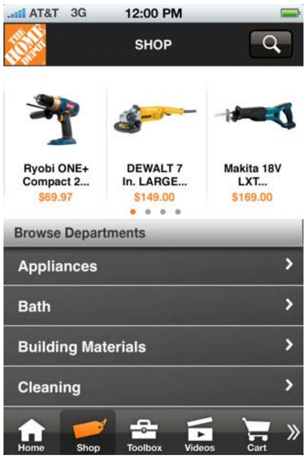
topDesign mag

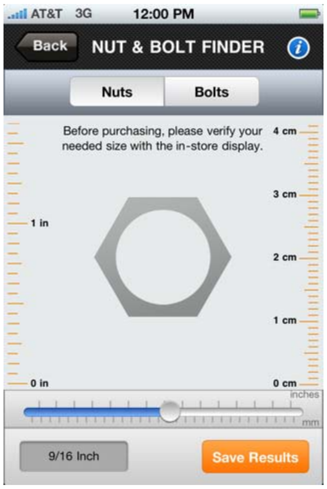
Simplicity

support few tasks – but do them well

Home Depot

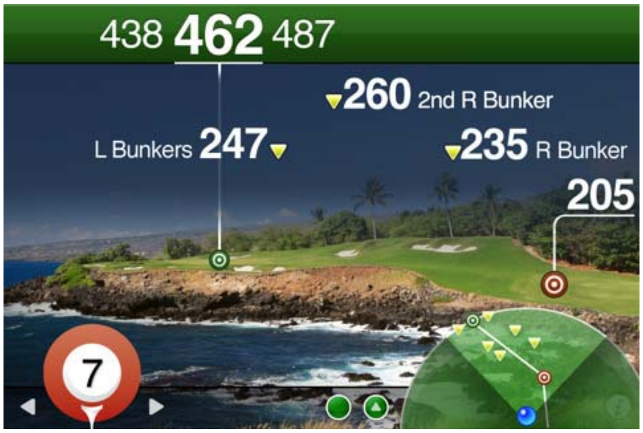
research and purchase over 100,000 products



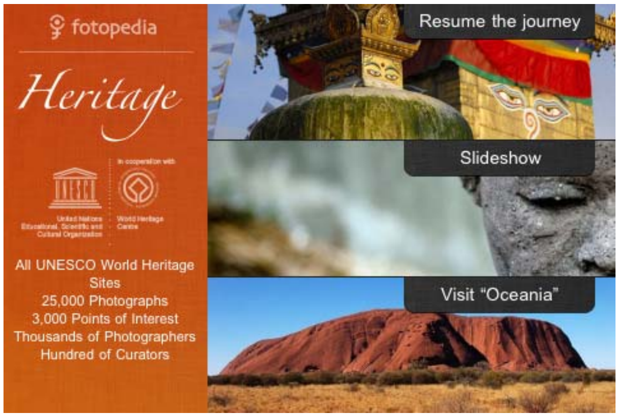


Before purchasing, please verify your needed size with the in-store display.

Golfscape
Augmented Reality Rangefinder



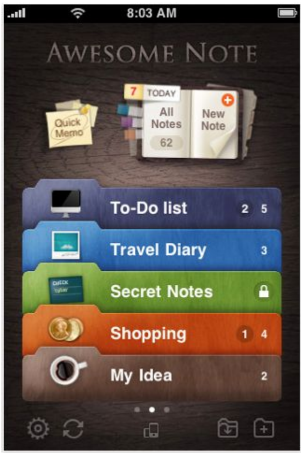
Fotopedia Heritage

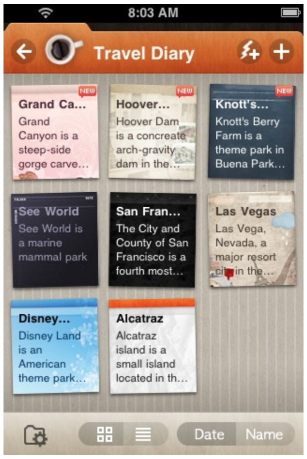


All UNESCO World Heritage Sites
25,000 Photographs
3,000 Points of Interest
Thousands of Photographers
Hundred of Curators

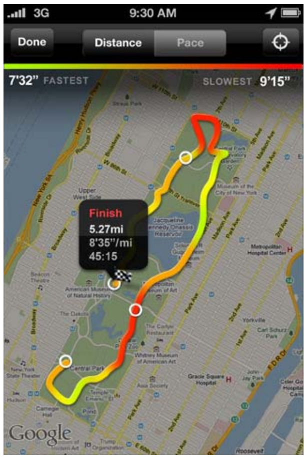
Awesome Note

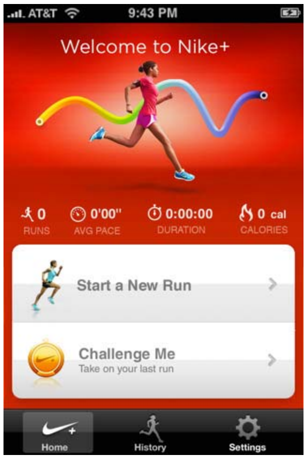
an innovative note taking application and to-do manager that allows you to combine notes with to-do flexibility



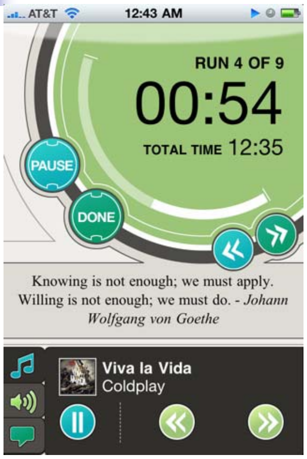

8:03 AM
Travel Diary
Grand Ca…
Grand Canyon is a steep-side gorge carve…
Hoover…
Hoover Dam is a concreate arch-gravity dam in the…
Knott's…
Knott's Berry Farm is a theme park in Buena Park…
See World
See World is a marine mammal park
San Fran…
The City and County of San Francisco is a fourth most…
Las Vegas
Las Vega, Nevada, a major resort city in the…
Disney…
Disney Land is an American theme park…
Alcatraz
Alcatraz island is a small island located in th…
Date
Name

Nike+ GPS

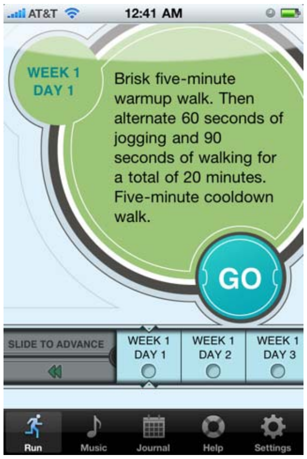




Couch to 5K



Knowing is not enough; we must apply. Willing is not enough; we must do. - *Johann Wolfgang von Goethe*



Brisk five-minute warmup walk. Then alternate 60 seconds of jogging and 90 seconds of walking for a total of 20 minutes. Five-minute cooldown walk.

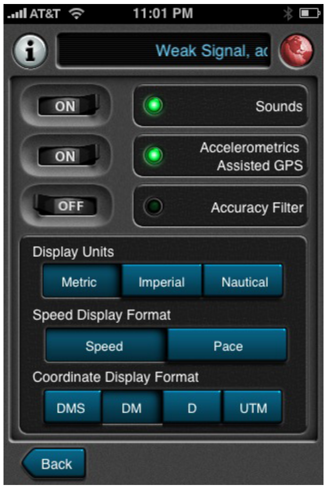
Your favorite (well-designed) apps?

Bad Examples..

Komarov, “iPhone Apps Design Mistakes”, *smashing magazine*

Olsen, “10 Surefire Ways to Screw Up Your iPhone App”, *UX magazine*

Motion X GPS Settings



AT&T 11:01 PM

Weak Signal, ac

ON Sounds

ON Accelerometrics Assisted GPS

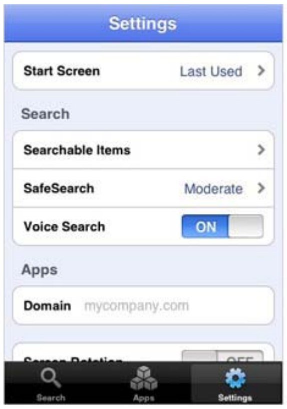
OFF Accuracy Filter

Display Units: Metric | Imperial | Nautical

Speed Display Format: Speed | Pace

Coordinate Display Format: DMS | DM | D | UTM

Back



Settings

Start Screen: Last Used

Search

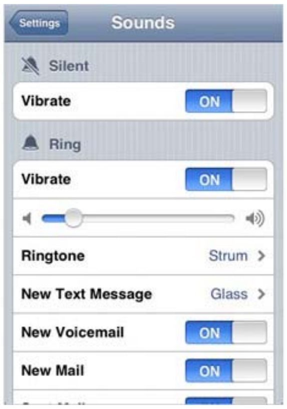
Searchable Items

SafeSearch: Moderate

Voice Search: ON

Apps

Domain: mycompany.com



Settings Sounds

Silent

Vibrate: ON

Ring

Vibrate: ON

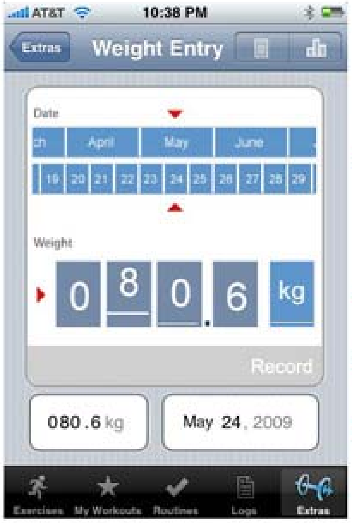
Ringtone: Strum

New Text Message: Glass

New Voicemail: ON

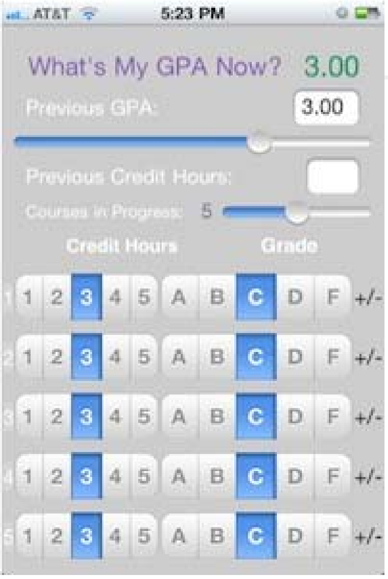
New Mail: ON

iFitness – weight entry



080.6 kg

May 24, 2009



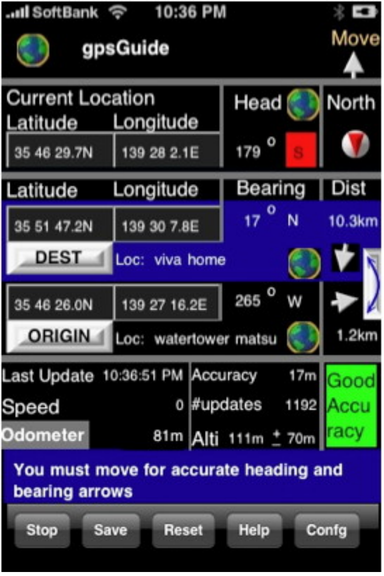
What's My GPA Now? 3.00

Previous GPA: 3.00

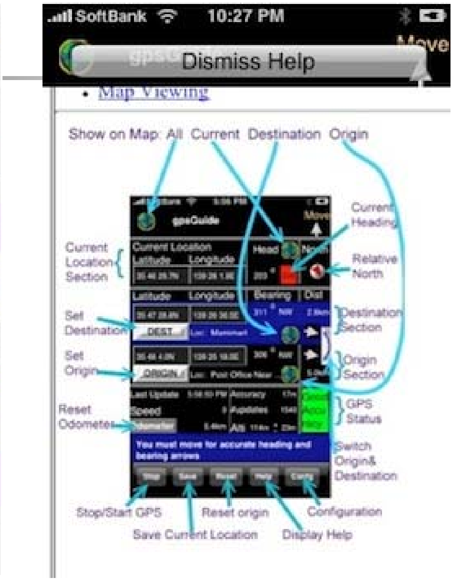
Previous Credit Hours:

Courses in Progress: 5

Help Screen

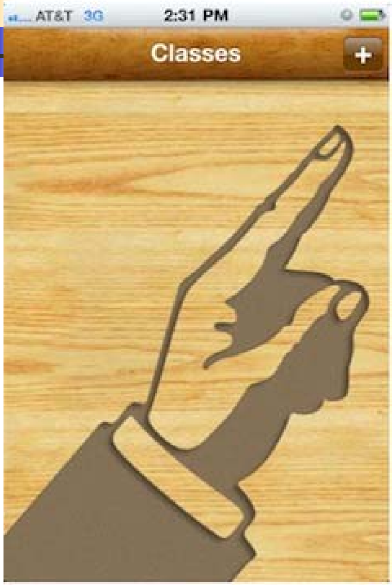


You must move for accurate heading and bearing arrows

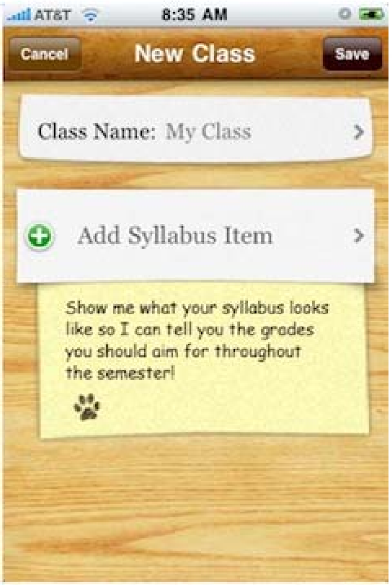


Instead of help manual…

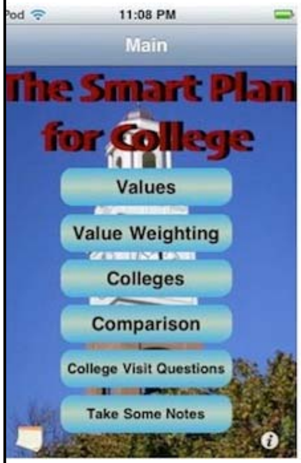
Graphical instructions

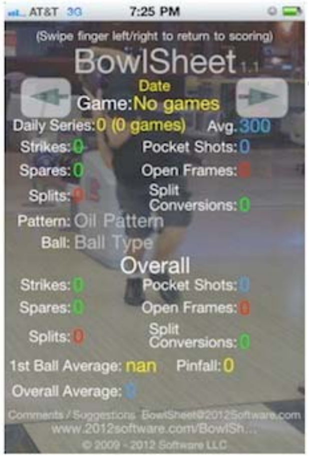


In-context text help



Show me what your syllabus looks like so I can tell you the grades you should aim for throughout the semester!





AT&T 3G 7:25 PM

(Swipe finger left/right to return to scoring)

BowlSheet 1.1

Date

Game: No games

Daily Series: 0 (0 games) Avg. 300

Strikes: 0 Pocket Shots: 0

Spares: 0 Open Frames: 0

Splits: 0 Split Conversions: 0

Pattern: Oil Pattern

Ball: Ball Type

Overall

Strikes: 0 Pocket Shots: 0

Spares: 0 Open Frames: 0

Splits: 0 Split Conversions: 0

1st Ball Average: nan Pinfall: 0

Overall Average: 0

Comments / Suggestions BowlSheet@2012Software.com
www.2012software.com/BowlSh...
© 2009 - 2012 Software LLC

Basics of Graphic Design

Contrast: poor contrast between the background and the content.

Repetition: Last two rows in the left example break the font size pattern, and the right example doesn't have much repetition at all

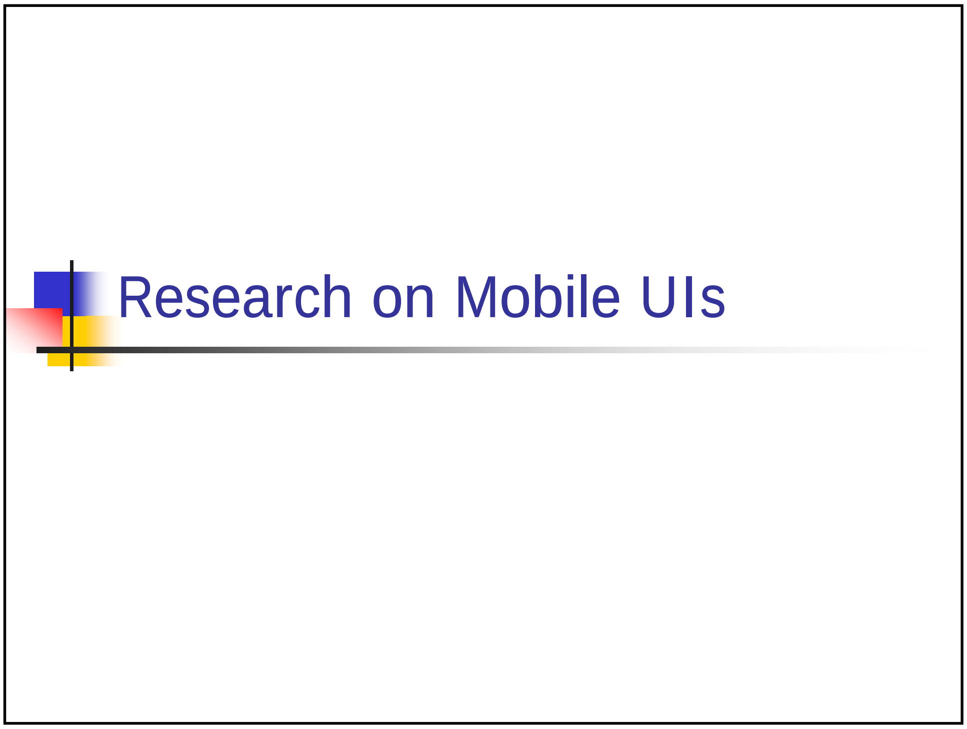
Alignment: Left alignment generally looks more professional than centered alignment (left) or no alignment (right).

Proximity: Very weak spatial groupings

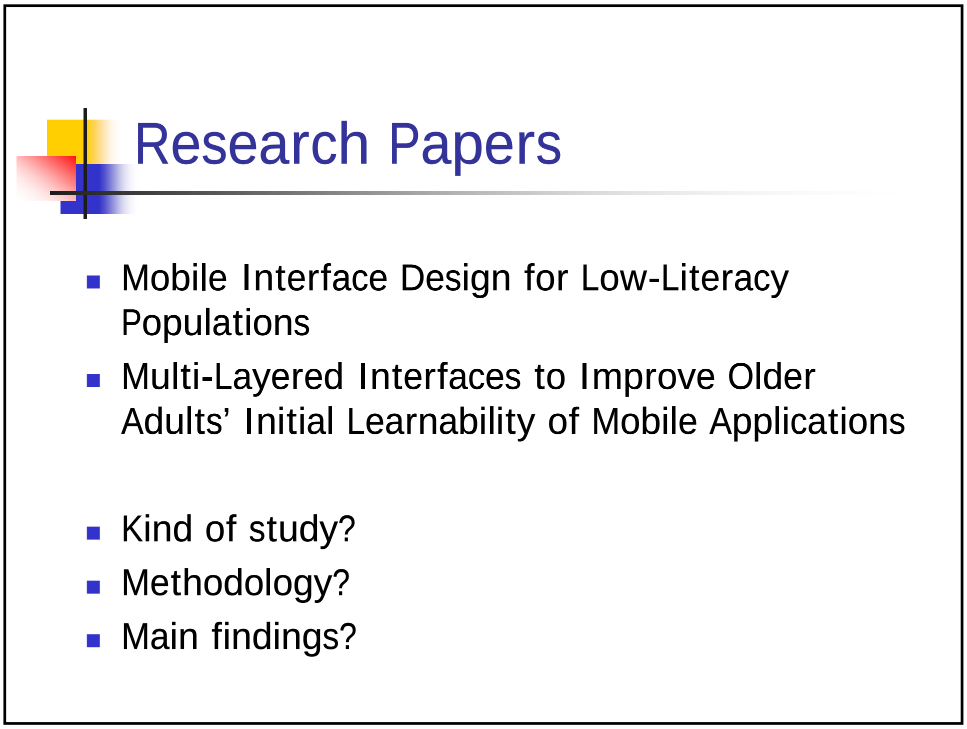
Exercise

- Break into teams
- Design a new myNEU portal* for an iPhone
  - Determine most important subset of tasks
  - Sketch conceptual design
  - Sketch main app page
  - How is your design different from a desktop app?

* *or other NU-related app*



Research on Mobile UIs

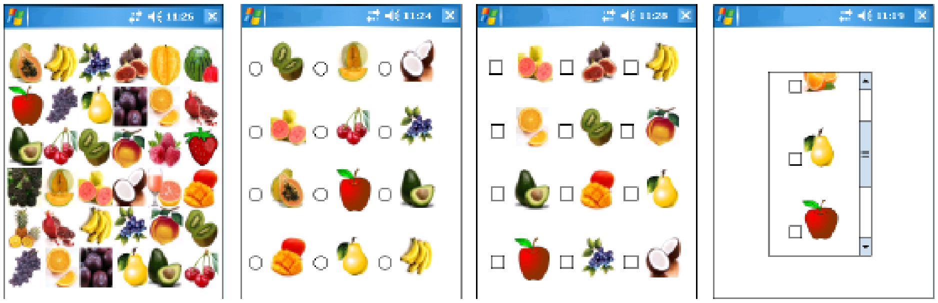


Research Papers

- Mobile Interface Design for Low-Literacy Populations
- Multi-Layered Interfaces to Improve Older Adults' Initial Learnability of Mobile Applications

- Kind of study?
- Methodology?
- Main findings?

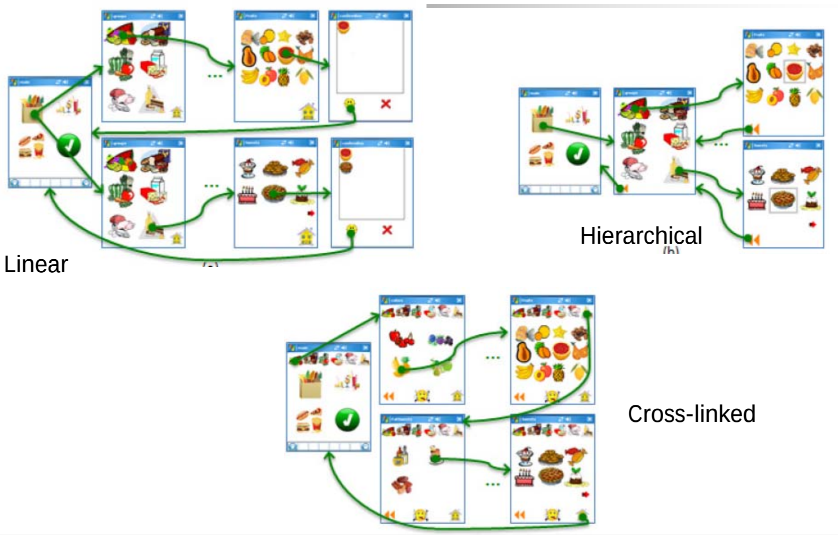
Mobile Interface Design for Low-Literacy Populations



Study 1 – which widget is best?

- Icon vs. Radio Button vs. Checkbox vs. Scrollbar x 3 sizes
- N=17, all below 9th grade reading (REALM)
- Within subjects
- Results
  - Radio buttons best (performance & pref)
  - Large widgets best (performance & pref)

Study 2 – which navigation structure is best?



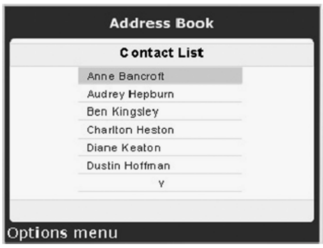
Study 2 – which navigation structure is best?

- N=19, low lit
- Users first trained on each interface
- Task = selecting a set of food items
- Results:
  - Linear is best (most tasks completed, most completed without error, recovered faster)
  - But – preferred cross-linked
  - Depth of 5, breadth 5-10 best (fewest errors)
  - Always provide BACK and HOME buttons

Multi-Layered Interfaces to Improve Older Adults' Initial Learnability of Mobile Applications

- "gray digital divide"
- Mobile devices require greater working memory (small UI, overloaded controls), which declines with age.
- Multi-Layered interface
  - "Training Wheels" aka scaffolding
  - Simplified interfaces decrease working memory load
  - May reduce abandonment of device

- N=16 older (65-81), 16 younger (21-36)
- Between subjects, stratified by age
  - ML: first master simple, then complex
  - Control: first master complex

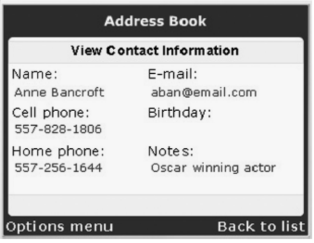


Address Book

Contact List

Anne Bancroft
Audrey Hepburn
Ben Kingsley
Charlton Heston
Diane Keaton
Dustin Hoffman
v

Options menu



Address Book

View Contact Information

Name:
Anne Bancroft

E-mail:
aban@email.com

Cell phone:
557-828-1806

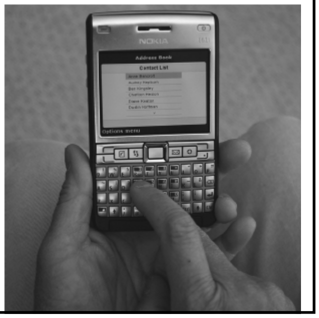
Birthday:

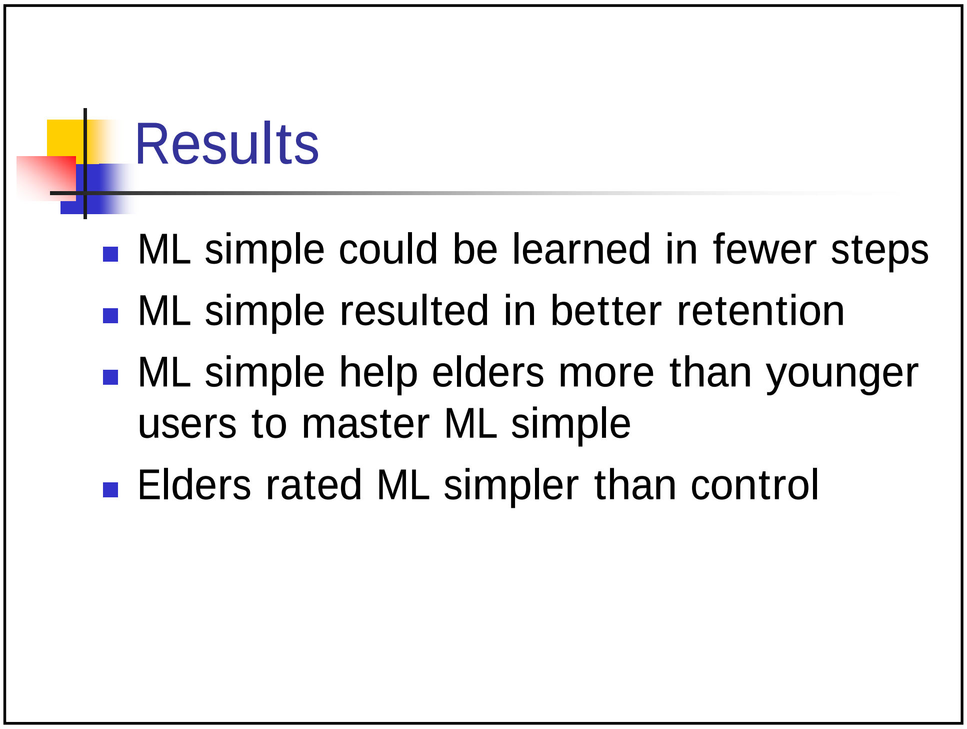
Home phone:
557-256-1644

Notes:
Oscar winning actor

Options menu

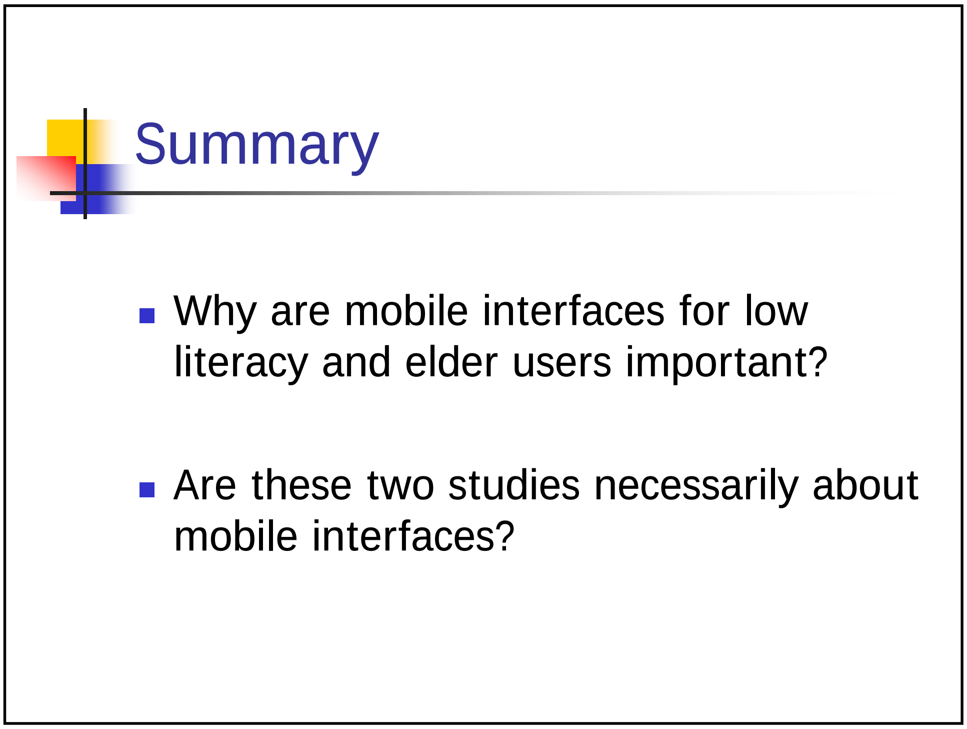
Back to list





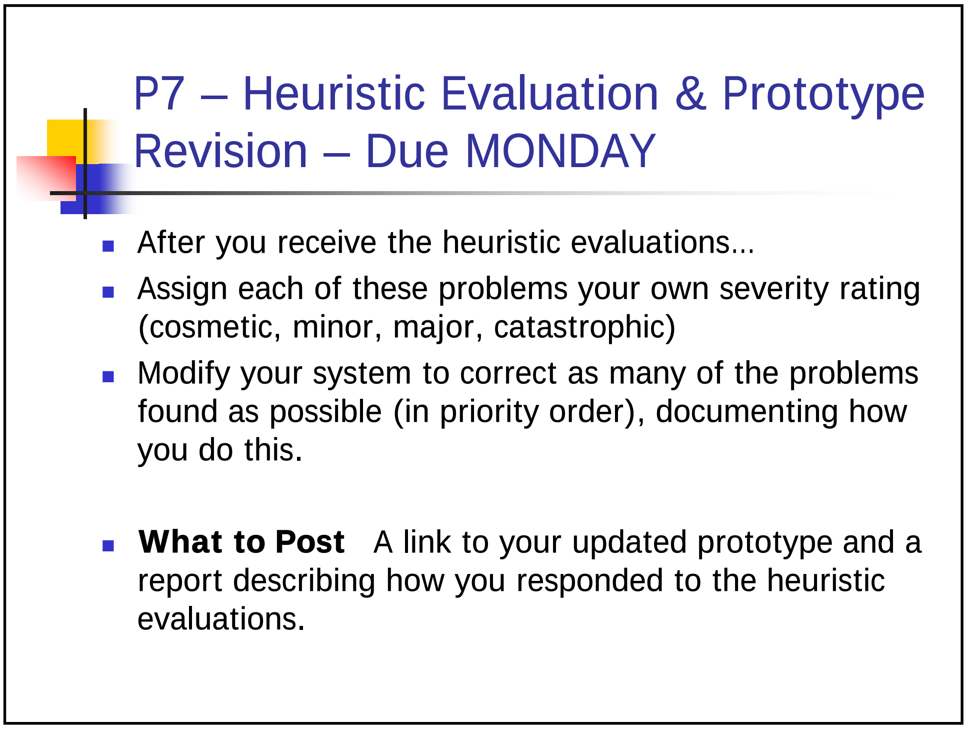
Results

- ML simple could be learned in fewer steps
- ML simple resulted in better retention
- ML simple help elders more than younger users to master ML simple
- Elders rated ML simpler than control



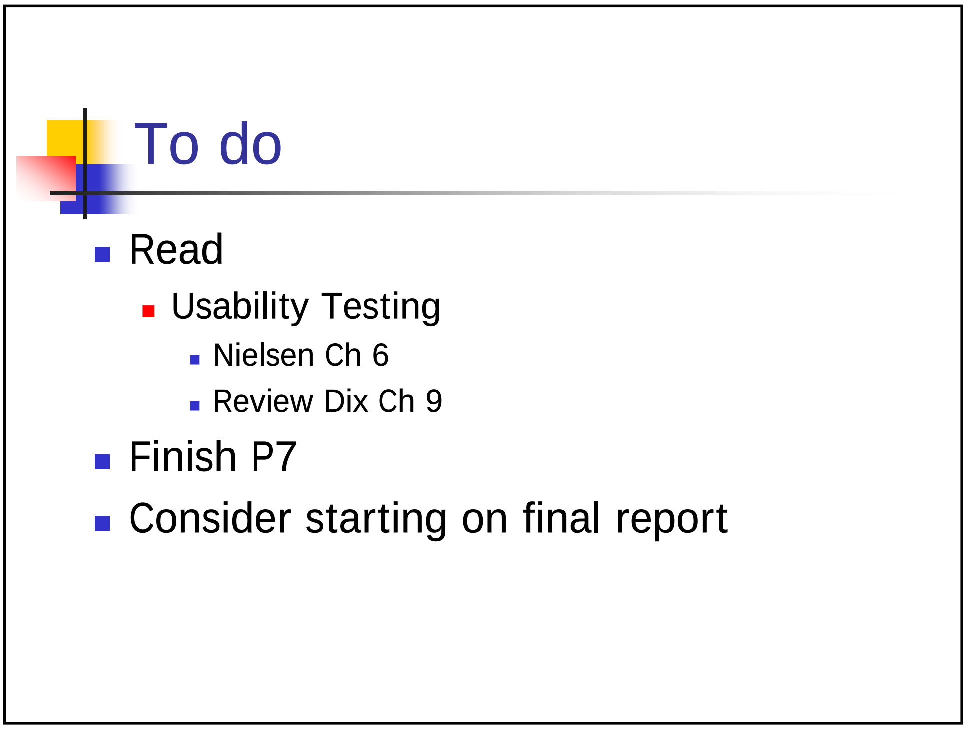
Summary

- Why are mobile interfaces for low literacy and elder users important?
- Are these two studies necessarily about mobile interfaces?



P7 – Heuristic Evaluation & Prototype Revision – Due MONDAY

- After you receive the heuristic evaluations…
- Assign each of these problems your own severity rating (cosmetic, minor, major, catastrophic)
- Modify your system to correct as many of the problems found as possible (in priority order), documenting how you do this.
- **What to Post** A link to your updated prototype and a report describing how you responded to the heuristic evaluations.



To do

- Read
  - Usability Testing
    - Nielsen Ch 6
    - Review Dix Ch 9
- Finish P7
- Consider starting on final report

UI & Mobile Apps

Mansoor Pervaiz

PHD student (Personal Health Informatics)

College of Computer & Information Science

Bouve' College of Health Sciences

Challenges

Higher Expectations

Do not know about gestures and are not interested to learn new ones

Find features by accident

Meaning of well designed icons can be lost

Challenges

Do not explore all screens or details

Use phones

- on treadmills
- in cars
- super markets

Short interactions

- 30 second sprints

Challenges

Small screen

Using one hand

HuffPost

Challenges

When designing: Center of our attention

When using: One app in millions

Hop from one app to another

Other apps interrupt our app

- Push notifications
- Phone calls
- Text messages

Challenges

The right tool for the right job

Figure out the minimum, build it

Challenges

The right tool for the right job

Figure out the minimum, build it

Polish Polish Polish

Why Polish?

User bored and disloyal

If the application does not hold their interest they move on

Why Polish?

User bored and disloyal

If the application does not hold their interest they move on

No Viral Marketing

Get it right the first time

Figure out what users need

Further challenges

Big appetite for apps

Download 10 apps per month

Don’t use an app for more than 20 times

Use less than 33% apps, 2 months after purchase

Design needs

Give importance to users’ habits

Users’ need front and center

Design needs

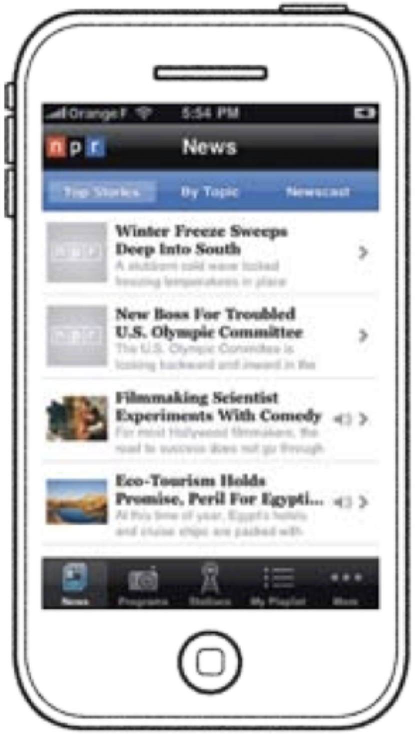
Give importance to users’ habits

Users’ need front and center

Harder than it looks

Design needs

What we build:



Winter Freeze Sweeps Deep Into South
A stubborn cold wave locked freezing temperatures in place

New Boss For Troubled U.S. Olympic Committee
The U.S. Olympic Committee is looking backward and inward in the

Filmmaking Scientist Experiments With Comedy
For most Hollywood filmmakers, the road to success does not go through

Eco-Tourism Holds Promise, Peril For Egypti...
At this time of year, Egypt's hotels and cruise ships are packed with

Design needs

What users see:

Design needs

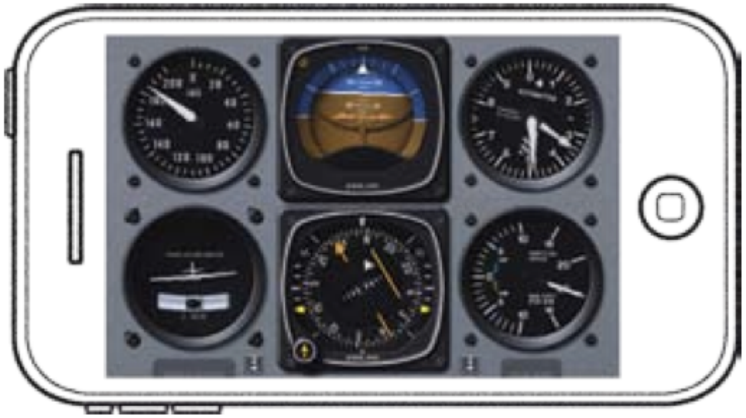
No intricate details

Design should be Large & Simple

Minimum taps should get the job done

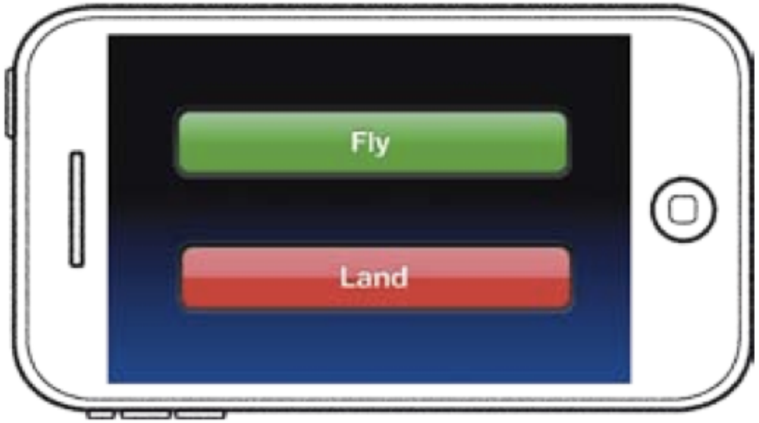
Design needs

Our flight simulator



Design needs

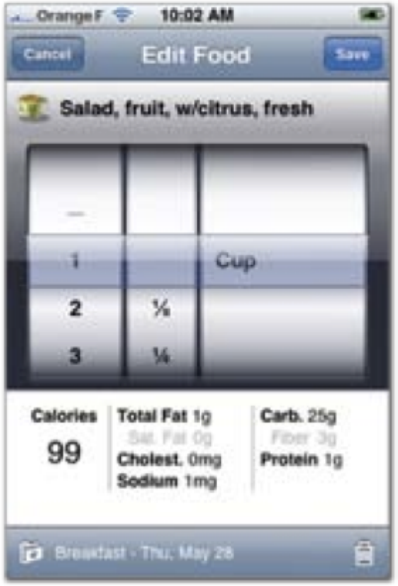
User Expectation



Design needs

Single finger gestures

Interfaces drawn on similar experiences



Salad, fruit, w/citrus, fresh

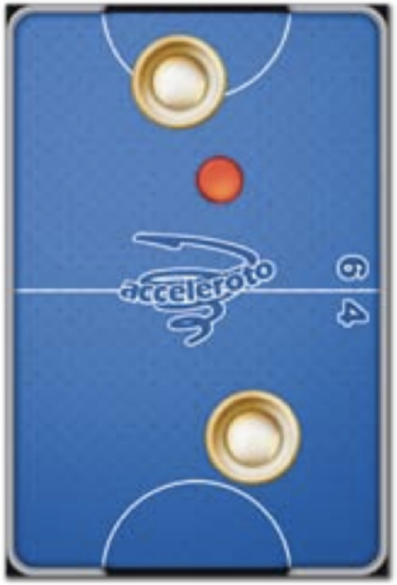
–		
1		Cup
2	⅛	
3	¼	

Calories 99

Total Fat 1g
Sat. Fat 0g
Cholest. 0mg
Sodium 1mg

Carb. 25g
Fiber 3g
Protein 1g

Breakfast - Thu, May 28



Design needs

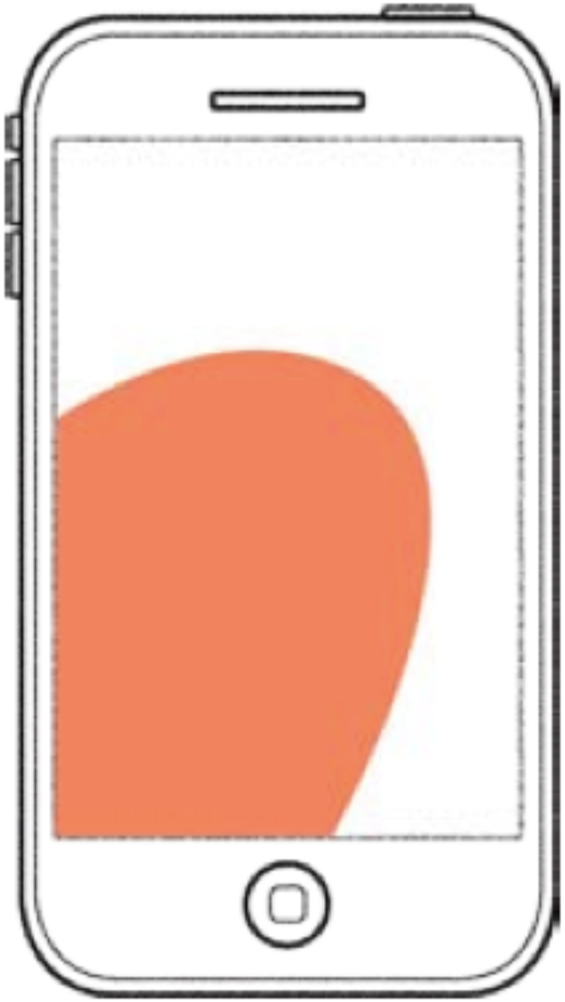
Clumsy fingers

Build for a thumb

Thumb can reach all parts of screen

But only third of it effortlessly

Interface design should focus on this area

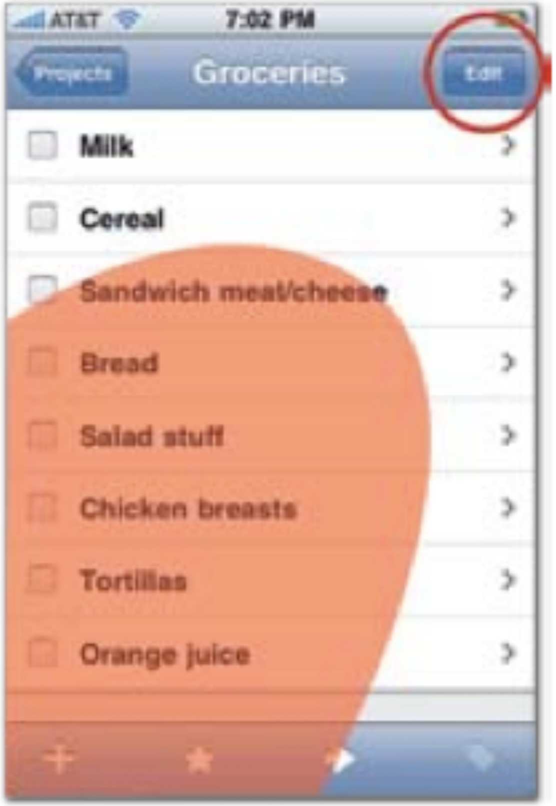


Build for a thumb

Frequently used buttons in hot zone

Buttons for changing/deleting

On top right to avoid accidental taps



□ Milk

□ Cereal

□ Sandwich meat/cheese

□ Bread

□ Salad stuff

□ Chicken breasts

□ Tortillas

□ Orange juice

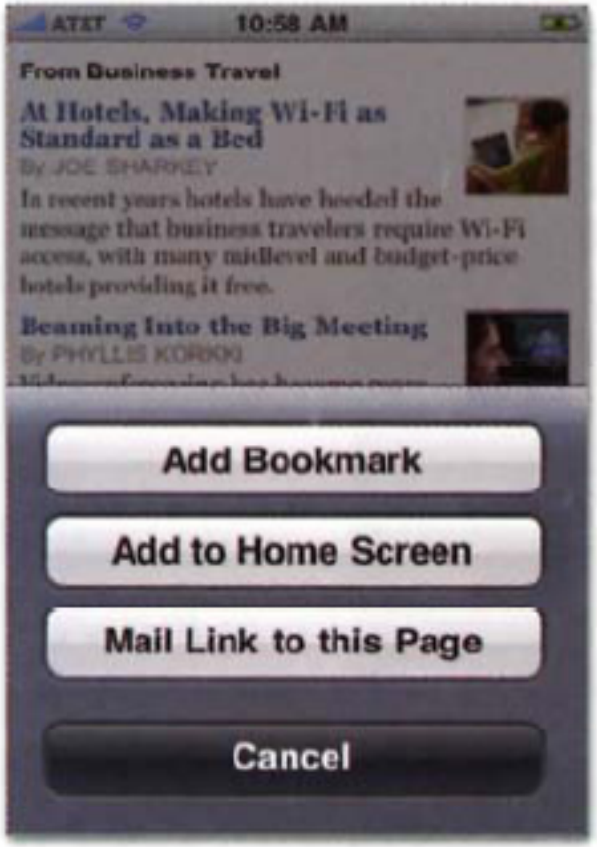
Left handed users?

“Swap” feature for left handed users

Left handed users?

“Swap” feature for left handed users

Better solution: full width buttons



From Business Travel

At Hotels, Making Wi-Fi as Standard as a Bed

By JOE SHARKEY

In recent years hotels have heeded the message that business travelers require Wi-Fi access, with many midlevel and budget-price hotels providing it free.

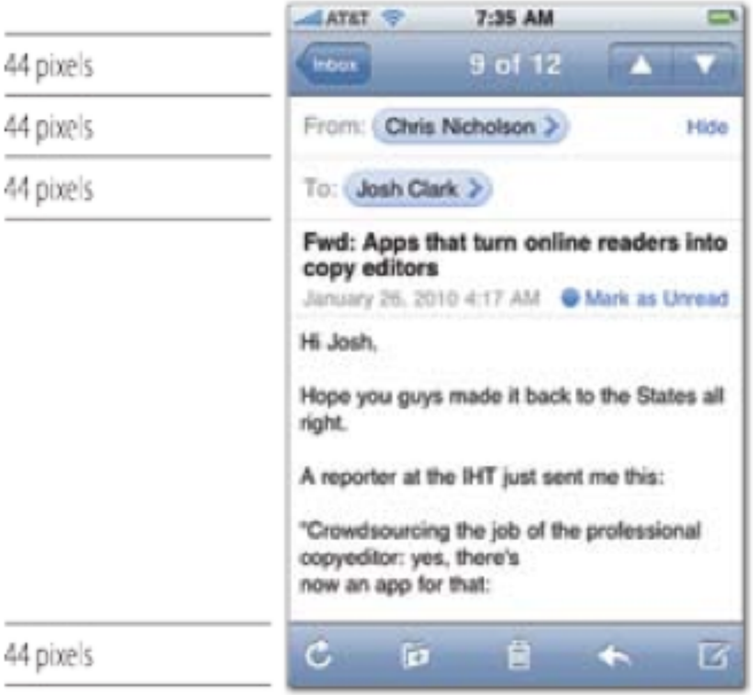
Beaming Into the Big Meeting

By PHYLLIS KORKKI

Magic Number

Average fingertip: 44 pixels

Apple keyboard 44x30
(absolute minimum)



From: Chris Nicholson

To: Josh Clark

Fwd: Apps that turn online readers into copy editors

January 26, 2010 4:17 AM

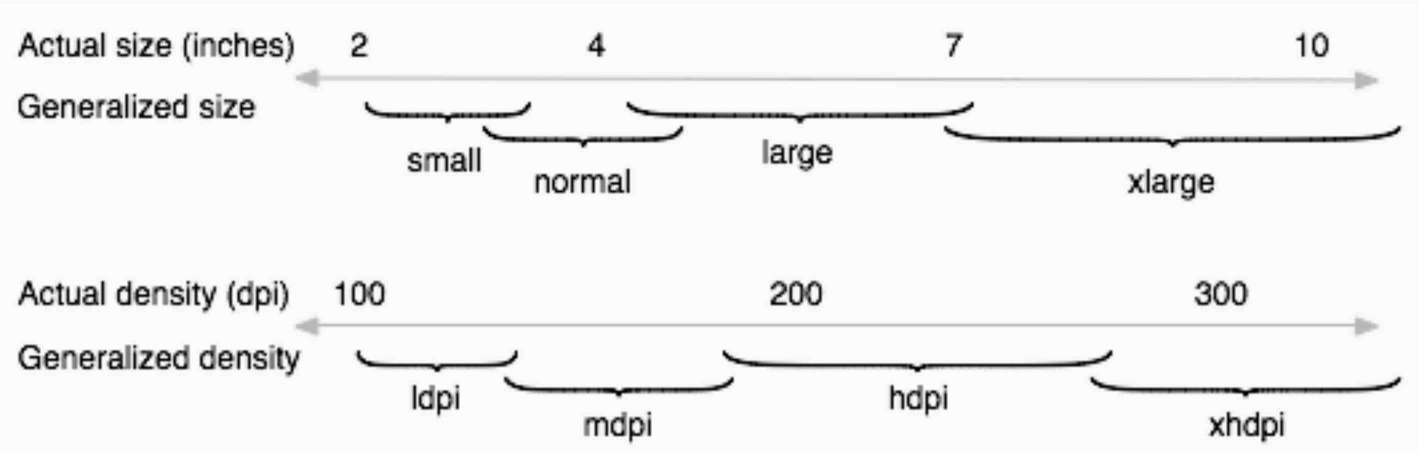
Hi Josh,

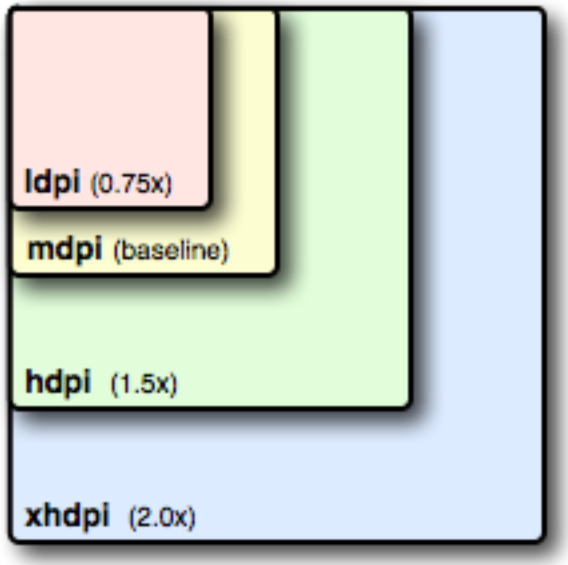
Hope you guys made it back to the States all right.

A reporter at the IHT just sent me this:

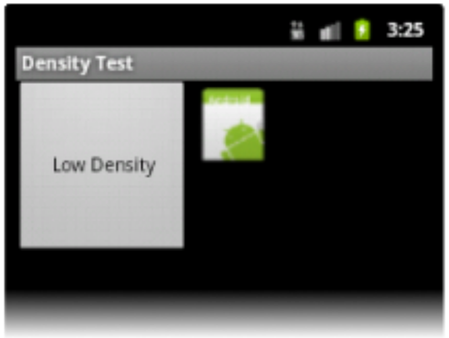
"Crowdsourcing the job of the professional copyeditor: yes, there's now an app for that:

Different px densities



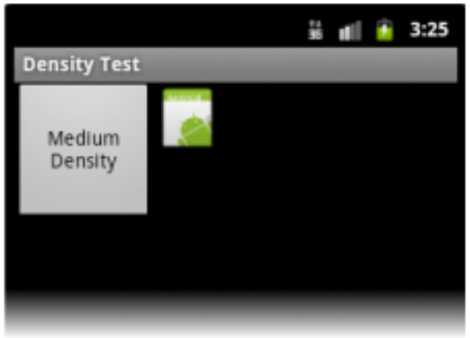


Different px densities


3:25

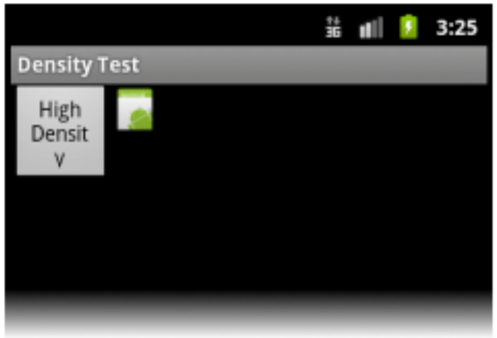
Density Test

Low Density


3:25

Density Test

Medium Density


3:25

Density Test

High Densit y

Different px densities

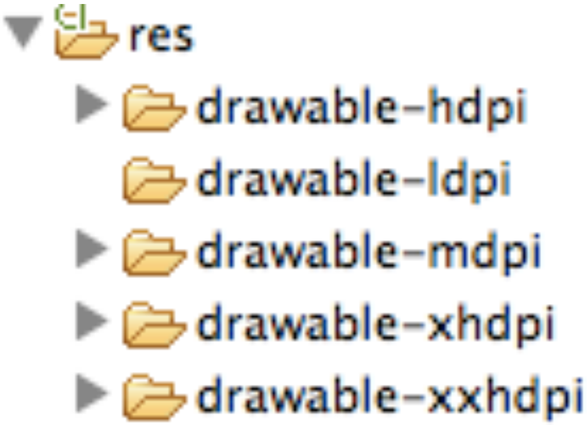
DP (Density Independent Pixels) – images

SP (Scale Independent Pixels) – fonts

Different px densities

DP (Density Independent Pixels) – images

SP (Scale Independent Pixels) – fonts

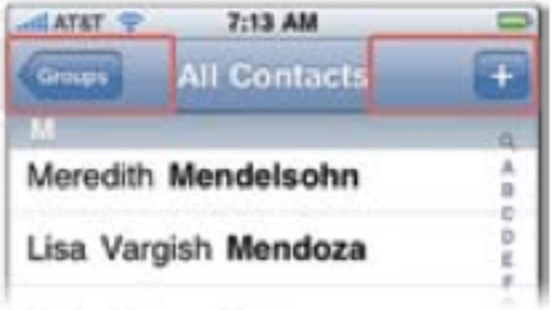


res
- drawable-hdpi
- drawable-ldpi
- drawable-mdpi
- drawable-xhdpi
- drawable-xxhdpi

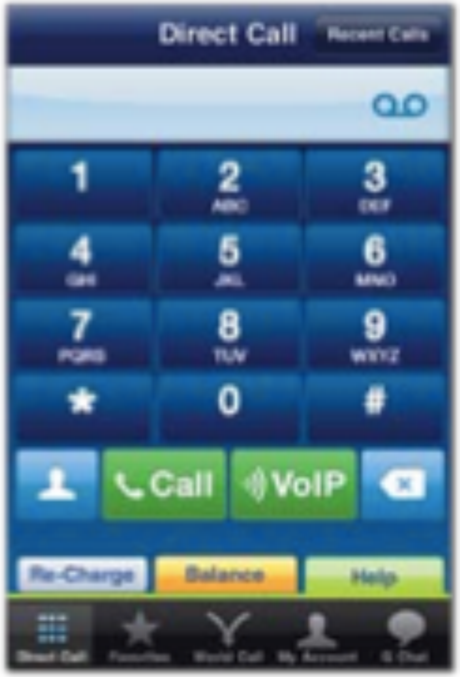
Buttons vs Tap Areas

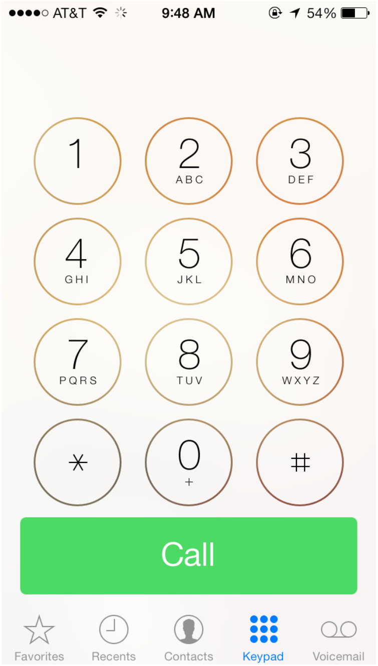
Button image != tap area

44 x 44 should be enforces for tap areas



Don’t overcrowd

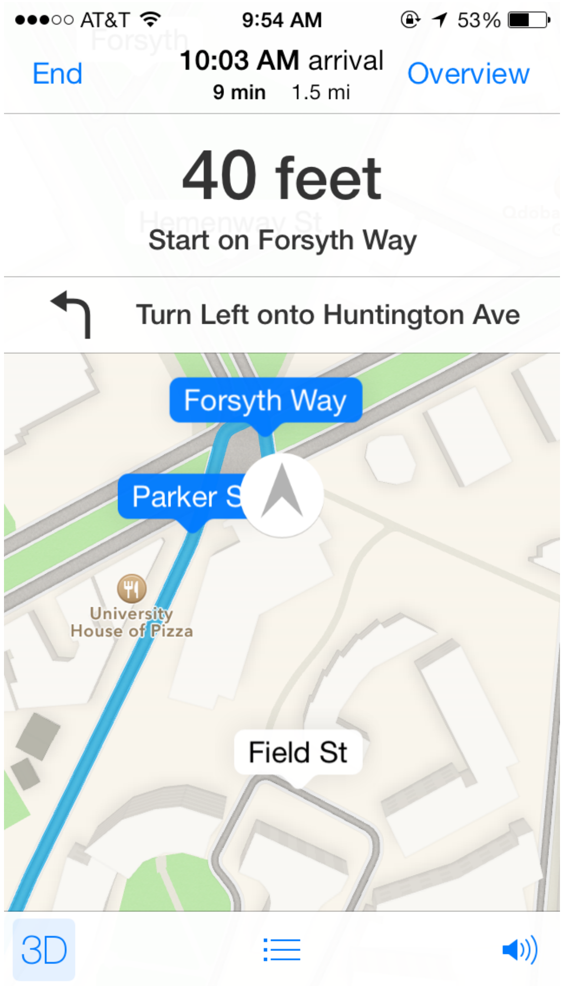




Organize as Top Down

Important info at the top

Primary buttons at bottom



10:03 AM arrival

9 min 1.5 mi

40 feet

Start on Forsyth Way

Turn Left onto Huntington Ave

No scrolling

Taking in content should be as effortless as possible

Scrolling takes effort

Absorbing new content as you scroll takes effort

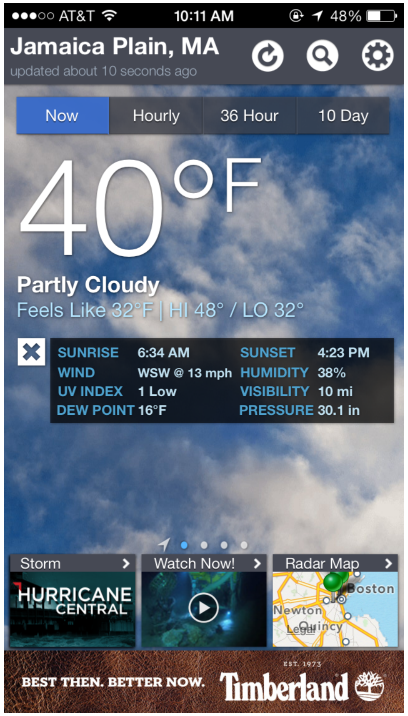
Act of realizing that you need to scroll takes effort

Single screen

No need to display everything

Just show the basics

Clarity trumps density



●●●○○ AT&T 10:11 AM 48%

Jamaica Plain, MA

updated about 10 seconds ago

Now | Hourly | 36 Hour | 10 Day

40°F

Partly Cloudy

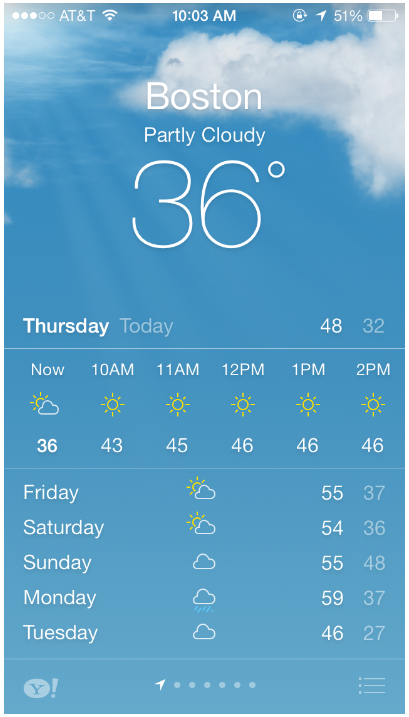
Feels Like 32°F | HI 48° / LO 32°

SUNRISE	6:34 AM	SUNSET	4:23 PM
WIND	WSW @ 13 mph	HUMIDITY	38%
UV INDEX	1 Low	VISIBILITY	10 mi
DEW POINT	16°F	PRESSURE	30.1 in

Storm › HURRICANE CENTRAL

Watch Now! ›

Radar Map › Boston Newton Quincy



●●●○○ AT&T 10:03 AM 51%

Boston

Partly Cloudy

36°

Thursday Today 48 32

Now	10AM	11AM	12PM	1PM	2PM
36	43	45	46	46	46

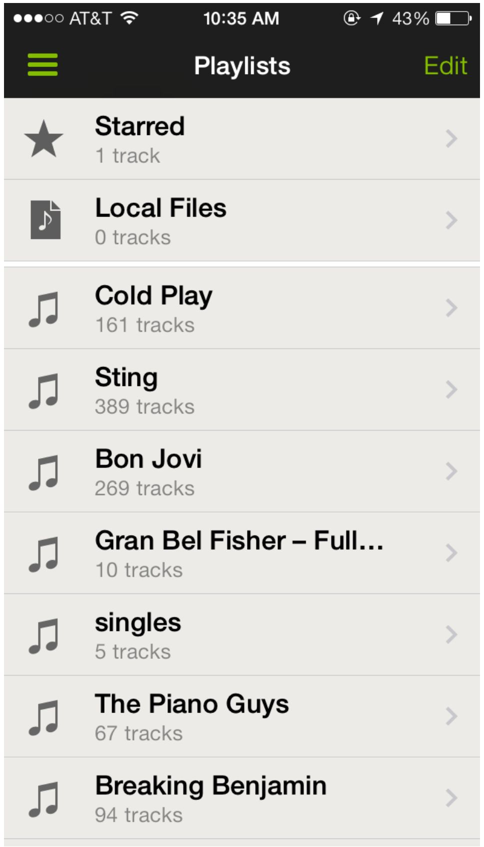
Friday	55	37
Saturday	54	36
Sunday	55	48
Monday	59	37
Tuesday	46	27

Simple vs Advance

Balance between simplicity and supply all tools to the user

Advanced tools shouldn’t clutter experience

Advance Features



●●●○○ AT&T 10:35 AM 43%

Playlists Edit

Starred
1 track

Local Files
0 tracks

Cold Play
161 tracks

Sting
389 tracks

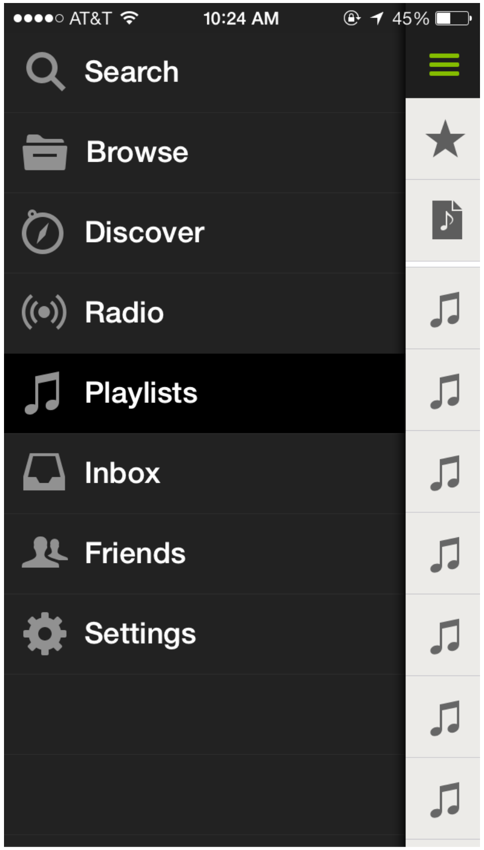
Bon Jovi
269 tracks

Gran Bel Fisher – Full…
10 tracks

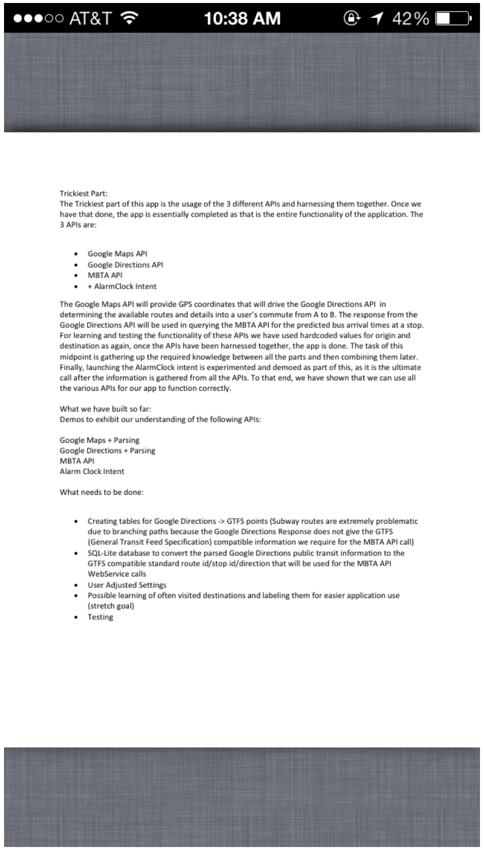
singles
5 tracks

The Piano Guys
67 tracks

Breaking Benjamin
94 tracks



Advance Features



Trickiest Part:
The Trickiest part of this app is the usage of the 3 different APIs and harnessing them together. Once we have that done, the app is essentially completed as that is the entire functionality of the application. The 3 APIs are:

- Google Maps API
- Google Directions API
- MBTA API
- + AlarmClock Intent

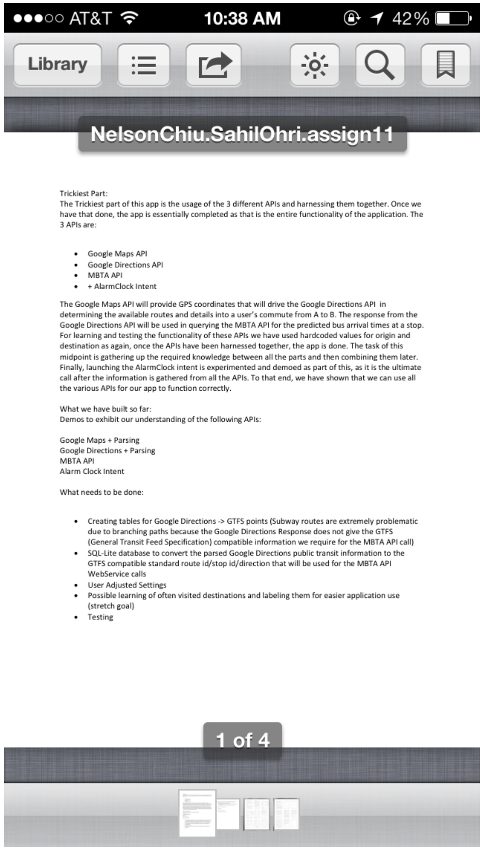
The Google Maps API will provide GPS coordinates that will drive the Google Directions API in determining the available routes and details into a user's commute from A to B. The response from the Google Directions API will be used in querying the MBTA API for the predicted bus arrival times at a stop. For learning and testing the functionality of these APIs we have used hardcoded values for origin and destination as again, once the APIs have been harnessed together, the app is done. The task of this midpoint is gathering up the required knowledge between all the parts and then combining them later. Finally, launching the AlarmClock intent is experimented and demoed as part of this, as it is the ultimate call after the information is gathered from all the APIs. To that end, we have shown that we can use all the various APIs for our app to function correctly.

What we have built so far:
Demos to exhibit our understanding of the following APIs:

Google Maps + Parsing
Google Directions + Parsing
MBTA API
Alarm Clock Intent

What needs to be done:

- Creating tables for Google Directions -> GTFS points (Subway routes are extremely problematic due to branching paths because the Google Directions Response does not give the GTFS (General Transit Feed Specification) compatible information we require for the MBTA API call)
- SQL-Lite database to convert the parsed Google Directions public transit information to the GTFS compatible standard route id/stop id/direction that will be used for the MBTA API WebService calls
- User Adjusted Settings
- Possible learning of often visited destinations and labeling them for easier application use (stretch goal)
- Testing



Trickiest Part:
The Trickiest part of this app is the usage of the 3 different APIs and harnessing them together. Once we have that done, the app is essentially completed as that is the entire functionality of the application. The 3 APIs are:

- Google Maps API
- Google Directions API
- MBTA API
- + AlarmClock Intent

The Google Maps API will provide GPS coordinates that will drive the Google Directions API in determining the available routes and details into a user's commute from A to B. The response from the Google Directions API will be used in querying the MBTA API for the predicted bus arrival times at a stop. For learning and testing the functionality of these APIs we have used hardcoded values for origin and destination as again, once the APIs have been harnessed together, the app is done. The task of this midpoint is gathering up the required knowledge between all the parts and then combining them later. Finally, launching the AlarmClock intent is experimented and demoed as part of this, as it is the ultimate call after the information is gathered from all the APIs. To that end, we have shown that we can use all the various APIs for our app to function correctly.

What we have built so far:
Demos to exhibit our understanding of the following APIs:

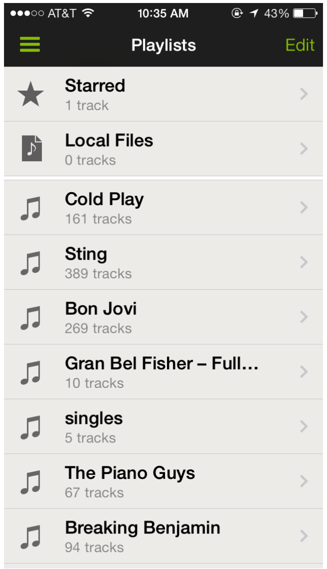
Google Maps + Parsing
Google Directions + Parsing
MBTA API
Alarm Clock Intent

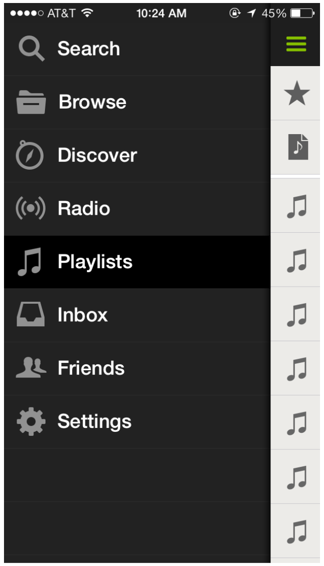
What needs to be done:

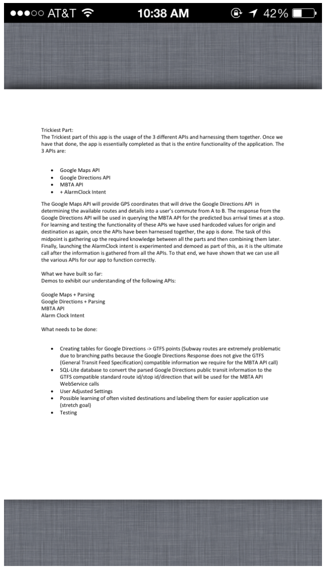
- Creating tables for Google Directions -> GTFS points (Subway routes are extremely problematic due to branching paths because the Google Directions Response does not give the GTFS (General Transit Feed Specification) compatible information we require for the MBTA API call)
- SQL-Lite database to convert the parsed Google Directions public transit information to the GTFS compatible standard route id/stop id/direction that will be used for the MBTA API WebService calls
- User Adjusted Settings
- Possible learning of often visited destinations and labeling them for easier application use (stretch goal)
- Testing

Easy way out

Make it simple to toggle between Advance and Regular layout







Trickiest Part:
The Trickiest part of this app is the usage of the 3 different APIs and harnessing them together. Once we have that done, the app is essentially completed as that is the entire functionality of the application. The 3 APIs are:

- Google Maps API
- Google Directions API
- MBTA API
- + AlarmClock Intent

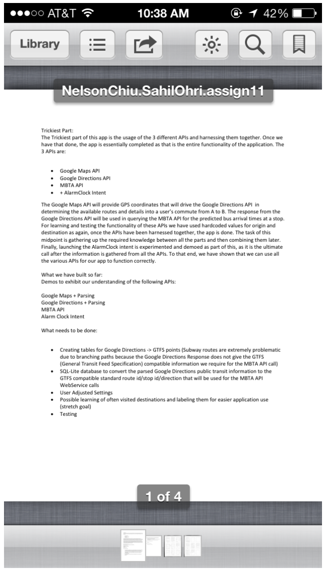
The Google Maps API will provide GPS coordinates that will drive the Google Directions API in determining the available routes and details into a user's commute from A to B. The response from the Google Directions API will be used in querying the MBTA API for the predicted bus arrival times at a stop. For learning and testing the functionality of these APIs we have used hardcoded values for origin and destination as again, once the APIs have been harnessed together, the app is done. The task of this midpoint is gathering up the required knowledge between all the parts and then combining them later. Finally, launching the AlarmClock intent is experimented and demoed as part of this, as it is the ultimate call after the information is gathered from all the APIs. To that end, we have shown that we can use all the various APIs for our app to function correctly.

What we have built so far:
Demos to exhibit our understanding of the following APIs:

Google Maps + Parsing
Google Directions + Parsing
MBTA API
Alarm Clock Intent

What needs to be done:

- Creating tables for Google Directions -> GTFS points (Subway routes are extremely problematic due to branching paths because the Google Directions Response does not give the GTFS (General Transit Feed Specification) compatible information we require for the MBTA API call)
- SQL-Lite database to convert the parsed Google Directions public transit information to the GTFS compatible standard route id/stop id/direction that will be used for the MBTA API WebService calls
- User Adjusted Settings
- Possible learning of often visited destinations and labeling them for easier application use (stretch goal)
- Testing



Trickiest Part:
The Trickiest part of this app is the usage of the 3 different APIs and harnessing them together. Once we have that done, the app is essentially completed as that is the entire functionality of the application. The 3 APIs are:

- Google Maps API
- Google Directions API
- MBTA API
- + AlarmClock Intent

The Google Maps API will provide GPS coordinates that will drive the Google Directions API in determining the available routes and details into a user's commute from A to B. The response from the Google Directions API will be used in querying the MBTA API for the predicted bus arrival times at a stop. For learning and testing the functionality of these APIs we have used hardcoded values for origin and destination as again, once the APIs have been harnessed together, the app is done. The task of this midpoint is gathering up the required knowledge between all the parts and then combining them later. Finally, launching the AlarmClock intent is experimented and demoed as part of this, as it is the ultimate call after the information is gathered from all the APIs. To that end, we have shown that we can use all the various APIs for our app to function correctly.

What we have built so far:
Demos to exhibit our understanding of the following APIs:

Google Maps + Parsing
Google Directions + Parsing
MBTA API
Alarm Clock Intent

What needs to be done:

- Creating tables for Google Directions -> GTFS points (Subway routes are extremely problematic due to branching paths because the Google Directions Response does not give the GTFS (General Transit Feed Specification) compatible information we require for the MBTA API call)
- SQL-Lite database to convert the parsed Google Directions public transit information to the GTFS compatible standard route id/stop id/direction that will be used for the MBTA API WebService calls
- User Adjusted Settings
- Possible learning of often visited destinations and labeling them for easier application use (stretch goal)
- Testing

Summary

Challenges

Design Concerns

Challenges

Very high expectations of ease of use

Apps have to compete for user's attention

Short interactions

Just do the absolute minimum and polish it again and again

Design Concern

Less is more

Assuming single finger (thumb)

Show the least about of information

Do not clutter

Easy to use